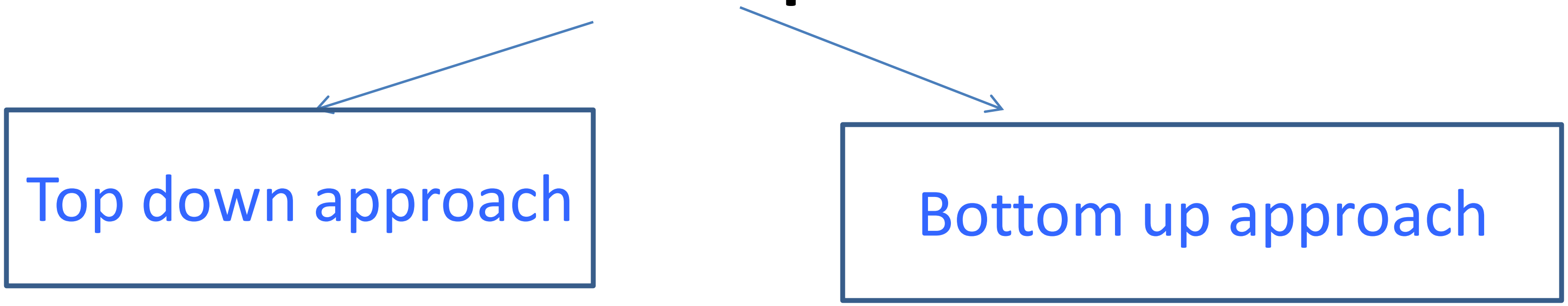
Database Implementation



Top down approach

Bottom up approach

Top down approach

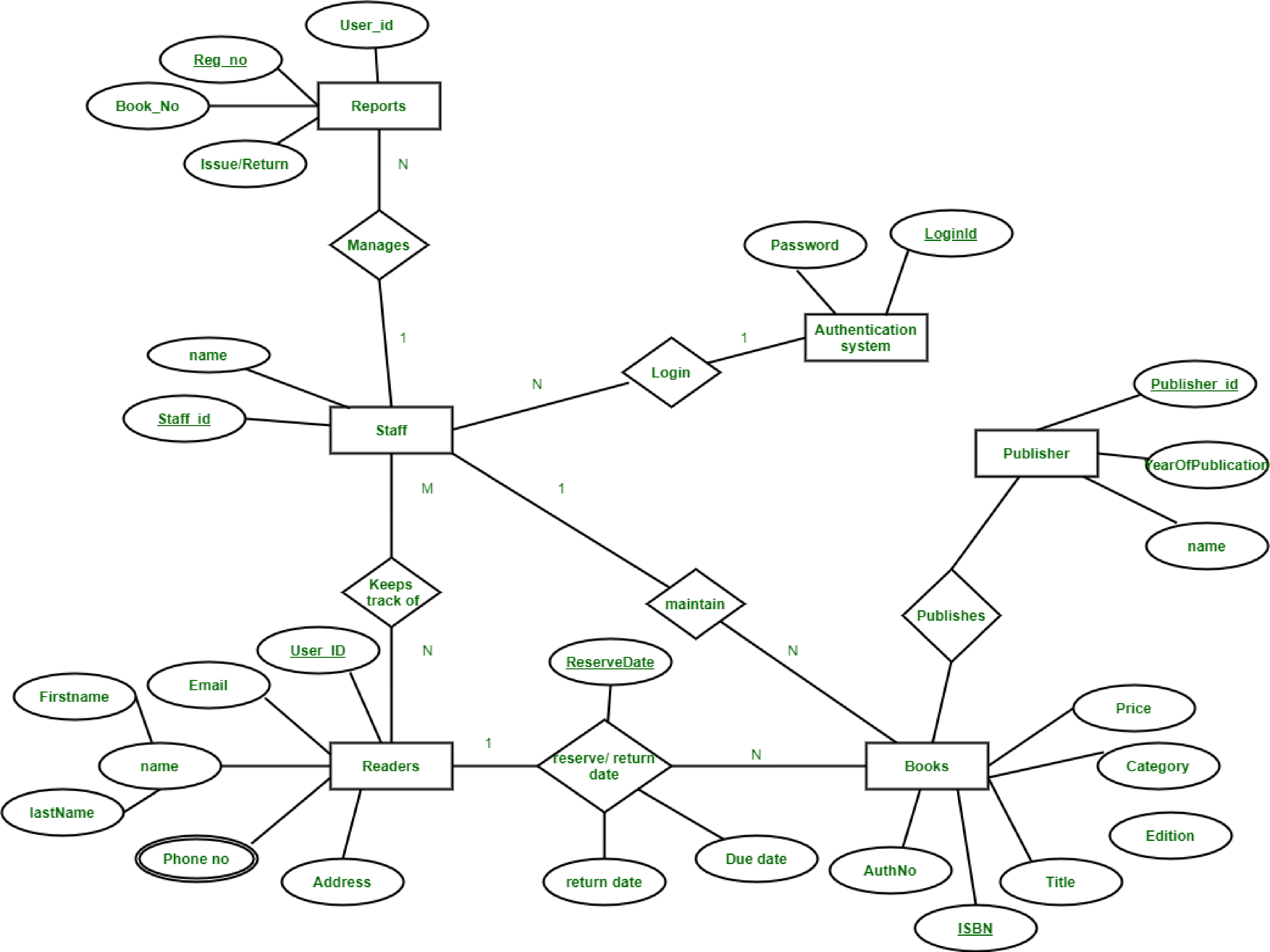
1. Requirements from customer
2. Database design – Logical Design :ER diagram
3. Implementation – Relational Model : SQL

Bottom up approach

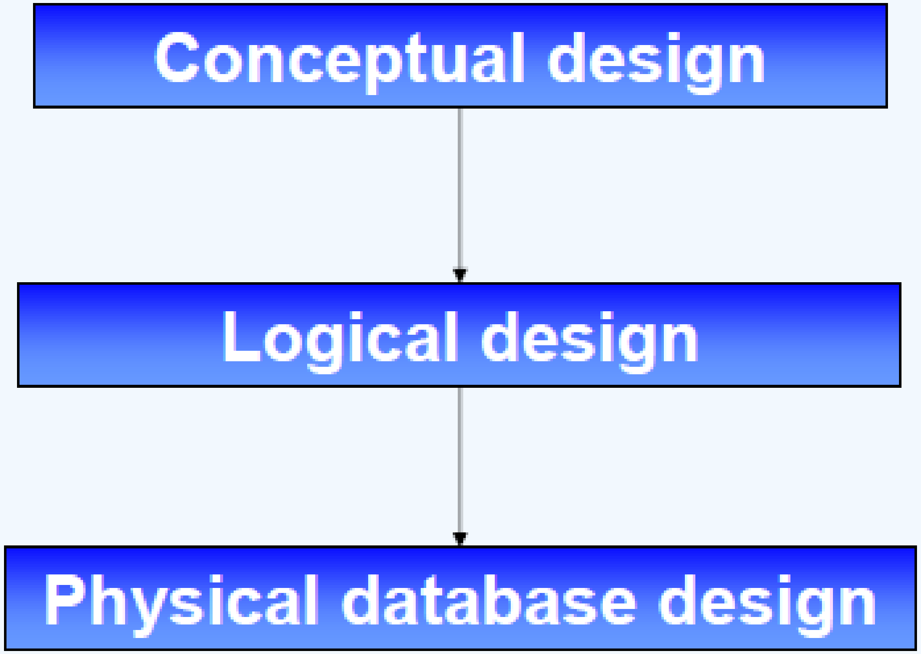
1. Requirements from customer
2. Database design – Logical Design :Normalization
3. Implementation – Relational Model : SQL

The Library Management System database keeps track of readers with the following Requirements

- The system keeps track of the staff with a single point authentication system comprising login Id and password.
- Staff maintains the book catalog with its ISBN, Book title, price(in INR), category(novel, general, story), edition, author Number and details.
- A publisher has publisher Id, Year when the book was published, and name of the book.
- Readers are registered with their user_id, email, name (first name, last name), Phone no (multiple entries allowed), communication address. The staff keeps track of readers.
- Readers can return/reserve books that stamps with issue date and return date. If not returned within the prescribed time period, it may have a due date too.
- Staff also generate reports that has readers id, registration no of report, book no and return/issue info.


User_id
Reg_no
Book_No
Reports
Issue/Return
N
Manages
Password
LoginId
1
Authentication system
1
name
N
Login
Staff_id
Staff
Publisher_id
Publisher
YearOfPublication
M
1
name
Keeps track of
maintain
Publishes
User_ID
N
N
ReserveDate
Email
Firstname
Price
name
Readers
1
reserve/ return date
N
Books
Category
lastName
Edition
Phone no
Due date
AuthNo
Address
return date
Title
ISBN

Design levels of a database



ER modeling

- **ER modeling** : A graphical technique for *understanding* and organizing the data independent of the actual database implementation

- **Entity:** Any thing that may have an independent existence and about which we intend to collect data.

  Also known as **Entity type.**

- **Entity instance:** a particular member of the entity type e.g. a particular student

- **Attributes:** Properties/characteristics that describe entities

- **Relationships:** Associations between entities

Attributes

- The set of possible values for an attribute is called the **domain** of the attribute

  Example:

  - The domain of attribute ***marital status*** is just the four values: single, married, divorced, widowed

  - The domain of the attribute month is the twelve values ranging from January to December

- **Key attribute**: The attribute (or combination of attributes) that is unique for every entity instance
  - E.g the account number of an account, the employee id of an employee etc.

- If the key consists of two or more attributes in combination, it is called a **composite key**

Simple Vs composite attribute

- **Simple attribute**: cannot be divided into simpler components

  E.g age of an employee

- **Composite attribute**: can be split into components

  E.g Date of joining of the employee.

  » Can be split into day, month and year

Single Vs Multi-valued Attributes

- **Single valued** : can take on only a single value for each entity instance

  E.g. ***age*** of employee. There can be only one value for this

- **Multi-valued**: can take many values

  E.g. ***skill set*** of employee

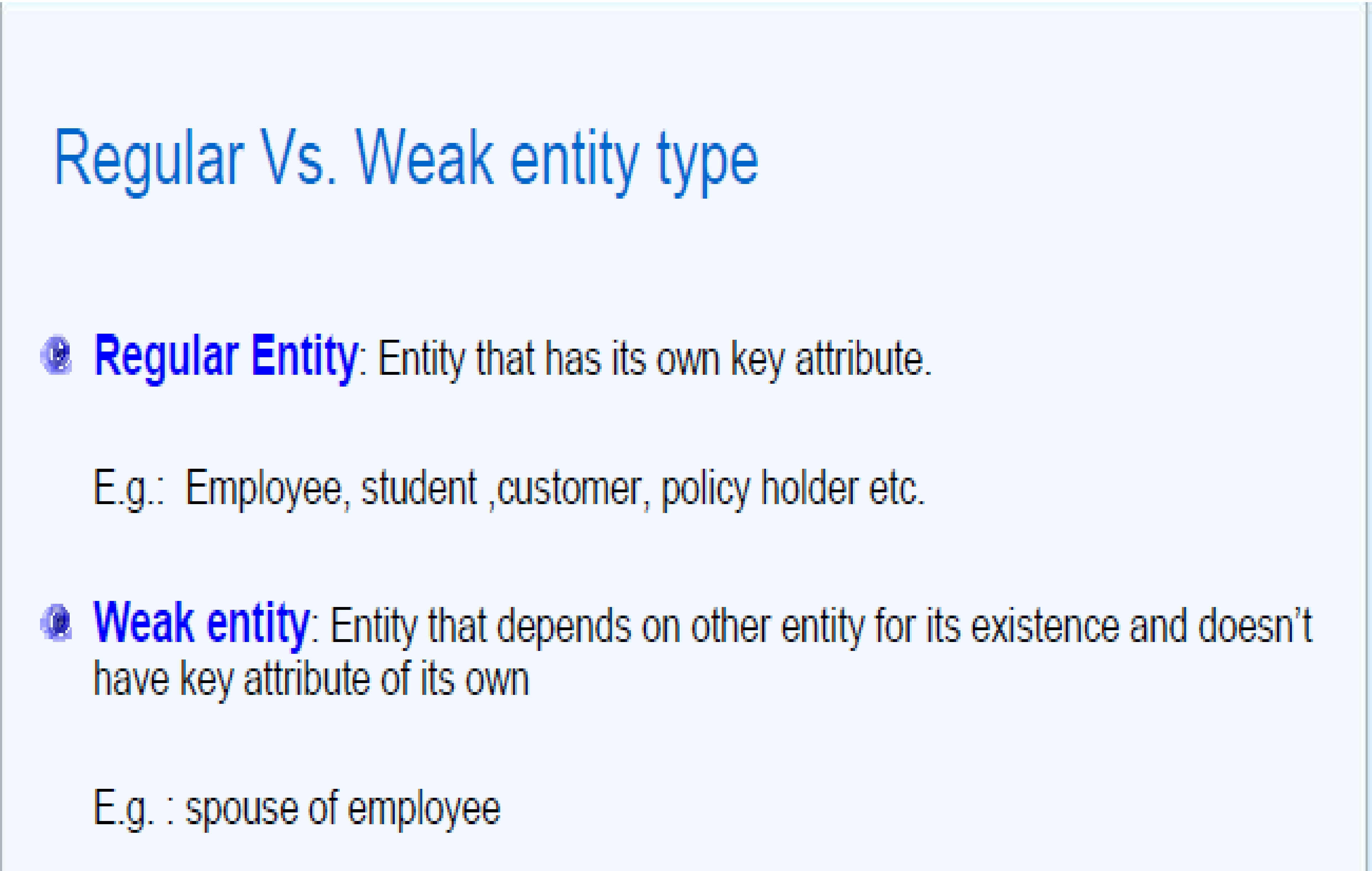
Stored Vs Derived attribute

- **Stored Attribute**: Attribute that need to be stored permanently.
    - E.g. ***name*** of an employee
- **Derived Attribute**: Attribute that can be calculated based on other attributes

.

- E.g. : ***years of service*** of employee can be calculated from date of joining and current date

Two types of entities : 1. Strong (or) Regular entity
2. Weak entity



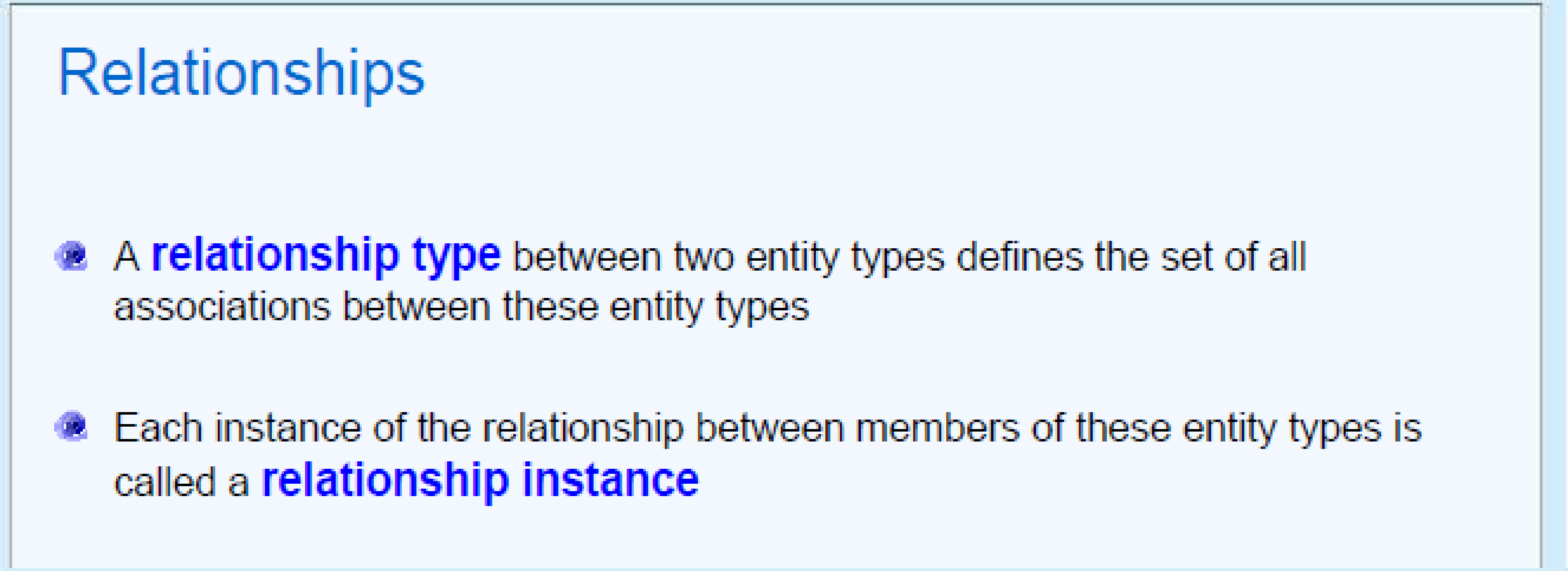
Regular Vs. Weak entity type

- **Regular Entity**: Entity that has its own key attribute.

  E.g.: Employee, student ,customer, policy holder etc.

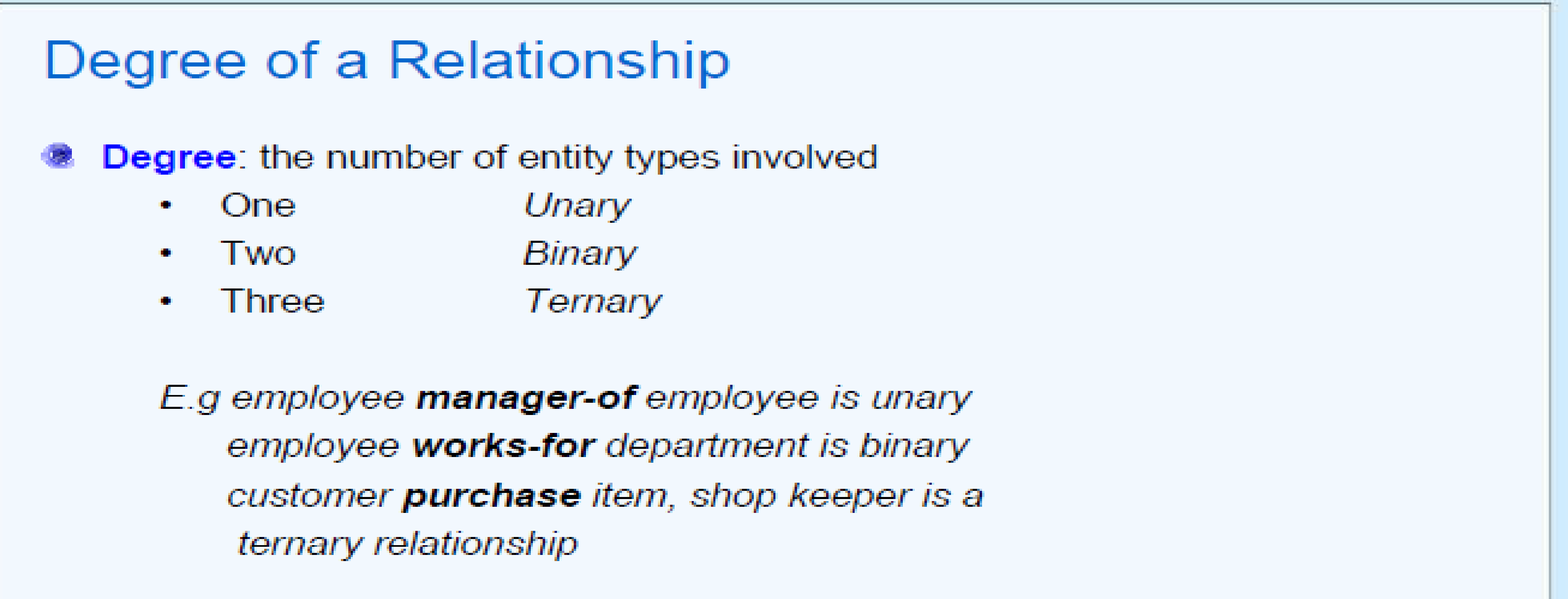
- **Weak entity**: Entity that depends on other entity for its existence and doesn't have key attribute of its own

  E.g. : spouse of employee



Relationships

- A **relationship type** between two entity types defines the set of all associations between these entity types
- Each instance of the relationship between members of these entity types is called a **relationship instance**



