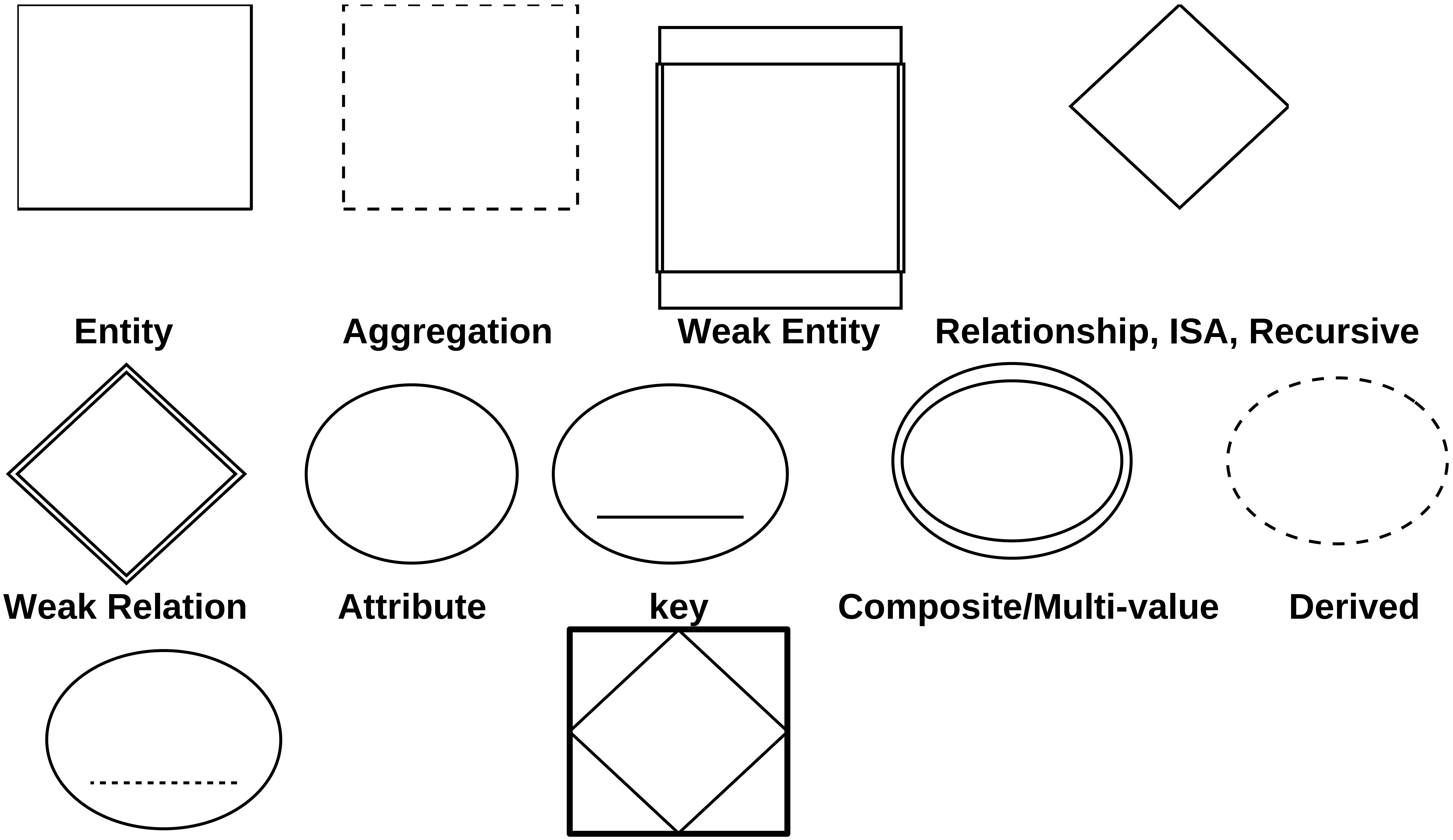
Databases Systems

The Conceptual/Semantic Model
Entity Relationship Diagrams (ERD)

Entity Relationship Diagram (ERD)

- Based on database requirements and driven by use cases
- Abstract perspective of the database design
- High level
- Clarify two important concepts:
- The major entities within the system scope
- The interrelationships among those entities

ER Model Basics

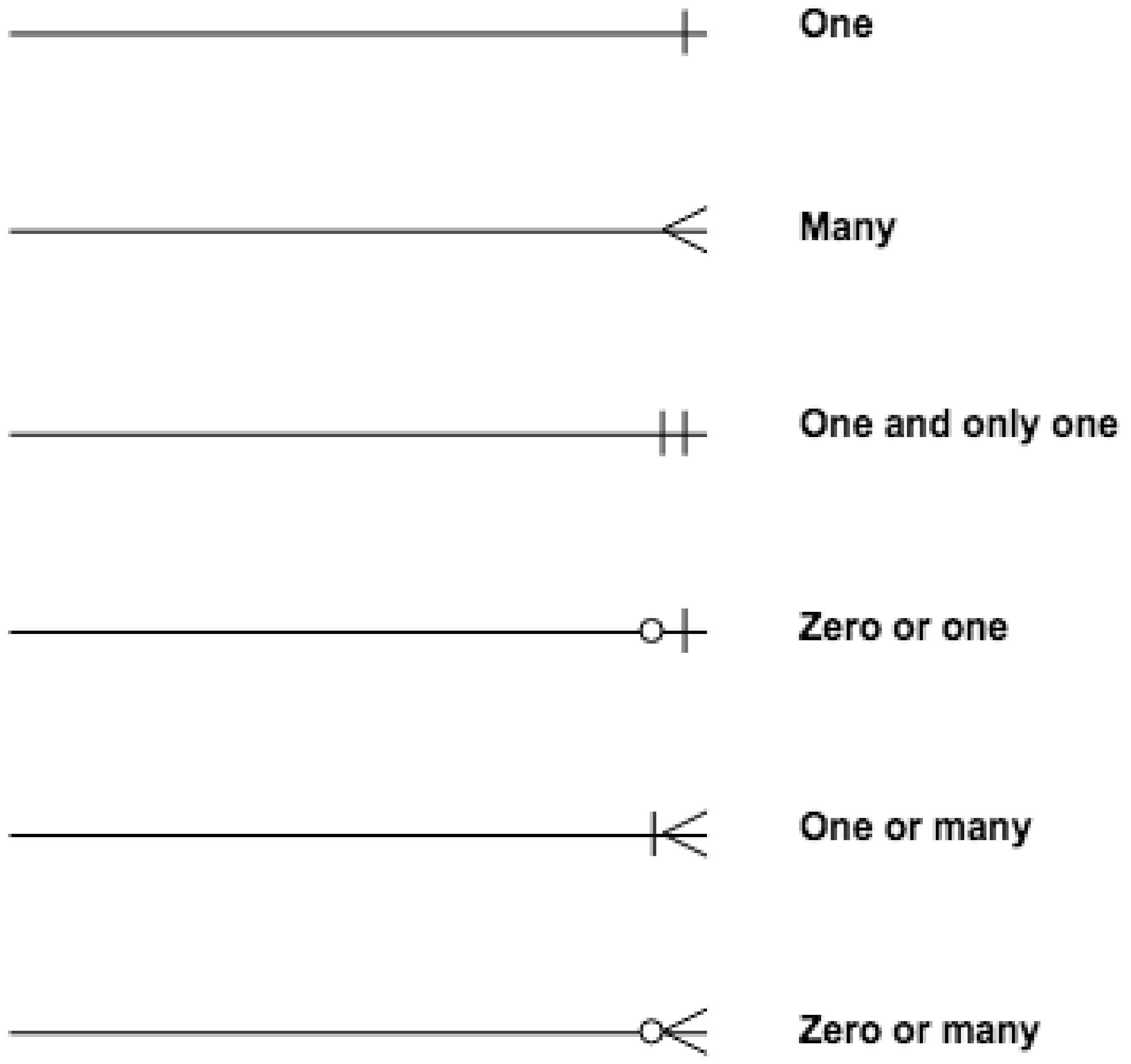


Weak key

Relation becomes an entity

3

ER Model Basics
Crow's foot notation



ER Model Basics

Many-to-Many (M:N):
An employee shall work in **many** departments
A department shall have **many** employees.

Many-to-One (M:1):
A department shall have **many** employees A employee shall work in **only one** department

One-to-Many (1:M):

Only and Only One:
An employee shall have **only and only one** unique SSN.

Recursive:
An employee shall have a supervisor **which is also an employee**

ISA:

Type of relationships

An employee shall work in **only one** department A department shall have **many** employees.

An employee **is a hourly or contract employee.**

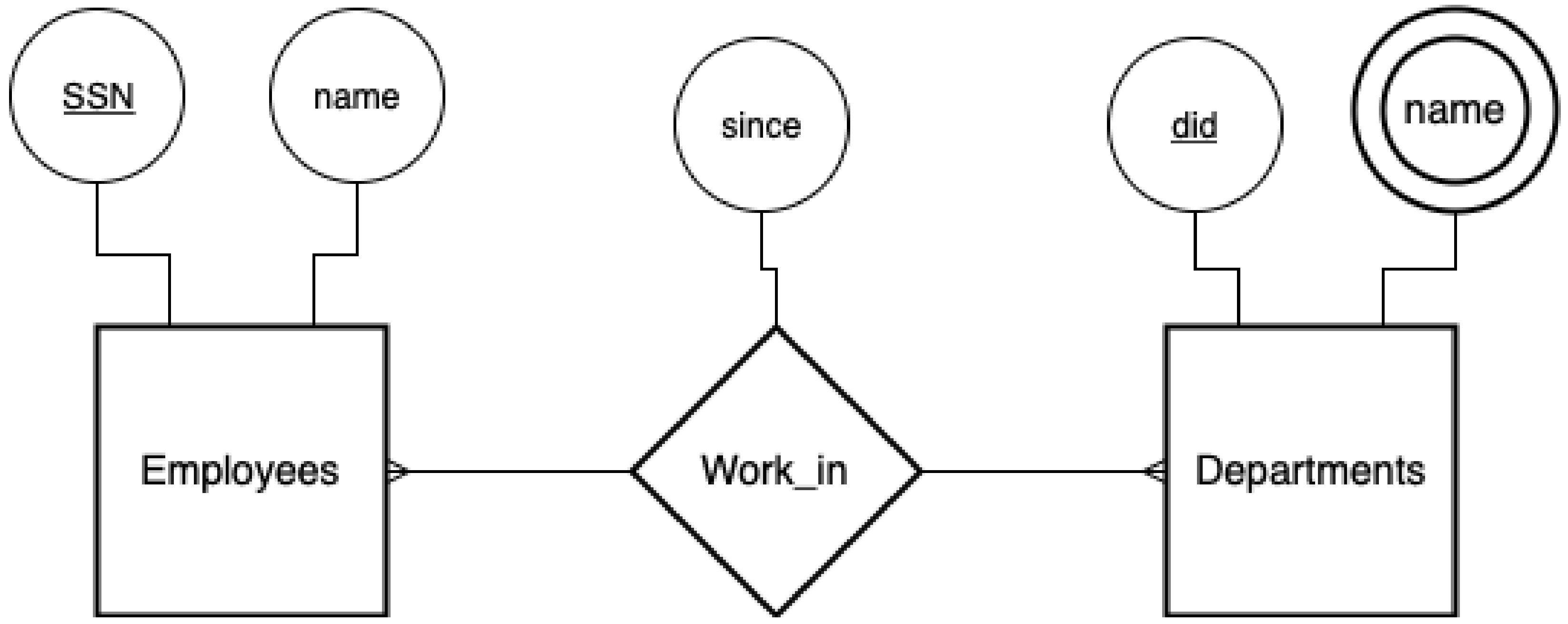
One-to-One (1:1):

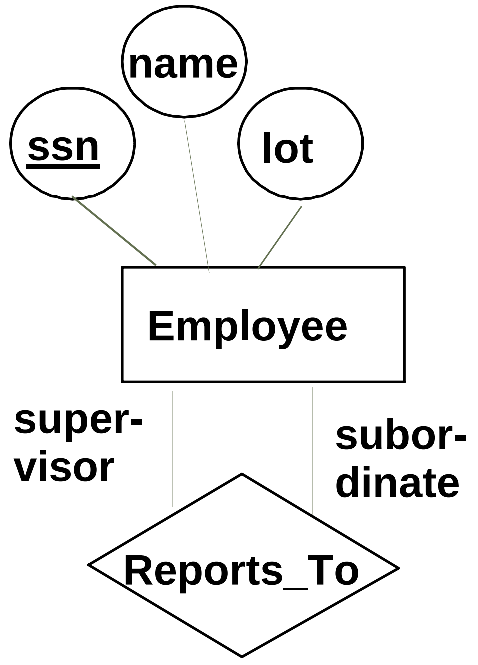
An employee shall work in **only one** department A department shall have **only one** employee.

Aggregation:

An employee shall be assigned to the SW engineering, and testing team, **but not to the marketing team**

ER Basics





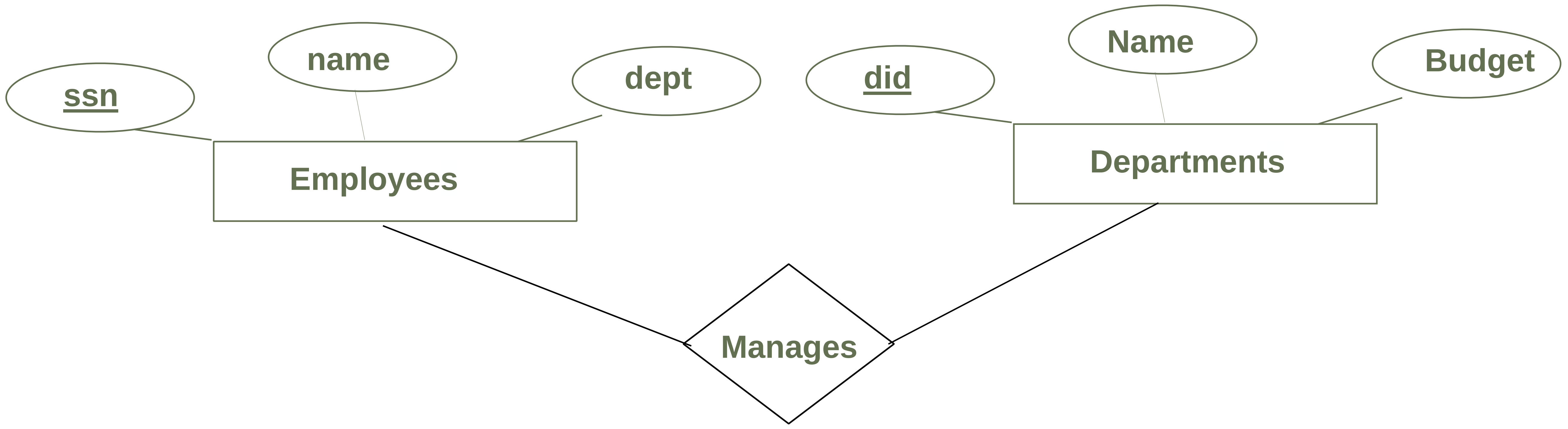
Relationship Set (RS): Collection of similar relationships.

The same **entity** could participate in different relationships sets:

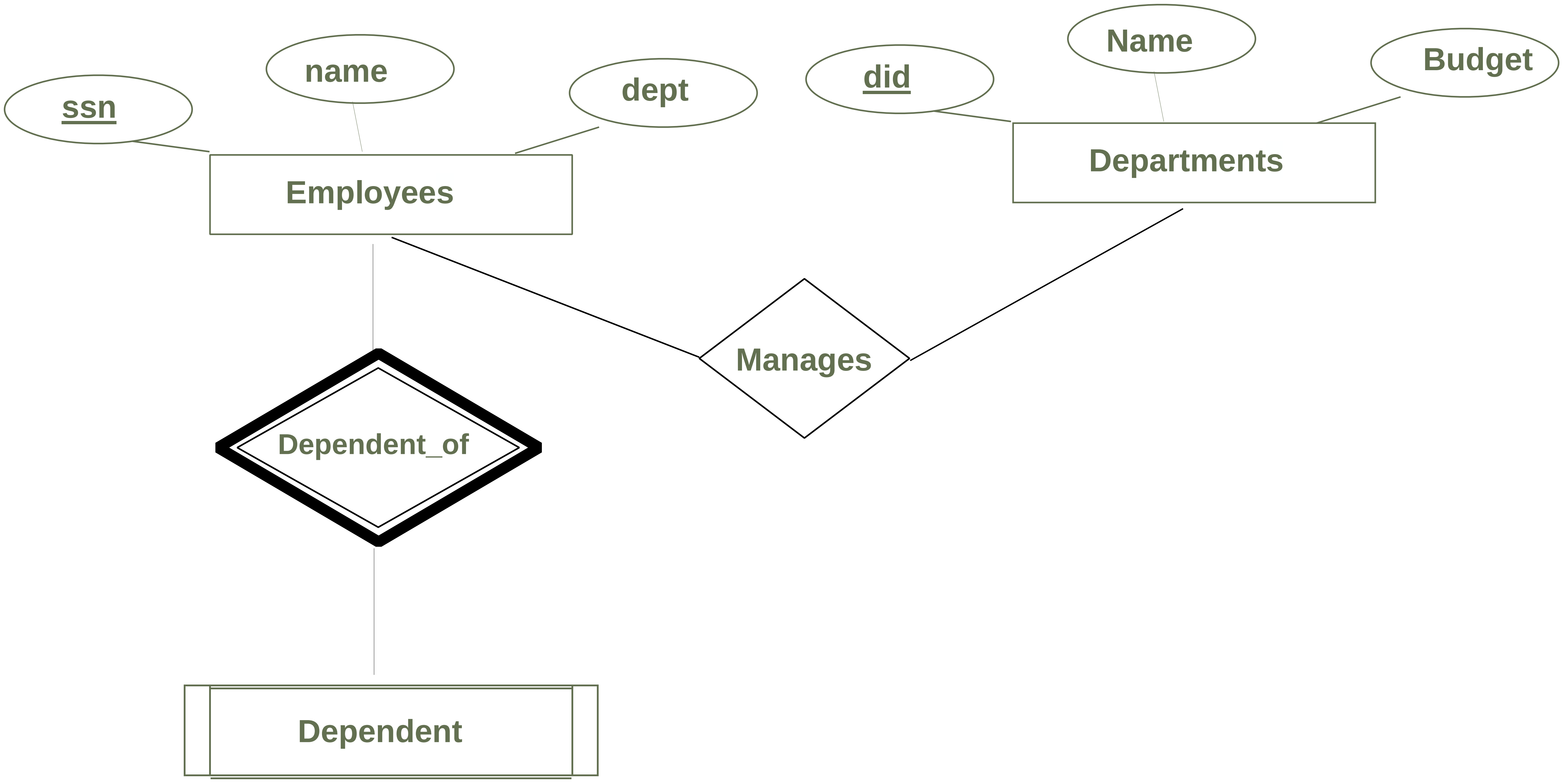
e.g Employee <Works_In> Dept, and Employee <Reports_To> supervisor

