Introduction to Database Systems

Conceptual Modelling: The Entity-Relationship Model

Lee Mong Li

Recap: Relational Model

- Relational database is a set of relations
- Relation has a schema – name, attributes and data constraints
- Attribute has a domain
- Database schema is the schema of all the relations

```
CREATE TABLE book (
title VARCHAR(256),
authors VARCHAR(256),
publisher VARCHAR(64),
ISBN10 CHAR(10) NOT NULL UNIQUE,
ISBN13 CHAR(14) PRIMARY KEY );

CREATE TABLE student (
name VARCHAR(32) NOT NULL,
email VARCHAR(256) PRIMARY KEY,
year DATE,
faculty VARCHAR(62),
department VARCHAR(32),
graduate DATE,
CHECK(graduate >= year) );
```

Recap: Relational Model

```
CREATE TABLE copy (
owner VARCHAR(256) REFERENCES student(email) ON DELETE CASCADE,
bookID CHAR(14) REFERENCES book(ISBN13),
num INT CHECK(num>0),
available VARCHAR(6) CHECK(available = 'TRUE' OR available='FALSE'),
PRIMARY KEY (owner, bookID, num));

CREATE TABLE loan (
borrower VARCHAR(256) REFERENCES student(email),
owner VARCHAR(256),
bookID CHAR(14),
num INT,
borrowed DATE,
returned DATE,
FOREIGN KEY (owner, bookID, num) REFERENCES copy(owner, bookID, num) ON
DELETE CASCADE,
PRIMARY KEY (borrowed, borrower, owner, bookID, num),
CHECK(returned >= borrowed));
```

Database Design Process

1. **Requirements Analysis**
   - What does the user want from the database?
   - Find out the data/application/performance requirements
2. **Conceptual Database Design**
   - Capture data requirements using a conceptual data model, e.g. ER model
   - High level description of data to be stored in database, and constraints that hold over the data
3. **Logical Database Design**
   - Convert conceptual database design to a logical schema supported by DBMS
4. **Schema Refinement**
   - Use data constraints to improve the logical schema.
   - Theory of normalizing relations
5. **Physical Database Design**
   - Use performance criteria to design physical schema, e.g. build indexes
6. **Application and Security Design**
   - Specify access control policies

Example: Online Book Exchange Application

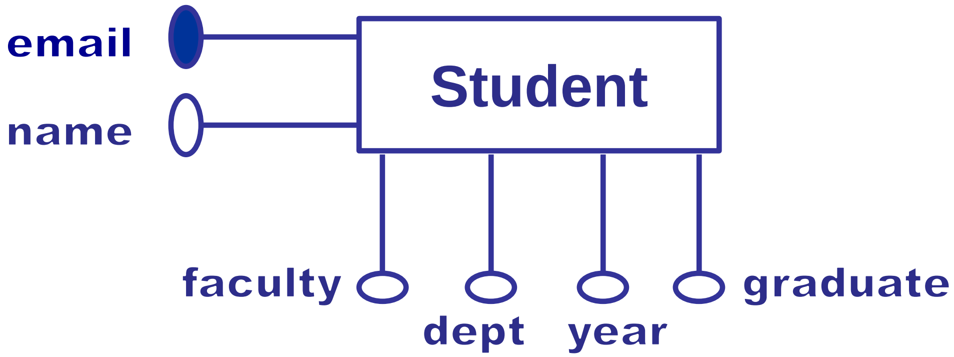
Students at the National University of Ngendipura (NUN) buy books for their studies. They also lend and borrow books from other students.

Design and implement an online book exchange system. This application needs to record information about students, books they own, and books they borrow from other students.

Example: Online Book Exchange Application

Requirements Analysis

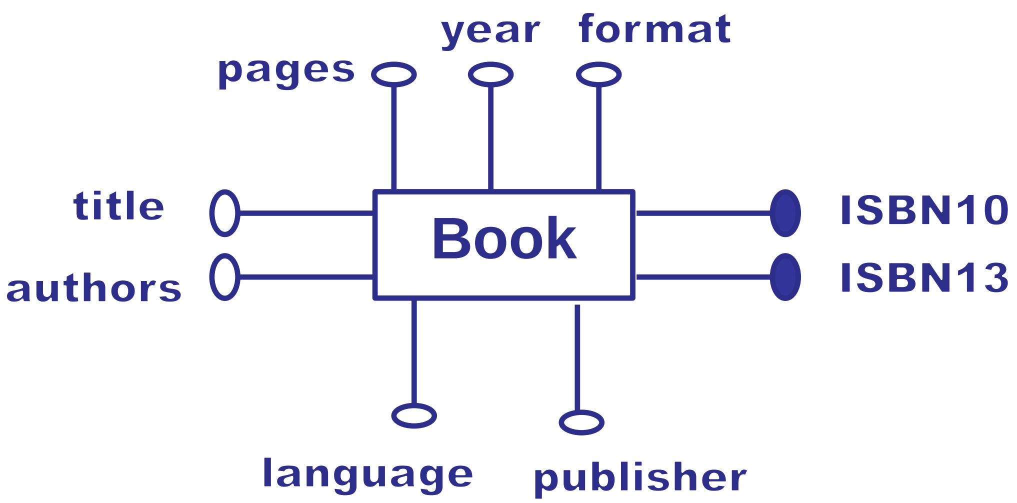
- Record the name, faculty, department and student number of each student
- Each student is identified in the system by email
- Database also records the date at which student joined and graduated/left the university

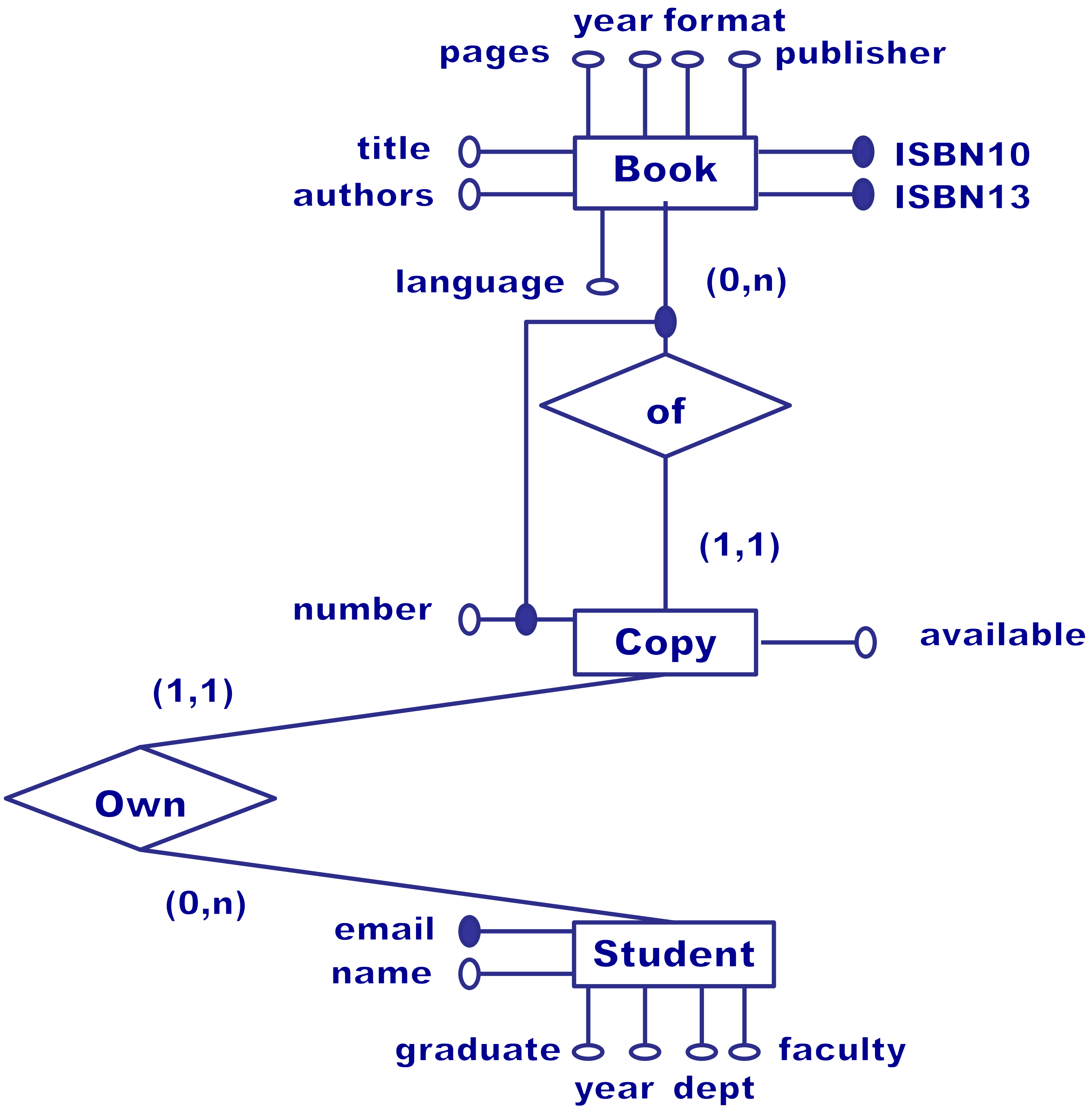


Example: Online Book Exchange Application

Requirements Analysis

- **For each book, the database records the title, format, pages, language, authors, publisher, year and ISBN-10 and ISBN-13.**
- **The International Standard Book Number, ISBN-10 or -13, is an industry standard for the unique identification of books.**



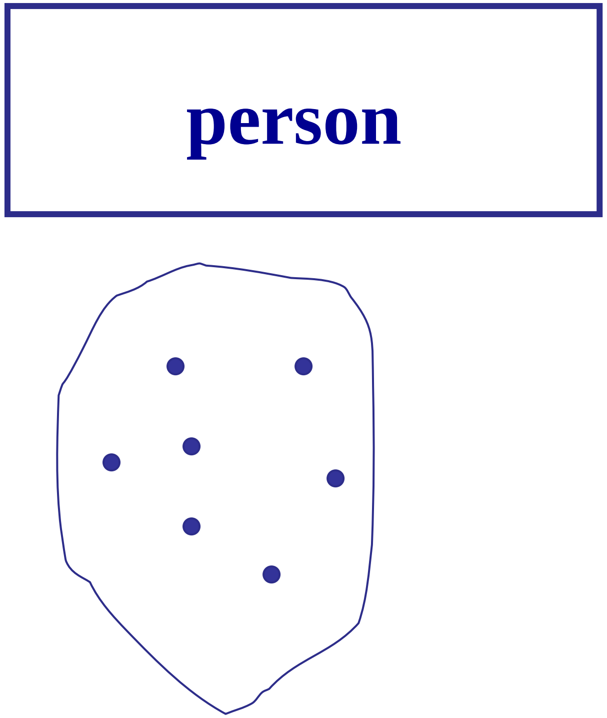

year format
pages
publisher
title
authors
Book
ISBN10
ISBN13
language
(0,n)
of
(1,1)
number
Copy
available
(1,1)
Own
(0,n)
email
name
Student
graduate
faculty
year
dept

Entity Relationship (ER) Model

- **The Entity-relationship model is a graphical model for the conceptual design of data centric applications.**
- **Data is described in terms of entities and their relationships**
- **Information about entities and relationships are described using attributes**
- **Certain data constraints are represented using additional annotations**

