

Received May 12, 2020, accepted May 24, 2020, date of publication May 27, 2020, date of current version June 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997934

Structured Query Language Learning: Concept Map-Based Instruction Based on Cognitive Load Theory

SHIN-SHING SHIN 

Department of Information Science and Management Systems, National Taitung University, Taitung 950, Taiwan

e-mail: shinshing@gmail.com

This work was supported by the Taiwan Ministry of Science and Technology under Grant 108-2221-E-143-002.

ABSTRACT Structured query language (SQL) is difficult to master because the execution process of SQL statements is invisible. When learning to construct an SQL query, learners must visualise the evolution process of the intermediate datasets of the SQL statement in working memory, which may burden learners' cognitive load and consequently jeopardise learning outcomes. This study describes the execution process of SQL statements by using concept maps to improve learners' understanding of SQL. An empirical experiment was conducted using two database courses, namely concept map-based and conventional instruction, to examine the relationship between concept maps and the understanding of SQL from a cognitive load theory perspective. The experimental results demonstrated the superiority of concept map-based instruction over conventional instruction because concept map-based instruction reduces extraneous load but increases germane load. Concept map construction facilitated learner engagement and promoted meaningful learning. Studying the instructors' concept maps helped learners follow the cognitive structures used by instructors to perform SQL queries, and enabled them to perceive the execution process of SQL queries relatively easily. These results potentially help educators understand the learning difficulties caused by the declarative nature of SQL and motivating researchers to resolve the inherent problem by considering learners' cognitive processes.


INDEX TERMS Structured query language (SQL), cognitive load theory, concept maps, semantic network theory.

I. INTRODUCTION

Structured query language (SQL) is the standard for accessing relational databases. An aim of database courses is to enable learners to express data retrieval requests in SQL statements. However, research demonstrates that SQL is a complex language that is difficult to learn [1]–[4]. SQL is essentially a declarative language that allows users to specify what they want and not how to obtain it. The declarative nature of SQL is difficult for learners to grasp because the execution process of SQL statements is invisible to learners [5]–[7]. Therefore, when learning to compose an SQL query, learners must visualise the initial datasets obtained from the 'from' clause and mentally evolve them into the intermediate datasets and then into the resultant dataset. According to cognitive load theory, mentally visualising the query process may place a burden on learners' cognitive load

and consequently jeopardise learning outcomes [8]–[10]. These considerations prompt the following question: *how does the mental imagery problem affect learners' understanding of SQL, and how can the impact of the problem on learning SQL be reduced?*

A main challenge of SQL learning is that its declarative nature compels learners to expend considerable mental effort in understanding SQL. Laybi *et al.* [2] indicated that learners have difficulties in visualising the results of their written SQL queries. Renaud and van Biljon [6] mentioned that learners must have an appropriate understanding of what exactly is happening when SQL queries are executed. Prior and Lister [5] found that learning SQL is particularly difficult if learners cannot understand the evolution process of the intermediate datasets of SQL queries. To alleviate difficulties in learning SQL, some learning tools have been proposed, such as animated pedagogies. eSQL [11] is a tool for learning SQL, which provides the step-by-step animation of an executed SQL statement. However, the animation is

The associate editor coordinating the review of this manuscript and approving it for publication was Gang Li .

scattered over these steps. Learners still must mentally integrate all the fragmented information to understand the overall execution process of the SQL statement. Furthermore, animating advanced SQL queries may be beyond the capabilities of graphical interfaces, thereby reducing the usable range of such a tool [9]. In addition, the mental imagery problem may become worse with the learning difficulties posed by SQL syntax and semantics. SQL syntax is counterintuitive [6], [12]; moreover, understanding the notions of multiple tuple variables, such as join operations and nested-type queries, is difficult. Other concepts that are particularly difficult to grasp include self-join, multitable joins, correlated subquery, and group by with having [11]. The semantics of SQL can quickly become complex, particularly when a query involves aggregate functions, join statements, and subqueries [9].

Instructions for learning SQL may require further strengthening of the process of SQL statement execution by incorporating materials that appropriately represent this process. Kolloffel *et al.* [13] indicated that with regard to complex instructional materials (high mental effort), for learners with little or no prior knowledge of the material, representation formats are particularly critical. According to the learning psychology of Ausubel *et al.* [14] such as that related to concept maps, the use of graphic organisers has been recognised as effective for priming learners to learn by activating prior knowledge and illustrating its relationship with new concepts. Concept maps—the use of which is widespread in science education [15]—are useful for representing the knowledge structure used by learners to interpret problem domains by physically presenting associated information together in a diagrammatic format [16], [17]. Representing materials in an integrated format (e.g. diagrams) can reduce learners' mental effort spent in understanding the related information [18], [19]. Therefore, diagrammatically displaying the intermediate datasets of an executed SQL statement and their involvement into the resultant dataset by using concept maps may reduce learners' mental effort invested in visualising the evolution process. Concept maps can help learners interrelate declarative and procedural knowledge [20]. Accordingly, this study proposes that by incorporating concept mapping techniques into SQL instruction, mental imagery problems may be overcome by describing the execution process of SQL queries in diagrammatic representations.

The current study's motivation for examining the effects of concept maps on SQL learning mainly relates to the inherent problems caused by SQL's declarative nature, difficulties with SQL syntax and semantics, and essential features of concept maps. Thus, the main study objectives were to obtain a deep insight into the difficulties in learning SQL from the perspective of learners' cognitive processes and to understand whether concept map-based instruction assists learners in developing a better comprehension of SQL than does conventional instruction. An empirical experiment was conducted using two database courses, namely a concept map-based course and a conventional course; when

the courses ended, the participants were asked to answer data retrieval questions by writing SQL statements. Next, the participants' problem-solving performance and mental effort were analysed to evaluate the relevance of concept map-based instruction for effectively understanding SQL on the basis of cognitive load theory and semantic network theory. Factors undermining learners' abilities to learn SQL were analysed.

Next, we provide a brief overview of pedagogies for SQL learning (Section 2), explain the related theory and hypothesis (Section 3), describe the research methodology used to examine the research questions (Section 4), elaborate on result analyses (Section 5), describe the findings and their implications (Section 6), and provide a conclusion and suggestions for future studies (Section 7).

II. SQL LEARNING PEDAGOGIES

A literature review revealed that techniques for alleviating the SQL learning difficulties caused by conventional instruction can be categorised into animation, graphical query builders, and feedback.

A. ANIMATION

Animated pedagogies resolve SQL queries from a data-oriented perspective, which diagrammatically illustrates how initial datasets are obtained from an SQL statement and processed by the other clauses to ultimately obtain the resultant dataset. Animated pedagogies are superior to the usual pen-and-paper explanations given in lectures and textbooks. Some animated pedagogies have been proposed, such as eSQL [11], SQL Advanced Visualization (SAVI) [8], Database Query Analyzer (DBQA) [9], and Animated Database Courseware (ADbC) [21].

eSQL, one of the first animated pedagogies for learning SQL, provides step-by-step animation of the execution of SQL select queries. In each execution step, eSQL highlights the clause of the query being executed and provides a textual explanation for the clause in the text area. Rows, columns, and cells targeted in each step are highlighted to emphasise the portion of data being processed by the query. During the process, learners can move forward to the next step or to the final result directly. SAVI is a web-based animated pedagogy with a visualisation approach similar to eSQL. The difference is that the animations generated by eSQL focus on displaying the sequence of intermediate datasets derived from the execution of SQL queries, whereas SAVI places more emphasis on visualising and explaining how the SQL operator works and the way the information is transformed. Moreover, SAVI extends eSQL by adding reversible animation, in which learners are permitted to backtrack in the evaluation of SQL queries. However, SAVI does not provide an explanation for SQL operators in each step. Similar to eSQL and SAVI, DBQA evaluates queries one step at a time and displays visual representations of the intermediate datasets. Moreover, DBQA provides a more finely grained animation by supporting subqueries, which are usually particularly difficult

for learners to comprehend [22]. DBQA also allows learners to move forward or backward during a query evaluation. Because learners are usually confused regarding default error messages provided by a database, DBQA provides an error interpreter that translates a database's messages to more understandable messages. ADbC divides the building process of SQL queries into several steps (group by, having, distinct, order by, outer join, and function) in a graphic interface and provides animations for each step.

Animation pedagogies simplify the understanding of SQL by displaying the intermediate datasets of an SQL query one clause at a time. This can reduce the mental effort invested in visualising the execution process; however, learners must mentally integrate all the correlated intermediate datasets in working memory to understand the entire execution process of an SQL query because the animation is distributed over these steps. Furthermore, because of the limitations of graphical interfaces, animation pedagogies do not support advanced SQL queries (e.g. correlated subqueries), which are among the most difficult for students to learn [22].

B. GRAPHICAL QUERY BUILDER

Graphical query builders bridge the gap between database systems and learners by providing a graphical user interface that automatically generates SQL statements for data requests to assist learners in understanding SQL. Several graphical query builders have been proposed; these include SQL in Steps (SiS) [23] and Query by Example (QBE) [24].

SiS combines graphic query builders and automatic SQL translations to improve the manner through which learners understand SQL. The process of SQL query building is divided into several steps in the graphic interface, which guides learners to build an SQL query step-by-step until the query is completed. During the process, every change made in the interface triggers a change in the SQL translation, which automatically generates the corresponding SQL statement and refreshes the output dataset in its current state in the interface. In addition, SiS provides learners with a graphical representation of the database schema so that they need not seek to identify where the elements of data required are located and how they can be extracted. QBE, developed by Zloof [24], provides a visual approach for accessing data from databases through table skeletons. Learners express queries by inserting examples in these skeletons to generate the logic of the query. QBE builders have become common as a means of performing database queries, such as Microsoft (MS) SQL Server and Access. In MS Access, learners can create a query in the Query Design function by selecting the needed tables and entering the values and conditions of the columns in these tables. The SQL statement is then automatically generated using SQL View. However, when learners move to more advanced queries, graphical query builders may go beyond the capabilities of graphical interfaces [25]. For instance, SiS has limited support for subqueries, which can only be used in the 'from' clause of SQL statements [23].

