CST 204 Database Management Systems

B.Tech. CSE

Semester IV

Viswajyothi College of Engineering and Technology

Sindhu Jose, AP, CSE Dept

MODULE 1

Introduction & Entity Relationship (ER) Model

Sindhu Jose, AP, CSE Dept

Syllabus of Module 1

- ▫ ER model - Basic concepts, entity set & attributes, notations, Relationships and constraints,
- ▫ cardinality, participation, notations, weak entities, relationships of degree 3.

Database Modeling using Entity-Relationship – ER Model

Sindhu Jose, AP, CSE Dept

ER Diagram

- It is not a technical method
- It is a High level conceptual data model
- It is used for conceptual data design of database applications
- It represents a collection of entities and their properties called attributes and relationship between them
- Diagrammatic representation and easy to understand for non technical users.

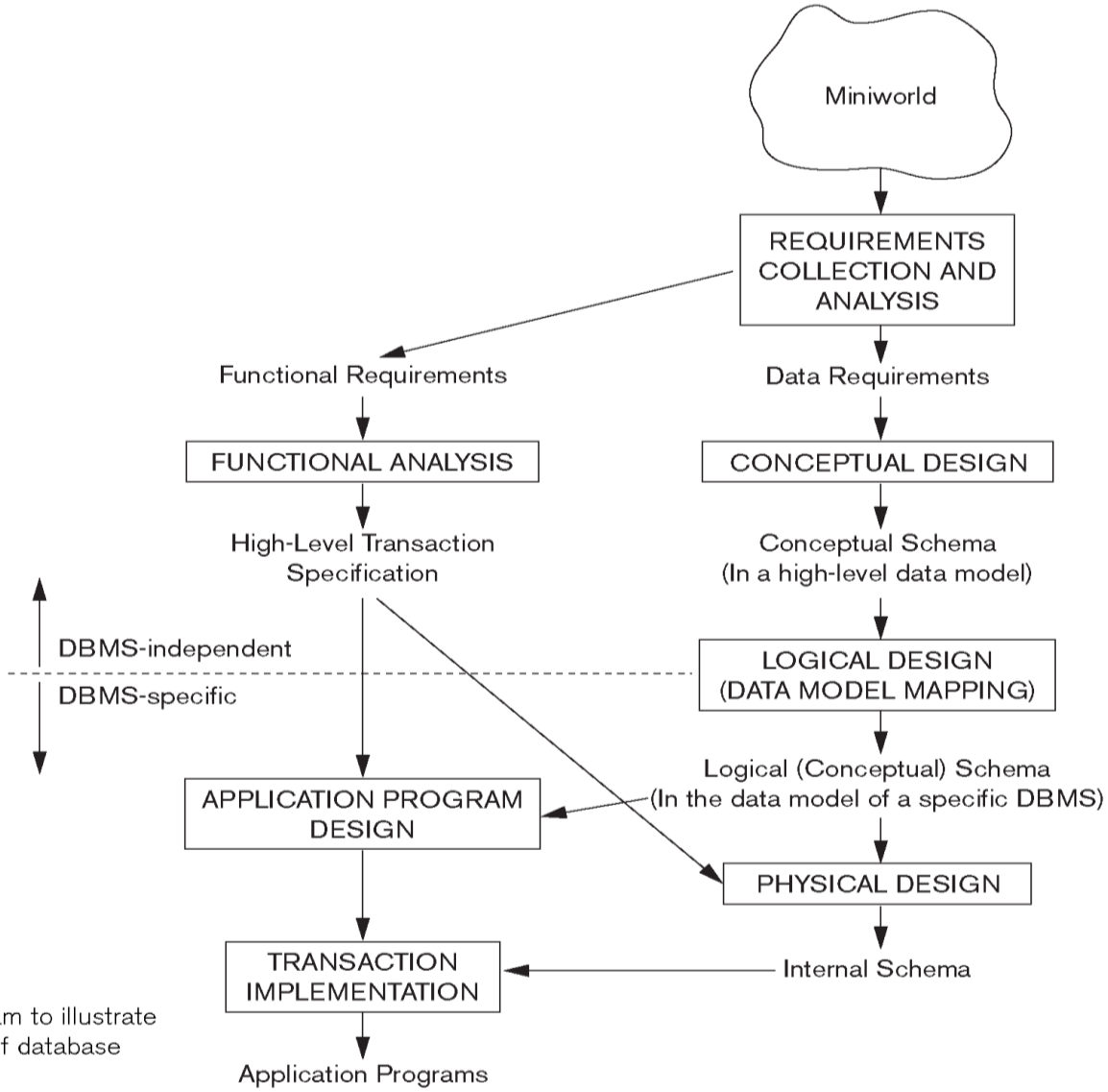


Figure 7.1
A simplified diagram to illustrate the main phases of database design.

Entity

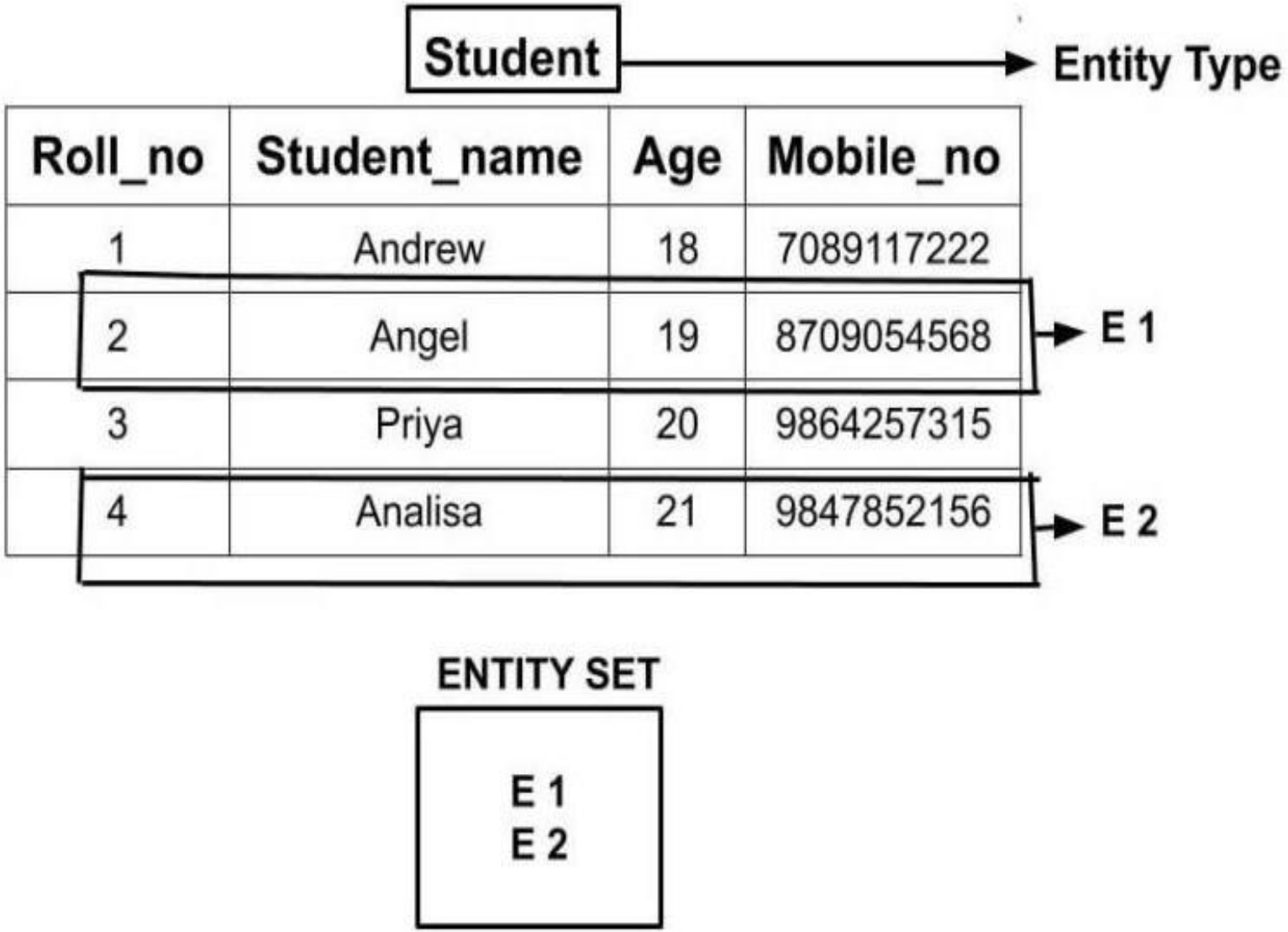
- The basic object that the ER model represents
- A thing in real world with existence

Entity Type

- The entity type is a collection of the entity having similar attributes.
- An entity type in an ER diagram is defined by a name and a set of attributes.

Entity set

- The collection of same type of entities that is their attributes are same is called entity set
- We can say that entity type is a superset of the entity set as all the entities are included in the entity type


Student
Entity Type
Roll_no
Student_name
Age
Mobile_no
1
Andrew
18
7089117222
2
Angel
19
8709054568
E 1
3
Priya
20
9864257315
4
Analisa
21
9847852156
E 2
ENTITY SET
E 1
E 2

Attributes

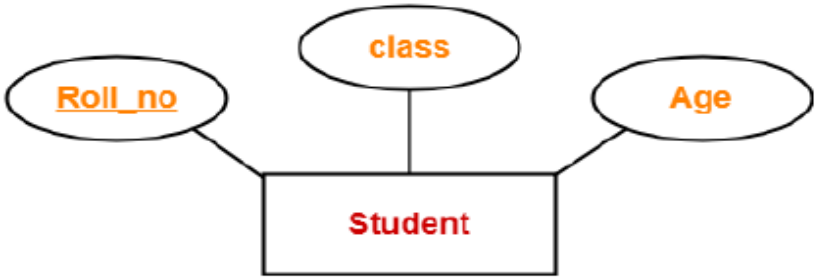
- The properties of entity that basically describes it
- Attributes describes characteristics of entity
- Suppose we have a entity EMPLOYEE and its attributes are ENO, ESAL, ENAME etc..
- Attributes have some set of allowed or permitted values called Domain
- Attributes are represented by OVAL
- Each attribute of an entity set is associated with domain that means the set of values that can be assigned to that attribute for an entity

Types of attribute

- Simple attribute vs Composite attribute
- Single valued vs Multivalued attributes
- Stored vs Derived attributes

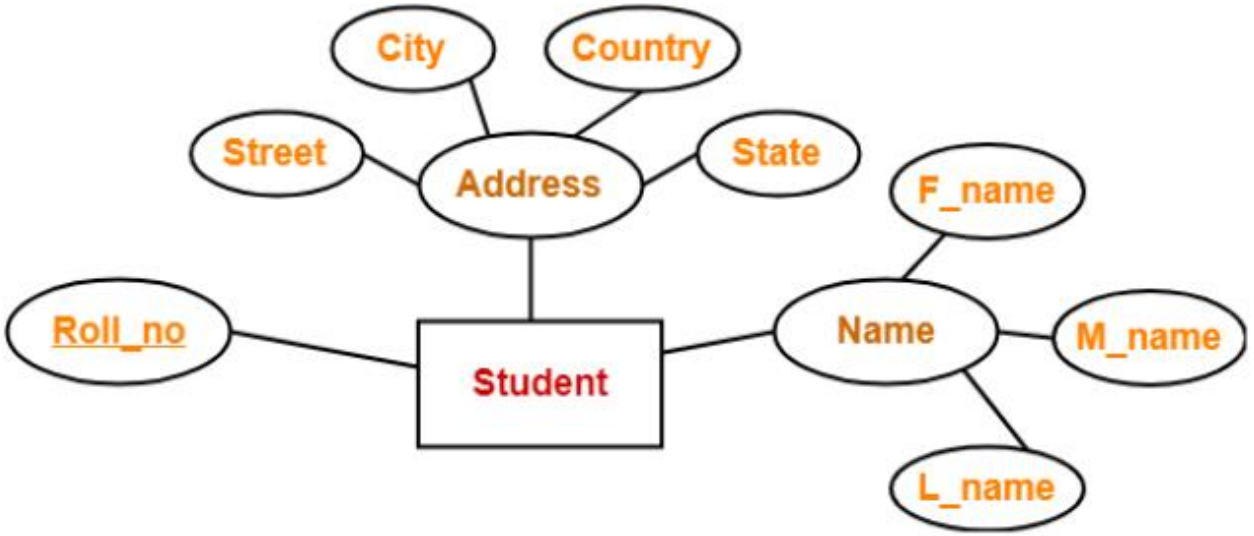
Simple attribute vs Composite attribute

- Simple attributes
  - Attributes which are not divisible that is they cannot be divided
  - Eg: City, State, etc,.