Entities and Entity Sets

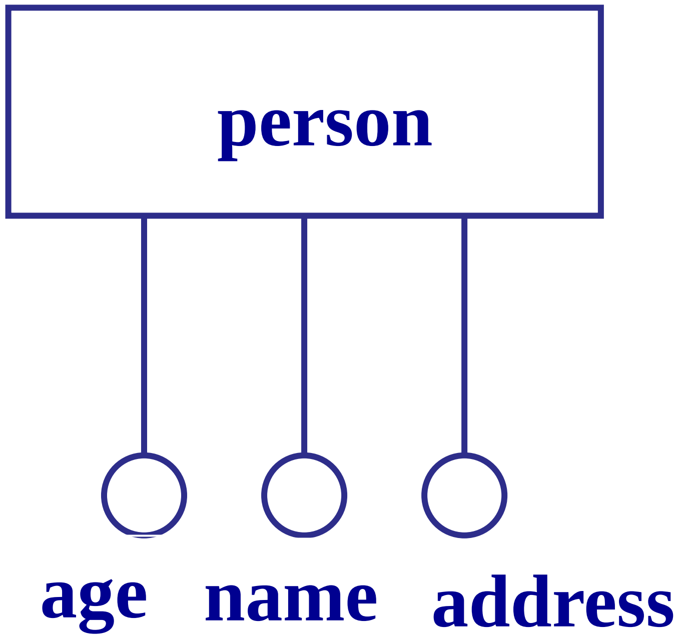
- Entities are identifiable “things”, or real world objects distinguishable from other objects
- The named box represents a set of entities or entity set.



person

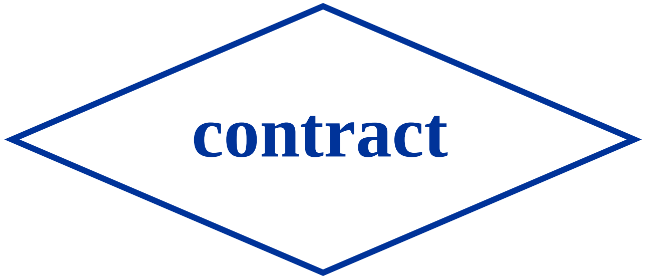
Attributes of Entities

- Entities can have attributes
- Each attribute has an atomic domain e.g. integer, string
- All entities in one entity set have the same attributes
- However, the attributes take different values for each entity



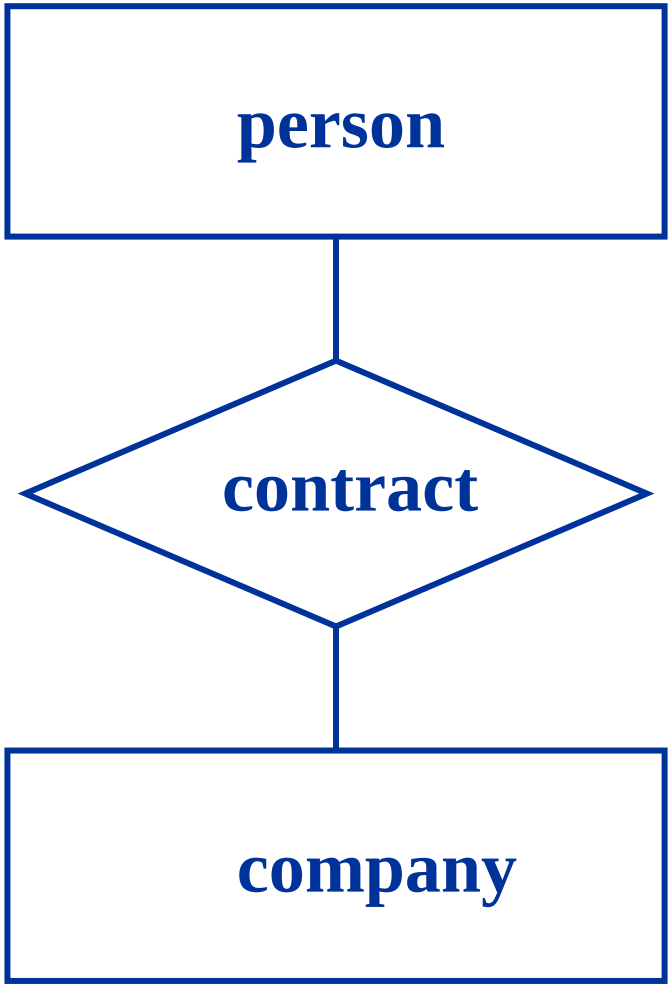
Relationships and Relationship Sets

- A named diamond represents a set of relationships or a relationship set.



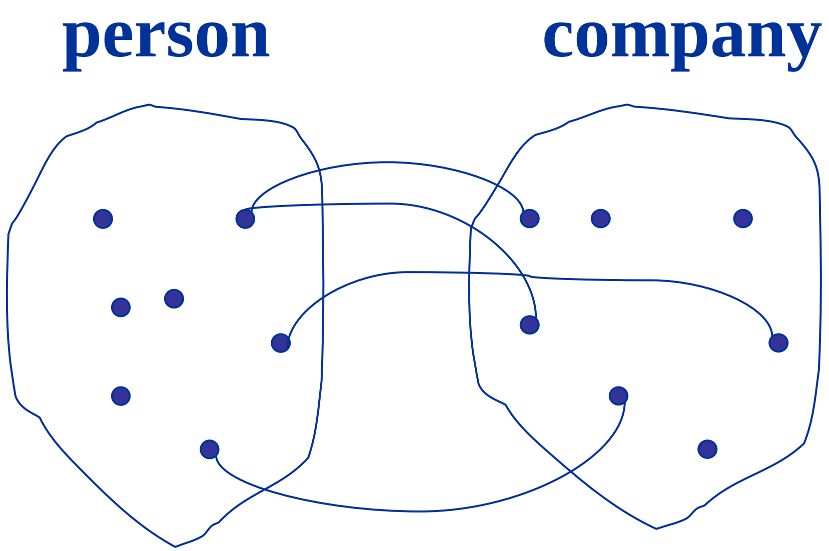
Relationships and Relationship Sets

- A relationship associates two or more entities.
- A relationship set is a set of relationships associating entities from the same entity sets.



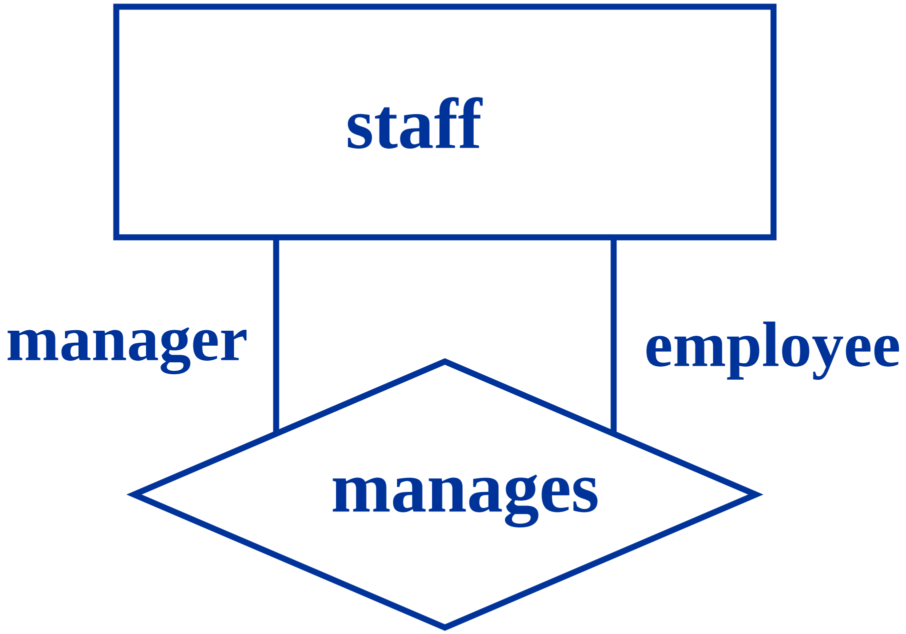
Relationships and Relationship Sets

- A relationship associates two or more entities.
- A relationship set is a set of relationships associating entities from the same entity sets.



Relationships and Relationship Sets

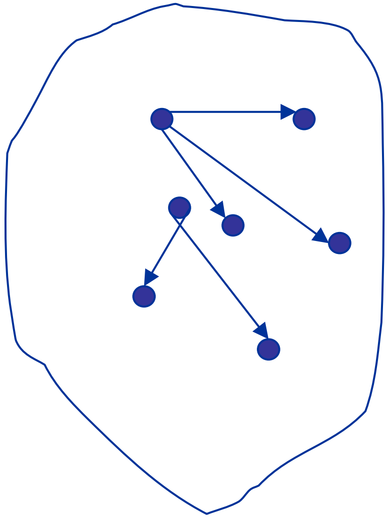
- Relationships can associate entities from the same entity set.
- In this case and in general, participation, or role, in the relationship can be named.



Relationships and Relationship Sets

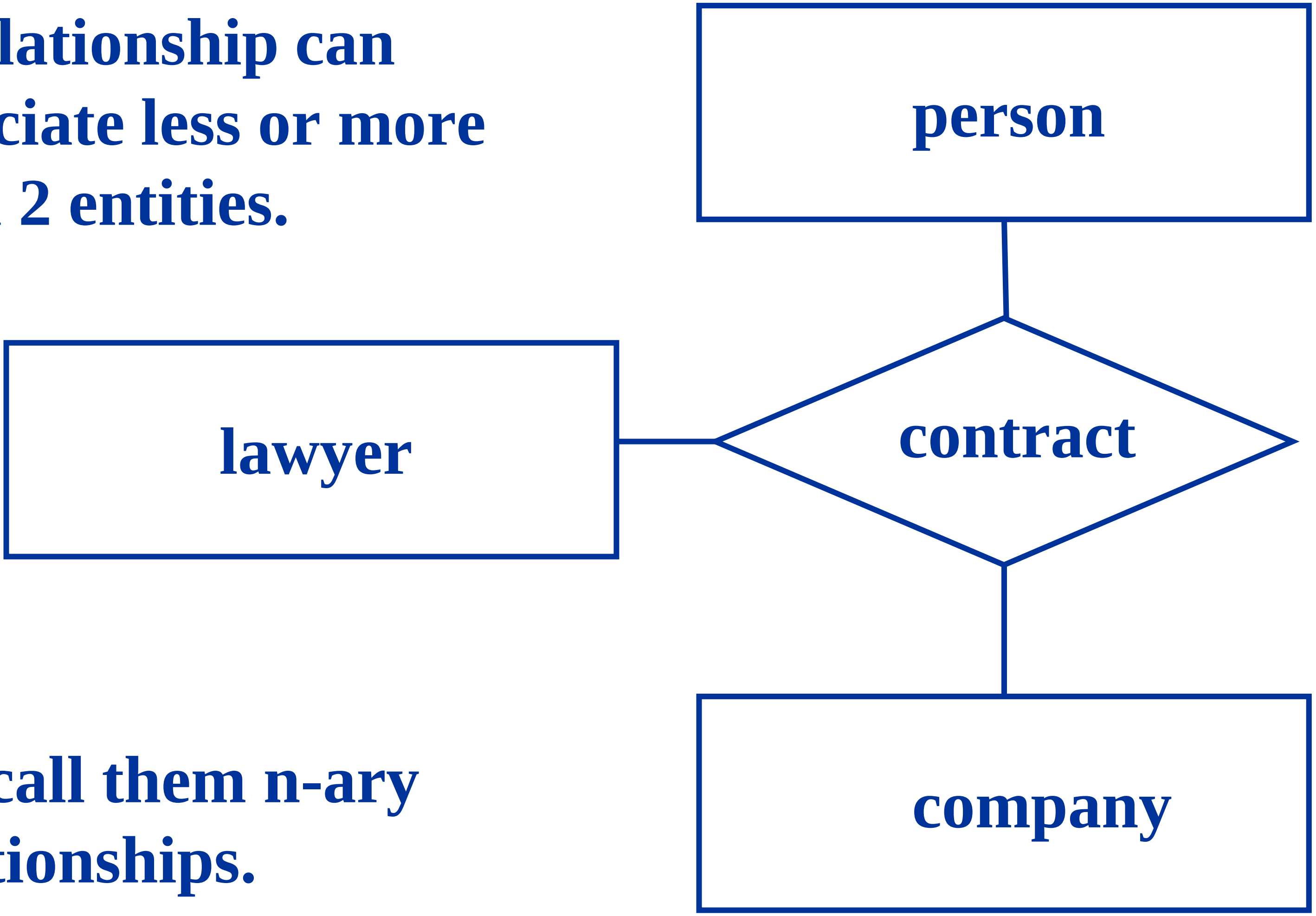
- Relationships can associate entities from the same entity set.
- In this case and in general, participation, or role, in the relationship can be named.

staff



Relationships and Relationship Sets

- A relationship can associate less or more than 2 entities.

