Accessing and Analyzing Spreadsheets using SQL and NLP

Mrs. Dipalee Chaudhari[1], Mr. Pramod B. Deshmukh[2], Mr. Navneet Kumar Rajoria[3],

Mr. Tushar Kamble[4], Miss. Tejashree Rokade[5], Miss. Manisha Sharma[6]

[1,2] Asst. Professor, Department of Computer Engineering, D Y Patil College of Engineering, Akurdi, Pune,
MH, India
[3,4,5,6] BE Scholar, Department of Computer Engineering, D Y Patil College of Engineering, Akurdi, Pune,
MH, India

ABSTRACT

Spreadsheets are the most widely used database management applications with interactive Graphical User Interface. Almost all the business, industrial or commercial databases are stored using spreadsheets. It has been observed that a spreadsheet can be used as a relational database management system without using any programming language paradigm. Utilizing various operators of relational algebra by using spreadsheet functions it can be achieved. By giving a particular definition of database in SQL, a workbook of spreadsheet with empty datasheets and sheets of formulas for SQL queries can be implemented. All the data manipulations operations such as insertion, deletion, retrieval and modification performed by users can then be handled by spreadsheet functions. But the end-users must be aware of SQL Queries in order to use such a model. In this paper, we approach such problem by carrying out mapping of natural language questions into their associated SQL queries.

Keywords – *Natural Language Processing, Relational Database Management Systems, Spreadsheets, SQL.*

1. INTRODUCTION

Spreadsheets are counterpart of desktop application. And SQL is query language to access data from database like MySQL. Spreadsheets and SQL stores data in the structured format. Spreadsheets are direct manipulation interface for non - technical person for handling datasets. Most commonly users uses spreadsheets for data storage. But for using databases user requires the knowledge of syntax of query languages such as SQL.

Spreadsheets and SQL has some challenges. Consider example that a person is interested in late bikes models (2008 or later) in good or excellent condition, and he would like the results grouped by Model and ordered by Price.
This is a simple query to specify in SQL. But, on a spreadsheet, just by pointing and clicking, there is no straightforward way to state all these requirements at once in a required view. Instead, the person has to break down his need into parts, and specify one part at a time (e.g. “select late model bikes”). Challenges face by such queries are specified by Bin Liu, H.V. Jagadish [1].

1. Query division challenge: In spreadsheets large query has to apply on data by dividing data in parts and then apply query on each part. Easy to apply on SQL, as query get executed on all database.
2. Grouping challenge : Due to division problem in spreadsheets intermediate result is generated and finally intermediate results are combined
3. Aggregation challenge: Aggregation is easy in spreadsheets. Computation of a relational aggregate query results in the definition of a new relation which is typically not union compatible with the original relation, making it hard to store the data.

It is demonstrated that spreadsheets can implement all data transformations that are definable in SQL by utilizing spreadsheet formulas. It provides query compiler that can transform SQL queries into spreadsheet of same semantics. It is get easier for users who do not want to migrate to database to access data. It also provides basic relational algebraic data transformation formulas. This paper proposed translator which can transform data in between spreadsheets and SQL using SQL queries [2]. This functions can be categories as,

1. Implementation of algebraic notations: It implement algebraic notations on spreadsheets. During applying algebraic notation it does not consider null values or eliminate duplicate values
2. Performing aggregation operation: For efficient searching it uses aggregation functions
3. Sorting and searching: It provides way for efficient searching and sorting data in ascending or descending order

Spreadsheets can act as computational tool for data storage and querying on relational data. But existing system does not provide efficient searching or transformation on spreadsheet .To overcome this drawback this paper implements the searching algorithm DFS/BFS which provides the efficient searching. This paper provides tool called as Query Converter. Query-Converter is an automated tool converts the SQL query into the spreadsheets [3]. The steps are as follows:

1. The SQL expression is translated into relational algebra expression as per the algorithm [3].
2. Then relational algebra expression translated into spreadsheet as per operator implementations [3].

As described above spreadsheets and SQL can correlate. Spreadsheets can be handled by SQL query commands as spreadsheets are difficult to handle that is retrieving and updating in spreadsheet is exhaustive task. But non- technical person will face problem to access data using query language. To address this problem Natural Language Processing can be used. If there is facility for user to give a query to user in the form of natural language then it will be efficient for end user. So that end user no need to know about backend processing of database ,user can only type query in natural language preferably English(as a standard) and get direct output.