In this specific example, supervisor and subordinate are called **roles indicators**

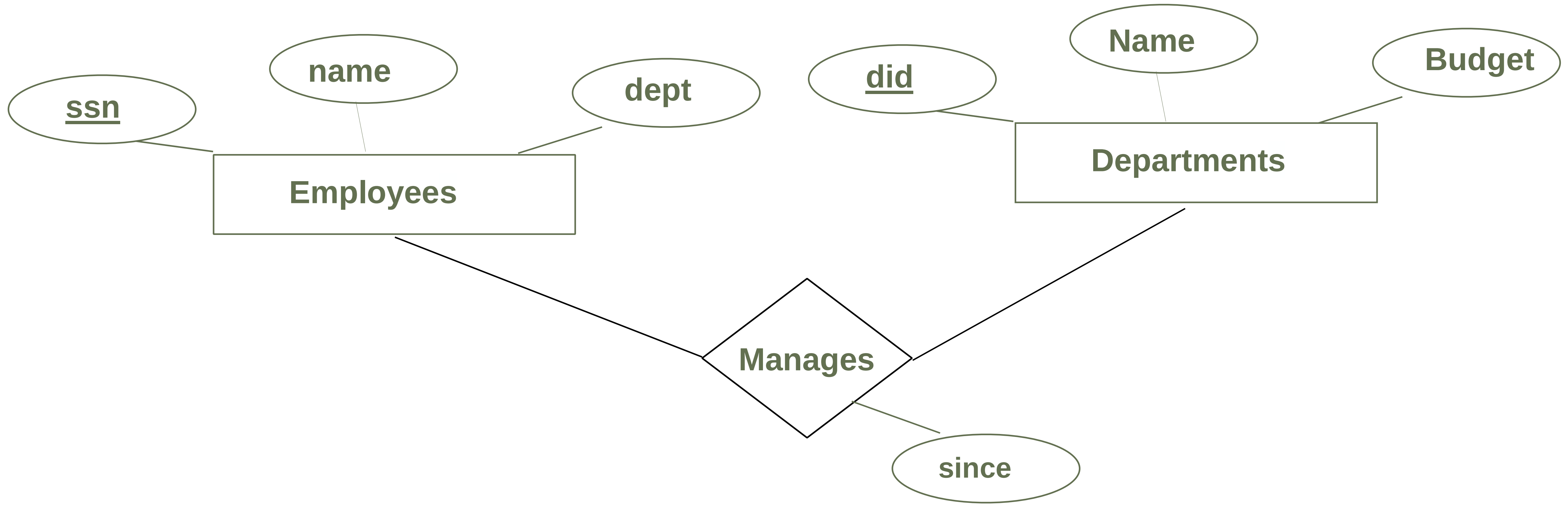
ER MODEL ENTITIES Strong Entities



Weak Entities



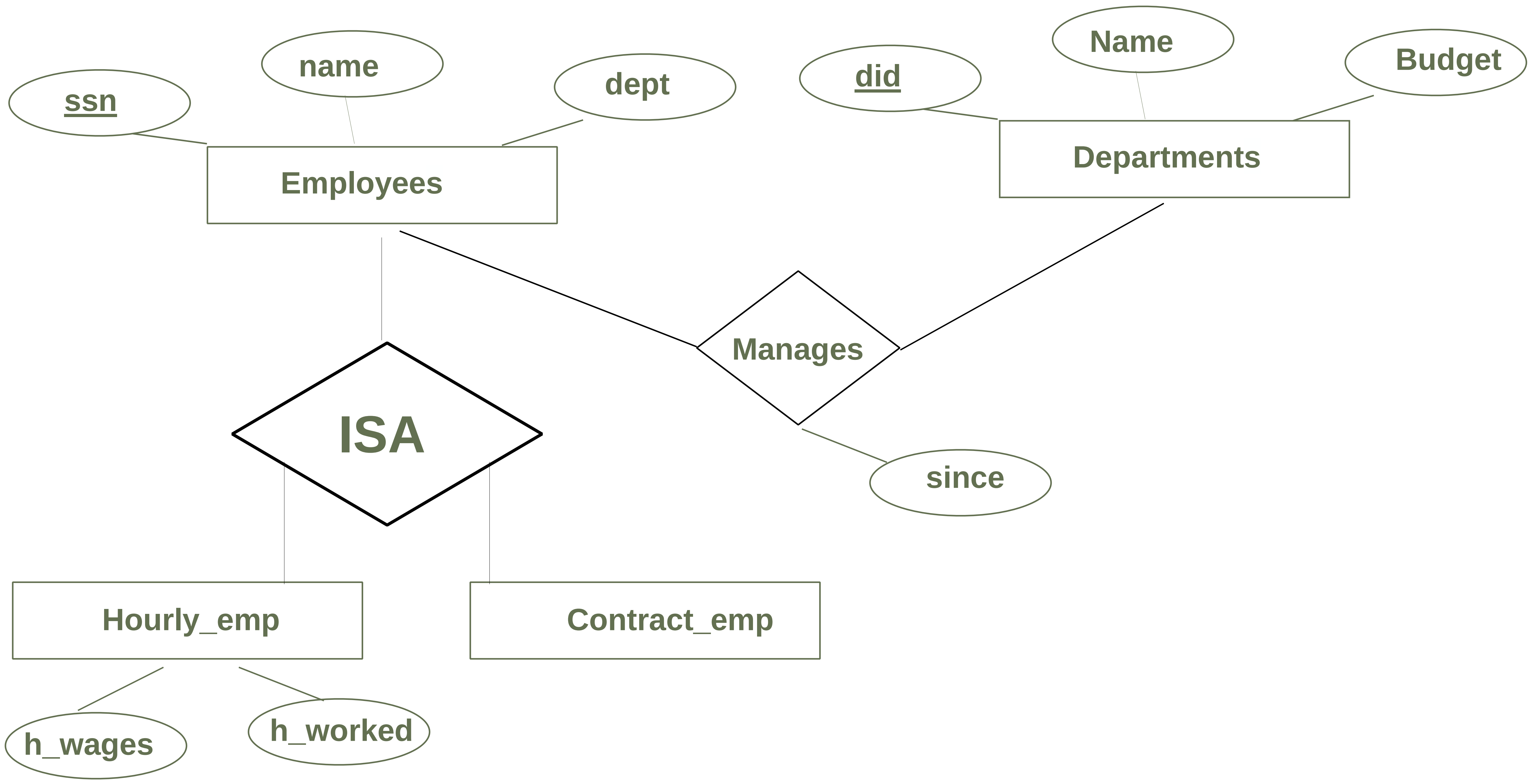
Ternary Relationship



Hierarchies

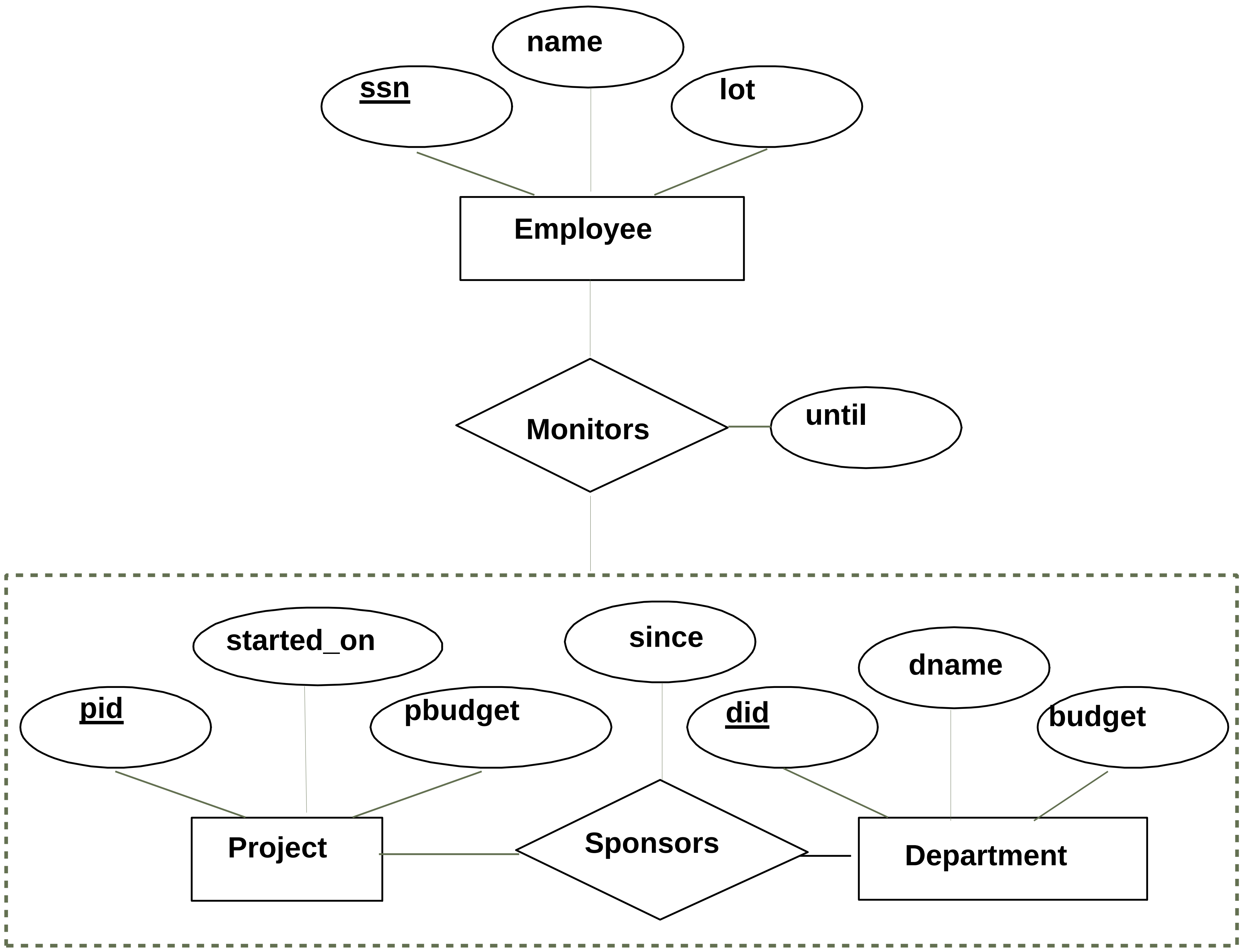
ISA (‘is a’)

If we declare A **ISA** B, every A entity is also considered to be a B entity.

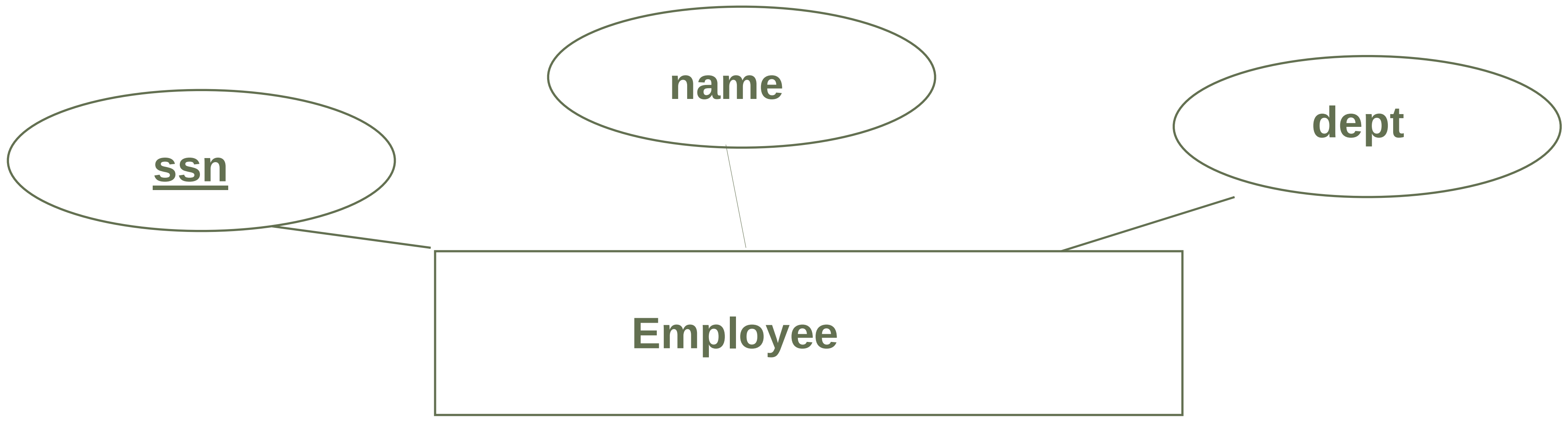


Aggregation

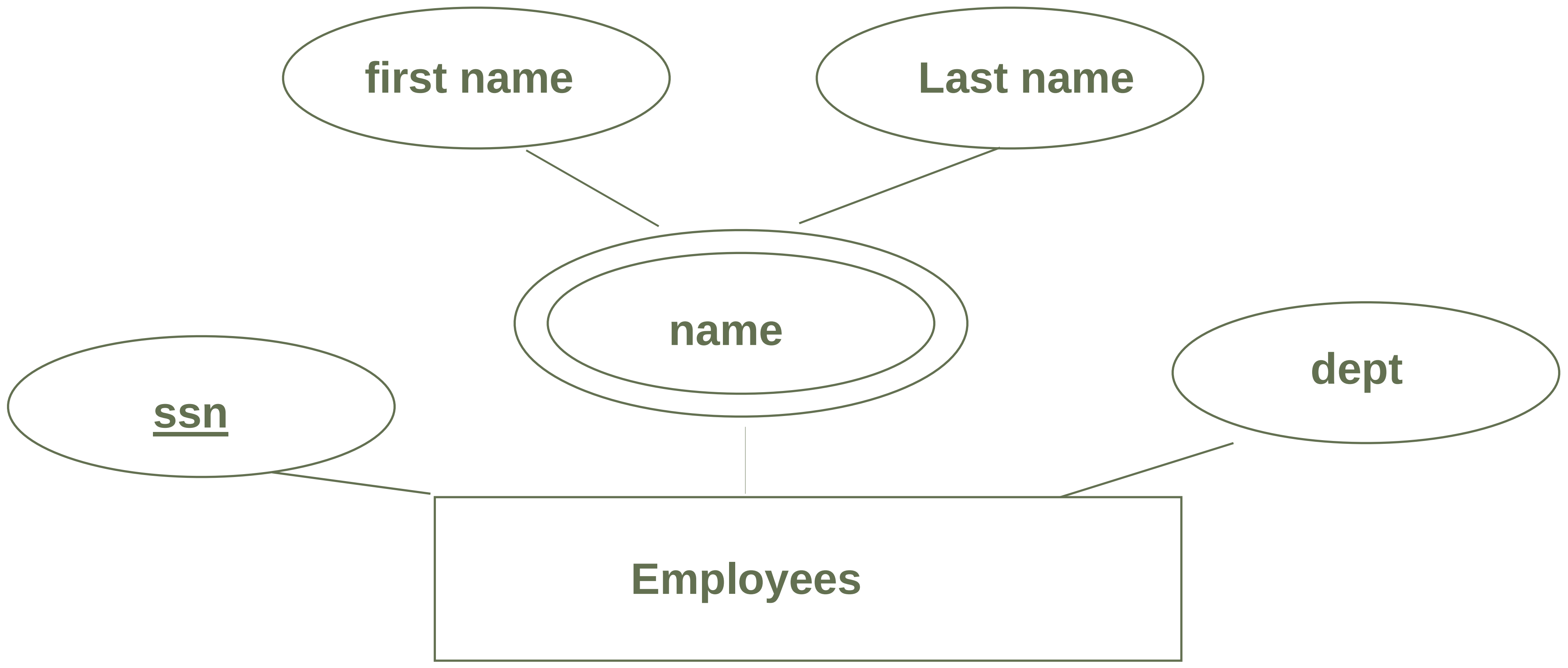
allows us to treat a relationship set as an entity set for the purposes of participating in another relationship.


name
ssn
lot
Employee
Monitors
until
started_on
since
dname
pid
pbudget
did
budget
Project
Sponsors
Department

