SQL education: A systematic mapping study and future research agenda

TONI TAIPALUS and VILLE SEPPÄNEN*, University of Jyvaskyla, Finland

Structured Query Language (SQL) skills are crucial in software engineering and computer science. However, teaching SQL effectively requires both pedagogical skill and considerable knowledge of the language. Educators and scholars have proposed numerous considerations for the betterment of SQL education, yet these considerations may be too numerous and scattered among different fora for educators to find and internalize, as no systematic mappings or literature reviews regarding SQL education have been conducted. The two main goals of this mapping study are to provide an overview of educational SQL research topics, research types and publication fora, and to collect and propagate SQL teaching practices for educators to utilize. Additionally, we present a short future research agenda based on insights from the mapping process. We conducted a systematic mapping study complemented by snowballing techniques to identify applicable primary studies. We classified the primary studies according to research type, and utilized directed content analysis to classify the primary studies by their topic. Out of our selected 89 primary studies, we identified six recurring topics: *(i)* student errors in query formulation; *(ii)* characteristics and presentation of the exercise database; *(iii)* specific and *(iv)* non-specific teaching approach suggestions; *(v)* patterns and visualization; and *(vi)* easing teacher workload. We list 66 teaching approaches the primary studies argued for (and in some cases against). For researchers, we provide a systematic map of educational SQL research, and future research agenda. For educators, we present an aggregated body of knowledge on teaching practices in SQL education over a time frame of 30 years. In conclusion, we suggest that replication studies, studies on advanced SQL concepts, and studies on aspects other than data retrieval are needed to further educational SQL research.

CCS Concepts: • **Information systems → Query languages**; • **Social and professional topics → Computing education**; **Computer science education**; **Software engineering education**.

Additional Key Words and Phrases: Structured Query Language (SQL), education, database, query language, student

ACM Reference Format:
Toni Taipalus and Ville Seppänen. 20xx. SQL education: A systematic mapping study and future research agenda. 1, 1 (May 20xx), 31 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Among the core topics in software engineering, computer science, and information systems curricula in higher education are databases and Structured Query Language (SQL) [62, 103, 105]. Since SQL is prevalent in database systems, SQL skills are also valued in the software industry, and consequently, teaching SQL effectively is essential in training future software professionals. However, teaching databases requires considerable subject knowledge in addition to pedagogical skill [102]. Additionally, there are several approaches to teaching SQL, and especially for an inexperienced database course teacher, differentiating between patterns (i.e., an effective teaching approach) and anti-patterns (i.e., what merely looks like an effective teaching approach) is difficult [89].

Authors' address: Toni Taipalus, toni.taipalus@jyu.fi; Ville Seppänen, ville.r.seppanen@jyu.fi, University of Jyvaskyla, Faculty of Information Technology, Agora, P.O. Box 35, Jyvaskyla, Finland, FI-40014.

Over the years of teaching SQL, we have come across numerous teaching practices proposed in scientific literature. Yet, these numerous studies may be overly onerous for educators to find and internalize, even superficially. Furthermore, it may not be clear whether a proposed teaching approach has been supported or contested by other scholars, as there are no systematic mapping studies or literature reviews regarding SQL education. To that end, and inspired by the propagation of ideas proposed by Bort et al. [13], we collected 66 teaching approaches from 89 primary studies over the course of 30 years with respective arguments for and against for educators to utilize. For researchers, we present a systematic mapping of SQL education with classifications regarding both the nature and the topic of the studies. Finally, based on the mapping, we present considerations for future research on *what* to study and *how* to study SQL education.

The rest of this study is structured as follows. In Section 2 we discuss the background of this study, i.e., SQL and its role in higher education. In Section 3 we describe the systematic mapping process, the research type and topic classifications, and threats to validity. In Sections 4 and 5 we present the results of our study, i.e., the systematic mapping and the lists of teaching practices, respectively. In Section 6 we discuss our results, and in Section 7 present the future research agenda. Finally, in Section 8 we conclude the study. In Appendices A, B, and C, we present the list of primary studies, a more detailed primary study classifications, and a list of the number of participants in each primary study, respectively.

2 BACKGROUND

2.1 SQL

The evolution of SQL (formerly SEQUEL, or Structured English Query Language [29, 30]) from the theoretical foundations of relationally complete query languages [38] to what SQL is today has been driven by both standardization, and vendors behind database management systems (DBMSs). The first available implementations of SQL were introduced in the late 1970s and early 1980s, and the first SQL standard, SQL-86 in 1986 by the standard groups ANSI and ISO [28]. Since SQL-86, the standard has received eight revisions, SQL:2016 being the latest. Each revision has added additional and alternative features. Despite its age, SQL remains the de facto query language in relational database management systems.

The SQL language features in the SQL standard are divided into *mandatory* (i.e., *core*) and *optional* features, and DBMSs implement both mandatory and optional features to varying degrees. Additionally, DBMSs may implement features differently to how they are described in the SQL standard, and most DBMSs have additional, vendor specific features [87]. Finally, SQL standard conformance testing of SQL implementations by the U.S. National Institute of Standards and Technology (NIST) has been discontinued in 1996, and a feature being *mandatory* is not a guarantee for a feature's implementation. Despite all these points, at least basic SQL statements remain portable with little or no modifications between DBMSs.

SQL is a versatile language, and allows users to retrieve, store, modify and delete data, create, modify and delete database objects (e.g, tables, columns, procedures, users), grant and revoke user privileges, and group statements into transactions. An SQL command is called a *statement*, and a statement which retrieves data from the database a *query*. SQL statements consist of *clauses* (e.g., SELECT, FROM, WHERE), which mainly contain database object names, *predicates* (e.g., LIKE, BETWEEN, EXISTS), *operators* (e.g., AND, OR, NOT), *quantifiers* (e.g., ANY, ALL, UNION), and *functions* (e.g, COUNT, SUM, AVG) [59, 60]. We call these collectively *concepts*, when it is not necessary to differentiate between, for example, clauses and predicates.

SQL is commonly divided into at least two sublanguages, Data Manipulation Language (DML, e.g., SELECT, INSERT, UPDATE, DELETE) and Data Definition Language (DDL, e.g, CREATE, ALTER, DROP) [28, 60]. Additionally, as the revisions of the SQL standard have introduced more features, two more sublanguages, Data Control Language (DCL, e.g., GRANT, REVOKE) and Transaction Control Language (TCL or TxCL, e.g., BEGIN, COMMIT, ROLLBACK), are sometimes discussed in literature. The origins of the sublanguage names DCL and TCL remain unclear, as these names are not explicitly mentioned in the SQL standard. Nevertheless, we have found this division into four sublanguages helpful and rather intuitive, and utilize it in this study.

2.2 SQL in Higher Education

SQL teaching in higher education is both long-lived and widespread. In the information technology subfields, SQL is explicitly mentioned in software engineering (SE) [103], computer science (CS) [62], and information systems (IS) undergraduate curricula guidelines [105], and additionally in areas such as business analytics [113]. These three information technology curricula guidelines expectedly overlap [64], and recommend SQL education on a relatively high level. SE guidelines recommend DML, DDL, and indexes, views, sequences, joins, and triggers in the context of database design. IS guidelines recommend DML, DDL, DCL, and transactions. CS guidelines provide the finest level of detail among the three, and recommend DDL, primary and foreign key attribute and schema definition, query formulation, UPDATE, integrity constraints, selection, projection, aggregate functions, GROUP BY, subqueries, division, stored procedures, and transaction control.

As these guidelines are merely guidelines, and presented at a high level, it is unclear how comprehensively and in depth SQL is covered in courses. Our impression, based on the primary studies and our teaching experience, is that basic DML is commonly discussed. Basic DML includes SELECT, FROM, WHERE, GROUP BY, HAVING, ORDER BY, INSERT INTO, VALUES, UPDATE and DELETE clauses, different types of joins, including the different variations of JOIN, certain predicates like IN, EXISTS, LIKE, BETWEEN, and IS NULL, and standard SQL aggregate functions MIN, MAX, AVG, SUM, and COUNT. However, advanced concepts like recursion, common table expressions, or derived tables are seldom included. It is unclear whether these advanced features are not widely known to educators and curricula designers, or have they been omitted from course contents by design. Basic DDL is commonly discussed, and includes CREATE, ALTER, and DROP statements on tables, views, and users (i.e., *roles*). Table creation includes column name and data type definitions, primary and foreign keys, and the CHECK constraint. However, more advanced concepts such as assertion, trigger, and procedure manipulations are rare. DCL is discussed to a lesser extent than DML and DCL, even though this sublanguage is relatively small and simple. If TCL is included in a course, it is often discussed with examples outside SQL, such as simple *read(a)*, *write(b)* [e.g., 39, p. 669ff.], even though transactions are often defined using SQL. After SQL, database education may focus on non-relational extensions [107], other data models, and data analytics [109].

2.3 Learning Context

In the aforementioned information technology curricula guidelines, SQL is not taught in isolation, but as a part of a database course. Before learning SQL, students need to know, at the very least, about the relational data model, and possibly the theoretical foundations of relational query languages. Nowadays, students learn SQL in digital, interactive environments using an exercise database [23]. This kind of environment can simply be a DBMS to which a student submits queries and receives output.

Alternatively, the environment may be a DBMS's SQL interface embedded in a web page, and the web page fitted with auxiliary elements, for example, a representation of the underlying database schema, a natural language request (i.e., *data demand*) to which a students must write an SQL equivalent, and the correct result table [101]. The database may be represented at the conceptual level as an Entity–Relationship (ER) diagram [37], or at the logical level as a database schema diagram [43]. For both levels, numerous additional or alternative notations can be utilized, for example, Unified Modeling Language class diagrams, enhanced/extended ER, and Logical Data Structures. When a student submits a query, the DBMS outputs either a result table, or an error message. Commonly, SQL errors have been divided into syntax and semantic errors [95]. More recently, research has identified the concept of complications, for example, tautologies and unnecessary elements in queries, which do not affect the result table but performance and readability [17]. Furthermore, semantic errors have been further divided into errors which are evident without knowledge of the underlying data demand, and errors which are only recognizable if the data demand is known [17]. The former kind of semantic errors are called *semantic*, and the latter *logical* [101].

Finally, more advanced environments may be used [19, 20]. These environments provide different additional features, for example, non-binary grading of queries [1], personalized feedback [77], and visualized query execution [49]. These more advanced environments are outside the scope this study.

3 RESEARCH METHOD

3.1 Research Questions

We divided our research questions into two categories. Research questions 1 and 2 are closely related to typical outcomes of the systematic mapping process in software engineering [83], and research question 3 and 4 related to the proposed SQL teaching practices, and to the nature of the proposing studies:

RQ1: In which fora is SQL education research published? There are no publication fora which are specifically focused on SQL education, or even SQL in general. However, both computing education and database research fora in general are plentiful. Answers are presented in Section 4.1.

RQ2: What types of research are represented and to what extent? Educational research is diverse by nature, and while some studies test clearly formulated hypotheses, others report opinions and experiences. We want to understand the SQL education landscape to identify potential dearths in research. Answers are presented in Section 4.2 and Appendix B.

RQ3: Which practices have been proposed for teaching SQL? In addition to lectures, textbooks, and practical exercises, studies have identified and proposed practices (e.g., new tools, teaching methods, or increased emphasis on specific topics) to more effectively teach SQL. Answers are presented in Section 5.

RQ4: What kind of evidence is presented to support the proposed practices? Whereas some SQL education studies report practices as results, others merely suggest different practices in their respective discussion sections. We want to differentiate between educated opinions and scientifically supported (or contested) propositions. Answers are presented in Section 5 and Appendix C.

3.2 Search Strategy

We searched four digital libraries without applying date or publication type restrictions: ACM Digital Library, IEEExplore, ISI Web of Science, and Scopus, which include arguably the most recognized computing science education research fora such as *ACM Transactions on Computing Education*, *Computer Science Education*, and the SIGCSE and ITiCSE

Table 1. Search strings

Database	Search string
ACM DL	("structured query language" OR SQL) AND (education OR teaching OR student OR students OR learning)
IEEExplore	(("structured query language" OR SQL) AND (education OR teaching OR student OR students OR learning))
Web of Science	TS=(("structured query language" OR SQL) AND (education OR teaching OR student OR students OR learning))
Scopus	TITLE-ABS-KEY(("structured query language" OR SQL) AND (education OR teaching OR student OR students OR learning))
AIS eLibrary	(SQL OR "structured query language")
JISE	SQL

conferences. Due to the pervasive nature of SQL in the information and communication technology (ICT) field, we also searched two information systems focused databases: AIS eLibrary and the database of *Journal of Information Systems Education* (JISE). Search strings are presented in Table 1.

The database searches yielded a total of 2,709 studies, 414 from ACM Digital Library, 646 from IEEExplore, 228 from Web of Science, 1,361 from Scopus, 46 from AIS eLibrary, and 14 from JISE. Additionally, we had 16 papers [4–6, 16, 17, 25, 40–42, 70, 71, 94, 98, 100, 101, 121] which we knew well enough to deem them suitable for closer criteria evaluation (cf. Table 2). The AIS eLibrary search was limited to peer-reviewed repositories. Both the AIS eLibrary and JISE search strings were more inclusive than the others due to the relative small size of the databases, although the former is only small if limited to peer-reviewed repositories. Google Scholar was considered, but a preliminary search returned too many results to inspect in a feasible timeframe.

The study selection process is illustrated in Fig. 1. The rectangles labeled A1 and A2 indicate the authors who performed the corresponding step. We performed backward snowballing (i.e., following reference lists to find relevant studies) [114] twice. Both authors studied the papers independently, and marked them both according to the research type facet classification (Table 3), and whether the paper should be included or excluded and why. We then compared our notes. In case of disagreement, we discussed until we reached a consensus on whether to include or exclude a paper, and how to classify the paper according to the research type facets. The comparison and discussion step was performed twice, and 89 primary studies were selected.

3.3 Study Selection

The searches yielded many papers concerning machine learning, SQL injection, and SQL learning environments. These papers were excluded because different learning environments were outside the scope of this study, *learning* in *machine learning* is not related closely enough to human learning in the context of this study, and education regarding SQL injection is more concerned with the design of the application program rather than SQL. To a lesser extent, the searches returned papers concerned with procedural extensions of SQL (e.g., T-SQL and PL/SQL), query optimization, NoSQL, data warehousing, and web development, all of which were excluded. We were relatively unanimous in our study inclusion/exclusion discussions, yet one study [106] was particularly difficult. The study explores the effects of task complexity and time limitations on query writing, and gives every implication that the query language

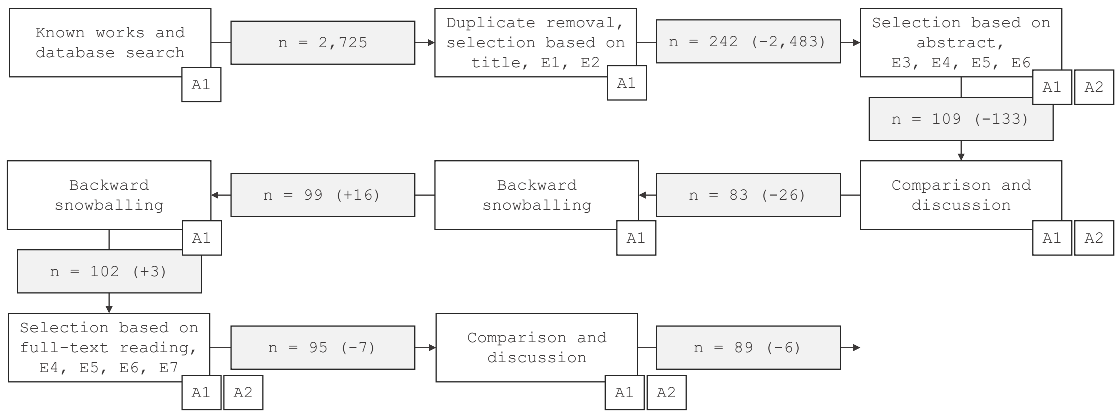

Fig. 1. Study selection process - A1 and A2 refer to the authors, *E* refers to an exclusion criterion described in Table 2, and *n* indicates the number of included papers

Table 2. Inclusion (I) and exclusion (E) criteria

ID	Criterion	Example studies
I1	present considerations on teaching or learning SQL	
I2	published online during the time frame 1989 to Sep. 2019	
E1	published in non-peer reviewed forum	[48]
E2	not written in English	
E3	full text we could not find or download	[36, 68, 85, 120, 123]
E4	do not mention SQL, or SQL is merely an example or a vehicle	[9, 11, 27, 55, 67, 92, 93, 96, 111]
E5	concerned with SQL alternatives rather than teaching SQL	[22, 26, 31, 33, 35, 81]
E6	focus on describing an SQL learning environment	[2, 19, 20, 76, 78, 82, 86]
E7	lack sufficient detail to suggest a detailed teaching approach	[3, 52, 57, 69, 79, 116–118, 122]

under study is indeed SQL. However, SQL is not mentioned in the paper, with the exception of a table summarizing prior work. The study was not included, along with several other borderline exclusions [7, 11, 19, 20, 23, 27, 31–33, 45, 46, 50, 52, 55, 61, 63, 73, 79, 93, 97, 104]. Finally, we recognize that it is increasingly common for ICT research to report implications for research, industry, and teaching. It follows that there are most likely papers that report implications to SQL education, but were not found by our search criteria. As indications for finding these implications are not often found in abstracts, reading, for example, all relational database related research was not feasible, and not done in this study.

