



Vikram
@code_learning

100 SQL Commands



- **SELECT** - retrieves data from a database
- **INSERT** - inserts new data into a database
- **UPDATE** - updates existing data in a database
- **DELETE** - deletes data from a database
- **CREATE DATABASE** - creates a new database
- **CREATE TABLE** - creates a new table in a database
- **ALTER TABLE** - modifies an existing table structure
- **DROP TABLE** - deletes a table from a database
- **TRUNCATE TABLE** - removes all records from a table
- **CREATE INDEX** - creates an index on a table
- **DROP INDEX** - deletes an index from a table
- **JOIN** - combines rows from two or more tables based on a related column
- **INNER JOIN** - returns rows when there is a match in both tables
- **LEFT JOIN** - returns all rows from the left table, and the matched rows from the right table
- **RIGHT JOIN** - returns all rows from the right table, and the matched rows from the left table

@code._learning



- **FULL JOIN** - returns rows when there is a match in one of the tables
- **UNION** - combines the results of two or more SELECT statements
- **UNION ALL** - combines the results of two or more SELECT statements, including duplicates
- **GROUP BY** - groups rows that have the same values into summary rows @code._learning
- **HAVING** - filters records based on a specified condition
- **ORDER BY** - sorts the result set in ascending or descending order
- **COUNT** - returns the number of rows that satisfy the condition
- **SUM** - calculates the sum of a set of values
- **AVG** - calculates the average of a set of values
- **MIN** - returns the smallest value in a set of values
- **MAX** - returns the largest value in a set of values
- **DISTINCT** - selects unique values from a column
- **WHERE** - filters records based on specified conditions



- **AND** - combines multiple conditions in a WHERE clause
- **OR** - specifies multiple alternative conditions in a WHERE clause
- **NOT** - negates a condition in a WHERE clause
- **BETWEEN** - selects values within a specified range
- **IN** - specifies multiple values for a column
- **LIKE** - selects rows that match a specified pattern
- **IS NULL** - checks for NULL values in a column
- **IS NOT NULL** - checks for non-NULL values in a column
- **EXISTS** - tests for the existence of any record in a subquery
- **CASE** - performs conditional logic in SQL statements
- **WHEN** - specifies conditions in a CASE statement
- **THEN** - specifies the result if a condition is true in a CASE statement
- **ELSE** - specifies the result if no condition is true in a CASE statement
- **END** - ends the CASE statement

@code._learning



- **PRIMARY KEY** - uniquely identifies each record in a table
- **FOREIGN KEY** - establishes a relationship between tables
- **CONSTRAINT** - enforces rules for data in a table
- **DEFAULT** - specifies a default value for a column
- **NOT NULL** - ensures that a column cannot contain NULL values
- **UNIQUE** - ensures that all values in a column are unique
- **CHECK** - enforces a condition on the values in a column
- **CASCADE** - automatically performs a specified action on related records
- **SET NULL** - sets the value of foreign key columns to NULL when a referenced record is deleted
- **SET DEFAULT** - sets the value of foreign key columns to their default value when a referenced record is deleted
- **NO ACTION** - specifies that no action should be taken on related records when a referenced record is deleted

@code._learning

