



ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 2, Issue 11 , November 2015

SQL Parser

Omkar Pukale, Anand Rananaware, Sushant Rasalkar, Sanket Shah, S.B.Tatale

BE Students, Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India.

BE Students, Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India.

BE Students, Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India.

BE Students, Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India.

Assistant Professor, Department of Computer Engineering, Vishwakarma Institute Information Technology, Pune, Maharashtra, India.

ABSTRACT: Business Intelligence, Reporting and analysis of data are major requirement of the business. It would be better if all the information is available in one place. Spreadsheet is very popular in business community to view data / information. It also supports all kinds of reporting, data analysis so business people can easily get the information from spreadsheet. Therefore there is a huge requirement in market to translate the business logic written in SQL to be converted into spreadsheets for ease in understanding.

KEYWORDS: SQL, Parse Engine, Spreadsheets, Database, Business Intelligence, Business Rules.

I. INTRODUCTION

There is a need to transit from traditional legacy system to newer technologies. Information about legacy systems is not completely available. In order to develop software having similar functionalities one needs to understand the working of legacy system. Moving from a legacy to new system means rewriting the existing code and rewriting the mapping of fields from input to Output which is tedious and time consuming work for system developers. Many of the legacy software are SQL based. Parser engine would help the developers to map the SQL query to newer technologies and create an understandable document.

Storing and managing of sets of data is done by spreadsheets and databases. Spreadsheet or a database store data in the form of set of data values. The difference between spreadsheets and databases lies in how they store and manipulate the data. A spreadsheet stores data values in cells, with multiple cells represented in a system of rows and columns. Cells communicate to other cells, and the spreadsheet has cells that carry out different processing functions on other cell values. A database generally stores data in form of tables. Each table consists of a name and one or more rows and columns. A row in a table is called a Record. Databases can establish relationships between records in different tables. In business perspective lot of users come across complicated database; so they simply prefer word processor or spreadsheets which are having better GUIs and simpler processing functions and for performing database operations in spreadsheet user has to learn many new commands. The idea is to build software that has the data from database and convert it to the understandable document format.

Spreadsheets and databases use different technologies. The most widely used spreadsheet program is Microsoft® Excel, which is part of Microsoft® Office. Other spreadsheet programs form part of Open office and Google™ Docs. Database technologies include Microsoft® Access, Oracle, MySQL and SQL Server, among many others. Some databases are accessed over networks which run on the servers, including the Internet. In most cases, a database will have a software application used for providing user interface built on top of it, providing user access to the data. Databases are built and managed by Web developers and software programmers, often using SQL. The disadvantages of databases include requiring the user to learn a new system, and a greater investment in training and software.

Query Converter tool main aim is convert SQL language into the spread-sheet macro.

