TASK DAY -1

1. Installation of MySQL Workbench and setup a local instance connection for hands on.

2. Create a word documentation of Introduction to DBMS, RDMS

    - Include all points that we have discussed so far.

    - Include additional important points from your research related to DBMS

    - Include Advantages and disadvantages of DBMS and RDMS with examples

    - Include different other types of normalization techniques.

Database Management System (DBMS)

- Database Management System (DBMS) is a software that is used to manage the data. Some of the popular DBMS software are MySQL, IBM Db2, Oracle, PostgreSQL etc.

- DBMS provides many operations e.g. creating a database, storing in the database, updating an existing database, delete from the database. DBMS is a system that enables you to store, modify and retrieve data in an organized way. It also provides security to the database.

Why DBMS?

DBMS is used for small organization and deal with small data. The system offers many benefits over the traditional file system, including the following:

- It helps maintain data uniformity.
- Handles large sets of data efficiently.
- Versatile
- Faster way of managing data

Types of DBMS

1. **Relational DBMS (RDBMS):** An RDBMS stores data in tables with rows and columns and uses SQL (Structured Query Language) to manipulate the data.

2. **Object-Oriented DBMS (OODBMS):** An OODBMS stores data as objects, which can be manipulated using object-oriented programming languages.

3. **NoSQL DBMS:** A NoSQL DBMS stores data in non-relational data structures, such as key-value pairs, document-based models, or graph models.

Applications of DBMS

- Telecom: DBMS is crucial for managing information related to calls, network usage, and customer details in the telecom industry. Without DBMS, handling the continuous flow of data would be impractical.

- Industry: Whether it's a manufacturing unit, warehouse, or distribution centre, all require a database to track inventory, shipments, and product movements.

- Banking System: DBMS is essential for storing customer information, tracking transactions, and ensuring the security of sensitive data in banking systems.

- Sales: DBMS helps in storing customer information, managing inventory, and generating reports for sales analysis.

- Airlines: DBMS is used to manage flight reservations, schedules, and real-time updates to ensure accurate and efficient allocation of seats.

  Education Sector: Schools and colleges use DBMS to store and retrieve data related to student records, staff details, course information, and attendance.

- Online Shopping: E-commerce platforms like Amazon and Flipkart rely on DBMS to manage product catalogues, user preferences, and secure transactions, ensuring a smooth shopping experience for users.

Pros of DBMS:

- **Multiple User Interface:** DBMS provides users with different types of UI like graphical UI, application program interfaces (API), etc.
  Example: An employee uses a graphical UI to access and update customer records in a retail DBMS.

- **Controls Database Redundancy:** DBMS can control data redundancy by storing all information in a single database file.
  Example: In a university database, student information like name, address, and course enrolment is stored in a single table, reducing redundancy.

- **Reduce Time:** DBMS reduces the time of development and the need for maintenance.
  Example: A software company uses a DBMS to quickly develop and deploy a new customer relationship management (CRM) system, reducing development time.

- **Data Sharing:** Authorized users in an organization can easily share data and info among multiple other users in DBMS.
  Example: A marketing team accesses customer data stored in a DBMS to personalize marketing campaigns.

- **Easily Maintenance:** DBMS can be easily maintained because of the database system's centralized nature.
  Example: An IT administrator performs routine maintenance tasks like backups and updates on a centralized DBMS server.

- **Backup:** DBMS provides users with subsystems of backup and recovery that create an automatic backup of data from software and hardware failures.
  Example: A financial institution uses DBMS backup features to recover customer transaction data in case of a hardware failure.

Cons of DBMS:

- **Cost of Hardware and Software:** A high-speed data processor and large memory size are required to run any DBMS software.
  Example: A company invests in high-end server hardware and DBMS software licenses to support its enterprise-level database.

- **Size:** DBMS occupies large disk space and memory to run efficiently.
  Example: A database for a multinational corporation occupies several terabytes of disk space to store vast amounts of transaction data.

- **Complexity:** Additional complexity and requirements are created by the database systems.
  Example: DBMS configuration and administration require specialized knowledge and training, adding complexity to IT operations.

- **Higher Impact of Failure:** Failure impacts the database significantly because all data is usually stored in a single database.
  Example: A retail business faces severe disruptions in operations if its centralized DBMS server experiences a critical failure, leading to loss of sales data and customer information.

