International Journal of Latest Research in Science and Technology **ISSN (Online):2278-5299**
Volume 4, Issue 6: Page No.1-6, NovemberDecember 2015
https://www.mnkpublication.com/journal/ijlrst/index.php

DYNAMIC SQL CODEBUILDER

[1]Tonáhtiu Arturo Ramírez Romero, [2]Gumersindo David Fariña López, [3]Francisco Ramírez Torres.
[1]Instituto Politécnico Nacional, D.F. México
[2] Centro de Estudios Científicos y Tecnológicos No.7, IPN, D.F. México
[3]Unidad profesional interdisciplinaria de Ciencias Sociales y Administrativas, IPN, D.F. México
[1]Corresponding author: tonahtiu@yahoo.com

Abstract- In recent years the use of databases in computer systems been on the rise, thus the trend has been to use the Structured Query Language (SQL), for accessing and management. However in computer sciences an expert has to develop these codes to address these questions, so here generator simple SQL code is presented, but it can make way for some time users or programmers themselves use them to generate the desired code faster or it is used as a tool to support other programs.

Keywords: SQL Generator, Dynamic Code

I. INTRODUCTION

In recent decades, the computer community has worked intensively on systems using databases, this haveraised manysolutionsthathave evolved in different ways. In this paper we will focus in relational databases and his access.

The relational databases such as take boom since the publication of the paper "A Relational Model of Data for Large Shared Data Banks." of Mr. Edgar F. Codd's [1]. In this paper Codd´s provide substantial grounds for dealing with data semantics, consistency, and redundancy problems. The Codd’s paper introduces the concept of normalized relations, In other words, groups that are not repeated.

Later in 1976 Peter Chen presented the Entity – Relationship model [2]Codd´s work as the work of Chen, initiated technological developments around relational database.

Discuss relational database is talk of several concepts, which are mentioned here briefly, then proceed to the development of the proposal.

SQL stands for Structured Query Language, The scope of SQL include access and manipulate databases;SQL is an ANSI (American National Standards Institute) standard, is standardized language by the ISO(International Organization for Standardization)[3]. SQL can execute queries against a database, retrieve data from a database, insert, delete, update records in a database and create new tables,views and databases, create stored procedures, set permissions on tables, procedures, and views described as data access control[4]

There are different releases likeSQL 1999(SQL3), SQL 2003, SQL 2008 and SQL 2011.

SQL 2011 or ISO/IEC 9075:2011 (under the title "Information technology – Database languages – SQL") SQL became a standard in the seventh revision of the ISO -1987 and ANSI - 1986 [8] standard for the SQL database query language. It was formally adopted in December 2011 [5]**.**

Although SQL is often described as, and to a great extent is, a declarative language (4GL), it also includes procedural elements.

The database is created as a set of tables, relationships are represented by values in tables and data is retrieved by specifying a result table that can be derived from one or more base tables. SQL statements are executed by a Database manager[6]

Nowadays the database system tends to be the most important development in the field of information systems.

The Database Management System (DBMS) is a software system that enables users to define, create, maintain, and control access to the database.

Usually, provides the following facilities:

- Data Definition Languages (DDL), allows users to specific: The data types structures and constrains on the data in the database.

- Data Manipulation Language (DML). It Allows users to insert, delete, update and retrieve data from the database. A query language is a general inquiry facility to the data the provision on this language alleviates the problems. The most common query language is the Structured Query Language (SQL). Which is now both the formal and facto standard language for relational DBMS. Data manipulation statements of the SQL language: Select, insert, delete and update.

- A security, an integrity system, a concurrency control system, a recovery control system, a user accessible catalog. [7]

Publication History
Manuscript Received : 4 November 2015
Manuscript Accepted : 12 November 2015
Revision Received : 25 December 2015
Manuscript Published : 31 December 2015

Relational algebra.

The relational algebra arises precisely from the calculus and algebra, is a theoretical language with operations that work on one or many relations to retrieve new information.

This field of mathematics has been the basis for creating other high-level languages, for example in the SQL the Data Manipulation Languages (DMLs) for relational databases.

