Database Fundamentals & Design

Presented by:

Abdelrahman Eid

Agenda

- Data & Information
- File Based System
- What is Database, Database System?
- DBMS Definition & Functions
- Database Properties
- Advantages and Disadvantages of Database Systems
- DB Architecture
- Who Deals with Database
- Data Models
- What is a Relational Database?
- Basic Database Structure
- Entity Relationship Modelling
- ERD Notation

Data & Information

- Data is the raw input (numbers, characters, images…) which when processed or arranged makes meaningful output (Information)

- Data is the lowest level of knowledge and information is the second level.

- Data by itself alone is not significant. Information is significant by itself.

- Observations and recordings are done to obtain data, while analysis and processing are done to obtain information.

File Based System

- It is a collection of programs that perform services for the end user.
- Each Program defines and manages its own data.
- Limitations:
  - ✓ Isolation of data
  - ✓ Duplication of data
  - ✓ Program Data Dependence
  - ✓ Incompatible File Formats

What is a database?

- “A database is an organized collection of **related data**.”

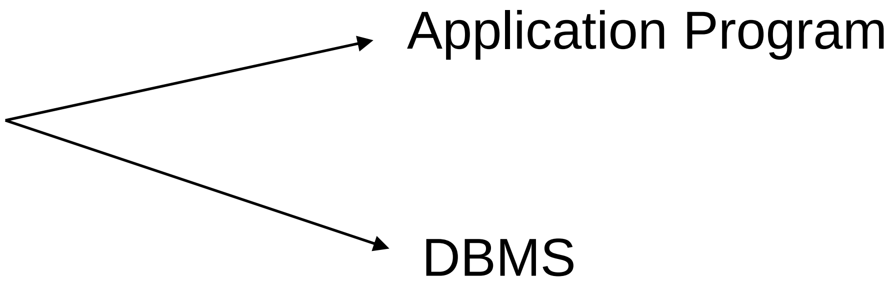
- The data is typically organized to model relevant aspects of reality in a way that supports processes requiring this information

Database System

- A database system is composed of :

✓ The Database.

✓ The Software.



Application Program

DBMS

Database Management System (DBMS)

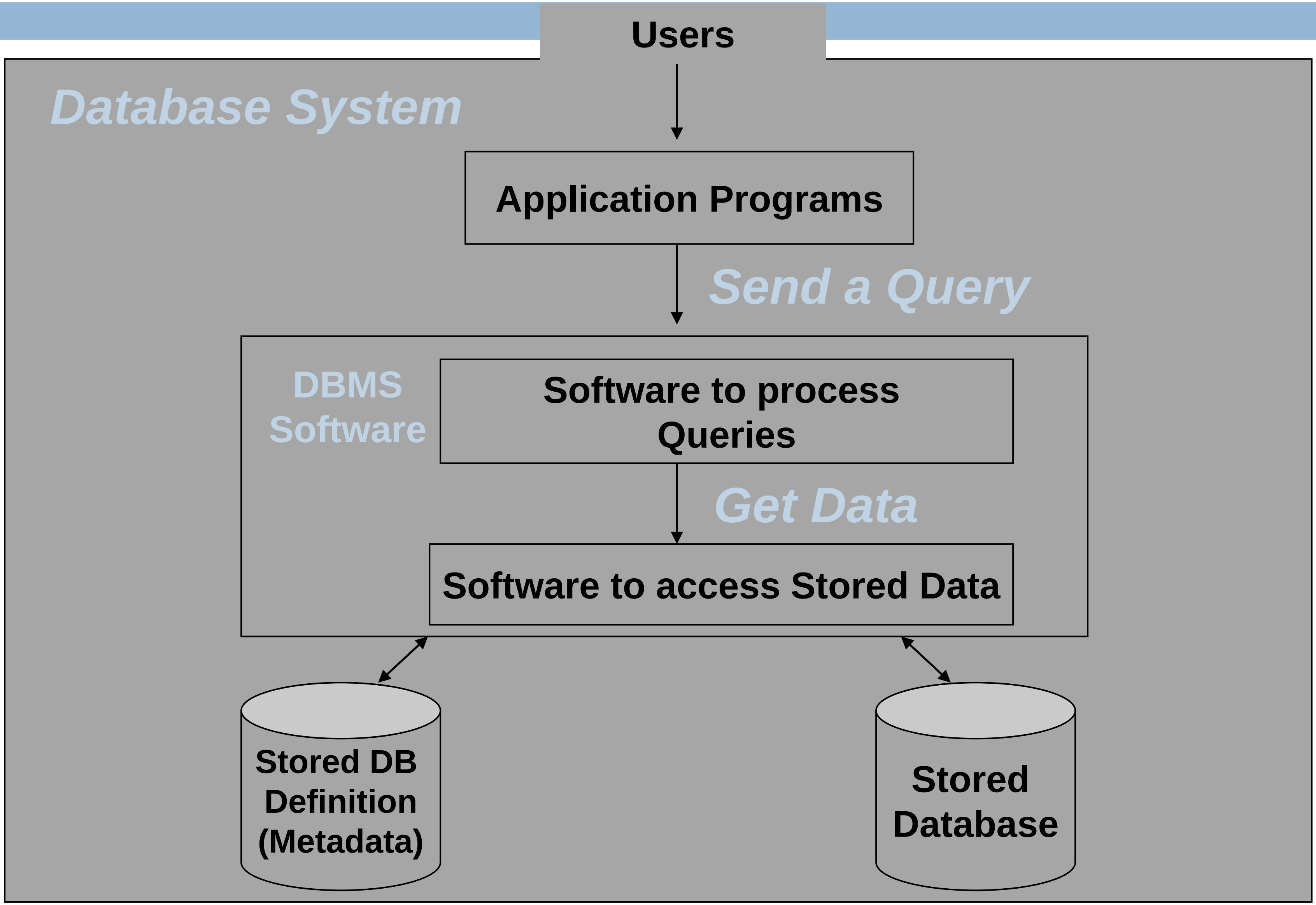
- It is the intermediate layer between the database and the programs that access the data.

