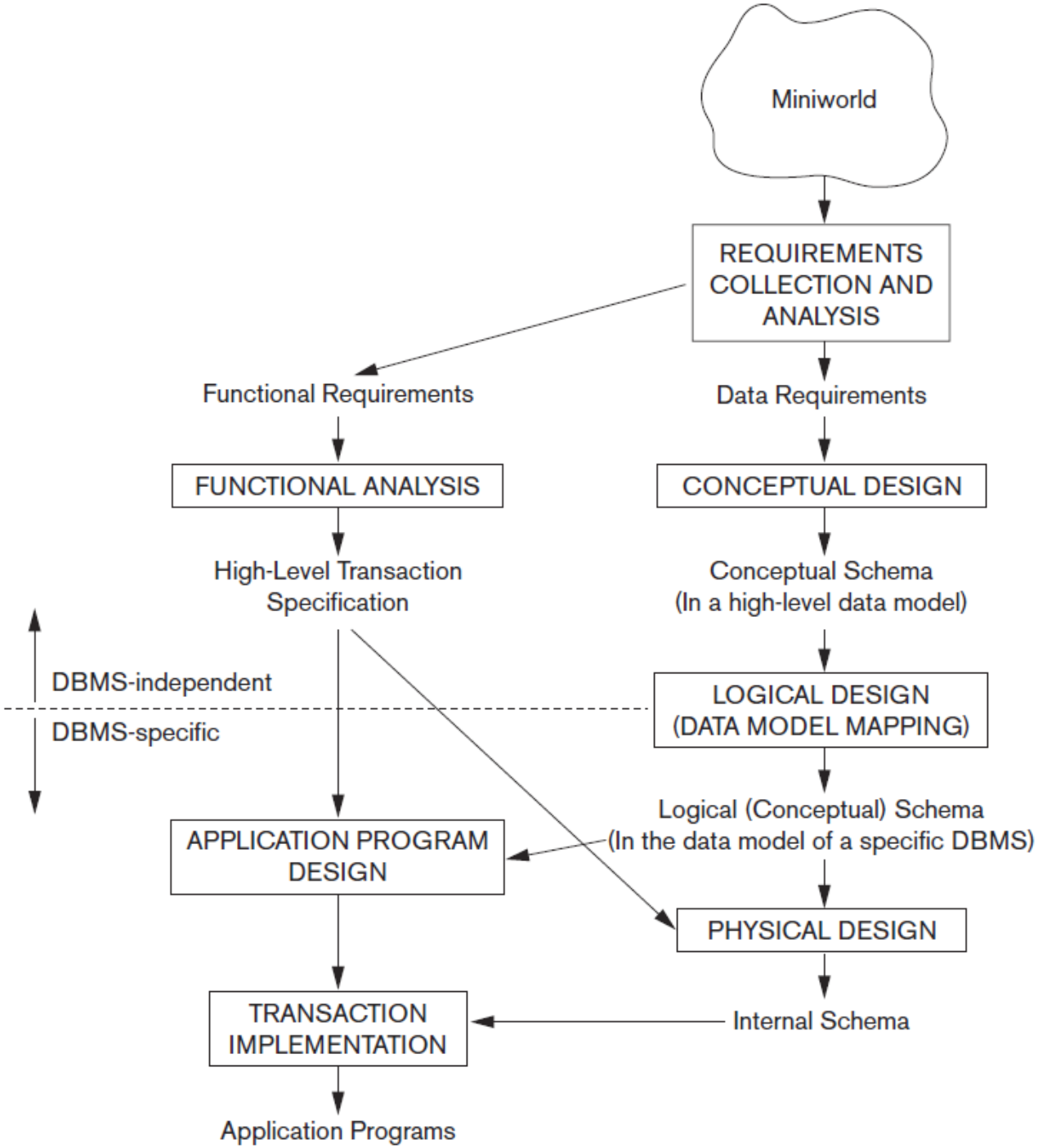
Domain model vs. ER Diagram

→ ISO/IEC/IEEE 24765 : Systems and software engineering – defines the vocabulary as:
  → **domain model** "a product of domain analysis that provides a representation of the requirements of the domain."
  → **entity-relationship diagram** "a diagram that depicts a set of real-world entities and the logical relationships among them.“
→ But a domain model can evolve into an ER diagram.

The process



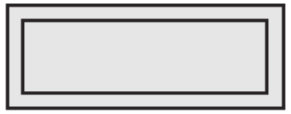
→ DBMS independent works with entities and concepts.
→ DBMS specific works with SQL and Tables.
→ The mapping in the middle is the process of working from the ER diagrams to the table mapping.

In relation to UP

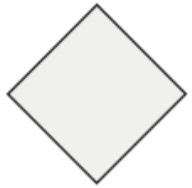
Phase	Artifact
Business Modelling	Domain Model (None from DM)
Requirements	(None from DM)
Analysis	ER, EER, UML (Entities only)
Design	UML (Tables)
Implementation	SQL Creation Script
Test	(None from DM)
Deployment	(None from DM)

ER Components 1/3

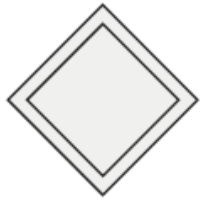
Entity



Weak Entity



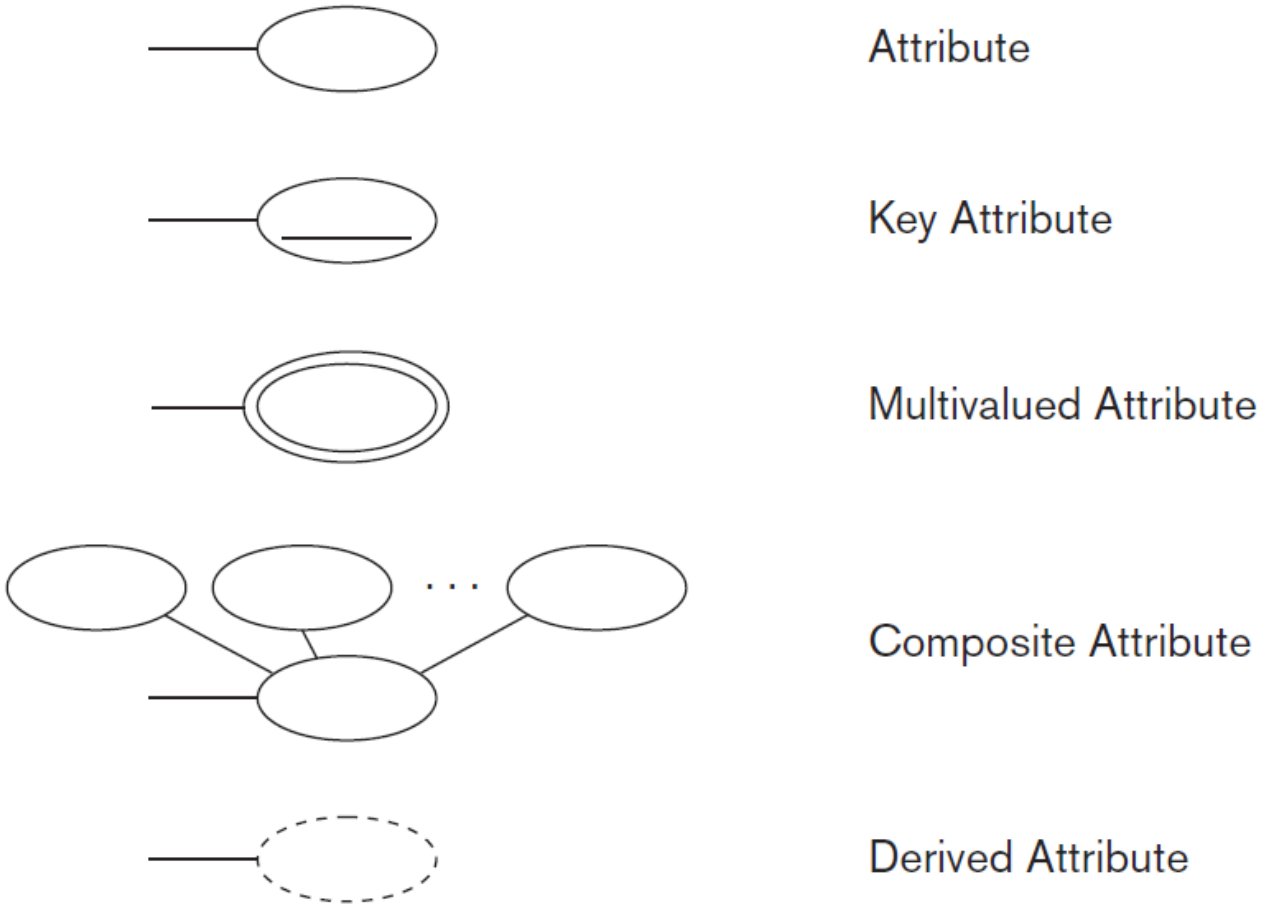
Relationship



Indentifying Relationship

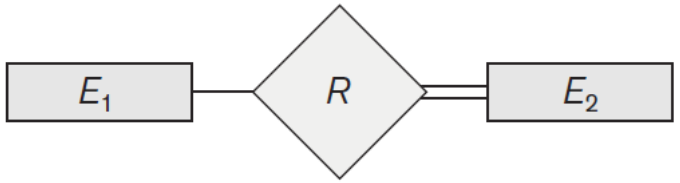
→ An entity type is strong if its existence does not depend on another entity type. Otherwise, the entity type is weak.

ER Components 2/3

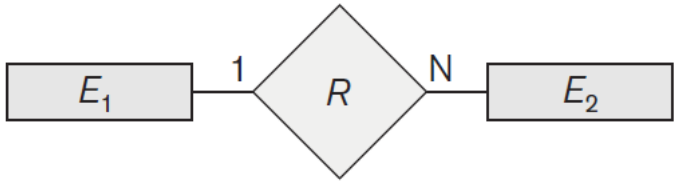


→ …

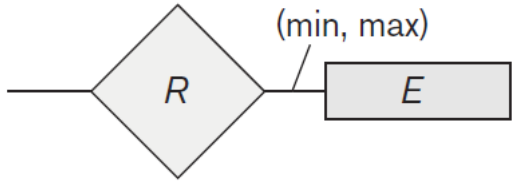
ER Components 3/3



Total Participation of E_2 in R



Cardinality Ratio 1: N for $E_1 : E_2$ in R

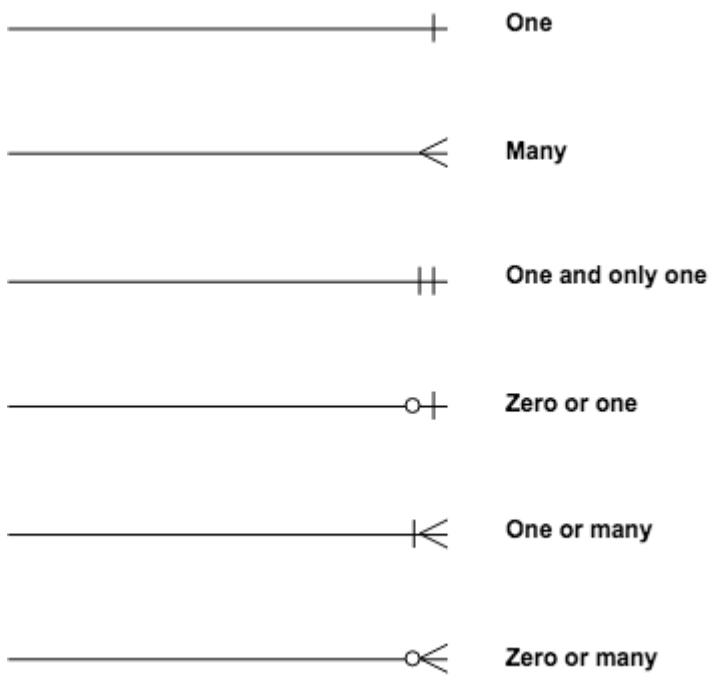


Structural Constraint (min, max) on Participation of E in R

→ Total participation can also be understood as E2 HAS to have one entry.

