

# 데이터베이스시스템

## 13. 데이터종속성과 정규화

나 홍 석    교수



13  
LESSON

# 데이터종속성과 정규화

# 학습 목표

- 1 회사데이터베이스를 SQL로 구현할 수 있다.
- 2 함수 종속성을 설명할 수 있다.
- 3 정규형의 종류를 이해하고 정규화 과정을 수행할 수 있다.

# 학습 내용

**1** 회사 데이터베이스 구현

---

**2** 이상과 함수종속

---

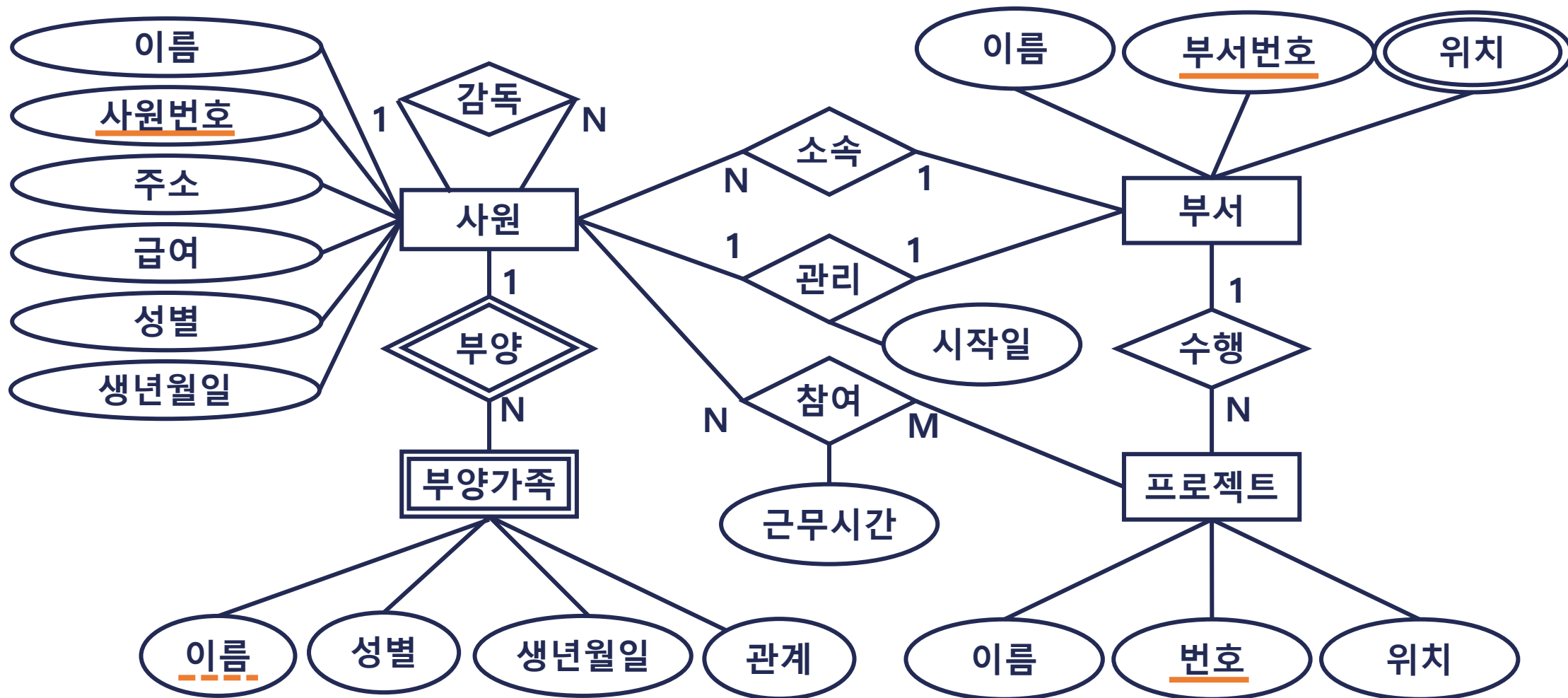
**3** 릴레이션 정규화

---

# Chapter 01 회사 데이터베이스 구현

# 1 사원관리 시스템의 릴레이션 스키마

## 1 사원관리 시스템 ERD



# 1    사원관리 시스템의 릴레이션 스키마

## 2    릴레이션 스키마

**사원**

<u>사원번호</u>	이름	주소	급여	성별	생년월일	감독자(FK)	부서번호(FK)
-------------	----	----	----	----	------	---------	----------

**부서**

<u>부서번호</u>	부서이름	부서관리자(FK)	시작일
-------------	------	-----------	-----

**부서위치**

<u>부서번호(FK)</u>	<u>위치</u>
-----------------	-----------

**프로젝트**

<u>프로젝트번호</u>	프로젝트이름	위치	수행부서(FK)
---------------	--------	----	----------

**참여**

<u>사원번호(FK)</u>	<u>프로젝트번호(FK)</u>	근무시간
-----------------	-------------------	------

**부양가족**

<u>사원번호</u>	<u>가족이름</u>	성별	생년월일	관계
-------------	-------------	----	------	----

## 2 테이블 정의서 작성

### 1 테이블 및 컬럼명의 영문화

No	한글단어	영문 전체 이름	영문약어
1	사원	Employee	EMP, E
2	부서	Department	DEPT, D
3	프로젝트	Project	P
4	부양가족	Dependent	DPNT
5	위치	Locations	LO
6	사원번호	Social Security Number	SSN
7	이름	Name	NN
8	번호	Number	NO
9	관리자	Manager	MGR
10	시간	Hours	HR



## 2 테이블 정의서 작성

### 2 물리적 데이터베이스 스키마

- ☑ 릴레이션으로부터 테이블명과 컬럼명을 도출하여 각각의 **테이블 구조를 정의**
- ☑ 논리적인 모델의 데이터 타입을 물리적인 DBMS의 특성과 성능을 고려하여 **최적의 데이터 타입을 선택**
- ☑ 기본키, 외래키 등 **제약조건 정의**

## 2 테이블 정의서 작성

### 3 사원 테이블

테이블명		employee										
테이블정의		K회사 전체 직원의 기본 정보를 저장										
No	컬럼명	한글명	타입	길이	NOT NULL	UK	PK	FK	참조테이블	참조컬럼	비고	
1	ename	이름	VARCHAR	15	Y							
2	ssn	사원고유번호	CHAR	9			Y					
3	address	주소	VARCHAR	30								
4	salary	급여	INT									
5	sex	성별	CHAR	1	Y						'M' or 'F'	
6	bdate	생년월일	DATETIME									
7	dno	부서번호	SMALLINT					Y	department	dnumber		
8	superssn	직속상사	CHAR	9				Y	employee	ssn		

## 2 테이블 정의서 작성

### 4 부서 테이블

테이블명		department										
테이블정의		K회사의 부서에 관한 정보를 저장										
No	컬럼명	한글명	타입	길이	NOT NULL	UK	PK	FK	참조테이블	참조컬럼	비고	
1	dnumber	부서번호	SMALLINT				Y					
2	dname	부서이름	VARCHAR	15	Y	Y						
3	mgrssn	관리자	CHAR	9				Y	employee	name		
4	mgrstartdate	시작일	DATETIME									

## 2 테이블 정의서 작성

### 5 부서위치 테이블

테이블명		DEPT_LOCATIONS									
테이블정의		K회사의 부서가 위치하고 있는 장소(한 부서가 여러 장소를 가질 수 있음)									
No	컬럼명	한글명	타입	길이	NOT NULL	UK	PK	FK	참조테이블	참조컬럼	비고
1	dnumber	부서번호	SMALLINT				Y	Y	department	dnumber	
2	dlocations	부서위치	VARCHAR	15			Y				

