Development of a Relational Database Management System.

Universidad Tecnológica Nacional
Facultad Córdoba

Laboratorio de Investigación de Software
Departamento de Ingeniería en Sistemas de Información

Anabel Natalia Ruiz — anabelru@bbs.frc.utn.edu.ar
Ing. Calixto Maldonado — calixto@bbs.frc.utn.edu.ar
Ing. César Ignacio Martínez Spessot — spessot@bbs.frc.utn.edu.ar

Universidad Tecnológica Nacional - Facultad Regional Córdoba
Maestro M. López esq. Cruz Roja Argentina
Ciudad Universitaria - C.P. (5016)
Córdoba - República Argentina.

ABSTRACT

The studies of Relational Data Base Management Systems and Databases constitute nowadays a main trend of investigation in large businesses and universities of the entire world. The creation of a data base engine keeps being a challenge since there are aspects in which have not been achieved a final solution. For example, the requirements of storage of images, video, sound, etc. The purpose of this project is to implement a Relational database Manager System (RDBMS) and an Interpreter of Data Query Language that can be utilized in investigation and data processing development of Applications. It is important to count on a RDBMS developed in our universities so they can study an do modifications, which is practically impossible with the commercial products due to that does not be available neither their Source code neither the documentation detailed neither the necessary training. Our investigation includes the development of innovating algorithms for access and maintenance of the data and procedures; techniques and tools to carry out Datawarehouse and DataMining, offering a good fulfillment as for time of answer and extensive capacity of backup to requirements such as operational as strategic (Decision Support Systems).

Keywords : databases, RDBMS, SQL, Query Language Interpreter, Architecture of a DBMS, DataWarehouse, DataMining, Relational databases, SMBDR, Relational Model, Architecture of a RDBMS, Components of a RDBMS.

1. INTRODUCTION

A database (BD)[1], is in essence a collection of information that will exist in a period of time, information that is managed for a RDBMS. Ullman and Widom [1], also establish that an Administrative System of database owes:

- Allow the users create new bases of data and specify its plan (logical structure of the data), using a language specialized called Database Definition Language
- To give to the users the ability of consulting the data (a query is an answer of the base of data to a question upon the data) and to modify the data, using an appropriate language, often called Data Management Language.
- To bear the storage of a large quantity of data upon a great period of time, keeping the security in an accident or a use not authorized and to permit the efficient access to the data for the queries or modifications to the Database.
- To Control the accesses to the data in simultaneous form of various users, without permitting that the actions of one affect other users and to impede that simultaneous accesses damage the organization's data.

These are the main objectives of our investigation. In which at the beginning started with the definition of a plan and a language that permit to carry out the definition and database queries. The Language implemented for the communication was SQL for which we design its syntax by means of the Formalism of Backus Naur (BNF). The Notation BNF or free grammar of context handles a syntax of a very simp le since alone form is interested in as are structured the distinct prayers being basing on diverse rules.

This permitted us to create an interpreter of the language SQL and utilize the same one as native way of communication with the database engine. At present our focus is to create and to improve algorithms to bear storage of large quantities of data and concurrent access.

2. ARCHITECTURE OF THE RDBMS

The main components of the Relational Data Base Management System (see figure 1) :

- IDE of Development
- Interpreter of Language
- Engine Basic of Base of Data
- Objects of Messages

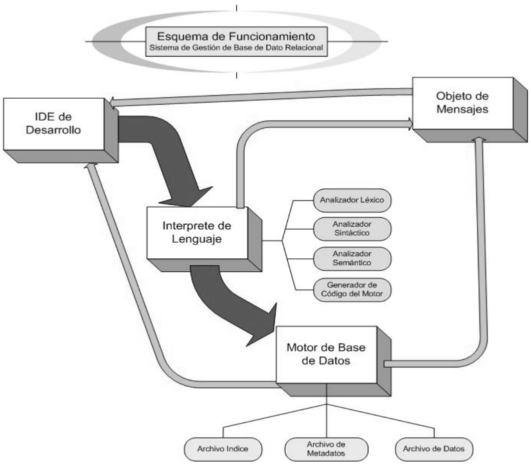


Figure 1: DBMS Schema

We make a Development IDE in order to the user of the engine had a framework with whom interact. This one offers several basic options as open save and execute a file SQL as also text edition options (undo, to cut, to hit, etc.) with the objective of improving the relation with the user. The user has the possibility to keep the query carried out in order to returning it to execute in another moment.

Besides, the encapsulation in an object of the Development IDE will permit us in a future to be able to be replaced by another component as an ODBC driver, necessary to integrate the engine with other languages as Visual C++, C#, Kylix, and others.