→ Example, a teacher MUST teach a class (or they are not a teacher).

Cardinalities



Crow’s foot notation

→ 1 - * , one to many, 1:N

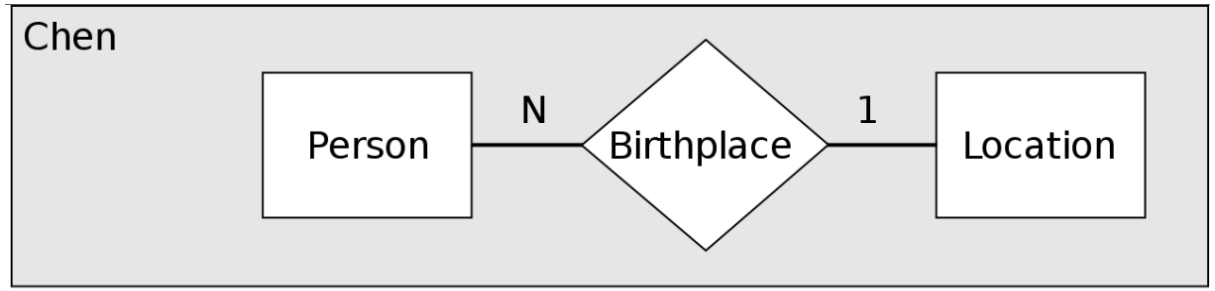
→ * - 1, many to one, N:1

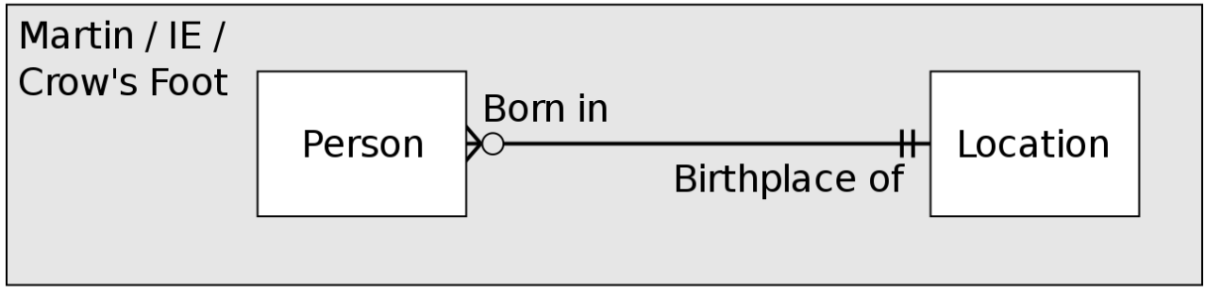
→ * - *, many to many M:N

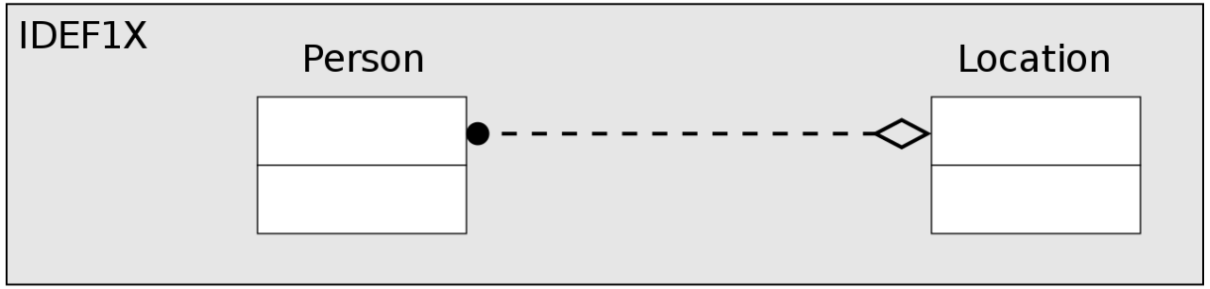
→ 1 - 1, one to one 1:1

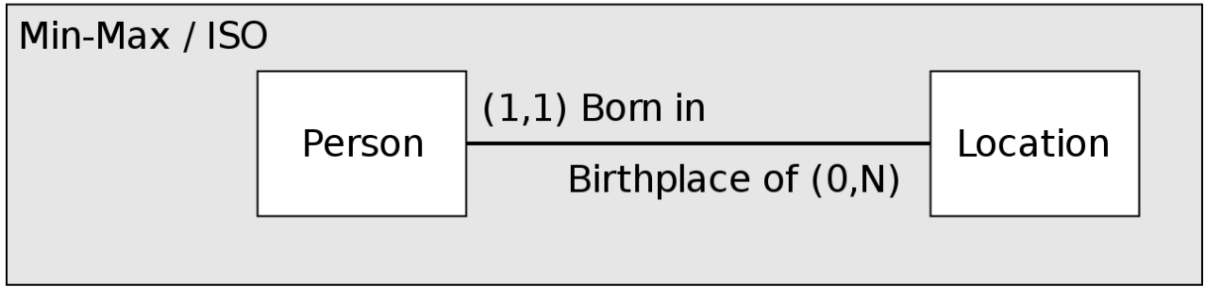
→ 0..1 - 1, zero/one to one, 0..1:1

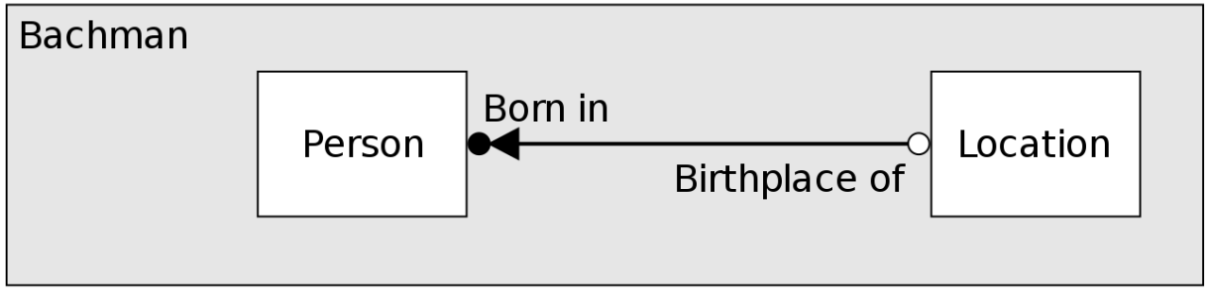
And there are many alternatives

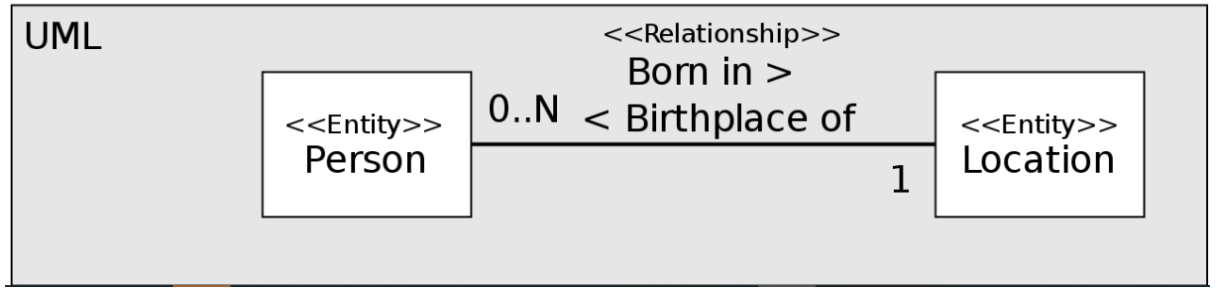




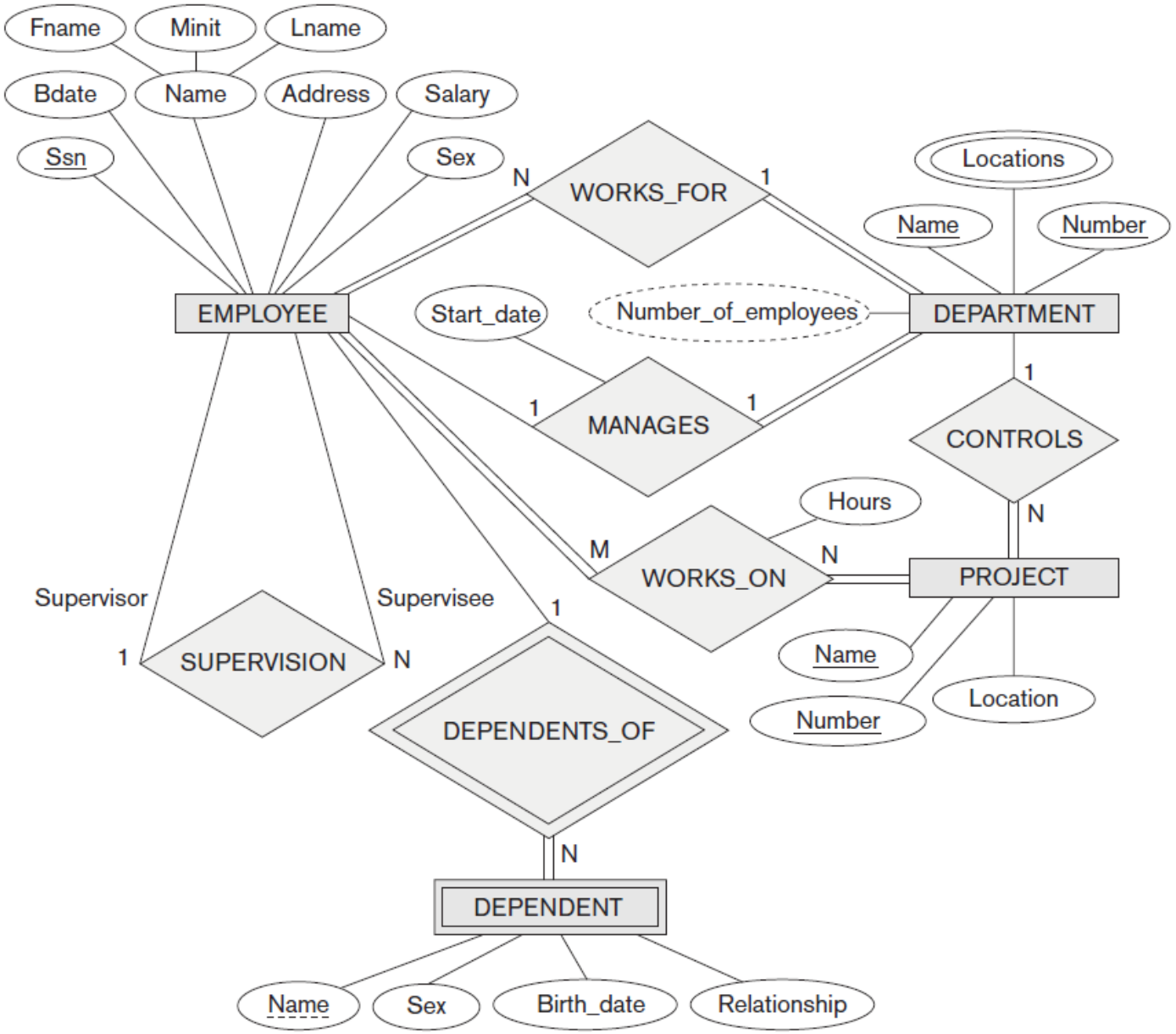