Inclusion (I1–I2) and exclusion (E1–E7) criteria are presented in Table 2. We decided to exclude papers published before 1989. Although this year marked the publication of the first revision to the SQL standard, SQL-89, this was not an educated choice as much as conveniently including 30 years of SQL education research.

3.4 Data Extraction

We extracted basic reference information from the database searches: names of the authors, title, publication year, name of the publication forum, and issue number, volume, and page numbers, where applicable. Once the primary studies were selected, we classified each paper according to the research type and research topic facets. We also marked why a paper was excluded according to our predefined exclusion criteria, the number of citations from Google Scholar, and the number of participants in each study.

3.5 Classification

Wieringa et al. [112] propose six classes (or research type facets, as Petersen et al. [83] summarize) for requirements engineering papers. As we began categorizing our primary studies according to these facets, it became clear that they are not a natural fit with educational research papers. Two particularly problematic classes were validation and evaluation research. To summarize Wieringa et al. [112], validation research in requirements engineering is effectively prototyping, simulation, and experiments (i.e., *in vitro*), whereas evaluation research is effectively case study, field study, or survey (i.e., *in vivo*). In educational research, the dividing line between *in vitro* and *in vivo* is more difficult to determine. If *in vivo* studies are concerned with students in their natural learning environments, how much restriction (e.g., limiting the time to complete an exercise, forbidding communication or use of online materials) constitutes in making the research setting *in vitro*? Rather than trying to estimate how natural the research settings of the primary studies were, we adapted the research type facet classification to better fit educational research (Table 3).

As Wieringa et al. [112] point out, it is possible that one study covers more than one research type facet. For example, Taipalus et al. [101] present a query concept framework based on their teaching experience, an error categorization framework based on a qualitative study, and opinions on how SQL should be taught. By these three aspects, their study could be classified as a philosophical paper, evaluation research, or an opinion paper. We classified each primary study according to what we perceived as their primary contribution. As discussed in Wohlin et al. [115], these classifications merely represent an overview of the type of research in a given mapping.

After the final 89 primary studies were selected, the first author classified the studies into categories according to their topics. This was done based on full text reading, and according to directed content analysis [56] with the utilization of prior knowledge on SQL education research categories. Using preconceived topic categories, the first author classified each primary study into a category. If a study was concerned with a topic which did not fit to any category, a new category was considered. Topic categories are reported in Section 4.2, and this category scheme is used to structure Section 5, in which we report the teaching approaches in more detail.

3.6 Threats to Validity

3.6.1 Descriptive Validity. Descriptive validity concerns the objectivity and accuracy of the data gathering. We utilized a data collection form described in Section 3.4 to increase the objectivity and accuracy of the classification and study selection. Both authors used the same form when selecting the studies and classifying the research type.

3.6.2 Theoretical Validity. Theoretical validity concerns the selection and classification of the data. We tried to minimize the possibility of missing relevant studies by searching several databases, and by performing backward snowballing twice (Fig. 1). The first snowballing yielded 16 additional studies, and the second 3, yet after closer inspection, not all these studies were included in the final selection. As Petersen et al. [84] point out, researcher bias is a known threat to validity in the study selection phase. We tried to mitigate this by performing research type classification and applying

Table 3. Research type facet in educational research (adapted from Wieringa et al. [112] and Petersen et al. [83])

Category	Description
Evaluation research	Hypotheses are tested on (or phenomena studied among) their natural target populations, and preferably in as natural environments as possible. This means that if the hypotheses are concerned with evaluating a new method for teaching students, the natural target is a student or a novice in a given technique (e.g., a language), and the most natural testing environment should be the environment the students would be using regardless of the research setting. This is not always possible, and varying degrees of unnatural elements must often be included. Sufficient quantifiable evidence is presented.
Solution proposal	Paper presents a new or significantly improved solution for a common and recognized problem. The topic may be related to concepts that are difficult for students to learn, teacher's workload, or curriculum improvement. Solid arguments for (and preferably against) the proposal are presented.
Replication study	Paper replicates a previously reported research setting as accurately as possible, or with premeditated alterations (e.g., a different teacher, students who major in a different subject, or undergraduate instead of graduate students). The goal of the study is to check the validity of the previous study, or to study generalizability of the results. Sufficient quantifiable evidence is presented.
Philosophical paper	Paper presents a new conceptual framework, taxonomy, general teaching approach, new, improved or adapted research method, or simply summarizes existing work in a form of systematic literature review or systematic mapping. Depending on the type of philosophical paper, the paper may utilize existing literature, or be based on professional opinions or experiences.
Opinion paper	Paper expresses opinions of the author or a third party. These opinions may be concerned with, for example, whether something should be taught, how it should be taught, or to whom it should be taught. Typically no scientific evidence is presented.
Experience report	Paper describes how something was done, for example, a course or a curriculum implementation. The new setting should be described in sufficient detail, so that others may replicate it. Paper should report what worked and what did not.

exclusion criteria E3–E7 independently, and comparing results afterwards. Topic classification (Section 4.2) was done solely by the first author, and is the main threat to validity in this regard.

3.6.3 Interpretive Validity. Interpretive validity concerns researchers' biases in the interpretation of the data. The first author is an author of several selected primary studies, which may induce bias in interpretation. The second author, however, is not, and there were no disagreements on whether to include or exclude those studies. The first author's experience in educational research concerning SQL was also considered helpful in the study selection and classification processes, although this may have biased the interpretation of primary study results.

3.6.4 Repeatability and Generalizability. In order to increase the repeatability of our results, we followed systematic mapping guidelines proposed in Petersen et al. [83] and complemented later in Petersen et al. [84]. We also reported threats to validity, and how we tried to mitigate them. However, study selection and classification involve human judgment, and another group of researchers might select at least slightly different set of primary studies.

4 PUBLICATIONS

4.1 Publication Fora

Out of the 89 primary studies, 38 (43%) were journal articles published in 18 different journals. 50 (56%) studies were presented in 31 different conferences, and one primary study was a workshop paper. As can be observed in Table 4, the studies subject to this mapping study have been published in various journals and conference proceedings, and searching teaching approach proposals should not be limited to merely educational fora. Appendix A lists the primary studies and their corresponding identifiers.

Publication fora and citations among primary studies are illustrated in Fig. 2. We have clustered the primary studies according to the journal (JISE, JCSC) or the organization (ACM, IEEE, Elsevier, and top IS) the forum is associated with. In addition to the AIS *senior scholars' basket of journals*[1] in IS, the top IS cluster contains primary studies from associated fora (CAIS and ICIS) with the exception of JISE, which formed a large enough cluster on its own. The color of an edge corresponds to the citing publication, for example, a blue edge between a red and a blue node indicates that the blue node is citing the red. Alternatively, a clockwise curving edge from node x to node y indicates that x cites y.

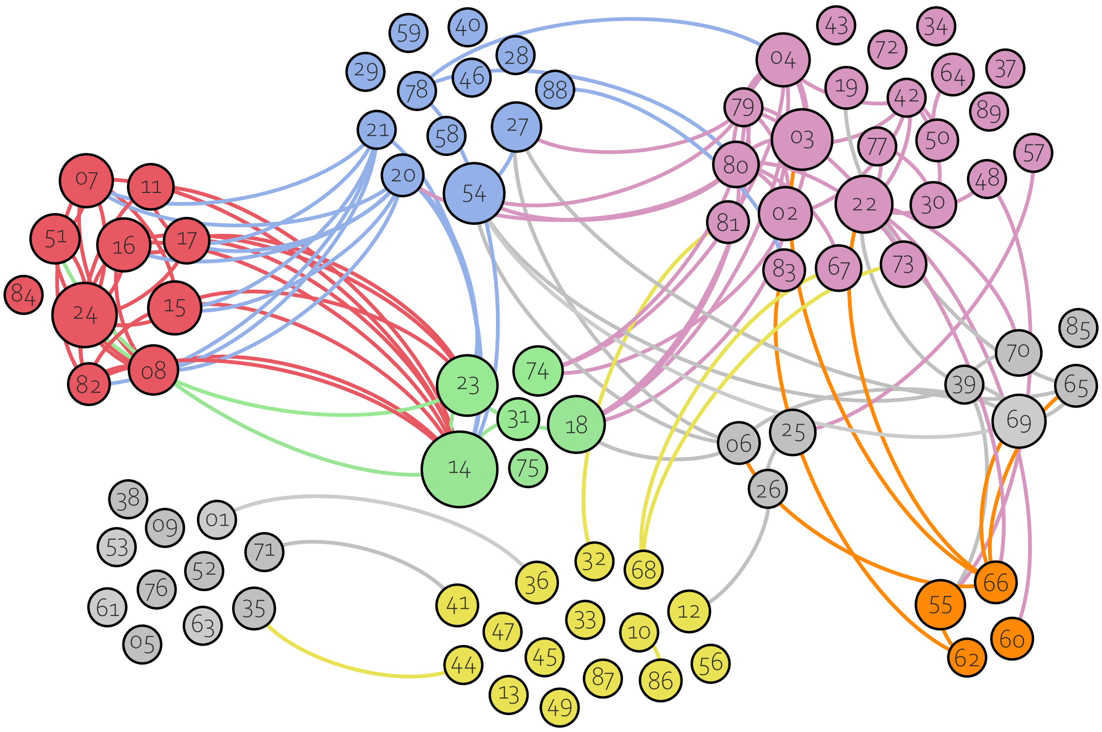

Fig. 2. Publication fora clusters and citations among primary studies - studies published in ACM journals or conference proceedings are clustered top right (purple), *Journal of Computing Sciences in Colleges* bottom right (orange), IEEE bottom center (yellow), top IS fora top left (red), *Journal of Information Systems Education* top center (blue), Elsevier center (green), and other fora (gray); size of a node represents in-degree, color of an edge corresponds to citing publication, edge curves clockwise from the citing to the cited publication, and numbers refer to primary study IDs

With a few exceptions, the top IS studies cite each other extensively, while citing among ACM studies varies. JISE, IEEE, Elsevier and JCSC studies cite each other relatively seldom. A small percentage of JISE studies cite top IS studies and some ACM and Elsevier studies. Top IS studies cite some Elsevier studies, but nothing else. ACM studies cite mostly Elsevier and JISE studies, but not IEEE or top IS studies. JISE studies cite mostly top IS, Elsevier, and ACM studies, but

[1] https://aisnet.org/page/SeniorScholarBasket

Table 4. Number of primary studies published in each forum

Forum name	Type	#
Journal of Information Systems Education (JISE)	Journal	12
ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)	Conference	6
ACM Technical Symposium on Computer Science Education (SIGCSE)	Conference	6
IEEE Frontiers in Education Conference (FIE)	Conference	4
Journal of Computing Sciences in Colleges (JCSC)	Journal	4
ACM Conference on Information Technology Education (SIGITE)	Conference	3
British National Conference on Databases (BNCOD)	Conference	3
Intl. Journal of Human-Computer Studies (IJHCS)*	Journal	3
ACM Conference on Management of Data (SIGMOD)	Conference	2
Information Systems Research (ISR)	Journal	2
Intl. Conference on Information Systems (ICIS)	Conference	2
Intl. Journal of Engineering Education	Journal	2
Journal of the Association for Information Systems (JAIS)	Journal	2
MIS Quarterly	Journal	2
ACM Annual Southeast Regional Conference (ACM-SE)	Conference	1
ACM Conference on Computer Personnel Research (SIGCPR)	Conference	1
ACM Conference on Extending Database Technology (EDBT)	Conference	1
ACM Conference on Information Technology Curriculum (CITC)	Conference	1
ACM Transactions on Computing Education (TOCE)	Journal	1
Annual Meeting of the Decision Sciences Institute	Conference	1
ASEE Annual Conference and Exposition	Conference	1
Communications of the Association for Information Systems (CAIS)	Journal	1
Computers & Education	Journal	1
Decision Support Systems	Journal	1
IEEE Annual Computer Software and Applications Conference (COMPSAC)	Conference	1
IEEE Conference on Industrial Electronics and Applications (ICIEA)	Conference	1
IEEE Global Engineering Education Conference (EDUCON)	Conference	1
IEEE Integrated STEM Education Conference (ISEC)	Conference	1
IEEE Intl. Conference on Data Engineering (ICDE)	Conference	1
IEEE Intl. Conference on Information Systems and Economic Intelligence (SIIE)	Conference	1
IEEE Intl. Conference on Intelligent Computer Communication and Processing (ICCP)	Conference	1
IEEE Intl. Conference on Networked Computing and Advanced Information Management (NCM)	Conference	1
IEEE Intl. Conference on Scalable Computing and Communications (ScalCom)	Conference	1
IEEE Transactions on Education (TOE)	Journal	1
IEEE Transactions on Software Engineering (TOSE)	Journal	1
Information Systems Education Conference (ISECON)	Conference	1
Intl. Conference on Cognition and Exploratory Learning in Digital Age (CELDA)	Conference	1
Intl. Conference on Computer Supported Education (CSEDU)	Conference	1
Intl. Conference on Interactive, Collaborative and Blended Learning (ICBL)	Conference	1
Intl. Conference on Web-Based Learning (ICWL)	Conference	1
Intl. Conference on Very Large Data Bases (VLDB Conference)	Conference	1
Intl. Convention on Information, Communication and Electronic Technology (MIPRO)	Conference	1
Intl. Journal of Learning, Teaching and Educational Research (IJLTER)	Journal	1
Intl. Journal on Very Large Data Bases (VLDB)	Journal	1
Intl. Scientific Conference Computer Science	Conference	1
Intl. Workshop on Teaching, Learning and Assessment of Databases (TLAD)	Workshop	1
Journal of Management Information Systems (JMIS)	Journal	1
Journal of Systems and Software	Journal	1
Journal on Systemics, Cybernetics and Informatics (JSCI)	Journal	1
UK & Ireland Computing Education Research Conference (UKICER)	Conference	1
Total		89

* Formerly Intl. Journal of Man-Machine Studies

not IEEE studies. IEEE studies cite some ACM studies, but nothing else. Two of the JCSC studies cite a total of three ACM studies. Some Elsevier studies cite some top IS studies. Publications per year are presented in Fig. 3.

