fingerTips

# Module-1

# Data, Database & DBMS

## 1) Data

Data refers to any information that can be stored, processed, and analyzed by a computer. This information can take many forms, such as text, numbers, images, audio, and video.

For example, suppose you have a dataset containing information about customers who have purchased products from an online store. The data might include:

- Customer name
- Address
- Email address
- Phone number
- Product name
- Product description
- Price
- Quantity
- Order date

## 2) Database

A database is a collection of organized data that is stored electronically in a computer system. The purpose of a database is to provide a way to store and retrieve data efficiently and securely. Databases are used in a wide range of applications, including business, finance, healthcare, education, and more.

## 3) Types of Databases

There are several types of databases, each with its own unique features and characteristics. Here are some of the most common types of databases:

i. Relational Databases: Relational databases are the most widely used type of database. They store data in tables with a predefined structure, where each row represents a record and each column represents a field. The tables are related to each other through keys that enable data to be accessed and managed efficiently. Examples of relational databases include MySQL, Oracle, Microsoft SQL Server, and PostgreSQL.

ii. NoSQL Databases: NoSQL databases are non-relational databases that store data in a variety of formats, including key-value pairs, document-oriented, and graph-based. NoSQL databases are designed to handle large amounts of unstructured or semi-structured data, making them well-suited for use cases such as big data analytics, content management, and real-time applications. Examples of NoSQL databases include MongoDB, Couchbase, Cassandra, and Amazon DynamoDB.

iii. Object-Oriented Databases: Object-oriented databases store data in the form of objects, which contain both data and methods. This type of database is particularly useful for applications that require complex data structures, such as software development and scientific research. Examples of object-oriented databases include ObjectDB and db4o.

iv. Cloud Databases: Cloud databases are databases that are hosted in the cloud, providing users with scalable and flexible storage solutions. Cloud databases can be accessed from anywhere with an internet connection, making them ideal for applications that require remote access or collaboration. Examples of cloud databases include Amazon RDS, Microsoft Azure SQL Database, and Google Cloud SQL.

### *4) Relational & Non-Relational databases*

Relational and non-relational databases are two different types of databases that differ in how they organize and store data.

Relational databases store data in a structured format, where information is organized into tables with columns and rows. The relationships between these tables are established through the use of keys, which are unique identifiers that allow data from one table to be linked to data in another table. SQL (Structured Query Language) is the primary language used to manage and query data in relational databases. Examples of relational databases include MySQL, Oracle, and Microsoft SQL Server.

On the other hand, non-relational databases, also known as NoSQL databases, do not rely on the same rigid structure as relational databases. Instead, they use a more flexible schema-less approach to store data. Non-relational databases can be categorized into four types: document-oriented, key-value, column-family, and graph databases. Each type of NoSQL database is optimized for different types of data and use cases. Examples of non-relational databases include MongoDB, Cassandra, and Neo4j.

In short, relational databases are structured and use SQL to manage data, while non-relational databases are flexible and schema-less, allowing for easier scalability and data management.

## *5) Database components*

There are several components that make up a typical database management system. These components can vary depending on the specific database system being used, but here are some of the most common ones:

i. Data: This refers to the actual information being stored in the database. This can include anything from customer names and addresses to sales figures and product inventory levels.
ii. Database schema: This is the overall structure of the database, including its tables, columns, relationships, and constraints. The schema defines how the data is organized and how it can be accessed.
iii. Database management system (DBMS): This is the software that manages and controls access to the database. The DBMS is responsible for tasks such as data backup, security,

performance tuning, and data retrieval.

iv. Users: These are the individuals or applications that interact with the database. Users can be categorized into different roles with varying levels of permissions to access, modify or manipulate the data stored in the database.

v. Queries: This is the means by which users can retrieve and manipulate data in the database. Queries are typically expressed using SQL (Structured Query Language), which is a standard language for managing relational databases.

vi. Indexes: These are special structures that help to improve the performance of queries by speeding up data retrieval. Indexes are created on specific columns in a table and can be used to quickly locate specific rows of data.

vii. Transactions: These are a sequence of database operations that are performed as a single unit of work. Transactions help to ensure the consistency and integrity of the database by ensuring that all changes are either fully committed or fully rolled back.

viii. Backup and recovery: These are critical components of any database management system that help to ensure that data can be recovered in the event of a system failure or other catastrophic event. Backups are typically taken on a regular basis, and recovery procedures are tested to ensure that they work as expected.

ix. These are just some of the key components that make up a typical database management system. Other components may include security controls, data dictionaries, triggers, and stored procedures, depending on the specific database system being used.

## *6) Database management System (DBMS)*

DBMS stands for Database Management System. A DBMS is a software system that allows users to create, manage, and manipulate databases. A database is a collection of data that is organized in a way that enables efficient storage, retrieval, and manipulation of the data. The purpose of a DBMS is to provide an interface between the user and the database, allowing the user to interact with the data in a meaningful way.