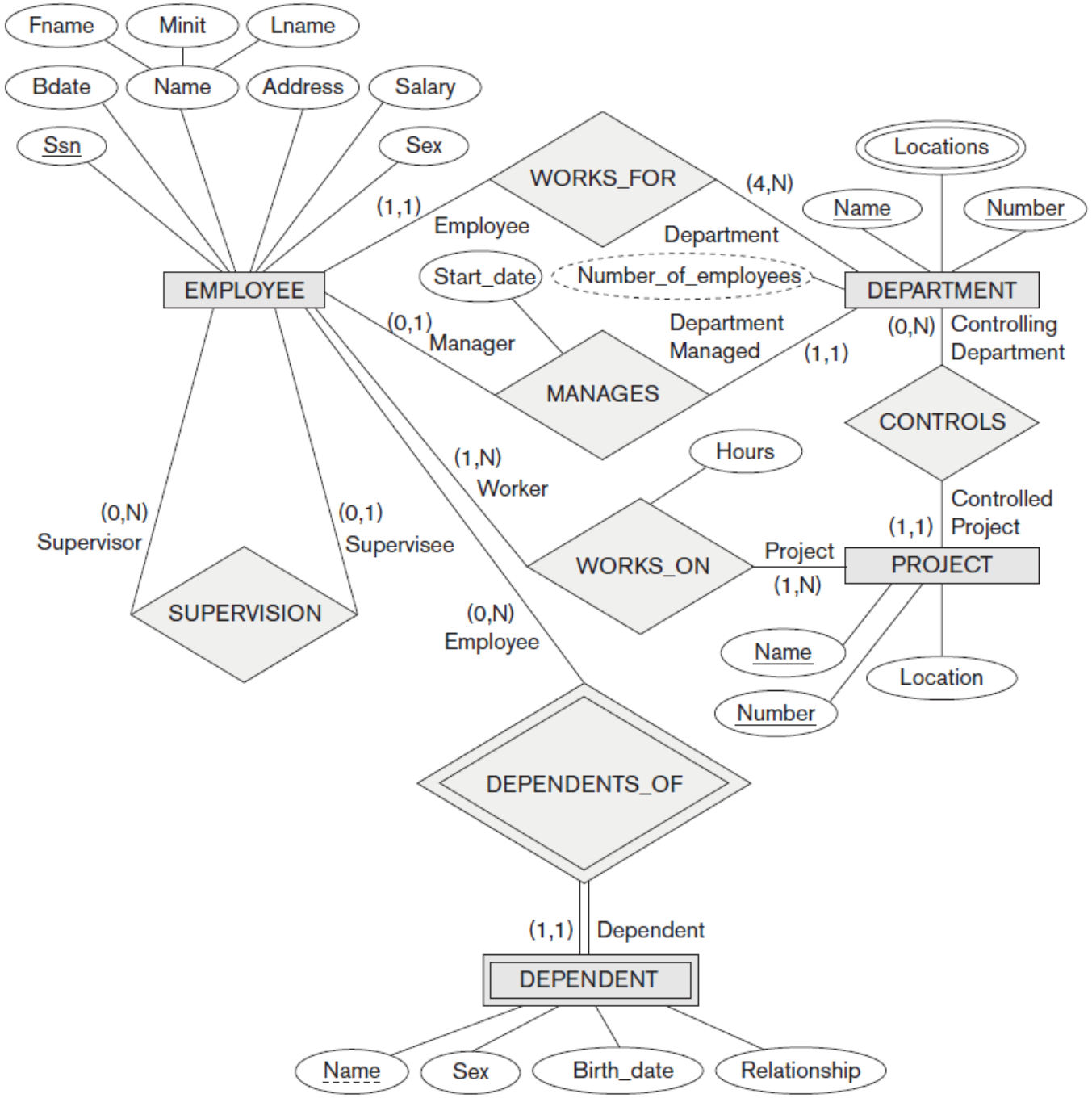


ER Diagram Example



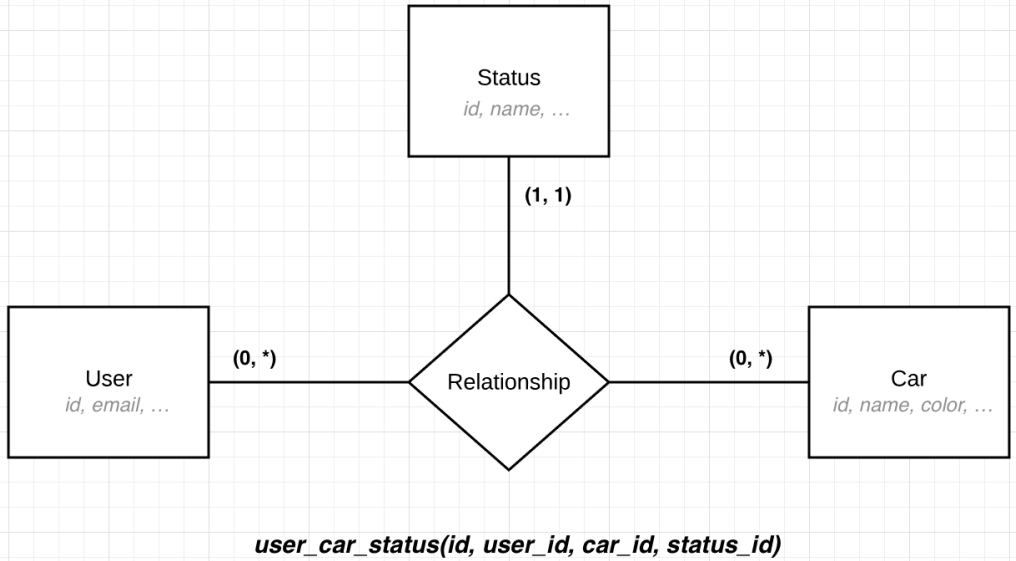
- → Entity types (strong/weak)
- → Relationship types (Identifying or not)
- → Recursive relationship
- → Attribute types (composite, key, derived, multiple values, etc.)
- → Cardinalities

Alternative structural constraints



→ Min, Max notation

N-ary



→ Multiple relationships in one

→ Results in table with multiple foreign keys

EER: Enhanced entity-relationship diagram

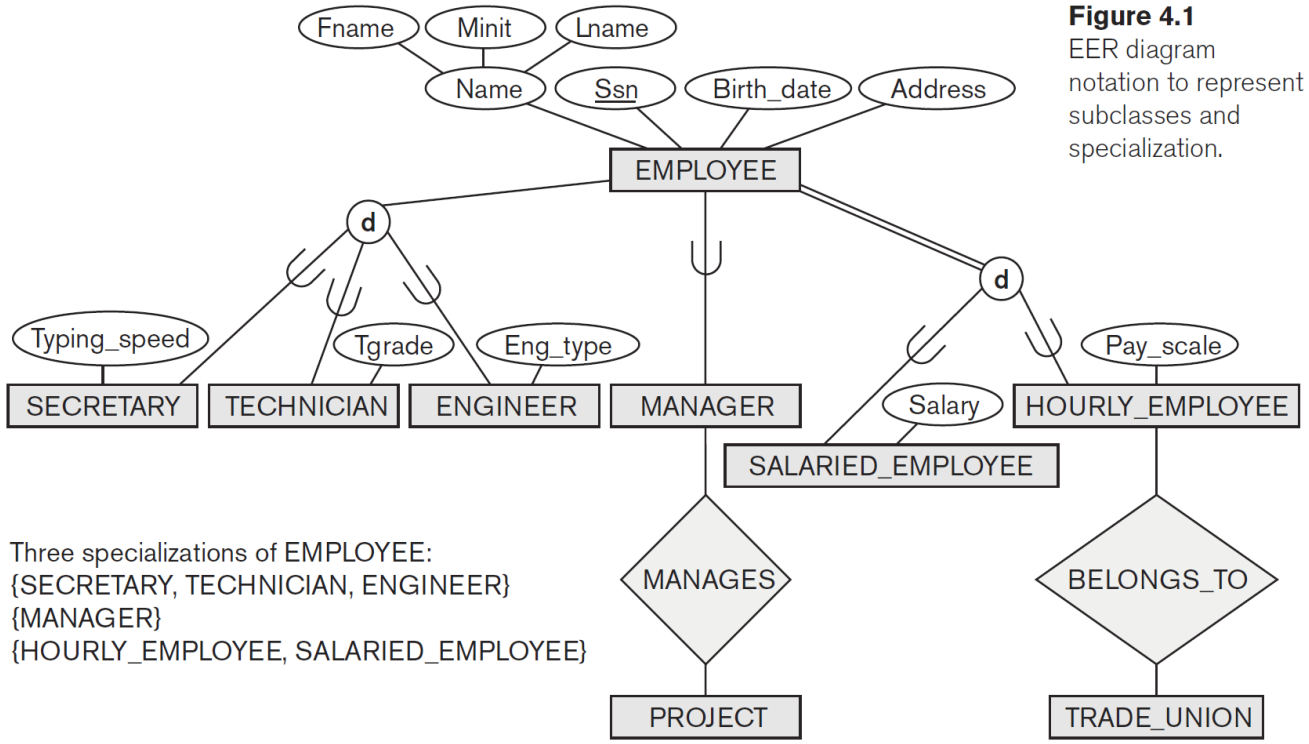


Figure 4.1
EER diagram notation to represent subclasses and specialization.