Relational Database Management System (RDBMS)

- RDBMS or Relational Database Management System is a database system software that manages and maintains data in a tabular format.

- It is the software that operates on a relational schema (database arranged in tables with rows and columns).

- In a relational database, each row in the table is a record with a unique ID called the key. The columns of the table hold attributes of the data, and each record usually has a value for each attribute, making it easy to establish the relationships among data points.

- A relational database management system (RDBMS) is a program used to create, update, and manage relational databases. Some of the most well-known RDBMSs include MySQL, PostgreSQL, MariaDB, Microsoft SQL Server, and Oracle Database.

Why RDBMS?

An RDBMS offers businesses a systematic view of data, which can be used to enhance different aspects of decision-making. Relational databases offer several other benefits as well, including:

- Allow multiple user access.
- Store large packs of data.
- Minimum Data Redundancy
- Maintains Data Integration
- Better Tools for Structuring and Organizing Data

Primary key:

The primary key finds out the similarity in the relationship. For the entire table, there is only one primary key. Every table has got a particular primary key that cannot be shared by other tables.

Foreign key:

The foreign key is a key used for a different table of data which is referred by the primary key. There are many foreign keys for a single table. It depends on the primary key and its decision to refer those foreign keys to the table. Every foreign key can be shared, and it speaks about the coordination among the data of different tables.

Properties of Relational Database

There are following four commonly known properties of a relational model known as ACID properties, where:

1. A means Atomicity: This ensures the data operation will complete either with success or with failure. It follows the 'all or nothing' strategy. For example, a transaction will either be committed or will abort.

2. C means Consistency: If we perform any operation over the data, its value before and after the operation should be preserved. For example, the account balance before and after the transaction should be correct, i.e., it should remain conserved.

3. I mean Isolation: There can be concurrent users for accessing data at the same time from the database. Thus, isolation between the data should remain isolated. For example, when multiple transactions occur at the same time, one transaction effects should not be visible to the other transactions in the database.

4. D means Durability: It ensures that once it completes the operation and commits the data, data changes should remain permanent.

Applications of RDBMS:

- Business Management: They help companies organize customer data, manage supply chains, and track inventory.

- Web Services: RDBMS form the backbone of websites and apps, storing user profiles, posts, and comments.

- Data Analysis: They're crucial for analysing large datasets, providing insights for businesses and scientific research.

- Scientific Research: Scientists use RDBMS to manage experimental data and simulations, aiding research in fields like genomics and climate modelling.

- Government Services: Governments use RDBMS to manage citizen records, tax data, and public service information.

Pros of RDBMS:

- **Data Integrity**: RDBMS enforces relationships between data, ensuring referential integrity.
  Example: In a university database, the relationship between student records and course enrolment ensures that only valid student IDs are associated with enrolled courses.

- **Query Optimization:** RDBMS optimizes queries for efficient data retrieval and manipulation.

Example: A retail business executes complex SQL queries to analyse sales data and identify trends, helping in strategic decision-making.

- **Scalability:** RDBMS systems can scale to accommodate growing volumes of data and users.
  Example: An e-commerce platform expands its database infrastructure to handle increased website traffic during holiday seasons without sacrificing performance.

- **Standardization**: SQL provides a standardized language for interacting with RDBMS systems.
  Example: An IT professional proficient in SQL can seamlessly transition between different RDBMS platforms like MySQL, PostgreSQL, and Oracle Database.

- **ACID Compliance:** RDBMS systems ensure ACID (Atomicity, Consistency, Isolation, Durability) properties for transactions.
  Example: A banking system guarantees that a funds transfer transaction is atomic, consistent, isolated from other transactions, and durable even in the event of a system failure.

Cons of RDBMS:

- **Normalization Overhead:** Normalization can introduce complexity and overhead for database design and maintenance.
  Example: In a healthcare database, normalization may require splitting patient information across multiple tables, increasing the complexity of queries to retrieve patient records.

- **Performance Limitations:** Complex queries or large datasets can impact performance in RDBMS systems.
  Example: A data analytics platform experiences slowdowns when processing large volumes of transactional data stored in an RDBMS, requiring optimization of SQL queries.

