



DBMS

Ahmed Elnemr



Objectives



- The difference between data and information.
- What a database is, the various types of databases, and why they are valuable assets for decision making.
- The importance of database design.
- How modern databases evolved from file systems.
- About flaws in file system data management.
- The main components of the database system.
- The main functions of a database management system (DBMS)



Why Databases?



- Databases solve many of the problems encountered in data management.
- Used in almost all modern settings involving data management:
 - Business
 - Research
 - Administration



Data vs. Information



- Data are raw facts
- **Information** is the result of processing raw data to reveal meaning
- **Information** requires context to reveal meaning
- Raw data must be formatted for storage, processing, and presentation
- Data are the foundation of information, which is the bedrock of *knowledge*.
- Data: building blocks of information.
- **Information** produced by processing data.
- **Information** used to reveal meaning in data.

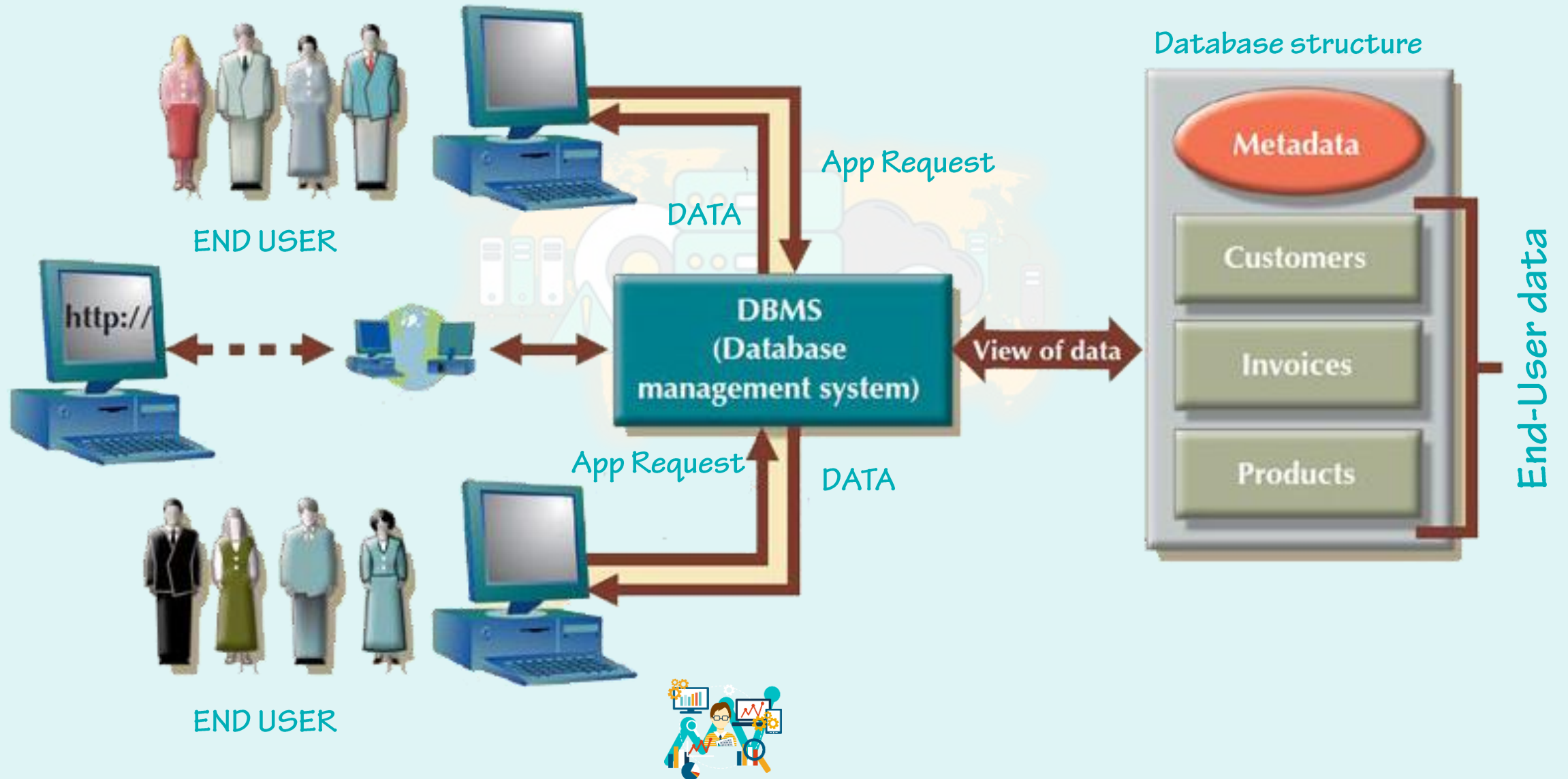


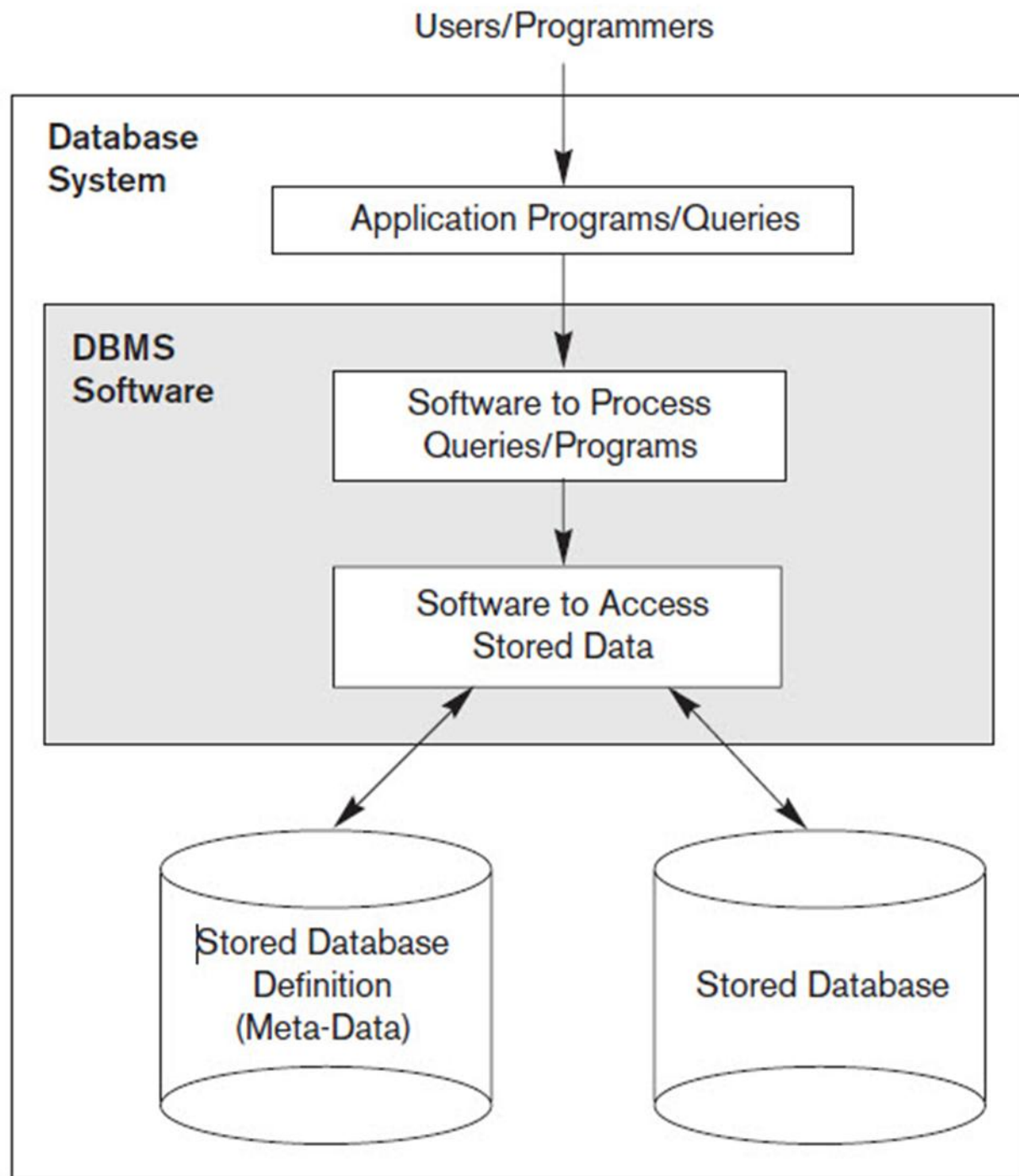