Three specializations of EMPLOYEE:
{SECRETARY, TECHNICIAN, ENGINEER}
{MANAGER}
{HOURLY_EMPLOYEE, SALARIED_EMPLOYEE}

→ Expands with the following components:

- → Attribute or relationship inheritances
- → Category or union types
- → Specialization and generalization
- → Subclasses and superclasses

Inheritance

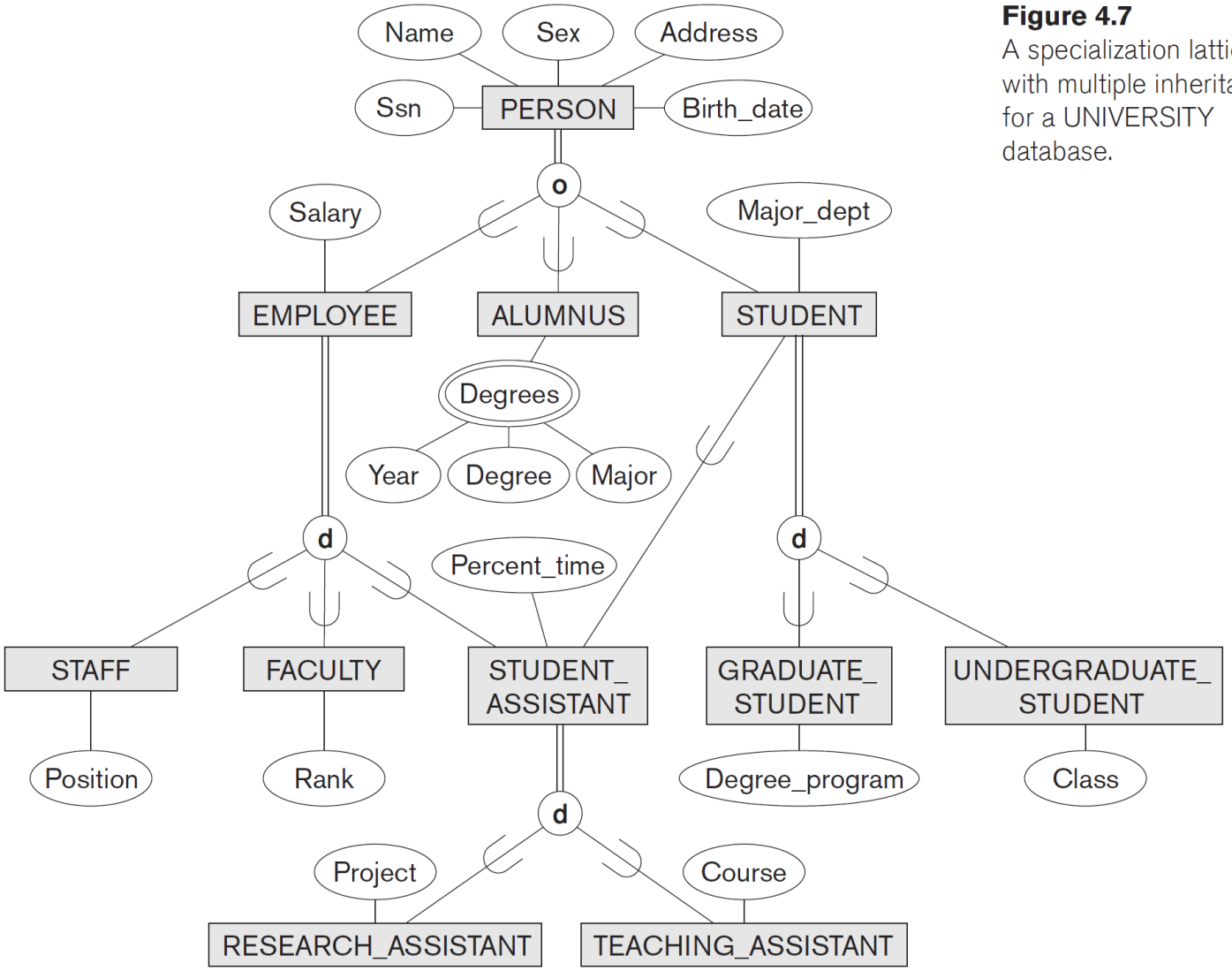
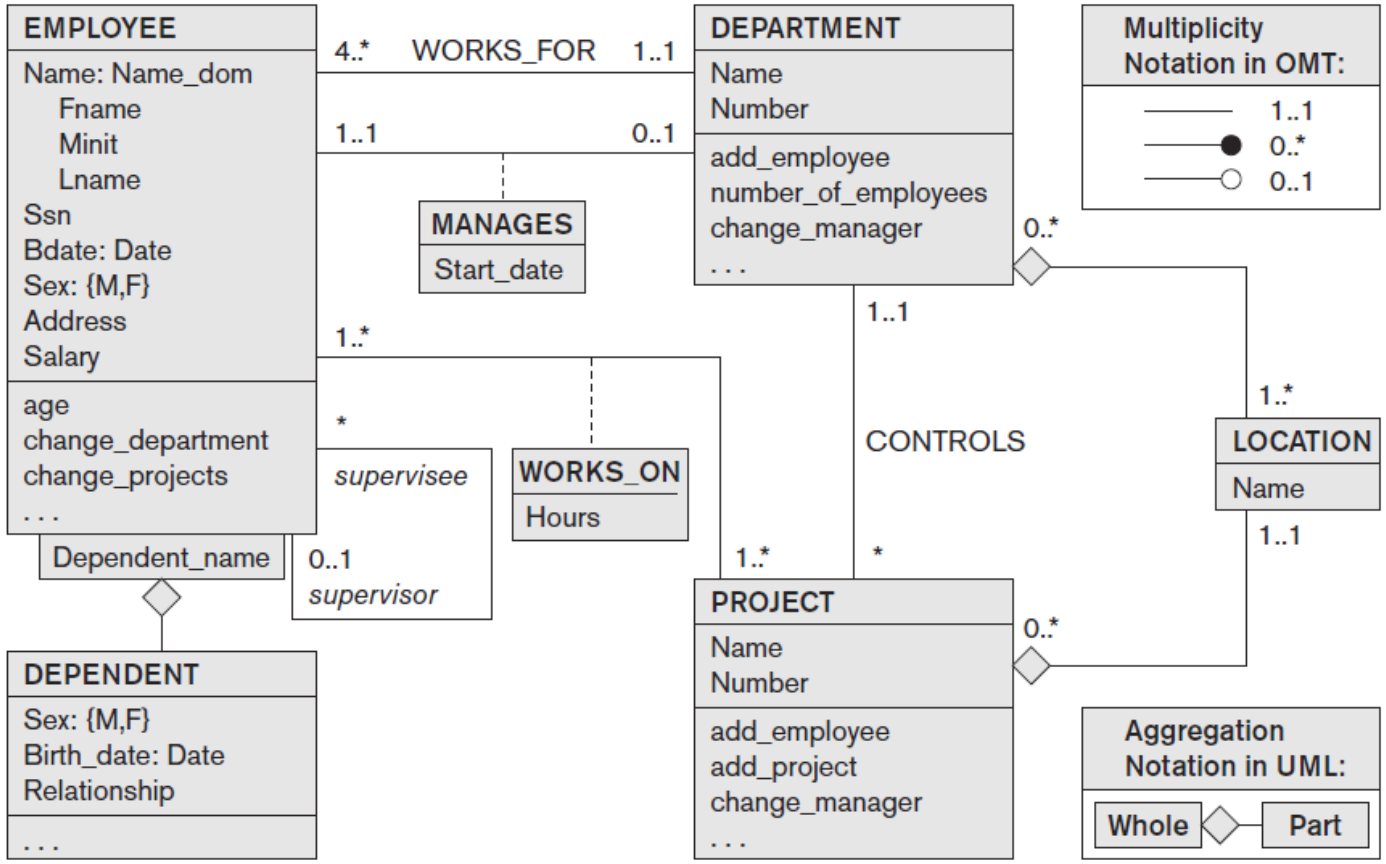


Figure 4.7
A specialization lattice with multiple inheritance for a UNIVERSITY database.

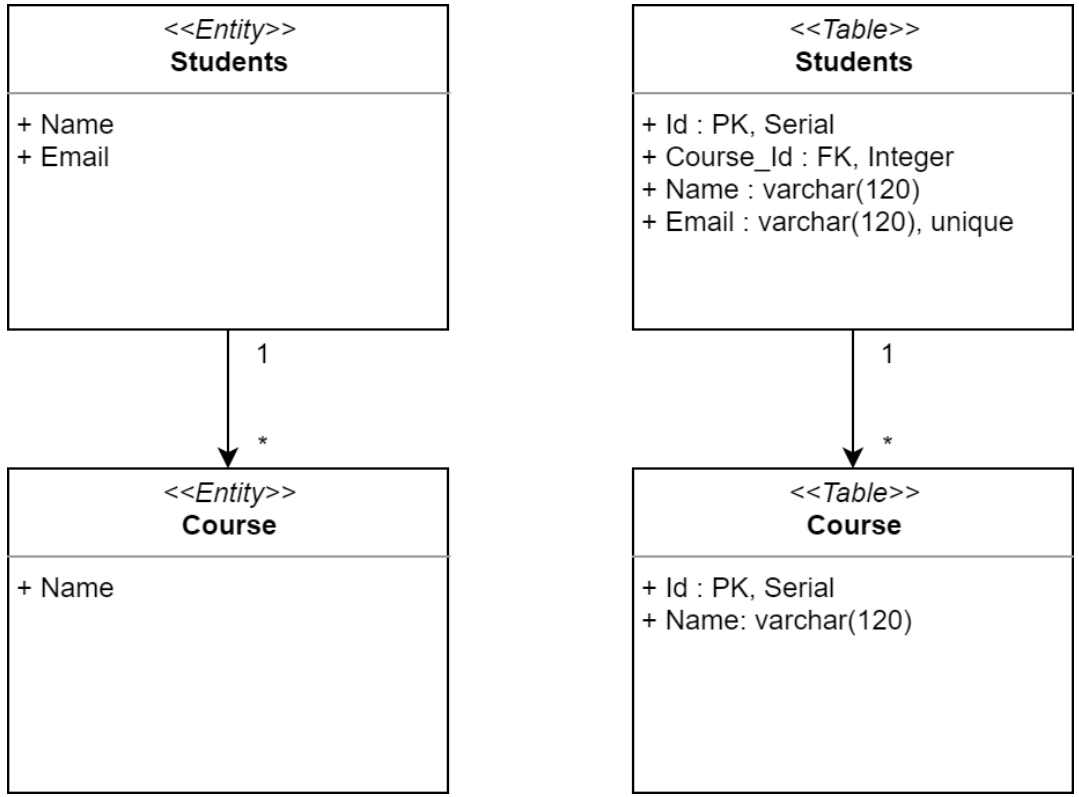
- → Disjoint
  - → Is one of
- → Overlapping
  - → Can be any of, and more than one.

