

Graphical Web based Tool for Generating Query from Star Schema

Mohammed Anbar^a, Ku Ruhana Ku-Mahamud^b

^aCollege of Arts and Sciences
Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia
Tel: 6014-2449604
E-mail: m.3anbar@gmail.com

^bCollege of Arts and Sciences
Universiti Utara Malaysia, Sintok, Kedah, Malaysia
Tel: 604-9284717, Fax: 604-9284753
E-mail: ruhana@uum.edu.my

ABSTRACT

This paper presents the development of a graphical SQL query tool that allows novice and non-technical users to navigate through database tables and generate their own queries. The tool enables the query output to be presented in graphical and tabular forms, which can help users, especially top management in better understanding and interpreting query results. The algorithms to construct complex SQL query from star schema in databases is also presented.

Keyword:

SQL Query Generation, Star Schema, Graphical and Tabular forms

1.0 INTRODUCTION

Database applications which are deployed in many corporations for the purpose of storing and retrieving data in a structured way have become ubiquitous and indispensable. Many transactions are being performed by users daily and these transactions represent the questions asked by users in natural language which are formulated as structured query language (SQL) to deal with database. The ways queries are formulated depend on users' knowledge and experience.

Data which exists in relational database consists of answers for SQL queries. However, the generation of SQL queries are too complex when the database is large where attributes in one table may depend on other attributes in other tables in the same database or in other databases. This will lead to a difficult, complex and time consuming task in the formulation of query. To overcome the problem, Stoffel et al. (1998) proposed the graphical SQL query to allow users to navigate through database tables and generate their own queries.

Many applications such as web based applications are developed with database which acts as a depository of information. In order to achieve access to this information, web based interface is needed for these applications. Chen and Li (2007) highlighted that user can query for information through web based interface. However, the user

may not find an answer for his query for several reasons. One of these reasons is that the query condition is too general, so the user may stop trying before reaches his answer. For this reason, query results must be categorized in a way which will fulfil the user needs. An example of a query is the navigational tree. Each level in the tree represents a condition and if going deeply in navigational tree condition, the result from the query will be more accurate.

The star schema is a way to implement multi-dimensional database functionality using mainstream relational database. It is also the simplest style of data warehouse schema and being implemented widely because it tend to perform the best in decision support applications. Star schema as reported by Rob and Coronel (2008) contains four components namely facts, dimensions, attributes and attribute hierarchies. It maintains one-to-many relations between facts tables and dimension tables. Fact tables are located at the centre of the data warehouse and contains metrics related to multiple dimensional databases (Song et al., 2001). This is supported by Krippendorf and Song (1997), which highlighted that fact table, contains facts linked through their dimensions. This is considered as an underlying model for data warehouse design. Novice and non technical users have difficulty to generate complex structured query language that can produced specific result from database with star schema (Jung & Lee, 2002). Most of the time, the output of the query is not presented in user friendly, traceable and interesting way which requires a user to take considerable time to understand the output.

Top management, customers, stakeholders and top level authority members rarely have enough time to go through the whole reports presented to them. Hence data should be presented in such a manner that enables reader to interpret important data with minimum effort and time. Data presentation can be generally classified into non graphical form such as table and graphical form such as chart. Law et al. (2005) in their study on comparison between graphical and textual presentations of data showed that users prefer to use graphical data presentation rather than textual data presentation.

According to Borgman (1996), textual query language based on Boolean logic is common amongst the search facilities of online information warehouse. However, this query has standard syntax and semantic which lead to user errors and time consuming in formulating query. Experience users deal with systems that provide textual language for query specification to formulate complex information seeking requirement. Difficulties in dealing with Boolean logic are common, particularly when restricted syntax is used.

There is a direct correlation between the scalability of database structure and complexity of queries used to manipulate data. Relational database structure does not support this complexity of queries (Rob & Coronel, 2008). One of the most important issues in data warehousing is how to design appropriate database structure to support end user queries. One of the ways to solve this problem is to develop graphical query tool.

This study has developed a graphical SQL query tool that will allow the user to navigate through database tables and generate their own queries. The tool will enable the query output to be presented either in graphical or tabular form. Section 2 describes the research approach followed by the algorithm to formulate complex queries from star schema tables and strategy for presenting output in graphical and tabular forms in Section 3. The development of the prototype is presented in Section 4 and concluding remarks are given in Section 5.

