CS 564: Database Management Systems

Lecture 6: ER Model

Xiangyao Yu

2/5/2024

Announcement

Group selection **(Due Today, 11:59pm)**

- Canvas -> People -> Project Groups
- Randomly assigned if no group selected

Assignment #1 **(Due Wednesday, 11:59pm)**

- Submit using *submission_template.txt*
- Carefully read the instructions in the submission file
- E.g., do not include the initial *%sql* or *%%sql* in your submission

Module A2: Database Design

ER Model

Functional Dependency

Normalization

Outline of this Lecture

Database design steps

ER model basics

- Attributes, Entity, and relationship
- Multi-way relationship
- Key constraints
- Participation constraint

Database Design

- Requirements Analysis
- Conceptual Database Design (This week)
- Logical Database Design (This week)
- Schema Refinement (Next week)
- Physical Database Design (Module B2)
- Application and Security Design

Entity-Relationship (ER) Model

A visual language to specify

- What information the DB must hold
- What are the relationships among components of that information

Proposed by Peter Chen in 1976

- In contrast, relational model was proposed in 1969

ER Model Basics

Entity

- Distinguishable real-world object

ER Model Basics

Entity

- Distinguishable real-world object

Entity Set

- A collection of similar entities
- All entities in an entity set have the same set of attributes. (Until we consider ISA hierarchies!)
- represented by rectangles

ER Model Basics

Entity

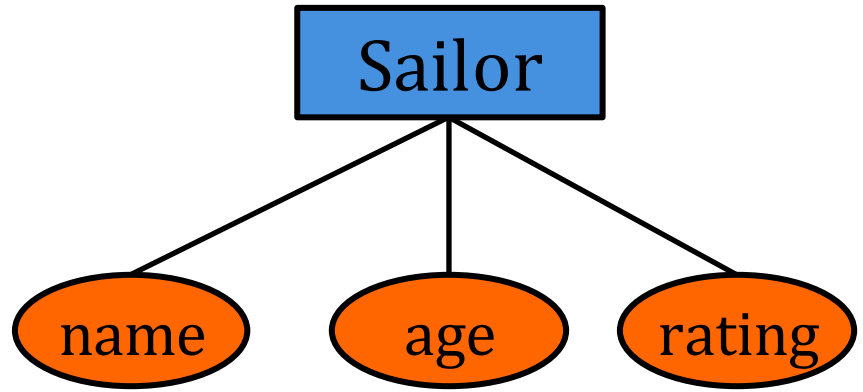
- Distinguishable real-world object

Entity Set

- A collection of similar entities
- All entities in an entity set have the same set of attributes. (Until we consider ISA hierarchies!)
- represented by rectangles

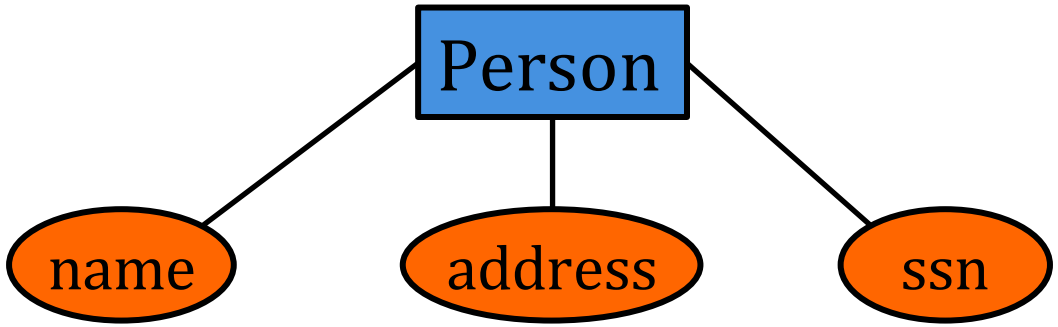
Attribute

- represented by ovals attached to an entity set



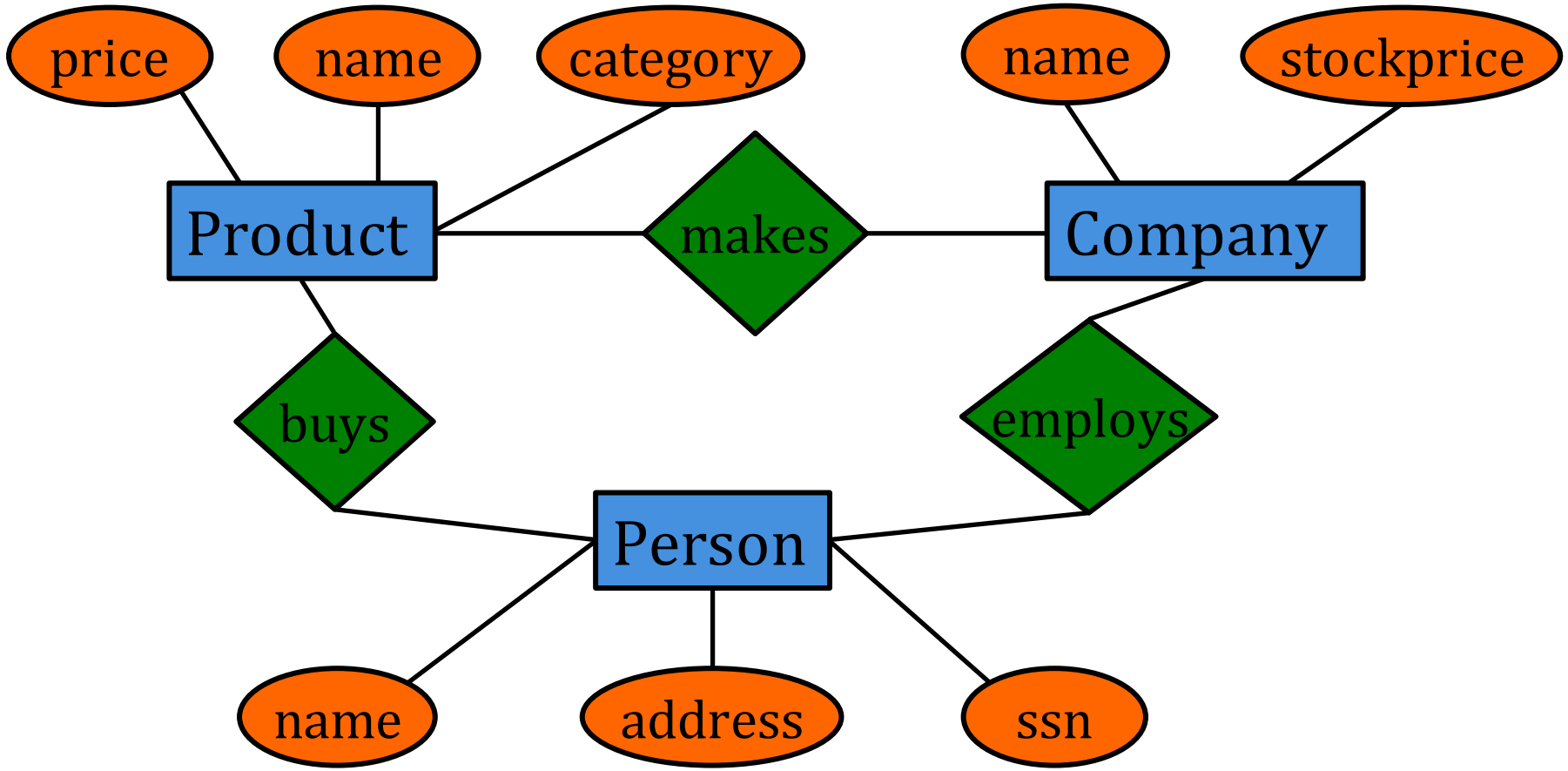
Entity Sets and Attributes

Entities are not explicitly represented in E/R diagrams!



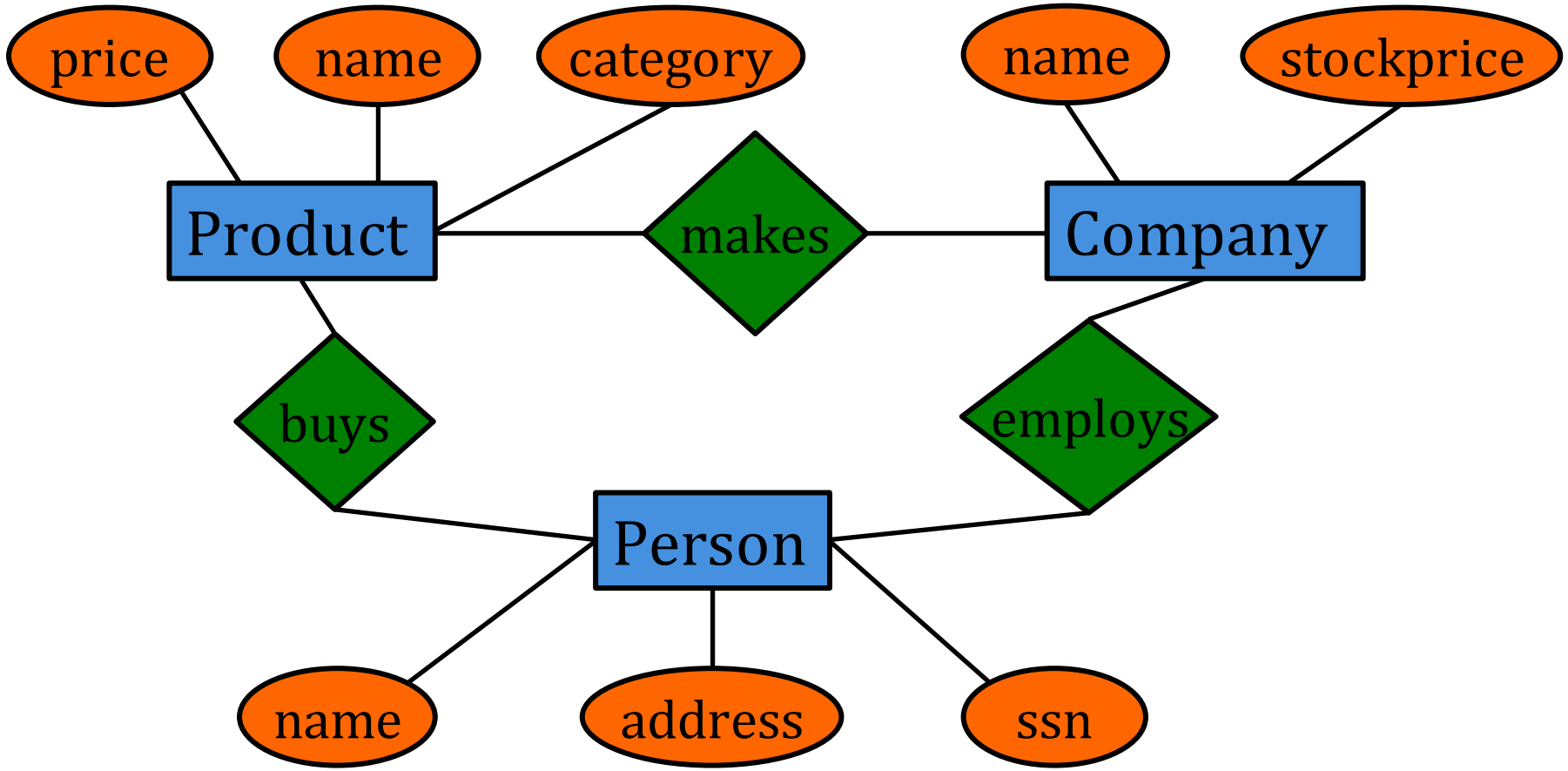
Intuitively, an **entity set maps to a relation**, an **attribute maps to a table attribute**

Relationship



Relationship represented by diamonds between entity sets

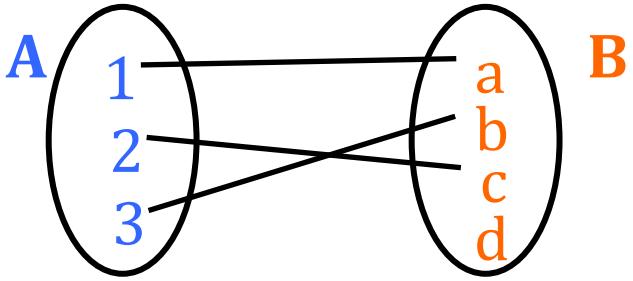
Relationship

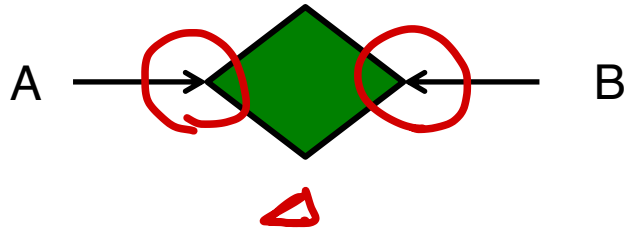


Relationship represented by diamonds between entity sets
What should a relationship map to? (A table? An attribute?)

