

## Query Tuning in Oracle Database

Nofri Alhadi and Kamsuriah Ahmad

Strategic Information System Research Group  
Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia

Received 2012-08-03, Revised 2012-08-07; Accepted 2012-11-5

### ABSTRACT

Oracle database is well suited to use for managing and exchanging of corporate data, especially for companies that have large-scale databases. Even though Oracle can work best, however, there are conditions where Oracle does not perform well and consume more time to perform the query transaction. This is when the database tuning process is needed. This process will solve the problems of low performance and high execution time when accessing data in the database. This study aims to propose an improvement process of query tuning that can reduce queries execution time. The cause of the problems can be discovered by reviewing the source code, List of Value (LOV) and view on the current system. To enhance the query, existing query scripts need to be modified to fit the tuning process. As an evaluation on the effectiveness of the proposed query tuning process, a comparison of study is done. The result of the study shows that before the tuning process, a system takes 156 milliseconds to execute the query and after implementing the proposed query tuning process, the execution time decreases to less than one millisecond. It means hundred times performance improvement on the query execution. The outcome of the study proved that the query execution time decreased after implementing the proposed query tuning process in the system.

**Keywords:** Query Tuning Process, Large Scale Database, Oracle

### 1. INTRODUCTION

Relational database technology is the most suitable technology to manage large amounts of data. Relational database stores data in separate tables instead of placing all data in one large table. Data Base Administrators (DBAs) defines the relationships between these tables that based on normalization concept. These relationships enable DBAs to combine data from several tables for querying and reporting. This is accomplished through the use of keys, which uniquely identify specific records in a table. Relational database schemas integration is important in various areas such as merging information systems of two or more companies, merging database systems of two or more departments of one company (Koh, 2007). Relational database technology allows databases to have large storage, execute faster and more efficient. Dr. Edger F. (Ted) Codd first developed

the concept of a relational database in 1970. The software used to perform a relational database is called a relational database management system (RDBMS). Three leading companies that develop a relational database management system commercially are Oracle, Microsoft and IBM (Koh, 2007). Other relational database management systems in Open Source are MySQL, PostgreSQL, SQLite, Firebird, MaxDB, MySQL, PostgreSQL and Sleepycat Software's Berkeley DB (Paulson, 2004).

Oracle is the most widely used large-scale database in business society at present and its performance influences the efficiency of application directly (Li *et al.*, 2009). Oracle database is very suitable for managing and exchanging of corporate data, especially for companies that have large scale databases. It is important for applications to rely on Oracle databases because its ability to maintain and preserve the integrity of data even for a huge database. Database tuning is needed to make

**Corresponding Author:** Nofri Alhadi, Strategic Information System Research Group, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia

the system running smoothly, especially for system with huge database and the transaction is real-time.

Many efforts have been made to optimize uncertain databases, including the development of special index data structures (Kimura *et al.*, 2010). Database tuning not only makes the system run faster but also makes the system to be lighter in terms of data capacity. This is because the database tuning minimizes data to be executed where it just used necessary data only. On the other hand, database tuning is an activity of making a database application execute faster. Increase execution time means higher throughput though it may mean lower response time for some applications (Shasha and Bonnet, 2003).

Information System application is growing very fast in the future and the database is going to be larger. It is because the intensive using of information system in the company. Every day, data is manipulated by inserting, updating and deleting. This condition makes the system running slow and consumes more time to perform the transaction. In other words, it consumes more time for the query execution to access data from the database. To solve this problem, the database tuning is needed. In this study, the database tuning that being applied is the query tuning process. The query tuning itself is also part of the database tuning process. This study is conducted to study this problem and provide the best solution to solve it. This study propose a query tuning process because rewriting query has the positive effect on adding an index, changing the schema and modifying transaction lengths.

The objectives of this study are threefold: (i) to propose a query tuning process to improve the system performance by reducing queries execution time, (ii) to develop a new query based on the suggestion from the query tuning tool and (iii) to evaluate the new query by comparing the execution time of the query to access the database, before and after the query tuning process.

