DATA BASE MODELLING USING ENTITY RELATIONSHIP MODELLING

SONY P, Assistant Professor

Model Engineering College

May 9, 2021

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

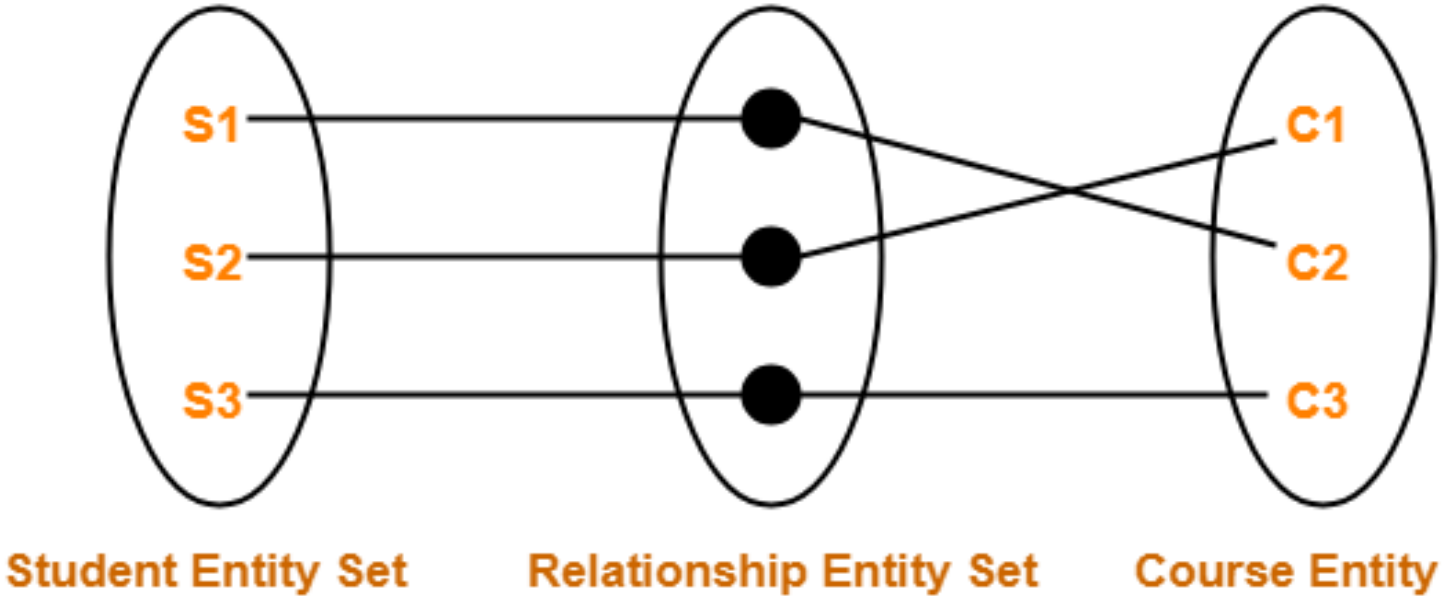
EMPLOYEE Franklin **manages** the Research DEPARTMENT.

Relationships of the same type are grouped or typed into a relationship type.

