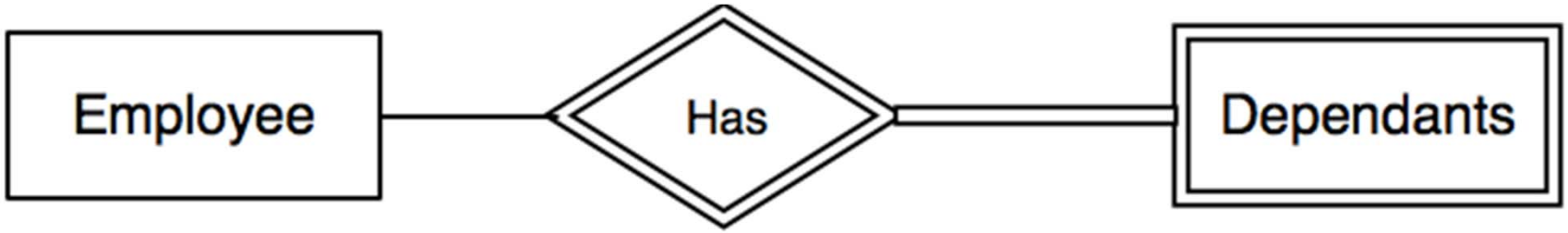
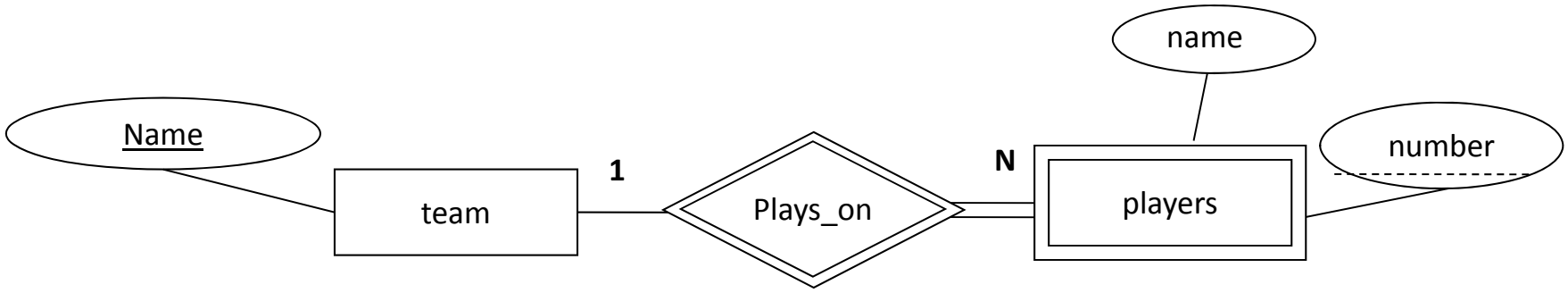
Weak Entity Types

- Do not have key attributes of their own
  - Identified by being related to specific entities from another entity type
- **Identifying relationship**
  - Relates a weak entity type to its owner
- Always has a total participation constraint

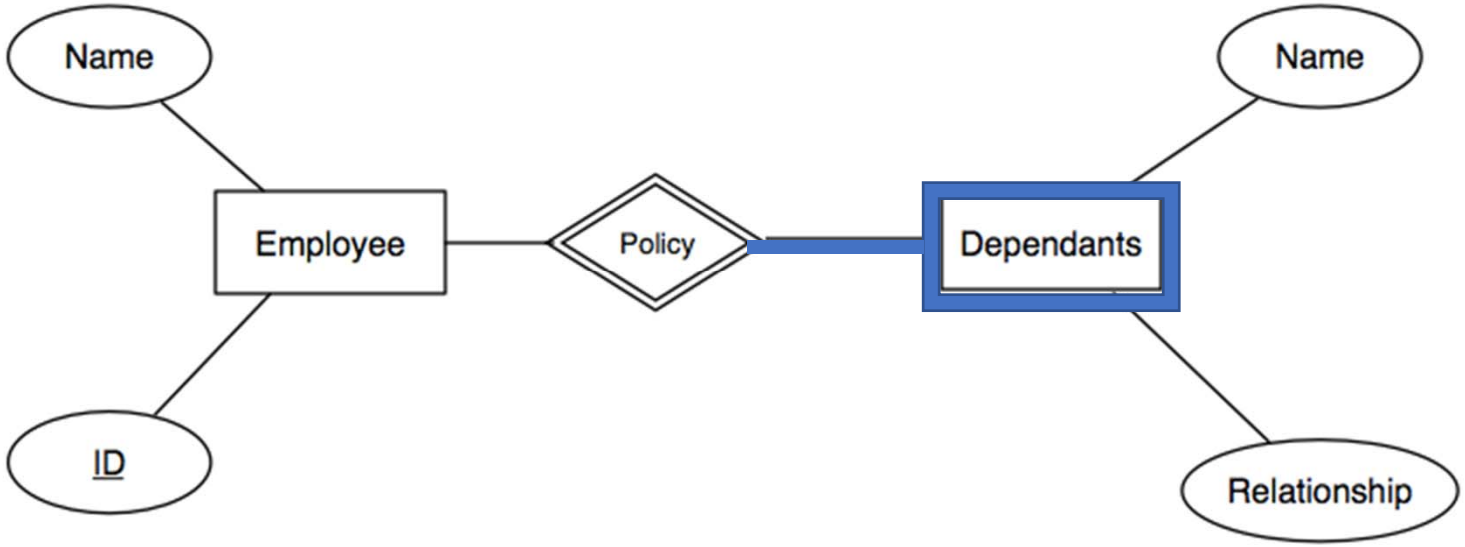


Constraints on Relationship Types

- Weak entity types: They do not have key attibutes of their own.
- A weak entity can be identified uniquely by being related to another entity (together with its own attributes).



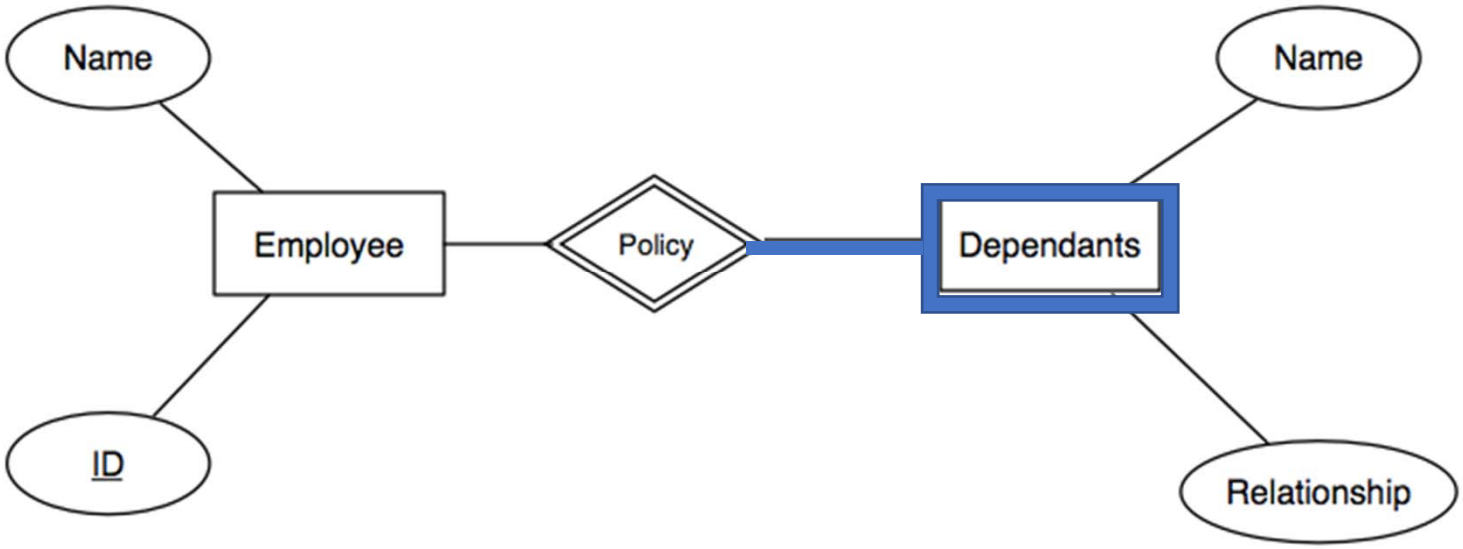
Weak Entity Sets



- If an entity set is weak, it will be shown as a rectangle with a double/thick border.
- Its supporting many-one relationships will be shown as diamonds with a double border.
- If an entity set supplies any attributes for its own key, then those attributes will be underlined.

