جامــــــعـة المـــــســتـقـبـل
AL MUSTAQBAL UNIVERSITY

كلية العلوم
قســـم الانظمة الطبية الذكية

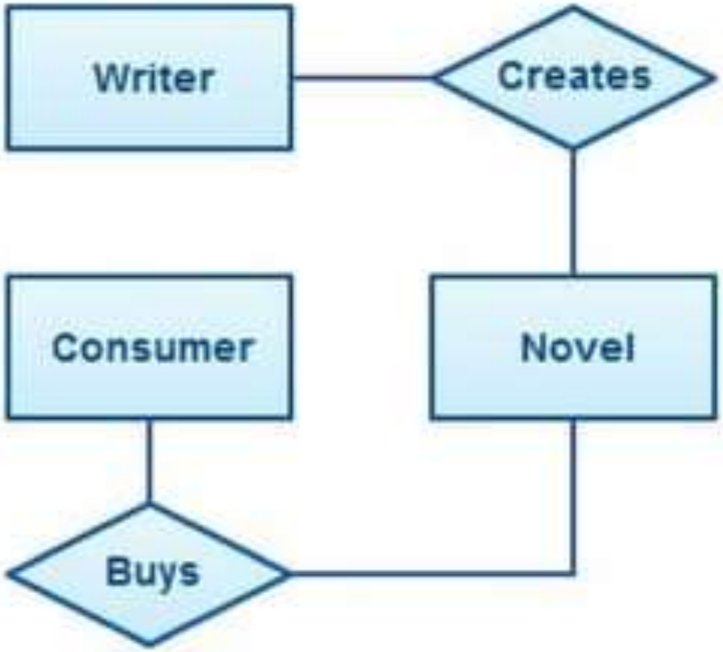
Lecture: (6)

Subject: **Entity Relationship Model**

Level: Second
Lecturer: MSc. Mustafa Yousif

# Entity Relationship Model

- An Entity Relationship Diagram (ERD) is a ***visual representation of different data using conventions*** that ***describe how these data are related to each other***.
- For example, the elements writer, novel, and consumer may be described using ER diagrams this way:



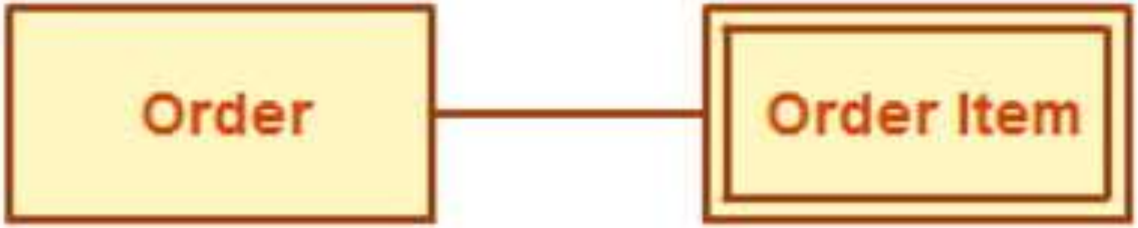
The elements inside **rectangles** are called **entities** while the items inside **diamonds** denote the **relationships** between entities.

# Entity

- ***An 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**.
  - All these entities have some ***attributes or properties*** that give them their identity.
- An **entity set** is a collection of similar types of entities.
  - An entity set may contain ***entities with attribute sharing similar values.***
  - For example,
  - a Students set may contain all the students of a school;
  - likewise a Teachers set may contain all the teachers of a school from all faculties. Entity sets need not be disjoint.

# Weak Entity

- **A weak entity is an entity that depends on the existence of another entity.**
- In more technical terms it can defined as an entity that cannot be identified by its own attributes.
- It uses a foreign key combined with its attributed to form the primary key.
- An entity like **order item** is a good example for this.
  - The order item will be meaningless without an order so it depends on the existence of order.



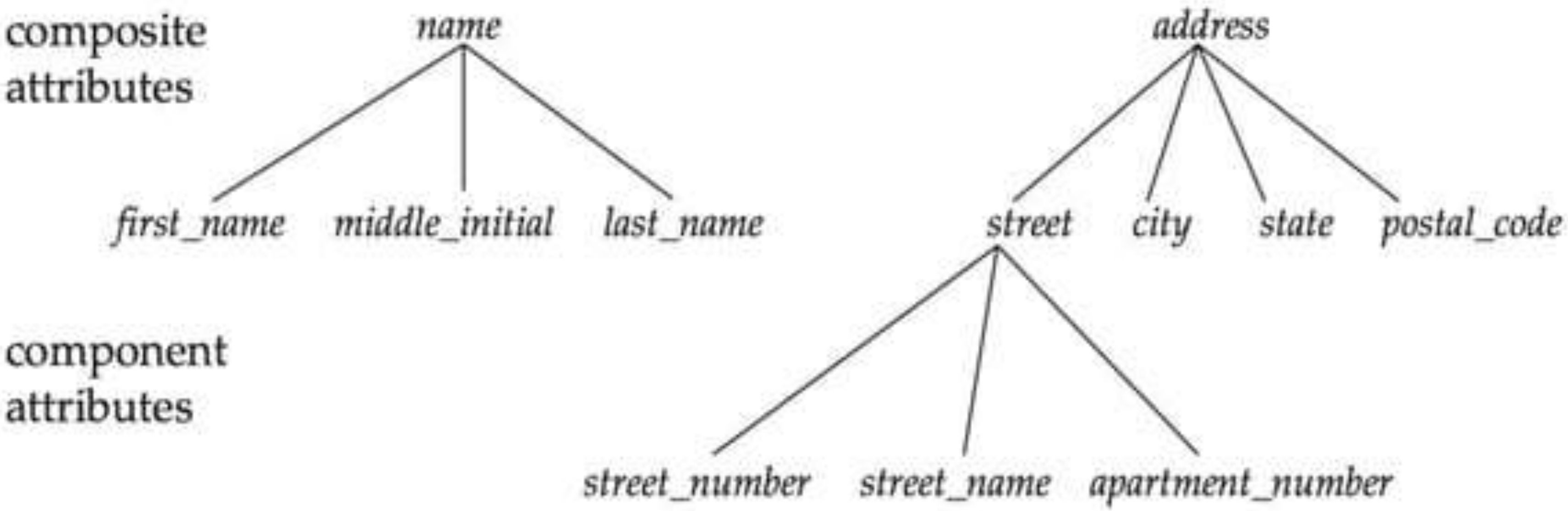
# Attributes

- 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.
- ***There exists a domain or range of values that can be assigned to attributes.***
- For example, a **student's name** cannot be a numeric value. It has to be ***alphabetic***. A **student's age cannot be negative**, etc.