II. EXISTING SYSTEM

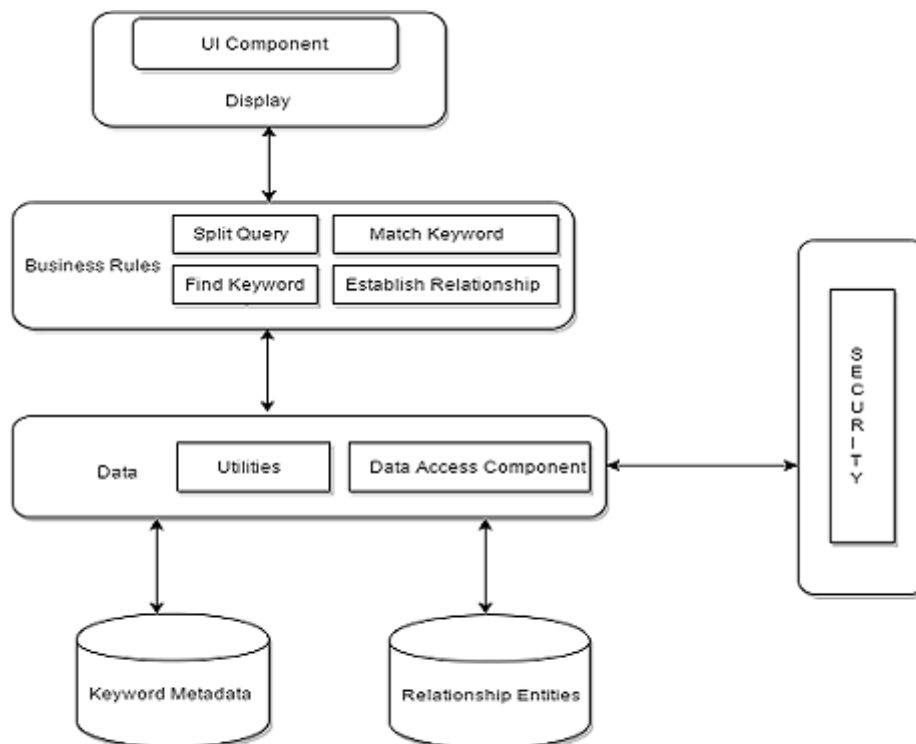
There is no existing system based on translating SQL Queries into spreadsheet database. It is required for the developers working on migration of legacy system to newer ones. This work is currently done manually. The developers which are working on converting legacy script to newer technology required to do this work manually which is inefficient and mostly time consuming. It also increases the probability of error because of increased human interaction. So, there is need to need to automate this process.

To overcome this problem SQL Parser is the option. It takes the input from the user in the form .sql file format. Then it Splits the query and Processes it to form the output. The output of the system will be shown in spreadsheet (Microsoft® Excel). It is called as Mapping Document.

III. PROPOSED SYSTEM

Following section contains architecture of the proposed system for the SQL parser. It contains different components working together to form a system required to convert the traditional SQL legacy scripts into more human readable format. The output of which is shown in spreadsheet format. This will be helpful to the developers for further use.

The proposed system for SQL Parser is:



SYSTEM ARCHITECTURE DIAGRAM



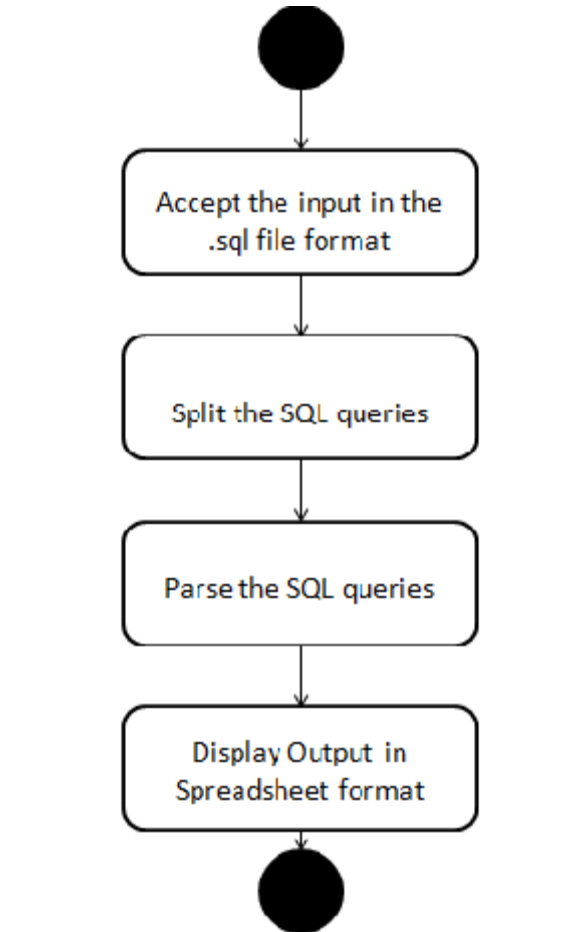
ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 2, Issue 11 , November 2015

