Sandbox Databases in SQL University Courses – State of the Art

Georgi Tuparov

New Bulgarian University, Department of Informatics, 21 Montevideo str. Sofia, 1618, Bulgaria

Abstract

Nowadays the body of knowledge in IT courses includes topics about relational databases in which students have to obtain at least basic knowledge and skills to perform CRUD (Create, Read, Update, and Delete) operations using SQL. For the students in the area of computer science, the advanced SQL topics like data definition, transaction processing, database administration, and some SQL procedural extensions are included, in addition to the basic SQL knowledge and skills mentioned above. By default, the theoretical lectures in SQL topics are supplemented by practical training using a real DBMS environment, which is appropriate for self-training too. A basic requirement for the used environments is to provide the SQL functionalities needed. Another basic requirement is to have easy procedures for recovering the database schema and state, or the entire DBMS environment with database schema(s) and state(s) inside. In the real database design and development process, this kind of DBMS environment is called a "sandbox database". Additionally, sandbox databases used in university courses have to be free of charge for the students (in general or covered by the university license), and to not have special hardware and/or software requirements. In this way, all students will be in an equal state during class or individual training i.e. the "sandbox" each time will be the same for all of them.

In this paper, a simple framework is developed for comparing sandbox databases for SQL training at university level and some popular database sandboxes available free. Variants for individual sandbox creation are analyzed using this framework.

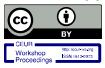
Keywords

SQL, training environments, sandbox database

1 Introduction

Nowadays the body of knowledge in IT courses includes topics about databases and data management. Although NoSQL databases are becoming more and more common, relational databases are still the main technology for storing and managing data, and SQL is an integral part of this. Since SQL is the major language in database systems, SQL knowledge and skills are also an important part of students' preparation for their professional career in the software industry [1]. In general IT courses, the students have to obtain at least basic knowledge and skills to perform CRUD (Create, Read, Update, and Delete) operations using SQL. For the students in the Computer science, Information Systems, Information Technology and Software engineering majors, the comprehensive database course is an obligatory part of the curriculums [2, 3, 4, 5]. In such database courses the advanced SQL topics like data definition, transaction processing, database administration, and some SQL procedural extensions like triggers and stored procedures are included, in addition to the basic SQL knowledge and skills mentioned above.

By default, the theoretical lectures in SQL topics are supplemented by practical sessions in computer labs where students have to solve tasks using SQL in a real DBMS environment with the example

Education and Research in the Information Society, October 13–14, 2022, Plovdiv, Bulgaria
EMAIL: gtuparov@nbu.bg (A. 1)
ORCID: 0000-0003-4162-5106 (A. 1)


database schema(s) in particular state(s) installed. Self-training using a real DBMS environment is also an important part of the learning process too.

A basic requirement for the environments used for practical training is to provide the SQL functionalities needed to cover the SQL topics learned. During the training process, it is possible that the database schema, state or DBMS environment settings are changed as a result of solving tasks, making experiments or by mistake, so another basic requirement is to have easy procedures for recovering the database schema and state, or the entire DBMS environment with database schema(s) and state(s) inside.

In the real database design and development process, this kind of DBMS environment is called a "sandbox database". Additionally, to support self-training, sandbox databases used in university courses have to be free of charge for the students (in general or covered by the university license), and not to have special hardware and/or software requirements. Another option is to provide sandbox databases as a service which can be accessed with minimal local software installed. In this way, all students will be in an equal state during class or individual training i.e. the "sandbox" will be the same for all of them at all times. Moreover, using sandbox database(s) during classes prepares the students for the real working environment in the software companies.

