Introduction to Database Systems

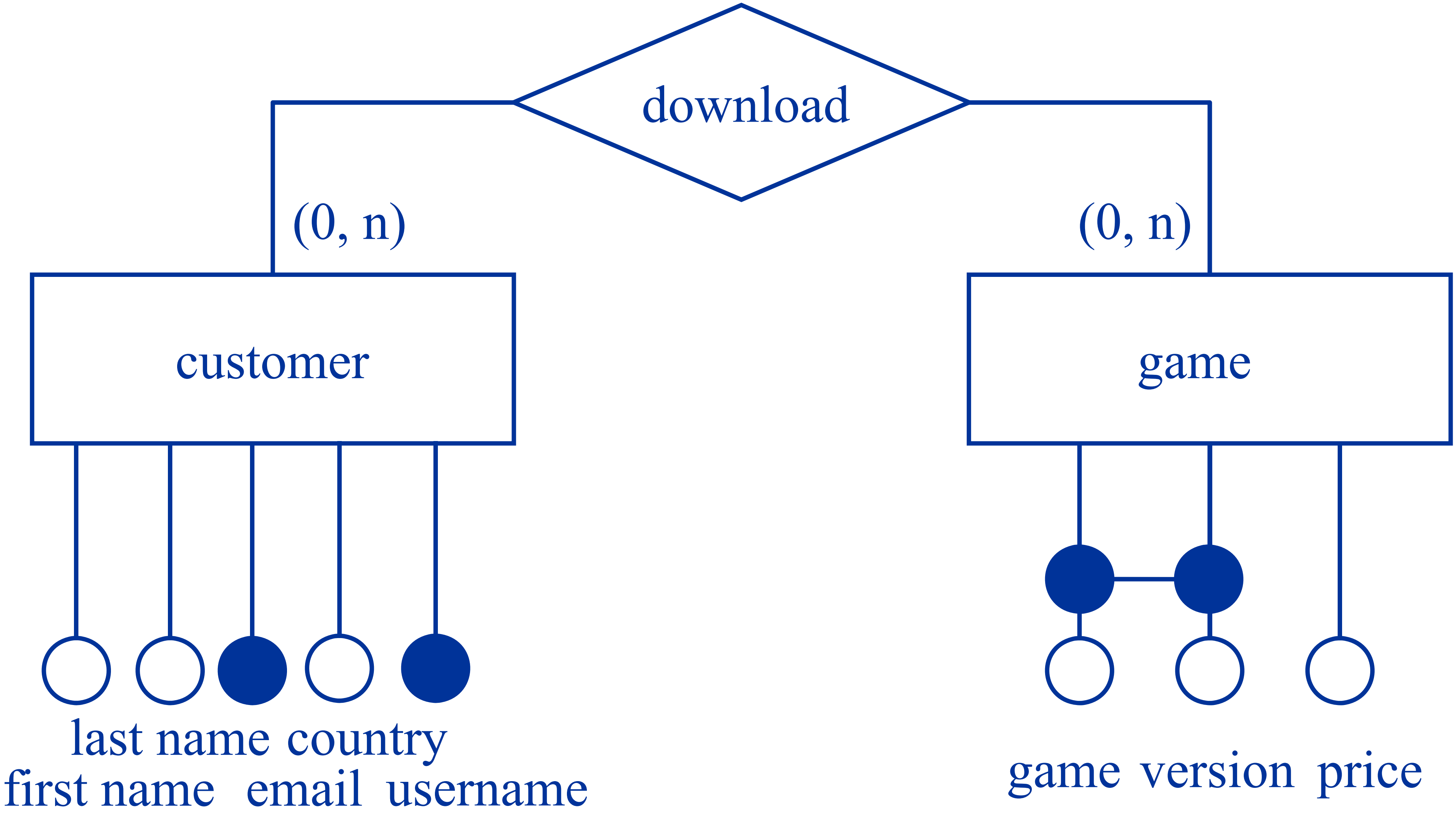
Conceptual Modelling with the Entity-relationship Model and Diagrams

Stéphane Bressan

NUS
National University of Singapore

First Case

We want to develop a sales analysis application for an online gaming store. We want to store several pieces of information about the customers: their first name, last name, email, the country of their registration to the online sales service and the username that they chose. We also want to manage the list of products, games, that are offered in the online store. We want to record the names, versions and prices of the games. The price is set for each version of each game. Finally, our customers download games. We need to remember which version of which game each customer downloaded. We consider that a customer downloads a given version of a game at most once. We do not need to remember the download date.

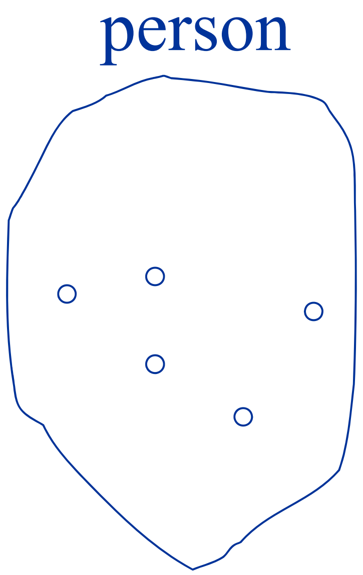

download
(0, n)
(0, n)
customer
game
last name
country
first name
email
username
game
version
price

Diagrams:

Entities, Relationships and Attributes

Entities and Entity Sets

Entities are identifiable "things".
The named box represents a set
of entities or "entity set".

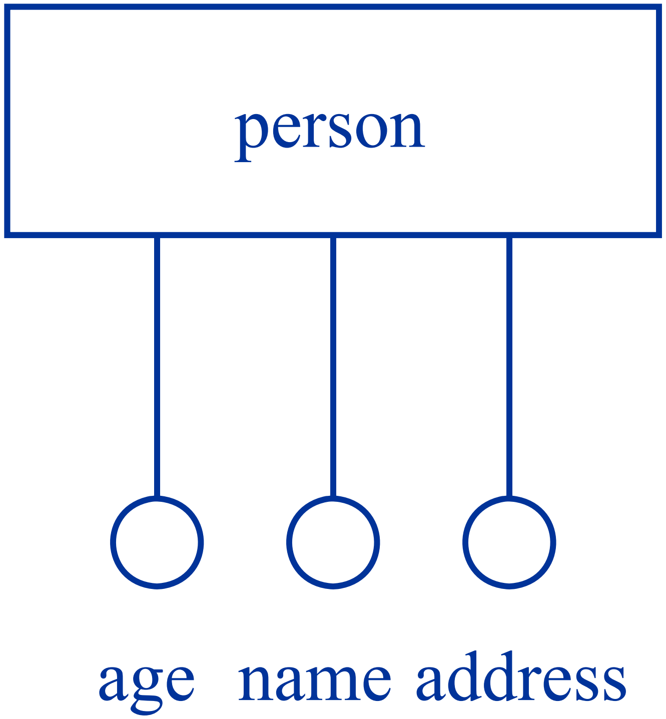


Attributes,Values and Value Sets

The E-R model is value-oriented. Values can be integer, strings, or atoms.

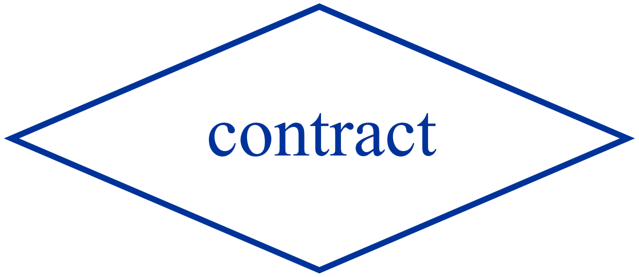
Attributes of Entities

Entities can have attributes. All entities in one entity set have the same attributes. However the attributes take different values for each entities.



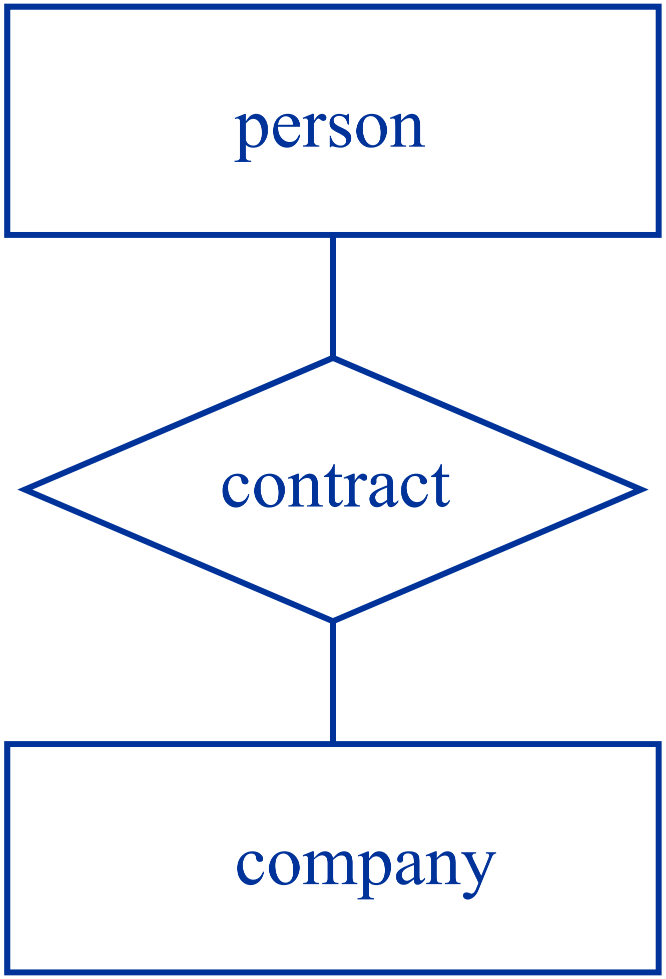
Relationships and Relationship Sets

A lozenge represents a set of relationships or "relationship set".



