ISSN: 2350-0328

International Journal of Advanced Research in Science, Engineering and Technology

Vol. 3, Issue 3 , March 2016

Translating SQL Queries into Mapping Document

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

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

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 3 , March 2016

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. LITERATURE SURVEY

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.
Description: Spreadsheets are among the most commonly used applications for data management and analysis. They combine data processing with very diverse supplementary features: statistics, visualization, reporting, linear programming solvers, Web queries periodically downloading data from external sources, etc. However, the spreadsheet paradigm of computation still lacks sufficient analysis. In this article, we demonstrate that a spreadsheet can implement all data transformations definable in SQL, merely by utilizing spreadsheet formulas. We provide a query compiler, which translates any given SQL query into a worksheet of the same semantics, including NULL values. Thereby, database operations become available to the users who do not want to migrate to a database. They can define their queries using a high-level language and then get their execution plans in a plain vanilla spreadsheet. The functions available in spreadsheets impose limitations on the algorithms one can implement. This paper offers O (n log2 n) sorting spreadsheet, using a non-constant number of rows, and, surprisingly, Depth-First-Search and Breadth-First-Search on graphs.

2) Sagar Sunkle, Martin Kuhlemann, Norbert Siegmund, Marko Rosenmuller, Gunter Saake, **Generating Highly Customizable SQL Parsers**, School of Computer Science University of Magdeburg, 39106 Magdeburg, Germany.
Description: Database technology and the Structured Query Language (SQL) have grown enormously in recent years. Applications from different domains have different requirements for using database technology and SQL. The major problem of current standards of SQL is complexity and unmanageability. This paper present an approach based on software product line engineering which can be used to create customizable SQL parsers and consequently different SQL dialects. It gives an overview of how SQL can be decomposed in terms of features and how different features can be composed to create tailor-made parsers for SQL.

3) Lukas Diekmann and Laurence Tratt, **Parsing Composed Grammars with Language Boxes**, Software Development Team, King's College, London.
Description: Parsing uncovers if, and how, an input stream conforms to a grammar. Language composition requires combining grammars together, yet all traditional parsing techniques have limitations when parsing composed grammars.Thos paper augment an incremental parser with language boxes, which allows user to retain the feel of traditional parsing whilst allowing arbitrary grammar composition.

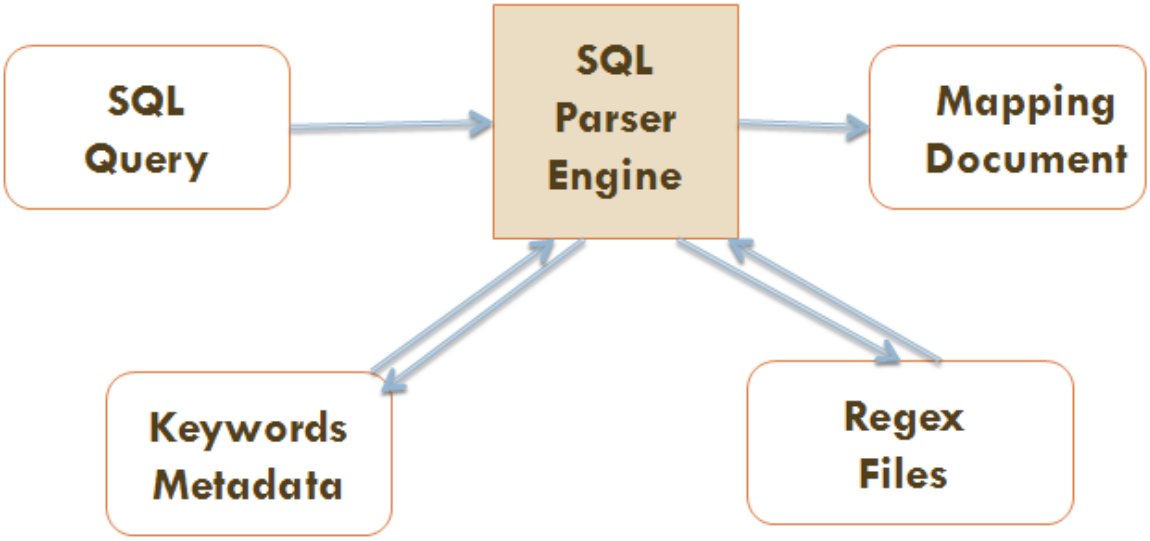
III. EXISTING SYSTEM

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.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 3 , March 2016

IV. IMPLEMENTED SYSTEM ARCHITECTURE



Block Diagram

SQL Query:

SQL (Structured Query Language) is a standard interactive and programming language for getting information from and updating a database.Queries take the form of a command language that lets you select, insert, update, find out the location of data, and so forth.

SQL Parser:

SQL Parser identifies the pattern in the given SQL query. It looks for the transitions on the table which may be complete or partial. It refers to metadata dictionary in order to check the functionality of the keyword found.

Keywords Metadata:

It consists of keyword dictionary. Also a functionality of the keyword is specified for better understanding and avoiding computational complexity. It is accessed by SQL Parser in order to identify the query pattern.

Regex Files:

It consists of regular expressions for each type SQL query. Contents of this entity are derived from SQL grammar. Types of Regex files used are: Select, Update, Insert, Delete, Alter, Drop. The regex files contain every possible case for specific type of query according to 1st keyword in the query. These files are loaded into system as soon as system is initiated.

Mapping Document:

This is the final output generated by the Query Convertor tool. It consists of human understandable form of the complex SQL query. It is of .csv or .xlxs form.

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 3 , March 2016

Graphical User Interface

i. Input Query and Corresponding Output

SELECT (IA.INVOICE_ID - 9999) AS HSA_ID,ABS(SUP.INVOICE_AMOUNT) ABS_INV_AMT INTO TMPTBL_AP FROM AP_INVOICES_ALL IA LEFT OUTER JOIN AP_SUPPLIERS SUP ON IA.VENDOR_ID = CAST(SUP.VENDOR_ID AS DECIMAL(15,0)) LEFT OUTER JOIN AP_TERMS_TL TT ON IA.TERMS_ID = TT.TERM_ID WHERE IA.INVOICE_ID > 1223 AND SUP.INVOICE_NUM != QWER2987 AND TT.TERM ID = 12987;

ISSN: 2350-0328
International Journal of Advanced Research in Science, Engineering and Technology
Vol. 3, Issue 3 , March 2016

	A	B	C	D	E	F
1	***TABLES USED:***					
2	AP_TERMS_TL					
3	AP_INVOICES_ALL					
4	AP_SUPPLIERS					
5						
6	***JOINS USED:***					
7	***TABLE NAME***	***FIELD NAME***	***JOIN TYPE***	***TABLE NAME***	***FIELD NAME***	
8	AP_INVOICES_ALL	IA.VENDOR_ID	LEFT OUTER JOIN	AP_SUPPLIERS	CAST(SUP.VENDOR_ID AS DECIMAL(15,0))	
9	AP_INVOICES_ALL	IA.TERMS_ID	LEFT OUTER JOIN	AP_TERMS_TL	TT.TERM_ID	
10						
11	***COLUMN RULE:***					
12	***INITIAL TABLE NAME***	***INITIAL FIELD NAME***	***TRANSFORMATION RULE***	***FINAL TABLE NAME***	***FINAL FIELD NAME***	
13	AP_INVOICES_ALL	INVOICE_ID	(IA.INVOICE_ID - 9999) AS HSA_ID	TMPTBL_AP	HSA_ID	
14	AP_SUPPLIERS	INVOICE_AMOUNT	ABS(SUP.INVOICE_AMOUNT) ABS_INV_AMT	TMPTBL_AP	ABS_INV_AMT	
15						
16	***TABLE RULE:***					
17	***TABLE(S)***	***RULE***				
18	AP_INVOICES_ALL IA , AP_SUPPLIERS SUP , AP_TERMS_TL TT	WHERE Clause: IA.INVOICE_ID > 1223 AND SUP.INVOICE_NUM != QWER2987 AND TT.TERM ID = 12987				

PARSED QUERIES

Output(Mapping Document)

V. 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 Rosenm¨uller, 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
[7] http://www.developer.com/db/understanding-a-sql-server-query-execution-plan.html