2 Related works and technological solutions

2.1 Example databases in the university textbooks

The example database schemas and SQL topics in most of the popular database books used in the university courses [6, 7, 8, 9] are mainly focused on online transaction processing (OLTP) features of relational databases, so SQL class examples, projects and exam tasks are traditionally based on application of Data Manipulation Language (DML) statements (INSERT, SELECT, UPDATE and DELETE) and partially on Transaction Control Language (TCL) statements (BEGIN, COMMIT, and ROLLBACK). Application of Data Definition Language (DDL) statements (CREATE, ALTER, and DROP) and Data Control Language (DCL) statements (GRANT, REVOKE) are partially included too. The same topics are covered in the books dedicated to SQL [10, 11]. With respect to the rising Big Data phenomena, some authors argued that this body of knowledge is not enough for data scientists and also SQL topics about data cleaning, data transformation and statistical querying are needed. [12, 13]

In some books [6, 7, 10, 11] there are no preferences about what DBMS environment to be used in the practical sessions, so examples are based on DML and DDL SQL statements which are supported in most of the popular DBMS. In others, like [8, 9], DDL scripts for creation and population of the example database, compatible with one or more DBMS, are provided and the examples are suitable for the particular DBMS(s). Generally, no special sandbox database is proposed or provided.

2.2 Sandbox databases for university courses in research papers

There exists a set of standalone SQL training environments which provide automatic evaluation and assessment of the students' SQL solutions. Such environments are directed mainly to students' self-training and/or formal grading. In self-training mode, these environments return some feedback about correctness of the solution to the students. By default, these environments implement only a fixed set of the database schemas, states and set of the queries included. In addition, a lot of these environments – SQLator [14], SQLify [15], SQL Exploratorium [16], testSQL [17], SQL-Tutor [18], SQLT-Web [19], AcceSQL [20], SQL Tester [21] and aSQLg [22] support only SELECT statement and don't allow students to execute "dangerous" DML (i.e. INSERT, UPDATE, DELETE) or DDL (i.e. CREATE, ALTER, and DROP) statements. SQL-Tutor and its web-based extension SQLT-Web use Constraint-Based Modeling [20, 21] instead of a real database engine for query evaluation. XDa-TA [23] environment allows usage only of DML "dangerous" statements, but internally they are converted for execution to equivalent for evaluation SELECT statements. In general, these environments are not fully-functional sandbox databases, because they are more focused on guided self-training or formal grading, than to the flexible database schema and state, and rich SQL functionalities.

One interesting and more powerful solution is MySQL Sandbox [24] which implements natively all DML and DDL statements and has no database schema fixed. Students are allowed to work simultaneously with the same database schema or with different ones, according to their needs. When a student starts the session, his/her own copy of the database is created and he/she is able to execute the full set of DML and DDL statements. MySQL Sandbox is integrated into DBLearn tool [25] and provides grading capabilities. Authors claim about 100% of effectiveness of this tool in automatic grading during the experimental phase.

An interactive SQL tutorial, integrated in learning management system (LMS) is presented in [12] where the sandbox database is part of the used LMS. Because each student's query is executed in a separate sandbox on a freshly initialized database instance, a rich set of SQL functionalities can be applied in the tutoring examples, including DML and DDL statements, stored procedures, triggers and user-defined functions. The database schemas, states and queries used are limited to the tutorial examples and test cases for evaluation of the students' solutions.

Another environment that implements more successfully the idea of a sandbox database for university courses is ClassDB [26], which supports SQL topics from basic to advanced ones with possibilities to run a mix of SQL and Python/R code inside a DBMS. Each student or team works in their own sandbox database to which instructors have access to provide feedback if it is needed. Both students and instructors can review activity logs to analyze activities in the sandbox database.

2.3 Sandbox databases suitable for university courses

Sandbox databases suitable for university database courses can be created in many ways using a rich set of technological solutions available. The classical approach is to provide the university level service as a central DBMS server with remote access. Another possible solution is to have local DBMS on each computer as an installed application, in Virtual Machines (VM) or in containers (i.e. Docker). In this case it is preferable the DBMS used to be free/community/express versions in order to avoid licensing problems for students' computers. In addition, sandbox databases provided as a free service are very popular in informal and self-training. Some of them can be successfully used in the university database courses as a primary sandbox database or as a complementary one, suitable for out of class training or for project development. These services are web-based or cloud-based and depending on the particular service could provide different SQL functionalities or database size. Free sandbox databases with fixed database schema and state are not flexible enough to fit in the university courses, thus are excluded from this review.