## 2 테이블 정의서 작성

### 6 프로젝트 테이블

테이블명		project									
테이블정의		프로젝트에 대한 정보를 저장									
No	컬럼명	한글명	타입	길이	NOT NULL	UK	PK	FK	참조테이블	참조컬럼	비고
1	pnumber	프로젝트번호	INT				Y				
2	pname	프로젝트명	VARCHAR	50	Y	Y					
3	plocation	수행위치	VARCHAR	15							
4	dnum	수행부서	SMALLINT		Y			Y	department	dnumber	

## 2 테이블 정의서 작성

7

참여 테이블

테이블명		works_on									
테이블정의		프로젝트에 참여하는 직원과 참여시간을 저장									
No	컬럼명	한글명	타입	길이	NOT NULL	UK	PK	FK	참조테이블	참조컬럼	비고
1	essn	참여사원번호	CHAR	9			Y	Y	employee	name	
2	pno	프로젝트번호	INT				Y	Y	project	pnumber	
3	hours	주당참여시간	DECIMAL	(3,1)							

## 2 테이블 정의서 작성

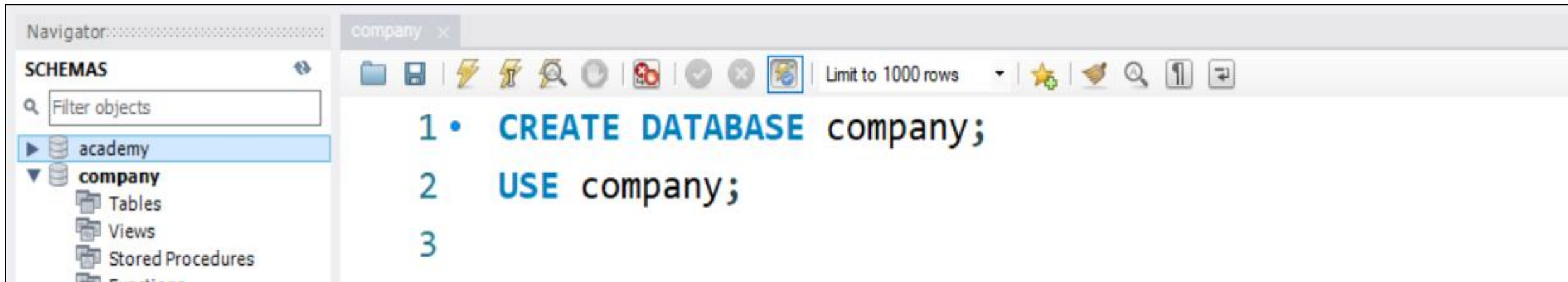
### 8 부양가족 테이블

테이블명		dependent									
테이블정의		직원의 부양가족 정보를 저장									
No	컬럼명	한글명	타입	길이	NOT NULL	UK	PK	FK	참조테이블	참조컬럼	비고
1	essn	사원번호	CHAR	9			Y	Y	employee	ssn	
2	dpntname	이름	VARCHAR	15	Y		Y				
3	sex	성별	CHAR	1							'M' or 'F'
4	bdate	생년월일	DATETIME								
5	relation	관계	VARCHAR	10	Y						

# 3 데이터베이스 구현

## 1 CREATE TABLE 명령어

- ✓ SQL의 **CREATE TABLE 명령어**를 사용하여 테이블 구조를 정의함
- ✓ 하나의 파일에 전체 테이블에 대한 정의를 SQL로 작성해 놓고, 이 파일을 불러들여서 필요한 부분들을 실행함
- ✓ 테이블 생성하기 전에 **데이터베이스를 먼저 생성함**





# 3 데이터베이스 구현

## 2 company.sql #1

### 사원 테이블 정의

```
4  -- 사원테이블 정의
5  CREATE TABLE employee
6  (
7      ename      VARCHAR(15) NOT NULL,
8      ssn        CHAR(9) PRIMARY KEY,
9      address    VARCHAR(30),
10     salary     INT,
11     sex         CHAR(1) NOT NULL CHECK (sex IN ('M', 'F')),
12     bdate       DATETIME,
13     dno         SMALLINT,
14     superssn    CHAR(9)
15 );
```

# 3 데이터베이스 구현

## 2 company.sql #2

- ☑ 부서 테이블 정의
- ☑ 부서위치 테이블 정의

```
17  -- 부서테이블 정의
18  CREATE TABLE department
19  (
20      dnumber    SMALLINT PRIMARY KEY,
21      dname      VARCHAR(15) NOT NULL UNIQUE,
22      mgrssn     CHAR(9),
23      mgrstartdate DATETIME
24  );
25
26  -- 부서위치테이블 정의
27  CREATE TABLE dept_locations
28  (
29      dnumber    SMALLINT,
30      dlocations  VARCHAR(15),
31      CONSTRAINT PRIMARY KEY (dnumber, dlocations)
32  );
```

# 3 데이터베이스 구현

## 2 company.sql #3

- ✓ 프로젝트 테이블 정의
- ✓ 프로젝트 참여 테이블 정의

```
34  -- 프로젝트테이블 정의
35  CREATE TABLE project
36  (
37      pnumber    INT PRIMARY KEY,
38      pname      VARCHAR(50) NOT NULL UNIQUE,
39      plocation  VARCHAR(15),
40      dnum       SMALLINT NOT NULL
41  );
42
43  -- (프로젝트)참여테이블 정의
44  CREATE TABLE works_on
45  (
46      essn    CHAR(9),
47      pno     INT,
48      hours   DECIMAL(3,1),
49      CONSTRAINT PRIMARY KEY (essn, pno)
50  );
```

# 3 데이터베이스 구현

## 2 company.sql #4

☑ 부양가족 테이블

정의

```
-- 부양가족테이블 정의
52
53 CREATE TABLE dependent
54 (
55     essn      CHAR(9),
56     dpntname  VARCHAR(15) NOT NULL,
57     sex       CHAR(1) CHECK (sex IN ('M', 'F')),
58     bdate     DATETIME,
59     relation  VARCHAR(10) NOT NULL,
60     CONSTRAINT PRIMARY KEY (essn, dpntname)
61 );
62
```

# 3 데이터베이스 구현

## 3 company.sql #5

☑ 사원, 부서, 부서위치 테이블의 외래키 정의

```
63  -- 외래키정의
64 • ALTER TABLE employee
65  ADD CONSTRAINT fk_emp_dept FOREIGN KEY (dno) REFERENCES department(dnumber);
66
67 • ALTER TABLE employee
68  ADD CONSTRAINT fk_emp_emp FOREIGN KEY (superssn) REFERENCES employee(ssn);
69
70 • ALTER TABLE department
71  ADD CONSTRAINT fk_dept_emp FOREIGN KEY (mgrssn) REFERENCES employee(ssn);
72
73 • ALTER TABLE dept_locations
74  ADD CONSTRAINT fk_d1_dept FOREIGN KEY (dnumber) REFERENCES department(dnumber);
75
```