A DBMS provides the following capabilities:

i. Data definition: This involves defining the structure of the database, including its tables, columns, data types, and relationships between tables.
ii. Data manipulation: This involves inserting, updating, and deleting data in the database. Users can interact with the database using SQL (Structured Query Language), which is a standard language for managing relational databases.
iii. Data retrieval: This involves retrieving data from the database in response to user queries. Users can specify complex queries using SQL to retrieve specific data from the database.
iv. Data security: This involves controlling access to the database and ensuring that sensitive data is protected from unauthorized access.
v. Data backup and recovery: This involves creating backups of the database to protect against data loss due to system failure, human error, or other catastrophic events.

## *7) Relational Database management System (RDBMS)*

RDBMS stands for Relational Database Management System. It is a software system that enables users to create, update, and manage relational databases.

Relational databases are a type of database that stores data in tables, which are related to each other by common fields or keys. In an RDBMS, the data is organized into tables, each of which consists of rows and columns. Each row represents a unique record, and each column represents a specific attribute of that record.

The key benefits of an RDBMS are that it provides a standardized way of organizing and accessing data, and ensures data integrity and consistency. It also supports complex queries and provides the ability to manage large datasets efficiently.

Some popular examples of RDBMS software include MySQL, Oracle, Microsoft SQL Server, and PostgreSQL.

### 8) Features of RDBMS

Some of the key features of RDBMS include:

i. Support for relational data structures
ii. Ability to enforce data integrity through the use of constraints
iii. Support for transactions to ensure data consistency
iv. Use of SQL for data manipulation and retrieval
v. Support for indexing and querying of data

### 9) Entity & Relationship

In context of database design, an entity is a person, place, thing, concept, or event that can be identified and represented in the database. An entity can be tangible (such as a customer, product, or employee) or intangible (such as an order, appointment, or transaction). In the context of a database, an entity is represented as a table, with each row in the table representing a specific instance of the entity.

A relationship is a connection or association between two or more entities in the database. Relationships are expressed as verbs, such as "belongs to", "is part of", "is managed by", or "is located in". Relationships are important because they help to establish the structure and organization of the database, and can be used to ensure data integrity and consistency.

### 10) Entity Relationship Diagram (ERD)

An Entity Relationship Diagram (ERD) is a graphical representation of the entities, relationships, and attributes that make up a database. An ERD is typically created during the database design process to help visualize the structure of the database and to identify potential issues and areas for optimization. An ERD consists of entities represented as rectangles, relationships represented as lines connecting the entities, and attributes represented as ovals within the rectangles.

In an ERD, entities are represented by rectangles, with the entity name written inside the rectangle. Each entity has a unique identifier, called a primary key, which is used to distinguish it from other entities in the database. Relationships between entities are represented by lines connecting the rectangles, with the relationship type (such as "belongs to" or "has many") written on the line. Attributes are represented as ovals within the rectangles, with the attribute name written inside the oval.

ERDs are an important tool for database designers and developers, as they provide a clear and intuitive way to understand the structure of the database and the relationships between its components. They can also be used to identify potential issues with the database design, such as redundant data or inefficient relationships.

### *11) Normalization & Normal forms*

Normalization is the process of organizing data in a database to minimize redundancy and dependency, thus improving data integrity and efficiency of data retrieval. There are different types of normal forms that describe how normalized a database is:

i. First Normal Form (1NF): a table is in 1NF if it has no repeating groups or arrays, and each attribute contains only atomic values (i.e., values that cannot be further decomposed).

ii. Second Normal Form (2NF): a table is in 2NF if it is in 1NF and every non-key attribute is functionally dependent on the primary key (i.e., no partial dependencies).

iii. Third Normal Form (3NF): a table is in 3NF if it is in 2NF and every non-key attribute is independent of any other non-key attribute (i.e., no transitive dependencies).

iv. There are additional normal forms beyond 3NF, such as

Boyce-Codd Normal Form (BCNF) and Fourth Normal Form (4NF), but they are less commonly used.

## *12) ACID properties*

ACID is an acronym that stands for Atomicity, Consistency, Isolation, and Durability. These are properties that guarantee the reliability and consistency of transactions in a database:

i. Atomicity: a transaction is atomic if it is treated as a single, indivisible unit of work, meaning that either all of its changes are committed or none of them are. This ensures that the database remains consistent even in the event of failures or errors during the transaction.

ii. Consistency: a transaction is consistent if it ensures that the database remains in a valid state after it completes. This means that all constraints, rules, and relationships defined in the database are respected by the transaction.

iii. Isolation: a transaction is isolated if its changes are invisible to other transactions until it is committed. This ensures that concurrent transactions do not interfere with each other and that the database remains in a consistent state.

iv. Durability: a transaction is durable if its changes are persistent and cannot be lost or undone even in the event of system failures or crashes. This ensures that the database remains reliable and recoverable.