The Development IDE sends the code of the program to the Queries Language Interpreter with the objective that It carry out the Analysis Dictionary, Syntactic and Semantic corresponding, finally and in case that not no error exist, translates to code base of the Engine. If in the course of the different analysis there was an error, it is immediately informed of the same one to the Object of Messages.

With the code Base, the Database Engine carries out the corresponding petition and if there was no error, the result is transmitted to the Development IDE for its subsequent notification to the user. In case that the execution of some instruction generates an error, said problem is sent to the Messages Object.

Three files by each Table are defined: one of metadata, another of data and finally one of indexes. The indexes utilized for the efficient recuperation of the information are tree B+ and digital tree. The Messages Object has the function to receive all the problems occurred as also to report (not the results) the execution of instructions. At present we have aggregated a module with the objective that this object can send packages TCP-IP to transmit the information by the network.

3. IMPLEMENTATION OF THE DATA BASE ENGINE

The implementation of the engine has been carried out in the language C++. The architecture of the RDBMS has been developed in objects and interfaces private for each one; an object for each component exists explicated previously in Architecture of the RDBMS. At the same time each one of these possesses an assembly of classes to bear the structures (general database, tables, indexes, elements of language, instructions, etc.) It is permitted to define various types of fields: character, date, entire, logical, floating and binary. This last one this prepared to keep images, video, sound or another program executable.
The structure of the file of metadata is carried out in the following way:

```
Struct metadata
 {
          char name [30];  //Name of the field
          int   type;       //Type   of   field:   1-date,
                              // 2 - character,etc.  According to
                              //enum of the headline

          int integer; //so Great of length of the field
          int  decimals;  //so  Great  of  the  length  of  the
                             // part decimal (used only in
                             // floating)

          bool iskey;  // if is a key field or not

};
```

The maximum length for a name of a field can be observed is 30 characters. With the file of metadata in memory the file of data can raise himself which is stored for bytes without a structure predetermined.
The System handles three basic files to store, recuperate and modify the data and the structure of the table, which are: file of data, metadata and procedures or programs. Besides it has been incorporated two types of indexes to select of, according to the type of data that is desired to order and to seek, to know: tree B + and digital tree.
The interface of the engine counts on a series of windows for a correct edition, interpretation, and correction of errors of the sentences SQL. The bar of tools counts on the typical edition buttons
(copy, paste, undo, etc.), to open, to keep, new, aid, etc. The window of edition is simply a window with the elementary characteristics of a simple text editor. Here it is where the instructions are written in SQL, such as INSERT, UPDATE, SELECT, DELETE, CREATE TABLE, etc. It fits to clear up that the sentences borne by the engine are based on the characteristics of ANSI SQL. The viewer of state of the program shows the errors found (indicating position and type of error) and the actions carried out (for example: have been erased three registrations; the table has been created, etc.) This it serves to locate easily the committed errors and to proceed to its correction.

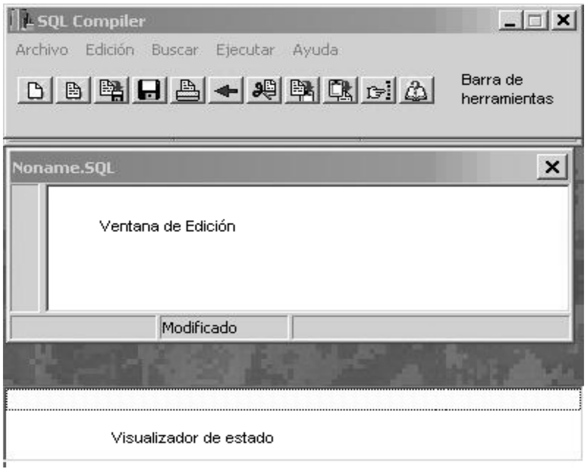
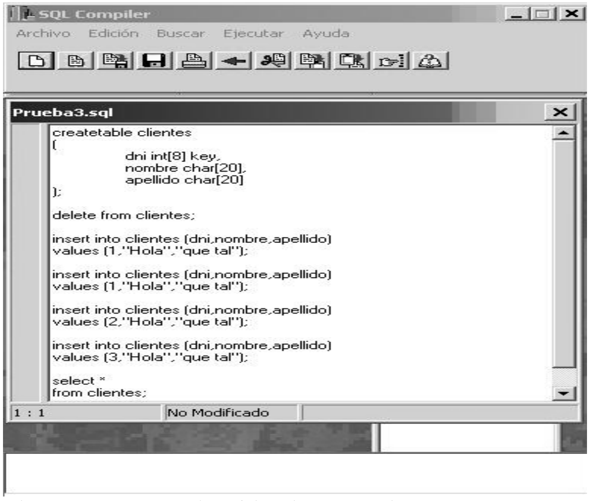