2.0 RESEARCH APPROACH

The approach adopted by this study consists of four major stages as depicted in Figure 1. The stages are problem analysis, formulating algorithm for query construction and strategy for query result presentation, graphical query formulation and documentation.

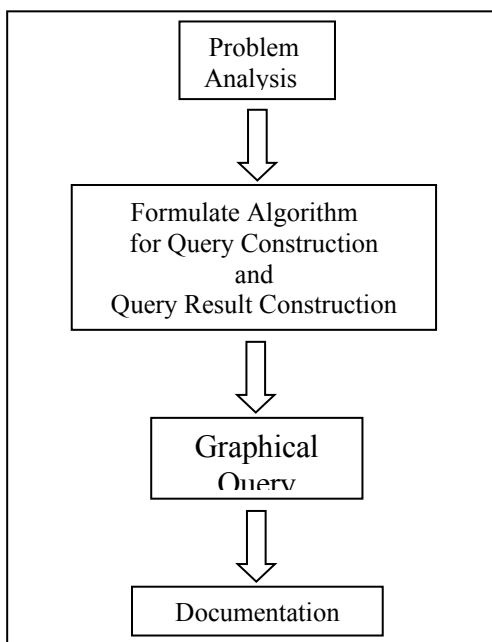


Figure 1: Research approach stages

The first stage in this methodology is to analyze the problem that exists in formulating queries in large databases. Information on star schema and the types of output preferred by users must also be collected and analyzed. This preliminary investigation must identify the alternative ways in presenting queries' results and their significance.

In the second stage of the research approach, algorithm to construct complex queries must be formulated. This formulation depends on the components of the star schema. Decision on the sequence of assessing the components will have to be made as well as assessing the data within the components. Strategy to present the output of the queries will also have to be constructed. The output of this stage is also known as the design of the graphical web-based query tool.

Graphical query formulation is the third stage of the research approach. In this stage the tentative design is implemented. In other word, the development of the prototype is performed. The prototype has been developed using the Prototyping System Development Methodology.

The fourth stage is documentation of the user manual. The manual is used to describe the application, all source information about the application contained in design documents and detailed code comments. In addition, this documentation will be used as reference for future enhancement such as addition of extra functions to existing system. User manual is also used to by developer to track problems in the system.

3.0 ALGORITHM FOR QUERY CONSTRUCTION

Formulating algorithm for query construction is done by allowing user to navigate through star schema tables. Figure 2 shows an example of the star schema structure.

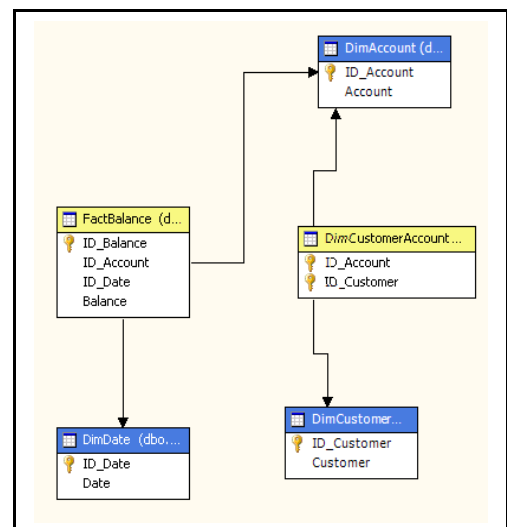


Figure 2: Star schema structure in data warehouse

Star schema contains dimension tables and one or a few fact tables. Fact tables hold the main data, while the usually smaller dimension tables describe each value of a dimension. Dimension tables have a simple primary key, while fact tables have a compound primary key consisting of the aggregate of relevant dimension keys. Fact table is the first table that a user will use to formulate his query. From there, dimension table will be accessed by using their primary keys which exist in the fact table.

An algorithm has been developed for the purpose of navigating through the tables and generating complex SQL queries. The algorithm consists of seven steps as follows:

1. Select fact table
2. Select aggregation function (min, max, count, AVG and sum).
3. Select parameter for aggregate function
4. Select two fields from fact table
5. Select dimensions for each field
6. Specify report configuration such as which field will be at the header or column and determines report color.
7. Determined tables, aggregation functions and save it in database with its configuration

This algorithm is simple to implement even for complex query that the user wants to construct. The presentation of the output can be determined by the user once the query has been formulated. The user then specifies the name of the report as well as the presentation form either graphical or tabular.