## 2. MATERIALS AND METHODS

The query performance is very influential on the system performance. There are many studies done in the field of query tuning (Khafajeh *et al.*, 2012; Elfaki *et al.*, 2011; Haw and Lee, 2008). Various problems and solutions in the field of query tuning have been proposed with different aims and various approaches. However, previous studies are focused more on the system simplification when building a query system and ignores on the performance of queries.

To overcome the limitation of existing approaches, this study proposed an improved method of query tuning to reduce queries execution time. Based on the reviews,

the process of reducing the query execution time can be done manually or automatically using a tool (Kanaan *et al.*, 2008). The manual query tuning process is done by optimizing the factors that affects the query performance, whereas automatic query tuning process is done using tools. This study proposes a new tool for automatic query tuning process. A study on a new tool is needed, in order to find a powerful and accurate tool for a better query tuning process. This tool must also interact well with Oracle database which is a reliable database management tool. By performing the optimization on database platforms that are already reliable, hopefully it will produce a better query result.

This study proposes to use a tool to automate the result. The results generated by using a tool are a lot better compared with manual process. The human skill limitation on the query tuning can be avoided. But sometimes, manually process is better for a simple query (Khafajeh *et al.*, 2012). Simple and short queries are appropriate when dealing with query that contains keywords such as DISTINCT, HAVING and ORDER BY. As far as the study concerns, there is no query tuning method exists that combines both manual and automatic query tuning process. Therefore a new query tuning method is needed in order to optimize the process. This study proposes a new query tuning method that also includes the analysis of the query statement.

Query enhancement using tuning process in oracle database: The query tuning process done based on the rules in query tuning listed as follows (Shasha and Bonnet, 2003):

- In most systems, DISTINCT will entail a sort or other overhead, so should be avoided.
- Many subsystems handle subqueries inefficiently
- The unnecessary use of temporaries can affect the performance
- Complicated correlation subqueries may execute inefficiently
- Temporaries may also help avoid ORDER BYs and scans when there are many queries with slightly different bind variables
- It is a good idea to express joins conditions on clustering indexes
- Do not use HAVING when WHERE is enough

Description of query tuning process is shown in **Fig. 1**.

**Figure 1** shows the proposed query tuning process. The process is divided into three major stages which are Query Tuning Pre-Processing, Query Tuning Processing and Query Tuning Post-Processing. The description of each stage is discussed below.