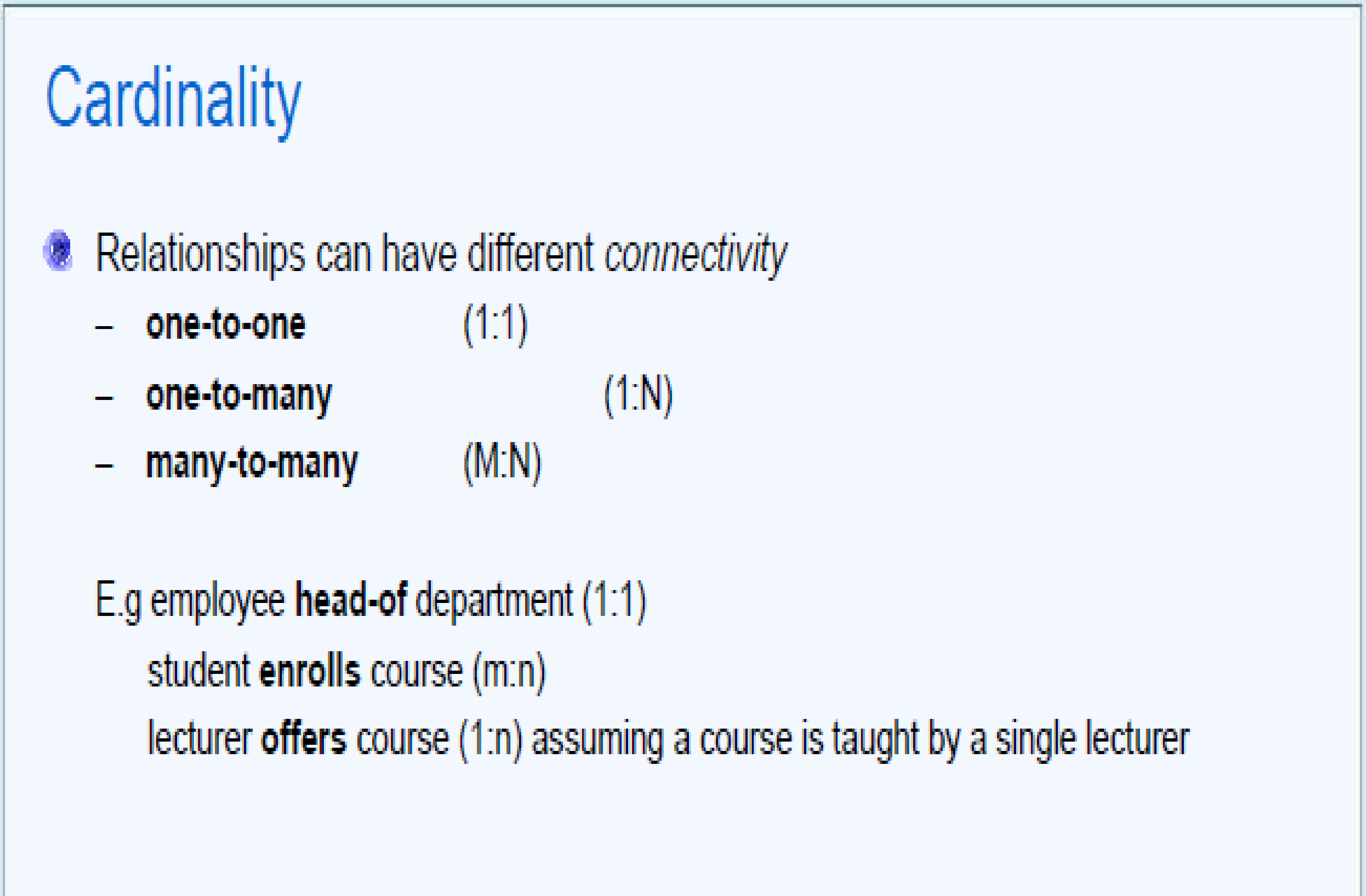
Degree of a Relationship

- **Degree**: the number of entity types involved

• One	*Unary*
• Two	*Binary*
• Three	*Ternary*

*E.g employee **manager-of** employee is unary*
*employee **works-for** department is binary*
*customer **purchase** item, shop keeper is a ternary relationship*

The minimum and maximum values of this connectivity is called the **cardinality of the relationship**



Cardinality

- Relationships can have different *connectivity*
  - **one-to-one** (1:1)
  - **one-to-many** (1:N)
  - **many-to-many** (M:N)

E.g employee **head-of** department (1:1)

student **enrolls** course (m:n)

lecturer **offers** course (1:n) assuming a course is taught by a single lecturer

Relationship Participation

- **Total** : Every entity instance must be connected through the relationship to another instance of the other participating entity types
- **Partial**: All instances need not participate

E.g Employee **Head-of** Department

Employee: partial

Department: total

Copyright © 2004,
Infosys Technologies Ltd

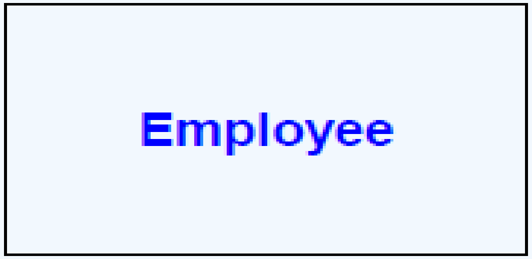
An **entity is a “thing” or “object” in the real world that is distinguishable from** all other objects.

- An **entity set is a set of entities of the same type that share the same properties,** or attributes.

ER modeling -Notations

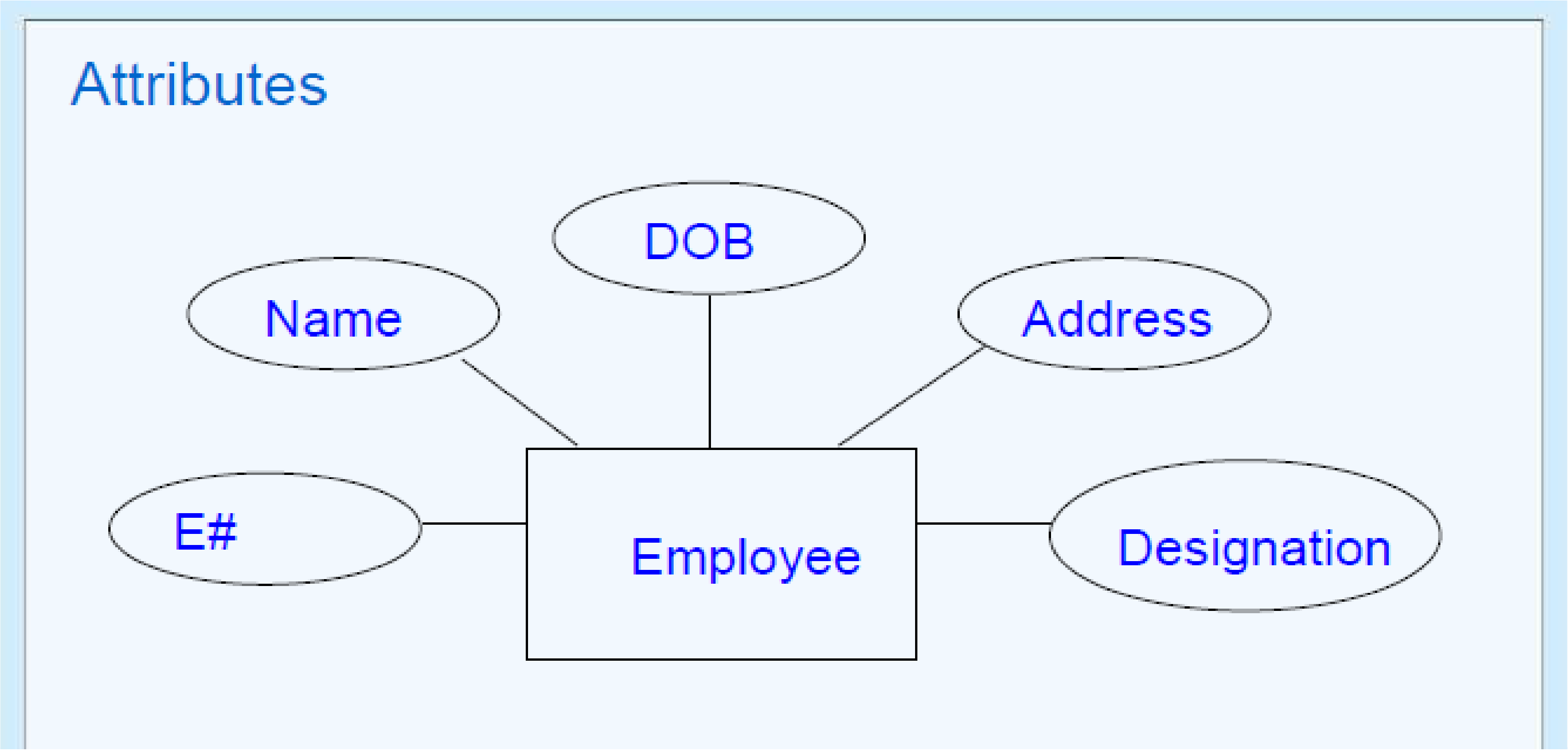
Entity

- Usually a noun in singular
- Represented by a rectangle with a label
- First letter in capitals

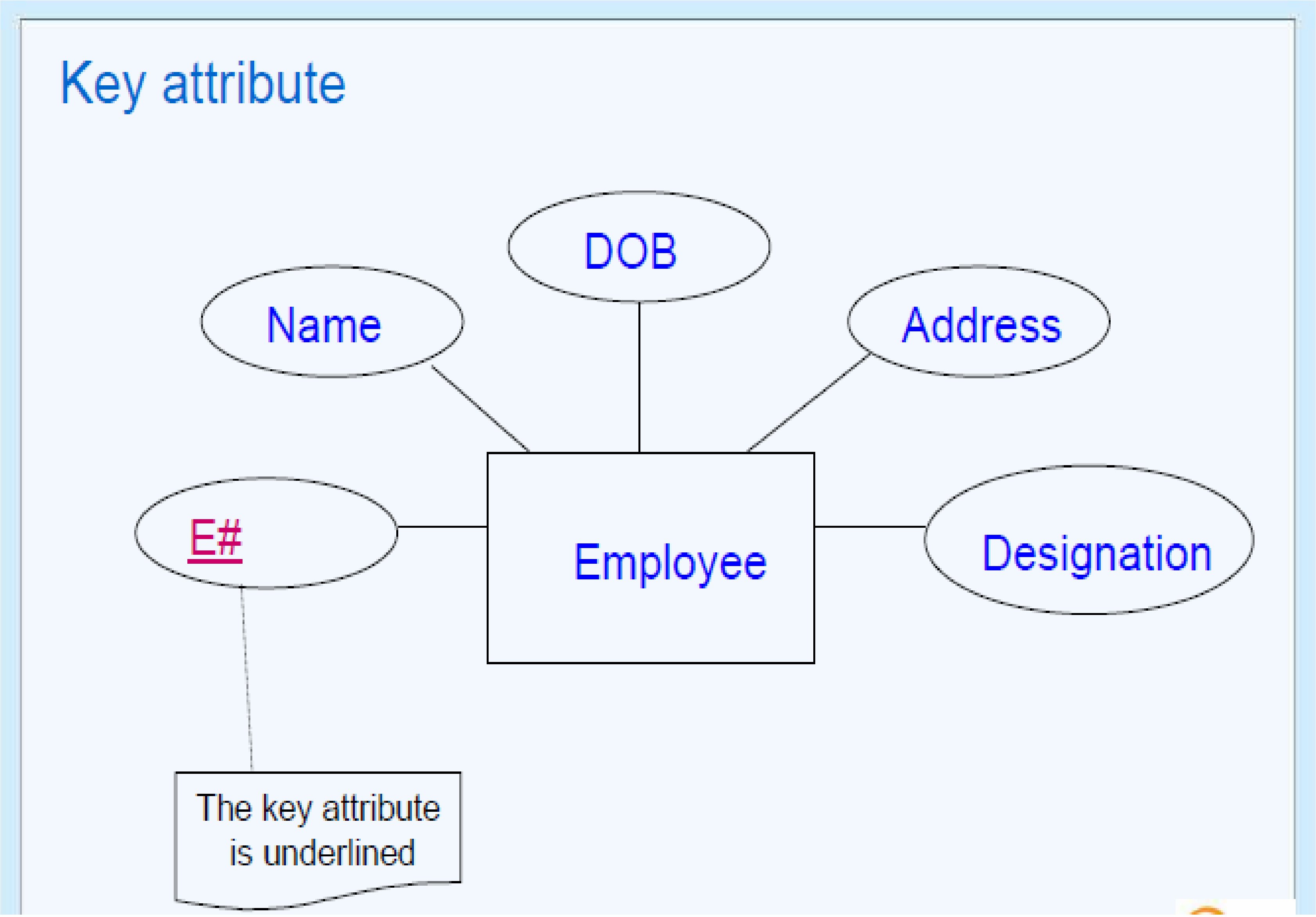


Employee

Attributes : Represented by ellipses connected to the entity type by straight lines

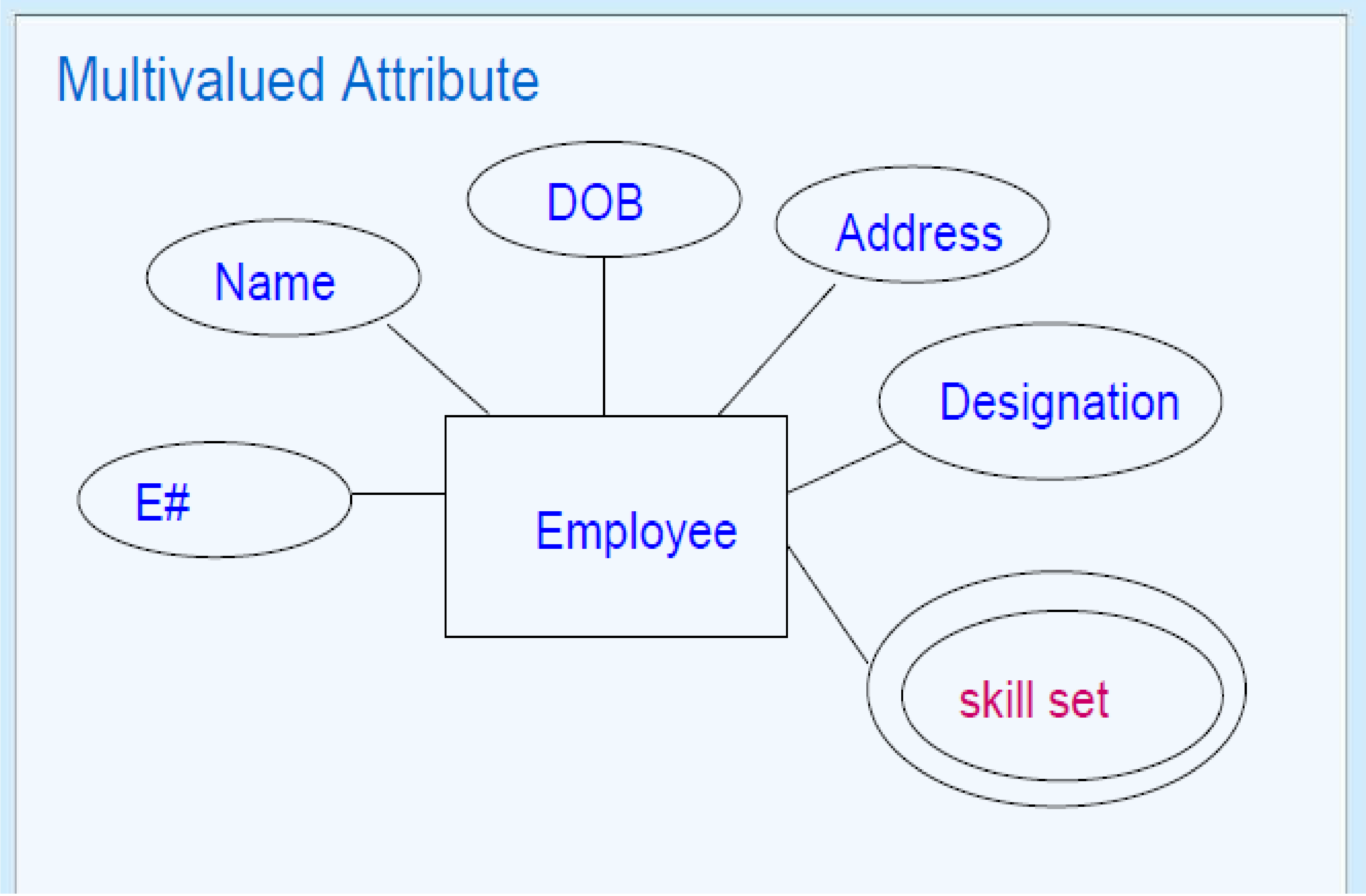


Key attribute

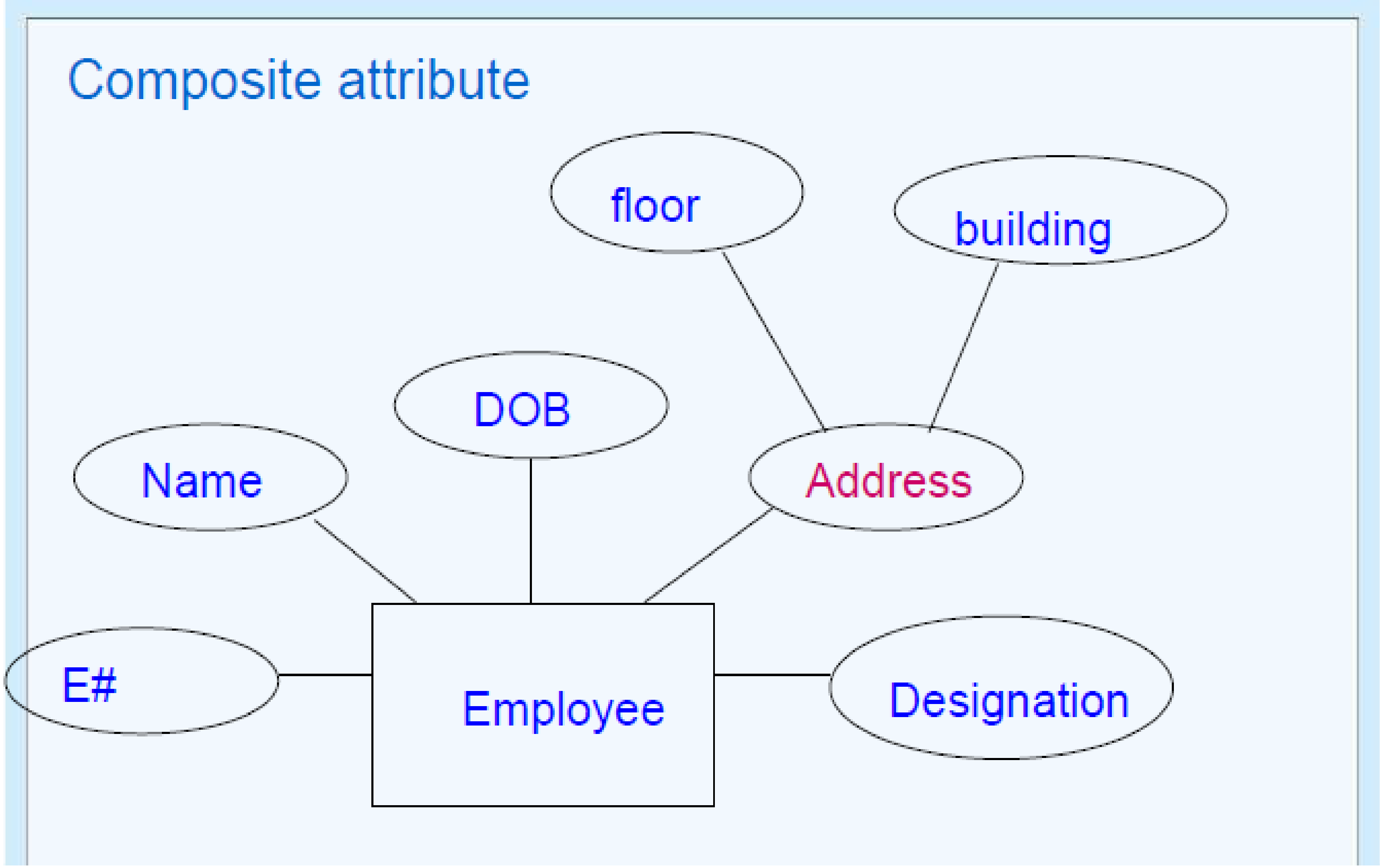


The key attribute is underlined

Multivalued Attribute : Indicated by a double lined ellipse

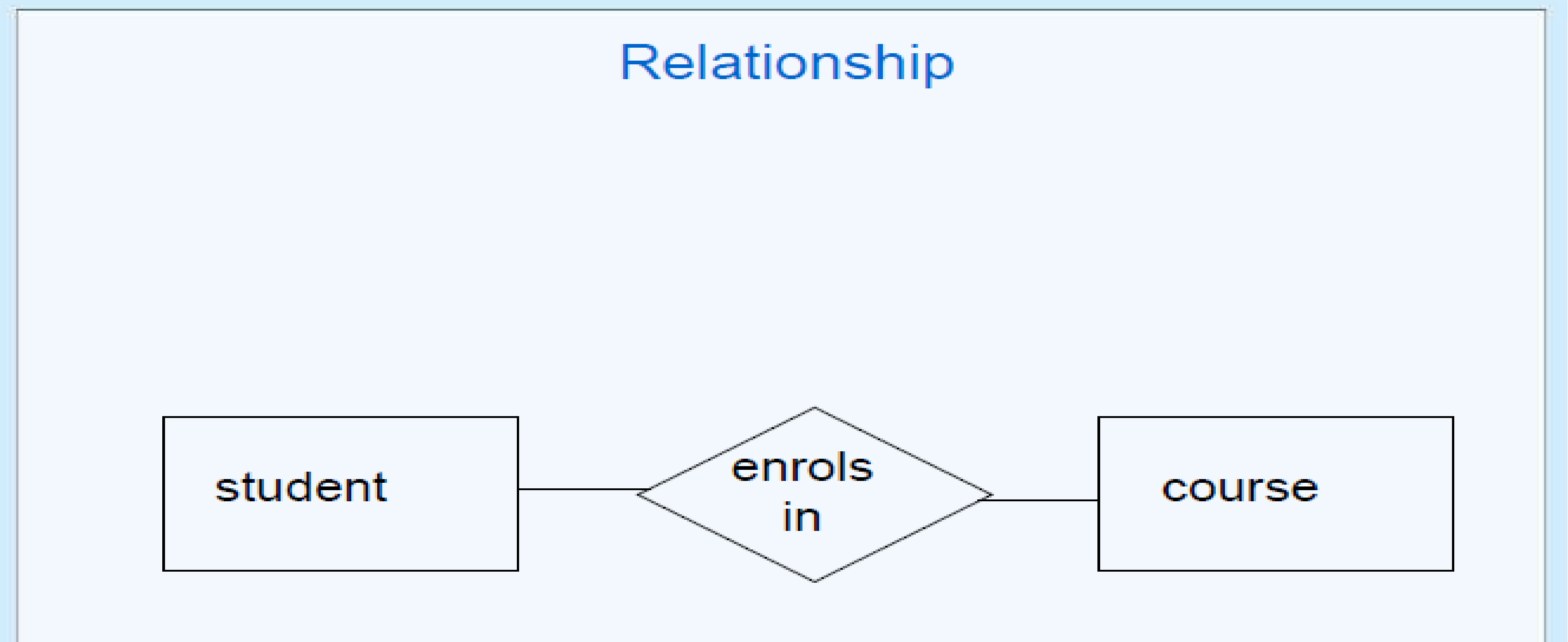


Composite Attribute : Represented by an ellipse from which other ellipses emanate and represent the component attributes. E.g Address

