



Theses and Dissertations

2005-12-01

Advanced Structured Query Language Instruction for Engineers of the Office of Information Technology at Brigham Young University

Vincent Brian Rackliffe
Brigham Young University - Provo

Follow this and additional works at: <https://scholarsarchive.byu.edu/etd>



Part of the [Educational Psychology Commons](#)

BYU ScholarsArchive Citation

Rackliffe, Vincent Brian, "Advanced Structured Query Language Instruction for Engineers of the Office of Information Technology at Brigham Young University" (2005). *Theses and Dissertations*. 720.
<https://scholarsarchive.byu.edu/etd/720>

This Selected Project is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.

ADVANCED STRUCTURED QUERY LANGUAGE INSTRUCTION FOR
ENGINEERS OF THE OFFICE OF INFORMATION TECHNOLOGY
AT BRIGHAM YOUNG UNIVERSITY

by

Vincent B. Rackliffe

A master's project submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of
Master of Science

Department of Instructional Psychology and Technology
Brigham Young University

December 2005

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a project submitted by

Vincent B. Rackliffe

This project has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date

Paul F. Merrill, Chair

Date

Richard R Sudweeks

Date

Stephen C. Yanchar

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the selected project of Vincent B. Rackliffe in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable to fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Paul F. Merrill
Chair, Graduate Committee

Accepted for the Department

Andrew S. Gibbons
Department Chair

Accepted for the College

K. Richard Young
Dean, David O. McKay School of Education

ABSTRACT

ADVANCED STRUCTURED QUERY LANGUAGE INSTRUCTION FOR ENGINEERS OF THE OFFICE OF INFORMATION TECHNOLOGY AT BRIGHAM YOUNG UNIVERSITY

Vincent B. Rackliffe

Department of Instructional Psychology and Technology

Master of Science

This report describes the purpose, design, development and analysis of SQLTips, an online instructional delivery framework and set of instructional modules relating to advanced features and performance tuning of Oracle's Structured Query Language (SQL). SQLTips was developed using Wiki, server-side software that allows users to edit web pages with almost any browser. The report includes a literature review of existing SQL instructional materials and a review of instructional theory. The report also includes a description of the formative evaluation process and results. These results show that SQLTips is easy and enjoyable to use. Based on a scale of 1 to 7 with 7 being the most positive, the 10 modules comprising SQLTips averaged a 6.1 for *ease of use* and a 6.2 for *enjoyability*. Posttest results also showed an average increase of 46% upon completion of the instruction. The report also contains a critique of the project.

ACKNOWLEDGEMENTS

I would like to thank my parents, Paul and Vernita for their encouragement, examples, and most of all, friendship. I would also like to thank my committee—Dr. Paul F. Merrill, Dr. Richard R Sudweeks and Dr. Stephen C. Yanchar—with special thanks to Dr. Merrill for his concern, time and encouragement.

Finally I wish to thank my sweetheart and wife, Suzanne, who has encouraged and emotionally supported me through many long years of schooling. Suzanne, I think I'm finished now.

TABLE OF CONTENTS

Introduction.....	1
Purpose of the Project.....	1
Target Audience Analysis.....	2
Literature Review.....	2
Instructional Materials Review.....	3
Content Research Review.....	8
Instructional Theory Review.....	9
Description of the Instructional Materials.....	20
Application Architecture.....	20
SQLTips Framework.....	23
SQLTips Modules.....	29
Evaluation.....	31
Aesthetics and Usability Expert Evaluation.....	32
Subject Matter Expert Evaluation.....	33
Field Test.....	35
Conclusion.....	41
Critique.....	41
Schedule.....	43
Budget.....	44
References.....	46
Appendix A.....	48

Appendix B.....	52
Appendix C.....	64
Appendix D.....	66
Appendix E.....	67
Appendix F.....	70
Appendix G.....	71

LIST OF TABLES

Table 1. Analysis of quantitative data.....	38
Table 2. Analysis of qualitative data.....	39
Table 3. Estimated and Actual Schedule for SQLTips	44
Table 4. Estimated and Actual Costs for SQLTips.....	45

LIST OF FIGURES

Figure 1. An empty Wiki page.....	22
Figure 2. Detail of screen shot of SQLTips introduction page.....	23
Figure 3. Example module introduction/grabber section (lesson 10).....	24
Figure 4. Example practice exercise.....	25
Figure 5. Example of expected results and explanation of practice exercise.....	27
Figure 6. Example module summary section (lesson 9).....	28

Introduction

Purpose of the Project

This project really has two purposes. The first one is to provide advanced Structured Query Language (SQL) instruction for the engineers of Brigham Young Universities (BYU) Office of Information Technology (OIT). The second purpose of this project is to create a sound framework or template for the creation of future instructional materials for OIT.

OIT is responsible for the management of terabytes of data. These data include current student and class information, employee records, financial data and historical data, just to mention a few areas. Some of the data are static, such as historical data, which are used for the university's decision support system, while other data change regularly such as class schedules and employee personal information. Most of the data owned by BYU are stored in Oracle relational databases. Oracle is currently the top vendor of relational database management systems (RDBMS). Structured Query Language (SQL) is the industry standard language used to access relational databases. SQL provides the means to retrieve data from, add data to, manipulate data in or delete data out of a relational database. During peak usage times such as 12:00 noon on the first day of each semester, hundreds of transactions can occur each second. Each transaction represents a change to the underlying data. When dealing with such a large quantity of data and such a large number of transactions, it becomes crucial to have an efficient system. "While tuning the database engine can help poorly performing applications, nothing can match improving the efficiency of SQL for getting massive performance

improvements” (Harrison, 2001, p. 2). Inefficient programming techniques can cause system slow downs and if not discovered can lead to unnecessary hardware purchases.

Target Audience Analysis

The target audience consists of the software developers and database administrators for the Office of Information Technology – Engineering (OIT-E) department of Brigham Young University. All of them can be classified as technical workers. Within this group is a wide range of experience levels and ages. The least experienced are the sophomore student interns of OIT-E who have limited professional experience; while some members have had many years of software development experience and are close to retirement. Most of them, perhaps all, love learning—as long as it’s something that interests them. Many of them hate taking tests. By the nature of the work they do, most, if not all, are proficient in problem solving skills.

Literature Review

According to Osguthorpe, there are “three types of literature searches that should be conducted as part of the instructional design process: (a) an instructional materials review, (b) a content research review, and (c) an instructional theory review” (Osguthorpe, 1985, p. 20). This section of the report will review the results from these searches. He further states that the purpose of the review is to answer questions and that “the question[s] that initiate . . . the search must be carefully conceived” (Osguthorpe, 1985, p. 20). At the beginning of each subsection, I will state the question(s) that I am trying to answer.

Instructional Materials Review

In this part of the report I will answer two questions. The first question is, “has the proposed instructional product . . . already been developed?” and the second is, “how can I ‘build on existing instructional approaches’” (Osguthorpe, 1985, p. 21). The purpose of the project was to assist software engineers in becoming more effective in their use of Structured Query Language. As training and instructional material was already available for the basic elements of SQL, the project was to focus on several topics in advanced language features and performance tuning as identified by the stake-holders. After discussions with the stakeholders, it was determined that the new instructional material should have the following characteristics.

1. Adequately cover the learning objectives
2. Be available to all interested employees
3. Be easy to update as new features become available
4. Be extensible
5. Be economical
6. Be concise

The search proved that meeting all of these criteria is indeed difficult. For the purposes of the search, I have categorized the existing training material by format. The first characteristic, that of adequately covering all of the learning objectives does not present a problem if the last criteria, that of being concise is sacrificed; the material is available, but it’s typically interspersed with a lot of material that OIT-E is frankly not interested in. A case in point is the SQL instructor led instruction offered by Oracle Education. It spends a significant amount of time on Oracle’s *iSQL* product which, due

to security reasons, OIT-E does not use. The second criteria dealing with availability tends to favor computer based training modules (CBTs) and web based training (WBT)—not only due to their round-the-clock usage capability but also due to the relationship of the availability criteria to that of the fifth criteria, “Be economical.” As the price of the training goes up, the less the likelihood that the part-time, student interns will be able to participate. The third and fourth criteria are also related. As new features become available, it would be nice to be able to add them to the training. It would also be nice to be able to add modules for areas that OIT-E may expand into. Perhaps Oracle will correct the security issues relating to *iSQL* and OIT-E may decide to add the tool to their developer’s options.

In looking for available instructional material that has these characteristics, I first looked at the types of material already being used by OIT-E. One of these was outside classroom instruction. Although outside instruction is generally of high quality, it fails in several areas. Oracle Education has been used by OIT-E for similar instruction. Each day of instruction through Oracle Education runs \$600 USD. In order to cover as many of the instructional objectives as possible would require three classes, *Oracle9i: Database Fundamentals for Developers, Introduction to Oracle9i: SQL, and Oracle9i Database: SQL Tuning Workshop R2*. These three courses would require ten days of instruction for a total cost of \$6,000 USD per student. In addition to the class cost, there is usually additional expense for travel and accommodations. As mentioned previously, due to its cost, it would not be possible to open up this kind of training to all interested employees. It also fails in that the classes being offered do not cover all of the instructional objectives and cover other areas that are outside of the objectives. Learning

Tree is less expensive and provides an alternative to Oracle Education but was still deemed too expensive and failed in the same areas as Oracle Education. Bringing in an outside instructor to teach a custom designed class was also deemed too expensive and failed other criteria as well.

The next option considered was CBTs. OIT-E has licenses for several CBTs, but none that cover SQL. The price of CBTs, is more affordable than classroom instruction, between USD \$100 and \$349 per module, but in a search of available CBTs, I was not able to find any (or combination thereof) that covered all of the instructional objectives. I suspect that this is due to the specialized nature of some of our instructional objectives. Web based training was found to be similarly lacking. In OIT-E we currently use two free WBT modules, *SQLCourse.com* and *SQLCourse2.com*. Both of these provide excellent content, but do not go into the depth that we need.

The final area of available instructional material focused on books, both hard copy and online. There are many books on SQL; for the purposes of this report I reviewed eight books. In reviewing the books I was looking for their potential as self-contained tutorials. In this regard they all fell short—none of them provided structured practice. During the review, I classified the books into three categories, (a) reference without explanations, (b) reference with explanations and (c) certification preparation books. Of these three groups the first was the least useful as a tutorial. The only book in this group was *Oracle DBA SQL: quick reference* (Russel & Cordingley, 2003). The second group, reference with explanation, contained the largest number of books and included the following, *Oracle SQL: the essential reference* (Kreines, 2000), *Oracle SQL: high performance tuning* (Harrison, 2001), *Tuning Oracle* (Corey & Dechichio,

1995), *Mastering Oracle SQL* (Mishra & Beaulieu, 2002), and *Oracle SQL tuning: pocket reference* (Gurry, 2002). The final category was most useful as a tutorial, providing greater interactivity through the use of quizzes and questions. The two books in this final category were : *Oracle9i performance tuning study guide* (Johnson, 2002) and *OCA/OCP: introduction to Oracle9i SQL study guide* (Dawes & Thomas, 2002).

The books also covered the material well, but as with the other available training materials, failed in the “concise” category. These books ranged from 101 pages, for a SQL tuning pocket reference, (Gurry, 2002) up to 630 pages for *Oracle SQL: High Performance Tuning* (Harrison, 2001), either of which is much more reading than OIT-E wanted. As far as the criteria is concerned, books were the most economical, averaging around USD \$30 per hard-copy book. Another problem with the books is that some were already outdated at the time of purchase. The best book for conciseness was *Oracle SQL Tuning*, (Gurry, 2002). It covered a lot of the SQL performance objectives without a lot of extraneous material. It was the smallest and cheapest (USD \$12.95). But even though it claimed to cover Oracle 9i, the version that OIT currently uses, it was apparent that it had been written for an earlier version—some of the material was outdated. In some regards, books are both extensible and updateable—just buy additional books—but with each purchase, the student has to weed through a lot of material to get what he or she is after.

The results of this review indicated to me that there was no training material available that matched the stake-holder’s criteria, so the instructional material would need to be developed.

The second question of this section of the review is “how can I build on existing approaches?” To a certain extent, the media used in the instructional materials will lend itself to certain approaches. The classroom instruction offered by Oracle Education makes heavy use of lab assignments. This is an effective approach for teaching skill based learning since it requires the student to perform the skill being taught. There is an element of practice built into the instruction. And if a student runs into difficulty during the lab, the instructor is available to provide guidance and feedback. The lab assignments, however, do not always apply very well to the tasks the student will be asked to perform in their actual job assignments. In the worst cases this is due to a misalignment of instructional objectives with the real world tasks, but even at best, the student is working with databases and tables that are different than the ones he uses at work. As for computer and web based training, *SQLCourse.com* and *SQLCourse2.com* provide a good example of a hands-on approach. The student is given an opportunity, via his or her browser, to submit SQL statements to an actual database. This not only provides practice, but also immediate feedback. The database being used for the back-end is not an Oracle database, however, and the error message is always the same, “Error occurred. Recheck your SQL statement.” And if a student gets lost there is no instructor available to help. The books I reviewed were of two main types, (a) “Instructional” books that provide information, ask quiz questions, and provide the answer and (b) “Reference” books that provide information and examples. One of the features that I found most useful was the ability to electronically search the text.

To build on these approaches, I decided to develop web-based modules that are designed to be used in conjunction with several of our existing development databases.

