6. DATABASE SYSTEMS

Lecturer:
Elvira Aitmukhanbetova

Information and Communications Technologies
Elvira.aitmukhanbetova@astanait.edu.kz

ASTANA IT UNIVERSITY

6.1: Introduction to Database Systems

6.1: Introduction to Database Systems

6.2: Data Models

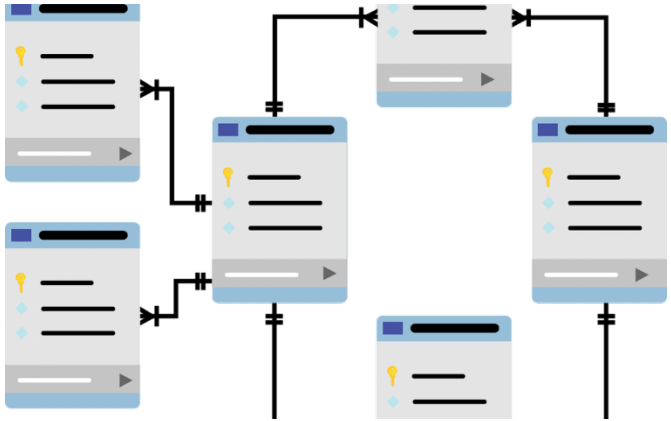
6.3: ER Diagram Representation

Learning Objectives

- Define database, DBMS and database systems
- Describe the database importance, its functions, advantages and disadvantages
- List database characteristics
- Understand business rules
- Define database users

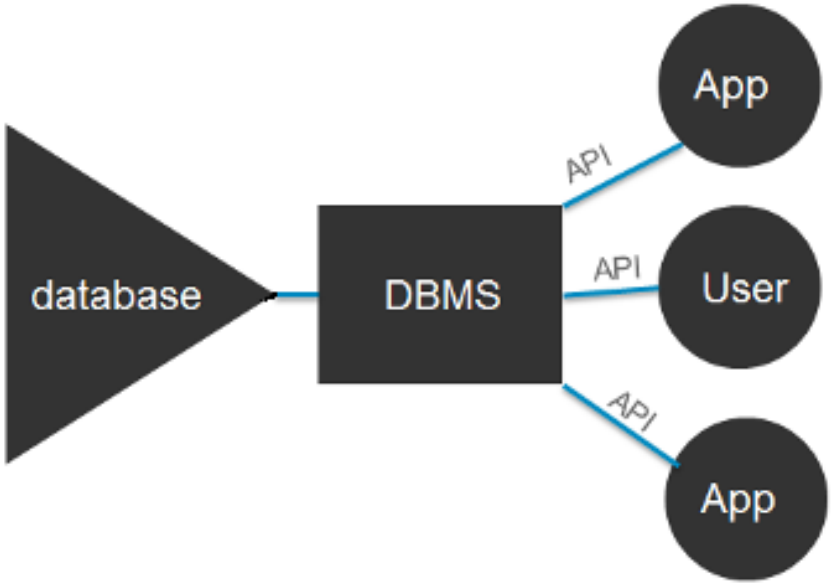
Introduction

- Database
  - a collection of **related data** and its metadata organized in a **structured** format for optimized information management



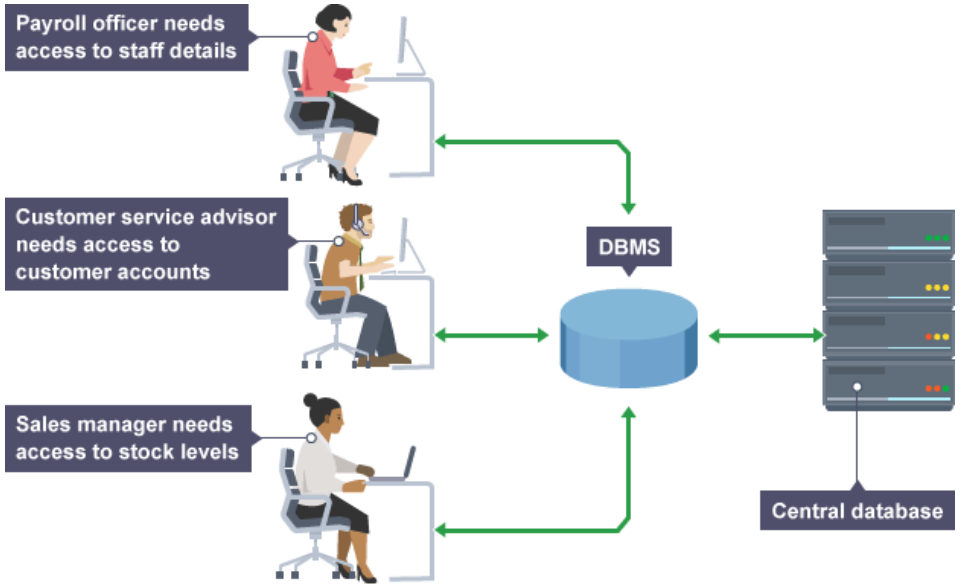
Introduction (2)

- Database Management System (DBMS)
  - is software that enables the easy creation, access, and modification of databases for efficient and effective database management



