Optimizing the Capability of SQL Queries towards Improving Database Transparency

Ruben A. Parazo[*], Dr. James A. Esquivel [**]

[*] Tarlac Agricultural University, Angeles University Foundation
[**] Angeles University Foundation

Abstract- Utilizing previously executed queries contributes in speeding up the performance of database in responding to future queries as it can reduce the number of database queries to be processed and sent back to the user. This technique avoids the re-evaluation and validation of the query being sent to the database. On the testing of the developed algorithm, shows that a one-time access of a certain query to the database was able to reused, respond and served the requirements of seven (7) different requested queries. The number could be higher if the number of fields in a query will be increased. In a scenario where the same information are to be accessed by a substantial number of users, a single access can be possibly cater the needs of the entire users which substantially reduced the amount of request being sent to the database. The process shifted some of the workloads to the application program that is usually processed by the database was achieved in the study which gives partial autonomy to the application program in terms of responding to users' requests in the form of queries. The designed algorithm was able to fetched and reuse previously executed query to respond to future queries executed by the user which applies the principle of prefetching operation. Hence, database transparency is improved.

Index Terms- Optimization, Database, Transparency, Reusability, Queries, Result set, Mapping, Prefetching.

I. INTRODUCTION

Reusing Queries contributes in speeding up the performance of database in responding to future queries as it can reduce the number of database queries to be processed and sent back to the user. Moreover it can also decrease the utilization of database resources according to West **[1].**

A typical request of information in the database performs some steps before a user able to view the requested information. 1. The user formulates query using the application software. 2. The application software connects to the database and submits the query. 3. The database retrieves data and returns these to the user. 4. The application software receives the incoming data, and presents them to the user. These four steps will be repeated from time to time for every request that will be made **[2].** This architecture has placed the bulk of the workloads on information retrieval operation on the database. Unanticipated database problems will prevent the user to request information.

Retrieved data from the database will be profiled and this will be used to respond to future user's queries in order to avoid the re-evaluation of the query by the database as pointed out in the study of Henrik Hygerth and Michael Hakansson entitled Efficient Web Scrapping and Database Synchronization **[3]**.

The users' machine will play active role in responding future queries by storing and reusing the previously executed query. Every executed query was evaluated and compared to the previously executed queries. Queries that cannot be possibly answered by the previously stored query was profiled while queries that are similar or contains the subset of the previously executed queries then the previously executed query was used in providing the information in response to the requested query **[4].**

Process of query profiling is by way of assigning a QUERY IDENTIFICATION NUMBER for the particular query. The same QIN will be used to profile its result set. The QUERY IDENTIFICATION NUMBER will serve not only as the association between the query and its result set but also used to map the correct result set to be displayed to the user. The reuse of previous query not only saves time in processing query requested by users but also the evaluation process to be done by the database based on the research conducted in 2013 by Atul Thakare et al.**[5]**

Organization of stored queries was done to effectively response to the user's request and to facilitate the processing, searching, storing as well as the retrieving information and data. Filing system reduces the amount of resources in processing user's request **[6]**. File system is a systematic way of organizing files to facilitate the process of searching, storing and retrieving information and data. It guides the user where and how to obtain data it needs. Effective filing system reduces the amount of resources in processing data.

The principle behind in this study that it partakes in the information technology area is "ACCESS ONCE VIEW MANY". One time access of data from the database generates profile of the query which was used repetitively in responding to the future queries to be executed by the user. By utilizing the previously stored query, dependency of the application program to the database will be minimized **[7].**

The unique contribution of this study is by making the application program active instead of being passive in responding to users request by delegating some of the work to the application program that was usually process by database. The storing, evaluation of queries and reusability of its result set will give partial autonomy to the application program in providing data to the user. **[8].**

II. OBJECTIVES

The objectives this study is to developed algorithm that will optimize the capability of SQL Queries Towards Improving Database Transparency.

Specifically it aims to;

1. Profile a query and assign particular Query Identification No.
2. Store query and its result set.
3. Reuse and extract future queries and its result set in responding future queries to be executed by the user.
4. Evaluate a query to determine the viability of sourcing out the data either in the previously executed query or in a database.