Furthermore, users find it difficult to transform from graphical user interfaces to textual ones [6].

C. FEEDBACK

Feedback represents a key component in a learning loop [26], [27]. Feedback pedagogies parse learners' submitted SQL queries and compare them with the correct solution to determine their correctness and even provide intelligent instructions or guidelines for them to undertake. Tools that provide this ability, such as SQL-Tutor-Web (SQLT-Web) [28], AsseSQL [27], SQLator [10], SQLify [29], Learning Environment for Automatic Rating of Notions of SQL (LEARN-SQL) [30], SQL Lightweight Tutoring Module (SQL-LTM) [31], Automated Database Verification with Interactive Counter Example (ADVICE) [32], Acharya [33], and SQL-Trainer [34], have been developed. These tools provide different methods for verifying learners' written SQL queries and provide various degrees of feedback. An overview of these tools is provided in Table 1. These are analysed according to the following characteristics.

- *Execution process of SQL queries:* This tool provides information about the execution process of SQL queries.
- *Intelligent feedback:* This tool automatically provides meaningful hints and explanations for executed SQL queries to enable learners to correct errors and understand why their corrections were successful rather than leading them to a solution by telling them the answer.
- *Correctness checking:* This tool shows the correctness of learners' SQL statements.
- *Distance learning:* This tool provides a web-based interactive interface for learning SQL.
- *Database schema:* This tool shows learners the database schema used in SQL questions to reduce learners' cognitive load.
- *Learning status monitoring:* This tool collects the history of learners' previously solved SQL questions and then assigns the next practice question according to their learning status.

Feedback pedagogies assist learners in understanding SQL by supporting their own solution paths rather than forcing them to accept the ideal solution provided by the instructor. Table 1 shows that none of these pedagogies provide information about the execution process of SQL queries, indicating that conceptualisation and visualisation remain necessary.

The animation, graphical query builder, and feedback pedagogies have promoted the study of SQL learning. However, considering the inherent difficulties posed by the declarative nature of SQL, feedback pedagogies and graphic query builder pedagogies do not demonstrate the execution process of SQL queries. Although animation-based pedagogies provide this information, learners still must visualise the entire evolution process because the animation is distributed over the steps of the execution process of SQL queries. SQL learners still encounter mental imagery problems.

TABLE 1. Overview of feedback pedagogies and their characteristics.

System	Execution process of SQL queries	Intelligent feedback	Correctness checking	Distance learning	Database schema	Learning status monitoring
SQTL-Web	No	Yes	Yes	Yes	Yes	Yes
AsseSQL	No	No	Yes	Yes	Yes	Yes
SQLator	No	No	Yes	Yes	Yes	Yes
SQLify	No	Yes	Yes	Yes	Yes	No
LEARN-SQL	No	Yes	Yes	Yes	No	No
SQL-LTM	No	Yes	Yes	Yes	No	No
ADVICE	No	Yes	Yes	Yes	Yes	Yes
Acharya	No	Yes	Yes	Yes	Yes	Yes
SQL-Trainer	No	Yes	Yes	Yes	No	Yes

III. THEORY AND HYPOTHESIS

This section introduces cognitive load theory, a cognitive model for SQL query-writing, semantic network theory, concept maps, and the method of representing the cognitive model in concept maps.

A. COGNITIVE LOAD THEORY

Cognitive load theory, which provides instructional design guidelines, divides cognitive load imposed by learning materials into intrinsic, extraneous, and germane loads [35], [36]: Intrinsic load relates to the number of interacting information elements in a material [37]. High intrinsic load is imposed by materials with high element interactivity [38]. With an increase in the prior knowledge of the material, intrinsic load decreases. The main reason for this decrease is the reduction in the number of elements present in the learning materials, which occurs for learners with a high level of prior knowledge, who incorporate several information elements about materials into a cognitive schema, which can be considered a single element in learners' working memory. Extraneous load is caused by inappropriate instructional designs that require learners to engage in activities not relevant to schema acquisition [39]. For instance, learners' working memory resources are used for searching and organising the information necessary for learning. The material becomes difficult to understand if a high extraneous load is imposed by an instructional design [16], [40]. When extraneous load decreases, learners achieve a higher cognitive capacity, which can be invested in germane processing. Germane load is caused by processes contributing to schema construction and automation [41]. Thus, germane load is effective for learning. Germane load can be induced by instruction that stimulates learners to invest cognitive resources in learning-related activities [38]. Germane and extraneous loads are imposed by learning material design, whereas intrinsic load is inherent to the material. According to cognitive load theory, good instruction reduces extraneous load but increases germane load. In this study, extraneous and germane loads imposed by concept map-based and conventional instruction were measured and their effects on learning SQL were thus analysed.

B. COGNITIVE MODEL OF SQL QUERY-WRITING

Gould and Ascher [42] proposed a high-level process of SQL query-writing, which includes formulation, planning,

Data request:

Which members browsed products but have not purchased them yet?

SQL statement:

1. Select member_name from browse b, member m
2. where b.member_id = m.member_id
3. and product_id not in
4. (select product_id from purchase p, transaction t
5. where p.transaction_no = t.transaction_no
6. and t.member_id = m.member_id)

FIGURE 1. Data retrieval request and SQL statement for the request.

and coding stages. Learners first formulate a data retrieval request, then plan a strategy to solve the request, and finally implement this plan using an SQL statement. Ogden [43] further established a cognitive model to describe the cognitive process of learners who write SQL queries. The cognitive model comprises formulation, translation, and writing phases. In the formulation phase, learners decide on the data they require in the context of an application domain. Fig. 1 illustrates a data request example for the formulation phase, 'Which members browsed products but have not purchased them yet?' The entity-relationship (ER) diagram and relational database schema associated with the request are shown in Appendix A. In the translation phase, learners translate the example into a data access plan in terms of the constructs of the relational database schema using their own words. An output for the example is as follows: 'First, identify members who browsed products. Second, find out which products members purchased. Finally, check that each product browsed by a member is included in products purchased by the member. If any product browsed by a member is not included in what the member has purchased, the member is added into the resultant dataset.' In the writing phase, learners further convert this plan into an SQL statement in terms of the syntax of a specific SQL language (Fig. 1). This indicates that the problem of visualising the evolution process of the intermediate datasets of SQL queries is most related to the translation phase. This study focused on the translation phase and attempted to represent the intermediate datasets of a data access plan and their transformation into the SQL statement to alleviate the influence of the mental imagery problem.

C. REPRESENTING THE COGNITIVE MODEL IN CONCEPT MAPS

Semantic network theory proposes that semantic memory is structured as a network of nodes [44]. Concepts are independently stored as a node in semantic memory and connected by links through which the semantic relationships between them are defined [45]. When learners learn a semantic relation meaningfully, they must be able to create nodes that represent newly learned knowledge and establish links that connect the newly learned nodes and the already known nodes in their existing semantic memory. According to the cognitive model of SQL query-writing, learning an SQL statement for a data request is a semantic transformation from the formulation phase to translation phase and then to the writing phase. Therefore, from the perspective of semantic network theory [44], learning the transformation from a data request to an SQL statement can be regarded as a process of establishing the semantic relations from the formulation phase to the translation phase and then from the translation phase to the writing phase. In the conventional instruction, learners must visualise the semantic relations in working memory to understand SQL queries. A network representation is useful for understanding the semantic relations by showing the semantic relationship between the phases. Concept mapping is a learning strategy based on Ausubel–Novak–Gowin theory [17], which can represent the cognitive structure used by people to interpret a problem domain [46]. Through a diagrammatic network representation (with nodes = concepts and links = relationships between concepts), concept maps describe salient concepts and their structures. Through explicit link labelling, semantic interconcept relationship information can be obtained. Two or more concepts joined in this manner form a meaningful statement, which is called a proposition or a semantic unit [47]. A concept map is built piece by piece of interacting semantic units [46].

This study integrated the cognitive model of SQL query-writing with concept mapping techniques to represent the evolution of the intermediate datasets of SQL queries by using concept maps. In this study, a concept map consists of three segments—namely formulation, translation, and writing—that correspond to the three phases of the cognitive model. The use of concept maps is demonstrated using the data request of Fig. 1 as an example. The concept map in Fig. 2 represents a possible cognitive model in which learners translate the data request of Fig. 1 to the SQL statement of Fig. 1. The translation segment of the concept map, which is the focus of this study, has seven concepts and related links (four tables, two intermediate datasets, and one resultant dataset). The concept map explicitly shows what initial tables are required and how they are joined to generate the intermediate datasets and further evolve them into the resultant dataset. For example, the concept map shows that the data request requires four tables (browse, member, purchase, and transaction). Table browse and table member are

joined through column member_id to generate an intermediate dataset (members who browsed products).

Chen *et al.* [48] indicated the importance of assimilating new concepts into existing knowledge structures to achieve meaningful learning. Concept maps demonstrate the relationships between new and old knowledge and integrate them, thus promoting meaningful learning [46], [47], [49]. Erdogan [50] emphasised that concept map development and study can assist learners in correlating known concepts and experiences with a new subject, helping them acquire new knowledge on the subject. Therefore, building a concept map for learning an SQL query may be beneficial for learners to integrate new and old SQL knowledge, because when they are constructing a concept map, they must make decisions about how the new and old SQL knowledge can be applied together to accomplish the data request. This process could promote learner engagement and thus increase germane load. When a learner is studying a concept map for learning an SQL query, the concept map may aid the learner in focusing his/her attention on the key concepts of the SQL query because this map can explicitly depict the evolution of the intermediate datasets of the SQL query and thus reduce extraneous load. Furthermore, concept maps have been widely used in software engineering learning research, such as in research on programming languages [51], [52], data modelling [53], requirements analysis [54], and communication problems between analysts and users [55]. Hence, this study proposed integrating concept maps techniques into SQL learning.

Readers may consider why query trees are not used to represent the execution process of SQL statements because Fig. 2 has some similarities with a query tree. Query trees represent a tree data structure that corresponds to a relational algebra expression [56]. In a query tree, the input relations of a relational algebra expression are represented as leaf nodes and the relational algebra operations are represented as internal nodes. Query trees give a favourable visual representation and understanding of the query in terms of the relational operations it uses. However, in the context of learning SQL, representing SQL queries in query trees compels learners to develop data access plans in terms of the limited operations (such as project, Cartesian product, and division) of relational algebra. By contrast, concept maps are not as formal as query trees; thus, learners may find it easier to depict their cognitive structure by using their own words and understand concept maps depicted by the instructors. Gould and Ascher [42] indicated that for easily developing a data access plan to solve a data request, learners should use their own words in the plan.

According to cognitive load theory, the cognitive model for SQL query-writing, semantic network theory, and the essential features of concept maps, concept maps may be a worthwhile instructional approach for learning SQL. This study thus proposes the following hypothesis:

Hypothesis: Learners who receive concept map-based instruction acquire a clearer understanding of SQL than do learners who receive conventional instruction.

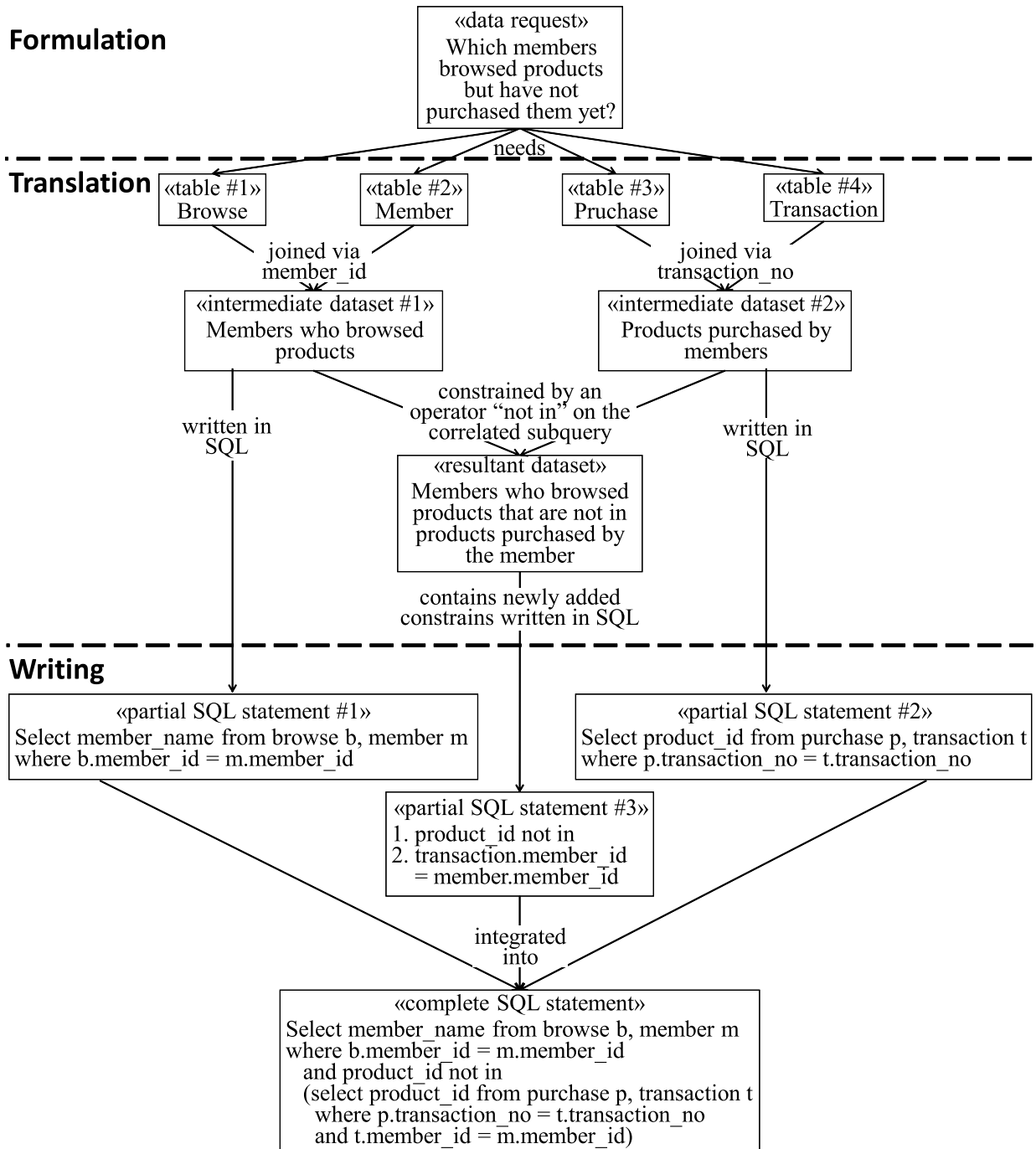


FIGURE 2. Concept Map for the Data Request in Fig. 1.

IV. RESEARCH METHODOLOGY

To examine the relationship between concept maps and understanding SQL, we evaluated the performance of two database courses, one with concept map-based instruction and the other with conventional instruction. Each course was conducted by the same instructor. The aim was that both courses cover completely identical database information at approximately the same pace. In total, 39 and 42 participants [i.e. undergraduate management information system (MIS) majors] were enrolled in the concept map-based

and conventional courses, respectively. This study analysed the participants' background and knowledge to ensure that the two treatment groups are comparable. The participants were students of the same grade in the same department of the same university. All participants had taken the same computer science-related courses, such as object-oriented programming. No significant difference was noted in their object-oriented programming scores between the conventional ($M = 81.31, SD = 10.743$) and the concept map-based ($M = 78.09, SD = 8.226$) courses ($t(81) = 1.507, p > 0.05$).

To further determine whether the participants had prior knowledge of SQL, before the courses began, they were asked to participate in an SQL query-writing test; the results showed that both groups had no SQL-related expertise. At the end of the two courses, the participants' SQL comprehension was measured and the differences between the two groups were compared to clarify the relationship between concept map-based instruction and SQL learning.

A. CONCEPT MAP-BASED VERSUS CONVENTIONAL INSTRUCTION

Both courses lasted 3 hours per week for 18 weeks, including the 2 weeks of the midterm and final exams. Database learning began with fundamental database concepts (1 week), followed by the ER model (2 weeks), the relational data model, and the transformation from the ER model to the relational data model (1.5 weeks). The students then received instruction on relational algebra (1 week), Oracle SQL (5 weeks), normalisation (1.5 weeks), data storage and indexing (1 week), query processing and optimisation (2 weeks), and transaction processing and concurrency control (1 week). The SQL teaching material included data manipulation language (DML), data definition language, and data control language. The DML material contained four types of SQL statements: select, insert, update, and delete. The students learned the four types of statements, but the study paid most attention to select statements because many of the concepts covered by select statements are directly relevant to other statement types [23], [28]. Select statements are typically the most relevant when learning SQL [8]. Concepts covered in the material of select statements included simple queries with one table, order by, build-in functions, arithmetic operators, simple subquery, exists, in, correlated subquery, inner join, outer join, self-join, group by, group by with having, aggregate functions, and set operators. Completely identical information was covered in both database courses, with the only difference being the instruction type used for learning SQL.

1) CONVENTIONAL INSTRUCTION

Students learn SQL queries from the verbal description of instructors, without the support of concept maps. Taking the data request in Fig. 1 as an example, the ER diagram and the relational database schema (shown as Appendix A) are first introduced by the instructor, who then verbally introduces the SQL query for the data request as follows.

To implement this data request, one possible plan is to check if products browsed by a member are not included in what the member purchased. When a product satisfies this condition, the member who browsed the product is classified as the resultant dataset. According to this plan, first, we must identify members who browsed products, then find out products purchased by members, and finally identify members who browsed products that are not included in products the member purchased. According to the ER diagram, information about members who browsed products is in the two entity types (member and product) and the relationship

type (browses). They can be transformed into three relations: member, product, and browse. The join attribute between member and browse is member_id. Thus, data about members who browsed products can be retrieved using the SQL statement 'select member_name from browse b, member m where b.member_id = m.member_id,' which is lines 1 and 2 of the complete SQL statement in Fig. 1. The information about products purchased by members is stored in the two entity types (transaction and product) and the relationship type (purchases). They can be transformed into three relations: transaction, purchase, and product. The join attribute between transaction and purchase is transaction_no. Thus, the data about products purchased by members can be retrieved using the SQL statement 'select product_id from purchase p, transaction t where t.transaction_no = p.transaction_no,' which is lines 4 and 5 of the complete SQL statement. Finally, we must identify members who browsed a product and have not purchased it. This is implemented by 3–6 lines of the complete SQL statement. A 'not in' operator (line 3) is applied on the correlated subquery (4–6 lines), which is evaluated once for each product browsed by members and uses the value of the member from the outer query to check that the product is not included in the dataset of products purchased by the member. If a product fits this condition, the member is classified as the resultant dataset.

