**Acropolis Institute Of Technology And Research,**

**Indore (M.P.)**

****

**Subject – Database Management System (DBMS)**

**(CY-405)**

**Name - Sumit Kumar Chaturvedi**

**Enrollment No. - 0827CY221062**

**Branch - CS(Cyber Security)**

**Semester - 4th  (2nd year)**

**Submitted To – Mrs. Nidhi Nigam Ma’am**

**Index:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S No.** | **Date** | **Title** | **Remark** | **Teacher**  **Sign** |
| **1.** | **28/03/23** | **To study DBMS and RDBMS, its characteristic comparisons and study of popular DB software** |  |  |
| **2.** | **01/04/23** | **Study of MySQL, Features of MySQL, Installation steps** |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Lab Work – 01**

**Introduction To DBMS :-**

A **Database Management System (DBMS)** is specialized software designed to efficiently organize, store, and retrieve data in a structured format. Here are some key points about DBMS:

1. **Purpose**: DBMS serves as an intermediary between applications and the underlying database. It enables users to interact with data without worrying about low-level details.
2. **Functions**:
   * **Data Storage**: DBMS stores data in an organized manner, ensuring efficient access and retrieval.
   * **Data Retrieval**: Users can query the database to retrieve specific information.
   * **Data Modification**: DBMS allows adding, updating, or deleting data.
   * **Security**: It provides access control and ensures data integrity.
   * **Concurrency Control**: Manages simultaneous access by multiple users.
   * **Backup and Recovery**: DBMS handles data backups and restores.
3. **Popular DBMSs**:
   * **MySQL**: An open-source relational database management system.
   * **Oracle**: Widely used in enterprise applications.
   * **MongoDB**: A NoSQL database for handling unstructured data.
   * And many more!
   * **Concepts**:
   * **Entity-Relationship (ER) Model**: Describes entities, their attributes, and relationships.
   * **Relational Model**: Represents data as tables (relations) with rows and columns.
   * **Normalization**: Process to minimize redundancy and improve data integrity.
   * **SQL (Structured Query Language)**: Used for querying and managing databases.

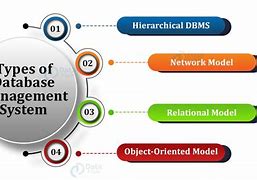
**Key Features :-**

1. **Minimum Duplication and Redundancy**:
   * In a DBMS, data files are shared, minimizing data duplication and redundancy.
   * Information occurs only once, reducing the chances of duplicity.
2. **Cost-Effective Storage**:
   * Proper integration of data saves storage space and costs.
   * Efficient data management reduces the money spent on storing and maintaining data.
3. **User-Friendly Query Language**:
   * Non-technical users can work with DBMS.
   * The query language provided by DBMS is easy to understand for tasks like updating, inserting, deleting, and searching records.
4. **Large Database Maintenance**:
   * DBMS can handle large databases with features like security, backup, and recovery.
   * Big companies rely on DBMS to maintain extensive data repositories.
5. **High-Level Security**:
   * Only authorized users (such as DBAs or department heads) have full access to the database.
   * Security restrictions are enforced based on user roles and responsibilities.
6. **Permanent Data Storage**:
   * DBMS stores data files permanently, ensuring no loss of data.
   * Backup and recovery methods provide additional safeguards.
7. **Multi-User Access**:
   * Multiple users can access data stored in a single data store.
   * Collaboration and concurrent access are supported.

**Applications Of DBMS :-**

1. **Railway Reservation System**:
   * DBMS is crucial for storing ticket booking records, train arrival status, and flight information.
   * It keeps travelers informed about delays through database updates.
2. **Library Management System**:
   * Managing numerous books in a library requires an efficient system.
   * DBMS maintains information related to book names, issue dates, availability, and authors.
3. **Banking Database Management**:
   * Banks store customer transaction data in databases.
   * DBMS ensures secure and organized management of financial records.
4. **Education Sector**:
   * Online exams conducted by schools and universities rely on DBMS.
   * Student registrations, grades, courses, fees, attendance, and results are stored in databases.
5. **Credit Card Transactions**:
   * DBMS handles credit card purchases and generates monthly statements.
6. **Social Media Sites**:
   * Platforms like Facebook, Twitter, and Pinterest store user information using DBMS.
   * It enables interaction and connection among users.
7. **Telecommunications**:
   * Call details and monthly postpaid bills are stored in databases using DBMS.
8. **Accounting and Finance**:
   * DBMS manages information about sales, stock holdings, and financial instruments like stocks and bonds.
9. **E-Commerce Websites**:
   * Online shopping platforms (e.g., Amazon, Flipkart) rely on DBMS for product management, invoicing, and payments.
10. **Human Resource Management**:
    * Large organizations store employee details, salaries, and taxes using DBMS.

