Introduction to database design

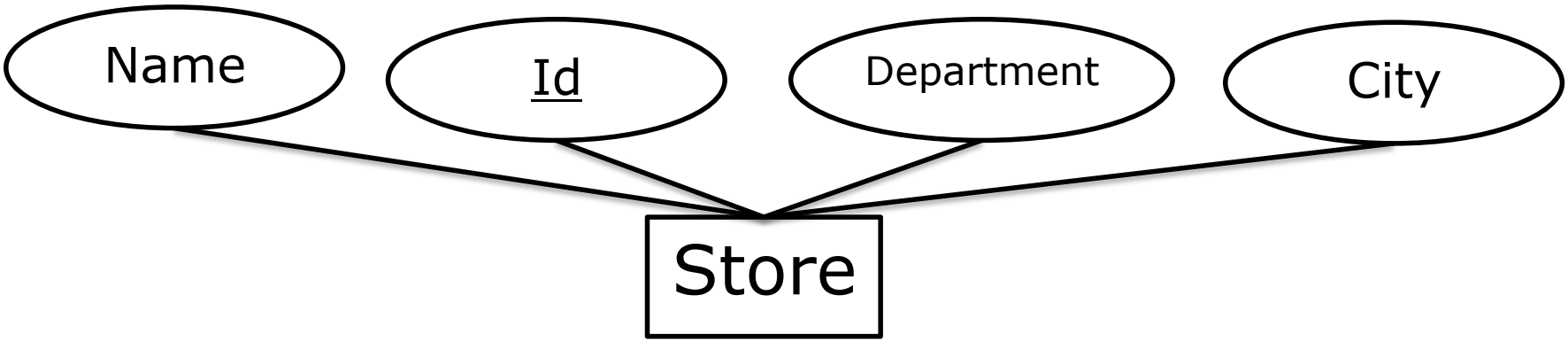
RG 2, 3.5, 3.8,
and note with examples of different ER notations

Carsten Schürmann

Today's lecture

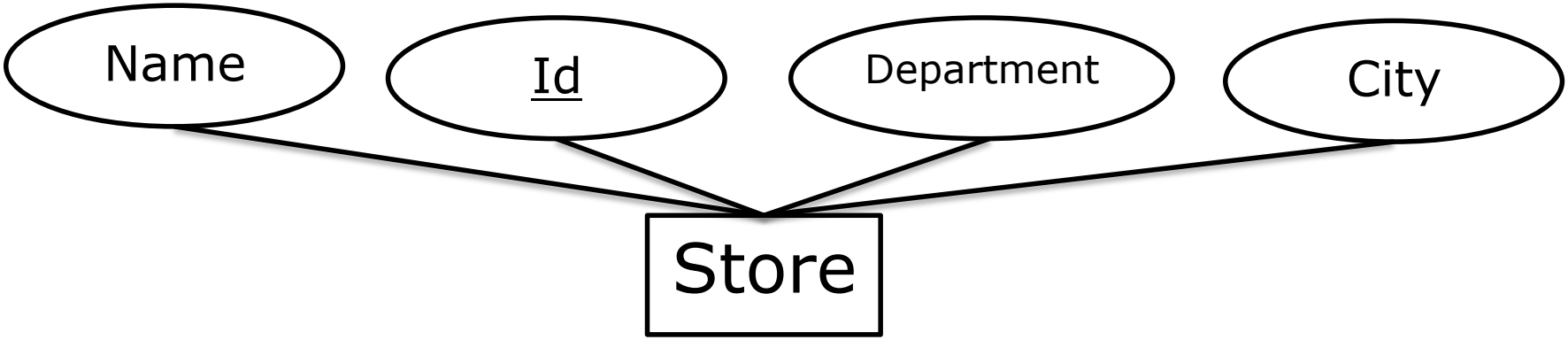
- Data Models
- Entity-Relationship (E-R) model concepts
- Translation to a schema in SQL
- Hand-in, part 1

Entity



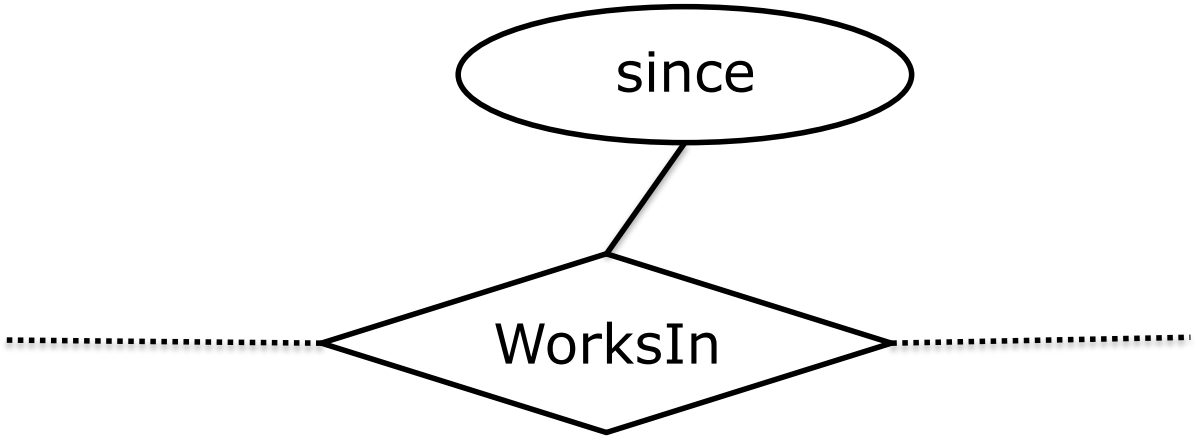
- An **entity** models an **object**
- An **entity type** (rectangle) models a class of entities
  - Described by **attributes** (ovals)
  - Can have a **key** (underlined)

Corresponding relation




```
CREATE TABLE Store (
Id INTEGER PRIMARY KEY,
Name INTEGER,
Department VARCHAR(20),
City VARCHAR(20));
```

