

16th Feb Assignment

February 27, 2023

1 Assignment 15

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

Ans. A database is an organized collection of data that can be accessed, managed, and updated easily. A database management system (DBMS) is software that helps users and applications interact with databases by providing tools for data storage, organization, retrieval, and security.

SQL (Structured Query Language) and NoSQL (Not only SQL) are two types of databases that differ in their data models, querying language, and storage mechanisms.

SQL databases are based on a relational data model and use SQL as the primary language for querying and managing data. SQL databases store data in tables with predefined columns and data types, and relationships between tables are defined through foreign keys. Examples of SQL databases include MySQL, Oracle, and PostgreSQL.

NoSQL databases, on the other hand, use various data models such as key-value, document, column-family, or graph to store and retrieve data. NoSQL databases typically use a non-SQL query language and can handle large volumes of unstructured or semi-structured data with high velocity. Examples of NoSQL databases include MongoDB, Cassandra, and Redis.

The main differences between SQL and NoSQL databases are:

- Data model: SQL databases use a relational data model based on tables with predefined columns and relationships between them, while NoSQL databases use various data models such as key-value, document, column-family, or graph.
- Query language: SQL databases use SQL as the primary language for querying and managing data, while NoSQL databases use non-SQL query languages specific to the data model used.

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

Ans. DDL stands for Data Definition Language. It is a subset of SQL that is used to define the database schema and its structure. The four main DDL commands are CREATE, DROP, ALTER, and TRUNCATE, and they are used for the following purposes:

- **CREATE:** The CREATE command is used to create a new table or a new database in SQL. For example, consider the following command that creates a new table named “employees” with three columns - “id”, “name”, and “salary”: `CREATE TABLE employees (id INT PRIMARY KEY, name VARCHAR(50), salary INT);`
- **DROP:** The DROP command is used to delete an existing database or table in SQL. For example, consider the following command that drops the “employees” table: `DROP TABLE employees;`
- **ALTER:** The ALTER command is used to modify the structure of an existing table in SQL. For example, consider the following command that adds a new column “age” to the “employees” table: `ALTER TABLE employees ADD age INT;`
- **TRUNCATE:** The TRUNCATE command is used to remove all the data from a table in SQL. For example, consider the following command that removes all the data from the “employees” table: `TRUNCATE TABLE employees;`

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

Ans.DML stands for Data Manipulation Language. It is used to manipulate data within a database. The three main commands in DML are INSERT, UPDATE, and DELETE.

- **INSERT:** This command is used to insert new data into a table. The basic syntax for INSERT command is: `INSERT INTO students (id, name, age, class) VALUES (101, 'John', 20, 'B.Tech');`
- **UPDATE:** This command is used to modify existing data in a table. The basic syntax for UPDATE command is: `UPDATE students SET age = 21 WHERE id = 101;`
- **DELETE:** This command is used to delete existing data from a table. The basic syntax for DELETE command is: `DELETE FROM students WHERE id = 101;`

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

Ans.DQL stands for Data Query Language, which is used to retrieve data from a database. The SELECT statement is the most common DQL command used to retrieve data from one or more tables

Suppose we have a table named students with the following columns: id, name, age, and score. We want to retrieve the names and scores of all the students who are above 18 years old. The SQL query for this would be:

`SELECT name, score FROM students WHERE age > 18;`

Q5. Explain Primary Key and Foreign Key.

Ans.Primary Key:

A primary key is a unique identifier for each record in a table. It is a column or a set of columns that uniquely identifies each row in the table. A primary key is used to ensure that each row in the table is unique and can be identified uniquely. In most

cases, a primary key is a single column, such as an ID number, but it can also be a combination of columns.

For example, consider a table named students that contains information about students. Each student has a unique ID number, which can be used as a primary key for the table. The primary key ensures that no two students in the table have the same ID number. In database management systems, a primary key and a foreign key are two important concepts that are used to establish relationships between tables.

Foreign Key:

A foreign key, on the other hand, is a column or a set of columns in one table that refers to the primary key of another table. It is used to establish relationships between tables. In other words, a foreign key is a way to link two tables together based on a common field.

For example, consider two tables: students and courses. The students table has a primary key column named student_id, while the courses table has a primary key column named course_id. To establish a relationship between the two tables, the courses table can include a foreign key column named student_id that references the student_id column in the students table. This foreign key ensures that each course in the courses table is associated with a specific student in the students table.

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

Ans.To connect MySQL to Python, we can use the mysql-connector-python module. Here's an example code that connects to a MySQL database, creates a table, inserts some data, and retrieves it using the cursor() and execute() methods:

```
[ ]: import mysql.connector
# Connect to the MySQL database
mydb = mysql.connector.connect(
    host="localhost",
    user="yourusername",
    password="yourpassword",
    database="mydatabase"
)
# Create a cursor object
mycursor = mydb.cursor()
# Create a table
mycursor.execute("CREATE TABLE customers (name VARCHAR(255), address_
↳ VARCHAR(255))")
# Insert some data
sql = "INSERT INTO customers (name, address) VALUES (%s, %s)"
val = ("John", "Highway 21")
mycursor.execute(sql, val)
```

```
mydb.commit()
print(mycursor.rowcount, "record inserted.")
# Retrieve the data
mycursor.execute("SELECT * FROM customers")
myresult = mycursor.fetchall()
for x in myresult:
    print(x)
```

we create a cursor object using the `cursor()` method. The cursor is used to execute SQL statements and retrieve data from the database.

We then create a table using the `execute()` method and passing the SQL statement as a string.

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

Ans.The order of execution of SQL clauses in an SQL query is as follows:

FROM, WHERE, GROUP BY, HAVING, SELECT, DISTINCT, ORDER BY, LIMIT/OFFSET