1. Introduction to SQL

What is SQL?
Types of SQL:

- **DDL (Data Definition Language)** – `CREATE`, `ALTER`, `DROP`, `TRUNCATE`
- **DML (Data Manipulation Language)** – `SELECT`, `INSERT`, `UPDATE`, `DELETE`
- **DCL (Data Control Language)** – `GRANT`, `REVOKE`
- **TCL (Transaction Control Language)** – `COMMIT`, `ROLLBACK`, `SAVEPOINT`

  SQL vs NoSQL

---

2. Database Basics

What is a database?
Types of databases (Relational & Non-relational)
Popular relational databases: MySQL, PostgreSQL, SQL Server, Oracle, SQLite

---

3. SQL Basics (CRUD Operations)

Creating a database & tables (`CREATE DATABASE`, `CREATE TABLE`)
Inserting data (`INSERT INTO`)
Reading data (`SELECT`)
Updating data (`UPDATE`)
Deleting data (`DELETE`)

Example:

```
sql
CopyEdit
CREATE TABLE employees (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    department VARCHAR(50)
);

INSERT INTO employees (id, name, age, department)
VALUES (1, 'Alice', 30, 'IT');
```

---

4. Filtering Data

WHERE Clause – Filtering rows
Operators:

- Arithmetic (`+`, `-`, `*`, `/`)
- Comparison (`=`, `!=`, `>`, `<`, `>=`, `<=`)

- Logical (`AND`, `OR`, `NOT`)

    BETWEEN, IN, LIKE, IS NULL

Example:

```
sql
CopyEdit
SELECT * FROM employees WHERE age > 25 AND department = 'IT';
```

---

5. Sorting and Aggregation

ORDER BY – Sorting results (`ASC`, `DESC`)

GROUP BY – Grouping data

HAVING – Filtering groups

Aggregate functions:

- `COUNT()`, `SUM()`, `AVG()`, `MIN()`, `MAX()`

Example:

```
sql
CopyEdit
SELECT department, COUNT(*) AS total_employees
FROM employees
GROUP BY department
HAVING COUNT(*) > 5
ORDER BY total_employees DESC;
```

---

6. Joins in SQL

Types of Joins:

- **INNER JOIN** – Matches rows in both tables
- **LEFT JOIN** – All rows from the left table, matching from right
- **RIGHT JOIN** – All rows from the right table, matching from left
- **FULL OUTER JOIN** – Combines both tables
- **SELF JOIN** – Joins the table with itself

    Using ON and USING clauses

Example:

```
sql
CopyEdit
SELECT e.name, d.department_name
FROM employees e
INNER JOIN departments d ON e.department = d.id;
```

---

7. Subqueries & CTEs

Subqueries (Nested Queries)
Common Table Expressions (CTEs)

Example:

```
sql
CopyEdit
WITH employee_count AS (
    SELECT department, COUNT(*) AS total FROM employees GROUP BY department
)
SELECT * FROM employee_count WHERE total > 10;
```

---

8. Indexes & Performance Optimization

What are indexes?
Clustered vs Non-clustered Indexes
Creating and Dropping Indexes
Query Optimization Techniques
EXPLAIN / ANALYZE Statements

Example:

```
sql
CopyEdit
CREATE INDEX idx_employee_name ON employees (name);
```

---

9. Views & Stored Procedures

Creating Views (`CREATE VIEW`)
Stored Procedures (`CREATE PROCEDURE`)
Functions (`CREATE FUNCTION`)
Triggers (`CREATE TRIGGER`)

Example:

```
sql
CopyEdit
CREATE VIEW active_employees AS
SELECT * FROM employees WHERE status = 'active';
```

---

10. Transactions & ACID Properties

What are transactions?
ACID Properties (Atomicity, Consistency, Isolation, Durability)
Using COMMIT, ROLLBACK, and SAVEPOINT

Example:

```
sql
CopyEdit
BEGIN TRANSACTION;
UPDATE accounts SET balance = balance - 100 WHERE id = 1;
UPDATE accounts SET balance = balance + 100 WHERE id = 2;
COMMIT;
```