The graph in Fig. 2 can be considered an indication of potentially untapped relevant primary research between clusters, yet it should be interpreted with caution. First, the edges only represent citations *among primary studies*, and are not an indication of how many citations a study has received (which, in itself, is not an indication of, for example, quality of a study). Second, the age of a study has a natural effect on the number of citations. The number of citations overall are presented in Table 5, to give an indication of the most commonly cited primary studies in SQL education. It is worth noting that even though a primary study is not cited among the selected primary studies, it may have received scientific attention outside these primary studies, as is the case with, for example, PS75. Finally, all SQL related research is not, intuitively, relevant to each other, but the graph propounds the view that researchers are studying similar aspects of SQL education without knowledge of each other. We give examples that support this argument in Section 7.3.

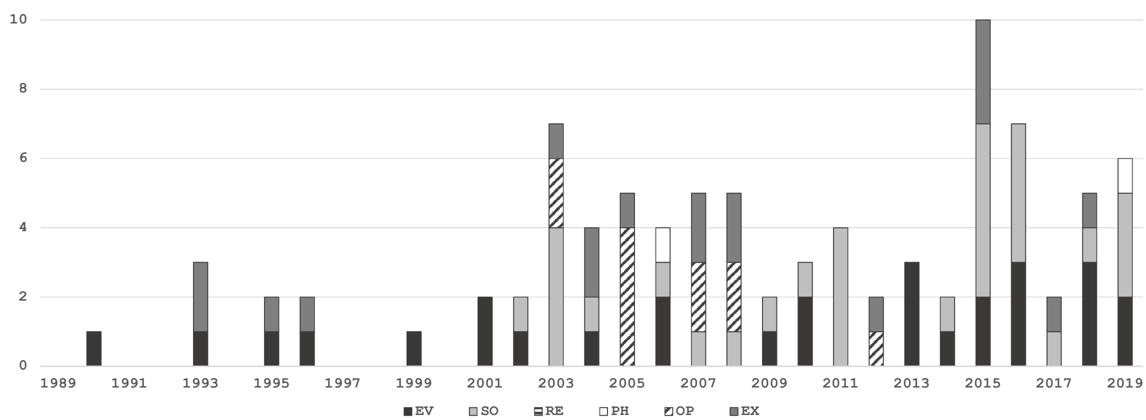

Fig. 3. Number of publications per year, and research type facets: evaluation research (EV), solution proposal (SO), replication study (RE, none present), philosophical paper (PH), opinion paper (OP), and experience report (EX)

4.2 Classification

We presented our adapted research type facet classifications in Table 3. Additionally, we classified the primary studies according to their topics, which we next describe briefly, and in detail in Section 5. It is worth noting that the categories overlap, and that a number of studies were candidates to more than one category. The names and descriptions of the categories are based on full-text reading of the primary studies, and constructed using directed content analysis [56]. The summary of primary study distribution between these two classifications is presented in Fig. 4, and detailed classification in Appendix B.

Studies concerning *student errors* (11 papers): these studies are concerned with presenting what kind of errors students commit during their query formulation processes, what types of errors students usually cannot fix, possible reasons why query formulation fails, and how to teach SQL in a DBMS or context independent viewpoint. Understanding what are the most common errors and what causes them is a crucial step in demonstrating and mitigating these errors in teaching. Most of these studies are evaluative in nature.

Table 5. Primary studies, number of citations from Google Scholar in Sep. 2019, and citations divided by publication age in full years - PS09 was not indexed by Google Scholar

ID	citations	citations/y	ID	citations	citations/y	ID	citations	citations/y
PS75	70	14	PS76	6	1.2	PS62	2	0.3
PS02	22	5.5	PS45	11	1.1	PS82	1	0.3
PS18	70	5	PS69	18	1.1	PS27	4	0.2
PS08	43	4.3	PS70	17	1.1	PS29	2	0.2
PS30	39	4.3	PS01	4	1	PS38	3	0.2
PS07	59	4.2	PS57	1	1	PS48	1	0.2
PS24	113	4.2	PS81	17	1	PS55	3	0.2
PS16	58	4.1	PS84	15	0.9	PS61	3	0.2
PS17	38	3.8	PS15	13	0.8	PS63	3	0.2
PS25	19	3.8	PS19	10	0.8	PS65	1	0.2
PS73	15	3.8	PS31	22	0.7	PS67	3	0.2
PS03	13	2.6	PS52	2	0.7	PS87	2	0.2
PS12	13	2.6	PS53	5	0.7	PS32	2	0.1
PS86	71	2.6	PS36	5	0.6	PS37	1	0.1
PS04	10	2.5	PS46	5	0.6	PS72	4	0.1
PS80	5	2.5	PS49	7	0.6	PS21	0	0
PS33	12	2.4	PS83	11	0.6	PS42	0	0
PS64	38	2.4	PS06	5	0.5	PS44	0	0
PS14	44	2.3	PS13	1	0.5	PS47	0	0
PS26	9	2.3	PS66	1	0.5	PS58	0	0
PS50	33	2.2	PS88	3	0.5	PS59	0	0
PS23	44	2.1	PS11	8	0.4	PS68	0	0
PS22	17	1.9	PS28	5	0.4	PS77	0	0
PS10	32	1.8	PS41	5	0.4	PS78	0	0
PS74	36	1.4	PS54	8	0.4	PS79	0	0
PS20	9	1.3	PS56	7	0.4	PS85	0	0
PS34	31	1.3	PS71	2	0.4	PS89	0	0
PS39	6	1.2	PS35	2	0.3	PS05	0	0
PS51	29	1.2	PS40	5	0.3	PS09	–	–
PS60	14	1.2	PS43	4	0.3			

Studies concerning the *exercise database* and elements closely related to it (20 papers): these studies evaluate, report experiences, and present opinions and solutions in regard to what kind of an exercise database is efficient in facilitating SQL learning. The studies discuss how to visually present the exercise database schema to students, how to express the data demands, what kind of database business domains should be used, how realistic the database should be in terms of data, and whether the students should be made aware if their SQL queries are logically correct. Most of these studies are evaluative in nature.

Studies presenting a *specific teaching approach* (9 papers): these studies present a teaching approach which concerns a specific subset of SQL, for example, how relational division, outer join, or existence negation should be taught. Most of these studies are solution proposals and opinion papers.

Studies presenting a *non-specific teaching approach* (22 papers): these studies discuss a more general teaching approach which should or could be used in teaching all SQL in a given course. The studies propose, for example, group learning and projects, how a course should be structured, and what kind of general techniques can facilitate SQL learning. Most of these studies are experience reports.

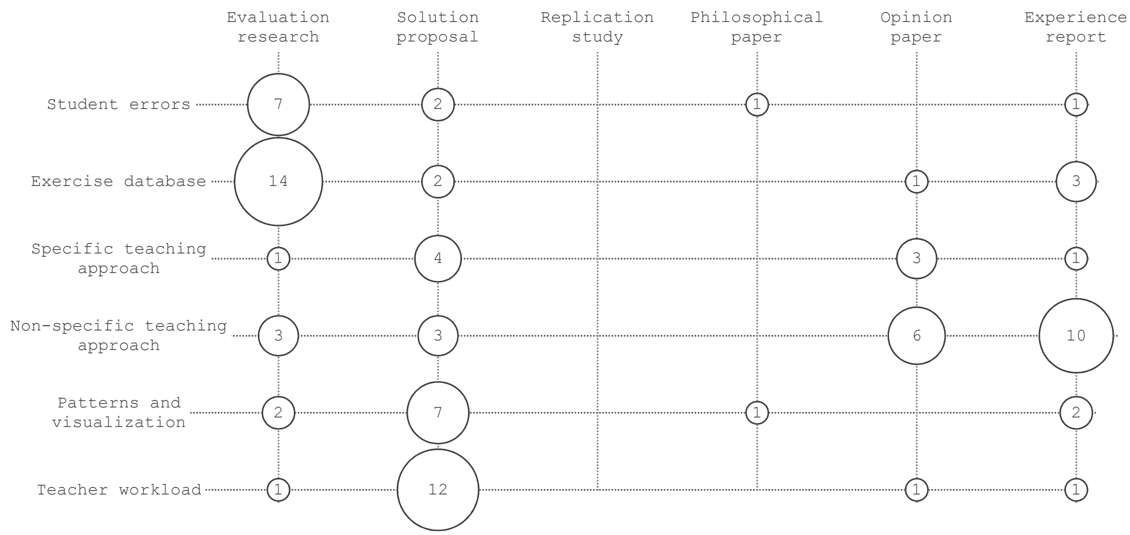

Fig. 4. Number of primary studies in each research type facet (x-axis) and topic facet (y-axis) intersection

Studies discussing *patterns and visualization* (12 papers): these studies mainly propose solutions on how to visualize the query execution process to students, whether to use visual query builders to facilitate SQL learning, planning queries before writing using a specialized notation, and utilizing steps and natural language patterns in query formulation. Most of these studies are solution proposals, and many overlap with the previously described category.

Finally, a number of studies proposed approaches to ease *teacher workload* (15 papers): these studies proposed solutions concerning, for example, automated exercise generation, automated grading and feedback, and pointed educators to materials available online. Arguably, as the teacher workload lightens, educators can focus more on difficult concepts regarding SQL.

5 SQL TEACHING PRACTICES

All the teaching considerations listed in this section are not actionable advice per se, but, for example, concerned with the most common errors students commit. These insights may be utilized by the teacher to focus on certain query concepts during lectures or in exercise design. Furthermore, these errors can be utilized in exercise database data generation, so that at least incorrect queries with the most common logical errors return data that is different from the correct result table. Finally, these errors may be used to guide digital learning environment development, so that feedback for the most common errors may be generated. It is worth noting that we have applied the nomenclature discussed in Section 2 to all the following teaching practice presentations.

Teaching considerations regarding student errors, the exercise database, specific and non-specific teaching approaches, patterns and visualization, and teacher workload are compiled into Tables 6, 7, 8, 9, 10, and 11, respectively. The teaching approaches are not in any particular order regarding arguments for and against. In other words, it is arbitrary whether an approach is presented as *Teach x [PS01; argued against in PS02]* or *Do not teach x [PS02; argued against in PS01]*. Appendix C lists the number of participants in each primary study.

Table 6. Teaching approaches or considerations regarding student errors

ID	Teaching approach or consideration
SE1	A list of semantic errors and complications can be used to support discussion with students on bad query writing practices [PS18]. This list is complemented with syntax and logical errors [PS80], and together give high level representation of what kinds of errors students can commit. Both of these lists are too long to discuss here.
SE2	Self-join is the most difficult query concept overall [PS02, PS03, PS04, PS79], and these queries fail due to logical errors [PS02], namely join errors: joins are formed with incorrect tables, columns, or comparison operators, a join is missing, or a join is extraneous and needs to be omitted [PS79].
SE3	After self-join, the most difficult query concepts are, not in order, correlated subquery [PS03, PS04, PS79], simple one-table query [PS04], simple subquery [PS03], grouping restrictions [PS04], uncorrelated subquery [PS79], and expressions with nesting [PS79].
SE4	The most common errors are, in order, incorrect ordering of columns in the SELECT clause, undefined column name used, joining incorrect columns from correct tables, unnecessary joins, extraneous tables, omitting tables, missing expression, aliases that are always identical, extraneous GROUP BY clause and COUNT function, and incorrect ordering of clauses [PS24].
SE5	Logical errors are the most common class of errors overall [PS79, PS80], and the most difficult class of errors to fix [PS04, PS79]. 40% of errors students commit are semantic or logical in nature, and occur in the SELECT and WHERE clauses [PS04].
SE6	The most frequent errors that student cannot fix are, in order, illegal or insufficient grouping, *common syntax errors*, inconsistent expression, inconsistent joins, missing joins, expression errors such as missing or extraneous expressions, or expressions in incorrect clause, and projection errors such as missing or extraneous columns in the main the SELECT clause [PS79].
SE7	54% of errors student commit are syntactical in nature, and 69% of syntax errors are caused by typing errors [PS04].
SE8	Most frequent syntax errors are *common syntax errors* [PS02, PS79, PS80] and the use of undefined database objects [PS02, PS24, PS79], although the latter type of errors are usually fixed [PS79].
SE9	The next most frequent syntax errors are, in order, grouping errors, use of aggregate functions in the GROUP BY clause, use of undefined operators, and problems with writing expressions [PS02].
SE10	Among queries requiring the use of aggregate functions, illegal or insufficient grouping is the the most frequent type of error [PS02, PS79], followed by the use incorrect functions, incorrect columns as function parameters, missing DISTINCT from the function parameter, and DISTINCT as a function parameter where not applicable [PS79].
SE11	Syntax errors are the cause of failure particularly in queries involving GROUP BY and HAVING clauses, as well as NATURAL JOIN [PS02].
SE12	In multi-table queries, the most frequent errors are, in order, inconsistent joins, missing joins, and join errors such as joins on incorrect table or using incorrect columns [PS79].
SE13	Unnecessary complications are frequent in all queries, regardless of the query concepts required [PS79, PS80].
SE14	Errors are usually caused by short-term memory limitations, absence of a clue in the data demand, procedural fixedness, or ignorance [PS74].
SE15	Most frequent omission errors are, in order, omitting a join clause, omitting a subquery, and omitting the HAVING clause [PS04].
SE16	When a student attempts an exercise more than 30 times, and there is at least one error regarding aggregate function usage in the GROUP BY or WHERE clause, it is statistically unlikely that the student can successfully formulate the correct query [PS02].
SE17	Teach standard SQL because using merely one DBMS will confuse students what is standard and what is DBMS specific. A practical approach to this is to choose two DBMSs to teach students [PS67].
SE18	Teach SQL as a general language that is used in modern tools (e.g., NewSQL DBMSs) as well to mitigate motivational concerns on the relevance of the language [PS73].

Table 7. Teaching approaches or considerations regarding the exercise databases

ID	Teaching approach or consideration
DB1	A list of 19 query concepts that introductory database course exercises may test, and corresponding 15 exercises with example answers are presented [PS80]. This list is too long to be presented here.
DB2	Data demands should be formulated with as little ambiguity as possible [PS11, PS65]. A less ambiguous data demand entails fewer attempts [PS11, PS14], more perceived confidence [argued against in PS14] and correctness [PS11], less time spent [PS11], and less errors [PS11, PS14, PS21]. There exist at least seven types of ambiguity [PS11].
DB3	Unambiguous data demands are more and more important as data demands' complexity increases [PS20].
DB4	With low complexity queries, less ambiguous data demands produce less query formulation errors, but with high complexity queries, data demand ambiguity has no effect on errors [PS21, PS82].
DB5	As training progresses, students should be introduced to more and more ambiguous data demands, which better reflect their future work environments [PS82].
DB6	When GROUP BY clause is needed, the natural language representations should (at least in early exercises) contain a clear indication to use it [PS03].
DB7	Presenting the database as an event-based ER or state-based ER does not affect query accuracy or student confidence [PS07], but in regard to query formulation success rates, it is better to represent a database schema rather than a list of database contents or an ER diagram [PS31]. Furthermore, database representation semantics [PS51], symbols [PS51; argued against in PS10], and foreign key constraint representation [PS51] all have influence on query formulation success.
DB8	The three most important factors in query formulation success rates and time needed are, in order, data model representation realism, high expressive ease, and query complexity. Data model representation realism refers to which level the data model is represented, and the levels are, in ascending order of realism, physical, logical, and conceptual. Expressive ease is concerned with the language used, were it SQL, natural language, or something else [PS23].
DB9	If incongruence (i.e., how well or poorly real world constructs match their equivalents in the database) increases, success rates fall, more time is needed, and students feel less confident [PS14]. However, best design practices (e.g., database normalization) should not be sacrificed in order to reach more ontological clarity, as the implications for benefits are conflicted [PS15, PS16, PS17; argued against in PS74].
DB10	Provide an interface (or a cheatsheet) that allows students to see SQL keywords and database object names to reduce typing errors [PS05]. Consider highlighting relevant parts of the data model for each data demand [PS82].
DB11	If data demand complexity increases, success rates fall, more time is needed, and students feel less confident [PS14].
DB12	Allowing students reuse similar queries in exercises leads to faster query formulation, but results in more errors, and a poorer relationship between confidence and query correctness [PS08].
DB13	Students should not execute queries in the same exercise database, because modifications affect others [PS44].
DB14	Use complex [PS41, PS62, PS83; argued against in PS57 because students cannot manually check problems with erroneous queries against complex data] and low quality [PS83] exercise data because students need to gain understanding of complex environments, and that real data contains errors and missing values. Furthermore, use databases with business domains which are novel to the students so that students learn the importance of domain knowledge and can recognize abstract patterns and utilize them in different domains [PS46]. More realistic databases are perceived more interesting an useful by students [PS88].
DB15	Provide students with the correct result table [PS68, PS80; argued against in PS03 as students may use brute force to write correct queries], or the number of rows in the correct result table [PS68]. If these are not provided, students should validate their results by manually writing tests [PS20]. Students should understand that query evaluation against a single dataset is not enough [PS39].

