SQL TIPS AND TRICKS

PART 24

SQL Order Of Execution

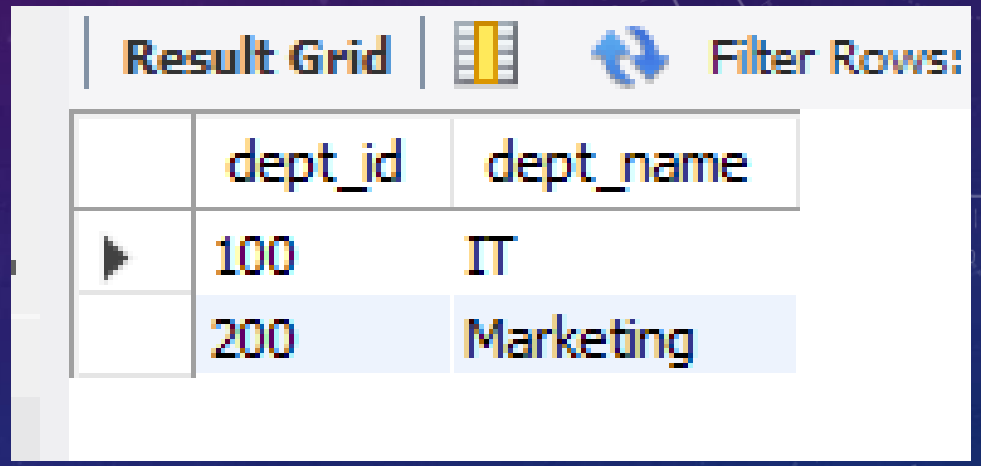
MAYURI DANDEKAR

Example--

emp

emp_id	department_id	salary
1	100	11000
2	100	16500
3	100	22000
4	200	30000
5	200	12000
6	200	14400
7	200	14400
8	200	6000
9	300	15000

dept



dept_id	dept_name
100	IT
200	Marketing

```
5
6 •   select   e.emp_id, e.department_id, d.dept_name, (e.salary) as dept_salary
7     from emp e
8     inner join dept d on e.department_id= d.dept_id
9     where e.salary > 6000
10    group by e.emp_id, e.department_id, d.dept_name, dept_salary
11    having sum(e.salary) >1000
12    order by dept_salary desc
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_id	department_id	dept_name	dept_salary
4	200	Marketing	30000
3	100	IT	22000
2	100	IT	16500
6	200	Marketing	14400
7	200	Marketing	14400
5	200	Marketing	12000
1	100	IT	11000

The order of execution in SQL, especially in a typical SELECT statement, generally follows these logical steps:

1. **FROM**: The first step involves identifying the tables involved in the query. The database engine locates the specified tables and prepares to retrieve data from them.

2. **JOIN**: If there are multiple tables involved and they need to be joined, this step occurs next. The JOIN operation combines rows from two or more tables based on a related column between them.

3. **WHERE**: Once the tables are identified and joined, the WHERE clause filters the rows returned by the JOIN operation. Rows that do not meet the specified conditions are excluded from the result set.

4. **GROUP BY**: If the query includes a GROUP BY clause, the rows are grouped into sets based on the values of one or more columns. This is often used with aggregate functions like COUNT, SUM, AVG, etc.

5. **HAVING**: The HAVING clause filters the grouped rows based on specified conditions. It's similar to the WHERE clause but operates on grouped rows rather than individual rows.

6. **SELECT**: After the rows are filtered, grouped, and sorted (if necessary), the SELECT clause specifies which columns should be included in the final result set. Expressions, calculations, and aliases defined in the SELECT clause are also evaluated at this stage.

7. **DISTINCT**: If the query includes the DISTINCT keyword, duplicate rows are removed from the result set.

8. ORDER BY: If the query includes an ORDER BY clause, the result set is sorted based on the specified column(s) and sort order.

9. **LIMIT/OFFSET**: Finally, if the query includes LIMIT and OFFSET clauses, the result set is limited to a specified number of rows, starting from a specified offset.

It's important to note that not all queries will involve all of these steps. The order and presence of these steps depend on the specific components of the query. For instance, if a query doesn't involve any joins, the JOIN step would be skipped. Similarly, if there's no GROUP BY clause, the GROUP BY step wouldn't occur.

THANK YOU

MAYURI DANDEKAR