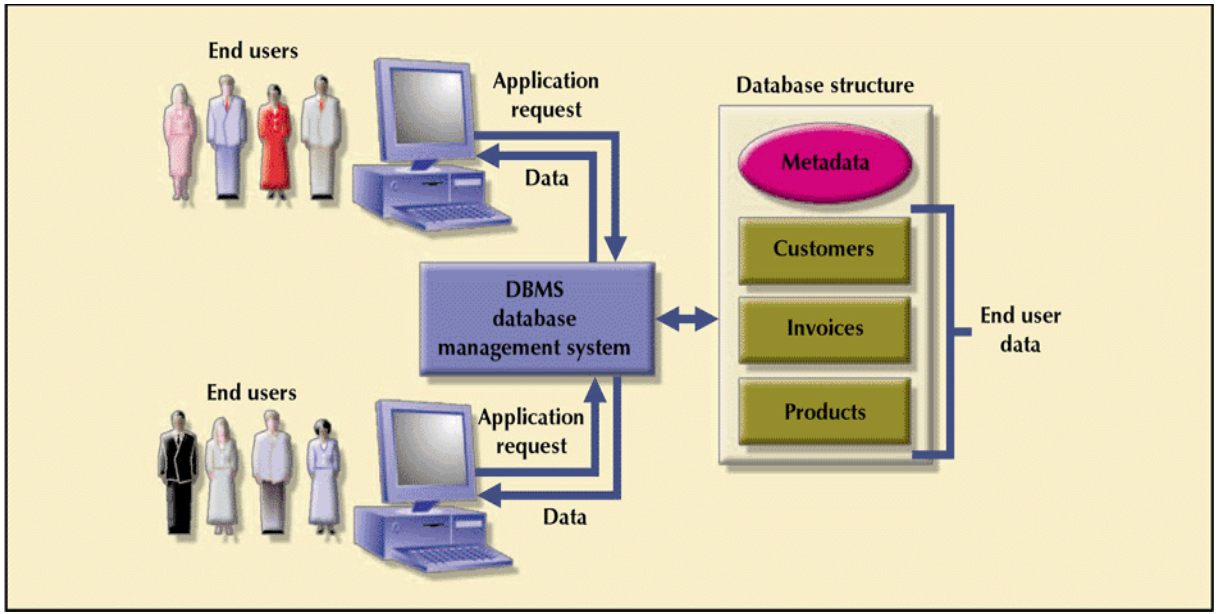
Introduction (3)

- Database System
  - is an **integrated system** of hardware, software, people, procedures, and data **that define** and regulate the collection, storage, management, and use of data within **a database environment**



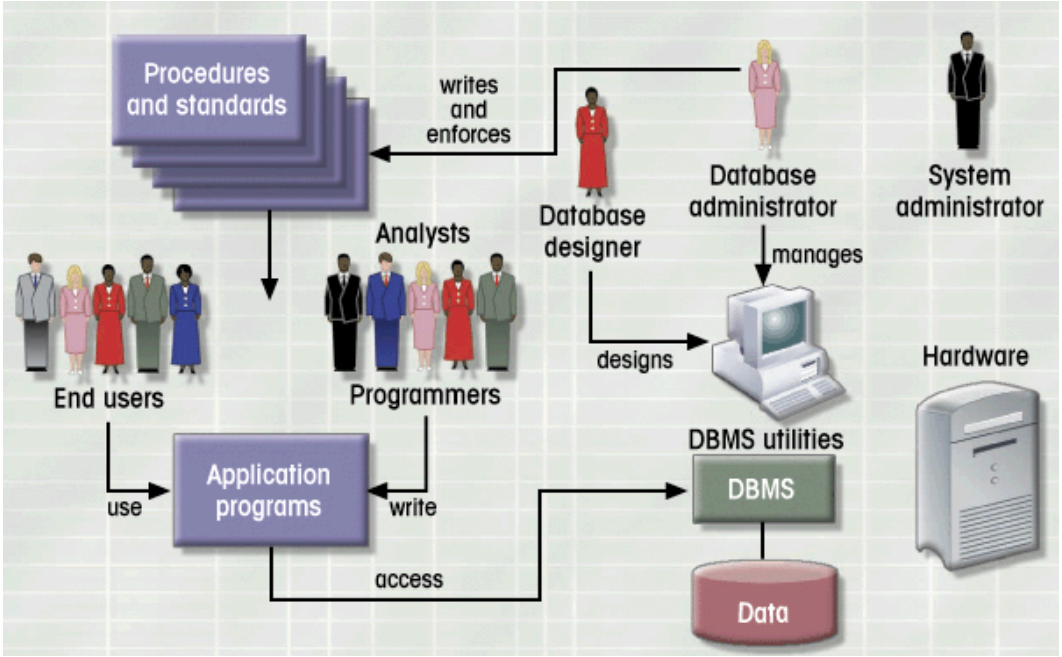
Database Management System (DBMS)

- Manages **interaction** between end users and database



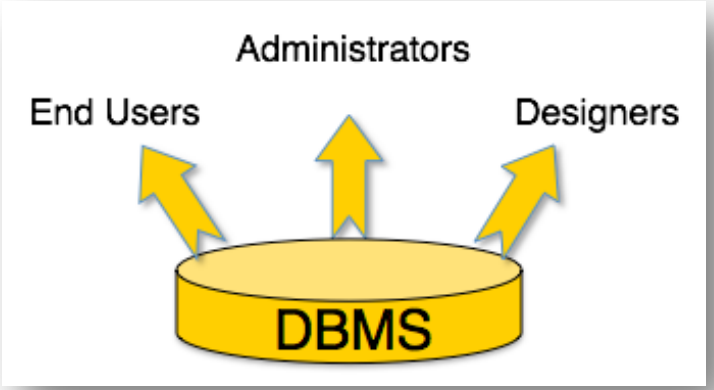
Database System Environment

- Hardware
- Software
  - OS
  - DBMS
  - Applications
- People
- Procedures
- Data



Users

- Administrators
  - maintain the DBMS and are responsible for administrating the database
- Designers
  - the group of people who actually work on the designing part of the database
- End Users
  - are those who actually reap the benefits of having a DBMS



Database: Importance

- Purpose of databases
  - Optimizes data management
  - Transforms data into information