Key Constraints

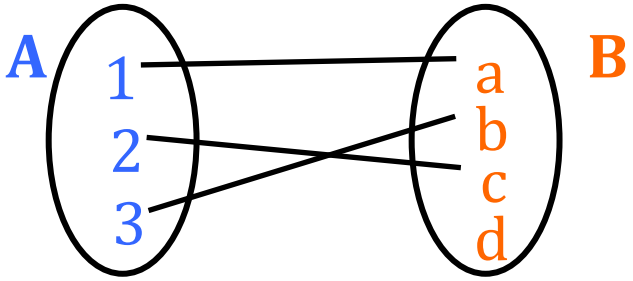
- one-one

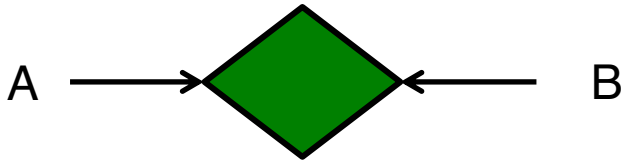




Key Constraints

- one-one



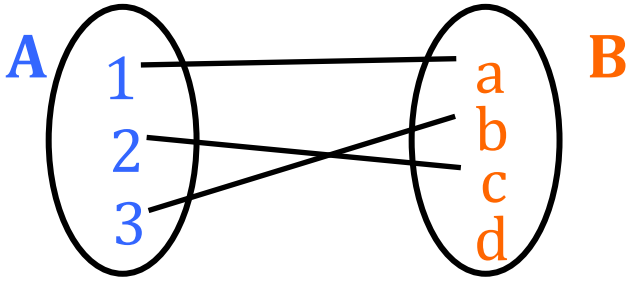


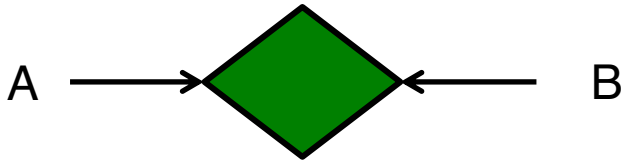
A and B are both keys to the relationship

