World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:10, No:9, 2016

A Framework for SQL Learning: Linking Learning Taxonomy, Cognitive Model and Cross Cutting Factors

Huda Al Shuaily, Karen Renaud

Abstract**—**Databases comprise the foundation of most software systems. System developers inevitably write code to query these databases. The de facto language for querying is SQL and this, consequently, is the default language taught by higher education institutions. There is evidence that learners find it hard to master SQL, harder than mastering other programming languages such as Java. Educators do not agree about explanations for this seeming anomaly. Further investigation may well reveal the reasons. In this paper, we report on our investigations into how novices learn SQL, the actual problems they experience when writing SQL, as well as the differences between expert and novice SQL query writers. We conclude by presenting a model of SQL learning that should inform the instructional material design process better to support the SQL learning process.

Keywords**—**Pattern, SQL, learning, model.

I. Introduction

An understanding of the way novices approach SQL is essential in order to offer insights into the problems students encounter and to improve SQL teaching. Researchers have attempted to isolate those factors that impact on and influence SQL learning. Some point to inherent human factors [30], [31], [41] where others have focused on query language features and tried to explain how these affect the learning process [30], [31], [36], [40], [42], [43]. Even the effects of a particular teaching method have been studied [35]. While these studies have merit, our approach was to start off by considering human cognition as the initial focus, and to consider the impact of human cognition on SQL learning.

A. SQL Learning Model

To support SQL learning, we first have to consider what cognitive steps SQL writers engage in.

Reisiner [29] proposed a SQL writing process whereby a user generates a set of lexical items and a query template, merging of the lexical items with the template to generate the final query as shown in Fig. 1. This model is also related to the idea of semantic and articulator distances as used in Liao and Palvia [20]. Mannino [23] proposed a simpler two-step model: progressing from problem statement to database representation, and from database representation to a database query language statement, as illustrated in Fig. 2.

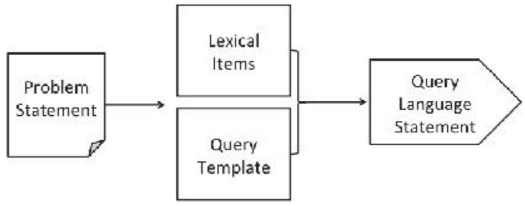

Fig. 1 Query writing model adapted from [29]

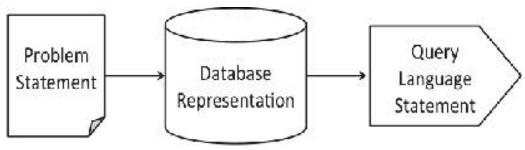

Fig. 2 Mannino's [23] Query Writing Model

Fig. 3 presents an alternative three-stage cognitive model of database query proposed by Schlager & Ogden [34]:

- *Query formulation* (stage 0): Decide what data they need to solve the problem. One example is "*I need to know the average salary of employees who work in the sales department*." This stage relies on knowledge of the application domain.
- *Query translation* (stage 1): Use the output from stage 0 as input, and decide what elements of the data model are relevant, and what the necessary operations are. One example of the output of this stage is "*The employee relation is needed, the column salary is to be selected and the average to be calculated and a restriction of working in the sales department must be specified on column department*." The output of this stage is usually retained internally by experts but written down by novices.
- *Query writing* (stage 2): For the example in the previous stage, to translate into SQL, would be: "*select AVG (salary) from employee where…*" This stage is heavily dependent on the particular query language's syntax and semantics.

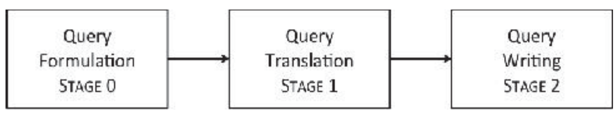

Fig. 3 Three stage cognitive model adapted from [34]

Huda Al Shuaily is with the Higher College of Technology, Muscat, Oman, P.O Box 74, Code 133, (e-mail: huda.alshuaily@hct.edu.om).

Karen Renaud School of Computing Science, University of Glasgow Sir Alwyn Williams Building, Glasgow, G12 8RZ (e-mail: karen.renaud@glasgow.ac.uk).

