DBMS Module 1

ER Modeling

Relational model basics

- Data is viewed as existing in two dimensional tables known as relations
- A relation (table) consists of unique attributes (columns) and tuples (rows)
- Tuples are unique
- Sometimes the value to be inserted into a particular cell may be unknown, or it may have no value. This is represented by a **NULL**
- Null is not the same as zero, blank or an empty string
- Relational Database: Any database whose logical organization is based on relational data model.
- RDBMS: A DBMS that manages the relational database.

What is the technical term for ***row*** in DBMS?

A. Tuple
B. Attribute
C. Relation
D. Null

RollNo	Name	CPI
1	Amit	7.2
2	Ajay	8.1
3	Manoj	8.3
4	Vijay	7.9
:	:	:
:	:	:

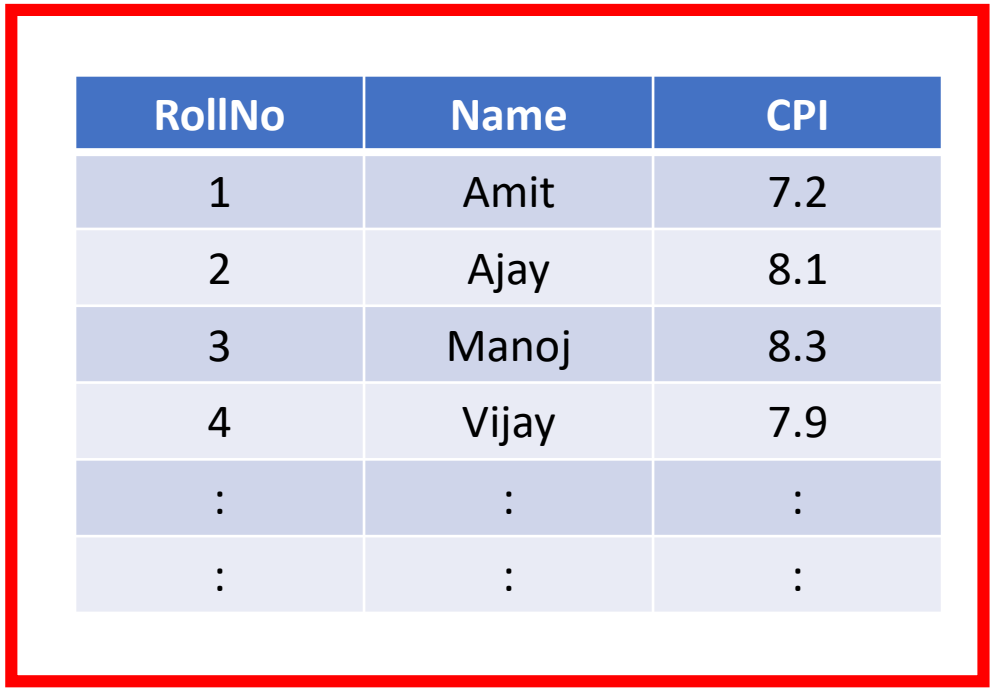
What is the technical term for ***column*** in DBMS?

A. Tuple
B. Attribute
C. Relation
D. Null

RollNo	Name	CPI
1	Amit	7.2
2	Ajay	8.1
3	Manoj	8.3
4	Vijay	7.9
:	:	:
:	:	:

What is the technical term for ***table*** in DBMS?

A. Tuple
B. Attribute
C. Relation
D. Null



RollNo	Name	CPI
1	Amit	7.2
2	Ajay	8.1
3	Manoj	8.3
4	Vijay	7.9
:	:	:
:	:	:

Sometime data is unknown or missing it is represented by ?

A. Tuple
B. Attribute
C. Relation
D. Null

RollNo	Name	CPI
1	Amit	7.2
2	Ajay	?
3	Manoj	?
4	Vijay	7.9
:	:	:
:	:	:

Keys in Relational Model

- Candidate key

  A Candidate key is a set of **one or more attributes(minimal)** that can uniquely identify a row in a given table.

- Primary Key

  During the creation of the table, the Database Designer chooses one of the Candidate Key from amongst the several available, to uniquely identify row in the given table.

ClassRollNo	UnivRollNo	aadhaarNo	Name	31-01-2022
1	201500001	626987564329	Amit	P
2	201500056	876945214872	Arun	A
3	201500132	762419735736	Binod	P

Keys in Relational Model

- Candidate key
- Primary Key
- Alternate Key

The candidate key that is chosen to perform the identification task is called the *primary key* and the remaining candidate keys are known as alternate keys.

No of Alternate Keys = No of Candidate Keys - 1

- Super Key

Any superset of a candidate Key is a super key.

ClassRollNo	UnivRollNo	aadhaarNo	Name	31-01-2022
1	201500001	626987564329	Amit	P
2	201500056	876945214872	Arun	A
3	201500132	762419735736	Binod	P

Which of the following are **candidate keys**?

LicensePlateNumber	ChasisNo	ModelName	ManufacturerName	No_of_seats

A. LicensePlateNumber
B. ChasisNo
C. ModelName
D. ManufacturerName
E. No_of_seats

Which of the following you will choose as **Primary key**?

LicensePlateNumber	ChasisNo	ModelName	ManufacturerName	No_of_seats

A. LicensePlateNumber
B. ChasisNo
C. ModelName
D. ManufacturerName
E. No_of_seats

Which of the following will be ***alternate key*** if **LicensePlateNumber** is your **Primary key**?

LicensePlateNumber	ChasisNo	ModelName	ManufacturerName	No_of_seats

A. LicensePlateNumber
B. ChasisNo
C. ModelName
D. ManufacturerName
E. No_of_seats

Which of the following are ***super key***?

LicensePlateNumber	ChasisNo	ModelName	ManufacturerName	No_of_seats

A. LicensePlateNumber
B. ChasisNo
C. ChasisNo, ModelName
D. ManufacturerName
E. No_of_seats, LicensePlateNumber
F. ManufacturerName, No_of_seats

Key and Non-key Attributes in Relational Model

- **Key Attributes**
  - The attributes that participate in the Candidate key are Key Attributes

- **Non-Key Attributes**
  - The attributes other than the Candidate Key attributes in a table/relation are called Non-Key attributes.

OR

  - The attributes which do not participate in the Candidate key.

Example

Given a relation

Trainee(Empno, FirstName, LastName, Email, PhoneNo)

Candidate key:

{Empno},{Email},{PhoneNo}

Primary key:

{Empno}

Alternate Key:

{Email},{PhoneNo}

Super Key:

{Empno},{Empno,PhoneNo},{Email,FirstName}, etc…

Each candidate key is a super key. But vice versa is not true

Exercise on Key attributes

Given a relation ***R1(X,Y,Z,L)*** and the following attribute(s) can uniquely identify the records of relation R1.

1)X

2)X,L

3)Z,L

Identify the following in relation R1?

Candidate Key(s) :	**{X}, {Z,L}**
Primary Key:	**ANYONE OF {X}, {Z,L}**
Alternate Key	**{Z,L} *if* {X} *is primary key***
Key attribute(s)	**X, Z, L**
Non-key attribute(s)	**Y**

Which of the following are **key attributes**?

LicensePlateNumber	ChasisNo	ModelName	ManufacturerName	No_of_seats

A. LicensePlateNumber
B. ChasisNo
C. ModelName
D. ManufacturerName
E. No_of_seats

Which of the following are **non-key attributes**?

LicensePlateNumber	ChasisNo	ModelName	ManufacturerName	No_of_seats

A. LicensePlateNumber
B. ChasisNo
C. ModelName
D. ManufacturerName
E. No_of_seats

Foreign Key

- **Foreign key**
  - A Foreign Key is a set of attribute (s) whose values are required to match values of a column in the same or another table.

DEPT

(Parent /Master/Referenced Table)

DeptNo	DName
D1	IVS
D2	ENR

EMP

(Child /Referencing Table)

EmpNo	EName	EDeptNo
1001	Elsa	D1
1002	John	D2
1003	Maria	Null
1004	Maida	D1

- Point to remember
  - Foreign key values do not (usually) have to be unique.
  - Foreign keys can also be *null* .

Foreign Key

- Foreign key

- Points to remember
  - A Foreign Key is a set of attributes of a table, whose values are required to match values of some Candidate Key in the same or another table
  - The constraint that values of a given Foreign Key must match the values of the corresponding Candidate Key is known as Referential constraint
  - A table which has a Foreign Key referring to its own Candidate Key is known as Self-Referencing table

A foreign key is:

- **A.** a column containing the primary key of another table.
- **B.** used to define data types.
- **C.** used to define null status.
- **D.** all of the above are above correct.

Can following scenario is possible??

- **Adding a row with D3 ??**

- Yes
- No

DEPT

(Parent /Master/Referenced Table)

DeptNo	DName
D1	IVS
D2	ENR

EMP

(Child /Referencing Table)

EmpNo	EName	EDeptNo
1001	Elsa	D1
1002	John	D2
1003	Maria	Null
1004	Maida	D1
1005	Imran	D3

Can following scenario is possible??

- **Adding a row with NULL ??**

- Yes
- No

DEPT

(Parent /Master/Referenced Table)

DeptNo	DName
D1	IVS
D2	ENR

EMP

(Child /Referencing Table)

EmpNo	EName	EDeptNo
1001	Elsa	D1
1002	John	D2
1003	Maria	Null
1004	Maida	D1
1005	Imran	Null

- Consider attributes ID, CITY and NAME. Which one of this can be considered as a super key?
  a) NAME
  b) ID
  c) CITY
  d) CITY, ID
- The subset of a super key is a candidate key under what condition?
  a) No proper subset is a super key
  b) All subsets are super keys
  c) Subset is a super key
  d) Each subset is a super key

- Consider attributes ID, CITY and NAME. Which one of this can be considered as a super key?
  a) NAME
  b) ID
  c) CITY
  d) CITY, ID
- The subset of a super key is a candidate key under what condition?
  a) No proper subset is a super key
  b) All subsets are super keys
  c) Subset is a super key
  d) Each subset is a super key

Database Design Techniques

Top down Approach

Top down starts by defining the data sets and then define the data elements within those sets. As a result of this method, you generally end up with redundant information in one or more tables.

Some references call this **Entity - Relationship modeling**.

Bottom Up approach

Bottom up starts by defining the required attributes and then grouping them to form the entities. Another term used for this method is **normalization** from functional dependencies.

ER Modeling
Top down Approach

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.** *E.g.:* **Trainee**
- **Relationships:** *Associations between entities. E.g.: Trainee belongs to a Batch*
- **Attributes:** *Properties/characteristics that describe entities.eg: Trainee Name, BatchName, DOB, Address, etc.*

Association between the instances of one or more entity types

Person, place, object, event or concept about which data is to be maintained

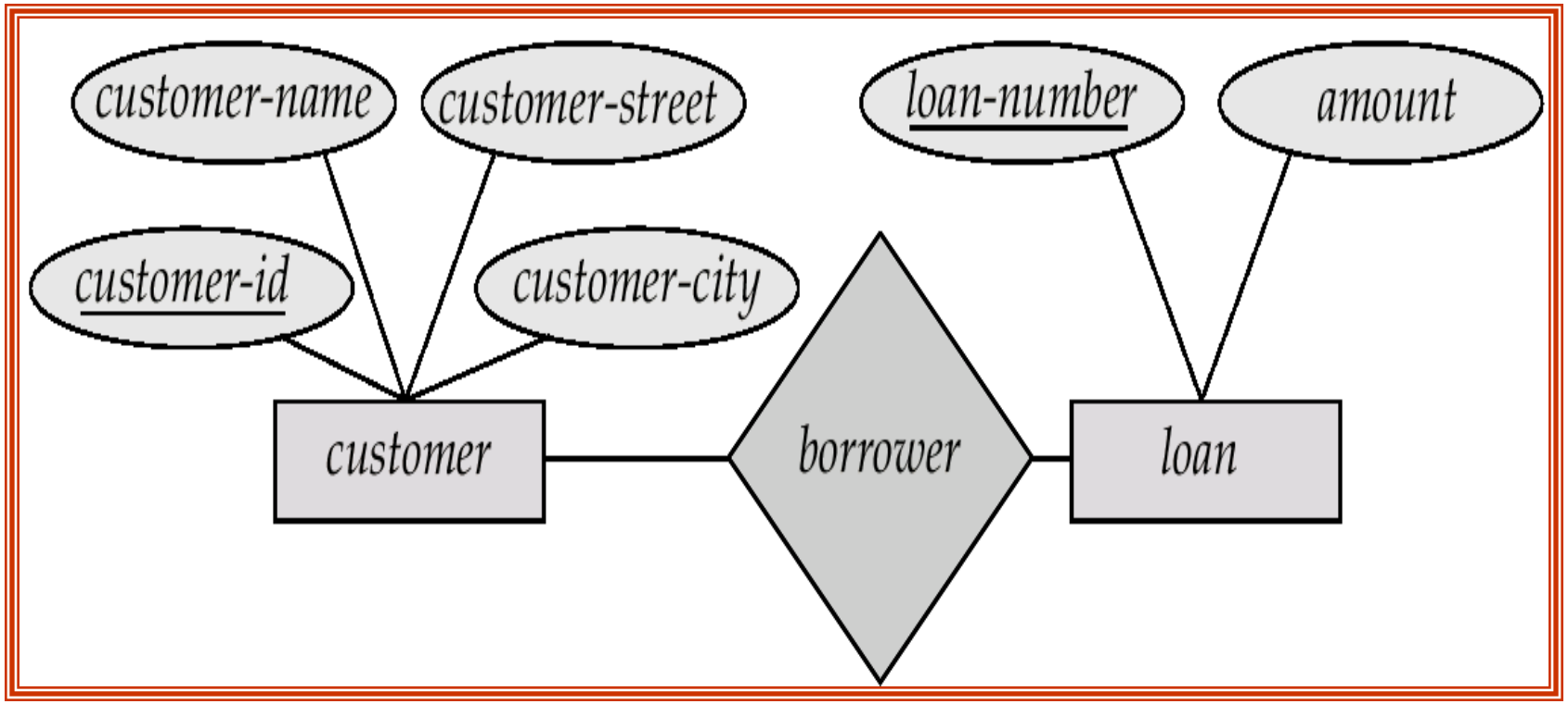
named property or characteristic of an entity

Represents a set or collection of objects in the real world that share the same properties

E-R modeling technique is a :

A. Top-down approach
B. Bottom-up approach
C. Left-right approach
D. None of the above

An Example


customer-name
customer-street
loan-number
amount
customer-id
customer-city
customer
borrower
loan

Entities

- Examples of entities:
  - Person: EMPLOYEE, STUDENT, PATIENT
  - Place: STORE, WAREHOUSE
  - Object: MACHINE, PRODUCT, CAR
  - Event: SALE,REGISTRATION, RENEWAL
  - Concept: ACCOUNT, COURSE

- Guidelines for naming and defining entity types:
  - An entity type name is a singular noun
  - An entity type should be descriptive and specific
  - An entity name should be concise
  - Event entity types should be named for the result of the event, not the activity or process of the event.

Attributes

- Example of entity types and associated attributes:

  STUDENT: Student_ID, Student_Name, Home_Address, Phone_Number, Major

- Guidelines for naming attributes:
  - An attribute name is a noun.
  - An attribute name should be unique
  - To make an attribute name unique and clear, each attribute name should follow a standard format
  - Similar attributes of different entity types should use similar but distinguishing names.
- Attribute Types:
  - Simple and composite attributes
  - Single-valued and multi-valued attributes
  - Derived attribute

Entity Types

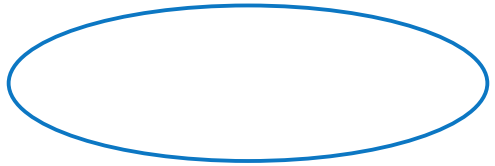
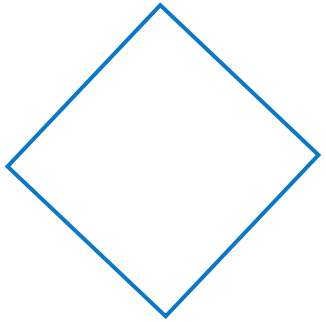
- **Regular Entity** : Entity that has its own key attribute (s). Aka Strong entity

  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 (s) of its own

  E.g. : spouse of employee

Which of the following used to denote entity?

