# Siddhesh Singh

# 23070521150

# SQL JOIN Practice Questions

Assume you have the following two tables:

**Employees**

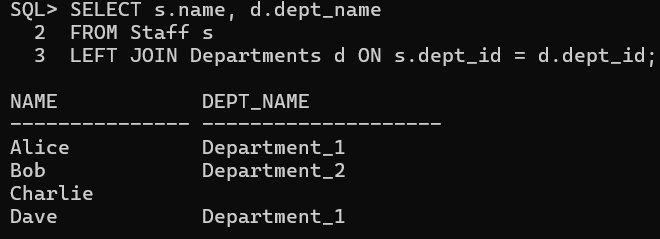
|  |  |  |  |
| --- | --- | --- | --- |
| **emp\_id** | **name** | **department\_id** | **salary** |
| 1 | Alice | 101 | 5000 |
| 2 | Bob | 102 | 6000 |
| 3 | Charlie | NULL | 5500 |
| 4 | Dave | 101 | 7000 |

**Departments**

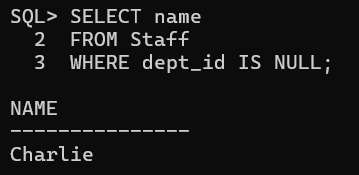
|  |  |
| --- | --- |
| **department\_id** | **department\_name** |
| 101 | HR |
| 102 | IT |
| 103 | Marketing |

**Basic JOINs**

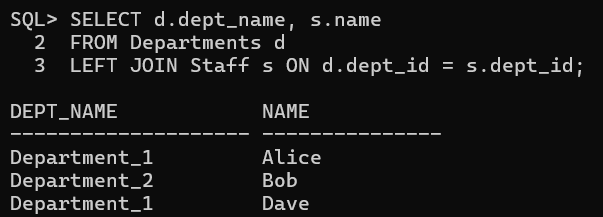
1. Write a query to get all employees along with their department names.



1. List the names of employees who **do not belong** to any department.

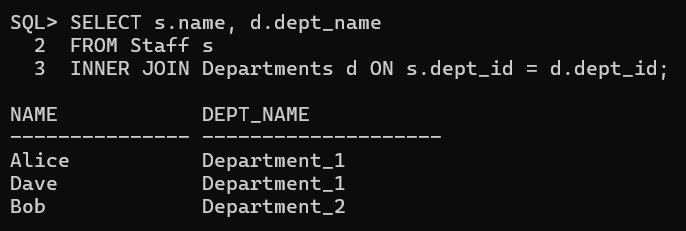


1. List all departments and the names of employees in them (if any).

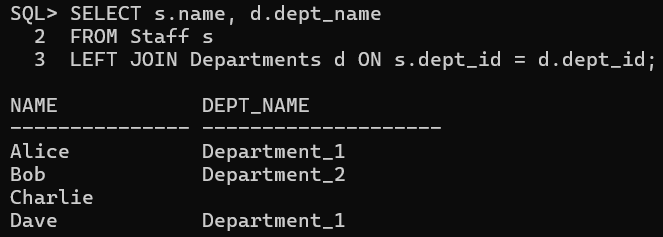


**INNER, LEFT, RIGHT, FULL JOINs**

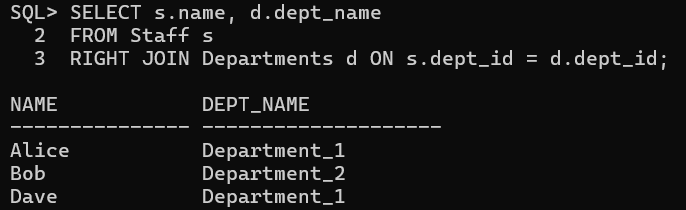
1. Use an **INNER JOIN** to find employees who are assigned to a department.



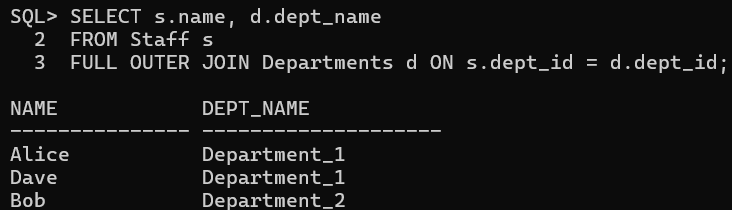
1. Use a **LEFT JOIN** to list all employees and their department names (even if department is NULL).