These models do not particularly address learner cognition and learning stages. SQL novices seem to lack a rich deep understanding of the language construct and the way in which such constructs are used to solve problems [25]. This suggests that Mannino's [23] model might more accurately depict an expert query writing process than that of a novice. Novices may lack the strategic knowledge i.e. the ability to apply syntactic and semantic knowledge to solve novel problems [37]. Strategic knowledge supports stage 0 and stage 1 of the model in Fig. 2, and without it, a novice might very well go straight to stage 2, to the detriment of learning and subsequent query quality.

II. Cognition & Mental Models

To develop SQL mastery, it is essential to consider the supporting cognition. According to Robins et al. [32], "learning" means the construction of schemas, where a schema is "*a structured chunk of related knowledge*". Learning either constructs new schemas or modifies and combines existing schemas in order to produce new, more abstract schemas. A mental model is made up of a schema plus the cognitive processes for manipulating and modifying the knowledge stored in a schema [26]. In order to master SQL, query writers draw on a mental model that is constructed from the requisite concepts (syntax and semantics), together with an understanding of how to apply the concepts within a particular context.

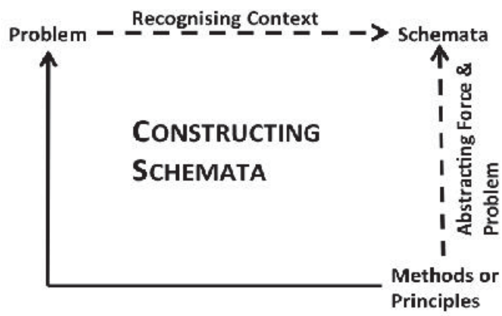

Fig. 4 (a) The role of schemata in Problem Solving

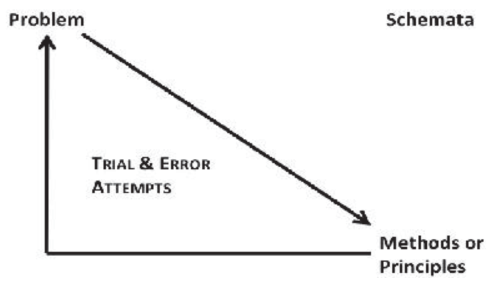

Fig. 4 (b) the trial and error approach

Mere knowledge of SQL syntax and semantics is not sufficient to achieve mastery. Many students can parrot such knowledge in exams yet do not know how to apply it in context. What they appear to lack is an abstract construct which can be applied to a variety of matching contexts. The mental model, with its schemata building blocks, is iteratively constructed as learners write SQL during problem solving. This process is depicted in Fig. 4 (a). This depicts an ideal situation, where schemata are constructed during problem solving. What it does not convey is how the process can become derailed, and how best to ensure that problem-solving does indeed result in essential schemata formation.

In our combined years as database lecturers, the authors have frequently observed students engaging in the process depicted on the right of Fig. 3. These learners inhabit the bottom left triangle. The net effect is that schema are not constructed; for whatever reason learners experience difficulties matching the knowledge learnt in lectures with the knowledge required solving SQL problems. They fail to abstract principles and solve posed problems using a trial and error approach. They try using SQL constructs randomly without any perceivable strategy and without developing any deep understanding of the underlying principles. An understanding of why this is happening will inform our teaching process, and help us to design our instructional materials so that they will better support and encourage the schema construction process.

III. SQL Learning Taxonomy

According to Cutts et al. [10] learning taxonomies provide researchers with an essential shared vocabulary. Probably the most widely applied taxonomy was proposed by Bloom [7]. Andreson et al. [6] proposed an updated version of Bloom's taxonomy to correspond with the ways learning objectives are typically described as cognitive activities. They argue that students' progress through *Remembering*; *Understanding; Applying; Analysing; Evaluating* and finally *Creating* stages. Gorman [15] proposes a simpler model, arguing a progression from an understanding of *what*, followed by *how*, then *when* and finally *why*.

