

Database Tuning and Its Role In Information Technology Education

Alexander P. Pons

Computer Information Systems
University of Miami
Coral Gables, Florida 33146, USA
apons@exchange.sba.miami.edu

ABSTRACT

Course curriculum in database management systems encompasses many topics, from data modeling to implementation and testing. These topics establish a strong foundation for Information Technology students to analyze, design, and implement a database system. Beyond these basic skills, Information Technology students must be equipped with the necessary capabilities to address database performance issues, such as when end-user expectations are not met. The topic of database tuning includes various techniques associated with enhancing database performance, and is important in providing a well-rounded database curriculum, but this skill is not generally covered in most database management systems courses. Our study focuses on testing the researcher's assumption that database tuning is not part of current Information Technology curriculums. In addition, its significance and importance in the curriculums is researched. We surveyed academics and professionals to quantify the importance of database tuning and establish an understanding of its role within the education of students taking database related courses.

Keywords: Database Tuning, CIS curriculum, Database education

1. INTRODUCTION

As with most complex subjects, the definition of database tuning varies across the variety of available informative resources on the subject. Instead of trying to identify an all-encompassing definition, we will simply define database tuning as an ongoing process of achieving maximum performance from all components of a database system. Unfortunately, a database management system (DBMS, or alternatively, RDBMS) does not run in isolation; therefore systems other than the database engine can influence the performance of the DBMS. These include the operating system running the database, applications that interface with the database, and the network over which the database system and applications communicate. An oft-confronted reality of the database developer is end-user discontent with the performance of the database application. These applications may be functionally correct, but poor response time or performance frustrates end-users, creating a sense of dissatisfaction that may result in the abandonment of the application (Nielsen, 1999).

The concept of database tuning is critical to the success of most database-centric applications. An organization that establishes a database application with the aim of reducing errors, improving consistency, and increasing employee effectiveness must first convince its employees of the

application's benefits. Slow performance can negatively impact employee perceptions, which could lead to a resistance in using the applications, despite the application's correct and enhanced functionality. In addition, when customers access an application, their overall impressions of the organization can be influenced by the ease and speed associated with the application. For example, an organization that offers bill payment and claim service through the Web must address not only the ease of navigating their Web pages, but also the speed at which transactions occur, since both can influence the overall level of customer satisfaction with the organization. Therefore, database applications must function not only correctly but also quickly to meet user expectations.

The importance that applications meet user expectations is well recognized (Hoxmeier and DiCesare, 2000) and Information Technology (IT) students should be prepared to solve performance issues associated with end-user application dissatisfaction. IT students seem to be well accomplished in database analysis, design, and implementation of correctly functioning systems. But database development only constitutes an operational system. When performance issues arise, these IT students must know how to address the system's performance and response time deficiencies. Our aims are to determine the importance and significance of database tuning to database professionals and academics, to understand to what extent

database tuning is covered in undergraduate and graduate education, and to evaluate the theoretical components of database tuning as part of a university's IT curriculum. These are students pursuing a major from a variety of departments such as an Information System (CIS) department.

The remainder of this paper is organized as follows: section 2 covers some basic concepts in database tuning; section 3 reviews the current methodology and approach to database tuning; section 4 discusses database tuning as a topic in a traditional university education and includes faculty and professional surveys that assess the significance of the topic; section 5 highlights the survey results and evaluates the merits of incorporating database tuning into undergraduate and graduate IT curriculums; section 6 explains the limitations of this study and proposes guidelines for its continuation; finally, 7 provides a conclusion and suggests directions for future research

2. UNDERSTANDING DATABASE TUNING

The concept of database tuning involves many factors beyond the DBMS, but understanding what these factors are permits an assessment of its integration in database management courses. First, let's indicate that operational inadequacy and degradation most often come from performance bottlenecks. A performance bottleneck occurs when a database component is assigned an excessive amount of work at a given time, which it is incapable of satisfying adequately. Also, it is possible that these bottlenecks can extend beyond the DBMS and encompass other external components, such as the client application, network communication, operating systems, or some combination thereof. For this reason, a successful database administrator must understand each of these components and how they interact. A sufficient understanding of this architecture will assist the database administrator in isolating a performance bottleneck and then identifying its causes and candidate solutions. We list some of these components in order to provide a better understanding of the tasks facing IT graduates during database tuning (Vaidyanatha, 2001; Niemiec and Lane, 2002; McGehee, 2003a,b; SQL-Server-Performance.com, 2003; Mohamed, 2002; and Quest Software, 2003).

