



☆ 0 stars  429 forks  0 watching  1 Branch  0 Tags  Activity  Custom properties

 Public repository



debironhack adding "bonus" for some queries

d55ef65 · last year 



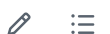
README.md

adding "bonus" for some queries

last year



 README



LAB | SQL Basic Queries

▼ Learning Goals

This lab allows you to practice and apply the concepts and techniques taught in class.

Upon completion of this lab, you will be able to:

- Use SQL queries to extract insights from the data and answer research questions or problem statements, using techniques such as selecting columns with **SELECT** clause, filtering with **WHERE** clause, sorting data with **ORDER BY** clause, limiting results with **LIMIT**, using **DISTINCT** to retrieve unique values, and counting records with **COUNT**.

▼ Prerequisites

Before this starting this lab, you should have learnt about:

- **SELECT**, **FROM**, **ORDER BY**, **LIMIT**, **WHERE** clauses.

- DISTINCT keyword to return only unique values, AS keyword for using aliases, COUNT function for counting records.

Introduction

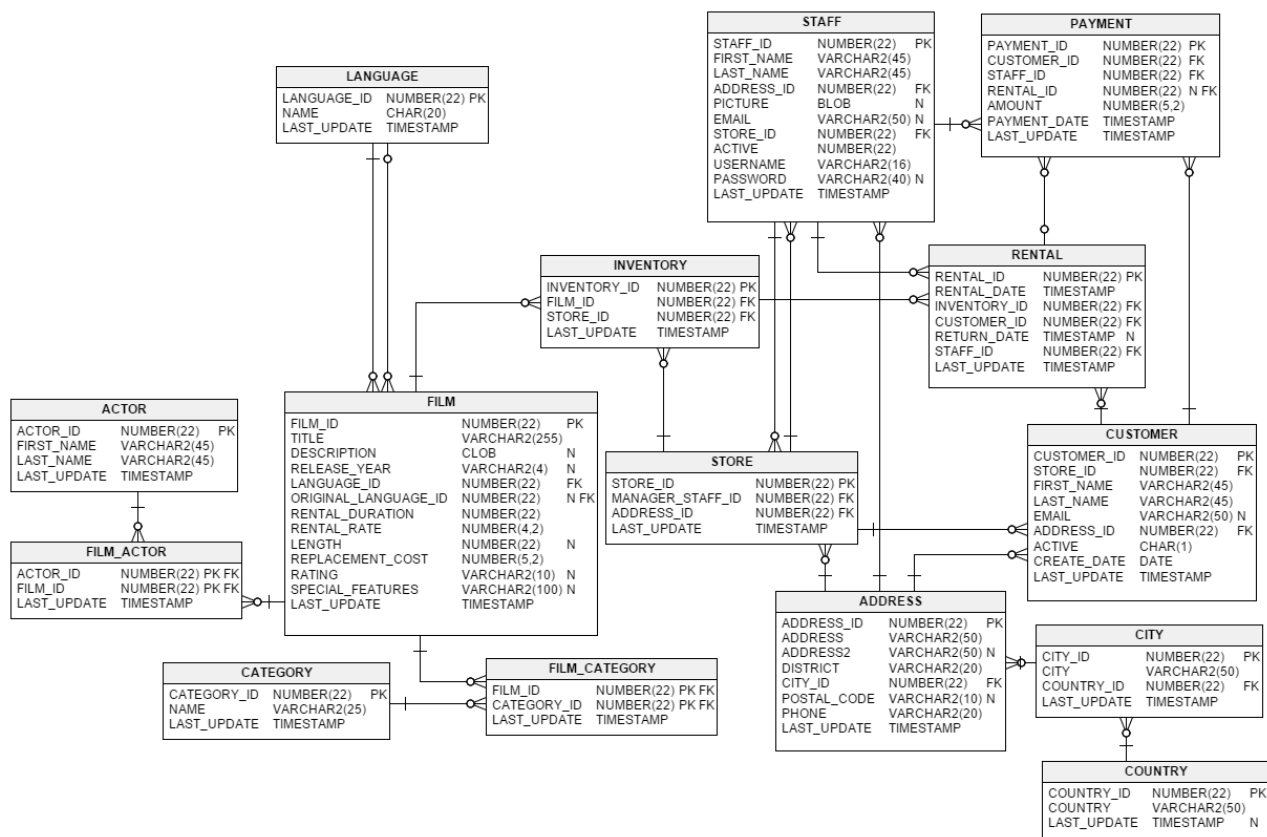
Welcome to the SQL Basic Queries lab!

In this lab, you will practice how to use SQL queries to extract insights from the [Sakila](#) database which contains information about movie rentals.

In this lab, you'll practice the basics of SQL, including how to select data from tables, filter results using the WHERE clause, aggregate data with functions like COUNT, and sort results using the ORDER BY clause.

Throughout the lab, you will work with two SQL query files: `sakila-schema.sql`, which creates the database schema, and `sakila-data.sql`, which inserts the data into the database. You can download the necessary files locally by following the steps listed in [Sakila sample database - installation](#).

You can also refer to the Entity Relationship Diagram (ERD) of the database to guide your analysis:



Challenge

Use the sakila database to do the following tasks:

1. Display all available tables in the Sakila database.
2. Retrieve all the data from the tables `actor`, `film` and `customer`.

3. Retrieve the following columns from their respective tables:

- 3.1 Titles of all films from the `film` table
- 3.2 List of languages used in films, with the column aliased as `language` from the `language` table
- 3.3 List of first names of all employees from the `staff` table

4. Retrieve unique release years.

5. Counting records for database insights:

- 5.1 Determine the number of stores that the company has.
- 5.2 Determine the number of employees that the company has.
- 5.3 Determine how many films are available for rent and how many have been rented.
- 5.4 Determine the number of distinct last names of the actors in the database.

6. Retrieve the 10 longest films.

7. Use filtering techniques in order to:

- 7.1 Retrieve all actors with the first name "SCARLETT".

BONUS:

- 7.2 Retrieve all movies that have ARMAGEDDON in their title and have a duration longer than 100 minutes.
 - Hint: use `LIKE` operator. [More information here.](#)
- 7.3 Determine the number of films that include Behind the Scenes content

Requirements

- Fork this repo
- Clone it to your machine

Getting Started

Complete the challenges in this readme in a `.sql` file.

Submission

- Upon completion, run the following commands:

```
git add .
git commit -m "Solved lab"
git push origin master
```



- Paste the link of your lab in Student Portal.