# TYPES OF ATTRIBUTES

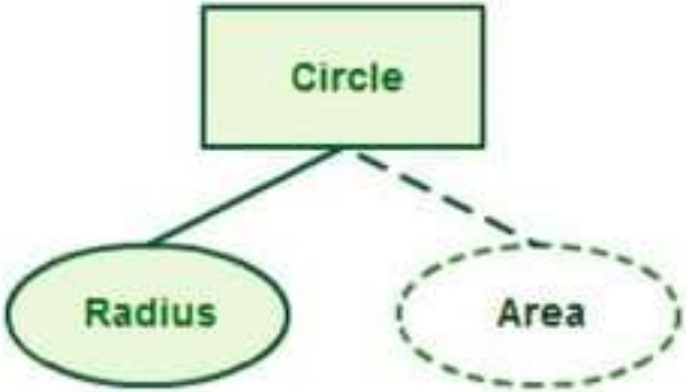
**Simple Attribute**: Attribute that consist of a single atomic value.
Example: Salary

**Composite Attribute** : Attribute value not atomic.
Example : Address : 'House _no:City:State
Name : 'First Name: Middle Name: Last Name'



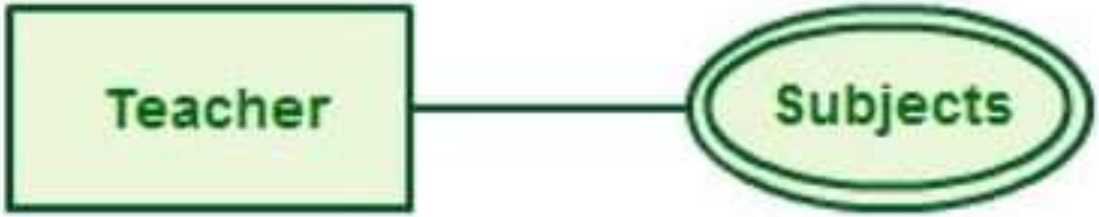
# Types of Attributes (contd.,)

- **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.
- For example for a circle the area can be derived from the radius.



# Types of Attributes (contd.,)

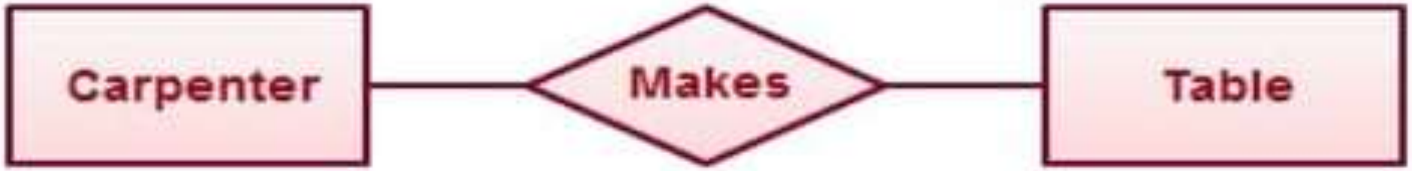
- **Single-value attribute** − Single-value attributes **contain single value**. For example − Social_Security_Number.
- **Multi-value attribute** − Multi-value attributes may **contain more than one values**. For example, a person can have more than one phone number, email_address, etc.
- For example a teacher entity can have multiple subject values.



- These attribute types can come together in a way like −
- ***simple single-valued attributes***
- ***simple multi-valued attributes***
- ***composite single-valued attributes***
- ***composite multi-valued attributes***

# Relationship

- The association among entities is called a relationship.
- For example, an employee **works_at** a department, a student **enrolls** in a course.
- Here, Works_at and Enrolls are called relationships.
- For example, the entity "**carpenter**" may be related to the **entity** "**table**" by the **relationship "builds" or "makes**". Relationships are represented by diamond shapes and are labeled using verbs.

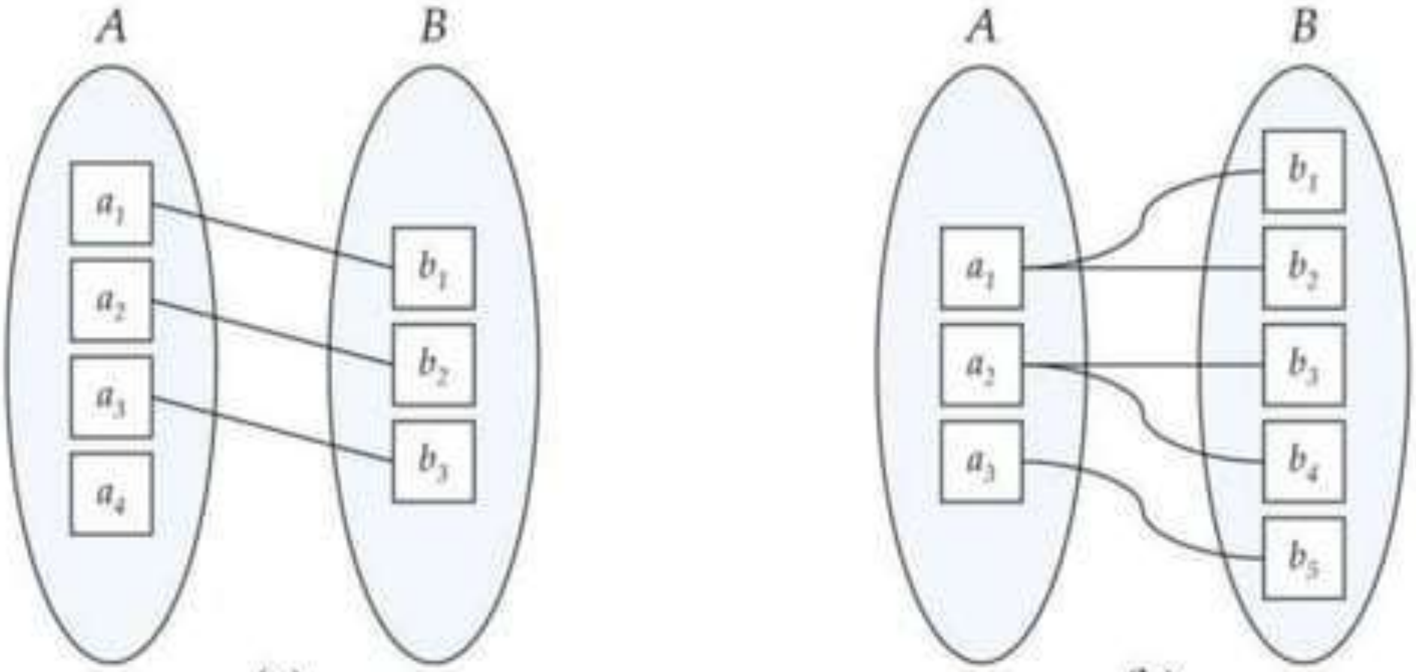


# ENTITY-RELATIONSHIP DIAGRAMS

## Types of Relationships

- Three types of relationships can exist between entities
- One-to-one relationship (1:1): One instance in an entity (parent) refers to one and only one instance in the related entity (child).
- One-to-many relationship (1:M): One instance in an entity (parent) refers to one or more instances in the related entity (child)

