Development of Database Structure and Indexing Technique for the Wireless Response System

Alhadi Klaib[1], Joan Lu[2]
Department of Informatics
University of Huddersfield
Huddersfield, UK
{[1]Alhadi.klaib,[2] j.lu}@hud.ac.uk

Abstract—**The relevance of this paper to the topics of the conference is that this paper provides a contribution to the area of systems communications, since this paper participates in the improvement of the Wireless Response Systems (WRS). In more detail, the rapid development of computer and wireless technologies improves many aspects of daily life. The objective of this research is to develop a database for the WRS in order to gain an efficient, fast, and reliable database management system. Furthermore, this research proposes a generic database structure for the Wireless Response System. Moreover, it investigates and studies advanced database indexing techniques and then performs a comparison between them. Subsequently, this work makes an argument to find out the most appropriate indexing technique for the WRS. Consequently, this research has achieved a great deal of success and has met the objectives and aims. To conclude, a framework for the Wireless Response System database has been developed. Additionally, the B+ Tree and Hash indexing techniques have been examined successfully. Thus, it is found that the B+ Tree is a powerful technique for this particular system.**

Keywords-wireless response system; student response system; indexing techniques.

I. INTRODUCTION

Since mobile computing has grown rapidly in recent years, it has been involved in many life aspects such as e-commerce, e-education etc. The Student Response System (SRS) is an example of the e-education aspect. The SRS has improved the learning environment [1]. Moreover, the revolution in smart phone devices, Personal Digital Assistant (PDA) and other software development platforms has provided much support for the SRS. These devices have many advantages such as touch screens and Wi-Fi access which also gives them a great opportunity to be used as clickers [2]. The development of handheld devices has provided a motivation for developing the Wireless Response Systems (WRS) [3][4].

Database Management systems (DBMS) are widely used as a method for manipulating large datasets. Indexing technology is used by these systems to enable fast access to stored data. The index structures are additional files that provide alternative methods to access the data without affecting the physical placement of data on the disk. In fact, large databases need to be indexed to meet performance requirements [5][6].

Objectives and Aims: It is an attempt to enhance the WRS by proposing an appropriate generic architecture for the WRS database. This research concentrates on the development of an evolutionary scheme to support future improvement of the WRS and mobile computing technology in the education field. Subsequently, develop this proposal by using MySQL technology. The research will also investigate the possibility of implementing an appropriate advanced indexing technique for the WRS in order to gain an efficient and reliable system.

The remainder of this paper is structured as follows: Section II discusses related works. Section III defines database structure for the WRS as a case study. Section IV describes the methods and explains the implementation of the design. Section V shows the results and discusses them. Section VI concludes the work and recommends some future work.

II. LITERATURE REVIEW

A. *Student Response System :*

SRS has been used for many years, usually in large classes and lectures to increase the involvement of the student in the learning process. Traditionally, it began as a clicker device used to transfer student responses to the teacher or instructor as a receiver. The SRS has many different names such as clickers, personal response system, audience response system etc. The SRS has shown efficiency as a tool for improving the educational environment [7][8]. The EduMECCA SRS project was a part of the European funded Lifelong Learning Program. The aim of this project is to develop an easy to use SRS for iPod Touch and iPhone and PC. It is also to provide instructors and teachers with a new tool to improve the integration between the students and the teachers [7]. This SRS uses broadly tablet PCs and smart mobile devices, which can be connected to the internet, for collection of student responses [9]. There are some assessments and study cases that show how the SRS has contributed to improvement of the educational environment. For instance, the EduMECCA SRS tool was used by the University of Buckingham as a pilot study in a number of courses. The feedback shows that the WRS has a positive impact on student learning [9]. Another example is that an assessment of the effects of SRS on student learning and

 ISBN: 978-1-61208-310-0

attitudes over a number of courses at the Department of Biology, New Mexico State University shows that the use of SRS in classes had a positive effect on student performance in all biology courses [10].

