QL-RTDB: QUERY LANGUAGE FOR REAL-TIME DATABASES

Cicília R. M. Leite, Yáskara Y. M. P. Fernandes and Angelo Perkusich
Electrical Engineering Department
Federal University of Campina Grande, Campina Grande - PB

Pedro F. R. Neto
Informatic Department
State University of Rio Grande do Norte, Mossoró - RN

Maria L. B. Perkusich
Informatics and Statistics Department
Catholic University of Pernambuco, Recife, PE

Keywords: Real-Time Systems, Real-Time Databases, Query Language.

Abstract: In the last years, many research directed for concurrency control mechanism, scheduling policy and query language in real-time database are being done. This paper presents an extensions of standard query language (SQL), called SQL-99. SQL-99 includes oriented-object functionalities. The query language for real-time database (QL-RTDB) to provide support the real-time requirements imposed the both data and transactions. A prototype implementation of the QL-RTDB is also shown.

1 INTRODUCTION

Real-time databases (RTDB) are typically used to manage applications with critical data. Among these application is programmed stock trading, medical patient monitoring, automated manufacturing, and military command and control (PRICHARD, 1995). The RTDB project has to take in consideration characteristics of traditional database system (DBMS), and of real-time system. The RTDB must keeping data integrity, and to guarantee timing constraints to both transactions and data (STANKOVIC et al., 1999).

Consequently, the data must not only be logically consistent, but temporally consistent as well for these applications. This is due to the fact that data used by critical time applications must closely reflect the current state of the environment application.

However, although some researches directed toward real-time database, new approach must be providing by these, related the concurrency control, scheduling policy and query language (TESANOVIC et al., 2002; FERNANDES et al., 2004; LINDSTROM, 2002; PRICHARD, 1995).

In the query language conception, the data definition, data manipulation, and transaction control must to be provided. Data definition is made through of the data definition language (DDL). Data manipulation is realized through of the data manipulation language (DML). The transaction control includes commands for specification of transactions (ELMASRI and NAVATHE, 2004).

However, the available query language not supports complex applications with timing constraints. In this way, the definition of a query language becomes important to support such restrictions (SUDARSHAN et al., 2001). The objective of this work is define a query language to support requirements of timing to both data and transaction, called QL-RTDB. This will consist in extending SQL-99 (O´NEIL and O'NEIL, 2000; STEPHENS and PLEW, 2003; STEEB, 2004).

This work is organized as follow: in Section 2, we present related works. In Section 3, we describe the basic characteristics of RTDB. In Section 4, we show the SQL-99. In Section 6, we define an extension of the SQL-99. Finally, in Section 7, summarizes the main ideas of the paper and describes futures work.

2 RELATED RESEARCH

In the context of RTDB, a query language, called real-time SQL (RTSQL), was developed in (PRICHARD, 1995) as an extension of standard SQL-92. SQL-92 is a language that allows us to query and manipu-

R. M. Leite C., Y. M. P. Fernandes Y., Perkusich A., F. R. Neto P. and L. B. Perkusich M. (2005).
QL-RTDB: QUERY LANGUAGE FOR REAL-TIME DATABASES.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 420-423
DOI: 10.5220/0002524404200423

late data on relational database system (O´NEIL and O'NEIL, 2000). RTSQL is defined to support real-time queries, including extensions that specify the temporal consistency of the data and transactions. A prototype developed in C, was implemented to validate and to test the correctness of them.

In 1999, a new standard, called SQL-99, was published by American National Standards Institute (ANSI) and International Organization Standardization (ISO) for the object-relational model. This standard extends the SQL-92 to include oriented-object functionalities (O´NEIL and O'NEIL, 2000). However, the SQL-99 does not consider the definition and manipulation of data with timing constraints. In this way becomes necessary the development of extensions for the SQL-99 in order to attend the timing constraints.

3 REAL-TIME DATABASE

A real-time database system has three distinguishing features: the requirement of temporally consistent data, the requirement of timing constraints on execution, and the requirement that certain execution exhibit predictable timing behavior. These features are useful to time critical applications that need to collect, modify, and retrieve shared data. Support for these features will add new requirements to the DDL and DML of the database query language (PRICHARD, 1995).