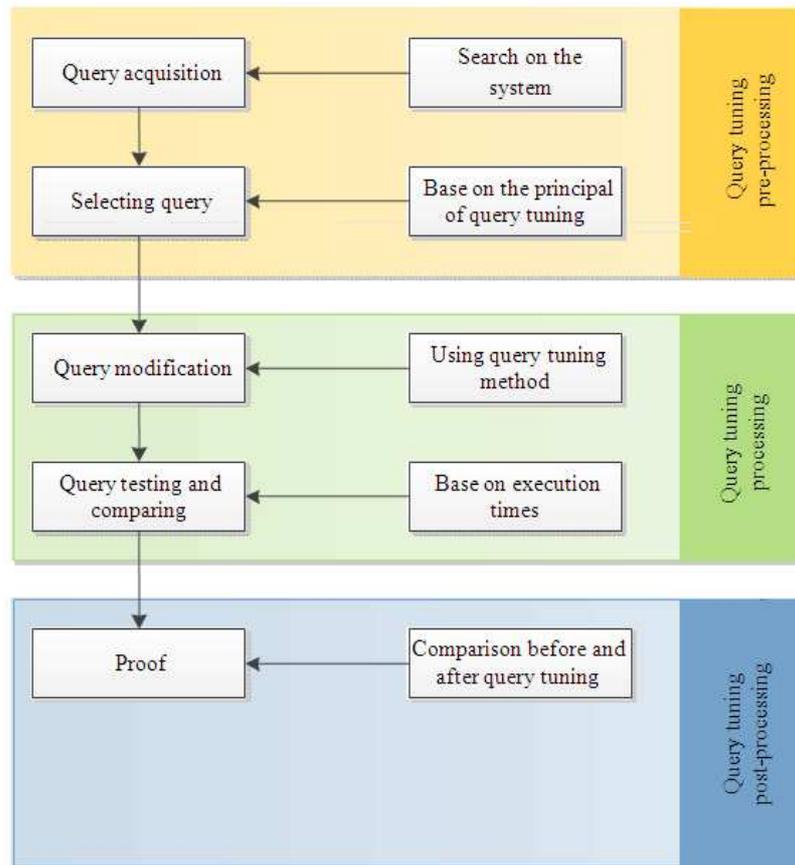


Fig. 1. The process of the implementation of query tuning

### 2.1. Query Tuning Pre-Processing

The first thing to be done at this stage is searching the query in the system. The query is selected that matched the query tuning process. Selection of this query is based on the basic principal of query tuning. The query that met the criteria is processed at the next stage, which is the implementation of query tuning process.

### 2.2. Query Tuning Processing

At this stage, modification is done on the selected query on the Pre-Processing stage. This modification is made based on query tuning techniques. Its structure is modified and adjusted so that the query execution time is faster when accessing the database.

The modified query is tested at this state. This test aims to compare whether the execution time of the modified query is faster than the unmodified query. The tool that is being used to run the tests on the modified

query is Oracle Tuning advisor in Toad version 9.1.0.62. This tool run under the Personal Computer (PC) which has the specs: Operating System Microsoft Windows XP Professional Version 2002 Service Pack 3 (5.1, Build 2600), Processor Intel ® Core™ 2 Duo CPU T5450@1.66GHz (2 CPU) and Memory 1528MB of RAM.

For the query tuning process that done manually, if the results is good, then the new query can be used to make the system run faster. However, if the query execution time is longer than the unmodified query, there will be some techniques performed on the query to get the execution time runs faster than the unmodified one. If the execution time is not increasing as well, then the structure of query is considered reaching its best condition. The next measure to be taken is continuing the query tuning process for another query. After all selected queries from the earlier stage are tuned and tested, the next stage is Post-Processing stage.

