Entity-Relationship Model

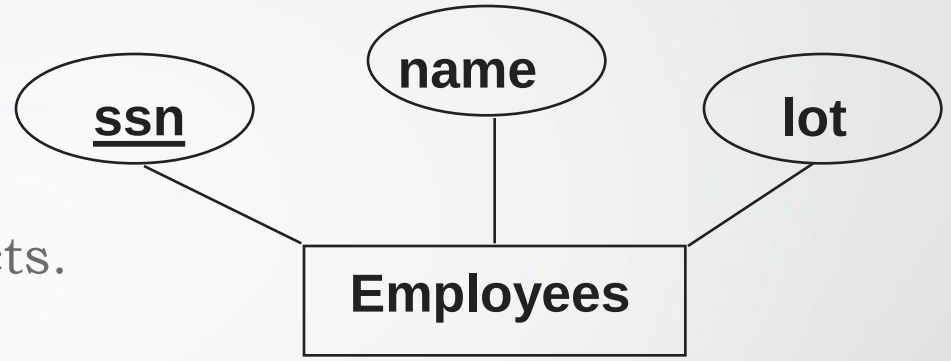
Overview of Database Design

- *Requirements Analysis:* Understand what data will be stored in the database, and the operations it will be subject to.
- *Conceptual Design*: *(ER Model is used at this stage.)*
  - What are the *entities* and *relationships* in the enterprise?
  - What information about these entities and relationships should we store in the database?
  - What are the *integrity constraints* or *business rules* that hold?
  - A database 'schema' in the ER Model can be represented pictorially (*ER diagrams*).
  - Can map an ER diagram into a relational schema.
- *Logical Design*: Convert the conceptual database design into the data model underlying the DBMS chosen for the application.

Overview of Database Design

- *Schema Refinement*: *(Normalization)* Check relational schema for redundancies and anomalies.
- *Physical Database Design and Tuning*: Consider typical workloads and further refinement of the database design (v.g. build indices).
- *Application and Security Design*: Consider aspects of the application beyond data. Methodologies like UML often used for addressing the complete software development cycle.

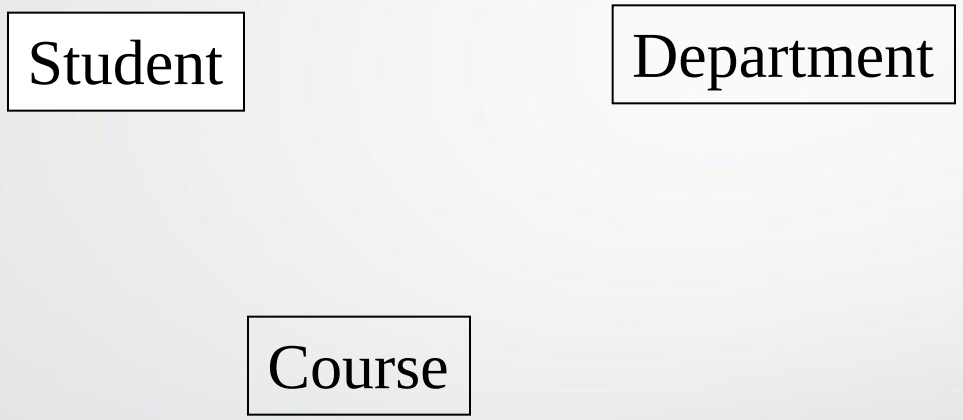
ER Model Basics



- *Entity:*
  - Real-world object distinguishable from other objects.
  - An entity is described using a set of *attributes*.

- *Entity Set*:
  - A collection of entities of the same kind. E.g., all employees.
  - All entities in an entity set have the same set of attributes.
  - Each entity set has a *key : a set of attributes uniquely identifying an entity*.
  - Each attribute has a *domain*.

We use a rectangular symbol to represent an entity set in the Entity Relationship Diagram.



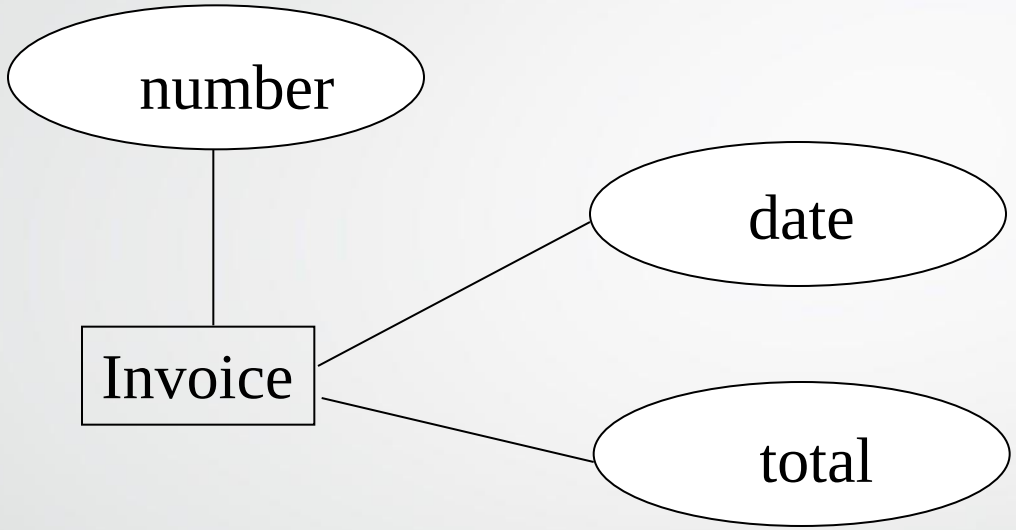
Attributes

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.
- Example:

  customer = (*customer_id*, *customer_name*, *customer_street*, *customer_city*)
  loan = (*loan_number*, *amount*)

- **Domain** – the set of permitted values for each attribute
- Attribute types:
  - *Simple* and *composite* attributes.
    - Surname; date of birth
    - address (street, town, postcode)
  - *Single-valued* and *multi-valued* attributes
    - Example: multivalued attribute: *phone_numbers*
  - *Derived* attributes
    - Can be computed from other attributes
    - Example: age, given date_of_birth

Attributes are represented using ovals.



An entity set, Invoice, with three attributes.

Atomic Attributes

- A simple, or atomic, attribute is one that cannot be decomposed into meaningful components
- Example : product price, prodPrice, cannot be decomposed, because you cannot subdivide prodPrice into a finer set of meaningful attributes.
- The value of the attribute prodPrice could be ₹21.50.
- one could decompose prodPrice into two attributes
  - The Rupees component (21), and
  - the Paisa component (50)
- But our assumption here is that such a decomposition is not meaningful to the intended application or system that will make use of it.

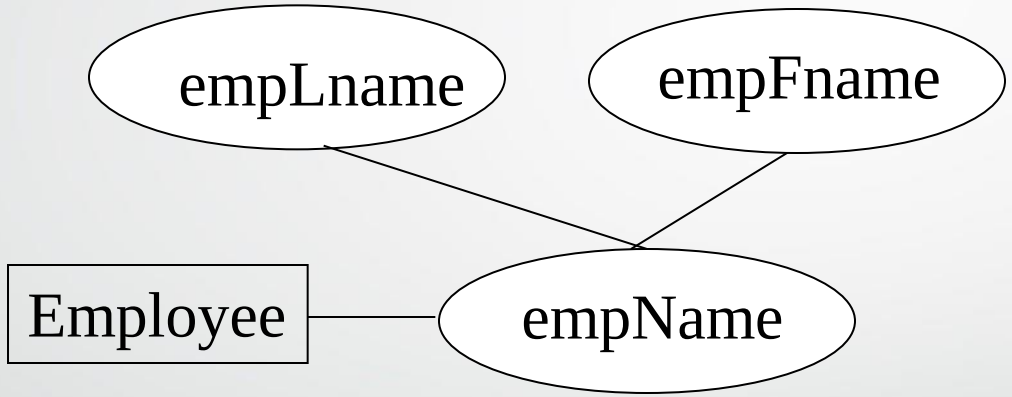
Exercise : Make a list of attributes that would be required for entity sets

a) a student,
b) an employee .

Consider each of your attributes, and verify that each is atomic. Create an ERD that illustrates the entity set and its attributes.

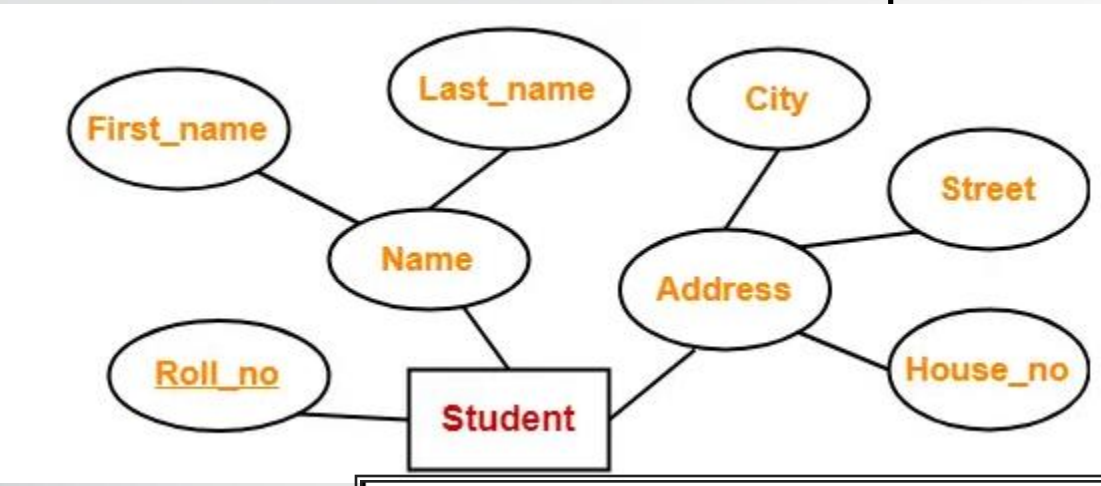
Composite Attributes

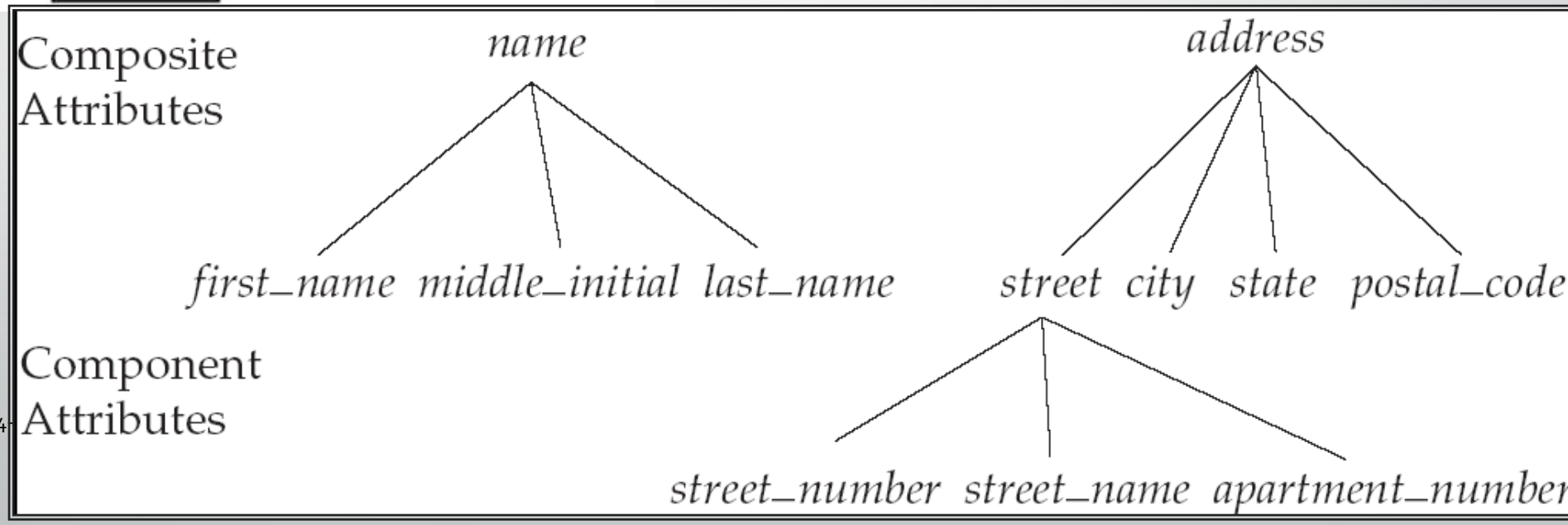
A composite attribute is an attribute that is shown as comprising two or more simpler attributes; we show a composite attribute in the figure below.



empName shown as a composite attribute

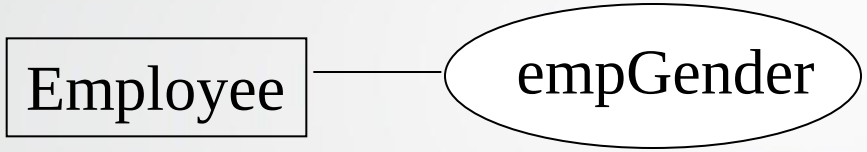
Exercise : Many entities will need an address attribute. For example, a student has an address, an employee has an address. Show, in an ERD, how the address attribute can be shown as a composite attribute.





Single-valued Attributes

An attribute is single-valued if, for any instance of the pertinent entity set, there is only one value at a given time for the attribute.



Exercise : A college or university will keep track of several addresses for a student, but each of these can be named differently: for example, consider that a student has a mailing address and a home address. Create an ER for a student entity set with two composite single-valued attributes for student addresses.

Multi-valued Attributes

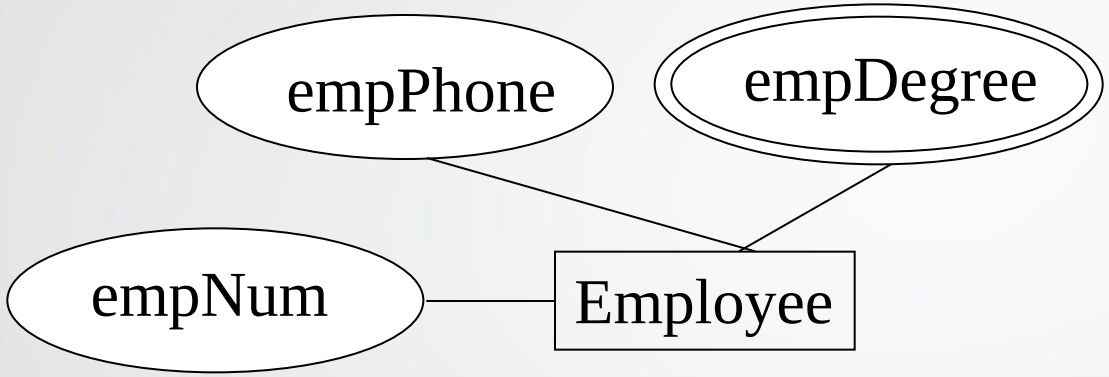
- Consider an attribute to track each employee's university degrees - empDegree.
- Since an employee could have none, one, or several degrees, we say empDegree is multi-valued.

sample data for three employees.

empNum	empPhone	empDegree
123	233-9876	
333	233-1231	BA, BSc, PhD
679	233-1231	BSc, MSc

One of these employees has no degrees, another has 3 degrees, and the last one has 2 degrees.

empDegree is shown as a multi-valued attribute



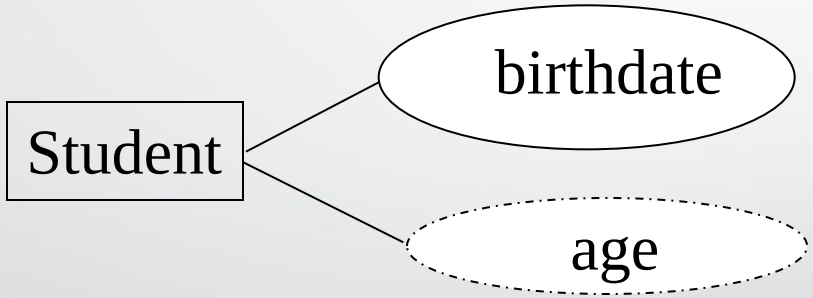
The presence of a multi-valued attribute indicates an area that may require more analysis.

