Lab: SQL

Lab Overview

This Lab consists of a series of queries against one of MySQL's sample databases which contains data from a movie rental company. You will create and perform these queries using MySQL Workbench, using a copy of the database on your local machine.

While performing this lab, incorporate the principles and practices that you've learned from the SQL lesson in this course.

Submission Instructions

Create a new feature branch in your TEKcamp repository. Commit and push frequently. When your work is done and ready for review, please merge the feature branch into the develop branch, and push.

Following instructions for CodeGrade submission:

1. Click on the "Launch exercise" button on the exercise page at TEKsystems Academy.
2. In the new CodeGrade window, click on the "Connect Git" button.
3. Select GitHub as your Git host. Connect your TEKcamp repository to the exercise*.

After selecting a repository, this repository will be cloned to CodeGrade as a first submission. After this, you can start to use Git like you usually would. Every time you push to the develop branch, it will automatically result in a new submission in CodeGrade. Pushes made after the submission deadline will not be taken into consideration.

* You will only have to log in and authorize CodeGrade once; after that, it will be available for all your other assignments inside CodeGrade too.

WARNING: Be sure to connect the proper repository to the exercise. You will not be able to connect to another one immediately using the CodeGrade connection. You can undo your Git connection with CodeGrade by revoking access to the External Tool "CodeGrade" in your GitHub account only.

Grading Expectations

A solution that meets all the requirements in this lab constitutes a grade of Proficient (3/5). To earn a higher grade, your solution must also apply the code quality concepts you've learned up to this point. Additional functionality that does not improve the quality of the code and only serves to make your assignment look more impressive will not help to raise your grade.

Plagiarism Warning

This assignment is a demonstration of your understanding of the topics covered to-date in the boot camp. You may need to reference tutorials or Q&A sites such as StackOverflow to complete parts of the assignment. It is acceptable to use tutorials/Q&As to learn how to add specific functionality to your

project, but you cannot simply copy a tutorial repository, follow a single tutorial from start to finish, or copy the code from a Q&A site to develop your application. See guidelines below for using tutorials:

- Do not fork a tutorial repository. This would be considered plagiarism and result in an automatic grade of zero and potential removal from the boot camp.
- Do not simply follow the tutorial from start to finish.
- Do not copy code from a Q&A site and then change it for your application (a few lines of code are permissible)
- You must be able to explain how all functionality included in your project works.
- List ALL tutorials or aids used, either in your README file, and/or as code comments. Include:
    - The name of the tutorial,
    - Which section(s) you used, and
    - The functionality you learned from the tutorial.
- Failure to identify tutorials/Q&A responses will result in significant reduction of your grade.
- If you work with other boot camp members – or instructional team members – to find solutions, that is acceptable, but must also be clearly identified

Submission Instructions

For this Lab, create a new feature branch in your git repository and add this Lab document with answers to the questions below, as well as any specified SQL scripts from the exercises.

Estimated Duration

6-8 Hours

Resources

1. Sakila sample database reference: https://dev.mysql.com/doc/sakila/en/sakila-structure.html

Project: SQL

Project Overview

This will be the project for all exercises in this Lab

Project Setup Instructions

Download and install the Sakila sample database and use it for all exercise questions below. Create a new folder in your student repo named sql_exercises.

NOTES:

If an exercise specifies certain columns, only return those columns; otherwise, return the columns that you think are relevant.

For exercises that ask you to use a personal data point for comparison (like your name) but don't return any data, use a different value that does return records. E.g. if your initials are XQ but no records are returned for these values, then select some other initials that do return records (you should include both queries in your response with a disclaimer that the first set of initials are confirmed as returning zero records).

Exercise 1, Day 1

Files Location

Files for this exercise should be placed in a new subfolder of sql_exercises named num_stars.