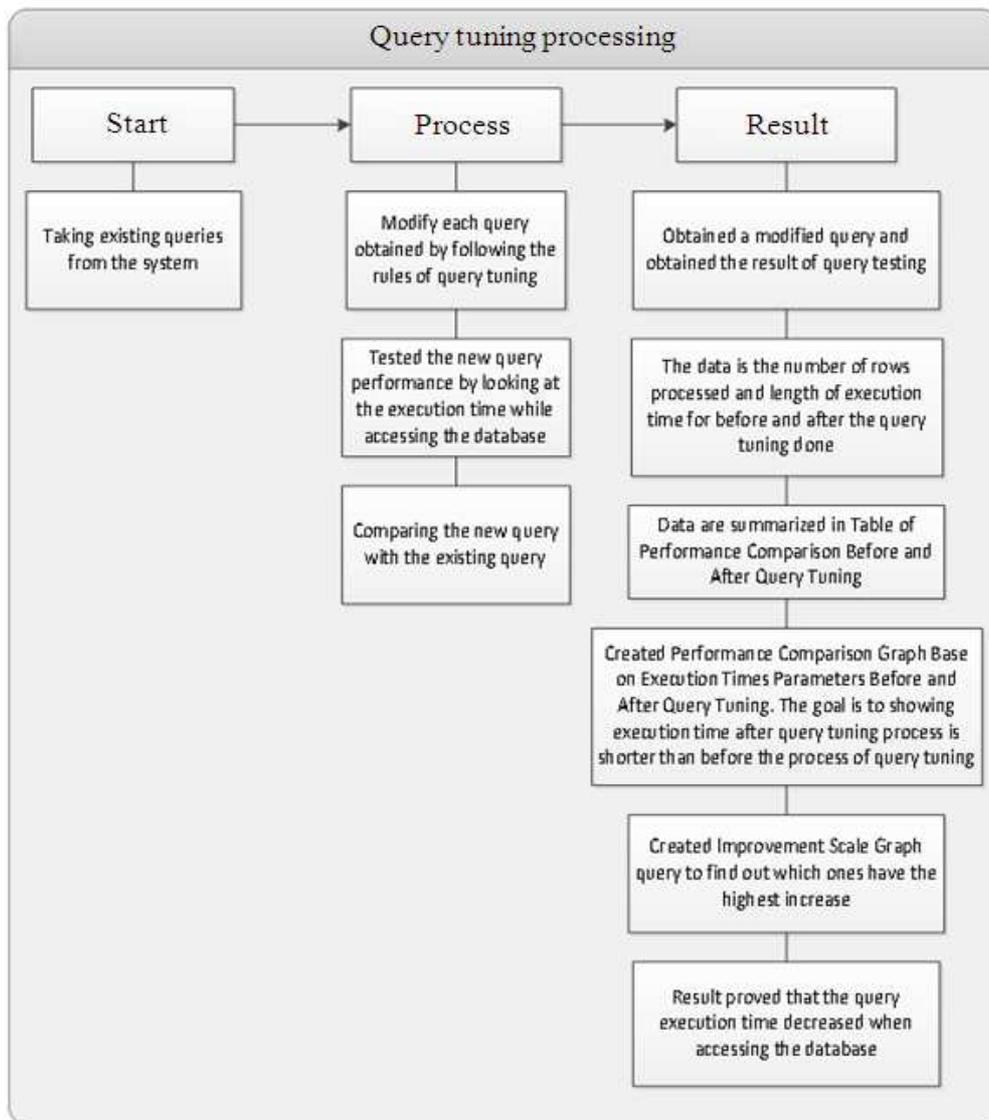


Fig. 2. Query tuning processing

### 2.3. Query Tuning Post-Processing

At this stage, tested queries from the earlier stage are summarized and compared to the unmodified queries. Comparisons are made based on the number of rows processed data and the length of execution time needed before and after the query tuning process WAS done. The results of these comparisons are evidence that the queries tuning process done on the system has managed the execution time run faster when query accessing the database. It means that the query tuning process done on

the system has reduced the query execution time when accessing the database. The detail steps performed on query tuning processing are shown in Fig. 2.

Based on Fig. 2, the query tuning process is divided into three stages. Starting from the “Start” and then going to the stage of “Process” and ended with the stage of “Results”. On the “Start” stage, the existing queries from the earlier stage are processed. At the “Process” stage, the queries are modified based on query tuning rules manually as well as automatically by using tool. The ability of modified query is tested in accessing the

database by viewing the execution time needed. The modified and the unmodified query are compared based on the execution time needed when accessing the database.

The next stage is “Result”. Query scripts and data resulted from the earlier stage are accessed and summarized. The data is in terms of the number of rows processed and the length of execution time needed before and after the query tuning is done. The data are displayed in a table of comparison that shows the time taken before and after query tuning is done. In this study, a comparison chart is made based on the performance of execution time before and after tuning the query. The aim is to show that the query execution time needed after the query tuning process done is shorter than the previous ones. A graph of query improvement scale is created to find out which query has the highest improvement. After the data summarized, the result proves that the query execution time when accessing the database has reduced.

The query tuning process is done on the system manually and automatically by using a tool. For the query tuning process that manually conducted, it is derived from the Form and LOV. The query generated from the Form and LOV is simple and can be done manually. This simple and short query only contains the keyword DISTINCT, HAVING and ORDER BY. For example, to process a query tuning manually, the query is obtained from the Form and LOV, it is simple and can be done manually. A query is called simple and short query when it contains only keyword DISTINCT, HAVING and ORDER BY.

