Learning Analytics Solution for Monitoring and Analyzing the Students' Behavior in SQL Lab Work

Abdelkader Ouared[a], Moussa Amrani[b] and Pierre-Yves Schobbens[c]
Faculty of Computer Science / NaDI, University of Namur, Rue Grandgagnage, Namur 5000, Belgium

Keywords: SQL Language Learning, Database, Learning Analytics, Learning Analytics Dashboard, Students' Behavior.

Abstract: Computer-assisted learning is widely discussed in the literature to aid the comprehension of SQL queries (Structured Query Language) in higher education. However, it is difficult for educators/instructors to track, monitor and analyze students' learning situation due to the higher education massification, and institutions with large classes. Consequently, we need to provide for educators a learning dashboard to monitor and analyze the digital traces issued from students during the practice learning in SQL course. We propose a system called LSQL (Learning Analytics for SQL) that is a solution based on the learning analytics' methodology. To this end, we propose (i) learning environment dedicated to help students understand the syntax and logic of SQL and getting data issued from these students during online SQL lab work, (ii) trace model which is designed to more effectively represent and capture the complex interactions/actions carried by a student during practice learning activities in virtual/remote laboratories, and (iii) learning analytics dashboard for educators to visualize the statistics and metrics that represent the students' behavior, and control the students progress in SQL skills to enhance the teaching activities. Tool support is fully available.

1 INTRODUCTION

The follow-up of the students during the SQL lab work in virtual and remote laboratories are an important element in evaluation techniques to judge their learning progress by educators/instructors (Venant et al., 2016; Pfeiffer et al., 2020). The learning analytics' methodology (LA) aims to understand the student behaviour by collecting the data from digital tools (e.g. the keyboard/mouse/screen actions or camera), processed and analyzed this data relating to students and their environment (Siemens and Gasevic, 2012). By exploring the literature, seminal work have proposed a computer-assisted learning to teaching SQL in higher education (e.g. (Mitrovic, 2003) (Ahadi et al., 2016) (Ahadi et al., 2015) (Brass and Goldberg, 2006) (Taipalus et al., 2018) (Karimzadeh and Jamil, 2022)). In our study, we will focus on the monitoring/analyzing students during the practice learning (or laboratory work) in SQL course that is a starting point for several purpose such as, the errors analysis, analysis of students' results, help to discover which parts of SQL material students are having trouble with, what should be improved or explained more clearly, also to optimize SQL education materials.

[a] https://orcid.org/0000-0003-1837-832X
[b] https://orcid.org/0000-0002-6987-1037
[c] https://orcid.org/0000-0001-8677-4485

1.1 Problem Statement

In SQL practice learning in virtual/remote laboratories, the student tries to formulate a query in SQL based on a given natural language (NL) queries with relational database schema. During the SQL lab, the student performs a set of solution attempts to write the correct query in SQL. However, monitoring and analyzing the student's activities is difficult and an increasingly harder challenge. *First*, this due to the large number of students in the lab room, especially in the context of Higher Education (HE), with following difficulty of each one individually, knowing who is doing what and when, and difficulty in assessing quantitatively and qualitatively the students' effort without exploring the solution attempts details. *Second*, analyzing students' writing SQL style and their behavior during SQL problem solving requires monitoring them in each solution attempts how they decompose the problem into a small and a sequencing problem. *Third*, the lack of a tailored trace model to handle the digital traces of training SQL activities for

Ouared, A., Amrani, M. and Schobbens, P.
Learning Analytics Solution for Monitoring and Analyzing the Students' Behavior in SQL Lab Work.
DOI: 10.5220/0011848200003470
In *Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023) - Volume 2*, pages 184-195
ISBN: 978-989-758-641-5; ISSN: 2184-5026

later analysis. Consequently, to analyze and monitor the students' behavior during SQL problem solving, we need to capture students' interactions with the measurements and metrics related to the learning process to understand the academic performance of students relative to SQL skills.

1.2 Our Contribution

As we said before, we need to monitor and analyze data logs, that is the starting point of any learning analytic solution. Despite event logs capturing behavioral information for large numbers of students, this information must be represented explicitly in a process model in order to propose an automatic approach for understanding the progress of students in SQL lab to enhance the teaching activities. To address this, in this paper, we propose a system called LSQL (Learning Analytics for SQL), a learning analytic solution allowing educators to monitor and analyze digital practices of the practice learning in SQL course. To support these needs, LSQL provides: **(i)** learning environment for helping students learn SQL and assisting them in the laboratory during query formulation in relational DBMSs (Database Management Systems), **(ii)** trace model to represent actions carried by a student on the SQL learning environment, comprises objects affected, and the time when the action was performed, and **(iii)** learning analytics dashboard (LAD) to support many visualization of the traces issued from these students which allows educators to track and understand student progress based on a set of pedagogical indicators (i.e. learning behavior indicators and content progress indicators) to enhance the SQL teaching activities.

1.3 Roadmap

We start in Section 2 by motivating example and related Work. In Section 2.1 we present a motivation scenario of a SQL learning to highlight the usefulness to bring educators and support staff back in control of the students during practical activities through a data-driven analytics solutions. We also in the Section 2.2 we present the related work. In Section 3, we introduce our system to monitor and analyze students behavior in SQL lab work. Then, in Section 4, we detail our tool support. Finally, we conclude in Section 5.

2 MOTIVATING EXAMPLE AND RELATED WORK

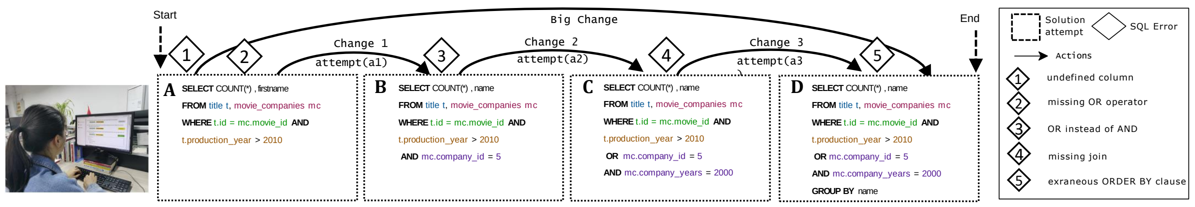
2.1 Motivating Example

Consider a scenario of a student cognitively engaging during a practical activity, she/he wants to write the correct query in SQL depending on a database schema hosted in a relational DBMSs (i.e. translate NL queries to SQL). Figure 1 shows the different attempts made by the student to come up with the correct query. In Figure 1, we can see an active window of history trace that is the complete record of all changes that have occurred during the query formulation. The trace is composed of four SQL query states (A,B,C, and D), in which the steps of the query execution generate one or more errors types. One query may contain a number of errors comes from various categories (Ahadi et al., 2016; Taipalus et al., 2018), it can be (i) *syntax errors* (i.e. occurs when the SQL grammar are violated) like specifying an invalid statement order, (ii) *semantic errors* (i.e. occurs when statements has no meaning) like table or view does not exist in the database, (iii) *logical errors* or *analyzing errors*, which causes a query to produce incorrect or undesired output, like the choice of table involved in the join operation and using the grouping functions, and (iv) *complications* like unnecessary join.