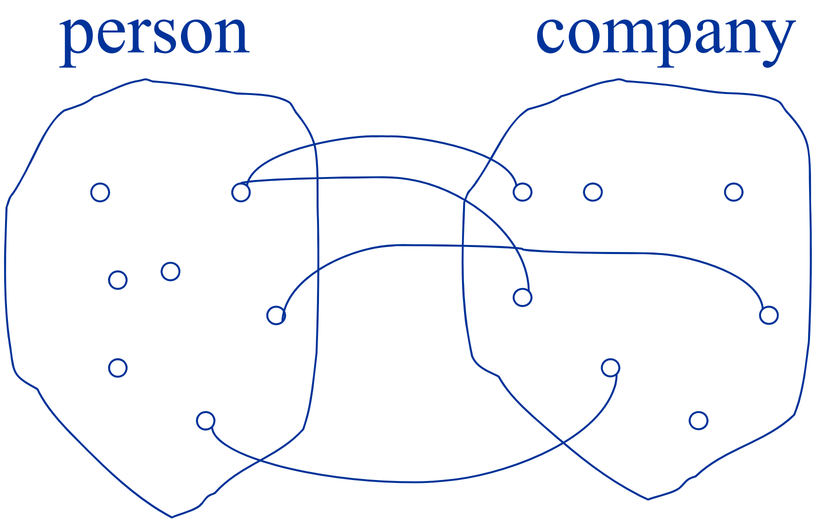
Relationships and Relationship Sets

A relationship associates zero or more entities (most commonly two). A "relationship set" is a set of relationships associating entities from the same entity sets.



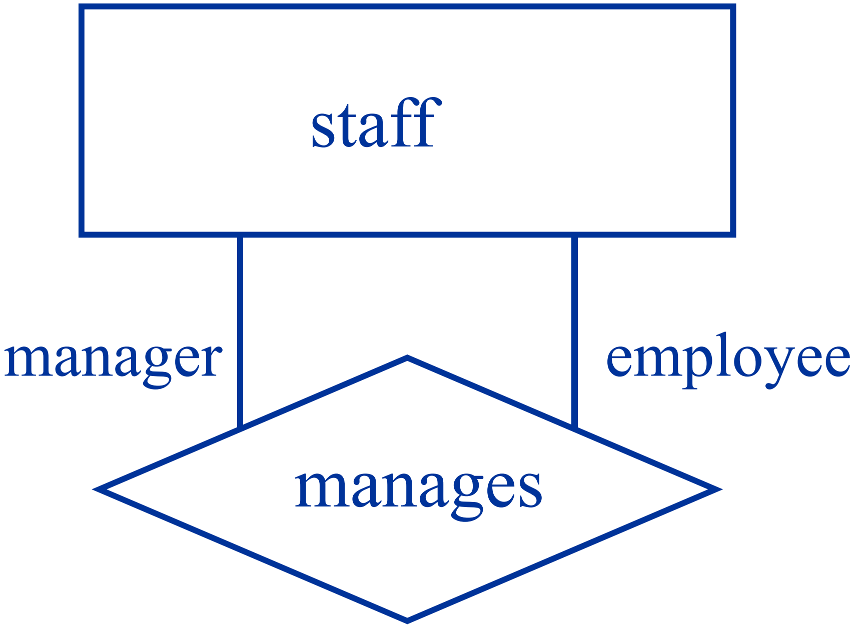
Relationships and Relationship Sets

A relationship associates zero or more entities (most commonly two). A "relationship set" is a set of relationships associating entities from the same entity sets.



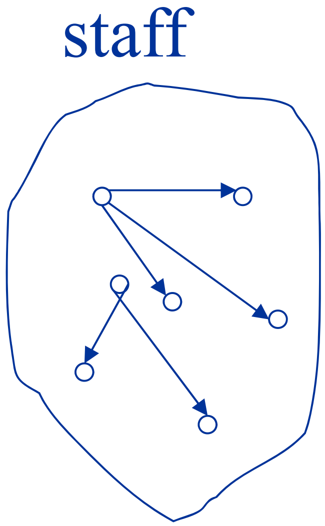
Relationships and Relationship Sets

Relationships can associate entities from the same entity set. In this case and in general, participation, or role, in the relationship can be named.



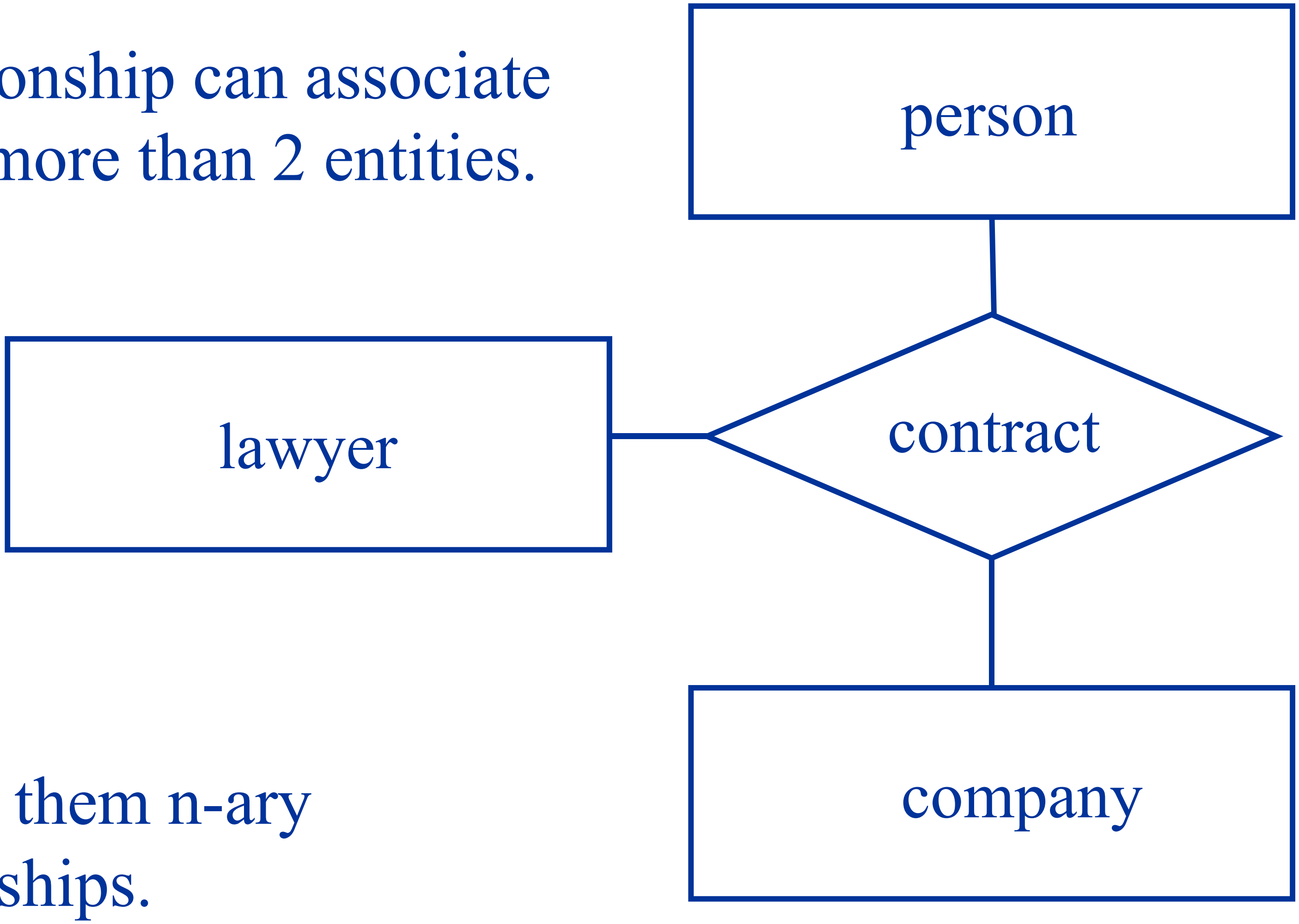
Relationships and Relationship Sets

Relationships can associate entities from the same entity set. In this case and in general, participation, or role, in the relationship can be named.



Relationships and Relationship Sets

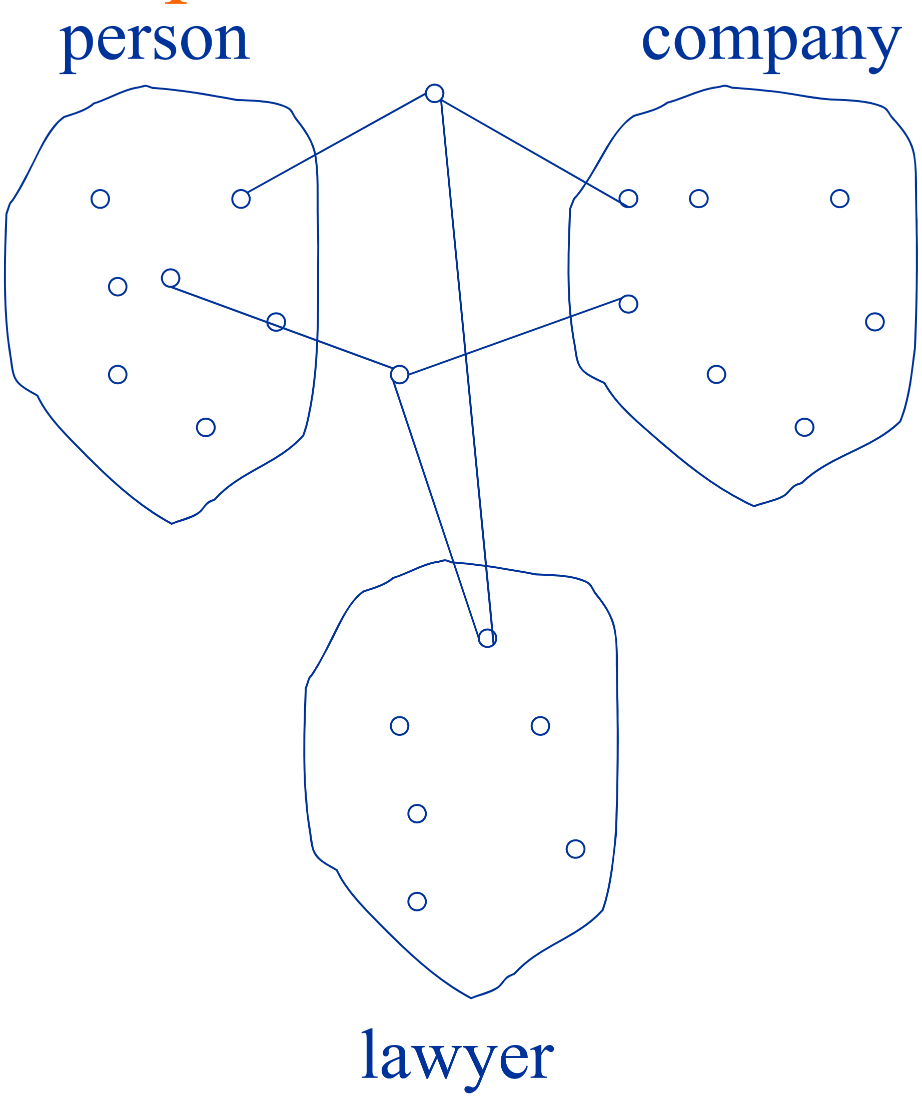
A relationship can associate less or more than 2 entities.



