Zero Tolerance for Incorrect Data: Best Practices in SQL Transaction Programming

M. Laiho[1], C. Skourlas[2,α)] and D.A. Dervos[3]

[1]*HAAGA-HELIA UAS, IT Department, Ratapihantie 13, 00520 Helsinki, Finland*
[2]*Technological Education Institute of Athens, Department of Informatics, Aigaleo 12210, Athens, Greece*
[3]*Alexander Technological Educational Institute, IT Department, P.O. BOX 141, 57400 Sindos, Thessaloniki, Greece*

[α)]*Corresponding author: cskourlas@teiath.gr*

Abstract. DBMS products differ in the way they support even the basic SQL transaction services. In this paper, a framework of best practices in SQL transaction programming is given and discussed. The SQL developers are advised to experiment with and verify the services supported by the DBMS product used. The framework has been developed by DBTechNet, a European network of teachers, trainers and ICT professionals. A course module on SQL transactions, offered by the LLP "DBTech VET Teachers" programme, is also presented and discussed. Aims and objectives of the programme include the introduction of the topics and content of SQL transactions and concurrency control to HE/VET curricula and addressing the need for initial and continuous training on these topics to in-company trainers, VET teachers, and Higher Education students. An overview of the course module, its learning outcomes, the education and training (E&T) content, virtual database labs with hands-on self-practicing exercises, plus instructions for the teacher/trainer on the pedagogy and the usage of the course modules' content are briefly described. The main principle adopted is to "Learn by verifying in practice" and the transactions course motto is: "Zero Tolerance for Incorrect Data"

INTRODUCTION

According to the IS 2010 model, the database related courses comprise an essential part of Informatics education [1][2]. Longenecker et al. based on current governmental and academic surveys of industry explored the IT labour market expectations, and suggested that "considerably more technical focus as well as depth of learning may well be required to meet the needs of the professional community befitting of the Computer Information Systems (CIS) designation" [3]. They also proposed an extension of the IS 2010 model. Several surveys of the IT Labour Market revealed that the demand for IT professionals was considering DB Technology expertise as "required" or "strongly recommended". Concerning specific products, Oracle, SQL Server, DB2, and MySQL are the champions. Such conclusions seem to be valid for the IT labour market all over the world. A survey of the IT Labour Market, directed at the alumni who graduated from the Haaga-Helia University of Applied Sciences Degree Programme in Information Technology, also, revealed that the responders in working life valued databases as the most important topic, and DB technology is a "ticket" to the profession.

DBTechNet is a network of teachers, trainers and professionals from HE institutions, VET institutions, and ICT companies. To synchronize the coverage of basic database topics in VET and Higher Education (HE) curricula, DBTechNet has conducted a survey in a number of European VET institutions. The survey has revealed the absence of SQL transactions in many of today's VET curricula, although transaction technology is crucial for reliable data access. In response to this finding, the LLP "DBTech VET Teachers" project got initiated in order to introduce the topics of SQL transactions and concurrency control to VET teachers and trainers. This is along the lines of the need for a priority shift towards flexible and effective professional skills (re-)training programs that facilitate the adaptability of the workforce to new labour market trends in today's volatile EU economy.

International Conference on Integrated Information (IC-ININFO 2014)
AIP Conf. Proc. 1644, 113-119 (2015); doi: 10.1063/1.4907825

A major aim of the LLP "DBTech VET Teachers" programme is to enhance the collaboration between HE institutions, VET institutions, and ICT industry, by promoting the transfer of knowledge and skills between and across them. Equivalently, the aim is to facilitate the introduction of new topics and content to the HE and VET curricula; topics and content that reflect (a) the current trends in database technologies, and (b) the needs of the European labour market [4], [5], [6].

"DBTech VET Teachers" is a Transfer of Innovation project in Leonardo da Vinci Multilateral Project programme, partly funded by European Commission (https://sites.google.com/site/dbtechvet/project).

