SQL Interview Questions for Experienced Candidates (3+ years)

1. What are the different types of SQL joins (`INNER JOIN`, `LEFT JOIN`, `RIGHT JOIN`, `FULL JOIN`, `CROSS JOIN`, etc.) and when would you use each?

Answer:

SQL joins are used to combine rows from two or more tables based on related columns. The main types are:

- **INNER JOIN:** Returns only rows with matching values in both tables. Use when you need records present in both tables.
- **LEFT JOIN (LEFT OUTER JOIN):** Returns all rows from the left table and matched rows from the right table. Unmatched rows from the right table return NULLs. Use when you want all records from the left table, regardless of matches.
- **RIGHT JOIN (RIGHT OUTER JOIN):** Returns all rows from the right table and matched rows from the left table. Unmatched rows from the left table return NULLs. Use when you want all records from the right table.
- **FULL JOIN (FULL OUTER JOIN):** Returns all rows when there is a match in either table. Unmatched rows from either side return NULLs. Use when you want all records from both tables.
- **CROSS JOIN:** Returns the Cartesian product of both tables (every row of the first table joined with every row of the second). Use rarely, typically for generating combinations.

Example:

INNER JOIN:

```
SELECT a.*, b.*
FROM TableA a
INNER JOIN TableB b ON a.id = b.a_id;
```

LEFT JOIN:

```
SELECT a.*, b.*
FROM TableA a
LEFT JOIN TableB b ON a.id = b.a_id;
```

RIGHT JOIN:

```
SELECT a.*, b.*
FROM TableA a
RIGHT JOIN TableB b ON a.id = b.a_id;
```

FULL OUTER JOIN:

```
SELECT a.*, b.*
FROM TableA a
FULL OUTER JOIN TableB b ON a.id = b.a_id;
```

CROSS JOIN:

```
SELECT a.*, b.*
FROM TableA a
CROSS JOIN TableB b;
```

Explanation:
- Use **INNER JOIN** when you only want rows with matches in both tables.
- Use **LEFT JOIN** to get all rows from the left table, even if there are no matches in the right.
- Use **RIGHT JOIN** to get all rows from the right table, even if there are no matches in the left.
- Use **FULL OUTER JOIN** to get all rows from both tables, with NULLs where there are no matches.
- Use **CROSS JOIN** to get every combination of rows from both tables (rarely used in practice).

2. What is the difference between a `CROSS JOIN` and a `FULL OUTER JOIN`?

Answer:

`CROSS JOIN` and `FULL OUTER JOIN` are both used to combine rows from two tables, but they operate very differently:

Feature	CROSS JOIN	FULL OUTER JOIN
Purpose	Returns the Cartesian product of both tables (all possible combinations of rows).	Returns all rows from both tables, matching rows where possible, and filling with NULLs where there is no match.
Join Condition	No join condition is used.	Join condition is required (typically ON clause).

Feature	CROSS JOIN	FULL OUTER JOIN
Result Size	Number of rows = rows in TableA × rows in TableB.	Number of rows = all matched rows + unmatched rows from both tables.
Typical Use Case	Generating all possible combinations (e.g., scheduling, permutations).	Combining all data from both tables, showing matches and non-matches.

Example:

CROSS JOIN:

```
SELECT a.*, b.*
FROM TableA a
CROSS JOIN TableB b;
```

FULL OUTER JOIN:

```
SELECT a.*, b.*
FROM TableA a
FULL OUTER JOIN TableB b ON a.id = b.a_id;
```

Summary:
- **CROSS JOIN** creates every possible pair of rows from both tables.
- **FULL OUTER JOIN** returns all rows from both tables, matching where possible, and filling with NULLs where there is no match.

3. Write a SQL query to retrieve the first and last names of employees along with the names of their managers (given `Employees` and `Managers` tables).

4. Write a SQL query to find the average salary for each department, given tables `Employees` (with `DepartmentID`) and `Departments` (with `DepartmentName`).

5. Write a SQL query to list all products that have never been ordered (products in a `Product` table with no matching rows in the `Orders` table).

6. Write a SQL query to list all employees who are also managers (for example, employees who appear as managers in the same table).

7. What is a `self-join`, and when might you use it? Provide an example scenario.

8. How would you join more than two tables in a single SQL query? What factors affect the performance when joining multiple tables?

9. Explain how an `OUTER JOIN` works when one side has no matching rows. How does this differ from an `INNER JOIN` in practice?

Working with Views

Stored Procedures & Functions

Triggers & Automation

Transactions & Concurrency Control

Performance Tuning & Query Optimization