- It is collection of programs that enables users to create and maintain a database.

DBMS Functions

- Defining Database.
- Constructing Database.
- Manipulating Database.
- Data Independence.
- Data Security & Integrity.
- Concurrency.
- Backup & Recovery.
- Data Dictionary (Meta Data).
- Performance.

Database System (Cont.)


Users
Database System
Application Programs
Send a Query
DBMS Software
Software to process Queries
Get Data
Software to access Stored Data
Stored DB Definition (Metadata)
Stored Database

Database Properties

- Self-describing nature.
- Insulation between program and data.
- Sharing of data and multi-user transaction processing.

Advantages of Database

- Redundancy can be reduced.
- Inconsistency can be avoided.
- Data can be shared.
- Security restrictions can be applied.
- Enforcing Integrity Constraints.
- Providing Backup and Recovery.

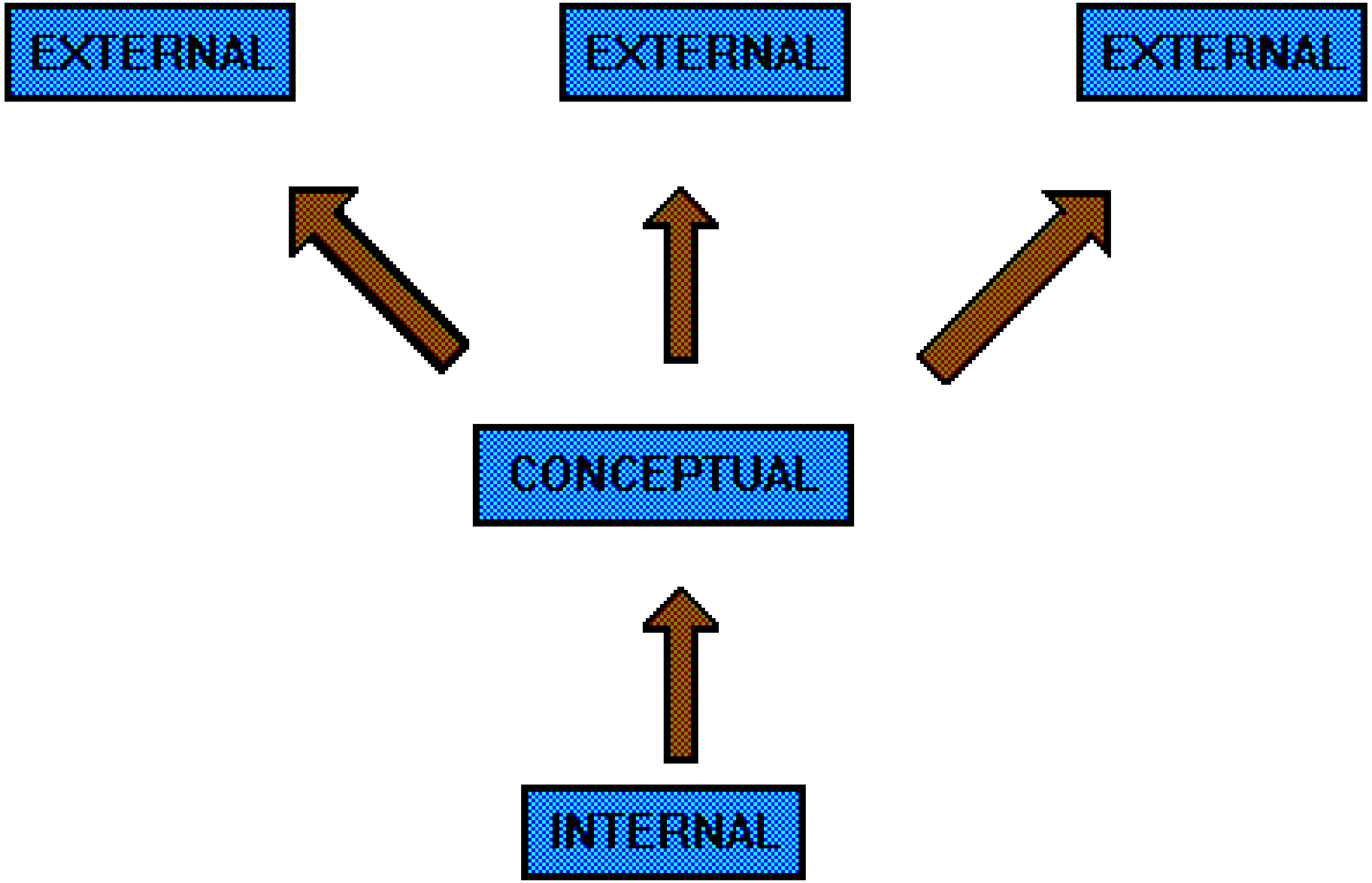
Disadvantages of Database

- It needs expertise to use (which is expensive).
- DBMS itself is expensive.
- DBMS may be incompatible with any other available DBMS.

Database Architecture

- The Three Levels of the Architecture.
- Advantages of the Architecture.
- How does the Application Program Access the Database.

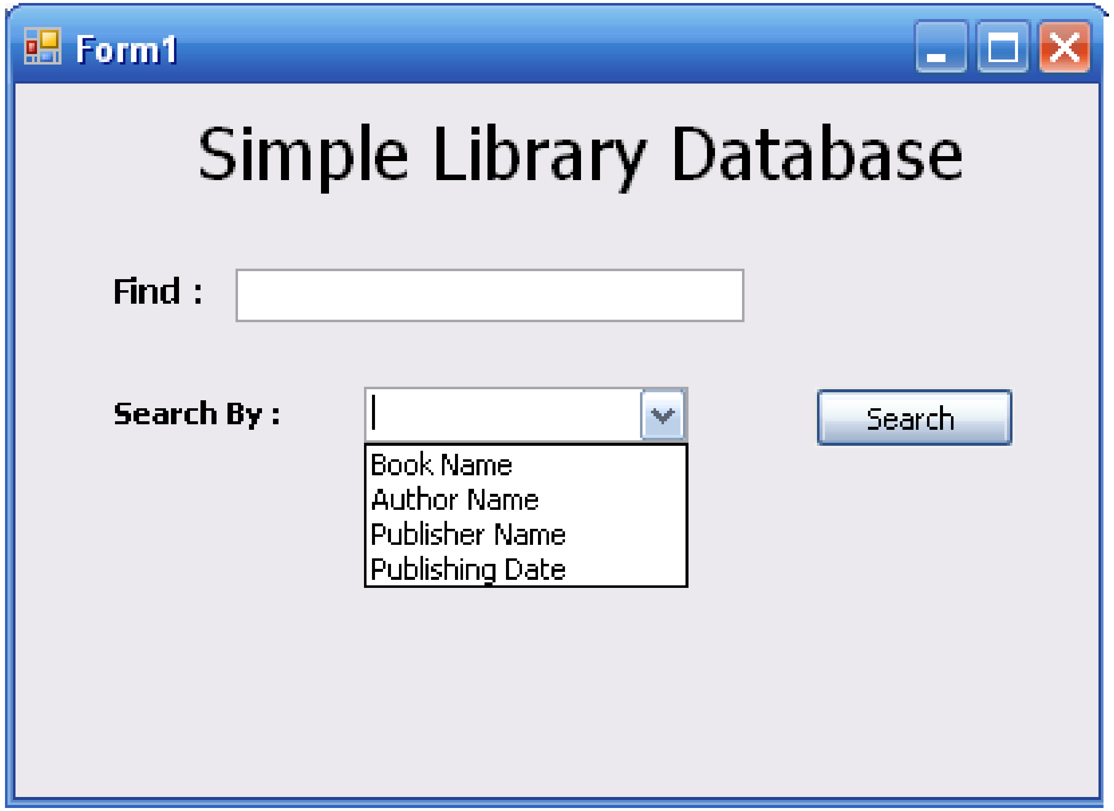
Three Level/Schema Architecture



Three Level/Schema Architecture (cont.)