1. Use a **RIGHT JOIN** to list all departments and employees (if any).

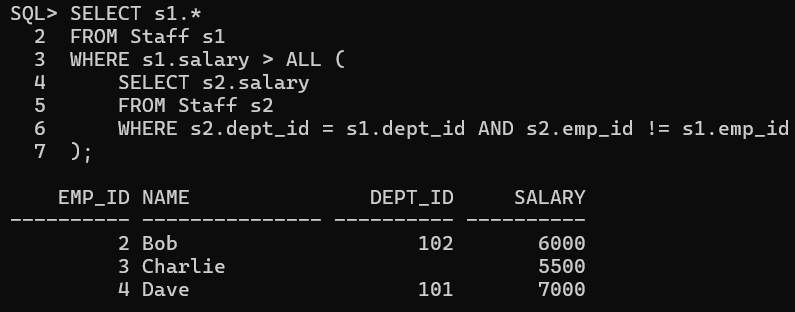


1. Use a **FULL OUTER JOIN** to show all employees and all departments, matching where possible.

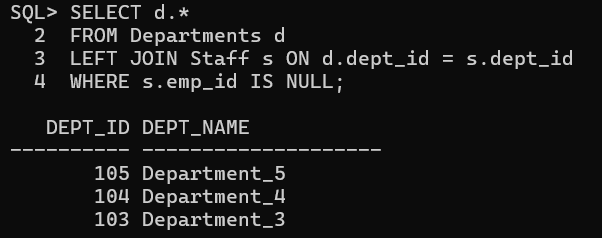


**JOIN with Conditions**

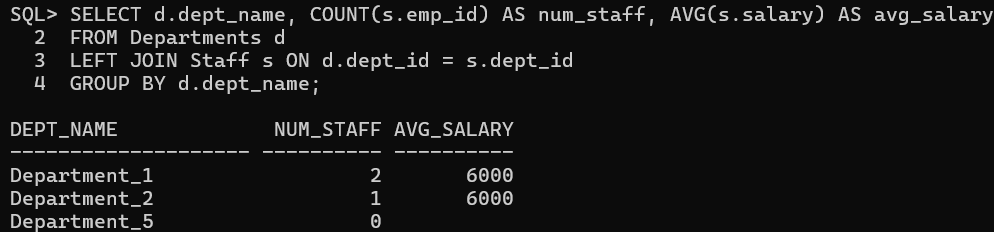
1. Write a query to list employees who earn more than any other employee in their department.



1. List all departments where **no employees** are assigned.



1. For each department, show the number of employees and the average salary.



# SQL Trigger Practice Questions

Assume you have a table:

**Orders**

|  |  |  |  |
| --- | --- | --- | --- |
| **order\_id** | **customer\_name** | **amount** | **status** |
| 1 | John Doe | 200 | pending |
| 2 | Alice Smith | 350 | shipped |

And a log table:

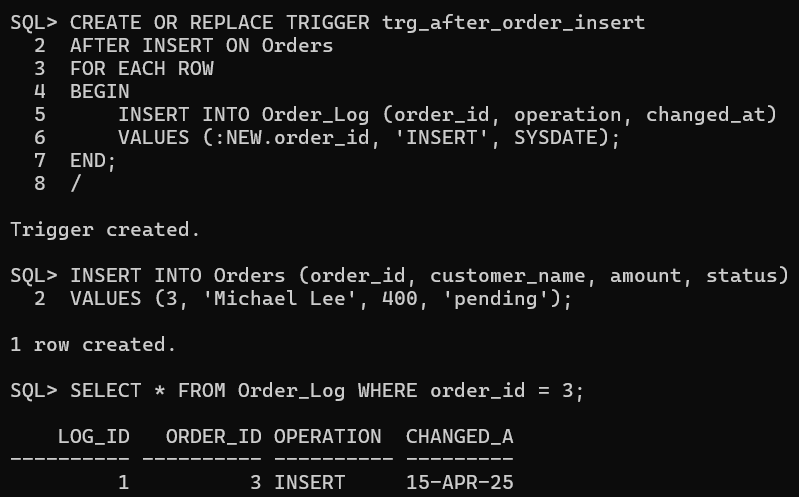
**Order\_Log**

|  |  |  |  |
| --- | --- | --- | --- |
| **log\_id** | **order\_id** | **operation** | **changed\_at** |
|  |  |  |  |