The steps from the first query Q_A to the last query Q_D represents the students' traceability. Q_A contains two syntax errors (*undefined column* and *missing* OR *operator*), Q_B contains a semantic error (OR instead of AND), Q_C, however, does not contain a semantic error, but it contains a logical error depends on the data fetched associated with the Q_C (*missing join*). Q_D contains also a logical error (i.e.*exraneous* ORDER BY *clause*) that produces incorrect result tables.

While, in practice the execution traces can take various forms (e.g. logs, list of events, method calls), we consider in this work a trace all students interactions that is a sequence of SQL subqueries and steps responsible for the query formulation changes. For each solution attempt, the student can made an error which may be syntactical, semantic or logical error. However, If the instructor only analyzed the last attempt (Q_D), in this situation it is difficult to perceive the history of attempts to check the types of mistakes made by the student and quantify the students' effort during the practical activity sessions. So, we need to identify trace related to SQL practice learning activities among exercises and analyze the data related to the different classes of errors.

Usually, the student solves SQL problems by breaking them down into smaller a small and a se-



A
```
SELECT COUNT(*) , firstname
FROM title t, movie_companies mc
WHERE t.id = mc.movie_id AND
t.production_year > 2010
```

B
```
SELECT COUNT(*) , name
FROM title t, movie_companies mc
WHERE t.id = mc.movie_id AND
t.production_year > 2010
 AND mc.company_id = 5
```

C
```
SELECT COUNT(*) , name
FROM title t, movie_companies mc
WHERE t.id = mc.movie_id AND
t.production_year > 2010
 OR mc.company_id = 5
AND mc.company_years = 2000
```

D
```
SELECT COUNT(*) , name
FROM title t, movie_companies mc
WHERE t.id = mc.movie_id AND
t.production_year > 2010
 OR mc.company_id = 5
AND mc.company_years = 2000
GROUP BY name
```

Figure 1: Scenario student during SQL query formulation, annotated with intermediate steps (student's attempts to write a SQL), and errors associated with each attempt to solve a single query.

quencing problem (Qian, 2018). To analyze and monitor the students' behavior during SQL problem solving, we need to capture students' actions with its corresponding sub-problems SQL formulation, to understand how before and after a choice of a set of steps, operation/traces affects the final result.

The change on the query state is identified when a user tests the input query based on the SQL query signature (e.g. tables involved in the query, relations used in the join operation, selection/join predicates used). For example, in the scenario presented in Figure 1, the educator could track errors related to query states (A,B,C, and D) to get a mental image about a particular student behavior and the overall students. We believe providing a history of SQL learning activities in more granularity as small changes instead of allowing instructors to more understand and analyze the students' behavior during SQL lab work. Furthermore, this analytic evaluation must be based on a set of indicators to understand the students' behaviors from their digital learning activities. Usually in LA, they considered measurements and metrics related to the learning process that are calculated from the data captured: learning behavior indicators (e.g. *timing of starting activities*, *timing of completing activities*) and content progress indicators (e.g. *completed course activities*, *current course grade*, *completed graded assignments*) (Jivet et al., 2020).

2.2 Related Work

As we said before, there has been a many efforts to assess student understanding of SQL in computer science education research. Previous SQL research studies mainly focuses on one of two content areas. First, on the development and analysis of a particular tool for facilitating SQL learning, and second, on the study of student errors in SQL.

In (Brusilovsky et al., 2010; Mitrovic, 2003; Karimzadeh and Jamil, 2022), the authors provide intelligent tutors to help teach SQL without the instructor's intervention. In the same direction, authors in (Xu et al., 2022) propose Intelligent Tutoring System (ITS) that automatically predicts difficulty of SQL programming problems (Xu et al., 2022). Many recent research papers start studying the common errors and pitfalls that people run into while trying to learn to write queries in SQL (Ahadi et al., 2016; Ahadi et al., 2015; Brass and Goldberg, 2006; Taipalus et al., 2018). In (Taipalus and Perälä, 2019), the authors explored which errors are persistent, the results show that syntax and semantic errors are less likely to persist than logical errors and complications. The study of (Brass and Goldberg, 2006) attempts to presented an extensive list of semantic errors to be used in Database Management System (DBMS) compilers, and a set of studies (Ahadi et al., 2016), which inspired us to this research, explored the frequencies of syntax and semantic errors students made when learning SQL. At the same time, several studies propose solution to enhance the SQL learning such as ScaffoldSQL for student learning reducing students' cognitive load towards solving SQL queries, providing immediate feedback for students, and help educators of flipped classrooms identify students who are struggling early (Borchert, 2021; Borchert and Walia, 2022) and gamification to improve student motivation and learning (Deterding et al., 2011; Thom et al., 2012; Fitz-Walter et al., 2011; Narasareddygari et al., 2019). Similarly, (Tuparov and Tuparova, 2021) a pilot study of the implementation of gamified self-training and self-assessment in an online SQL course as solution using Moodle[1]-based LMS practices. The authors exploit the generated digital traces expressed in xAPI[2] that would be interesting to collected and analyzed to motivate students to attend pure online activities in the educational process to improve learning and help educational stakeholders in their decision-making. In another line of research, the authors of (Godinez and Jamil, 2019) addressed the case of advancing NLP-based voice querying. By having an interactive spoken dialogue interface for querying relational databases, they were able to build a functional learning management system for SQL that has the potential to support the classes of queries novice SQL

[1]A learning management system (LMS) is a software application or web-based technology used to plan, implement and assess a specific learning process.

[2]http://experienceapi.com/overview

students encounter frequently (Godinez and Jamil, 2019). In a similar trend, the authors in (Turčínek and Farana, 2022) present a database and SQL microlearning course that offers students the opportunity to more easily absorb through small chunks that usually last no longer than a few minutes. In (Ouared and Chadli, 2021) the authors present the usefulness of domain-specific languages in education of SQL query optimizer.

By exploring the literature, to the best of our knowledge, there is no work propose a dedicated model trace in order to capture students' writing SQL style and their behavior, this model will be serve as a starting point for more comprehensive students' behavior using learning analytic methodology. Similar efforts have been conducted using learning analytics to analyze the students' behavior in programming language courses (e.g. (Fu et al., 2017; Chaparro et al., 2021)).

3 OUR SYSTEM

Figure 2 shows an overall view of our proposed solution. Our system LSQL (Learning Analytics for SQL) starts by storing data from the students' interactions (i.e. *event log*) during the SQL query formulation for later analysis. Thus, LSQL offers constructive feedback in incremental way to help students to spot the errors and solving SQL queries.

The trace represents the history of the necessary data for the calculation of the indicator such as, the actions related to the SQL learning activities, and user statistics (i.e. input users, click action sequences, options[choice], browse SQL education materials). On the other hand, educators use a learning analytics dashboard (LAD) to track students from the collected data to identify and monitor the important indicator related to SQL learning that reflects the student's behavior and understanding progress from the event log. Based on these pedagogical indicators, educators identify error types, and understand the progress of their students in the knowledge of SQL queries in order to enhance the teaching activities.