Derived Attributes

If an attribute's value can be derived from the values of other attributes, then the attribute is derivable, and is said to be a **derived** attribute

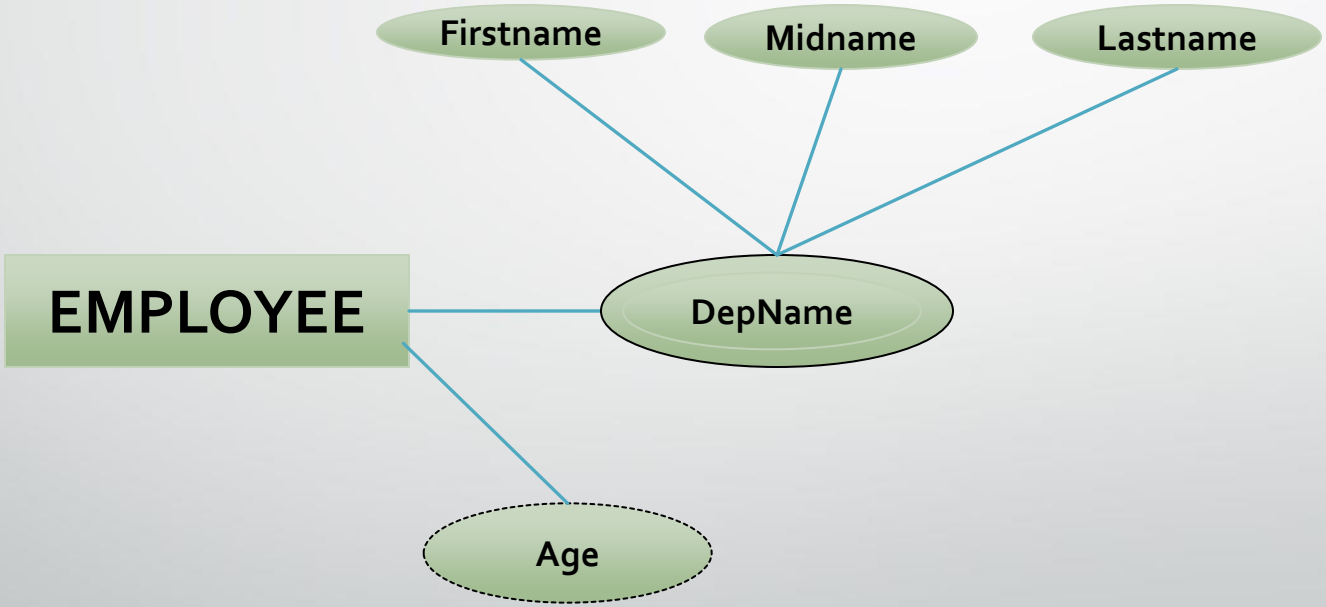
Example

if we have two attributes for an employee, birth date and current age, then age is derivable by subtracting the birth date from the current date.



The age of the employee is a derived attribute.

Task: Consider the employee entity set. Suppose the company needs to track the names of the dependents for each employee. Show the empDependentName as a multi-valued attribute. Modify ERD to show empDependentName as a composite attribute comprising first and last names and middle initials. Represent empAge as derived attribute.



Relationship

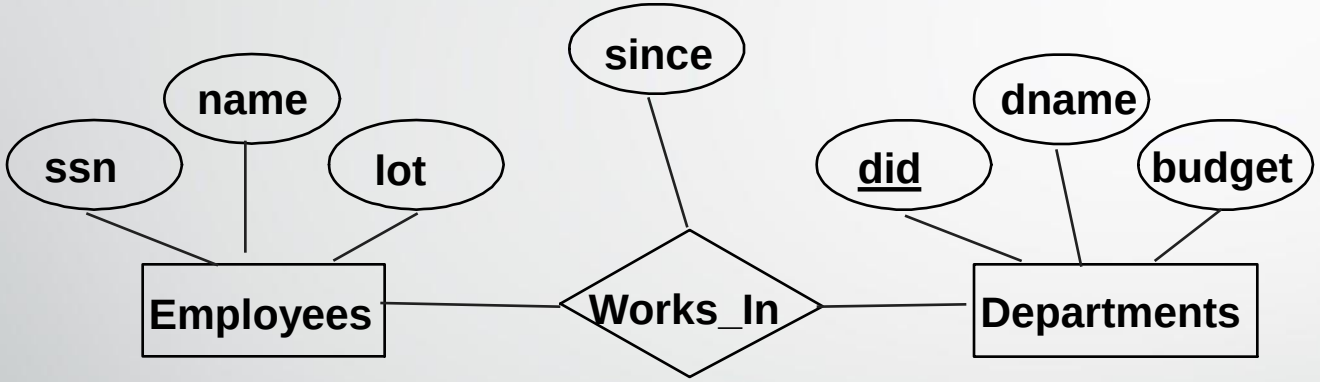
- *Relationship*:
  - Association among two or more entities. E.g., Rahul works in Analytics dept.
- *Relationship Set*: Collection of similar relationships.
  - Relationship sets can also have *descriptive attributes* (e.g., the *since* attribute of Works In).
  - A relationship is uniquely identified by participating entities without reference to descriptive attributes.

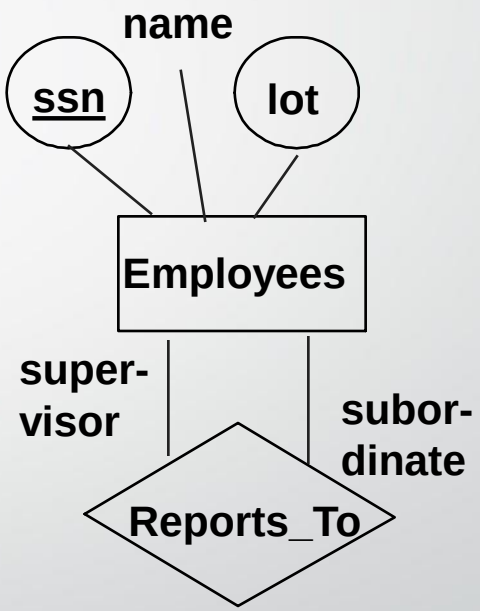
Degree of a Relationship Set

- Refers to number of entity sets that participate in a relationship set.
  - **binary**
  - Ternary
  - N-ary

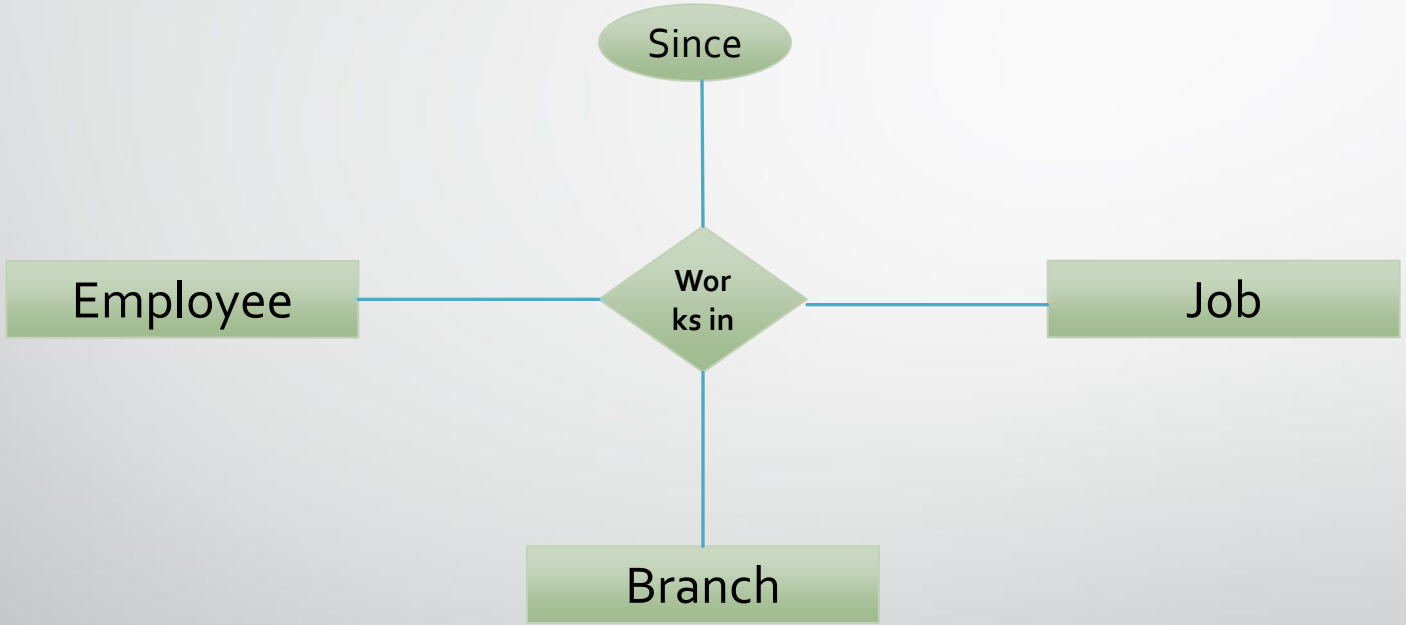
- Example: Suppose employees of a bank may have jobs (responsibilities) at multiple branches, with different jobs at different branches. Then there is a **ternary relationship** set between entity sets *employee, job, and branch*

Relationship





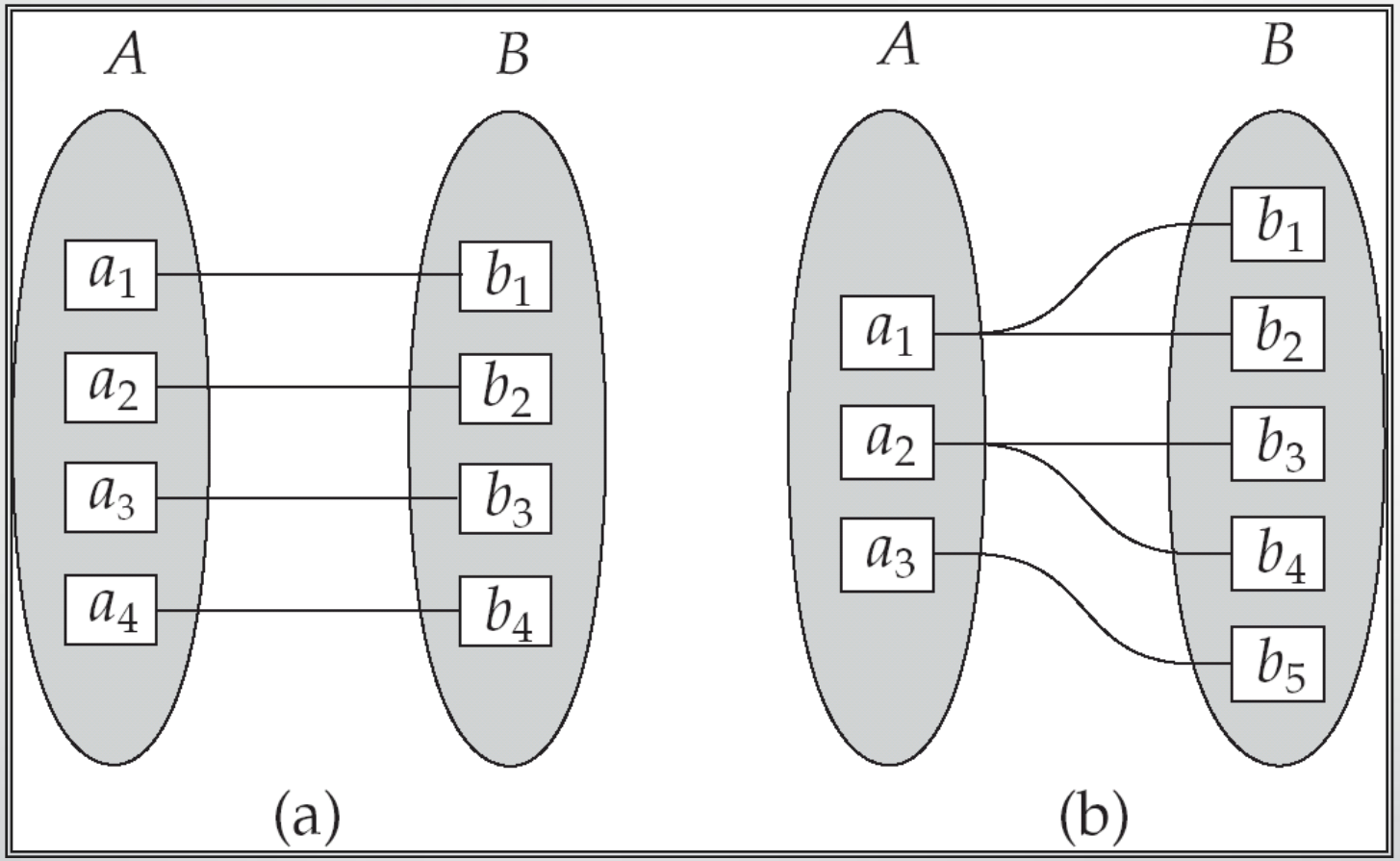
Ternary relationship



Mapping Cardinality Constraints

- Express the number of entities to which another entity can be associated via a relationship set.
- Most useful in describing binary relationship sets.
- For a binary relationship set the mapping cardinality must be one of the following types:
  - One to one –Indian citizen & Aadhar Number
  - One to many – academic advisors
  - Many to one – same as one-to-many
  - Many to many – depositors

Mapping Cardinalities

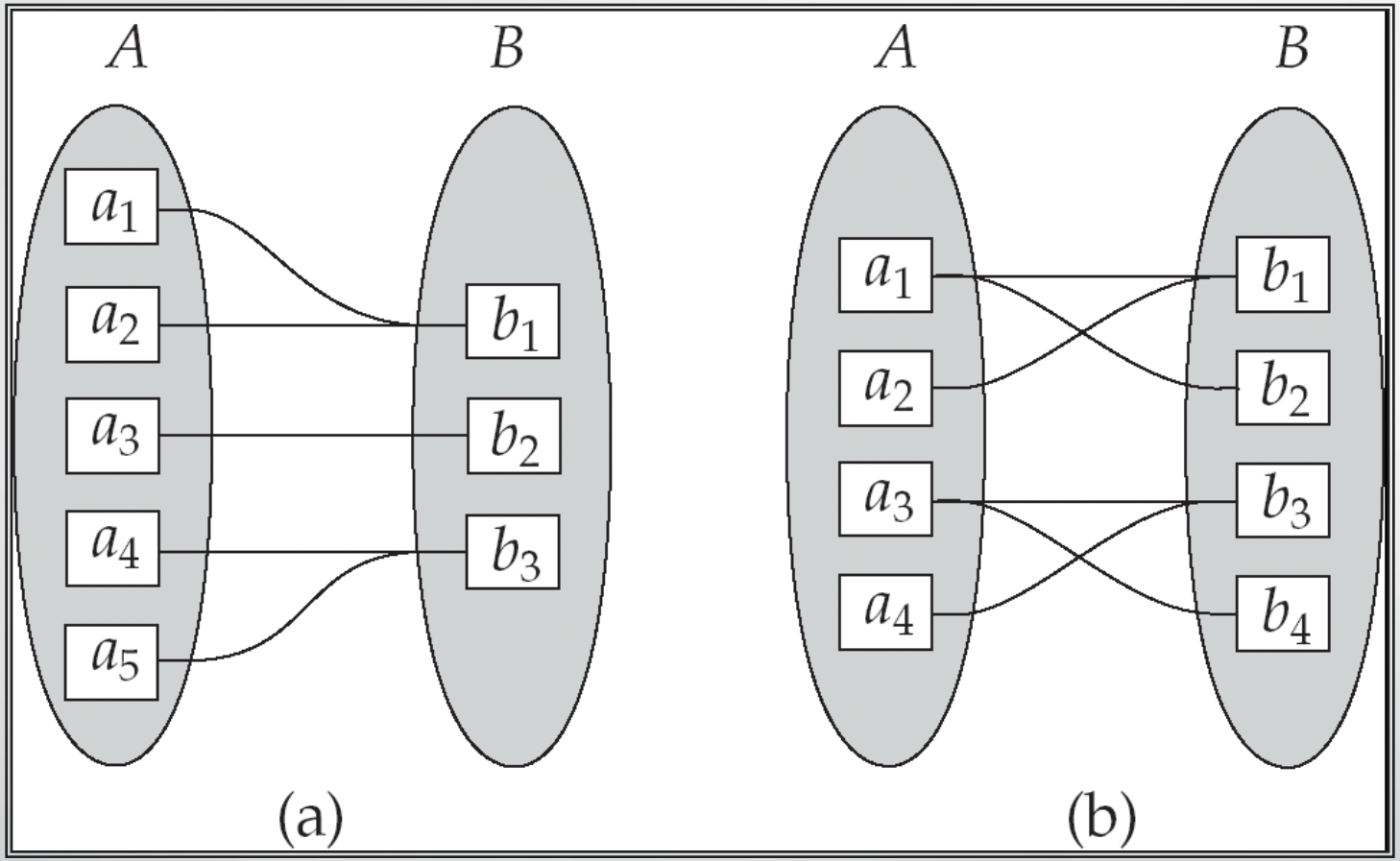


One to one

One to many

Note: Some elements in *A* and *B* may not be mapped to any elements in the other set