In the following sections the methodology is formulated following the five axes: Overview of the course module, Learning Outcomes, Course contents, Training, Recommendations and related educational material. DBTechNet perspective is given towards a framework for a Course Module on Database SQL Transactions. The course module is presented, and some best practices in SQL transaction programming are presented and briefly discussed. Conclusions are also presented.

TOWARDS A FRAMEWORK FOR A COURSE MODULE ON DATABASE SQL TRANSACTIONS

According to CS2013 (2013) [7], "The activity of developing or acquiring information technology applications for organizational and inter-organizational processes involves projects that define creative and productive use of information technology for transaction processing, data acquisition, communication, coordination, analysis, and decision support." "Information Management (IM) plays a critical role in almost all areas where computers are used" [8]. Transaction processing (IM7) is one of the elective units for IM. The description of the Transaction Processing unit of the new Computer Science Curricula (CS2013, 2013, page 117) [7] (see also [8], [9]) is given including topics (the fourth topic is a new one) and learning outcomes (the seventh outcome is a new one):

Transaction Processing Topics: Transactions Failure and recovery, Concurrency control, Interaction of transaction management with storage, especially buffering

Transaction Processing Learning Outcomes:

1. Create a transaction by embedding SQL into an application program. [Usage]
2. Explain the concept of implicit commits. [Familiarity]
3. Describe the issues specific to efficient transaction execution. [Familiarity]
4. Explain when and why rollback is needed and how logging assures proper rollback. [Assessment]
5. Explain the effect of different isolation levels on the concurrency control mechanisms. [Assessment]
6. Choose the proper isolation level for implementing a specified transaction protocol. [Assessment]
7. Identify appropriate transaction boundaries in application programs. [Assessment]

The DBTechNet Course Module not only conforms to, but it also extends the content of the above mentioned recommendations. It also conforms to the international practice focusing on a practitioners' approach based on the principles of "Learning-by-doing", and "Learn by verifying in practice". It is expected that the "learning-by-doing" and experimenting with the live problematic situations raise the motivation of the learners, and their interest to database technologies [4].

The following perspectives are considered for the formation of the Course module on Database SQL Transactions:

1) Course contents, topics, and learning outcomes are related to the needs of the labour market
2) Training, recommendations and related educational material are conforming to the principle of "Learn by verifying in practice".

THE COURSE MODULE ON SQL TRANSACTIONS

The DBTechNet framework forms a basis for a curriculum for VET Teachers Training and Higher Education Informatics Education, and it includes:

- The development of a course module on SQL transactions for VET teachers/trainers, HE graduates, and IT professionals.
- The creation of a transferable learning model for VET and lifelong learning training. One that tailors an up-to-date pedagogical approach by taking into account the needs for curricular reform, national qualification frameworks, and recognition of prior learning

The course module begins by considering transaction services in a single user environment, using the big picture with buffers and transaction logs, and focusing on how the fast and reliable rollback service is implemented. The learners need to understand that the application is responsible for implementing the transaction logic and for checking the diagnostics after every service request. Then, the course proceeds by considering the multi-user environment, where the solution of the single-user environment may not always work properly any more. A short list of topics addressed in theory as well as in practice includes:

- SQL transactions basics
- SQL error diagnostics in various DBMS products
- Concurrency issues (lost updates, dirty reads, non-repeatable reads, phantoms)
- ISO SQL isolation levels
- Concurrency control mechanisms: Locking Scheme Concurrency Control (LSCC), Multi-Versioning Concurrency Control (MVCC), Optimistic Concurrency Control (OCC)
- Best practices in SQL transaction programming

The SQL Transactions course motto is: "Zero Tolerance for Incorrect Data" and "Don't believe all that you read (in the textbooks or even in the DBMS product reference manuals …)"! Course participants are advised to experiment with and verify the services supported by the DBMS product used. They are also warned that the DBMS products differ in the way they support even the basic SQL transaction services.

