

Northeastern University

---

# ENTITY RELATIONSHIP DIAGRAM

***INFO6210 Data Management and Database Design***

*Professor Chaipayorn Mutsaklisana*

Northeastern University

---

## Agenda

- What is a Database?
- Entity Relationship Model / Diagram
- Components of ERD
- Types of Keys
- Crow's Foot Notation (Cardinality)
- Relationship Degree
- Example of ERD
- Enhanced ER Diagram
- Generalization and Specialization
- Example of EERD
- ERD vs EERD
- References

## What is a Database?

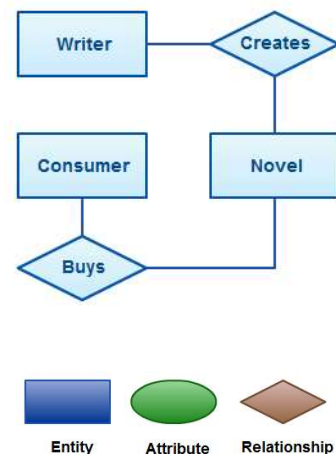
- In simple terms, a database is a system for storing and taking care of data
  - More specifically, a database is an electronic system that allows data to be easily stored, accessed, manipulated and updated.
  - For e.g.;
- An organization's database will store its employee data



Employee_Id	Employee_Name	Employee_Address
A123	Bill Label	15 Benton St
B234	Nancy Edge	118 Boylston St
C345	James Cooper	215 Harrison St
D456	Mark Wright	88 Rector St

## Entity Relationship Model / Diagram

- An Entity Relationship (ER) diagram displays relationship between entities (e.g.; object or a concept)
- Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes

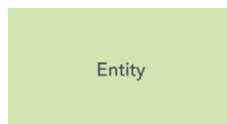


## Entity Relationship Model / Diagram contd.

- ERDs mirror grammatical structure, with entities as nouns and relationships as verbs
- Components of an ERD are:
  - Entity
  - Relationship
  - Attribute
  - Key
  - Cardinality

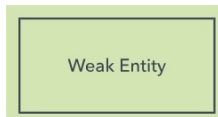
## Entity

- An entity can be a person, place, event, concept or object whose data is relevant to a given system
- An entity might have:
  - Physical existence (e.g., a lecturer, a student, a car)
  - Conceptual existence (e.g., a course, a job, a position)
- They are labeled using single nouns and are represented in a rectangle in the diagram



## Entity contd.

- **Entity type:** A group of definable things, such as students or athletes, whereas the entity would be the specific student or athlete
- **Entity set:** Same as an entity type, but defined at a particular point in time, such as students enrolled in a class on the first day
- **Weak Entity:**
  - Entities can be classified based on their strength
  - An entity is considered to be weak if it cannot exist without a relationship with another entity e.g. Order Item entity is dependent on Order entity
  - Typically represented as a rectangle-inside-a-rectangle



## Relationship

- A relationship describes how entities interact with each other
- They can be thought of as verbs and are represented in a diamond in the diagram
- For e.g.; the entity "Student" may be related to the entity "Course" by the relationship "enrolls" or "studies"
- A relationship is displayed to be weak (diamond-in-a-diamond) while displaying relationship of a weak entity with another entity



## Attribute

- An attribute is a property, trait, or characteristic of an entity, relationship, or another attribute.
- For e.g.; A “Student” entity will have attributes such as Name, Age etc.
- They are represented in a oval shape in the diagram
- Types of Attributes:
  - Simple attributes
  - Composite attributes
  - Multivalued attributes
  - Derived attributes



## Attribute contd.

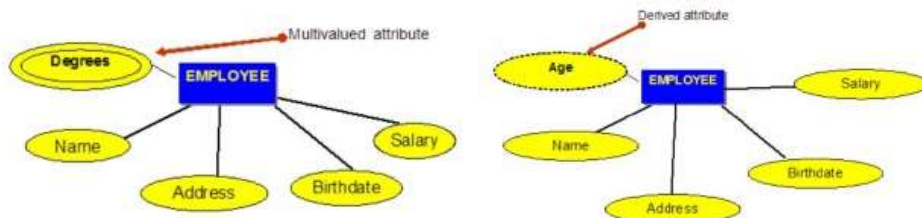
- Types of attribute:
  - **Simple attributes** are those drawn from the atomic value domains i.e. which cannot be divided further ; they are also called single-valued attributes. For e.g.; Name, Salary etc.
  - **Composite attributes** are those that consist of a hierarchy of attributes i.e. which can be further divided to more attributes. For e.g.; Address