- Given an entity in A (or B), we can uniquely determine the relationship

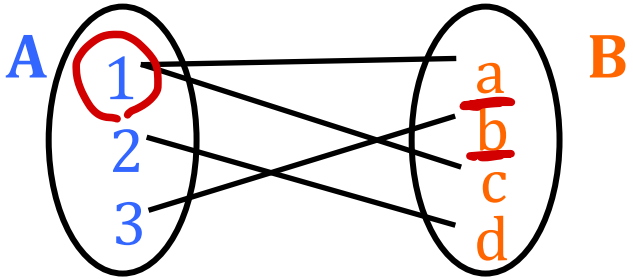
Key Constraints

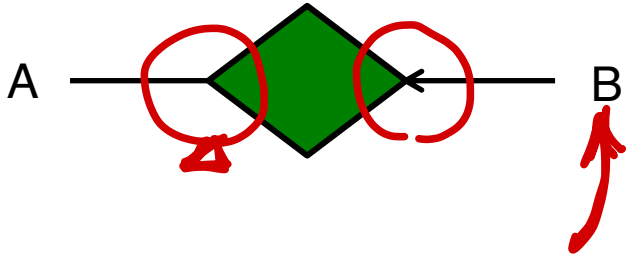
- one-one





- many-one



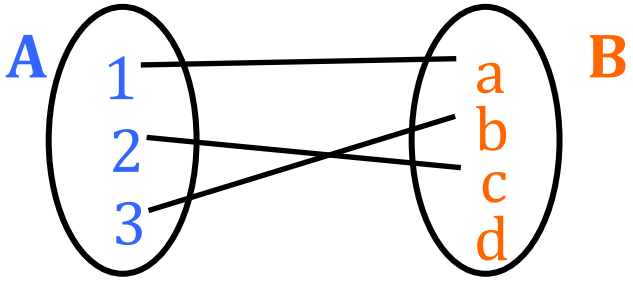


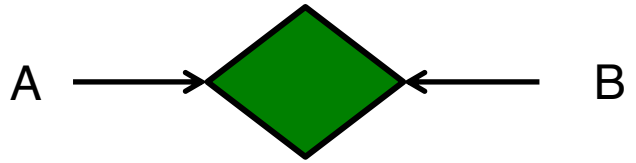
B is a key to the relationship

– Given an entity in B, we can uniquely determine the relationship

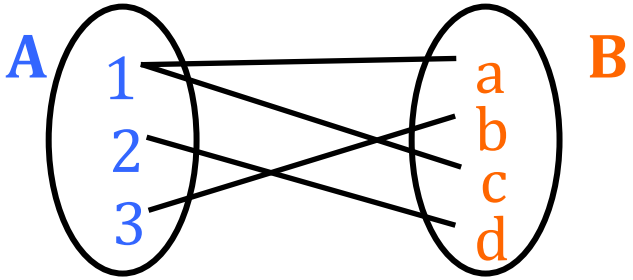
Key Constraints

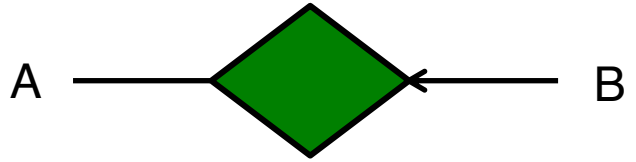
- one-one



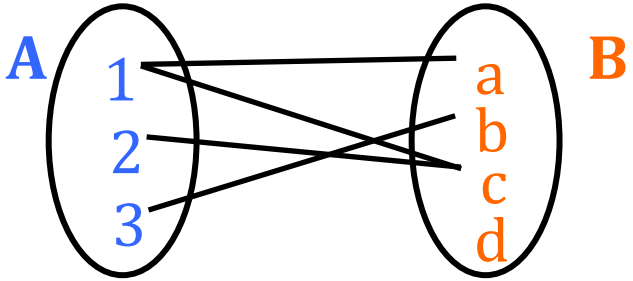


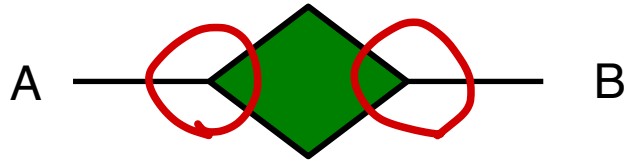
- many-one