A language based on relational calculus that can generate any relation is said to be relationally complete.

In the relational model, all is logically structured within relations (tables). Each relation has a name of an entity and is constituted of attributes (columns) of data.

Because relational databases are based on relational algebra and since here is the access to these databases via SQL, then briefly describes the main terms used:

Relation. It is an entity called table.
Tuple.It´s a record or row.
Compatible union.It refers to the compatibility between two relations, if they have the same degree.
Attributes. Columns
Degree.Number of attributes.
Type. Row

Before continuing a framework where you can use the SQL code generator will be given, the idea begin from the need to generate SQL code for a rules-based system that aims to filter data input using rules knowledge base stored in a database, such filtering rules can be queries to other databases schemes, however the rules have their own grammar and are not SQL code, so a SQL translator is needed, this paper showthe SQL translator with the respective progress as of today,the idea proposed here is simple but works well.

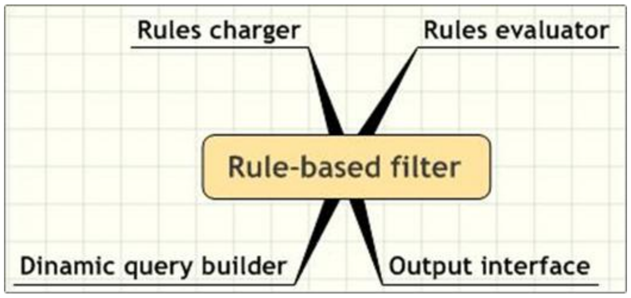

Figure 1. General diagram of the filter system [9]

Lock the figure 1. We show the relation with other elements into a system, the SQL builder o generator is a part of this.

Also SQL generator can be used with a GUI to build the query simply intuitive for users who know the procedures in the organization and are not necessarily SQL experts or system architecture[9].Although Such as this line of research will be discussed in another work.

After a brief introduction we proceed to show the development below.

II. DEVELOPMENT

Based on the aforementioned idea that part of the formality of SQL is the relational algebra, we proposed that it is possible to develop a computer program capable of generate SQL code, when the structure of the database is provided to the system. Then apply the rules of relational algebra in building code

In relational algebra there are five operations: Selection, projection, Cartesian product, Union and Set difference. In addition there are also the join, intersection and division operations,but they can be produced based on the five basic operations.

For the product of this research, we focus here on SELECT or projection from relational algebra. To continue the development this command is defined the basic structure in SQL.

Table 1. Basic structure of SQL

```
SELECT [DISTINCT | ALL]{*|[columnExpression[AS newName]][,…]}
FROM TableName [alias][,…]
[WHERE condition]
[GROUP BY columnList][HAVING condition]
[ORDER BY columnList]
```

Where
Uppercase are reserved words in SQL
| Symbol of disjunctive
* Symbol to specify all
[] Symbol to separate instructions, but only is to show, it isn´t part of the code.

Now, we proceed to explain the reserved keywords in uppercase

SELECT is the keyword that includes the column or columns single o compounds to include in the final result usually linked to a table or relation. An asterisk "*" can be used to specify that the query should return all columns of the queried tables. SELECT can contain optional keywords and clauses.

DISTINCT or *. This keyword is optative, and is used to retrieve only different values from the column.

FROM clause, this indicates the table(s) to retrieve data from, is like a data source.It’s possible include a schema of the database, first database schema – point – table.

WHERE clause is used to apply a conditional comparison between two variables or a variable and constant. The WHERE clause show only rows that meet the condition is true. In the table 2, we show some symbols that can apply into clause WHERE.

Table 2. Symbols for conditional

Symbol	Description
=	Equal to
<>, !=	Notequal to
>	Greater than to
<	Less tan to
>=	Greater than or equal to
<=	Less than or equal to
LIKE	Match a character pattern
IN	It is to compare a value with one or more values
NOT IN	It is to eliminate where a values is in set of values after the symbol.
BETWEEN	For delimit rows into a range

GROUP BY,It lets you group the same records for example to count, is often used in conjunction with SQL aggregation functions or to eliminate duplicate rows from a result set. The main aggregate functions are:

- COUNT
- SUM
- AVG
- MIN
- MAX

First need to group tuples and then use the aggregate functions to produce the required relation.