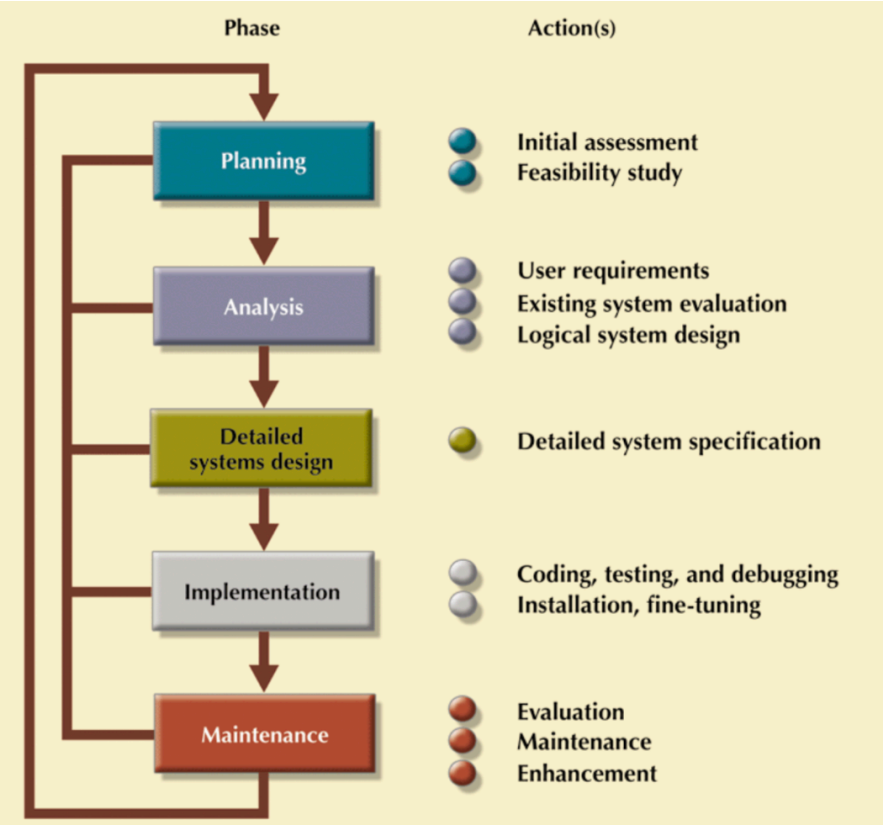
Database: Importance (2)

- Importance of Database Design
  - Defines the database's expected use
  - Avoid data redundancy & ensure data integrity
  - Poorly designed database generates errors

Database: Functions

- Functions of DBMS/Database System
  - Stores data and related data entry forms, report definitions, etc.
  - Hides the complexities of relational database model from the user
  - Enforces data integrity
  - Implements data security management
  - Provides backup and data recovery

Database Development Life Cycle



Business Rules

- **Definition:** Brief, precise, and unambiguous descriptions of operations in an organization
  - based on policies, procedures, or principles within a specific organization
  - help to create and enforce actions within that organization's environment
  - apply to any organization that stores and uses data to generate information

Business Rules (3)

- Sources
  - Interviews
    - Company managers
    - Policymakers
    - Department managers
    - End users
  - Written documentation
    - Procedures, Standards, Operations manuals
  - Observation
    - Business operations

Business Rules (2)

- Purposes
  - Enhance understanding & facilitate communication
    - Standardize company's view of data
    - Constitute a communications tool between users and designers
    - Allow designer to understand business process as well as the nature, role, and scope of data
  - Promote creation of an accurate data model

Database Characteristics

- 1) Real-world entity
  - A modern DBMS is more realistic and uses real-world entities to design its architecture
- 2) Relation-based tables
  - DBMS allows entities and relations among them to form tables

Database Characteristics (2)

- 3) Isolation of data and application
  - A database system is entirely different than its data
  - A database is an *active entity*, whereas data *is passive* (allowing change without resistance)
- 4) Less redundancy
  - DBMS follows the rules of normalization, which splits a relation when any of its attributes is having redundancy in values

Database Characteristics (3)

- 5) Consistency
  - Consistency is a state where every relation in a database remains consistent

- 6) Query Language
  - DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data

Database Characteristics (4)

- 7) ACID Properties
  - DBMS follows the concepts of Atomicity, Consistency, Isolation, and Durability (ACID)
  - These concepts are applied to transactions, which manipulate data in a database
  - ACID properties help the database stay healthy in multi-transactional environments and help prevent failure

Database Characteristics (5)

- 8) Multi User and Concurrent Access
  - DBMS supports multi-user environment and allows them to access and manipulate data in parallel
- 9) Multiple views
  - DBMS offers multiple views for different users

Database Characteristics (6)

- 10) Security
  - Features like multiple views offer security to some extent where users are unable to access data of other users and departments
  - DBMS offers methods to impose constraints while entering data into the database and retrieving the same at a later stage

Advantages of DBMS

- Improved data sharing
- Improved data security
- Minimized data inconsistency
- Improved data access
- Improved decision making

Disadvantages of DBMS

- Increased costs
- Management complexity
- Maintaining currency
- Frequent upgrade/replacement cycles

Introduction to Database Systems **Summary**

- **Database System**
  - is an integrated system of hardware, software, people, procedures, and data
  - that define and regulate the collection, storage, management, and use of data within a database environment
- **Business rules**
  - Brief, precise, and unambiguous descriptions of operations in an organization
- **Users**: administrators, end users, and designers
- **Advantages of DBMS**: improved data sharing, improved data security, minimized data inconsistency , improved data access, improved decision making
- **Disadvantages of DBMS: i**ncreased costs, management complexity, maintaining currency, frequent upgrade/replacement cycles

6.2: Data Models

6.1: Introduction to Database Systems

6.2: Data Models

6.3: ER Diagram Representation

Learning Objectives

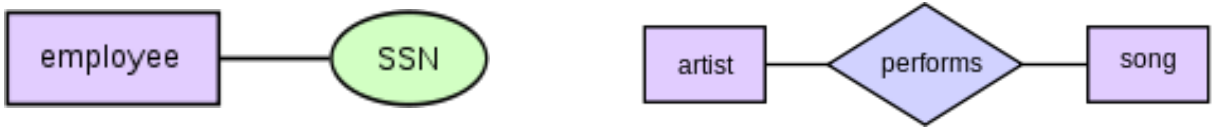
- Define data models
- Describe Entity-Relationship model
- Describe Relational model

Data Model

- **Data models** define how the logical structure of a database is modeled
  - Are fundamental entities to introduce abstraction in a DBMS
  - Define how data is connected to each other and how they are processed and stored in the system.