- **External – What the user sees:**

  Deals with methods through which users may access the schema, such as through the use of a data input form. The external model allows relationships to be created between the user application and the data model.



Three Level/Schema Architecture (cont.)

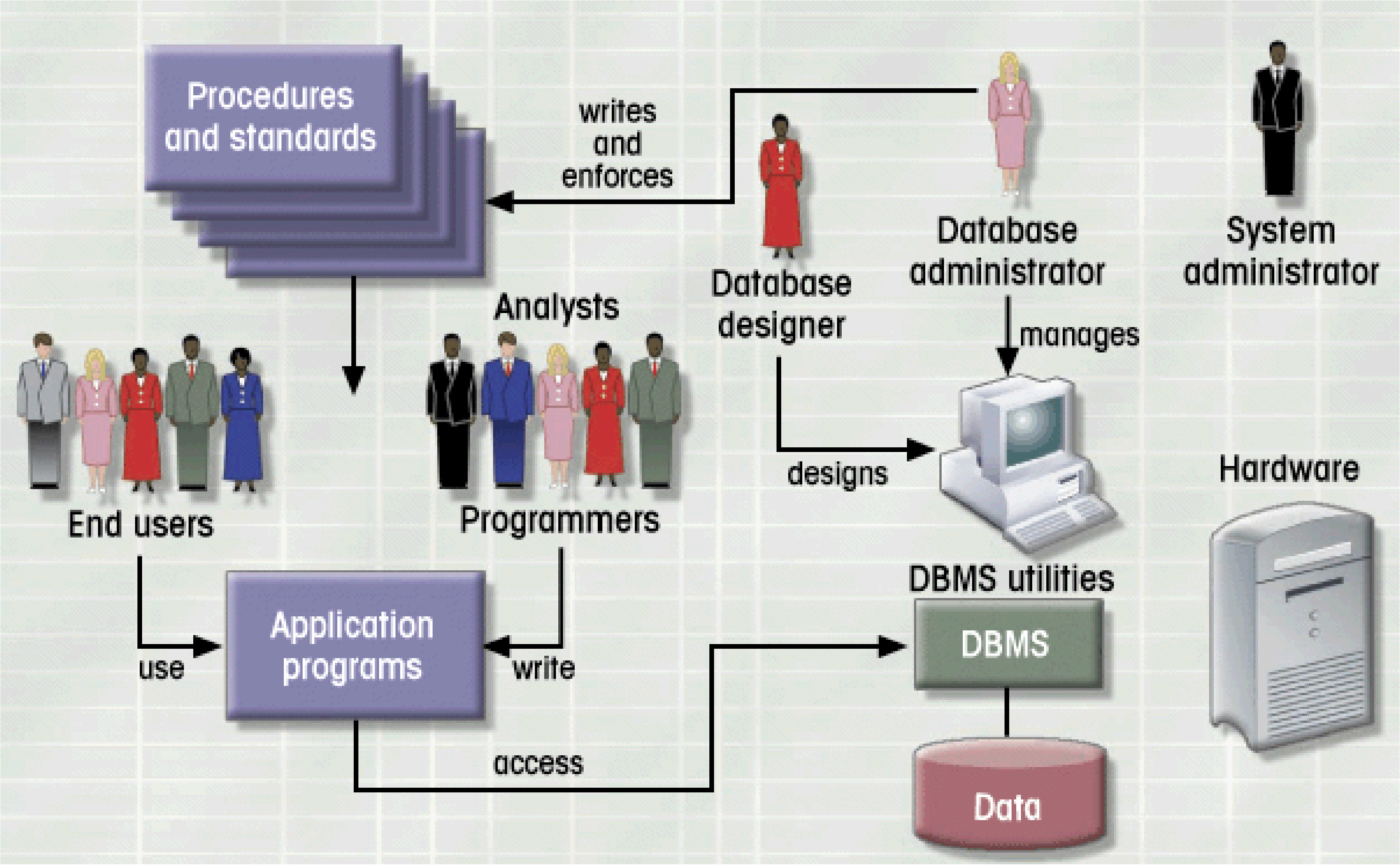
- **Conceptual – The logical model:**

  It is the basic database model, which deals with organizational structures that are used to define database structures such as tables and constraints.

