UNIVERSITY OF WISCONSIN–MADISON

CS 564: Database Management Systems

Lecture 7: ER Model and Functional Dependency

Xiangyao Yu

2/7/2024

Announcement

Assignment #1 due today

– Assignment #1 (Grades and solutions will be posted next week)

Assignment #2 posted

– Group assignment. One submission per group.

Module A2: Database Design

ER Model

Functional Dependency

Normalization I

Normalization II

Outline of this Lecture

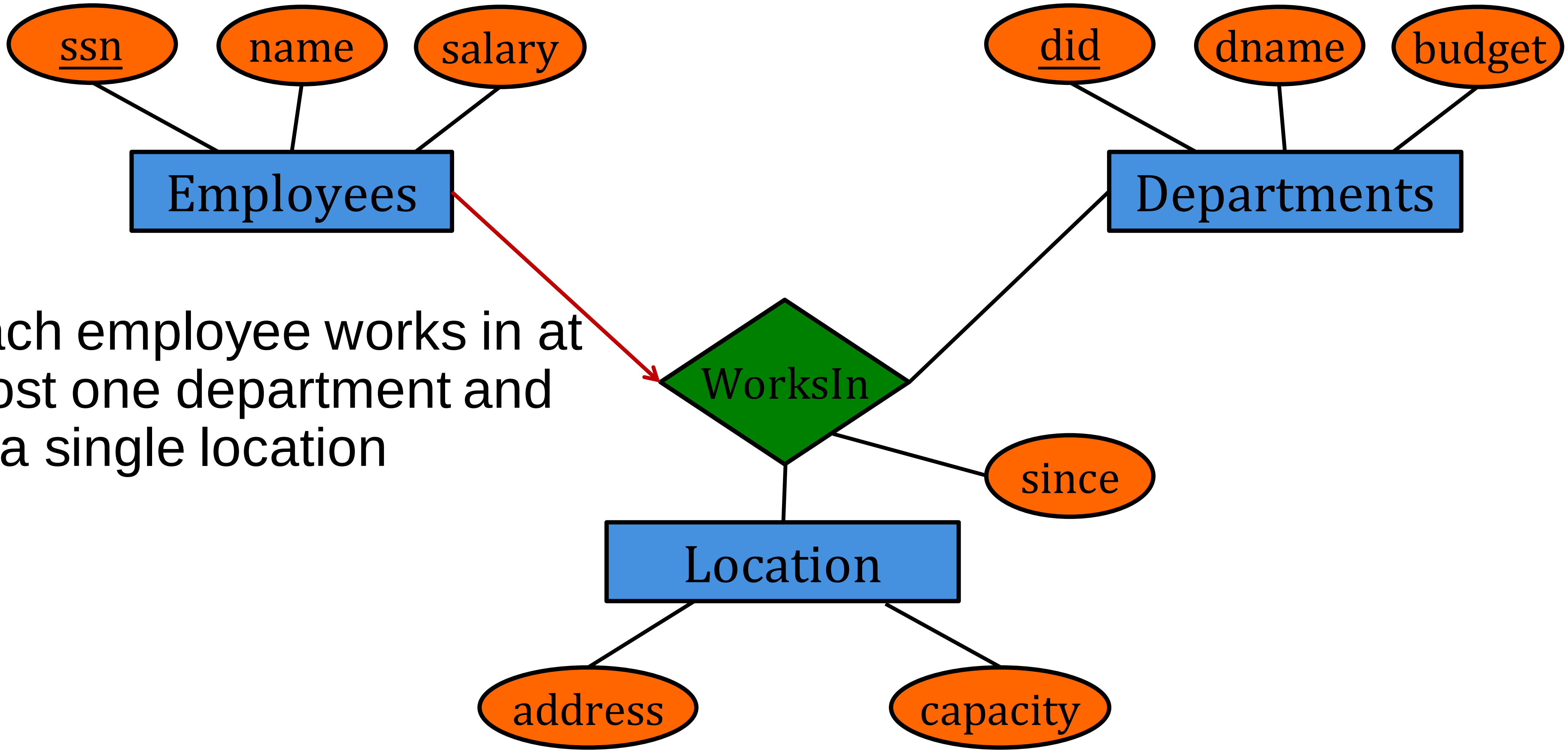
ER Model

- Multi-way relationship
- Roles in relationships
- Weak entities
- Class Hierarchies