Introducing the Database



- Database: shared, integrated computer structure that stores a collection of:
 - **End-user data**: raw facts of interest to end user,
 - **Metadata**: data about data.
- Database management system (DBMS):
 - collection of programs
 - **Manages structure and controls access to data**









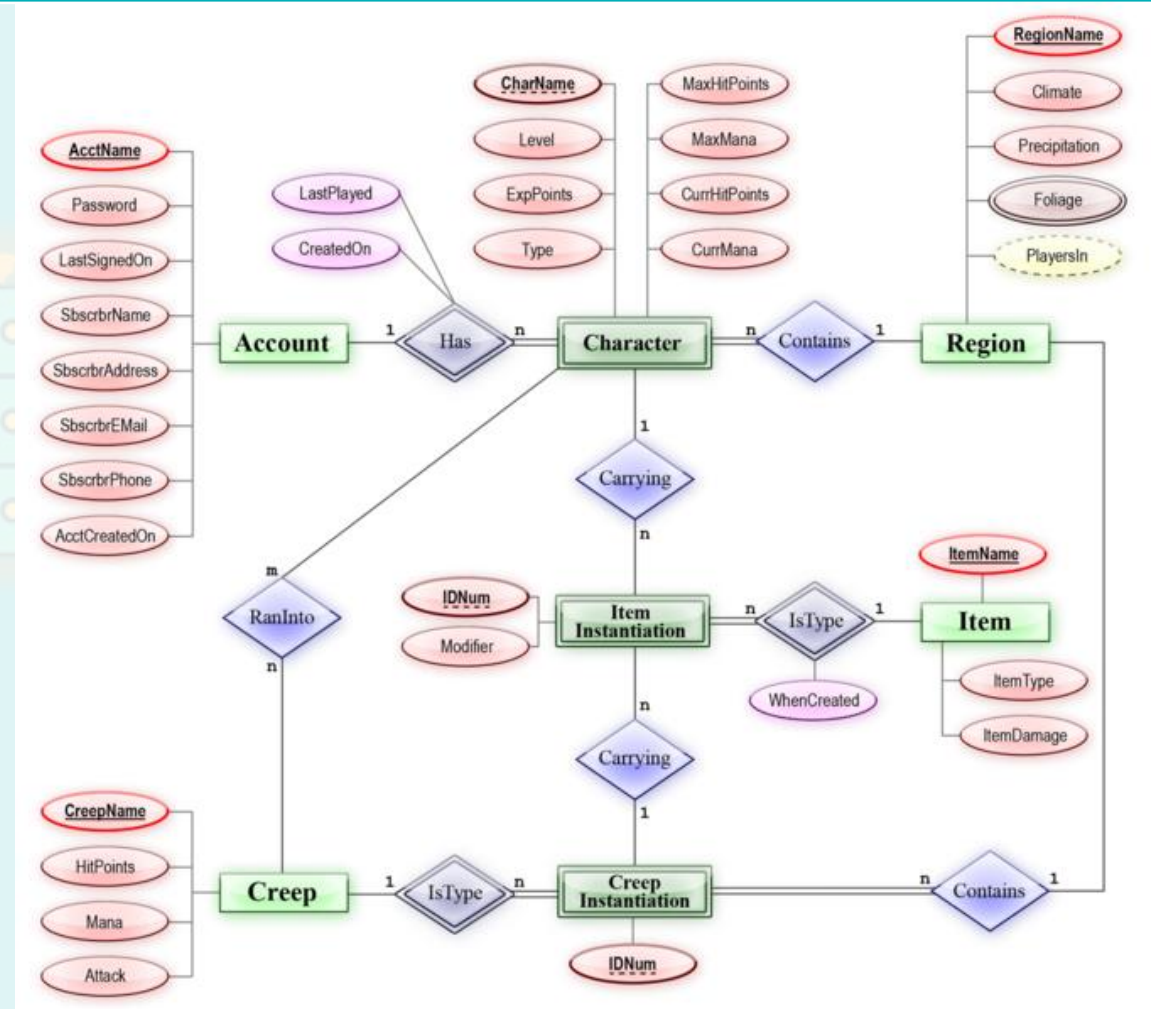
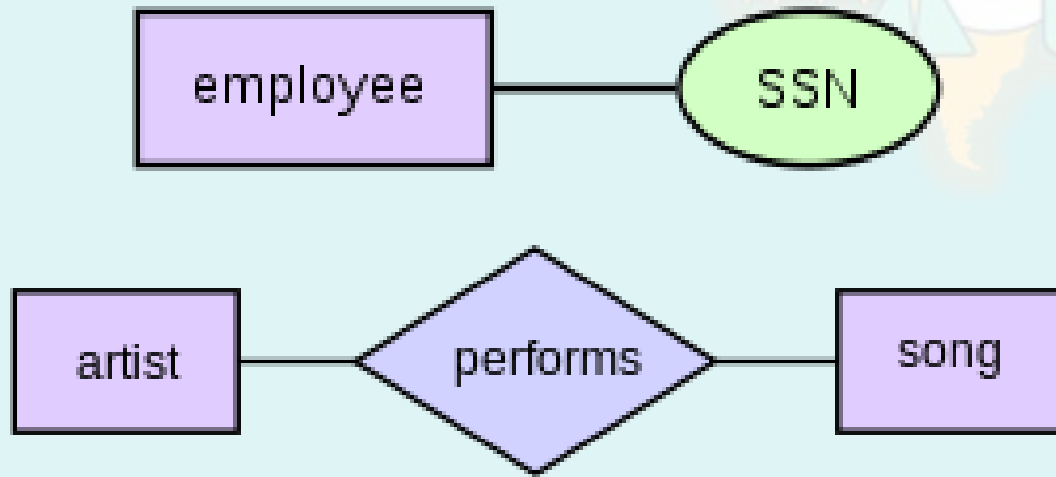
- Database Administrators.
 - Database Designers.
 - End Users / Database Developer.
-
- System Analysts and Application Programmers (Software Engineers)