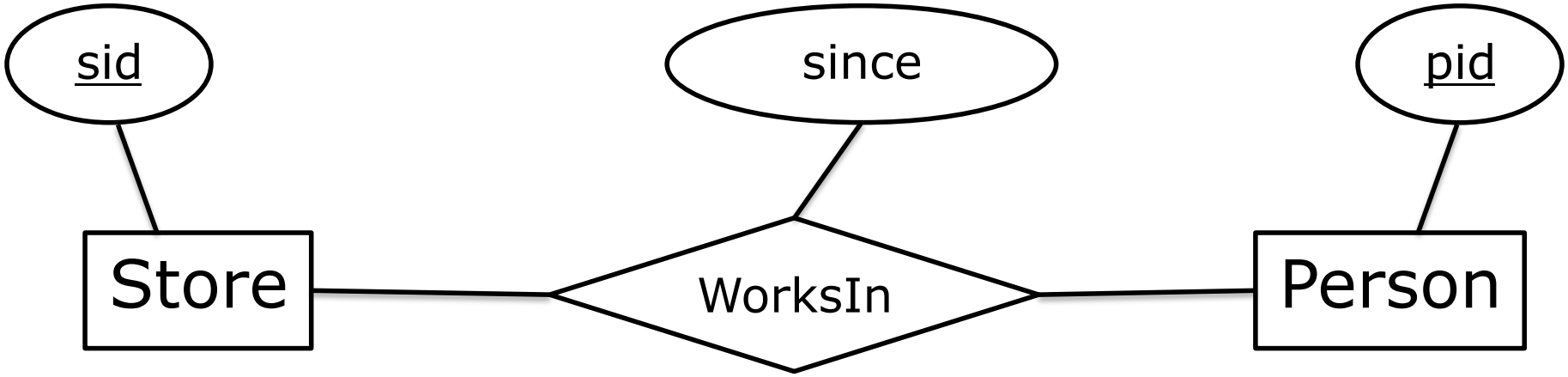
Relationships



A relationship type (diamond) models a mechanism that **relates** entities to each other.

A relationship type may have **attributes**.

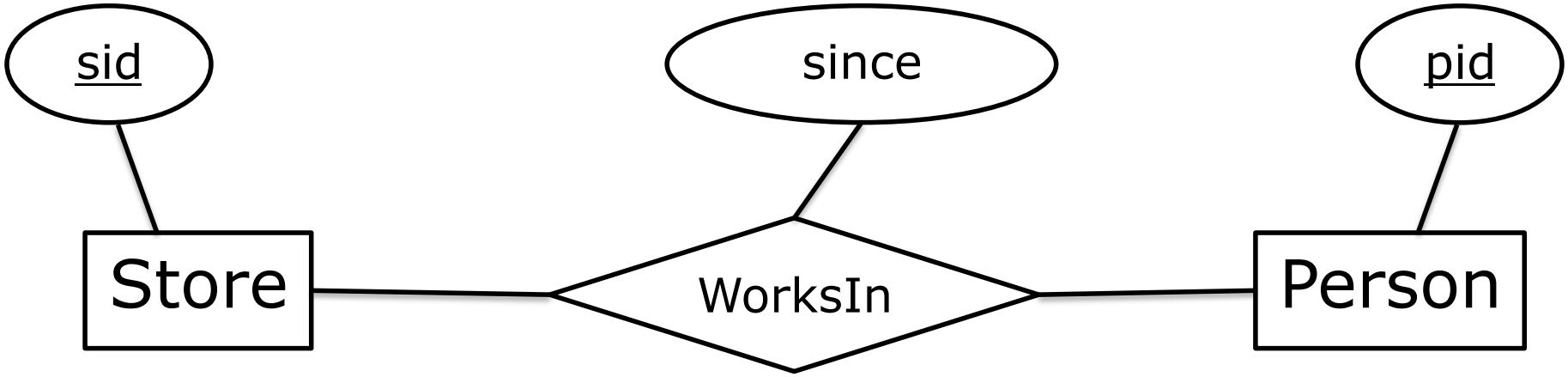
Relationships



A relationship is connected to one, two or more entity types, called **roles**.

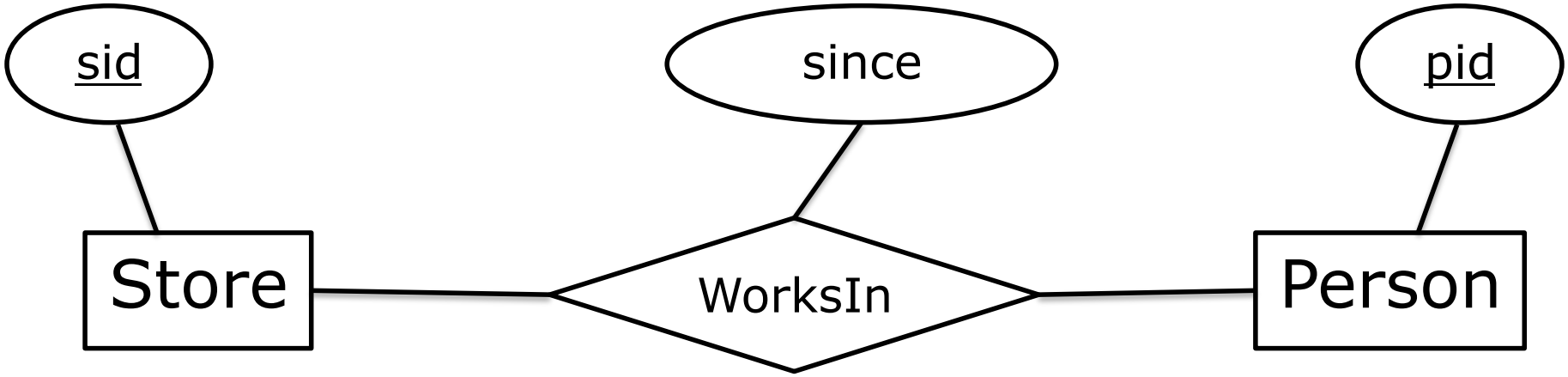
A role specifies **how** an entity participates in a relationship.

