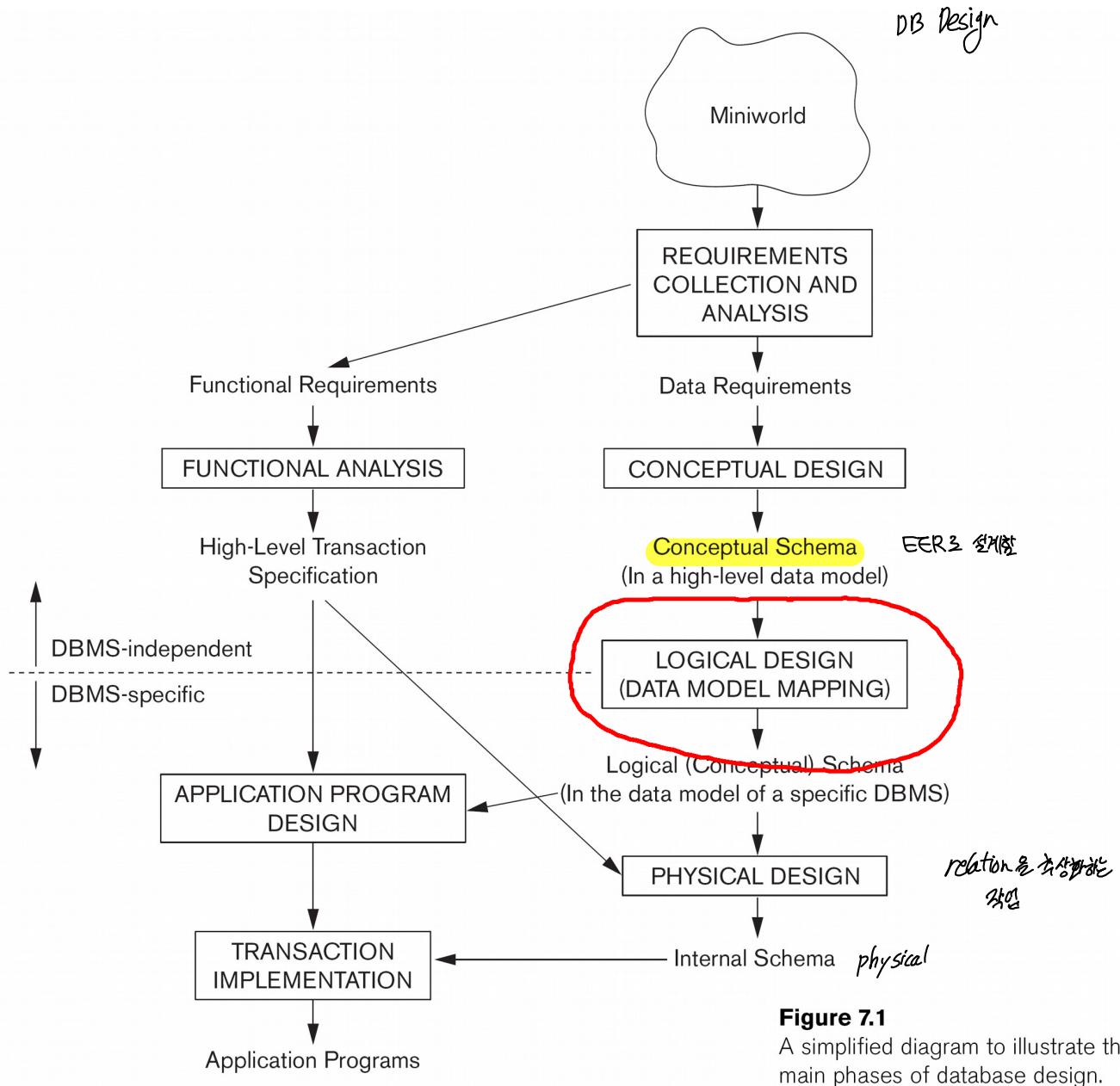


Ch 8. Relational Database Design by ER- and EER-Relation Mapping

Where are we?



Relational Database Design by ER-Relation Mapping

- STEP 1
 - 엔티티 타입은 릴레이션으로 매핑한다. 엔티티 타입의 키 중에서 하나를 릴레이션의 기본 키로 지정한다.
- STEP 2
 - 약한 엔티티 타입도 릴레이션으로 매핑하되 소유 릴레이션 (owner relation)의 키 속성을 포함시킨다. 생성된 릴레이션의 기본 키는 소유 릴레이션의 키와 약한 엔티티 타입의 부분키를 합쳐서 만든다.
- STEP 3

