

Major sql operation:

Structure of DB -

Database → table → row, column
 ↓
 one or multiple

hold multiple
table

"The SELECT statement retrieves data from columns that exist within tables."

SQL Query Methods Summary

This document provides an overview of major SQL query methods, categorized by their function.

Select ~~one or multiple~~
 table ~~to column~~ ~~multiple~~
 Data retrieve ~~time~~

{combine all row is}
 {called a record}

1 Data Query Language (DQL) — Used to retrieve data
 Main Command: A single table, multiple table,
 • SELECT — Retrieves data from one or more tables.

Common Clauses with SELECT:

| Clause | Purpose |
|----------------------|--|
| FROM | Specifies the table(s) to query |
| WHERE | Filters records based on conditions |
| GROUP BY | Groups rows sharing common column values |
| HAVING | Filters groups after aggregation |
| ORDER BY | Sorts result set |
| LIMIT / TOP / ROWNUM | Limits number of returned rows |
| DISTINCT | Returns unique values |
| JOIN | Combines rows from two or more tables |

Starting point

| Command | Purpose |
|----------------------------|---|
| INSERT INTO | Adds new records to a table |
| UPDATE | Modifies existing records |
| DELETE | Removes records from a table |
| MERGE (Oracle, SQL Server) | Performs insert, update, or delete in one command |

DML
 |
 manipulate

3 Data Definition Language (DDL) — Used to define the structure of the database

Commands:

| Command | Purpose |
|----------|---|
| CREATE | Creates objects like tables, indexes, views, etc. |
| ALTER | Modifies the structure of an existing object |
| DROP | Deletes an object |
| TRUNCATE | Removes all records from a table (faster than DELETE) |
| RENAME | Renames an object (varies by DBMS) |

```

1. CREATE TABLE
sql
2. CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    Name VARCHAR(50),
    Age INT,
    Email VARCHAR(50),
    Phone VARCHAR(15),
    FeePaid DATE
);

2. ALTER TABLE
sql
3. ALTER TABLE Students
    ADD NewColumn;
    ALTER TABLE Students
    DROP NewColumn;

4. DROP TABLE
sql
If the table is no longer needed
5. USE NewStudents;
6. TRUNCATE TABLE
To remove all student records but keep the table structure
7. TRUNCATE TABLE Students;

```

DELETE in DML delete on record(row)

4 Data Control Language (DCL) — Used to control access to data

Commands:

| Command | Purpose |
|---------|---|
| GRANT | Gives user privileges to database objects |
| REVOKE | Removes previously granted privileges |

5 Transaction Control Language (TCL) — Used to manage transactions

Commands:

| Command | Purpose |
|-----------------|--|
| COMMIT | Saves transaction changes |
| ROLLBACK | Undoes transaction changes |
| SAVEPOINT | Sets a point in the transaction to roll back to |
| SET TRANSACTION | Sets transaction properties like isolation level |

6 Common SQL Functions & Operators

Aggregate Functions:

counts the number of rows that match a specified condition.

| Function | Purpose |
|----------|-----------------------|
| COUNT() | Counts number of rows |
| SUM() | Sums values |
| AVG() | Calculates average |
| MAX() | Gets maximum value |
| MIN() | Gets minimum value |

Scalar Functions:

| Function | Purpose |
|--------------------------------|---------------------------|
| UPPER(), LOWER() | Converts case |
| SUBSTRING(), LEFT(), RIGHT() | Extracts parts of strings |
| ROUND(), CEIL(), FLOOR() | Math functions |
| COALESCE(), IFNULL(), ISNULL() | Handles NULL values |

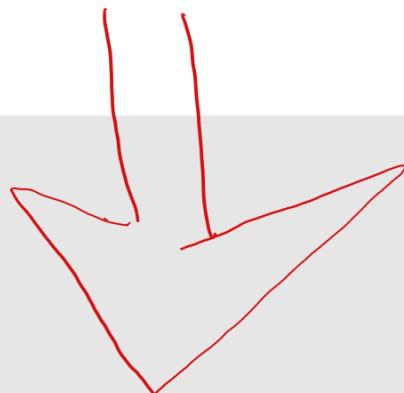
Operators:

- Comparison: =, <>, <, >, <=, >=
- Logical: AND, OR, NOT
- Special: IN, BETWEEN, LIKE, IS NULL