This provides the hands-on practice of Oracle Education's classroom instruction, and does so in a database environment which is already (or should be), familiar to the student. Additionally, error messages from the database engine will provide better feedback than that provided by the WBTs mentioned above. Also, the text of the instruction will be electronically searchable and if a student needs further assistance an email address of a content expert will be provided.

Content Research Review

The purpose of this section is to review both practical and theoretical studies on the teaching of SQL or possibly similar subjects. In reviewing the available literature in this area, it is hoped that I can use the information from past studies to improve the effectiveness of the instruction I'm developing. Specifically the question to be answered by this section is, "what insights can be gleaned from other studies on the teaching of SQL that will help me in developing SQLTips?" According to Osguthorpe, the big challenge for this part of the review is "deciding how broad to make it" (Osguthorpe, 1985, p. 20).

I was not able to find anything on the theoretical aspects of teaching SQL specifically, but I did find two articles that deal with the theory of teaching computer programming in general. The first argued that because programming is a skill, the best way to learn is through doing, and the purpose of the teacher is not so much to provide syntax—this can be obtained from numerous printed and online sources—but rather to motivate the students to practice programming. The way to motivate a student is by first achieving a level of trust, next to find out how they are motivated, be it extrinsic rewards, intrinsic desire, etc., and then to promote the art of programming in that area. Crucial to

this process is reassuring the learner that they can be successful in their pursuit of programming. The article further states that over half of the students surveyed report an intrinsic motivation in programming (Jenkins, 2001).

The second article encourages a peer-based collaborative learning environment for the teaching of programming skills. Learners should be provided an opportunity “to discuss the topic with either a tutor or expert” (Jakovljevic, 2003, p. 314).

SQLTips promotes a peer-based collaborative learning environment in the following ways, (a) students are able to add their own insights and experiences directly into the instructional material, and (b) each lesson ends with encouragement to the learner to contact the primary subject matter expert involved with the development of the instruction should there be any questions.

Instructional Theory Review

In this section I review instructional theory for “prescriptive advice” (Osguthorpe, 1985, p. 21) on the design and development of SQLTips. According to Osguthorpe, three types of literature should be reviewed for this section, “(a) general principles of instructional design, (b) theory and research related to a particular category of learning, and (c) principles associated with a specific deliver system” (Osguthorpe, 1985, p. 21)

General principles of instructional design. For designing the instruction on this project I used the Dick and Carey Systems Approach Model (Dick & Carey, 1996).

There are three sound reasons for using this model. The first reason is the focus provided by the precise statement of what the learner is supposed to know or be able to do as a consequence of the instruction. Without this precise statement, the other steps can become “unclear and ineffective.” The second reason is the close relationship between

the various components of the model; the choice of the instructional strategy based directly on the specific learner outcomes. The final reason provided for using this model is its reliance on empiricism. Usage data are collected and analyzed to show the efficacy of the instruction and, based on the analysis, the instructional material is modified and improved (Dick & Carey, 1996, pp. 2-7). As such this process is an iterative one.

There are ten components to the model.

1. Determine the instructional goal
2. Analyze the instructional goal
3. Analyze learners and contexts
4. Write performance objectives
5. Develop assessment instruments
6. Develop instructional strategy
7. Develop and select instruction
8. Design and conduct the formative evaluation of instruction
9. Revise the instruction
10. Conduct summative evaluation

For SQLTips, the instructional goals or General Instructional Objectives (GIOs) were developed in conjunction with several of the stakeholders. One of the stakeholders is a team-lead of a development group. He had designed some basic SQL instruction for new employees and recognized a need for instruction in more advanced topics in SQL. He approached the lead of the database administration team who referred him to me. I get approached somewhat regularly by developers needing assistance with SQL. As the content expert, I put together an initial list of objectives based on my “practical

experience with learning difficulties of students” (Dick & Carey, 1996, p. 5). These original GIOs were then reviewed and added on by several stakeholders. Additional needs were also recognized and included. Stakeholders that supplied input consisted of (a) the chief engineer of one of the development teams, (b) the chief engineer of the database administration team, (c) one of the directors and (d) the chairman of one the advisory groups. Due to limitations in time and budget, some of the original GIOs were discarded in favor of others.

The purpose of the second component, that of analyzing the instructional goal, is to determine “entry behaviors...required of learners to be able to begin the instruction” (Dick & Carey, 1996, pp. 5-6). As mentioned above, there was already some basic SQL instruction being given to new employees. This basic instruction was used to define those “entry behaviors.” Additionally, within SQLTips itself, some of the modules require the learner to have knowledge or skills addressed by a previous module. An attempt has been made to order the modules in such a way as to address this issue. However in constructing the material into modular format, it’s understood that learners will have the flexibility to customize their own training—the instruction catering to the “different initial knowledge levels” of the students (Lewis, Dean, Whitlock, & Bryson, 1987, p. 194).

The next component is that of analyzing the learners and contexts. The analysis of the learners is listed elsewhere in this report. The context in which the learning will take place is through on-the-job, self-directed study. It is anticipated that the modules will be used in two primary ways by the learners—initially as a complete “beginning-to-

end” set of instructions on advanced topics in SQL and secondly as a reference providing “just-in-time learning” (Simon, 2001, p. 64) when the need arises.

The next component is to write the performance objectives. A performance objective is a “detailed descriptions” of what a learner should be able to do after completing the instructional activities (Dick & Carey, 1996, p. 119). It is sometimes referred to as a Specific Learning Outcome (SLO). SLOs are associated with GIOs. As an example one of the GIOs for SQLTips, was to “Understand key points about indexes”; two of its associated SLOs were (a) “List the two primary purposes of indexes” and (b) “Describe the costs of indexes.” As mentioned previously, the GIOs for SQLTips were agreed upon by the stakeholders—the SLOs were also. This agreement is critical to the success of the instruction.

Some of the original GIOs and SLOs attempted to go into too much detail in areas that were of less importance. There is a wealth of highly detailed information on almost every aspect of SQL already available, both on the web and in books. One of the purposes of SQLTips is to filter out all but the critical content from the “traffic jam” of information available (Simon, 2001, p. 64). As an example of this filtering, one of the objectives of SQLTips is for the learner to be able to determine which indexes are being used by a query. In order to do this, the learner must also have some understanding of Oracle’s autotrace facility which shows a query’s execution path as determined by the Oracle optimizer, but they don’t really need to understand all of the workings of the Oracle optimizer. The original objectives were changed, and a module devoted the Oracle optimizer was consequently not developed. A complete listing of the GIOs and SLOs for SQLTips can be found in Appendix A.

The next component of the design process is the creation of assessment instruments. Ideally the items in the assessment instrument should parallel the objectives (Dick & Carey, 1996). For the field test portion of this project, I decided to use a closed-book pretest/posttest comparison. Since the behavioral objectives themselves are not *closed-book*, this posed some difficulty in designing the assessment instrument. As an example, one of the objectives is to “create a table and index according to our standards.” It is not expected that the learner would be able to do this, at the conclusion of instruction, without referring to an example template. So, rather than using an assessment item such as “Create a table according to our standards” which would require unwanted memorization, I used the item, “You want to create a table; where would you find syntax examples for following our standards?” By wording the item in this manner, although not directly testing the behavior, it provides a strong indication that the behavior exists. As a side note, most of the assessment items are free-response much like the one just mentioned. A complete copy of the assessment instruments used for SQLTips can be found in Appendices B and C.

The sixth component of Dick and Carey’s model is to “Develop an Instructional Strategy” (Dick & Carey, 1996, p. 6). This component includes deciding on what media to use, how to sequence and chunk the information, how to motivate the learner, the type of practice that will be provided, what feedback will be offered, and what follow-through advice will be given.

For SQLTips, the media choice was simple to make; since it was decided that the instruction would be self-directed and all participants would have access to the internet, a web-based system was chosen. Furthermore, in consideration of the department’s recent

adoption of Wiki as a knowledge-base, it was decided to expand its use to include instruction.

According to Kierns, self-directed instruction “must be presented in small enough increments to assure frequent response, and the pace at which the learner moves must be his own” (Kierns, 1999, p. 10). SQLTips is composed of ten modules with each module covering one general instructional objective. Each module is designed to take approximately thirty minutes to complete. The material in each module is chunked in several logical places allowing interaction by the learner.

For theory on motivation, many instructional designers turn to Keller’s ARCS model, (Kierns, 1999; Dick & Carey, 1996). According to Keller there are four components to motivation in learning. They consist of Attention, Relevance, Confidence and Satisfaction. “Attention” refers to gaining and keeping the attention of the learner. “Relevance” pertains to the convincing the learner that the material is important for them to understand; “Confidence” deals with self-efficacy regarding the learner’s ability to understand the material; “Satisfaction” means that the learner is convinced that training meets their needs (Dick & Carey, 1996, p. 185).

Each module in SQLTips opens with an introduction. Each introduction is designed to capture the attention of the learner by posing interesting questions, problems or facts. Attempts are also made to maintain this attention throughout the material by providing “learning situations with information which learners see as possibly contradictory, incomplete or unexpected to stimulate them to seek new information to resolve the conflict” (Keirns, 1999, p. 143). The introductions also serve the purpose of providing the learner a list of “objectives” for the lesson. These objectives are usually

worded less formally than the specific learner outcomes as listed in appendix A to avoid losing the learners attention. Furthermore, each introduction provides an HTML link to a pre-test, so that the learner can do a self-assessment prior to starting a module. In summary the introduction in each module serves to “to convince the learner of the usefulness and interest of the instructional experience” (Keirns, 1999, p. 19).

After the introduction, the module is broken into several sections dealing with the information and concepts to be learned. Often a single section will deal with only one specific learner outcome. At the end of each section the learner is given a question or task. In doing this, I am following the advice of Keirns, “learners must respond actively to the material being presented. Questions, choices, and interactions must be available constantly for the learner to respond to. Just reading or observing is not enough to make the presentation of information function as instruction” (Kierns, 1999, p. 10). After performing the task, the learner has the option to click on an HTML link to get the correct answer and additional feedback. Again from Keirns (1999),

Every response is immediately followed by a consequence: the answer to a question is affirmed or corrected, a choice leads immediately to a result. These interactions go beyond the ‘stimulus-response’ pattern of behaviorism when they represent the cognitive activity of the learners as they interact with the information. (p. 10)

At the end of each module is a reference section providing resources for additional information and learning. The learner is also encouraged to speak to the author about any difficulties they’re having with the material.

The next step in the Dick and Carey model is to “Develop Instruction.” It’s critical in this step to keep in mind the objectives and goals so that the instruction is aligned with them. Dick and Carey state that at the completion of this phase of the design “you should have a draft set of instructional materials [that] will be reviewed and revised

based upon feedback from learners...and subject matter experts” (Dick & Carey, 1996, p. 243).

The eighth and ninth steps in the model are to “Design and Conduct the Formative Evaluation of Instruction” and to “Revise the Instruction.” These steps illustrate the iterative nature of the process and are important in producing an effective product. According to Flagg (1990, p. 1) , “formative evaluation helps the designer of a product, during the early development stages, to increase the likelihood that the final product will achieve its stated goals.” Two techniques were used for the formative evaluation the first was one-to-one evaluations and the second was a field test. For the one-to-one evaluations, each module was evaluated by two learners selected from the target audience. After each evaluation, modifications were made to the module and any preceding modules to which that particular improvement also applied. Following the one-to-one evaluations. Two field tests were conducted with five learners in each group. The first test was done after completion of the first five modules. An anonymous pre-test was administered to each of the participants and minimal instructions were provided on where to find the modules. Following completion of the modules, an anonymous post-test was administered. The results of these tests are included in this report.

The final step in the Dick and Carey model is to conduct a summative evaluation. This step is outside of the scope of the project.

Theory and research related to a particular category of learning. Gagne, Briggs and Wager (1988) delineate five kinds of learned capabilities, namely (a) intellectual skill, (b) cognitive strategy, (c) verbal information, (d) motor skills, and (e) attitude. According to Gagne et al. (1988, p. 55), “intellectual skills can be [further] categorized

by complexity” and that “such categories cut across and are independent of types of subject matter.” These categories are hierarchical by nature in that the more complex “require as prerequisites” an understanding of the less complex. This hierarchy, within “intellectual skills” is composed, from the least to the most complex, of (a) discriminations, (b) concrete concepts, (c) rules and defined concepts, (d) higher order rules and (e) problem solving. The category into which the instructional objectives belong should be considered when designing the instruction, however, “with the exception of motor skills, all of these categories are likely to be involved in the planning of any course” (Gagne et al., 1988, pp. 50-51).

As stated previously, the intended audience for SQLTips consists of professional programmers and database administrators. As such it is assumed that they are already experienced in problem solving skills, so the instructional objectives deal primarily with the categories of “rules and defined concepts”—the purpose being to add to the arsenal of tools available to the participants in their problem solving activities. To a lesser degree, SQLTips also deals with “verbal information”, “higher order rules” and directly with “problem solving” (Gagne et al., 1988, pp. 61-67). In this part of the review, I’ll be dealing with instructional theory on the following categories (a) defined concepts, (b) rules, (c) higher order rules and (d) problem solving.

Defined concepts are abstract in nature. A defined concept is understood when the learner “can demonstrate the meaning of some particular class of objects, events or relationships.” To do so, the learner must be able to break down the concept into its component concepts and show an understanding of each component and its relationship to the others (Gagne, et al., 1988, pp. 59-62).