6 DISCUSSION

6.1 Patterns and Anti-patterns

Even though we listed numerous teaching approaches in the previous section, it remains unclear which approaches are patterns and which are anti-patterns, and in which contexts. As may be observed in the previous section, we do not differentiate between approaches based on objectively interpreted results and subjective discussion. Consequently, we

Table 8. A list of teaching approaches or considerations regarding a specific teaching approach

ID	Teaching approach or consideration
SA1	Teach relational division with GROUP BY and HAVING, rather than multiple existence negations. This is easier for students to learn [PS54, PS56], and the written queries are computationally faster [PS54; the latter point is argued against in PS56]. This work is extended from teaching relational division to teaching set comparison with a general approach [PS27].
SA2	Teach OUTER JOIN according to ANSI SQL-92, i.e., with OUTER JOIN rather than UNION or derived tables. This is perceived easiest and it is computationally faster than the alternatives [PS55].
SA3	Teach existence negation with an English-like query language before teaching the SQL equivalent [PS48].
SA4	Explain the differences in the logic of NOT EXISTS and NOT IN subqueries [PS18].
SA5	Teach strict grouping [PS21, PS80]. Effectively, this means that if the main SELECT clause contains at least one aggregate function, and at least one grouping column, all and only the grouping columns must be included in the GROUP BY clause.
SA6	Teach integrity constraints by dividing them into five classes: dynamic, domain, tuple, relation, and database integrity constraints [PS29].
SA7	If you use Microsoft Access to teach SQL, and want to teach recursive joins which are not supported, stored procedures can be used to complement SQL [PS28].
SA8	Teach transaction control using real SQL examples, and not simple READ(a) and WRITE(b) that are usually found in database textbooks [PS37].
SA9	Teaching SQL after QBE yields better results than teaching SQL first [PS86; the use of QBE is argued against in PS69 because mental models must be changed when switching to SQL].

Table 9. Teaching approaches or considerations regarding a non-specific teaching approach

ID	Teaching approach or consideration
NA1	Emphasize practical work [PS64, PS71].
NA2	Teach SQL with short online lectures [PS81].
NA3	Students should learn SQL in teams [PS01, PS09, PS34, PS41, PS53] and group projects [PS32, PS40, PS59, PS63, PS72], and the project should be based on realistic and reported specification [PS63]. These groups should be formed based on student skill, and the level of difficulty of the exercises set accordingly [PS38]. The online environment utilized in the course should facilitate team forming [PS01].
NA4	Teach students how to read SQL [PS61] before writing SQL [PS19]. Furthermore, demonstrate DBMS error messages [PS20] and erroneous queries [PS43], especially regarding difficult concepts such as ALL and NOT EXISTS [PS61]. Have students explain why they are erroneous, and why a certain solution works [PS43, PS61, PS89].
NA5	Have students come up with analogies for SQL query concepts and predicates. This helps students understand the concepts, and remember them longer [PS58].
NA6	Teach DDL first, then integrity constraints, and finally DML [PS32; argued against in PS85]. Teach SQL before relational algebra [PS61], and introduce relational algebra only in the context of implementation and optimization to avoid students confusing relational algebra with SQL [PS61]. Regarding concurrent courses, do not teach SQL at the same time with a procedural language [PS39].
NA7	Instead of a final exam, organize intermediary assessments which can be taken after a certain number of exercises have been passed [PS70]. This helps especially weaker students [PS70]. Giving the assessments in a digital learning environment positively affects grades [PS01]. SQL skills should not be assessed through SQL code alone, but also with multiple choise questions [PS13]. Brighter students' motivation suffers if a course is not challenging enough [PS70].
NA8	Demonstrate difficult SQL concepts with animations [PS33, PS60, PS89].
NA9	Encourage students against unnecessary SQL elements, even though such omittances affect readability [PS61].
NA10	Use SQLite to teach SQL, because it is lightweight and students do not need to configure anything [PS52; argued against in PS80].

Table 10. Teaching approaches or considerations regarding patterns and visualization

ID	Teaching approach or consideration
PV1	Utilize a template to help students write more complex SQL queries [PS06, PS20, PS82]. This increases success rates [PS82], and decreases errors in the FROM and ORDER BY clauses, but not in the GROUP BY clause [PS21].
PV2	Have students plan more complex queries to ease cognitive load [PS78]. A planning notation is introduced and described [PS78]. As data demand complexity increases, *a priori* planning decreases the number of errors more and more [PS21].
PV3	Teach students how to identify certain natural language patterns (e.g., *never*, *all*, *sum*) and their corresponding SQL clauses, constructs, and keywords [PS66, PS77].
PV4	Teach SQL query formulation in steps (i.e., procedurally) [PS04, PS21, PS66, PS77]. Alternatively, introduce both procedural and set-based query formulation approaches at the start of a course. Students can choose which to use [PS65]. Procedural approach to query formulation is more natural to students, but fails at complex queries [PS65].
PV5	Visualize query execution [PS22, PS30, PS42]. It is helpful if students can visualize the query step by step, and go forward and backward, similar to programming language debuggers [PS22, PS42]. If possible present the query simultaneously visually and textually [PS39].
PV6	If students are likely to never write complex SQL, alternatives such as QBE should be considered, as it is faster to utilize, and perceived more comfortable [PS45].

Table 11. Teaching approaches or considerations regarding teacher workload

ID	Teaching approach or consideration
WL1	A list of 14 small SQL course modules is presented [PS85]. The list is too long to be presented here. SQL concepts are divided into basic, advanced, and expert level modules [PS38] These modules may be used as, for example, a structure for short online lectures [PS81].
WL2	Learning environments that allow teachers to monitor student activity, and also allow students to give feedback to the teacher [PS01] are available. Furthermore, large online learning environments with exercises and exercise databases are available without fee [PS84].
WL3	Exercise database datasets can be generated automatically [PS12, PS25, PS26], and tested against expected erroneous queries automatically [PS12]. A query should be tested against multiple datasets [PS01], and discrepancies can be used to automatically provide feedback [PS01, PS49, PS87]. Alternatively, a query's correctness can be evaluated using string metrics [PS76] or XML transformations [PS39].
WL4	As an alternative to automatic exercise database generation, students may be required to create their own exercise databases and grant appropriate privileges [PS44].
WL5	Data demands can be automatically generated based on correct SQL queries [PS47].
WL6	Utilize examinations and exercises which can be automatically graded [PS81].
WL7	Students should be given the opportunity to select themselves how complex queries they want to practice writing (query concepts, number of tables etc.), and these exercises can be automatically generated [PS35]. Furthermore, students should be allowed to choose a level of hints which the system suggests [PS50].
WL8	SQL taught through game based learning significantly increases student performance when compared to textbooks [PS75].

advise a level of caution when interpreting the reported teaching approaches in the previous section, and the number of corresponding participants Appendix C.

As the nature of opinion papers and experience reports is as their names suggest, these approaches are seldom tested in a scientific setting. As an example, Matos and Grasser [70] suggested a teaching approach for teaching relational division which is easier for students to understand. The authors report no numbers concerning how many students found the approach easier. However, by comparing the proposed teaching approach and the commonly used alternative, the benefits are apparent; in addition to being computationally faster, the approach of using GROUP BY with HAVING is arguably easier to read than multiple existence negations, at least in our opinion. In contrast, Borthick et al. [15]

studied how the database normalization level affects errors committed in query writing, and found out that end-users commit fewer errors in queries against a database adhering to the first normal form than end-users against a database adhering to the third normal form. The hypotheses were tested with 80 undergraduate and masters level students.

Based on reported quantifiable evidence supporting the views presented in these two studies, it might be compelling to advise the use of lower normal form databases over higher ones, and to dismiss the one regarding relational division. Although fewer errors might be a desirable goal to strive for, lower normal forms in database education present significant downsides. Students learn bad design practices which later need to be unlearned, the database is subject to anomalies [38], and requires more disk space due to redundancy. Finally, it is not clear whether students should strive for fewer errors, (although other database end-users arguably should), as errors are arguably an efficient way through which students learn, as argued in SQL education research [54, 100] as well as broader educational contexts [74].

6.2 Natural and Unnatural Learning Environments

A recurring theme in the primary studies, regardless of the research topic, was argumentation for [5, 8, 101, 121] and against [75, 88, 101] natural learning environments. A natural learning environment better reflects industry, i.e., students' future work environments. Environmental traits differ between workplaces, job titles, and used technologies. For the sake of discussion, we state that in a workplace there is no known correct result table for a query [5], the data demand is ambiguous [24], the datasets are complex [51], and the business domain is unfamiliar [58]. In contrast, peers are often present to offer help, use of textbooks and the internet is naturally not forbidden, and the query may be formulated as many times as necessary in a feasible timeframe. In an unnatural learning environment, these characteristics are reversed. The underlying arguments for natural environments are that students need to be prepared for their future work, and the arguments against are usually that natural environments hinder the learning of SQL (e.g., perceived confidence and success rates decrease). In teaching, these two approaches are usually mixed to varying degrees, for example, Taipalus et al. [101] report giving students the correct result table but designing exercise database data to contain no anomalies, yet Wagner et al. [110] report utilizing low quality data.

If the goal of SQL education is to prepare students to effectively work in their future work environments, learning should take place in more natural environments, and there is no need to exclusively choose a natural, mixed, or unnatural environment. SQL should first be taught in an unnatural environment [12, 14], and when the syntax and semantics are mastered to a degree, natural elements such as data demand ambiguity may be introduced gradually [108], or a natural environment used in the final exam. Naturally, grading team performance is more difficult to the teachers, and students should be prepared to work independently in their future workplace, even though help is available. We discuss natural environments more in Section 7.1.

Although Lertnattee and Pamonsinlapatham [66] argue for using SQLite due to its relatively easy configuration, teachers should be aware that SQLite 3 contains features[2] which, in our experience, confuse students. For example, in SQLite 3, data types have little meaning (strings can be stored in INT columns), some arguably important SQL concepts are not implemented (ALL, RIGHT OUTER JOIN), PRIMARY KEY does not imply NOT NULL, and strict grouping is not enforced.

[2]https://www.sqlite.org/quirks.html

6.3 Decay

In Section 3.3, we wrote that we rather conveniently chose to include SQL education research from a timeframe of 30 years. However, we advise caution when interpreting the results from the older primary studies, as these teaching considerations *decay* over time. Both the SQL standard and its implementations develop over time, as do the technologies in IT field in general. For example, a learning environment from 1990s appears naïve in terms of features, and the general look of the user interface. Some, mostly older works study the effects of a conceptual database structure representation instead of logical representation [61], while others criticize the very purpose of such a research setting [90]. More importantly, examining some older studies raises questions whether the SQL language itself has changed too much for a teaching approach to hold true anymore. This point is further emphasized with the notion that the SQL standard has never been a simple source to interpret. Three examples follow.

First, a seminal study from 1995 [95] considered "omitting the FROM clause" a semantic rather than a syntax error, even though (at least current) SQL standard considers the FROM clause mandatory in a query. Furthermore, the study demonstrated all table joins with explicit WHERE clause conditions, without the use of JOIN predicate or subqueries. This might be an educated approach, a coincidence, or resulting from the fact that these concepts were introduced in the SQL-92 standard. At least one study [90] suggests that separating expressions and joins in their respective clauses reduces some types of query formulation errors. It is unclear why Smelcer [95] demonstrates table joins using only explicit join conditions in the WHERE clause, but this is a reason to infer that the students who participated in the study were taught table joins with explicit WHERE clause conditions.

Second, another study from 1993 [119] demonstrated erroneous queries with subqueries formulated with NOT EXISTS, in which the subqueries' SELECT clauses contains multiple column names, and stated "Both cases contain errors of form. The subqueries used with EXISTS (NOT EXISTS) should use the SELECT * ... format." Nowadays, it is more of a widely accepted practice to use simple (NOT) EXISTS subquery SELECT clauses such as SELECT * or SELECT 1, but effectively it does not matter what is selected, and even division by zero is accepted by DBMSs.

Third, a study from 1988 [21] demonstrates how the aggregate function SUM handled NULL at the time. The study demonstrated how SUM would return NULL if even one of the items was NULL. Nowadays, the standard has been revised, and in most implementations, SUM handles NULL similar to zero. Rather than criticism toward the aforementioned studies, we are trying to communicate that even though the language we are using today has the same name as decades ago, SQL has undergone notable changes, and for this reason alone older studies should be given closer scrutiny.

7 FUTURE RESEARCH AGENDA

7.1 Research Dearths

Concepts beyond SELECT have received little attention in educational research. The studied query concepts [4, 6, 101], and formulated error frameworks [17, 101] focus solely on data retrieval. Intuitively, the transition from SELECT to UPDATE and DELETE is relatively easy [49], as the query concepts in the WHERE clause are the same. In terms of SQL, DCL and TCL concepts are relatively simple, and the difficulty comes from the design of privileges and transactions rather than implementation. However, DDL statements and INSERT are both a fundamental and important part of SQL which have not been studied in detail.

Advanced SQL features have not been studied in educational contexts. If we consider the SQL concepts reported in the primary studies, most of them could be based on the SQL-92 standard, and in some cases, even on SQL-89. Since then, numerous features have been added to the SQL standard, and they remain untapped from a research perspective. Such features are, for example, online analytical processing aggregate functions, the WINDOW clause, table functions, multisets, the MERGE statement, and generated columns added in SQL:1999 and SQL:2003 [47]. Additionally, SQL:2011 introduced both temporal [65] and non-temporal features [124] such as pipelined DML and enhancements to several older concepts. As we mentioned in Section 2.2, it is unclear whether knowledge about these features need to be propagated, or have they been omitted from course contents on purpose. Finally, in addition to software development, further research could also explore how SQL has extended to adjacent fields such as data science [18], broader contexts in general [44], and what types of SQL extensions have been introduced to better fit field specific needs [80].

Are unnatural environments beneficial remains an open question. Studies in which students or novices are aided by for example, automated feedback, simpler data, or unambiguous data demands achieve higher success rates in query formulation. This, however, does not necessarily reflect their future work environments. Furthermore, a recent study [99] discovered that as the logical complexity of the exercise database increases, students are less likely to succeed in query formulation. The same study, however, cautions the use of success rates alone in evaluating different teaching approaches; it is possible that although the students who fail in query formulation with a complex database, are more prepared for natural environments than students who succeed in query formulation with a simple database. Studies that test student skill in natural environments are needed, preferably so that one group of students learns SQL in an unnatural environment, and another in a natural environment, after which both groups are tested in a natural environment. Furthermore, as unnatural environments are intuitively targeted to help poor performing students, Russell and Cumming [91] raise an important concern that a certain level of simplification may impede both the learning, and the ardor towards the IT field of brighter students.