Instructions: Write a SQL query to perform the following actions and include both the .sql file with the query and a screen shot of the Action Output in this exercise's subfolder:

- Switch to the sakila schema
- List the tables in your schema
- In your schema, create a table named "rating", with fields for
    - an INT primary key
    - number of stars (must be unique across all records in the table)
    - description of what the number of stars means (ex: 0 stars = "worst movie ever")

  (if the table already exists, your query should replace it)
- Populate your table with appropriate data
- List the data in your table

Exercise 2, Day 1

Files Location

Files for this exercise should be placed in a new subfolder of sql_exercises named categories.

Instructions: Write a SQL query to perform the following actions and include both the .sql file with the query and a screen shot of the Result Grid in this exercise's subfolder:

- Retrieve a list of film category names and their database IDs

Exercise 3, Day 2

Files Location

Files for this exercise should be placed in a new subfolder of sql_exercises named employees.

Instructions: Write a SQL query to perform the following actions and include both the .sql file with the query and a screen shot of the Result Grid in this exercise's subfolder:

- Retrieve a list of the first 20 employee names (results should have the format "lastname, firstname")

Exercise 4, Day 2

Files Location

Files for this exercise should be placed in a new subfolder of sql_exercises named film_counts.

Instructions: Write a SQL query to perform the following actions and include both the .sql file with the query and a screen shot of the Result Grid in this exercise's subfolder:

- Retrieve a list of the number of copies of each film that are in each store's inventory

Exercise 5, Day 3

Files Location

Files for this exercise should be placed in a new subfolder of sql_exercises named customer_search.

Instructions: Write a SQL query to perform the following actions and include both the .sql file with the query and a screen shot of the Result Grid in this exercise's subfolder:

- Retrieve a list of all customers whose first and last names contain one of your initials

Exercise 6, Day 3

Files Location

Files for this exercise should be placed in a new subfolder of sql_exercises named movie_search.

Instructions: Write a SQL query to perform the following actions and include both the .sql file with the query and a screen shot of the Result Grid in this exercise's subfolder:

- Retrieve a list of all the films that have the same 2-3 letters (of your choice, and they don't need to be together) in the title, and that are over 2 hours long

Exercise 7, Day 3

Files Location

Files for this exercise should be placed in a new subfolder of sql_exercises named early_rentals.

Instructions: Write a SQL query to perform the following actions and include both the .sql file with the query and a screen shot of the Result Grid in this exercise's subfolder:

- Retrieve a list of the most recent 10 rentals that occurred before 10AM
    - Include all columns from rental, as well as the date (not including the time) that the rental occurred

Exercise 8, Day 4

Files Location

Files for this exercise should be placed in a new subfolder of sql_exercises named customer_ratings.

Instructions: Write a SQL query to perform the following actions and include the .sql file with the query, a screen shot of the Action Output, and a screen shot of the Result Grid in this exercise's subfolder:

- Add an additional table to your schema, to store records of customer ratings for movies
    - Foreign key constraints/ cardinality should be set appropriately
- Create (manually or randomly) 20-30 records in the customer ratings table
- List each movie, including the highest, lowest, and average score they received, as well as the number of reviews

Exercise 9, Day 4

Files Location

Files for this exercise should be placed in a new subfolder of sql_exercises named staff_cities.

Instructions: Write a SQL query to perform the following actions and include both the .sql file with the query and a screen shot of the Result Grid in this exercise's subfolder:

- Retrieve a list of all staff that have their home store in a certain city (a city of your choice)

Exercise 10, Day 4

Files Location

Files for this exercise should be placed in a new subfolder of sql_exercises named multi_rental.

Instructions: Write a SQL query to perform the following actions and include both the .sql file with the query and a screen shot of the Result Grid in this exercise's subfolder:

- Retrieve a list of customers that rented more than one movie on the same day, and list the customer, rental date, number of rentals, and total amount paid for that day's rentals

Exercise 11, Day 4

Files Location

Files for this exercise should be placed in a new subfolder of sql_exercises named customer_favorites.

Instructions: Write a SQL query to perform the following actions and include both the .sql file with the query and a screen shot of the Result Grid in this exercise's subfolder:

- For those customers with one of your initials in their name, list their most-often-rented movie category, from their past 10 rentals