data-bootcamp-v4 /
lab-sql-aggregation-and-transformation

Code Issues Pull requests 177 Actions Projects Security Insights

2 stars 418 forks 0 watching 1 Branch 0 Tags Activity Custom properties

Public repository

1 Branch 0 Tags Go to file t Go to file Add file Code

debironhack Update README.md 1b9fd1c · last year

README.md	Update README.md	last year

Lab | SQL Data Aggregation and Transformation

README

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 built-in functions such as COUNT, MAX, MIN, AVG to aggregate and summarize data, and use GROUP BY to group data by specific columns. Use the HAVING clause to filter data based on aggregate functions.
- Use SQL to clean, transform, and prepare data for analysis by handling duplicates, null values, renaming columns, and converting data types. Use functions like ROUND, DATE_DIFF, CONCAT, and SUBSTRING to manipulate data and generate insights.
- Use conditional expressions for creating new columns.

▾ Prerequisites

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

- SELECT, FROM, ORDER BY, LIMIT, WHERE, GROUP BY, and HAVING clauses.
- DISTINCT keyword to return only unique values, AS keyword for using aliases.
- Built-in SQL functions such as COUNT, MAX, MIN, AVG, ROUND, DATEDIFF, or DATE_FORMAT.
- CASE statement for conditional logic.

Introduction

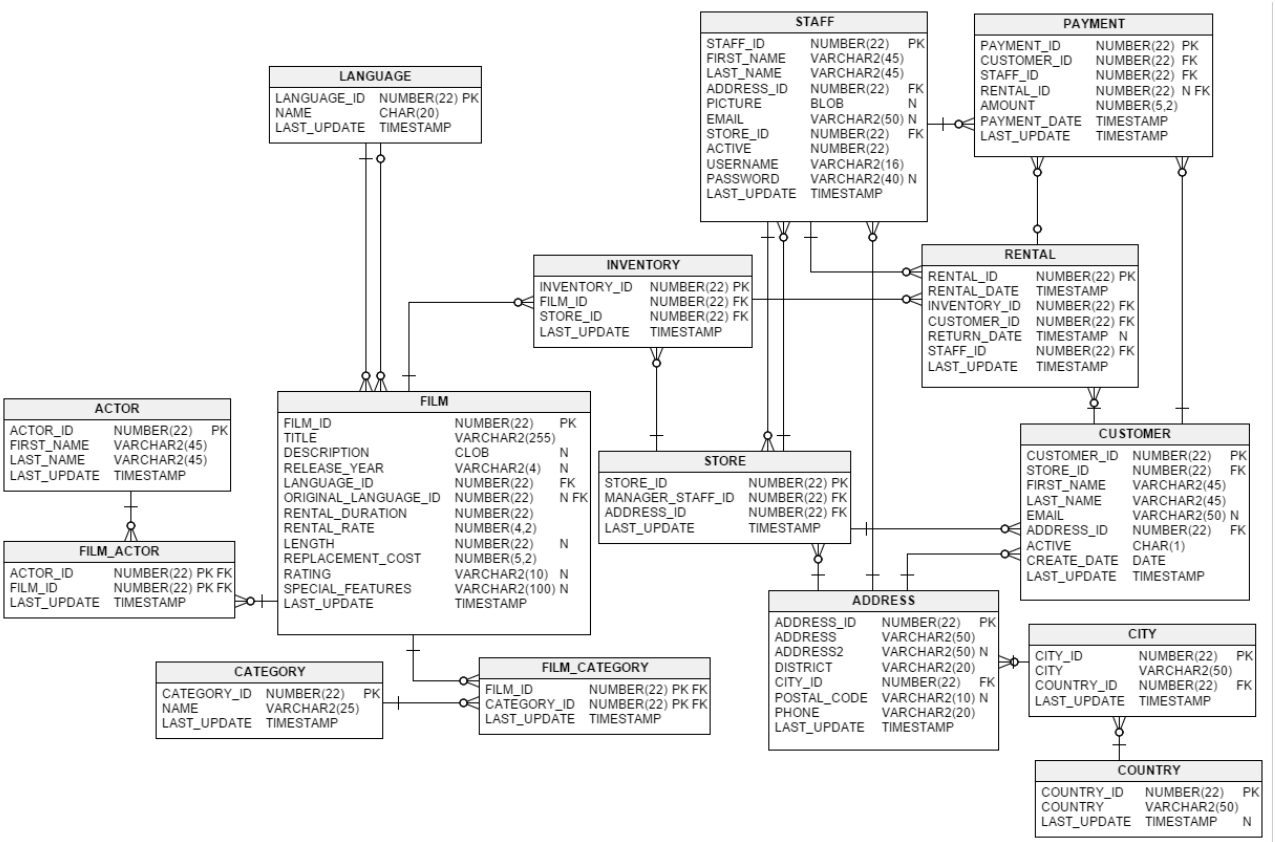
Welcome to the SQL Data Aggregation and Transformation lab!