HAVING clause includes a predicate used to filter rows resulting from the GROUP BY clause. it acts on the results of the GROUP BY clause, aggregation functions can be used in the HAVING clause predicate.

ORDER BY clause identifies which column(s) to use to sort the resulting data it can use asc (ascendant) or desc (descendent) to determine the direction.

This code structure has a great combination that can display data, but in this work will only showsome examples, for future work we will show a generator throughout the range of valid SQL, given certain standard ISO and ANSI.

It should be noted that not all syntactically correct SQL code according to some standard ISO or ANSI works in all relational databases managers, in later work evidence that code do not work ever.

It is important to explain that the SQL code generator presented here, not doing chores DBMS, rather generated codeleaving the database management system (DBMS) to carry out planning, optimizing, and performing the operations necessary to produce that result as it selected.

A simple but functional method is proposed to generate SQL code.In this work PHP[10, 11] language is used to develop SQL Builder,

Begin with particular examples then give way to general case. Using the structure showed in Table 1, we propose the next code.

Number	Code in PHP
1	$SQL_query = "SELECT $fields FROM $table WHERE $condition";

Note, This PHP´s code fragment represents the SQL code, storage in the variable $SQL_query that can be sent to Database Manager, building the SQL code is important for it is constructed as follows: the variable $fields contain the selection of field or fields,the variable $table contains the tables or tables where the information is extracted, and finally the variable *$condition*, detailing the conditions applying in that include necessary to link the tables, if more than one, to these conditions is added the characteristics of the query.

However the rest of the string appears as fixed because what is sought is to create a string in SQL code that meets what we need.

To select tables is different because its relations previously should be fed into systems and the relations between them indicating which are the primary keys and foreign keys [12].

For this there are several ways of doing:

The first is perhaps the simplest for small systems, is to translate the code structure in a simple if-then code, this method though limited, is functional for many cases,

The second option is to create a scheme that indicates the content of table databases and index.

The third is to take the structure of the database manager, because if the database is well defined and documented we can retrieve from if self.

The first method includes the table's names and its correspondent attributes from the database structure, as part of the SQL generator code, in this case developed in PHP.

The second option is create a table's set of tables, rows, data types, index, and access control and maybe constrains.

The third option is retrieve the information directly from the DBMS, for example Oracle [13]. have the table *all_tables* and other tools, MySQL [14] have the schema called mysql, and have commands like show tables, show processlist, show status, and many others, these tools can be used by the SQL generator. In general the Database Manager has tools for retrieve information about the databases inside.

Management tables

Now we explain the first solution mentioned previously. Consider the next PHP code:

```
if($table == 'Employees')
 $compatible_field = 'empl_num_emp';

if($table == 'Control')
 $compatible _field= 'ctl_num_emp';
```

In this example we use two data source storing information about Employees and Control of them in the relations (Tables) *E* and *C*, respectively, like show figure 2.

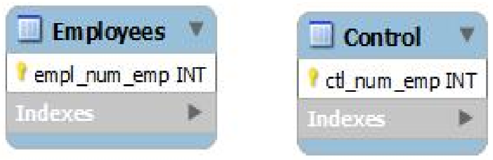

Figure 2. Tables of Employees and Control

If we want to retrieve attributes (columns) using Code PHP 1, *$compatible_field* can take two values: *empl_num_emp*&*ctl_num_emp* ,the value taken depends on the value of the table.

The same can be done with code from different database schema, Consider the next PHP code:

```
if($table == 'DB1.Employees')
 $compatible_field = 'empl_num_emp’;

if($table == 'DB2.Control')
 $compatible _field= 'ctl_num_emp';
```

The same applies to fields that serve as both primary and secondary keys to build the link between table conditions, thus forming code that works well.