We call them n-ary relationships.

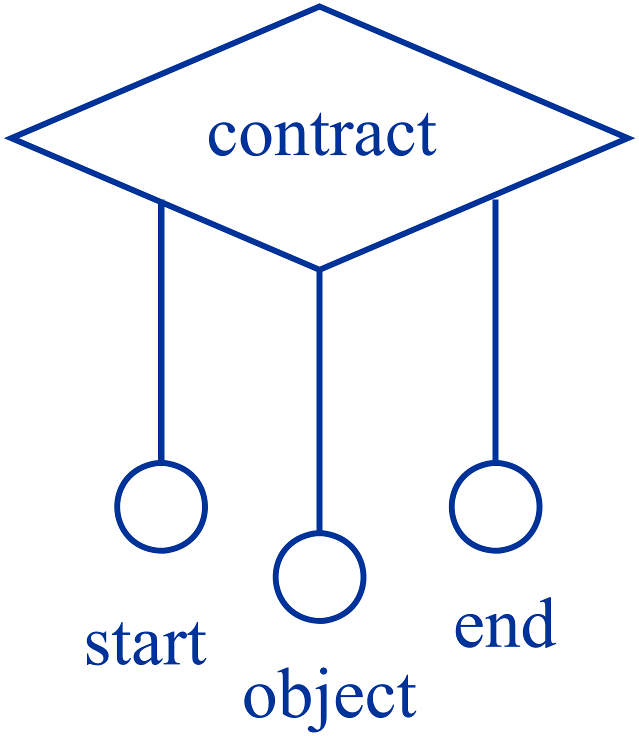
Relationships and Relationship Sets

Relationships can associate entities from the same entity set. In this case and in general, participation, or role, in the relationship can be named.



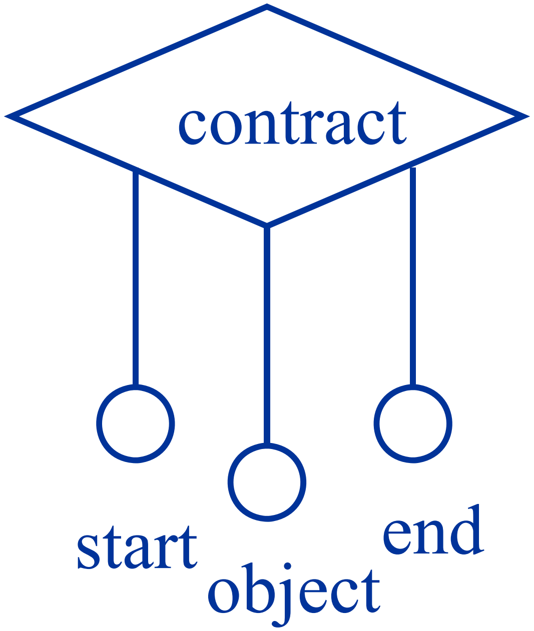
Attributes of Relationships

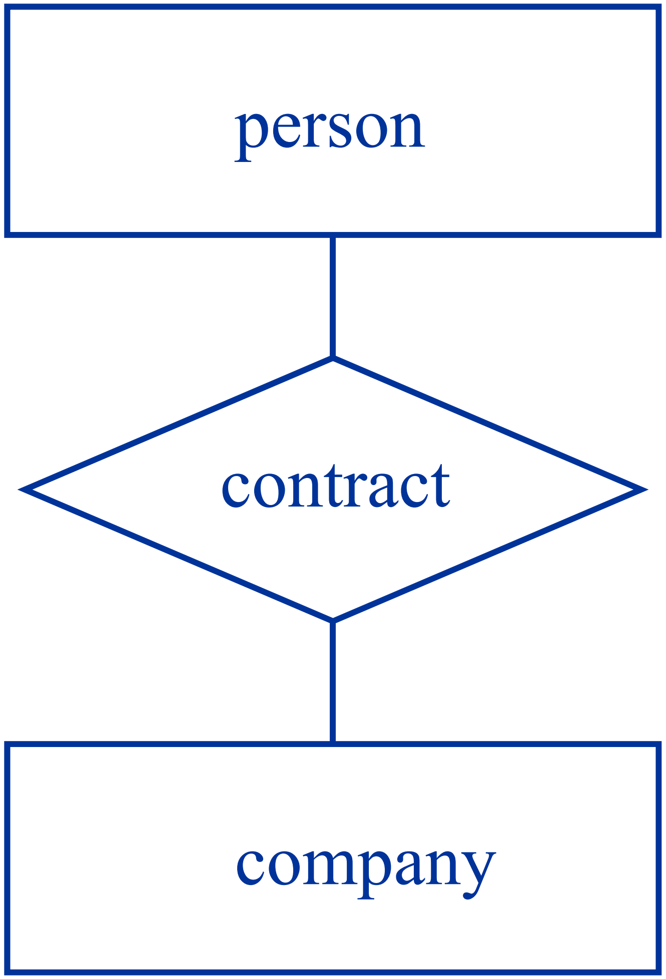
Relationship can have attributes. All relationships in one relationship set have the same attributes.



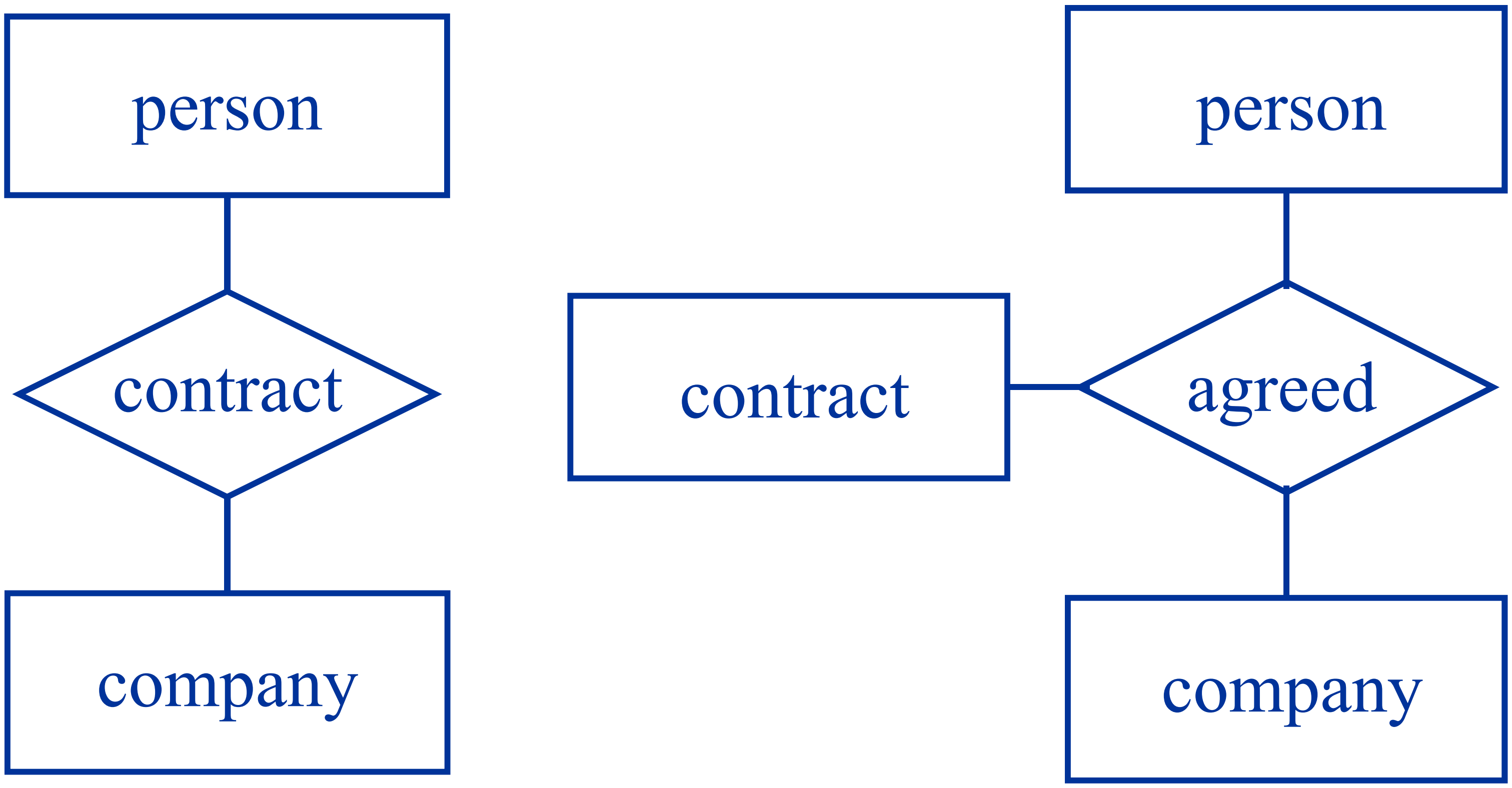
Attributes of Relationships

Relationships are distinguished not by their attributes but by their participating entities.