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

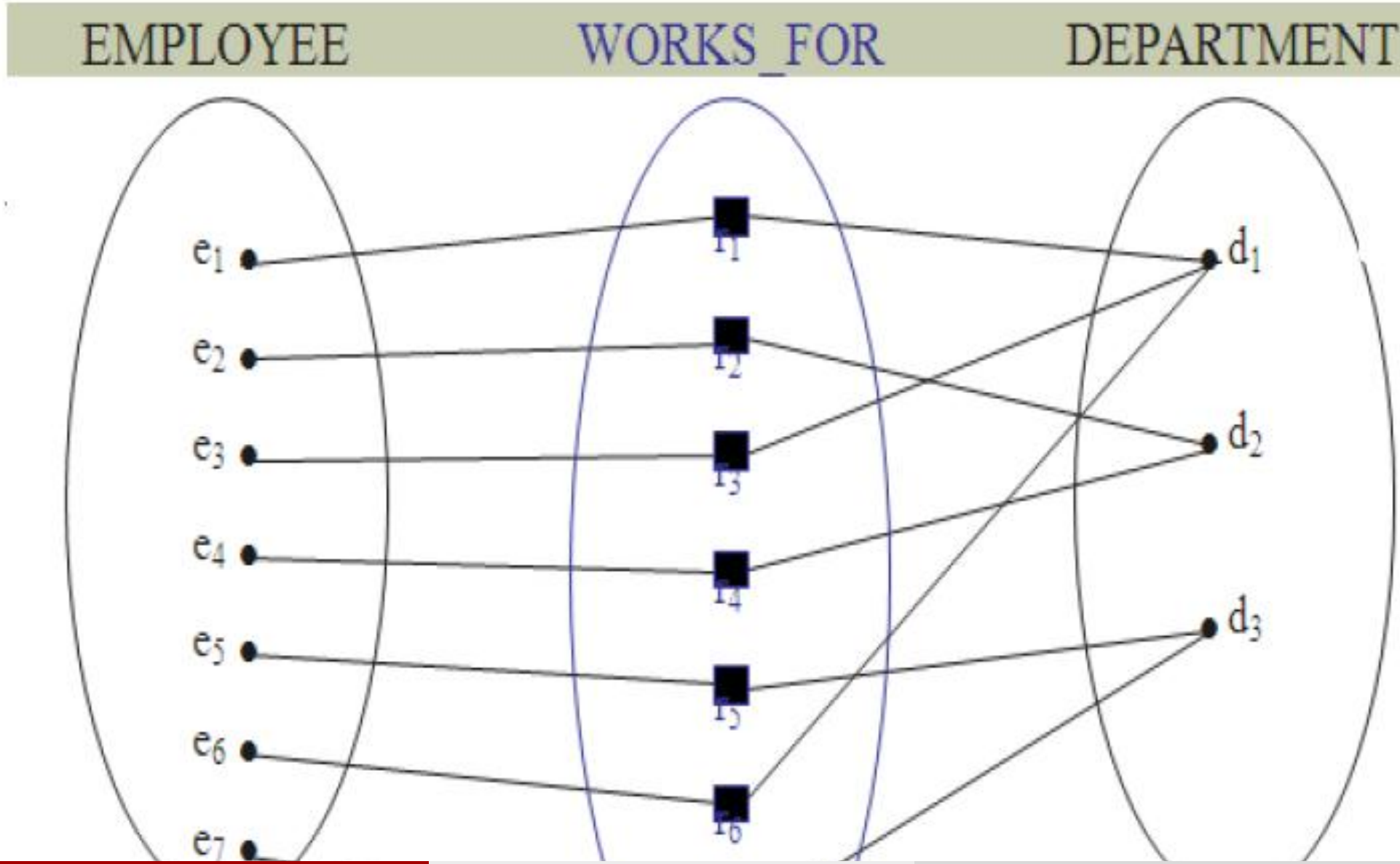
Figure: Relationshiptype

Realtionship Type is set of relationship of the same type



Set Representation of ER Diagram

Realtionship Type is set of relationship of the same type



Relationship Type

1. Relationship Type is the **schema description** of a relationship
2. Identifies the relationship name and the participating entity types
   Also identifies certain relationship constraints
   Relationship Set
3. The current set of relationship instances represented in the database
4. The current state of a relationship type

The company is organized into DEPARTMENTs. Each department has a name, number and an employee who **manages** the department. We keep track of the start date of the department manager.

A department may have several locations. Each department **controls** a number of PROJECTs. Each project has a unique name, unique number and is located at a single location

We store each EMPLOYEE's social security number, address, salary, sex, and birthdate. Each employee **works for** one department but may **work on** several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the direct **supervisor** of each employee.

Each employee **may have a number of DEPENDENTs.** For each dependent, we keep track of their name, sex, birthdate, and relationship to the employee.

Relationship

1. WORKS_FOR between(EMPLOYEE ,DEPARTMENT)
2. MANAGES (also between EMPLOYEE, DEPARTMENT)
3. CONTROLS (between DEPARTMENT, PROJECT)
4. WORKS_ON (between EMPLOYEE, PROJECT)
5. SUPERVISION (between EMPLOYEE (as subordinate), EMPLOYEE (as supervisor))
6. DEPENDENTS_OF (between EMPLOYEE, DEPENDENT)

