learn-co-curriculum / dsc-sql-interview-questions-lab Public

View license

8 stars 179 forks

Star Watch

Code Issues Pull requests 1 Actions Projects Security Insights

solution

This branch is 8 commits ahead, 7 commits behind master. Contribute

mas16 update learning objectives … on Oct 28, 2019 9

View code

README.md

SQL Interview Questions - Lab

Introduction

In this lab, we'll test our SQL skills against some real-world interview questions from major companies!

Objectives

You will be able to:

- Write SQL queries to filter and order results
- Decide and perform whichever type of join is best for retrieving desired data
- Write subqueries to decompose complex queries

Getting Started

In this lab, we'll see four different interview questions that test your SQL knowledge. We didn't write these questions -- instead, we found them out in the real-world. These are questions that have been used in the past by major technology companies such as Facebook, Amazon, and Twitter. Our goal here isn't to memorize the questions or anything like that -- after all, it's extremely unlikely that these questions are still in use, now that they've become publicly available on the interwebs. Instead, our goal is to treat these questions as if they are the real thing, and give us some insight into the types of questions we'll need to be able to answer in order pass an interview involving SQL.

If these questions seem hard to you, don't sweat it, they're supposed to be tough! These are meant to help you identify any areas of knowledge where you still need to grow! Use these questions as a way to see where your SQL knowledge is strong, and where it's a bit weak. Then, go study and **practice** in the areas where you still need work!

A Note on Answering These Questions

Since these are interview questions, they'll almost always be posed as hypotheticals. This means that you won't have a real database to work with and test your code on. This also means that there are multiple different solutions to any given problem listed here. Be sure to doublecheck the code you write for bugs and errors. It's much harder to write bug-free code when you aren't able to test it against a database!

If these questions seem hard, that's normal. These are real questions that have been reported to online forums from job seekers at major companies. Obviously, it's unlikely that they're still in use at these companies, but they still represent a great way for us to test our skills against the kinds of questions we can expect to be asked in an interview!

Question 1

From Facebook:

Assume we have a table of employee information, which includes salary information. Write a query to find the names and salaries of the top 5 highest paid employees, in descending order.

```
"""
SELECT name, salary from Employees
ORDER BY salary DESC
LIMIT 5;
"""
```

Question 2

From Amazon:

Assume we have two SQL tables: `authors` and `books` . The authors table has a few million rows, and looks like this:

author_name	book_name
author_1	book_1
author_1	book_2
author_2	book_3
author_2	book_4
author_2	book_5
author_3	book_6

The books dataset also has a few million rows, and looks like this:

book_name	copies_sold
book_1	10000
book_2	2575
book_3	60000
book_4	98000
book_5	5250
book_6	19775

Write an SQL query that shows the top 3 authors who sold the most total books.

```
"""
SELECT a.author_name, SUM(b.copies_sold) as total_sold from Authors a
JOIN Books b ON a.book_name = b.book_name
GROUP BY a.author_name
ORDER BY total_sold DESC
LIMIT 3;
"""
```

Question 3

From Amazon:

Assume you have two tables, `customers` and `orders`. Write a SQL query to select all customers who purchased at least 2 items on two separate days.

```
"""
SELECT c.name, COUNT(DISTINCT o.OrderDate) as NumOrderDates FROM (SELECT c.name, o.c
    JOIN Orders o ON c.orderNumber = o.OrderNumber
    WHERE o.quantity > 1)
WHERE NumOrderDates > 1
"""
```

Question 4

From Twitter:

A company uses 2 data tables, `Employee` and `Department`, to store data about its employees and departments.

Table Name: Employee
Attributes:
ID Integer,
NAME String,
SALARY Integer,
DEPT_ID Integer

Table Name: Department
Attributes:
DEPT_ID Integer,
NAME String,
LOCATION String

Write a query to print the respective Department Name and number of employees for all departments in the Department table (even unstaffed ones).

Sort your result in descending order of employees per department; if two or more departments have the same number of employees, then sort those departments alphabetically by Department Name.

```
"""
SELECT d.name, COUNT(e.ID) as EmployeeCount
FROM Department d
LEFT JOIN Employee e on d.dept_id, = e.dept_id
GROUP BY d.dept_id, d.name
ORDER BY EmployeeCount DESC, d.name
"""
```

Summary

In this lab, we tested our knowledge of SQL queries against some real-world interview questions!

Releases

No releases published

Packages

No packages published

Contributors 5

Languages

● **Jupyter Notebook** 100.0%