- A 
- B 
- C 
- D 

An Entity is a_____

A. Relationship model

B. Relational Model

C. Object having same Value

D. Object in the real world distinguishable from all other objects

An entity is represented by set of _____

A. Attributes

B. Relationship

C. Model

D. None of the above

Entity can be which of the following:

A. Person
B. Place
C. Object
D. All of the above

Attributes

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

  Example:
  - The domain of attribute ***marital status*** is having four values: single, married, divorced or widowed.
  - The domain of the attribute month is having 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.

What is domain of Mobile No?

A. Numbers starting from 6, 7 , 8 and 9

B. Numbers starting from 6, 7 , 8 and 9 , & 10-digit long

C. Numbers starting from 9 & whole number

D. Numbers starting from 6, 7 , 8 and 9 & having only 10 digits

Attribute Type

Types of Attributes	Definition	Example
Simple attribute	Cannot be divided into simpler components	**Gender** of the employee
Composite attribute	Can be split into components	**Date of joining** of the employee
Single valued	Can take on only a single value for each entity instance	**Age** of the employee
Multi-valued	Can take up many values	**Skill set** of the employee
Stored Attribute	Attribute that need to be stored permanently	**Date of joining** of the employee
Derived Attribute	Attribute that can be calculated based on other attributes.	**Years of service** of the employee

An Attribute takes a _____value when an entity does not have a value for it.

A. Zero

B. 0

C. Null

D. Not Applicable

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 Relationship sets that involve two entity sets in known as___

A. Unary Relationship set

B. Binary Relationship set

C. Ternary Relationship set

D. None of the above

Cardinality

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

E.g.:

Employee **head-of** department (1:1)

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

Student **enrolls** course (M:N)

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

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

All employees will not be head-of some department. So only few instances of employee entity participate in the above relationship. But each department will be headed by some employee.
So department entity's participation is total and employee entity's participation is partial in the above relationship.

In a Entity sets Customer and Loan, an Attribute name consisting of first-name ,middle-name, and last-name. An Attribute name is _____

A. Simple Attribute

B. Composite Attribute

C. Single-Valued Attributes

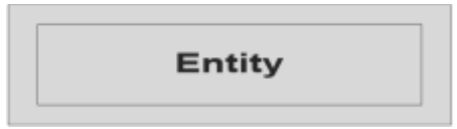
D. Multi-valued Attributes

ER Modeling - Notations

Education and Research

ER Modeling -Notations

An Entity is an object or concept about which business user wants to store information.



A weak Entity is dependent on another Entity to exist. Example Order Item depends upon Order Number for its existence. Without Order Number it is impossible to identify Order Item uniquely.

Attributes are the properties or characteristics of an Entity

A key attribute is the unique, distinguishing characteristic of the Entity

A multi-valued attribute can have more than one value. For example, an employee Entity can have multiple skill values.

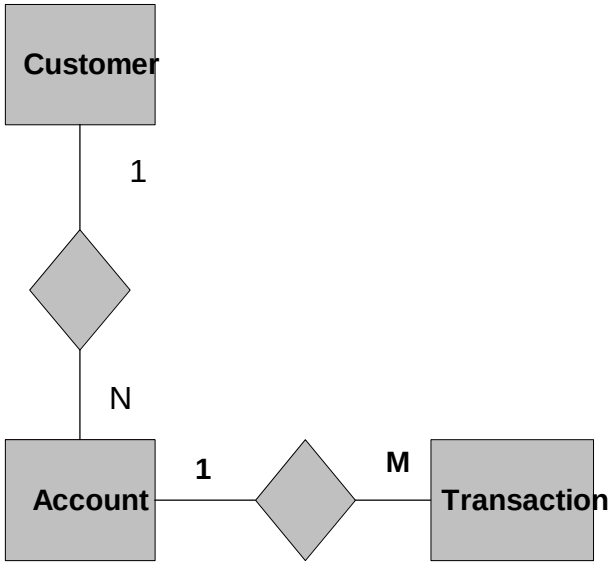
ER Modeling -Notations

A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's basic salary and House rent allowance.

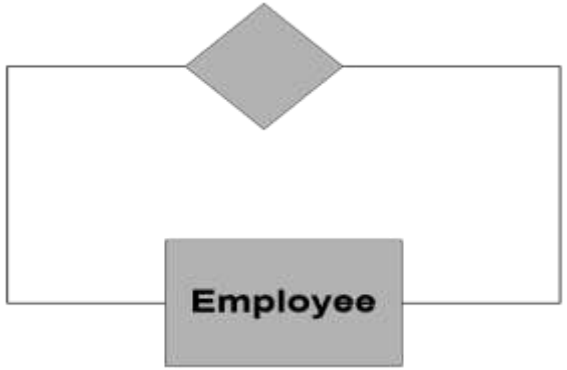
Relationships illustrate how two entities share information in the database structure.

To connect a weak Entity with others, you should use a weak relationship notation.

ER Modeling -Notations

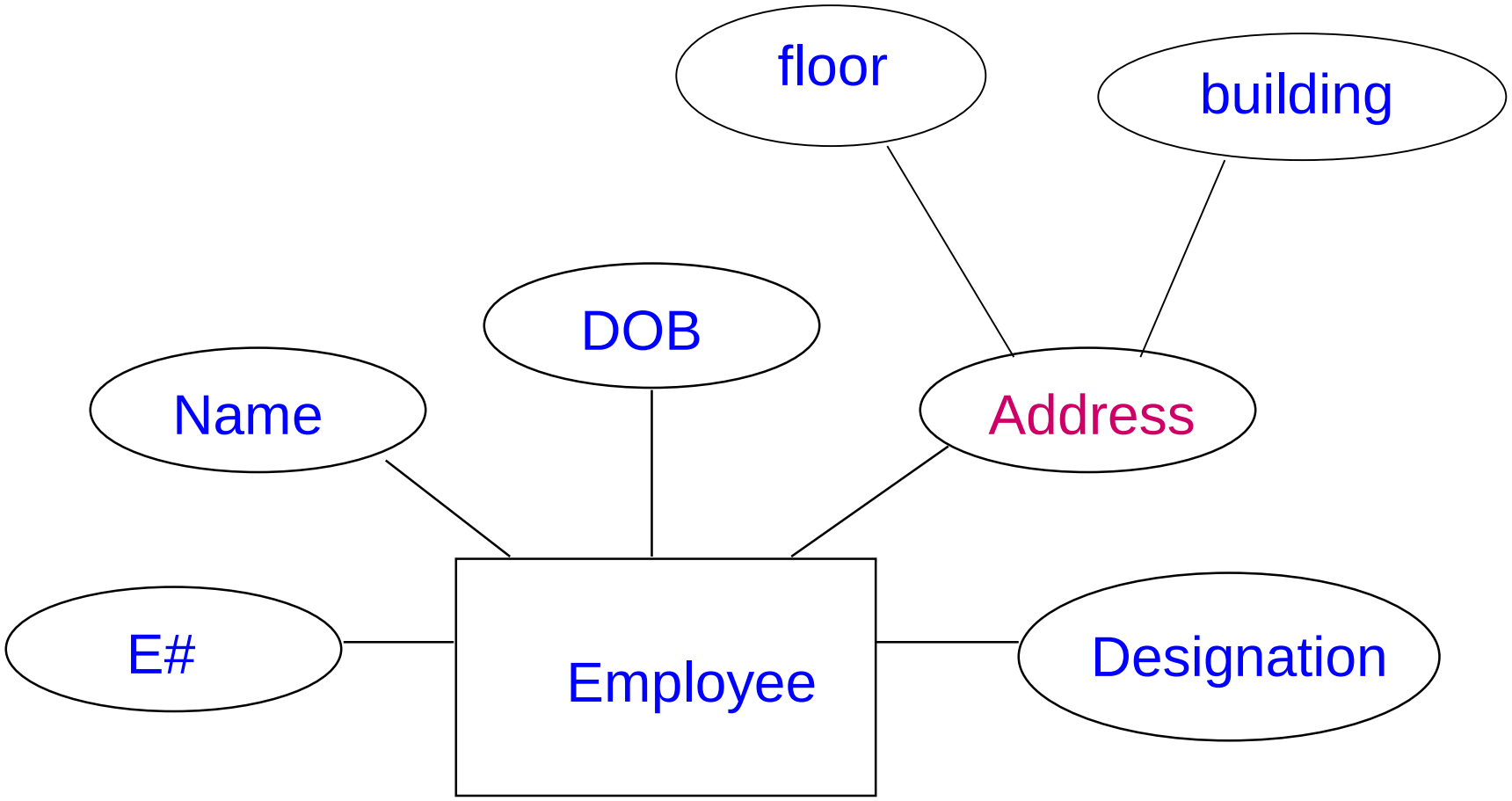


Cardinality specifies how many instances of an Entity relate to one instance of another Entity. M,N both represent ‘MANY’ and 1 represents ‘ONE’ Cardinality



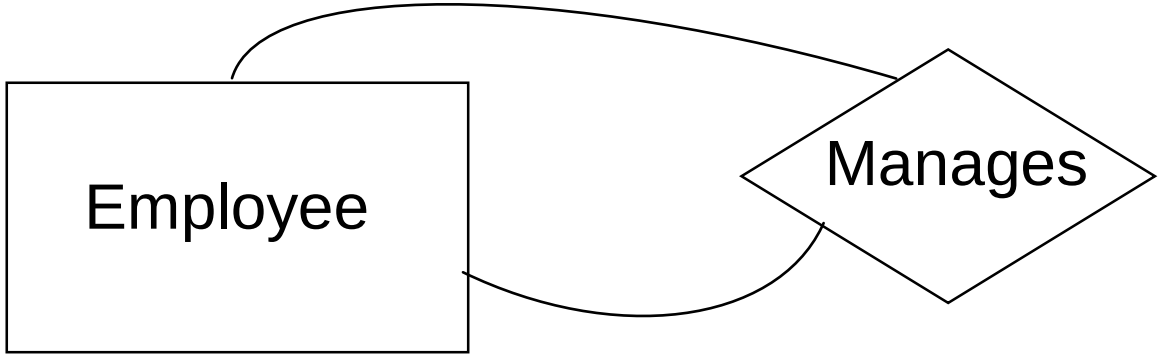
In some cases, entities can be self-linked. For example, employees can supervise other employees

Composite attribute



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

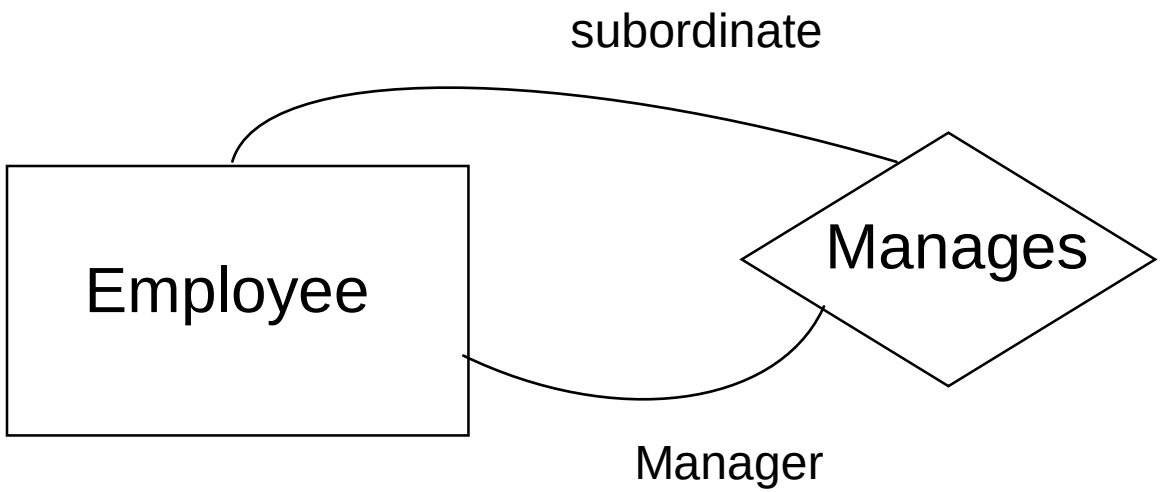
Unary Relationship



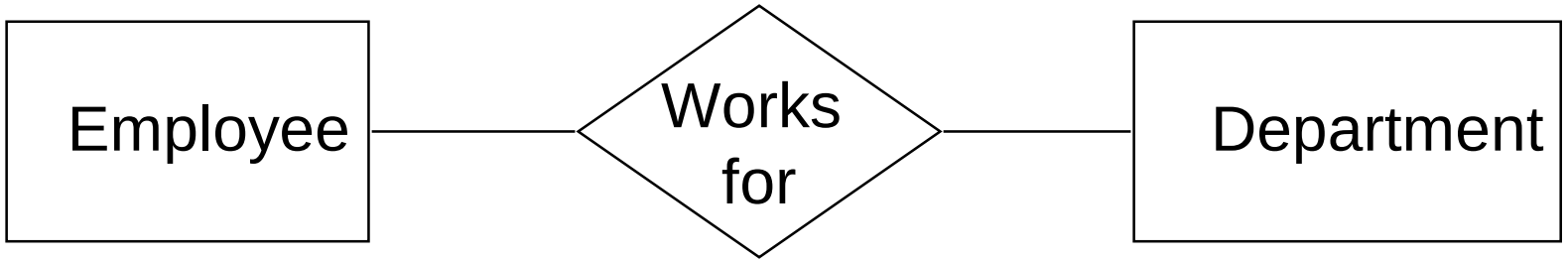
- A unary relationship is represented as a diamond which connects one entity to itself as a loop.
- The relationship above means, some instances of employee manage other instances of Employee.