Disk Input/Output (I/O) Bottlenecks – i/o operations require read/write disk drive heads to physically move across the drive platters, potentially incurring a significant time penalty in the process. The amount of time each i/o operation requires depends mostly on the disk's revolutions per minute (rpm) and whether the operation is sequential or non-sequential. In non-sequential drive reads, the drive heads must scan the disk to look for the next logically sequential block, increasing the total i/o operation time.

Central Processing Unit (CPU) Bottlenecks – CPU bottlenecks occur when too many resources compete for computer processing time at once. In this case, the

database administrator may consider a combined solution of adding physical resources as well as system restructuring to alleviate the bottleneck.

Random Access Memory (RAM) Bottlenecks – RAM, like the CPU, is a physical resource of the server itself, so any other processes running on the server will take away from the amount of RAM available to the database system. In addition, if the server running the DBMS system is also being used as a network domain controller, more RAM will be taken away from the database system for that process. Therefore, the database server should be a member server of the network and not acting as a primary or a backup domain controller system.

Network Bottleneck – A network bottleneck is related to traffic problems on the network, but also to the way that the database server behaves within the network. A network administrator should be consulted to look at how to address poor database performance due to network traffic. They might for instance, locate the database server on a different network segment or change the network protocols for token handling, depending on the network topology.

Application Decomposition – Perhaps the greatest stress on the network is a poorly structured client/server application. Applications that run over networks should utilize the client/server system architecture, in which server-side processing minimizes network traffic. Database objects such as views, stored procedures, and triggers allow client applications to issue simple requests of the database that travels over the network using a minimal amount of bits. The database server then receives, processes, and sends back over the network the smallest amount of data necessary to satisfy the client's request. This is called the thin-client/fat-server architecture.

SQL – SQL (Structured Query Language) is a declarative language, which hides many of the details concerning the manner that operations are performed. For example, to retrieve a set of records, a given criteria is stated and the database engine will interpret and executes the operation to obtain a result set. Unfortunately, this can cause performance problems since a SQL statement's execution plan can vary depending on how the statement is coded. The way that a SQL statement is written will almost certainly affect the manner that hidden implementation details are performed. Poorly constructed SQL is a major cause of database system performance degradation, and usually is one of the first factors addressed in performance tuning. There are countless examples of poorly structured SQL (Mohamed, 2002), especially when multiple joins and sub-queries are used. The good news is that most DBMS vendors provide tools to analyze queries and recommend changes based on what the DBMS knows about the databases' logical and physical structures and indexing schemes. These tools, as well as careful evaluation of database objects that contain SQL (like views, stored

procedures and triggers) can drastically alleviate performance problems caused by poorly constructed SQL.

Denormalization – This is a technique to move from higher normal forms of database modeling to lower ones in order to speed up database access. Denormalization is applied during the process of deriving a physical data model from a logical form. Often times, tables must be denormalized (combined together) in order to minimize the number of joins required to extract the desired results. This obviously comes with a price of data duplication, which leads to data inconsistencies and inconsistent enforcement of integrity constraints. An issue that is difficult to answer is how far should table combinations be performed, in other words, when does the process stop yielding the required performance enhancement.

Indexes – Indexes may provide the most difficult challenge, as their haphazard use can harm as much as help database performance. The purpose of an index is to enhance the performance of select statements against a table. Since indexes must be modified to reflect table changes, their use incurs a certain amount of time overhead. A high degree of scrutiny is often required when using indexes, since tables that are searched frequently are generally involved in modification and insertion activities. The definitions of indexes on tables for the purpose of assisting performance will typically slow down the database system as inserts and updates are performed. For these cases, the database administrator must continuously monitor DBMS performance statistics to re-evaluate the creation and deletion of indexes.