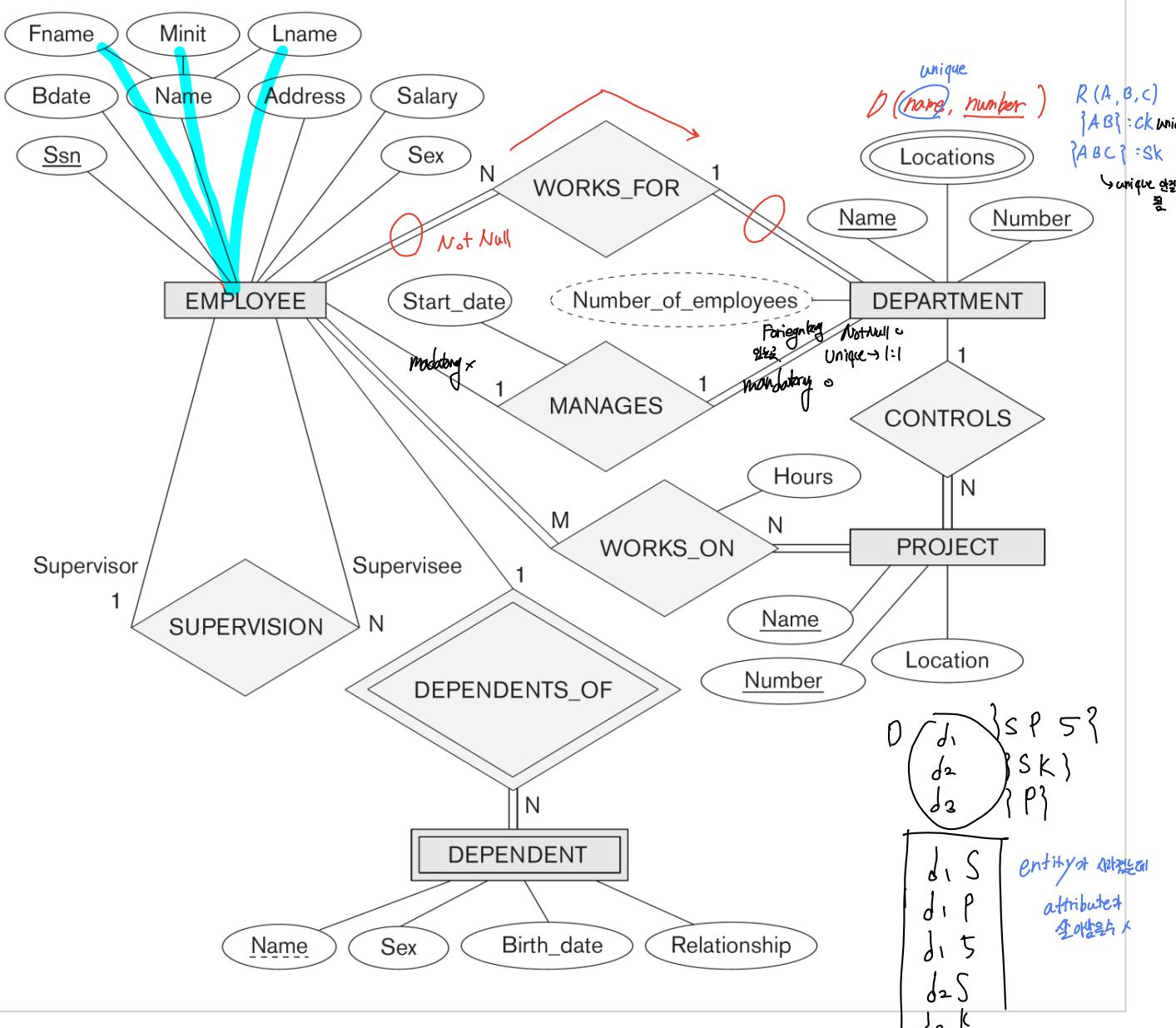
- 1:1 이진 관계는 관계에 참여하는 두 릴레이션 중에서 어느 하나의 (전체 참여 엔티티가 있으면 그 엔티티를 선택) 외래키 속성으로 매핑한다. 두 엔티티 탑입이 모두 완전하게 참여하는 경우, 두 테이블을 하나의 테이블로 병합할 수 있다.
 - STEP 4
 - 1:N 이진 관계는 N-side 릴레이션의 외래키 속성으로 매핑하며, 1-side의 주 키를 참조하도록 한다.
 - STEP 5
 - N:M 이진 관계는 별도의 릴레이션 (이를 관계 릴레이션이라고 부름)으로 생성하고, 관계에 참여하는 두 릴레이션의 기본 키를 각각 참조하는 외래키로 애트리뷰트를 구성한다. 이 때 두 외래키가 관계 릴레이션의 기본키를 형성한다.
 - STEP 6
 - 다중값 애트리뷰트는 키를 포함하는 릴레이션으로 매핑된다.
 - STEP 7
 - n 차 관계는 관계에 참여하는 n 개의 릴레이션의 키들로 구성되는 관계 릴레이션으로 매핑된다. 관계 릴레이션의 애트리뷰트들은 참여 릴레이션의 주 키를 참조하는 외래키들과 관계 속성(들)으로 구성된다.

ER Model : Company

Figure 9.1

The ER conceptual schema diagram for the COMPANY database.

관계형 DB에서는 오직 Relation 밖에 없으니.



STEP 1: Mapping of Regular Entity Types

- ER 스키마의 각 정규(강한) 엔티티 타입 E에 대해 E의 모든 simple attribute(또는 composite attribute의 단순 구성 요소)을 포함하는 관계 R을 작성한다.
 - E의 key attribute 중 하나를 R의 primary key 키로 선택한다. 다른 key attribute는 unique(candidate key)로 선언하여야 한다.
 - Figure 9.1에서 STEP 1을 적용해 볼 것!
 - Figure 9.3(a)에서 확인
- 연관 key가 Composite인 경우, 선택한 key의 simple attributes의 set이 Rel primary key를 구성.

Figure 9.3

Illustration of some mapping steps.
a. Entity relations after step 1.

b. Additional weak entity relation after step 2.

c. Relationship relation 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
------	----------------	-----	-------	--------------

DEPENDENT PK

WORKS_ON

Essn	Pno	Hours
------	-----	-------

DEPT_LOCATIONS

Dnumber	Dlocation
---------	-----------

STEP 2: Mapping of Weak Entity Types

- owner entity type E와 연관된 weak entity type W에 대해, W의 모든 simple attribute를 포함하는 관계 R을 만든다.
- 모든 owner entity type의 primary key attribute를 R의 foreign key로 포함한다.
- R의 primary key는 owner entity type의 primary key와 weak entity type W의 partial key(있다면)를 포함한다.
- Figure 9.1에서 STEP 2를 적용해 볼 것!
- Figure 9.3(b)에서 확인
- Weak entity는 Strong Entity에 존재 종속적(existence dependency)이기 때문에 Foreign Key Option에서 (ON UPDATE, ON DELETE) CASCADE option을 선택하는 것이 일반적이다.

STEP 3: Mapping of Binary 1:1 Relationship Types

- ER 스키마에서 각 이진 1:1 관계 타입 R에 대해, R에 참여하는 엔티티 타입에 해당하는 relation S와 T를 찾는다.
- Three possible approaches:
 - Foreign Key approach: relation S를 선택하고 S의 foreign key로 T의 primary key를 포함한다. (S와 T가 전체 참여(total participation)하는 entity type을 S로 선택하는 것이 좋다. Why?)
 - Example (see Figure 9.2): DEPARTMENT는 1:1 relationship MANAGES에 전체 참여이다. (Fig. 9.1)

다른 relation T의 PK를 S의 PK로 포함

- DEPARTMENT의 Mgr_SSN이 EMPLOYEE를 참조하는 foreign key가 된다. (NULL에 대한 Option은? Allowed?) EMPLOYEE의 PK인 Ssn을 DEPARTMENT의 fk인 Mgr_ssn으로 가진다 (Mgr_ssn이 NULL을 허용)
 - MANAGES의 애트리뷰트는 DEPARTMENT의 애트리뷰트가 된다.
 - Merged relation option: S와 T를 하나의 relation R로 병합한다. (S와 T의 참여가 모두 전체 참여(total participation)인 경우에 가능한 선택사항이다. Why?) entities가 total participation인 경우 적절
 - R의 primary key, candidate key?
 - Relationship relation option: 1:1 relationship을 나타내는 새로운 relation R을 만들 수 있다.
 - extra JOIN 비용이 단점이다.
 - R의 primary key, candidate key?
 - Figure 9.1에서 STEP 3을 적용해 볼 것!
 - Figure 9.2에서 확인
-