Corresponding relation



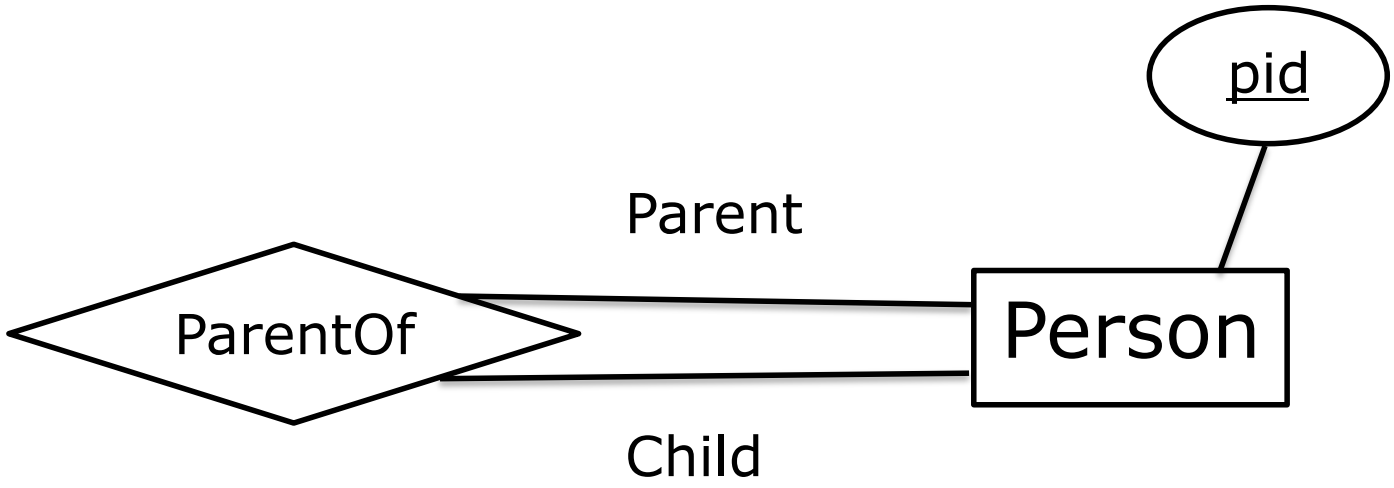

```
CREATE TABLE WorksIn (
pid INTEGER NOT NULL REFERENCES Person(pid),
sid INTEGER NOT NULL REFERENCES Store(sid),
since DATE,
PRIMARY KEY (sid,pid));
```

Problem session



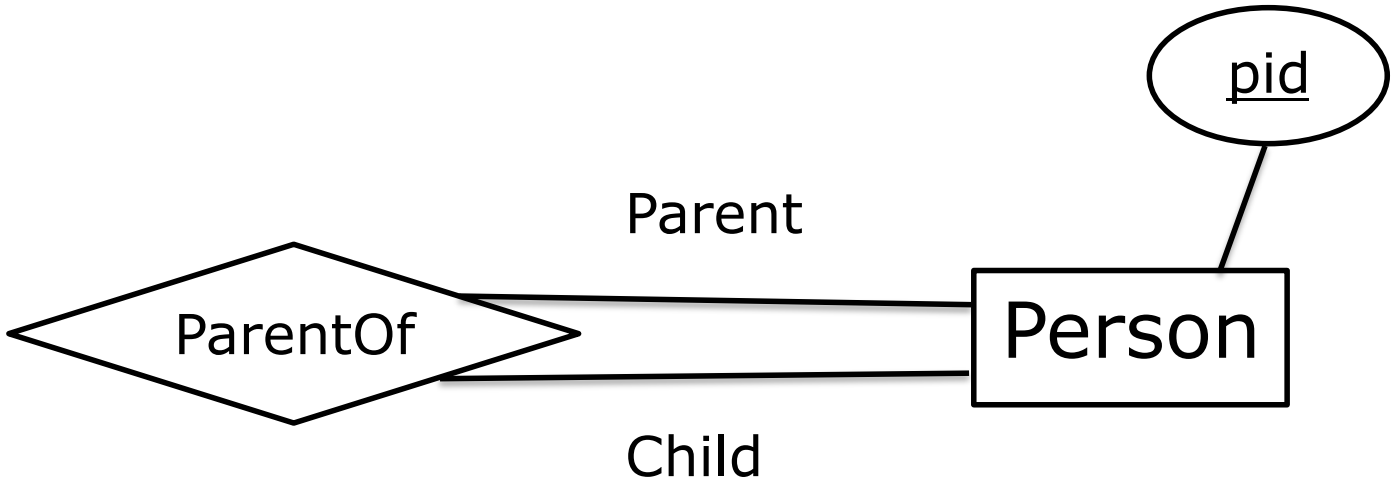
- Suppose that workers are employed on several 1-year contracts
  - Why does the above not model this?
  - How should one modify the E-R diagram?

Multiple roles



One entity can participate in **several** roles in a relationship.

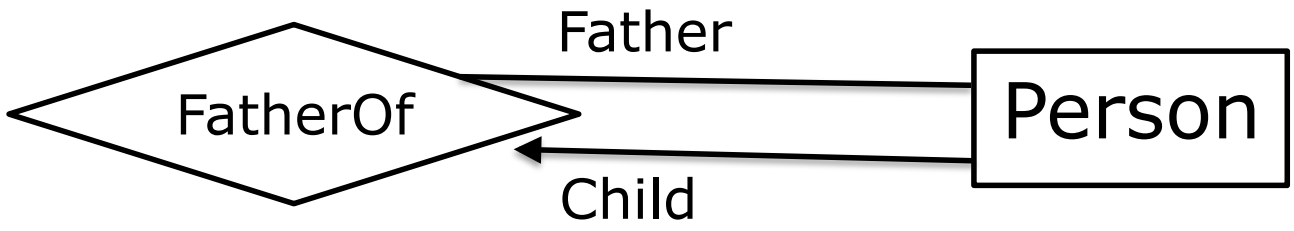
Corresponding relation




```
CREATE TABLE ParentOf(
parId INTEGER NOT NULL REFERENCES Person(pid),
chdId INTEGER NOT NULL REFERENCES Person(pid),
PRIMARY KEY (parId,chdId));
```