The applicability of a number of learning taxonomies to Computer science has been considered [14], [18], [19], [39]. Lahtinen [19], in particular, investigated whether a subject-specific taxonomy would be of more use to computer science instructors than the existing generic ones. Lahtinen concluded that Bloom's cognitive activities were indeed applicable to computing generally. Hence, there is some justification for applying them to SQL learning. Since Computing is essentially a skill-based subject, the three stages of Bloom, which comprise application of principles, are particularly important. These stages reflect the fact that problem solving is the essence of computing skill mastery. The fact that researchers recommend incorporating problem solving as a primary learning activity confirms this [1], [8], [16], [17], [21].

One could argue that solving problems and producing an effective and efficient solution is the core activity of the computer science practitioner. Computer science, at its core, involves modelling the real world, representing domains of the most varied nature and complexity, representing knowledge in general and dealing with processes and solutions to problems in such domains. Therefore, any proposed taxonomy should have, at its core, problem solving, to be engaged in after the basic knowledge is delivered and comprehended. We therefore propose an SQL learning taxonomy, which models how learners should assimilate specific SQL topics (Fig. 5). It consists of four main areas, each of which map to the related cognitive processes:

- *Remembering SQL Concepts*: includes the following cognitive process: Recognizing and Recalling.
- *Comprehending SQL Concepts*: includes the process of SQL Reading, Interpreting and Explaining.
- *Practicing SQL* (problem solving) this level consists of three interleaving activities that represent SQL problem solving:
  - Problem formulation (Analysis)
  - Problem translation (Synthesis): Differentiating, Organizing, Attributing and translation of the given problem
  - SQL Query writing (Application): Executing and Implementing query
  - SQL Query checking (Evaluation): Checking, Critiquing and evaluating the output results. For example, query debugging.
- *Creating* includes reflecting, making judgements, and conceivably constructing mental models. This includes, for example, query modification.

This entire process will undoubtedly be affected by factors other than human cognition. The next step in understanding the SQL learning process is to consider the cross-cutting factors that affect SQL learners in general.

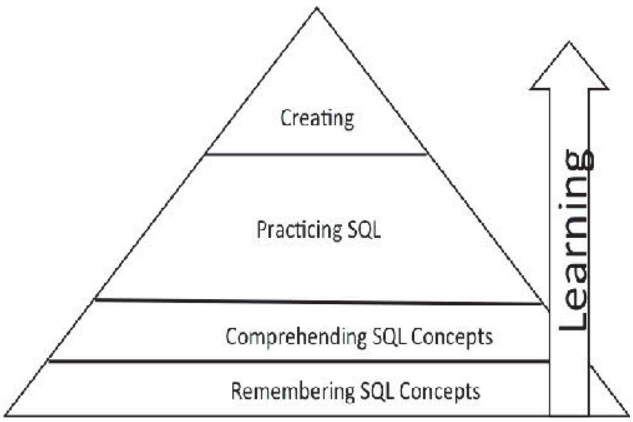

Fig. 5 SQL Learning Taxonomy

IV. Cross-Cutting Factors

The SQL learning taxonomy proposed in Fig. 5 depicts a perfect learning model. It incorporates knowledge of human cognition, but does not necessarily accommodate individual learner differences. To identify the other factors that influence SQL learnability we interviewed a group of participants who had learnt SQL. In addition, a questionnaire was used to collect feedback from another 75 students. Another questionnaire collected feedback on teaching SQL from fourteen experienced academics. Grounded Theory was used to identify the factors in the responses, identifying themes related to success in learning SQL.

The emergent themes were:

1. Learner attributes such as personal attitude, previous experience, lack of problem solving skills, and general skill acquisition abilities;
2. The features of the SQL language;
3. SQL-specific cognitive tasks involved in the problem-solving process;
4. The instructional materials provided during teaching activities.

These themes are explored within the following subsections.

B. Learner Attributes

The role of personal attitude towards learning in general is undoubtedly one of the most controversial and fascinating areas of research. Researchers have proved that a personally positive attitude towards learning plays a vital role in educational settings and influences learning processes and affects achievement [4], [28]. Some studies measured the relationship between students' achievements and personal attitude towards learning mathematics [2], [9], [11], [13], [22]. The impact of negative attitudes among novices learning programming languages has been highlighted as well [32], [33].