2) CONCEPT MAP-BASED INSTRUCTION

The core of the concept map-based instruction is to ask learners to describe the query logic they use to perform a data request by using concept maps and then provide them with a concept map prepared by the instructor for the data request to verify what they understand and misunderstand. Before teaching SQL, the instructor teaches learners the concept mapping technique and how to use concept maps to represent their cognitive structure regarding the transformation from a data request to an SQL statement. When building a concept map for a data request, learners analyse the data request, the ER diagram, and the relational database schema to identify appropriate concepts (e.g. table, intermediate dataset) and links (e.g. joined via) according to the following steps until a potential data access plan is established: (1) Use the data request as the main concept of the concept map, which is presented in the formulation segment of the concept map. Taking the data request in Fig. 2 as an example, the main concept is "Which members browsed products but have not purchased them yet?" (2) Determine the required tables based on the data requirements of the data request and connect them to the main concept as the initial datasets. For example, the data request of Fig. 2 requires four tables (member, browse, transaction, and purchase), and these are connected to the main concept as the initial datasets and presented in the translation segment of the concept map. (3) Determine the logic of evolving the initial datasets into the intermediate datasets and then into the resultant dataset. The evolution process is presented in the translation segment of the concept map. For example, the two initial tables, member and browse, in Fig. 2 are joined through column member_id to generate

the intermediate dataset #1, members who browsed products. (4) Create partial SQL statements for the concepts in the translation segment and integrate them into a complete SQL statement for the data request. These SQL statements are presented in the writing segment of the concept map. For example, the intermediate dataset #1, members who browsed products, in Fig. 2 is written in SQL statements: select member_name from browse b, member m where b.member_id = m.member_id. These partial SQL statements in Fig. 2 are integrated into a complete SQL statement for the data request. The concept map-based instruction assists learners in understanding SQL by encouraging them to begin with the main concept and expand outward to more in-detail involvement process from initial datasets to the resultant dataset. After the learners build a concept map for the data request, the instructor guides them to study a concept map prepared by the instructor for the query to verify what they understand and misunderstand. The instructor's concept map is only one of the possible concept maps for the data request because a data request typically has various possible query logics. Appendix B presents an example of the concept map drawn by a student for the data request in Fig. 1. However, the concept map provides learners with an insight into the instructor's thought process regarding the data request. Furthermore, when a learner's concept map for a data request cannot retrieve the correct data, the learner's concept map enables the instructor to understand how well the learner understands the data request being taught.

B. MEASUREMENT

Problem-solving performance, mental efficiency, response latency, and recall accuracy are the metrics typically used to measure understanding. Problem-solving performance relates to the ability to use knowledge gained from materials to resolve related problems in new situations [39]. Mental efficiency represents a cognitive schema's efficiency that is acquired, elaborated on, or automated in semantic memory when a problem is being resolved [38]. Response latency indicates the time that is required to retrieve information from semantic memory [57]. Recall accuracy measures the proportion of the total information that has been recalled correctly [58]. Response latency and recall accuracy are used to measure surface-level understanding; by contrast, to measure deep-level understanding, problem-solving performance and mental efficiency are used [38], [59]. By observing students' learning process, it was revealed that most students could clearly comprehend the individual syntaxes of SQL; however, when asked to apply these concepts together, they were overwhelmed by their complexity. Many studies have indicated that the syntax of SQL appears to be simple but counterintuitive [10], [12]. This implies that learning SQL requires an in-depth understanding of complicated semantic transformations from a data request to an SQL statement, rather than a superficial understanding of SQL syntax. Therefore, in this study, problem-solving performance and mental efficiency were used to test the current hypothesis.

Problem-solving performance concerns the ability to use knowledge gained from materials to solve related problems in new situations [60]. The measure of problem-solving performance is based on the performance of tasks being completed to obtain information on understanding [61]. In this study, problem-solving performance was defined as the score of correctly answered SQL query-writing tasks. This measure has been widely used in research on SQL learning [1], [62]. At the end of both courses, an SQL query-writing test was conducted to measure the problem-solving performance. This study scored problem-solving performance by using Reiser's [63] grading method, which is commonly used in related research on SQL learning [4], [7]. The participants' solutions for each SQL query-writing question were scored into one of two categories: essentially correct or incorrect. Solutions were considered essentially correct if they were either completely correct or had only minor errors. Complete correctness means that a solution can retrieve correct data. Minor errors are small errors that could be easily discovered by the participant or corrected automatically by an intelligent system (e.g. a spelling corrector). Examples of minor errors include misspelled column names, omitted or extra quotation marks, and misspelled data values. Solutions that led to incorrect data retrieval were scored as incorrect. One MIS professor and one database professional were recruited to score the participants' solutions according to the aforementioned criterion. The two graders integrated their scoring results after discussion and review.

Mental efficiency was measured using a computational approach widely used in related education research. This approach was developed by Paas and van Merriënboer [64], and it is based on problem-solving performance and the mental effort that is invested in achieving the problem-solving performance. To calculate relative mental efficiency scores, the scores of problem-solving performance and mental effort were transformed into standardised z-scores based on the grand mean across instructions obtained using equation (1). Positive and negative efficiency scores represent efficient and inefficient learning, respectively. Moreover, efficiency scores are positive when problem-solving performance exceeds invested mental effort. Mental effort, indicating the cognitive load invested to satisfy a task's demands, was measured on the subjective rating scale of Paas and van Merriënboer [64] (range: 1–7, with 1 being extremely low mental effort and 7 being extremely high mental effort). According to Gopher and Braune [65], people assign numerical values to their mental effort without difficulty. Moreover, subjective measures of task difficulty strongly correlate with objective measures [66]. Paas *et al.* [67] considered subjective measures of mental effort to be valid, reliable, and sensitive to relatively small cognitive load differences.

Mental Efficiency Score

$$= \frac{Z_{\text{Problem-solving Performance}} - Z_{\text{Mental Effort}}}{\sqrt{2}} \quad (1)$$

Mental effort was measured in the learning and testing phases of the courses. In the learning phase, this study measured the mental effort of participants receiving conventional instruction shortly after they learned SQL queries. By contrast, for the mental effort of participants receiving concept map-based instruction, mental effort measurement was conducted shortly after the participants constructed concept maps for learning SQL queries and after they studied the concept maps prepared by the instructor. In the test phase, the mental effort invested in the SQL query-writing test was measured after the test was completed. This study used an independent sample *t* test to determine possible statistically significant differences in problem-solving performance, mental effort, and mental efficiency between the two instruction approaches.

C. TASK MATERIAL

An SQL query-writing task is presented in Appendix A, including the ER diagram, relational database schema, and SQL questions. Most studies have divided the complexity of SQL questions into two or three levels [7], [62], [63]. The current study used a three-level distinction: easy, medium, and difficult. An easy query was defined as one covering a simple query with one table, arithmetic operations, built-in functions, selection, projection, and/or, and chaining. A medium query was defined as one covering join, group by, nesting, and set functions. A difficult query was defined as one covering more tables, more join operations, more nested operations, and combinations of elements used in any of the levels. The participants of the two groups were requested to answer the same 18 SQL questions (six for each level) using paper and pencil in 3 hours at the end of the courses. Two MIS professors and two senior database professionals reviewed the task material and recommended some changes. Then, 10 undergraduate majors in MIS who had completed a database course were recruited to conduct a pilot test. These undergraduate majors were required to answer these questions, and the results indicated that the task material was presented accurately.

TABLE 2. Problem-solving performance and mental effort scores: mean (standard deviation) and independent sample *t* test (*p*).

Dependent Variable	Conventional Instruction (n=42)	Concept Map-based Instruction (n=39)	Independent Sample <i>t</i> Test (<i>p</i>)
Problem-solving Performance	12.19 (2.308)	14.33 (2.030)	-4.422*** (0.000)
Mental effort			
• Test phase	5.19 (1.273)	4.051 (1.276)	4.019*** (0.000)
• Learning phase	5.119 (1.087)	Constructing maps 5.923 (1.01)	Studying maps 3.615 (1.115)
		Constructing maps -3.442*** (0.001)	Studying maps 6.146*** (0.000)

****p* < .001

V. DATA ANALYSIS AND RESULTS

Table 2 presents the means (standard deviations) of problem-solving performance and mental effort, along with

independent sample *t* test results. Compared with those in the conventional course, the participants in the concept map-based course demonstrated significantly higher problem-solving performance. Regarding mental effort, in the test phase, compared with those in the conventional course, the mental effort invested by the participants in the concept map-based course for the SQL query-writing task was significantly lower. Moreover, in the learning phase, the participants in concept map-based course invested significantly more mental effort in concept map construction but significant less mental effort in studying the instructor's concept maps than those in the conventional course.

To calculate the relative mental efficiency scores, the scores of problem-solving performance and mental effort in the test phase were transformed into standardised z-scores by using the grand mean across the two instruction types. The data presented in Table 3 demonstrates that participants who received concept map-based instruction achieved significantly higher mental efficiency in the test phase than did those who received the conventional instruction, implying that concept map-based instruction is superior to conventional instruction in terms of facilitating SQL understanding.

TABLE 3. Relative Mental Efficiency in the Test Phase: Mean (Standard Deviation) and Independent Sample *t* Test (*p*).

Conventional Instruction (n=42)	Concept Map-based Instruction (n=39)	Independent Sample <i>t</i> Test (<i>p</i>)
-0.58 (1.303)	0.625 (1.231)	-4.274*** (0.000)

****p* < .001

VI. DISCUSSION AND IMPLICATIONS

The current data analysis results support the hypothesis that concept map-based instruction facilitates SQL understanding superior to that facilitated by conventional instruction. Table 3 indicates that concept map-based course participants exhibited relatively high mental efficiency, implying that concept map-based instruction may improve learners' comprehension of SQL attained using efficient schema acquisition and automation.

This study analysed the possible effects of concept map-based instruction on learning SQL from a cognitive load theory perspective. Mental effort represents overall cognitive load. According to cognitive load theory, variations among different mental effort constituents can be derived if intrinsic load is held constant and extraneous and germane loads are examined in relation to problem-solving performance scores [38], [68]. When learners have the same level of prior knowledge in a material, their intrinsic load imposed by the material is the same [38]. In the current study, for both courses, the pretest results indicated that the participants had identical prior SQL knowledge levels, implying that they had equal intrinsic loads. Therefore, variations in problem-solving performance scores were associated with extraneous

and germane loads alone. In the subsequent sections, analyses of the effects of the two instruction types on SQL comprehension in terms of germane and extraneous loads are discussed.