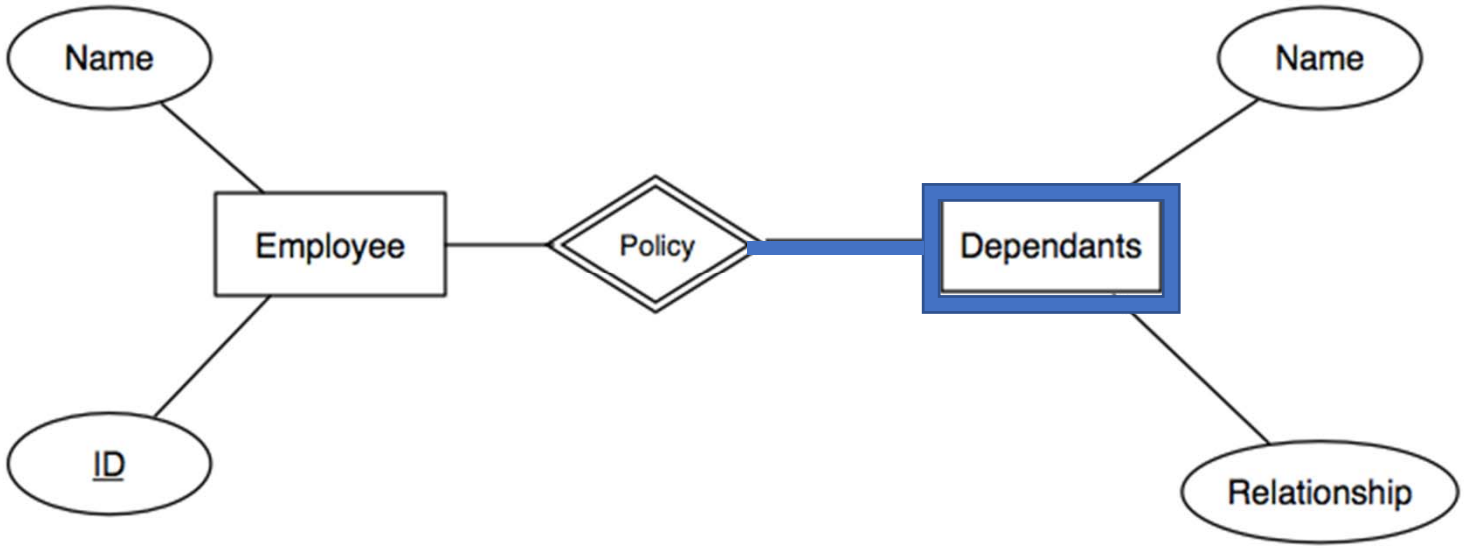
Requirements for Weak Entity Sets

- if E is a weak entity set, then its key consists of:
  - Zero or more of its own attributes, and
  - Key attributes from entity sets that are reached by certain many-one relationships from *E* to other entity sets. These many-one relationships are called supporting relationships for *E*.



Requirements for Weak Entity Sets

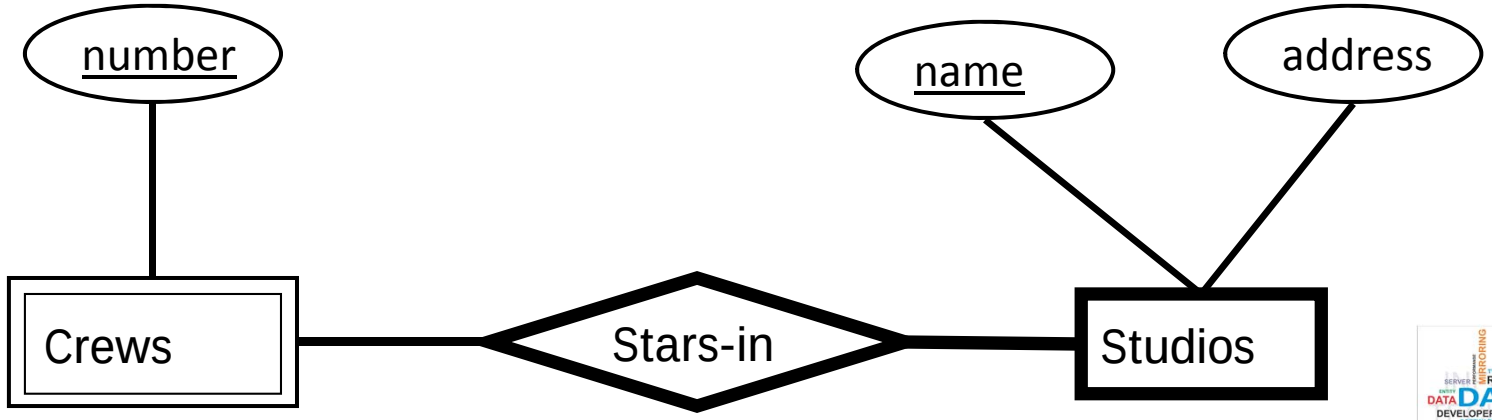
- In order for *R*, a many-one relationship from *E* to some entity set *F*, to be a supporting relationship for *E*, the following conditions must be obeyed:
  - *R* must be a binary, many-one relationship from *E* to *F*.
  - *R* must have referential integrity from *E* to *F*.
  - The attributes that *F* supplies for the key of *E* must be key attributes of *F*.
- Multiple supporting relationships are possible



Weak Entity Sets Notation

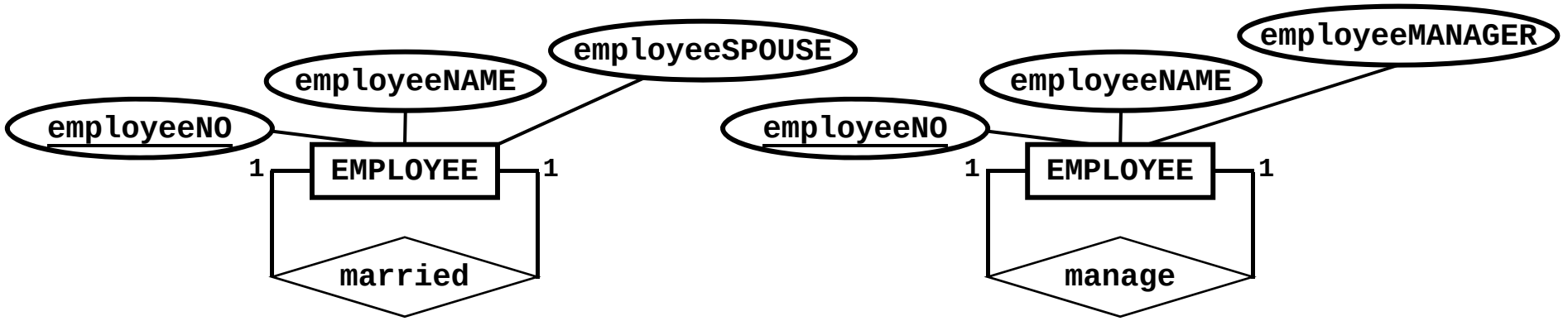
1. If an entity set is weak, it will be shown as a rectangle with a double border
2. Its supporting many-one relationship will be shown as diamonds with a double border
3. If an entity set supplies any attributes for its own key, then those attributes will be underlined

- Whenever we use an entity set *E with a double border,* it is weak. The key for *E is whatever attributes of E* are underlined plus the key attributes of those entity sets to which *E is connected by many-one* relationships with a double border.



Recursive Entity/Relationship

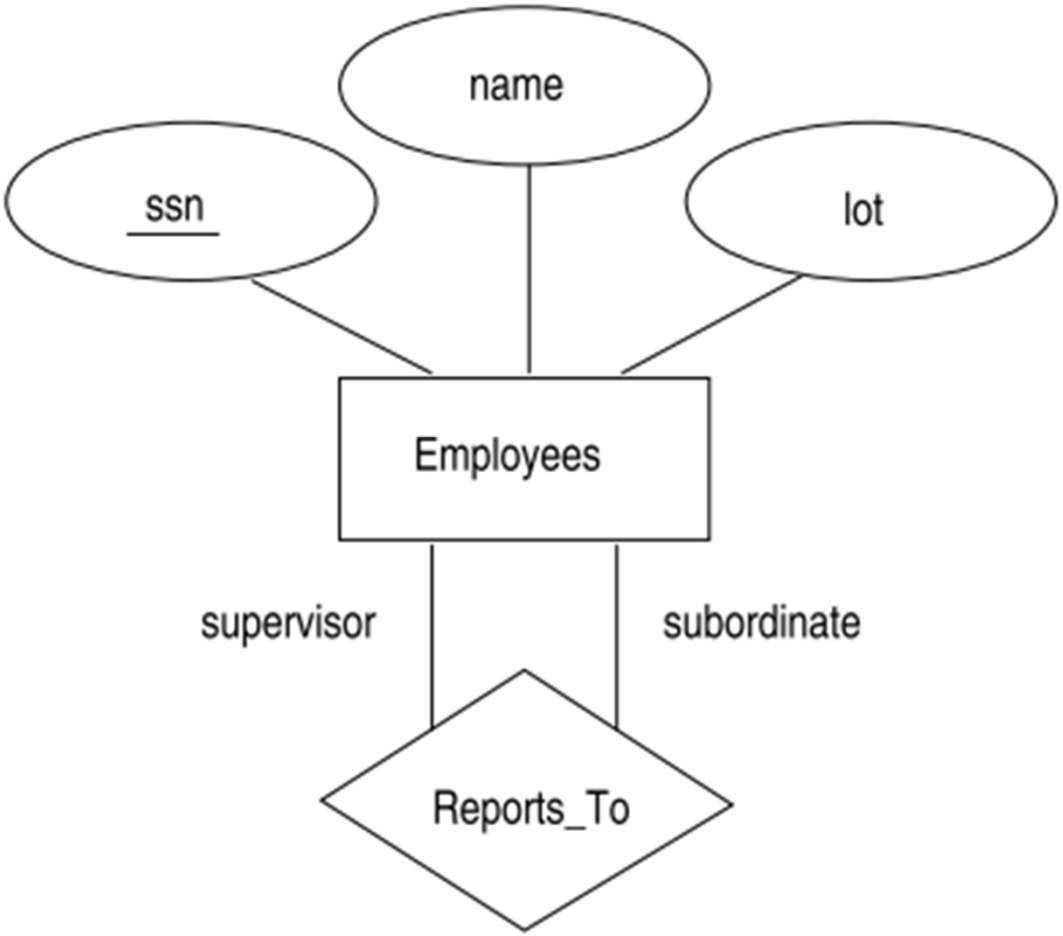
- Entity set that have relationship with the same entity set
- Example: **EMPLOYEE** entity



