Regis University SPS
MSCIT Program
Daniel Martin, Affiliate Professor
MSCD 650
Course Project: Database Development

Activities 1-5 should be submitted in one zip file at the end of the course. Each activity should be created in a separate folder, as described in each activity.

Activity #1
Creating the database schema and necessary installation scripts

Description:
You must produce an "installation kit" for your database project. You need to package the scripts, including the directory structure and files, according to the requirements.

Requirements:
1. The kit should include DDL script(s) to create your database schema, including all tables, constraints, views, indexes, etc.
2. The kit should include a script that will load sample data into your database tables.
3. The kit should include another script that will remove all data from your tables, so that the sample data can be reloaded.
4. All of your files should be organized into one folder called activity1 that matches the following directory structure:
   C:\MSCD650\<your_last _name>\activity1
5. The kit should include a readme.txt file to provide installation instructions. Also, briefly describe the nature of the database schema and its business purpose.

Deliverables:
Store the files in a folder named activity1 in the following directory structure: C:\MSCD650\<your_last _name>\activity1. The last step of the course project will be to package and turn in your deliverables in a windows zip file named <your_last_name>_mscd650_courseproject.zip.

Activity #2
Developing and implementing anonymous PL/SQL blocks

Description:
This activity has 2 parts. For each part you need to write an anonymous PL/SQL block that performs the given task or tasks. You must decide which table(s) and records to use based on your own project's schema.

Requirements:

- Part 1: write an anonymous PL/SQL block that loops through all the records of a table, and updates the value of one of the fields in each record. The field updated may be the same for each record; or it could be different depending on values in other fields of the record. Also, the value used to update the field may be the same for all records; or the value may be different depending on other fields of the record. In addition, when looping through the records, you should print out some sort of debug statement to keep track of what's going on in your code.

- Part 2: write an anonymous PL/SQL block that reads records from a master table (or a parent table), and inserts records in a detail table (or child table) for each master record. You don't need to insert detail records for all records in the master table, but your script should loop through at least 3 master records, and insert at least 1 detail record (can be more) for each master record. The data inserted may be the same for all records, except where you have primary key or unique constraints enabled.

Deliverables:
Package your scripts and any other files that are needed to run your scripts (including a readme) in a folder named activity2 (C:\MSCD650\<your_last _name>\activity2).

Turn in a .sql script file, which includes code comments, for each part. Your script files should be named <yourlastname>_act2_part1.sql and <yourlastname>_act2_part2.sql.

Activity #3
Developing and implementing PL/SQL stored programs – procedures and functions

Description:
This assignment has 4 parts. For each part you need to write a PL/SQL function or procedure that performs the given task or tasks. You will need to pick a master table and a corresponding detail table from your schema.

Requirements:

- Part 1: Create a stored function that determines the number of records that exist in a master table. Create a stored function that determines the number of records that exist in a detail table. The function should take, as arguments one or more search criteria to be applied to certain fields of the master table. Use the argument(s) to find the number of records matching the search criteria. The function should return the number of records found.

- Part 2: Create a stored procedure that inserts a record into a master table; and create a second stored procedure that inserts a record into a detail table. The procedures should take, as arguments any/all required values needed to insert the record. Use parameter defaults if necessary. The procedures should insert the record and end; they may display some debug statements where needed.

- Part 3: Create a stored procedure that deletes a record from a master table; and create a second procedure that deletes a record from a detail table. The procedures should take as arguments the primary key value of the record to be deleted. It should delete the record and end; it may display some debug statements where needed. Also, include an exception handling section to handle errors (such as deleting a record that still has child tables, no records found, etc.).

- Part 4: Write an anonymous PL/SQL block that inserts a new record in the master and detail tables and then deletes those records. It must use the functions and procedures created from parts 1, 2, and 3. It should first display the count from the master and detail tables; then insert a record in the master table; then insert a record in the detail table related to the master record that was just created; display the counts again from the master and detail tables; and finally delete the master and detail records.

Deliverables:
Package your scripts and any other files that are needed to run your scripts (including a readme) in a folder named activity3 (C:\MSCD650\<your_last _name>\activity3).

Turn in a .sql script(s) file, which includes comments, for each part. Your script files should be named in accordance with the tasks they perform (i.e. get_<table>_count.sql, insert_<table>_rec.sql, delete_<table>_rec.sql, test_act3.sql)

Activity #4
Developing and implementing PL/SQL packages

Description:
This assignment utilizes the code written from activity 3. We will transform existing code to packages and also add two update procedures to the existing database application. It has 4 parts.

Requirements:
- Part 1: Create two packages, one for the master table and one for the detail table, containing the count function, the insert procedure, and the delete procedure from activity 3. Rename the functions and procedures appropriately.

- Part 2: Modify the insert procedures so that instead of taking as arguments the necessary fields, they take a Record as an IN argument (if not already done in activity 3).

- Part 3: Create an Update procedure for the master table and one for the detail table, and include these procedures in the appropriate package. The update procedure should pass these arguments: the primary key of the table (used in the WHERE clause) and a record with the appropriate fields modified.

- Part 4: Modify your script from part 4 of activity 3 so that it calls the correct functions/procedures from the appropriate packages in parts 1-3. Add a section to the script that updates a record from either the master table or the detail table.

Deliverables:
Package your scripts and any other files that are needed to run your scripts (including a readme) in a folder named activity4 (C:\MSCD650\<your_last _name>\activity4).

Turn in a .sql script(s) file, which includes comments, for each part. Your script files should be named in accordance with the tasks they perform.

Activity #5
Developing and implementing triggers

Description:
This assignment will create an audit table that will track updates and deletes on a specific table of your choice. This will be done by implementing a DML trigger.

Requirements:

- Part 1: Create an audit table that will hold the following information: primary key of table, type of DML (insert or delete) operation performed on the table, primary key of record updated or deleted, and date of DML transaction (sysdate).
- Part 2: Create a DML trigger for one table that fires when an update or delete occurs. Use the same table that is used in part 1 of activity 4, so we can reuse the package calls to update and delete the records that will cause the trigger to fire. The trigger should insert a record in the audit table (described in part 1).
- Part 3: Create an anonymous block that calls the package created from activity 4 to insert, update, and delete a record. Two records should be inserted into the audit table created in part1 (one for insert and one for delete).

Deliverables:
Package your scripts and any other files that are needed to run your scripts (including a readme) in a folder named activity5 (C:\MSCD650\<your_last _name>\activity5).

Turn in a .sql script(s) file, which includes comments, for each part. Your script files should be named in accordance with the tasks they perform.

END OF PROJECT – ZIP THE DELIVERABLES:

- You should have five folders (activity1, etc.) in the folder of your last name that matches this directory structure:
  C:\MSCD650\<your_last_name>\activity<#>
- You can download winzip if you do not have it. The easiest way to zip the deliverables is to right click on the folder MSCD650 and choose winzip -> Add to Zip File. Rename the file to <your_last_name>_mscd650_courseproject.zip. Also, choose the option 'Save full path info'. Let me know if you have any questions.