3.1 Trace of SQL Learning Activity

As we said previously, data logs can be extracted from different student's actions (i.e. input user, click action sequences, options[choice], browse SQL education materials). It consists of a set of the required digital practices of SQL practice learning in virtual/remote laboratories. A digital practice is defined as a set of events. Each event involves several attributes

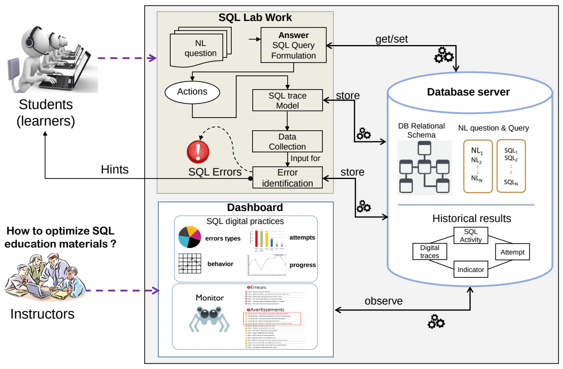


Figure 2: High-level view of the system architecture.

such as: the query input, query rewrite, query test, and report errors generated by DBMS when execute an SQL query.

SQL query problems solving are very popular in many application domains, and often tackled by decomposing them into a small problem and a sequencing subproblems. Student actions sequencing are not independent subproblems of SQL formulation, since finding the basic building blocks that are introduced and composed to realise complex queries. To perceive these digital traces related to the practice learning in SQL course, we must construct a SQL learning trace to handle the data log entries during the learning activities (Figure 3).

The change is identified when a user test the input query based on the SQL query signature that is expressed in the class `SQLQuery` (Figure 3). This class describe the number of tables involved in the query, multiple tables used in the join operation, join and selection predicates used in the query, number of equality selection predicates, number of non-equality selection predicates, number of equijoin predicates, number of non-equijoin predicates, number of sort columns (ORDER BY clause), aggregate functions (GROUP BY with HAVING clause) used , and number of nested subqueries.

To handle the digital traces of training SQL activities, we have propose a SQL trace constructor based on a set of operations. These latter provide the solution path that reflect the students' writing SQL style and their behavior aspects during SQL problem solving. Figure 4 visualizes the main operations required for students' trace construction.

- **initializeQuery:** initialize the first SQL query introduced by a student as first attempt in the activity log.
- **addStudentAction:** add a student action operation in the event trace if query state created/deleted or changed by a student.

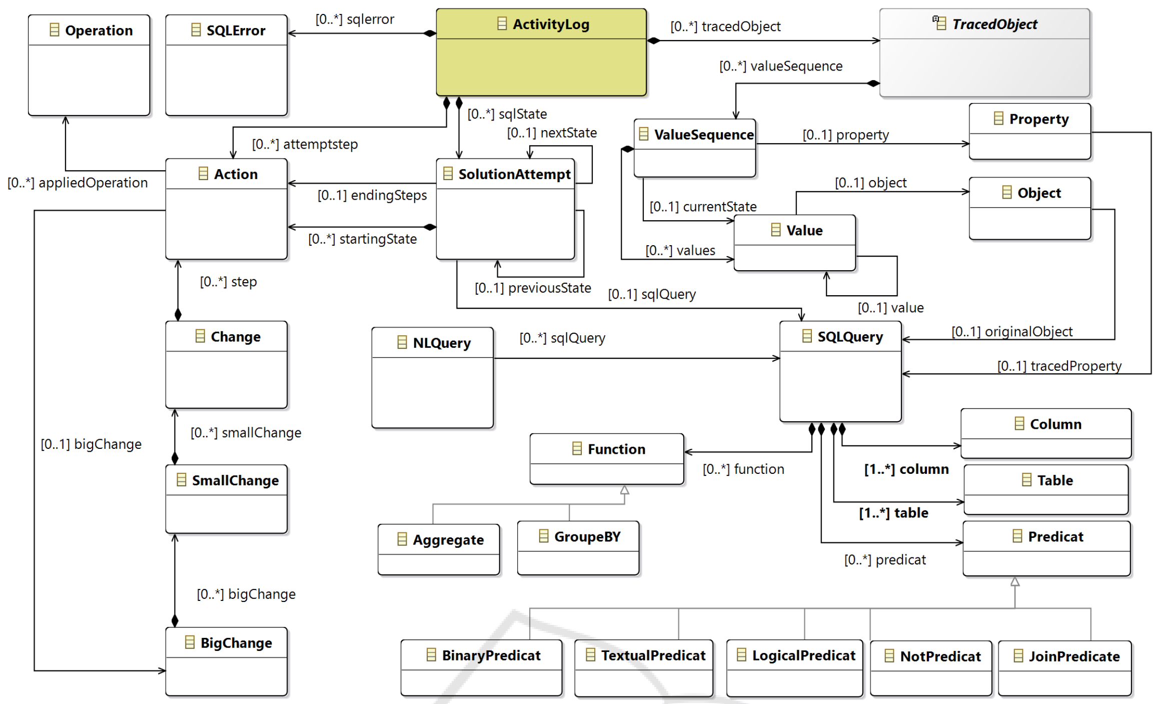


Figure 3: Excerpt of SQL trace model.

- **addSQLSmallChange:** add a small change in the SQL trace.
- **bigSQLChangeStarted:** means that a big change has started on the initial query signature.
- **bigSQLChangeEnded:** means that a big step has ended in the last attempt.

We remind you that we want to know the students behavior during SQL lab work. As illustrated in our motivating example (Figure 2 of Section 1), a big step during SQL formulation is simply a sequence of small steps, we only need to capture query states with its error type before and after small steps. However, we also need to capture when steps occur, hence addSQLSmallChange must be called at small steps, while bigSQLChangeStarted and bigSQLChangeEnded must be called before and after a big step, respectively.

In summary, all the solution attempts required to construct the trace to represent actions carried by a student during practice learning activities. Figure 5 shows the construction process as follows: stay tuned to check if a student do an action (cf. ①), if a student introduce an action (e.g. input query, test, choice), our system execute the operation *initializeQuery* to initialize the first attempt in the activity log and and annotate the *bigSQLChangeStarted* (cf. ②), then, its analyze change in solution attempt, for that, it start by identify the traced objects affected (i.e. tables, attributes, functions, predicates) by changing based on SQL query signature (cf. ③), based on the object affected by students' actions, our system identify the kind change in terms of *SmallChange* and *BigChange* related to solution attempts (cf. ④), after based on query testing, our system identify the SQL class error with its category corresponding to the solution attempts (cf. ⑤). At the last attempt, a big step has ended in the last attempt, i.e. *bigSQLChangeEnded* (cf. ⑥).

3.2 Traceability and Constructive Feedbacks

The collected data through our SQL trace constructor are persisted and exploited graphically to represent attempts, type of errors, students' behavior and progress. We based on the classification proposed by (Taipalus et al., 2018) for identifying various categories of SQL query such as syntax errors, logical errors, semantic errors and complications (see Figure 6). This identification is based on DBMS error codes and underlying reasons, for example in PostgresQL[3] DBMS, the error code 42601 means "wrong syntax".

