

# A Review on Translation of SQL Queries to Spreadsheet Formulae

Priyanka U. Wani, Professor K.V.Metre

Priyanka U. Wani, Dept of Computer Engineering, MET's Institute of Engineering Nashik, Maharashtra, India

Professor K. V. Metre, Dept of Computer Engineering, MET's Institute of Engineering Nashik, Maharashtra, India

\*\*\*

**Abstract** - The main topic of this article is to convert the SQL queries to Formulas based Spreadsheet. Many people find difficulties in working with databases queries so they completely want to migrate from databases to another application. Hence here it is the solution to combine the database concept with another application to make it simple and easy. The concept called spreadsheet is combined with the databases which forms a method which is called as a SQL Query Translator. Spreadsheets are the most popular application for data analysis and manipulations. Thus SQL Query Translator is an automated tool which translates the Database SQL query to Formula based Spreadsheet. Also it parses the statements into the parse tree and generates the syntax tree providing validation to the statements at an early stage.

**Key Words:** Databases, Relational Algebra, SQL Queries, Spreadsheet, Parser.

## 1.INTRODUCTION

Spreadsheets are the most commonly used application or tool. Spreadsheet is a computational tool for data management and analysis in business modeling applications. It is very easy to understand and to work on by any individual user or any organization. The most commonly used spreadsheet is the Microsoft Excel, used by any user. Spreadsheets offer many functions in the field of engineering, mathematics, financial, statistics so on. It also offers services like pivot, aggregation, lookups, graphs, customizable menus and so on. They are highly portable and act as a virtual machine where spreadsheet applications offered by vendors can run.

The notable thing is that spreadsheet language of formulas of Excel has become a de facto standard. It is design in a large number of spreadsheet systems, available for all major operating systems and hardware platforms, starting from handhelds and ending in the cloud, from proprietary to close source. Computer applications in the form of formula-only spreadsheets are therefore highly manageable, possibly to the extent comparable with Java bytecode. Spreadsheet systems can be observe as virtual machines, offered by various vendors, on which spreadsheet applications can be run. It is therefore extremely surprising that those machines are substantially programmed manually, with no compilers producing spreadsheet code from higher-level languages.

Still spreadsheets may have certain disadvantages. It lacks the analysis required enough to be work on. It may have scalability and share ability problems. Spreadsheet is an analytical tool having data and formula bounded together at one place giving rapid prototype.

## 2. LITERATURE SURVEY

In 2015, Jery Tyszkiewicz [2] stated that spreadsheets can act as a relational database engine, just by using spreadsheet formulas and no any programming language to be built in. The implementation is based on Microsoft Excel thus contributing on other spreadsheet models too. When the user enters, alters or deletes data in the spreadsheet worksheet, the formulas in query worksheet automatically computes the actual results of the queries.

In 2010, Bin Liu et al. [3] presented direct data manipulation query interface. Main objective is to create a spreadsheet like interface that directly query and access relational databases through direct manipulation. Nontechnical users find it challenging defining queries against structured databases. So a direct manipulation interface provides some datasets to the user on hand to be analysed or manipulated.

Andrew Witkowski et al. [4], [5] proposed spreadsheet like computation in RDBMS through extensions to SQL, allowing OLAP tools to handle the user interface. Spreadsheet is a successful analytical tool for business data but it has scalability problem. SQL doesn't support for n-dimensional array based computations which are frequent in OLAP scale computation while spreadsheets and specialized MOLAP engines are good for constructing formulas for mathematical modelling.

In 2003, Andrew Witkowski et al. [6] Proposed automatically translation of Excel computation into SQL. Main goal was to translate Excel computation to SQL and user extensions to Excel formulas and menus to perform relational operations on RDBMS.

A fundamental challenge in spreadsheet is lack of ill-defined schema. So L. V. S. Lakshmanan [7] Proposed a framework where user specify the layout of data in a spreadsheet. Introduced the concept abstract database machine (ADM) which user layout specification viewed as a physical schema.

of spreadsheet which provides relational view of data in spreadsheet applications.

Error rate is much high in the spreadsheets. Robin Abraham [8] introduced a type system and type inference algorithm for spreadsheets and demonstrated how these algorithms will identify the programming errors in spreadsheets.

Simon Peyton Jones et al. [9] contributed for a possible extension to Excel that supports user defined functions. Michael Kassoff et al. [10] presented a PrediCalc, a spreadsheet system which allows general logical constraints, preserving benefits of automatic calculation of values.

### 3. SYSTEM ARCHITECTURE

#### 3.1 Database Architecture

The Database Architecture of relational database has following procedure:

- The tool Query-Translator implements the SQL or relational query in the .xlsx file form.
- The result obtained contains a single worksheet including number of columns for data-tables and those columns performing the computation.
- There are two rows of formulas to be implemented in which the second row contains all the detail information of the formulas that the user enters.