We interviewed seven students and asked database educators to complete an online questionnaire. The interview questions sought to examine the relationship between personal feelings and achieved SQL expertise. Five of the students reported feeling slightly uncomfortable about using SQL and only two were comfortable with SQL. Those who felt slightly uncomfortable attributed this to a lack of practice. Students claimed that they did not use SQL after completing their database course and even within the course had limited opportunities to practice. Some examples of their responses are given in Table I.

TABLE I
`Participants' Responses Revealing Personal Attitude

"I do not have all that much experience with it, the only time that I have contact with it is only during class".
"I have learned specific things, not getting the chance to apply them, practicing the logic of it makes it easier"
"It needs practice, perhaps not very easy to stay motivated doing these concepts of harder nature"

The two students who did feel comfortable with SQL attributed this either to their own attitude towards database concepts in general, or to their accumulated experience with SQL. They rated themselves as advanced SQL writers. One of the students provided the following comment in this respect:

"*I like the whole concepts of databases; I am very keen in learning about databases rather than programming.*"

This comment suggests a clear positive attitude towards SQL, notably missing from the responses of those who were not as comfortable with SQL. This finding supports the earlier finding by [27] in terms of all database concepts:

"*Our experience has demonstrated that beginning* *database students are often lackadaisical, in terms of motivation, to grasp the precise meaning and definitions of key terms used in the database field*."

During this study, many students spontaneously attempted to relate their experience in learning programming languages to experiences learning SQL. This is evident from their responses as shown in Table II. Hence, the difference between imperative and object oriented programming, on the one hand, and a functional paradigm, on the other, seemed to make SQL hard to understand and use. Apart from programming languages, other courses that might have an impact on SQL learning are mathematics courses. This was referred to in some of the responses provided by educators in the online questionnaire:

"Mathematical notions help (e.g. set theory) but vice versa, the mathematical notions can also follow learning database concepts."

"*Not enough knowledge about mathematics and logic.*"

The knowledge of some concepts, such as relational database theory, database structure and relational algebra, might also have some impact on learners' ability to master SQL concepts. Relational algebra courses are difficult to teach because they appear too theoretical, and do not have an obvious practical application. This was addressed in some of the responses provided by educators in the online questionnaire:

"Because of the relational algebra behind it, students struggle with relational algebra but not the SQL itself."

Knowledge of query language syntax without an understanding of relational database structure can lead to expensive end-user mistakes [11].

C. SQL Features

Some participants reported experiencing difficulties with SQL syntax. This is shown in the relevant quotes provided by students and educators in Table III below.

The ordering of SQL syntax (SELECT…FROM…WHERE) is an unintuitive way of expressing a query. SQL is a declarative language that expresses *what* the desired result is, without the writer specifying *how* it is to be achieved. This is undeniably counter-intuitive to minds that are accustomed to specifying everything in intricate detail in other programming.

D. SQL-Specific Cognition

Felix [12] highlighted the relationship between the cognitive structure of a language and cognition involved in problem solving tasks. Hence, the development of SQL-specific problem solving skills must be considered. Table IV provides some quotes from educators related to applying SQL concepts in order to solve specific problems. Moreover, Soloway and Spohrer [38] observed that while students might know the individual statement syntax and semantics, they often fail to combine different features correctly. Some students reported writing simple SQL queries with ease but being stumped by complex queries.

TABLE II
PARTICIPANTS' EXPERIENCE WITH OTHER LANGUAGES

"SQL is quite different from programming languages that I study. It requires a certain reasoning that I did not have."
"SQL is not like Java when you solve SQL problem you do not know which answer is the right one."
"the problem is getting into the way of thinking in terms of tables, as opposed to classes, etc.
"It is different to other programming language; the logic behind it is different."

