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.

DBMS

A database management system (DBMS) is system software for creating and managing databases. A DBMS makes it possible for end users to create, protect, read, update and delete data in a database. The most prevalent type of data management platform, the DBMS essentially serves as an interface between databases and users or application programs, ensuring that data is consistently organized and remains easily accessible.

Database Management System is used to manage data from a database. Some popular databases are MySQL, Oracle, MongoDB, etc.

With the help of DBMS, data Security, data Backup, manages huge amount of data, data export and import, serving multiple concurrent database requests, and it gives us a way to manage the data using programming languages are fundamental aspects that contribute to the overall efficiency, reliability, and accessibility of a well-organized and structured database system.

A collection of information which is managed such that it can be updated and easily accessed is called a database. A software package which can be used to manipulate, validate and retrieve this database is called a **Database Management System**.

For example, Airlines use this software package to book tickets and confirm reservations which are then managed to keep a track of the schedule.

There are majorly four types of database:

- **Network Database:** When the details of multiple members can be linked to the files of multiple owners and vice versa, it is called a network database.
- **Hierarchical Database:** When the data stored in the form of records and is connected to each other through links is called hierarchical database. Each record comprises fields and each field comprises only one value.
- **Relational Database:** When the data is organised as a set of tables comprising rows and columns with a pre-defined relationship with one another, it is called a relational database.
- **Object-oriented Database** – the information is represented as objects, with different types of relationships possible between two or more objects. Such databases use an object-oriented programming language for development.

Advantages of DBMS

- DBMS offers a variety of techniques to store & retrieve data

Example: In a relational DBMS, data can be stored using tables, and queries can be executed using SQL to retrieve information. For instance, consider a customer database where information is organized into tables like 'Customers', 'Orders' and 'Products'.

- DBMS serves as an efficient handler to balance the needs of multiple applications using the same data

Example: Consider a retail business where the same database is used by an inventory management system, an online shopping platform, and a customer relationship management (CRM) system. A DBMS efficiently manages concurrent requests from these diverse applications, ensuring data consistency and integrity.

- Uniform administration procedures for data

Example: A DBMS provides a centralized interface for administrators to manage the database. Regardless of the underlying complexity of the data structures or storage mechanisms, administrators can use a standardized set of commands or tools to perform tasks like backup, recovery, and access control.

- Application programmers are never exposed to details of data representation and storage.

Example: In a DBMS, application programmers can focus on writing code for business logic rather than dealing with low-level details of how data is stored on disk. This abstraction enhances productivity and allows developers to work at a higher level of abstraction.

- A DBMS uses various powerful functions to store and retrieve data efficiently.

Example: Functions like indexing and query optimization in a DBMS contribute to efficient data retrieval. For instance, an index on a 'CustomerID' column allows quick lookup of customer information based on their unique identifier, speeding up search operations.

- Offers Data Integrity and Security

Example: Functions like indexing and query optimization in a DBMS contribute to efficient data retrieval. For instance, an index on a 'CustomerID' column allows quick lookup of customer information based on their unique identifier, speeding up search operations.

Disadvantage of DBMS

- The cost of Hardware and Software of a DBMS is quite high, which increases the budget of your organization.

  Example: Purchasing license for commercial database management systems, such as Oracle Database or Microsoft SQL Server, can be expensive. Additionally, the hardware

infrastructure needed to support a large-scale DBMS deployment, including servers, storage, and networking equipment, contributes to the overall cost.

- Most database management systems are often complex, so training users to use the DBMS is required.

  Example: Training employees on the usage of a complex DBMS like Oracle Database may require significant time and resources. For instance, teaching staff how to design tables, create queries, and manage access control within the system may involve formal training programs.

- In some organizations, all data is integrated into a single database that can be damaged because of electric failure or corruption in the storage media.

  Example: If an organization consolidates all its data into a single centralized database and experience a catastrophic event such as a power outage or storage corruption, it could lead to the loss of critical data. For example, a finance department storing all financial records in a single database faces higher risks if that database becomes corrupted.

- Using the same program at a time by multiple users sometimes leads to data loss.

  Example: If multiple users attempt to modify the same piece of data simultaneously without proper concurrency control, it may result in data loss or corruption. For instance, two users updating the inventory quantity of a product simultaneously might lead to inconsistencies if not properly managed.

- DBMS can’t perform sophisticated calculations.

  Example: While DBMS systems are excellent at managing and retrieving data, they may not be well-suited for performing complex mathematical or statical calculations. For example, executing intricate scientific simulations or data analytics tasks might be more efficiently handled by specialized software outside of the DBMS.

RDBMS

A relational database management system (RDBMS) is a program that allows you to create, update, and administer a relational database. Most relational database management systems use the SQL language to access the database.