Functional Dependency

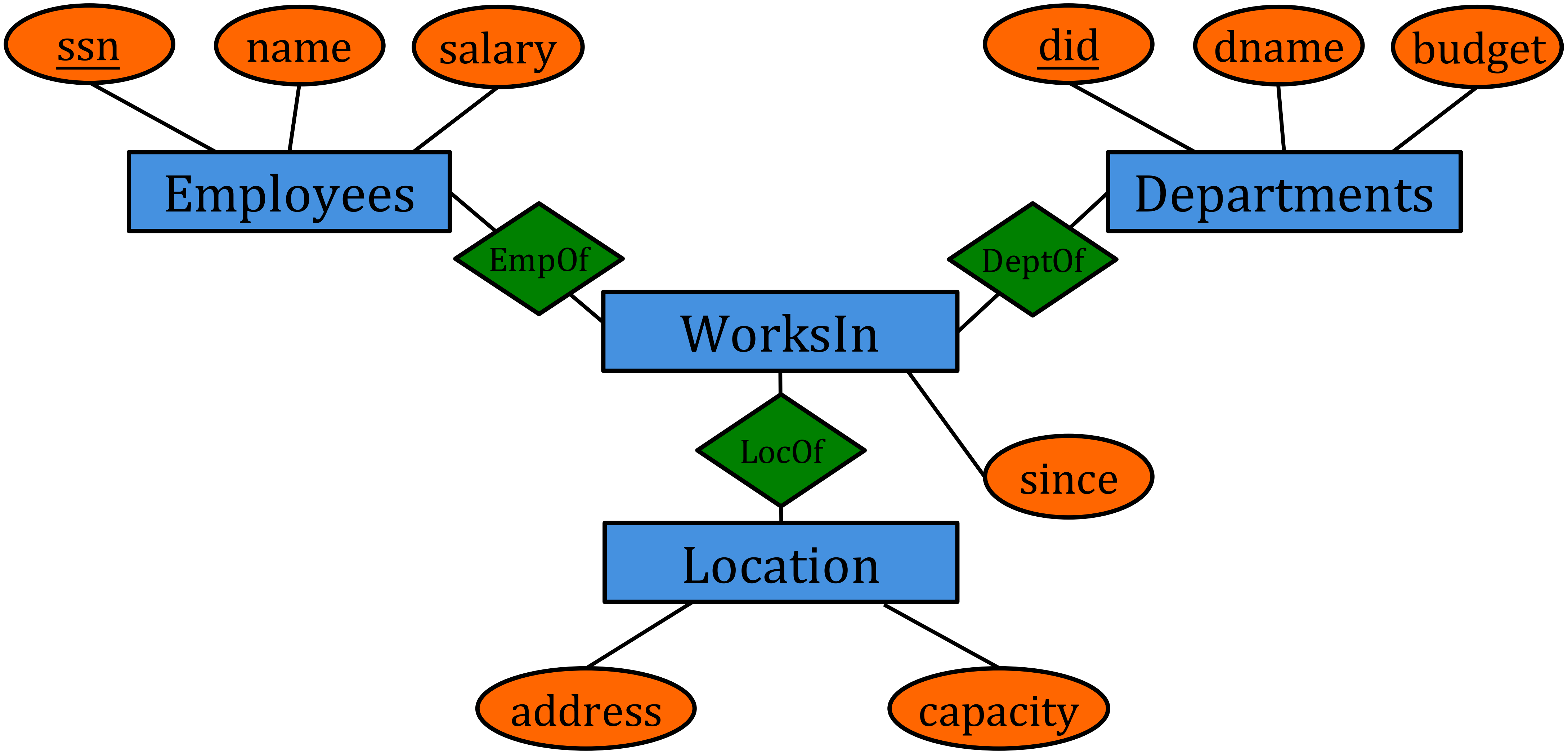
- Functional dependencies
- Armstrong’s rules
- Keys and superkeys

