Motivational and Metacognitive Feedback in an ITS: Linking past states and experiences to current problems

Alison Hull

Thesis submitted for Degree of Doctor of Philosophy

University of Sussex

September 2014

University of Sussex

Thesis submitted by Alison Hull for the Degree of Doctor of Philosophy

Motivational and Metacognitive Feedback in an ITS: Linking past states and experiences to current problems

Summary

Feedback is an important element in learning as it can provide learners with both information about progress as well as external motivational stimuli, providing them with an opportunity for reflection. Motivation and metacognition are strongly intertwined, with learners high in self-efficacy more likely to use a variety of self-regulatory learning strategies, as well as to persist longer on challenging tasks. Learning from past experience involves metacognitive processes as an act of reflecting upon one's own experience and, coupled with existing knowledge, aids the acquisition and construction of further knowledge.

The aim of the research was to improve the learner's focus on the process and experience of problem solving while using an Intelligent Tutoring System (ITS), by addressing the primary question: what are the effects of including motivational and metacognitive feedback based on the learner's past states and experiences? An existing ITS, SQL-Tutor, was used in a study with participants from first year undergraduate degrees studying a database module. The study used two versions of SQL-Tutor: the Control group used a base version providing domain feedback and the Study group used an extended version that also provided motivational and metacognitive feedback.

Three sources of data collection were used: module summative assessments, ITS log files and a post-study questionnaire. The analysis included both pre-post comparisons and how the participants interacted with the system, for example their persistence in problem-solving and the degree to which they referred to past learning. Comparisons between groups showed some differing trends both in learning and behaviour in favour of the Study group, though these trends were not significantly different. The study findings showed promise for the use of motivational and metacognitive feedback based on the learners' past states and experiences that could be used as a basis for future research work and refinement.

Acknowledgements

The time spent on doing this research degree project has definitely been a journey full of a mixture of flat, smooth roads, steep hills and the odd ditch. There has been a wealth of skills learnt, learnt anew or further strengthened throughout the whole journey. I would like to take this opportunity to express my thanks and gratitude to the following people who played important roles along the way.

From the University of Sussex, first and foremost, a huge thank you to my PhD supervisor Benedict du Boulay, who has been there through the whole journey. For his foresight in agreeing to supervise myself as a self-funded, part-time student who also had a full time job that could be very demanding most of the time. It was Benedict's excellent reputation in the research area of artificial intelligence in education that led me to first approach him and the University of Sussex with the idea for the research project. So thank you Benedict for your knowledge, your ability to motivate, and your patience and understanding throughout. I would also like to thank Nicola Yuill, from the Psychology department in the University, who was part of my Annual Review Committee. Nicola always gave positive and valid feedback and encouragement during the annual process. Also to Pablo Romero for being part of my Annual Review Committee and for his feedback and support during the first few years of this journey.

From other universities, a big thank you goes to Tanja Mitrovic and the members of the Intelligent Computer Tutoring Group (ICTG) at the University of Canterbury, Christchurch, New Zealand. She allowed me to use SQL-Tutor as the base tutoring system in this project and provided help and ongoing encouragement. Thank you to ICTG member Moffat Mathews for allowing me to use his previous work on problem templates as a basis for relating problems. Thank you to another ICTG member Jay Holland who had the task of putting my SQL-Tutor code changes live. As a quick aside, SQL-Tutor is written using Allegro Common Lisp (CL) from Franz Inc, so I would like to thank Franz Inc for giving me an extension on the Allegro CL license to cover a six month period to span the final thesis write-up through to the viva. I would also like to thank former colleagues at the University of Northumbria: Emma-Jane Phillips and Malcolm Scott for allowing me to approach the students to participate in the study. Emma-Jane as the database module tutor, thank you for giving me lecture time and access to the module materials. Thank you also to the students who volunteered and participated in the study.

I also want to say a big thank you to a group of good friends who have helped and been there along the way. To Kenneth Fearis for above all giving me the extremely valuable gift of 'time', especially when it was needed the most. To Jason Hunt for the tsunami of 'positive waves'[1] and for Harry (the four and half foot tall teddy bear, it's a long story!). To Lynne Newall for her insight and on occasions putting up with me 'thinking out loud'. To the two generous people who agreed to proof read this thesis in a very short period of time: Lynne Newall and Frank Hindle. Frank, thank you also for the chats over coffee throughout the years.

Last, but definitely not least, I would like to give my biggest thank you to my parents, Ron and Margaret Hull. For their continued, unwavering and unconditional love, support and belief in everything I do and for everything that I have yet to dream of doing. Thank you.

[1] From the 1970 film "Kelly's Heroes", directed by Brian G. Hutton and starring Clint Eastwood, Telly Savalas, Don Rickles, Carroll O'Connor, and Donald Sutherland. Positive and negative waves are from Donald Sutherland's character, Sergeant "Oddball".

Table of Contents

List of Tables

List of Figures

List of Publications

Hull, A (2007) Use of feedback guided by past learner actions in a pedagogical agent to aid the motivational state and the meta-cognitive process involved in problem solving in a programming domain, *10th Human-Centred Technology Postgraduate Workshop*, University of Sussex, 6 December 2007

Hull, A, du Boulay, B (2008) Cognitive and Motivational Feedback in Learning Programming, *Proceedings of WIP-PPIG, 4th Annual Work-in-Progress Meeting of the Psychology of Programming Interest Group*, University of Sussex, 21-22 February 2008

Hull, A. & du Boulay, B. (2009). Scaffolding Motivation and Metacognition in Learning Programming. In V. Dimitrova, R. Mizoguchi, B. du Boulay & A. Graesser (Eds.), Proceedings of the 2009 conference on *Artificial Intelligence in Education. Building Learning Systems that Care: from Knowledge Representation to Affective Modelling,* pp. 755-756. Amsterdam: IOS Press.

Hull, A. & du Boulay, B. (2011). Motivational and Metacognitive Feedback: Linking the Past to the Present. In G. Biswas, S. Bull, J. Kay & A. Mitrovic (Eds.), *Artificial Intelligence in Education: 15th International Conference, AIED 2011* (Lecture Notes in Artificial Intelligence No. 6738 pp. 600-602). Berlin: Springer.

1 Introduction

1.1 Background

Motivation and metacognition are key factors that influence learning across any domain. Motivation can directly influence the learner's self-regulatory control and the level of success in their learning outcome (Hodges, 2004; Schunk, Pintrich and Meece, 2007). Motivation involves and influences cognition, with this relationship being reciprocal. Metacognition allows learners to gain an insight into their own thinking, as well as promoting independent learning (Paris and Winograd, 1990).

Contemporary theories of motivation all agree that motivation involves and influences cognition, with the relationship being reciprocal, but they differ in the labels given to the constructs and the nature of the underlying processes involved (Schunk et al., 2007). In motivation, a construct "is some postulated attribute of people" (Cronbach and Meehl, 1955), referring to a complex psychological concept. Motivation can be defined as "the process whereby goal-directed activity is instigated and sustained" (Schunk et al., 2007), though the constructs used in motivation theories are not limited to 'goals' and use similar constructs that overlap across the theories. One construct that is an important mediator across all theories of motivation is that of *self-efficacy* (Schunk et al., 2007). Research has shown that self-efficacy beliefs are correlated with other motivation constructs, such as attributions, goal setting, problem solving, self-regulation, social comparisons, strategy training and expectancy across domains (Pajares, 1997).

Metacognition is an area of psychology research that has been defined as "knowledge concerning one's own cognitive processes and products or anything related to them" (Flavell, 1976). The influence of motivation and metacognition on learning is not isolated, with strongly entwined relationships existing between them. Learners high in self-efficacy are generally more likely to use "various cognitive and self-regulatory learning strategies" (Schunk et al., 2007). Likewise metacognitive skills are required for motivation (e.g. mastery in goal theory requires insight into one's own knowledge and experience).

Reflecting and drawing upon prior experience and knowledge are important in the construction of knowledge in terms of using and further developing mental representations and cognitive relationships in memory (Mayer, 2003). The use of prior experience and knowledge is an important aspect in learning and solving similar problems. Learning from past experience involves metacognitive processes such as "reflection on experience" (Boreham, 1987), where the learner must learn how to learn (Weinberg, 1971).

This thesis has focussed on those key factors that influence learning, motivation and metacognition, to focus on using the learner's self-efficacy and their experiences to provide tailored feedback in an *intelligent tutoring system* (ITS). An ITS is a computer-based tutoring system that can adapt to each learner as an individual by maintaining a learner model, which is the system's view of what the learner knows (their level of domain knowledge). A motivationally aware ITS may deal with different motivational constructs, but at two levels: measuring the motivational state of the learner and techniques utilised to influence the motivational state of the learner in order to promote learning. The thesis has focussed both on providing motivational and metacognitive feedback based on the previous states and experiences of the learner in order to improve the learner's focus, and on experience of the learning process.

The overall methodology was to compare the educational effects of two versions of an ITS; one being motivationally aware and the other not. An existing ITS was used as the base version, which was modified in order to provide the required feedback, as well as to gauge the learner's self-efficacy levels via a self-reporting mechanism. The base ITS used in this thesis was SQL-Tutor, which is a learn-by-doing problem solving environment for undergraduates dealing with the domain of *Structured Query Language* (SQL) SELECT queries. SQL is a declarative language (Kenny and Pahl, 2005) designed for database programming, although there are extensions that provide procedural functionality. Learners often find solving SQL problems quite difficult as they not only have to know the syntax and concepts of SQL, but they must also understand the relational database model, as well as the specific schema of the database which the problem relates to (Mitrovic, 2012).

1.2 Research Questions

This research is grounded in the constructivist approach to learning, as it promotes learning from past experiences, learning from errors and actively involving the learners (the study participants) in constructing knowledge, with such knowledge being acquired recursively (Ben-Ari, 2001). This research explores incorporating motivational and metacognitive feedback based on the learner's past states and experiences in order to improve the learner's focus on the process and experience of problem solving while using an Intelligent Tutoring System (ITS). The primary research question has been divided into three supporting research questions.

> What are the effects on the learner of including motivational and metacognitive feedback based on the learner's past states and experiences?
>
> - Does providing such feedback lead to any measurable learning gains?
> - Does providing such feedback lead to any measurable gains in learner focus?
> - Do the learners perceive any benefit from using an ITS to aid their learning?

Where *feedback* is used in the supporting research questions, this refers specifically to providing motivational and metacognitive feedback based on the learner's past states and experiences. *Experiences* refers to the interactions that the learner has had with the ITS. *Learner focus* in the scope of this thesis refers to the participant's focus on the task at hand, so using SQL-Tutor to help them to practice and develop their SQL SELECT capability. Areas examined in order to determine learner focus were effort, persistence and learning behaviour.

1.3 Methodology

A study was designed and conducted where the participants were divided into one of two groups: the Control and Study groups. The Control group used a version of SQL-Tutor that only gave the base version of domain-level feedback, while the Study group used a version of SQL-Tutor that provided the same domain-level feedback as well as motivational and metacognitive feedback based on the past states and experiences of the learner. Comparing two versions of an ITS that differ only in a particular feature is a well-known and much used tactic in artificial intelligence in education (AIED) evaluation research. For two examples of this technique, see Mark and Greer (1995), and Luckin and du Boulay (1999). In order to gauge the learner's self-efficacy level, the Study group version of the tutor also prompted the learner to indicate their self-efficacy level via self-reports.

The participants in the study were first year students studying a database module taught on several undergraduate degree courses at the University of Northumbria. The study was embedded into the University's module delivery and used their summative assessments as pre-and post-tests. The study was designed to be an open, long-lasting study with no fixed contact points, where the participants were asked to use SQL-Tutor over a period of three and a half months. This period of time was designed to allow the features in the Study group version, which related to past problems and metacognition to be fully utilised. It also fitted in with the academic calendar. Refer to Appendix A (p.200) for the study's timeline.

As already mentioned, the study used the University's own summative assessments as a source of data. Two other sources of data were used: the log files produced by SQL-Tutor (both the base log files and additional ones that were added for this study) and a post-study questionnaire that was used to gauge the participants' perceptions of any benefits of using an ITS to aid their learning.

1.4 Outline of Thesis

Chapter 2 Motivation, Metacognition & Feedback

This chapter discusses contemporary theories of motivation in relation to learning before discussing the role of motivation and metacognition in learning. The role of feedback in learning is then covered, followed by the commonly used measures for measuring motivational state and the use of metacognitive skills. The problems that learners face when learning problem solving within a programming domain is also discussed. Finally, several ITSs are compared, paying particular attention to motivation and metacognition.

Chapter 3 System Design

As developing a new ITS from the very start was far beyond the scope of the research thesis, an existing ITS called SQL-Tutor was used as the base system. This chapter outlines the functionality that is core to SQL-Tutor before going onto discuss the modifications that were made to SQL-Tutor in order to accommodate the functionality required for the aim of this study. Two broad categories of modification were made: modifications that were implemented and made available in the two versions used by the Control and Study groups, and modifications that were available only to the version of SQL-Tutor that was used by the Study group.

Chapter 4 Study Design and Implementation

This chapter details the design and implementation of the study, covering who the participants were and the volunteering process, the procedure and timeline of the study, as well as the sources of data collected during the study.

Chapter 5 Analysis of Study Results

This chapter analyses the results obtained from the study in order of the various data sources used, both in terms of pre/post measures as well as in terms of process measures. It then discusses these results within the context of the research questions given in Section 1.2 (p.3).

Chapter 6 Conclusions and Further Work

This final chapter provides a summary of the work presented in this thesis and the research questions, along with the conclusions. The key contributions of the thesis are also discussed, as well as the limitations and lessons learnt from the research undertaken. Finally, potential further work is outlined, drawing upon the experiences and outcomes from this thesis.

2 Motivation, Metacognition & Feedback

This chapter discusses contemporary theories of motivation in relation to learning before discussing the role of motivation and metacognition on learning. The roles of feedback in learning are then covered, followed by the commonly used measures for measuring motivational state and the use of metacognitive skills. The problems that learners face when learning problem solving within a programming domain are also discussed. Finally, several ITSs are compared, paying particular attention to motivation and metacognition.

2.1 Motivation

Learner-centred research recognises the effect of emotions on learning, with Emotional Intelligence (EI) a key factor in the cognitive process (Benchetrit and Frasson, 2004). For example, emotions such as frustration and anxiety can cause cognitive confusion (Reeve, 2005). Negative emotions are not necessarily bad, nor are positive emotions always good in the learning process (Kort, Reilly and Picard, 2001). For example, the "affective state of confusion… is believed to play an important role in learning… and has a significant positive correlation with learning gains" (Graesser, D'Mello, Craig, Witherspoon, Sullins, McDaniel and Gholson, 2008). Confusion signifies a state of cognitive disequilibrium, which is addressed by undertaking cognitive and metacognitive activities such as thought through to reflection and problem solving (Graesser et al., 2008). Linnenbrink (2007) found in their previous research that there appeared to be more evidence to support that "unpleasant affect could undermine learning" (e.g. anxiety, boredom), with their "findings regarding pleasant affect were inconsistent" (e.g. hope, enjoyment). The optimum learning experience is the "enjoyment of a learning activity for its own sake" (Boekaerts, 2007) as it increases the learner's self-efficacy judgements (refer to Section 2.1.3 (p.10) for a discussion of socio-cognitive theory where self-efficacy is a key component). However, not all learning experiences are like this and in such situations Boekaerts (2007) suggests that "a good ratio of positive to negative emotions is beneficial for learning".

The learning process is a journey that will involve the learner experiencing a wide spectrum of emotions as they progress. Studies involving the use of three ITSs by Baker, D'Mello, Rodrigo and Graesser (2010) showed that confusion and engaged concentration, followed by boredom were the emotions that were the most persistent across the use of the ITSs. Their studies also found that emotions of delight and surprise were rare. Refer to Section 2.6 (p.34) for a discussion of ITSs. Such emotions "influence how efficiently, effectively, and enjoyably" a learner succeeds in this process (Kort et al., 2001). For example, Baker et al. (2010) found that the emotion of boredom has a negative affect during learning as it "was associated with poorer learning and problem behaviours, such as gaming the system". Pekrun, Goetz, Frenzel, Barchfeld and Perry (2011) discusses the view of emotions as "sets of interrelated psychological processes" involving "affective, cognitive, motivational, and physiological components". For example, the emotion of anxiety can comprise feelings of affect (e.g. uneasy and tense), cognition (e.g. worries), motivation (e.g. impulses to leave the situation) and physiological (e.g. peripheral activation) (Pekrun et al., 2011). The importance of emotion on the learning process is further strengthened by the research into neuroscience and education, with research showing that when a brain is at ease it improves both learning and brain development (Immordino-Yang and Damasio, 2007; Hinton, Miyamoto and Della-Chiesa, 2008; J Davidson, Dunne, Eccles, Engle, Greenberg, Jennings, Jha, Jinpa, Lantieri and Meyer, 2012). Immordino-Yang (2011) states that "emotion and cognition are intertwined, and involve interplay between the body and mind" with emotions (e.g. happiness, fear, anger) being "cognitive and physiological processes".

Motivation is an aspect of emotion that has a significant impact on learning, with research showing that motivated learners enjoy higher levels of success, display better self-regulatory control and will be more motivated to learn in the future (Hodges, 2004; Schunk et al., 2007). Motivation can be defined as "the process whereby goal-directed activity is instigated and sustained", with motivational learning defined as "motivation to acquire skills and strategies rather than to perform tasks" (Schunk et al., 2007).

The study of motivation in learning and educational psychology are established fields of research that have undergone a paradigm shift from drive (e.g. Hull's incentive motivation) and behaviourism (e.g. Thorndike's connectionism, Pavlov's classical conditioning) to cognitive psychology during the latter half of the twentieth century (Reynolds and Miller, 2003; Schunk et al., 2007). This shift moved away from viewing academic cognition as a 'cold' domain, purely focussed on knowledge and strategies, to a domain that recognises that motivation and emotions are entwined elements within academic cognition (Pintrich, 2003). The relationship between motivation and academic cognition is reciprocal; motivation can facilitate or constrain learning, as learning can facilitate or constrain motivation.

Rather than review the history of motivation theories, the remainder of this 'Motivation' section will concentrate on discussing the relevance of the contemporary theories of motivation to building educational systems (Schunk et al., 2007): expectancy-value, attribution, socio-cognitive, goal-orientation and self-determination theories. To that end Table 2.1 (p.15) at the end of this section provides an overview.

2.1.1 Expectancy-Value Theory

In expectancy-value theory, as the name suggests, the two constructs that are central to this theory are expectancies and values. In motivation, a construct "is some postulated attribute of people" (Cronbach and Meehl, 1955), referring to a complex psychological concept.

Expectancies are the learner's beliefs about their ability to undertake a task successfully. Wigfield and Eccles (2000) distinguish conceptually between expectancy and ability beliefs (although noting that they are empirically related); the latter are the learner's beliefs about the competence for a current task, whereas expectancy beliefs are the learner's beliefs about future tasks.

Values are the learner's reasons for engaging in a task. Values contain the following four components (Eccles, Adler, Futterman, Goff, Kaczala, Meece and Midgley, 1983):

- Attainment value – the importance of doing well on a task

- Intrinsic value – the personal enjoyment the learner gets from undertaking the task
- Utility value – how doing the task fits into the learner's future plans
- Cost – how undertaking a task limits the undertaking of any other tasks, how much effort is involved, as well as the emotional cost

Expectancies answer the "can I do this task?" question, while values answer the "do I want to do this task and why?" question (Schunk et al., 2007). The answers to these questions greatly impact on the learner's engagement and the outcome of the task involved. The expectancy and value beliefs that the learner has for a task will influence their choice of task. For example, if someone expects to fail in a task or they do not value it, the more likely they are to either avoid the task or not engage fully with the task. Although this is a core idea that applies across many motivation theories, it is explicitly represented in expectancy-value theory (Schunk et al., 2007).

Within the framework of expectancy-value theory, a tutor (human or computer) could promote the learner's expectancy belief of their own ability, emphasising that it is a changeable and controllable part of the learning process. The tutor could also communicate the value of the task, for example, explicitly stating how the task fits into the overall teaching scheme of the topic being taught.

2.1.2 Attribution Theory

In contrast to expectancy-value theory that focuses on the learner's expectancy beliefs at being able to undertake a future task, the key concept in attribution theory is that learners retrospectively attribute causes for task outcomes (Weiner, 1985). Kelley (1973) defines attribution theory as "a theory about how people make causal explanations, about how they answer questions beginning with 'why?'". The theory is based on two assumptions; we strive to understand and master our environment, as well as try to understand the causes of our own and others behaviour.

Attributions are categorised along three dimensions; stability (how stable an attribution is over time), locus (whether the attribution is internal or external to the learner) and control (whether the learner has control over the cause). Such attributions need not be the real causes for an event to have psychological (affect, self-efficacy and expectancy for success) and behavioural (choice, persistence, level of effort and achievement) consequences (Schunk et al., 2007).

Within attribution theory, attributions that are perceived as internal, controllable and stable increase a learner's effort and expectancies for success. An example of an action that a tutor (human or computer) could take is to attribute lack of success in a task to internal, but controllable factors (e.g. attribute the outcome to a lack of effort rather than a lack of ability). Internal attributions of failure can result in lowering self-esteem, but introducing a controllable attribution (e.g. lack of effort or that the task itself was too hard) rather than the lack of the learner's ability (uncontrollable) can help to limit the negative effect on the learner's self-esteem.

2.1.3 Socio-Cognitive Theory

Central to Bandura's socio-cognitive theory (1986) is the acceptance that social factors influence behaviour. The theory is based on the framework of triadic reciprocity (Bandura, 1986), which models the interactions between personal, environmental and behavioural factors. For example, self-efficacy (a personal factor) can influence a learner's choice of or persistence on a task (behaviour factors). Personal factors can also influence each other, for example, self-regulation can promote skill acquisition and lead to higher self-efficacy in the learner (Zimmerman, 2000).

Self-efficacy is a key component of socio-cognitive theory. Self-efficacy is a learner's beliefs about their capabilities to undertake given tasks to attain goals (Bandura, 1977). *Self-efficacy* is different from *confidence*; "the construct of self-efficacy differs from the colloquial term 'confidence.' Confidence is a nondescript term that refers to strength of belief but does not necessarily specify what the certainty is about" Bandura (1997). Personal efficacy beliefs are influenced by four main sources of information: performance accomplishments, vicarious experience, verbal persuasion, and physiological states (Bandura, 1977).

- Performance accomplishments relate to the learner’s mastery experiences (e.g. success increases self-efficacy and repeated failure decreases self-efficacy. If repeated success is experienced, thereby developing strong self-efficacy, then this can greatly reduce the negative impact of occasional failures). This is the most influential source of self-efficacy beliefs (Pajares, 1997), as individuals gauge and interpret the effects of their actions.
- Vicarious experience refers to learning by observing others. Learning, not only from our own experiences, but also the experiences of others greatly enhances the cognitive process and social development, which would otherwise be “greatly retarded, not to mention exceedingly tedious and hazardous” (Bandura, 1989).
- Verbal persuasion is the ability to influence a learner’s self-efficacy by another person (or even a computer tutor) suggesting that they can do a particular task. Self-efficacy beliefs based on verbal persuasion are likely to be weaker than those beliefs based on a learner’s own experience and accomplishments.
- Physiological states relates to emotional arousal. For example, stressful situations tend to lead to high emotional arousal and if such arousals are negative (e.g. anxiety) then this will have a negative effect on self-efficacy.

Self-efficacy influences the choice of activities, effort, and persistence (Bandura, 1986). Schunk (1991) argues that the influence of these motivational outcomes are limited within the learning process that takes place in schools, purely as the learners are in a closed academic environment where they have little or no choice of which activities they can undertake and are kept working on tasks because of teacher actions and the classroom environment. Self-efficacy can also be a double-edged sword in terms of effort and persistence, e.g. a learner with high self-efficacy may not feel the need to expend much effort (Bandura, 1986).

In the framework of socio-cognitive theory a tutor should provide tasks that are challenging, but attainable (e.g. the learner’s self-efficacy fractionally exceeds their actual skills). They could also encourage practice of a task to minimise task avoidance, increase sense of mastery and help develop self-efficacy. Apart from providing examples which tend to show the final solution, worked examples could be provided (e.g. a video that captures the computer screen while a tutor works through an example from initial problem definition to solution). Such examples help to demonstrate the cognitive and metacognitive processes involved, as well as showing that even an 'expert' will not derive the final solution straightaway (e.g. in programming, the initial code that is written may undergo a series of alterations before the final code is formed). A tutor could include learning strategies (e.g. self-regulation and reflection) as a specific skill acquisition and provide feedback based on their progress.

2.1.4 Goal-Orientation Theory

Goals are a construct that is deeply rooted in motivational psychology as they help explain the impetus for people to act (Schunk et al., 2007). In relation to achievement behaviour, goal-orientation theories have been developed to explain the reasons for engaging in achievement motivation and behaviour. The term "goal-orientation" refers to the idea that achievement goals "represent a general orientation to the task that includes a number of related beliefs about purposes, competence, success, ability, effort, errors, and standards" (Pintrich, 2000a).

Two orientation constructs that are presented in goal-orientation theories are mastery and performance (Ames, 1992). Different labels have been used to describe these constructs: "learning", "task" and "task-involved" for 'mastery goals' and "relative ability" and "ego-involved" for 'performance goals'. Mastery goals orient the learner on mastering (learning) how to do a task, while performance goals orient the learner on their ability and performance in relation to others (Pintrich, 2000a). For example, the statement "I like to learn new things" shows a mastery orientation, with "I want to do better than others in the class" shows a performance orientation.

Goal orientations affect motivational, cognitive and behavioural outcomes (Schunk et al., 2007). Mastery goal orientation is linked to positive and adaptive attributions, with learners more likely to see a link between effort and outcome, with effort positively linked with ability. Performance goal orientation is linked to negative and maladaptive attributions, with learners attributing success or failure to ability. Traditionally mastery and performance oriented goals have been viewed as a dichotomy, although correlated studies have found that they may be positively or negatively correlated or even uncorrelated (Pintrich, 2000c). The research by Elliot and Harackiewicz (1996) extends the performance goal orientation by partitioning into approach and avoidance goals. For example, a learner with an approach performance goal orientation would undertake a task with the aim of being the best at the task in comparison to others. Whereas a learner with an avoidance performance goal orientation would aim to avoid looking stupid in comparison to others.

Within the framework of goal-orientation theory, a tutor (human or computer) could aim to minimise the opportunity for social comparison, basing feedback on progress rather than performance. They could also vary and communicate the relevance of the task, as well as providing the learner with a degree of choice (control) in their learning at a task level while still working towards the learning outcomes of the specific domain of study.

2.1.5 Self-Determination Theory

Just as goal-orientation theories focus on the orientation of motivation in terms of mastery and performance goals, Self-Determination Theory (SDT, Deci and Ryan, 1985), which is supported by two sub-theories (Cognitive Evaluation Theory and Organismic Integration Theory), focuses on the orientation of motivation in terms of intrinsic and extrinsic motivation. Intrinsic motivation refers to "doing something because it is inherently interesting or enjoyable" and extrinsic motivation refers to "doing something because it leads to a separable outcome" (Deci and Ryan, 1985). There is no automatic relation between the two orientations (Lepper, Corpus and Iyengar, 2005), as in any given task a person may experience a mixture of intrinsic and extrinsic motivation (e.g. high in both, low in both, high in one and medium in another, and so on). Intrinsic and extrinsic motivational orientations are time and context dependent, and are viewed as separate continuums (Schunk et al., 2007).

Central to the orientation of motivation in SDT is the concept of self-determination, which is the "process of utilizing one's will" (Deci, 1980), with will relating to the capacity for a person to choose how to satisfy their needs. Self-determination relates to a sense of autonomy in a person; their need to feel a sense of control in themselves and their interactions with their environment. Ryan and Deci (2000) identify three innate psychological needs that underline behaviour and effect motivation; competence (the need for understanding and mastery), autonomy (sense of control) and relatedness (the need to belong to a group).

In terms of achievement behaviour, intrinsic motivation refers to activities that are autotelic ("the purpose of the activity is the activity itself") and have positive effects on autonomy, creativity, flexibility, and spontaneity (Deci and Ryan, 1985). Whereas extrinsic motivation relates to external influences and are negatively characterized with feelings of pressure, tension and anxiety leading to low self-esteem (Deci and Ryan, 1985). Intrinsic motivation can be facilitated by providing the learner optimal challenges, the amount of choice allowed and providing positive feedback relating to competence and self-efficacy (Deci and Ryan, 1987; Ryan and Deci, 2000). The use of rewards tied to skill acquisition help to inform learners of their progress in developing skills, which in turn can raise their self-efficacy (Cameron and Pierce, 1994). Rewards given purely for undertaking an activity, threats, deadlines, evaluation and surveillance are examples of features that can curtail self-determination and intrinsic motivation (Deci and Ryan, 1987).

2.1.6 Motivation Summary

The motivation theories in this section deal with achievement motivation research and, as just mentioned, they all view and give differing levels of importance to the constructs involved. This has an implication for teaching with differing and overlapping educational implications. Table 2.1 summarises the motivation theories outlined in this section by highlighting the educational implications in relation to each theory. The table also provides examples of the type of strategies a tutor (human or computer-based) may undertake to promote and maintain motivation in learners. Such strategies do not exclusively belong to one particular motivation theory, with overlapping across theories occurring.

Table 2.1 Educational implications and types of strategies relating to motivation theories

Theory	Educational Implications	Types of Tutor Strategies
Expectancy-Value Theory	Self-perceptions of ability and expectancies for success influence a learners choice of task, cognitive engagement and persistence on a task.	- Promote ability as a controllable changeable part of learning - Communicate the value of a task (i.e. cover the components of task value; attainment value, intrinsic value, utility value & cost belief)
Attribution Theory	Attributions that are perceived as internal, controllable & stable increase effort & expectancies of success	- Attribute failure to controllable & internal factors (i.e. lack of effort) rather than lack of ability
Socio-Cognitive Theory (Self-Efficacy)	Domain and task specific. Challenging, but attainable tasks (self-efficacy fractionally exceeds actual skills). Low efficacy lowers cognitive engagement, persistence and may lead to task avoidance	- Include learning strategies in skill acquisition - Encourage practice of a task type to minimize task avoidance - Provide progress feedback, but not excessive feedback - Tailor & limit help to reduce perceived learner/tutor dependence
Goal Theory	Mastery: focus is on learning & understanding (deep learning) Performance: focussed towards superiority in comparison to others (surface learning).	- Vary tasks - Minimize opportunity for social comparison - Focus feedback on progress rather than performance - Communicate the relevance of the task - Give the learner a degree of choice (control) in their learning
Self-Determination Theory	Intrinsic: promotes cognitive engagement & learning. Extrinsic: more focused on rewards or punishment avoidance.	- Positive feedback on competence & self-efficacy - Degree of choice allowed - If rewards used, only link to the development of the learner's competencies & skill acquisition - Provide tasks that challenge the learner, raise their curiosity, give them control and allow for fantasy

While the contemporary motivation theories, as outlined in this section, assume that motivation involves, and influences, cognition and that this relationship is reciprocal, the labels given to the numerous constructs used and the nature of the underlying processes involved are where the theories provide differing views (Schunk et al., 2007). Although each theory treats constructs differently, all motivation theories are concerned with four key outcomes: choice, cognitive engagement (effort), persistence and achievement/performance (Pintrich, 2003; Schunk et al., 2007; Wigfield and Eccles, 2000). In the discussion of the results from this thesis study in relation to the research questions presented in Chapter 1 (Section 1.2, p.3), Section 5.2 (p.157) discusses three of the four outcomes: effort, persistence and performance.

One construct that is a key mediator across all types of achievement behaviour is *self-efficacy* (Schunk et al., 2007). Research has shown that self-efficacy beliefs in educational research are correlated with other motivation constructs, such as attributions, goal setting, problem solving, self-regulation, social comparisons, strategy training and expectancy across domains (Pajares, 1997). Research has shown that self-efficacy is a motivational construct that has strong links to performance and achievement (e.g. Jiang, Song, Lee and Bong, 2014; Wilson and Narayan, 2014), along with learning strategies with learners high in self-efficacy more likely to use "various cognitive and self-regulatory learning strategies" (Schunk et al., 2007).

As outlined in Chapter 1 (p.1) the focus of this thesis is to provide motivational and metacognitive feedback in a tutoring system which is based on the past motivational states and the previous problems solved by the learner in order to improve the learner's focus on and experience of the learning process. This study has used self-efficacy as the motivational construct to measure during the learning process as it is key in educational motivation. Refer to Section 3.2 (p.65) for the modifications made to SQL-Tutor for this study, as well as Chapter 5 (p.105) for the analysis and discussion of the study results. Section 2.6.3 (p.47) discusses some examples of ITSs that are motivationally aware.

2.2 Motivation, Metacognition and Problem Solving

Just as motivation is a key factor in learning, metacognition is an example of another key factor influencing learning. Metacognition is an area of psychology research that has been defined as "knowledge concerning one's own cognitive processes and products or anything related to them" (Flavell, 1976). Metacognition allows an individual to gain an insight into their own thinking and also promotes independent learning (Paris and Winograd, 1990). Paris and Winograd (1990) acknowledge that most researchers in this area emphasise two aspects of metacognition: knowledge about cognition (states and processes) and control over cognition (planning, evaluation and regulation). These aspects highlight two features of metacognition: cognitive self-appraisal and self-management.

The influence of motivation and metacognition on learning are not isolated, with strongly entwined relationships existing between them. Learners high in efficacy are generally more likely to use "various cognitive and self-regulatory learning strategies" (Schunk et al., 2007). Likewise metacognitive skills are required for motivation, e.g. mastery in goal theory requires insight into one's own knowledge and experience, and the use of adaptive learning strategies (Pintrich, 2000b). Motivation, metacognition and affect are all key components of self-regulated learning which are intertwined (Azevedo and Witherspoon, 2008; Efklides, 2011; Vollmeyer and Rheinberg, 1999; Vollmeyer and Rheinberg, 2013); "metacognition does not embrace just metacognitive experience, but overlaps with the concept of motivation" (Vollmeyer and Rheinberg, 1999). Azevedo and Witherspoon (2008) discuss this relationship in the context of using computer-based learning environments (CBLE) to assist in the learning of conceptually-rich domains, especially recognising that using CBLEs to learn complex, conceptually-rich domains is particularly difficult because using the CBLE requires the learner to regulate their own learning. While the term CBLEs can cover a wide spectrum of computer technologies, early CBLEs focussed on modelling the domain knowledge, whereas intelligent tutoring systems aim to not only model the domain, but also model and adapt to the learner as an individual, and in doing so they aim to assist the learner in self-regulated learning (refer to Section 2.6.1 (p.35) for further discussion of ITSs).

Problem solving is an example of goal directed behaviour that involves metacognitive processes to guide the learner's thinking towards the resolution of the problem (McCormick, 2003) and improve the efficiency of goal directed behaviour (Davidson, Deuser and Sternberg, 1996). This is achieved by helping the learner to recognise the problem, determine what the problem is and understand how to reach a solution (Davidson et al., 1996). Metacognition can also be used to predict achievement of complex tasks. For example, the study by van der Stel and Veenman (2010) found that "metacognitive skilfulness contributed to learning performance" and that "metacognitive skills predominantly appear to be general" across domains with domain-specific metacognitive skills playing a minor role in learning.

Research on the metacognitive processes of problem solving shows the same basic progression from problem identification through to problem resolution and evaluation, although the stages and stage labels used may differ slightly. For example, Pólya (1945), as referenced by Farnham-Diggory (1972), outlines four stages; understand the problem, devise a plan (determine a connection between the known and unknown), carry out the plan and then look back on the solution (determine if the solution can be used in another problem). In keeping with this, Blakey and Spence (1990) outline the basic metacognitive strategies as "connecting new information to former knowledge, selecting thinking strategies deliberately, and planning, monitoring and evaluating thinking processes".

Reflecting and drawing upon prior experience and knowledge are important in the construction of knowledge in terms of drawing upon and further developing mental representations and cognitive relationships in memory (Mayer, 2003). Learning from past experience involves metacognitive processes as an act of "reflection on experience" (Boreham, 1987) where the learner must learn how to learn (Weinberg, 1971). However, Robertson (2001) acknowledges that "we seem to be very bad at recognizing that a problem we have just done can actually help us solve the problem we are currently attempting". This tends to be the case for novice learners and less successful problem solvers who fail to reflect on their approach and therefore do not look and learn from their actions (Gage and Berliner, 1979).

2.3 The Role of Feedback

The learning process is a journey which not only involves the learner experiencing a spectrum of emotions as they progress (Section 2.1, p.6), but it also subjects the learner to numerous internal and external motivational stimuli which can change throughout the learning process. Feedback is an example of external motivation stimuli that is integral to learning as it allows a learner to not only gauge their level of understanding, but to also help facilitate their understanding (e.g. corrective feedback, hints), thereby alleviating confusion. Feedback provides the learner with an opportunity for reflection and growth (Heywood, 2000). Research by Narciss, Sosnovsky and Andres (2014) showed that the effect of feedback that was adaptive to the student using the ActiveMath ITS was dependent on the motivation of the individual students. Their research found that adaptive feedback benefited the performance of the students who were low in motivation, whereas there were no significant effects for highly motivated students. The results from the early research of Dennis, Masthoff, Pain and Mellish (2011) showed that there was reason to consider adapting feedback for learners low in self-efficacy. Although the study involved a very small number of participants (18) it is consistent with the view that "positive persuasory feedback enhances self-efficacy" (Schunk, 1991).

The use of domain, motivational and metacognitive levels of feedback are intertwined and have a reciprocal relationship, as the initial content intent of the feedback given can be different to its impact. This can be explained by use of an example. Using a type of action relating to motivation theories in Table 2.1, p.15 (the 'tailor help' action from the Socio-Cognitive theory) and some feedback examples, Table 2.2 outlines the intent of the initial content and the potential impact of the feedback.

Table 2.2 Initial content intent and potential impact of feedback

Feedback Given	Initial Content Intent	Potential Impact
"To select a list of only the city names from the 'publisher' table use the DISTINCT keyword"	Domain	Domain Motivational Metacognitive
"Do you remember how you felt when you solved task x? You achieved that without requiring a lot of help. Do you think you can do the same for this task?"	Motivational	Motivational Metacognitive
"You have already solved a similar problem in this database. Can you remember how you solved it and can the same be applied here?"	Metacognitive	Metacognitive Motivational Domain

As shown in the examples in Table 2.2, the actual impact of the feedback given can be further reaching than intended from the initial content of the feedback. Domain level feedback, apart from providing domain knowledge, could also impact the learner's motivational and metacognitive skills. For example, the domain feedback given could increase or decrease a learner's motivation depending on whether it led to understanding or confusion about the focus of the feedback. Motivational level feedback could also impact on the learner's metacognitive skills, especially considering that the feedback is based on past experiences and knowledge of the learner, which is the focus of this thesis as stated at the beginning of Section 2.1 (p.6). By providing feedback based on a past motivational state of the learner it could have a direct impact on the learner's metacognition, as in the example used, the past motivational state is related to a past task experienced by the learner. Metacognitive level feedback could also impact on the learner at both the motivational and domain level. Providing metacognitive feedback that refers the learner to the actual steps or processes that they encountered to solve a previous task can increase their motivational state and domain knowledge. It provides the learner with the opportunity to reflect on their past experience and draw upon their knowledge to work out how such previous steps can be applied to the current task, which also has the potential of increasing the learner's domain knowledge. Similarly, such metacognitive feedback has the potential to decrease the learner's motivation if they cannot see how the steps undertaken in a previous task can be applied to the task at hand. Such issues relating to the content of feedback were considered when developing the motivational and metacognitive feedback phases to be used in the Study version of SQL-Tutor (refer to Section 3.3.2, p.78, especially the discussion on p.83).

The quantity, type and timing of feedback deemed to be the most beneficial to the learner are issues that are subject to wide ranging research. Providing little feedback may lead to heightened stress levels in the learner (Heywood, 2000), whereas providing excessive feedback may have a negative effect when trying to promote a learner's sense of efficacy (Schunk et al., 2007). These issues are discussed further in the following paragraphs.

Types of feedback have differing effects on self-efficacy, indicating that feedback is context dependent. For example, providing performance feedback when a learner can obtain such information themselves will have little effect, but providing performance feedback when the learner cannot easily ascertain this information for themselves will have a positive effect on self-efficacy. Similarly if a learner attributes receiving help (e.g. via tutor feedback) as a sign that they cannot complete a task, then motivation can become weakened (Weiner, 2005) because it raises negative emotions within the learner. Feedback must also be credible, for example, telling a learner that they are good at a task when they are struggling to progress will not raise self-efficacy (Schunk et al., 2007). The studies conducted by Narciss (2004), which provided informative tutor feedback to students undertaking concept identification tasks, showed that motivation and achievement of a learner were dependent on the type of feedback received and their own level of self-efficacy. The studies found that students who were low in self-efficacy cancelled more tasks, solved fewer tasks and were less satisfied with their performance. Barrow, Mitrovic, Ohlsson and Grimley (2008) conducted a study using two versions of SQL-Tutor that provided different types of feedback; the control group were given negative feedback, while another group of students were given positive feedback. The study found that the students receiving positive feedback needed less time and fewer attempts to solve the same number of problems as the control group. The research into the content of feedback covers a broad spectrum of considerations, e.g. positive and negative comments just mentioned, content (qualitative information) and progress (quantitative information) categories (Jackson and Graesser, 2007), and politeness (Johnson and Rizzo, 2004; Porayska-Pomsta and Mellish, 2013).

Renkl and Atkinson (2003) state that cognitive load research shows that feedback gained via worked-out examples are more beneficial during the initial stages of study, with problem-solving proving to be superior in later stages. Whereas Kirschner, Sweller and Clark (2006) argue that minimum guidance, including the problem-based learning approach, does not work and that learning should involve direct guidance. They do recognise, however, the use of worked examples as a form of direct guidance.

In terms of the timing of feedback, especially when errors arise, there are two main schools of thought: (1) providing immediate feedback in order to prevent the learner from becoming discouraged and stuck on a task (2) delaying feedback in order to provide the learner with an opportunity to acquire and engage in self-regulated learning strategies (Mathan and Koedinger, 2005; Person and Graesser, 2003). Mathan and Koedinger (2005) argue that the timing of feedback given by any underlying cognitive tutor should be based on the "model of desired performance" using feedback that is relevant to the performance that is expected from the learner. This is a view that is held by Anderson and colleagues (Anderson, Corbett, Koedinger and Pelletier, 1995; Corbett and Anderson, 2001) who detail experiment results where the timing of feedback was context dependent. For example, in coding exercises learners benefited the most from immediate feedback, whereas in debugging exercises it would be beneficial to delay feedback given the nature of the exercise. In Ben-Ari's (2001) brief comparison of constructivism and minimalism, one of the common points mentioned is that errors should be employed as a pedagogical device, rather than being viewed as a symptom of failure. Weinberg (1971) notes that all too often programmers are pressed to eliminate a bug too quickly, which may not be the best strategy. From analysing human-to-human tutoring dialogs Person and Graesser (2003) found that regardless of which timing strategy may be optimal in a learning experience, human tutors favoured responding to errors with immediate feedback.

The points discussed here with regard to the type, timing and quantity of feedback are all still applicable to feedback received from interactions with computer systems. Ben-Ari (2001) recognises that receiving such computer-based feedback may be discouraging to some learners, especially if they "prefer a more reflective or social style of learning". Lepper, Woolverton, Mumme and Gurtner (1993, p.102) identified the *plausibility problem* as "whether the same actions and the same statements that human tutors use will have the same effect if delivered instead by a computer, even a computer with a virtually human voice". This problem of what learners will accept in their interactions with tutoring systems (e.g. feedback comments, hints, whether written or spoken, and with or without the use of embodied agents) was discussed further by du Boulay, Luckin and del Soldato (1999). In du Boulay and Luckin (2001) three potential kinds of plausibility problem were discussed; whether the learner perceived the system to be acting on its own, on behalf of a human or it is unclear between the first two perceptions.

2.4 Gauging Motivational State and Using Metacognitive Skills

Motivational states can be gauged in various ways, ranging from direct observations and ratings by others to self-reports which include questionnaires, interviews, stimulated recalls, think-alouds and dialogues (Schunk et al., 2007; Porayska-Pomsta, Mavrikis, D'Mello, Conati and Baker, 2012; Porayska-Pomsta, Mavrikis, D'Mello, Conati and Baker, 2013). Such methods have also been used within Intelligent Tutoring Systems (ITS). For example, the Wayang-West mathematics ITS was extended to gauge learner motivation via self-reports (Beal and Lee, 2005). Self-reports were also used in an early version of MOODS, MOtivation Diagnosis Study (de Vicente and Pain, 1999), and used post-hoc teacher/third party reports and think-alouds via video observation in later versions to infer the motivational state of the learner in the video (de Vicente and Pain, 2002; de Vicente and Pain, 2003). D'Mello, Taylor, Davidson and Graesser (2008a) also used teacher post-hoc observations, but these observations were compared to the post-hoc reports of the learner. Their study found that the emotional states inferred post hoc by the teachers were not reliable and did not match the learners' post-hoc reports. Arroyo, Cooper, Burleson, Woolf, Muldner and Christopherson (2009) used both physiological sensors and self-reports in their study to measure learner emotions as they used the Wayang Outpost geometry ITS. The self-reports were used to compare to the data from the sensors, with the learner being prompted to self-report every five minutes and after they had completed a problem. Such a relatively high frequency of prompts for self-report may have had the potential to interrupt the flow of learning/using the tutor, but this is not mentioned by the authors of that study. Diagnosis of a learner's motivational state that does not interrupt the learning process is one of the requirements for motivational state detection outlined by McQuiggan, Mott and Lester (2008), the other requirements are discussed on p.25. The Wayang Output study mentioned provides an example of the limitation of using specialised equipment such as the sensors used, as the number of participants is limited to the availability of the sensors. The authors of that study did not have enough sensors for all of the thirty-eight participants and instead they had to use different sensors at different sessions and had to combine data to formulate full data sets (Arroyo et al., 2009, p.22). More recently the use of self-reports have been used in studies such as (but not limited to) the use of an 'AffectButton' by Broekens and Brinkman (2013) and the use of the 'Affect Grid' by Fulmer, D'Mello, Strain and Graesser (2014).

Stimulated recalls are a useful method for gauging motivation as they provide the individual with an opportunity to link "thoughts to specific behaviours" (Schunk et al., 2007). However, not only are they reliant on the individual's memory, but there is also the potential for a shift in judgement between the actual event and the post event recall, especially dependent on the elapsed time between the two. For example, if a learner feels extremely confused and is suffering from low self-efficacy while undertaking a task, after the event, the recall of these feelings may change, they may recall it as a less negative event in relation to all of the other events that may have occurred since.

In terms of observations by others, the observers may have a more objective view compared to the people being observed. When these observations are post hoc then there is the potential for the observations to be taken out of context or for the observers not to be fully aware of the context being observed. The studies by Rodrigo, Rebolledo-Mendez, Baker, du Boulay, Sugay, Lim, Espejo-Lahoz and Luckin (2008) is an example where real-time observations were used with the observers standing diagonally behind the individuals being observed.

One of the most direct methods of gauging motivation is the use of self-reports, especially in a real-time situation, in order to capture the person's thoughts in situ; this is referred to as *ground truth* by Beal and Lee (2005). This is important for attempting to capture the current motivational state of a person in a learning context, as the affective states experienced are fluid and continually change, e.g. confusion, boredom, curiosity (Kort et al., 2001; Graesser et al., 2008). This is also acknowledged by Weinberg (1971), "once the problem solution has been shown, it is easy to forget the puzzlement that existed before it was solved". This is further strengthened in the studies by Arroyo et al. (2009) which showed "that students' self-report of emotion depends on events that occurred in the previous problem". Self-reports also mean that the person's thoughts are being captured, rather than the opinions of an external observer. However, there are some problems associated with self-reports. For example, a learner engaging with an ITS may give answers that they believe are socially acceptable (Howard, 1994; Schunk et al., 2007) and they believe will 'please' the ITS (de Vicente and Pain, 2003). Schwarz (1999) discusses the potential influences on responses, such things as wording, formatting and scale can have, along with various techniques to minimise or negate their effect. In addition to these issues of self-report Porayska-Pomsta et al. (2013) also raises the point that "self-reports are limited to situations where the emotional episode is sufficiently pronounced to enter learners' consciousness so that it can be subjectively accessed". Refer to Porayska-Pomsta et al. (2013) for a discussion covering the issues of not only self-reports, but other forms of knowledge elicitation for affective modelling. Schunk et al. (2007) mention some steps that can be taken to minimise such potential problems (e.g. guarantee the confidentiality of the data, use multiple methods to gauge motivation and validate self-reports). Using self-reporting methods provides the learner with an opportunity to reflect on their motivational state and their learning experiences. Self-reports are not only a primary data source in the fields of psychology and social sciences (Schwarz, 1999), but also within learning, e.g. self-reports have been one of the most widely used forms of self-regulated learning (Cleary and Callan, 2014) and are an efficient source of information (Beal and Lee, 2005). Although Bandura (1986, p.396) states such measures have to be "tailored to the domain of psychological functioning being explored", so the self-efficacy reports are directly related to the experience being measured.

In the discussion of the cognitive presentation and measurability of goals, Pintrich (2000a) states that achievement goals are likely to be articulated, providing a better opportunity to assess them, which "can be accomplished with verbal report methods (e.g., self-report surveys, interviews, think-alouds, stimulated recall)". In terms of the issue regarding the accuracy of these methods Pintrich (2000a) points out that such methods would not be "trying to tap deeply seated unconscious motives or finding the deeply buried 'true' self", but rather assessing the cognitive achievement goals which can be accessed by the individual's conscious awareness.

McQuiggan et al. (2008) outline three requirements that should be satisfied for measuring a learner's motivational state (focused on self-efficacy) in an ITS:

- The computational mechanism used to diagnose the motivational state should operate at runtime. This means that the learner's state is captured as it happens and as it changes throughout the learning process.
- The means of diagnosis should be efficient.
- The diagnosis should not interrupt the learning process.

The first requirement specified by McQuiggan et al. (2008) has been discussed on p.24 in relation to capturing the *ground truth*. The other two requirements are highly dependent on each other, as providing an efficient mechanism by which the learner can quickly complete their self-report will also have the benefit of not interrupting the learning process. Refer to Porayska-Pomsta et al. (2013) for further discussion of the timing of self-reports in terms of minimising any interruption they may cause.

In the study conducted by Wilson and Narayan (2014) self-reports were used to determine the participants' task self-efficacy. The study involved the participants (undergraduate computer science students) undertaking three tasks (one each week) using the SEREBRO (Software Engineering REwards for BRainstorming Online) computer educational program, and were prompted for their self-reports via an online questionnaire using a seven-point Likert scale. This study found that self-efficacy was a predictor of performance and vice versa. This relates to the *performance accomplishments* discussed in Section 2.1.3 (p.10).

The experiments conducted by McQuiggan et al. (2008) measure a learner's self-efficacy by measuring physiological responses (heart rate and galvanic skin response). The aim of these experiments was to monitor the learner's self-efficacy in order to build a predictive model and to eventually reduce the need for biofeedback devices. Underlying the biofeedback data were the responses from learner self-reports taken throughout the session via an on-screen slider bar control. Self-reports were still used to give meaning and verification of the biofeedback data. In measuring mood, Khan, Brinkman and Hierons (2008) monitored the user's interaction with the keyboard and mouse. Burleson and Picard (2004) used "a rich set of sensors including skin conductivity, facial expression analysis (eyebrow raise, head nod and shake, mouth smiles and fidgets, and blink rate), pressure mouse, and a seat posture sensor" to infer a learner's affective state.

While the use of external sensors can help to increase the accuracy of measuring a learner's motivational state, they may also interfere with this process. For example, the learner may become distracted by the external sensors or they may even act differently. There is a potential with the use of external sensors and equipment to interrupt the learning process (the point mentioned above) or curtail self-determination and intrinsic motivation (as discussed at the end of section 2.1.5, p.13). Depending on the nature and speciality of the external sensors, there is the potential that their use could limit the environments in which they are used (e.g. confined to a controlled laboratory environment).

Taking all of the points raised in this section regarding gauging the motivational state of the learners, the study for this thesis was designed to use self-report to gauge the learner's self-efficacy levels. Refer to Section 3.3.2 (p.78) for details on how self-reports were incorporated into the version of ITS used by the Study group, especially to Figure 3.12 (p.82) which shows the self-report user interface that was used in this study.

2.5 The Problem with Programming

Programming is generally regarded as a difficult skill to learn (Robins, Rountree and Rountree, 2003; Lahtinen, Ala-Mutka and Järvinen, 2005). Programming by its very nature requires the learner to understand abstract concepts, as well as utilise a range of cognitive activities during the course of understanding the problem, designing, developing and debugging a program (regardless of whether it is a declarative, procedural or object-oriented program for example). The scope of this section is to provide an overview of these problems as background information. For a broad review of previous research into the psychological and educational aspects of programming, refer to Robins et al. (2003).

Research has highlighted several problems faced by someone learning to program and also recognises that, while programming is often likened to an 'engineering' discipline, it is also viewed as a psychological task (for example, the seminal work of Weinberg (1971) helped to stimulate research in this area). Work by du Boulay (1989) categorises the problems of learning to program into five overlapping areas, as discussed below: general problem orientation, the notional machine, notation, structures and pragmatics.

- *General problem orientation* relates to the learner finding out what programming is for, including what types of problems can be resolved by programming. It also relates to the benefits of learning to program from the learner's viewpoint. Too often students come to further and higher education computer courses with little or no understanding of programming. Earlier stages of education shelter the learner from programming, as they are focussed on teaching them how to use software packages to interact with computers (information technology literacy), but not how to create such software (Clark and Boyle, 2006). In this sense the computer remains a mystical, closed world.

  The benefits of learning to program are not limited to people who want to develop a career as a programmer. Programming involves a number of transferable skills (refer to the Pragmatics point below), which may be beneficial across disciplines. For example, du Boulay (1980) used learning programming as a means to aid trainee teachers develop their mathematics skills. Palumbo (1990) reviews several studies focussing on the transferable skills related to programming (e.g. the relationship between programming instruction and "perceptual language skills in language-impaired preschool children").

- *The notional machine* is the model of the computer in relation to the program and the physical machine on which the program executes. Programming languages provide a layer of abstraction between the programmer and the underlying hardware structures. Learner programmers tend to have difficulty distinguishing between the notional machine and its relationship with the physical machine (Bladek and Deek, 2005).

  Many programming languages are now supported by rich development environments aimed at supporting the programmer, but for learner programmers they can increase cognitive load. For example, Integrated Development Environments (IDE), e.g. IBM's

Eclipse and Microsoft's Visual Studio .NET, can cause difficulties for the learner, as these environments in themselves can be quite complex to learn and can overwhelm the user (Scott, Watkins and McPhee, 2008). Although such environments can be complex, they also provide software visualisation to varying degrees and step-wise program execution. Such tools help to provide a 'glass box approach' (du Boulay, O'Shea and Monk, 1981) by presenting a window onto the execution cycle of the program as the learner can see, and even effect, changes in data structures as the program progresses. Visibility and memory overload in learner programmers was an issue identified in Pane and Myers' (2000) review of studies of beginner programmers.

- *Notation* – in addition to learning the nature of the notional machine, learner programmers also have to learn the notation of the given language being taught. Apart from the need to gain syntax and semantic knowledge, the learner must also gain an understanding of why and when to use one syntactic construct over another (e.g. when to use a 'while' loop in lieu of a 'for' loop, or when to use an array instead of a linked list). The syntax of a programming language can have a negative effect on the efficiency of the problem solving process (Bladek and Deek, 2005).

A "pedagogic IDE" (Reis and Cartwright, 2004) refers to programming environments that have been designed specifically for learning. They provide the learner with a syntactically reduced version of the programming language, as well as a simplified development environment. Gómez-Albarrán (2005) provides a review of some of these educational tools. For example, DrJava (Allen, Cartwright and Stoler, 2002; Reis and Cartwright, 2004) and BlueJ (Kölling, Quig, Patterson and Rosenberg, 2003) both teach Java programming by masking certain aspects of the full Java language. They both exclude the need for the main method to start a program in order to avoid overloading the learner at the beginning with complex concepts (e.g. arrays and access modifiers). The disadvantages of masking elements of a language are that learners are often reluctant to move to the standard IDEs and have difficultly transferring to them, with learners who are successful in using the reduced language/environment experiencing difficulty in applying the concepts to the full IDE (Kölling et al., 2003). To an extent the learners have to 'unlearn' behaviour when moving to a standard IDE (e.g. they have to start to use, and remember to use, the main method as an entry point for the program).

- *Structures* relate to obtaining and applying knowledge schemas or plans in order to resolve parts of the overall programming problem. Experienced programmers have sources of knowledge that include abstract schemas or patterns, which they can recognise and adapt to a problem definition (Robins et al., 2003). For example, an experienced programmer would be able to use a schema for coding a solution to sort elements in an array. The use of such schemas tend to reduce the effort or computation required to produce a programming solution (Robins et al., 2003). In teaching SQL, Al-Shuaily and Renaud (2010) used SQL patterns to aid the learners to understand the common design patterns. The participants on their study reported that the patterns made solving SQL problems easier as they found it difficult to remember all of the required syntax (e.g. to remember beyond the initial SELECT clause in a SELECT query). However, this study can only be treated as a very small pilot-type study as there were only three participants, although it provides the basis for further research.

- *Pragmatics* of developing a program involves various cognitive and metacognitive skills (Pea and Kurland, 1984). For example, planning, developing, testing and debugging are all part of solving a programming problem. Developing a program is a process that can be divided into smaller sub-processes, but the process of programming is often invisible or seen as a "single, monolithic solution" (Caspersen and Kölling, 2006).

While teaching undergraduate University students on various programming modules (e.g. Java, VB.NET, Perl, SQL), the author witnessed the students facing the problems discussed here. The author also reflected on how they and their colleagues would often respond to a student's question with a question in order to prompt to student to reflect on problems they had previously worked on, as the students tended to work on each problem in isolation from the previous ones. The combination of these factors led to the initial idea for this research degree.

Bladek and Deek (2005) discuss learner programmer problems categorised as pedagogical roots (the methodologies used to support teaching), psychological roots, programming language paradigms, programming language intricacies, debugging skills, and external influences. Whereas du Boulay (1989) categories are learner-centric, Bladek and Deek broaden these categories to external influences, including pedagogical roots. Although a different categorisation has been used, Bladek and Deek's categories complement and overlap those presented by du Boulay. For example, Bladek and Deek's 'psychological roots' category includes discussion of the learner's difficulties with the notional machine and conceptualising what is meant by such things as variable assignment.

External factors that influence the learner programmer are related to hardware and software (e.g. complex compiler error messages) dependencies (Bladek and Deek, 2005). As part of the notional machine and its relationship to the physical machine, a learner may also be subjected to hardware constraints relating to portability issues.

Many pedagogic techniques have been employed for teaching programming, for example, procedural and objects first, the use of mini languages. Mini languages are limited to a subset of the programming languages functionality, e.g. BlueJ (Kölling et al., 2003). Robins et al. (2003) provide a short review of studies focussed on procedural versus objects first learning. Such techniques, along with textbooks that teach programming, are often attributed with focussing on the notation of the language and ignore the pragmatics involved (Bladek and Deek, 2005), with the 'program' treated "as a noun rather than as a verb" (Caspersen and Kölling, 2006). This issue is highlighted by Lister (2011) where some of his previous studies have shown that novice programmers can trace code in terms of the variable values and flow, but they have difficulty providing meaningful realisation as to the intent of the code. This is in contrast to experienced programmers that will examine the code in order to determine its intent without the need to execute the code (Lister, 2011). There are exceptions to the issue of focussing novice programmers on the notation over the pragmatics involved; for example, Vickers (2008) approaches learning programming via learning problem-solving skills that are not programming language specific.

In their review of studies of learning to program, Robins et al. (2003) highlight that many studies reach the same conclusion as to the one overriding problem faced when learning to program: the learner's inability to combine various programming elements together to form/develop a program and the "structures" area in du Boulay (1989). For example, Spohrer and Soloway (1989) recognise that learners are not given enough instruction on how to "put the pieces together", which relate to the underlying skills of problem solving, design and expressing the design through to the program solution. This problem with re-conceptualisation (working from an initial problem statement through to the finished solution) is applicable to a range of programming language types (e.g. declarative, procedural, object-oriented). For example, Prior and Lister (2004) acknowledge this same problem is faced by learners while constructing SQL (Structured Query Language) statements. The problems experienced while learning other programming languages are the same when learning SQL, although such problems tend to be compounded in relation to the length of time and effort required to write an SQL query compared to a program. SQL is a declarative language (Kenny and Pahl, 2005) designed for database programming, although there are extensions to SQL that provide procedural functionality such as flow of control (e.g. Transact-SQL in Microsoft SQL Server and PL/SQL in Oracle). The ability to use SQL efficiently to query a database is a pivotal skill that is required by most software developers (Prior and Lister, 2004). SQL is a well-structured language; although the queries can range from quite simple to complex (e.g. queries can include case statements (if-then-else type statements), nested queries, unions). Students learning SQL often find it quite difficult to learn (Mitrovic and Ohlsson, 1999; Myers and Douglas, 2007), experiencing semantic, syntactical and pragmatic errors (as experienced in any programming language). Learners experience a high memory load as they "have to keep in mind database schemas, names for attributes and tables, the semantics of the latter and the corresponding integrities" (Mitrovic and Ohlsson, 1999).

In order for a learner to be able to apply their newly acquired knowledge of the language syntax and semantics they need to develop problem solving strategies and schema knowledge; this directly relates to the "pragmatics" area in du Boulay (1989). Reflecting and drawing upon prior knowledge and experience (as discussed in Section 2.3, p.18) is an essential part of the learning process, but in the case of programming, the concepts involved and the abstractness of the discipline (Návrat, 1994) can make knowledge acquisition difficult for the learner to master without any prior knowledge to encompass (Bladek and Deek, 2005). Linn and Dalbey (1989) explicitly include problem solving as a skill to be taught as part of learning programming in their "chain of cognitive accomplishments" (the chain consists of learning the language features, design skills and problem solving). Robins et al. (2003) reflect that this chain provides a good summary of what "could be meant by deep learning in introductory programming". A deep approach to learning has been shown to correlate positively with marks on introductory programming courses, with surface learning negatively correlated (Simon et al, 2006).