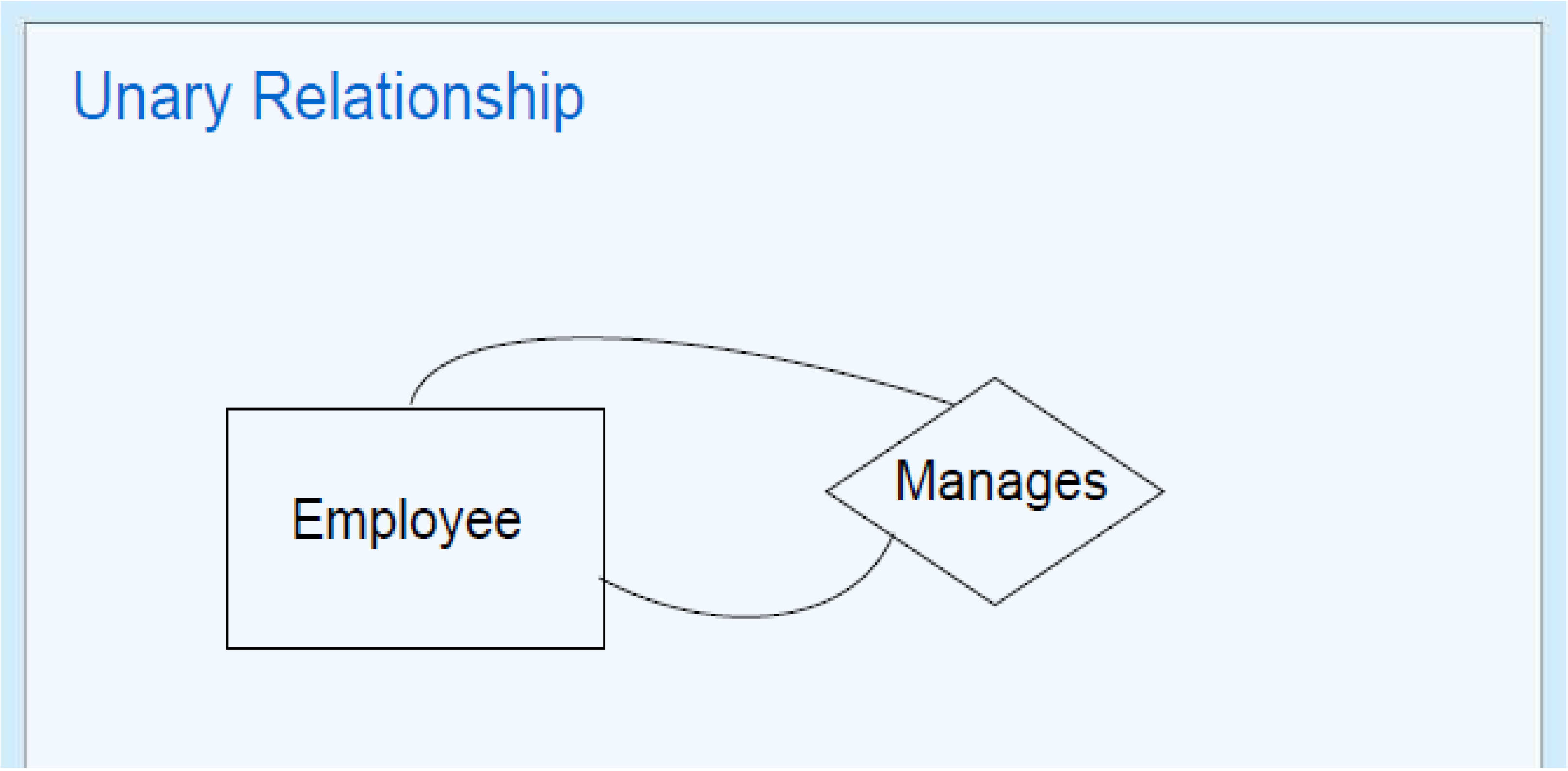


Relationship :

- •A relationship is represented as a diamond between two entity types.
- •It has a label that explains the relationship. Usually the convention is to read the ER diagram from top to bottom and from left to right.
- •So, the relationship name is so chosen as to make sense when read from left to right.
- •The relationship above is read as student enrolls-in course

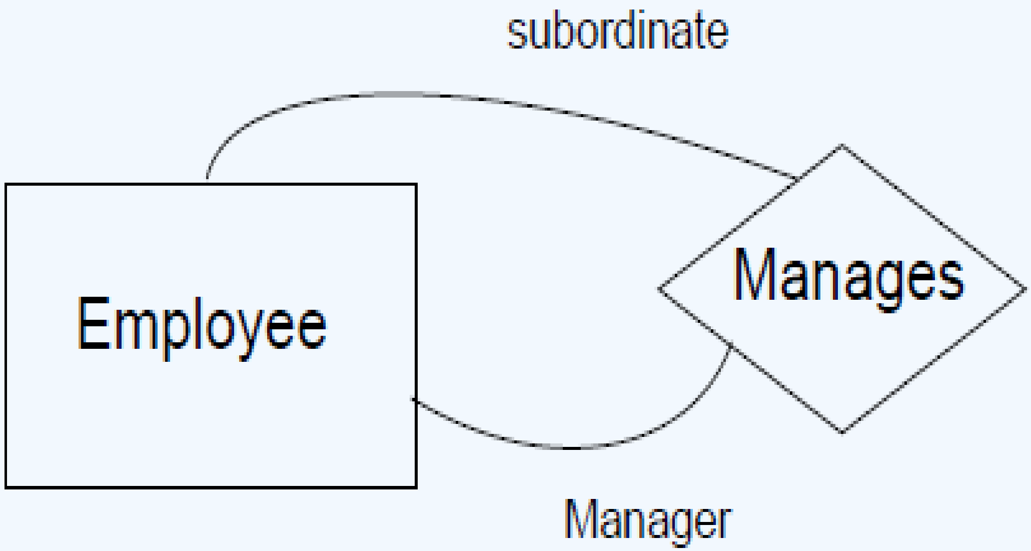


- A **unary relationship** is represented as a diamond which connects one entity to itself as a loop.
- The relationship above means, some instances of employee manage other instances of Employee.

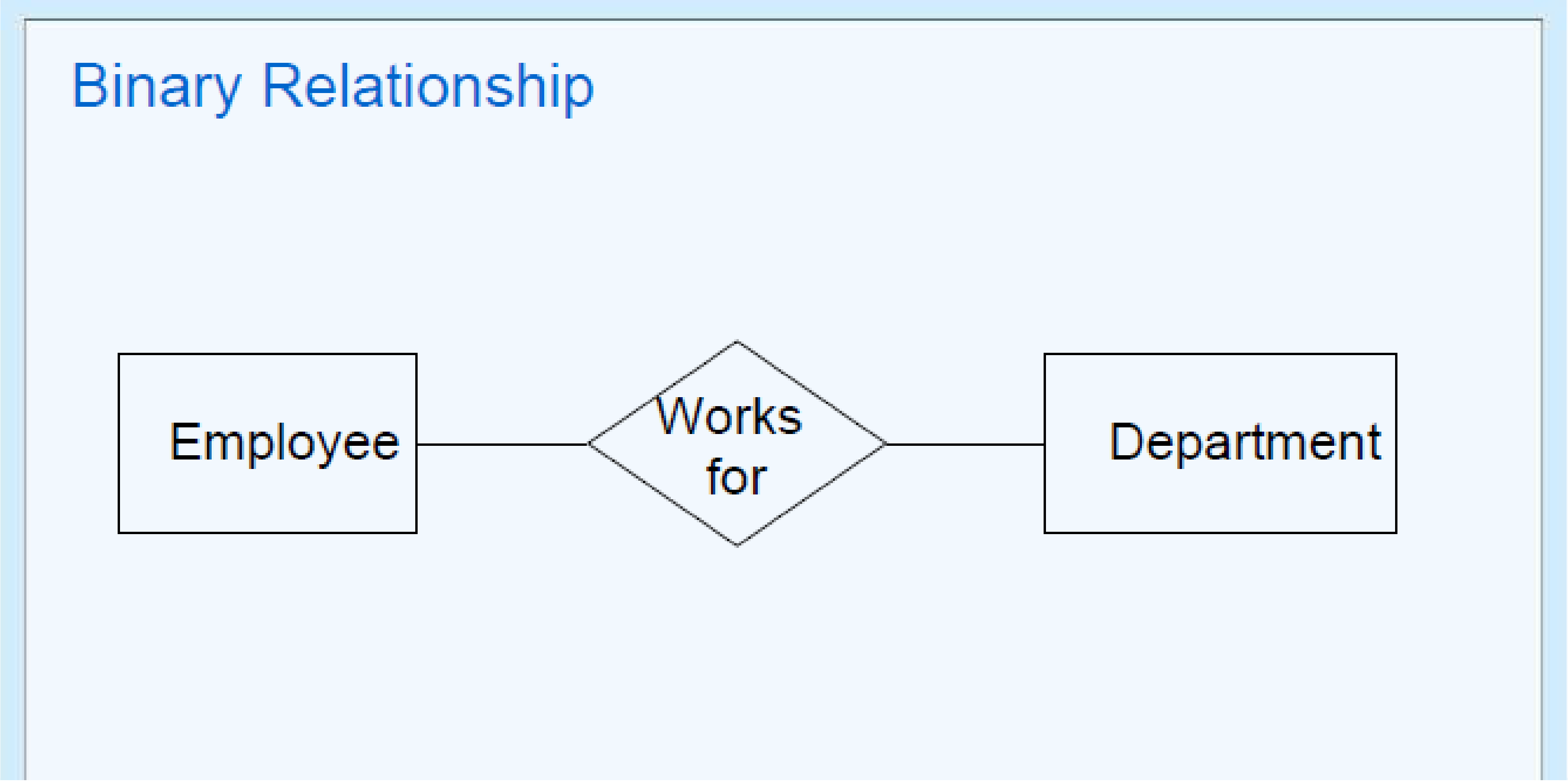


Role names

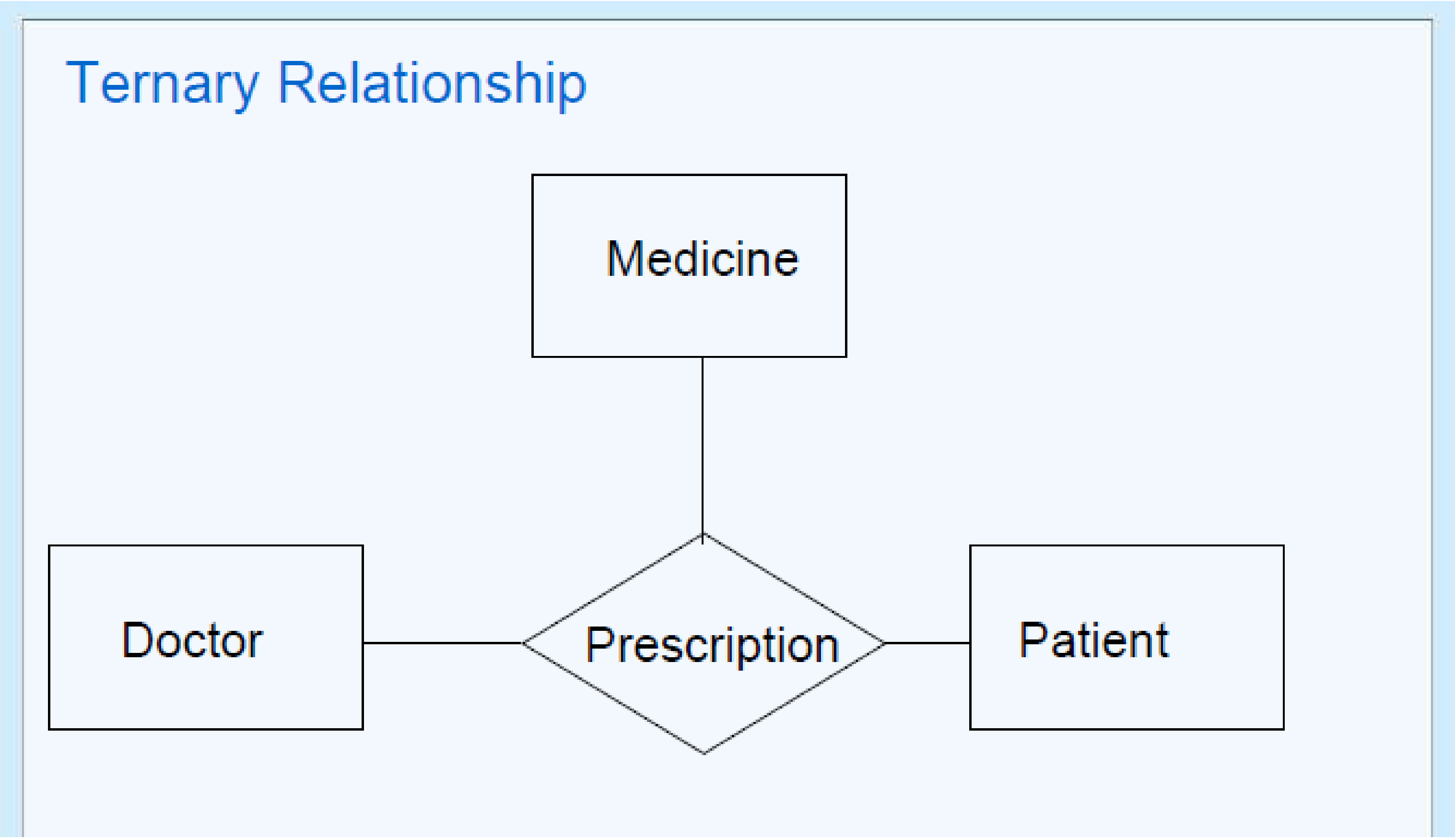
- Role names may be added to make the meaning more explicit



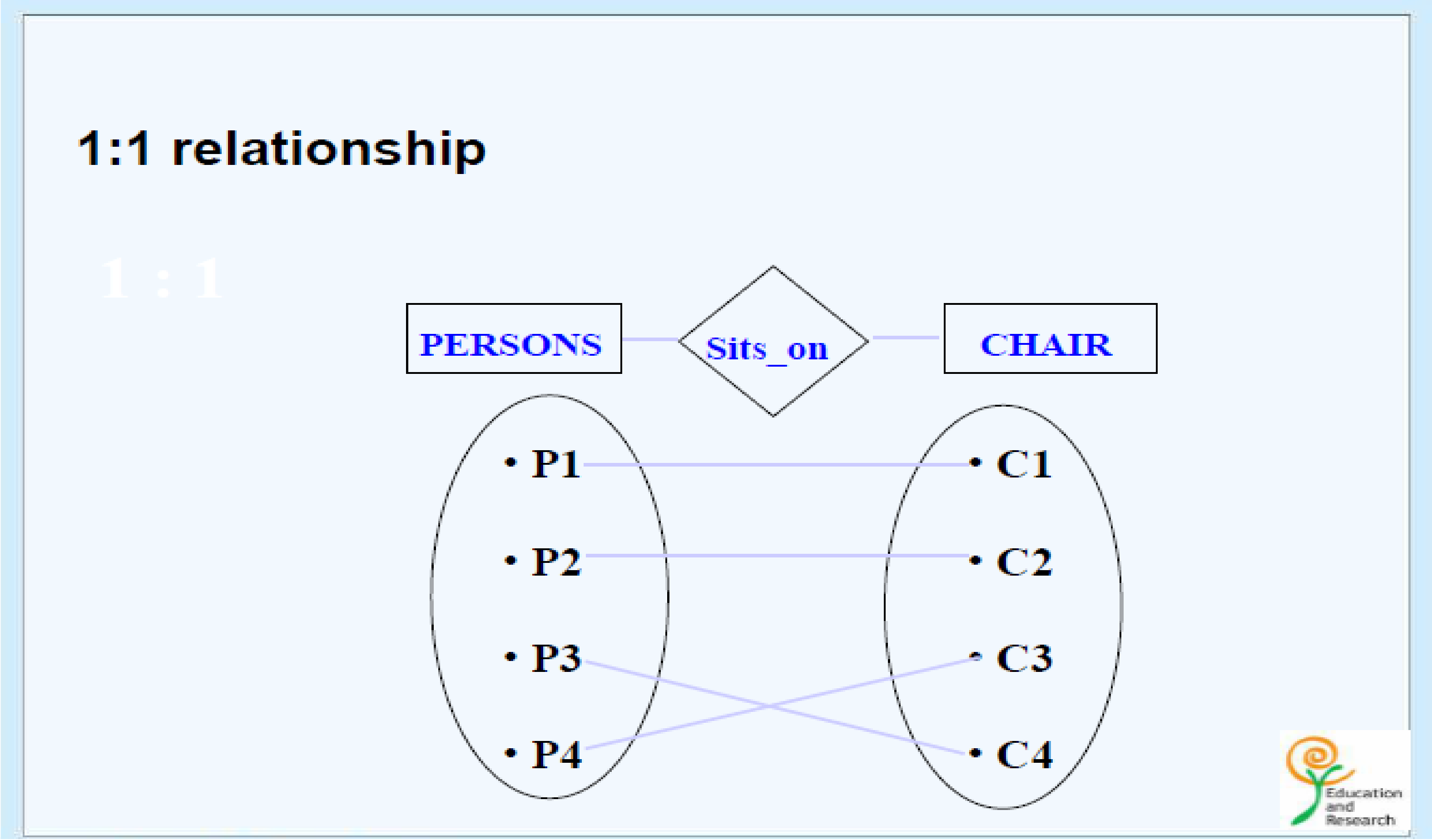
A **binary relationship** exists between two entity types



A **ternary relationship** is connecting three entity types.