TABLE III
PARTICIPANT'S RESPONSE TO SQL SYNTAX

Student	*"I cannot see the relation between the statements and their context."*
Student	*"SQL syntax is quite difficult to remember…"*
Educator	*"Syntax details can be difficult; making concepts hard to embed in formal language," and "I think that it is because the ordering of the syntax (SELECT…FROM…WHERE) is not a natural way of expressing a query - it is more usual to identify the constraints first, and then work out what tables are needed, and then work out how to join them together."*
Educator	*"SQL is declarative and having a procedural mind set is easier."*
Student	*"SQL problems are an inherent problem in a declarative natural language."*

TABLE IV
EDUCATORS' COMMENTS ABOUT SQL PROBLEM SOLVING

"SQL concepts are not difficult to understand or apply as an individual concept, but when you are given a complex situation where you have to apply many concepts then there is the problem."
"In general, queries require different aspects of SQL to perform the requested function."
"Knowing a language's syntax does not mean students will be able to use it in problem solving."

Our previous discussion referred to the construction of schemata during the learning process. Once constructed, these are deployed in solving problems. This means that experts use their schemata while novices do not have these at their disposal. Table V shows the participants' comments related to problem solving.

TABLE V
COMMENTS RELATED TO PROBLEM SOLVING

Student	*"SQL requires a certain reasoning that I did not have."*
Student	*"when you solve SQL problem you do not know which answer is the right one"*
Student	*"the problem getting in the way of thinking in terms of tables, as opposed to classes, etc."*
Educator	*"... queries require different aspect of SQL to perform the request function."*
Educator	*"SQL is a very "tight" and minimalist language..."*
Student	*"SQL is not the natural way of thinking."*
Student	*"It's too ambiguous; there are too many ways to achieve the same thing, with no standard approach to problem solving."*

Learners should ideally proceed through all four cognitive stages (see Fig. 5).

1. The first step is to have an understanding of the underlying SQL concepts and how they are used (remembering and comprehension of SQL concepts.)
2. Next they should analyse: divide the problem into parts within the context of the problem. Thereafter, they

synthesise: match SQL concepts to different constituent parts.

3. Only then should they start writing the query.
4. Finally, they check to see whether the query has indeed solved the problem. If not, they should ideally go back to one of the previous stages until the checking stage delivers a judgement that the problem has been solved.

We observed two groups of 15 novices and two experts solving SQL problems to observe their problem solving strategies. What we noticed was that the experts, when presented with a problem, generate a number of alternative ways of solving the problem. They then decided on an optimal approach and start to write the query. The novices, however, start writing the SQL immediately, skipping the analysis and synthesis stages. The observation process led to the SQL problem-solving model proposed in Fig. 6.

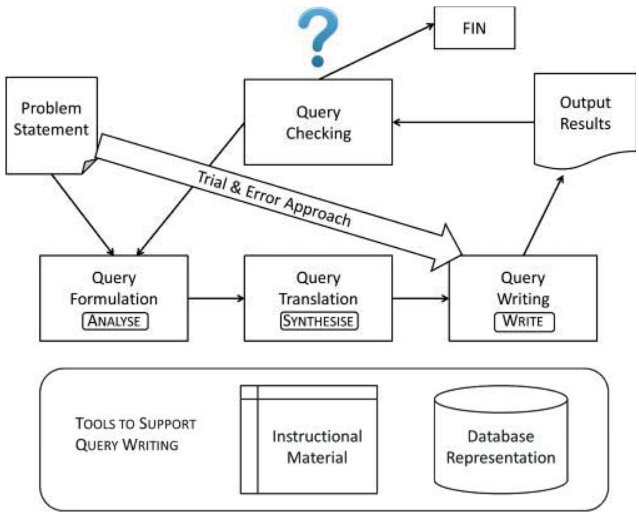

Fig. 6 SQL problem solving – The Practice Stage of SQL Learning Taxonomy

Our observation was that novices did not engage in analysis and synthesis: they moved straight on to writing the query. This kind of practice does not construct of schema.

Al Shuaily [3] analysed the influence of the current SQL teaching methods and approaches. She concluded that SQL learners needed a well-designed teaching method to expose them to common SQL problems. In this way, problem-solving skills can be nurtured and expertise developed. Novices should not deploy a tactical trial and error approach and rather move to a more strategic approach which is more likely to result in schema construction.