7.2 Replication

As presented in Fig. 4, there were no replication studies among the primary studies. While experience reports, opinion papers, solutions proposals, and philosophical papers are problematic to replicate due to their nature, even the most fundamental evaluation research studies [14, 34, 95] remain without replication. This is problematic, as central premises of subsequent studies are occasionally based on the results of the fundamental studies. The lack of replication studies in computing education in general has only recently received scientific attention [53]. Partly because of the lack of replication, we argue that educational SQL research is not mature enough to distinguish between patterns and anti-patterns. Moreover, a particularly insightful study by Rho and March [90] noted that some studies evaluated SQL on such a simple level, that the *ceiling effect* (i.e., variance in an independent variable is not measurable due to simplicity of the task) might explain the lack of differences in the results. Replication studies are not needed only because of reliability and the ceiling effect, but also because of obsolescence, as discussed in Section 6.3.

Beyond replication, and to uncover patterns and anti-patterns, it is crucial to evaluate proposed teaching approaches, especially those of solution proposals and opinion papers, in a scientific environment. Preferably, these evaluations should be done by researchers independent of the original authors, as it is common that reported solutions are considered helpful by the original authors. With propagation concerns [13] in mind, approaches supported by scientific evidence are likely to receive more attention among practitioners.

7.3 Building upon Existing Body of Knowledge

Based on the insights from the mapping process summarized in Fig. 2 and discussed in Section 4.1, we urge researchers to utilize and build upon existing body of knowledge in new approach proposals, and to critically evaluate all approaches.

Matos and Grasser [70] authored a study showcasing a new approach for teaching relational division. The study was published in the summer 2002 issue in JISE. Dadashzadeh [40] authored a study expanding and generalizing similar approach to other set comparison queries. This study shows relational division similarly to Matos and Grasser, and was published in the winter 2003 issue in JISE. Finally, McCann [72] authored a study presenting relational division similarly to Matos and Grasser, and this study was presented in the FIE conference in November 2003. Neither of the two latter studies cited Matos and Grasser, even though relational division is a specific concept. This might be a result of all the studies published within a short timeframe, but also due to potentially fragmented educational research fora. Regarding primary study citations in Fig. 2, is it that, for example, ACM studies in general considered top IS studies, but did not find them relevant, or is it that they did not find them? Did they not find them because of different nomenclature, or did they not utilize searches which included them? Would their research settings and conclusions have been different in this regard? As IEEE studies are seldom cited among the primary studies, we might have missed them if we did not know about IEEE beforehand. That being said, there might be relevant pockets of research that we have missed. Based on our results, we advise educational researchers and reviewers to utilize and search prior works widely, as educational considerations may be found in numerous fora.

Most of the 29 opinion papers and experience reports did not discuss potential downsides of their proposed or tested approaches, and only one [10] had a section dedicated to discussing disadvantages. We urge authors of studies of this nature to either critically evaluate their approaches, or discuss why the approach does not need critical evaluation. In comparison, even a course given as a textbook based exam (and nothing else) has positive implications, as students can study with a flexible schedule, and choose learning strategies based on their own preferences. With this in mind, one critical factor to discuss is time. Elements cannot be added to a course without expanding it or removing other elements. Expanding a course arguably has potential downsides, and, for example, sacrificing best database design practices to more efficiently teach SQL is not a desired goal in a database course. Alternatively, a teaching approach may be replaced altogether, as presented by for example, Matos and Grasser [70] and Matos et al. [71].

8 CONCLUSION

In this study, we set out to systematically map educational SQL research, and to list teaching approaches proposed in scientific literature. Our mapping shows that primary studies are published in numerous fora, not all of which are educational in nature. Recurring themes in educational SQL research are improved teaching approaches, students errors, the exercise database and related concepts, and easing teacher workload, and all types of research are represented, with the exception of replication studies. Furthermore, based on the 89 primary studies, we listed 66 teaching approaches to help educators teach SQL more efficiently. For researchers, and in addition to the systematic mapping, we proposed future research avenues, and general suggestions on how to conduct educational SQL research.

REFERENCES

[1] Alberto Abelló, Xavier Burgués, María José Casany, Carme Martín, Carme Quer, M. Elena Rodríguez, Óscar Romero, and Toni Urpí. 2016. A software tool for E-assessment of relational database skills. *International Journal of Engineering Education* 32 (2016), 1289—-1312. Issue 3. http://hdl.handle.net/2117/89668

[2] Alberto Abelló, M. Elena Rodríguez, Toni Urpí, Xavier Burgués, M. José Casany, Carme Martín, and Carme Quer. 2008. LEARN-SQL: Automatic Assessment of SQL Based on IMS QTI Specification. In *2008 Eighth IEEE International Conference on Advanced Learning Technologies*. IEEE. https://doi.org/10.1109/icalt.2008.27

[3] Elizabeth S. Adams, Mary Granger, Don Goelman, and Catherine Ricardo. 2004. Managing the introductory database course. In *Proceedings of the 35th Technical Symposium on Computer Science Education (SIGCSE)*. ACM Press. https://doi.org/10.1145/971300.971467

[4] Alireza Ahadi, Vahid Behbood, Arto Vihavainen, Julia Prior, and Raymond Lister. 2016. Students' Syntactic Mistakes in Writing Seven Different Types of SQL Queries and its Application to Predicting Students' Success. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE)*. ACM Press, New York, New York, USA, 401–406. https://doi.org/10.1145/2839509.2844640

[5] Alireza Ahadi, Julia Prior, Vahid Behbood, and Raymond Lister. 2015. A Quantitative Study of the Relative Difficulty for Novices of Writing Seven Different Types of SQL Queries. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*. ACM Press, New York, New York, USA, 201–206. https://doi.org/10.1145/2729094.2742620

[6] Alireza Ahadi, Julia Prior, Vahid Behbood, and Raymond Lister. 2016. Students' Semantic Mistakes in Writing Seven Different Types of SQL Queries. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*. ACM Press, New York, New York, USA, 272–277. https://doi.org/10.1145/2899415.2899464

[7] Judith D. Ahrens and Chetan S. Sankar. 1993. Tailoring Database Training for End Users. *MIS Quarterly* 17, 4 (1993), 419. https://doi.org/10.2307/249586

[8] A. AL-Salmi. 2018. A web-based semi-automatic assessment tool for formulating basic SQL statements: Point-and-click interaction method. In *Proceedings of the 10th International Conference on Computer Supported Education (CSEDU)*, Vol. 1. 191–198. https://doi.org/doi.org/10.5220/0006671501910198

[9] Brett Allenstein, Andrew Yost, Paul Wagner, and Joline Morrison. 2008. A Query Simulation System to Illustrate Database Query Execution. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '08)*. ACM, New York, NY, USA, 493–497. https://doi.org/10.1145/1352135.1352301

[10] W.J. Amadio. 2003. The dilemma of Team Learning: An assessment from the SQL programming classroom. In *Proceedings of the Annual Meeting of the Decision Sciences Institute*. 823–828. https://www.scopus.com/inward/record.uri?eid=2-s2.0-0442278473&partnerID=40&md5=ffb48d535e80751aaf8b153af6a13c42

[11] Neal Ashkanasy, Paul L. Bowen, Fiona H. Rohde, and Chiu Yueh Alice Wu. 2007. The Effects of User Characteristics on Query Performance in the Presence of Information Request Ambiguity. *Journal of Information Systems* 21, 1 (2007), 53–82. https://doi.org/10.2308/jis.2007.21.1.53

[12] Micheal Axelsen, A Faye Borthick, and Paul L. Bowen. 2001. A model for and the effects of information request ambiguity on end-user query performance. *ICIS 2001 Proceedings* (2001), 68. http://aisel.aisnet.org/icis2001/68

[13] Heather Bort, David P. Bunde, Zack Butler, Christopher Lynnly Hovey, and Cynthia Taylor. 2019. Propagating Educational Innovations. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. ACM, New York, NY, USA, 167–168. https://doi.org/10.1145/3287324.3287526

[14] A.Faye Borthick, Paul L. Bowen, Donald R. Jones, and Michael Hung Kam Tse. 2001. The effects of information request ambiguity and construct incongruence on query development. *Decision Support Systems* 32, 1 (nov 2001), 3–25. https://doi.org/10.1016/s0167-9236(01)00097-5

[15] A. Faye Borthick, Paul L. Bowen, S.T Liew, and Fiona H. Rohde. 2001. The effects of normalization on end-user query errors: An experimental evaluation. *International Journal of Accounting Information Systems* 2, 4 (2001), 195 – 221. https://doi.org/10.1016/S1467-0895(01)00023-9

[16] Paul L. Bowen, Robert A. O'Farrell, and Fiona H. Rohde. 2009. An Empirical Investigation of End-User Query Development: The Effects of Improved Model Expressiveness vs. Complexity. *Information Systems Research* 20, 4 (dec 2009), 565–584. https://doi.org/10.1287/isre.1080.0181

[17] Stefan Brass and Christian Goldberg. 2006. Semantic errors in SQL queries: A quite complete list. *Journal of Systems and Software* 79, 5 (may 2006), 630–644. https://doi.org/10.1016/j.jss.2005.06.028

[18] Jennifer E. Broatch, Suzanne Dietrich, and Don Goelman. 2019. Introducing Data Science Techniques by Connecting Database Concepts and dplyr. *Journal of Statistics Education* 27, 3 (2019), 147–153. https://doi.org/10.1080/10691898.2019.1647768 arXiv:https://doi.org/10.1080/10691898.2019.1647768

[19] Peter Brusilovsky, Sergey Sosnovsky, Danielle H. Lee, Michael Yudelson, Vladimir Zadorozhny, and Xin Zhou. 2008. An Open Integrated Exploratorium for Database Courses. In *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '08)*. ACM, New York, NY, USA, 22–26. https://doi.org/10.1145/1384271.1384280

[20] Peter Brusilovsky, Sergey Sosnovsky, Michael V. Yudelson, Danielle H. Lee, Vladimir Zadorozhny, and Xin Zhou. 2010. Learning SQL Programming with Interactive Tools: From Integration to Personalization. *ACM Transactions on Computing Education* 9, 4, Article 19 (Jan. 2010), 15 pages. https://doi.org/10.1145/1656255.1656257

[21] R. B. Buitendijk. 1988. Logical errors in database SQL retrieval queries. *Computer Science in Economics and Management* 1, 2 (1988), 79–96. https://doi.org/10.1007/BF00427157

[22] Clifford G. Burgess. 1991. A graphical, database-querying interface for casual, naive computer users. *International Journal of Man-Machine Studies* 34, 1 (1991), 23–47. https://doi.org/10.1016/0020-7373(91)90049-d

[23] Luca Cagliero, Luigi De Russis, Laura Farinetti, and Teodoro Montanaro. 2018. Improving the Effectiveness of SQL Learning Practice: A Data-Driven Approach. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Vol. 01. 980–989. https://doi.org/10.1109/COMPSAC.2018.00174

[24] Gretchen Irwin Casterella and Leo Vijayasarathy. 2013. An Experimental Investigation of Complexity in Database Query Formulation Tasks. *Journal of Information Systems Education* 24, 3 (2013), 211–221. http://jise.org/Volume24/24-3/pdf/Vol24-3pg211.pdf

[25] Gretchen Irwin Casterella and Leo Vijayasarathy. 2019. Query Structure and Data Model Mapping Errors in Information Retrieval Tasks. *Journal of Information Systems Education* 30, 3 (2019), 178–190. http://jise.org/Volume30/n3/JISEv30n3p178.pdf

[26] Tiziana Catarci, Maria F. Costabile, Stefano Levialdi, and Carlo Batini. 1997. Visual Query Systems for Databases: A Survey. *Journal of Visual Languages & Computing* 8, 2 (1997), 215–260. https://doi.org/10.1006/jvlc.1997.0037

[27] Claudio Cerullo and Marco Porta. 2007. A System for Database Visual Querying and Query Visualization: Complementing Text and Graphics to Increase Expressiveness. In *18th International Conference on Database and Expert Systems Applications (DEXA 2007)*. IEEE. https://doi.org/10.1109/dexa.2007.91

[28] Donald D. Chamberlin. 2012. Early History of SQL. *IEEE Annals of the History of Computing* 34, 4 (Oct 2012), 78–82. https://doi.org/10.1109/MAHC.2012.61

[29] Donald D. Chamberlin, M. M. Astrahan, K. P. Eswaran, P. P. Griffiths, R. A. Lorie, J. W. Mehl, P. Reisner, and B. W. Wade. 1976. SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control. *IBM Journal of Research and Development* 20, 6 (Nov 1976), 560–575. https://doi.org/10.1147/rd.206.0560

[30] Donald D. Chamberlin and Raymond F. Boyce. 1974. SEQUEL: A Structured English Query Language. In *Proceedings of the 1974 ACM SIGFIDET (Now SIGMOD) Workshop on Data Description, Access and Control (SIGFIDET '74)*. Association for Computing Machinery, New York, NY, USA, 249–264. https://doi.org/10.1145/800296.811515

[31] Hock Chan, Keng Siau, and Kwok-Kee Wei. 1997. The effect of data model, system and task characteristics on user query performance: an empirical study. *ACM SIGMIS Database: the DATABASE for Advances in Information Systems* 29, 1 (1997), 31–49. https://doi.org/10.1145/506812.506820

[32] Hock Chuan Chan. 2007. A two-stage evaluation of user query performance for the relational model and SQL. *PACIS 2007 Proceedings* (2007), 118. https://aisel.aisnet.org/pacis2007/118

[33] Hock Chuan Chan, Hock-Hai Teo, and XH Zeng. 2005. An evaluation of novice end-user computing performance: Data modeling, query writing, and comprehension. *Journal of the American Society for Information Science and Technology* 56, 8 (2005), 843–853. https://onlinelibrary.wiley.com/doi/full/10.1002/asi.20178

[34] Hock Chuan Chan, Kwok Kee Wei, and Keng Leng Siau. 1993. User-Database Interface: The Effect of Abstraction Levels on Query Performance. *MIS Quarterly* 17, 4 (Dec 1993), 441. https://doi.org/10.2307/249587

[35] Hock Chuan Chan, Kwok Kee Wei, and Keng Leng Siau. 1994. An empirical study on end-users' update performance for different abstraction levels. *International Journal of Human-Computer Studies* 41, 3 (1994), 309 – 328. https://doi.org/10.1006/ijhc.1994.1061

[36] H. C. Chan, K. K. Wei, and K. L. Siau. 1995. The effect of a database feedback system on user performance. *Behaviour & Information Technology* 14, 3 (May 1995), 152–162. https://doi.org/10.1080/01449299508914642

[37] Peter Pin-Shan Chen. 1976. The Entity-relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems* 1, 1 (March 1976), 9–36. https://doi.org/10.1145/320434.320440

[38] E. F. Codd. 1970. A Relational Model of Data for Large Shared Data Banks. *Commun. ACM* 13, 6 (June 1970), 377–387. https://doi.org/10.1145/362384.362685

[39] Thomas Connolly and Carolyn Begg. 2015. *Database Systems (6th. ed.)*. Pearson.