Here, all the attributes are simple attributes as they can not be divided further.

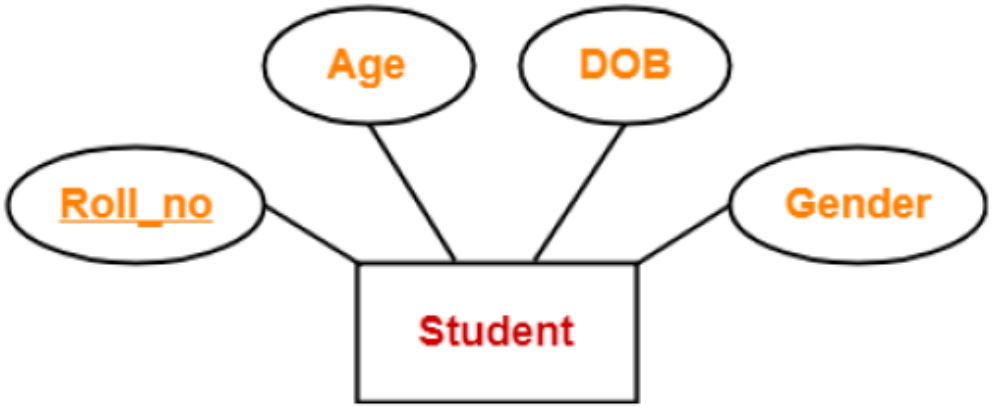
- Composite Attribute
  - Attributes that can be divided into smaller sub parts
  - Example: Name attribute can be divided into FirstName, MiddleName, LastName



Here, the attributes "Name" and "Address" are composite attributes as they are composed of many other simple attributes.

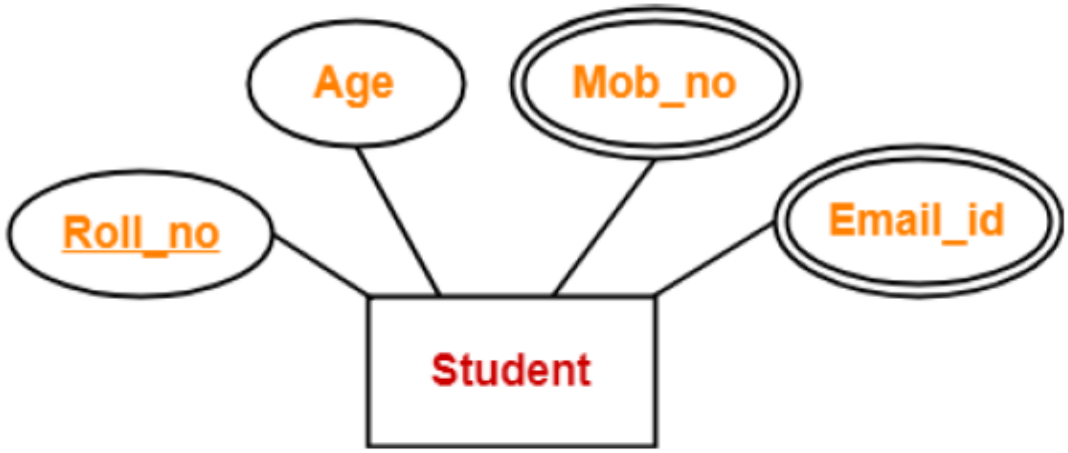
Single valued vs Multivalued Attributea

- Single Valued
  - Attributes which are having single values
  - Example: Age



Here, all the attributes are single valued attributes as they can take only one specific value for each entity.

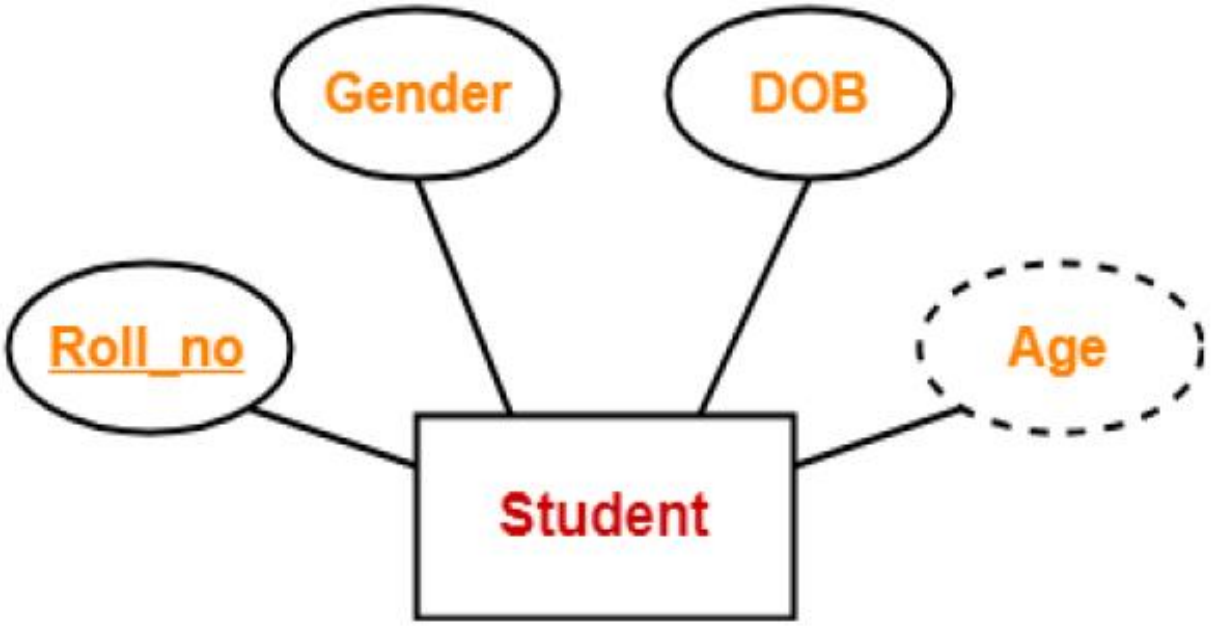
- Multi Valued Attributes
  - Multi valued attributes are those attributes which can take more than one value for a given entity from an entity set.
  - Represented by double oval



Here, the attributes "Mob_no" and "Email_id" are multi valued attributes as they can take more than one values for a given entity.

Stored Vs Derived Attribute

- In some cases, two (or more) attribute values are related ,for example, the Age and Birthdate attributes of a person.
- For a particular person entity, the value of Age can be determined from the current (today's) date and the value of that person's birthdate.
- The Age attribute is hence called a derived attribute and is said to be derivable from the birthdate attribute, which is called a stored attribute .



Here, the attribute “Age” is a derived attribute as it can be derived from the attribute “DOB”.

Complex Attribute

- Complex attribute is a combination of composite and multi-valued attributes.
- Complex attributes are represented by { } and composite attributes are represented by ( ).
- Example: Address_phone attribute will hold both the address and phone_no of any person.
- Example: {(2-A, St-5, Sec-4, Bhilai), 2398124}

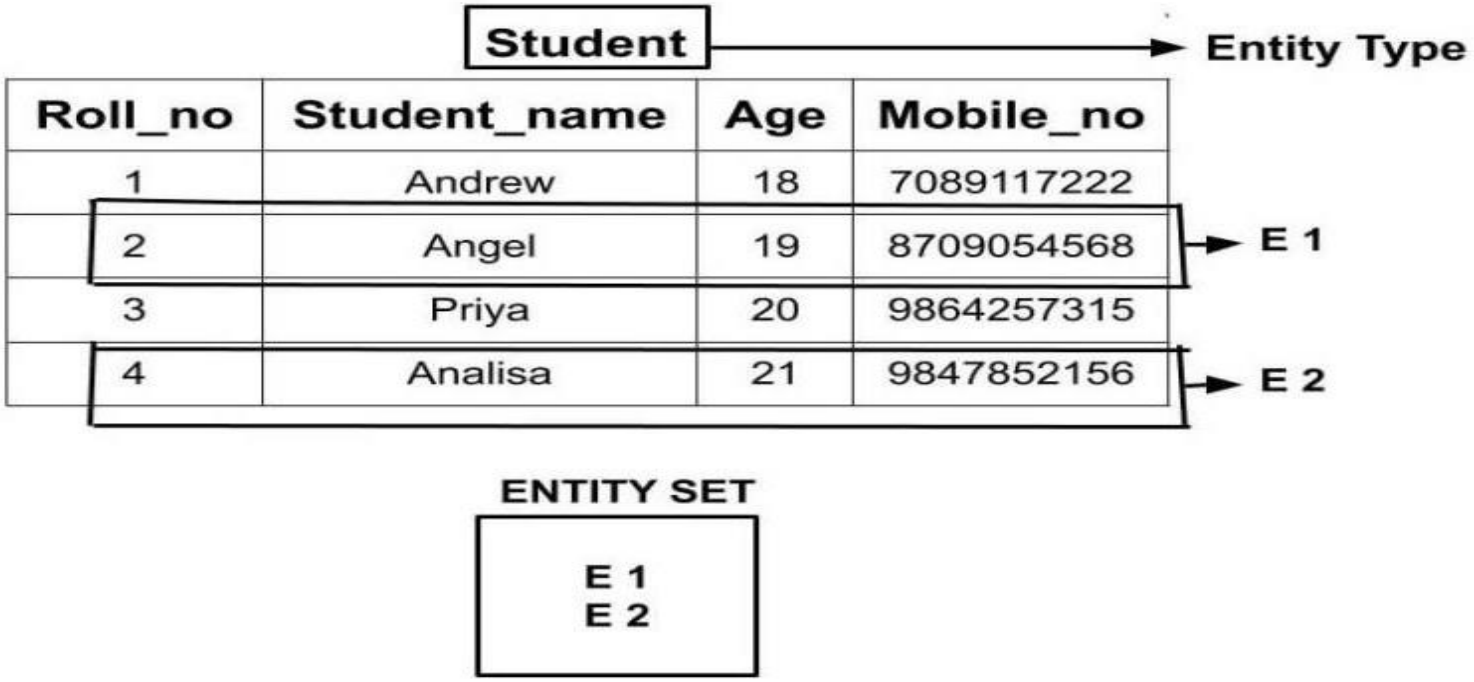
Null Valued Attributes

- Null value is a value which is not inserted but it does not hold zero value.
- The attributes which can have a null value called null valued attributes.
- Example: Mobile_no attributes of a person may not be having mobile phones.

Key attribute in an entity type

- Key attributes will be having a unique value for each entity of that attribute.
- It identifies every entity in the entity set.
- Key attribute will never be a null valued attribute.
- Any composite attribute can also be a key attribute.
- There could be more than one key attributes for an entity type.
- Example: roll_no, enrollment _no

In the table below, **Roll_no** is the **key attribute** because no two students will have same Roll no.



Domain of value set of an attribute

- Domain of an attribute is the allowed set of values of that attribute.
- Example: if attribute is ‘grade’, then its allowed values are A,B,C,F.
- Grade ={A, B,C,F}

Relationship

- Relates two or more distinct entities with a specific
- meaning.
- It is an association between two or more entities of same or different entity set
  - For example, EMPLOYEE John works on the ProductX PROJECT or
  - EMPLOYEE Franklin manages the Research DEPARTMENT.
- **Terms used:**
  - Relationship type,
  - Relationship set,
  - Relationship instances.