2.3.1 Sandbox databases provided at university level

Central DBMS server provides good functionality including DDL, DML, TCL and other specific to the DBMS used extensions. Due to security reasons, DCL commands are forbidden in the students' accounts. To access the DBMS server, "fat" (special client software, dependent on the DBMS used) or "thin" (SSH tool, web browser) clients have to be used in computer labs and on students' personal computers. DBMS administration is centralized and students don't have to care about it. Each student is provided with a personal database workspace to train SQL. Creating or restoring the database schema and state can be done by DB administrators or by the students, using scripts provided by the instructor.

2.3.2 Sandbox databases set locally

- **Free/community/express version of the DBMS** like MySQL [27], PostgreSQL [28], MS SQL Server Express [29], Oracle DB Express [30] etc., installed locally. This allows full SQL functionality without any limitation. Creating or restoring of the database schema and state can be done using scripts provided by the instructor, but restoring DBMS settings after wrong command(s) could be a very problematic task, and sometimes a reinstall of the DBMS is needed. Although setting or managing this kind of sandbox databases should not be a problem for system administrators in computer labs, it could be a problem for some of the students without some operating system knowledge and skills.

- **Local virtual machine** (VM) with VM images with free/community/express version of the DBMS provided by the instructor. Like in local DBMS installation, a VM allows full SQL functionality. Creating and restoring the database schema and state can be done using scripts provided by the instructor and DBMS settings can be restored by reloading the image in the VM. This kind of the sandbox database is easy to set up and manage, but uses more hardware resources than local DBMS installation. Free VM images of some DBMS listed above can be found in example at Bitnami site [31].
- **Local Docker container** with Docker images with free/community/express version of the DBMS provided by the instructor. This kind of sandbox database allows full SQL functionality. Creating and restoring the database schema and state can be done using scripts provided by the instructor and DBMS settings can be restored by reloading the Docker image. Initial setting of the Docker Desktop could make setting of this kind of sandbox databases more complicated for the students, but uses less hardware resources than DBMS installed in VM. Some free DBMS Docker images can be found in example at Bitnami site [31].

2.3.3 Web-based sandboxes available for free

- **Oracle Live SQL** [32] (Fig. 1) works with Oracle DB and allows fully functional DDL, DML and TCL queries to be executed. Also stored procedures and UDFs can be defined and used. After each login the user is provided with an empty workspace in which a new database schema can be created and populated. After logoff the workspace is cleared, but the created scripts are saved and can be used in the next session. This sandbox cannot execute DCL queries.

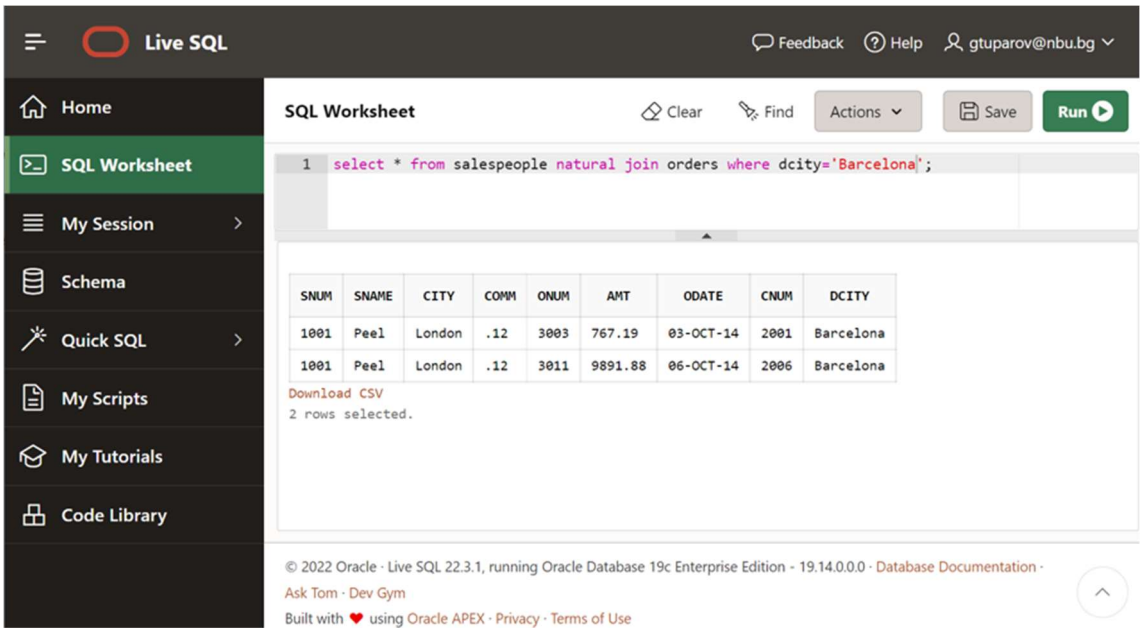


Figure 1: Oracle Live SQL query execution example using modified SCO database schema [11]