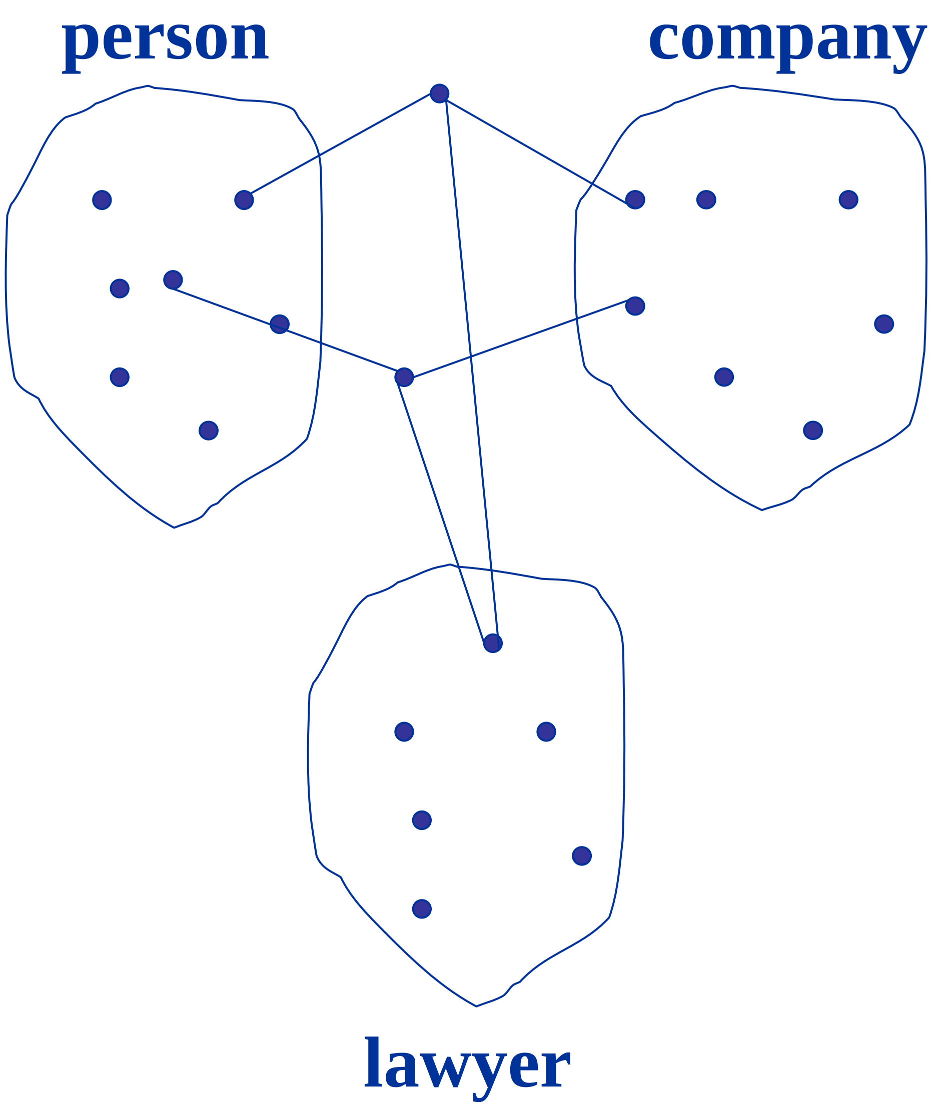


- We call them n-ary relationships.

Relationships and Relationship Sets

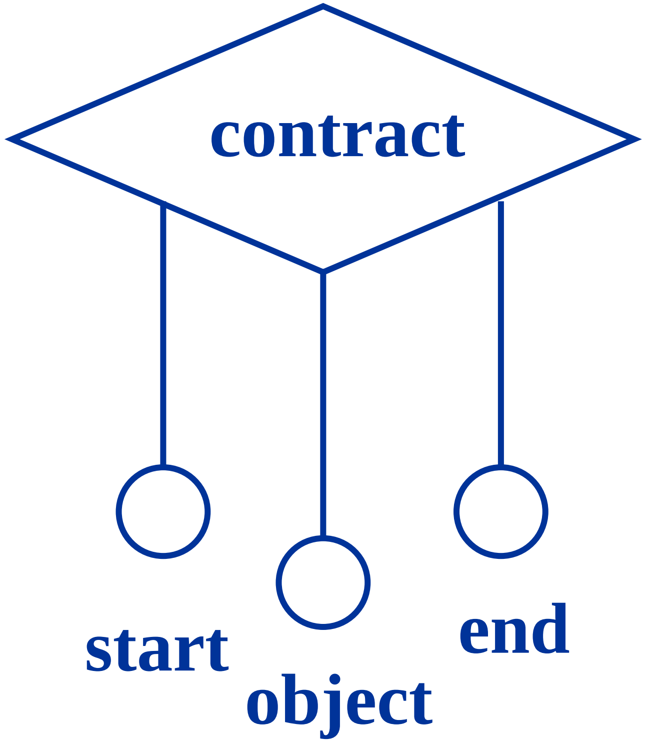
- A relationship can associate less or more than 2 entities.

- We call them n-ary relationships.



Attributes of Relationships

- Relationship can have attributes.
- All relationships in one relationship set have the same attributes.
- Relationships are distinguished not by their attributes but by their participating entities.



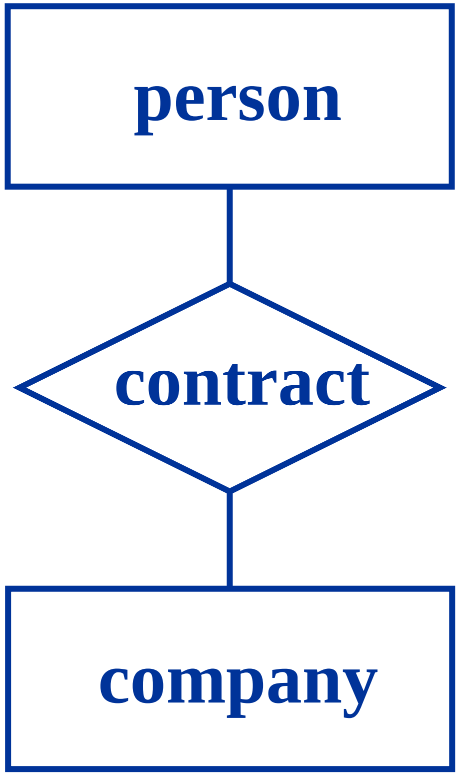
ER Model - Design Decisions

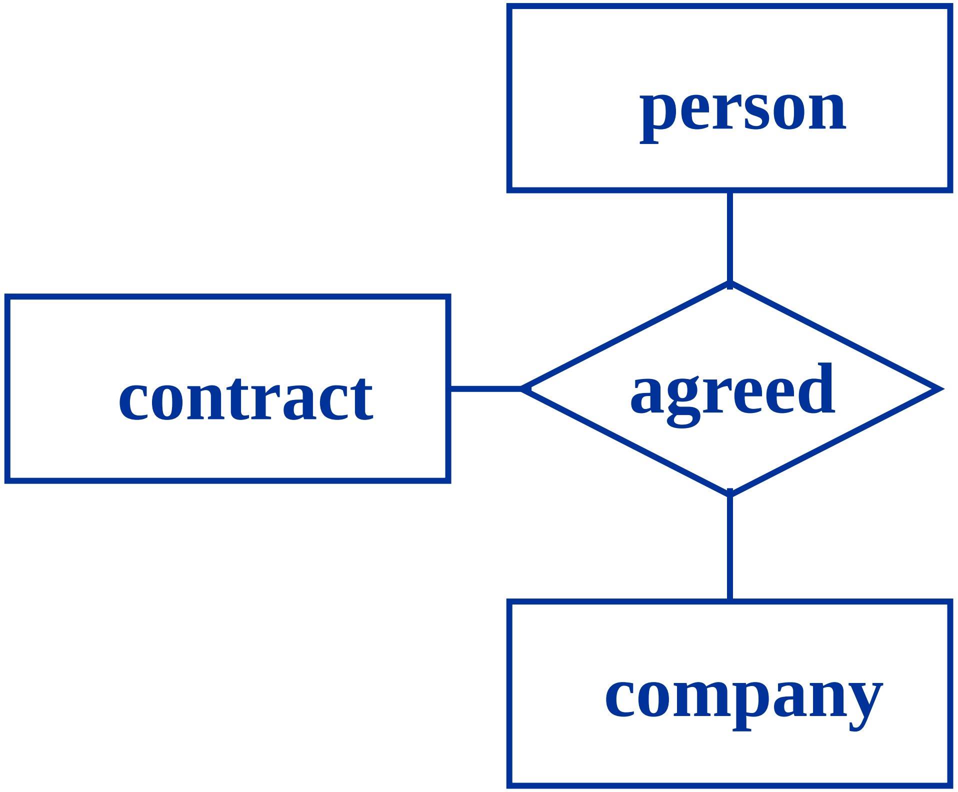
- **Should a concept be modeled as an entity or an attribute?**
- **Should a concept be modeled as an entity or a relationship?**
- **Should a concept be modeled as multiple binary relationship sets or a single n-ary relationship set?**

Entity vs. Attribute

- **Should *address* be an attribute of *person* or an entity (connected to *person* by a relationship)?**
- **Depends upon how we want to use address information, and the semantics of the data**
- **If each person has several addresses, *address* should be an entity.**

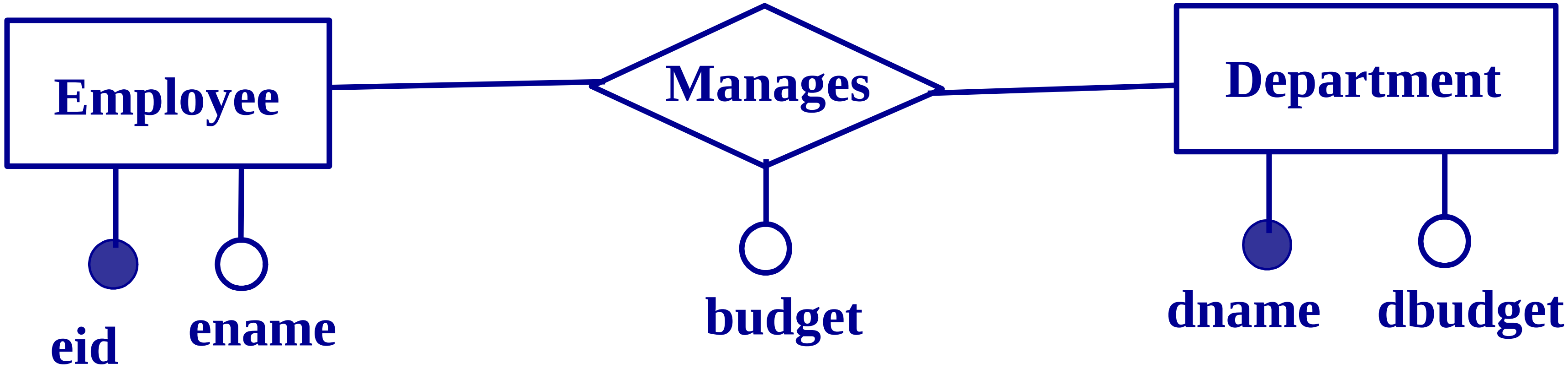
Entity or Relationship?



