DATABASE MANAGEMENT SYSTEM – DAY 01

1) What is a database?

* A database is an organized collection of structured information, or data, which can be stored electronically in a computer system.

2) What is DBMS (Database Management System)?

* It is a collection of programs that enables user to create and maintain a database. In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications.

The database and DBMS software together is called as Database system.

A DBMS serves as an interface between the database and its end users or programs, allowing users to retrieve, update, and manage how the information is organized and optimized.

3) What are the examples for popular database management software?

- Oracle Database
- MySQL.
- Microsoft SQL Server.
- PostgreSQL.
- MongoDB.
- IBM DB2

4) What are the advantages of DBMS?

1. Redundancy is controlled.
2. Unauthorised access is restricted.
3. Providing multiple user interfaces.
    * Command-Line Interface (CLI)
    * Graphical User Interface (GUI)
    * Web Interface
    * Mobile Interface
4. Providing backup and recovery.

5) What are the disadvantage in File Processing System?

1. Data redundancy and inconsistency.

* In file processing system, the same data may be duplicated in several files. This is known as data redundancy. Redundancy causes higher storage.

* This situation can also result in data inconsistency. Inconsistency means that two files may contain different data of the same person or thing.

2. Difficult in accessing data.

3. Data isolation.

* The data in the file processing system is stored in various files. It becomes very difficult to write new application programs to retrieve the appropriate data.

Suppose that student emails are stored in the "students" file and fee information is stored in the "fee" file. The data from both files are required to send an email message to inform a student that the date for fee payment is over. In a file processing system, it is difficult to generate such type of list from multiple files.

4. Data integrity.

* Integrity means reliability and accuracy of data. The stored data must satisfy certain types of consistency constraints.
For example, Marks of a student should be a numeric value. It is very difficult to apply these constraints on files in the file processing system.

5. Concurrent access is not possible.

* Accessing the same data from the same file is called concurrent access. In the file system, concurrent access leads to incorrect data.

For example, a student wants to borrow a book from the library. He searches for the book in the library file and sees that only one copy is available. At the same time, another student also wants to borrow the same book and checks that one copy available. The first student option for borrow and gets the book.
But it is still not updated to zero-copy in the file and the second student also opt for borrow! But there are no books available. This is the problem of concurrent access to the file system.

6. Security Problems.

* File processing system does not provide sufficient security on data. In some situations, it is required to provide different types of access to data for different users. For example, a data entry operator should only be allowed to enter data. The chairman of the organization should be able to access or delete the data completely. Such types of security options are not available in file processing system.

6. What are the two types of DBMS?

* R – DBMS (Relational Database / Store data as tables - referred to as SQL)
* NR – DBMS (None - Relational Database / Store data as document - no SQL)

7. What is Structured Query Language (SQL)?

* SQL is a programming language used by nearly all relational databases to query, manipulate, and define data, and to provide access control.

8. What is client server architecture in DBMS?

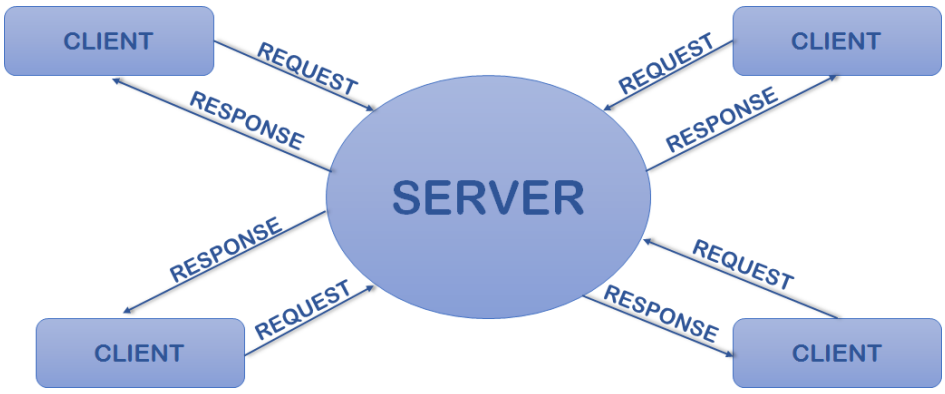
* Client-server architecture in DBMS means there are two parts: the client (user interface) and the server (data storage and processing). The client sends requests to the server over a network, and the server responds with the requested data or performs operations on the data.