Rules are similar to “defined concepts”, in fact, “defined concepts” are a classifying type of rule. A rule is differentiated from discrimination by its complexity; it consists of a class of responses to a class of objects. A learner is understood to have learned a rule when his or her behavior is consistent in following the rule (Gagne, et al., 1988, pp. 63-65).

Higher order rules are “complex combinations of simpler rules” and are often used for problem solving. The key to this kind of instruction is that the learner develops the ability to use a combination of rules that have been previously learned for solving a type of problem that has not been encountered previously by the learner (Gagne, et al., 1988, pp. 65-67).

SQLTips covers all of these areas. Through their interaction with the instruction participants gain an understanding of the defined concepts of “indexes” practice using rules on object creation and index use. As a final result, the learners should have enough background to be able to use these concepts and rules to solve the problems of poorly performing queries.

Principles associated with computer based, self-directed learning. In this final section of the literature review, I will review, as per the recommendations of Osguthorpe (1985), the principles associated with the delivery system I have used for this project. The delivery system for SQLTips is computer assisted, self-directed learning. In this section I will cover both aspects, (a) the computer assisted element and (b) the self-directed learning element of the deliver system.

The term “computer assisted instruction” has come to mean more than just instruction via a computer screen. It is possible to create content, to be delivered via

computer that mimics other methods. In many cases this mimicry may result in an inferior product. A simple video which is played on a computer showing an instructor is not as effective as having an instructor present. An eBook is arguably not superior to a printed book. According to Simon, “We must not use technology just because it is available. We must use it when and only when, we can see how it will enable us to do the educational job better” (Simon, 2001, p. 63).

Lewis et al. (1987), list several possible advantages of computer based training over other training methods. Good computer based instruction should be designed to capitalize on these advantages where possible. The first advantage is increased “trainee throughput.” If the material is modular, it allows for the “different initial knowledge levels” of the students. Students can customize their own learning and those “who could not justify attending a course for a few days only part of which would be relevant can study the relevant parts of a CBT course and learn the missing skills” (Lewis, et al., 1987, p. 194). They also point out the advantage that computer based training has in providing interactive training. A case study was mentioned in which the performance of newly hired data entry operators was substantially improved by changing their training program to use the actual terminals in the training program that they would be using on the job. Prior to this, they had been using typewriters in the training program.

Gagne, et al. (1988) set some guidelines for the modules as minimally including

1. A performance objective.
2. A set of materials and learning activities either self-contained in the module or external to the module.
3. A method for self-evaluation of mastery of the objective.

4. A provision for verification of the learning outcome by the teacher.

For the most part, SQLTips attempts to follow these guidelines. However, since the instruction is not required for a class and there is no teacher, the last guideline is ignored.

Gagne et al. also state that “individualized instruction typically provides more frequent feedback and more frequent progress checks that is the case for conventional instruction. It may permit more freedom of choice on the part of the learner, depending on the extent to which objectives are optional or required. Usually, as a minimum, the learner sets his own pace in learning activities (Gagne, et al., 1988, p. 315). SQLTips follows these principles as well.

One of the strengths of SQLTips is that it lends itself to learner modification. According to Keirns (1999, p. 165), “as we move into ‘distributed’ or ‘distance’ or ‘online’ learning environments, the changed roles of learner-designer are even more evident. The interaction between peers, content, and the directing element of formal courses in these formats alter dramatically and must draw on the learners’ input in new ways.” SQLTips is easily modifiable by the users and learners are encouraged to add example statements that illustrate the given topic. So far there has been only one case of an addition made by a participant, but I assume they will increase as the intended audience becomes more aware of the existence of the instruction.

Description of the Instructional Materials

Application Architecture

As mentioned previously, SQLTips was developed in Wiki. Wiki is an open source, server-side software package that allows users to create and edit web pages from most client web-browsers. Prior versions of pages are automatically stored on the server

for change rollback capability. Wiki was used for the following reasons, (a) the intended audience already uses a knowledge base for their work that resides in Wiki and they are familiar with it's use, (b) Wiki provides a simple update capability so that the instruction can be kept current with industry enhancements and changes, (c) additional modules can be easily added, and (d) the learners can contribute real-life examples and experiences that relate to the topics. This last point was critical. SQLTips provides a collaborative learning environment. According to Keirns (1999, p. 118), "the range of understanding achieved by . . . several members of the group may well be more extensive than that arrived at by any single individual within the group, and all can benefit from sharing personal insights."

Figure 1 shows an empty Wiki page of the OIT-E Wiki. The reason I put this figure in the text is to show what the pages look like prior to adding the instructional material. The page is broken up into two main sections. The dark area that covers the left and the top of the page is not directly modifiable by an end-user. The inner white section is editable by anyone with access to the page and a standard web browser. The page is edited by first clicking on one of the "Edit This Page" links either at the top of the white section or at the bottom of the text in the white area. Clicking on one of these will put the user into a Microsoft Notepad-like editor. In the editor, one is limited to very few formatting options. There are four font sizes to choose from and there are bold, italics and monospaced font options. Text color and link color are not modifiable at the page level. A note about the color of the hyperlinks, normally hyperlinks are blue, but the developer that set up the Wiki for OIT-E set the hyperlink text color to orange. This non-standard color has caused some difficulties which I'll discuss later.

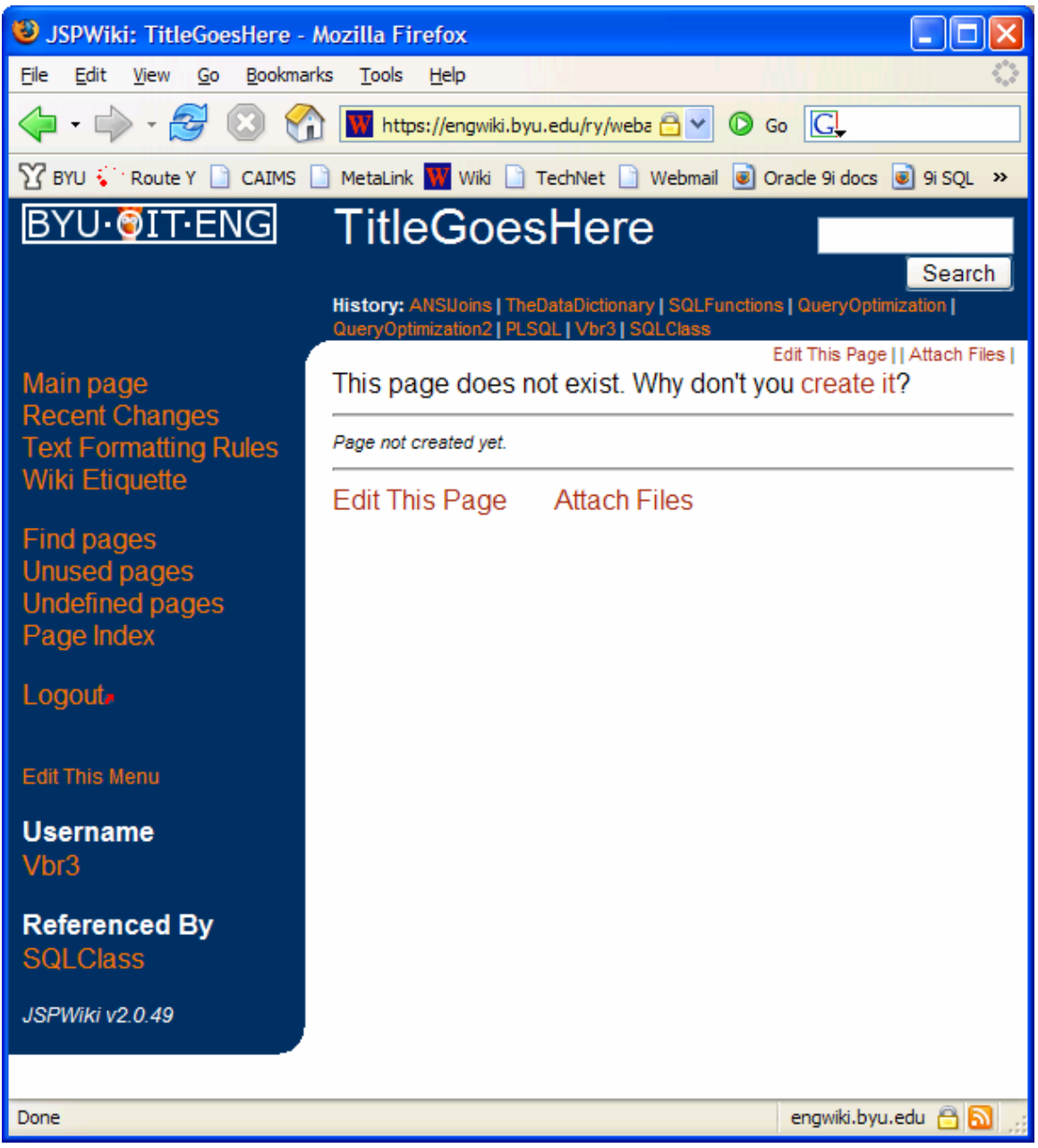


Figure 1. An empty Wiki page

SQLTips Framework

One of the purposes of the project was to develop a framework for other instructional design projects for OIT-E. Collectively the engineering staff of OIT have a tremendous amount of expertise in their respective fields. In the department, however, there is limited experience in instructional design. By putting together good instruction, I'm hoping that others in the department will use this format in the development of their own instruction.

SQLTips has the following components, (a) an introduction page and (b) lesson modules. Figure 2 shows a detail of a screen shot of the introduction page. (For a full screen shot see appendix F.)

SQLTips

SQLTips is a tutorial/reference for advanced SQL. The modules in SQLTips presume a prerequisite background in basic SQL. This background can be obtained via the following resources:

- ◆ [Crowther and Buck's Training Manual](#) (chapter 11)
- ◆ [sqlcourse.com](#)
- ◆ [sqlcourse2.com](#)

All of the examples and exercises in SQLTips are done through SQL*PLUS. I strongly recommend that you review the following module prior to beginning SQLTips.

- ◆ [Using SQL*PLUS](#)

SQLTips Modules

1. [Tablespaces, Schemas and Database Objects](#)
2. [Understanding Indexes](#)
3. [Data Types](#)
4. [Creating TABLES, INDEXES and KEYS](#)
5. [New ANSI Join Syntax](#)
6. [The Data Dictionary](#)
7. [Functions and Case Statements](#)
8. [Query Optimization - Index Use](#)
9. [Query Optimization 2](#)
10. [PL/SQL](#)

Figure 2. Detail of screen shot of SQLTips introduction page.

The introduction page contains a description of the nature of the instruction, a list of prerequisite skills with links to resources to aid in developing those prerequisite skills and links to the lesson modules.

From a framework standpoint, each lesson module has the following components,

1. An introduction that lists the objectives in an informal manner and provides an attention grabber.
2. Practice code along with expected output.
3. Problems with answers.
4. A summary section.
5. Navigation links to the SQLTips Introduction page and the next module.

The first section of each module is the Introduction/Grabber section. Figure 3 provides an example from one of the modules. The introduction is intended to serve the dual purpose of capturing the attention of the learner while providing a brief overview of the nature of the material—in this regard it's similar to an *objectives* section.

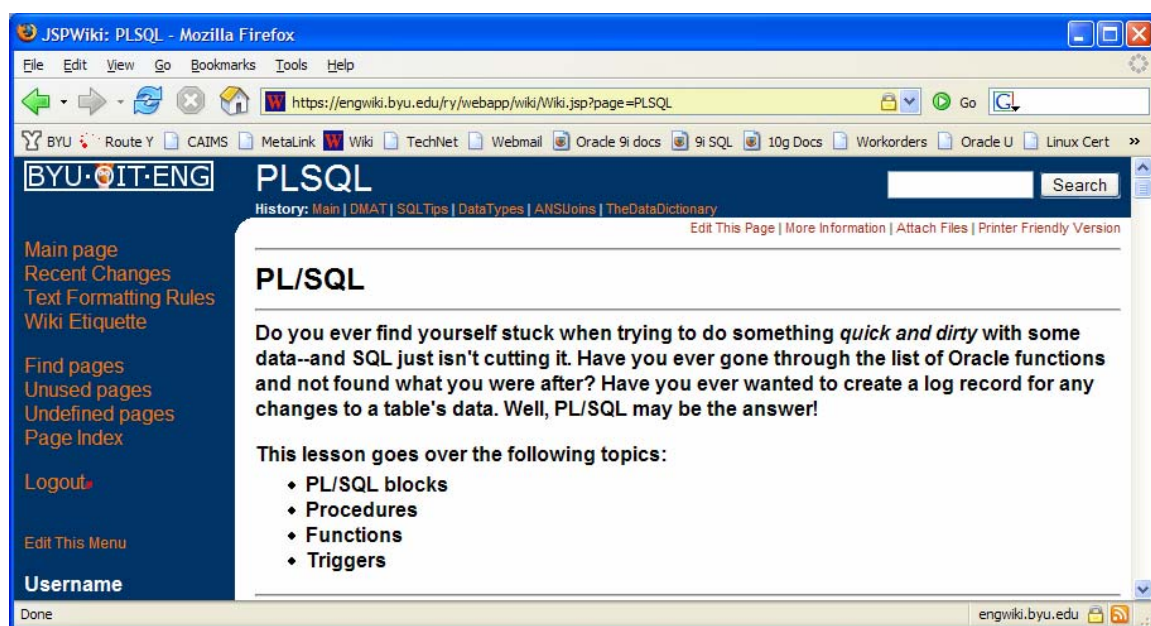


Figure 3. Example module introduction/grabber section (lesson 10).