Research such as Bosch, D'Mello and Mills (2013) have conducted studies that focus on the emotions experienced by novice programmers, in particular the novice's first ever encounter with a programming language (Python). Their study had twenty-nine participants who were university students from psychology subjects and who had no prior programming experience. The study required the participants to take part in one forty-minute session where they wrote code to solve problems via a simple computer environment (as opposed to something more complex and adaptive as an ITS) that provided hints, if the participants requested to view them. The session was divided into two phases: 1) a twenty-five minute scaffolding phase where explanation and hints were available, and 2) a fifteen-minute fadeout phase where no explanations or hints were available. The reason why the study was limited to the use of just one such session was because it explicitly focussed on the emotions experienced during the participants' first encounter. As the study wanted to focus on the emotions experienced and any relation to performance, the system used a points scoring system that was displayed on the screen the whole time, and every time a participant selected to view a hint, then a point was deducted. The authors have not mentioned whether they considered if the potential pressure or anxiety over losing points could have an influence on the results in terms of the emotions experienced by the participants. The emotional states were determined using retrospective self-reports, where the participants viewed videos of their faces and the computer screens that were recorded during their session. Each self-report occurred immediately after the session, with the participants prompted to self-report on their emotions approximately every twenty seconds. While such high frequency of self-report prompts would have been intrusive if they had occurred during the actual session, conducting self-reports retrospectively negated such disruption to the learning process. This is consistent with one of the requirements for measuring motivational states specified by McQuiggan et al. (2008), refer to the discussion on p.25 of Section 2.4. However, the authors do not mention if they considered whether the self-report being retrospective, despite immediately after the session, had any influence on the participants' self-report judgements (refer to p.24 of Section 2.4 for further discussion of such issues of self-reports and ground truth). The results of the Bosch et al. (2013) study indicated that "flow/engaged (23%), confusion (22%), frustration (14%), and boredom (12%) were the major emotions students experienced, while curiosity, happiness, anxiety, surprise, anger, disgust, fear, and sadness were comparatively rare". These results are comparable to the results from the study by Baker et al. (2010), as discussed in this document on p.7 of Section 2.1. Frustration was reported more in the fadeout phase, than the scaffolding phase which the author's attributed to the lack of hints available during the fadeout phase. Boredom was negatively correlated with performance during the fadeout phase, but positively correlated during the scaffolding phase, which could also be related to the availability of hints and their benefit to the participants.

2.6 Intelligent Tutoring Systems

The previous sections have discussed metacognition and motivation as two key factors that influence learning across any domain. The last section focussed on problems relating to learning to program, especially learners acquiring the ability to work through a problem and combine programming elements together to develop a complete program. This section begins by discussing what intelligent tutoring systems are and their role in supporting learning. Then it builds on discussions from the previous sections by reviewing a selection of tutoring systems from (meta) cognitive and motivational viewpoints to determine how such systems incorporate a learner's metacognitive skills and motivational state into the learning process.

2.6.1 Intelligent Tutoring Systems: What Are They?

The initial development of computer-based learning environments (CBLE) such as computer aided learning (CAL) environments and e-learning systems focused on modelling the domain knowledge and the cognitive processes of learning, which meant that every learner received the same interaction with the system, with the learner having been viewed as a concept and not as an individual entity. The focus of some aspects of technology-enabled learning research, e.g. artificial intelligence in education (AIED), has moved from an instructional/teacher-centred approach to a social/learner-centred approach, incorporating the use of an adaptive learner model. This has helped to develop tutoring systems that maintain a model of what the system perceives is reflective of the learner's domain knowledge and adapts its interactions with the learner to take this into account. Such systems are known as *intelligent* tutoring systems (ITS), as they use the learner model to tailor the learning experience to each learner individually. Burns and Capps (2013) give a finer-grained definition of an ITS, stating that it must pass "three tests of intelligence" which they describe as follows.

> *"First, the subject matter, or domain, must be "known" to the computer system well enough for this embedded expert to draw inferences or solve problems in the domain. Second, the system must be able to deduce a learner's approximation of that knowledge. Third, the tutorial strategy or pedagogy must be intelligent in that the "instructor in the box" can implement strategies to reduce the difference between expert and student performance."*
>
> (Burns and Capps, 2013, p.1)

Further research using the social/learner-centred approach has seen the development of ITS that not only deal with domain knowledge and cognitive processes, but also with the affective dimension of the learner by attempting to dynamically take in to account the motivational, metacognitive and affective states of the learner (du Boulay et al, 2010). Such systems aim to maintain a model of the learner and dynamically adapt to the learner as an individual, making the interactions with such systems unique to each learner.

Research and development of some ITSs has involved incorporating a *pedagogical agent* into the user interface. Pedagogical agents are on-screen characters that are used in the ITS to guide the learner through the system and they have taken many forms throughout the various ITSs that use them; for example, Herman the bug in Design-a-plant (Lester, Converse, Kahler, Barlow, Stone and Bhogal, 1997), Merlin the wizard in Prime Climb (Conati and Zhao, 2004) and human-like agents used in MIMIC (Multiple Intelligent Mentors Instructing Collaboratively) web-based environment (Baylor and Kim, 2005). Heidig and Clarebout (2011) conducted a review of seventy-five articles relating to thirty-nine studies using pedagogical agents in order to determine if using such agents facilitate learner motivation and learning, which they are expected to do due to the social cues that such animated characters can provide. Firstly their review found that very few studies made use of control groups as part of their methodology (fifteen out of the thirty-nine studies reviewed). Out of the fifteen studies that did use control groups, nine of them showed that using a pedagogical agent made no difference on learning. In terms of motivational measures, only four out of the fifteen studies applied them and out of those three reported no difference to learning. The conclusion drawn from the Heidig and Clarebout (2011) review is that using pedagogical agents generally has no effect on motivation or learning and their use "has to be questioned", although they do acknowledge that the question of whether they are effective is too broad given the variety of agents used and the specific functions each is designed for.

The aim of every ITS (indeed of every source of learning from paper-based classroom exercises to problem solving with an ITS) is to stimulate learning by improving its effectiveness (Mathews and Mitrovic, 2007). In working towards this achievement, the one role model that ITS strive to emulate is the *human tutor*, which is undertaken by observing and modelling aspects of human tutors' teaching strategies and interactions with the learner (Graesser, Person, Harter and TRG, 2001; Johnson, Wu and Nouhi, 2004; Sykes and Franek, 2007). While ITS are not as effective as one-to-one human tuition, this gap is narrowing as research progresses and they generally show an improvement to learning compared when not used (VanLehn, 2011) and even, in some instances, when compared to a human tutor. For example, in the recent meta-analysis of ITS and learning outcomes presented in Ma, Adesope, Nesbit and Liu (2014), they found that the use of an ITS showed an achievement gain compared to using textbooks or workbooks, other use of computer instruction and human tutors in large group instruction. They also found that there was no significant difference when using an ITS compared to one-to-one human tuition and human tutors in small group instruction.

ITS that focus on problem solving domains tend to use one of two learner models: Cognitive Tutors and Constraint-Based Modelling (CBM) (Desmarais and Baker, 2012). Cognitive tutors (Corbett and Anderson, 2001) represent procedural knowledge and deal with the actions of a learner (the individual steps involved in solving a problem), whereas CBM tutors (Mitrovic and Ohlsson, 1999) represent declarative knowledge and deal with constraints over a submitted answer. Cognitive tutors are based on the ACT-R theory, which is a model for higher-level cognitive processes (Anderson et al., 1995). The model assumes that declarative and procedural knowledge exists and uses 'chunks' ("schema-like structures") and production rules to process the learner's actions (Anderson, Matessa and Lebiere, 1997). As the name suggests, CBM is based on the use of constraints, with domain knowledge held in constraints against correct solutions, with each constraint representing "an atomic fact or principle of the domain" (Mathews, 2012) (refer to Section 3.1.1 (p.60) for further details of constraints). Desmarais and Baker (2012) provide an overview comparison between the knowledge tracing rules used in a cognitive tutor and the constraints in CBM. Tutoring systems based on both models have successfully made the transition from controlled research to classroom environments, including commercial distribution. PAT (Practical Algebra Tutor) (Koedinger, Anderson, Hadley and Mark, 1997) and APT (ACT Programming Tutor) (Corbett and Anderson, 2001) are examples of cognitive tutors. SQL-Tutor and EER-Tutor (Mitrovic, 2012) are examples of CBM tutors.

As SQL-Tutor has been used in this thesis, three tutoring systems that also focused on the domain of SQL SELECT queries were found in the review of the literature. The systems are SQL-LTM

(Dollinger, 2010), Acharya (Bhagat, Bhagat, Kavalan and Sasikumar, 2002) and the name of the third system was not mentioned in Kenny and Pahl (2005), although it may simply be called the SQL Tutoring System. These systems are outlined here and have not been included in the next section alongside SQL-Tutor because of either the lack of information available and/or they do not focus on metacognition (the system from Kenny and Pahl does include some adaptive feedback that would aid reflection).

- SQL-LTM (Lightweight Tutoring Module) passes the learner's submitted SQL query to a backend database to determine if it is syntactically correct by being able to execute it in the database. It then uses LINQ (Language-INtegrated Query) to XML capabilities to check semantically an XML (eXtensible Markup Language) representation of the query against a model answer. SQL-LTM provides domain feedback, but further details are unknown. Dollinger (2010, p.3324) provides a comparison between SQL-LTM, Acharya and SQL-Tutor, although this comparison states that both SQL-LTM and Acharya differ from SQL-Tutor in that they execute the submitted queries in a backend database to provide query output to the learner and SQL-Tutor does not. While SQL-Tutor does not execute the query to output its results when it is submitted, it does however have the functionality for the learner to request the output from the query as and when they want it (refer to Section 3.2.2 for this functionality in SQL-Tutor). It also mentioned that neither SQL-Tutor nor Acharya cater for queries that are correct, but are not exactly the same as the model answer. SQL-Tutor can handle such instances as it checks the submitted query against all of the constraints, so its ability in this situation is only limited by the constraints contained in the system (refer to Section 3.1.1 (p.60) for further details of constraints inside SQL-Tutor).
- As already mentioned and like SQL-LTM, the Acharya system executes the query submitted by the learner against a backend database, it also analyses the query clause-by-clause against a model answer and also each element of a clause are compared using truth tables. As with SQL-Tutor and SQL-LTM, Acharya can handle different versions of a correct answer. As in the Dollinger (2010) paper, Bhagat et al. (2002) also incorrectly

state that SQL-Tutor does not contain the functionality to allow the learner to execute their query and view the output. Acharya provides domain feedback on the submitted queries. Similar to the interface in SQL-Tutor, Acharya presents the learner with a text box per possible clause in a SELECT query. As commented by Dollinger (2010), this type of interface is restrictive when it comes to far more complex SELECT queries (e.g. UNION queries).

- The tutoring system from Kenny and Pahl (2005; 2008) uses pattern-based error classification to assess the correctness of the submitted queries. It also provides three different levels of feedback (error flagging, hints and partial solutions), which is comparable to SQL-Tutor's six levels of feedback (refer to Section 3.1.4 (p.64) for further details of the help levels in SQL-Tutor). One element that is different to SQL-Tutor is that Kenny and Pahl's system provides adaptive feedback comments based on the learner's current performance, e.g. "you are having a lot of problems with aggregate functions" (Kenny and Pahl, 2005, p.61).

2.6.2 Metacognition in Tutoring Systems

SQL queries, the domain focus of this thesis, involves metacognitive processes associated with problem solving. The tutoring systems reviewed in this section represent a cross-section of systems relating to the programming domain, with the exception of AutoTutor that is a domain-independent system. The five tutoring systems are AutoTutor, BITS, ELM-ART, JV2M and SQL-Tutor, which were chosen because they are well-established systems, especially in the case of AutoTutor, ELM-ART and SQL-Tutor. The review of metacognitive systems is not exhaustive in that there are a number of other systems that promote the learner's metacognitive skills, but they do not deal with metacognition in programming. For example, MIRA (Metacognitive Instruction using a Reflective Approach) (Gama, 2004; Gama, 2005) is an ITS focused on solving algebra word problems. MIRA categorised students as to whether they were optimistic, pessimistic or realistic with respect to their own estimates of how well they would tackle particular problems. It then compared the predictions to the outcomes and presented this to the student as a means of reflection to promote metacognition and self-assessment. Another example of a system concerned with metacognition is EcoLab II (Luckin and Hammerton, 2002), which is an interactive learning environment used to teach food webs and chains, and which incorporates a metacognitive learner model.

Table 2.3 shows the main points of comparison between the five tutoring systems. The comparison focuses on the content structure, framing, feedback including help and the type of user control used in the interface.

Table 2.3 (Meta) Cognition in tutoring systems

	AutoTutor	BITS	ELM-ART	JV²M	SQL-Tutor
Programming Language	Various	C++	Lisp	Java	SQL
Platform	Web	Web	Web	Desktop	Web
Content Structure	Topics & subtopics	Lecture notes organised by concepts	Lessons consisting of topics & subtopics	Virtual environment divided into levels	Problems based on DB structures
Task Framing	End of task – tutor summary	Probability gauge, learner directly asked if they understand the concept	Limited via topic explanation	Implied by story line	Complexity rating, visualisation of open learner model
Session Framing	None / Unknown	None	None	None	None
Feedback Used	Positive, neutral or negative feedback on last learner action	Immediate on answer submission, adaptive annotation	Immediate on answer submission	Unknown	Immediate on answer submission, level chosen by learner
Help Given	(via animated agent) prompts,hints, answers Misconceptions	Links to pre-requisite concepts, animated agent (Genie), hints based on inferred solution plan	Display learner's own code from previous examples, hints, email human tutor, learner forum	Hints supplied by an animated pedagogical agent (Javy)	Simple, error flag, hint, partial solution, list all errors & complete solution
User Control	None – sequential coverage of topics	Navigation support, pre-requisite recommendations, learning sequence generation	Adaptive navigation – restricted by learner model	None – sequential progression through levels	Full user control guided by system suggestions

The following list gives a brief outline of each of the tutoring systems compared in Table 2.3.

- There are many versions of AutoTutor (Graesser, Jackson and McDaniel, 2007; Graesser et al., 2001; Nye, Graesser and Hu, 2014), which corresponds to its domain independence and the numerous other versions created for research studies. The version that has been developed to respond to the learner's cognitive state is included here. AutoTutor is an animated conversational tutoring system that uses natural language dialogue to provide scaffolding throughout the tasks (e.g. it provides prompts, 'pumps' the learner for further response, answers and so on). As mentioned, AutoTutor is domain independent with the provision of authoring tools for the creation of domain-specific content.

- BITS (Bayesian Intelligent Tutoring System) is a tutoring system that supports the learning of the C++ programming language (Butz, Hua and Maguire, 2006; Butz, Hua and Maguire, 2008). BITS is organised into a series of lecture notes which are grouped by concepts. It provides learners with the opportunity to read the lecture notes and undertake multiple-choice quizzes, with feedback and help provided by the animated agent called 'Genie'.
- ELM-ART (Weber, 1996b; Weber and Brusilovsky, 2001) is a Web-based tutoring system used for teaching the Lisp programming language. ELM-ART is organised into a series of lessons consisting of topics and subtopics, using the metaphor of an electronic book, but with the ability to provide adaptive navigation and annotation based on the episodic learner model.
- In contrast to the other three systems, JV2M (Gómez-Martín, Gómez-Martín and González-Calero, 2006; Gómez-Martín, Gómez-Martín and González-Calero, 2004) is a tutoring system for teaching Java compilation which is designed as a virtual environment/game which sets out a story line that then carries the learners through the different levels. It represents the learner in the virtual environment as an avatar, with assistant available from an animated agent called 'Javy'.
- SQL-Tutor (Mitrovic and Hausler, 2003; Mitrovic and ICTG.Team, 2008; Mitrovic and Ohlsson, 1999) is a Web-based tutoring system used to support the learning of SQL. There is also a Windows-based version, but this was not used in this thesis. It is organised as a series of problems based around database structures to support different scenarios (e.g. computer shop and library). The learner model uses a Constraint-Based Modelling approach, which is used to model the learner's evaluative knowledge rather than their

generative knowledge (Mitrovic and ICTG.Team, 2006). Refer to Section 3.1.1 (p.60) for details of constraints and to Section 3.1 (p.58) for further information of the functionality contained in SQL-Tutor.

In relation to the two 'framing' rows in Table 2.3, the learning process involves reflecting and drawing upon past experiences and knowledge in order to construct new knowledge. Various techniques can be used by a tutor (human or computer-based) to aid the learner and encourage them to develop their metacognitive skills, such as the nature of feedback given and framing the tasks and sessions to explicitly focus the learner on drawing upon their past experiences. For example, providing feedback at the beginning of and during a task to relate to previous tasks (e.g. "do you remember when you worked on task x and the way you solved the problem? Can the same technique be applied here?"), as well as summarising the tasks and progress made at the end of a task (this also appears at a session level, which is defined here as the use of the tutoring system by a learner in one sitting/logon). Framing tasks and sessions to encourage the learner to think about their previous experiences and knowledge can also influence their motivational state (e.g. in expectancy-value theory in Section 2.1.1, p.8) explaining how a task fits into the overall scheme of the topic being taught can influence the value that the learner places on the task. In turn, this can have psychological and behavioural consequences). Unfortunately the systems presented in Table 2.3 provide limited forms of task framing (as discussed below), with none providing framing throughout the whole task (beginning, during and end of a task).

AutoTutor provides very limited framing by summarising the current task at the end of each task, with no apparent framing at a session level or at any other point in a task or session. The task summary is focussed on the domain knowledge of the topic covered and does not reference or review the learner's progress, although they can start with a personal tone. For example, starting the summary with "Right. Let's review what we've just gone over…." (Graesser et al., 2001) would be in keeping with the animated agent (human appearance) and the natural language dialog interface of AutoTutor.

BITS provides a gauge of the system's inferred probability (derived from the learner model) that the learner can complete the concept that they have clicked on. The probability gauge provides the learner with information to help them decide whether they should continue with the current concept, thereby encouraging them to think about their level of knowledge and their own beliefs on their ability to undertake the concept presented. The learner is asked directly, via options, once they have read through the lecture notes whether they understand the concept; "I understand this concept; I don't understand this concept; I'm not sure (quiz me)" (Butz et al., 2006). The first option updates the learner model to reflect the learner's understanding of the topic, although the learner's response is taken to be correct which could provide the learner with a sense of control over their learning. The other two options present the learner with a multiple-choice quiz in order to gauge and test their level of understanding.

ELM-ART (Weber and Brusilovsky, 2001) provides a very tenuous form of task framing by sometimes mentioning the previous topic or subtopic in the explanation for the current topic. However, ELM-ART does relate to previous tasks in one form of help given to the learner. When the learner requests help in solving a problem, ELM-ART uses a case-based reasoning technique to display a solution from past problems that closely matches the concepts and rules required in the current problem to provide a reminder to the learner (Weber, 1996b). Where possible ELM-ART will display the learner's own code from a past problem, although the impact of this tends to be lost as the formatting of the code is not always kept. Displaying past tasks as examples is a key benefit of the ELM-ART system as it encourages the learner to reflect on their past experiences and knowledge. Although such a reference is implied as the past task is not presented to the learner in a direct manner. For example, the effect of displaying a past task could be strengthened by explicitly drawing the learner's attention to it (e.g. "Do you remember when you did this task? Is there anything from the code that you wrote that can be useful to the current task?"). Refer to Appendix B for details of an experiment that has been undertaken that aimed to augment ELM-ART with such functionality via a 'Wizard of Oz' methodology.

JV2M does not explicitly provide task framing; rather it is implied via the learner progressing through the story line, with the final level using concepts from the previous levels.

SQL-Tutor uses its adaptive difficulty ratings to provide a form of link to previous and outstanding problems (refer to Section 3.1.3 (p.62) for further details). When the learner is presented with a list of problems, each problem is given a difficulty rating (e.g. '1' for easy through to '9' for really difficult). The ratings are dynamically generated each time the learner returns to the problem listing and are applied to all of the problems (completed as well as outstanding problems). Such ratings not only allow the learner to select a problem to work on, based on its indicated complexity, they can also gauge the difficulty of a problem by relating back to the rating that has been given to a problem that they have already solved. SQL-Tutor displays a visualisation (bar charts and percentages) of the open learner model between each problem so that the learner can gauge their progress and any areas of weakness in their SQL skills. Feedback is provided at a task level and is given immediately upon the submission of an answer. The level of feedback provided is chosen by the learner themselves by selecting from a list, ranging from 'simple feedback' to the 'complete solution'. The feedback level chosen is only applicable for the current answer submission, which means that unless the learner wants the default level, they have to explicitly decide and select the level of feedback that they think would be most beneficial to them for each submission.

None of the tutoring systems presented in Table 2.3 appear to provide framing at a session level, with the learner left to carry on from where they left the previous session. Framing at this level would be an ideal opportunity to draw the learner back into the tasks of the system by explicitly encouraging the learner to reflect on the previous session (or sessions) and remembering their progress, as well as the overall scheme of the topic being taught.

The remaining part of this section discusses the outcomes of studies that have been conducted to evaluate the tutoring systems reviewed in this section. The general nature of the discussion, as opposed to discussing individual studies in detail, is reflective of the fact that the tutoring systems presented in this section have undergone numerous studies relating to (meta) cognition. This is in contrast to the number of studies involving the tutoring systems focussed on motivational and affective states (refer to Section 2.6.3 (p.47) for further discussion).

As already mentioned there have been numerous versions of AutoTutor over the past seventeen years (Nye et al., 2014). There are three themes that have been core to the development of AutoTutor across all versions: 1) tutoring strategies inspired by human-tutors, 2) the use of pedagogical agents, and 3) the use of technologies which support the use of natural-language in tutoring. On average the learning gains reported across the AutoTutor studies have been approximately 0.8σ, with higher gains for deep learning than shallow" (Nye et al., 2014). The various studies have reported mixed results in skills relating to metacognition and self-regulated learning (SRL). For example, higher learning efficiency was reported in the AutoTutor-AS version (Nye et al., 2014). The think-alouds used in a study using the MetaTutor version showed that, as a SRL process, learning strategies were deployed 77% of the time, with metacognitive judgements deployed 16% of the time, showing that in a sixty-minute session "approximately two learning strategies every minute and made a metacognitive judgment approximately once every 4 mins" (Azevedo, Johnson, Chauncey and Burkett, 2010). Nye et al. (2014) note that "research needs to determine when tutoring SRL outweighs the benefits of spending time directly tutoring domain content".

While AutoTutor, ELM-ART and SQL-Tutor are all established ITSs that have been the focus of numerous research projects, BITS is an example of a tutoring system that showed early promise, but despite the plans of the BITS author's to formal evaluate the system from its planned use as part of a university programming module (Butz et al., 2006; Butz et al., 2008), at the time of writing no published work could be found that detailed any such studies.

Similar to AutoTutor, ELM-ART is a tutoring system that has undergone numerous studies, although it has not had quite the number of different versions as AutoTutor; ELM-PE, ELM-ART and ELM-ART II (Weber and Brusilovsky, 2001). Earlier studies appeared to be system focussed by evaluating the episodic learner model, as opposed to leaner-centric in terms of the learner's progress. However the result of one evaluation discussed by Weber (1996a) observed that "it sometimes happened that a previous error was produced again in the same context as before. This can be explained assuming that students try to solve a new task in analogy to a previous task - very often the task they just solved". This is in line with the view expressed by Robertson (2001) that as learner's we tend to be bad at linking problems we have just worked on (refer to the discussion relating to metacognition on p.18 of Section 2.2). As already mentioned on p.30 of Section2.5, this point of working on problems in isolation to previous problems was also observed by this thesis author while teaching students programming and was part of the motivation for the focus of this thesis. Studies that have evaluated the learning gains of learners using ELM-ART have shown that it is the learners with the little or no programming experience that tend to benefit more from the learning support provided by ELM-ART (Weber and Brusilovsky, 2001). This finding is comparable to a study conducted on the version of AutoTutor reviewed in the next section, in that it was the leaner's with low-domain knowledge that tended to gain the most benefit from the support provided by the tutoring system (refer to Section 2.6.3 (p.47) for further details).

Similar to BITS, JV2M appears to be another instance of the start of interesting research, but where no published work detailing evaluations of the tutoring game can be found. So it would appear that the system did not progress any further. It raised an interesting idea in relation to learning to program given that it is a complex domain, and that is how to develop a game environment with an avatar as a game companion/pedagogical agent instead of the tutoring environments presented by the other systems reviewed in this section. The game environment would have to not only be interesting/fun to play, but also effectively aid the learner in learning the domain.

As with AutoTutor and ELM-ART, SQL-Tutor is an established tutoring system that has been the focus was a lot of studies (Mitrovic, 2012), with only a few of studies mentioned here. Studies using SQL-Tutor have shown that the open student model, which is presented to the student as skill meters, has helped students to improve higher-level skills such as self-assessment and helping them to reflect on their knowledge (Mitrovic, 2012). Another study assessing the impact of the use of positive feedback in SQL-Tutor found that providing such feedback did facilitate learning as the participants needed fewer time and answered more problems that the study group that did not receive positive feedback (Barrow et al., 2008). The study conducted by Mathews and Mitrovic (2008b) showed that there was a direct correlation between learning gains and help seeking relating to both the frequency and the level of help. Students who requested higher levels of help the most tended to show the least learning gains.

2.6.3 Motivation in Tutoring Systems

Although this thesis is interested in the interaction between metacognition and motivation in a programming tutoring system, the number of existing ITS focused on the programming domain that are motivationally aware is limited compared to metacognitive tutoring systems. With this in mind, the five systems, regardless of their domain focus, are a cross representation of the techniques that are being incorporated into tutoring systems that gauge the learner's affective state. The systems reviewed are AutoTutor, Easy with Eve, MORE, Prime Climb and SELF.

Table 2.4 shows the main points of comparison between the five tutoring systems. The comparison focuses on the motivational constructs, measurements of motivation used, emotions recognised and the system actions based on the measurements. The following list gives a brief outline of each of the tutoring systems.

- As mentioned in Section 2.6.2 (p.39), there are many versions of AutoTutor (domain independent) in existence (Nye et al., 2014), but the version that includes cognitive and affective measurements is the one that is discussed here and is a different version to the one discussed in Section 2.6.2 (D'Mello, Craig, Witherspoon, McDaniel and Graesser, 2008b; Graesser, Chipman, King, McDaniel and D'Mello, 2007; D'Mello, Lehman and Graesser, 2011).

- Easy with Eve is an affective tutoring system that teaches mathematics (Alexander, Sarrafzadeh and Hill, 2006; Sarrafzadeh, Alexander, Dadgostar, Fan and Bigdeli, 2008).
- MORE (Motivational Reactive plan) is a tutoring system for teaching debugging in Prolog (del Soldato and du Boulay, 1995). It was one of the first tutoring systems to explicitly model and react to the motivational state of the learner.
- Prime Climb is an educational game designed to teach number factorisation (Conati and Maclaren, 2005; Hernández and Sucar, 2007; Conati, 2011).
- SELF (Self-Efficacy Learning Framework) is not a stand-alone tutoring system, but a framework which has so far been used against an online tutorial system and an interactive learning environment (Crystal Island), both of which teach the topics in the domain of genetics (McQuiggan et al., 2008).

Table 2.4 Motivation in tutoring systems

	AutoTutor	Easy with Eve	MORE	Prime Climb	SELF
Domain	Various	Mathematics	Prolog	Mathematics	Various
Platform	Desktop	Desktop	Desktop	Desktop	Desktop
Motivational Constructs	None specified	None specified	None specified	Goals	Self-efficacy
Measurements Used	Dialog history, facial expressions, body posture, speech parameters	Facial expressions & gestures	Temporal data, learner self-report	Infer goals from personality and interaction patterns	Physiological signals (heart rate & galvanic skin response), learner self-report, temporal data
Emotions Recognised to Aid Measurements	Anger, boredom, confusion, eureka, frustration, contempt, curiosity, disgust	Fear, anger, happiness, sadness, disgust, surprise	-	Joy/distress (emotion for game), pride/shame (emotion for self), admiration/reproach (emotion for agent)	-
Action Based on Measurements	Animated agent action (dialogue, expressions & gestures)	Animated agent action (dialogue, expressions & gestures)	Pedagogical & motivational action	Pedagogical action & affective action	Pedagogical planning, error correction, determining when to intervene with guidance

Most of the systems in Table 2.4 use more than one means of measuring the learner's motivational state. Three of the systems mentioned recognise emotions in order to aid the measurements used and help to infer the learner's motivational state. Emotions are inextricably related to motivation and cognition (Dai and Sternberg, 2004). For example, enjoyment can increase motivation and cognitive engagement, whereas boredom can decrease such motivational states. Emotions tend to be short-lived, but generally have a clear cognitive referent (Forgas, 2000). Easy with Eve uses facial expression and gesture analysis, similarly AutoTutor uses a combination measures such as body posture, speech parameters and the dialog history between the learner and the animated agent. Easy with Eve explicitly uses the six basic facial expressions relating to emotions defined by Ekman (1997), as referenced by Alexander et al. (2006), and it used an observational study of human tutors to help guide the actions/reactions the pedagogical agent should take in response to the learner (Sarrafzadeh et al., 2008). The applicability of these general emotions to learning are discussed in D'Mello et al. (2008b). In comparison to Easy with Eve, AutoTutor determined the emotions to measure, based on the compilation of emotions reported in a number of previous studies, which also used various forms of measurement such as 'emote-aloud' which is based on a 'think-aloud' approach, multiple-judge and retrospective self-reports (D'Mello et al., 2011).

SELF uses a mixture of physiological signals (heart rate and galvanic skin response), temporal data (e.g. the length of time spent on a task) and learner self-report to measure the learner's self-efficacy beliefs. The self-report measure is only used in the system training stage of the framework, as it is used to give meaning and validate the physiological and temporal measures. The self-report measure was designed following the self-efficacy scale guidelines by Bandura (2006) and is presented as a graphical slider bar ranging from zero to a hundred which the learner sets after they have solved each problem. The framework is designed to create dynamic models of learner self-efficacy in the training stage, which is then used to infer the learner's self-efficacy when the tutoring system is fully used.

MORE determines the motivational state of the learner by using temporal data and self-reports. The temporal data is used to measure effort and independence, while the learner's confidence is measured via self-reports and system inference. To determine the affective state of a learner, Prime Climb not only uses the data from a pre-use personality questionnaire and the interaction patterns during the sessions, but also from electromyography (EMG) data and learners' self-reports of their feelings for the agent and the game (Conati, 2011). It does not use any external measures to aid this inference. As with AutoTutor and Easy with Eve, Prime Climb measures a range of emotions, although it differs from the others by grouping the emotions in relation to the focus of the emotion (e.g. joy and distress are measured in association with the learner's affective state for the game itself. Table 2.4 details the other emotion-focus groups).

SELF is a general framework that could potentially be incorporated into some existing tutoring systems. With this view, how the measurements are used would be dependent on the tutoring system, but McQuiggan et al. (2008) provide example usage as informing pedagogical planning, error correction and aiding the decision on when the system should intervene by offering guidance.

Easy with Eve uses the affective measurements to decide on the action of the animated agent, including gestures and expressions used (Sarrafzadeh et al., 2008). MORE maintains a domain planner and a motivational planner in order to determine the next pedagogical action of the system. At times the recommended action from both of these planners may conflict with each other, so MORE includes a negotiation planner to resolve any such conflicts (del Soldato and du Boulay, 1995). Prime Climb uses a range of variables to maintain an affective student model (e.g. knowledge state, personality traits, goals, affective state). The tutor model combines an affective and pedagogical mode, each of which outputs a suggested action based on the current state. Prime Climb presents both an affective and a pedagogical action to the learner. For example, if the agent (Merlin) is providing the learner with an explanation, the pedagogical action would be the explanatory text appearing in Merlin's speech bubble and the affective action is Merlin extending his arms and appearing with a "conciliating" face (Hernández and Sucar, 2007).

Prime Climb and SELF have both been designed to explicitly measure a particular motivational construct; Prime Climb infers goals from the measurements made and SELF infers self-efficacy. One limitation with Prime Climb is that the affective model assumes that a learner has the same goals throughout the session, although this limitation is recognised and is being addressed in the model design (Conati and Maclaren, 2005). SELF creates a model of self-efficacy based on measurements obtained from one type of task in the given domain. McQuiggan et al. (2008) note that this is a limitation and that it is unclear whether such measurements could accurately be used as a universal measure. Neither AutoTutor nor Easy with Eve make any explicit mention in terms of the motivation construct used, although motivation is supported via recognising and adapting to emotions. While MORE was designed to measure the motivational state of the learner, it was not designed to measure a particular construct. However self-efficacy can be inferred by the measures of effort, confidence and independence within MORE.

When considering the studies that the systems reviewed in this section have had conducted to assess them, there is a difference to most of the tutors that were reviewed in the previous section (2.6.2, p.39). The systems relating to cognition and metacognition would appear from reviewing the available literature to have undergone much more evaluations than the systems relating to motivational and affective states. Such an observation would be consistent with the fact that researching tutoring systems that take into account the affective state is a far newer area of research, relatively speaking, than the longer established research into using tutoring systems to effect the learner's (meta) cognition. For example, the work by del Soldato and du Boulay (1995) noted that the "explicit teaching knowledge implemented in the current generation of Intelligent Tutoring Systems (ITSs) concerns mostly domain-based aspects of instructional processes, overlooking motivational aspects"; a view which was highly relevant at the time of their research. For this reason, the discussion of studies conducted against the systems reviewed in this section are presented at a much finer detail than the more general discussion of the studies in the previous section.

The version of AutoTutor discussed in this section was used in a study that covered the subject of computer literacy and was used alongside a base version of the tutor that did not contain the affective features. The study had eighty-four participants and involved three elements: 1) a pre-test, 2) use of the tutor across two thirty-minute sessions, and 3) a post-test (D'Mello et al., 2011). The study showed that the affective version of the tutor was more effective than the base version for low-domain knowledge students, but only on the second session and not on the first. This implied that that it was inappropriate to provide affective support to low-domain knowledge students until they actually needed such support. The study also showed that the low-domain knowledge students also performed better on knowledge transfer when completing tasks covering topics that were not included in the study sessions. The students with higher levels of domain knowledge did not gain anything from the affective support, showing that they did not require such support.

A study conducted with Easy with Eve (Sarrafzadeh et al., 2008) used four experimental groups, each using a different version of the tutor: 1) the tutor contained the pedagogical agent and facial expressions were detected, 2) only the detection of facial expressions was used with text-based feedback, 3) only the pedagogical agent was used and 4) neither the agent or facial expression detection was used, instead only text-based feedback was used. Data was obtained from fifty-nine participants (eight to nine year olds) who were randomly divided between the four groups. The participants' performance was measured via pre- and post-tests, with their opinions measured via a questionnaire at the end of the study. As only four computers where set-up, one participant from each of the experimental groups used the tutor at any one time. The participants each used Easy with Eve for one session which was designed to last approximately twenty-minutes (this excluded the time required for the pre-, post-tests and the questionnaire. In terms of the performance measures using pre- and post-tests, the two groups that contained only text-based feedback performed slightly better than the two groups that had the pedagogical agents. In terms of the two groups that used the versions that detected the participants' emotions via facial recognition, the groups where facial recognition was used performed better than the two groups with facial recognition. There was overall an increase in performance for all four groups, however, as there was no group that did not use Easy with Eve at all, the effect on performance of using any version of Easy Eve cannot be determined. In terms of the questionnaire, it produced mixed results as to whether the use of a pedagogical agent has a positive effect: 1) the two groups that only received text-based feedback found the use of the tutor more enjoyable than the groups that used the agent, and 2) for the two groups that used the agent, the group that also contained facial recognition responded more positively in their enjoyment of the system than the group without facial recognition. The results of the questionnaire also showed no difference in the participants' perception of whether they thought they were using a version that detected their facial expressions or not. Further work would need to be conducted as the results can only be treat as tentative and offer quite mixed results.

As MORE was designed as a framework that could be applied across tutors, a simple tutor for teaching Prolog debugging was designed and implemented in order to be able to test MORE (del Soldato, 1993). However, only a brief formative evaluation was conducted with comments gathered from a small number of participants. The evaluation involved the participants using the tutor for approximately an hour, where they reported their motivational states during the session, which was also recorded. The evaluation did confirm that the tutor did adjust its behaviour, but not all of the rules were appreciated by the participants (e.g. when the model determined that help should be refused because the learner did not need it). While such actions are employed and accepted by human tutors, the evaluation did raise the issue of the plausibility problem (refer to p.22 of Section 2.3 for further discussion of this issue). Despite the limited evaluation of MORE, the results have been influential due to their timing, as it formed the early stage of research to incorporate the learner's motivational state into a tutoring system.

In the Prime Climb system Conati and Maclaren (2005) discuss analysis was a study that was conducted to evaluate refinements made to the model's causal affective assessment within Prime Climb as a direct result from the outcome of previous evaluations of Prime Climb. The newer study was similar to design to the previous one, with sixty-six participants. A pre-test was used, as well as a personality test and a post-questionnaire to determine the goals that the participants had during playing the game. The participants were prompted to self-report their feelings towards the game and the agent throughout the use of the system, but unfortunately Conati and Maclaren (2005) did not provide details to the frequency of the self-report used during the study. Their study showed that participants who were generally successful were also either happy or neutral towards the pedagogical agent, which in turn suggested that the participants' positive feelings towards the game would also influence their feelings towards the agent. This is in direct contrast to the findings of the Easy with Eve study that was discussed earlier in this section where the groups that did not use the version of Easy with Eve that included a pedagogical agent reported higher levels of enjoyment of using the system. The potential reasons for such a contrast are wide-ranging (e.g. the appearance of the agent, the interactions and feedback provided by the agent, the timings of such feedback), although with only one study conducted using Easy with Eve, it would suggest that additional refinement and studies would need to be conducted in order to start to get comparable results on which strong conclusions could be drawn. The results from the Prime Climb study in terms of the accuracy for determining the affective state of the participants was generally increased when compared to the pre-refinement model (e.g. emotions such as joy, distress and admiration increased from accuracy levels in the low to mid 60% to levels in the low to mid 70% range. However, the accuracy for reproach dramatically reduced from 80% to 38%. The study's authors conclude that this change is due primarily to two factors: 1) the goals that were declared by the participants at the end of the game session did not match their goals throughout the game and 2) using only the participants previous knowledge of maths to help assess their attitude towards wanting help meant that some of the participants were incorrectly modelled. The study found that the outcome of the participants' moves influenced the satisfaction of the 'Have Fun' goal and whether they were given a large fraction number also influenced the satisfaction of the 'Learn Math'. Such findings are in-line with studies that showed "that students' self-report of emotion depends on events that occurred in the previous problem" (Arroyo et al., 2009) which was discussed in the use of self-reports on p.24 of Section 2.4. The Prime Climb also highlighted that two of the assumptions made prior to the study by the study's authors were not completely correct: 1) the goal that the learner is trying to achieve does not stay the same throughout the session and 2) that such assessments using these goals cannot be done without modelling goal priority. Such findings are also in line with the widely recognised view that affective states undergo continuous changes throughout the learning process (refer to the start of Section 2.1 (p.6) for further discussion of this point).

Like MORE, SELF is a framework which is designed to be applied across tutors, as opposed to an ITS in its own right. McQuiggan et al. (2008) discuss the outcome of two similar studies using SELF, where they followed the same procedure but in the first study SELF was incorporated with an online tutoring system and an interactive learning environment, Crystal Island, in the second study. The first study had thirty-three participants, with forty-two in the second study. Both studies used self-reports of self-efficacy using a hundred point scale presented as a slider control to the user, as well as physiological data (heart rate and skin galvanic data which was collected approximately thirty times a second). This data was incorporated with the pre-study demographic survey and a general post-study survey. In both studies SELF constructed two models: 1) static which used self-reports of self-efficacy and demographic data and 2) dynamic that also incorporated the physiological data. A session consisted of the participants completing the pre-study survey, reading the domain problem instructions, during which time base physiological data was taken, then they completed twenty multiple-choice problems and had to provide self-reports of self-efficacy before moving onto the next question. The results of the self-reports were used to produce four models of varying degrees of granularity for reporting self-efficacy (e.g. two levels with low and high ratings through to five levels with very low, low, medium, high and very high ratings). The results found that age was the only attribute from the demographic survey that had a significant effect on all levels; when using higher levels of granularity for modelling self-efficacy the studies showed that it become more important to account for the participants; demographics. Both studies were consistent in their findings when assessing the accuracy of determining the participants' self-efficacy using the static and dynamic models. Both studies showed that the static model was 73% accurate at predicting self-efficacy, while the dynamic model was 83% accurate in the first study and 87% in the second study. However, the study's authors noted that there was "a noticeable decay in model performance as the granularity is increased in both evaluations." (McQuiggan et al., 2008). Both SELF and Prime Climb have reported consistent results from more than one study against the systems. Although Prime Climb showed an increase in accuracy between studies it can be attributed to changes made to the underlying affective model used in the system. The studies for SELF did not report any changes to it between the studies and its consistent results show good promise, especially when considering that SELF is a genetic framework for predicting self-efficacy which is designed to attach onto other tutoring systems aimed at differing domains.

2.7 Study Relevance

The review of the existing research that has been presented in this chapter has discussed research areas that are relevant to this thesis study and has been used to show the relevance of this study. The focus of this study has been to explore the effects to the learner of incorporating, into an ITS, motivational and metacognitive feedback based on the learner's past states and experiences in order to improve the learner's focus on the process and experience of problem solving (refer to the research questions in Section 1.2, p.3). In order to establish what motivational construct to use in this study, contemporary theories of motivation in learning were discussed in Section 2.1 (p.6). The discussion showed that one construct is a key mediator across all of the contemporary theories: self-efficacy. Section 2.2 (p.16) discussed the role and effect of motivation and metacognitive during the learning process, especially with focus on problem-solving which is a key skill required for using SQL-Tutor (the ITS used in this study). The review of existing research in the areas of motivation and metacognition in learning showed that they are key components in the learning process that have a tightly entwined relationship with each other; motivation can effect metacognition and vice versa.

As this study explores motivational and metacognitive feedback, Section 2.3 (p.18) discussed the role and influence of feedback on the learning process, especially focussing on the potential difference between the initial intent of feedback and its wider impact, along with the effect feedback can have on a leaner's self-efficacy. In order to provide motivational and metacognitive feedback which is based on the past states and experiences of the learner, this study needed to also determine how motivational states and past experiences could be measured. In Section 2.4 (p.22) discussed existing research relating to different methods for gauging the motivational states and how to draw upon metacognitive skills. From this discussion, the use of self-reports was shown to be the most direct method for gauging the learner's self-efficacy levels in-situ during the learning process. Section 2.5 (p27) discussed the problems faced by people learning to program as the domain of the ITS used in this study involved learning SQL SELECT statements which is a programming, problem-solving domain.

The focus from the chapter sections so far were combined in Section 2.6 (p.34) which discussed what an ITS is, then reviewed research using existing ITS: five metacognitive systems and five motivational systems. Although this chapter acknowledges that existing research has been conducted on tutoring systems that cover a wide spectrum of domains, the systems reviewed in Section2.6 focussed on the programming domain as it was the target domain in this study. The review showed that existing research has focussed primarily on either motivation or metacognitive, but not explicitly on both within the same research study. This leads to the unique focus of this study where motivation and metacognition with past states and experiences was the central consideration from the outset of this study.

2.8 Chapter Summary

This chapter has discussed the contemporary theories of motivation in learning, along with the influence that metacognition and motivation have on the learning process and the entwined relationship that exists between them. The content of feedback was also discussed, especially feedback that includes the learner's previous experience and motivational states during the use of an ITS. Measuring a learner's self-efficacy via self-reports was also discussed, before reviewing several existing ITSs. The review of the existing ITSs has shown that some explicitly focus on motivation and some explicitly on metacognition, but none to date has focussed on both motivation *and* metacognition by referring to past learner states and experiences. This is the unique focus of this thesis.

3 Update SQL-Tutor System Design

Given the research questions outlined in Chapter 1 (p.1), developing a new ITS, whilst it would be an extremely interesting development project, was far beyond the scope required by this research thesis. This chapter discusses the existing ITS SQL-Tutor which was selected for use in this thesis, the base functionality provided by SQL-Tutor and the two levels of modifications that were implemented for this thesis: modifications applicable to the versions used by both the Control and Study groups, and the modifications central to this research thesis.

3.1 Base SQL-Tutor System

The choice to use SQL-Tutor as the base tutoring system was based upon two primary facts: the learning domain of SQL SELECT queries would provide the largest pool of students to ask for study volunteers and SQL-Tutor was already an established, mature ITS. At the stage in the thesis where an existing ITS was chosen, it was still uncertain if more than one study could be undertaken and whether volunteers were going to be sourced from the Universities of Sussex or Northumbria. SQL SELECT queries were widely taught at both of the Universities. The development of SQL-Tutor initially started in 1995 by Tanja Mitrovic as the leader of the Intelligent Computer Tutoring Systems (ICTG), University of Canterbury, Christchurch, New Zealand. SQL-Tutor successfully made the transition from a pure research-based system to actively being used in classrooms in 1999 at the University of Canterbury and it has been available as part of Addison-Wesley's DatabasePlace Web portal since 2003 (Mitrovic and ICTG.Team, 2006; Mitrovic and Ohlsson, 1999). SQL-Tutor itself has been used in over eleven evaluation studies since 1998. For example, the use of eye-tracking to improve learning (Najar, Mitrovic and Neshatian, 2014), negotiable student models (Thomson and Mitrovic, 2010) and use of positive and negative feedback (Barrow et al., 2008).

SQL-Tutor is an ITS that provides an environment for learners to practise and develop their SQL SELECT query writing skills. The tutor assumes that the learners have covered the concepts of relational databases and of SELECT queries prior to using it. It is designed as a practice tool that complements learning of the related concepts via other sources (e.g. lectures). SQL-Tutor is available in three versions covering the Solaris, Windows and Web platforms (Mitrovic, 2007), all of which were developed using Allegro Common Lisp (CL) from Franz Inc. This study has used the Web version, as not only is it the most widely used version, it also provides the greatest level of flexibility to the learner in terms of access. Figure 3.1 shows the architecture of the Web version. SQL-Tutor presents the learner with problems that require the learner to write an SQL SELECT query to solve the problem. There are just under three hundred problems in total, which are presented across thirteen different databases (e.g. books, CD collection, and movies).

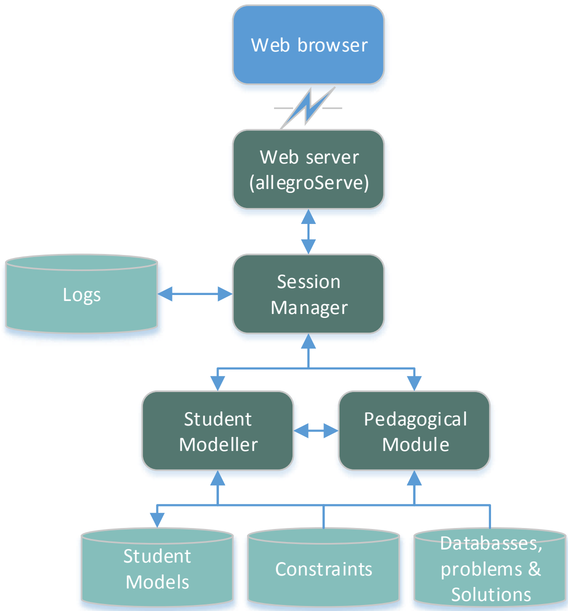


Figure 3.1 SQL-Tutor Architecture of the Base Version.
Adapted from SQL-Tutor web page, Mitrovic (2007)

The 'Session Manager' directs all user interaction with SQL-Tutor for the logon session, as well as automatically logging the user out if there has not been any activity for thirty minutes. Every time a student submits an attempt at a problem, the 'Student Modeller' analyses the submitted code to identify any mistakes, if there are any, and updates the student model. The 'Pedagogical Module' generates feedback when required, as well as selecting the next appropriate problem based on data received from the student module.

3.1.1 Constraint-Based Modelling

At the core of SQL-Tutor is Constraint-Based Modelling (CBM) which was proposed by Ohlsson (1994). With CBM domain knowledge is held in constraints against correct solutions, with each constraint representing "an atomic fact or principle of the domain" (Mathews, 2012). There are over seven hundred constraints in SQL-Tutor. The formal notation for a constraint is an ordered pair <C_r, C_s>, where C_r is the 'relevant condition' and C_s is the 'satisfaction condition' (Mitrovic and Ohlsson, 1999). Read the ordered pair like an *if... then...* statement, in that if the 'relevant condition' is relevant to a solution, then the 'satisfaction condition' must be satisfied/true for the constraint to be met. Otherwise, the constraint is considered violated and an error has occurred. The following is an example of a constraint from Mitrovic (2012).

```
Constraint 358:
Cr: the student's solution contains the JOIN and ON keywords in FROM
Cs: the FROM clause must match the following pattern
     (?*d1 ?t1 ??s1 "JOIN" ?t2 ??s2 "ON" ?a1 "=" ?a2 ?*d2)
```

In a constraint-based tutor the model answers to the problems are tested against all of the constraints, which identifies which constraints are relevant for each problem. The above example shows how SQL-Tutor has in-built flexibility when checking the learner's answer. SQL-Tutor is not just restricted exactly to the model answer, which is especially beneficial to the learner as there is often more than one version of an SELECT query that will correctly answer the problem. When a learner submits an attempt at solving a problem, their answer is tested against the relevant constraints and, if there are any violated constraints, their answer is incorrect, with the tutor formulating the feedback based on the current help level. A CBM maintains a history per user of the constraints used and the number of times each one has been violated. For example, '(123 (1 1 0 0 1))' shows that constraint number 123 has been triggered five times, but on two occasions (third and fourth) it was violated. This constraint history forms part of the open learner model implemented in SQL-Tutor. It can also be used in analysing post-study results, for example, it is used to calculate learning curves for learners and groups of learners (refer to Section 5.3.3 (p.151) for the group learning curves relating to this study).

3.1.2 Open Student Model

The Student Modeller module of SQL-Tutor maintains a learner model (also referred to as student model) for each person who uses the tutor. The student model holds the data that SQL-Tutor uses to determine where in learning SQL SELECT queries the student is. It is an open model because SQL-Tutor displays a graphical representation of the current model to the student. Figure 3.2 shows a Web page where the student model is displayed at the top half of the Web page. In this Web page shown, the page also shows SQL-Tutor's suggestion on the next SELECT clause it has determined the student should work on next.

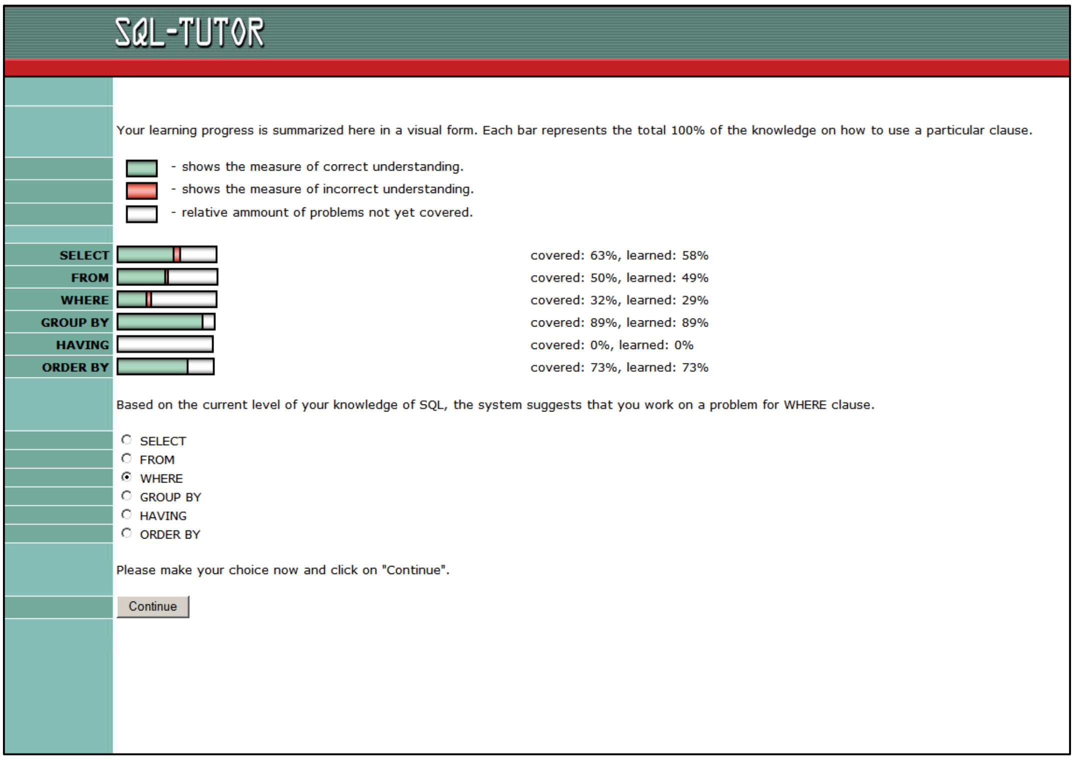


Figure 3.2 Web page showing the open learner model
and a suggestion as to what part of the SELECT query to concentrate on next

The display of the open student model occurs in one of two ways: automatically by SQL-Tutor, as shown in Figure 3.2, or the student may select to view their current model via the 'Student Model' button on the main problem Web page (see Figure 3.4, p.64). SQL-Tutor is an *intelligent* tutoring system (ITS) because it maintains the open student model and the system's ability to use the model in order to adapt to each student as an individual.

3.1.3 Difficulty Levels

As a form of feedback and to provide the student with information on the basis of which they can inform their decision of which problem to attempt next, SQL-Tutor annotates each problem with an initial difficulty rating which is the same for all students. Figure 3.3 shows the list of problems that are relevant to the SELECT query clause they selected on the Web page shown in Figure 3.2. The difficulty ratings appear as numbers between one (easiest) and nine (hardest), immediately to the left of the problem number.

The difficulty ratings for all problems are re-calculated by SQL-Tutor after a student has attempted a problem (regardless of whether the problem was solved or remains unsolved). This continuous re-calculation is so that the difficulty rating is up-to-date with respect to the progress of that student, which is important as it is used by the open student model to reflect the system's perception of the student's domain knowledge. The difficulty ratings are also influenced by the number of constraints for a problem and/or the complexity of the constraints (Mathews, 2012), and are specific to each student and their progress at solving the problems.

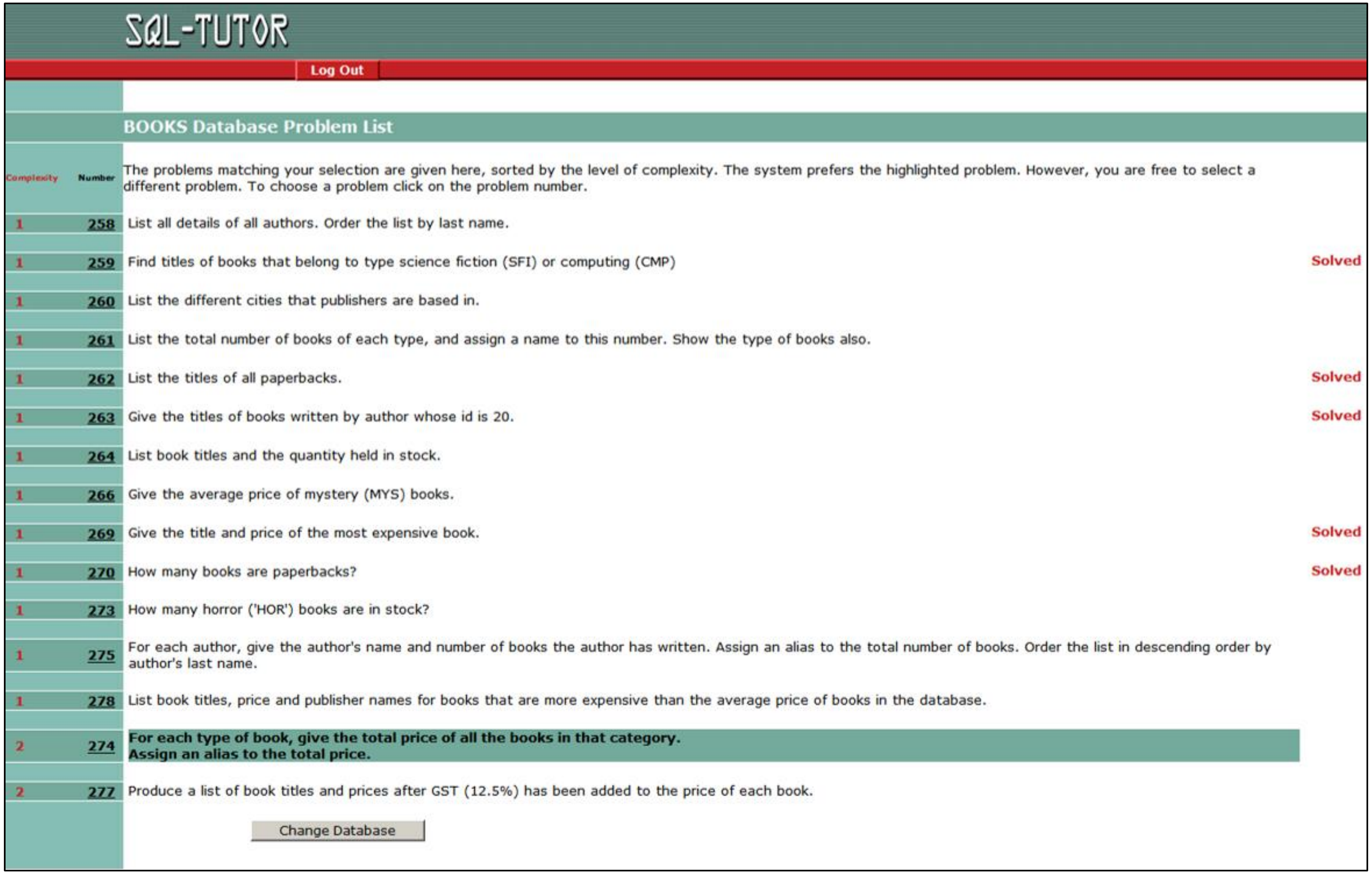


Figure 3.3 List of problems

3.1.4 Feedback Help Levels

As standard in the base version, while working on a problem SQL-Tutor provides levels of help via one of six levels of feedback. Figure 3.4 shows the main problem page with the feedback level drop-down list box expanded to show the available levels of feedback. The type of feedback chosen by the student affects what feedback they will receive after they have submitted their answer. The feedback is displayed in the large text field shown on the right of the form, where currently "Well done, choose another problem." is shown Figure 3.4.

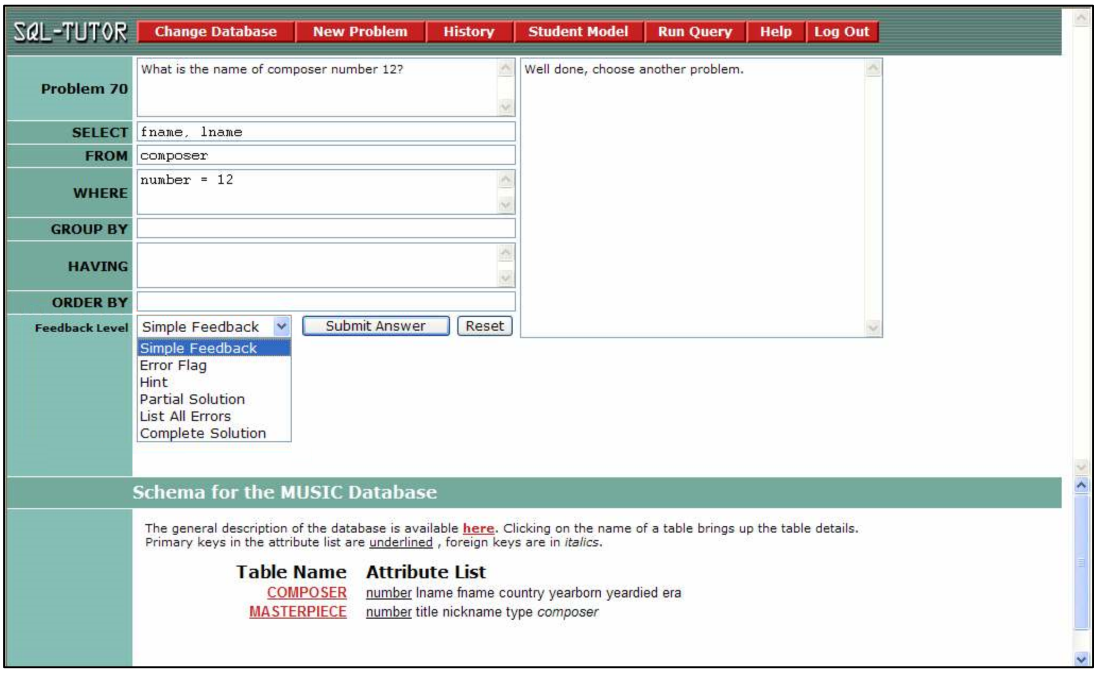


Figure 3.4 The main problem page showing the available help levels in the drop-down 'Feedback Level' combo box

Internally each help level is numbered from 0 = 'Simple Feedback' through to 5 = 'Complete Solution', this is purely for internal purposes and ease of reference, with the numbers unknown to the student. When the student first opens a problem in the main problem page, the system sets the help level to the lowest level of help ('Simple Feedback') and automatically increments the help level by one up to 'Hint' (2) after every attempt resulting in an incorrect answer. The student must explicitly chose one of the final three help levels, as SQL-Tutor will never automatically display feedback relating to these three higher levels of help. However, the student can override the help level selection at any time (i.e. they could select to view the 'Complete Solution' when submitting their very first attempt at a problem).

Figure 3.4 also shows four other forms of help that is available on the main problem page. In the toolbar displayed at the top of the page there are the 'History', 'Run Query' and 'Help' buttons. The 'History' button displays in a separate window the student's past attempts at problems for this session; this information is not saved, so cannot be recalled in the next session. The 'Run Query' button will attempt to run the SQL SELECT code entered by the student and display the results in a separate window. The 'Help' button purely displays the default help text shown in the feedback box on the right of the page which is shown when a student selects to open a problem and the main problem page is shown (refer to Figure 3.6 for an example of the help text in the text box towards the top, right of the page). This was the functionality in the base version as received for this study and may not be reflective of other SQL-Tutor versions. The final form of help is the display of the table and fieldnames in the bottom half of the page which provides an aide memoir for the student of the database structure they are currently working with. Clicking on a table name will display the fieldnames, data types and keys for the selected table.

3.2 Base System Modifications

There were two levels of modifications made to the base version of SQL-Tutor for this study: (1) modifications that affected the versions used by both the Control and Study groups, and (2) modifications which implemented the functionality that is central to this thesis (providing motivational and metacognitive feedback to the learner, based on their past states and experiences). This section details the first level of modifications; the modifications used by *all* study participants.

3.2.1 Database and Problem Review

A full review of all of the problems across the thirteen databases was conducted in order to make sure that the contents of any problems (especially the model answers) did not have a negative interaction with the material being taught on the database module on which the participants were registered. Section 4.2 (p.97) details an example where the reverse situation unexpectedly occurred: a particular table *join* short cut was taught on the database module, but there were no constraints inside SQL-Tutor that handled the shortcut to determine if its usage in the answer was correct. This resulted in some confusion and frustration among the participants who tried to use the shortcut inside SQL-Tutor.