Because these factors pervade the computing field they represent an important facet of the education of IT professionals. The problem most often encountered is the lack of a comprehensive and formal methodology for conducting database tuning. In fact, according to the 2001 U.S. New & World Report's top ten graduate Information Systems programs, database tuning didn't appear in any of other their course curriculums. When actually present in a database course, these topics are typically discussed in isolation, without any specific strategy.

3. DATABASE TUNING METHODOLOGY

The factors mentioned can adversely affect database systems performance, irrespective of the DBMS vendor. Thus, the underlying principle is that these factors must and should be considered to improve performance, not individually, but in their totality. This is the concept advocated by Oracle Corporation, called Total Performance Management (TPM) (Shallahamer, 1995). This project approaches database tuning from a holistic standpoint, in order to move database tuning from a reactive, micro-level technique that addresses problems as they arise, to a proactive, macro-level technique that detects and prevents degradation of performance. The TPM method is described as a "living holistic methodology," which is "neither a top-down nor bottom-up approach." The phrase "living holistic methodology" is

explained as follows; "It is holistic since problems are attacked from multiple fronts. It is a method because it provides a structured approach to synthesize the use of tools and techniques in a dynamic computing environment. It is living as its programs evolve to meet changing requirements and are usually run very selectively and purposely – not routinely" (Shallahamer, 1995). The methodology considers the database as an entity that grows in size and complexity over time; therefore, the importance of managing performance grows similarly. The methodology continuously cycles through three stages, the Audit Stage, the Tuning Stage and the Performance Management Stage.

This is just one example of several methodologies advocated by DBMS vendors, database administrators and other IT professionals. With all of the information available, it seems that a successful methodology for database tuning needs to be customized for each database, taking into consideration the DBMS, network topology, application interfaces, physical architecture, system objectives, and numerous other factors. An established methodology must be re-evaluated and changed as the system is modified as well. Factors like DBMS version, upgrades and the number of overall users severely impact the performance of a database system, making the process of database tuning imperative.

Along these lines, Microsoft Corporation advocates integrating performance monitoring and DBMS configuration into their SQL Server system, although this only addresses a part of the overall problem. Microsoft (Miller and Lau, 2001), repeatedly emphasizes that the built-in management tools offered by the newer versions of the DBMS do most of the database tuning processes, even for data warehouse, more effectively. They state: "The goal of SQL Server is to make manual configuration and tuning of a database server an obsolete and archaic practice" (Miller and Lau, 2001). While it is still possible to manually configure options, it is recommended that database administrators refrain from doing so and instead allow SQL Server to automatically configure and tune itself. SQL Server 7.0 has been successful in making automatic adjustments to improve performance and its successor SQL Server 2000 has significantly improved on these self-adjusting techniques. While the degree to which the database can self-tune is probably debatable, these facts certainly indicate that the degree and manner in which earlier versions of SQL Server were tuned will change with upgrades to 7.0, and then again to SQL 2000. However, as mentioned before, this addresses only a few factors in the overall performance tuning problem. Although it will enhance the process, a holistic viewpoint would produce more effective long-term solutions and discourage ad-hoc patchwork. The book (Shasha and Bonnet, 2002) supports a broad perspective on database tuning independent of any particular DBMS vendor, which presents a paradigm suitable for teaching the salient concepts of database tuning. Such an approach to the topic of database tuning would give IT students an extensive understanding of the

main issues required to solve most performance related problems.