A. GERMANE LOAD

Table 2 demonstrates that the mental effort expended in constructing concept maps for learning SQL queries was significantly higher in the concept map-based course participants than in the conventional course participants in learning SQL queries through verbal description. According to cognitive load theory, at a constant intrinsic load, the increased mental effort was derived from the germane load because increasing extraneous load reduced problem-solving performance. Moreover, Table 2 indicates that concept map-based course participants achieved relatively high problem-solving performance scores. Thus, the increased mental effort was not derived from the extraneous load. In other words, concept map-based instruction induced a significantly higher germane load when concept maps were being constructed than did conventional instruction. Moreover, germane load helps learners learn. The influence of the concept map-based instruction on the learners' germane load are analysed as follows.

Numerous researchers have recognised concept maps as being an effective tool for externalising learners' knowledge structures [69]–[71]. The network representations of concept maps can aid in presenting the relationships between concepts [45]. Learning by externalising knowledge structure can engage learners in organising their knowledge and learning experiences because learners have to actively seek information for describing concepts and relationships that link the concepts [72]. This process can help learners identify misconceptions or contradictions in the knowledge structure [73]. Cognitive load theory indicated that instructions that engage learners in learning-relevant activities can induce germane load [38]. Therefore, learning through concept map construction engenders an increase in germane load because when constructing concept maps, learners are compelled to make decisions on how these concepts can work together. In the context of concept map-based instruction, concept maps can act as a cognitive tool externally representing the knowledge structure used by learners to solve SQL queries. When constructing a concept map for learning an SQL query, learners must identify useful SQL concepts and determine how they can be applied together to generate a possible data access plan. For instance, to construct a concept map for the data retrieval request in Fig. 1, learners must identify the required SQL concepts, such as 'join', 'not', 'in', 'subquery', and 'correlated subquery', and then determine their relationships to accomplish the data request. This process engages learners in SQL learning-relevant activities and thus increases germane load, eventually enhancing learners' understanding of SQL. This is consistent with previous study results that learning by creating a concept map of knowledge structure is considerably more useful than rote memorisation [49]; this is also supported by the current study results that the participants

receiving the concept map-based instruction exhibited higher germane load and mental efficiency.

Furthermore, concept map construction for learning SQL possibly promotes meaningful learning, which involves assimilating new concepts into existing knowledge structures [48]. Novak and Cañas [46] indicated that instructional strategies that emphasise the relation of new knowledge to learner's existing knowledge foster meaningful learning. Studies have found that the use of concept mapping strategies can facilitate meaningful learning [48], [75], [74]. Accordingly, concept map-based instruction may enhance meaningful learning because constructing a concept map for learning an SQL query stimulates the learners to search their existing knowledge of SQL and integrate it with the newly learned SQL concepts. For instance, a learner who has learned 'join' wants to learn 'correlated subquery' through the data retrieval request in Fig. 1. If the learner draws Fig. 2, it relates the newly learned concept (correlated subquery) with already known concept (join) to represent the possible data access plan in a diagrammatic format. Specifically, the known concept 'join' is used in generating the two intermediate datasets (members who browsed products and products purchased by members). The newly learned concept 'correlated subquery' is applied on the two intermediate datasets to generate the resultant dataset (members who browsed products that are not the products purchased by the members). The process of constructing a concept map may compel learners to reflect on the relationships between the newly learned concept and their existing knowledge about SQL in a meaningful manner. This is consistent with a previous result: concept maps facilitate learning because they can explicitly integrate new and old knowledge and thus help diagnose misunderstandings and communicate complex concepts [48], [50]. Concept maps aid in triggering memory and focusing attention on the relationships between newly learned concepts and known concepts [73]. Marra and Jonassen [76] indicated that concept map construction can engage learners in analysing their existing task-related knowledge structures and then in relating these structures to the content being learned. Anderson [77] found that students learn better from discovery than from direct instruction, and such knowledge is retained for longer than when students learn by being told. Hence, concept map-based instruction may help learners assimilate newly learned SQL concepts into existing knowledge structures and form so-called meaningful learning [48].

B. EXTRANEIOUS LOAD

In the concept map-based course, after learners built the concept maps, they studied the instructor's concept maps to review their understanding. Table 2 demonstrates that the conventional course participants expended significantly higher mental effort in learning SQL queries than did their concept map-based course counterparts in studying the concept maps of SQL queries. According to cognitive load theory, with a constant intrinsic load, the increased mental effort was derived from the extraneous load because

increasing germane load contributed to problem-solving performance. Moreover, Table 2 indicates that conventional course participants achieved relatively low problem-solving performance scores. Thus, the increased mental effort was not derived from the germane load. In other words, conventional instruction induced a significantly higher extraneous load in learning SQL queries than concept map-based instruction in the study of the concept maps of SQL queries. Moreover, extraneous load hampers learning. The following section analyses the influence of extraneous load on the learners of the two courses.

The learners in the conventional course learned SQL queries from the instructor's verbal description. The learning process involves sequential processing and assimilation because in verbal representations, information is organised sequentially [13]. Larkin and Simon [78] indicated that sequentially indexing information leads to further extraneous load for keeping the information in working memory. Learning SQL queries typically requires considering multiple intermediate datasets and their evolution simultaneously. Therefore, learners receiving conventional instruction may need to invest additional extraneous load to keep information in working memory until all information required is received. Taking the data retrieval request in Fig. 1 as an example, learners follow the verbal description described in Section 4.A to identify members who browsed products. They bring the information on relation browse and relation member into working memory, mentally perform a join operation on the two relations to generate an intermediate dataset (members who browsed products), and expend mental effort to keep the partial plan in working memory. They then follow the verbal description to generate another intermediate dataset (products purchased by members), and keep it in working memory. Finally, they mentally apply an operator 'not in' on the two intermediate datasets by using 'correlated subquery' to generate the resultant dataset. This process shows that relevant information that has to be integrated so that the SQL query is understood is presented in a sequential manner. The learners are required to invest mental effort so that information is retained in their working memory and then to wait for more information. Their attention is moved along with sequential processing. Moreover, learners' distraction related to various information sources causes excessive extraneous load [39]. An instructional procedure, which requires learners to organise information necessary for learning, imposes a high extraneous load; the reason for this is that working memory resources are invested in activities not related to schema acquisition [19], [39], [79]. This suggests that sequential reasoning is not applicable to SQL learning; thus, conventional instruction may inhibit learners' SQL comprehension.

Furthermore, extraneous load may explain why students can understand individual concepts of SQL but have difficulty applying these concepts together. When learners attempt to express a data retrieval request in an SQL query, they must search for useful SQL concepts in their existing knowledge of SQL, bring them from semantic memory into

working memory, and determine how they can be applied together to build a possible data access plan. In this context, the students must retrieve all possible SQL concepts into working memory and evaluate them mentally before finally selecting the appropriate concepts to perform data requests. The entire process involves extensive working memory-semantic memory interaction, and the need to retain the related information in working memory is continual. When the resultant total cognitive load induced is excessive, learners are overwhelmed by information and encounter difficulty.

By contrast, in the concept map-based course, after constructing a concept map for learning an SQL query, the learners studied the instructor's concept map. Paivio and Csapo [80] indicated that diagrammatic and verbal representations are most effectively used in parallel and sequential processing, respectively. Learning an SQL query usually requires simultaneously considering multiple intermediate datasets and their evolution. Concept maps can diagrammatically present all associated information together, which can eliminate for the need to keep track of related elements [78]. Therefore, concept map-based instruction may reduce the learners' extraneous load. The concept map in Fig. 2 shows that the SQL query required four tables to generate two intermediate datasets and an operator 'not in' was applied on the two intermediate datasets by using 'correlated subquery' to generate the resultant dataset. The concept map simultaneously and explicitly displays the initial tables, the intermediate datasets, and their evolution, through which learners may more easily perceive the evolution process of the intermediate datasets of the SQL query with lower extraneous load, because the associated information is presented together. This is consistent with the results of Marcus *et al.* [16]: diagrammatic representation increased learning effectiveness by reducing extraneous load because pieces of information were presented together physically. Concept maps aid in highlighting key factors for learners [73]. In this context, concept map-based course learners may expend less extraneous load in holding relevant information in working memory for visualising the evolution process. Furthermore, extraneous load that is freed can be invested into germane load, which can consequently enhance SQL learning. This is supported by data analysis results in Table 2 that concept map-based instruction expended considerably lower mental effort in studying concept maps than did conventional instruction. According to cognitive load theory, this reduction in mental effort was derived from extraneous load as mentioned before. Moreover, some of the freed extraneous load was invested in germane load, which can be demonstrated by concept map-based course participants achieving increased problem-solving performance.

Furthermore, the similarity between the concept maps of the instructor and learners can be used to assess the learners' understanding of SQL [48]. When studying the instructor's concept map for learning SQL queries, learners follow the cognitive structure used by the instructor in performing the data retrieval request; this can assist learners in

the identification of faulty reasoning or inappropriate representations. If these misunderstandings are corrected, further understanding is attained [81]. According to cognitive load theory, the instructor's concept map can direct learners' cognitive processes toward the relevant constructs and lead them to obtain the relevant information being brought into working memory with less extraneous load to gain SQL-related knowledge. The reduced extraneous load contributes to enhanced learning outcomes [16], [40].

VII. THREATS TO VALIDITY

In this section, threats to validity that could affect the study results are discussed; these threats are analysed according to the four types proposed by Wohlin *et al.* [82].