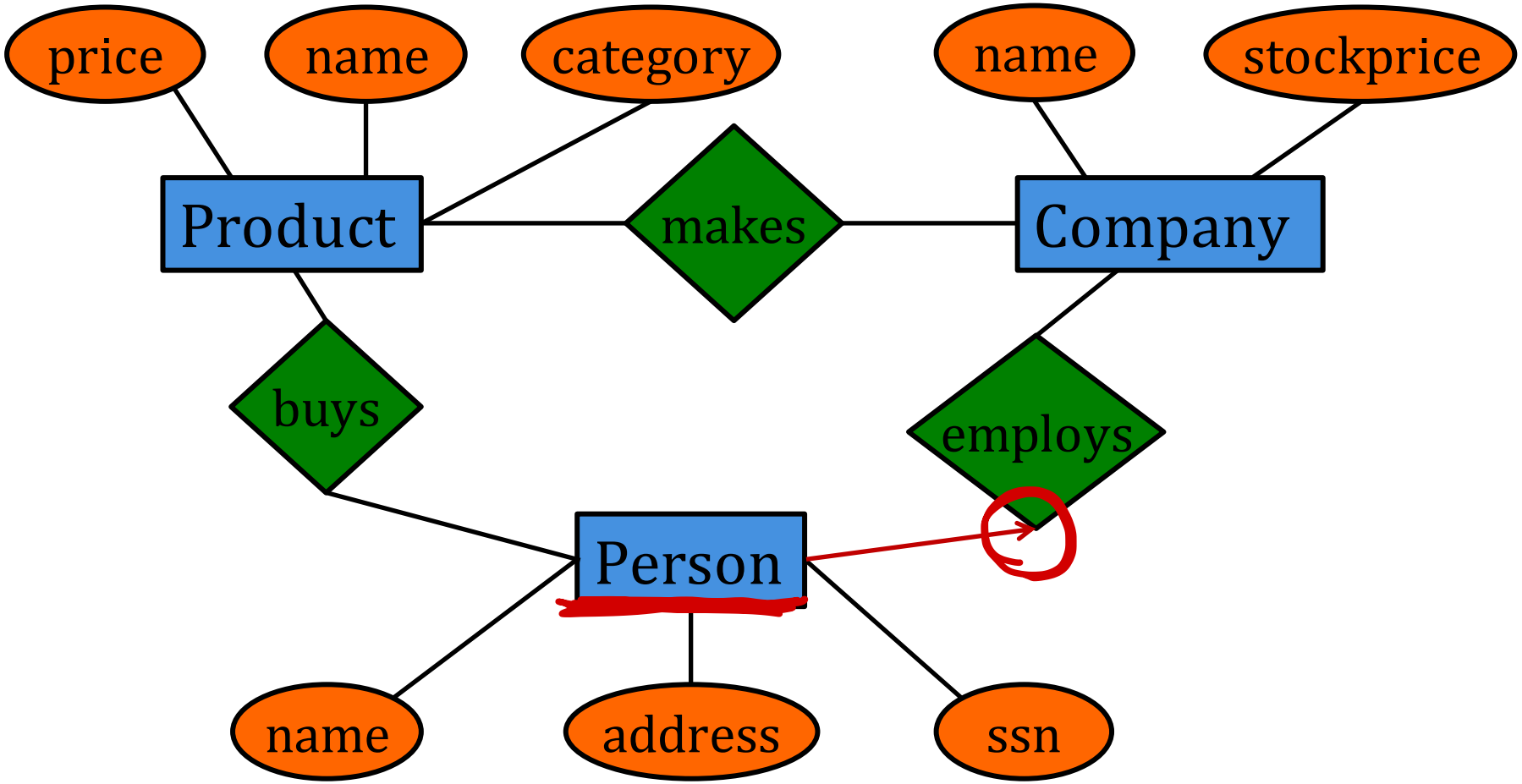


- many-many

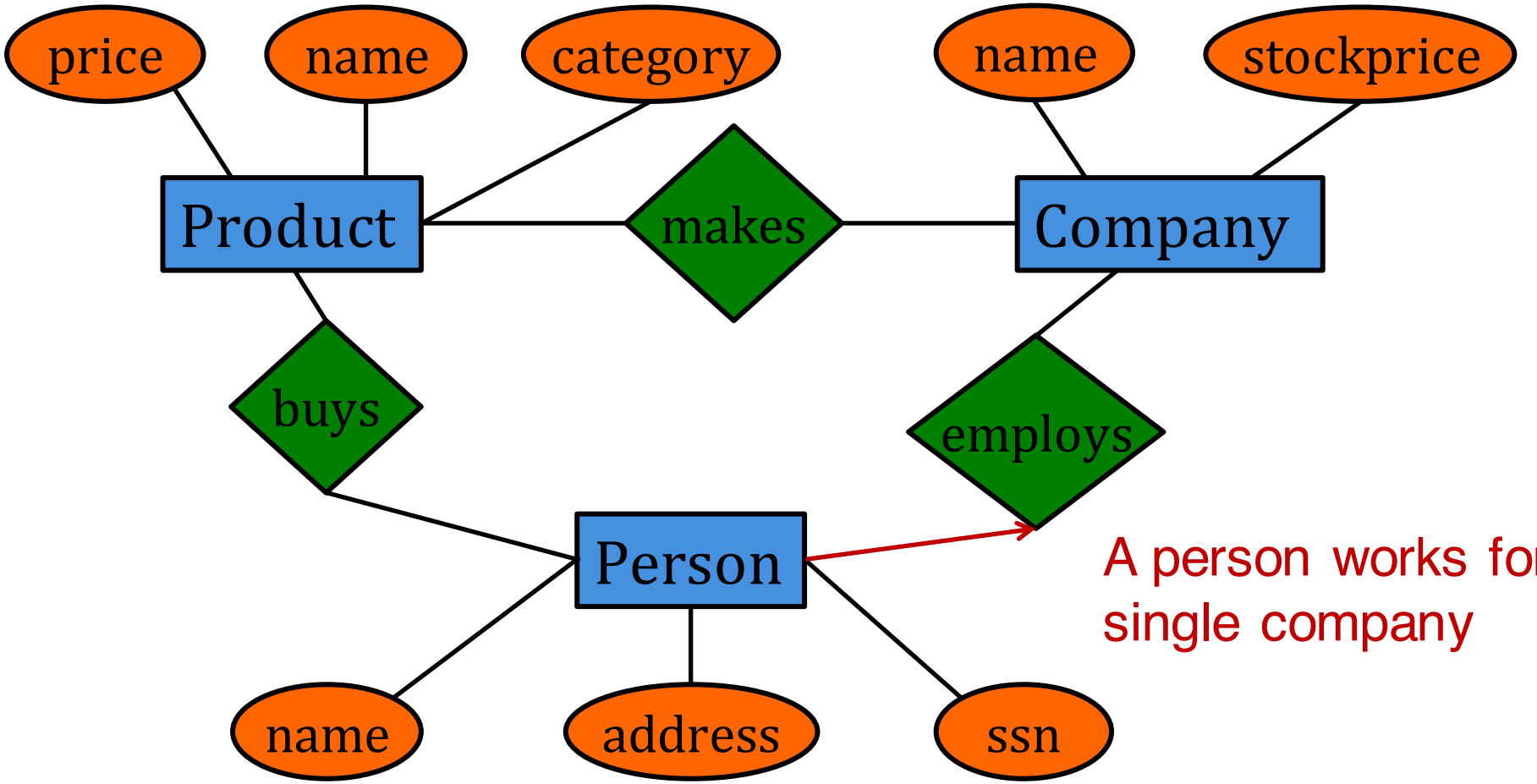




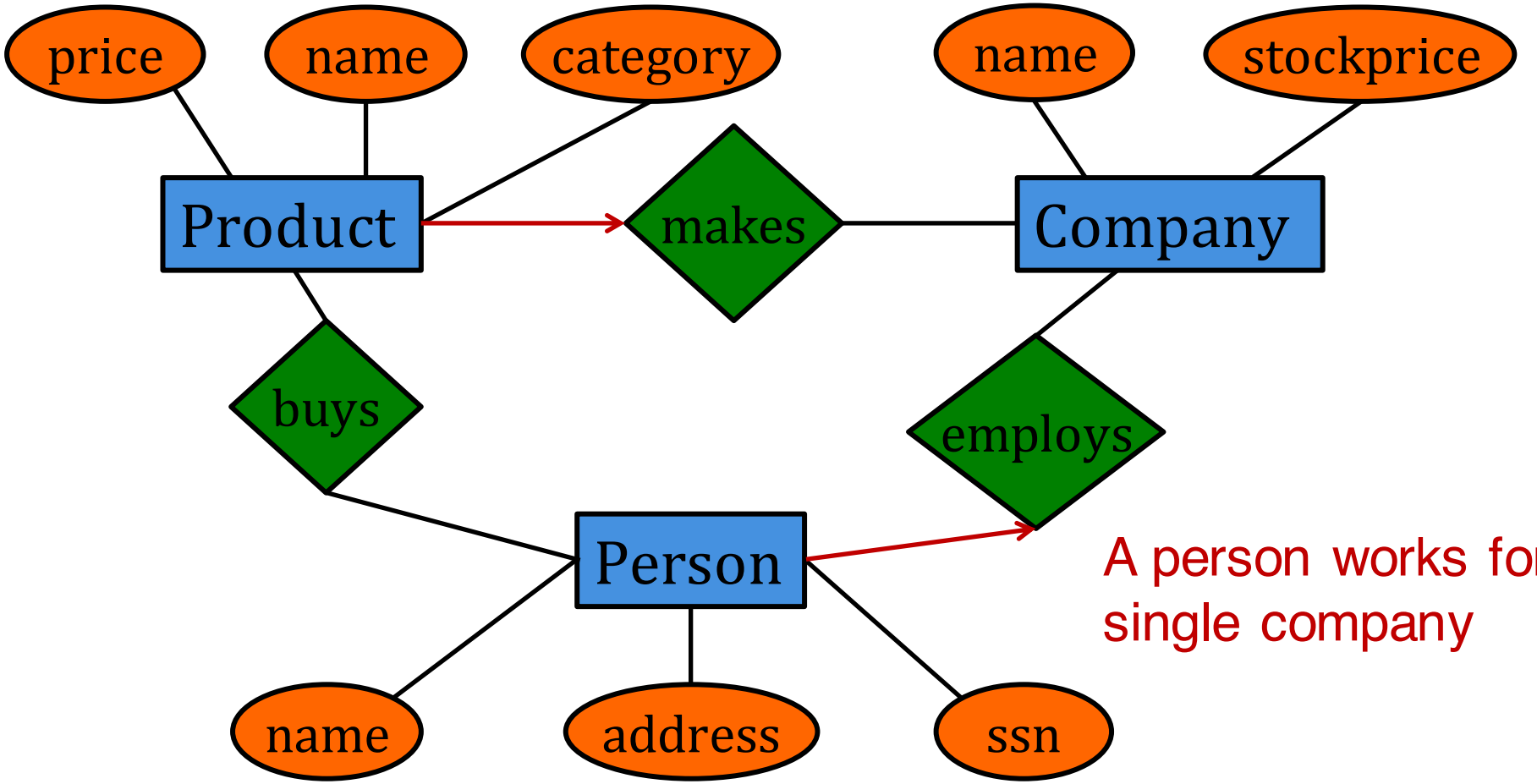
Key Constraints Example



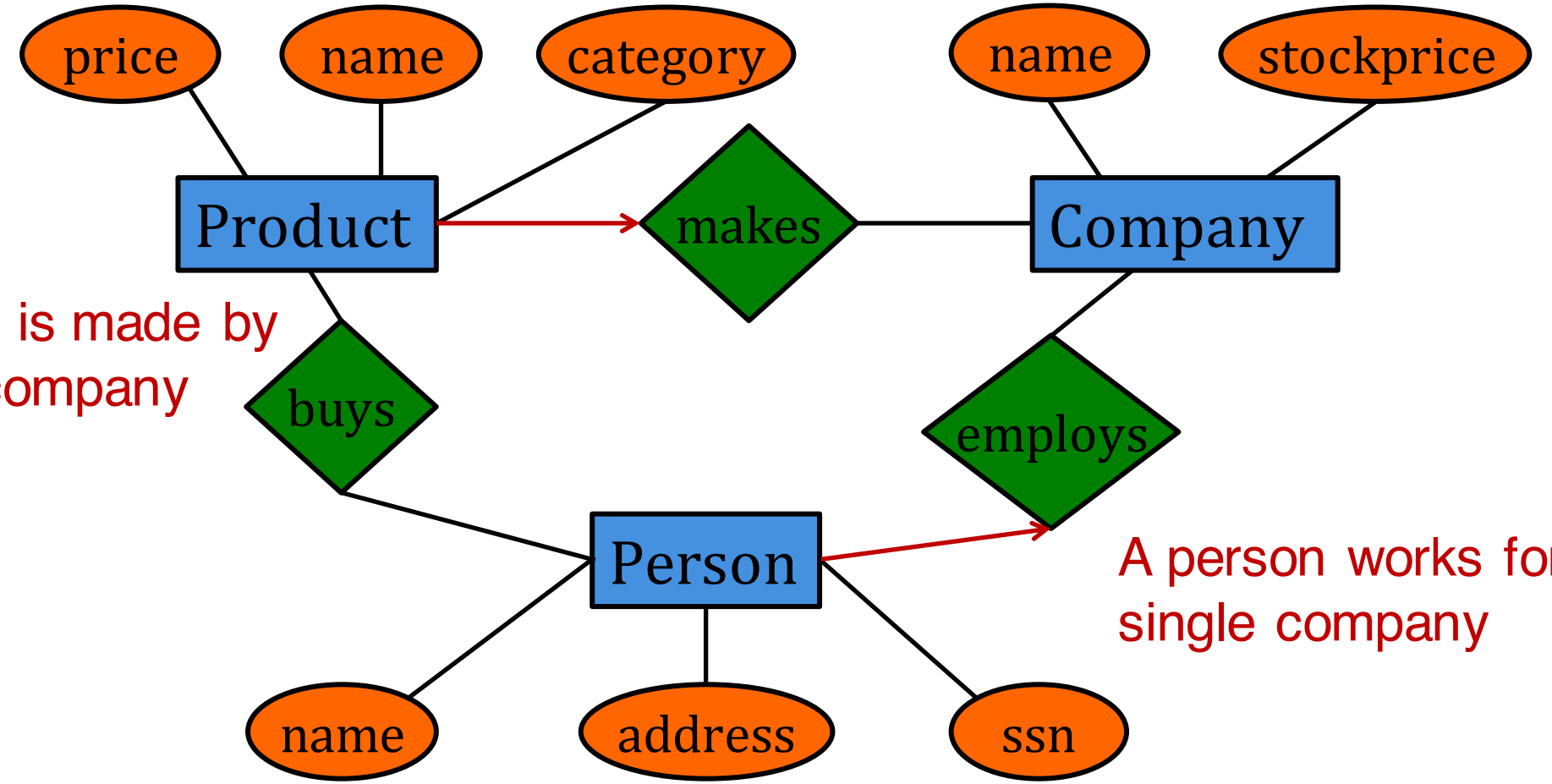
Key Constraints Example



Key Constraints Example



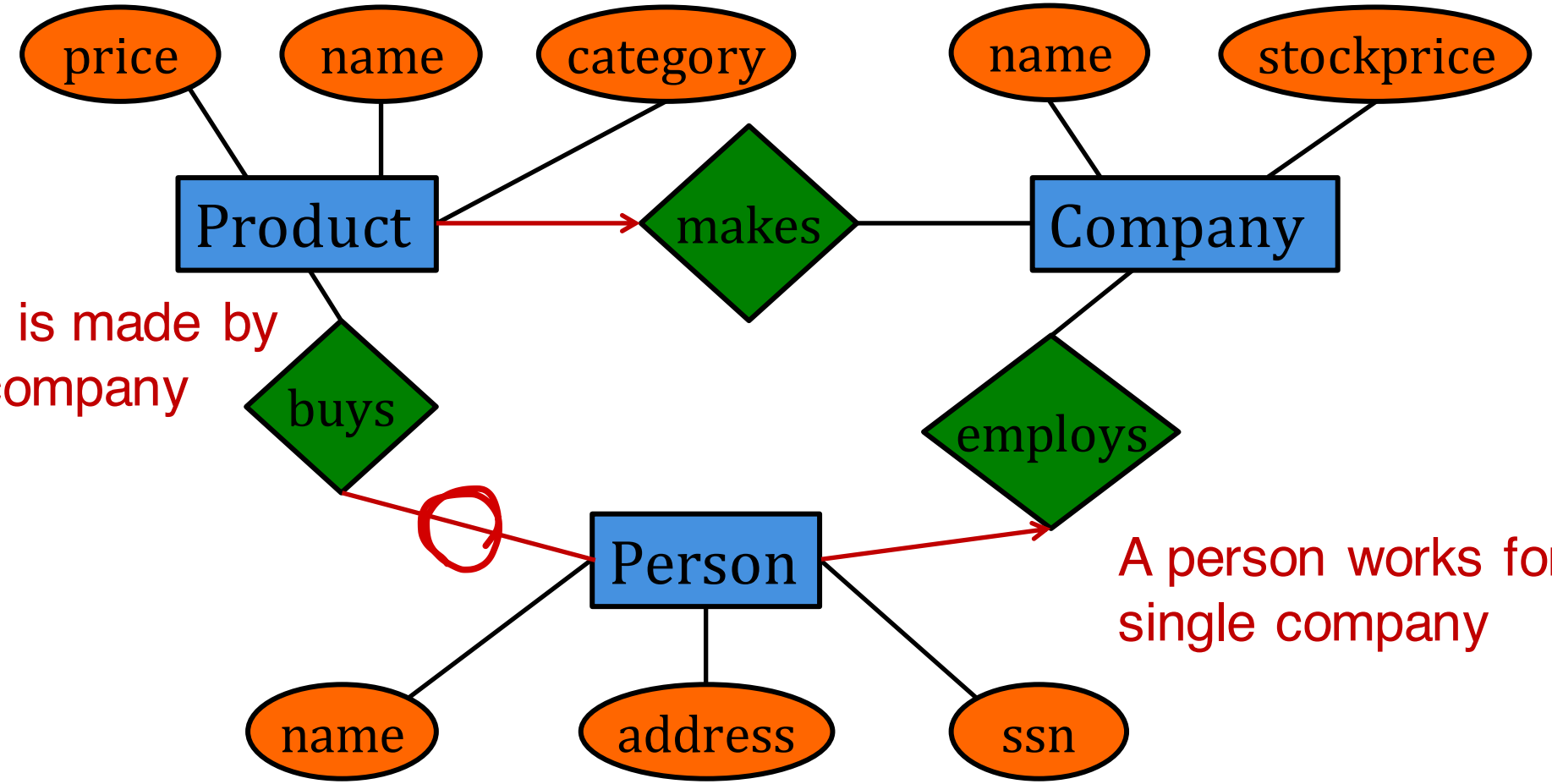
Key Constraints Example



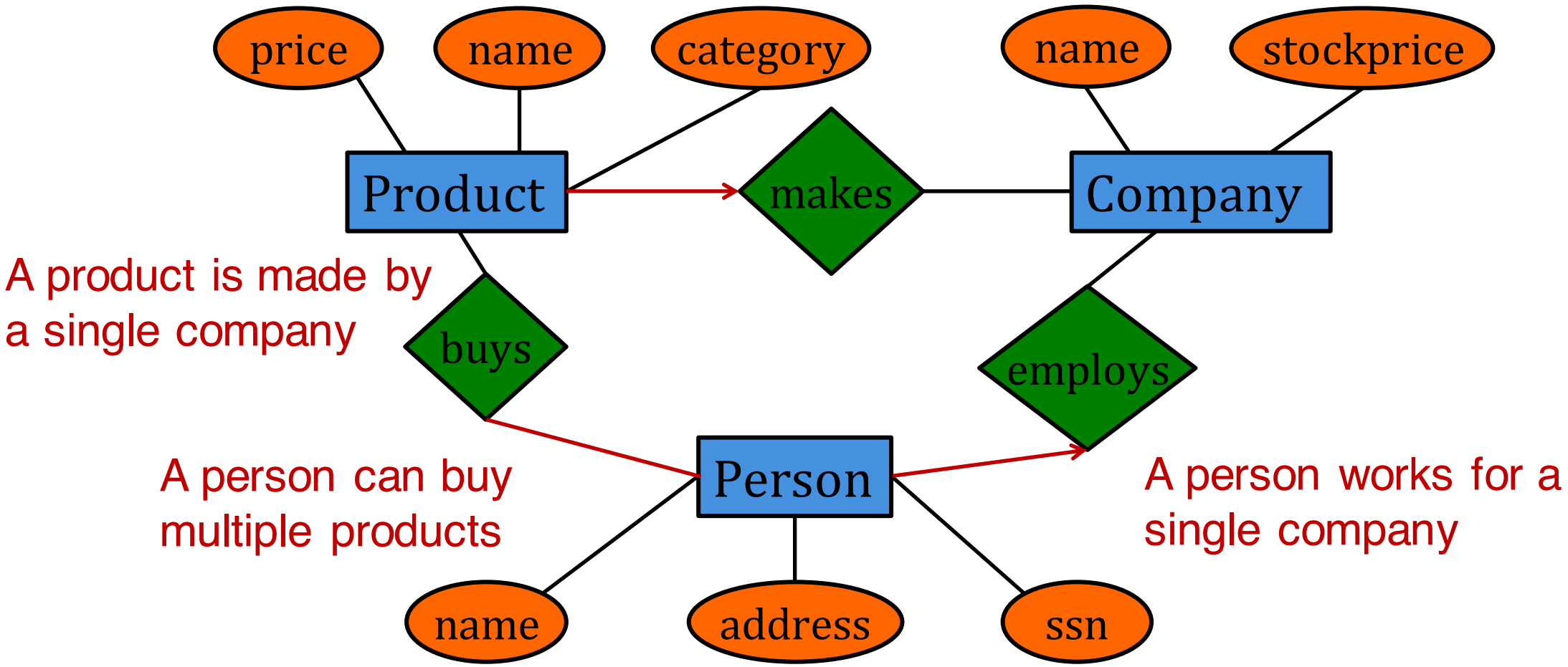
A product is made by a single company

A person works for a single company

Key Constraints Example

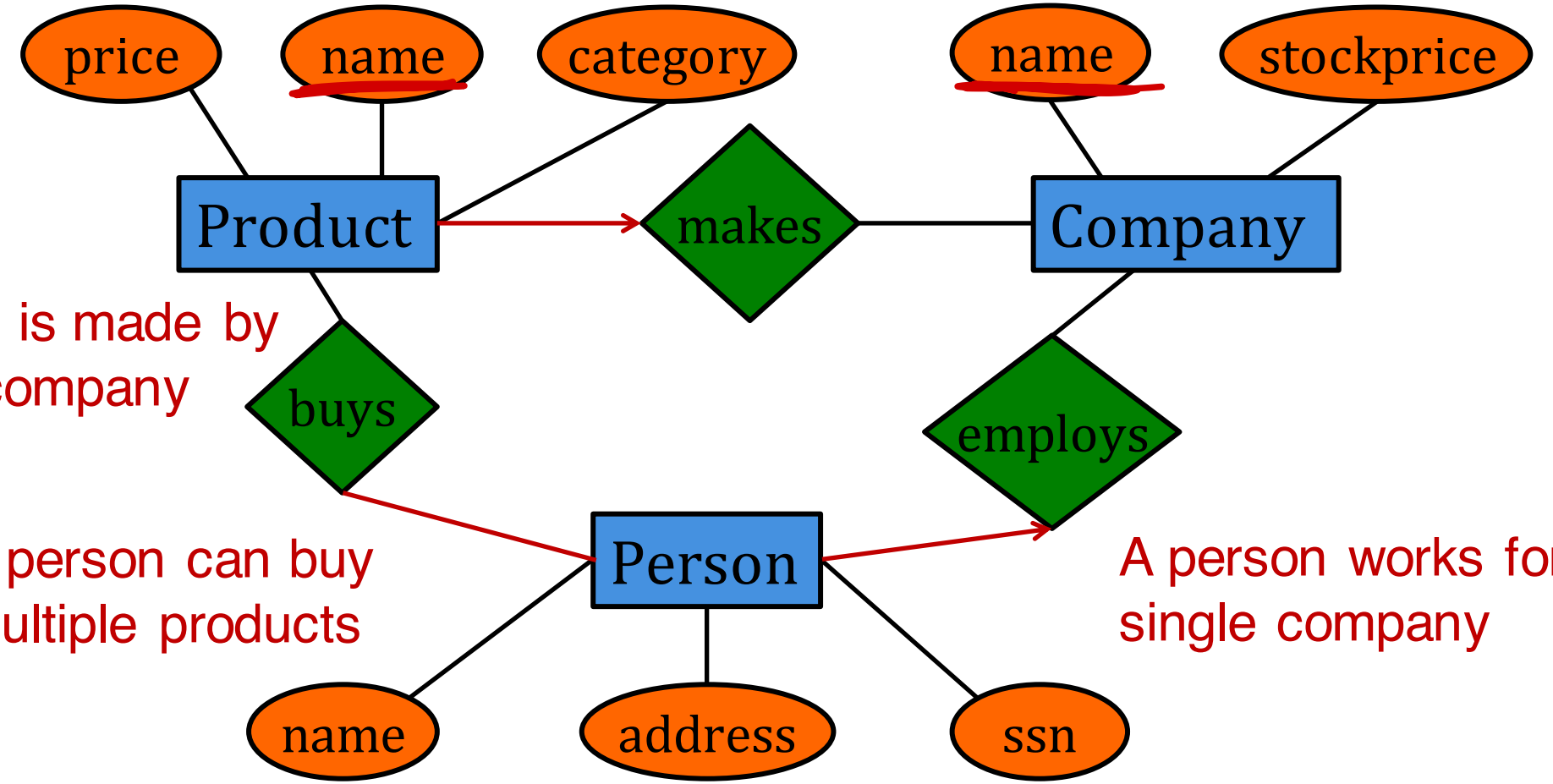


Key Constraints Example



Key Constraints Example

Makes
pname, cname.
'ipad', 'Apple'