- **Internal – The physical model:**

  Deals with the physical storage of the database, as well as access to the data.

Who deals with a database?



Data Models

A model is a pictorial representation of reality.

Data modeling is a technique for organizing and documenting a system's data, a collection of concepts that can be used to describe the structure and logical organization of the database.

- High Level or Conceptual data models provide concepts that are close to the way many users perceive data, entities, attributes and relationships. (Ex. ERD)
- Physical data models describes how data is stored in the computer and the access path needed to access and search for data.

Database Model

- Relational
- Network
- Hierarchical

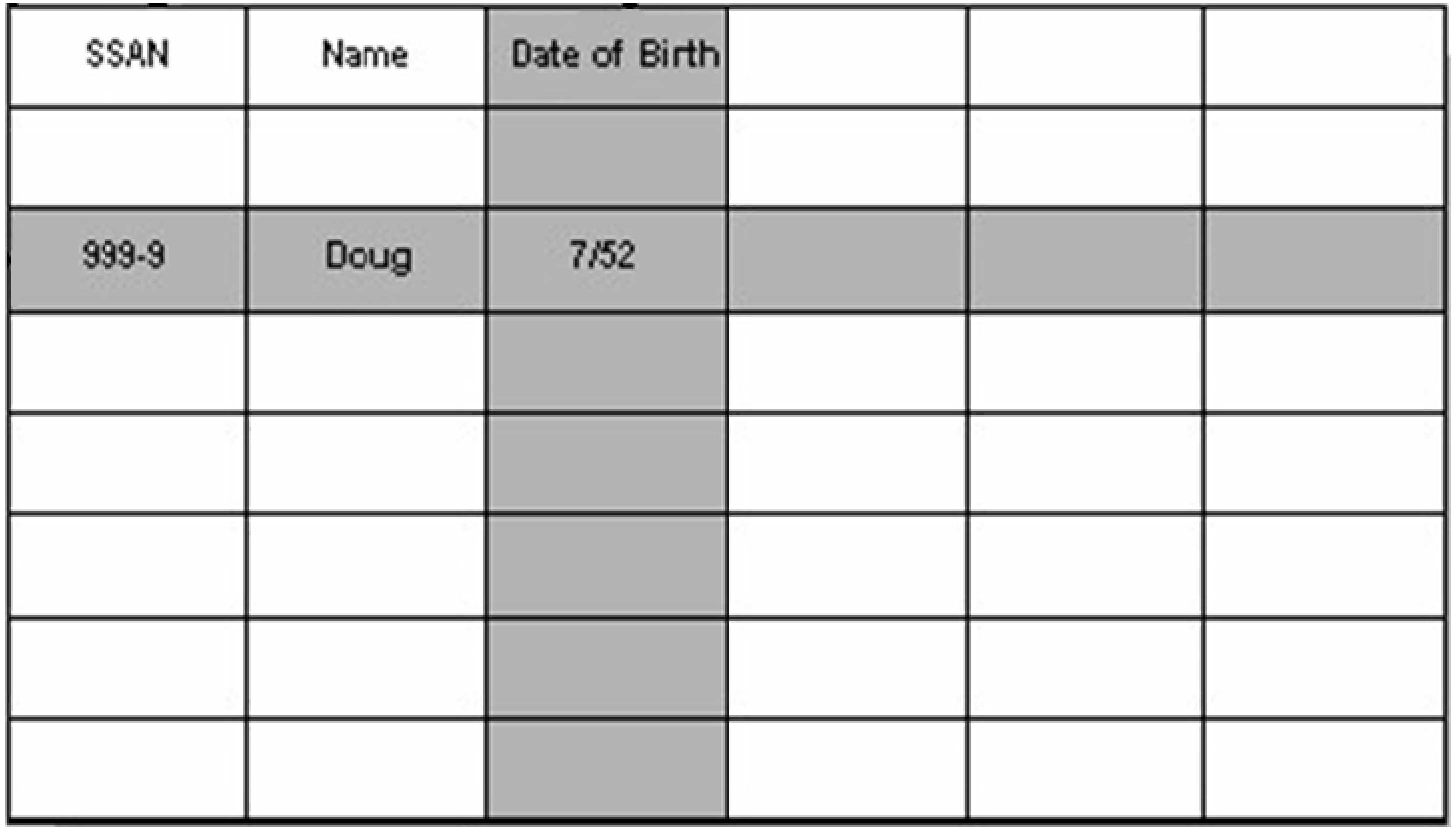
What is a Relational Database?