Multi-Way Relationship

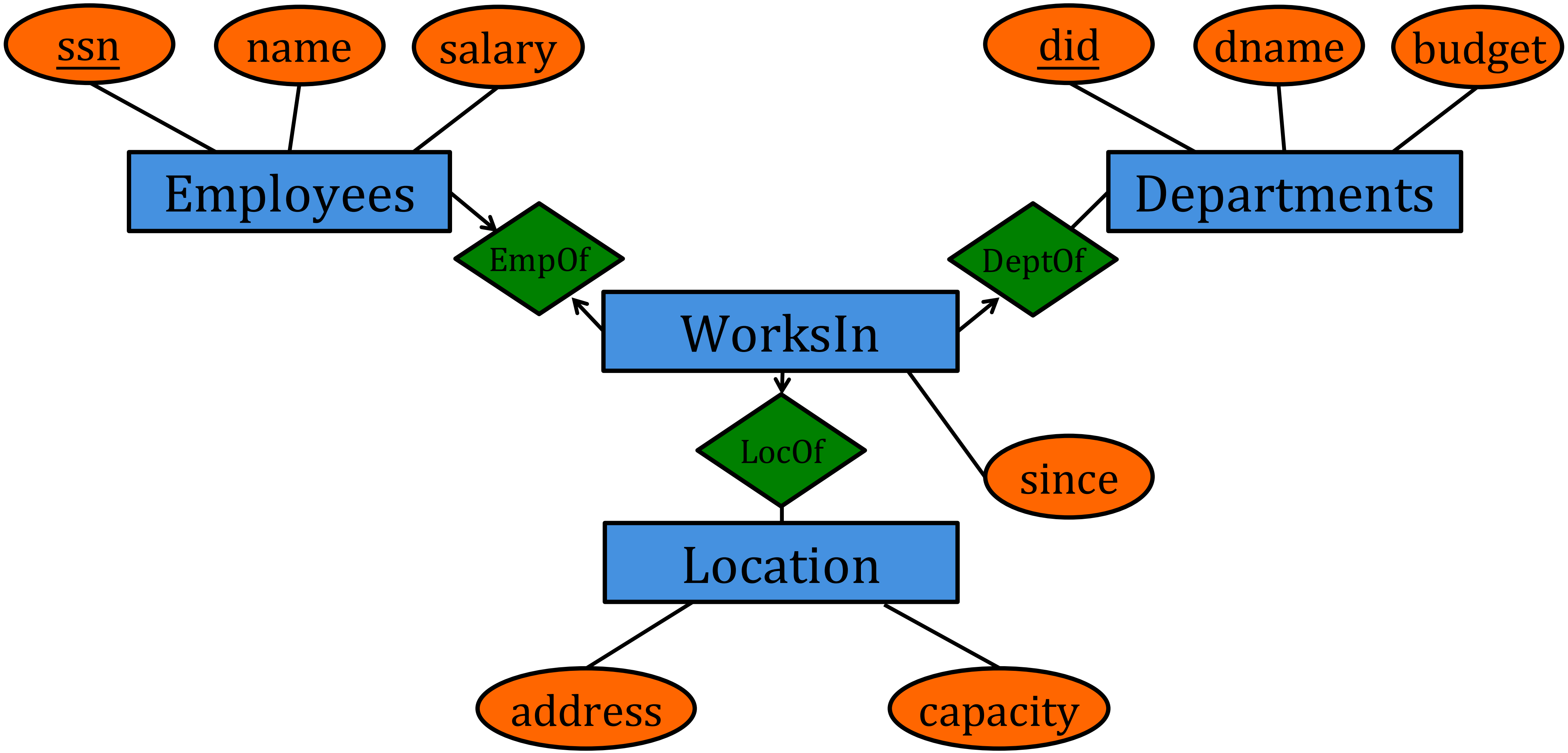


Each employee works in at most one department and at a single location

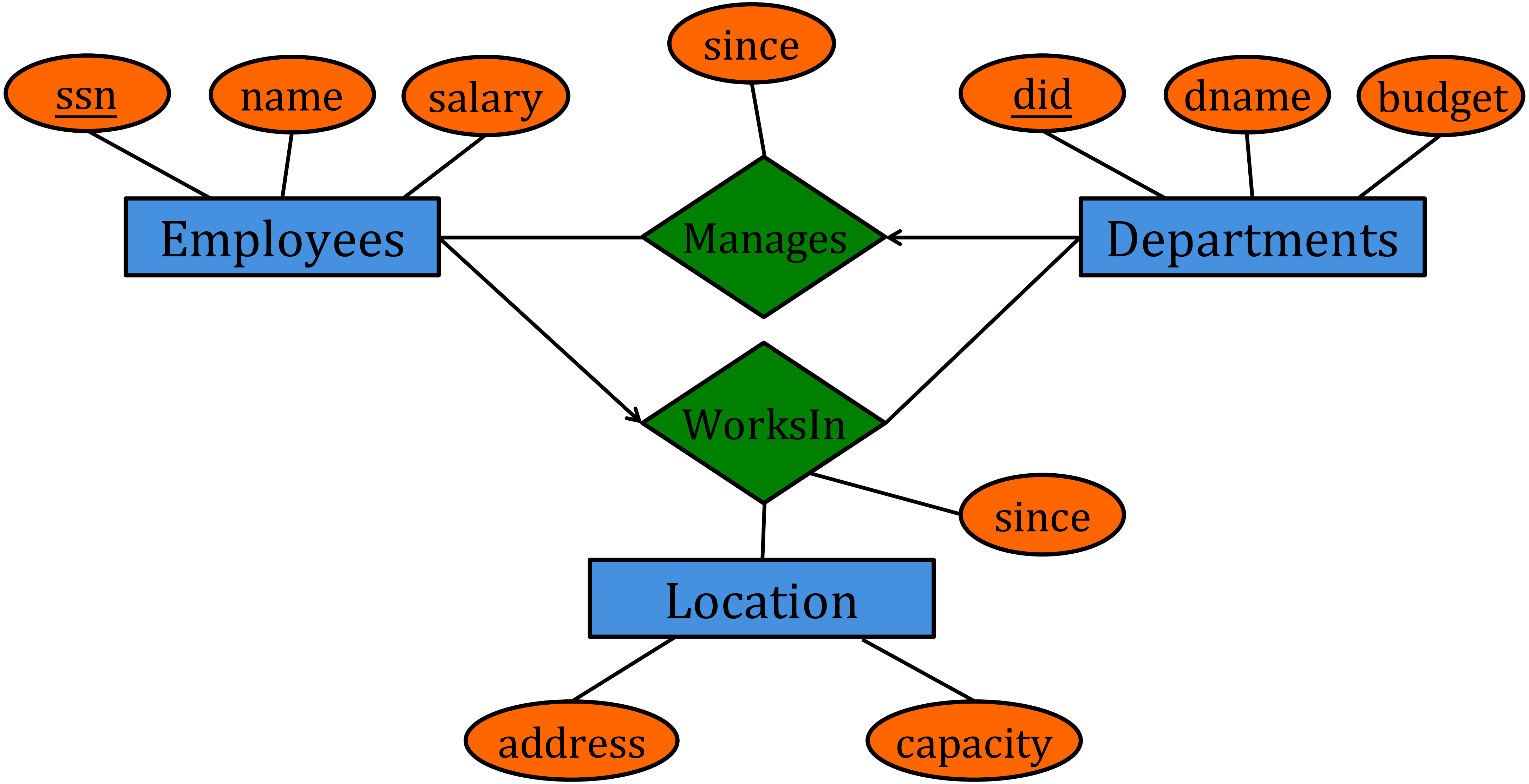
Multi-Way to Binary Relationship



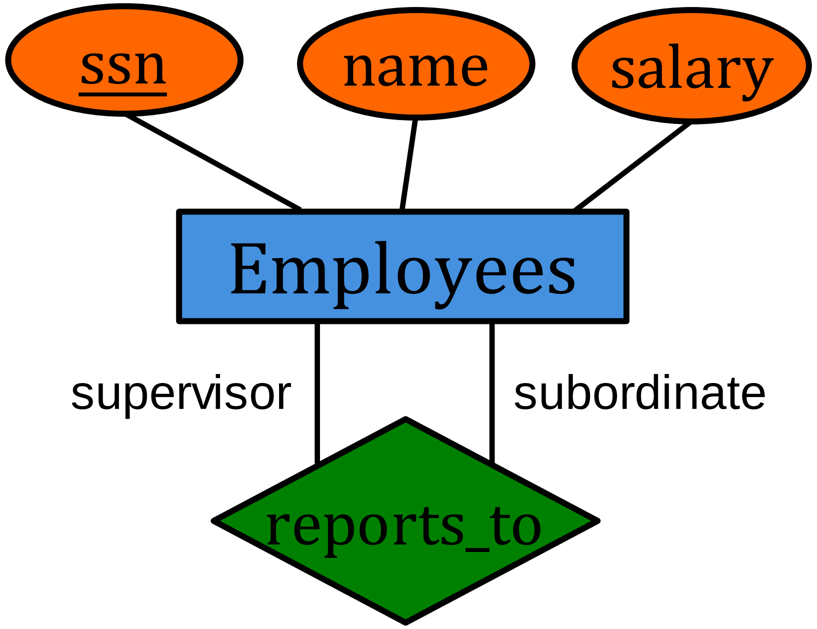
Multi-Way to Binary Relationship



Multiple Relationships