B. Database Management System for the Wireless Response System

There are number of DBMS available such as:

1) Relational Database Management System: There are many types of Relational Database Management Systems (RDBMS) in the market such as My-SQL, Oracle, MS SQL server, MS Access. Additionally, many well-known firms use My-SQL for their systems, for instance YouTube, social networks, and Wikipedia [1].

2) Object-Relational Database Management System : it is similar to RDBMS. The difference is the Object-Relational Database Management System (ORDBMS) is based on supporting an object oriented database model such as class, objects, and inheritance. The objective of ORDBMS is to connect Entity Relationship with Object-Relational Mapping [3].

3) Object Database Management System: it is also called Object-Oriented Database. The Object Database Management System (ODBMS) is a kind of database that stores objects rather than data such as numbers, integers, and strings [1].

4) XML Database Management System: It is an advanced database technology which considers data as an XML format. The XML Database Management System (XDBMS) has the advantage that data can be queried in many different formats. There are mainly two kinds of XML database, namely XML Enabled and Native XML. XML has a great deal of advantages such as: efficiency, eliminating redundancy, fast, simple, strong, and object-oriented and relational [1][11].

The database technology for the WRS system was considered by the XDIR group at Huddersfield University [1]. It is an important step and it was proposed that using RDBMS is perhaps the most appropriate method [1].

C. Database Indexing Techniques:

The aim of database indexing is to improve the performance of the system. This goal can be achieved by making the retrieval of the data very fast. Therefore, indexing is a vital task to ensure a massive database is efficient and robust [3]. There are many indexing techniques. However, there is no indexing technique that is all-powerful and ideal for all types of queries. As far as classifications of indexing techniques are concerned, there are different kinds of classifications, each of which is based on particular criteria. Nevertheless, generally speaking, there are two kinds of indices, namely ordered indices and hash indices [5][12]. Ordered indices: this sort of index is based on a sorted ordering of data. Hash indices: such index is based on the values being distributed across a range of buckets [12]. Ordered Indices: we use index structure in order to get quick access to data. Each index structure is associated with a specific search key. A file may have a number of indexes [12].

1) Primary Index: in this kind of index we assume that data are sorted sequentially on a search key. These types of indices are developed for applications that request sequential and random access [12]. There are two types of ordered indices as follows. First, dense Indices is seen in every record of the file. The index record has a pointer to the first data record in the file/table/relation that has its search-key value; since it's a clustering index, all other records with that search-key value will be known to follow that record [12]. Second, sparse Indices: A sparse index holds index records only for some of the search-key values that are known. Sparse indices must be clustering – the sparse approach doesn't make sense for non-clustering indices.

2) Secondary Index: it provides a secondary way of accessing data. This data might be ordered, unordered, or hashed. B+ Tree indexes have been broadly used in data heavy systems to ease query retrieval. It is widely accepted due to height balanced tree [13]. In more detail, each path from the root of the tree to a leaf of the tree is the same length. Moreover, this indexing technique is the most widely used since it keeps its efficiency in the insertion and deletion data. B+ Tree indexes are an alternative to indexed-sequential files. The disadvantage of indexed-sequential files is that the performance degrades as the file grows, since many overflow blocks get produced. Periodic reorganization of the entire file is required. Advantage of B+ Tree index files are as follows: it automatically reorganizes itself with small, local, changes, in the face of insertions and deletions. Reorganization of the entire file is not required to maintain performance. The disadvantage of B+ Tree indexes is extra insertion and deletion overhead, space overhead. The advantages of B+–trees outweigh the disadvantages, and they are used extensively. Moreover, B Tree supports access concurrency, which is an important feature of the WRS. This maximizes access concurrency by multiple users [14][15][16].