Mapping Cardinalities

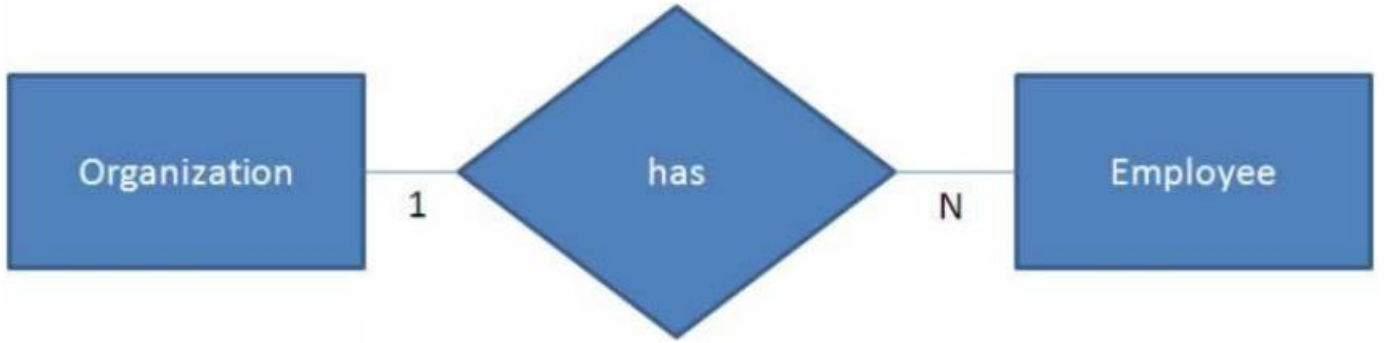


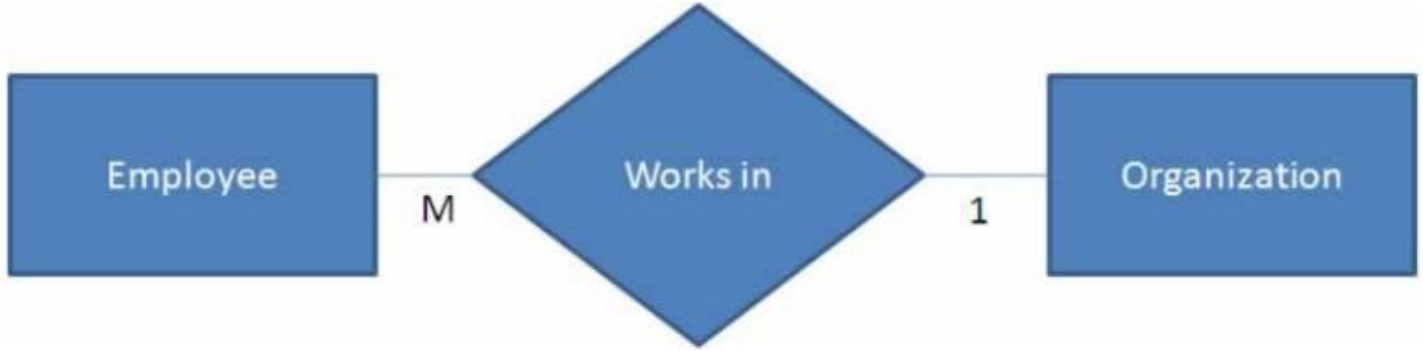
Many to one

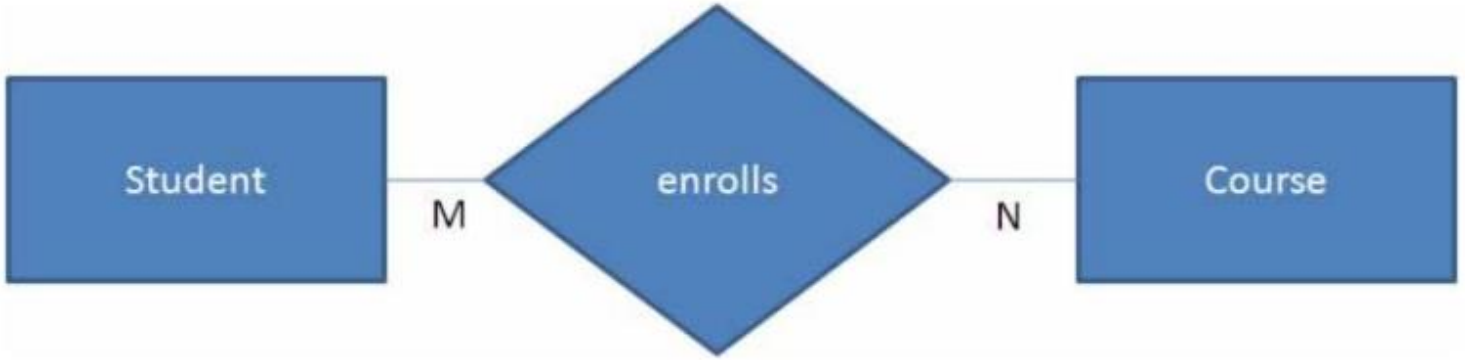
Many to many

Note: Some elements in A and B may not be mapped to any elements in the other set


Employee
1
Assigned With
1
Parking Space


Organization
1
has
N
Employee


Employee
M
Works in
1
Organization

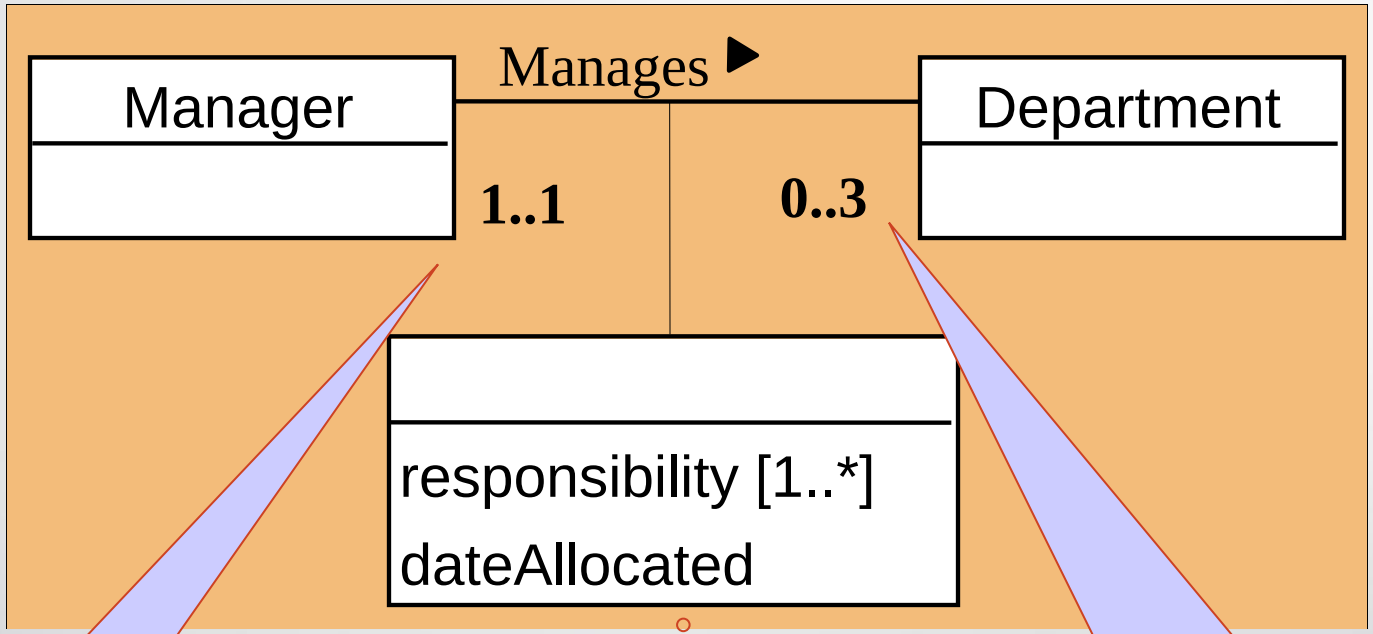

Student
M
enrolls
N
Course

Relationships: Multiplicity

label lines to show cardinality and participation

0..1	"zero or one"	optional
0..*	"zero or more"	optional
1..1	"one"	mandatory
1..4	"between 1 and 4"	mandatory
1..*	"one or more"	mandatory

Relationships example

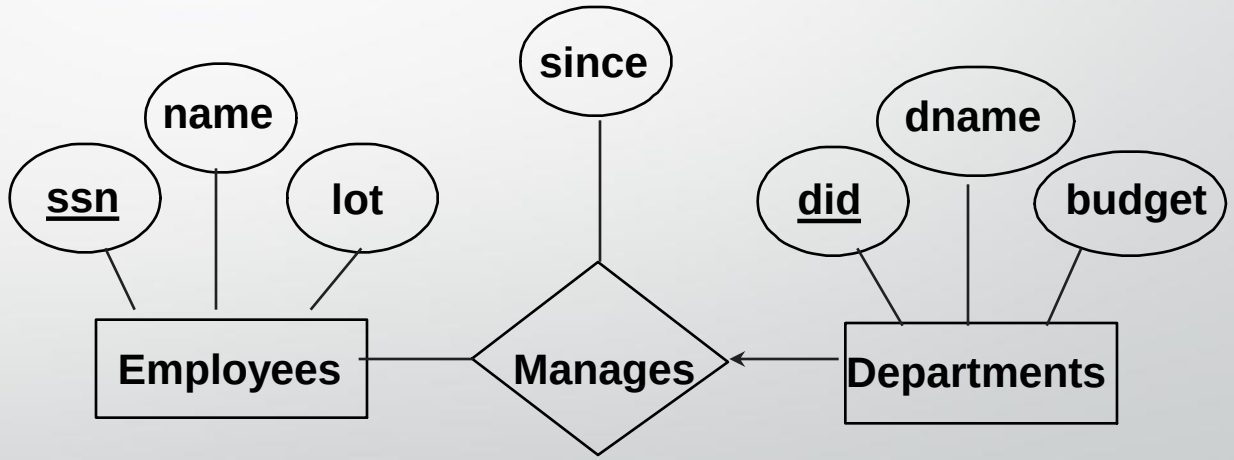


E-R Model

- Draw an E-R diagram for this scenario "A student registers for up to 8 modules and each module has many students on it. Record the student ID, their full name and address and also each module ID and title. We also want to hold the grade attained by each student for each module"
    - Remember to show in your model:
        - All primary keys,
        - Entities
        - Relationships
        - Attributes

Key Constraints (a.k.a. Cardinality)

- Consider Works_In : An employee can work in many departments; a dept can have many employees.
- In contrast, each dept has at most one manager, according to the *key constraint* on Manages.

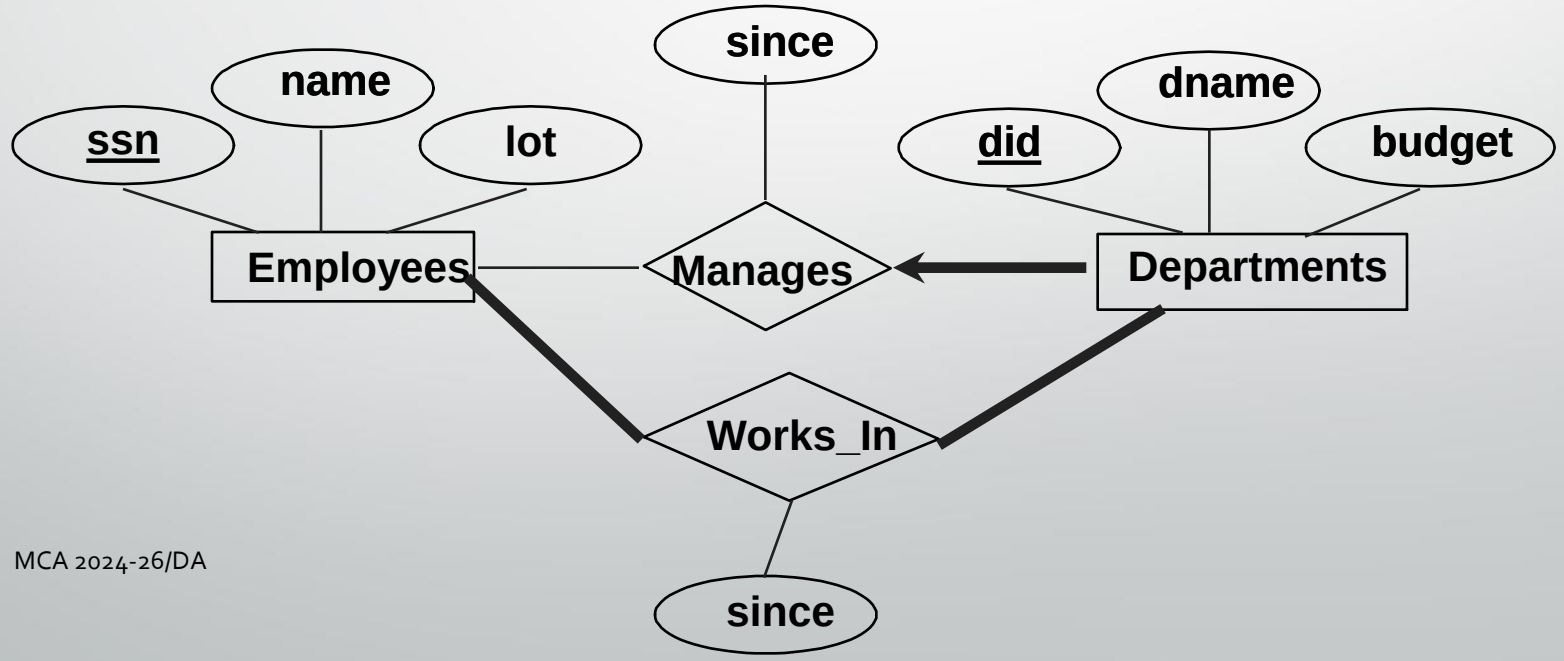


Participation Constraints

- **Total Participation**: Each entity in the entity is involved in the relationship. Total participation is represented by double lines.
- **Partial participation**: Not all entities are involved in the relationship. Partial participation is represented by single line.

Participation Constraints

- Does every department have a manager?
  - If so, this is a *participation constraint*: the participation of Departments in Manages is said to be *total* (vs. *partial*).
    - Every Department MUST have at least an employee
    - Every employee MUST work at least in one department
    - There may exist employees managing no department



Create an ER model to depict Total Participation

Relationships: constraints

- The ***degree*** of a relationship type
  - binary (connects 2 entity types)
  - unary/ recursive (connects 1 entity type with itself)
  - complex (connects 3 or more entity types)
    - Ternary (connects 3)

Degree

- Relationship constraints - ***cardinality***
  - one to one (1:1)
  - one to many (1:m)
  - many to many (m:n)
- Relationship constraints – ***participation***
  - full/mandatory
  - or partial/optional