Entity or Relationship?



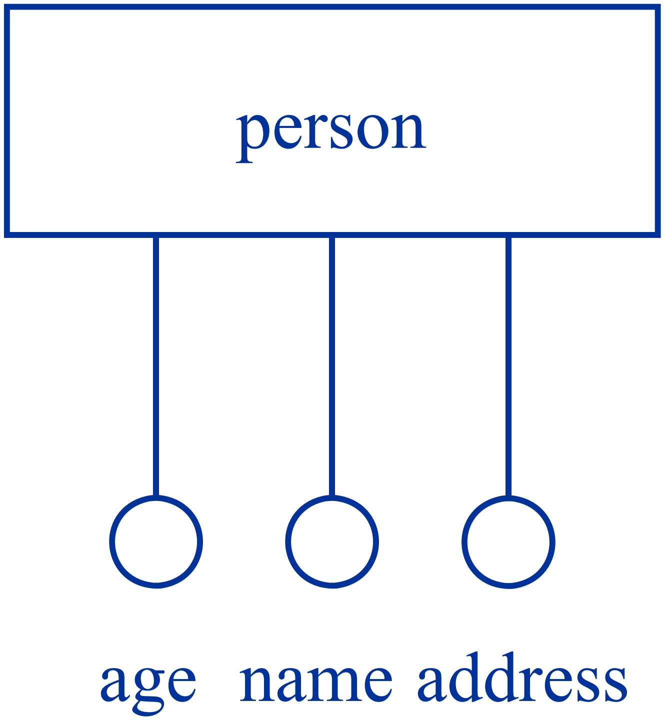
Integrity:

Keys and Participation Constraints

Entities' Identity

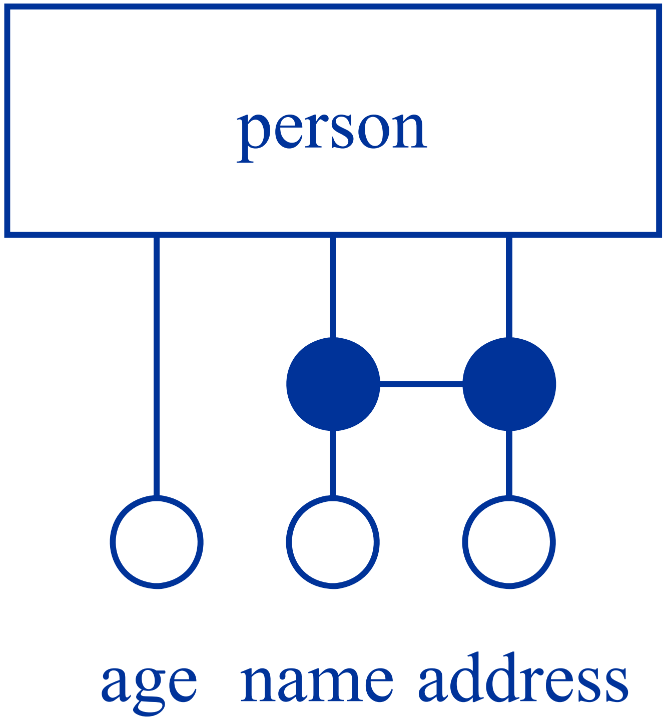
One attribute can identify the entity. This is a property of all entities in an entity set

Notice: at least all attributes identify the entity.



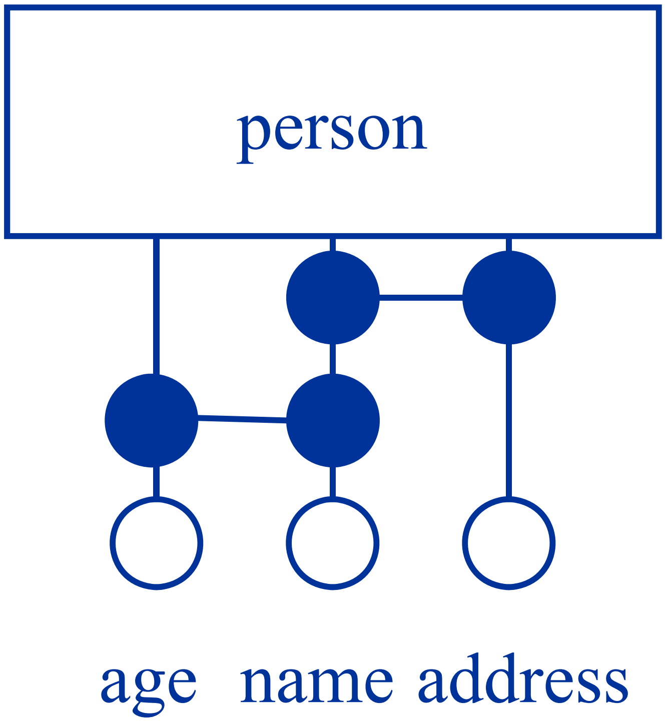
Entities' Identity

A combination of attributes can identify the entity.



Entities' Identity

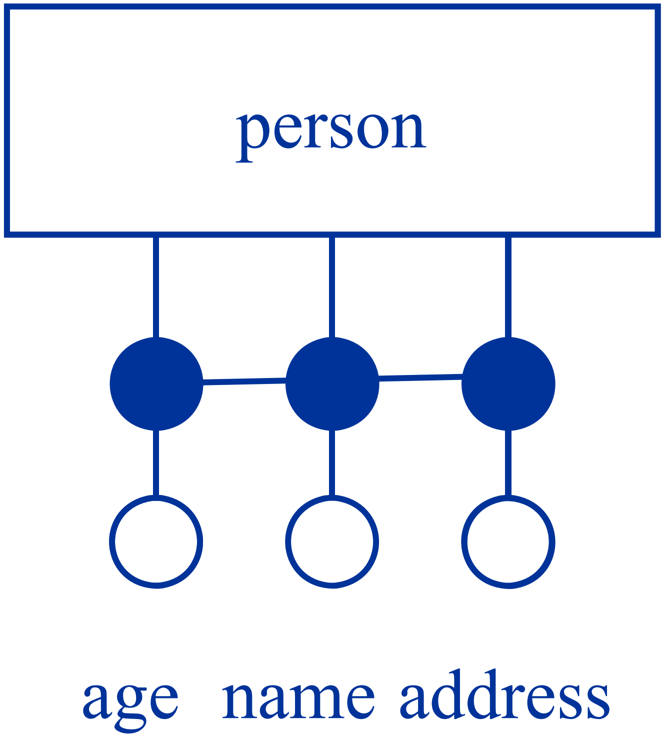
There might be several possible combination of attributes to identify an entity.



Entities' Identity

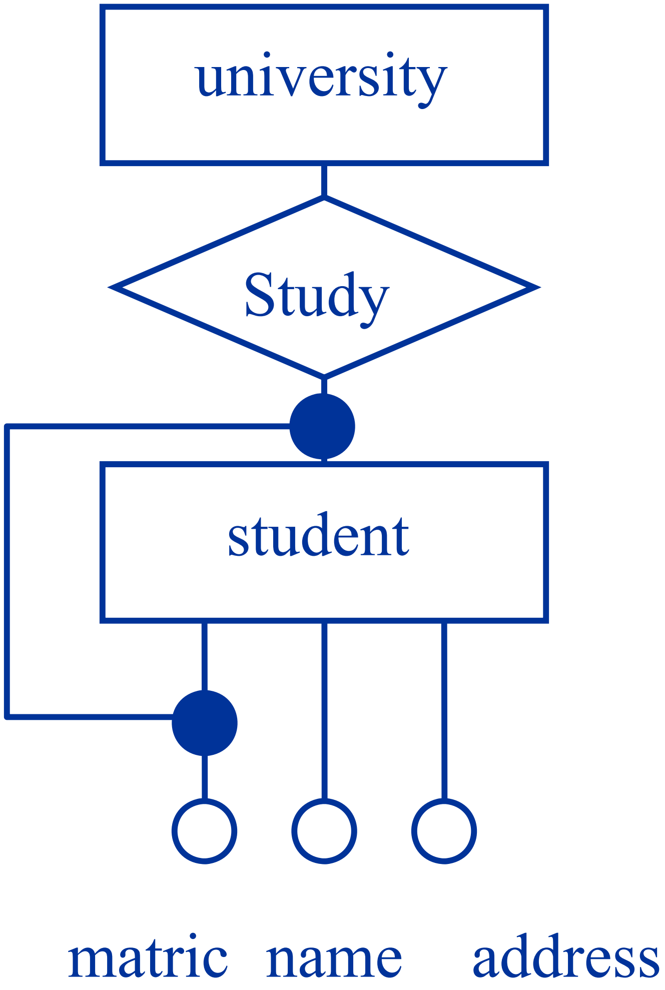
Notice: at least all attributes identify the entity

But we might prefer a minimum set of attributes.