- **Oracle APEX [33]** (Fig. 2) is a rapid application development (RAD) environment over Oracle DB. APEX can be used successfully as a sandbox database with full DDL, DML and TCL functionalities, stored procedures and UDFs, but DCL statements cannot be used – the administration is through a web interface. The workspace created is not cleared after logoff and all currently created database objects and scripts can be stored for future use. In addition, one workspace can have many users, so this sandbox is also suitable for group database project development. Oracle APEX can be used also as university level sandbox service, because it is freeware.

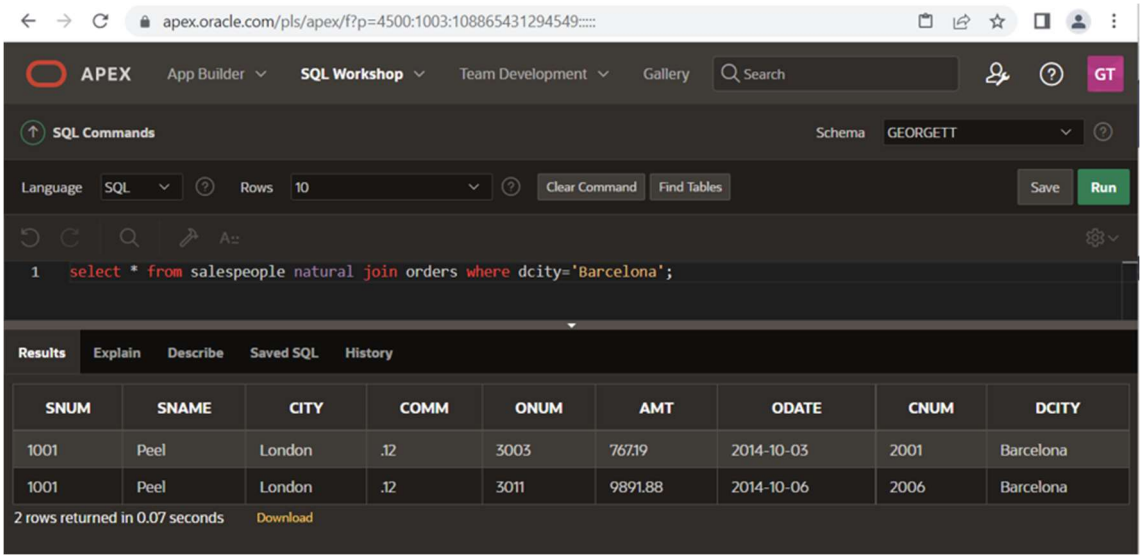


Figure 2: Oracle Apex query execution example using modified SCO database schema [11]

- **"Fiddle" like sandboxes** – db<>fiddle [34], SQL Fiddle [35] (Fig. 3), DB-Fiddle [36] etc. are popular free sandboxes with web client interface. DBMS used in the backend can be selected from the list and includes MySQL, Oracle, PostgreSQL, SQLite, MS SQL Server etc. according to the particular system. Registration is not needed and each new session starts with an empty database schema. Full set of DDL and DML queries can be executed, and created database schema, state and queries can be shared as links with other users.