When the user enters data into tables, the tool automatically computes the query and output it in the form of columns. The list of operators for implementation is: Sorting, Duplicate removal, selection, projection, union, difference, Cartesian product, grouping with aggregation.

#### 3.2 System Architecture

In the Fig. 1, the actors involved are the end user, the tool SQL Query Parser and the application Spreadsheet. Following are the steps of the system:

1. User needs to first define the database schema.
2. User has to give the database SQL query as an input to the system.
3. The system analyses the query, validates whether the entered query is correct or not. If correct then generates parse tree.
4. Then it converts SQL query to Relational Algebra expression applying the algorithm explained above and displays result.
5. It again converts this Relational Algebra Expression into formula based Spreadsheet and displays result.
6. Finally the system generates and displays the output.

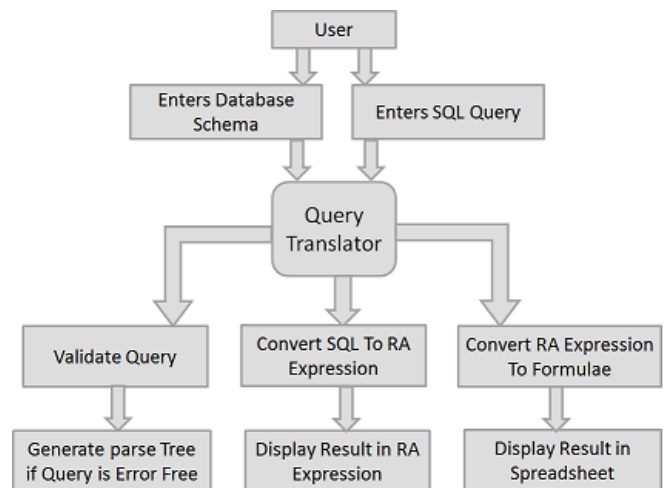


Fig -1: System Architecture

### 3. CONCLUSIONS

The proposed system is to build an automated tool for converting the SQL Queries to formula based Spreadsheets. Thus, we have shown the power of the spreadsheet paradigm, which subsumes the paradigm of relational databases. In this system Query Translator helps users to reduce human generating error while writing SQL queries. The drawbacks of SQL and spreadsheets are also covered in the system which improves the performance of the system. The main goal of the system is to translate the queries into spreadsheets. The proposed tool is useful for the professional users as well as nonprofessional users who are learning and using new programming languages.

### 4. REFERENCES

- [1] JacekSroka, Adrian Panasiuk, Krzysztof Stencel, and Jerzy Tyszkiewicz, "Translating Relational Queries into Spreadsheets", IEEE Trans. On Knowledge and Data Engineering, Vol. 27, No. 8, August 2015
- [2] J. Tyszkiewicz, "Spreadsheet as a relational database engine," in Proc. ACM SIGMOD Int. Conf. Manag. Data, 2010, pp. 195–206.
- [3] B. Liu and H. V. Jagadish, "A spreadsheet algebra for a direct data manipulation query interface," in Proc. IEEE Int. Conf. Data Eng., 2009, pp. 417–428.
- [4] A. Witkowski, S. Bellamkonda, T. Bozkaya, G. Dorman, N. Folkert, A. Gupta, L. Shen, and S. Subramanian, "Spreadsheets in RDBMS for OLAP," in Proc. ACM SIGMOD Int. Conf. Manag. Data, 2003, pp. 52–63.
- [5] A. Witkowski, S. Bellamkonda, T.Bozkaya, N. Folkert, A. Gupta, L. Sheng, and S. Subramanian, "Business modeling using SQL spreadsheets," in Proc. 29th Int. Conf. Very Large Data Bases, 2003, pp. 1117–1120.

[6] A. Witkowski, S. Bellamkonda, T. Bozkaya, A. Naimat, L. Sheng, S. Subramanian, and A. Waingold, "Query by excel," in Proc. 31st Int. Conf. Very Large Data Bases, 2005, pp. 1204–1215.

[7] L. V. S. Lakshmanan, S. N. Subramanian, N. Goyal, and R. Krishnamurthy, "On query spreadsheets," in Proc. Int. Conf. Data Eng., 1998, pp. 134–141.

[8] R. Abraham and M. Erwig, "Type inference for spreadsheets," in Proc. 8th ACM SIGPLAN Symp. Principles Prac. Declarative Program. 2006, pp. 73–84.

[9] S. P. Jones, A. Blackwell, and M. Burnett, "A user-centred approach to functions in Excel," in Proc. 8th ACM SIGPLAN Int. Conf. Functional Program. 2003, pp. 165–176.