The category of errors gives the possibility to classify each attempt of the student in a category. In order to help the student to identify the cause of any errors and increase the interactivity with our system, we

[3]https://www.postgresql.org/

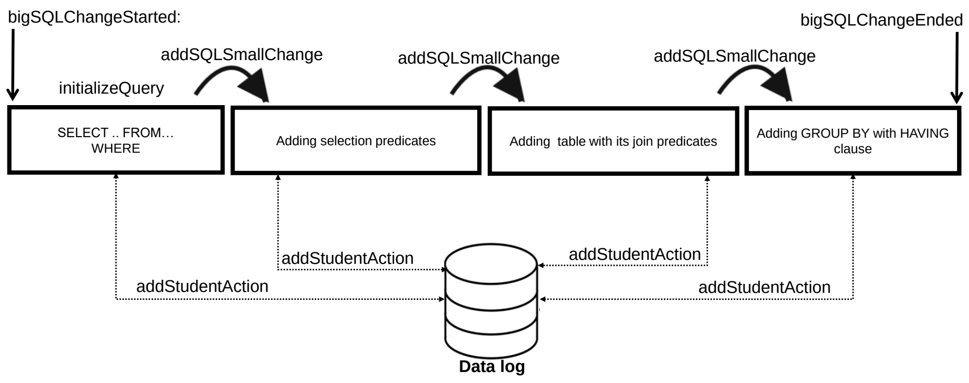


Figure 4: Illustration of operations of the SQL trace constructor.

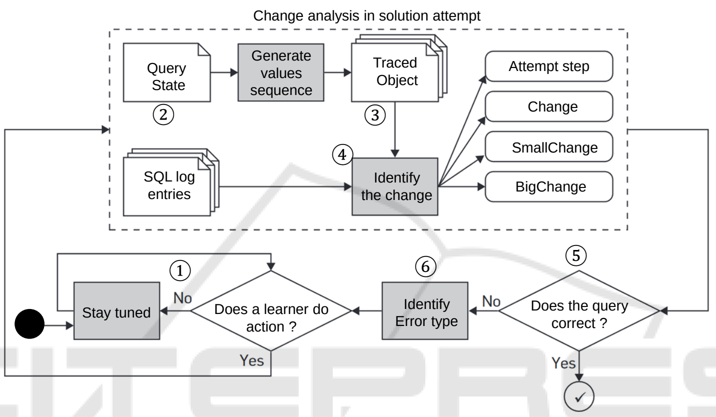


Figure 5: Process to construct the trace of actions carried by a student during training activities.

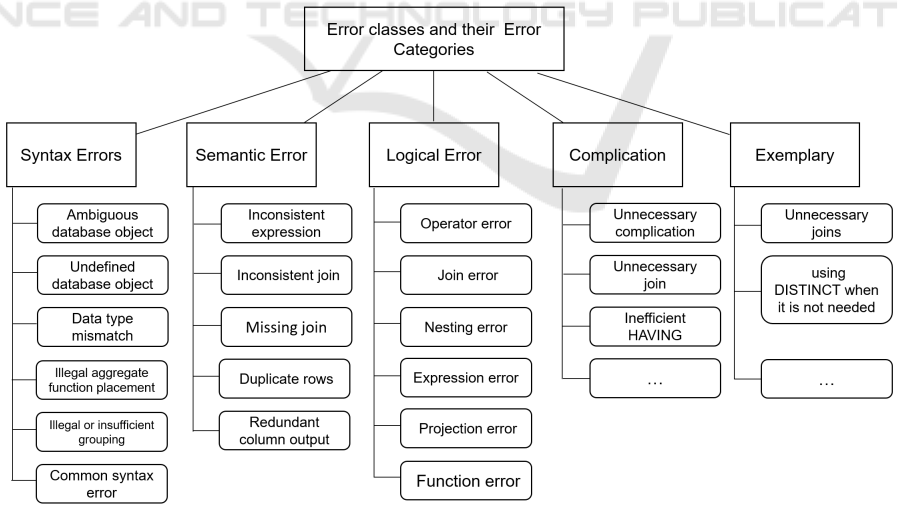


Figure 6: Error classes and their error categories, and complications with an exemplary query in SQL query formulation.

have proposed an algorithm to obtain an immediate answer as hints, or constructive feedbacks (see Figure 7). If a query contains an error, a guidance through incremental feedback (hint) is dedicated to guide the student to spot the SQL errors. The hints are applied based on the error type and the number of solution attempts. Table 1 shows a constructive feedbacks in incremental way with its corresponding level.

- **Level 1:** provides hints expressed in general statement at high level of abstraction, in which students must cognitively engaging with these explanations.
- **Level 2:** expresses in specific statement with vocabulary depending on SQL language query, with more explanation than the level 1, as guidance for students.
- **Level 3:** provides hint contains the error type by comprehensible explanations than the level 2 to spot the error type in the current query.

We formalized the application of the hints as ECA type (Event, Condition -Action) using the tables that are defined as follows:

- **ActivityLog** (*LogID*, *TracedObject*, *currentStep*, *windowSize*, *description*, *location*)
- **SolutionAttempt** (*attemptID*, *eventType*, *objectAffected*, *actionPerf*, *QueryEvaluation*, *SQLError*, *description*)
- **Action** (*actionID*, *actionType*, *time*, *event*, *prevAction*, *postAction*)
- **Change** (*changeID*, *object*, *SmallChange*, *BigChange*, *bigSQLChangeStarted*, *bigSQLChangeEnded*)

So, here's an example of hint application.

Example of hint application: Use hint contains the error type (level 3).
- When: Logical Errors (**Event**).
- When: Query SQL contain Join error, [not updating](**Event**).
- Where: [always] Join on multiple relations (**Condition**).
- What: join on incorrect table (**Action**).

3.2.1 Analyze Students' Behavior

Analyzing students' behavior that makes attempts to solve the given NL query during the learning activity can be traced as a path using the tree structure at a detailed level of granularity according to the educator's requirements. When the educator zooming on the behavior or content progress indicators of a student, he/she can see more information about the types of actions he/she makes and we can filter actions and activity timeline for a specific student. These indicates a status in different aspects of the student's behavior (e.g. Engagement in the SQL lab work, Timing of starting activities, Timing of completing activities) thereby providing a global overview of the student.

Analyzing students' behavior is a process composed by a set of steps that must be applied to the data sources in order to understand his/her behavior from the performance of learning activity. With the aim of guiding educators to analyze students' behavior, we propose a tree structure, the aspects to be analyzed are represented as nodes to drive user to the next analysis. The first step is selecting the aspect that will be analyzed, the leaf nodes represent the possible metrics and measurement that would be useful for the educator. Figure 8 shows an example of to analyze students' behavior in SQL lab work, in which the students' interactions compose the path leading to understanding the student behaviour. Thanks to SQL model traces of learning activities (i.e. operation, small, and big before and after change with the corresponding feedback), the educators can track and analyze students behavior step by step by checking the logic leading to the final query proposed by students. This structure indicates attempts made by the student with their results, and their digital learning activities using measurements and metrics of the learning process such as the learning behavior indicators (e.g. timing of starting activities, timing of completing activities, user statistics) and content progress indicators (e.g. completed learning activities, success rate, error rate of wrong answer, and the type of errors). On the other hand, this trace helps to see how the student behaves and identify the difficulty of SQL writing problems.

