5. DATABASE SYSTEMS

Айтмұханбетова Эльвира

D. Serikbayev East Kazakhstan Technical University

Copyright Notice

- This presentation uses materials from

Database Management System Tutorial

- https://www.tutorialspoint.com/dbms/

5.1: Introduction to Database Systems

5.1: Introduction to Database Systems

5.2: Data Models

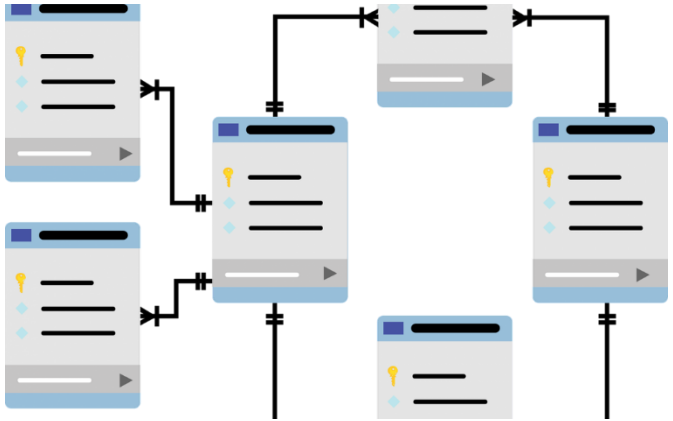
5.3: ER Diagram Representation

Learning Objectives

- Define database, DBMS and database systems
- Describe the database importance and its functions
- List database characteristics
- Understand business rules
- Define database users
- Understand 3-tier architecture

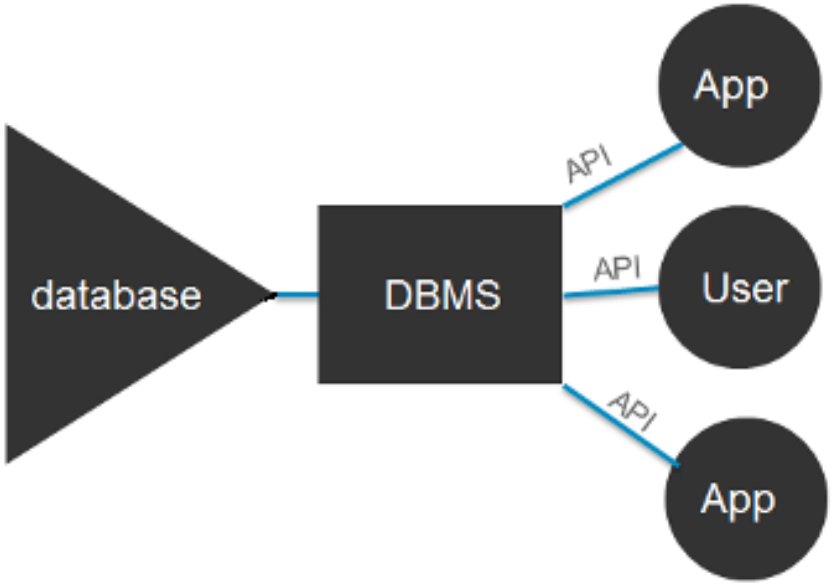
Introduction

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



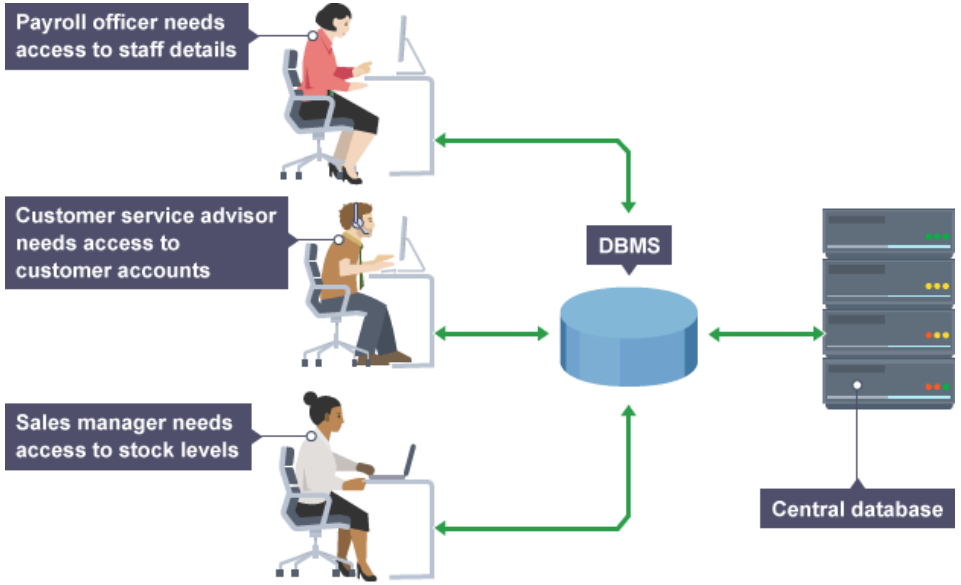
Introduction (2)

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



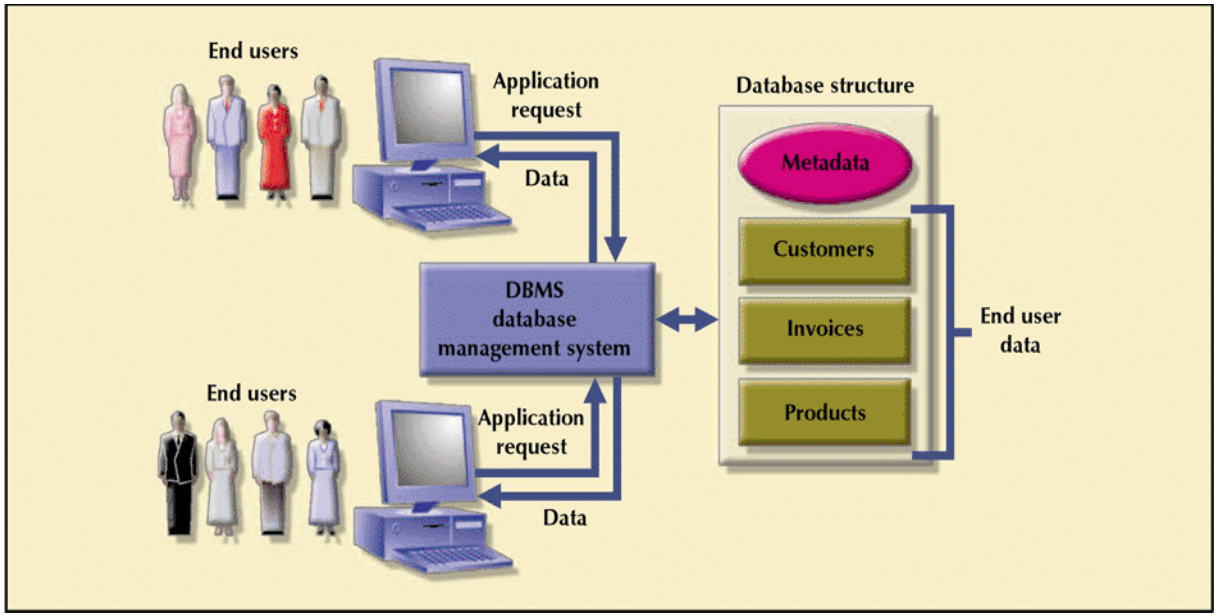
Introduction (3)