Hash index: The disadvantage of sequential file indexing techniques is that there is a need to access the index structure to access the data or alternatively use a binary search. The consequence of this technique is more I/O operations. The hash index technique needs no such mechanism [12]. The hash index technique is another type of primary index file organization that is based on hashing, which provides very fast access to records under certain search conditions. This organization is usually called a hash file. The search condition must be an equality condition on a single field, called the hash field. In most cases, the hash field is also a key field of the file, in which case it is called the hash key [17]. The main advantage of hash tables over other table data structures is speed. This advantage is more apparent when the number of entries is large (thousands or more). Hash tables are particularly efficient when the maximum number

 ISBN: 978-1-61208-310-0

of entries can be predicted in advance, so that the bucket array can be allocated once with the optimum size and never resized. If the set of key-value pairs is fixed and known ahead of time (so insertions and deletions are not allowed), one may reduce the average lookup cost by a careful choice of the hash function, bucket table size, and internal data structures. In particular, one may be able to devise a hash function that is collision-free, or even perfect. In this case the keys need not be stored in the table [18].

D. Comparison of B+ Tree and Hashing indexes:

Selecting criteria for suitable comparison is a sophisticated task in many cases. Broadly speaking, there are four methods by which indexing techniques can be compared with respect to these criteria, namely direct argument, mathematical modelling, simulation, and experimentation [19]. Perhaps this research will just use the argument method as the research for selecting the most appropriate technique for the WRS [19][20].

Criteria of Comparison: there are many criteria by which index techniques can be compared, such as space requirement, CPU time, and memory requirement, overall speed etc. [19]. Perhaps we need to consider some of these criteria in case of addition, modification and deletion of records. Having studied the B+ Tree and Hashing indices, the results will be summarised in a couple of tables to show the advantages and disadvantages for each of these techniques. Table number I shows the advantages and disadvantages of the B+ Tree index technique [20][21][22]. Table number II illustrates the advantages and disadvantages of the Hash index technique [23].

TABLE I. ADVANTAGES AND DISADVANTAGES OF B+ TREE INDEX TECHNIQUE.

No	Advantages	Disadvantages
1	It automatically re-organizes itself with small, local, changes when insertions and deletions occur.	Extra insertion and deletion cause space overhead.
2	The overhead is acceptable as the cost of file reorganisation is avoided.	
3	Reorganization of entire file is not required to maintain performance	
4	B+–Tree is an alternative to indexed-sequential file.	
5	MySQL servers support B Tree index	
6	B Tree supports access concurrency	

TABLE II. ADVANTAGES AND DISADVANTAGES OF HASH INDEX TECHNIQUE.

No	Advantages	Disadvantages
1	Hash performance does not degrade with growth of file minimal space overhead	Hash index may be less efficient than B tree for particular string processing applications
2	Hash indexing is efficient when the maximum number of entries can be predicted in advance.	Hash tables may also be used as disk-based data structures and database indices, although B–trees are more popular in these applications
3	Fast for exact-match comparison	Hash is used only for = or <=> comparison
4	Whenever the indexed table is updated, the hash index is updated automatically.	Poor for looking for range of values
5	Search keys can be prefixed if the key can't be used to find rows.	Only whole search keys can be used to search for row, prefix of the key can't be used to find rows.
6		Hash index is stored in the main memory only.

III. CASE STUDY: DATABASE STRUCTURE FOR THE WIRLESS RESPONSE SYSTEM

Background: The WRS is being developed by the XDIR Research Group. This system is based on a wireless network environment by using the most recent smart phones and handheld devices [2]. The aim of the XDIR group is to develop an easy-to-use, efficient and fast system. Thus, this group proposed a prototype based on mobile devices [2]. There were found to be many advantages in this system such as: the system users do not always need an internet connection, and instead Wi-Fi can be sufficient for using the system. Moreover, mobile phone devices are a portable, easy to carry, and cost-effective tool compared with PCs [1].

Responding Relationship: there is a formula that defines the technique of the responding relationships as shown in (1) below [2].

$$SD = \sum qi \leftarrow\rightarrow \sum rj \quad (1)$$

Whereas SD is a subject domain, *qi* means a group of questions that respond to a group of responses in a particular session. *rj* stands for a particular response in a session [2].