E. Instructional Material

The instructional materials should help learners to understand SQL concepts and guide them through the essential problem solving steps. The instructional materials should present different types of knowledge, such as:

- declarative and procedural knowledge [5]; and,
- conditional knowledge [44].

Bloom & Broder and Mayer [7], [24] suggest that problem-solving teaching methods should focus on the modelling of the "process" steps, rather than on the "product" and give students practice in comparing their strategies to those of expert models.

Unfortunately, the majority of current teaching and learning approaches do not encourage analogical problem solving. Perhaps that is why novices deploy a trial and error approach. It is possible to argue that students suffer from an inability to recognize the context of the problem and lack the ability to apply abstract knowledge because they do not have the necessary mental models to do this. Thus, students solve problems by mapping random SQL concepts and knowledge to the set problem without having any plan or justified approach. In other words, students might have progressed through the remembering and understanding stages of Bloom, but falter during the practice stage and do not really master SQL.

V. SQL Learning Framework

In this paper, we present a model of SQL learning that models the dimensions of SQL mastery as shown in Fig. 7.

The framework aims to guide both educators and course designers toward effective instructional materials and consists of three main areas:

1- Cognition and mental models: Presents the development of mental model throughout the learning process.
2- SQL learning taxonomy (from Fig. 5): Illustrates the proposed learning taxonomy in, which relates to SQL knowledge and skill acquisition.
3- Cross-cutting factors that highlight the factors that influence ease of learning.

VI. Conclusion

SQL is a declarative computer language and is the *de facto* industry database language. This paper suggests a framework of SQL learning. The purpose of the proposed SQL learning framework is to articulate and highlight those factors that that impact on SQL mastery.

The study also used various research methods to identify the cross-cutting factors and processes that impact mastery of SQL. While human cognition is fixed, we would argue that the one factor we, as educators, *can* influence, is to align the instruction and the instructional materials as closely to the cognitive process as possible.

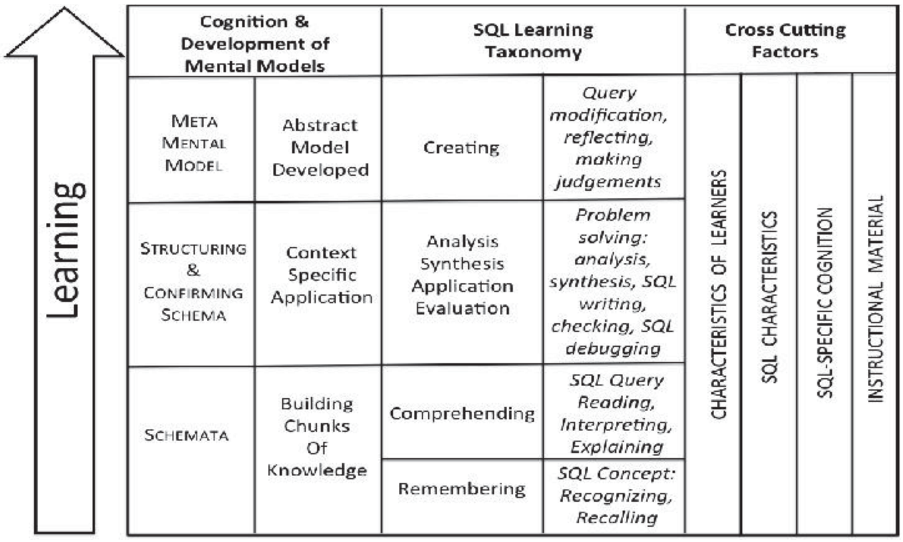

Fig. 7 SQL Learning Framework

REFERENCES

[1] Aamodt, A. (1991). A knowledge-intensive, integrated approach to problem solving and sustained learning. *Knowledge Engineering and Image Processing Group. University of Trondheim*, 27-85.

[2] Aiken Jr, L. R. (1976). Update on attitudes and other affective variables in learning mathematics. *Review of Educational Research*, 293-311.