Suppose we want to perform a natural join in relational algebra [15], continue with the example of Employees and Control, for the purpose of reducing table names take the first letter *E* and *C*to represent these relations in relational algebra.

$$\varepsilon : W = E \bowtie_{E.emplnumemp=C.ctrnumemp}$$

This example is formal, into SQL we can represent it using reserved word WHERE. Check the next example:

Management WHERE
WHERE empl_num_emp = ctr_num_emp

In this case we use two tables, Employees and Control, but the DBMS automatically detect the correspondence between the attributes and tables.

Other cases more simples can occur for example

WHERE field1 > 50

In this example is simple only check that a field is greater than 50, but other conditional symbols must be supported for example the symbols from the table 2.

After presenting two examples and the representation of an example in relational algebra, we proceed to show the treatment of the SQL WHERE component, we show a structure in figure 3.

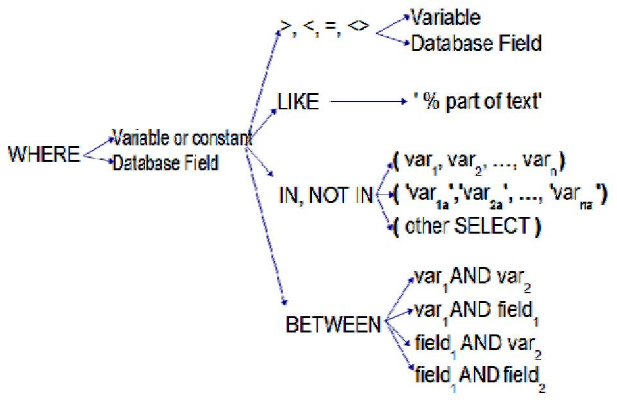

Figure 3. SQL WHERE

As you can see inthe figure 3, WHERE havemany options, these option can perform a combinatorial and thus *n*permutations

We show the most substantial of all permutations, not all code is placed, three sections to generate this code SQL in language PHP, which is then integrated into the chain of full SQL is as follows:

The most representative cases of all permutations are shown, as the language in which it was mentioned develops is PHP. Not all code is placed, but the overall aim is to show the method. The SQL's chain is built in four sections, after thus is integrated into a single *$condition* chain as follows:

First section of WHERE.
This is the more simple case, only compare two statements, where both are numeric as follows:

$conditional = “WHERE $field1 > $value”;
or
$conditional = “WHERE $value < $field1”;

Other case is where values are characters or character´s chain:

$conditional = “WHERE $field1 > '$value'”;
or
$conditional = “WHERE '$value' < $field1”;

Other case is where value are characters or character´s chain:

Second section of WHERE

In this section we describe the option like, that use exclusively to character´s chain, it´s used with the symbol % that can occur once or more times within the chain delimited by apostrophes and tells the database manager that this symbol may be replaced by a letter or a string, then indicates that the like string comparison is relative and not an exact comparison lexically, see the next PHP code:

$value = “My % string”;

$conditional = “WHERE $field1 like “ . $value;

Notes that this code string is divided into two sections, unified by a point the reason is because as the second variable includes apostrophes, in some versions of interpreter it's fails, but giving the treatment presented here the problem is solved.

Third section of WHERE

For the IN and NOT IN options from figure 3, we propose the next PHP code.

For numeric values:
$values = “$val1, $val2, $val3, …, $valn”;
$conditional = “WHERE $var1 IN($values) ”;
$conditional = “WHERE $var1 NOT IN ($values) ”;

For characters or strings:
$values = “’$val1’, ‘$val2,’‘$val3’, …, ‘$valn’”;
$conditional = “WHERE $var1 IN ($values) ”;
$conditional = “WHERE $var1 NOT IN ($values) ”;

Fourth section of WHERE

For the BETWEEN use, the following code is used, for the purpose of reducing the size of the *$condition* chain, abbreviated *data_field* as a *dfield*, in figure 3, one can observe four cases, vars$_n$ were used as dates; variables, data field before the WHERE are called $element as follows:

Case 1
$condition = “WHERE $element BETWEEN 'date1' and date2”;

Case 2
$condition = “WHERE $element BETWEEN 'date1' AND dfield”;

Case 3
$condition = “WHERE $element BETWEEN dfield and date2”

Case 4
$condition = “WHERE $element BETWEEN dfield1 AND dfield2”;

Care should be taken in consideration that the dates have to be between apostrophes as they are considered text.