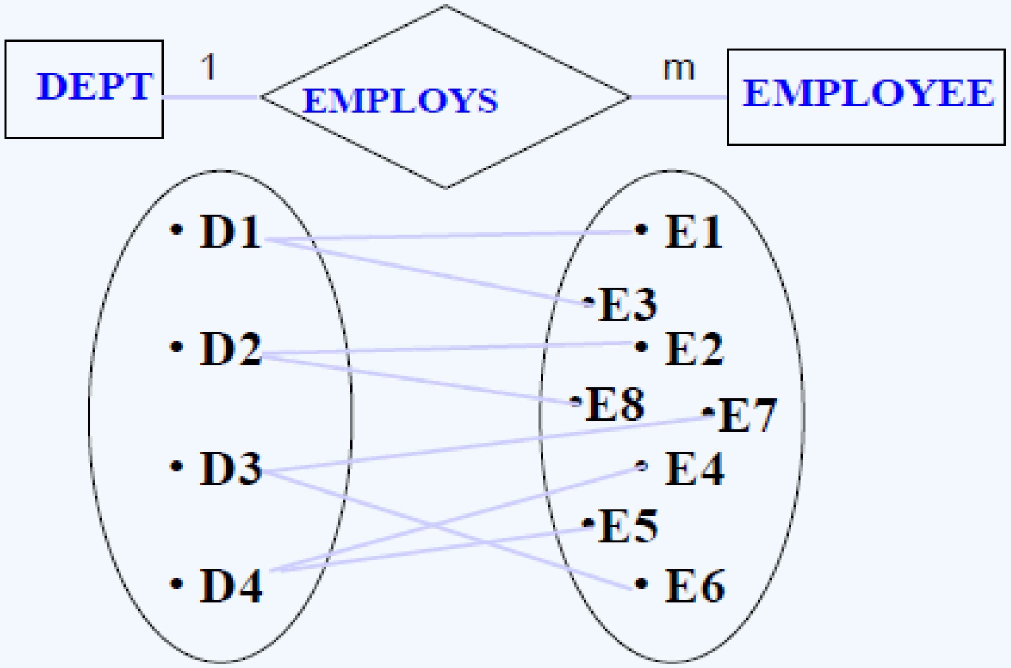
One instance of entity type 1 is connected to exactly one other entity instance of entity type 2



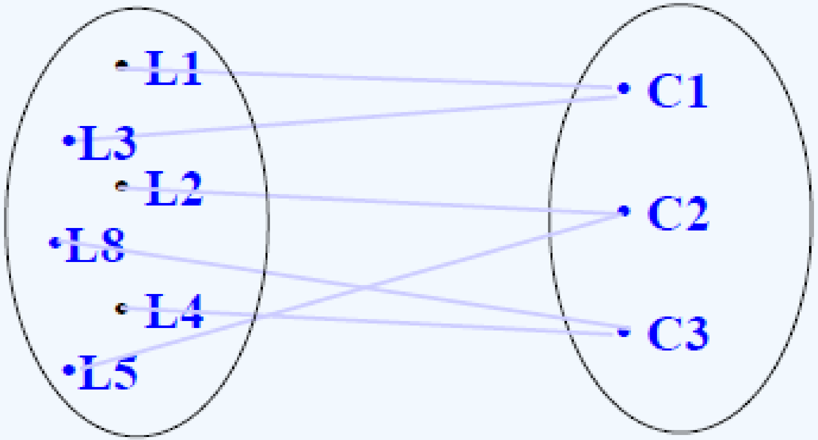
1:1 relationship

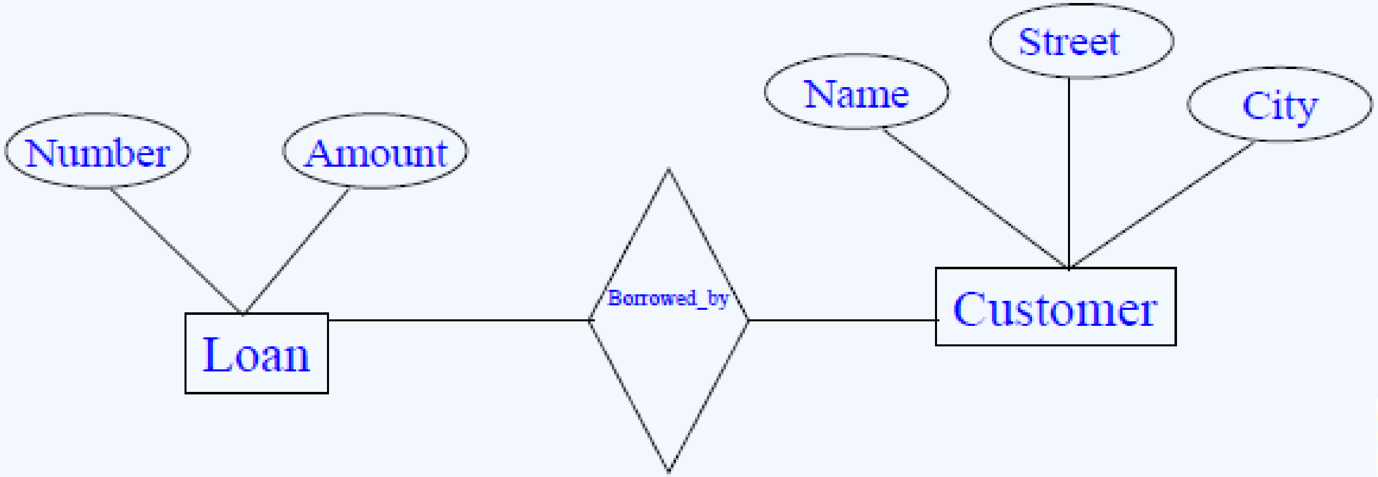
1:m relationship

1 : M



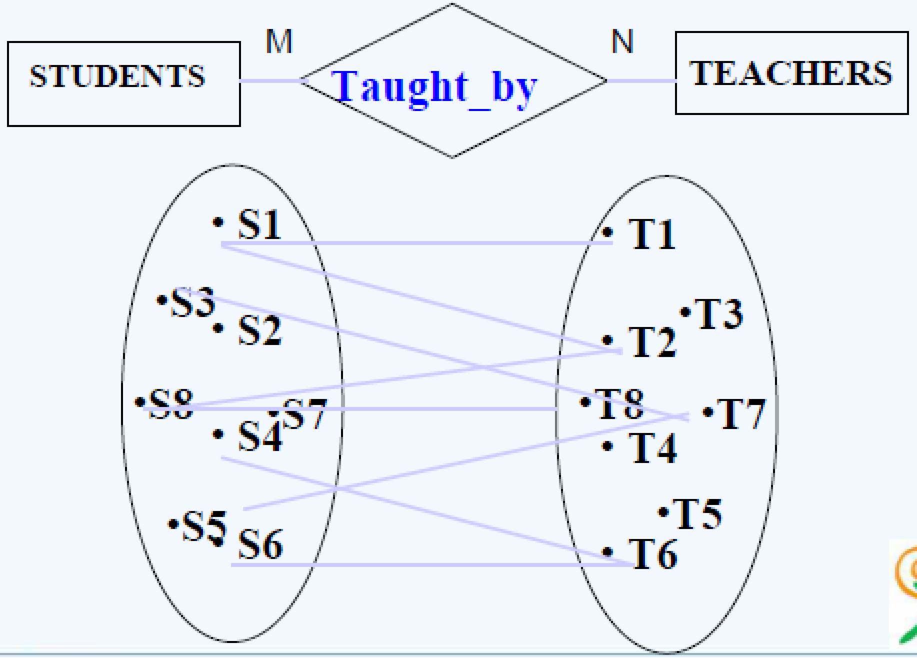
M:1 relationship





M:N relationship

M : N



STUDENTS

M

Taught_by

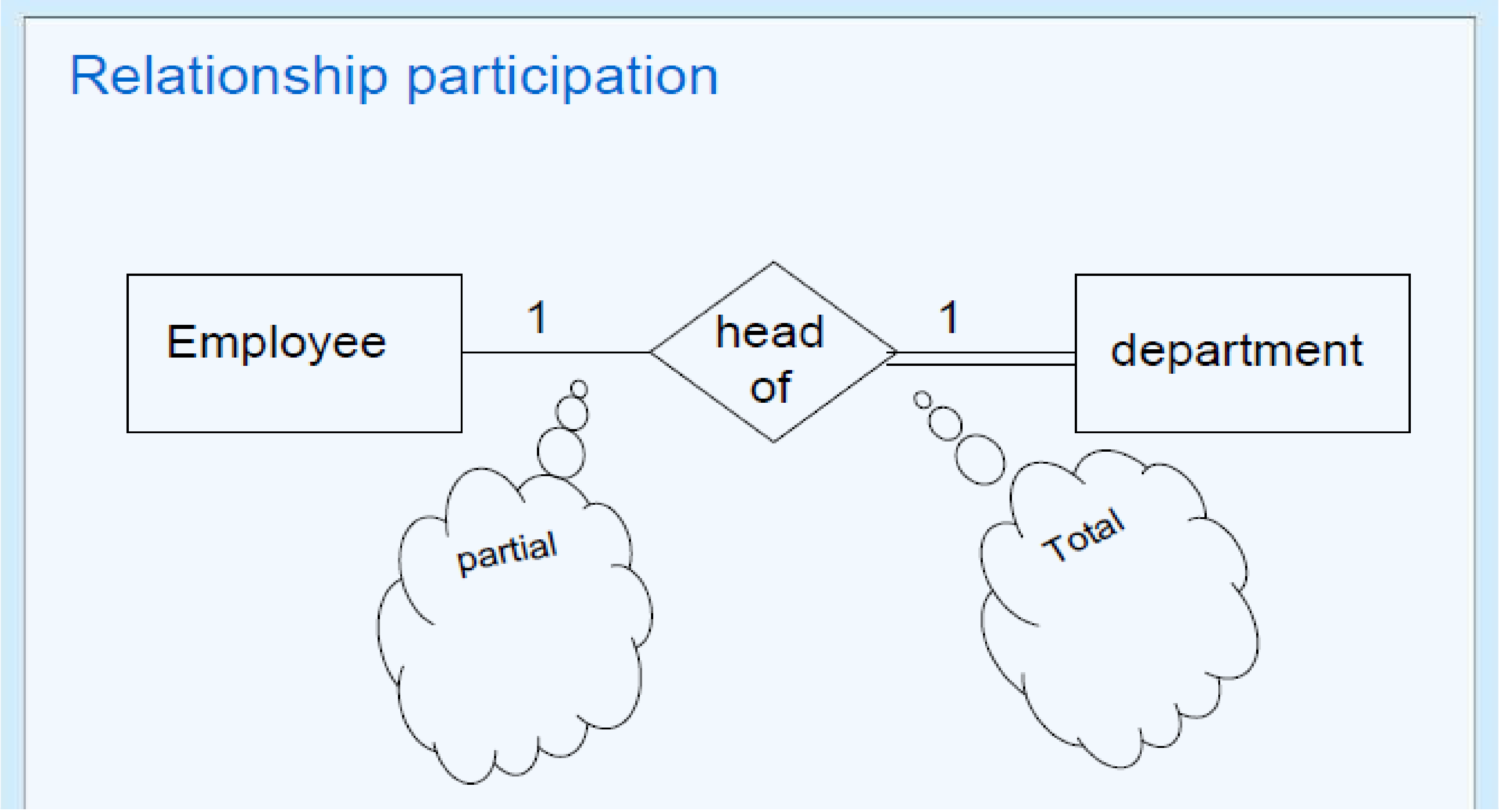
N

TEACHERS

• S1
•S3
• S2
•S8
•S7
• S4
•S5
• S6

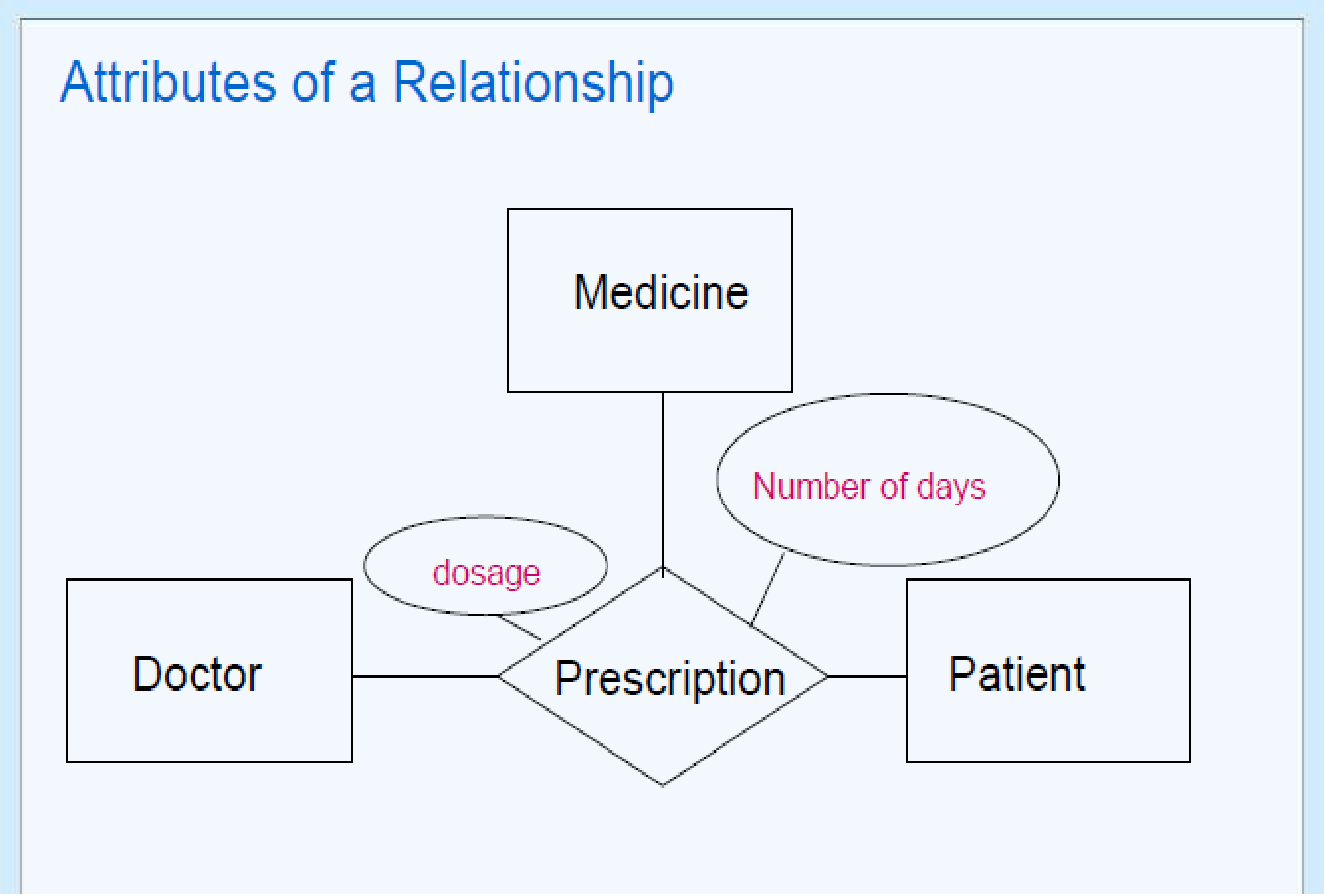
• T1
•T3
• T2
•T8
•T7
• T4
•T5
• T6

- All instances of the entity type Employee don’t participate in the relationship, Head-of.
- Every employee doesn’t head a department. So, employee entity type is said to partially participate in the relationship.
- But, every department would be headed by some employee.
- So, all instances of the entity type Department participate in this relationship. So, we say that it is total participation from the department side.



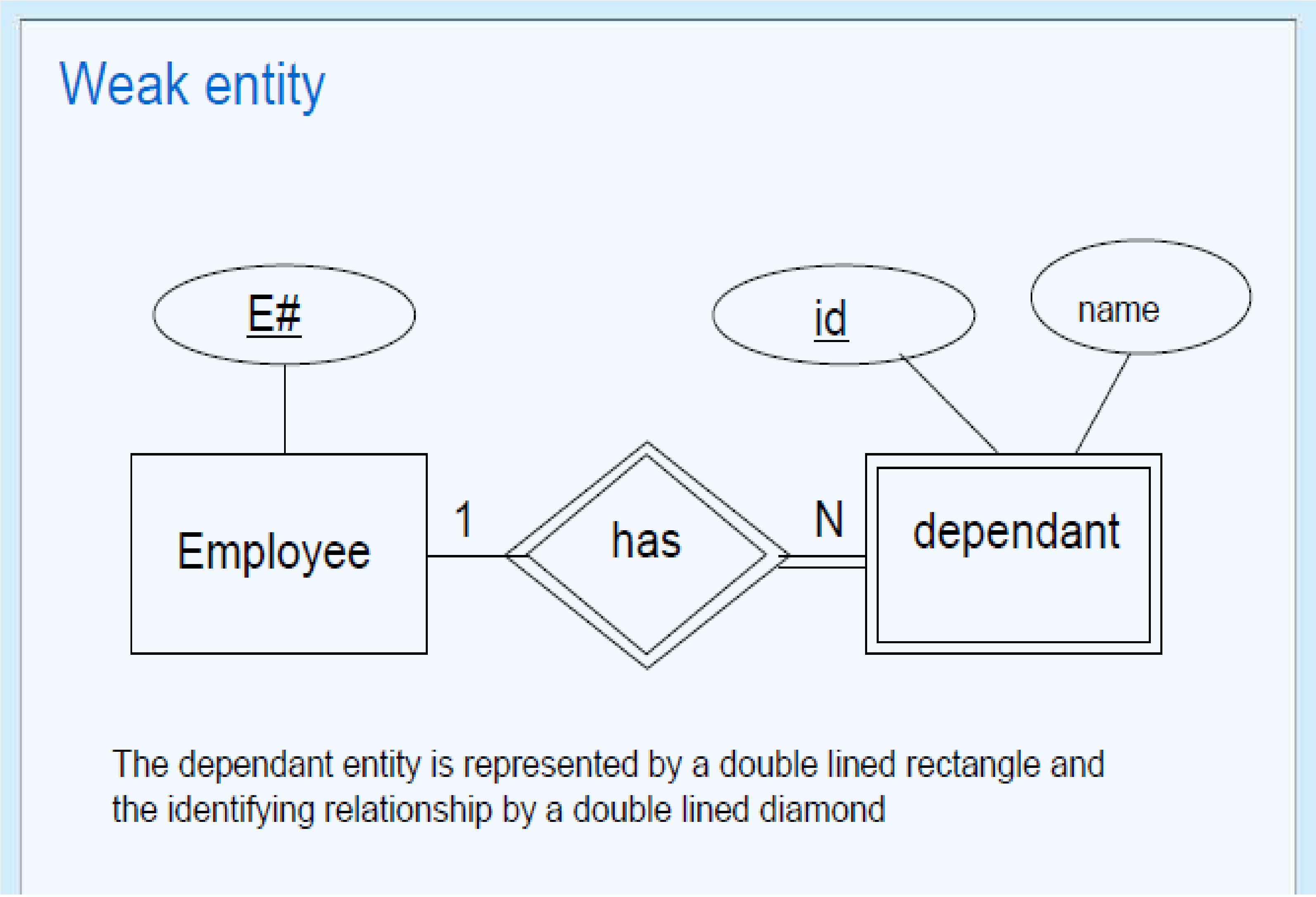
Relationship participation

Descriptive attributes: These attributes best describe the relationship rather than any individual entity



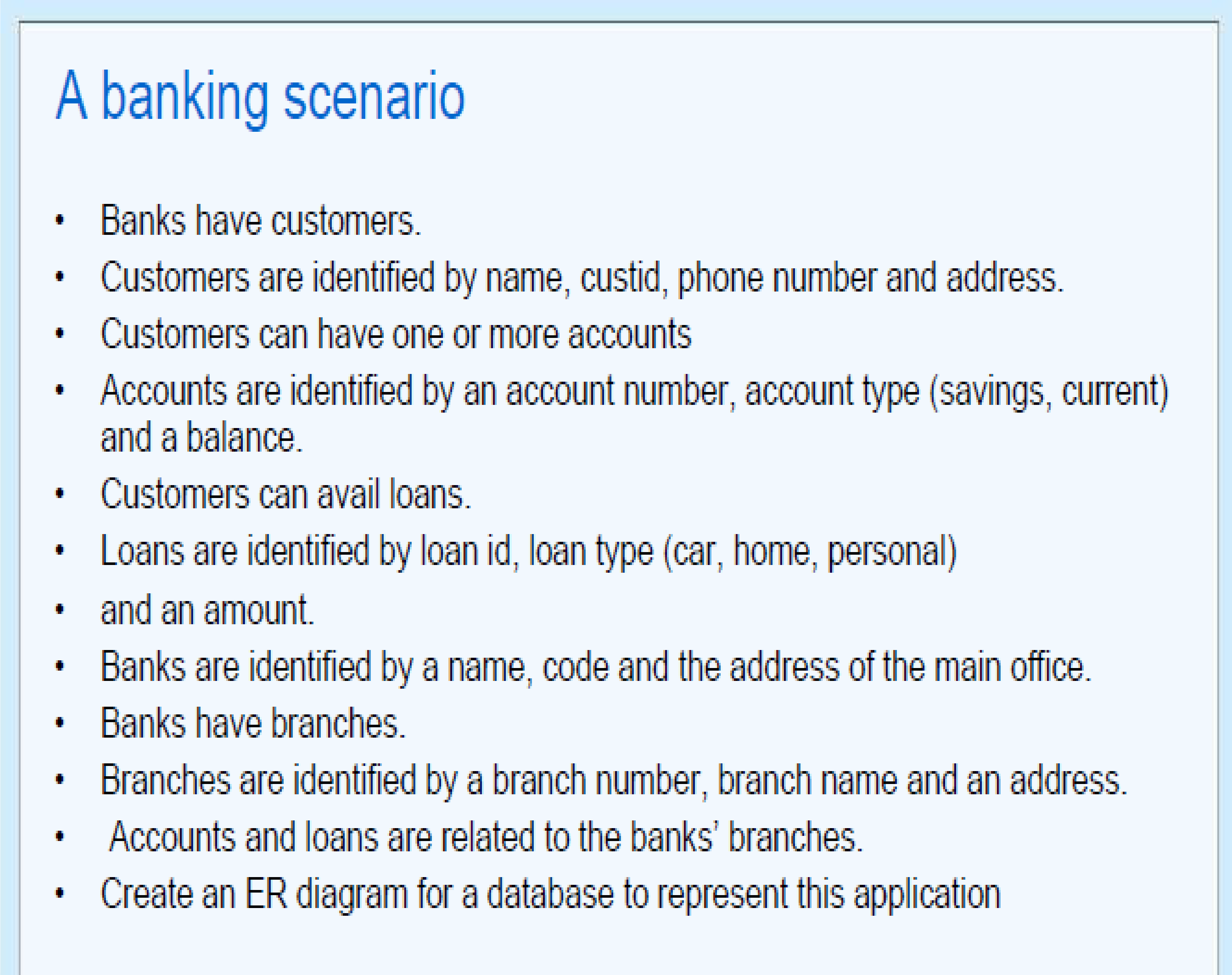
Attributes of a Relationship

Identifying relationship is the one which relates the weak entity with the strong entity.