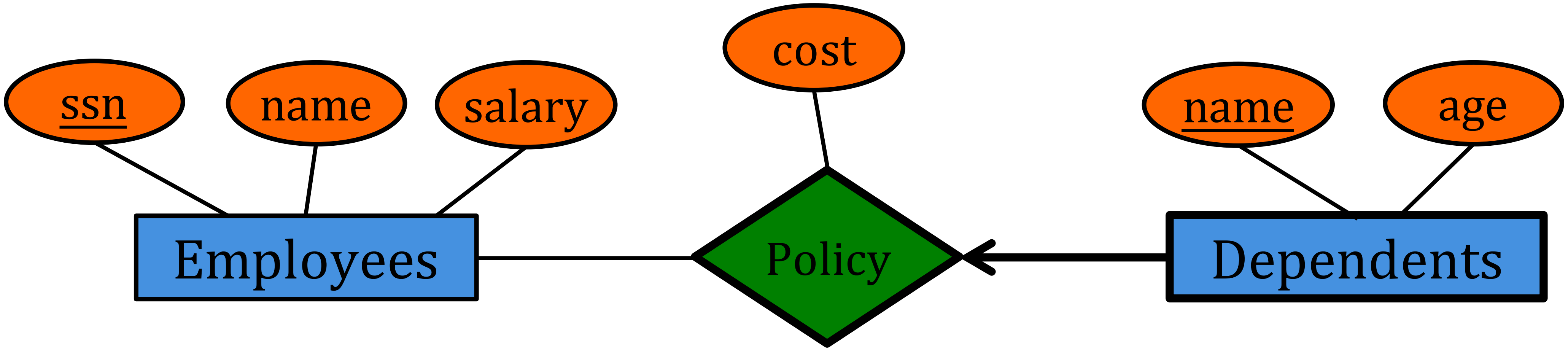


Roles in Relationship



Label the edges to indicate the roles if an entity set plays more than one role

Weak Entities



A **weak entity** can be identified uniquely only by considering some of its attributes in conjunction with the primary key of another entity