# 3 데이터베이스 구현

## 3 company.sql #6

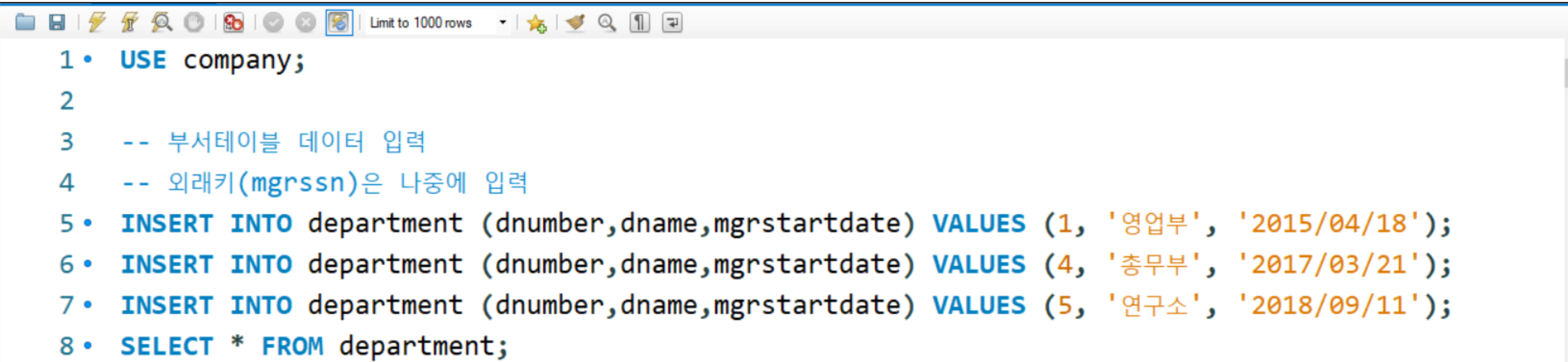
☑ 프로젝트, 참여, 부양가족 테이블의 외래키 정의

```
76 • ALTER TABLE project
77   ADD CONSTRAINT fk_pj_dept FOREIGN KEY (dnum) REFERENCES department(dnumber);
78
79 • ALTER TABLE works_on
80   ADD CONSTRAINT fk_wo_emp FOREIGN KEY (essn) REFERENCES employee(ssn);
81
82 • ALTER TABLE works_on
83   ADD CONSTRAINT fk_wo_pj FOREIGN KEY (pno) REFERENCES project(pnumber);
84
85 • ALTER TABLE DEPENDENT
86   ADD CONSTRAINT fk_dpnt_emp FOREIGN KEY (essn) REFERENCES employee(ssn);
87
```

# 4 초기 데이터 입력

## 1 company\_data.sql

- ✓ 처음 생성된 데이터베이스는 구조만 있으며, 여기에 실제 운영 데이터들을 변환, 입력함
- ✓ SQL의 **INSERT INTO 명령어**를 사용하여 초기 데이터를 입력함
- ✓ 외래키가 있는 경우 데이터 입력 순서에 주의해야 함



```
1 • USE company;
2
3  -- 부서테이블 데이터 입력
4  -- 외래키(mgrssn)은 나중에 입력
5 • INSERT INTO department (dnumber,dname,mgrstartdate) VALUES (1, '영업부', '2015/04/18');
6 • INSERT INTO department (dnumber,dname,mgrstartdate) VALUES (4, '총무부', '2017/03/21');
7 • INSERT INTO department (dnumber,dname,mgrstartdate) VALUES (5, '연구소', '2018/09/11');
8 • SELECT * FROM department;
```



```
10  -- 사원테이블 데이터 입력
11  -- 외래키에 유의해서 입력(직속상사가 먼저 입력되어야 함)
12 • INSERT INTO employee (ename,ssn,address,salary,sex,bdate,dno,super_ssn)
13  VALUES ('강명석','333445555','서울 송파구 풍납동',5000,'M','1965/12/08',5,NULL);
14 • INSERT INTO employee (ename,ssn,address,salary,sex,bdate,dno,super_ssn)
15  VALUES ('김유현','987987987','서울 강남구 압구정동',4800,'M','1968/12/12',4,NULL);
16 • INSERT INTO employee (ename,ssn,address,salary,sex,bdate,dno,super_ssn)
17  VALUES ('박현식','444889999','서울 강동구 암사동',4000,'M','1975/10/09',1,NULL);
18 • INSERT INTO employee (ename,ssn,address,salary,sex,bdate,dno,super_ssn)
19  VALUES ('한명숙','666884444','경기 용인시 풍덕천동',4300,'F','1972/07/05',5,'333445555');
20 • INSERT INTO employee (ename,ssn,address,salary,sex,bdate,dno,super_ssn)
21  VALUES ('김창현','123456789','서울 광진구 구의동',4000,'M','1975/01/09',5,'666884444');
22 • INSERT INTO employee (ename,ssn,address,salary,sex,bdate,dno,super_ssn)
23  VALUES ('남석훈','999887777','경기 구리시 인창동',4000,'M','1978/06/19',4,'987987987');
24 • INSERT INTO employee (ename,ssn,address,salary,sex,bdate,dno,super_ssn)
25  VALUES ('전명희','987654321','서울 종로구 안국동',3500,'F','1981/06/30',4,'987987987');
26 • INSERT INTO employee (ename,ssn,address,salary,sex,bdate,dno,super_ssn)
27  VALUES ('이혜진','453453453','경기 용인시 동천동',4000,'F','1975/09/12',5,'666884444');
28 • INSERT INTO employee (ename,ssn,address,salary,sex,bdate,dno,super_ssn)
29  VALUES ('이지은','888665555','서울 강동구 고덕동',3800,'F','1977/12/11',1,'444889999');
30 • SELECT * FROM employee;
```



# 4 초기 데이터 입력

## 2 입력 데이터 #1

SELECT \* FROM employee;

Result Grid     Filter Rows: <input type="text"/>   Edit:      Export/Import:     Wrap Cell Content:								
	ename	ssn	address	salary	sex	bdate	dno	superssn
▶	김창현	123456789	서울 광진구 구의동	4000	M	1975-01-09 00:00:00	5	666884444
	강명석	333445555	서울 송파구 풍납동	5000	M	1965-12-08 00:00:00	5	NULL
	박현식	444889999	서울 강동구 암사동	4000	M	1975-10-09 00:00:00	1	NULL
	이혜진	453453453	경기 용인시 동천동	4000	F	1975-09-12 00:00:00	5	666884444
	한명숙	666884444	경기 용인시 풍덕천동	4300	F	1972-07-05 00:00:00	5	333445555
	이지은	888665555	서울 강동구 고덕동	3800	F			
	전명희	987654321	서울 종로구 안국동	3500	F			
	김유현	987987987	서울 강남구 압구정동	4800	M			
	남석훈	999887777	경기 구리시 인창동	4000	M			
*	NULL	NULL	NULL	NULL	NULL			

SELECT \* FROM dependent;

Result Grid     Filter Rows: <input type="text"/>   Edit:      Export/Import:					
	essn	dpntname	sex	bdate	relation
▶	333445555	강동진	M	1995-04-05 00:00:00	아들
	333445555	강수진	F	1998-05-05 00:00:00	딸
	333445555	김명희	F	1967-10-13 00:00:00	배우자
	444889999	한명숙	F	1975-04-06 00:00:00	배우자
	987987987	김다래	F	2000-02-21 00:00:00	딸
	987987987	남희정	F	1970-07-09 00:00:00	배우자
*	NULL	NULL	NULL	NULL	NULL

SELECT \* FROM dependent;

# 4 초기 데이터 입력

## 2 입력된 데이터 #2

SELECT \* FROM department;

	dnumber	dname	mgrssn	mgrstartdate
▶	1	영업부	333445555	2015-04-18 00:00:00
	4	총무부	987987987	2017-03-21 00:00:00
	5	연구소	444889999	2018-09-11 00:00:00
*	NULL	NULL	NULL	NULL

SELECT \* FROM project;

	pnumber	pname	plocation	dnum
▶	10	차세대 정보 시스템 도입	서울	5
	20	효과적 조직 개편안 연구	서울	4
	30	온라인 마케팅 강화를 위한 전략 수립	부산	1
	40	보안 강화를 위한 인프라 구축	대전	5
*	NULL	NULL	NULL	NULL