The main principle adopted is to "Learn by verifying in practice". This approach is mainly tuned to the needs of the VET and lifelong learning/education course curricula. Examples of applying this principle include practicing with: sensitive updates and the lost update anomaly, the SQL GET DIAGNOSTICS statement, serializable schedules, and transaction concurrency control (CC) implementations. To provide learners equal possibilities for the experimental learning the free editions of current mainstream DBMS products have been bundled on a Debian Linux platform in free virtual machine appliance with free software tools. This appliance is available for downloading from the DBTechNet website (http://www.dbtechnet.org/) and it can be used in courses of any educational institute or as a self-study environment by interested individuals, including professionals in industry or currently unemployed people.

BEST PRACTICES IN SQL TRANSACTION PROGRAMMING: THE CLIENT-SERVER PERSPECTIVE

Today, the society infrastructure, business, and the life of citizens all depend on ICT and OLTP database application software utilizing services for reliable data storage and retrieval. Improper access to database services results into erroneous and/or missing data: missing orders, shipments, payments, double-bookings, double-invoicing, delays, erroneous information, etc. Transaction, as defined in textbooks, could be seen as a sequence of client operations set to comprise a logical unit of work and execute in an "all or nothing" fashion. As far as the state at which it leaves the database, a transaction appears to execute in isolation with respect to transactions of other clients that execute concurrently. Upon recovering from a system crash, the transaction leaves the database in a consistent state, i.e. either as it was prior to the starting of its execution, or as it is meant to become after the transaction executes to its entirety. The user perspective of transaction is related to business rules, requirements specifications, and use cases. Therefore, a business or user transaction relates to a data processing task that needs to be executed as a whole. It may involve one or more use cases and multiple SQL transactions. It is the application logic that implements user transactions as well as the programmatic retry actions/wrappers that resolve concurrency failures. Figure 1 depicts the conceptual level of the DB application architecture, concurrent transactions, and the related retry wrappers. User interface and application logic are separate. The client-server perspective is illustrated in Figure 2. The DB application perspective views SQL transactions as logical units of work (LUW), units of consistency, and units of recovery. A well designed transaction involves short, concise interaction instances with the DB server, and does not contain any user intervention in between. It inspects the diagnostic messages that are generated by the data access services, and incorporates the logic for taking appropriate action whenever necessary. It handles exceptions, and it is restarted each time it is rolled back due to some concurrency, or connection failure problem. Furthermore, it is restarted in a way that avoids livelocks.