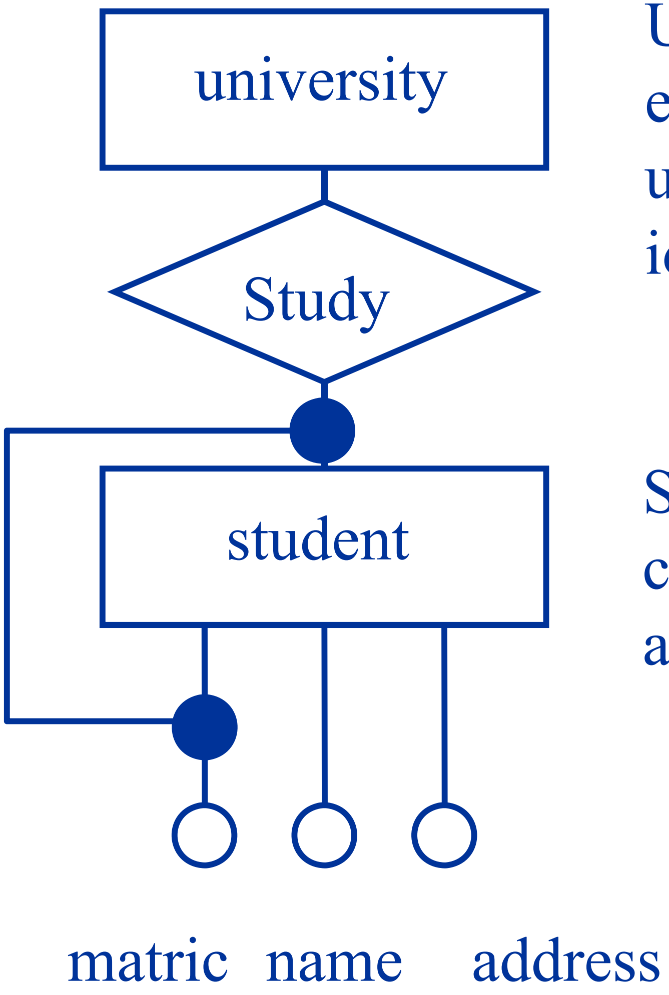
Weak Entities

Some entities can only be identify within the scope of a relationship with another entity set. Notice that the relationship must exist and be unique for each entity in the set.



Weak Entities

Matric numbers are given by the universities. The same number can be used by different universities.



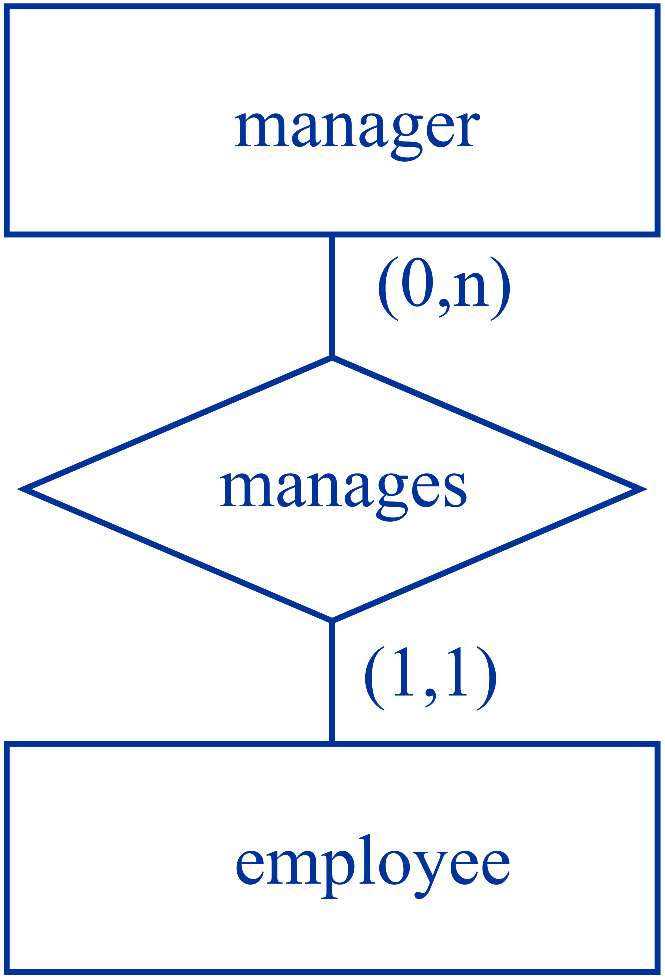
University is a dominant entity. We need to know the university in order to identify the student.

Student is a weak entity. It can be identified by its attributes alone.

Participation Constraints

The participation in a relationship can be constrained by a minimum and maximum: (1,1), (0, n), (2, 5).

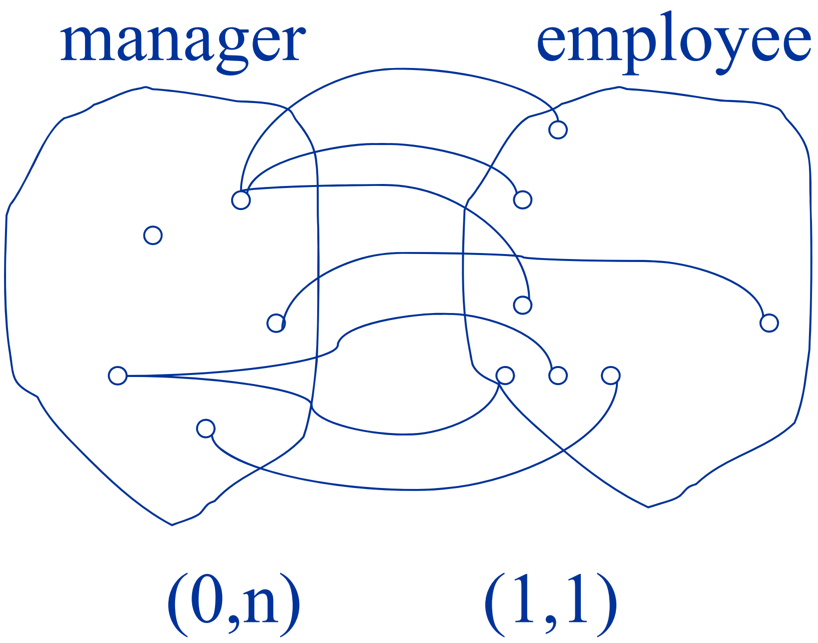
An employee is managed by at least and at most one manager (exactly one manager).



Participation Constraints

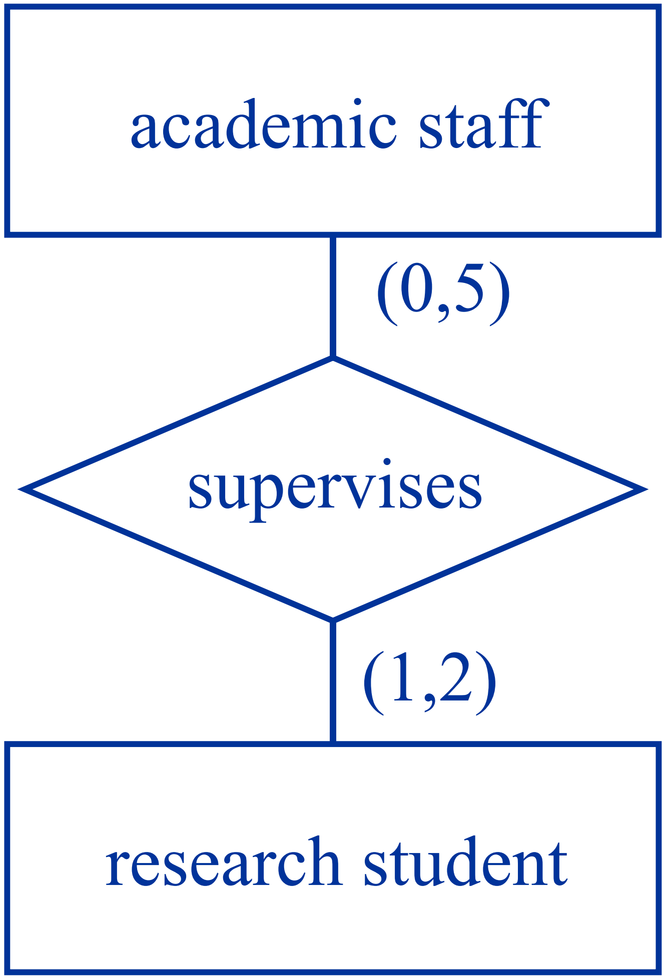
The participation in a relationship can be constrained by a minimum and maximum: (1,1), (0, n), (2, 5).

These two numbers are the minimum and maximum number of outgoing lines, respectively.



Participation Constraints

Another example: academic staffs can supervise up to 5 research students. Some staffs do not supervise students. Research students can have one or two supervisors.



Participation and Cardinality

(1, x) mandatory participation.

(0, x) optional participation.

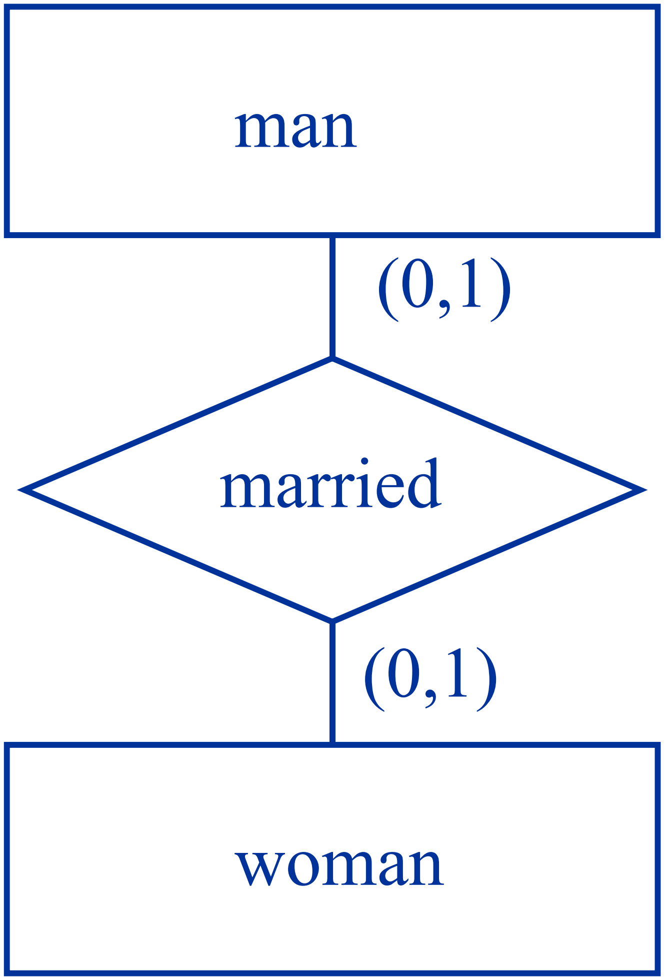
(x, 1) for all entities involved characterizes a one-to-one relationship.