- **Vendor Lock-in:** Switching between RDBMS vendors can be challenging due to differences in SQL dialects and features.
  Example: A company faces migration challenges when transitioning from Oracle Database to PostgreSQL due to differences in stored procedures and data types.

- **Scalability Challenges:** Scaling RDBMS systems horizontally can be difficult compared to NoSQL databases.
  Example: A social media platform encounters difficulties in horizontally scaling its RDBMS to handle the exponential growth of user-generated content, prompting consideration of NoSQL alternatives.

Difference between DBMS and RDBMS

No.	DBMS	RDBMS
1)	DBMS applications store data as file.	RDBMS applications store data in a tabular form.
2)	In DBMS, data is generally stored in either a hierarchical form or a navigational form.	In RDBMS, the tables have an identifier called primary key and the data values are stored in the form of tables.
3)	Normalization is not present in DBMS.	Normalization is present in RDBMS.
4)	DBMS does not apply any security with regards to data manipulation.	RDBMS defines the integrity constraint for the purpose of ACID (Atomicity, Consistency, Isolation and Durability) property.
5)	DBMS uses file system to store data, so there will be no relation between the tables.	in RDBMS, data values are stored in the form of tables, so a relationship between these data values will be stored in the form of a table as well.
6)	DBMS must provide some uniform methods to access the stored information.	RDBMS system supports a tabular structure of the data and a relationship between them to access the stored information.
7)	DBMS does not support distributed database.	RDBMS supports distributed database.
8)	DBMS is meant to be for small organization and deal with small data. it supports single user.	RDBMS is designed to handle large amount of data. it supports multiple users.
9)	Examples of DBMS are file systems, xml etc.	Example of RDBMS are MySQL, Postgres, SQL server, oracle etc.

Normalization

Normalization is a process used in database design to organize tables and minimize redundancy. There are several normalization techniques, each with its own rules and objectives.

Types of Normalization Techniques

1. **First Normal Form (1NF):**
    - Eliminates repeating groups within a table.
    - Ensures that each column contains atomic values.
    - Example: Breaking down a table of customer orders into individual rows for each order item.

2. **Second Normal Form (2NF):**
    - Builds on 1NF.
    - Ensures that non-key attributes are fully functionally dependent on the primary key.
    - Removes partial dependencies.
    - Example: Splitting a table of orders into separate tables for orders and order details, where order details depend fully on the order ID.

3. **Third Normal Form (3NF):**
    - Builds on 2NF.
    - Eliminates transitive dependencies between non-key attributes.
    - Ensures that every non-key attribute is dependent only on the primary key.
    - Example: Breaking down a table of employees into separate tables for employee details and department details, removing dependencies on department attributes from the employee table.

4. **Boyce-Codd Normal Form (BCNF):**
    - A stricter form of 3NF.
    - Ensures that every determinant is a candidate key.
    - Example: Decomposing a table of student enrolments into separate tables for student details and course details, ensuring each determinant uniquely determines all other attributes.

5. **Fourth Normal Form (4NF):**
    - Addresses multi-valued dependencies.
    - Ensures that no multi-valued dependencies exist between attributes.
    - Example: Splitting a table of customer preferences into separate tables for individual preferences, removing multi-valued dependencies.

6. **Fifth Normal Form (5NF):**
    - Addresses join dependencies.
    - Ensures that all join dependencies are implied by the candidate keys.
    - Example: Decomposing a table of sales transactions into separate tables for products, customers, and sales details, eliminating join dependencies.

7. **Domain-Key Normal Form (DK/NF):**
    - Ensures that every constraint on the relation is a logical consequence of the definition of keys and domains.
    - Example: Verifying that all attributes in a table are dependent on the key and domain constraints defined for that table.

8. **Sixth Normal Form (6NF):**

- Deals with the reduction of data redundancy by further eliminating inter-table redundancy.
- Involves decomposing multi-valued dependencies (MVDs) into separate tables.
- Example: Breaking down a table of product specifications into separate tables for each product attribute, ensuring no redundancy across tables.

9. **Domain-Key 4th Normal Form (DKNF):**
    - An extension of BCNF that considers functional dependencies on individual attributes, known as domains.
    - Ensures that the database schema is free of all modification anomalies and can accommodate all possible updates without any inconsistencies.
    - Example: Splitting a table of employee salaries into separate tables for employee details and salary details, ensuring that changes in salary do not affect other employee attributes.