**Chapter 6 – Team Project: Normalizing the Relational Model for the Team Project and Creating a Normalized Oracle Database**

Read the sample project steps for this chapter and apply the same techniques to the team project that you are developing. For the team project, do the following:

**Step 6.1 - Begin with the list of the tables that the entities and relationships from the E-R diagram mapped to naturally, from the sample project section at the end of chapter 4.**

For each table on the list, identify functional dependencies and normalize the relation to BCNF. Then decide whether the resulting tables should be implemented in that form. If not, explain why.

**Step 6.2 – Review and update the data dictionary and list of assumptions (as needed).**

**Step 6.3 - For each table, write the table name and write out the names, data types, and sizes of all the data items.**

Identify any constraints, using the conventions of the DBMS you will use for implementation.

**Step 6.4 - Design SQL statements to create all tables needed to implement the design. Then create the tables in the database.** Show your work by providing screenshots of executing the CREATE TABLE SQL statements in the database.

**Step 6.5 - Design SQL statements to create indexes for foreign keys and for any other columns that will be used most often for queries. Then execute the SQL statements in the database.** Show your work by providing screenshots of executing the SQL statements in the database.

**Step 6.6 - Design SQL statements to insert at least five records in each table, preserving all constraints.** **Then insert the records into the tables.** Show your work by providing screenshots of executing the INSERT SQL statements in the database.

**Step 6.7 - Design SQL statements that will process five non-routine requests for information from the database. Then execute the SQL statements in the database.** Show your work by providing screenshots of executing the SQL statements in the database along with the results.

Note: These five non-routine requests should be different from the ones you created in Chapter 5.

**Step 6.8 - Design one trigger for your project. Then create the trigger in the database.**

Create a trigger that will update an attribute on one table whenever a row is updated on another table. Show your work by providing screenshots of creating the trigger in the database.

Note: This trigger should be different from the one you created in Chapter 5.

**Step 6.9 - Design and execute SQL statements to demonstrate that the trigger is working as expected.** To demonstrate that the trigger is working as expected, provide a screenshot of the data before and after the trigger is executed.