# Entity-Relationship Diagrams



(a)

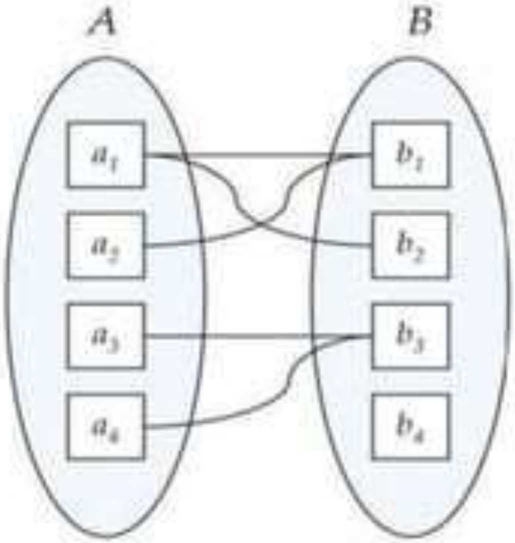
(b)

One to one

One to many

# ENTITY-RELATIONSHIP DIAGRAMS

- Types of Relationships
  - Many-to-many relationship (M:N): exists when one instance of the first entity (parent) can relate to many instances of the second entity (child), and one instance of the second entity can relate to many instances of the first entity.

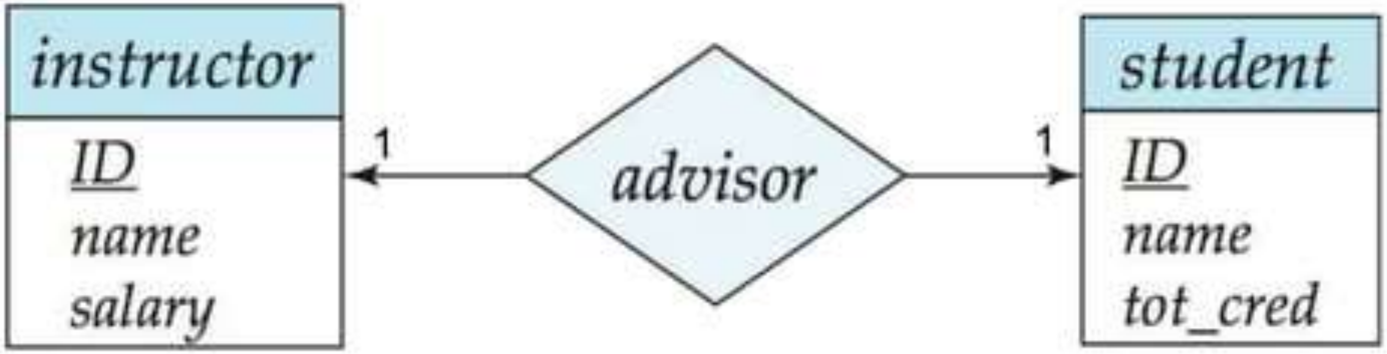


Many to many

# CARDINALITY CONSTRAINTS

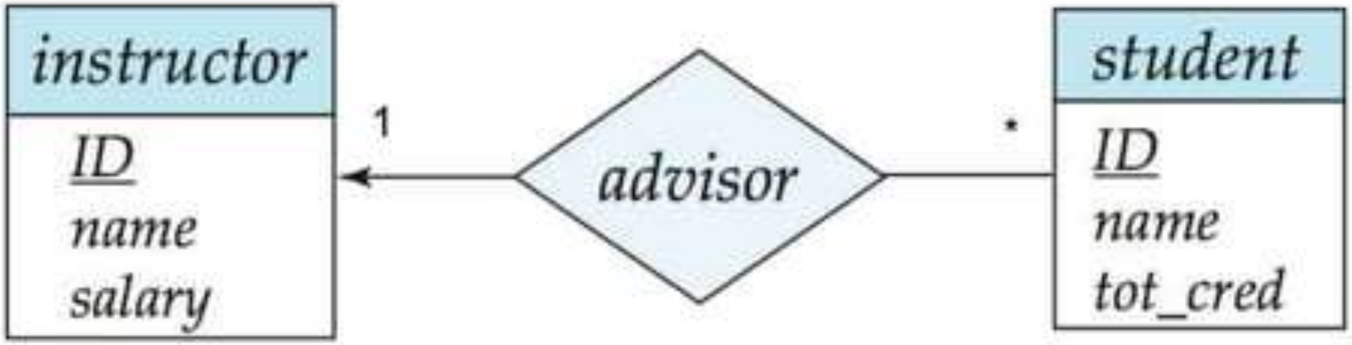
- We express cardinality constraints by drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.
- Or, by numbering each entity. * or, m for many.

- One-to-one relationship:
  - A student is associated with at most one *instructor* via the relationship *advisor*
  - A *student* is associated with at most one *department* via *stud_dept*



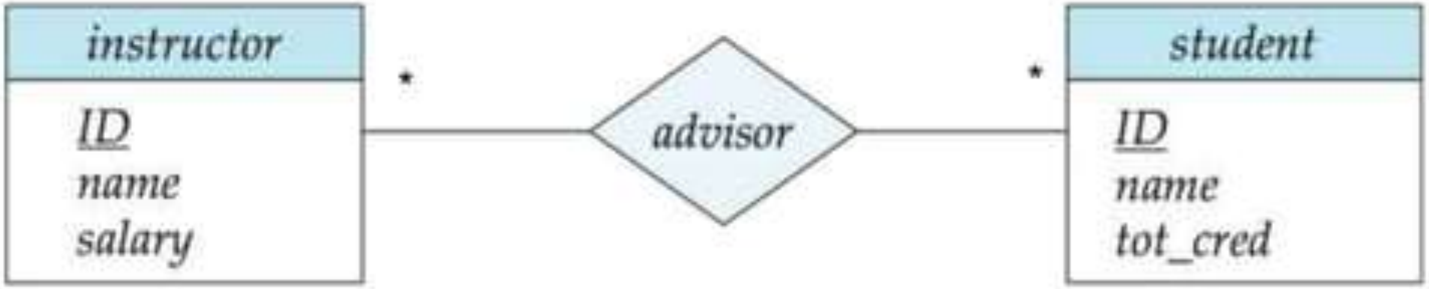
# ONE-TO-MANY RELATIONSHIP

- one-to-many relationship between an *instructor* and a *student*
  - an instructor is associated with several (including 0) students via *advisor*
  - a student is associated with at most one instructor via advisor,