Role names

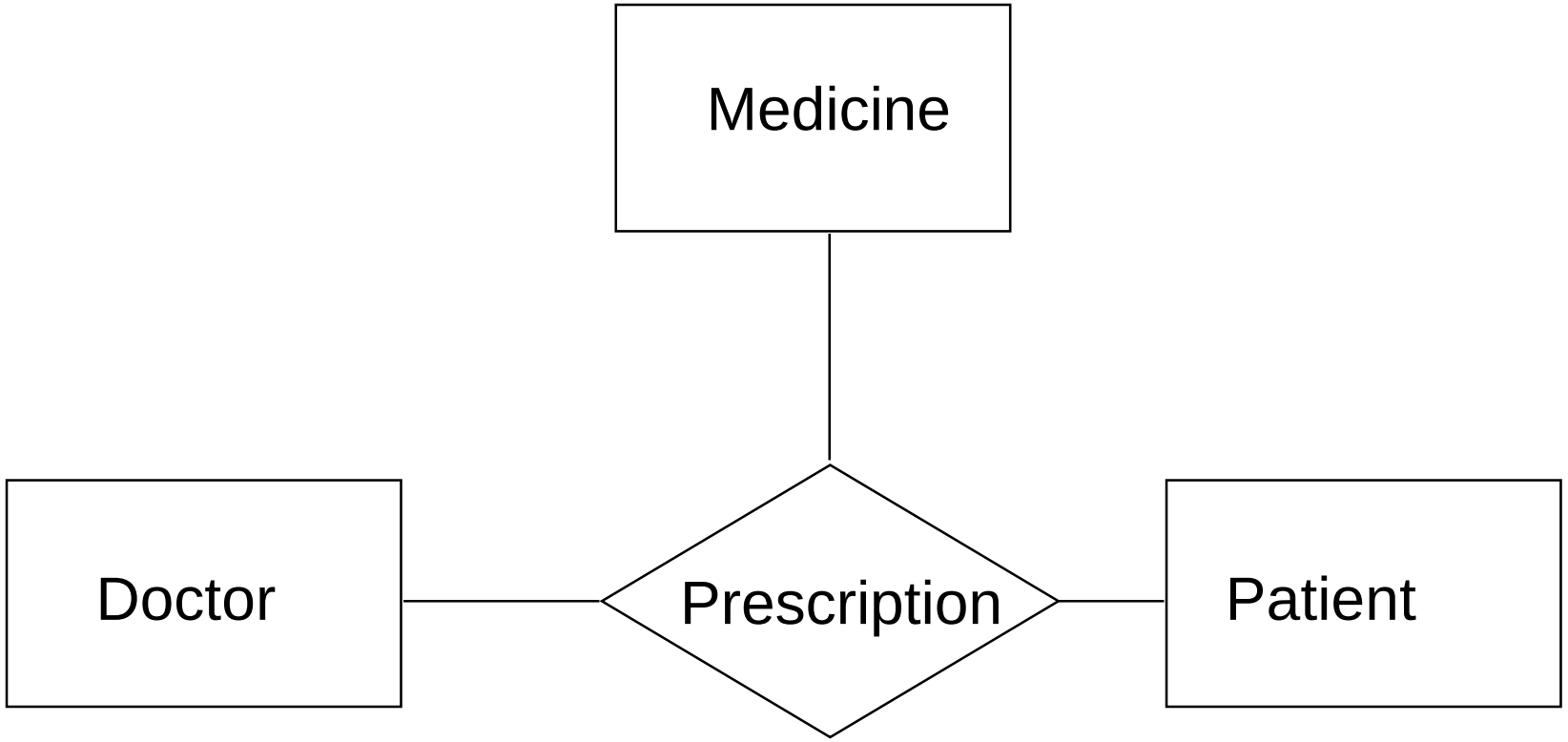
- Role names may be added to make the meaning more explicit



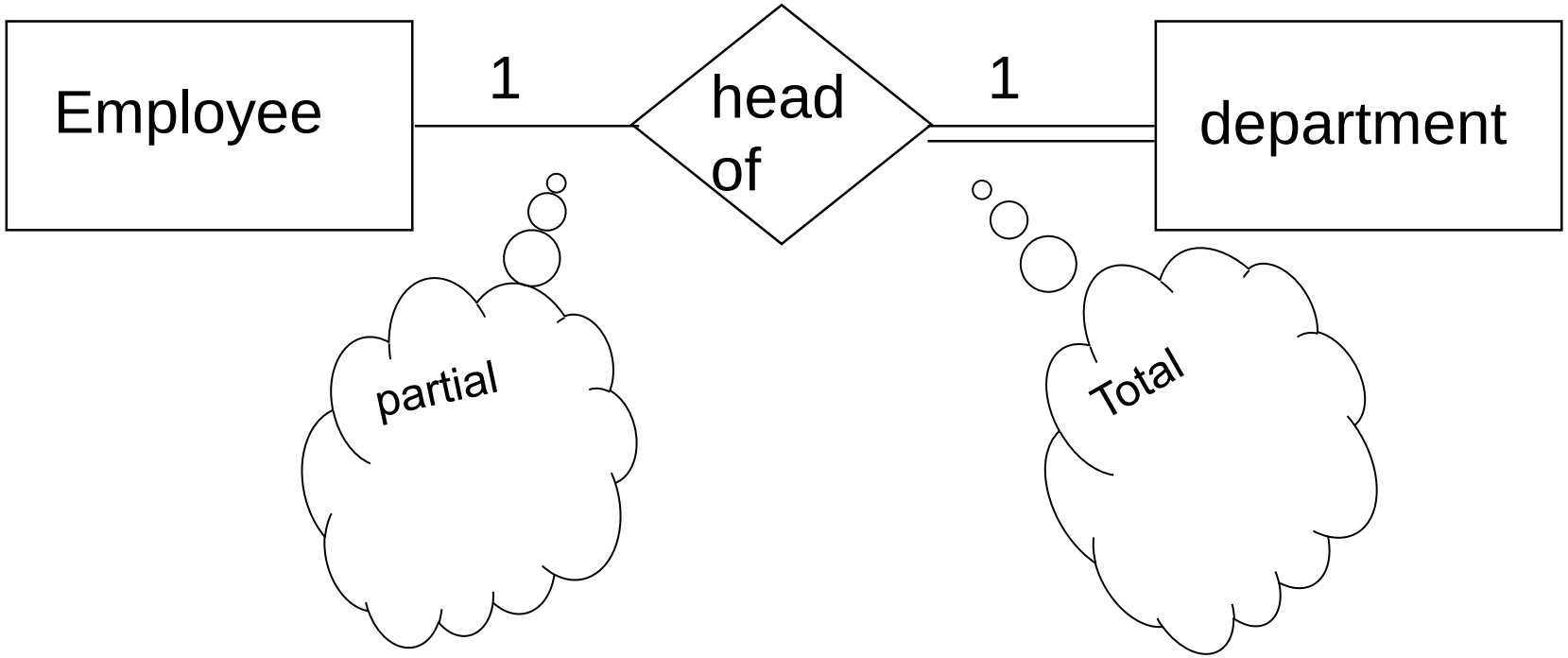
Binary Relationship



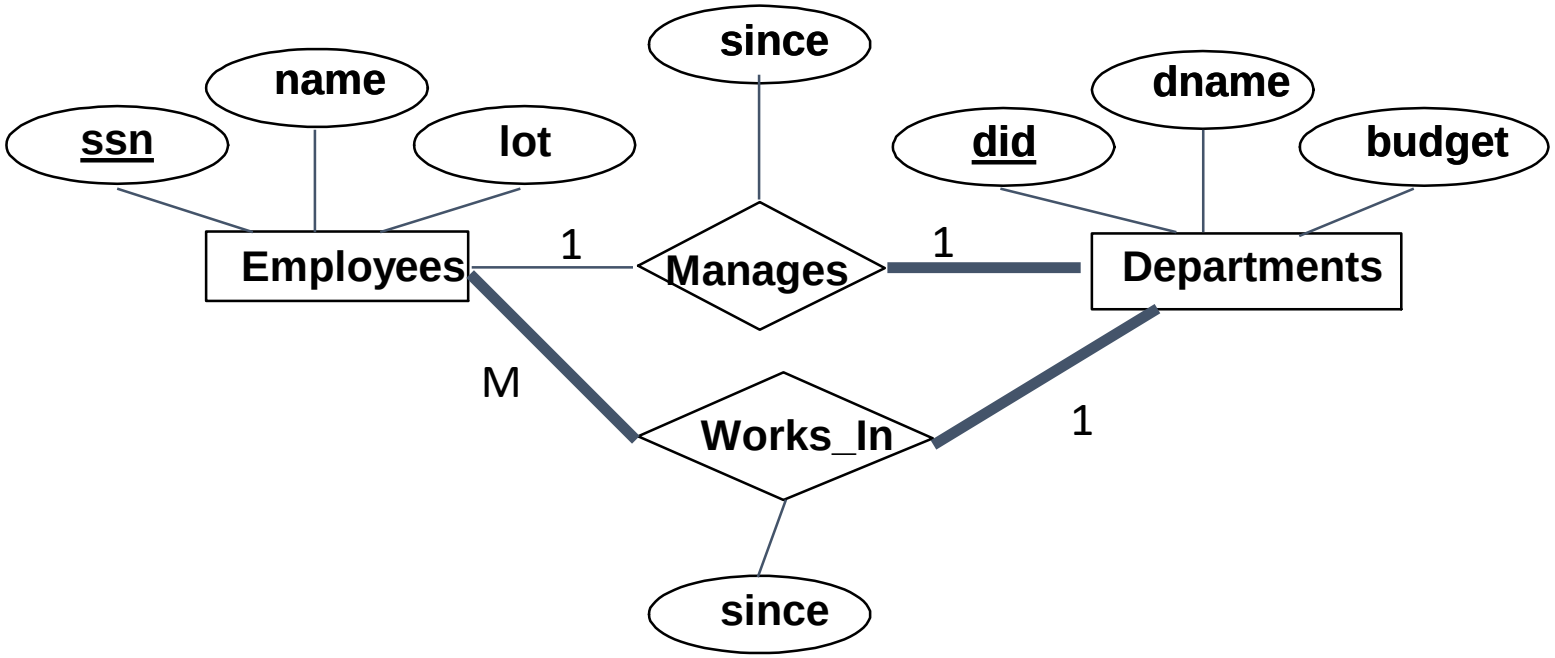
Ternary Relationship



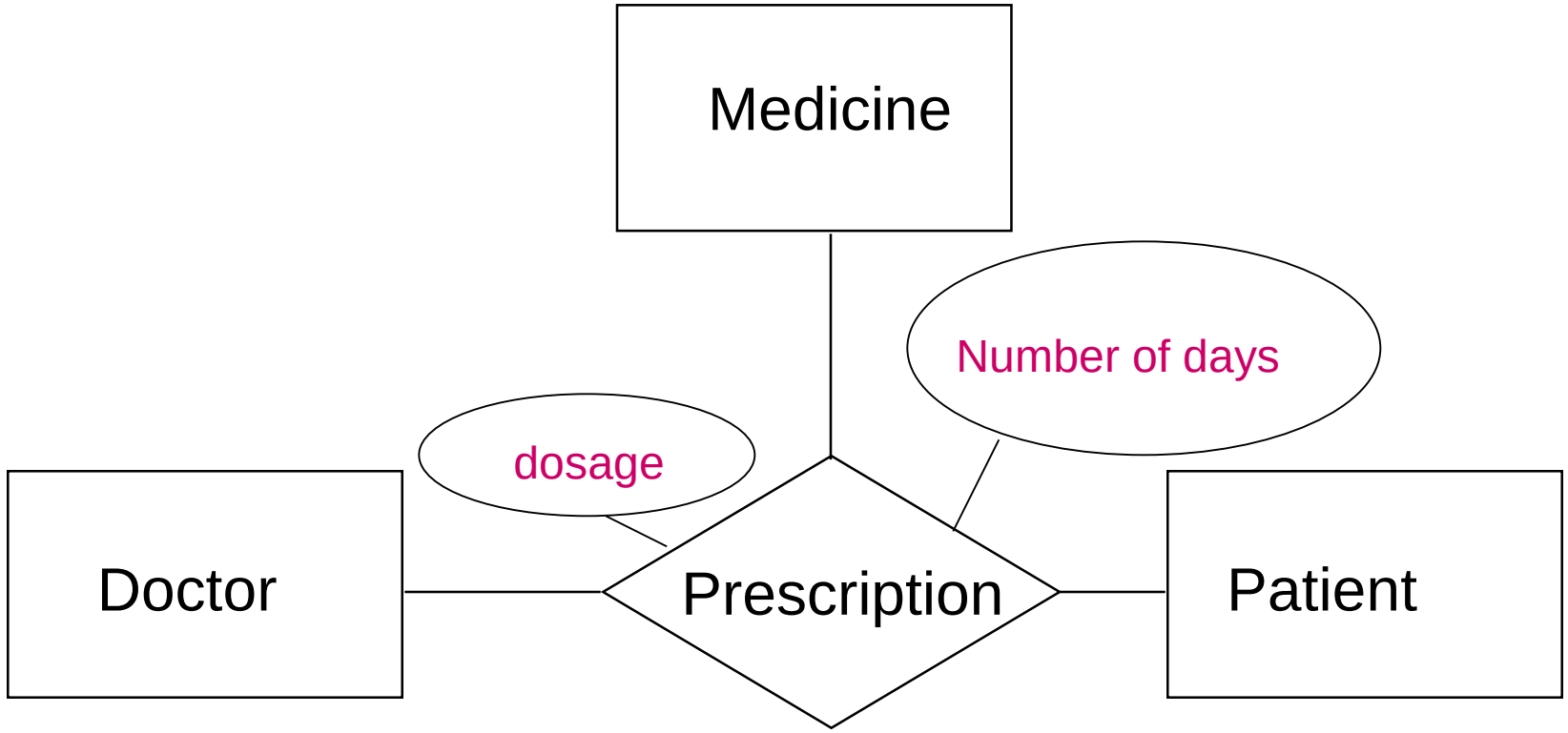
Relationship participation



- 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.

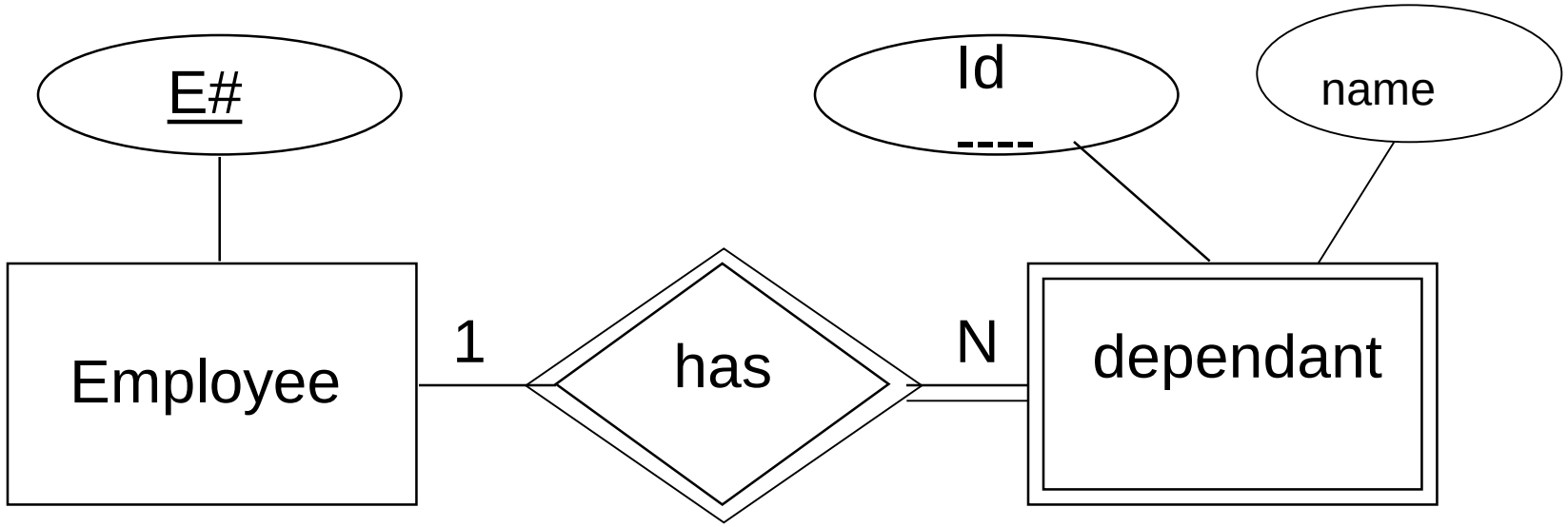

since
name
dname
ssn
lot
did
budget
Employees
1
Manages
1
Departments
M
Works_In
1
since

Attributes of a Relationship



These attributes best describe the relationship prescription rather than any individual entity Doctor, Patient or Medicine.