Just below the introduction, on the same page, begins the actual instruction. Interspersed with the instruction are practice problems. Figure 4 shows an example of what a practice problem looks like. Most practice problems have example code that the learner is supposed to run against a development database. An attempt was made to make as much code, as possible, similar to actual code the learners will be expected to write. The development database has the same tables and other database objects as BYU's central production database, so an added benefit is that any new employees will gain important experience with the structure of actual production tables while going through the SQLTips lessons.

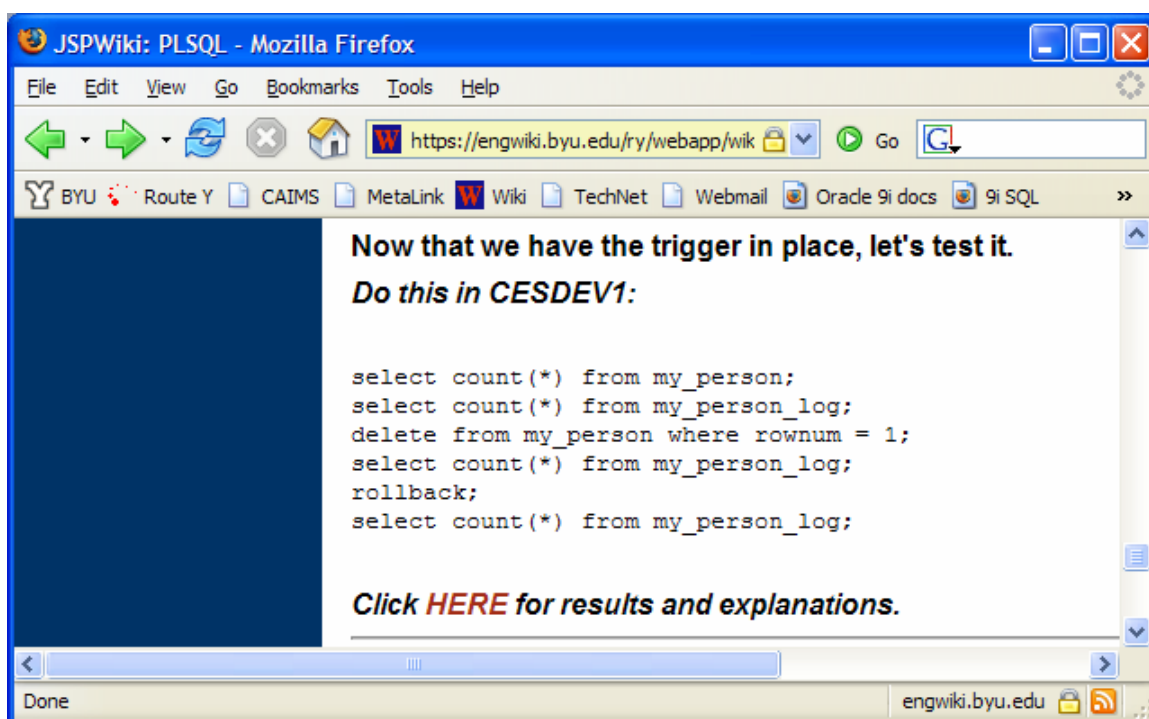


Figure 4. Example practice exercise.

Practice problems usually begin with the italicized phrase, “Do this in CESDEV1”; CESDEV1 is the name of the database. Following this phrase is a set of code written in a monospaced font. The learner can either type it out by hand in

SQL*Plus (a tool used to submit SQL statements to the database), or they can copy and paste the code into a script, which can then be saved and submitted to the database. Following the code sample is the phrase, “Click HERE for results and explanations.” I realize that in using the words “Click HERE”, that I’m going contrary to conventional wisdom—hyperlinks should normally have a more descriptive name to facilitate screen readers for the vision impaired. The reason I chose to do this was from the input I received from two different reviewers in one of the original modules. In that module I used the phrase “Results and Explanations” for the hyperlink. Due to the non-standard color, orange, of the hyperlinks in OIT-E’s Wiki, these reviewers failed to notice that there was a link there at all, and one of the reviewers made the criticism that I didn’t provide any results for the practice problems—even though they were there all along. After changing the hyperlink text to “Click HERE”, none of the other reviewers have appeared to have any problem recognizing the hyperlinks. As alluded to earlier in this report, I’ve submitted a request to have the hyperlink color changed to the standard blue, with underlines. When this change occurs I may change the hyperlink text to be more descriptive, however, due to the fact that none of our developers currently use a screen reader, I don’t believe the change is critical.

If a learner chooses to follow the “results and explanations” link, they are taken to a page that provides actual output from a SQL*Plus session, along with commentary.

Figure 5 provides an example of a “results and explanations” page.

The “results and explanations” pages usually follow the same format. The title of the page (generated automatically by Wiki based on the name of the page) contains an abbreviated name of the lesson module along with a sequential number. This convention

was done primarily for the convenience of the designer. In the white text area of the page, at the top is the main header. This header usually contains the phrase “Here’s what I got” written in a large font, two sizes up from normal, and with a line below it. The phrase is a bit colloquial, but this has been done on purpose to make the instruction less formal.

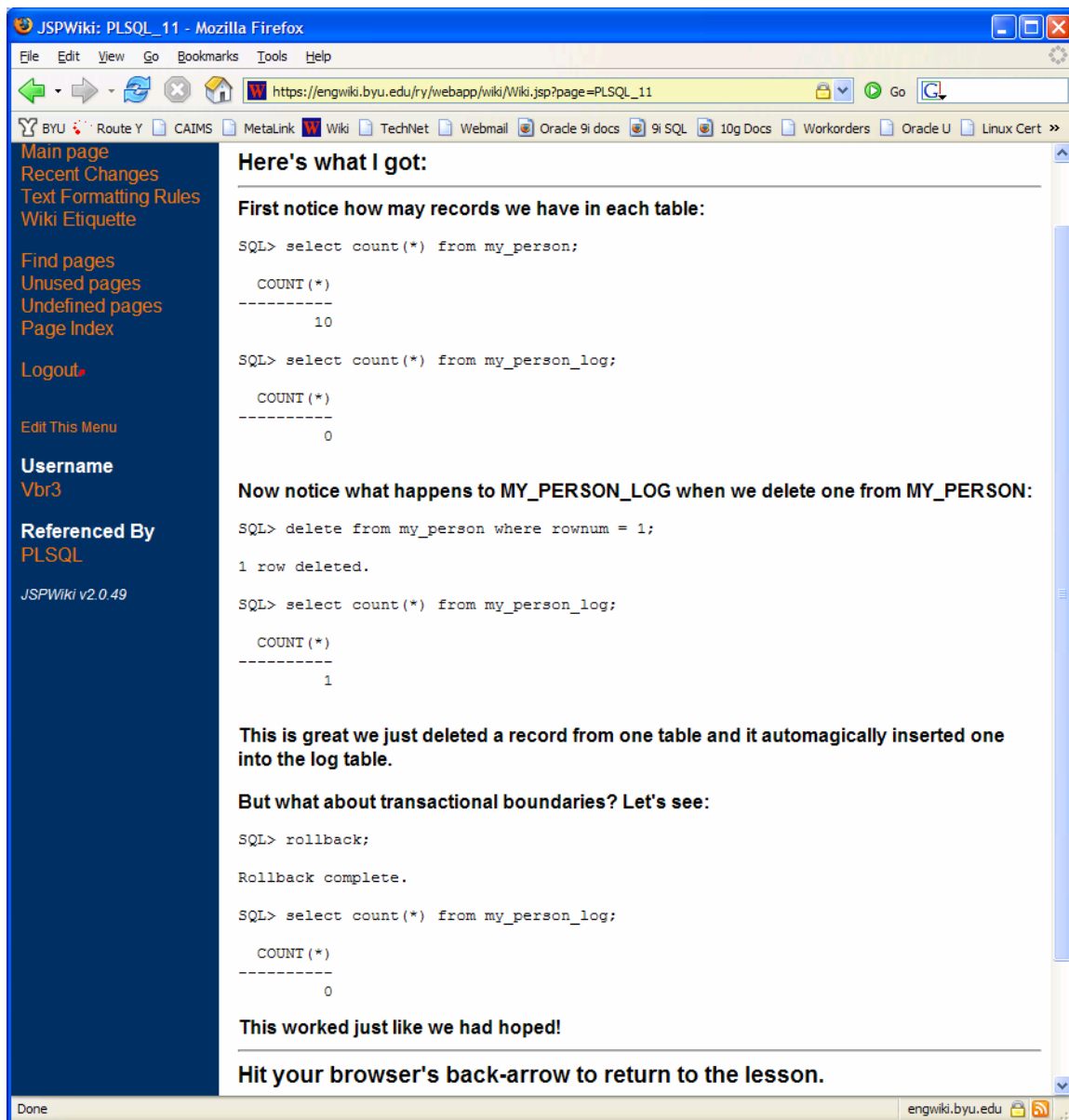


Figure 5. Example of expected results and explanation of practice exercise.

Below this phrase is the output and commentary, with the output being displayed in a monospaced font and the commentary in a larger normal font. At the bottom of the “results and explanations” page, is the instruction to “Hit your browser's back-arrow to return to the lesson.” I had originally put in a hyperlink to return to the main page of the module, but this did not work out well. By using the browser’s back-arrow, the learner is returned to the same location in the module from where they clicked the hyperlink. Since some of the modules are quite long, it made sense to do it this way, so they wouldn’t have to spend time finding where they left off.

At the conclusion on each module’s instructional content is a summary. Figure 6 shows an example of a lesson summary and concluding elements. This summary is used to remind the learner of the main points of the instruction and reiterate the most critical elements.

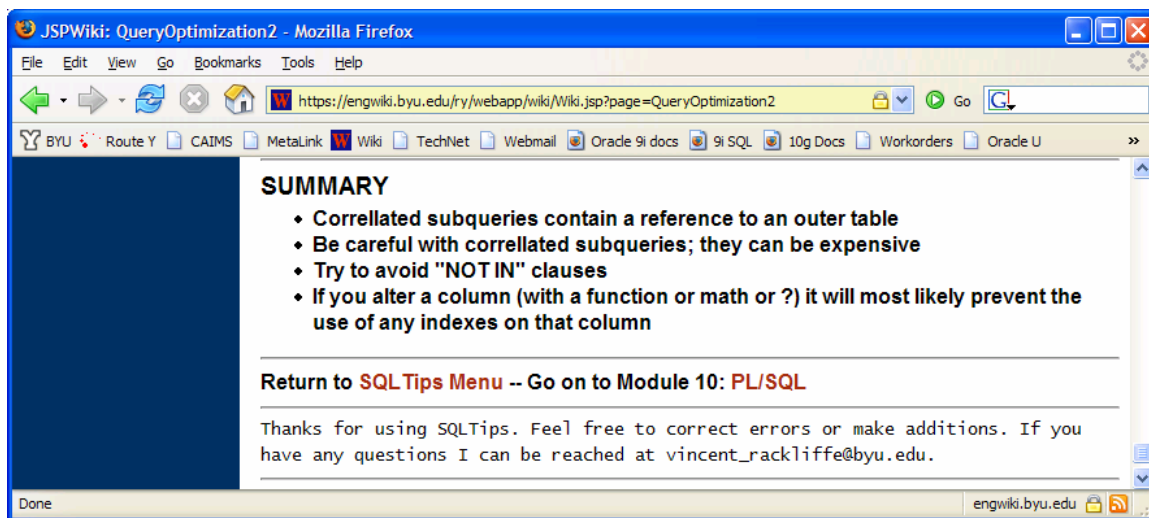


Figure 6. Example module summary section (lesson 9).

In most cases I chose to format the summary as a bulleted list. Following the summary are two hyperlinks, one that points back to the SQLTips Introduction/Menu page and the other points to the next lesson module. In this case the odd color of the

hyperlinks does not appear to be an issue, since they are located in a position in which a learner expects to find a link. The hyperlinks are separated from the rest of the module with a horizontal line above and below them. Below the hyperlinks is a note to the learner to “Feel free to correct errors or make additions” and “If you have any questions I can be reached at vincent_rackliffe@byu.edu.” This note is intended to reinforce the Wiki concept of collaborative learning. The next subsection will provide greater a synopsis of the content of each of the modules.

SQLTips Modules

As mentioned previously, SQLTips contains ten lesson modules. This section of the report gives a brief overview of what each module contains.

The first module provides some background on physical and logical storage devices, namely tablespaces and schemas, and also on database objects, including tables, indexes, views, sequences and synonyms.

The second module goes into greater depth on indexes reiterating the purposes of indexes, briefly explaining some of their physical properties and describing conceptually, how they work. It illustrates the performance benefits of indexes with an example that the student runs against a development database. It also provides examples on how to determine which indexes exist for a given table by querying the data dictionary. It concludes by pointing out some of the negative aspects of indexes.

The next module provides instruction in datatypes. It is assumed that everyone in the target audience has some understanding and experience with some datatypes, so the module’s focus is on those aspects that a typical developer may not have considered. First it provides instruction on some newer datatypes that a developer may not be aware

of, such as NVARCHAR2 and BFILE. It also provides instruction on three benefits of proper datatyping, (a) built-in type constraints, (b) built-in functions and (c) performance benefits. It also provides practice to illustrate the first two of these benefits.