4. DATABASE TUNING AS A TOPIC IN TRADITIONAL UNIVERSITY EDUCATION

Coursework and internships are sufficient to prepare graduating IT students to develop and modify database applications for their future employers. Unfortunately, these IT curriculums and internships do not prepare graduates to solve the widespread problem of poor database performance, which usually has no ready explanation. Since they cannot draw from their academic or internship experiences for guidance, they are left to solve a problem about which they possess little knowledge. This no doubt happens frequently, since the term “database tuning” is barely mentioned in database textbooks, and even less in course lectures of the universities in our study. It seems likely then that this topic is not covered in the vast majority of IT curriculums. With so little direct mention of tuning in the curriculum, we might conclude that database tuning is performed through an ad-hoc approach lacking any formal methodology or theoretical foundation suitable for course lecture. Another conclusion might be that a wide spectrum of computer topics are necessary in order to understand tuning, thus making this topic prohibitively broad to study. A more basic concept, possibly aggravating the previous ones, is the thin line that often divides traditional university curriculums from vocational technical education. Topics such as database tuning have customarily been associated with specific software vendors and not regarded as intellectual disciplines that have a place in university education. The role of traditional university education is not to provide job training to its students, but to prepare them intellectually by expanding their knowledge and ability to learn. In describing its core values, the American Council on Education (ACE), an umbrella organization for American Colleges and Universities, states that it “values inclusiveness and diversity, recognizes higher education's responsibility to society, and embraces the belief that widespread access to excellent postsecondary educational opportunities is the cornerstone of a democratic society” (American Council on Education, 2003a). At the same time, the ACE admits that schools will work towards “preserving an appropriate degree of academic breadth as they identify and market their unique niche. Some colleges and universities will continue to provide a traditional, comprehensive program of undergraduate and graduate studies. Others will choose to sharpen the distinctiveness and focus of their offerings, with the conviction that specificity of mission will be a source of strength in meeting the needs of a diverse community of learners” (American Council on Education, 2003b).

4.1 Database Faculty and Professional Survey

It is clear that there are many factors involved in database tuning, and that there exists an underlying theoretical component independent of any particular software vendor. Therefore, in order to understand why the topic of database

tuning is not more widespread in database courses, we conducted a survey to gauge its perceived importance among database experts. To learn more about how university faculty and database professionals view the role of database tuning as a subject in traditional university education, a group of each was surveyed. Figures 1 and 2 present the instrument utilized in the study, with the survey questions asked of database faculty and professional, respectively.

<p>Database class is taught to (check one): <input type="checkbox"/> Undergraduates <input type="checkbox"/> Graduates <input type="checkbox"/> Both</p> <p>Database Tuning is taught (check one): <input type="checkbox"/> As an entire course <input type="checkbox"/> As one of many topics in a database course (Indicate % of total class time spent on database tuning: _____) <input type="checkbox"/> Not at all</p> <p>If “Not at All”, please continue:</p> <p>I would like to see Database Tuning taught as: <input type="checkbox"/> As an entire course <input type="checkbox"/> As one of many topics in a database course <input type="checkbox"/> Not at all</p> <p>I would like to see Database Tuning taught to: <input type="checkbox"/> Undergraduates <input type="checkbox"/> Graduates <input type="checkbox"/> Both <input type="checkbox"/> Neither</p>
--

Figure 1. Survey: University Database Faculty

<p>Information Systems Related Degrees Attained (check one): <input type="checkbox"/> PhD <input type="checkbox"/> M.S. <input type="checkbox"/> MBA <input type="checkbox"/> BS/BA</p> <p>Specific Area of Study (check one): <input type="checkbox"/> Computer Science <input type="checkbox"/> Computer/Management Information Systems <input type="checkbox"/> Engineering <input type="checkbox"/> Other</p> <p>How Important is knowledge of Database Tuning practices in your job: <input type="checkbox"/> Very <input type="checkbox"/> Somewhat <input type="checkbox"/> Not Very</p> <p>Did your academic coursework include Database Tuning education? <input type="checkbox"/> Yes <input type="checkbox"/> No</p> <p>If Yes, was it helpful in your success as a professional?: <input type="checkbox"/> Very much <input type="checkbox"/> Somewhat <input type="checkbox"/> Not at all</p> <p>If No, would it have been helpful to cover the topic in school? <input type="checkbox"/> Yes <input type="checkbox"/> No</p>
--

Figure 2. Survey: Database Professionals

5. RESULT DISCUSSION

The survey was emailed through several listservs and targeted database academics and professional database developers, administrators, and designers. The survey produced 42 faculty and 24 professional respondents, which indicates some interesting points on the perception of database tuning in academia and among database practitioners. The results obtained from the survey are tabulated in Tables 1 and 2, indicating the faculty and professional questionnaires percent responses, respectively.