- **Display:** It consists of the components dealing with external interface with the user. Main component of the same is UI.
 1. **User Interface (UI):** UI plays an important role in any type of system. To interact efficiently with the SQL Parser its UI consists of file selection option. Also the format in which to output is required can be selected. There are usual buttons to close, minimize the software UI.
- **Business rules:** This part is the one where core functioning of the software is done. It consists of splitting the query, finding keywords, matching of keywords and establishes relationship.
 1. **Split Query:** This component plays a part in arranging the query according to the software perspective. Splitting of multiple queries when present is done by the same. It also preserves the order of the queries as SQL queries are processed sequentially. Query separation can be based on occurrence of semi colon';'.
 2. **Find Keyword and Match Keyword:** This part deals with identification of the keywords in the query. Excluding unnecessary words and comments is an important function of the same. It refers to the keyword metadata in order to carry out its functionality. In order to reach to keyword metadata it has to interact with data access component.
 3. **Establish Relationship:** To get the desired output, the most important part of the processing is establishing relationships. Its correct processing leads to proper understanding of the query. Relationship among the keywords is established. This component interacts with data access layer in order to reach to relationship entity data. Without this connectivity functioning and establishing relations is not possible.
- **Data:** It provides interface to all data required for the system to work. It is connected to database components. It is provided with security features.
 1. **Data access component:** It acts as a bridge between the components in business rules section and the default data repository of the system. It provides authentic and constant connection between components responsible for matching and finding of keywords and the keyword metadata file. It also connects component responsible for establishing relationships and the relationship entity file. Interaction with security layer is also present as authentication and authorization is carried out by security later.
- **Security:** It is one of the most important parts of any software being developed. It interacts with the data access layer and controls the data access authorization. No component can access the data beyond its scope. Any malfunctioning component if found is immediately stopped and reported. Also access to the system by unauthorized users is checked by the same.
- **Database:** Database of the system consists of two different components.
 1. **Keyword metadata:** It is an imported part of the default data repository of the system. It consists of list of all the keywords in SQL. Newly learnt keywords can also be added to the same without making any changes in the source code of the program. Adding of keywords can be done manually. It can also be done automatically if machine learning is used. Data in this file is used for finding and matching keywords.
 2. **Relationship Entities:** This is the second part of the data repository. It consists of all the probable relations among the keywords. It consists of execution order of the keywords. Also the dependencies among the same are present in this file. Data in this file can be updated without making changes to the actual source code of the system. This file is accessed by the relationship establishing component in the business rules section of the system. Access is facilitated by the data access component.

- Following is the **Flow** of the System:



IV.CONCLUSION

SQL Parser is a tool to analyze the SQL queries using various mathematical and functional aspects of language. It generates a final output as an understandable document in .csv or .xlsx format. Hence, this tool will be useful for programmers and business analyst in order to understand the functionalities of the queries and SQL based systems in general.

REFERENCES

- [1]. Jacek Sroka, Adrian Panasiuk, Krzysztof Stencel, and Jerzy Tyszkiewicz "Translating Relational Queries into Spreadsheets", IEEE Transactions on knowledge and data engineering [Vol. 27, NO.8] August 2015.
- [2]. Sagar Sunkle, Martin Kuhlemann, Norbert Siegmund, Marko Rosenmüller, Gunter Saake "Generating Highly Customizable SQL Parsers" School of Computer Science University of Magdeburg 39106 Magdeburg, Germany.
- [3]. Lukas Diekmann and Laurence Tratt, "Parsing Composed Grammars with Language Boxes", Software Development Team, King's College, London.
- [4]. SQL Server execution plan by G. Fritchey.
- [5]. http://my.safaribooksonline.com/book/databases/sql/9781906434939/chapter-1-execution-plan-basics/chap01_sub1_xhtml
- [6]. [http://msdn.microsoft.com/en-us/library/ms191227\(v=sql.105\).aspx](http://msdn.microsoft.com/en-us/library/ms191227(v=sql.105).aspx)
- [7]. <http://www.developer.com/db/understanding-a-sql-server-query-execution-plan.html>