SELECT \* FROM dept\_locations;

	dnumber	dlocations
▶	1	대구
	1	부산
	1	서울
	4	서울
	4	인천
	5	대전
*	NULL	NULL

SELECT \*  
FROM works\_on;

	essn	pno	hours
▶	123456789	10	25.0
	333445555	30	20.0
	444889999	10	20.0
	444889999	40	10.5
	453453453	10	20.0
	666884444	40	20.0
	888665555	30	25.0
	987654321	10	5.0
	987654321	40	5.0
	987987987	10	5.5
	987987987	20	20.0
	987987987	40	5.5
	999887777	10	5.0
	999887777	20	10.5
*	NULL	NULL	NULL



실습 영상

# 회사 데이터베이스 구현

## Chapter 02 이상과 함수 종속

# 1 데이터 중복과 이상현상

1

사원관리 시스템 릴레이션 스키마

**사원**

<u>사원번호</u>	이름	주소	급여	성별	생년월일	감독자(FK)	부서번호(FK)
-------------	----	----	----	----	------	---------	----------

**부서**

<u>부서번호</u>	부서이름	부서관리자(FK)	시작일
-------------	------	-----------	-----

**부서위치**

<u>부서번호(FK)</u>	<u>위치</u>
-----------------	-----------

**프로젝트**

<u>프로젝트번호</u>	프로젝트이름	위치	수행부서(FK)
---------------	--------	----	----------

**참여**

<u>사원번호(FK)</u>	<u>프로젝트번호(FK)</u>	근무시간
-----------------	-------------------	------

**부양가족**

<u>사원번호</u>	<u>가족이름</u>	성별	생년월일	관계
-------------	-------------	----	------	----

# 1 데이터 중복과 이상현상

## 2 데이터 중복이 발생하는 릴레이션 스키마 예

사원부서

사원명	<u>사원번호</u>	생년월일	주소	소속부서 번호	부서명	부서 책임자
김동길	E0001	1975/01/09	서울 광진구 구의동	D5	연구소	권현식
권현식	E0002	1965/12/08	서울 송파구 풍납동	D5	연구소	권현식
김미순	E0003	1978/06/19	경기 구리시 인창동	D4	총무부	안준태
박성호	E0004	1981/06/30	서울 종로구 안국동	D4	총무부	안준태
이지선	E0005	1972/07/05	경기 용인시 풍덕천동	D5	연구소	권현식
주봉석	E0006	1975/09/12	경기 용인시 동천동	D5	연구소	권현식
안준태	E0007	1968/12/12	서울 강남구 압구정동	D4	총무부	안준태
유명희	E0008	1977/12/11	서울 강동구 고덕동	D1	영업부	한상진
한상진	E0009	1975/10/09	서울 강동구 암사동	D1	영업부	한상진

# 1 데이터 중복과 이상현상

## 3 삽입이상(Insertion Anomaly)

- ☑ 사원부서 릴레이션에 새로운 사원 튜플을 삽입하려면 반드시 사원이 일하는 부서에 대한 정보를 명시하거나, 널 값을 명시해야 함
- ☑ 또한, 사원이 없는 새 부서를 사원부서 릴레이션에 삽입하기 어려움

강남길	E0010	1970/12/17	서울 송파구 가락동	D5	연구소	권현식
-----	-------	------------	------------	----	-----	-----

- ☑ 즉, 어떤 데이터를 삽입 하려고 할 때 불필요하고 원하지 않는 데이터도 함께 삽입해야만 하고 그렇지 않으면 삽입이 되지 않는 현상을 삽입 이상(insertion anomaly)라고 함

# 1 데이터 중복과 이상현상

## 4 삭제이상(Deletion Anomaly)

- ☑ 만약 어떤 부서에서 일하는 유일한 사원에 대한 튜플을 삭제하게 되면, 그 부서에 대한 정보도 데이터베이스에서 없어지게 됨

유명희	E0008	1977/12/11	서울 강동구 고덕동	D1	영업부	한상진
한상진	E0009	1975/10/09	서울 강동구 암사동	D1	영업부	한상진

- ☑ 한 튜플을 삭제함으로써 유지해야 될 정보까지도 삭제되는 현상이 일어나게 되어 정보의 손실이 발생하게 되는데 이러한 현상을  
삭제 이상(deletion anomaly)라고 함



# 1 데이터 중복과 이상현상

## 5 수정이상(Modification Anomaly)

- ☑ D5번 부서의 관리자를 변경하면 그 부서에서 일하는 모든 사원 튜플에 대해서 갱신해야 함
- ☑ 만일 일부 튜플만 변경시킨다면 데이터베이스의 일관성이 없어짐

김미순	E0003	1978/06/19	경기 구리시 인창동	D4	총무부	한상진
박성호	E0004	1981/06/30	서울 종로구 안국동	D4	총무부	안준태

- ☑ 중복된 튜플들 중에서 일부 튜플의 애트리뷰트 값만을 갱신시킴으로써 정보의 모순성(inconsistency)이 생기는 현상을 수정 이상(Modification anomaly) 이라 함

## 2 해결방안

### 1 이상의 원인과 해결방안

#### 이상의 원인

�트리뷰트들 간에 존재하는 여러 종속 관계를 하나의 릴레이션에 표현



#### 이상의 해결

�트리뷰트들 간의 종속관계를 분석하여 여러 개의 릴레이션으로 분해(decomposition)

➡ 정규화(normalization)

## 2 해결방안

### 2 스키마 변환

- ☑ 애트리뷰트들과 이들의 제약 조건(종속성)들을 수집
- ☑ 수집된 결과를 명시된 제약 조건에 따라 여러 개의 릴레이션으로 분할

➔ 스키마변환 (schema transformation)

1. 정보의 무손실

2. 데이터의 중복성 감소

3. 분리의 원칙

# 3 함수 종속성

## 1 정의

- ☑ 데이터 애트리뷰트들의 의미와 애트리뷰트들 간의 상호 관계로부터 유도되는 제약조건의 일종
- ☑ '애트리뷰트들의 집합  $X$ 의 값이 애트리뷰트들의 집합  $Y$ 의 값을 유일하게(unique) 결정한다면  $X$ 는  $Y$ 를 함수적으로 결정한다(functionally determines)'고 정의

# 3 함수 종속성

## 2 의미

- ☑ 함수 종속은 데이터의 의미(data semantics)를 표현함
- ☑ 학년이 학번에 종속되어 있다는 것은 학번이 지정되면 학년 값이 유일하게 결정되어야 함
- ☑ 만일, 데이터베이스에서 어느 한 학생의 학년이 두 가지 값으로 나타나 있다면, 이 데이터베이스는 잘못된 데이터를 가지고 있다는 것을 의미함

# 3 함수 종속성

## 3 예제 #1

☑️ 사원번호는 사원명을 결정한다.

사원번호 → 사원명

☑️ 과제번호는 과제명과 수행장소를 결정한다.

과제번호 → {과제명, 수행장소}

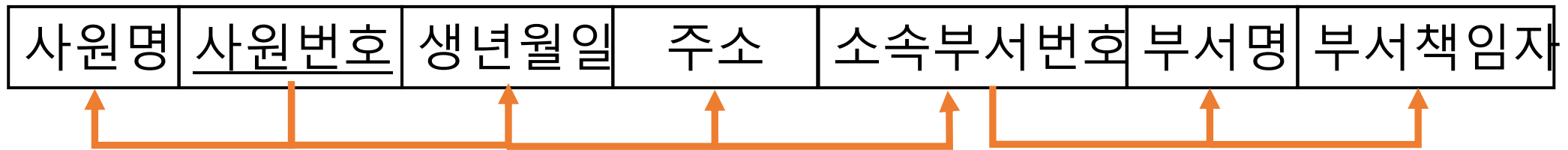
☑️ 사원의 사원번호와 프로젝트의 과제번호는 그 사원이 일주일 동안 그 프로젝트를 위해서 일하는 시간을 결정한다.