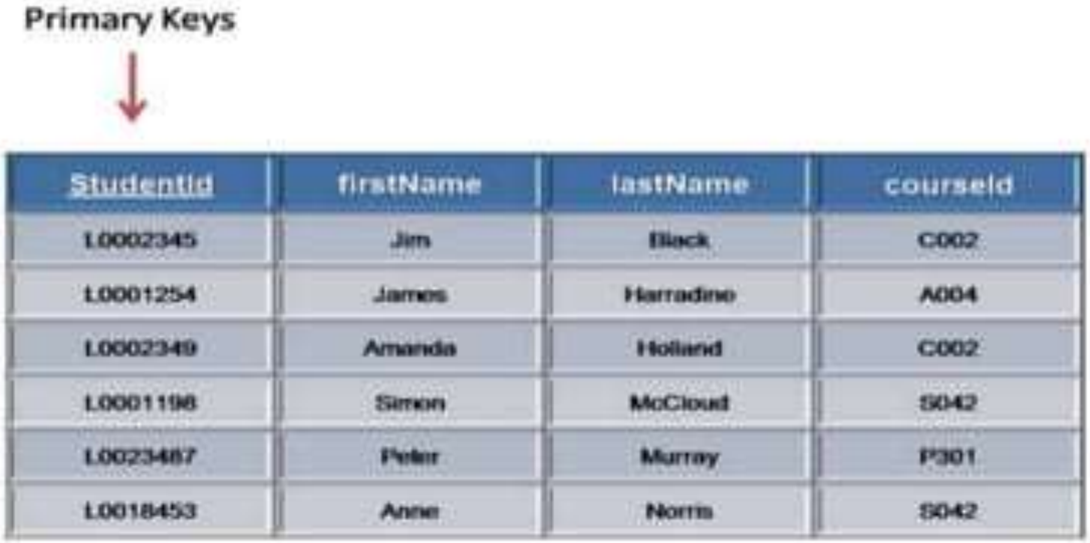
# MANY-TO-MANY RELATIONSHIP

- An instructor is associated with several (possibly 0) students via *advisor*
- A student is associated with several (possibly 0) instructors via *advisor*



# PRIMARY KEY

- A primary key is a candidate key that is most appropriate to be the main reference key for the table. As its name suggests, it is the primary key of reference for the table and is used throughout the database to help establish relationships with other tables.
- **The primary key must contain unique values, must never be null and uniquely identify each record in the table**

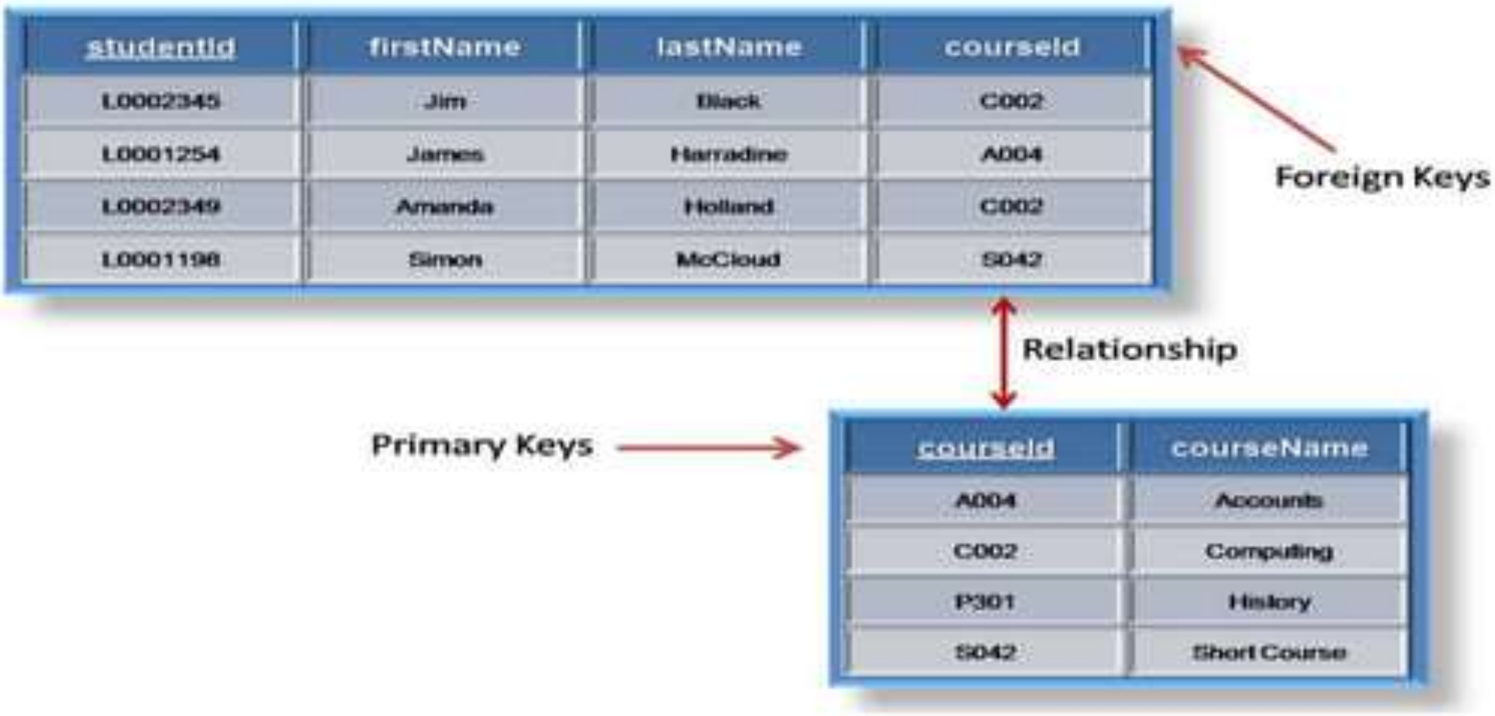


Primary Keys

| StudentId | firstName | lastName | courseId |
|---|---|---|---|
| L0002345 | Jim | Black | C002 |
| L0001254 | James | Harradine | A004 |
| L0002349 | Amanda | Holland | C002 |
| L0001198 | Simon | McCloud | S042 |
| L0023487 | Peter | Murray | P301 |
| L0018453 | Anne | Norris | S042 |