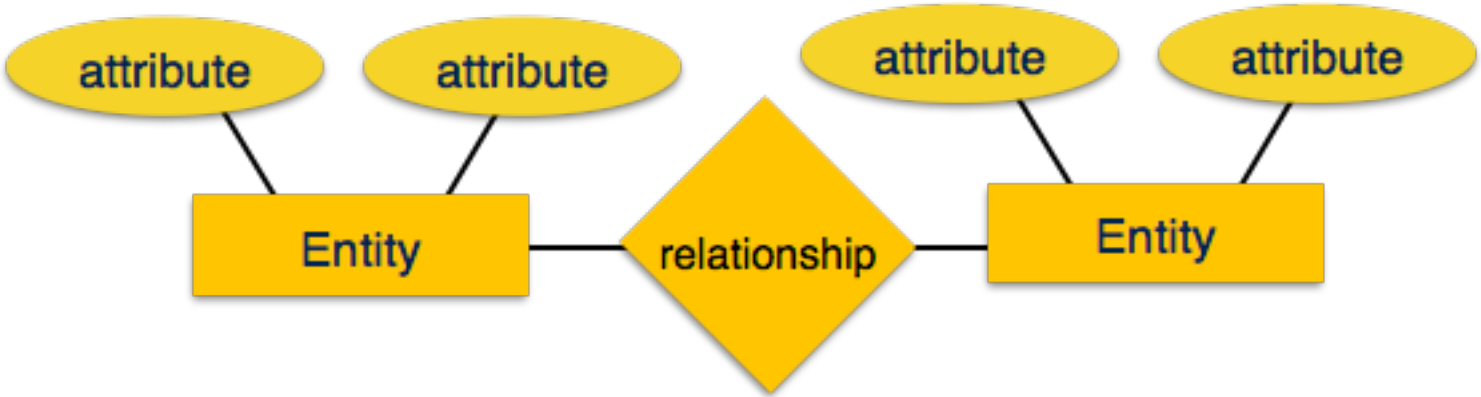
Data Models: Entity-Relationship Model

- **Entity-Relationship (ER) Model** is based on the notion of *real-world entities* and relationships among them
  - While formulating real-world scenario into the database model, the ER Model creates an *entity set*, a *relationship* set, *general attributes*, and *constraints*



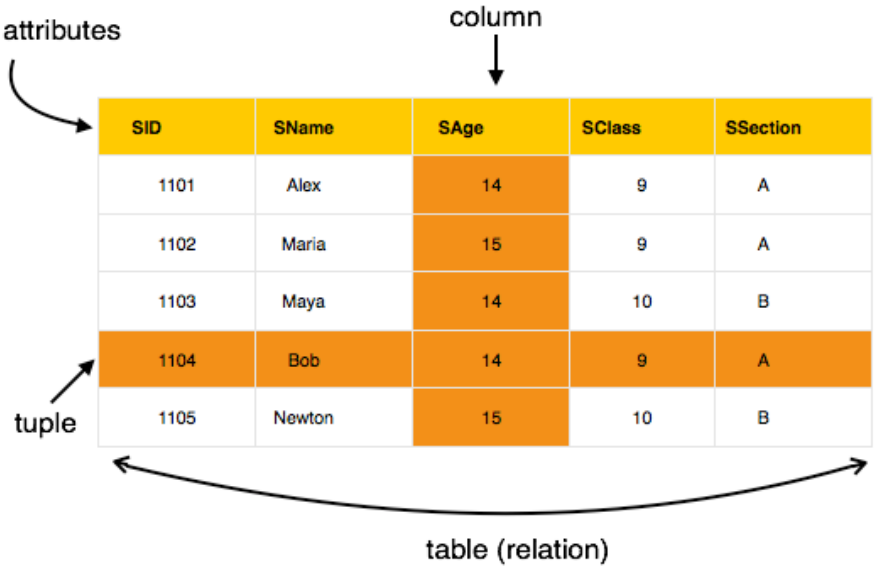
Data Models: Entity-Relationship Model (2)

- ER Model is based on
  - **Entities** and their *attributes*
  - **Relationships** among entities



Data Models: Relational Model

- The most popular data model in DBMS is the **Relational Model**
  - Based on first-order predicate logic and defines a table as an **n-ary relation**



Data Models: Relational Model (2)

- Data is stored in tables called **relations**
- Relations can be normalized
- In normalized relations, values saved are atomic values
- Each row in a relation contains a unique value
- Each column in a relation contains values from the same domain

Data Models: Relational Model (4)

STUDENT_ID	STUDENT_NAME	COLLEGE_YEAR	MAJOR
1	Madina	2019	Computer Science
2	Dimash	2018	Finance
3	Kayrat	2019	Media Technologies
4	Batyr	1998	Economics
5	Kamila	2000	IT Management
6	John	2015	Economics

Data Models: Relational Model (3)

student_id	name	age
1	Akon	17
2	Bkon	18
3	Ckon	17
4	Dkon	18

subject_id	name	teacher
1	Java	Mr. J
2	C++	Miss C
3	C#	Mr. C Hash
4	Php	Mr. P H P

student_id	subject_id	marks
1	1	98
1	2	78
2	1	76
3	2	88

Entity-Set and Keys

- **Key** is an attribute or collection of attributes that uniquely identifies an entity among entity set.
- Types:
  - **Primary Key (PK)**
    - an attribute of the table that uniquely identifies every row in that table
    - value cannot be a NULL
    - function is to guarantee entity integrity
    - doesn't allow to appear the same value more than once
  - **Foreign key (FK)**
    - a field in the table whose values match the primary key of related table
    - may accept multiple null values
    - might be many foreign keys in a table