Figure 1 shows a corresponding relationship between the questions and responses. As soon as *qi* has been generated by the system, some *rj's* will be hit back to *qi's*. At the same time a number of *rj's* will receive some qi's concurrently. As a result the system has to consider this relationship as many to many [2].

Since the WRS is based on the usage of mobile phone devices, and the communication method is wireless networking, the WRS should thus support a variety of sorts of both hardware and software mobile phones. Figure 1 illustrates the relationships between questions and responses. When a *qi* has been triggered by the system, a number of *rj's* will be hit back to *qi's*. Meanwhile, the number of *rj's* will receive a number of *qi's* at the same time [2].

Specifications and Features of the WRS:

A number of features and specifications need to be considered, which in some way are associated with the WRS database. Perhaps, some of these specifications are generic and will not be considered in this stage. The following specifications are the most important ones for the WRS:

- Concurrency: it is obvious that the nature of the WRS leads to heavy concurrent operations in the session's time.

 ISBN: 978-1-61208-310-0

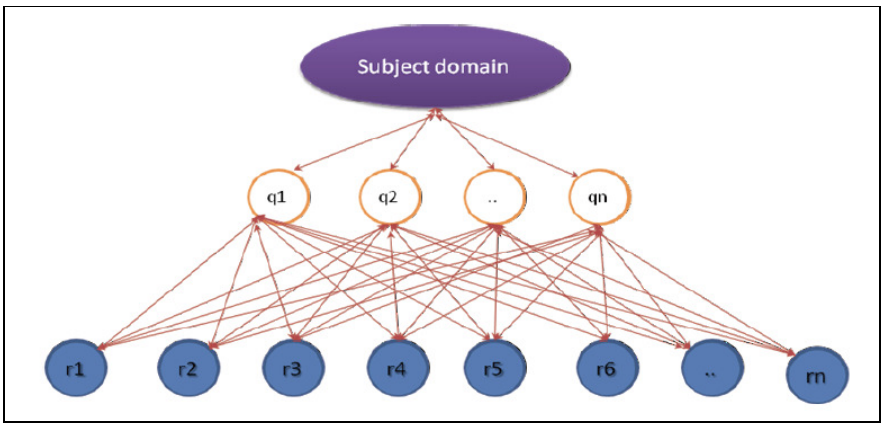

Figure 1. Logical relationship of wireless response system.

- Usability: the WRS is being developed to be simple to use and fast in storing and retrieving data [1].
- Scalability: the volume of the disk space needed for this system is crucial. Furthermore, the growth of the database files is also needs to be worked out.
- Extensibility: the numbers of queries in a certain time and to certain records of the database have to be figured out.
- Data types: the WRS is a sophisticated system that is supposed to support different kinds of data such as video, sound, picture, and text.
- Types of Queries: Different kinds of queries need different indexing techniques. For instance, some indexing techniques are good for exact-match comparisons; however, they are poor for queries that look for a range of values [24]. Therefore different queries scenarios need to be defined.
- Internationalization**:** the WRS is being developed to be an international system and can be used in different languages and different countries.
- Generalization**:** in this stage the WRS should be a generic system, so it can be used for different life aspects and the Mobile Exam System (MES) is an example [25][26][27].

IV. METHODS

A logical diagram was proposed for the WRS database. It is a generic framework for the WRS database. Basically, this framework includes three components, namely, disciplines, users, and results. Then, this framework was implemented using the appropriate technology which is MySQL. It is the most reliable and proper technology to implement this database and its relationships. Secondly, the current indexing techniques were studied and their advantages and disadvantages investigated. Thirdly, these techniques were compared in terms of the specifications and features of the WRS.

A. *Database Design*

It is assumed that the WRS needs a certain database management system in order to match the WRS's specifications. Figure 2 shows the proposed framework for the WRS database. As can be seen, the diagram has three components as follows: first of all, disciplines: this component includes all entities representing subjects and discipline materials. Second, users: this component includes all people that deal with the system such as teachers, students and administrators. Third and finally, results: this is the biggest and most sophisticated component which includes exam sessions and the relevant materials and marking. This design provides a clear and flexible scheme. Thus, it can be reformed in future for other systems.