employeeNO	employeeNAME	employeeSPOUSE
111	Ali	444
222	Ah Chong	
333	Bazil	
444	Sheriz	111

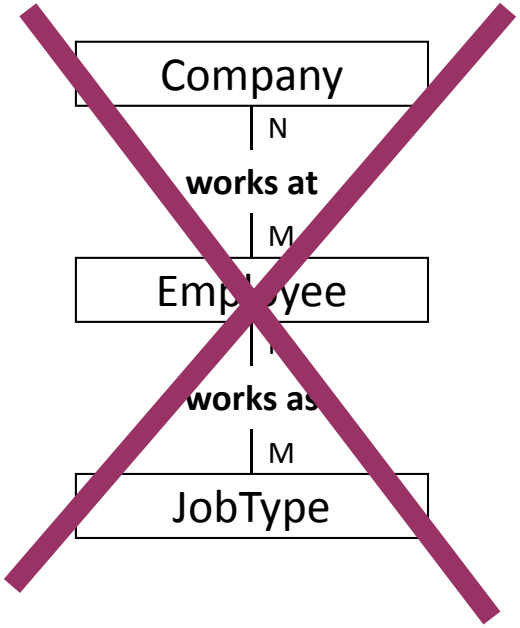
employeeNO	employeeNAME	employeeMANAGER
111	Ali	333
222	Ah Chong	333
333	Bazil	333
444	Sheriz	

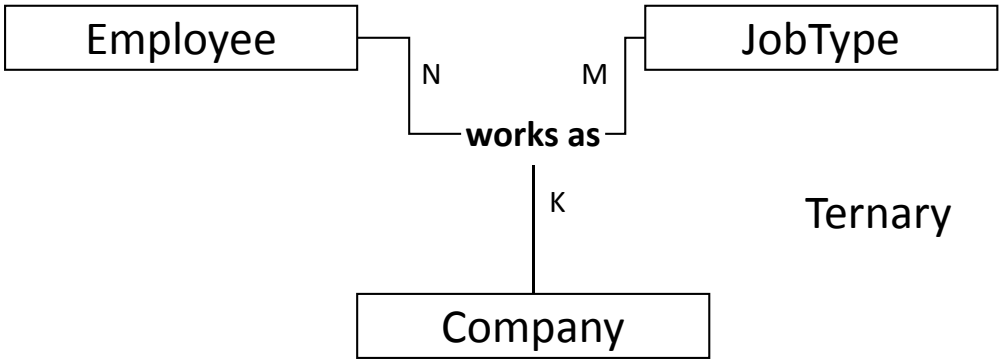
Recursive Entity/Relationship



N-ary relationships

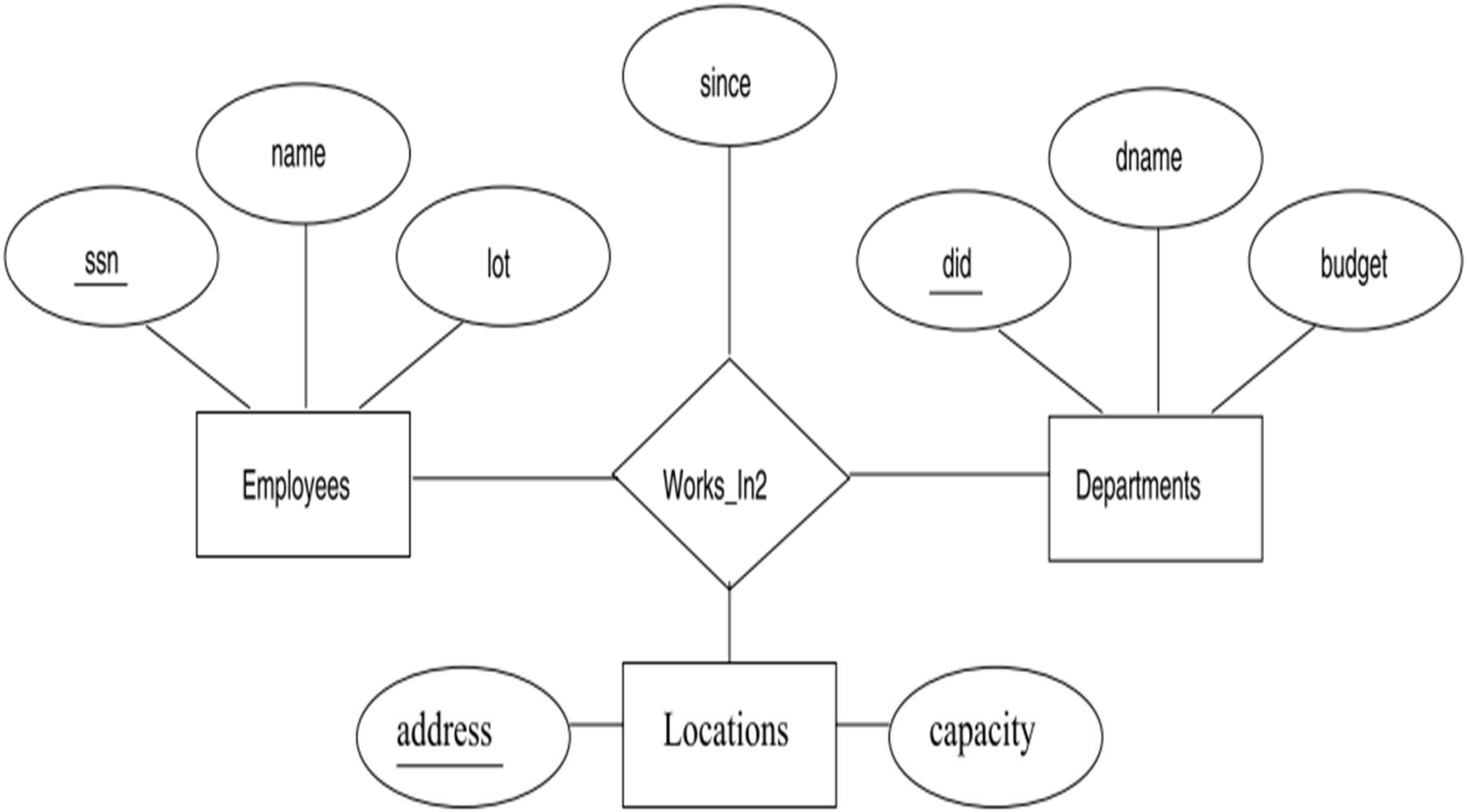
- Example. A person works as an engineer at one company and as a gym instructor at another company.