UML Notation



- → Often preferred due to the common use of UML for class diagrams.
- → Notation differs.
- → Not the main notation for this class!
- → At the exam use the notation shown prior to UML in these slides.

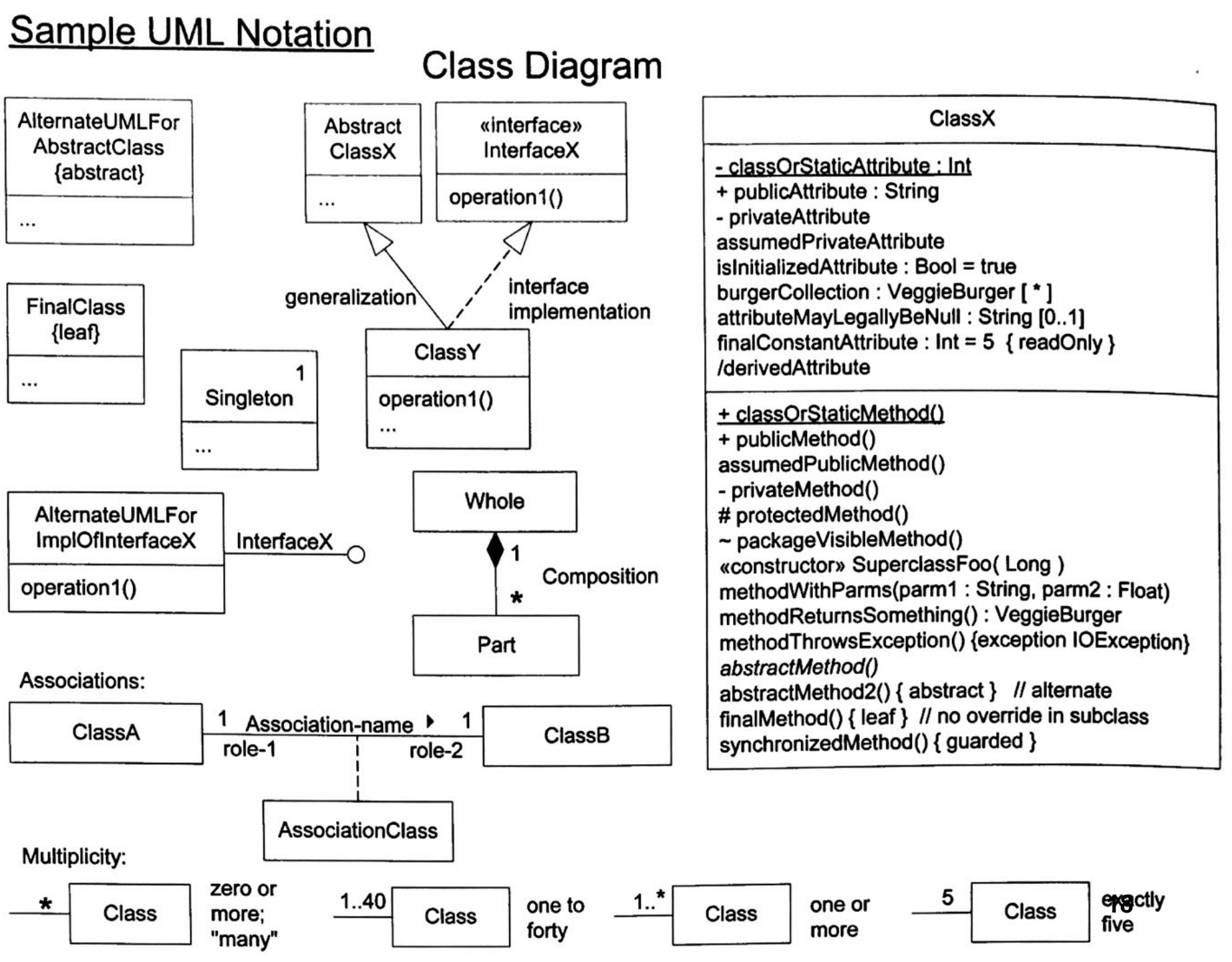
UML Entity vs Table



→ If no << >> is present, we are talking about an entity.

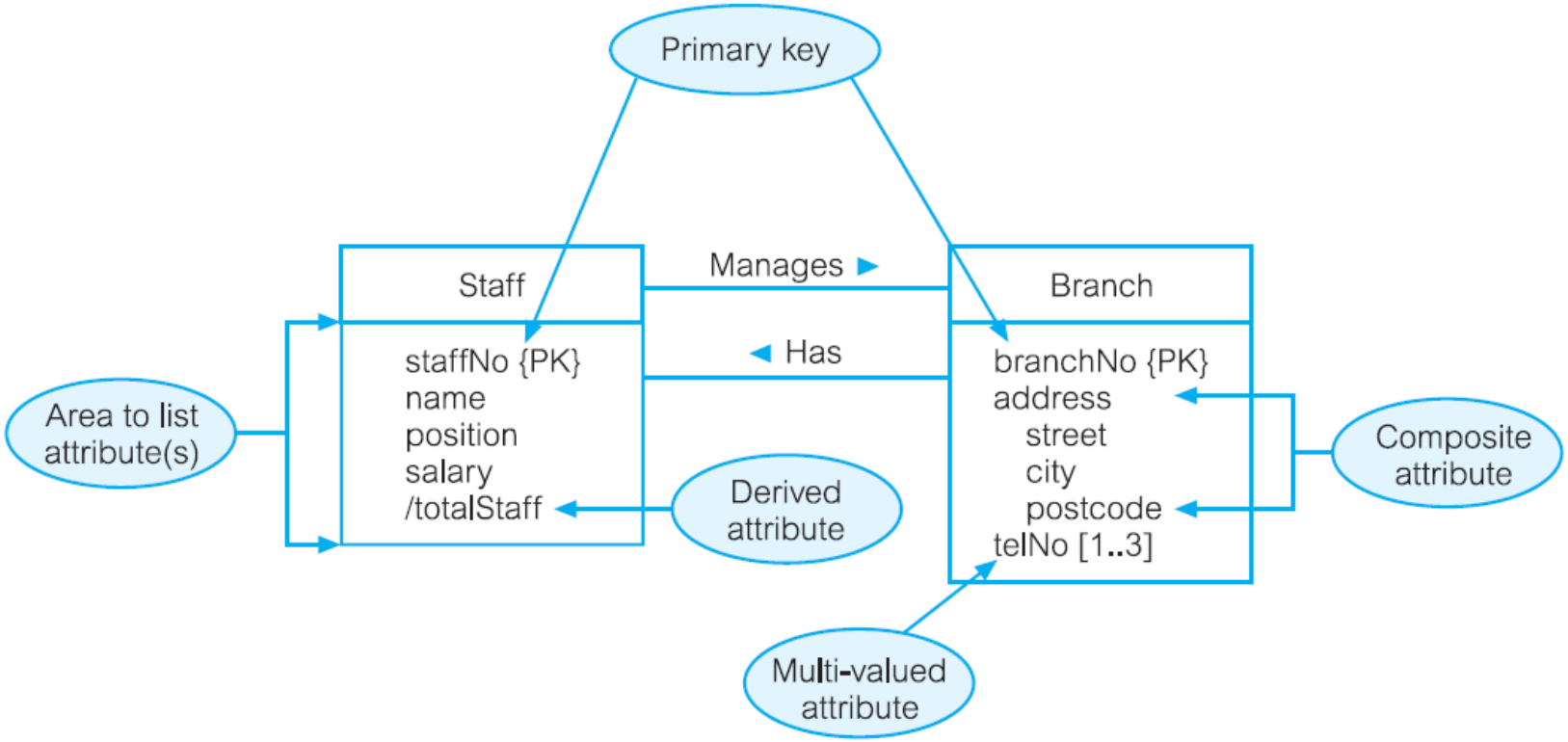
→ **Please be explicit in this course about what type it is!**