III. METHODOLOGY

Figure 1. The Diagram for query profiling.

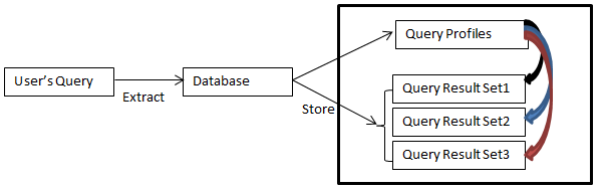


Figure 1, shows the diagram on how to profile a query. The query requested by users will be extracted from the database. The extracted information along with the query will profiled. The query profile will contain QUERY IDENTIFICATION NUMBER which will be concatenated in the query. The QUERY IDENTIFICATION NUMBER and the QUERY will be use for the evaluation and mapping to identify the correct result set to be extracted and to be displayed to the user. It also serves as the key to established association between the Query and to the Result set. Every requested query that could have a major different result such as the fields and table source will be profiled.

Figure 2. Diagram for extracting data in the local data source for the re-executed query.

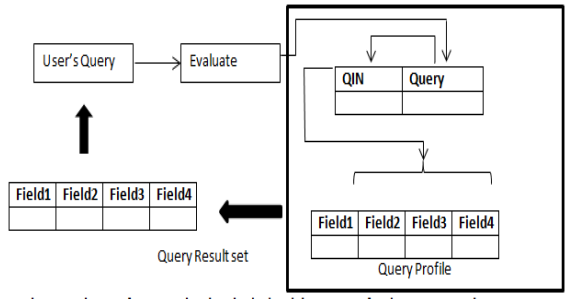


Figure 2, shows the diagram on how to extract and display data. Queries that are evaluated and found to be similar to the previously executed then the query requested will routed to source out its data to the previously executed and profiled query. The QUERY IDENTIFICATION NUMBER which was profiled in conjunction with the Query will be used to map the correct result set to be displayed to the user.

Figure 3. Diagram for extracting data in the previously profiled queries with minor difference.

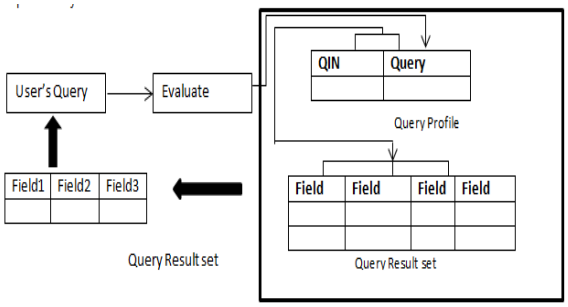


Figure 3, shows the diagram on how query served by the previously executed query. A subset of the query which can be found on the previously executed query can also source out its result set on the previously executed query. When a query will be run, it will be evaluated to the lists of previously executed and profiled query. After evaluation, it will be followed by the population of the result set using the QIN. The next step is to identify the subset of the query followed by the iteration to populate the result set which then displayed to the user.

Method of REQUESTING QUERY in the database

Query Decomposition was employed in order to enhance its ability to answer future queries to be executed by the user. Query decomposition is a useful mechanism to speed up query evaluation based on the claim. of Athul Thakare et al in their study entitled Query Optimization in OODBMS using Query Decomposition & Query Caching **[9]**. From the decomposed query, only the Select and From statement will sent to the database for evaluation. The information sent back by the database is the result set of the query. The result set will be profiled and used later as a reference to answer future queries to be executed after determining its viability through evaluation . (Yongjoo Park, Database Learning:Toward a Database that Becomes Smarter Every Time, 2015)**[10]**.

Method of Displaying Information in response to subsequent query with condition

For subsequent query that will have condition, the query will undergo decomposition process. The condition will be taken out from the query before the evaluation and comparison to the past and profiled queries to determine if the query is capable to be served by the previously executed query. The condition will be organize and used later to filter out the information to be displayed to the user.