Note: Based on how 'Name' is stored, it can be a simple or composite attribute

## Attribute contd.

- **Multivalued attributes** are attributes that have a set of values for each entity. For e.g.; the Degrees of an Employee: BSc, MIT, PhD, etc.
- **Derived attributes** are attributes that contain values calculated from other attributes. For e.g.; Age can be derived from the attribute Birthdate



## Types of Keys

- Key is an attribute or a set of attribute used to identify records stored in tables of the database
- **Primary key:** An attribute or set of attributes that is unique, not null and cannot be changed  
e.g.; Employee\_ID of an employee, Student\_ID of a student
- **Super key:** An attribute or set of attributes using which can uniquely identify or select a row among the set of rows
- **Candidate Key:** A 'subset' of Super Key (minimal set of attribute(s)) which fulfills the role of a Super Key
- All 'Candidate Keys' are 'Super Keys' but the inverse is **not** true

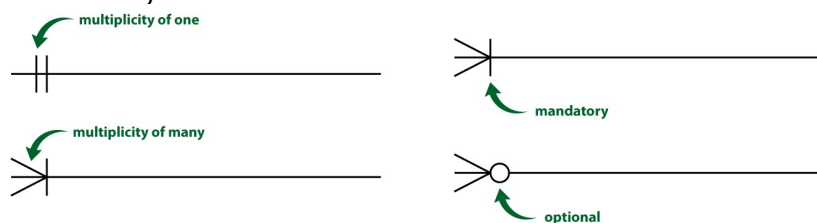


## Types of Keys contd.

- **Foreign key:** A foreign key (FK) is an attribute in a table that references the primary key in another table OR it can be null
- **Composite key :** A composite key is composed of two or more attributes, but it must be minimal
- **Alternate key :** Alternate keys are all candidate keys not chosen as the primary key.

## Crow's Foot Notation (Cardinality)

- Relationships have two indicators and are shown on both sides of the line
- The first one (often called multiplicity) refers to the maximum number of times that an instance of one entity can be associated with instances in the related entity which can be one or many
- The second describes the minimum number of times one instance can be related to others. It can be zero or one, and accordingly describes the relationship as optional or mandatory



## Crow's Foot Notation contd.

- In Crow's foot notation:
  - A multiplicity of **one** and a **mandatory** relationship is represented by a straight line perpendicular to the relationship line
  - A multiplicity of **many** is represented by the three-pronged 'crow-foot' symbol
  - An **optional** relationship is represented by an empty circle
- Four possible edges for a relationship:
  - Zero or many



## Crow's Foot Notation contd.

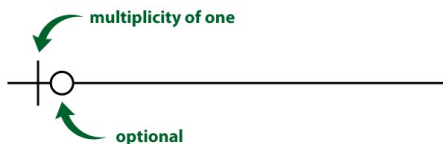
- One or many



- One and only one



- Zero or one





## Crow's Foot Notation contd.

- Example:

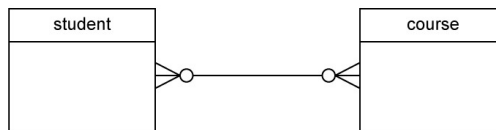
■ One-to-one



■ One-to-many



■ Many-to-many

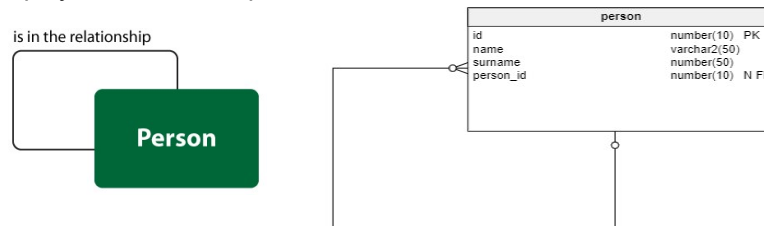


## Relationship Degree

The most common types of relationship degrees are:

**1) Unary** (one entity is involved in the relationship)

- A Unary relationship usually exist between entities of a single entity type is displayed in the example below:

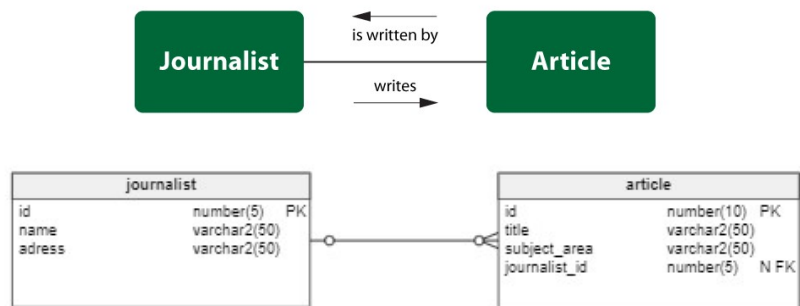


- As we see, a person can be in the relationship with another person, such as:
  - A woman who can be someone's mother
  - A person that is a someone's child

## Relationship Degree contd.

### 2) Binary (two entities are involved in the relationship)

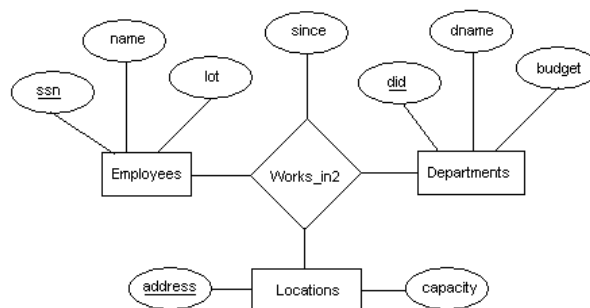
- A Binary relationship type is shown in the simple example below
- This is definitely the most used relationship type
- E.g.; Journalist writes an article



## Relationship Degree contd.

### 3) Ternary (three entities are involved in the relationship)

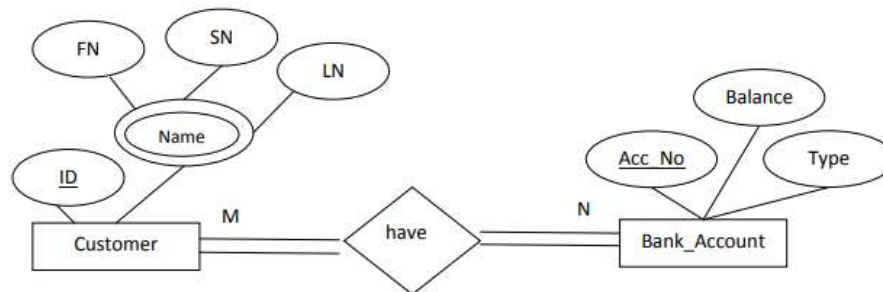
- A ternary relationship is a relationship type that involves many to many relationships between three tables
- The following example shows the relationship between Employees, Departments and Locations



## Entity Relationship Diagram Example

1. The following ER diagram displays many to many relationship between the Customer and Bank\_Account entities

Ex :- Simple Transaction Processing System of a Bank.



## Enhanced Entity Relationship Diagram(EERD)

- Enhanced Entity Relationship Diagram(EERD), also known as Extended Entity Relationship Model, is an advanced database diagram very similar to regular ER diagram
- In addition to the same concepts that the ordinary ER diagram encompasses, EERD includes:
  - Subtypes and supertypes (sometimes known as subclasses - and superclasses)
  - Specialization and Generalization
  - Category or Union type
  - Attribute and Relationship inheritance

### Subtypes, Supertypes and Inheritance:

- Subtype - A subgroup of entities with unique attributes
- Supertype - An entity type that relates to one or more subtypes
- Inheritance - The concept that subtype entities inherit the values of all supertype attributes

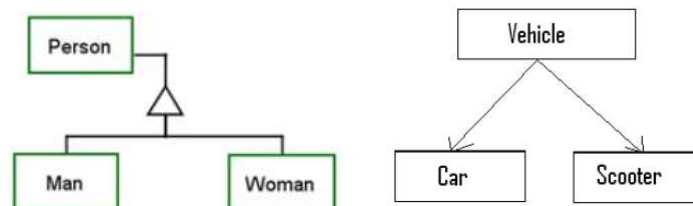
## Generalization & Specialization

- **Generalization & Specialization** represent logical links between an entity E, known as parent entity, and one or more entities  $E_1, \dots, E_n$  called child entities, of which E is more general, in the sense that they are a particular case
- In this situation we say that E is a **generalization** of  $E_1, \dots, E_n$  and that the entities  $E_1, \dots, E_n$  are **specializations** of E
- A generalization is **total** if every instance of the parent entity is also an instance of one of its children, otherwise it is **partial**
- A generalization is **exclusive or disjoint** if every instance of the parent entity is at the most an instance of one of the children, otherwise it is **overlapping**