Standard UML Notation

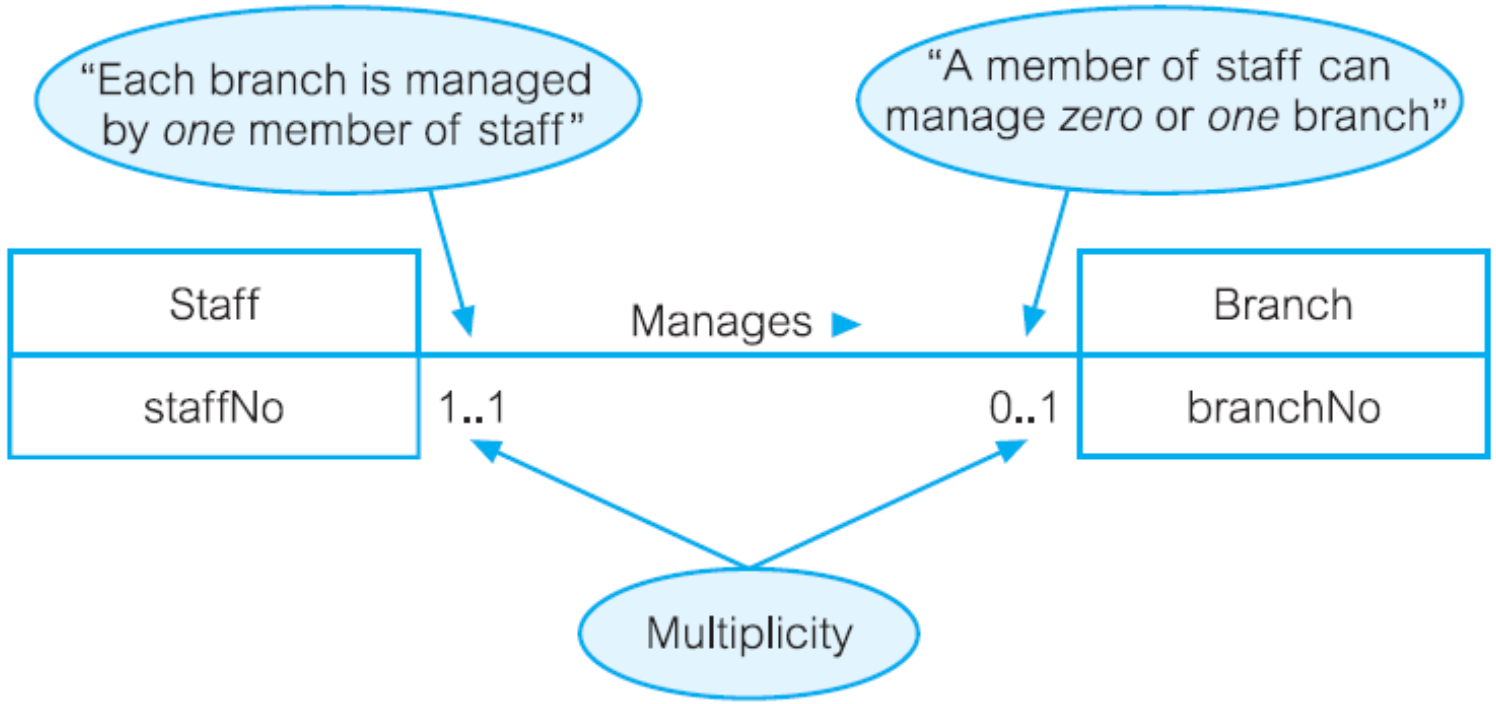


Source:
Applying UML and Patterns, Larman, 2009

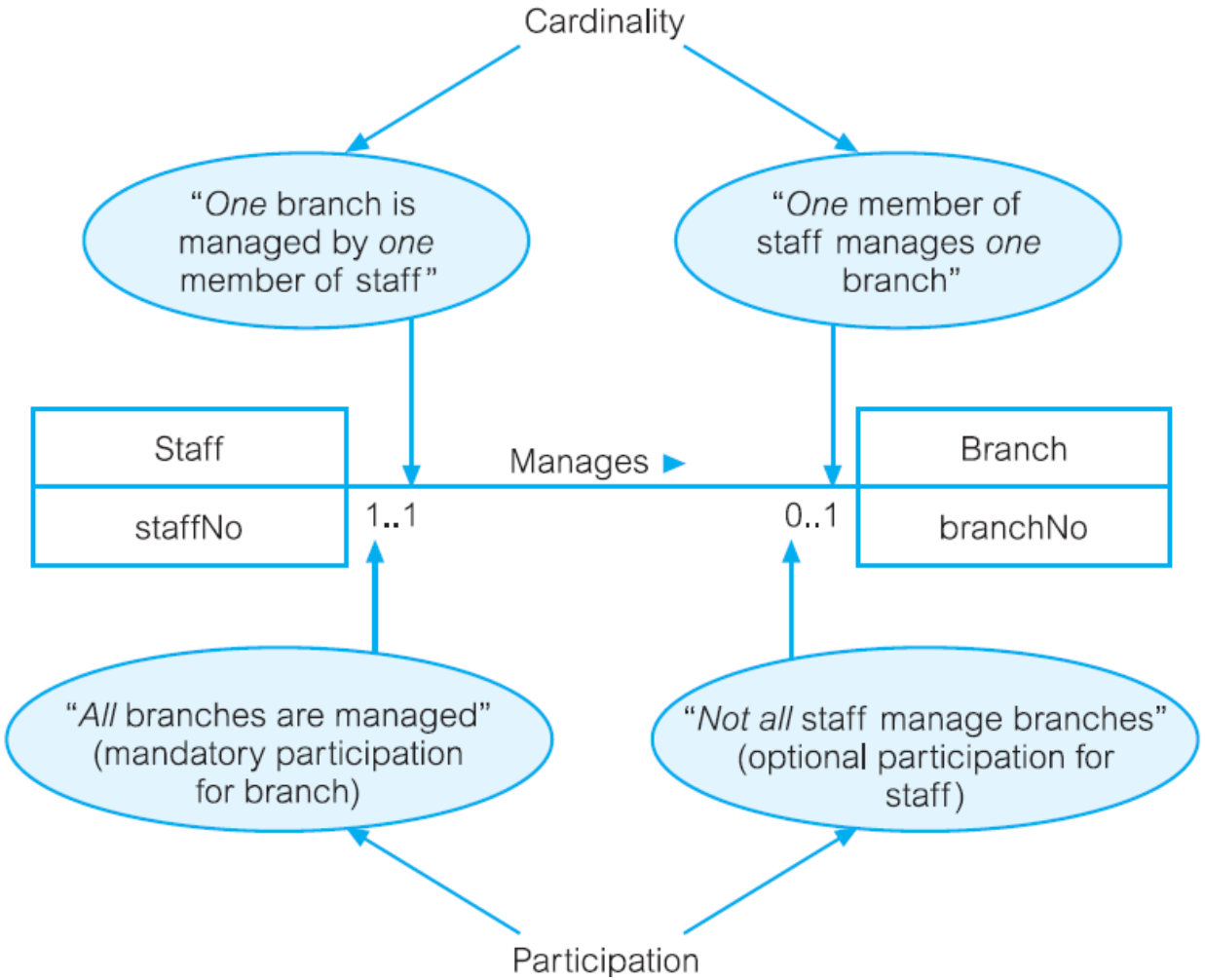
Attributes in UML



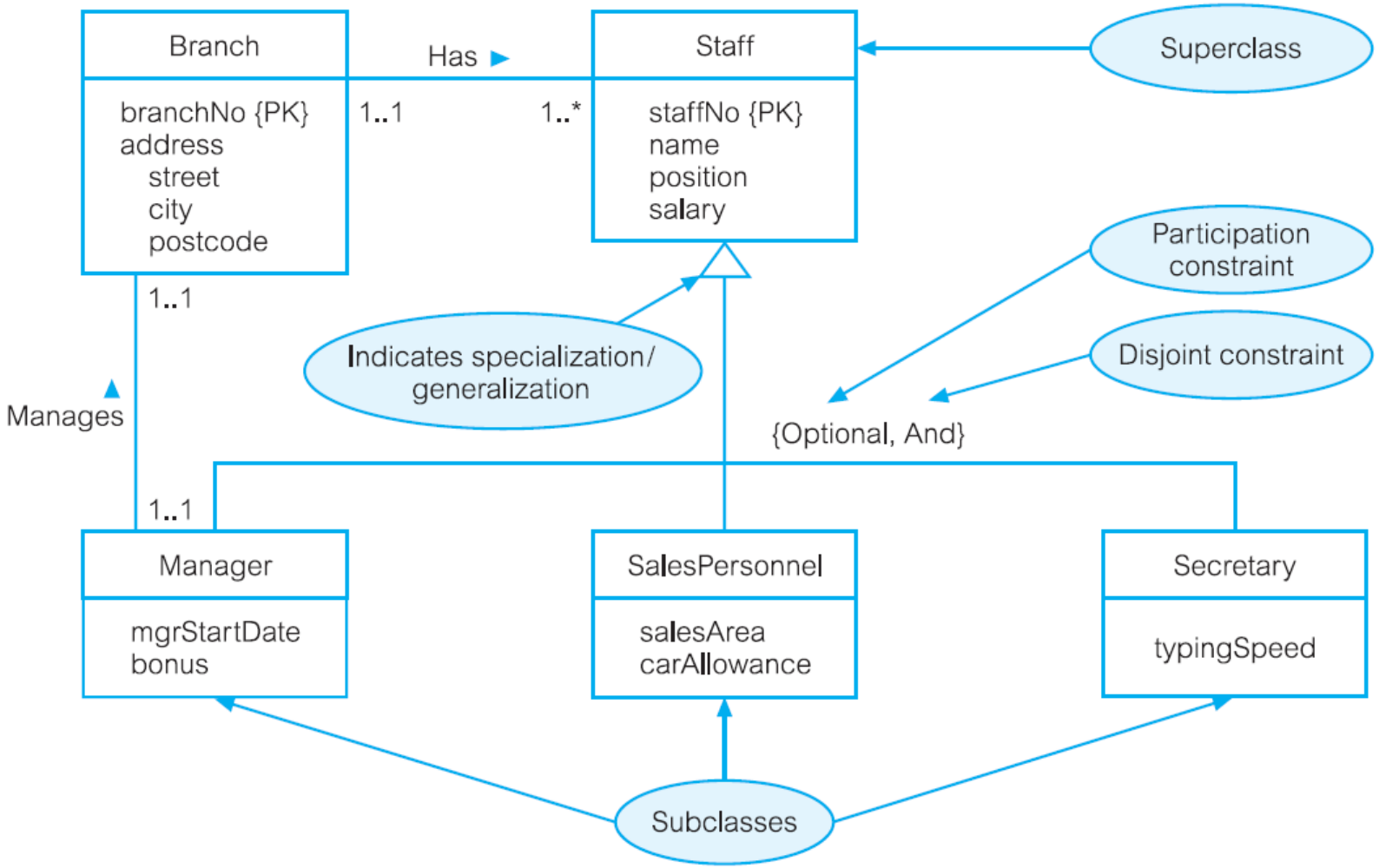
Cardinalities in UML - 1/2



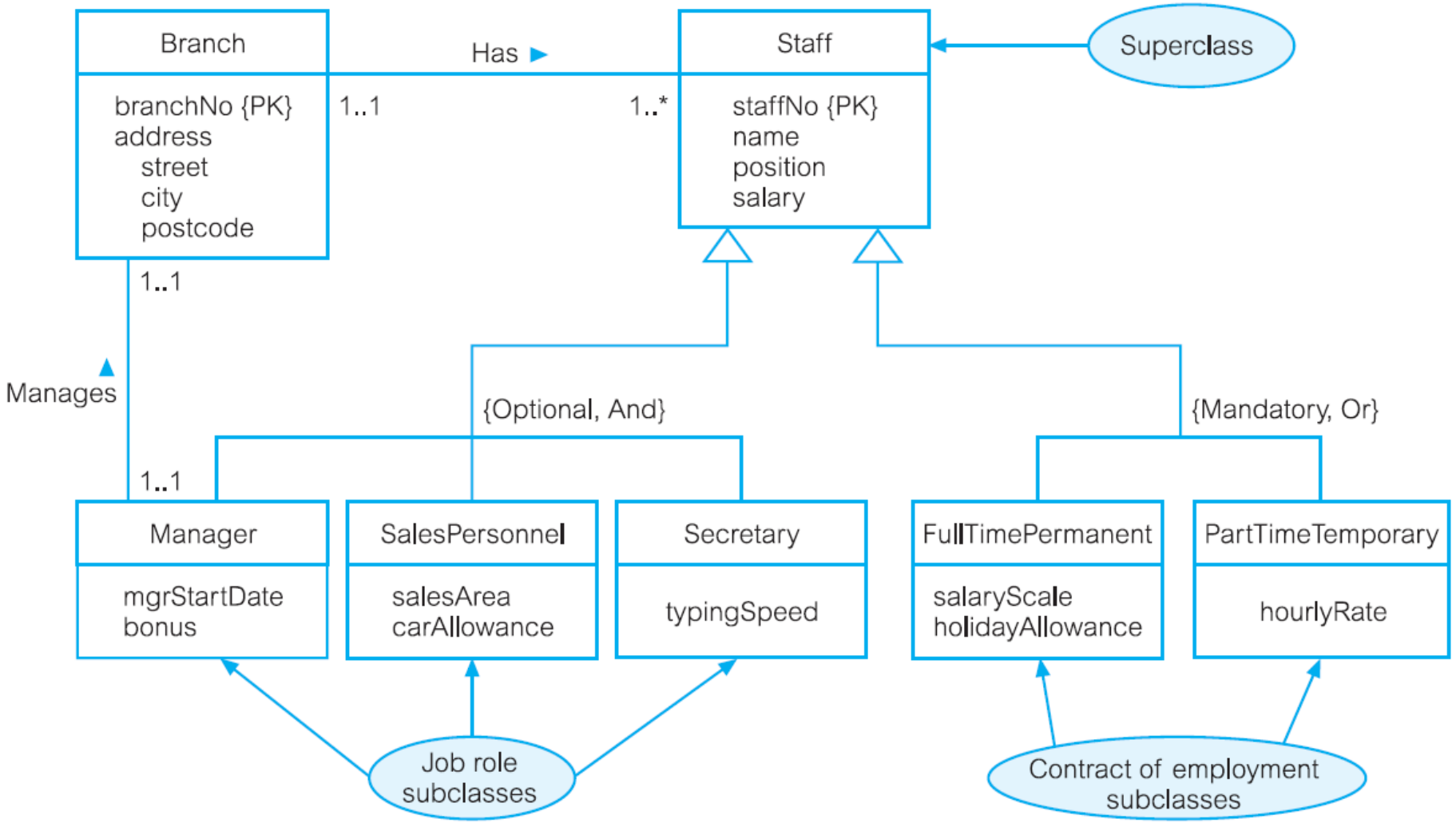
Cardinalities in UML – 2/2



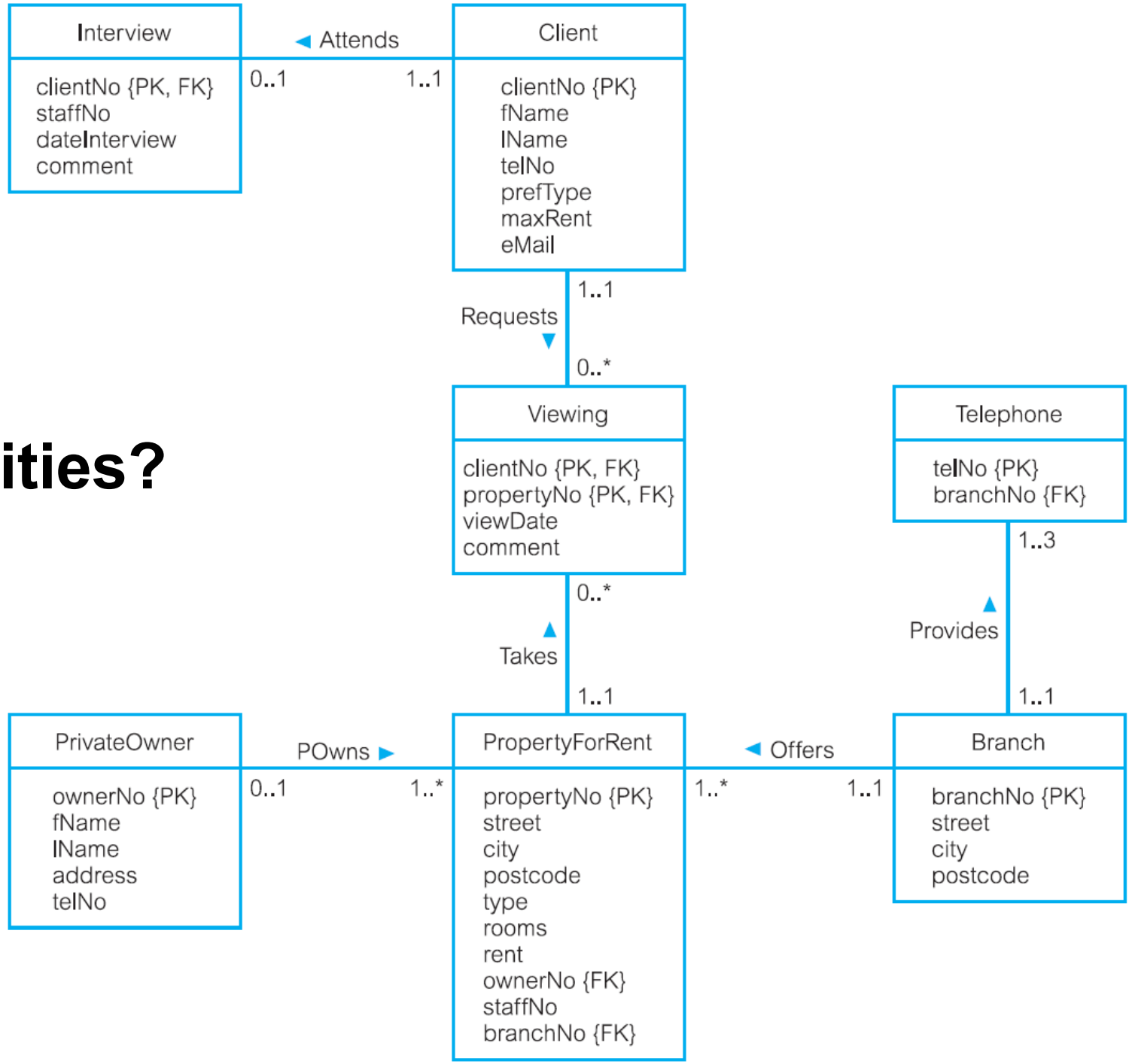
EER in UML – 1/2

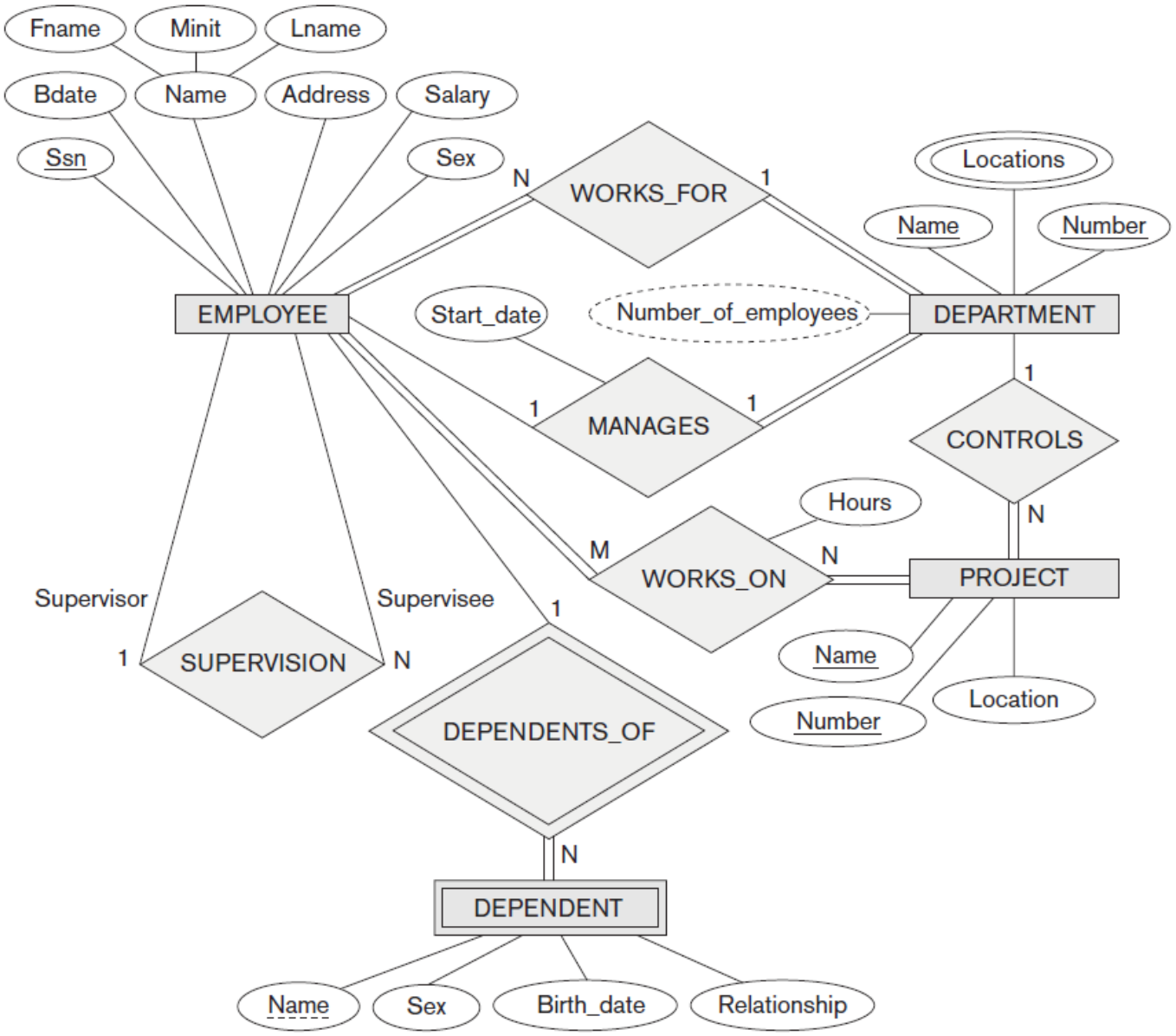

Branch
branchNo {PK}
address
street
city
postcode
Has
1..1
1..*
Staff
staffNo {PK}
name
position
salary
Superclass
Participation constraint
Disjoint constraint
Indicates specialization/ generalization
{Optional, And}
Manages
1..1
1..1
Manager
mgrStartDate
bonus
SalesPersonnel
salesArea
carAllowance
Secretary
typingSpeed
Subclasses

EER in UML – 2/2


Branch
branchNo {PK}
address
street
city
postcode
Has
1..1
1..*
Staff
staffNo {PK}
name
position
salary
Superclass
1..1
Manages
1..1
{Optional, And}
{Mandatory, Or}
Manager
mgrStartDate
bonus
SalesPersonnel
salesArea
carAllowance
Secretary
typingSpeed
FullTimePermanent
salaryScale
holidayAllowance
PartTimeTemporary
hourlyRate
Job role subclasses
Contract of employment subclasses

Tables or entities?