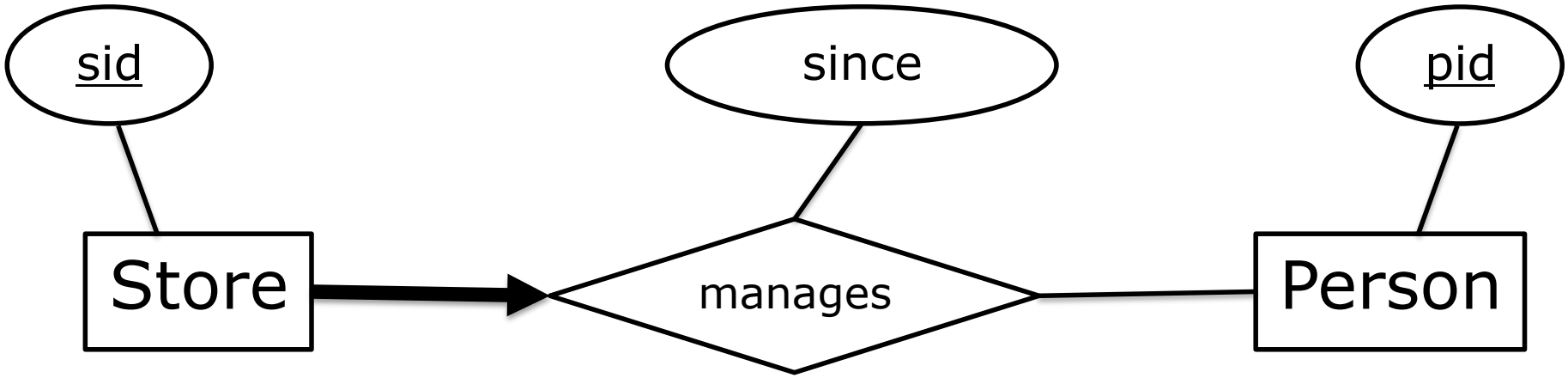
Cardinality constraints

- Constrains the number of times an entity can participate in a relationship in a specific role.
- RG uses arrows to indicate the "key constraint" that an entity can only occur once in a relationship set.



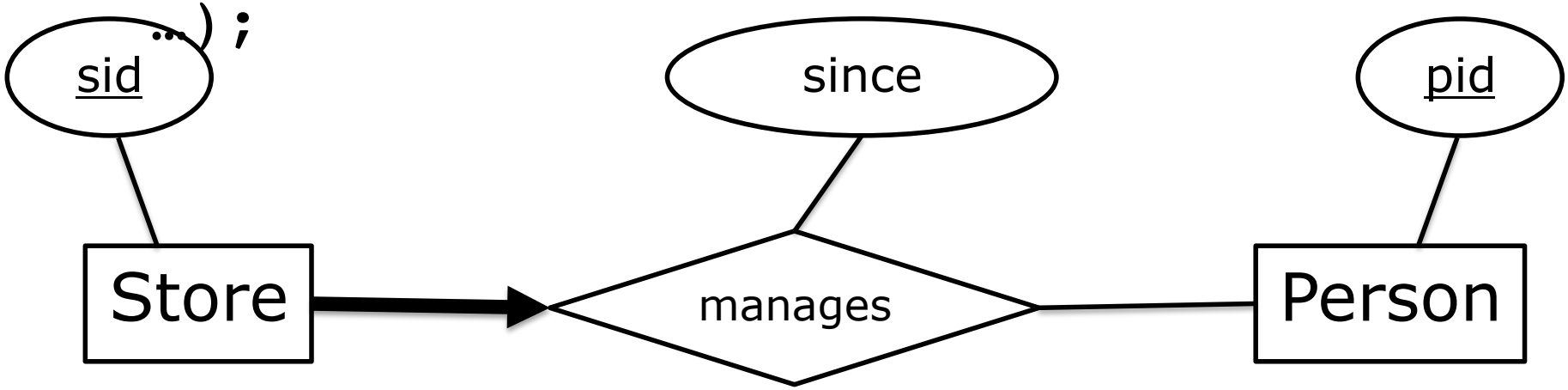
Participation constraints

- Says that an entity **must** participate in a relationship in a specific role.
- RG uses bold lines to indicate this.