(x, 1) for one entity involved and (x, N) or (x, y) y > 1 for the others characterizes a one-to-many relationship.

(x, N) or (x, y) y > 1 for all entities involved characterizes a many-to-many relationship.

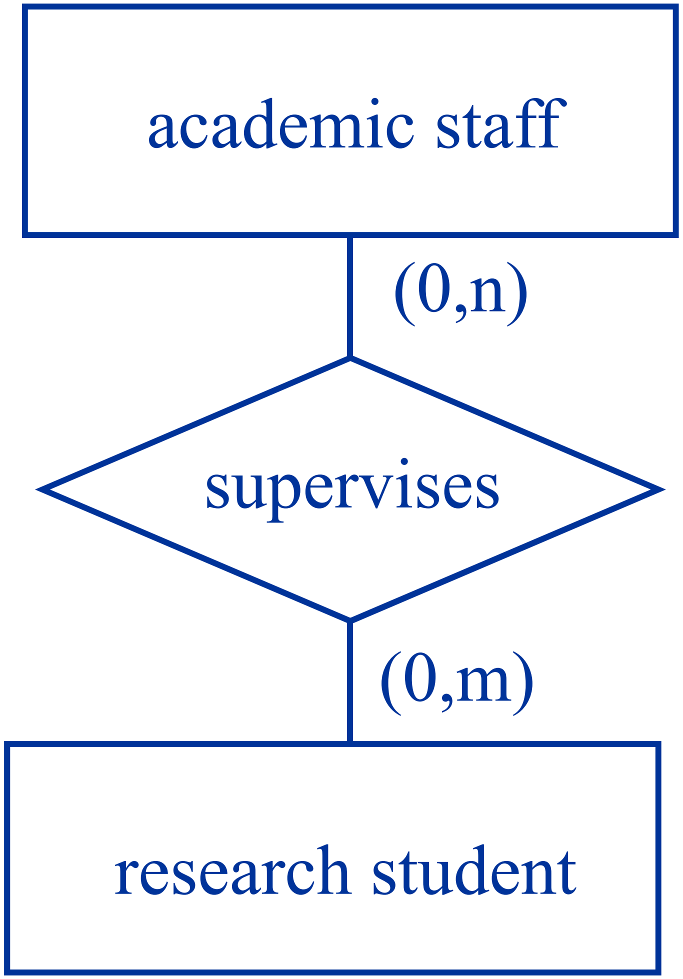
Participation and Cardinality

Example of a one-to-one relationship.



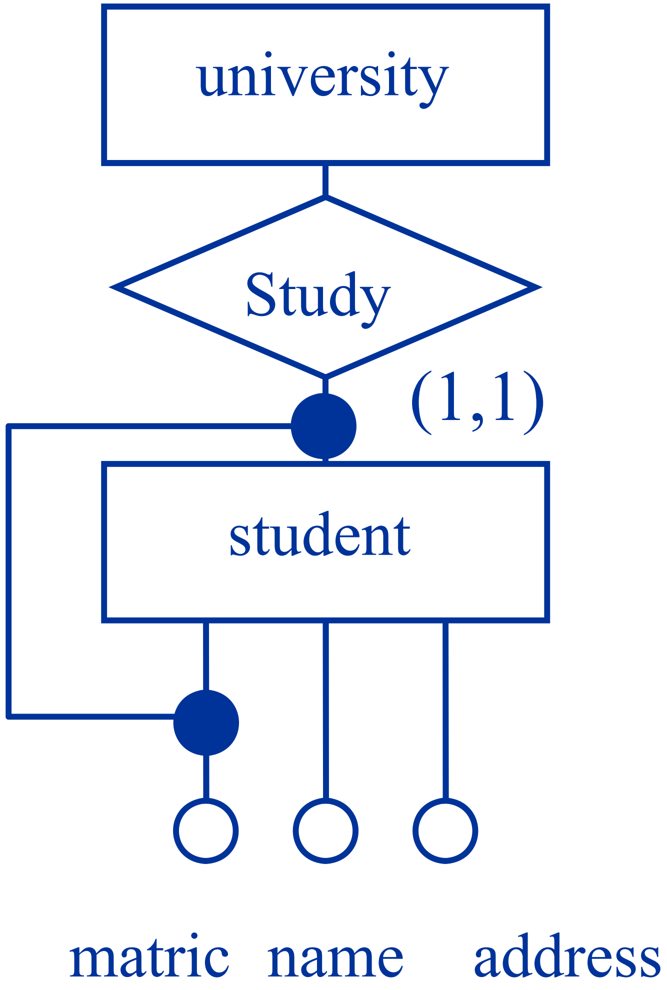
Participation and Cardinality

By default we have many-to-many relationships.



Weak Entities

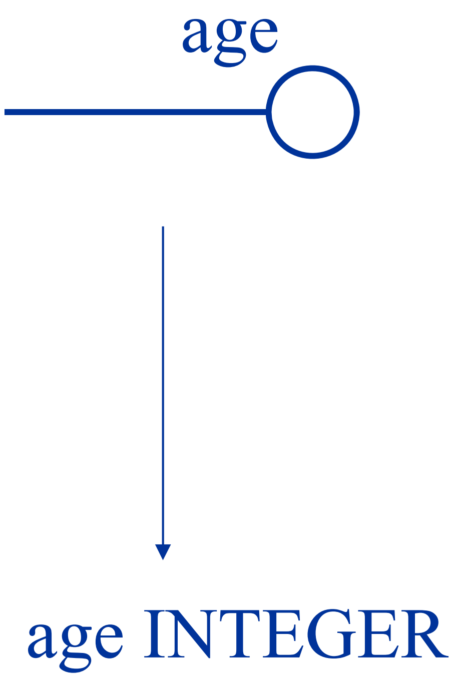
Weak entities can only be define for a participation constrained by (1,1) cardinalities. Also called mandatory one-to-many relationships.



Translation to Relational:
3 Rules and 3 Exceptions

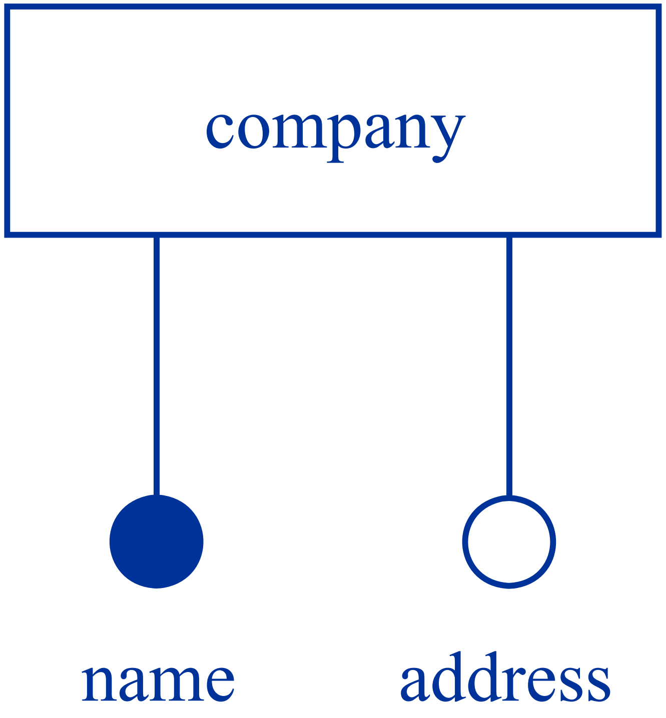
Rule 1: Value Sets

An attribute is mapped to one (or more) attribute of a relation with a domain. This is a first step towards the logical design.

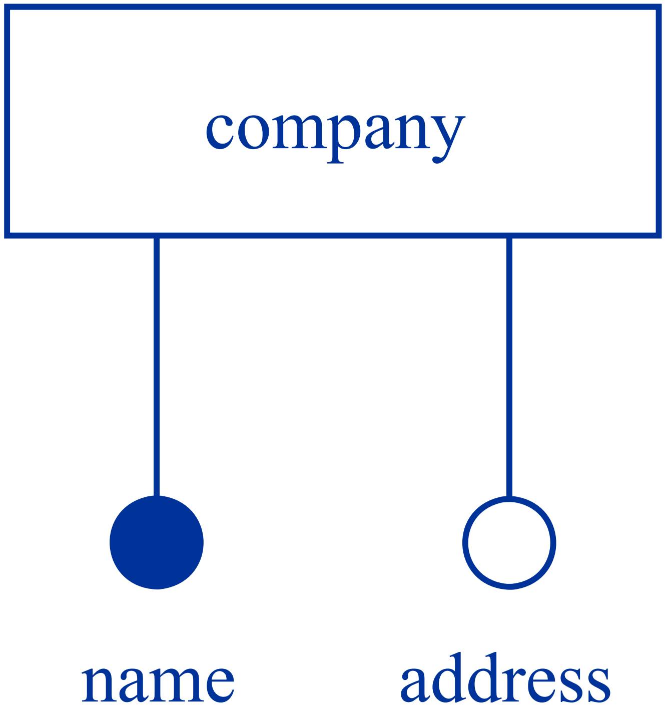


Rule 2: Entity Sets

An entity set is mapped to a relation. The attributes of the entity set are mapped to attributes of the relation. The keys are mapped to the primary key of the relation and to unique combinations of attributes , if there is more than one key.



