International Journal of Engineering In Advanced Research Science and Technology
ISSN: 2278-2566
VOLUME-2
ISSUE-5
NOV-2016

AN EFFICIENT APPROACH FOR CONVERTING RELATIONAL QUERIES TO SPREAD SHEETS

S.Surekha [1],G.Rajeshekar Reddy [2]

[1]M.Tech Student, Dept.Of CSE, Sarojini Institute of Technology, Telaprolu(V), Unguturu (M), Gannavaram, krishna (D). A.P

[2]Assistant Professor, Dept.Of CSE, Sarojini Institute of Technology, Telaprolu(V), Unguturu (M), Gannavaram, krishna (D). A.P

ABSTRACT:

Data mining the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Spreadsheets are the desktop counterpart of databases and OLAP in enterprise scale computing. They serve basically the same purpose data management and analysis but at the opposite extreme of the data quality scale. We demonstrate that a spreadsheet can implement all data transformation definable in SQL merely by utilizing spreadsheets 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 spreadsheets. 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.

Keywords— Relational databases, physical database design prototypes, spreadsheets, query languages

1. INTRODUCTION:

They combine data processing with very diverse supplementary features: statistics, visualization, reporting, linear programming solvers, Web queries periodicallydownloading data from external sources, etc. However, the spreadsheet paradigm of computation still lacks sufficient analysis. Computer applications in the form of formulaonly spreadsheets are therefore highly portable, probably to the extent comparable with Java byte code. Spreadsheet systems can be regarded as virtual machines, offered by various vendors, on which spreadsheet applications can be run. It is therefore extremely surprising that those machines are predominantly programmed manually, with no compilers producing spreadsheet code from higher-level languages. In our proposed system, we offer a fully automated method to construct spreadsheet implementations for a wide class of relational data transformations. We have reimplemented all operators of relational

algebra to accept a variable number of input columns and to support NULL values. The full automation of the translation process reduces the number of human-introduced errors in the spreadsheet application. End users can still work in the vanilla spreadsheet, benefit from its features like data analysis and visualization, while the complex formulas are generated by a tool that allows to express them in a better suited high level language and avoids errors. One of them is an efficient sorting algorithm, implemented by spreadsheet formulas, improving on the quadratic sorting.

2. SCOPE OF THE PROJECT

2.1. OBJECTIVE:

The main aim of this project is to propose translating relational queries into spreadsheets and can define their queries using a high level language and then get their execution plan in spreadsheet.Spreadsheets are the desktop counterpart of databases and OLAP in enterprise scale computing. They serve basically the same purpose data management and analysis but at the opposite extreme of the data quality scale. We demonstrate that a spreadsheet can implement all data transformation definable in SQL merely by utilizing spreadsheets formulas.

2.2. SCOPE:

Spreadsheets are the desktop counterpart of databases and OLAP in enterprise scale computing. They serve basically the same purpose data management and analysis but at the opposite extreme of the data quality scale. We demonstrate that a spreadsheet can implement all data transformation definable in SQL merely by utilizing spreadsheets 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 spreadsheets.

3. SYSTEM ANALYSIS EXISTING SYSTEM

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. Computer applications in the form of formulaonly spreadsheets are therefore highly portable, probably to the extent comparable with Java byte code. Spreadsheet systems can be regarded as virtual machines, offered by various vendors, on which spreadsheet applications can be run. It is therefore extremely surprising that those machines are predominantly programmed manually, with no compilers producing spreadsheet code from higher-level languages.

Limitations

- Retrieving Null values from spreadsheet is difficult
- Implementation of sorting and applying join on spreadsheet is not efficient.
- Quality of maintaining spreadsheet is not feasible

4. MAIN FEATURES

- In our proposed system, we offer a fully automated method to constructspreadsheet implementations for a wide class of relational data transformations.
- We have re-implemented all operators of relational algebra to accept a variable number of input columns and to support NULL values.
- The full automation of the translation process reduces the number of humanintroduced errors in the spreadsheet application.
- End users can still work in the vanilla spreadsheet, benefit from its features like data analysis and visualization, while the complex formulas are generated by a tool that allows to express them in a better suited high level language and avoids errors.

5. IMPLEMENTATION:

Application Creation & Inserting Details: In this phase, we are going to develop an ecommerce application for mobile shop. In this application owner as to maintain the details of the each product (incoming and outgoing). To maintain all products details by using excel sheet. All the product information's are stored in excel sheet for easy access. Owner has to insert all the product details, customer details, workers details and bill details in their own excel database.

Implementing algebraic notation: Once excel database has been created, we need to implement the algebraic notation on it. Here we are going to perform selection, union, difference and duplicate removal information. Translating all relational queries into excel sheet for performing notation. All the database will not allow duplicate records into their database. Every application has to maintain removing of duplicate values. In this application, algebraic notations are used when entering product details, viewing product details and entering bill details.

PERFORMING AGGREGATION FUNCTION:

In excel sheet we need to implement aggregation functions. To perform aggregation we need to transform relational queries for excel sheet. For quick searching on product we are going to apply aggregation function. Aggregation functions are max, min, count, sum, less than, greater than etc, Sorting & Searching details: In this phase, we are going to implement sorting and searching operations on excel sheet. Sorting operations will be apply on showing workers performance of every month. For sorting we are implementing both ascending

and descending order. For searching details, we are implementing BFS and DFS searching algorithm. To implementing searching algorithms, we are using join operations on excel sheet. Using join query, searching will be performing on excel sheet.

6. SYSTEM DESCRIPTION System Architecture consist of various blocks as follows:

1. Spreadsheets

2. Worksheet.

3. Query compiler.

4. Resultset.

7. CONCLUSION: We have established that SQL can be automatically converted into spreadsheet code. Thus, we have shown the power of the spreadsheet paradigm, which subsumes the paradigm of relational databases.Apart from SQL, we have also applied a few specific algorithms: a linearithmic sorting procedure and two graph traversing algorithms: BFS and DFS. As the next steps we plan to develop optimizations for SQL queries translated into spreadsheets. We are also interested whether spreadsheets can naturally implement other models of databases, like semi-structured or object-relational ones.. This requires translating Google specific functions QUERY, SORT, FILTER and UNIQUE which are not recognized by other spreadsheet systems.

REFERENCES:

[1] J. Tyszkiewicz, Spreadsheet as a relational database engine, in Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ser.SIGMOD 10. New York, NY, USA: ACM, 2010,pp.195206.

[2] B. Liu and H. V. Jagadish, A spreadsheet algebra for a direct data manipulation query interface, in ICDE 09: Proceedings of the 2009 IEEE International Conference on Data Engineering. Washington, DC, USA: IEEE Computer Society, 2009,pp. 417 428.

[3] A. Witkowski, S. Bellamkonda, T. Bozkaya, G. Dorman, N. Folkert, A. Gupta,L. Shen, and S. Subramanian, Spreadsheets in RDBMS for OLAP, in SIGMOD 03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data. New York, NY, USA: ACM, 2003, pp. 5263.

[4] A. Witkowski, S. Bellamkonda, T. Bozkaya, N. Folkert, A. Gupta, L. Sheng, and S. Subramanian, Business modeling using SQL spreadsheets, in VLDB 2003: Proceedings of the 29th international conference on Very large data bases. VLDB Endowment, 2003, pp. 11171120. [5] H. Garcia-Molina, J. D. Ullman, and J. Widom, Database System Implementation. Prentice-Hall, 2000.

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