[3] Al-Shuaily, H. (2012, July 2012). Analyzing the Influence of SQL Teaching and Learning Methods and Approaches. Paper presented at the 10th International Workshop on the Teaching, Learning and Assessment of Databases. UK, London

[4] Ames, C., & Archer, J. (1988). Achievement goals in the classroom: Students' learning strategies and motivation processes. *Journal of Educational Psychology, 80*(3), 260.

[5] Anderson, J. (1987). Skill acquisition: Compilation of weak-method problem solutions. *Psychological Review, 92*, 192-210.

[6] Anderson, L. W. E., Krathwohl, D. R. E., Airasian, P. W., K.A.Cruikshank, R.E.Mayer, P.R.Pintrich, . . . M.C.Wittrock. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives (Complete edition)*. New York: Longman.

[7] Bloom, B. S., & Broder, L. J. (1950). *Problem-solving Processes of College Students: An Explanatory Investigation*: University of Chicago Press.

[8] Bonar, J., & Soloway, E. (1985). Preprogramming Knowledge: A major source of misconceptions in novice programmers. *Human-Computer Interaction, 1*, 133-161.

[9] Caraway, S. D. (1985). Factors Influencing Competency in Mathematics Among Entering Elementary Education Majors.

[10] Cutts, Q., Esper, S., Fecho, M., Foster, S. R., & Simon, B. (2012). The abstraction transition taxonomy: developing desired learning outcomes through the lens of situated cognition 10.1145/2361276.2361290 *Proceedings of the ninth annual international conference on International computing education research* (pp. 63-70). Auckland, New Zealand: ACM.

[11] Davies, N., & Savell, J. (2000). *Maths is like a bag of tomatoes": Student attitudes upon entry to an Early Years teaching degree.* Paper presented at the Teacher Education Forum of Aotearoa New Zealand Conference, Christchurch.

[12] Felix, S. (1981). *Competing cognitive structures in second language acquisition.* Paper presented at the European-North American Workshop on Cross-Linguistic Second Language Acquisition Research, Lake Arrowhead CA, September. Cited and discussed in Birgit Harley (1986) Age in Second Language Acquisition. Clevedon: Multilingual Matters.

[13] Fennema, E., & Sherman, J. (1977). Sex-related differences in mathematics achievement, spatial visualization and affective factors. *American educational research journal, 14*(1), 51-71.

[14] Fuller, U., Johnson, C. G., Ahoniemi, T., Cukierman, D., Hern\, I., \#225, . . . Thompson, E. (2007). Developing a computer science-specific learning taxonomy. *SIGCSE Bull., 39*(4), 152-170. doi: 10.1145/1345375.1345438

[15] Gorman, M. E. (2002). Types of Knowledge and Their Roles in Technology Transfer *The Journal of Technology Transfer* (Vol. 27, pp. 219-231): Springer Netherlands.

[16] Gray, P. H. (2001). A problem-solving perspective on knowledge management practices. *Decision Support Systems, 31*(1), 87-102. Grootenboer, P. (2010). Beliefs, attitudes and feelings students learn about mathematics. *Far East Journal of Mathematical Education, 5*(1), 31-52.

[17] Ismail, M. N., Azilah, N., Naufal, U., & Kelantan, U. T. M. C. (2010). Instructional strategy in the teaching of computer programming: A need assessment analyses. *TOJET, 9*(2), 125-131.

[18] Johnson, C. G., & Fuller, U. (2006). *Is Bloom's taxonomy appropriate for computer science?* Paper presented at the Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006.

[19] Lahtinen, E. (2007). *A categorization of novice programmers: A cluster analysis study.* Paper presented at the Proceedings of the 19th annual Workshop of the Psychology of Programming Interest Group.

[20] Laio, C., & Palvia, P. C. (2000). The impact of data models and task complexity on end-user performance: an experimental investigation. doi: 10.1006/ijhc.1999.0358. *International Journal of Human-Computer Studies, 52*(5), 831-845.

[21] Linn, M. C., & Clancy, M. J. (1992). The case for case studies of programming problems. *Communications of the ACM, 35*(3), 121-132.

[22] Ma, X., & Kishor, N. (1997). Assessing the relationship between attitude toward mathematics and achievement in mathematics: A meta-analysis. *Journal for research in mathematics education*, 26-47.