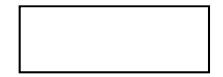
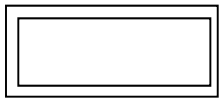


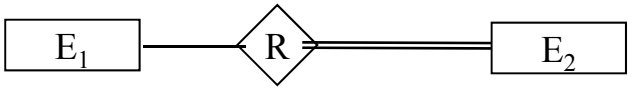
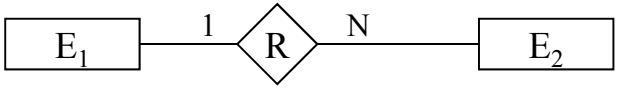
Ternary

Ternary Relationship

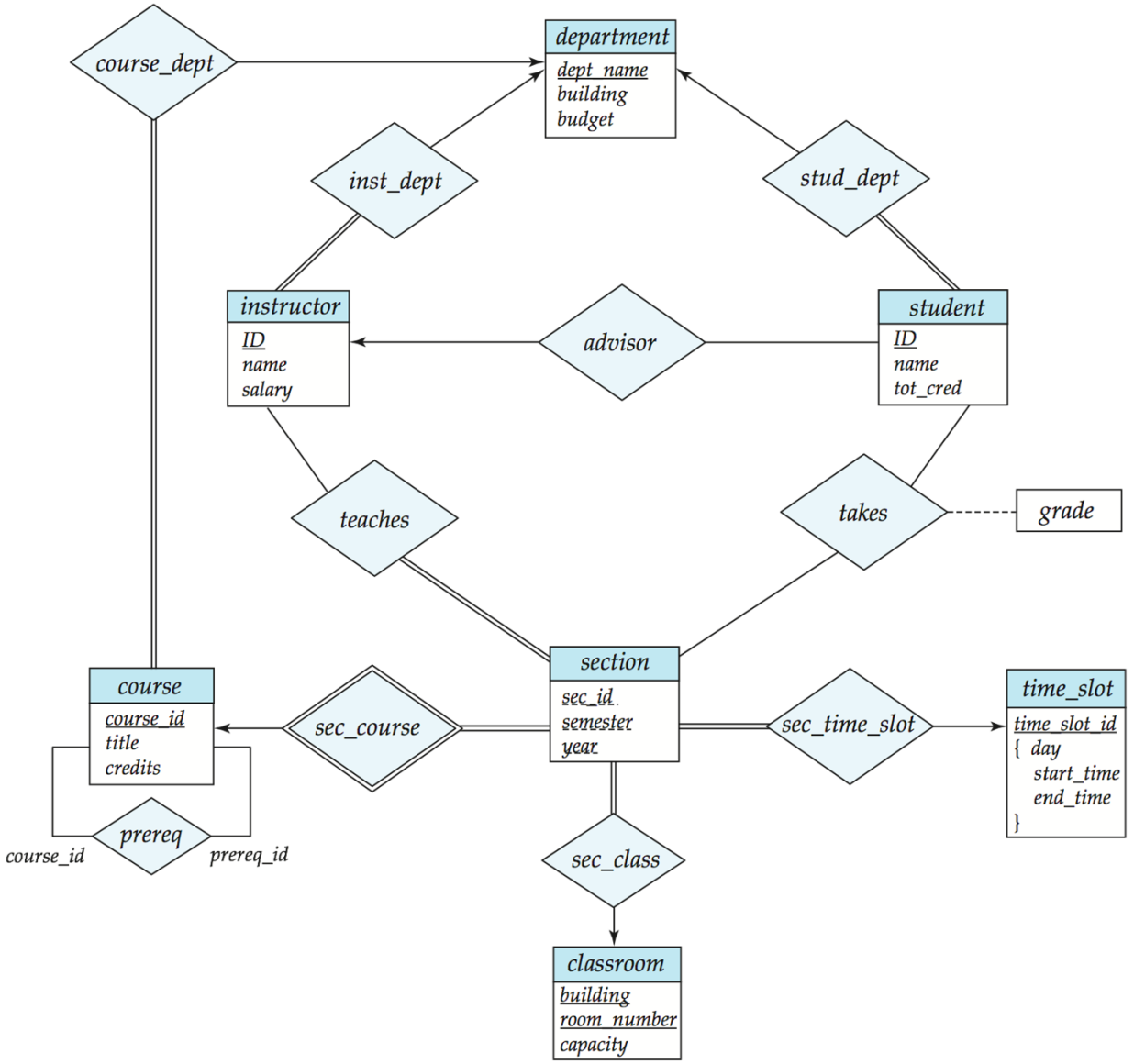


ER Notation

Symbol	Meaning
	ENTITY TYPE
	WEAK ENTITY TYPE
	RELATIONSHIP TYPE
	IDENTIFYING RELATIONSHIP TYPE
	ATTRIBUTE
	KEY ATTRIBUTE
	MULTIVALUED ATTRIBUTE