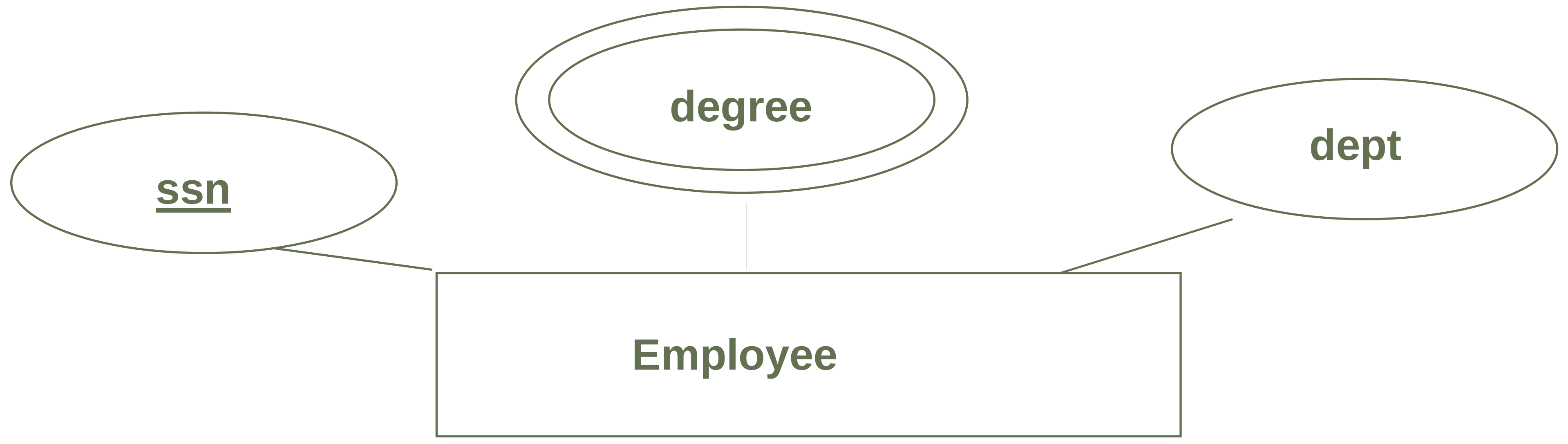
ER MODEL ATTRIBUTES Key Attributes



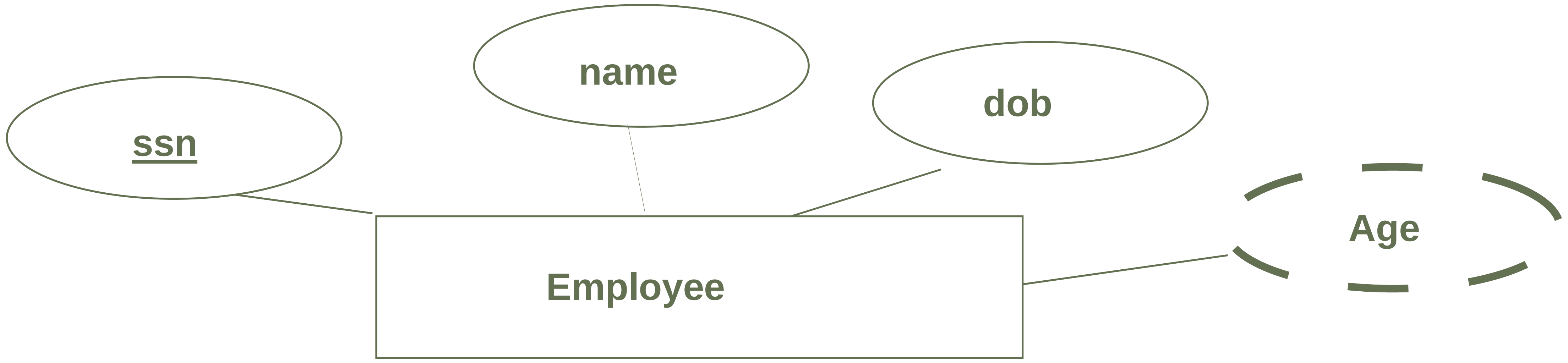
Composite Attributes



Multivalued Attributes



Derived Attributes



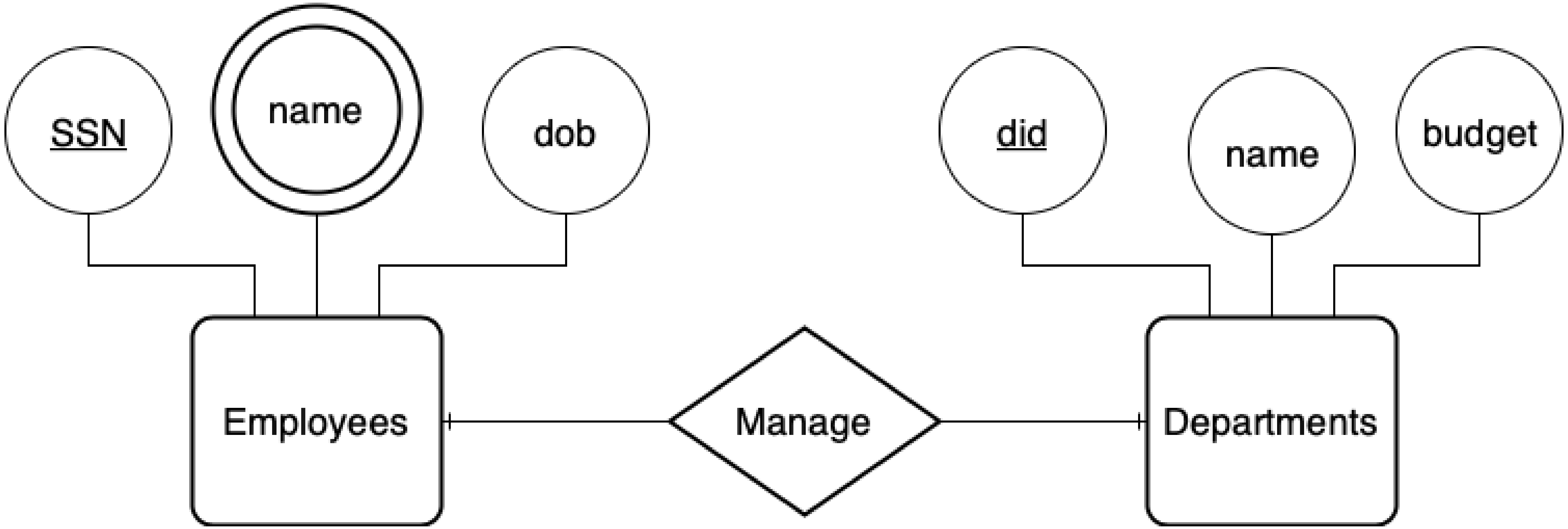
ER MODEL

ENTITY SETS RELATIONSHIPS

AND CONSTRAINTS

Entity Sets Relationships
Degrees of Cardinality

1:1 Relationship



An employee shall manage **only one** department
A department shall have **only one** employee as a manager

Entity Sets Relationships Degrees of Cardinality

1:1 Relationship

Employee

ssn	name	dob
617335456	John	08/03/75
345444567	Mary	03/06/90
345223456	Jane	05/12/99

Department

did	name	budget	manager
1234	eng	35000	617335456
1235	marketing	50000	345444567
1236	HR	30000	345223456

Manager is FK and unique

An employee shall manage **only one** department
A department shall have **only one** employee as a manager

Entity Sets Relationships Degrees of Cardinality

1:1 Relationship

Employee

ssn	name	dol
617335456	John	08/03
345444567	Mary	03/06
345223456	Jane	05/12

Department

did	name	budget
1234	eng	35000
1235	marketing	50000
1236	HR	30000

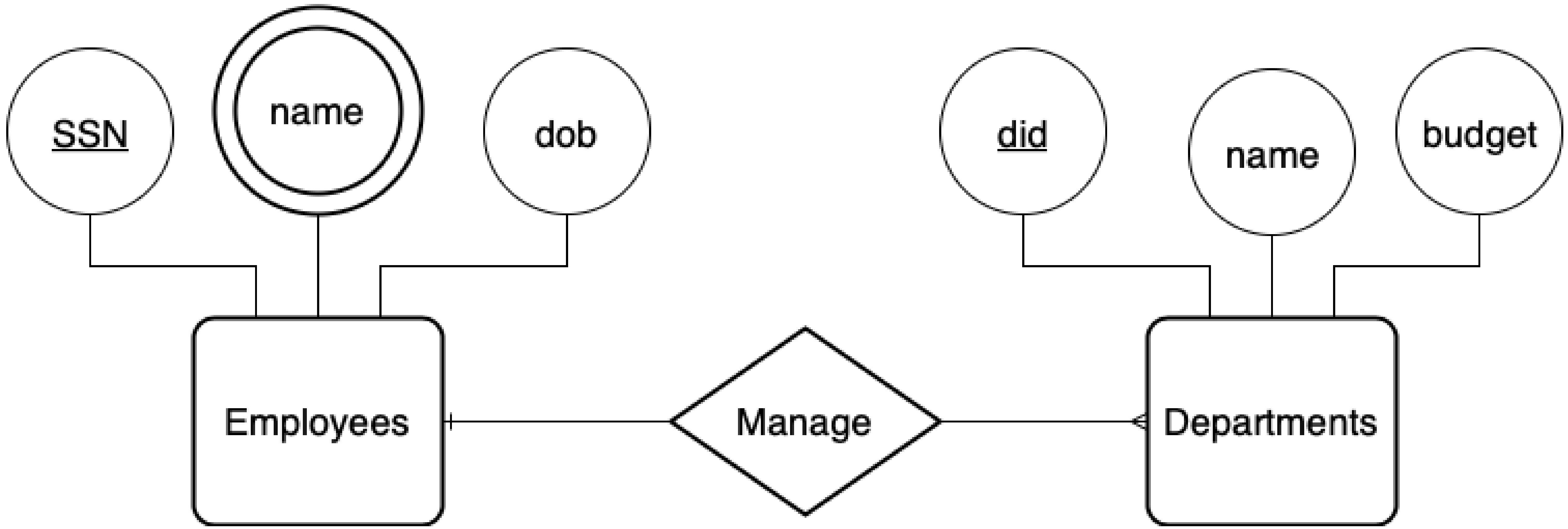
Manages is FK and unique

An employee shall manage **at most one** department

A department shall have **only one** employee as a manager

Entity Sets Relationships
Degrees of Cardinality

1:M Relationship



An employee shall manage **multiple** departments
A department shall have **only one** employee as a manager

Entity Sets Relationships Degrees of Cardinality

1:M Relationship

Employee

ssn		dob
617335456	John	08/03/75
345444567	Mary	03/06/90
345223456	Jane	05/12/99

Department

did	name		manager
1234	eng	35000	617335456
1235	marketing	50000	617335456
1236	HR	30000	345223456

Entity Sets Relationships

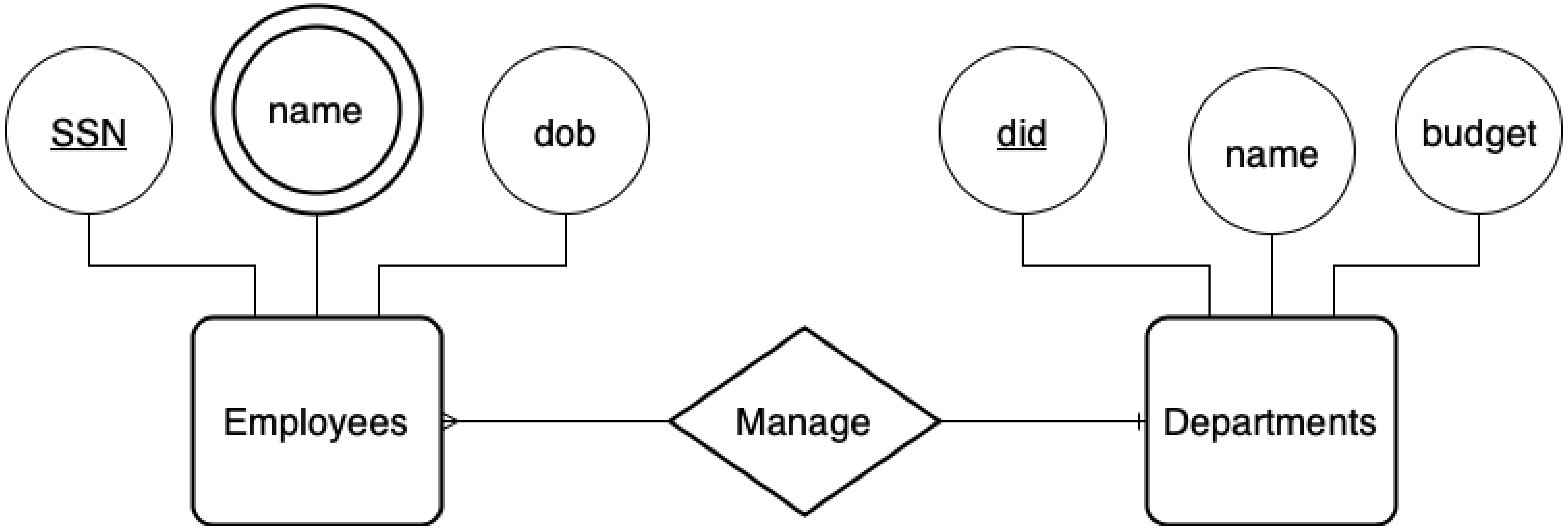
Manager is FK and not unique

An employee shall manage **multiple** departments
A department shall have **only one** employee as a manager

Degrees of Cardinality

Entity Sets Relationships

M:1 Relationship



An employee shall manage **only** in one department
A department shall be managed by **multiple** managers

Entity Sets Relationships Degrees of Cardinality

M:1 Relationship

Employee

ssn	name	dob	dept
617335456	John	08/03/75	1234
345444567	Mary	03/06/90	1234
345223456	Jane	05/12/99	1236

Department

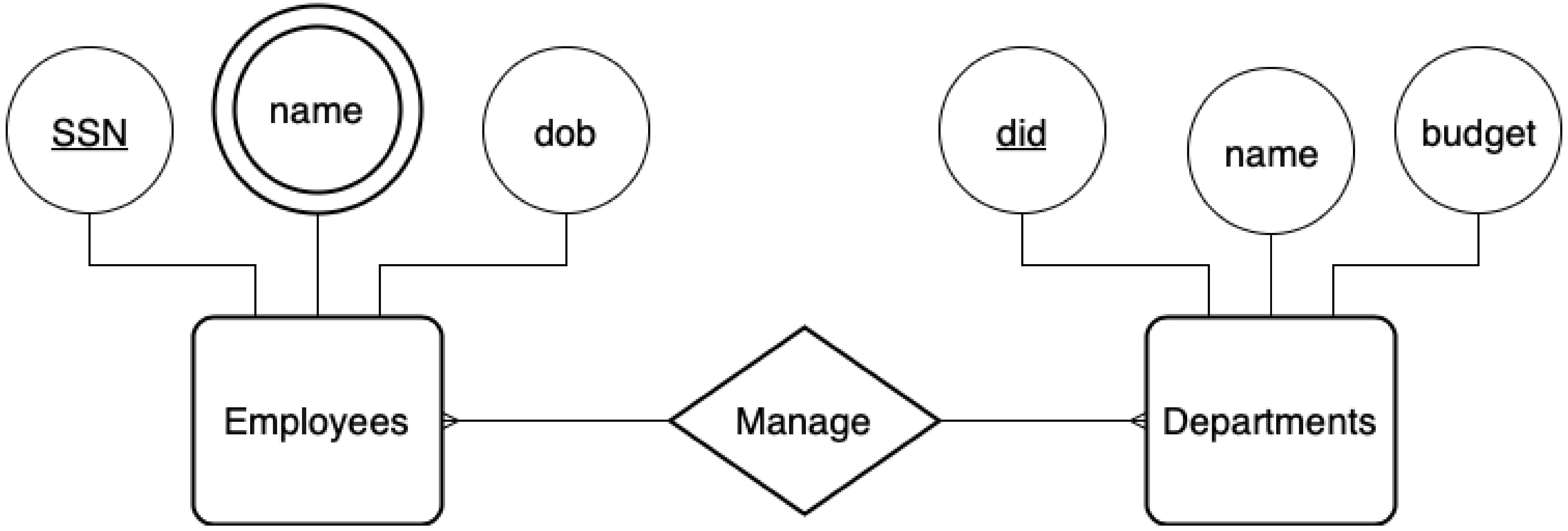
did	name	budget
1234	eng	35000
1235	marketing	50000
1236	HR	30000

dept is FK and not unique

An employee shall manage **only** one department
A department shall be managed by **multiple** managers

Entity Sets Relationships Degrees of Cardinality

M:N Relationship



An employee shall manage **multiple** departments
A department shall be managed by **multiple** employee

Entity Sets Relationships Degrees of Cardinality

M:N Relationship

Employee

ssn	name	dob
617335456	John	08/03/75
345444567	Mary	03/06/90
345223456	Jane	05/12/99

Managers

manager	dept
617335456	1234
345444567	1234
345223456	1235
345223456	1236

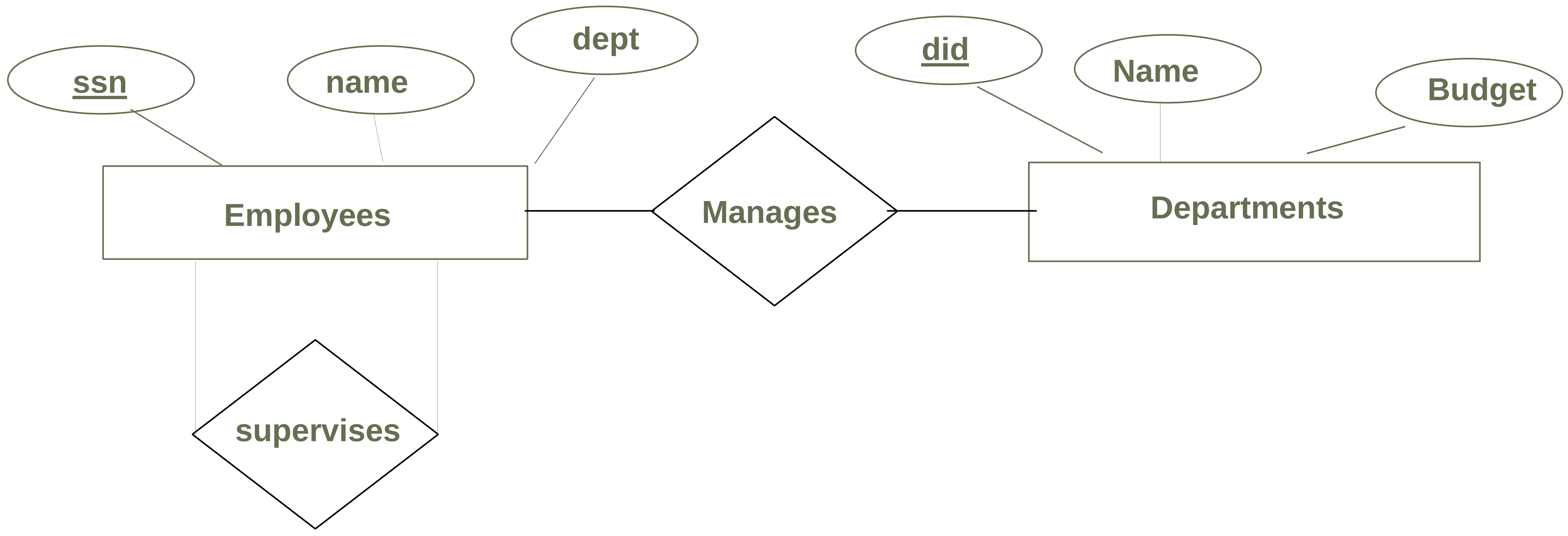
Department

did	name	Budget
1234	eng	35000
1235	market	50000
1236	HR	30000

An employee shall manage **multiple** departments
A department shall be managed by **multiple** employee