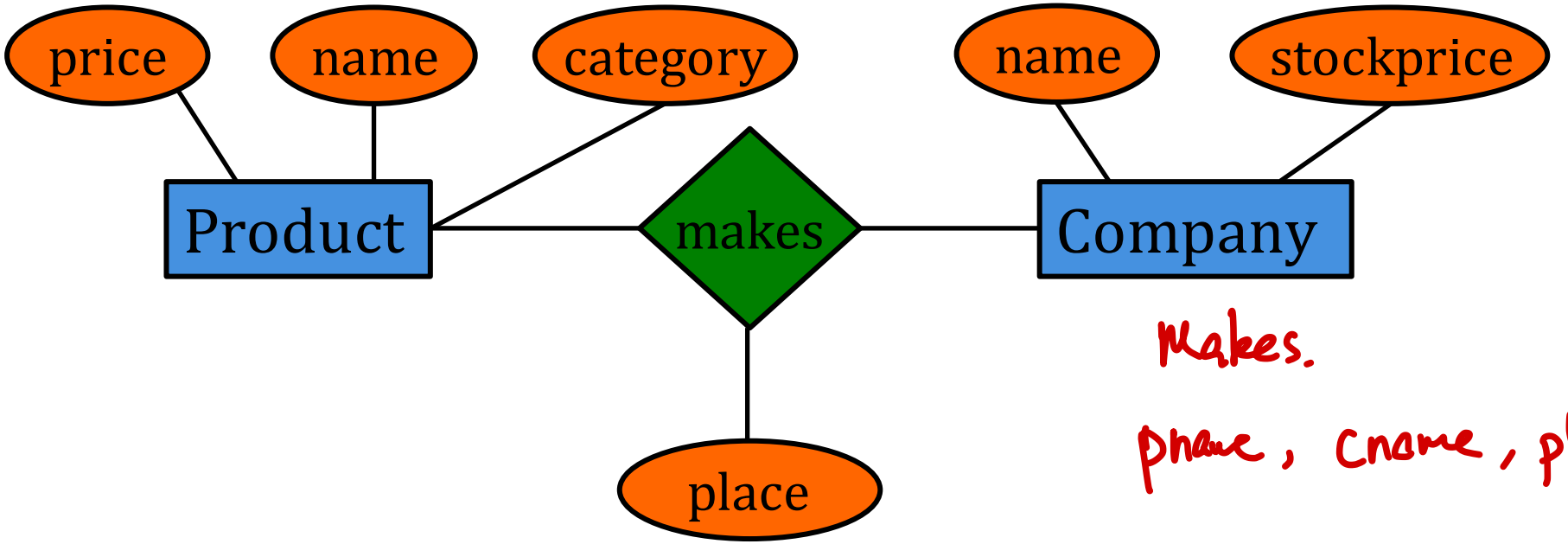
A product is made by a single company

A person can buy multiple products

A person works for a single company

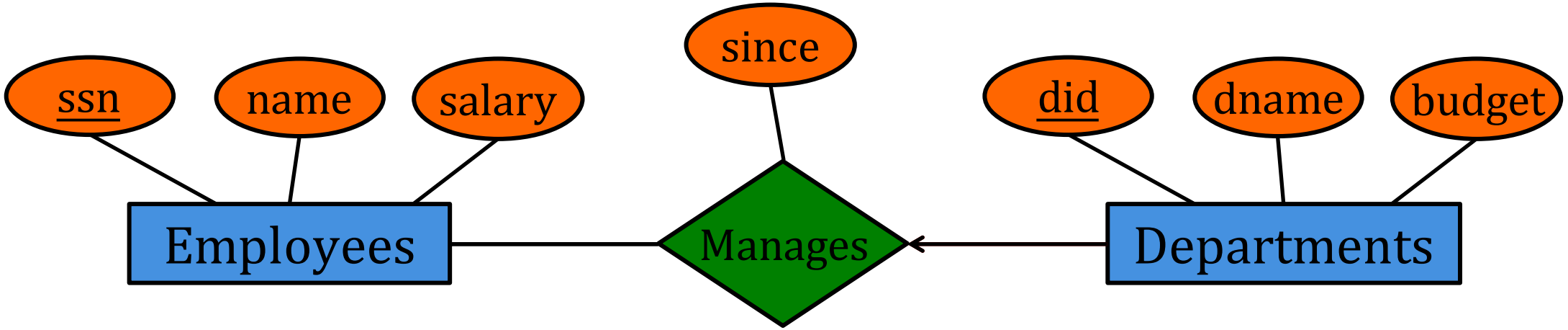
How to design schema for this database?

Descriptive Attribute in Relationship



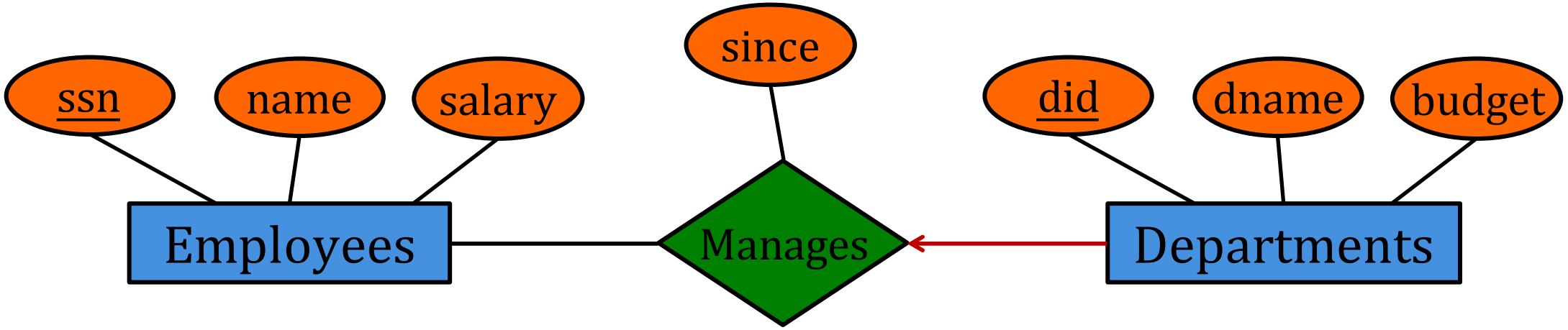
Record information about the relationship, rather than about any one of the participating entities

Participation Constraint – Motivation



Key constraint: each department has **at most** one manager

Participation Constraint – Motivation

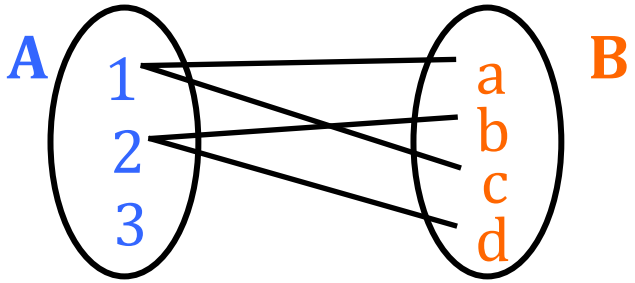


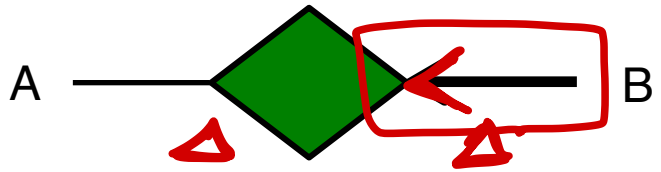
Key constraint: each department has **at most** one manager

How to express the constraint that each department must have a manager? (i.e., **at least** one manager)

– Answer: Participation constraint

Participation Constraint

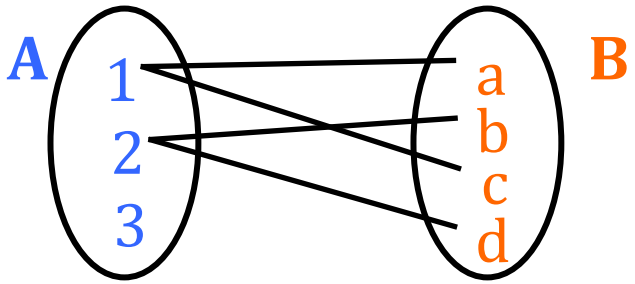


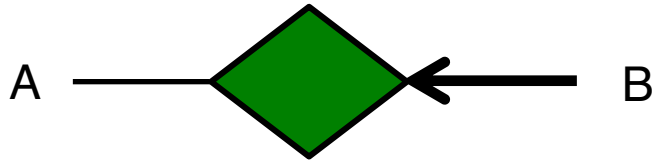


Total participation: Every entity participates at least one relationship

– We use a thick line to represent a total participation

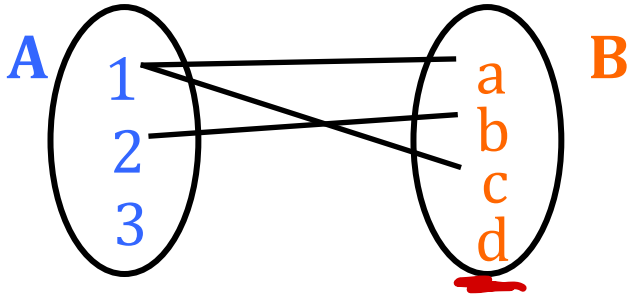
Participation Constraint

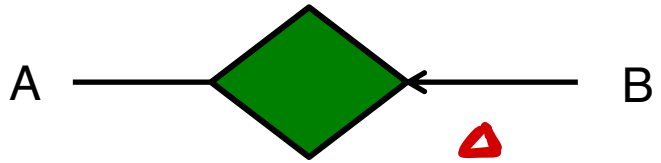




Total participation: Every entity participates at least one relationship

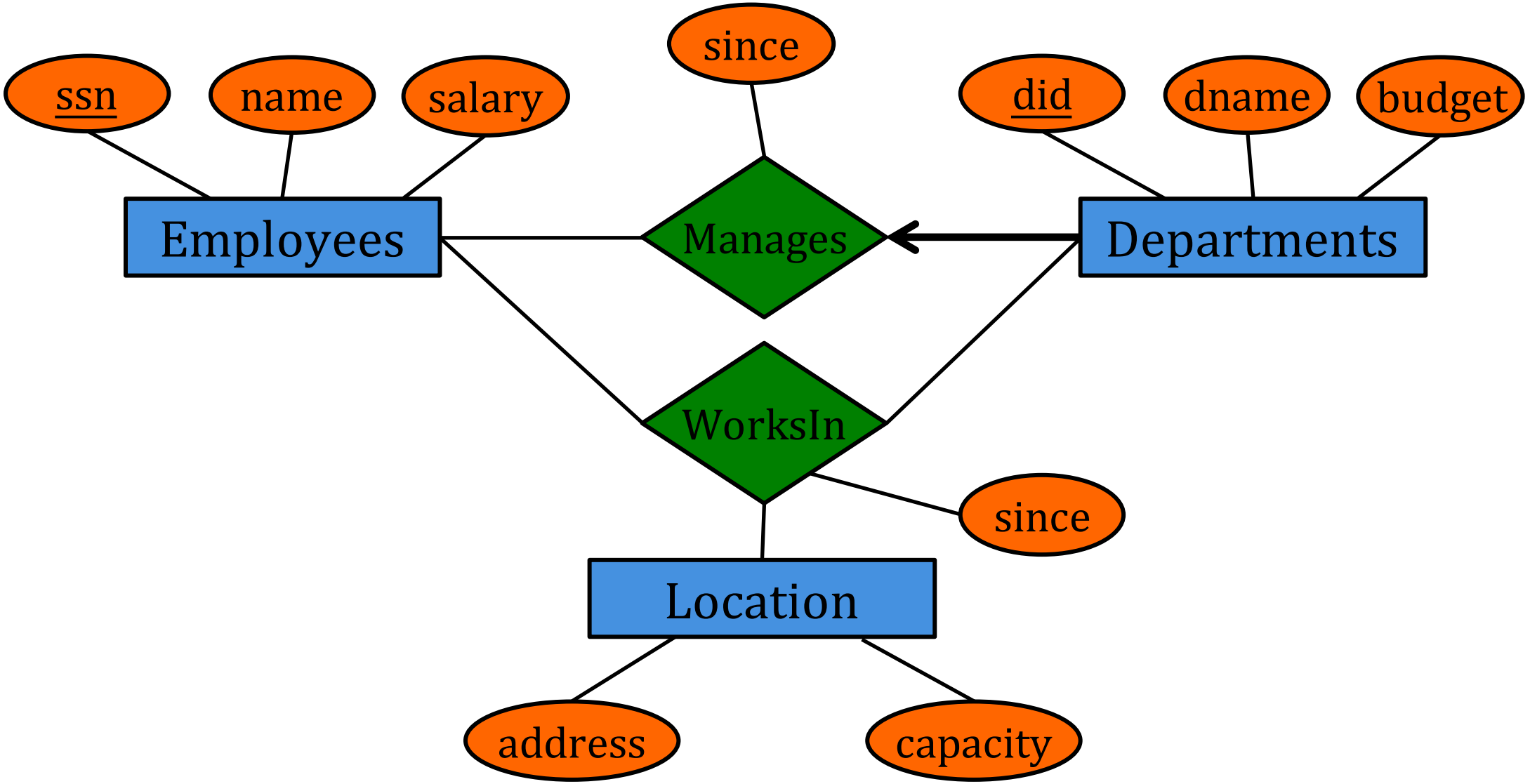
– We use a thick line to represent a total participation