Weak entity



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

Case Study – ER Model For a college DB

Assumptions :

- A college contains many departments
- Each department can offer any number of courses
- Many instructors can work in a department
- An instructor can work only in one department
- For each department there is a Head
- An instructor can be head of only one department
- Each instructor can take any number of courses
- A course can be taken by only one instructor
- A student can enroll for any number of courses
- Each course can have any number of students

Steps in ER Modeling

- Identify the Entities
- Find relationships
- Identify the key attributes for every Entity
- Identify other relevant attributes
- Draw complete E-R diagram with all attributes including Primary Key
- Review your results with your Business users

Visit the following link for more information

10 Easy Steps to Create an ER Diagram in VISIO 2000

Step 1: Identify the Entities

- DEPARTMENT
- STUDENT
- COURSE
- INSTRUCTOR

Steps in ER Modeling

Step 2: Find the relationships

- One course is enrolled by multiple students and one student enrolls for multiple courses, hence the cardinality between course and student is Many to Many.
- The department offers many courses and each course belongs to only one department, hence the cardinality between department and course is One to Many.
- One department has multiple instructors and one instructor belongs to one and only one department , hence the cardinality between department and instructor is one to Many.
- Each department there is a "Head of department" and one instructor is "Head of department ",hence the cardinality is one to one .
- One course is taught by only one instructor, but the instructor teaches many courses, hence the cardinality between course and instructor is many to one.

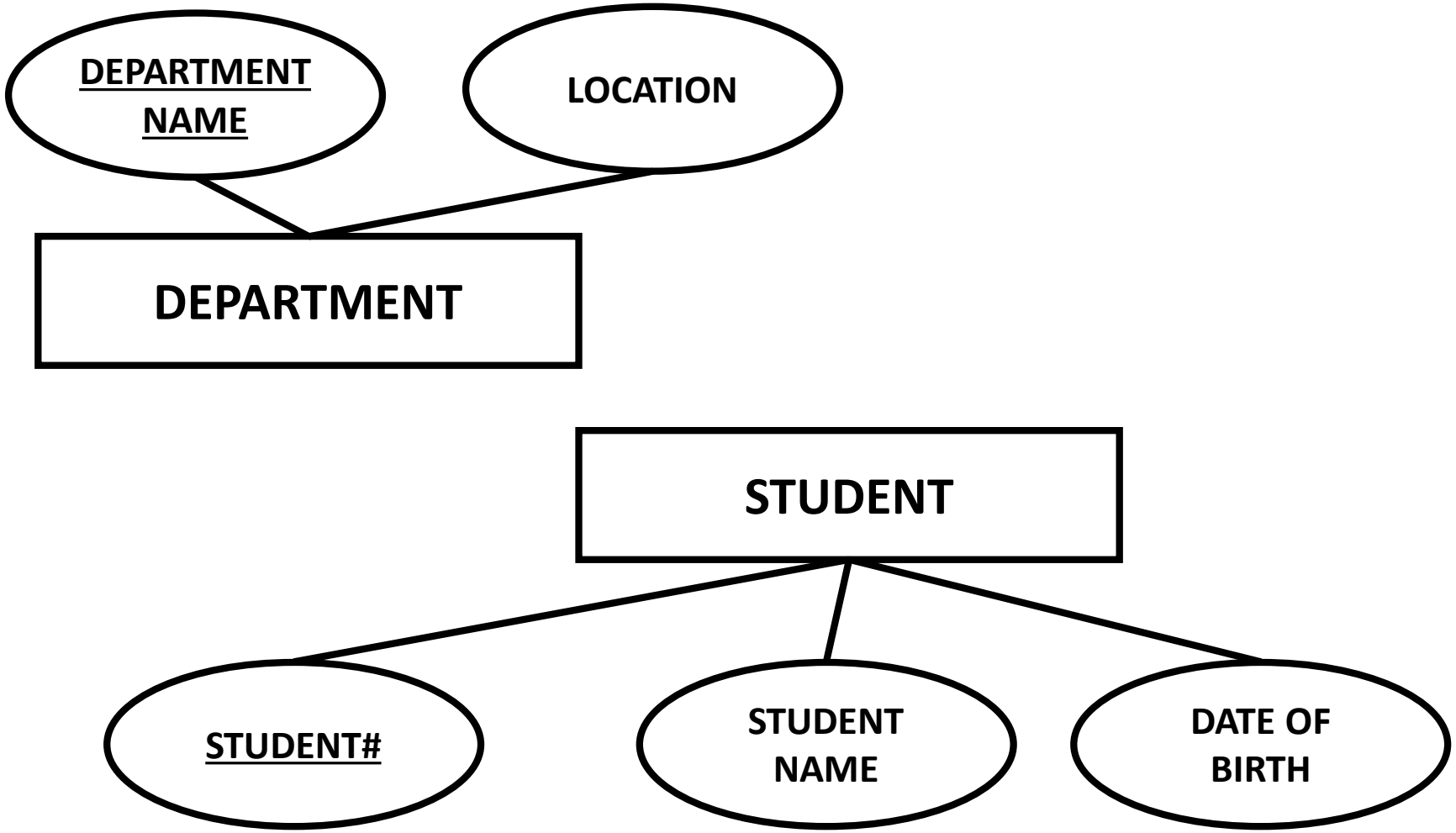
Step 3: Identify the key attributes

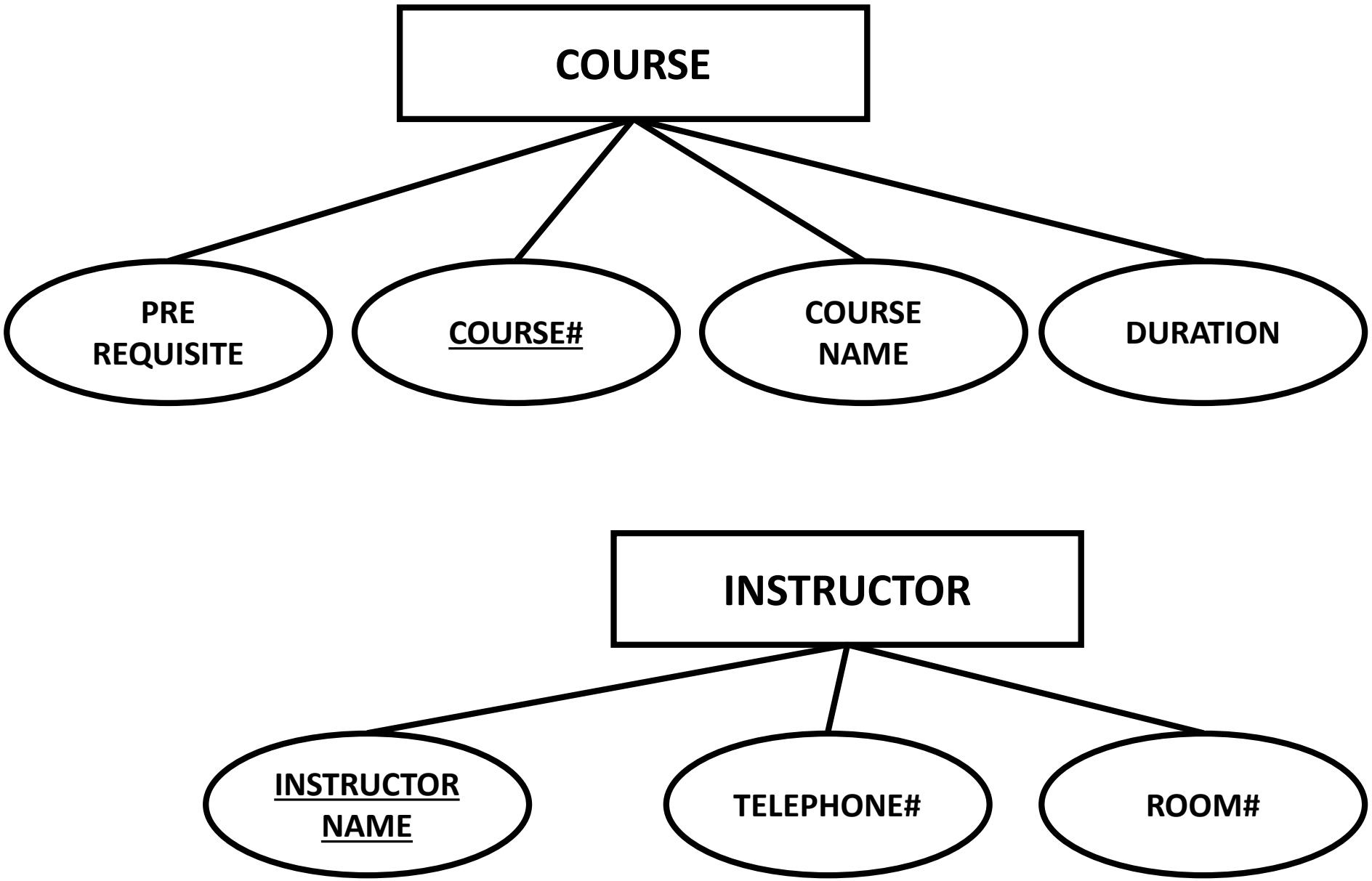
- Deptname is the key attribute for the Entity "Department", as it identifies the Department uniquely.
- Course# (CourseId) is the key attribute for "Course" Entity.
- Student# (Student Number) is the key attribute for "Student" Entity.
- Instructor Name is the key attribute for "Instructor" Entity.

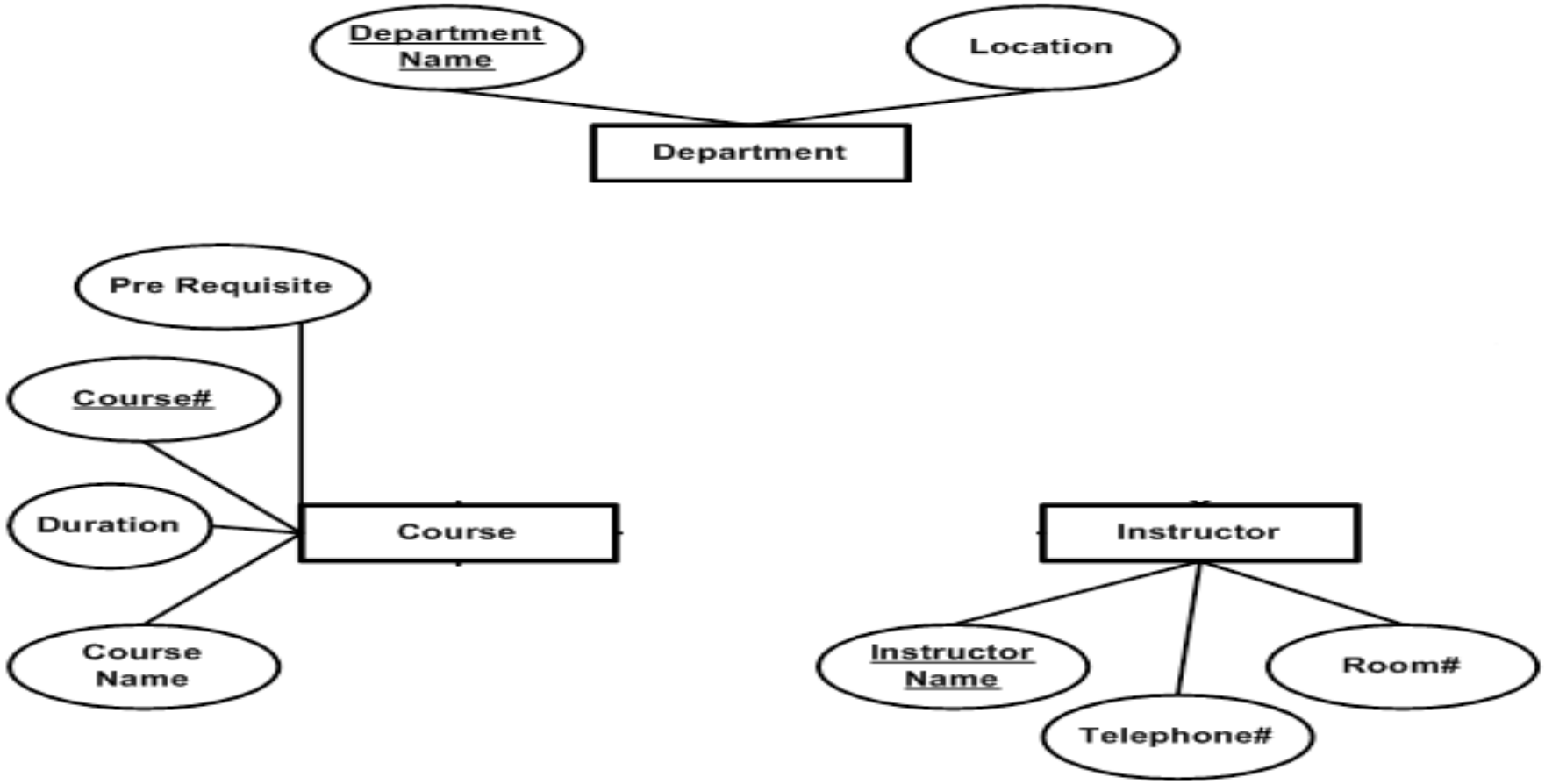
Step 4: Identify other relevant attributes

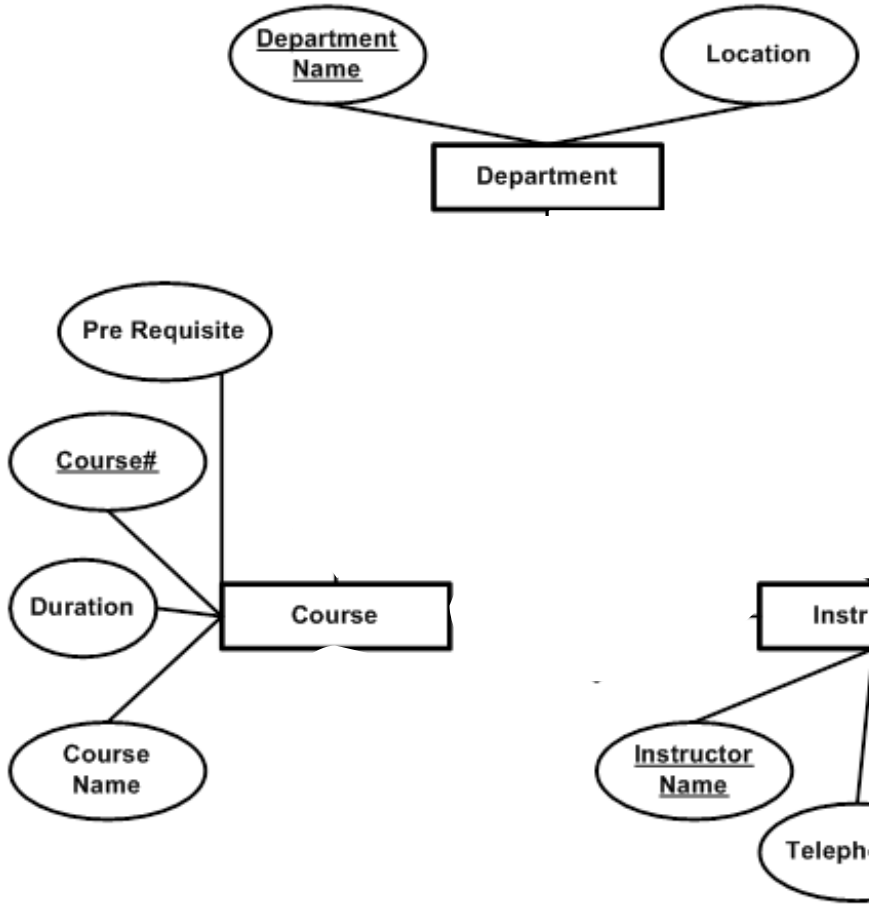
- For the department entity, the relevant attribute is location
- For course entity, course name,duration,prerequisite
- For instructor entity, room#, telephone#
- For student entity, student name, date of birth