Entity Sets Relationships Degrees of Cardinality

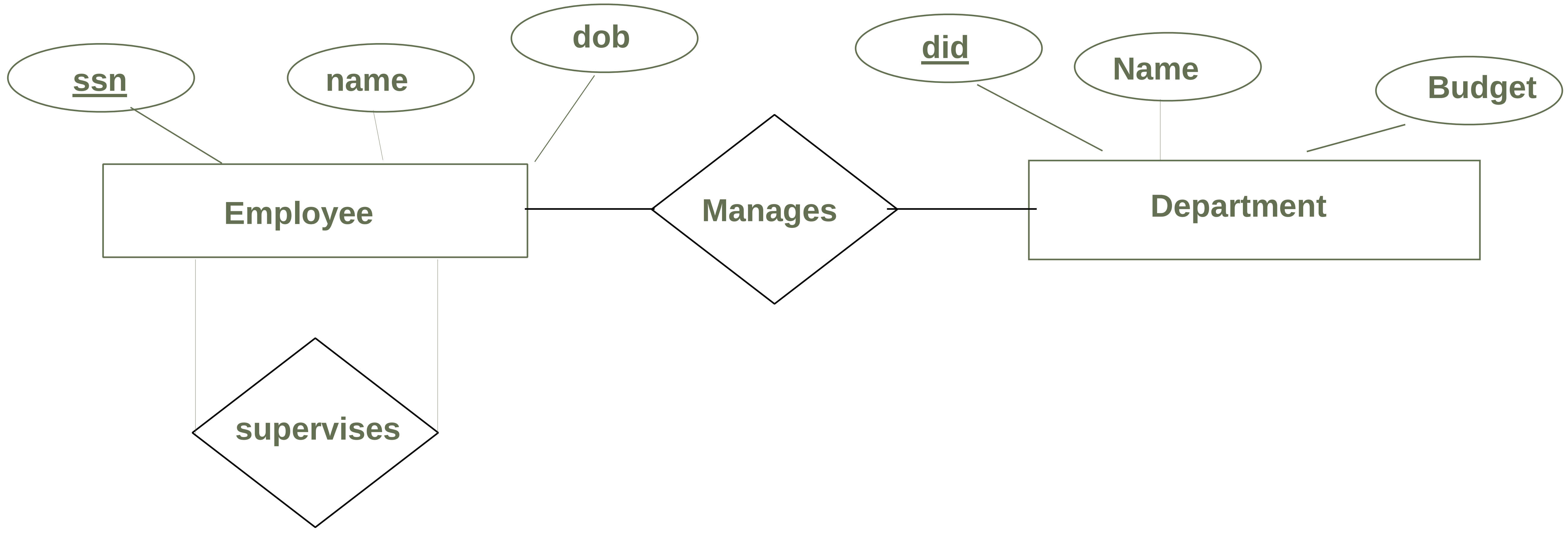
Recursive Relationship



Employees can supervise other employees

Entity Sets Relationships Degrees of Cardinality

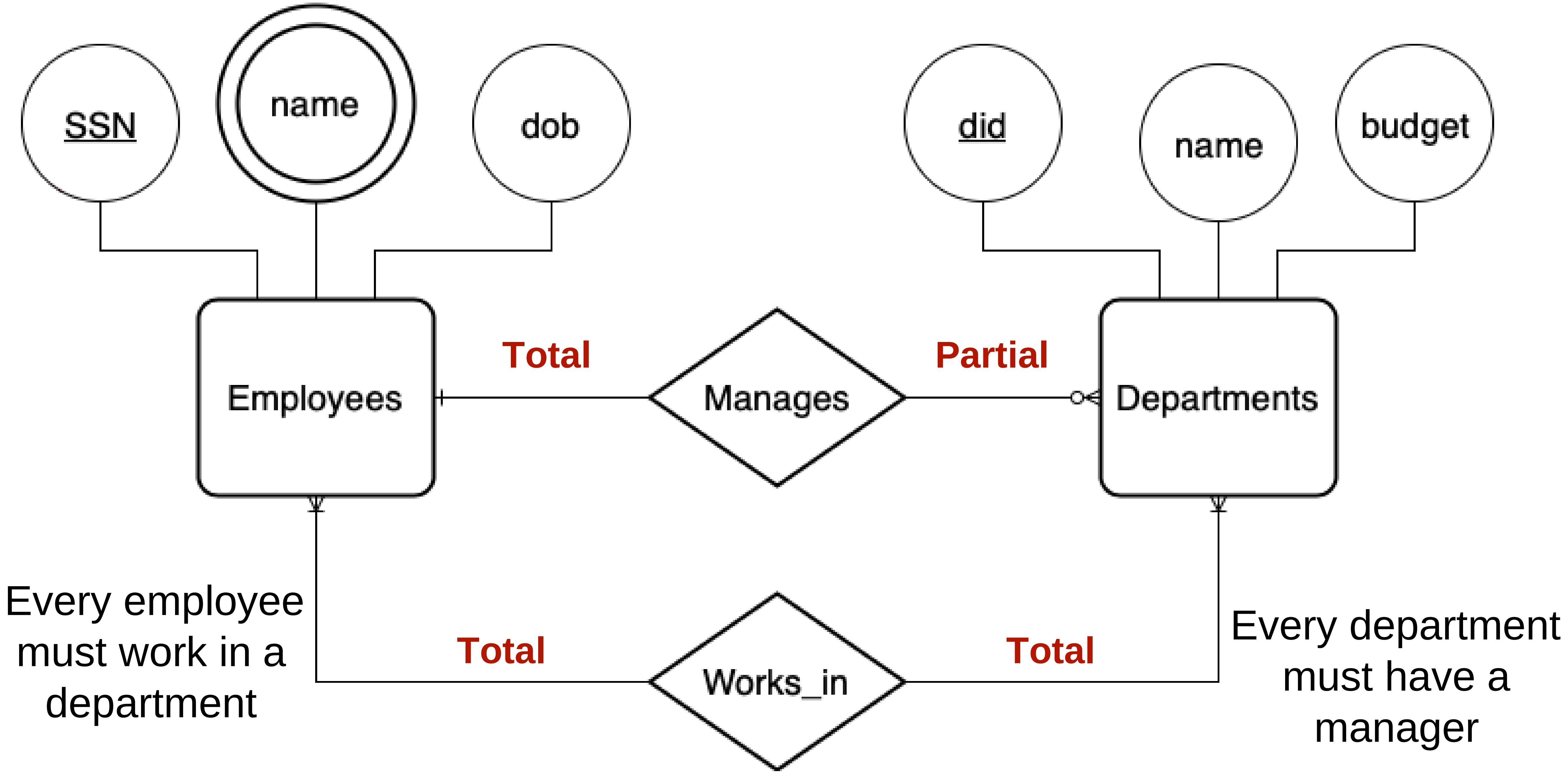
Recursive Relationship



ssn	name	dob	supervised_by
617335456	John	08/03/75	345444567
345444567	Mary	03/06/90	617335456

Type of Constrains

- Keys are attributes or sets of attributes that uniquely identify an entity within its entity set.
- Single-value constraints require that a value be unique in certain contexts.
- Referential integrity constrains or participation constrains require that a value referred to actually exists in the database. Total and partial
- Domain constraints specify what set of values an attribute can take.
- General constraints are arbitrary constraints that should hold in the database.
- **Constraints are part of the schema of a database.**



Participation Constraints

Every employee
must manage a department?

NO!!!

Every department
must have an
employee?

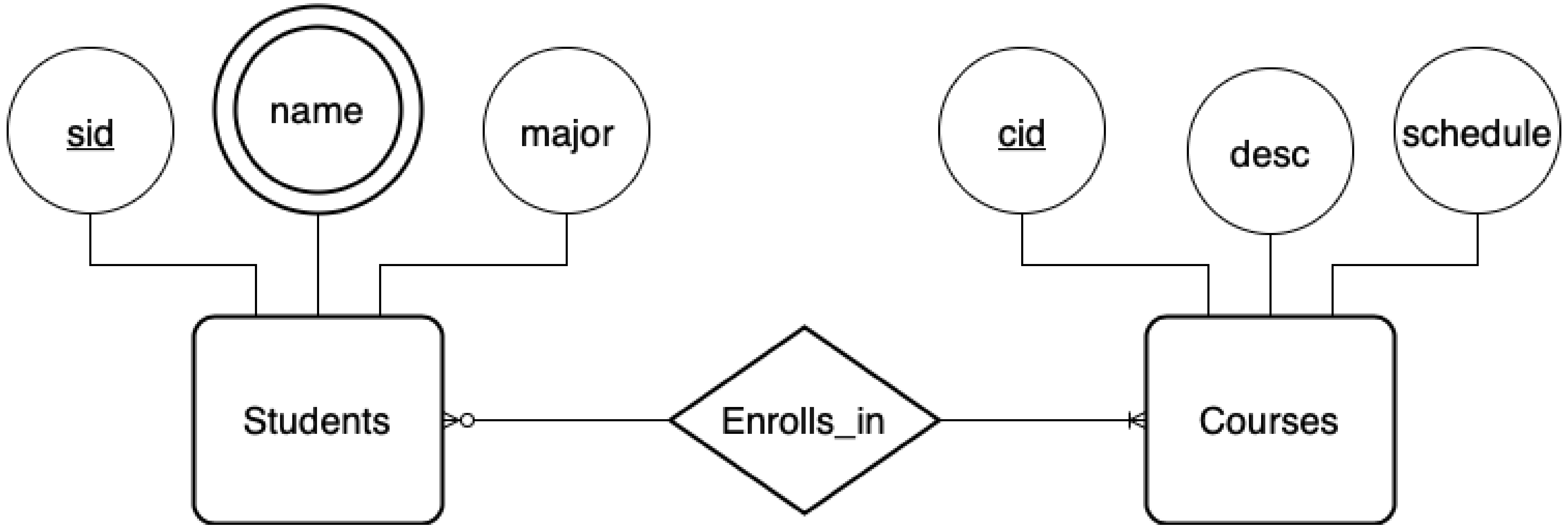
YES!!!

Participation Constraints and Cardinality Constraints

- Rules:
- Cardinality Constraints: total or partial participation
- Total Participation Constraints: At least or Minimum
- Weak entities: must have total participation

- Strong entities: may have total or partial participation

Class Exercise Reading a ERD Model



1. How many Entity Set are in this model?

Two entity sets: Students and Courses

2. How many Relationships are in this model?

One relationship: Enrolls_in

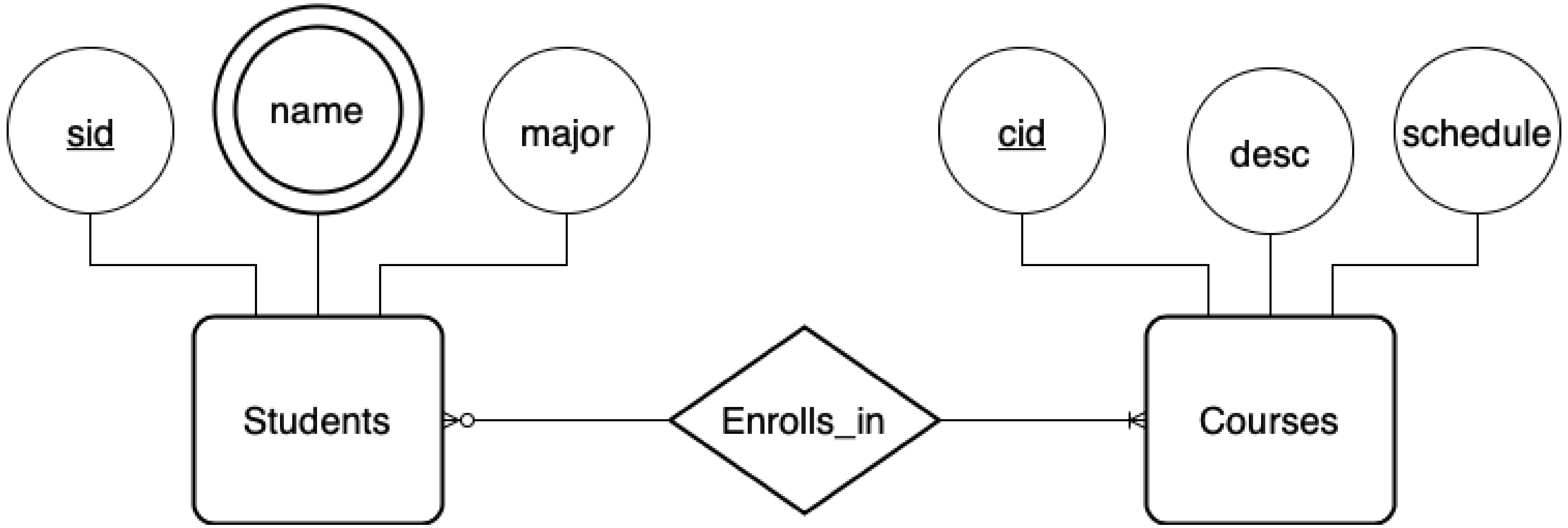
3. What are the keys constraints in this model?

PK: sid : distinct students in the Students entity set

PK: cid: distinct courses in the Course entity set

FK: cid, and sid must exist in the Enroll relationship as foreign keys

Class Exercise
Reading a ER Model



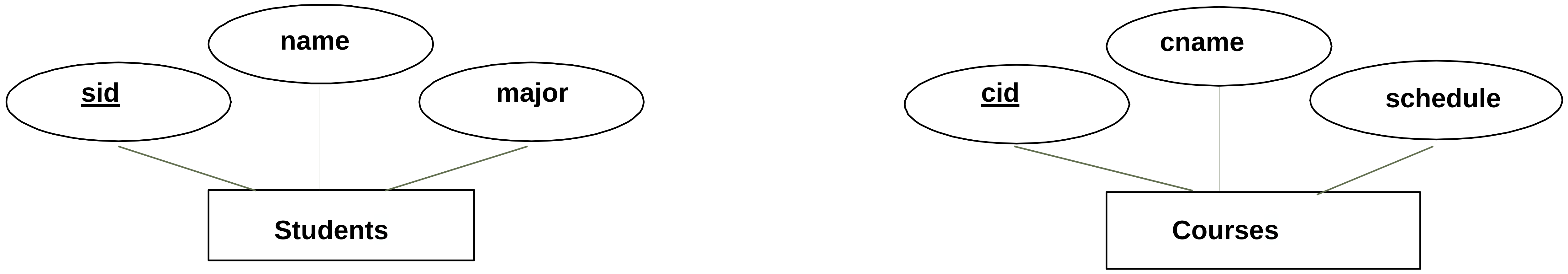
4. Which are the cardinality constraints in this model?

Many-to-Many: A student must be enrolled in at least one course, and a course can have zero, one or multiple students. **5. Which are the participation constraints in this model?**

Students entity set has a partial participation.

So, courses can have zero, one or many students enrolled. However, Courses entity set has total participation, meaning that Students must be enrolled in at least one course.

Class Exercise Reading a ER Model



6. Define which entity set is strong or weak.

Both of them are strong. Students can exist without being enrolled in any course, and courses can exist without students enrolled.