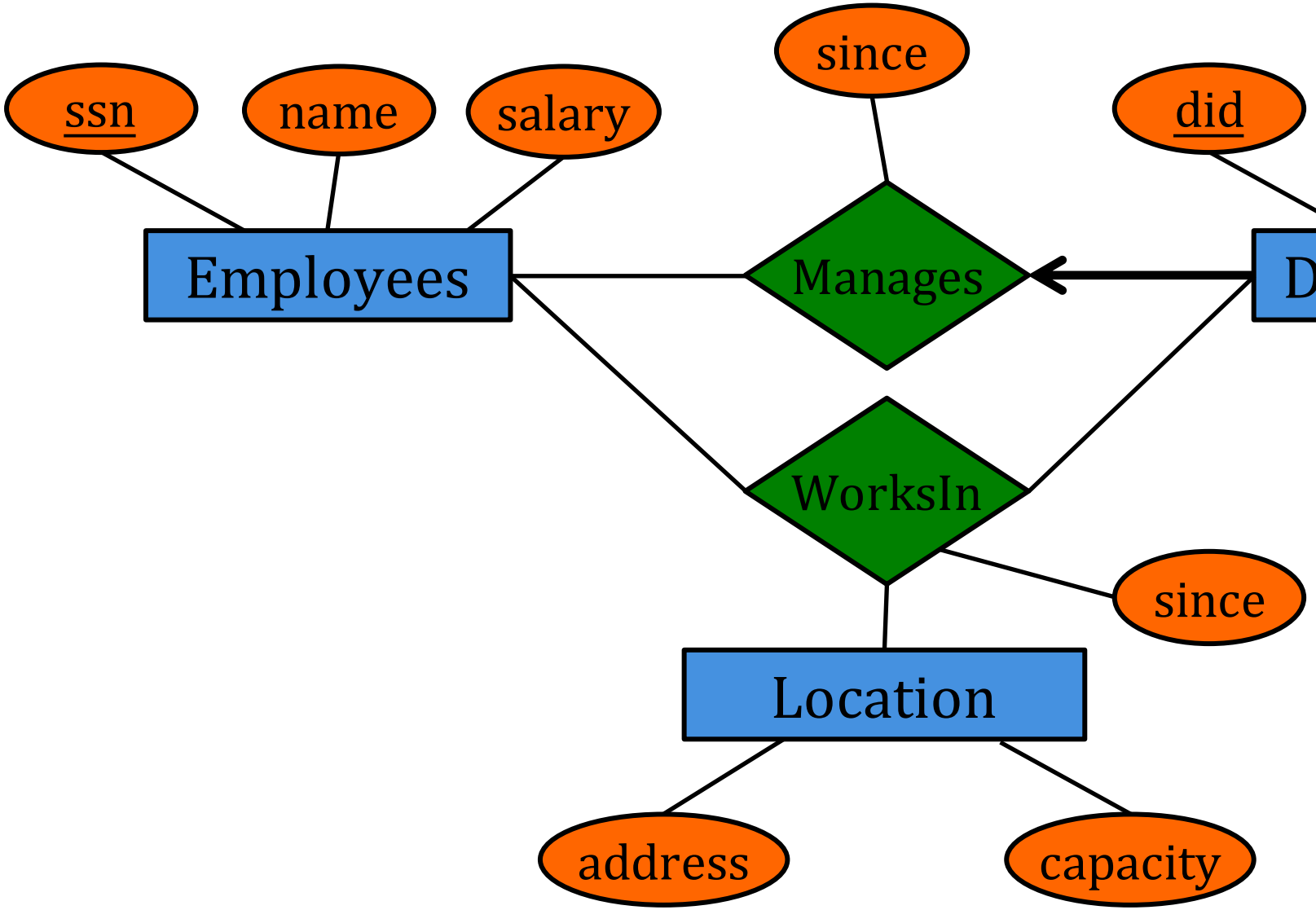


Partial participation: A participation that is not total is said to be partial

Participation Constraint – Example



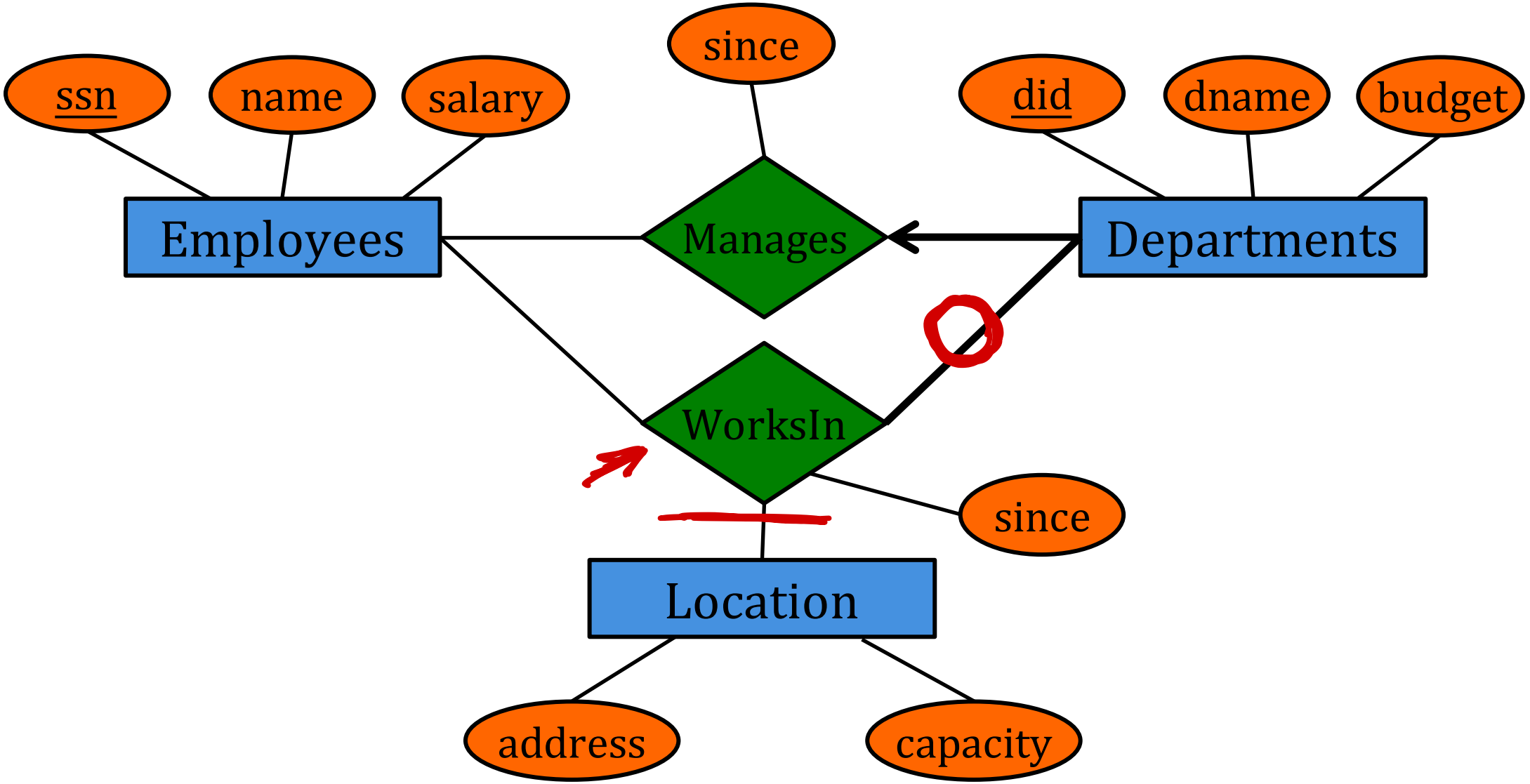
Participation Constraint – Example



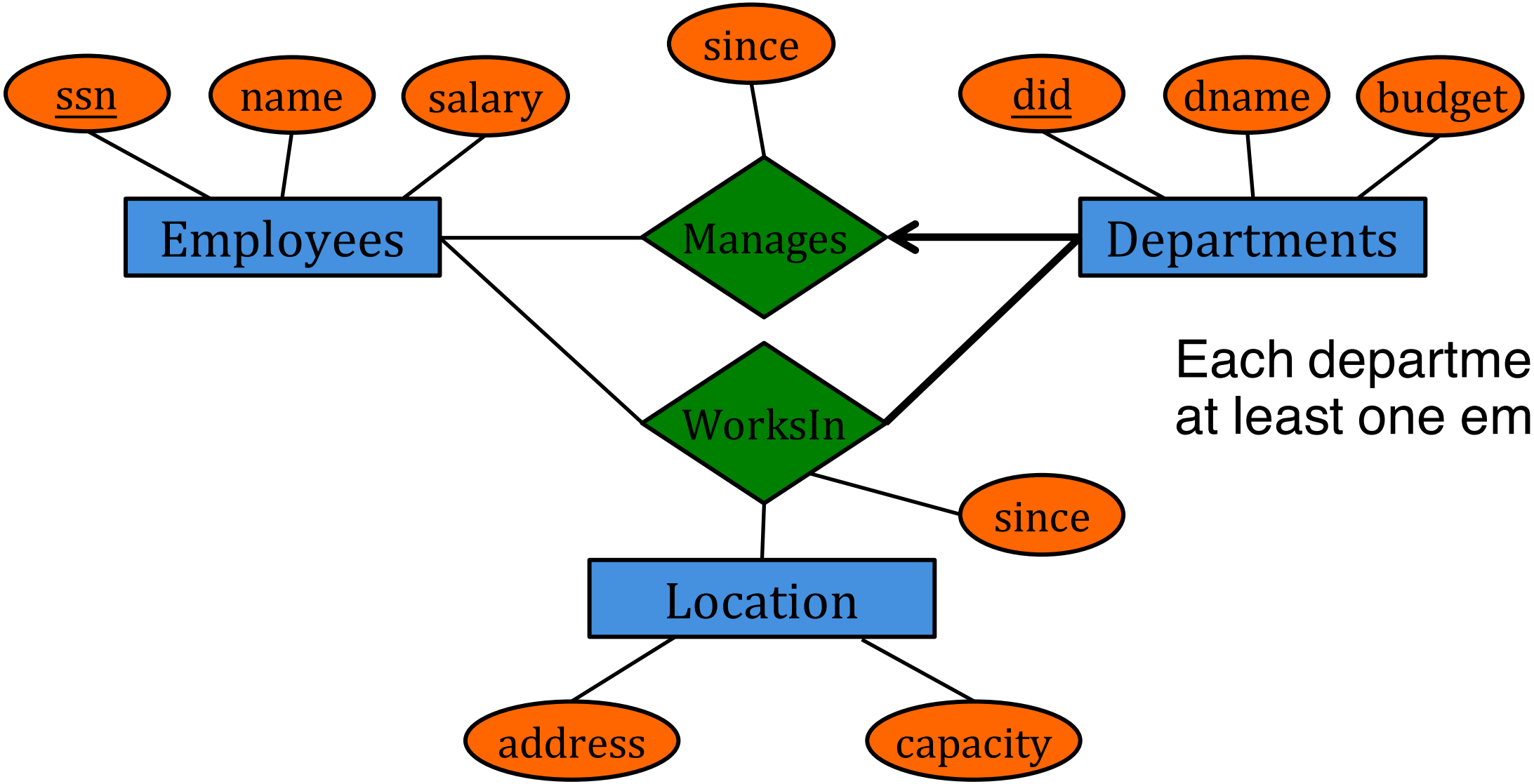
Each department has exactly one manager

- **arrow**: has at most one manager
- **thick line**: has at least one manager

Participation Constraint – Example

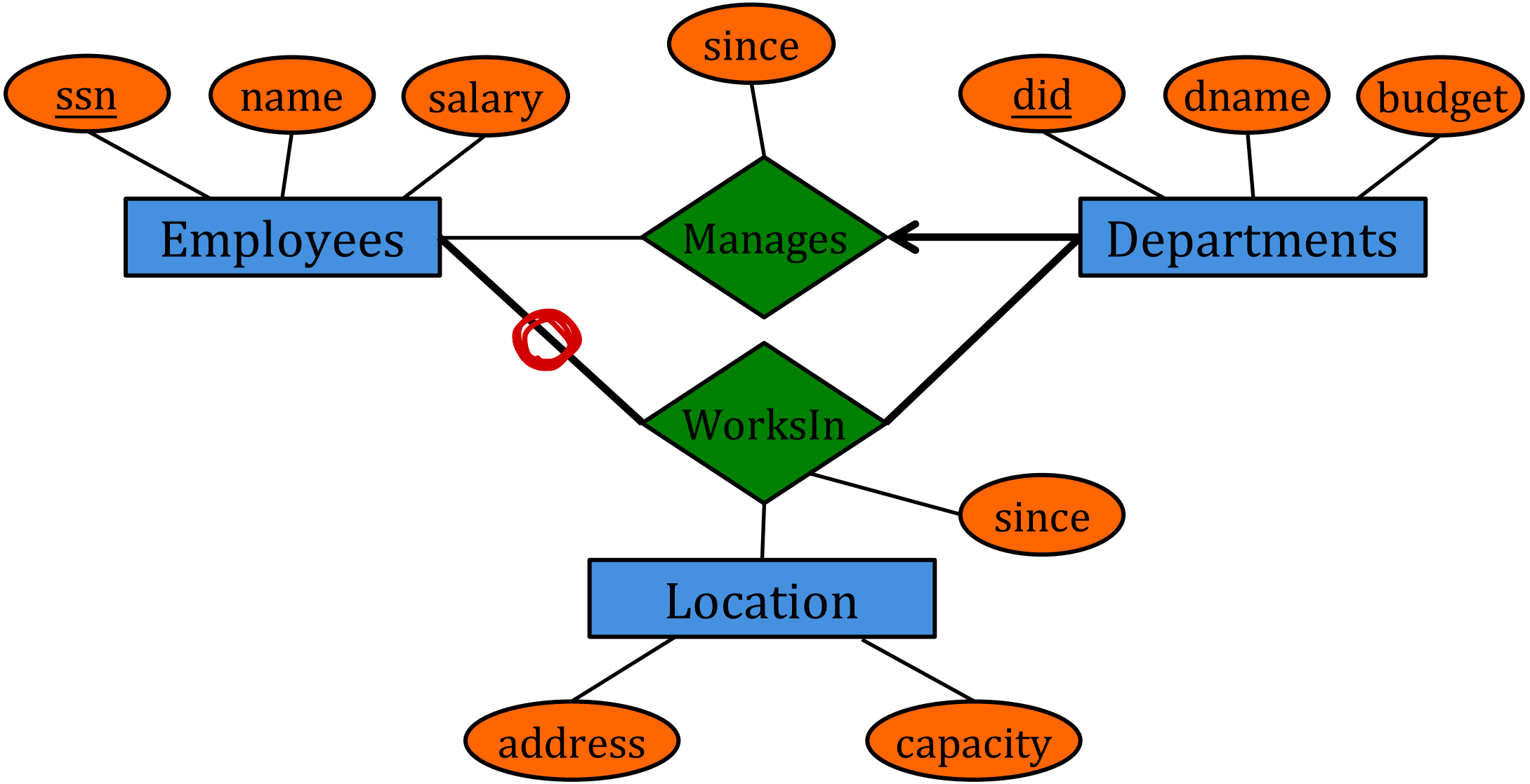


Participation Constraint – Example

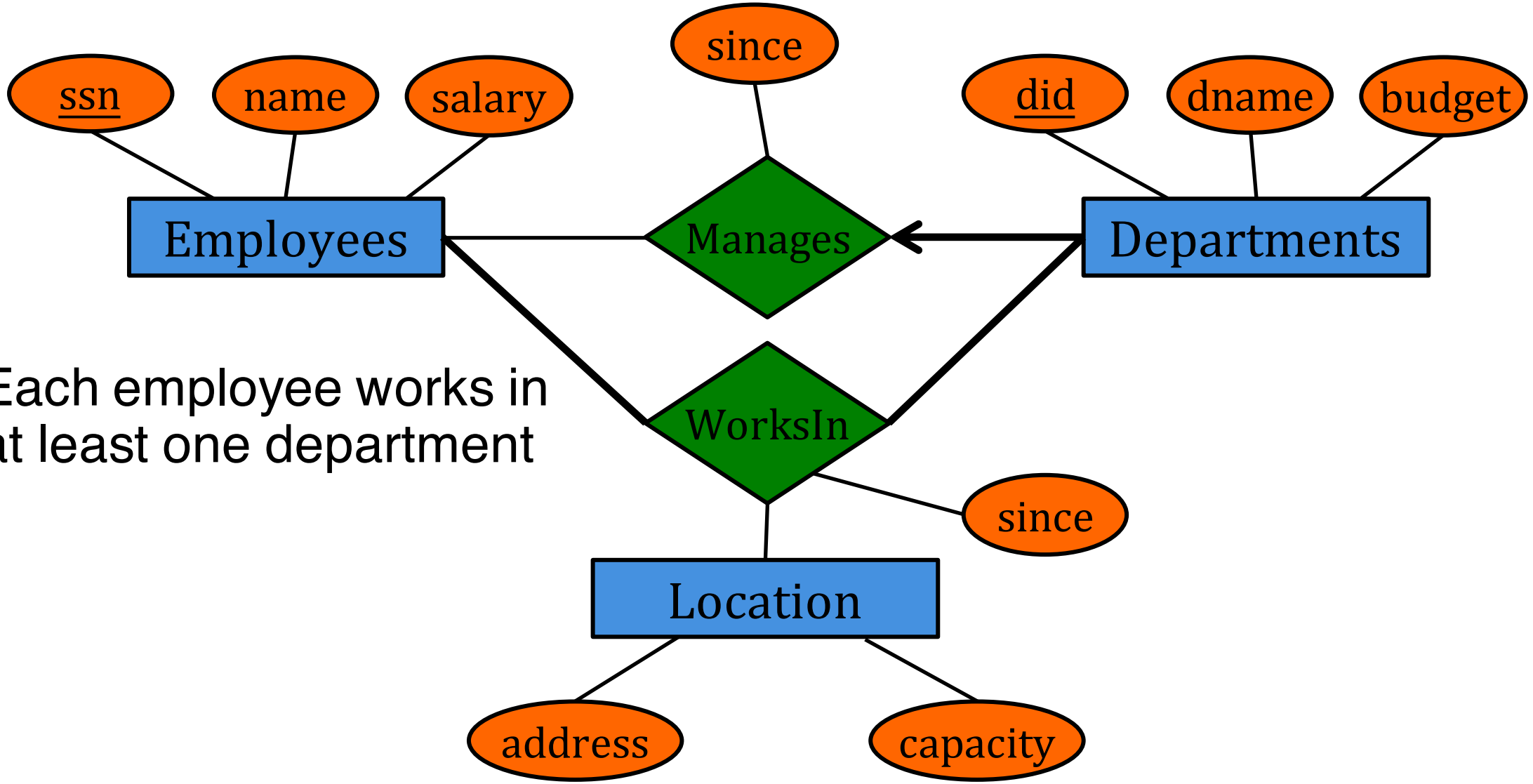


Each department has at least one employee

Participation Constraint – Example

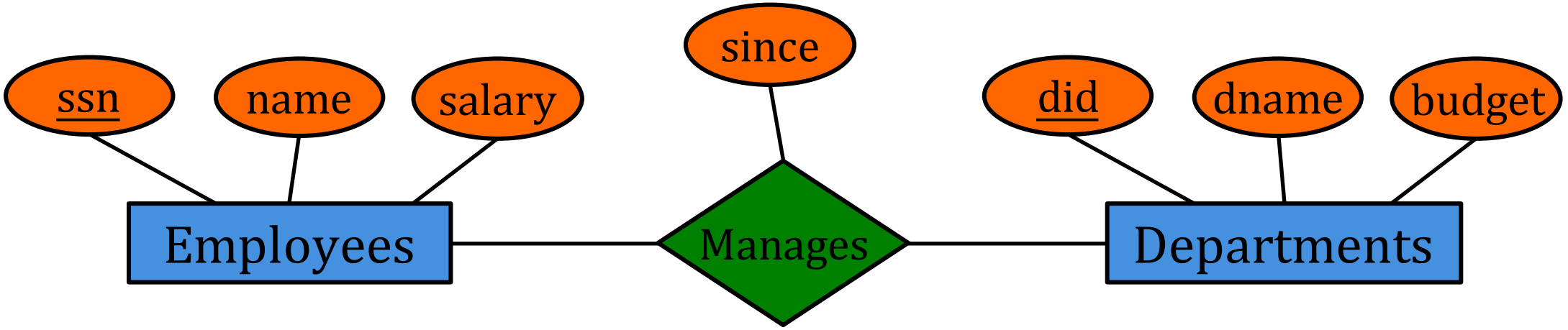


Participation Constraint – Example

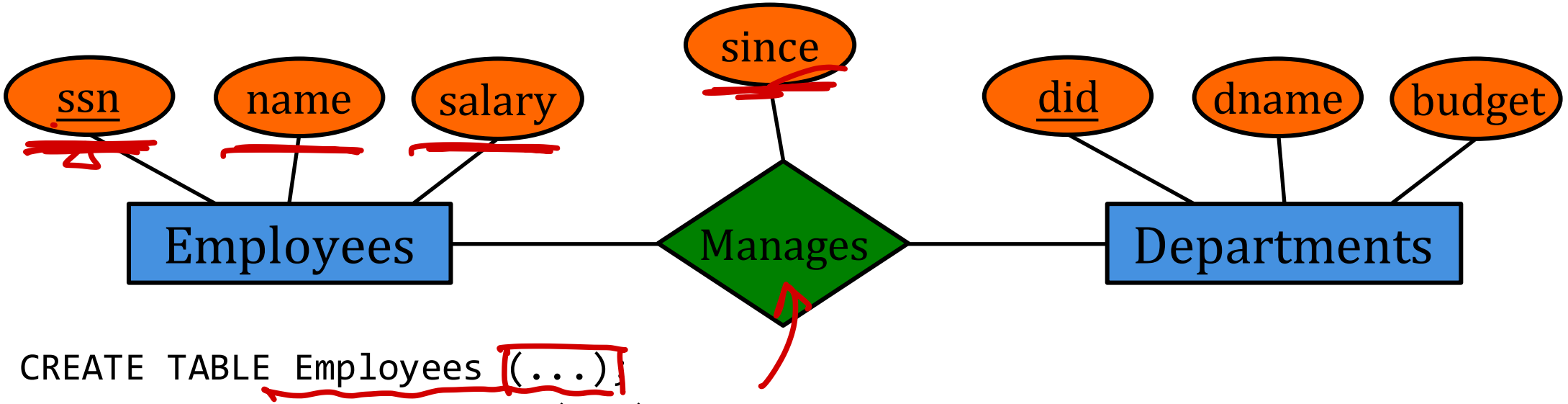


Each employee works in at least one department

Constraints -> Schema

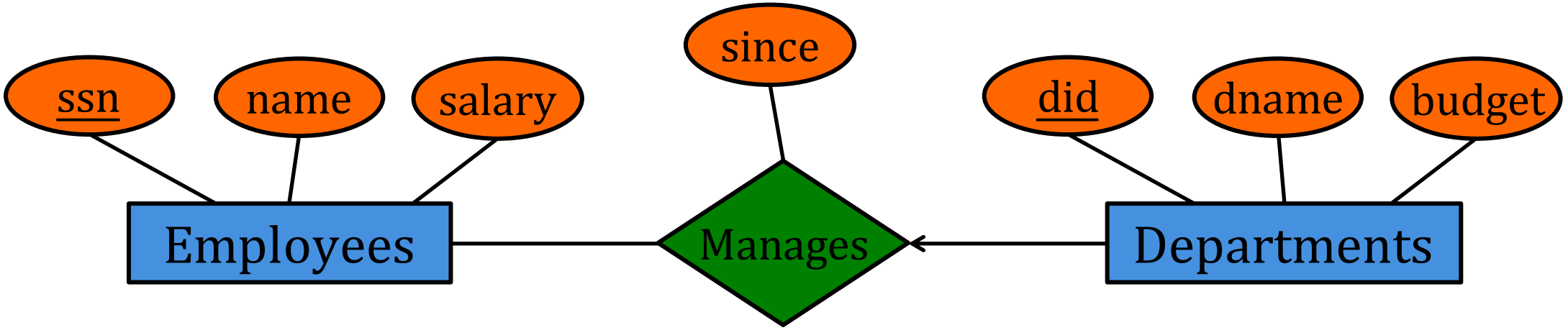


Constraints -> Schema