4 TOOL SUPPORT

Our system has been implemented with MVC (Model View Controller) framework, using Java FX with PostgreSQL[4], available on Github [5]. A prototype of our system was developed to be used by the second-year Bachelor's program at *the computer science department*. Our tool support contains two parts:

- *SQL learning environment Side:* the first part shows the learning environment dedicated for SQL self-training in undergraduate database courses.
- *Learning analytics dashboard Side:* the second part provides the learning analytics dashboard for educators and the overall tendencies of learning behaviors.

The usage scenario of LSQL is organized as it is shown in Figure 9. In the following we detail it step by step.

4.1 SQL Learning Environment Side

Our tool is deployed using three layered components, including *presentation and access*, *compuation* and *storage*. A student connects to a database server layer using a TCP/IP local-area network (LAN) with no or

[4]https://www.postgresql.org/
[5]https://github.com/OUARED-A/LSQL

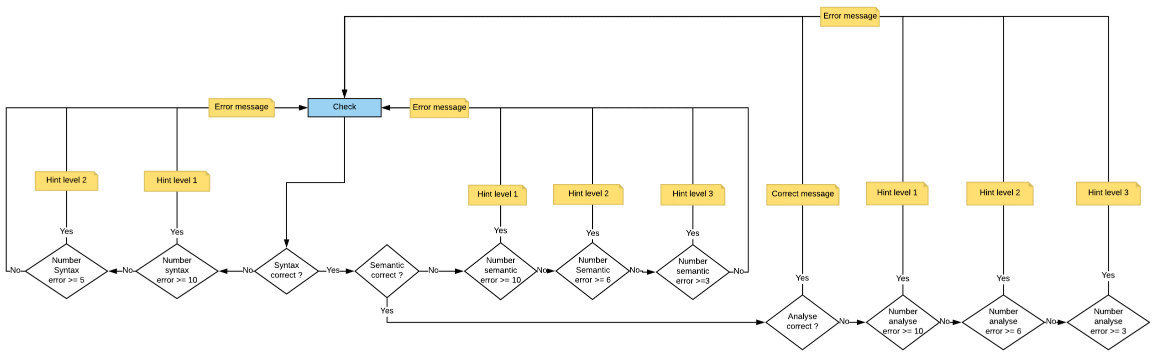


Figure 7: constructive feedbacks in incremental way with its corresponding levels.

Table 1: The three level of SQL hints.

Hint Level	Description	SQL Syntax Error Example
Level 1	Hint is expressed in general statement	Wrong syntax
Level 2	Hint is expressed in specific statement	Invalid pedicat syntax
Level 3	Hint contains the error type	Argument of AND must be type Boolean

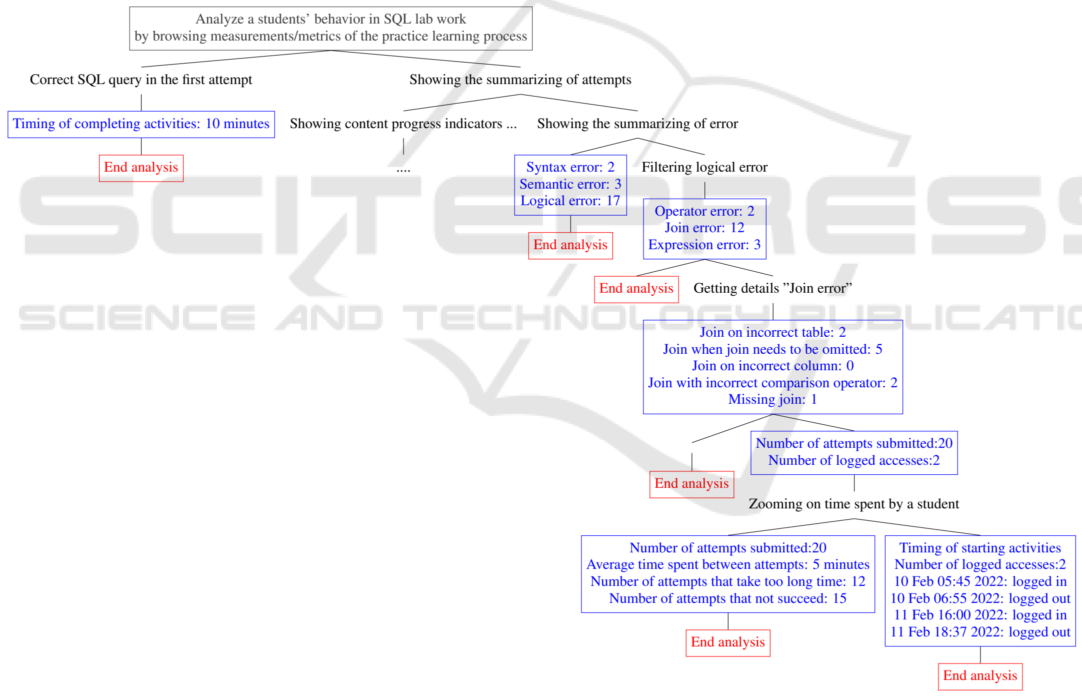

Analyze a students' behavior in SQL lab work
by browsing measurements/metrics of the practice learning process
Correct SQL query in the first attempt
Showing the summarizing of attempts
Timing of completing activities: 10 minutes
Showing content progress indicators ...
Showing the summarizing of error
End analysis
....
Syntax error: 2
Semantic error: 3
Logical error: 17
Filtering logical error
End analysis
Operator error: 2
Join error: 12
Expression error: 3
End analysis
Getting details "Join error"
Join on incorrect table: 2
Join when join needs to be omitted: 5
Join on incorrect column: 0
Join with incorrect comparison operator: 2
Missing join: 1
Number of attempts submitted:20
Number of logged accesses:2
End analysis
Zooming on time spent by a student
Number of attempts submitted:20
Average time spent between attempts: 5 minutes
Number of attempts that take too long time: 12
Number of attempts that not succeed: 15
Timing of starting activities
Number of logged accesses:2
10 Feb 05:45 2022: logged in
10 Feb 06:55 2022: logged out
11 Feb 16:00 2022: logged in
11 Feb 18:37 2022: logged out
End analysis
End analysis

Figure 8: Example of analyze students' behavior from to the learning activities history.

internet connections. The choice of LAN network is motivated by conducting a fairer evaluation of the groups during SQL lab work to identify students cognitively engaging during a practical activity, and avoid students that submit reports copied from others.