	TOTAL PARTICIPATION OF E_2 IN R
	CARDINALITY RATIO 1:N FOR E_1:E_2 IN R

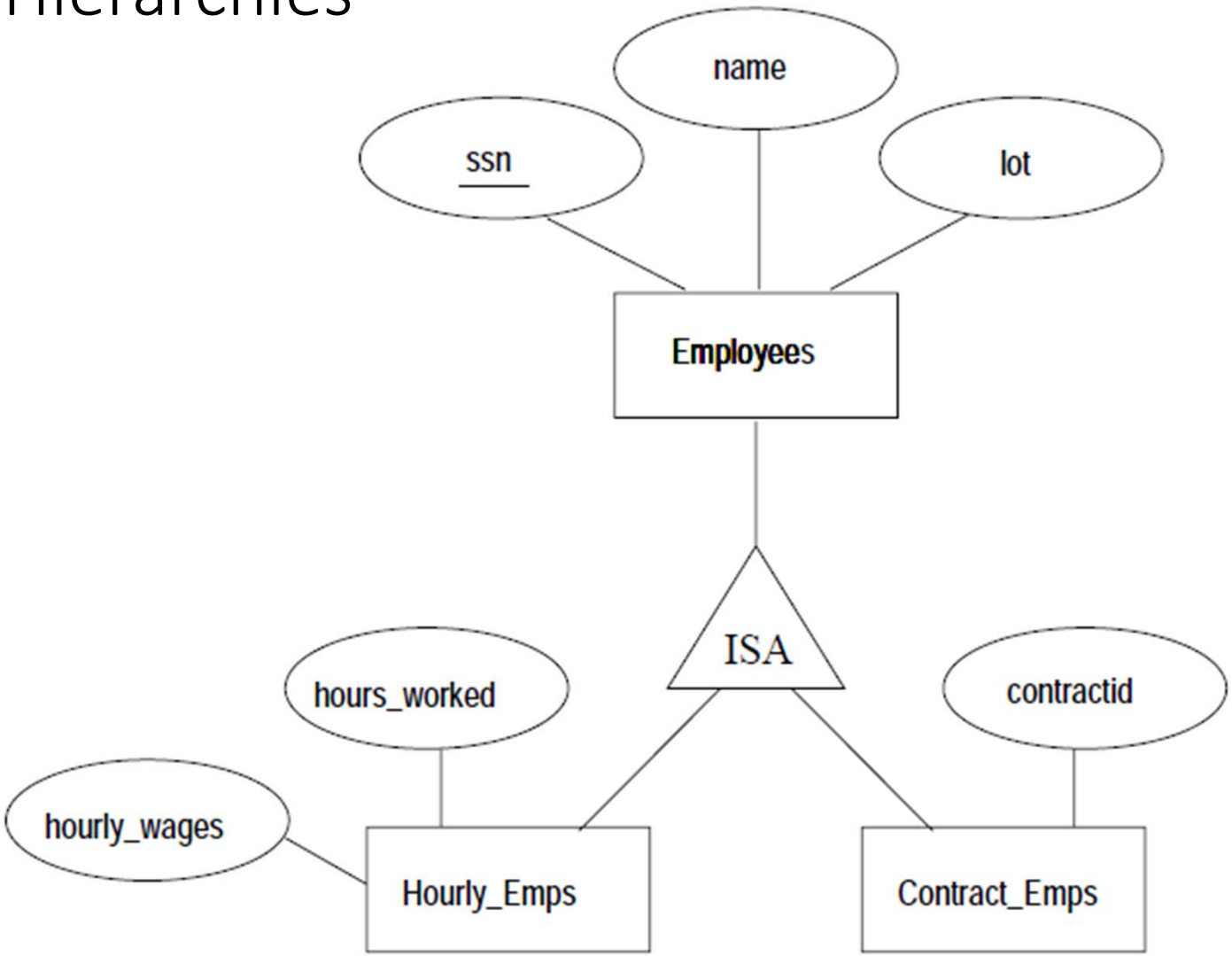
E-R Diagram for a University Enterprise

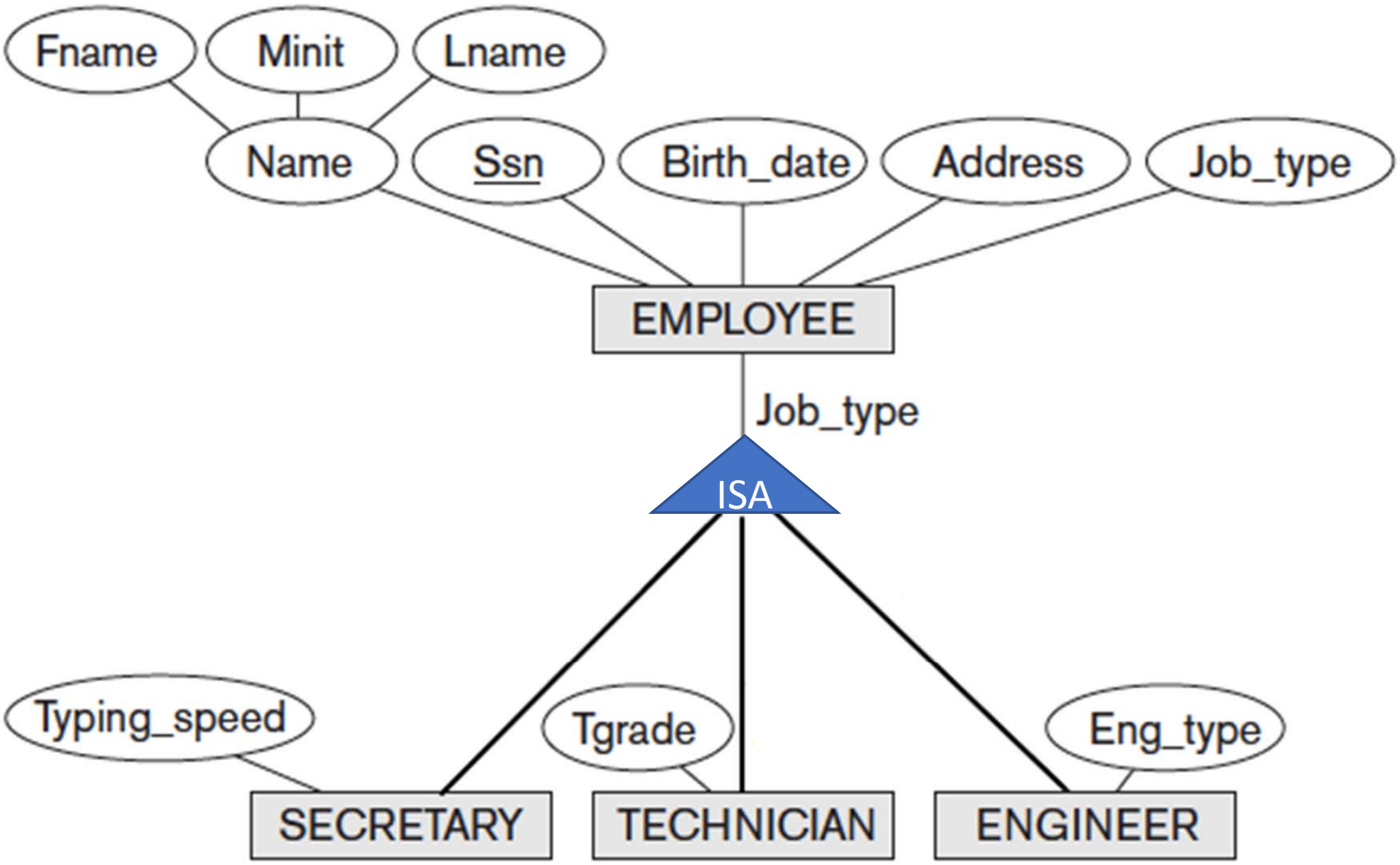


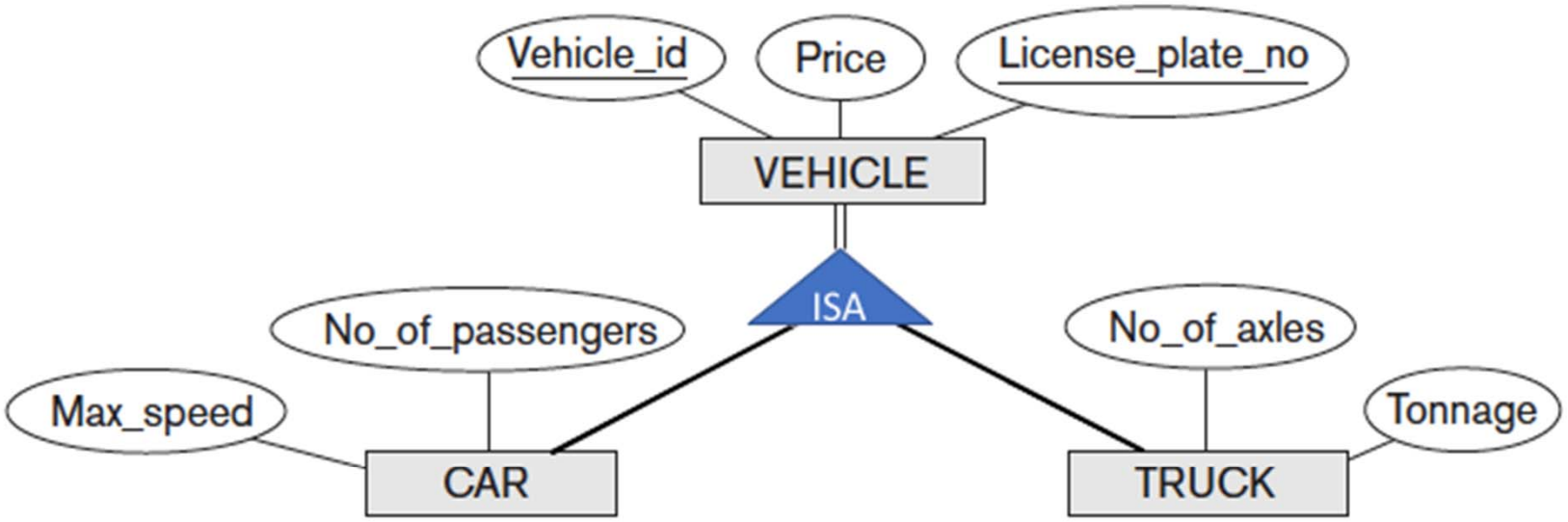
Enhanced ER (EER) model

- Why more? To comply with more complex data requirements.
  - Example. Only some employees can use a company car, only managers have to write a monthly report, but all employees have assigned personal number, salary account and a place in the office.
- Class Hierarchies
  - Subclass/superclass,
  - specialization/generalization,
  - union/category, and
  - attribute and relationship inheritance.

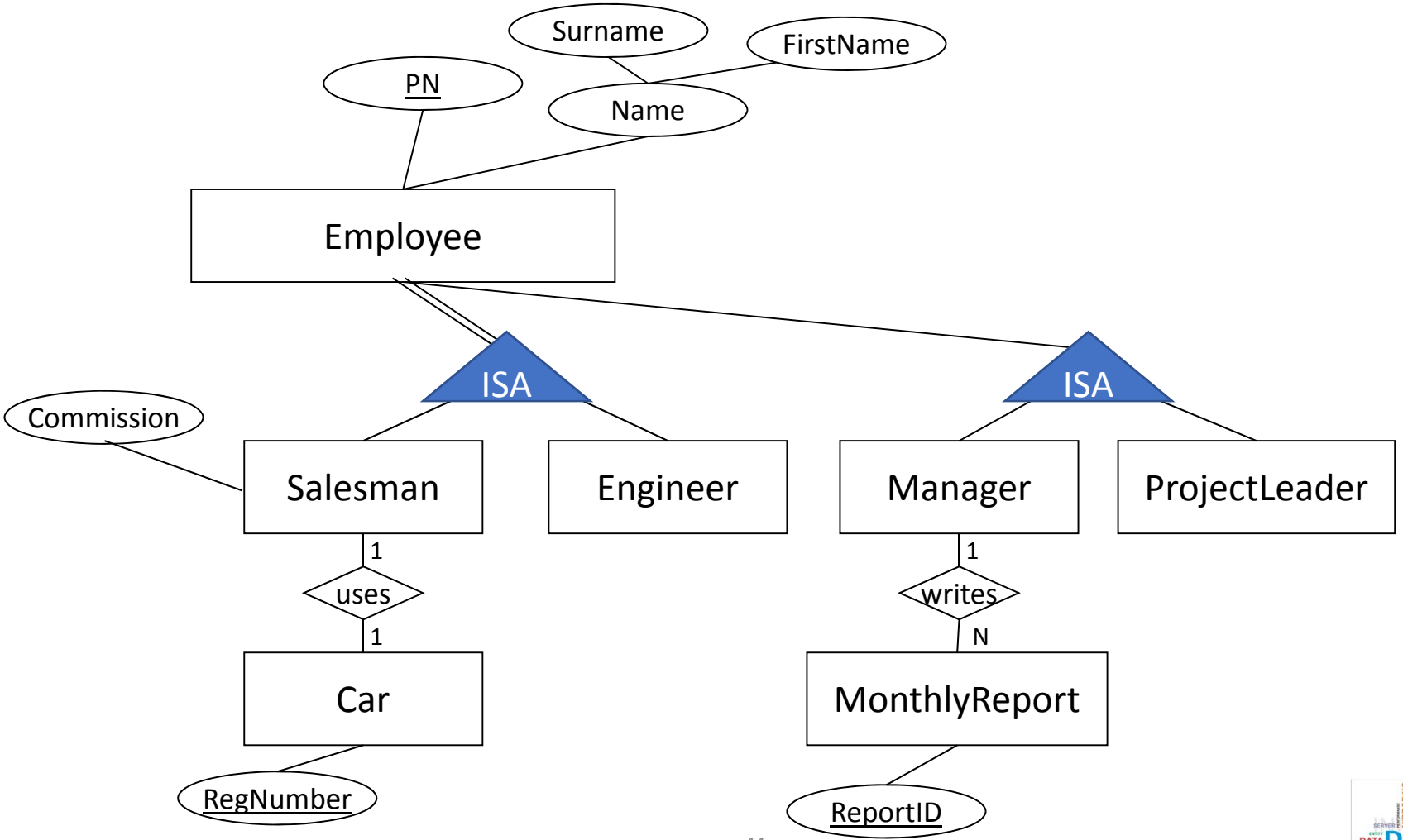
Class Hierarchies




Fname
Minit
Lname
Name
Ssn
Birth_date
Address
Job_type
EMPLOYEE
Job_type
ISA
Typing_speed
Tgrade
Eng_type
SECRETARY
TECHNICIAN
ENGINEER

