ANKKIT KUMAR GUPPTA
DATA ANALYST | PROMPT ENGINEER

WITHOUT DATA, YOU'RE JUST ANOTHER PERSON WITH AN OPINION

# FROM WHERE TO ORDER BY
# [ DECODE SQL EXECUTION LOGIC TODAY! ]

The **execution order of SQL statements** defines how the SQL engine processes a query step by step to produce the desired result. Although SQL queries are written in a specific syntax, the database internally processes them in a logical order to optimize and execute them efficiently.

**Here's a quick explanation of each step:**

1. **FROM Clause**:
   - Identify the tables or views from which data will be retrieved.
   - *Example*: FROM Employees
2. **JOINS**:
   - Combine data from multiple tables if required.
   - *Example*: FROM Employees JOIN Departments ON Employees.DeptID = Departments.DeptID
3. **WHERE Clause**:
   - Filter rows based on conditions.
   - *Example*: WHERE Employees.Salary > 50000
4. **GROUP BY Clause**:
   - Group rows that have the same values in specified columns.
   - *Example*: GROUP BY Departments.DeptName
5. **HAVING Clause**:
   - Filter the grouped data (after the GROUP BY clause).
   - *Example*: HAVING COUNT(Employees.ID) > 5
6. **SELECT Clause**:
   - Choose the columns to display in the result.
   - *Example*: SELECT Departments.DeptName, COUNT(Employees.ID) AS EmployeeCount
7. **DISTINCT**:
   - Eliminate duplicate rows in the output.
   - *Example*: SELECT DISTINCT Departments.DeptName
8. **ORDER BY Clause**:
   - Sort the rows in ascending or descending order.
   - *Example*: ORDER BY EmployeeCount DESC
9. **LIMIT/OFFSET**:
   - Restrict the number of rows displayed and skips a specified number of rows.
   - *Example*: LIMIT 10 OFFSET 5

This order ensures that data is filtered, grouped, and processed in a logical and efficient manner, producing accurate results for complex queries. **Understanding this sequence is key to writing optimized SQL queries.**

## Full Example's Query:

```
SELECT DISTINCT Departments.DeptName, COUNT(Employees.ID) AS EmployeeCount
FROM Employees
JOIN Departments ON Employees.DeptID = Departments.DeptID
WHERE Employees.Salary > 50000
GROUP BY Departments.DeptName
HAVING COUNT(Employees.ID) > 5
ORDER BY EmployeeCount DESC
LIMIT 10 OFFSET 5;
```

## Explanation of the Query

- **FROM**: The query retrieves data from the Employees and Departments tables.
- **JOIN**: Combines Employees and Departments on the DeptID.
- **WHERE**: Filters employees whose salary is greater than 50,000.
- **GROUP BY**: Groups employees by their department names.
- **HAVING**: Retains only departments with more than 5 employees.
- **SELECT**: Chooses distinct department names and the count of employees.
- **ORDER BY**: Sorts the results by employee count in descending order.
- **LIMIT/OFFSET**: Displays a maximum of 10 rows starting from the 6th row.

# ANKKIT KUMAR GUPPTA

DATA ANALYST | PROMPT ENGINEER

**WITHOUT DATA, YOU'RE JUST ANOTHER PERSON WITH AN OPINION**

If you would like to learn more & stay updated, please follow me on LinkedIn