The fourth module was added at the request of OIT-E's Data Model Advisory Team, to provide instruction on creating database objects according to their new standards. This module explains what a primary and foreign key are, and provides practice with creating tables that implement both. These practice problems allow the learner to see Oracle errors that may occur when working with these keys. Syntax examples for creating both natural primary keys and surrogate keys are given at the end of the lesson.

The fifth module provides instruction on the use of the ANSI standard JOIN syntax newly adopted by Oracle. It provides experience with writing SQL statements using the new syntax, pointing out its benefits and drawbacks. It encourages the use of some of the ON and LEFT OUTER clauses and discourages the use of the USING clause and the NATURAL JOIN clause.

The sixth module deals with Oracle's data dictionary and was added at the request of one of OIT-E's directors. This module provides practice with the two primary types of dictionary views, the dynamic views and the static views. Since the static views are of greater use to developers it deals with them in greater depth. Some real-world examples are used to illustrate the benefits of being familiar with the data dictionary. One example shows how to determine if an index has been dropped from a database, by comparing the DBA_INDEXES view with that of another database across a database link.

The seventh module introduces the learner to some additional ANSI standard syntax that was recently adopted by Oracle, namely the CASE statement. It also provides instruction on several of the more useful Oracle functions and provides practice on using the new syntax and functions.

The eighth module is the first of two modules on query optimization. This first one concentrates on making sure the correct indexes are in place. It starts by having the learner create some tables to practice with. Next it emphasizes good database design by having the learner create primary and foreign keys. Finally it provides instruction and practice in using AUTOTRACE to determine if indexes are being used.

The ninth module also deals with query optimization, by going over several constructs that can hurt performance. It compares the NOT IN clause to the NOT EXISTS clause and then shows how either one can be rewritten using a MINUS set operator. It shows through guided practice that the relative performance between these different constructs varies and recommends testing to determine which is best for any given situation. The module also shows through practice, several ways to rewrite simple clauses that prevent index use so that they are able to make use of available indexes.

The final module introduces the learners to PL/SQL, Oracle's procedural language. It provides examples of how using PL/SQL can do some tasks significantly faster than straight SQL. It also provides practice on storing PL/SQL in the database as procedures, functions and triggers, and explains the benefits of doing so.

Evaluation

SQLTips was subjected to various evaluations throughout the inception, design, and production stages. Instructional objectives were written up prior to the design of

instruction. These objectives were then reviewed by two stakeholders consisting of the chief engineer of the Enterprise Data Services team, and the chief engineer of one of the development teams. One of the Directors of OIT-E also provided input. The objectives were then modified based on their inputs. The designer then created a small example module which was reviewed by the same two stakeholders and by Dr. Merrill to see if the intended instructional approach and delivery mechanism were acceptable. SQLTips was now ready to begin its formative stages.

In accordance with Dick and Carey's (1996) recommendations on instructional design, SQLTips underwent several evaluations while in the formative stages. These evaluations were used to "obtain data that [could] be used to revise [the] instruction to make it more efficient and effective" (Dick & Carey, 1996, p.256). These formative evaluations consisted of a graphic arts, aesthetics and usability evaluation, subject matter expert (SME) evaluations and one-to-one evaluations. After completion of the modules, they were also field tested to determine effectiveness of the instruction.

Aesthetics and Usability Expert Evaluation

After completion of the first module, a graphic arts and usability professional reviewed it. After understanding the limitations inherent in Wiki, he had only a couple of recommendations,

1. Make all headings one incremental size larger.
2. Use all-caps for the lesson titles.

Subject Matter Expert Evaluation

The primary SME on the project was the designer. However, after the first draft of each module was completed, two additional SMEs reviewed the module and filled out a questionnaire containing the following the questions,

1. Did you notice any inaccuracies in the content of this module?
2. In what areas, if any, is the module weak in answering the objectives listed at the top of this questionnaire?
3. Suggestions on how to make the module more effective?
4. Any other comments or recommendations?

Any inaccuracies were corrected or clarified. Some recommendations were included while others, due to time constraints, will be implemented at a later time. A complete listing of their responses is contained in Appendix G. This evaluation was invaluable in that it showed some weaknesses in the material that I had overlooked. On occasion I disregarded comments, such as one for module 8 which read, “On the explain page QueryOptimization_09 – didn’t like the font. And it was different that (sic) the other explains.” In this case, the font was identical, and since the passage in question was what Wiki refers to as “preformatted text”, there were no other options. I believe the evaluator was just mistaken in what she was seeing. With some modules, I made a modification for each comment made by an SME. For example in module 9, I received the following comments.

1. “Poorly (sic) performance could be other things than the 3 items listed. I think this should be noted somewhere.”

2. “Where discussing autotrace – refer back to the previous module; just in case [someone is] taking these modules out of sequence.”

I considered both comments to be good observations and made changes accordingly.

When the SME evaluations of each module were finished and corrections made, I conducted one-to-one evaluations. According to Dick and Carey (1996), the purpose of the one-to-one evaluations is to gather initial information about the clarity of the instruction, the impact that the instruction will have on the intended audience and the feasibility of the instruction, given the available time and context. Although they recommend using three or more learners for these evaluations, due to limited personnel resources I used only two (Dick & Carey, 1996, p. 258).

For these evaluations I sat with each evaluator individually as they went through the instructional material. The evaluators were really good at catching typos and asking questions that showed weaknesses in the explanations. Most of the problems that were identified by these evaluators were worked on. On several occasions it was a humbling experience to see that something that I had thought was very clear and easily understood wasn't. I considered this set of evaluations critical to the quality of SQLTips. Based on these evaluations more modifications were made to the instruction. As an example, one of these evaluators made the recommendation for module 9 that I “explain what a ‘reference to the outside’ means in the example.” I had another evaluator go over that same section of instruction, both before and after clarifying modifications, and he felt the modifications really helpful. Another example was that after having completing module 4 one of the evaluators was confused with the relationships between *surrogate keys*, *natural keys*, *intelligent keys* and *non-intelligent keys*. Although the text defined the

relationships, because of his confusion, I reemphasized the definitions in the module summary.

Field Test

According to Flagg (1990), the field test or *implementation formative evaluation* requires “testing the effectiveness of the [instruction] under approximately normal use conditions (p. 6).”

Description of participants. For the field test of SQLTips, ten participants were originally chosen from the intended audience pool based on availability and willingness. Not all of the ten completed all ten modules due to other more pressing commitments. In fact, for modules 6 through 10, five additional participants were added to the pool. Fourteen of the participants are full time employees of OIT-E with several years of professional software development experience. One participant was a student intern of the department.

Evaluation instruments. The evaluation instruments consisted of a pretest and identical posttest, a qualitative module-specific questionnaire and a qualitative SQLTips framework questionnaire that covered SQLTips as a whole.

The pretest/posttest instruments were designed to address each of the specific learning outcomes as contained in Appendix A. In total there were nine questions for module 1, nine for module 2, ten for module 3, seven for module 4, six for module 5, six for module 6, twelve for module 7, eight for module 8, ten for module 9 and five for module 10, for a grand total of eighty-two questions. The bulk of the questions, 45, were constructed response questions. Item 9 for module 2 provides an example of this type of question; it asks, “Why is an index usually stored in a different tablespace than the table

it references?” Of the remaining questions, 23 were fill-in-the-blank or required a single word response, 13 required writing code and two were multiple choice. One of the questions, had two parts, the first was multiple choice and the second required a longer response; I counted this question in both the multiple choice and constructed response categories. This particular item referred to a box with an SQL clause in it, and posed the question “Given the following WHERE clause, pick which index(es) you would build.” It then provided the following options: (a) Two indexes: one on PERSON_ID and one on ADDRESS_TYPE, (b) One index on (PERSON_ID, ADDRESS_TYPE), or (c) All three indexes listed above. The item then had a follow-up of “Explain your choice.” I put this last part in so that I could have a better idea as to the learner’s understanding of the concept. Many of the items had the learner refer to a text box with a code sample in it. A complete copy of the pretests/posttests can be found in Appendix B.

The qualitative module specific questionnaire contained 16 items. The first three items dealt with the learner’s background and comfort level with SQL. Six of the items asked the participant to rate the quality of the module on a scale of 1 to 7 with 1 representing a very negative opinion and 7 being the most positive. These items dealt with (a) ease of use, (b) quality of code examples, (c) clarity, (d) ability of the instruction to stand on its own, (e) level of enjoyment and (f) strength as a reference. The remaining items asked for recommendations for improving the module. The module-specific questionnaire is contained in Appendix C.

The SQLTips framework questionnaire asked questions concerning SQLTips as a reference and framework. There were five questions in the questionnaire. The first two were again on a scale of 1 to 7 asking, (a) how likely are you to use SQLTips as a

reference and (b) how likely are you to add your own insights into the material. The last three questions asked if the format of SQLTips is worth using for other OIT-E instruction, and what topics, if any, would the participant like to see in SQLTips format. The framework questionnaire is contained in Appendix D.

Procedures. The participants were recruited by the designer by walking through several OIT-E offices and speaking first with the chief engineer of a team, and then after obtaining his approval, asking the team members if they would like to participate. The participants were instructed as to the nature of the instruction, the estimated time it would take to complete and the various instruments that would be used to determine SQLTips effectiveness. Over a period of several days, as the modules became available, module packets were handed out to the participants. Each packet contained the instruments for one module, including (a) a pretest, (b) a posttest, and (c) a module qualitative questionnaire. These were stapled together and the participants were asked to not put their names on the packets. Each participant was also sent an email explaining the procedure of filling out the pretest, going through the instructional material, and then taking the posttest and filling out the qualitative questionnaire. The participants were also instructed that the tests were “closed book and closed friend.” They were also given a location to drop off the packets upon completion. Efforts were made to assure the participants that I did not want to be able to identify any packets with the individual that turned it in. This was to minimize any test related stress that the participants may have felt and to also encourage honest responses to the qualitative questions.

Quantitative results and discussion. Through a two-tailed dependent pairs *t*-test analysis of the pretests and posttests, SQLTips was shown to provide effective

instruction. The results from each module were analyzed separately, and all but two showed statistical significance. A summary of the scores are located in table 1. The mean pretest percent correct for all the modules was 42% with a low of 6% for module 5 and a high of 75% correct for module 1. The mean percent correct on the posttests was 87% with a low of 56% on module 5 and a high of 100% correct on module 6. The mean of the differences between the pretests and posttests was 45.8%.

Table 1

Mean Percentage Score by Module and Test Occasion

Module	n	Pretest		Posttest		Mean Difference	t	p
		M	SD	M	SD			
1	8	75.0	11.5	97.2	5.1	22.2	6.11	0.00
2	6	64.8	8.4	90.7	8.4	25.9	11.07	0.00
3	5	66.0	11.4	98.0	4.5	32.0	6.53	0.00
4	4	46.4	35.7	89.3	13.7	42.9	3.67	0.03
5	6	8.3	13.9	75.0	27.4	66.7	5.86	0.00
6	5	30.0	34.2	100.0	0.0	70.0	4.58	0.01
7	4	39.6	30.0	70.8	30.8	31.3	15.00	0.00
8	5	60.0	22.4	92.5	11.2	32.5	2.65	0.06
9	2	20.0	0.0	75.0	21.2	55.0	3.67	0.17
10	4	20.0	16.3	100.0	0.0	80.0	9.80	0.00
Means	4.9	43.0	18.4	88.9	12.2	45.8		

Qualitative results and discussion. The qualitative results were determined from analysis of two different types of questionnaires, the module-specific questionnaires and the SQLTips framework questionnaire. Both of these types were described above.

The questions on the module specific questionnaires could be placed into three specific categories; the first (composed of the first 3 questions) attempts to determine the background of the user in regards to SQL and were used for analysis of the participants rather than qualitative analysis of the instruction. The next category dealt with rating the

instruction in several areas based on a scale of 1 to 7 with 1 being the most negative and 7 being the most positive. The results from this category are contained in Table 2.

Table 2

Mean Rating of Modules

Module	<i>n</i>	Ease of Use	Quality of Code Examples	Clarity	Stand on Its own	Enjoyability	Value as a Reference
1	8	6.7	6.5	6.5	6.1	6.4	4.8
2	6	6.5	6.7	6.3	6.2	6.3	4.8
3	5	6.2	6.2	6.2	6.2	6.4	5.4
4	3	5.7	6.3	6.0	5.0	6.0	6.0
5	2	6.5	7.0	7.0	7.0	7.0	7.0
6	3	7.0	7.0	7.0	6.3	6.3	6.0
7	3	4.7	6.0	6.0	6.0	5.7	6.7
8	2	6.5	7.0	6.5	7.0	7.0	5.5
9	2	6.5	6.5	6.0	5.0	6.0	6.5
10	2	5.0	5.0	5.5	5.0	4.5	5.5
Means		6.1	6.4	6.3	6.0	6.2	5.8

The next category of questions on the module specific questionnaires provided the participants an opportunity to make suggestions for improving the instruction and to give other free-response feedback. The responses for this part of the questionnaire revolved around (a) usability, (b) the quality of code examples, (c) clarity, (d) the ability of the module to stand on its own, (e) any deficiencies and (f) which part was most enjoyable or valuable. For ease of use the overall average was 6.1 and for enjoyability it was 6.2. The overall averages across all categories were 5.8 and higher.