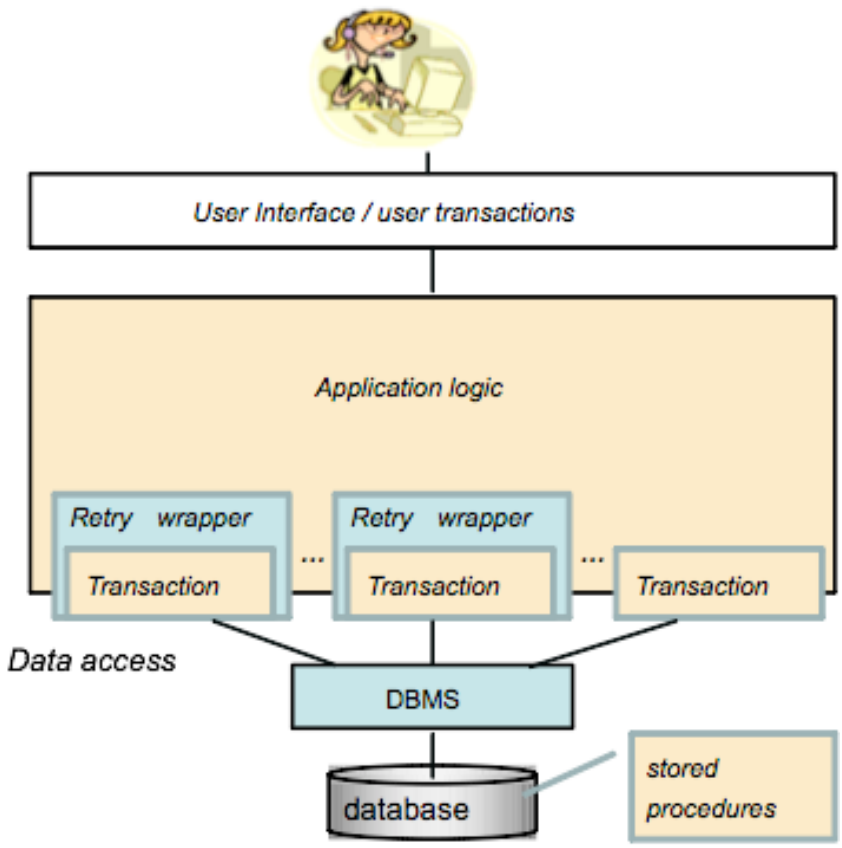

FIGURE 1. Conceptual level of db application architecture

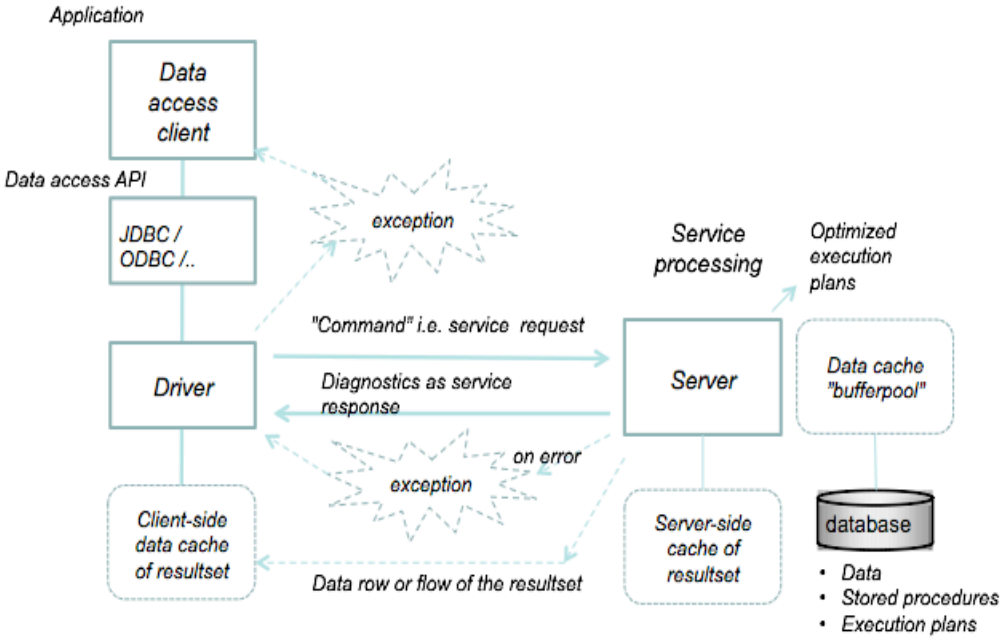

FIGURE 2. Client server perspective of transaction processing

Developers need to be aware of the way DBMS products behave, and of how the SQL diagnostics are inspected in the data access interface of the API used. For example, SQL standards have defined various diagnostic indicators some of which are implemented in most DBMS products, and some are not, while DBMS products have implemented indicators or exception handling methods of their own. Procedural extensions of SQL dialects are also different, especially in terms of error handling. In general, DBMS products differ from each other in terms of concurrency control services and transaction managing behavior, so it is important for reliability and performance purposes that the application developer knows the behavior of the DBMS to be used. As a conclusion:

Developers must be aware that, if the DBMS is lacking the expected support of basic concurrency control (CC) services, or the programmer is lacking the knowledge of how to make proper use of the DBMS supported CC services, then data update operations may end up corrupting the DB data content.

A series of hints are given for transaction isolation level:

SQL transactions, even in the same user transaction sequence, may have different reliability and isolation requirements. We must always define transaction isolation level in the beginning of every transaction.

According to the SQL standard READ UNCOMMITTED isolation level can only be used in READ ONLY transactions [10], but the products do not force this.