Summary Table

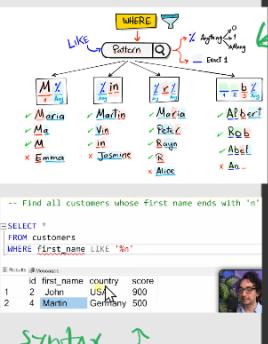
| Category | Commands/Methods |
|----------|-------------------------------------|
| DQL | SELECT, JOIN, WHERE, GROUP BY, etc. |

| Category | Commands/Methods |
|-----------|-------------------------------|
| DML | INSERT, UPDATE, DELETE, MERGE |
| DDL | CREATE, ALTER, DROP, TRUNCATE |
| DCL | GRANT, REVOKE |
| TCL | COMMIT, ROLLBACK, SAVEPOINT |
| Functions | Aggregate, Scalar, Window |



Sql operations
 more details

| Category | Operator | Description | Example |
|-----------------|----------------|---|--|
| 💡 Arithmetic | + | Addition | SELECT price + tax FROM orders; |
| | - | Subtraction | SELECT price - discount FROM products; |
| | * | Multiplication | SELECT quantity * price FROM items; |
| | / | Division | SELECT total / quantity FROM sales; |
| | % / MOD | Modulus (remainder) | SELECT 10 % 3; |
| ⚖️ Comparison | = | Equal to | WHERE age = 25; |
| | <> / != | Not equal to | WHERE status <> 'inactive'; |
| | > | Greater than | WHERE salary > 50000; |
| | < | Less than | WHERE rating < 3.5; |
| | >= | Greater than or equal to | WHERE age >= 18; |
| | <= | Less than or equal to | WHERE price <= 100; |
| 🔍 Logical | AND | All conditions must be true | WHERE age > 18 AND status = 'active'; |
| | OR | Any condition can be true | WHERE city = 'NY' OR city = 'LA'; |
| | NOT | Negates a condition | WHERE NOT status = 'inactive'; |
| | BETWEEN | Checks if value is within a range | WHERE price BETWEEN 10 AND 50; |
| | IN | Checks if value is in a given list | WHERE id IN (1, 2, 3); |
| | LIKE | Pattern matching with % and _ | WHERE name LIKE 'J%'; |
| | IS NULL | Checks if value is NULL | WHERE email IS NULL; |
| | IS NOT NULL | Checks if value is not NULL | WHERE phone IS NOT NULL; |
| | EXISTS | True if subquery returns any rows | WHERE EXISTS (SELECT 1 FROM orders); |
| | | | |
| 🧠 Set Operators | UNION | Combines results, removes duplicates | SELECT a FROM t1 UNION SELECT b FROM t2; |
| | UNION ALL | Combines results, keeps duplicates | SELECT a FROM t1 UNION ALL SELECT b FROM t2; |
| | INTERSECT | Returns common rows from both queries | SELECT a FROM t1 INTERSECT SELECT b FROM t2; |
| | EXCEPT / MINUS | Rows from first query not in second (EXCEPT=Postgres, MINUS=Oracle) | SELECT a FROM t1 EXCEPT SELECT b FROM t2; |
| 💡 String | ' ' | | +/CONCAT()' |
| Category | Operator | Description | Example |
| ⚡ Bitwise | & | Bitwise AND | SELECT 5 & 3; → 1 |
| | | Bitwise OR | SELECT 5 3; → 7 |
| | ^ | Bitwise XOR | SELECT 5 ^ 3; → 6 |
| | ~ | Bitwise NOT (inverts bits) | SELECT ~1; |
| 🌐 Other | DISTINCT | Returns only unique values | SELECT DISTINCT city FROM customers; |
| | * | Selects all columns | SELECT * FROM employees; |
| | CASE | Conditional branching logic | SELECT CASE WHEN age > 18 THEN 'Adult' ELSE 'Minor' END; |
| | :: / CAST() | Converts data from one type to another | SELECT price::INT FROM products; |
| | -> | Access JSON field (PostgreSQL only) | SELECT data->'name' FROM users; |



Syntax ↗

Broad Discussion about Syntax:

order by

sql

 Copy  Download

```
SELECT column1, column2, ...
FROM table_name
ORDER BY column1 [ASC|DESC], column2 [ASC|DESC], ...;
```

sql

 Copy  Download

```
ORDER BY country ASC, customer_name DESC
```

→ Second column works as ties breaker

Key Features

1. Sorting Direction:

- ASC (default): Sorts in ascending order (A-Z, 0-9)
- DESC: Sorts in descending order (Z-A, 9-0)

2. Multiple Columns:

You can sort by multiple columns (sorts by first column, then by second column for ties, etc.).