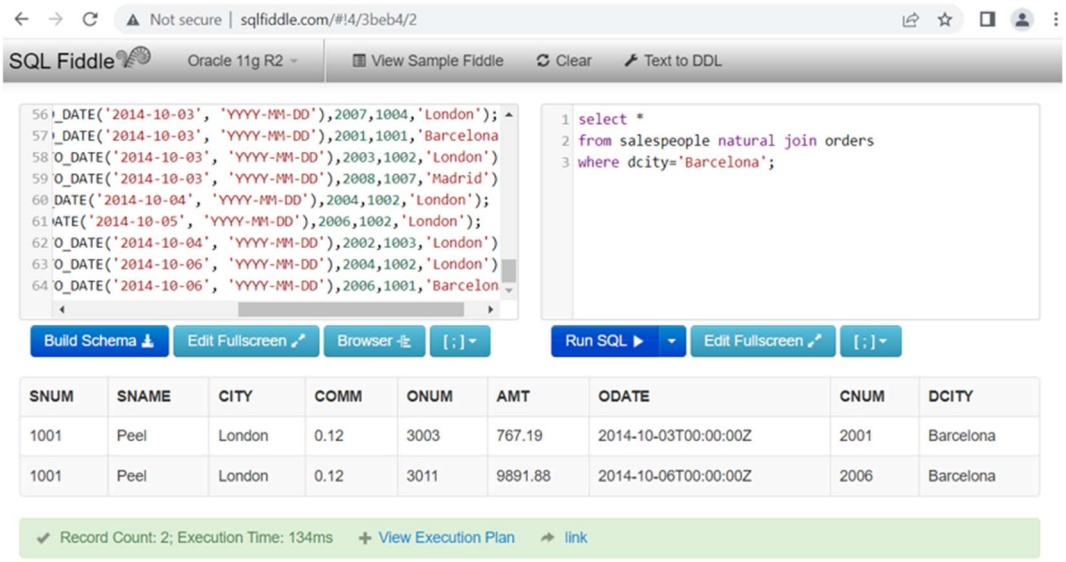


Figure 3: SQL Fiddle query execution example using modified SCO database schema [11]

3 Framework for comparing sandbox databases

Requirements for the sandbox database to be used in practical sessions and self-training in university database courses can be classified into four groups:

- **SQL functionality support**. The full list of SQL functionalities includes:
  a. DML statements (SELECT, INSERT, DELETE, UPDATE);
  b. DDL statements (CREATE, ALTER, DROP);
  c. TCL statements (BEGIN, END, COMMIT, ROLLBACK);

    d. DCL statements (GRANT, REVOKE);
    e. Procedural extensions.
- **Creating and recovering efforts**. Available procedures for:
    a. creating and recovering the database schema and state;
    b. recovering of the entire DBMS environment with database schema(s) and state(s) inside after wrong DCL operation.
- **Software/hardware efforts:**
    a. license type – free/commercial;
    b. install/managing skills needed.
- **Equality efforts**. Ways to guarantee the equal state of the DBMS/database schema/state for all of the students during class or individual training:
    a. centralized or personal;
    b. special skills/efforts needed.

In Table 1 below this framework is applied to part of the considered sandbox databases. MySQL Sandbox [25] and ClassDB [26] are classified as "University level services", because they have a centralized administration of the sandbox database. Oracle APEX is also not included in Table 1 because it can be classified as a free web service sandbox with the same functionalities like Oracle Live SQL, or as university level service in case of usage as university web service sandbox.

Table 1: Framework application

Requirements/ Sandbox	University level service	Local install	Local VM	Local Docker container	Oracle Live SQL	"Fiddle" like
SQL support						
DML	Y	Y	Y	Y	Y	Y
DDL	Y	Y	Y	Y	Y	Y
TCL	Y	Y	Y	Y	Y	Y
DCL	N	Y	Y	Y	N	N
Proc. extensions	Y	Y	Y	Y	Y	Q
C/R efforts						
DB schema/state	C	P	P	P	P	P
DBMS	C	P	P	P	P	P
S/H efforts						
License	F	F	F	F	F	F
Special hardware req.	N	Q	Q	Q	N	N
Install/managing skills	N	Y	N	Q	N	N
Equality efforts						
Centralized/Personal	C	P	P	P	P	P
Special efforts	N	Q	Q	Q	N	N

Legend: Y (yes), N (no), C (centralized), P (personal), Q (questionable, not clear), F (free or have to be free)

4 Conclusions

Practical training is an important part of the university IT or database courses. To be successful, SQL practical sessions have to be supported not only by appropriate database schemas, states and tasks, but also by a sandbox database where students are able to solve tasks, experiment with different approaches, create their own schemas etc. Also, the sandbox database has to be suitable for self-training, i.e. to be used on students' personal computers without any special hardware, software or licensing requirements and to have easy procedures for database recovery.