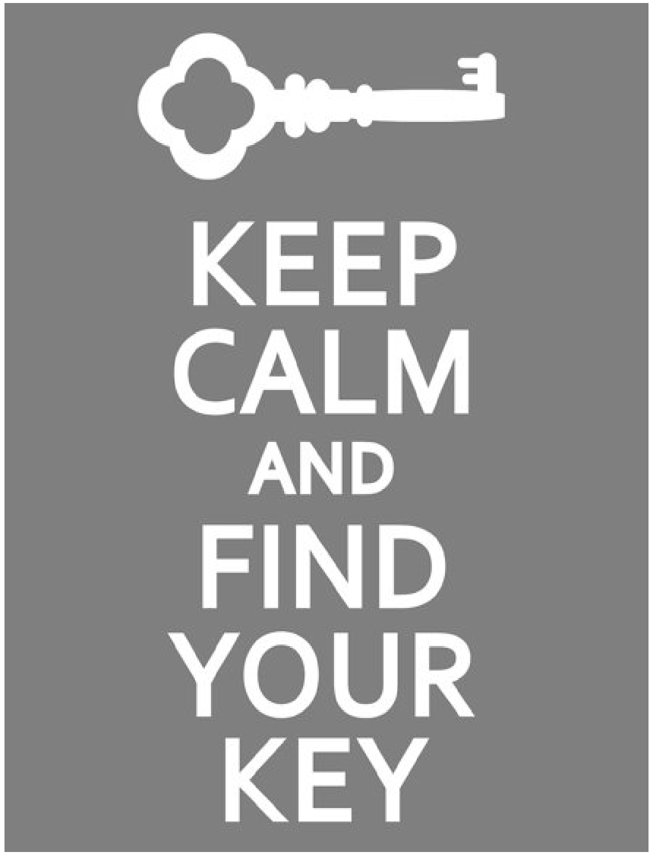


Attribute of Entity or Relationship

- ER diagram express that a manager gets a separate *discretionary budget* for each department.

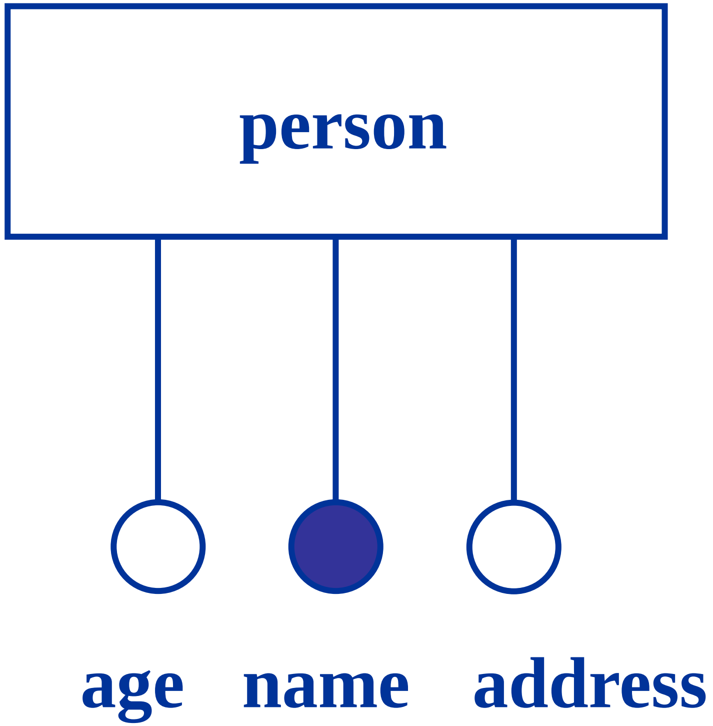


Integrity: Keys and Participation Constraints



Entities’ Identity

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

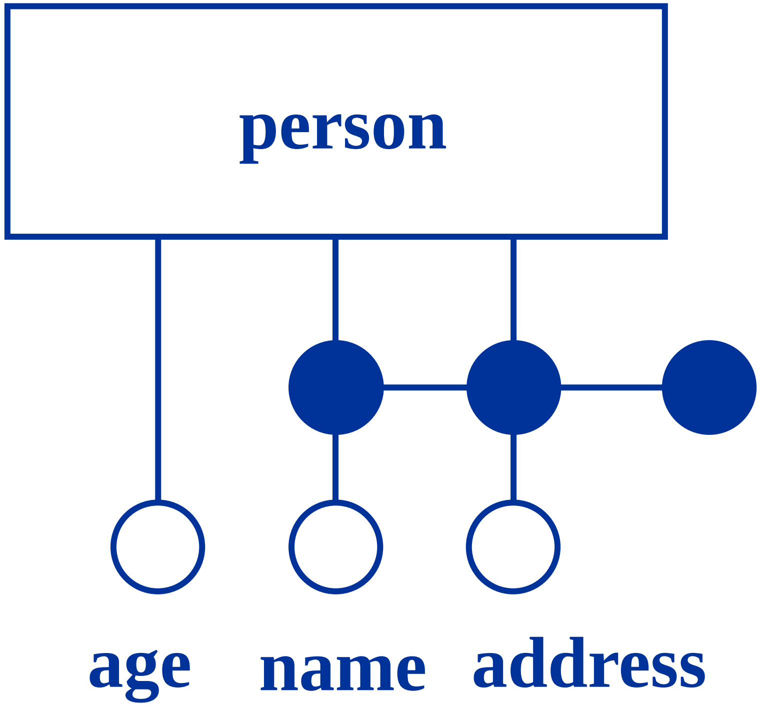


person

age name address

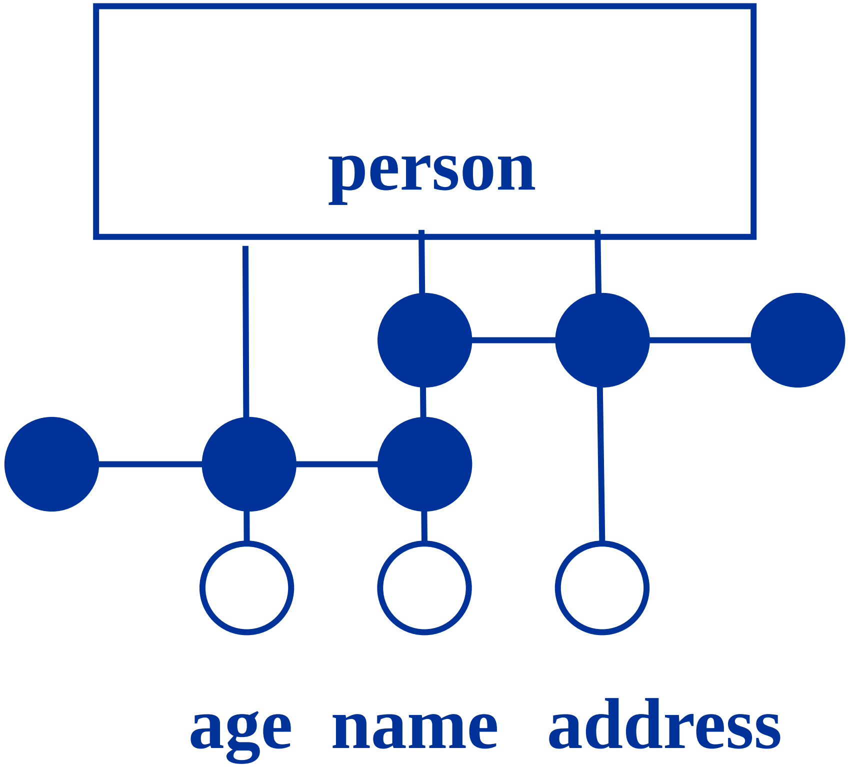
Entities’ Identity

- A combination of attributes can identify the entity.



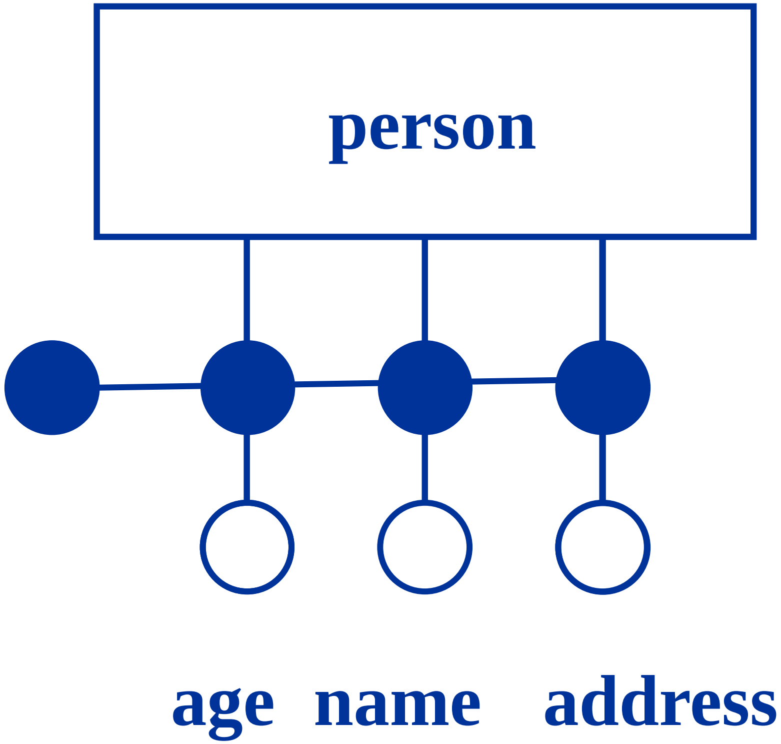
Entities' Identity

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



Entities' Identity

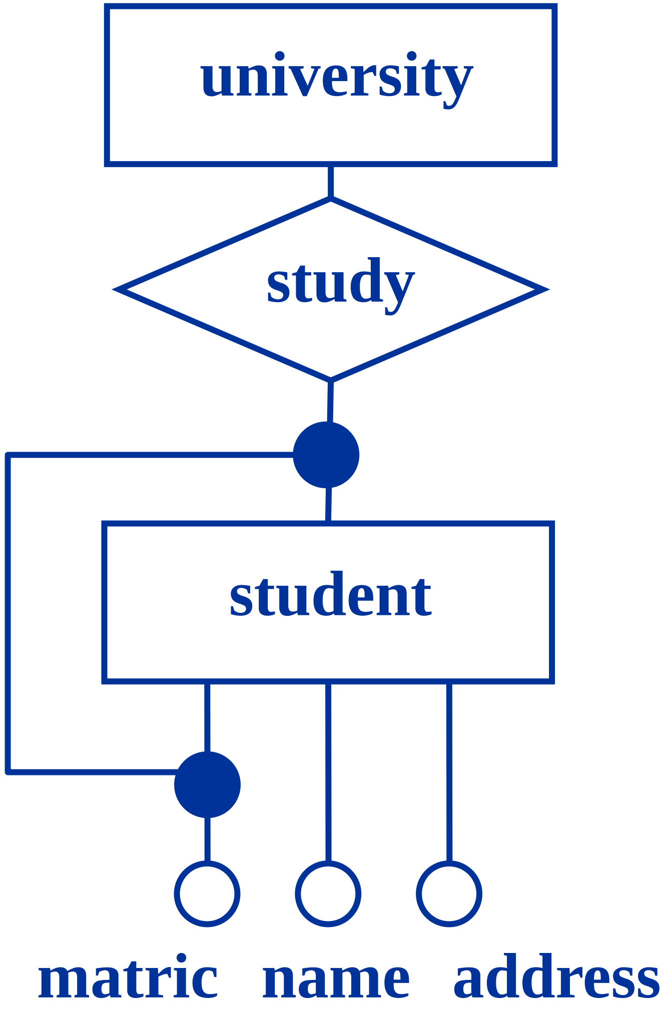
- Note: at least all attributes identify the entity
- But we prefer a minimum set of attributes.