[40] Mohammad Dadashzadeh. 2003. A Simpler Approach to Set Comparison Queries in SQL. *Journal of Information Systems Education* 14, 4 (2003), 345–348. http://jise.org/Volume14/n4/JISEv14n4p345.pdf

[41] Mohammad Dadashzadeh. 2007. Teaching Tip: Recursive Joins to Query Data Hierarchies in Microsoft Access. *Journal of Information Systems Education* 18, 1 (2007), 5–10. http://jise.org/Volume18/n1/JISEv18n1p5.pdf

[42] Mohammad Dadashzadeh. 2007. Teaching Tip: Specification and Enforcement of Semantic Integrity Constraints in Microsoft Access. *Journal of Information Systems Education* 18, 4 (2007), 393–398. http://jise.org/Volume18/n4/JISEv18n4p393.pdf

[43] J. Steve Davis. 1990. Experimental investigation of the utility of data structure and E-R diagrams in database query. *International Journal of Man-Machine Studies* 32, 4 (1990), 449 – 459. https://doi.org/10.1016/S0020-7373(05)80142-7

[44] S. W. Dietrich, D. Goelman, C. M. Borror, and S. M. Crook. 2015. An Animated Introduction to Relational Databases for Many Majors. *IEEE Transactions on Education* 58, 2 (May 2015), 81–89. https://doi.org/10.1109/TE.2014.2326834

[45] Doug Downey, Susan Dumais, Dan Liebling, and Eric Horvitz. 2008. Understanding the Relationship Between Searchers' Queries and Information Goals. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*. ACM, New York, NY, USA, 449–458. https://doi.org/10.1145/1458082.1458143

[46] A. Efendioglu and T. Yanpar Yelken. 2010. Programmed instruction versus meaningful learning theory in teaching basic structured query language (SQL) in computer lesson. *Computers and Education* 55, 3 (2010), 1287–1299. https://doi.org/10.1016/j.compedu.2010.05.026 cited By 9.

[47] Andrew Eisenberg, Jim Melton, Krishna Kulkarni, Jan-Eike Michels, and Fred Zemke. 2004. SQL:2003 has been published. *ACM SIGMOD Record* 33, 1 (mar 2004), 119. https://doi.org/10.1145/974121.974142

[48] Ramez Elmasri and Shamkant B. Navathe. 2016. *Fundamentals of Database Systems (7th. ed.)*. Pearson.

[49] Philip Garner and John Amedeo Mariani. 2015. Learning SQL in steps. *Journal of Systemics, Cybernetics and Informatics* 13, 4 (2015), 19–24.

[50] Don Goelman. 2008. Databases, non-majors and collaborative learning. In *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE)*. ACM Press. https://doi.org/10.1145/1384271.1384281

[51] Venkat N. Gudivada, Jagadeesh Nandigam, and Yonglei Tao. 2007. Enhancing student learning in database courses with large data sets. In *2007 37th annual frontiers in education conference (FIE)*. IEEE. https://doi.org/10.1109/fie.2007.4418135

[52] Mario Guimaraes. 2005. Integrating the Kennesaw Database Courseware and Other Database Coursewares in Database Classes. In *Proceedings of the 43rd Annual Southeast Regional Conference - Volume 1 (ACM-SE 43)*. ACM, New York, NY, USA, 15–15. https://doi.org/10.1145/1167350.1167362

[53] Qiang Hao, David H. Smith IV, Naitra Iriumi, Michail Tsikerdekis, and Andrew J. Ko. 2019. A Systematic Investigation of Replications in Computing Education Research. *ACM Transactions on Computing Education* 19, 4, Article Article 42 (Aug. 2019), 18 pages. https://doi.org/10.1145/3345328

[54] Joseph E. Hollingsworth. 2008. Teaching Query Writing: An Informed Instruction Approach. In *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '08)*. ACM, New York, NY, USA, 351–351. https://doi.org/10.1145/1384271.1384393

[55] A. S. M. L. Hoque, G. M. M. Bashiry, and M. R. Uddin. 2014. Equivalence of Problems in Problem Based e-Learning of Database. In *2014 IEEE Sixth International Conference on Technology for Education*. 106–109. https://doi.org/10.1109/T4E.2014.23

[56] Hsiu-Fang Hsieh and Sarah E. Shannon. 2005. Three Approaches to Qualitative Content Analysis. *Qualitative Health Research* 15, 9 (2005), 1277–1288. https://doi.org/10.1177/1049732305276687 arXiv:https://doi.org/10.1177/1049732305276687 PMID: 16204405.

[57] Ching-yu Huang. 2019. Integrative Curriculum for Teaching Databases. *Journal of Computing Sciences in Colleges* 34, 3 (Jan. 2019), 131–131. http://dl.acm.org/citation.cfm?id=3306465.3306487

[58] Gretchen Irwin, Wessel Lark, and Harvey Blackburn. 2012. Teaching Case: The Animal Genetic Resource Information Network (AnimalGRIN) Database: A Database Design & Implementation Case. *Journal of Information Systems Education* 23, 1 (2012), 19–28. http://jise.org/Volume23/n1/JISEv23n1p19.pdf

[59] ISO/IEC. 2016. ISO/IEC 9075-1:2016, "SQL - Part 1: Framework". https://www.iso.org/standard/63555.html

[60] ISO/IEC. 2016. ISO/IEC 9075-2:2016, "SQL - Part 2: Foundation". https://www.iso.org/standard/63556.html

[61] W.J. Kenny Jih, David A. Bradbard, Charles A. Snyder, and Nancy G.A. Thompson. 1989. The effects of relational and entity-relationship data models on query performance of end users. *International Journal of Man-Machine Studies* 31, 3 (Sep 1989), 257–267. https://doi.org/10.1016/0020-7373(89)90007-2

[62] Joint Task Force on Computing Curricula, Association for Computing Machinery (ACM) and IEEE Computer Society. 2013. *Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*. Technical Report. New York, NY, USA. https://doi.org/10.1145/2534860

[63] Nenad Jukic and Paul Gray. 2008. Using Real Data to Invigorate Student Learning. *SIGCSE Bulletin* 40, 2 (June 2008), 6–10. https://doi.org/10.1145/1383602.1383604

[64] Jai W. Kang, Qi Yu, Edward P. Holden, and Xumin Liu. 2019. Complementing Course Contents Between IT/CS: A Case Study on Database Courses. In *Proceedings of the 20th Annual SIG Conference on Information Technology Education (SIGITE '19)*. ACM, New York, NY, USA, 10–15. https://doi.org/10.1145/3349266.3351414

[65] Krishna Kulkarni and Jan-Eike Michels. 2012. Temporal features in SQL:2011. *ACM SIGMOD Record* 41, 3 (oct 2012), 34. https://doi.org/10.1145/2380776.2380786

[66] Verayuth Lertnattee and Perayot Pamonsinlapatham. 2017. Blended Learning for Improving Flexibility of Learning Structure Query Language (SQL). In *International Conference on Blended Learning (ICBL)*. Springer, 343–353. https://doi.org/10.1007/978-3-319-59360-9_30

[67] M. Luo. 2010. Study on design of graduation Practical Teaching management system based on SQL Server 2008 for Vocational College. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, Vol. 2. V2–292–V2–296. https://doi.org/10.1109/ICCASM.2010.5620476

[68] Maslin Masrom, Halimah Hasan, and Habsah Abdullah. 2007. Using a team-based approach in teaching database course. In *National Conference on Programming Science (ATUR 07)*, Vol. 5.

[69] Ramon A. Mata-Toledo and Carlos A. Reyes-Garcia. 2002. A Model Course for Teaching Database Administration with Personal Oracle 8i. *Journal of Computing Sciences in Colleges* 17, 3 (Feb. 2002), 125–130. http://dl.acm.org/citation.cfm?id=772636.772658

[70] Victor M. Matos and Rebecca Grasser. 2002. Teaching Tip A Simpler (and Better) SQL Approach to Relational Division. *Journal of Information Systems Education* 13, 2 (2002), 85–88. http://jise.org/Volume13/Pdf/085.pdf

[71] Victor M. Matos, Rebecca Grasser, and Paul Jalics. 2006. The Case of the Missing Tuple: Teaching the SQL Outer-join Operator to Undergraduate Information Systems Students. *Journal of Computing Sciences in Colleges* 22, 1 (Oct. 2006), 23–32. http://dl.acm.org/citation.cfm?id=1181811.1181814

[72] L. I. McCann. 2003. On making relational division comprehensible. In *Proceedings of the 2003 33rd Annual Frontiers in Education Conference (FIE)*, Vol. 2. F2C–6. https://doi.org/10.1109/FIE.2003.1264699

[73] Kirby McMaster, Samuel Sambasivam, Steven Hadfield, and Stuart Wolthuis. 2013. Relational Algebra and SQL: Better Together. *Information Systems Education Journal* 11, 1 (2013), 4–13. http://isedj.org/2013-11/N1/ISEDJv11n1p4.pdf

[74] Janet Metcalfe. 2017. Learning from Errors. *Annual Review of Psychology* 68, 1 (2017), 465–489. https://doi.org/10.1146/annurev-psych-010416-044022 PMID: 27648988.

[75] Zhengjie Miao, Sudeepa Roy, and Jun Yang. 2019. Explaining Wrong Queries Using Small Examples. In *Proceedings of the 2019 ACM Conference on Management of Data (SIGMOD) (SIGMOD '19)*. ACM, New York, NY, USA, 503–520. https://doi.org/10.1145/3299869.3319866

[76] Antonija Mitrovic. 1998. Learning SQL with a Computerized Tutor. In *Proceedings of the Twenty-ninth SIGCSE Technical Symposium on Computer Science Education (SIGCSE '98)*. ACM, New York, NY, USA, 307–311. https://doi.org/10.1145/273133.274318

[77] Antonija Mitrovic. 2003. An intelligent SQL tutor on the web. *International Journal of Artificial Intelligence in Education* 13, 2-4 (2003), 173–197. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85013603633&partnerID=40&md5=7e8de8bf8e5659c6720ec83f64e2e388

[78] Antonija Mitrovic. 2006. Large-Scale Deployment of three intelligent web-based database tutors. *Journal of Computing and Information Technology* 14, 4 (2006), 275–281. https://doi.org/10.2498/cit.2006.04.02

[79] Keith Moss. 2010. DataBase teaching tools. In *IEEE EDUCON 2010 Conference*. IEEE. https://doi.org/10.1109/educon.2010.5492432

[80] Barzan Mozafari, Kai Zeng, and Carlo Zaniolo. 2010. K*SQL: A Unifying Engine for Sequence Patterns and XML. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD '10)*. Association for Computing Machinery, New York, NY, USA, 1143–1146. https://doi.org/10.1145/1807167.1807302

[81] David Olsen and Karina Hauser. 2007. Teaching Tip: Teaching Advanced SQL Skills: Text Bulk Loading. *Journal of Information Systems Education* 18, 4 (2007), 399–402. http://jise.org/Volume18/n4/JISEv18n4p399.pdf

[82] T. Permpool, S. Nalintippayawong, and K. Atchariyachanvanich. 2019. Interactive SQL Learning Tool with Automated Grading using MySQL Sandbox. In *2019 IEEE 6th International Conference on Industrial Engineering and Applications, ICIEA 2019*. 928–932. https://doi.org/10.1109/IEA.2019.8715175

[83] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. 2008. Systematic Mapping Studies in Software Engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering (EASE'08)*. BCS Learning & Development Ltd., Swindon, UK, 68–77. http://dl.acm.org/citation.cfm?id=2227115.2227123

[84] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology* 64 (2015), 1 – 18. https://doi.org/10.1016/j.infsof.2015.03.007

[85] Hongsiri Piyayodilokchai, Pintip Ruenwongsa, Watcharee Ketpichainarong, Parames Laosinchai, and Patcharin Panjaburee. 2011. Promoting Students' Understanding of SQL in a Database Management Course: A Learning Cycle Approach. *The International Journal of Learning: Annual Review* 17, 11 (2011), 325–338. https://doi.org/10.18848/1447-9494/cgp/v17i11/47345

[86] Julia Coleman Prior. 2003. Online Assessment of SQL Query Formulation Skills. In *Proceedings of the Fifth Australasian Conference on Computing Education - Volume 20 (ACE '03)*. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, 247–256. http://dl.acm.org/citation.cfm?id=858403.858433

[87] Gary B. Randolph. 2003. The Forest and the Trees: Using Oracle and SQL Server Together to Teach ANSI-standard SQL. In *Proceedings of the 4th ACM Conference on Information Technology Curriculum (CITC) (CITC4 '03)*. ACM, New York, NY, USA, 234–236. https://doi.org/10.1145/947121.947174

[88] Christine F. Reilly. 2018. Experience with Active Learning and Formative Feedback for a SQL Unit. In *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE. https://doi.org/10.1109/fie.2018.8659173

[89] Karen Renaud and Judy Van Biljon. 2004. Teaching SQL - Which Pedagogical Horse for This Course?. In *British National Conference on Databases (BNCOD)*. Springer, 244–256. https://doi.org/10.1007/978-3-540-27811-5_22

[90] Sangkyu Rho and Salvatore T March. 1997. An analysis of semantic overload in database access systems using multi-table query formulation. *Journal of Database Management* 8, 2 (1997), 3–15. https://www.igi-global.com/gateway/article/51176

[91] Gordon Russell and Andrew Cumming. 2004. Improving the Student Learning Experience for SQL Using Automatic Marking. In *Cognition and Exploratory Learning in Digital Age (CELDA)*. 281–288.

[92] Edward Sciore. 2007. SimpleDB: A Simple Java-based Multiuser System for Teaching Database Internals. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '07)*. ACM, New York, NY, USA, 561–565. https://doi.org/10.1145/1227310.1227498

[93] Bilal Shebaro. 2018. Using Active Learning Strategies in Teaching Introductory Database Courses. *Journal of Computing Sciences in Colleges* 33, 4 (April 2018), 28–36. http://dl.acm.org/citation.cfm?id=3199572.3199576

[94] Yasin N. Silva, Isadora Almeida, and Michell Queiroz. 2016. SQL: From Traditional Databases to Big Data. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE) (SIGCSE '16)*. ACM, New York, NY, USA, 413–418. https://doi.org/10.1145/2839509.2844560

[95] John B. Smelcer. 1995. User errors in database query composition. *International Journal of Human-Computer Studies* 42, 4 (Apr 1995), 353–381. https://doi.org/10.1006/ijhc.1995.1017

[96] M. Soflano, T.M. Connolly, and T. Hainey. 2015. Learning style analysis in adaptive GBL application to teach SQL. *Computers and Education* 86 (2015), 105–119. https://doi.org/10.1016/j.compedu.2015.02.009

[97] Song Jian-gong. 2010. Design and application of collaborative learning system based on web to database experiment teaching. In *2010 International Conference on Educational and Information Technology*, Vol. 3. V3–140–V3–143. https://doi.org/10.1109/ICEIT.2010.5608405

[98] Toni Taipalus. 2019. Teaching Tip: A Notation for Planning SQL Queries. *Journal of Information Systems Education* 30, 3 (2019), 160–166. http://jise.org/Volume30/n3/JISEv30n3p160.pdf

[99] Toni Taipalus. 2020. The effects of database complexity on SQL query formulation. *Journal of Systems and Software* 165 (2020), 110576. https://doi.org/10.1016/j.jss.2020.110576

[100] Toni Taipalus and Piia Perälä. 2019. What to Expect and What to Focus on in SQL Query Teaching. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE) (SIGCSE '19)*. ACM, New York, NY, USA, 198–203. https://doi.org/10.1145/3287324.3287359

[101] Toni Taipalus, Mikko Siponen, and Tero Vartiainen. 2018. Errors and Complications in SQL Query Formulation. *ACM Transactions on Computing Education* 18, 3, Article 15 (Aug. 2018), 29 pages. https://doi.org/10.1145/3231712

[102] Josh Tenenberg and Robert McCartney. 2010. Why Discipline Matters in Computing Education Scholarship. *ACM Transactions on Computing Education* 9, 4 (2010), 1–7. https://doi.org/10.1145/1656255.1656256

[103] The Joint Task Force on Computing Curricula. 2015. *Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. Technical Report. New York, NY, USA. https://dl.acm.org/citation.cfm?id=2965631

[104] Richard C. Thomas and Rebecca Mancy. 2004. Use of large databases for group projects at the nexus of teaching and research. In *Proceedings of the 9th annual conference on Innovation and technology in computer science education (ITiCSE)*. ACM Press. https://doi.org/10.1145/1007996.1008039