In Figure 10a, the student uses the module responsible for gives information about his/here identity, and the connection establishment with the DBMS server. Users use this interface to introduce connection parameters, i.e. server name, DBMS, database name, destination port, IP address, user, and password. When connected, our system provides a simple and and intuitive user interface to build a query and to aid SQL understanding as shown in the Figure 10b. A student starts by selecting an exercise from the repository of questions or exercises. Each exercise presented by a relational database with a set of queries expressed in natural language (NL). Through

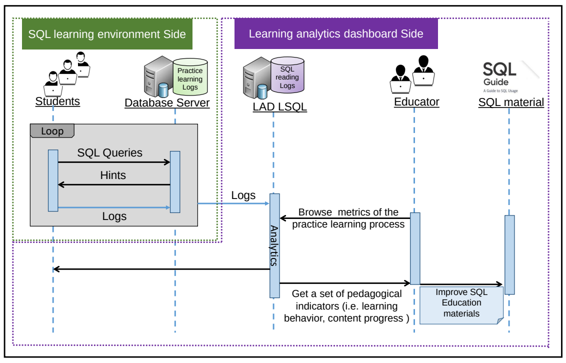

SQL learning environment Side
Learning analytics dashboard Side
Students
Practice learning Logs
Database Server
SQL reading Logs
LAD LSQL
Educator
SQL Guide
A Guide to SQL Usage
SQL material
Loop
SQL Queries
Hints
Logs
Logs
Analytics
Browse metrics of the practice learning process
Get a set of pedagogical indicators (i.e. learning behavior, content progress)
Improve SQL Education materials

Figure 9: Conceptual processes of LSQL components.

this user interface, the user has first to select the database schema, and NL query to be translated in SQL language (Figure 10b) and check the correctness of the solution. Note that every change made to the user interface is stored using our SQL trace model. The student introduces the SQL query and executes his/here query to receive the result table or error messages with some improvement hints. This interface illustrated in Figure 10b includes the constructive feedbacks process to help when a student makes different errors among incorrect final solutions after a number of attempts. This process sends hints as feedback without direct instructor intervention when an answer was considered incorrect.

4.2 Learning Analytics Dashboard Side

For educators, our tool supports provide a learning analytics dashboard to more understand visual perception and perceive visual changes states to each student. Educators use their account to browse the learning progress in SQL knowledge and behavioral tendencies were shown from charts and events logs. The detailed functions of the dashboard are described in the following subsections.

4.2.1 Interface for Reporting the Students' Activities

Our dashboard shows an overview of a given student activity on his/her profile (Figure 11). It gives viewers more detailed information about the digital practices of the lab work per student and actions types he/she makes. Figure 11a shows a graphical user dedicated for educators to track the learning behaviour of digital practices. This interface offers a wide possibility to monitor the performance of students relative to SQL skills, also allow educators to have reports of their activity in a timely manner and a human understandable way. Moreover, our LAD provides a students list providing correct responses with a good response time to spot these students since the number of students is very high.

The errors type of a student during the lab work indicate a status in different aspects of the students' behavior thereby providing a global overview of the student. When they select a given student, our tool gives an overview section to show all actions performed by this student with its evaluation: true, false (see Figure 11b).

4.2.2 SQL Error Analysis

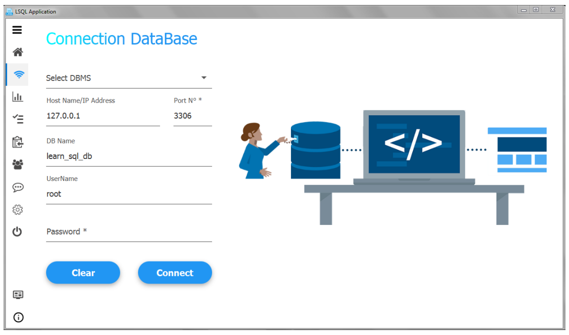
To monitor the students' learning activity during the sessions of practical work, our LAD reports contextual information and activities related to a given course to help educators identify and understand the common mistakes that students run into while trying to learn to write SQL queries. Identifying the most common SQL mistakes can also help educator discover which parts of material students are having trouble with, what should be improved or explained more clearly, how to optimize SQL education materials, etc.

The visualization offered by our LAD will facilitate instructors analyzing students' behavior, as it is shown in Figure 12, our dashboard shows the actual state of difficulties in SQL skills. The dashboard contains two four parts: the top-left part indicates the various SQL errors for a selected student to better understand each student's writing SQL style and behavior (Fig. 12a), this chart allows us to characterize students' weaknesses in their understanding of SQL material and reflects students' progression, the top-right part indicates total number of solution attempts (Fig. 12b), the down-left part indicates the traceability of SQL questions (Fig. 12c), and the down-right part indicates total number of errors for each question using a bar chart to show the error distribution by class (i.e. syntax, semantic and logic) based on the log data produced by students' interactions (Fig. 12d).

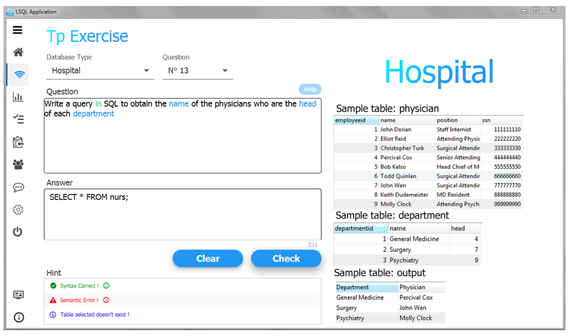
Using these visualizations, any change in a student's learning behavior reflects a change in his/her reporting state. This change in behavior is caused by providing constructive feedback for students to reduce students' cognitive load towards solving SQL queries and reducing instructor intervention.

4.2.3 Timing of Starting Activities

Another factor that must be considered to analyze the students behavior aspects, is to examine the timing of starting activities. In the context of SQL lab work, the repetition and the longest in activity are indicators to perceive the students' efforts. A repetition usually leads to an improvement of exercise results and in-



(a) System connection



Hospital

Question

Write a query in SQL to obtain the name of the physicians who are the head of each department

Answer

```
SELECT * FROM nurs;
```

Hint

✓ Syntax Correct !

⚠ Semantic Error !

ⓘ Table selected doesn't exist !

Sample table: physician

employeeid	name	position	ssn
1	John Dorian	Staff Internist	111111110
2	Elliot Reid	Attending Physic	222222220
3	Christopher Turk	Surgical Attendir	333333330
4	Percival Cox	Senior Attending	444444440
5	Bob Kelso	Head Chief of M	555555550
6	Todd Quinlan	Surgical Attendir	666666660
7	John Wen	Surgical Attendir	777777770
8	Keith Dudemeister	MD Resident	888888880
9	Molly Clock	Attending Psych	999999990

Sample table: department

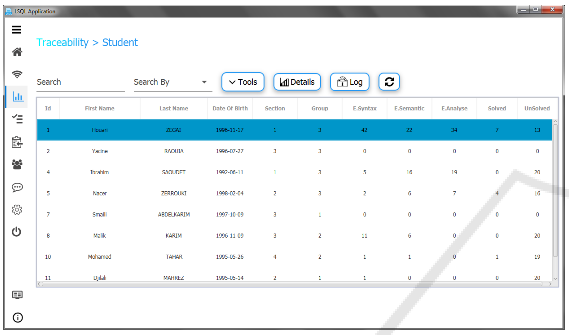
departmentid	name	head
1	General Medicine	4
2	Surgery	7
3	Psychiatry	9

Sample table: output

Department	Physician
General Medicine	Percival Cox
Surgery	John Wen
Psychiatry	Molly Clock

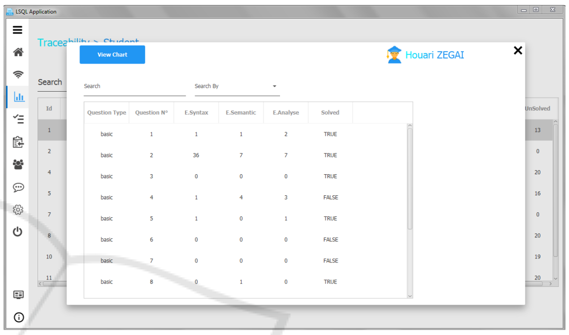
(b) Interactive interface of SQL practice learning

Figure 10: The LSQL learning environment.