The following list details the review changes made.

- During initial testing and familiarisation sessions using SQL-Tutor, the author attempted a problem that continually returned errors when trying to use the 'Run Query' function. As SQL-Tutor uses Microsoft Access (MS Access) as the database management system for running the query code, the query still resulted in errors when executed directly inside MS Access. MS Access 2007 was the version that this was tested against. The issue was eventually found to be the way in which MS Access formatted queries with more than two table joins. If there was more than one join, MS Access required parentheses placed around the first join. For example,

  ```
  FROM (TableA INNER JOIN TableB ON ….)
       INNER JOIN TableC ….
  ```

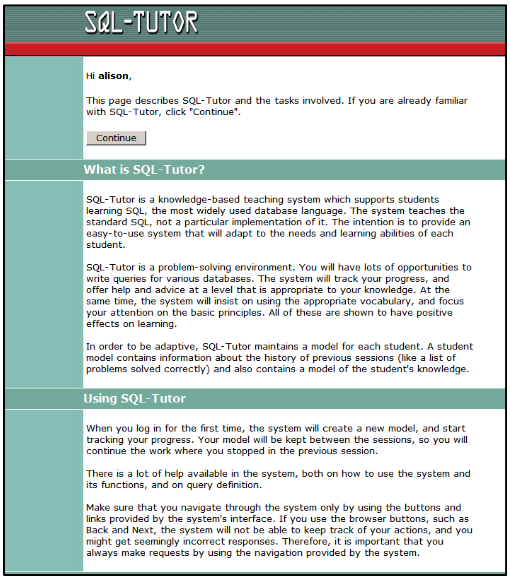
  As the parentheses are not required in this scenario as part of the SQL standard, any problem that contained more than one table join as part of the solution was removed.
- Any problem that included the use of EXISTS in the WHERE clause was also removed, as this function was not taught in the database module.
- Any query that had a table join in the WHERE clause, the model answer was changed to place the join in the FROM clause, unless the problem explicitly requested the join to be in the WHERE clause. These were tested in SQL-Tutor to make sure the queries would still be recognised as correct. The reason why the joins were moved is that the module presented the joins in the FROM clause.
- The 'Woodwork' database was completely removed purely because after removing queries as detailed above, it left only seven problems.

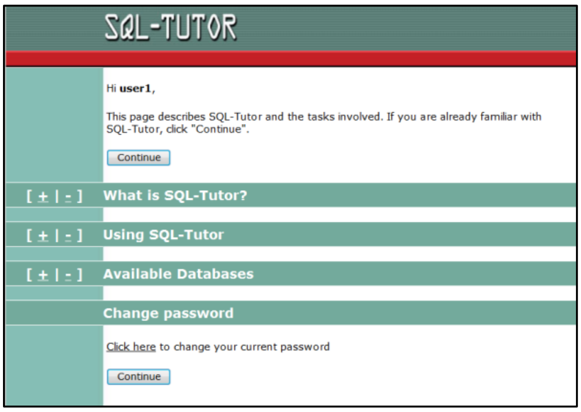
The database and problem review resulted in the number of databases reduced from thirteen to twelve and the number of problems reduced from two hundred and seventy eight down to two hundred and thirty six. This still left a good variety of problems and database scenarios for the study participants to work on.

3.2.2 User Interface Modifications

From the initial sessions using SQL-Tutor, the author made a series of small interface changes that were not part of the focus of this thesis, but were felt to be generic interface modifications that should be available to both the Control and Study groups. Interface modifications were with consideration to international web design guidelines; Bevan (2005) provides a succinct comparison of the guidelines from the U.S. Department of Health and Human Services (HHS), the International Standard ISO 9241-151 and the UK JISC (Joint Information Systems Committee for higher education). All three set of guidelines have commonality and overlap (e.g. the JISC guidelines (Bevan and Kincla, 2003) were produced for creating academic websites and were largely based on the HHS guidelines). The guidelines aim to address a key issue in any computer, and non-computer, system (not just web-based ones) which is the users' perceptions of *usability*. Refer to such sources as Lindgaard, Fernandes, Dudek and Brown (2006), Nielsen and Loranger (2006) and Robins and Holmes (2008) for discussions on web-based usability issues. Although the web design guidelines are not directly targeted at web-based tutoring systems, the research leading to the JISC guidelines found that "the main guidelines are widely applicable across most types of web services" (Bevan, 2005). The issue of usability, and indeed credibility as discussed by Nielsen and Loranger (2006), was relevant to the use of SQL-Tutor in this study; refer to the issue with navigation and lack of logout options discussed in Section 4.2 (p.99), and the discussion of the study participants' views of SQL-Tutor's interface as obtained from the post-study questionnaire (in Section 5.2.2.2, p.127 and p.128).

The first interface modification came about because of the inclusion of self-reports for the Study group (discussed in Section 3.3.1, p.73). The first stage in a session where a Study group participant was to be prompted to report on their self-efficacy was at the start of the session. Immediately following the logon Web page the welcome page is displayed, as shown in Figure 3.5a, which provides the learner with introductory information about SQL-Tutor.





↑ Fig. 3.5b The modified welcome page

← Fig. 3.5a The original welcome page

Figure 3.5 Shows the welcome Web page pre- and post-modification

The Web page that follows the welcome page is the database selection page for the learner to select which database problems they would like to work on. In terms of prompting the Study group participants for the start of session self-report, the ideal location in terms of page flow and timing, as determined by the author, was between the welcome and database selection pages. However, placing a completely separate self-report page in between these two pages had the potential of being reducing usability in terms of the number and flow of pages to navigate just to start the session. Therefore, the introductory text on the welcome page was formatted into expandable and collapsible sections, which were automatically shown collapsed when the Web page was first displayed. This allowed room for the self-report section displayed to the Study group participants (refer to Figure 3.12, p.81), but it was also introduced into the version for the Control group as it presented them with a shorter welcome page that was not a page full of text. This modification was in line with several of the HHS web design guidelines as listed by Bevan (2005) e.g. "6:1 Set Appropriate Page Lengths ", "15:9 Limit the Number of Words and Sentences " and "16:1 Organize Information Clearly".

Figure 3.6 is the main problem window, showing all of the modifications that the Control group participants could see. This page is where most of the interface modifications occurred.

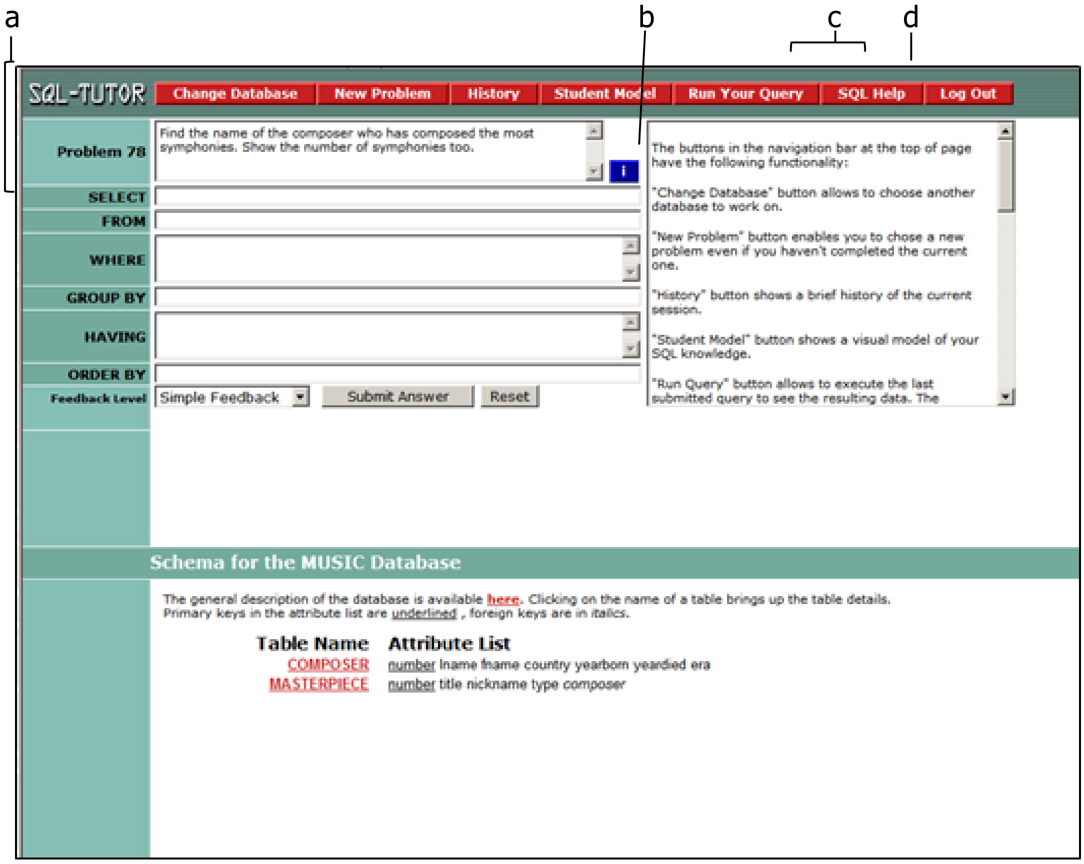


Figure 3.6 Shows the main problem window with the user interface modifications (a, b, c, d, e – refer to the following list below)

The following list details the user interface modifications made to the main problem page. The letters of the list items correspond to the letters at the top of Figure 3.6.

a) Tooltips/alternative text was placed on all of the toolbar buttons and the new 'i' button. This means that when the mouse pointer hovers over the button without clicking on it, the text is displayed as help to explain the functionality of the button. For example, the tooltip text on the 'Change Database' button is "Change the database scenario to work on a different set of problems". This modification was in line with several of the HHS web design guidelines as listed by Bevan (2005) e.g. "10:2 Avoid Misleading Cues to Click " and "10:4 Use Meaningful Link labels".

b) A new 'i' button was located immediately to the right of the text field that contains the problem text. Clicking the button displayed the expected output from the model query in a separate window (refer to Figure 3.7). This covered those instances when a learner was still unsure what was required even after reading the problem text. It provided the

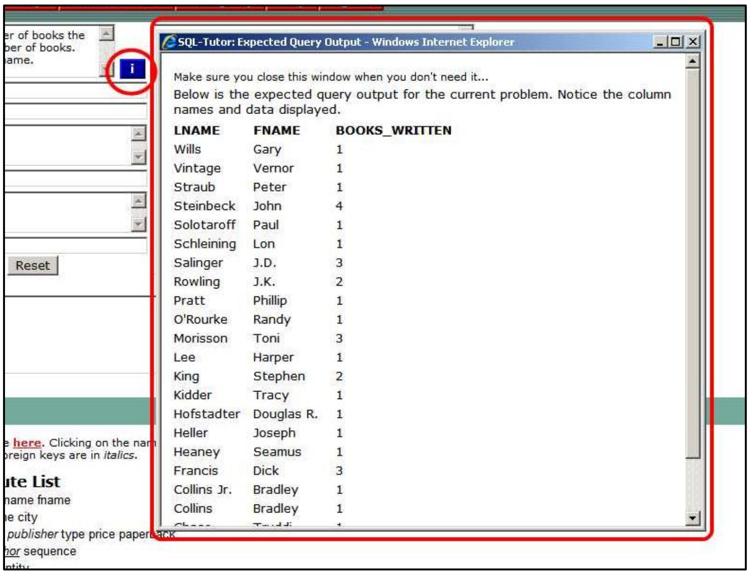


Figure 3.7 Separate window that displays the expected query output (user displays by clicking on the 'i' button)

learner with the query result that was expected, but without actually giving any of the answer away. For example, the first part of problem number 275 in the 'books' database asks 'for each author, give the author's name and number of books the author has written…'. In the database, the author's name is divided into two fields (LNAME and FNAME), but it is unclear from the problem text what exactly is required for the name. Figure 3.7 shows the expected query output for this problem and it shows that both name fields are used. The letter 'i' (for 'information') was chosen as the button label for two reasons: to maintain the interface standard and space restrictions. All of the buttons used in SQL-Tutor have text labels only and no graphics/icons, so the new button kept to that standard. From a usability viewpoint, the new button was ideally suited to be located close to the problem text, so that it was readily at hand for the learner, if they did not fully understand the problem text. Locating the button immediately to the right of the problem text meant that the problem text field had to be narrowed slightly to accommodate it. This meant that the new button had to be as small as possible. Choosing the button background colour of blue made the button standout from the standard green and red SQL-Tutor interface theme. This modification was in line with several of the HHS web design guidelines as listed by Bevan (2005) e.g. "6:5 Place Important Items Consistently" and "6:7 Use Moderate White Space".

c) The labels on two buttons were changed slightly to give them clearer meanings. The 'Run Query' button was changed to 'Run Your Query' and the 'Help' button was changed to 'SQL Help'. This modification was in line with the HHS web design guideline "10:2 Avoid Misleading Cues to Click " (Bevan, 2005).

d) Related to the change of label to 'SQL Help', the functionality of this button was also changed. Instead of the default help text that was shown when the main problem page was opened (refer to the text shown in Figure 3.6, p.69), this button was set to open an external Web site in another browser window. The Web site used was the SQL SELECT tutorial and help pages of w3schools (www.w3schools.com). This was changed in order to provide the learners with an instantly accessible, online SQL source for help and reference.

One final user interface modification was proposed, but was not implemented before the start of the study due to development deadlines governed by the study start date as planned by the author. The only 'Logout' button in SQL-Tutor appears on the main problem Web page. It was planned to include a 'Logout' button on all of the other main SQL-Tutor pages after the welcome page in order to promote correct logout actions by the learners. This modification was implemented after the start of the study, as it was determined by the author that such a change did not influence or interfere with the learning process (refer to Section 4.2 (p.97) for further details). The 'Logout' button was put on the database selection, open learner model and problem list pages.

3.3 Learner's Past States and Experiences

This section details the code modifications made to SQL-Tutor that were central to this thesis and only available to the Study group participants. These modifications were designed to provide adaptive motivational and metacognitive feedback to the Study group participants based on the self-reports of self-efficacy and the previous problems they had solved. While the changes and the versions of SQL-Tutor used in the study were referred to as two separate versions of the base system, there is physically only one instance of the code running on the Web server. The Control and Study group participants were presented with two different interfaces and functionality by using their username (e.g. 'user14' and 'user53') to identify which group they belong to and manipulate the interface accordingly; all the usernames ending in an even number belonged to the Study group and the odd number belonged to the Control group. For ease of reference, the two sets of functionality are referred to as two versions in this thesis. Comparing two versions of an ITS that differ only in a particular feature is a well-known and much used tactic in artificial intelligence in education (AIED) evaluation research. For two examples of this technique, see Mark and Greer (1995), and Luckin and du Boulay (1999). For more general discussions of this issue see Mark and Greer (1993), and Shute and Regian (1993).

Figure 3.8 extends the base SQL-Tutor architecture diagram from Figure 3.1 (p.59) to illustrate the modifications that were made for this study. The area bordered with a dotted line represents the new functionality. The modification included a new module called 'Feedback Rules Engine', additional log files and files containing the content rules and feedback statements. The 'Feedback Rules Engine' was responsible for generating all of the additional feedback which is motivational and metacognitive based on the past states and experiences of the participants. Refer to Section 3.3.2 (p.78) for further details of the additional rules engine.

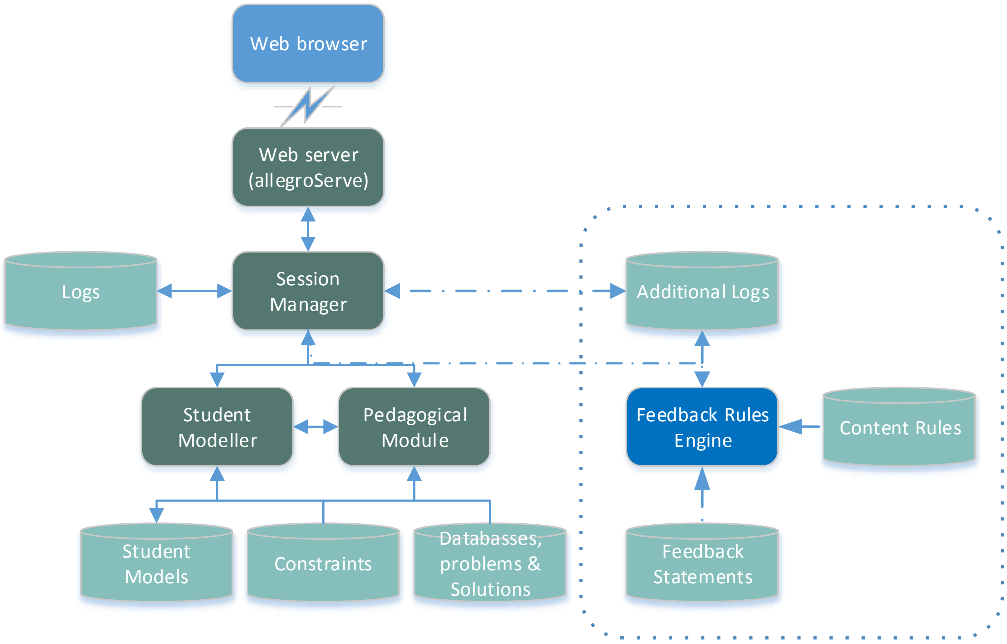


Figure 3.8 SQL-Tutor Architecture of the Study Group Version

3.3.1 Human Tutor Interaction Experiment & User Testing

Prior to detailing the modifications that were central to this thesis, two small tests are explained, as their results were influenced the modifications.

3.3.1.1 Human Tutor Interaction Experiment

Near the beginning of the thesis workl, a small design experiment was conducted to observe the effect on the learner of using the Lisp ITS called ELM-ART[2] (Weber and Brusilovsky, 2001). The ITS was used by one person (the author), with an additional person taking on the role of the human tutor who referred to the learner's past interactions with the aim of aiding motivation and metacognition. The methodology used was inspired by and was a kind of "Wizard of Oz" methodology (Kelley, 1983; Green and Wei-Haas, 1985), where the human tutor mimicked part of the functionality of the proposed system, while an existing web-based system (ELM-ART) provided the "base level" interactivity. It was not a standard "Wizard of Oz" experiment because the learner in this experiment was aware that the additional feedback was being supplied by a human, whereas in a standard "Wizard of Oz" experiment the learner would not be aware what

[2] ELM-ART was available at http://art2.ph-freiburg.de/Lisp-Course (01/04/2010)

interaction or feedback came from a computer system or a human. As already discussed in Section 2.6.2 (p.39), ELM-ART was chosen as it refers the learner, if they choose, to the code from a previous exercise in order to help them with the current exercise. Although the results of such a small experiment, which also used the author as the learner, is very limited in producing meaningful results, the experiment was purely conducted early in the research degree process to gain an insight into referring learns to their past code as used in ELM-ART.

The experiment involved two sessions (two hours each) held one week apart, with one person, the author, who had no LISP experience (but was an experienced programmer) taking the part of the learner and another person, who had limited past experience of LISP, as the human tutor. Both participants kept a written record of the interactions, emotions and motivation throughout. ELM-ART consists of six lessons, each divided into subtopics. This experiment covered the first three lessons.

The motivation of the learner in this experiment was different from a student who was using the ELM-ART ITS as part of their University course. While the learner had a level of intrinsic motivation to learn, extrinsic motivation was dominant (e.g. using ELM-ART, not with the primary aim of learning LISP, but to experience the functionality provided by ELM-ART, along with the additional interactions from a human tutor).

The interactions initiated by the human tutor across both sessions can be categorised into three types, as detailed in Table 3.1 (p.75). While the learner felt that all of the types of human tutor interaction helped to maintain interest and task-focus, the main emphasis (as wanted) was on the interaction where the human tutor reminded the learner of either the past content or their own past work/answers. This type of interaction aided the learner in answering the current question by prompting their recognition and providing them with an opportunity to form a link between past experience and the current question. It also aided the learner’s sense of achievement at having derived the correct answer without having to be explicitly told it, with feelings of accomplishment and raised self-efficacy being experienced.

The human tutor interactions in this experiment had two limitations: (1) only one direct reference was made to the learner's motivational state (the tutor asked the learner if they were bored during answering a series of questions without a break of an explanation page) and (2) the reference to past content and learner's work did not directly reference the actual steps that a learner took to answer a past question. This could have been influenced by the short length of work required to answer even the coding questions.

Table 3.1 Types of interaction with human tutor

Type of Interaction	Context When Occurred
Tutor reminding learner of past content & work	Largely when the learner was attempting a question and sometimes when reading through an explanation page
Tutor explanations & discussions	After the learner got an answer wrong or when the human tutor could see that there was a pause in the learner's actions/progress (e.g. if paused while reading an explanation page or when attempting a question)
Tutor questions regarding learning preferences	During session two when the style and frequency of questions changed per lesson subtopic

The experiment did show that one of ELM-ART's major features (displaying a solution from a past problem) was very understated when the learner's own code was used. ELM-ART did not highlight the fact that it was the learner's own code and it was also displayed slightly differently (refer to Figures 3.9a and 3.9b below for an example). While the difference in display format was limited, unless the learner was aware of this feature in ELM-ART it could easily go unnoticed without the learner realising that it was their own code. The learner in this experiment commented that seeing their own code used as an example provided a mild feeling of achievement, but it was mainly a novelty factor.

```
(DEFUN area-of-square (side-A side-B)
(* side-A side-B))
```

Figure 3.9a Code as input into ELM-ART

```
(DEFUN AREA-OF-SQUARE (SIDE-A SIDE-B) (* SIDE-A SIDE-B))
```

Figure 3.9b Code as displayed by ELM-ART

3.9 Code examples when using ELM-ART

In terms of the effect on the learner's motivation, the second session had the greatest effect, but in a negative sense. The learner experienced a lot of frustration, confusion and annoyance, which had a direct impact on their motivation level and their desire to continue. It was an unexpected outcome as these emotions were not because of covering the LISP content, but were directly attributable to ELM-ART itself.

ELM-ART uses adaptive navigation which incorporates the use of the traffic light metaphor to highlight which topics the learner has completed, along with the system's inference from the episodic learner model of the learner's ability to undertake a topic. The negative learner emotions were largely related to the navigation through the lessons and subtopics. The learner progressed through ELM-ART by always following the link that it presented at the bottom of the page to take them to the next suggested page. The consequences of this are summarised as follows:

- Sometimes ELM-ART would jump ahead a few subtopics. This did not cause concern until the learner chose to view a past example and was shown an example of a function that had been bypassed by following ELM-ART's suggested navigational path.
- At the end of ELM-ART lesson three, the learner was suddenly taken back to lesson one to cover questions and content that had previously been bypassed. It was this backward jump without any form of pedagogical explanation that had the largest impact on the learner's motivational state.

In addition to demonstrating the impact the system interface can have on the learning experience, this experiment provides a good example of the importance of communicating the relevance of the task to the learner in order to maintain their interest and motivation. When ELM-ART suddenly jumped from the end of lesson three back to lesson one no explanation was given, which negatively impacted on the learner's motivational state. Communicating the relevance of the task is a type of action that can be undertaken across several motivation theories (refer back to Table 3.1, p.75).

This small experiment demonstrated that showing the learner a problem that they had previously solved had the potential to be beneficial not only to aid metacognition, but also to motivate the learner. The learner in the experiment reported that on the occasions when they knew that the past problem that was being presented was their own code, gave them a mild sense of achievement.

3.3.1.2 User Testing

While the experiment with ELM-ART occurred near the beginning of the thesis, this second experiment one occurred two weeks before the start of the study for this thesis. A user testing session was arranged where the Study group version of SQL-Tutor was used by some second-year students at the University of Northumbria. Three out of the six students who were invited to test the system attended the test session. The second-year students were approached as they had all previously taken the same database module on which the study participants were registered, so they could not only test the system, but also do so in the context of the scope of the database module.

Apart from testing the modifications to SQL-Tutor, the aim was also to receive the testers' views of the additional feedback that the system provided, as well as the self-report facility. In general the feedback presented while working on a problem was well received, but there were mixed reviews of the self-report functionality. Two testers did not think that the self-report detracted from working on the problems, with one of the testers stating that it was good to reflect and to be prompted for their self-efficacy level. However, the third tester thought that being prompted to self-report at the end of every problem interfered with their use of the tutor and progress through the problems, especially if they had only viewed the problem and not actually attempted it.

The feedback received from this user testing session directly influenced the self-report frequency in the Study group version of SQL-Tutor ready for the start of the study, as discussed in the next Section – 3.3.2.

3.3.2 Feedback Rules Engine

The motivational and metacognitive feedback based on the learner's past states and experiences was given in two different contexts within SQL-Tutor: in the prompts to the learners to report on their self-efficacy and also in feedback given while the students attempted to solve a problem. This section details the modifications made to incorporate the feedback and self-reports from the technical systems architecture through to the user interface.

In terms of the actual changes to the SQL-Tutor code (developed using Allegro Common Lisp), the author was careful to maintain full record of all changes made. This was achieved by carrying out the following:

- Rather than extend existing classes (e.g. student, problem) when additional properties were required, new classes were created in order to make a distinction between the existing and new code.
- Where new functions were created, they were created inside new code files and annotated to document code history.
- Any changes/insertions made to existing code was annotated with at least the author's initials. For example, ';; (AH)' or ';; (AH 25/05/2011)' or ';; (AH 25/05/2011) Included past problem'.
- New files were created where additional data was required to be imported into the system (e.g. the content rules) or output by the system (e.g. the additional log files). For example, when additional events were required to be recorded by the log files, additional log files were generated instead of changing the existing core log file.

Figure 3.8 (p.73) showed how the modifications fit within the existing system architecture of SQL-Tutor with the introduction of additional log and data files, as well as a 'Feedback Rules Engine' module (referred to as FRE in the remainder of this chapter). Figure 3.10 expands the FRE to outline the main processes that occur to generate either the self-report or the in-problem feedback.

As mentioned, the motivational and metacognitive feedback occurred in two different contexts. Collectively, they occurred at set stages within a session: self-report at the start of a session, during working on a problem, self-report at the end of attempted problems (whether solved or unsolved) and self-report at the end of session. The first process in the FRE is to execute situation rules; these are rules that generate facts relevant to the stage in which they have been executed. For example, during working on a problem the FRE will generate facts based on such things as the help level of the problem, the difficulty rating, the number of attempts and the learner's last self-report of self-efficacy. Whereas at the end of session facts were based on such things as the effort shown in the session, the self-efficacy at the start of the session and the self-efficacy self-reports throughout the session. As SQL-Tutor was developed using Allegro

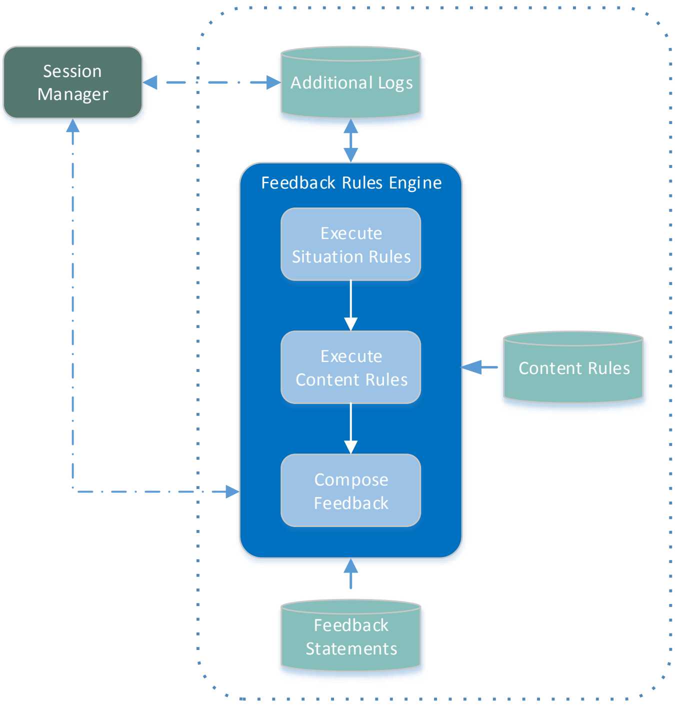


Figure 3.10 The process involved in the 'Feedback Rules Engine'

Common Lisp, a fact is a simple two-item list in the form of *(<fact name> <value>)*. For example, (HELP-LEVEL THREE) indicates that the help level on the previous attempt of a problem was 3/Partial Solution and (CHANGED YES) indicates that the learner has changed their answer since the previous attempt.

Once the FRE has generated the facts relevant to the current stage in a session, the FRE executes a set of content rules, which matches all the facts against the content rules and outputs a set of feedback comment references. These rules were responsible for creating the actual learner feedback. The code to create the content rules was based on code for a forward chaining production rule system that was available on the Web[3]. When initially trying to use the code, it was found that an earlier version of such a production rule system already existed inside SQL-Tutor for processing the constraints. In order to not risk breaking the core functionality of SQL-Tutor, this second production rule system was implemented using a prefix of 'fps_' before any of the code that would have been duplicate. A production rule system was used because the content rules were not hard-coded into SQL-Tutor, but instead were specified in a content rules file and loaded into SQL-Tutor when it was first started on the Web server. Figure 3.11a gives an example of a content rule and Appendix C (p.216) contains a listing of all of the rules.

Any matched and executed content rules output a list of feedback comment references, as can be seen from the content rules in Figure 3.11a. The FRE then formulates the actual feedback by retrieving the text associated with the comment references. Figure 3.11b gives examples of the feedback statements and Appendix D (p.221) contains a listing of all of the feedback statements. The feedback is displayed to the learner either as part of a self-report or as feedback to the submitted attempt at solving a problem. The next four pages give examples of the feedback given in screen prints of the actual Tutor, along with explanations of the user interface, before discussing considerations taken when designing the motivational and metacognitive feedback phrases.

[3] Base production rule system code sourced from http://faculty.hampshire.edu/lspector/fps6.0b2.lisp (20/10/2012).

```
    (ss ((if (logon one) then (add-form (M1)) (add-form (Q1)))
         ...
         (if (logon two) (attempted no) (last-se middle) then
              (add-form (M3)) (add-form (M5)) (add-form (Q4)))
         ...
```

Fig. 3.11a Example content rules

```
(Q1 "Given what you have covered in lectures and workshops so far, how
     confident are you about creating SQL SELECT queries? Please
     indicate below.")
...
(Q4 "Do you still feel the same? Please indicate below.")
...
(M1 "Welcome to SQL-Tutor. Take some time to become familiar with the
     features it offers and the types of problems it contains.")
...
(M3 "You appeared to use your last session purely to become familiar
     with SQL-Tutor and look at some of the problems. Hopefully you
     found that helpful.")
...
(M5 "You ended your last session quite confident about being able to
     answer SQL SELECT problems.")
...
```

Fig. 3.11b Example feedback statements

Figure 3.11 Samples of the feedback rules and statements
(… indicates that the text is an extract and lines are missing)

Figure 3.12 shows an example of the self-report of self-efficacy presented to the learner at the start of session. Figure 3.12a shows the page as it is first displayed to the user and Figure 3.12b shows the page once the user has used the control to report their current self-efficacy levels. The self-reports presented at the end of a problem and at the end of session are formatted in the same way, but shown on dedicated Web pages, as opposed to the start of session self-report shown on the welcome page. The self-report consists of two short paragraphs of text, unless no content rules have been matched, then only the default self-report question is displayed. The first paragraph presents a summary of either the previous session, the problem just attempted or the current session, depending on whether it is shown at the start of a session, end of a problem or end of a session respectively. The second paragraph contains the question for the learner to report on their level of self-efficacy. Below the second paragraph, there are two controls allowing the learner to indicate their self-efficacy level. The available values are using a five-point Likert scale. The first control is a drop-down list box that lists the options of ‘Not set’, ‘1- Low’, ‘2’, ‘3’, ‘4’ and ‘5 – High’, with the first option set as the default in order to clearly distinguish between a learner selection and no selection. The slider control is a graphical representation of the same Likert scale, but it is not continuous in that the slider control has fixed points; clicking on the control will set the slider value to the nearest scale item. Only the first and last items are labelled ('Low' and 'High'), which indicates the orientation of the scale.

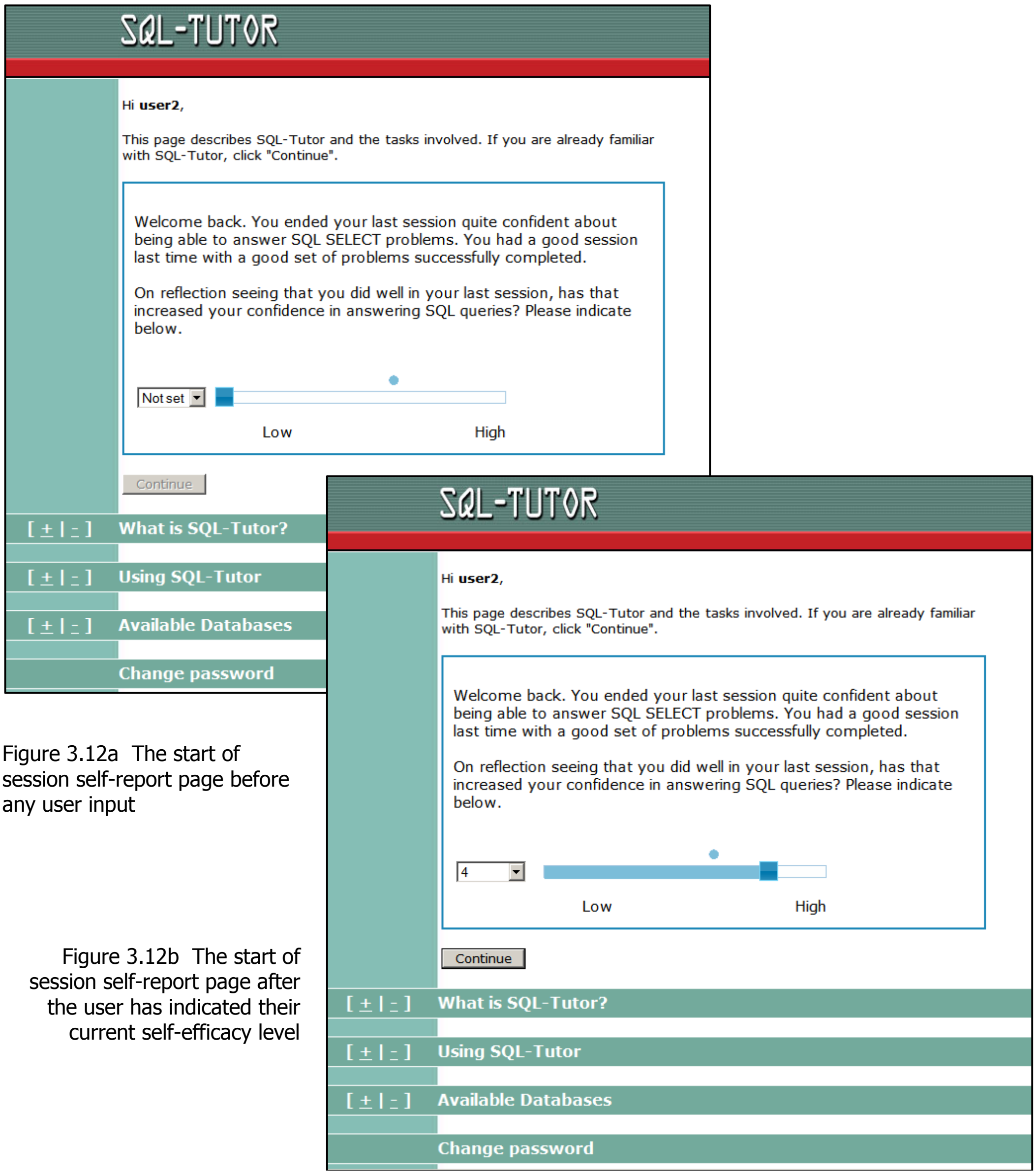


Figure 3.12a The start of session self-report page before any user input

Figure 3.12b The start of session self-report page after the user has indicated their current self-efficacy level

Figure 3.12 The start of session self-report page

For this particular scale, placing labels on every item had the potential to become too crowded and lead to confusion over the meaning of the scale as a whole. The drop-down list box was included alongside the slider control to provide an alternative input method as a failsafe, especially in any instance where the slider control failed to display. The slider control was created using the jQuery JavaScript library[4], which is a code library that has to be explicitly installed on the hosting Web server, as opposed to standard JavaScript functionality. There was a risk that the slider control would fail to be displayed due to such issues as browser security restrictions or browser incompatibility. The 'Continue' button to move on to the next page in SQL-Tutor remains disabled until the learner has chosen a level using either control. This was introduced as a means of preventing the learner from ignoring the self-report. This modification was in line with several of the HHS web design guidelines as listed by Bevan (2005) e.g. "13:3 Minimize User Data Entry", "13:8 Allow Users to See Their Entered Data" and "13:13 Use Familiar Widgets".

The advantages and disadvantages of using self-reports to gain a measure of the learner's self-efficacy level, as discussed in Chapter 2 (p.6), were taken into account when deciding to use self-reports. As mentioned in Chapter 2, the risk of using self-reports is minimised when used in conjunction with other forms of data collection. While the self-report is the sole collection method for the learner's self-efficacy, it was used in conjunction with data gathered from the learner's interaction with SQL-Tutor (e.g. number of attempted problems and length of session). Self-reports were used in this study as a direct means of gaining self-efficacy states of the learner and allowing data collection to occur in-situ, rather than after the fact when there is the risk that the learner's perception and response would differ.

The aim of the wording used in the questions and feedback was to not be read as condescending to the learner. Taking into consideration the *'plausibility problem'* discussed in Section 2.3 (p.22), the feedback also aimed not to promote SQL-Tutor by giving it any say/persona, so the second person ("you", "your" and "yours") was used in the feedback, not the first person ("I" and "we"). The wording used for the self-report questions did not use the word "*self-efficacy*" as the author felt that most of the study participants would not fully understand what it meant. Instead the word "*confidence*" was used even though they are not exactly the same things, as stated by

[4] jQuery JavaScript library is obtainable from http://jquery.com (30/08/2014)

Bandura (1997) "the construct of self-efficacy differs from the colloquial term ‘confidence.’ Confidence is a nondescript term that refers to strength of belief but does not necessarily specify what the certainty is about”. In order to provide a focus for what the word was related to, the questions include words to give the word ‘confidence’ context. For example, “Given what you have covered in lectures and workshops so far, how confident are you about creating SQL SELECT queries?” and also refer to the text in Figure 3.12 (p.81) for another example. The wording in the feedback statements and self-report questions were constructed by bearing in mind the review of work in Section 2.3 (p.18), e.g. the initial intent and the actual impact of feedback (refer to Table 2.2, p.19), and based on the skill and judgement of the author with over five years University lecturer experience, which also included one-on-one tutoring. They were also fine-tuned with the aid of a former University lecturer with over thirty years experience. Table 3.2 outlines guidelines that the author developed to aid the creation of the motivational and metacognitive feedback phrases used in the study version of SQL-Tutor. The table sets out the aim of the feedback for each of the nine learner states relating learner self-efficacy judgements to performance. The basis of the guidelines was in line with the view expressed in Schunk et al. (2007); “in general, it is most adaptive to have self-efficacy that slightly exceeds actual skills at any given time” in order to maintain learner interest and motivation. At the same time the feedback phrases used aimed to indirectly aid the learner to manage their expectations and self-efficacy in the advice that was provided.

The idea of whether to have different variations of the same feedback statement was considered in order to make sure that the learner did not receive exactly the same feedback wording twice in a row. This was thought to potentially be a good idea, but it was decided that this would be an enhancement which would fit into any future work in this area (refer to Section 6.5 (p.183) for potential future work).

Table 3.2 Guidelines developed to aid metacognitive and motivational feedback phrases

		Performance [P]		
		Doing Well	**Middling**	**Not doing well**
Self-efficacy [SE]	**Optimistic**	=P =/⤓SE - Ask learner how far do they think they are through the learning outcomes and are they ready for next stage. - The aim is for the learner to be thoughtful about their own learning. Reflection.	⤒P ⤓SE - Suggest need to do more tasks at the same level	⤒P ⤓SE - Suggest less difficult tasks. - Remind learner of last successful (related) task.
	Realistic	=P =⤒SE - Relate to and promote past & current performance. - Suggest task difficulty increase.	⤒P =⤒SE - Suggest stay at same task difficulty (in fact this may mean working on easier tasks depending on how SQL-Tutor re-adjusts the level for recently completed tasks). - Point to steady progress and remind learner of successful, recent tasks.	⤒P =⤒SE - Suggest less difficult tasks. - Remind learner of last successful (related) task.
	Pessimistic	=P ⤒SE - Relate to and promote past & current performance. - Highlight changes in SQL-Tutor task difficulty ratings on tasks learner has completed. - Potentially maintain task difficulty level to reduce pessimism	⤒P ⤒SE - Relate to and promote past & current performance.	⤒P ⤒SE - Reminder of last successful task completed. - Suggest that "we" work on less difficult task to establish strong understanding. - Point to further reading (based on selected clause or clause where most errors exist?)

Key: = maintain/keep the same ⤓ Decrease ⤒ Increase

If the current stage of the session is working on a problem, then the FRE would also try to find any related past problems the learner has previously solved. This was one of the elements fundamental to the aim of this thesis, i.e. to use experiences (and past states) of a learner to aid their metacognition. Figure 3.13a shows an example of when a related past problem has been found (any motivational and metacognitive feedback was also displayed in the same location; in this example there were no matching content rules to produce such feedback).

The feedback message was displayed in the text field that was newly added to the main problem page for this study. It was located in the existing whitespace between the bottom of the problem area and the start of the database schema area on the page. The message explicitly informed the learner what match had been found and invited them to view the past problem by clicking on the 'View Past Problem' to the right of the text field. Figure 3.13b shows the past problem displayed in a separate window. In addition to matching problems on a problem level, they were also matched at a clause level. During the initial familiarisation sessions with SQL-Tutor, it was noted that the feedback provided by the base version of SQL-Tutor sometimes mentioned a particular clause in the SELECT query that was in error. The decision was made to use this and to also match any past problems based on a specific clause (e.g. FROM or WHERE clause). Matches for problems at a problem level were initiated after the second, fourth and eighth attempts at a problem. If one problem was matched after the second attempt, a search was conducted to find a second match after the fourth attempt. In terms of matching on a specific clause, this was done on any attempt after the second where the feedback from the base SQL-Tutor included mention of a particular clause. Section 3.3.2.1 (p.88) provides details on how the problems and clauses were related.

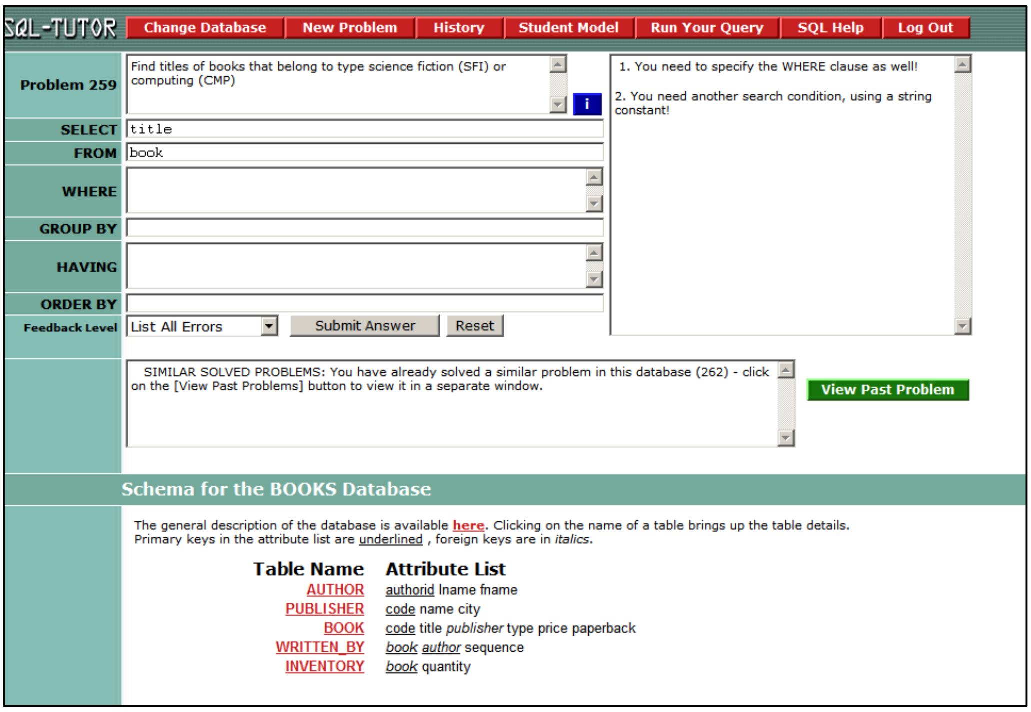


Fig. 3.13a Feedback showing a related past problem has been found

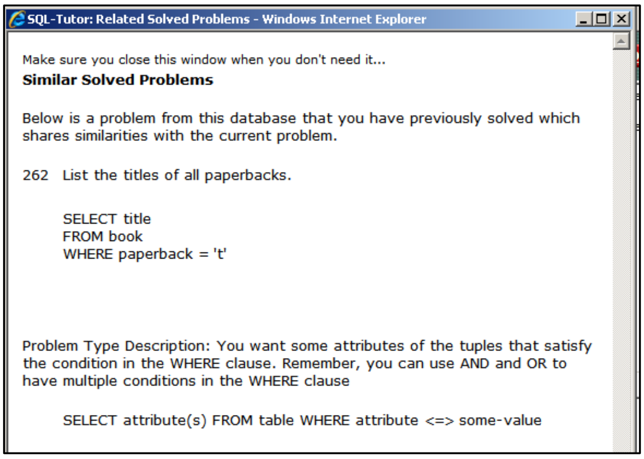


Fig. 3.13b Shows the separate window displaying the past related problem

Figure 3.13 Shows an example of the additional feedback, including an example of a past related problem

The FRE, along with the 'Session Manager' from the base SQL-Tutor version, recorded any data regarding learner interactions with the system to a set of three additional log files. There were also the log and learner model files that the base system also produced, but as mentioned earlier, the base files were left untouched. Out of the three additional log files, one was a pure data recording/output only log file, while the other two files were actively used by SQL-Tutor by importing their contents at the point where a learner logged onto the system. Figure 3.14 shows examples of all three additional log files.

```
21:59:34 15/11/2012; Logged in.
21:59:34 15/11/2012; Loaded feedback data.
21:59:34 15/11/2012; Session count: 6
21:59:34 15/11/2012; content-facts (initial) type: ss: ((USER USER))
21:59:34 15/11/2012; Just about to calculate effort
21:59:34 15/11/2012; Just calculated effort
21:59:34 15/11/2012; content-facts (current) type: ss : ((EFFORT POOR)
                                  (PERFORM LOW)
                                  (SESSION SHORT)
                                  (LAST-SE MIDDLE)
                                  (ATTEMPTED YES)
                                  (LOGON XTH)
                                  (USER USER))
21:59:34 15/11/2012; Formulating feedback for type: ss
21:59:34 15/11/2012; Loaded past problems  data.
21:59:37 15/11/2012; Database: books
21:59:40 15/11/2012; Student chose problem: 259   Rating: 1
21:59:59 15/11/2012; Help-level: 1
21:59:59 15/11/2012; processed submission
21:59:59 15/11/2012; just about to process rules
21:59:59 15/11/2012; content-facts (initial) type: ip: ((USER USER))
21:59:59 15/11/2012; content-facts (current) type: ip : ((HELP-LEVEL NOTPARTFULL)
                                  (STATE UNSOLVED)
                                  (CHANGED NO)
                                  (ENTERED YES)
                                  (VISITS XTH)
                                  (ATTEMPTS ONE)
                                  (USER USER))
21:59:59 15/11/2012; Formulating feedback for type: ip
21:59:59 15/11/2012; There are no feedback facts - using vanilla feedback.
21:59:59 15/11/2012; now going check on past problems as this problem is UNSOLVED
```

Fig. 3.14a Log file

```
(SQL-TUTOR::SESSION-COUNT 7)
(SQL-TUTOR::SR-LAST 4)
(SQL-TUTOR::PROB-ATTEMPT 0)
(SQL-TUTOR::PROB-VISIT 1)
(SQL-TUTOR::PROB-SOLVED 0)
(SQL-TUTOR::PERFORMANCE 0)
(SQL-TUTOR::EFFORT 0)
(SQL-TUTOR::SS-EFFICACY 4)
(SQL-TUTOR::SE-EFFICACY 4)
(SQL-TUTOR::LASTPROB-RATING 1)
(SQL-TUTOR::PS-DURATION 129312))
```

Fig. 3.14b Feedback log file

```
"pnum-100"
"library"
"problem text"
"SOLVED"
"1"
0
0
((1 1 4 2 5 4 3 1 1 0 1))
((13 13 13 6 6 6 6 6 6 6 0))
((1 1 1 1 1 1 1 1 1 1 1))
((1 0 0 1 0 0 0 0 0 0 1))
"address"
"branch"
"branchid='CEN'"
""
""
""
"pnum-96"
"library"
"problem text"
"SOLVED"
```

Fig. 3.14c Problems log file

Figure 3.14 Additional log files

As can be seen in Figure 3.14a, the pure output only log file recorded all of the interactions between SQL-Tutor and the learner. Some of the data recorded overlapped with the data recorded by the base SQL-Tutor log file, but such details were included to give context to the other data recorded. This log file also recorded the facts that were generated by the situation rules in the FRE. These facts could be used when analysing the log files to reconstruct the feedback received by the Study group participants. One difference between this log file and the base SQL-Tutor version log file was the number produced per user. In the base version one log file was created per user and the data for every session is appended to the end of the file. For the additional log file, one file was created per user per logon with the filename containing a date/time stamp. The filename format was '*<username>-<yyyymmddhhmmss>.log*' (*yyyymmdd* refers to the current year, month and date, and *hhmmss* refers to the current time in hours, minutes and seconds), so for example, the name of the log file in Figure 3.14a would be 'user2-20121115215934.log'.

Figure 3.14b shows an example of the feedback log file that was created once per user, then updated in every session. The format of the filename was '*<username>-feedback.log'* (e.g. 'user2-feedback.log'). This file contained the key facts that were used by the content rules to generate the start of session feedback text and self-report question, as well as in the rules for generating the end of session feedback and self-reports (refer to Appendix C (p.216) for a full list of the content rules).

Figure 3.14c shows an example of the problems log file. Again, it was created once per user, then updated after the learner had finished working on a problem. The format of the filename was '*<username>-problems.log'* (e.g. 'user2-problems.log'). This file was used to record key data about every problem attempted or even just visited by the learner. It included such data as the problem number, database name, status (solved or unsolved), difficulty rating, help levels selected across all attempts, number of errors (if any) at every attempt and also the text entered by the learner in each clause for the last attempt. Apart from post-study data analysis of these files, the FRE used the data to maintain a list of problems previously solved by the learner and then used this to determine whether any of the problems could help the learner with the current problem. If a match was found for a past problem, then the text that the learner entered for the answer is then displayed in the separate window shown in Figure 3.13b.

3.3.2.1 Relating Problems and Clauses

In order to relate problems to other problems for use in the metacognitive feedback given while working on a problem, the idea of relating problems by the constraints used in SQL-Tutor could be a feasible method. This was discussed during a presentation the author gave in 2010 to the ICTG research group, University of Canterbury, New Zealand, and via this discussion it soon became apparent that using the constraints would be too finely grained for the purposes required. Instead work previously carried out by Mathews and Mitrovic (2007) with the use of *problem templates* in SQL-Tutor was used as a starting point. A problem template is a method of grouping similar problems together and thirty-eight groups were created. Table 3.3 shows two examples of problem templates. Each template contains a unique number as a key/identifier (e.g. 1 and 3), a list of related problems, a general statement of the template and finally a feedback statement for the template.

Table 3.3 Examples of problem templates.
Reproduction of Table 1, p2 in Mathews and Mitrovic (2007)

```
(1 (1 26 59 132 135 164 158 151 199 235)
("SELECT * FROM table" "Retrieve all attributes of one table"))
```

```
(3 (152 237 255 260)
("SELECT DISTINCT attribute(s) FROM table"
"You want the details without duplicates (DISTINCT) of the attribute(s) of a table"))
```

These problem templates were used as a starting point for relating problems at a problem level for use in this study. The templates and grouping of related problems were analysed, with some small adjustments made. Two of the templates did not have any problems assigned and similarly there were some unassigned problems. From the original thirty-eight templates, this study used thirty-two, by adjusting the focus of some of the templates in order to accommodate problems from templates that only contained one problem. For the templates used, the same general and feedback statements for the templates were used, with only a few small changes. In SQL-Tutor, the related problems were stored in a global list variable using the structure shown in Table 3.4.

Table 3.4 Examples of related problems used in this study

```
(("SELECT * FROM table" "Retrieve all attributes of one table"
(168 1 151 235 99 26 68 59 199 132 135))
```

```
("SELECT DISTINCT attribute(s) FROM table"
"You want the details without duplicates (DISTINCT) of the attribute(s) of a table"
(260 237 255 83 91 101 152))
```

In addition to relating problems to other problems at problem level, relating problems at a clause level was also included in study version of SQL-Tutor. Every problem was analysed across each of the clauses (e.g. SELECT, FROM, WHERE, ORDER BY). For each clause, a set of similar groups were created, with the problems assigned to the appropriate groups. Table 3.5 shows examples of such clause level groupings. For the clause level groupings the grouping statement is also used as a feedback statement.

Table 3.5 Examples of problems related at a clause level as used in this study

SELECT clause	(("Select all attributes" (258 168 1 8 145 151 235 243 246 99 26 68 59 159 199 204 211 214 221 225 132 133 135))
WHERE clause	("Uses IN / NOT IN (may also include a subquery)" (10 36 200 212 213 217 232 194 13 190))

The SELECT clause contained eight groups, the FROM clause had five, the WHERE clause had eight, and the GROUP BY, HAVING and ORDER BY clauses all had two groups each.

3.3.2.2 Settings Rules and Parameters

The functionality to incorporate self-reports, and motivational and metacognitive feedback based on past learner states and experiences involved setting numerous levels (e.g. the number of attempts a learner should make before the feedback prompts them to study the partial solution as a means of help). The decision on such levels were informed by referring to results from a previous SQL-Tutor study, direct feedback from test users and reasonable assessments based on experience.

A previous study looked at the effect of using positive and negative feedback inside SQL-Tutor (Barrow, 2008; Barrow et al., 2008). Table 3.6 shows the results as shown in Barrow (2008). As an example of how these results influenced some levels within this study version of SQL-Tutor, the rule that determined whether the additional feedback during solving a problem should suggest to the learner to study the partial solution (achieved by the learner selecting the required help level, refer to Section 3.1.4 (p.64) for further discussion on help levels) was set to six attempts. This was set to six after looking at the average attempts per solved problem (last line of Table 3.6) and rounding the number up to six attempts. For the scope of the current study, a basic calculation of the total attempts divided by problems attempted was used to determine the effort of a learner. The associated decision of what equalled a poor or good effort was influenced by looking at the attempts per problem result (second line from the bottom of Table 3.6). For this study, an effort less than three was determined as ‘poor’, this was nearly one standard deviation away from the mean reported in Table 3.6.

Table 3.6 Summary analysis of student interaction.
Reproduction of Table 6.3, p99 in Barrow (2008)

	Control	**Experimental**	***p***
Number of Participants	23	18	ns
Pretest mean (sd)	1.7 (0.8)	2.1 (1.3)	ns
Constraints Learned	10.0 (6.1)	9.3 (6.8)	ns
Time (min)	193.8 (198.7)	92.3 (44.7)	0.012
Problems Solved	25 (24)	22 (15)	ns
Total Problem Attempts	119 (99)	98 (66)	ns
Problems Attempted	28 (25)	26 (15)	ns
Lab-test mean (%)	57.0 (26.5)	59.3 (24.3)	ns
Average Time per Solved Problem	9.8 (7.9)	5.8 (4.8)	0.024
Average Time per Attempted Problem	7.5 (4.5)	4.1 (2.0)	0.002
Attempts per Problem	4.5 (1.6)	3.7 (0.9)	ns
Attempts per Solved Problem	5.5 (2.9)	5.2 (3.4)	ns

Section 3.3.1.2 (p.77) detailed the user testing that was carried out on the study version of SQL-Tutor. In the version of the tutor that was used for that testing, the learner was prompted to self-report their self-efficacy levels every time they finished working on a problem by leaving the main problem Web page. While a couple of the testers were not concerned with this frequency of self-report, another tester found it intrusive, especially if they had just visited the problem and not actually made any attempts. As a result of that testing session and the testers feedback, the frequency of self-reports when leaving a problem was modified so that a learner was only prompted to report on their self-efficacy if they had actually attempted a problem (regardless of whether the problem was solved or not). Also introduced was the test for learners who just visited a problem. If a learner had just visited a problem they were not prompted to self-report unless the number of visited problems in the current session was a multiple of three. So the learner would be prompted to self-report after visiting three, six, and so on, problems (the visits did not have to be consecutive; the learner may have visited a problem, solved one problem, then visited two problems in a row). Setting this value to a multiple of three is an example of a level set by reasonable assessment.

3.4 Chapter Summary

This chapter has detailed the primary features that were core to all versions of SQL-Tutor. It then detailed what modifications were made for this study, along with how and why such modifications were made. The modifications made fell into two categories: the modifications that were available to both the Control and Study groups, and the modifications made to just the version of SQL-Tutor used by the Study group, such modifications were core to the focus of this thesis. This chapter also discussed the small "Wizard of Oz" inspired experiment and user testing, which were conducted during the course of this thesis and that influenced the modifications to the given system. The next chapter details the study design and implementation that used the two versions of SQL-Tutor detailed here.

4 Study Design & Implementation

This chapter describes the main study that was undertaken with the aim of evaluating the modified SQL-Tutor, as described in the previous chapter, in relation to the research questions. It was the original intent of this thesis to conduct a pilot study prior to the main study in order to help refine the modifications made to SQL-Tutor and the content of the additional feedback. Unfortunately due to circumstances (e.g. time), it was not possible to conduct the pilot study. The main study took place over a period of three and a half months from late November 2012 to the beginning of March 2013, with the process of enlisting volunteers starting in October 2012. Two groups were used in the study; the control group used a base version of SQL-Tutor and the study group used the modified version of SQL-Tutor.

Formal ethical approval was obtained from both of the Universities involved; the University of Sussex, where this thesis is registered and the University of Northumbria where the study participants are registered. Refer to Appendix B (p.201) for an overview of the ethical approval process, the documents submitted to support the application (including the volunteer consent form, B4 p.213) and the ethical approval certificate (B5a & B5b, p.214-215).

This chapter details the method used in the study by dividing into subsections covering participants, procedure and data collection.

4.1 Participants

The study participants were students from a Relational Databases module at the University of Northumbria who volunteered to take part in the study. The volunteers received no academic credits or monetary reward for participating in the study. However a prize draw with the opportunity to win one of two £25 Amazon or iTunes vouchers was offered after the submission of a post-study questionnaire by participants.

The module is a core, first year undergraduate, 20-credit module that is delivered throughout the year (across both semesters) to students across seven degree courses. The summative assessment consisted of three milestone tests spread across the semesters and one examination at the end of semester two. The assessment results were used in this study (refer to Sections 4.1.1 (p.95) and 4.3.1 (p.100) for further details). The taught element of the module was delivered by a one hour lecture and a one hour computer lab-based workshop each week, along with an additional five hours of independent study each week. The module is taught across two semesters with each comprising of twelve teaching weeks and an examination period of three weeks. Semester one, during which this study ran was taught consecutively from Monday, 24th September 2012 to Friday, 14th December 2012. The first nine teaching weeks of semester two ran consecutively from Monday, 21st January 2013 to Friday, 22nd March 2013. The remaining three teaching weeks started on Monday, 15th April 2013 after a three week spring/Easter break. Appendix A (p.200) shows the study timeline, especially in how it cohered with the academic year.

The seven degree courses have differing levels of technical content in terms of computer technology. This meant that when dividing volunteers into the two groups required for the study (control and study groups) there was a risk that the groups would be unbalanced in terms of the volunteers' technical ability. For example, the control group may have contained mostly volunteers from degree courses with high technical content (i.e. games programming) and the study group may have contained the opposite (e.g. softer skills such as systems analysis). In light of this risk and the fact that the author wanted to compare study data on a like-for-like basis across participants, a simple rating system was created. In discussion with the module tutor (the lecturer who has overall management responsibility for the module), the courses were rated by technical content; 1 = low, 2 = medium and 3 = high. Table 4.1 outlines the courses and the assigned ratings. The ratings are used to reflect the level of technical content of a course in relation to the other courses, as opposed to rating the course in isolation. For the purposes of this study technical content refers to modules that cover such topics as computer programming, networking and maths, which are distinguishable from less technical topics such as project management and systems analysis.

Table 4.1 Degree courses with technical content rating

Course Name & Code		Rating
BIS	Business Information Systems	1
ITMB	IT Management for Business	2
CFO	Computer Forensics	2
CS	Computer Science	2
WDD	Web Design & Development	3
ETH	Ethical Hacking	3
GAP	Games Programming	3

4.1.1 Volunteering Process and Number of Participants

Prior to the start of the study, the author gave a short presentation at the start of the module lecture in week five, semester one. This week was chosen as SQL SELECT statements had been taught since week three, meaning they had been covered for two weeks, allowing the students to start to form their personal skills and attitudes towards the subject. It also gave the students an opportunity to make an informed decision as to whether they thought using such a tool as SQL-Tutor by participating in the study would be beneficial to them. The presentation introduced the author, outlined the research degree focus, introduced SQL-Tutor, how the students could use the tool in their studies and the start date for the study.

The level of interest for volunteers was gauged by using a one-question survey on the Blackboard [5] site for the module. The question was "Do you want to volunteer for the study and use SQL-Tutor?" with "yes/no" option buttons presented. From the two-hundred and twenty-three students registered on the seven courses, there was a thirty-six percent take-up with eighty students responding "yes" to the survey.

[5] The Blackboard Learning System (known as 'Blackboard') is the virtual learning environment and module management tool that forms the basis of the University of Northumbria's "e-Learning Portal".

The division of volunteers into two groups of forty (known as 'Study' and 'Control' groups) was organised using the course rating in Table 4.1 and the volunteers' marks from the first summative assessment on the module: Milestone 1, see Table 4.3. The marks from Milestone 1 were used as a starting indicator of the volunteers' SQL SELECT statement knowledge. This meant that as far as reasonably possible two balanced groups could be formed with an even spread of both marks and course representation.