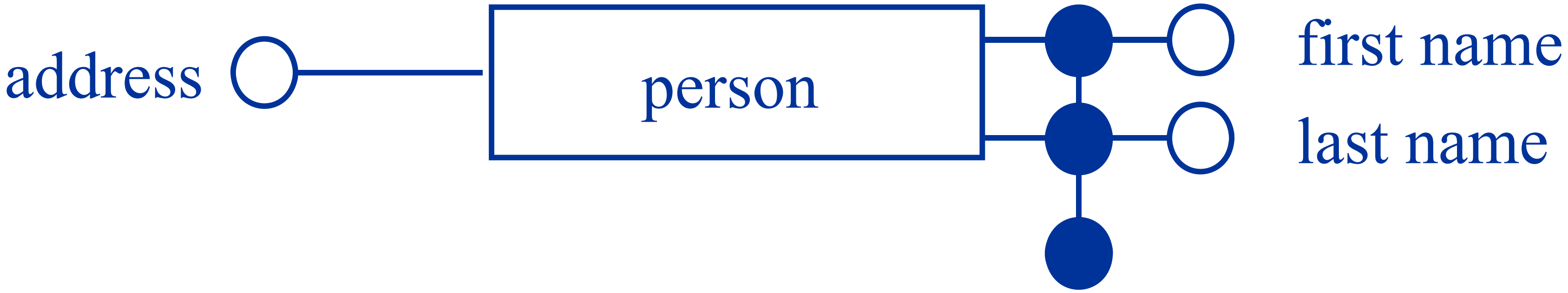
Rule 2: Entity Sets



Rule 2: Entity Sets

```
CREATE TABLE company(
name VARCHAR(64) PRIMARY KEY,
address VARCHAR(128),
)
```

Rule 2: Entity Sets



Rule 2: Entity Sets

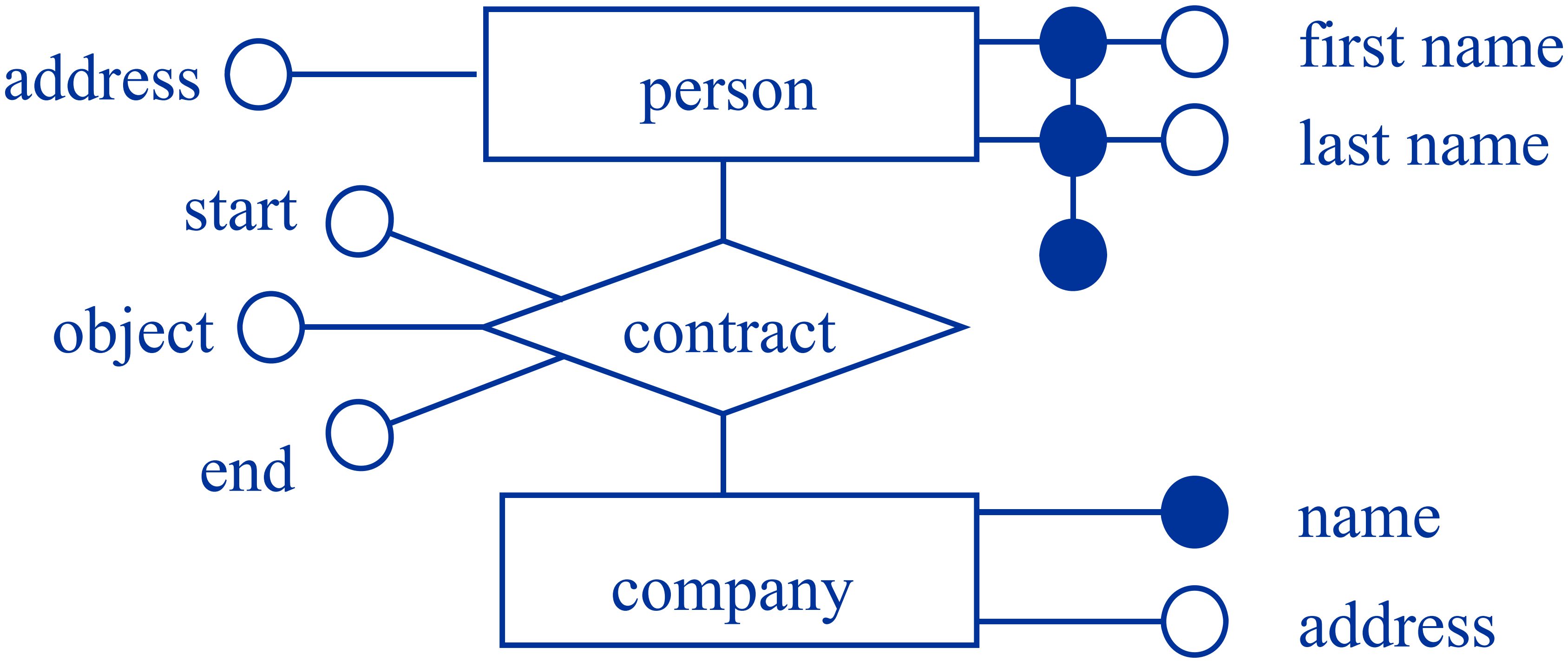
```
CREATE TABLE person (
first name VARCHAR(32),
last_name VARCHAR(32),
address VARCHAR(128),
PRIMARY KEY (first_name, last_name))
```

Rule 3: Relationship Sets

A relationship sets is mapped to a relation. The attributes of the relation consist of the attributes of the relationship set as well as of the keys of the participating entities.

contract

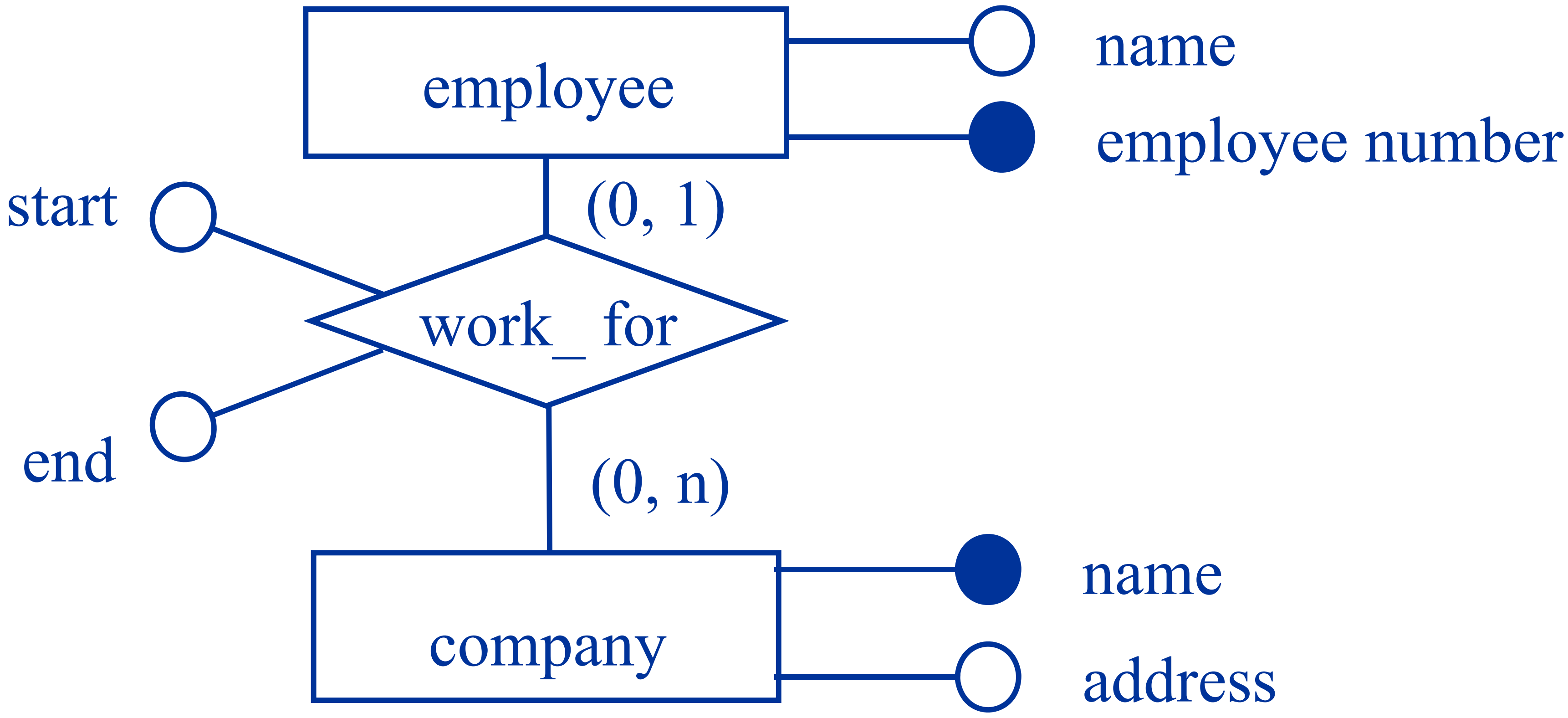
Rule 3: Relationship Sets



Rule 3: Relationship Sets

```
CREATE TABLE contract(
start DATE,
end DATE,
object VARCHAR(128),
pfirst_name VARCHAR(32),
plast_name VARCHAR(32),
cname VARCHAR(64),
PRIMARY KEY (pfirst_name, plast_name, cname),
FOREIGN KEY (pfirst_name , plast_name )
REFERENCES person(first_name, last_name),
FOREIGN KEY (cname ) REFERENCES company(name))
```

Exception 1: One-to-many Relationships



A one-to-many relationship indicates a key constraint

Exception 1: One-to-many Relationships

If we use the standard mapping, the primary key of the relationship table is inadequate to capture the fact that an employee works for at most one company.

