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

Database: A database is a structured collection of data that is stored and organized for efficient retrieval and manipulation. Databases are designed to store and manage large volumes of data and provide mechanisms for querying, updating, and securing the data. They are essential for various applications, including business systems, websites, and more.

SQL (Structured Query Language) Database: SQL databases are relational databases that use structured query language (SQL) for defining and manipulating data. They are based on the relational model, with data organized into tables with predefined schemas. Key characteristics of SQL databases include data integrity, ACID (Atomicity, Consistency, Isolation, Durability) compliance, and the use of structured schemas. Examples of SQL databases include MySQL, PostgreSQL, and Oracle Database.

NoSQL Database: NoSQL databases are a category of databases that do not use traditional SQL for querying and manipulating data. They are designed to handle large volumes of unstructured or semi-structured data and offer more flexibility and scalability. NoSQL databases can be document-oriented, key-value stores, column-family stores, or graph databases. They are often used for big data and real-time applications. Examples of NoSQL databases include MongoDB, Cassandra, and Redis.

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 to define and manage the structure of a database, including creating, altering, and deleting database objects like tables, indexes, and views. It does not deal with the data within the database but focuses on defining its structure.

CREATE: Used to create database objects such as tables, indexes, and views. For example, to create a table:

In [ ]:
```
CREATE TABLE employees (
    id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50)
);
```

DROP: Used to delete existing database objects. For example, to drop a table:

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

ALTER: Used to modify the structure of an existing database object. For example, to add a new column to a table

In [ ]:
```
ALTER TABLE employees
ADD COLUMN email VARCHAR(100);
```

TRUNCATE: Used to remove all records from a table while keeping the table structure intact. For example, to truncate a table:

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

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

DML (Data Manipulation Language): DML is a subset of SQL used for manipulating data within a database, including inserting, updating, and deleting records.

INSERT: Used to add new records into a table. For example, to insert a new employee record:

In [ ]:
```
INSERT INTO employees (id, first_name, last_name, email)
VALUES (1, 'John', 'Doe', 'john.doe@email.com');
```

UPDATE: Used to modify existing records in a table. For example, to update an employee's email:

In [ ]:
```
UPDATE employees
SET email = 'updated.email@email.com'
WHERE id = 1;
```

DELETE: Used to remove records from a table based on a specified condition. For example, to delete an employee's record:

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

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

DQL (Data Query Language): DQL is a subset of SQL used for querying and retrieving data from a database.

SELECT: Used to retrieve data from one or more tables. For example, to retrieve all employee names and email addresses:

In [ ]:
```
SELECT first_name, last_name, email
FROM employees;
```

Q5. Explain Primary Key and Foreign Key.

Primary Key: A primary key is a column or a set of columns in a database table that uniquely identifies each record in the table. It enforces data integrity by ensuring that each row has a distinct and non-null primary key value. Primary keys are used as the basis for creating relationships between tables. For example, in an "employees" table, the "id" column could serve as the primary key.

Foreign Key: A foreign key is a column or a set of columns in one table that establishes a link to the primary key of another table. It enforces referential integrity by ensuring that values in the foreign key column match values in the primary key of the referenced table.

Foreign keys are used to create relationships between tables. For example, in an "orders" table, the "employee_id" column can be a foreign key that references the "id" column i

Q6. Write a Python code to connect MySQL to Python. Explain the cursor() and execute() method.

In [ ]:
```
import mysql.connector

# Establish a connection to the MySQL database
connection = mysql.connector.connect(
    host="your_host",
    user="your_user",
    password="your_password",
    database="your_database"
)

# Create a cursor object to interact with the database
cursor = connection.cursor()

# Execute an SQL query using the execute() method
query = "SELECT * FROM your_table"
cursor.execute(query)

# Fetch and print the results
for row in cursor.fetchall():
    print(row)

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

Explanation:

mysql.connector is a Python module for MySQL database connectivity. You establish a connection to the MySQL database using mysql.connector.connect() by providing the host, user, password, and database name. cursor() method creates a cursor object that allows you to interact with the database. execute() is used to execute SQL queries. In the example, a SELECT query retrieves data from a table. You can iterate through the results using cursor.fetchall() and print them. Finally, close the cursor and the database connection to release resources.

Q7. Give the order of execution of SQL clauses in an SQL query.

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

FROM: Specifies the source table(s) from which the data will be retrieved.
1. WHERE: Filters the rows from the source table(s) based on a specified conditi
2.GROUP BY: Groups the selected rows into sets based on the values of one or mor
3.HAVING: Filters the grouped rows based on a specified condition.
4.SELECT: Specifies the columns to be retrieved from the source table(s).
5.DISTINCT: Removes duplicate rows from the result set if used.
6.ORDER BY: Sorts the result set based on specified columns and sorting order.
7.LIMIT/OFFSET (if supported): Specifies the number of rows to be returned and t
```