Weak entity

The dependant entity is represented by a double lined rectangle and the identifying relationship by a double lined diamond



A banking scenario

- Banks have customers.
- Customers are identified by name, custid, phone number and address.
- Customers can have one or more accounts
- Accounts are identified by an account number, account type (savings, current) and a balance.
- Customers can avail loans.
- Loans are identified by loan id, loan type (car, home, personal)
- and an amount.
- Banks are identified by a name, code and the address of the main office.
- Banks have branches.
- Branches are identified by a branch number, branch name and an address.
- Accounts and loans are related to the banks' branches.
- Create an ER diagram for a database to represent this application

Solution Step 1: Identify the entities

- Bank
- Branch
- Customer
- Account
- Loan

Solution Step 2: Identify attributes of entities

- Bank

  Name
  Code
  Address

Branch

Branch#
Branch name
Address

Customer

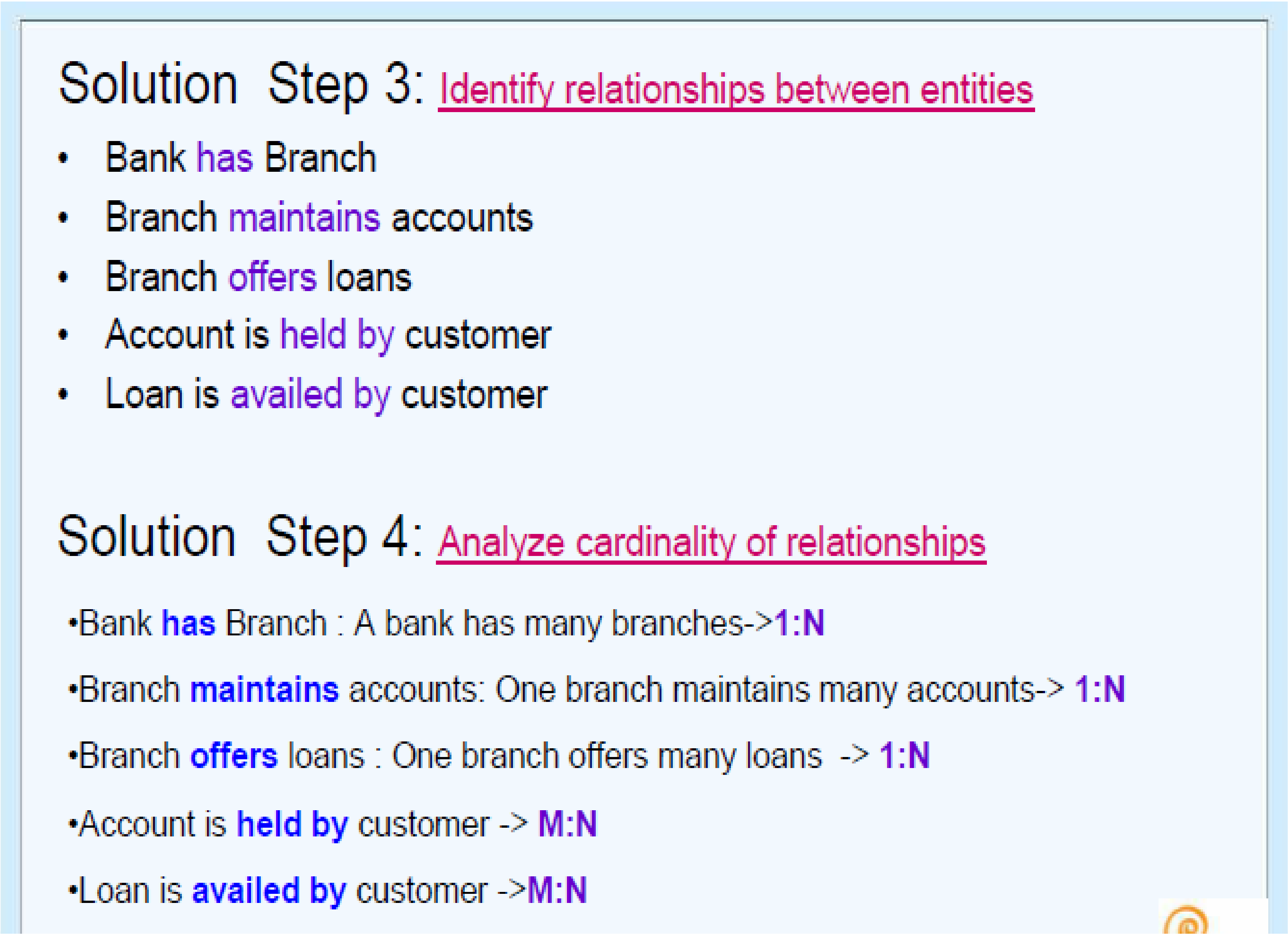
Name
Custid
Phone
Address

Account

Account number
Account type
Balance

Loan

Loan id
Loan type
Amount

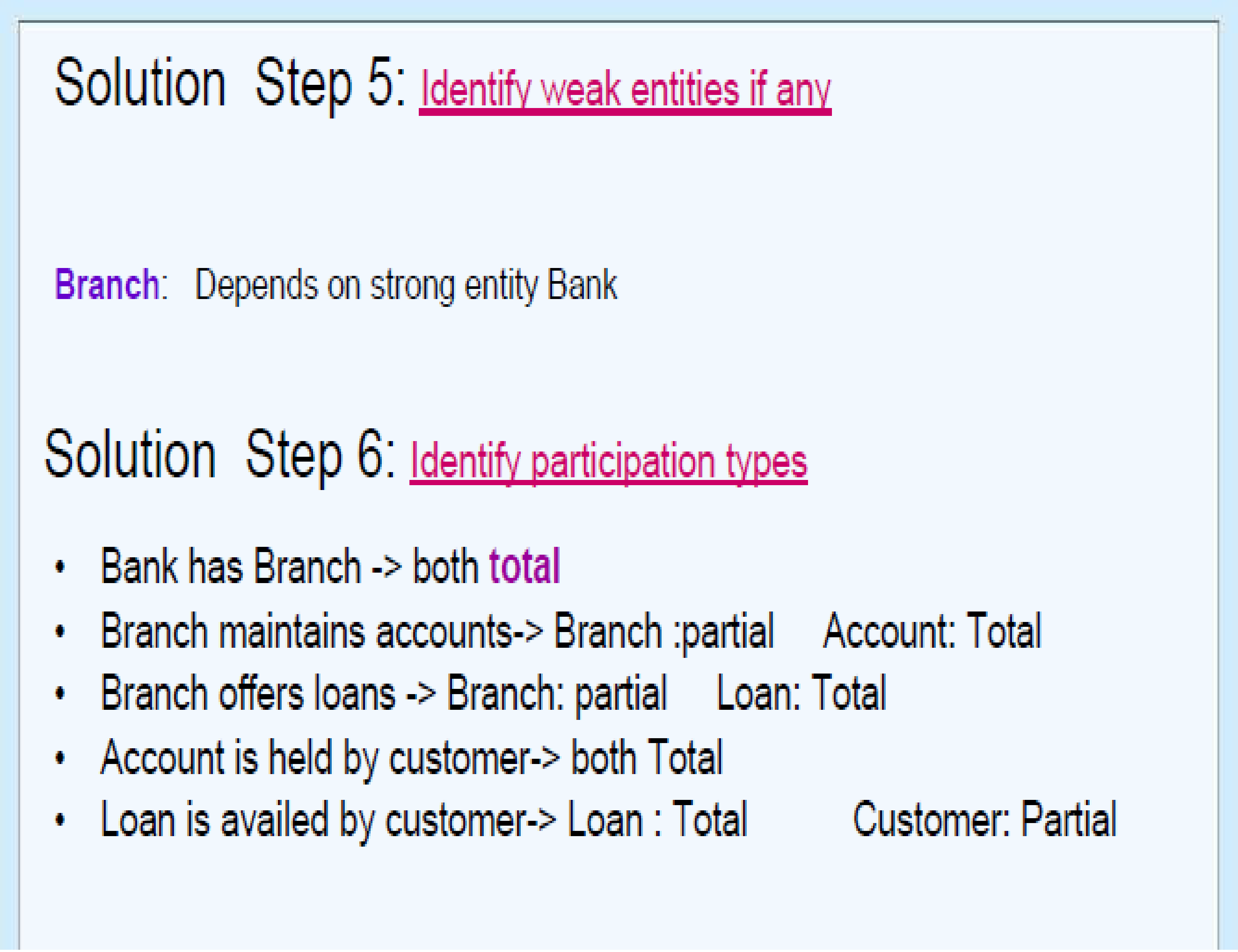


Solution Step 3: Identify relationships between entities

- Bank has Branch
- Branch maintains accounts
- Branch offers loans
- Account is held by customer
- Loan is availed by customer

Solution Step 4: Analyze cardinality of relationships

- Bank **has** Branch : A bank has many branches->**1:N**
- Branch **maintains** accounts: One branch maintains many accounts-> **1:N**
- Branch **offers** loans : One branch offers many loans -> **1:N**
- Account is **held by** customer -> **M:N**
- Loan is **availed by** customer ->**M:N**



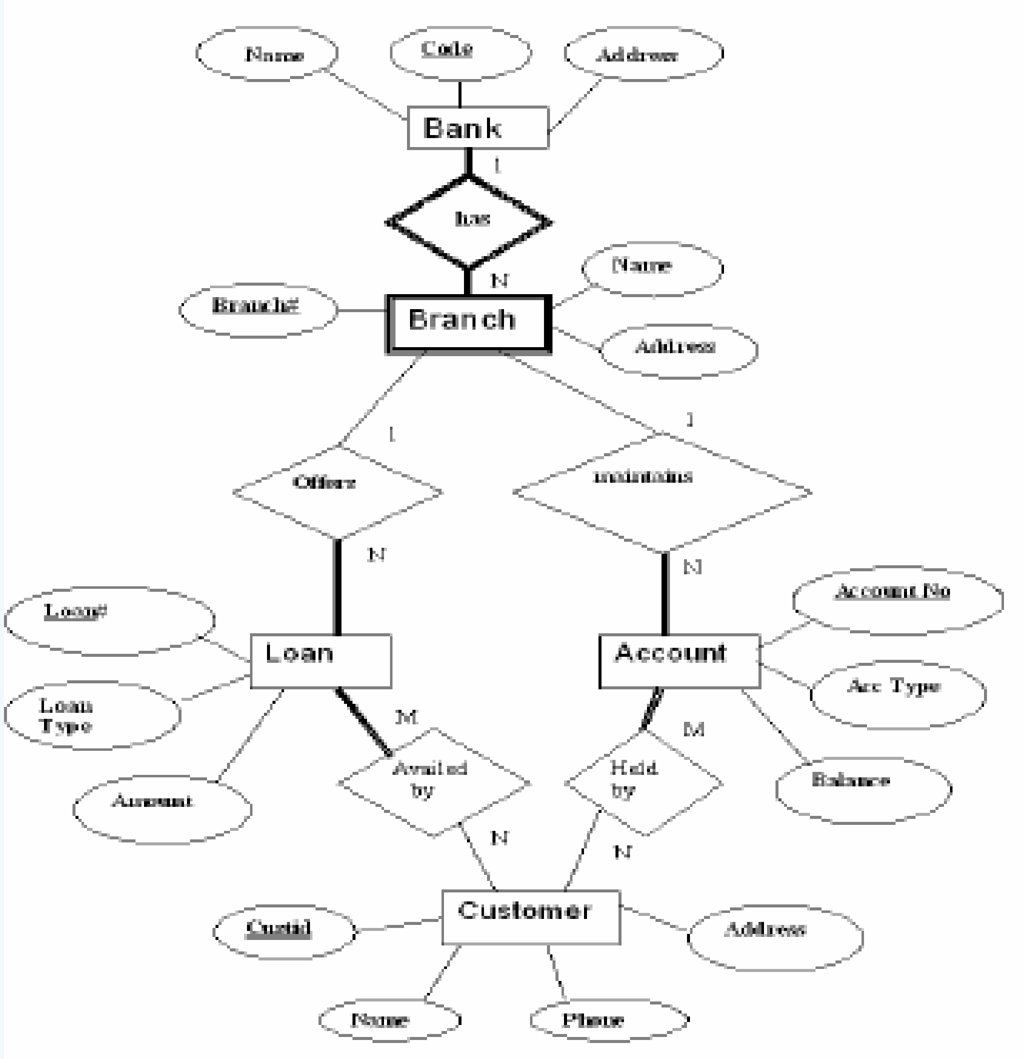
Solution Step 5: Identify weak entities if any

Branch: Depends on strong entity Bank

Solution Step 6: Identify participation types

- Bank has Branch -> both **total**
- Branch maintains accounts-> Branch :partial Account: Total
- Branch offers loans -> Branch: partial Loan: Total
- Account is held by customer-> both Total
- Loan is availed by customer-> Loan : Total Customer: Partial

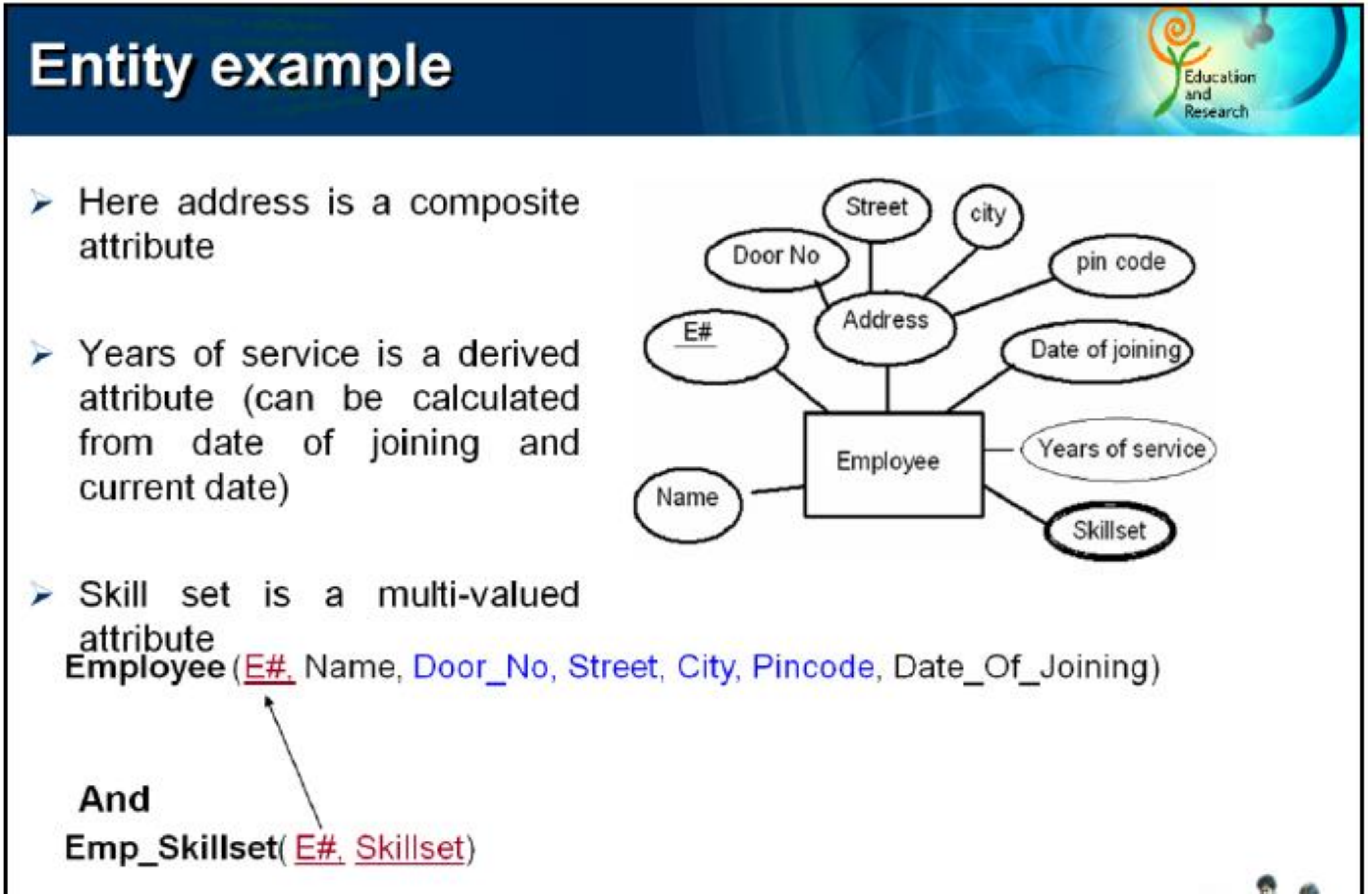
Represented diagrammatically



Converting ER diagram into Relational Schema

Converting Strong entity types

- Each **entity type** becomes a **table**
- Each **single-valued attribute** becomes a **column**
- **Derived attributes** are **ignored**
- **Composite attributes** are represented by its **equivalent parts**
- **Multi-valued attributes** are represented by a **separate table**
- The **key attribute** of the entity type becomes the **primary key** of the table



Entity example

- Here address is a composite attribute
- Years of service is a derived attribute (can be calculated from date of joining and current date)
- Skill set is a multi-valued attribute

Employee (E#, Name, Door_No, Street, City, Pincode, Date_Of_Joining)

And

Emp_Skillset(E#, Skillset)

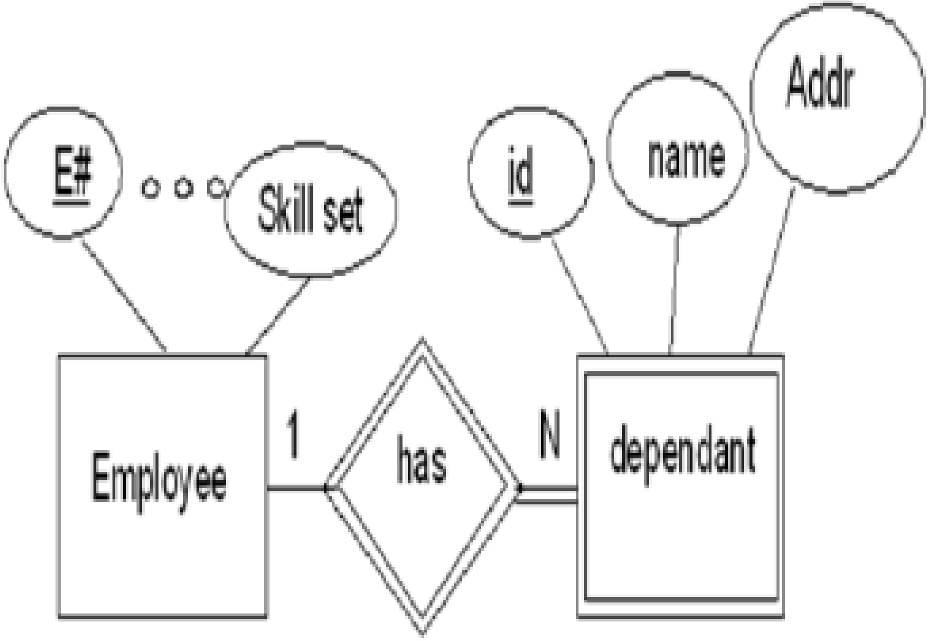
As per the rules: 1. Derived attributesare **ignored.**

2. Composite attributesare represented by **components.**

3. Multi-valued attributesare represented by a **separate table.**

Converting weak entity types

- Weak entity types are converted into a table of their own, with the primary key of the strong entity acting as a foreign key in the table.