STEP 4: Mapping of Binary 1:N Relationship Types

- 1:N relationship type에서 N-side entity type에 해당하는 relation S를 찾는다.
 - 1:N relationship에 참여하는 1-side entity type에 해당하는 relation T의 primary key를 S의 foreign key로 포함한다.
 - 1:N relationship type의 simple attribute를 S의 attribute로 포함시킨다.
 - Examples (Figures 9.1): 1:N relationship types - WORKS_FOR, CONTROLS, and SUPERVISION에 대해 STEP 4를 적용해 볼 것!
 - Figure 9.2에서 확인해 볼 것!
 - N-side entity type이 전체 참여인 경우, 취해야 할 행동은?
 - 1-side entity type이 전체 참여인 경우, 취해야 할 행동은?
 - ON DELETE시에 취해야 할 Foreign Key Option은?
 - Relationship relation option: 1:N relationship을 나타내는 새로운 relation R을 만들 수 있다.
 - extra JOIN 비용이 단점이다.
 - R의 primary key, candidate key?
-

STEP 5: Mapping of Binary M:N Relationship Types

- M:N relationship type R에 해당하는 새로운 relation S를 만든다.
 - R에 참여하는 두 entity type에 해당하는 두 relation의 primary key들을 S에 foreign key로 포함한다.
 - S의 primary key는 두 foreign key의 조합으로 구성된다.
 - M:N relationship type의 simple attribute를 S의 attribute로 포함시킨다.
 - Relationship instance는 참여하는 Entity에 존재 종속적(existence dependency)이기 때문에 Foreign Key Option에서 (ON UPDATE, ON DELETE) CASCADE option을 선택하는 것이 일반적이다.
 - Example: M:N relationship type WORKS_ON (Figure 9.1)에 대해 STEP 5를 적용해 보자.
 - Figure 9.2에서 확인
 - S의 Foreign Key는 NULL 값이 허용되는가?
-

STEP 6: Mapping of Multivalued attributes

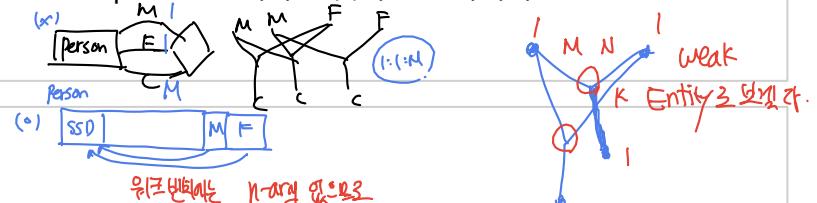
- relation S(primary key: K)의 multivalued attribute A에 대해 새로운 relation R을 만든다.
 - R에 K를 포함한다. R의 K는 S를 참조하는 foreign key가 된다.
 - K와 A의 조합이 R의 primary key가 된다.
 - Attribute는 소속된 Entity에 존재 종속적(existence dependency)이기 때문에 Foreign Key Option에서 (ON UPDATE, ON DELETE) CASCADE option을 선택하는 것이 일반적이다.
 - Example (Figure 9.3(d)): relation DEPT_LOCATIONS이 생성되었다.
-

- DEPT_LOCATIONS의 foreign key는?
- DEPT_LOCATIONS의 primary key는?

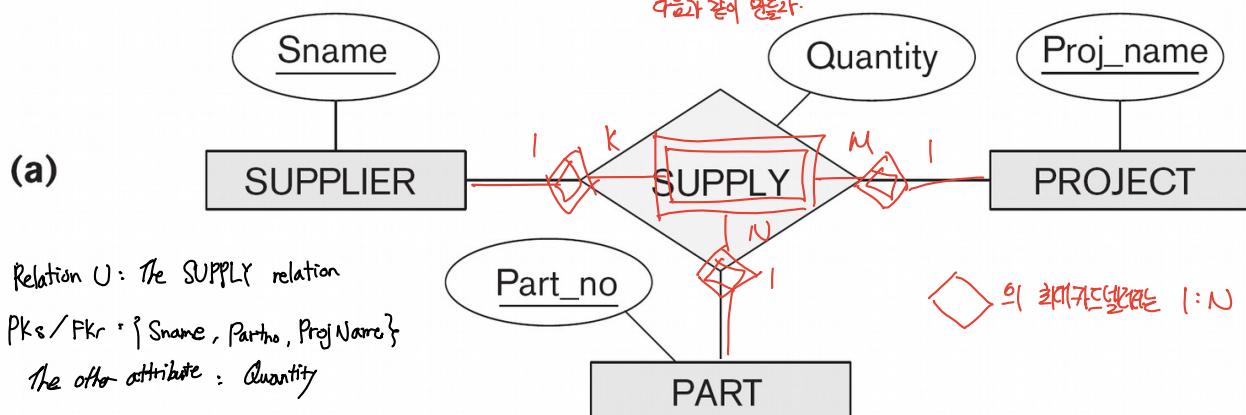
A: DEPARTMENT entity type Locations로부터 가져온 Dlocation
 FK: DEPARTMENT entity type의 PK를 가져온 Dnumber
 PK: ?Dlocation, Dnumber?

STEP 7: Mapping of N-ary Relationship Types

- n>2인 n-ary relationship type R에 대해 새로운 relationship relation S을 만든다.
- 참여하는 모든 entity type의 primary key는 S의 foreign key로 포함시킨다.
- 참여하는 모든 entity type의 primary key들의 조합으로 S의 primary key를 구성한다.
- N-ary relationship type의 attribute들을 S의 attribute로 포함시킨다.
- Relationship instance는 참여하는 Entity에 존재 종속적(existence dependency)이기 때문에 Foreign Key Option에서 (ON UPDATE, ON DELETE) CASCADE option을 선택하는 것이 일반적이다.
- 다음 슬라이드에 STEP 7을 적용해 보자.