[23] Mannino, M. V. (2001). *Database Application Development and Design,* : McGraw-Hill Company,Inc.

[24] Mayer, R. E. (2008). *Learning and instruction*: Merrill.

[25] McGill, T. J., & Volet, S. E. (1997). A conceptual framework for analysing students' knowledge of programming. *Journal of Research on Computing in Education, 29*(3), 276-297.

[26] Merrill, M. D. (2000, 2000). *Knowledge objects and mental models.* Paper presented at the Advanced Learning Technologies, 2000. IWALT 2000. Proceedings. International Workshop on.

[27] Mohtashami, M., & Scher, J. M. (2000). *Application of Bloom's Cognitive Domain Taxonomy to Database Design.* Paper presented at the Proceedings of ISECON (information systems educators conference).

[28] Pekrun, R. (2011). Emotions as drivers of learning and cognitive development. *New Perspectives on Affect and Learning Technologies*, 23-39.

[29] Reisner, P. (1977). Use of Psychological Experimentation as an Aid to Development of a Query Language. *Software Engineering, IEEE Transactions on, SE-3*(3), 218-229.

[30] Reisner, P. (1981). Human Factors Studies of Database Query Languages: A Survey and Assessment. *ACM Comput. Surv., 13*(1), 13-31.

[31] Reisner, P., Boyce, R. F., & Chamberlin, D. D. (1975). Human factors evaluation of two data base query languages: square and sequel *Proceedings of the May 19-22, 1975, national computer conference and exposition* (pp. 447-452). Anaheim, California: ACM.

[32] Robins, A., Rountree, J., & Rountree, N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education, 13*(2), 137-172.

[33] Rodrigo, M. M. T., Baker, R. S., Jadud, M. C., Amarra, A. C. M., Dy, T., Espejo-Lahoz, M. B. V., . . . Tabanao, E. S. (2009). *Affective and behavioral predictors of novice programmer achievement.* Paper presented at the ACM SIGCSE Bulletin.

[34] Schlager, M. S., & Ogden, W. C. (1986a). A cognitive model of database querying: a tool for novice instruction. *SIGCHI Bull., 17*(4), 107-113.

[35] Schlager, M. S., & Ogden, W. C. (1986b). A cognitive model of database querying: a tool for novice instruction 10.1145/22339.22357. *SIGCHI Bull., 17*(4), 107-113.

[36] Shneiderman, B. (1978). Improving the human factors aspect of database interactions. *ACM Trans. Database Syst., 3*(4), 417-439.

[37] Shneiderman, B., & Mayer, R. (1979). Syntactic/semantic interactions in programmer behavior: A model and experimental results. *International Journal of Parallel Programming, 8*(3), 219-238.

[38] Soloway, E., & Spohrer, J. C. (1989). *Studying the novice programmer*: Lawrence Erlbaum Hillsdale, NJ.

[39] Starr, C. W., Manaris, B., & Stalvey, R. A. H. (2008). *Bloom's taxonomy revisited: specifying assessable learning objectives in computer science.* Paper presented at the ACM SIGCSE Bulletin.

[40] Suh, K. S., & Perkins, W. C. (1994, 4-7 Jan. 1994). *The effects of a system echo in a restricted natural language database interface for novice users.* Paper presented at the System Sciences, 1994. Proceedings of the Twenty-Seventh Hawaii International Conference on

[41] Thomas, J. C., & Gould, J. D. (1975). A psychological study of query by example *Proceedings of the May 19-22, 1975, national computer conference and exposition* (pp. 439-445). Anaheim, California: ACM.

[42] Vassiliou, Y., Jarke, M., Stohr, E. A., Turner, J. A., & White, N. H. (1983). Natural Language for Database Queries: A Laboratory Study. *MIS Quarterly, 7*(4), 47-61.

[43] Welty, C., & Stemple, D. W. (1981). Human factors comparison of a procedural and a nonprocedural query language. *ACM Trans. Database Syst., 6*(4), 626-649.

[44] Winograd, P., & Hare, V. (1988). *Direct instruction of reading comprehension strategies: The nature of teacher explanation*. San Diego: Academic Press.