4.0 PROTOTYPE DEVELOPMENT

The Prototyping System Development methodology was used to develop the graphical web-based tool prototype. The three steps involved in this development methodology are as shown in Figure 3.

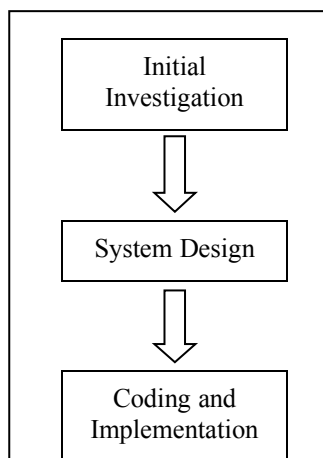


Figure 3: Prototyping system development methodology

The first step is the initial investigation where the requirements of the prototype are identified. This includes

system functionality such as users will be able to generate cross tabulation, vertical bar chart and pie chart report forms. Users will also be able to create, delete, view and update the outputs. Non-functional requirements of the prototype have also been identified. This can be used to judge the operation of a system, rather than specific behaviors. The non-functional requirements are availability, reliability, manageability, flexibility, extensibility and security.

Administrator and user are two actors that have been identified for the prototype. Figure 4 depicts the used case diagram that displays the relationship between actors of the prototype.

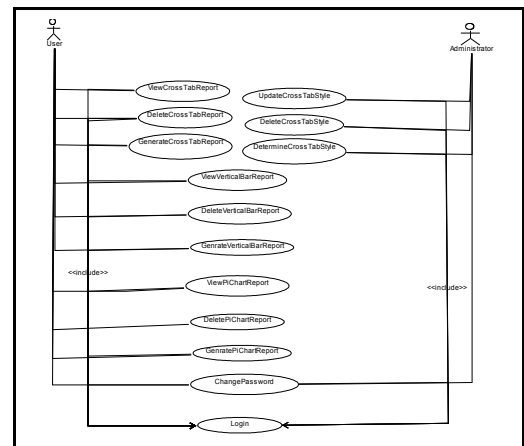


Figure 4: Use case diagram

Figure 5 displays the class diagram that shows the classes within a model. In an object oriented application, classes have attributes (member variables), operations (member functions) and relationships with other classes.

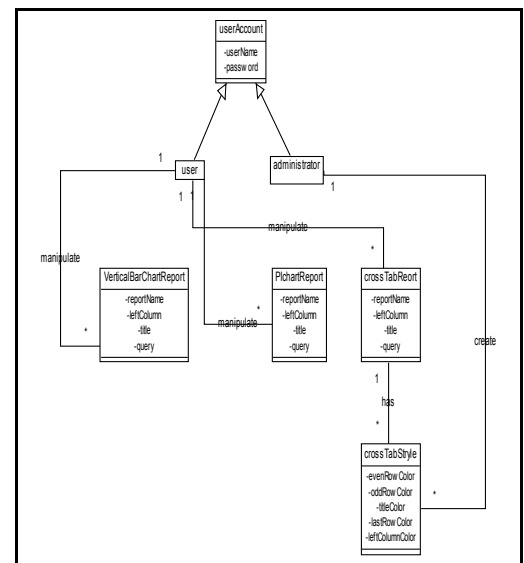


Figure 5: Class Diagram

The sequence diagram (refer Figure 6) is developed in the second step of the methodology. Developers use sequence diagrams to document the dynamic aspects of an object model. It could also display object interactions more understandable by tackling the dynamics behind major business methods.

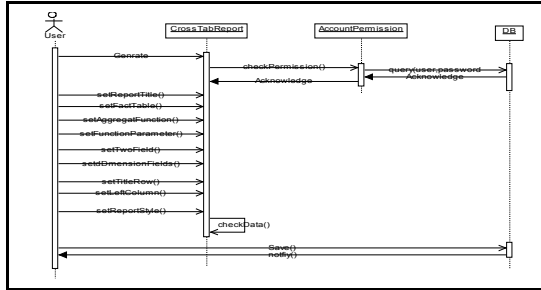


Figure 6: Sequence diagram to generate tabular report

The third step of the development methodology is to transform the design of the prototype into executable application that can be run and maintained against errors.