- 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**



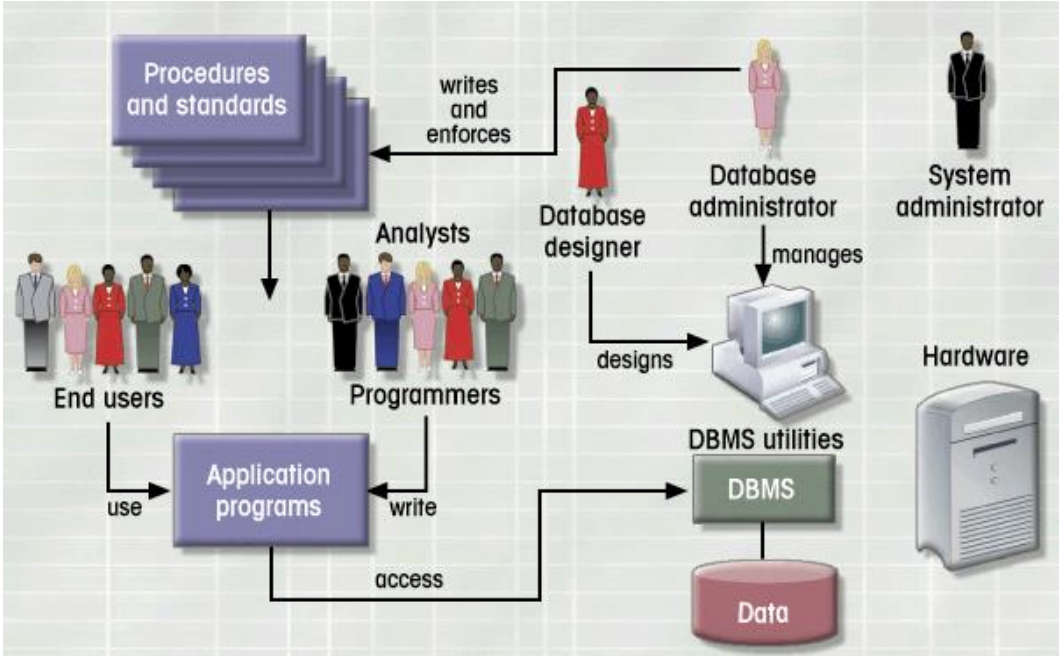
Database Management System (DBMS)

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



Database System Environment

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



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
    - different approach needed for different types of databases
  - Avoid data redundancy & ensure data integrity
    - data is accurate and verifiable
  - Poorly designed database generates errors
    - leads to bad decisions
    - can lead to failure of an organization or business

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
    - facilitates the construction/definition of data elements and their relationships
    - enables data transformation and presentation
  - Enforces data integrity
  - Implements data security management
    - access, privacy, backup & restoration

Database: Planning and Analysis

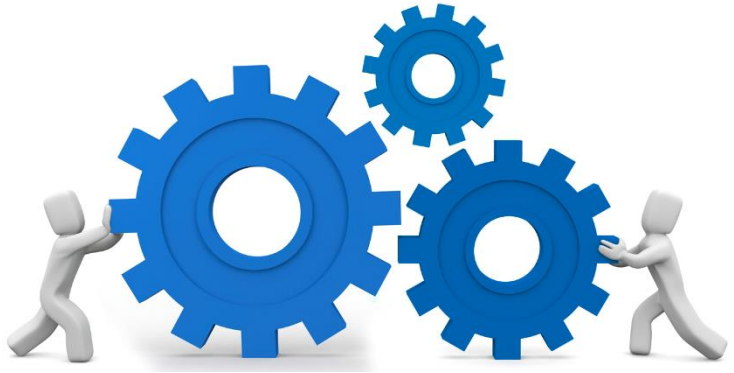
- Assess
  - Goal of the organization
  - Database environment
    - existing hardware, software, raw data, data processing procedures

Database: Planning and Analysis (2)

- Identify
  - Database needs
    - what database can do to further the goal of the organization
  - User needs and characteristics
    - who the users are, what they want to do, how they envision doing it
  - Database system requirements
    - what the database system should do to satisfy the database and user need

Database: Implementation

- Design
  - From conceptual design to a detailed system specification
- Implementation
  - Create the database
- Maintenance
  - Troubleshoot, update, streamline the database



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

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 (2)

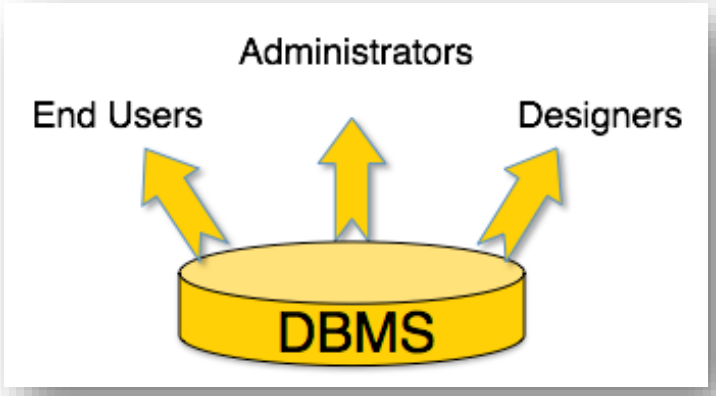
- Purposes
  - Enhance understanding & facilitate communication
    - Standardize company's view of data
    - Constitute a communications tool between users and designers
    - Allow designer to understand business process as well as the nature, role, and scope of data
  - Promote creation of an accurate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

Business Rules (3)

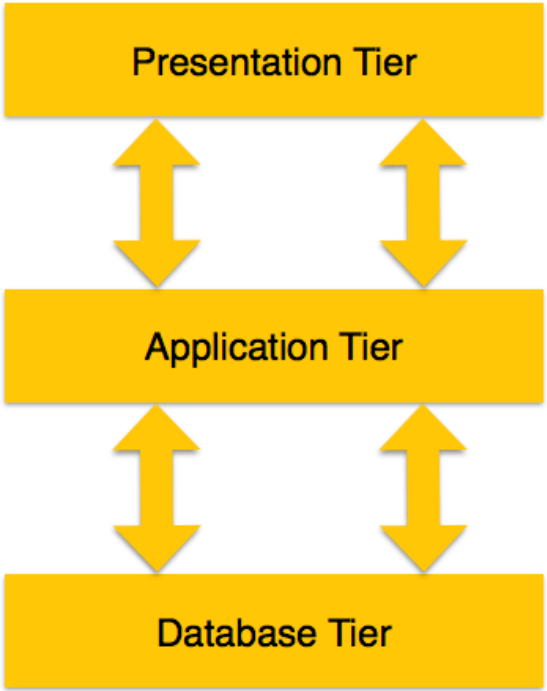
- Sources
  - Interviews
    - Company managers
    - Policymakers
    - Department managers
    - End users
  - Written documentation
    - Procedures, Standards, Operations manuals
  - Observation
    - Business operations

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



3-Tier Architecture



3-Tier Architecture (2)

- Database (Data) Tier
  - The database resides along with its query processing languages.
  - We also have the relations that define the data and their constraints at this level.
- Application (Middle) Tier
  - At this tier reside the application server and the programs that access the database.
  - For a user, this application tier presents an abstracted view of the database.