{사원번호, 과제번호} → 참여시간

### 3 함수 종속성

#### 3 예제 #2

☑ 사원부서 릴레이션의 함수 종속성



### 3 함수 종속성

#### 4 완전 함수 종속과 부분 함수 종속

☑ 복합 애트리뷰트  $X$ 에 대하여  $R.X \rightarrow R.Y$ 가 성립할 때  
다음과 같이 정의

완전 함수 종속  
Full functional dependency

$X' \subset X$  이고  $R.X'$   
 $\rightarrow R.Y$  를 만족하는  
애트리뷰트  $X'$ 이 존재하지 않음

VS

부분 함수 종속  
Partial functional dependency

$X' \subset X$  이고  $R.X'$   
 $\rightarrow R.Y$  를 만족하는  
애트리뷰트  $X'$ 이 존재함



## Chapter 03 릴레이션 정규화

# 1 정규형(정규화)

## 1 정규화(Normalization) 개념

### 1 무손실 표현

- 같은 의미의 정보 유지, 그러나 더 바람직한 구조
- 

### 2 데이터 중복성 감소

### 3 분리의 원칙

- 독립적인 관계는 별개의 릴레이션으로 표현
  - 릴레이션 각각에 대해 독립적인 조작 기능
-

# 1 정규형(정규화)

## 2 정규화(Normalization) 원칙

### ☑ 기본 아이디어

: 서로 독립적인 관계는 별개의 릴레이션으로 표현해야 함

Normal Form

어떤 일련의 제약 조건을  
만족하는 릴레이션

VS

Normalization

정규화 = 스키마 변환  
( $S \rightarrow S'$ )

## 2 제1정규형

### 1 정의(제1정규형, 1NF : First Normal Form)

- ☑ 릴레이션에서 애트리뷰트의 값은  
도메인에 속하는 **원자값(Atomic Value)**이어야 한다는 제약을 의미함
- ☑ 복합 애트리뷰트(composite attribute), 다치 애트리뷰트 (multivalued attribute),  
중첩 릴레이션(nested relation) 등  
**비원자적(non-atomic) 애트리뷰트를 허용하지 않음**

## 2 제1정규형

### 2 스키마 변환

- ☑ 제1정규형을 만족시키기 위해서 다중 값을 갖는 릴레이션을 단일 값을 갖도록 변환해야 하며, 3가지 변환 방법이 있음

부서	부서번호	부서명	부서책임자	부서위치
	D1	영업부	한상진	{서울, 부산, 대구}
	D4	총무부	안준태	{서울, 인천}
	D5	연구소	권현식	대전

## 2 제1정규형

### 3 변환 방법 #1

☑ 기본키를 확장해서 튜플이 중복될 수 있게 함

부서	부서번호	부서명	부서책임자	부서위치
	D1	영업부	한상진	서울
	D1	영업부	한상진	부산
	D1	영업부	한상진	대구
	D4	총무부	안준태	서울
	D4	총무부	안준태	인천
	D5	연구소	권현식	대전

## 2 제1정규형

### 3 변환 방법 #2

☑ 애트리뷰트가 가질 수 있는 최대수 만큼 애트리뷰트를 확장함

부서	부서번호	부서명	부서책임자	부서위치1	부서위치2	부서위치3
	D1	영업부	한상진	서울	부산	대구
	D4	총무부	안준태	서울	인천	
	D5	연구소	권현식	대전		

## 2 제1정규형

### 3 변환 방법 #3

☑ 다중값을 갖는 애트리뷰트를 별도의 릴레이션으로 분리함

부서	부서번호	부서명	부서책임자
	D1	영업부	한상진
	D4	총무부	안준태
	D5	연구소	권현식

부서위치	부서번호	부서위치
	D1	서울
	D1	부산
	D1	대구
	D4	서울
	D4	인천
	D5	대전



### 3 제2정규형

#### 1 제1정규형의 이상(anomaly)

☒ 제1정규형

다중 값을 갖는 애트리뷰트를 제거하는 것

☒ 제1정규형을 만족하더라도

이상(anomaly)이 존재하는 릴레이션 스키마가 있을 수 있음

### 3 제2정규형

#### 사원\_프로젝트

ESSN	PNO	HOURS	ENAME	PNAME	PLOCATION
444889999	10	20.0	박현식	차세대 정보 시스템 도입	서울
444889999	40	10.5	박현식	보안 강화를 위한 인프라 구축	대전
987987987	10	5.5	김유헌	차세대 정보 시스템 도입	서울
987987987	20	20.0	김유헌	효과적 조직 개편안 연구	서울
987987987	40	5.5	김유헌	보안 강화를 위한 인프라 구축	대전
333445555	30	20.0	강명석	온라인 마케팅 강화를 위한 전략 수립	부산
123456789	10	25.0	김창헌	차세대 정보 시스템 도입	서울
999887777	20	10.5	남석훈	효과적 조직 개편안 연구	서울
999887777	10	5.0	남석훈	차세대 정보 시스템 도입	서울
987654321	40	5.0	전명희	보안 강화를 위한 인프라 구축	대전
666884444	40	20.0	한명숙	보안 강화를 위한 인프라 구축	대전
453453453	10	20.0	이혜진	차세대 정보 시스템 도입	서울
888665555	30	25.0	이지은	온라인 마케팅 강화를 위한 전략 수립	부산
987654321	10	5.0	전명희	차세대 정보 시스템 도입	서울

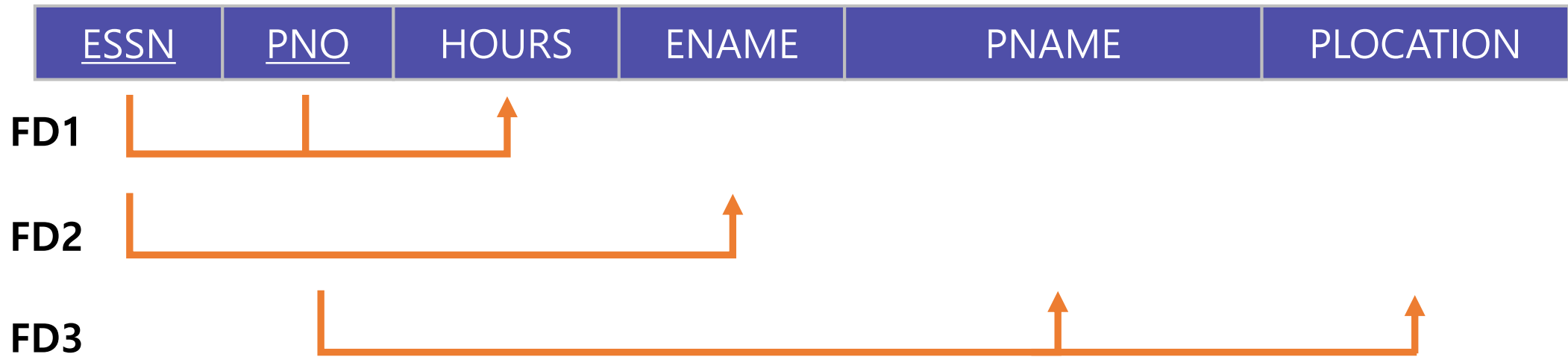
### 3 제2정규형

2

#### 함수 종속성 분석

☑ 기본키에 부분 함수종속된 애트리뷰트가 존재함

→ PNAME, PLOCATION, ENAME



### 3 제2정규형

#### 3 이상(anomaly)

##### 삽입 이상

새로운 프로젝트를 삽입할 경우  
참여하는 직원을 1명 이상 등록시키지 않는 한 삽입 불가능

##### 삭제 이상

사원 강명석, 이지은이 사퇴하는 경우 20번 프로젝트의 정보도 삭제됨

##### 수정 이상

20번 프로젝트의 이름을 '정보시스템 고도화'로 변경하는 경우  
6개 튜플의 값을 모두 변경해야 함

### 3 제2정규형

#### 4 해결방안

- ☑️ 프로젝션에 의한 릴레이션의 분해(부분함수종속을 제거함)