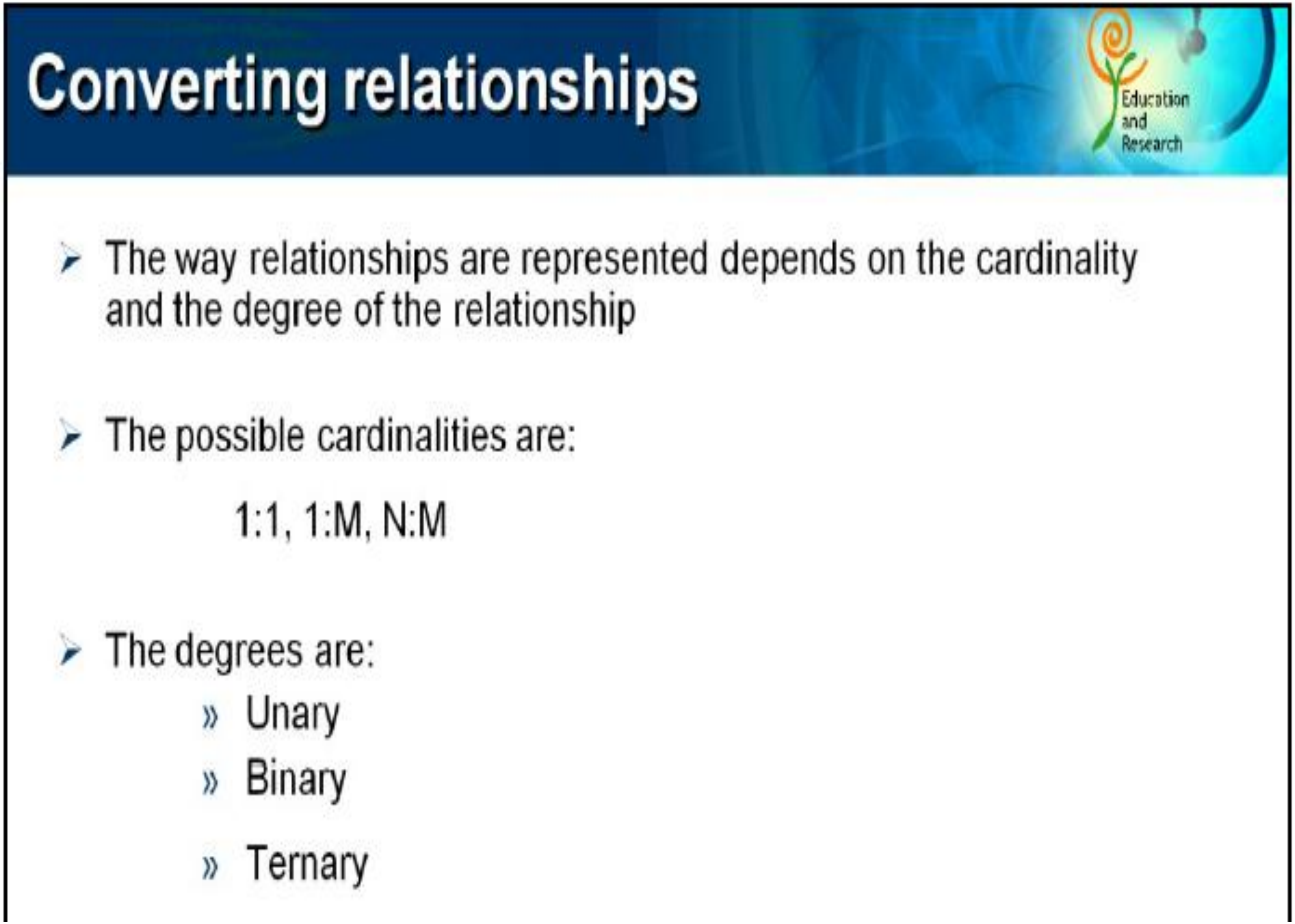


- This foreign key along with the key of the weak entity forms the composite primary key of this weak table

- **The Relational Schema**

Employee (E# ,EmpName,DateOfJoining,SkillSet)

Dependant (Employee, Dependant_ID, Name, Address)



Converting relationships

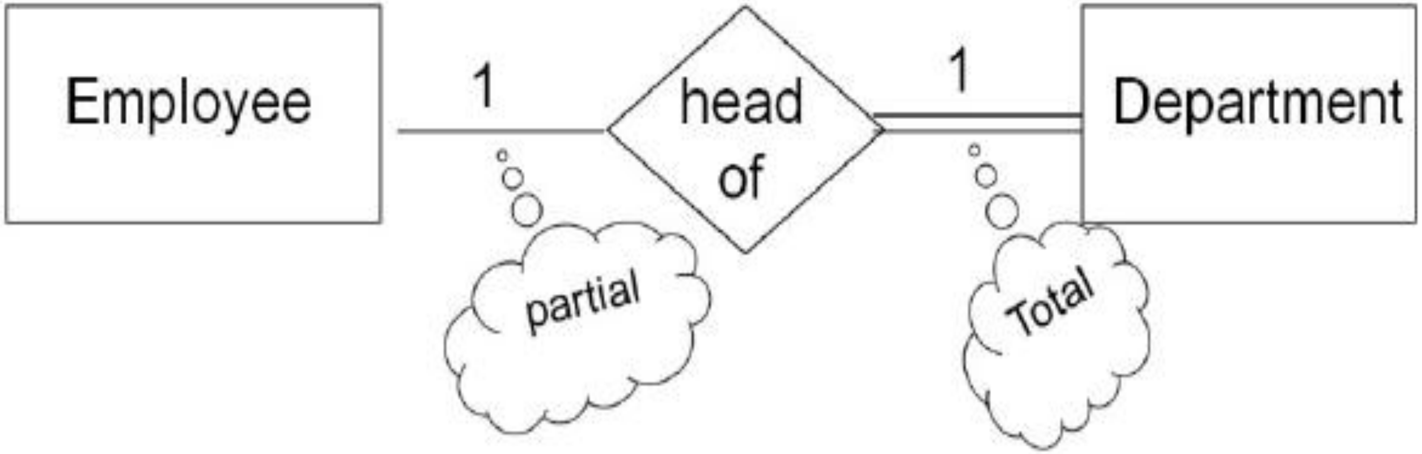
- The way relationships are represented depends on the cardinality and the degree of the relationship

- The possible cardinalities are:

  1:1, 1:M, N:M

- The degrees are:
  - Unary
  - Binary
  - Ternary

Binary 1:1



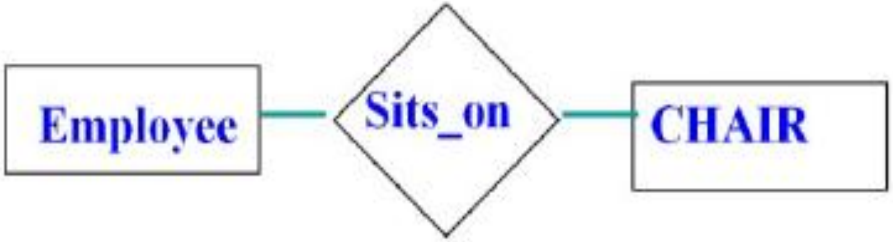
- Case 1: Combination of participation types

- The primary key of the partial participant will become the foreign key in the total participant.

Employee(E#, EName,DateOfJoining,SkillSet)

Department (Dept#, DName,Location,**Head**)

Binary 1:1


Employee

Sits_on

CHAIR

- Case 2: Uniform participation types
- The primary key of either of the participants can become a foreign key in the other

Employee (EmpCode, EmpName, DateOfJoining)

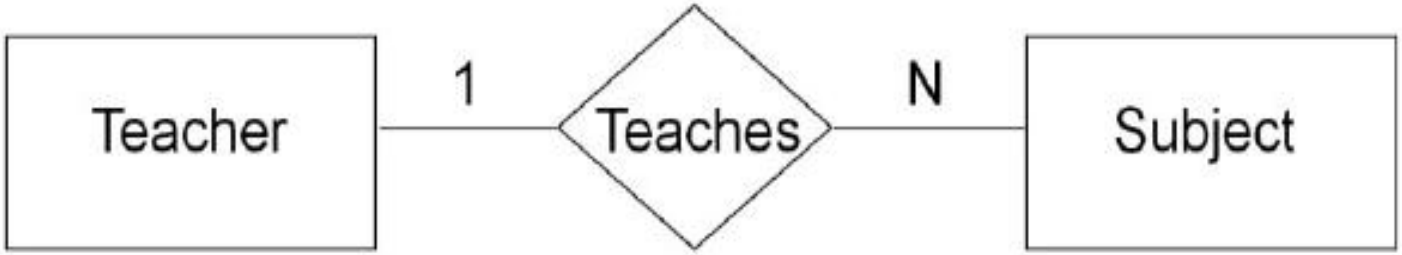
Chair(Item#, Model, Location, Used_by)

(OR)

Employee (EmpCode, EmpName, DateOfJoining, Sits_on)

Chair (Item#, Model, Location)

Binary 1:N



Teacher

1

Teaches

N

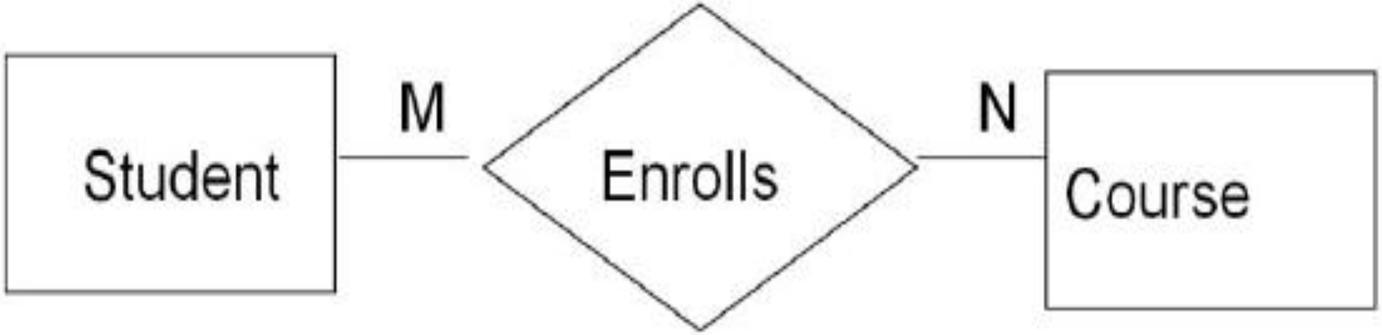
Subject

The primary key of the relation on the "1" side of the relationship becomes a foreign key in the relation on the "N" side.

Teacher (TeacherID, Name, Telephone, Cabin)

Subject (SubCode, SubName, Duration, TeacherID)

Binary M:N



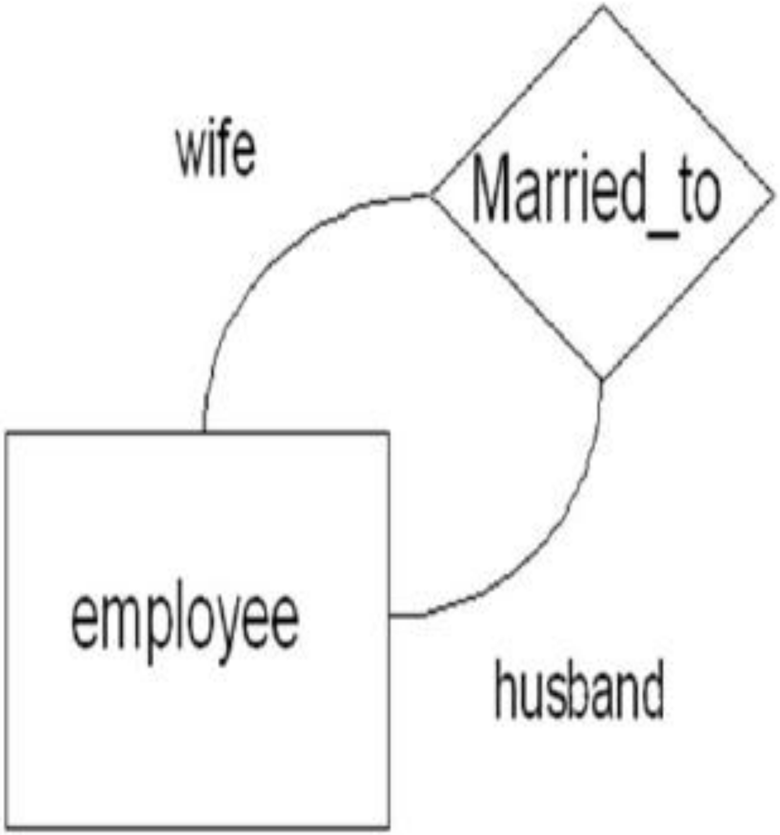
- A new table is created to represent the relationship which contains two foreign keys - one from each of the participants in the relationship.
- The primary key of the new table is the combination of the two foreign keys.
- Student (StudentID,SName,DOB,Address) Course(CourseID,CName)

Enrolls (SID, CID)

Unary 1:1

- Consider employees who are also a couple
- The primary key field itself will become foreign key in the same table

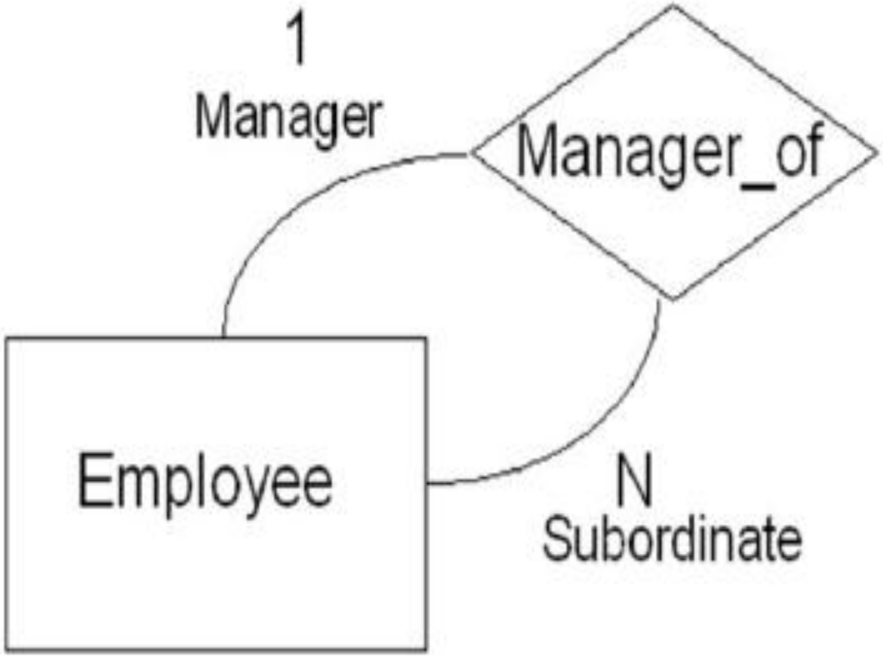
Employee(E#, EName,…**Spouse**)



Unary 1:N

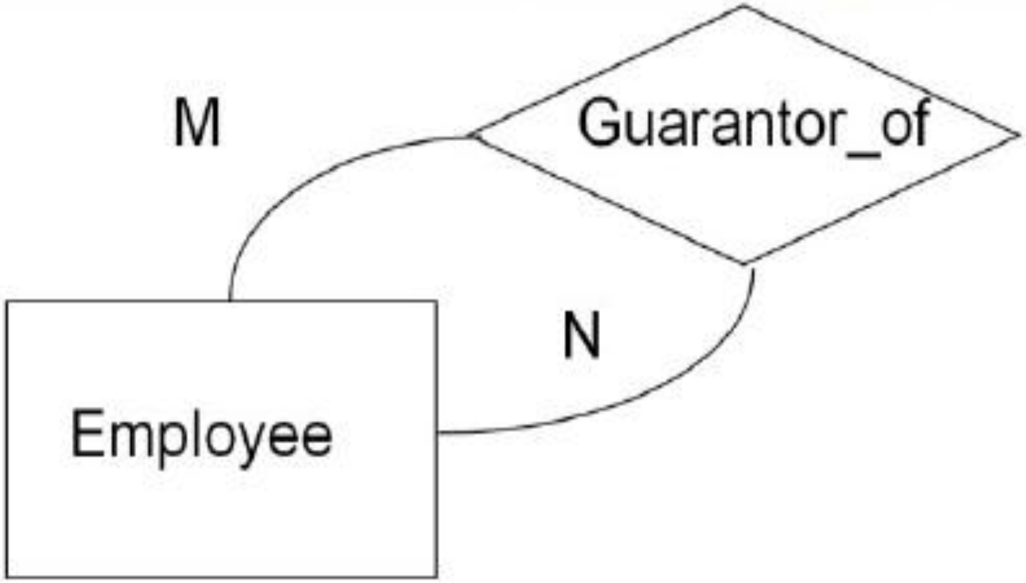
- The primary key field itself will become foreign key in the same table.

- Same as unary 1:1



Employee(E#, EName, DateOfJoining, SkillSet, **Manager**)

Unary M:N



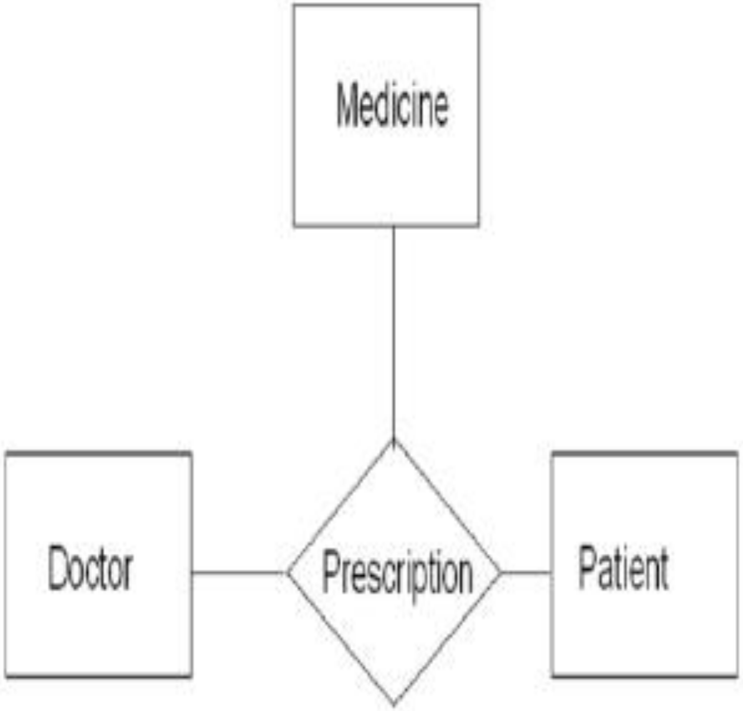
- There will be two resulting tables. One to represent the entity and another to represent the M:N relationship as follows

Employee(E#, EName, DateOfJoining, SkillSet)

Guaranty(Guarantor, Beneficiary)

Ternary relationship

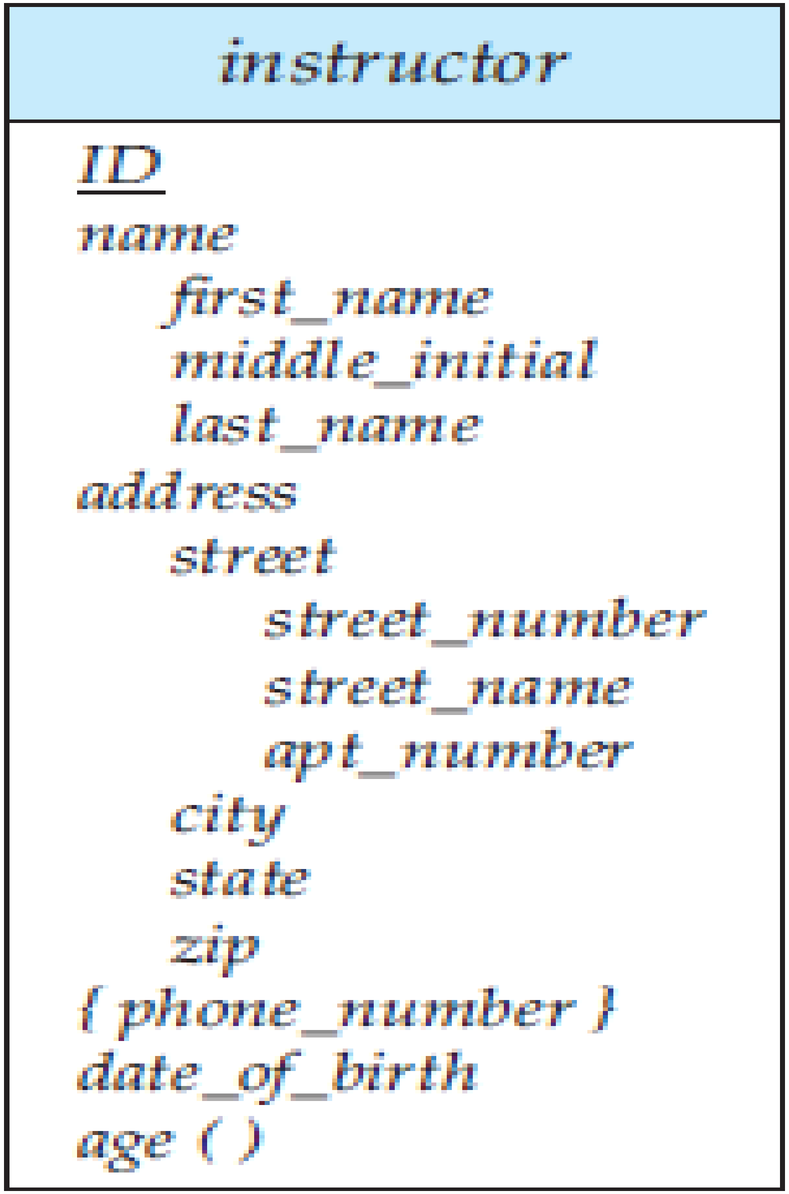
- Represented by a new table.
- The new table contains **three** foreign keys - one from each of the participating Entities.
- The primary key of the new table is the combination of all three foreign keys.
- Prescription (DocID, PatCode, MedName)