**Types Of DBMS :-**



1. Relational DBMS (RDBMS)
2. Object-oriented DBMS (OODBMS)
3. Hierarchical DBMS (HDBMS)
4. Network DBMS (NDBMS)
5. NoSQL DBMS
   * Document-oriented DBMS
   * Column-family DBMS
   * Key-value databases
   * Graph DBMS
6. In-memory DBMS

**Object-oriented DBMS (OODBMS)**

An Object-oriented Database Management System (OODBMS) is a type of DBMS that organizes data into objects and allows for the creation of classes and inheritance. In an OODBMS, data is stored in a format that is similar to objects in object-oriented programming languages, such as Java or C++. Each object has its own properties, methods, and behaviors, and can be part of a class or hierarchy of classes.

**Features of OODBMS**

One of the key features of an OODBMS is the ability to model complex relationships and hierarchies within the data. For example, an OODBMS can model an object such as a car, which has properties such as make, model, and year, and methods such as start and stop. It can also model relationships between objects, such as a car having a one-to-many relationship with its parts.

**Examples of OODBMS**

OODBMSs are well suited for handling complex, unstructured, or semi-structured data, and are often used in applications such as engineering, geographic information systems, and multimedia. Some popular examples of OODBMSs include MongoDB, Apache Cassandra, and ObjectDB. They are not as popular as RDBMSs but still have their use cases.

However, OODBMSs are not as standardized as RDBMSs, and the query languages and interfaces used to interact with them can vary widely between different systems. Additionally, OODBMSs generally have less support for ad-hoc queries, reporting, and analysis than RDBMSs.

**Hierarchical DBMS (HDBMS)**

A Hierarchical Database Management System (HDBMS) is a type of DBMS that organizes data in a hierarchical tree-like structure. In an HDBMS, data is represented as a series of records, with each record having one parent record and one or more child records. This creates a parent-child relationship between records, with the parent record being at the top of the hierarchy and child records being at the bottom.

**Features of HDBMS**

The hierarchical model is closely related to the tree data structure, with the parent-child relationship being similar to the parent-child relationship between nodes in a tree. The data is stored in a tree-like structure, with each node representing a record and each branch representing a relationship between records. This structure allows for easy traversal of data, but can make it more difficult to represent more complex relationships between records.

**Examples of HDBMS**

HDBMSs were popular in the past, especially for applications such as data modeling for manufacturing systems and other similar applications. IBM’s Information Management System (IMS) is an example of a HDBMS. However, they have largely been replaced by more modern DBMSs, such as RDBMSs and OODBMSs, which provide more flexibility and scalability.

HDBMSs are not as popular as RDBMSs or OODBMSs and are not used as much as they used to be. They are not suitable for large and complex data sets, and are not very good at handling many-to-many relationships between records. They are best suited for highly structured data and applications that require a high level of data integrity and security.

**Network DBMS (NDBMS)**

A Network Database Management System (NDBMS) is a type of DBMS that organizes data in a network structure. In a NDBMS, data is represented as a series of records, with each record having multiple parents and children. This creates a many-to-many relationship between records, with records being connected to multiple other records, creating a web-like structure.

The network model is closely related to the graph data structure, with the many-to-many relationship being similar to the edges between nodes in a graph. The data is stored in a web-like structure, with each node representing a record and each edge representing a relationship between records. This structure allows for easy traversal of data, and can represent complex relationships between records.

NDBMSs were popular in the past and were used for applications such as data modeling for manufacturing systems, inventory management, and other similar applications. Integrated Data Store (IDS) and Integrated Data Store II (IDS II) developed by Integrated Data Systems (IDS) are examples of NDBMS. However, they have largely been replaced by more modern DBMSs, such as RDBMSs and OODBMSs, which provide more flexibility and scalability.