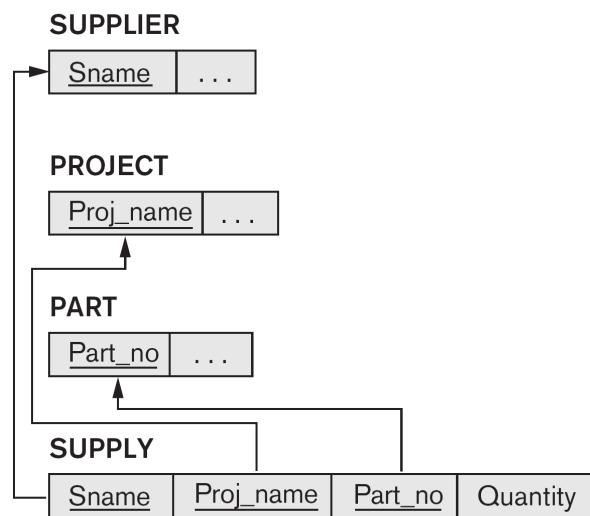
Ternary Relationship: Supply



Ternary Relationship: Supply

Figure 9.4

Mapping the n -ary relationship type SUPPLY from Figure 7.17(a).

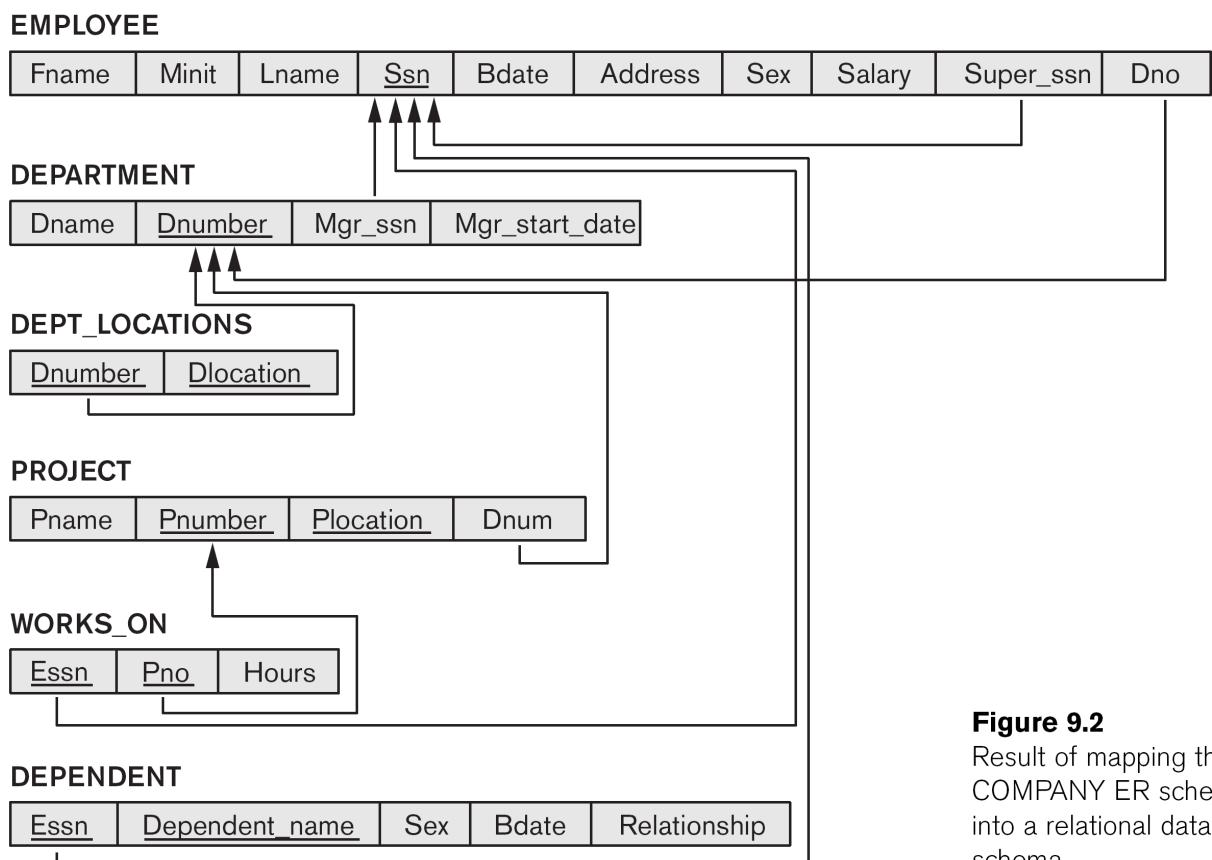


Correspondence between ER and Relational Model

Table 9.1 Correspondence between ER and Relational Models

ER MODEL	RELATIONAL MODEL
Entity type	<i>Entity</i> relation
1:1 or 1:N relationship type	Foreign key (or <i>relationship</i> relation)
M:N relationship type	<i>Relationship</i> relation and <i>two</i> foreign keys
<i>n</i> -ary relationship type	<i>Relationship</i> relation and <i>n</i> 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