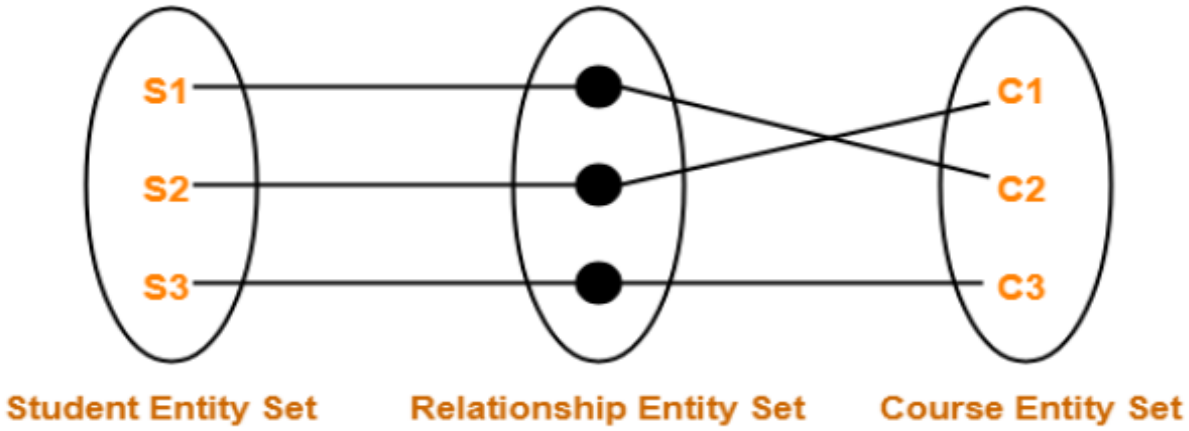
Relationship type

- A set of similar types of relationship

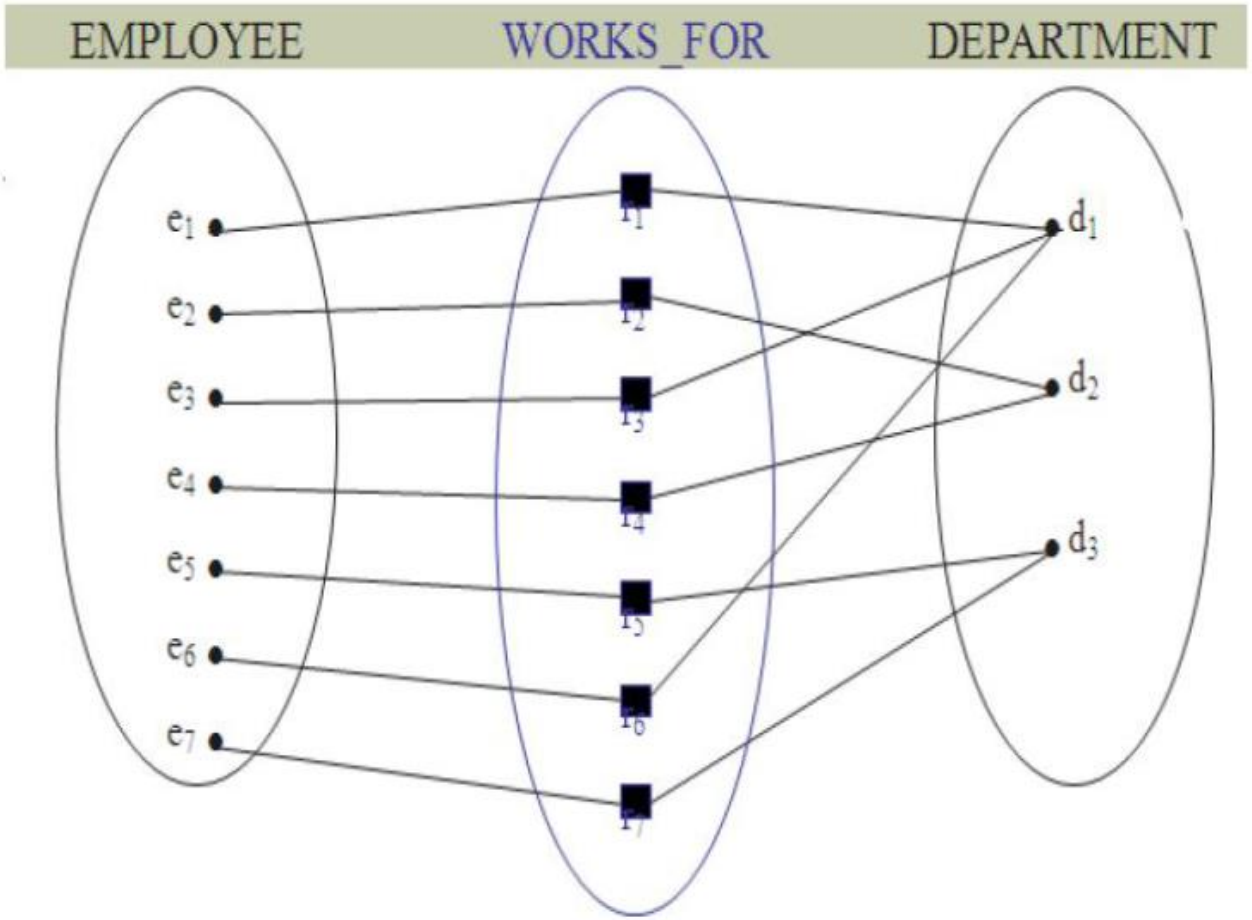
'Enrolled in' is a relationship that exists between entities **Student** and **Course**.

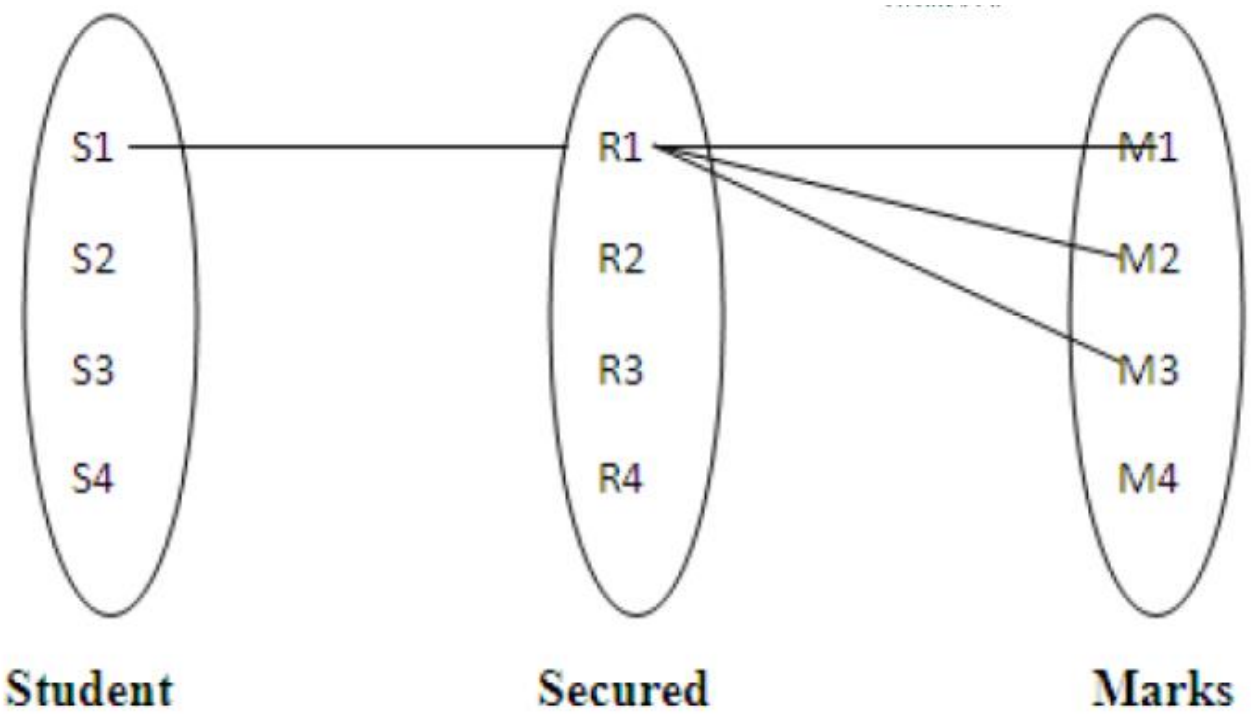
Relationship Set

- A **relationship set** is a set of relationships of the same type.



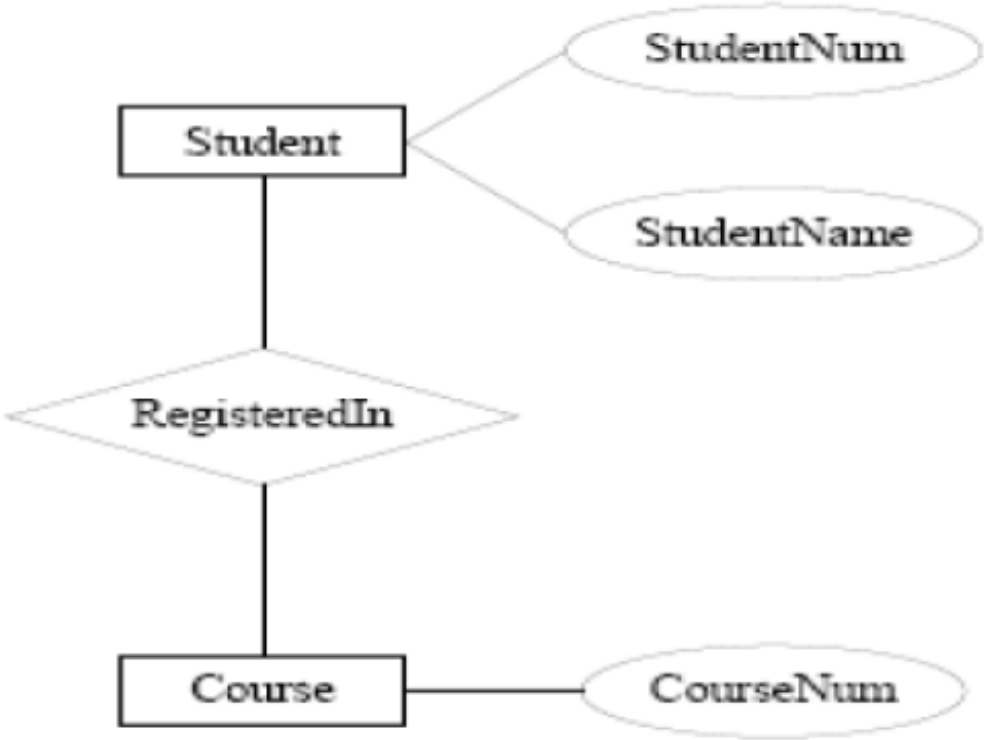
Set Representation of ER Diagram


EMPLOYEE
WORKS_FOR
DEPARTMENT
e1
e2
e3
e4
e5
e6
e7
r1
r2
r3
r4
r5
r6
r7
d1
d2
d3



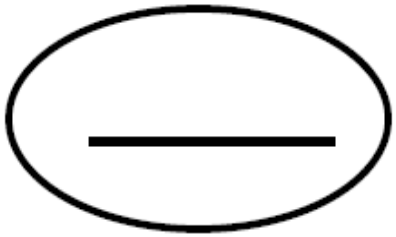
Relationship type: secured
Relationship set: {R1, R2, R3, R4}
Relationship instance : R1

Graphical Representation of Relationship Sets

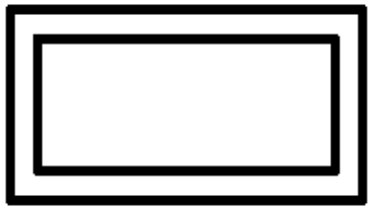


NOTATIONS USED IN E-R DIAGRAM

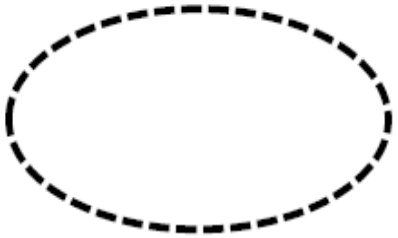
Entity



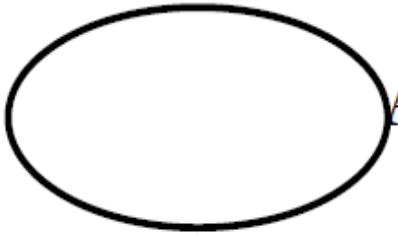
Key attribute



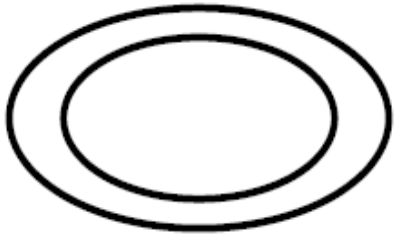
Weak Entity



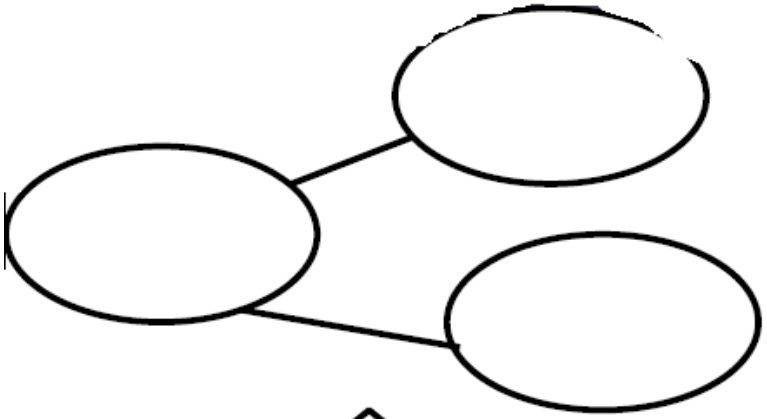
Derived attribute



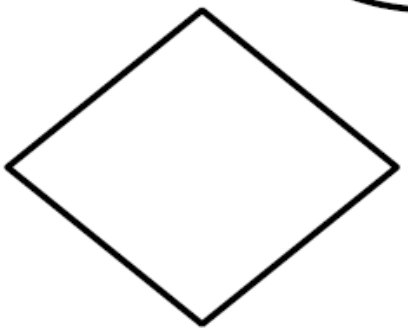
Attribute



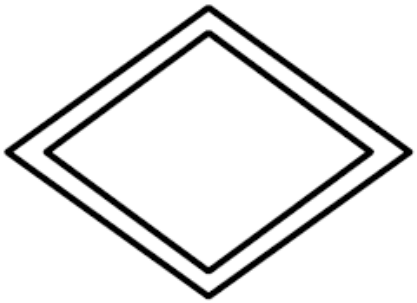
Multivalued Attribute



Composite Attribute



Relationship type



Identifying Relationship

Constraints

- Relationship types usually have certain constraints. Two main types of relationship constraints:
  1. Mapping cardinalities
  2. Participation constraints