According to the faculty survey, undergraduate database courses are more prevalent than graduate courses. In fact, 57 percent are taught at the undergraduate level while 12 percent are taught at the graduate level. In 29 percent of the cases it was taught at both levels. Looking at all programs, database tuning is not covered at all in 50 percent of the courses. In 40 percent of the courses database tuning is covered as one of several topics. Therefore, in half of all database classes taught, database tuning is not even mentioned. Of the respondents who indicated that database tuning is taught as a part of an overall course, only one respondent indicated how much time was devoted to database tuning (4 percent). According to the respondents that do not teach database tuning, 75 percent believe it should be covered in some capacity and mostly to undergraduates (over 80 percent).

The data leads the researchers to conclude that currently there is not a great emphasis on teaching database tuning as a topic in the traditional university curriculum. This is evident when we pair the faculty responses with the 90 percent of practitioner's responses which indicate that database tuning was not covered at all throughout their coursework, regardless of whether they had studied Computer Science, Computer/Management Information Systems, or some other computer discipline. Despite our conclusions, over 80 percent of faculty members, who do not teach the topic at all, indicated that they would like to see it covered as a part of the course they currently teach. In addition, of the 90 percent of professionals that indicated a complete lack of database tuning educational background, 75 percent specified that the topic would have been helpful in their university curriculum.

An interesting result is that 75 percent of professionals surveyed believe that knowledge of database tuning is not very important in their current job, and only 25 percent indicated that it is somewhat important. Of the ones that did cover it in school, 33 percent believed that it was somewhat useful and 66 percent consider that it was not very useful. A remarkable paradox exists in the professional responses in that they generally believe that database tuning does not play a major role in their professional careers, but would have liked the topic to be part of their academic experience.

The inconsistency requires further research, but initial

analysis indicates two possible reasons. First, it could be attributed to a general lack of knowledge and understanding of the topic. It might also be based on the rigors of their particular endeavors during their professional careers, such as the demands and performance expectations of the projects they were associated with during their employment.

Database class is taught to:
Undergraduates (57%) Graduates (12%) Both (29%)
Database Tuning is taught as:
As an entire course (9.5%) As one of many topics in a database course (40%) Not at all (50%)
If "Not at All":
I Would like to see Database Tuning taught as:
As an entire course (5%) As one of many topics in a database course (70%) Not at all (25%)
I Would like to see Database Tuning taught to:
Undergraduates (81%) Graduates (4.7%) Both (14%)

Table 1. University Database Faculty Survey Results

Information Systems Related Degrees Attained:
PhD (50%) M.S. (33%) MBA (12.5%) BS/BA (4%)
Specific Area of Study:
Computer Science (37.5%) CIS/MIS (37.5%) Engineering (0%) Other (25%)
How Important is knowledge of Database Tuning practices in your job:
Very (0%) Somewhat (25%) Not Very (75%)
Did your academic coursework include database-tuning education?
Yes (10%) No (90%)
If Yes, was it helpful in your success as a professional?
Very much (0%) Somewhat (33%) Not at all (66%)
If No, would it have been helpful to cover the topic in school?
Yes (75%) No (25%)

Table 2. Database Professional Survey Results

6. LIMITATIONS AND FUTURE WORK

Our initial exploratory study was aimed at determining the perspective of database academics and professionals regarding database tuning. The use of listservs yielded a limited number of both academic and practitioner respondents. In addition, the survey relied on the respondent categorizing themselves as database faculty and professionals, which could have adversely affected the results. This was done in an attempt to maximize the number of respondents in the study.

From our current study we have gained a significant understanding of how database tuning is viewed by both the academic and professional practitioners. There are various aspects of our study that we would like to further explore. These include establishing a more focused group of respondents that we have identified as being database faculty and professionals and answering some more pointed questions. These questions require further investigation, such as: Why is it that database faculty do not cover database tuning in their courses? Why is it that database practitioners believe that database tuning is important, but not a major issue in their career? What would be the best approach to covering database tuning in IT curriculums? and How can database professionals be made to value the importance of database tuning? Our future work will consist of these two aspects in order to learn more about the viewpoint and thoughts in teaching and utilizing database tuning. In addition, we will investigate the discrepancy of practitioner's responses in their view of database tuning and to develop software vendor independent case studies that incorporate a methodology suitable for an advanced database course.