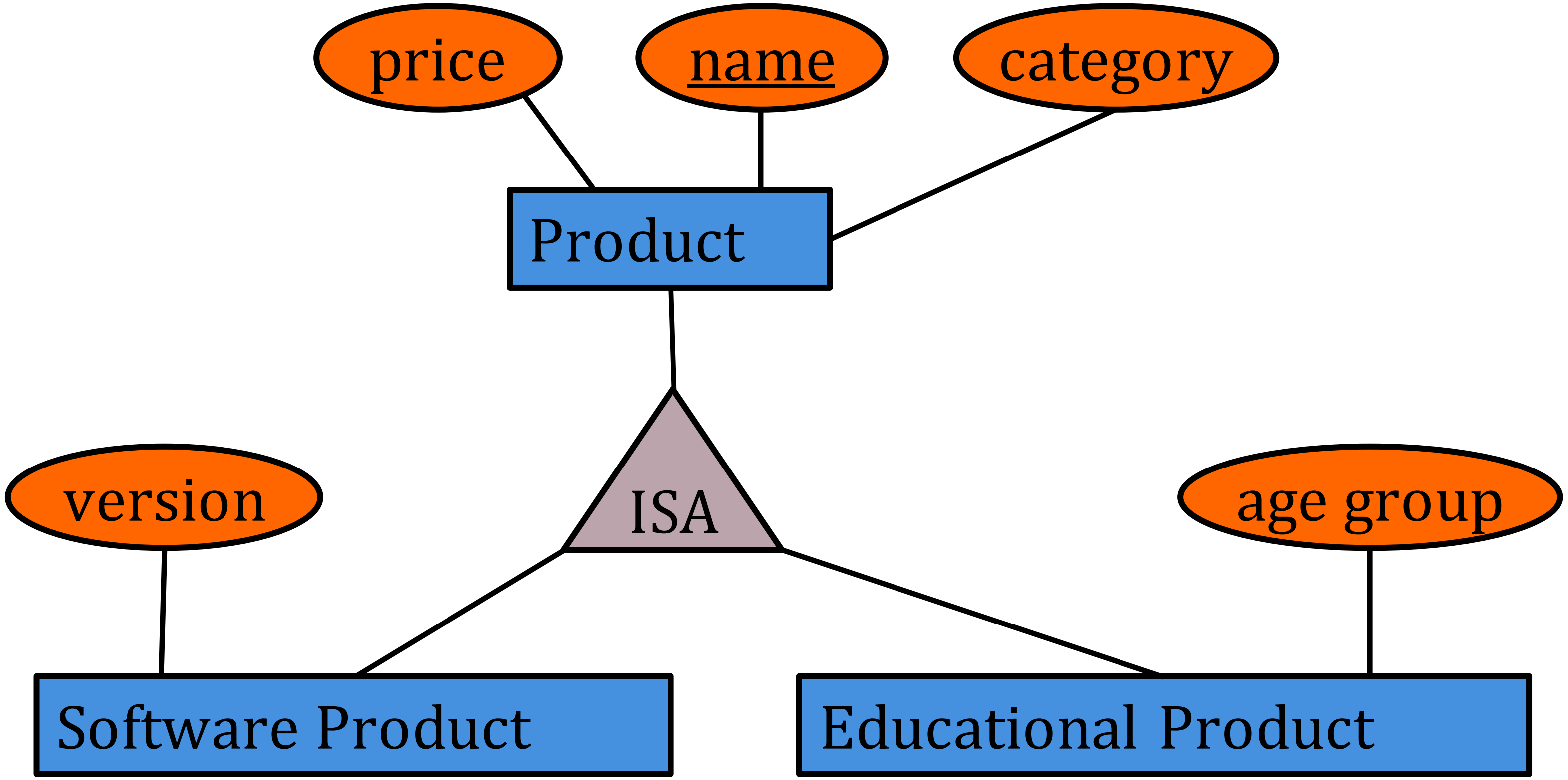
- Owner entity set
- Weak entity set

Restrictions:

- Weak entity set must participate in **one-to-many** relationship set
- Weak entity set must have **total participation**