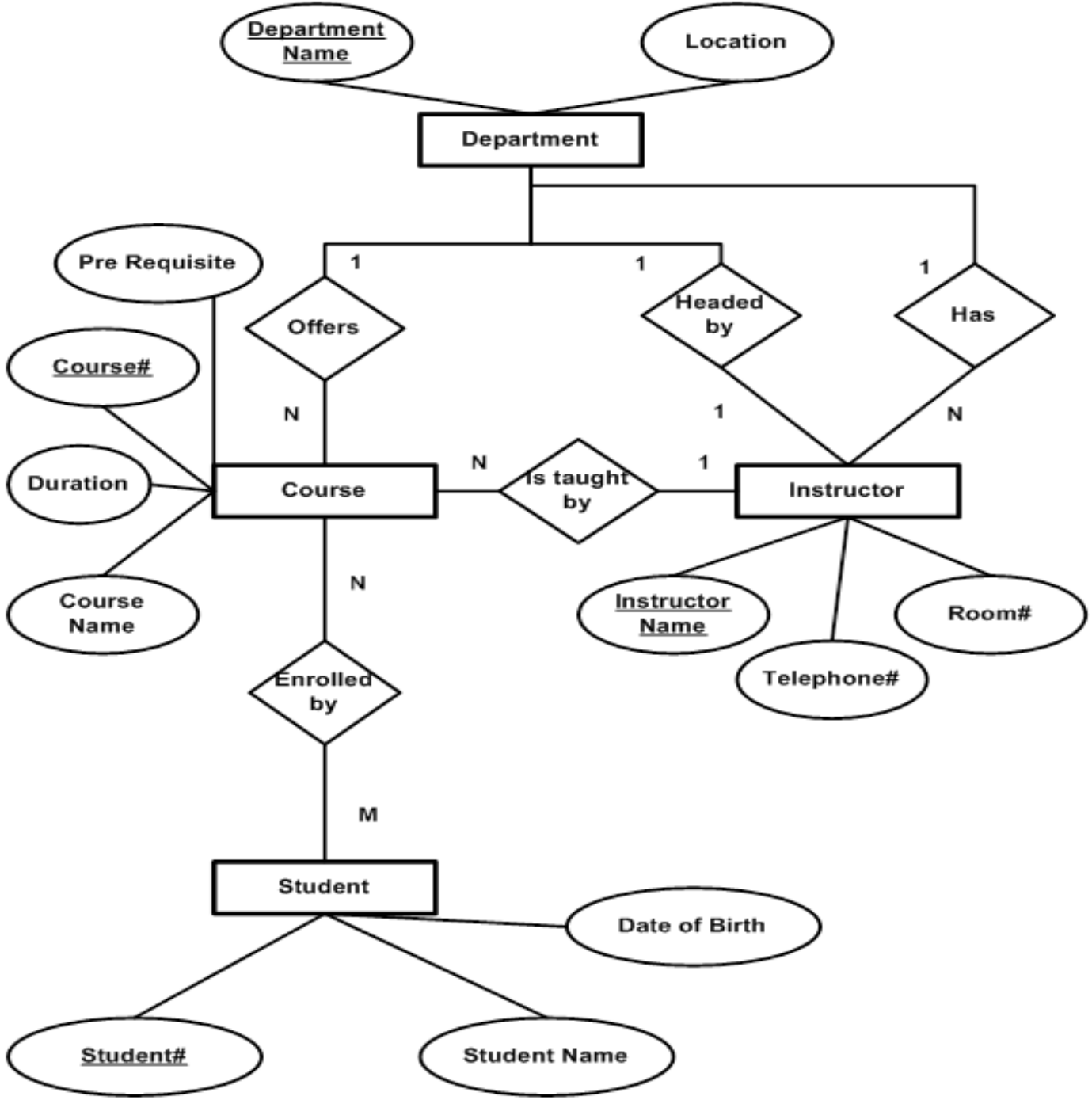
Entities


DEPARTMENT NAME
LOCATION
DEPARTMENT
STUDENT
STUDENT#
STUDENT NAME
DATE OF BIRTH


COURSE
PRE REQUISITE
COURSE#
COURSE NAME
DURATION
INSTRUCTOR
INSTRUCTOR NAME
TELEPHONE#
ROOM#


Department Name
Location
Department
Pre Requisite
Course#
Duration
Course
Course Name
Instructor
Instructor Name
Room#
Telephone#
Student
Date of Birth
Student#
Student Name


Department Name
Location
Department
Pre Requisite
Course#
Duration
Course
Course Name
Instructor
Instructor Name
Room#
Telephone#
Student
Date of Birth
Student#
Student Name

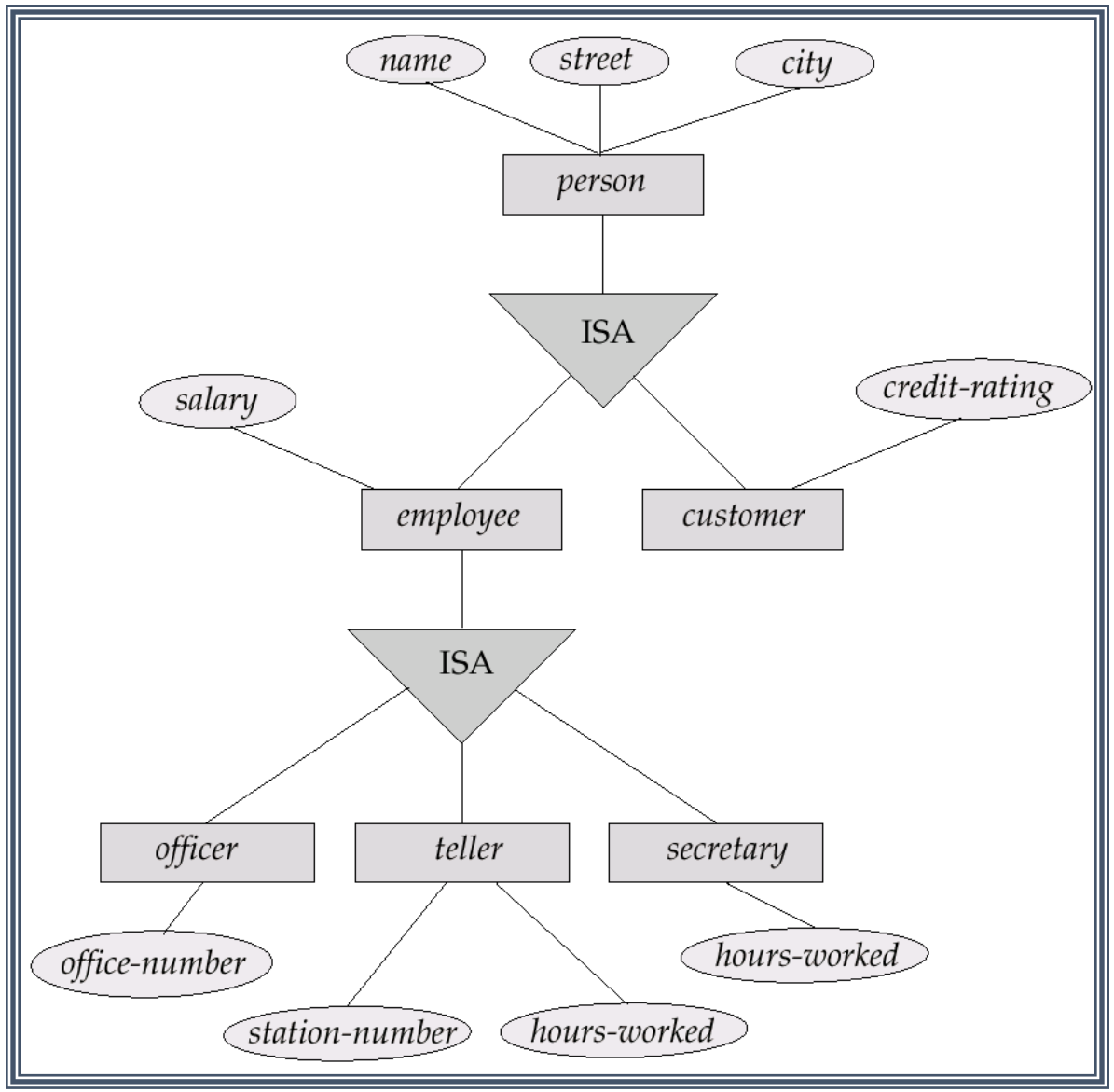

Department Name
Location
Department
Pre Requisite
1
1
1
Offers
Headed by
Has
Course#
N
1
N
Duration
N
1
Is taught by
Course
Instructor
Course Name
N
Instructor Name
Room#
Telephone#
Enrolled by
M
Student
Date of Birth
Student#
Student Name

Case Study

The Alphabet Company needs a database to track employee information. When an employee is hired they are assigned to a department. Each employee is assigned an employee id and a manager. The HR department also needs to track the employee's name, date of birth and hire date. Department information, such as department code, name, and budget code should also be tracked.

Specialization

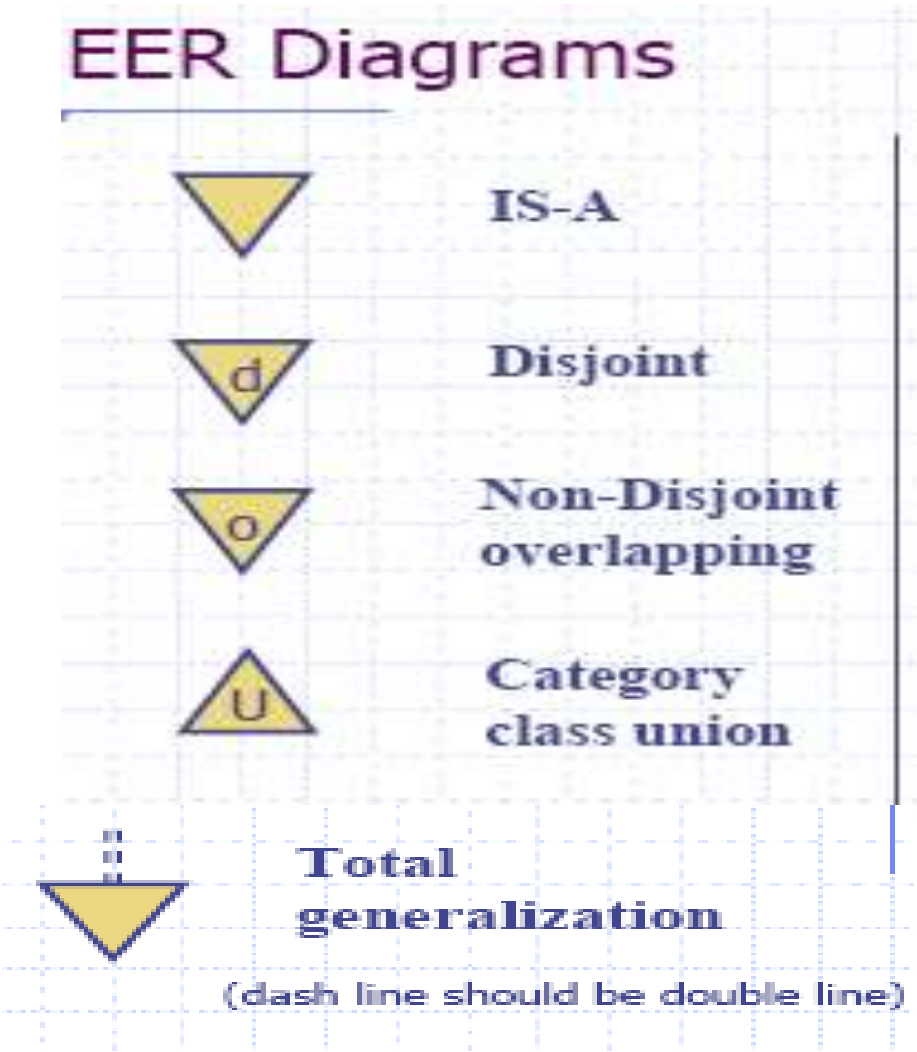
- A lower-level entity set inherits all the attributes and relationship participation of the higher-level entity set to which it is linked.
- A lower-level entity set may have additional attributes and participate in additional relationships


name
street
city
person
ISA
salary
credit-rating
employee
customer
ISA
officer
teller
secretary
office-number
station-number
hours-worked
hours-worked

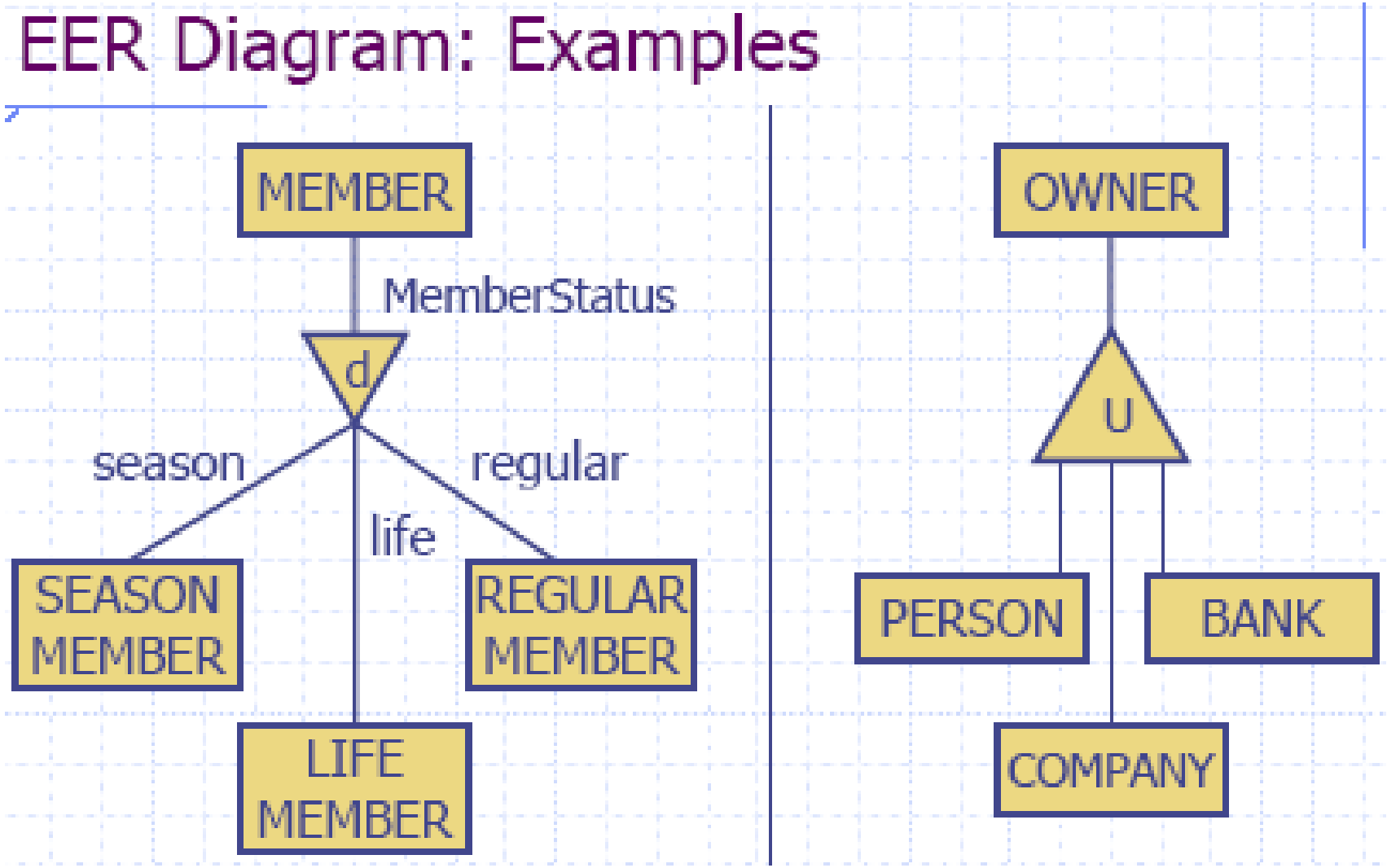
Specification

- Disjoint/Overlapping
- Completeness constraint

  (use double lines)
  - **total** : an entity must belong to one of the lower-level entity sets
  - **partial**: an entity need not belong to one of the lower-level entity sets