Some of the comments showed problems with the format. For module 7, one participant wrote, “Give an example of a function explaining its arguments.” In module 7 each of the function names contains a hyperlink to an example of the function in use and instructions are provided to “Click on a function name to see an example.” As with the

other hyperlinks on OIT-E's Wiki, these links are colored orange. I wonder if the participant in this case, missed that instruction, and failed to recognize the hyperlinks due to their non-standard format. Another participant commented in the "Stand on Its Own" category that definitions were not provided for DDL and DML. This comment points to a possible benefit of adding an online glossary with hyperlinked access from the modules. Another participant commented in the "Stand on Its Own" category for module 4, that the pretest "questions 4-7 were not covered at all." In actuality, the material was covered, but it was covered in the one of the "Click HERE for results and explanations" pages. This illustrates a flaw in the modules lesson flow. Should the learner be required to click on a results page to get critical information? I think the modules would be better if all critical information was put on the main page of the instruction, rather than buried in a hyperlinked (optional) page.

One of the questions on the questionnaire asked if the number of examples was sufficient or if there should have been more or less. About half of the participants thought there should be more, about half thought it was okay the way it was. None of the respondents felt there should be less.

The framework questionnaire had five items on it. The first three were (a) how likely are you to use SQLTips as a reference, (b) how likely are you to add or modify the material with your own insights, (c) do you think the format of SQLTips is worth using for other instructional products for OIT-E. The last two items dealt asked for suggestions for future topics. For item 1, dealing with the likelihood of using SQLTips as a reference, the average response was a 6.0 on a scale of 1 to 7 with seven being the most likely. For item 2, dealing with the likelihood of adding to the instructional material the

average response was a 3.7 on the same 7-point scale, showing a reluctance to add to the instruction. As for item 3, all of the responses were positive as to the format of SQLTips.

Conclusion

Critique

In this section I will provide some of the positive and negative aspects of both the SQLTips product and the process I went through in developing it. I will also provide some of the insights I've gained through the experience.

Product weaknesses. Module 1 included some material that was too basic for the audience. The pretest scores were high for that module and overall, I suspect that much of it was a waste of the reviewers' time.

The instructional designer sometimes has a problem with being too succinct. He doesn't always explain things completely. This problem was caught on many occasions by the reviewers but I suspect there are still many examples in the modules where a few more words might help. In some cases the module specific questionnaires point to this fact.

Although SQLTips lists some prerequisites reading and courses, most of it requires a background in using SQL. Having gone through just the prerequisite material does not prepare someone for the most of material that's covered in the SQLTips modules. This became apparent while reviewing the comments in the module specific questionnaires.

Product strengths. Overall I feel that SQLTips was highly successful in accomplishing its objectives. Posttest scores showed a strong average improvement and the qualitative results were high. In my opinion, the best strengths of SQLTips is (a) it's

easy to update, (b) it's readily accessible by the developers as a reference and (c) it has some really good example code.

Process weaknesses. There was a pretty heavy time commitment for all involved with the review process. Developing ten modules using this process was perhaps a bit optimistic. Both of the SMEs involved in the review process were very concerned with doing a thorough job and neither had the time to do so. I had to remind both of them to ignore typos; their primary concerns were (a) making sure there were no inaccuracies, and (b) checking to see if the SLOs were being adequately addressed. Most of the typos were corrected in the one-to-one evaluations.

Both of the one-to-one evaluators were student interns of OIT-E. Although both were majoring in computer-related fields, neither had a lot of SQL experience. On several occasions the review sessions would turn into mentoring sessions. Reviewers with more SQL experience were not available for that great of a time commitment.

The small group evaluators were picked from the intended audience pool based primarily, on availability and interest. Almost all of them were full-time employees with more than ten years professional experience. I wish that I had been able to get more student interns so that the small group would represent the intended audience better, but was unable to do so. Another problem with the make-up of the small group was that it was based a lot on interest level of the participants. Because of the inherent interest of the participants, I believe that the ratings (discussed later) may have been inflated at the positive end of the spectrum—participants who enjoy SQL are more apt to rate instruction on SQL positively, if it provides them with a few new ideas. Because of the heavy time commitment required of the field test participants, the “n” value gradually

dropped starting with module 1. Ten learners agreed to participate, but only nine turned in the pretest/posttest for module 1. Each successive module had fewer participants. For modules 6 through 10, I recruited more, but the SQL experience level for the later recruits was not as high and in several comments in the module-specific questionnaires, it was mentioned that the material was too difficult. Several participants turned in blank posttests.

Process strengths. I am sold on the review process. I had some really good reviewers. They spotted a lot of problems with the instruction before it went out for the field test. Although I struggled with it, the literature review proved very helpful as a reminder of instructional design concepts I had previously learned in Dr. Osguthorpe's class.

Other insights. I found it somewhat humbling to have my work so soundly criticized by the reviewers. I was glad that I put an item in the module-specific questionnaire asking "What did you enjoy most about the module?" If it wasn't for that item, it would have been depressing to read through the filled-out questionnaires.

Schedule

In this section I discuss the schedule for the project. One of the stakeholders, a chief engineer for one of the development teams, wanted the project done by mid November for training the new employees on his team prior to beginning a new project. The other stakeholders did not have any firm deadlines. Outside of the stakeholders, I just needed to work within the semester deadlines. The projected delivery dates in Table 3 reflect a misunderstanding I had of the semester deadlines. In most cases I met the deadlines and in some cases accomplished them ahead of schedule.

Table 3

Estimated and Actual Schedule for SQLTips

Milestones	Projected Delivery	Actual Delivery
Final draft of objectives and methods	Aug 13	Aug 19
Assessment instruments written	Aug 20	Aug 27
Module 1 completed	Aug 27	Sep 10
Module 2 completed	Sep 3	Sep 13
Module 3 completed	Sep 10	Sep 17
Module 4 completed	Sep 17	Sep 21
Module 5 completed	Sep 24	Sep 27
Small group evaluations (M1–M5)	Oct 1	Oct 24
Module 6 completed	Oct 8	Sep 28
Module 7 completed	Oct 15	Oct 1
Module 8 completed	Oct 22	Oct 5
Module 9 completed	Oct 29	Oct 6
Module 10 completed	Nov5	Oct 6
Small group evaluations (M6-M10)	Nov12	Oct 24
Final report	Dec 3	Oct 27

Budget

All of the individuals involved with the planning, creation, and evaluation of this project are employees of the sponsor. With the exception of the designer's time, the personnel costs were considered primarily a training expense, as evaluations were considered a training exercise. All personnel rates were approximated based on the very limited knowledge of the designer. The designer's rate was also approximated loosely, based on hourly wages for the billed-time he spent on the project averaged in with the non-billed hours he spent. There was no travel involved, and the project didn't add anything additional to existing phone costs. The bulk of the communication was accomplished in person and through email. The small group evaluations were done on paper. The other evaluations were done either orally or through electronic means. Table 4 shows a breakdown of estimated and actual costs.

Table 4

Projected and Actual Expenses for SQLTips

Resource	Rate	Estimated Quantity	Actual Quantity	Estimated Cost	Actual cost
Designer	\$15/hr	262	270	\$3930	\$4050
SME	\$25/hr	4	6	\$100	\$150
One-to-one	\$15/hr	20	15	\$300	\$225
Small group	\$15/hr	24	18	\$360	\$270
Photocopies	\$0.05/copy	180	280	\$9	\$14
Total				\$4699	\$4709

References

- Corey, M. J., Abbey, M., & Dechichio, D. J. Jr. (1995). *Tuning Oracle*. Berkeley, CA: McGraw-Hill.
- Dick, W., & Carey, L. (1996). *The systematic design of instruction (4th ed.)*. New York, NY: Harper Collins.
- Dawes, C., & Thomas, B. (2002). *OCA/OCP: Introduction to Oracle9i SQL study guide*. Alameda, CA: SYBEX.
- Flagg, B. (1990). *Formative evaluation for educational technologies*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, Publishers.
- Gagne, R. M., Briggs, L. J., & Wager, W. W. (1988). *Principles of instructional design (3rd ed.)*. New York, NY: Holt, Reinhart and Winston, Inc.
- Gurry, M. (2002). *Oracle SQL tuning: Pocket reference*. Sebastopol, CA: O'Reilly & Associates.
- Harrison, G. (2001). *Oracle SQL: High performance tuning (2nd ed.)*. Upper Saddle River, New Jersey: Prentice Hall.
- Jakovljevic, M. (2003). Concept mapping and appropriate instructional strategies in promoting programming skills of holistic learners. *Proceedings of SAICSIT 2003*, (pp. 308-315).
- Jenkins, T. (2001). Teaching Programming—A Journey from Teacher to Motivator. *2nd Annual LTSN-ICS Conference, London*.
- Johnson, J. C. (2002). *OCP: Oracle9i performance tuning study guide*. Alameda, CA: SYBEX.

- Keirns, J. L. (1999). *Designs for self-instruction: principles, processes, and issues in developing self-directed learning*. Allyn & Bacon.
- Kreines, D. C. (2000). *Oracle SQL: the essential reference*. Sebastopol, CA: O'Reilly & Associates.
- Lewis, B., Dean, C., Whitlock, Q., & Bryson, M. (1987). *Computer based training*. New Jersey, Parthenon Publishing.
- Mishra, S., & Beaulieu, A. (2002). *Mastering Oracle SQL*. Sebastopol, CA: O'Reilly & Associates.
- Osguthorpe, R. T. (1985). Conducting literature searches for instructional development projects. *Journal of instructional development*, 8(4), 20-24.
- Russel, C., & Cordingley, R. (2003). *Oracle DBA SQL: Quick reference*. Upper Saddle River, New Jersey: Prentice Hall.
- Simon, H. (2001). Cooperation between educational technology and learning theory to advance higher education. In P. S. Goodman (Ed.), *Technology enhanced learning: Opportunities for change* (pp. 61-74). Mahwah, New Jersey: Lawrence Erlbaum Associates, Publishers.

Appendix A

SQLTips General Instructional Objectives and Specific Learning Outcomes

1. Understand key points about OBJECTS and SCHEMAS
 - 1.1 Describe the purpose of a database table and its structure.
 - 1.2 Describe the purpose of an index and its structure
 - 1.3 Define a schema
 - 1.4 Describe the purpose of a sequence
 - 1.5 Explain the difference between a private and public synonym
 - 1.6 Define a TABLESPACE
2. Understand key points about indexes
 - 2.1 List the two primary purposes of indexes
 - 2.2. Describe the costs of indexes
 - 2.3 Define rowid
 - 2.4 Explain the differences between simple and concatenated indexes
 - 2.5 Describe the reasoning behind column order in a concatenated index
 - 2.6 Determine what indexes exist for a given table
 - 2.7 List three types of indexes and the conditions for their optimal use
 - 2.8 Explain why an index and table are usually in different tablespaces.
3. Understand datatypes
 - 3.1 Describe the following datatypes:
 - a) char, varchar2, nchar, and nvarchar2
 - b) number
 - e) date,, timestamp, and interval

- f) long, long raw, clob, blob, and bfile
- 3.2 Explain the “gotchas” for DATE datatypes
- 3.3 Explain the optimizer consideration of datatype use
- 3.4 Describe when you would long rather than CLOB
- 4. Understand how to create database objects well
 - 4.1 Explain what a primary key is
 - 4.2 Explain what a foreign key is
 - 4.3 Create a table and index according to our standards
 - 4.4 Describe where to find syntax examples
- 5. Understand new JOIN syntax
 - 5.1 Explain the advantages to the new join syntax
 - 5.2 Convert a query with WHERE join to JOIN clause syntax
 - 5.3 Convert an outer join query to the new syntax
 - 5.4 Explain pre-compiler issues with outer join
 - 5.5 Convert a cross join query to the new syntax
 - 5.6 Explain the difference between the ON and USING clauses
- 6. Understand the data dictionary
 - 6.1 Describe the difference between V\$ views and static views
 - 6.2 Describe the difference between DBA, ALL and USER views
 - 6.3 Explain how to find the names of dictionary views
 - 6.3 Explain two examples that illustrate the advantages of querying the data dictionary vs. using GUI tools

7. Understand key functions and constructs

7.1 Explain the use of the DECODE function

7.2 Write a simple CASE statement and a searched CASE statement

7.3 Write a query containing the following functions

a) substr

b) instr

c) replace

d) to_char

e) ascii

f) length

g) ceil

i) floor

j) nvl

7.4 Explain where to find Oracle's documentation on functions

8. Analyze the use of indexes in poorly performing SQL statements

8.1 Determine which indexes exist for a table

8.2 Determine if new indexes may help

8.3 Analyze join statement for full key joining

8.4 Determine if optimizer statistics are fresh

8.5 Determine if indexes are being used

8.6 Analyze SELECT for adding columns to existing index

9. Rewrite poorly performing SQL statements

9.1 Explain the benefits and drawbacks of correlated sub-queries

9.2 Convert a “not in” to a “not exists”

9.3 Convert statements that can't use indexes to ones that can

10. Understand how to use PL/SQL

10.1 Write a simple PL/SQL block

10.2 Write a simple trigger

10.3 Write a function

10.4 Write a procedure

10.5 List two advantages to using PL/SQL over SQL

10.6 State the policy against business logic in the database

Appendix B

Pretests/Posttest

OBJECTS and SCHEMAS

1. What is the primary purpose of a database table?

2. When a record (or row) is inserted into a standard table, is it sorted in any particular order based on the values of the columns?