Weak entity set and the associated relationship are drawn with dark lines

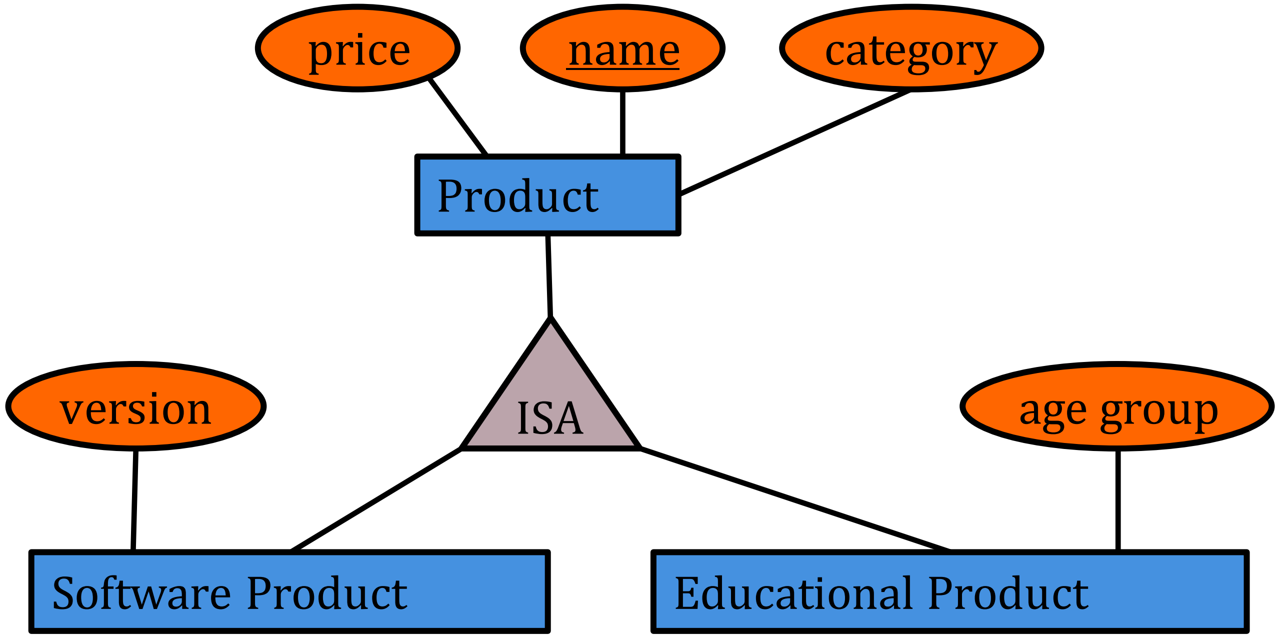
Class Hierarchy



Superclass: Product

Subclass: Software Product and Educational Product

Class Hierarchy -> Schema



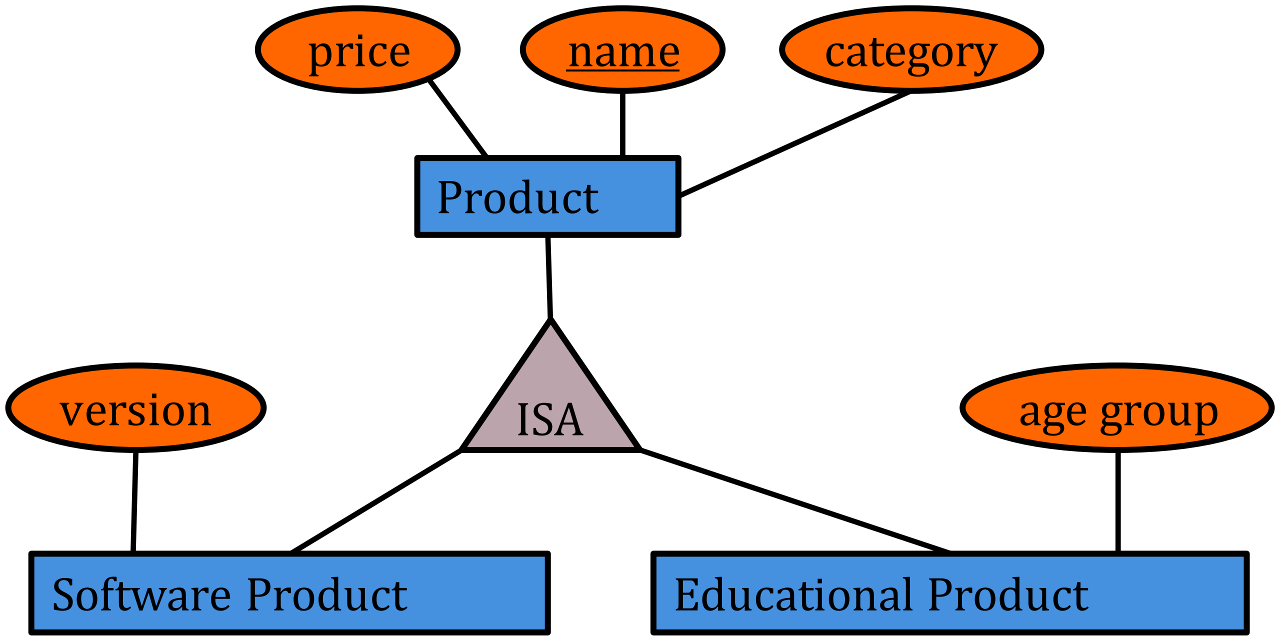
Option 1

Product (<u>name</u>, price, category)

SoftwareProduct (<u>name</u>, price, category, version)

EducationalProduct (<u>name</u>, category, price, age-group)