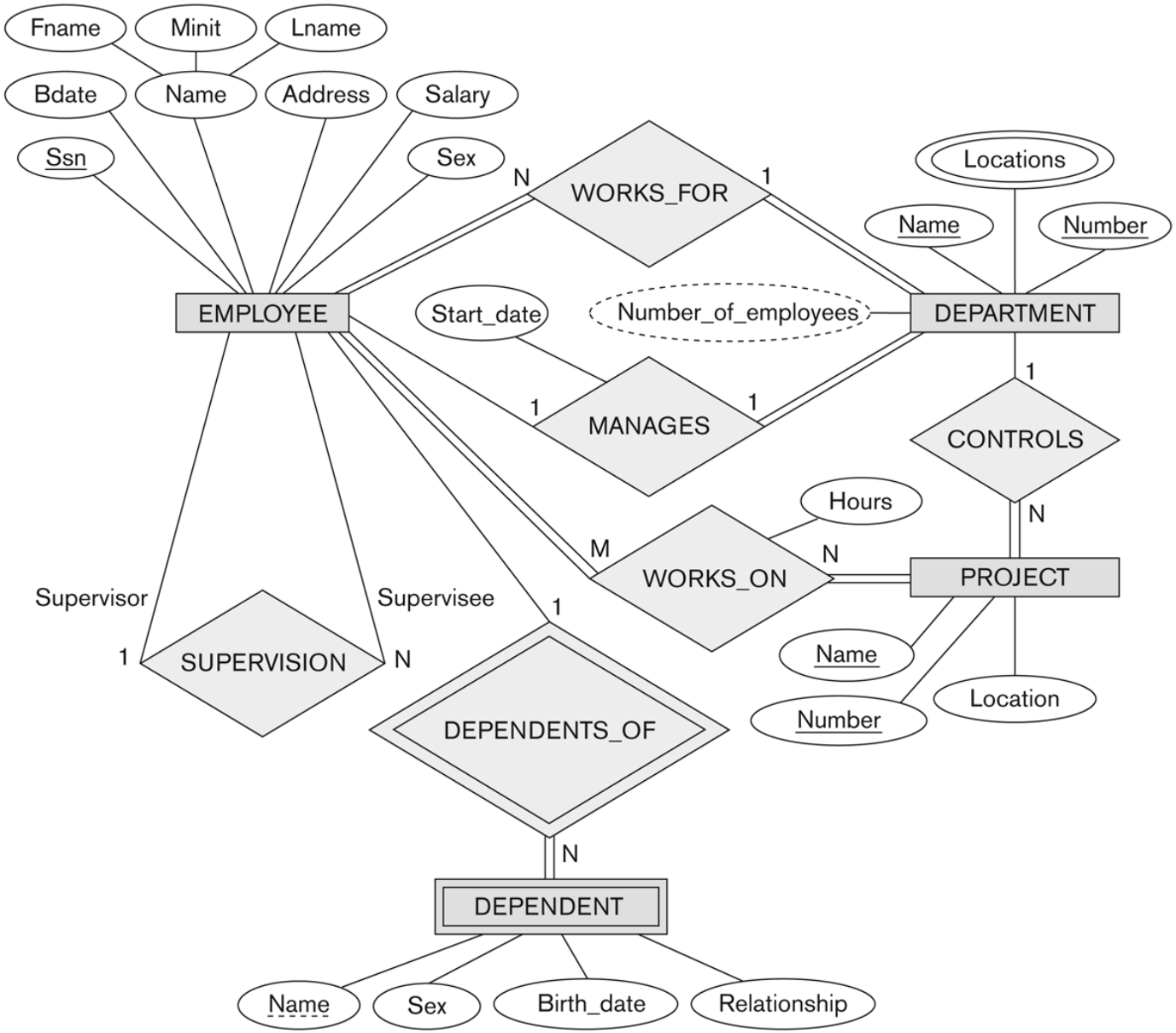


Figure 3.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter.

In refined design, some attributes from the initial entity types are refined into relationships

1. Manager of DEPARTMENT **MANAGES**
2. Works_on of EMPLOYEE **WORKS_ON**
3. Department of EMPLOYEE **WORKS_FOR**
4. more than one relationship type can exist between the same participating entity types **MANAGES and WORKS_ FOR** are distinct relationship types between **EMPLOYEE and DEPARTMENT** Different meanings and different relationship instances

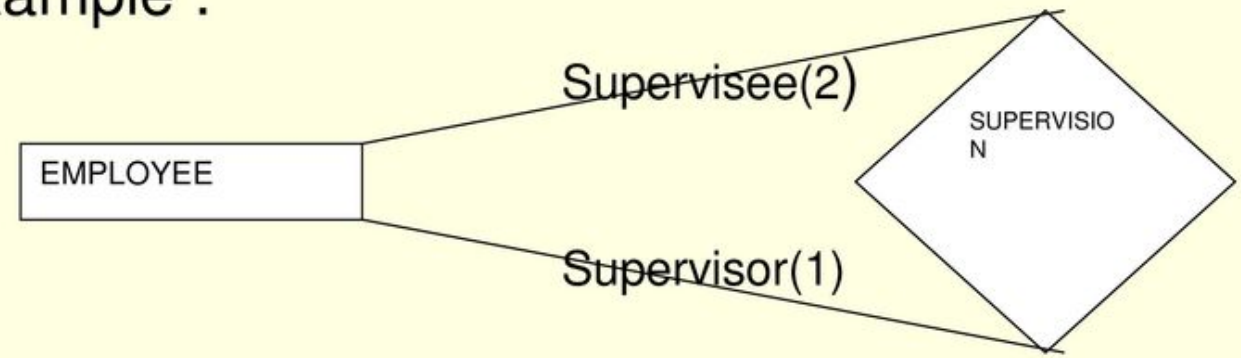
Recursive Relationship Type

1. An relationship type whose with the same participating entity type in distinct roles
2. Example the **SUPERVISION** relationship
3. EMPLOYEE participates twice in two distinct roles
4. **supervisor (or boss) role**
5. **supervisee (or subordinate) role**
6. Each relationship instance relates two distinct EMPLOYEE entities
7. One employee in **supervisor role**
8. One employee in **supervisee role**

Recursive Relationships:

The same entity type participates more than once in a relationship type in different roles. Each role is given a name.