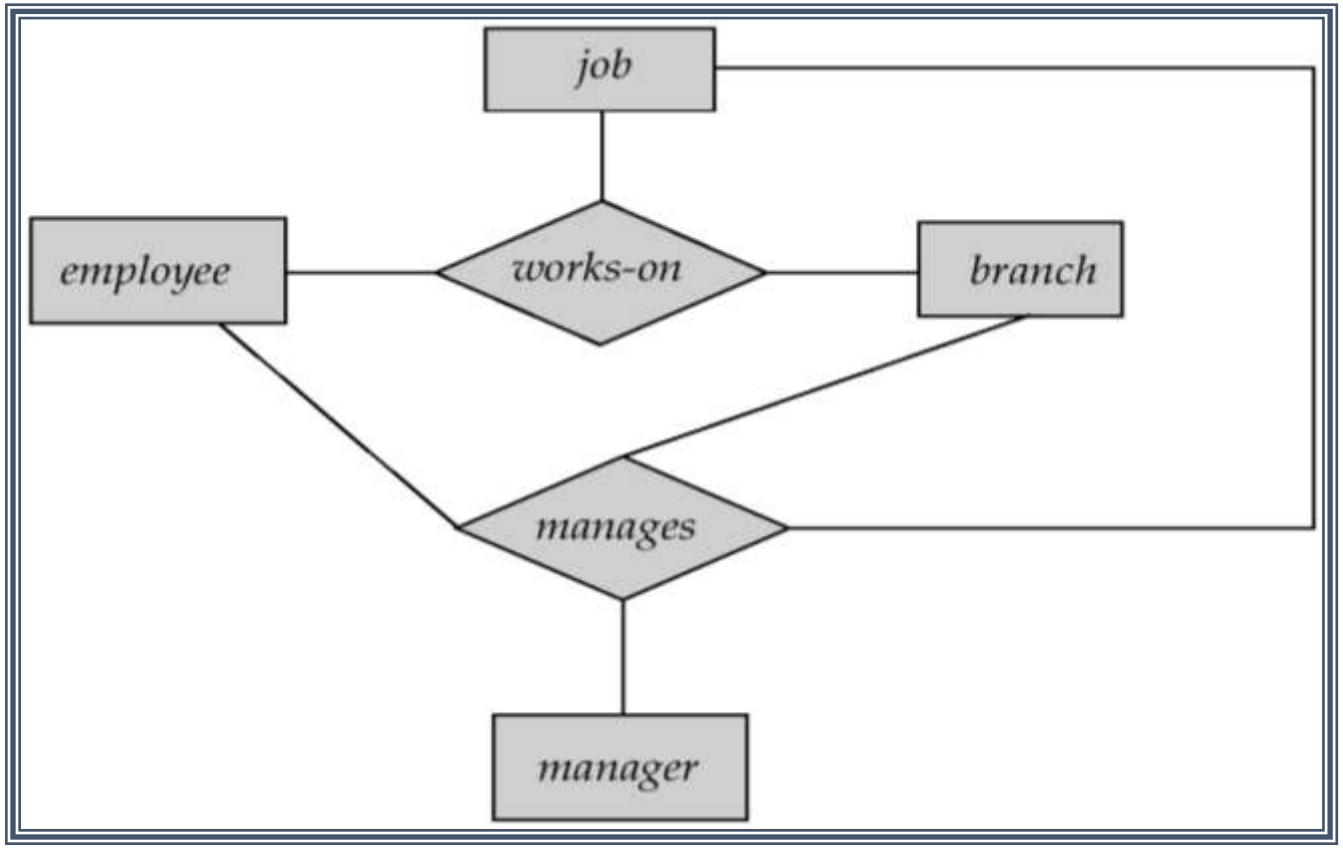
EER Diagram: Examples



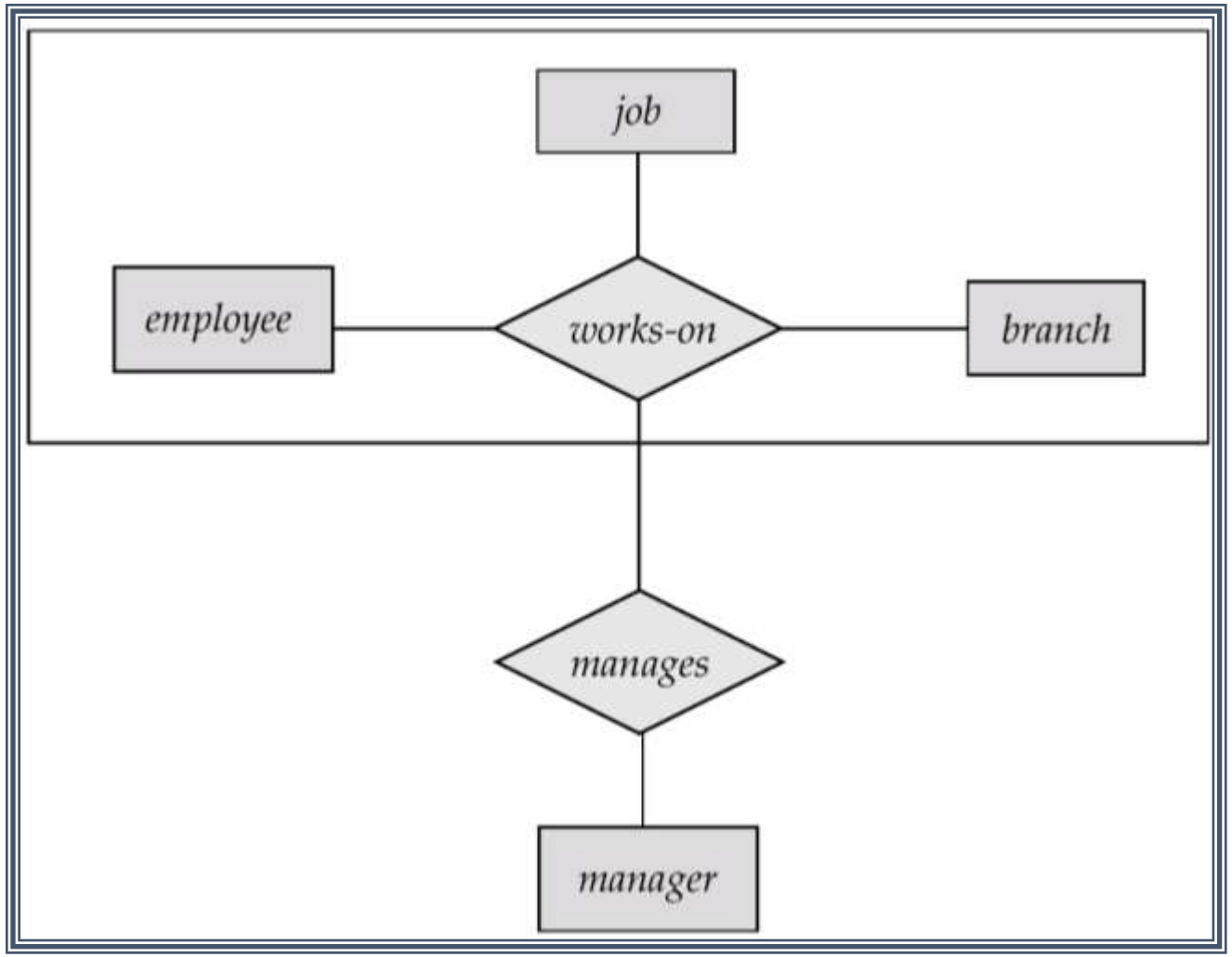
Category owner is the subclass of the set union of entity types person, company and bank

Aggregation

- Consider the ternary relationship *works-on*, which we saw earlier
- Suppose we want to record managers for tasks performed by an employee at a branch



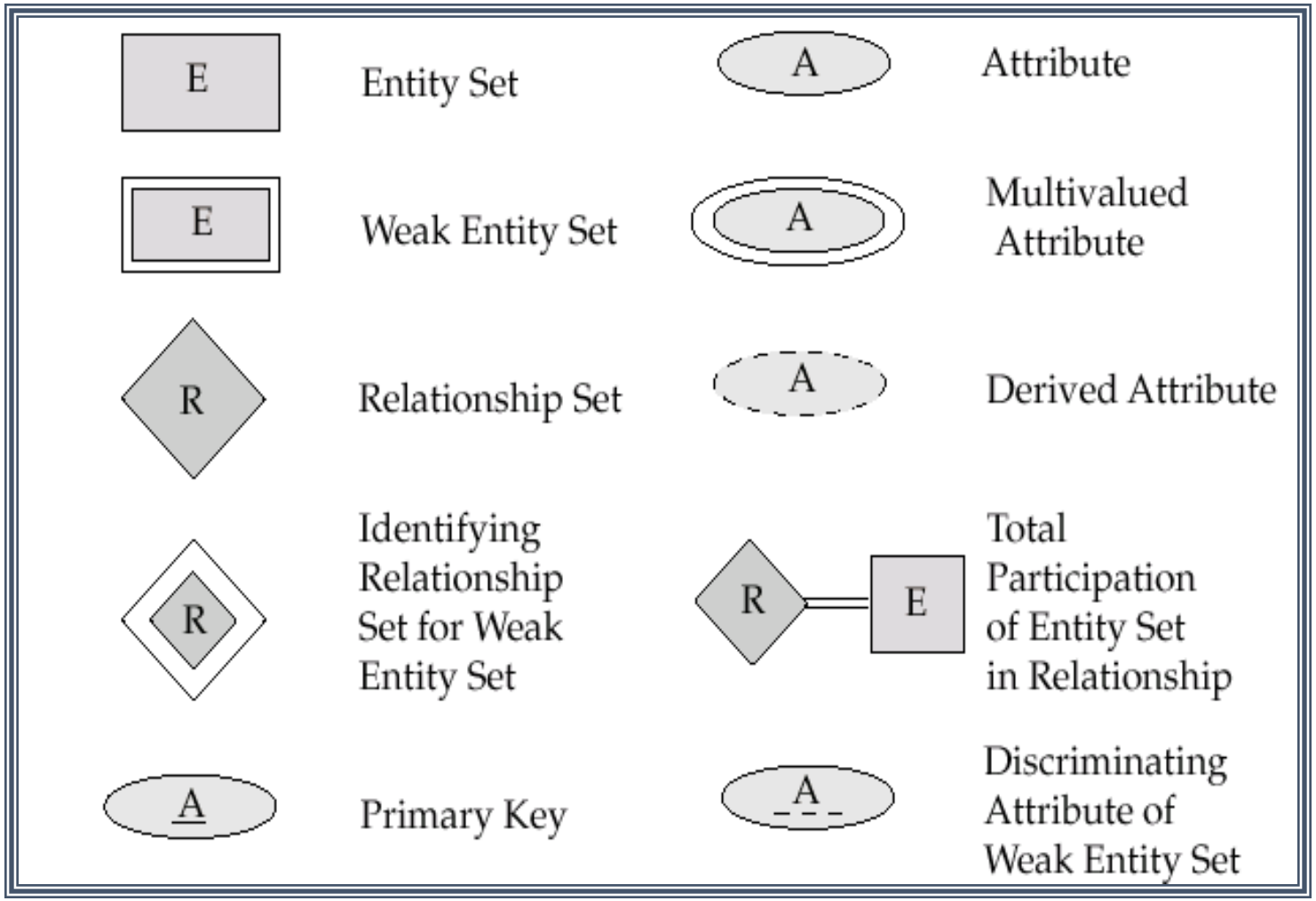
E-R Diagram With Aggregation



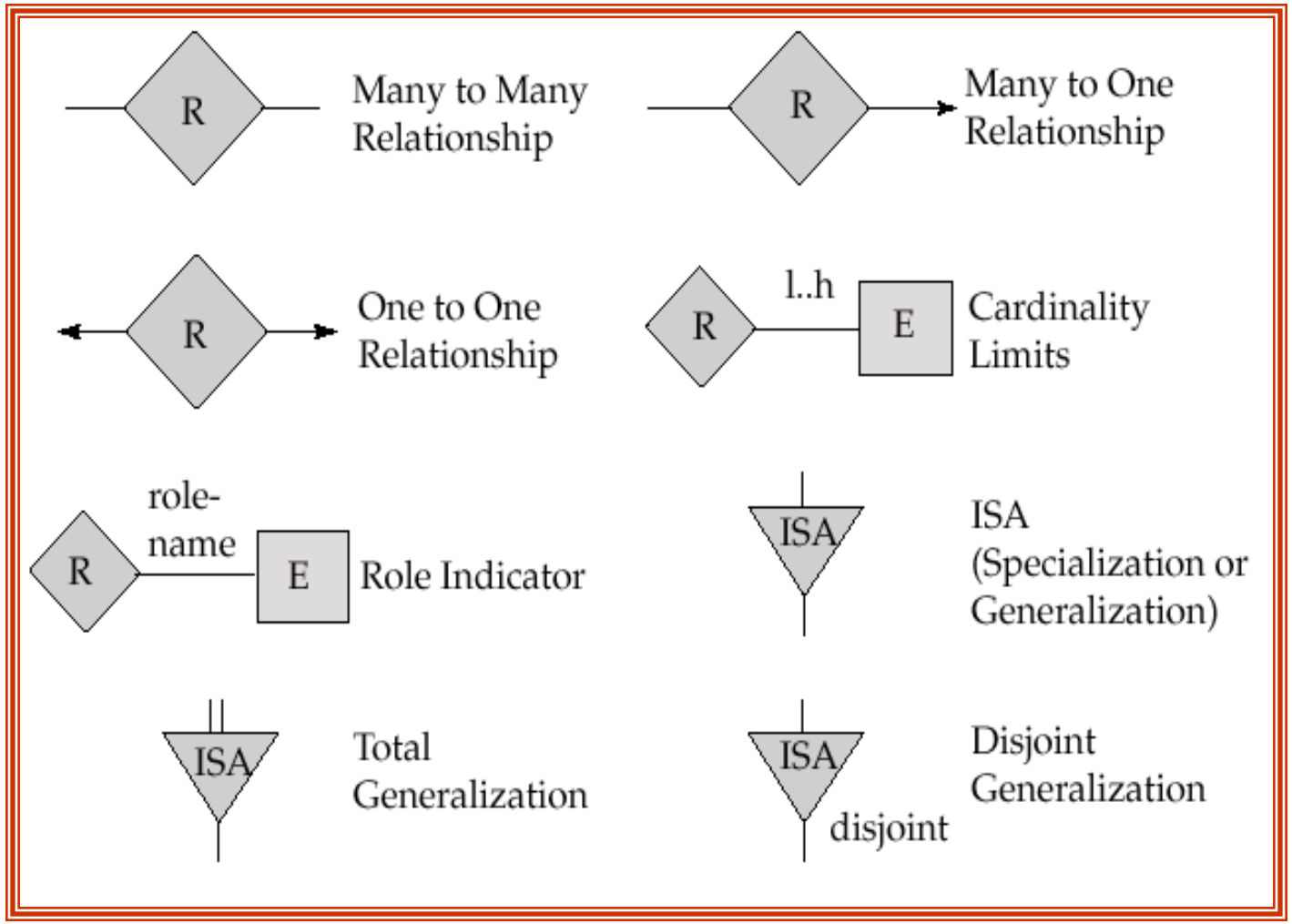
Design Considerations

- Use of entity sets vs. attributes
  - Whether we want to keep additional information
- Use of entity sets vs. relationship sets
  - Actions among entities are usually represented by relationships
- Binary versus *n*-ary relationship sets
  - N-nary relationships are usually more natural for actions among entity sets
- Weak entity set vs. strong entity set
- Generalization