[105] Heikki Topi, Kate M. Kaiser, Janice C. Sipior, Joseph S. Valacich, J. F. Nunamaker, Jr., G. J. de Vreede, and Ryan Wright. 2010. *Curriculum Guidelines for Undergraduate Degree Programs in Information Systems*. Technical Report. New York, NY, USA. https://www.acm.org/binaries/content/assets/

education/curricula-recommendations/is-2010-acm-final.pdf
[106] Heikki Topi, Joseph S Valacich, and Jeffrey A Hoffer. 2005. The effects of task complexity and time availability limitations on human performance in database query tasks. *International Journal of Human-Computer Studies* 62, 3 (2005), 349–379. https://doi.org/10.1016/j.ijhcs.2004.10.003
[107] Susan D. Urban and Suzanne W. Dietrich. 2001. Advanced Database Concepts for Undergraduates: Experience with Teaching a Second Course. *SIGCSE Bull.* 33, 1 (Feb. 2001), 357–361. https://doi.org/10.1145/366413.364648
[108] Leo Vijayasarathy and Gretchen Casterella. 2016. The Effects of Information Request Language and Template Usage on Query Formulation. *Journal of the Association for Information Systems* 17, 10 (Oct 2016), 674–707. https://doi.org/10.17705/1jais.00440
[109] Adam H. Villa. 2016. Big Data: Motivating the Development of an Advanced Database Systems Course. *Journal of Computing Sciences in Colleges* 31, 3 (2016), 119–128. https://dl.acm.org/doi/abs/10.5555/2835377.2835395
[110] Paul J. Wagner, Elizabeth Shoop, and John V. Carlis. 2003. Using Scientific Data to Teach a Database Systems Course. In *Proceedings of the 34th ACM Technical Symposium on Computer Science Education (SIGCSE) (SIGCSE '03)*. ACM, New York, NY, USA, 224–228. https://doi.org/10.1145/611892.611975
[111] R. T. Watson. 2006. The Essential Skills of Data Modeling. *Journal of Information Systems Education* 17, 1 (2006), 39–42. http://jise.org/Volume17/n1/JISEv17n1p39.pdf
[112] Roel Wieringa, Neil Maiden, Nancy Mead, and Colette Rolland. 2005. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Engineering* 11, 1 (nov 2005), 102–107. https://doi.org/10.1007/s00766-005-0021-6
[113] Coleen R. Wilder and Ceyhun O. Ozgur. 2015. Business Analytics Curriculum for Undergraduate Majors. *INFORMS Transactions on Education* 15, 2 (Jan. 2015), 180–187. https://doi.org/10.1287/ited.2014.0134
[114] Claes Wohlin. 2014. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14)*. ACM, New York, NY, USA, Article 38, 10 pages. https://doi.org/10.1145/2601248.2601268
[115] Claes Wohlin, Per Runeson, Paulo Anselmo da Mota Silveira Neto, Emelie Engström, Ivan do Carmo Machado, and Eduardo Santana de Almeida. 2013. On the reliability of mapping studies in software engineering. *Journal of Systems and Software* 86, 10 (2013), 2594 – 2610. https://doi.org/10.1016/j.jss.2013.04.076
[116] David A. Wolff. 2001. MySQL, PostgreSQL, and PHP: Open Source Technologies for a Database Management Course. *Journal of Computing Sciences in Colleges* 17, 2 (Dec. 2001), 91–92. http://dl.acm.org/citation.cfm?id=775339.775358
[117] Donna Wright. 2013. Database: CSI, Inc. *Journal of Computing Sciences in Colleges* 28, 5 (May 2013), 80–81. http://dl.acm.org/citation.cfm?id=2458569.2458586
[118] Wu Da-sheng and Wu Sheng-yu. 2010. Dynamically maintain the teaching examples of triggers and stored procedures about the course of database application. In *2010 2nd International Conference on Education Technology and Computers*, Vol. 1. V1–525–V1–527. https://doi.org/10.1109/ICETC.2010.5529193
[119] M. Y. . Yen and R. W. Scamell. 1993. A human factors experimental comparison of SQL and QBE. *IEEE Transactions on Software Engineering* 19, 4 (April 1993), 390–409. https://doi.org/10.1109/32.223806
[120] M.-H. Ying, N.-W. Chi, and Y. Hong. 2012. Enhancement of learning by using an Online SQL Learning System with Automatic Checking Mechanism. *International Journal of Digital Content Technology and its Applications* 6, 6 (Apr 2012), 239–248. https://doi.org/10.4156/jdcta.vol6.issue6.28
[121] Kwok-Bun Yue. 2013. Using a Semi-Realistic Database to Support a Database Course. *Journal of Information Systems Education* 24, 4 (2013), 327–336. http://jise.org/Volume24/n4/JISEv24n4p327.pdf
[122] Liu Yuelan, Liao Yiwei, Huang Yuyan, and Liu Yuefan. 2011. Study on Teaching Methods of Database Application Courses. *Procedia Engineering* 15 (2011), 5425–5428. https://doi.org/10.1016/j.proeng.2011.08.1006
[123] Abe Zeid and Sagar Kamarthi. 2008. Best teaching practices in database courses for engineering students. *International Journal of Engineering Education* 24, 5 (2008), 980–989.
[124] Fred Zemke. 2012. What's new in SQL:2011. *ACM SIGMOD Record* 41, 1 (2012), 67–73. https://doi.org/10.1145/2206869.2206883

A LIST OF PRIMARY STUDIES

Selected primary studies

ID	Publication
PS01	Abelló, A., Burgués, X., Casany, M.J., Martín, C., Quer, C., Rodríguez, M.E., Romero, Ó., & Urpí, T. (2016). A software tool for e-assessment of relational database skills. *International Journal of Engineering Education 32*(3), 1289-1323. http://hdl.handle.net/2117/89668
PS02	Ahadi, A., Behbood, V., Vihavainen, A., Prior, J., & Lister, R. (2016). Students' Syntactic Mistakes in Writing Seven Different Types of SQL Queries and its Application to Predicting Students' Success. In *Proceedings of the 47th ACM Technical Symposium on Computer Science Education (SIGCSE)* (pp. 401–406). ACM. doi.org/doi:10.1145/2839509.2844640
PS03	Ahadi, A., Prior, J., Behbood, V., & Lister, R. (2015). A Quantitative Study of the Relative Difficulty for Novices of Writing Seven Different Types of SQL Queries. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)* (pp. 201—206). ACM. doi.org/doi:10.1145/2729094.2742620
PS04	Ahadi, A., Prior, J., Behbood, V., & Lister, R. (2016). Students' Semantic Mistakes in Writing Seven Different Types of SQL Queries. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)* (pp. 272—277). ACM. doi.org/doi:10.1145/2899415.2899464
PS05	AL-Salmi, A. (2018). A web-based semi-automatic assessment tool for formulating basic SQL statements: Point-and-click interaction method. In *Proceedings of the 10th International Conference on Computer Supported Education (CSEDU)* (pp. 191–198). doi.org/10.5220/0006671501910198
PS06	Al-Shuaily, H., & Renaud, K. (2010). SQL patterns - a new approach for teaching SQL. In *8th HEA Workshop on Teaching, Learning and Assessment of Databases (TLAD)* (pp. 29–40). rke.abertay.ac.uk/ws/portalfiles/portal/9116027/TLAD2010Proceedings.pdf
PS07	Allen, G. N., & March, S. T. (2006). The effects of state-based and event-based data representation on user performance in query formulation tasks. *MIS Quarterly 30*(2), 269–290. doi.org/10.2307/25148731
PS08	Allen, G. N., & Parsons, J. (2010). Is query reuse potentially harmful? Anchoring and adjustment in adapting existing database queries. *Information Systems Research 21*(1), 56–77. doi.org/10.1287/isre.1080.0189
PS09	Amadio, W. (2003). The dilemma of team learning: An assessment from the SQL programming classroom. In *Proceedings of the Annual Meeting of the Decision Sciences Institute* (pp. 823—828).
PS10	Aversano, L., Canfora, G., De Lucia, A., & Stefanucci, S. (2002). Understanding SQL through iconic interfaces. In *Proceedings of the IEEE 26th Annual International Computer Software and Applications (COMPSAC)* (pp. 703-–708). IEEE. doi.org/10.1109/CMPSAC.2002.1045084
PS11	Axelsen, M., Borthick, A. F., & Bowen, P. L. (2001). A model for and the effects of information request ambiguity on end-user query performance. *ICIS 2001 Proceedings*, p.68. aisel.aisnet.org/icis2001/68
PS12	Bhangdiya, A., Chandra, B., Kar, B., Radhakrishnan, B., Reddy, K.V.M., Shah, S., & Sudarshan, S. (2015). The XDa-TA system for automated grading of SQL query assignments. In *Proceedings of the 2015 IEEE 31st International Conference on Data Engineering (ICDE)* (pp. 1468—1471). IEEE. doi.org/10.1109/icde.2015.7113403
PS13	Boisvert, C., Domdouzis, K., & License, J. (2018). A comparative analysis of student SQL and relational database knowledge using automated grading tools. In *Proceedings of the 2018 IEEE International Symposium on Computers in Education (SIIE)*. IEEE. doi.org/10.1109/siie.2018.8586684
PS14	Borthick, A., Bowen, P. L., Jones, D. R., & Tse, M. H. K. (2001). The effects of information request ambiguity and construct incongruence on query development. *Decision Support Systems 32*(1), 3-25. doi.org/10.1016/s0167-9236(01)00097-5
PS15	Bowen, P., O'Farrell, R., & Rohde, F. (2004). How does your model grow? an empirical investigation of the effects of ontological clarity and application domain size on query performance. *ICIS 2004 Proceedings*, p.7. aisel.aisnet.org/icis2004/7/
PS16	Bowen, P. L., O'Farrell, R. A., & Rohde, F. H. (2006). Analysis of competing data structures: Does ontological clarity produce better end user query performance. *Journal of the Association for Information Systems 7*(22), 514-544. doi.org/10.17705/1jais.00098
PS17	Bowen, P. L., O'Farrell, R. A., & Rohde, F. H. (2009). An Empirical Investigation of End-User Query Development: The Effects of Improved Model Expressiveness vs. Complexity. *Information Systems Research 20*(4), 565–584. doi.org/10.1287/isre.1080.0181
PS18	Brass, S., & Goldberg, C. (2006). Semantic errors in SQL queries: A quite complete list. *Journal of Systems and Software 79*(5), 630-644. doi.org/10.1016/j.jss.2005.06.028
PS19	Caldeira, C. P. (2008). Teaching SQL: A case study. In *Proceedings of the 2008 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)* (pp. 340-–340). ACM. doi.org/10.1145/1384271.1384382
PS20	Casterella, G.I., & Vijayasarathy, L. (2013). An Experimental Investigation of Complexity in Database Query Formulation Tasks. *Journal of Information Systems Education 24*(3), 211-221. http://jise.org/Volume24/24-3/pdf/Vol24-3pg211.pdf
PS21	Casterella, G.I., & Vijayasarathy, L. (2019). Query Structure and Data Model Mapping Errors in Information Retrieval Tasks. *Journal of Information Systems Education 30*(3), 178-190. http://jise.org/Volume30/n3/JISEv30n3p178.pdf
PS22	Cembalo, M., De Santis, A., & Ferraro Petrillo, U. (2011). SAVI: A New System for Advanced SQL Visualization. In *Proceedings of the 2011 ACM Conference on Information Technology Education (SIGITE)* (pp. 165–170). ACM. doi.org/10.1145/2047594.2047641
PS23	Chan, H.C., Tan, B.C.Y., & Wei, K.K. (1999). Three important determinants of user performance for database retrieval. *International Journal of Human-Computer Studies 51*(5), 895-918. doi.org/10.1006/ijhc.1999.0272
PS24	Chan, H.C., Wei, K.K., & Siau, K.L. (1993). User-Database Interface: The Effect of Abstraction Levels on Query Performance. *MIS Quarterly 17*(4), 441. doi.org/10.2307/249587
PS25	Chandra, B., Chawda, B., Kar, B., Reddy, K.V., Shah, S., & Sudarshan, S. (2015). Data Generation for Testing and Grading SQL Queries. *The VLDB Journal 24*(6), 731-755. doi.org/10.1007/s00778-015-0395-0
PS26	Chandra, B., Mathew, J., Radhakrishnan, B., Acharya, S., & Sudarshan, S. (2016). Partial Marking for Automated Grading of SQL Queries. In *Proceedings of the VLDB Endowment 9*(13) (pp. 1541–1544). doi.org/10.14778/3007263.3007304
PS27	Dadashzadeh, M. (2003). Teaching Tip: A Simpler Approach to Set Comparison Queries in SQL. *Journal of Information Systems Education 14*(4), 345-348. http://jise.org/Volume14/n4/JISEv14n4p345.pdf
PS28	Dadashzadeh, M. (2007). Teaching Tip: Recursive Joins to Query Data Hierarchies in Microsoft Access. *Journal of Information Systems Education 18*(1), 5-10. http://jise.org/Volume18/n1/JISEv18n1p5.pdf

Selected primary studies (cont.)

