Unit 1. Introduction

1.1. History, database system and its applications

1.2. Characteristics of DBMS

1.3. Application architecture (one tier, two tier, n-tier)

1.4. Data abstraction and Independence

1.5. Schemas and Instances

1.6. Database Manager and users

Datum:

- **Definition:** A datum is a single piece of raw data. It is an individual element of information that by itself may not convey much meaning.
- **Example:**
  - "42" is a datum representing a number.
  - "September 14, 2024" is a datum representing a specific date.

Data:

- **Definition:** Data is a collection of multiple pieces of data (or datums) that are often unprocessed. Data can be quantitative or qualitative and consists of various units of measurement or observation.
- **Example:**
  - A list of numbers like [42, 67, 89] represents data.
  - A database table with multiple rows of entries such as names, ages, and addresses.

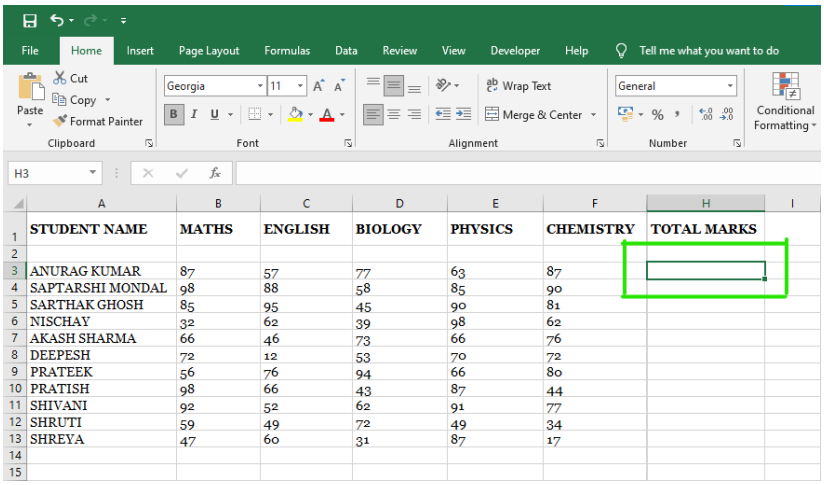
Information:

- **Definition:** Information is what you get when data is processed, organized, or interpreted to provide meaning and context. It helps in decision-making and understanding patterns or insights.
- Converting data into information involves organizing and analyzing it to make it meaningful.
- **Example:**
  - A sales report that shows trends and insights over a period, derived from raw sales data.
  - A summary of customer reviews that highlights common themes and sentiment.

History, Database System, and Its Applications

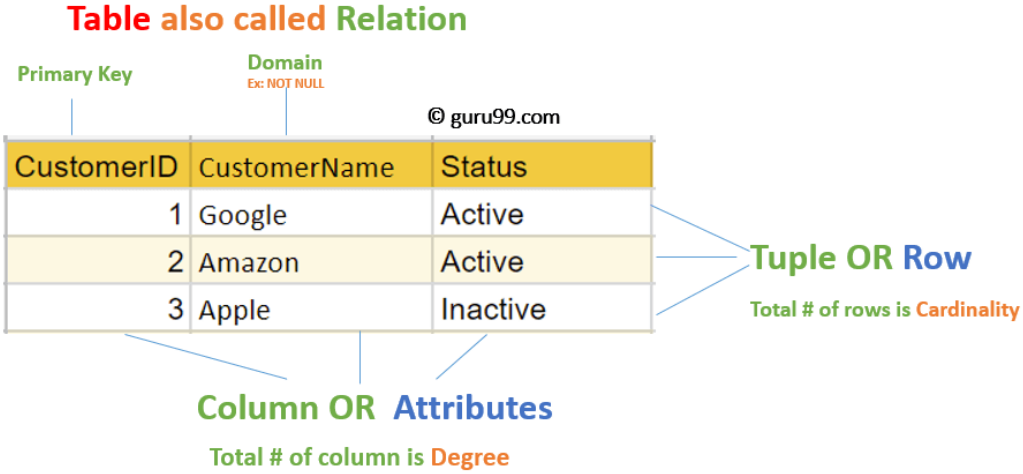
History:

- **Early Beginnings:** Databases have their roots in early data management systems like file systems and flat files. In the 1960s, the concept of databases started to evolve with hierarchical and network databases.