# Foreign Key

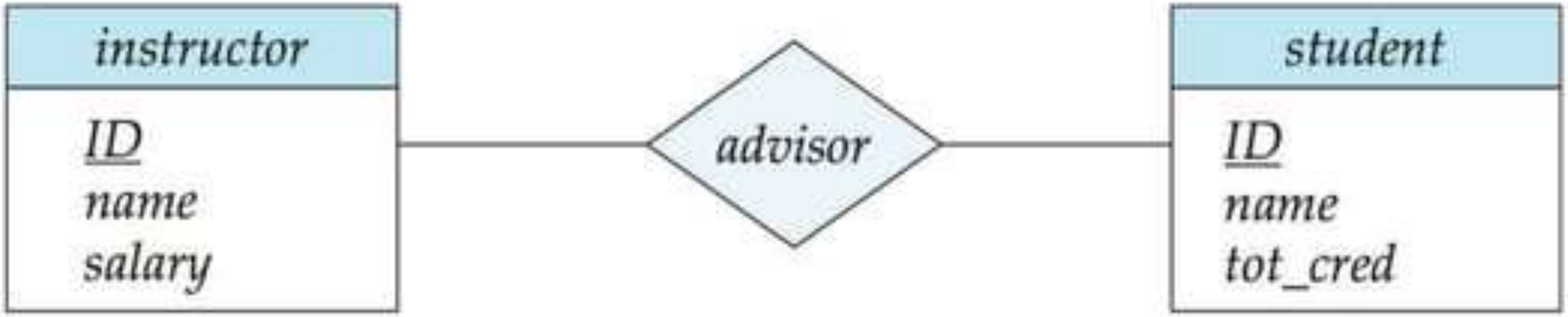
- A foreign key is generally a primary key from one table that appears as a field in another where the first table has a relationship to the second. In other words, if we had a table A with a primary key X that linked to a table B where X was a field in B, then X would be a foreign key in B



| studentId | firstName | lastName | courseId |
|---|---|---|---|
| L0002345 | Jim | Black | C002 |
| L0001254 | James | Harradine | A004 |
| L0002349 | Amanda | Holland | C002 |
| L0001198 | Simon | McCloud | S042 |

| courseId | courseName |
|---|---|
| A004 | Accounts |
| C002 | Computing |
| P301 | History |
| S042 | Short Course |

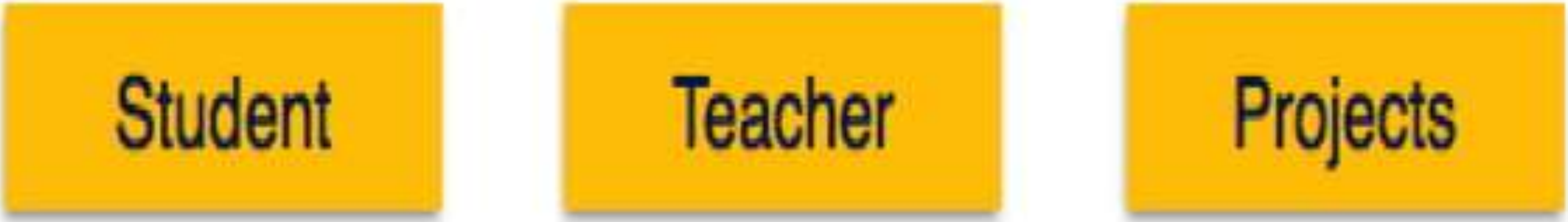
# E-R Diagrams



- Rectangles represent entity sets.
- Diamonds represent relationship sets.
- Attributes listed inside entity rectangle. Or , as oval shape along with the rectangle.
- Underline indicates primary key attributes

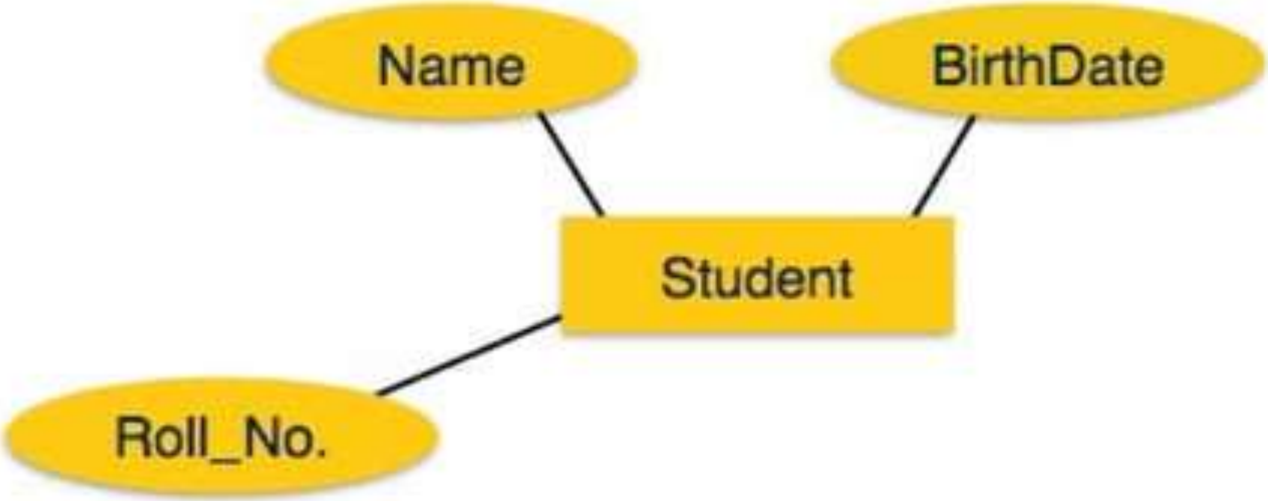
# Entity

- Entities are represented by means of rectangles. Rectangles are named with the entity set they represent.