When the study started in week nine of semester one, the number of students who signed consent forms and collected SQL-Tutor user IDs was down to fifty-nine. Refer to Appendix B4 (p.213) for a sample of the consent forms used. The shift in volunteer numbers included eight students who had not volunteered via the survey and the removal of one student who was no longer registered on a course. The initial group assignment was created using the data for the eighty survey respondents. In order to maintain balanced groups the new volunteers were assigned a group based on their course rating and Milestone 1 marks, but taking into consideration the balance of the existing group assignments. Unfortunately the slight shift that was caused by students who had originally volunteered, but then did not start the study, could not completely be compensated for in terms of balancing the groups. Table 4.2 shows the distribution of participants in terms of course ratings and Milestone 1 marks. The final group sizes (known as the participants) used in the study were thirty-one (six female and twenty-five male) in the Control group and twenty-eight (four female and twenty-four male) in the Study group. The Milestone 1 marks for the two study groups were normally distributed as assessed Shapiro-Wilk test ($p > .05$), with $p = .089$ for the Control group and $p = .073$ for the Study group. On average, participants in the Study group achieved the nearly the same marks in Milestone 1 (M = 72.68, SE = 3.731), as those in the Control group (M = 72.69, SE = 3.208). The difference of 0.01 was not significant, as confirmed by an independent t-test, $t(57) = .003$, $p = .998$.

Table 4.2 Distribution of participants including course rating and Milestone 1 marks
(Ratings: 1= low, 2= medium, 3 = high technical course content)

Milestone 1	Control Group				Study Group			
	Group	Ratings			Group	Ratings		
	Level	1	2	3	Level	1	2	3
Participants	**31**	4	17	10	**28**	6	15	7
Mean	**72.69**	74.50	72.88	71.66	**72.68**	84.96	70.71	66.39
Median	**76.47**	76.94	76.47	76.47	**77.41**	83.35	77.41	65.65
Std Dev	**17.86**	14.26	19.90	17.00	**19.74**	8.99	22.19	18.21
Range	**67**	34	67	56	**67**	23	67	56
Min	**33**	55	33	44	**33**	77	33	44
Max	**100**	89	100	100	**100**	100	100	100

4.2 Procedure

Once the initial list of volunteers had been allocated to the study or control groups, the study started during week nine of semester one. The volunteers were invited to attend one of four scheduled thirty-minute introduction sessions. The aim of the sessions was to demonstrate SQL-Tutor, get study participation consent forms signed (a generic University of Northumbria consent form was used), issue the SQL-Tutor user IDs and provide guidance to the level of use that was expected for the study. The author originally thought that the sessions should be specific for each study group, especially in the case of the study group that would be using the extra functionality that had been incorporated into SQL-Tutor. However, the author realised that even if someone was invited to attend a particular session, there would likely be instances where people decided to attend a different session. With this in mind all of the sessions were the same, with the base version of SQL-Tutor used for the demonstration and everyone was told that if they were prompted by the system for input, then they should follow the instructions on screen. In addition to the information provided in the introduction sessions, each participant was sent via email a small user guide for SQL-Tutor and a study information sheet presented in the form of questions and answers.

The use of SQL-Tutor could not be incorporated into the standard delivery of the module because the module itself was already well established, with a clear schedule and dedicated teaching material. This meant that the usage of SQL-Tutor not only had to be on a voluntary basis, but also in addition to the module delivery. In the introduction sessions the volunteers were told that the time they spent using SQL-Tutor could count towards the five hours per week of independent study that is expected by the module. The following guidance was given by the author in terms of expected usage:

> *"The study does not have any strict times for using SQL-Tutor. The day & time is fully flexible to work around your schedule, you can practice SQL SELECT queries with SQL-Tutor as often as you want. However there are general guidelines; use SQL-Tutor at least weekly (although over the duration of the study there will be exceptions to this, i.e. Christmas & New Year). Each use (session) of SQL-Tutor should be long enough to actually use it to gain some benefit from it (i.e. logging on for only 10 minutes and not attempting any problems, is not beneficial to you for learning and practicing queries)."*

A study using Andes, the physics ITS used to complete homework, was similar in that they did not have any fixed class sessions for using the tutor, instead the learners chose when to use it (Vanlehn et al, 2005). However, the homework that was completed using Andes had to be handed in to the teachers/instructors. This means that the participants in the Andes studied had a stronger level of commitment and incentive to use the tutor in their own time, as opposed to a situation where the use of a tutoring system was optional.

During the introduction sessions eight new volunteers attended who had not previously expressed an interest in the study. They were asked to complete consent forms, with SQL-Tutor user IDs assigned after the sessions and sent to them via email by the module tutor. The students that had already volunteered, but did not attend an introduction session were contacted via email and given until the end of week eleven, semester one, to complete a consent form so that they could receive their user ID (no user IDs were supplied until a consent form had been signed). At the beginning of week twelve, semester one, anyone who had been allocated to a group, but still had not collected their user ID was removed from the study and placed in the non-participant list. This led to the participant groups outlined in the previous section.

In week eleven, semester one, after receiving support request emails from a few participants, an email containing further guidance to the use of SQL-Tutor was issued to all participants. As described in the previous chapter, all of the problems available in the tutor had been analysed, including the removal of any problems that contained content not covered by the module, but from the support requests it became apparent that the module taught some short-cut techniques in the SQL SELECT statement syntax. For example, when using an *equijoin* where the fields to join are named the same in both the tables the full answer would be ... FROM tblA INNER JOIN tblb ON tblA.field1 = tblB.field1. However, the module taught a short-cut; ... FROM tbla INNER JOIN tblB USING (field1). The use of such short cuts had not been considered and SQL-Tutor also did not recognise them as correct answers to problems. The participants were given the following guidance in order to prevent confusion not only in the use of SQL-Tutor, but also in what was expected in the summative assessments of the module.

> *"Any electronic tutoring systems are aimed at a particular level of learning and SQL-Tutor is no exception. The tutor is aimed at the initial learning of the core concepts involved in SQL SELECT queries. This means that it will generally expect answers written in full without any of the short-cuts that may be available.*
>
> *NOTE: 1. While the short-cut is valid, SQL-Tutor will expect the full version as it is aimed at the core concepts and not the short-cuts.*
> *2. Either the full or short-cut versions are valid in the module milestones."*

On the 21st December 2012 two code changes were made live (available to use by the study participants) and an email outlining the changes was sent to all participants by the author. One change was essential as it fixed a bug that some participants had experienced and reported. The bug was concerned with reading and writing the new log files as sometimes additional blank lines were output to the files, which caused problems when SQL-Tutor read the log files the next time the participant logged in. The second change was not essential in terms of a bug fix in code, but it was an important user interface improvement concerning the participants' ability to log out of SQL-Tutor (refer to the discussion of web design guidelines at the beginning of Section 3.2.2, p.67). In both the base and study versions of SQL-Tutor the system only presented a log out button on one web page only. Due to the additional log files and self-report interface presented in the study version of the tutor, the participant was required to log out of the system properly (using a log out button) so that the additional log files would be correctly updated and an end of session self-report web page would be presented. With only one web page containing a log out button it meant that some participants were not logging out of the tutor correctly, but instead were ending the session by closing the actual web browser window. The second change implemented a log out button on an additional three web pages in order to encourage the participant to log out using the SQL-Tutor button.

Due to the fact that the study had no fixed sessions for the participants to attend and use SQL-Tutor, reminder emails were sent to all participants on three occasions; week twelve in semester one, week one in semester two and in week five in semester two. The emails were general reminders to use SQL-Tutor, along with a reminder of the end of study date.

During the course of the study there was only one instance of five days where SQL-Tutor was not available. The reason why the system was down was not known and occurred in January 2013.

A post-study questionnaire was issued to all participants in week seven, semester two. The questionnaire (refer to Section 4.3.3 (p.103) for further details) was presented on-line, so an email containing the URL and instructions was sent to the participants. Originally a two week deadline was given for the completion of the questionnaire, but this was finally extended to six weeks (including reminder emails being sent) in order to try and maximise the number of responses.

4.3 Data Collection

Three primary sources were used for data collection in this study: module summative assessments, SQL-Tutor log files and a post-study questionnaire.

4.3.1 Module Summative Assessments

The course module had four individual summative assessments; three were in-class tests referred to as 'Milestones' and the final assessment was an examination. All assessments were conducted on-line using the test facility inside Blackboard and consisted of multiple choice, true/false and fill-in the blank question types. Table 4.3 shows the schedule and percentage weightings of each assessment.

Table 4.3 Summative assessment schedule and percentage weightings

Assessment	Schedule	Percentage of Module
Milestone 1	Week 6, Semester 1	10%
Milestone 2	Week 12, Semester 1	20%
Milestone 3	Week 6, Semester 2	20%
Final Examination	Week 12, Semester 2	50%

During the initial discussions with the module tutor regarding the study requirements and design, it was agreed that the summative assessments could be used as pre- and post-tests of the study, as opposed to creating dedicated tests, as the study was embedded within an existing university course with limitations on what extra testing of the students was possible. The author felt that using existing tests, while not specifically tailored to the study, would still allow for measureable data to be collected. The marks from Milestone 1, as mentioned in Section 4.1 (p.93), were used to help create two statistically balanced groups for the study. The remaining assessment marks were used for progress and post-test analysis (Chapter 5 (p.105) discusses the study findings in detail). The schedule of the summative assessments also aided the decision of the study start and end dates; the study started after Milestone 1 and ended after Milestone 3. Only the actual dates were influenced by the assessment schedule. It did not determine the aim of the study, which was for SQL-Tutor to be used over a period of time that would allow for the metacognitive feedback aspects of the tutor to be realistic. For example, the study wanted to be able to remind the participants of previous problems encountered a week or month ago, not just a few hours or days ago.

One disadvantage of using the existing module assessments was that not all questions in the Milestone tests were relevant to the study. For example, a question covering the SQL CREATE TABLE statement was outside of the scope of problems presented in SQL-Tutor. SQL-Tutor purely covers SQL SELECT queries. Questions that did not explicitly test knowledge of elements of SELECT queries were excluded. Table 4.4 outlines the full marks (refer to Table 4.3 (p.101) for the percentage weightings of each assessment) and question count per assessment, along with the adjusted marks and question count used by this study. Appendix E (p.225) contains all of the assessment questions as well as indicating which questions were excluded from the study.

Table 4.4 The complete and adjusted summative assessment marks & questions

Assessment	Complete Assessment		Adjusted Assessment	
	Marks	Question Count	Marks	Question Count
Milestone 1	10.50	6	8.50	4
Milestone 2	26.00	13	25.00	12
Milestone 3	19.00	10	9.00	4
Final Examination	40.00	13	4.00	4

4.3.2 SQL-Tutor Log Files

The following log files produced by SQL-Tutor were used to collect data about the study participants and their actions.

- Log files produced by the base version of SQL-Tutor:
    - The activity log detailing the actions of the user (i.e. log on date/time, problem number selected, answer submitted, and so on).
    - The text file containing student model data.
- The additional log files that are produced by the study version of SQL-Tutor:
    - An activity log containing additional activity that is specific to the study version of the tutor (i.e. the participants' self-report, feedback given, and so on).
    - A problem log that contains data about each problem the participant has ever attempted. Such data includes, the problem number, number of attempts, the current status of the problem (has it been solved) and the last answer input for the problem by the participant.

- The feedback log contains data used to determine the motivational feedback to give to the participant.

Refer to Section 3.3.2 (p.78) for details of the additional log files that were used in the study version of SQL-Tutor.

4.3.3 Post-Study Questionnaire

A post-study questionnaire was designed to gauge the participants' opinions of using SQL-Tutor to improve their SQL SELECT query skills, as well as the different types of feedback provided, the accuracy of the participants' self-efficacy judgements and any motivation gained from using SQL-Tutor.

The questionnaire was created on-line using Toluna QuickSurveys[6] with the aim that an on-line survey would increase the number of completed questionnaires. The questionnaire (Appendix F (p.235) contains screen prints of the questionnaire from the Toluna QuickSurveys website) consisted of four web pages. The first page contained a welcome statement, instructions and a prompt for the participants' SQL-Tutor user ID. The second page contained closed questions presented as a series of nine statements using a five-point Likert scale (strongly disagree = 1, disagree = 2, neutral = 3, agree = 4, strongly agree = 5). With the inclusion of the 'neutral' item the questions are balanced, in that it is not a forced choice scale where the participants must either agree or disagree. The nine statements were related to the domain, the learning process, metacognition, motivation and general use of the tutor. The third page contained open questions which were used to gauge the participants' opinions of using SQL-Tutor and the feedback comments that it provided. The final page was used to display a thank you message and ascertain whether the participant wanted to be entered into a draw for a chance to win either Amazon or iTunes vouchers.

---

[6] Toluna QuickSurveys are available at http://www.quicksurveys.com (03/05/2014)

4.4 Chapter Summary

This chapter has described the study used for this thesis, detailing the methods for gaining and organising study participants, the procedure and the sources of data collection. The next chapter goes on to analyse the data extracted from the data sources mentioned in this chapter and discusses the results to address the research questions presented in Chapter 1 (p.1).

5 Analysis of Study Results

This chapter presents the analysis and discussion of the results gained from the different data sources outlined in the previous chapter. Section 5.1 below explains the final Control and Study group sizes, and how their sizes changed from the start of the study. Section 5.2 (p.106) presents the analysis of the pre- and post- process results which looks at what the overall effect was. Section 5.3 (p.135) presents the analysis of log files that were generated by SQL-Tutor and shows a process view to determine if there were any differences to the way participants responded at the time of using SQL-Tutor. The analysis of the results are grouped by the data sources from where the raw data originated. Section 5.4 (p.157) discusses the results within the context of answering the research questions outlined in Chapter 1 (p.1).

5.1 Participant Group Sizes

The number of participants who started the study and were issued with SQL-Tutor user IDs was fifty-nine in total, as detailed in Section 4.1.1 (p.95). Throughout the study it was clear from looking at the log files produced per user by SQL-Tutor that not all of the participants were actually using SQL-Tutor. At the end of the study thirty-nine (twenty in the Control group and nineteen in the Study group) of the original participants had logged onto SQL-Tutor. From this three participants from the Study group were discounted from the analysis as the log files showed that they had purely logged onto SQL-Tutor once and did not progress further than the welcome screen.

The analysis presented in this chapter uses the study results for the participants that logged onto and made use of SQL-Tutor (submitted answers to problems): thirty-six participants in total (twenty in the Control group and sixteen in the Study group), which is 61% of the participants who were issued with SQL-Tutor user IDs. The only exception to this is the analysis of the post-study questionnaire in Section 5.2.2 (p.120); only twenty-five of the thirty-six participants submitted a questionnaire. Table 5.1 shows the distribution of the final thirty-six participants in terms of course ratings and Milestone 1 marks. The distribution of the Milestone 1 marks was assessed using the Shapiro-Wilk test (sig. $p < .05$), which showed only the Study group marks were normally distributed, with $p = .123$, whereas $p = .035$ for the Control group was not normally distributed. Using the non-parametric Mann-Whitney U test (sig. $p < .05$), the marks for the Study group (M = 75.37, Mdn = 77.41) were not significantly different to the Control group (M = 77.44, Mdn = 77.41), U = 157, z = -.097, p = .937, r = -.016. This meant that the two study groups were still balanced in terms of using the Milestone 1 marks as pre-process comparison.

Table 5.1 Distribution of final participants including course rating and Milestone 1 marks
(Ratings: 1= low, 2= medium, 3 = high technical course content)

Milestone 1	Control Group				Study Group			
	Group Level	Ratings			Group Level	Ratings		
		1	2	3		1	2	3
Participants	**20**	2	10	8	**16**	3	7	6
Mean	**77.44**	83.29	79.69	73.15	**75.37**	81.37	80.39	66.51
Median	**77.41**	83.29	77.41	76.94	**77.41**	78.47	88.24	61.29
Std Dev	**18.15**	8.32	20.11	18.05	**17.86**	5.97	18.13	19.94
Range	**67**	12	67	56	**57**	11	57	56
Min	**33**	77	33	44	**43**	77	43	44
Max	**100**	89	100	100	**100**	88	100	100

5.2 Pre- and Post-Process Results

Section 4.3 (p.100) outlined three primary sources of data collection that were used in this study. Two of those sources were used to provide results for the pre- and post-study process: the summative assessments from the course module provided both pre- and post-process data, while the post-study questionnaire, as the name suggests, provided post-study process data. Section 5.2.1 below analyses the data from the summative assessments and Section 5.2.2 (p.120) analyses the responses gathered from the post-study questionnaire.

5.2.1 Module Summative Assessments

The data from the module summative assessments presented a unique opportunity in the study, compared to the other sources of data collected, in that the two study groups could also be compared to the students that did not take part in the study. This Non-Participants group allowed not only for the comparison between the two groups using different versions of SQL-Tutor, but also comparisons between non SQL-Tutor users and SQL-Tutor users. There were four assessments throughout the year-long module: Milestones 1, 2 and 3, as well as an Exam. Section 4.3.1, Tables 4.3 and 4.4 (p.100-102) provide further details of the summative assessments, their weightings and a breakdown of the proportion of each assessment that was used for this study. Section 4.1.1 (p.95) mentioned there were two-hundred and twenty-three students enrolled on the courses the database module was delivered to; by the end of the module delivery there were results available for two-hundred and nine students (twenty in the Control group, sixteen in the Study group and the remaining one-hundred and seventy-three in the Non-Participants group). The decline of fourteen was due to circumstances such as course withdrawals and changes.

Table 5.2 on the next page shows the distribution of the summative assessment marks for the Control, Study and Non-Participants groups. The statistics in the table are shown at three levels: group level (Control, Study, Non-Participants), course rating level (1, 2, 3) and for each of the courses that formed each course rating. The rating levels were used to group courses together that had similar levels of technical content (refer to Section 4.1 (p.93) for further details of ratings, along with course codes and names). The rest of this section uses content from Table 5.2.

Prior to detailing further analysis, the results in Table 5.2 were examined to see if there were any trends or values that stood out, with the observations detailed here. Comparing the mean and medians indicates that the results for all of the assessments for the Non-Participants group have a slight tendency towards the negative skew relating to a wider spread in the lower mark ranges. For the Study group, the results for all three Milestones have a negative skew, with only the Exam marks being symmetric. The Control group has a mixture, with the Milestone 1 results being symmetric, Milestone 2 has a negative skew tendency, and Milestone 3 and the exam have positive skews indicating a wider spread of marks occurred in the higher mark ranges. This shows that the Study and Non-Participants groups have a tighter cluster in the higher mark ranges and a wide spread across the lower mark ranges. The reverse of this is true for the last two assessments for the Control group.

Table 5.2 Distribution of all students including course rating and marks for all of the module summative assessments
(Ratings: 1= low, 2= medium, 3 = high technical course content)

Control Group

	Group	Rating 1 Total (BIS)	Rating 2 Total	Rating 2 ITMB	Rating 2 WDD	Rating 2 CS	Rating 3 Total	Rating 3 CFO	Rating 3 ETH	Rating 3 GAP
Participants	**20**	**2**	**10**	4	4	2	**8**	3	3	2
Mean										
Milestone 1	**77.44**	**83.29**	**79.69**	91.41	71.79	72.06	**73.15**	66.31	65.69	94.59
Milestone 2	**72.30**	**84.00**	**69.53**	78.83	60.50	69.00	**72.83**	69.77	68.67	83.66
Milestone 3	**57.50**	**72.22**	**55.56**	62.50	40.28	72.22	**56.25**	55.56	46.30	72.22
Exam	**56.25**	**37.50**	**47.50**	37.50	50.00	62.50	**71.88**	83.33	58.33	75.00
Median										
Milestone 1	**77.41**	**83.29**	**77.41**	94.12	76.94	72.06	**76.94**	77.41	66.71	94.59
Milestone 2	**77.00**	**84.00**	**71.68**	81.00	63.68	69.00	**75.00**	72.00	68.68	83.66
Milestone 3	**55.56**	**72.22**	**55.56**	61.11	38.89	72.22	**52.78**	50.00	44.44	72.22
Exam	**50.00**	**37.50**	**50.00**	37.50	50.00	62.50	**75.00**	75.00	50.00	75.00
Std Dev										
Milestone 1	**18.15**	**8.32**	**20.11**	10.86	27.88	6.24	**18.05**	19.22	11.33	7.65
Milestone 2	**13.45**	**7.52**	**15.89**	13.25	16.92	14.60	**10.41**	11.50	9.34	2.35
Milestone 3	**18.05**	**31.43**	**18.17**	8.33	12.32	7.86	**18.17**	25.46	8.49	7.86
Exam	**27.95**	**17.68**	**24.86**	14.43	20.41	53.03	**28.15**	14.43	38.19	35.36
Range										
Milestone 1	**67.00**	**12.00**	**67.00**	23.00	67.00	9.00	**56.00**	33.00	23.00	11.00
Milestone 2	**55.00**	**11.00**	**55.00**	31.00	40.00	21.00	**28.00**	23.00	19.00	3.00
Milestone 3	**67.00**	**44.00**	**16.36**	17.00	28.00	11.00	**50.00**	50.00	17.00	11.00
Exam	**75.00**	**25.00**	**75.00**	25.00	50.00	75.00	**75.00**	25.00	75.00	50.00
Min										
Milestone 1	**33.00**	**77.00**	**33.00**	77.00	33.00	68.00	**44.00**	44.00	54.00	89.00
Milestone 2	**37.00**	**79.00**	**37.00**	61.00	37.00	59.00	**57.00**	57.00	59.00	82.00
Milestone 3	**28.00**	**50.00**	**38.00**	56.00	28.00	67.00	**33.00**	33.00	39.00	67.00
Exam	**25.00**	**25.00**	**25.00**	25.00	25.00	25.00	**25.00**	75.00	25.00	50.00
Max										
Milestone 1	**100.00**	**89.00**	**100.00**	100.00	100.00	76.00	**100.00**	77.00	76.00	100.00
Milestone 2	**92.00**	**89.00**	**92.00**	92.00	77.00	79.00	**85.00**	80.00	78.00	85.00
Milestone 3	**94.00**	**94.00**	**78.00**	72.00	56.00	78.00	**83.00**	83.00	56.00	78.00
Exam	**100.00**	**50.00**	**100.00**	50.00	75.00	100.00	**100.00**	100.00	100.00	100.00

Study Group

	Group	Rating 1 Total (BIS)	Rating 2 Total	Rating 2 ITMB	Rating 2 WDD	Rating 2 CS	Rating 3 Total	Rating 3 CFO	Rating 3 ETH	Rating 3 GAP
Participants	**16**	**3**	**7**	4	1	2	**6**	2	2	2
Mean										
Milestone 1	**75.37**	**81.37**	**80.39**	74.26	77.41	94.12	**66.51**	49.53	88.71	61.29
Milestone 2	**75.58**	**78.23**	**75.62**	75.67	84.00	71.32	**74.22**	53.66	84.00	85.00
Milestone 3	**57.29**	**42.59**	**61.11**	54.17	94.44	59.33	**60.19**	44.44	61.11	75.00
Exam	**62.50**	**25.00**	**78.57**	68.75	100.00	87.50	**62.50**	62.50	75.00	50.00
Median										
Milestone 1	**77.41**	**78.47**	**88.24**	82.82	77.41	94.12	**61.29**	49.53	88.71	61.29
Milestone 2	**77.68**	**78.68**	**76.68**	78.34	84.00	71.32	**78.34**	53.66	84.00	85.00
Milestone 3	**61.11**	**61.11**	**61.11**	52.78	94.44	59.33	**58.33**	44.44	61.11	75.00
Exam	**62.50**	**25.00**	**75.00**	75.00	100.00	87.50	**50.00**	62.50	75.00	50.00
Std Dev										
Milestone 1	**17.86**	**5.97**	**18.13**	21.34	-	8.32	**19.94**	7.65	15.97	7.65
Milestone 2	**13.08**	**1.38**	**10.95**	14.28	-	0.00	**19.04**	19.32	5.66	11.77
Milestone 3	**23.19**	**36.99**	**22.91**	23.73	-	11.79	**16.64**	7.86	15.71	11.79
Exam	**27.39**	**25.00**	**17.25**	12.5	-	17.68	**20.92**	17.68	35.55	0
Range										
Milestone 1	**57.00**	**11.00**	**57.00**	45.00	0.00	12.00	**56.00**	11.00	23.00	11.00
Milestone 2	**53.00**	**3.00**	**34.00**	34.00	0.00	0.00	**53.00**	27.00	8.00	17.00
Milestone 3	**94.00**	**67.00**	**67.00**	56.00	0.00	17.00	**44.00**	11.00	22.00	17.00
Exam	**100.00**	**50.00**	**50.00**	25.00	0.00	25.00	**50.00**	25.00	50.00	0.00
Min										
Milestone 1	**43.00**	**77.00**	**73.00**	43.00	77.00	88.00	**44.00**	44.00	77.00	56.00
Milestone 2	**40.00**	**77.00**	**56.00**	56.00	84.00	71.00	**40.00**	40.00	80.00	77.00
Milestone 3	**0.00**	**0.00**	**28.00**	28.00	94.00	50.00	**39.00**	39.00	50.00	67.00
Exam	**0.00**	**0.00**	**50.00**	50.00	100.00	75.00	**50.00**	50.00	50.00	50.00
Max										
Milestone 1	**100.00**	**88.00**	**100.00**	88.00	77.00	100.00	**100.00**	55.00	100.00	67.00
Milestone 2	**93.00**	**79.00**	**90.00**	90.00	84.00	71.00	**93.00**	67.00	88.00	93.00
Milestone 3	**94.00**	**67.00**	**94.00**	83.00	97.00	67.00	**83.00**	50.00	72.00	83.00
Exam	**100.00**	**50.00**	**100.00**	75.00	100.00	100.00	**100.00**	75.00	100.00	50.00

Non-Participants

	Group	Rating 1 Total (BIS)	Rating 2 Total	Rating 2 ITMB	Rating 2 WDD	Rating 2 CS	Rating 3 Total	Rating 3 CFO	Rating 3 ETH	Rating 3 GAP
Participants	**173**	**13**	**102**	26	25	51	**58**	12	31	15
Mean										
Milestone 1	**58.31**	**50.98**	**59.08**	62.96	55.72	58.74	**58.59**	56.37	56.71	64.26
Milestone 2	**55.68**	**52.26**	**55.26**	55.69	45.42	59.85	**57.20**	54.06	56.93	60.27
Milestone 3	**47.01**	**44.44**	**49.02**	56.84	42.22	48.37	**44.06**	35.65	46.24	46.30
Exam	**46.68**	**44.23**	**45.59**	53.85	36.00	46.08	**49.14**	47.92	49.19	50.00
Median										
Milestone 1	**65.65**	**45.06**	**65.65**	65.65	54.94	65.65	**65.65**	55.88	65.65	65.65
Milestone 2	**63.36**	**62.00**	**63.00**	62.66	42.00	66.68	**64.34**	51.34	64.68	65.32
Milestone 3	**50.00**	**50.00**	**50.00**	61.11	44.44	50.00	**44.44**	41.67	50.00	50.00
Exam	**50.00**	**50.00**	**50.00**	50.00	50.00	50.00	**50.00**	50.00	50.00	50.00
Std Dev										
Milestone 1	**27.93**	**31.95**	**28.04**	30.86	24.77	28.35	**27.05**	34.82	26.34	22.18
Milestone 2	**23.35**	**23.87**	**23.08**	23.89	20.81	22.65	**23.99**	27.18	22.14	26.33
Milestone 3	**22.32**	**33.64**	**21.10**	24.86	17.57	19.67	**21.40**	21.38	20.95	22.09
Exam	**30.50**	**32.52**	**31.15**	30.57	33.14	29.74	**29.23**	29.11	29.92	29.88
Range										
Milestone 1	**100.00**	**100.00**	**100.00**	100.00	88.00	100.00	**100.00**	100.00	88.00	77.00
Milestone 2	**92.00**	**73.00**	**88.00**	87.00	68.00	88.00	**92.00**	92.00	83.00	82.00
Milestone 3	**100.00**	**100.00**	**100.00**	100.00	78.00	94.00	**83.00**	67.00	83.00	72.00
Exam	**100.00**	**100.00**	**100.00**	100.00	100.00	100.00	**100.00**	75.00	100.00	100.00
Min										
Milestone 1	**0.00**	**0.00**	**0.00**	0.00	12.00	0.00	**0.00**	0.00	0.00	23.00
Milestone 2	**0.00**	**0.00**	**0.00**	0.00	19.00	0.00	**0.00**	0.00	0.00	0.00
Milestone 3	**0.00**	**0.00**	**0.00**	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00
Exam	**0.00**	**0.00**	**0.00**	0.00	0.00	0.00	**0.00**	0.00	0.00	0.00
Max										
Milestone 1	**100.00**	**100.00**	**100.00**	100.00	100.00	100.00	**100.00**	100.00	88.00	100.00
Milestone 2	**92.00**	**73.00**	**88.00**	87.00	87.00	88.00	**92.00**	92.00	83.00	92.00
Milestone 3	**100.00**	**100.00**	**100.00**	100.00	78.00	94.00	**83.00**	67.00	83.00	72.00
Exam	**100.00**	**100.00**	**100.00**	100.00	100.00	100.00	**100.00**	75.00	100.00	100.00

Examining the means across all of the individual degree courses (refer to Table 4.1 (p.95) for the full names of the degree courses) for all of the assessments shows that the means for BIS (rating 1) and ITMB (rating 2) courses in the Control group gradually decreased for each assessment, with the other courses for the Control group fluctuating (moving up and down). The Study group only had one instance (BIS, rating 1) where the means decreased for each assessment, the means for five out of the six other courses within the Study group fluctuated. One course (WDD rating1) for the Study group was the only instance across all three groups where the means gradually increased for each assessment. The Non-Participants group had four instances where the means decreased for each assessment ITMB and WDD in rating 2, and CFO and ETH in rating 3), with the remaining three courses showing fluctuating means (BIS in rating 1, CS in rating 2 and GAP in rating 3).

For the three groups, the Non-Participants group has the lowest means for all of the assessments. The Control and Study groups swap positions on this, with the Study group having the highest means for Milestone 2 and the exam, with differences of 3.28 and 6.25 respectively. The Control group had higher means than the Study group for Milestones 1 and 3, with differences of 2.07 and 1.25. So where the Control group were higher than the Study group the differences were smaller than the instances where the Study group had the highest mean marks.

In terms of the minimum and maximum marks, the Non-Participants group had minimum marks of zero across all four assessments which was due to students not doing an assessment and therefore gaining the default mark of zero. Given the size of the Non-Participants group, this was not unexpected. Unfortunately the Study group had minimum marks of zero for Milestone 3 and the Exam due to one participant who had not taken those assessments. Where there are minimum marks above zero for Milestones 1 and 2, then the Study group has higher minimum marks than the Control group. For the maximum marks, they are nearly all the same between all three groups. The only differences are the maximum for Milestone 2 is one mark higher for Study group compared to the other two groups and the maximum mark for Milestone 3 is six marks higher for the Non-Participants group compared to the other two.

The basic statistics showed fluctuation across all groups and all of the assessments. Although the marks for the Control and Study groups dropped as the assessments progressed, the difference in the medians for Milestone 1 and the Exam was less for the Study group at 14.91 compared to 27.40 for the Control group. This would indicate a slight trend in favour of the Study group in terms of an increase in learning gains. Refer to Section 5.4 (p.157) for further discussion of these and the other results presented in this chapter. It is important to note that the assessments were conducted as part of a live delivery of a first year undergraduate degree module and were not subject to experimental control or conditions. So the decrease in marks from Milestone 1 to the Exam do not necessarily reflect the decline in understanding of the participants, but are more reflective of the increase in difficulty of the assessments which would have been designed to mark the progression of the learning process throughout the module. For example, some questions contained in the Exam could not have been placed in Milestone 1 as the required topic would not have been taught at the point of Milestone 1. Refer to Appendix E (p.225) for the questions contained in the assessments.

Following on from the basic statistics and observations in Table 5.2 (p.108) Shapiro-Wilk tests were conducted to determine whether the marks for the three groups were normally distributed. The test results for the twelve samples (four assessment marks across three groups) showed that seven samples were not normally distributed with $p < .05$: Control group Milestone 1 $p = .035$ and exam $p = .005$, Study group Milestone 2 $p = .04$ and all four assessments for the Non-Participants $p < .001$. The five samples that were normally distributed had *p-values* between .090 and .834. Due to the varied distributions presented across the samples, non-parametric tests were conducted to determine if any differences existed between the samples.

Kruskal-Wallis H tests, as the non-parametric version of parametric one-way independent ANOVAs, were conducted to determine if there were any statistically significant differences in the marks across the three groups for each of the summative assessments. When significant differences were detected, pairwise comparisons were conducted to help identify where the differences were between the groups. The adjusted *p*-values are presented.

The median Milestone 1 marks were significantly different between groups, $\boldsymbol{H}(2) = 23.081$, $p < .001$. The pairwise comparisons showed significant differences in the median Milestone 1 marks between the Non-Participants (*Mdn* = 65.65) and Study groups (*Mdn* = 77.41) ($p = .047$), as well as between the Non-Participants and the Control groups (*Mdn* = 77.41) ($p = .007$). There was no significant difference between the Control and the Study groups ($p = 1.000$).

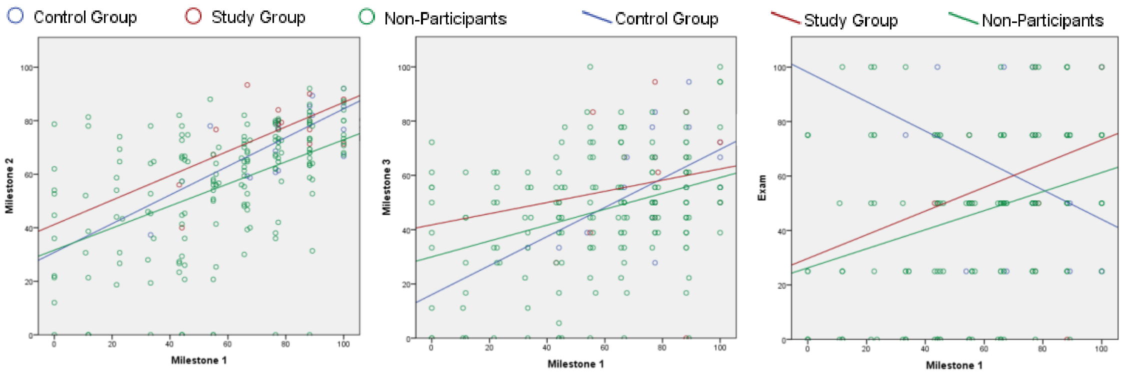
The median Milestone 2 marks were significantly different between groups, $\boldsymbol{H}(2) = 13.984$, $p = .001$. The pairwise comparisons showed significant differences in the median Milestone 2 marks between the Non-Participants (*Mdn* = 63.36) and Control groups (*Mdn* = 77.00) ($p = .004$), as well as between the Non-Participants and the Study groups (*Mdn* = 77.68) ($p < .001$). There was no significant difference between the Control and the Study groups ($p = 1.000$).

The median Milestone 3 marks were significantly different between groups, $\boldsymbol{H}(2) = 6.525$, $p = .038$. Despite a significant overall difference in the group marks, there were no significant differences shown in any of the specific comparisons between groups. Although not significant, the *p*-values follow the trend of the pairwise comparisons for the other two Milestone marks, with the lower values reported for the comparisons between Non-Participants (*Mdn* = 50.00) and Control groups (*Mdn* = 55.56) ($p = .179$), and Non-Participants and Study groups (*Mdn* = 61.11) ($p < .175$), with the Control and Study groups ($p = 1.000$) the same as the previous two Milestones.

The median Exam marks were not significantly different between groups, $\boldsymbol{H}(2) = 4.625$, $p = .099$. Overall the results for the Kruskal-Wallis H tests showed no significant differences between the Control and Study groups in the marks across the groups for each of the assessments. The pairwise comparisons were only significantly different between the two study groups and the Non-Participants, this would largely be expected as a reflection of the size of the Non-Participants group (173) compared to the Control (20) and Study groups (16) provides greater scope for the spread of marks. Refer to Section 5.4 (p.157) for further discussion of these and the other results presented in this chapter.

As a measure of learning gain ANCOVA tests would ideally have been conducted to determine whether there was any effect on the outcome of Milestones 2 and 3, and the Exam across the groups when using the marks for Milestone 1 as the covariate. The Milestone 1 marks could be used as the covariate in this instance because the marks can be treated as an independent variable, with Milestone 1 being taken by the participants prior to the start of the study and the use of SQL-Tutor. However, as already mentioned earlier in this Section, most of the mark samples for the summative assessments were not normally distributed and normal distribution is a requirement of ANCOVA as a parametric test.

As ANCOVA is a linear model, one assumption is the homogeneity of regression slopes, which would mean that the relationship between the covariate and the dependent variable is the same across the Control, Study and Non-Participants groups. Although it was already known that samples from the summative assessments were not the correct fit to conduct an ANCOVA, linear regression lines were applied against the data to see if they would show any points of interest. Figure 5.1 shows the linear regressions lines plotted on scatterplots with Milestone 1 as the covariate and the other summative assessments as the dependent variable: Milestone 2 in Figure 5.1a, Milestone 3 in Figure 5.1b and the Exam in Figure 5.1c.



	R^2 Linear
Control Group	0.523
Study Group	0.388
Non-Participants	0.243

Fig. 5.1a Milestone 1 against Milestone 2

	R^2 Linear
Control Group	0.290
Study Group	0.025
Non-Participants	0.134

Fig. 5.1b Milestone 1 against Milestone 3

	R^2 Linear
Control Group	0.123
Study Group	0.081
Non-Participants	0.103

Fig. 5.1c Milestone 1 against Exam

Figure 5.1 Scatterplots and linear regression lines with Milestone 1 as the covariate

The positive slopes (moving upwards from the bottom-left side) imply that the marks achieved in Milestone 1 are positively correlated with the marks in the other assessments. In other words those who did well at first continued to do well and those who did badly continued to do badly. All except one of the regression lines fits into this category with some degree of positive relationship. The negative slope is for the relationship between Milestone 1 and the Exam for the Control group. The negative slope implies that those who did well at first did less well later, and those who did badly did better later.

The R^2 values shown in Figure 5.1 are the measure of goodness-to-fit of linear regression. With an R^2 value of 0.0 meaning that there is no linear regression relationship between Milestone 1 and the other assessment marks, which would mean that knowing the Milestone 1 marks would not help to predict the other mark (e.g. Milestone 2). Nearly all of the R^2 values across all three scatterplots and groups are below 0.5, which indicates a low relationship between the assessment outcomes for each group. The only exception to this is the relationship between the Milestone 1 and 2 marks for the Control group (R^2 = 0.523), which is a much stronger predictor.

While the marks from Milestone 1 cannot be used to predict the marks for the other assessments, the linear regression lines did show that generally the participants who did well in the first assessment continued to progress and do well throughout the assessments. However the linear regression line for Milestone 1 and the Exam for the Control group indicated a reverse of this, with some participants who did well in Milestone 1 did less well in the Exam. This regression fits with the decline of medians as shown in Table 5.2 (p.108) and could also be explained by the use of live module assessments as discussed on p.110. Refer to Section 5.4 (p.157) for further discussion of all of the results presented in this chapter.

Figure 5.2 (p.116) presents four line charts, one for each summative assessment, which illustrates the range of marks and the percentage of marks achieved across the range for each of the groups. The chart for Milestone 1 shows that the peak mark range (71-80) is the same for the Control and Study groups, although the Control has 40% of marks in that range compared to the 31% for the Study group. Neither the Control nor Study groups have any marks in the lower ranges, 41-50 the first mark range for the Study group and the lower 31-40 range for the Control group. In contrast, the marks for the Non-Participants group are spread across all of the mark ranges, with the marks plateauing across a series of ranges (41-90).

The chart for Milestone 2 shows the peak mark range (71-80) was achieved for the Control and Study groups as in Milestone 1. However, it is the Study group with 56% of marks in this range which is 21% ahead of the 35% achieved for the Control group. Again both the Control and Study groups do not have any marks in the lower ranges, with the first marks in the 31-40 range. The Non-Participants group shows a slight peak of 25% in the 61-70 mark range.

The chart for Milestone 3 shows that the Control group is closer to the Non-Participants across all of the mark ranges, although the Control group has slightly more marks across the higher mark ranges than the Non-Participants. The Study group has two clear peaks of marks, with 25% in the 41-50 range and the largest peak of 31% in the 61-70 range which is one-range lower for the Study group in Milestones 1 and 2. In comparison the peaks for both the Control and Non-Participants groups are two mark ranges lower than in the previous two Milestones.

The chart for the Exam shows a different pattern across the mark ranges for all groups, with several peaks and troughs where mark ranges with no marks are peppered throughout the ranges, as opposed to being grouped at the start of the lower mark ranges as seen in the previous three charts. The largest peak in the 41-50 mark range is the same for the first time for all three groups: 35% for the Control group, 38% for the Study group and 28% for the Non-Participants group. In the highest two ranges that have marks (71-80 and 91-100) the Study group has the highest combined percentage at 50%, with the Control and Non-Participants groups equal at 35%.

In terms of general patterns presented in the charts, the charts for Milestones 1 and 2 are quite similar with a gradual incline to the peaks around the 61-70 and 71-80 mark ranges before decline to the 91-100 mark range. Milestone 3 shows a reduction in bunching of marks in the higher mark ranges, with a shift towards the middle mark ranges. The Exam marks shows another shift to marks being spread across the mark ranges but in distinct groupings. Milestones 1 and 2 have the greatest concentration in the second half of mark ranges (> 51): 90% and 95% respectively for the Control group, 88% and 94% for the Study group, and 64% and 68% for the Non-Participants group. The concentration of marks in the second half of ranges declines for Milestone 3 and the Exam: 60% and 35% respectively for the Control group, 56% and 50% for the Study group, and 40% and 35% for the Non-Participants group. These concentration of marks showed that the Control and Study groups were very similar in the three Milestones with the Non-Participants having much lower percentages, but in the Exam the Control group was the same as the Non-Participants group with 15% less in the second half mark range than the Study group.

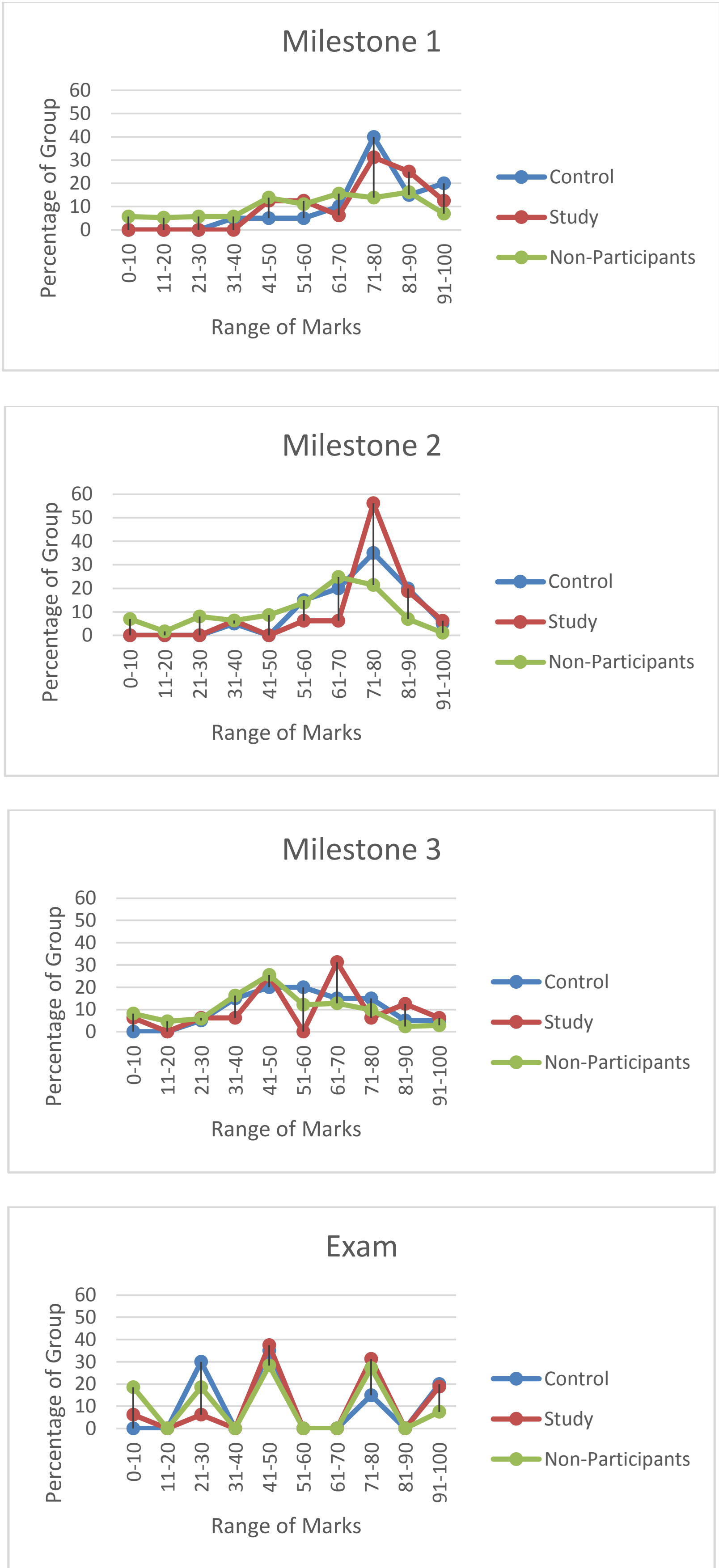


Figure 5.2 Percentage range of marks per assessment for all three groups

Earlier in this section the results of Kruskal-Wallis H tests were presented to detect any differences *across the groups for each of the summative assessments*. Here the results of Friedman tests, or Friedman's ANOVA (Field, 2013, p.250), are reported as a means to detect any statistically significant differences *across the summative assessments for each of the groups*. Friedman tests are the non-parametric equivalent of parametric Repeated Measures ANOVAs. When significant differences were detected, pairwise comparisons were conducted to help identify where the differences were between the groups. The adjusted *p*-values are presented. Friedman tests, followed by pairwise comparisons, with the adjusted *p*-values shown.

The marks for the Control group were significantly different across the four summative assessments, $\chi^2(3) = 14.894$, $p = .002$. Pairwise comparison showed significant differences in between the marks for Milestones 1 (*Mdn* = 77.41) and 3 (*Mdn* = 55.56) ($p = .009$), and Milestone 1 and Exam (*Mdn* = 50.00) ($p = .016$). There were no significant differences between the marks for the other pairings.

The marks for the Study group were significantly different across the summative assessments, $\chi^2(3) = 9.519$, $p = .023$. Pairwise comparison showed significant differences in between the marks only for Milestones 2 (*Mdn* = 77.68) and 3 (*Mdn* = 61.11) ($p = .045$). There were no significant differences between the marks for the other pairings.

The marks for the Non-Participants group were also significantly different across the summative assessments, $\chi^2(3) = 57.655$, $p < .001$. Pairwise comparison showed significant differences in between the marks for (a) Milestones 1 (*Mdn* = 65.65) and 3 (*Mdn* = 50.00) ($p < .001$), (b) Milestones 2 (*Mdn* = 63.36) and 3 ($p < .001$), and (c) Milestones 1 and 2 with the Exam (*Mdn* = 50.00) ($p < .001$). There were no significant differences between the marks for Milestone 1 and 2, and Milestone 3 and Exam.

Overall the Friedman tests showed some significant differences across the summative assessments for each of the groups, especially between the assessments conducted in Semester

1 (Milestones 1 and 2) and in Semester 2 (Milestone 3 and the Exam). Such differences do not show any learning gains in using either of the versions of SQL-Tutor or not using SQL-Tutor at all. Differences may be reflective of the length of time between the Semesters where the students were not actively engaged in study of the module subject (e.g. winter breaks, inter-semester assessment weeks). Although this may be somewhat negated by the fact that Milestone 3 occurred in week 6 of Semester 2. Refer to the study timeline in Appendix A, p.200) and to Section 5.4 (p.157) for further discussion of all of the results presented in this chapter.

Given the general decline in marks across the four assessments for all participants, the marks for Milestone 1 and the Exam (which were used as pre- and post-tests respectively) were ranked. These assessments and their marking were not under experimental control and they were of different kinds, gathered under different conditions and also marked more or less scrupulously. Tables 5.3 and 5.4 show the ranking data. Ranks were applied by ranking both groups as one, with all of the participants who achieved the highest mark (e.g. 100) ranked as 1. If there were four participants with 100, then the fifth participant would be ranked as 5, and so on. While the overall movement in ranking at group level reflected the means shown in Table 5.2, the mean change of rank of a Study group participant was between 3 and 4 places up the ranking, whereas the mean rank in the Control group stayed about the same. This indicates a small trend in favour of the Study group in terms of learning gains.

Table 5.3 Ranking data for the Control group

	Marks (%)		Ranks		Ranks	
User ID	Milestone 1	Exam	Milestone 1	Exam	Total	Movement
user11	76	50	24	16	40	8
user15	54	25	32	29	61	3
user25	77	50	15	16	31	-1
user27	77	75	15	8	23	7
user29	68	25	27	29	56	-2
user39	67	100	28	1	29	27
user43	100	100	1	1	2	0
user45	88	50	9	16	25	-7
user49	77	75	15	8	23	7
user51	89	25	7	29	36	-22
user53	100	25	1	29	30	-28
user55	76	50	24	16	40	8
user57	77	50	15	16	31	-1
user59	76	100	24	1	25	23
user61	89	50	7	16	23	-9
user63	100	25	1	29	30	-28
user67	77	50	15	16	31	-1
user69	100	25	1	29	30	-28
user75	33	75	36	8	44	28
user91	44	100	33	1	34	32
	Total		**330**	**314**	**644**	**16**
	Mean		**16.50**	**15.70**	**32.20**	**0.80**
	Median		**15.00**	**16.00**	**30.50**	**-0.50**

Table 5.4 Ranking data for the Study group

	Marks (%)		Ranks		Ranks	
User ID	Milestone 1	Exam	Milestone 1	Exam	Total	Movement
user14	78	50	14	16	30	-2
user22	88	0	9	36	45	-27
user24	77	25	15	29	44	-14
user26	77	50	15	16	31	-1
user38	77	75	15	8	23	7
user50	88	75	9	8	17	1
user58	67	50	28	16	44	12
user66	43	50	35	16	51	19
user68	56	50	30	16	46	14
user70	100	75	1	8	9	-7
user74	88	75	9	8	17	1
user76	88	100	9	1	10	8
user78	55	75	21	8	29	13
user88	100	100	1	1	2	0
user90	77	100	15	1	16	14
user92	44	50	33	16	49	17
	Total		**259**	**204**	**463**	**55**
	Mean		**16.19**	**12.75**	**28.94**	**3.44**
	Median		**15.00**	**12.00**	**29.50**	**4.00**

5.2.2 Post-Study Questionnaire

The final part of the study was a post-study questionnaire used to help gauge participants' opinions on using SQL-Tutor in terms of the perceived support and feedback that it provided. The questionnaire responses were analysed in direct relation to the research question "do the learners perceive any benefit from using an ITS to aid their learning" (refer to Section 1.2 (p.3) for all of the research questions and to Section 5.4 (p.157) for the discussion of the results presented throughout this chapter). Section 4.3.3 (p.103) introduced the structural aspects of the questionnaire; Section A of the questionnaire contained closed questions, with open questions contained in Section B. This section examines each question in turn and analyses the results from the completed questionnaires. There were no contradictions from the answers given in Section A to the opinions expressed in Section B of the questionnaire.

From the final thirty-six study participants that are included in the analysis of the study results, twenty-five participants submitted the post-study questionnaire. The questionnaire submission from one of the Study group participants was discounted from the analysis as it was a null response (all of Section A responses were 'Neutral' and all of the questions in Section B were 'n/a'). The remaining twenty-four complete questionnaire responses (12 in each of the Control and Study groups) gave a response rate of 67% of study participants.

5.2.2.1 Closed Questions

Section A of the post-study questionnaire contained nine closed questions which were presented as statements (Likert items) using a five-point Likert scale (strongly disagree = 1, disagree = 2, neutral = 3, agree = 4, strongly agree = 5). While closed questions mean that only simplistic responses are gathered, they also provide answers that are measurable and comparable. The Likert scale results for this analysis have been treated as categorical (ordinal) data, as the number assigned to each category is arbitrary (Field, 2013, p.721). Table 5.5 shows the results of the Pearson's chi-square tests (sig. $p < .05$) that were conducted against each of the statements to determine if any differences between the participants' responses were attributable to which study group they belonged to. The *p-value* for all of the statements were $p => .187$, so there was no significant association between the participant's study group and their responses. Table 5.5 also presents the results of the Fisher's exact tests (sig. $p < .05$), which were also conducted to compute the exact probability of the Pearson's chi-square statistic because the sample size was small and each of the statements had instances where the 'expected frequencies' were below 5. In small samples the approximate chi-square distribution *may* be inaccurate, whereas the Fisher's exact test *is* accurate for small samples (Field, 2013, pp.723-724). The Fisher's exact tests gave *p-values* for all of the statements that were $p => .213$, so these results confirm those of the Pearson's chi-square tests, with no significant association between the participant's study group and their responses.

The rest of this sub-section presents the analysis of the results for each closed question statement in turn, finishing with the summary analysis. The numbers and percentages of responses to the statements in Section A of the post-study questionnaire are presented in Figure 5.3, which contains ten bar charts: one for each of the nine statements and a summary chart.

Table 5.5 Pearson's chi-square and Fisher's exact tests on participants' responses to Section A of the Post-Study Questionnaire

Statements	**Pearson's chi- square**	**Fisher's exact test**
Statement 1 – Structure Queries	p = .327	p = .545
Statement 2 – Learn Concepts	p =.663	p = .829
Statement 3 – Find Mistakes	p =.952	p = 1.00
Statement 4 – Overcome Difficulties	p =.187	p = .213
Statement 5 – Relate Previous Problems	p =.478	p = .613
Statement 6 – Accurate Assessment	p =.653	p = .677
Statement 7 – Keep Going	p =.579	p = .739
Statement 8 – Integrate with Lab Sessions	p =.504	p = .564
Statement 9 – Slider Control Frequency	p =.327	p = .545

Statement 1: The SQL Tutor helped me to learn how to structure database queries.

This statement is domain related and was asked to determine whether the participants felt that SQL-Tutor helped them when structuring SQL SELECT queries. The majority of both groups 'agreed' with this statement, although the Study group were higher in their agreement (75% to 58% of the Control group). However the Control group had more participants who 'strongly agreed' with this statement (25% to 17% of the Study group). The Study group had a more positive lean across the responses to this statement, with 'neutral' (8%) being the lowest response. While the Control group had the highest 'strongly agree' responses, they also had the most negative responses with 17% 'disagreeing' with this statement.

Statement 2: The SQL Tutor helped me to learn database concepts.

This statement is also domain-related, but more general than Statement 1, as it deals with database concepts as a whole. The aim was to see whether the participants felt that by practicing SQL SELECT queries they felt it helped them to understand the underlying database concepts (e.g. relationships with primary and foreign keys, and table joins). The majority of both groups, not only 'agreed' with this statement, but also the response levels were the same as for Statement 1 (58% for the Control group and 75% for the Study group). While the Study group had a positive lean on responses to Statement 1, in this statement, their responses covered the three middle points of the scale, although it was still overwhelmingly positive with 75% 'agrees' to 8% 'disagrees' and 17% 'neutrals'. The Control group responses were spread across four points of the scale from 'disagree' to 'strongly agree' and while the same number of responses were reflected in 'disagrees' (17%) and 'agrees' (58%) as in Statement 1, the remainder for the Control group were split between 'neutral' (17%) and 'strongly agree' (8%).

Statement 3: The SQL Tutor helped me to find where my mistakes were.

This statement is related to the learning process and aimed to determine whether the participants felt that SQL-Tutor helped them to identify their mistakes. Not only did the majority of both groups 'agree' with this statement, exactly the same agreed, with 42% from each group. The spread of answers were also the same for both groups with the four points from 'disagree' to 'strongly agree' covered, with the responses from both groups the same for both 'disagree' (8%) and 'agree' (42%). The slight difference occurred for the 'neutral' and 'strongly agree' responses, with the Control group 25% for both points and the Study group leaning slightly more to the positive with 17% and 33% respectively.

Statement 4: The SQL Tutor helped me to overcome difficulties.

This statement is also concerned with the learning process and aimed to gauge the participants' responses to whether they thought that SQL-Tutor helped them deal with any difficulties they may have faced. Similar to the previous statement, the majority for both groups not only 'agreed', but also with the same percentage of responses at 50%. This was the same for 'strongly agree', with 17% of the responses from both groups. The Study group was spread more positively with responses covering three points from 'neutral' to 'strongly agree', although there was a large proportion (33%) of Study group responses that were 'neutral' for this statement. The Control group responses spread over four points from 'disagree' (25%) through to 'strongly agree' (17%), with only 8% responses 'neutral'.

Statement 5: SQL-Tutor helped me to solve problems by relating them to previous problems solved.

This statement is related to metacognition and was asked to determine whether using SQL-Tutor helped the participants solve problems by relating to previous problems. Since the Study group used the modified version of SQL-Tutor that explicitly gave feedback relating to previous problems, it would be reasonable to expect the highest of positive responses to this statement to come from participants in the Study group. While this was the case, with the majority of responses from the Study group positive (67% 'agree'), it was not as clear cut as may have been expected. The Study group responses covered the middle three points, with 8% 'disagreeing', 25% 'neutral' and the 67% 'agreeing'. The majority of both groups 'agreed' with this statement, and 8% of the Control group also 'strongly agreed'. This gave a combined positive response ('agree' and 'strongly agree') of 67% for the Study group and 58% for the Control group. The negative response of 'disagree' was highest for the Control group, at 25% compared to 8% for the Study group. With the majority of the Control group also agreeing with this statement, one element of discussion that cannot be extracted from the data is the degree to which learners naturally recall and relate

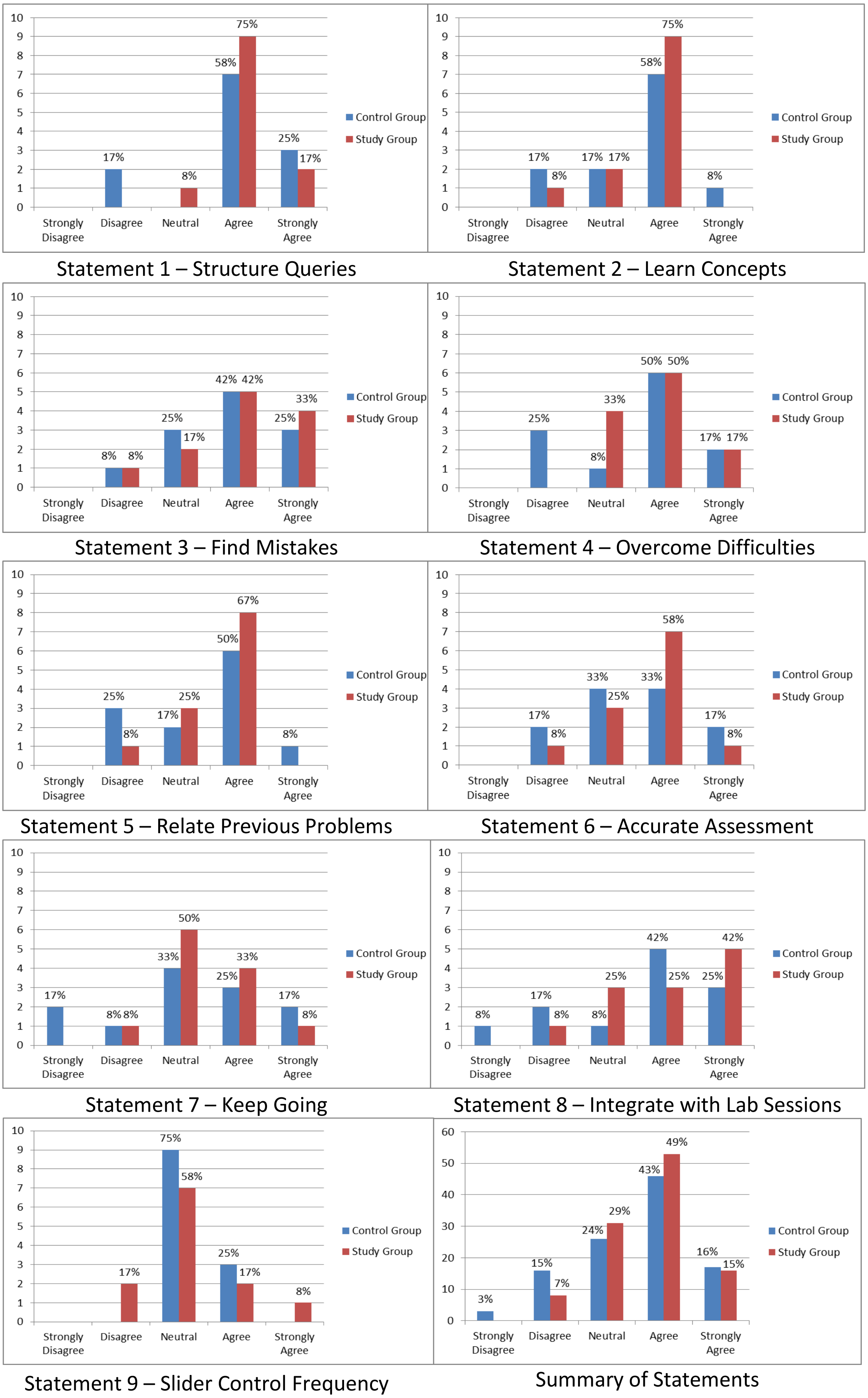


Figure 5.3 Participants' responses to Section A of the Post-Study Questionnaire

to previous problems during their normal learning process, however this would be an interesting potential point for further work. Although the majority of Study control responses agreed with this statement, the responses may have been affected by the fact that the rules governing the provision to explicitly give feedback relating to past problems was never triggered for five of the participants (31%) in the Study group (refer to Section 5.3.4.1 for further discussion).

Statement 6: The SQL Tutor helped me to gain an accurate assessment of what I understood and what I did not.

This statement is also related to metacognition and aimed to gauge the participants' self-assessment of their SQL SELECT query knowledge. This is the first statement so far where the majority of the responses are not for the same point on the scale. The majority of Study group participants 'agree' (58%) with this statement, where there is an equal majority of Control group responses split across 'neutral' and 'agree' (with 33% each). The responses to this statement also show the largest gap between the main response point, with 25% more 'agreeing' in the Study group as opposed to the Control group. Four points from 'disagree' to 'strongly' agree were covered by both groups, with the Study group leaning more towards the positive responses and the Control group responses spread in a symmetrical bell curve (17%, 33%, 33% and 17%).

Statement 7: The SQL-Tutor helped me keep going when things were tough.

This statement is related to motivation and aimed to determine whether using the tutor helped them to keep working at the problems. This is the second statement in a row where the majority of responses are not for the same point on the scale. Similar to the responses in the previous statement, the majority of responses for the Control group is split at 33% across two points ('neutral' and 'agree'). It is also the first instance where the majority, as guided by the clear majority (50%) of the Study group, is on the 'neutral' point as opposed to 'agree' in the previous statements. In addition to this, it is also the first instance where responses from a group have covered all five points of the scale; the Control group has 17% who 'strongly disagree', as well as 17% who 'strongly agree'. The Study group responses range from 8% 'disagree' to 8% 'strongly agree'.