Like a conventional database management systems, an RTDB must process transactions and guarantee that the database consistency is not violated, besides to keep time constraints or deadlines for transactions as well as the timing validity of the data (BESTRAVOS et al., 1997). The performance goal for these systems is usually expressed in terms of the desired average response times, rather than constraints associated to individual transactions. Thus, when the system makes scheduling decisions, individual real-time constraints are ignored (STANKOVIC et al., 1999).

The temporal consistency can be measured through of the absolute consistency and of the relative consistency of the data. The absolute consistency is the measure between the state of the environment and as it is reflected in the database. The relative consistency is the measure between the data that are used in the computation of other data (Bestravos et al., 1997).

The real-time data can be defined for tuple ($nr, vrr, avir, tsr, impr$), where: nr is the data item processed; vrr is the value that is updated for the item in server; vrr is a exclusive field of server and its value is obtained of the environment; $avir$ is the absolute validity of data, i.e., the time that the data is considered valid after transaction begins; tsr is the timestamp of data; $impr$ is the exported collected imprecision (Ribeiro Neto et al., 2004).

In this work, we will consider that transaction are defined by the quadruple (tl_i, tc_i, pr_i, pe_i), where: tl_i is the transaction release time, that is, the moment in which all the necessary resources to the execution of the transaction τ_i are available. Starting from this instant of time the transaction is ready to be executed; tc_i defines the transaction computation time, that is, the processing time necessary to execute it; pr_i defines the maximum transaction execution period, and; pe_i defines the periodicity of the transaction.

4 SQL-99

Committee ANSI/ISO SQL-99 introduced a standard query language that supports the object-relational model. The SQL-99 to provide a series of new capabilities: user types defined; collection types, such as, vectors or table nesting; and methods and user defined functions (O´NEIL and O'NEIL, 2000; STEPHENS and PLEW, 2003).

User Defined Types (UDT) - It is a nominated group of attributes of different types of data (objects) that can be used to define tuples of a table or the type of some attribute of a table.

Collection Types - A column value (on a single row) to contain a set (or some other type collection, such as an array) of row-like values. In some systems a single column value can itself hold a table, a feature know as table nesting.

Methods and User Defined Functions (UDF) - In object-oriented languages such as Java, any private data in a object can be accessed only via object methods, such methods are functions that can be invoked to operate (ECKEL, 2002) on one particular object. Normally, the set of methods for an object will deal with all properties of the object that can be treated in isolation (O´NEIL and O'NEIL, 2000).

Therefore, the specification SQL-99 extends the SQL-92 standard including object-oriented functionalities (O´NEIL and O'NEIL, 2000). Some commercial DBMS as: DB2, PostgreSQL, Oracle9i, among others, already implement SQL-99.

5 SQL-99 EXTENSION

Real-time programming languages must provide mechanism to defined timing constrains, used to express the start times, periodicity, release time and deadlines (or completion times) of transactions. To satisfy such constraints, the query language also must to incorporate these properties.

Considering the exposed above, in this paper we will extend the syntax of the DDL of the SQL-99 to consider beyond the definitions of the logical constraints, the definitions of timing constraints of the data, such as: timestamp, and absolute validity of data. The syntax of the DML and the transaction control will be extended in order to consider timing clauses, such as: start time, release time, transaction periodic, computational time, deadline, among others.

6 QL-RTDB

The QL-RTDB was implemented in the java programming language. Java is multiplatform, allowing that programs can be executed in machines with different operational systems, guaranteeing the portability of any application implemented in this language (ECKEL, 2002). Specifically, we utilized the real-time specification for Java (RTSJ), so that this available extensions for real-time applications (RTS, 2003).

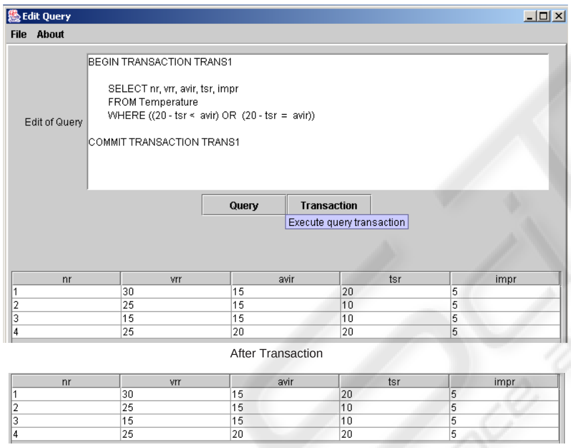

Figure 1: Query Transaction

For the QL-RTDB implementation, the DB2 was utilized as the data repository. For this, was necessary to establish the connection of the QL-RTDB application with the database through of driver JDBC (Java DataBase Connectivity). Then, we load driver for the database in the Java Virtual Machine (JVM) of the application. The driver is registers to *DriverManager* and is available for the application. We use the classe *DriverManager* to open a connection with the database. The interface *Connection* it assigns an object, in the case *con*, to receive the connection established.