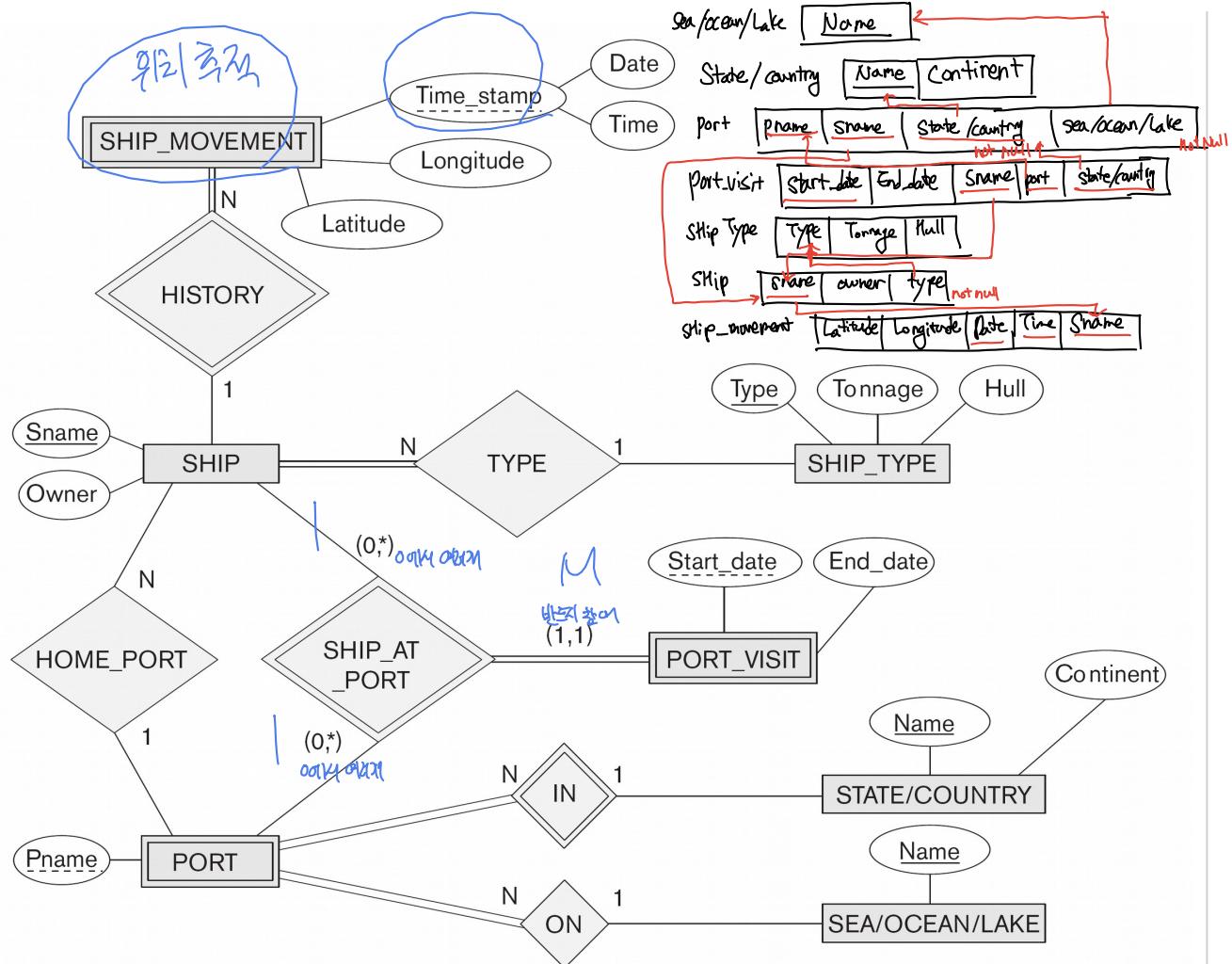
ER Model : Company

**Figure 9.2**

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

In-Class Exercise

- Draw a relational schema diagram from this ER diagram by applying STEP 0 through 7.

**Figure 9.8**

An ER schema for a SHIP_TRACKING database.

EER-Relation Mapping

EER을 relation mapping x

STEP 8: Mapping of Specialization or Generalization

- Option 8A: 다수의 테이블(Multiple relations)—superclass 와 subclasses 모두 테이블로 변환
 - For any specialization (total or partial, disjoint or overlapping)
- Option 8B: 다수의 테이블 (Multiple relations)—subclass 만 테이블로 변환 Drill down
 - Subclasses are total
 - Specialization has disjointedness constraint
- Option 8C: 하나의 type attribute를 포함하는 하나의 테이블로 변환
 - Type (or discriminating) attribute가 subclass를 결정한다.
- Option 8D: 여러 개의 type attribute를 포함하는 하나의 테이블로 변환
 - Subclasses are overlapping
 - Will also work for a disjoint specialization

수퍼클래스
테이블 합계
→ Flags을 통해 원본 테이블 Roll-up

Regular (strong) entity

- ER Schema의 일반적인 E의 모든 attribute는 relation R을 형성할 때 모두 포함한다.
- 복합 attribute의 경우 각각의 요소를 Simple attribute로 포함한다.
- fname과 lname은 포함하지만, name이라는 복합 attribute는 추가하지 않는다.
- E가 가진 attribute 중 하나를 Primary key로 설정한다.
- 만약 선택된 attribute가 복합 attribute라면, 이를 구성하는 Simple attribute set이 모두 primary key가 된다.

Weak entity

- weak entity가 포함하는 모든 attribute를 Relation R의 주체한다.
- weak entity의 partial key와 owner entity의 primary key를 통해 Relation R의 PK가 된다.
- 이 때 weak entity와 owner entity는 identifying relationship을 형성한다.
- Referential triggered action
 - owner 와 weak 사이에는 CASCADE option이 적용, UPDATE / DELETE operation
 - 이 때 CASCADE 하게 적용된다

Binary 1:1 relationship

- relation S와 T가 1:1 관계를 형성할 때의 경우이다.
- 양쪽 모두 Partial participating인 경우, S의 PK가 T의 FK가 될 수도 있고, 반대도 가능하다.
- 한 쪽이 total participation인 경우 partial participation인 relation의 PK를 total 쪽의 FK로 추가한다.

Binary M:N relationship

relation S와 W가 M:N 관계를 형성할 때의 경우이다.

PK와 FK로는 M:N 관계를 표현할 수 없다.

N-ary relationship type

N개의 entity에 대한 관계를 나타내는 relation S를 정의한다.

각 entity의 PK를 S의 FK로 설정한다.

S의 PK는 모든 FK의 집합으로 구성된다.

단, N개의 entity 중 cardinality가 1인 entity의 PK는 S의 PK를 구성하는 PK set의 원소로 포함하지 않도록 한다.

option 8A

- total / partial, disjoint / overlapping 모두 경우에 사용할 수 있는 방법



• Create a relation L for C

부모 entity에 대해서는 자신의 attribute와 key만을 포함한다.

$$\text{Attrs}(L) = \{\text{PK}, a_1, \dots, a_n\} \text{ and } \text{PK}(L) = k$$

For each subclass S_i , $1 \leq i \leq m$, create a relation L_i

- 각 entity의 경우 자신의 attribute와 key + 부모의 key attribute를 갖는다.

$$\cdot \text{Attrs}(L_i) = \{\text{PK}\} \cup \{\text{attributes of } S_i\}$$

$$\cdot \text{PK}(L_i) = k$$

inclusion dependency

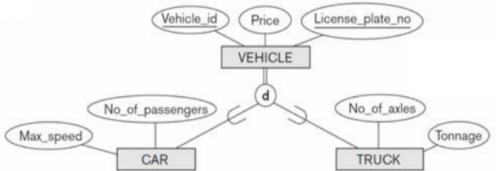
- 부모의 key 없이 자신의 key가 존재할 수 없다.