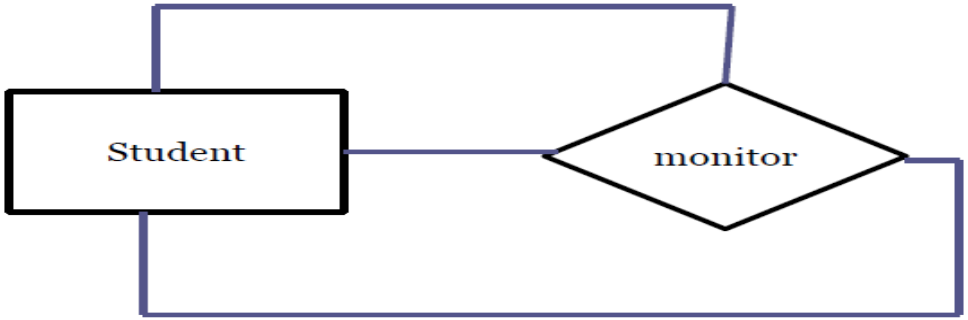
Degree of a relationship

- It is the number of entity set which are participating in a relationship
  - Unary relationship
  - Binary Relationship
  - Ternary Relationship

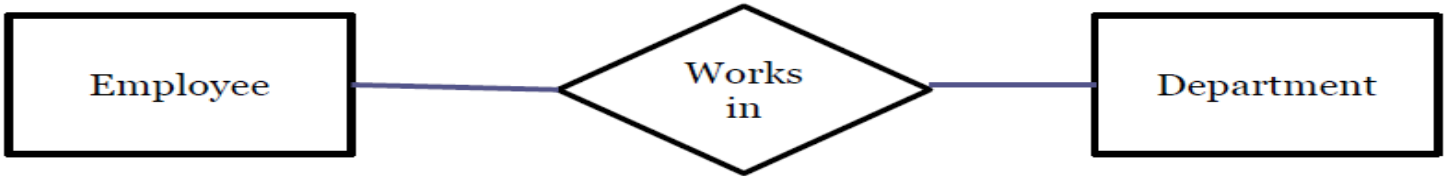
Degree of a Relationship Type

- **The degree of a relationship type** is the number of participating entity types.
- A relationship type of degree two is called **binary,** and one of degree three is called **ternary.**

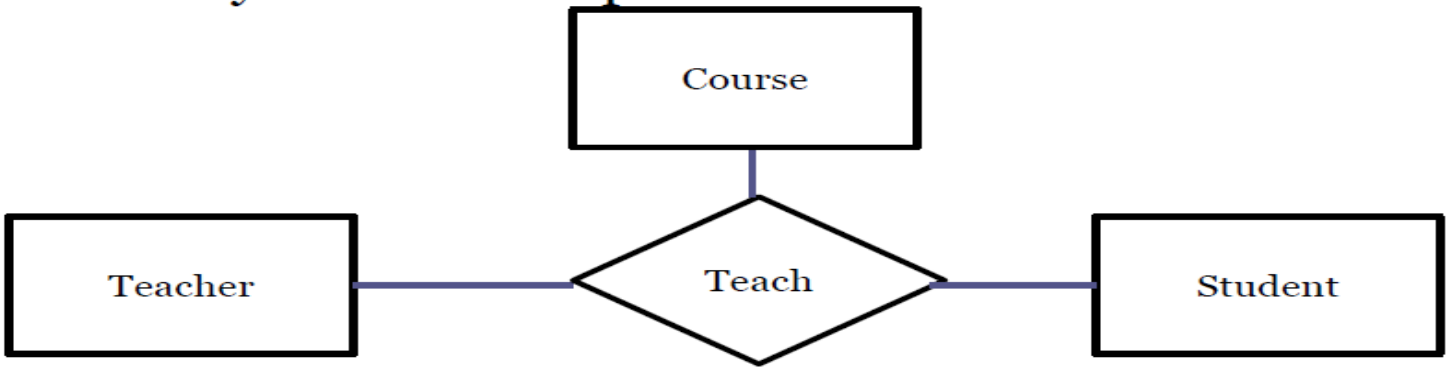
- Unary Relationship



- Binary Relationship



- Ternary Relationship



Recursive relationships

- In some cases the *same entity type participates* more than once in a relationship type in *different roles. In such cases the role name* becomes essential for distinguishing the meaning of the role that each participating entity plays. Such relationship types are called **recursive relationships.**
- The SUPERVISION relationship type relates an employee to a supervisor, where both employee and supervisor entities are members of the same EMPLOYEE entity set. Hence, the EMPLOYEE entity type *participates twice in* SUPERVISION: once in the role of *supervisor (or boss), and once in the role of supervisee (or subordinate).*

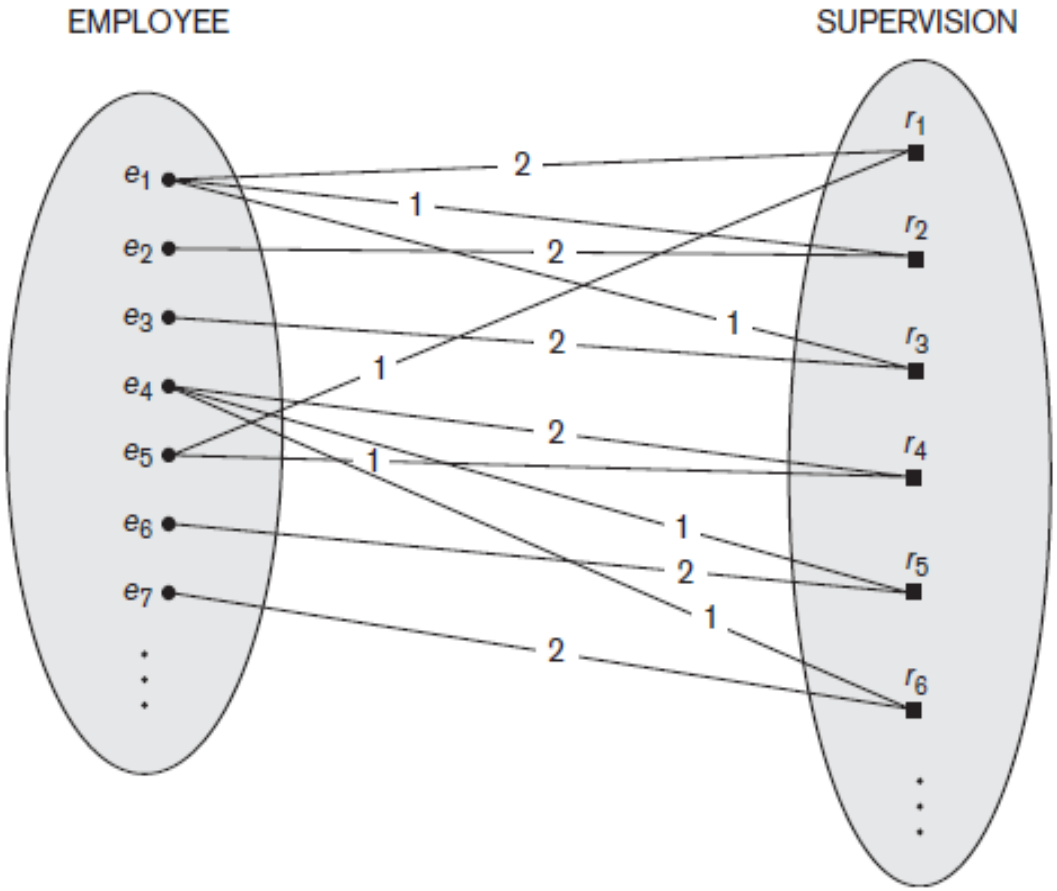


Figure 7.11
A recursive relationship SUPERVISION between EMPLOYEE in the *supervisor* role (1) and EMPLOYEE in the *subordinate* role (2).

Each relationship has

- Name
- Degree
- Cardinality ratio

Cardinality Ratio

- The cardinality ratio for a binary relationship specifies the maximum number of relationship instances to which an entity can take part in it .
- It also specifies number of entities to which other entity can be related by a relationship .

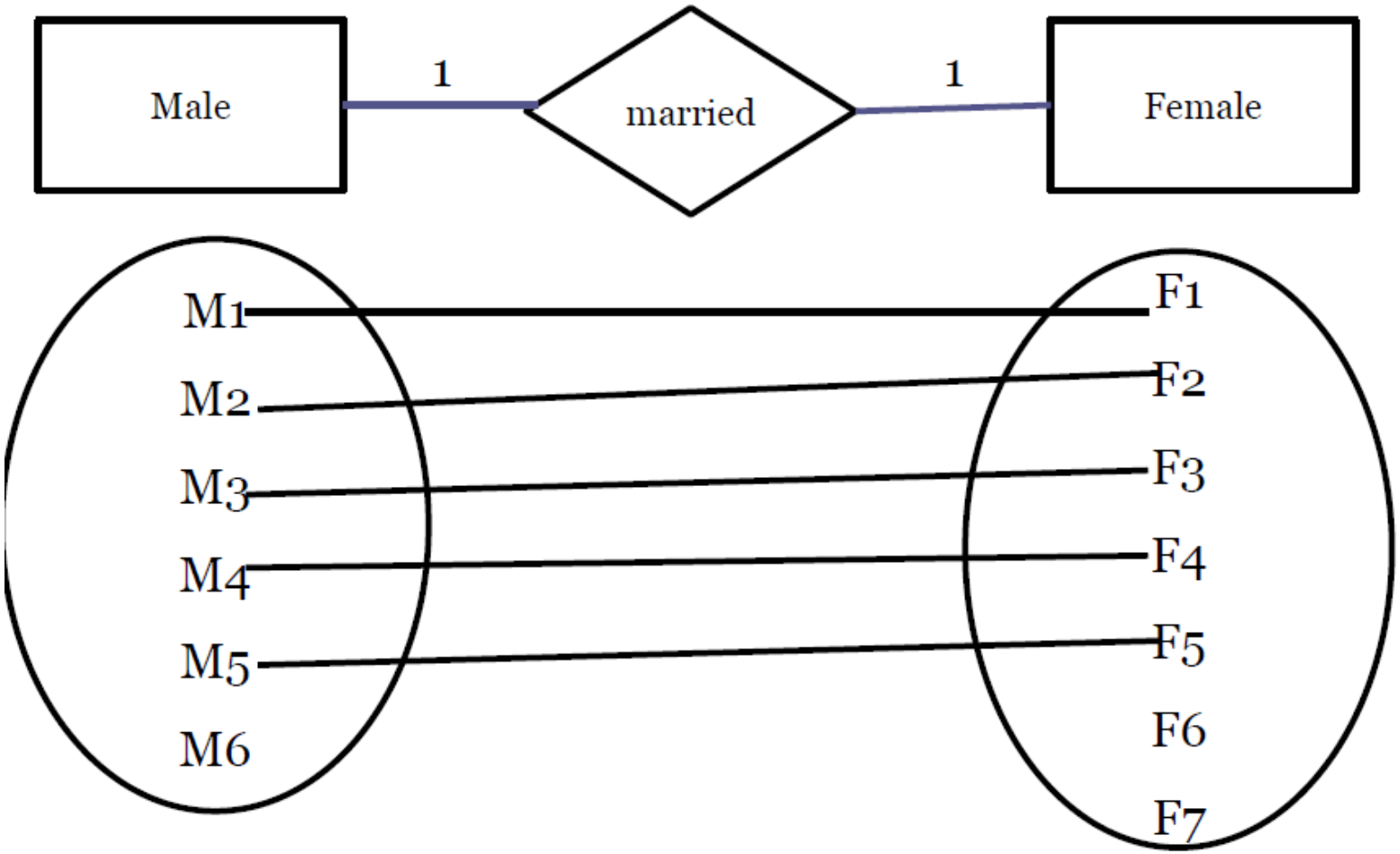
Types

- **One-to-one (1:1)**
- **One-to-many (1: N)**
- **Many-to-one (N: 1)**
- **Many-to-many (M: N)**