7. CONCLUSION

The objective of our work was to assess the importance of teaching database tuning as part of a university's computer information systems curriculum. It seems that some institutions cover certain factors of database tuning in their undergraduate courses, but lack a comprehensive coverage of the topic. Although there are many factors involved in teaching database tuning, it can and should be taught at the micro-level, but more importantly at the macro-level, using a general and total strategy. Database tuning has a theoretical foundation, which can be a significant portion of a course if not developed as a complete course, without specifically adhering or focusing on any particular DBMS vendor package. However, this does not preclude the need of hands-on practice with a DBMS.

Database tuning, like Entity-Relationship Modeling and SQL programming exists independent of the platform on which it will be used. Consequently, there is occasion to cover the topic in a platform-independent fashion, while enabling the academic institution to avoid the perception of itself as a training ground for a specific vendor package. All indications are that database tuning is a critical part of running a successful database system, and that it's

appropriate as a topic in traditional university education. To answer our initial study objectives: Database tuning is important in the careers of database application professionals, despite the fact that they believe it is not a major factor. They would like to have been exposed to it during their academic experience, but as we have found, the topic is most often ignored and scarcely covered in undergraduate and graduate programs. The existence of a substantial theoretical foundation provides an ample reason to incorporate database tuning in university IT curriculums.

8. REFERENCES

- American Council on Education (2003a), "Connection to the Future Core Values what we stand for." <http://www.acenet.edu/plan/core-values.cfm>.
- American Council on Education (2003b), "Connection to the Future Challenges of a new century." <http://www.acenet.edu/plan/challenges.cfm>.
- Hoxmeier, J.A. & DiCesare, C. (2000), "System response time and user satisfaction: An experimental study of browser-based applications," Proceedings of the Association of Information Systems Americas Conference, Long Beach, California.
- McGehee, B. (2003a), "Best SQL Server Performance Tuning Tips." <http://www.sql-server-performance.com>
- McGehee, B. (2003b), "Tips on Optimizing SQL Server Indexes." <http://www.sql-server-performance.com>
- Miller, J. and H. Lau (2001), Microsoft SQL Server 2000 RDBMS Performance Tuning Guide for Data Warehousing. Microsoft White Paper. <http://www.microsoft.com>
- Mohamed, A. (2002), "A Successful Performance Tuning Methodology." <http://www.quest.com/whitepapers/asuccessfulperformancetuningmethodologyfinal.pdf>
- Nielsen, J. (1999), "User interface directions for the web," *Communications of the ACM*, 42 (1), 65-72.
- Niemiec, R. and R. Lane (2002), Oracle 9i Performance Tuning Tips & Techniques. McGraw-Hill Companies.
- Quest Software (2003), White Papers. <http://www.quest.com>
- Shallahamer, C. (1995), "Total Performance Management." Oracle Magazine May/June and July/August.
- SQL-Server-Performance.Com (2003), White Papers. <http://www.sql-server-performance.com>
- Vaidyanatha, G., K. Deshpande and J. Kostelac (2001), Oracle Performance Tuning 101. McGraw-Hill Companies.
- Shasha, D. and P. Bonnet (2002), Database Tuning: Principles, Experiments, and Troubleshooting Techniques. Morgan Kaufmann.

AUTHOR BIOGRAPHY

Dr. Alexander Pons is an assistant professor in the Computer Information Systems department at the University of Miami. He received his Ph.D. from the University of Miami in Electrical and Computer Engineering in 1998. Dr. Pons has over fifteen years of industry and academic experience as an engineer, consultant, and professor. For the past several years he has been involved in various aspects of real-time systems and databases as a researcher and developer. His research interest includes real-time systems, programming languages, databases and Internet technology.





STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

Copyright ©2003 by the Information Systems & Computing Academic Professionals, Inc. (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to the Editor-in-Chief, Journal of Information Systems Education, editor@jise.org.

ISSN 1055-3096