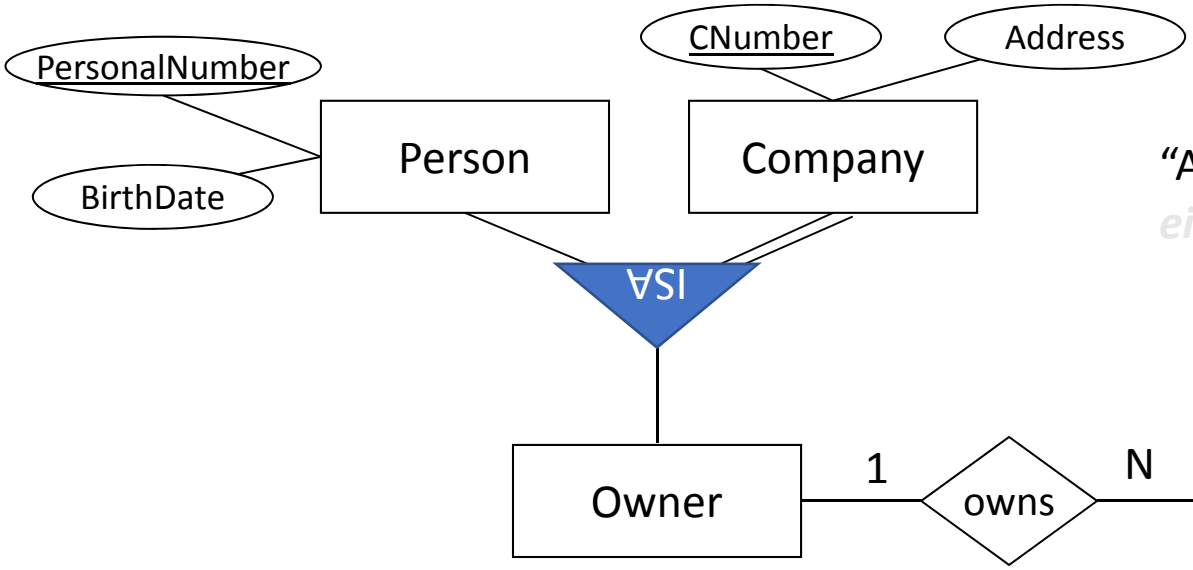

Vehicle_id
Price
License_plate_no
VEHICLE
ISA
No_of_passengers
Max_speed
CAR
No_of_axles
Tonnage
TRUCK

Subclass/superclass



Class Hierarchies

- A UNION subclass represents a collection of entities that is a subset of the UNION of the entities of the super classes.



"An owner of a car is *either* a person *or* a company."

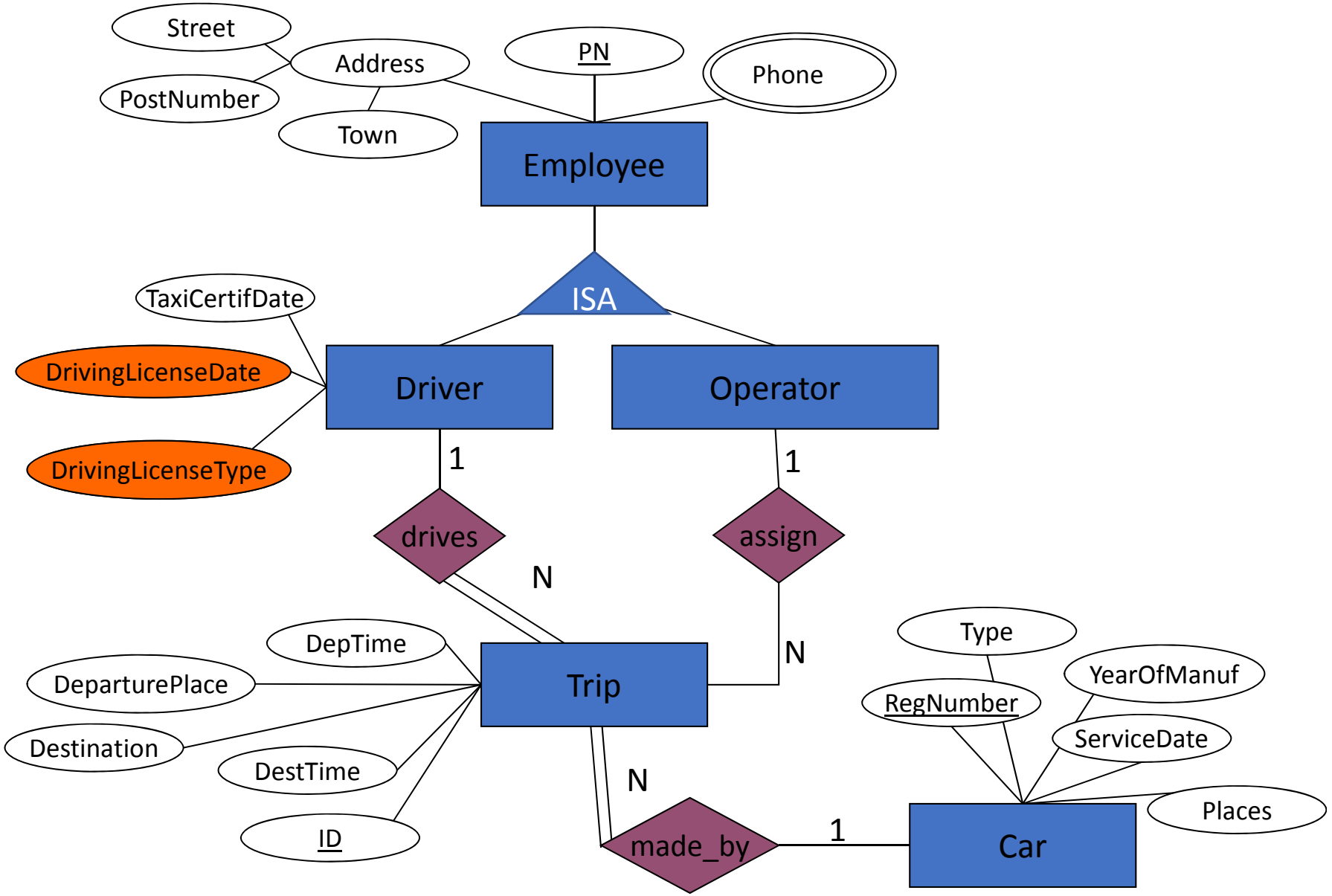
Examples

A taxi company needs to model their activities.

There are two types of employees in the company: drivers and operators. For drivers it is interesting to know the date of issue and type of the driving license, and the date of issue of the taxi driver's certificate. For all employees it is interesting to know their personal number, address and the available phone numbers.

The company owns a number of cars. For each car there is a need to know its type, year of manufacturing, number of places in the car and date of the last service.

The company wants to have a record of car trips (körningar). A taxi may be picked on a street or ordered through an operator who assigns the order to a certain driver and a car. Departure and destination addresses together with times should also be recorded.