7. Prove that Students and Courses are strong entity sets.

Let’s create an ERD representing a store ordering database

Business Rules

1. Customers can order multiple items.

2. Customers pay one order at a time.

3. Supplier sends multiple stocks by request

4. Requests can have many stocks

5. Stocks must contain at least one product

6. Orders, payments and stocks can be monitored by the system

Identifying Entities and Attributes

- **Customer**:
- cid: PK, name: composite (first name, last name), dob: multi-value
- **Supplier:**
- sid: PK, organization
- **Stock:**
- stkid: PK, supplier: FK, num_products: derived
- **System:**

- Stid: PK, oid: FK, sid: FK

Identifying Entities and Attributes

- **Product:**
- pid: PK, oid: FK, barcode: multi-value (UPC, EAN, ISBN)
- **Invoice:**
- iid: PK, oid: PK, date: multi-value (month, day, year), subtotal, total
- **Orders:**

- oid: PK, quantity

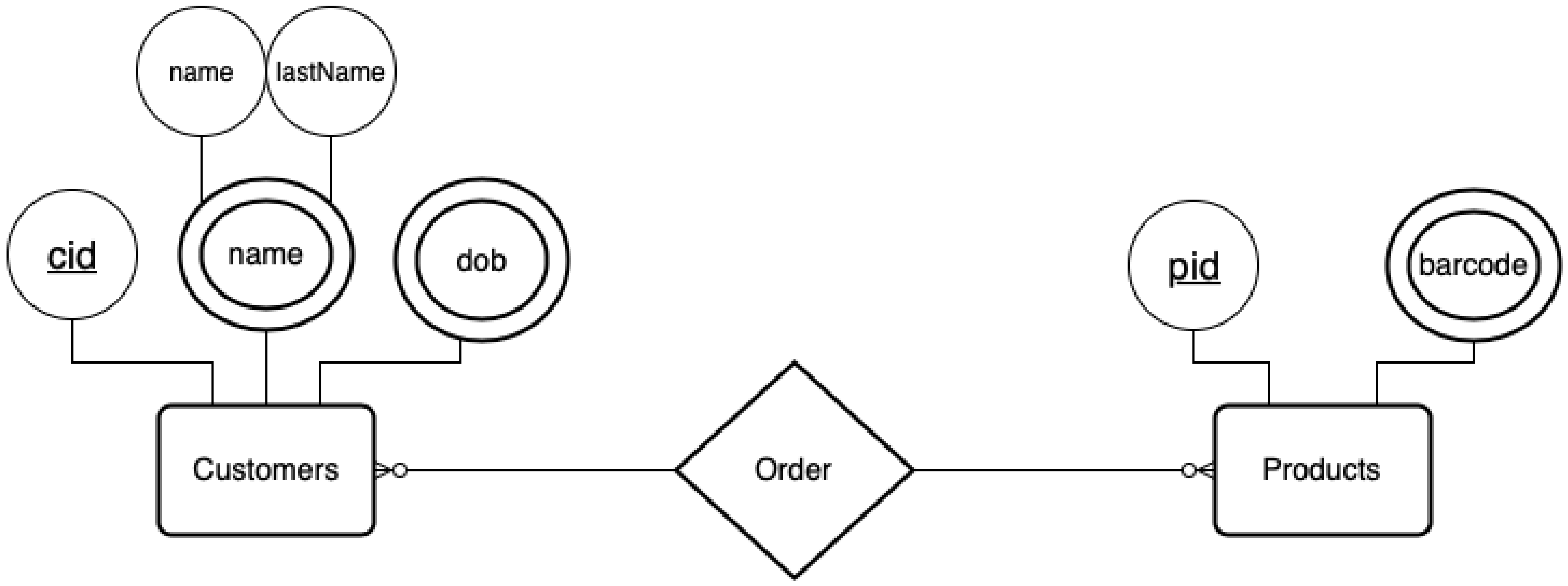
Identifying Strong Entities

- **Strong Entities:**
- Customer
- Supplier
- Stock
- Product
- System
- **Weak Entities:**

- Orders
- Payment

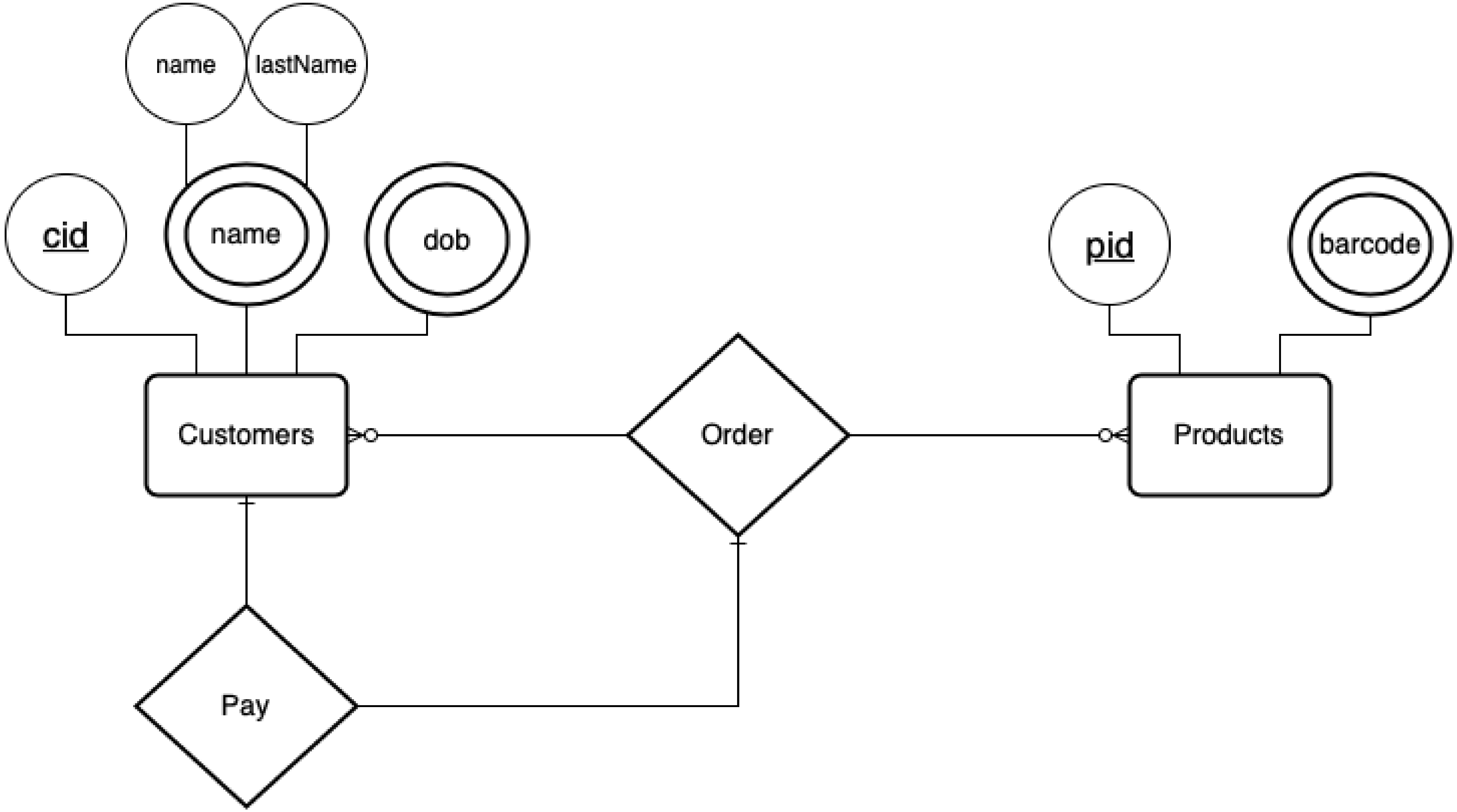
ERD

Customers can order multiple items.



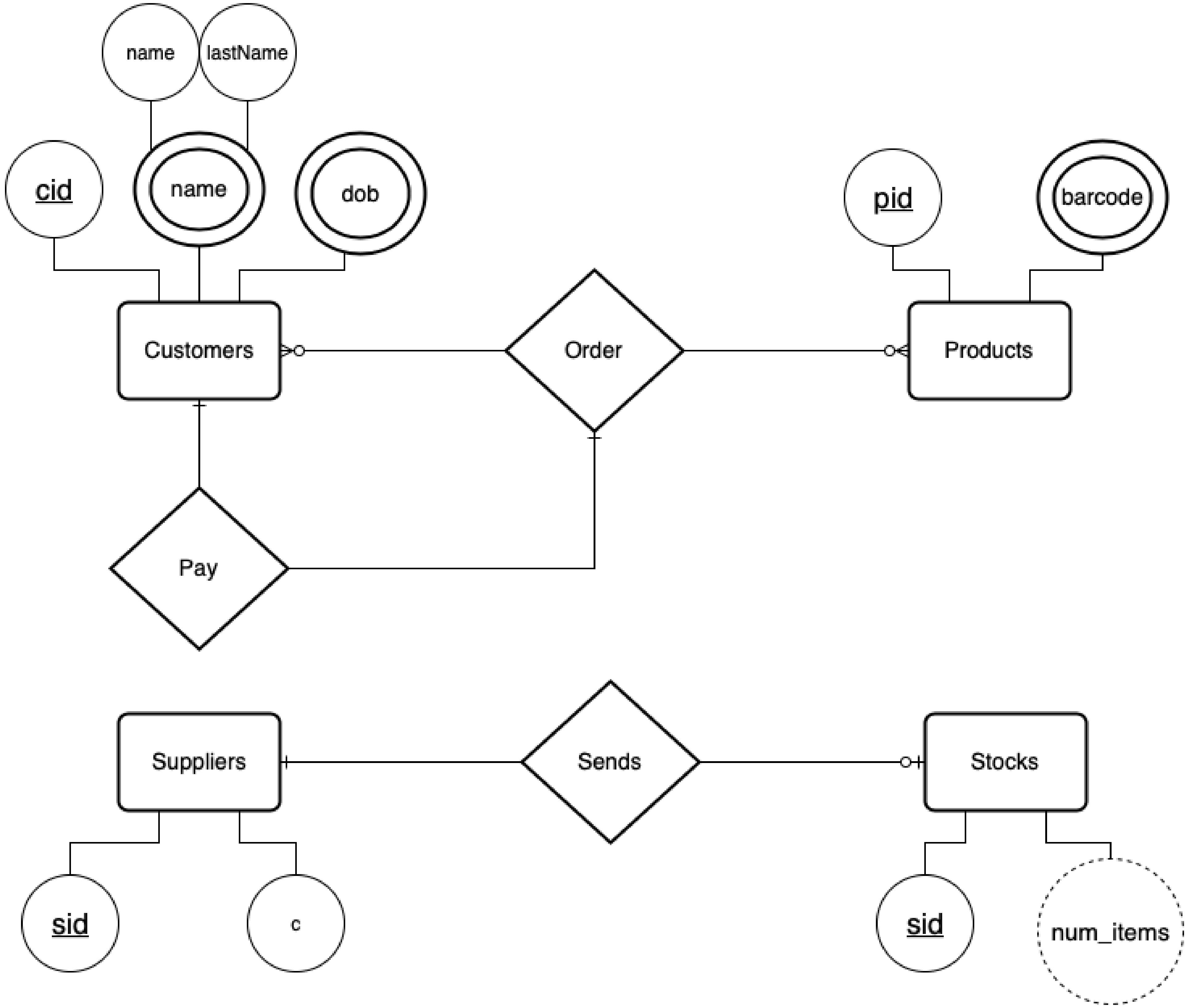
ERD

Customers pay one order at a time.


name
lastName
cid
name
dob
pid
barcode
Customers
Order
Products
Pay

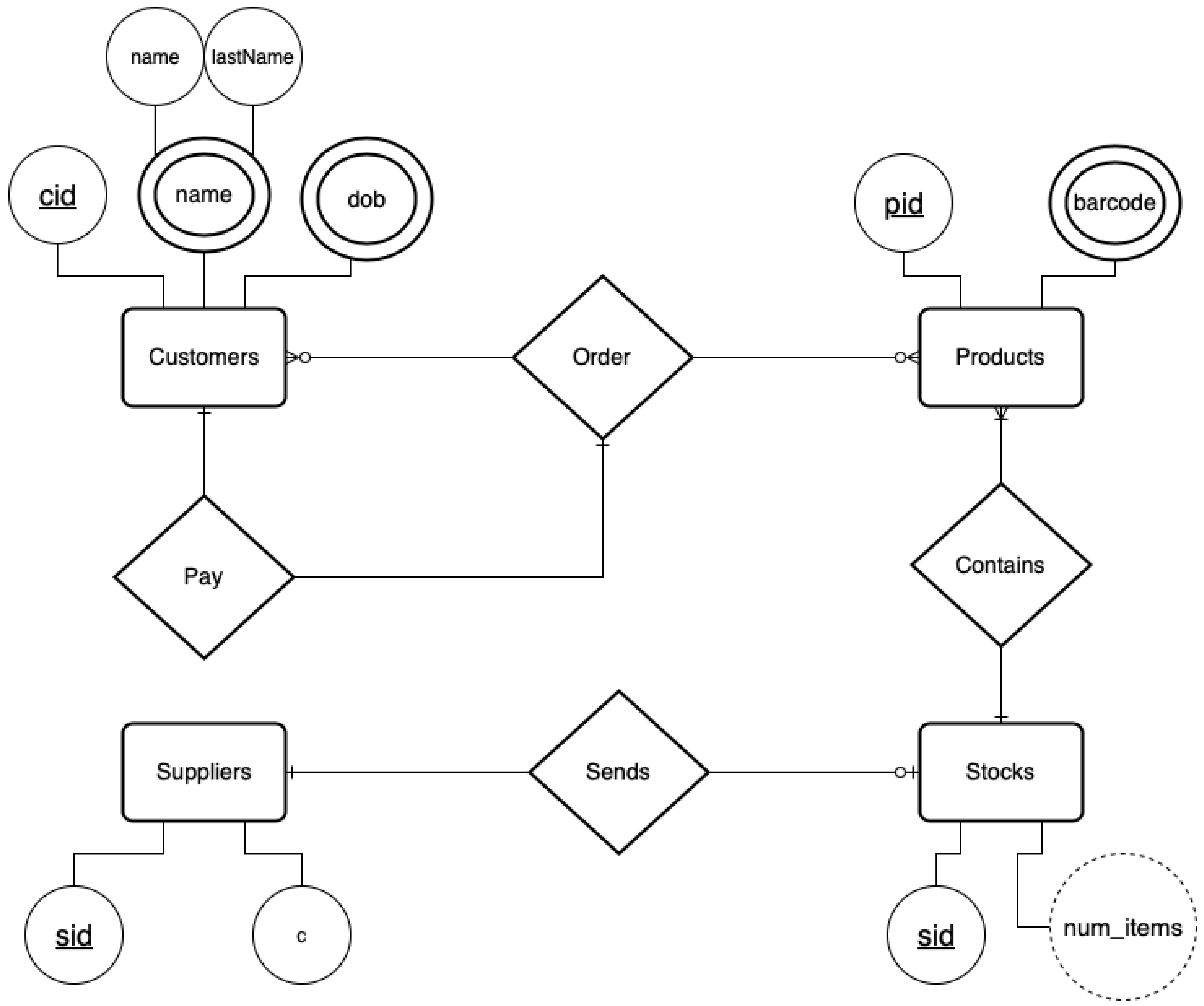
ERD

Supplier sends stocks by request


name
lastName
cid
name
dob
pid
barcode
Customers
Order
Products
Pay
Suppliers
Sends
Stocks
sid
c
sid
num_items

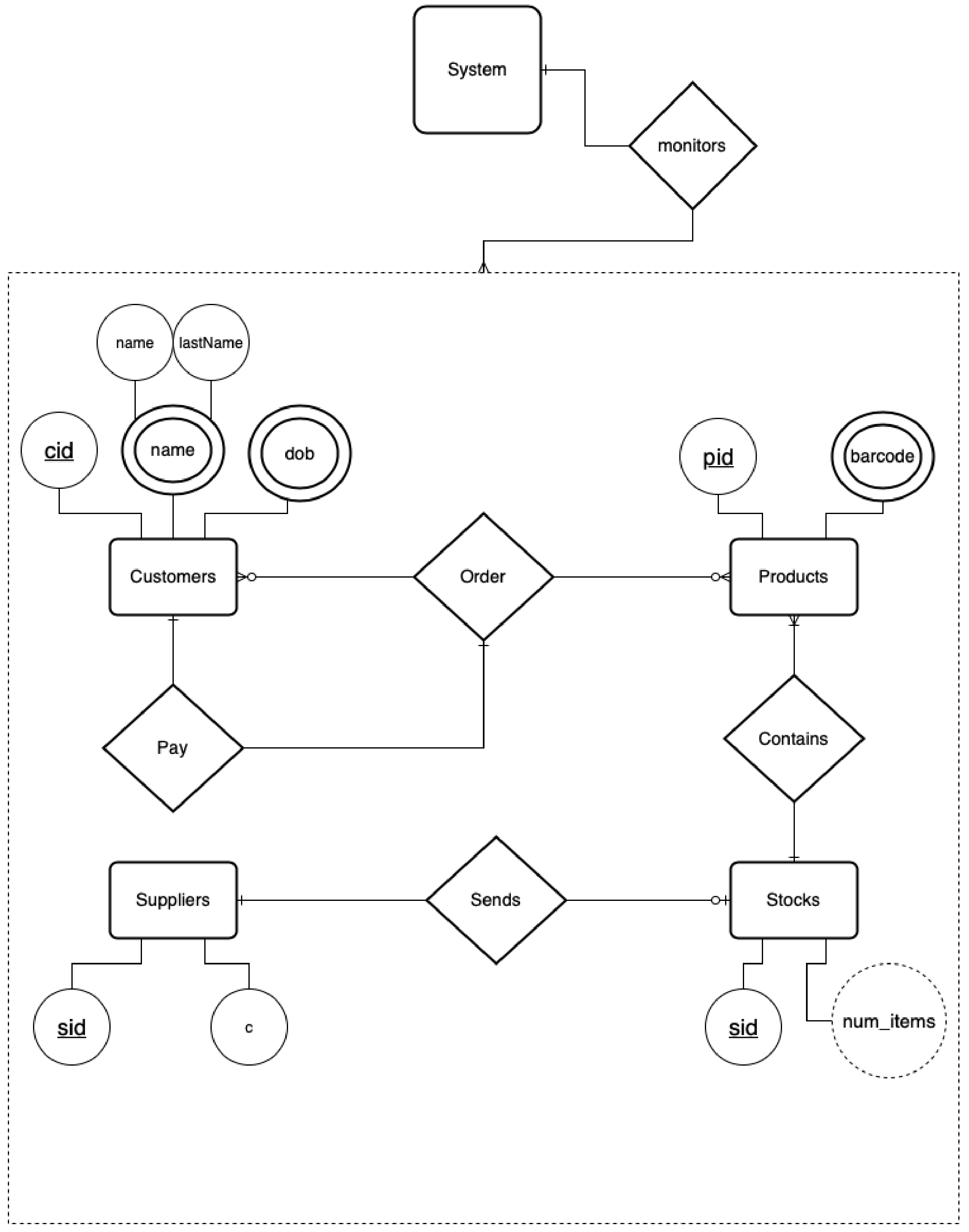
ERD

Stocks must contain one or more products


name
lastName
cid
name
dob
pid
barcode
Customers
Order
Products
Pay
Contains
Suppliers
Sends
Stocks
sid
c
sid
num_items