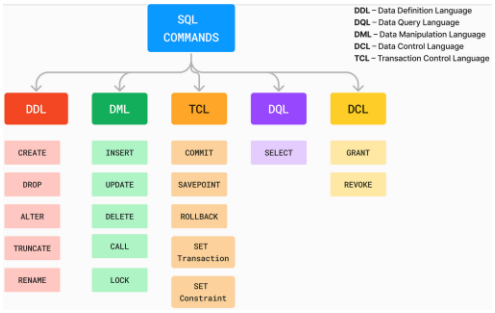


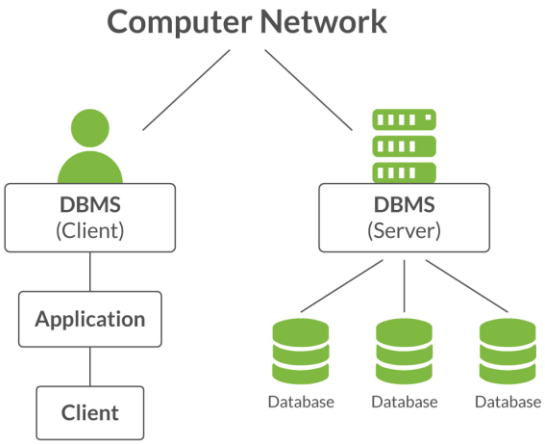
	A	B	C	D	E	F	H
1	STUDENT NAME	MATHS	ENGLISH	BIOLOGY	PHYSICS	CHEMISTRY	TOTAL MARKS
2							
3	ANURAG KUMAR	87	57	77	63	87	
4	SAPTARSHI MONDAL	98	88	58	85	90	
5	SARTHAK GHOSH	85	95	45	90	81	
6	NISCHAY	32	62	39	98	62	
7	AKASH SHARMA	66	46	73	66	76	
8	DEEPESH	72	12	53	70	72	
9	PRATEEK	56	76	94	66	80	
10	PRATISH	98	66	43	87	44	
11	SHIVANI	92	52	62	91	77	
12	SHRUTI	59	49	72	49	34	
13	SHREYA	47	60	31	87	17	
14							
15							

- **Relational Model:** Introduced by E.F. Codd in 1970, the relational model used tables (relations) to represent data and relationships. This became the foundation for modern relational database management systems (RDBMS).

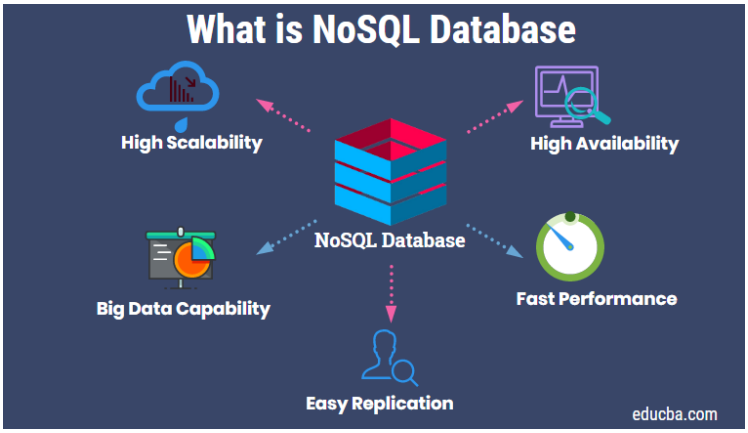


- **SQL:** Structured Query Language (SQL) became the standard for querying and managing relational databases. It was standardized by ANSI in 1986.