Statement 8: I would have preferred to have SQL-Tutor integrated into the lab sessions as opposed to a tool for extra study.

This statement is concerned with the general use of the tutor and aimed to determine whether the participants would have preferred the tutor to be incorporated into the timetabled lab sessions. This is the first instance where there is a two-response majority, although presented in reverse; the ‘agree’ point had 42% from the Control group and 25% from the Study group, whereas the ‘strongly agree’ point was the reverse, with 42% from the Study group and 25% from the Control group. While the responses from both groups were strongly positive, the Control group responses were spread across all five points of the scale, with the Study group spread across four points starting with ‘disagree’.

Statement 9: IF you were prompted to use a slider control in SQL-Tutor to indicate your confidence with SELECT queries, do you agree that the frequency of these prompts was unobtrusive? Select 'Neutral' if you did not receive these prompts.

This statement is also concerned with general use of the system, but was aimed at the Study group participants, as it aimed to determine whether the participants found the frequency of self-report to be unobtrusive to the flow of using the tutor. This is only the second instance where the majority of responses (58%) have been ‘neutral’ as opposed to positive. The remaining responses in the Study group were split between ‘disagree’ and ‘agree’ at 17% each, and 8% ‘strongly agreeing’. Taking ‘neutral’ as indifference, then the majority of the Study group participants were indifferent to the frequency of self-report prompts, followed by a combined 25% finding the frequency unobtrusive and finally 17% finding the frequency obtrusive. Although the Control group participants were instructed to give ‘neutral’ responses, 25% responded with ‘agree’.

As a summary of responses for all nine of the statements, the majority of answers for both groups was ‘agree’, with 43% of Control group and 49% of Study group responses. While the majority of responses for both groups leant towards the positive end of the scale (‘agree’ and ‘strongly agree'), the majority of negative responses ('disagree' and 'strongly disagree') were given by the Control group. From examining the responses at group level, there does not appear to be any clear instances of the biases that are known risks when using Likert scales; for example, central tendency bias, acquiescence response bias and social desirability bias (Barnette, 2010).

5.2.2.2 Open Questions

To help gauge the participants' opinions of using SQL-Tutor and the feedback comments that it provided, Section B contained six open questions. The free-format nature of the questions allowed the participants to answer in their own words and without restriction. A discussion of the analysis of the varying responses for each question follows.

Question 1: What did you like MOST about using the SQL-Tutor?

This question was aimed at drawing out what the participants thought were favourable aspects of using SQL-Tutor, with the potential for some of the points raised to influence any further work. All twenty-four participants supplied an answer to this question. The feedback from the answers provided can be divided into three general categories: user interface, feedback and availability. The split of the categories covered by the feedback comments was equal across both the Study and Control groups. There was one feedback comment that was an exception to this in that it simply stated "there was nothing I liked about the SQL-Tutor", although at least their comment is consistent with their answers to the nine statements in Section A of the questionnaire ('disagree' to statements 1-7, 'strongly disagree' to statement 8 and 'neutral' to statement 9).

Six participants (25%) provided comments about the user interface as the aspects they liked most. All of the comments for this said that they found SQL-Tutor easy to use. Refer back to the discussion of web design guidelines and usability at the beginning of Section 3.2.2 (p.67).

Thirteen participants (54%) mentioned feedback in various forms as the most favourable aspect. The feedback mentioned related to the tutor showing the errors (e.g. "showed my errors" and "shows you where you went wrong"), the instant feedback received (e.g. "having instant feedback on the mistakes I made"), the different hint levels availability (e.g. "the differing levels of hints") and the information and flexibility provided by the learner model (e.g. "easy to see where you need to work", "I enjoyed how it adapted to what I needed rather than sticking to a constant plan" and "that is was assessing the difficulty based on my knowledge"). In terms of the learner model one participant also mentioned that the tutor "stated what level I was on which was helpful as this showed me I was actually better at SQL coding than what I thought".

One participant (4%) also mentioned the availability of SQL-Tutor as a Web-based tool. It was available all hours of the day and every day of the week. Also the participants did not need any special software to be able to use the tutor; only a standard Web browser was required, as opposed to the participants doing their normal coursework being restricted on the University campus to computers that had Oracle's client interface installed.

Question 2: What did you like LEAST about using the SQL-Tutor?

This question provided a balance to the previous question, with the potential for some of the points raised to influence any further work. Twenty-three out of the twenty-four participants (96%) supplied an answer to this question. The feedback from the answers provided can be divided into three general categories: user interface, content and the learner model. The split of the categories covered by the feedback comments was equal across both the Study and Control groups.

Eight participants (33%) commented on their dislike of the user interface. The comments ranged from "UI isn't great" to "rather clunky interface that was confusing at the start". Refer back to the discussion of web design guidelines and usability at the beginning of Section 3.2.2 (p.67).

Nine participants (37%) provided comments relating to the content which can be broadly divided into two sub-categories: comparison to the style of questions in the module Milestone tests and restrictions with what SQL-Tutor accepted as a correct answer. The comments regarding the style of questions either directly related to the types of questions (e.g. "it wasn't styled like our milestone exams") or to the SQL SELECT syntax covered (e.g. "differing syntax to that taught in lectures, seminars/workshops"). The exception to this is in the comment "asking me about select clauses all the time" which is a little surprising given that SQL-Tutor purely covers SQL SELECT problems. In the module Milestone tests quite a few questions were multiple choice or only required part of the SELECT statement as the answer, whereas in SQL-Tutor every problem requires the full SELECT statement to be entered. In terms of the syntax covered by the problems in SQL-Tutor, Section 4.2 gave an example where some syntax short-cuts were taught in the module but not supported by SQL-Tutor. The comments regarding the restrictions of answers accepted by SQL-Tutor all said that it appeared that you had to input an exact answer expected by the tutor for it to be correct (e.g. "Also the queries were rigid.. I had to get it exactly as the computer knew the answer"). Following further investigations into the apparent 'rigidity' of answers in the tutor, the next two points are relevant:

- The first is the participant's perception of the question requirements and the accuracy of such perceptions, coupled with their level of SQL SELECT query knowledge. This can be seen from looking through the system log files and reviewing the attempts submitted, especially for the participants who gave the comments on restrictive answers. Table 5.6 shows an example of the attempts submitted by a participant to answer problem number 258 in the books database ("List all details of all authors. Order the list by last name"). This example shows an instance where the tutor is being flexible in the answer that is accepted as correct; the participant's final attempt is marked as correct as it is a valid, although longer version of the model answer. At the same time this example shows how the participant's perception on restrictive answers may have arisen. From the attempts submitted, the participant may have been aiming for the same as the model answer, but the submissions appear to show a lack of knowledge with the misplacement of the table name (placing 'author' in the SELECT clause instead of in the FROM clause).

Table 5.6 Example of a participant's problem attempts

Problem: List all details of all authors. Order the list by last name.
Model Answer in SQL-Tutor: SELECT * FROM author ORDER BY lname

Participant input	SQL-Tutor Marked as
SELECT author* FROM ORDER BY	Incorrect
SELECT author* FROM ORDER BY	Incorrect
SELECT authored, lname, fname FROM author ORDER BY lname	Correct

- Given that SQL-Tutor implements Constraint-Based Modelling (as described in Section 3.1.1, p.60), each problem in the system has a series of constraints that are to be met for a correct answer. Tightly integrated with the first point discussed, there were examples of participants submitted a different list of fields in the SELECT clause than was required for the problem (e.g. "city, name" entered instead of the required "city"). Apart from an initial misunderstanding of the fields required by the participant, the continuation of this through further attempts may also relate to the granularity of the constraints themselves. The feedback for a constraint may be either too narrow or not fine-tuned enough to point to the error. While outside the scope of this thesis, the granularity of constraints and associated feedback could be reviewed in future work.

Two participants (8%) commented on the learner model which is maintained by SQL-Tutor to track their level of understanding of SQL SELECT query elements and that is also display to the user to help them in their choice of problems. The comments from both participants showed that they did not agree with the learner model: "The progress bar did not seem to represent where I thought I was with the skills" and "The way it judges which topics are strongest and weakest doesn't seem to be very accurate". This could be explored by extending the current open learner model to be negotiated between the ITS and the learner (see Bull and Kay, 2013; Bull and Vatrapu, 2012). This point is discussed in Section 6.5 (p.183) in relation to further work.

Question 3: What did you think of the feedback comments from the SQL-Tutor?

This question aimed to gain the participants' opinions of the feedback given by SQL-Tutor, however it did not explicitly target the different types of feedback (domain, motivation, metacognitive). This point is discussed in Section 6.5 (p.183) in relation to further work. All of the twenty-four participants supplied an answer to this question, which can be divided into two general categories: positive and negative comments. The split of comments into the categories was equal across both the Study and Control groups.

Nineteen participants (79%) responded with positive comments, with 58% of those comments purely positive feedback, with the remainder also provided constructive critiques. The purely positive comments ranged from "helpful" and "pretty good" to "really easy to understand, and really helped understand which part was going wrong". The comments that included constructive critiques related to the perceived understanding and helpfulness of the feedback; for example, "they were helpful, although sometimes they tend to be vague" and "some were very good and easy to see what was wrong but others were slightly more confusing".

Five participants (21%) responded with negative comments in terms of their dislike of the feedback provided by SQL-Tutor. One comment was purely a statement ("didn't agree with the feedback") while the rest provided constructive critiques (e.g. "feedback could be worded better, found it quite difficult to understand at times").

Question 4: Please mention any other aspect of the SQL-Tutor that you would like to comment on.

This general question aimed to capture any salient points that did not necessarily fit into the previous three questions. Eight out of the twenty-four participants (33%) supplied an answer to this question. The feedback from the answers provided can be divided into four general categories: feedback, user interface, usage and technical issues.

Two participants (8%) commented on the feedback provided by SQL-Tutor. One suggested that the tutor should include all possible answers, if more than one answer is applicable to a problem. The other comment suggests that the feedback "needs to exactly state where you went wrong". In relation to this, the same participant responded positively to the previous question regarding system feedback: "pretty good".

Two participants (8%) commented on the user interface of SQL-Tutor, with both stating that the interface should be updated: "it needs to be updated" and "I think the interface should be modernised and improved greatly but other than that, the system was fairly good".

Two participants (8%) provided comments relating to the usage of SQL-Tutor. One suggested that it should be available as part of the module workshops. This corresponds to Statement 8 in Section A of the post-study questionnaire which explicitly asked whether the tutor should be available in such sessions, with 67% of the respondents in both the Study and Control groups agreeing/strongly agreeing with the suggestion. The other comment was concerned with usage in terms of the content purely using the same database that was used in the module workshops.

Two participants (8%) commented on technical difficulties that they had experienced using SQL-Tutor during the study: "sometimes I couldn't log on" and "had a small connecting problem… but it was all sorted by the administrator and worked flawlessly". There was one instance where the system was down, as mentioned in Section 4.2 (p.97).

Question 5: Did you use anything else to help you practice SQL SELECT queries outside of the timetabled lecture and lab session? If yes, what else did you use and why.

This question aimed to determine what, if any, other forms of help or tools the participants used to help them practice SQL SELECT queries. Twenty-three out of the twenty-four participants (96%) supplied an answer to this question. Table 5.7 outlines any sources that the participants used, with the majority purely using the module materials and SQL-Tutor.

Table 5.7 Alternative sources of material used by participants to aid learning SQL SELECT queries

Other forms of material	**Number of participants (%)**
Nothing	20 (83%)
W3schools (www.w3schools.com)	2 (8%)
Web resources (no examples were given)	1 (4%)

Question 6: IF you were prompted to use a slider control in SQL-Tutor to indicate your confidence with SELECT queries AND you thought the frequency of prompts was obtrusive, when and how often would you have preferred to be prompted (i.e. at the start & end of a session only)?

This question aimed to find out what the participants thought would be an acceptable use of the self-report slider control that monitored the participant's self-efficacy. Since the slider control was only available to the Study group, it was expected that comments would only be received from the Study group participants. However, in addition to the comments received from five (42%) Study group participants, a further five Control group participants also replied. The comments are discussed in relation to statement 9 in Section A of the post-study questionnaire (referred to as Statement 9 in the following two paragraphs) which asked whether the participant found the frequency of the prompts to use the slider control unobtrusive.

From the five Study group participant answers, two did not think they had been prompted to use the slider control. In relation to the participant's response to Statement 9, one participant was 'neutral' to the frequency of the prompts, where the other participant found it unobtrusive ('agree'). The remaining three Study group participants each had a different suggestion: "start and end of a session or after a set number of questions", "end of session only" and "not every time I used the tutor e.g. every three times". Given individual learner tolerances of receiving such prompts, future work may include a version where the frequency could be set by the learner, but also warn the learner about the potential consequences of doing this (e.g. the system may not be quite as reactive to the learner). This point is discussed further in Section 6.5 (p.183) in relation to further work.

From the five Control group participant answers, two thought that they should be prompted at the start and end of a session, one thought "when I've run the query", one thought in the middle of a session to help them "get back on the right track" and the last one thought "at the start of the first session and the end of each one after that". Despite the fact that these participants did not get prompted to use the slider control, the first and last suggestion appear feasible and could be considered in any further work. In terms of their answers to Statement 9, one of the 'start and end of session' and the when 'the query is run' participants answered with 'agree' and the other three of these participants answered 'neutral' to the frequency of the slider control.

5.2.2.3 Post-Study Questionnaire Summary

Overall the responses gained from the post-study questionnaire showed that both groups agreed that using SQL-Tutor helped them to strengthen their knowledge of database concepts (domain-related Statements 1, p.121 and 2, p.122). This would indicate that the motivational and metacognitive feedback provided in the SQL-Tutor version used by the Study group did not influence the overall perception of the usefulness of using an ITS. However, in the responses to the learning process statements in Section A of the questionnaire (Statement 3 (p.122) – SQL-Tutor helping to identify their mistakes and Statement 4 (p.125) – SQL-Tutor helping the participants to overcome any difficulties), the Control had the most negative and neutral responses. In contrast to the responses to the domain-related statements, the responses to the learning process statements would indicate a slight trend in favour of the Study group and to including motivational and metacognitive feedback having a positive influence on the participants' perceptions of using a tutoring system to help them learn. This is also reflected by the responses for the statements related to aiding metacognition (Statements 5, p.123 and 6, p.125), although the fact that the Control group also responded positively to the effect of relating to past problems was a bit of a surprise as they were not given feedback that explicitly related to past problems. Such a response indicates that linking back to previous problems must naturally occur as part of the metacognitive process. The response to Statement 7 (p.125) which was related to motivation was mostly neutral for both groups, although there was still a large positive response from both groups (41% for the Study group and 42% for the Control group). Statements 8 (p.126) and 9 (p.126) of Section A of the questionnaire gauged the participants' views about SQL-Tutor being included in the standard delivery of the module and whether the frequency of self-reports were unobtrusive. Both groups strongly agreed that SQL-Tutor should be included in the module, such a response is in-line with their reception to the usefulness of using a tutor like SQL-Tutor. While the majority of participants found the frequency of self-report prompts was unobtrusive to their use of the tutoring system, it was surprising that 25% of the Control group responded to this. Such a response can be classified as the participants not ready the instruction in this statement correctly and providing a non-neutral response.

Refer to Section 5.4 (p.157) for further discussion of all of the results presented in this chapter, especially to the discussion of participants' perception of using ITS to learn as gained from the post-study questionnaire results (p.162).

5.3 Process Results

Recording the participants' interactions with SQL-Tutor was achieved by the system maintaining a set of four log files: a core log file produced by the base version of SQL-Tutor and three additional log files produced by the study version of SQL-Tutor. This means that the base log files were produced for all participants in both groups (Control and Study), whereas the additional logs were produced for the Study group participants only. Refer to Sections 3.3.2 (p.78) and 4.3.2 (p.102) for further details of the log files and also to Sections 6.4 (p.180) and 6.5 (p.183) for lessons learnt and potential further work. Data was extracted from the log files to analyse the use of SQL-Tutor as part of the learning process of SQL SELECT queries. The analysis has been grouped into the following four sub-sections: the problems attempted and viewed by the study participants, the use of the SQL-Tutor's help resources, participants' learning curves and the Study group's use of past problems and self-reports. The data was analysed towards providing an answer to the research questions, especially "does providing such feedback [motivational and metacognitive based on past states and experiences] lead to any measurable learning gains" and "does providing such feedback lead to any measurable gains in learner focus (refer to Section 1.2 (p.3) for all of the research questions and to Section 5.4 (p.157) for the discussion of the data analysis).

5.3.1 SQL-Tutor Problems Visited

The fundamental process of SQL-Tutor is problem-solving by presenting users with requirements for SQL SELECT queries for them to solve. Combined with the activity recorded in the log files, then there is a rich source of data relating to problems that can provide an insight to the progress of the person using SQL-Tutor. It is this analysis of the problem data that is covered in this section.

Tables 5.8, p.139, (Control group) and 5.9, p.139, (Study group) show a breakdown of the total time the participants spent logged into SQL-Tutor, the number problems attempted and viewed, along with the number of attempts submitted for the problems. For the scope of this thesis, attempted problems refer to problems that the study participant attempted to solve at least once, this is regardless of whether or not they managed to solve the problem. Viewed problems refer to problems that the study participant opened in the main problem page, but did not submit any attempt at an answer. The statistics (mean, median, standard deviation, range, along with minimum and maximum marks) are shown at group level, as well as for each area of problem analysis (solved, unsolved and viewed).

As mentioned in Section 3.3.2.2 (p.90) some of the results (Table 3.6, p.90) from a previous study using SQL-Tutor (testing use of negative and positive feedback) were used to help set parameters used in version of SQL-Tutor used in this study (e.g. effort levels). Prior to analysing the results in Tables 5.8 and 5.9, the results were reviewed in relation to the previous study to see whether the usage results were approximate to another SQL-Tutor study. Where results are listed for this comparison, they are given for the Control, then Study groups. Although the group sizes were not dissimilar between the two studies (twenty-three and eighteen, as opposed to twenty and sixteen in this study), the amount of time spent using SQL-Tutor was extremely different: (00:03:13 and 00:01:32, compared to 28:31:17 and 17:21:38). Although such time differences are to be expected as this study was designed for the participants to use SQL-Tutor over a period of three and a half months and the previous study covered a one-month period (Barrow et al., 2008). The average time (in minutes) spent per solved problem was 9.8 and 5.8 for the previous study, compared to 6.95 and 6.2 for this study. This showed the same pattern, in that the Control group took longer to solve a problem, but the difference for this study was not significant. The average attempts per solved problem were 5.5 and 5.2 for the previous study, compared to 4.3 for both the Control and Study groups in this study. This showed the same pattern between studies with no significant difference in the previous study and no difference for this study, although the participants in this study took on average one attempt less than the previous study participants. While this and the previous study are not directly comparable as they had different focuses, this comparison does help to show that this study did not introduce anything that made the usage data go adrift from previous usage of SQL-Tutor.

In Tables 5.8 and 5.9 (p.139) the samples represented in the total session time and the problems solved, unsolved and viewed columns, were assessed for normal distribution using Shapiro-Wilk tests. Only two samples fitted a normal distribution: the average initial rating of unsolved problems for the Control group (p = .064) and the average initial rating of solved problems for the Study group (p = .070). All other samples had *p-values* of < .05, so contained values that were different from a normal distribution. Taking this into consideration to determine whether the samples across groups were significantly different, and thereby subject to more than chance, the non-parametric Mann-Whitney U tests were conducted, with the results presented throughout this sub-section.

The ‘Total Sessions Time’ columns in Table 5.8 and 5.9 show the total length of time (in the format of ‘hours:minutes:seconds’) across all sessions that each of the participants spent logged into and using SQL-Tutor. In the scope of this thesis, a session refers to an instance where the participant has logged into SQL-Tutor and represents the time from logging in to logging out of the system. The times shown are the adjusted times after periods of inactivity and automatic logouts have been deducted. If there was a period of approximately thirty-minutes where the participant had not used SQL-Tutor while logged on, then the system would automatically end the session by logging the participant out. The Control group spent just over eleven-hours longer using the system than the Study group, although some of that difference would be expected as the Control group had four participants more than the Study group. Using the non-parametric Mann-Whitney U test, the mean time spent using SQL-Tutor for the Study group (M = 01:05:06, Mdn = 00:51:58) was not significantly different to the time spent by the Control group (M = 01:25:34, Mdn = 00:56:57), U = 150, z = -.318, p = .765, r = -.053.

In terms of the number of problems that were solved by the participants, the Control group solved 32% more problems than the Study group at 284 and 215 respectively. This is also reflected in the number of attempts made at solving the problems; the Control group (1229) made 32% more attempts at the problems than the Study group (930). The Mann-Whitney U tests showed that the mean number of problems solved and the mean number of attempts for the Control and Study groups were not statistically different. The Study group (M = 13.44, Mdn = 7.50) and the Control group (M = 14.20, Mdn = 10.00), U = 149.5, z = -.335, p = .741, r = -.056 for problems solved, and for the attempts taken the Study group (M = 58.13, Mdn = 39.50) and the Control group (M = 61.45, Mdn = 37.00), U = 161, z = .032, p = 1.000, r = .005.

Table 5.8 Control Group – Summary of participants' activity using SQL-Tutor

User ID	Total Sessions Time	Solved			Unsolved			Viewed		Total Problems	Total Attempts	Avg Time per Solved Problem
		Problems	Attempts	Average Initial Rating	Problems	Attempts	Average Initial Rating	Problems	Average Initial Rating			
user11	00:16:05	1	1	1	1	12	7	0	0	2	13	0:16:05
user15	01:52:43	31	148	4	2	26	7	0	0	33	174	0:03:38
user25	02:27:20	34	178	2	6	13	4	3	6	43	191	0:04:20
user27	00:18:40	8	18	2	4	12	2	0	0	12	30	0:02:20
user29	01:37:53	17	73	3	2	3	6	2	4	21	76	0:05:45
user39	05:14:48	52	194	3	1	13	9	1	1	54	207	0:06:03
user43	00:14:48	2	7	2	1	26	5	1	2	4	33	0:07:24
user45	00:02:25	1	1	2	0	0	0	0	0	1	1	0:02:25
user49	00:23:54	13	63	3	0	0	0	1	6	14	63	0:01:50
user51	02:56:05	16	84	3	0	0	0	0	0	16	84	0:11:00
user53	03:37:33	23	125	3	3	15	7	0	0	26	140	0:09:28
user55	00:43:55	11	43	2	1	1	5	4	5	16	44	0:04:00
user57	01:34:39	7	31	1	5	10	3	6	3	18	41	0:13:31
user59	02:33:54	35	131	3	1	9	9	0	0	36	140	0:04:24
user61	00:29:23	5	14	2	0	0	0	0	0	5	14	0:05:53
user63	00:17:51	4	12	1	4	13	2	1	2	9	25	0:04:28
user67	01:55:10	12	52	3	3	29	2	3	5	18	81	0:09:36
user69	00:32:22	9	28	2	0	0	0	1	2	10	28	0:03:36
user75	00:11:50	0	0	0	3	17	2	1	1	4	17	0:00:00
user91	01:09:59	3	26	2	3	14	4	0	0	6	40	0:23:20
Total	**28:31:17**	**284**	**1229**		**40**	**213**		**24**		**348**	**1442**	
Mean	01:25:34	14.20	61.45	2	2.00	10.65	4	1.20	2	17.40	72.10	00:06:57
Median	00:56:57	10.00	37.00	2	1.50	12.00	4	1.00	1	15.00	42.50	00:05:07
Std Dev	01:23:12	14.09	61.79	1	1.81	9.24	3	1.64	2	14.13	62.24	00:05:36
Range	05:12:23	52	194	4	6	29	9	6	6	53	206	00:23:20
Min	00:02:25	0	0	0	0	0	0	0	0	1	1	00:00:00
Max	05:14:48	52	194	4	6	29	9	6	6	54	207	00:23:20

Table 5.9 Study Group – Summary of participants' activity using SQL-Tutor

User ID	Total Sessions Time	Solved			Unsolved			Viewed		Total Problems	Total Attempts	Avg Time per Solved Problem
		Problems	Attempts	Average Initial Rating	Problems	Attempts	Average Initial Rating	Problems	Average Initial Rating			
user14	00:40:05	2	9	1	0	0	0	0	0	2	9	0:20:03
user22	01:02:31	6	39	2	1	9	7	0	0	7	48	0:10:25
user24	01:05:55	24	85	2	1	2	5	0	0	25	87	0:02:45
user26	00:50:26	11	54	2	4	7	3	1	5	16	61	0:04:35
user38	04:21:55	38	171	2	0	0	0	1	2	39	171	0:06:54
user50	00:21:50	0	0	0	1	8	6	0	0	1	8	0:00:00
user58	00:37:29	7	29	3	2	2	7	0	0	9	31	0:05:21
user66	00:10:10	0	0	0	0	0	0	1	2	1	0	0:00:00
user68	01:18:53	31	132	3	3	14	5	0	0	34	146	0:02:33
user70	00:53:30	24	98	3	1	2	4	3	4	28	100	0:02:14
user74	00:28:02	7	31	2	1	2	2	0	0	8	33	0:04:00
user76	01:46:07	4	32	2	1	4	4	1	2	6	36	0:26:32
user78	00:32:56	12	40	3	1	1	7	0	0	13	41	0:02:45
user88	01:56:17	41	151	4	1	1	6	5	7	47	152	0:02:50
user90	01:09:50	8	59	4	1	2	7	0	0	9	61	0:08:44
user92	00:05:42	0	0	0	0	0	0	1	1	1	0	0:00:00
Total	**17:21:38**	**215**	**930**		**18**	**54**		**13**		**246**	**984**	
Mean	01:05:06	13.44	58.13	2	1.13	3.38	4	0.81	1	15.38	61.50	00:06:14
Median	00:51:58	7.50	39.50	2	1.00	2.00	5	0.00	0	9.00	44.50	00:03:25
Std Dev	01:01:00	13.74	54.57	1	1.09	4.05	3	1.38	2	14.26	53.40	00:07:23
Range	04:16:13	41	171	4	4	14	7	5	7	46	171	00:26:32
Min	00:05:42	0	0	0	0	0	0	0	0	1	0	00:00:00
Max	04:21:55	41	171	4	4	14	7	5	7	47	171	00:26:32

The total session time and the number of problems that each participant solved gave a mean time per problem for each of the participants. Excluding any participant that did not solve any problems, the number of participants who had a mean time per solved problem under five minutes was greater for the Study group with 54% (13 out of 16), where it was 47% of the Control group (19 out of 20). Looking at the mean times at group level showed only a forty-three second difference between the groups, with the Control group having a mean time of 00:06:57 and the Study group slightly less at 00:06:14.

There were a lot fewer problems attempted and left unsolved compared to attempted and solved for both groups; the Control group had 40 unsolved problems (*M* = 2, *Mdn* = 1.50) and the Study group had 18 unsolved problems (*M* = 1.13, *Mdn* = 1.00), which was shown by the Mann-Whitney U test to not be significantly different, $U = 119.5$, $z = -1.342$, $p = .240$, $r = -.224$. The Control group had more attempts at the unsolved problems than the Study group, with the ratio of five attempts for every unsolved problem (5:1) for the Control group and a ratio of 3:1 for the Study group. While there was no significant difference in the number of problems attempted and unsolved, there was a significant difference in the number of attempts for the unsolved problems as assessed by the Mann-Whitney U test; Control group (*M* = 10.65, *Mdn* = 12.00) and the Study group (*M* = 3.38, *Mdn* = 4.00), $U = 94$, $z = -2.122$, $p = .036$, $r = -.356$.

The average initial ratings columns in Tables 5.8 and 5.9 relate to the difficultly rating that was already contained in the problem definition files and assigned to each problem purely to provide a consistent starting measure against which the collected data could be analysed. The base version of SQL-Tutor included a default difficulty rating for each problem in the problem definition files used by the system, but they were not used by SQL-Tutor in any way. Instead they were left in the problem definitions from earlier SQL-Tutor versions. Refer to Section 3.1.3 (p.62) for details regarding difficulty levels which are calculated per student by SQL-Tutor. The core log files produced by the system unfortunately did not record the difficulty rating of a problem at the time that the participant selected that problem. This information was recorded in one of the additional log files created for this study, however the log file concerned was only produced for the Study group participants (refer to Sections 6.4 for lessons learnt). SQL-Tutor continually recalculates the difficulty levels for problems based on the participants' learner model, which means that the initial difficulty ratings contained in the problem definition files are not an accurate account of the difficultly of a problem for the participants at the moment they selected it. However they have been included here to provide an approximate view of difficulty levels involved for each group. For the solved problems the average initial ratings as shown in the group statistics (mean, median, standard deviation, range along with minimum and maximum marks) were the same for both groups, which was reflected in the Mann-Whitney U test with no significance between the Study group ($M = 2$, $Mdn = 2$) and the Control group ($M = 2$, $Mdn = 2$), $U = 153$, $z = -.234$, $p = .838$, $r = -.039$. Although the group statistics for the average initial ratings for the unsolved and viewed problems showed slight differences, these differences were not significant when assessed via Mann-Whitney U tests; for ratings of unsolved problems the Study group ($M = 4$, $Mdn = 5$) and the Control group ($M = 4$, $Mdn = 4$), $U = 169.5$, $z = .307$, $p = .765$, $r = .051$; for ratings of the viewed problems the Study group ($M = 1$, $Mdn = 0$) and the Control group ($M = 2$, $Mdn = 1$), $U = 140.5$, $z = -.666$, $p = .539$ $r = -.111$.

As mentioned, the difficulty rating for a problem at the time it was chosen was only recorded in the additional log files created for the Study group participants. Table 5.10 shows the comparison between the average initial difficulty rating and the average rating at the time the problem was chosen. The averages for the participants have largely remained the same between initial and chosen ratings, with four changes for Solved problems (25% of participants), six changes for Unsolved problems (38%) and two changes for Viewed problems (13%). When examining the underlying data at the difficulty rating per problem level, the majority of the ratings remained the same between initial and chosen: 79% for Solved problems, 77% for Unsolved problems and 84% for Viewed problems. Where the initial and chosen ratings differed the majority of the change was for instances where the chosen rating was lower than the initial rating: 31% for Solved problems, 22% for Unsolved problems and 8% for Viewed problems. In Table 5.10 where the number of problems for a participant is one, it can sometimes highlight how different the ratings were. For example, 'user66' viewed one problem and at the time that they chose that problem the difficulty rating was at its maximum difficulty of nine, which was seven higher than the initial rating for that problem. The opposite in ratings is also true; 'user90' attempted, but did not solve a problem that had a difficulty rating at the time the problem was chosen six less than the initial rating. These two examples show the extreme differences, with the main differences between one and three.

Table 5.10 Study Group – Comparison between average initial and chosen difficulty ratings

	Solved			Unsolved			Viewed		
User ID	**Problems**	**Average Initial Rating**	**Average Chosen Rating**	**Problems**	**Average Initial Rating**	**Average Chosen Rating**	**Problems**	**Average Initial Rating**	**Average Chosen Rating**
user14	2	1	1	0	0	0	0	0	0
user22	6	2	4	1	7	3	0	0	0
user24	24	2	2	1	5	2	0	0	0
user26	11	2	2	4	3	3	1	5	5
user38	38	2	2	0	0	0	1	2	2
user50	0	0	0	1	6	9	0	0	0
user58	7	3	3	2	7	7	0	0	0
user66	0	0	0	0	0	0	1	2	9
user68	31	3	3	3	5	5	0	0	0
user70	24	3	2	1	4	2	3	4	2
user74	7	2	2	1	2	2	0	0	0
user76	4	2	4	1	4	5	1	2	2
user78	12	3	3	1	7	7	0	0	0
user88	41	4	4	1	6	6	5	7	7
user90	8	4	3	1	7	1	0	0	0
user92	0	0	0	0	0	0	1	1	1
Total	**215**	**3**	**3**	**18**	**5**	**4**	**13**	**4**	**4**
Mean	13.44	2	2	1.13	4	3	0.81	1	2
Median	7.50	2	2	1.00	5	2	0.00	0	0
Std Dev	13.74	1	1	1.09	3	3	1.38	2	3
Range	41	4	4	4	7	7	5	7	9
Min	0	0	0	0	0	0	0	0	0
Max	41	4	4	4	7	7	5	7	9

The analysis presented in this sub-section has shown differences between the Control and Study groups, however Mann-Whitney U tests have also shown that these differences are not statistically significant, with the exception of the number of attempts for the unsolved problems. For each of the Mann-Whitney U tests the effect size estimates (Pearson's correlation coefficient, *r*-values) have also been included. Effect sizes help to give objective meaning to the practical or theoretical importance of an effect and are independent of sample sizes (Field, 2013; Fritz, Morris and Richler, 2012). The effect size estimates were calculated as follows, where *z* is the *z-score* from the Mann-Whitney U tests and *N* is the total sample size of both groups.

$$r = \frac{z}{\sqrt{N}}$$

A correlation coefficient of 0 equates to no effect through to a perfect negative to positive effect of -1 to 1. Most of the effect size estimates presented in this sub-section fall far below a small effect size of $r = 0.10$, with only the size for unsolved problems ($r = -.224$) approaching, but still short of the medium effect size of $r = 0.30$; using Cohen's effect size guidelines (Field, 2013). Although the effect sizes are small, they do show that differences in performance between the Control and Study groups do exist.

5.3.2 Help Resource Usage

One of the indicators that can be used to help to determine the learner's focus during a learning activity is the use of help resources via their help seeking behaviour. The versions of SQL-Tutor used in this study provided several forms of help ranging from displaying the databases and table structures as an aide memoire through to the help provided after every submission of an attempt to solve a problem (refer to Chapter 3 (p.58) for further details of the forms of help and in which version they were available; either the base version used by the Control group or the modified version used by the Study group). There are four forms of help whose usage is not recorded and therefore cannot be analysed: displaying of the database and table structures, running the participant's SELECT code entered, the session history, and using the 'SQL Help' button to display an external web site (refer to Chapter 6, Sections 6.4 (p.180) and 6.5 (p.183) for lessons learnt and potential further work). The use of domain feedback help levels, extended help feedback and the 'view expected query output' function are analysed here.

As mentioned in further detail in Chapter 3 (p.58), there are six domain feedback help levels as standard in SQL-Tutor: 0 = simple feedback, 1 = error flag, 2 = hint, 3 = partial solution, 4 = list all errors and 5 = complete solution. The numbers are purely used as a reference key in the system and are not shown at all in the user interface. The help level is set by the system to

0/simple feedback at the start of a problem and with each submission of an attempt by the user the help level is incremented by one until level 2/hint is reached. The remaining help levels have to be manually selected by the user, who can also manually select any of the help levels at any point. The way that SQL-Tutor sets the help level after each problem attempt further emphasises the categorisation of the help levels into two categories; levels 0 – 2 provide low- levels of help (known as LLH), with levels 3-5 providing high-levels of help (known as HLH) in terms of the detail that the domain feedback contains. Given these two help categories, the HLH ratio (shown below) was created and reported in Mathews and Mitrovic (2008b) and Mathews (2012).

$$HLH\ ratio = \frac{Number\ of\ high\ level\ attempts}{Total\ number\ of\ attempts}$$

The ratio means the HLH attempts for each participant can be normalised, allowing for the values to be readily compared with each other. For example, a ratio of 1 means the participant selects HLH on every attempt, whereas a ratio of 0.10 indicates the participant select HLH on average 10% of the attempts. A point to note concerning the two references given above, in the versions of SQL-Tutor used in this study the key values that reference the different help levels are zero-based (0 – 5). In the versions used in the cited references, the help levels are exactly the same, but used one-based numbering (1 – 6). While this makes no difference to the outcome of the ratio or any other data recorded in the log files, it could become a point of confusion, so is worth noting.

Tables 5.11, p.146 (Control group) and 5.12, p.146 (Study group) shows the number of LLH and HLH attempts, and the HLH ratio for solved and unsolved problems against each participant. The number of problems and attempts per participant have been included from earlier tables in this chapter for ease of reference and readability of the table data.

There are two instances where participants have a HLH ratio of 0.00 against attempted problems. The instances occurred for solved problems for users 'user11' and 'user45' in the Control group (Table 5.11) and where the participants have only solved one problem and solved it on the first attempt. Unless manually changed, the help level would automatically have been set to 0/simple feedback for the first attempt at the question.

There are five instances where participants have a HLH ratio of 1.00; users 'user29' and 'user55' in the Control group and users 'user58', 'user78' and 'user88' in the Study group. All instances occurred for unsolved problems where the participants had made between 1 – 3 total attempts against 1 or 2 problems. When comparing the HLH ratios for attempts at solved problems for the five participants, only two also had very high ratio values; 'user29' in the Control group had a HLH ratio of 0.75 and 'user78' in the Study group had a HLH ratio of 0.78. The trend of these participants having high HLH ratio values for solved problems was not reflected in the number of problems solved and the number of attempts involved; the Control group user was just above average and the Study group user was below average (refer back to Tables 5.8 and 5.9).

When looking at the HLH ratios at a group level, the Control and Study groups are very similar for unsolved problems, with the mean value for the Study group (0.48) being 3% more than the Control group (0.45). For the solved problems there is only a 1% difference between the groups, with the Control group (0.39) having the higher HLH ratio by only 1% compared to the Study group (0.38). This is also reflected in the total attempts HLH ratio values, with the Control group's (0.45) value 1% higher than the Study group (0.48). The 1% difference is also mirrored in the mean HLH ratios for solved problems (the Control group mean is 0.39, with 0.38 for the Study group) and for the total attempts at both solved and unsolved problems (the Control group mean is .045, with 0.44 for the Study group).

Table 5.11 Control group participants' use of lower- and higher-level help

User ID	Solved Problems	Solved Attempts	Solved LLH	Solved HLH	Solved HLH Ratio	Unsolved Problems	Unsolved Attempts	Unsolved LLH	Unsolved HLH	Unsolved HLH Ratio	Total Attempts HLH Ratio
user11	1	1	1	0	0.00	1	12	6	6	0.50	**0.46**
user15	31	148	40	108	0.73	2	26	2	24	0.92	**0.76**
user25	34	178	122	56	0.31	6	13	11	2	0.15	**0.30**
user27	8	18	14	4	0.22	4	12	7	5	0.42	**0.30**
user29	17	73	18	55	0.75	2	3	0	3	1.00	**0.76**
user39	52	194	80	114	0.59	1	13	6	7	0.54	**0.58**
user43	2	7	5	2	0.29	1	26	14	12	0.46	**0.42**
user45	1	1	1	0	0.00	0	0	0	0	0.00	**0.00**
user49	13	63	30	33	0.52	0	0	0	0	0.00	**0.52**
user51	16	84	51	33	0.39	0	0	0	0	0.00	**0.39**
user53	23	125	68	57	0.46	3	15	3	12	0.80	**0.49**
user55	11	43	32	11	0.26	1	1	0	1	1.00	**0.27**
user57	7	31	17	14	0.45	5	10	4	6	0.60	**0.49**
user59	35	131	92	39	0.30	1	9	2	7	0.78	**0.33**
user61	5	14	6	8	0.57	0	0	0	0	0.00	**0.57**
user63	4	12	4	8	0.67	4	13	10	3	0.23	**0.44**
user67	12	52	26	26	0.50	3	29	19	10	0.34	**0.44**
user69	9	28	16	12	0.43	0	0	0	0	0.00	**0.43**
user75	0	0	0	0	0.00	3	17	7	10	0.59	**0.59**
user91	3	26	16	10	0.38	3	14	5	9	0.64	**0.48**
Total	**284**	**1229**	**639**	**590**		**40**	**213**	**96**	**117**		
Mean	14.20	61.45	31.95	29.50	0.39	2.00	10.65	4.80	5.85	0.45	0.45
Median	10.00	37.00	17.50	13.00	0.41	1.50	12.00	3.50	5.50	0.48	0.45
Std Dev	14.09	61.79	34.15	33.83	0.23	1.81	9.24	5.36	5.98	0.35	0.17
Range	52	194	122	114	0.75	6	29	19	24	1.00	0.76
Min	0	0	0	0	0.00	0	0	0	0	0.00	0.00
Max	52	194	122	114	0.75	6	29	19	24	1.00	0.76

Table 5.12 Study group participants' use of lower- and higher-level help

User ID	Solved Problems	Solved Attempts	Solved LLH	Solved HLH	Solved HLH Ratio	Unsolved Problems	Unsolved Attempts	Unsolved LLH	Unsolved HLH	Unsolved HLH Ratio	Total Attempts HLH Ratio
user14	2	9	4	5	0.56	0	0	0	0	0.00	**0.56**
user22	6	39	30	9	0.23	1	9	5	4	0.44	**0.27**
user24	24	85	41	44	0.52	1	2	1	1	0.50	**0.52**
user26	11	54	28	26	0.48	4	7	3	4	0.57	**0.49**
user38	38	171	96	75	0.44	0	0	0	0	0.00	**0.44**
user50	0	0	0	0	0.00	1	8	2	6	0.75	**0.75**
user58	7	29	24	5	0.17	2	2	0	2	1.00	**0.23**
user66	0	0	0	0	0.00	0	0	0	0	0.00	**0.00**
user68	31	132	69	63	0.48	3	14	5	9	0.64	**0.49**
user70	24	98	20	78	0.80	1	2	1	1	0.50	**0.79**
user74	7	31	12	19	0.61	1	2	1	1	0.50	**0.61**
user76	4	32	21	11	0.34	1	4	3	1	0.25	**0.33**
user78	12	40	9	31	0.78	1	1	0	1	1.00	**0.78**
user88	41	151	103	48	0.32	1	1	0	1	1.00	**0.32**
user90	8	59	36	23	0.39	1	2	1	1	0.50	**0.39**
user92	0	0	0	0	0.00	0	0	0	0	0.00	**0.00**
Total	**215**	**930**	**493**	**437**		**18**	**54**	**22**	**32**		
Mean	13.44	58.13	30.81	27.31	0.38	1.13	3.38	1.38	2.00	0.48	0.44
Median	7.50	39.50	22.50	21.00	0.41	1.00	2.00	1.00	1.00	0.50	0.47
Std Dev	13.74	54.57	32.37	26.78	0.25	1.09	4.05	1.75	2.53	0.36	0.24
Range	41	171	103	78	0.80	4	14	5	9	0.25	0.78
Min	0	0	0	0	0.00	0	0	0	0	0.00	0.00
Max	41	171	103	78	0.80	4	14	5	9	1.00	0.78

Figure 5.4 shows the frequency ranges of values from the 'Total Attempts HLH Ratio' columns in Tables 5.11 and 5.12. Even though the group level ratio values are quite similar, with only a 1% difference, the frequency chart shows some interesting distribution similarities and differences between the Control and Study groups. The highest HLH ratio range for both groups is between 0.76 and 0.85, with both groups having two participants in that range. The lowest HLH ratio range of between 0.00 and 0.05 has participants from both groups, but the Control group has one participant against the two from the Study group. While the Study group has a wider spread of ratio values across the ranges, the Control group is more concentrated in the middle of the populated ranges, with HLH ratio values clustered between 0.26 and 0.65.

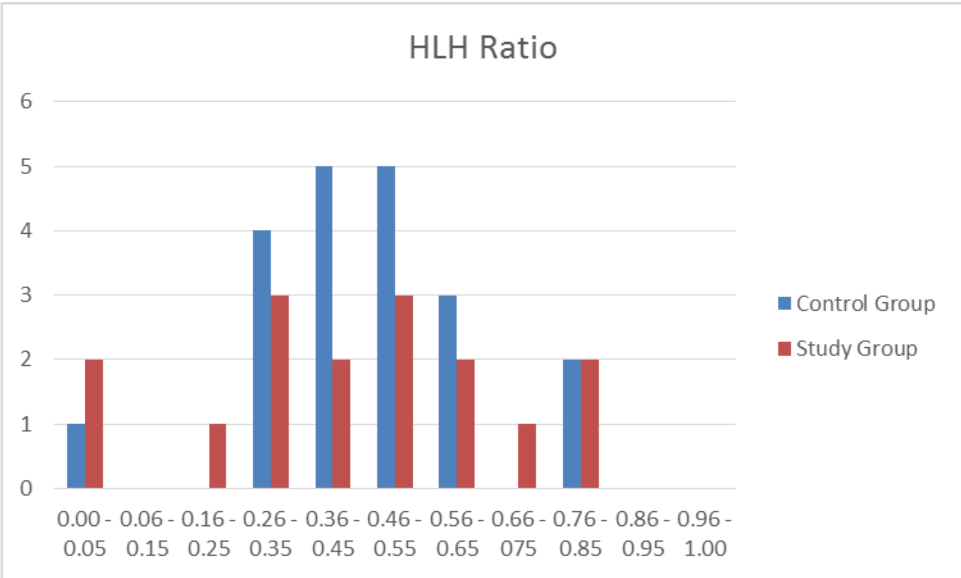


Figure 5.4 Frequency distribution of high-level help

The problem attempts with the corresponding help levels were analysed to determine how many times HLH was manually selected on the first attempt at problems. For this analysis only the partial and complete solution help levels were examined, as they are the two HLH options that explicitly present the part or full solution to the participant. The third HLH 'list all errors' option provides a lot of domain feedback, but does not include the solution, whether part or full. The 'partial solution' help level was selected three times (two for solved and one for unsolved problems) by participants in the Control group, which was 1% of the problems attempted (solved and unsolved). The Study group has seven (four for solved and three for unsolved problems) instances of selecting the 'partial solution', which equates to 3% of the problems attempted. Selecting the 'complete solution' occurred with a higher frequency than selecting the 'partial solution' help level. The highest help level of 'complete solution' was selected thirty-seven times (twenty-eight for solved problems and nine for unsolved problems) for the Control group, which is 11% of all the problems attempted. The same help level was selected twenty-five times (nineteen for solved and six for unsolved problems) for the Study group, which is 11% of all the problems attempted and the same as the Control group.

From analysing the selection of HLH on the first attempts, the same was also analysed but for instances where HLH was selected in any of the first three attempts at a problem. The first three attempts was chosen as it represents the number of attempts where SQL-Tutor will automatically increment the help level. The Control group had HLH selected in any of the first three attempts for one-hundred and forty-seven problems, which is 45% of all problems attempted (solved and unsolved). This was broken down further with one-hundred and nineteen (42%) solved problems and twenty-eight (70%) unsolved problems. The percentages are slightly increased for the Study group with one-hundred and fourteen problems, which is 49% of all problems attempted. This represents ninety-eight (46%) solved problems and sixteen (89%) unsolved problems.

The attempts and the corresponding help levels were also analysed to determine if any problems had been attempted without any HLH being selected for all the attempts at a problem. The Control group solved one-hundred and nine (38% of solved problems) and attempted five (13%) unsolved problems. This indicates that 35% of all the problems attempted (solved and unsolved) were done without using any form of HLH. The Study group solved seventy-six (35%) problems, which represents 33% of all the problems attempted (solved and unsolved). The Study group did not attempt any unsolved problem without selecting HLH at least once in their attempts.

One of the new elements of functionality that was included into SQL-Tutor was the ability of a participant to view the query output expected from a correct answer for the problem as a form of help (refer to Section 3.2.2 (p.67) for further details). This functionality was made available in the versions of SQL-Tutor used by the Control and Study groups. The expected output was displayed in a new pop-up web page and, because it was the only way to see the expected output, this option had to always be chosen by clicking on the button provided. This meant that it could reliably be recorded in the SQL-Tutor log files every time this option was chosen. From the Control group 85% (17) of participants used this form of help, with 75% (12) of participants from the Study group. It was selected one-hundred and sixty-four times (11% of all attempts; both for solved and unsolved problems) by Control group participants and it was selected one-hundred and sixty-three times (17% of all attempts) by the Study group participants. Although the actual percentages of the stages within working on a problem a participant selected to view the expected query output differed slightly between the two groups, the rank order of the three stages involved (start of a problem, during a problem and the end of a problem) were the same. The Control group selected the functionality 69% of the time at the beginning of a problem (prior to making any attempts to answer the problem), then 29% of the time was during a problem (after at least one attempt has been submitted) and finally 2% of the time was at the end of a problem (the participant has chosen to close the main problem page in SQL-Tutor). The Study group mirrored this, with the most selections (76%) made at the beginning of a problem, 18% of the selections made during a problem and 6% at the end of a problem.

5.3.2.1 Motivational and Metacognitive Feedback

While the Control group used a version of SQL-Tutor that only gave domain feedback, the version used by the Study group provided participants with additional feedback. Such feedback linked the participants' motivation and metacognitive states with their current learning state and their last self-report of their self-efficacy (e.g. the feedback was influenced by the number of attempts at a problem and how they had last rated their self-efficacy). The inclusion of such feedback is core to the focus of this research project of aiming to improve the participant's focus and experience of problem solving via motivational and metacognitive feedback based on the participants past states and experiences. Refer to Section 3.3 (p.71) for further details on the rules engine formulating this additional feedback. From examining the additional log files that were created by the version of SQL-Tutor used by the Study group only, there are instances where the participant did react to suggestions contained in the additional feedback received. The direct suggestions that were provided by the feedback revolved around reminding the participant of the various sources of help available in SQL-Tutor and asking the participant to either try *not* to look at the partial/full solution or suggesting that the participant *does* look at the solution in order to study and learn from it (the latter suggestion was only made after multiple unsuccessful attempts had been made). Unfortunately where there are instances that the participant reacted (approximately 30%) to the suggestions by either changing the help level on the next attempt or by using the functionality to view the expected output, it is unknown whether such actions were as a direct result of feedback. The reason is that the same behaviour patterns can also be witnessed in the participants' interaction with SQL-Tutor without the additional feedback. So it is difficult to determine whether a reaction was as result of receiving targeted feedback or was a natural reaction to the situation. The percentage quoted here is approximate due to a processing issue that was found late in the software that was developed to extract the data from the log file (refer to Section 6.4 (p.180) for lessons learnt). In the reactions seen from analysing the log files that involved the participant changing the help level on the next attempt of a problem, it was interesting to see that the help levels were not just increased, but also decreased. For example, when 'user22' was reminded of the various sources of help available, the level of help selected on the next attempt moved from 2/hint to 1/error flag. Similarly the same user increased the help level from 3/partial solution to 5/complete solution after receiving the suggestion "How about selecting the help level to look at the 'full' solution?" as part of the feedback given after the eighth attempt at a problem. In the case of the first example where the help level was decreased by the participant, it is difficult to determine whether this was as a direct reaction to the feedback received, especially as a related problem had also been found and the participant had also viewed it, or a reaction to SQL-Tutor having automatically increased the help level to 2/hint from the previous attempt. The same holds true for the second example where the help level was increased, it could have been a direct response to the feedback given or it could have been the participant choosing to view the complete solution after having viewed the partial solution in the previous two attempts at the problem.

5.3.3 Learning Curves

Learning curves are graphical representations of learning over a period of time and help to convey the speed of learning, along with the type (e.g. shallow or deep learning). Learning any new skill tends to follow a power law, "where the greatest improvement occurs early in the learning process" (Mathews and Mitrovic, 2008b). The research group from where SQL-Tutor originates (Intelligent Computer Tutoring Group (ICTG) at the University of Canterbury, Christchurch, New Zealand) had developed a Lisp-based data analysis tool that calculates learning curves, which have been reported in past studies involving SQL-Tutor (Martin, Mitrovic, Koedinger and Mathan, 2011; Mathews and Mitrovic, 2008b). The learning curves for SQL-Tutor are based on the occasions (x-axis) when constraints are used and the error rates (y-axis) experienced against the constraints., as explained in the quotation below by Mitrovic (2014) in an email to the thesis author.

> *"An occasion is when the constraint is used. Therefore we plot the first, second, third etc time when they used constraints. A learning curve for a particular student is averaged over all constraints that student used. Please note that the first occasion then means the first time they used any constraint - and therefore it is not related to problems. When you look at 339 constraints for the first occasion, that would be over all the problems that user15 attempted. The student has used 339 constraints overall, and the other number tells you the proportion of constraints he/she violated. We normally do not report learning curves for individual students, but average over all constraints and all students."*

Figure 5.5 shows the compound learning curves for the Control and Study groups. The learning curves for both groups are very similar in both slope and fit. The slopes are quite steep, suggesting that the initial learning occurred quite quickly in terms of the number of occasions, with the greatest reduction in error rates occurring in the first three to four occasions. These results are in-line with previous evaluation studies using SQL-Tutor that found "that students acquire constraints at a high rate" (Mitrovic and ICTG.Team, 2008). The R^2 values for both groups are over 0.8 which indicates that the power lines are a reliable, good fit for the data.

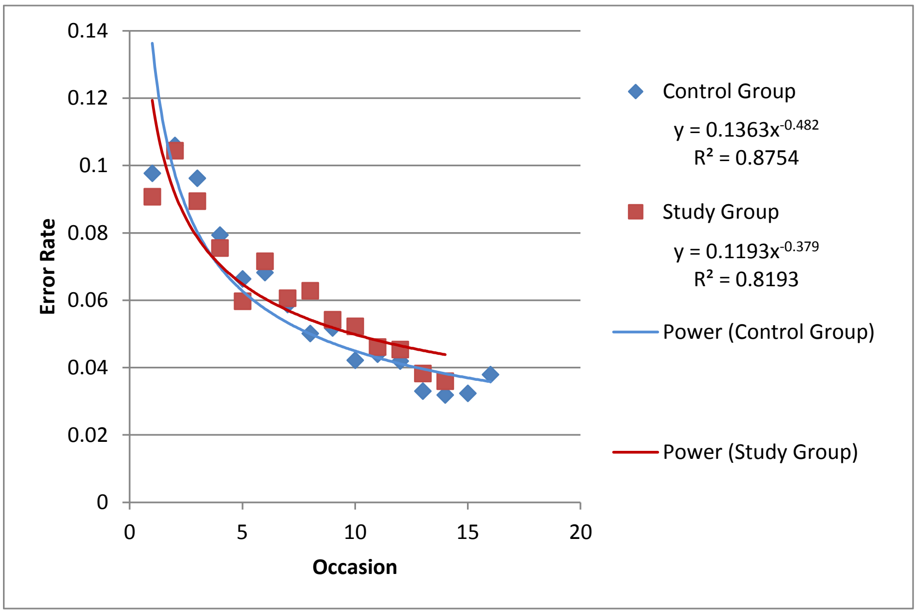


Figure 5.5 Learning curves showing power trend lines for Control and Study groups

With the learning curves for the groups being very similar, the individual learning curves for the participants were analysed. Unfortunately the R^2 values were low which indicates that the power lines were not a good fit against the individual learning curves (ranging between 0.056 – 0.867 for the Control group and 0.159 – 0.798 for the Study group).

5.3.4 Past Problems and Self-Reports

In Section 5.3.2 (p.143) there has already been analysis of two areas of data that were specific to the Study group only: recording the difficulty rating of a problem at the time the participant selected it and the additional feedback (motivational and metacognitive) that the participants received in SQL-Tutor. There are two other areas of data collection based on functionality that were available only to the Study group: receiving feedback linking the current problem to similar problems the participant had previously worked on, and the self-report slider interface to record the participant's self-efficacy (refer to Section 3.3 (p.71) for further details of the functionality mentioned). They were both core to the focus of this research project; explicitly relating the current problem to a past problem the participant had worked on was a key part of the metacognitive feedback and using the slider interface control to obtain the participant's feelings of self-efficacy helped to form the motivational and metacognitive statements included in the feedback.

5.3.4.1 Past problems

One of the core functionalities that was introduced in the SQL-Tutor version that the Study group used was the ability of the system to link the current problem to similar problems that the participant had worked on. The problems were linked at a SELECT query clause-level (e.g. SELECT, FROM, WHERE, and so on) and also at the problem-level. If any related problems were found after the participant had submitted two incorrect attempts at a problem, a link to the first related problem found would be displayed to the participant for them to select and review. Refer to Section 3.3.2.1 (p.88) for further details.

Table 5.13 shows for each user in the Study group the number of past problems (problem- and clause-level) that were linked to the additional system feedback provided. The table also shows how many times the participants used this functionality by selecting to view the related past-problem.

Table 5.13 Occurrences of past problems being referenced
and shown to the Study group participants

	Past Related Problems			
User ID	**Problems**	**Clauses**	**Total**	**Viewed**
user14	0	0	**0**	**0**
user22	1	3	**4**	**2**
user24	9	7	**16**	**0**
user26	1	2	**3**	**0**
user38	11	10	**21**	**2**
user50	0	0	**0**	**0**
user58	3	2	**5**	**0**
user66	0	0	**0**	**0**
user68	16	20	**36**	**0**
user70	3	3	**6**	**0**
user74	1	0	**1**	**0**
user76	0	0	**0**	**0**
user78	3	3	**6**	**0**
user88	7	8	**15**	**0**
user90	4	2	**6**	**0**
user92	0	0	**0**	**0**
Total	59	60	**119**	**4**

Where related past solved problems were not found for some participants, this can be clearly explained with one of two reasons: the participant had only viewed problems and the participant had only attempted a few problems. Users 'user14' and 'user92' were the two users in the Study group who did not attempt to solve any problems, but merely viewed some problems. The other three users ('user14', 'user50' and 'user76') only attempted a few problems (between one and five combined solved and unsolved), so there were very few problems with which to find related problems with.

The remaining 69% of Study group participants received feedback from SQL-Tutor that included links to past problems. Unfortunately the system log files only recorded four instances where the participant selected the link and viewed the past problem. Disappointingly this meant that the past related problems were only selected 3% of the times that they were presented to the participants. What is not known is whether just the feedback containing the link, which also referenced the problem number and database name, was enough of a metacognitive aid for the participant to progress (refer to Figure 3.13a (p.82) for example of the user interface and the feedback given). Similarly it is also unknown whether the participants read the additional feedback provided. While the number of views of past related problem was disappointing, refer to Section 6.5 (p.183) where this is discussed further.

5.3.4.2 Self-Reports

Another core functionality that was introduced in the SQL-Tutor version developed for the Study group was the ability for the learner to report their self-efficacy at key stages of a session: beginning of a session, at the end of some problems and at the end of the session. Refer to Section 3.3.2 (p.78) for further details.

As the Milestone 1 summative assessment was taken by the participants a short time before the start of this study and before they started using SQL-Tutor, the marks for this assessment were compared to the first and second self-reports of self-efficacy that the participants recorded. This was used as an initial check to help to determine the participants' accuracy of self-report. The first self-report was used as it was the first report at the start of the first session. Taking into account that the participants' reports of self-efficacy may also have been affected by it being the start of the first session of using SQL-Tutor which they had only seen and not used before, the second self-report was also considered, as this would have been after the participants had at least viewed or attempted a few problems. Apart from two participants, the reported self-efficacy levels remained the same until after the third report.

The self-reports were recorded in the additional log files for the Study group as 'Low', 'Middle' and 'High'. In order to compare the Milestone 1 marks directly to the self-report, the marks were divided into ranges following the marking scheme for a fail and a first class: 0 – 39 = 'Low', 40 – 69 = 'Middle' and 70 – 100 = 'High'. Out of the sixteen participants in the Study group, there were no self-reports recorded for two of them (refer to Section 6.4 for lessons learnt), but the percentages used in this paragraph still remain for the whole Study group (16 participants) so that they can be used in comparisons. From the remaining participants, four (25%) had marks in the 'Middle', with the remaining ten (62%) in the 'High' range. From the first self-reports, nine (57%) reported 'Low' self-efficacy, four (25%) reported 'Middle' and one (6%) reported 'High'. This was quite a surprise, as it meant that no one was optimistic in their reports of self-efficacy, with only one (6%) participant accurate when comparing their self-report to their Milestone 1 mark (this was the participant who had 'High' mark range and self-efficacy). The remaining thirteen (81%) were pessimistic when reporting their self-efficacy. While it is feasible to consider that the participants gave pessimistic self-reports because they may have thought that SQL-Tutor would compare them to their actual achievements, it is highly unlikely as the Study group participants were never told that the system would use their self-reports. In terms of the second self-reports, only two participants changed their level of self-efficacy from the first self-report. They both changed from 'Low' to 'Middle'. This did not have any effect on the comparison between self-report and Milestone 1 marks, as both participants had achieved marks in the 'High' range, so their self-reports were still pessimistic.

When considering the participants' self-reports throughout all of their time using SQL-Tutor, the self-reporting accuracy changed only slightly from the starting figures outlined above. As discussed in the literature review in Chapter 2 (p.6), there are many sources that influence a person's feelings of self-efficacy, one of which is 'performance accomplishments' (Bandura, 1977). Performance (the ratio of problems solved against the attempts) was used as a basic measure in the version of SQL-Tutor used by the Study group to help determine the accuracy of the participants' reports of self-efficacy. Table 5.14 shows the comparison between the participants' average self-efficacy report and their performance.

There are no results for three participants as they did not solve any problems. The percentages of participants given in this paragraph are for the Study group as a whole, so they can be compared to the percentages already mentioned regarding the initial accuracy of self-reports. As can be seen from Table 5.14, four (25%) participants were accurate in their reports of self-efficacy, while three (19%) are optimistic and the majority of six (38%) pessimistic in reporting their self-efficacy beliefs.

Table 5.14 Comparison of participants' average self-efficacy against performance

	Solved			
User ID	**Attempts to Problems Ratio**	**Performance Level ***	**Average Self-Efficacy Level**	**Accuracy**
user14	5:1	Middle	Middle	Accurate
user22	7:1	Low	Middle	Optimistic
user24	4:1	Middle	Middle	Accurate
user26	5:1	Middle	Low	Pessimistic
user38	5:1	Middle	Low	Pessimistic
user50	-	-	-	-
user58	4:1	Middle	Low	Pessimistic
user66	-	-	-	-
user68	4:1	Middle	Low	Pessimistic
user70	4:1	Middle	Low	Pessimistic
user74	4:1	Middle	Middle	Accurate
user76	8:1	Low	Low	Accurate
user78	3:1	High	Low	Pessimistic
user88	4:1	Middle	High	Optimistic
user90	7:1	Low	Middle	Optimistic
user92	-	-	-	-

* >5:1 = Low, > 3:1 & <=5:1 = Middle, <=3:1 = High

One issue with a web interface (not just limited to SQL-Tutor) is that a user can skip web pages via the navigation buttons of the web browser itself (e.g. back and forward buttons), as well as being able to close the web page. These issues can be controlled to an extent by placing additional code within the web pages and this is a point which is discussed further in Chapter 6 (p.172). Unfortunately the recording of self-reports was affected by this very issue, with participants sometimes skipping over the self-report web pages and, even more frequently, not logging out of SQL-Tutor correctly by using the 'Logout' button. Not logging out of the system using the 'Logout' button meant that the majority of users did not complete the end of session self-report. Seven out of the sixteen participants (44%) did log out correctly and complete end of session self-reports, with the remaining nine (56%) participants who never logged out of SQL-Tutor correctly. Out of the seven participants who did log out correctly, only three of them did so every time. The rest logged out correctly only 19% of the time. This had an effect on the motivational and metacognitive feedback they would have received at the start of the next session.