Example :



Roles are optional – they clarify semantics of a relationship

Weak Entity Types

1. An entity that does not have a **key attribute**
2. A **weak entity** must participate in an **identifying relationship** type with an **owner or identifying entity type**
3. Weak Entities are identified by the combination of: A **partial key** of the weak entity type and **key attribute of the Owner entity** Example: A DEPENDENT entity is identified by the dependent's first name and EMPLOYEE'S ssN Name of DEPENDENT is the partial key **DEPENDENT** is a weak entity type
4. EMPLOYEE is its identifying entity type via the identifying relationship type DEPENDENT_OF

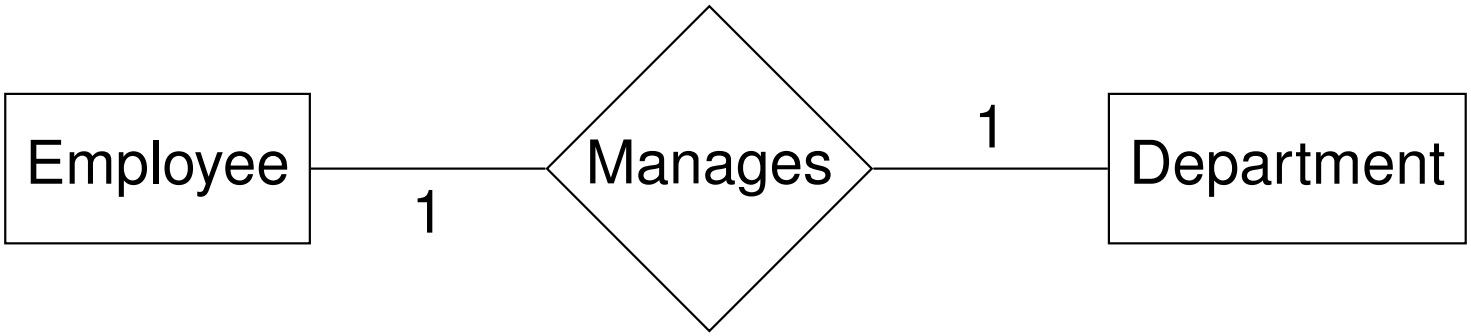
Constraints on Relationships

- Mapping Constraints
- Participation Constraints

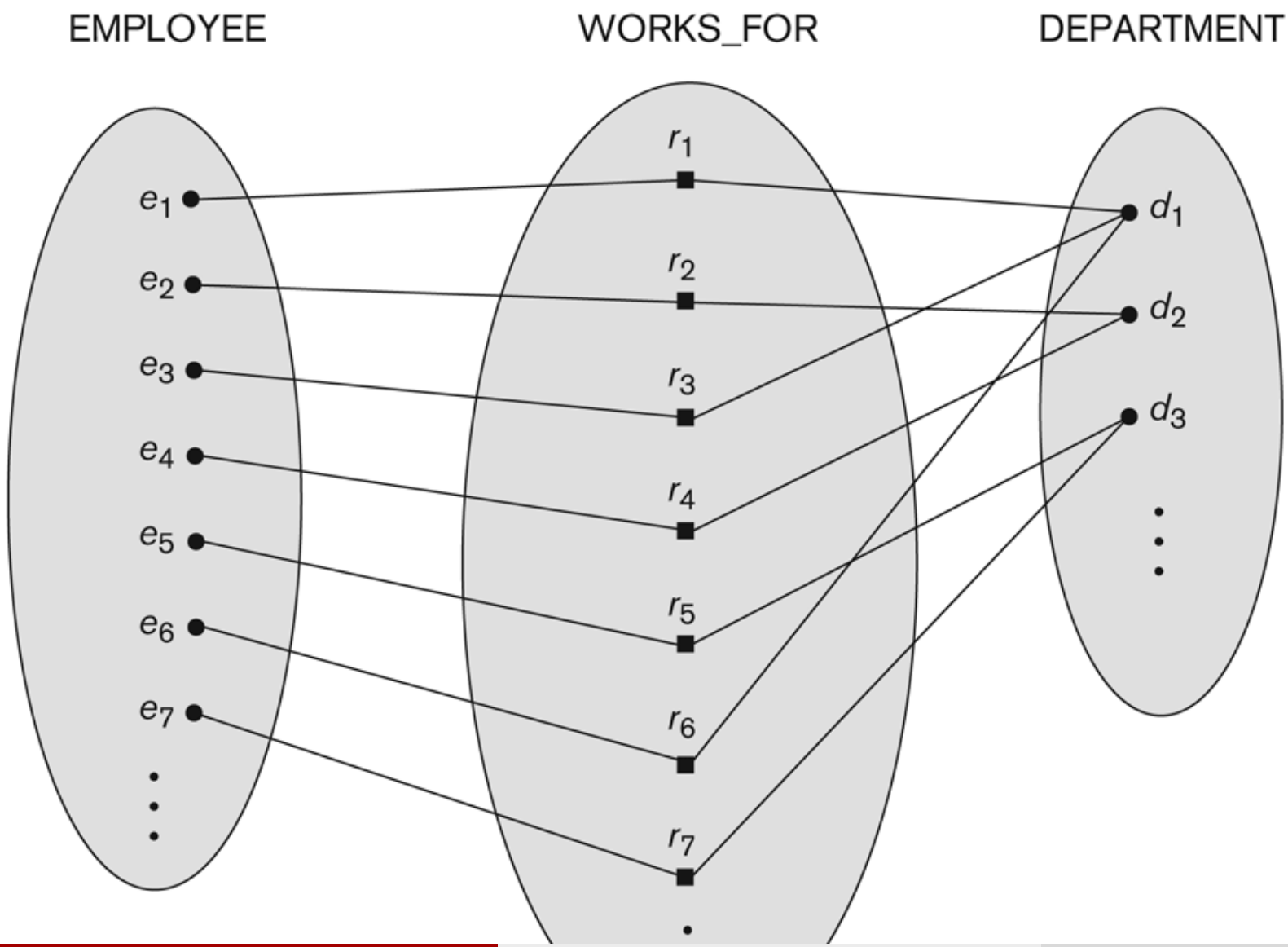
