**Advanced SQL and XML**

**CSC317 Database Systems II**

**Professor: Dr. Kurt Diesch**

**By Odiscious Dozier**

**Trident University**

**SLP 01 – Part B - Summary**

**Summary**

This assignment was very rewarding, for me. It was probably a lot more difficult than it was ever intended to be, but I finally got through it and I’m happy to share the ups and downs as I progressed through the assignment. This can be found in chapter 18 of Oracle’s XML DB Developer’s Guide – available via Oracle’s Help Center (“18 XMLType Views”, n.d.).

It took me weeks and a lot of reaching out to Trident’s faculty to get enough of the configuration issues sorted out to be able to complete this assignment. I’ve even had to request additional time to be able to learn many of the valuable concepts covered in Oracle’s extensive tutorial for this assignment.

As I started reading the tutorials provided via the links in the SLP 01 assignment, I was introduced to XQuery; which, was a real treat because I’ve heard about XML in databases but have never seen or used it. Noting that XQuery is case sensitive and must be in lower case format was something that I was glad to pick up along the way.

I also came across many new terms and acronyms. FLWOR, pronounced flower, is one such term and it refers to the basic **for-let-where-order by-return** expression syntax that is at the core of working with XQuery. “The let clause extracts three values using XPath notation (“Gennick”, 2005). The order by sorts the results similarly to SQL. The return generates a child <node> for each subset of the parent <node>. **XMLType** view is another term that I enjoyed learning about. While they wrap existing relational and object-relational data in XML formats, they do not store in the database in a way that is easy to read and understand XML; therefore, SQL Developer provides functionality to output XML to the resultant screen for human interpretation. I searched long and hard for a solution to render the results of my queries in such a format, but I was unsuccessful. I have reached out to faculty at Trident University and have high hopes that the answer to this puzzle will be given.

This SQL query creates the XML instances with the correct namespace, prefixes, and target schema location, and can be used as the query in the **emp\_simple\_xml** view definition. The instance created by this query looks like the following:

* ***SQLType for attributes or elements based on simpleType.*** This is compatible with the corresponding **XMLType**. For example, an XML string datatype can only be mapped to **VARCHAR2** or a Large Object (LOB) datatype.
* ***SQLType specified for elements based on complexType.*** This is either a LOB or an object type whose structure is compatible with the declaration of the **complexType**, that is, the object type has the right number of attributes with the right datatypes.

REFERENCES

18 XMLType Views. (n.d.). Retrieved February 12, 2016, from <http://docs.oracle.com/cd/B19306_01/appdev.102/b14259/xdb14vie.htm#i1026153>

Gennick, J. (2005, September). XQuery Flowers. Retrieved February 12, 2016, from <http://www.oracle.com/technetwork/issue-archive/2005/05-sep/o55xquery-097999.html>