

# DATABASE MANAGEMENT SYSTEM

## BASICS

### Table of Contents

<b>What is database?</b> .....	2
1. Database Management System ( DBMS) .....	2
2. Why we need DBMS when we had Excel ? .....	2
<b>Types of Databases:-</b> .....	3
1. Relational Database Model .....	4
2. What we used before SQL? .....	5
3. Declarative vs Imperative.....	5
<b>Subparts of SQL:-</b> .....	6
1. OLTP VS OLAP .....	6

# What is database?

A database is an organized collection of data that allows for efficient storage, retrieval, and manipulation of large amounts of data.

## 1. Database Management System ( DBMS)

A database management system (DBMS) is a software system that is used to manage databases. It provides a way for users to create, store, modify, and retrieve data from a database. Examples of popular DBMSs include Oracle, Microsoft SQL Server, MySQL, PostgreSQL, MongoDB, and Cassandra.

DATABASE = Data + DBMS

\*\* Database is software + hardware that allow user to store , organize and use data.

## 2. Why we need DBMS when we had Excel ?

DBMS is required because it can handle large amounts of data efficiently, enforces data integrity, provide multi-user access, and robust security mechanisms, while Excel lacks these features.

## Types of Databases:-

1. **Relational Model:** The relational model is the most widely used database model today. It organizes data into one or more tables consisting of rows and columns, with each table representing a specific entity or relationship between entities. Eg: MySQL , SQLite , PostgreSQL
2. **Object-Oriented Model:** The object-oriented model is based on the concepts of object-oriented programming. It represents data as objects that can contain both data and methods, allowing for more complex data structures and relationships. Eg: db4o, ObjectStore
3. **Document Model:** The document model is used for managing semi-structured or unstructured data, such as JSON or XML documents. In this model, data is stored as a collection of documents, with each document containing a set of key-value pairs. Eg: MongoDB, CouchDB
4. **Key-Value Model:** The key-value model is a simple model that stores data as a set of key-value pairs. This model is useful for applications that require fast access to data, such as caching or session management.  
Eg. Redis, DynamoDB
5. **Hierarchical Model:** The hierarchical model organizes data into a tree-like structure, with each parent node having one or more child nodes. This model is used for storing data with a one-to-many relationship, such as file systems. Eg: IBM Information Management System (IMS)
6. **Network Model:** The network model is similar to the hierarchical model but allows for more complex relationships between entities. In this model, data is organized into sets of records, with each record linked to one or more other records. Eg: Integrated Data System (IDS)
7. **Graph Model:** The graph model represents data as nodes and edges, allowing for the modeling of complex relationships between entities. It is commonly used in applications such as social networks or recommendation engines. Eg: Neo4j, Titan

## 1. Relational Database Model

The relational database model is a way of organizing and managing data in a database. It is based on the concept of a relation or a table, which consists of rows and columns.

The following are some key terms related to the relational database model:

1. **Table or Relation:** A table is a collection of related data, consisting of rows and columns. Each row represents a unique record, and each column represents a specific attribute of that record.
2. **Primary Key:** A primary key is a column or set of columns that uniquely identifies each row in a table.
3. **Foreign Key:** A foreign key is a column or set of columns that refers to the primary key of another table. It is used to establish relationships between tables.
4. **Index:** An index is a data structure that provides fast access to specific rows in a table based on the values in one or more columns.
5. **Normalization:** Normalization is the process of organizing data in a database to reduce redundancy and improve data consistency.
6. **SQL:** SQL (Structured Query Language) is a programming language used to manage and query relational databases. It allows users to create, modify, and query data in tables.

Overall, the relational database model provides a flexible and efficient way to organize and manage data in a database, allowing for fast and easy retrieval of data based on various criteria.

**\*\* Main point that makes relational model special is the ability to link relationships between different types of data using primary and foreign keys.**

Other terms:-

1. Cardinality: Number of rows (tuples) in table.
2. Degree: Number of attributes (columns) in a table.
3. Schema: Description of structure of database including tables, columns, relationships.
4. Entity: Real-world object or concept stored in database.

## 2. What we used before SQL?

Before SQL, the most common method of storing and managing data was through file systems. Data was stored in individual files, and applications would read and write to those files directly. This method had several limitations, including:

1. Data redundancy: Each file would often contain redundant data, leading to wasted storage space.
2. Limited data access: Accessing data required navigating through file directories, which could be time-consuming and difficult.
3. Lack of data consistency: There was no easy way to ensure that data was consistent across multiple files or applications.
4. Limited data querying: Retrieving data required custom programming for each application, making it difficult to perform complex queries.

## 3. Declarative vs Imperative

SQL is declarative language. We just need to say what we require and we get the output.

Declarative programming focuses on what the program should accomplish rather than how it should accomplish it. In declarative programming, the programmer specifies the desired outcome or goal, and the programming language or system determines how to achieve that goal. Examples of declarative programming languages include SQL and HTML.

Imperative programming, on the other hand, focuses on how the program should accomplish its tasks. In imperative programming, the programmer specifies a series of instructions or steps that the computer must follow to achieve the desired outcome. Examples of imperative programming languages include C, Python, and Java.

## Subparts of SQL:-

1. **DDL (Data Definition Language):** DDL is used to define the structure of the database. Commands such as CREATE, ALTER, and DROP are used to create and modify database objects such as tables, indexes, and views.
2. **DML (Data Manipulation Language):** DML is used to manipulate data within the database. Commands such as SELECT, INSERT, UPDATE, and DELETE are used to retrieve, add, modify, and delete data from tables.
3. **DCL (Data Control Language):** DCL is used to control access to the database. Commands such as GRANT and REVOKE are used to grant or revoke privileges and permissions to database users.
4. **TCL (Transaction Control Language):** TCL is used to control transactions in the database. Commands such as COMMIT and ROLLBACK are used to commit or rollback transactions and manage transactional integrity.
5. **DQL (Data Query Language):** DQL is a subset of DML that is focused specifically on querying data from the database. The SELECT command is the primary command used in DQL.

### 1. OLTP VS OLAP

OLTP and OLAP are two different types of systems for managing and analyzing data in a database.

**OLTP (Online Transaction Processing)** systems are designed for processing and managing large volumes of transactions in real-time. They are used in day-to-day business operations to handle activities such as order processing, inventory management, and financial transactions. OLTP systems are typically optimized for data insert, update, and delete operations, and they focus on maintaining data integrity and ensuring that transactions are processed quickly and accurately.

**OLAP (Online Analytical Processing)** systems, on the other hand, are designed for analyzing large volumes of data to support decision-making processes. They are used to query, analyze, and report on data in a database to help identify trends, patterns, and insights. OLAP systems are optimized for complex queries that involve aggregating and summarizing large amounts of data, and they typically use multi-dimensional data models to provide users with a more intuitive view of the data.