5.4 Discussion of Results

The section discusses the results given above in direct relation to the three supporting research questions presented in Chapter 1 (p.1). The discussion of the primary research question is presented in Chapter 6 (p.172). The "feedback" mentioned in the following three research questions refer to motivational and metacognitive feedback based on the learner's past states and experiences.

Does providing such feedback lead to any measurable learning gains?

Measuring learning gains typically involves measuring the achievements or the ability of the learner to solve particular problems, for example, by comparing pre- and post-tests. Given the structure and content of this study, the summative assessments marks for the database module were used as part of measure of learning gains, with Milestone 1 being the pre-test and the Exam marks being the post-test (refer to Table 5.2, p.108). This study was designed to use the module summative assessments as pre- and post-tests, as opposed to creating dedicated tests, as the study was embedded within an existing university course with limitations on what extra testing of the students was possible.

Examining the means of the marks for Milestone 1 showed the Control group was just slightly ahead of the Study group, with means of 77.44 and 75.37 respectively. The Mann-Whitney U test against the Control and Study groups, showed that there was no significant difference in the

Milestone 1 marks, which meant that the study had an equal starting point as pre-test measures. The Control group showed a decline in marks as the summative assessments progressed, whereas the Study group fluctuated up and down across the assessments. Despite this, the marks of both groups decreased from pre-test Milestone 1 to post-test Exam; the Control group had the largest decrease of 21.19, compared to 12.87 for the Study group. Looking at the marks for all four assessments, there is a distinct pattern in the marks, which shows a grouping of Milestones 1 and 2, then another grouping of Milestone 3 and the Exam. The means of the marks for Milestones 1 and 2 are very similar, with a decrease of 5.14 for the Control group and a small increase of 0.21 for the Study group. However, there are then large decreases between Milestones 2 and 3, with 14.80 for the Control group and 18.29 for the Study group. Then between Milestone 3 and the Exam the level of movement reverts to being small, with 1.25 decrease for the Control group and a 5.21 increase for the Study group. Refer to Figure 5.2 (p.116) for the graphs showing the range of marks per assessment for the groups.

It is important to note that these assessments and their marking were not under experimental control and the low marks in the Exam compared to Milestone 1 (say) do not necessarily reflect a decline in understanding so much as the fact that the these assessments were of different degrees of difficulty, with exams and assessments generally assessing different skills/abilities to reflect the students' progress through the subject being taught, and gathered under different conditions and also marked more or less scrupulously.

Taking into account the academic calendar and the study duration (both planned and actual SQL-Tutor usage) the sharp decline occurring between Milestones 2 and 3 could also be influenced by the timing of the assessments (refer to Appendix A for the study timeline against the academic year). While Milestone 1 occurred prior to the start of the study, the participants were still covering SQL SELECT queries in the lectures and computer lab sessions. This is the same for Milestone 2 which occurred in the final teaching week of semester one, six weeks after Milestone 2. Not only were SELECT queries still being covered in class at that stage, from looking at the weeks during which the participants used SQL-Tutor (refer to Appendix A, p.200), it was also the last week that the system was used at the end of a block of four weeks (with the exception of one participant logging onto SQL-Tutor one more time four weeks later). Milestones 1 and 2 form the first grouping in terms of small amount of changes in marks. There is then a gap of eleven weeks before Milestone 3 occurred and in the meantime there had been a five week gap in teaching for the winter break and assessments, before semester two teaching started. Apart from within the context of creating views, SELECT queries were not taught in semester two. This means that the eleven weeks prior to Milestone 3 both SQL-Tutor was not used by participants (with the one exception previously mentioned) and the module content did not explicitly cover SELECT queries. This gap could explain the clear line that the results put in the assessments between a group Milestones 1 and 3, followed by Milestone 3 and Exam.

Taking into account that the assessments and their marking were not done under experimental control, the marks from Milestone 1 and the Exam were ranked in order to determine any differences between the two groups (refer to Tables 5.3 (p.119) and 5.4 (p.119) for the ranking data). While the movement in ranking between Milestone 1 and the Exam reflected the general movement in means of the marks, the mean change of rank of a Study group participant was between 3 and 4 places up the ranking, whereas the mean rank in the Control group stayed about the same. This indicates a slight trend in favour of the Study group in terms of learning gains.

Analysing the assessment results for each group did show some significant differences when the Friedman tests and pairwise comparisons were conducted (refer to Section 5.2.1, p.106). For example, there was a significant difference between the Milestones 1 and 3 marks for the Control group and Milestones 2 and 3 for the Study group. The linear regression lines also showed that there were positive correlations between the Milestone 1 marks and the other assessment marks per group (except between Milestone 1 and the Exam for the Control group). However, the R^2 values showed that the relationships were quite small, which means that the Milestone 1 marks cannot be used as an accurate predictor of the other assessment marks. A comparison of the means and medians for the assessments did show that the distribution for the Study group tended to have a more negative skew than the Control group, which indicated that the Study group tended to have a cluster of marks in the higher mark ranges and a wide spread of marks across the lower ranges, whereas the reverse tended to be the case for the Control group.

In order to determine if there were any differences between the groups for each assessment Kruskal-Wallis tests were conducted. While there were small differences, they were not significantly different.

Another aspect of measuring learning gains is to analyse the learner's performance, of which the previously discussed pre- and post-test results are part. The learner's performance while using SQL-Tutor is discussed here by focussing on their progress in SQL-Tutor in terms of the problems solved, the attempts taken and time spent (refer to Section 5.3, p.135). Given the differences in group sizes the mean number of problems solved per participant were compared. The Control group solved only 0.76 more of a problem than the Study group, but they took 3.32 more attempts to solve a problem (refer to Tables 5.8 and 5.9 (p.139) for the base data). The calculation used inside SQL-Tutor to determine the level of performance was a percentage of the problems solved against the attempts taken. Despite the Control group having taken more attempts to solve a problem than the Study group, both groups had a slightly low performance calculation of 23% of problems solved against attempts taken. The average time to solve a problem was similar for the two groups, with the Study group taking forty-three seconds less to solve a problem than the Control group. The Study group did have the most participants who took on average less than five-minutes to solve a problem: 54% for the Study group and 47% for the Control group.

Does providing such feedback lead to any measurable gains in learner focus?

As mentioned in Chapter 1 (p.1) *learner focus* refers to the participant's focus on using SQL-Tutor to help them to practice and develop their SQL SELECT capability. Areas examined to determine learner focus were effort, persistence and learner behaviour.

As mentioned in Section 3.3.2.2 (p.90) effort in the system was calculated as total attempts divided by the number of problems attempted (e.g. the number of attempts per problem, both solved and unsolved), with anything less than three considered poor effort. Although SQL-Tutor only performed this calculation for the Study group version, the effort for both groups was calculated using the data presented in Tables 5.8 and 5.9 (p.139). On this basis, the Control group had 65% of participants who showed good levels of effort and the Study group were very close with 63% of participants with good levels of effort.

Closely entwined with effort there is persistence, with the general notion that persistence is the continued effort when the things become difficult. Taking this and the calculation that was used in this study for effort, the ratio of the number of attempts per *unsolved* problem was used as a measure of persistence. For unsolved problems, the ratios are different for the groups with a ratio of 5:1 for the Control group and 3:1 for the Study group, which a Mann-Whitney U test confirmed as being significantly different (p = .036). When looking at solved problems, the number of attempts divided by the problems solved gives a ratio of 4:1 for both the Control and Study groups. So the Control group appeared to be more persistent in that they made nearly double the number of attempts at problems before giving up on a problem, and both groups were the same in the effort focussed on the solved problems.

Another indication of persistence that is related to the unsolved problems attempted, is the difficulty rating of the problems (refer to Tables 5.8 and 5.9, p.139). The means of the initial ratings for the unsolved problems were the same at 4 for both the Control and Study groups. As a point of interest, the mean difficulty rating for solved problems, although they were the same for both groups, they were lower, at 2, than the unsolved problems. This would be expected given that it was the higher mean difficulty ratings for problems that remained unsolved.

One area that could show effort and/or persistence is the period of time over which the participants using SQL-Tutor. The figure in Appendix A (p.200) shows the timeline for the study against the academic year. As already discussed in relation to the first supporting research question was the fact that the participants' use of SQL-Tutor appeared to be tightly correlated with the module content and delivery schedule. All participants used SQL-Tutor over a four week period at the end of semester one in December 2012, with the exception of one Control group participant who logged in and actively used SQL-Tutor for forty-eight minutes in January 2013.

Taking into consideration the sources of data available from this study for both the Control and Study groups, the discussion of learning behaviour focuses on the participants' use of help within SQL-Tutor to help to identify the learners' focus on the task of using SQL-Tutor.

As mentioned in Section 5.3.2 (p.143), the HLH (high level help) ratio was used to determine the participant's use of HLH when working on the problems (solved and unsolved). HLH included the participant looking at the partial and full problem solutions. When looking at the means for all problems (solved and unsolved) there was only 1% difference with an HLH ratio of 45% for Control group and 44% for Study group. The frequency of HLH ratios showed that the Study group had a wider spread of HLH ratios, where the Control group was more clustered in the middle ranges. This implies that when the Control group participants used HLH they did so consistently between 26% and 65% of the time, whereas when the Study group participants used HLH their levels were not as consistent with a wider spread of percentages. When working on solved problems both groups used LLH (low level help) more than HLH, but the reverse was true for both groups when attempting unsolved problems where more HLH was requested. This would tend to be consistent with not only the higher means of difficulty ratings of unsolved problems, but also with the fact that the problems remained unsolved.

The ability to view the expected output for a problem was a new form of help introduced in the versions of the tutor used by both groups. It was used consistently by the Control and Study groups, with the most usage at the beginning of a problem to help clarify the problem requirements (69% for the Control group and 76% for the Study group).

Do the learners perceive any benefit from using an ITS to aid their learning?

The one source of data collected as part of this study that directly elicited the participants' perceptions of using SQL-Tutor to aid their learning was the post-study questionnaire (refer to

Appendix F for the actual questionnaire and Section 5.2.2 (p.120) for the results analysis). Section A of the questionnaire consisted of nine closed questions (referred to as Statements) and Section B contained six open questions (referred to as Questions).

Although the post-study questionnaire was designed to be used immediately at the end of the study, and indeed that is what happened, as already discussed in this section, the participants stopped using the tutor in-line with when SELECT queries stopped being taught as part of the module delivery. This meant that there was a gap of twelve weeks from when all but one participant last used SQL-Tutor and the post-study questionnaire was available online. Given the amount of time that elapsed, the participants appeared to have given considered responses, although it is completely unknown whether this elapsed time influenced any of the responses given. The elapsed time may also explain the 67% completion rate for the questionnaire, although some non-completions would generally be expected from any questionnaire that was not issued and completed under controlled conditions (e.g. within a classroom session). The usage pattern of SQL-Tutor in-line with the module contents delivery could also be an indicator to the participants' perceived benefit of using the tutor, in so far as they used the tutor from the time it was available through to the end of the teaching in semester one (four weeks). When the module moved onto other content, SQL-Tutor was not used. So the participants used the tutor while the module content was directly covering SELECT queries, which could indicate, by default, that the four weeks where the participants used the tutor, they found it of some benefit. This is further strengthened by the participants not using SQL-Tutor once the module content had moved on, in that they viewed the use of the tutor as primarily a tool to aid their learning and not just a research tool for the author's benefit.

Section A of the questionnaire contained nine statements arranged on a five-point Likert scale. The nine statements were concerned with the participant's perception of how SQL-Tutor helped them to learn, with the statements divided into five areas: domain-related, learning process, metacognition, motivation and general tutor usage. Pearson's chi-square and Fisher's exact tests confirmed that there was no significant differences between responses from the Control and

Study groups, although this would not be expected except for those two statements that were explicitly focussed on the past problems and self-report of the version used by the Study group.

The domain-related area was concerned with the participants' perception of SQL-Tutor helping them to learn SELECT queries (Statement 1, p.121) and database concepts (Statement 2, p.122). The majority of both groups responded positively to both of these statements, with 92% and 75% of the Study group agreeing/strongly agreeing with each of the statement respectively, and 83% and 64% of the Control group agreeing/strongly agreeing. While the stronger agreement was for SQL-Tutor helping to learn SELECT queries, both groups still strongly perceived that SQL-Tutor helped them to strengthen their knowledge of database concepts via learning by doing.

The learning process area covered SQL-Tutor helping the participants to identify their mistakes (Statement 3, p.122) and overcome any difficulties (Statement 4, p.125). Again the majority of both groups responded positively (agreeing/strongly agreeing), 75% and 67% for the Study group and 67% and 67% for the Control group. The responses for these two statements were the only statements where, not only was the majority response the same for both groups, but they also had the same percentages: 42% agreed about identifying mistakes and 50% agreed about overcoming difficulties. The Control group had the most neutral/negative responses with 33% for both statements, as opposed to 25% and 33% (only neutral) for the Study group.

Both Statements 3 and 4 imply that the way SQL-Tutor helped was via the feedback it gave, so the responses to three of the questions in Section B of the questionnaire are also relevant. Question 1 asked what the participants liked most about SQL-Tutor and 54% of the responses said it was the feedback. This was also echoed in the responses to Question 3 which asked what they thought of the feedback comments; 79% of the responses to this question were positive, with 58% of these purely positive comments and the rest positive, but with constructive critique (e.g. "they were helpful, although sometimes they tend to be vague"). The 21% who did not like the feedback ranged from not agreeing with it to it could have been worded better. Question 4 asked the participants if they wanted to comment on any other aspect of SQL-Tutor and 8% who responded to this question mentioned feedback in terms of it needing to be exact. The perceived helpfulness of the feedback from SQL-Tutor could be linked to the level of domain knowledge and the amount of use of SQL-Tutor, or even effort. The participants who replied quite negatively about the feedback received also tended to spend little time logged into SQL-Tutor, with below average number of problems solved and in all, but one case, the participants obtained summative assessment marks which were in the low to mid mark ranges. Rather than these responses being negative, they raise an interesting point about how multifaceted the influences on perception are.

The area of metacognition covered SQL-Tutor helping the participants to solve problems by relating to previous problems (Statement 5, p.123) and SQL-Tutor helping participants to gain an accurate assessment of what they understood and did not understand (Statement 6, p.125). The responses were again positive for both groups with the majority agreeing, with 67% and 58% agreeing from the Study group, and 50% and 33% from the Control group. Although it was the Control group that strongly agreed the most, with 8% and 17%, as opposed to just 8% for Statement 6 only from the Study group. The strong positive response from the Control group to Statement 5 (relating to previous problems) was a bit of a surprise, as they were not the group that received feedback that explicitly related to previous problems and would seem to support the linking back to previous problems that must naturally occur as part of the metacognitive process. The 67% agreement from the Study group was an interesting response; on the one hand it was good that the biggest agreement came from the Study group who explicitly received feedback relating to previous problems, but on the other hand it was confusing given the 69% of the Study group participants received feedback relating to previous problems, but the participants only followed through to selecting to view the previous problem 3% of the time. This implies that the participants' perception was different to receiving explicit feedback or that they were not paying attention to the question in the questionnaire.

Statement 6 (accurate assessment) could imply the self-report of self-efficacy that the Study group were prompted to do, as well as the continual self-assessment that is part of the metacognitive process. The Control group were equally spread between disagree/neutral and agree/strongly agree with 50% either way. Although the Study group had 33% disagree/neutral, the remaining 67% responded with agree/strongly agree. There was no correlation between the Study group participants' responses to Statement 6 and both their average self-efficacy levels and their self-report accuracy.

The area of motivation covered one statement: how SQL-Tutor helped the participant to keep going when things were tough (Statement 7, p.125). This was the first statement where the majority responses for both groups was neutral, with 50% of the Study group and 33% of the Control group. There was still a high percentage who agreed/strongly agreed, with 41% for the Study group and 42% for the Control group. This is also the first statement with strongly disagree responses; 17% of the Control group which was four participants. Out of the four Control group participants that responded with strongly disagree, three of them were below average in terms of the number of problems solved and the attempts taken. However, the fourth participant was above average in the number of problems solved and approximately double the average attempts at problems solved. This would indicate that there were definitely forms of motivation other than the helpfulness of SQL-Tutor for this participant to show a high level of persistence.

The final area was the general usage of SQL-Tutor, asking whether they would have preferred SQL-Tutor to be integrated into their computer lab sessions (Statement 8, p.126) and, aimed at the Study group, whether they agreed that the frequency of the self-reports slider control was unobtrusive (Statement 9, p.126). Both groups agreed/strongly agreed with 67% that the use of SQL-Tutor should be integrated into the timetabled computer lab sessions. This was the second statement that had strongly disagree responses of 8% of the Control group (one participant). The participant who strongly disagreed with it being used in the lab sessions was the one user who replied to Question 1 in Section B (what did they like most about SQL-Tutor?) with "there is nothing I liked about the SQL-Tutor". This same participant also said that they found using SQL-Tutor hard in response to Question 2 about what they liked least about SQL-Tutor. This last response could indicate why they responded negatively to the first eight statements in Section A of the questionnaire, with their perception of not finding SQL-Tutor helpful because they found it difficult.

Although Statement 9 was only aimed at the Study group participants and started with "IF you were prompted to use a slider control…", 25% of the Control group responded with 'agree' that the frequency of the slider control was unobtrusive. Even of the participants of the Control group that correctly responded to this statement with 'neutral', gave frequency suggestions to Question 6, Section B of the questionnaire (asks for alternative frequency of the self-reports slider control). The majority of the Study group were neutral with 58%, which indicates that while they did not completely agree, they still did not find the slider control obtrusive. However, this statement had the largest disagree percentage from the Study group (17%) compared to the rest of the statements. The Study group participants in response to Question 6 and suggesting an alternative frequency for self-reports, had three general suggestions: "start and end of session or after a set number of questions", "end of session only" and "prompted not every time I used the tutor e.g. every three times". The participant who responded with the second suggestion also responded to Statement 9 as strongly agree that it was unobtrusive (the other two participants responded with disagree). The low level of disagrees was a pleasant surprise, as the author expected the use of self-reports to gauge self-efficacy levels would be a lot more decisive. The fact that self-reports did not just ask for the participants' input, but also gave motivational and/or metacognitive feedback, may have been an influence on the acceptance in the self-reports frequency, although there is no way of knowing for this study. If the analysis of the post-study questionnaire had been started immediately after the completed questionnaires were received, then there may have been the possibility of asking follow-up questions.

Some of the responses to the open questions in Section B of the questionnaire have already been discussed, but here is a summary of the remaining questions and responses. For what was liked most about SQL-Tutor in Question 1 (p.127), other than the feedback, there was the user interface (25%) and that it was available all of the time on the Web without the need for any special software (4%). However, the user interface was also the second least liked thing about

SQL-Tutor with 33% (Question 2, p.128). The least liked things about SQL-Tutor were the learner model and the content (e.g. a comparison was made between the style of queries in SQL-Tutor and those in the module). For these two least liked things, there appears to be a link to the participants' understanding of certain problem requirements and their domain knowledge (refer to Table 5.6 for such an example, along with how SQL-Tutor supports variations on the model answer). The responses to Question 3 (p.131) relating to feedback have already been discussed. Question 4 (p.131) asked if there was any other aspect of SQL-Tutor the participants wanted to comment on. There were four categories of responses, each from 8% of the responses for this question; feedback, user interface, usage and technical issues. The responses for feedback were that it needed to be exact, which tends to link to the level of domain knowledge. The user interface responses related to it needing to be updated. The usage comments said it should be used in lab sessions, which is also reflected by the response to Statement 8 in Section A of the questionnaire, and also that SQL-Tutor should use the same database as used in the module lab sessions. This particular comment is a little concerning as it implies that the participant has become too used to the one database scenario that is used throughout the module and potentially finds it difficult to transfer their domain knowledge to different database schemas. The final comments were about the technical issues where SQL-Tutor was not available for approximately a week, the reason is unknown. Question 5 (p.133) asked if the participants had used any other source to help them practice SELECT queries outside the timetabled lab sessions. The majority of 84% had only had the lab sessions and used SQL-Tutor, with 8% using the w3schools Web site which is the site that SQL-Tutor directed the participants to and the remaining 1% used other, unspecified Web resources. It shows a strong reliance on the module lectures and lab sessions, although the author does not know how much of the lab exercises were also finished as self-study outside of the timetabled lab sessions. Question 6 (p.133) regarding suggestions of alternative frequencies of the self-report slider control has already been discussed.

5.4.1 Limitations of the Results

Although analysis of the data has shown the start of some trends in favour of the Study group, the results are not clear cut in providing definitive answers to the research questions. The results are also subject to several limitations, discussed in this section, which needs to be taken into consideration when reviewing the results.

The study did not use dedicated pre- and post-tests in order to help measure any potential learning gains for the participants. Instead the summative assessments from the live delivery of the University module were used (ref to Section 4.3.1 (p.100) for details of the summative assessments). This had the disadvantage of not using the results of tests that had been created specifically tailored for the study and conducted under experimental conditions (refer to this issue in the discussion of the results in Section 5.4, p.157). However, using the summative assessments did allow for the study to be undertaken alongside the delivery of a University module. It cannot be known whether using dedicated tests for the study would have yielded results showing any differences, significant or otherwise, to the summative assessment.

Conducting the study *in the wild* alongside the live delivery of a University module provided a positive contribution (refer to Section 6.2.1, p.176) as it meant data was collected from the use of SQL-Tutor in-situ of the learning of SQL SELECT statements and it was not used under the confines of experimental settings. The use of the study in the wild could be further strengthened by fully incorporating the use of SQL-Tutor into the module delivery itself by using is as standard in the module's workshop/computer laboratory sessions (refer to the first point in the possible further projects in Section 6.5.2, p.184). However, conducting the study in the wild showed a limitation in the period of time over which the participants used SQL-Tutor. They only used SQL-Tutor during the weeks that corresponded to the relevant coverage of SQL SELECT statements in the module delivery, whereas the study was designed to be used over a longer period of time (refer to the schedule in Appendix A, p.200). This would lend further support for incorporating the use of SQL-Tutor into the module delivery itself as just discussed above.

Although the module was delivered to a large number of students, with module assessments results for two hundred and nine students, the final number of study participants was relatively small at thirty six. Refer to Section 5.2.1 (p.106) and Table 5.2 (p.108) for a breakdown of the student numbers across degree courses and study participants/non-participants. The small numbers of study participants has a direct impact on the strength of the results, as they do not provide representation of a wider number of students. However, the lack of participants in proportion to the number of students on the module (17%) also provides further support to incorporating the use of SQL-Tutor into the module delivery as already discussed.

One of the aims of the thesis work was to provide metacognitive feedback relating the learner back to previous related problems they had solved and to see whether this has any measurable gains for the learner on learning and focus (refer to the research questions in Section 1.2, p.3). However, as the analysis of the log files have shown, the Study group participants made very little use of the prompt to view the related past problems (refer to the discussion in Section 5.3.4.1, p.153). This finding greatly limits the impact of the study results in terms of being able to determine whether using such a metacognitive strategy promotes learning gains for the learner. Despite this, this outcome also provides focus for further research work (refer to the last point on p.187 of the further projects Section 6.5.2, p.184).

Another limitation to be taken into consideration of the results is that not all of the data was recorded due to a) data not recorded due to navigation issues of a Web interface (e.g. missing self-reports of self-efficacy as discussed in Section 5.3.4.2, p.154) and b) further refinement of the data logged is required (e.g. recording the difficulty ratings of problems at the point they were selected by the participants as discussed on p.140 of problems visited Section 5.3.1, p.136).

All-in-all the results of this study cannot be taken as the definitive answer to the research questions defined in Section 1.2, p.3. Instead they should be considered more as a starting point on which to develop and refine further research. Refer to Chapter 6 (p.172) for further discussion of the contributions, limitations, lessons learnt and potential future work that the results of this study have made.

5.5 Chapter Summary

This chapter has analysed the data collected during the study by focussing on the pre- and post-results with the summative assessments and the post-study questionnaire. Followed then by the process results which were extracted from the data recorded in the various system log files. Then these results were discussed in the context of the three supporting research questions.

Looking at the data overall, there are good indications that using the SQL-Tutor helped those that chose to take part in the study, though whether this was because those students were more committed from the start is a moot point. There are trends in the data that favour the Study group over the Control group. The results did present quite a few differences, as well as similarities between the Control and Study groups. However, apart from the ratio of the number of attempts per unsolved problems, the differences were shown as not being significantly different.

Unfortunately one crucial aspect of the feedback to the Study group, the references to past problems, were not used to the full, so the effective difference in the learning experience of the two groups was less than intended.

6 Conclusions & Further Work

This chapter summarises the study results in relation to the research questions as presented in Chapter 1 (p.1). Next the key contributions of this study are discussed, along with the limitations and lessons learnt from the study. Finally, points of interest that have resulted from this study are outlined as the basis for potential for further work.

6.1 Research Questions

The supporting research questions outlined in Chapter 1 (p.1) were discussed in detail in Section 5.4 (p.157) based on the completed research. Those discussions are reflected upon and summary answers are presented here in line with the research questions as a whole. The "feedback" mentioned in the support questions relates to the motivational and metacognitive feedback based on the learner's past states and experiences. Table 6.1 below summarises the main outcome from the thesis study in relation to the research questions. The first three rows of the table contain the supporting questions, with the last row relating to the primary question. The last row has been formatted using **bold** text to highlight that it is the primary research question, which the supporting questions (other tables rows) build to. The use of the '+' and '-' symbols has been used to provide a quick visual reference as to whether the study results provided outcomes that positively helped to address the questions.

Table 6.1 Summary thesis study outcomes in relation to the research questions

Research Questions	Study Outcomes
Does providing such feedback lead to any measurable learning gains?	+ Yes, the Study group moved slightly up class rankings; both groups solved similar number of problems, but the Control group took more attempts; both groups better than no Tutor.
Does providing such feedback lead to any measurable gains in learner focus?	- Control group were a bit more persistent on unsolved problems. + Study group made a bit better use of the system.
Do the learners perceive any benefit from using an ITS to aid their learning?	+ Yes for both groups, with the Study group slightly more positive.
What are the effects of including motivational and metacognitive feedback based on the learner's past states and experiences?	**+ Some positive trends.** **- Study group students typically did not follow up references to past similar problems.**

Does providing such feedback lead to any measurable learning gains?

Milestone 1 and the Exam of the University's own summative assessments for the module were used as the pre- and post-tests respectively. As the marks from Milestone 1 were used to assign the participants to a group, the groups for Milestone 1 were statistically equal. However, in the Exam the mean marks for the Study group were 6.25% higher than the Control group. This difference was not significant.

As the assessments and their marking were not done under experimental controls, the Milestone 1 and Exam marks were ranked. The movement in ranking reflected the general movement in the means of the marks. However, the mean change of rank of a Study group participant was between 3 and 4 places up the ranking, whereas the mean rank for the Control group stayed about the same.

In terms of the participants' performance, the Control group solved only 0.76 more problems than the Study group, but they took 3.32 more attempts to do so. The average time spent to solve a problem was close between the groups, with the Study group taking just forty-three seconds less to solve a problem than the Control group. The Study group also had the most participants who took on average less than five-minutes to a problem.

To summarise the answer to this question, there are trends in the data to support the hypothesis that feedback led to learning gains.

Does providing such feedback lead to any measurable gains in learner focus?

The learning focus was assessed by examining effort, persistence and learner behaviour. The levels of effort were similar for both groups, with the Control group containing 2% more participants with good levels of effort compared to the Study group. For unsolved problems, the persistence ratio was, surprisingly, nearly double for the Control compared the Study group, but both groups were equal for the persistence when working on solved problems. The initial difficulty rating was also another indicator of persistence, which showed the same level of difficulty in problems chosen (both solved and unsolved) by both groups. The period of time over which the participants used SQL-Tutor was also an indicator of effort and/or persistence, and this showed that both groups used SQL-Tutor over the same four-week period. There was one exception where one Control group participant logged in one more time four-weeks after the rest of the Control and Study groups.

Learner focus looked at the levels of help used and the use of the functionality that showed the participants the expected output of a problem (this was newly added to both the Control and Study group versions of SQL-Tutor). High-level help (HLH) were help levels that involved showing part or all of the solution, or listing all errors. HLH had to be explicitly selected by the participants to be shown. When looking at all of the problems (solved and unsolved), the Study group used HLH 1% less than the Control group, with both groups using HLH more on unsolved problems, than solved problems. Looking at the expected query output was used 7% more by the Study group than the Control group.

To summarise the answer to this question again based on the results of this research, there were trends in the data that implied gains in learner focus for both groups, although none were significant and they fluctuated in favouring each group.

Do the learners perceive any benefit from using an ITS to aid their learning?

The post-study questionnaire was designed to ascertain the perception of the participants as to the benefit of using ITS as part of learning. Section A of the questionnaire contained closed questions in the form of nine statements using a five-point Likert scale, with the statements covering five areas: domain-related, learning process, metacognition, motivation and general usage. While the participants from both groups tended to respond positively (agree/strongly agree), it was the Study group who responded more positively, with most of the disagree and all of the strongly disagree responses coming from Control group participants.

Section B contained six open questions asking the participants what they liked and disliked the most about SQL-Tutor, what they thought of the feedback, alternative self-report frequencies and what other resources did they use for learning SQL SELECT statements. Both groups mentioned the same items in their feedback. The user interface was both liked and disliked, along with the learner model and the feedback. Unfortunately, the question(s) were not specific to the type of feedback (mentioned in Section 6.3 (p.178) as a limitation). Hardly any participants used any other resource than SQL-Tutor and the module material to aid their learning.

To summarise this question based on the questionnaire responses, participants from both groups did think that SQL-Tutor helped in their learning, with slightly stronger positive responses received from the Study group. The opinions of likes and dislikes were generally balanced between the two groups.

What are the effects on the learner of including motivational and metacognitive feedback based on the learner's past states and experiences?

This is the main research question that is answered by the three supporting questions just discussed. Overall this study has yielded some mixed results with some trends favouring the Control group, but most favouring the Study group. However, the trends in gain witnessed in answer to the first question (learning gains), while most are in favour of the Study group, they cannot be categorically attributed to the inclusion of the motivational and metacognitive feedback over pure chance. For the second question (learner focus), again there are some trends in gain, but in total the gains do not favour one group over another. Finally the third question (participant perception), the responses received from the post-study questionnaire did slightly favour the Study group over the Control group, with the Study group participants slightly more perceiving the benefit out of using SQL-Tutor. However, the differences to Section A of the questionnaire were not significantly different.

In summary, comparisons between groups showed some differing trends both in learning and behaviour in favour of the Study group, though these trends were not significantly different. The study findings showed promise for the use of motivational and metacognitive feedback based on the learners' past states and experiences that could be used as a basis for further research work and refinement.

6.2 Contributions

The key contributions of this study that have a wider applicability beyond the scope of this project are discussed here.

6.2.1 Evaluation Context

This study has evaluated the system in the context of an undergraduate curriculum of a university, where the participants used the system on a voluntary basis. The study was not confined to experimental lab sessions; instead, the participants used it alongside the live delivery of a university module (it was used *in the wild*). This study adds to the growing literature of studies conducted in the wild (e.g. Arroyo et al., 2009; Mathews, 2012; Vanlehn et al., 2005) and shows up some of the difficulties of this type of study (refer to the limitations Section 6.3, p.178, and lessons learnt Section 6.4, p.180).

Although the study involved the use of SQL-Tutor alongside the delivery of a live University module, the participants of the study used SQL-Tutor on a voluntary basis as a source for learning in additional to the module workshops and lectures. In the study by Vanlehn et al. (2005) the Andes physics tutor was used to complete homework tasks and although the participants could use Andes in their own time, just as the participants in this study chose when to use SQL-Tutor, they did not have any choice in using Andes itself. The participants of the Andes study had to use Andes to complete their homework tasks as set by their teachers/instructors. This differs from this study where the use of SQL-Tutor was completely voluntary and not a requirement of the module. Similarly the study by Arroyo et al. (2009) used physiological sensors while participants were using the Wayang Outpost geometry tutor during standard class sessions, which meant it was still used in a controlled environment. While the log file data from SQL-Tutor has been previously collected from the live use of SQL-Tutor (Mathews, 2012; Mathews and Mitrovic, 2008a), it did not involve the use of two different versions of SQL-Tutor nor were the participants students studying on a live delivery of a database module. The results discussed by Mathews and Mitrovic (2008a) used an online version of SQL-Tutor that was made freely available to people around the world that bought particular database textbooks. Although using the log files from the online version allowed for a very large result set of one thousand and eighty-three users, any conclusions drawn from the data is done so without the context of the users' use of the tutor being known. For example, in the Andes study the tutor was used in the context of completing homework tasks, in the Wayang Outpost study the tutor was used as part of the class sessions for the subject being taught (geometry) and likewise, SQL-Tutor in this study was used in the context of learning SQL SELECT statements as part of a University module.

6.2.2 Combined Motivation and Metacognition

As far as the author is aware, based on the review of literature and feedback received from paper submission for publication, this is the first study that attempts to deal with both motivation and metacognition at the same time. As seen from the ITS review in Chapter 2 (p.6), other ITS's focus on one or the other, and maybe the second may be a by-product of some other action, but they do not explicitly focus on both motivation and metacognition.

6.2.3 Relating to Past Problems

Relating the current problem to a previous problem as done in the version of SQL-Tutor used by the Study group in this study has gone beyond the use of past problems in ELM-ART (Weber and Brusilovsky, 2001). The feedback in this study explicitly mentioned that there were related problems the participant had previously solved, along with giving the database name and problem number. Then when the participant viewed the related problem, their code was displayed exactly as they had entered it. In comparison, ELM-ART did not make any explicit reference to the problem being one that the participant had previously solved, nor that it was their code, and the code was displayed differently to entered (e.g. all in uppercase when mixed case had been used by the participant). So this study incorporated the use of past problems as did ELM-ART, but this study made references to past problems explicit for the learner; the feedback explicitly mentioned that the learner had previously solved a similar problem, as well as signposting it further by labelling the on-screen button used to display the past problem as "View Past Problem". Refer to Figure 3.13 (p.82) for example of the user interface and the feedback given.

6.2.4 Participants' Best Interests

Considering that "we seem to be very bad at recognizing that a problem we have just done can actually help us solve the problem we are currently attempting" (Robertson, 2001), the Study group participants were presented with the opportunity to view related past problems to try and help them, but they chose not to take the opportunity. The participants did not behave as the author thought was in their best interest. Generally, it was taken for granted that the Study group participants would use the link to view the related previous problem when it was explicitly included in the feedback. Refer to Section 6.5 (p.183) for further discussion of this. This study provides evidence to support the above statement by Robertson (2001) and it also provides evidence that even prompting a learner to study past problems is not sufficient on its own. The learners' might recognise the prompt, but they do not necessarily do anything about it.

6.2.5 Modified SQL-Tutor

This study has designed and implemented modifications to a version of SQL-Tutor, making changes to its interface, logging mechanism and to the feedback provided. While some modifications were purely to implement the functionality required from the Study group version of the tutor, other changes were applied to both the Control and Study group versions (e.g. viewing the output expected for a problem).

6.3 Limitations

The following outlines the limitations with this study that should be taken into account when reviewing the results.

- The final number of participants that actively took part in the study was quite small, with twenty participants in the Control group and sixteen in the Study group. Although the number of participants was comparable, as an example, to one of the studies mentioned in Section 6.2.1 (p.); the study using Wayang Outpost tutor in standard/timetabled class sessions had thirty –eight participants (Arroyo et al., 2009).

- The fact that most of the Study group participants did not view past problems when they had the opportunity meant that the difference in the experiences of the Study and Control groups was less than expected. This issue may have been identified in a pilot study and then potential methods for addressing it could have been considered in preparation for the main study. Instead, such considerations are discussed as potential further work in Section 6.5.2 (p.184). Also refer to Section 6.4 (p.180) for discussion on how a future pilot study could be used.

- Despite the study design and intention, the actual length of elapsed time that the participants used SQL-Tutor was far less than planned. While the study design was an open, long-lasting study with no fixed contacts, where SQL-Tutor was to be used across a period of three and a half months, it was actually only used over a four week period. Refer to Section 6.4 (p.180) for further discussion.

- The use of SQL-Tutor was embedded into the live delivery of an undergraduate database module, with the marks from summative assessments used in place of dedicated pre- and post-tests. Unfortunately, the data from the summative assessment for the post-test was sparse as only four questions were relevant to SELECT queries.

- Not all of the data was recorded that should have been due to the fact the participants could by-pass Web pages by using the navigation controls in the Web browser itself. For the Study group this meant that not all of participants' self-reports of their self-efficacy were recorded and this did affect the feedback statements that some of the Study group participants received.

- Some of the questions in the post-study questionnaire were too general and either needed to be more specific or provide some means for follow-up questions.

6.4 Lessons Learnt

The following outlines the lessons that have been learnt from the experience of this research study process.

- Despite being able to use the summative assessments from the module as pre- and post-tests, it would still be of more benefit for the post-study comparison to use dedicated pre- and post-tests. For example, issue test A to half the participants and test B to the other half at the start of the study, then swap the tests around for the post-tests.

- Designing a study as an open, long-lasting study with no fixed contact points was a good idea and was a good fit in terms of letting the participants use SQL-Tutor over an extended period in order to try to maximise the benefit of using metacognitive feedback relating to the past problems solved by the participants. In reality, this did not work as well as planned, with the participants using SQL-Tutor for approximately a third of the planned period. This appears to be clearly linked to two factors: the module content and structure of the academic year. The participants used SQL-Tutor for a four-week period from the time it was issued through to the last teaching week of semester one. Apart from one exception, SQL-Tutor was not used again. The participants appear to only have used SQL-Tutor while SELECT queries were being directly taught in the module class sessions. The timing of the winter break could also have influenced a break in momentum for using SQL-Tutor. Planning such studies need to be scheduled to start as soon into the teaching semester as possible (e.g. either the first or second week that SELECT statements started to be taught in the module sessions) in order to maximise the time that the use of SQL-Tutor mirrors the module content delivery. Planned pauses either need to be incorporated into the study timeline or the study should not span non-teaching weeks such as winter breaks and assessment weeks.

- As already mentioned in the previous section (p.178) and at the beginning of Chapter 4 (p.93) a pilot study was not carried out prior to the main thesis study. As a lesson learnt, in hindsight a pilot study would have been conducted that purely involved the use of the modified version of SQL-Tutor (the version used by the Study group in the main thesis study). The pilot study would have involved a group of volunteers to use SQL-Tutor over a period of time such as two weeks. The volunteers would either have been second year undergraduate students who had taken the database module in their previous year of study or, if the thesis study schedule allowed, first year students. Having volunteers from the first year would mean that the pilot study occurred during Semester 1 in one academic year and the main study would then have to take place in the following academic year in order to fit with the module delivery schedule. The advantage of having volunteers from the first year students is that they would be using SQL-Tutor at the same time as they were actually learning SQL SELECT statements. So they would be the same level of volunteers as designed for the main study and would also be experiencing using the tutoring system as a learning tool, as opposed to second year students where using the tool would be more akin to revision. In essence a pilot study would have developed upon the limited user testing session that was conducted (refer to Section 3.3.1.2, p.77) in order to provide a study that would not only test the modified SQL-Tutor, but also the forms of data collection that were used in this thesis study.

  A pilot study would have elicited feedback from the participants using the log files produced by SQL-Tutor and the post-study questionnaire. Informal interviews would also be used to follow-up on the post-study questionnaires in order to gain further feedback and to clarify any responses as required. This means that the pilot study as a whole would have the advantage of potentially identifying any issues that were discovered in the study discussed in this thesis and allowing such issues to be addressed ready for the main study. For example, as discussed in Section 6.3, p.178, the issue that the Study group did not make use of the feature to view their answer from a related problem they had previously solved or that some of the questions in the post-study questionnaire needed to be finer grained in their focus. A pilot study may have also helped to mitigate such issues as loss of data recording due to the participants being able to circumvent the navigation controls in SQL-Tutor by using the standard Web browser buttons (discussed in the next bullet point below).

- Tighter controls need to be included to either prevent or correctly manage the instances when participants try to skip SQL-Tutor Web pages by using the navigation controls in the Web browser itself or close the Web browser window without logging out of SQL-Tutor correctly. In relation to the Study group, data directly relevant to the study was lost due to participants not just using the navigation buttons built into SQL-Tutor itself. The base version of SQL-Tutor only had a 'logout' button on one page, so in order to promote participants logging out correctly; 'logout' buttons were displayed on main Web pages that make up the SQL-Tutor interface. On the Web pages where the self-report slider control was shown, the 'continue' button was purposefully disabled until the participant had selected a self-report value from either the drop-down list box or the slider control. Unfortunately, this did not prevent the Web browser buttons from being used.

- In simple terms, record everything! It is better to have too much data to extract elements from as opposed to not having enough data in the first place. While additional log files were produced for all participants and just for the Study group, when analysing the log files it became apparent where logging some data that was only recorded by the Study group would have been beneficial if recorded for all participants. For example, when a participant from the Study group selected a problem to work on, the current difficulty rating were recorded in a log file. It would have been beneficial for this to also be recorded for the Control group as another element of comparison (refer to Section 5.3.1 (p.136) for the analysis including difficulty ratings). The log file that recorded the content facts used by the Feedback Rules Engine (FRE), refer to Section 3.3.2 (p.78) for details, would have also benefited from recording the actual feedback statements given to the Study group participants. This would mean that the data is captured at source rather than having to develop a separate utility to extract the content facts from the log files and then retrace the processing the FRE did to determine the feedback provided to the participants.

- Do not assume that the participants will use a new help feature in the system. As discussed in Section 5.3.4.1 (p.153), while Study group participants received metacognitive feedback explicitly linking to previous problems that they had solved as a means of help, very few participants chose to click on the link to view the past problem. There are various reasons why this was the case, but which one(s), if any, apply is unknown. For example, it could have been the timing or manner of the feedback that prevented the participants to looking at the previous problem. They simply may not have seen the point of doing so or they may not have actually read the feedback. As a counter point to that, it may have been that the feedback itself, which mentioned the problem number and database name of the past problem, may have been enough of a prompt at a metacognitive level. This is discussed further in the next section.

6.5 Further Work

The results from this project have raised some interesting questions that could be incorporated into the potential further projects that are discussed in this section. For example, why were the differences between the Control an Study groups not more pronounced? Did any gain in self-efficacy from using SQL-Tutor make the participants more likely to succeed in other tasks, both related and unrelated to SELECT statements? Did the participants feel that the praise coming from SQL-Tutor was not valuable/believable as that from a human tutor?

This section outlines areas of potential further work by dividing them into two categories: supporting changes and further projects.

6.5.1 Supporting Changes

The following lists potential further work that may not necessarily constitute a complete research project, instead support changes within a project.

- Refine the post-study questionnaire by providing study group specific versions, so that finer grained questions could be asked that are only relevant to a given group. For example, for the Study group ask questions that are targeted at the different types of feedback received from the tutoring system.

- In SQL-Tutor, support the functionality of the 'SQL Help' button to not only go to a SQL reference/tutorial Web site, but to directly go to a page that is relevant to the main source of errors in the last attempt at a problem. For example, if the main errors relate to the WHERE clause, then open the relevant Web page on the external supporting Web site.

- Considering the large response to the open questions in Section B of the post-study questionnaire were concerned with the interface of SQL-Tutor, update the HTML interface of the tutor while keeping the same functionality of flow of Web pages. This could also be expanded into a project in its own right to assess the influence, if any, of the user interface with the current interface assessed via a modified interface.

- Directly related to a point in Section 6.4 (p.180), extend the data that is recorded in the logs files in order to maximise the data that can be extracted and analysed at the end of a study.

6.5.2 Further Projects

The following lists the areas of potential further research projects based on the results and experience gained from this study.

- Conduct another study similar to this one, but with some changes in terms of the level it is embedded into the University's module, along with the use of study-specific pre- and post-tests. For example, start the study as soon into the semester as possible following the module starting to teach SQL SELECT queries and only run the study for as long as SELECT queries are covered in the module. Also fully incorporate the use of SQL-Tutor into the module delivery by using it as standard in the module's workshop/computer laboratory sessions. This would not only address the limitations of the number of participants as discussed in Sections 5.4.1 (p.168) and 6.3 (p.178), but also the majority view of participants as gained from Statement 8 (integrate into lab sessions), Section A of the post-study questionnaire (Section 5.2.2.1, p.126).

- Some feedback received from participants in answer to the open Question 2 in Section B of the post-study questionnaire related to the perception that one had to enter an answer that was exactly the same as the model answer (refer to Section 5.2.2.2, p.127). A further research project could examine the level of granularity of constraints and the associated feedback given. The points discussed and the example given Section 5.2.2.2 showed that the learner's perception of the problems requirements, along with their level of domain knowledge and capability might have influenced their opinion about how "rigid" the queries were. In addition, the feedback for a constraint may be either too narrow or not sufficiently finely-tuned to point to the error.

- Statement 5 in Section A of the post-study questionnaire asked the participants "SQL-Tutor helped me to solve problems by relating them to previous problems solved" (refer to Section 5.2.2.1, p.120). While this implied the metacognitive feedback relating to previously solved problems for the Study group, the majority (58%) of the Control group also responded positively (agree/strongly agree) to the statement. A project to research the degree to which learners naturally recall and relate to previous problems during their normal learning process compared to directed metacognitive feedback would be interesting.

- Following on from some feedback to Question 6 in Section B of the post-study questionnaire (frequency of self-reports), investigate the effects of allowing the learner to control the frequency, but with warnings of potential consequences.

- Again related to response to Question 2 in Section B of the post-study questionnaire (what was liked least about SQL-Tutor), some responses said that they did not agree with the "progress bar" and the "way it judges which topics are strongest and weakest". Researching the effects of using negotiated open learner models to aid accuracy and metacognition would be an interesting point. Research has been conducted into negotiated open learner models such as the work related to Mr Collins (domain of learning Portuguese) and the CALMsystem (science questions) as described by Bull and Kay (2013), but also a basic negotiated model was implemented by Thomson and Mitrovic (2010) in an ITS called EER-Tutor (entity-relationship diagrams). EER-Tutor is a constraint-based tutor like SQL-Tutor and uses the same learner model representation as SQL-Tutor. Findings generally point to an increase the participants' self-reflection and self-assessment.

- One element that had been considered, but not included in this study was the inclusion of versions of the same feedback statement, which would give the same feedback message intent, but worded differently. Having feedback statement versions would not only reduce the risk of receiving the same feedback twice in succession, but it could also be more closely linked to motivation and metacognition as per the current state of the learner. This would also allow for different levels of scaffolding by means of the feedback statements. This would build on the feedback statements used in this study, which depended upon the rules triggered in the 'Feedback Rules Engine' (refer to Section 3.3.2 (p.78) for further details).

- Another potential for further work is to concentrate on the actual content of the feedback. For example, using positive and negative feedback (Barrow, 2008) and politeness in feedback (Johnson and Rizzo, 2004). The feedback content could also be combined with the potential further work mentioned in the previous point, different versions of feedback statements.

- While this study provided metacognitive feedback that explicitly linked to previous solved problems, unfortunately the number of instances where the participants clicked on the link to view the previous problem was unexpectedly low. As discussed in Sections 5.3.4.1 (p.153) and 6.4 (p.180), one of the potential reasons why the number viewing the problems were low is that they simply did not read the feedback in the first place. Further research could be undertaken to determine whether the participants do read the feedback or not. This could potentially be achieved by using eye-tracking technology and build on the work done by Najar et al. (2014) where eye-tracking was used to study where the participants looked when viewing on a problem in SQL-Tutor (e.g. the problem explanation and the database schema. Another possibility might be to test different page layouts for presenting the motivational and metacognitive feedback.

6.6 Closing Remarks

This thesis has demonstrated that there is potential for including motivational and metacognitive feedback based on the learner's past state and experiences when using an ITS. Although there were no definitive answers, the study did yield some interesting results. This thesis has provided a basis from which the aims presented can be further developed and explored in order to refine the gains achieved.

References

Al-Shuaily, H. & Renaud, K. (2010). SQL Patterns - A New Approach for Teaching SQL. 8th HEA Workshop on Teaching, Learning and Assessment of Databases, Abertay-Dundee. 29-40.

Alexander, S., Sarrafzadeh, A. & Hill, S. (2006). Easy with Eve: A Functional Affective Tutoring System. *In:* REBOLLEDO-MENDEZ, G. & E., M.-M., eds. Workshop on Motivational and Affective Issues in ITS, 8th International Conference on ITS 2006, Jhongli, Taiwan. 38-45.

Allen, E., Cartwright, R. & Stoler, B. (2002). DrJava: a lightweight pedagogic environment for Java. Proceedings of the 33rd SIGCSE technical symposium on Computer science education, Cincinnati, Kentucky. ACM, 137-141.

Ames, C. (1992). Classrooms: Goals, Structures, and Student Motivation. *Journal of Educational Psychology,* 84(3)**,** 261-271.

Anderson, J., R., Corbett, A. T., Koedinger, K. R. & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences,* 4(2)**,** 167-207.

Anderson, J. R., Matessa, M. & Lebiere, C. (1997). ACT-R: a theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction,* 12**,** 439-462.

Arroyo, I., Cooper, D. G., Burleson, W., Woolf, B. P., Muldner, K. & Christopherson, R. (2009). Emotion Sensors Go To School. AIED. 17-24.

Azevedo, R., Johnson, A., Chauncey, A. & Burkett, C. (2010). Self-regulated learning with MetaTutor: Advancing the science of learning with MetaCognitive tools. *New science of learning.* Springer, 225-247.

Azevedo, R. & Witherspoon, A. M. (2008). Detecting, Tracking, and Modeling Self-Regulatory Processes during Complex Learning with Hypermedia. In AAAI Fall Symposium: Biologically Inspired Cognitive Architectures. 16-26.

Baker, R. S., D'Mello, S. K., Rodrigo, M. M. T. & Graesser, A. C. (2010). Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive–affective states during interactions with three different computer-based learning environments. *International Journal of Human-Computer Studies,* 68**,** 223-241.

Bandura, A. (1977). Self-efficacy: Toward a Unifying Theory of Behavioural Change. *Psychological Review,* 84(2)**,** 191-215.

Bandura, A. (1986). *Social foundations of thought and action: A social cognitive theory,* Englewood Cliffs, NJ, Prentice- Hall, Inc.

Bandura, A. (1989). Social Cognitive Theory. *In:* VASTA, R. (ed.) *Annals of child development. Vol. 6. Six theories of child development.* Greenwich, CT: JAI Press, 1-60.

Bandura, A. (1997). *Self-Efficacy: The Exercise of Control,* New York, Freeman.

Bandura, A. (2006). Guide for Constructing Self-Efficacy Scales. *In:* PAJARES, F. & URDAN, T. (eds.) *Self-Efficacy Beliefs of Adolescents.* IAP, 307-337.

Barnette, J. J. (2010). Likert Scaling. *In:* SALKIND, N. J. (ed.) *Encyclopedia of Research Design*, (Vol. I, pp 714-716), Sage Publications, 978-1-4129-6127-1

Barrow, D. (2008). *Assessing the Impact of Positive Feedback in Constraint-Based Tutors.* MSc, University of Canterbury, Christchurch, New Zealand.

Barrow, D., Mitrovic, A., Ohlsson, S. & Grimley, M. (2008). Assessing the Impact of Positive Feedback in Constraint-Based Tutors. *In:* WOOLF, B. P., AÏMEUR, E., NKAMBOU, R. &

LAJOIE, S., eds. 9th International Conference of Intelligent Tutoring Systems (ITS2008), Montreal, Canada. 250-259.

Baylor, A. L. & Kim, Y. (2005). Simulating instructional roles through pedagogical agents. *International Journal of Artificial Intelligence in Education,* 15**,** 95-115.

Beal, C. R. & Lee, H. (2005). Creating a pedagogical model that uses student self reports of motivation and mood to adapt ITS instruction. Workshop on Motivation and Affect in Educational Software. 12th International Conference on Artificial Intelligence in Education., Amsterdam, Netherlands. 39-46.

Ben-Ari, M. (2001). Constructivism in Computer Science Education. *Journal of Computers in Mathematics and Science Teaching* 20(1)**,** 45-73.

Benchetrit, O. & Frasson, C. (2004). Controlling Emotional Conditions for Learning. Intelligent Tutoring Systems: 7th International Conference, ITS2004, Maceio-Alagoas, Brazil.

Bevan, N. (2005). Guidelines and Standards for Web Usability. Proceedings of HCI International 2005. Lawrence Erlbaum, 10-19.

Bevan, N. & Kincla, S. (2003). Usability guidelines for academic websites. Joint Information Systems Committee.

Bhagat, S., Bhagat, L., Kavalan, J. & Sasikumar, M. (2002). Acharya: An intelligent tutoring environment for learning SQL. Proceedings of the Vidyakash 2002 International Conference on Online Learning. 15-17.

Bladek, C. & Deek, F. P. (2005). Understanding novice programmers difficulties as a requirement to specifying effective learning environments. *In:* NATA, R. (ed.) *New Directions in Higher Education.* Nova Science Publishing, 1-22.

Blakey, E. & Spence, S. (1990). Developing Metacognition. *ERIC Digest. ED327218* [Online]. Available: http://www.vtaide.com/png/ERIC/Metacognition.htm [Accessed 11/01/2009].

Boekaerts, M. (2007). Understanding students' affective processes in the classroom. *Emotion in education***,** 37-56.

Boreham, N. C. (1987). Learning from Experience in Diagnostic Problem Solving. *In:* RICHARDSON, J. T. E., EYSENCK, M. W. & PIPER, D. W. (eds.) *Student Learning: Research in Education and Cognitive Psychology.* Milton Keynes, UK: The Society for Research into Higher Education and Open University Press, 89-97.

Bosch, N., D'Mello, S. & Mills, C. (2013). What emotions do novices experience during their first computer programming learning session? Artificial Intelligence in Education. Springer, 11-20.

Broekens, J. & Brinkman, W.-P. (2013). AffectButton: A method for reliable and valid affective self-report. *International Journal of Human-Computer Studies,* 71**,** 641-667.

Bull, S. & Kay, J. (2013). Open Learner Models as Drivers for Metacognitive Processes. *In:* AZEVEDO, R. & ALEVEN, V. (eds.) *International Handbook of Metacognition and Learning Technologies.* Springer New York, 349-365.

Bull, S. & Vatrapu, R. (2012). Negotiated Learner Models for Today. The 20th International Conference on Computers in Education (ICCE 2012). 96-100.

Burleson, W. & Picard, R. (2004). Affective Agents: Sustaining Motivation to Learn Through Failure and a State of "Stuck". Social and Emotional Intelligence in Learning Environments Workshop in conjunction with the 7th International Conference on Intelligent Tutoring Systems, Maceio - Alagoas, Brasil.

Burns, H. L. & Capps, C. G. (2013). Foundations of Intelligent Tutoring Systems: An Introduction. *In:* POLSON, M. C. & RICHARDSON, J. J. (eds.) *Foundations of Intelligent Tutoring Systems.* Taylor & Francis, Psychology Press, 1-19.

Butz, C. J., Hua, S. & Maguire, R. B. (2006). A Web-based Bayesian Intelligent Tutoring System for Computer Programming. *Intelligence and Agent Systems, 4(1),* 4(1)**,** 77-97.

Butz, C. J., Hua, S. & Maguire, R. B. (2008). Web-Based Bayesian Intelligent Tutoring Systems. *In:* NAYAK, R., ICHALKARANJE, N. & JAIN, L. C. (eds.) *Evolution of the Web in Artificial Intelligent Environments.* Springer, 221-242.

Cameron, J. & Pierce, W. D. (1994). Reinforcement, Reward, and Intrinsic Motivation: A Meta-Analysis. *Review of Educational Research,* 64(3)**,** 363-424.

Caspersen, M., E & Kölling, M. (2006). A novice's process of object-oriented programming. Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications, Portland, Oregon, USA. ACM.

Clark, M. & Boyle, R. (2006). Computer Science in English High Schools: We Lost the S, Now the C Is Going. *Informatics Education – The Bridge between Using and Understanding Computers.* Lecture Notes in Computer Science, Vol. 4226, Springer-Verlag, 83-93.

Cleary, T. J. & Callan, G. L. (2014). Student Self-Regulated Learning in an Urban High School: Predictive Validity and Relations Between Teacher Ratings and Student Self-Reports. *Journal of Psychoeducational Assessment,* 32**,** 295-305.

Conati, C. (2011). Combining cognitive appraisal and sensors for affect detection in a framework for modeling user affect. *New Perspectives on Affect and Learning Technologies.* Springer, 71-84.

Conati, C. & Maclaren, H. (2005). Data-driven Refinement of a Probabilistic Model of User Affect. Proc. of 10th International Conference UM'05, Edinburgh, Scotland. Springer-Verlag, 40-49.

Conati, C. & Zhao, X. (2004). Building and Evaluating an Intelligent Pedagogical Agent to Improve the Effectiveness of an Educational Game. IUI'04, Madeira, Funchal, Portugal, January 13-16. 6-13.

Corbett, A., T. & Anderson, J., R. (2001). Locus of feedback control in computer-based tutoring: impact on learning rate, achievement and attitudes. Proceedings of the SIGCHI conference on Human factors in computing systems, Seattle, Washington, United States. ACM.

Cronbach, L. J. & Meehl, P. E. (1955). Construct validity in psychological tests. *Psychological bulletin,* 52**,** 281.

D'Mello, S., Taylor, R., Davidson, K. & Graesser, A. (2008a). Self Versus Teacher Judgments of Learner Emotions During a Tutoring Session with AutoTutor. *In:* WOOLF, B. P., AÏMEUR, E., NKAMBOU, R. & LAJOIE, S., eds. 9th International Conference of Intelligent Tutoring Systems (ITS2008), Montreal, Canada. 9-18.

D'Mello, S. K., Craig, S. D., Witherspoon, A., McDaniel, B. & Graesser, A. (2008b). Automatic Detection of Learner's Affect from Conversational Cues. *User Modeling and User-Adapted Interaction,* 18 (1-2)**,** 45-80.

D'Mello, S. K., Lehman, B. & Graesser, A. (2011). A Motivationally Supportive Affect-Sensitive AutoTutor. *In:* CALVO, R. A. & D'MELLO, S. K. (eds.) *New Perspectives on Affect and Learning Technologies.* Springer New York, 113-126.

Dai, D. Y. & Sternberg, R. J. (2004). Beyond Cognitivism: Toward an Integrated Understanding of Intellectual Functioning and Development. *In:* DAI, D. Y. & STERNBERG, R. J. (eds.) *Motivation, Emotion, and Cognition: Integrative Perspectives on Intellectual Functioning and Development.* Lawrence Erlbaum Associates, 3-38.

Davidson, J. E., Deuser, R. & Sternberg, R. J. (1996). The Role of Metacognition in Problem Solving. *In:* METCALFE, J. & SHIMAMURA, A. P. (eds.) *Metacognition: knowing about knowing.* Cambridge, MA, US: The MIT Press, 207-226.

de Vicente, A. & Pain, H. (1999). Motivation self-report in ITS. *In:* LAJOIE, S. P. & VIVET, M., eds. Proceedings of the Ninth World Conference on Artificial Intelligence in Education, Amsterdam. IOS Press, 648-650.

de Vicente, A. & Pain, H. (2002). Informing the Detection of the Students' Motivational State: an Empirical Study. *In:* CERRI, S. A., GOUARDERES, G. & PARAGUACU, F., eds. Proceedings of the Sixth International Conference on Intelligent Tutoring Systems. 933-943.

de Vicente, A. & Pain, H. (2003). Validating the Detection of a Student's Motivational State. *In:* MENDEZ VILAS, A., MESA GONZALEZ, J. A. & MESA GONZALEZ, J., eds. Proceedings of the Second International Conference on Multimedia Information & Communication Technologies in Education (m-ICTE2003), Merida, Junta de Extramadura. 2004-2008.

Deci, E. L. (1980). *The Psychology of Self-determination*, Lexington Books.

Deci, E. L. & Ryan, R. M. (1985). *Intrinsic motivation and self-determination in human behavior,* New York, Plenum.

Deci, E. L. & Ryan, R. M. (1987). The Support of Autonomy and the Control of Behavior. *Journal of Personality and Social Psychology,* 53(6)**,** 1024-1037.

del Soldato, T. (1993). *Motivation in Tutoring Systems.* D.Phil DPhil Thesis, University of Sussex.

del Soldato, T. & du Boulay, B. (1995). Implementation of motivational tactics in tutoring systems. *Journal of Artificial Intelligence in Education,* 6**,** 337-378.

Dennis, M., Masthoff, J., Pain, H. & Mellish, C. (2011). Does self-efficacy matter when generating feedback? Artificial Intelligence in Education. Springer, 444-446.

Desmarais, M. & Baker, R. J. d. (2012). A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction,* 22**,** 9-38.

Dollinger, R. (2010). SQL Lightweight Tutoring Module–Semantic Analysis of SQL Queries based on XML Representation and LINQ. World Conference on Educational Multimedia, Hypermedia and Telecommunications. 3323-3328.

du Boulay, B. (1989). Some Difficulties of Learning to Program. *In:* SOLOWAY, E. & SPOHRER, J. C. (eds.) *Studying the Novice Programmer.* Hillsdale, NJ: Lawrence Erlbaum, 283-299.

du Boulay, B., Avramides, K., Luckin, R., Martínez-Mirón, E., Rebolledo-Mendez, G. & Carr, A. (2010). Towards Systems That Care: A Conceptual Framework based on Motivation, Metacognition and Affect. *International Journal of Artificial Intelligence in Education (IJAIED),* 20 (3)**,** 197-229.

du Boulay, B. & Luckin, R. (2001). The Plausibility Problem: An Initial Analysis. Cognitive Technology: Instruments of Mind. 4th International Conference CT 2001, Coventry, UK. Springer Berlin Heidelberg, 289-300.

du Boulay, B., Luckin, R. & del Soldato, T. (1999). The plausibility problem: Human teaching tactics in the 'hands' of a machine. Artificial intelligence in education, open learning environments: New computational technologies to support learning, exploration, and collaboration, proceedings of AIED-99. 225-232.

du Boulay, B., O'Shea, T. & Monk, J. (1981). The black box inside the glass box: presenting computing concepts to novices. *international Journal of Man-Machine Studies,* 14**,** 237-249.

du Boulay, J. B. H. (1980). Teaching teachers mathematics through programming. *International Journal of Mathematical Education in Science and Technology,* 11**,** 347-360.

Eccles, J. S., Adler, T. F., Futterman, R., Goff, S. B., Kaczala, C. M., Meece, J. L. & Midgley, C. (1983). Expectancies, values, and academic behaviors. *In:* SPENCE, J. T. (ed.) *Achievement and achievement motivation.* San Francisco, CA: W. H. Freeman, 75-146.

Efklides, A. (2011). Interactions of metacognition with motivation and affect in self-regulated learning: The MASRL model. *Educational psychologist,* 46**,** 6-25.