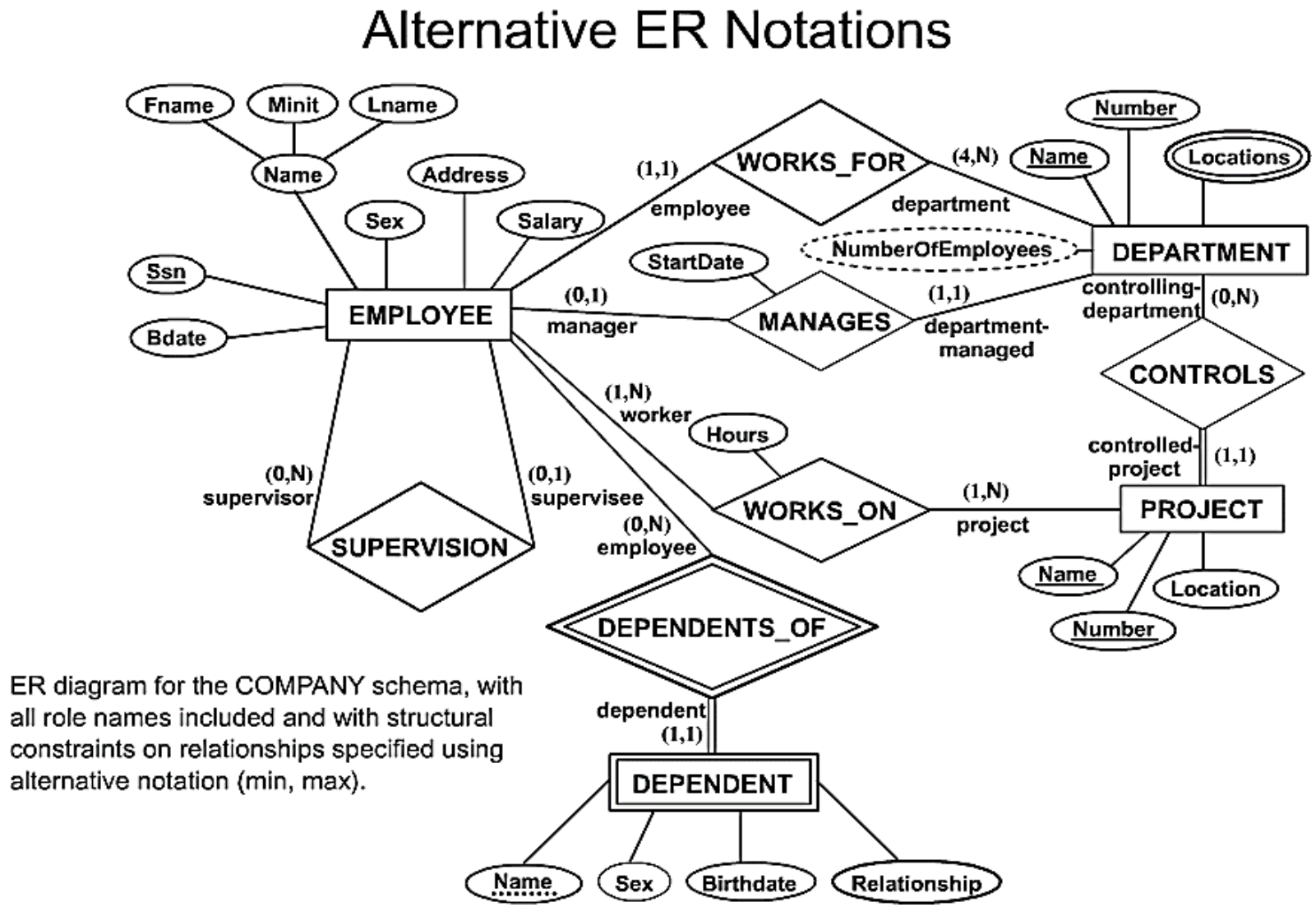
Notations


E
Entity Set
A
Attribute
E
Weak Entity Set
A
Multivalued Attribute
R
Relationship Set
A
Derived Attribute
R
Identifying Relationship Set for Weak Entity Set
R
E
Total Participation of Entity Set in Relationship
A
Primary Key
A
Discriminating Attribute of Weak Entity Set

Notations


R
Many to Many Relationship
R
Many to One Relationship
R
One to One Relationship
R
l..h
E
Cardinality Limits
R
role-name
E
Role Indicator
ISA
ISA (Specialization or Generalization)
ISA
Total Generalization
ISA
disjoint
Disjoint Generalization

COMPANY ER Schema Diagram using (min, max) notation



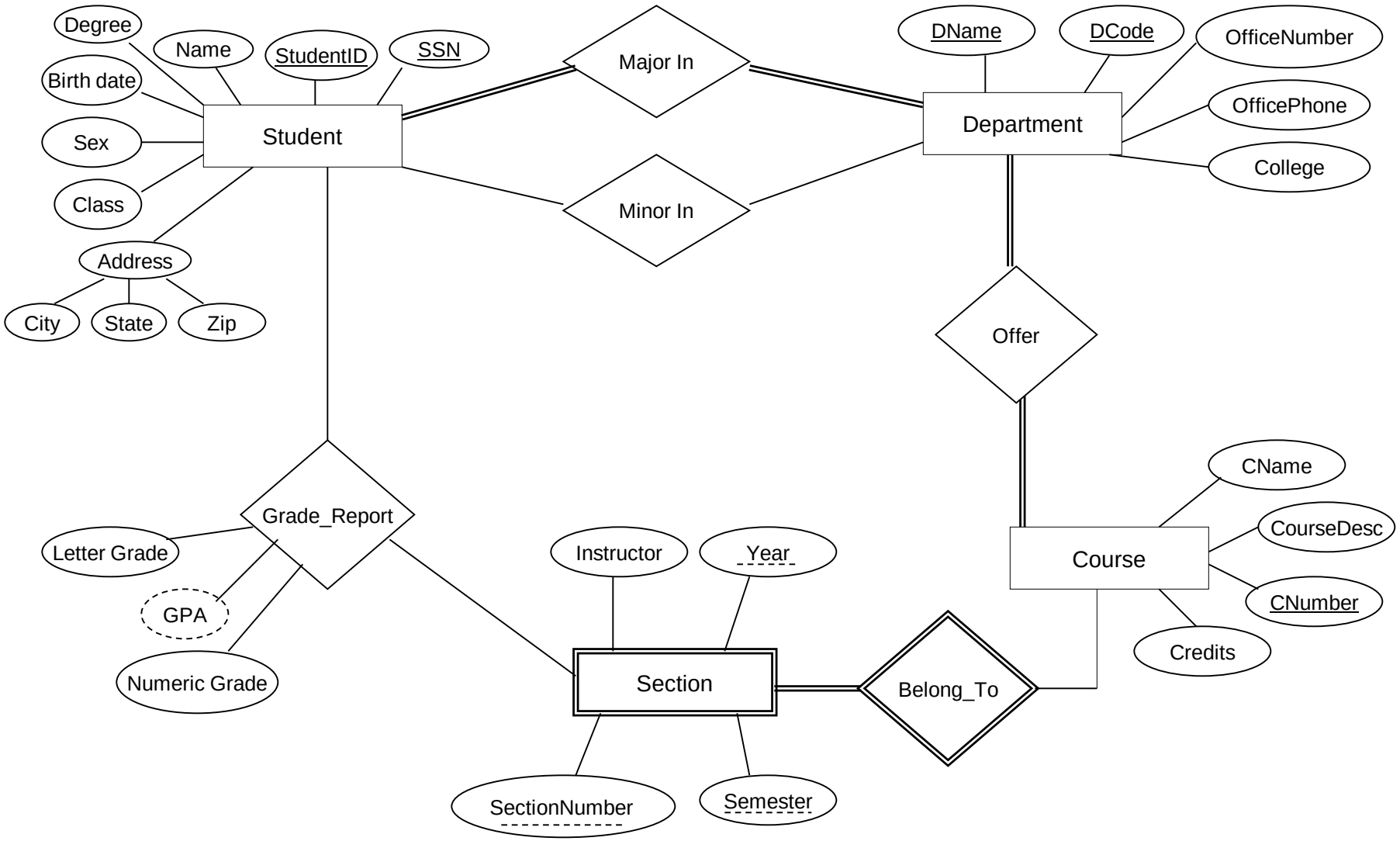
ER diagram for the COMPANY schema, with all role names included and with structural constraints on relationships specified using alternative notation (min, max).

Designing an ER Diagram

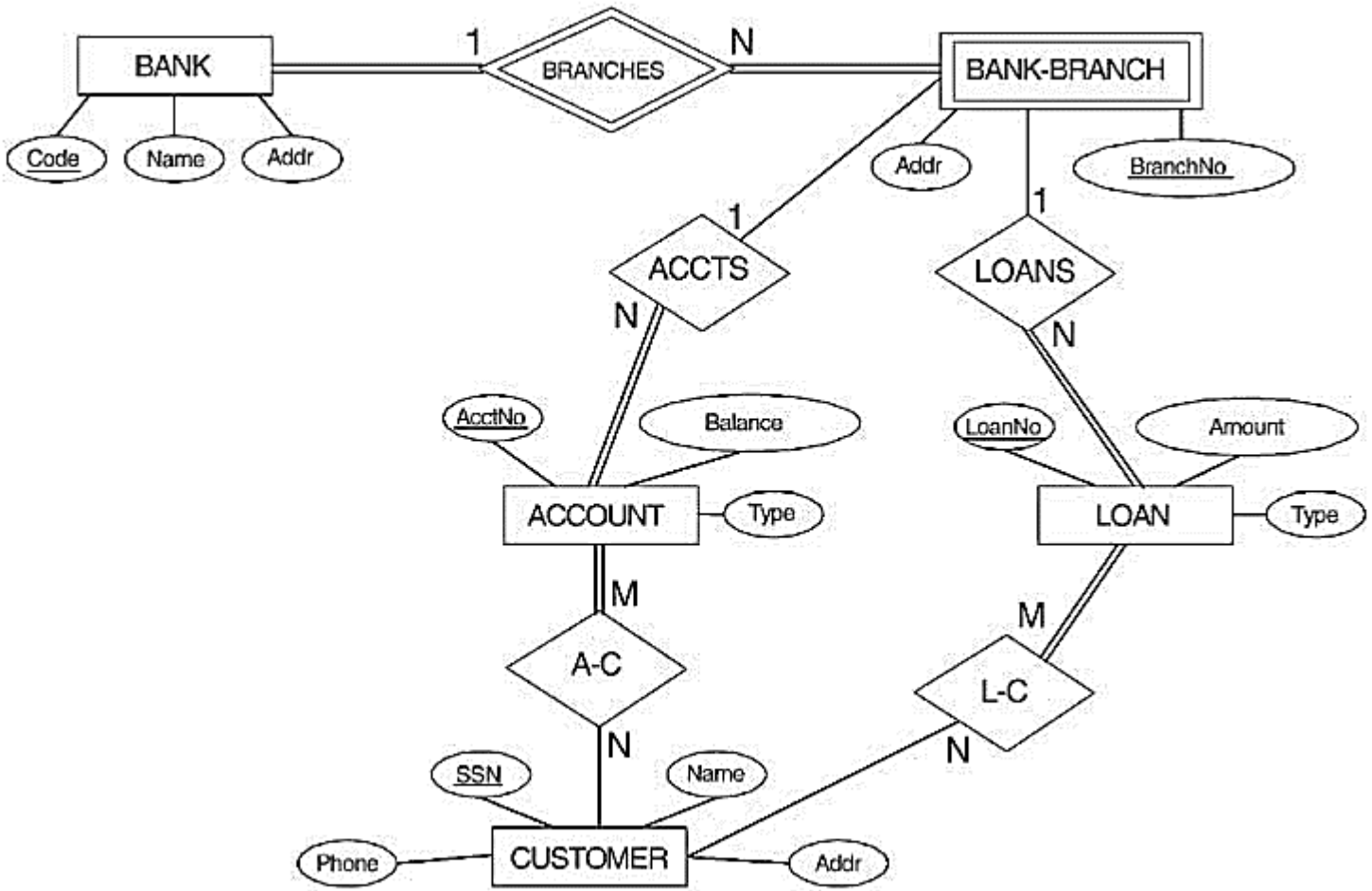
Consider the following set of requirements for a University database. Design an ER diagram for this application:

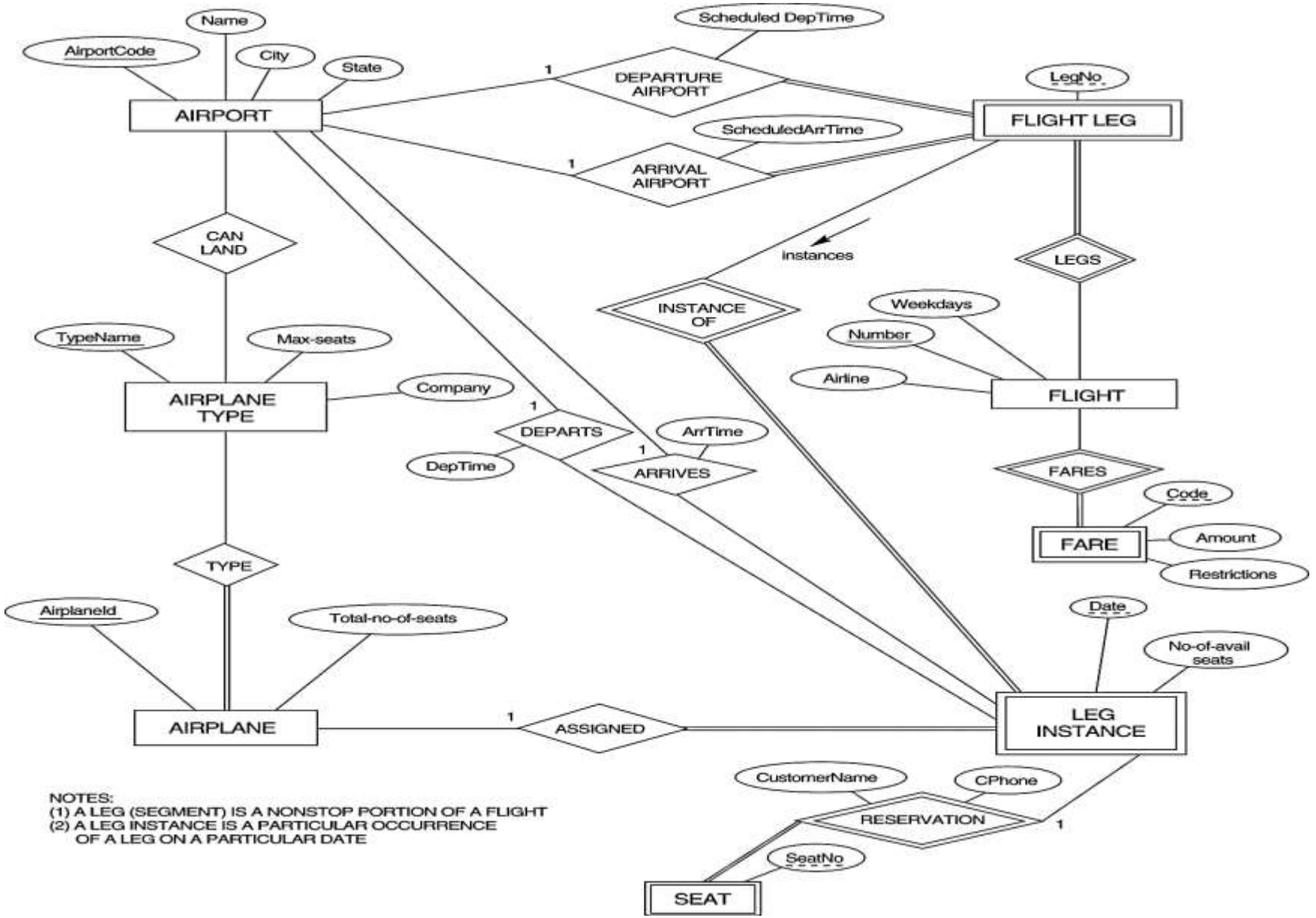
- The university keeps track of each student's name, student number, social security number, current address and phone number, permanent address and phone number, birthdate, sex, class (freshman, graduate), major department, minor department (if any), degree program (B.A., B.S., ... Ph.D.). Some user applications need to refer to the city, state, and zip code of the student's permanent address and to the student's last name. Both social security number and student number are unique for each student. All students will have at least a major department.
- Each department is described by a name, department code, office number, office phone, and college. Both the name and code have unique values for each department.
- Each course has a course name, description, course number, number of credits, level and offering department. The course number is unique for each course.
- Each section has an instructor, semester, year, course, and section number. The section number distinguishes sections of the same course that are taught during the same semester/year; its value is an integer (1, 2, 3, ... up to the number of sections taught during each semester).
- A grade report must be generated for each student that lists the section, letter grade, and numeric grade (0,1,2,3, or 4) for each student and calculates his or her average GPA.