Four possible types of conditional within the command are mentioned, but in SQL are cumulative, to resolve this chain SQL would be built like this:

$conditional = “WHERE comparison_1... “;
$conditional .= “ AND comparison_2 … “;
$conditional .= “ AND comparison_n … “;

Finally it's possible build the $SQL_code, taking the structure that fall within the context of Table 1. We present some structures that to integrate the parts, as follows:

1. $SQL_script = $select . “ “ . $source;
2. $SQL_script = $select . “ “ . $source . “ “ . $conditional;
3. $SQL_script = $select . “ “ . $source . “ “ . $conditional . “ “ . $order;

The case 1 is only retrieve data from a source, the case 2 aggregate a conditional and the case 3 use order.

As a last step is run from the PHP application code built chain as follows:

```
$result = mysql_query($SQL_script);
while($regs = mysql_fetch_array($result, MYSQL_ASSOC))
{
$data1 = $regs["field1"];
$data2 = $regs[“field2”];
…
}
```

The values returned by the last code and can be used for what is needed, and therefore terminated the proposal.

III. CONCLUSIONS

The proposal presented here is the simplest and although functional option, option two and three are larger and work in them, when this list some of them, will be presented. Although many cases the option presented here is useful and is easy to program and implement in different computer applications.

IV. ACKNOWLEDGMENT

We like to express sincere appreciation and deep gratitude to all participants in this work.

We thank the InstitutoPolitécnico Nacional for computational infrastructure to facilitate the development of this proposal, which has applied to a process on production of information.

V. REFERENCES

[1] Codd, Edgar F (June 1970). "A Relational Model of Data for Large Shared Data Banks". *Communications of the ACM* (Association for Computing Machinery) **13** (6): 377–87.doi:10.1145/362384.362685. Retrieved: 2015-20-10. https://www.seas.upenn.edu/~zives/03f/cis550/codd.pdf Retrieved 2015-20-10.

[2] Chen Peter, “The Entity – Relationship model – Toward a unified view of data”, 1976

[3] ISO, 2015, http://www.iso.org/iso/home.html Retrieved: October 25, 2015.

[4] W3, 2015, http://www.w3schools.com/sql/sql_intro.asp. Retrieved: October 25, 2015.

[5] SQL, 2011,https://en.wikipedia.org/wiki/SQL , Retrieved 2015-25-10.

[6] IBM, 2015, http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.1.0/com.ibm.db2.udb.admin.doc/doc/c0004100.htm Retrieved October 25, 2015.

[7] Connoly Thomas, Begg Carolyn, “Database systems - A practical Approach to Design, Implementation, and Management”, Addison Wesley – Pearson, 5th ed., 2010, USA.

[8] ANSI, http://www.ansi.org/Retrieved: October 27, 2015.

[9] Ramírez Romero T.A.,Patiño Ortiz. M., Velazquez Benina, Filtering the input data by production rules on open Database, International Journal of Latest Research in Sciences and Technology, Vol.2, Issue 5, pp 1-8, Sep - oct 2013. India. http://www.mnkjournals.com/ijlrst_files/Download/vol%202%20Issue%205/1-28-23102013-Tonahtiu%20Ram%C3%ADrez.pdf

[10] **PHP, https://secure.php.net/**Retrieved: October 28, 2015.

[11] **Helma**Spona, *Programación de bases de datos con MySQL y PHP*, MarcomboPress, 2010, Spain.

[12] **Coronel**, Morris, Rob, "Bases de datos – Diseño, implementación y administración", Novena edición, PP.155, CengageLearning, 2011. México.

[13] Oracle, https://www.oracle.comRetrieved: October 28, 2015.

[14] MySQL, http://www.mysql.comRetrieved: October 29, 2015.

[15] **Thomas** Jörg, Stefan Dessloch, *Near Real – Time Data Warehousing State-of-the-Art ETL Tools*. TechnischeUniversitätKaiserslauten, 2010. Germany.