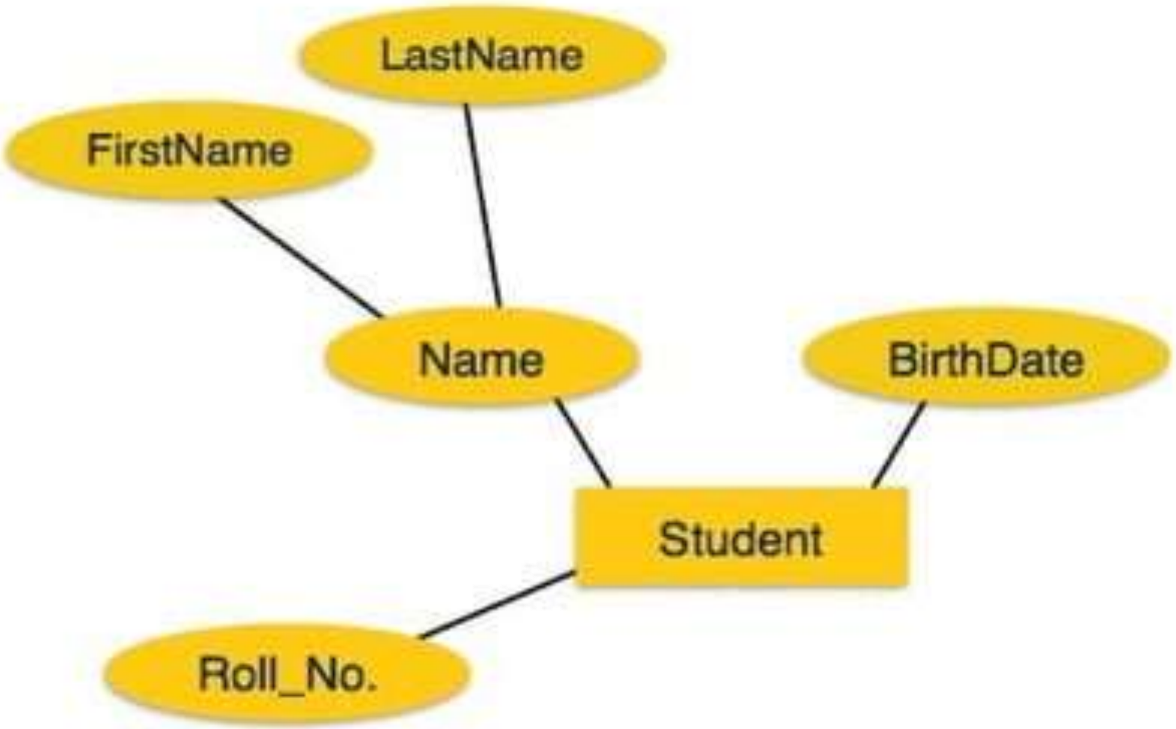


# Attributes

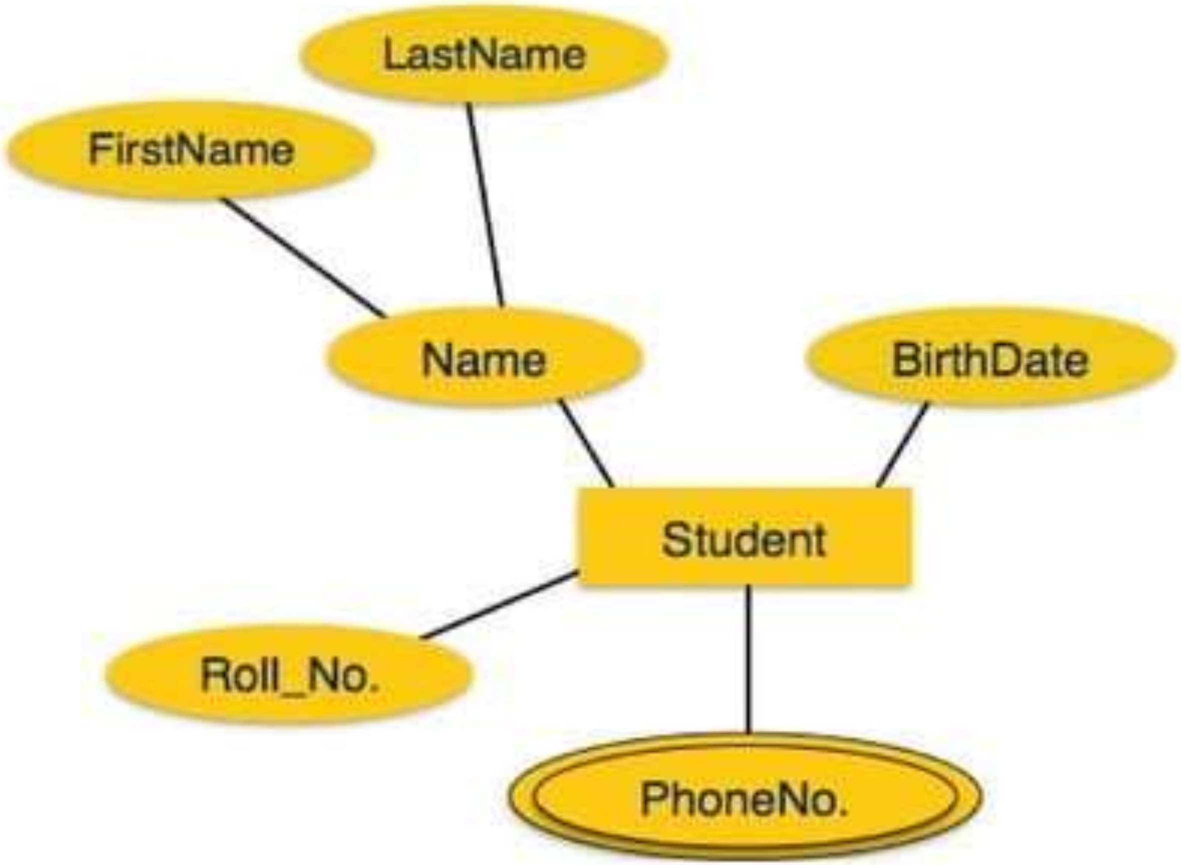
- Attributes are the properties of entities. Attributes are represented by means of ellipses.
- Every ellipse represents one attribute and is directly connected to its entity (rectangle).