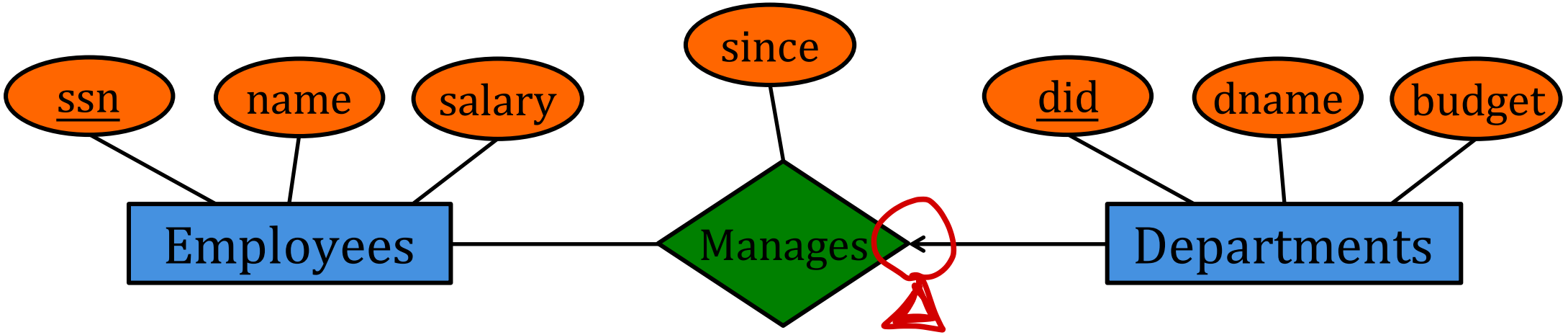



```
CREATE TABLE Employees (...);
CREATE TABLE Departments (...);
CREATE TABLE Manages(
    ssn CAHR(11),
    did CHAR(11),
    since DATE,
    PRIMARY KEY (ssn, did, since),
    FOREIGN KEY (did) REFERENCES Departments,
    FOREIGN KEY (ssn) REFERENCES Employees
);
```

Constraints -> Schema

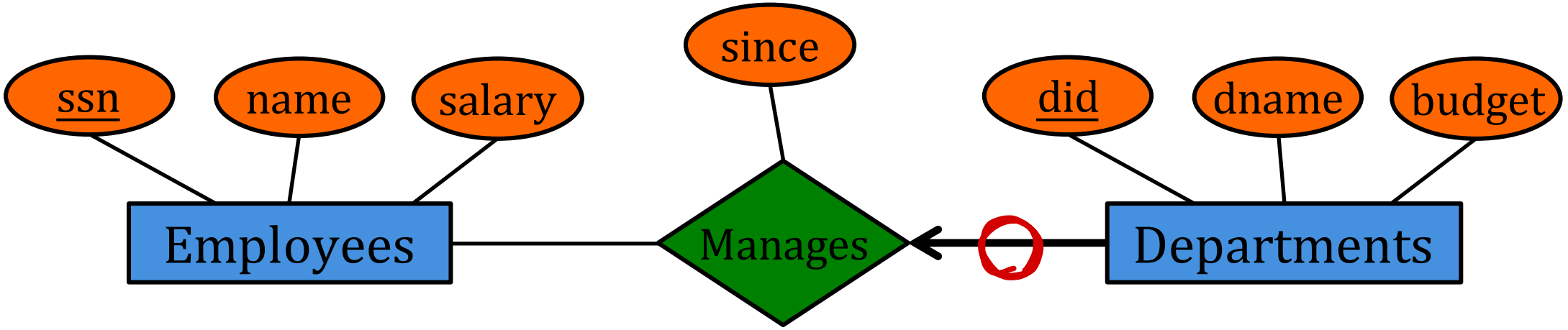


Constraints -> Schema

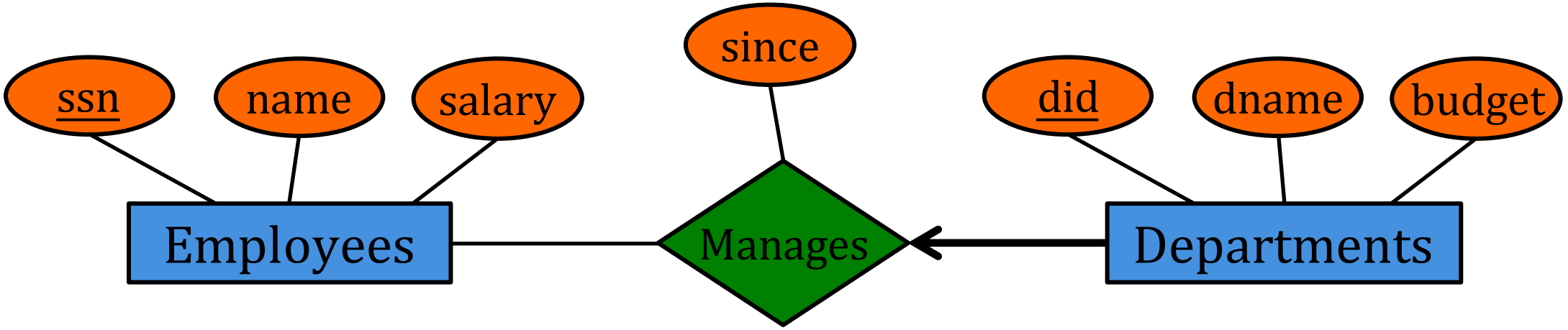



```
CREATE TABLE Employees (...);
CREATE TABLE Departments (
    did CHAR(11) PRIMARY KEY,
    dname CHAR(20),
    budget REAL,
    since DATE,
    manager CHAR(11),
    FOREIGN KEY (manager) REFERENCES Employees
);
```

Constraints -> Schema

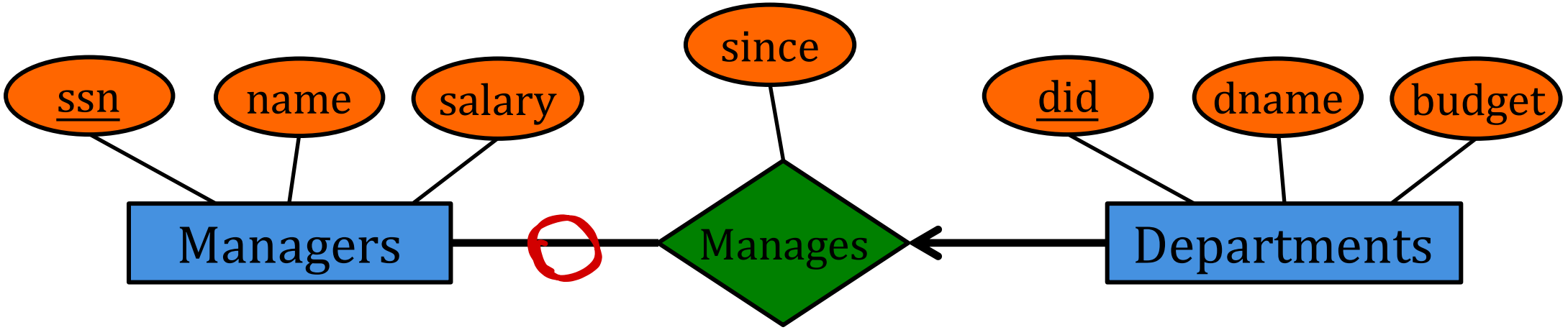


Constraints -> Schema




```
CREATE TABLE Employees (...);
CREATE TABLE Departments (
    did CHAR(11) PRIMARY KEY,
    dname CHAR(20),
    budget REAL,
    since DATE,
    manager CHAR(11) NOT NULL,