Id	First Name	Last Name	Date Of Birth	Section	Group	E.Syntax	E.Semantic	E.Analyse	Solved	UnSolved
1	Houari	ZEGAI	1996-11-17	1	3	42	22	34	7	13
2	Yacine	RAOUIA	1996-07-27	3	3	0	0	0	0	0
4	Ibrahim	SAOUDET	1992-06-11	1	3	5	16	19	0	20
5	Nacer	ZERROUKI	1998-02-04	2	3	2	6	7	4	16
7	Smaili	ABDELKARIM	1997-10-09	3	1	0	0	0	0	0
8	Malik	KARIM	1996-11-09	3	2	11	6	0	0	20
10	Mohamed	TAHAR	1995-05-26	4	2	1	1	0	1	19
11	Djilali	MAHREZ	1995-05-14	2	1	1	0	0	0	20

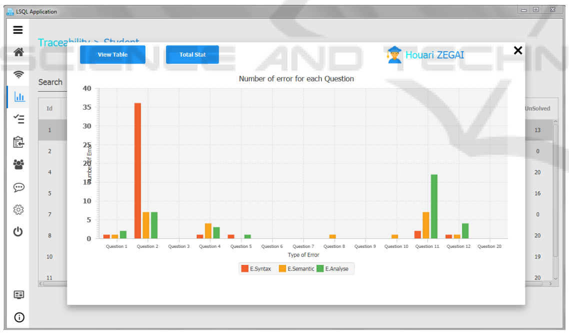
(a) Digital practices of the Lab Work per student



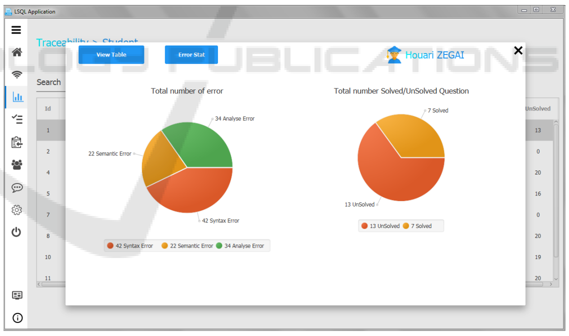
Question Type	Question N°	E.Syntax	E.Semantic	E.Analyse	Solved
basic	1	1	1	2	TRUE
basic	2	36	7	7	TRUE
basic	3	0	0	0	TRUE
basic	4	1	4	3	FALSE
basic	5	1	0	1	TRUE
basic	6	0	0	0	FALSE
basic	7	0	0	0	FALSE
basic	8	0	1	0	TRUE

(b) Detail trace of digital practices of a selected student

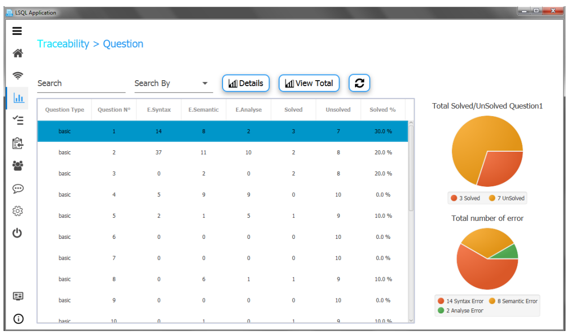
Figure 11: Analyzing the digital practices of students in the lab work.



(a) Number of error for each question for a selected student

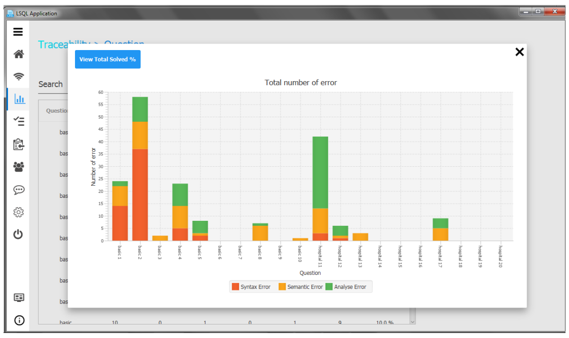


(b) SQL error distribution.



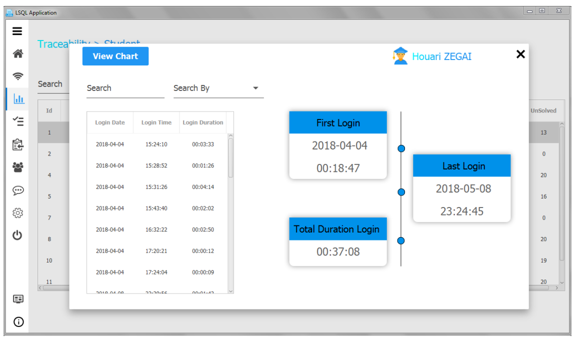
Question Type	Question N°	E.Syntax	E.Semantic	E.Analyse	Solved	Unsolved	Solved %
basic	1	14	8	2	3	7	30.0 %
basic	2	37	11	10	2	8	20.0 %
basic	3	0	2	0	2	8	20.0 %
basic	4	5	9	9	0	10	0.0 %
basic	5	2	1	5	1	9	10.0 %
basic	6	0	0	0	0	10	0.0 %
basic	7	0	0	0	0	10	0.0 %
basic	8	0	6	1	1	9	10.0 %
basic	9	0	0	0	0	10	0.0 %
basic	10	0	1	0	1	9	10.0 %

(c) Traceability on questions with SQL error distribution



(d) Total number of errors among questions

Figure 12: Situation of different students upon trying to run SQL query.



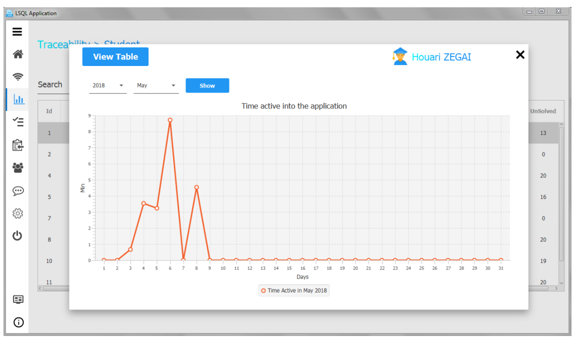
Login Date	Login Time	Login Duration
2018-04-04	15:24:10	00:03:33
2018-04-04	15:28:52	00:01:26
2018-04-04	15:31:26	00:04:14
2018-04-04	15:43:40	00:02:02
2018-04-04	16:32:22	00:02:50
2018-04-04	17:20:21	00:00:12
2018-04-04	17:24:04	00:00:09

First Login: 2018-04-04 00:18:47

Last Login: 2018-05-08 23:24:45

Total Duration Login: 00:37:08

(a) Logged in/logged out statistics



(b) Student's active time

Figure 13: A student's detailed timing of starting activities.

creased success rates at the last attempt. Our dashboard provides information about the timing of starting activities to qualitatively appreciate the students' activities during a learning session (Figure 13).