As is visible from the literature review, a lot of research is focused not on the development of suitable sandbox databases for this general usage, but more on the environment for guided self-training,

combined with automatic self-assessment and/or grading. These environments are not exactly sandbox databases in the meaning used in IT practice and restrict students' ability to experiment.

Using sandbox databases as university level services is a preferable solution, with rich SQL functionality and centralized administration of workspaces which guarantees all students to be in an equal state during class or individual training. Web-based sandboxes available for free are a good solution too, and combined with scripts for database schema and state creation, achieve a similar effect as university level services, but without financial cost for the university and special efforts for the students. For example, Oracle Live SQL and Oracle Apex were successfully used to support practical sessions and projects development in database courses during COVID-19 crisis at New Bulgarian University and at American University in Bulgaria.

Because of students' restrictions to execute DCL statements, solutions mentioned above are not applicable for practical training in database administration topics. For this purpose, sandbox databases in VM or Docker containers are applicable although they require extra skills and efforts from the students. In fact, this is not a real problem because database administration topics are directed to the advanced students.

5 References

[1] Taipalus, T., & Seppänen, V. (2020). SQL education: a systematic mapping study and future research agenda. ACM Transactions on Computing Education, 20(3), Article 20. doi:10.1145/3398377

[2] ACM, CS2013 Curriculum Guidelines for Undergraduate Programs in Computer Science, URL https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf, ACM, 2013, last accessed on 19.08.2022.

[3] ACM, IS2020 A Competency Model for Undergraduate Programs in Information Systems, URL https://www.acm.org/binaries/content/assets/education/curricula-recommendations/is2020.pdf, ACM 2020, last accessed on 19.08.2022.

[4] ACM, IT2017 Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology, URL https://www.acm.org/binaries/content/assets/education/curricula-recommendations/it2017.pdf, ACM 2017, last accessed on 19.08.2022.

[5] ACM, SE2014 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, URL https://www.acm.org/binaries/content/assets/education/se2014.pdf, ACM, 2014, last accessed on 19.08.2022.

[6] Elmasri, R., Navathe, S., Fundamentals of Database Systems, 7th ed, Pearson Education, (2016)

[7] Garcia-Molina, C., Ullman, J., Widom, J., Database Systems. The Complete Book. 2nd ed. Pearson Education, (2014)

[8] Coronel, C., Morris, S., Database Systems. Design, Implementation, &Management, 13th ed., Cengage Learning Inc. (2017)

[9] Silberschatz, A., Korth, H. F., Sudarshan, S., Database System Concept, 7th ed. McGraw Hill Education, (2020)

[10] Date, C. J., SQL and Relational Theory, 3rd edition, O'Reilly Media, Inc, (2015)

[11] Gruber, M., Mastering SQL, Sybex Inc. (2000)

[12] Röhm, U., Brent, L., Dawborn, T., Jeffries, B., SQL for data scientists: designing SQL tutorials for scalable online teaching, Proceedings of the VLDB Endowment, Vol. 13 Issue 12, August 2020 pp 2989–2992 doi:10.14778/3415478.3415526

[13] Silva, Y.N., Almeida, I., Queiroz, M., SQL: From Traditional Databases to Big Data, SIGCSE '16: Proc. of the 47th ACM Technical Symposium on Computing Science Education, Feb 2016, pp 413–418 doi:10.1145/2839509.2844560

[14] Sadiq, S., Orlowska, M., Sadiq, W., & Lin, J. (2004). SQLator – an online SQL learning workbench. Paper presented at the SIGCSE Bulletin (ACM, Special Interest Group on Computer Science Education), 36(3) pp. 223-227. doi:10.1145/1026487.1008055

[15] De Raadt, M., Dekeyser, S., & Lee, T. Y. (2006). Do students SQLify? improving learning outcomes with peer review and enhanced computer assisted assessment of querying skills. Paper presented at the