2. PROPOSED PROJECT

Issues related to the drawback of the practical implementation of databases operations in spreadsheets have been tackled. First we will look at the issues, first being the limitation on the numbers and size of relations, views and intermediate results, which are imposed by maximal available number of worksheets, columns and rows in present spreadsheet systems. Second, the size of data value is limited (integers, string...). Variety of data types in spreadsheets is restricted when compared to databases system.

Architecture of a relational database implemented in a spreadsheet. When we give a specific query in SQL, its implementation is generated by our query compiler, in the format of .xlsx extension. The generated spreadsheet is implementation of the given query, is a single sheet, containing required number of columns for the data tables and next to them, the columns performing the computations. At first user will have two rows of formulas, user is required to mark second row of formulas then fill with its content maximum necessary to store that data, also intermediate and final results should be produced.[5]

No action should be performed on first row, as many a times it has formulas which are different from those that fill the remaining rows. So, particularly this means, formulas are totally discrete of the data these formula work on. When data is entered by user manually into tables, involuntary recomputation of worksheet is initiated that causes output of queries to be evaluated and appear in columns with result. It is a critical assumption to assume that data entered by user does not contain any spreadsheet error codes. This code information is used for representation of special information, and our assumption puts us at risk as it would have been misinterpreted considering the initial data.[5]

3. IMPLEMENTATION

System is similar to a layered architecture functioning layer after layer. Each layer can be considered as a module. Spreadsheet and database are important elements but they complement each other. The functionalities of both the systems are considered to overcome the challenges of each system. The architecture represents the way to store data in the spreadsheets using SQL queries in the natural language form.

The initial module, module 1 is the user interface part. User will submit the query to the system through the interface, query can be a SQL or a natural language (preferably English). For example, to find the total number of employees working in a company can be the user input.

Next is the converter phase, module 2 of the system. Module 2 will map user input to its respective SQL query. It is basically a natural language processor which converts the natural language to appropriate SQL queries. It uses the Hidden Markov Model to find the probability of any word from natural language being a particular SQL clause or keyword. It classifies the text of natural language to a particular domain.

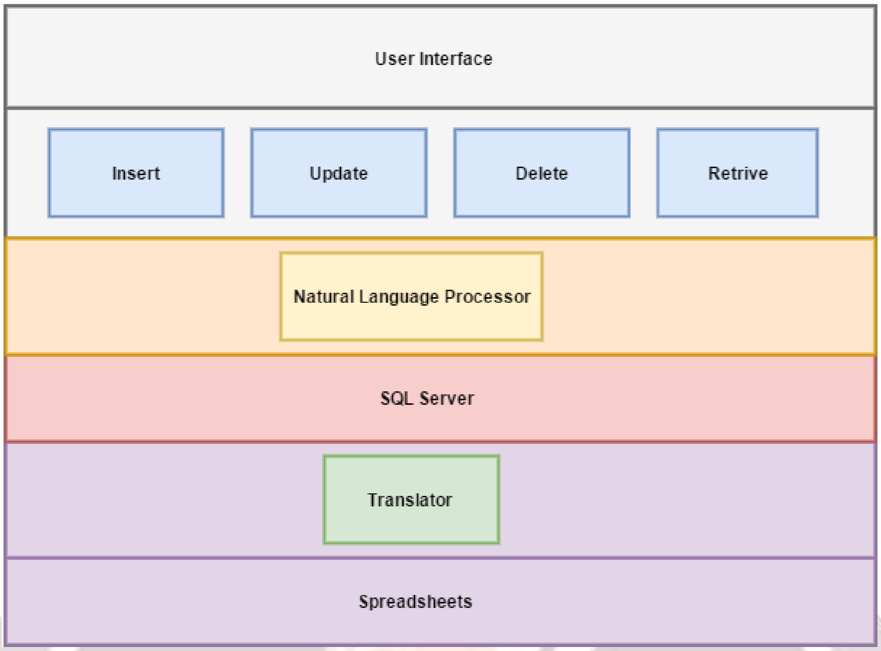

Fig 1-System Architecture

Module 3 is the translator system mapping the SQL function to the corresponding spreadsheet function. When an operation is performed on spreadsheet the required action is performed at backend, mapping process will consider this process. Translator working will be similar to functionality of compiler. The SQL to Excel [4] like translator is considered as an existing work. For efficient data searching or retrieval Depth-First-Search and Breadth-First-Search is used. At the end of processing user will get the result at the user interface layer.