Weak Entities

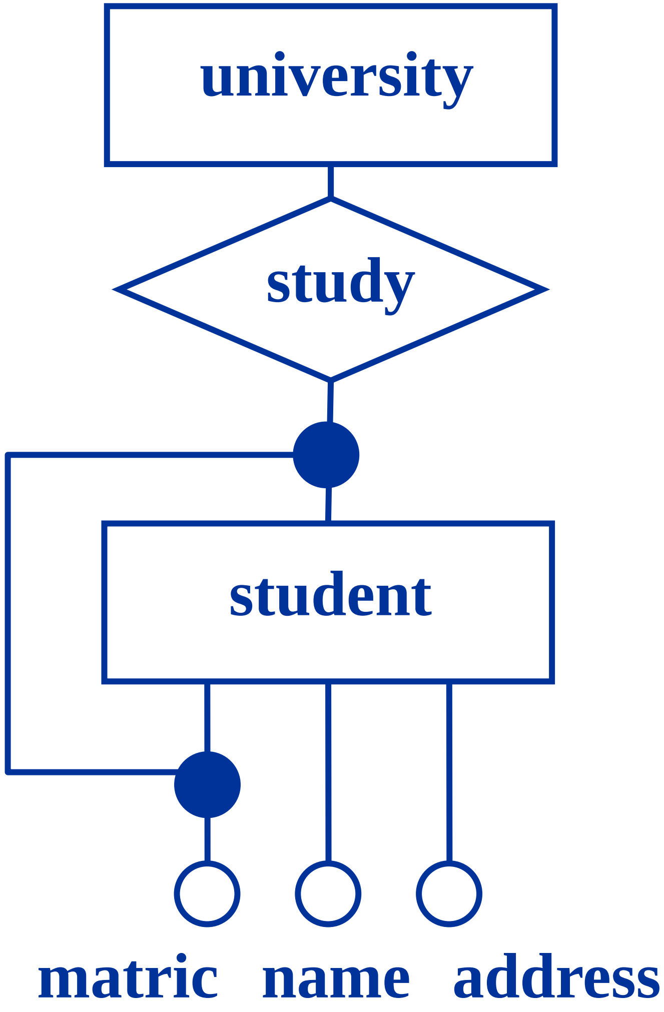
- Some entities can only be identified within the scope of a relationship with another entity set.

- Note that the relationship must exist and be unique for each entity in the set.



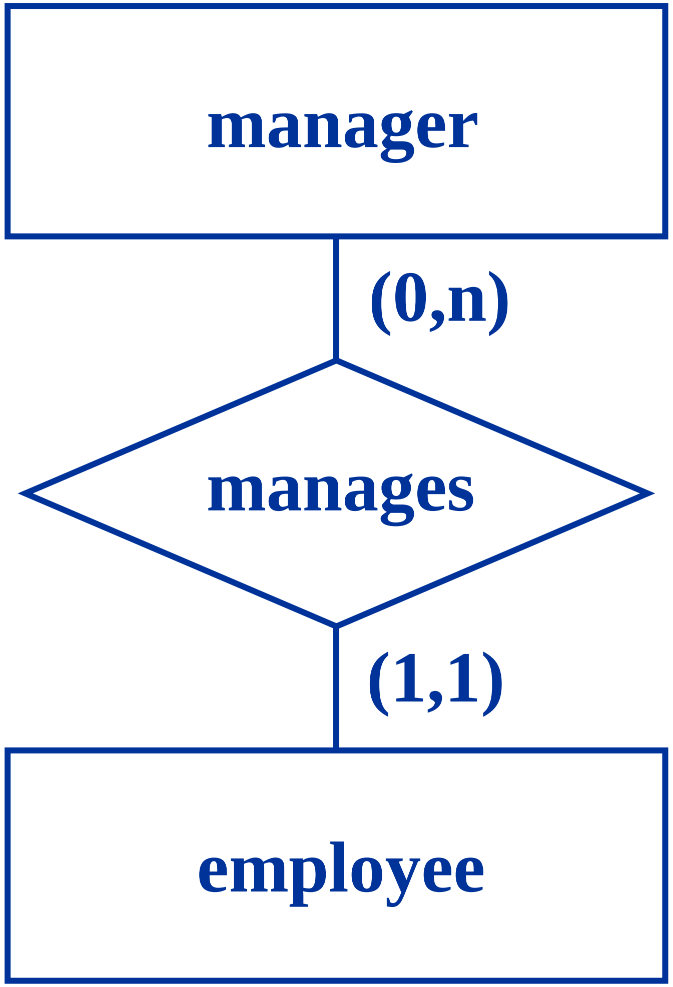
Weak Entities

- 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 cannot be identified by its attributes alone.



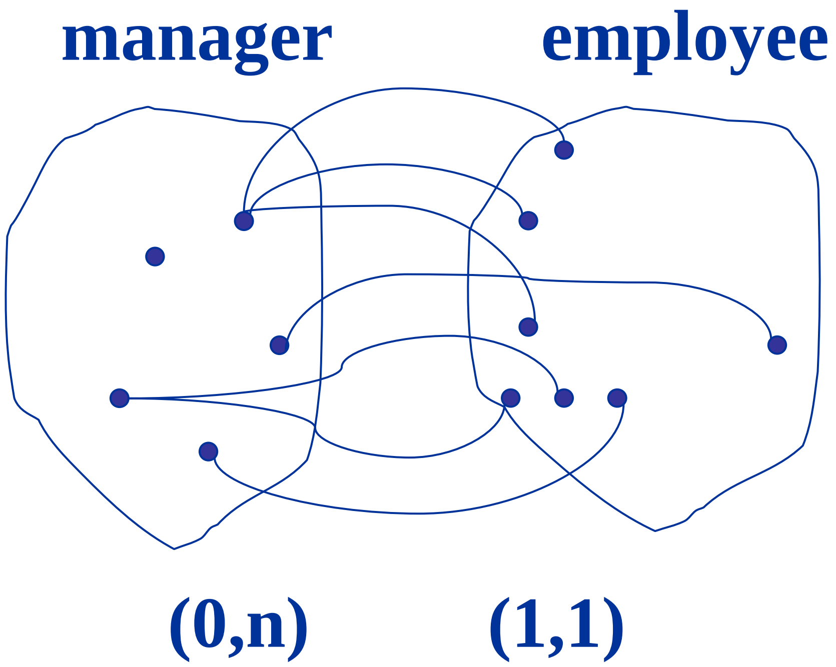
Relationships’ Cardinality

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



Relationships’ Cardinality

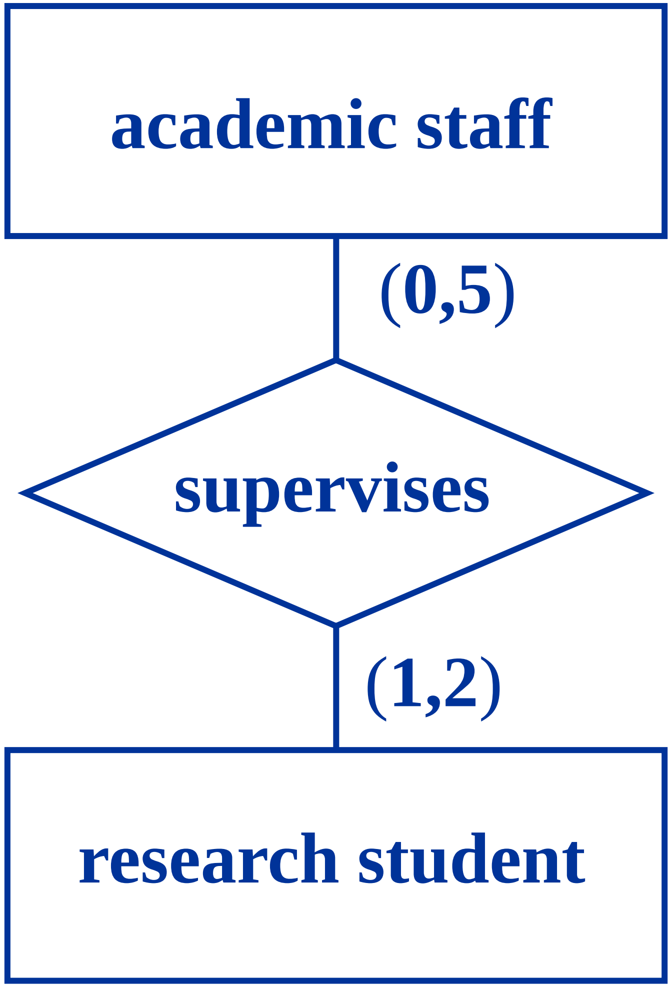
- **The cardinality of the participation in a relationship can be constrained by a minimum and maximum value: (1,1), (0, n), (2, 5).**



Relationships' Cardinality

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.**



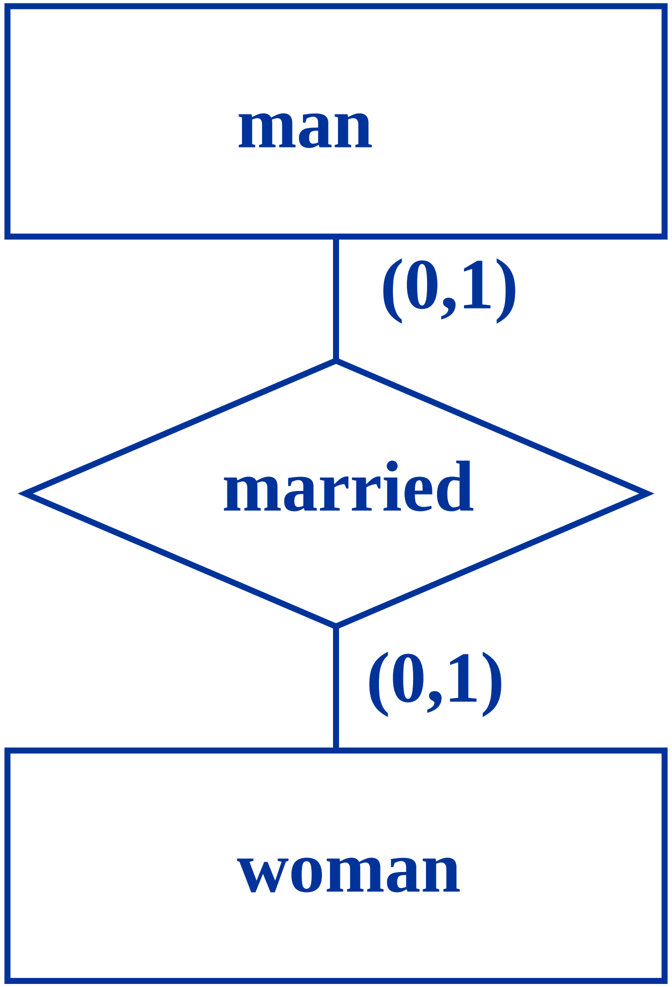
Relationships' Cardinality

- (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.

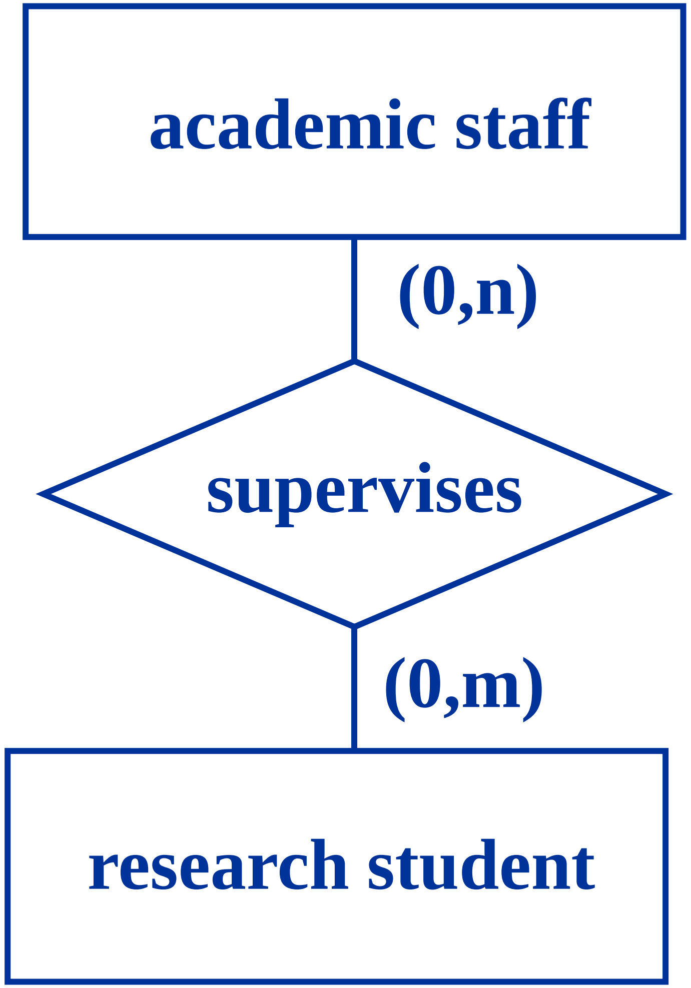
Relationships’ Cardinality

- **Example of a one-to-one relationship.**



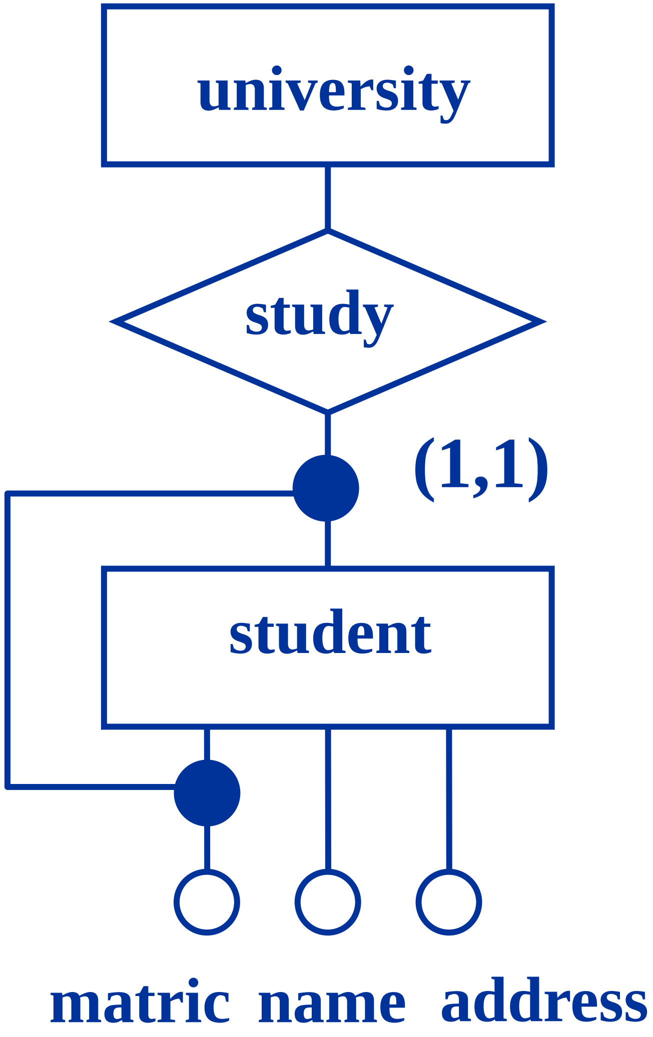
Relationships’ Cardinality

- **By default we have many-to-many relationships.**



Weak Entities

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

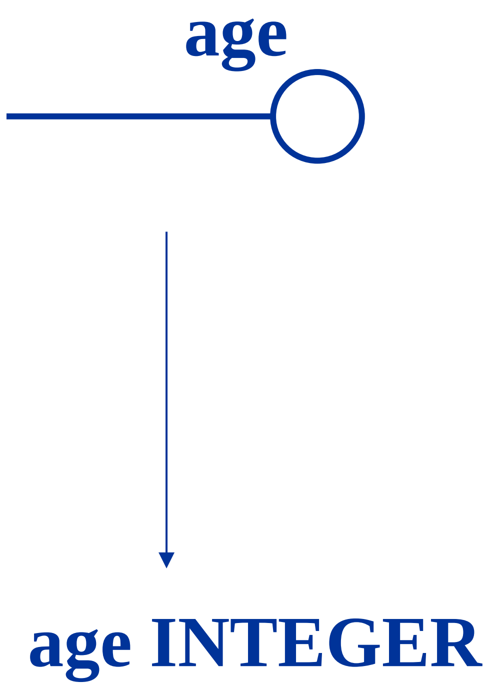


Conceptual to Logical Design

From E-R to Relational:
3 Rules and 3 Exceptions

Rule 1: Value Sets

- **Value sets are mapped to domains.**
- **In practice this is a first step towards the physical design.**
- **E-R attributes are mapped to attributes of relations.**

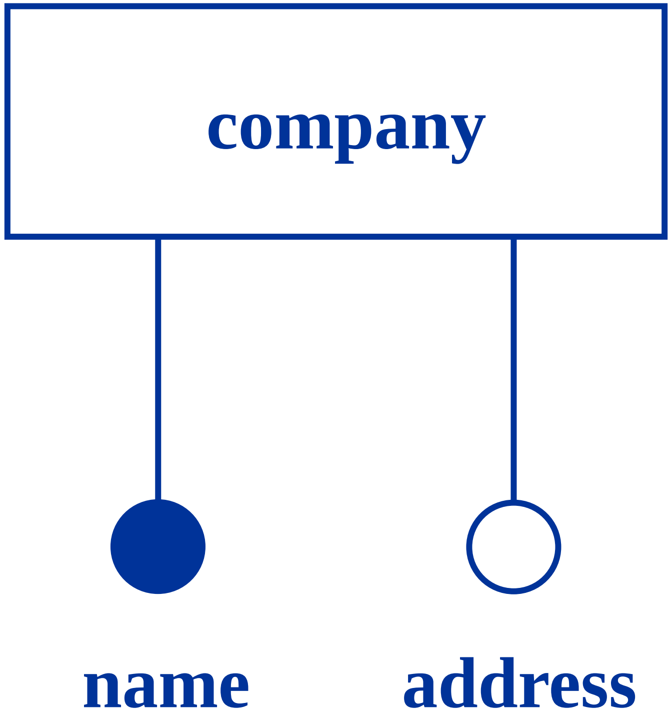


age

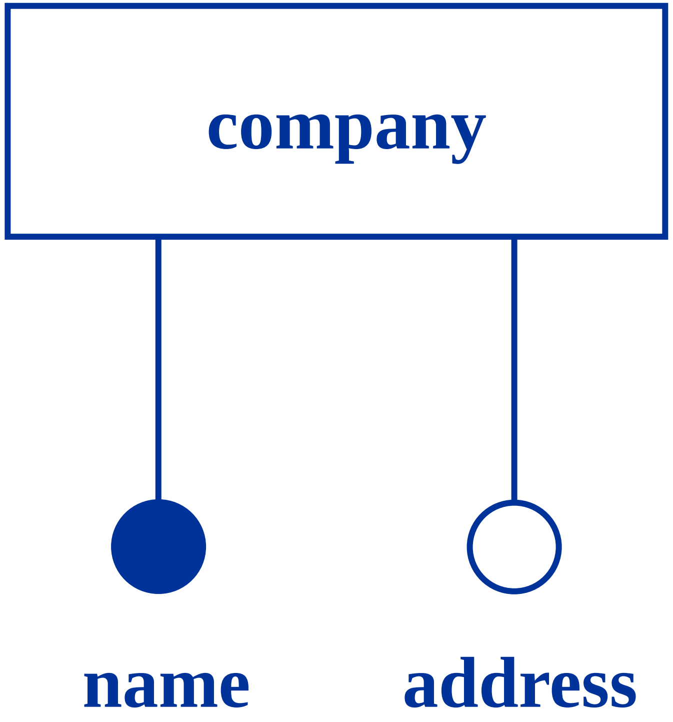
age INTEGER

Rule 2: Entity Sets

- Entity sets are mapped to relations.
- The entity set attributes are mapped to attributes of the relation.
- The keys are mapped to primary key.

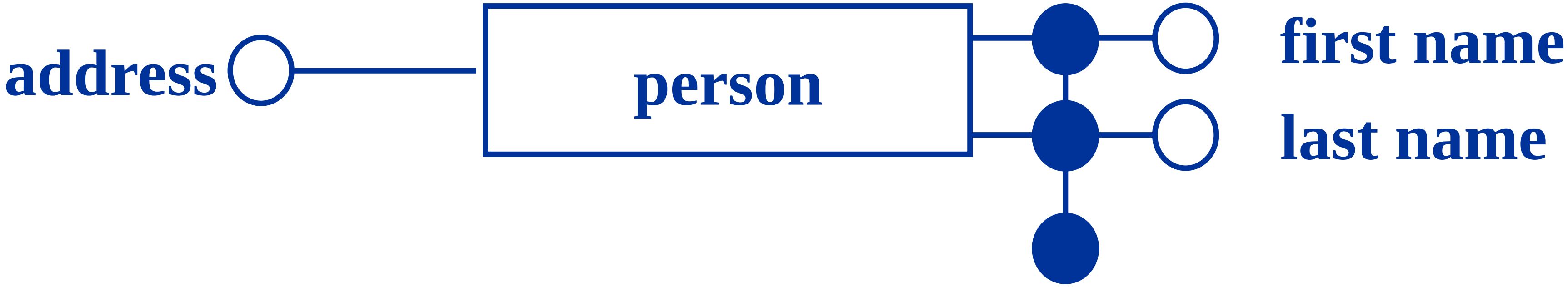


Rule 2: Entity Sets



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

Rule 2: Entity Sets



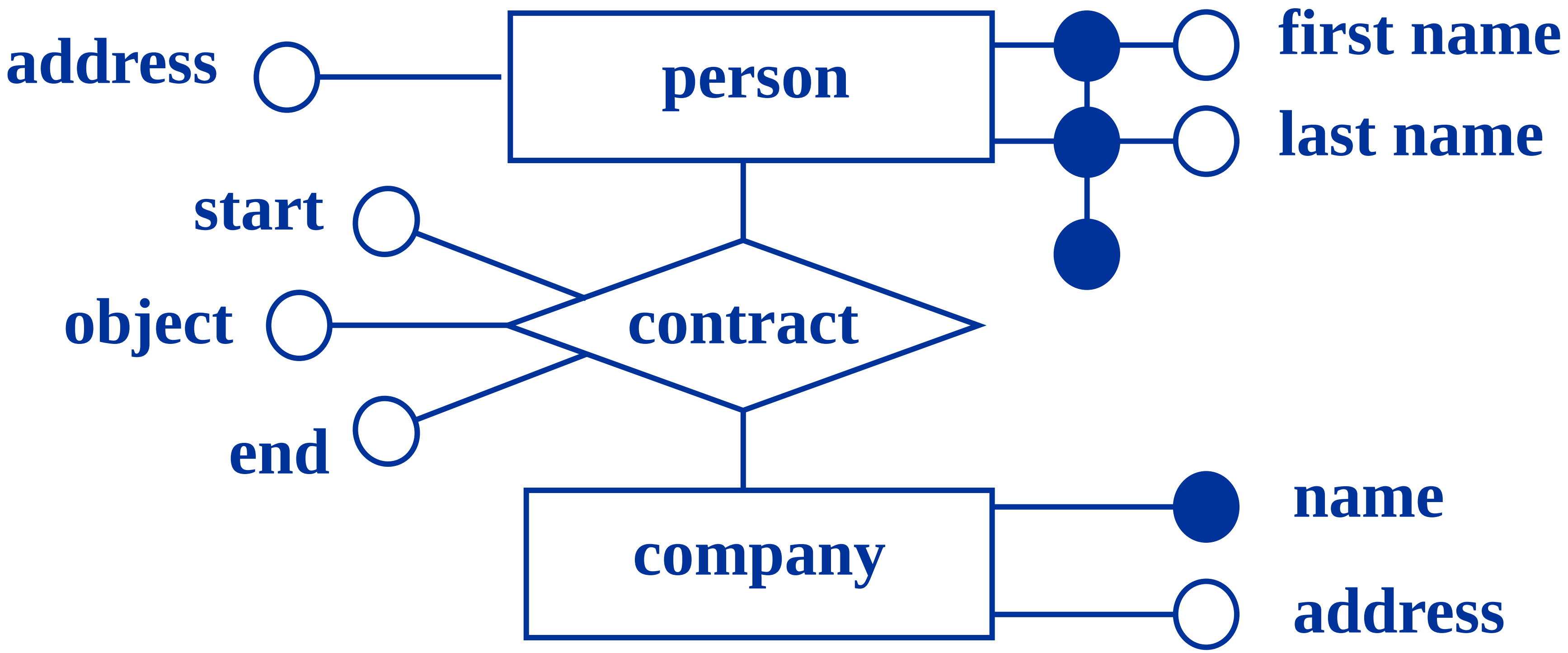
```
CREATE TABLE person (
first name VARCHAR(32),
last_name VARCHAR(32),
address VARCHAR(128),
PRIMARY KEY (first_name, last_name))
```