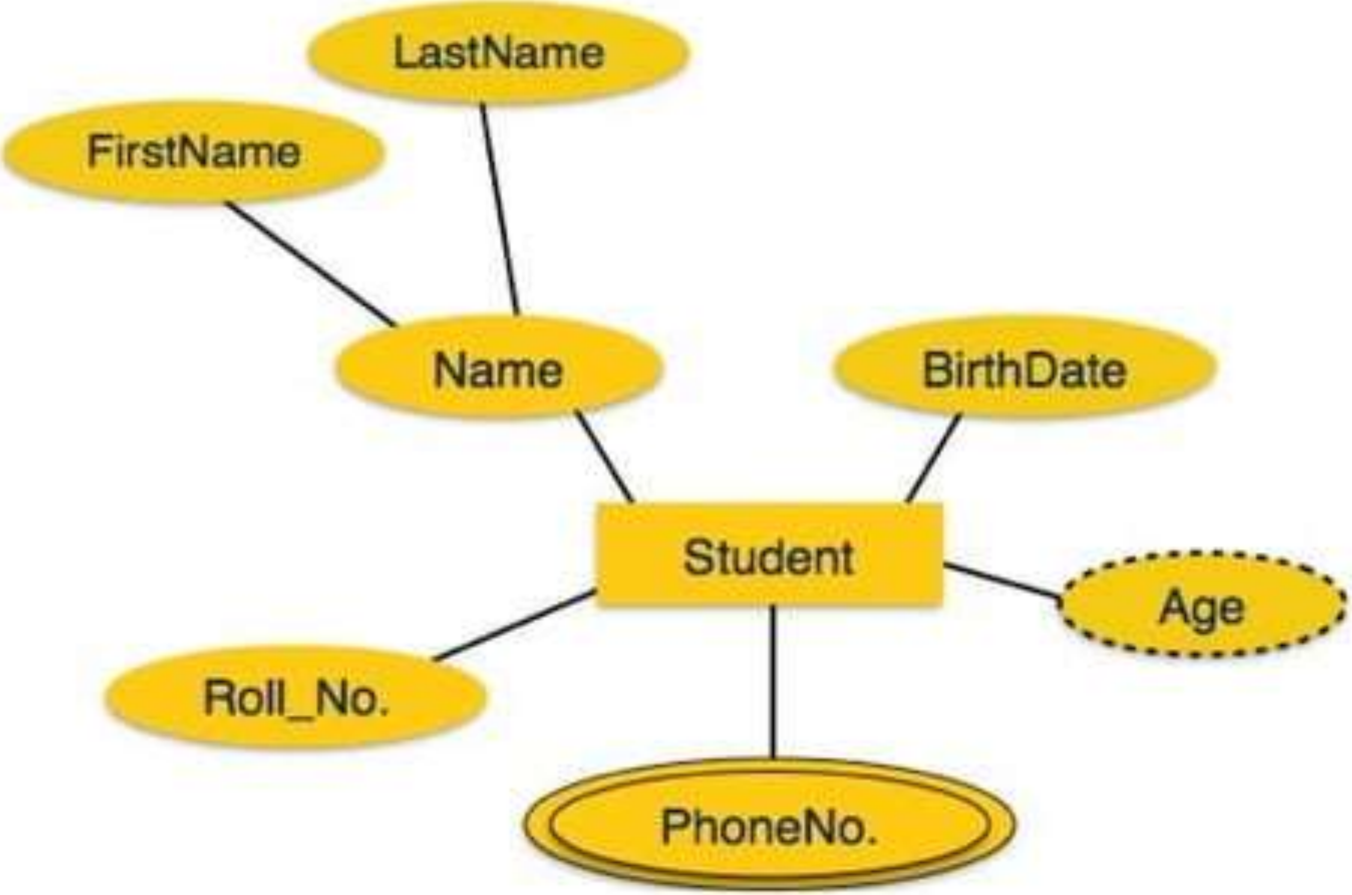
- If the attributes are **composite**, they are **further divided in a tree like structure**. Every node is then connected to its attribute. That is, composite attributes are represented by ellipses that are connected with an ellipse.



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

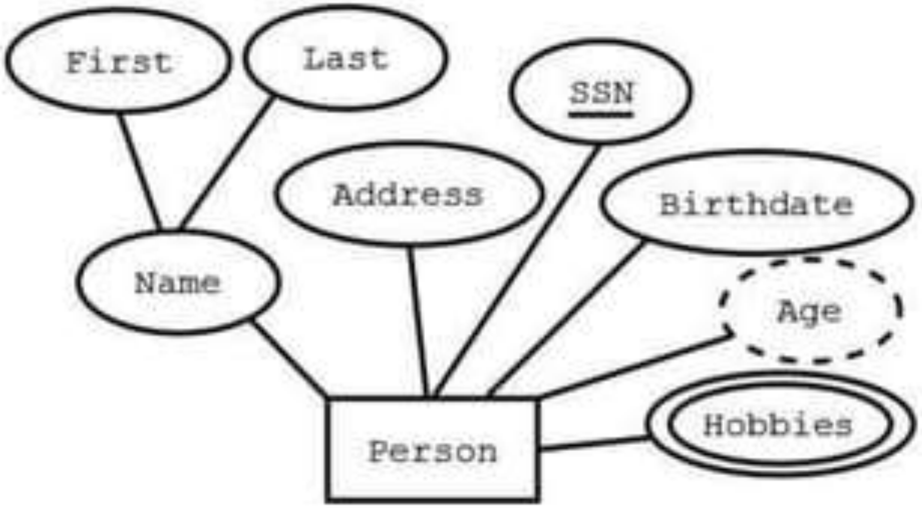


- **Derived** attributes are depicted by dashed ellipse.



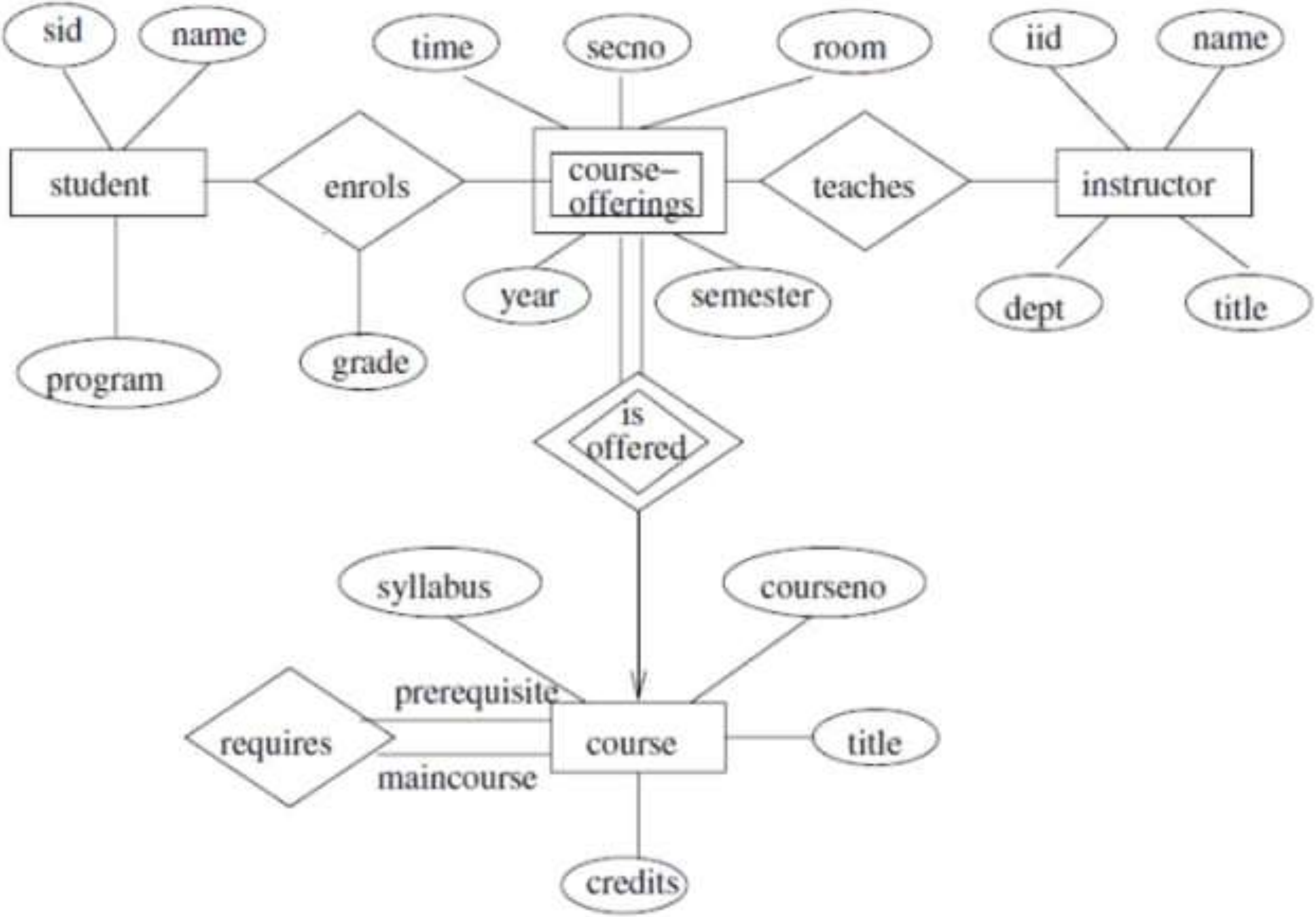
# SUMMARY OF SYMBOLS USED IN E-R NOTATION