UML Notations for ER diagram

- **Rectangles divided into two parts represent entity sets. The first part,** contains the name of the entity set. **The second** part contains the names of all the attributes of the entity set.
- **Diamonds represent relationship sets.**
- **Undivided rectangles represent the attributes of a relationship set.**
- **Attributes** that are part of the primary key are underlined.
- **Lines link entity sets to relationship sets.**
- **Dashed lines link attributes of a relationship set to the relationship set.**
- **Double lines indicate total participation of an entity in a relationship set.**
- **Double diamonds represent identifying relationship sets linked to weak** entity sets

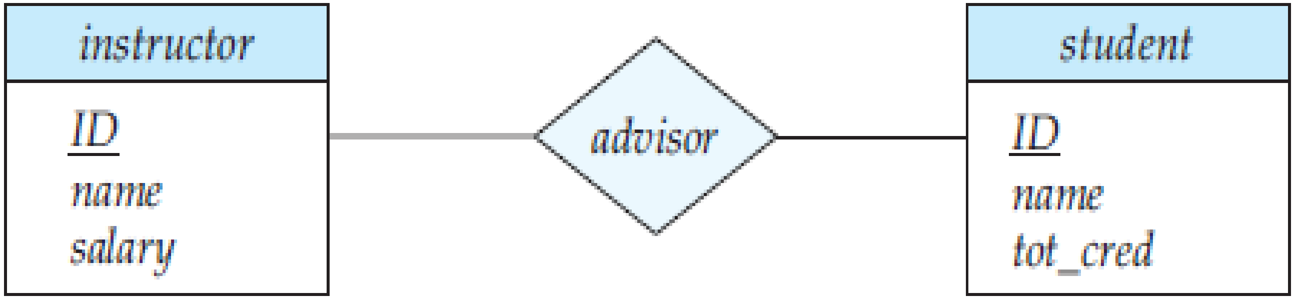
E-R diagram with composite, multivalued, and derived attributes



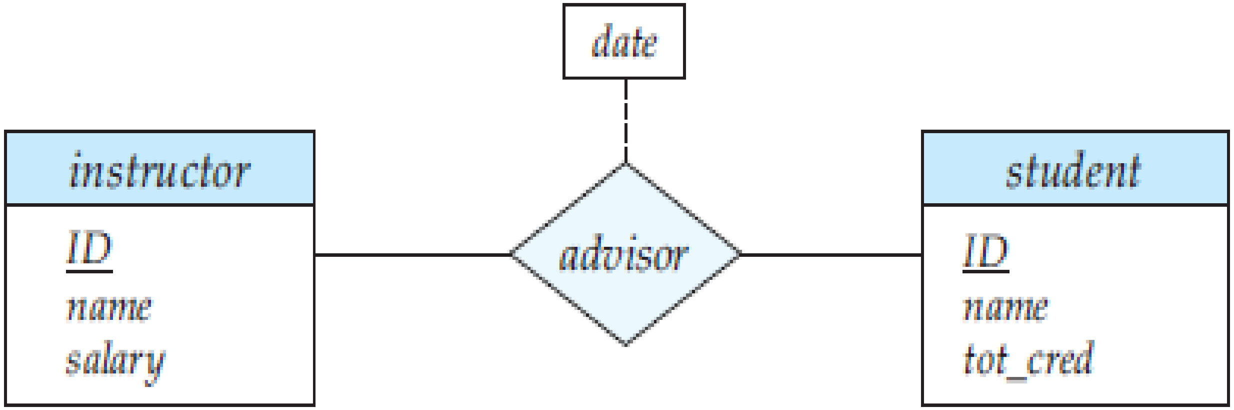
instructor

- *ID*
- *name*
    - *first_name*
    - *middle_initial*
    - *last_name*
- *address*
    - *street*
        - *street_number*
        - *street_name*
        - *apt_number*
    - *city*
    - *state*
    - *zip*
- *{ phone_number }*
- *date_of_birth*
- *age ( )*

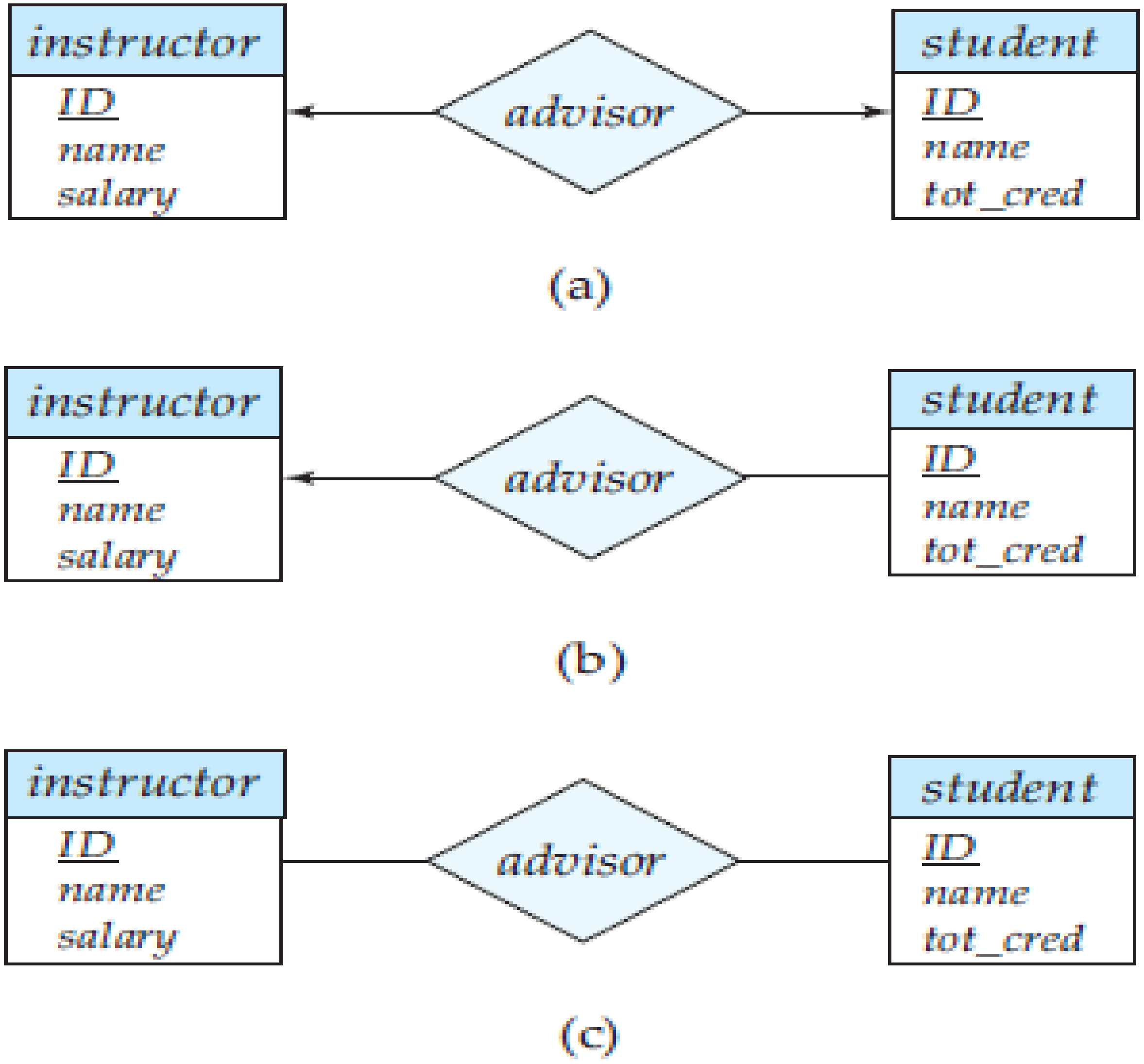
E-R diagram corresponding to instructors and students



E-R diagram with an attribute attached to a relationship set



Relationships. (a) One-to-one. (b) One-to-many. (c) Many-to-many

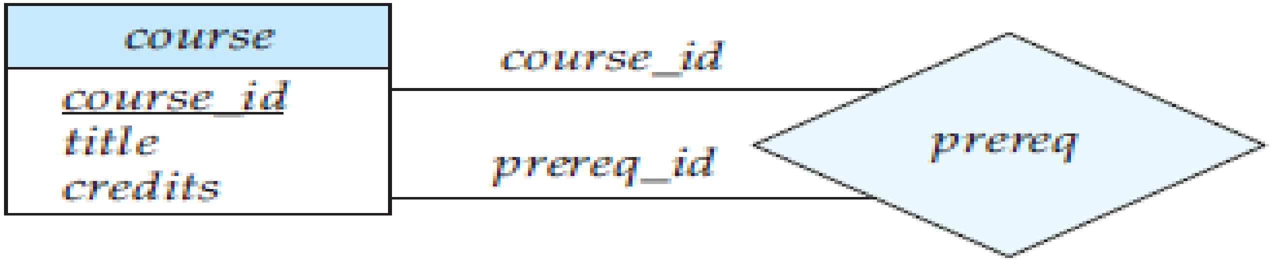


(a)

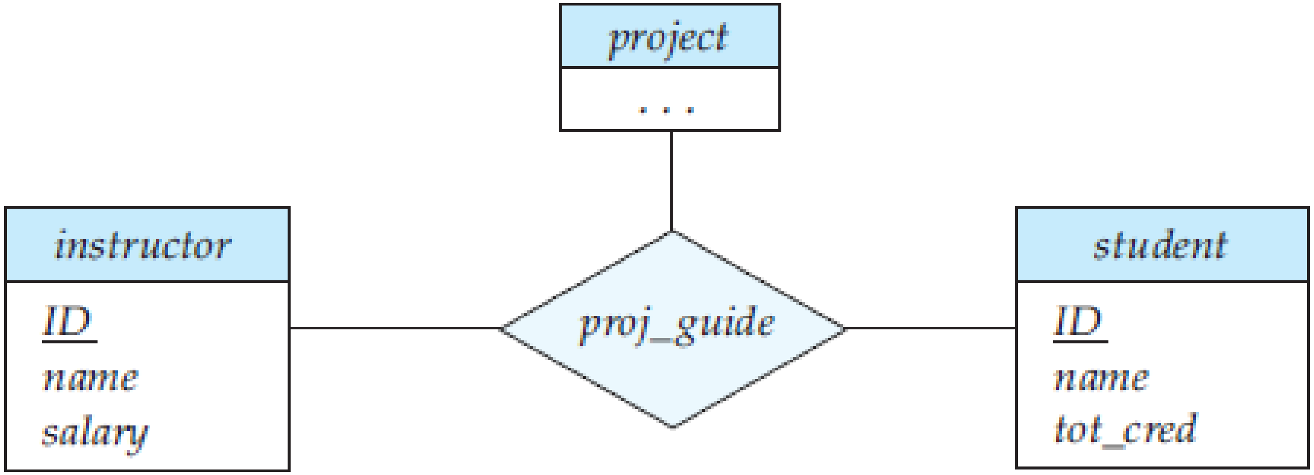
(b)

(c)

E-R diagram with role indicators & Unary relationship



E-R diagram with a ternary relationship



Strong and Weak entity set

➢An entity set that does not have sufficient attributes to form a primary key is termed a **weak entity set.**

➢ **An entity set that has a primary key is termed a strong entity set.**

➢A weak entity set must be associated with another entity set, called the **identifying or owner entity set.**

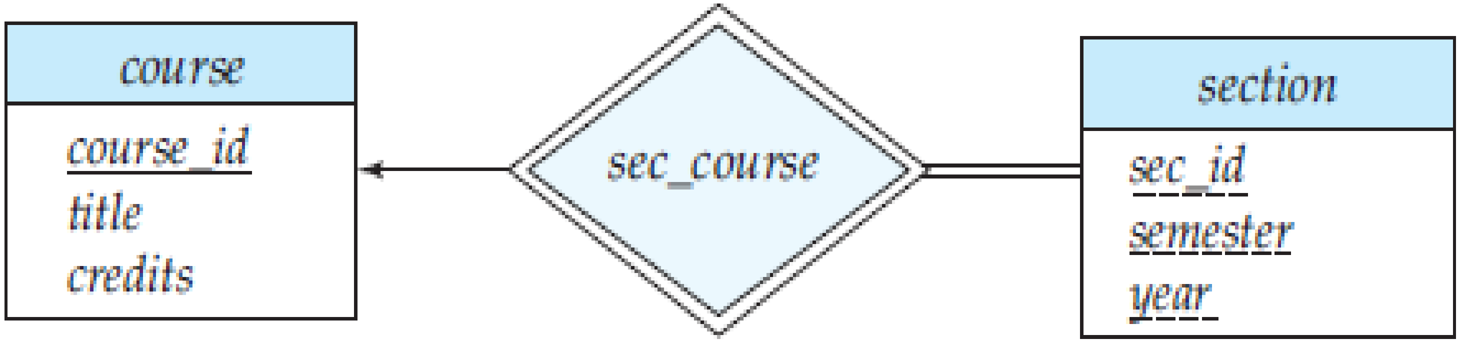
➢**Every weak entity must** be associated with an identifying entity; that is, the weak entity set is said to be **existence dependent on the identifying entity set.**

➢**The identifying entity set is said** to **own the weak entity set that it identifies.**

➢**The relationship associating the weak** entity set with the identifying entity set is called the **identifying relationship**.

- A weak entity set does not have a primary key, but distinguish all the entities in the weak entity set that depend on one particular strong entity. **The discriminator of a weak entity set is a set of attributes that allows this distinction to be made**. The discriminator of a weak entity set is also called the *partial key of the entity set.*

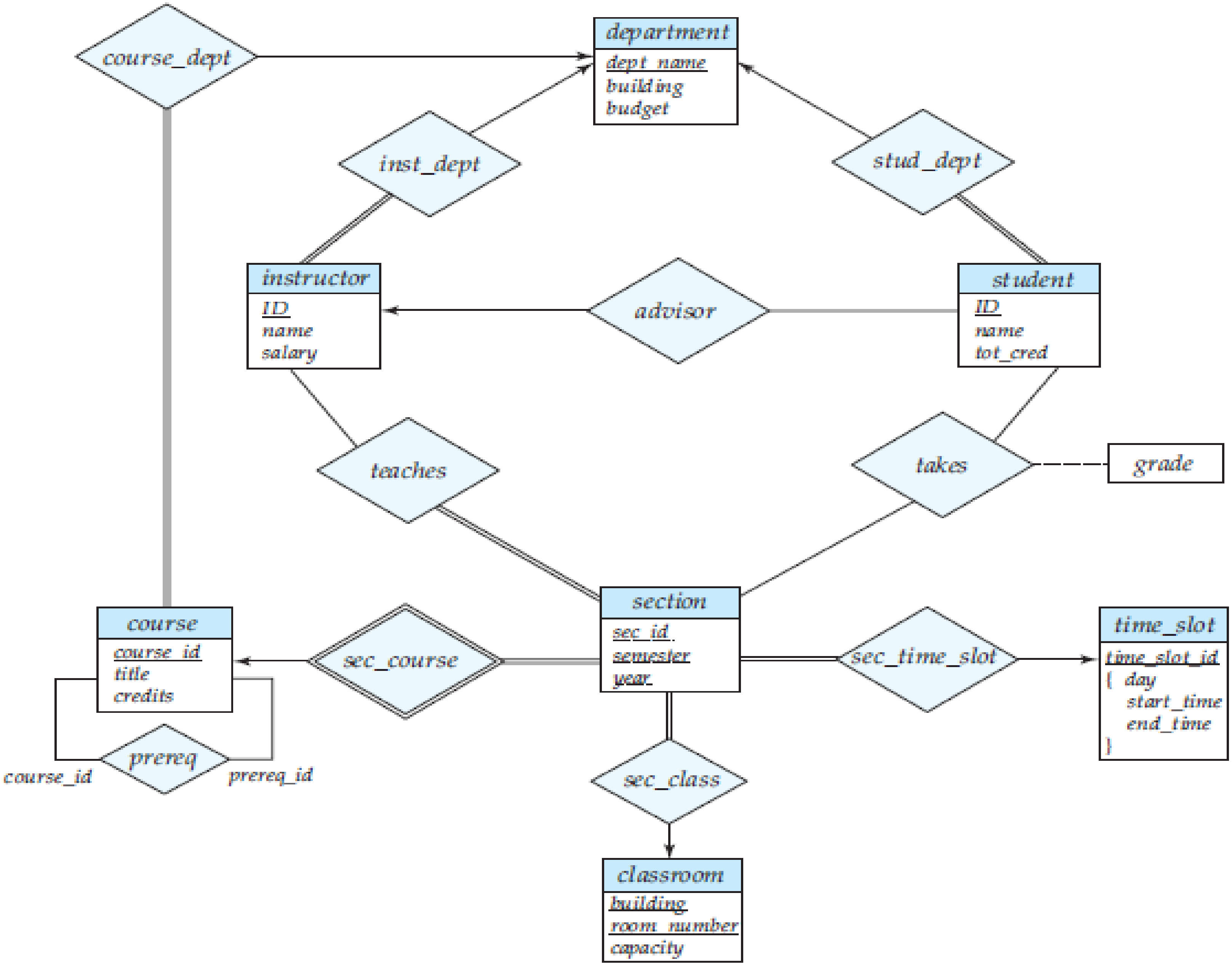
- **The primary key of a weak entity set is formed by the primary key of the identifying entity set, plus the weak entity set's discriminator**



In the case of the entity set *section, its primary key is {course_ id, sec_id, year, semester}, where course_id is the primary key of the identifying entity set, namely course, and {sec_id,year,semester} distinguishes section entities for the same course.*

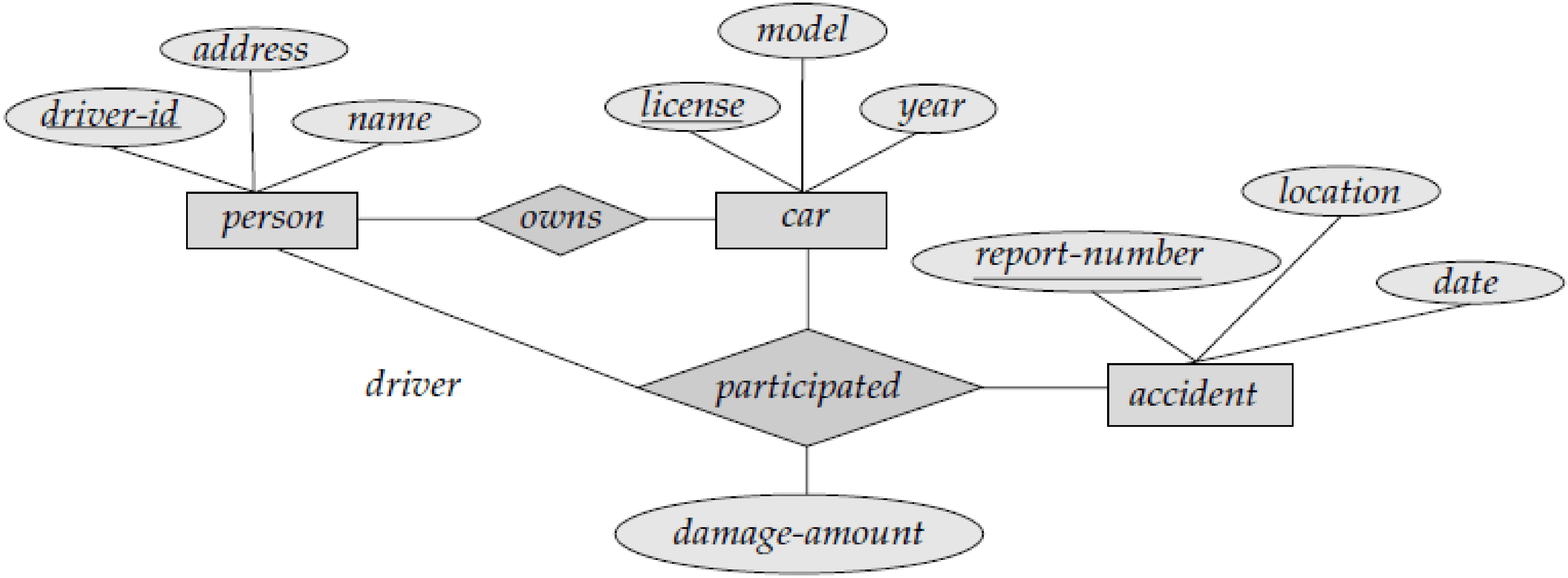
- The discriminator of a weak entity is underlined with a dashed, rather than a solid line.
- The relationship set connecting the weak entity set to the identifying strong entity set is depicted by a double diamond.