Ekman, P. (1997). Should We Call it Expression or Communication? *Innovations in Social Science Research,* 10(4), 333-344.

Elliot, A. J. & Harackiewicz, J. M. (1996). Approach and Avoidance Achievement Goals and Intrinsic Motivation: A Mediational Analysis. *Journal of Personality and Social Psychology,* 70(3), 461-475.

Farnham-Diggory, S. (1972). *Cognitive Processes in Education: A Psychological Preparation for Teaching and Curriculum Development,* New York, Harper & Row Publishers Inc.

Field, A. P. (2013). *Discovering statistics using IBM SPSS statistics : and sex and drugs and rock 'n' roll,* London, SAGE.

Flavell, J. H. (1976). Metacognitive Aspects of Problem Solving. *In:* RESNICK, L. B. (ed.) *The Nature of Intelligence.* Lawrence Erlbaum Associates, 231-235.

Forgas, J. (2000). The Role of Affect in Social Cognition. *In:* FORGAS, J. (ed.) *Feeling and Thinking: The Role of Affect in Social Cognition.* New York: Cambridge University Press, 1-28.

Fritz, C. O., Morris, P. E. & Richler, J. J. (2012). Effect Size Estimates: Current Use, Calculations, and Interpretation. *Journal of Experimental Psychology: General,* 141, 2-18.

Fulmer, S. M., D'Mello, S. K., Strain, A. & Graesser, A. C. (2014). Interest-based text preference moderates the effect of text difficulty on. *Educational Psychology,* 35, 17-27.

Gage, N. L. & Berliner, D. C. (1979). *Educational Psychology,* Chicago, Rand McNally College Pub. Co.

Gama, C. (2004). Metacognition in Interactive Learning Environments: The Reflection Assistant Model. *In:* LESTER, J. C., VICARI, R. M. & PARAGUAÇU, F., eds. Proceedings of the International Conference on Intelligent Tutoring Systems, Berlin. Springer-Verlag, 668-677.

Gama, C. A. (2005). *Integrating metacognition instruction in interactive learning environments.* University of Sussex.

Gómez-Albarrán, M. (2005). The Teaching and Learning of Programming: A Survey of Supporting Software Tools. *The Computer Journal, British Computer Society,* 48(2), 130-144.

Gómez-Martín, M., Gómez-Martín, P. & González-Calero, P. (2006). Dynamic Binding Is the Name of the Game. *Entertainment Computing - ICEC 2006.* 229-232.

Gómez-Martín, M. A., Gómez-Martín, P. P. & González-Calero, P. A. (2004). Game-Driven Intelligent Tutoring Systems. *Entertainment Computing – ICEC 2004.* 108-113.

Graesser, A., Chipman, P., King, B., McDaniel, B. & D'Mello, S. (2007). Emotions and Learning with AutoTutor. Artificial Intelligence in Education: Building Technology Rich Learning Contexts. Proceedings of AIED2007, Los Angeles. 569-571.

Graesser, A. C., D'Mello, S. K., Craig, S. D., Witherspoon, A., Sullins, J., McDaniel, B. & Gholson, B. (2008). The Relationship between Affective States and Dialog Patterns during Interactions with AutoTutor. *Journal of Interactive Learning Research,* 19(2), 293-312.

Graesser, A. C., Jackson, G. T. & McDaniel, B. (2007). AutoTutor Holds Conversations with Learners that are Responsive to their Cognitive and Emotional States. *Educational Technology,* 47, 19-22.

Graesser, A. C., Person, N. K., Harter, D. & TRG (2001). Teaching tactics and dialog in AutoTutor *International Journal of Artificial Intelligence in Education,* 12, 257-279.

Green, P. & Wei-Haas, L. (1985). The Rapid Development of User Interfaces: Experience with the Wizard of Oz Method. *Human Factors and Ergonomics Society Annual Meeting Proceedings,* 29, 470-474.

Heidig, S. & Clarebout, G. (2011). Do pedagogical agents make a difference to student motivation and learning? *Educational Research Review,* 6**,** 27-54.

Hernández, Y. & Sucar, L. E. (2007). User study to evaluate an affective behavior model in an educational game. 13th International Conference on Artificial Intelligence in Education, AIED2007, Workshop on Modeling and Scaffolding Affective Experiences to Impact Learning, Marina del Rey California. 57-66.

Heywood, J. (2000). *Assessment in Higher Education,* UK, Jessica Kingsley Publishers.

Hinton, C., Miyamoto, K. & Della-Chiesa, B. (2008). Brain Research, Learning and Emotions: implications for education research, policy and practice. *European Journal of education,* 43**,** 87-103.

Hodges, C. B. (2004). Designing to Motivate: Motivational Techniques to Incorporate in E-Learning Experiences. *The Journal of Interactive Online Learning,* 2(3)**,** 1-7.

Howard, G. S. (1994). Why Do People Say Nasty Things About Self-Reports? *Journal of Organizational Behavior,* 15**,** 399-404.

Immordino-Yang, M. H. (2011). Implications of affective and social neuroscience for educational theory. *Educational Philosophy and Theory,* 43**,** 98-103.

Immordino-Yang, M. H. & Damasio, A. (2007). We feel, therefore we learn: The relevance of affective and social neuroscience to education. *Mind, brain, and education,* 1**,** 3-10.

J Davidson, R., Dunne, J., Eccles, J. S., Engle, A., Greenberg, M., Jennings, P., Jha, A., Jinpa, T., Lantieri, L. & Meyer, D. (2012). Contemplative practices and mental training: Prospects for American education. *Child Development Perspectives,* 6**,** 146-153.

Jackson, G. T. & Graesser, A. C. (2007). Content Matters: An Investigation of Feedback Categories within an ITS. Artificial Intelligence in Education: Building Technology Rich Learning Contexts. Proceedings of AIED2007, Los Angeles. 127-134.

Jiang, Y., Song, J., Lee, M. & Bong, M. (2014). Self-efficacy and achievement goals as motivational links between perceived contexts and achievement. *Educational Psychology,* 34**,** 92-117.

Johnson, W. L. & Rizzo, P. (2004). Politeness in Tutoring Dialogs: "Run the Factory, That's What I'd Do". *In:* LESTER, J. C., VICARI, R. M. & PARAGUAÇU, F., eds. Intelligent Tutoring Systems. Springer Berlin / Heidelberg, 206-243.

Johnson, W. L., Wu, S. & Nouhi, Y. (2004). Socially Intelligent Pronunciation Feedback for Second Language Learning. Modeling Human Teaching Tactics and Strategies Workshop in conjunction with the 7th International Conference on Intelligent Tutoring Systems, Maceio - Alagoas, Brasil.

Kelley, H. H. (1973). The Processes of Causal Attribution. *American Psychologist***,** 107-128.

Kelley, J. F. (1983). An empirical methodology for writing user-friendly natural language computer applications. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Boston, Massachusetts, USA. ACM, 193-196.

Kenny, C. & Pahl, C. (2005). Automated tutoring for a database skills training environment. Proceedings of the 36th SIGCSE technical symposium on Computer science education, St. Louis, Missouri, USA. ACM, 58-62.

Kenny, C. & Pahl, C. (2008). Personalised correction, feedback and guidance in an automated tutoring system for skills training. *International Journal of Knowledge and Learning,* 4**,** 75-92.

Khan, I. A., Brinkman, W.-P. & Hierons, R. M. (2008). Towards a Computer Interaction-Based Mood Measure Instrument. 20th Workshop of the Psychology of Programming Interest Group, University Lancaster, UK.

Kirschner, P. A., Sweller, J. & Clark, R. E. (2006). Why Minimal Guidance during Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychologist,* 41(2)**,** 75-86.

Koedinger, K., Anderson, J., Hadley, W. & Mark, M. (1997). Intelligent Tutoring Goes to School in the Big City. *International Journal of Artificial Intelligence in Education,* 8**,** 30-43.

Kölling, M., Quig, B., Patterson, A. & Rosenberg, J. (2003). The BlueJ system and its pedagogy. *Journal of Computer Science Education, Special Issue on Learning and Teaching Object Technology,* 13(4)**,** 1-12.

Kort, B., Reilly, R. & Picard, R. W. (2001). External Representation of Learning Process and Domain Knowledge: Affective State as a Determinate of its Structure and Function. AI-ED 2001, Workshop: External Represenations in AIED: Mutliple Forms and Multiple Roles, San Antonio, Texas.

Lahtinen, E., Ala-Mutka, K. & Järvinen, H.-M. (2005). A Study of the Difficulties of Novice Programmers. 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE'05), Monte de Caparica, Portugal.

Lepper, M. R., Corpus, J. H. & Iyengar, S. S. (2005). Intrinsic and Extrinsic Motivational Orientations in the Classroom: Age Differences and Academic Correlates. *Journal of Educational Psychology,* 97(2)**,** 184-196.

Lepper, M. R., Woolverton, M., Mumme, D. L. & Gurtner, J.-L. (1993). Motivational Techniques of Expert Human Tutors: Lessons for the Design of Computer-Based Tutors. *In:* LAJOIE, S. P. & DERRY, S. J. (eds.) *Computers as Cognitive Tools.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A. & Bhogal, R. S. (1997). The persona effect: affective impact of animated pedagogical agents. Proceedings of the ACM SIGCHI Conference on Human factors in computing systems. ACM, 359-366.

Lindgaard, G., Fernandes, G., Dudek, C. & Brown, J. (2006). Attention web designers: You have 50 milliseconds to make a good first impression! *Behaviour & information technology,* 25**,** 115-126.

Linn, M. C. & Dalbey, J. (1989). Cognitive Consequences of Programming Instruction. *In:* SOLOWAY, E. & SPOHRER, J. C. (eds.) *Studying the Novice Programmer.* NJ: Lawrence Erlbaum Associates, 57-81.

Linnenbrink, E. A. (2007). The role of affect in student learning: A multi-dimensional approach to considering the interaction of affect, motivation, and engagement.

Lister, R. (2011). Concrete and other neo-Piagetian forms of reasoning in the novice programmer. Proceedings of the Thirteenth Australasian Computing Education Conference-Volume 114. Australian Computer Society, Inc., 9-18.

Luckin, R. & du Boulay, B. (1999). Ecolab: The Development and Evaluation of a Vygotskian Design Framework. *International Journal of Artificial Intelligence in Education,* 10**,** 198-220.

Luckin, R. & Hammerton, L. (2002). Getting to Know Me; Helping Learners Understand Their Own Learning Needs through Metacognitive Scaffolding. Proceedings of the sixth Conference on Intelligent Tutoring Systems, Biarritz, France. Berlin: Springer, 759-771.

Ma, W., Adesope, O. O., Nesbit, J. C. & Liu, Q. (2014). Intelligent Tutoring Systems and Learning Outcomes: A Meta-Analysis. *Journal of Educational Psychology*.

Mark, M. A. & Greer, J. E. (1993). Evaluation Methodologies for Intelligent Tutoring systems. *International Journal of Artificial Intelligence in Education,* 4**,** 129-153.

Mark, M. A. & Greer, J. E. (1995). The VCR Tutor: Effective Instruction for Device Operation. *Journal of The Learning Sciences,* 4**,** 209-246.

Martin, B., Mitrovic, A., Koedinger, K. R. & Mathan, S. A. (2011). Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Interaction,* 21**,** 249-283.

Mathan, S. A. & Koedinger, K. R. (2005). Fostering the Intelligent Novice: Learning From Errors With Metacognitive Tutoring. *Educational Psychologist,* 40(4) 257-265.

Mathews, M. (2012). *A Framework for Multiple Adaptable Pedagogical Strategies in Intelligent Tutoring Systems.* Doctor of Philosophy, University of Canterbury, Christchurch, New Zealand.

Mathews, M. & Mitrovic, A. (2007). The Effect of Problem Templates on Learning in Intelligent Tutoring Systems. *In:* LUCKIN, R., KOEDINGER, K. & GREER, J., eds. Proceedings of the 13th International Conference of Artificial Intelligence in Education (AIED 2007), Los Angeles, USA.

Mathews, M. & Mitrovic, A. (2008a). Do Students Who See More Concepts in an ITS Learn More? *In:* BAKER, R. S. J. D., BARNES, T. & BECK, J. E., eds. 1st International Conference on Educational Data Mining Montreal, Quebec, Canada. 266-273.

Mathews, M. & Mitrovic, T. (2008b). How Does Students' Help-Seeking Behaviour Affect Learning? *In:* WOOLF, B. P., AÏMEUR, E., NKAMBOU, R. & LAJOIE, S., eds. 9th International Conference of Intelligent Tutoring Systems (ITS2008), Montreal, Canada. 363-372.

Mayer, R. E. (2003). Memory and Information Processes. *In:* REYNOLDS, W. M., MILLER, G. E. & WEINER, I. B. E. I. C. (eds.) *Handbook of Psychology.* 47-57.

McCormick, C. B. (2003). Metacognition and Learning. *In:* REYNOLDS, W. M., MILLER, G. E. & WEINER, I. B. E. I. C. (eds.) *Handbook of Psychology.* 79-102.

McQuiggan, S., Mott, B. & Lester, J. (2008). Modeling self-efficacy in intelligent tutoring systems: An inductive approach. *User Modeling and User-Adapted Interaction,* 18(1)**,** 81-123.

Mitrovic, A. (2007). *SQL-Tutor: an ITS for SQL programming* [Online]. Available: http://www.cosc.canterbury.ac.nz/tanja.mitrovic/sql-tutor.html [Accessed 21/08/2014].

Mitrovic, A. (2012). Fifteen years of Constraint-Based Tutors: What we have achieved and where we are going. *User Modeling and User-Adapted Interaction,* 22**,** 39-72.

Mitrovic, A. (2014). *RE: Learning Curves.* Personal Communication*,* Type to HULL, A., 24/01/2014

Mitrovic, A. & Hausler, K. (2003). An Intelligent SQL Tutor on the Web. *International Journal of Artificial Intelligence in Education,* 13**,** 171-195.

Mitrovic, A. & ICTG.Team (2006). Large-Scale Deployment of Three Intelligent Web-based Database Tutors. *Journal of Computing and Information Technology,* 14(4)**,** 275-281 (Reprinted from V. Luzar, V. Hljuz-Dobric (eds) Proc. ITI 2006, pp. 135-140, Cavtat, Croatia, 19-22.6.2006).

Mitrovic, A. & ICTG.Team. (2008). Constraint-Based Tutors. *In:* MAYERS, A., DUFRESNE, A., HEFFERNAN, N. (EDS), ed. Demo Proc. ITS 2008. 29-32.

Mitrovic, A. & Ohlsson, S. (1999). Evaluation of a Constraint-Based Tutor for a Database Language. *International Journal of Artificial Intelligence in Education,* 10**,** 238-256.

Myers, C. & Douglas, P. (2007). The Unstructured Student. 5th HEA Workshop on Teaching, Learning and Assessment of Databases, Glasgow.

Najar, A., Mitrovic, A. & Neshatian, K. (2014). Utilizing Eye Tracking to Improve Learning from Examples. *In:* STEPHANIDIS, C. & ANTONA, M. (eds.) *Universal Access in Human-Computer Interaction. Universal Access to Information and Knowledge.* Switzerland: Springer International Publishing, 410-418.

Narciss, S. (2004). The Impact of Informative Tutoring Feedback and Self-Efficacy on Motivation and Achievement in Concept Learning. *Experimental Psychology,* 51**,** 214-228.

Narciss, S., Sosnovsky, S. & Andres, E. (2014). Adapting Tutoring Feedback Strategies to Motivation. *Open Learning and Teaching in Educational Communities.* Springer, 288-301.

Návrat, P. (1994). Hierarchies of programming concepts: abstraction, generality, and beyond. *SIGCSE Bulletin,* 26(3)**,** 17-21.

Nielsen, J. & Loranger, H. (2006). *Prioritizing Web Usability* New Riders Press, Berkeley.

Nye, B. D., Graesser, A. C. & Hu, X. (2014). AutoTutor and Family: A Review of 17 Years of Natural Language Tutoring. *International Journal of Artificial Intelligence in Education,* 24**,** 427-469.

Ohlsson, S. (1994). Constraint-Based Student Modeling. *In:* GREER, J. & MCCALLA, G. (eds.) *Student Modelling: The Key to Individualized Knowledge-Based Instruction.* Springer Berlin Heidelberg, 167-189.

Pajares, F. (1997). Current Directions in Self-efficacy Research. *In:* MAEHR, M. & PINTRICH, P. R. (eds.) *Advances in motivation and achievement.* Greenwich, CT: JAI Press, 1-49.

Palumbo, D. B. (1990). Programming Language/Problem-Solving Research: A Review of Relevant Issues. *Review of Educational Research,* 60**,** 65-89.

Pane, J. F. & Myers, B. A. (2000). The Influence of the Psychology of Programming on a Language Design: Project Status Report. 12th Annual Meeting of the Psychology of Programmers Interest Group, Corigliano Calabro, Italy.

Paris, S. G. & Winograd, P. (1990). How Metacognition Can Promote Academic Learning and Instruction. *In:* JONES, B. F. & IDOL, L. (eds.) *Dimensions of Thinking and Cognitive Instruction: Implications for Educational Reform.* Hillsdale, NJ: Lawrence Erlbaum Associates, 15-51.

Pea, R. D. & Kurland, D. M. (1984). On the cognitive effects of learning computer programming. *New Idea in Psychology,* 2**,** 137-168.

Pekrun, R., Goetz, T., Frenzel, A. C., Barchfeld, P. & Perry, R. P. (2011). Measuring emotions in students' learning and performance: The Achievement Emotions Questionnaire (AEQ). *Contemporary Educational Psychology,* 36**,** 36-48.

Person, N. K. & Graesser, A. C. (2003). Fourteen facts about human tutoring: Food for thought for ITS developers. *In:* ALEVEN, V., HOPPE, U., KAY, J., MIZOGUCHI, R., PAIN, H., VERDEJO, F. & YACEF , K., eds. AIED Supplemental Proceedings, Sydney, Australia. 335-344.

Pintrich, P. (2003). Motivation and Classroom Learning. *In:* REYNOLDS, W. M., MILLER, G. E. & WEINER, I. B. E. I. C. (eds.) *Handbook of Psychology.* 103-122.

Pintrich, P. R. (2000a). An Achievement Goal Theory Perspective on Issues in Motivation Terminology, Theory, and Research. *Contemporary Educational Psychology,* 25**,** 92-104.

Pintrich, P. R. (2000b). Multiple goals, multiple pathways: The role of goal orientation in learning and achievement. *Journal of Educational Psychology,* 92**,** 544-555.

Pintrich, P. R. (2000c). The role of goal orientation in self-regulated learning. *In:* BOEKAERTS, M., PINTRICH, P. R. & ZEIDNER, M. (eds.) *Handbook of self-regulation: Theory, research, and applications.* San Diego, CA.: Academic Press, 451-502.

Pólya, G. (1945). *How to Solve It: A New Aspect of Mathematical Method,* Princeton, N.J., Princeton University Press.

Porayska-Pomsta, K., Mavrikis, M., D'Mello, S., Conati, C. & Baker, R. (2013). Knowledge Elicitation Methods for Affect Modelling in Education. *International Journal of Artificial Intelligence in Education,* 22**,** 107-140.

Porayska-Pomsta, K., Mavrikis, M., D'Mello, S., Conati, C. & Baker, R. (2012). Contemporary Methods for Learner Affect Modelling. *In:* DU BOULAY, B., LUCKIN, R., PORAYSKA-POMSTA, K. & ARROYO, I. (eds.) *In the Mood for Learning.* Brighton: Sussex University, Department of Informatics. Cogsci Research Paper number: 603.

Porayska-Pomsta, K. & Mellish, C. (2013). Modelling human tutors' feedback to inform natural language interfaces for learning. *International Journal of Human-Computer Studies,* 71**,** 703-724.

Prior, J. C. & Lister, R. (2004). The backwash effect on SQL skills grading. 36**,** 32-36.

Reeve, J. (2005). *Understanding Motivation and Emotion,* USA, John Wiley & Sons Inc.

Reis, C. & Cartwright, R. (2004). Taming a Professional IDE for the Classroom. Proc. 35th SIGCSE Technical Symposium on Computer Science Education, Norfolk, VA. ACM Press, New York, 156-160.

Renkl, A. & Atkinson, R. K. (2003). Structuring the Transition From Example Study to Problem Solving in Cognitive Skill Acquisition: A Cognitive Load Perspective. *Educational Psychologist,* 38(1)**,** 15-22.

Reynolds, W. M. & Miller, G. E. (2003). Current Perspectives in Educational Psychology. *In:* REYNOLDS, W. M., MILLER, G. E. & WEINER, I. B. E. I. C. (eds.) *Handbook of Psychology.* 3-20.

Robertson, S. I. (2001). *Problem Solving,* Hove, East Sussex, Psychology Press Ltd.

Robins, A., Rountree, J. & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education,* 13(2)**,** 137-172.

Robins, D. & Holmes, J. (2008). Aesthetics and credibility in web site design. *Information Processing & Management,* 44**,** 386-399.

Rodrigo, M. M. T., Rebolledo-Mendez, G., Baker, R. S. J., du Boulay, B., Sugay, J. O., Lim, S. A. L., Espejo-Lahoz, M. B. & Luckin, R. (2008). The Effects of Motivational Modeling on Affect in an Intelligent Tutoring System. 16th International Conference on Computers in Education, ICCE 2008, Taipei, Taiwan. 57-64.

Ryan, R. M. & Deci, E. L. (2000). Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions. *Contemporary Educational Psychology,* 25**,** 54-67.

Sarrafzadeh, A., Alexander, S., Dadgostar, F., Fan, C. & Bigdeli, A. (2008). "How do you know that I don't understand?" A look at the future of intelligent tutoring systems. *Computers in Human Behavior,* 24**,** 1342-1363.

Schunk, D. H. (1991). Self-Efficacy and Academic Motivation. *Educational Psychologist,* 26(3)**,** 207-231.

Schunk, D. H., Pintrich, P. R. & Meece, J. L. (2007). *Motivation in Education: Theory, Research, and Applications,* Upper Saddle River, NJ, Pearson Merrill Prentice Hall.

Schwarz, N. (1999). Self-reports: how the questions shape the answers. *American psychologist,* 54**,** 93.

Shute, V. J. & Regian, J. W. (1993). Principles for Evaluating Intelligent Tutoring Systems. *International Journal of Artificial Intelligence in Education,* 4**,** 245-271.

Simon, S., Fincher, S., Robins, A., Baker, B., Box, I., Cutts, Q., de Raadt, M., Haden, P., Hamer, J., Hamilton, M., Lister, R., Petre, M., Sutton, K., Tolhurst, D. & Tutty, J. (2006). Predictors of Success in a First Programming Course. Eighth Australasian Computing Education Conference (ACE2006), Hobart, Australia.

Spohrer, J. C. & Soloway, E. (1989). Novice Mistakes: Are the Folk Wisdoms Correct? *In:* SOLOWAY, E. & SPOHRER, J. C. (eds.) *Studying the Novice Programmer.* Hillsdale, NJ: Lawrence Erlbaum, 401-416.

Sykes, E. R. & Franek, F. (2007). Developmental Process Model for the Java Intelligent Tutoring System. *Journal of Interactive Learning Research.,* 18(3), 399–410.

Thomson, D. & Mitrovic, A. (2010). Preliminary Evaluation of a Negotiable Student Model in a Constraint-Based ITS. *Research and Practice in Technology Enhanced Learning,* 5, 19-33.

van der Stel, M. & Veenman, M. V. J. (2010). Development of metacognitive skillfulness: A longitudinal study. *Learning and Individual Differences,* 20, 220-224.

VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist,* 46, 197-221.

Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A. & Wintersgill, M. (2005). The Andes Physics Tutoring System: Lessons Learned. *Int. J. Artif. Intell. Ed.,* 15, 147-204.

Vickers, P. (2008). *How To Think Like A Programmer: Problem-solving for the Bewildered*, Thomson Learning.

Vollmeyer, R. & Rheinberg, F. (1999). Motivation and metacognition when learning a complex system. *European Journal of Psychology of Education,* 14, 541-554.

Vollmeyer, R. & Rheinberg, F. (2013). The role of motivation in knowledge acquisition. *International Handbook of Metacognition and Learning Technologies.* Springer, 697-707.

Weber, G. (1996a). Episodic Learner Modeling. *Cognitive Science,* 20, 195-236.

Weber, G. (1996b). Individual Selection of Examples in an Intelligent Learning Environment. *Journal of Artificial Intelligence in Education,* 7(1), 3-31.

Weber, G. & Brusilovsky, P. (2001). ELM-ART: An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education,* 12, 351-384.

Weinberg, G. M. (1971). *The Psychology of Computer Programming,* New York, Van Nostrand Reinhold.

Weiner, B. (1985). An Attributional Theory of Achievement Motivation and Emotion. *Psychological Review,* 92(4), 548-573.

Weiner, B. (2005). Motivation from an Attributional Perspective and the Social Psychology of Perceived Competence. *In:* ELLIOTT, A. J. & DWECK, C. S. (eds.) *Handbook of Competence and Motivation.* The Guilford Press, 73-84.

Wigfield, A. & Eccles, J. S. (2000). Expectancy-Value Theory of Achievement Motivation. *Contemporary Educational Psychology,* 25, 68-81.

Wilson, K. & Narayan, A. (2014). Relationships among individual task self-efficacy, self-regulated learning strategy use and academic performance in a computer-supported collaborative learning environment. *Educational Psychology*, 1-18.

Zimmerman, B. J. (2000). Self-Efficacy: An Essential Motive to Learn. *Contemporary Educational Psychology,* 25, 82-91.

Appendix A Study Timeline

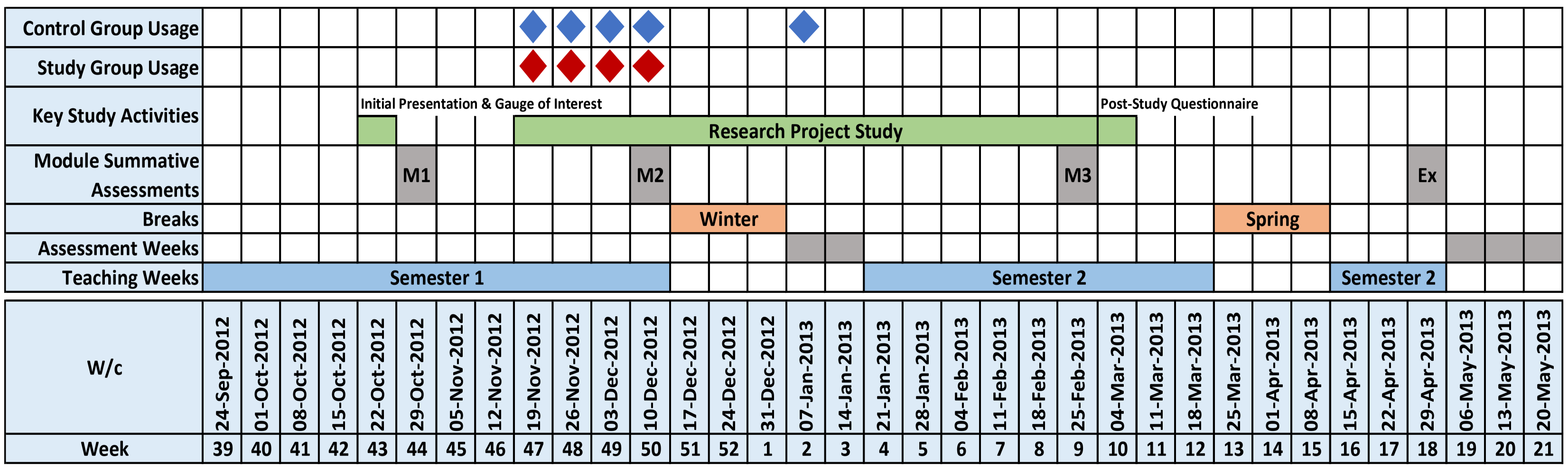


Figure A.1 Research Project Timeline

Figure A.1 outlines the timeline for the study, showing how the study fitted into the academic year and the summative assessments for the database module where the study participants were registered. The timeline also shows the weeks where the study participants logged into and used SQL-Tutor.

Appendix B Ethical Approval

This appendix contains copies of the documentation submitted as part of the application for the ethical review process at both the Universities of Sussex and Northumbria. Ethical approval was applied for at the University of Sussex first as the University where the author was registered for the Doctorate degree. The result of this approval process, including the approval certificate (included in this appendix), were used to gain ethical approval from the University of Northumbria as the University were the student volunteers were registered on undergraduate computing degree courses.

The application for ethical approval at the University of Sussex was an on-line process with the following documents contained in this appendix also being supplied as part of the application.

The final two documents in this appendix are the approval certificates from the University of Sussex. The initial approval certificate was issued in October 2012, with the second certificate issued in April 2013. The second certificate was issued to extend the ethical approval period from March to the end of April 2013 in order to provide more time for study participants to complete the online post-study questionnaire (refer to B3 (p.211) and Appendix F (p.235) for the questionnaire).

Following the approval from the University of Sussex, the same documentation including the approval certificate (B5a) was submitted as part of the on-line approval process at the University of Northumbria. As the representative who was internal to Northumbria, the module tutor of the first year database module on which the study volunteers were studying was the primary contact for the ethical approval application at Northumbria. Approval was granted by the review panel on the 12th November 2012 with the following comments

> *"The researcher has fully discussed this proposal and sought ethics advice for the study. The only concern raised was to ensure that no students would be disadvantaged from taking part (or not participating) in the study - as this could have an impact on the module performance and grades. As this study is open to everyone, and is also part of a self-directed programme, all assurances/conditions have been met. The only 'grey' area might be that some students will be part of a control group and perhaps will not benefit from the extra features of the SQL tutor system. As stated above however, this is not linked directly to any summative assessment and only to independent learning. Therefore this seems a reasonable research strategy."*
>
> University of Northumbria ethical approval feedback (2012)

B1 Ethical Approval Application Questions

This document was submitted as part of the ethical review process at both the Universities of Sussex and Northumbria. The actual questions are part of the review application at the University of Sussex.

University of Sussex Ethical Approval

Section A

Project Start Date – 15th October 2012

Project End Date – 8th March 2013

External Funding in Place – n/a

External Collaborators - School of Computing, Engineering and Information Sciences (CEIS), Northumbria University

Funder/Project Title – Scaffolding Metacognitive & Motivational Feedback in a Problem-Solving Domain

Name of Funder – n/a

Project Description

The aim of the research is to improve the learner's focus on the process and experience of problem-solving by incorporating motivational and metacognitive feedback based on the learner's past experience and motivational states into an existing Intelligent Tutoring System (ITS); SQL-Tutor. SQL-Tutor provides learner's with a Web-based tool for practicing SQL (Structured Query Language), which is a language for querying data in relational databases.

The study involves dividing the participants into two groups; the control group will use the base SQL-Tutor which provides domain only feedback and the test group will use an extended version of SQL-Tutor which provides the motivational and metacognitive feedback in addition to the domain feedback. The test version will prompt the participants to self-report, via a Likert scale, to gauge their self-efficacy of SQL and the questions the ITS prompts them to solve. The questions and answers used by the ITS will be the same across both versions.

The participants will be volunteers from a first year (Level 4) undergraduate database module taught at the Northumbria University. The study will require the participants to use the ITS over a period of time. Apart from an initial class-based session to provide participants with user IDs, demonstrate how to use the tutor and to answer any questions that may arise, the participants will use the tutor in their own time and at their own pace. At the end of the study, participants will be asked to complete a questionnaire, the aim of which is to gauge their opinions on using the ITS and the types of feedback given by the ITS.

The completed questionnaires, along with the log files from the ITS will be analysed to determine whether the use of motivational and metacognitive feedback has aided the participants focus and experience of problem solving. The ITS-generated log files record user IDs and user actions. No personal data is recorded in the log files and no individual participant can be identified from the log file data. As part of the databases module, there are three milestone, in-class tests which form part of the summative assessment of the module. The marks of these tests will also be used by the study; the marks from the first test will be used to ensure an even spread of participants between the control and test groups. The marks from the other two tests will aid the analysis of participant progress. To ensure complete anonymity, only the Module Tutor at Northumbria University will know the relationship between participant and SQL-Tutor user ID. All data (including questionnaires) will only be identifiable in the study, and known to the PhD student, by the SQL-Tutor user ID.

Section B

B1 Data Collection and Analysis

1. **PARTICIPANTS: How many people do you envisage will participate, who they are, and how will they be selected?**
   The participants will be volunteers from the first year (Level 4) database module within the School of Computing, Engineering and Information Sciences (CEIS) at Northumbria University. There are expected to be approximately 300 students taking the module this academic year. There is no limit to the number of volunteers that will be accepted to participate in the student, especially as it is expected that data collected from some participants may eventually be discounted (i.e. due to none completion of the end of study questionnaire, in-appropriate or in-adequate use of the tutoring system).

2. **RECRUITMENT: How will participants be approached and recruited?**
   First year (Level 4) undergraduate students on a database module at Northumbria University will be asked to volunteer to participate in the study. The study design and schedule has already been discussed with the Module Tutor and the Subject Head at the University. I will give a 10-minute talk at the end of the module's lecture in week commencing 15th October 2012 in order to introduce the study and ask for volunteers. Students will then be able to volunteer over a two-week period.

3. **METHOD: What research method(s) do you plan to use; e.g. interview, questionnaire/self-completion questionnaire, field observation, audio/audio-visual recording?**
   The study involves the participants using an Intelligent Tutoring System as a Web-based tool that allows them to practise writing SQL SELECT queries. A questionnaire will be given to all participants at the end of the study. The study will also analyse log files created by the tutoring system, as well as marks from the module's three in-class tests (summative).

4. **LOCATION: Where will the project be carried out e.g. public place, in researcher's office, in private office at organisation?**
   Apart from an initial class-based session to demonstrate how to use the tutoring system, the participants will use the tutoring system in their own time and at their own pace. As a Web-based system it gives the participants the full flexibility of when and where they use it.

B2 Confidentiality and Anonymity

5. **Will questionnaires be completed anonymously and returned indirectly?**
   The end of study questionnaire will be completed anonymously, with only the participant's SQL-Tutor user ID being entered as identification on the questionnaire. The aim is to create an on-line questionnaire, so submission will be indirect.

6. **Will questionnaires and/or interview transcripts only be identifiable by a unique identifier (e.g. code/pseudonym)?**
   The participants will only be known to the study organiser by their SQL-Tutor user ID. This also applies to all of the data used in the study. Only the Module Tutor at Northumbria University will know the name of the student that relates to a given SQL-Tutor user ID (this information will never been known or made available to the study organiser). Similarly the Module Tutor will only know the internal module data (i.e. module test results) and will not know or have access to any data relating to individuals from the study.

7. **Will lists of identity numbers or pseudonyms linked to names and/or addresses be stored securely and separately from the research data?**
   Yes. As mentioned in question 6, the list that links named and SQL-Tutor user IDs will only be held by the Module Tutor at Northumbria University. This list is completely separate to the research data held from the study organiser.

   The study will only hold the following data about each participant. From the data held by the study, no one student can be identified.
   - SQL-Tutor user ID
   - Gender
   - Age
   - Course route (the database module is a generic module taught to all first year computing and business information students in the School of Computing, Engineering and Information Sciences). The course route is purely held so that participants from extreme discipline courses are not directly compared (i.e. so that forensic science students are not compared with business information system students).
   - The results from the four in-class, summative tests from the module.
   - The log files produced by the tutoring system.
   - The end of study questionnaire.

8. **Will all place names and institutions which could lead to the identification of individuals or organisations be changed?**
   Northumbria University will be named as the institution where the participants came from, but this would not be enough to identify any individuals that take part in the study.

9. **Will all personal information gathered be treated in strict confidence and never disclosed to any third parties?**
   Yes. As mentioned in questions 6 & 7, this study will not hold data that will allow any individual to be identified. All data held will only be used within the study and not shared beyond that.

10. **Can you confirm that your research records will be held in accordance with the data protection guidelines (see guidelines on research governance website)?**
    Yes. The data will even be anonymised to the study organiser. All data will be held on a secure, password and anti-virus protected computer system. The data will not be held in a public domain of any sort.

11. **Can you confirm that you will not use the research data for any purpose other than that which consent is given?**
    Yes. The research data will only be used for the purposes of the study which is part of a PhD and any subsequent publications arising from the research.

B3 Informed Consent and Recruitment of Participants

12. **Will all respondents be given an Information Sheet and be given adequate time to read it before being asked to agree to participate?**
    Yes. Once a student as volunteered to participate they will be given an information sheet prior to the initial class-based session that demonstrates how to use the tutoring system. The information sheet will be given to the participants at least two-days before the introductory session. At this session participants will be required to sign a consent form. The introductory session (scheduled for week commencing the 12th November 2012) would mark the start of the study in terms of the participants actively using the tutoring system.

13. **Will all participants taking part in an interview, focus group, observation (or other activity which is not questionnaire based) be asked to sign a consent form? If you are obtaining consent another way, please explain under 15a below.**
    Yes. Please refer to the answer given to question 12.

14. **Will all participants self-completing a questionnaire be informed that returning the completed questionnaire implies consent to participate?**
    The questionnaire is an end of study questionnaire, so this is not applicable. A separate consent form will be issued for signing at the start of the study.

15. **Will all respondents be told that they can withdraw at any time, ask for their data to be destroyed and/or removed from the project until it is no longer practical to do so?**
    A participant can formally withdraw from the study at any point that they are actively using the tutoring system (from week commencing the 12th November 2012 to the week commencing the 7th March 2013). If no formal withdrawal request is made, but the participant fails to complete the end of study questionnaire, then that also would mean that data relating to that participant would be discounted from the two groups used in the study.

B4 Context

16. **Is Criminal Records Bureau clearance necessary for this project? If yes, please ensure you complete the next question.**
    No.

17. **(17a) Are any other ethical clearances or permissions required?**
    No.

18. **Does the research involve any fieldwork - Overseas or in the UK?**
    No.

19. **(19a) Will any researchers be in a lone working situation?**
    No.

B5 Any further concerns

20. **(20a) Are there any other ethical considerations relating to your project which have not been covered above?**
    Following initial meetings with the Module Tutor at Northumbria University, it was suggested to use a draw for participants; all participants taking part in the study and who submit the end of study questionnaire will be placed in a draw to win a voucher (i.e. iTunes or Amazon).

    Every consideration has been taken so as not to disadvantage either participants and none-participants of this study in relation to the course module. The databases and questions contained in the tutoring system do not directly relate to those used in the summative, in-class tests of the module. The tutoring system does not teach SQL SELECT queries (that is covered by the lectures and learning material of the module), but merely provides a tool by which participants can practice writing SQL SELECT queries. None-participants from the module may decide to use other such practice tools that may be available (i.e. on the Web).

B2 Study Overview Sheet Given to the Study Volunteers

This is the contents of a study overview sheet that was supplied to the study volunteers prior to the start of the study and the consent forms being issued.

SQL-Tutor Study Questions and Answers

The information provided here compliments the "SQL-Tutor Overview" document and the information that was presented at the beginning of lectures in week 5 of Semester 1.

NOTE: If you have any questions regarding the study and SQL-Tutor that is not covered either here or in the documents, please do not hesitate to either email the study organiser, Alison Hull (A.Hull@sussex.ac.uk) or your Module Tutor; Emma-Jane Philips.

What is the aim of the study?

The study is part of a research project being undertaken as part of a postgraduate research degree (a PhD). The aim of the study is to see how helpful SQL-Tutor is.

What is a tutoring system?

The tutoring system that is being used as part of the study is called SQL-Tutor. It is a Web-based system that provides an environment that you can use to learn and practice writing SQL SELECT queries to solve problems. SQL-Tutor is more than an e-learning system, as it adapts to each user. SQL-Tutor creates and maintains a student model for each user, which helps the system to adapt and make suggestions based on what it thinks your current level of SQL knowledge is. The tutoring system determines your level of knowledge of your previous use of the system and the problems you have solved or attempted to solve.

SQL-Tutor was originally developed in the 1990's and is available commercially to organisations such as schools and colleges to use with their students. The version that you will access to has been adapted to meet the aims of this study.

Where can I use SQL-Tutor?

Anywhere where you have access to the internet. SQL-Tutor is a Web-based system which provides you with full flexibility as to not only the pace you attempt the problems, but also when and where choose to use it.

Do I have to attempt the problem that SQL-Tutor suggests?

No. While SQL-Tutor makes suggestions on what clause of the SQL SELECT statement to focus on and on possible problems to attempt next, you do not need to follow the suggestions. They are only *suggestions* and not mandatory next steps.

How long will I have access to SQL-Tutor?

You will have access to use SQL-Tutor as often as you like for the full duration of the study; from week commencing 12th November 2012 until week commencing the 7th March 2013.

Will taking part in the study and using SQL-Tutor involve extra work?

It is hoped that only a very small amount of extra time will be required. At the start of the study a short (approximately 30 minutes) session will take place where you will be given your SQL-Tutor user ID and shown how to use the tutor. You will also be required to sign a consent form to indicate that you are willing to be part of the study (the form is the standard form used by the School). You will also be expected to complete a short questionnaire at the end of the study to gauge your opinions about using SQL-Tutor to learn and practice SQL SELECT queries.

The actual time that you spend using SQL-Tutor throughout the study can be used to form part of the 5hrs per week independent study that is expected by the CM0429 Database module as a 20-credit, year-long module.

How long and how often am I expected to use SQL-Tutor?

The study does not have any strict times for using SQL-Tutor. The day & time is fully flexible to work around your schedule, you can practice SQL SELECT queries with SQL-Tutor as often as you want. However there are general guidelines; use SQL-Tutor at least weekly (although over the duration of the study there will be expectations to this, i.e. Christmas & New Year). Each use (session) of SQL-Tutor should be long enough to actually use it to gain some benefit from it (i.e. logging on for only 10 minutes and not attempting any problems, is not beneficial to you for learning and practicing queries).

What happens if I do not want to or I change my mind about taking part in the study and use SQL-Tutor?

Taking part in the study is voluntary, so if you do not want to take part and use SQL-Tutor, then don't volunteer. If you have either indicated your interest in taking part in the study via the 1-question survey on Blackboard or at any time during the study, <u>you can withdraw from study by simply sending the study organiser an email.</u>

Unfortunately only people taking part in the study will have access to use SQL-Tutor. It is purely a tool to provide an environment for practicing solving SQL problems, which can be used as part of the independent study part of this module. If you do not take part in the study, for practice and as part of your independent study you can look for and use other resources that are available via the internet and other sources. For example, www.w3schools.com (which is also linked to from inside SQL-Tutor) provides a section to explain SQL SELECT statements and provides some opportunities to try queries.

What will happen to the data collected by the study (i.e. the log files from SQL-Tutor)?

The data will only be used by this study to determine whether the aim of the study has been achieved or not (whether the different feedback provided by SQL-Tutor has been helpful to learn and practice SQL SELECT queries). **All data will be fully anonymous – it cannot be associated back to any named individual.**

While the study organiser will know the names of the people that have agreed to take part in the study and use SQL-Tutor, they will not know which SQL-Tutor user ID you are using. The Module Tutor will assign each of you a user ID and will be the only person who knows this link between actual names and user IDs. Likewise the Module Tutor will not have access to any data created by SQL-Tutor (i.e. log files and student model).

B3 Sample of the Questions in the Post-Study Questionnaire

This the questionnaire document submitted in support of the ethical approval application (refer to Appendix F (p.235) for screen prints of the actual on-line questionnaire)

SQL-Tutor: Post-Study Questionnaire

The aim of the questionnaire is to gauge the participants' views on using the tutoring system and the feedback that the system provided. The questionnaire will be given at the end of the study (week commencing the 7th March 2013).

The plan is to present the questionnaire electronically; however below are the proposed questionnaire questions which consist of open (Section B) and closed (Section A) questions. The questions gauge the participants' opinions on procedural & conceptual knowledge of relational databases and SQL SELECT queries, as well as the different type of feedback provided, the accuracy of the participants' self-efficacy judgements and any motivation gained from using SQL-Tutor.

SQL-Tutor User ID: ....

Section A

For each of the statements below, circle the number that reflects how much you either disagree or agree with each statement.

		Disagree.................................Agree
1.	The SQL Tutor helped me to learn how to structure database queries	1 ....2 ....3........... 4............5
2.	The SQL Tutor helped me to learn database concepts	1 ....2 ....3........... 4............5
3.	The SQL Tutor helped me to find where my mistakes were	1 ....2 ....3........... 4............5
4.	The SQL Tutor helped me to overcome difficulties	1 ....2 ....3........... 4............5
5.	SQL-Tutor helped me to solve problems by relating them to previous problems solved	1 ....2 ....3........... 4............5
6.	The SQL Tutor helped me to gain an accurate assessment of what I understood and what I did not	1 ....2 ....3........... 4............5
7.	The SQL Tutor helped me keep going when things were tough	1 ....2 ....3........... 4............5

Section B

1. What did you like most about using the SQL-Tutor?

2. What did you like least about using the SQL-Tutor?

3. What did you think of the feedback comments from the SQL-Tutor?

4. Please mention any other aspect of the SQL-Tutor that occurs to you.

5. Did you use anything else to help you practice SQL SELECT queries outside of the timetabled lecture and lab session? If yes, what else did you use and why.

Section C

Each SQL-Tutor user ID from completed questionnaires will automatically be placed in a draw for 1 of 2 £25 vouchers.

☐ Please place X in the box if you do not want your SQL-Tutor user ID to be entered into the draw.

Indicate what type of voucher you would prefer if your SQL-Tutor user ID is drawn.

☐ Amazon.co.uk ☐ iTunes

B4 Example of the Consent Form Completed by the Study Volunteers

Northumbria University
CEIS Research Ethics Sub-Committee
CONSENT FORM – C

Project Title:

Name of the Researcher or Project Consultant:

Name of participant:

Participating Organisation:

I consent to take part in this project. ☐

I have had the project explained to me by the researcher/ consultants **and been given an information sheet.** I have read and understand the purpose of the study. ☐

I am willing to be interviewed. ☐

I understand and am happy that the discussions I will be involved in may be audio-taped and notes will be taken. ☐

I understand I can withdraw my consent at any time, without giving a reason and without prejudice. ☐

I know that my name and details will be kept confidential and will not appear in any printed documents. ☐

- The tapes and any personal information will be kept secure and confidential. They will be kept by the researcher/project consultants until the end of the project. They will then be disposed of in line with Northumbria University's retention policy.
- Anonymised summaries (if required) will be produced from the discussions to be used in the project report and in other publications. None of the participants will be identified in the project report or in other publications based on this project. Copies of any reports or publications will be available on request to participants.

I have been given a copy of this Consent Form.

Signed: **Date:**

Researcher/Project consultant: I confirm that I have explained the project to the participant and have given adequate time to answer any questions concerning it.

Signed: **Date:**

B5a The Initial Approval Certificate from the University of Sussex

University of Sussex

Sciences and Technology
Cross-Schools Research Ethics Committee

CERTIFICATE OF APPROVAL

Reference Number:	BdBAH1012 / ER/AH258/1
Title of Project:	Scaffolding Metacognitive & Motivational Feedback in a Problem-Solving Domain
Principal Investigator:	Ben du Boulay
Student:	Alison Hull
Collaborators:	-
Duration of Approval	4 months
Expiration of Approval:	08/03/2013
Expected Start Date:*	15/10/2012

This project has been given ethical approval by the Science and Technology Cluster Research Ethics Committee (C-REC).

*NB. If the actual project start date is delayed beyond 12 months of the expected start date, this Certificate of Approval will lapse and the project will need to be reviewed again to take account of changed circumstances such as legislation, sponsor requirements and University procedures.

Please note and follow the requirements for approved submissions:

Amendments to protocol.
- Any changes or amendments to approved protocols must be submitted to the C-REC for authorisation prior to implementation.

Feedback regarding the status and conduct of approved projects
- Any incidents with ethical implications that occur during the implementation of the project must be reported immediately to the Chair of the C-REC.

The principal investigator is required to provide a brief annual written statement to the committee, indicating the status and conduct of the approved project. These reports will be reviewed at the annual meeting of the committee. A statement by the Principal Investigator to the C-REC indicating the status and conduct of the approved project will be required on the following date(s):

December 2012.

Authorised Signature	Richard de Visser
Name of Authorised Signatory (C-REC Chair or nominated deputy)	**Richard de Visser**
Date	31 / 10 / 2012

B5b The Extended Approval Certificate from the University of Sussex

University of Sussex

Sciences and Technology
Cross-Schools Research Ethics Committee

CERTIFICATE OF APPROVAL

Reference Number:	BdBAH1012 / ER/AH258/1
Title of Project:	Scaffolding metacognitive & motivational feedback in a problem-solving domain
Principal Investigator:	Ben du Boulay
Student:	Alison Hull
Collaborators:	-
Duration of Approval	7 months
Expected Start Date:*	15 October 2012
Expiration of Approval:	30 April 2013

This project has been given ethical approval by the Science and Technology Cluster Research Ethics Committee (C-REC).

*NB. If the actual project start date is delayed beyond 12 months of the expected start date, this Certificate of Approval will lapse and the project will need to be reviewed again to take account of changed circumstances such as legislation, sponsor requirements and University procedures.

Please note and follow the requirements for approved submissions:

Amendments to protocol.
- Any changes or amendments to approved protocols must be submitted to the C-REC for authorisation prior to implementation.

Feedback regarding the status and conduct of approved projects
- Any incidents with ethical implications that occur during the implementation of the project must be reported immediately to the Chair of the C-REC.

The principal investigator is required to provide a brief annual written statement to the committee, indicating the status and conduct of the approved project. These reports will be reviewed at the annual meeting of the committee. A statement by the Principal Investigator to the C-REC indicating the status and conduct of the approved project will be required on the following date(s):

December 2013.

Authorised Signature	Richard de Visser
Name of Authorised Signatory (C-REC Chair or nominated deputy)	**Richard de Visser**
Date	03/04/13

Appendix C Content Rules

The content rules as stored in the 'content.rul' file and loaded into SQL-Tutor at runtime.

ss = session start, ip = in-problem, eop = end of problem, se = session end

```
(ss ((if (logon one) then (add-form (M1)) (add-form (Q1)))
     (if (logon two) then (add-form (M2)))
     (if (logon two) (attempted no) (last-se none) then (add-form (M3))
          (add-form (M6)) (add-form (Q0)))
     (if (logon two) (attempted no) (last-se low) then (add-form (M3))
          (add-form (M4)) (add-form (Q3)))
     (if (logon two) (attempted no) (last-se middle) then (add-form (M3))
          (add-form (M5)) (add-form (Q4)))
     (if (logon two) (attempted no) (last-se high) then (add-form (M3))
          (add-form (M7)) (add-form (Q4)))
     (if (logon two) (attempted yes) (last-se none) (perform high) then
          (add-form (M6)) (add-form (M8)) (add-form (Q0)))
     (if (logon two) (attempted yes) (last-se low) (perform high) then
          (add-form (M8)) (add-form (M4)) (add-form (Q5)))
     (if (logon two) (attempted yes) (last-se middle) (perform high)
          then (add-form (M8)) (add-form (M5)) (add-form (Q5)))
     (if (logon two) (attempted yes) (last-se high) (perform high) then
          (add-form (M8)) (add-form (M7)) (add-form (Q6)))
     (if (logon two) (attempted yes) (last-se none) (perform middle)
          then (add-form (M6)) (add-form (M9)) (add-form (Q0)))
     (if (logon two) (attempted yes) (last-se low) (perform middle) then
          (add-form (M9)) (add-form (M4)) (add-form (Q7)))
     (if (logon two) (attempted yes) (last-se middle) (perform middle)
          then (add-form (M9)) (add-form (M5)) (add-form (Q7)))
     (if (logon two) (attempted yes) (last-se high) (perform middle)
          then (add-form (M9)) (add-form (M7)) (add-form (Q8)))
     (if (logon two) (attempted yes) (last-se none) (perform low) (effort
          good) then (add-form (M6)) (add-form (M10)) (add-form (Q0)))
     (if (logon two) (attempted yes) (last-se none) (perform low) (effort
          poor) then (add-form (M6)) (add-form (M14)) (add-form (Q0)))
     (if (logon two) (attempted yes) (last-se low) (perform low) (effort
          good) then (add-form (M4)) (add-form (M10)) (add-form (Q3)))
     (if (logon two) (attempted yes) (last-se low) (perform low) (effort
          poor) then (add-form (M4)) (add-form (M15)) (add-form (M16))
          (add-form (Q3)))
     (if (logon two) (attempted yes) (last-se middle) (perform low)
          (effort good) then (add-form (M5)) (add-form (M10)) (add-
          form (Q9)))
     (if (logon two) (attempted yes) (last-se middle) (perform low)
          (effort poor) then (add-form (M5)) (add-form (M15)) (add-
          form (M16)) (add-form (Q9)))
     (if (logon two) (attempted yes) (last-se high) (perform low) (effort
          good) then (add-form (M7)) (add-form (M10)) (add-form (Q8)))
     (if (logon two) (attempted yes) (last-se high) (perform low) (effort
          poor) then (add-form (M7)) (add-form (M17)) (add-form (Q8)))
     (if (logon xth) then (add-form (M11)))
     (if (logon xth) (attempted no) (last-se none) (session short)  then
          (add-form (M6)) (add-form (M12)) (add-form (Q0)))
     (if (logon xth) (attempted no) (last-se none) (session ok)  then
          (add-form (M6)) (add-form (M13)) (add-form (Q0)))
```

```
      (if (logon xth) (attempted no) (last-se low) (session short)  then
             (add-form (M12)) (add-form (M4)) (add-form (Q3)))
      (if (logon xth) (attempted no) (last-se low) (session ok)  then
             (add-form (M13)) (add-form (M4)) (add-form (Q3)))
      (if (logon xth) (attempted no) (last-se middle) (session short)
             then (add-form (M12)) (add-form (M5)) (add-form (Q3)))
      (if (logon xth) (attempted no) (last-se middle) (session ok)  then
             (add-form (M13)) (add-form (M5)) (add-form (Q3)))
      (if (logon xth) (attempted no) (last-se high) (session short)  then
             (add-form (M12)) (add-form (M7)) (add-form (Q3)))
      (if (logon xth) (attempted no) (last-se high) (session ok)  then
             (add-form (M13)) (add-form (M7)) (add-form (Q3)))
      (if (logon xth) (attempted yes) (last-se none) (perform high)  then
             (add-form (M6)) (add-form (M8)) (add-form (Q0)))
      (if (logon xth) (attempted yes) (last-se low) (perform high)  then
             (add-form (M4)) (add-form (M8)) (add-form (Q5)))
      (if (logon xth) (attempted yes) (last-se middle) (perform high)
             then (add-form (M5)) (add-form (M8)) (add-form (Q5)))
      (if (logon xth) (attempted yes) (last-se high) (perform high)  then
             (add-form (M7)) (add-form (M8)) (add-form (Q6)))
      (if (logon xth) (attempted yes) (last-se none) (perform middle)
             then (add-form (M6)) (add-form (M9)) (add-form (Q0)))
      (if (logon xth) (attempted yes) (last-se low) (perform middle)  then
             (add-form (M9)) (add-form (M4)) (add-form (Q7)))
      (if (logon xth) (attempted yes) (last-se middle) (perform middle)
             then (add-form (M5)) (add-form (M9)) (add-form (Q7)))
      (if (logon xth) (attempted yes) (last-se high) (perform middle)
             then (add-form (M7)) (add-form (M9)) (add-form (Q8)))
      (if (logon xth) (attempted yes) (last-se none) (perform low) (effort
             good) then (add-form (M6)) (add-form (M10)) (add-form (Q0)))
      (if (logon xth) (attempted yes) (last-se none) (perform low) (effort
             poor) then (add-form (M6)) (add-form (M10)) (add-form (Q0)))
      (if (logon xth) (attempted yes) (last-se low) (perform low) (effort
             good) then (add-form (M4)) (add-form (M10)) (add-form (Q3)))
      (if (logon xth) (attempted yes) (last-se low) (perform low) (effort
             poor) then (add-form (M4)) (add-form (M15)) (add-form (M16))
             (add-form (Q3)))
      (if (logon xth) (attempted yes) (last-se middle) (perform low)
             (effort good) then (add-form (M5)) (add-form (M10)) (add-
             form (Q9)))
      (if (logon xth) (attempted yes) (last-se middle) (perform low)
             (effort poor) then (add-form (M5)) (add-form (M15)) (add-
             form (M16)) (add-form (Q9)))
      (if (logon xth) (attempted yes) (last-se high) (perform low) (effort
             good) then (add-form (M7)) (add-form (M10)) (add-form (Q8)))
      (if (logon xth) (attempted yes) (last-se high) (perform low) (effort
             poor) then (add-form (M7)) (add-form (M17)) (add-form (Q8)))
))


(ip ((if (state unsolved) (visits one) (attempts OR one two) (help-level
             OR three five) then (add-form (M100)) (add-form (M102)))
      (if (state unsolved) (visits xth) (attempts OR one two) (help-level
             OR three five) then (add-form (M105)) (add-form (M102)))
      (if (state unsolved) (attempts xth) (help-level OR three five) then
             (add-form (M111)))
      (if (state unsolved) (attempts xth) (help-level notpartfull)
             (changed no) then (add-form (M101)) (add-form (M205)))
      (if (state unsolved) (attempts xth) (help-level notpartfull)
             (entered no) then (add-form (M106)) (add-form (M205)))
```

```
      (if (state unsolved) (attempts xth) (help-level notpartfull)
            (entered yes) (changed yes) (errors many) then (add-form
            (M107)))
      (if (state unsolved) (attempts pretrigger) (help-level notpartfull)
            then (add-form (M108)) (add-form (M205)))
      (if (state unsolved) (attempts pretrigger) (help-level notpartfull)
            (rating higher) (prob-session solved) then (add-form
            (M109)))
      (if (state unsolved) (attempts pretrigger) (help-level notpartfull)
            (rating higher) (prob-session attempted) then (add-form
            (M110)))
      (if (state unsolved) (attempts triggerpartial) (help-level
            notpartfull) then (add-form (M112)) (add-form (M113)))
      (if (state unsolved) (attempts triggerfull) (help-level
            notpartfull) then (add-form (M115)) (add-form (M116)) (add-
            form (M117)))
))

(eop ((if (action visited) (visited trigger) (state unsolved) (rating
            not-one) then (add-form (M200)) (add-form (M206)) (add-form
            (Q2)))
       (if (action visited) (visited trigger) (state unsolved) (rating
            one) then (add-form (M200)) (add-form (M205)) (add-form
            (Q2)))
       (if (action attempted) (state solved) then (add-form (M201)))
       (if (action attempted) (state solved) (help-level OR three five)
            (attempts OR one two) then (add-form (M203)) (add-form
            (M102)) (add-form (M208)) (add-form (Q10)))
       (if (action attempted) (state solved) (help-level five) (attempts
            pretrigger) then (add-form (M202))  (add-form (Q11)))
       (if (action attempted) (state solved) (help-level notpartfull)
            (attempts pretrigger) (last-se OR low middle) then (add-form
            (M218)) (add-form (Q14)))
       (if (action attempted) (state solved) (help-level notpartfull)
            (attempts pretrigger) (last-se high) then (add-form (M218))
            (add-form (Q15)))
       (if (action attempted) (state solved) (help-level notpartfull)
            (attempts grtpretrigger) (last-se OR low middle) then (add-
            form (M209)) (add-form (M210)) (add-form (Q12)))
       (if (action attempted) (state solved) (help-level notpartfull)
            (attempts grtpretrigger) (last-se high) then (add-form
            (M209)) (add-form (Q13)))
       (if (action attempted) (state unsolved) (attempts one) (rating
            one) then (add-form (M204)) (add-form (M205)) (add-form
            (M211)) (add-form (Q16)))
       (if (action attempted) (state unsolved) (attempts one) (rating
            higher) then (add-form (M204)) (add-form (M205)) (add-form
            (M207)) (add-form (Q16)))
       (if (action attempted) (state unsolved) (help-level three) (last-
            se low) then (add-form (M212)) (add-form (M213)) (add-form
            (M208)) (add-form (Q17)))
       (if (action attempted) (state unsolved) (help-level three) (last-
            se middle) then (add-form (M212)) (add-form (M212)) (add-
            form (Q17)))
       (if (action attempted) (state unsolved) (help-level three) (last-
            se high) then (add-form (M212)) (add-form (Q17)))
       (if (action attempted) (state unsolved) (help-level five) then
            (add-form (M215)) (add-form (Q17)))
       (if (action attempted) (state unsolved) (help-level notpartfull)
            (attempts grtpretrigger) (last-se low) then (add-form
```

```
        (M214)) (add-form (M213)) (add-form (M216)) (add-form
        (Q18)))
    (if (action attempted) (state unsolved) (help-level notpartfull)
        (attempts grtpretrigger) (last-se middle) then (add-form
        (M214)) (add-form (M216)) (add-form (Q18)))
    (if (action attempted) (state unsolved) (help-level notpartfull)
        (attempts grtpretrigger) (last-se high) then (add-form
        (M214)) (add-form (M216)) (add-form (Q18)))
    (if (action attempted) (state unsolved) (help-level notpartfull)
        (attempts lesspretrigger) (last-se low) then (add-form
        (M217)) (add-form (M216)) (add-form (M208)) (add-form
        (Q17)))
    (if (action attempted) (state unsolved) (help-level notpartfull)
        (attempts lesspretrigger) (last-se middle) then (add-form
        (M217)) (add-form (M216)) (add-form (Q17)))
    (if (action attempted) (state unsolved) (help-level notpartfull)
        (attempts lesspretrigger) (last-se high) then (add-form
        (M217)) (add-form (M216)) (add-form (Q17)))
))

(se ((if (perform high) (se-accuracy pessimistic) then (add-form (M308))
        (add-form (M314)) (add-form (Q19)))
    (if (perform high) (se-accuracy realistic) then (add-form (M309))
        (add-form (Q20)))
    (if (perform high) (se-accuracy optimistic) then (add-form (M310))
        (add-form (Q21)))
    (if (perform middle) (se-accuracy pessimistic) then (add-form
        (M312)) (add-form (Q22)))
    (if (perform middle) (se-accuracy realistic) then (add-form (M313))
        (add-form (Q20)))
    (if (perform middle) (se-accuracy optimistic) then (add-form
        (M313)) (add-form (M318)) (add-form (Q21)))
    (if (perform low) (se-accuracy pessimistic) (effort good) then
        (add-form (M316)) (add-form (M317)) (add-form (M315)) (add-
        form (Q23)))
    (if (perform low) (se-accuracy pessimistic) (effort poor) then
        (add-form (M315)) (add-form (M317)) (add-form (Q23)))
    (if (perform low) (se-accuracy realistic) (effort good) then (add-
        form (M316)) (add-form (M315)) (add-form (M317)) (add-form
        (Q23)))
    (if (perform low) (se-accuracy realistic) (effort poor) then (add-
        form (M319)) (add-form (M315)) (add-form (M317)) (add-form
        (Q23)))
    (if (perform low) (se-accuracy optimistic) (effort good) then (add-
        form (M316)) (add-form (M317)) (add-form (Q23)))
    (if (perform low) (se-accuracy optimistic) (effort poor) then (add-
        form (M319)) (add-form (M317)) (add-form (Q23)))
    (if (ss-se low) (eop-avg same) then (add-form (M300)) (add-form
        (M306)) (add-form (M303)))
    (if (ss-se low) (eop-avg increase) then (add-form (M300)) (add-form
        (M307)) (add-form (M304)))
    (if (ss-se middle) (eop-avg same) then (add-form (M301)) (add-form
        (M306)) (add-form (M303)))
    (if (ss-se middle) (eop-avg increase) then (add-form (M301)) (add-
        form (M307)) (add-form (M304)))
    (if (ss-se middle) (eop-avg decrease) then (add-form (M301)) (add-
        form (M306)) (add-form (M305)))
    (if (ss-se high) (eop-avg same) then (add-form (M302)) (add-form
        (M306)) (add-form (M303)))
    (if (ss-se high) (eop-avg decrease) then (delete-form (M310)) (add-
        form (M302)) (add-form (M306)) (add-form (M305)))
```

```
    (if (logon one) (perform low) then (delete-form (M308)) (delete-
          form (M309)) (delete-form (M310)) (delete-form (M311))
          (delete-form (M312)) (delete-form (M313)) (delete-form
          (M314)) (delete-form (M315)) (delete-form (M316)) (delete-
          form (M317)) (delete-form (M318)) (delete-form (M319)) (add-
          form (M320)) (add-form (M321)) (add-form (M315)) (add-form
          (Q23)))
    (if (logon one) (perform middle) then (delete-form (M314)) (delete-
          form (M315)) (delete-form (M316)) (delete-form (M317))
          (delete-form (M318)) (delete-form (M319))      (add-form
          (M320)) (add-form (M322)) (add-form (Q23)))
    (if (logon one) (perform high) then (delete-form (M308)) (delete-
          form (M309)) (delete-form (M310)) (delete-form (M311))
          (delete-form (M312)) (delete-form (M313)) (delete-form
          (M314)) (delete-form (M315)) (delete-form (M316)) (delete-
          form (M317)) (delete-form (M318)) (delete-form (M319)) (add-
          form (M320)) (add-form (M323)) (add-form (Q23)))
))
```

Appendix D Feedback Statements

These are the feedback statements that were stored in 'feedback.dat' file and automatically loaded at the start of SQL-Tutor.

```
(Q0 "How confident are you about answering some SQL SELECT problems in
     this session? Please Indicate below." "Reflecting on the problem
     you just left and this session so far, how confident are you about
     answering further SQL SELECT problems in this session? Please
     Indicate below." "Reflecting on this session, how confident are you
     about answering SQL SELECT problems? Please Indicate below.")
(Q1 "Given what you have covered in lectures and workshops so far, how
     confident are you about creating SQL SELECT queries? Please indicate
     below.")
(Q2 "Is this a reflection of how confident you feel about being able to
     attempt the problems? What would you rate your current confidence
     level to be? Please indicate below.")
(Q3 "Have you been able to do any work on SQL since your session to
     increase your confidence? What would you rate your current
     confidence level to be? Please indicate below.")
(Q4 "Do you still feel the same? Please indicate below.")
(Q5 "On reflection seeing that you did well in your last session, has
     that increased your confidence in answering SQL queries? Please
     indicate below.")
(Q6 "Well done, your performance matched your confidence in the last
     session. How confident do you feel about answering further SQL
     queries in this session? Please indicate below. ")
(Q7 "On reflection seeing that you did quite well in your last session,
     has that increased your confidence in answering SQL queries? Please
     indicate below.")
(Q8 "On reflection seeing that you didn't do quite so well in your last
     session as you expected to, has that changed your confidence in
     answering SQL queries? Please indicate below.")
(Q9 "On reflection seeing that you didn't do quite so well in your last
     session, has that changed your confidence in answering SQL queries?
     Please indicate below.")
(Q10 "On reflection seeing that you solved the problem, but viewed the
     partial or full solution, has that changed your confidence in
     answering SQL queries? Please indicate below.")
(Q11 "On reflection seeing that you solved the problem, but viewed the
     full solution, has that changed your confidence in answering SQL
     queries? Please indicate below.")
(Q12 "By reflecting on what you have achieved by solving the problem
     through putting in the effort, has this increased your confidence
     in answering SQL queries at all? Please indicate below.")
(Q13 "By reflecting on what you have achieved by solving the problem
     through putting in the effort, has this helped to maintain your
     confidence in answering SQL queries? Please indicate below.")
(Q14 "By reflecting on what you have achieved by solving the problem,
     has this increased your confidence in answering SQL queries at all?
     Please indicate below.")
(Q15 "By reflecting on what you have achieved by solving the problem
     through putting in the effort, has this helped to maintain your
     confidence in answering SQL queries? Please indicate below.")
(Q16 "What is your current confidence level for really attempting some
     more SQL problems? Please indicate below.")
```

(Q17 "What is your current confidence level for attempting some more SQL problems? Please indicate below.")
(Q18 "You're putting the effort in for working on these problems, so how would you rate your current confidence level for attempting more SQL problems?")
(Q19 "Your performance was much higher than your confidence this session. How confident do you feel about answering SQL queries? Please indicate below. ")
(Q20 "Your performance matched your confidence this session. How confident do you feel about answering SQL queries? Please indicate below. ")
(Q21 "Your performance was in-line with your confidence this session. How confident do you feel about answering SQL queries? Please indicate below. ")
(Q22 "Your performance was higher than your confidence this session. How confident do you feel about answering SQL queries? Please indicate below. ")
(Q23 "Reflecting on your performance and effort in this session, what is your current confidence level about answering SQL queries? Please Indicate below.")
(M0 "Welcome back.")
(M1 "Welcome to SQL-Tutor. Take some time to become familiar with the features it offers and the types of problems it contains.")
(M2 "Welcome back for your second session.")
(M3 "You appeared to use your last session purely to become familiar with SQL-Tutor and look at some of the problems. Hopefully you found that helpful.")
(M4 "You ended your last session not that confident about being able to answer SQL SELECT problems. ")
(M5 "You ended your last session quite confident about being able to answer SQL SELECT problems.")
(M6 "SQL-Tutor has not recorded a confidence level from you for the end of the last session. Remember to only use SQL-Tutor's buttons and not the browser buttons for navigation, otherwise SQL-Tutor could give incorrect feedback.")
(M7 "You ended your last session confident about being able to answer SQL SELECT problems.")
(M8 "You had a good session last time with a good set of problems successfully completed.")
(M9 "You had a fairly good session last time with some problems successfully completed.")
(M10 "You attempted some problems last time, let's see if you can complete some more.")
(M11 "Welcome back.")
(M12 "You did not attempt any problems last time, but it was quite a short session.")
(M13 "You did not attempt any problems last time.")
(M14 "You didn't have a very productive session last time, let's see if you can progress further now.")
(M15 "Although you didn't make a lot of progress last time, that doesn't mean this session will be the same.")
(M16 "SQL SELECT problems will become easier the more you practice.")
(M17 "However you didn’t make a lot of progress last time, let's see you change that and make good progress in this session.")
(M100 "That was the first time you have visited this problem, but you chose to directly view the partial/full solution.")
(M101 "The text entered for any of the clauses wasn't changed from the previous submission.")
(M102 "Try to keep this as a last resort in order to maximise the benefits of practising the problem solution yourself.")