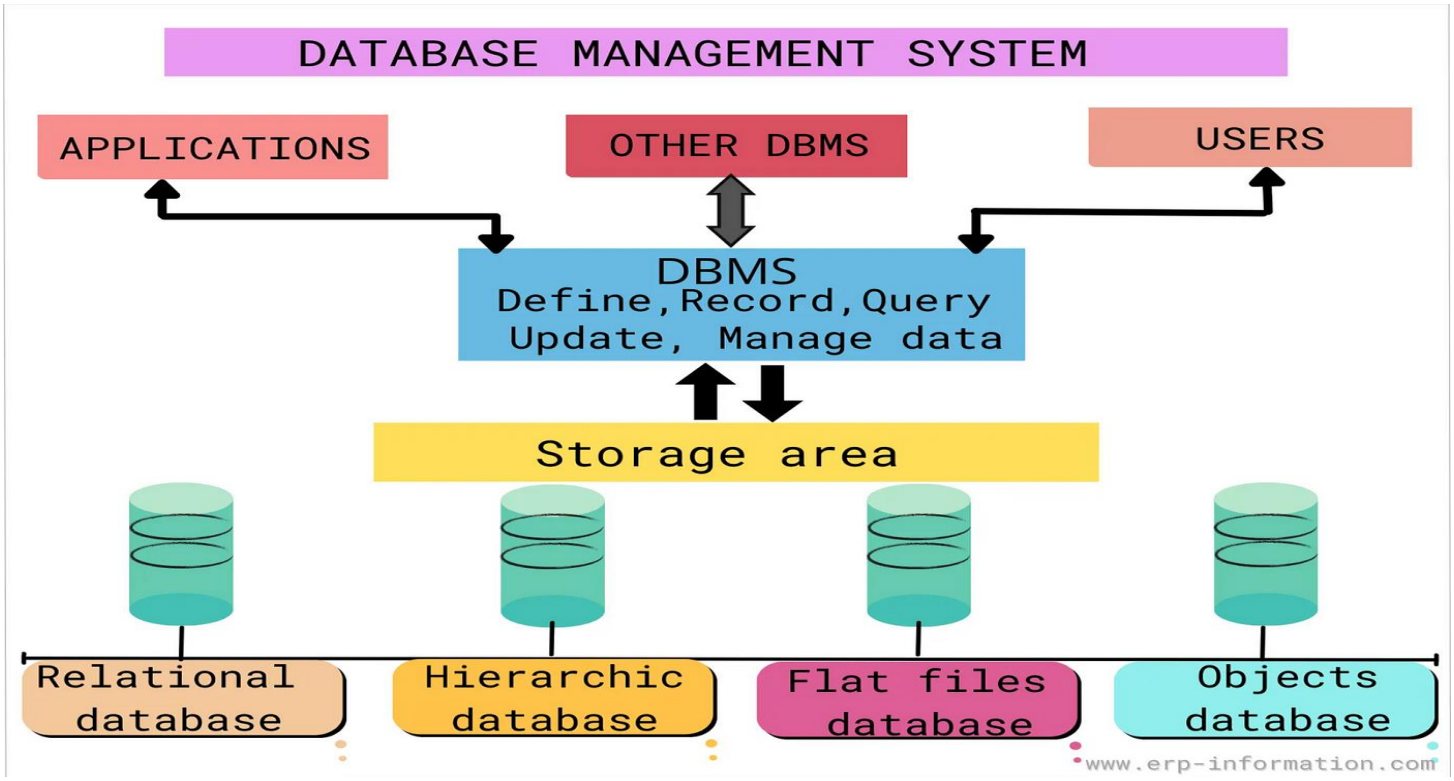
- **NoSQL and Beyond:** In the 2000s, NoSQL databases emerged to handle unstructured data and high-velocity workloads. These include key-value stores, document stores, column-family stores, and graph databases.