Figure 2: Interface of the database Engine

To illustrate the operation of the engine we show a small example in the following figure:

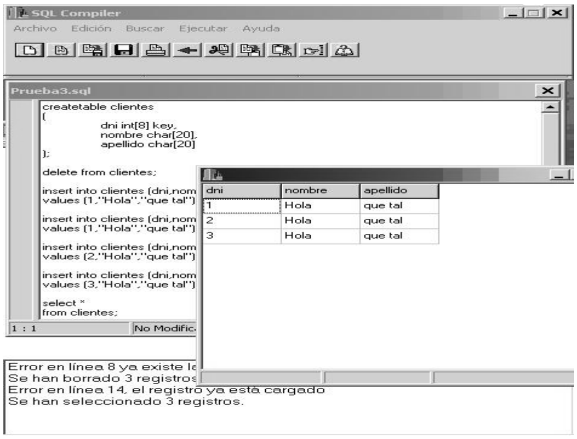


```
createtable clientes
(
            dni int[8] key,
            nombre char[20],
            apellido char[20]
);

delete from clientes;

insert into clientes (dni,nombre,apellido)
values (1,"Hola","que tal");

insert into clientes (dni,nombre,apellido)
values (1,"Hola","que tal");

insert into clientes (dni,nombre,apellido)
values (2,"Hola","que tal");

insert into clientes (dni,nombre,apellido)
values (3,"Hola","que tal");

select *
from clientes;
```

Figure 3 : *Example of* database Engine

The first sentence creates the table clients with three fields (id, name and surname) and establishes a key field (id); the second erases all the registrations of the clients table; the third, quarter, fifth and sixth instruction insert new registrations to the same table; and the last sentence carries out a query of all the rows of the clients table. Immediately after initiating the execution of the code, begins to function the interpreter of query, the same one goes examining each sentence and analyzes it lexically, syntactically and semantically. When it concludes with said analysis, a message appears indicating that the program is lexically, syntactically and semantically correct, in case of have found no error. If some contingency is produced, the interpreter prints it in the state viewer; the same thing does when brings up to date a Table, whether for erased, selection, modification or insertion of registrations. In the Figure 4 the result after the execution of the sentences is illustrated SQL. Here we can appreciate that during the compilation 2 errors were found: the first one doe's reference to that the

Table already has been created and the second to that the registration already was loaded (to maintain the integrity referential, that is to say, themselves not key repeated primaries are permitted). In the viewer also two lines are presented indicating that three registrations were erased and that three registrations were selected; this is used to reveal the quantity of registrations selected, erased, inserted or modified; also information about the Tables can appear. When we have a sentence SELECT, the registrations that comply with the conditions stipulated in the WHERE, are shown in a new window that contains the table with the fields selected, its respective names and the listing of the chosen registrations.



```
createtable clientes
(
            dni int[8] key,
            nombre char[20],
            apellido char[20]
);

delete from clientes;

insert into clientes (dni,nom
values (1,"Hola","que tal")

insert into clientes (dni,nom
values (1,"Hola","que tal")

insert into clientes (dni,nom
values (2,"Hola","que tal")

insert into clientes (dni,nom
values (3,"Hola","que tal")

select *
from clientes;
```

dni	nombre	apellido
1	Hola	que tal
2	Hola	que tal
3	Hola	que tal

```
Error en línea 8 ya existe la
Se han borrado 3 registros
Error en línea 14, el registro ya está cargado
Se han seleccionado 3 registros.
```

Figure 4: Answer of *Prueba3.sql*

4. PEDAGOGICAL USE OF THE DATA BASE ENGINE "TECNODB"

Upon presenting the development of this Relational Database Management System to the educational community, arose that this development would be able to serve like framework of a learning methodology of the basic, and advanced, notions of Databases.

This methodology, based on this software tool, would permit to direct the fundamental process of learning for the development of applications, which is head competence in the professional activity of the data processing.

This pedagogical objective did us to think about a name for the development and arose for "TecnoDB" as the name that would carry this engine. The development of this methodology would be able to be implemented in a web environment, being able to consider it like an e-learning tool.

The project would have like objectives related the advance in the construction of the RDBMS, already developed in its first's aspects, and as complement to build a SQL-Query Generation Tool.

To reach these two initial objectives will permit us:

The engine advanced, to show the operation of a mechanism of administration of a Database that permit us to know in practical form the resolution of the task entrusted, the selection, insertion, updating and table's rows elimination.