- We express cardinality ratio by drawing
  - directed line (→), signifying "one," or an
  - undirected line (—), signifying "many,"

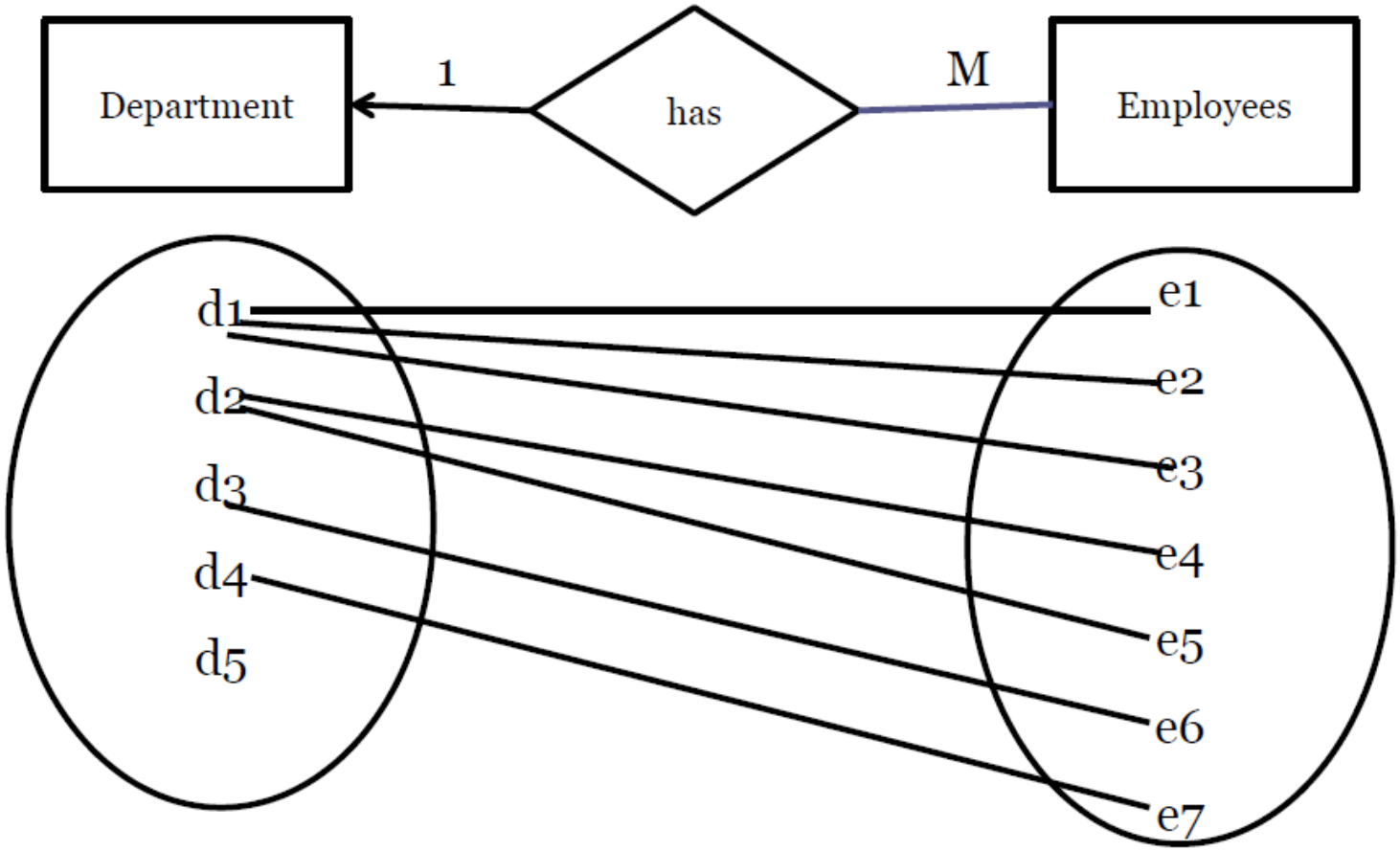
One to One(1:1)

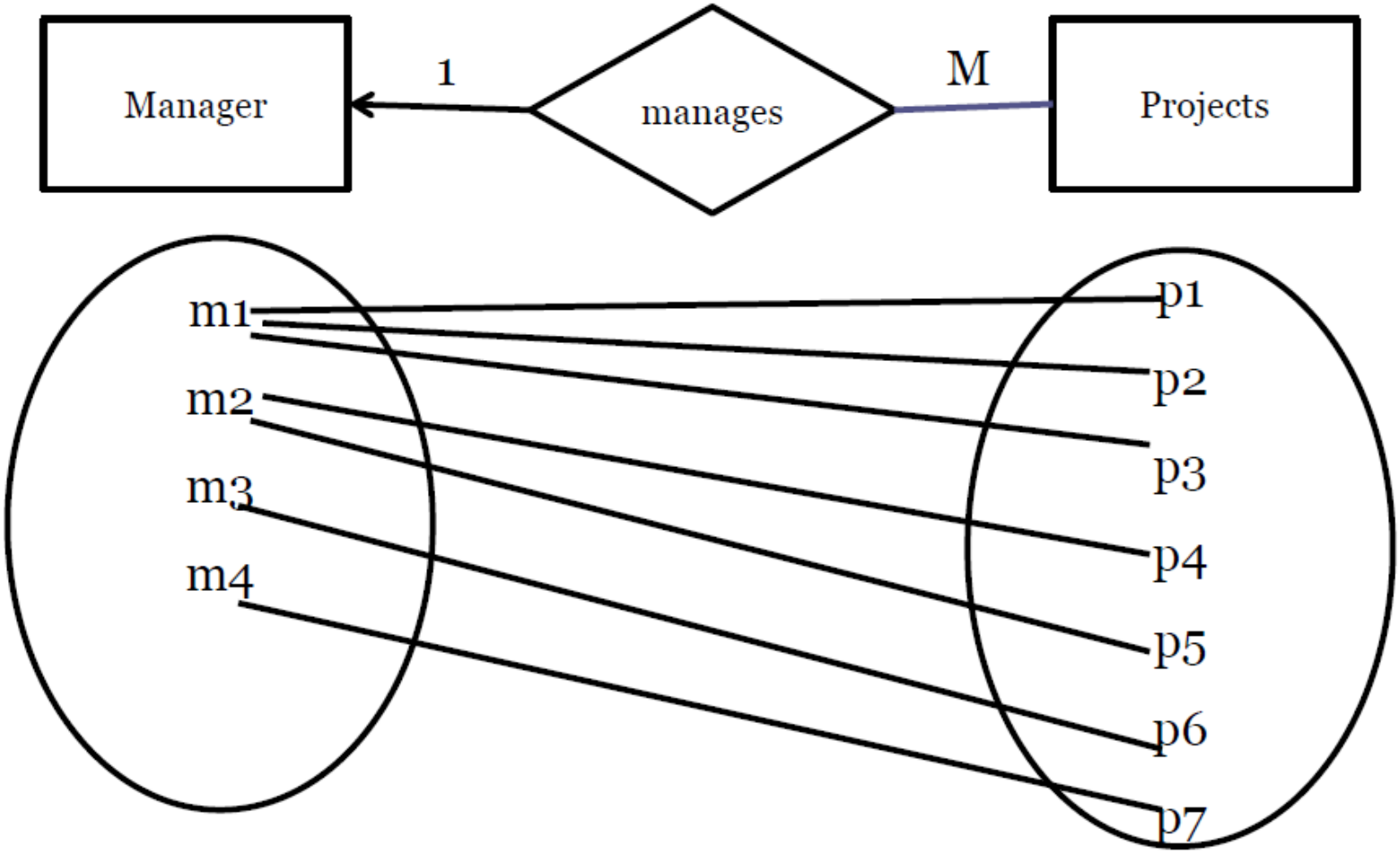
- When only a single instance of an entity is associated with single instance of other entity by a relationship.
- When every entity of one entity set is related to maximum one entity of other entity set.


Male
1
married
1
Female
M1
M2
M3
M4
M5
M6
F1
F2
F3
F4
F5
F6
F7

One to Many (1:M)

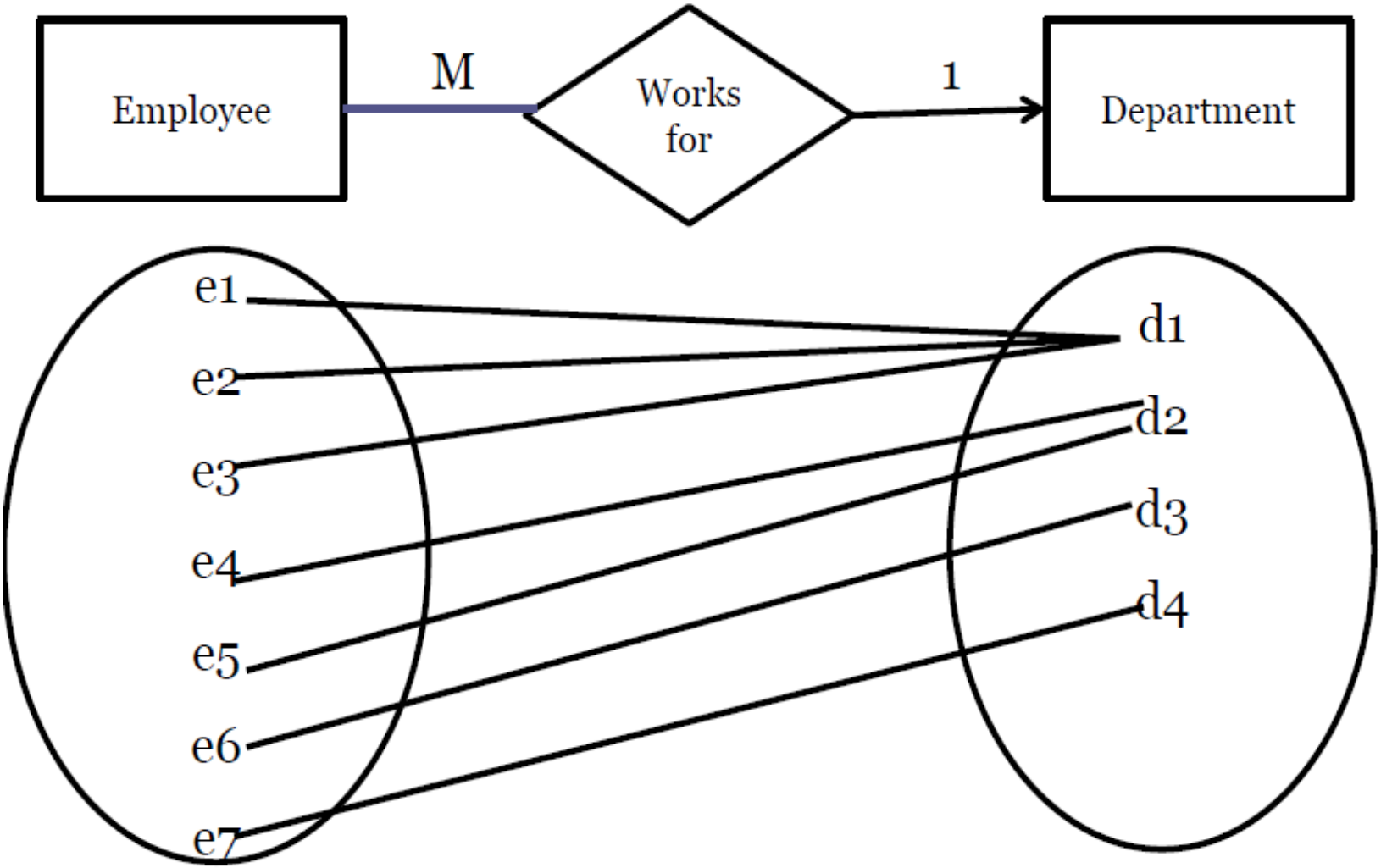
- When every entity of first entity set is related to at most (max) n entities of other entity set then it is one to many