3-tier Architecture (4)

- User (Presentation) Tier
  - End-users operate on this tier
  - They know nothing about any existence of the database beyond this layer
  - The application provides multiple views of the database at this layer

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
- **Purpose of a Database**
  - Optimizes data management
  - Transforms data into information
- **Business rules**
  - Brief, precise, and unambiguous descriptions of operations in an organization
- **Users:** administrators, end users, and designers
- **3-tier architecture:** database tier, application tier, and user tier

5.2: Data Models

5.1: Introduction to Database Systems

5.2: Data Models

5.3: ER Diagram Representation

Learning Objectives

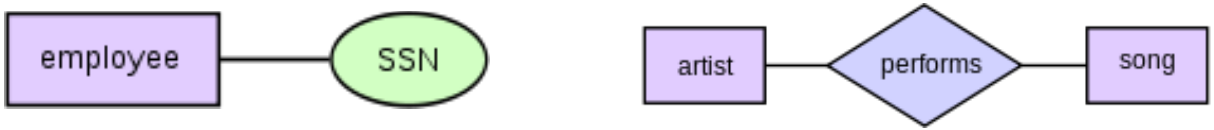
- Define data models
- Describe Entity-Relationship model
- Describe Relational model
- Understand ER model basic concepts

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.

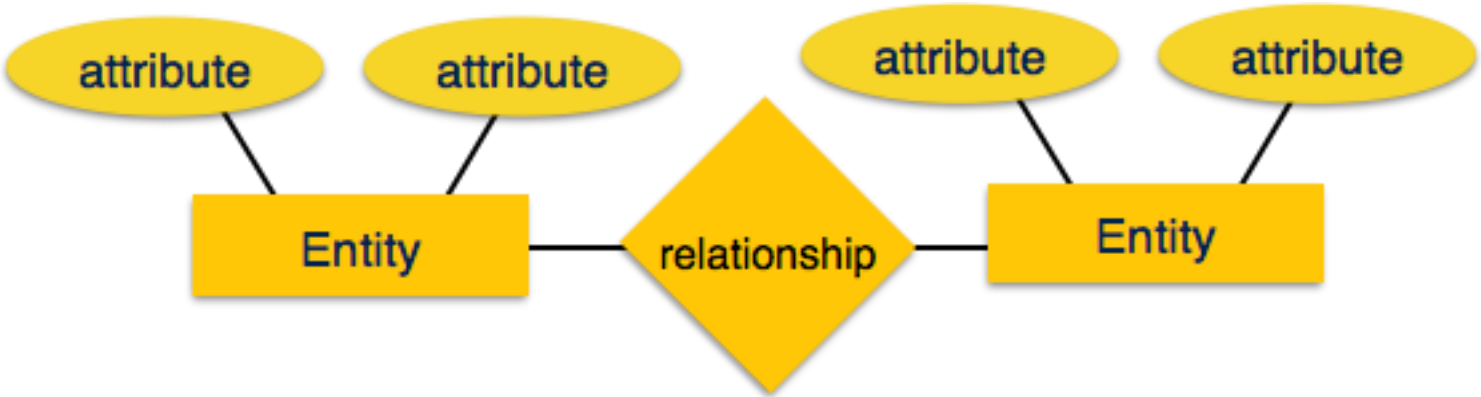
Data Models: Entity-Relationship Model

- **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*



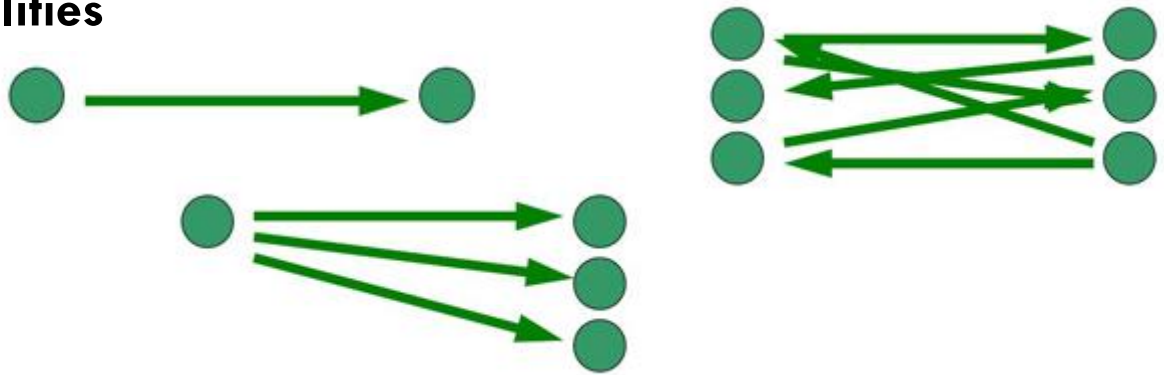
Data Models: Entity-Relationship Model (2)

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



Data Models: Entity-Relationship Model (3)

- **Entity** – An entity in an ER Model is *a real-world entity* having properties called **attributes**
  - Every **attribute** is defined by its set of values called **domains**
- **Relationship** – The logical association among entities is called ***relationships***.
  - Relationships are mapped with entities in various ways.
  - Mapping cardinalities define the number of association between two entities.
- **Mapping cardinalities**
  - one-to-one
  - one-to-many
  - many-to-one
  - many-to-many

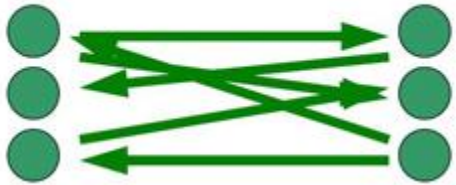


Data Models: Entity-Relationship Model (4)

One to one relationships connect one entity to one other entity:

