**Q1. What is a Database? Differentiate between SQL and NoSQL Databases.**

**Database:**  
A database is an organized collection of data that can be easily accessed, managed, and updated. It is used to store and retrieve information systematically.

**Differences between SQL and NoSQL Databases:**

| **Feature** | **SQL Databases** | **NoSQL Databases** |
| --- | --- | --- |
| **Data Model** | Relational (Tables with rows and columns) | Non-relational (Key-value, Document, Column, Graph) |
| **Schema** | Fixed schema, structured | Dynamic schema, flexible |
| **Scalability** | Vertically scalable | Horizontally scalable |
| **Query Language** | Structured Query Language (SQL) | No standardized query language |
| **Use Case** | Complex queries, ACID transactions | Big Data, real-time applications, scalability |
| **Examples** | MySQL, PostgreSQL, Oracle | MongoDB, Cassandra, Redis |

**Q2. What is DDL? Explain CREATE, DROP, ALTER, and TRUNCATE with an Example.**

**DDL (Data Definition Language):**  
DDL statements define and manage the structure of a database and its objects such as tables, indexes, and views.

* **CREATE**: Used to create a database object like a table.
* **DROP**: Deletes a database object and its data permanently.
* **ALTER**: Modifies the structure of an existing database object.
* **TRUNCATE**: Deletes all rows from a table without logging individual row deletions.

**Examples:**

-- CREATE

CREATE TABLE Employees (

ID INT PRIMARY KEY,

Name VARCHAR(50),

Age INT,

Department VARCHAR(50)

);

-- DROP

DROP TABLE Employees;

-- ALTER

ALTER TABLE Employees ADD Salary INT;

-- TRUNCATE

TRUNCATE TABLE Employees;

**Q3. What is DML? Explain INSERT, UPDATE, and DELETE with an Example.**

**DML (Data Manipulation Language):**  
DML statements are used to manipulate data within tables.

* **INSERT**: Adds new records to a table.
* **UPDATE**: Modifies existing records in a table.
* **DELETE**: Removes specific records from a table.

**Examples:**

-- INSERT

INSERT INTO Employees (ID, Name, Age, Department) VALUES (1, 'Alice', 30, 'HR');

-- UPDATE

UPDATE Employees SET Age = 31 WHERE Name = 'Alice';

-- DELETE

DELETE FROM Employees WHERE Name = 'Alice';

**Q4. What is DQL? Explain SELECT with an Example.**

**DQL (Data Query Language):**  
DQL is used to retrieve data from a database.

* **SELECT**: Fetches data from one or more tables.

**Example:**

-- Retrieve all data

SELECT \* FROM Employees;

-- Retrieve specific columns

SELECT Name, Age FROM Employees WHERE Department = 'HR';

**Q5. Explain Primary Key and Foreign Key.**

* **Primary Key:**  
  A column or combination of columns uniquely identifies each record in a table. It cannot contain NULL values.  
  Example:
* CREATE TABLE Students (
* StudentID INT PRIMARY KEY,
* Name VARCHAR(50),
* Age INT
* );
* **Foreign Key:**  
  A column in one table that establishes a relationship with a Primary Key in another table.  
  Example:
* CREATE TABLE Orders (
* OrderID INT PRIMARY KEY,
* StudentID INT,
* FOREIGN KEY (StudentID) REFERENCES Students(StudentID)
* );

**Q6. Write a Python Code to Connect MySQL to Python. Explain cursor() and execute() Method.**

**Python Code:**

import mysql.connector

# Establish connection

connection = mysql.connector.connect(

host='localhost',

user='root',

password='password',

database='testdb'

)

# Create a cursor object

cursor = connection.cursor()

# Execute an SQL query

cursor.execute("SELECT \* FROM Employees")

# Fetch and print results

for row in cursor.fetchall():

print(row)

# Close the connection

cursor.close()

connection.close()

**Explanation:**

* **cursor():** Creates a cursor object to interact with the database, allowing execution of SQL commands.
* **execute():** Executes an SQL query or command passed as a string.

**Q7. Give the Order of Execution of SQL Clauses in an SQL Query.**

1. **FROM**: Identifies the table(s) involved.
2. **JOIN**: Combines data from multiple tables.
3. **WHERE**: Filters rows based on conditions.
4. **GROUP BY**: Groups rows for aggregation.
5. **HAVING**: Filters grouped data.
6. **SELECT**: Chooses columns to retrieve.
7. **ORDER BY**: Sorts the result set.
8. **LIMIT/OFFSET**: Restricts the number of rows returned.

**Example:**

SELECT Department, COUNT(\*) AS EmployeeCount

FROM Employees

WHERE Age > 25

GROUP BY Department

HAVING COUNT(\*) > 2

ORDER BY EmployeeCount DESC

LIMIT 5;