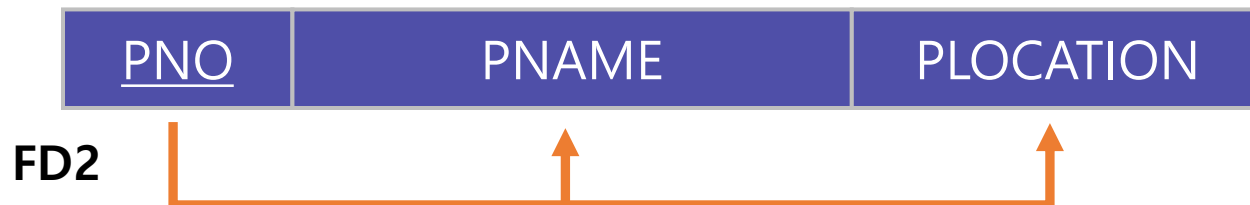
사원\_프로젝트1



사원\_프로젝트2



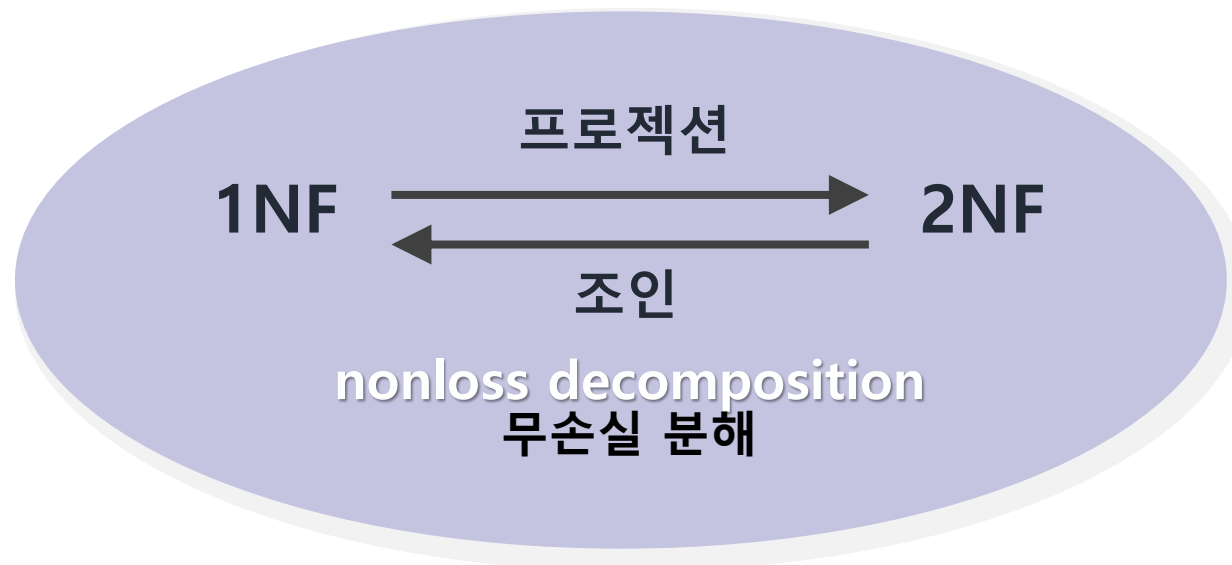
사원\_프로젝트3



### 3 제2정규형

#### 5 정의(제2정규형, 2NF : Second Normal Form)

- ☑ 1NF이고, 키에 속하지 않는 모든 애트리뷰트들이 기본키에 완전 함수 종속
- ☑ 프로젝션하여 분해된 릴레이션들은  
자연 조인을 통해 원래의 릴레이션으로 복귀 가능



# 4 제3정규형

1

## 제2정규형의 이상

### ☒ 제2정규형

기본키에 부분함수 종속이 존재하지 않도록 하는 제약조건

### ☒ 제2정규형을 만족하더라도

이상(anomaly)이 존재하는 릴레이션 스키마가 있을 수 있음

## 4 제3정규형

### 사원부서

사원명	<u>사원번호</u>	생년월일	주소	소속부서번호	부서명	부서책임자
김동길	E0001	1975/01/09	서울 광진구 구의동	D5	연구소	권현식
권현식	E0002	1965/12/08	서울 송파구 풍납동	D5	연구소	권현식
김미순	E0003	1978/06/19	경기 구리시 인창동	D4	총무부	안준태
박성호	E0004	1981/06/30	서울 종로구 안국동	D4	총무부	안준태
이지선	E0005	1972/07/05	경기 용인시 풍덕천동	D5	연구소	권현식
주봉석	E0006	1975/09/12	경기 용인시 동천동	D5	연구소	권현식
안준태	E0007	1968/12/12	서울 강남구 압구정동	D4	총무부	안준태
유명희	E0008	1977/12/11	서울 강동구 고덕동	D1	영업부	한상진
한상진	E0009	1975/10/09	서울 강동구 암사동	D1	영업부	한상진