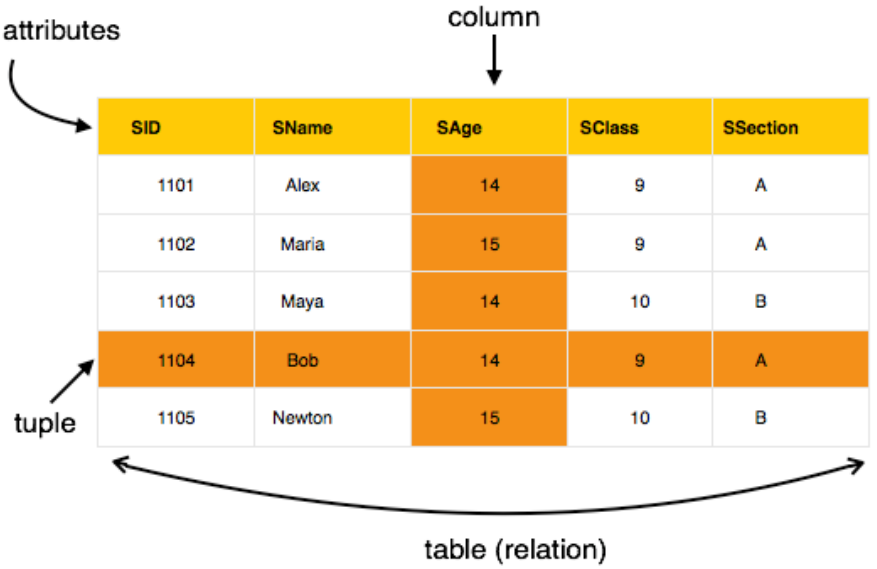
One to many relationships connect one entity to one or more other entities:

Many to many relationships connect many entities to many other entities:



Data Models: Relational Model

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



SID	SName	SAge	SClass	SSection
1101	Alex	14	9	A
1102	Maria	15	9	A
1103	Maya	14	10	B
1104	Bob	14	9	A
1105	Newton	15	10	B

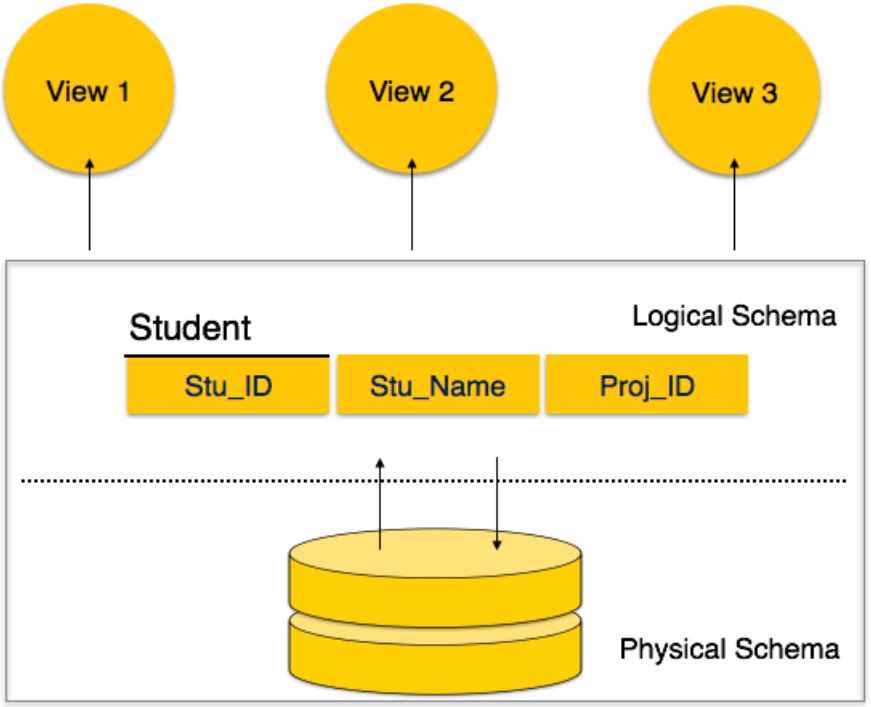
Data Models: Relational Model (2)

- 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

Database Schema

- Is the **skeleton structure** that represents the **logical** view of the entire database
  - It defines how the data is organized and how the relations among them are associated
- **Defines** its **entities** and the **relationship** among them
  - It contains a descriptive detail of the database, which can be depicted using schema diagrams

Database Schema (2)



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

ER Model - Basic Concepts

- **Entity**
  - Can be a real-world object, either animate or inanimate, that can be easily identifiable
  - For example, in a school database, students, teachers, classes, and courses offered can be considered as entities.
- **Attributes**
  - Entities are represented by their properties
    - All attributes have values
  - For example, a student entity may have name, class, and age as attributes

Types of Attributes

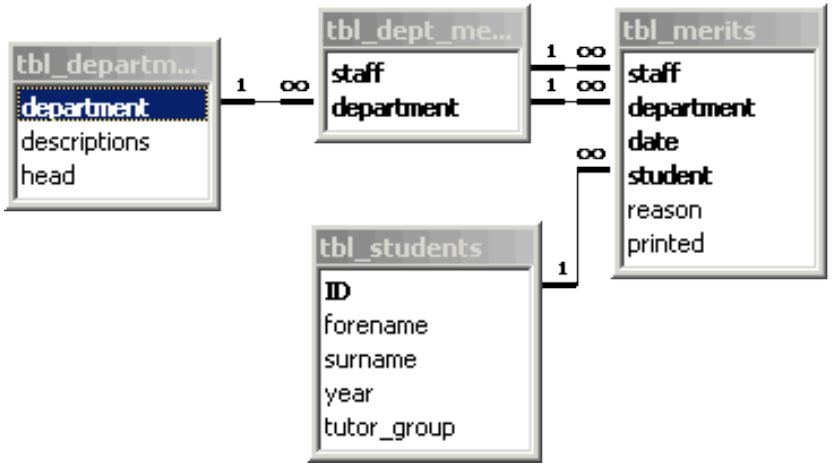
- **Simple attribute**
  - Atomic values, which cannot be divided further
- **Composite attribute**
  - Are made of more than one simple attribute. For example, a student's complete name may have first_name and last_name.
- **Derived attribute**
  - The attributes that do not exist in the physical database, but their values are derived from other attributes present in the database
- **Single-value attribute**
  - Contains a single value
- **Multi-value attribute**
  - May contain more than one values

Entity-Set and Keys

- **Key** is an attribute or collection of attributes that uniquely identifies an entity among entity set.
- Types:
  - **Super Key**
    - A set of attributes (one or more) that collectively identifies an entity in an entity set.
  - **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.