ERD

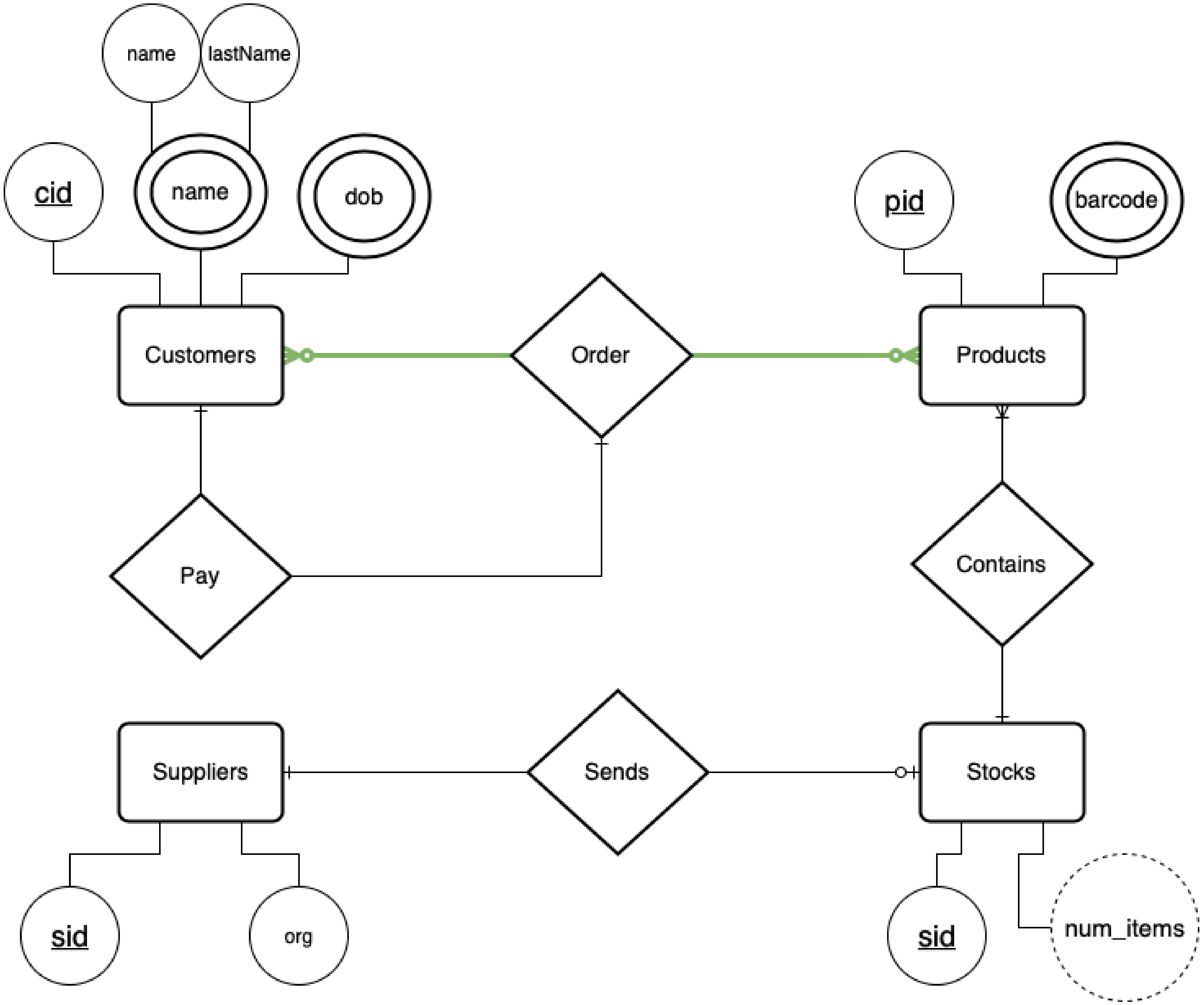
Orders, payments and stocks can be monitored by the system



How to test ERD models

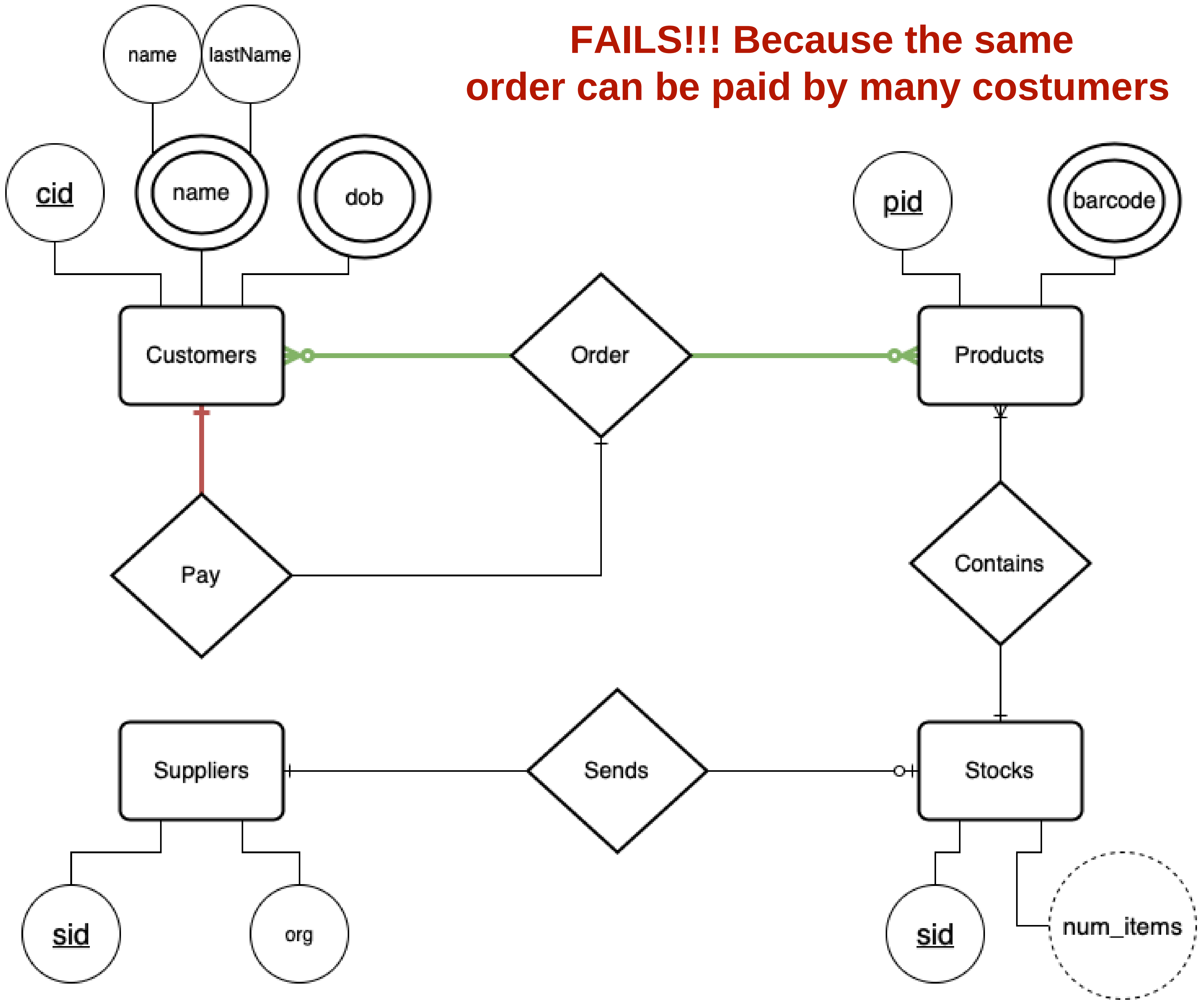
“Customers can order multiple items.”

Testing the system

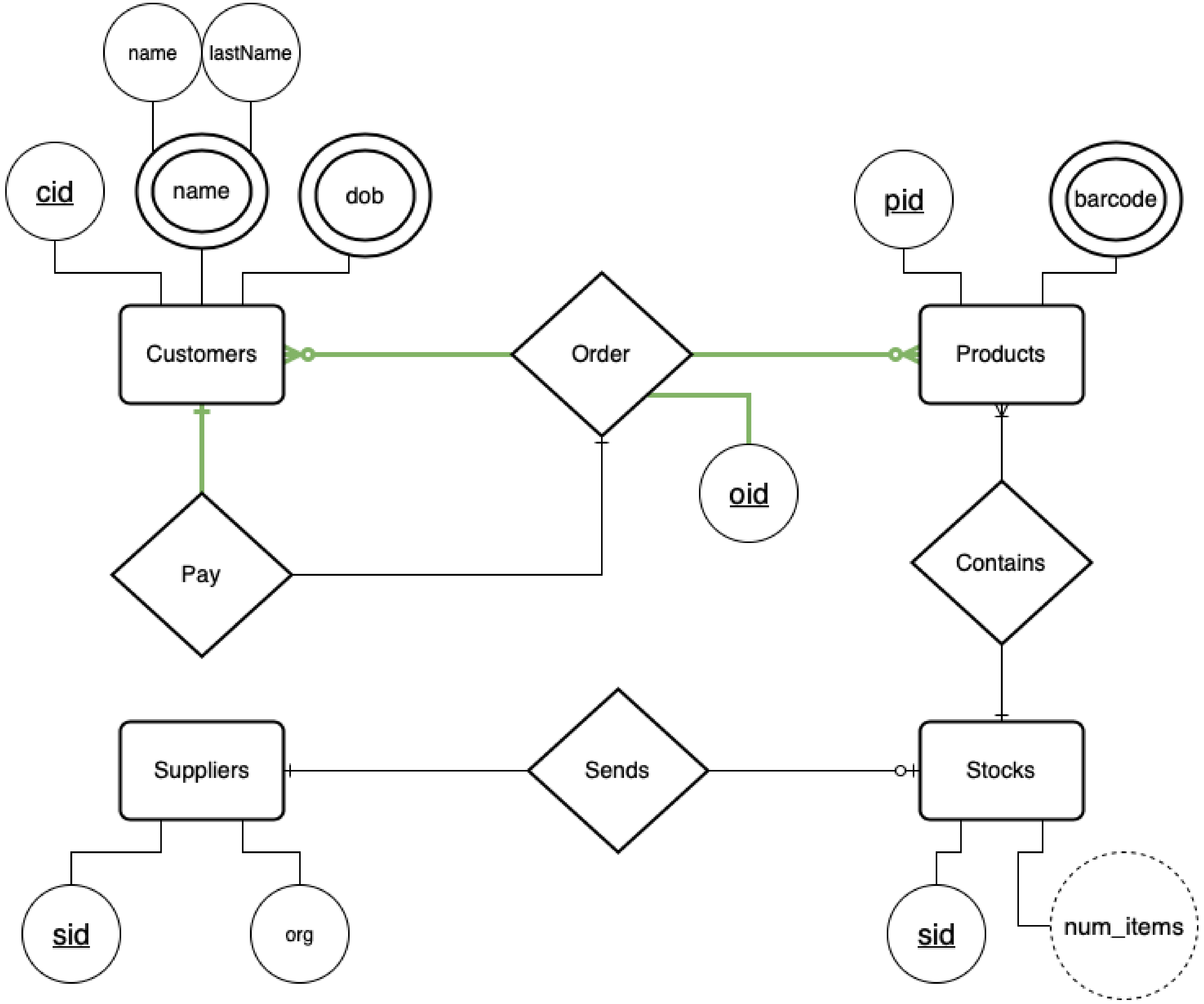


Testing the system
“Customers pay one order at a time..”

FAILS!!! Because the same order can be paid by many costumers

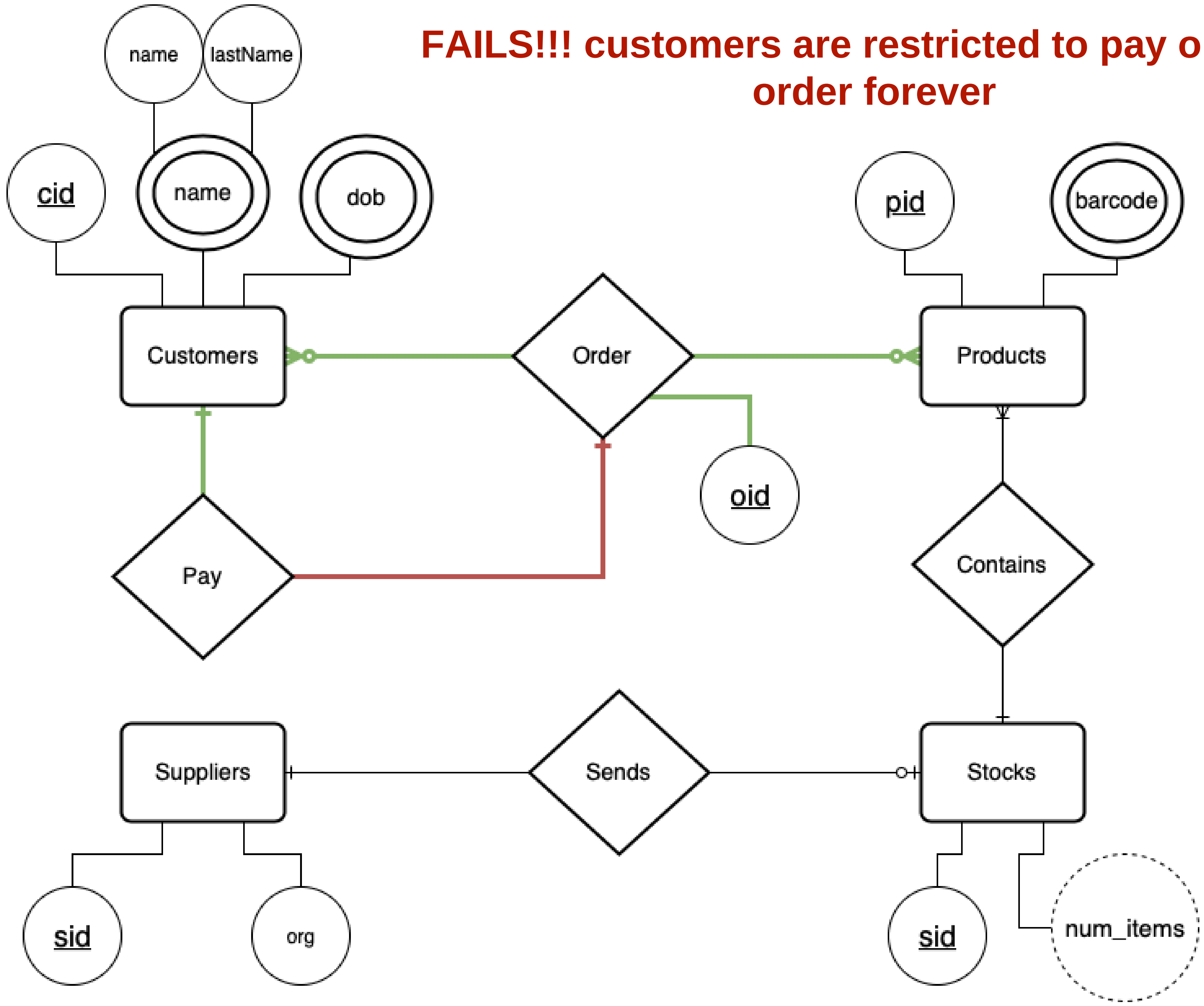


Testing the system
"Customers pay one order at a time.."

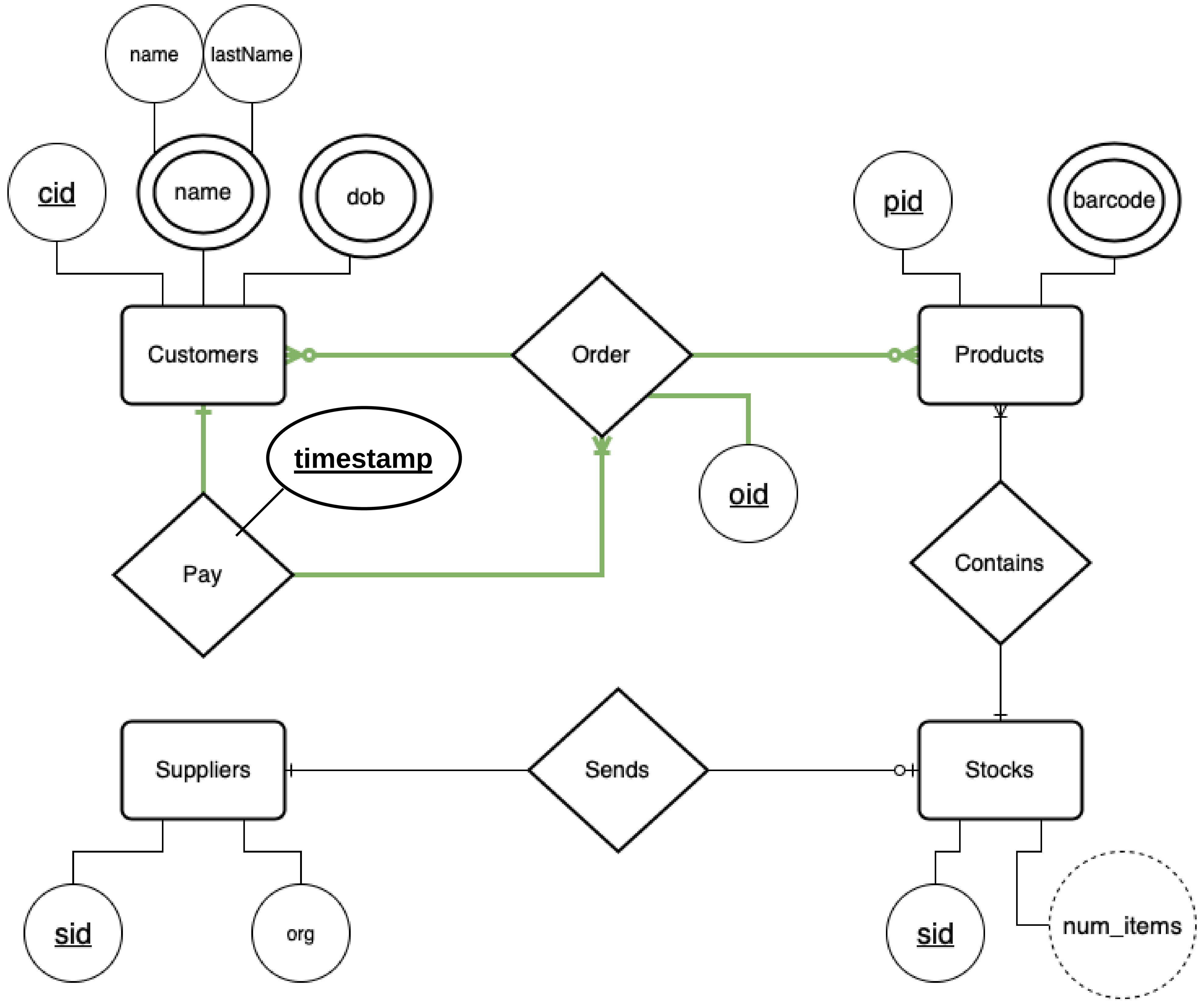


Testing the system
“Customers pay one order at a time..”