We present the QL-RTDB language through of Figure 1. This shows the transaction T1 that is a query transaction. The time parameters and logical parameters are passed through of clauses for the RTDB, where the information will be stored in the database, for latter query, updates and reports.

The query transaction T1 verify the absolute consistency of the data through of expression $((t - tsr) \leq avir)$, where t is the current time; tsr is the data timestamp; $avir$ is the data absolute validity. The query result, illustrate in the table of Figure 1, show whole data temporality consistent. The table field was defining in the Section 3.

7 CONCLUSION

In this article we made a state of art survey from about query language, where each work has its specification and its contribution. Thus, we search extract the contributions of each work and to adapt to our context.

We define a query language for real-time database, called QL-RTDB, that providing support the real-time requirements imposed the both data and transactions. A prototype implementation of the QL-RTDB is also shown. We choose the programming language java for real-time, due to this permitting extension, and oriented-object. As future work, we will add functionalities with timing parameters. These functionalities are specified through of the DDL, DML and transaction control.

REFERENCES

(2003). *The Real Time Specification for Java*. Sun Microsystems. http://rtsj.dev.java.net.

BESTRAVOS, A., LIN, K., and H.S., S. (1997). Real-Time Database Systems: Issues and Applications. *Kluwer Academic Publishers*.

Bestravos, A., Lin, K.-J., and S.H., S. (1997). *Real-Time Database Systems: Issues and Applications*. Kluwer Academic Publishers, Boston.

ECKEL, B. (2002). *Thinking in Java*. Prentice-Hall, 3rd edition. Available in web, http://www.bruceeckel.com, Access in May.

ELMASRI, R. and NAVATHE, S. B. (2004). *Fundamentals of Database Systems*. Addison Wesley, international edition.

FERNANDES, Y., RIBEIRO NETO, P., PERKUSICH, M., and PERKUSICH, A. (2004). Implementation of transactions scheduling for real-time databases management. *To appear in IEEE Systems, Man and Cybernetics Conference*.

LINDSTROM, J. (2002). Integrated and adaptive optimistic concurrency control method for real-time databases. *VIII International Conference on Real-Time Computing Systems and Application*. VIII International Conference on Real-Time Computing Systems and Application.

O´NEIL, P. and O'NEIL, E. (2000). *Database Principles, Programming, Performance*. Morgan Kaufmann, 2 edition.

PRICHARD, J. (1995). *"RTSQL": Extending The "SQL" Standard to Support Real-Time Databases*. PhD thesis, Department of Computer Science and Statistics, University of Rhode Island.

Ribeiro Neto, P., Perkusich, M., and Perkusich, A. (2004). Scheduling real-time transactions for sensor networks applications. In *Proceedings of the 10th International Conference on Real-Time Computing Systems and Applications (RTCSA)*, pages 181 – 200, Godenburg, Suécia. Lecture Notes Series by Springer-Verlag.

STANKOVIC, J., SON, S., and HANSSON, J. (1999). Misconceptions about real-time databases. *IEEE Computer*, 32(6):29–36.

STEEB, W.-H. (2004). Relational and object-oriented. Available in web, http://issc.rau.ac.za/downloads/downloads.html, Access in May.

STEPHENS, R. and PLEW, R. (2003). *Sams Teach Yourself SQL in 24 Hours*. Sams, 3 edition.

SUDARSHAN, S., KORTH, F. H., and SILBERSCHATZ, A. (2001). *Database System Concepts*. McGraw-Hill Education, 4 edition.

TESANOVIC, A., HANSSON, J., NYSTROM, D., and NORSTROM, C. (2002). Embedded databases for embedded real-time systems: A component-based approach. Technical report, Dept. of Computer Science, Linkoping University, and Dept. of Computer Engineering, Malardalen University.