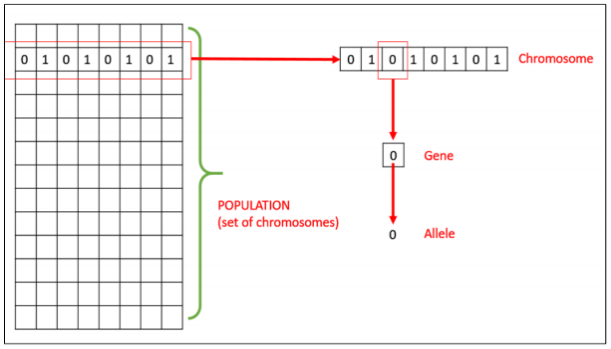
If the evaluation found to be positive then the algorithm will mapped the result set of the past query then it will extract and populate. From the populated record, It will choose those field that are present in the query. Every record extracted from the profiled result set must satisfy the condition set (the condition taken out during decomposition will be used) in the query to be

able to be included in the list of result set to be displayed to the user as response to the requested query.

Process of reusing extracted query result set (partly based on Genetic Algorithm).

1. Query evaluation will be done to determine which the previously profiled query will be used to provide data for the requested query.
2. Determine the subset to be used (subset of the result set of the previously stored query)
3. Extraction using iteration to populate results that fits on a certain condition set.(Genes> Allele)

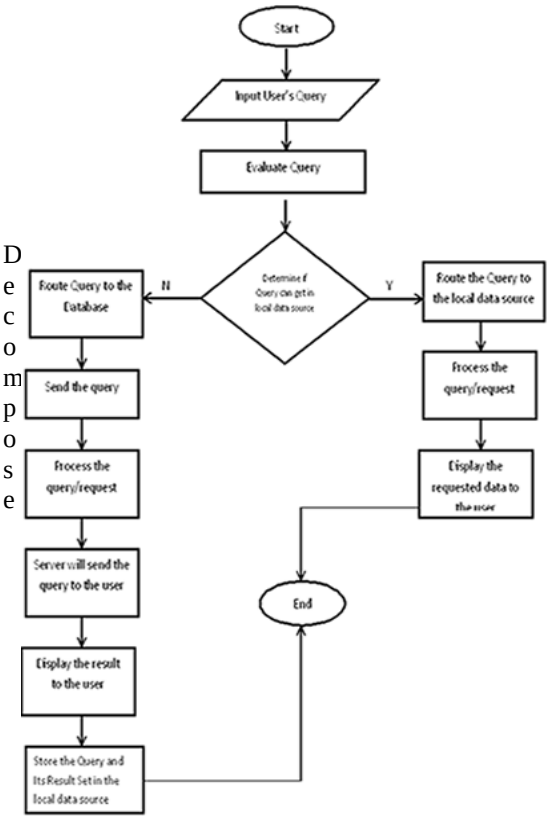
~~Figure 4~~).Diagram for reusing extracted query result set.



Design of the Algorithm

1. Decompose the query.
2. Analyze and query.
3. Determine the Select and From statement in the query.
4. Evaluate the query to the previously executed query.
5. If it is possible to source the data from the result set of previously executed query then
   a. Determine which of the result set of the previously executed queries will be used.
   b. Evaluate and Determine the qualified subset/field.
   c. Do iteration to extract the record one by one and apply the condition set.
   d. Display the information to the user.
6. If the query cannot be serve any of the previously executed query and its result set then.
   a. Send the query to database.
   b. Profile the query.
   c. Profile the result set.
   d. Do iteration to extract the record one by one and apply the condition to filter the information to be displayed to the user.
   e. Display the information to the user.

Figure 5. The flow chart showing the details of how the architecture/design will respond to the user's request



.