A relational database (RDB) is a way of structuring information in tables, rows, and columns. An RDB has the ability to establish links—or relationships–between information by joining tables, which makes it easy to understand and gain insights about the relationship between various data points.

A relational 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. RDBMS store data in the form of tables, with most commercial relational database management systems using Structured query language (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.

Types of RDBMS (Relational Database Management Systems)

Some of the examples of relational database are:

1. **Oracle Database**: Oracle Database is a widely used RDBMS known for its scalability, security, and comprehensive feature set. It offers robust support for data management, high availability, and advanced analytics capabilities.
2. **MySQL**: MySQL is a popular open-source RDBMS that is known for its simplicity, speed, and ease of use. It is widely used in web applications and is known for its scalability, reliability, and compatibility with various platforms.
3. **SQL Server**: Microsoft SQL Server is a powerful RDBMS developed by Microsoft. It offers a wide range of features, including advanced security, business intelligence tools, and integration with other Microsoft products and technologies.
4. **PostgreSQL**: PostgreSQL is a feature-rich open-source RDBMS known for its extensibility and compliance with industry standards. It offers a strong emphasis on data integrity, reliability, and support for advanced SQL features.
5. **IBM Db2:** IBM Db2 is an enterprise-level RDBMS designed for high-performance and scalability. It offers advanced data management capabilities, support for large-scale deployments, and integration with other IBM products and technologies.
6. **Microsoft Access**: Microsoft Access is a relational database management system (RDBMS) developed by Microsoft. It is primarily designed for small-scale database applications, allowing users to create and manage databases without requiring extensive programming knowledge.
7. **Azure SQL:** Azure SQL is a cloud-based relational database service provided by Microsoft Azure. It is based on the popular Microsoft SQL Server database engine and offers a scalable, secure, and managed platform for hosting and managing relational databases in the cloud.

These are just a few examples of the types of RDBMS available. Each RDBMS has its own strengths, features, and use cases, so the choice of the RDBMS depends on factors such as specific

requirements, scalability needs, budget, and preferences of the organization or application developer.

Advantages of RDBMS

- Data Organization:

Example: In an RDBMS, data is organized into tables. For instance, a customer database may have a table for customers, a table for orders, and a table for products, making data organization intuitive and structured.

- Data Integrity:

Example: Using primary keys and foreign keys, an RDBMS ensures data integrity. For example, a "Customers" table might have a primary key on the "CustomerID" column, preventing duplicate customer entries.

- Data Relationships:

Example: In an e-commerce database, a relationship between a "Customers" table and an "Orders" table can be established using customer IDs. This enables querying to retrieve all orders associated with a specific customer.

- Querying and Reporting:

Example: With SQL, users can easily retrieve specific data. For instance, using SQL SELECT statements, one can query a database to find all products with a certain price range or retrieve sales data for a particular period.

- Scalability:

Example: As a business grows, an RDBMS can scale horizontally by distributing data across multiple servers. For instance, a large online retailer may distribute its product catalog across multiple database servers to handle increased traffic.

- Security:

Example: RDBMS systems implement access controls and authentication. For example, a financial database may restrict access to sensitive financial records, allowing only authorized personnel to view or modify the data.

- Data Consistency:

Example: RDBMS maintains referential integrity. If a product's category is updated, this change is automatically reflected in all related tables, ensuring consistency across the database.

Disadvantages of RDBMS

RDBMS may face challenges when dealing with extremely large datasets, as the complexity of managing relationships and maintaining ACID properties may impact performance.

Example: In scenarios where large-scale data processing is required, such as analyzing massive volumes of real-time data in social media analytics or streaming platforms, NoSQL databases designed for horizontal scalability may offer better performance.

Database Normalization

Normalization is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalisation in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

- 1NF (First Normal Form)

Each column is unique in 1NF.

- 2NF (Second Normal Form)

The entity should be considered already in 1NF, and all attributes within the entity should depend solely on the unique identifier of the entity.

- 3NF (Third Normal Form)

The entity should be considered already in 2NF, and no column entry should be dependent on any other entry (value) other than the key for the table. If such an entity exists, move it outside into a new table.3NF is achieved, considered as the database is normalized.

- **BCNF (Boyce-Codd Normal Form)**

Even when a database is in 3rd Normal Form, still there would be anomalies resulted if it has more than one **Candidate** Key.

- 4NF (Fourth Normal Form)

Tables cannot have multi-valued dependencies on a Primary Key.

- 5NF (Fifth Normal Form)

A composite key shouldn't have any cyclic dependencies. Well, this is a highly simplified explanation for Database Normalization. One can study this process extensively, though. After working with databases for some time, you'll automatically create Normalized databases, as it's logical and practical.

- 6NF (Sixth Normal Form)

6th Normal Form is not standardized, yet however, it is being discussed by database experts for some time. Hopefully, we would have a clear & standardized definition for 6th Normal Form in the near future