B. *Implmentation*

The WRS database has been developed using MySQL by the phpMyAdmin**.** This database has sixteen entities (tables) so far. However, the work on it is still on-going and further development and enhancement are being considered. The diagram in Figure 3 provides a visual overview of the WRS database design and the relationships between the tables. Each table includes additional details on the columns (attributes).

V. RESULTS AND DISCUSSION

A. *Results*

An implementation of the database model for the WRS has been developed. This implementation is a generic scheme that can be adopted by other systems in future. The system is being integrated in the WRS by a researcher in the XDIR Group. It is being tested in order to find out any limitations. As far as the indexing technique is concerned, here is the comparison of B+ tree and Hash indexing techniques as follows:

Having considered the advantages and disadvantages for both indexing techniques, as well as the comparison of these techniques in Tables I, II and III, we are able to figure out the following results:

 ISBN: 978-1-61208-310-0

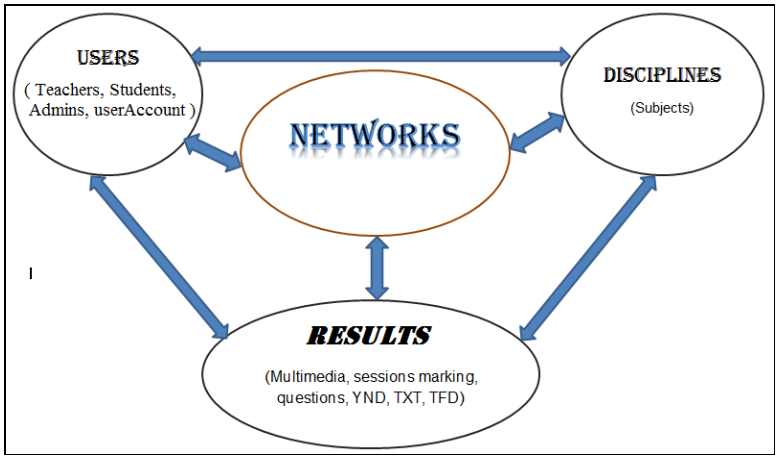

Figure 2. Generic Framework for WRS Database.

- Since WRS is a system that offers flexible and easy-to-use interfaces and functions, flexible and wide queries thus need to be accepted and accomplished. As a result, the B + tree indexing was selected to provide the required flexibility for broad queries.
- The WRS database is being developed by MySQL technology. Hence this technology supports both indexing techniques. B+ Tree is more flexible and suitable for WRS with wide queries support.
- The developed WRS database is a generic system, since the B+ tree indexing technique is intended for general purpose and is supported for different storage engines. Therefore, B+ tree would be the compatible technique for this system.
- In terms of types of queries, the WRS database is a generic and flexible system. Therefore, the indexing that will be used must be efficient for a varying range of queries, which in this case is the B+ Tree.

B. Discussion

The efficiency of WRS is crucial as this system needs to be fast and reliable in order to cope with its purposes such as MES. As a consequence, this research investigated the most powerful indexing techniques in order to achieve an efficient DBMS. Subsequently, a comparison among these techniques has been accomplished. As a result, this research proposes to use an indexing technique in order to achieve an efficient database system. The B+ Tree indexing technique is proposed to be the technique for the WRS as this is the most suitable one for this system.