A. CONCLUSION VALIDITY

This validity is concerned with the statistical relationship between the treatment and the outcome of an experiment [82]. In this study, threats to conclusion validity identified and addressed are reliability of measures, random heterogeneity of subjects, and sample size. To address the reliability of measures, this study adopted well-documented measures to assess the understanding of SQL. Problem-solving performance was assessed using query accuracy, which has been considered as an effectiveness measure of SQL learning performance [1], [62]. A subjective rating scale was used to assess mental effort. Studies have concluded that this scale demonstrates a strong correlation with objective measures and sensitivity to relatively small cognitive load differences [65]–[67]. The issue of random heterogeneity of subjects was considered while recruiting participants in the experiment. Before the courses began, all the participants were asked to participate in an SQL query-writing test, and the results demonstrated that they had equal levels of prior SQL knowledge. That is, they had approximately similar SQL knowledge and experience, thus reducing the heterogeneity. With regard to sample size, although participants in the experiment were limited in number, the sample size was sufficient to verify the hypothesis through independent sample *t* tests and obtain conclusion validity.

B. INTERNAL VALIDITY

This validity relates to the ability of a study to establish a causal link between the treatment and the outcome within a given environment [82]. Two threats to internal validity were addressed in this study: instrumentation and setting. With regard to instrumentation, both the concept map-based and conventional courses were taught by the same instructor in such a way that both courses covered exactly the same information at nearly the same pace, with the only difference being the instruction type used for learning SQL; this difference ensured that learners' SQL comprehension was related only to instruction. With regard to the issue of setting, the participants of the two groups were requested to answer the same SQL questions under the same conditions and all followed the same procedure.

C. CONSTRUCT VALIDITY

This validity is concerned with the extent to which an experiment setting reflects the construct under study [82]. To ensure that the measures provide an accurate representation of the effect construct, this study measured learners' understanding of SQL by using mental efficiency and problem-solving performance instead of response latency and recall accuracy because learning SQL requires a deep-level understanding of complicated semantic transformation from a data request to an SQL statement. Mental efficiency and problem-solving performance have been analysed extensively in related studies [38], [39]. Three other threats to construct validity identified and addressed are confounding constructs with levels of constructs, interaction of different treatments, and experimenter expectancies. To avoid the problem of having confounding constructs with levels of constructs, this study used a three-level distinction (i.e. easy, medium, and difficult) to represent the complexity of SQL questions. A key concern related to the interaction of different treatments is ensuring that the participants of a study do not participate in other studies because treatments from different studies may interact. In this condition, the researcher cannot conclude whether the effect is due to one treatment or a combination of treatments. In this study, the participants did not participate in other studies; thus, there was no possibility of interactions among other treatments. Regarding experimenter expectancies, the participants were unaware of the experimental hypotheses.

D. EXTERNAL VALIDITY

This validity is concerned with the extent to which the result of a study is generalised outside the scope of the study [82]. The main concern related to interaction of selection and treatment is ensuring that the participants in a study are representative of the population to which this study seeks to generalise. This study aimed to evaluate the effects of concept map-based instruction on learners' SQL comprehension. MIS majors were recruited in this study to examine the research question. Considering manipulating and retrieving information from a relational database is a core competence of MIS majors, the participants were considered to be suitable representatives of SQL learners.

VIII. CONCLUSION

The inherent mental imagery problem caused by the declarative nature of SQL burdens learners' cognitive load and consequently jeopardises learning outcomes. In this study, concept mapping techniques were integrated into conventional database instruction to represent the evolution process of the intermediate datasets of SQL queries in concept maps. The study results show the superiority of concept map-based instruction over conventional instruction. From the cognitive load theory perspective, the influence of concept map-based instruction was explored. In the

ER diagram

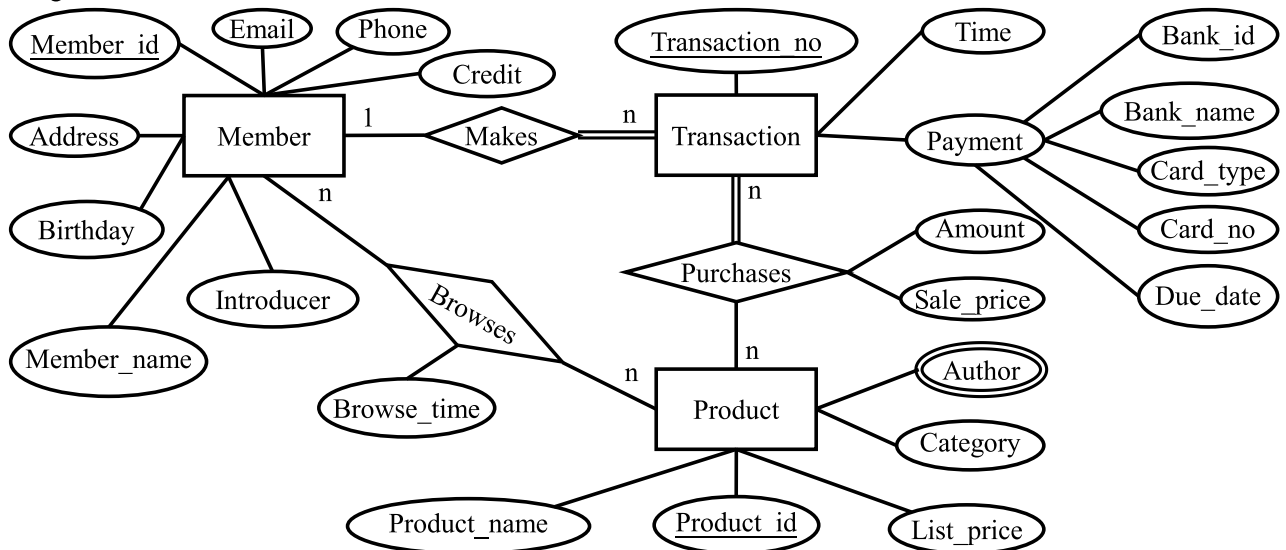


FIGURE 3. ER diagram.

learning phase, compared with those receiving conventional instruction, learners receiving concept map-based instruction expended higher germane load in concept map construction and lower extraneous load in studying the instructor's concept maps. In the test phase, compared with those receiving conventional instruction, learners receiving concept map-based instruction achieved higher problem-solving performance and they spent lower mental effort in achieving that performance level.

Four main reasons may explain the advantages of concept map-based instruction in SQL learning: (1) Concept map construction facilitated learner engagement, which increased learners' germane load, ultimately enhancing their understanding of SQL. (2) Concept map construction promoted meaningful learning because in the process, learners analysed their existing knowledge structures about SQL and related them to what they were learning; this helped learners assimilate newly learned SQL concepts into existing knowledge structures. (3) Studying instructors' concept maps helped learners follow the cognitive structures used by instructors to perform SQL queries, assisting them in clarifying any misunderstanding and thus aiding them in understanding SQL. (4) Studying the instructors' concept maps enabled learners perceive the execution process of SQL queries relatively easily because these concept maps presented relevant information that must be mentally integrated for understanding SQL queries.

In this study, an empirical insight into the effects of concept map-based instruction on learning SQL was obtained. However, additional follow-up studies are required to elucidate the factors limiting SQL learning from the perspectives of related theories, such as semantic distance between a data retrieval request and the corresponding SQL statement. These studies may provide further information for enhancing SQL learning. A broader perspective on SQL learning may

aid researchers in developing relatively more effective SQL learning instructions.

APPENDIX A QUERY-WRITING QUESTIONS

The query-writing questions performed by the participants at the end of courses are on an e-bookstore system. The members of the system can browse products and engage in transactions to buy products. The following figure contains the ER diagram, relational schemas, and SQL writing questions provided to the participants.

ER diagram: see Figure 3.

Relational Schemas:

Member (member_id, member_name, birthday, phone, address, email, introducer, credit)

Product (product_id, product_name, list_price, category)

Transaction (transaction_no, member_id, time, bank_id, bank_name, card_type, card_no, due_date)

Author (product_id, author_name)

Browse (member_id, product_id, browse_time)

Purchase (transaction_no, product_id, amount, sale_price)

SQL Writing Questions:

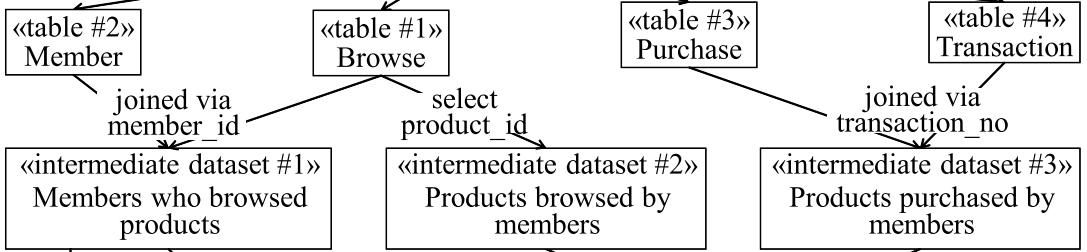
Easy:

1. Find the names of members whose addresses are in Texas.
2. Find the identities of members who have ever been an introducer.
3. Find the names of members who have the same birthday as the member called Tony.
4. Find the names of products whose list price is higher than that of product b30999.
5. For the Christmas season, each member will have their credit limit doubled. Find the identities of these members and the new credit limits.

Formulation

«data request»
Which members browsed products but have not purchased them yet?

