

Stop Guessing, Start Querying: Your Data Awaits.



The SELECT & FROM

- Retrieves data from specified columns in a table.
- FROM pinpoints the exact table you want to query.
- Use SELECT * to grab every column available.
- Always start your query with these two clauses.
- They form the foundational structure of almost every SQL statement.
- Combine them with other clauses to filter and sort.



WHERE Filters Data

- Narrows down results based on specific conditions.
- Use operators like =, <>, >, <, >=, <=.
- Filter text strings using the LIKE operator.
- Combine multiple conditions with AND & OR.
- Isolate exact records you need for your analysis.
- Crucial for answering specific business questions.



Aggregate with **GROUP BY**

- Groups rows that have the same values into summary rows.
- Essential for calculating metrics by categories (e.g., sales by region).
- Often used with aggregate functions like `COUNT()`, `SUM()`, `AVG()`.
- Turns detailed data into high-level insights.
- Helps in identifying trends across different segments.
- Remember: any column in `SELECT` not aggregated must be in `GROUP BY`.



Filter Groups with HAVING

- Filters records after the GROUP BY has been applied.
- WHERE filters rows, HAVING filters aggregated groups.
- Use it to find groups meeting a specific condition.
- Example: finding regions with total sales greater than \$1M.
- It works on the result of aggregate functions.
- Perfect for excluding insignificant data groups.



ORDER BY for Sorting

- Sorts the result-set in ascending (ASC) or descending (DESC) order.
- Essential for ranking top performers or worst cases.
- Default sort order is ascending if not specified.
- You can sort by one or multiple columns.
- Use it to organize data for clear presentation.
- Makes reports and exported data much more readable.



JOIN Tables Together

- Combines rows from two or more tables based on a related column.
- INNER JOIN returns records with matching values in both tables.
- LEFT JOIN returns all records from the left table and matched ones from the right.
- Crucial for working with relational databases.
- Allows you to create a unified dataset from separate sources.
- Understand your keys (Primary & Foreign) to use it correctly.



Deduplicate with DISTINCT

- Returns only unique values from a column or set of columns.
- Eliminates duplicate rows from your query results.
- Useful for finding unique entries, like a list of all customers.
- Placed right after the SELECT keyword.
- Helps in assessing the cardinality of your data.
- Use it to avoid skewed counts in your aggregations.



COUNT for Totals

- Returns the number of rows that match a specified criterion.
- **COUNT(*)** counts all rows in the table.
- **COUNT(column_name)** counts non-NULL values in that column.
- Fundamental for calculating volumes and totals.
- Often the first metric you check in a dataset.
- Combine with **GROUP BY** for counts by category.



SUM, AVG, MIN/MAX

- **SUM()** adds together all values in a numeric column.
- **AVG()** calculates the average value of a numeric column.
- **MIN()** and **MAX()** find the smallest and largest values.
- These are your core tools for quantitative analysis.
- Use them to calculate revenue, average order value, etc.
- Provide quick insights into performance metrics.



CASE for Logic

- The SQL way of writing IF-THEN-ELSE statements.
- Creates new columns based on conditional logic.
- Can categorize data into custom buckets or flags.
- Powerful for feature engineering within a query.
- Example: classifying customers as 'New' or 'Returning'.
- Makes your data more insightful and ready for reporting.



Alias with AS

- Gives a temporary, simpler name to a table or column.
- Makes complex queries more readable and easier to write.
- Use it for calculated fields or long table names.
- Column aliases are great for renaming outputs in your results.
- Table aliases are essential when working with multiple joins.
- Improves clarity and reduces typing errors.



FOLLOW FOR MORE



Like this post? Repost!

Share your thoughts in the comments!

