**Advanced SQL and XML**

**CSC317 Database Systems II**

**Professor: Dr. Kurt Diesch**

**By Odiscious Dozier**

**Trident University**

**SLP 01 – Part A - Summary**

**Summary**

There were many challenges experienced in accomplishing the tasks in this assignment. After many hours of research and consulting with faculty, I was able to operate both the Oracle Express Edition as well as the Oracle Enterprise Edition and was able to successfully accomplish this portion of the SLP assignment. And I am very grateful to have a group of people that are able and ready to help students when they come across such circumstances.

Step one required the opening of a browser and signing into the Oracle database. After getting all the configuration issues worked out, this step was fairly straight forward. I did, however, have to input my own password and not ‘oracle’ – the password stated in the instructions. Once signed in, the joy I experienced was profound. I ended up taking a short break.

Step four provided a way to tell Oracle what ports to communicate over. The File Transfer Protocol and Hyper-Text Transfer Protocol were set, but not much of an explanation was provided as to what this means. Web monkey helped me understand the particulars of a web address more clearly. In particular, with respect to port numbers, “the port number will be attached to the end of the server’s Web address, separated by a colon; for example, [ftp.fakesite.org:21](ftp://ftp.fakesite.org:21)” (“FTP”, n.d.).

Steps five through ten, inclusive, were really great at showing how Oracle stores XML files in a database. Although I’m not sure how to generate my own XML files via code, I am excited to learn how to save them into a database and then be able to connect to that database from anywhere provided it’s connected to the internet.

After starting SQL Developer, the instructions in step 2 require creating a new database connection by inserting the given data into the associated fields within the graphical user interface (GUI) that is provided by SQL Developer. My version of Oracle Database Express edition is listening on port 1521; therefore, I had to set my version of Oracle Database Enterprise edition to port 1522. I had experienced further struggles when clicking the save button. Oracle did retain the fields of information that I entered, but it did not retain the password in an XML file like task 2 b states.

In previous studies, I became familiar with SQL statements like, CREATE, UPDATE, and DELETE. I was also introduced to the idea of query times, but nothing deeper than suggestive arguments for query time considerations. Oracle offers reporting “on the execution path used by the SQL optimizer and the statement execution statistics” (“11 Using Autotrace”, n.d.).

Initially, when searching for the Autotrace button to click, in Oracle SQL Developer, I could not find it. Eventually I found the button, but it was subdued and not clickable. After doing a bit of online searching I found instructions that stated Autotrace was disabled by default. To enable it I executed the statement, set autotrace on.

Cardinality and cost are two Autotrace fields used in conjunction with each other that I found to be particularly interesting during this assignment. Cardinality defined as, “when tracing a statement in a parallel or distributed query, the execution plan shows the cost based optimizer estimates of the number of rows” (“11 Using Autotrace”, n.d.). These two fields and the object name were selected to be displayed, when a query is executed with Autotrace, during the configuration of the Autotrace and after some practice the resultant table became obvious, to me.

Creating XML type tables was a crucial connection in understanding, for me. I’ve come to realize that XML storage is very fast and useful for smaller quantities of data. What I did not know is the way to store data as XML in a database and how to query that data. The tutorial did not demonstrate this so, I searched Google and found information from Oracle’s document site that explained this concept and gave examples to perform in the ‘oe’ database (“Using XML”, n.d. ).

When looking to improve performance of XQuery expressions through index creation, I did not struggle with, finding the appropriate file, importing it, and executing the SQL statements. I did, however, not really understand what it was that I was doing. I didn’t know what an index was, in this sense, and how it works. After a quick online search I discovered that XML indexes are meant to decrease the execution time of a query. “When no indexes are present, an XML field must be ‘shredded’ when the query is executed. This means that the XML data is peeled apart from the XML tags, and organized into a relational format. An XML index does this work ahead of time, representing the XML data in an already-shredded version, thereby allowing easy filtering” (“Getting Started With XML Indexes”, n.d.).

An observation I made when using the XMLTable() function, in the last task, was that more fields were displayed in the resultant table than I initially observed. I originally did not notice that directly after the XMLTable() function, a variable can be given, in SQL, that will represent that table. Then, that table can be referenced with the variable name and all the columns from it can be accessed with the ‘\*’ symbol.

REFERENCES

11 Using Autotrace in SQL\*Plus. (n.d.). Retrieved February 09, 2016, from <http://docs.oracle.com/cd/B10500_01/server.920/a96533/autotrac.htm>

FTP for Beginners. (n.d.) Retrieved February 09, 2016, from <http://www.webmonkey.com/2010/02/ftp_for_beginners/>

Using XML in SQL Statements. (n.d.). Retrieved February 09, 2016, from <http://docs.oracle.com/cd/B10501_01/server.920/a96540/ap_examples3.htm>