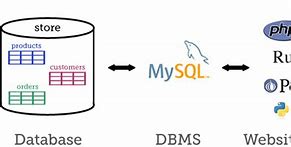
NDBMSs are not as popular as RDBMSs or OODBMSs and are not used as much as they used to be. They are best suited for applications that require a high level of data integrity and security, and are not suitable for large and complex data sets.

**NoSQL DBMS**

NoSQL (Not Only SQL) Database Management Systems (DBMS) refer to a type of DBMS that do not use the traditional relational model and SQL (Structured Query Language) for data storage and retrieval. Instead, they use alternative data models and query languages that are more suitable for handling large and unstructured data sets, as well as for distributed and cloud-based environments.

NoSQL DBMSs are designed to handle big data and high-scale data processing workloads, they can handle large volume, high velocity, and varied data. They are often used in web, mobile, gaming, and social media applications, as well as in the Internet of Things (IoT), real-time analytics, and machine learning applications.

**What Is My SQL :-**



MySQL is the world’s most popular open source database. According to [**DB-Engines**](https://db-engines.com/en/ranking)**,** MySQL ranks as the second-most-popular database, behind [**Oracle Database**](https://www.oracle.com/database/)**.** MySQL powers many of the most accessed applications, including Facebook, Twitter, Netflix, Uber, Airbnb, Shopify, and Booking.com.

Since MySQL is open source, it includes numerous features developed in close cooperation with users over more than 25 years. So it’s very likely that your favorite application or programming language is supported by MySQL Database.

### How do you pronounce “MySQL”?

“My ess-cue-el” is the “official” way to pronounce “MySQL,” but pronouncing it “my sequel” is common too.

### What is the name of the MySQL dolphin?

The MySQL logo is a dolphin named Sakila. The name was chosen from a large list suggested by users during the “Name the Dolphin” contest. The winning name was submitted by Ambrose Twebaze, an open source software developer from Eswatini (formerly Swaziland), Africa.

## MySQL is a relational database management system

[**Databases**](https://www.oracle.com/database/what-is-database/) are the essential data repository for all software applications. For example, whenever someone conducts a web search, logs in to an account, or completes a transaction, a database system is storing the information so it can be accessed in the future.

A [**relational database**](https://www.oracle.com/database/what-is-a-relational-database/) stores data in separate tables rather than putting all the data in one big storeroom. The database structure is organized into physical files optimized for speed. The logical data model, with objects such as data tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one to one, one to many, unique, required, or optional, and “pointers” between different tables. The database enforces these rules so that with a well-designed database your application never sees data that’s inconsistent, duplicated, orphaned, out of date, or missing.

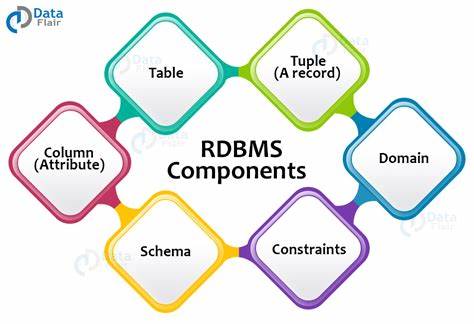
The “SQL” part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

## MySQL is open source

[**Open source**](https://developer.oracle.com/open-source/what-is-open-source/) means it’s possible for anyone to use and modify the software. Anybody can download MySQL software from the internet and use it without paying for it. You can also change its source code to suit your needs. MySQL software uses the[**GNU General Public License**](http://www.fsf.org/licenses/) (GPL) to define what you may and may not do with the software in different situations.

If you feel uncomfortable with the GNU GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from Oracle. See the **[MySQL Licensing Information section](https://www.mysql.com/about/legal/)** for more information.