---

11. Advanced SQL Topics

Partitioning Tables (`PARTITION BY`)
JSON Functions in SQL
Full-Text Search
Materialized Views
Recursive Queries

---

12. SQL for Data Analysis

Window Functions (`ROW_NUMBER()`, `RANK()`, `DENSE_RANK()`)
PIVOT and UNPIVOT Operations
Case Statements

Example:

```
sql
CopyEdit
SELECT name, department,
       RANK() OVER(PARTITION BY department ORDER BY age DESC) as rank
FROM employees;
```

How to Learn SQL Effectively?

Practice with Real Data – Use sample datasets like `Sakila`, `Chinook`, or `AdventureWorks`.
Work on Projects – Build a mini database for inventory, employee management, or e-commerce.
Use Online Platforms – Leetcode (SQL), HackerRank, Mode Analytics.
Explore Advanced Topics – Learn **PL/SQL** (Oracle) or **T-SQL** (SQL Server).

SQL (Structured Query Language) - Detailed Guide

1. Introduction to SQL

SQL (**Structured Query Language**) is a programming language used to **manage and manipulate relational databases**. It allows users to store, retrieve, modify, and delete data efficiently.

Types of SQL Commands:

SQL commands are categorized into four main types:

1.1 DDL (Data Definition Language)

DDL commands are used to **define, alter, or delete** the structure of database objects (tables, indexes, views, etc.).

- `CREATE` – Creates tables, databases, indexes, etc.
- `ALTER` – Modifies an existing table structure.
- `DROP` – Deletes a database object.
- `TRUNCATE` – Deletes all records but keeps the structure.

Example:

```
sql
CopyEdit
CREATE TABLE employees (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    department VARCHAR(50)
);
```

1.2 DML (Data Manipulation Language)

DML commands deal with the **manipulation of data** (inserting, updating, and deleting records).

- `INSERT` – Adds new data into a table.
- `UPDATE` – Modifies existing records.
- `DELETE` – Removes records.

Example:

```
sql
CopyEdit
INSERT INTO employees (id, name, age, department) VALUES (1, 'Alice', 30, 'IT');
UPDATE employees SET age = 32 WHERE id = 1;
DELETE FROM employees WHERE id = 1;
```

1.3 DCL (Data Control Language)

DCL commands are used for **user access control**.

- `GRANT` – Provides privileges to users.
- `REVOKE` – Removes privileges from users.

Example:

```
sql
CopyEdit
GRANT SELECT, INSERT ON employees TO user1;
REVOKE INSERT ON employees FROM user1;
```

1.4 TCL (Transaction Control Language)

TCL commands manage **transactions**.

- `COMMIT` – Saves changes permanently.
- `ROLLBACK` – Undoes changes.
- `SAVEPOINT` – Creates checkpoints for rollback.

Example:

```
sql
CopyEdit
BEGIN TRANSACTION;
UPDATE employees SET age = 35 WHERE id = 1;
ROLLBACK; -- Undo changes
```

---

2. Database Basics

A **database** is a collection of structured data stored electronically. SQL works with **Relational Databases**, where data is organized into **tables** (rows & columns).

Popular Relational Databases:

- **MySQL** – Open-source, widely used.
- **PostgreSQL** – Advanced features, open-source.
- **SQL Server** – Microsoft's enterprise solution.
- **Oracle SQL** – High-performance databases.
- **SQLite** – Lightweight database.

