AN EFFICIENT APPROACH FOR CONVERTING RELATIONAL QUERIES TO SPREAD SHEETS

Mrs.Razeena Begum[1], Assistant professor Andhra Loyola Institute Of Engineering&Technology(ALIET),

N.Navya Sri[2] B.Tech(ALIET), R.Sravya Sri[3] B.Tech(ALIET), R.Anu Sri[4] B.Tech(ALIET)

Vijayawada

Abstract— Data mining the consideration of hidden predictive advice from large table, is a powerful new terminology with great hidden to help association focus on the most critical advice in their data establishment. Spreadsheets are the desktop analogue of directory and OLAP in enterprise scale computing. They serve mostly the same purpose data management and inquiry but at the opposite intense of the data aspect scale. We demonstrate that a spreadsheet can appliance all data transformation apparent in SQL merely by utilizing spreadsheets custom. We provide a query compiler, which convert any given SQL query into a worksheet of the same denotation, including NULL values. Thereby database activity become available to the users who do not want to emigrate to a database. They can construe their queries using a high level language and then get their beheading plans in a plain spreadsheets. Spreadsheets are amidst the most commonly used utilization for data management and inquiry. They associate data processing with very disparate supplementary features: statistics, visualization, reporting, linear programming solvers, Web queries annually downloading data from external sources, etc. However, the spreadsheet paradigm of estimation still lacks acceptable analysis.

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

1 .INTRODUCTION

They associate data processing with very differing supplementary features: statistics, visualization, reporting, linear programming solvers, Web queries periodically downloading data from foreign sources, etc. However, the spreadsheet paradigm of computing still lacks acceptable analysis. Computer applications in the form of formula only spreadsheets are therefore highly compact, probably to the breadth comparable with Java byte code. Spreadsheet systems can be observe as virtual machines, afford by various vendors, on which spreadsheet function can be run. It is therefore extremely astonishing that those machines are predominantly programmed manually, with no compilers producing spreadsheet code from higher-level languages. In our expected system, we offer a fully electrical method to compose spreadsheet implementations for a wide class of comparative data transformations. We have reimplemented all operators of comparative algebra to obtain a variable number of input columns and to backing NULL values. The full automation of the explanation process reduces the number of human-introduced errors in the spreadsheet utilization. End users can still work in the vanilla spreadsheet, assistance from its features like data reasoning and visualization, while the complex formulas are achieve by a tool that allows to expressed them in a better suited high level language and avoids errors. One of them is an adequate sorting algorithm, achieve by spreadsheet formulas, developing on the quadratic sorting.

2. SCOPE OF THE PROJECT

2.1. OBJECTIVE:

The main aim of this project is to introduce translating relative queries into spreadsheets and can define their queries using a high level language and then get their beheading plan in spreadsheet. Spreadsheets are the desktop counterpart of databases and OLAP in activity scale computing. They serve essentially the same aspiration data management and analysis but at the opposite intense of the data quality scale. We demonstrate that a spreadsheet can appliance all data transformation apparent in SQL merely by appropriate spreadsheets formulas.

2.2. SCOPE:

Spreadsheets are the desktop counterpart of databases and OLAP in activity scale computing. They serve essentially the same purpose data management and inquiry but at the opposite intense of the data quality scale. We demonstrate that a spreadsheet can appliance all data transformation apparent in SQL merely by employ spreadsheets formulas. We arrange a query compiler, which decipher any given SQL query into a worksheet of the same definition, including NULL values. Thereby database operations become accessible to the users who do not want to migrate to a database. They can construe their queries using a high level language and then get their decapitation plans in a plain spreadsheets.

3. SYSTEM ANALYSIS

EXISTING SYSTEM

They combine data processing with very disparate supplementary features: statistics, visualization, reporting, linear programming solvers, Web queries periodically downloading data from foreign sources, etc. However, the spreadsheet paradigm of computation still lacks acceptable analysis. Computer applications in the form of formula only spreadsheets are therefore highly compact, probably to the breadth comparable with Java byte code. Spreadsheet systems can be observe as virtual machines, offered by assorted vendors, on which spreadsheet applications can be run. It is therefore acutely surprising that those machines are predominantly programmed manually, with no hobbyist producing spreadsheet code from higher-level languages.

Limitations

- Retrieving Null values from spreadsheet is ambitious
- Implementation of sorting and applying join on spreadsheet is not adequate.
- Quality of maintaining spreadsheet is not attainable

4. MAIN FEATURES

- In our expected system, we offer a fully automatic method to construct spreadsheet implementations for a wide class of relational data transformations.
- We have re-implemented all operators of relational algebra to obtain a variable number of input columns and to support NULL values.
- The full automation of the translation process curtail the number of human imported errors in the spreadsheet utilization.
- End users can still work in the vanilla spreadsheet, assistance from its appearance like data analysis and visualization, while the convoluted formulas are achieve by a tool that grant to express them in a better appropriate high level language and avoids errors.

5. IMPLEMENTATION

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

Implementing algebraic characters: Once excel database has been created, we need to appliance the algebraic notation on it. Here we are going to achieve selection, union, difference and duplicate deportation information. Translating all related queries into excel sheet for performing characters. All the database will not allow duplicate records into their database. Every application has to cultivate 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.

[7] P. W. P. J. Grefen and R. A. de By, A multi-set extended relational algebra – a formal approach to a practical issue, in ICDE. IEEE Computer Society, 1994.

.