ACM International Conference Proceeding Series, vol. 276 pp. 101-108. doi:10.1145/1315803.1315821

[16] Brusilovsky, P., Sosnovsky, S., Yudelson, M.V., Lee, D.H., Zadorozhny, V., Zhou, X., Learning SQL Programming with Interactive Tools: From Integration to Personalization, ACM Transactions on Computing Education, Vol. 9, Issue 4, Jan 2010 Art. No.: 19 pp 1–15, doi:10.1145/1656255.1656257

[17] Joshua License. 2017. testSQL: Learn SQL the Interactive Way. In Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education – ITiCSE '17. ACM Press, New York, New York, USA, 376–376, doi:10.1145/3059009.3072991

[18] Mitrovic, A., & Martin, B. (2000). Evaluating the effectiveness of feedback in SQL-tutor. Paper presented at the Proceedings – International Workshop on Advanced Learning Technologies: Advanced Learning Technology: Design and Development Issues, IWALT 2000, 143-144. doi:10.1109/IWALT.2000.890591

[19] Mitrovic, A. (2003). An intelligent SQL tutor on the web. International Journal of Artificial Intelligence in Education, 13(2-4), pp. 173-197

[20] J. C. Prior, AsseSQL: an Online, Browser-based, SQL Skills Assessment Tool, ITiCSE '14: Proceedings of the 2014 conference on Innovation & technology in computer science education, June 2014, pp 327, doi:10.1145/2591708.2602682

[21] Kleerekoper, A. Schofield, SQL Tester: An Online SQL Assessment Tool and Its Impact, ITiCSE 2018: Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, July 2018 pp. 87–92 doi:10.1145/3197091.3197124

[22] C. Kleiner, C. Tebbe, and F. Heine, "Automated grading and tutoring of SQL statements to improve student learning," in Proceedings of the 13th Koli Calling International Conference on Computing Education Research, Koli, Finland, 2013, pp. 161-168, doi:10.1145/2526968.2526986

[23] A. Bhangdiya et al., "The XDa-TA system for automated grading of SQL query assignments," 2015 IEEE 31st International Conference on Data Engineering, 2015, pp. 1468-1471, doi:10.1109/ICDE.2015.7113403.

[24] Atchariyachanvanich, K., Nalintippayawong, S., & Permpool, T. (2017). Development of a MySQL sandbox for processing SQL statements: Case of DML and DDL statements. In Proc. of the 2017 14th International Joint Conference on Computer Science and Software Engineering, JCSSE 2017, doi:10.1109/JCSSE.2017.8025930

[25] Permpool, T., Nalintippayawong, S., & Atchariyachanvanich, K. (2019). Interactive SQL learning tool with automated grading using MySQL sandbox. 2019 IEEE 6th International Conference on Industrial Engineering and Applications, ICIEA 2019, 928-932. doi:10.1109/IEA.2019.8715175

[26] Murthy, S., Figueroa, A., Rollo, S., Toward a Large-Scale Open Learning System for Data Management, L@S '18: Proc. of the Fifth Annual ACM Conference on Learning at Scale, June 2018 Article No.: 16, Pages 1–4, doi:10.1145/3231644.3231673

[27] MySQL Community Server, URL https://dev.mysql.com/, last visited on 21.08.2022

[28] PostgreSQL, URL https://www.postgresql.org/download/, last visited on 22.08.2022

[29] MS SQL Server Express, URL https://www.microsoft.com/en-us/sql-server/sql-server-downloads, last visited on 22.08.2022

[30] Oracle DB Express, URL https://www.oracle.com/database/technologies/appdev/xe.html, last visited on 22.08.2022

[31] Bitnami by VMWare, https://bitnami.com, last visited on 21.08.2022

[32] Oracle Live SQL, URL https://livesql.oracle.com, last visited on 21.08.2022

[33] Oracle APEX, URL https://apex.oracle.com, last visited on 21.08.2022

[34] db<>fiddle, URL https://dbfiddle.uk/, last visited on 22.08.2022

[35] SQL Fiddle, URL http://sqlfiddle.com, last visited on 21.08.2022

[36] DB-Fiddle, URL https://www.db-fiddle.com, last visited on 21.08.2022