- A data structure through which data is stored in tables that are related to one another in some way.
- The way the tables are related is described through a **relationship**.

Basic Database Structure

Relation

Column



SSAN	Name	Date of Birth			
999-9	Doug	7/52			

Record

Entity Relationship Modeling

Entity-Relationship Diagram (ERD):

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

Definitions

- **Entity** :

  It is any thing about which data is collected (any thing a user want to track) .

- **Weak Entity** :

  It is an entity whose existence is dependent on another entity.

- **Entity Instance** :

  An instance is a particular occurrence of an entity. For example, each person is an instance of an entity, each car is an instance of an entity, etc.

Definitions (Cont.)

Attributes:

They are the characteristics of entities.

Types of Attributes:

- **Simple (Scalars)** - smallest semantic unit of data.
- **Composite** - group of attributes e.g., address (street, city, state, zip)
- **Multi-valued (list)** - multiple values e.g., phone numbers.
- **Stored or Derived.**
- **Complex**

Attribute Values:

- ✓ Sometimes attribute values is set to null.
- ✓ Default Value.

Definitions (Cont.)

- **Primary Key**:

  Identifier used to uniquely identify one particular instance of an entity.

  - ✓ Can be one or more attributes.
  - ✓ Must be unique .
  - ✓ Value should not change over time.
  - ✓ Must always have a value .

Definitions (Cont.)

- **Candidate Key**:

  When multiple possible identifiers exist, each is a candidate key.

- **Foreign Keys:**

  Foreign keys reference a related table through the primary key of that related table.

- **Referential Integrity Constraint**:

  For every value of a foreign key there is a primary key with that value in the referenced table e.g., if student name is to be used in a dormitory table, then that name must exist in the student table.

Relationships

- **Relationships** - A relationship is a connection between entity classes.

1. **Degree of a Relationship**:

   Is the number of participating entities.

2. **Cardinality Ratio**:

   Specifies the maximum number of relationship (type of relation).

3. **Participation**:

   Specifies the minimum number of relationship instances that each entity can participate with.

Degrees of Relationships

- Number of entity types that participate in a relationship
- Three cases
  - Unary: between two instances of one entity type
  - Binary: between the instances of two entity types
  - Ternary: among the instances of three entity types

Cardinality

Cardinality specifies the maximum number of relationship.

- **Types of Relationships (Cardinality) :**
  - One-to-one relationship (1:1)
  - One-to-many relationship (1:M)
  - Many-to-many relationship (N:M)

Cardinality (Cont.)

- **One-to-one relationship (1:1):**

A single record in table A is related to only one record in table B, and vice versa.

Ex. : Emp. Uses at most one car, a car is used at most by one emp.
Manager – Department.

Cardinality (Cont.)

- **One-to-many relationship (1:M):**

A single record in table (A) can be related to one or more records in table (B), but a single record in table (B) can be related to only one record in table (A).

Ex. **:** Emp. Uses at most one car, a car is used by many or several employees, Student - Advisor, Customer - Order

Cardinality (Cont.)

- **Many-to-many relationship (M:M):**

A single record in table A can be related to one or more records in table B, and vice versa.