**About RDBMS :-**



**A relational**[**database**](https://www.techtarget.com/searchdatamanagement/definition/database)**management system (RDBMS) is a collection of programs and capabilities that enable IT teams and others to create, update, administer and otherwise interact with a**[**relational database**](https://www.techtarget.com/searchdatamanagement/definition/relational-database)**. RDBMSes store data in the form of tables, with most commercial relational database management systems using**[**Structured Query Language**](https://searchsqlserver.techtarget.com/definition/SQL)**(SQL) to access the database. However, since SQL was invented after the initial development of the relational model, it is not necessary for RDBMS use.**

**The RDBMS is the most popular database system among organizations across the world. It provides a dependable method of storing and retrieving large amounts of data while offering a combination of system performance and ease of implementation.**

**RDBMS vs. DBMS**

**In general, databases store sets of data that can be queried for use in other applications. A database management system supports the development, administration and use of database platforms.**

**An RDBMS is a type of**[**database management system**](https://searchsqlserver.techtarget.com/definition/database-management-system)**(DBMS) that stores data in a row-based table structure which connects related data elements. An RDBMS includes functions that maintain the security, accuracy, integrity and consistency of the data. This is different than the file storage used in a DBMS.**

**Other differences between database management systems and relational database management systems include:**

* **Number of allowed users. While a DBMS can only accept one user at a time, an RDBMS can operate with multiple users.**
* **Hardware and software requirements. A DBMS needs less software and hardware than an RDBMS.**
* **Amount of data. RDBMSes can handle any amount of data, from small to large, while a DBMS can only manage small amounts.**
* **Database structure. In a DBMS, data is kept in a hierarchical form, whereas an RDBMS utilizes a table where the headers are used as column names and the rows contain the corresponding values.**
* **ACID implementation. DBMSes do not use the atomicity, consistency, isolation and durability (**[**ACID**](https://searchsqlserver.techtarget.com/definition/ACID)**) model for storing data. On the other hand, RDBMSes base the structure of their data on the ACID model to ensure consistency.**
* **Distributed databases. While an RDBMS offers complete support for**[**distributed databases**](https://www.techtarget.com/searchoracle/definition/distributed-database)**, a DBMS will not provide support.**
* **Types of programs managed. While an RDBMS helps manage the relationships between its incorporated tables of data, a DBMS focuses on maintaining databases that are present within the computer network and system**[**hard disks**](https://www.techtarget.com/searchstorage/definition/hard-disk)**.**
* **Support of database normalization. An RDBMS can be**[**normalized**](https://searchsqlserver.techtarget.com/definition/normalization)**, but a DBMS cannot.**

**Lab Work -02**

**DDL Commands: Data definition language (DDL)** refers to the set of SQL commands that can create and manipulate the structures of a database. DDL statements are used to create, change, and remove objects including indexes, triggers, tables, and views

**Database Select/Switch Command:**

* **Use command : this command is used to select the database or switch between the databases**
* **Syntax: use table\_name;**

**C:\Users\csit\Pictures\Screenshots\Screenshot 2024-04-01 124548.png**

**Create Table Command:**

**This command is used to create the table in the selected database.**

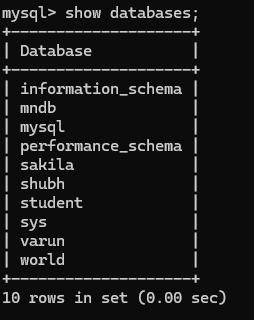
**Syntax: create table table\_name;**

**C:\Users\csit\Pictures\Screenshots\Screenshot 2024-04-01 124611.png**

**Show Table and values in Table:**

**This command is used to show all the created databases in the system.**

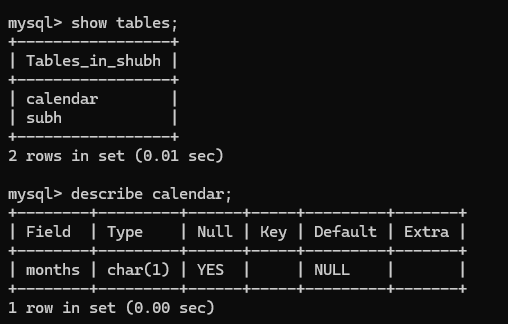
**Syntax: show databases;**

****

**Using Show Command:**

**This command is used to show all the created tables in the selected database.**

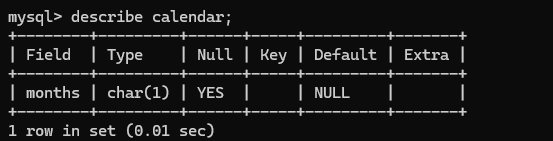
**Syntax: show tables;**

****

**Desc Command :**

**This command is used to describe all the attributes in the table with the values.**

**Syntax: desc table\_name;**

****