Results and Discussion

Figure 6. The first executed query

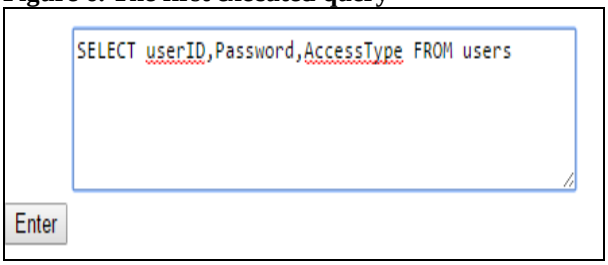


Figure 7. Result set of the first executed query.

admin	admin	admin
11	11	teacher
22	22	teacher
1	a	stud
2	2	stud
3	3	stud
4	4	stud
5	5	stud
33	33	teacher
44	44	teacher
55	55	teacher
66	66	stud
77	77	stud
88	88	stud

On the first execution of the query **SELECT userID,Password,AccessType FROM users** the algorithm profiled the query and its result set. The query has been routed to the database, processed the query and returned its result set to the user. There are two (2) reasons why the algorithm routed the query to source its result set to the database. 1.) This was done because the query is either the first time to execute. 2.)No previous queries that are executed to be used that fits to the requirement of the executed query. After the profiling process the query is now qualified to be utilized for comparison and for possible fetching of its result set to response to the subsequent queries that will executed.

From the query executed, the following possible queries was answered and their result set fetched-in in the result set of the previously executed query.

1. SELECT userID FROM users
2. SELECT Password FROM users
3. SELECT AccessType FROM users
4. SELECT userID,Password FROM users
5. SEECT userID,AccessType FROM users
6. SELECT Password,AccessType FROM users
7. SELECT userID,Password,AccessType FROM users

The result set of the abovementioned queries are obtained in the previously executed query. Their evaluation and validation by the database are prevented. Therefore the resources of the database needed to respond from these queries are not utilized. With one time execution of the query **SELECT userID,Password,AccessType From users,** this query are capable of responding to the seven (7) possible queries that might be executed by the user. The number of queries to be served by the executed query could be higher if the number of fields or subset will be increased.

This indicates that the more queries to be executed and profiled the more possible queries to be served even without the support of the database. Therefore in case of databased related issues that may affect the requesting of information; those possible queries are shielded from such problems because the data are now available in the client side and ready for fetching.

In the study, the prefetching process was utilized. Prefetching is done by the extraction of information prior to its utilization. This principle was takes place in the reused of the result set of the previously executed query which was profiled in the application program as a response to the subsequent queries to be formulated and executed by the user.

In a scenario where the same information are to be accessed by a substantial number of users like their individual information, a single access can be possibly cater the needs of the entire users which substantially reduced the amount of request being sent to the database.

SAMPLE QUERIES AND THEIR RESULT SET WHICH THAT WAS OBTAIN FROM THE PAST QUERY.

Figure 8.The second query executed.

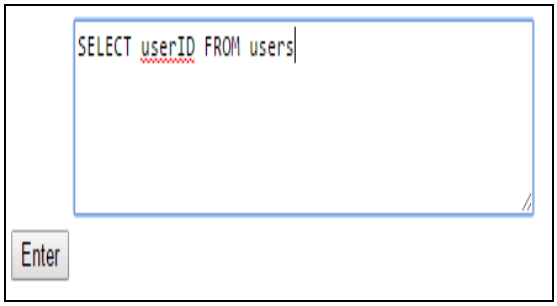


Figure 9.)The result set of the second query

admin
11
22
1
2
3
4
5
33
44
55
66
77
88

Figure 10. The third executed query.

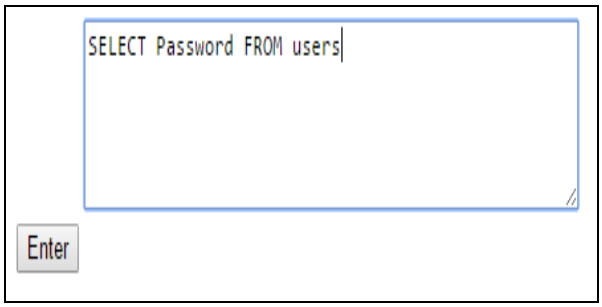


Figure 11. Result set of the third query executed query.

admin
11
22
a
2
3
4
5
33
44
55
66
77
88

Figure 12.)The fourth executed query.

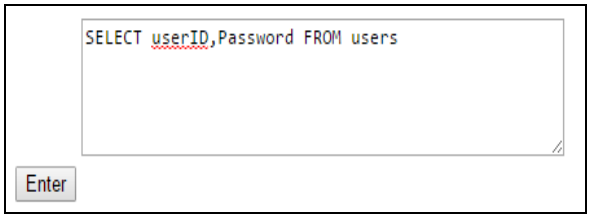


Figure 13.The result set of the fourth query executed.

admin	admin
11	11
22	22
1	a
2	2
3	3
4	4
5	5
33	33
44	44
55	55
66	66
77	77
88	88

Figure 14. The fifth executed query.

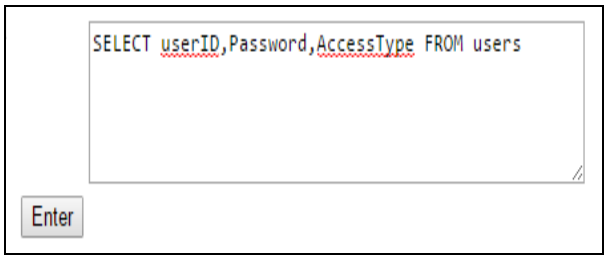


Figure 15.)The result set of the fith query executed.

admin	admin	admin
11	11	teacher
22	22	teacher
1	a	stud
2	2	stud
3	3	stud
4	4	stud
5	5	stud
33	33	teacher
44	44	teacher
55	55	teacher
66	66	stud
77	77	stud
88	88	stud

IV. CONCLUSIONS

It is concluded that in order to lessen the dependency of application programs to the database, every query executed must be profiled and reuse it as a reference to the future queries to be executed. This is one the strategy that can be implemented towards attaining database transparency as well as reducing database workloads. The algorithm was able fetched and reuse

previously executed query as a response to future queries to be executed by a user. By doing this some of the workload of the database was shifted from the application program.

V. Recommendation

In order to maintain the integrity of result set of the previously executed query, It must be synchronized to the database

References

[1] West, M. (2013, January 8). *Storing data on the local client with LocalStorage*. Retrieved August 18, 2017, from http://blog.teamtreehouse.com/storing-data-on-the-client-with-localstorage: http://blog.teamtreehouse.com/

[2] Mark Raasveldt, H. M. (2017). Don't Hold My Data Hostage – A Case For Client Protocol Redesign. *Proceedings of the VLDB Endowment, Vol. 10, No. 10.* Creative Commons AttributionNonCommercialNoDerivatives.

[3] Henrik Hygerth, M. H. (2016). Effcient Web Scraping and Database Synchronization. *Modern Database Systems and Their Applications*.

[4] Yongjoo Park, A. S. (2015). Patent No. Copyright 2015 VLDB Endowment 21508097/15/11. United State of America.

[5] Atul Thakare, P. M. (2013). Query Optimization in OODBMS using Query Decomposition & Query Caching. *International Journal on Recent and Innovation Trends in Computing and Communication*, 469 – 474.

[6] Athicha Muthitacharoen, B. C. (2014). *A Low-bandwidth Network File System.* New York: MIT Laboratory for Computer Science and NYU Department of Computer Science.

[7] Browman-Amuah, M. K. (2003). *Patent No. US 6615253 B1*. USA.

[8] Nodira Khoussainova, Y. K.-T. (2013). *Session-based Browsing for More Effective Query Reuse*. Washington.

[9] Atul Thakare, P. M. (2013). Query Optimization in OODBMS using Query Decomposition & Query Caching. *International Journal on Recent and Innovation Trends in Computing and Communication*, 469 – 474.

[10] Yongjoo Park, A. S. (2015). *Patent No. Copyright 2015 VLDB Endowment 21508097/15/11*. United State of America.

Authors

First Author – Ruben A. Parazo, Tarlac Agricultural University Angeles University Foundation, raparazo@yahoo.com
Second Author – Dr. James A. Esquivel, Angeles University Foundation, esquivel.james@auf.edu.ph