- $\exists : \text{key}_L \cap \text{key}_S$

- $\forall (\text{자식 entity}) \sqsubseteq \forall (\text{부모 entity})$

- $\exists (L_i) \sqsubseteq \exists (L)$

Option 8B



CAR

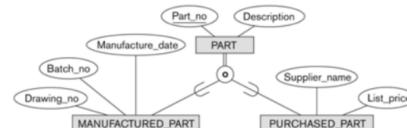
Vehicle_id	License_plate_no	Price	Max_speed	No_of_passengers
------------	------------------	-------	-----------	------------------

TRUCK

Vehicle_id	License_plate_no	Price	No_of_axles	Tonnage
------------	------------------	-------	-------------	---------

- 부모 relation은 생성하지 않음
- disjoint이고 total인 경우 잘 적용된다.
 - 만약 total이 아닌 경우(CAR, TRUCK이 아닌 VEHICLE이 있는 경우) : 정보를 표현할 수 없음
 - 만약 disjoint이 아닌 경우(CAR 이면서 TRUCK 인 경우) : 동일한 정보가 여러 subtype에 중복되어서 나오는 문제가 있음
 - 따라서 VEHICLE 전체의 구조를 보고 싶으면 OUTERJOIN(구조가 서로 다른 table을 JOIN) 또는 FULL OUTERJOIN을 CAR와 TRUCK에 적용해야 한다.
- Create a relation Li for each subclass Si, $1 \leq i \leq m$,
 - Attrs(Li) = { attributes of Si } \cup {k, a1,..., an}
 - PK(Li) = k

Option 8D

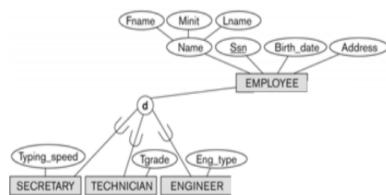


PART

Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price
---------	-------------	-------	------------	------------------	----------	-------	---------------	------------

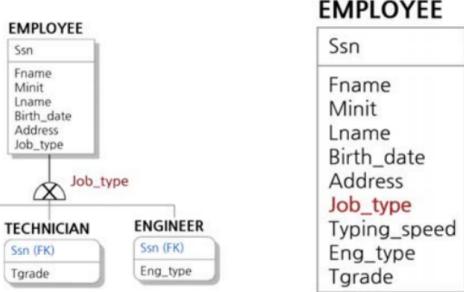
- subclass들이 overlap되는 경우 사용할 수 있는 방식
 - Create a single relation schema L
 - Attrs(L) = {k,a1,...,an} \cup {attributes of S1} $\cup \dots \cup$ {attributes of Sm} \cup {t1, t2, ..., tm}
 - PK(L) = k
 - each ti, $1 \leq i \leq m$, is a Boolean attribute indicating whether a tuple belongs to subclass Si,
 - subclass의 attribute가 많으면 주천하지 않는 방식
 - subclass의 attribute가 별로 없고, JOIN이 자주 사용되는 경우.(저장공간과 성능 사이의 상충 관계)

Option 8C



EMPLOYEE

Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type
-----	-------	-------	-------	------------	---------	----------	--------------	--------	----------



- 부모 relation만 생성한다.

- Create a single relation L
 - Attrs(L) = {k,a1,...,an} \cup {attributes of S1} $\cup \dots \cup$ {attributes of Sm} $\cup \{t\}$
 - t = type (or discriminating) attribute that indicates the subclass to which each tuple belongs, if any.
 - PK(L) = k

- t를 추가하기 때문에 disjoint인 상황에서만 사용할 수 있는 방법

- subtype 중 하나에만 속하기 때문에 굉장히 많은 NULL 값이 생성될 수 있는 방법

Regular (strong) entity

- ER Schema의 일반적인 Entity의 모든 attribute는 relation R을 형성할 때 포함된다.
- 복합 attribute의 경우 각각의 요소를 Simple attribute로 포함한다.
- name과 |name은 포함하지만, name이라는 복합 attribute는 추가하지 않는다.
- E가 가진 attribute 중 하나를 Primary key로 설정한다.
- 만약 선택된 attribute가 복합 attribute라면, 이를 구성하는 Simple attribute Set이 primary key가 된다.

Weak entity

- weak entity가 포함하는 모든 attribute를 Relation Key에 추가한다.
- weak entity의 partial key or owner entity의 primary key를除 Relation Key PK가 된다.
- 이 때 weak entity or Owner entity는 identifying relationship을 형성한다.
- Referential triggered action
 - owner or weak entity는 CASCADE optional 적용, UPDATE / DELETE operation
 - ① 대해서 CASCADE 하지 적용된다

Binary 1:1 relationship

- relation S와 T가 1:1 관계를 형성할 때의 경우이다.
- 양쪽 모두 Partial participation인 경우, S의 PK가 T의 FK가 될 수도 있고, 반대로 가능하다.
- 한 쪽이 total participation인 경우 partial participation인 relation의 PK를 다른 쪽의 FK로 추가한다.

Binary M:N relationship

relation S와 W가 M:N 관계를 형성할 때의 경우이다.

PK or FK로는 M:N 관계를 표현할 수 없다.

N-ary relationship type

N개의 entity에 대한 관계를 나타내는 relation S를 정의한다.

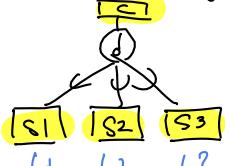
각 entity의 pk를 S의 FK로 설정한다.

S의 PK는 모든 PK의 합집합으로 구성된다.

단, N개의 entity 중 cardinality가 1인 entity의 PK는 S의 PK를 구성하는 PK set의 원소로 포함되어야 한다.

option 8A

- total / partial, disjoint / overlapping 모든 경우에 사용할 수 있는 방법



- Create a relation L for C

부모 entity에 대해서는 자신의 attribute와 key만을 포함한다.

$$\text{Attrs}(L) = \{k, a_1, \dots, a_n\} \text{ and } \text{PK}(L) = k$$

For each subclass S_i , $1 \leq i \leq m$, create a relation L_i

- 각각 entity의 경우 자신의 attribute와 key + 부모의 key attribute를 갖는다.

$$\text{Attrs}(L_i) = \{k\} \cup \{\text{attributes of } S_i\}$$

$$\text{PK}(L_i) = k$$

- inclusion dependency

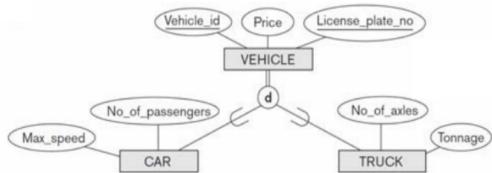
• 부모의 key 없이 자신의 key가 존재할 수 없다.

