📘 SQL Interview Practice Notes for Students
==========

These are the top 9 SQL query-based interview questions often asked in interviews.

Each question includes explanation, sample query, and key points.

---

1. How to find the third highest salary from the Employee table?

✅ Query (MySQL / PostgreSQL):

```
SELECT DISTINCT Salary
FROM Employee
WHERE Salary IS NOT NULL
ORDER BY Salary DESC
LIMIT 1 OFFSET 2;
```

✅ Query (Oracle/SQL Server (MSSQL)):

```
SELECT DISTINCT Salary
FROM Employee
WHERE Salary IS NOT NULL
ORDER BY Salary DESC
OFFSET 2 ROWS FETCH NEXT 1 ROW ONLY;
```

💡 Explanation:

- 'ORDER BY Salary DESC' sorts salaries from highest to lowest.

- 'OFFSET 2' skips first two rows, giving the 3rd highest.

- 'IS NOT NULL' ensures null salaries don’t interfere.

🧠 Common mistake:

- Forgetting to handle NULLs may shift results in DISTINCT queries.

---

2. Write a query to find the Nth highest salary without using TOP or LIMIT.

✅ Query:

```
SELECT DISTINCT Salary
FROM Employee e1
WHERE N - 1 = (
  SELECT COUNT(DISTINCT Salary)
  FROM Employee e2
  WHERE e2.Salary > e1.Salary
);
```

💡 Explanation:

- Replaces LIMIT/TOP with a COUNT-based logic.

- Works on any database.

🧠 Common mistake:

Common Mistake	Why it's Wrong
Forgetting DISTINCT	Counts duplicates, returns wrong rank
Using >= instead of >	Off-by-one logic error
Writing (N) instead of (N-1)	Counts too many

3.Find duplicate employee names in the Employee table.

✅ Query:

```
SELECT Name, COUNT(*)
FROM Employee
GROUP BY Name
HAVING COUNT(*) > 1;
```

💡 Explanation:

- GROUP BY aggregates rows with same Name.

- COUNT(*) counts how many times each appears.

- HAVING COUNT(*) > 1 filters groups that appear more than once.

🧠 Common mistake:

Common Mistake	Why it's Wrong
Using `WHERE COUNT(*) > 1`	You can't use aggregate functions in WHERE
Forgetting `GROUP BY` columns	Causes syntax error
Ignoring NULLs	NULLs can form duplicate groups too

4.Write a query to separate even and odd-numbered records in a table.

✅ Query (MySQL / Oracle):

-- Even-numbered

```
SELECT * FROM Employee WHERE MOD(Id, 2) = 0;
```

-- Odd-numbered

```
SELECT * FROM Employee WHERE MOD(Id, 2) = 1;
```

✅ Query (MySQL /SQL Server(MSSQL)):

-- Even-numbered

```
SELECT * FROM Employee WHERE Id % 2 = 0;
```

-- Odd-numbered

```
SELECT * FROM Employee WHERE Id % 2 = 1;
```

💡 Explanation:

- MOD(Id, 2) returns remainder when dividing by 2.

- Even = 0, Odd = 1.

---

5. How do you fetch the first and last records from the Employee table?

✅ Query (MySQL) :

-- First record

```
SELECT * FROM Employee ORDER BY Id ASC LIMIT 1;
```

-- Last record

```
SELECT * FROM Employee ORDER BY Id DESC LIMIT 1;
```

✅ Query (SQL Server(MSSQL)) :

-- First record

```
SELECT TOP 1 * FROM Employee ORDER BY Id ASC;
```

-- Last record

```
SELECT TOP 1 * FROM Employee ORDER BY Id DESC;
```

✅ Query (Oracle) :

-- First record

```
SELECT * FROM Employee ORDER BY Id ASC FETCH FIRST 1 ROW ONLY;
```

-- Last record

```
SELECT * FROM Employee ORDER BY Id DESC FETCH FIRST 1 ROW ONLY;
```

🧠 Common mistake:

Common Mistake	Why it's Wrong
Not using `ORDER BY`	SQL tables have no default order — "first" and "last" are undefined
Using `TOP` / `LIMIT` without `ORDER BY`	Returns random row
Assuming `ID` always reflects insert order	Not true in all systems unless it's auto-increment

---

6. How to copy all rows from one table to another using SQL?

Create + Copy (Create Table from Existing Data)

✅ Query (MySQL/Oracle):

```
CREATE TABLE EmployeeNew AS
SELECT * FROM Employee;
```

✅ Query (SQL Server(MSSQL)):

```
SELECT * INTO EmployeeNew FROM Employee;
```

Copy All Rows (Existing Target Table)

✅ Query (MySQL/Oracle/SQL Server(MSSQL)):

```
INSERT INTO EmployeeNew
SELECT * FROM Employee;
```

Copy All Rows (When Table Structures Differ)

✅ Query (MySQL/Oracle/SQL Server(MSSQL)):

```
INSERT INTO EmployeeNew (Id, name, salary)
SELECT Id, name, salary
FROM Employee;
```

💡 Explanation:

- Copies all data between tables with identical structure.

- Use column lists if structures differ.

---

7. How do you retrieve all employees working in the same department?

✅ Query (Using GROUP BY):

```
SELECT Name, Department
FROM Employee
WHERE Department IN (
  SELECT Department
  FROM Employee
  GROUP BY Department
  HAVING COUNT(*) > 1
);
```

💡 Explanation:

- Groups employees by department.

- Returns only those departments where more than one employee exists.

---

8. Write a query to fetch the last 3 inserted records from the Employee table.

✅ Query (MySQL/PostgreSQL):

```
SELECT * FROM Employee
ORDER BY Id DESC
LIMIT 3;
```

✅ Query (SQL Server (MSSQL)):

```
SELECT TOP 3 *
FROM Employee
ORDER BY Id DESC;
```

✅ Query (Oracle):

```
SELECT *
FROM Employee
ORDER BY Id DESC
FETCH FIRST 3 ROWS ONLY;
```

💡 Explanation:

- Orders by Id descending and picks top 3.

🧠 Common mistake:

Common Mistake	Why it's Wrong
Not using `ORDER BY`	"Last inserted" has no meaning without ordering
Assuming physical order = insertion order	Not guaranteed unless using auto-increment or timestamp

---

9. Find employees whose names end with the letter 'A' and are exactly 5 characters long.

✅ Query (Pattern-based):

```
SELECT * FROM Employee
WHERE UPPER(Name) LIKE '____A';
```

💡 Explanation:

- Each '_' matches one character → 4 + 'A' = 5 letters.

- UPPER() ensures case-insensitive comparison.

📘 End of Notes

Practice these queries with sample data and understand the logic behind each.