Class Hierarchy -> Schema



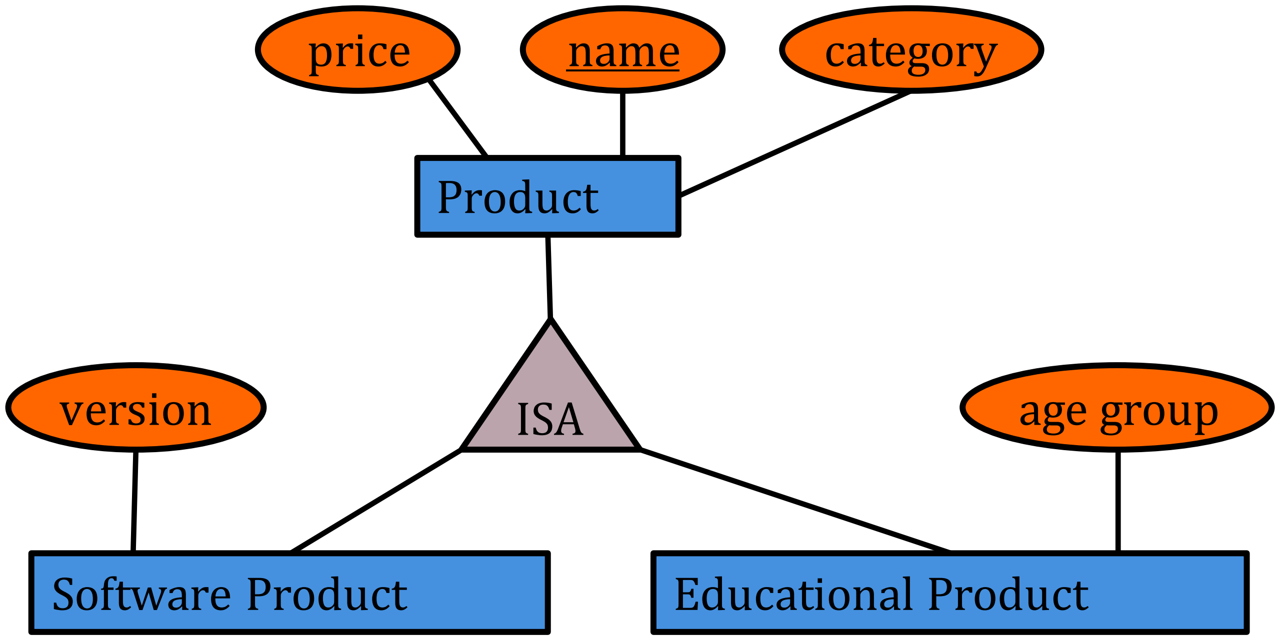
Option 2

Product (<u>name</u>, price, category)

SoftwareProduct (<u>name</u>, version)

EducationalProduct (<u>name</u>, age-group)

Class Hierarchy -> Schema



Option 3

Product (<u>name</u>, price, category, version, age-group)

- Use NULL to denote that the attribute makes no sense for a specific tuple

Outline of this Lecture

ER Model

- Multi-way relationship
- Roles in relationships
- Weak entities
- Class Hierarchies

Functional Dependency

- **Functional dependencies**
- Armstrong's rules
- Keys and superkeys

Motivation

SSN	**name**	**rating**	**hourly_wages**	**hours_worked**
123-22-3666	Attishoo	8	10	40
231-31-5368	Smiley	8	10	30
131-24-3650	Smethurst	5	7	30
434-26-3751	Guldu	5	7	32
612-67-4134	Madayan	8	10	40

Functional Dependency (FD)

- rating determines hourly_wages
- If two tuples have the same rating, they must have the same hourly_wages
- This leads to possible redundancy: rating=8 corresponding to hourly wage=10 is repeated three times

Solution: Decomposition

SSN	name	rating	hourly_wages	hours_worked
123-22-3666	Attishoo	8	10	40
231-31-5368	Smiley	8	10	30
131-24-3650	Smethurst	5	7	30
434-26-3751	Guldu	5	7	32
612-67-4134	Madayan	8	10	40

SSN	name	rating	hours_worked
123-22-3666	Attishoo	8	40
231-31-5368	Smiley	8	30
131-24-3650	Smethurst	5	30
434-26-3751	Guldu	5	32
612-67-4134	Madayan	8	40

rating	hourly_wages
8	10
5	7

Solution: Decomposition

<u>SSN</u>	name	rating	hourly_wages	hours_worked
123-22-3666	Attishoo	8	10	40
231-31-5368	Smiley	8	10	30
131-24-3650	Smethurst	5	7	30
434-26-3751	Guldu	5	7	32
612-67-4134	Madayan	8	10	40

<u>SSN</u>	name	rating	hours_worked
123-22-3666	Attishoo	8	40
231-31-5368	Smiley	8	30
131-24-3650	Smethurst	5	30
434-26-3751	Guldu	5	32
612-67-4134	Madayan	8	40

rating	hourly_wages
8	10
5	7

More on decomposition next week!

Functional Dependency (FD)

Functional dependencies (FDs) are a form of constraint;

they generalize the concept of keys

> If two tuples agree on the attributes
>
> $$A = A_1, A_2, \ldots, A_n$$
>
> then they must agree on the attributes
>
> $$B = B_1, B_2, \ldots, B_m$$
>
> Formally:
>
> $$A_1, A_2, \ldots, A_n \longrightarrow B_1, B_2, \ldots, B_m$$
>
> We then say that *A* **functionally determines** *B*

FDs help us detect redundancy in a schema and tell us how to normalize it

Identifying FDs

An FD is domain knowledge:

- An inherent property of the application & data
- Not something we can infer from a set of tuples

Given a table with a set of tuples

- We can confirm that a FD **seems** to be valid
- To infer that a FD is **definitely** invalid
- We can never prove that a FD is valid

Example 1

<u>**SSN**</u>	**name**	**rating**	**hourly_wages**	**hours_worked**
123-22-3666	Attishoo	8	10	40
231-31-5368	Smiley	8	10	30
131-24-3650	Smethurst	5	7	30
434-26-3751	Guldu	5	7	32
612-67-4134	Madayan	8	10	40

rating ⟶ hourly_wages

SSN ⟶ name, rating, hourly_wages, hours_worked

Example 2

Given a particular relation:

name	category	color	department	price
Gizmo	Gadget	Green	Toys	49
Tweaker	Gadget	Black	Toys	99
Gizmo	Stationary	Green	Office-supplies	59

Q1: Is $name \longrightarrow department$ an FD?

- Not possible!