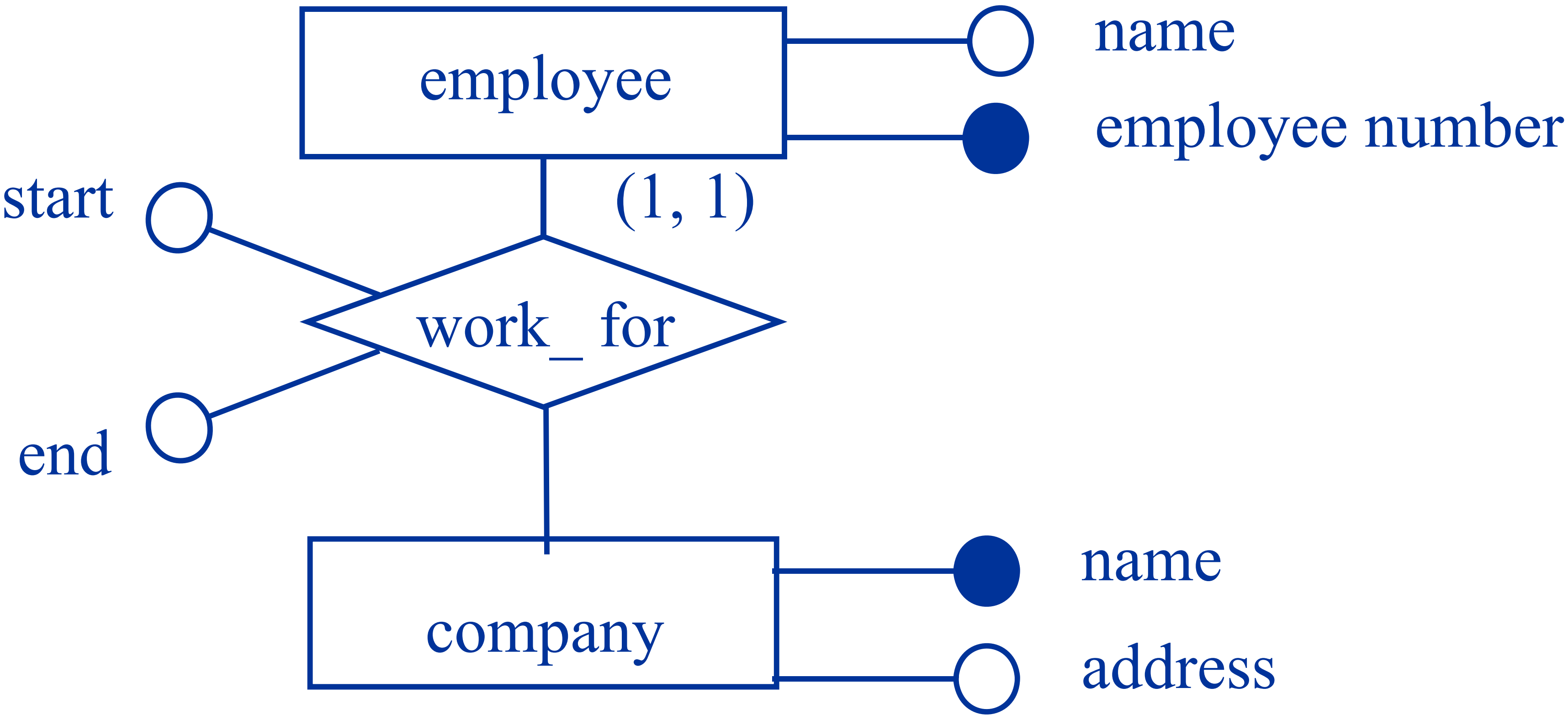
```
CREATE TABLE work_for (
start DATE,
end DATE,
enumber CHAR(8),
cname VARCHAR(32),
PRIMARY KEY (enumber, cname),
FOREIGN KEY (enumber) REFERENCES
employee(number),
FOREIGN KEY (cname) REFERENCES company(name))
```

Exception 1: One-to-many Relationships

Instead, we change the primary key of the relationship table or add UNIQUE constraints. Now an employee works for at most one company.

```
CREATE TABLE work_for(
start DATE,
end DATE,
enumber CHAR(8) PRIMARY KEY,
cname VARCHAR(32),FOREIGN KEY (enumber)
REFERENCES employee(number),
FOREIGN KEY (cname) REFERENCES company(name))
```

Exception 2: (1, 1) Participation Constraints

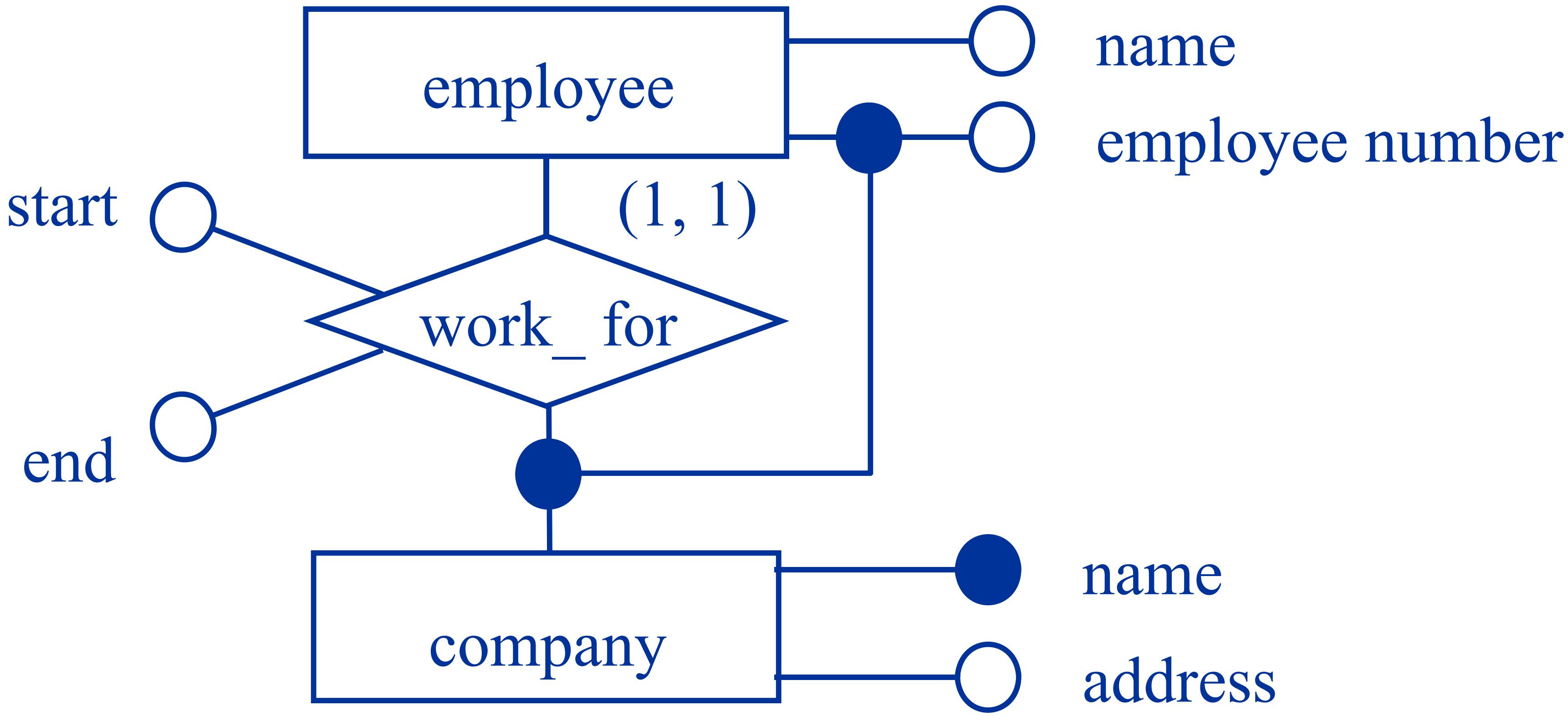


Exception 2: (1, 1) Participation Constraints

We merge the table employee and the table work_for and use the primary key of the employee table. Now an employee works for exactly one company.

```
CREATE TABLE employee_work_for (
start DATE,
end DATE,
enumber CHAR(8) PRIMARY KEY,
ename CHAR(32),
cname VARCHAR(32),
FOREIGN KEY (cname) REFERENCES company(name))
```

Exception 3: Weak Entity



Exception 3: Weak Entity

We cannot use the mapping of “exception 2” although The primary key of the employee table is not enumber because it is a weak entity.

```
CREATE TABLE employee_work_for(
start DATE,
end DATE,
enumber CHAR(8) PRIMARY KEY,
ename CHAR(32),
cname VARCHAR(32),
FOREIGN KEY (cname) REFERENCES company(name))
```

Exception 3: Weak Entity

We merge the table employee and the table work_for and use the primary key of the weak entity. The primary key is the composite key enumber and cname because it is a weak entity.

```
CREATE TABLE employee_work_for(
start DATE,
end DATE,
enumber CHAR(8),
ename CHAR(32),
cname VARCHAR(32),
PRIMARY KEY (enumber, cname),
FOREIGN KEY (cname) REFERENCES company(name))
```

Credits

The content of this lecture is based on chapter 7 of the book “Introduction to database Systems”
By
S. Bressan and B. Catania,
McGraw Hill publisher

Images and clips used in this presentation are licensed from Microsoft Office Online Clipart and Media

For questions about the content of this course and about copyrights, please contact Stéphane Bressan

steph@nus.edu.sg