Limit, **Order By,** Group By

PRESENTED BY **FILSAN MUSA & FADUMO DIRIYE**

TABLE OF CONTENT

LIMIT Clause

Definition: The LIMIT clause restricts the number of rows returned by a query.

```
SELECT *
FROM table_a
LIMIT 10;
```

```
SELECT column_1, column_2
FROM table_a
LIMIT 10;
```

Retrieves the first **10 rows** of all columns from **table_a.**

Retrieves the first **10 rows** for **column_1** and **column_2** from **table_a.**

Best Practice Tip: Use LIMIT when dealing with large datasets to improve performance.

ORDER BY Clause

Definition: The ORDER BY clause sorts query results based on specified columns.

```
SELECT column_1, column_2
FROM table_a
ORDER BY column_1 ASC, column_2 DESC;
```

```
SELECT column_1, column_2
FROM table_a
ORDER BY 1 ASC, 2 DESC;
```

Use Case: Arrange data in ascending (ASC) or descending (DESC) order.

GROUP BY Clause

Definition: The GROUP BY clause groups rows that have the same values in specified columns.

Use Case: Aggregate data (e.g., calculate totals, averages, counts).

```
SELECT column_1, SUM(column_2)
FROM table_a
GROUP BY column_1;
```

Explanation:
- Groups rows by column_1.
- Applies aggregate function (SUM) to grouped rows in column_2.

Combining LIMIT, ORDER BY, and GROUP BY

```sql
SELECT customer_id, SUM(purchase_amount)

FROM sales

GROUP BY customer_id

ORDER BY SUM(purchase_amount) DESC

LIMIT 5;
```

Best Practice Tip: Always order results before applying LIMIT for meaningful subsets.

Explanation:
- Groups sales data by customer_id.
- Sums purchase_amount for each customer.
- Orders the result by total purchases (highest first).
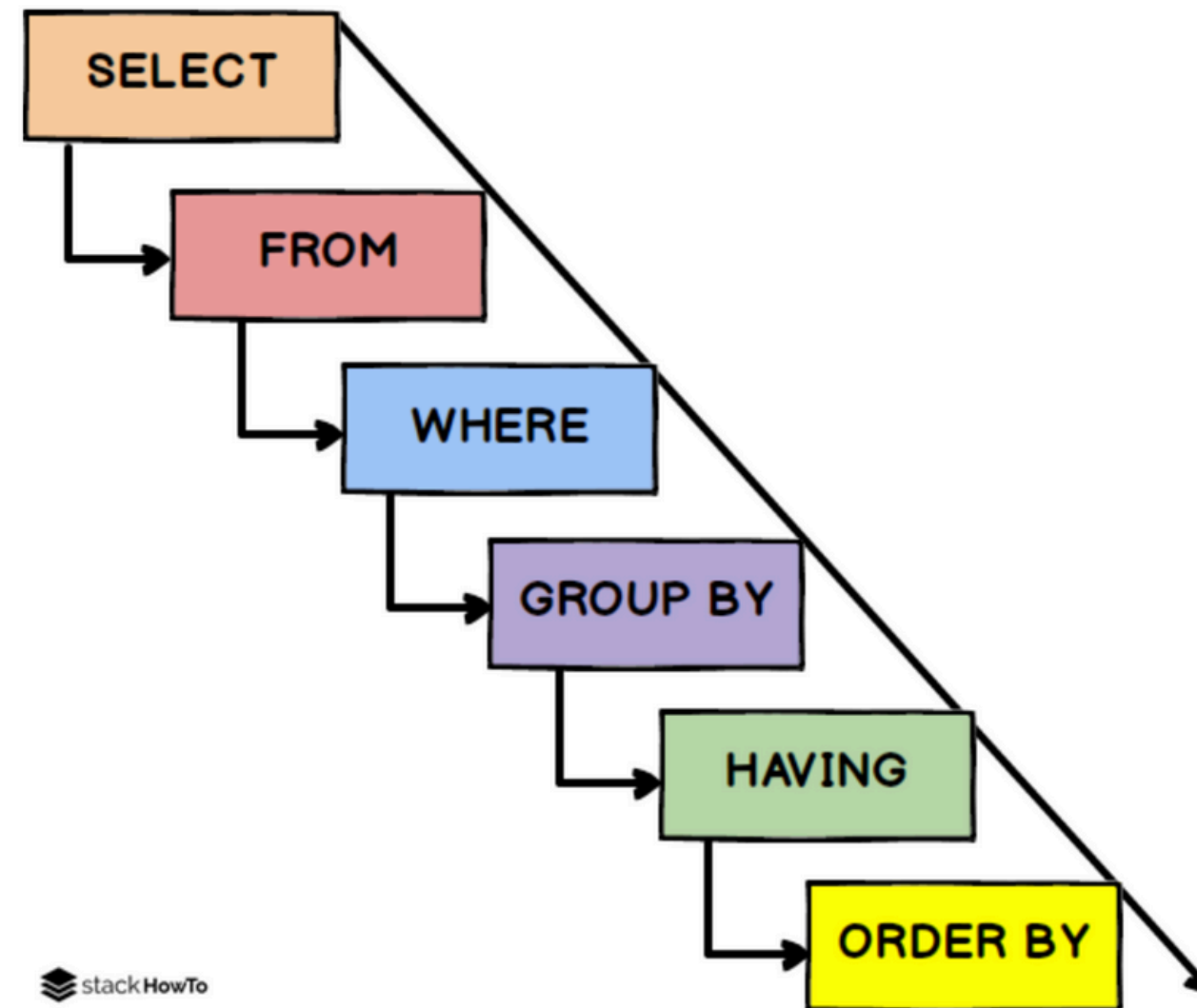- Limits output to the top 5 customers.

Key Differences Between ORDER BY and GROUP BY

Feature	ORDER BY	GROUP BY
Purpose	Sort rows	Group rows
Aggregate Functions	Optional	Commonly used
Resulting Rows	All rows remain	Reduced to groups
Example	Sort by column value	Aggregate by column value

Common Errors & Best Practices

Common Mistakes:

- Using GROUP BY without aggregate functions.
- Applying LIMIT before ORDER BY (it won't affect ordering).

Best Practices:

- Always pair GROUP BY with aggregate functions.
- Apply LIMIT after sorting with ORDER BY.

Recap and Next Steps

Execution Order:

1️⃣ GROUP BY → 2️⃣ ORDER BY → 3️⃣ LIMIT

Key Takeaways:
- Use LIMIT to reduce dataset size.
- Apply ORDER BY to sort data meaningfully.
- Leverage GROUP BY for aggregating grouped results.

- Next Steps:
  - Complete Week 9 Quiz