Q1. What is a database? Differentiate between SQL and NoSQL databases.

Database:
A database is an organized collection of data that can be stored, managed, and retrieved efficiently.

Difference between SQL and NoSQL:

Feature	SQL Database	NoSQL Database
Data Model	Relational (tables with rows & columns)	Non-relational (documents, key-value, graph, column)
Schema	Fixed schema	Dynamic or flexible schema
Query Language	SQL (Structured Query Language)	Depends on DB type (e.g., MongoDB uses JSON-like queries)
Transactions	ACID properties supported	Eventual consistency, BASE model
Examples	MySQL, PostgreSQL, Oracle	MongoDB, Cassandra, Redis

Q2. What is DDL? Explain CREATE, DROP, ALTER, and TRUNCATE.

DDL (Data Definition Language):
Used to define and manage **database structures** like tables, schemas, and indexes.

1. CREATE

- Creates a new table or database.

```
CREATE TABLE Students (
    ID INT PRIMARY KEY,
    Name VARCHAR(50),
    Age INT
);
```

---

2. DROP

- Deletes a table or database permanently.

```
DROP TABLE Students;
```

---

3. ALTER

- Modifies the structure of an existing table.

```
ALTER TABLE Students ADD COLUMN Grade VARCHAR(10);
```

---

4. TRUNCATE

- Removes all rows from a table but **keeps the structure**.

```
TRUNCATE TABLE Students;
```

Q3. What is DML? Explain INSERT, UPDATE, and DELETE.

DML (Data Manipulation Language):
Used to **manipulate the data** stored in tables.

1. INSERT

- Adds new records.

```
INSERT INTO Students (ID, Name, Age) VALUES (1, 'Alice', 20);
```

---

2. UPDATE

- Modifies existing records.

```
UPDATE Students SET Age = 21 WHERE Name = 'Alice';
```

---

3. DELETE

- Deletes records from a table.

```
DELETE FROM Students WHERE Name = 'Alice';
```

Q4. What is DQL? Explain SELECT with an example.

DQL (Data Query Language):
Used to **query data** from database tables.

SELECT

- Retrieves data from one or more tables.

```
SELECT Name, Age FROM Students WHERE Age > 18;
```

Q5. Explain Primary Key and Foreign Key.

Primary Key (PK)

- Uniquely identifies each row in a table.
- Cannot be `NULL`.
- Example:

```
ID INT PRIMARY KEY
```

Foreign Key (FK)

- Links one table to another by referring to the primary key of the other table.
- Maintains **referential integrity**.

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    StudentID INT,
    FOREIGN KEY (StudentID) REFERENCES Students(ID)
);
```

Q6. Python code to connect MySQL to Python

```
import mysql.connector

# Connect to MySQL
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="password",
    database="school"
)

# Create a cursor object
cursor = conn.cursor()

# Execute SQL query
cursor.execute("SELECT * FROM Students")

# Fetch all results
rows = cursor.fetchall()
for row in rows:
    print(row)

# Close connection
conn.close()
```

Explanation:

- `cursor()`
    - Creates a cursor object used to **execute SQL queries** and fetch results.
- `execute()`
    - Executes a SQL statement (DDL, DML, DQL).

Q7. Order of execution of SQL clauses in a query

When executing a query, SQL follows this logical order:

1. **FROM** → Specifies the table(s)
2. **JOIN** → Combines tables if required
3. **WHERE** → Filters rows
4. **GROUP BY** → Groups rows
5. **HAVING** → Filters grouped rows
6. **SELECT** → Chooses columns
7. **DISTINCT** → Removes duplicates (applied after SELECT)
8. **ORDER BY** → Sorts the result
9. **LIMIT / OFFSET** → Limits the number of rows

Example:

```
SELECT Name, COUNT(*)
FROM Students
WHERE Age > 18
```

```
GROUP BY Name
HAVING COUNT(*) > 1
ORDER BY Name
LIMIT 5;
```