

All SQL Queries

Query	Description	Example
1) SELECT	Retrieves data from one or more tables.	'SELECT name, age FROM Users;'
2) INSERT	Adds new records to a table	'INSERT INTO users (name, age) VALUES ('Alice', 30);'
3) UPDATE	Modifies existing records in a table.	'UPDATE users SET age = 31 WHERE name
4) DELETE	Removes records from a table	'DELETE FROM users WHERE name = 'Alice';'
5) CREATE TABLE	Creates a new table in the database.	'CREATE TABLE users (id INT PRIMARY KEY, name VARCHAR(100), age INT);'
6) ALTER TABLE	Modifies an existing table structure, such as adding or dropping columns.	'ALTER TABLE users ADD email VARCHAR(100);'

SQL Query	Description	Example
7) DROP TABLE	Deletes a table and all its data.	'DROP TABLE users;'
8) CREATE DATABASE	Creates a new database	'CREATE DATABASE my-database;'
9) DROP DATABASE	Deletes a database and all its data.	'DROP Database my-database;'
10) SELECT DISTINCT	Retrieves unique records from a table	'SELECT DISTINCT age FROM users;'
11) WHERE	Filters records that meet a specific condition.	'SELECT name FROM users WHERE age > 25;'
12) UNION	Combines the result sets of two or more SELECT queries.	'SELECT name FROM users UNION SELECT name FROM employee;'
13) SUBQUERY	A query nested inside another query.	'SELECT name FROM users WHERE age = (SELECT MAX (age) FROM users;)'

Query	Description	Example
14) EXISTS	check if a subquery returns any record	'SELECT name FROM users WHERE age = (SELECT MAX (age) FROM users);'
15) ANY	Compares a value to any value in a list or subquery	'SELECT name FROM users WHERE age > ANY (SELECT age FROM users WHERE age < 30);'
16) ALL	Compares a value to all values in a list or subquery	'SELECT name FROM users WHERE age > ALL (SELECT age FROM users WHERE age < 30);'
17) LIKE	Searches for a specified pattern in a column.	'SELECT name FROM users WHERE name LIKE 'A%';'
18) IN	checks if a value is within a set of values.	'SELECT name FROM users WHERE age IN (25, 30, 35);'

Query	Description	Example
19) BETWEEN	selects value within a range.	'SELECT name FROM users WHERE age BETWEEN 20 AND 30;'
20) LIMIT	Specifies the number of records to return.	'SELECT name FROM users LIMIT 5;'
21) ORDER BY	Sorts the result set by one or more columns.	'SELECT name, age FROM users ORDER BY age DESC;'
22) GROUP BY	Groups rows that have the same values into summary rows.	'SELECT age, COUNT (*) FROM users GROUP BY age;'
23) HAVING	Filters groups based on a condition.	'SELECT age, COUNT (*) FROM users GROUP BY age HAVING COUNT(*) > 1;'
24) JOIN	Combines rows from two or more tables based on a related column.	'SELECT users.name, orders.amount FROM users INNER JOIN orders ON'

Query	Description	Example
		on Users.id = orders.user-id;
25) INNER JOIN	Retrieves records with matching values in both tables.	'SELECT users.name, orders.amount FROM users INNER JOIN orders ON users.id= orders.user-id;'
26) LEFT JOIN	Retrieves all records from the left table and matched records from the right table.	'SELECT users.name, orders.amount FROM users LEFT JOIN orders ON users.id= orders.user-id;'
27) RIGHT JOIN	Retrieves all records from all the right table and matched records from the left table.	'SELECT users.name, orders.amount FROM users RIGHT JOIN orders ON users.id= orders.user-id;'
28) FULL OUTER JOIN	Retrieves records when there is a match in one of the tables.	'SELECT users.name, orders.amount FROM users FULL OUTER JOIN orders ON users. id = orders.user-id;'

Query	Description	Example
29) LIMIT	Specifies the number of records to return.	'SELECT name FROM users LIMIT 5;'
30) OFFSET	Specifies the starting point for records to return.	'SELECT name FROM users LIMIT 5 OFFSET 10;'
31) TRUNCATE TABLE	Removes all records from a table, structure	'TRUNCATE TABLE users;'
32) CREATE INDEX	Creates an index on a table to improve query performance	'CREATE INDEX idx_name ON users (name);'
33) DROP INDEX	Deletes an index from a table.	'DROP INDEX idx_name;'
34) CASE	Provides conditional logic in SQL queries.	'SELECT name, age CASE WHEN age < 30 THEN 'Young' ELSE 'Old' END as age_group FROM users;'
35) CAST	Converts a value from one data type to another.	'SELECT name, CAST (age AS CHAR) FROM users;'