one to one

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

One to One Relationship

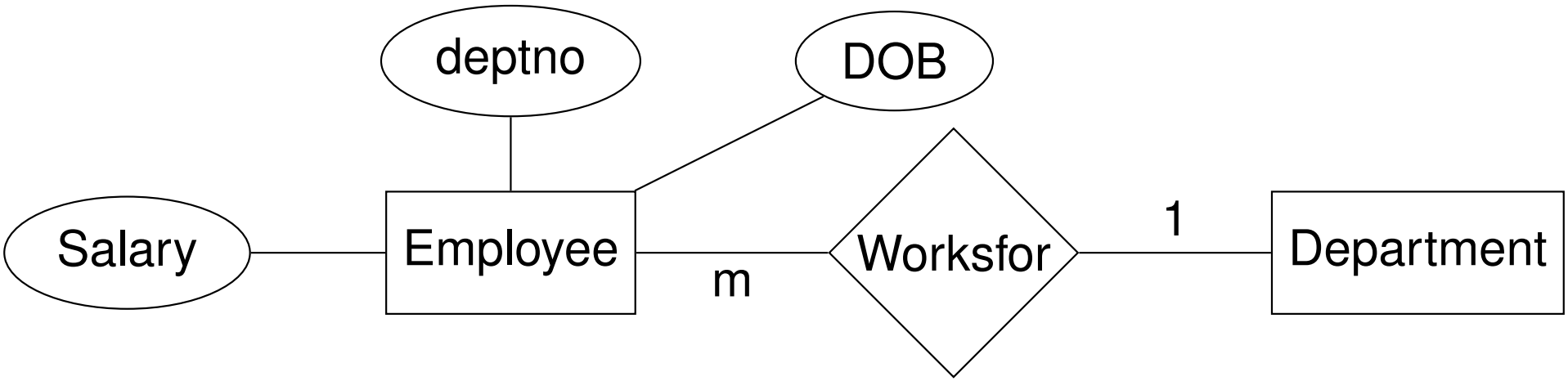


one to many



F

Some instan
WORKS_FOR re
set, which represe
tionship type WO


deptno
DOB
Salary
Employee
m
Worksfor
1
Department

Many to many

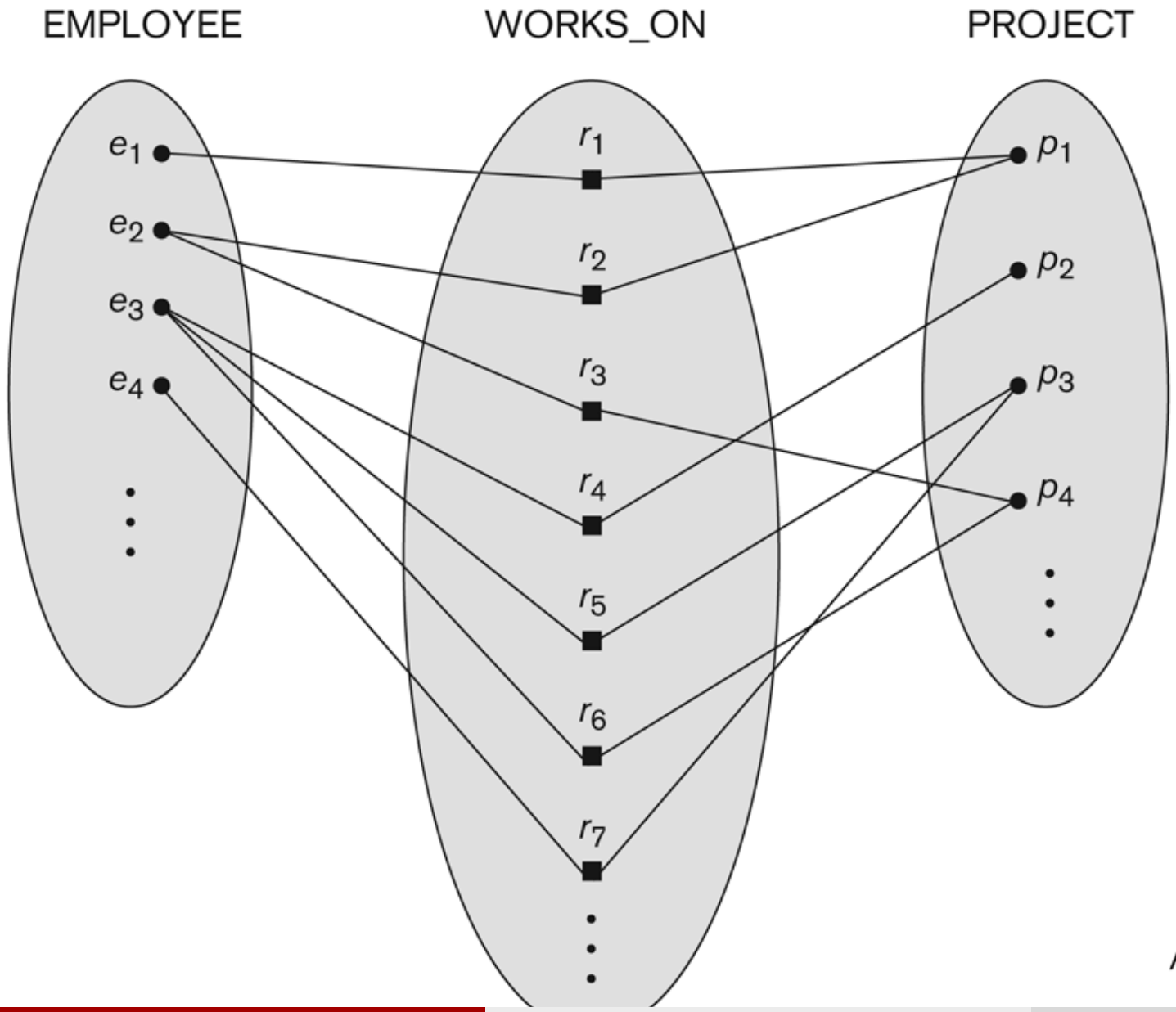
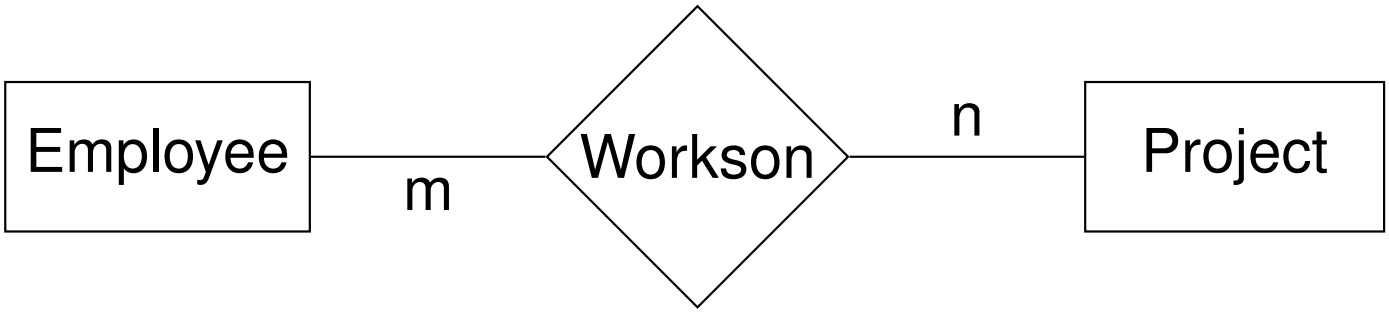
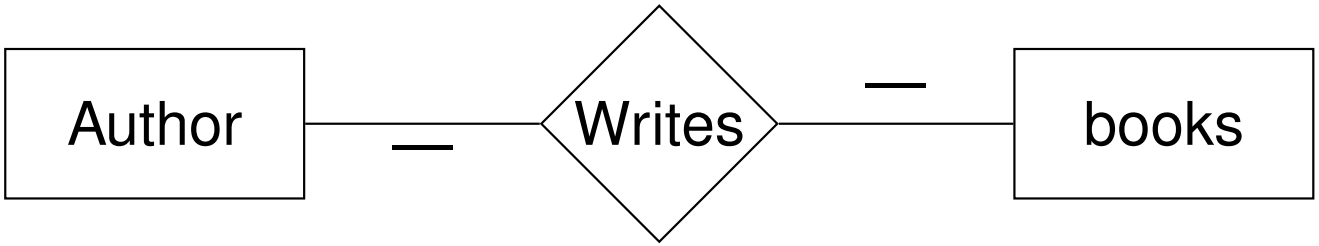