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

You will build on your SQL skills by practicing how to use the `GROUP BY` and `HAVING` clauses to group data and filter results based on aggregate values. You will also practice how to handle null values, rename columns, and use built-in functions like `MAX`, `MIN`, `ROUND`, `DATE_DIFF`, `CONCAT`, and `SUBSTRING` to manipulate and transform data for generating insights.

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:



Imagine you work at a movie rental company as an analyst. By using SQL in the challenges below, you are required to gain insights into different elements of its business operations.

Challenge 1

1. You need to use SQL built-in functions to gain insights relating to the duration of movies:
    - 1.1 Determine the **shortest and longest movie durations** and name the values as `max_duration` and `min_duration`.
    - 1.2. Express the **average movie duration in hours and minutes**. Don't use decimals.
        - *Hint: Look for floor and round functions.*
2. You need to gain insights related to rental dates:
    - 2.1 Calculate the **number of days that the company has been operating**.
        - *Hint: To do this, use the `rental` table, and the `DATEDIFF()` function to subtract the earliest date in the `rental_date` column from the latest date.*
    - 2.2 Retrieve rental information and add two additional columns to show the **month and weekday of the rental**. Return 20 rows of results.
    - 2.3 *Bonus: Retrieve rental information and add an additional column called `DAY_TYPE` with values **'weekend' or 'workday'**, depending on the day of the week.*
        - *Hint: use a conditional expression.*
3. You need to ensure that customers can easily access information about the movie collection. To achieve this, retrieve the **film titles and their rental duration**. If any rental duration value is **NULL, replace** it with the string **'Not Available'**. Sort the results of the film title in ascending order.
    - *Please note that even if there are currently no null values in the rental duration column, the query should still be written to handle such cases in the future.*
    - *Hint: Look for the `IFNULL()` function.*
4. *Bonus: The marketing team for the movie rental company now needs to create a personalized email campaign for customers. To achieve this, you need to retrieve the **concatenated first and last names of customers**, along with the **first 3 characters of their email** address, so that you can address them by their first name and use their email address to send personalized recommendations. The results should be ordered by last name in ascending order to make it easier to use the data.*

Challenge 2

1. Next, you need to analyze the films in the collection to gain some more insights. Using the `film` table, determine:
    - 1.1 The **total number of films** that have been released.
    - 1.2 The **number of films for each rating**.
    - 1.3 The **number of films for each rating, sorting** the results in descending order of the number of films. This will help you to better understand the popularity of different film ratings and adjust purchasing decisions accordingly.
2. Using the `film` table, determine:
    - 2.1 The **mean film duration for each rating**, and sort the results in descending order of the mean duration. Round off the average lengths to two decimal places. This will help identify popular movie lengths for each category.
    - 2.2 Identify **which ratings have a mean duration of over two hours** in order to help select films for customers who prefer longer movies.
3. *Bonus: determine which last names are not repeated in the table `actor`.*