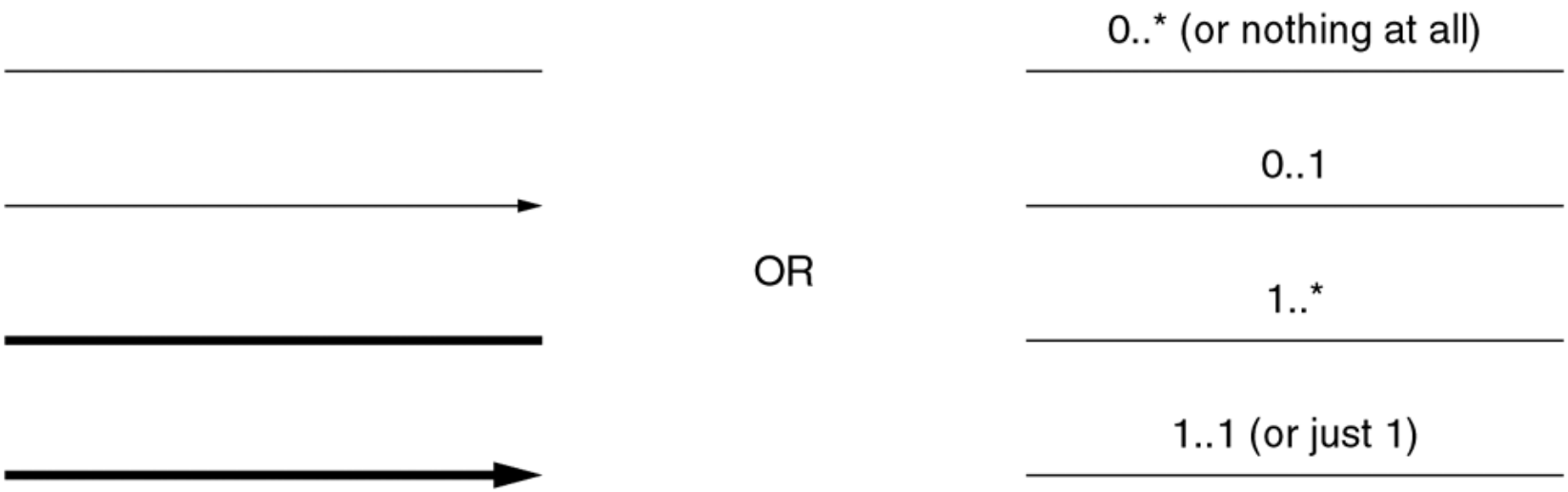


Corresponding relation

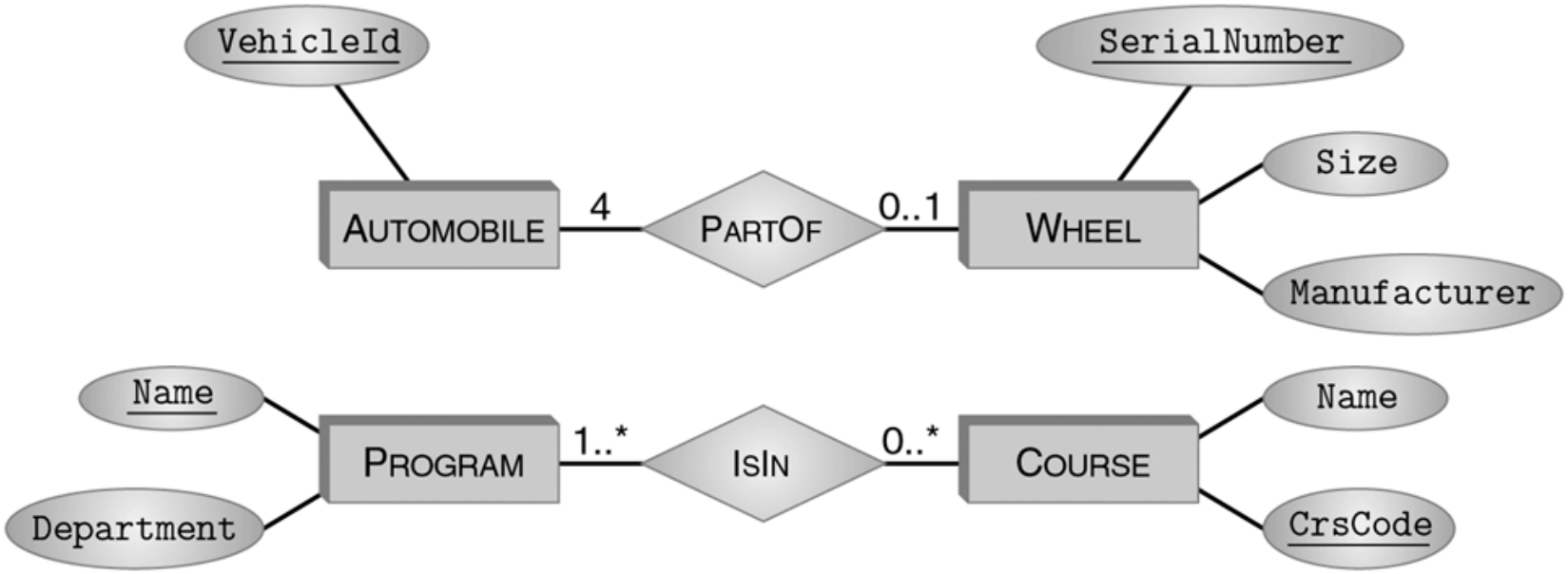
```
CREATE TABLE Store(
manId INTEGER PRIMARY KEY NOT NULL
REFERENCES Person(pid),
Since DATE
...);
```