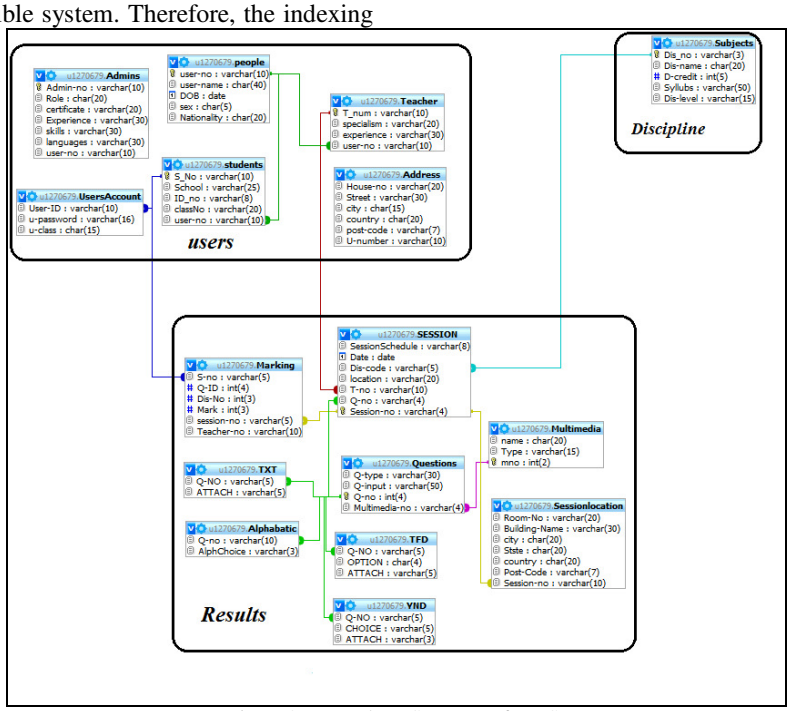

Figure 3. Database Structure of WRS.

TABLE III. A COMPARISON OF B+ TREE AND HASH INDEXING.

FEATURE OF WRS	B+ tree	Hash	Remarks
Keys of search and types of query.	Efficient for range of values (e.g., prefix)	Not efficient for range of values	B+ Tree preferred
Reorganization of data	It is updated automatically	It is updated automatically	Both are acceptable
Range of values	Supports queries for a wide range of values	Inefficient for range of values	B+ Tree preferred
Storage location		Only in main memory	B+ Tree preferred
ACCESS CONCURRENCY	SUPPORTS CONCURRENCY	Supports concurrency	BOTH ARE ACCEPTABLE

VI. CONCLUSION AND FUTURE WORK

A. *Conclusion*

This research proposes an indexing technique for the WRS database, in order to achieve fast and reliable system. It is found that the B+ Tree is a powerful technique in terms of the WRS specification, and it is a great deal to use it. Moreover, it has been developed and implemented a generic architecture for the WRS database that can be adapted by other relevant systems in future. In brief, this research has accomplished a solid base for building an efficient and reliable database system for the WRS.

B. *Future Work*

Further research and investigation need to be carried out. Furthermore, it is recommended to be in the following areas:

- Test the WRS database performance in order to confirm the indexing technique.
- Investigate the possibility of implementing XML technology in the WRS database in order to make it secure and reliable.

ACKNOWLEDGMENT

The authors thank Mr. **Wei Guo** and others from the XDIR Group at Huddersfield University for the help and discussion that they provided during this study.

REFERENCES

[1] J. Lu, A. Sundaram, and V. Arumugam, "Innovative Evaluation System – IESM: An Architecture for the Database Management System for Mobile Application," The 2011 International Conference on Internet Computing, Las Vegas, USA, July 2011.

[2] J. Lu, Z. Meng, G. Lu, and J. B. Stav, "A New Approach in Improving Operational Efficiency of Wireless Response System," IEEE 10th International Conference on Computer and Information Technology (CIT), 2010, pp. 2676-2683, ISBN 9781424475476.

[3] J. Lu, A. Sundaram, M. Zhaozong, V. Arumugam, and L. Gehao, "Mobile Exam System – MES: Architecture for Database Management," The International Conference on Education & Learning in Mobile Age - CELMA, Lake District, UK, June 2011.

[4] P. Pocatilu, "Building Database-Powered Mobile Applications," Informatica Economica, vol. 16, 2012, pp. 132-142.

[5] C. Zaniolo, Advanced database systems. San Francisco, CA: Morgan Kaufmann, 1997.