• π_{key} 포함.

$$\pi_{(자식 entity)} \subseteq \pi_{(부모 entity)}$$

$$\pi_{(L_i)} \subseteq \pi_{(L)}$$

Option 8B



CAR

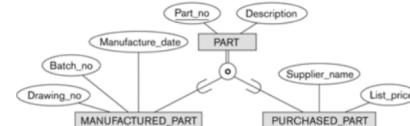
Vehicle_id	License_plate_no	Price	Max_speed	No_of_passengers
------------	------------------	-------	-----------	------------------

TRUCK

Vehicle_id	License_plate_no	Price	No_of_axles	Tonnage
------------	------------------	-------	-------------	---------

- 부모 relation은 생성하지 않음
- disjoint이고 total인 경우 잘 적용된다.
 - 만약 total이 아닌 경우(CAR, TRUCK이 아닌 VEHICLE이 있는 경우) : 정보를 표현할 수 없음
 - 만약 disjoint이 아닌 경우(CAR 이면서 TRUCK 인 경우) : 동일한 정보가 여러 subtype에 중복되어서 나오는 문제가 있음
 - 따라서 VEHICLE 전체의 구조를 보고 싶으면 OUTERJOIN(구조가 서로 다른 table을 JOIN) 또는 FULL OUTERJOIN을 CAR와 TRUCK에 적용해야 한다.
- Create a relation Li for each subclass Si, $1 \leq i \leq m$,
 - Attrs(Li) = { attributes of Si } \cup {k, a1,..., an}
 - PK(Li) = k

Option 8D

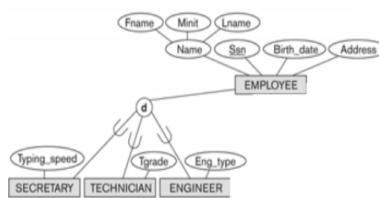


PART

Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price
---------	-------------	-------	------------	------------------	----------	-------	---------------	------------

- subclass들이 overlap되는 경우 사용할 수 있는 방식
- Create a single relation schema L
 - Attrs(L) = {k,a1,...,an} \cup {attributes of S1} $\cup \dots \cup$ {attributes of Sm} \cup {t1, t2, ..., tm}
 - PK(L) = k
 - each ti, $1 \leq i \leq m$, is a Boolean attribute indicating whether a tuple belongs to subclass Si,
- subclass의 attribute가 많으면 추천하지 않는 방식
- subclass의 attribute가 별로 없고, JOIN이 자주 사용되는 경우.(저장공간과 성능 사이의 상충 관계)

Option 8C



EMPLOYEE

Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type
-----	-------	-------	-------	------------	---------	----------	--------------	--------	----------

EMPLOYEE

EMPLOYEE

Ssn

Fname

Minit

Lname

Birth_date

Address

Job_type

Typing_speed

Tgrade

Eng_type

- 부 모 relation만 생성한다.

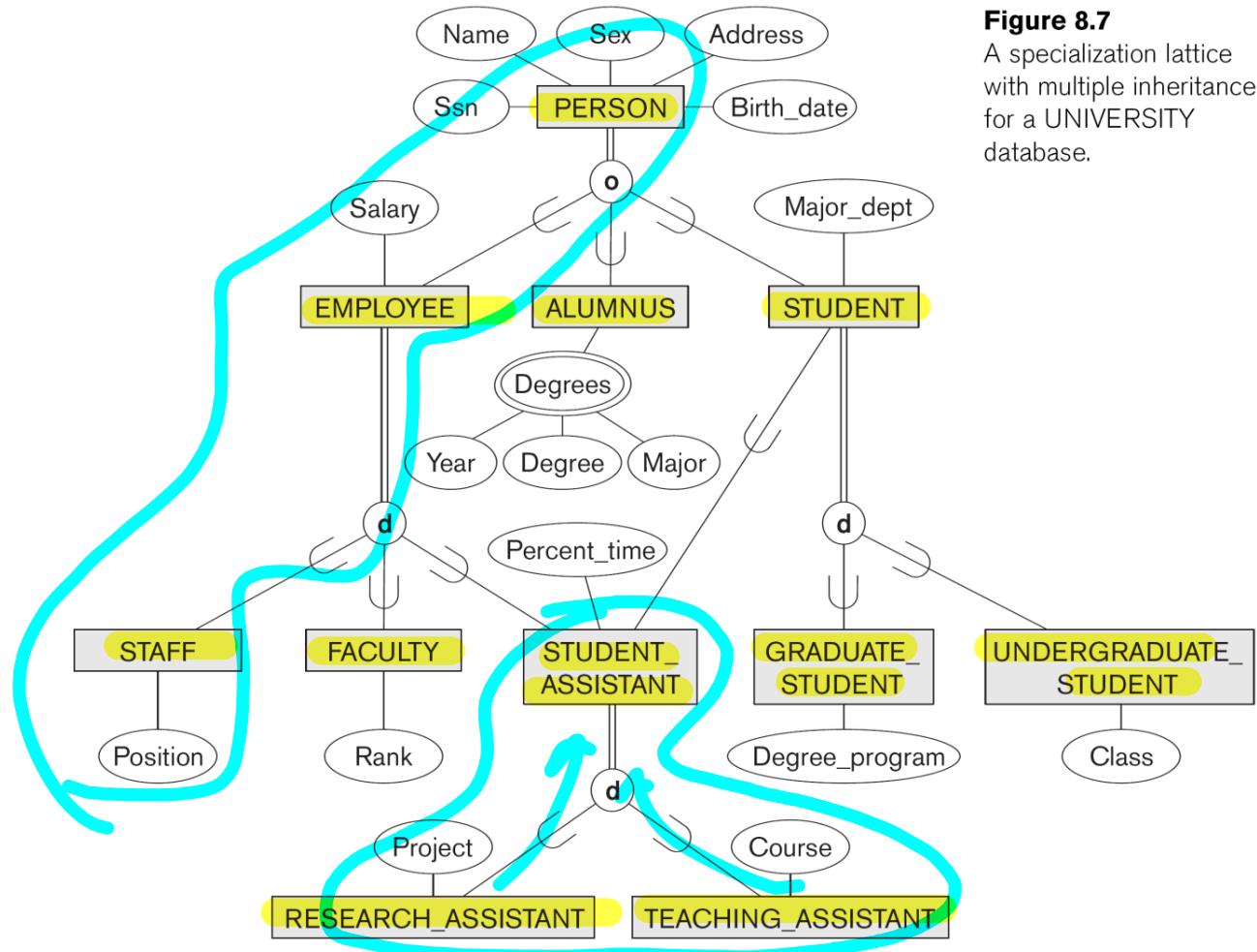
- Create a single relation L

- Attrs(L) = {k,a1,...,an} \cup {attributes of S1} $\cup \dots \cup$ {attributes of Sm} $\cup \{t\}$
- t = type (or discriminating) attribute that indicates the subclass to which each tuple belongs, if any.
- PK(L) = k