## 4 제3정규형

### 2 함수 종속성 분석

☑ 이행적 함수 종속이 존재

사원명 → 소속부서번호 → 부서명, 부서책임자



## 4 제3정규형

### 3 이상(anomaly)

#### 삽입 이상

사원부서 릴레이션에 새로운 사원 튜플을 삽입하려면 반드시 사원이 일하는 부서에 대한 정보를 명시하거나, 널 값을 명시해야 함

#### 삭제 이상

만약 어떤 부서에서 일하는 유일한 사원에 대한 튜플을 삭제하게 되면, 그 부서에 대한 정보도 데이터베이스에서 없어지게 됨

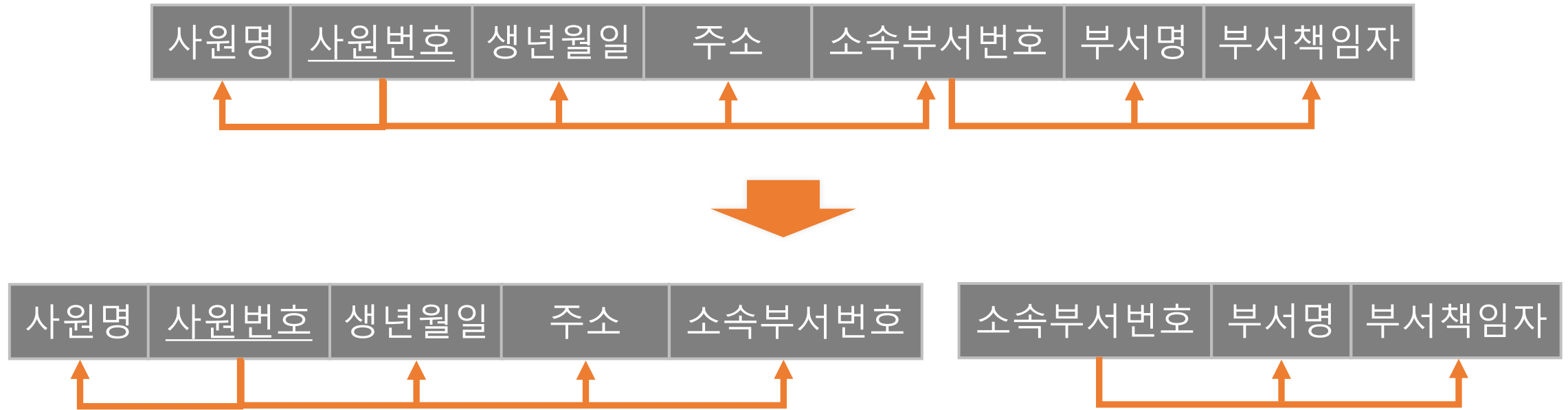
#### 수정 이상

D5번 부서의 관리자를 변경하면 그 부서에서 일하는 모든 사원 튜플에 대해서 갱신해야 함

## 4 제3정규형

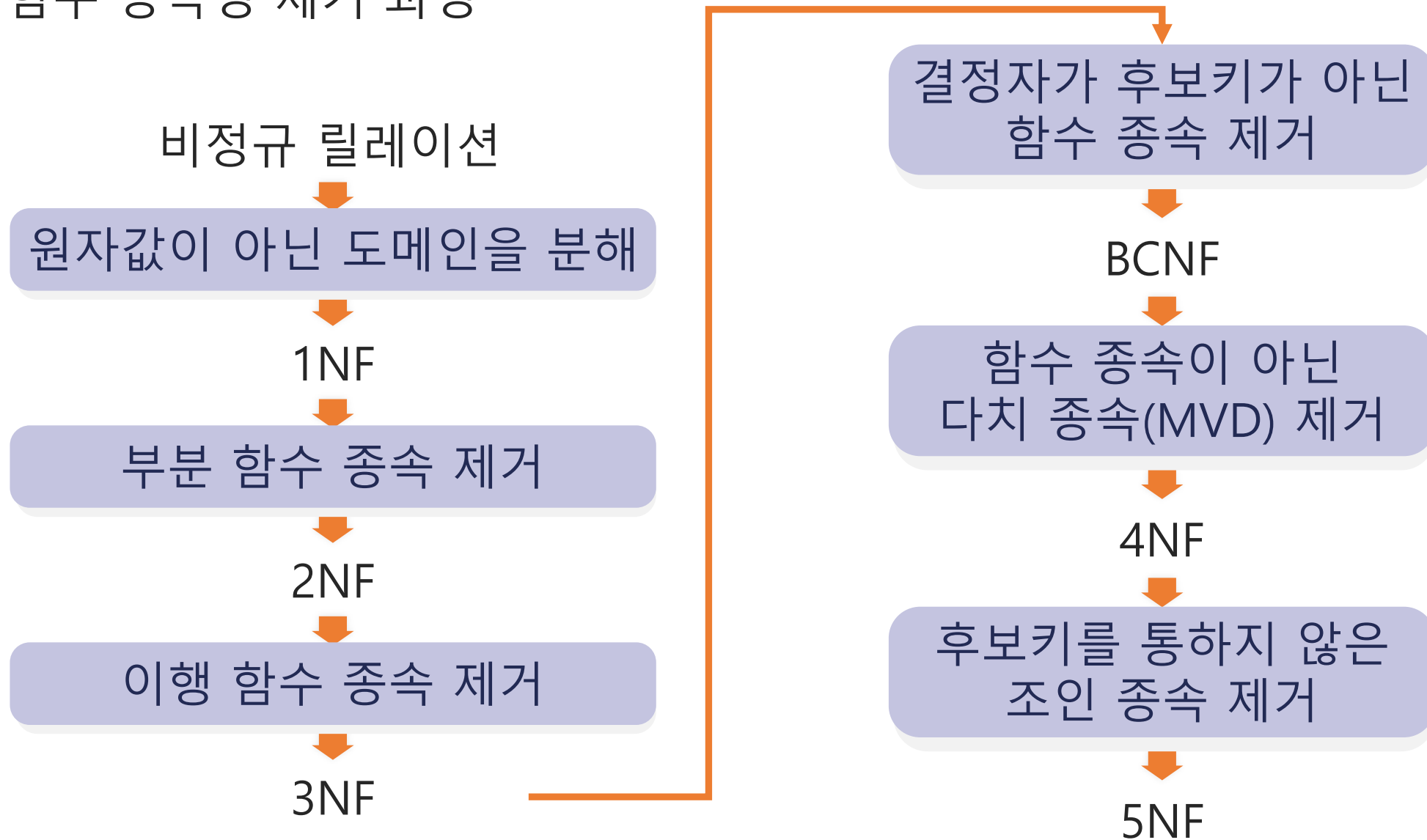
### 4 해결방안

☑ 프로젝트에 의한 릴레이션의 분해(부분함수종속을 제거함)