### 3. RESULTS AND DISCUSSION

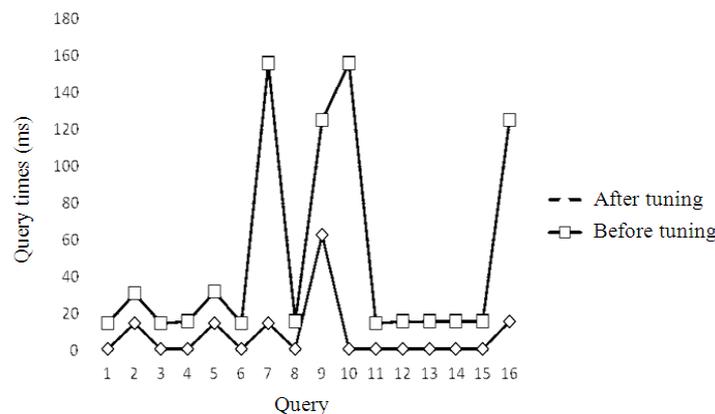
This section will discuss in details about the query tuning process on 39 queries. 16 queries are obtained from the object Form and LOV and the tuning process is

done manually. The other 23 queries are obtained from the View and the tuning process is done automatically by using a tool. First, the script query that will be tuned is displayed and then followed by displaying the query modification processes. The next step is conducting a test on query when accessing the database based on the execution time. The data on the execution time form is displayed in graph and table.

#### 3.1. Performance Comparison of Query Tuning

After the query tuning process was done on the system, the result of query is obtained. The data is the number of rows and the length of execution time taken before and after the query tuning. The comparison of query tuning for the query that generated from Form, LOV and View is shown in **Table 1 and 2** and **Fig. 3 and Fig. 4**.

**Figure 3 and Fig. 4** show the performance comparison graph based on execution time needed before and after the query tuning process done on Form, LOV and View. It can be seen there is a difference between the execution time needed. The execution time after the query tuning process done on the Form, LOV and View is shorter than the previous one. As an example in Query 16, the query taken from Form and LOV shown on the **Fig. 3**, the execution time needed before the query tuning process is 125 milliseconds and after the query tuning process is 16 milliseconds. This shows that the query tuning process is able to reduce the execution time of a query when accessing the database. Other example is shown in query 23, the query that taken from View as shown in **Fig. 4**, the execution time before the query tuning process is 48 seconds and after the query tuning process is 11 seconds. **Figure 5 and 6** for the improvement scale of the query tuning process on Form, LOV and View.



**Fig. 3.** Performance comparison graph base on execution times parameters before and after query tuning on form and LOV

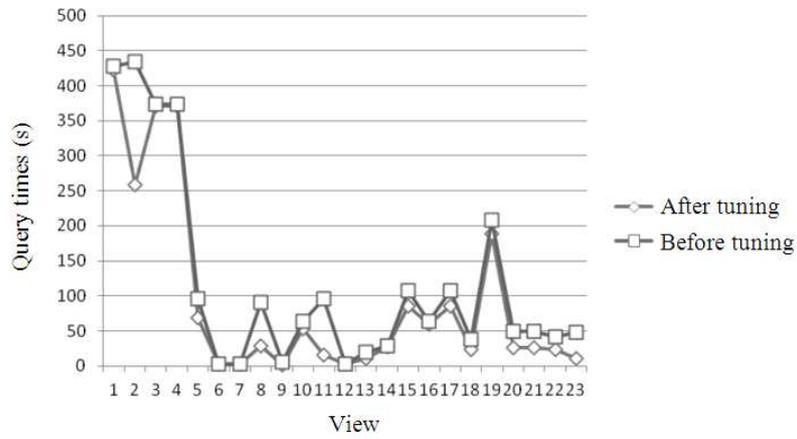


Fig. 4. Performance comparison graph base on execution times parameters before and after query tuning on View