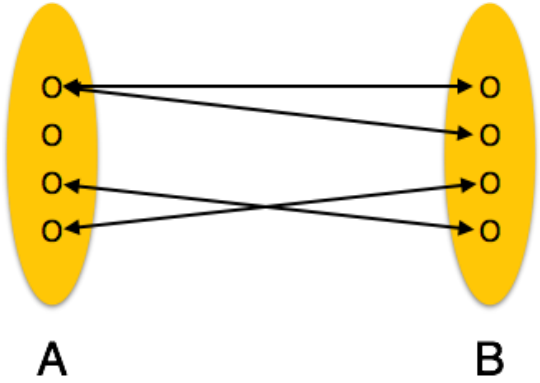
Relationship

- The association among entities is called a **relationship**. For example, an employee **works at** a department, a student **enrolls** in a course. *Works at* and *Enrolls* are called relationships.
- **Degree of relationship**
  - Binary = degree 2
  - Ternary = degree 3
  - n-ary = degree



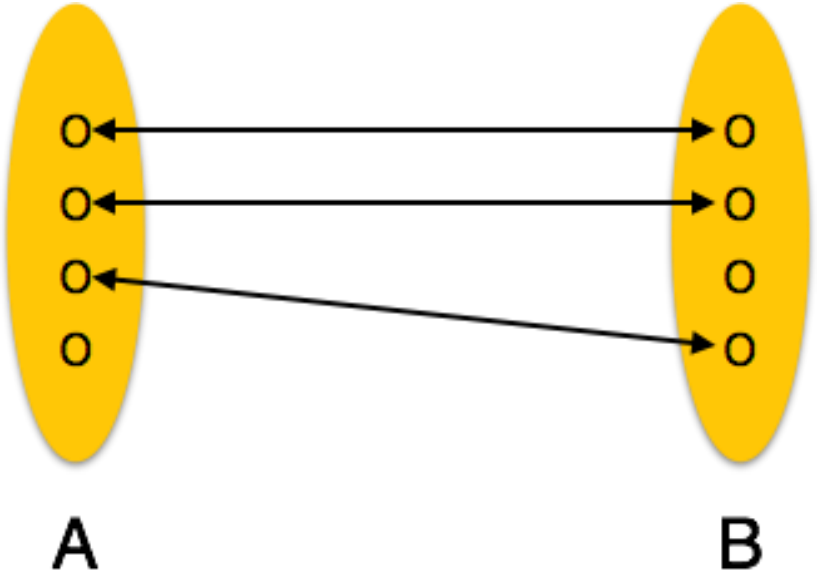
Mapping Cardinalities

- **Cardinality** defines the number of entities in one entity set, which can be associated with the number of entities of another set via relationship set.
  - **One-to-one**
  - **One-to-many**
  - **Many-to-one**
  - **Many-to-many**



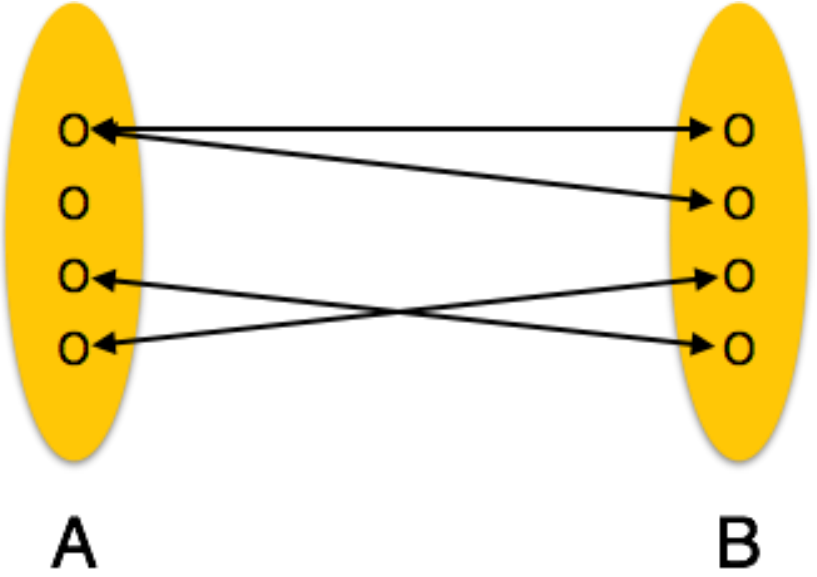
Mapping Cardinalities (2)

- **One-to-one** – One entity from entity set A can be associated with at most one entity of entity set B and vice versa



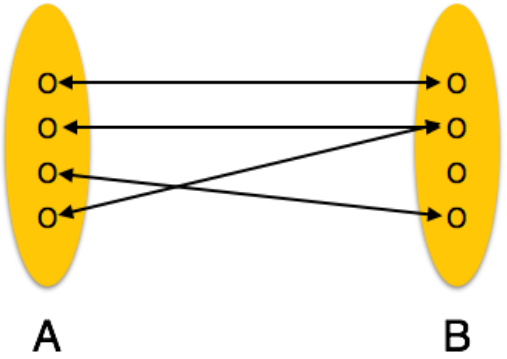
Mapping Cardinalities (3)

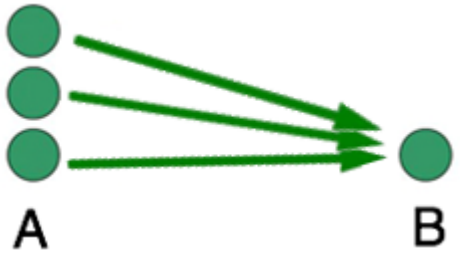
- **One-to-many** – One entity from entity set A can be associated with more than one entities of entity set B
- However, an entity from entity set B, can be associated with at most one entity



Mapping Cardinalities (4)

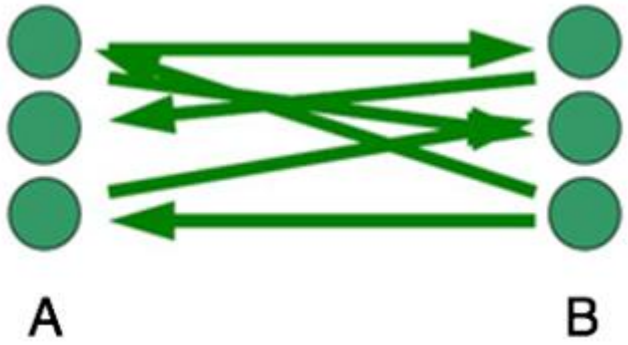
- **Many-to-one** – More than one entities from entity set A can be associated with at most one entity of entity set B
- However, an entity from entity set B can be associated with more than one entity from entity set A

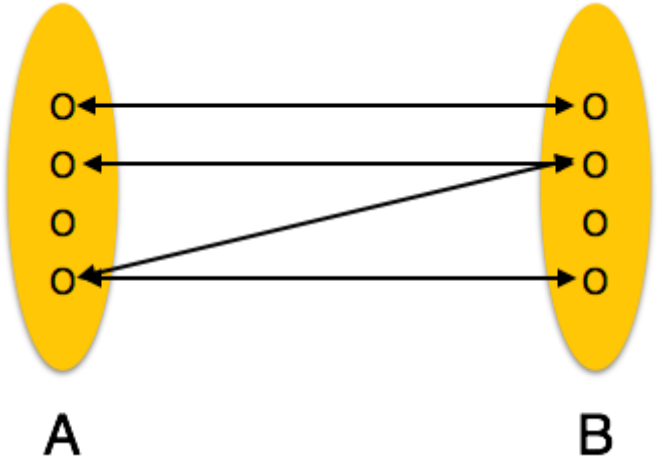




Mapping Cardinalities (5)

- **Many-to-many** – One entity from A can be associated with more than one entity from B, and vice versa.