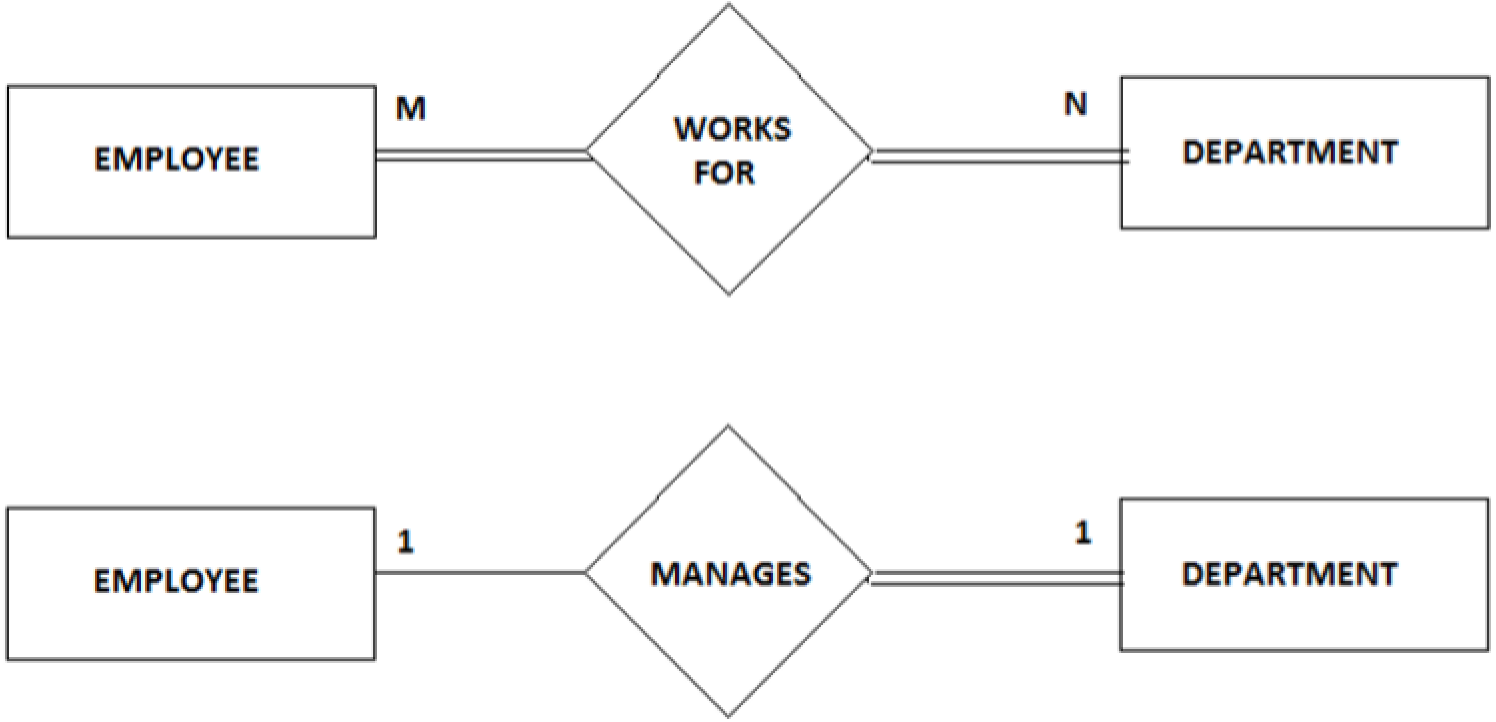
Ex. An emp. Uses several cars, a car can be used by several employees. Student - Club, Order - Products.

Participation in Relationships

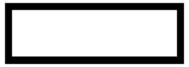
In a Relationship, Participation constraint specifies the existence of an entity when it is related to another entity in a relationship type. It is also called minimum cardinality constraint.

This constraint specifies the number of instances of an entity that can participate in a relationship type.

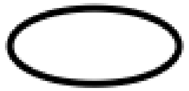
Participation in Relationships (Cont.)



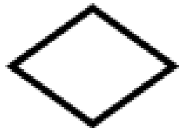
ERD Notation



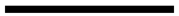
Represents Entity



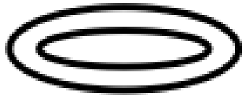
Represents Attribute



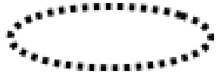
Represents Relationship



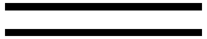
Links Attribute(s) to entity set(s) or
Entity set(s) to Relationship set(s)



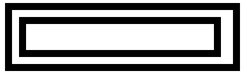
Represents Multivalued Attributes



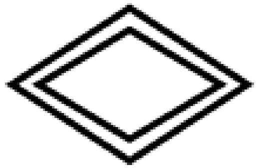
Represents Derived Attributes



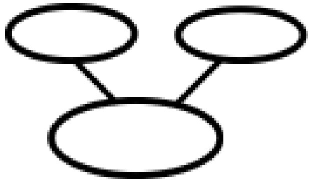
Represents Total Participation of Entity