3. What are the two primary purposes of indexes?

4. What name is used to describe the structure of an index?

5. Can two tables in the same database have the same name?

Explain your answer :

6. What is a sequence used for?

7. What is a synonym?

8. What is a public synonym?

9. What is a tablespace?

Indexes

1. What are the two primary purposes of indexes?
2. What are the primary costs of indexes?
3. What is a rowid?
4. Fill in the blank:

```
create unique index sysadm.empl_address_a
on sysadm.empl_address (emplid, address_type)
_____ psindex;
```

5. Given the following WHERE clause, pick which index(es) you would build:

WHERE person_id = :b and address_type = 'WORK'

- A) Two indexes: one on PERSON_ID and one on ADDRESS_TYPE
- B) One index on (PERSON_ID, ADDRESS_TYPE)
- C) All three indexes listed above

Explain your choice.

6. Given the following :

WHERE Clause	Columns in index
where person_id = :b	(address_type, person_id)

Is it likely the index will be used?

Explain your reasoning.

7. What tool would you personally use to determine what indexes exist for a given table?

8. Outside of the standard B-Tree index, name two other types and the optimal conditions for their use.
9. Why is an index usually stored in a different tablespace than the table it references?

Creating Database Objects

1. What are the two characteristics of a primary key?
2. Should all tables have a primary key? Explain?
3. What is a foreign key constraint?
4. How would you insert records when there is a foreign key constraint?
5. How would you delete them?
6. You want to create a table; where would you find syntax examples for following our standards?
7. Where would you find the syntax for creating a sequence?

Datatypes

What datatype would you use to store the following:

1. Grade points _____
2. Grade _____
3. JPEG file _____
4. A person's last name _____
5. Updated time _____
6. An external file locator _____
7. Chinese characters _____

8. Explain the results from the second query shown below:

```
SQL> select sysdate from dual;

SYSDATE
-----
05-SEP-05

SQL> select sysdate from dual where sysdate = '05-SEP-05';

no rows selected
```

9. Why is the following statement false?

“It’s a good idea to set all VARCHAR2s to the maximum value for greatest flexibility.”

10. When would you use a long rather than a CLOB?

ANSI standard Join Syntax

Rewrite each statement using the new (to Oracle) ANSI standard

1.

```
Select a.sort_name, b.state, c.phone  
From person a, address b, phone c  
Where a.person_id = b.person_id  
And a.person_id = c.person_id;
```

2.

```
Select p.sort_name, s.curriculum_id  
From person p, std_classes s  
Where p.person_id = s.person_id (+)  
And s.year_term (+) = '20045';
```

3.

```
Select a.name, b.table_name  
From v$database a, dba_tables b  
Where b.owner = 'PRO';
```

4. Name two advantages to the new syntax?

5. Name any disadvantages to the new syntax.

6. What is the difference between the ON and USING clauses?

The Data Dictionary

Next to each prefix, describe its purpose or what it displays:

1. V\$ views _____

2. DBA_ views _____

3. ALL_ views _____

4. USER_ views _____

5. What view would you query to find other dictionary views?

6. List two examples that illustrate an advantage of querying the dictionary vs. using a GUI tool.

Functions

1. What will the query below return? _____

```
select decode
(
  to_char(sysdate, 'yyyy')
  , '2005' , 'This year'
  , dummy
)
from dual;
/
```

2. Fill in the function names in the table below

Argument 1	Argument 2	Argument 3	Result	Function
3.14	None	None	4	
1.7532	None	None	1	
Null	'x'	None	'x'	
'Vincent'	1	5	'Vince'	
'Rackliffe'	'k'	None	4	
'ORACLE'	'OR'	'SPECT'	'SPECTACLE'	
SYSDATE	'yyyy'	None	'2005'	
' '	None	None	32	
'NUMBER'	None	None	6	

3. YES/NO Have you ever looked at the Oracle documentation on FUNCTIONS?

4. What functions would you use to do the following:

a) return a 4 for a 'A', a 3 for 'B', a 2 for a 'C', etc. . _____

b) return an 'A' for 3.5-4.0, a 'B' for 2.75-3.49, etc. _____

Query Optimization – Index Use

For the following statements list what indexes you would hope to find. List the indexes in the following format

<table_name>(<column_name>[, <column_name>...]) ie, for the address table:
ADDRESS(person_id, address_type)

1.

```
SELECT count(*)
FROM PSPRCSQUE R, PSPRCSPARMS P
WHERE R.PRC SINSTANCE = P.PRC SINSTANCE;
```

2.

```
SELECT count(*)
FROM PSPRCSQUE R, PSPRCSPARMS P
WHERE R.RUNDTM <= SYSDATE
AND R.PRC SINSTANCE = P.PRC SINSTANCE;
```

3.

```
SELECT R.RUNSTATUS, COUNT(*)
FROM PSPRCSQUE R, PSPRCSPARMS P
WHERE R.RUNDTM <= SYSDATE
AND R.PRC SINSTANCE = P.PRC SINSTANCE;
```

4. Based on the info below, write a JOIN clause for ps_jrnl_header and ps_jrnl_ln

INDEX_NAME	UNIQUENES	COLUMN_NAME
PS_JRNL_HEADER	UNIQUE	JOURNAL_ID BUSINESS_UNIT JOURNAL_DATE UNPOST_SEQ
PS_JRNL_LN	UNIQUE	JOURNAL_ID BUSINESS_UNIT JOURNAL_DATE UNPOST_SEQ JOURNAL_LINE LEDGER

5. How would you determine what indexes exist for a given table?
6. How can one determine if statistics are fresh?
7. How would you determine if an index is being used?
8. What is the advantage to having all the columns in the SELECT clause in the same index as those of the WHERE clause.

Rewriting Queries

```
SQL> select count(*) from xxx;
```

```
  COUNT(*)  
-----  
         10
```

```
SQL> select count(*) from yyy;
```

```
  COUNT(*)  
-----  
         5
```

```
SQL>  select a.col_1  
2     from yyy a  
3     where a.col_1 in (  
4         select b.col_1  
5         from xxx b  
6         where b.col_1 = a.col_1  
7     )  
8     /
```

1. How many times will the inner query execute:

- A) 1
- B) 5
- C) 10

2. Defend your answer to question 1

3. Convert the following to a “not exists”

```
SQL> select a.col_1
2   from yyy a
3   where a.col_1 not in (
4       select b.col_1
5       from xxx b
6   )
7   /
```

For the following WHERE clauses assume there is a standard index on the columns listed. However in each case, due to the way the WHERE clause is written, the index will not be used. Rewrite the WHERE clause so the index is usable:

4. WHERE credit_hours != 0
5. WHERE substr(c1, 1, 3) = 'xxx'
6. WHERE trunc(date_of_birth) = '28-MAY-59'
7. WHERE dept_name||catalog_number = 'ENGL101'
8. WHERE c1 + 300 > :b
9. WHERE c1 = nvl(:b, c1)
10. WHERE person_id = 911206182

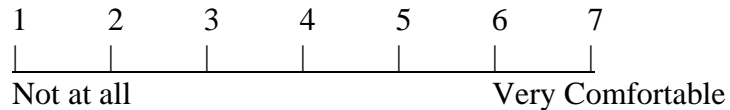
PL/SQL

1. Write a simple PL/SQL block
2. How would you store your block in the database?
3. In one (or two) sentence(s), what is a trigger?
4. What is our policy on business logic “stored in the database”?
5. List two advantages of using PL/SQL over SQL?

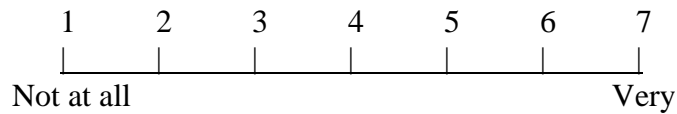
Appendix C

Qualitative Questionnaire for Small Group Evaluation of Module

1. Have you gone through SQLcourse1.com and SQLcourse2.com? (Y/N)
2. Have you gone through Crowther and Buck's SQL tutorial? (Y/N)
3. On a scale of 1 – 7, how comfortable are you with SQL?

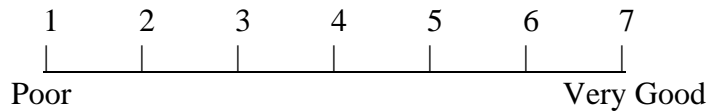


4. How easy was the module to use?



5. Suggestions, if any, for improving its usability:

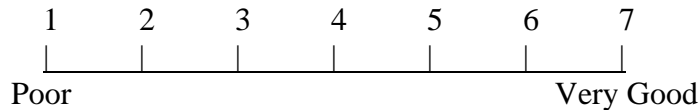
6. How would you rate the quality of the code examples?



7. Suggestions, if any, for improving the code examples:

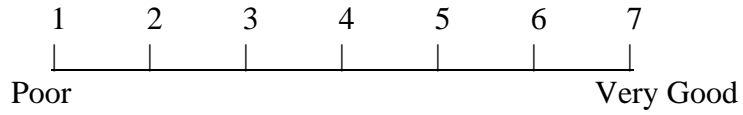
8. Would you like more code examples, less, or is it okay the way it is?

9. How would you rate the module for clarity?



10. Suggestions, if any, for improving the clarity:

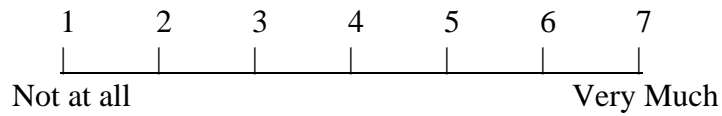
11. How would you rate the module on its ability to “stand on its own”, ie, as a self-instructional tool, with no input from an instructor?



12. Suggestions, if any, to help it “stand on its own”:

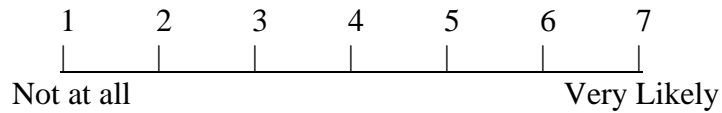
13. Specifically, which part(s) of the module, if any, need work?

14. Did you enjoy going through the module?



15. Which part(s) of the module did you find the most enjoyable or most valuable?

16. How likely are you to use this module as a future reference?



Appendix D

Qualitative Evaluation Form for SQLTips

1. How likely are you to use SQLTips as a future reference?

1	2	3	4	5	6	7

Not at all						Very Likely

2. Seeing as SQLTips is Wiki based, how likely are you to add to or modify the material with your own insights?

1	2	3	4	5	6	7

Not at all						Very Likely

3. Do you think the format of SQLTips is worth using for other instructional products for OIT-E?

4. What other, if any, SQL topics would you like to see in SQLTips?

5. What other, if any, work related topics would you like to see in this format?

Appendix E

Log of Qualitative Responses to Module Specific Questionnaires

Module	Comments
Module 1	<p>criticisms and suggestions</p> <p>Seems cryptic. Of course it depends on the target audience.</p> <p>Several places say "Click here for results and explanations", but there are no explanations</p> <p>I'm not sure what question #4 means.</p> <p>I'm still not sure what the difference between tablespace and schema is and how that affects table with same name.</p> <p>More examples--try to illustrate different types of queries</p> <p>More explanations</p> <p>Have the results of the queries appear to the right of the query, pop up, or have hover (title) text.</p> <p>tablespace needs more info about what it is, what it does, and why we should know about it.</p> <p>It would be nice to have a lot of useful samples in case we forget.</p> <p>Needs more on indexes.</p> <p>Example of how a secondary index can improve performance</p> <p>Separate definitions in some way (font, color, etc).</p> <p>What they enjoyed</p> <p>synonyms</p> <p>I can look here to remember how to see which fields are in an index because I always forget.</p> <p>The DBA_ object stuff was useful.</p> <p>I liked the simple clear sections.</p> <p>The organization is very good</p>
Module 2	<p>criticisms and suggestions</p> <p>The test asked about rowids but there was nothing about them in the lesson.</p> <p>More detail if you want to read it.</p> <p>It would be interesting to see examples of real BYU problems and how adding or changing a particular index helped and why. I don't see that very often.</p> <p>What they enjoyed</p> <p>the composite index explanation</p> <p>learning about types of indexes</p> <p>explanation of types of indexes and their suggested uses.</p> <p>the explanation of concatenated indexes</p>
Module 3	<p>criticisms and suggestions</p> <p>a whole tutorial on the difference between null, space and all spaces--SQL inserts and updates are not consistent in apps.</p> <p>more info on each type</p> <p>What they enjoyed</p> <p>I like the examples from our own tables that we work with all the time.</p> <p>the description of each of the commonly used datatypes</p>
Module 4	<p>criticisms and suggestions</p> <p>questions 4 - 7 are not covered at all</p>

the documentation is missing "how to create a sequence."
 need explanation on inserting and deleting parent/child records where standards exist

What they enjoyed

DMAT standards
 great the way it is
 the definitions of primary keys and foreign keys and when to use.

Module 5

criticisms and suggestions

Last code example contained a query of a non-existent table. I wanted to see the query run!