## Generalization & Specialization contd.

For example;

- Generalization of Person as Man and Woman is **total** (the sets of men and the women constitute 'all' the people) and **exclusive** (a person is either a man or a woman)
- Generalization of Vehicle as Car and Scooter is **partial** and **exclusive**, because there are other types of vehicle (for e.g.; Truck) that are neither Cars nor Scooter
- The generalization of Person as Student and Employee is **partial** and **overlapping**, because there are students who can also be employed

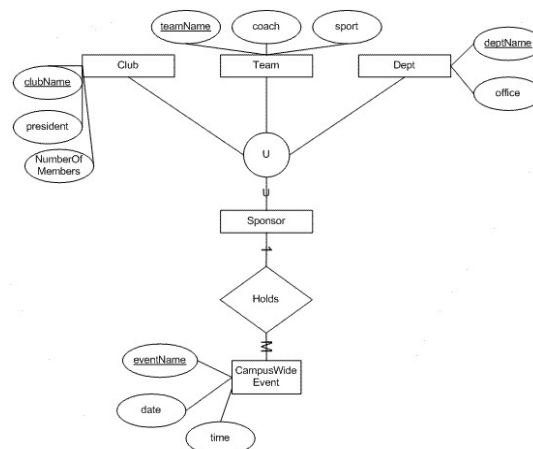


## Union or Category

- Subclass related to a collection of superclasses
- Each instance of subclass belongs to one, not all, of the superclasses
- Superclasses form a union or category
- For e.g.;
  - A Sponsor for an event can be a team, a department or an entire club
  - Each Sponsor entity instance is a member of one of these superclasses, so Sponsor is a subclass of the union of Team, Dept, Organization

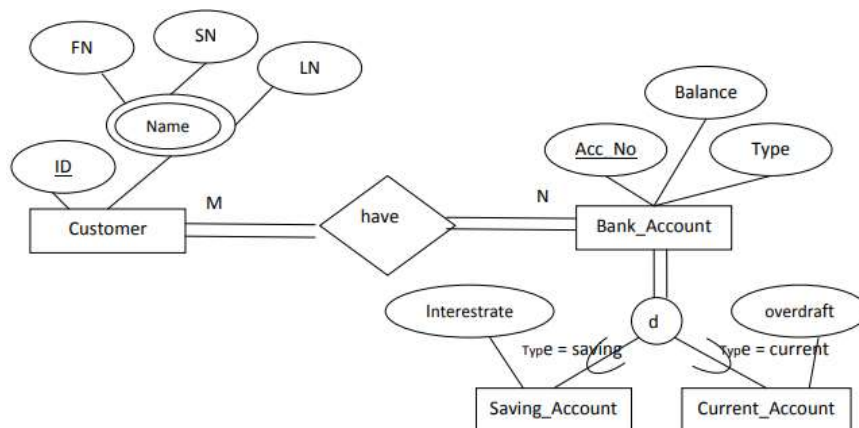
## Union or Category contd.

- Consider the following EER diagram- connect each superclass using union circle symbol, which then connect to subclass



## EERD Example

- The following EER diagram displays many to many relationship between the Customer and Bank\_Account entities similar to an ER diagram but it also displays inheritance (Specialization/Generalization)



## ERD vs EERD

Sr. No.	Entity Relationship Diagram	Enhanced Entity Relationship Diagram
1.	ER Diagram consists of set of entities and relationship among these entities	EER Diagram is used to represent a collection of objects i.e union of objects of different entity types
2.	The concept of specialization and generalization is absent	It includes the concept of specialization and generalization
3.	ER Diagram is the basic model	EER Diagram includes all the modelling concept of ER Diagram along with features such as subclass and superclass, Specialization and Generalization, Union and category, Aggregate, etc.

## References

- <https://www.smartsheet.com/relational-database>
- <https://creately.com/blog/diagrams/er-diagrams-tutorial/>
- <http://www.vertabelo.com/blog/technical-articles/crow-s-foot-notation>
- <http://www.itpanthiya.com/downloads/knowledgebase/articles/ERVSEER.pdf>
- <http://www.ques10.com/p/14149/compare-er-and-eer-models-1/>
- <https://www.geeksforgeeks.org/dbms-enhanced-er-model/>