FAILS!!! customers are restricted to pay only one order forever

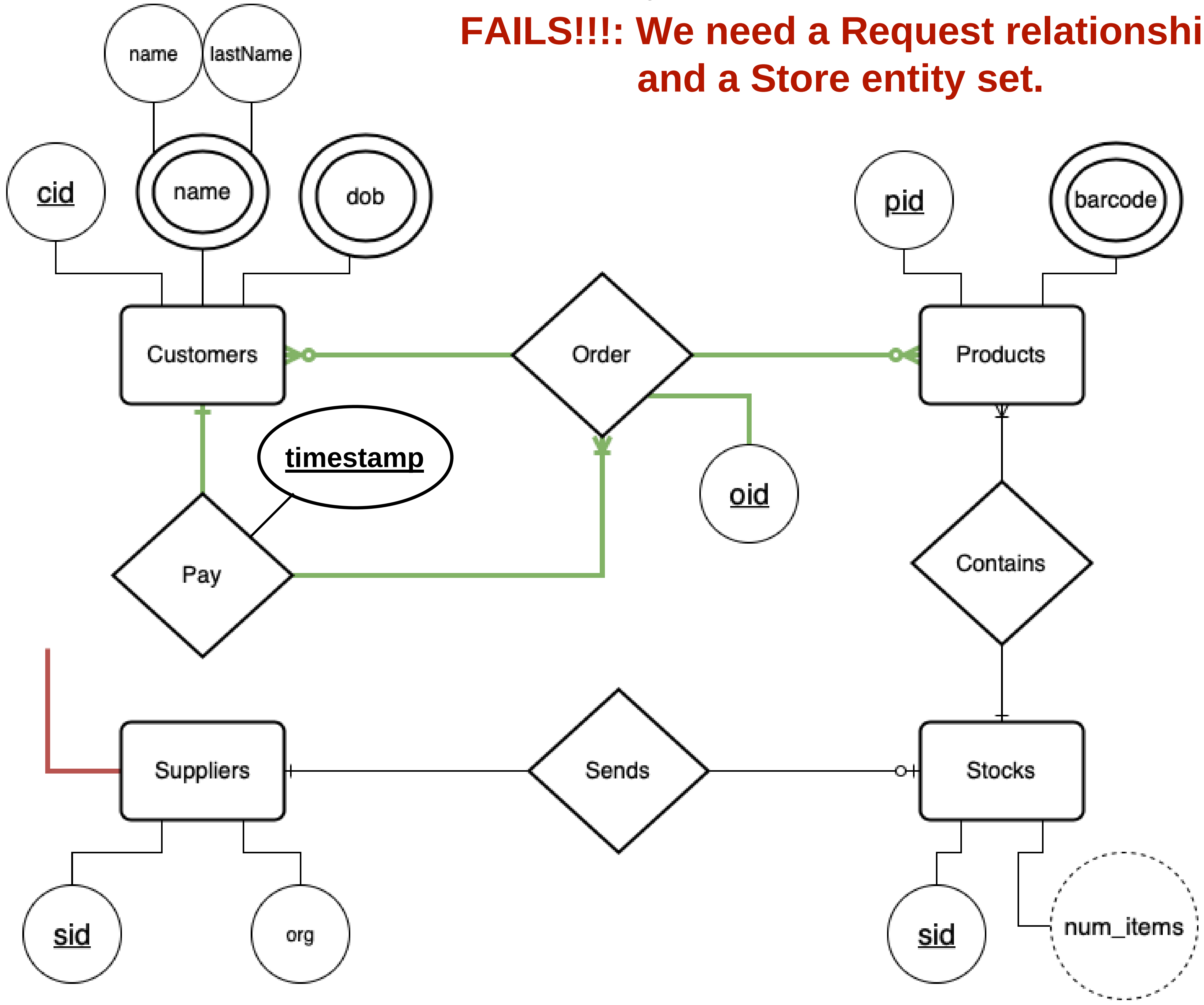


Testing the system
“Customers pay one order at a time..”



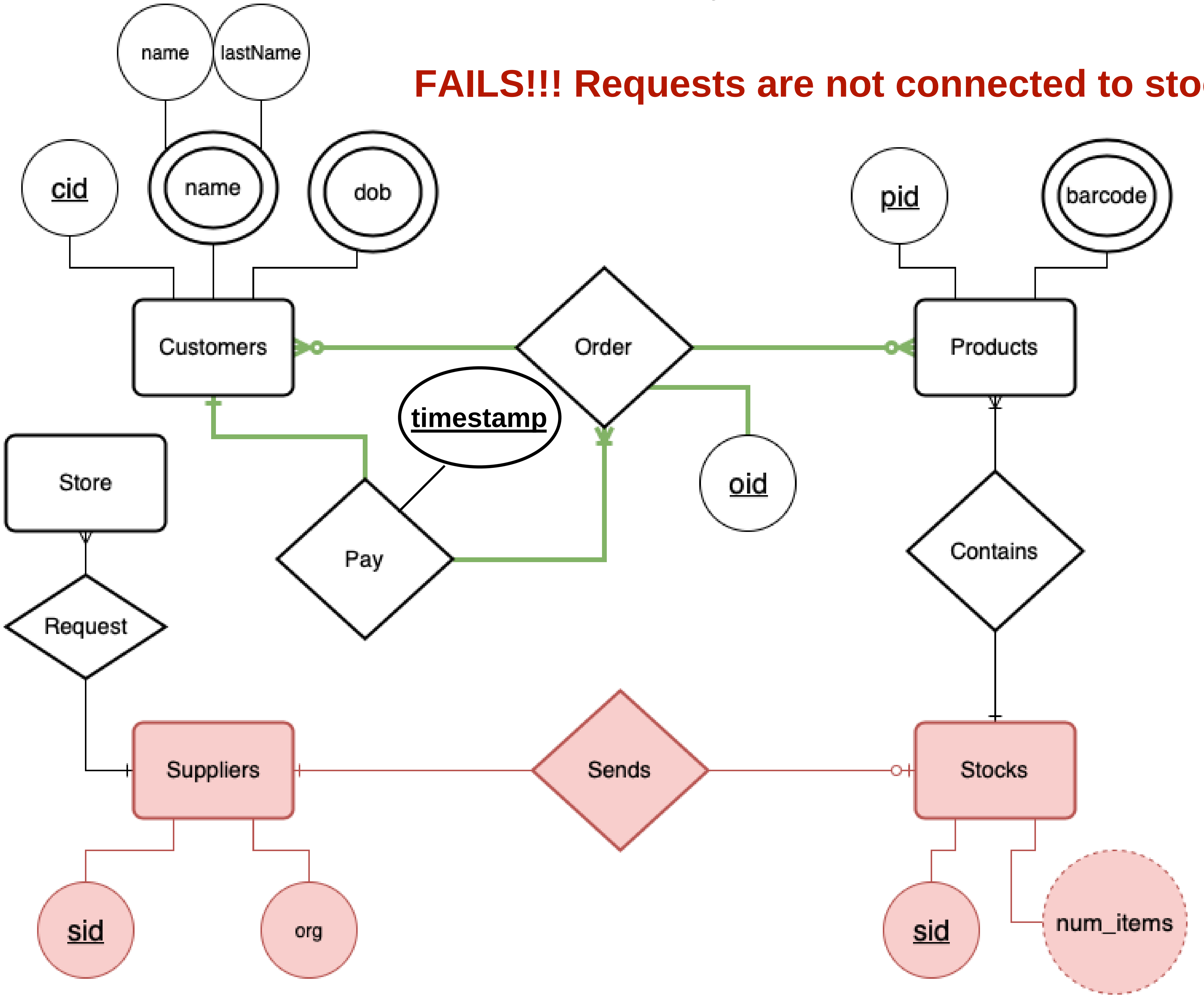
Testing the system “Supplier sends stocks by request”

FAILS!!!: We need a Request relationship, and a Store entity set.

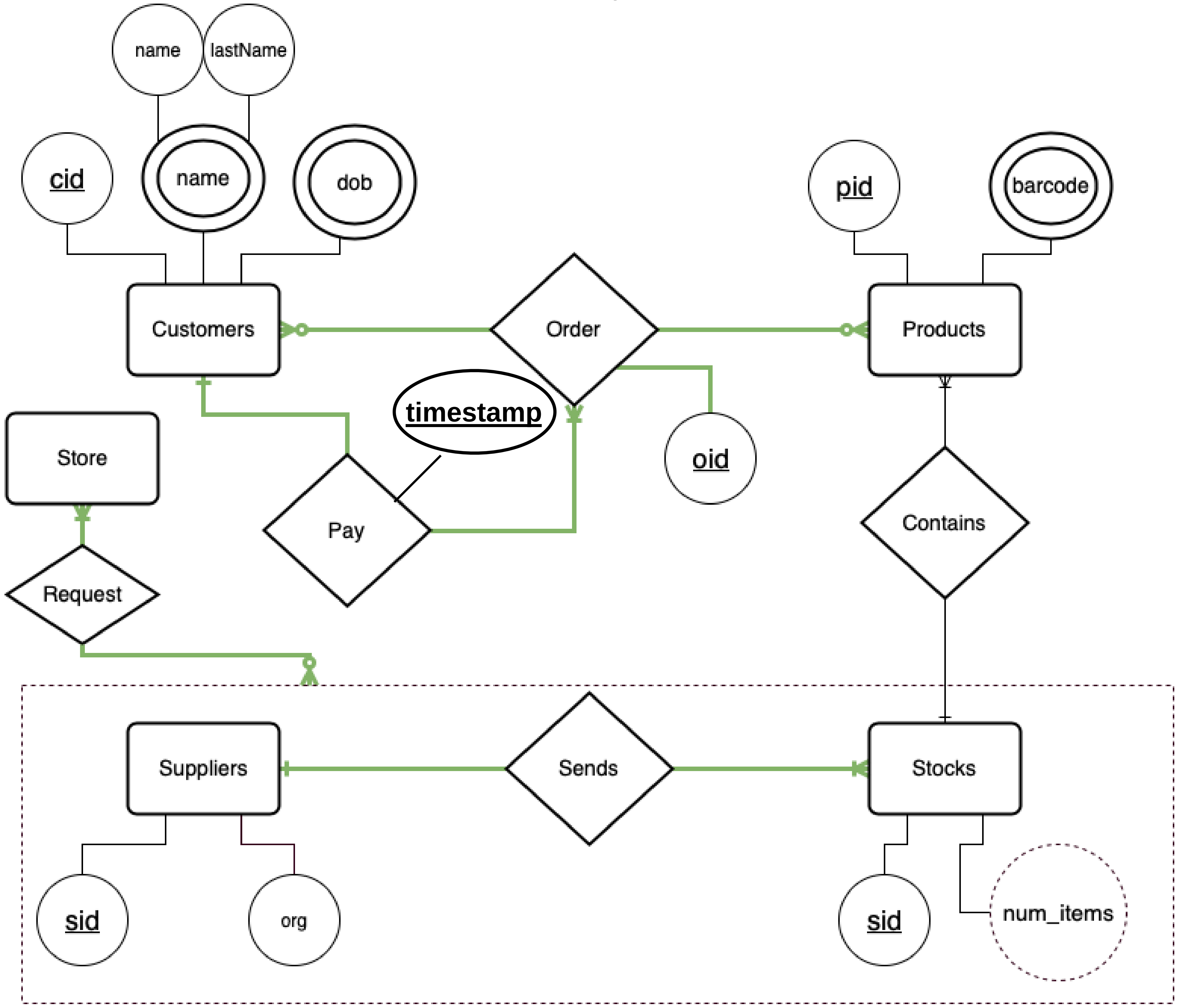


Testing the system “Supplier sends stocks by request”

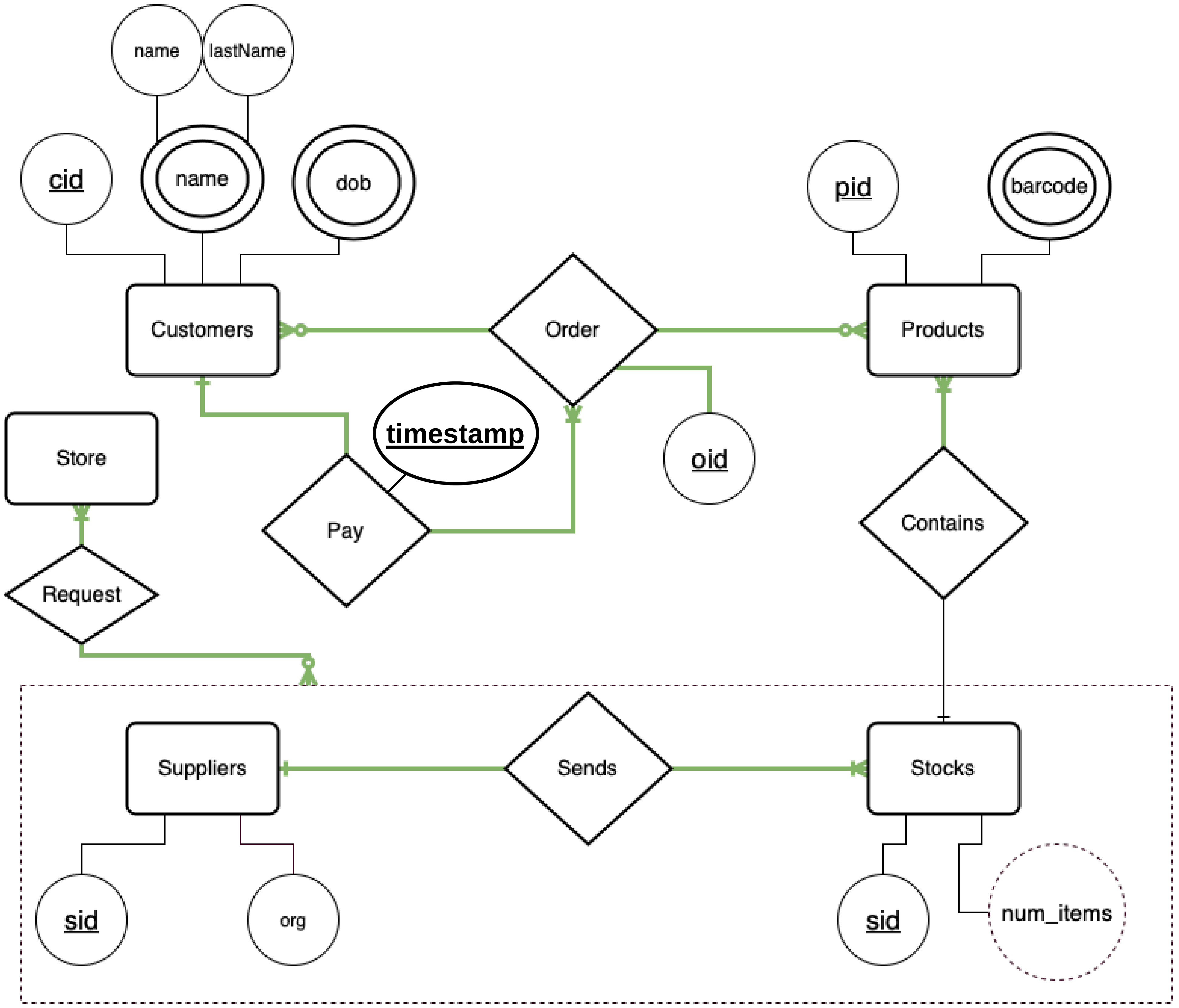
FAILS!!! Requests are not connected to stocks



Testing the system "Supplier sends stocks by request"

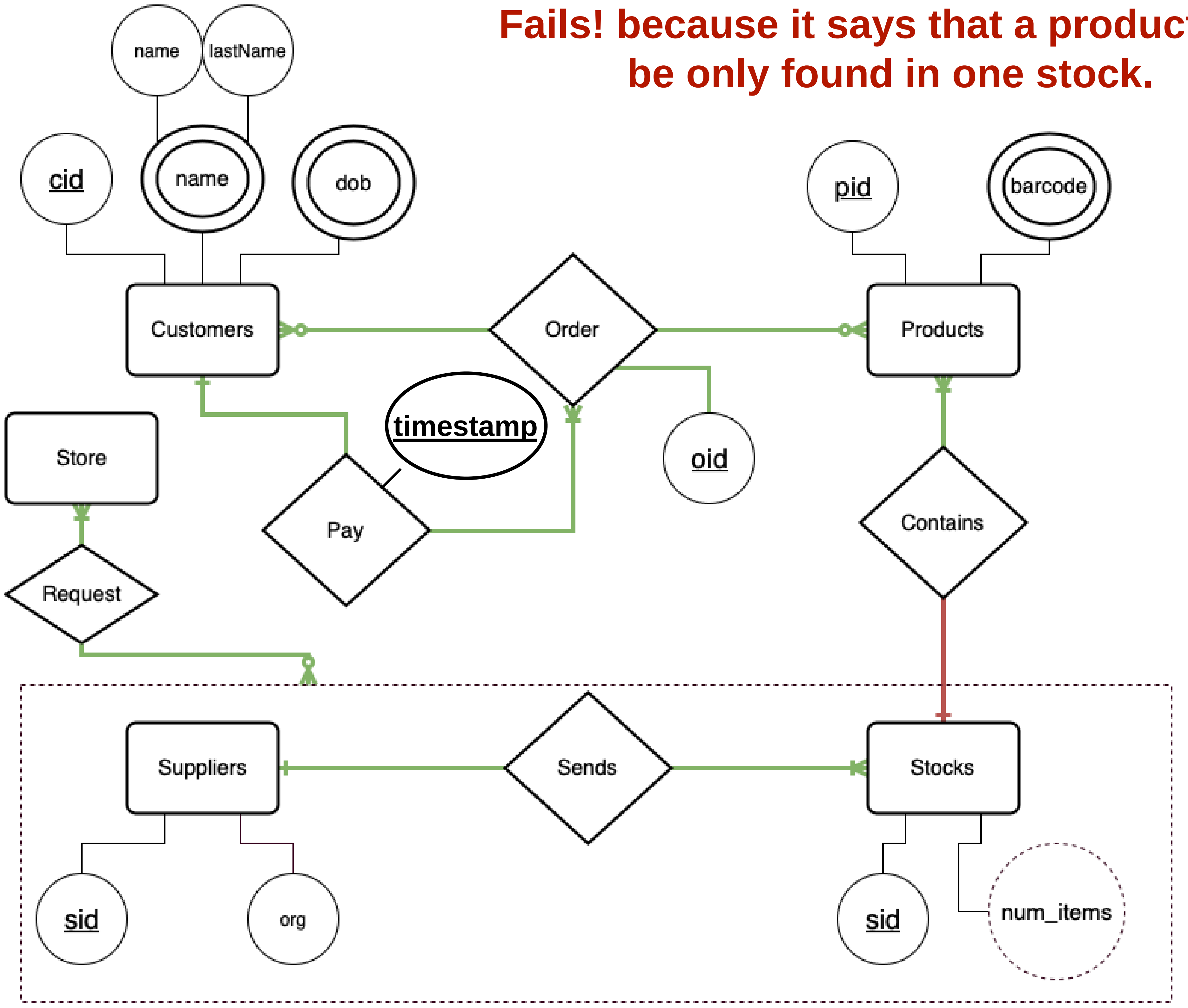


Testing the system
“Stocks must contain one or more products”