Alternative cardinality notation



Example

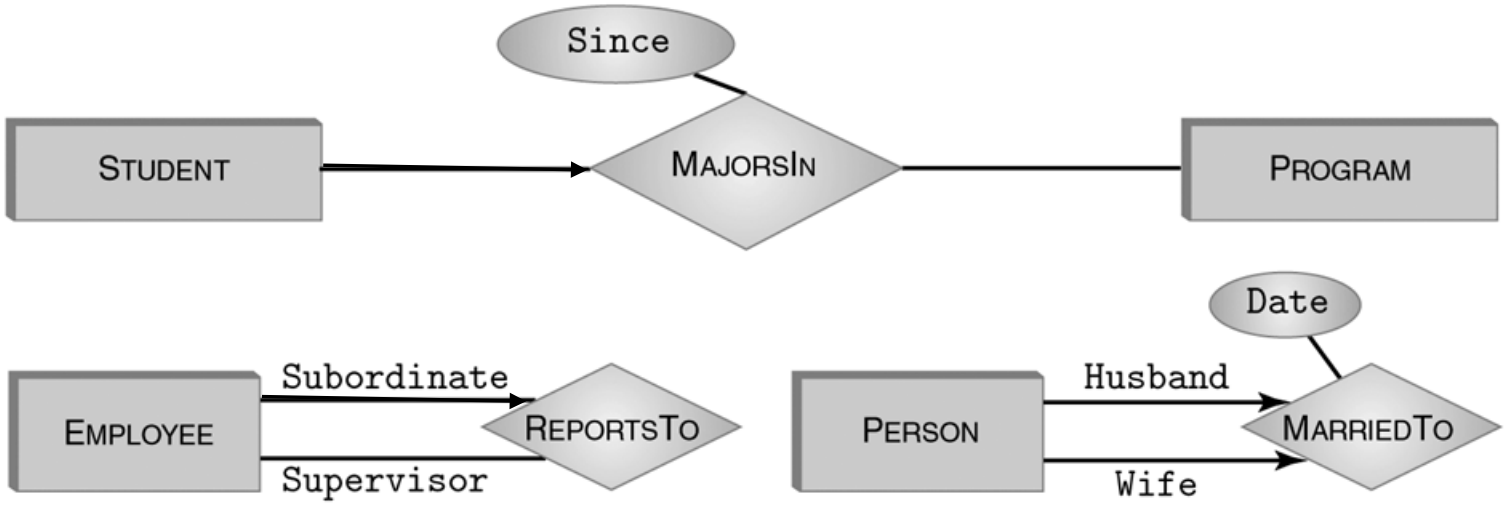


NB! Roles reversed compared to UML…

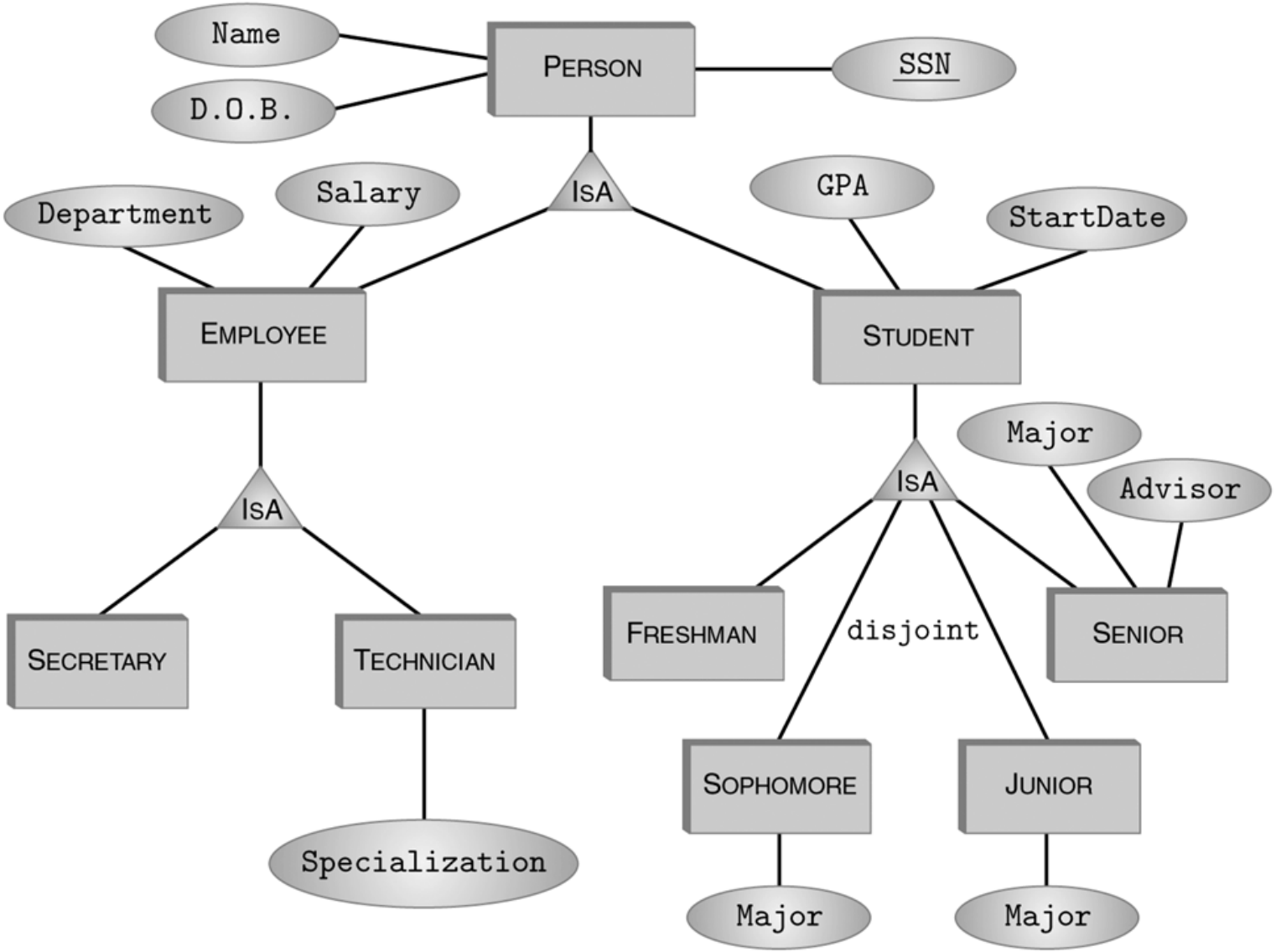
Many-one relationships

A very common special case:
- One entity type participates **0 or 1** times in a relationship.
- Then a relation is not needed to represent the relationship.

Examples:
STUDENT can store Since and primary key of PROGRAM
EMPLOYEE can store primary key for the Supervisor
PERSON can store Date and SSN for Husband or Wife, resp.

