

Lab Assignment 1: SQL Commands and Database Sequences

Equipment and Materials

For this lab assignment, you will need:

- A Windows computer with a minimum of 16 GB RAM and 250 GB of free disk space, capable of nested virtualization
- Access to ORACLE SQL*PLUS or Oracle Academy APEX
- The More Movies database

Note: Execute the **Create_MM.sql** script file, located in the *Course Resources* section of Brightspace.

- A CPRG 307 user account (see instructions below) if using the locally installed Oracle Express software.

Instructions

Part A: Complete the Pre-Lab Tasks

1. Attend the lectures related to the lab activities.
2. Complete the out-of-class learning activities, as indicated by your instructor.
3. Download and review the More Movies database.
4. Create a CPRG 307 user account (*only need to do this step if using the locally installed Oracle Express software; Oracle Academy APEX users do not need to perform this step*)
 - a. Execute the **Create_CPRG307_Users.sql** script located in the *Course Resources* section of Brightspace.
 - b. Ensure that you log into the database as the user SYSTEM.

This script creates two new users with specific privileges you will be using in the labs for this course.

Notes:

- You don't need to understand the code in this script.
- If you see errors when executing the script, ask your instructor for guidance.
- If you need to remove the users, execute **Drop_CPRG307_Users.sql** script also located in the *Course Resources* section of Brightspace.
- This step does not need to be presented in your discussion board post.

5. Create the More Movies database tables in your database.
 - a. *If using the Oracle Express software:* Log into the database using SQL*Plus as user **CPRG307** with the password: **password**.

Execute the script **Create_MM.sql** located in the *Course Resources* section of Brightspace.

Notes:
 - This step does not need to be presented in your discussion board post.
6. See Brightspace for the lab due date.

Part B: Complete the Lab Tasks (create a script that includes the solutions)

1. Display the structure of the MM_MEMBER table.
2. Add yourself as a member.
Hint: Only populate the first three columns.
3. Modify your membership by adding a made-up credit card number. **Do not** use your real-life credit card number.
Hint: There is a check constraint on this column.
4. Remove your membership.
5. Save your data changes.

***NOTE:** If using the Oracle APEX software, saving of the data changes is not needed as this software will perform and auto commit after each transaction.*

Oracle APEX is considered stateless, so it does not remember what has happened between transactions; this is why it auto commits to make sure that the data changes are not accidentally lost. The following articles explain this:

<https://connor-mcdonald.com/2022/11/09/from-stateful-to-stateless-pl-sql/>

https://asktom.oracle.com/ords/f?p=100:11:::P11_QUESTION_ID:9545548900346904742

<https://forums.oracle.com/ords/apexds/post/how-the-auto-commit-works-in-apex-5128>

A “transaction” is not always one command. With the auto commit, after each SQL command, a COMMIT is performed, but what if our business logic requires multiple SQL commands to succeed or fail together as one unit? This is where PL/SQL can help in this stateless environment. A PL/SQL program is considered one unit, one transaction, so if we want to be able to explicitly perform transactional control commands, we need to use PL/SQL.

6. Display the title of each movie, the rental ID and the last names of all members who have rented those movies.
 - a. Sort the result set by the rental ID.
 - b. Ensure that no other information appears.
 - c. Use three tables for this query: MM_MEMBER, MM_MOVIE and MM_RENTAL.

Restriction: Solve using JOIN...ON as your join method.

7. Display the title of each movie, the rental ID, and the last names of all members who have rented those movies.
 - a. No other information should appear.
 - b. Use three tables for this query: MM_MEMBER, MM_MOVIE and MM_RENTAL.

Restriction: Solve using the traditional join method, where join is in the WHERE clause.

8. Create a new table called MY_TABLE that is made up of three columns: MY_NUMBER, MY_DATE and MY_STRING, and that have data types: NUMBER, DATE and VARCHAR2(5), respectively.
9. Create a new sequence called **seq_movie_id**. Have the sequence start at 20 and increment by 2.
10. Display the sequence information (at least the last number and increment by) from the data dictionary's **user_sequences** view.
- Note:** Your output should only show this one sequence.
11. Use a query to display the next sequence number on the screen.
12. Change the sequence created in Step 9 to increment by 5 instead of 2.
13. Add your favorite movie to the MM_MOVIE table using the sequence from the previous step for the movie_id.

Notes:

- You can create values for the other columns (all columns must be given a value).
 - MM_MOVIE has a foreign key, which means any value placed in this column must already exist as primary key value in the table being referenced.
 - MM_MOVIE has a check constraint.
14. Create a view named **VW_MOVIE_RENTAL** using the query from either Step 6 or Step 7.
 15. Use a query to display the data accessed by the **VW_MOVIE_RENTAL** view.
 16. Submit your completed script to the corresponding topic area in the Brightspace Discussion board indicated by your instructor by the due date.

Note: Submit your code in the body of the discussion board post rather than as an attachment.

Part C: Complete the Post-Lab Tasks

1. Compare your posted solution to the solution posted by your instructor.
2. Talk with your instructor if you are unsure why there are differences between the solutions.