Scenario-Based SQL Interview Questions

1. Write a Query to Find Duplicate Rows in a Table

Answer:

To find duplicate rows in a table, group by the columns that define a duplicate and use the `HAVING` clause to filter groups with more than one occurrence.

Example:

Suppose you have a table called `employees` with columns `first_name`, `last_name`, and `email`. To find duplicates based on `first_name` and `last_name`:

```
SELECT
    first_name,
    last_name,
    COUNT(*) AS duplicate_count
FROM
    employees
GROUP BY
    first_name,
    last_name
HAVING
    COUNT(*) > 1;
```

Explanation:

- **GROUP BY** groups rows with the same `first_name` and `last_name`.
- **COUNT(*)** counts the number of occurrences for each group.
- **HAVING COUNT(*) > 1** filters only those groups that have duplicates.

> **Tip:** Adjust the columns in the `GROUP BY` clause to match the definition of a duplicate in your specific table.

2. Explain the Difference Between INNER JOIN and OUTER JOIN with Examples

Answer:

INNER JOIN returns only the rows that have matching values in both tables.
OUTER JOIN returns all rows from one or both tables, filling in NULLs where there is no match.

Example:

Suppose you have two tables: `employees` and `departments`.

- `employees(employee_id, name, department_id)`
- `departments(department_id, department_name)`

INNER JOIN Example: Returns only employees who belong to a department.

```
SELECT
    e.name,
    d.department_name
FROM
    employees e
INNER JOIN
    departments d ON e.department_id = d.department_id;
```

OUTER JOIN Example (LEFT OUTER JOIN): Returns all employees, including those who do not belong to any department.

```
SELECT
    e.name,
    d.department_name
FROM
    employees e
LEFT OUTER JOIN
    departments d ON e.department_id = d.department_id;
```

Explanation:

- **INNER JOIN** includes only rows with matching `department_id` in both tables.
- **LEFT OUTER JOIN** includes all rows from `employees`, and fills `department_name` with NULL if there is no matching department.
- You can also use **RIGHT OUTER JOIN** or **FULL OUTER JOIN** to include all rows from the right table or both tables, respectively.

> **Tip:** Use `INNER JOIN` when you need only matching records, and `OUTER JOIN` when you want to include unmatched rows as well.

---

3. Write a Query to Fetch the Second-Highest Salary from an Employee Table

Answer:

To get the second-highest salary, you can use the `ORDER BY` and `LIMIT` clauses, or use a subquery to exclude the highest salary.

Example:

Suppose you have a table called `employees` with a column `salary`.

Using LIMIT/OFFSET (works in MySQL, PostgreSQL):

```
SELECT
    DISTINCT salary
FROM
    employees
ORDER BY
    salary DESC
LIMIT 1 OFFSET 1;
```

Using Subquery (works in most SQL dialects):

```
SELECT
    MAX(salary) AS second_highest_salary
FROM
    employees
WHERE
    salary &lt; (SELECT MAX(salary) FROM employees);
```

Explanation:

- The first query orders salaries in descending order, skips the highest, and fetches the next one.
- The second query finds the maximum salary that is less than the overall maximum, effectively giving the second-highest salary.
- **DISTINCT** ensures duplicate salaries are not counted multiple times.

> **Tip:** If there are multiple employees with the same second-highest salary, both queries will return that value. Adjust the query if you need all employees with the second-highest salary.