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

5.3: ER Diagram Representation

5.1: Introduction to Database Systems

5.2: Data Models

5.3: ER Diagram Representation

Learning Objectives

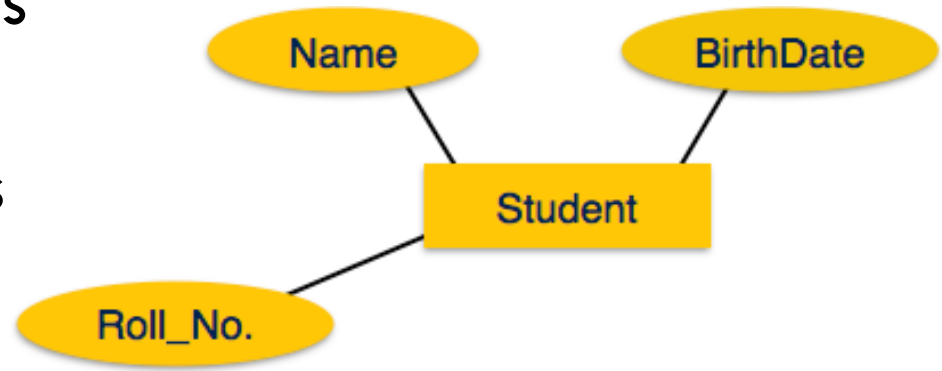
- Understand entity representation
- Describe attribute representation
- Describe relationship representation
- Understand generalization
- Define specialization
- Describe inheritance

ER Diagram Representation: Entity

- **Entities** are represented Using rectangles
- Rectangles are named with the entity set they represent

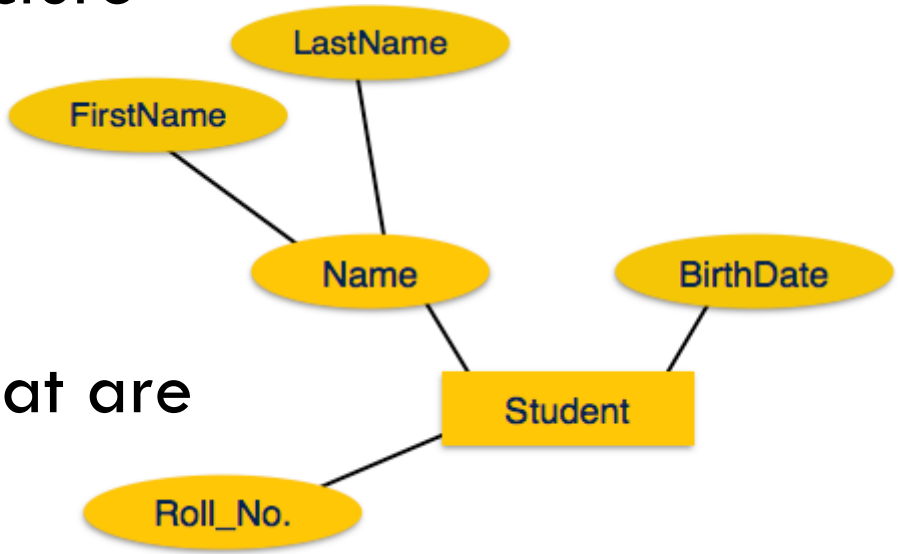
ER Diagram Representation: Attributes

- **Attributes** are the properties of entities
- Attributes are represented using ellipses
- Every ellipse represents one attribute and is directly connected to its entity (rectangle)



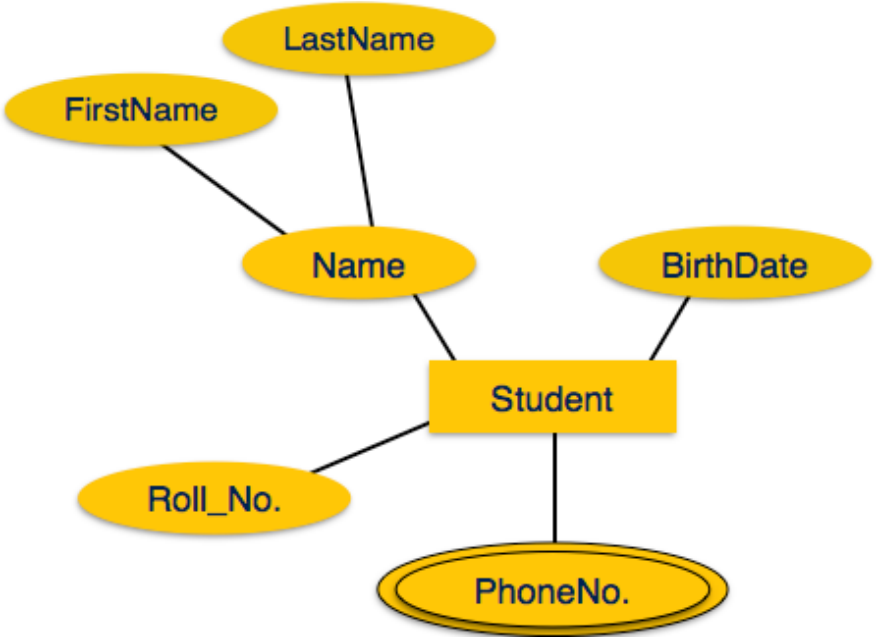
ER Diagram Representation: Attributes (2)

- 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



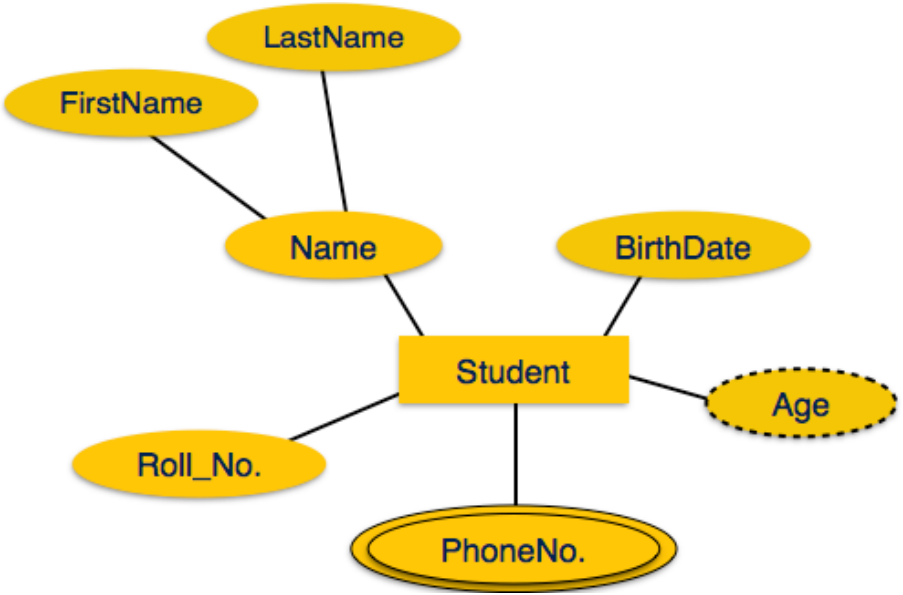
ER Diagram Representation: Attributes (3)