E-R diagram for a university enterprise



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 :

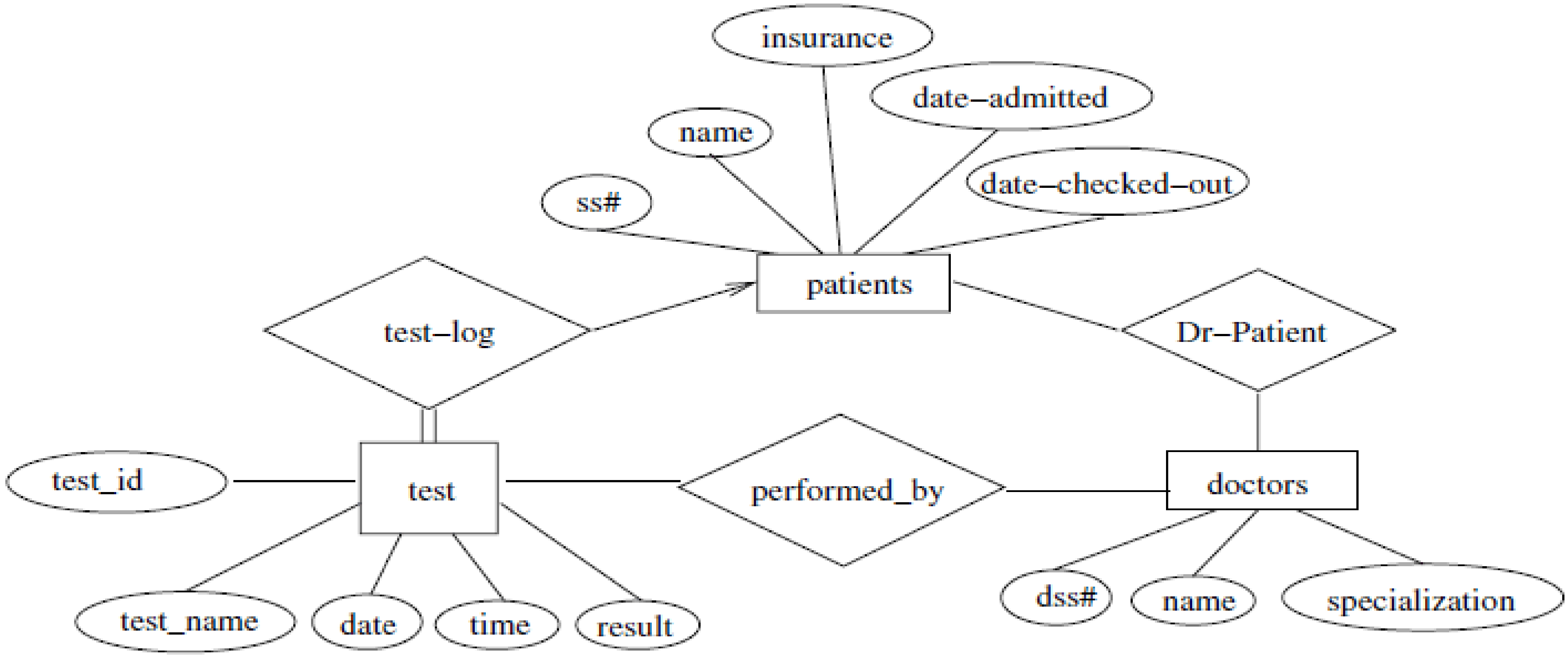


E-R diagram for a Car-insurance company.

Car insurance tables:
person (<u>driver-id</u>, name, address)
car (<u>license</u>, year,model, driver-id)
accident (<u>report-number</u>, date, location)
participated(<u>driver-id</u>, <u>license</u>, <u>report-number</u>, damage-amount)

(a)Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.

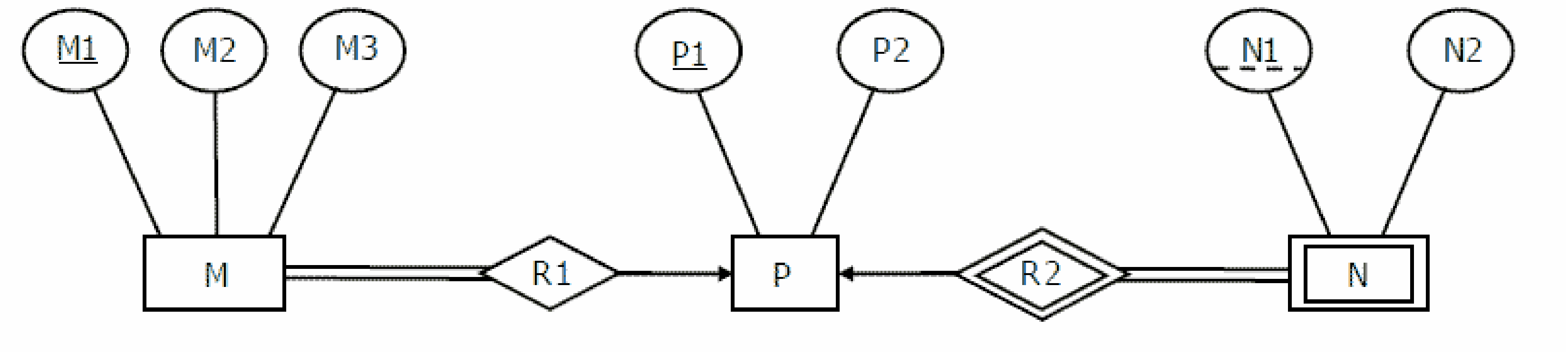
(b) Construct appropriate tables for the above ER Diagram :



E-R diagram for a hospital.

Patient(SS#, name, insurance,date-admitted,date-checked-out)
Doctor (dss#, name, specialization)
Test(test_id, test-name, date, time, result, SS#)
Doctor-patient (dss#, SS#)
Performed_by(dss#, test_id)

Consider the following ER diagram.



G1. The minimum number of tables needed to represent M, N, P, R1, R2 is

(A) 3 (B) 4 (C) 5 (D) 2

G2. Consider the data given in above question. Which of the following is a correct attribute set for one of the tables for the correct answer to the above question?

A.{M1, M2, M3, P1} B.{M1, P1, N1, N2}

C.{M1, P1, N1} D.{M1, P1}

G3. Mention the attributes of each table.

Suppose you are given the following requirements for a simple database for the National Hockey League (NHL):

➢the NHL has many teams,

➢each team has a name, a city, a coach, a captain, and a set of players,

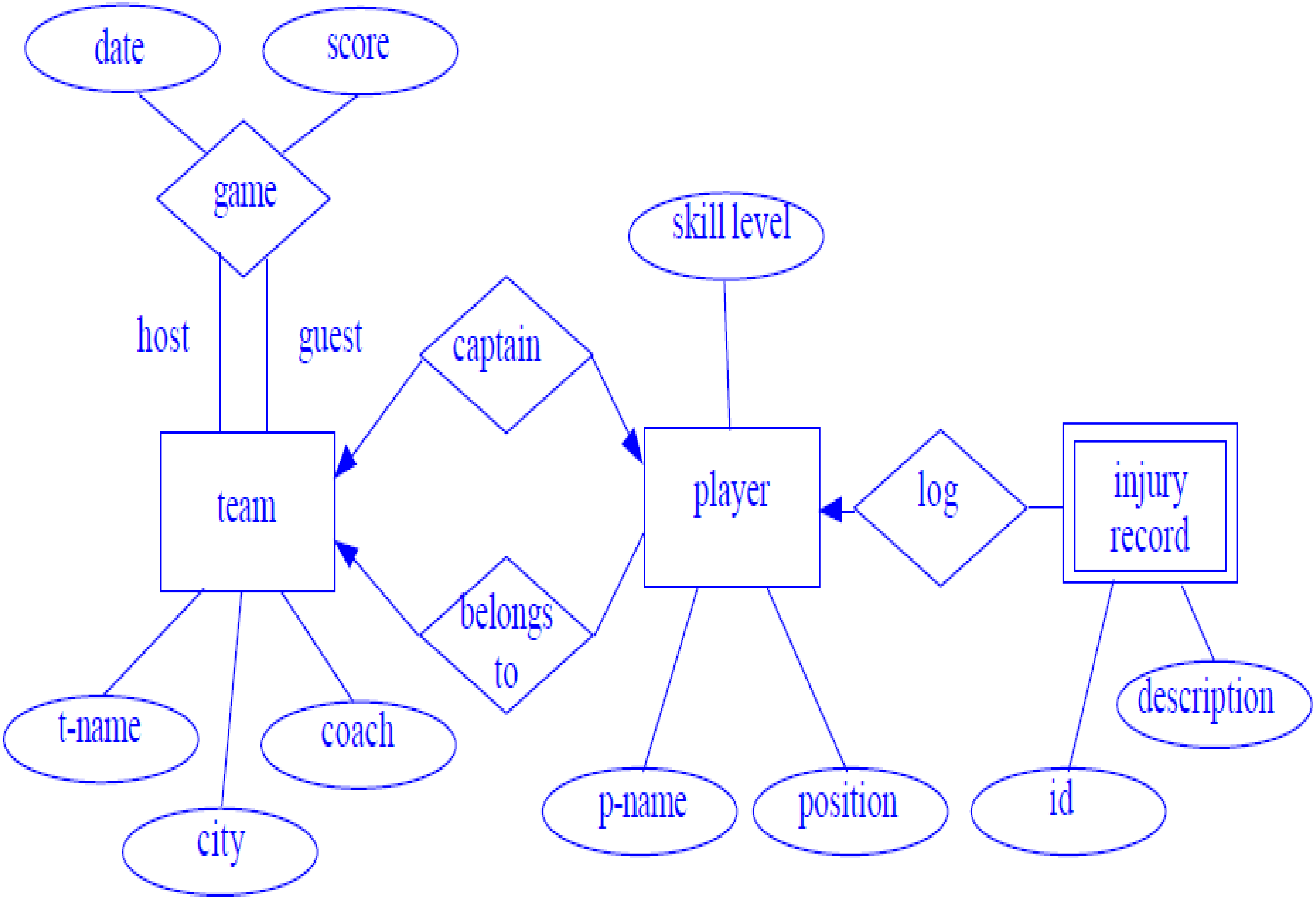
➢each player belongs to only one team,

➢each player has a name, a position (such as left wing or goalie), a skill level, and a set of injury records,

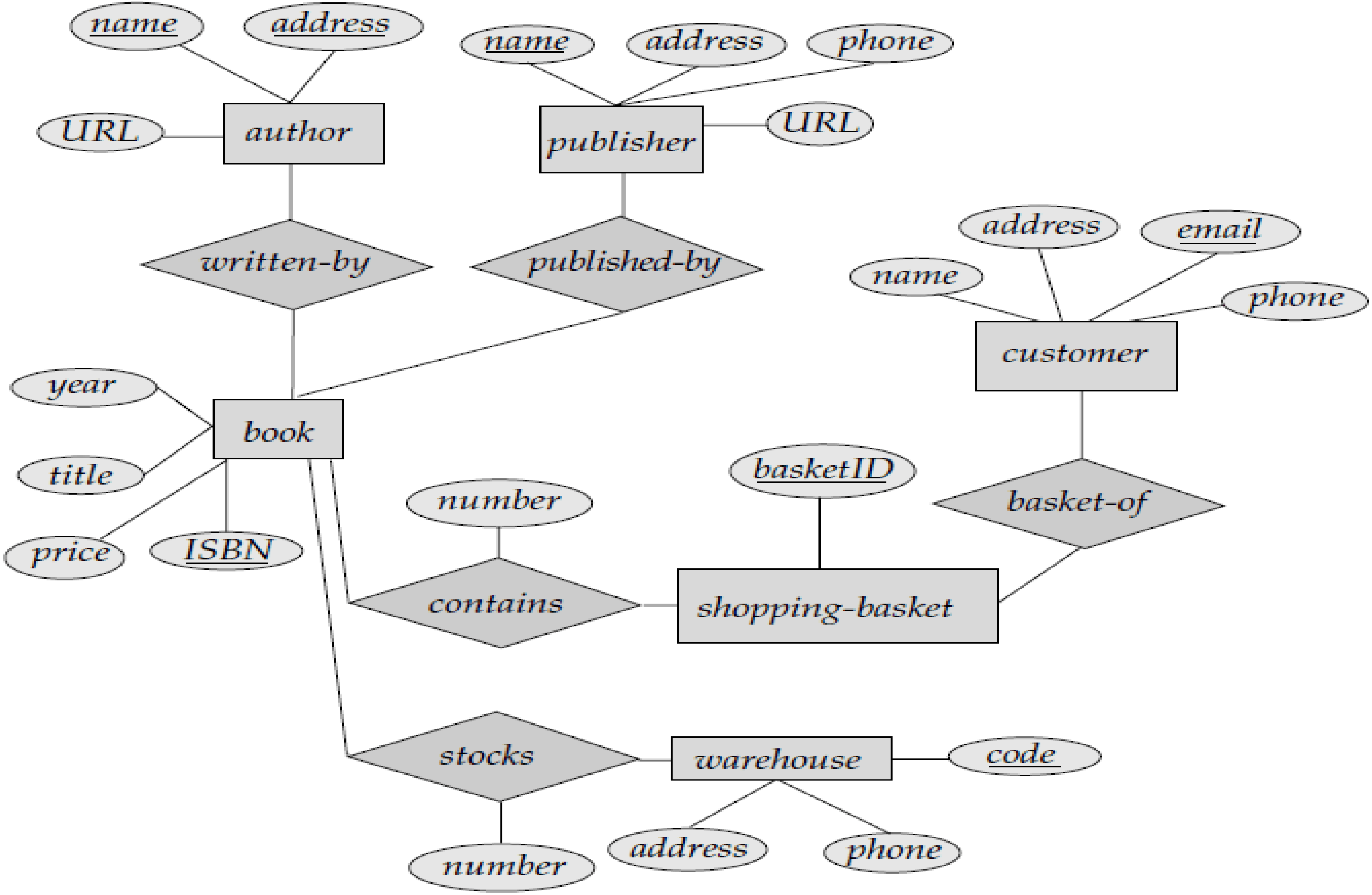
➢a team captain is also a player,

➢a game is played between two teams (referred to as host_team and guest_team) and has a date (such as May 11th, 1999) and a score (such as 4 to 2).

Construct a clean and concise ER diagram for the NHL database.

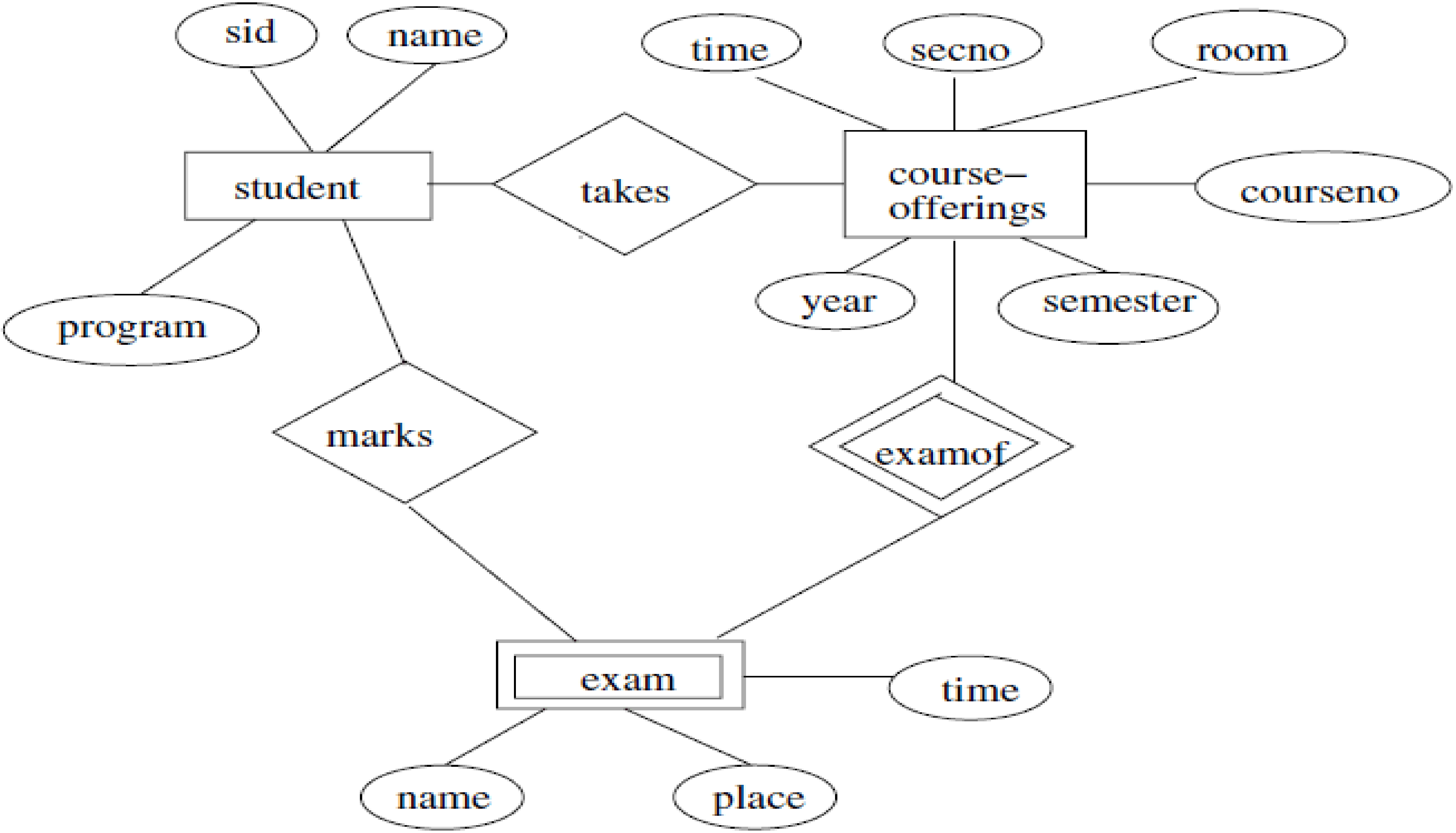

date
score
game
host
guest
skill level
captain
team
player
log
injury
record
belongs
to
t-name
coach
city
p-name
position
id
description

Draw the E-R diagram which models an online bookstore.



ER Diagram for Online BookStore

Construct an alternative E-R diagram that uses only a binary relationship between students and course-offerings. Make sure that only one relationship exists between a particular student and course-offering pair, yet you can represent the marks that a student gets in different exams of a course offering.



Another E-R diagram for marks database.

G4. Given the basic ER and relational models, which of the following is INCORRECT?

A.An attribute of an entity can have more than one value

B.An attribute of an entity can be composite

C.In a row of a relational table, an attribute can have more than one value

D.In a row of a relational table, an attribute can have exactly one value or a NULL value

G5. Let E1 and E2 be two entities in an E/R diagram with simple single-valued attributes. R1 and R2 are two relationships between E1 and E2, where R1 is one-to-many and R2 is many-to-many. R1 and R2 do not have any attributes of their own. What is the minimum number of tables required to represent this situation in the relational model?

(A) 3 (B) 4 (C) 5 (D) 2

G6. What is the min and max number of tables required to convert an ER diagram with 2 entities and 1 relationship between them with partial participation constraints of both entities?

A. Min 1 and max 2

B. Min 1 and max 3

C. Min 2 and max 3

D. Min 2 and max 2

G7. In an Entity-Relationship (ER) model, suppose R is a many-to-one relationship from entity set E1 to entity set E2. Assume that E1 and E2 participate totally in R and that the cardinality of E1 is greater that the cardinality of E2. Which one of the following is true about R?

A.Every entity in E1 is associated with exactly one entity in E2.

B.Some entity in E1 is associated with more than one entity in E2.

C.Every entity in E2 is associated with exactly one entity in E1.

D.Every entity in E2 is associated with at most one entity in E1

G8. Which symbol denote derived attributes in ER Model?

A.Double ellipse

B.Dashed ellipse

C.Squared ellipse

D.Ellipse with attribute name underlined

G9. Let M and N be two entities in an E-R diagram with simple single value attributes. R1 and R2 are two relationship between M and N, where as R1 is one-to-many and R2 is many-to-many. The minimum number of tables required to represent M, N, R1 and R2 in the relational model are

A.4 B.6 C.7 **D.3**

G10. Which one is correct w.r.t. RDBMS ?

A.primary key ⊆ super key ⊆ candidate key

B.primary key ⊆ candidate key ⊆ super key

C.super key ⊆ candidate key ⊆ primary key

D.super key ⊆ primary key ⊆ candidate key