In [ ]:
```
Q1. What is a database? Differentiate between SQL and NoSQL databases.
```

```
Database:

A database is a structured collection of data organized in a way that facilitates efficient
retrieval, storage, and management of data. It can store and retrieve data quickly and
accurately using appropriate data models.

SQL vs. NoSQL:

SQL (Structured Query Language):

Data Structure: SQL databases are table-based and follow a predefined schema.
Scalability: Vertically scalable (scaling up by increasing the power of an existing
machine).

Examples: MySQL, PostgreSQL, SQLite, Oracle.

NoSQL:

Data Structure: NoSQL databases are document-based, key-value pairs, graph databases, or
wide-column stores and don't require a fixed schema.
Scalability: Horizontally scalable (scaling out by adding more machines to the database).
Examples: MongoDB, CouchDB, Cassandra, Redis.
```

In [ ]:
```
Q2. What is DDL? Explain why CREATE, DROP, ALTER, and TRUNCATE are used with an example.
DDL (Data Definition Language):
DDL is a subset of SQL used for defining and managing the structure of a database.
```

In [ ]:
```
CREATE TABLE students (
    id INT PRIMARY KEY,
    name VARCHAR(50),
    age INT
);
```

In [ ]:
```
DROP TABLE students;
```

In [ ]:
```
ALTER TABLE students ADD COLUMN grade CHAR(1);
```

In [ ]:
```
TRUNCATE TABLE students;
```

In [ ]:
```
Q3. What is DML? Explain INSERT, UPDATE, and DELETE with an example.
```

In [ ]:
```
DML (Data Manipulation Language):
DML is a subset of SQL used for manipulating data stored in the database.
```

In [ ]:
```
INSERT INTO students (id, name, age, grade) VALUES (1, 'John Doe', 25, 'A');
```

In [ ]:
```
UPDATE students SET age = 26 WHERE id = 1;
```

In [ ]:
```
DELETE FROM students WHERE id = 1;
```

In [ ]:
```
Q4. What is DQL? Explain SELECT with an example.
```

In [ ]:
```
DQL (Data Query Language):
DQL is a subset of SQL used for querying the database to retrieve specific information
```

In [ ]:
```
SELECT * FROM students WHERE age > 20;
```

In [ ]:
```
Q5. Explain Primary Key and Foreign Key.
```

In [ ]:
```
Primary Key:

A primary key is a column or a set of columns that uniquely identifies each record in a table
It must contain unique values, and it cannot have NULL values.
```

In [ ]:
```
CREATE TABLE students (
    student_id INT PRIMARY KEY,
    name VARCHAR(50),
    age INT
);
```

In [ ]:
```
Foreign Key:

A foreign key is a column or a set of columns that refers to the primary key of another table
It establishes a link between two tables
```

In [ ]:
```
CREATE TABLE courses (
    course_id INT PRIMARY KEY,
    course_name VARCHAR(50),
    student_id INT,
    FOREIGN KEY (student_id) REFERENCES students(student_id)
);
```

In [ ]:
```
Q6. Write a python code to connect MySQL to python. Explain the cursor() and execute() method
```

In [ ]:
```
import mysql.connector

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

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

# Execute SQL query using execute method
cursor.execute("SELECT * FROM students")

# Fetch all the rows using fetchall method
rows = cursor.fetchall()

# Iterate through the rows
for row in rows:
    print(row)

# Close the cursor and connection
cursor.close()
conn.close()
```

In [ ]:
```
Q7. Give the order of execution of SQL clauses in an SQL query.
```

In [ ]:
```
The order of execution of clauses in an SQL query is as follows:

FROM: Specifies the table or tables from which data is to be retrieved.
WHERE: Filters the rows based on the specified conditions.
GROUP BY: Groups the result set by one or more columns.
HAVING: Filters the grouped rows based on the specified conditions.
SELECT: Specifies the columns to
```