What they enjoyed

New syntax - CASE
 Instruction on the ON clause

Module 6

criticisms and suggestions

maybe a link to a "more useful scripts" page

What they enjoyed

the entire module was a good review
 good as is
 The practical examples like finding missing indexes
 understanding the differences between DBA_, ALL_, USER_ views.
 the clever code examples

Module 7

criticisms and suggestions

there is a missing piece in the coalesce function description and the summary is missing
 give an example of a function explaining its arguments
 more examples from easy to complex
 add a "last year" and "next year" to the examples
 there are a few wrinkles that need smoothing

What they enjoyed

the searched CASE explanation
 finding a reference for functions

Module 8

criticisms and suggestions

use less or combine examples
 I notice a redundancy in the creation of your unique indexes and subsequent primary key constraints.

What they enjoyed

How to check if stats were run, seeing if indexes are used
 practical application of building queries to use indexes.
 use of indexes to optimize queries
 viewing the changes that indexes and statistics make to performance

Module 9

criticisms and suggestions

The "not exists" section needs more detailed explanation.

What they enjoyed

I loved seeing the comparisons
 WHERE clauses

Module 9

criticisms and suggestions

SQL examples were way over my head
since I don't know what DDL and DML are, I didn't understand the comment about output.
for some reason I did not have permission to create a trigger - insufficient privilege error.

What they enjoyed

I think its good

learning how to write procedures

Appendix F

JSPWiki: SQLTips - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://engwiki.byu.edu/ry/webapp/wiki/Wiki.jsp?page=SQLTips

BYU · IT · ENG

SQLTips

History: [Main](#) | [Vbr3](#)

[Edit This Page](#) | [More Information](#) | [Attach Files](#) | [Printer Friendly Version](#)

SQLTips

SQLTips is a tutorial/reference for advanced SQL. The modules in SQLTips presume a prerequisite background in basic SQL. This background can be obtained via the following resources:

- [Crowther and Buck's Training Manual](#) (chapter 11)
- [sqlcourse.com](#)
- [sqlcourse2.com](#)

All of the examples and exercises in SQLTips are done through SQL*PLUS. I strongly recommend that you review the following module prior to beginning SQLTips.

- [Using SQL*PLUS](#)

SQLTips Modules

1. Tablespaces, Schemas and Database Objects
2. Understanding Indexes
3. Data Types
4. Creating TABLES, INDEXES and KEYS
5. New ANSI Join Syntax
6. The Data Dictionary
7. Functions and Case Statements
8. Query Optimization - Index Use
9. Query Optimization 2
10. PL/SQL

Resources

[Crowther and Buck's Training Manual](#)

[sqlcourse.com](#)

[sqlcourse2.com](#)

[9i documentation](#)

[Using SQL*PLUS](#)

This page last changed on 20-Oct-2005 12:21:49 MDT by Vbr3.

[Edit This Page](#) [More Information](#) [Attach Files](#)

Done [engwiki.byu.edu](#)

Appendix G

Log of Input from SME and One-to-one Evaluators

 *** INTRO

OTO1: 8-29 3:43 - 4:07
 Exclamation mark in SQLHe!p is a typo
 Doen't know what monospaced font is
 Click for result confusing
 typos: States - stats
 autotrace mentioned before definition
 "spool off" should have quotes
 ED should mention blank line.

OTO2: 8/30 2:55 - 3:09
 Numbering at top not clear
 typos
 What is monospaced font
 comparing two queries - run it twice

*** Module 1 - Objects

SME1:
 Didn't note that VIEWS were discussed.
 Might want to explain that some views can be used to update the underlying table
 Good high level intro.

SME2:
 There were some minor misspellings and missing words, but overall it was very accurate

All of the objectives were met, but the information in the lesson covered more areas than was mentioned in the Intro. Views and synonyms were not mentioned in the Intro

Some kind of diagram in the intro identifying how each of the objects relate to each other would be helpful.

I really liked the format that was used to present the lesson material. The option to click to see the results of the queries was very well done.

OTO1: 9/22 4:10-4:35
 More clear on INDEXES, "you want to know the page number" of what?
 tablespaces_04 - explanation awkward
 tablespaces_05 - add "indexes hide complexity"
 tablespace_06 - sequences need to be initialized

OTO2: 9/26 11:00 - 11:15

Tablespaces - mention error but don't show it - should show it.

Left out mention of schemas in "attention" questions at top

Could use a diagram showing relationships

need explanation on sequence results.

 *** Module 2 - Understanding INdexas

 SME1:

Better definition of rowid: explicitly define it.

Under function based index, a better explanation of the examples of when the index will and will not be used: ie: some where clauses.

Explain the meaning of modified B-Tree statement. B-tree, binary-tree and m-star stuff.

Like the hymn example. Also good description of why tables and indexes in different tablespaces.

 SME2:

There were some minor misspellings and missing words, but overall it was very accurate.

When explaining how "indexes help enforce uniqueness", the first two sentences seem to leave you hanging. The last two sentences of the paragraph don't really act as a clear explanation of why you mentioned the information in the first two. In other words, how does sorting a flat file to find duplicate rows help you see why an insert fails due to a unique index? You might want to clarify the second sentence to say something like: "the database sorts the data in the unique index and uses it to see if the value to be inserted already exists."

You mention the optimizer in module 1 and 2. Although the name "optimizer" is some what descriptive itself, this lesson might be a good place to give a quick description of what it does.

Very good analogies.

 OTO1: 8-29, 4:07 - 5:37

Put is "SET echo on" in first script

Preface exercises with what to expect and look for on *_02 typo

Explain separate index concept better

What's the optimizer

Need a "TERMS" page?

explain "compute statistics"

*_03 bottom query IS using index

Results page so long forgot wasn't on main screen

Define function-based index before referencing it.

What is DML

Explain that overview of indexes is not supposed to be thorough

Link to ORACLE docs on bottom

 OTO2: 8/30 3:09 - 4:20

Explain how to read an execution plan

what is the optimizer

*_03 explain what I'm doing better before the "do this"

Explain concatenated indexes

How to create a function based index

How to create an index

Indexes have to be created

explain how to use sqlplus more/ afiedt.buf

wild card % with like operator

create an upper(sort_name) on my_person

define cardinality

define dml

 *** Module 3 - datatypes

 SME1:

Since this is being written for BYU use, I would further explain that we have just started using one application that uses the Unicode database settings. And that application, PeopleSoft, does not use the nvarchar2 datatype.

Would like to see a brief example showing the use of to_char with number datatype

 SME2:

Data Types:

1. Accuracy of content

Very accurate

2. Does it meet the stated objectives

Yes

3. Recommendations for improvement.

- Item #2 Under "Why not make everything VARCHAR2?" The reason could be clarified by indicating that, when using the other datatypes, coding is simplified because of the ability to use their built-in methods.

- "What datatypes are there?" - The results of the first example would be more readable if you had them type in a format statement for the "data_type" column that limits it to fewer displayable characters.

- Syntax error in "Character Datatypes" section for "varchar."

Second sentence, remove "is" between "which" and "there."

- Second example "results and explanations." Last bolded sentence.

You might want to indicate that the error occurred because of the length constraint in the char datatype.

- Instructions following the third example for "date and time datatypes." Items #2 and #3 appear to have been concatenated together.

- Under "Optimizer considerations." Item #2. What are you referring to when you say "for the same reason"?

 OTO1: 9/22 4:35 - 5:15

Datatypes_01 - add sqlplus format commands

Datatypes_02 - Better explanation on '*||c|*'

Datatypes_04 - Explain significance of results

Interval definition - what is MONTH TO YEAR?

Datatypes_09 - better explanation

Definition CLOB - why is it easier to use?

 OTO2: 9/28 4:05 - 5:10

Varchar2 - add stop bit idea

Explain DataTypes_04

 *** Module 4 - Creating Objects

 SME1 (2:30 - 2:45):

Possibly more explanation on Rowid and using it in a where clause

Like that you explained that PeopleSoft doesn't use primary and foreign keys and what they do use to get unique rows

 SME2:

Creating TABLES, INDEXES and KEYS:

1. Accuracy of content

Very accurate

2. Does it meet the stated objectives

Yes

3. Recommendations for improvement.

-- "What is a primary key?", third paragraph, first sentence.

You should identify which two examples you are referring to.

-- Footnote 4: Fix the word "fastes" to say "fastest." Also

"DBs" should be "DB's." And last of all, the last sentence appears to have been truncated or not completed.

-- #2, second set of examples, third bolded explanation, second sentence. Add "with" or "to" to the end of the phrase "associate it" at the end of the sentence.

 OTO1: 9/27 3:20 - 3:40:

"P" = Primary, "R" = Relation

 OTO2: 10-03 4:20-5:00:

Intelligent = Natural

non-intelligent = surrogate

 *** Module 5 - ANSI Join syntax

 SME1: (3:00 - 3:35)

Question – on joins with the using clause, don't the fields have to have and same name and datatype?

With Using clause of the joins – qualifying the column is never allowed? This is not clear to me in your example.

Can you use the Using clause of the join to join on multi columns?

What do you mean by 'equality', 1st line of the JOIN with the ON clause paragraph. And again, can you join on multi columns using the ON clause?

In the Note , just before Outer joins, list all 3 tables.

Love the commentary on what is good and back about the different joins.

Liked the practice of converting the old syntax (+) to the new syntax.

SME2:

New ANSI Join Syntax:

1. Accuracy of content
very accurate - great examples
2. Does it meet the stated objectives
yes

3. Recommendations for improvement.

-- "Joins with the USING clause", last sentence. This sentence needs clarification to indicate that you are referring to the third query (the second one following the label "--NEW:").

-- "Joins with the USING clause", second query. You might want to add a "set pagesize" to the query so that format of the student's results matches your results.

OTO1: 9/27 3:40 - 4:45

Grabber - Not FUN but more aesthically beautiful

Clean up tables script

What is a cartesian join

"NO CARD" missing in primary text

More and better examples

Explain the differences between the joins better.

OTO2:

*** Module 6 - Data Dictionary

SME1:

Inaccuracies: No.

Weaknesses:

Even though it is implicitly stated that dynamic views (V\$) are different that static views, I think it would be helpful to explain that dynamic views info is for the lifetime of the database since it was last started. Also – 6.1 lists V\$ views and the heading is dynamic views. I think the same description should be used.

Increase effectiveness:

Under dynamic views you show 2 V\$ views that you use, but should you explain that there more many more? And that posters exist with all these views?

Other:

Another example that I believe developers would find helpful is listing constraints associated with tables.

SME 2:

The Data Dictionary:

1. Accuracy of content

very accurate

2. Does it meet the stated objectives

yes

3. Recommendations for improvement.

-- "You want to find out which tables have a column like '%SSN%', query. You might want to add format statements to the query so that the students results are formatted like yours.

-- "You want to see which indexes are in production but not stage", note for the query. Add the word "have" between "must" and "identical."

OTO1: - 10/7 3:08 - 3:33

Fixed typos - Everything else okay

OTO2:

 *** Module 7 - Functions and CASE statement

SME1:

Inaccuracies:

The results/explanation for the first decode statement was just the statement, no results were shown.

Weaknesses:

none

Increase Effectiveness:

Would it make since to switch the order? Present the case statement first, then the decode? Most developers have already been exposed to a case statement logic, so it would be quicker for them to understand the case statement.

On the case statement area, the first results/explanations, would it be helpful to have some results be index – not just table?

On the functions, explain which are ansi standard and which one are oracle. Also add the trim function (developers use it a lot.)

Other:

Like the link to other Oracle functions

SME 2:

Module 7

Functions and Case Statements:

1. Accuracy of content
very accurate
2. Does it meet the stated objectives
yes
3. Recommendations for improvement.

-- "DECODE", results and explanation of first query. In your results you only show the query and not a sample of the output from the query. For previous queries you showed the output.

-- "The Simple CASE Statement", first paragraph, last sentence.

Change "your" to "you're" or "you are."

OTO1: 10/7 3:33 - 4:15

Group by Clause a bit confusing other than that good.

OTO2:

*** Module 8 - Query Optimization - Indexes

SME1:

On the explain page QueryOptimization_09 – didn't like the font. And it was different that the other explains.

Like the final wrap-up.

SME 2:

Query Optimization - Index Use:

1. Accuracy of content
very accurate
2. Does it meet the stated objectives
yes
3. Recommendations for improvement.

-- You might want to use a table with more data for the examples.

When I ran the example query for the first time I got the same timing of 00:00:00.01.

OTO1:

QO_02--explain results a bit better

OOPS! section -- Had to read it twice to understand it.

OTO2:

*** Module 9 - Query Optimization 2

SME1:

Poorly performance could be other things than the 3 items listed. I think this should be noted somewhere.

Where discussing autotrace – refer back to the previous module; just in case taking these modules out of sequence.

Liked the where clause examples. Very practical.

SME2:

OTO1:

Explain what a "reference to the outside" means in the example.
Stress the point that when you modify a column in a WHERE clause the underlying index can't be used.

Why did they come up with the "NOT IN" clause if it's so bad.

OTO2:

Why did they come up with the "NOT IN" clause if it's so bad.
Point out that "NOT EXISTS" is using a correlated subquery.
Mention that "--" is a comment

*** Module 10 - PL/SQL

SME1:

I have received multi error on compiles of procedures before, so also explain how to show a list of errors. I don't remember the syntax. OK... I see that show err returns multi errors. I don't remember what my issue was, but I had one.

Maybe a mention to if we get heavy into using PL/SQL, there are initora parms that need to be reviewed for better performance. So they (the developers), would need to coordinate with their friendly DBA.

Show that thing you wrote for reserved words. Very helpful with PL/SQL – I think.

SME2:

Looks good.

OTO1:

Explain what a cursor is.

OTO2:

Looks good.