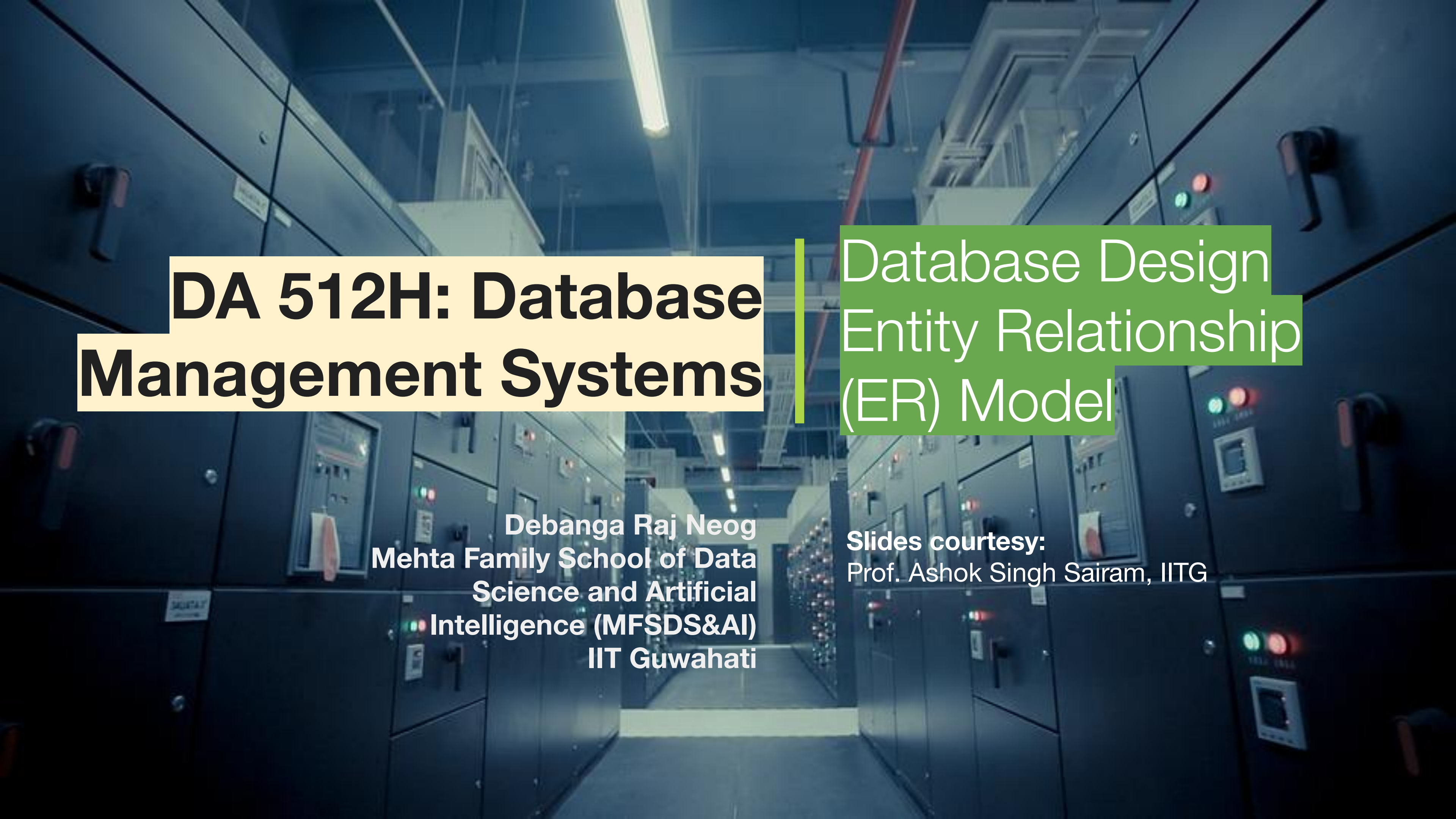


DA 512H: Database Management Systems

Database Design
Entity Relationship (ER) Model

Debanga Raj Neog
Mehta Family School of Data Science and Artificial Intelligence (MFSDS&AI)
IIT Guwahati

Slides courtesy:
Prof. Ashok Singh Sairam, IITG

Steps in Database Design

1. **Requirement Analysis:** understand what data is to be stored

Steps in Database Design

1. **Requirement Analysis**: understand what data is to be stored
2. **Conceptual Database Design**: high level description of the data to be stored
    - Entity Relationship (ER) Model
    - Entities, attributes, and relationships

Steps in Database Design

1. **Requirement Analysis**: understand what data is to be stored
2. **Conceptual Database Design:** high level description of the data to be stored – *Entity Relationship (ER) Model*
3. **Logical Database Design:** select a DBMS to implement and convert the conceptual design to a database schema

Steps in Database Design

1. **Requirement Analysis**: understand what data is to be stored
2. **Conceptual Database Design:** high level description of the data to be stored – *Entity Relationship (ER) Model*
3. **Logical Database Design:** select a DBMS to implement and convert the conceptual design to a *database schema*
4. **Schema Refinement**: identify problems and refine
5. **Physical Database design**: understand expected workloads and make provisions accordingly (build indexes etc);
6. **Application and Security Design**

Overview of Database Design

Requirement Analysis:

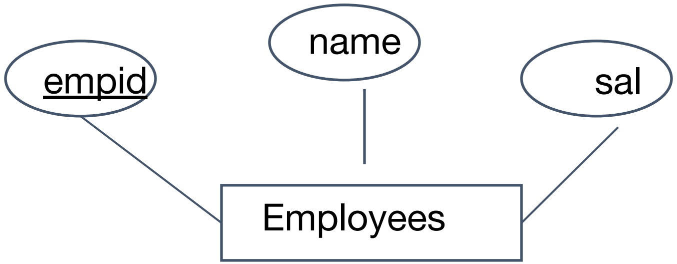
- The very first step in designing a database application is to understand **what data is to be stored** in the database, **what applications must be built** on top of it, and **what operations are most frequent** and subject to performance requirements.
- In other words, we must find out **what the users want from the database**.
- This is usually an informal process that involves discussions with user groups, a study of the current operating environment and how it is expected to change, analysis of any available documentation on existing applications that are expected to be replaced or complemented by the database, and so on.

Overview of Database Design

- **Conceptual design (ER Model):**
  - 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 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 (*in the logical database design step*)

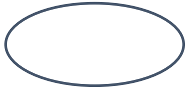
Entity

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

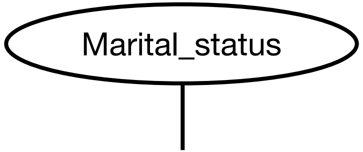


- **Entity Set**: A collection of similar entities. (same attributes)
- Each attribute associated with a **domain** (set of permitted values)
    - For name, the domain can be a the set of 20 character strings

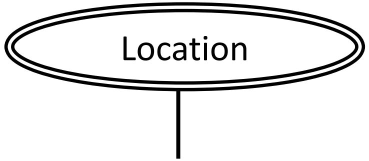
Types of attributes



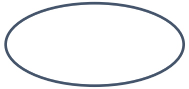
- **Single Valued Attributes:** A single valued or atomic attribute is an attribute that has only one value associated with it. Examples include: customer ID, employee ID, product ID, first name, last name, address, email address.



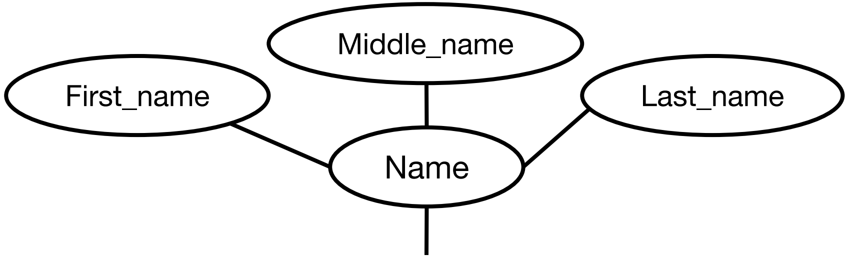
- **Multi Valued Attributes:** A multi valued attribute is an attribute that has more than one value associated with it. Examples include: phone numbers, skills, hobbies, interests.



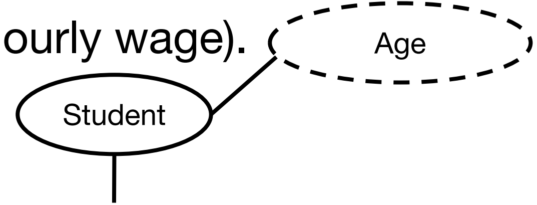
Types of attributes



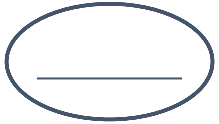
- **Composite Attributes:** A composite attribute is an attribute that is composed of multiple attributes. Examples include: full name (first name + middle name + last name), address (street + city + state + zip code).



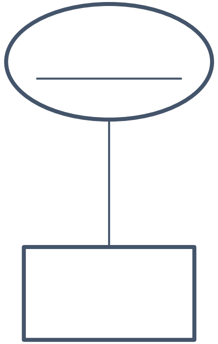
- **Derived Attributes:** A derived attribute is an attribute that is derived from other attributes. Examples include: age (derived from date of birth), salary (derived from hours worked + hourly wage).



Key

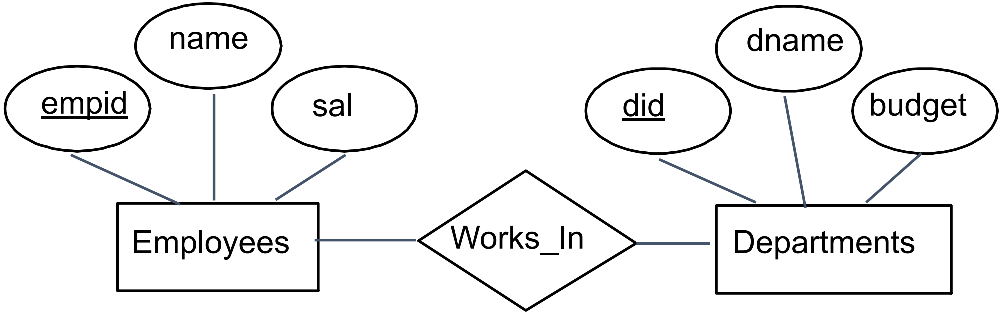


- **Key**: an attribute or a set of attributes that are used to identify a unique instance of an entity.
- A key can be a **primary key**, which is a single attribute, or a **composite key**, which is a combination of multiple attributes.
  - *For example, a composite key made up of StudentID and CourseID together can uniquely identify a course enrollment.*
- Keys are used to establish relationships between entities.
- A **primary key** is a unique identifier, while a **foreign key** references the primary key of another entity. The primary key ensures the uniqueness of entity, while the foreign key helps maintain referential integrity between two entities.



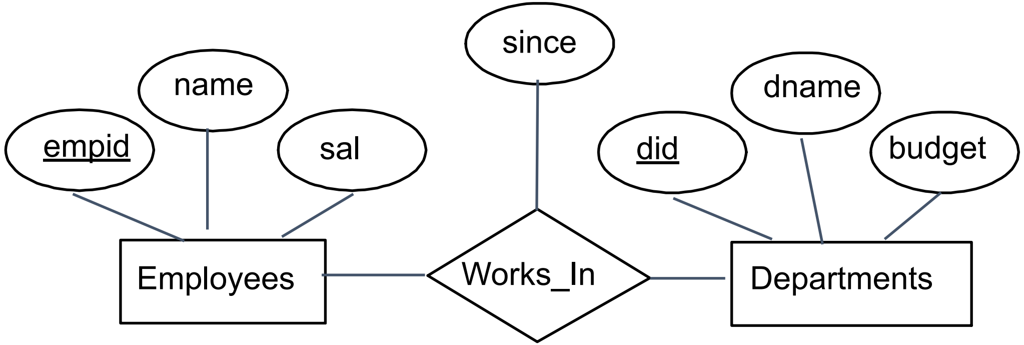
Relationship

- **Relationship**: Association among two or more entities.
  - The relationship is uniquely determined by the keys of its entities
  - E.g., **David works in Design department.**



Relationship

- Relationship can have attributes
  - E.g. **David works in Design department since January 2018**



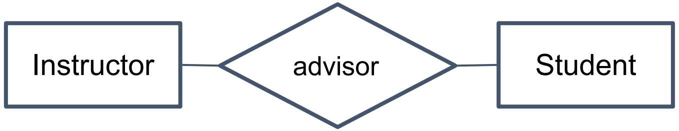
Relationship Set

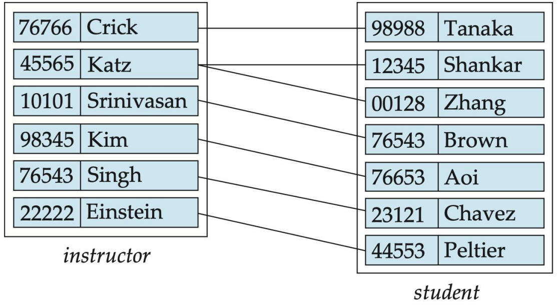
- **Relationship Set:** A relationship set is a **set** of relationships of the same type.

If $E_1, E_2, \ldots, E_n$ are entity sets, then a relationship set *R* is a subset of

$$\{(e_1, e_2, \ldots, e_n) \mid e_1 \in E_1, e_2 \in E_2, \ldots, e_n \in E_n\}$$

Example: Consider the two entity sets ***instructor*** and ***student***. We define the relationship set ***advisor*** to denote the association between instructors and students.





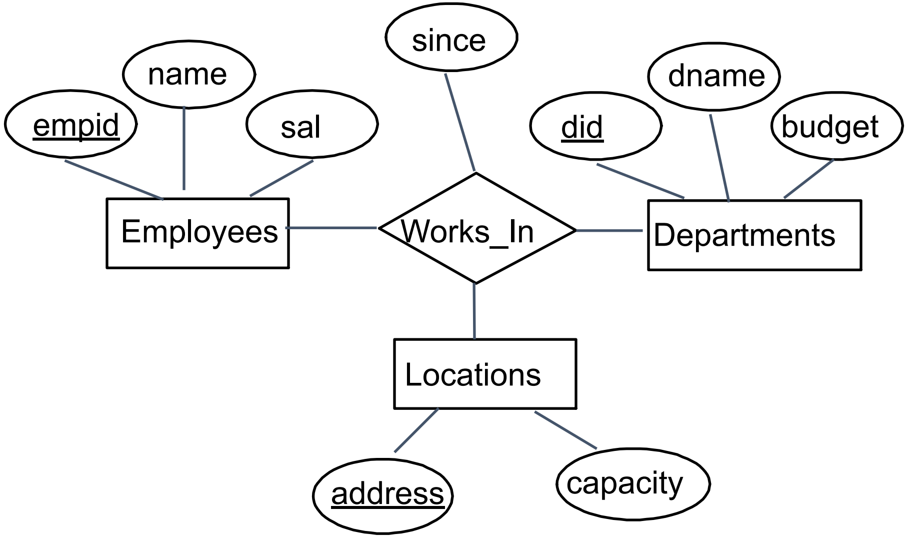
76766	Crick
45565	Katz
10101	Srinivasan
98345	Kim
76543	Singh
22222	Einstein

instructor

98988	Tanaka
12345	Shankar
00128	Zhang
76543	Brown
76653	Aoi
23121	Chavez
44553	Peltier

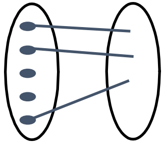
student

More examples of relationships

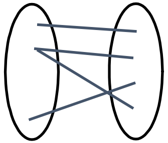


Types of relationship

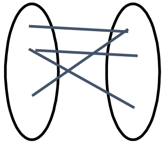
- **One-to-one relationship**(1:1) One instance in the (parent) entity refer to one and only one instance in the related (child) entity
- **One-to-many relationship**(1:N) One instance in the (parent) entity refer to one or more instance in the related (child) entity
- **Many-to-one relationship**(M:1) Many instance in the (parent) entity refer to one instance in the related (child) entity
- **Many-to-many relationship**(M:N) One instance of the first entity (parent) can relate to many instances of the second entity (child), and one instance of the second entity can relate to many instances of the first entity
- **Example**: If, in a particular university, a student can be advised by only one instructor, and an instructor can advise several students, then the relationship set from instructor to student is one-to-many. If a student can be advised by several instructors (as in the case of students advised jointly), the relationship set is many-to-many.



1-to-1



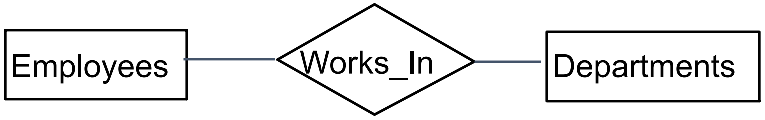
1-to Many

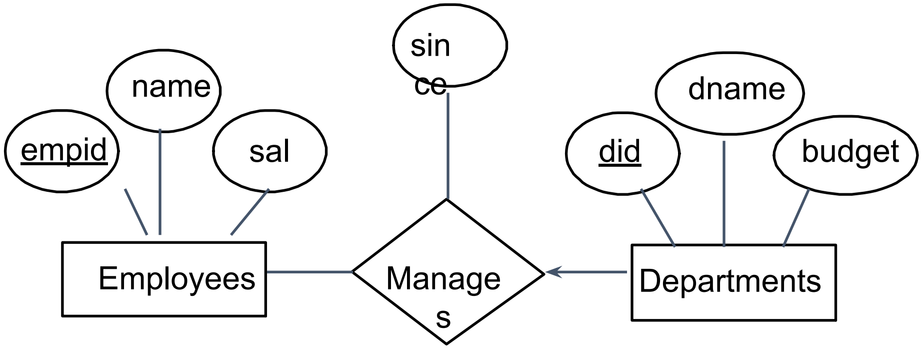


Many-to-Many

Key constraints

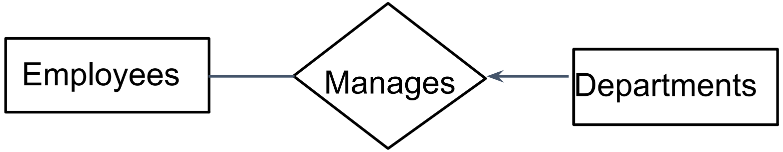
- Consider Works_In: An employee can work in many departments; a department can have many employees. (no constraints)
- Consider another relationship called Manages between Employees and Departments
    - In contrast, each dept has **at most one** manager, an example of key constraint on Manages.



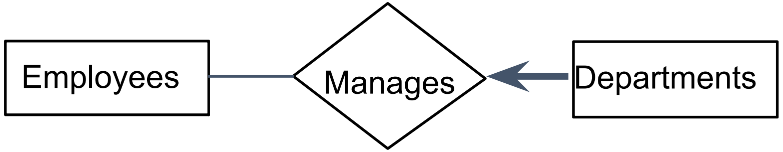


Arrows in relationships

- What does the arrow mean ?
    - Given a department, we can uniquely identify its manager
    - Some departments may have no manager
    - Draw an arrow from the entity set with **key constraint to the relationship**

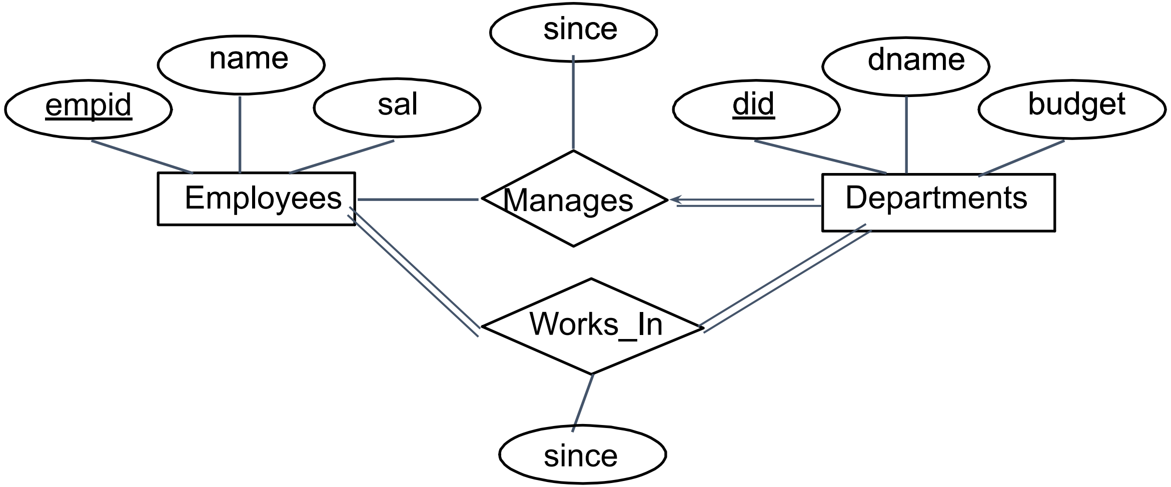


- **Bold arrow**
    - Each department has exactly one manager
    - 1:1 & strong relationship



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 (whereas Employees is partial, since every employee will not be a manager).
    - Every Departments entity must appear in an instance of the Manages relationship.

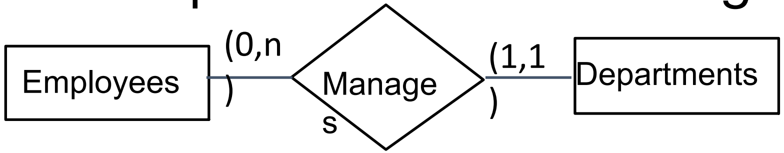


Cardinality constraints

- Each entity set participates in a relationship set with a minimum (min) and a maximum (max) cardinality
- Express cardinality constraints by drawing either a directed line (→) signifying one or an undirected line (——) signifying many between the relationship set and the entity
- Alternative: Structural constraint
  - cardinalities are pairs of non-negative integers (n, N) such that n ≤ N, where N means "any number“
  - Minimum cardinality: 0 (participation optional); 1 (participation mandatory)
  - Maximum cardinality: 1 (each entity instance associated at most with a single instance of the relationship); N (each entity instance associated with many instances of the relationship).

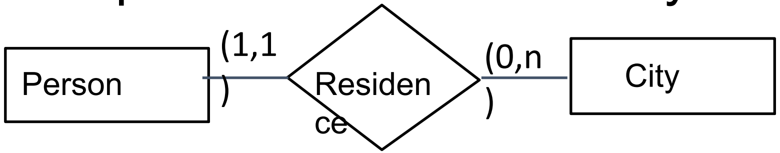
Example: Cardinality Constraints

- Each department has a manager



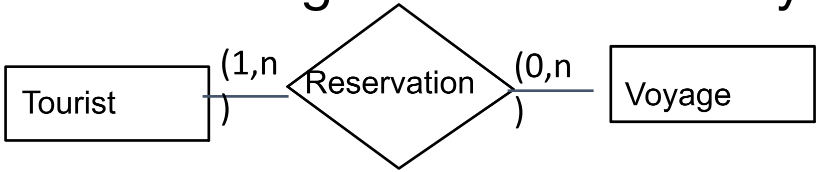
Employees (0,n) Manages (1,1) Departments

- Each person resides in a city

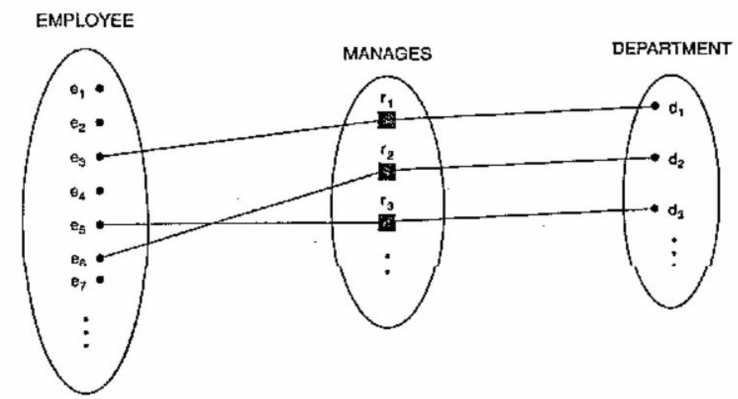


Person (1,1) Residence (0,n) City

- A tourist can go on 1 or more voyages



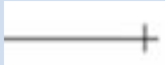
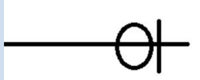
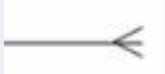
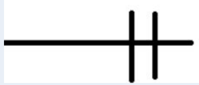
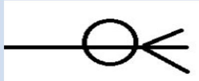
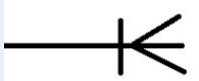
Tourist (1,n) Reservation (0,n) Voyage

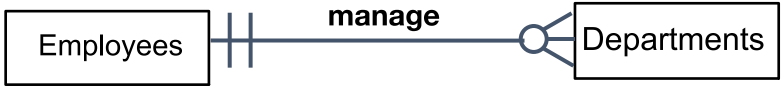


Upper:Lower

Crow's foot notation

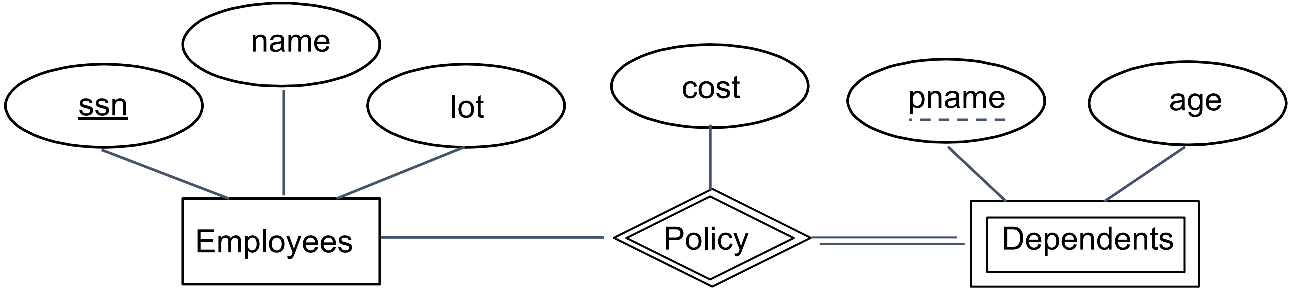
- Crow's feet is a common method of indicating cardinality

Symbol	Meaning	Symbol	Meaning
	One		Zero or one
	Many		One and only one
			Zero or many
			One or many

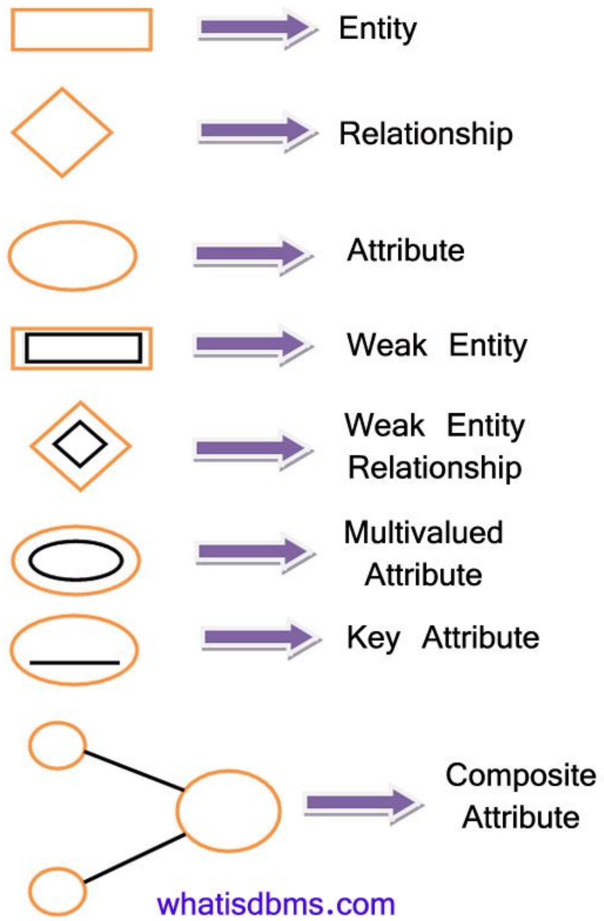


Weak Entities

- A **weak entity** can be identified uniquely only by considering the primary key of another (*owner*) entity.
  - Owner entity set and weak entity set must participate in a one-to-many relationship set (one owner, many weak entities).
  - Weak entity set must have total participation in this **identifying** relationship set.



ER Symbols


Entity
Relationship
Attribute
Weak Entity
Weak Entity Relationship
Multivalued Attribute
Key Attribute
Composite Attribute
whatisdbms.com

Problem

Draw the ER model of university database application considering the constraints:

- A university has many departments.
- Each department has multiple instructors (one person is HOD). Here the HOD refers to the head of department.
- An instructor belongs to only one department.
- Each department offers multiple courses, each subject is taught by a single instructor.
- A student may enroll for many courses offered by different departments.

A Solution

Entities:

- Department
- Course
- Student
- Instructor

Attributes:

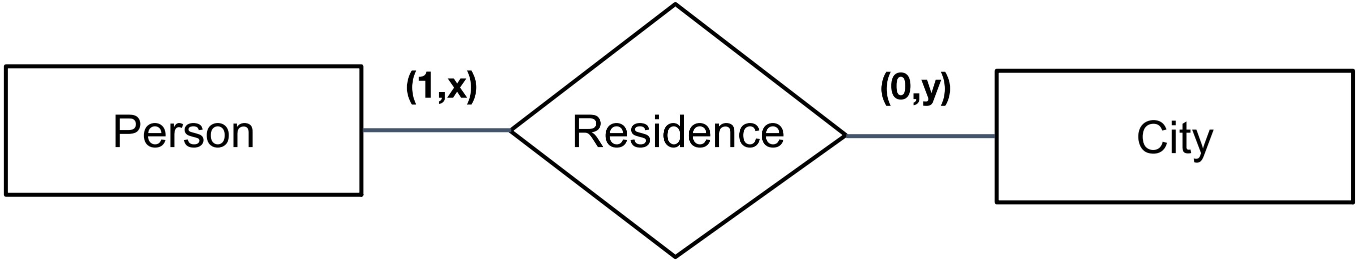
- Department − the relevant attributes are department Name and location.
- Course − The relevant attributes are courseNo, course Name, Duration, and prerequisite.
- Instructor − The relevant attributes are Instructor Name, Room No, and telephone number.
- Student − The relevant attributes are Student No, Student Name, and date of birth.

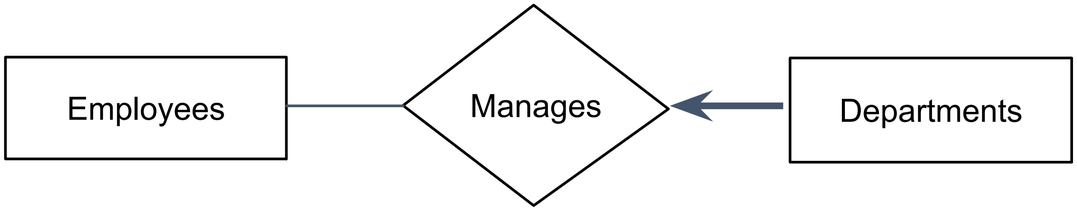
Key Attributes:

- Department Name is the key attribute for Department.
- CourseNo is the key attribute for Course entity.
- Instructor Name is the key attribute for the Instructor entity.
- StudentNo is the key attribute for Student entities.

Relationships:

- Department offers course (1:N)
- Course enrolled by student (M:N)
- Department has instructor (1:N)
- Department is headed by instructor (1:1)
- Course is taught by instructor (N:1)


Person
(1,x)
Residence
(0,y)
City


Employees
Manages
Departments

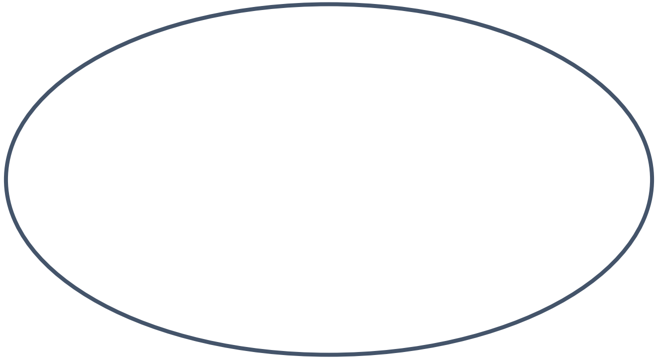
Reviewing

- **After Requirement Analysis**
  - **Conceptual design (ER Model):**
    - 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 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 (*in the logical database design step*)

Reviewing

Reviewing

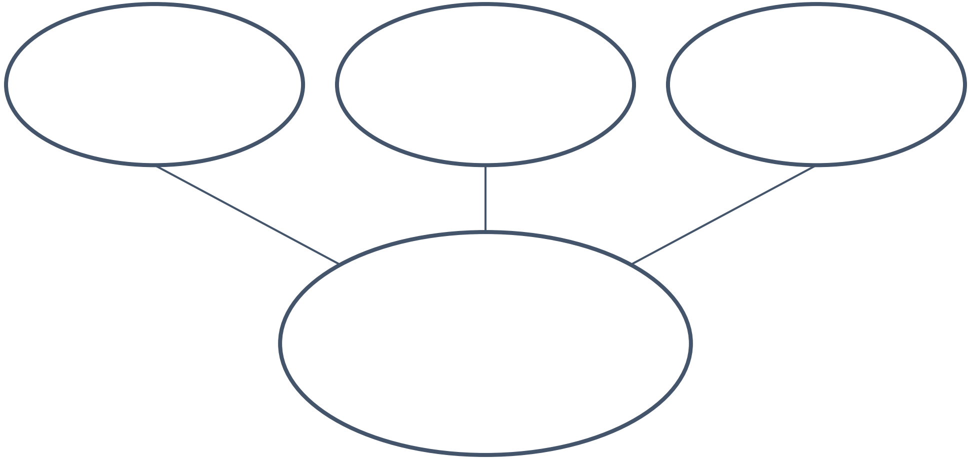
Reviewing



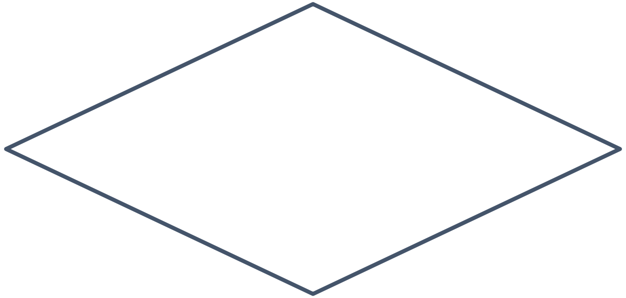
Reviewing

Reviewing

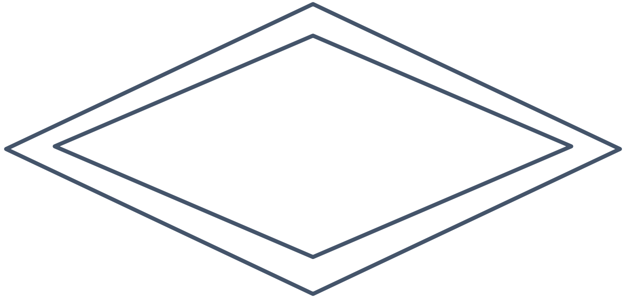
Reviewing



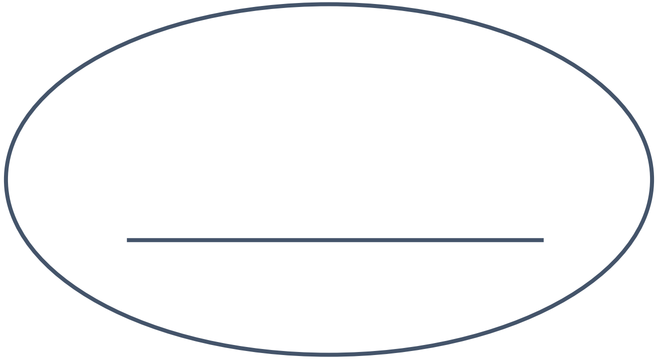
Reviewing



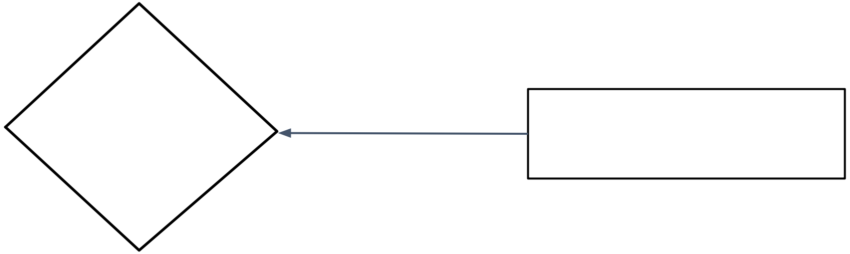
Reviewing



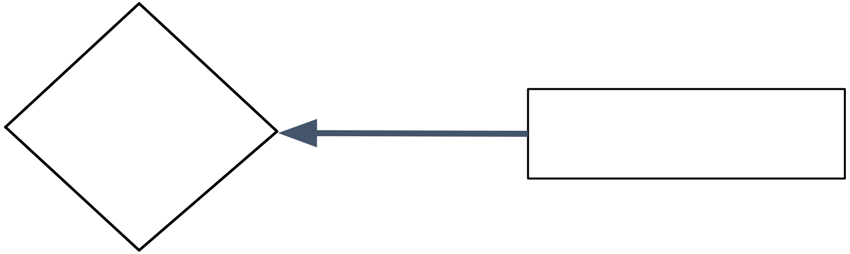
Reviewing



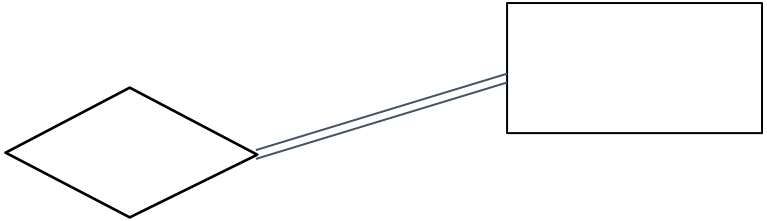
Reviewing



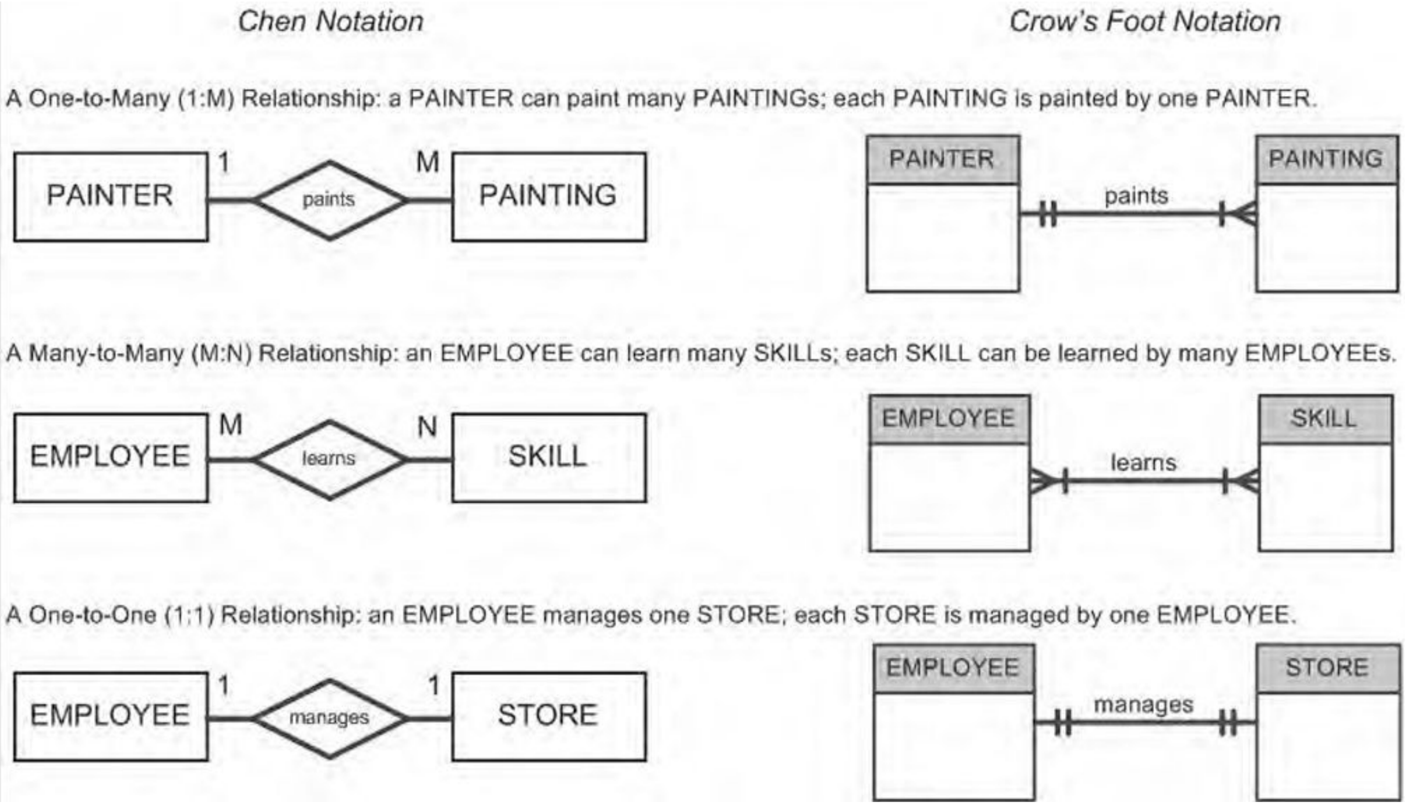
Reviewing



Reviewing



Reviewing



Chen Notation

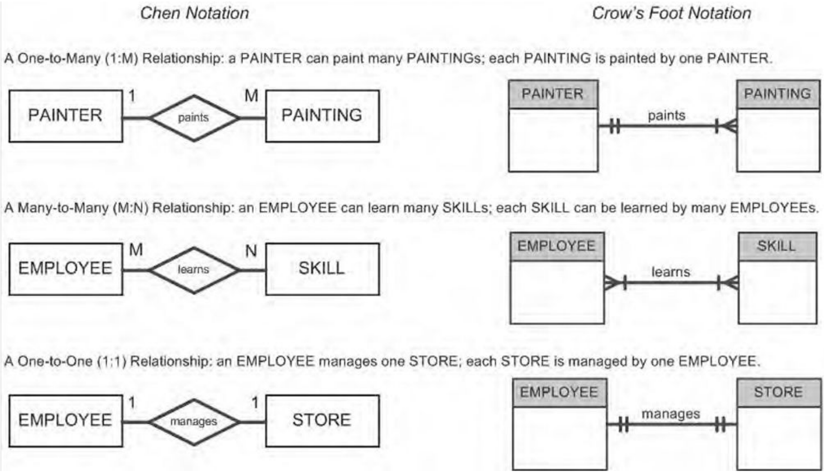
Crow's Foot Notation

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

Reviewing

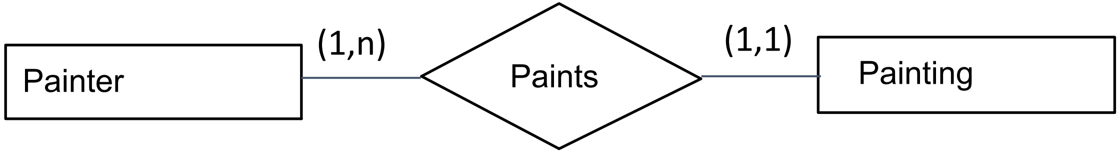


Chen Notation *Crow's Foot Notation*

A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

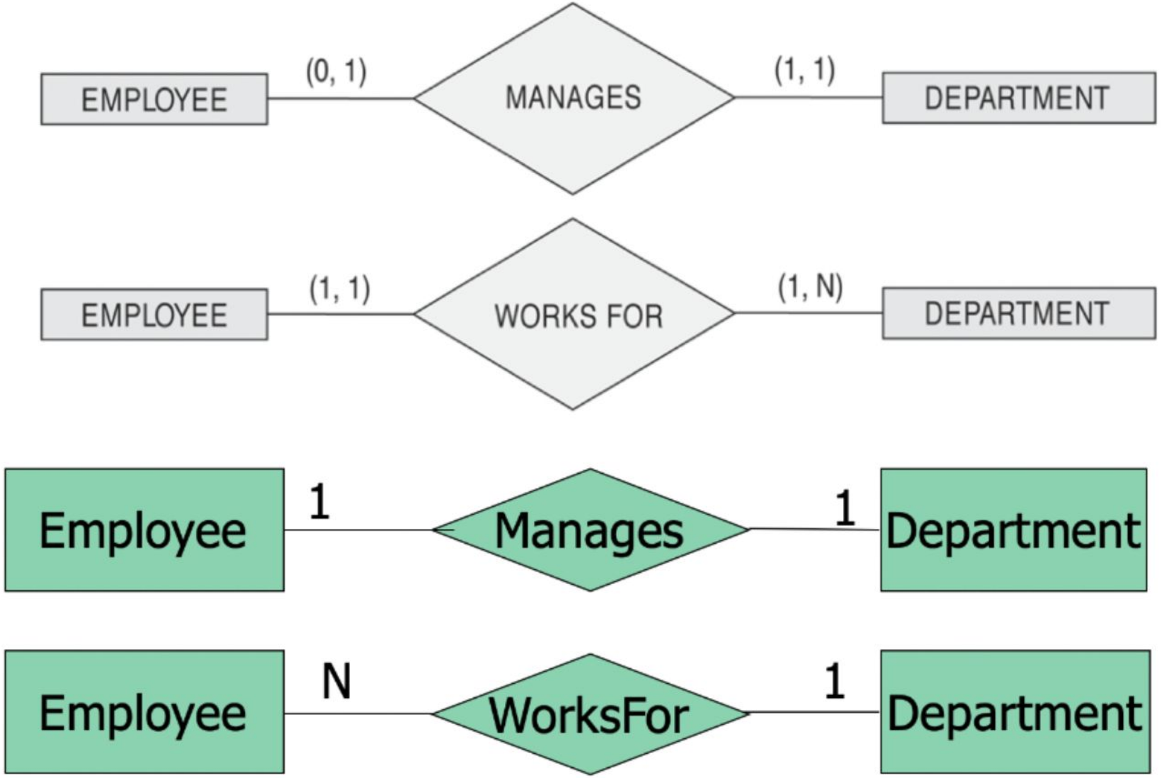
A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

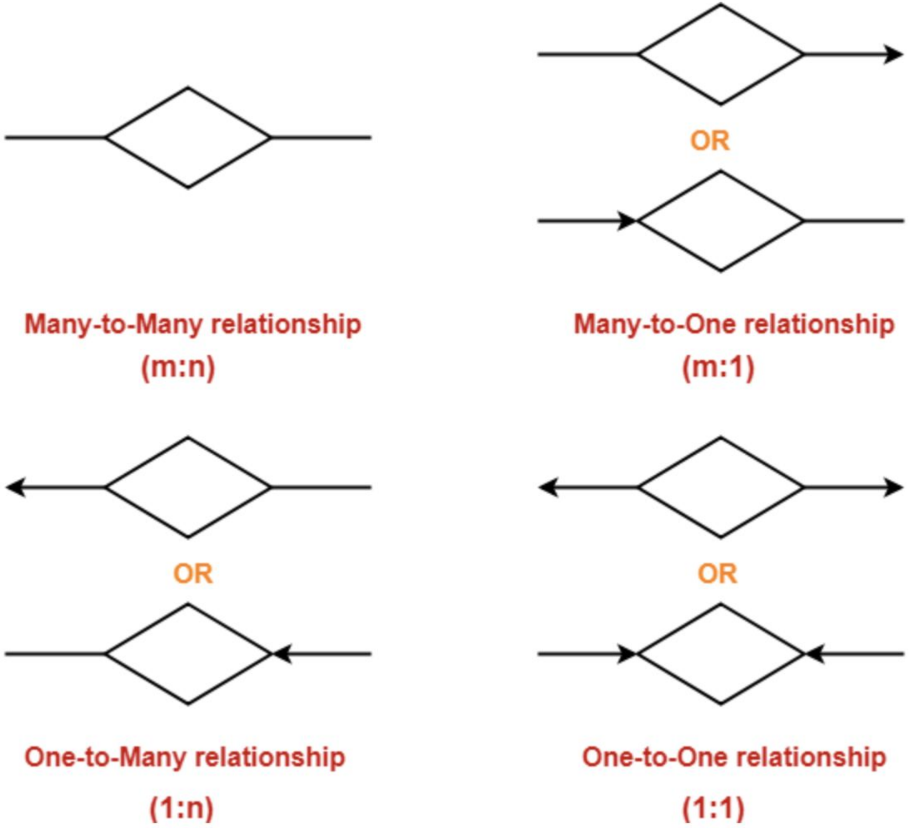


Painter (1,n) Paints (1,1) Painting

Reviewing


EMPLOYEE
(0, 1)
MANAGES
(1, 1)
DEPARTMENT
EMPLOYEE
(1, 1)
WORKS FOR
(1, N)
DEPARTMENT
Employee
1
Manages
1
Department
Employee
N
WorksFor
1
Department

Reviewing



Many-to-Many relationship
(m:n)

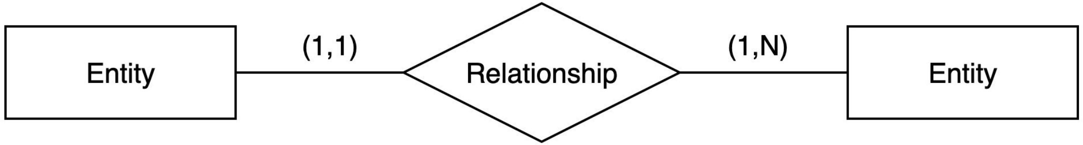
Many-to-One relationship
(m:1)

One-to-Many relationship
(1:n)

One-to-One relationship
(1:1)

Problem

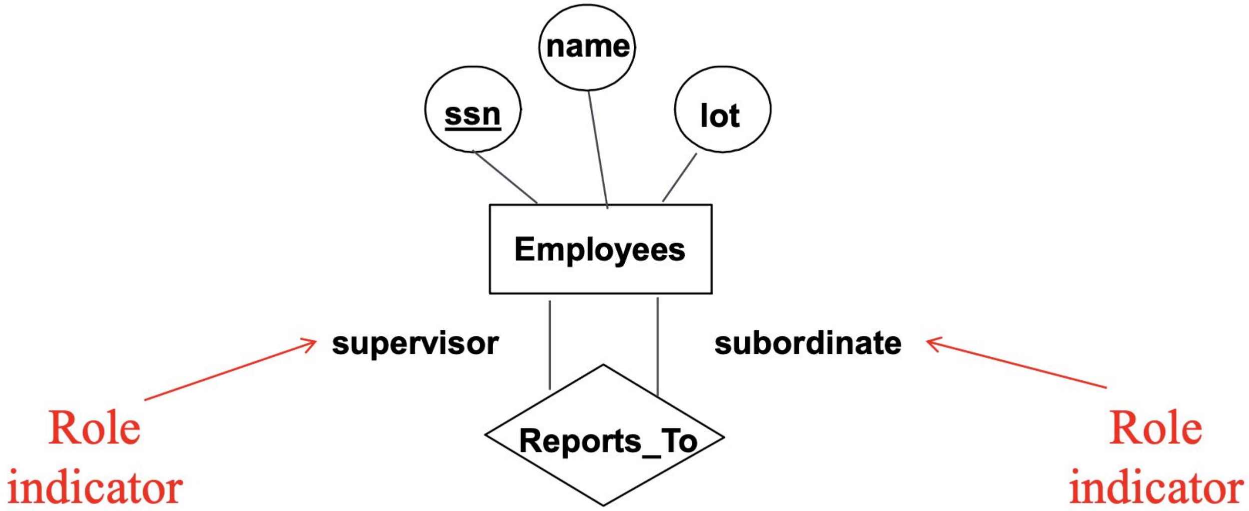
Generate other visualizations of cardinality constraints:



1. Crow’s Foot
2. Directed line based
3. Chen

A Special Case of Relationship

- An entity set can participate in a relationship set with itself
  - Entities in same set play different ***roles*** in the relationship
  - ***Role indicators*** express the role

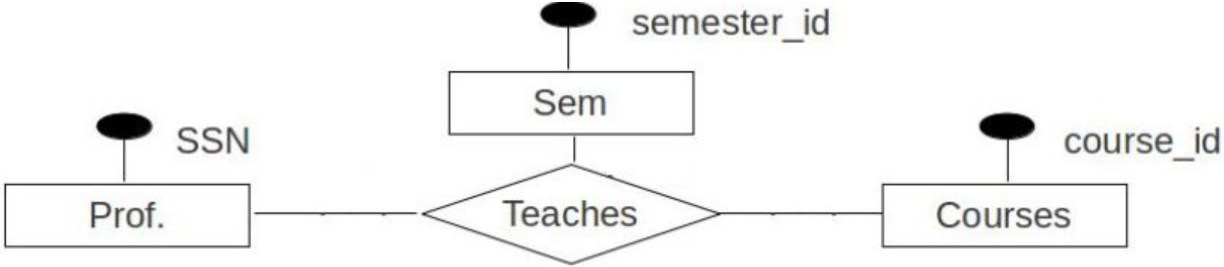


Example 1 (a)

- A university database contains information about **professors** (identified by social security number, or SSN) and **courses** (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).
  - Professors can teach the same course in several semesters, and each offering must be recorded

Example 1 (a)

- A university database contains information about **professors** (identified by social security number, or SSN) and **courses** (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).
  - Professors can teach the same course in several semesters, and each offering must be recorded

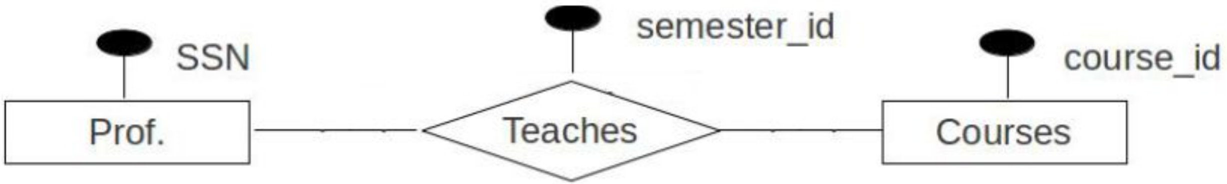


Example 1(b)

- A university database contains information about **professors** (identified by social security number, or SSN) and **courses** (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).
  - Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded (Assume this condition applies in all subsequent questions)

Example 1(b)

- A university database contains information about **professors** (identified by social security number, or SSN) and **courses** (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).
  - Professors can teach the same course in several semesters, and only the most recent such offering needs to be recorded (Assume this condition applies in all subsequent questions)

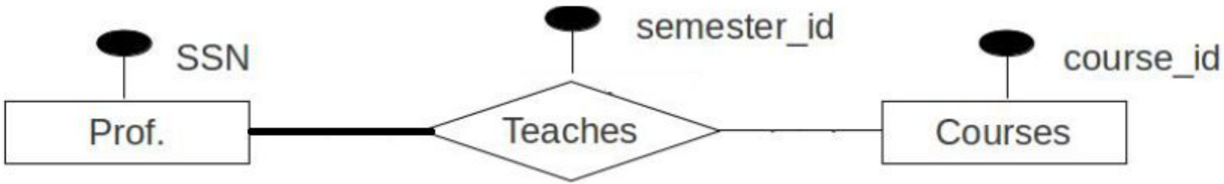


Example 1(c)

- A university database contains information about **professors** (identified by social security number, or SSN) and **courses** (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).
  - Every professor must teach some course

Example 1(c)

- A university database contains information about **professors** (identified by social security number, or SSN) and **courses** (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).
  - Every professor must teach some course

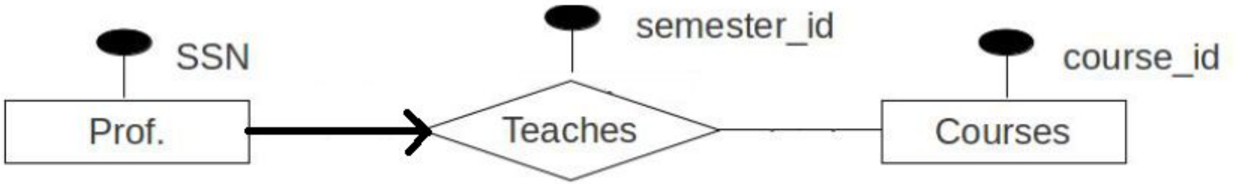


Example 1(d)

- A university database contains information about **professors** (identified by social security number, or SSN) and **courses** (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).
    - Every professor teaches exactly one course (no more, no less)

Example 1(d)

- A university database contains information about **professors** (identified by social security number, or SSN) and **courses** (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).
    - Every professor teaches exactly one course (no more, no less)

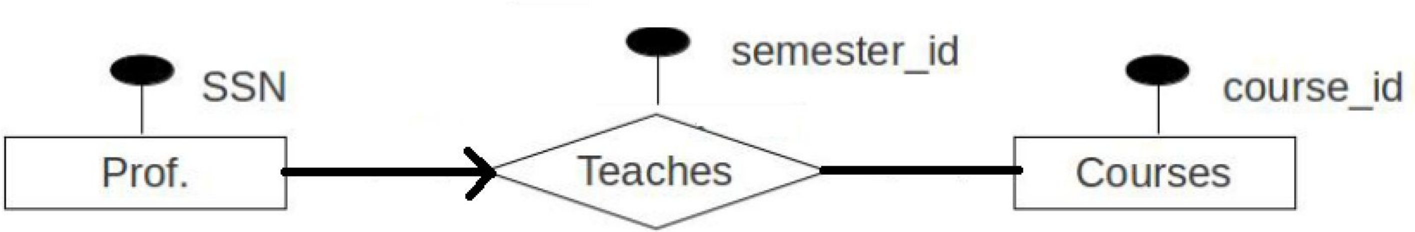


Example 1(e)

- A university database contains information about **professors** (identified by social security number, or SSN) and **courses** (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).
  - Every professor teaches exactly one course (no more, no less), and every course must be taught by some professor

Example 1(e)

- A university database contains information about **professors** (identified by social security number, or SSN) and **courses** (identified by courseid). Professors teach courses; each of the following situations concerns the Teaches relationship set. For each situation, draw an ER diagram that describes it (assuming no further constraints hold).
    - Every professor teaches exactly one course (no more, no less), and every course must be taught by some professor



Example 2

A company database needs to store information about employees (identified by ssn, with salary and phone as attributes), departments (identified by dno, with dname and budget as attributes), and children of employees (with name and age as attributes).

- Employees work in departments;
- Each department is managed by an employee;
- A child must be identified uniquely by name when the parent (who is an employee; assume that only one parent works for the company) is known.
- We are not interested in information about a child once the parent leaves the company.

Draw an ER diagram that captures this information (only major attributes shown).

Example 2

A company database needs to store information about **employees** (identified by **ssn**, with **salary** and **phone** as attributes), **departments** (identified by **dno**, with **dname** and **budget** as attributes), and **children** of employees (with **name** and **age** as attributes).

- Employees **work** in departments;
- Each department is **managed** by an employee;
- A child must be identified uniquely by name when the parent (who is an employee; assume that only one parent works for the company) is known.
- We are not interested in information about a child once the parent leaves the company.

Draw an ER diagram that captures this information (only major attributes shown).

Example 3

- A university has many departments.
- The department offers multiple courses and each course belongs to only one department.
- One course is enrolled by multiple students and one student for multiple courses.
- One department has multiple instructors and one instructor belongs to one and only one department.
- Each department has one “HOD” and one instructor is “HOD” for only one department. Here, HOD refers to the head of the department.
- One course is taught by only one instructor but one instructor teaches many courses.

Draw an ER diagram that captures this information

Example 3

- A **university** has many departments.
- The **department** **offers** multiple courses and each course belongs to only one department.
- One **course** is **enrolled** by multiple students and one **student** for multiple courses.
- One department **has** multiple **instructors** and one instructor belongs to one and only one department.
- Each department has one “**HOD**” and one instructor is “HOD” for only one department. Here, HOD refers to the head of the department.
- One course is **taught** by only one instructor but one instructor teaches many courses.

Draw an ER diagram that captures this information

Example 4 (Open Ended)

A pharmacy wants to create an ER diagram to represent the relationship between customers, prescriptions, drugs, and employees.

Draw an ER diagram with your own assumptions.

Example 5 (Observations)

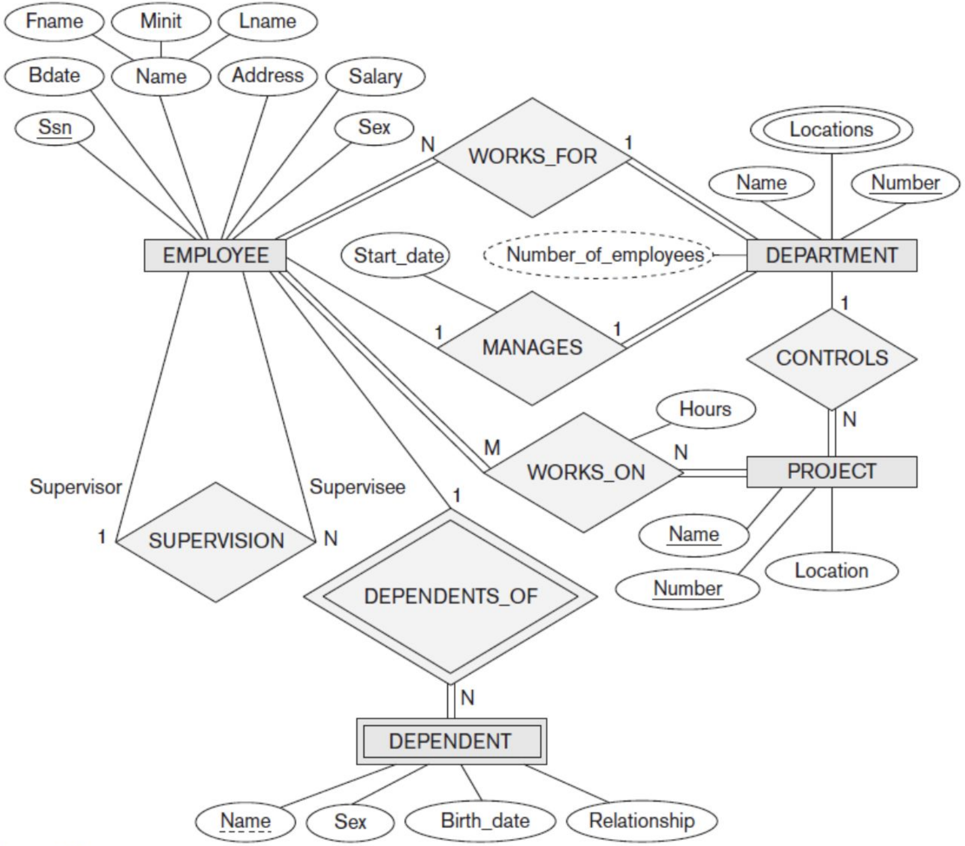
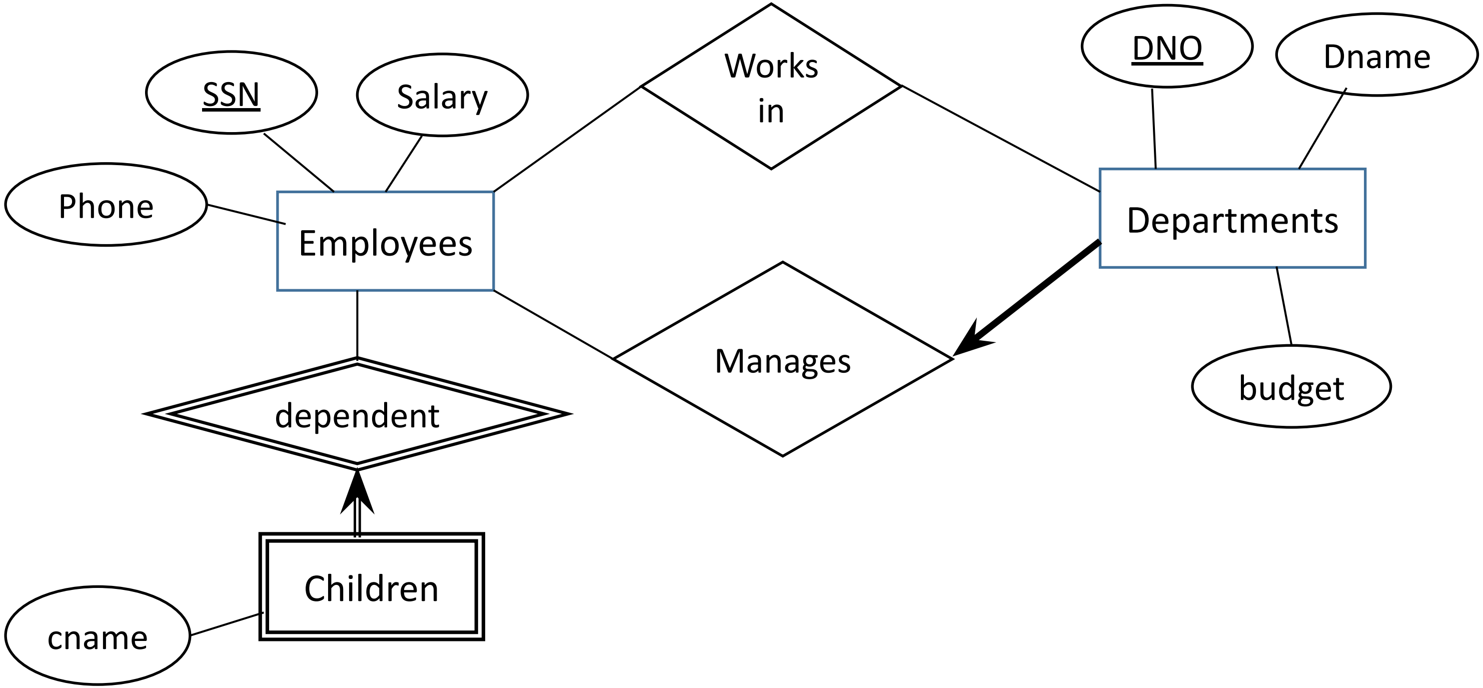
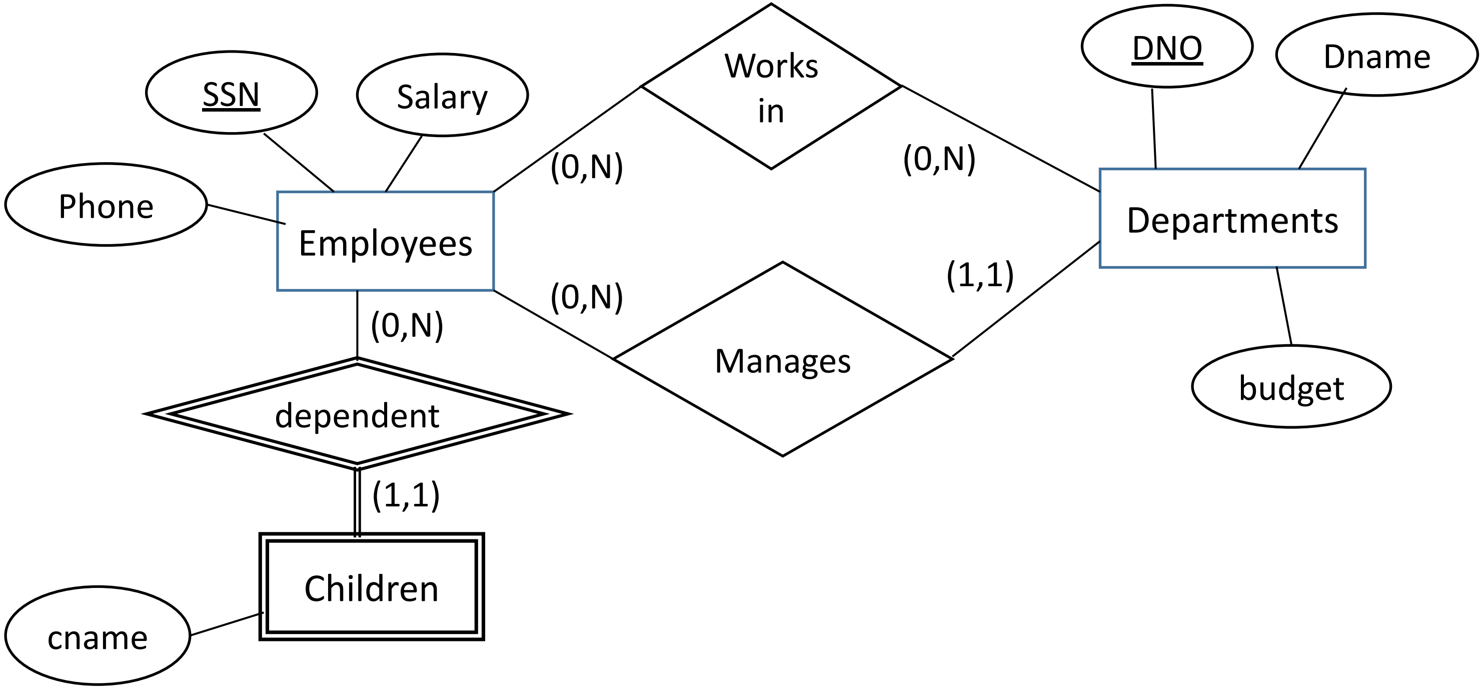
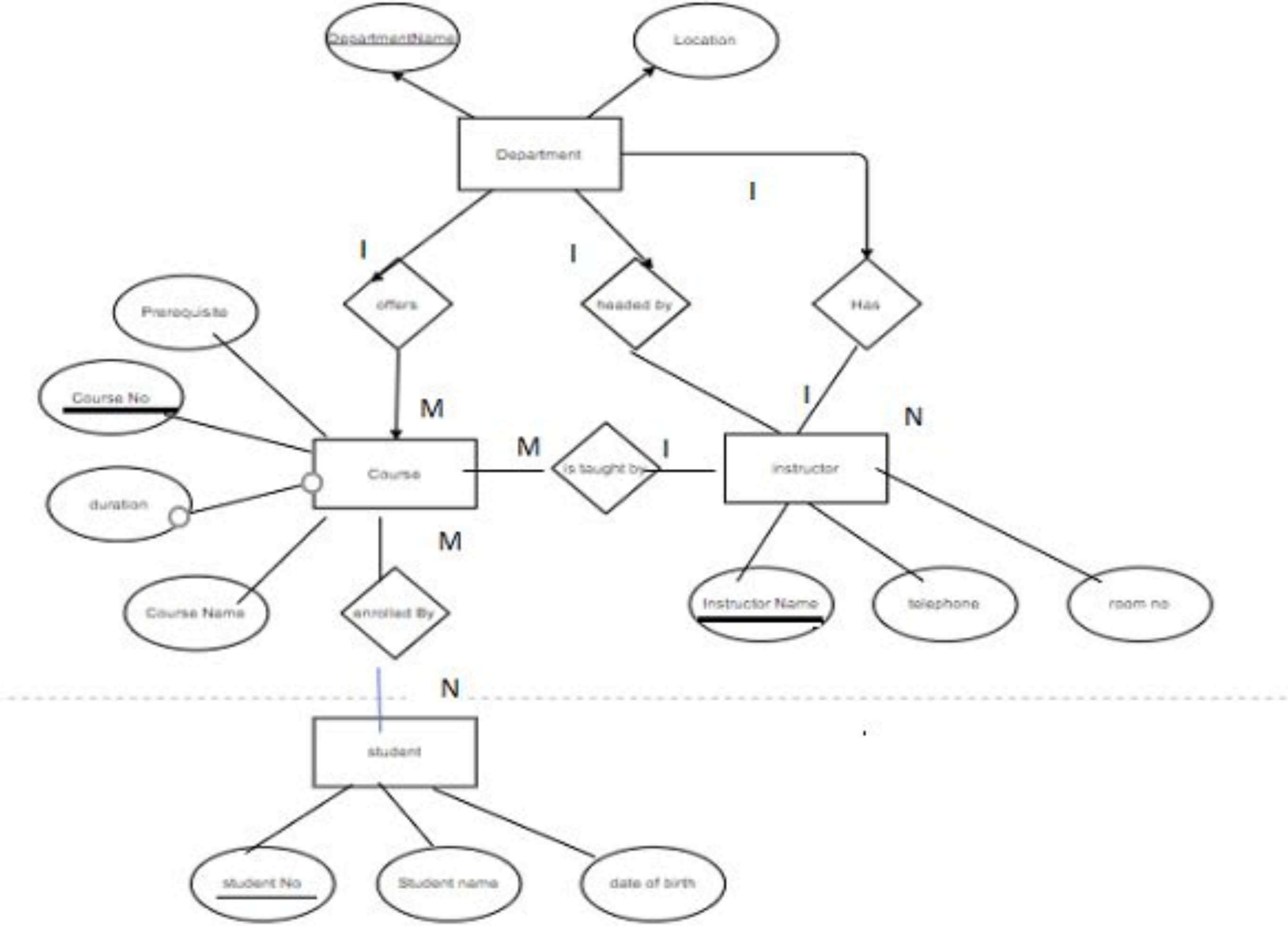


Figure 7.2


SSN
Salary
Phone
Employees
Works in
DNO
Dname
Departments
Manages
budget
dependent
Children
cname


SSN
Salary
Phone
Employees
Works in
(0,N)
(0,N)
DNO
Dname
Departments
(0,N)
(1,1)
Manages
budget
(0,N)
dependent
(1,1)
Children
cname


DepartmentName
Location
Department
I
I
I
offers
headed by
Has
Prerequisite
Course No
M
I
N
M
I
duration
Course
is taught by
instructor
M
Course Name
enrolled By
Instructor Name
telephone
room no
N
student
student No
Student name
date of birth

Example 3

- The entities in this diagram would be customers, prescriptions, drugs, and employees.
- Each customer would have attributes such as name, address, and customer ID.
- Each prescription would have attributes such as prescription ID, doctor's name, and date filled.
- Each drug would have attributes such as drug ID, name, and dosage.
- Each employee would have attributes such as employee ID, name, and position.
- The relationship between customers and prescriptions: "has."
- The relationship between prescriptions and drugs: "contains."
- The relationship between drugs and employees: "dispensed by."