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



ER Diagram Representation: Attributes (4)

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

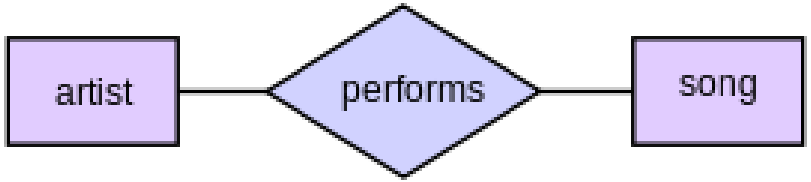


ER Diagram Representation: Relationship

- **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

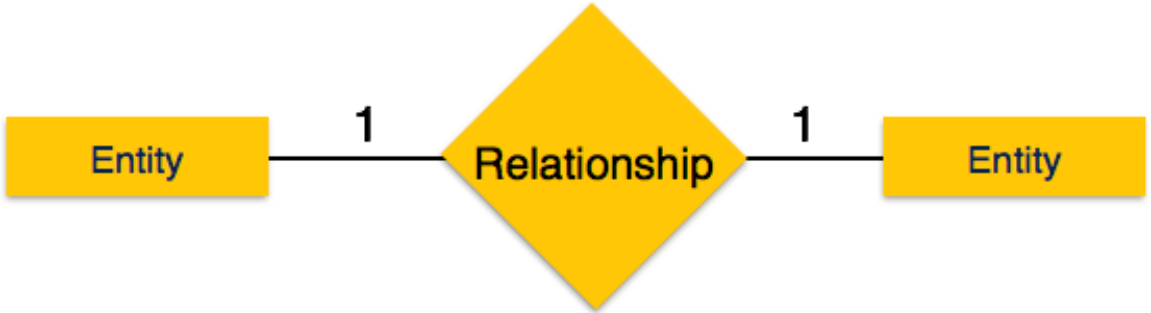
ER Diagram Representation: Relationship

- **Cardinality and Binary Relationship**
  - Cardinality is the **number of instances** of an entity from a relation that can be associated with the relation
  - A **binary relationship** is where only two entities are involved, such as artist and song



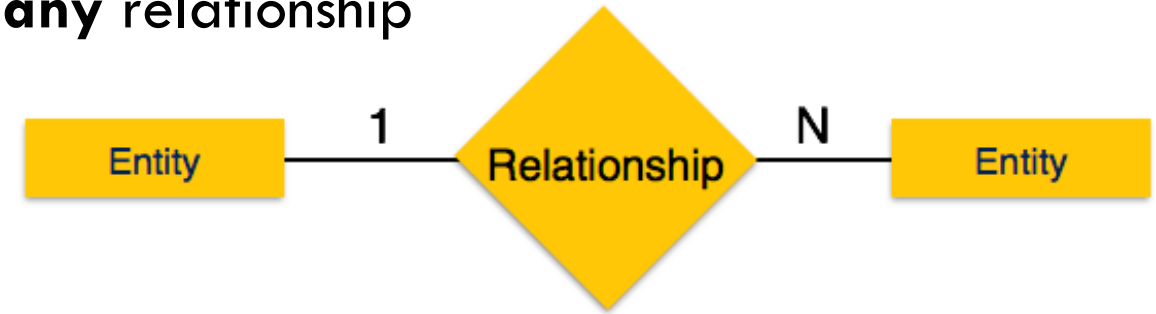
Relationship Cardinalities

- **One-to-one** – When only one instance of an entity is associated with the relationship, it is marked as '**1:1**'.
- The following image reflects that only one instance of each entity should be associated with the relationship.
- It depicts **one-to-one** relationship



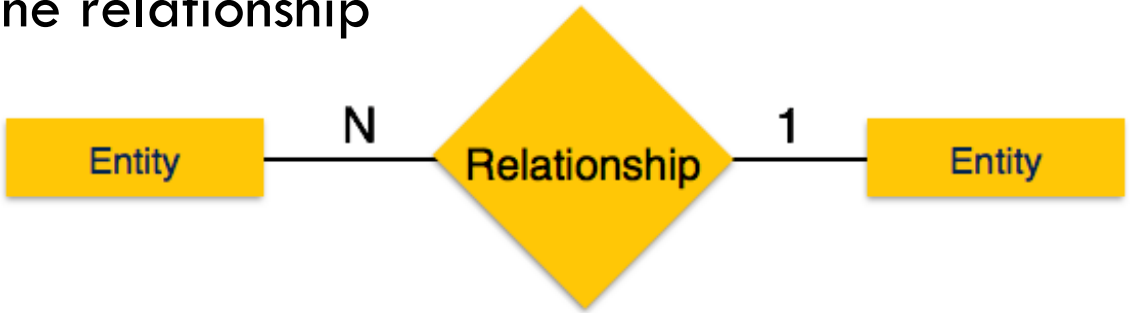
Relationship Cardinalities (2)

- **One-to-many** – When more than one instance of an entity is associated with a relationship, it is marked as '**1:N**'.
- The following image reflects that only 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.
- It depicts **one-to-many** relationship



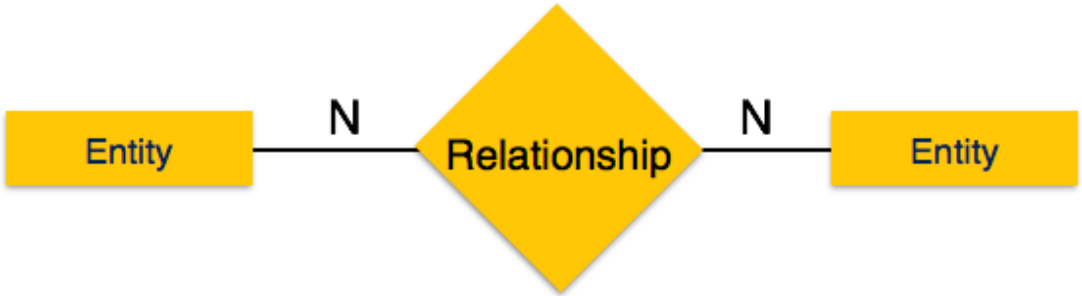
Relationship Cardinalities (3)