Translation



written in SQL

constrained by "exists" on the correlated subquery (intermediate dataset #4)

«resultant dataset»
Members who browsed products and the following exists: the products they browsed are not empty after subtracting the purchased products.

contains newly added constraints written in SQL

written in SQL

Writing

«partial SQL statement #1»
select member_name from browse b, member m
where b.member_id = m.member_id

«partial SQL statement #2»
(select product_id from browse)
minus
(select product_id from transaction t, purchase p
where t.transaction_no=p.transaction_no)

«partial SQL statement #3»
1. exists
2. member_id=b.member_id

integrated into

«complete SQL statement»
select member_name from browse b, member m
where b.member_id = m.member_id
and exists (
 (select product_id from browse where member_id=b.member_id)
 minus
 (select product_id from transaction t, purchase p
 where t.transaction_no=p.transaction_no and member_id=b.member_id))

FIGURE 4. A concept map example drawn by a student for the data request in Fig. 1

6. Find the number of products that have the highest sale price.

Medium:

1. Find the names and phone numbers of members who have not browsed any products.

2. Find the names of members who browsed product b30999.

3. Find the names of products that were either ever bought or ever browsed.

4. Find the identification numbers of all transactions and the corresponding transaction amounts.

5. Find the identities and names of all members and their introducers' names.

6. Find the name of member b0905555 and the names of members who were introduced by member b0905555.

Difficult:

1. Find how many distinct members have bought product b40555.

2. Find the introducer names of members who bought product b30999.

3. Find the names of products that have been browsed more than two times.

4. Find the names of members whose transaction amounts are more than two times the average transaction amount.

5. Find the average credit limit of members who have bought product b40555.

6. Find the identities of members who have bought all products they browsed and the names of the products.

APPENDIX B

See Figure 4.

REFERENCES

- [1] H. C. Chan, K. K. Wei, and K. L. Siau, "User-database interface: The effect of abstraction levels on query performance," *MIS Quart.*, vol. 17, no. 4, pp. 441–464, Dec. 1993.
- [2] D. Lavbi, T. Matek, and A. Zrnc, "Recommender system for learning SQL using hints," *Interact. Learn. Environ.*, vol. 25, no. 8, pp. 1048–1064, Nov. 2017.
- [3] W. S. Luk and S. Kloster, "ELFS: English language from SQL," *ACM Trans. Database Syst.*, vol. 11, no. 4, pp. 447–472, Dec. 1986.
- [4] A. Mitrovic, "An intelligent SQL tutor on the Web," *Int. J. Artif. Intell. Educ.*, vol. 13, nos. 2–4, pp. 173–197, Apr. 2003.
- [5] J. R. Prior and R. F. Lister, "The backwash effect on SQL skills grading," *ACM SIGCSE Bull.*, vol. 36, no. 3, pp. 32–36, Jun. 2004.
- [6] K. Renaud and J. Van Biljon, "Teaching SQL—Which pedagogical horse for this course?" in *Key Technologies for Data Management (Lecture Notes in Computer Science)*, vol. 3112. Berlin, Germany: Springer, 2004, pp. 244–256.
- [7] C. Welty and D. W. Stemple, "Human factors comparison of a procedural and a nonprocedural query language," *ACM Trans. Database Syst.*, vol. 6, no. 4, pp. 626–649, Dec. 1981.
- [8] M. Cembalo, A. De Santis, and U. F. Petrillo, "SAVI: A new system for advanced SQL visualization," in *Proc. Conf. Inf. Technol. Edu. (SIGITE)*, New York, NY, USA, 2011, pp. 165–170.
- [9] R. Hardt and E. Gutzmer, "Database query analyzer (DBQA): A data-oriented SQL clause visualization tool," in *Proc. 18th Annu. Conf. Inf. Technol. Edu. (SIGITE)*, New York, NY, USA, 2017, pp. 147–152.
- [10] S. Sadiq, M. Orlowska, W. Sadiq, and J. Lin, "SQLator: An online SQL learning workbook," *ACM SIGCSE Bull.*, vol. 36, no. 3, pp. 223–227, Jun. 2004.
- [11] R. Kearns, S. Shead, and A. Fekete, "A teaching system for SQL," in *Proc. ACSE*, 1997, pp. 224–231.
- [12] A. Mitrovic, "Learning SQL with a computerized tutor," *ACM SIGCSE Bull.*, vol. 30, no. 1, pp. 307–311, Mar. 1998.
- [13] B. Kolloffel, T. H. S. Eysink, T. de Jong, and P. Wilhelm, "The effects of representational format on learning combinatorics from an interactive computer simulation," *Instructional Sci.*, vol. 37, no. 6, pp. 503–517, Nov. 2009.
- [14] D. P. Ausubel, J. D. Novak, and H. Hanesian, *Educational Psychology: A Cognitive View*, 2nd ed. New York, NY, USA: Holt, Rinehart & Winston, 1968.
- [15] G.-J. Hwang, P.-H. Wu, and H.-R. Ke, "An interactive concept map approach to supporting mobile learning activities for natural science courses," *Comput. Edu.*, vol. 57, no. 4, pp. 2272–2280, Dec. 2011.
- [16] N. Marcus, M. Cooper, and J. Sweller, "Understanding instructions," *J. Educ. Psychol.*, vol. 88, no. 1, pp. 49–63, Mar. 1996.
- [17] J. D. Novak, and D. B. Gowin, *Learning How to Learn*. Cambridge, U.K.: Cambridge Univ. Press, 1984.
- [18] P. Ayres and J. Sweller, "The split-attention principle in multimedia learning," in *The Cambridge Handbook of Multimedia Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2005, pp. 135–146.
- [19] P. Chandler and J. Sweller, "Cognitive load theory and the format of instruction," *Cognition Instruct.*, vol. 8, no. 4, pp. 293–332, Dec. 1991.
- [20] R. B. Clariana, T. Engelmann, and W. Yu, "Using centrality of concept maps as a measure of problem space states in computer-supported collaborative problem solving," *Educ. Technol. Res. Develop.*, vol. 61, no. 3, pp. 423–442, Apr. 2013.
- [21] M. Murray and M. Guimaraes, "Animated database courseware: Using animations to extend conceptual understanding of database concepts," *J. Comput. Sci. Colleges*, vol. 24, no. 2, pp. 144–150, Dec. 2008.
- [22] A. Ahadi, J. Prior, V. Behbood, and R. Lister, "A quantitative study of the relative difficulty for novices of writing seven different types of SQL queries," in *Proc. ITiCS*, New York, NY, USA, 2015, pp. 201–206.
- [23] P. Garner and J. A. Mariani, "Learning SQL in steps," *J. Syst. Cybern. Informat.*, vol. 13, no. 4, pp. 19–24, 2015.
- [24] M. M. Zloof, "Query-by-example: A data base language," *IBM Syst. J.*, vol. 16, no. 4, pp. 324–343, Dec. 1977.
- [25] H.-J. Kim, H. F. Korth, and A. Silberschatz, "PICASSO: A graphical query language," *Softw., Pract. Exper.*, vol. 18, no. 3, pp. 169–203, Mar. 1988.
- [26] S. I. Mehta and N. W. Schlecht, "Computerized assessment technique for large classes," *J. Eng. Edu.*, vol. 87, no. 2, pp. 167–172, Apr. 1998.
- [27] J. R. Prior, "AsseSQL: An online, browser-based SQL skills assessment tool," in *Proc. Conf. Innov. Technol. Comput. Sci. Edu. (ITiCSE)*, New York, NY, USA, 2014, p. 327.
- [28] A. Mitrovic, "A knowledge-based teaching system for SQL," in *Proc. ED-MEDIA*, Charlottesville, VA, USA, 1998, pp. 1032–1037.
- [29] S. Dekeyser, M. de Raadt, and T. Y. Lee, "Computer assisted assessment of SQL query skills," in *Proc. 18th Conf. Australas. Database*, Darlinghurst, NSW, Australia, 2007, pp. 53–62.
- [30] Abelló, M. E. Rodríguez, T. Urpí, X. Bugués, M. J. Casany, C. Martín, and C. Quer, "LEARN-SQL: Automatic assessment of SQL based on IMS QTI specification," in *Proc. ICALT*, P. Díaz, Kinshuk, I. Aedo, and E. Mora, Eds. Los Alamitos, CA, USA, 2008, pp. 592–593.
- [31] R. Dollinger, "SQL lightweight tutoring module—semantic analysis of SQL queries based on XML representation and LINQ," in *Proc. ED-MEDIA*, Waynesville, NC, USA, 2010, pp. 3323–3328.
- [32] M. Cvetanovic, Z. Radivojevic, V. Blagojevic, and M. Bojovic, "ADVISE—Educational system for teaching database courses," *IEEE Trans. Educ.*, vol. 54, no. 3, pp. 398–409, Aug. 2011.
- [33] S. Bhagat, L. Bhagat, J. Kavalan, and M. Sasikumar, "Acharya: An intelligent tutoring environment for learning SQL," in *Proc. Vidyakash Int. Conf. Online Learn.*, Mumbai, India, 2002, pp. 67–77.
- [34] H. Laine, "SQL-trainer," in *Proc. 1st Annu. Finnish/Baltic Sea Conf. Comput. Sci. Educ.*, Koli, Finland, 2001, pp. 13–17.
- [35] J. Sweller, "Cognitive load theory, learning difficulty, and instructional design," *Learn. Instruct.*, vol. 4, no. 4, pp. 295–312, Jan. 1994.
- [36] N. Refat, M. A. Rahman, A. T. Asyhari, I. F. Kurniawan, M. Z. A. Bhuiyan, and H. Kassim, "Interactive learning experience-driven smart communications networks for cognitive load management in grammar learning context," *IEEE Access*, vol. 7, pp. 64545–64557, 2019.
- [37] P. Weichbroth, "Usability of mobile applications: A systematic literature study," *IEEE Access*, vol. 8, pp. 55563–55577, 2020.
- [38] T. van Gog and F. Paas, "Instructional efficiency: Revisiting the original construct in educational research," *Educ. Psychologist*, vol. 43, no. 1, pp. 16–26, Jan. 2008.

