

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

*SQL databases: Relational databases that store structured data in tables with predefined schemas. Follow the ACID (Atomicity, Consistency, Isolation, Durability) properties to ensure data integrity. Use SQL as a standard language for managing and querying the data. Best suited for applications with complex relationships and transactions, such as banking systems or enterprise applications. Examples: MySQL, Oracle, PostgreSQL, SQL Server.

*NoSQL databases: Non-relational databases that provide flexible schemas, allowing for dynamic and unstructured data. Do not follow the ACID properties strictly, favoring scalability and performance over strict data consistency. Use various data models like key-value pairs, documents, graphs, or wide-column stores. Best suited for handling large volumes of rapidly changing and unstructured data, such as social media feeds or IoT sensor data. Examples: MongoDB, Cassandra, Redis, Amazon DynamoDB.

Q2.What is DDL? Explain why CREATE, DROP, ALTER, and TRUNCATE are used with an example.

DDL (Data Definition Language) is a subset of SQL (Structured Query Language) that is used to define and manage the structure of a database. DDL statements are responsible for creating, altering, and dropping database objects such as tables, views, indexes, and constraints.

Here are explanations and examples of commonly used DDL statements:

1)CREATE: The CREATE statement is used to create a new database object, such as a table or view.

Example: `CREATE TABLE employees (id INT PRIMARY KEY, name VARCHAR(50), age INT`

2)DROP: The DROP statement is used to remove an existing database object, such as a table or view.

Example: `DROP TABLE employees;`

This example drops the "employees" table, removing it and all its associated data from the database.

3)ALTER: The ALTER statement is used to modify the structure of an existing database object, such as adding or dropping columns from a table.

Example: `ALTER TABLE employees ADD email VARCHAR(100);`

This example alters the "employees" table by adding a new column named "email" of type VARCHAR(100).

4)TRUNCATE: The TRUNCATE statement is used to remove all data from an existing table, but retains the table structure.

Example: `TRUNCATE TABLE employees;`

This example removes all data from the "employees" table, effectively emptying it, but keeps the table structure intact.

Q3.What is DDL? Explain why CREATE, DROP, ALTER, and TRUNCATE are used with an example.

DML (Data Manipulation Language) is a subset of SQL (Structured Query Language) that is used to manipulate and operate on the data within a database. DML statements are responsible for inserting, updating, and deleting data from tables

1)INSERT: The INSERT statement is used to add new data into a table.

Example:

```
INSERT INTO employees (id, name, age) VALUES (1, 'John', 30);
```

2)UPDATE: The UPDATE statement is used to modify existing data within a table.

Example:

```
UPDATE employees SET age = 35 WHERE id = 1;
```

3)DELETE: The DELETE statement is used to remove specific rows from a table.

example:

```
DELETE FROM employees WHERE id = 1;
```

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

DQL (Data Query Language) is a subset of SQL (Structured Query Language) that is used to retrieve and query data from a database. DQL statements, primarily the SELECT statement, are used to fetch data based on specific criteria.

Here is an explanation and an example of the SELECT statement:

SELECT: The SELECT statement is used to retrieve data from one or more tables in a database.

Example: SELECT * FROM employees;

This example retrieves all columns and rows from the "employees" table. The asterisk (*) represents all columns.

You can also specify specific columns to retrieve:

Example: SELECT id, name FROM employees;

This example retrieves only the "id" and "name" columns from the "employees" table.

You can use the SELECT statement with various clauses to filter, sort, and manipulate the retrieved data. Some commonly used clauses include:

WHERE: Used to specify conditions to filter the retrieved data based on specific criteria.

Example: `SELECT * FROM employees WHERE age > 30`; This example retrieves all columns and rows from the "employees" table where the "age" is greater than 30.

ORDER BY: Used to sort the retrieved data based on specified columns. Example: `SELECT * FROM employees ORDER BY name ASC`; This example retrieves all columns and rows from the "employees" table and sorts them in ascending order based on the "name" column.

LIMIT: Used to restrict the number of rows retrieved. Example: `SELECT * FROM employees LIMIT 10`; This example retrieves only the first 10 rows from the "employees" table.

In summary, the SELECT statement in DQL is used to retrieve and query data from a database. It allows you to specify the desired columns, apply conditions, sort the data, and limit the number of rows retrieved based on your requirements.

Q5. Explain Primary Key and Foreign Key.

*Primary Key: A primary key is a column or a set of columns in a table that uniquely identifies each row. It ensures the uniqueness and integrity of data within a table. A primary key constraint enforces the uniqueness and non-nullability of the key.

*Foreign Key: A foreign key establishes a relationship between two tables based on the values of a column(s). It refers to the primary key column(s) of another table, creating a link between the two tables. The foreign key constraint ensures the referential integrity of data across tables.

Q6. Write a python code to connect MySQL to python. Explain the cursor() and execute() method.`import mysql.connector`

Establish a connection to the MySQL database

```
conn = mysql.connector.connect( host="localhost", user="yourusername",  
password="yourpassword", database="yourdatabase" )
```

Create a cursor object to execute SQL queries

```
cursor = conn.cursor()
```

Execute a SQL query

```
cursor.execute("SELECT * FROM employees")
```

Fetch all the rows from the executed query

```
result = cursor.fetchall()
```

Print the retrieved data

```
for row in result: print(row)
```

Close the cursor and connection

```
cursor.close() conn.close()
```

A cursor object is created using the `cursor()` method on the connection object. The cursor allows executing SQL queries and fetching results.

The `execute()` method is used to execute SQL queries. In this example, a `SELECT` query is executed to fetch all rows from the "employees" table.

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

FROM: The FROM clause specifies the tables or views from which the data will be retrieved.

WHERE: The WHERE clause is used to filter the data based on specific conditions. It is executed after the FROM clause, and only the rows that satisfy the conditions are considered for further processing.

GROUP BY: The GROUP BY clause is used to group the rows based on specific columns. It is executed after the WHERE clause, and the data is grouped accordingly.

HAVING: The HAVING clause is used to filter the grouped data based on specific conditions. It is executed after the GROUP BY clause, and only the groups that satisfy the conditions are considered for further processing.

SELECT: The SELECT clause is used to specify the columns to retrieve from the tables or views. It is executed after the WHERE, GROUP BY, and HAVING clauses, and the specified columns are retrieved.

DISTINCT: The DISTINCT keyword eliminates duplicate rows from the retrieved data. It is executed after the SELECT clause, and duplicate rows are removed if

In []: