Tutorial 1: ER Diagram

CSCC43, 2026 Summer

TA: Ruiyu Wang

General Info

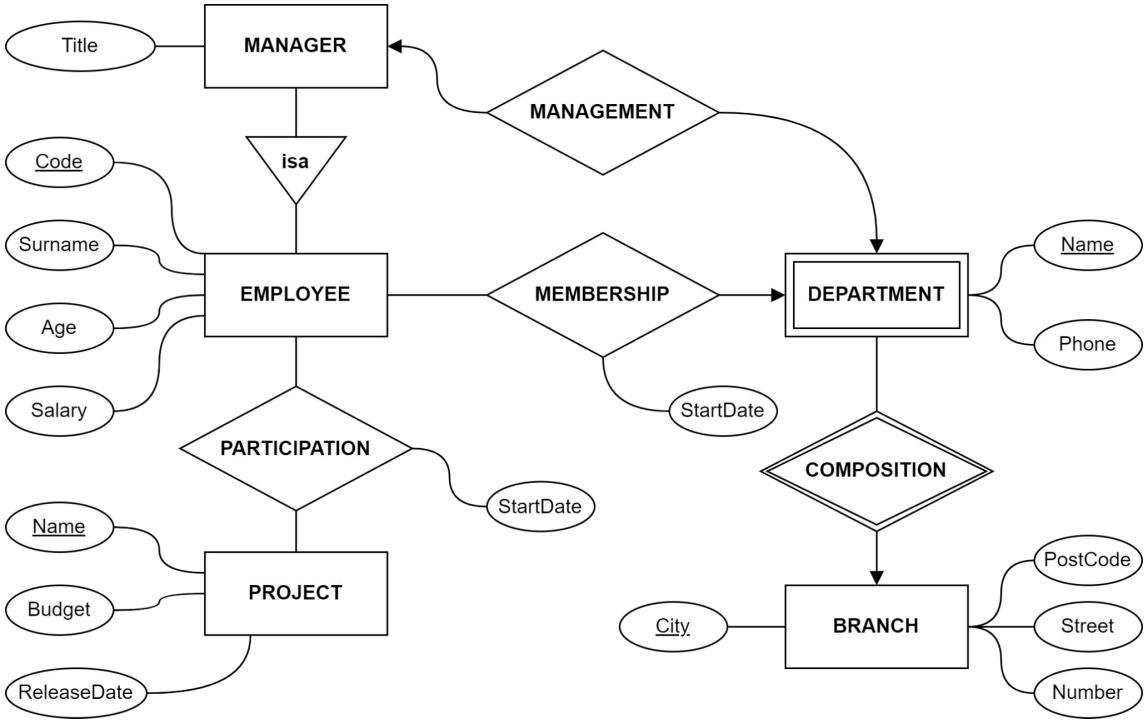
TA: Ruiyu Wang

Email: rwang@cs.toronto.edu (only for tutorial related questions)

Office hour: Monday 3-4 pm, CS Help Centre, 3rd floor of IA

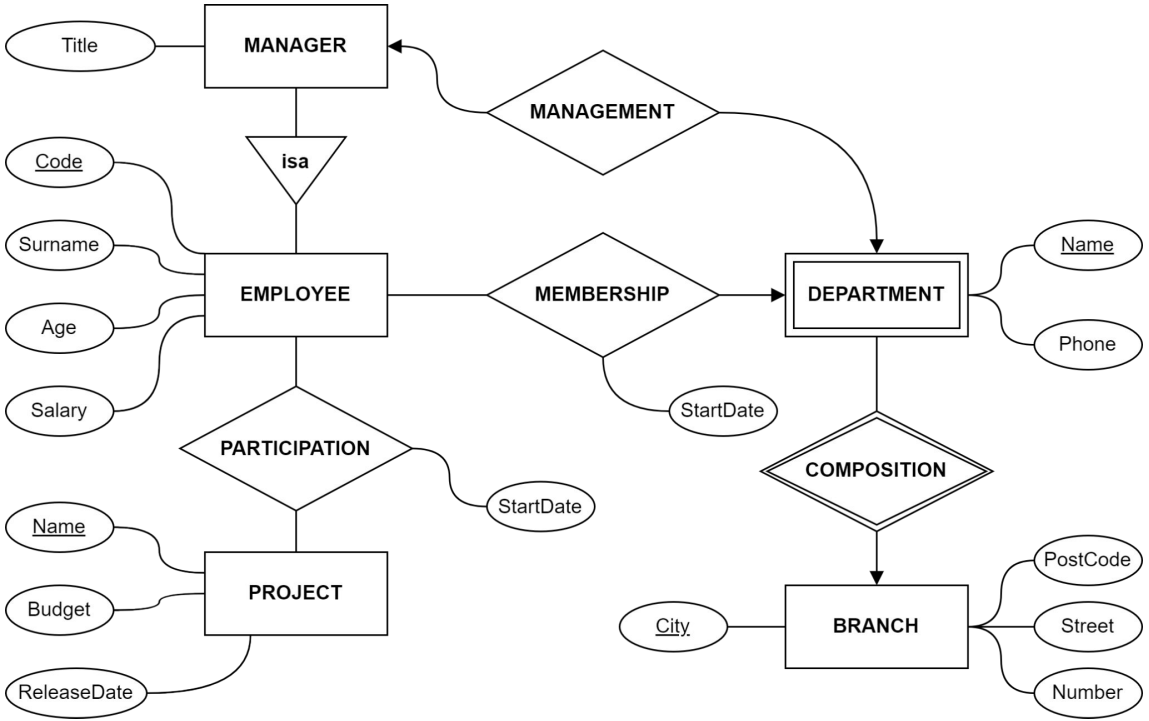
If you cannot make it to the office hour: it's fine! I will allocate some time for questions in each TUT session

ER Diagram Overview



1. Entities, relationships, and attributes
2. ER -> relational schema
3. Collapsing information

Entities & Relationships



Entities:

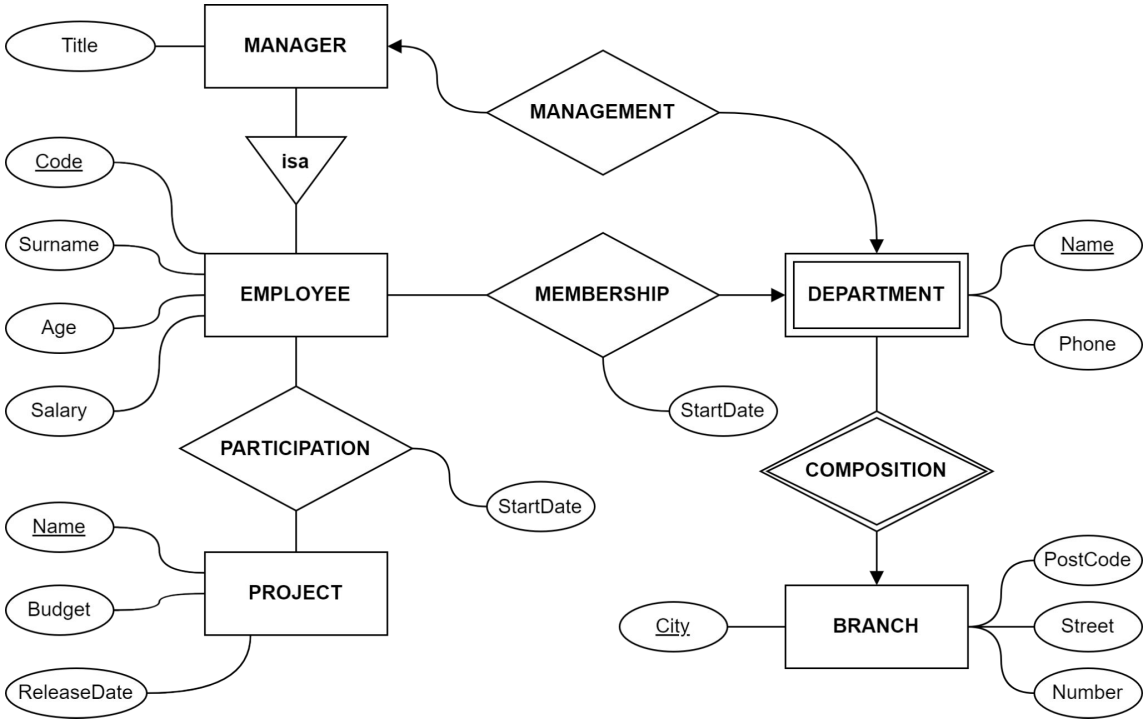
- Objects in the diagram
- Represented by rectangles
- Attributes: properties of entities, in oval

Relationships:

- Connections between entities
- Represented by diamonds

What are entities and what are relationships?

Weak Entities



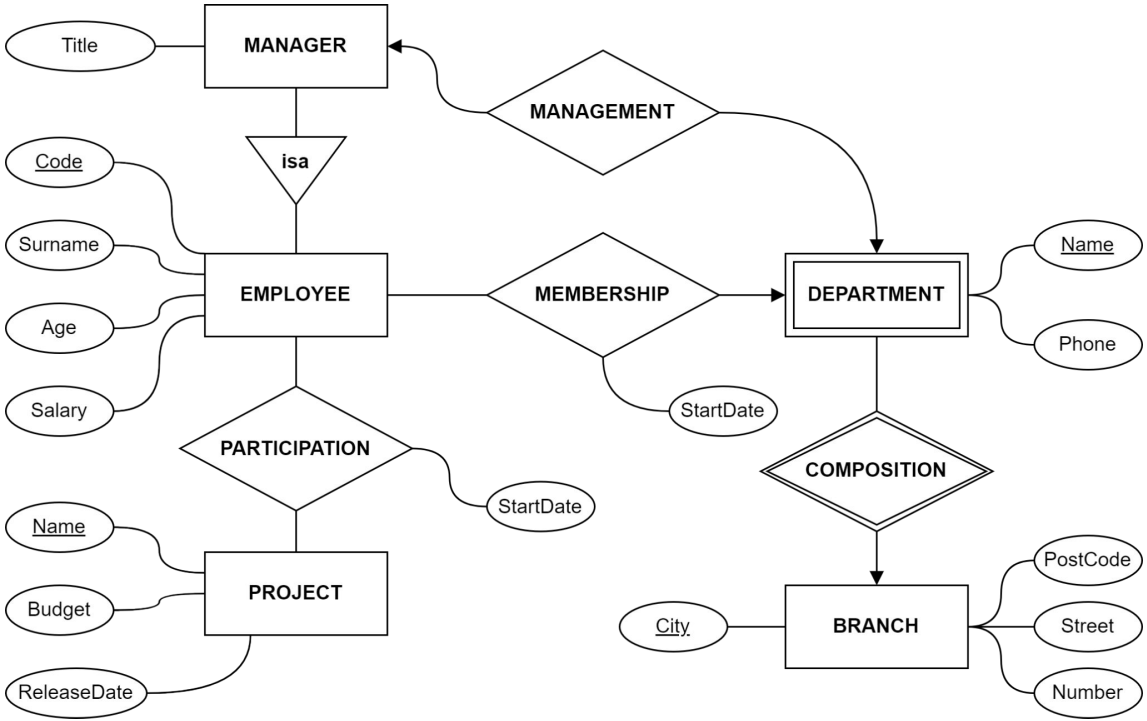
Entities can stand on its own.
i.e. a project is fully described by its name

Weak entities are entities that have to link with other entities to be unique.

In annotation, we make them **double rectangle**, and the supporting relationship as **double diamond**

What is the weak entity in the diagram?

Weak Entities

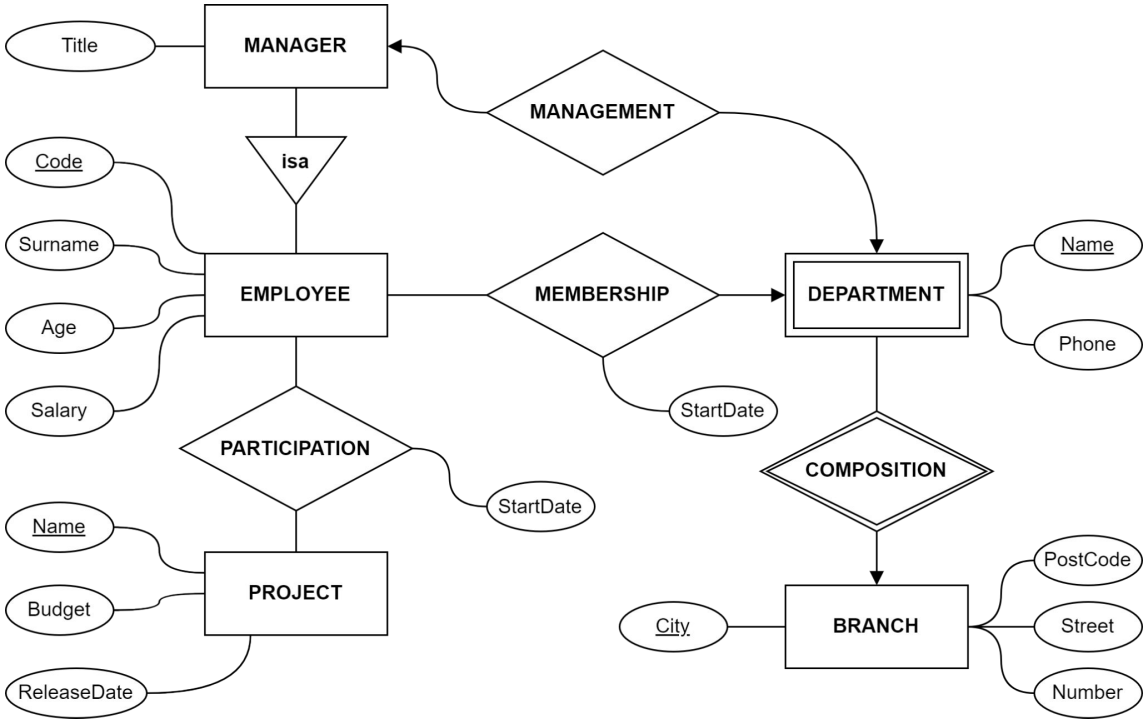


What is the weak entity in the diagram?

A: **department**, linked to **branch** by **composition**

Dept. Risk Management, Toronto branch is different from Dept. Risk Management, Montreal branch, right?

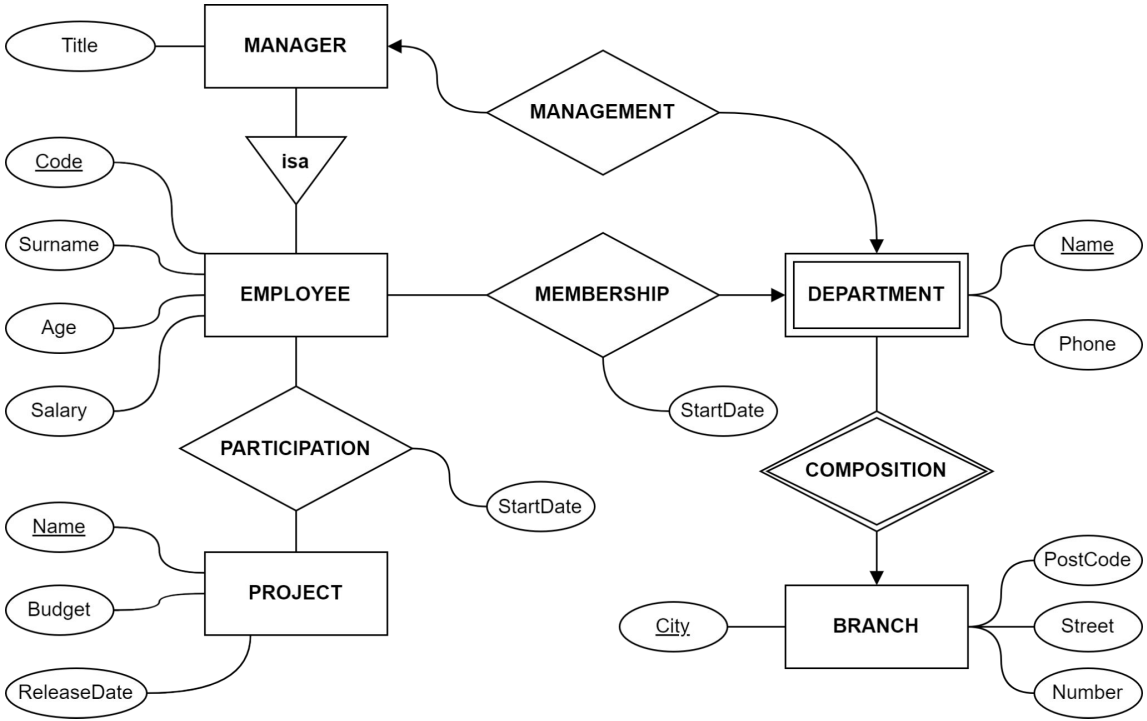
Weak Entities



What forms the key for this entity set?

Keys: the **minimal** attributes that makes this relation unique; otherwise superkeys.

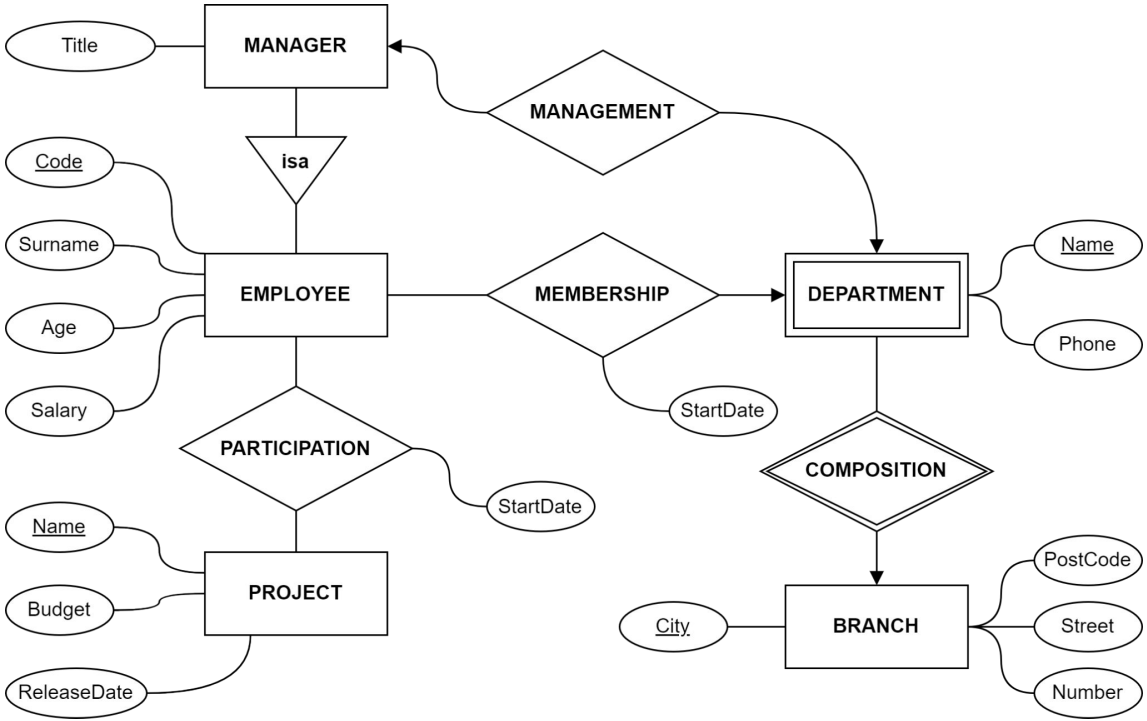
Weak Entities



What forms the key for this entity set?

A: Name, City

Relationships

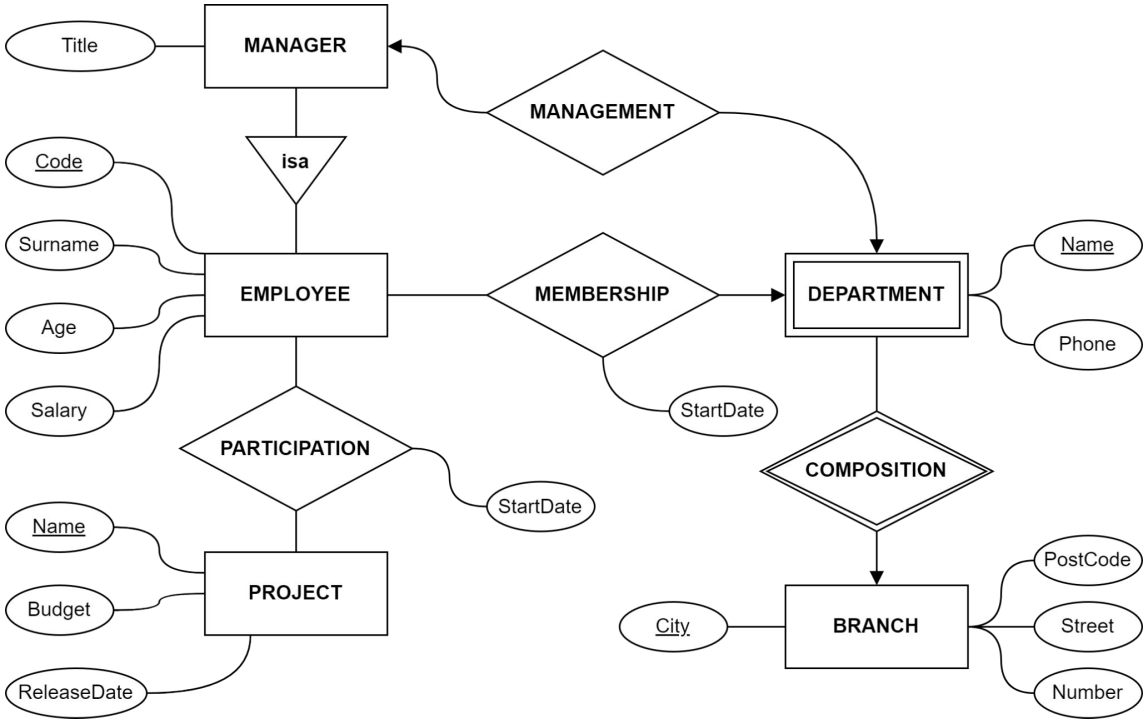


Relationships have arrows to describe multiplicity.

Arrows are pointing to the one's.

From the ER diagram, point out different relationships

Relationships



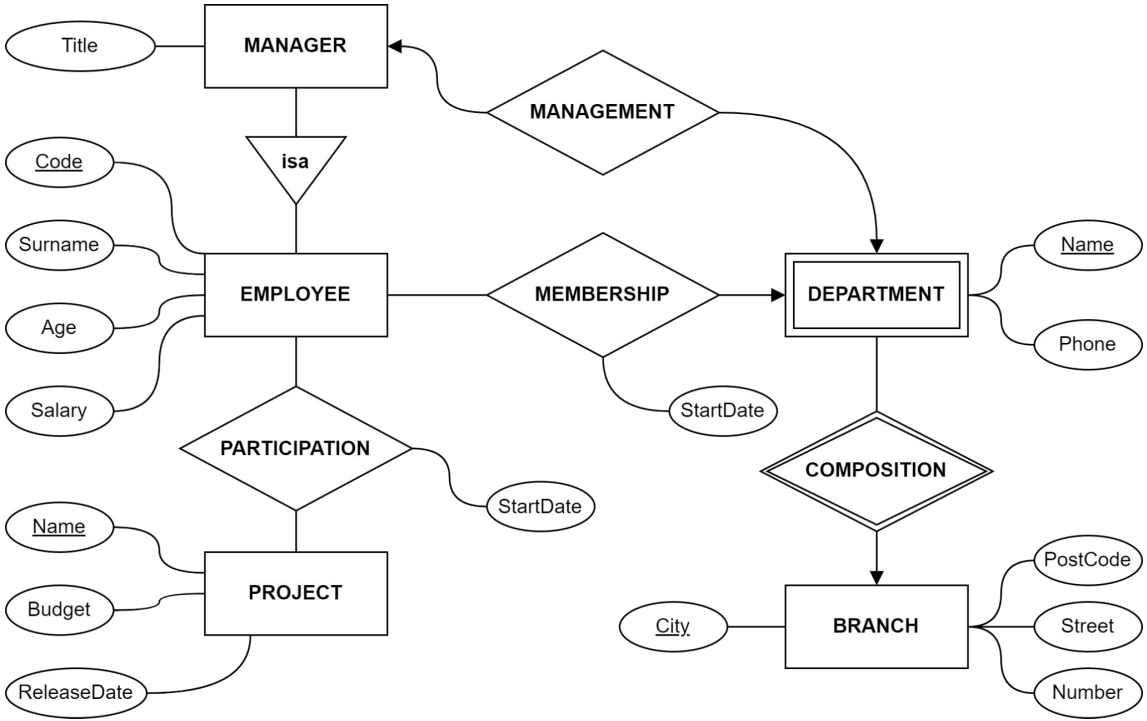
From the ER diagram, point out different types of relationships

One-to-one:

One-to-many:

Many-to-many:

Relationships



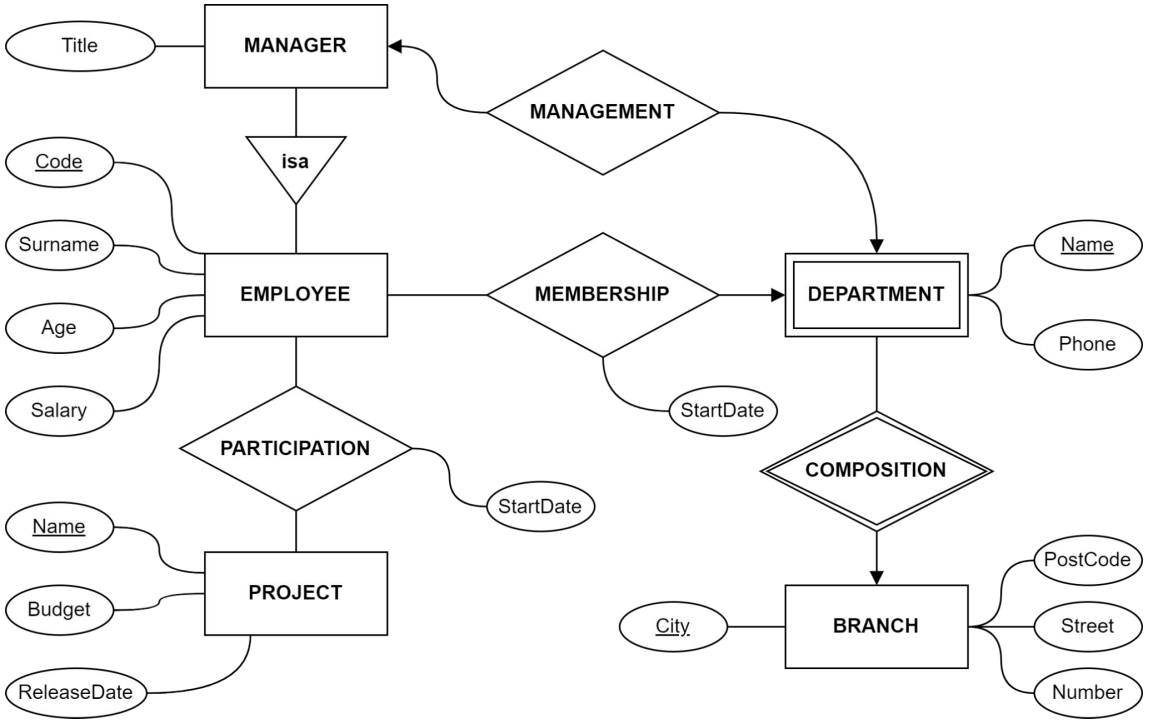
From the ER diagram, point out different types of relationships

One-to-one: **management**

One-to-many: **membership**

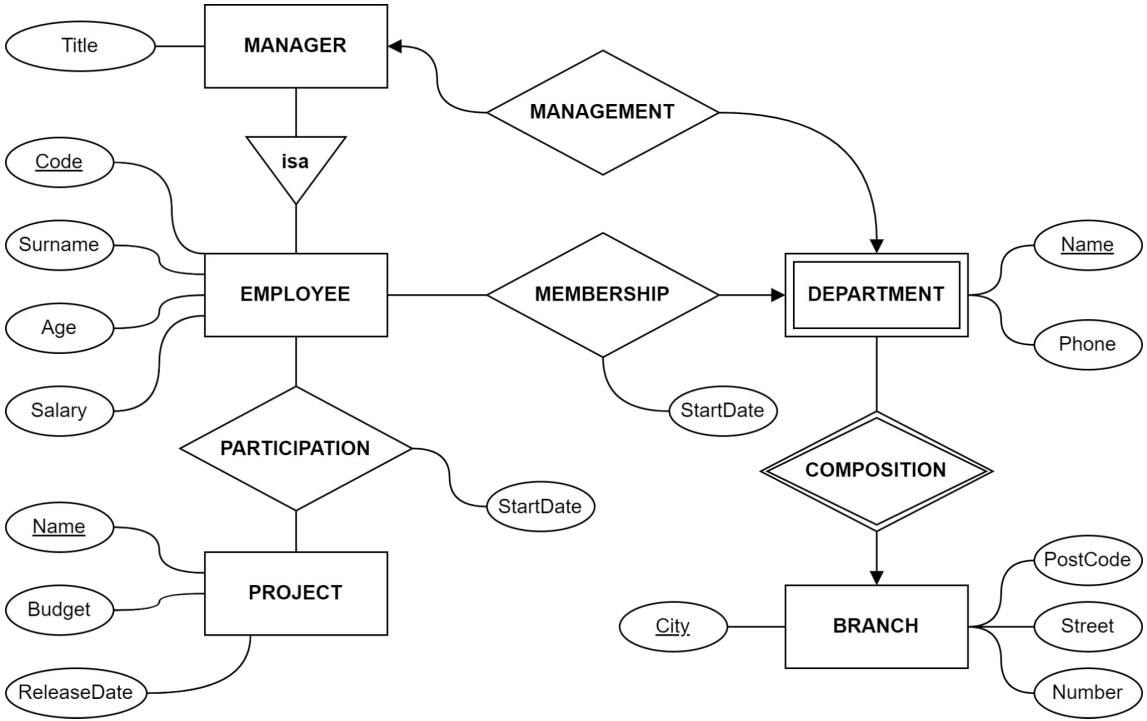
Many-to-many: **participation**

Relationships



Can a department exist without a manager?

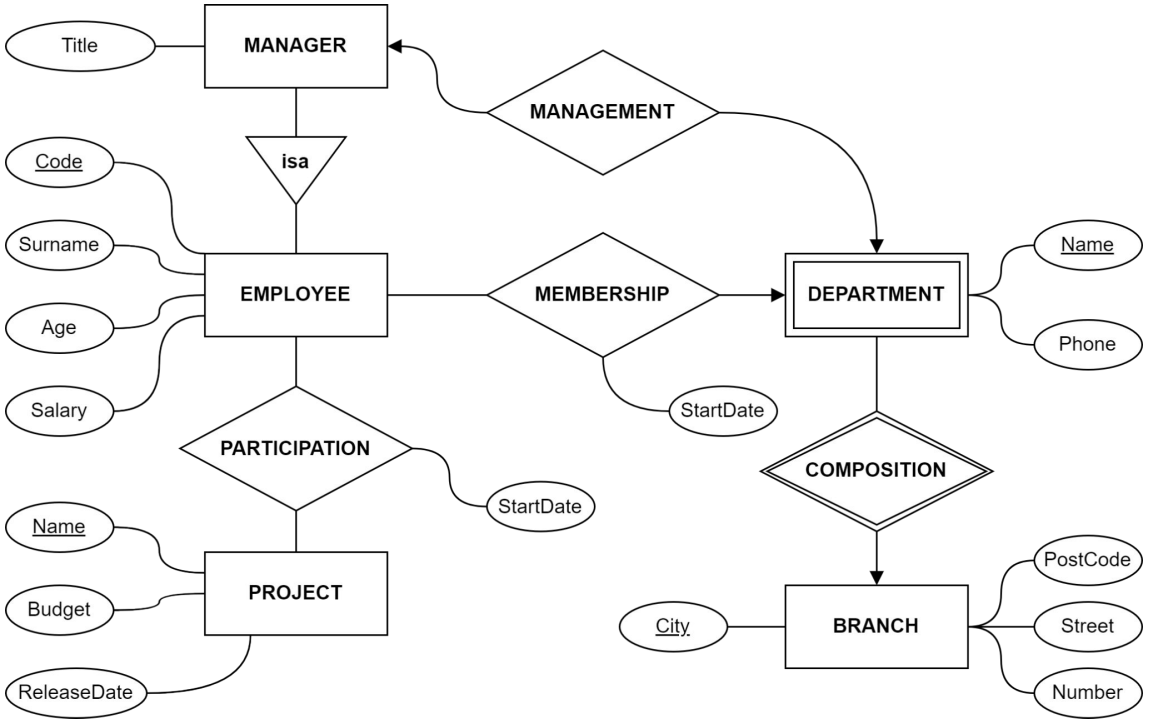
Relationships



Can a department exist without a manager?

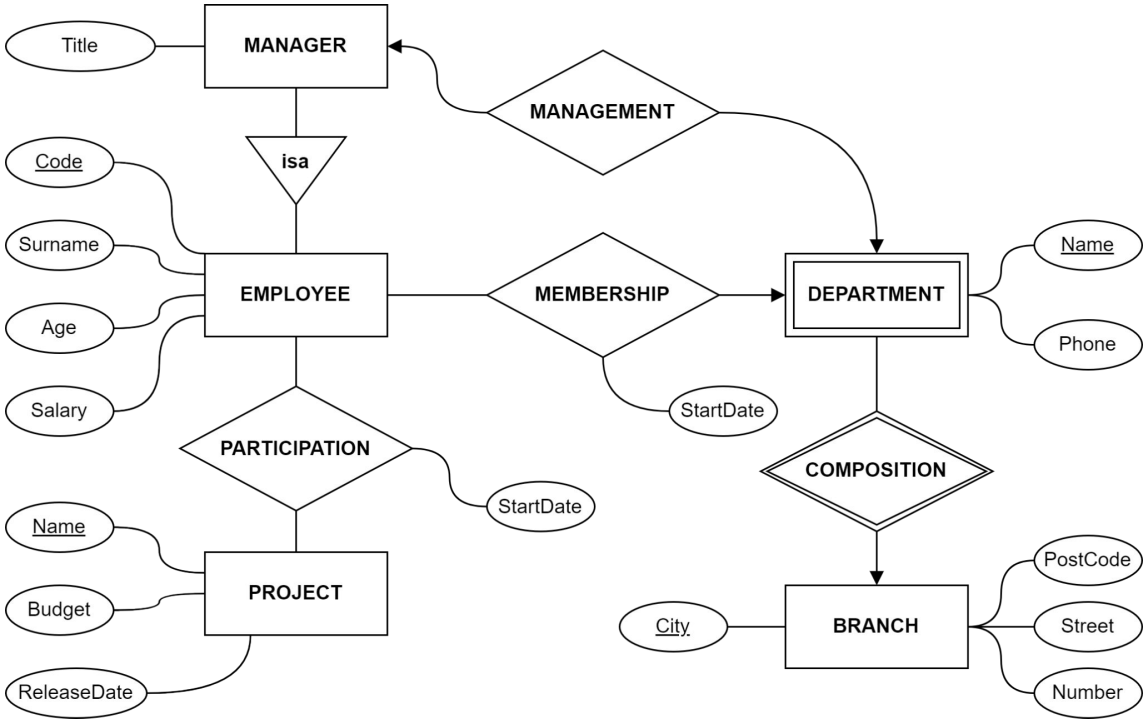
A: no, because the department and manager are described by a one-to-one relationship

Relationships



How many departments can an employee belong to?

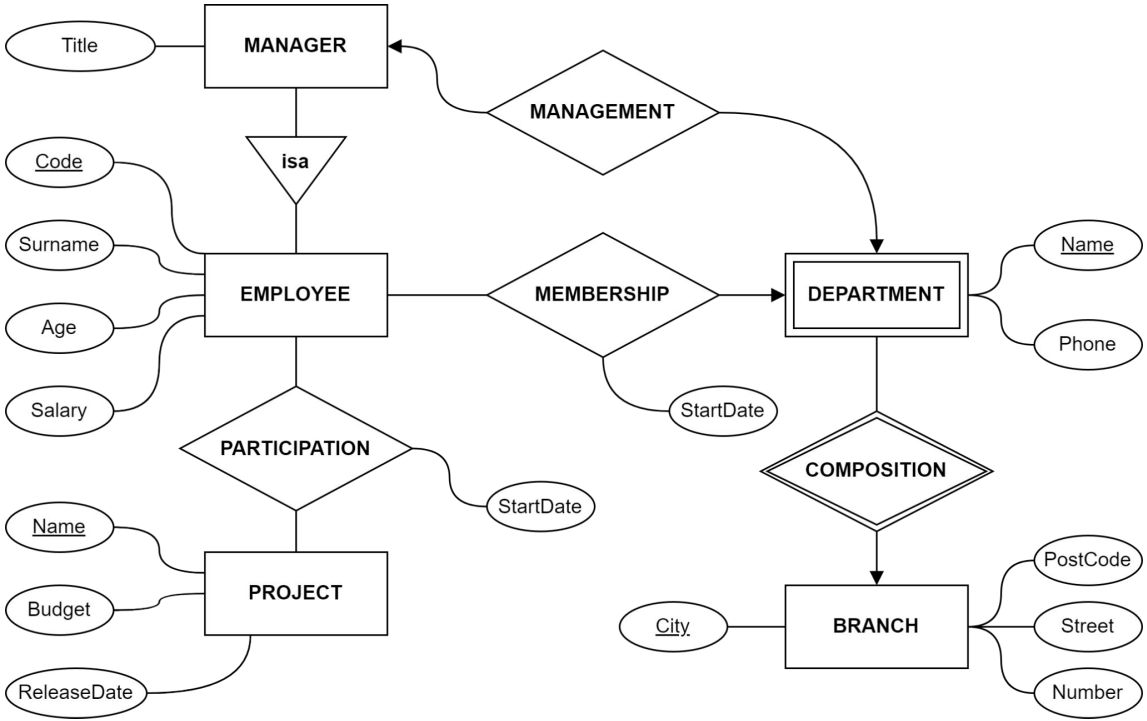
Relationships



How many departments can an employee belong to?

A: at most one.

ER Diagram -> Relational Schema



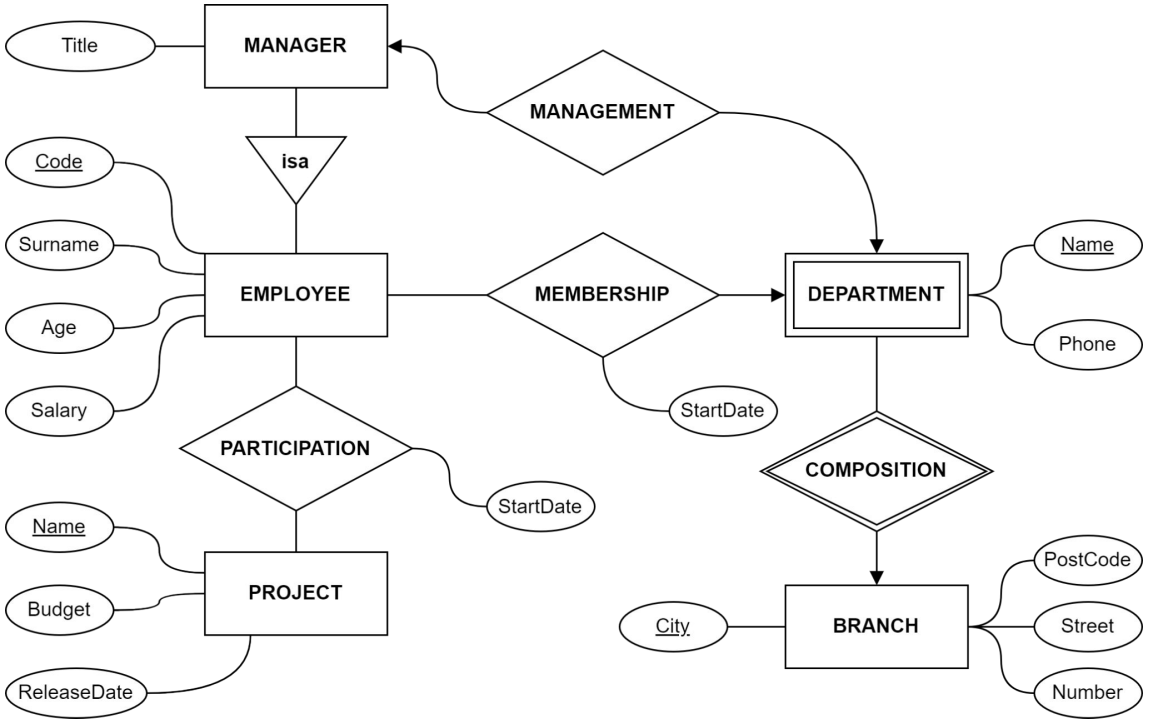
Translate the diagram into a relational schema.

What you need to do for this type of question: 1. Create one relation for every entity, and 2. Create one relation for every relationship set. 3. Underline keys and specify foreign keys

Keys: the **minimal** attributes that makes this relation unique; otherwise superkeys.

Foreign keys: attributes that are keys in other relations

ER Diagram -> Relational Schema

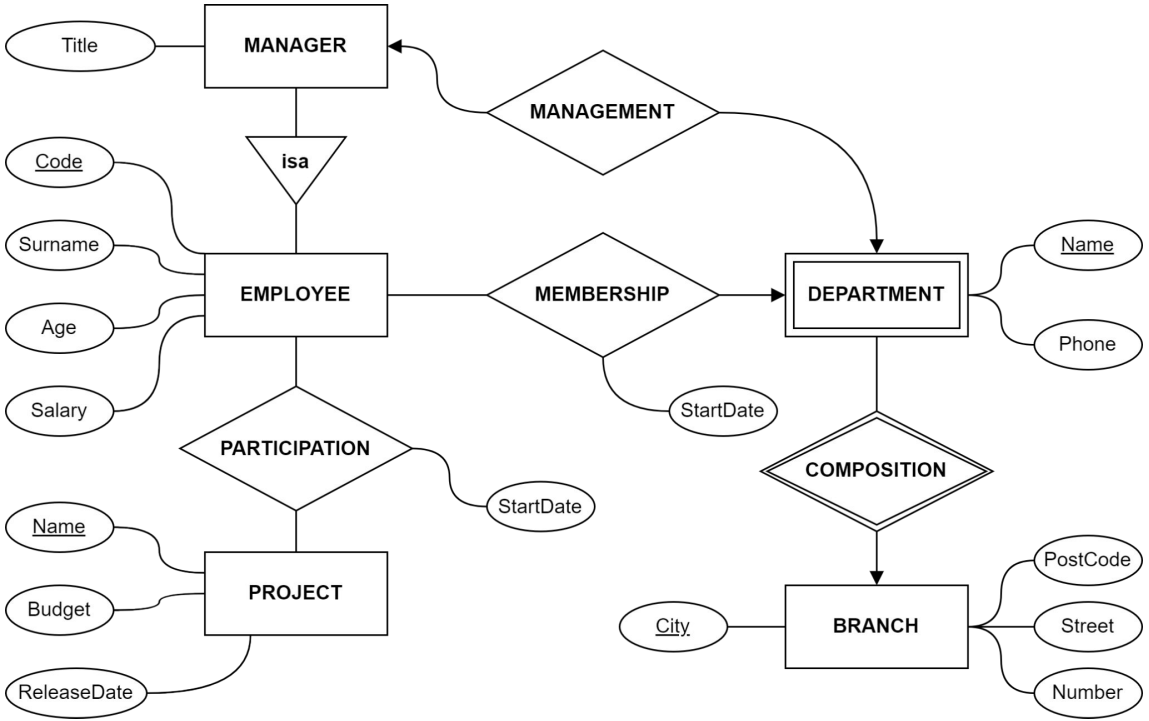


Translate the diagram into a relational schema.

Entities:

1. Manager(Code, Title); Code is a foreign key of Employee.

ER Diagram -> Relational Schema

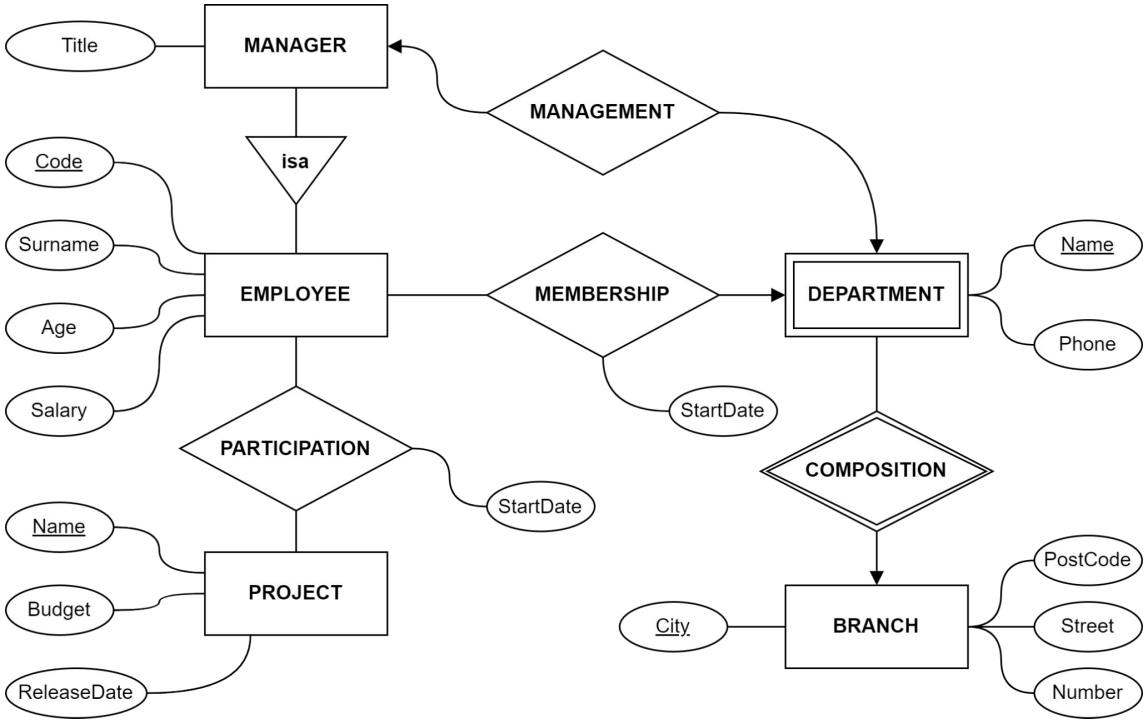


Translate the diagram into a relational schema.

Entities:

1. Manager(Code, Title); Code is a foreign key of Employee.
2. Employee(Code, Surname, Age, Salary)

ER Diagram -> Relational Schema

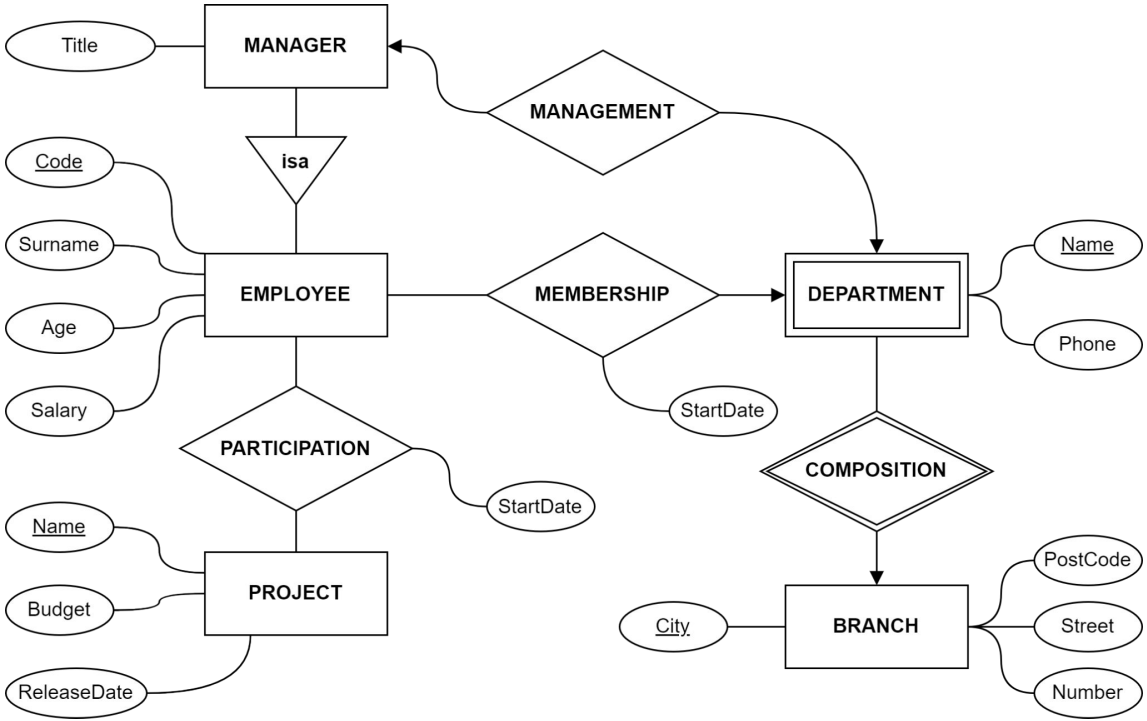


Translate the diagram into a relational schema.

Entities:

1. Manager(Code, Title); Code is a foreign key of Employee.
2. Employee(Code, Surname, Age, Salary)
3. Project(Name, Budget, ReleaseDate)

ER Diagram -> Relational Schema

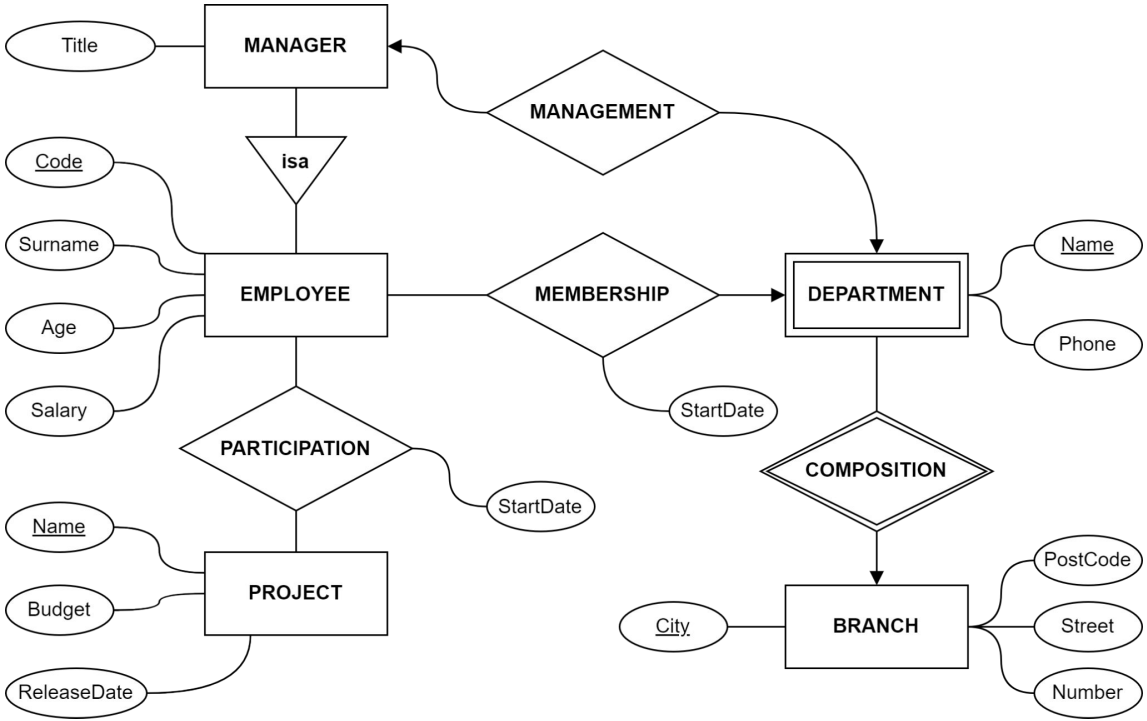


Translate the diagram into a relational schema.

Entities:

1. Manager(Code, Title); Code is a foreign key of Employee.
2. Employee(Code, Surname, Age, Salary)
3. Project(Name, Budget, ReleaseDate)
4. Department(Name, City, Phone); City is a foreign key of Branch

ER Diagram -> Relational Schema

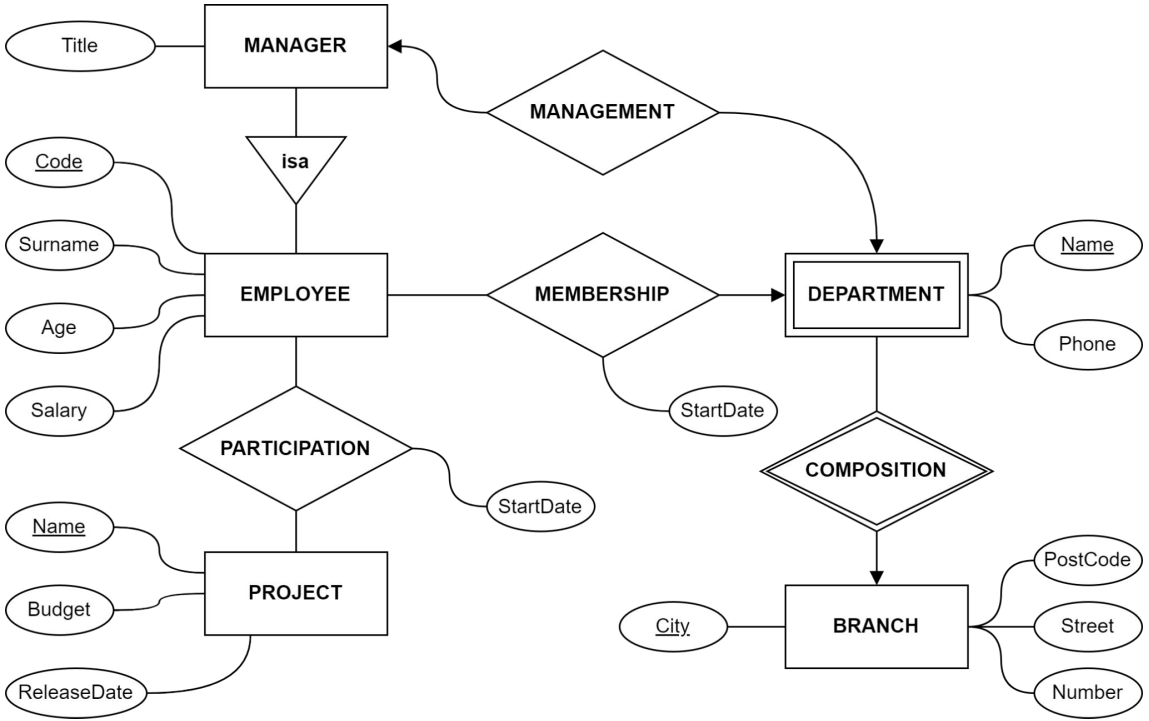


Translate the diagram into a relational schema.

Entities:

1. Manager(Code, Title); Code is a foreign key of Employee.
2. Employee(Code, Surname, Age, Salary)
3. Project(Name, Budget, ReleaseDate)
4. Department(Name, City, Phone); City is a foreign key of Branch
5. Branch(City, Number, Street, PostCode)

ER Diagram -> Relational Schema



Translate the diagram into a relational schema.

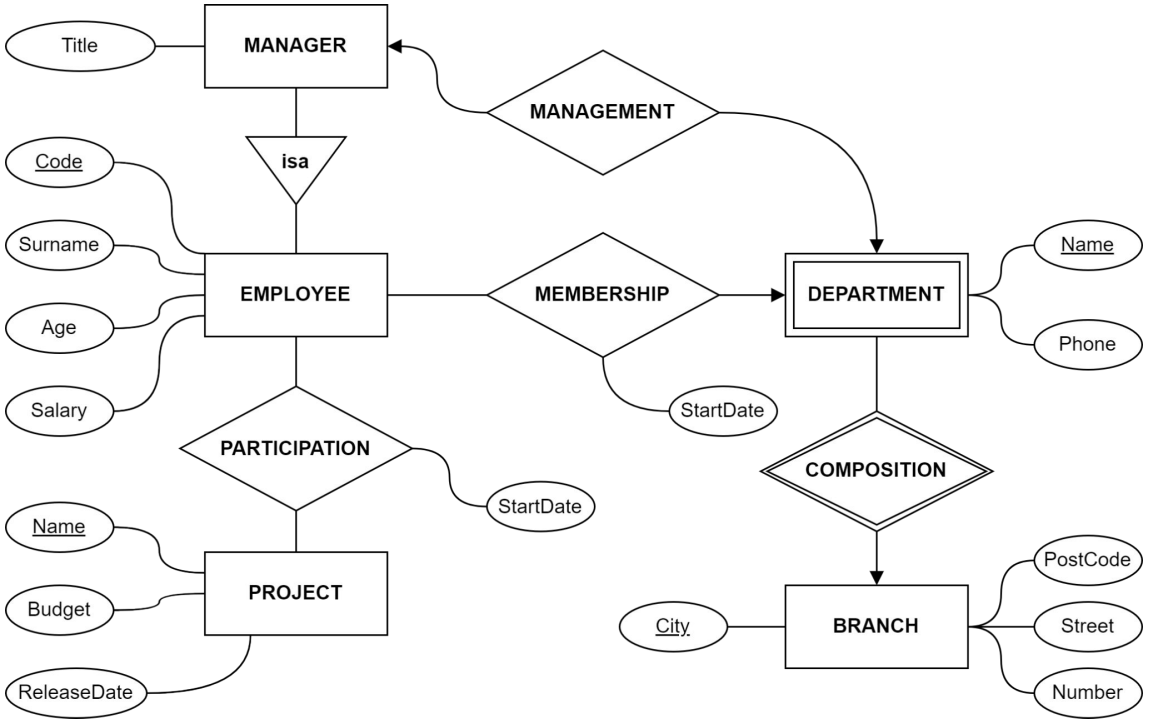
Relationships

1. Participation(<u>Code, Name</u>, StartDate)

Code is a foreign key of Employee;

Name is a foreign key of Project.

ER Diagram -> Relational Schema

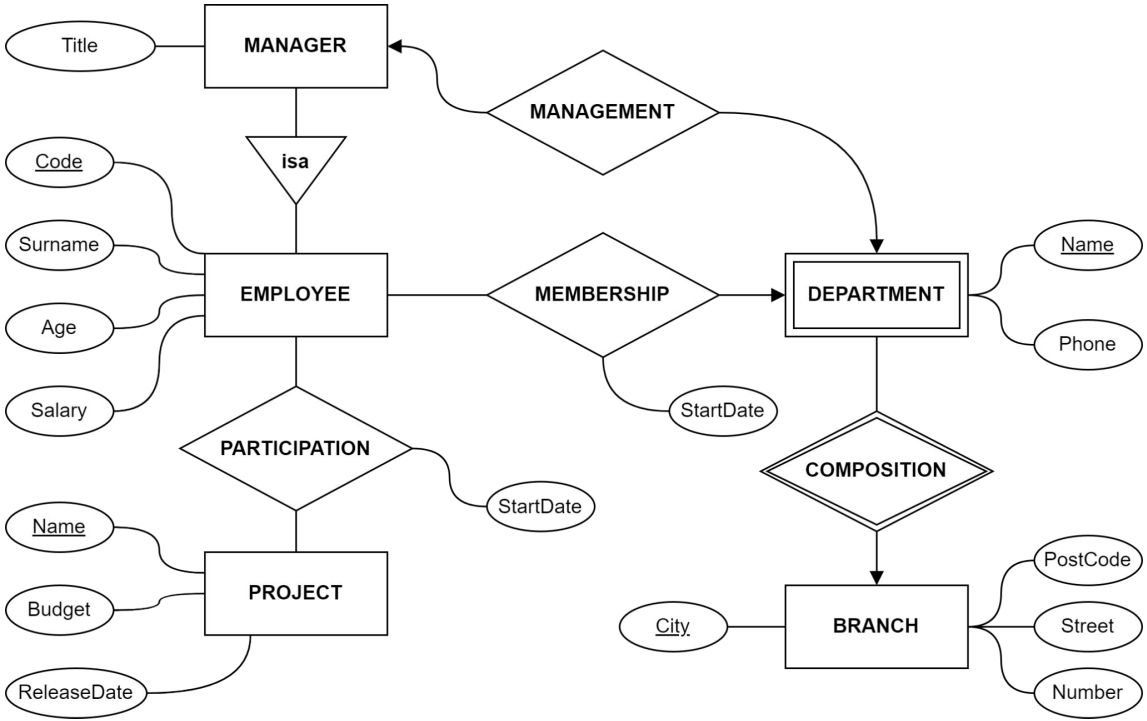


Translate the diagram into a relational schema.

Relationships

1. Participation(Code, Name, StartDate) Code is a foreign key of Employee; Name is a foreign key of Project.
2. Management(Code, Name, City) or Management(Code, Name, City); Code foreign key of Manager; (Name, City) foreign key of Department.

ER Diagram -> Relational Schema

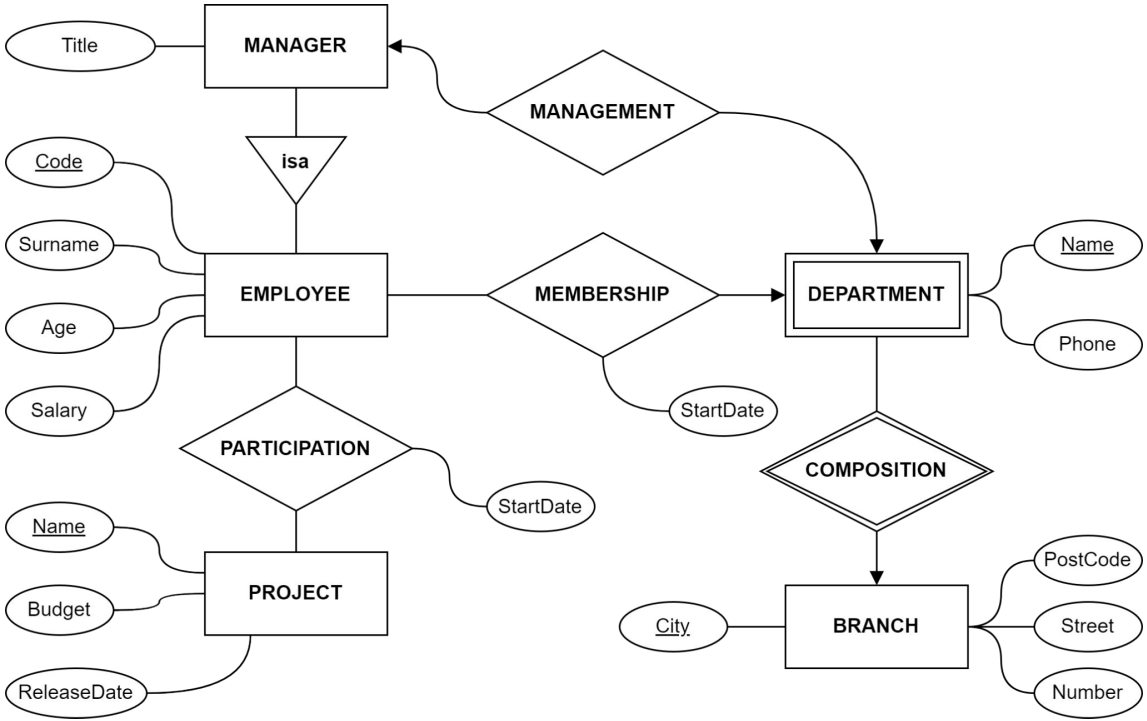


Translate the diagram into a relational schema.

Relationships

1. Participation(Code, Name, StartDate) Code is a foreign key of Employee; Name is a foreign key of Project.
2. Management(Code, Name, City) or Management(Code, Name, City); Code foreign key of Manager; (Name, City) foreign key of Department.
3. Membership(Code, Name, City, StartDate); Code foreign key of Employee; (Name, City) foreign key of Department.

ER Diagram -> Relational Schema

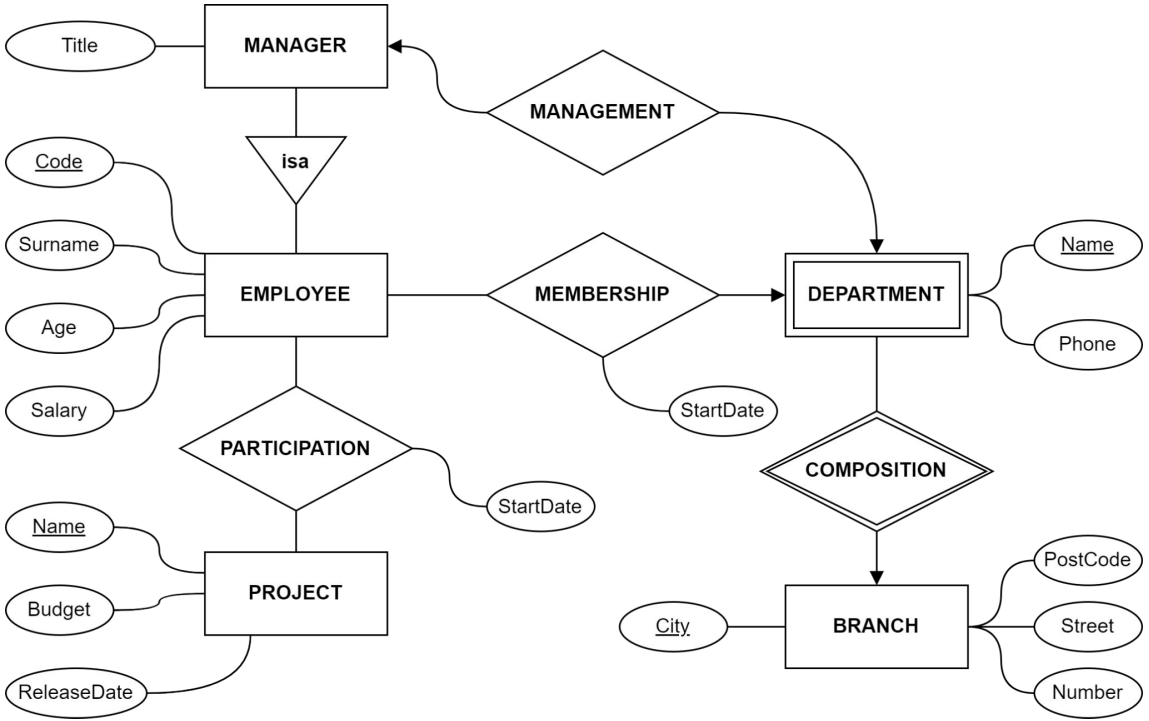


Translate the diagram into a relational schema.

Relationships

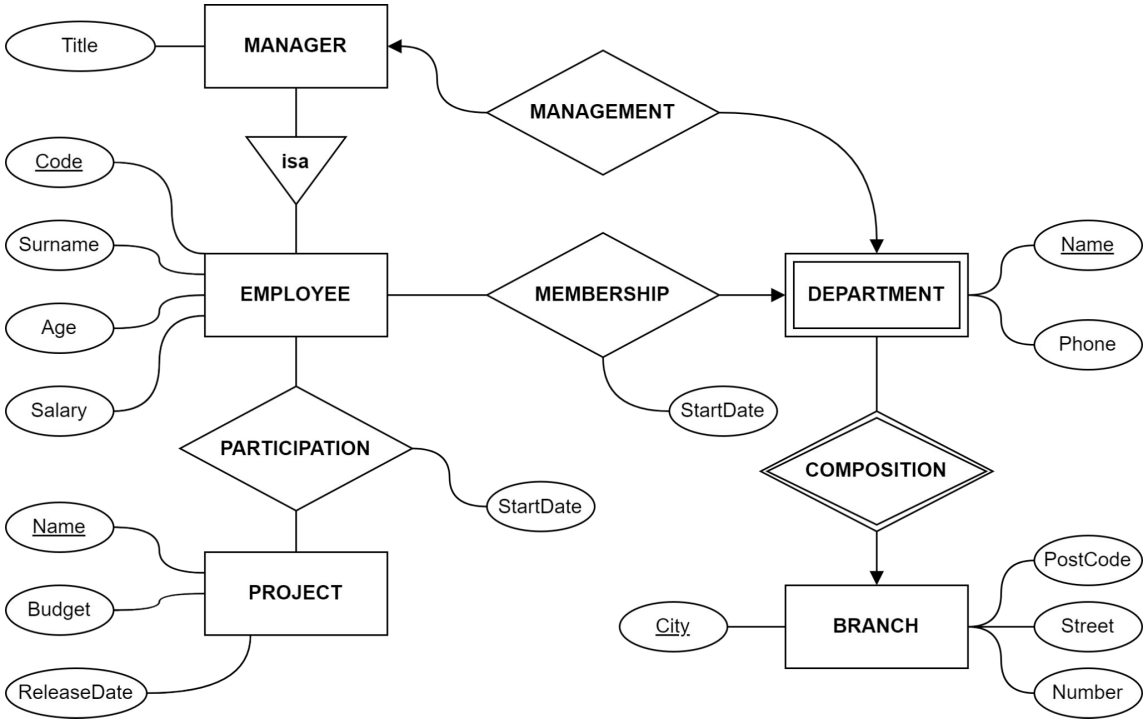
1. Participation(Code, Name, StartDate) Code is a foreign key of Employee; Name is a foreign key of Project.
2. Management(Code, Name, City) or Management(Code, Name, City); Code foreign key of Manager; (Name, City) foreign key of Department.
3. Membership(Code, Name, City, StartDate); Code foreign key of Employee; (Name, City) foreign key of Department.
4. Composition? Collapsed in Management

Collapsing Information



Can we collapse some relationships onto entity sets?

Collapsing Information

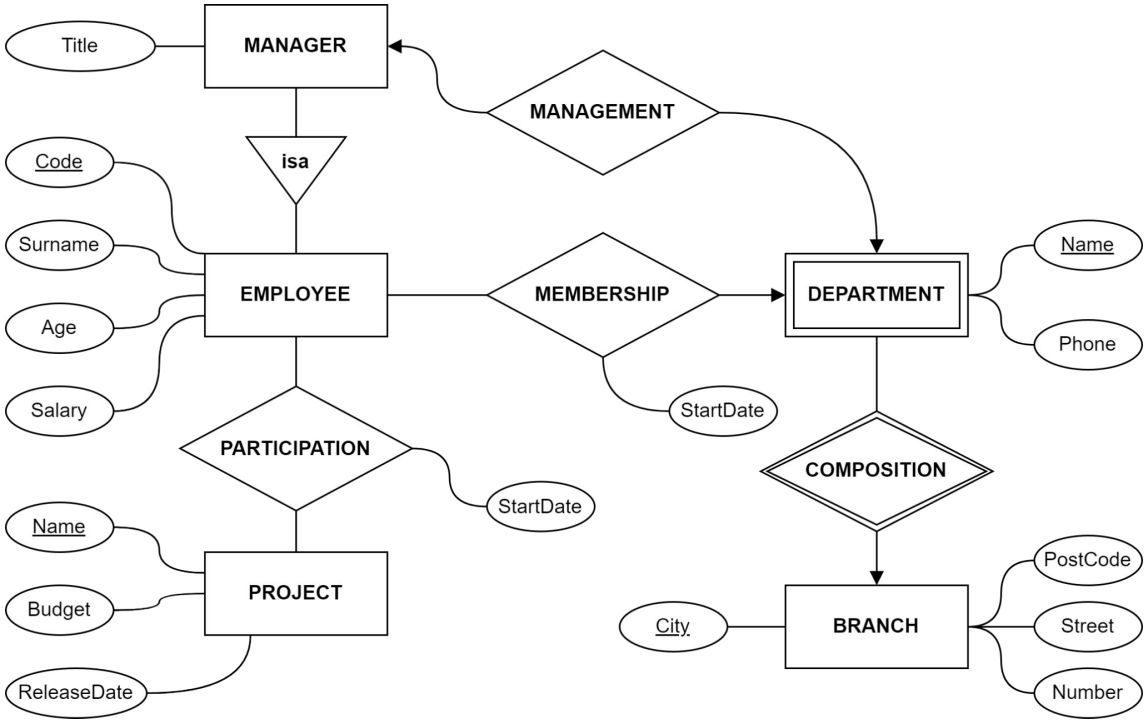


Can we collapse some relationships onto entity sets?

Composition: we have already done that;

Department(Name, City, Phone); City is a foreign key of Branch.

Collapsing Information



Can we collapse some relationships onto entity sets?

Management: one-to-one relationship. Store manager.code at department. *The minimum 1 constraint cannot be expressed (i.e. every dept. Must have a manager). However, we can still enforce it by making managerCode NOT NULL.*