University ER Diagram



ER DIAGRAM FOR A BANK DATABASE





An ER diagram for an AIRLINE database schema.

Enhanced ER Model

Superclass/Subclass

- Subclasses
  - Subgroupings of the entities of an entity type
  - An entity type as a superclass of subclasses
- Examples
  - EMPLOYEE ➡ {SALARIED_EMPLOYEE, HOURLY_EMPLOYEE}
  - PATIENT ➡ {OUTPATIENT, INPATIENT}
  - STUDENT ➡ {FULL-TIME, PART-TIME}

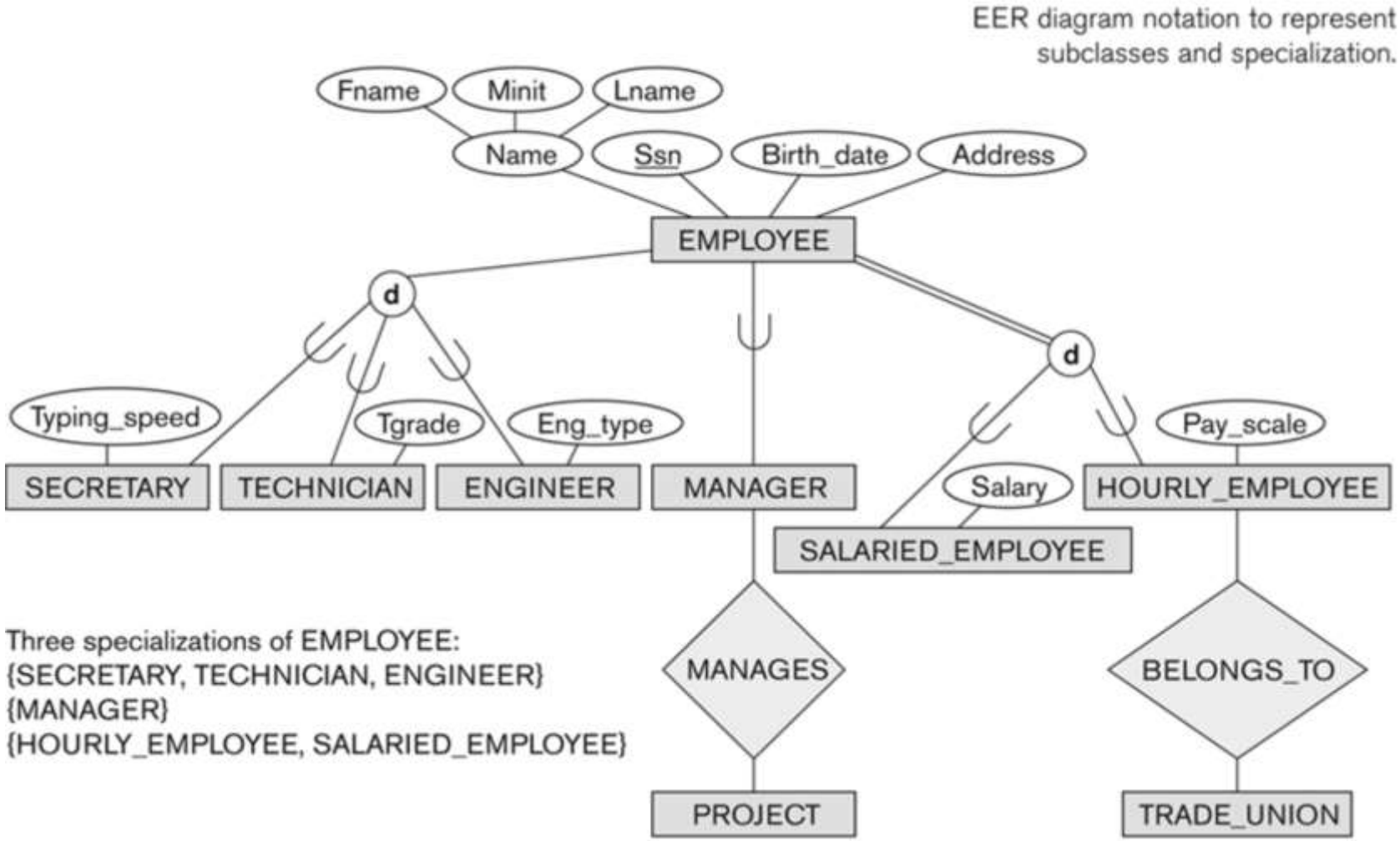
Superclass/Subclass

- “IS-A” relationship
  - Members of a subclass must be members of the superclass
  - A FULL-TIME STUDENT ISA STUDENT
  - Not every entity in a superclass be a member of a subclass
- Subclasses
  - Should have meaningful subgroupings
  - Should be related to database applications

Entity Type Inheritance

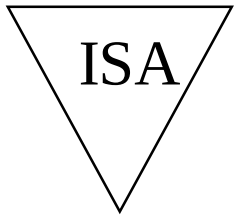
- Type Inheritance among Classes
- Inherited Attributes
  - a subclass inherits the attributes of the superclass
  - HOURLY_EMPLOYEE (Name, SSN, Address)
- Local (Specific) Attributes
  - a subclass may have its own attributes
  - HOURLY_EMPLOYEE (Hourly_rate)
  - SALARIED_EMPLOYEE (Annual_salary)

EER Example

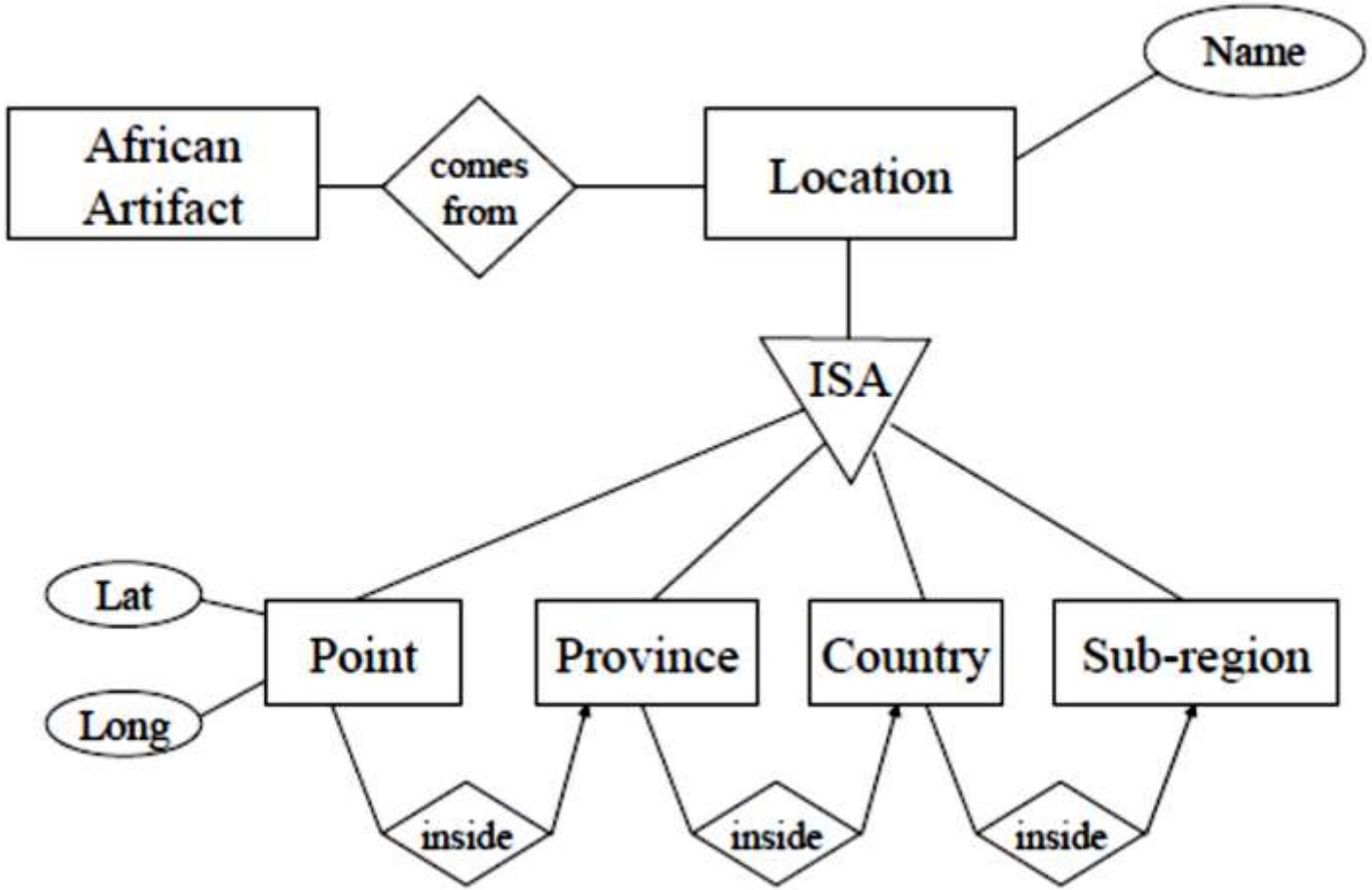

EER diagram notation to represent subclasses and specialization.
Fname
Minit
Lname
Name
Ssn
Birth_date
Address
EMPLOYEE
d
d
Typing_speed
Tgrade
Eng_type
Pay_scale
SECRETARY
TECHNICIAN
ENGINEER
MANAGER
Salary
HOURLY_EMPLOYEE
SALARIED_EMPLOYEE
Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}
MANAGES
BELONGS_TO
PROJECT
TRADE_UNION

Specialization & Generalization

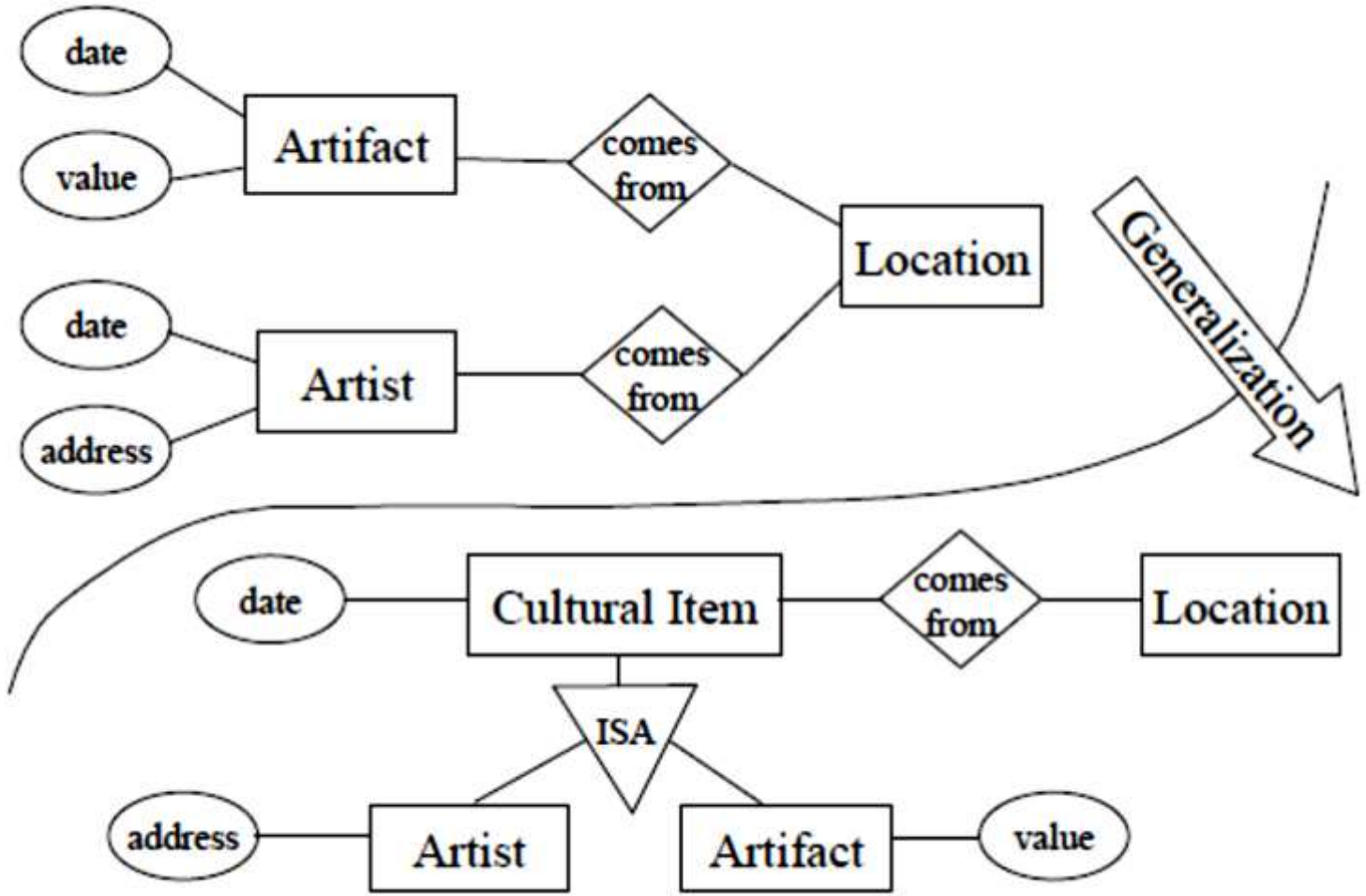
- superclass-subclass relationship
- **Specialization**: a means of identifying sub-groups within an entity set which have attributes which are not shared by all the entities (top-down)
- **Generalization**: Multiple entity sets are synthesized into a higher level entity set based on common features (bottom-up)
- E-R Symbol:



Specialization



Generalization



Inheritance

- A lower-level entity set **inherits** all the **attributes** of the higher level entity sets
- A subclass (lower level entity) also **inherits participation in the relationship** sets in which its superclass (higher-level entity) participates.
- The outcome of specialization and generalization is the same: hierarchy of entity sets.