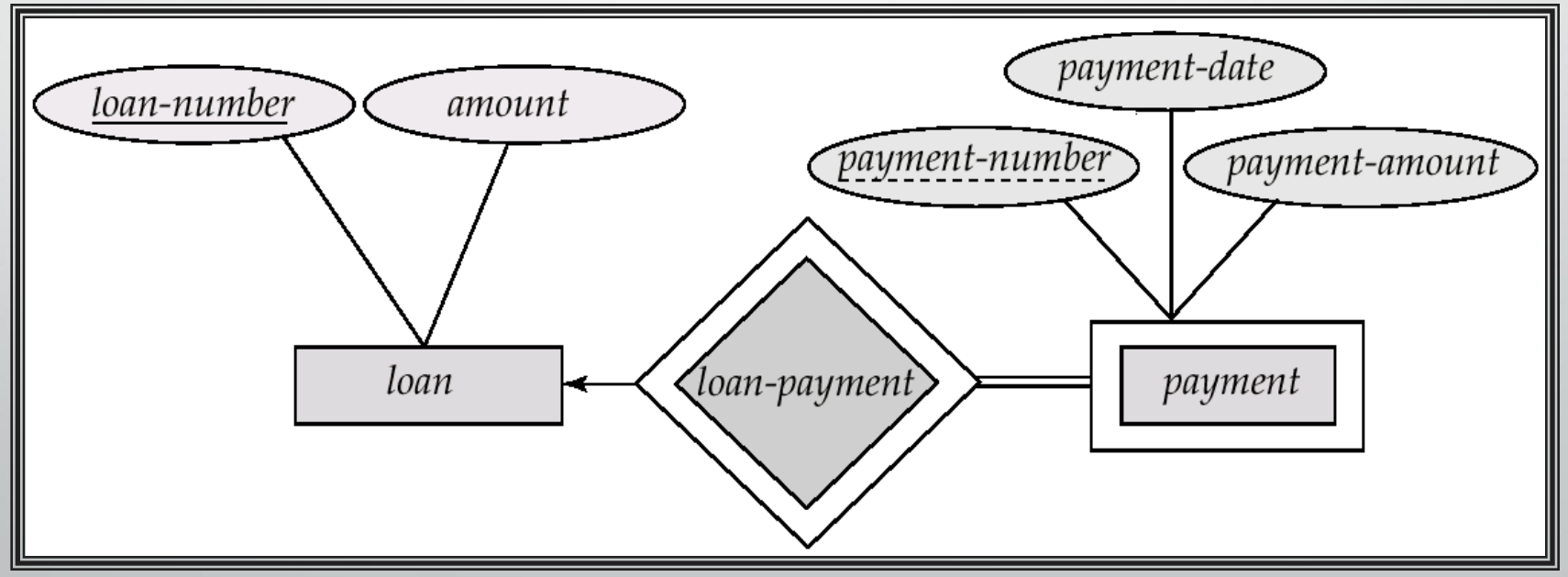
Multiplicity

Weak Entity Sets

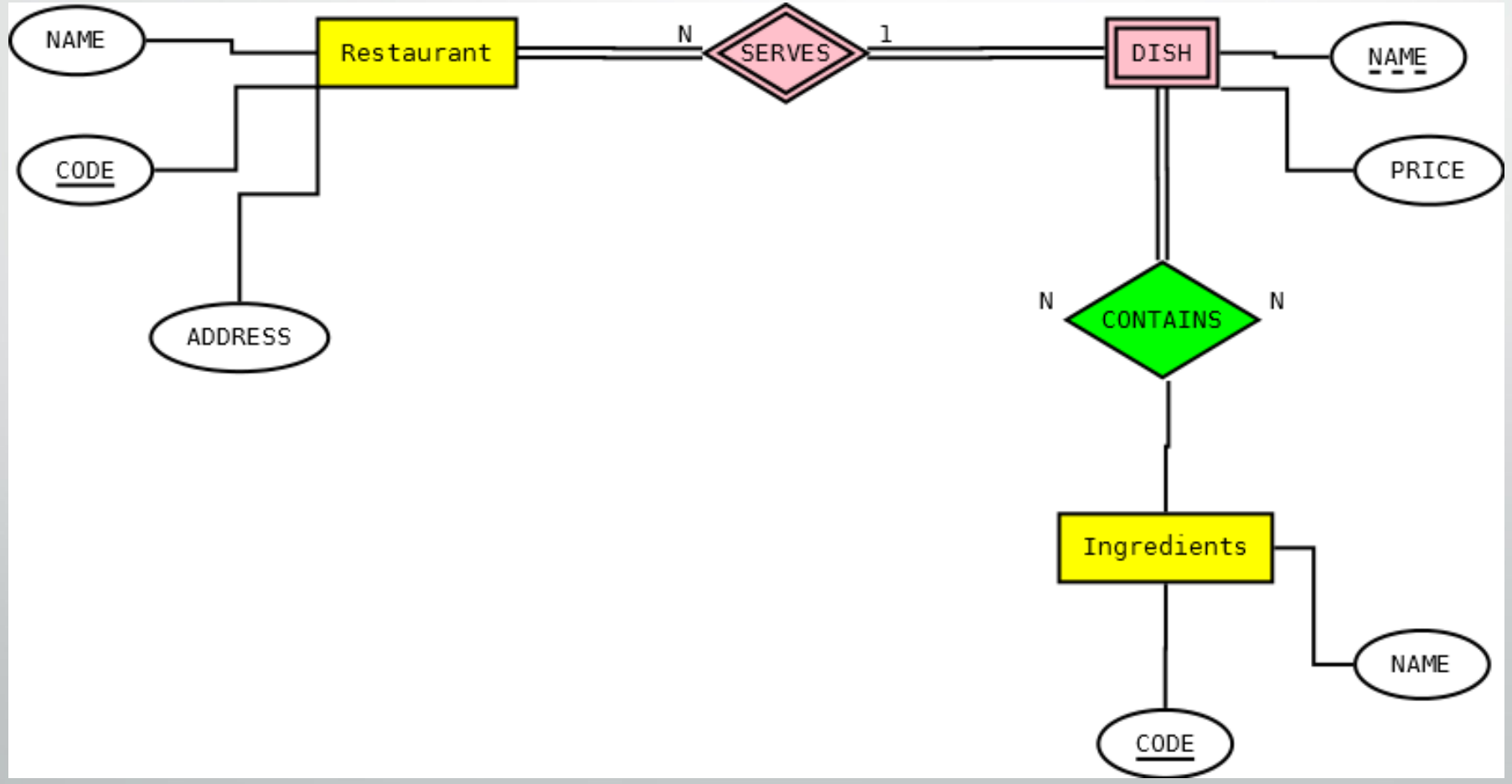
- An entity set that does not have a **primary key** is referred to as a **weak entity set**.
- The existence of a weak entity set depends on the existence of a **identifying entity set**
- The **discriminator** (*or partial key)* of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set.
- The primary key of a weak entity set is formed by the primary key of the strong entity set on which the weak entity set is existence dependent, plus the weak entity set's discriminator.

Weak Entity Set Notations

- Double rectangles for weak entity set
- Double diamond for weak entity relationship
- Dashed underscore for discriminator

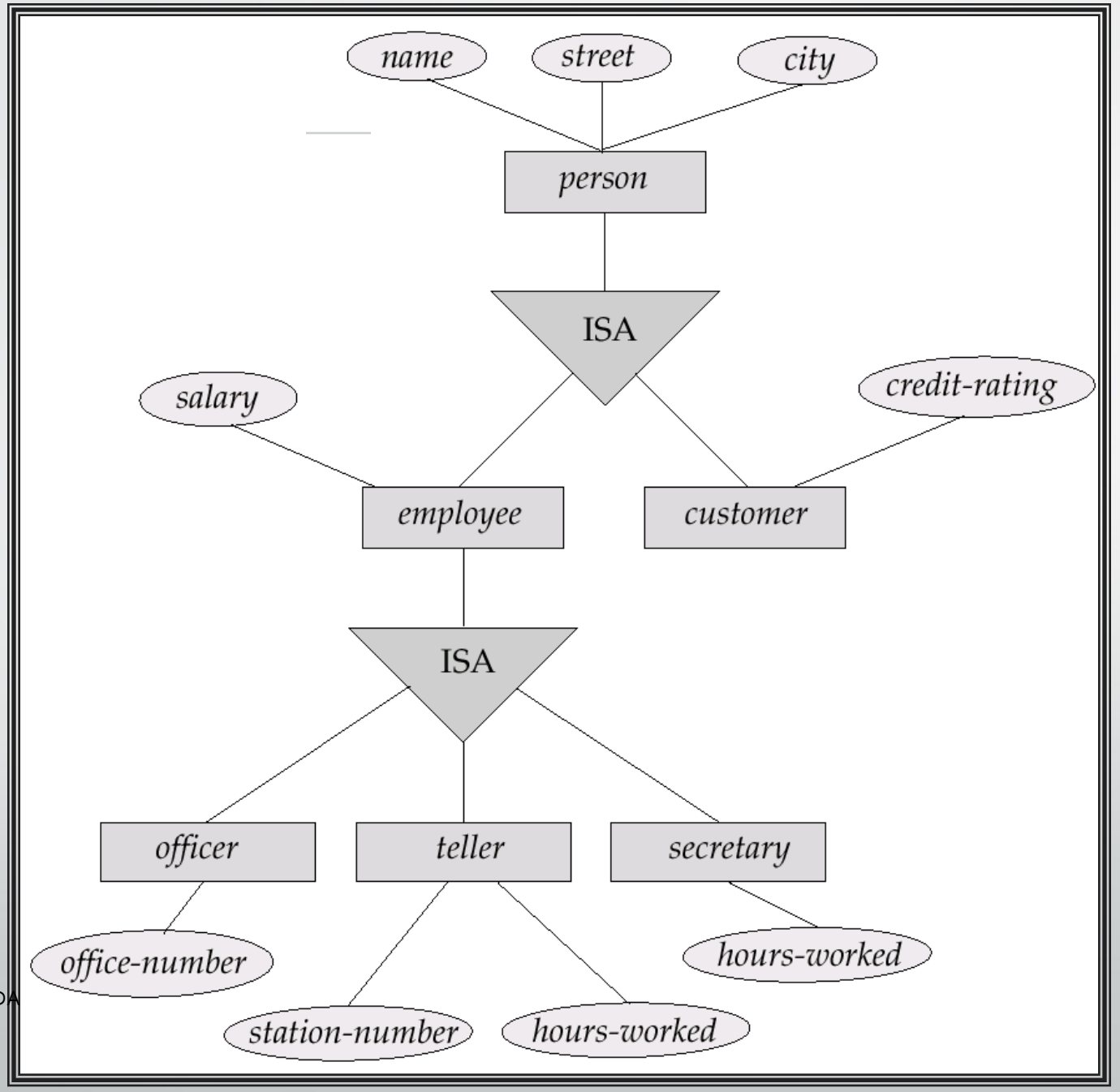


Design an ER model for a restaurant where each dish on the menu belongs to a specific restaurant, and every restaurant must have at least one dish with its ingredients in its menu.



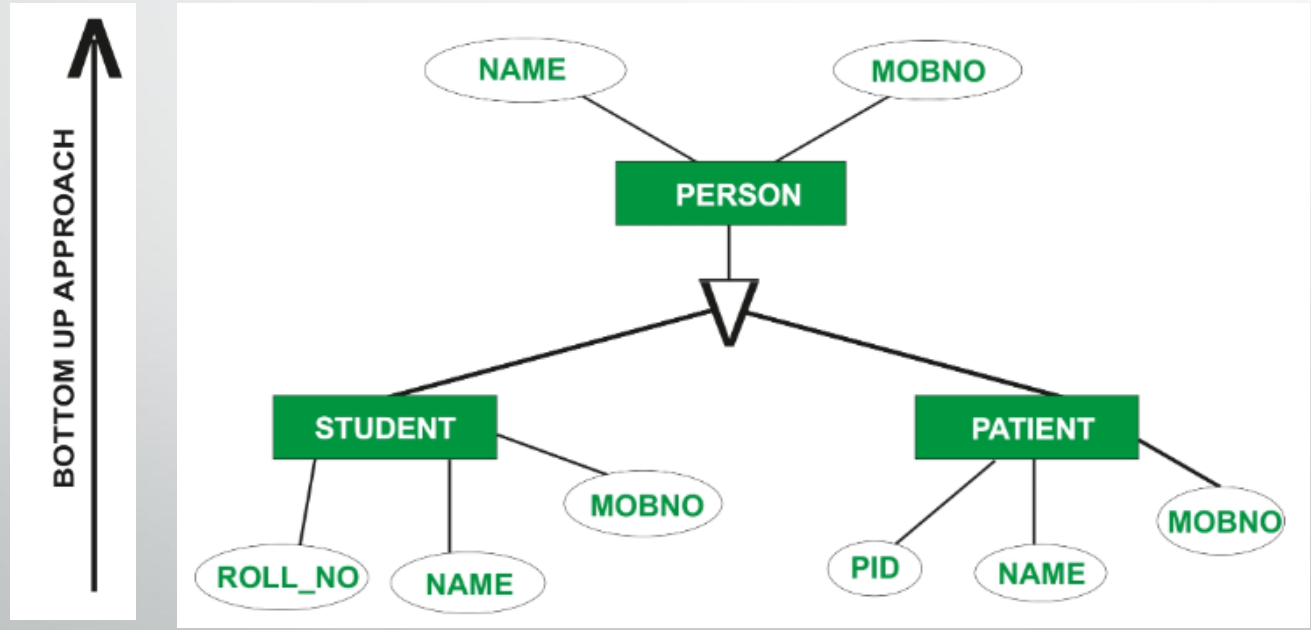
Specialization

- 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
hours-worked
station-number
hours-worked

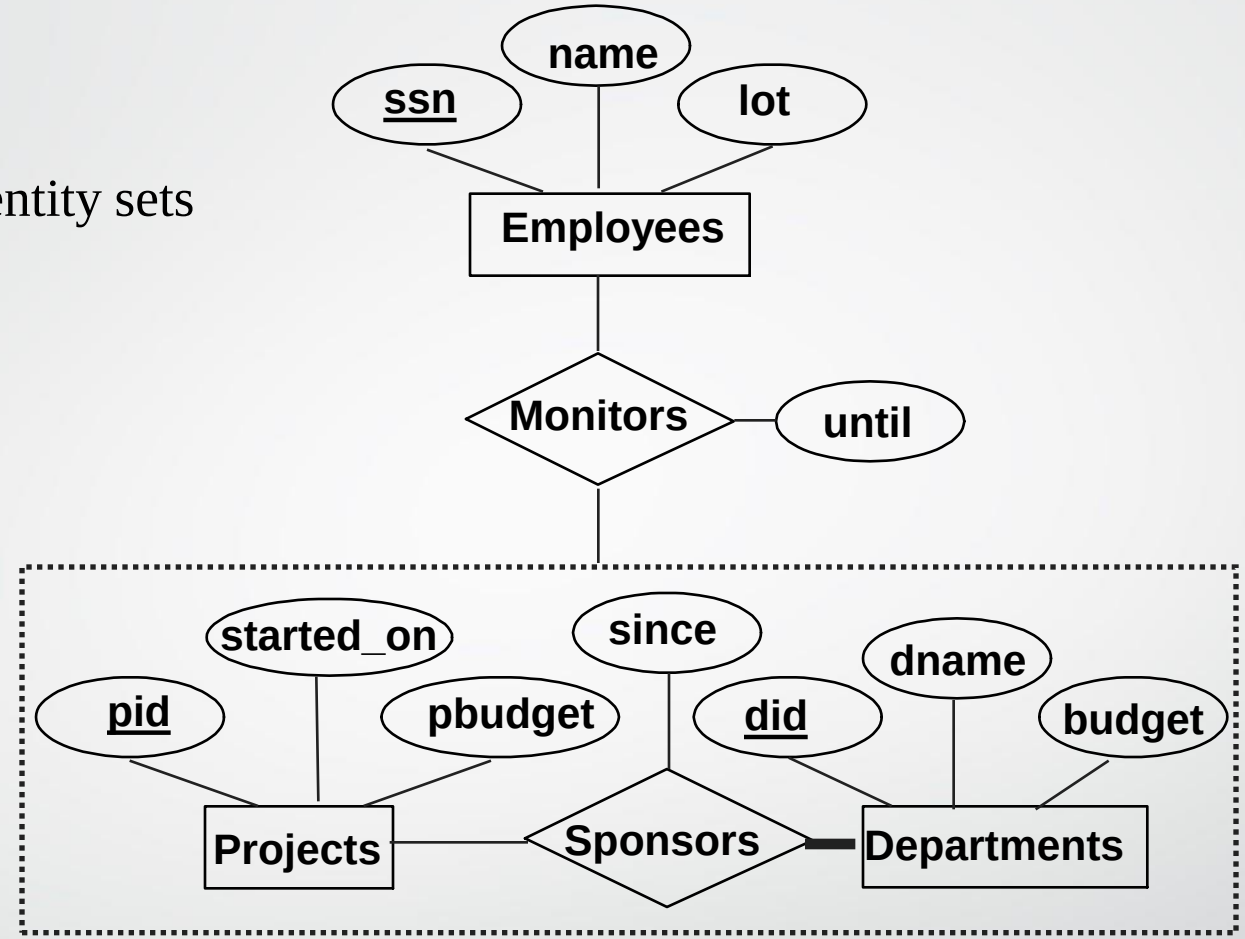
Generalization

- A bottom-up design process – combine a number of entity sets that share the same features into a higher-level entity set.
- Specialization and generalization are simple inversions of each other; they are represented in an E-R diagram in the same way.
- The ISA relationship also referred to as **superclass - subclass** relationship



Aggregation

Used to model a relationship involving (entity sets and) a relationship set.



- *Aggregation* allows us to treat a relationship set as an entity set for purposes of participation in (other) relationships.
- Employees are assigned to monitor SPONSORSHIPS.

 Aggregation vs. ternary relationship:

 Monitors and Sponsors are distinct relationships, with descriptive attributes of their own.

 Also, can say that each sponsorship is monitored by at most one employee (which we cannot do with a ternary relationship).

Making E/R Models

- To make an E/R model you need to identify
  - Enitities
  - Attributes
  - Relationships
  - Cardinality ratios
- from a description

- General guidelines
  - Since entities are things or objects they are often nouns in the description
  - Attributes are facts or properties, and so are often nouns also
  - Verbs often describe relationships between entities

A university consists of a number of departments. Each department offers several courses.

A number of modules make up each course. Students enroll in a particular course and take modules towards the completion of that course. Each module is taught by a lecturer from the appropriate department, and each lecturer tutors a group of students

Example

A university consists of a number of **departments**. Each department offers several **courses**. A number of **modules** make up each course. **Students** enrol in a particular course and take modules towards the completion of that course. Each module is taught by a **lecturer** from the appropriate department, and each lecturer tutors a group of students

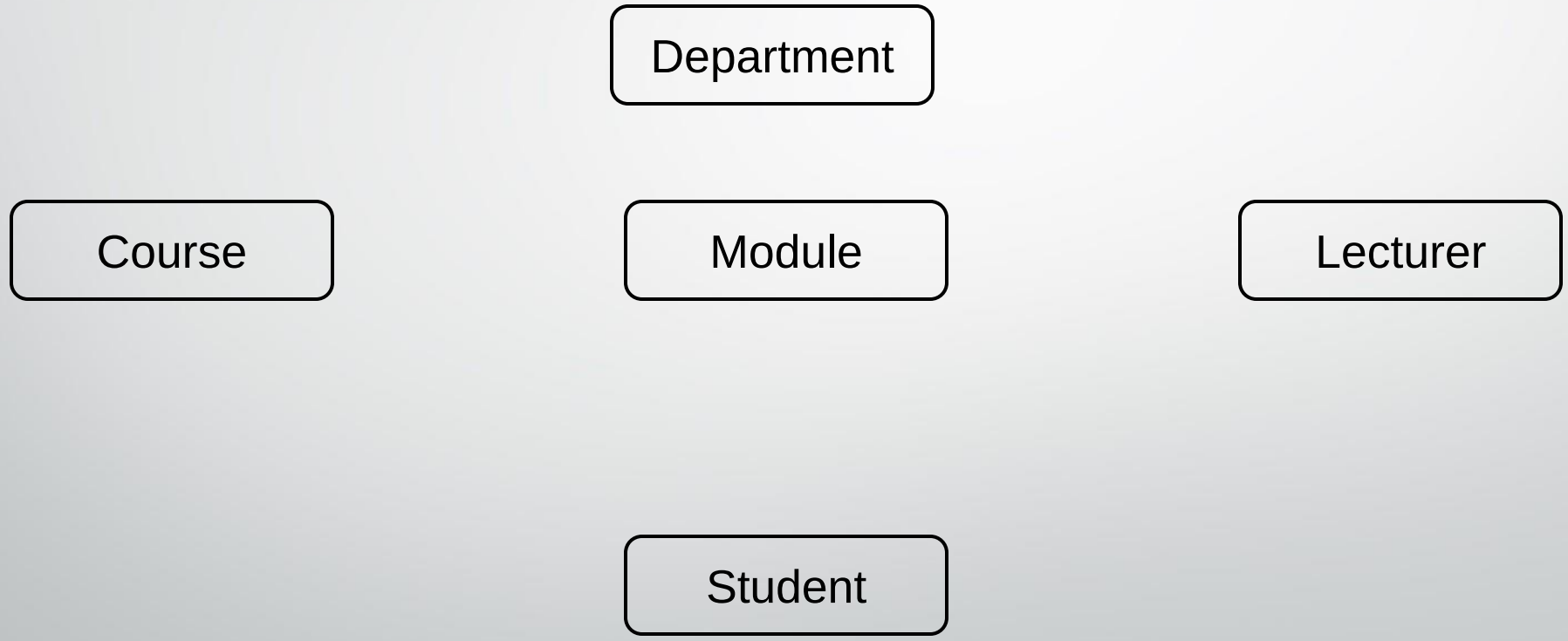
Example - Entities

A university consists of a number of departments. Each department **offers** several courses. A number of modules **make up** each course. Students **enrol in** a particular course and **take** modules towards the completion of that course. Each module is **taught by** a lecturer **from the** appropriate department, and each lecturer **tutors** a group of students

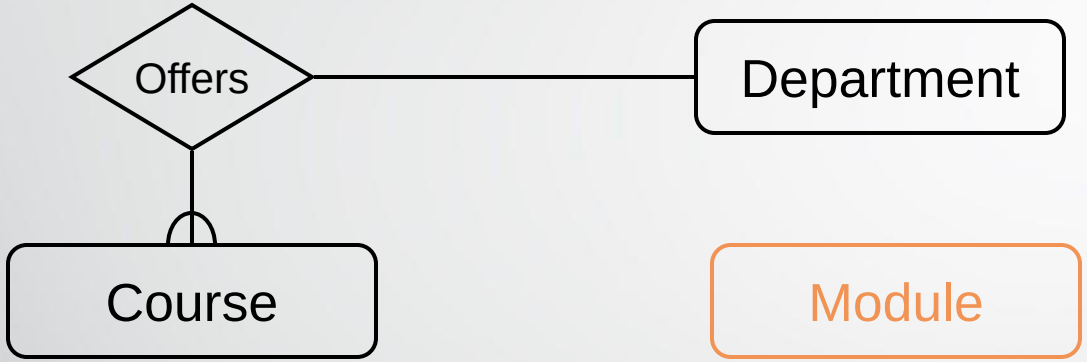
Example - Relationships

Example - E/R Diagram

Entities: Department, Course, Module, Lecturer, Student



Each department offers several courses

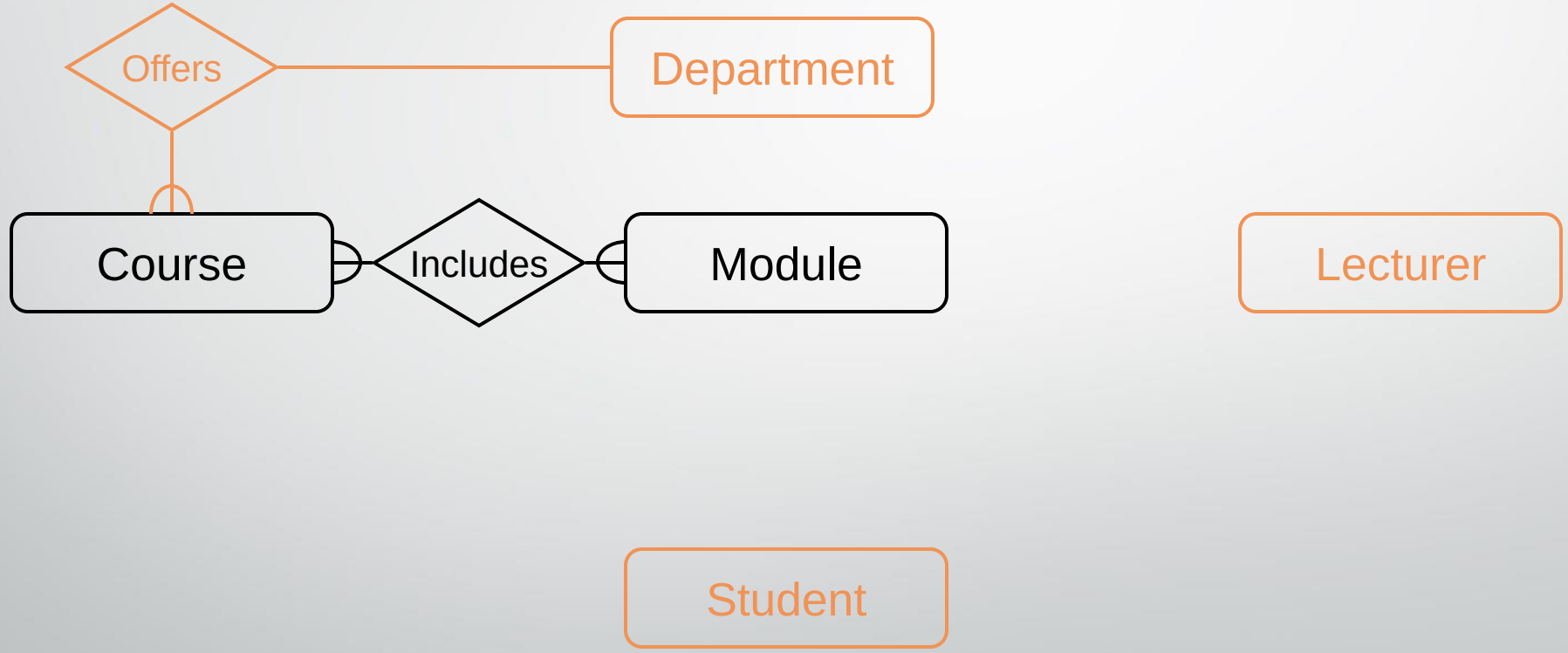


Module

Lecturer

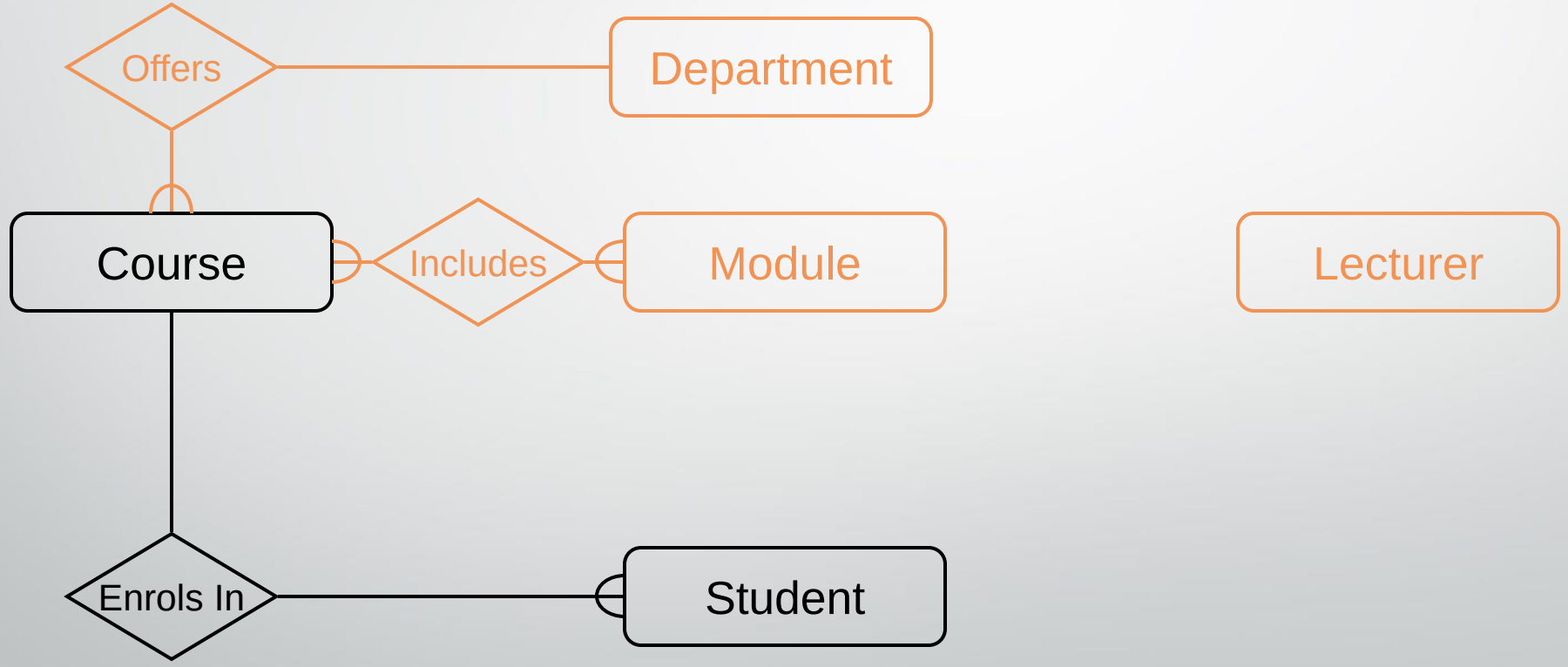
Student

A number of modules make up each courses

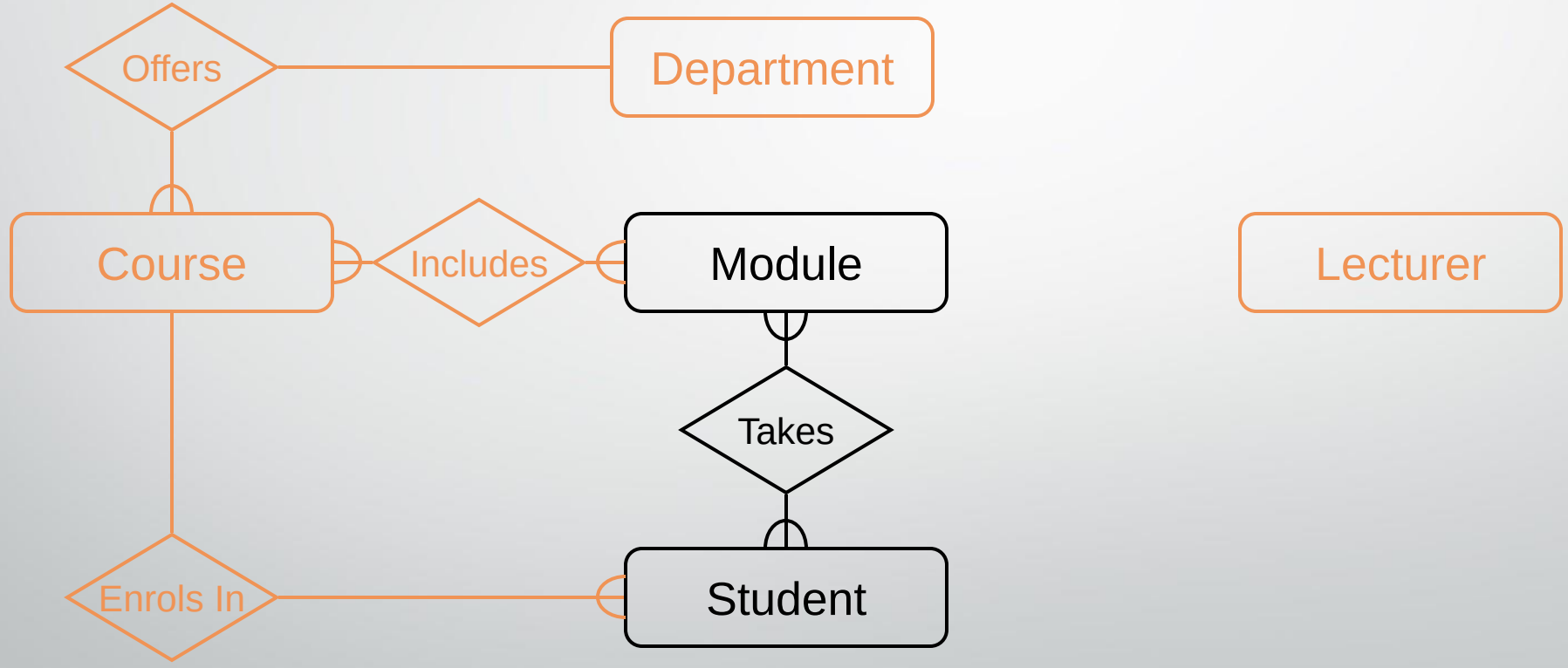


Example - E/R Diagram

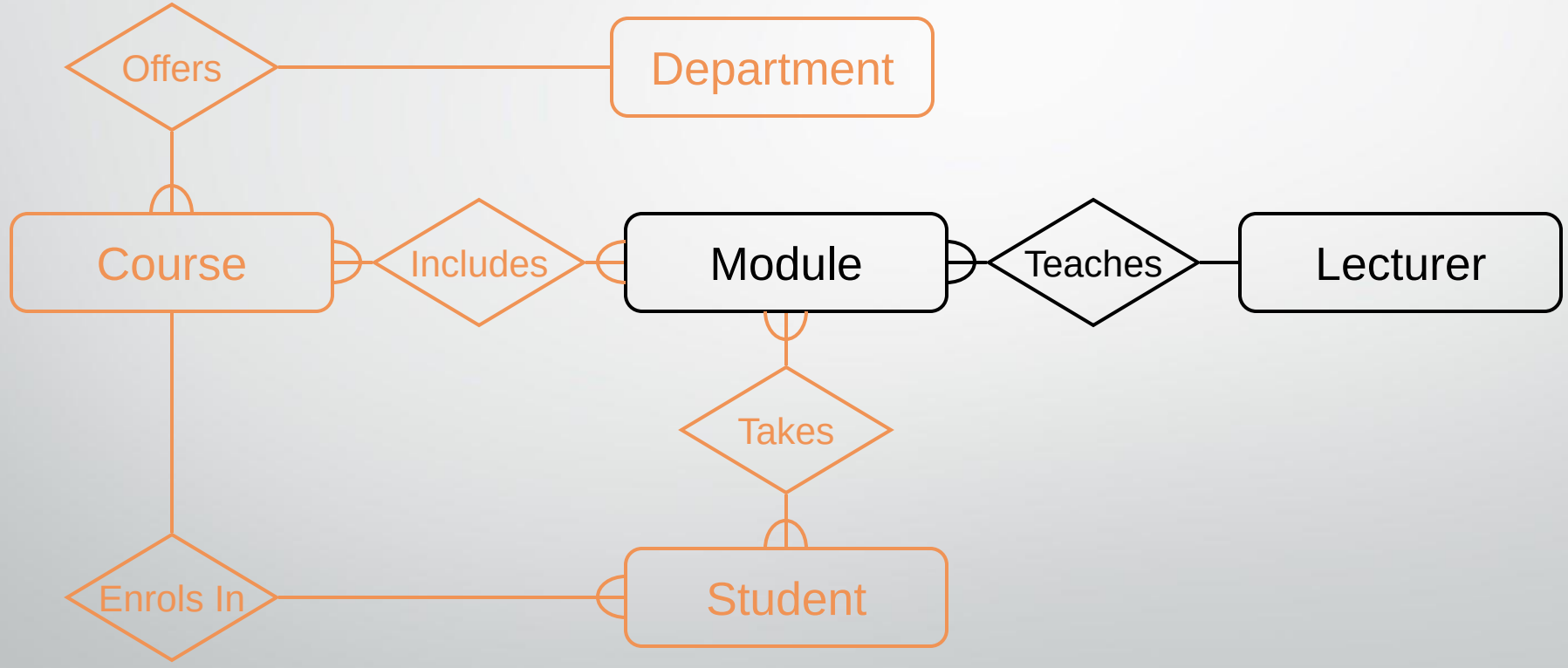
Students enrol in a particular course



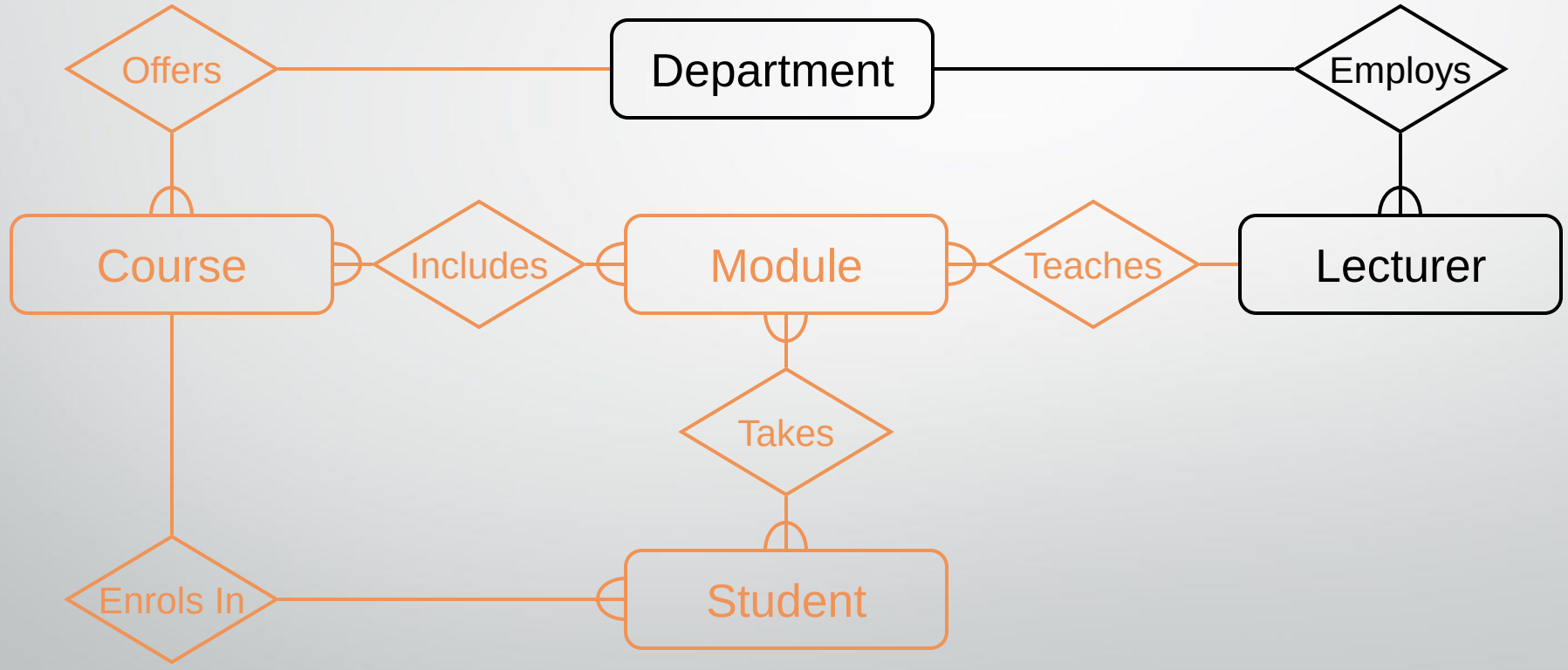
Students … take modules



Each module is taught by a lecturer

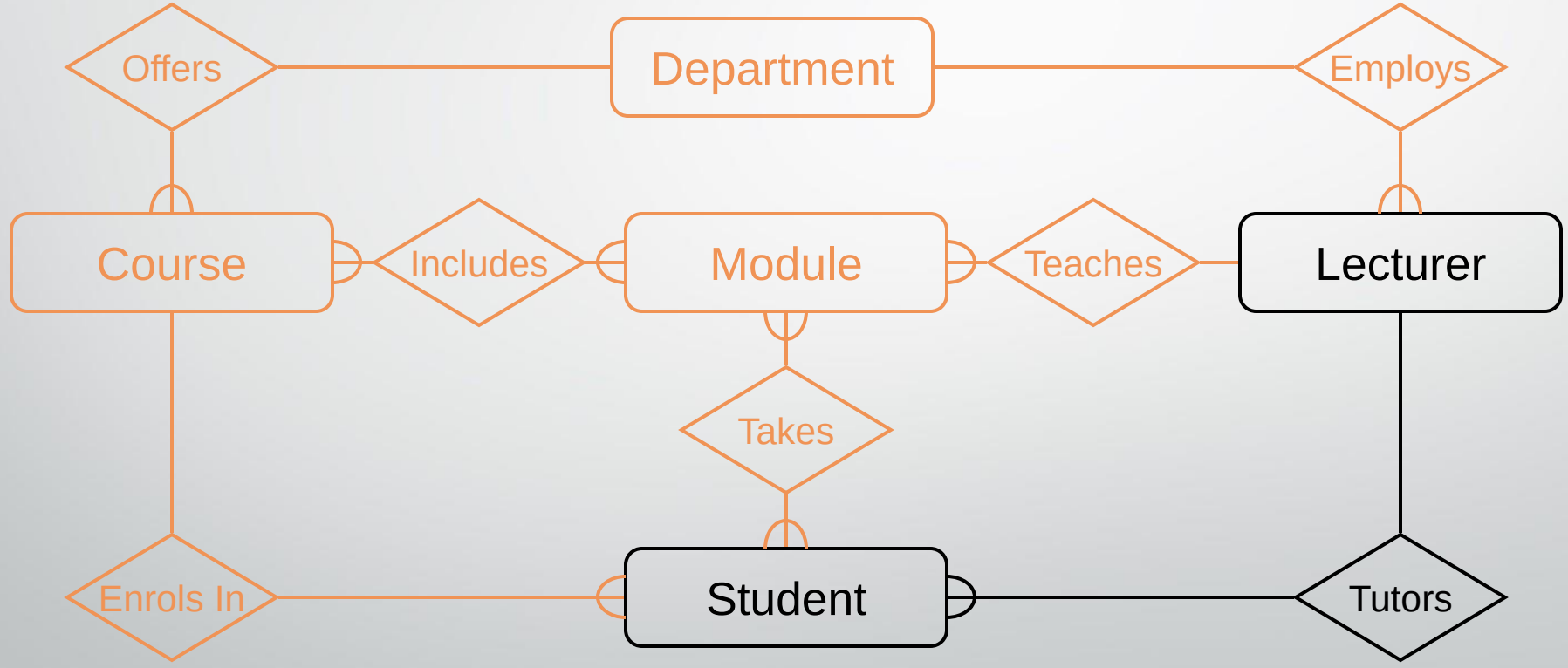


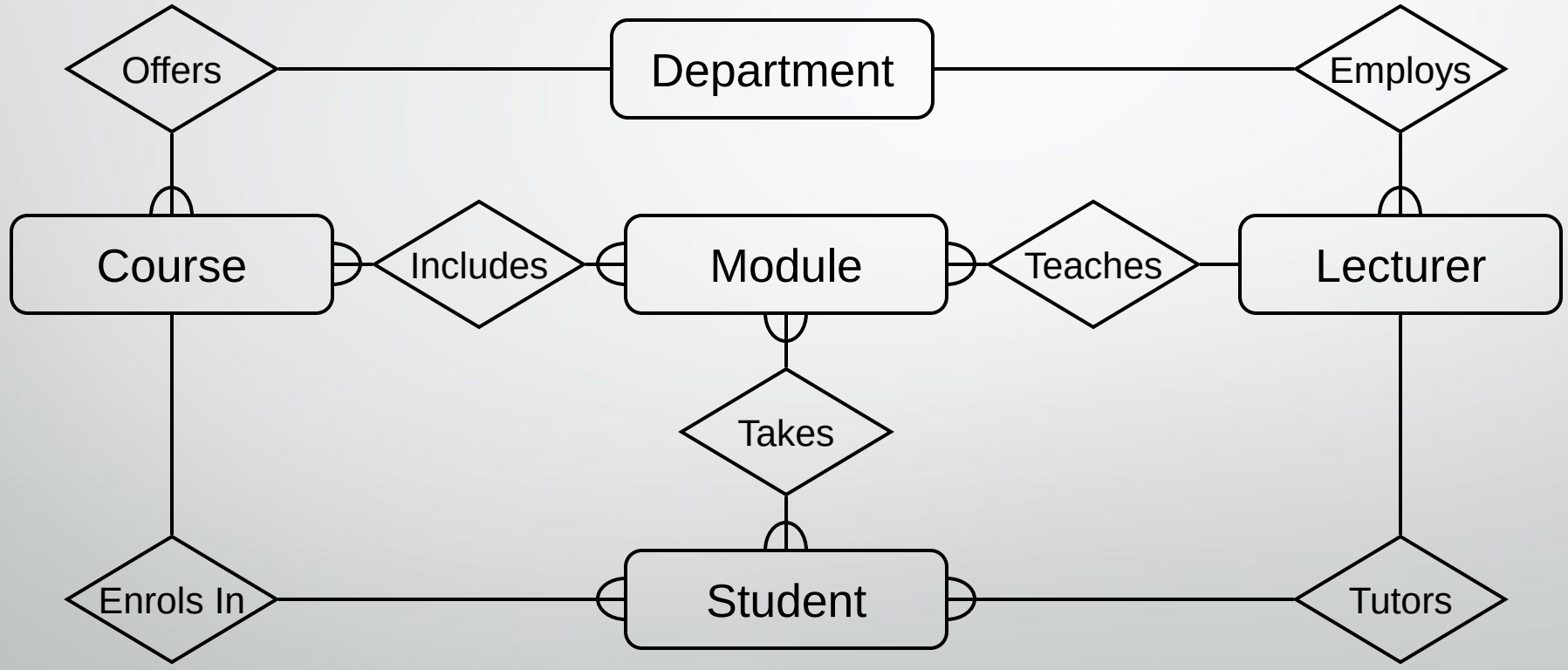
a lecturer from the appropriate department



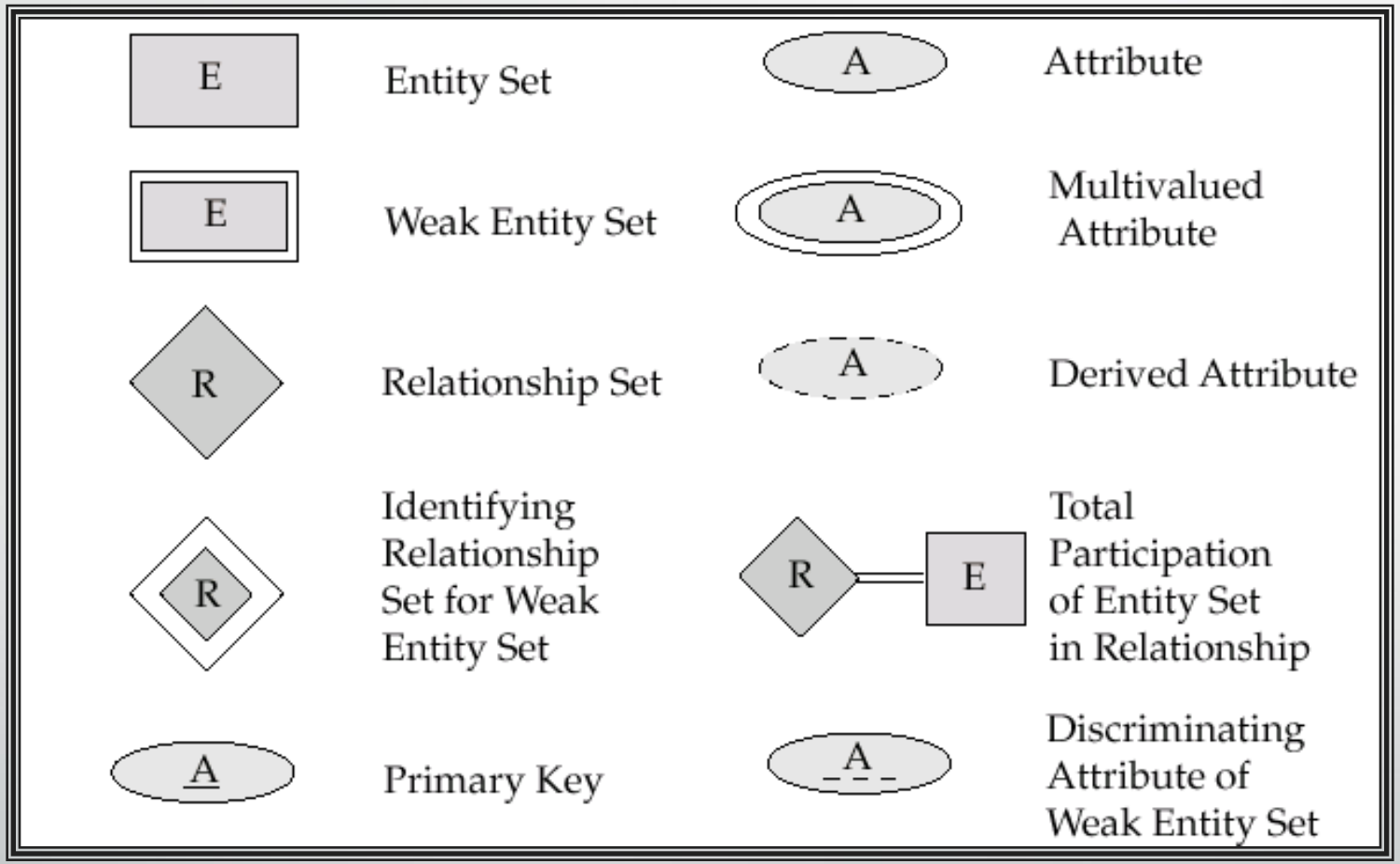
Example - E/R Diagram

each lecturer tutors a group of students

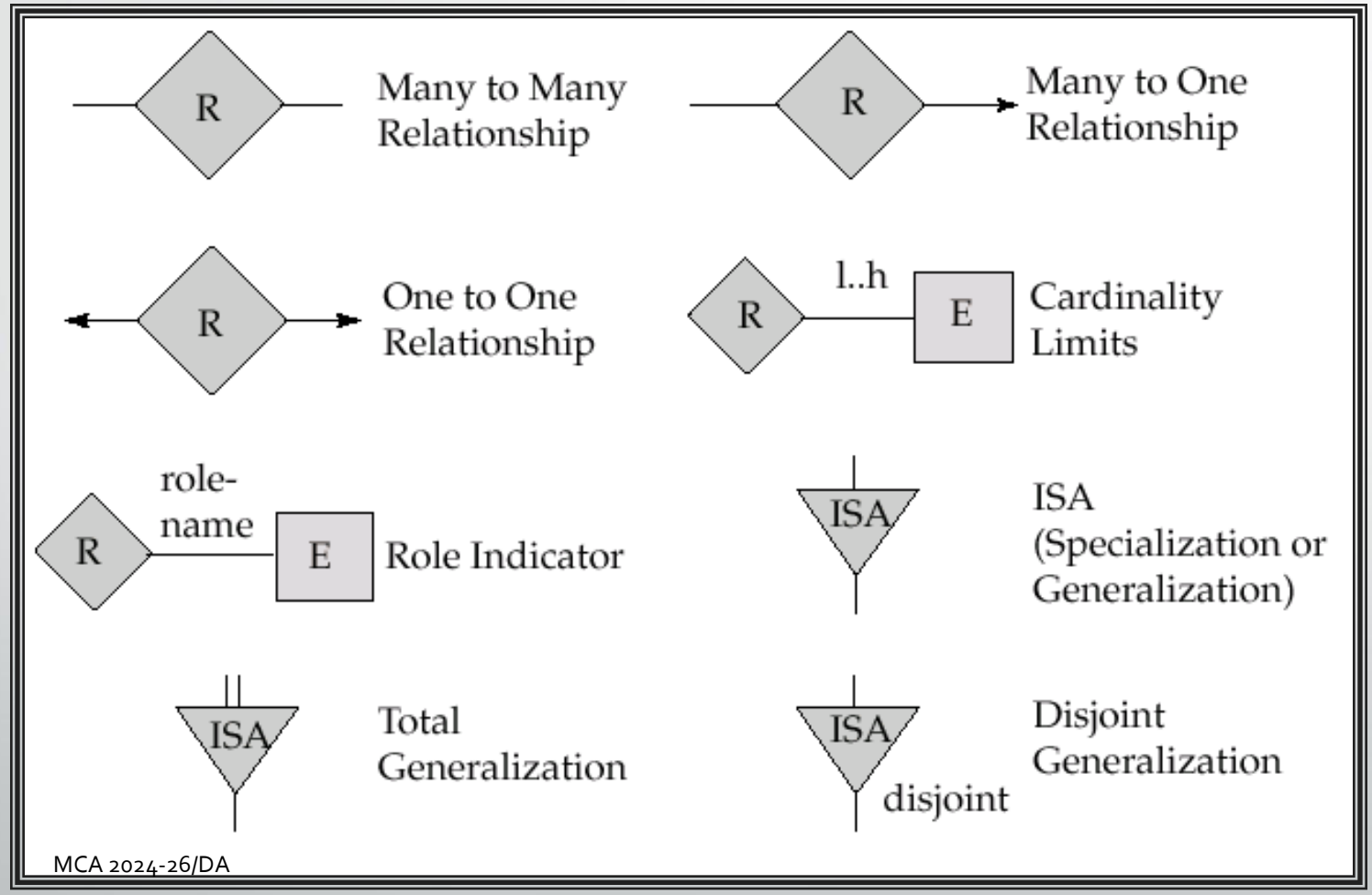



Offers
Department
Employs
Course
Includes
Module
Teaches
Lecturer
Takes
Enrols In
Student
Tutors

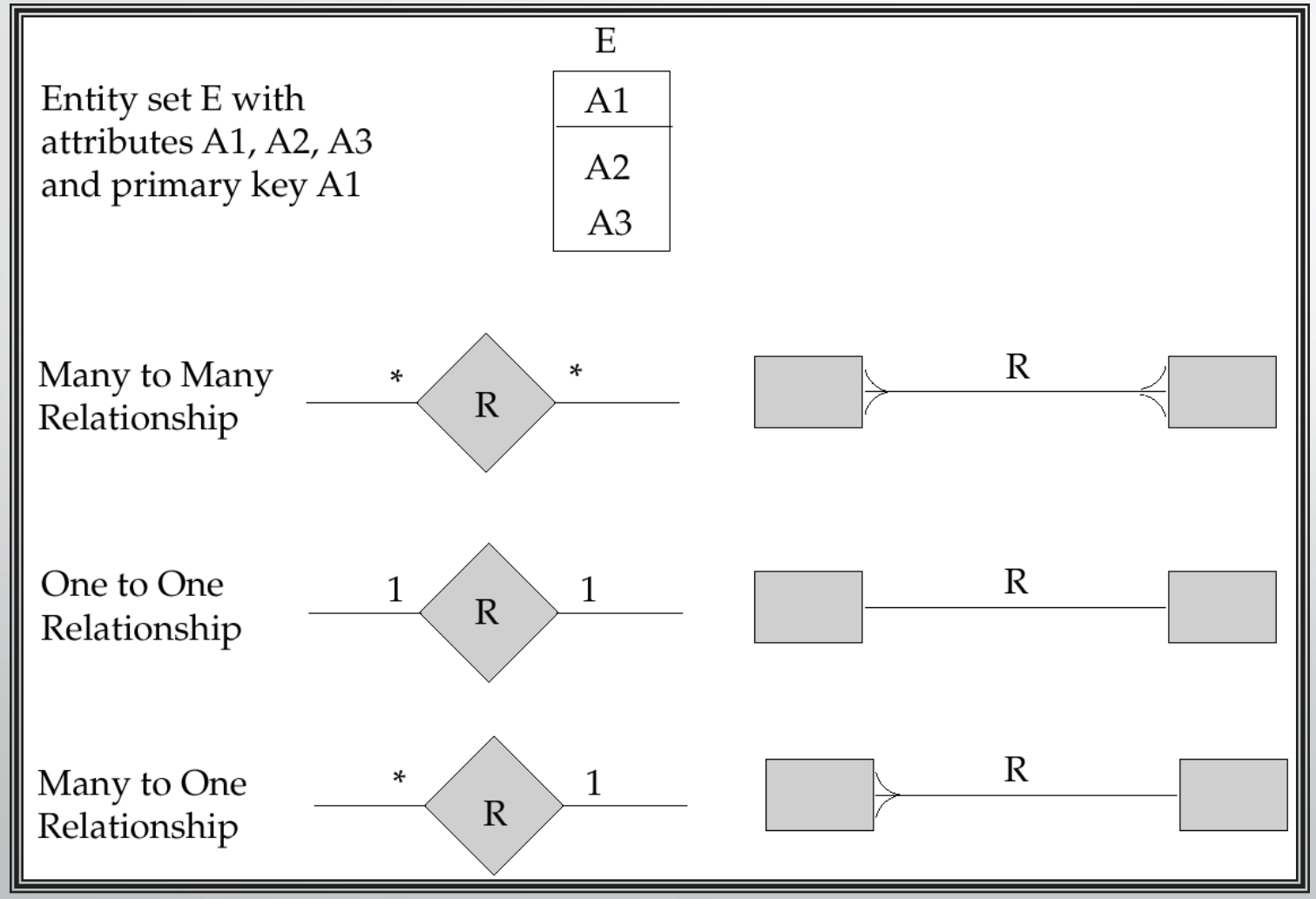
Summary of Symbols



Summary of Symbols (Cont.)



Alternative E-R Notations

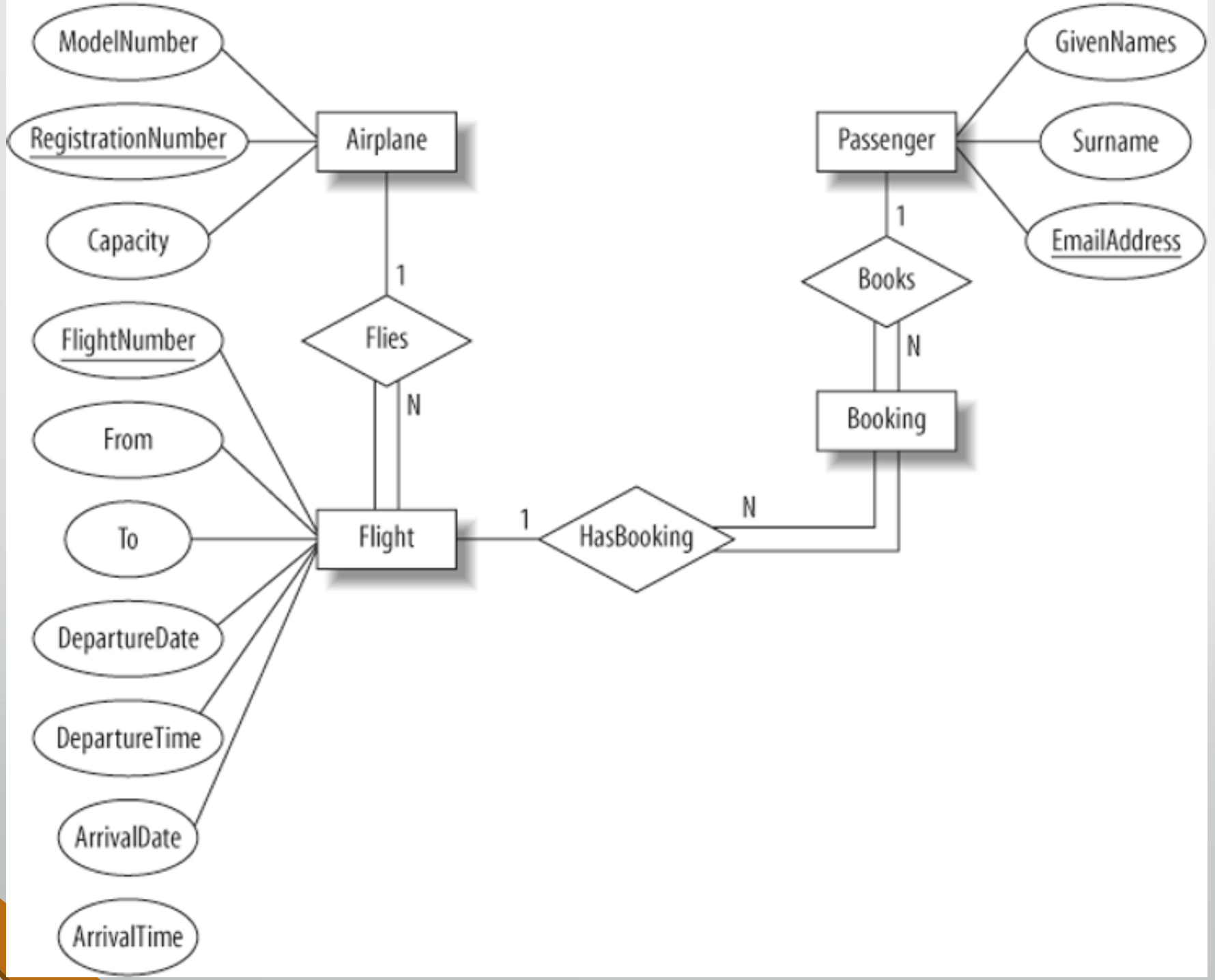


The flight database stores details about an airline's fleet, flights, and seat bookings. It's a hugely simplified version of what a real airline would use, but the principles are the same.

Consider the following requirements list:

i. The airline has one or more airplanes.
ii. An airplane has a model number, a unique registration number, and the capacity to take one or more passengers.
iii. An airplane flight has a unique flight number, a departure airport, a destination airport, a departure date and time, and an arrival date and time.
iv. Each flight is carried out by a single airplane.
v. A passenger has given names, a surname, and a unique email address.
vi. A passenger can book one or more seats on a flight.

Design an **ER diagram** to capture the above requirements. Make sure cardinalities and primary keys are clear**.**


ModelNumber
RegistrationNumber
Capacity
Airplane
1
Flies
N
FlightNumber
From
To
DepartureDate
DepartureTime
ArrivalDate
ArrivalTime
Flight
1
HasBooking
N
Booking
N
Books
1
Passenger
GivenNames
Surname
EmailAddress

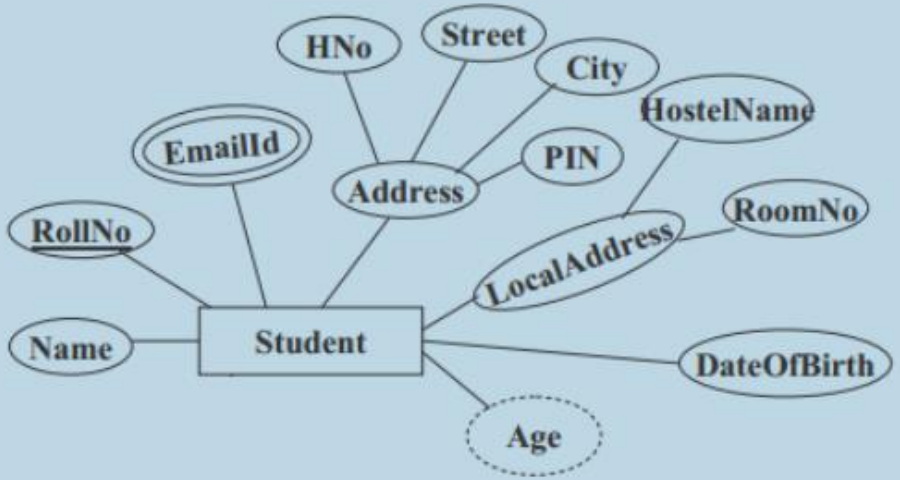
Design an ER diagram

In an educational institute, there are several departments and students belong to one of them. Each department has a unique department number, a name, a location, phone number and is headed by a professor. Professors have a unique employee Id, name, phone number. We like to keep track of the following details regarding students: name, unique roll number, phone number, date of birth, age and one or more email addresses. Students have a local address consisting of the hostel name and the room number.

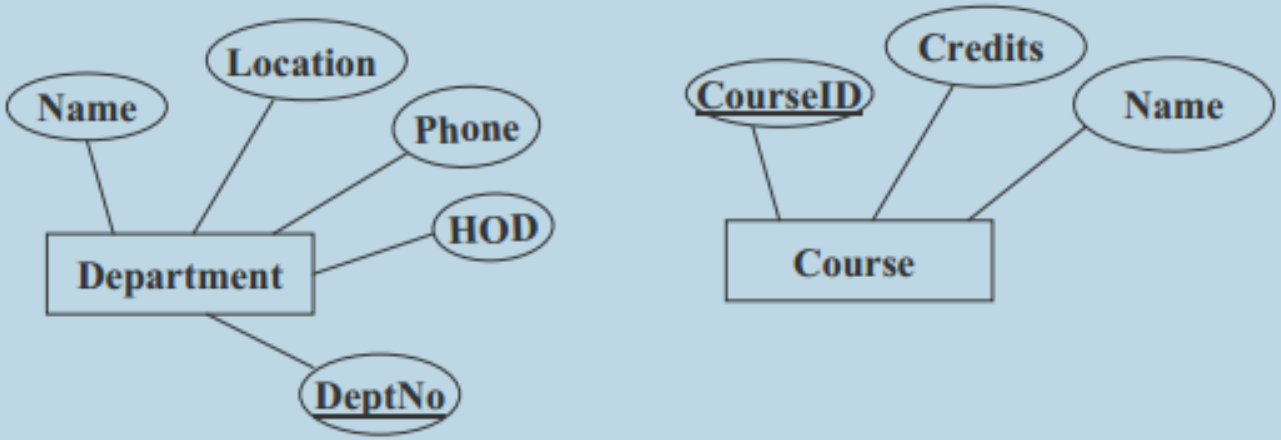
They also have home address consisting of house number, street, city and PIN. It is assumed that all students reside in the hostels.

A course taught in a semester of the year is called a section. There can be several sections of the same course in a semester; these are identified by the section number. Each section is taught by a different professor and has its own timings and a room to meet. Students enroll for several sections in a semester. Each course has a name, number of credits and the department that offers it. A course may have other courses as pre-requisites i.e, courses to be completed before it can be enrolled in. Professors also undertake research projects. These are sponsored by funding agencies and have a specific start date, end date and amount of money given. More than one professor can be involved in a project. Also a professor may be simultaneously working on several projects. A project has a unique projectId.

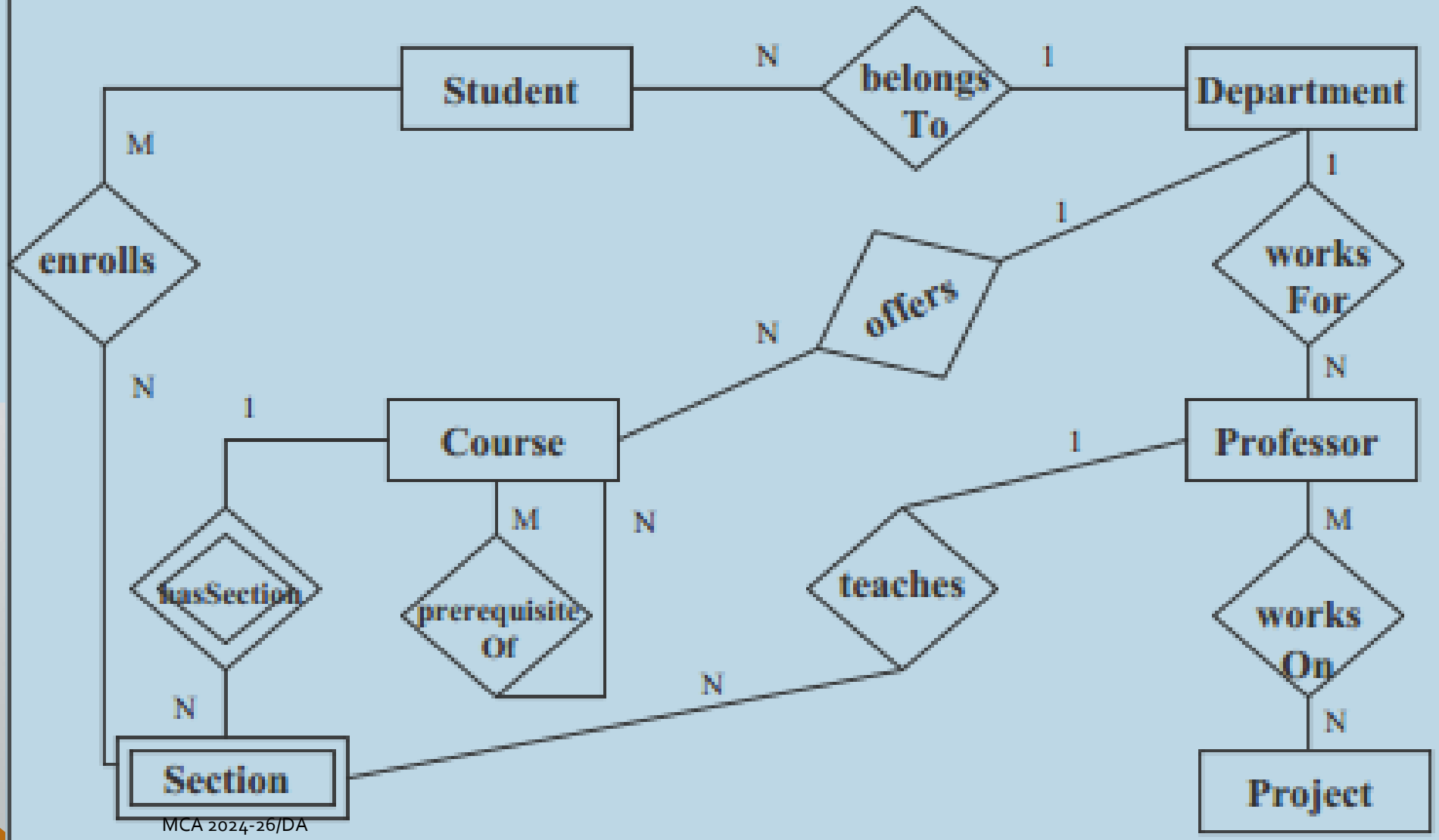
Entities - Student



Entities – Department and Course



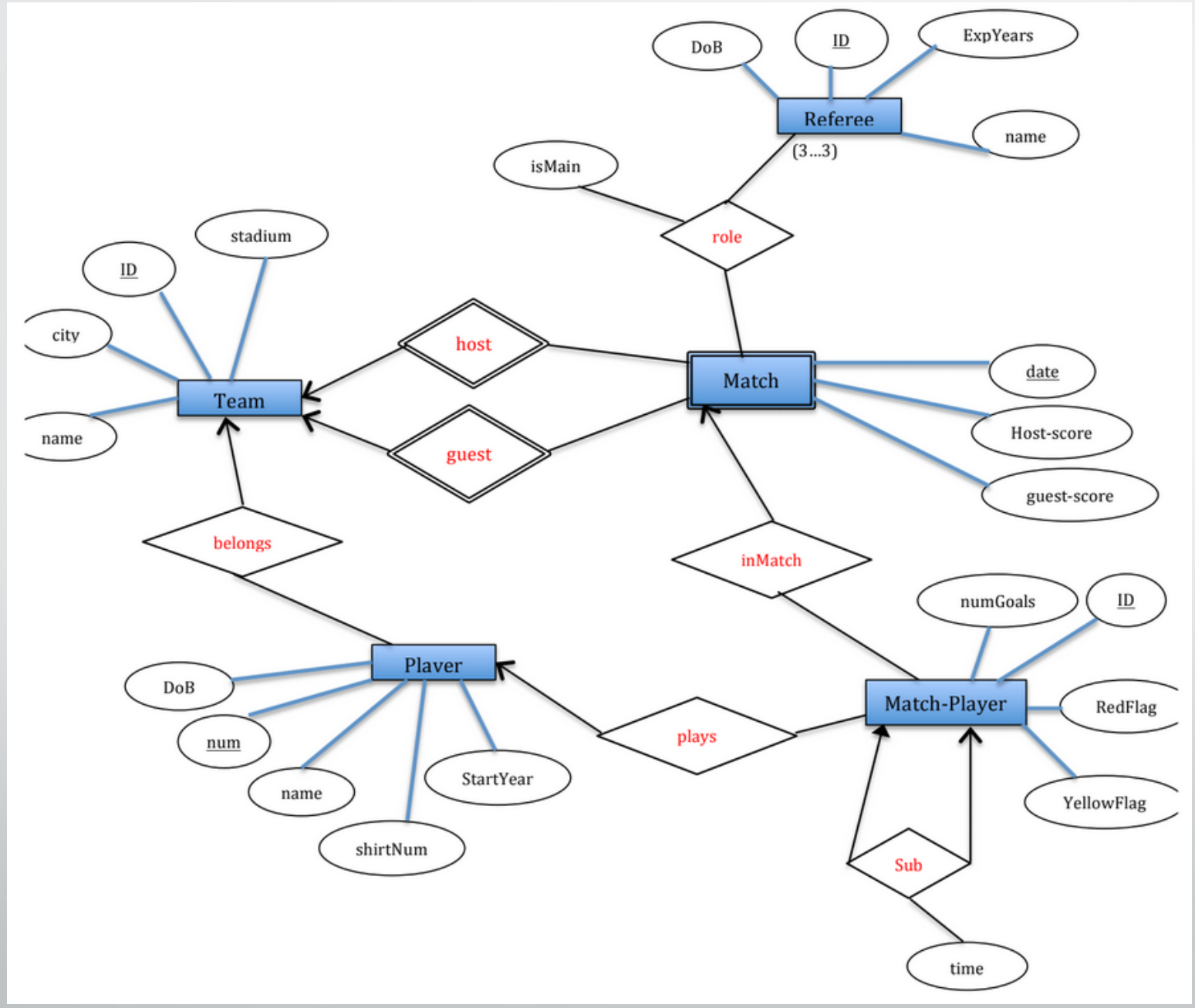
E/R Diagram showing relationships



Assume we have the following application that models soccer teams, the games they play, and the players in each team. In the design, we want to capture the following:

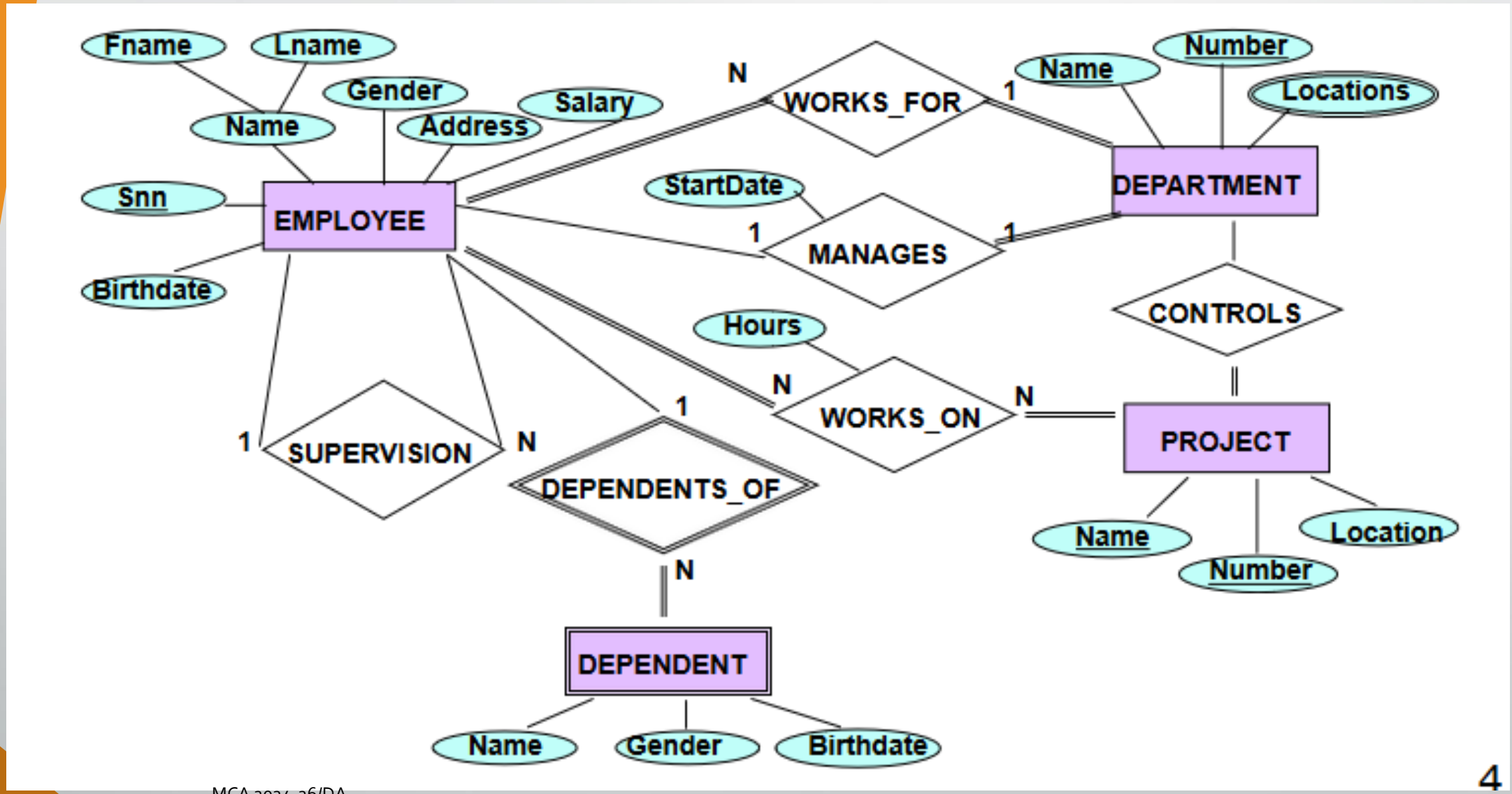
- We have a set of teams, each team has an ID (unique identifier), name, main stadium, and to which city this team belongs.
- Each team has many players, and each player belongs to one team. Each player has a number (unique identifier), name, DoB, start year, and shirt number that he uses.
- Teams play matches, in each match there is a host team and a guest team. The match takes place in the stadium of the host team.
- For each match we need to keep track of the following:
    - o The date on which the game is played
    - o The final result of the match
    - o The players participated in the match. For each player, how many goals he scored, whether or not he took yellow card, and whether or not he took red card.
    - o During the match, one player may substitute another player. We want to capture this substitution and the time at which it took place.
- Each match has exactly three referees. For each referee we have an ID (unique identifier), name, DoB, years of experience. One referee is the main referee and the other two are assistant referee.

Design an ER diagram to capture the above requirements. State any assumptions you have that affects your design (use the back of the page if needed). Make sure cardinalities and primary keys are clear.


DoB
ID
ExpYears
Referee
name
(3...3)
isMain
role
stadium
ID
city
host
Team
name
guest
Match
date
Host-score
guest-score
belongs
inMatch
numGoals
ID
Player
DoB
num
name
StartYear
shirtNum
plays
Match-Player
RedFlag
YellowFlag
Sub
time

Requirements of the Company (oversimplified for illustrative purposes)

- 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.
- Each department *controls* a number of PROJECTs. Each project has a name, 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 employee.

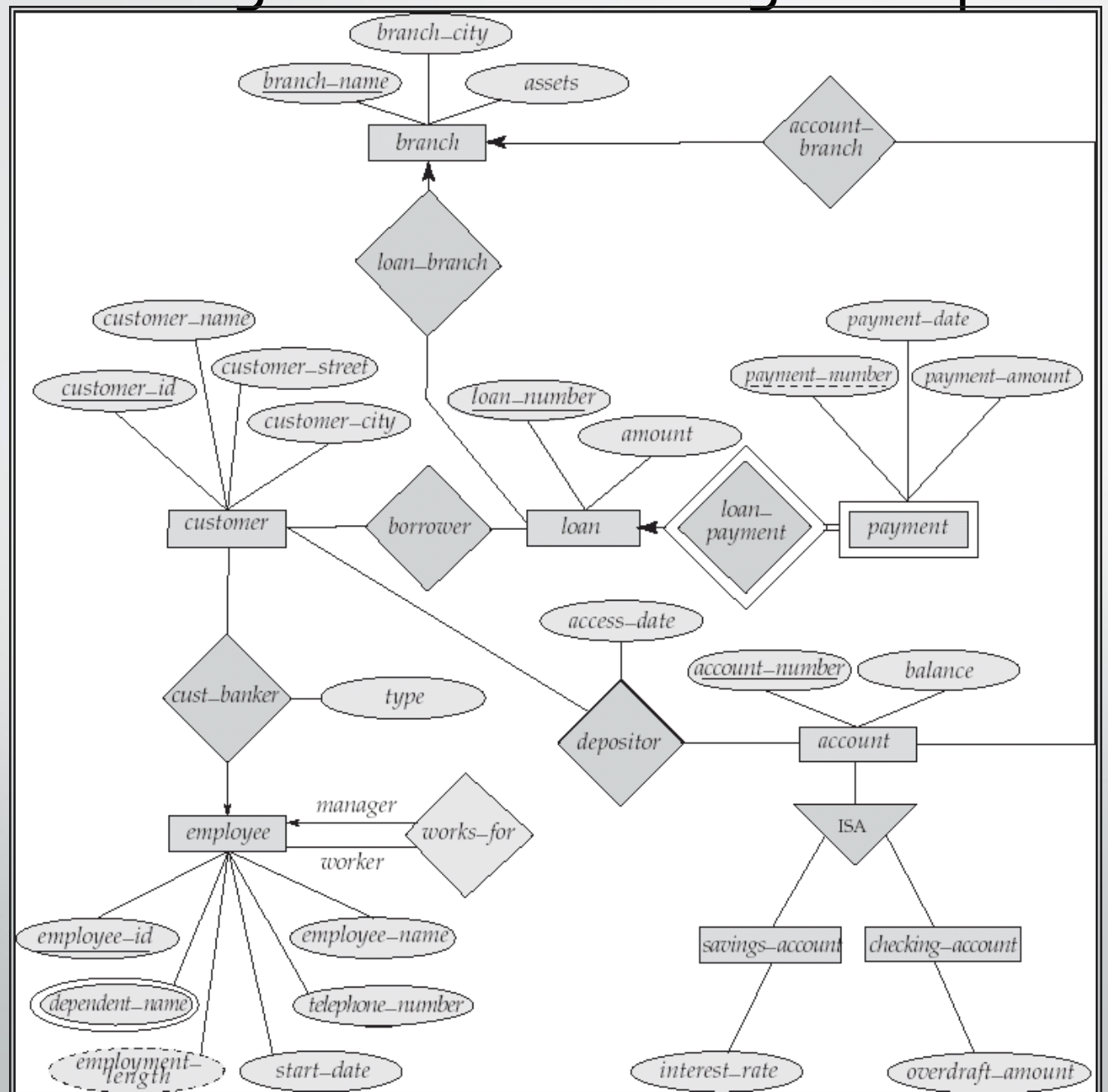

Fname
Lname
Gender
Salary
Name
Address
Snn
EMPLOYEE
Birthdate
N
WORKS_FOR
1
Name
Number
Locations
DEPARTMENT
StartDate
1
MANAGES
1
CONTROLS
Hours
N
WORKS_ON
N
PROJECT
1
SUPERVISION
N
1
DEPENDENTS_OF
N
Name
Location
Number
DEPENDENT
Name
Gender
Birthdate

Banking Application

- The bank is organized into branches. Each branch is located in a particular city and is identified by a unique name. The bank monitors the assets of each branch.
- Bank customers are identified by their customer-id values. The bank stores each customer's name, and the street and city where the customer lives. Customers may have accounts and can take out loans. A customer may be associated with a particular banker, who may act as a loan officer or personal banker for that customer.
- Bank employees are identified by their employee-id values. The bank administration stores the name and telephone number of each employee, the names of the employee's dependents, and the employee-id number of the employee's manager. The bank also keeps track of the employee's start date and, thus, length of employment.

- The bank offers two types of accounts—savings and checking accounts. Accounts can be held by more than one customer, and a customer can have more than one account. Each account is assigned a unique account number. The bank maintains a record of each account's balance, and the most recent date on which the account was accessed by each customer holding the account. In addition, each savings account has an interest rate, and overdrafts are recorded for each checking account.
- A loan originates at a particular branch and can be held by one or more customers. A loan is identified by a unique loan number. For each loan, the bank keeps track of the loan amount and the loan payments. Although a loanpayment number does not uniquely identify a particular payment among those for all the bank's loans, a payment number does identify a particular payment for a specific loan. The date and amount are recorded for each payment.

E-R Diagram for a Banking Enterprise



Medical Clinic ERD

Create an ERD that can be implemented for a medical clinic, using the following business rules:

A patient can make many appointments with one or more doctors in the clinic, and a doctor can accept appointments with many patients. However, each appointment is made with only one doctor and one patient. Emergency cases do not require an appointment. However, for appointment management purposes, an emergency is entered in the appointment book as "unscheduled." If kept, an appointment yields a visit with the doctor specified in the appointment. The visit yields a diagnosis and, when appropriate, treatment. With each visit, the patient's records are updated to provide a medical history. Each patient visit creates a bill. Each patient visit is billed by one doctor, and each doctor can bill many patients. Each bill must be paid. However, a bill may be paid in many installments, and a payment may cover more than one bill. A patient may pay the bill directly, or the bill may be the basis for a claim submitted to an insurance company. If the bill is paid by an insurance company, the deductible is submitted to the patient for payment

EXERCISE

ArtBase, that builds a product for art galleries. The core of this product is a database with a schema that captures all the information that galleries need to maintain. Galleries keep information about artists, their names (which are unique), birthplaces, age,and style of art. For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, lithograph, sculpture, photograph), and its price must be stored. Pieces of artwork are also classified into groups of various kinds, for example, portraits, still lifes, works by Picasso, or works of the 19th century; a given piece may belong to more than one group. Each group is identified by a name (like those just given) that describes the group. Finally, galleries keep information about customers. For each customer, galleries keep that person's unique name, address, total amount of dollars spent in the gallery (very important!), and the artists and groups of art that the customer tends to like. Draw the ER diagram for the database.