- [39] R. E. Mayer, "Models for understanding," *Rev. Educ. Res.*, vol. 59, no. 1, pp. 43–64, Mar. 1989.
- [40] J. Sweller, J. J. G. van Merriënboer, and F. Paas, "Cognitive architecture and instructional design," *Educ. Psychol. Rev.*, vol. 10, no. 3, pp. 251–296, Sep. 1998.
- [41] F. Paas, J. E. Tuovinen, H. Tabbers, and P. W. Van Gerven, "Cognitive load measurement as a means to advance cognitive load theory," *Educ. Psychol.*, vol. 38, no. 1, pp. 63–71, Mar. 2003.
- [42] J. D. Gould and R. N. Ascher, "Use of an IQF-like query language by non-programmers," IBM Thomas J. Watson Res. Center, Yorktown Heights, NY, USA, Res. Rep. RC 5279, 1975.
- [43] W. C. Ogden, "Implications of a cognitive model of database query: Comparison of a natural language, formal language and direct manipulation interface," *ACM SIGCHI Bull.*, vol. 18, no. 2, pp. 51–54, Oct. 1986.
- [44] A. M. Collins and M. R. Quillian, "Retrieval time from semantic memory," *J. Verbal Learn. Verbal Behav.*, vol. 8, no. 2, pp. 240–247, Apr. 1969.
- [45] J. Turns, C. J. Atman, and R. Adams, "Concept maps for engineering education: A cognitively motivated tool supporting varied assessment functions," *IEEE Trans. Educ.*, vol. 43, no. 2, pp. 164–173, May 2000.
- [46] J. D. Novak and A. J. Cañas, "The theory underlying concept maps and how to construct and use them," Inst. Hum. Mach. Cognition, Pensacola, FL, USA, IHMC CmapTools, Tech. Rep. IHMC CmapTools 2006-01, 2008.
- [47] C. Soderston, N. Kleid, and T. Crandell, "Concept mapping: A job-performance aid for hypertext developers," in *Proc. 14th Annu. Int. Conf. Syst. Documentation Marshaling New Technol. Forces: Building Corporate, Academic, User-Oriented Triangle (SIGDOC)*, New York, NY, USA, 1996, pp. 179–186.
- [48] H.-H. Chen, Y.-J. Chen, and K.-J. Chen, "The design and effect of a scaffolded concept mapping strategy on learning performance in an undergraduate database course," *IEEE Trans. Educ.*, vol. 56, no. 3, pp. 300–307, Aug. 2013.
- [49] N. Dabbagh, "Concept mapping as a mindtool for critical thinking," *J. Comput. Teach. Educ.*, vol. 17, no. 2, pp. 16–23, 2001.
- [50] Y. Erdogan, "Paper-based and computer-based concept mappings: The effects on computer achievement, computer anxiety and computer attitude," *Brit. J. Educ. Technol.*, vol. 40, no. 5, pp. 821–836, Sep. 2009.
- [51] M. N. Ismail, N. A. Ngah, and I. N. Umar, "The effects of mind mapping with cooperative learning on programming performance, problem solving skill and metacognitive knowledge among computer science students," *J. Educ. Comput. Res.*, vol. 42, no. 1, pp. 35–61, Jan. 2010.
- [52] J. Keppens and D. Hay, "Concept map assessment for teaching computer programming," *Comput. Sci. Edu.*, vol. 18, no. 1, pp. 31–42, Mar. 2008.
- [53] A. R. Montazemi and D. W. Conrath, "The use of cognitive mapping for information requirements analysis," *MIS Quart.*, vol. 10, no. 1, pp. 45–56, Mar. 1986.
- [54] V. P. Gurupur, U. Sakoglu, G. P. Jain, and U. J. Tanik, "Semantic requirements sharing approach to develop software systems using concept maps and information entropy: A personal health information system example," *Adv. Eng. Softw.*, vol. 70, pp. 25–35, Apr. 2014.
- [55] K. Siau and X. Tan, "Improving the quality of conceptual modeling using cognitive mapping techniques," *Data Knowl. Eng.*, vol. 55, no. 3, pp. 343–365, Dec. 2005.
- [56] R. Elmasri, and S. Navathe, *Fundamentals of Database Systems*, 7th ed. London, U.K.: Pearson, 2015.
- [57] G. L. Bradshaw and J. R. Anderson, "Elaborative encoding as an explanation of levels of processing," *J. Verbal Learn. Verbal Behav.*, vol. 21, no. 2, pp. 165–174, Apr. 1982.
- [58] R. Ratcliff, "A theory of memory retrieval," *Psychol. Rev.*, vol. 85, no. 2, pp. 59–108, Mar. 1978.
- [59] F. Bodart, A. Patel, M. Sim, and R. Weber, "Should optional properties be used in conceptual modelling? A theory and three empirical tests," *Inf. Syst. Res.*, vol. 12, no. 4, pp. 384–405, Dec. 2001.
- [60] A. King, "Effects of training in strategic questioning on children's problem-solving performance," *J. Educ. Psychol.*, vol. 83, no. 3, pp. 307–317, 1991.
- [61] P. Heller, R. Keith, and S. Anderson, "Teaching problem solving through cooperative grouping. Part 1: Group versus individual problem solving," *Amer. J. Phys.*, vol. 60, no. 7, pp. 627–636, Jul. 1992.
- [62] K. L. Siau, H. C. Chan, and K. K. Wei, "Effects of query complexity and learning on novice user query performance with conceptual and logical database interfaces," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 2, pp. 276–281, Mar. 2004.
- [63] P. Reisner, "Use of psychological experimentation as an aid to development of a query language," *IEEE Trans. Softw. Eng.*, vol. SE-3, no. 3, pp. 218–229, May 1977.
- [64] F. G. W. C. Paas and J. J. G. Van Merriënboer, "The efficiency of instructional conditions: An approach to combine mental effort and performance measures," *Hum. Factors*, vol. 35, no. 4, pp. 737–743, Dec. 1993.
- [65] D. Gopher and R. Braune, "On the psychophysics of workload: Why bother with subjective measures?" *Hum. Factors*, vol. 26, no. 5, pp. 519–532, Oct. 1984.
- [66] R. D. O'Donnell and F. T. Eggemeier, "Workload assessment methodology," in *Handbook of Perception and Human Performance*. New York, NY, USA: Wiley, 1986, pp. 1–49.
- [67] F. Paas, J. J. G. van Merriënboer, and J. J. Adam, "Measurement of cognitive load in instructional research," *Percept. Motor Skill.*, vol. 79, no. 1, pp. 419–430, Aug. 1994.
- [68] P. Ayres, "Using subjective measures to detect variations of intrinsic cognitive load within problems," *Learn. Instruct.*, vol. 16, no. 5, pp. 389–400, Oct. 2006.
- [69] L. Anderson-Inman and L. Ditson, "Computer-based concept mapping: A tool for negotiating meaning," *Learn. Lead. Technol.*, vol. 26, no. 8, pp. 6–13, 1999.
- [70] P. B. Horton, A. A. McConney, M. Gallo, A. L. Woods, G. J. Senn, and D. Hamelin, "An investigation of the effectiveness of concept mapping as an instructional tool," *Sci. Edu.*, vol. 77, no. 1, pp. 95–111, Jan. 1993.
- [71] H. Peng, Y. Su, C. Chou, and C. Tsai, "Ubiquitous knowledge construction: Mobile learning re-defined and a conceptual framework," *Innov. Edu. Teach. Int.*, vol. 46, no. 2, pp. 171–183, May 2009.
- [72] G. Kao, S. Lin, and C. Sun, "Breaking concept boundaries to enhance creative potential: Using integrated concept maps for conceptual self-awareness," *Comput. Edu.*, vol. 51, no. 4, pp. 1718–1728, Dec. 2008.
- [73] P.-H. Wu, G.-J. Hwang, M. Milrad, H.-R. Ke, and Y.-M. Huang, "An innovative concept map approach for improving students' learning performance with an instant feedback mechanism," *Brit. J. Educ. Technol.*, vol. 43, no. 2, pp. 217–232, Mar. 2012.
- [74] C. C. Liu, P. H. Don, and C. M. Tsai, "Assessment based on linkage patterns in concept maps," *J. Inf. Sci. Eng.*, vol. 21, no. 5, pp. 873–890, Sep. 2005.
- [75] J. C. Nesbit and O. O. Adesope, "Learning with concept and knowledge maps: A meta-analysis," *Rev. Educ. Res.*, vol. 76, no. 3, pp. 413–448, Sep. 2006.
- [76] R. M. Marra and D. H. Jonassen, "Transfer effects of semantic networks on expert systems: Mindtools at work," *J. Educ. Comput. Res.*, vol. 26, no. 1, pp. 1–23, Jan. 2002.
- [77] J. R. Anderson, *Rules of the Mind*. Hillsdale, NJ, USA: Lawrence Erlbaum Associates, 1993.
- [78] J. H. Larkin and H. A. Simon, "Why a diagram is (sometimes) worth ten thousand words," *Cognit. Sci.*, vol. 11, no. 1, pp. 65–100, Jan. 1987.
- [79] P. Chandler and J. Sweller, "Cognitive load while learning to use a computer program," *Appl. Cognit. Psychol.*, vol. 10, no. 2, pp. 151–170, Apr. 1996.
- [80] A. Paivio and K. Csapo, "Concrete image and verbal memory codes," *J. Exp. Psychol.*, vol. 80, no. 2, pp. 279–285, 1969.
- [81] D. C. Merrill, B. J. Reiser, S. K. Merrill, and S. Landes, "Tutoring: Guided learning by doing," *Cognition Instruct.*, vol. 13, no. 3, pp. 315–372, Sep. 1995.
- [82] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering*. Berlin, Germany: Springer-Verlag, 2012.



SHIN-SHING SHIN was a Software Project Manager with the Institute for Information Industry, Taiwan. He is an Associate Professor with the Department of Information Science and Management Systems, National Taitung University, Taiwan. He has participated and managed a lot of software projects for a variety of industries. He has published many articles in professional journals, such as *Information and Management*, the IEEE TRANSACTIONS ON EDUCATION, *ACM Transactions on Computing Education*, *Software and Systems Modeling*, and so on. His current research interests include software engineering, information systems evaluation, and computer science education.

• • •