Department
1
has
M
Employees
d1
d2
d3
d4
d5
e1
e2
e3
e4
e5
e6
e7


Manager
1
manages
M
Projects
m1
m2
m3
m4
p1
p2
p3
p4
p5
p6
p7

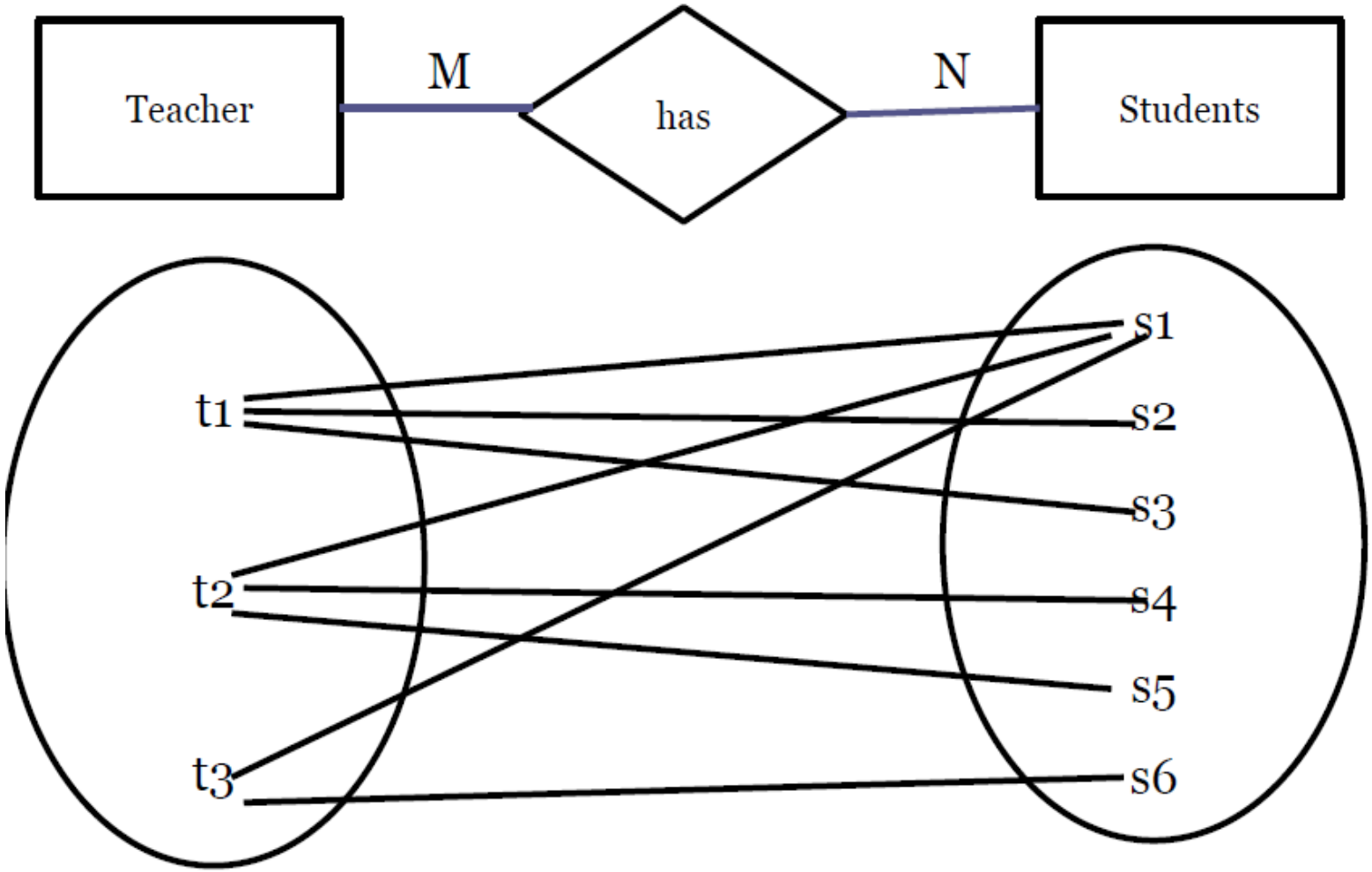
Many to One (M:1)

- When many entities of first entity set is related to 1 entity of other entity set then it is many to one


Employee
M
Works for
1
Department
e1
e2
e3
e4
e5
e6
e7
d1
d2
d3
d4

Many to Many(M:N)

- When many occurrences of one entity is related to many occurrences of another entity


Teacher
M
has
N
Students
t1
t2
t3
s1
s2
s3
s4
s5
s6

Participation Constraints and Existence Dependencies

- The **participation constraint** specifies whether the existence of an entity depends on its being related to another entity via the relationship type.

- This constraint specifies the *minimum* number of relationship instances that each entity can participate in, and is some times called the **minimum cardinality constraint.**

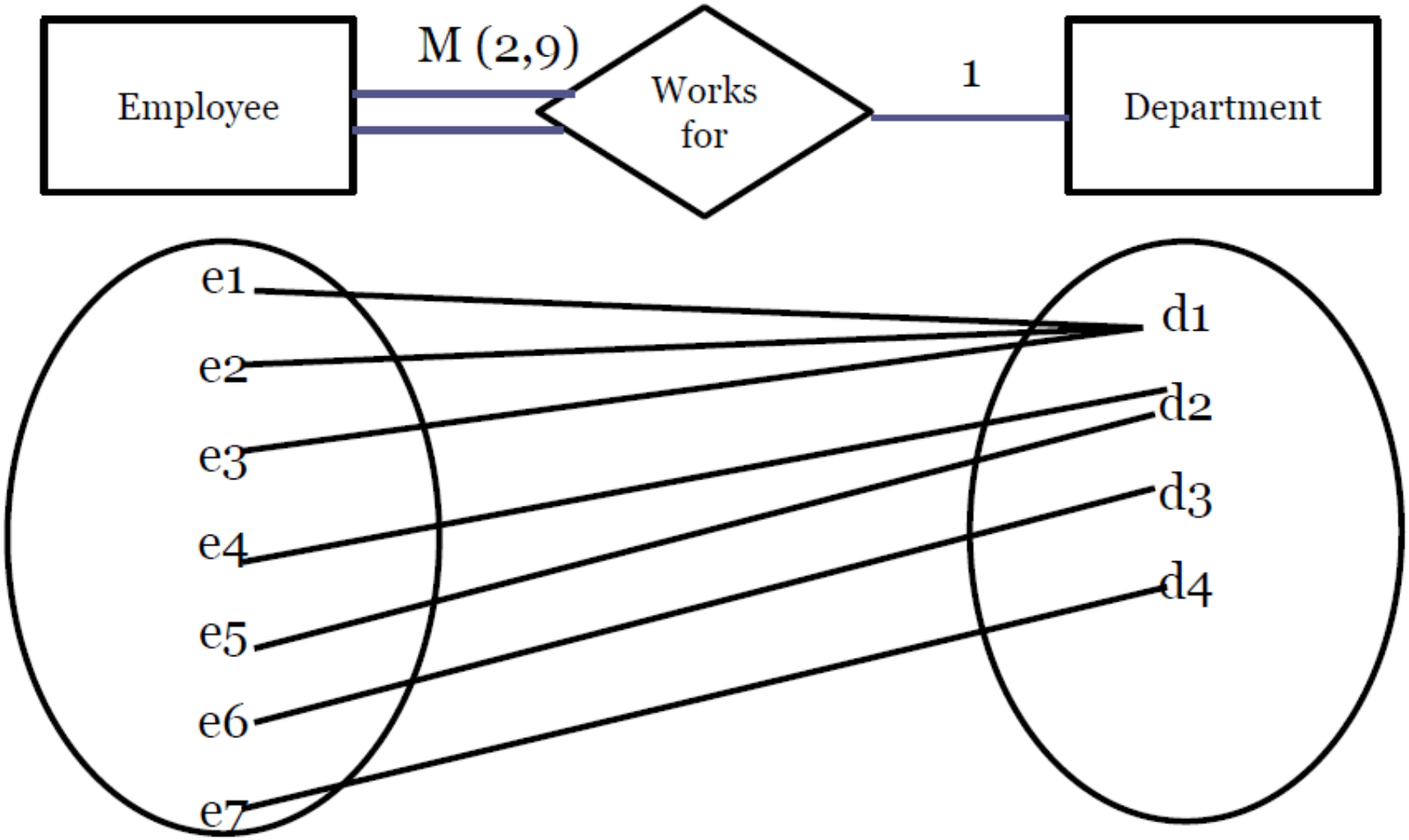
- There are two types of participation constraints

  1. Total Participation

  2. Partial Participation

Total participation

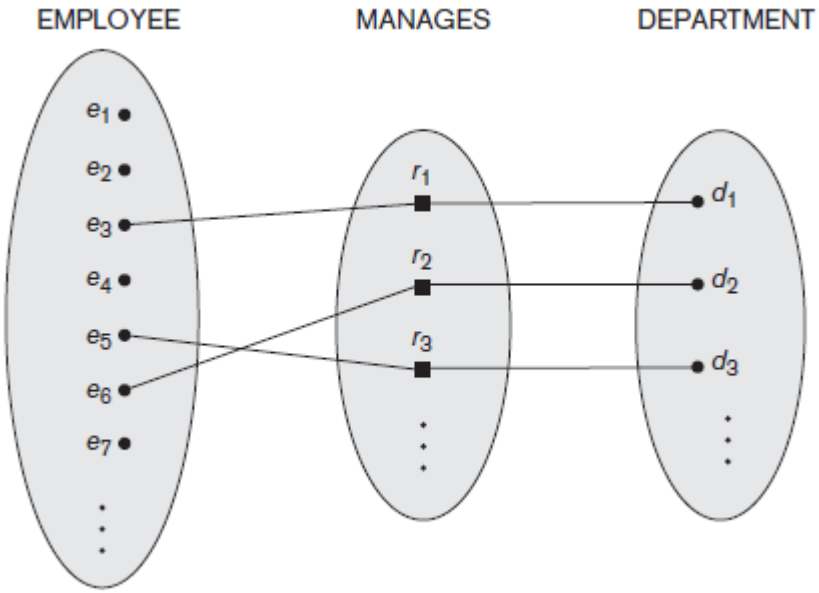
- Every entity in the entity type participates in at least one relationship in the relationship type.
- Example: If a company policy states that *every employee must work for a department, then an employee* entity can exist only if it participates in at least one WORKS_FOR relationship instance . Thus, the participation of EMPLOYEE in WORKS_FOR is called **total participation.**
- **Total participation** is also called **existence dependency.**
- It is represented by double lines.
- Minimum and maximum cardinality represented inside paranteses(m,n)


Employee
M (2,9)
Works for
1
Department
e1
e2
e3
e4
e5
e6
e7
d1
d2
d3
d4

Partial participation

- Some entities may not participate in any relationship in the relationship type.
- Example: In the figure, we do not expect every employee to manage a department, so the participation of EMPLOYEE in the MANAGES relationship type is **partial.**
- Represented by single line.