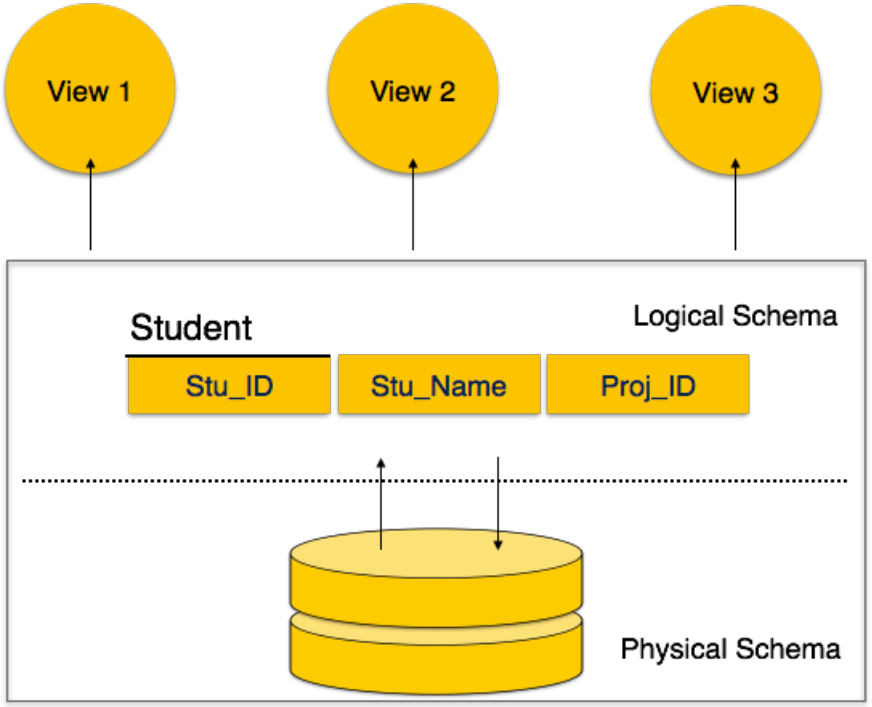
Database Schema

- Is the **skeleton structure** that represents the **logical** view of the entire database
- **Defines** its **entities** and the **relationship** among them

Database Schema (3)

- **Physical Database Schema**
  - Pertains to the actual storage of data and its form of storage like files, indices, etc.
  - It defines how the data will be stored in secondary storage
- **Logical Database Schema**
  - Defines all the logical constraints that need to be applied to the stored data
  - It defines tables, views, and integrity constraints

Database Schema (2)



Data Models **Summary**

- **Data models** define how the logical structure of a database is modeled.
- **Entity-Relationship (ER) model** is based on the notion of real-world entities and relationships among them.
- The **Relational model** is based on first-order predicate logic and defines a table as an **n-ary relation.**
- Database schema is the **skeleton structure** that represents the **logical** view of the entire database

6.3: ER Diagram Representation

6.1: Introduction to Database Systems

6.2: Data Models

6.3: ER Diagram Representation

Learning Objectives

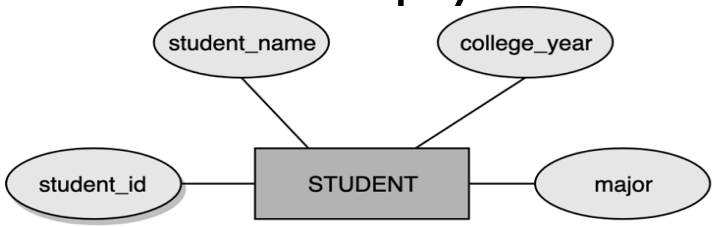
- Understand entity representation
- Describe attribute representation
- Describe relationship representation
- Understand generalization
- Define specialization
- Describe inheritance

ER Diagram Representation: Entity

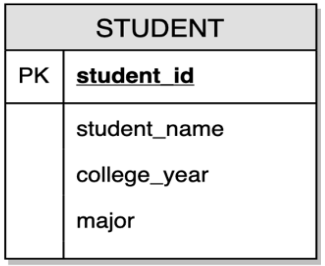
- **Entity** is a person, place, thing, or event about which data will be collected and stored.
- Entity is an object of interest to the end user
- Entities are represented using rectangles
- Rectangles are named with the entity set they represent

ER Diagram Representation: Attributes

- **Attributes** are the properties of entities
  - Every **attribute** is defined by its set of values called **domains**
- A **required** attribute is an attribute that must have a value, and it cannot be left empty
- An **optional** attribute is an attribute that does not require a value, it can be left empty.



Chen Model



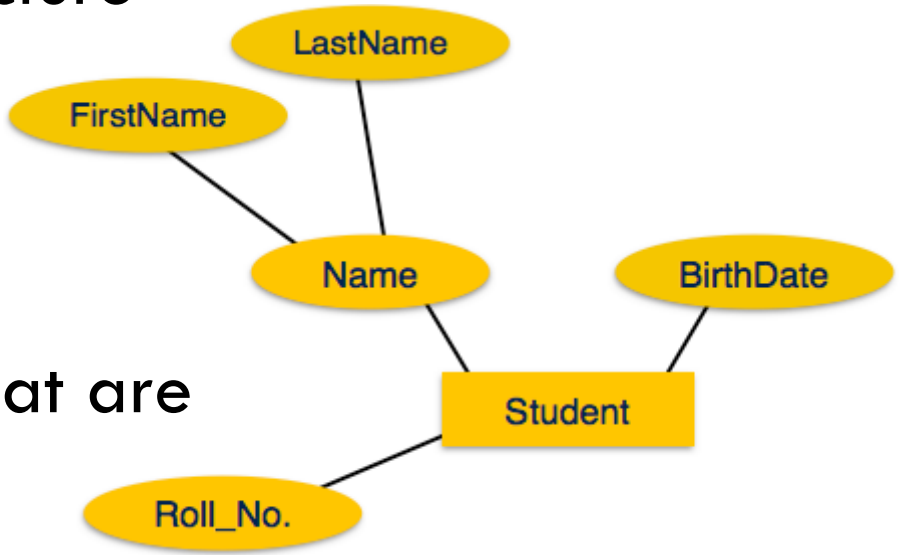
Crow's Foot Model

Types of Attributes

- Simple attribute
- Composite attribute
- Derived attribute
- Single-value attribute
- Multi-value attribute