Example 2

Given a particular relation:

name	category	color	department	price
Gizmo	Gadget	Green	Toys	49
Tweaker	Gadget	Black	Toys	99
Gizmo	Stationary	Green	Office-supplies	59

Q1: Is $name \longrightarrow department$ an FD?

– Not possible!

Q2: Is $name, category \longrightarrow department$ an FD ?

– We don't know!

Outline of this Lecture

- ER Model
  - Multi-way relationship
  - Roles in relationships
  - Weak entities
  - Class Hierarchies
- Functional Dependency
  - Functional dependencies
  - **Armstrong’s rules**
  - Keys and superkeys

Armstrong's Axioms: 1

Reflexivity

For any subset $X \subseteq \{A_1, .., A_n\}$:

$$A_1, A_2, \ldots, A_n \longrightarrow X$$

Examples

- $A \longrightarrow A$
- $A, B \longrightarrow B$
- $A, B, C \longrightarrow A, B$
- $A, B, C \longrightarrow A, B, C$

Armstrong's Axioms: 2

<u>Augmentation</u>
For any attribute sets *X, Y, Z* :

if $X \longrightarrow Y$ then $X, Z \longrightarrow Y, Z$

Examples

– $A \longrightarrow B$ implies $A, C \longrightarrow B, C$
– $A, B \longrightarrow C$ implies $A, B, C \longrightarrow C$

Armstrong's Axioms: 3

Transitivity

For any attribute sets *X, Y, Z* :

if $X \longrightarrow Y$ and $Y \longrightarrow Z$ then $X \longrightarrow Z$

Examples

- $A \longrightarrow B$ and $B \longrightarrow C$ imply $A \longrightarrow C$
- $A \longrightarrow C, D$ and $C, D \longrightarrow E$ imply $A \longrightarrow E$

Applying Armstrong's Axioms

Product(name, category, color, department, price)

1. $name \longrightarrow color$
2. $category \longrightarrow department$
3. $color, category \longrightarrow price$

Can we infer $name, category \longrightarrow price$?

Applying Armstrong’s Axioms

Product(name, category, color, department, price)

1. $name \longrightarrow color$
2. $category \longrightarrow department$
3. $color, category \longrightarrow price$

Can we infer $name, category \longrightarrow price$?

1. We apply the augmentation axiom to (1) to obtain
   (4) $name, category \longrightarrow color, category$
2. We apply the transitivity axiom to (4) and (3) to obtain $name, category \longrightarrow price$

Applying Armstrong's Axioms

Product(name, category, color, department, price)

1. $name \longrightarrow color$
2. $category \longrightarrow department$
3. $color, category \longrightarrow price$

Can we infer $name, category \longrightarrow color$?

Applying Armstrong's Axioms

Product(name, category, color, department, price)

1. $name \longrightarrow color$
2. $category \longrightarrow department$
3. $color, category \longrightarrow price$

Can we infer $name, category \longrightarrow color$?

1. We apply the reflexivity axiom to obtain
   (5) $name, category \longrightarrow name$
2. We apply the transitivity axiom to (5), (1) to obtain
   $name, category \longrightarrow color$

Outline of this Lecture

ER Model

- Multi-way relationship
- Roles in relationships
- Weak entities
- Class Hierarchies

Functional Dependency

- Functional dependencies
- Armstrong’s rules
- **Keys and superkeys**

Keys and Superkeys

Superkey: a set of attributes $A_1, A_2, \ldots, A_n$ such that for any other attribute *B* in the relation:

$$A_1, A_2, \ldots, A_n \longrightarrow B$$

key: a minimal superkey

- None of its subsets functionally determines all attributes of the relation

If a relation has multiple keys, we specify one to be the **primary key**

Keys and Superkey – Example

Product(name, category, price, color)

- $name \longrightarrow color$
- $color, category \longrightarrow price$

Superkeys:

Keys and Superkey – Example

Product(name, category, price, color)

- $name \longrightarrow color$
- $color, category \longrightarrow price$

Superkeys:

- $\{name, category\}, \{name, category, price\},$
  $\{name, category, color\}, \{name, category, price, color\}$

Keys and Superkey – Example

Product(name, category, price, color)

- $name \longrightarrow color$
- $color, category \longrightarrow price$

Superkeys:

- $\{name, category\}, \{name, category, price\},$
  $\{name, category, color\}, \{name, category, price, color\}$

Keys:

Keys and Superkey – Example

Product(name, category, price, color)

- $name \longrightarrow color$
- $color, category \longrightarrow price$

Superkeys:

- $\{name, category\}, \{name, category, price\},$
  $\{name, category, color\}, \{name, category, price, color\}$

Keys:

- $\{name, category\}$

Multiple Keys

Q: Is it possible to have many keys in a relation **R** ?

YES!! Take relation **R**(A, B, C) with FDs

- $A, B \longrightarrow C$
- $A, C \longrightarrow B$

Friday Lecture

TA Sweksha Sinha will go through Assignment #2

Summary

ER Model

- Multi-way relationship
- Roles in relationships
- Weak entities
- Class Hierarchies

Functional Dependency

- Functional dependencies
- Armstrong's rules
- Keys and superkeys