The developed prototype is a web based system application in which web pages were coded with a server side scripting language i.e Java Server Pages (JSP). As part of the Java technology family, JSP enables rapid development of web-based applications that are platform-independent. JSP separates user interfaces from content generation, enabling designers to change the overall page layout without altering the underlying dynamic content.

Figures 7 and 8 displayed the snapshots taken from the developed prototype where a query output is presented in tabular and graphical form respectively.



Figure 7: Tabular data presentation

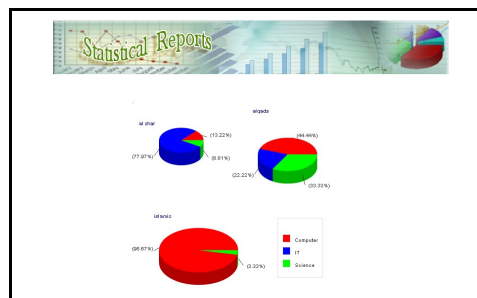


Figure 8: Graphical data presentation

The query that has been made is to count the number of staff in three faculties from three universities. Output from the query is presented in tabular form as shown in Figure 4 where the column represents the faculties and the row represents the three universities. Each university is presented by a pie chart and every color in the pie chart has a meaning (refer Figure 5). Red represents the computer faculty, blue represents the information technology faculty and the science faculty is represented by the green color.

The prototype has eliminated the problem faced by non-technical and novice user in formulating query that will generate specific output. In addition, location independent and accessibility is not an issue since the prototype can be accessed via the Internet.

Implementing graphical query generation with this prototype can overcome semantic and syntax errors in SQL. Graphical query language also provides a natural medium for Boolean query specification which can overcome the problems of textual query language. Table 2 displays the comparison in generating SQL queries using graphical tool and textual approaches.

Table 2: Generating SQL query using graphical tool versus textual tool

Criteria	Graphical	Textual
Time	Shorter time	Longer time
User	Novice	Sophisticated
Expiration connection	Easily done by navigating through tables and selecting appropriate attributes, conjunctions and output form	Users need to specify
Navigation	Allow users to navigate using graphical interface	Not allowed to navigation between tables
Presentation	Can add extra features to make the output attractive	Limited control in the output presentation

It is hoped that this prototype will fill the gap between sophisticated and novice users in formulating SQL query. In addition, presenting SQL result in graphical and tabular forms will make the tracing of data by users easier and more interesting

5.0 CONCLUSION

Implementing graphical query generation can guarantee that semantic and syntax in SQL, which can overcome the user's errors and time consuming procedure. Graphical query language provides a natural medium for Boolean query specification which overcomes the problems of textual query language.

Problem faced by non-technical and novice user in formulating query can be solved by using the developed prototype. In addition, location independent and accessibility will be supported because the proposed tool is web-based. The output of SQL query that is presented in tabular and graphical forms will facilitate users who hold crucial positions, in their decision making.

REFERENCES

- Borgman, C. L. (1996). Why are online catalogs still hard to use? *Journal of the American Society for Information Science*, 47(7), 493-503.
- Chen, Z. & Li, T. (2007). *Addressing diverse user preferences in SQL-query-result navigation*, New York: ACM Press.
- Jung, H., & Lee, G. G. (2002). Multilingual question answering with high portability on relational databases. *Proceeding of the International Conference On Computational Linguistics*, 1-8.
- Krippendorf, M. & Song, I. Y. (1997). The translation of star schema into entity-relationship diagrams, *Proceeding of the Database and Expert Systems Applications Workshop*, 390-395.
- Law, A. S., Freer, Y., Hunter, J., Logie, R. H., McIntosh, N., & Quinn, J. (2005). A Comparison of Graphical and Textual Presentations of Time Series Data to Support Medical Decision Making in the Neonatal Intensive Care Unit. *Journal of Clinical Monitoring and Computing*, 19(3), 183-194.
- Rob, P. & Coronel, C. (2008). *Database system: Design, implementation and management*, Fifth edition, Boston: Thompson Learning Inc.
- Stoffel, K., Davis, J. D., Rottman, G., Saltz, J., Dick, J. & Merz, W. (1998). A graphical tool for ad hoc query generation. *Proceedings of the American Medical Informatics Association's Annual Symposium*, 503-507.