    FOREIGN KEY (manager) REFERENCES Employees
);
```

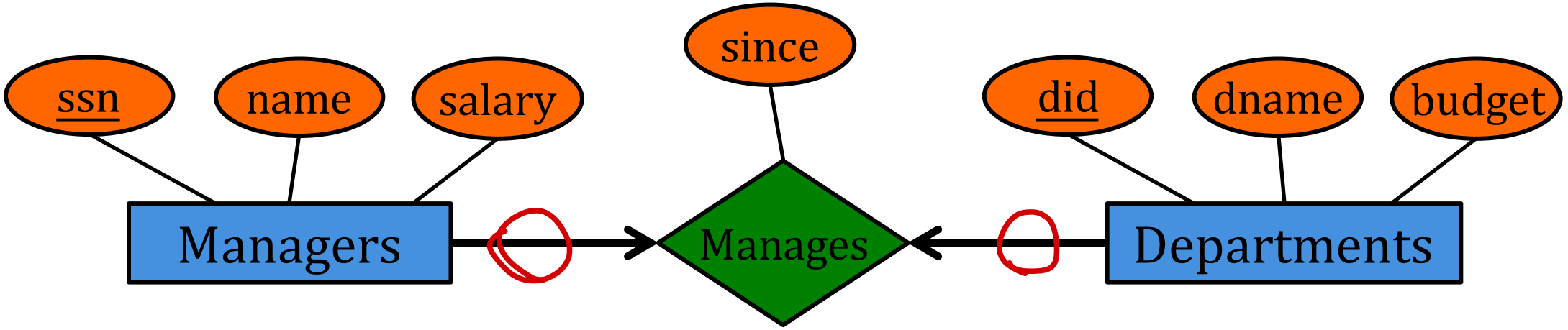
Constraints -> Schema



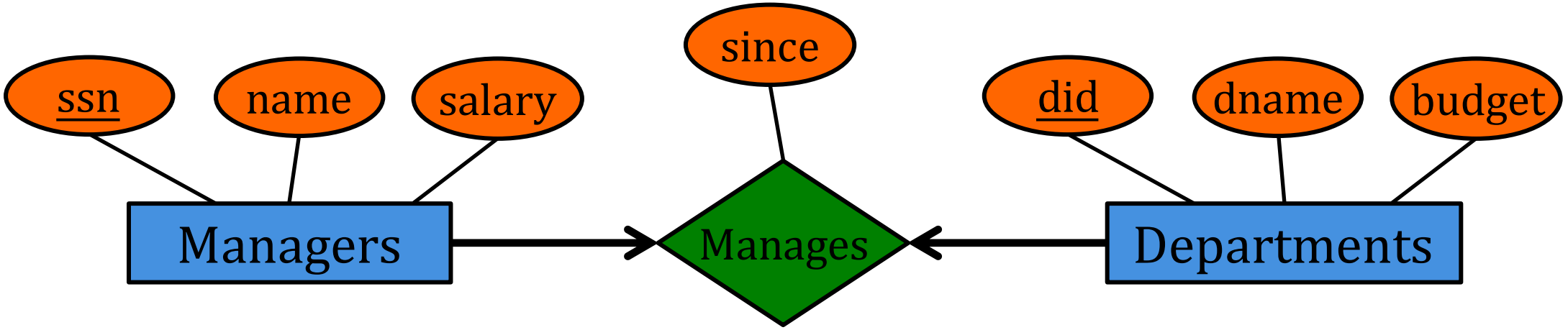

```
CREATE TABLE Managers (
    ssn CHAR(11) PRIMARY KEY,
    name CHAR(20),
    salary REAL
);
```

```
CREATE TABLE Departments (
    did CHAR(11) PRIMARY KEY,
    dname CHAR(20),
    budget REAL,
    since DATE,
    manager CHAR(11) NOT NULL,
    FOREIGN KEY (manager) REFERENCES Employees
);
```

Constraints -> Schema



Constraints -> Schema




```
CREATE TABLE Manages (
    ssn CHAR(11) PRIMARY KEY,
    name CHAR(20),
    salary REAL
    since DATE
    did CHAR(11) UNIQUE,
    dname CHAR(20)
    budget REAL
);
```

Summary

Attributes, entity, and relations

Key constraints

- Many-one, one-one, many-many
- Limitations of arrows

Participation constraints

Constraints to Schema

Next Lecture

More features of ER models

Functional dependency