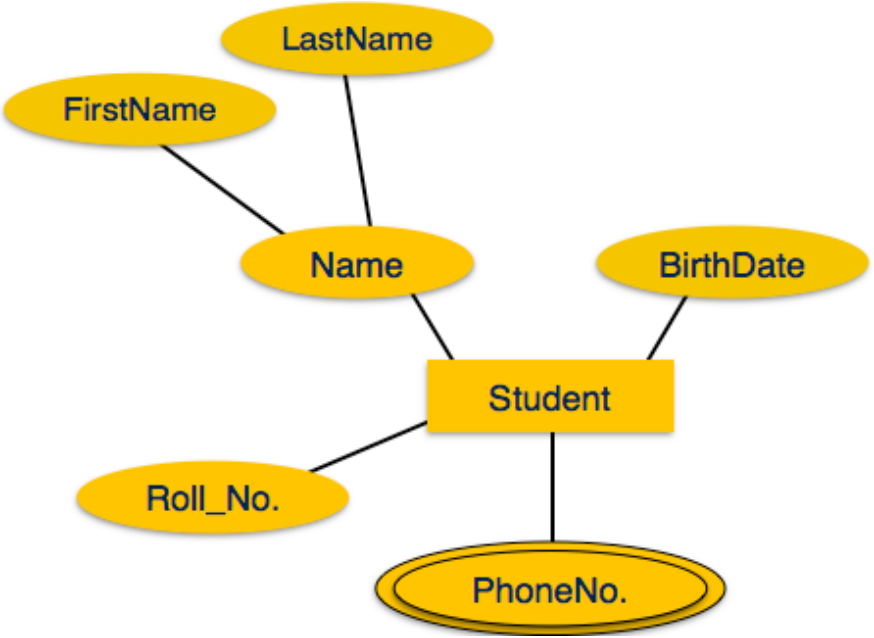
ER Diagram Representation: Attributes (2)

- If the attributes are **composite,** they are further divided in a tree-like structure
- Every node is then connected to its attribute
- Composite attributes are represented by ellipses that are connected with an ellipse



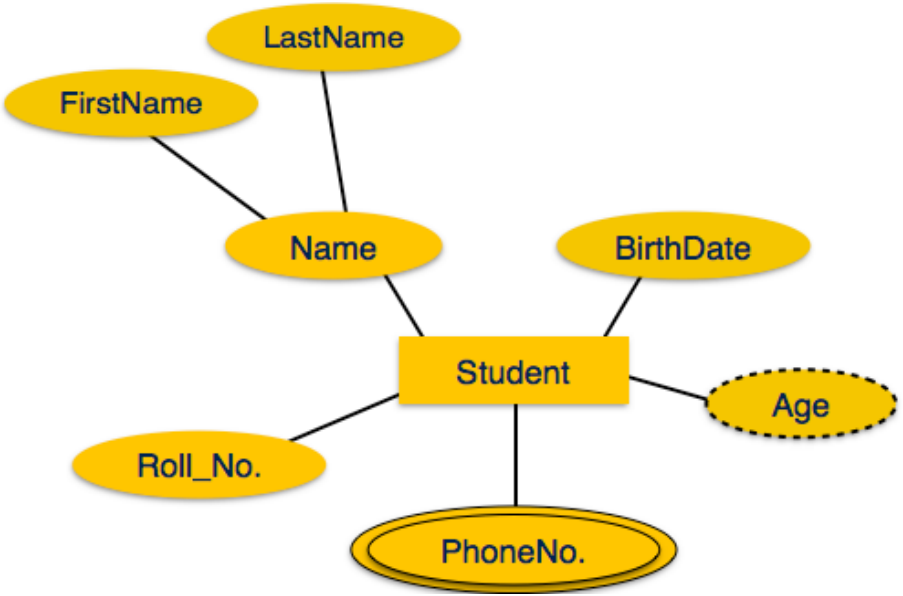
ER Diagram Representation: Attributes (3)

- **Multivalued** attributes are depicted by a double ellipse



ER Diagram Representation: Attributes (4)

- **Derived** attributes are depicted by a dashed ellipse

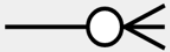
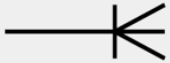
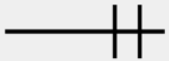
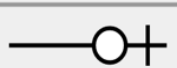


ER Diagram Representation: Relationship

- **Relationship** – the logical association among entities
  - Relationships are mapped with entities in various ways.
  - Mapping cardinalities define the number of association between two entities.
- In Chen model, relationships are represented by a diamond-shaped box
  - The name of the relationship is written inside the diamond-box
  - All the entities (rectangles) participating in a relationship, are connected to it by a line

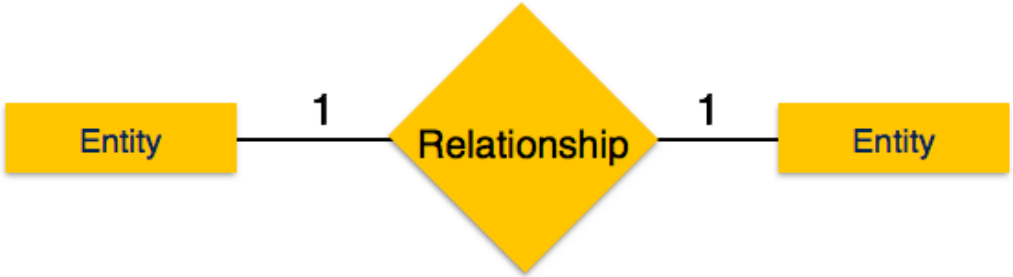
Relationship Cardinalities

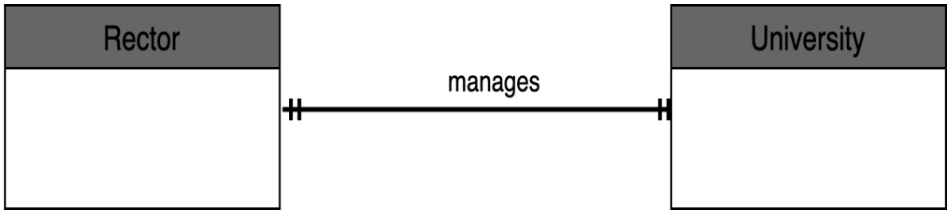
- **Cardinality**
  - the **number of instances** of an entity from a relation that can be associated with the relation