The generator of SQL queries will help us to understand the language SQL, to see an example of syntax and to resolve the requirements. Also it will permit to facilitate the learning of the development of queries, sentences of updating and erased that they utilize the indexes, to the power to compare it with its turned out, being this turned out a 'suggestion' to develop the sentence that the problem requires.

5. BASES OF THE SQL QUERY GENERATOR "TECNOGEN"

The SQL Query Generator, was based on the metadata of the dictionary "TecnoDB" and to offer to the user the tables and views available so that could be elected of them to show, the columns by which to filter rows, to group rows, to filter groups and to sort them. All the selection and display of the result was carried out in an Graphic interface. The result is the required Select statement, according to it defined by the user.

The perception upon the lack of tools that attend the user of the engines developed to write sentences SQL attended by a tool, and its use to evaluate the sentences that the student enter, as method of diagnosis to achieve a correction, went the antecedents for the development of the idea. Also the visualization of the use of this tool to attend the final user so that, with concepts but general still, that is to say that do not have to elect all the elements that needs to relate, obtain you consult complex of high aggregate value. As objective of greater reach we consider deepening the techniques of Dates Mining to find hidden correlation among variables of the business.

We understood that would be an aid to exceed the first's barrier, inherent in the control of the language SQL for the practical learning of the students without experience. We think about the name of the generator of Consultations and deriving from the Engine arose "TecnoGEN"

We saw like necessary characteristics for "TecnoGEN" that should be Multiplatform, referring us to multiplatform to the possibility of to be utilized independently of the Operating system of the client's equipment.

To be Multiplatform, is the characteristic one considered as one of the factors of success of the contents published in Internet by the "http" and the Browsers, that is to say the possibility to agree all kinds of contents, text, graphics, video, sound, independent of the machine and operating system that the use navigate.

Besides the applications developed in languages as Java do of its characteristic one multiplatform the main flag of its convenience.

This characteristic of Multiplatform arose then as the major value, aside from the to function correctly, of "TecnoGEN".

Originally we think that “TecnoGEN” should be capable to generate queries for distinct bases of data you relate them in the market, but we decide to limit this yearning to that function with the Engine presented in this job, “TecnoDB”.

Languages to utilize

The objectives of Multiplatform of execution, orient us to the family of C, that is to say C, C + + and Java as the languages that will permit to build “TecnoGEN”, with visual characteristics, with connection to the databases and that they can be executed in any operating System with the adequate installation, that is to say predicting the installation of the C libraries, or of the corresponding Java Virtual Machine.

Final words upon the generator of Consultations SQL “TecnoGEN”

A priori, we consider valuable the development of “TecnoGEN”, but we predict to seek comparable antecedents to be able to improve the requirements of the product.

In the plan of the project we consider necessary to enunciate the framework conceptual of what wants itself to teach, and to develop the pedagogical strategies to develop the e-learning environment that use the tools “TecnoDB” and “TecnoGEN”.

After the studies we have predicted the development of a Prototype in Java as an objective for March 2003 to have the version Alfa of “TecnoGEN”.

6. CONCLUSIONS

The Databases are a fundamental part of the Systems of Information, the same are considered the angular stone for the good operation of the modern institutions besides the quick one adaptability to the changes of the medium environment and processes of business such as production of services, logistic, takes of strategic decisions, etc. Argentina spends year to year enormous quantities of money in the acquisition of this type of technologies, without be glimpsed to short or medium time limit the creation of a Relational database Engine done in our country, functional complete, that contribute knowledge to our universities, that backup the administration of information of our organizations and that help to avoid the leak of currencies in this matter. Unfortunately, with exception of the countries highly developed, the other countries have only been pleased with the paper of consumers of the database technology. We have implemented an engine in language C + + that will enable ourselves:

- To arrange of a manager of data bases of high quality that permit to the participating institutions to carry out investigation and development that itself would not be able to do with commercial managers.
- To Form human resources that will require the country in the data bases technology, warehouses of data and search of information for intelligence of business.
- Elevation of the academic level in the data bases area, upon putting available to all the universities the manager in free form.

- Creation of new technologies for storage, query, indexing, etc.

7. REFERENCES

[1] Ullman J., Widom Jennifer, "To first course in database System", Prentice Hall, 1997

[2] Kimelman Mihail Leonidovich, "Research and Development of to Language Subsystem SQL", Russian Academy of Sciences, Institute of System Programming, Moscow 1996,; http: //www.ispras.ru/ ßkml gss/

[3] C. J. Date, they Go Introduction to Database Systems, Volume I, Fifth Edition, Addison Wesley Publishing Company, 1990

[4] A. Silberschatz, H. F. Korth, S. Sudarshan, Database System Concepts, 3 Edition, Mc Graw Hill, 1998