EMPLOYEE
1
MANAGES
1
DEPARTMENT


EMPLOYEE
MANAGES
DEPARTMENT
e1
e2
e3
e4
e5
e6
e7
r1
r2
r3
d1
d2
d3

ENTITY TYPES

Strong entity type

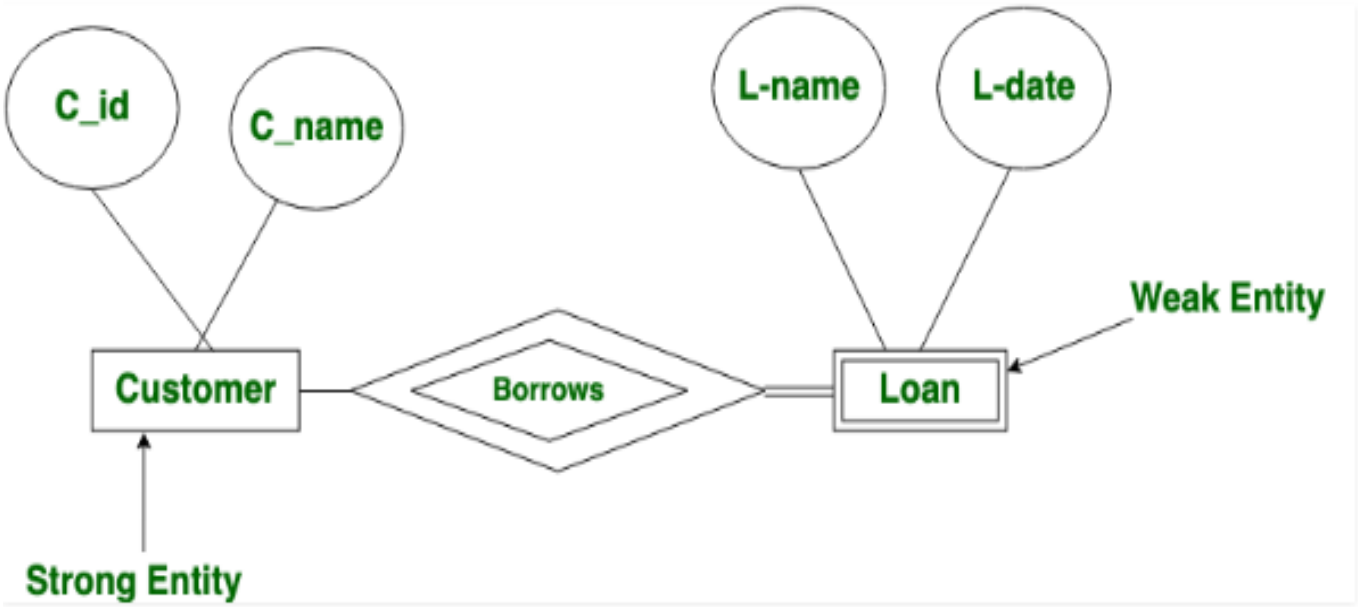
- Entity types that have at least one key attribute.
- A strong entity is not dependent of any other entity in the schema.
- A strong entity will always have a primary key.
- Strong entities are represented by a single rectangle.
- The relationship of two strong entities is represented by a single diamond.
- Various strong entities, when combined together, create a strong entity set.

Weak entity type

- Entity type that does not have any key attribute.
- A weak entity is dependent on a strong entity to ensure the its existence.
- Unlike a strong entity, a weak entity does not have any primary key.
- It instead has a partial discriminator key.
- A weak entity is represented by a double rectangle. The relation between one strong and one weak entity is represented by a **double diamond.**

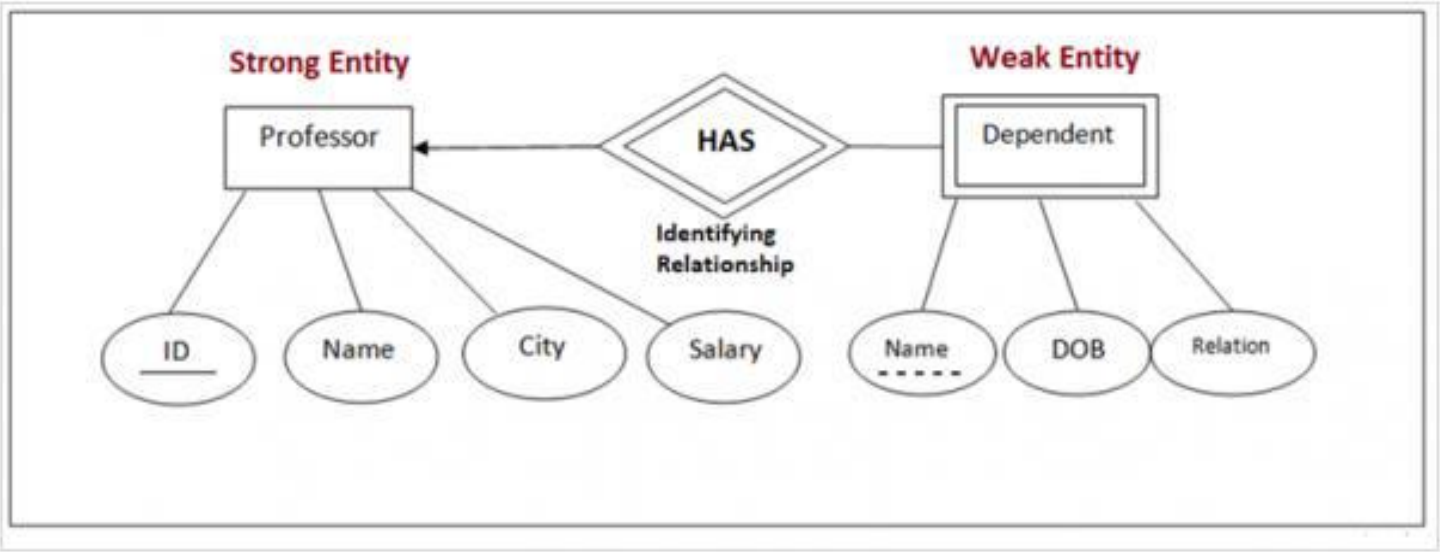
Difference between Strong and Weak Entity:

S.NO	Strong Entity	Weak Entity
1.	Strong entity always has primary key.	While weak entity has partial discriminator key.
2.	Strong entity is not dependent of any other entity.	Weak entity is depend on strong entity.
3.	Strong entity is represented by single rectangle.	Weak entity is represented by double rectangle.
4.	Two strong entity's relationship is represented by single diamond.	While the relation between one strong and one weak entity is represented by double diamond.
5.	Strong entity have either total participation or not.	While weak entity always has total participation.


C_id
C_name
L-name
L-date
Weak Entity
Customer
Borrows
Loan
Strong Entity

Identifying Relationship

- It links the strong and weak entity and is represented by a double diamond sign.
- Let us see with an example to link both the entities using Identifying Relationships:

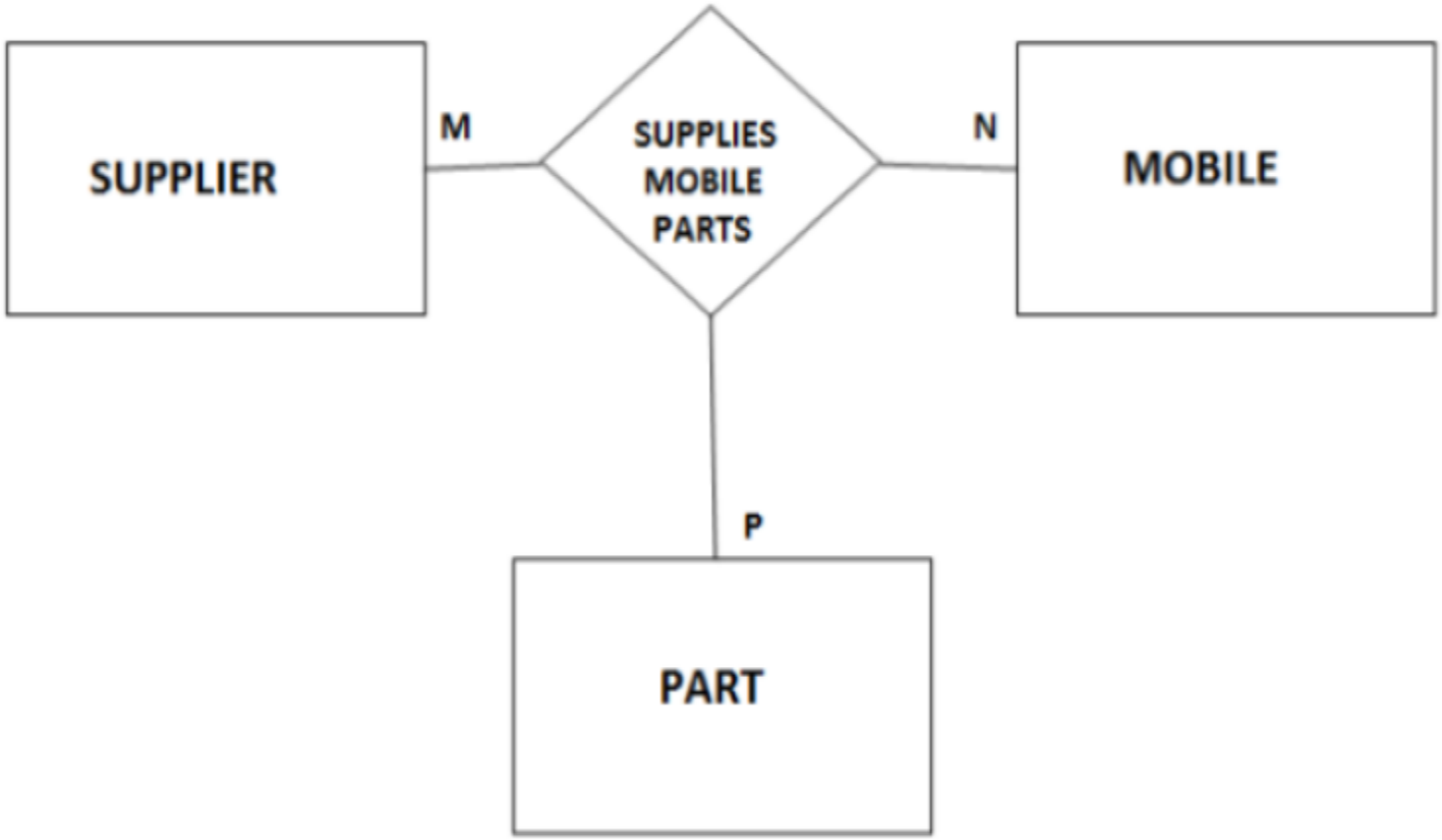


Above we saw that, Dependent Name could not exist on its own but in relationship to a Professor.

Professor	Strong Entity
Dependent	Weak Entity
Partial Key (Weak Entity)	Name
Primary Key (Strong Entity)	ID

Relationship of degree 3

- In Ternary relationship three different Entities takes part in a Relationship.
- Relationship Degree = 3
- For Example: Consider a Mobile manufacture company. Three different entities involved:
- **Mobile** - Manufactured by company.
- **Part** - Mobile Part which company get from Supplier.
- **Supplier** - Supplier supplies Mobile parts to Company.
- Mobile, Part and Supplier will participate simultaneously in a relationship. because of this fact when we consider cardinality we need to consider it in the context of two entities simultaneously relative to third entity.


SUPPLIER
M
SUPPLIES
MOBILE
PARTS
N
MOBILE
P
PART

Q) 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

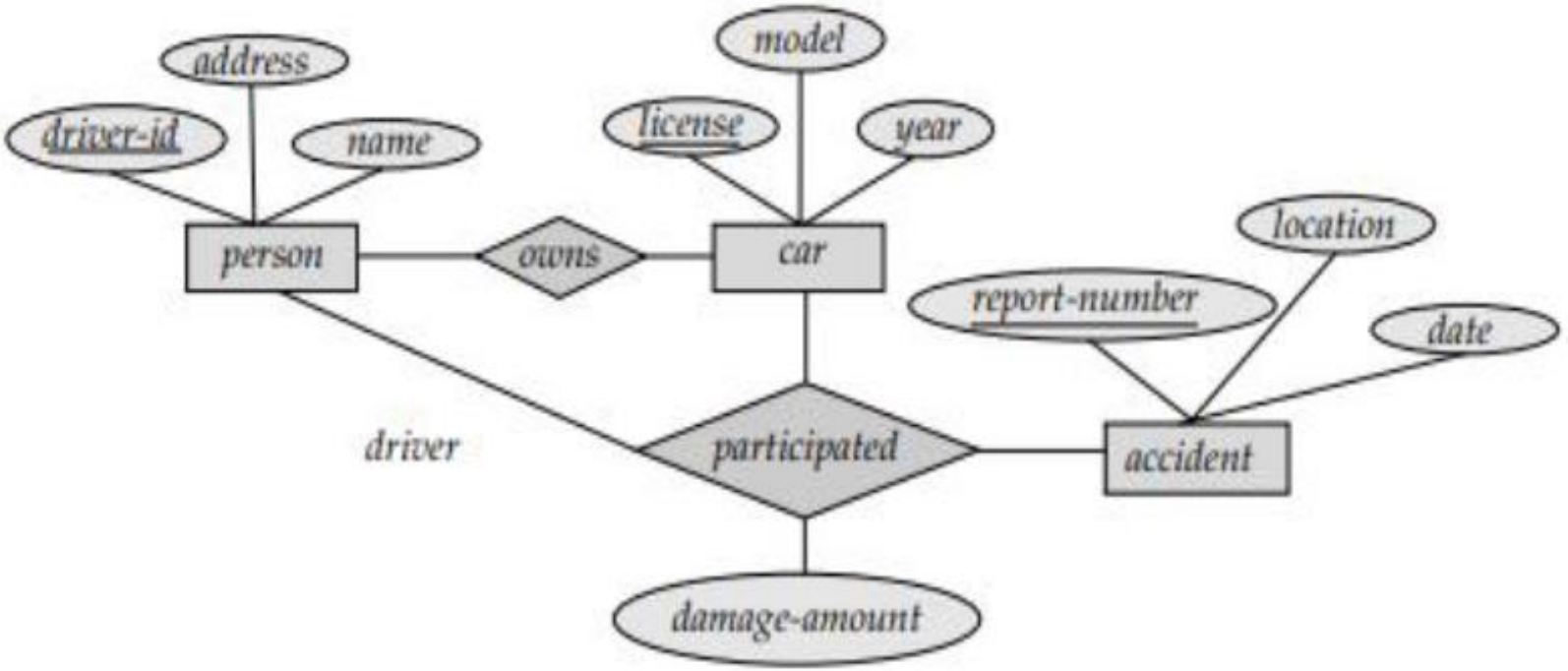


Figure 2.1 E-R diagram for a Car-insurance company.