- Representing attributes



- **Rectangle** -- Entity
- **Ellipses** -- Attribute (underlined attributes are [part of] the primary key)
- **Double ellipses** -- multi-valued attribute
- **Dashed ellipses**-- derived attribute, e.g. age is derivable from birthdate and current date.

- Question 2:
- A university registrar’s office maintains data about the following entities:
- courses, including number, title, credits, syllabus, and prerequisites;
- course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom;
- students, including student-id, name, and program;
- instructors, including identification number, name, department, and title.
- Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled. Construct an E-R diagram for the registrar’s office.Document all assumptions that you make about the mapping constraints.



E-R diagram for a university.

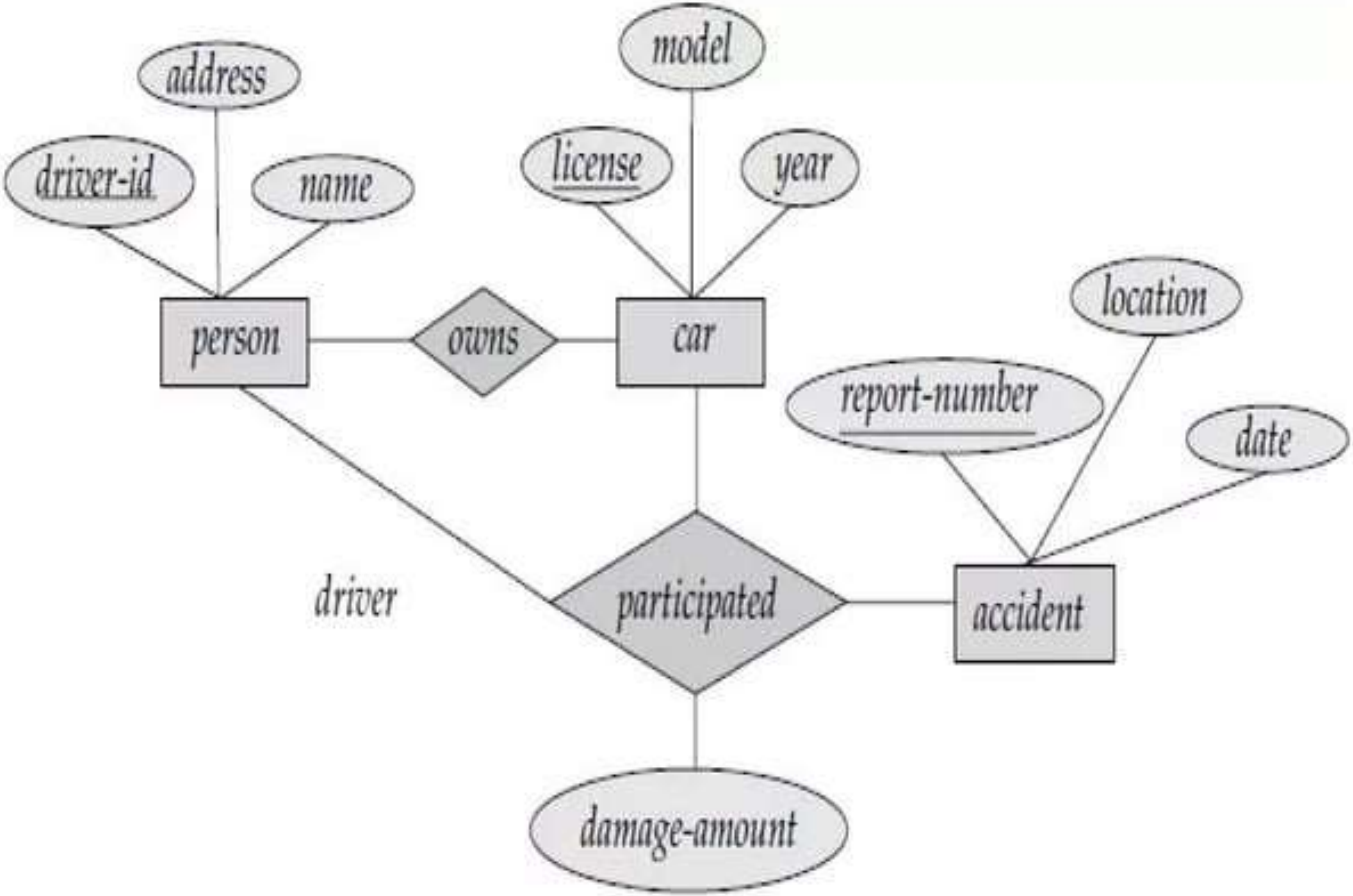
- Question 3:

(a) Construct an E-R diagram for a car-insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents.

(b) Construct appropriate tables for the above ER Diagram ?

**Car insurance tables:**

- person (driver-id, name, address)
- car (license, year,model)
- accident (report-number, date, location)
- participated(driver-id, license, report-number, damage-amount)



E-R diagram for a Car-insurance company.