The client is the part of the system that you interact with directly. It could be a desktop program, a web browser, or a mobile app. The client allows you to input commands, request data, and view information from the database.

The server, on the other hand, is responsible for storing and managing the actual database. It receives requests from the client, processes them, retrieves the necessary data, and sends it back to the client.

The client and server communicate with each other over a network. When you perform an action on the client, such as searching for information or updating data, the client sends a request to the server. The server then performs the requested operation and sends the results back to the client.

In summary, the client is the part you use to interact with the database, and the server is the part that stores and manages the data, fulfilling the client's requests.



9) What is “CRUD Operation” in DBMS?

"CRUD" is stands for Create, Read, Update, and Delete. It is a set of basic operations commonly used in database management systems (DBMS) to manipulate data within a database.

1. Create (C): This operation involves adding new data or records to the database. It could be creating a new entry with relevant information in a table.
2. Read (R): The read operation retrieves or reads existing data from the database. It involves querying the database to fetch specific records or retrieving all the records from a table.
3. Update (U): Updating allows modifying existing data in the database. It involves changing the values of specific fields or attributes within a record.
4. Delete (D): The delete operation removes data or records from the database. It involves permanently eliminating specific entries from a table.

These CRUD operations form the fundamental operations for interacting with a database. They provide the necessary functionality to manage and manipulate data effectively.

10) MySQL Data Types.

MySQL provides a variety of data types to store different types of data efficiently. Here are some commonly used data types in MySQL:

1. Numeric Data Types:
    - INT: Used to store whole numbers within a specified range.
    - FLOAT: Used to store single-precision floating-point numbers.
    - DOUBLE: Used to store double-precision floating-point numbers.
    - DECIMAL: Used to store exact numeric values with a specified precision and scale.
2. String Data Types:
    - VARCHAR: Used to store variable-length strings with a maximum length.
    - CHAR: Used to store fixed-length strings with a specified length.
    - TEXT: Used to store large strings of text.
3. Date and Time Data Types:
    - DATE: Used to store a date (year, month, and day).
    - TIME: Used to store a time (hour, minute, and second).
    - DATETIME: Used to store a date and time combination.
    - TIMESTAMP: Used to store a timestamp representing a specific point in time.

4. Boolean Data Type:

- BOOLEAN or BOOL: Used to store boolean values (true or false).

5. Binary Data Types:

- BINARY: Used to store fixed-length binary data.
- VARBINARY: Used to store variable-length binary data.
- BLOB: Used to store large binary objects.

6. Enumerated Data Type:

- ENUM: Used to store one value from a predefined set of values.

7. JSON Data Type:

- JSON: Used to store and manipulate JSON (JavaScript Object Notation) data.

These are just a few examples of the data types available in MySQL. Each data type has its own specific purpose and characteristics, and it's important to choose the appropriate data type based on the nature of the data you want to store.

11) Basic MySQL commands.

1. SHOW DATABASES;

2. CREATE DATABASE <database name>;
   or
   CREATE DATABASE IF NOT EXISTS <database name>;

3. USE <database name>;

4. DROP DATABASE <database name>;
   or
   DROP DATABASE IF EXISTS <database name>;

5. CREATE TABLE <table name> (
   <field name1> <DATA TYPE>,
   <field name2> <DATA TYPE>,
   <field name3> <DATA TYPE>,
   <field name4> <DATA TYPE>
   );

6. DESCRIBE <table name>;
   or
   DESC <table name>;

7. INSERT INTO <table name> VALUES (<data1>, <data2>, <data3>, <data4>);
   or

INSERT INTO <table name> (<field name1>, <field name2>, <field name3>)
VALUES (<data1>, <data2>, <data3>);

8. SELECT * FROM <table name>;

9. SHOW TABLES;

10. DROP TABLE <table name>;