Module 4 consists of relational database system, MySQL is proposed along with dataset imported from spreadsheets. Spreadsheet contains user dataset and query is going to be fired on this dataset to perform various data manipulation operations. MySQL and Spreadsheets are combined at this end. It means that Spreadsheet will be accessed but by using the queries of MySQL. Thus, Spreadsheets can be accessed as Relational Database Management System.

4. COMPARISON

Sr. No.	Year	Name	Feature
1	1994	A Multi-Set Extended Relational Algebra A Formal Approach to a Practical Issue[3] Paul W.P.J. Grefen Rolf A. de By	This paper proposes an extended relational algebra with multi-set semantics. This can be used as a well-defined database manipulation language on its own
2	1994	Real Spreadsheets for Real Programmers[6] Alan G. Yoder and David L. Cohn	This paper define a spreadsheet mini-language, then use it to program solutions for number of real-world problems with reasonable elegance, while maintaining the parallelism implicitly
3	2002	On Querying Spreadsheets[7] Laks V.S. Lakshmanan Nita Goyal Subbu N. Subramanian Ravi Krishnamurthy	Interoperating among spreadsheet, RDMS. The concept of an abstract database machine (ADM) that uses the layout specifications to provide a relational view of the data in spreadsheet applications, and similarly to DBMS, sup-ports efficient querying of the spreadsheet data.

4	2009	A Spreadsheet Algebra for a Direct Data Manipulation Query Interface[1] Bin Liu 1 , H.V. Jagadish	It states an important theorem: For every core SQL single-block query expression there exists an equivalent expression in the spreadsheet algebra such that the result of evaluating either expression against any set of relations is identical.
5	2009	Semantic Mapping Between Natural Language Questions and SQL Queries via Syntactic Pairing[8] Alessandra Giordani and Alessandro Moschitti	Proposes NLP to SQL mapping at syntactic level. Then applying machine learning algorithms to derive SQL queries. The proposed experiments of the automatic translation system show a satisfactory accuracy, i.e. 76%.

5. CONCLUSION

We have demonstrated that Spreadsheet can be implemented with the help of SQL. Thereby, increasing the strength of Spreadsheets by using relational database model. Moreover, searching efficiency of Spreadsheet is increased by using traversing algorithms: DFS and BFS.

A Natural Language Processor has been implemented to increase the interactivity of Spreadsheets for end-users. The efficiency of Natural Language Processor is increased by using Hidden Markov Models for finding the probability of a word from natural language in the clauses of SQL queries along with appropriate semantics.

As for the next step, we plan to develop the Natural Language Processor which can process input from different languages. The efficiency of Natural Language Processor needs to be increased so that it can efficiently convert the natural language statements to appropriate and accurate SQL queries so that relevant data can be acquired.

6. REFERENCES

[1]. 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.

[2]. J. V. den Bussche and S. Vansummeren, “Translating SQL into the relational algebra,” Lecture material, Universiteit Limburg, lecture INFO-H-417: Database Systems Architecture

[3]. P. W. P. J. Grefen and R. A. de By, “A multi-set extended relationalalgebra-A formal approach to a practical issue,” in Proc. Int. Conf. Data Eng., 1994, pp. 80–88

[4]. SQL to Excel translator, described in Implementation section is also available from http://sourceforge.net/ projects/sqltoalgebra/?source=directory

[5]. Jacek Sroka, Adrian Panasuik, Krzystof Stencel and Jerzy Tyszkiewicz,”Translating Relational Queries into Spreadsheets", IEEE Transactions on Knowledge And Data Engineering, Vol.27, No.8, August 2015

[6]. A. G. Yoder and D. L. Cohn, “Real spreadsheets for real programmers“, in Proc. Int. Conf. Comput. Language, 1994, pp. 20-30.

[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]. Alessandra Giordani and Alessandro Moschitti, "Translating Questions to SQL Queries with Generative Parsers Discriminatively Reranked", Proceedings of COLING 2012: Posters, pages 401–410, COLING 2012, Mumbai, December 2012.