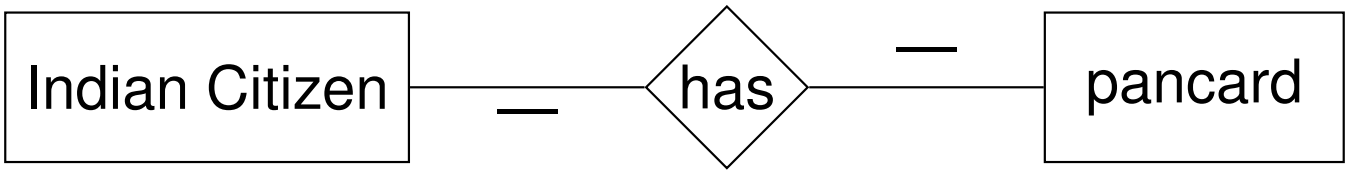
Figure 3.13
An M:N relationship, WORKS_ON


Employee
m
Workson
n
Project

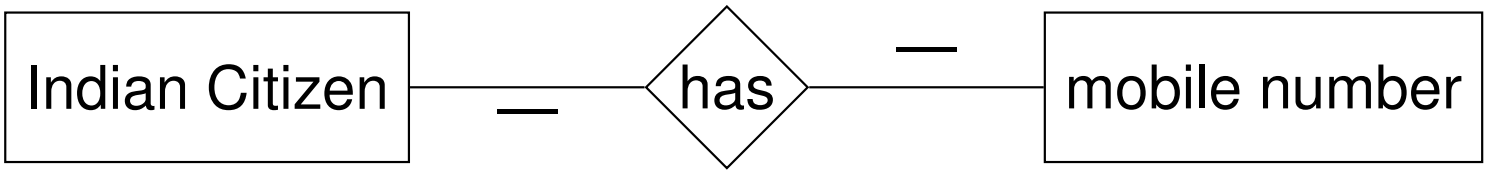
Exercise 1



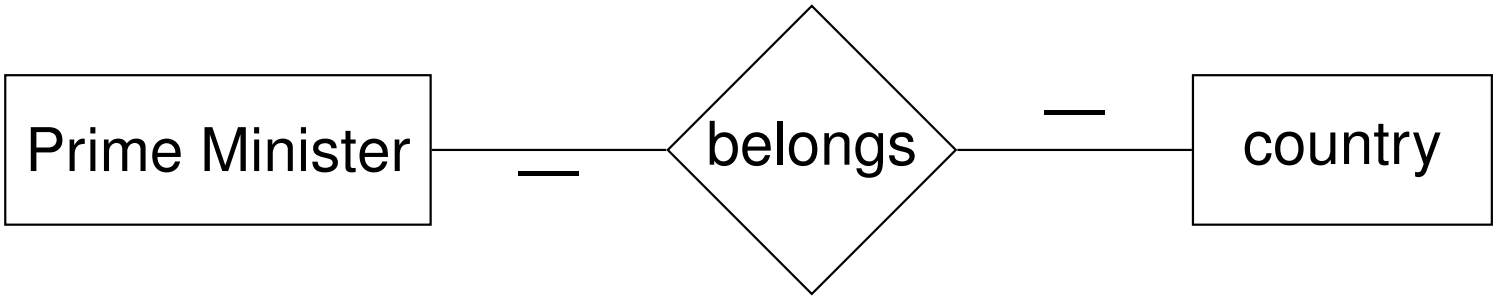
Exercise 2



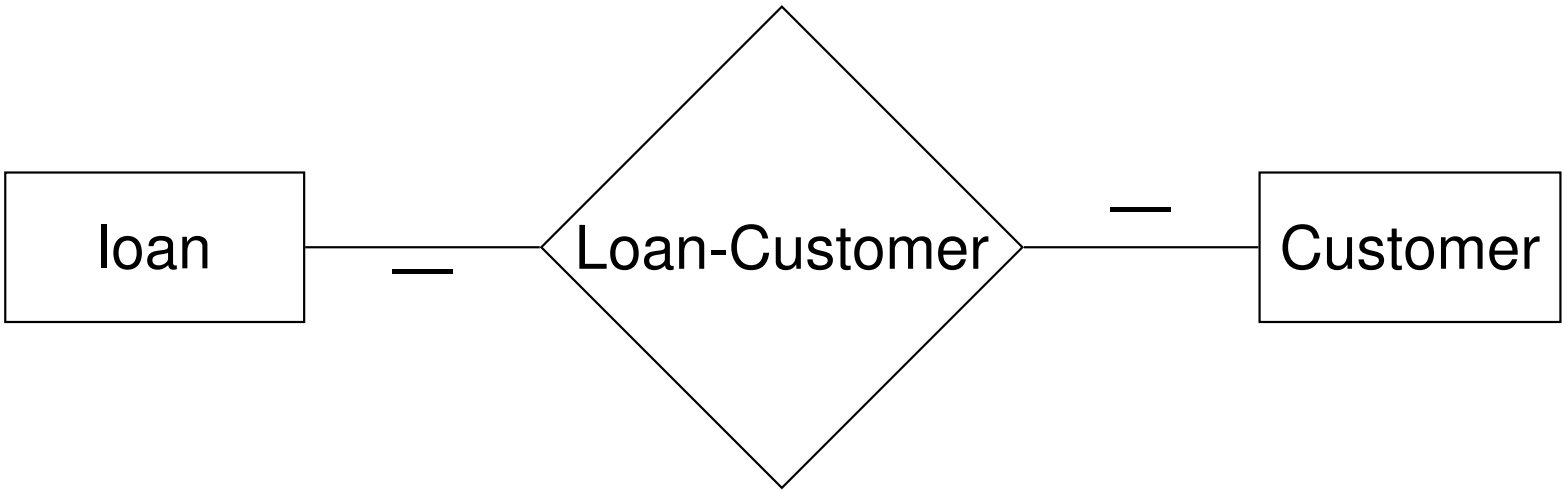
Exercise 3



Exercise 4



Exercise 5



Participation Constraints

1. If every entity in an entity set is participating in a relationship, then relationship is called **Total Participation**
