

Tasks assigned today

19-02-2025

Suneetha.V

1. Database:

A **database** is an organized collection of data, generally stored and accessed electronically from a computer system. It is designed to handle large amounts of information by storing, retrieving, and managing data efficiently.

- **Example:** A database in a school could store data like student names, ages, grades, and addresses.

2. Why Do We Need a Database?

Databases are used to manage and organize data systematically, and they provide several advantages:

- **Efficient Data Storage:** Large amounts of data can be stored in an organized way.
- **Data Retrieval:** Data can be easily retrieved, updated, and managed.
- **Data Integrity:** Ensures the accuracy and consistency of data over time.
- **Data Security:** Provides mechanisms to secure sensitive data.
- **Concurrency:** Multiple users can access the database simultaneously without interfering with each other's operations.
- **Backup and Recovery:** Offers ways to back up data and recover it if needed.

Tasks assigned today

3. What is SQL?

SQL (Structured Query Language) is the standard programming language used to manage and manipulate relational databases. It allows you to perform tasks like querying data, inserting records, updating data, and deleting records.

- **Common SQL Commands:**

- **SELECT:** Retrieve data from the database.
- **INSERT:** Insert new records into the database.
- **UPDATE:** Modify existing records.
- **DELETE:** Remove records.
- **CREATE:** Create a new database or table.
- **DROP:** Delete a table or database.

4. What is MySQL?

MySQL is an open-source relational database management system (RDBMS) that uses SQL as its query language. MySQL is widely used for developing web applications and managing large amounts of data. It is part of the LAMP stack (Linux, Apache, MySQL, PHP/Perl/Python).

- MySQL is **fast, reliable, and easy to use**, making it one of the most popular databases for web applications.

Tasks assigned today

Key Differences Between SQL and MySQL:

Aspect	SQL	MySQL
Type	SQL is a language used to manage and query relational databases.	MySQL is a database management system (RDBMS) that uses SQL as its language.
Purpose	SQL is used to interact with databases through queries and commands.	MySQL is a software system that stores, organizes, and manages databases.
Functionality	SQL provides a standard syntax for querying and manipulating data.	MySQL is the software that processes SQL queries to manage databases.

Tasks assigned today

Role	SQL is a query language for working with data.	MySQL is a relational database management system that uses SQL.
Scope	SQL is used by many RDBMS systems (e.g., MySQL, PostgreSQL, Oracle, SQL Server).	MySQL is one specific RDBMS that supports SQL.
Examples of Use	Writing queries to interact with any database.	Creating, managing, and interacting with a MySQL database.

5. Difference Between Excel and a Database

Both Excel and databases are used for storing and managing data, but they have several key differences:

Feature	Excel	Database
Data Storage	Data is stored in tables/sheets.	Data is stored in tables with structured schemas.

Tasks assigned today

Size of Data	Suitable for small datasets.	Can handle large volumes of data.
Data Integrity	Limited to user's manual entry and checks.	Data integrity constraints (primary keys, foreign keys) ensure accuracy.
Data Retrieval	Searching and filtering are less efficient.	Fast data retrieval through SQL queries.
Concurrency	Typically used by one user at a time.	Multiple users can access and modify data simultaneously.
Security	Limited security features.	Provides role-based access and strong security features.

6. What is DBMS?

DBMS (Database Management System) is a software system that provides an interface for users and applications to interact with a database. It manages data storage, retrieval, and manipulation.

Key Functions of DBMS:

- **Data Definition:** Defines the structure of the database, including tables and relationships.
- **Data Manipulation:** Allows users to add, modify, and delete data.

Tasks assigned today

- **Data Security:** Ensures that only authorized users can access the database.
- **Data Integrity:** Maintains data accuracy and consistency.
- **Backup and Recovery:** Provides mechanisms for data backup and recovery.

7. Types of DBMS

There are several types of DBMS based on how they store data:

- **Hierarchical DBMS:** Data is stored in a tree-like structure, with parent-child relationships (e.g., IBM's IMS).
- **Network DBMS:** Data is stored in a graph structure where entities can have multiple relationships (e.g., Integrated Data Store).
- **Relational DBMS (RDBMS):** Data is stored in tables with rows and columns, and relationships are maintained using foreign keys. SQL is used to manage the data. (e.g., MySQL, Oracle, PostgreSQL)
- **Object-Oriented DBMS (OODBMS):** Data is represented as objects, similar to object-oriented programming concepts (e.g., db4o, ObjectDB).

8. What is RDBMS?

RDBMS (Relational Database Management System) is a type of DBMS that stores data in a structured format using rows and columns. Data in an RDBMS is organized into tables, and relationships between data are defined using **foreign keys**. SQL is the standard language for managing RDBMS databases.

- Examples of RDBMS include **MySQL**, **PostgreSQL**, **Oracle**, **Microsoft SQL Server**.

Tasks assigned today

9. What is Non-RDBMS?

Non-RDBMS refers to database management systems that don't follow the relational model. These databases are more flexible and are suitable for unstructured or semi-structured data.

- **Examples:**
 - **NoSQL Databases** (e.g., MongoDB, Cassandra, CouchDB) for large, unstructured data.
 - **Key-Value Stores** (e.g., Redis, DynamoDB).
 - **Document Stores** (e.g., MongoDB, Couchbase).

10. What is Frontend?

The **frontend** refers to the **client-side** part of an application or system, which interacts directly with the user. In the context of web development, the frontend includes everything that the user experiences: the design, structure, content, and behavior of a webpage.

- **Frontend Technologies:**
 - HTML, CSS, and JavaScript.
 - Frameworks like React, Angular, Vue.js.

11. What is Backend?

The **backend** refers to the **server-side** part of an application. It is responsible for managing and processing data, handling requests from the frontend, interacting with databases, and performing computations.

- **Backend Technologies:**

Tasks assigned today

- Programming languages: Python, Java, Node.js, Ruby, etc.
- Databases: MySQL, MongoDB, PostgreSQL, etc.
- Frameworks: Django, Flask, Spring, Express, etc.

12. What is a Database Environment?

A **database environment** refers to the set of hardware, software, and procedures that support the creation, management, and interaction with databases. This includes:

- **Hardware:** Servers and storage systems.
- **Software:** DBMS software (like MySQL, PostgreSQL), application programs, and middleware.
- **Procedures:** Guidelines and rules for managing data, security, and access control.

13. What is a Client?

A **client** is a computer or application that requests services or data from a server. In a database context, the client could be a **user application** or an **SQL client tool** that sends queries to a database server.

- Examples of database clients: **phpMyAdmin**, **SQL Server Management Studio**, **pgAdmin**.

14. What is a Server?

A **server** is a system that provides services, such as hosting a database or providing data to client applications. In the context of databases, the server stores, manages, and serves data to clients.

Tasks assigned today

- **Database Server:** A server running a DBMS (e.g., MySQL Server, PostgreSQL Server).

15. What is CLI?

CLI (Command Line Interface) is a way to interact with a system using text-based commands. Instead of using a graphical interface, you type commands in a terminal or console.

- Examples: **MySQL Command Line Client, PostgreSQL CLI, Unix/Linux Terminal.**

16. What is GUI?

GUI (Graphical User Interface) is a user interface that allows users to interact with software through graphical icons and visual indicators, rather than text-based commands.

- Examples of Database GUI Tools: **MySQL Workbench, phpMyAdmin, pgAdmin.**