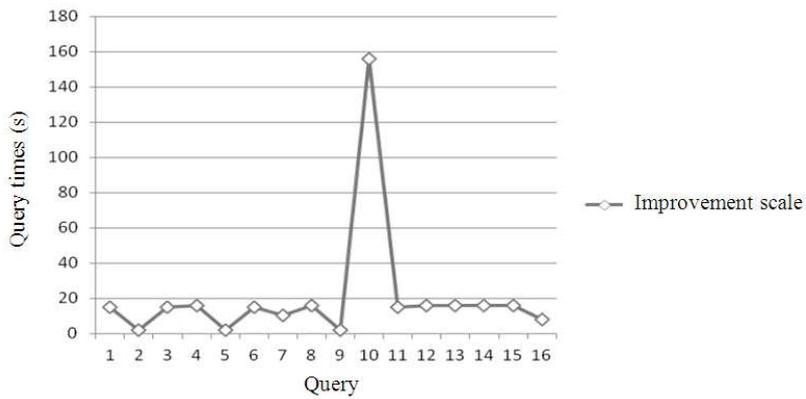


Fig. 5. Improvement scale graph on Form and LOV

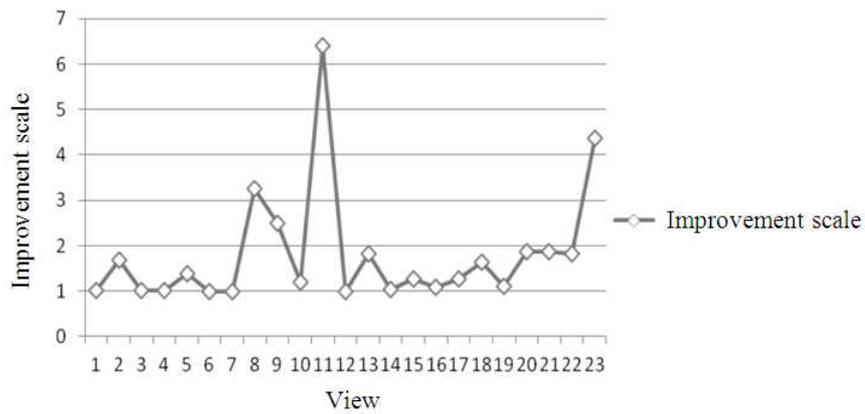


Fig. 6. Improvement scale graph on View

**Table 1.** Performance comparison before and after query tuning on Form and LOV

Query	Rows processed	Time before tuning (ms)	Time after tuning (ms)	Improvement scale
Query 1	1	15	1	15.00
Query 2	2	31	15	2.06
Query 3	3	15	1	15.00
Query 4	99	16	1	16.00
Query 5	500	32	15	2.13
Query 6	500	15	1	15.00
Query 7	500	156	15	10.40
Query 8	64	16	1	16.00
Query 9	217	125	63	1.98
Query 10	500	156	1	156.00
Query 11	4	15	1	15.00
Query 12	21	16	1	16.00
Query 13	125	16	1	16.00
Query 14	1	16	1	16.00
Query 15	1	16	1	16.00
Query 16	500	125	16	7.81

**Table 2.** Performance comparison before and after query tuning on view

Query	Rows processed	Time before tuning (ms)	Time after tuning (ms)	Improvement I scale
View 1	551	428	423	1.01
View 2	551	434	259	1.68
View 3	573	374	371	1.01
View 4	573	374	371	1.01
View 5	3623	96	69	1.39
View 6	3	3	3	1.00
View 7	1	3	3	1.00
View 8	87191	91	28	3.25
View 9	9281	5	2	2.50
View 10	1906	64	53	1.21
View 11	167	96	15	6.40
View 12	1	3	3	1.00
View 13	18	20	11	1.82
View 14	28	28	27	1.04
View 15	16564	108	86	1.26
View 16	1	64	59	1.08
View 17	16564	108	86	1.26
View 18	521	38	23	1.65
View 19	22754	208	189	1.10
View 20	22754	49	26	1.88
View 21	22754	49	26	1.88
View 22	22754	42	23	1.83
View 23	231	48	11	4.36

**Figure 5** shows the improvement of 16 queries on Form and LOV, 23 queries improvement on View as shown in **Fig. 6**. The improvement scale is obtained by comparing the execution time before and after the query tuning applied on Form, LOV and View when accessing the database. The query that has the highest improvement scale value in **Fig. 5** is a query 10 on Form and LOV. Query 10 shows the highest in execution time.