- **Many-to-one** – When more than one instance of an entity is associated with the relationship, it is marked as **'N:1'**.
- The following image reflects that more than one instance of an entity on the left and only one instance of an entity on the right can be associated with the relationship.
- It depicts many-to-one relationship



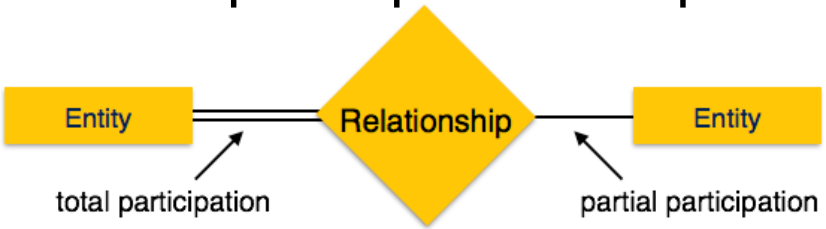
Relationship Cardinalities (4)

- **Many-to-many** – The following image reflects that 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.
- It depicts **many-to-many** relationship



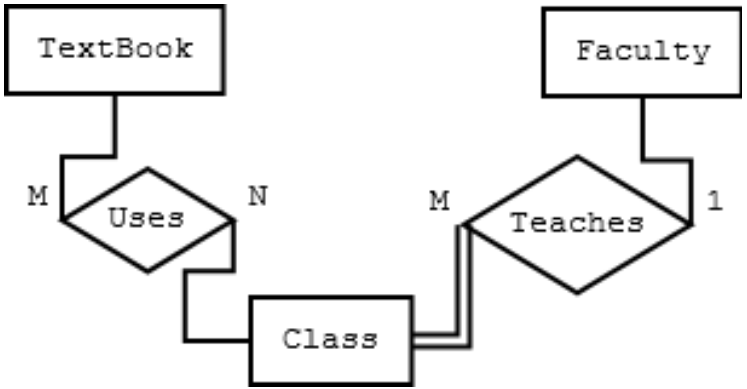
Participation Constraints

- **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.



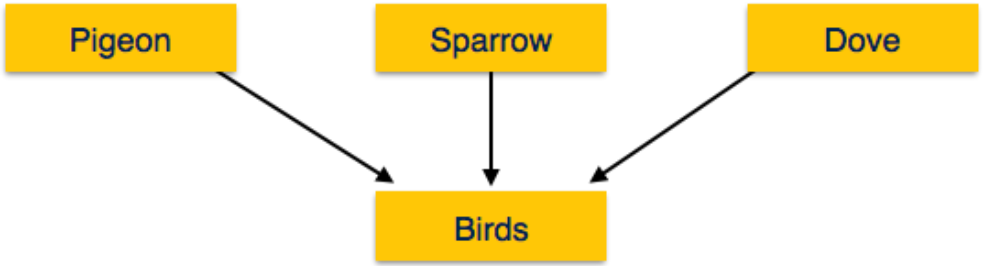
Participation Constraints (2)

- For example: A ***Class*** entity cannot exist unless it is related to a ***Faculty*** member entity



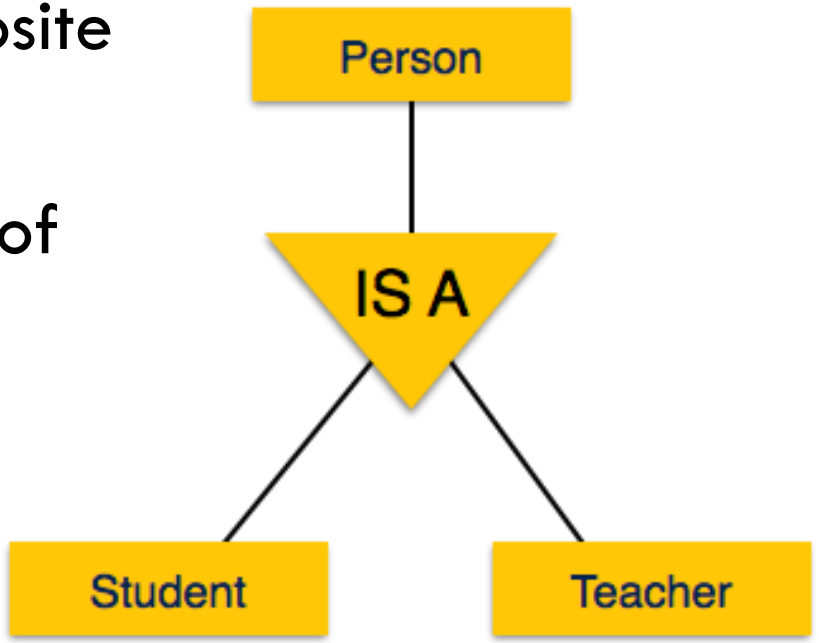
Generalization

- 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.



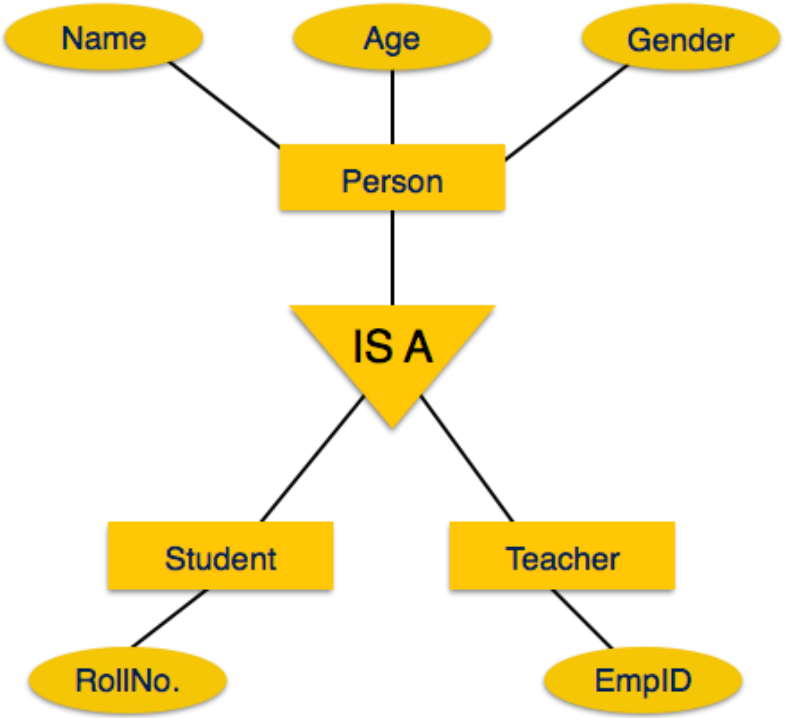
Specialization

- **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**

- **Entities** are represented using rectangles.
- **Attributes** are the properties of entities. Attributes are represented using ellipses.
- **Relationships** are represented by diamond-shaped boxes.
- 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.