Rule 3: Relationship Sets

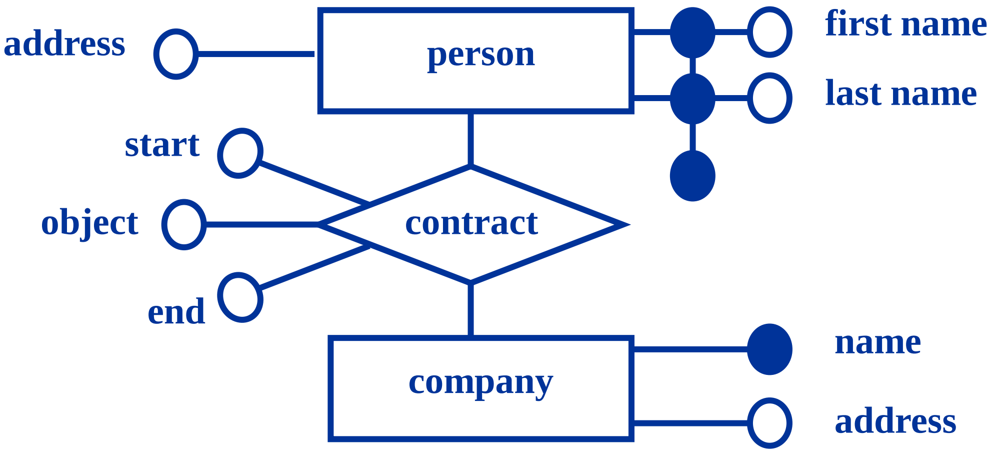
- Relationship sets are mapped to relations.
- The attributes of the relation consist of the attributes of the relationship set,
- As well as of the keys of the participating entities.

contract

Rule 3: Relationship Sets



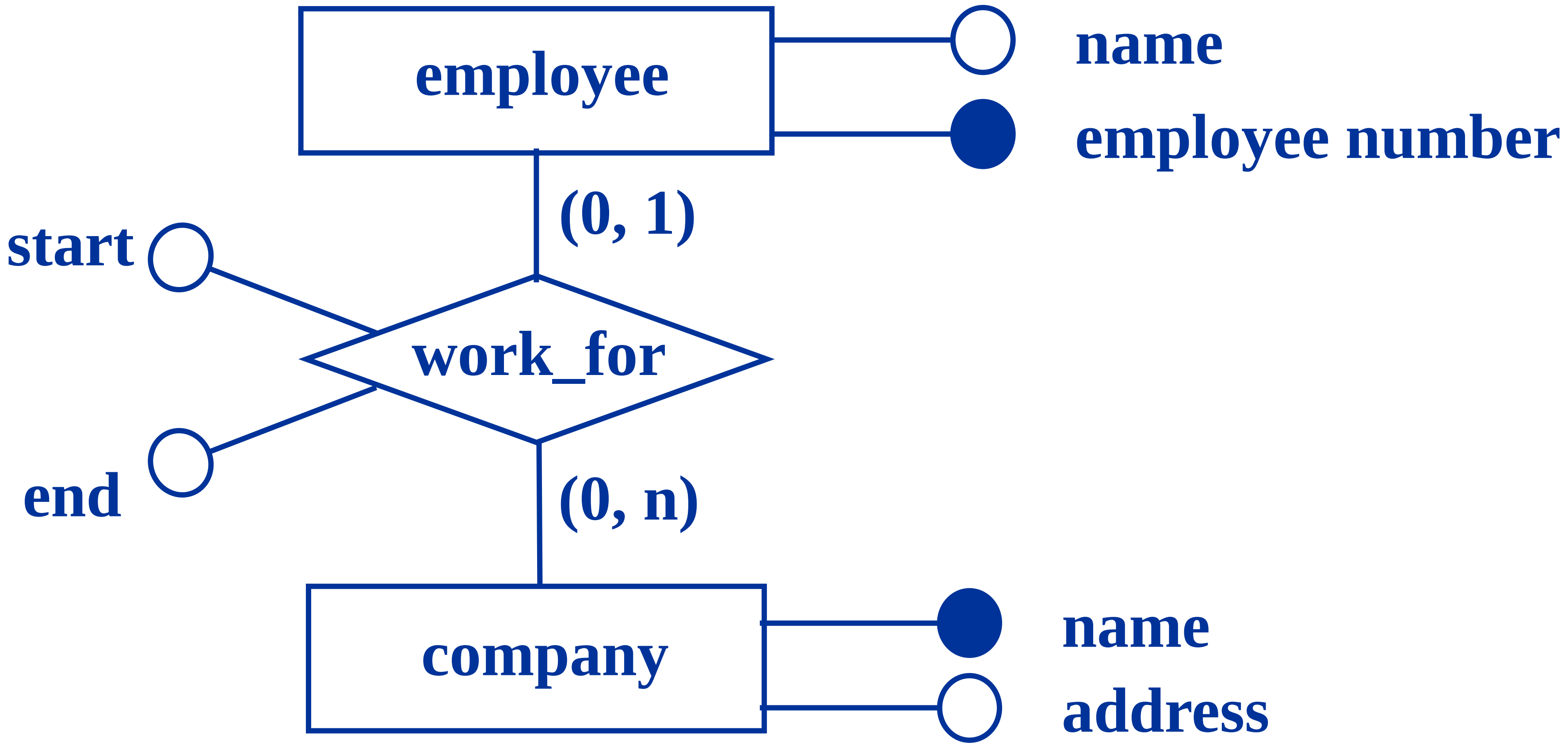
Rule 3: Relationship Sets



```
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))
```

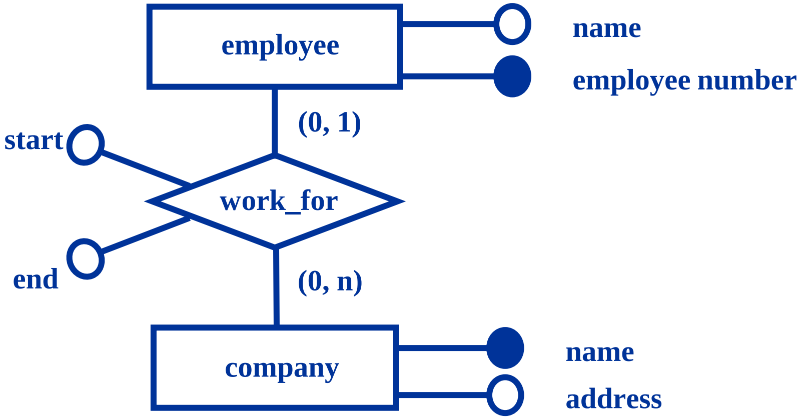
Exception 1: One-to-many Relationships

- A one-to-many relationship indicates a key constraint



Exception 1: One-to-many Relationships

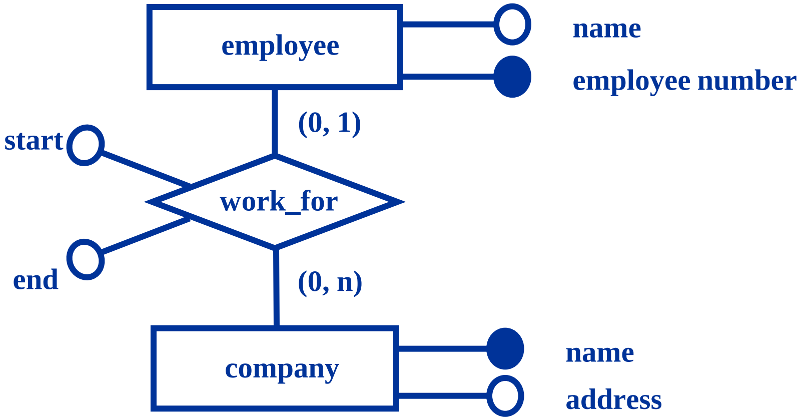
- The primary key of the relationship table is inadequate



```
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))
```

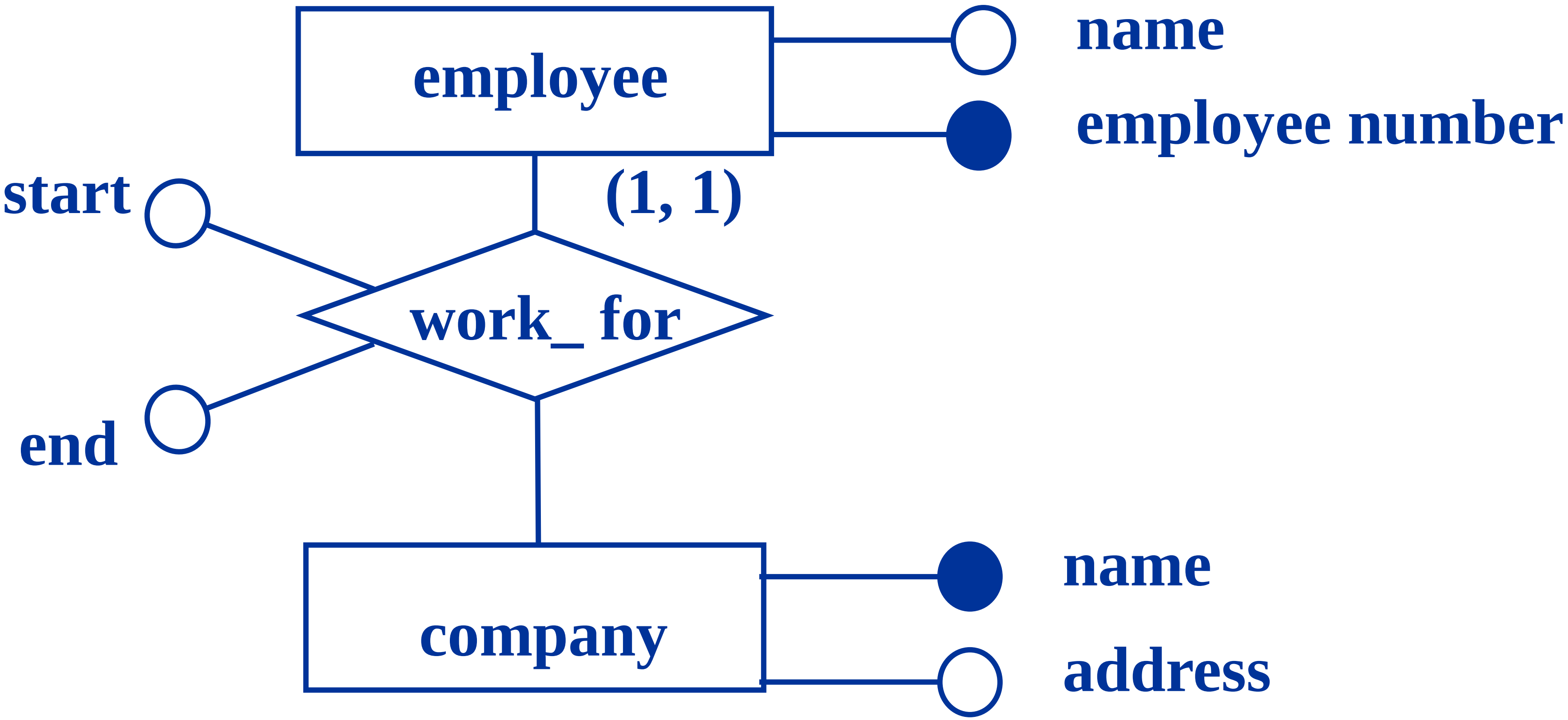
Exception 1: One-to-many Relationships

- We change the primary key of the relationship table



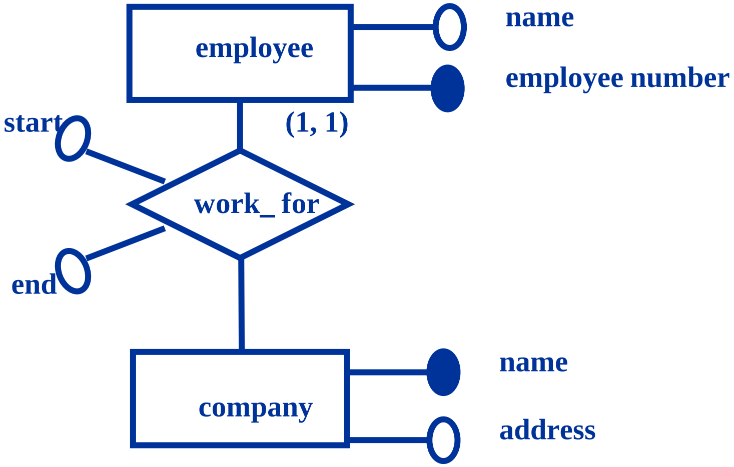
```
CREATE TABLE work_for(
start DATE,
end DATE,
enumber CHAR(8) PRIMARY KEY,
cname VARCHAR(32),
FOREIGN KEY (enumber) REFERENCES employee(number),
FOREIGN KEY (cname) REFERENCES company(name))
```