Street
Address
PostNumber
Town
PN
Phone
Employee
ISA
TaxiCertifDate
DrivingLicenseDate
DrivingLicenseType
Driver
Operator
1
1
drives
assign
N
N
DepTime
DeparturePlace
Destination
DestTime
ID
Trip
N
made_by
1
Type
RegNumber
YearOfManuf
ServiceDate
Places
Car

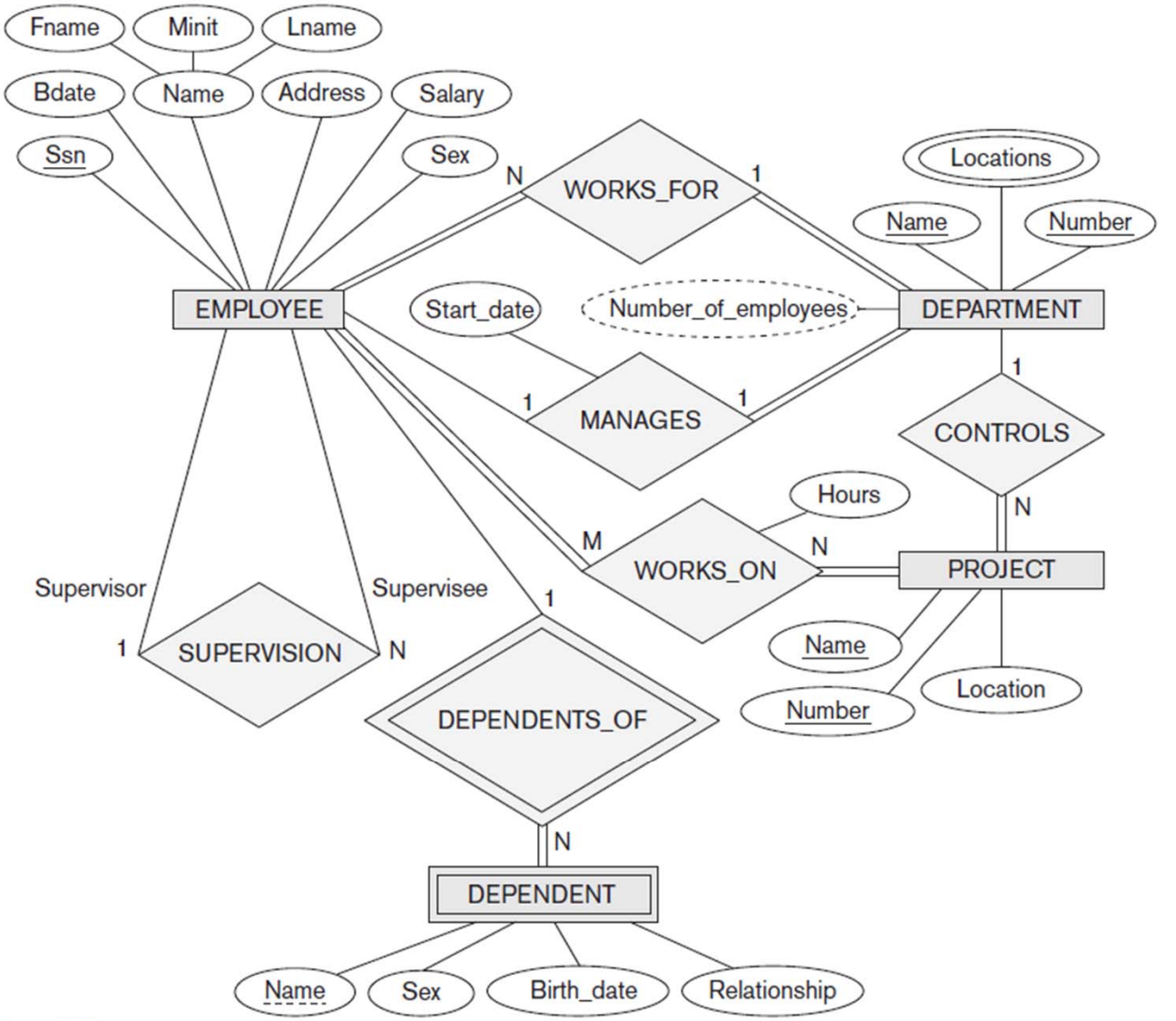
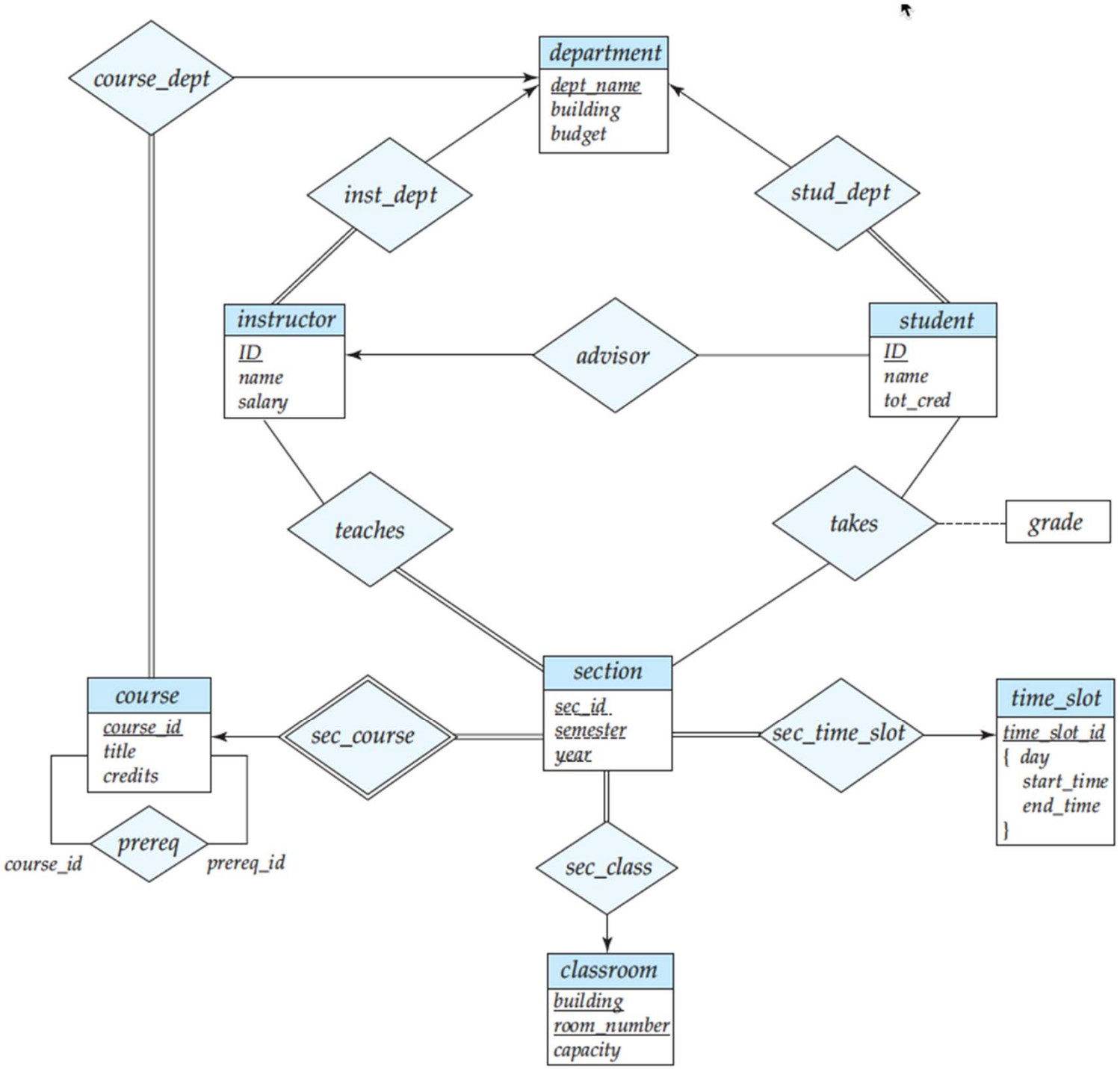
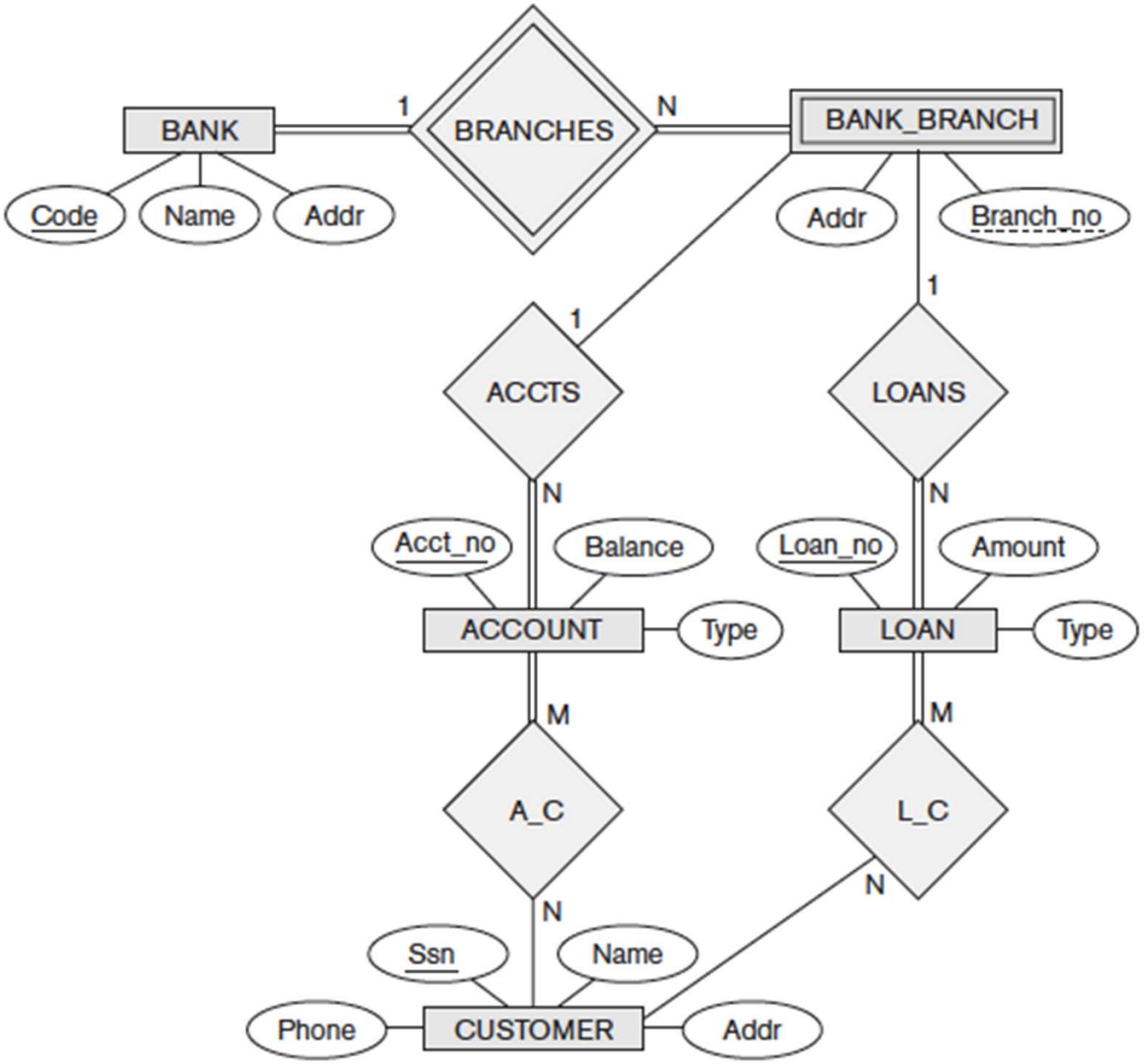
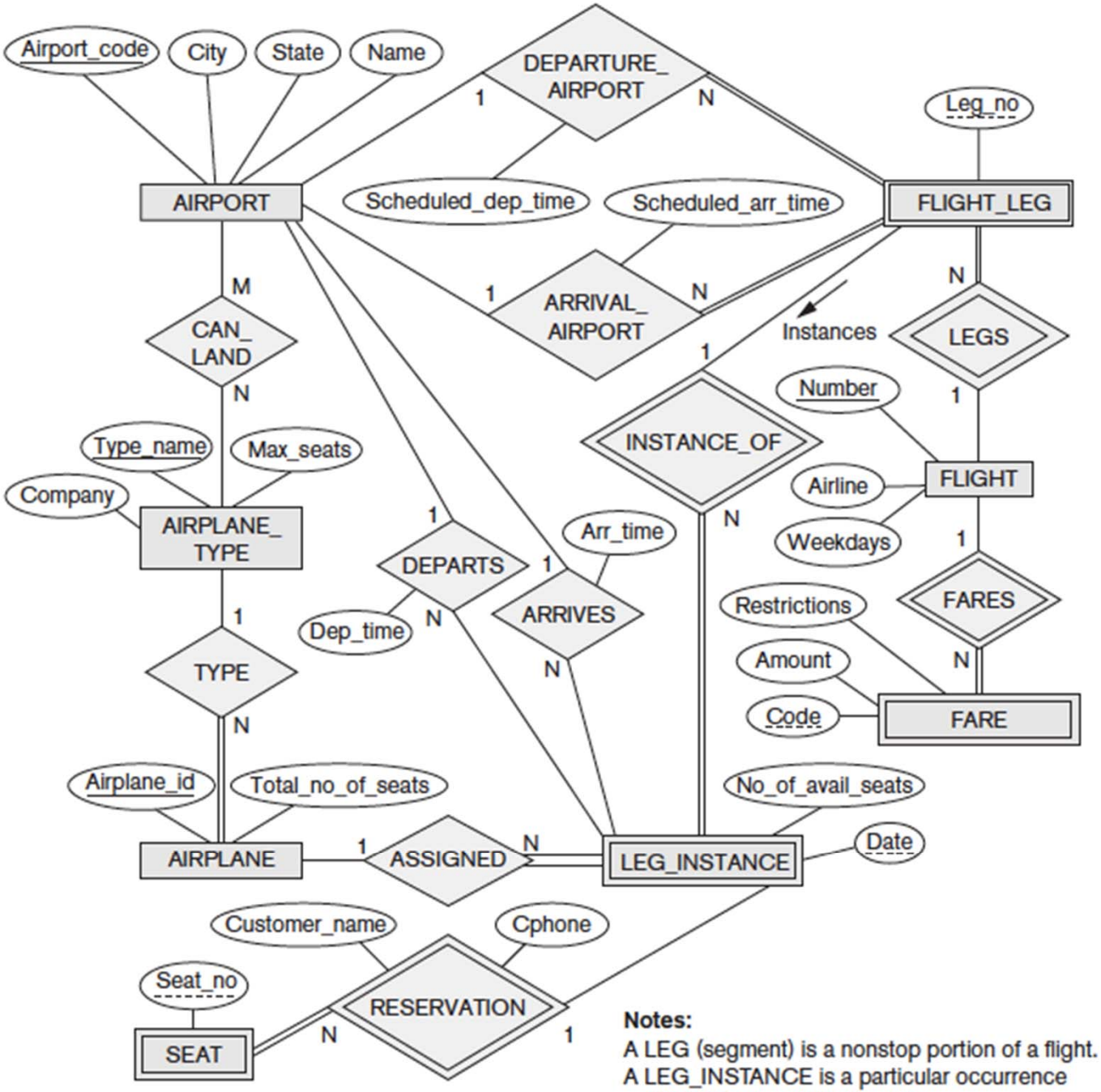


