1. What is data

Data is collection of raw facts , figures or observations stored and may not have meaning

Raw and unprocessed

Exists in various forms text numbers, images multimedia

EX. Student id 1o235

Name: avi

Mark: 90

* Whan we process or organize data it becomes information

Ex Data: “avi” , ”BCA” , ”75%”

Information: “avi is a student who scored 75%”

**Important Definitions:-**

Data = stored values

Database = collection of organized data

Application = software that uses that database to provide meaningful informaion

DBMS = software to manage and operate database

Scheme = Blueprint structure of the databser

Instance = Actual data ata specific point in time

Tuple (Record) = a row in table

Attrivute (field) = a colume in a table

Primary key = unique identifier for a record

Foreign key = attribute linking tow tables

Narmalization = process to reduce redundancy

ACID Properties atomicity , consistency, isolation, duracility

**Why use database**

* Data integrity – consistency across operations
* Security – controlled access
* Scalability – handles large datasets
* Concurrency – supports multiple users simulataneously
* Data independence
* Reduced redundancy – if the same piece of data is stored in many places it wastes space and can cause inconsistency. That to stored in one table to reduced data and space
* Efficient querying
* **Database Components**

1. Hardware – physical devices where the database runs and stored. Ex Servers, storage devices, hard disks.
2. Software – programs that manage the database. Ex DBMS Like mysql oracle.
3. Data – tha actual information stored in the database Ex student record.
4. Access language – the language used to interact with the database Ex SQL
5. Procedures – Rules and methods to use and manage the database Ex backup procedures

* **Database Management system**

Software that manages database

Provides interface between users/application and the database

* **Metadata in DBMS**

Definition: Metadata is “data about data”

**Purpose:**

Describes the structure of the databse

Defines tables attributes, data types, and constraints.

Includes indexing, storage details, and relationships.

**Stroed in :** **Data dictionary or system catalog.**

* **Database State / instance**

The data stored in the database at a particular moment in time.

* **Characteristics:**
  + Represents a snapshot of the database.
  + Can change frequently as updates, insertions, and deletions occur.

**Schema –** The structure. That was like a any table header part Ex. Name, id, mark.

**Instance/state –** The actual content, that was like a any table records Ex. {avi,1,75},{aaryaan,2,80}

* **Database Users**
  + **End Users**
    - People who use application to interact with the database.
    - Ex. Student checking results.
  + **Application programmers**
    - They was who write a software that user for database.
    - Ex. Developers creating apps for banking e-commerces.
  + **Database Administrators**
    - Manage the database system, ensure security, backup, and performance.
    - Ex IT staff handling database.
  + **System Analysts/Designers**
    - That person they design the database structure and define how data should be stored and accessed
* **Database Language:**
  + **Data Definition Language (DDL) –** schema creation
  + **Data Query Language (DQL)** – select queries
  + **Data Manipulation Language (DML)** – insert, update, delecte
    - In DML have 2 type
      * Procedural
      * Non-procedural
* **Database in modern application**
  + **Cloud database –** scalable, managed cost-effective **Ex.** AWS RDS, Google firestore, MongoDB Atlas
  + **Big data integration –** works with Hadoop/spark to handle volume, velocity, and variety of data
  + **AI/ML-ready database** – support vector search, predictive analytics, and recommendations. Preparing the data for training ML models. Running ML algorithms directly inside the database. Providing real-time predications without needing to move data outside
  + **Multi-Model Database** – support relational, document, graph, and key-value models in one system
  + **Edge database** – enable real time processing close to IOT devices.
  + **Serverless Database** – auto scale and charge only for usage.
* **Data model:**
  + A data model is way of organizing and representing data in a database
  + It defines:
    - How data is stored (tables, objects)
    - How data is related (relationships between entities)
    - What rules apply (constraints, keys).
  + **Tow broad categories**:
    - Relational (SQL)
    - Non-relational (NoSQL)

**A** Database model defines the logical design and structure of a database

* **Relational Model**
  + Based on tables (relations) -> rows (tuples) & columns (attributes)
  + Uses schema with predefined rules.
  + Uses SQL (Structured Query Language).
  + Focus on relationships between data entities.
  + Database systems: PostgreSQL, MySQL.
* Key features of relational model
  + Structure – organized into tables with rows and columns.
  + Integrity – enforces primary keys, foreign keys, and constraints.
  + Consistency – ACID compliance (Atomicity, consistency, Isolation, durability)
  + Querying – rich SQL operations (joins, subqueries, aggregations).
  + Best for transactional systems.
* When to use relational models?
  + Applications requiring strong consistency.
  + Structured data with predefined schema.
  + Complex queries and transactions.
  + Ex. Banking, ERP systems.
* **Non-relational model**
  + Not table-based – flexible structures (key-value, document, graph, column).
  + Often called NoSQL databases.
  + Querying via APIs or specific query language.
  + Schema-less or dynamic schema.
  + Ex. MongoDB storing JSON documents.
* Key features of non-relational model.
  + Schema flexibility – supports documents, key-value pairs, graphs.
  + Schema-less – data structures can vary across records.
  + Horizontal scalability – handles big data.
  + High performance.
  + Best for real-time, large-scale, unstructured or semi-structured data.
* When to use Non-relational models?
  + Applications needing flexibility and rapid iteration.
  + Large-scale, high-velocity data.
  + Seme-structured or unstructured data.
  + Ex. Social media, IOT, recommendation engines.