---

3. SQL Basics (CRUD Operations)

CRUD stands for:

- **C**reate → `INSERT`
- **R**ead → `SELECT`
- **U**pdate → `UPDATE`
- **D**elete → `DELETE`

3.1 Creating a Table

```
sql
CopyEdit
CREATE TABLE employees (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    age INT,
    department VARCHAR(50)
);
```

3.2 Inserting Data

```
sql
CopyEdit
INSERT INTO employees (id, name, age, department)
VALUES (1, 'Alice', 30, 'IT');
```

3.3 Reading Data (`SELECT`)

```
sql
CopyEdit
SELECT * FROM employees;
SELECT name, age FROM employees WHERE department = 'IT';
```

3.4 Updating Data

```
sql
CopyEdit
UPDATE employees SET age = 35 WHERE id = 1;
```

3.5 Deleting Data

```
sql
CopyEdit
DELETE FROM employees WHERE id = 1;
```

---

4. Filtering Data (`WHERE`, `BETWEEN`, `LIKE`, `IN`)

4.1 WHERE Clause

Filters data based on conditions.

```
sql
CopyEdit
SELECT * FROM employees WHERE age > 25;
```

4.2 BETWEEN

```
sql
CopyEdit
SELECT * FROM employees WHERE age BETWEEN 25 AND 35;
```

4.3 LIKE (Pattern Matching)

```
sql
CopyEdit
SELECT * FROM employees WHERE name LIKE 'A%'; -- Names starting with A
```

4.4 IN Clause

```
sql
CopyEdit
SELECT * FROM employees WHERE department IN ('IT', 'HR');
```

---

5. Sorting and Aggregation

5.1 ORDER BY (Sorting)

```
sql
CopyEdit
SELECT name, age FROM employees ORDER BY age DESC;
```

5.2 GROUP BY (Grouping)

```
sql
CopyEdit
SELECT department, COUNT(*) FROM employees GROUP BY department;
```

5.3 HAVING (Filtering Groups)

```
sql
CopyEdit
SELECT department, COUNT(*) FROM employees
GROUP BY department
HAVING COUNT(*) > 5;
```

---

6. Joins (Combining Tables)

6.1 INNER JOIN

```
sql
CopyEdit
SELECT e.name, d.department_name
FROM employees e
INNER JOIN departments d ON e.department = d.id;
```

6.2 LEFT JOIN

```
sql
CopyEdit
SELECT e.name, d.department_name
FROM employees e
LEFT JOIN departments d ON e.department = d.id;
```

---

7. Subqueries & CTEs

7.1 Subquery

```
sql
CopyEdit
SELECT name FROM employees WHERE age = (SELECT MAX(age) FROM employees);
```

7.2 CTE (Common Table Expression)

```
sql
CopyEdit
WITH dept_count AS (
    SELECT department, COUNT(*) AS total FROM employees GROUP BY department
)
SELECT * FROM dept_count WHERE total > 5;
```

---

8. Indexes & Performance Optimization

8.1 Creating an Index

```
sql
CopyEdit
CREATE INDEX idx_employee_name ON employees (name);
```

8.2 Using `EXPLAIN` for Optimization

```
sql
CopyEdit
EXPLAIN SELECT * FROM employees WHERE name = 'Alice';
```

---

9. Views & Stored Procedures

9.1 Creating a View

```
sql
CopyEdit
CREATE VIEW active_employees AS
SELECT * FROM employees WHERE status = 'active';
```

9.2 Stored Procedure

```
sql
CopyEdit
CREATE PROCEDURE GetEmployeeData()
BEGIN
    SELECT * FROM employees;
END;
```

---

10. Transactions & ACID Properties

ACID stands for:

- **A**tomicity → All or nothing
- **C**onsistency → Maintains integrity
- **I**solation → Transactions don't interfere
- **D**urability → Data is permanent after commit

```
sql
CopyEdit
BEGIN TRANSACTION;
UPDATE accounts SET balance = balance - 100 WHERE id = 1;
UPDATE accounts SET balance = balance + 100 WHERE id = 2;
COMMIT;
```

---

11. Advanced SQL Topics

- **Partitioning Tables** (`PARTITION BY`)
- **JSON Functions in SQL**
- **Full-Text Search**
- **Materialized Views**
- **Recursive Queries**
- **Window Functions (`RANK()`, `DENSE_RANK()`)**
- **PIVOT and UNPIVOT**

Conclusion

This covers **everything in SQL** from **basic queries** to **advanced database management**.