3. Positional Sorting:

~~Data Type~~ You can reference columns by their position in the SELECT list:

1 Integer Types

| Data Type | Description / Main Feature | Example Value |
|-----------|-----------------------------|---------------------|
| TINYINT | Very small integer (1 byte) | 127 |
| SMALLINT | Small integer (2 bytes) | 32767 |
| MEDIUMINT | Medium integer (3 bytes) | 8388607 |
| INT | Standard integer (4 bytes) | 2147483647 |
| BIGINT | Large integer (8 bytes) | 9223372036854775807 |

2 Floating-Point & Fixed-Point Types

| Data Type | Description / Main Feature | Example Value |
|---------------|--|---------------|
| FLOAT | Approximate single precision floating number | 3.14159 |
| DOUBLE / REAL | Approximate double precision floating number | 2.718281828 |
| DECIMAL(M,D) | Fixed precision decimal, exact for money | 12345.67 |

3 String Types

| Data Type | Description / Main Feature | Example Value |
|--------------------------------|---|---------------------|
| CHAR(n) | Fixed-length string; always uses n characters (padded if short) | 'USA' |
| VARCHAR(n) | Variable-length string; uses space equal to actual content length | 'nasim123' |
| TEXT | Large text (up to 64 KB); used for longer text | 'This is a comment' |
| TINYTEXT, MEDIUMTEXT, LONGTEXT | Text types for increasingly larger sizes | (long text) |
| ENUM | One value from a predefined set of strings | 'small' |
| SET | Multiple values from a predefined set of strings | 'a,b,c' |

4 Date and Time Types

| Data Type | Description / Main Feature | Example Value |
|-----------|-----------------------------------|-----------------------|
| DATE | Stores date only (YYYY-MM-DD) | '2025-07-26' |
| DATETIME | Stores date and time | '2025-07-26 10:30:00' |
| TIMESTAMP | Like DATETIME, often auto-updated | '2025-07-26 10:30:00' |
| TIME | Stores time only (HH:MM:SS) | '14:30:00' |
| YEAR | Stores year (YYYY) | '2025' |

5 Boolean Type

| Data Type | Description / Main Feature | Example Value |
|-----------|----------------------------------|---------------------|
| BOOLEAN | True/False, stored as TINYINT(1) | 1 (true), 0 (false) |

6 Binary and JSON Types

| Data Type | Description / Main Feature | Example Value |
|-------------------------------------|------------------------------------|---------------------|
| BLOB | Binary data, e.g. images or files | (binary data) |
| TINYBLOB , MEDIUMBLOB , LONGBLOB | BLOB variants with different sizes | (large binary data) |
| JSON | Stores JSON formatted data | '{"name": "Ali"}' |

sql

Copy Download

```
SELECT name, country FROM customers ORDER BY 2, 1
```

4. What is constraint in SQL?

- NULLs are considered the lowest possible values in ASC order
- NULLs appear last in DESC order

📌 What is a Constraint in SQL?

honesty, accuracy

A constraint is a rule applied to a column or table to enforce data integrity and correctness. It limits the kind of data that can be stored in the database to avoid errors and maintain consistency.

Integrity constraints in SQL are rules enforced on database tables to maintain data accuracy, consistency, and validity, such as ensuring unique primary keys and valid foreign key relationships. 19 Jun 2024

🔥 Common Types of Constraints

| Constraint | Purpose / What It Does | Example Usage |
|-------------|---|---|
| PRIMARY KEY | Uniquely identifies each row in a table | <code>id INT PRIMARY KEY</code> |
| FOREIGN KEY | Ensures referential integrity between tables | <code>FOREIGN KEY (user_id) REFERENCES users(id)</code> |
| UNIQUE | Ensures all values in a column are unique | <code>email VARCHAR(100) UNIQUE</code> |
| NOT NULL | Prevents a column from having NULL (empty) values | <code>name VARCHAR(50) NOT NULL</code> |
| CHECK | Ensures values satisfy a condition | <code>age INT CHECK (age >= 18)</code> |
| DEFAULT | Sets a default value if none is provided | <code>status VARCHAR(10) DEFAULT 'active'</code> |

🎯 Why Use Constraints?

- Prevent invalid data entry
- Keep relationships consistent
- Enforce business rules automatically
- Improve data reliability