Figure 7.2
An ER schema diagram for the COMPANY database. The diagrammatic notation is introduced gradually throughout this chapter and is summarized in Figure 7.14.


department
dept_name
building
budget
course_dept
inst_dept
stud_dept
instructor
ID
name
salary
advisor
student
ID
name
tot_cred
teaches
takes
grade
section
sec_id
semester
year
course
course_id
title
credits
sec_course
sec_time_slot
time_slot
time_slot_id
{ day
start_time
end_time
}
prereq
course_id
prereq_id
sec_class
classroom
building
room_number
capacity


BANK
1
BRANCHES
N
BANK_BRANCH
Code
Name
Addr
Addr
Branch_no
1
1
ACCTS
LOANS
N
N
Acct_no
Balance
Loan_no
Amount
ACCOUNT
Type
LOAN
Type
M
M
A_C
L_C
N
N
Ssn
Name
Phone
CUSTOMER
Addr



Notes:
A LEG (segment) is a nonstop portion of a flight.
A LEG_INSTANCE is a particular occurrence of a LEG on a particular date.

- Every airplane has a registration number, and each airplane is of a specific model.
- The airport accommodates a number of airplane models, and each model is identified by a model number (e.g., DC-10) and has a capacity and a weight.
- A number of technicians work at the airport. You need to store the name, SSN, address, phone number, and salary of each technician.
- Each technician is an expert on one or more plane model(s), and his or her expertise may overlap with that of other technicians. This information about technicians must also be recorded.
- Traffic controllers must have an annual medical examination. For each traffic controller, you must store the date of the most recent exam.
- All airport employees (including technicians) belong to a union. You must store the union membership number of each employee. You can assume that each employee is uniquely identified by a social security number.
- The airport has a number of tests that are used periodically to ensure that airplanes are still airworthy. Each test has a Federal Aviation Administration (FAA) test number, a name, and a maximum possible score.
- The FAA requires the airport to keep track of each time a given airplane is tested by a given technician using a given test. For each testing event, the information needed is the date, the number of hours the technician spent doing the test, and the score the airplane received on the test.

- Patients are identified by an SSN, and their names, addresses, and ages must be recorded.
- Doctors are identified by an SSN. For each doctor, the name, specialty, and years of experience must be recorded.
- Each pharmaceutical company is identified by name and has a phone number.
- For each drug, the trade name and formula must be recorded. Each drug is sold by a given pharmaceutical company, and the trade name identifies a drug uniquely from among the products of that company. If a pharmaceutical company is deleted, you need not keep track of its products any longer.
- Each pharmacy has a name, address, and phone number.
- Every patient has a primary physician. Every doctor has at least one patient.
- Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another.
- Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors. Each prescription has a date and a quantity associated with it. You can assume that, if a doctor prescribes the same drug for the same patient more than once, only the last such prescription needs to be stored.
- Pharmaceutical companies have long-term contracts with pharmacies. A pharmaceutical company can contract with several pharmacies, and a pharmacy can contract with several pharmaceutical companies. For each contract, you have to store a start date, an end date, and the text of the contract.
- Pharmacies appoint a supervisor for each contract. There must always be a supervisor for each contract, but the contract supervisor can change over the lifetime of the contract.