Day 2

1. What is SQL?

SQL (Structured Query Language) is used to:

- Create and manage databases
- Store, retrieve, update, and delete data
- Control access and maintain data integrity

SQL is **declarative** → you specify *what* you want, not *how* to get it.

---

2. SQL Language Categories

SQL commands are logically divided into **five categories**:

1. DDL – Data Definition Language
2. DML – Data Manipulation Language
3. DCL – Data Control Language
4. TCL – Transaction Control Language

---

3. DDL – Data Definition Language

Purpose

DDL commands define or modify the **structure of the database**.

They work at the **schema level**, not the data level.

Common DDL Commands

1. CREATE

Used to create:

- Databases

- Tables
- Views
- Indexes

Example – Create Database

```
CREATE DATABASE gl_da_cloud;
```

Example – Create Table

```
CREATE TABLE emp (
  emp_id INT PRIMARY KEY,
  first_name VARCHAR(50) NOT NULL,
  last_name VARCHAR(50),
  hire_date DATE,
  salary DECIMAL(10,2)
);
```

✔️ Defines structure

✔️ No data manipulation

---

2. ALTER

Used to **modify an existing table structure**.

You can:

- Add a column
- Drop a column
- Rename a column or table
- Change datatype (with caution)

Examples

```
ALTER TABLE emp ADD email VARCHAR(100);
ALTER TABLE emp DROP COLUMN last_name;
```

```
ALTER TABLE emp RENAME TO emp_123;
```

⚠️ ALTER can be expensive on large tables.

---

3. DROP

Used to **permanently delete database objects**.

```
DROP TABLE emp;
DROP DATABASE gl_da_cloud;
```

❌ Removes structure + data

❌ Cannot be rolled back

---

4. TRUNCATE

Used to remove **all records** from a table.

```
TRUNCATE TABLE emp;
```

✔️ Faster than DELETE

❌ No WHERE clause

❌ Cannot be rolled back

❌ Resets auto-increment counters

---

5. RENAME

Used to rename an object.

```
RENAME TABLE emp TO employee;
```

---

4. DML – Data Manipulation Language

Purpose

DML commands work on **data stored inside tables**.

DML Commands

1. INSERT

Adds new records.

```
INSERT INTO emp (emp_id, first_name, salary)
VALUES (1, 'Vaishnavi', 60000);
```

---

2. SELECT (DQL)

Used to retrieve data.

```
SELECT * FROM emp;
SELECT emp_id, first_name FROM emp;
```

This is the **most important SQL command**.

---

3. UPDATE

Modifies existing records.

```
UPDATE emp
SET salary = 65000
WHERE emp_id = 1;
```

⚠️ Without WHERE → updates **all rows**

---

4. DELETE

Removes records selectively.

```
DELETE FROM emp
WHERE emp_id = 1;
```

✔️ Can be rolled back

✔️ WHERE clause allowed

---

5. DELETE vs TRUNCATE vs DROP (CRITICAL)

Command	Deletes Data	Deletes Structure	WHERE	Rollback
DELETE	Yes	No	Yes	Yes
TRUNCATE	Yes (all)	No	No	No
DROP	Yes	Yes	No	No

---

6. DCL – Data Control Language

Purpose

Controls **user permissions and access**.

Commands

GRANT

```
GRANT SELECT ON emp TO analyst_user;
```

Gives read access only.

REVOKE

```
REVOKE INSERT, UPDATE ON emp FROM analyst_user;
```

Removes permissions.

---

7. TCL – Transaction Control Language

Purpose

Ensures **data consistency** during multiple operations.

A transaction = logical unit of work.

Commands

COMMIT

Permanently saves changes.

```
COMMIT;
```

ROLLBACK

Undo changes.

```
ROLLBACK;
```

SAVEPOINT

Rollback to a specific point.

```
SAVEPOINT sp1;
UPDATE emp SET salary = 80000 WHERE emp_id = 2;
ROLLBACK TO sp1;
```

⚠️ DDL statements auto-commit.

---

8. CRUD Operations

CRUD	SQL
Create	INSERT
Read	SELECT
Update	UPDATE
Delete	DELETE

---

9. Database Operations

Show Databases

```
SHOW DATABASES;
```

Use Database

```
USE gl_da_cloud;
```

---

10. Table Creation Techniques

10.1 Normal Table

Defines structure only.

10.2 Create Table from Another Table

```
CREATE TABLE high_paid_emp AS
SELECT emp_id, first_name, salary
FROM emp
WHERE salary > 60000;
```

✔️ Copies data + structure

❌ Does NOT copy constraints or indexes

10.3 Temporary Tables

```
CREATE TEMPORARY TABLE temp_high_paid_emp (
  emp_id INT,
  salary DECIMAL(10,2)
);
```

- Exists only for current session

- Automatically dropped

10.4 CTE (Common Table Expression)

Used for readable, reusable queries.

```
WITH high_salary_cte AS (
  SELECT * FROM emp WHERE salary > 70000
)
SELECT * FROM high_salary_cte;
```

✔️ Preferred in analytics

✔️ Interview favorite

---

11. ALTER TABLE – Detailed

```
ALTER TABLE emp ADD department VARCHAR(50);
ALTER TABLE emp MODIFY salary DECIMAL(12,2);
ALTER TABLE emp DROP COLUMN department;
```

---

12. Filtering Data – WHERE Clause

Used **before GROUP BY**.

```
SELECT * FROM emp
WHERE salary > 50000
AND hire_date >= '2022-01-01';
```

---

13. Sorting – ORDER BY

```
SELECT * FROM emp
ORDER BY salary DESC;
```

Default sorting = ASC

---

14. DESCRIBE Table Structure

```
DESC emp;
```

Output fields:

- Field
- Type
- Null
- Key
- Default
- Extra

---

15. Constraints (VERY IMPORTANT)

Constraints enforce **data integrity**.

Types of Constraints

PRIMARY KEY

- Uniquely identifies a row
- Cannot be NULL

```
emp_id INT PRIMARY KEY
```

UNIQUE

Ensures unique values.

```
email VARCHAR(100) UNIQUE
```

NOT NULL

Prevents NULL values.

DEFAULT

Assigns default value.

```
status VARCHAR(20) DEFAULT 'ACTIVE'
```

CHECK

Validates condition.

```
salary DECIMAL(10,2) CHECK (salary > 0)
```

FOREIGN KEY

Maintains relationship between tables.

```
FOREIGN KEY (emp_id) REFERENCES emp(emp_id)
```

Composite Primary Key

Multiple columns together act as primary key.

```
PRIMARY KEY (order_id, product_id)
```

⚠️ Separate PK + UNIQUE ≠ composite PK