Database System:

- **Definition:** A database system is a software application that manages and facilitates access to databases. It provides tools for creating, querying, updating, and managing data.

- **Components:** Typically includes a database engine (for handling data storage and retrieval), a database schema (defining the structure), and a database management system (DBMS) which provides the user interface and interaction tools.



Applications:

- **Transactional Systems:** Used in banking, retail, and e-commerce for handling transactions.
- **Analytical Systems:** Used in business intelligence and data warehousing for analysis and reporting.
- **Web and Mobile Applications:** Handle user data, preferences, and interactions.
- **Enterprise Applications:** Manage complex business processes and data across various departments.

Characteristics of DBMS

- **Data Abstraction:** Provides a simplified view of the data, hiding the complexities of the underlying data storage.
- **Data Integrity:** Ensures accuracy and consistency of data through constraints, rules, and validation.
- **Data Security:** Controls access to data through user authentication and authorization.

- **Concurrency Control:** Manages simultaneous data access by multiple users to prevent conflicts and ensure consistency.
- **Recovery:** Provides mechanisms to recover data in case of failures or crashes.
- **Data Independence:** Allows changes to the data structure without affecting the application programs.

Application Architecture (One-Tier, Two-Tier, N-Tier)

- **One-Tier Architecture:**
  - **Description:** The database and application logic are on the same machine or layer.
  - **Example:** A desktop application that directly accesses a local database.

- **Two-Tier Architecture:**
  - **Description:** The application is split into two layers: the client and the server. The client interacts with the database server directly.
  - **Example:** A client-server application where the client handles the user interface and business logic, while the server manages the database.

- **N-Tier Architecture:**
  - **Description:** The application is divided into multiple layers, typically including presentation, application logic, and data management. This design enhances scalability and maintainability.
  - **Example:** A web application with a presentation layer (web server), business logic layer (application server), and data layer (database server).

Data Abstraction and Independence

- **Data Abstraction:**
  - **Concept:** Hides the complexity of the database system by providing different levels of data representation.

- **Levels:**
  - **Physical Level:** Describes how data is physically stored.
  - **Logical Level:** Describes what data is stored and the relationships among the data.
  - **View Level:** Describes how data is presented to users.
- **Data Independence:**
  - **Concept:** The ability to change the schema at one level of the database system without affecting other levels.
  - **Types:**
    - **Physical Data Independence:** Changes to the physical storage do not affect the logical schema.
    - **Logical Data Independence:** Changes to the logical schema do not affect the view level.

Schemas and Instances

- **Schemas:**
  - **Definition:** The structure of the database, including tables, columns, relationships, and constraints. It defines the organization of data.
  - **Types:**
    - **Conceptual Schema:** Represents the entire database structure as a whole.
    - **Logical Schema:** Defines the logical structure of the data (e.g., tables, keys).
    - **Physical Schema:** Describes how data is stored on hardware (e.g., file structures).
- **Instances:**
  - **Definition:** The actual data stored in the database at a particular point in time. It represents the current state of the database.

Database Manager and Users

- **Database Manager:**
  - **Role:** A component of the DBMS responsible for handling database operations such as data storage, retrieval, and updates. It includes the database engine, which executes SQL queries and maintains data integrity.
  - **Responsibilities:**

- **Query Processing:** Interprets and executes database queries.
- **Transaction Management:** Ensures ACID properties (Atomicity, Consistency, Isolation, Durability) for transactions.
- **Backup and Recovery:** Manages data backups and restores in case of failure.

- **Users:**
  - **Types:**
    - **Database Administrators (DBAs):** Manage and maintain the database system, handle security, backups, and performance tuning.
    - **Application Programmers:** Develop and maintain applications that interact with the database.
    - **End Users:** Use applications to perform tasks such as data entry, retrieval, and reporting.
    - **Data Analysts:** Analyze and interpret data, often using queries and reporting tools.