(M103 "You have already opened this problem without solving it before. If you are still finding it difficult…")
(M104 "Very well done.")
(M105 "Although you have opened this problem before, this time you have chosen to directly view the partial/full solution.")
(M106 "No text was entered for any of the clauses.")
(M107 "The errors that are displayed may seem daunting, but try to focus on just the first error. Sometimes resolving the first error will actually solve them all, as they can be interlinked.")
(M108 "You're putting the effort in on this problem, keep at it. Not all problems can be solved quickly.")
(M109 "Bear in mind that this problem has a higher complexity rating than the last problem you solved.")
(M110 "Bear in mind that this problem has a higher complexity rating than the other problem(s) you have attempted in this session.")
(M111 "Try not to look at the partial or full solution after just a few attempts. If you are stuck remember the different sources of help to try first or leave this problem for now and select one that you think you will be able to solve.")
(M112 "You're really putting the effort in on this problem, keep it up.")
(M113 "How about selecting the help level to look at the 'partial' solution to see if that can give you any insight to solve the problem by yourself?")
(M114 "Try one of the other sources of help, then if the problem isn't solved maybe consider selecting the help level to look at the 'partial' solution. It may just be enough for you to solve the problem by yourself.")
(M115 "You're really working hard on this problem, you're making a really good effort.")
(M116 "How about selecting the help level to look at the 'full' solution?")
(M117 "Don't just copy & submit as the answer, but really take a long look at it to make sure you understand how it works and solves the problem.")
(M118 "You're working hard on this problem.")
(M119 "Try one of the other sources of help, then if the problem isn't solved maybe consider selecting the help level to look at the 'full' solution.")
(M200 "You're tending to open problems, but then leave them without attempting them in this session.")
(M201 "Well done, you solved the problem.")
(M202 "Although you selected the help level to view the full solution, you really tried to solve the problem yourself.")
(M203 "You selected the help level to view the partial or full solution very quickly.")
(M204 "You only submitted one try at the answer. ")
(M205 "Remember there are various sources of help (i.e. click on the [i] button if the question is ambiguous, the [SQL Help] will take you to the w3schools' SQL web pages, the different help levels). ")
(M206 "Consider trying a problem with a complexity level at one lower than this problem.")
(M207 "Bear in mind this problem had a higher complexity level than the last problem you solved.")
(M208 "It may seem like a cliché, but it's true… The more that you practise the more you'll improve.")
(M209 "Good effort - although it was hard you put the effort in and it paid off.")
(M210 "You didn't even need to select a help level to view the partial or full solution. Well done.")

(M211 "Have a go at another problem with the same complexity rating, then come back to this problem to give it another go.")
(M212 "You've looked at the partial solution, now take what you may have learnt from that and try a few other problems, before coming back to this one to give it another go. ")
(M213 "You didn't go for the easy option of viewing the full solution - well done for that. ")
(M214 "Although you haven't solved the problem yet, you have put the effort in.")
(M215 "You looked at full solution, but didn't submit it - well done for that. Instead it's a good opportunity; you can use what you have learnt from looking at the full solution, apply this when trying other problems and then come back to try this problem at another point.")
(M216 "Give yourself a break from the problem and try it again at another point.")
(M217 "You may not have solved the problem yet, but keep at it.")
(M218 "All of your work to-date with SQL SELECT queries it paying off.")
(M300 "Your confidence at the beginning of the session was quite low")
(M301 "Your confidence at the beginning of the session was good")
(M302 "Your confidence at the beginning of the session was high")
(M303 "on average, it has not changed throughout this session.")
(M304 "on average, it has increased throughout this session.")
(M305 "on average, it has decreased in this session.")
(M306 "and unfortunately,")
(M307 "however,")
(M308 "Even though you have not found the problems easy, you really have made good progress in successfully completing problems. ")
(M309 "You have made really good progress this session and appear to have good judgement on your ability with your confidence levels.")
(M310 "You have made really good progress this session, but have the problems challenged you enough? Have a think whether you are ready to try more difficult problems next time.")
(M311 "You have made really good progress this session, just look at the problems that you have successfully completed.")
(M312 "Even though you do not think so now, you have made good progress.")
(M313 "You have made steady progress this session.")
(M314 "Think about starting the next session working on problems with the same complexity rating has you successfully completed today in order to help to build your confidence.")
(M315 "Think about focussing on problems with complexity rating of 1 or 2 at the beginning of the next session to firmly establish your core knowledge before expanding on it.")
(M316 "Although you may not have completed as many problems as you would have liked, you are putting in the effort.")
(M317 "Keep practicing, re-read your lecture notes and do not forget about the [SQL Help] button along the top of the main problem page.")
(M318 "Think about starting the next session working on problems with the same complexity rating as you successfully completed today in order to re-enforce your understanding.")
(M319 "You have not made much progress this session, but that doesn't mean the next session has to be the same.")
(M320 "This was your first session")
(M321 ", which you have largely used to familiarise yourself with SQL-Tutor. This should prove useful in your next session.")
(M322 "and you have made steady progress already. Well done.")
(M323 "and you have already progressed well. Very well done.")

Appendix E Summative Assessments

This appendix shows the full list of questions that formed the summative assessments for the database module that the study participants were taking. Not all of the questions were relevant when used alongside the scope of SQL-Tutor, so results for the excluded questions not used in the analysis are also indicated by a red cross ✖ .

E.1 Milestone 1

1. Multiple Choice: Which of the following statements IS ... Points: 1

Question	Which of the following statements IS true in relation to a Cartesian product? (Select only 1)
Answer	A. A cartesian product occurs when a join is defined in the WHERE clause of SQL statement. B. ✔ A Cartisian product occurs when the DBMS has not been given a specified join clause for all the tables/relations that are used in the retrival C. Cartesian products are useful if the relationship between the tables/relations is not clearly defined. D. A Cartesian product is the prefered join method for optimising retrieval of data using SQL
Correct Feedback	correct
Incorrect Feedback	Cartesian products are never useful, they are triggered when the relationship between entities/relations are not defined or declared and the DBMS is unsure of how to relate the tables together, the cartisian product returns every possible combination of records between the relations, a cartesian product between 2 tables with 10 records in each will result in 100 rows being returned.

2. Multiple Choice: Which of the following statements is ... Points: 1

Question	Which of the following statements is NOT correct about natural joins? (select only 1)
Answer	A. ✔ The natural join will join on a shared column name regardless of the data type associated with the attribute B. A natural join will attempt to enforce a relationship where the DBMS can see a shared attribute name and type between 2 tables/relations C. Joining data on a natural join will not necessarily return the correct data set of results D. A natural join is a type of inner join and will only return results to the result dataset where a match is made
Correct Feedback	correct
Incorrect Feedback	A natural join is made where the DBMS can see a shared attribute, the shared attribute must have the same name AND data type in order to be used, however, where the identified match is incorrect and the shared elements are coincidence rather than related then the result of the join will be incorrect. Therefore, not only must the name and data type match, the MEANING must also be consistant in order for the join to be effective.

3. True / False: Please indicate if the following stat... Points: 1 ✖

Question	Please indicate if the following statement is True or False. *"Structured Query Language (SQL) is the tool used to implement the relational model and using SQL we are able to fully implement the relational model in the real world"*
Answer	True ✔ False
Correct Feedback	correct
Incorrect Feedback	The relational model is not able to be fully implemented in the real world, it is a theoretical model that can be partially realised using SQL but not fully.

4. Fill in Multiple Blanks: The following question is based on on... Points: 5.5

Question: The following question is based on on the ERD and tables that you have been using in the seminar tasks since week 2, a copy of the ERD and table descriptions was given out by your seminar tutor at the start of this assessment.

Complete the following code, when completed the SQL should generate a list of when and where Tim Rose teaches.

```
SQL> select day,classtime, [ATTRIBUTE1]
  2      from [TABLE1],[TABLE2]
  3      where [ATTRIBUTE2] = 'Tim Rose'
  4      and [ATTRIBUTE3] = [ATTRIBUTE4]
DAY          CLASSTIME  ROOM
------------ ---------- ------------
Tuesday      11am           Elec Eng G25
Friday       11am           Elec Eng G25
Friday       11am           Elec Eng G25
Tuesday      11am           Elec Eng G25
```

Answers for: ATTRIBUTE1	ROOM
Answers for: TABLE1	CLASS STAFFMEMBER
Answers for: TABLE2	STAFFMEMBER CLASS
Answers for: ATTRIBUTE2	NAME
Answers for: ATTRIBUTE3	TEACHER STAFFID CLASS.TEACHER STAFFMEMBER.STAFFID
Answers for: ATTRIBUTE4	STAFFID TEACHER STAFFMEMBER.STAFFID CLASS.TEACHER

Correct Feedback Correct

Incorrect Feedback The correct answer code would be:

select Day, classtime, room
from class, staffmember
where name = 'Tim Rose'
and teacher = staffid;

You will be give credit for the elements that you correctly identified.

---

5. Multiple Choice: Which of the following statements is ... Points: 1

Question Which of the following statements is true about extracting data from the database using SQL code? (Select only 1)

Answer

A.

✔ The only 2 clauses required by SQL in order for a simple statement/query to run is the SELECT and FROM clauses, these are required in order for the DBMS to know what attributes are being retrieved and from which tables

B.

The only 3 clauses required by SQL in order for a simple statement/query to run is the SELECT, FROM and WHERE clauses, these are required in order for the DBMS to know what attributes are being retrieved, where the data is held in the database system and how the data should be filtered, the statement won't run without a WHERE clause because the DBMS is unsure of which filters are required to format the data.

C.

In order for the DBMS to proceed and action an SQL statement all joins MUST be specified in the SQL query, any statements that do not define correctly the relationships between the relations/tables will not be carried out by the DBMS as it is uncertain how to match the tables.

D.

SQL is an intuative language, it is dynamic enough to understand which columns/attributes are derived from which tables/relations, where relations/tables share attibutes with the same name the DBMS will assume the required one is from the first table in the FROM clause.

Correct Feedback correct

Incorrect Feedback SQL is a very specific language, the only clauses that are required in order for a statement to run are the SELECT and FROM clause, the WHERE clause provide options for further filters but it is not a compulsory clause. SQL is not able to handle shared attribute names, where a column name exists in multiple tables within a statement the SQL code must specifiy which table the attribute is derived from, failure to do this will result in the error 'ambigious column definition' being returned and the query failing to run.

---

6. True / False: "The usefulness of any database ... Points: 1

Question *"The usefulness of any database is in converting the data contained into information that may be applied for a 3rd party use. Database systems which contain incorrect data are therefore useless as the information generated for unqualified data risks being intrinsically flawed."*

Is the above statement true or false?

Answer
✔ True
False

Correct Feedback correct

Incorrect Feedback A database is only as good as the data it contains, where the data is flawed the database is also flawed. Databases do not hold information they hold data, the extraction process converts the data to information and this information is then utilised by a 3rd party.

E.2 Milestone 2

1. Fill in Multiple Blanks: The university executive requires a r... Points: 4.5

Question The university executive requires a report of the number of modules that are not cost effective, they have therefore asked for a list of ALL the module names that have less than 15 students enrolled on them.

Complete the code below to generate a query that will display the outcome required above (1/2 a mark for each correct element)

```
Select [ATTRIBUTE1]
from subject [TYPE] join [RELATION2] using[ATRIBUTE2]
[CLAUSE] by [ATTRIBUTE3]
[CLAUSE2]  [FUNCTION2] ([ATTRIBUTE4]) < 15;
```

Answers for: ATTRIBUTE1	sname subject.sname
Answers for: TYPE	left
Answers for: RELATION2	enrolled
Answers for: ATRIBUTE2	subjectid
Answers for: CLAUSE	group
Answers for: ATTRIBUTE3	studentid enrolled.studentid
Answers for: CLAUSE2	Having
Answers for: FUNCTION2	count
Answers for: ATTRIBUTE4	studentid enrolled.studentid
Correct Feedback	correct
Incorrect Feedback	select sname from subject left join enrolled using (subjectid) group by sname having count(studentid) <15;

---

2. Jumbled Sentence: Complete the following sentence whi... Points: 2

Question Complete the following sentence which describes various functions and their actions

[FUNCTION1] amendes the value retrieved to the nearest whole number

[FUNCTION2] removes all the numeric data after a decimal point thus returning a whole number only

[FUNCTION3] determines the number of the selected element returned

Drop-down List of Answers
ROUND
TRIM
AVG
CHOP
TRUNK
LTRIM
RTRIM
TRUNC
FREQ
COUNT

Answer Complete the following sentence which describes various functions and their actions

✔ **ROUND** amendes the value retrieved to the nearest whole number

✔ **TRUNC** removes all the numeric data after a decimal point thus returning a whole number only

✔ **COUNT** determines the number of the selected element returned

Correct Feedback Correct

Incorrect Feedback
round amendes the value retrieved to the nearest whole number
trunc removes all the numeric data after a decimal point thus returning a whole number only
count determin the number of the selected element returned

3. Multiple Choice: SELECT SNAME, COUNT(STUDENTID)FROM EN... Points: 1

Question

```
SELECT SNAME, COUNT(STUDENTID)
FROM ENROLLED RIGHT JOIN SUBJECT USING (SUBJECTID)
WHERE STUDENTID IS NULL
GROUP BY SNAME;
```

Which of the following statements best describes the code above:

Answer

A.

✓ The query will display the subject name for all the subjects that have no students enrolled on them

B.

The query will select all the subject details for subjects that have students enrolled on them

C.

The query will not run because the join clause is not specified correctly

D.

The query will retrieve the ernollment details for all the subjects that have no students attached to them

Correct Feedback correct

Incorrect Feedback The query retrieves only the subject name for all the subjects that have noone enrolled on them, the outer join would leave the studentid blank if no student is found to match in the enrollment table.

4. True / False: Is the following statement TRUE or FA... Points: 1

Question Is the following statement TRUE or FALSE regarding OUTER JOINS?

"The outer join is a join type that is similar to the Natural join in that they should be avoided in normal situations as they can lead to cartisian joins and return excessive amounts of data that is irrelevant to the query"

Answer

True

✓ False

Correct Feedback Correct

Incorrect Feedback OUTER JOINS can be used to identify data where there is no corresponding match and they are often used by the DBA to try and identify orphan records which may have become orphaned due to data import from legacy systems. They do not retieve any additional data in the way a Natural join that can't find a match will.

5. Fill in Multiple Blanks: Complete the code below by filling ... Points: 4.5

Question Complete the code below by filling in the blanks to generate a piece of SQL code that will retrieve the names of the students who have failed more than one module

```
select [ATTRIBUTE1], [AGGREGATION1](*)
from marks [CLAUSE1] [TABLE1] [CLAUSE2] (studentid)
[CLAUSE3] [ATTRIBUTE2] < 40
[CLAUSE4] by stuname
[CLAUSE5]count(*) > 1;
```

Answers for: ATTRIBUTE1	STUNAME STUDENT.STUNAME
Answers for: AGGREGATION1	COUNT
Answers for: CLAUSE1	join
Answers for: TABLE1	STUDENT
Answers for: CLAUSE2	using
Answers for: CLAUSE3	WHERE
Answers for: ATTRIBUTE2	MARK MARKS.MARK
Answers for: CLAUSE4	GROUP
Answers for: CLAUSE5	HAVING
Correct Feedback	correct

Incorrect Feedback the correct code is

```
SQL>  select stuname, count(*)
  2  from marks join student using (studentid)
  3  where mark < 40
  4  group by stuname
  5  having count(*) > 1;
STUNAME              COUNT(*)
-------------------- ----------
Lesley Johnson                2
```

6. Multiple Choice: Which of the following statements IS ... Points: 1

Question Which of the following statements IS true in relation to a Cartesian product? (Select only 1)

Answer

A.

A Cartesian product occurs when a join is defined in the WHERE clause of SQL statement.

B.

✓ A Cartesian product occurs when the DBMS has not been given a specified join clause for all the tables/relations that are used in the retrieval

C.

Cartesian products are useful if the relationship between the tables/relations is not clearly defined.

D.

A Cartesian product is the prefered join method for optimising retrieval of data using SQL

Correct Feedback correct

Incorrect Feedback Cartesian products are never useful, they are triggered when the relationship between entities/relations are not defined or declared and the DBMS is unsure of how to relate the tables together, the cartisian product returns every possible combination of records between the relations, a cartesian product between 2 tables with 10 records in each will result in 100 rows being returned.

7. Multiple Choice: Which of the following join types is ... Points: 1

Question Which of the following join types is NOT valid (select only 1)

Answer

A.

inner Join

B.

full outer Join

C.

Equi join

D.

✓ Where join

Correct Feedback correct

Incorrect Feedback Full outer, Equi and Inner are all valid, however, there is no such join as a Where join

8. Multiple Choice: Which of the following statements is ... Points: 1

Question	Which of the following statements is TRUE (Select one answer)
Answer	A. Grouped data may be used in the WHERE clause B. When data is retrieved from the DBMS the data is joined then the filters are applied and the attributes that are not selected are deleted from the result data set before the group by clause is applied, this means that the grouping of data is the last action done by the DBMS before displaying the result set to the system that called the SQL code. C. ✔ Aggregated data may be grouped by any attribute in any table that is referenced in the FROM clause in the SQL statement, there is no need to ensure the attribute is retrieved in the SELECT statement. D. It is not possible to filter out the results of the query using aggregated data, data may be filtered only in the WHERE clause and grouping is referenced outside of this clause.
Correct Feedback	correct
Incorrect Feedback	It is possible to aggregate the data and group data based on attributes that have not been selected, in the seminar we retrieved the maximum average mark from the marks table and the data was grouped on the MAJOR column which was NOT in the select statement (select max(avg(mark)) from marks natural join student group by major). The HAVING clause will allow the data to be filtered on the grouped data, this clause works in the same way as the WHERE clause but rather than acting at a row level it acts on the grouped data only. Data is retrieved and joined before the DMBS applies the filters and any grouping clauses in the query, the last thing that is done by the DBMS is to strip off the attributes that are not selected within the query.

9. True / False: Is the following statement true or fa... Points: 1

Question	Is the following statement true or false? *"a database holds raw data, the order of the data is not important and has no impact on the interpretation of the data, data is then extracted and aggregation is performed in order to convert the raw, unorganised data into stats or data that can be useful to an organisation"*
Answer	✔ True False
Correct Feedback	Correct
Incorrect Feedback	A database only holds raw data, organisations require data to be converted to information, this information is often required to be in a format that will highlight the key statistics, aggregation will allow this formatting of data.

10. Multiple Choice: Which of the following statements des... Points: 1

Question: Which of the following statements describes what occurs in the code below (Select one answer):

```
select major, max(avg(mark))
from student join marks using (studentid)
group by major;
```

Answer:

A. ✔ The code will not run because you cannot max an average mark a sub query would be needed

B. The query will display the major which has the highest average mark

C. The query will display all the majors and their average marks

D. The query will display the major and number of students studying those majors

Correct Feedback: Correct

Incorrect Feedback: the code will not run! the logic of selecting the max of an average with a grouped data is flawed, a sub query would need to be used, the subquery would determin the max average then this would be filtered on.

```
SQL> select major, max(avg(mark))
  2  from student join marks using (studentid)
  3  group by major;
select major, max(avg(mark))
       *
ERROR at line 1:
ORA-00937: not a single-group group function
correct code is:
SQL> select major
  2  from student join marks using (studentid)
  3  group by major
  4  having avg(mark) = (select max(avg(mark)) from student join marks using (studentid) group by major);
MAJOR
----------
Comp Eng
```

11. Jumbled Sentence: There are a number of pre-defined f... Points: 2

Question	There are a number of pre-defined functions within the Oracle DBMS. Using the drop down lists match the function to the description (1/2 mark each) [FUNCTION1] Determines the location of a substring in a string [FUNCTION2] Retrieves part of a string expression [FUNCTION3] Determines how many characters are in a string
Drop-down List of Answers	COUNT SUBSTRING STRPOS INSTR INSTRING LENGTH TRIM LTRIM RTRIM SUM
Answer	There are a number of pre-defined functions within the Oracle DBMS. Using the drop down lists match the function to the description (1/2 mark each) ✔ **INSTR** Determines the location of a substring in a string ✔ **SUBSTRING** Retrieves part of a string expression ✔ **LENGTH** Determines how many characters are in a string
Correct Feedback	Correct
Incorrect Feedback	INSTR Determines the location of a substring in a string SUBSTRING Retrieves part of a string expression LENGTH Determines how many characters are in a string

12. Fill in Multiple Blanks: There are 20 students in each subject... Points: 5

Question: There are 20 students in each subject, the university is concerned by subjects where a high number of the students are passing. They want to identify the subjects that have more than 15 students passing.

Complete the code to generate a query that will determine the data required and ensure the results match the display below:

```
Subject Name                    Number passing
---------------------------------------------------
```

```
Concurrent Computing              17
Digital Systems Structures        24
Database System Implementation    22
Computing 1B                      34
Computing 1A                      22
Programming 1                     16
Operating Systems                 18
```

complete the code below:

```
select [ATTRIBUTE1]  as "[VALUE1]", count(studentid) [SYNTAX] "[VALUE2]"
 from subject left join [TABLE1] [SYNTAX2] (subjectid)
 where mark > [VALUE3]
 [CLAUSE1] by sname
 [CLAUSE2] count(studentid) > [VALUE4];
```

Answers for: ATTRIBUTE1	sname subjecti.sname
Answers for: VALUE1	Subject Name
Answers for: SYNTAX	as
Answers for: VALUE2	number passing
Answers for: TABLE1	marks
Answers for: SYNTAX2	using
Answers for: VALUE3	40
Answers for: CLAUSE1	group
Answers for: CLAUSE2	having
Answers for: VALUE4	15
Correct Feedback	correct
Incorrect Feedback	(see below)

```
SQL>  select sname as "Subject Name", count(studentid) as "Number passing"
  2   from subject left join marks using (subjectid)
  3   where mark > 40
  4   group by sname
  5   having count(studentid) > 15;
Subject Name                    Number passing
------------------------------ --------------
Concurrent Computing               17
Digital Systems Structures         24
Database System Implementation          22
Computing 1B                       34
Computing 1A                       22
Programming 1                      16
Operating Systems                  18
7 rows selected.
```

13. True / False: True or False: Is the following statement true or ... Points: 1

Question

Is the following statement true or false?

"When writing a query that uses an outer join between 2 tables, employing the COUNT function to determine the number of elements in the optional table, it is necessary to ensure that the COUNT function only counts elements from the optional table in order to ensure a true count is presented in the results"

Answer

- ✔ True
- False

Correct Feedback

correct

Incorrect Feedback

When using the outer join and counting the elements in the 'pile' the count must act on the outer element or it will not return an accurate result, for example:

```
select sname, count(*)
from subject left join enrolled using (subjectid)
group by sname
```

this code will return a minimum count of 1 even if there are no students enrolled on the module, this is because the data is placed into piles of sname and each pile has at least one record in the pile regardless.

```
select sname, count(studentid)
from subject left join enrolled using (subjectid)
group by sname
```

this will return a count of 0 or null if there are no students enrolled as each pile of sname is examined for studentid's not any data.

E.3 Milestone 3

1. True / False: Indicate whether the following statem... Points: 1

Question

Indicate whether the following statement is true or false

"A materialised view is a non-permanent store of data that can be used by other users, it is refreshed when the view is accessed and is dynamic in nature"

Answer

- True
- ✔ False

Correct Feedback

correct

Incorrect Feedback

materialised views are a permanent additional stores of data that allow users to access data, logical views are refresehed each time the views are accessed.

2. Fill in Multiple Blanks: Complete the following piece of code ... Points: 4.5

Question

Complete the following piece of code that will create a view called 'FAILING_SUBJECT' the view will identify the subjects that have less than 20 students enrolled on them and display the subject name and the number of students currently enrolled. The data will display the results under the following heading: sname, number_enrolled :

```
create [SYNTAX1] failing_subject as
  select sname, count(*) [COMMAND2]
   from [TABLE]  join subject [COMMAND3] (subjectid)
   [COMMAND4] by [ATTRIBUTE1]
  [COMMAND5] [COMMAND](*) < [VALUE];
```

Answers for: SYNTAX1	VIEW
Answers for: COMMAND2	NUMBER_ENROLLED
Answers for: TABLE	ENROLLED
Answers for: COMMAND3	USING
Answers for: COMMAND4	GROUP
Answers for: ATTRIBUTE1	SNAME SUBJECT.SNAME
Answers for: COMMAND5	HAVING
Answers for: COMMAND	COUNT
Answers for: VALUE	20
Correct Feedback	CORRECT
Incorrect Feedback	(see below)

```
create view failing_subject as
  select sname, count(*) number_enrolled
from enrolled join subject using (subjectid)
 group by sname
  having count(*) < 20;
```

3. Multiple Choice: Which of the following views is a sim... Points: 1

Question	Which of the following views is a simple view (select one answer)
Answer	A. Create my_simple_view as select * ✔ from students where major = 'ITMB' and age > 20; B. Create my_simple_view as select major, avg(age) from students group by major; C. Create my_simple_view as select stuname from marks natural join student where mark < 40 and subjectid = 'COMP1001'; D. none of the statements are simple views.
Correct Feedback	correct
Incorrect Feedback	if a view has aggregation or pulls from multiple tables then it is not a simple view but a complex one.

4. Multiple Choice: Which of the following statements is ... Points: 1

✖

Question	Which of the following statements is NOT true about the storage parameters used during the CREATE TABLE command:
Answer	A. ✔ Storage parameters must be declared for all tables created or the Table will not be created by the DBMS B. PCTINCREASE defines the percentage growth for subsequent extents when the DBMS instigates a new extent generation. C. INITIAL is the size of the first extent that is created when the table is created D. NEXT is attribute that defines the size of subsequent extent creation for a specified table.
Correct Feedback	correct
Incorrect Feedback	storage parameters are not compulsory, where they are not defined the DBMS will take the values from the default tablespace.

5. Fill in Multiple Blanks: The table Pet_overview is defined as:... Points: 4.5

✖

Question	The table Pet_overview is defined as: <u>Pet_ID number(8)</u> Pet_NAME varchar2(35) Pet_Type varchar2(15) must be CAT, DOG, MARINE, EQUESTRIAN or OTHER Pet_DoB date Pet_registration date Complete the create table command below that will create the table with the constraints as described in the description above. Create table [VARIABLE1] ([ATTRIBUTE1] number(8) [VARIABLE6] key, Pet_NAME varchar2(35), Pet_Type varchar2(15) [VARIABLE2] PET_TYPE_CK [VARIABLE3] ([VARIABLE4] in ('CAT', 'DOG', '[VARIABLE5]', 'EQUESTRIAN','OTHER'), Pet_DoB [DATATYPE1], [ATTRIBUTE2] date);
Answers for: VARIABLE1	PET_OVERVIEW
Answers for: ATTRIBUTE1	PET_ID
Answers for: VARIABLE6	PRIMARY
Answers for: VARIABLE2	CONSTRAINT
Answers for: VARIABLE3	CHECK
Answers for: VARIABLE4	PET_TYPE
Answers for: VARIABLE5	MARINE
Answers for: DATATYPE1	DATE
Answers for: ATTRIBUTE2	Pet_registration
Correct Feedback	correct
Incorrect Feedback	the correct code is: <u>Pet_ID number(8)</u> Pet_NAME varchar2(35) Pet_Type varchar2(15) must be CAT, DOG, MARINE, EQUESTRIAN or OTHER Pet_DoB date Pet_registration date Complete the create table command below that will create the table with the constraints as described in the description above. Create table **PET_OVERVIEW** (**PET_ID** number(8) **PRIMARY** key, Pet_NAME varchar2(35), Pet_Type varchar2(15) **CONSTRAINT** PET_TYPE_CK CHECK(**PET_TYPE** in ('CAT', 'DOG', '**MARINE**', 'EQUESTRIAN','OTHER'), Pet_DoB **DATE**, **PET_REGISTRATION** date)

6. Jumbled Sentence: The following view is created and upd...

Points: 1.5

Question	The following view is created and updates are permitted. In order to ensure that only validated data is inserted via the update there is a constraint placed on the view definition. Rearrange the code below to indicate the correct order of words. Create or Replace view FAILEDSTUDENTS as select * from marks where mark < 40 with [SYNTAX][SYNTAX2][SYNTAX3] failedstudent_chk;
Drop-down List of Answers	named Check constraint option checking values
Answer	The following view is created and updates are permitted. In order to ensure that only validated data is inserted via the update there is a constraint placed on the view definition. Rearrange the code below to indicate the correct order of words. Create or Replace view FAILEDSTUDENTS as select * from marks where mark < 40 with **Check** **option** **constraint** failedstudent_chk;
Correct Feedback	correct
Incorrect Feedback	*select * from marks* *where mark < 40* *with check option constraint failedstudent_chk;*

7. Multiple Choice: Which of the following statements bes...

Points: 1

Question	Which of the following statements best describes the code below: select sname, count(studentid) from enrolled right join subject using (subjectid) where subjectid like 'COMP%' or 'INFO%' group by sname having count(studentid) > 10;
Answer	A. The code will display the subject names of the subjects that start with a COMP or INFO code B. The code will display the subject names of the subjects that start with a COMP or INFO code and have more than 10 students enrolled on them C. The code will display the subject names of the subjects that start with a COMP or INFO code and have less than 10 students enrolled on them D. This code will not run
Correct Feedback	CORRECT
Incorrect Feedback	The syntax for the where clause is incorrect, the correct code would be Where (subjectid like 'COMP%' or Subjectid like 'INFO%')

8. Multiple Choice: Which one of the following statements...

Points: 1

Question	Which one of the following statements is true about a simple view?
Answer	A. It can be used for DML operations B. It can contain aggregate functions C. It can contain multiple tables D. It can contain a group by clause
Correct Feedback	Correct
Incorrect Feedback	A SIMPLE view can be used to update other tables and run DML commands but if a view contains aggregation, group by clauses or multiple tables it is not classed as a simple view it is a COMPLEX view

9. Fill in Multiple Blanks: The STUDENT table has been created wi...

Points: 2

Question	The STUDENT table has been created with a flaw in one of the attributes, the column AGE needs to be removed, complete the code below to alter the table and remove the attribute. [COMMAND1] TABLE [COMMAND2] [COMMAND3] [COMMAND4] AGE;
Answers for: COMMAND1	ALTER
Answers for: COMMAND2	STUDENT
Answers for: COMMAND3	DROP
Answers for: COMMAND4	COLUMN
Correct Feedback	Correct
Incorrect Feedback	The correct code is: Alter table STUDENT drop column AGE;

10. Fill in Multiple Blanks: The table STUDENT requires and additi...

Points: 1.5

Question	The table STUDENT requires and additional attribute to be added to hold the date of birth for the student. Complete the code below in order to add the column. [COMMAND1] [COMMAND2] STUDENT [COMMAND3] STUD_DOB DATE;
Answers for: COMMAND1	ALTER
Answers for: COMMAND2	TABLE
Answers for: COMMAND3	ADD
Correct Feedback	CORRECT
Incorrect Feedback	The correct code is: ALTER TABLE STUDENT ADD STUD_DOB DATE;

E.4 Exam

Fill in Multiple Blanks: The following equation calculates the... — Points: 5

Question: The following equation calculates the storage parameters for the ORDERLINE table. Complete the equation to determine the INITIAL and NEXT values for the table, the extents should hold 1 years data.

[A] * ([B]/[C]) = 6 bytes of characters

+ [D] bytes of numerical data

Number of orderlines a month = [E]

As no historical order data is to be transferred the INITIAL and NEXT are to be sized the same at 2M

Answers for: A	11
Answers for: B	10
Answers for: C	18
Answers for: D	20
Answers for: E	60000
Correct Feedback	correct

Incorrect Feedback: The key element to determine the number of lines held is the to identify that there are 1000 orders placed a month, each of these orders has an average of 5 lines associated with it, therefore there are 5,000 lines

11 * (10/18) = 6 bytes of characters

+ 20 bytes of numerical data

Number of orderlines a month = 60000

26 * 60000 = 1,560,000 bytes of data each month will be used. As no historical order data is to be transferred the INITIAL and NEXT are to be sized the same at 2M

True / False: The following statement relates to th... — Points: 1

Question: The following statement relates to the storage parameters of a TABLE.

"When creating a table storage parameters such as INITIAL or NEXT extent sizes must be calculated and defined based on the known or estimated data within the table. If a table is created without these being specifically defined then the DBMS will take the default storage parameters from the tablespace that the table is to be stored in and create the table inserting using these values."

Is the statement above true or false?

Answer:
- ✔ True
- False

Correct Feedback: correct

Incorrect Feedback: Where no storage parameters are stipulated then the DBMS will look at the tablespace that is associated and apply those, if no tablespace is stipulated then the DBMS will assume the table is to be held in the default tablespace for that user.

Multiple Choice: Which of the following statements is ... — Points: 1

Question: Which of the following statements is NOT correct about natural joins? (select only 1)

Answer:

A. ✔ The natural join will join on a shared column name regardless of the data type associated with the attribute

B. A natural join will attempt to enforce a relationship where the DBMS can see a shared attribute name and type between 2 tables/relations

C. Joining data on a natural join will not necessarily return the correct data set of results

D. A natural join is a type of inner join and will only return results to the result dataset where a match is made

Correct Feedback: correct

Incorrect Feedback: A natural join is made where the DBMS can see a shared attribute, the shared attribute must have the same name AND data type in order to be used, however, where the identified match is incorrect and the shared elements are coincidence rather than related then the result of the join will be incorrect.
Therefore, not only must the name and data type match, the MEANING must also be consistant in order for the join to be effective.

True / False: Please indicate if the following stat... — Points: 1

Question: Please indicate if the following statement is True or False.
"Structured Query Language (SQL) is the tool used to implement the relational model and using SQL we are able to fully implement the relational model in the real world"

Answer:
- True
- ✔ False

Correct Feedback: correct

Incorrect Feedback: The relational model is not able to be fully implemented in the real world, it is a theoretical model that can be partially realised using SQL but not fully.

True / False: Is the following statement true or fa... — Points: 1

Question: Is the following statement true or false?
"Aggregation is a way of converting the data held within a database into information useful to an organisation."

Answer:
- ✔ True
- False

Correct Feedback: Correct

Incorrect Feedback: A database only holds raw data, organisations required data to be converted to information, this information is often required to be in a format that will highlight the key statistics, aggregation will allow this formatting of data.

Multiple Choice: Which of the following statements is ... — Points: 1

Question: Which of the following statements is true regarding OUTER JOINS? (select one only)

Answer:

A. ✔ The OUTER JOIN retrieves data that is excluded by an INNER JOIN by including all of the records on the specified side regardless of a corresponding match on the other side of the join.

B. The OUTER JOIN will only retrieve data that is matched on attributes and is essensially the same as the INNER JOIN

C. The OUTER JOIN is a theoretical principle only and cannot be realised using standard SQL

D. The OUTER JOIN will retrieve all the data that is excluded by the INNER JOIN and only display the data that is not matched on both sides

Correct Feedback: correct

Incorrect Feedback: The outer join is used to retrieve data where there is not a complete match in all tables. The outer join will display all the values on one or more tables (as specified in the join) and where no corresponding data is found in the other attributes a NULL value is displayed.

Fill in Multiple Blanks: The following peice of code creates t...

Points: 4

Question	The following peice of code creates the table PRODUCT: create table product (product_id varchar2(10) constraint product_pk primary key, product_supplier number, product_name varchar2(30) not null, product_description varchar2(200), product_stock number check(product_stock > 0)); all the constraints required for the table have been added at the time of creation, with the exception of 1, the FK to the supplier table. Complete the code below to add the constraint AFTER the table has been created. [COMMAND1] table [TABLE1] add [COMMAND2] FK_SUPPLIER [CONSTRAINT_TYPE1]([ATTRIBUTE1]) [COMMAND3] [TABLE2]([ATTRIBUTE2]);
Answers for: COMMAND1	alter
Answers for: TABLE1	product
Answers for: COMMAND2	constraint
Answers for: CONSTRAINT_TYPE1	foreign key
Answers for: ATTRIBUTE1	product_supplier
Answers for: COMMAND3	references
Answers for: TABLE2	supplier
Answers for: ATTRIBUTE2	supplier_id
Correct Feedback	correct
Incorrect Feedback	The correct code to alter the table is: alter table PRODUCT
Answers for: ATTRIBUTE2	supplier_id
Correct Feedback	correct
Incorrect Feedback	The correct code to alter the table is: alter table PRODUCT add constraint FK_SUPPLIER foreign key (product_supplier) references supplier(supplier_id);

True / False: Consider the statement below and indi...

Points: 1

Question	Consider the statement below and indicate the statement is TRUE or FALSE *"The join commonly known as an equi join, requires the 2 values to be identical in format and value. The column/attribute names do not have to be the same but they have to have the same meaning."*
Answer	✔ True False
Correct Feedback	correct
Incorrect Feedback	the statement is true

Multiple Choice: Which of the following statements is ...

Points: 1

Question	Which of the following statements is TRUE regarding NATURAL JOINS (select only 1)
Answer	A. ✔ A Natural Join requires that ALL tables in the SQL statement can be joined by a common attribute that has both identical NAME and MEANING and DATATYPE B. Natural Joins will look at each pairing of tables in isolation and naturally join the pairs together to form a consolidated cluster of data that has been joined appropriately. C. If a SQL statement has been written using natural joins but there is no natural join found by the DBMS then the DBMS will report an error to the system informing it that the SQL statement cannot be run. D. Natural Joins are not a valid type of join in the SQL application of the relational model, they only exist within the theoretical model.
Correct Feedback	correct
Incorrect Feedback	Natural joins will join ALL elements together using a common attribute, if it can't find a common attribute it will join what it can and then use a cartesian product to join the rest to it.

Fill in Multiple Blanks: Complete the code below to create the...

Points: 5

Question: Complete the code below to create the table DELIVERYADDRESS as specified on page 6 of the scenario. There is no need to include storage parameters in this code.

```
CREATE TABLE DELIVERYADDRESS
(
DELIVERY_CUST   VARCHAR2(10)   CONSTRAINT DELADD_FK_CUST REFERENCES [A],
DELIVERY_ID   NUMBER      NOT NULL   [B] DELIDCHK CHECK ([C] > 0),
[D]   [E]      NOT NULL   CONSTRAINT DELHOUSECHK CHECK (DELIVERY_HOUSE > 0),
DELIVERY_ADDR1   VARCHAR2(100),
DELIVERY_ADDR2   VARCHAR2(100),
DELIVERY_TOWN   [F](100),
DELIVERY_CITY   VARCHAR2(100),
DELIVERY_PC   VARCHAR2(8)   NOT NULL,
CONSTRAINT DELADD_PK [G] [H] ([I],[J])
);
```

Answers for: A	CUSTOMER
Answers for: B	CONSTRAINT
Answers for: C	DELIVERY_ID
Answers for: D	DELIVERY_HOUSE
Answers for: E	NUMBER
Answers for: F	VARCHAR2
Answers for: G	PRIMARY
Answers for: H	KEY
Answers for: I	DELIVERY_CUST DELIVERY_ID
Answers for: J	DELIVERY_CUST DELIVERY_ID
Correct Feedback	CORRECT
Incorrect Feedback	the correct code to build the table as specified is:

```
CREATE TABLE DELIVERYADDRESS
(
DELIVERY_CUST   VARCHAR2(10)   CONSTRAINT DELADD_FK_CUST REFERENCES CUSTOMER,
DELIVERY_ID   NUMBER      NOT NULL   CONSTRAINT DELIDCHK CHECK (DELIVERY_ID > 0),
DELIVERY_HOUSE   NUMBER      NOT NULL   CONSTRAINT DELHOUSECHK CHECK (DELIVERY_HOUSE > 0),
DELIVERY_ADDR1   VARCHAR2(100),
DELIVERY_ADDR2   VARCHAR2(100),
DELIVERY_TOWN   VARCHAR2(100),
DELIVERY_CITY   VARCHAR2(100),
DELIVERY_PC   VARCHAR2(8)   NOT NULL,
CONSTRAINT DELADD_PK PRIMARY KEY (DELIVERY_CUST, DELIVERY_ID)
);
```

Fill in Multiple Blanks: Page 10 of the scenario (section 2.4....

Points: 5

Question: Page 10 of the scenario (section 2.4.1) describes a view that will list the "excessive fulfillment times", complete the code below to generate the SQL command to create the view as described.

```
create view excessive_fullfillment as
select cust_firstname [A] ' ' || [B] as "Customer name", order_id [C] "[D]", [E] as "placed on", order_completed as "fulfilled on", (([F] - [G]) - [H]) as "days exceeded"
from orderheader join customer1 on (cust_id = [I])
where [J] is not null;
```

Answers for: A	||
Answers for: B	CUST_SURNAME
Answers for: C	AS
Answers for: D	ORDER
Answers for: E	ORDER_DATE
Answers for: F	ORDER_COMPLETED
Answers for: G	ORDER_DATE
Answers for: H	14
Answers for: I	ORDER_CUSTOMER
Answers for: J	ORDER_COMPLETED
Correct Feedback	CORRECT
Incorrect Feedback	the correct code for the view is:

```
create view excessive_fullfillment as
select cust_firstname || ' ' || cust_surname as "Customer name", order_id as "order", order_date as "placed on", order_completed as "fulfilled on", ((order_completed - order_date) - 14) as "days exceeded"
from orderheader join customer1 on (cust_id = order_customer)
where order_completed is not null;
```

Fill in Multiple Blanks: As part of the exam preperation you w...

Points: 9

Question	*As part of the exam preperation you were given a script which generated existing tables and data, this data needs to be migrated to the tables created as specified within the scenario contained in the green booklet.* *Complete the code below which inserts into the new customer table based on the information contained in the old customer table (for the purposes of understanding this table has been renamed CUSTOMER_OLD).* *You are to assume that the table has been created as specified and that the holding values of 9999 and N/A should be used for the house number and postcode.* **INSERT INTO CUSTOMER(CUST_FIRSTNAME, [ATTRIBUTE1], CUST_ID, CUST_CREDITLIMIT, CUST_HOUSE, CUST_PC) [COMMAND1]([COMMAND2]([ATTRIBUTE2], 0, [COMMAND3](NAME,' ')), SUBSTR(NAME,INSTR([ATTRIBUTE3],' ')+1), [ATTRIBUTE4], CREDIT_LIMIT, [VALUE1],'[VALUE2]'** **FROM CUSTOMER_OLD**
Answers for: ATTRIBUTE1	cust_surname
Answers for: COMMAND1	select
Answers for: COMMAND2	substr
Answers for: ATTRIBUTE2	name
Answers for: COMMAND3	instr
Answers for: ATTRIBUTE3	name
Answers for: ATTRIBUTE4	cust_id
Answers for: VALUE1	9999
Answers for: VALUE2	n/a
Correct Feedback	correct
Incorrect Feedback	The correct code is insert into customer_exam (cust_firstname, cust_surname,cust_id, cust_creditlimit, cust_house, cust_pc) select substr(name,0,instr(name,' ')),substr(name,instr(name,' ')+1), cust_id, credit_limit, 9999, 'n/a' from customer;

Fill in Multiple Blanks: Not all the customers in the company ...

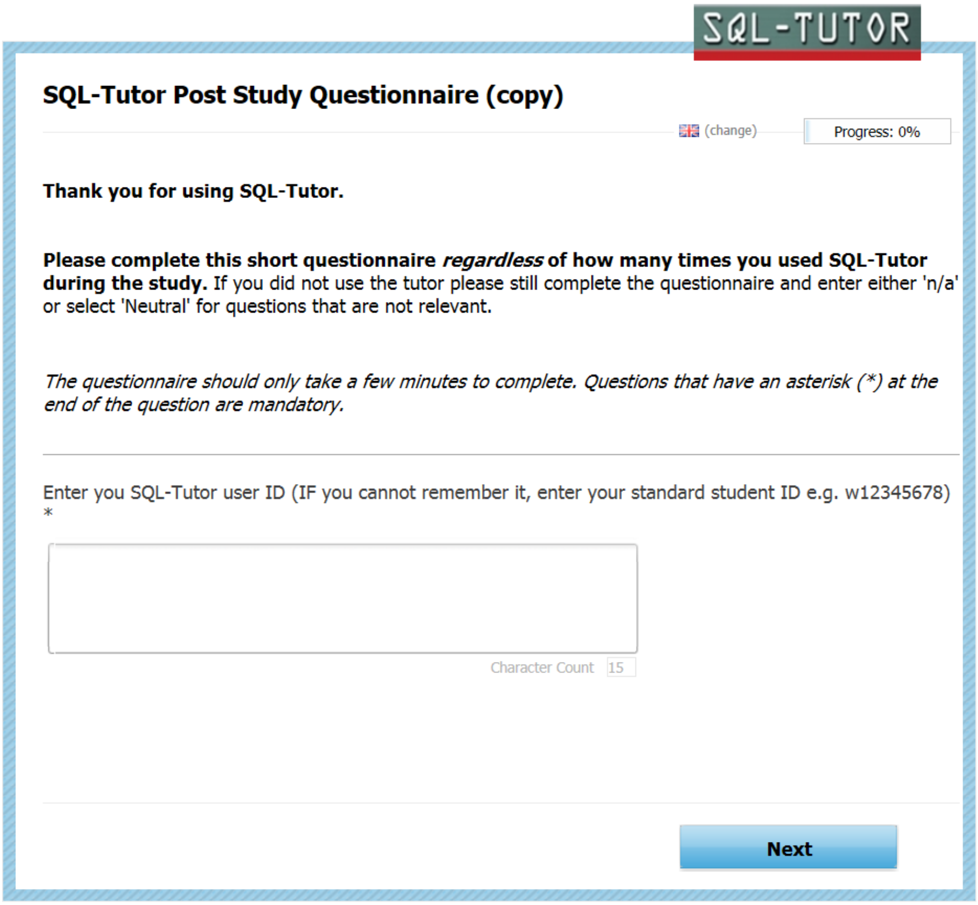
Points: 5

Question	Not all the customers in the company have first and surnames, this has resulted in the data import storing null values for the first name and entering the data into the surname feild for those customers which have only one name. The organisation has decided that customers who meet this criteria should have their name held in the firstname feild and the surname attribute should be set to 'N/A'. Complete the following code so the table is updated for those customers who only have one name. **update CUSTOMER [COMMAND1] cust_firstname = [ATTRIBUTE1], cust_surname = '[VALUE1]' where [ATTRIBUTE2] is [VALUE2];**
Answers for: COMMAND1	set
Answers for: ATTRIBUTE1	cust_surname
Answers for: VALUE1	n/a
Answers for: ATTRIBUTE2	cust_firstname
Answers for: VALUE2	null
Correct Feedback	correct
Incorrect Feedback	the correct code is update customer_exam set cust_firstname = cust_surname, cust_surname = 'n/a' where cust_firstname is null;

Appendix F Post-Study Questionnaire

The screen prints in this appendix show the post-questionnaire as hosted by Toluna QuickSurveys (http://www.quicksurveys.com/). Using such a web site had the advantage of providing access to a tool to create the questionnaire, present it and view the results. The disadvantage of such a tool was that the author had no control over the layout of the individual questions. For example, the text in the questions on the second page appear quashed, but there was no facility to change the width of that column.

Page 1: Opening Page



SQL-TUTOR

SQL-Tutor Post Study Questionnaire (copy)

(change) Progress: 0%

Thank you for using SQL-Tutor.

Please complete this short questionnaire *regardless* of how many times you used SQL-Tutor during the study. If you did not use the tutor please still complete the questionnaire and enter either 'n/a' or select 'Neutral' for questions that are not relevant.

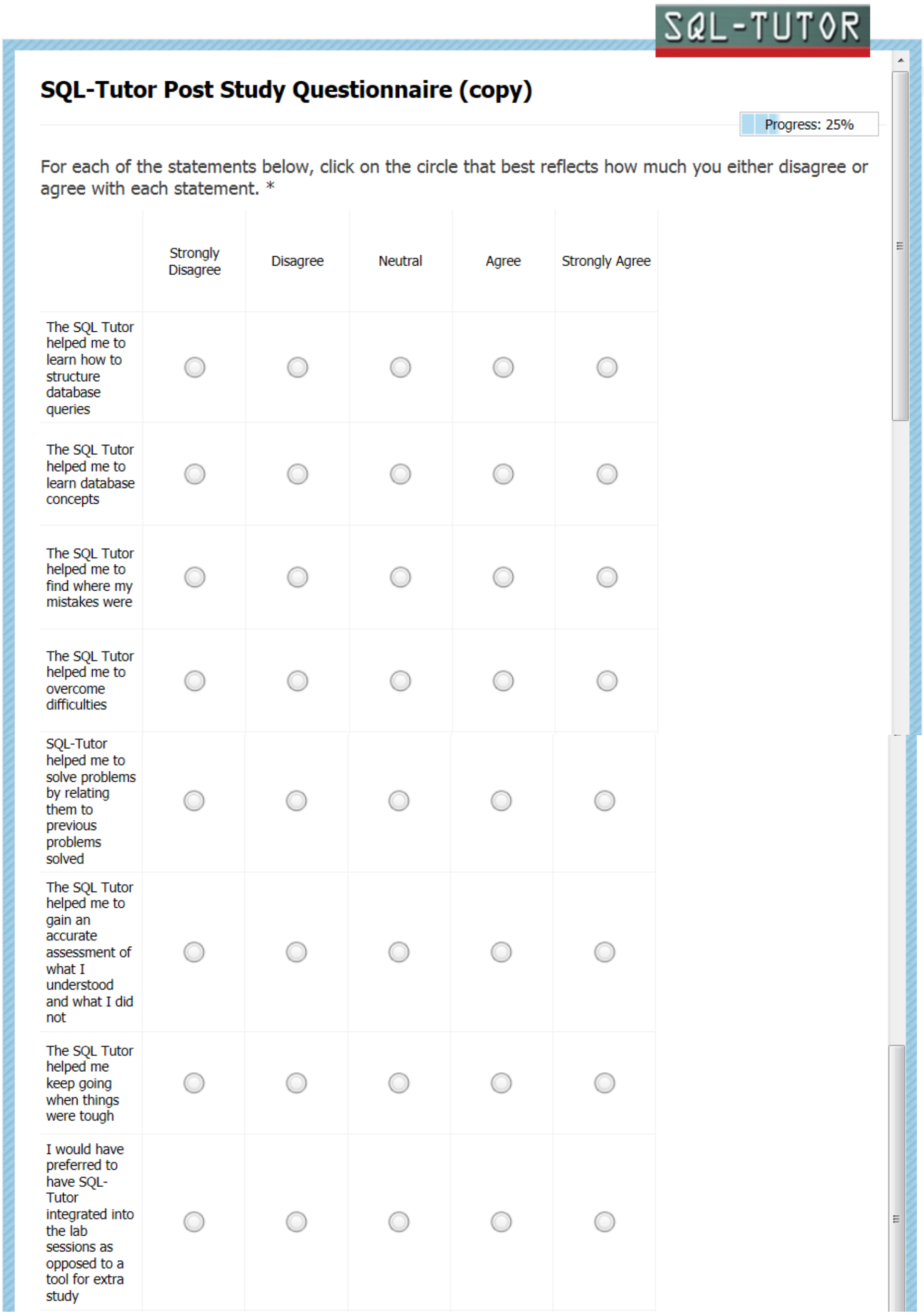
The questionnaire should only take a few minutes to complete. Questions that have an asterisk (*) at the end of the question are mandatory.

Enter you SQL-Tutor user ID (IF you cannot remember it, enter your standard student ID e.g. w12345678) *

Character Count 15

Next

Page 2: Closed Questions



SQL-TUTOR

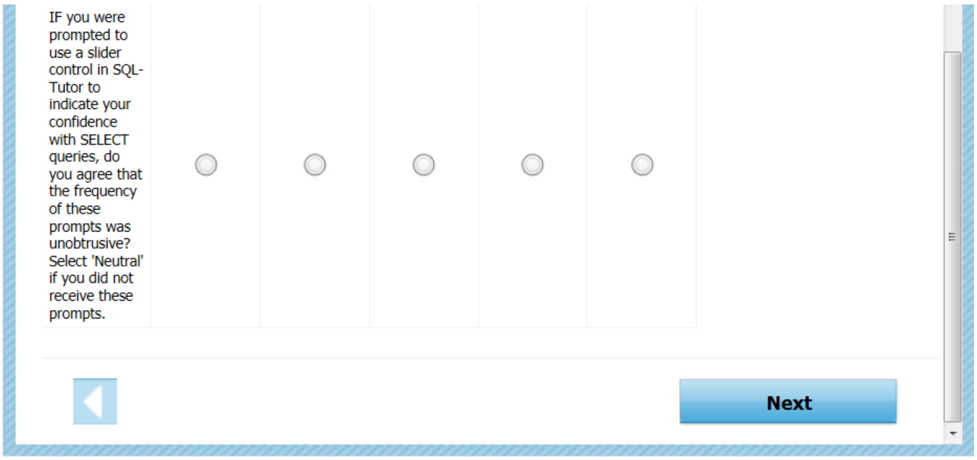
SQL-Tutor Post Study Questionnaire (copy)

Progress: 25%

For each of the statements below, click on the circle that best reflects how much you either disagree or agree with each statement. *

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The SQL Tutor helped me to learn how to structure database queries	○	○	○	○	○
The SQL Tutor helped me to learn database concepts	○	○	○	○	○
The SQL Tutor helped me to find where my mistakes were	○	○	○	○	○
The SQL Tutor helped me to overcome difficulties	○	○	○	○	○
SQL-Tutor helped me to solve problems by relating them to previous problems solved	○	○	○	○	○
The SQL Tutor helped me to gain an accurate assessment of what I understood and what I did not	○	○	○	○	○
The SQL Tutor helped me keep going when things were tough	○	○	○	○	○
I would have preferred to have SQL-Tutor integrated into the lab sessions as opposed to a tool for extra study	○	○	○	○	○

Page 2 continued…


IF you were prompted to use a slider control in SQL-Tutor to indicate your confidence with SELECT queries, do you agree that the frequency of these prompts was unobtrusive? Select 'Neutral' if you did not receive these prompts.

Next

Page 3: Open Questions

SQL-TUTOR

SQL-Tutor Post Study Questionnaire (copy)

Progress: 50%

What did you like MOST about using the SQL-Tutor? *

Character Count 1000

What did you like LEAST about using the SQL-Tutor? *

Character Count 1000

What did you think of the feedback comments from the SQL-Tutor? *

Character Count 1000

Please mention any other aspect of the SQL-Tutor that you would like to comment on.

Character Count 1000

Did you use anything else to help you practice SQL SELECT queries outside of the timetabled lecture and lab session? If yes, what else did you use and why. *

Character Count 1000

IF you were prompted to use a slider control in SQL-Tutor to indicate your confidence with SELECT queries AND you thought the frequency of prompts was obtrusive, when and how often would you have preferred to be prompted (i.e. at the start & end of a session only)?

Character Count 1000

Next

Page 4: Closing Page

SQL-TUTOR

SQL-Tutor Post Study Questionnaire (copy)

Progress: 75%

Your SQL-Tutor user ID from this completed questionnaire will automatically be placed in a draw for 1 of 2 £25 vouchers.

Select answer...

Indicate what type of voucher you would prefer if your SQL-Tutor user ID is drawn.

- Amazon.co.uk
- iTunes

I hope that you have found using SQL-Tutor useful. Thank you for, not only participating in the study, but also completing this questionnaire. The results will remain anonymous and cannot be linked to you after the questionnaires have been collected.

The study is part of my postgraduate research degree (PhD), so your input & feedback are really useful. The aim of the study was to determine if using a tutoring system such as SQL-Tutor was useful over a period of time for learning and practicing SQL SELECT queries.

Thank you and all the best for the rest of your degree studies.

Best wishes,

Alison Hull