Crow's Foot Symbols	Cardinality	Comment
	(0, N)	Zero or many; the "many" side is optional
	(1, N)	One or many; the "many" side is mandatory
	(1, 1)	One and only one; the "1" side is mandatory
	(0, 1)	Zero or one; the "1" side is optional

Relationship Cardinalities (2)

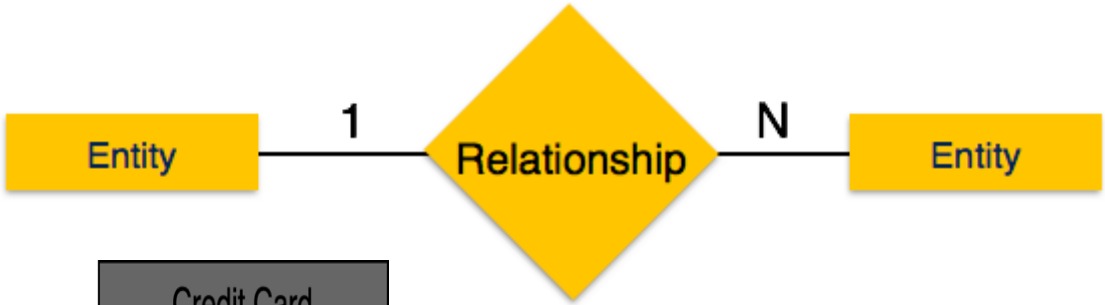
- **One-to-one** – when only one instance of an entity is associated with the relationship, it is marked as '**1:1**'.

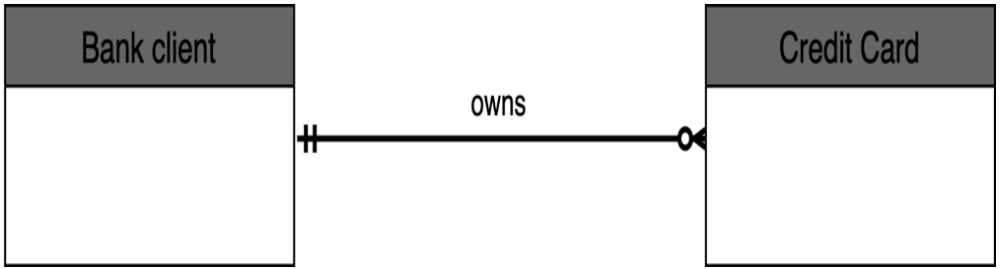




Relationship Cardinalities (3)

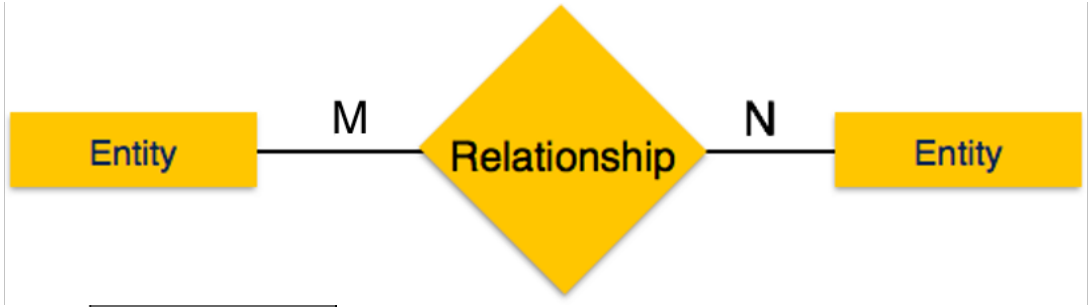
- **One-to-many** – When more than one instance of an entity is associated with a relationship, it is marked as '**1:N**'.

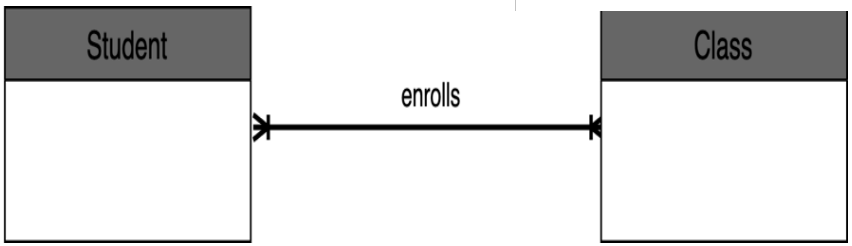




Relationship Cardinalities (4)

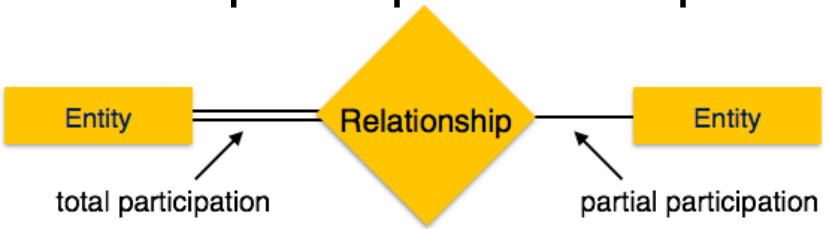
- **Many-to-many** – more than one instance of an entity on the left and more than one instance of an entity on the right can be associated with the relationship.





Participation Constraints

- **Total Participation**
  - Where each entity is involved in the relationship.
  - Total participation is represented by double lines.
- **Partial participation**
  - Not all entities are involved in the relationship.
  - Partial participation is represented by single lines.