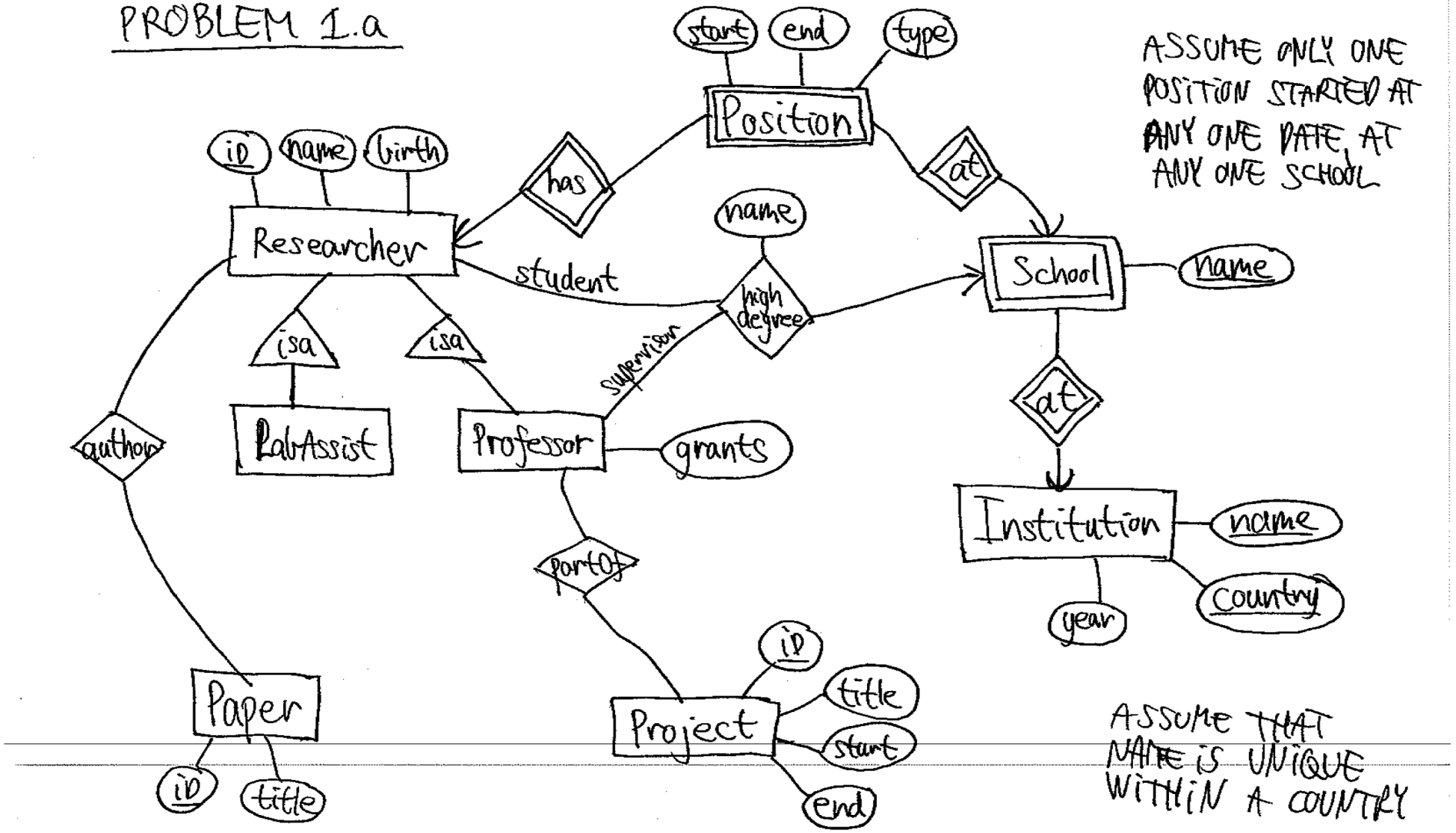


Type hierarchy (IsA)



Case study

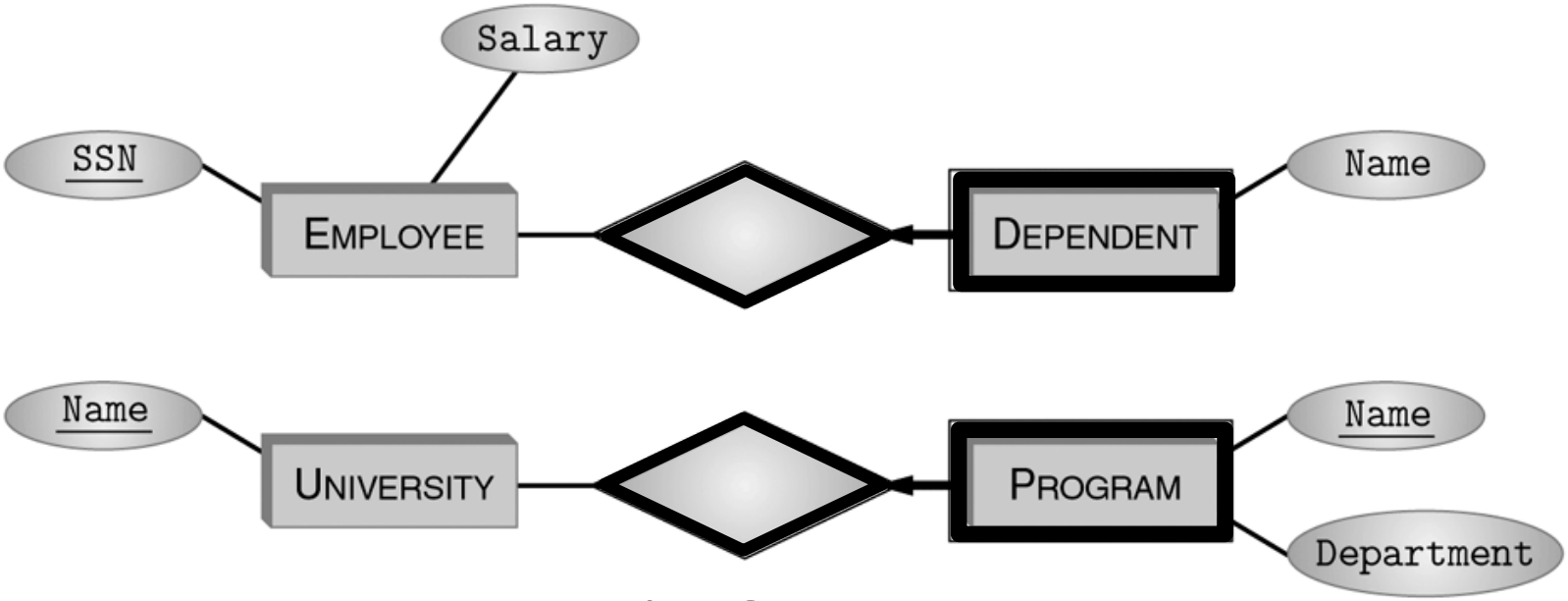
- For each researcher, his/her name, year of birth, and current position (if any).
- For each institution, its name, country, and inauguration year.
- For each institution, the names of its schools (*e.g. School of Law, School of Business, School of Computer Science,...*). A school belongs to exactly one institution.
- An employment history, including information on all employments (start and end date, position, and what school).
- Information about co-authorships, i.e., which researchers have co-authered a research paper. The titles of common research papers should also be stored.
- For each researcher, information on his/her highest degree (BSc, MSc or PhD), including who was the main supervisor, and at what school.
- For each professor, information on what research projects (title, start date, and end date) he/she is involved in, and the total amount of grant money for which he/she was the main applicant.


PROBLEM 1.a
start
end
type
Position
ASSUME ONLY ONE POSITION STARTED AT ANY ONE DATE, AT ANY ONE SCHOOL
ID
name
birth
has
at
Researcher
name
School
name
student
high degree
isa
isa
supervision
author
RAbAssist
Professor
grants
at
Institution
name
country
year
partOf
ID
title
start
end
Project
Paper
ID
title
ASSUME THAT NAME IS UNIQUE WITHIN A COUNTRY

Foreign keys in E-R diagrams

- Don’t put them there!
- Instead, foreign keys should be modeled through a relationship.
- Problem:
  What if a foreign key is (part of) a key?
  (E.g. school, institution in case study.)