[6] T. M. Connolly and C. E. Begg, Database systems: a practical approach to design, implementation, and management. Boston, Mass: Addison-Wesley, 2010.

[7] F. o. T. Sør-Trøndelag University College. (2013). EduMECCA. Available: http://histproject.no/node/18 [retrieved: Aug. 2013]

[8] F. o. T. Sør-Trøndelag University College. (2013). What is a Student Response System (SRS). Available: http://histproject.no/node/77 [retrieved: Aug. 2013]

[9] H. Sellahewa, "Enhancing small group teaching and learning using online student response systems," Higher Wducation Academy Higher Wducation Academy, 2012.

[10] R. W. Preszler, A. Dawe, C. B. Shuster, and M. Shuster, "Assessment of the Effects of Student Response Systems on Student Learning and Attitudes over a Broad Range of Biology Courses," CBE—Life Sciences Education, vol. 6, Spring 2007, pp. 29–41.

[11] M. J. Young, Step by step XML. Redmond, Wash: Microsoft, 2000.

[12] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database system concepts. New York: McGraw-Hill, 1997.

[13] E. Bertino, et al., Indexing techniques for advanced database systems (Advances in Database Systems), 1997.

[14] P. Lehman and s. Yao, "Efficient locking for concurrent operations on B-trees," ACM Transactions on Database Systems (TODS), vol. 6, Dec. 1981, pp. 650-670.

[15] H. Umesh, M. Agarwal, P. Rastogi, S. Agrawal, S. Mittal, S. Jain, V. L. R. Madhurima, and D. S. Kushwaha, "A new novel indexing technique using a bit-tuple tree," Recent Trends in Information, Telecommunication and Computing(ITC), Mar. 2010, pp. 224-226.

[16] t. f. e. Wikipedia. (2013). B-Tree. Available: https://en.wikipedia.org/wiki/B-tree#Access_concurrency [retrieved: Aug. 2013]

[17] R. Elmasri and S. Navathe, Fundamentals of database systems. Boston, Mass: Pearson Addison-Wesley, 2007.

[18] T. M. Connolly and C. E. Begg. Database solutions: a step-by-step guide to building databases. New York: Pearson Addison Wesley; 2004.

[19] J. Zobel, A. Moffat, and K. Ramamohanarao, "Guidelines for presentation and comparison of indexing techniques," ACM SIGMOD Record, vol. 25, Sept. 1996, pp. 10-15.

[20] MySQL, Comparison of B-Tree and Hash Indexes, 2013 Available: http://dev.mysql.com/doc/refman/5.5/en/index-btree-hash.html [retrieved: Aug. 2013]

[21] K. Yat-Sang and D. Wood, "A New Method for Concurrency in B-Trees," IEEE Transactions on Software Engineering, vol. SE-8, May 1982, pp. 211-222.

[22] MySQL. MySQL Customers [Online]. Available: http://www.mysql.com/customers/ [retrieved: Aug. 2013]

[23] Wikipedia tfe. Hash table 2013 Available from: http://en.wikipedia.org/wiki/Hash_table#Performance_analysis. [retrieved: Aug. 2013]

[24] M. Anca, P. Cerasela, and E. Cristi, "Indexing Strategies For Optimizing Queries On MySQL," Annals of the University of Petrosani : Economics, 2010, pp. 201-214.

 ISBN: 978-1-61208-310-0

[25] T. Haapasalo, I. Jaluta, S. Sippu, and E. Soisalon-Soininen, “Concurrency control and recovery for multiversion database structures,” 2008, pp. 73-80.

[26] I. Jaluta, S. Sippu, and E. Soisalon-Soininen, “Concurrency control and recovery for balanced B-link trees,” The VLDB Journal, vol. 14, April 2005, pp. 257-277.

[27] I. Jaluta, S. Sippu, and E. Soisalon-Soininen, “B-tree concurrency control and recovery in page-server database systems,” ACM Transactions on Database Systems (TODS), vol. 31, Mar. 2006, pp. 82-132.

 ISBN: 978-1-61208-310-0