Q)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

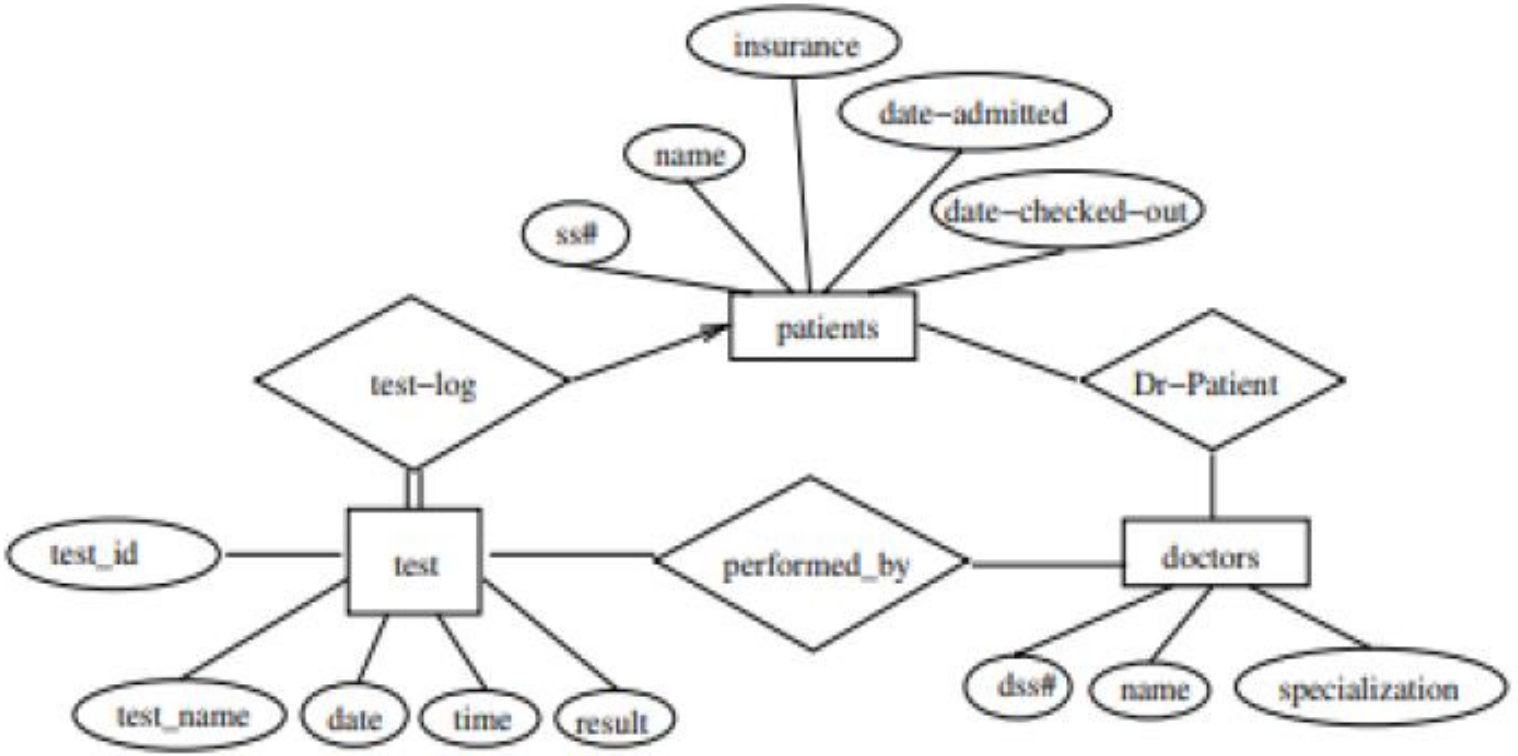


Figure 2.2 E-R diagram for a hospital.

Q)A university registrar's office maintains data about the following entities: (a) courses, including number, title, credits, syllabus, and prerequisites; (b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom; (c) students, including student-id, name, and program; and (d) 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 forthe registrar's office. Document all assumptions that you make about the mapping constraints.

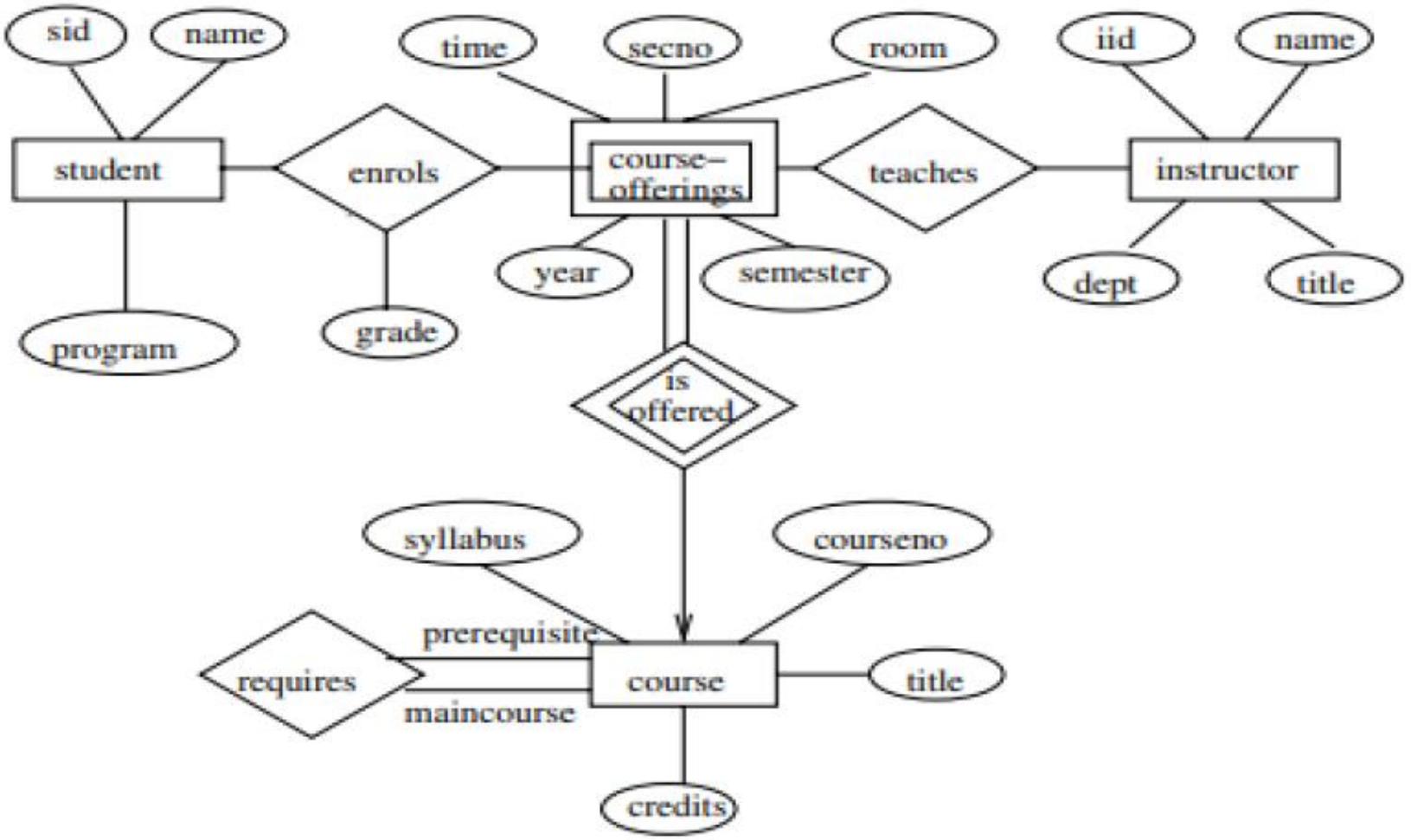


Figure 2.3 E-R diagram for a university.

Q)Consider a database used to record the marks that students get in different exams of different course offerings. Construct an E-R diagram that models exams as entities, and uses a ternary relationship, for the above database.

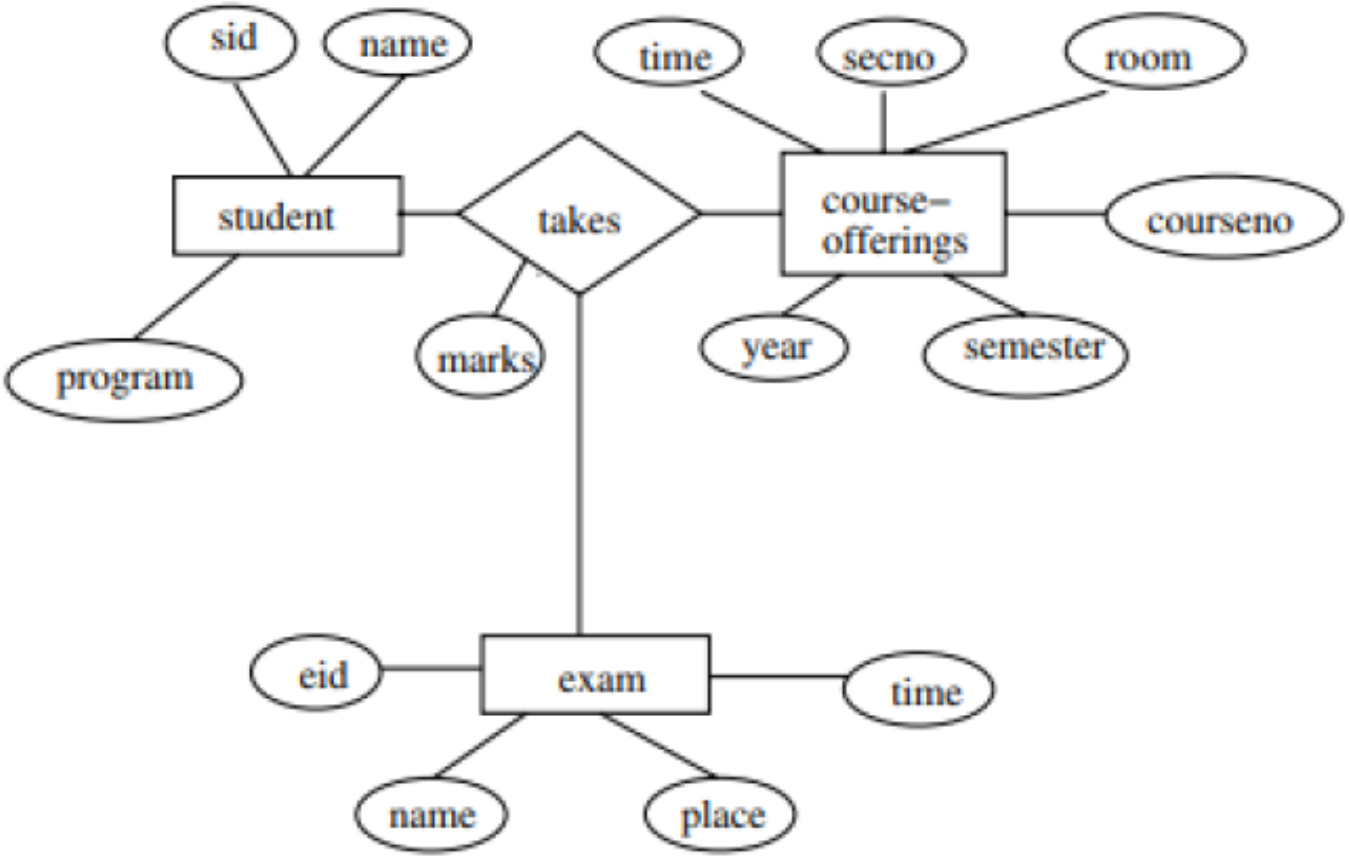


Figure 2.4 E-R diagram for marks database.