The ER model



- the conceptual view of a database. It works around real-world entities and the associations among them.



Entity & Attributes



Entity

- An entity can be a real-world object
- Have/Has Attributes or properties.

Student

Attributes

- Entities are represented by means of their properties, called attributes. All attributes have values.
- For example, a student entity may have name, class, and age as attributes.

username



Types of Attributes



- **Simple attribute** – Simple attributes are atomic values, which cannot be divided further. For example, a student's phone number is an atomic value of 10 digits.
- **Composite attribute** – Composite attributes are made of more than one simple attribute. For example, a student's complete name may have first_name and last_name.
- **Derived attribute** – Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database. For example, average_salary in a department should not be saved directly in the database, instead it can be derived. For another example, age can be derived from data_of_birth.
- **Single-value attribute** – Single-value attributes contain single value. For example – Social_Security_Number.
- **Multi-value attribute** – Multi-value attributes may contain more than one values. For example, a person can have more than one phone number, email_address, etc.



What is an Entity Relationship Diagram (ERD)



- type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.
- ERD symbols and notations (Chen)



Entity Set and Keys



- Key is an attribute or collection of attributes that uniquely identifies an entity among entity set.
- For example, the roll_number of a student makes him/her identifiable among students.
 - **Candidate Key** : A minimal super key is called a candidate key. An entity set may have more than one candidate key.
 - **Primary Key** : A primary key is one of the candidate keys chosen by the database designer to uniquely identify the entity set.
 - **Super Key** : A set of attributes (one or more) that collectively identifies an entity in an entity set.
 - **Foreign key** : Identifies the relationship between entities.

