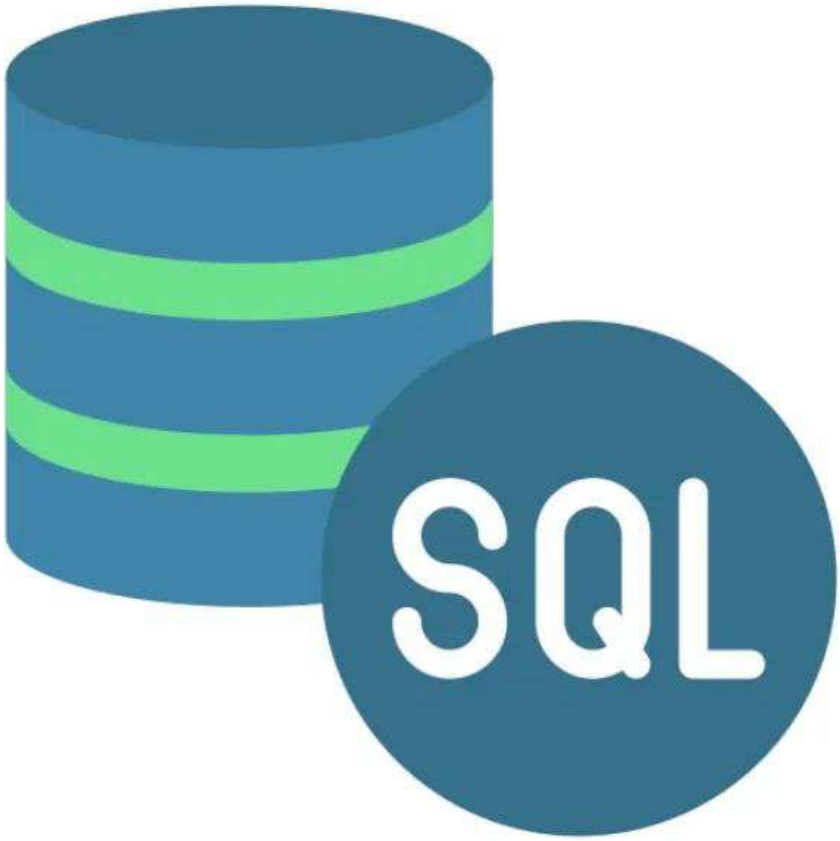
Harry
coding_knowledge

SQL Commands



Swipe

Harry
coding_knowledge

- **SELECT**: Retrieve data from a database.
- **FROM**: Specify the tables) to retrieve data from.
- **WHERE**: Filter rows based on a specified condition.
- **GROUP BY**: Group rows that have the same values into summary rows.
- **HAVING**: Filter group rows based on specified conditions.
- **ORDER BY:** Sort the result set in ascending or descending order.
- **INNER JOIN**: Combine rows from two or more tables based on a related column between them.
- **LEFT JOIN:** Return all rows from the left table, and the matched rows from the right table.
- **RIGHT JOIN:** Return all rows from the right table, and the matched rows from the left table.
- **FULL JOIN:** Return all rows when there is a match in either table.
- **CROSS JOIN:** Produce a Cartesian product of two tables.
- **UNION:** Combine the result sets of two or more SELECT statements.
- **DISTINCT:** Return unique values in the result set.
- **COUNT():** Return the number of rows that match a specified condition.
- **SUM()**: Calculate the sum of values in a column.

Swipe →

Harry
coding_knowledge

- **AVG():** Calculate the average of values in a column.
- **MIN()**: Return the minimum value in a column.
- **MAX()**: Return the maximum value in a column.
- **BETWEEN:** Select values within a given range.
- **LIKE:** Search for a pattern in a column.
- **IN**: Specify multiple values in a WHERE clause.
- **IS NULL**: Filter rows where a column is NULL.
- **IS NOT NULL**: Filter rows where a column is not NULL.
- **AS**: Rename a column or table with an alias.
- **INNER SELECT**: Perform a subquery within a SELECT statement.
- **CASE**: Perform conditional logic in SQL queries.
- **EXISTS:** Test for the existence of rows in a subquery result.
- **ANY/ALL:** Compare a value to a set of values.
- **LIMIT**: Limit the number of rows returned by a query.
- **OFFSET**: Skip a specified number of rows before returning the result set.
- **ROW _NUMBER()**: Assign a unique sequential integer to each row in the result set.
- **RANK()**: Assign a rank to each row in the result set.
- **DENSE_RANK():** Assign a dense rank to each row in the result set.
- **NTILE():** Divide the result set into a specified number of groups.
- **LEAD()**: Access data from a subsequent row in the result set.

Swipe →

Harry
coding_knowledge

- **LAG()**: Access data from a previous row in the result set.
- **OVER()**: Specify a window of rows for aggregate functions.
- **PARTITION BY**: Divide the result set into partitions to which the function is applied separately.
- **ORDER BY** (Window Function): Specify the order of rows within the partition for window functions.
- **FIRST_VALUE()**: Return the first value in an ordered set of values.
- **LAST_VALUE():** Return the last value in an ordered set of
- values.
- **PERCENTILE_DISC()**: Calculate a specific percentile value in a sorted set of values.
- **PERCENTILE_CONT():** Calculate a percentile value using linear interpolation.
- **FETCH:** Retrieve a specific number of rows from a query result.
- **EXPLAIN:** Provide information on how a SQL statement will be executed.
- **TRANSACTION:** Begin, commit, or rollback a transaction.
- **CREATE TABLE:** Create a new table in the database.
- **ALTER TABLE:** Modify an existing table's structure.
- **DROP TABLE**: Delete a table from the database.
- **CREATE INDEX:** Create an index on a table for faster data retrieval.

Swipe →