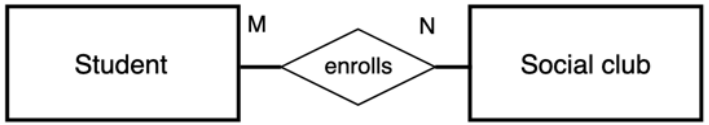
ER Model Notations

Chen Notation

Crow's Foot Notation

A **one-to-many** relationship. (1:M): A PAINTER paints many PAINTINGS, each PAINTING is painted by one PAINTER.

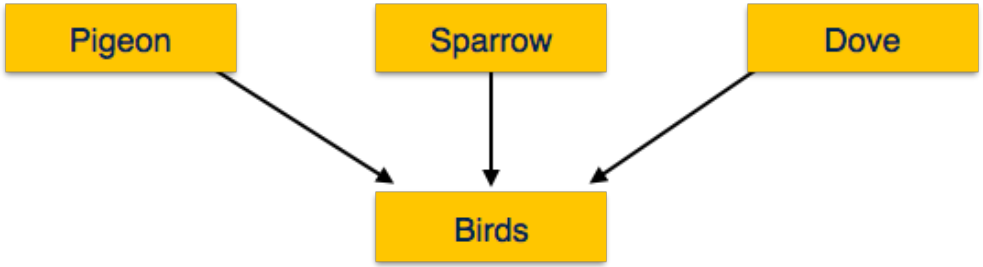
A **many-to-many** relationship. (M:N): A STUDENT can enroll to many SOCIAL CLUBS, each SOCIAL CLUB is enrolled by many STUDENTs.



A **one-to-one** relationship. (1:1): A CHAIRMAN manages one COMPANY, each COMPANY is managed by one CHAIRMAN.

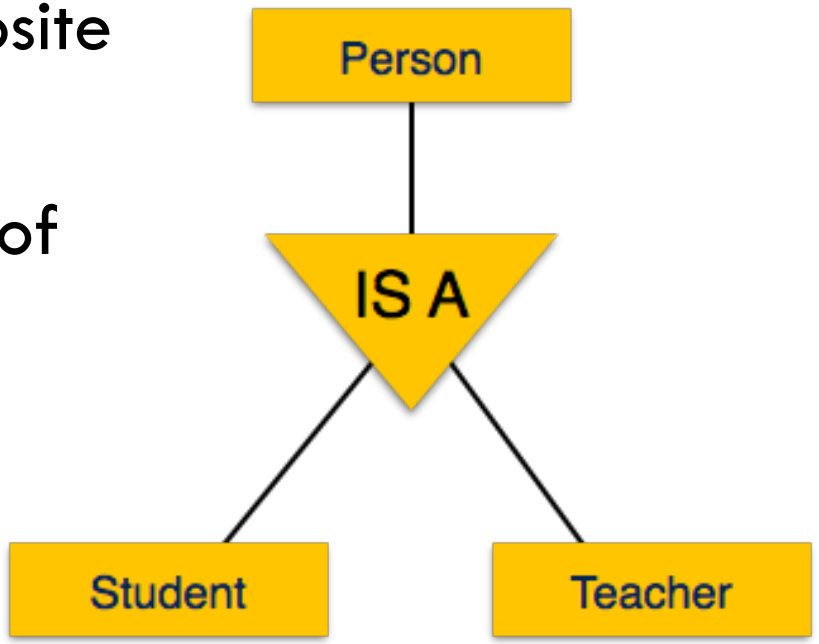
Generalization

- The process of generalizing entities, where the generalized entities contain the properties of all the generalized entities, is called **generalization**.
- In generalization, a number of entities are brought together into one generalized entity based on their similar characteristics.



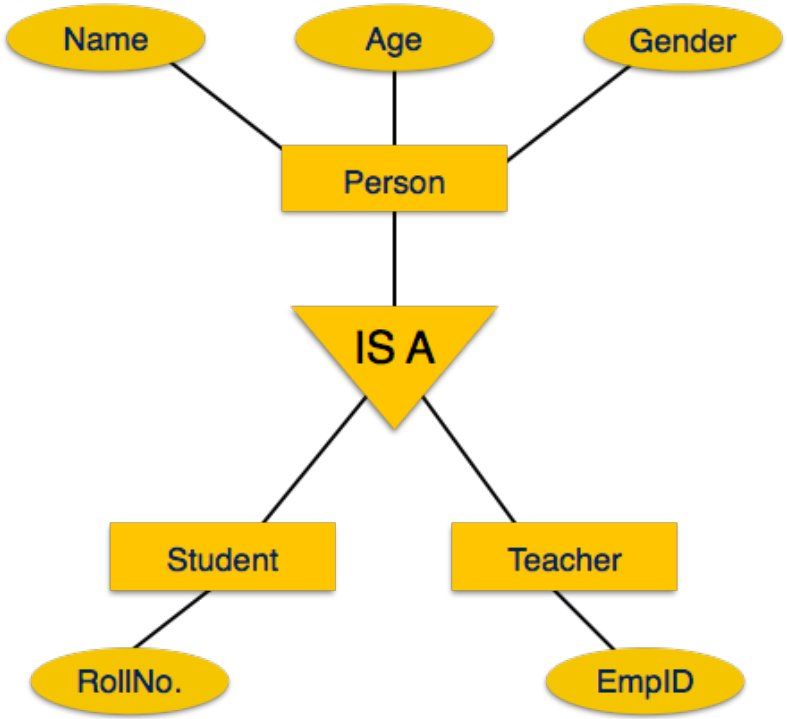
Specialization

- **Specialization** is the opposite of generalization
- In specialization, a group of entities is divided into subgroups based on their characteristics



Inheritance

- **Inheritance** is an important feature of Generalization and Specialization
- It allows lower-level entities to inherit the attributes of higher-level entities



ER Diagram Representation **Summary**

- **Entity** is a person, place, thing, or event about which data will be collected and stored.
- **Attributes** are the properties of entities.
- **Relationships** are the logical association among entities.
- The process of generalizing entities, where the generalized entities contain the properties of all the generalized entities, is called **generalization**.
- In **specialization**, a group of entities is divided into sub-groups based on their characteristics.
- **Inheritance** allows lower-level entities to inherit the attributes of higher-level entities.