Answer: Weak entity types

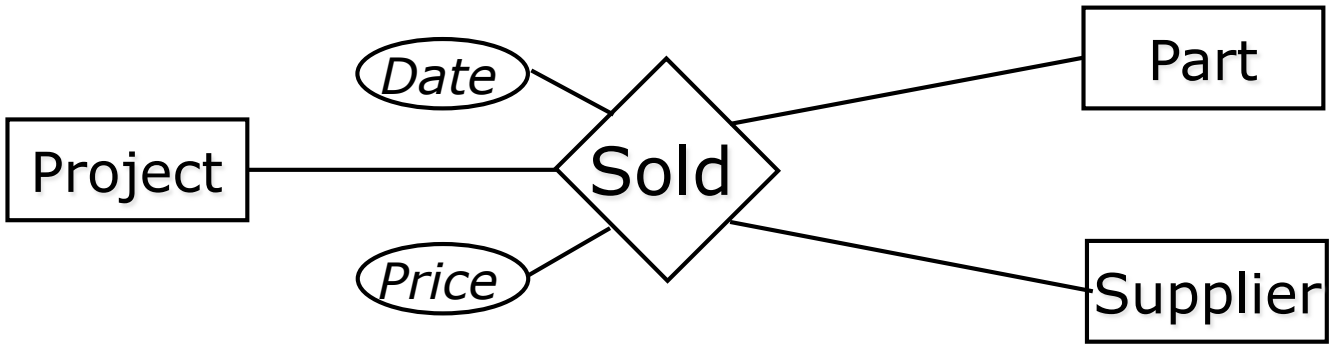


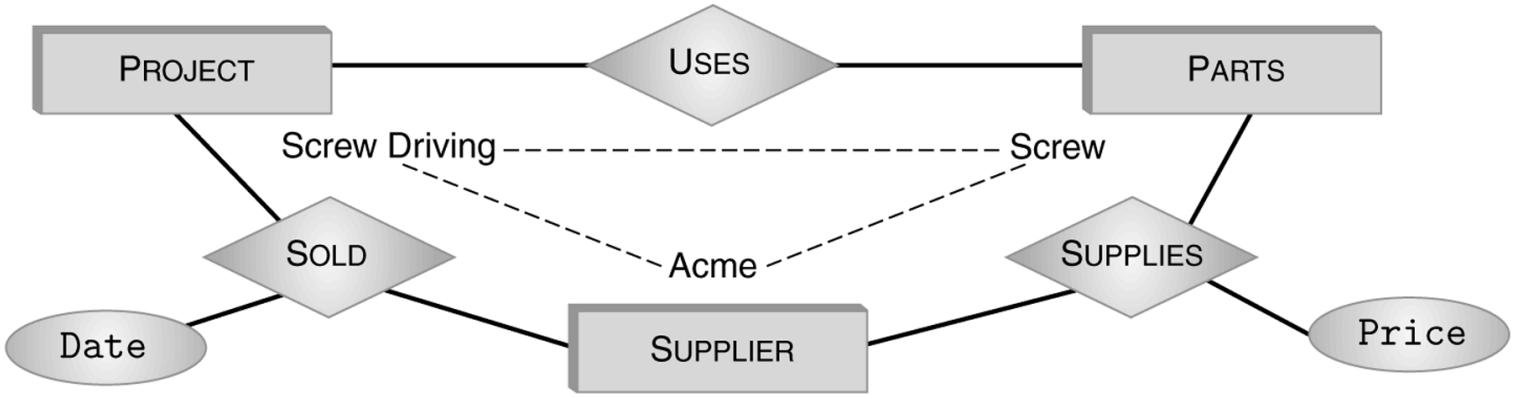
Corresponding relation


```
CREATE TABLE Program(
Name VARCHAR(50),
uName VARCHAR(20) REFERENCES WorksIn(sid),
…
PRIMARY KEY (name,uName));
```

Problem session

- What is the difference between the relational models corresponding to:





Translating m:m relationships

- The relationship becomes a relation
- Attributes:
  - Attributes of the relationship
  - The primary keys of the roles involved in the relationship (declare NOT NULL)
- Primary key (composite):
  - Any key attributes of the relationship
  - The primary keys from the roles in the key
- Foreign key references:
  - To primary keys of the roles

From IsA to relations

Can be done in different ways. Examples:

1. Treat like any other relationship.

2. All relations use the *same* primary key
- the key of the top entity type

3. A single relation for the whole hierarchy with attribute(s) defining the entity type(s)
- Binary (indicate all memberships), or
- Identifier for the most specific entity type.

Other issues discussed in RG

For self-study:

- Relationships that involves a relationship: Aggregation.
- Different types of IsA: Covering and overlap constraints

Tools

- Tools like Visio and MySQL Workbench support modeling using E-R notation
  - Some support for automatic translation to relation schemas.
- You may choose to use them in your coursework.
- Ideally, should also support ”reverse engineering” E-R model from relation schema.
  - Problem: This mapping is not unique.

Hand-in, part 1

- Available on course web page.
  - Modeling a database on movies, actors, etc.
  - We have covered the material to do most already. (The rest, normalization, next week.)
  - Due in two weeks, September 21, midnight.
  - Upload to LearnIT, one copy per group, with a file name containing your group number (to be assigned).

Group formation

- LearnIT has a link to a Google Forms page where you must register for a group.
  - All of you **must** register to indicate commitment to do hand-ins 1, 2, and 4 in a group.
    Please do this today!
  - Self-formed groups of size 5 will be untouchable.
  - Smaller self-formed (e.g. singleton) groups will be merged to form groups of size 4-6.
  - If you wish to work on the project outside of normal working hours, let us know and we will assign you group members with the same wish.

Next steps…

- Exercises from 10.00 as usual.
- Highly recommended:
  - Do the "typical ER modeling mistakes" exercise (and revisit before the exam).
  - Compare ER models made with another student and discuss differences:
    - Are our data models equivalent, or do they make different assumptions about the data?
    - Do they correspond to the case description?