ID	Publication
PS29	Dadashzadeh, M. (2007). Teaching Tip: Specification and Enforcement of Semantic Integrity Constraints in Microsoft Access. *Journal of Information Systems Education 18*(4), 393-398. http://jise.org/Volume18/n4/JISEv18n4p393.pdf
PS30	Danaparamita, J., & Gatterbauer, W. (2011). QueryViz: Helping Users Understand SQL Queries and Their Patterns. In *Proceedings of the 14th ACM International Conference on Extending Database Technology (EDBT)* (pp. 558–561). ACM. doi.org/10.1145/1951365.1951440
PS31	Davis, J.S. (1990). Experimental investigation of the utility of data structure and E-R diagrams in database query. *International Journal of Man-Machine Studies 32*(4), 449-459. doi.org/10.1016/S0020-7373(05)80142-7
PS32	Dean, T.J., & Milani, W.G. (1995). Transforming a database systems and design course for non computer science majors. In *Proceedings of the 1995 25th Annual Frontiers in Education Conference (FIE)*. IEEE. doi.org/10.1109/fie.1995.483191
PS33	Dietrich, S.W., Goelman, D., Borror, C.M., & Crook, S.M. (2015). An Animated Introduction to Relational Databases for Many Majors. *IEEE Transactions on Education 58*(2), 81-89. doi.org/10.1109/TE.2014.2326834
PS34	Dietrich, S.W., & Urban, S.D. (1996). Database Theory in Practice: Learning from Cooperative Group Projects. In *Proceedings of the 27th ACM Technical Symposium on Computer Science Education (SIGCSE)* (pp. 112–116). ACM. doi.org/10.1145/236452.236520
PS35	Do, Q., Agrawal, R.K., Rao, D., & Gudivada, V.N. (2014). Automatic generation of SQL queries. In *Proceedings of the 121st ASEE Annual Conference and Exposition.*
PS36	Dollinger, R., & Melville, N.A. (2011). Semantic evaluation of SQL queries. In *Proceedings of the 2011 IEEE 7th International Conference on Intelligent Computer Communication and Processing (ICCP)* (pp. 57–64). IEEE. doi.org/10.1109/ICCP.2011.6047844
PS37	Fekete, A. (2005). Teaching Transaction Management with SQL Examples. In *Proceedings of the 2005 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)* (pp. 163–167). ACM. doi.org/10.1145/1067445.1067492
PS38	Fong, J., Lee, J., & Fong, A. (2005). Student Centered Knowledge Level Analysis for eLearning for SQL. In *Advances in Web-Based Learning (ICWL) 2005*, (pp. 174–185). Springer. doi.org/10.1007/11528043_17
PS39	Garner, P., & Mariani, J.A. (2015). Learning SQL in steps. *Journal of Systemics, Cybernetics and Informatics 13*(4), 19-24.
PS40	Green, G.G. (2005). Teaching Case: Greta's Gym: A Teaching Case for Term-Long Database Projects. *Journal of Information Systems Education 16*(4), 387-390. http://jise.org/Volume16/n4/JISEv16n4p387.pdf
PS41	Gudivada, V.N., Nandigam, J., & Tao, Y. (2007). Enhancing student learning in database courses with large data sets. In *Proceedings of the 2007 37th Annual Frontiers in Education Conference (FIE)*. IEEE. doi.org/10.1109/fie.2007.4418135
PS42	Hardt, R., & Gutzmer, E. (2017). Database Query Analyzer (DBQA) - A Data-Oriented SQL Clause Visualization Tool. In *Proceedings of the 18th ACM Conference on Information Technology Education (SIGITE)*. ACM. doi.org/10.1145/3125659.3125688
PS43	Hollingsworth, J.E. (2008). Teaching Query Writing: An Informed Instruction Approach. In *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)* (pp. 351–351). ACM. doi.org/0.1145/1384271.1384393
PS44	Huang, C., & Morreale, P.A. (2016). A web-based, self-controlled mechanism to support students learning SQL. In *Proceedings of the 2016 IEEE Integrated STEM Education Conference (ISEC)* (pp. 218–223). IEEE. doi.org/10.1109/ISECon.2016.7457536
PS45	Hvorecký, J., Drlík, M., & Munk, M. (2010). Enhancing database querying skills by choosing a more appropriate interface. In *Proceedings of the IEEE Global Engineering Education Conference (EDUCON)* (pp. 1897–1905). IEEE. doi.org/10.1109/EDUCON.2010.5492434
PS46	Irwin, G., Wessel, L., & Blackburn, H. (2012). Teaching Case: The Animal Genetic Resource Information Network (AnimalGRIN) Database: A Database Design & Implementation Case. *Journal of Information Systems Education 23*(1), 19-28. http://jise.org/Volume23/n1/JISEv23n1p19.pdf
PS47	Julavanich, T., Nalintippayawong, S., & Atchariyachanvanich, K. (2019). RSQLG: The Reverse SQL Question Generation Algorithm. In *Proceedings of the 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA)* (pp. 908–912). IEEE. doi.org/10.1109/IEA.2019.8715233
PS48	Kawash, J. (2014). Formulating Second-order Logic Conditions in SQL. In *Proceedings of the 15th ACM Conference on Information Technology Education (SIGITE)* (pp. 115-120). ACM. doi.org/10.1145/2656450.2656452
PS49	Ke, H., Zhang, G., & Yan, H. (2009). Automatic Grading System on SQL Programming. In *Proceedings of the 2009 IEEE International Conference on Scalable Computing and Communications (ScalCom)* (pp. 537–540). IEEE. doi.org/10.1109/EmbeddedCom-ScalCom.2009.105
PS50	Kenny, C., & Pahl, C. (2005). Automated Tutoring for a Database Skills Training Environment. In *Proceedings of the 36th ACM Technical Symposium on Computer Science Education (SIGCSE)* (pp. 58–62). ACM. doi.org/10.1145/1047344.1047377
PS51	Leitheiser, R.L., & March, S.T. (1996). The Influence of Database Structure Representation on Database System Learning and Use. *Journal of Management Information Systems 12*(4), 187-213. doi.org/10.1080/07421222.1996.11518106
PS52	Lertnattee, V., & Pamonsinlapatham, P. (2017). Blended learning for improving flexibility of learning structure query language (SQL). In *International Conference on Blended Learning* (pp. 343–353). Springer. doi.org/10.1007/978-3-319-59360-9_30
PS53	Martín, C., Urpí, T., Casany, M.J., Burgués, X., Quer, C., Rodríguez, M.E., & Abelló, A. (2013). Improving learning in a database course using collaborative learning techniques. *International Journal of Engineering Education 29*(4), 986-997.
PS54	Matos, V.M., & Grasser, R. (2002). Teaching tip: A Simpler (and Better) SQL Approach to Relational Division. *Journal of Information Systems Education 13*(2), 19-28. http://jise.org/Volume13/n2/JISEv13n2p85.pdf
PS55	Matos, V.M., Grasser, R., & Jalics, P. (2006). The Case of the Missing Tuple: Teaching the SQL Outer-join Operator to Undergraduate Information Systems Students. *Journal of Computing Sciences in Colleges 22*(1), 23-32. http://dl.acm.org/citation.cfm?id=1181811.1181814
PS56	McCann, L.I. (2003). On making relational division comprehensible. In *Proceedings of the 2003 33rd Annual Frontiers in Education Conference (FIE)*. IEEE. doi.org/10.1109/FIE.2003.1264699
PS57	Miao, Z., Roy, S., & Yang, J. (2019). Explaining Wrong Queries Using Small Examples. In *Proceedings of the 2019 ACM Conference on Management of Data (SIGMOD)* (pp. 503–520). ACM. doi.org/10.1145/3299869.3319866
PS58	Mills, R.J., Dupin-Bryant, P.A., Johnson, J.D., & Beaulieu, T.Y. (2015). Examining learning styles and perceived benefits of analogical problem construction on SQL knowledge acquisition. *Journal of Information Systems Education 26*(3), 203-217. http://jise.org/Volume26/n3/JISEv26n3p203.pdf
PS59	Morris, S.A. (2008). Teaching Case: Remote Services, Inc. *Journal of Information Systems Education 19*(2), 147-156. http://jise.org/Volume19/n2/JISEv19n2p147.pdf
PS60	Murray, M., & Guimaraes, M. (2008). Animated Database Courseware: Using Animations to Extend Conceptual Understanding of Database Concepts. *Journal of Computing Sciences in Colleges 24*(2), 144-150. http://dl.acm.org/citation.cfm?id=1409823.1409855

Selected primary studies (cont.)

ID	Publication
PS61	Myers, C., & Douglas, P. (2007). The Un-Structured Student. In *Proceedings of the 24th British National Conference on Databases (BNCOD)* (pp. 3–9). doi.org/10.1109/BNCOD.2007.22
PS62	Ortiz, J., Dietrich, S.W., & Chaudhari, M.B. (2012). Learning from Database Performance Benchmarks. *Journal of Computing Sciences in Colleges 27*(4), 151-158. http://dl.acm.org/citation.cfm?id=2167431.2167457
PS63	Oussena, S., & Dunckley, L. (2007). Adopting Student-Centred Approach to Advanced Database Teaching. In *Proceedings of the 24th British National Conference on Databases (BNCOD)* (pp. 10–14). doi.org/10.1109/BNCOD.2007.5
PS64	Pahl, C., Barrett, R., & Kenny, C. (2004). Supporting Active Database Learning and Training Through Interactive Multimedia. In *Proceedings of the 2004 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)* (pp. 27–31). ACM. doi.org/10.1145/1007996.1008007
PS65	Petrov, P., & Djolev, D. (2015). Combining the procedural and the set-based approaches in the teaching of SQL select statements in the introductory databases course. In *Proceedings of the International Scientific Conference Computer Science* (pp. 249–254).
PS66	Qian, G. (2018). Teaching SQL: A Divide-and-conquer Method for Writing Queries. *Journal of Computing Sciences in Colleges 33*(4), 37-44. http://dl.acm.org/citation.cfm?id=3199572.3199577
PS67	Randolph, G.B. (2003). The Forest and the Trees: Using Oracle and SQL Server Together to Teach ANSI-standard SQL. In *Proceedings of the 4th ACM Conference on Information Technology Curriculum (CITC)* (pp. 234–236). ACM. doi.org/10.1145/947121.947174
PS68	Reilly, C.F. (2018). Experience with Active Learning and Formative Feedback for a SQL Unit. In *Proceedings of the 2018 48th Annual Frontiers in Education Conference (FIE)*. IEEE. doi.org/10.1109/FIE.2018.8659173
PS69	Renaud, K., & Van Biljon, J. (2004). Teaching SQL - Which Pedagogical Horse for This Course? In *Proceedings of the 21st British National Conference on Databases (BNCOD)* (pp. 244-256). doi.org/10.1007/978-3-540-27811-5_22
PS70	Russell, G., & Cumming, A. (2004). Improving the Student Learning Experience for SQL Using Automated Marking. In *Proceedings of the International Conference on Cognition and Exploratory Learning in Digital Age (CELDA)* (pp. 281–288).
PS71	Sastry, M.K.S. (2015). An Effective Approach for Teaching Database Course. *International Journal of Learning, Teaching and Educational Research 12*(1). https://www.ijlter.org/index.php/ijlter/article/download/357/162
PS72	Seyed-Abassi, B. (1993). A SQL Project As a Learning Method in a Database Course. In *Proceedings of the 1993 ACM Conference on Computer Personnel Research (SIGCPR)* (pp. 291–297). ACM. doi.org/10.1145/158011.158238
PS73	Silva, Y.N., Almeida, I., & Queiroz, M. (2016). SQL: From Traditional Databases to Big Data. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE)* (pp. 413–418). ACM. doi.org/http://doi.acm.org/10.1145/2839509.2844560
PS74	Smelcer, J.B. (1995). User errors in database query composition. *International Journal of Human-Computer Studies 42*(4), 353-381. doi.org/10.1006/ijhc.1995.1017
PS75	Soflano, M., Connolly, T.M., & Hainey, T. (2015). An application of adaptive games-based learning based on learning style to teach SQL. *Computers & Education 86*, 192-211. doi.org/10.1016/j.compedu.2015.03.015
PS76	Stajduhar, I., & Mausa, G. (2015). Using string similarity metrics for automated grading of SQL statements. In *Proceedings of the 2015 38th International Convention on Information, Communication and Electronic Technology (MIPRO)* (pp. 1250–1255). doi.org/10.1109/MIPRO.2015.7160467
PS77	Sundin, L. & Cutts, Q. (2019). Is it feasible to teach query programming in three different languages in a single session?: A study on a pattern-oriented tutorial and cheat sheets. In *Proceedings of the 1st UK & Ireland Computing Education Research Conference (UKICER)*, p. 7. ACM. doi.org/10.1145/3351287.3351293
PS78	Taipalus, T. (2019). Teaching Tip: A Notation for Planning SQL Queries. *Journal of Information Systems Education 30*(3), 160-166. http://jise.org/Volume30/n3/JISEv30n3p160.pdf
PS79	Taipalus, T., & Perälä, P. (2019). What to Expect and What to Focus on in SQL Query Teaching. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE)* (pp. 198–203). ACM. doi.org/10.1145/3287324.3287359
PS80	Taipalus, T., Siponen, M., & Vartiainen, T. (2018). Errors and Complications in SQL Query Formulation. *ACM Transactions on Computing Education 18*(3), p. 15. doi.org/10.1145/3231712
PS81	Ullman, J.D. (2003). Improving the Efficiency of Database-system Teaching. In *Proceedings of the 2003 ACM International Conference on Management of Data (SIGMOD)* (pp. 1–3). ACM. doi.org/10.1145/872757.872759
PS82	Vijayasarathy, L., & Casterella, G.I. (2016). The Effects of Information Request Language and Template Usage on Query Formulation. *Journal of the Association for Information Systems 17*(10), 674-707. doi.org/10.17705/1jais.00440
PS83	Wagner, P.J., Shoop, E., & Carlis, J.V. (2003). Using scientific data to teach a database systems course. In *Proceedings of the 34th ACM Technical Symposium on Computer Science Education (SIGCSE)* (pp. 224–228). ACM. doi.org/10.1145/611892.611975
PS84	Watson, H.J., & Hoffer, J.A. (2003). Teradata university network: A new resource for teaching large data bases and their applications. *Communications of the Association for Information Systems 12*(1), 131-144. doi.org/10.17705/1cais.01209
PS85	Wu, P.Y., Baugh, J.M., & Harvey, V.J. (2005). Teaching SQL in database management for adult continuing education. In *Proceedings of the 2005 Information Systems Education Conference (ISECON)*. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.561.522&rep=rep1&type=pdf
PS86	Yen, M.Y. & Scamell, R.W. (1993). A human factors experimental comparison of SQL and QBE. *IEEE Transactions on Software Engineering 19*(4), 390-409. doi.org/10.1109/32.223806
PS87	Ying, M. & Hong, Y. (2011). The development of an online SQL learning system with automatic checking mechanism. In *Proceedings of the 7th IEEE International Conference on Networked Computing and Advanced Information Management (NCM)* (pp. 346–351). IEEE.
PS88	Yue, K.B. (2013). Using a Semi-Realistic Database to Support a Database Course. *Journal of Information Systems Education 24*(4), 327-336. http://jise.org/Volume24/n4/JISEv24n4p327.pdf
PS89	Zilligen, R., & Hidayat, A. (2008). A Misconception Module to a Database Courseware. In *Proceedings of the 46th ACM Annual Southeast Regional Conference (ACM-SE)* (pp. 529–530). ACM. doi.org/10.1145/1593105.1593250

B PRIMARY STUDY CLASSIFICATION

Primary study classification by topic and research type facets, *PS*s from primary study identifiers are omitted for brevity

	Evaluation research	Solution proposal	Replication study	Philosophical paper	Opinion paper	Experience report
Student errors	02, 03, 04, 24, 74, 79, 80	61, 73		18		67
Exercise database	05, 07, 11, 14, 15, 16, 17, 20, 21, 23, 31, 51, 82, 88	57, 83			62	41, 46, 68
Specific teaching approach	48	27, 54, 55, 56			28, 29, 37	86
Non-specific teaching approach	08, 13, 53	43, 58, 64			09, 38, 40, 60, 81, 89	19, 32, 33, 34, 52, 59, 63, 70, 71, 72
Patterns and visualization	10, 45	06, 22, 30, 39, 42, 66, 78		77		65, 69
Teacher workload	75	01, 12, 25, 26, 35, 36, 44, 47, 49, 76, 84, 87			85	50

C NUMBER OF PARTICIPANTS IN EACH PRIMARY STUDY

Number and type of participants in each primary study; primary studies that are not listed involved no participants, or did not report participant numbers

Study	Evidence
PS01	1,584 students
PS02	approximately 161,000 queries from approximately 2,300 undergraduate students (possibly same data as PS04)
PS03	986 students
PS04	approximately 161,000 queries from approximately 2,300 undergraduate students (possibly same data as PS02), out of which 551 queries from 321 students studied in more detail
PS05	60 undergraduate students
PS06	3 postgraduate students
PS07	342 subjects
PS08	157 students
PS10	88 undergraduate telecommunication students
PS11	95 advanced undergraduate and postgraduate students
PS13	103 students
PS14	23 graduate students
PS15	81 advanced undergraduate and graduate commerce students (possibly same data as PS16 and PS17)
PS16	81 advanced undergraduate and graduate commerce students (possibly same data as PS15 and PS17)
PS17	81 advanced undergraduate and graduate commerce students (possibly same data as PS15 and PS16)
PS19	48 students
PS20	33 undergraduate junior and senior students in computer information systems department
PS21	63 undergraduate students
PS23	112 subjects, but not everyone participated in all experiments
PS24	47 subjects, out of which 24 used SQL
PS31	116 subjects
PS33	75 students and 32 students
PS44	21 students
PS45	116 students
PS51	52 graduate business students
PS52	4 graduated [sic] students
PS53	928 grades from 6 semesters
PS55	22 undergraduate information systems students
PS57	approximately 170 undergraduate students
PS58	80 students
PS66	120 students
PS70	over 300 undergraduate students
PS74	17 undergraduate business administration students
PS75	120 higher education students
PS76	393 student answers
PS77	21 students
PS79	approximately 123,000 queries, out of which 8,773 queries from 744 undergraduate computer science and information systems students studied in detail
PS80	approximately 33,000 queries from 237 students
PS82	63 students
PS86	65 students
PS88	186 students