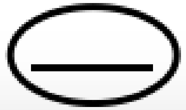
Represents Weak Entity



Represents Weak Relationships

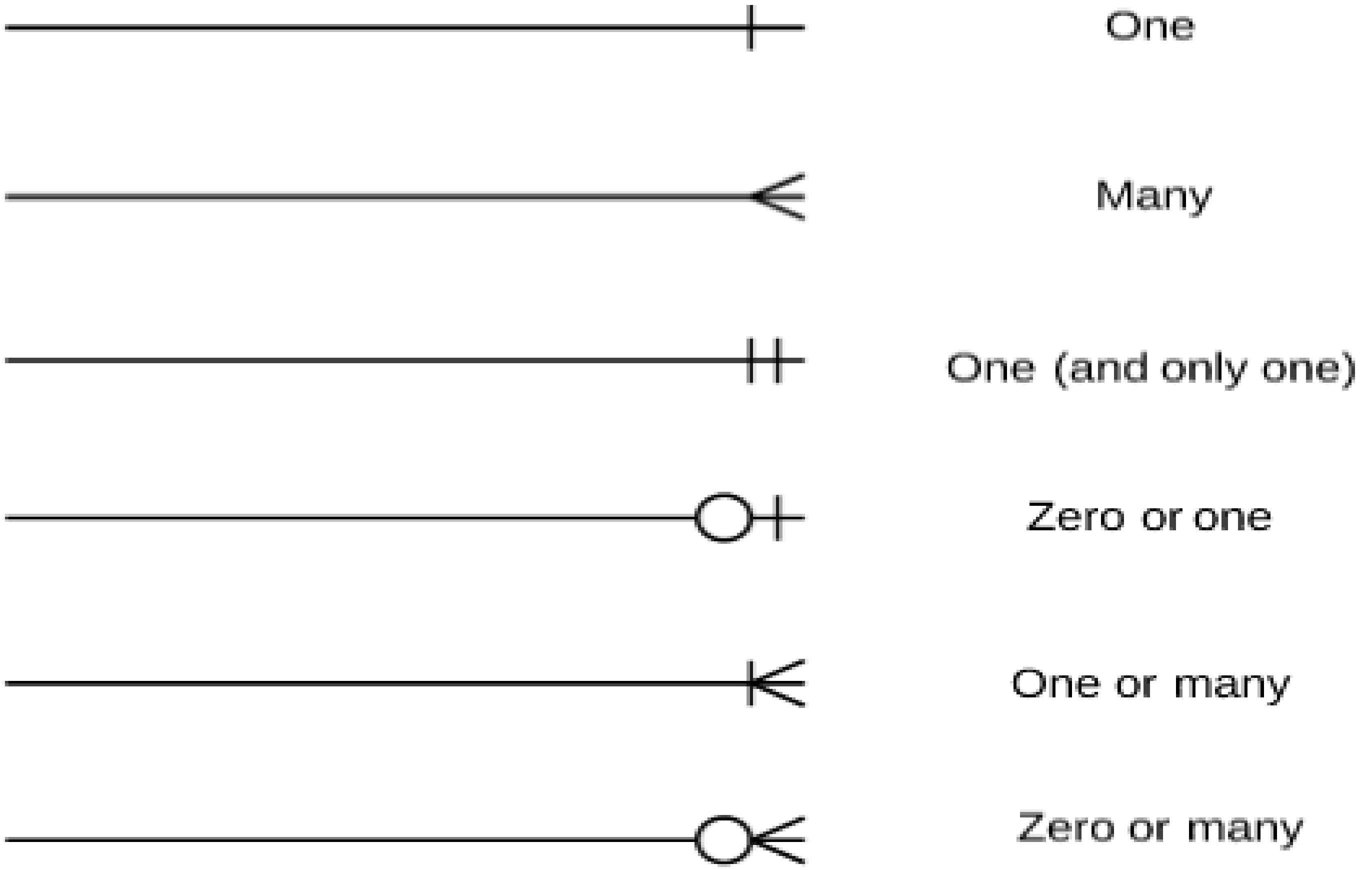


Represents Composite Attributes



Represents Key Attributes / Single Valued
Attributes

Cardinality and Participation



One

Many

One (and only one)

Zero or one

One or many

Zero or many

Guidelines

- When building a data model, a number of questions must be addressed:
  - What entities need to be described in the model?
  - What characteristics or attributes of those entities need to be recorded?
  - Can an attribute or a set of attributes be an identifier that will uniquely identify one specific occurrence of an entity?
  - What associations or relationships exist between entities?

Summary

- Data & Information
- File Based System
- What is Database, Database System?
- DBMS Definition & Functions
- Database Properties
- Advantages and Disadvantages of Database Systems
- DB Architecture
- Who Deals with Database
- Data Models
- What is a Relational Database?
- Basic Database Structure
- Entity Relationship Modelling
- ERD Notation

THANK YOU