Exception 2: (1, 1) Participation Constraints



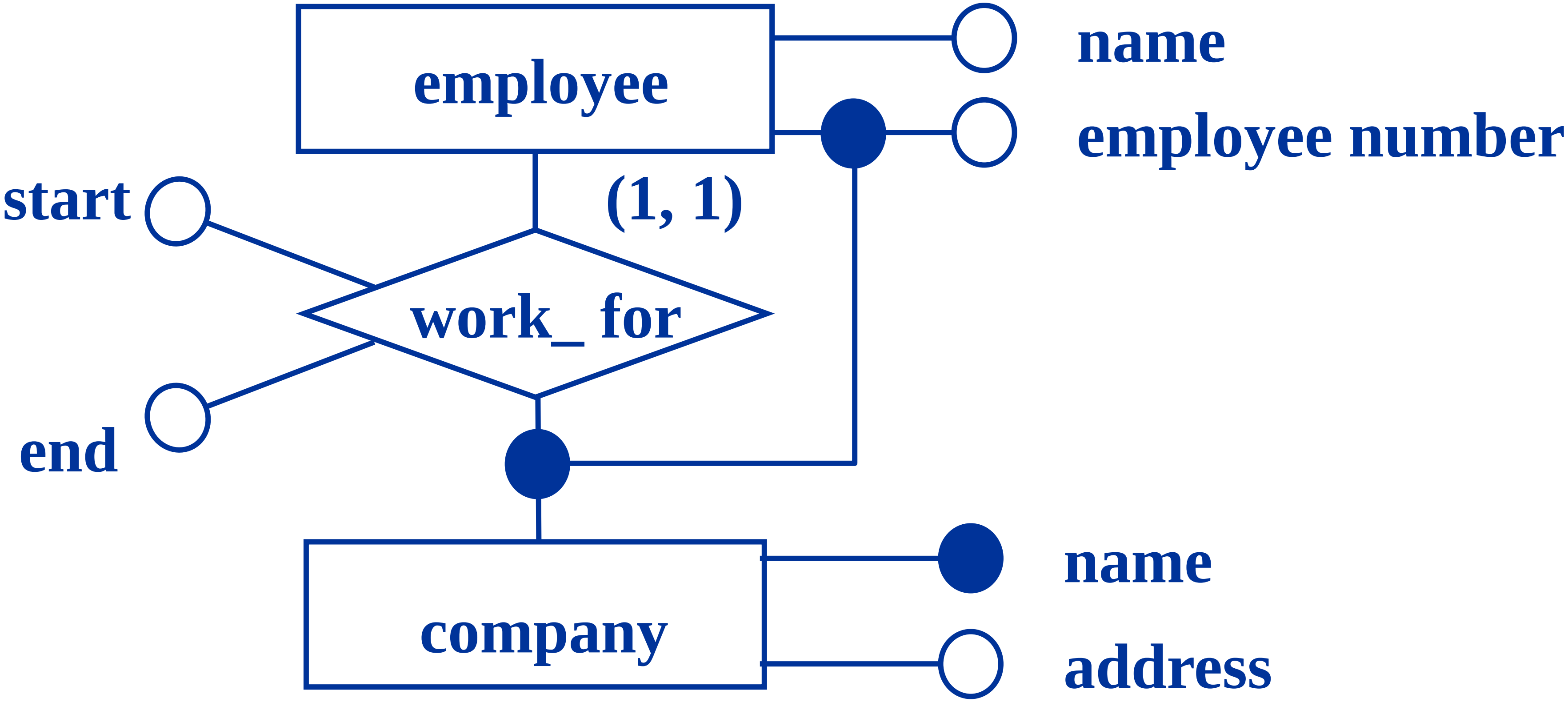
Exception 2: (1, 1) Participation Constraints

- We merge the table employee and the table work_for and use the primary key of the employee table.



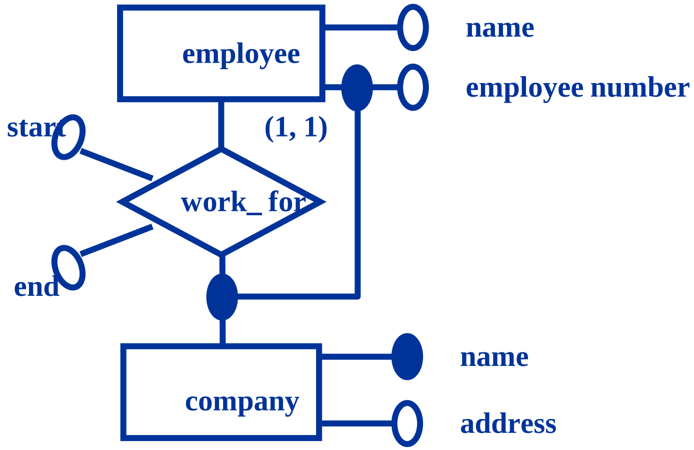
```
CREATE TABLE employee_work_for (
start DATE,
end DATE,
enumber CHAR(8) PRIMARY KEY,
ename CHAR(32),
cname VARCHAR(32),
FOREIGN KEY (cname) REFERENCES company(name))
```

Exception 3: Weak Entity



Exception 3: Weak Entity

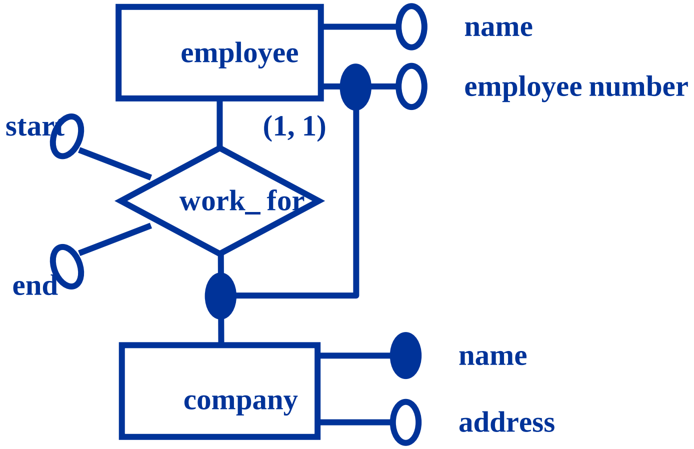
- The primary key of the employee table is not enumber because it is a weak entity.



```
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.

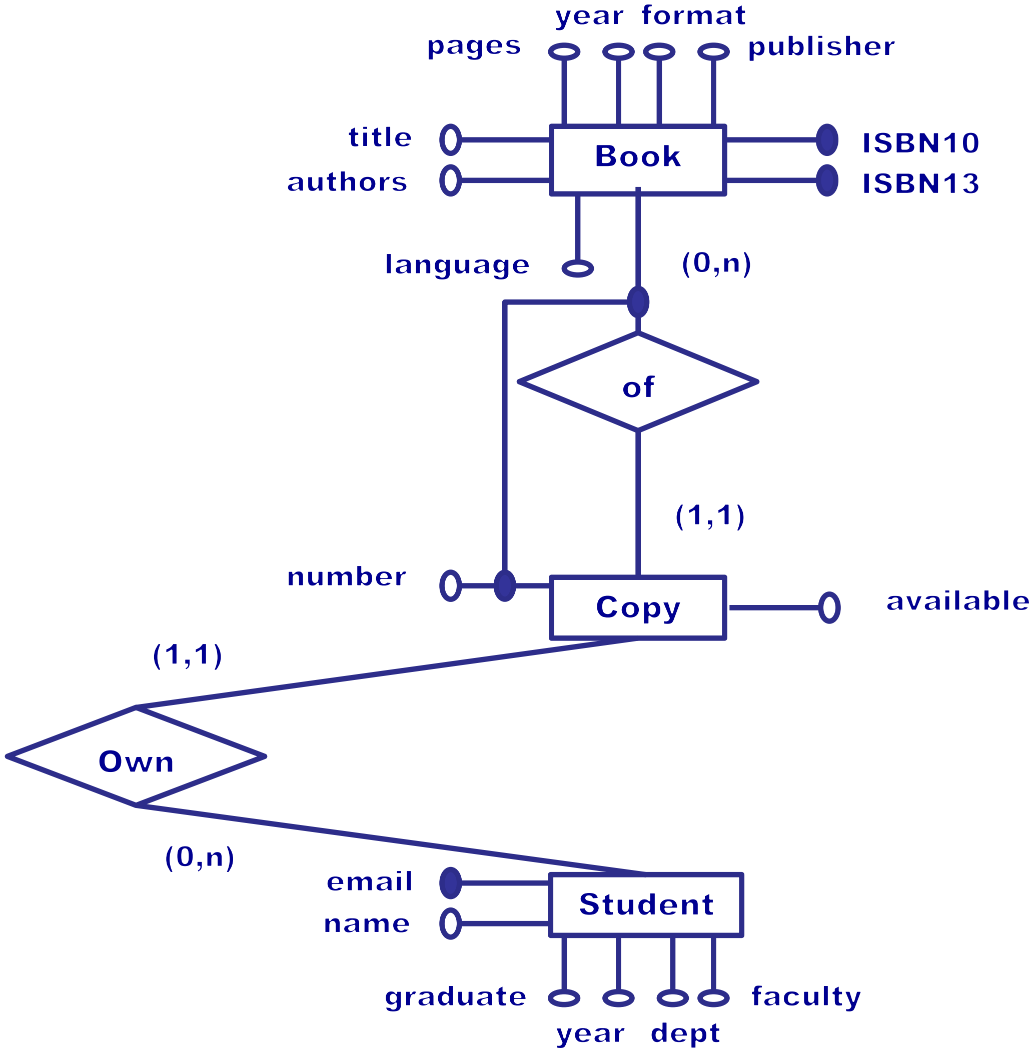


```
CREATE TABLE employee_work_for(
start DATE,
end DATE,
enumber CHAR(8),
ename CHAR(32),
cname VARCHAR(32),
PRIMARY KEY (enumber, cname),
FOREIGN KEY (cname) REFERENCES company(name))
```

Summary

- ER model has expressive constructs for conceptual data design
  - Concepts: entities, relationships, attributes, weak entities
  - Constraints: key constraints, participation constraints
- ER design is subjective
- Rules for mapping ER model to relational model
  - Entity and relationship sets
  - Key constraints
  - Participation constraints
  - Relationship roles
  - Weak entity sets

Exercise: Complete the ER Diagram for the SQL Lab



CREDITS

The content of this lecture is based on Chapter 1 of the book
“Introduction to database Systems”
by
S. Bressan and B. Catania
McGraw Hill publisher