- t를 추가하기 때문에 disjoint인 상황에서만 사용할 수 있는 방법

- subtype 중 하나에만 속하기 때문에 굉장히 많은 NULL 값이 생성될 수 있는 방법

Specialization / Generalization Lattice Example (UNIVERSITY)

**Figure 8.7**

A specialization lattice with multiple inheritance for a UNIVERSITY database.

Mapping of Shared Subclasses (Multiple Inheritance)

- STEP 8 선택사항의 어떤 것도 shared subclass에 적용 가능

PERSON

<u>Ssn</u>	Name	Birth_date	Sex	Address
------------	------	------------	-----	---------

EMPLOYEE

<u>Ssn</u>	Salary	<u>Employee_type</u>	Position	Rank	Percent_time	<u>Ra_flag</u>	<u>Ta_flag</u>	Project	Course
------------	--------	----------------------	----------	------	--------------	----------------	----------------	---------	--------

2017.8.13. 2교시

ALUMNUS

ALUMNUS_DEGREES

<u>Ssn</u>	<u>Ssn</u>	<u>Year</u>	<u>Degree</u>	<u>Major</u>
------------	------------	-------------	---------------	--------------

STUDENT

<u>Ssn</u>	Major_dept	Grad_flag	Undergrad_flag	Degree_program	Class	Student_assist_flag
------------	------------	-----------	----------------	----------------	-------	---------------------

Figure 8.6

Mapping the EER specialization lattice in Figure 7.26 using multiple options.

STEP 9: Mapping of Union Types

- Superclass들은 각각 다른 key를 가진다.
- Subclass의 key를 새로운 대체 key attribute로 도입한다.
 - Surrogate key

Two categories (UNION types): OWNER, REGISTERED_VEHICLE

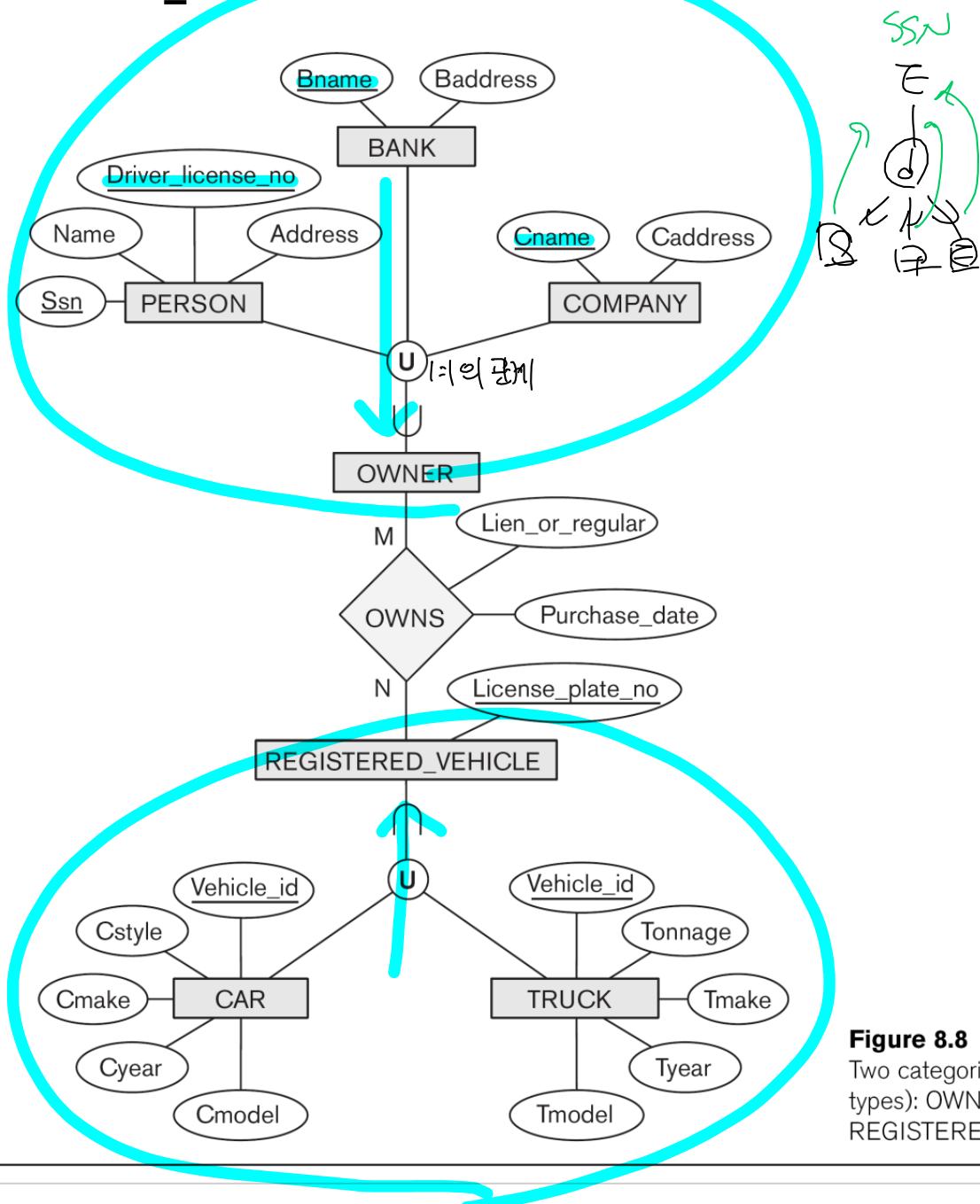


Figure 8.8
Two categories (union types): OWNER and REGISTERED_VEHICLE.

Two categories (UNION types): OWNER, REGISTERED_VEHICLE

Figure 8.7

Mapping the EER categories (union types) in Figure 7.26 to relations.