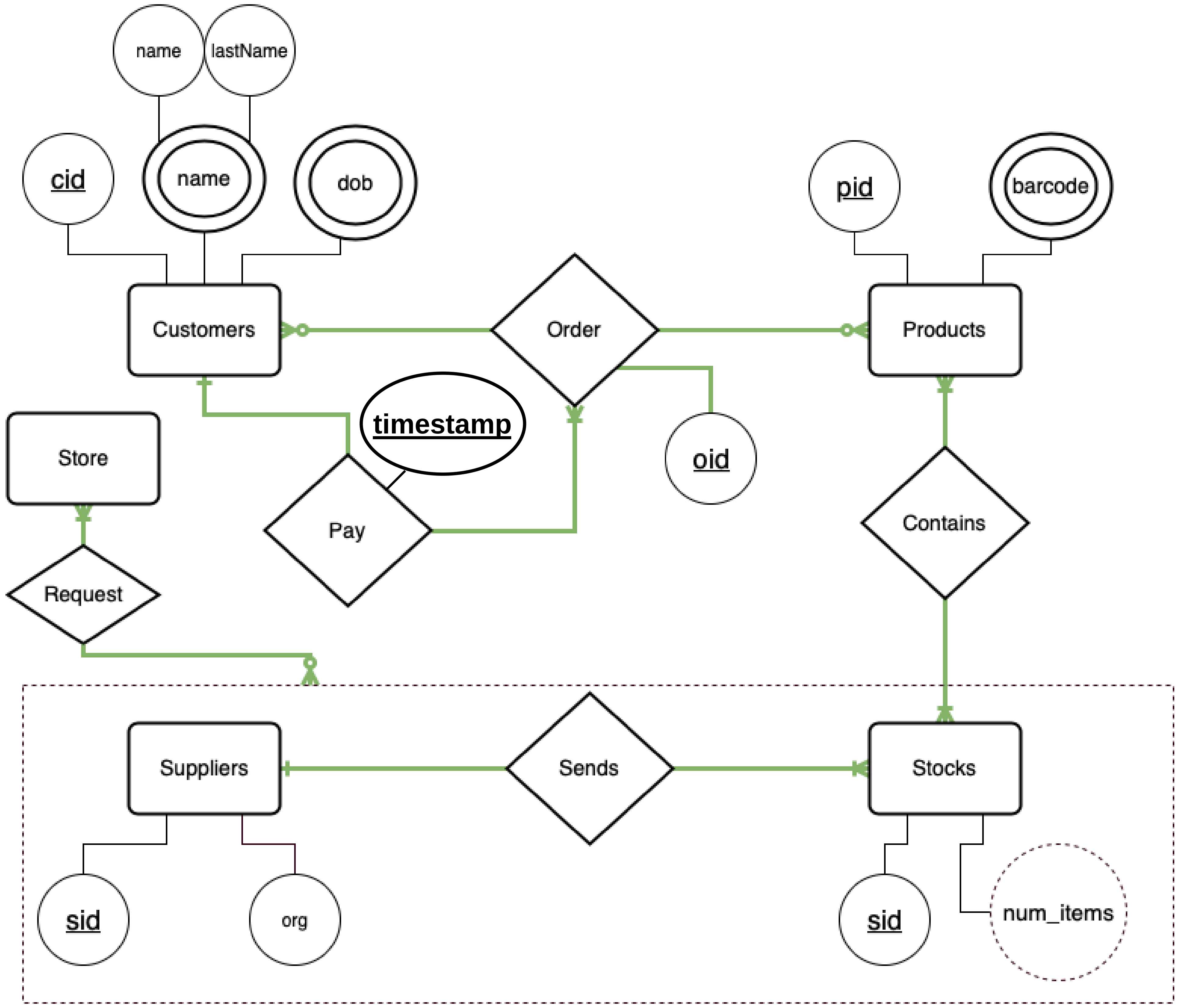


Testing the system
“Stocks must contain one or more products”

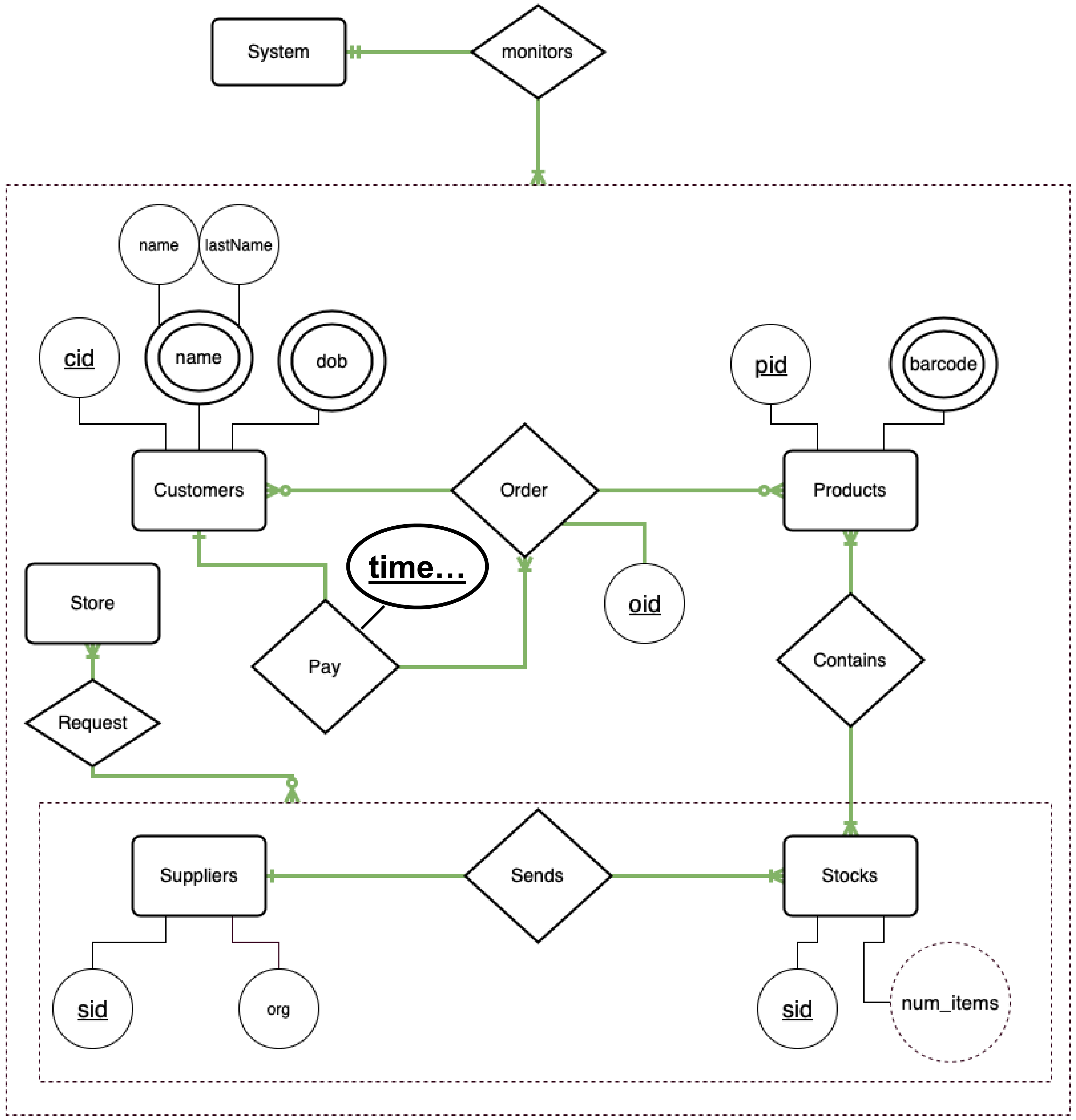
Fails! because it says that a product can be only found in one stock.



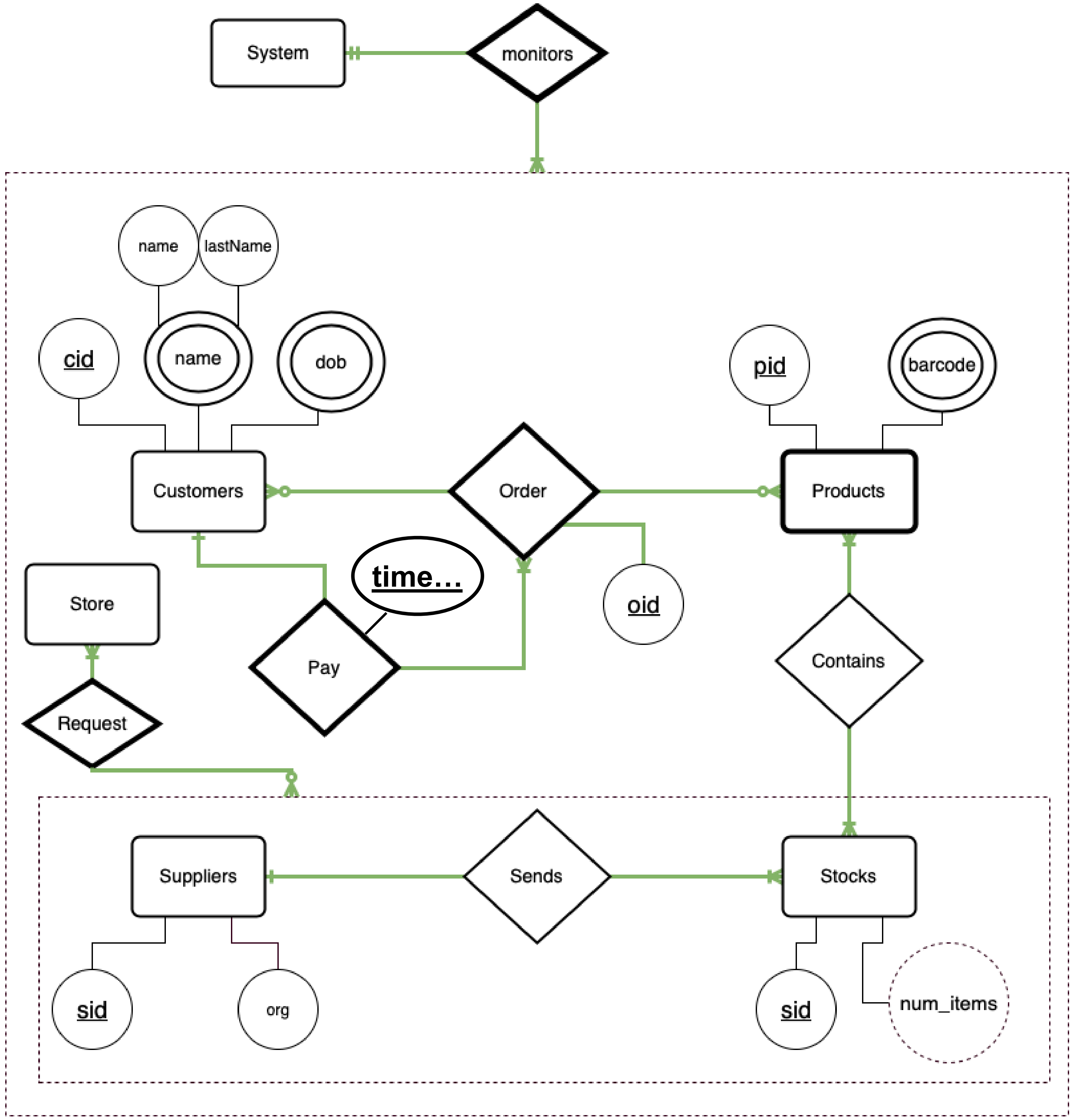
Testing the system
“Stocks must contain one or more products”



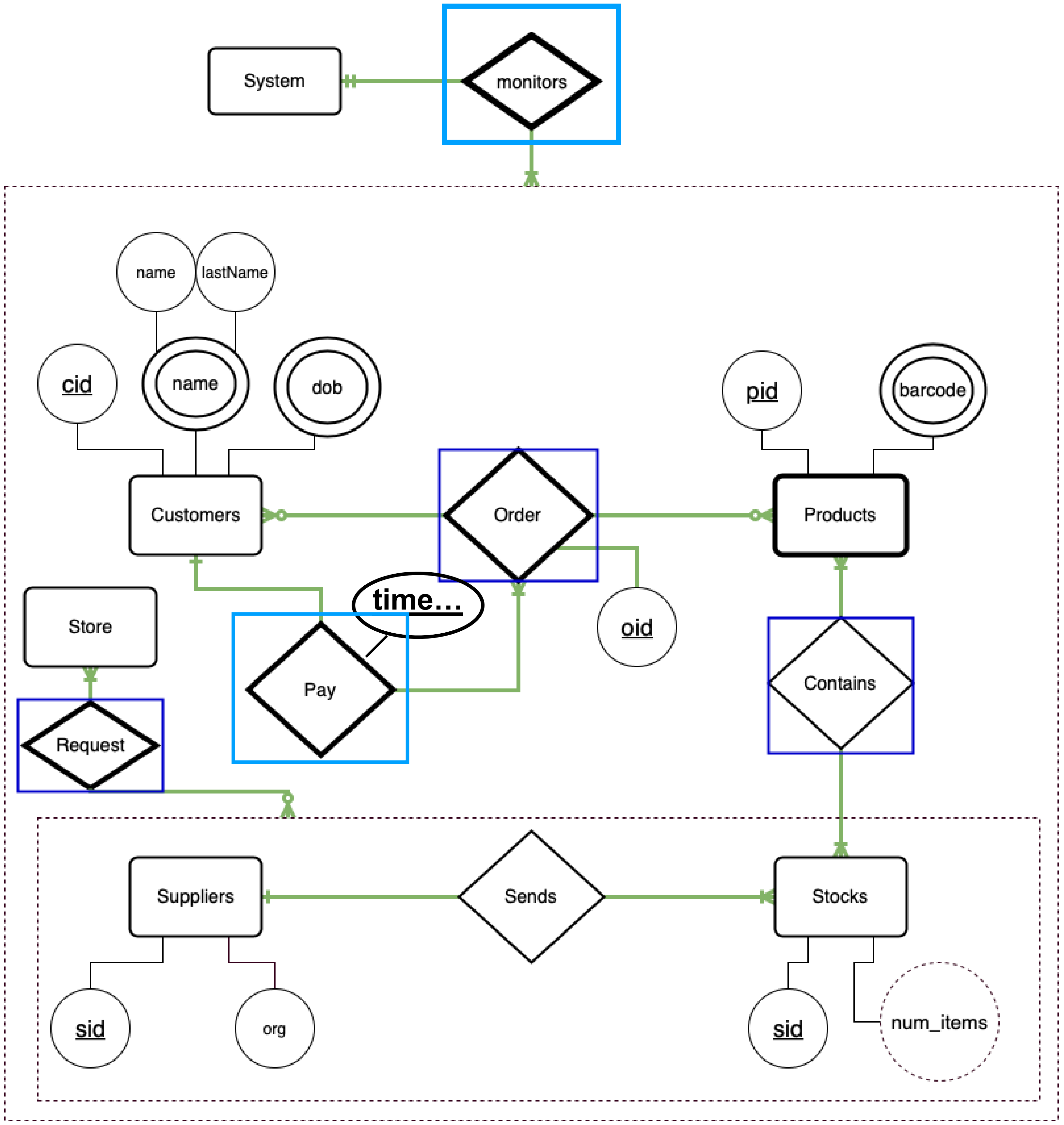
Testing the system
“Orders, payments and stocks can be monitored”


System
monitors
name
lastName
cid
name
dob
pid
barcode
Customers
Order
Products
time...
oid
Store
Pay
Contains
Request
Suppliers
Sends
Stocks
sid
org
sid
num_items

Testing the system “participation constraints”


System
monitors
name
lastName
cid
name
dob
pid
barcode
Customers
Order
Products
time…
oid
Store
Pay
Contains
Request
Suppliers
Sends
Stocks
sid
org
sid
num_items

Relationships that become tables/entities


System
monitors
name
lastName
cid
name
dob
pid
barcode
Customers
Order
Products
time…
Store
Pay
oid
Contains
Request
Suppliers
Sends
Stocks
sid
org
sid
num_items

Non-Functional Specs

- Performance
- Scalability
- Maintainability
- Memory usage
- Storage capacity
- DBMS