Reliability is the number one priority, before performance etc. but the default transaction isolation level used by DBMS products often favors performance before reliability! Proper isolation level should be planned with extra care, and the SERIALIZABLE isolation level with the ISO SQL semantics should be used, if the developer cannot decide which isolation level provides reliable enough isolation. It is important to understand that the SNAPSHOT isolation guarantees only consistent result sets, but does not preserve the database contents. If you cannot allow phantoms, and isolation levels in your DBMS support only snapshots, you need to consider the possibility of implementing explicit locking.

A series of hints are given for transaction applications:

A user transaction typically needs multiple dialogues with the database. Some of these dialogues will only collect data from the database supporting the user transaction, and as a final step of user transaction, some "save" button will trigger the SQL transaction which will update the database.

SQL transactions should not contain any dialogue with the end user, as this would slow down the processing. Since SQL transactions may get rolled back during the transaction, they should not affect anything else but the database. The SQL transaction should be as short as possible in order to minimize the concurrency competition and its associated blocking of concurrent transactions.

We have to avoid DDL commands in transactions. Implicit commits due to DDL may result into having unintentional transactions. Every SQL transaction should have a well-defined task, starting and ending in the same application component.

The technical context of an SQL transaction is a single database connection. If a transaction fails due to a concurrency conflict, it should, in many cases, have a retry wrapper in the application code with a limit of some 10 retries. However, if the transaction is dependent on the database content retrieved in some previous SQL transaction of the same user transaction, and some concurrent transactions have changed that content in the database and the current transaction therefore fails to update the content, then this transaction should not have a retry wrapper, but the control should be returned to the user for possible restart of the whole user transaction..

A new database connection needs to be initiated for the retry of the SQL transaction in case the connection was lost due to some network problems.

CONCLUSIONS AND FUTURE ACTIVITIES

In this paper, a framework for Database SQL Transactions Course modules for VET teacher training is presented and discussed. Besides its conformance to the ACM/IEEE CC 2001, and its 2013 revision, the proposed course module extends the content of these recommendations and conforms to the international practice. The DBTechNet course module has a great influence to the HE database curricula of the participating institutes that are interested in offering training for VET teachers/trainers, HE graduates, and IT professionals.

The framework can be also used for a comparative analysis of the course modules on SQL transactions of the Database Curricula in Higher Informatics Education. More precisely, the proposed DBTechNet course module and the Database Curricula of the participating Greek HE Departments in DBTechNet were comparatively analysed. We focused on the two oldest Departments of Informatics of the Technological Educational Institutes in Greece. Beyond

the core knowledge, which is common in their database curricula, each of the departments has incorporated the electives considered useful to its students depending on how it evaluates the market demand. The curricula of these Departments are under revision, in the framework of a new legislation for Higher Education in Greece, and the aim is to facilitate the introduction of new topics and content that reflects the new trends in database technologies, and the needs of the European labour market. In the future, we shall examine/focus on the potential influence of the DBTechNet Course Module to the HE database curricula of all the participating institutes that are interested in offering training for VET teachers/trainers, HE graduates, and IT professionals. We also plan to examine/focus on the potential influence of the DBTechNet course in the VET database curricula of the participating countries in the programme. The potential influence of the DBTechNet course in the database curricula of MSc programmes of the participating institutes will be also examined.

In this paper, we do not cover the topic of using SQL transactions in stored routines, since the stored procedure languages and implementations are different in different DBMS products. Some of the DBMS products in our DebianDB lab (http://www.dbtechnet.org/) do not allow COMMIT statements in stored routines. However, a transaction may get forced to rollback even in the middle of some stored routine, and this has to be handled also in the calling application code. We have a plan to cover this topic in the future.

Moreover, future researchers should also consider IT issues such as information security [11],[12] as well as other influential factors of the teaching quality provided (e.g. culture, leadership, commitment and satisfaction [13],[14]).