2. If only some entities in an entity set is participating in a relationship, then relationship is called **Partial Participation**

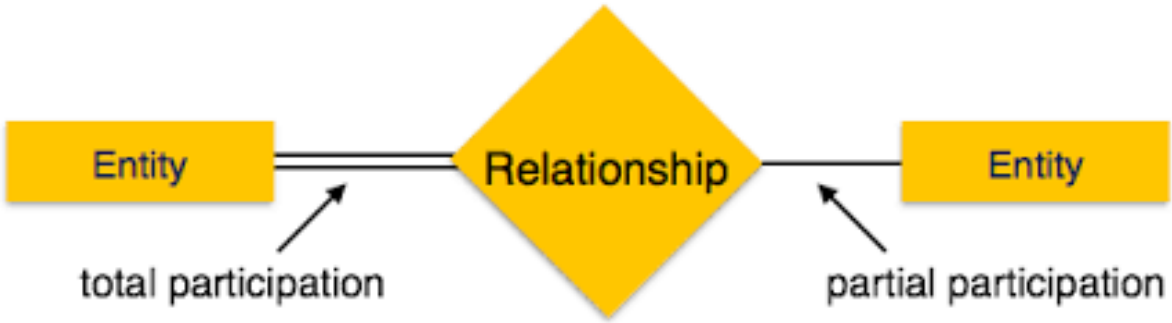


Figure: Relationshiptype

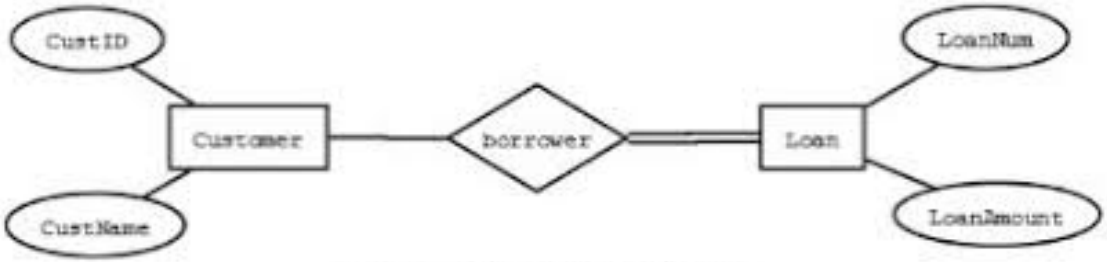


Figure: Total Participation

Figure: Total Participation

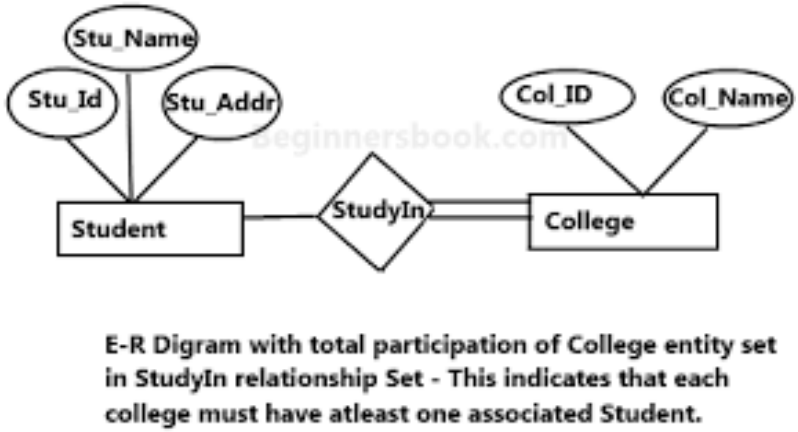


Figure: Total Participation

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.

Figure: Comparison between strong and weak entity