After the query tuning, query 10 is able to improve the execution time by 156 times. Other queries under study also able to improve the execution time by 15 times in average. It shown that the result performed by the proposed query process able to overcome the performance of previous other methods (Elfaki *et al.*, 2011; Kanaan *et al.*, 2008; Haw and Lee, 2008; Khafajeh *et al.*, 2012).

#### 4. CONCLUSION

Nowadays, many organizations use large-scale systems where the number of accessed data growing rapidly. Query operations that modify the data in databases such as insert, update and delete often done during the transaction. This condition slow down the transaction and takes more time to perform the query. In other words, it takes more time on query execution to access the data from database. To solve this problem, the database tuning is needed. Sometimes, there are some organizations needs the system running in real-time. It means a fast processing without any delay is needed. For very large data, it will be difficult to perform the data processing in real-time. Due to this reason, the precision is required in building a system. Query tuning is part of the process of database tuning. Indeed, the actual query tuning is not enough. In this study, enhancement of the query tuning process has managed to improve the system's speed and execution time. It shows that the proposed query tuning process managed to shorten the execution time when accessing the database. Based on this result, the problem stated in the problem statement has been resolved. In order to improve the query process, all the stages in database tuning should be investigated and this becomes the motivation of our study in future.

#### 5. REFERENCES

- Elfaki, M.A., H. Ibrahim, A. Mamat and M. Othman, 2011. Collaborative caching architecture for continuous query in mobile database. *Am. J. Econ. Bus. Admin.*, 3: 33-39. DOI: 10.3844/ajebasp.2011.33.39
- Haw, S.C. and C.S. Lee, 2008. TwigINLAB: A decomposition-matching-merging approach to improving XML query processing. *Am. J. Applied Sci.*, 5: 1199-1205. DOI: 10.3844/ajassp.2008.1199.1205
- Kanaan, G., R.A. Shalab, S. Ghwanmeh and B. Ismail, 2008. Interactive and automatic query expansion: A Comparative study with an application on Arabic. *Am. J. Applied Sci.*, 5: 1433-1436. DOI: 10.3844/ajassp.2008.1433.1436

- Khafajeh, H., A. Abu-Errub, A. Odeh and N. Yousef, 2012. Novel automatic query building algorithm using similarity thesaurus. *Am. J. Applied Sci.*, 9: 1373-1377. DOI: 10.3844/ajassp.2012.1373.1377
- Kimura, H., S. Madden and S.B. Zdonik, 2010. UPI: A primary index for uncertain databases. *Proceedings of the 36th International Conference on Very Large Data Bases, (VLDB' 10)*, pp: 630-637.
- Koh, J.J., 2007. Relational database schema integration by overlay and redundancy elimination methods. *Int. Forum Strategic Technol.* DOI: 10.1109/IFOST.2007.4798673
- Li, Q., X. Honglin and G. Yan, 2009. Research on performance optimization and implementation of oracle database. *Proceedings of the 3rd International Symposium Intelligent Information Technology Application*, Nov. 21-22, IEEE Xplore Press, Nanchang, pp: 520-523. DOI: 10.1109/IITA.2009.7
- Paulson, L., 2004. Open source databases move into the marketplace. *IEEE J.*, 37: 13-15. DOI: 10.1109/MC.2004.62
- Shasha, D.E. and P. Bonnet, 2003. *Database Tuning Principle, Experiments and Troubleshooting Techniques*. 1st Edn., Morgan Kaufmann Publisher, San Francisco, ISBN-10: 1558607536, pp: 415.