Mapping ER to Tables

- → Step 1: Mapping of Regular Entity Types.
- → Step 2: Mapping of Weak Entity Types.
- → Step 3: Mapping of Binary 1:1 Relationship Types.
  - → Foreign key approach
  - → Merged relation approach
  - → Cross-reference or relationship relation approach
- → Step 4: Mapping of Binary 1:N Relationship Types.
  - → The foreign key approach.
  - → The relationship relation approach.
- → Step 5: Mapping of Binary M:N Relationship Types.
- → Step 6: Mapping of Multivalued Attributes.
- → Step 7: Mapping of N-ary Relationship Types.

Figure 9.3
Illustration of some mapping steps.
(a) *Entity* relations after step 1.
(b) Additional *weak entity* relation after step 2.
(c) *Relationship* relations after step 5.
(d) Relation representing multivalued attribute after step 6.

(a)

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary

DEPARTMENT

Dname	Dnumber

PROJECT

Pname	Pnumber	Plocation

(b)

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

(c)

WORKS_ON

Essn	Pno	Hours

(d)

DEPT_LOCATIONS

Dnumber	Dlocation

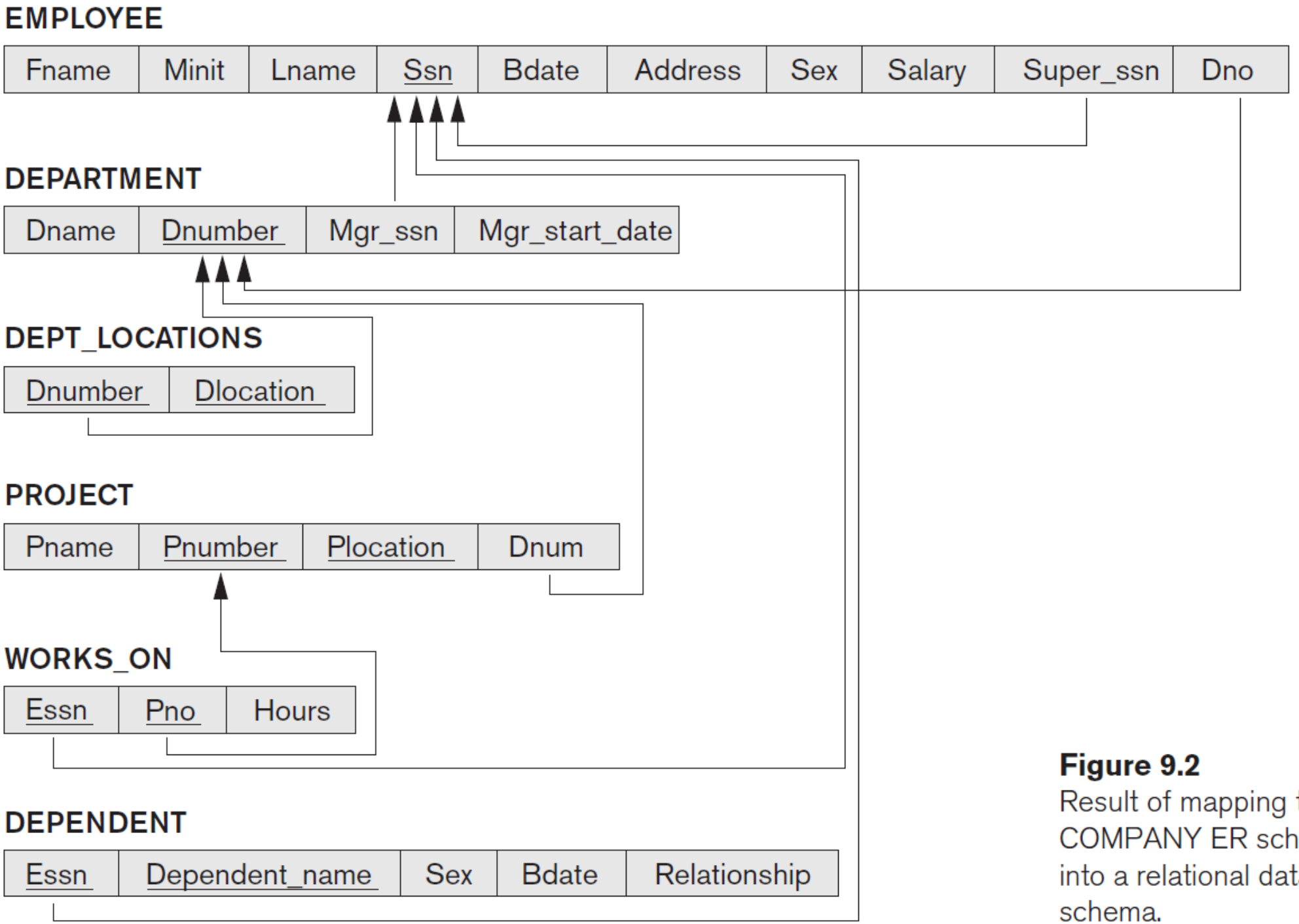
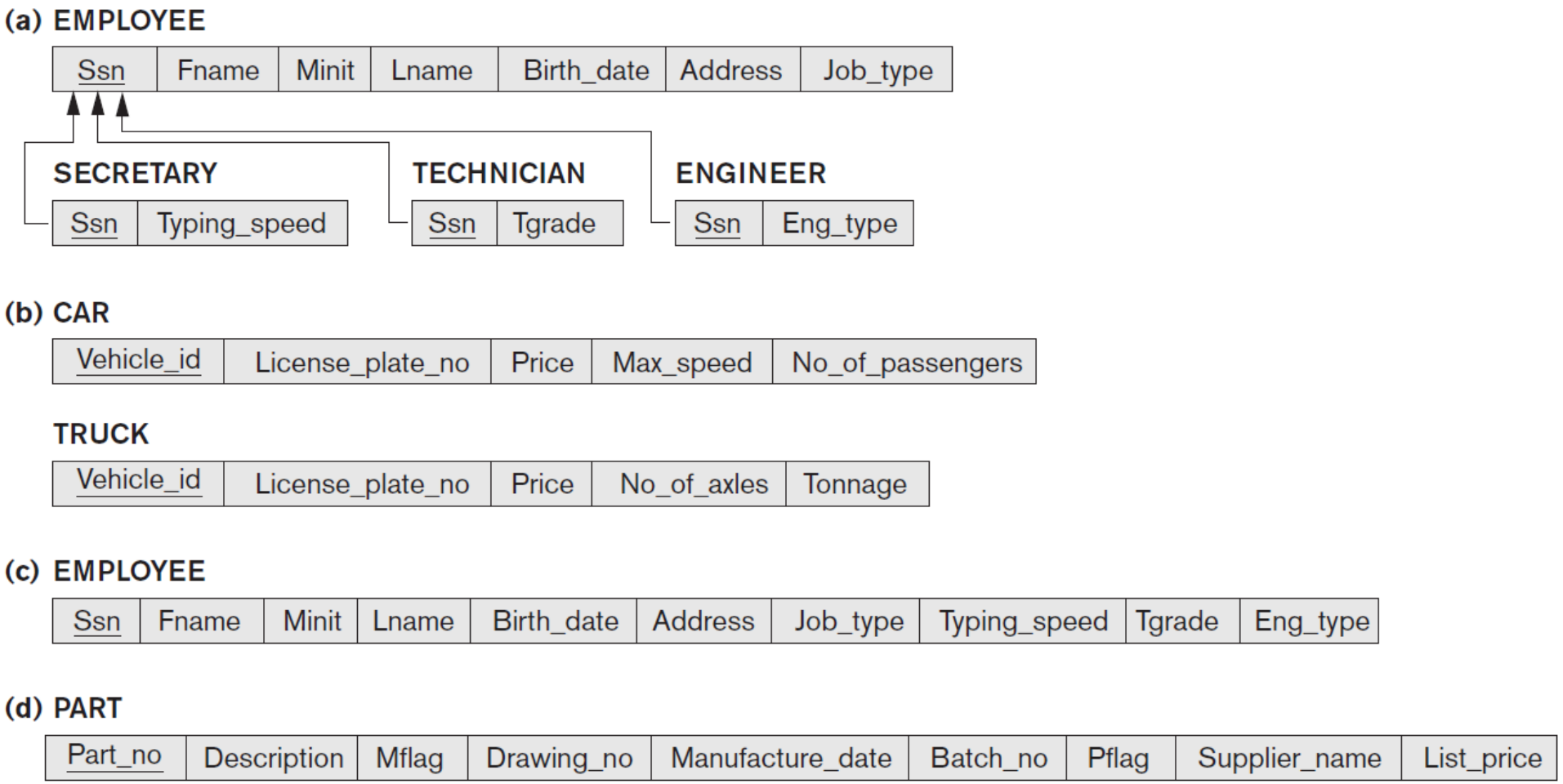


Figure 9.2
Result of mapping the COMPANY ER schema into a relational database schema.

Maturing the UML diagram to tables (mixed example!)



In relation to UP – Repeated

Phase	Artifact
Business Modelling	Domain Model (None from DM)
Requirements	(None from DM)
Analysis	ER, EER, UML (Entities only)
Design	UML (Tables)
Implementation	SQL Creation Script
Test	(None from DM)
Deployment	(None from DM)

ER model vs Relational Model (Tables)

Table 9.1 Correspondence between ER and Relational Models

ER MODEL	RELATIONAL MODEL
Entity type	*Entity* relation
1:1 or 1:N relationship type	Foreign key (or *relationship* relation)
M:N relationship type	*Relationship* relation and *two* foreign keys
n-ary relationship type	*Relationship* relation and *n* foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

Process Recap

ER / EER / UML (Entity)

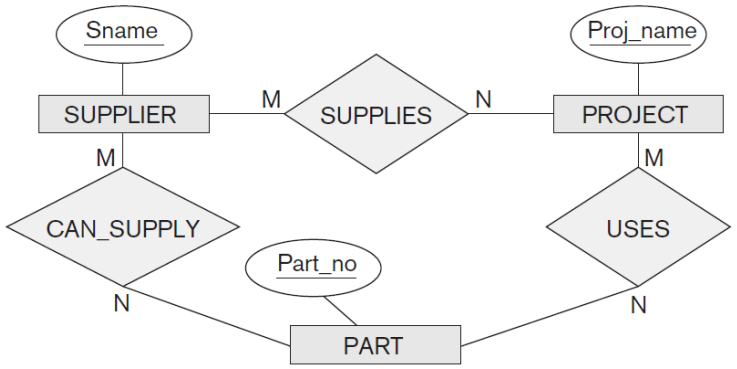


Table format / UML (Table)

Remember << Table >>

EMPLOYEE

Fname	Minit	Lname	Ssn

SQL Creation Script

```sql
CREATE TABLE account(
    user_id serial PRIMARY KEY,
    username VARCHAR (50) UNIQUE NOT NULL,
    password VARCHAR (50) NOT NULL,
    email VARCHAR (355) UNIQUE NOT NULL,
    created_on TIMESTAMP NOT NULL,
    last_login TIMESTAMP
);
```