This aspect appears in logged in/logged out statistics by student as shown in Figure 13a, and the student's active time as shown in figure 13b (active, not active). When educators look at a student active with a set of actions performed during SQL practicing, they can appreciate the effort of the students during the learning activity session. With this chart, we can easily detect the active and inactive students, and can help to conduct a fairer evaluation of the groups during SQL lab work. For example, the student generates a lot of log data by keeping trying to test a query with many solution attempts that can be interpreted as really engaged during lab work. In contrast, the student not active can be interpreted as not engaged with SQL, also, a pause period may be interpreted as a student taking a large thinking period without interactions. Furthermore, tutors can check if the timing of starting activities has negative or positive correlation with the performance of students' progress and understanding (e.g. completing activities, errors number). However, the educators can switch between the visualization, and cross all this visualization to link students' behavior aspects aspects with their digital learning activities to get a state about each student and a big picture about the overall students.

5 CONCLUSION

In this paper, we have tackled the problem of eliciting and analyzing students' behavior during the practice learning in SQL course in virtual/remote laboratories. This work is motivated by the difficulty of the evaluation due the higher education massification, and institutions with large classes. We have presented a learning analytics solution for monitoring and analyzing students behavior in SQL lab work. Behaviour analytics is driven by an automatic process to capture a student's activities from an event log that is purposely designed for any relational DBMS. The proposed approach is useful for helping students learn SQL and assisting them when formulating SQL queries, (ii) collecting digital traces before and after each small and big step performed by students, and (iii) learning analytics dashboard to support reporting the traces issued from these students which allows educators to track and understand student progress in the SQL skills to enhance their teaching activities. The trace model contains objects with their attributes, which represent the main entities involved in the process of solving query SQL, and the relationships between the small steps performed by the student to achieve the global query. We have considered the state before and after each execution step in the event log and represent them via the learning analytics dashboard. Two important directions for future works can be considered. First, we will conducted some form of exploratory statistical analysis, such as correlational analysis, regression, or factor analysis. Second, applied machine learning techniques to obtain predictive model dedicated to to identify students' behaviors and trends, also to build adaptive learning in which the problem difficulty or challenge is recommended according to students' SQL skills in our system.

ACKNOWLEDGEMENTS

We sincerely thank Mr. Houari ZEGAI (Software Engineer at DKV Belgium) for his contributions to the development of the support tool ”LSQL”.

REFERENCES

Ahadi, A., Prior, J., Behbood, V., and 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*, pages 201–206.

Ahadi, A., Prior, J., Behbood, V., and 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*, pages 272–277.

Borchert, O. (2021). Scaffoldsql: Sql test cases+ parson's problems. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, pages 1281–1281.

Borchert, O. and Walia, G. (2022). Scaffoldsql: Using parson's problems to support database pedagogy. In *Proceedings of the 55th Hawaii International Conference on System Sciences*.

Brass, S. and Goldberg, C. (2006). Semantic errors in sql queries: A quite complete list. *Journal of Systems and Software*, 79(5):630–644.

Brusilovsky, P., Sosnovsky, S., Yudelson, M. V., Lee, D. H., Zadorozhny, V., and Zhou, X. (2010). Learning sql programming with interactive tools: From integration to personalization. *ACM Transactions on Computing Education (TOCE)*, 9(4):1–15.

Chaparro, E., Restrepo-Calle, F., and Ramirez-Echeverry, J. J. (2021). Learning analytics in computer programming courses. In *LALA*, pages 78–87.

Deterding, S., Dixon, D., Khaled, R., and Nacke, L. (2011). From game design elements to gamefulness: defining" gamification". In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*, pages 9–15.

Fitz-Walter, Z., Tjondronegoro, D., and Wyeth, P. (2011). Orientation passport: using gamification to engage university students. In *Proceedings of the 23rd Australian computer-human interaction conference*, pages 122–125.

Fu, X., Shimada, A., Ogata, H., Taniguchi, Y., and Suehiro, D. (2017). Real-time learning analytics for c programming language courses. In *Proceedings of the seventh international learning analytics & knowledge conference*, pages 280–288.

Godinez, J. E. and Jamil, H. M. (2019). Meet cyrus: the query by voice mobile assistant for the tutoring and formative assessment of sql learners. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 2461–2468.

Jivet, I., Scheffel, M., Schmitz, M., Robbers, S., Specht, M., and Drachsler, H. (2020). From students with love: An empirical study on learner goals, self-regulated learning and sense-making of learning analytics in higher education. *The Internet and Higher Education*, 47:100758.

Karimzadeh, M. and Jamil, H. M. (2022). Visql: An intelligent online sql tutoring system. In *2022 International Conference on Advanced Learning Technologies (ICALT)*, pages 212–213. IEEE.

Mitrovic, A. (2003). An intelligent sql tutor on the web. *International Journal of Artificial Intelligence in Education*, 13(2-4):173–197.

Narasareddygari, M. R., Walia, G. S., Duke, D. M., Ramasamy, V., Kiper, J., Davis, D. L., Allen, A. A., and Alomari, H. W. (2019). Evaluating the impact of combination of engagement strategies in sep-cyle on improve student learning of programming concepts. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, pages 1130–1135.

Ouared, A. and Chadli, A. (2021). Using mde for teaching database query optimizer. In *ENASE*, pages 529–536.

Pfeiffer, A., Lukarov, V., Romagnoli, G., Uckelmann, D., and Schroeder, U. (2020). Experiential learning in labs and multimodal learning analytics. In *Adoption of Data Analytics in Higher Education Learning and Teaching*, pages 349–373. Springer.

Qian, G. (2018). Teaching sql: A divide-and-conquer method for writing queries. *Journal of Computing Sciences in Colleges*, 33(4):37–44.

Siemens, G. and Gasevic, D. (2012). Guest editorial-learning and knowledge analytics. *Journal of Educational Technology & Society*, 15(3):1–2.

Taipalus, T. and 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*, pages 198–203.

Taipalus, T., Siponen, M., and Vartiainen, T. (2018). Errors and complications in sql query formulation. *ACM Transactions on Computing Education (TOCE)*, 18(3):1–29.

Thom, J., Millen, D., and DiMicco, J. (2012). Removing gamification from an enterprise sns. In *Proceedings of the acm 2012 conference on computer supported cooperative work*, pages 1067–1070.

Tuparov, G. and Tuparova, D. (2021). Gamification in higher education-a pilot study with sql course. In *ERIS*, pages 81–90.

Turčínek, P. and Farana, R. (2022). Database and sql microlearning course. In *Microlearning*, pages 157–169. Springer.

Venant, R., Vidal, P., and Broisin, J. (2016). Evaluation of learner performance during practical activities: An experimentation in computer education. In *2016 IEEE 16th international conference on advanced learning technologies (ICALT)*, pages 237–241. IEEE.

Xu, J., Wei, T., and Lv, P. (2022). Sql-dp: A novel difficulty prediction framework for sql programming problems. In *Proceedings of the 15th International Conference on Educational Data Mining*, page 86.