ACKNOWLEDGEMENTS

This research is partly funded by the European Commission's Lifelong Learning Programme. "DBTech VET Teachers" is a Transfer of Innovation project in Leonardo da Vinci Multilateral Project programme, and funded by European Commission and project partners.

REFERENCES

1. ACM/IEEE: Computing Science Curriculum 2008,: An Interim Revision of CS 2001. Retrieved on 7 July 2013 from: http://www.acm.org//education/curricula/ComputerScience2008.pdf
2. Topi, H., Valacich, J-H,, Wright, R.T., Kaiser, K.M., Nunamaker, J.F., Sipior, J.C., de Vreede, G.J. (2010). Curriculum Guidelines for Undergraduate Degree Programs in Information Systems, ACM & AIS, http://www.acm.org/education/curricula/IS%202010%20ACM%20final.pdf
3. Longenecker, H.E., Feinstein, D.L., Babb, J.S. (2013). Is There a Need For a Computer Information Systems Model Curriculum?, Proceedings of the Information Systems Educators Conference ISSN: 2167-1435, USA, v30 n2528
4. Dervos D.A., Laiho M., Aldana-Montes J., Riihelä P., (2013), A DBTechNet Project for VET Teacher Training on Database SQL Transactions,Balkan Conference in Informatics (BCI'13), pp. 149-155, Thessaloniki, Greece, ACM 2013 ISBN 978-1-4503-1851-8,
5. Laiho, M., Dervos, D.A., Aldana-Montes, J.F., Laux, F. (2010). DBTech EXT: Education and Hands-on Training for the Database Professional. ADBIS'10, pp 15-22, Springer-Verlag.
6. Laux, F., Laiho, M., Connolly, T. (2012). E-learning with Hands-On Labs in Higher European Education. ICIW 2012, pp 231-237.
7. ACM & IEEE-CS, The Joint Task Force on Computing Curricula, (2013), Computer Science Curricula 2013, Final Report 0.9 (Pre-release version), http://ai.stanford.edu/users/sahami/CS2013/final-draft/CS2013-Final-v0.9-prerelease.pdf
8. Nasiopoulos K. Dimitrios, Damianos P. Sakas, D.S.Vlachos (2014). Analysis of Strategic Leadership Simulation Models in Non-profit Organizations. Procedia - Social and Behavioral Sciences (pp. 276-284)
9. The Joint Task Force on Computing Curricula, Association for Computing Machinery and IEEE-Computer Society, Computer Science Curricula 2013, Final Report 0.9 (Pre-release version), October 2013, http://ai.stanford.edu/users/sahami/CS2013/final-draft/CS2013-Final-v0.9-prerelease.pdf
10. Sahami, M., Roach, S.; LeBlanc, R., Special session - The CS2013 computer science curriculum guidelines project, Frontiers in Education Conference (FIE), 2011, vol., no., pp.F2B-1,F2B-3, 12-15 Oct. 2011 doi: 10.1109/FIE.2011.6143103

11. Melton, J. & Simon, A. R., SQL:1999: Understanding Relational Language components, Morgan Kaufmann, 2002
12. Metalidou E., Marinagi C., Trivellas P., Eberhagen N., Giannakopoulos G., Skourlas C., (2014) Human factor and information security in higher education, *Journal of Systems and Information Technology*, 16(3), 210 – 221.
13. Metalidou E., Marinagi C., Trivellas P., Eberhagen N., Skourlas C., Giannakopoulos G., (2014) The Human Factor of Information Security: Unintentional Damage Perspective, *Procedia - Social and Behavioral Sciences*, 147, 424-428.
14. Trivellas P. and Dargenidou D. (2009), Organisational Culture, Job Satisfaction and Higher Education Service Quality. The case of Technological Educational Institute of Larissa, *the TQM Journal,* Vol. 21 No. 4, pp. 382-399.
15. Trivellas P and Dargenidou D. (2009) Leadership and Service Quality in Higher Education: The case of the Technological Educational Institute of Larissa, *International Journal of Quality and Service Sciences,* Vol. 1, Issue 3, pp. 294-310.