**Trigger Practice**

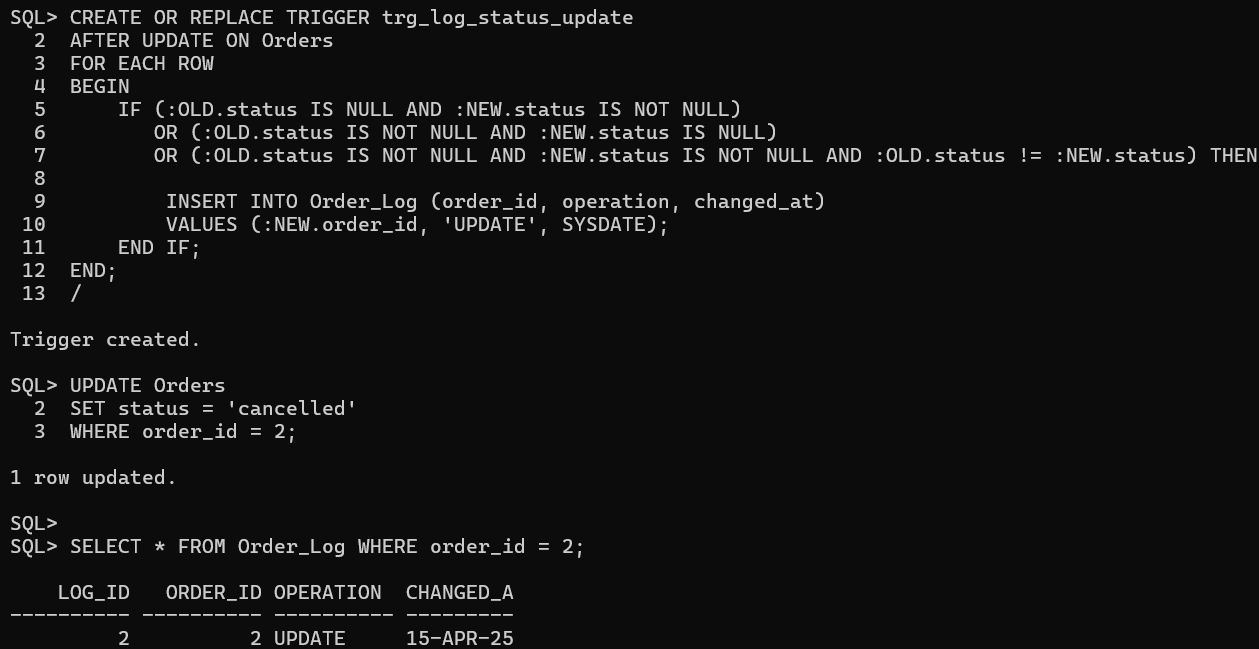
1. Create a trigger that inserts a row into Order\_Log **after an order is inserted** into

Orders.

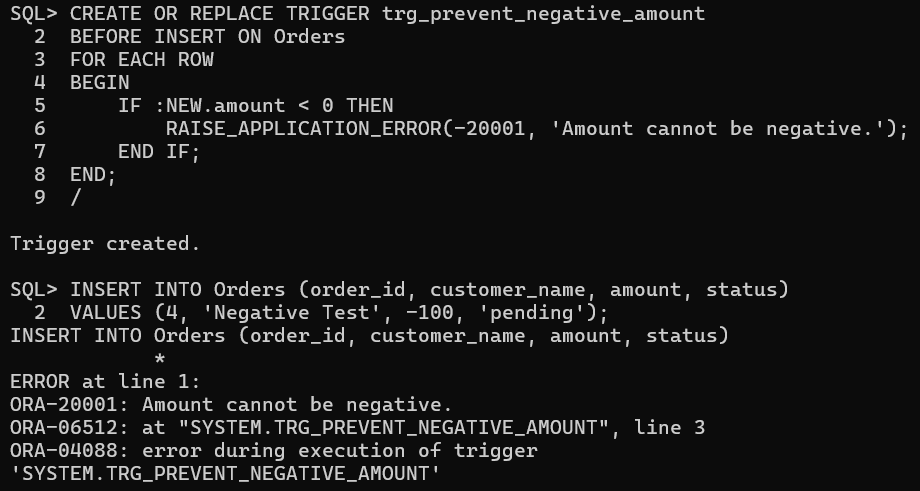


1. Create a trigger that logs an update to the Orders table whenever the status

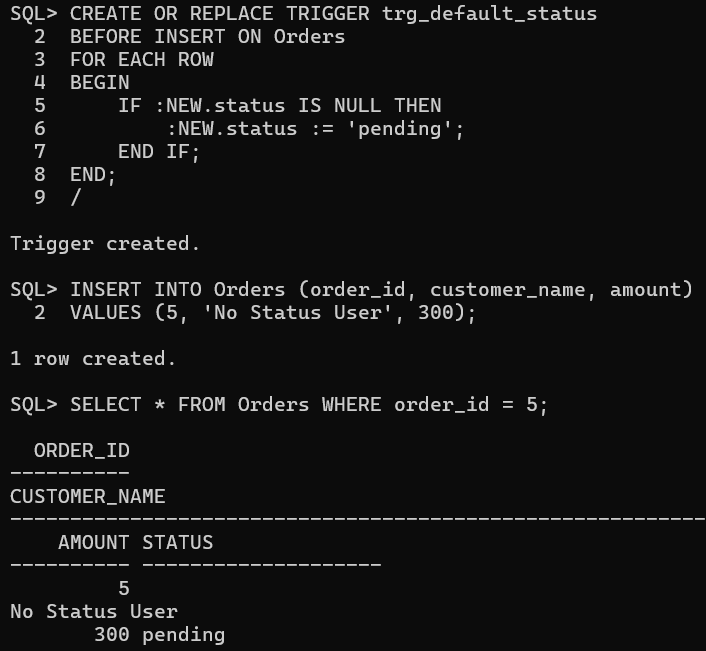
changes.



1. Write a trigger that prevents inserting an order with a negative amount.



1. Write a trigger that automatically sets the status to "pending" if no status is provided during insert.



1. Create a trigger that updates another table Customer\_Balance whenever an order is placed.