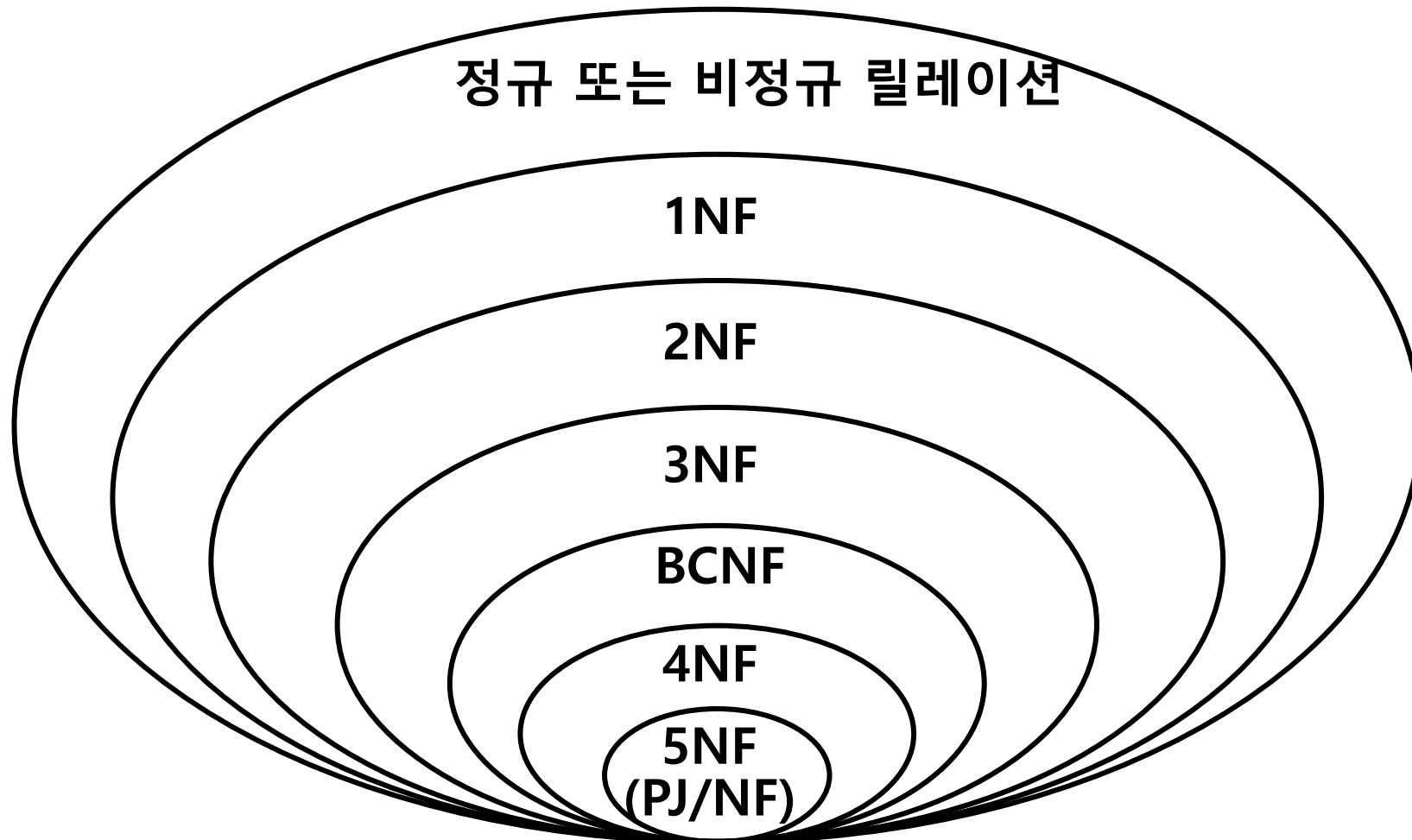
# 5 정규형들 간의 관계

## 1 함수 종속성 제거 과정



# 5 정규형들 간의 관계

## 2 정규형 들 간의 포함관계



## 5 정규형들 간의 관계

### 3 BCNF(보이스/코드 정규형, Boyce/Codd Normal Form)

- ☑ 3NF는 키가 아닌 애트리뷰트 값의 갱신 시 불필요한 부작용(이상)이 발생하지 않으며, 모든 이진 릴레이션은 3NF에 속함
- ☑ 하지만, ① 복수의 후보키를 가지고 있고, ② 후보키들이 복합 애트리뷰트들로 구성되며, ③ 후보키들이 서로 중첩되는 경우에는 적용이 불가능함
- ☑ 릴레이션 R의 모든 결정자(determinant)가 후보키(candidate key) 이면 릴레이션 R은 보이스/코드 정규형임
- ☑ 릴레이션 R이 BCNF에 속하면 R은 제1, 제2, 제3 정규형에 속함

# 학습 정리

## 회사 데이터베이스 구현

SELECT \* FROM employee;

Result Grid			Filter Rows:	<input type="text"/>	Edit:				Export/Import:			Wrap Cell Content:	
	ename	ssn	address	salary	sex	bdate	dno	superssn					
▶	김창헌	123456789	서울 광진구 구의동	4000	M	1975-01-09 00:00:00	5	666884444					
	강명석	333445555	서울 송파구 풍납동	5000	M	1965-12-08 00:00:00	5	NULL					
	박현식	444889999	서울 강동구 암사동	4000	M	1975-10-09 00:00:00	1	NULL					
	이혜진	453453453	경기 용인시 동천동	4000	F	1975-09-12 00:00:00	5	666884444					
	한명숙	666884444	경기 용인시 풍덕천동	4300	F	1972-07-05 00:00:00	5	333445555					
	이지은	888665555	서울 강	Result Grid   Filter Rows: <input type="text"/> Edit:    Export/Import:									
	전명희	987654321	서울 종	essn	dpntname	sex	bdate	relation					
	김유현	987987987	서울 강	▶ 333445555	강동진	M	1995-04-05 00:00:00	아들					
	남석훈	999887777	경기 구	333445555	강수진	F	1998-05-05 00:00:00	딸					
*	NULL	NULL	NULL	333445555	김명희	F	1967-10-13 00:00:00	배우자					

SELECT \* FROM dependent;

# 학습 정리



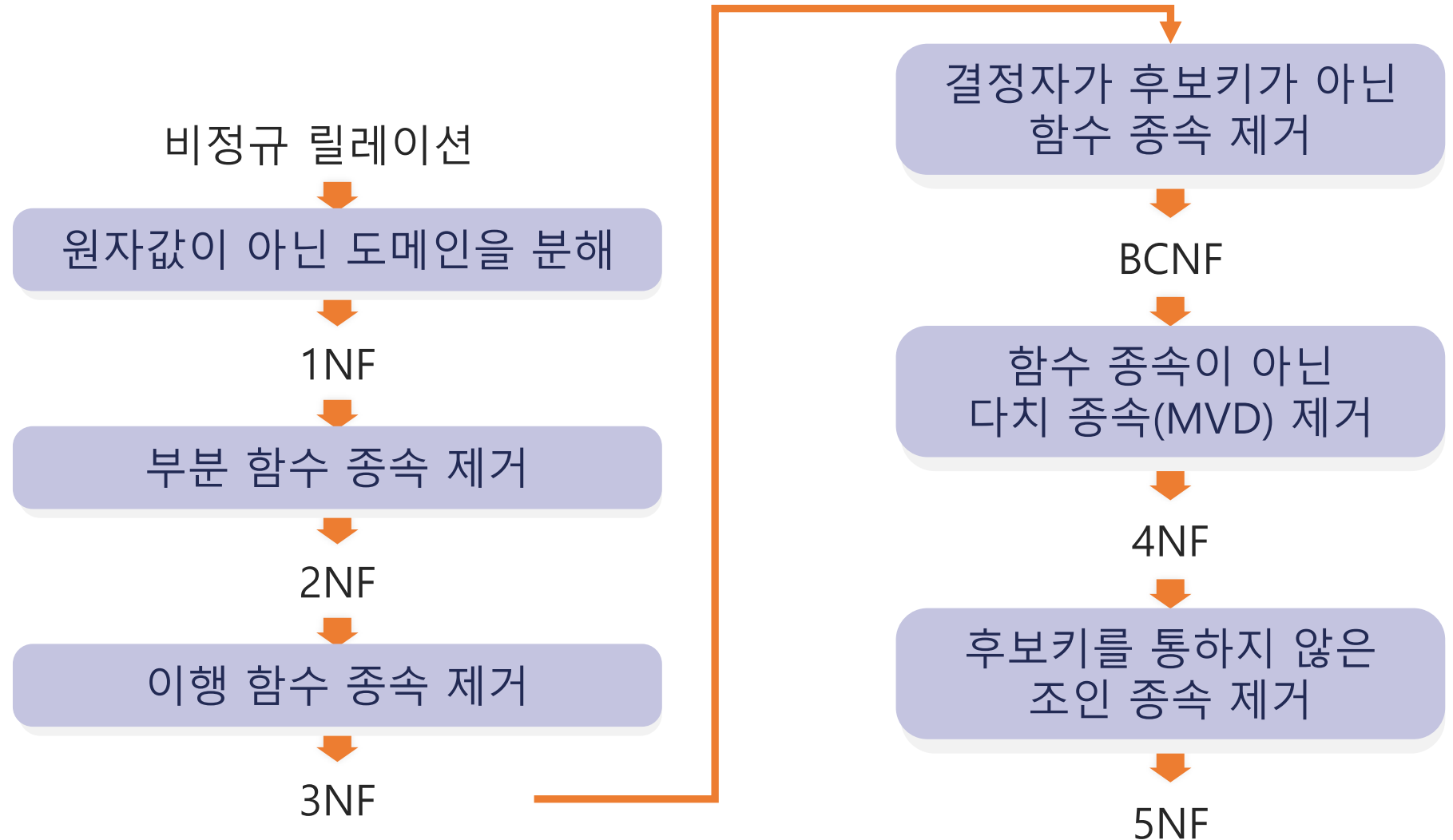
## 함수 종속성

- 데이터 애트리뷰트들의 의미와 애트리뷰트들 간의 상호 관계로부터 유도되는 제약조건의 일종
- 애트리뷰트들의 집합  $X$ 의 값이 애트리뷰트들의 집합  $Y$ 의 값을 유일하게(unique) 결정한다면  $X$ 는  $Y$ 를 함수적으로 결정한다(functionally determines)고 정의



# 학습 정리

## 정규형



# 참고 문헌



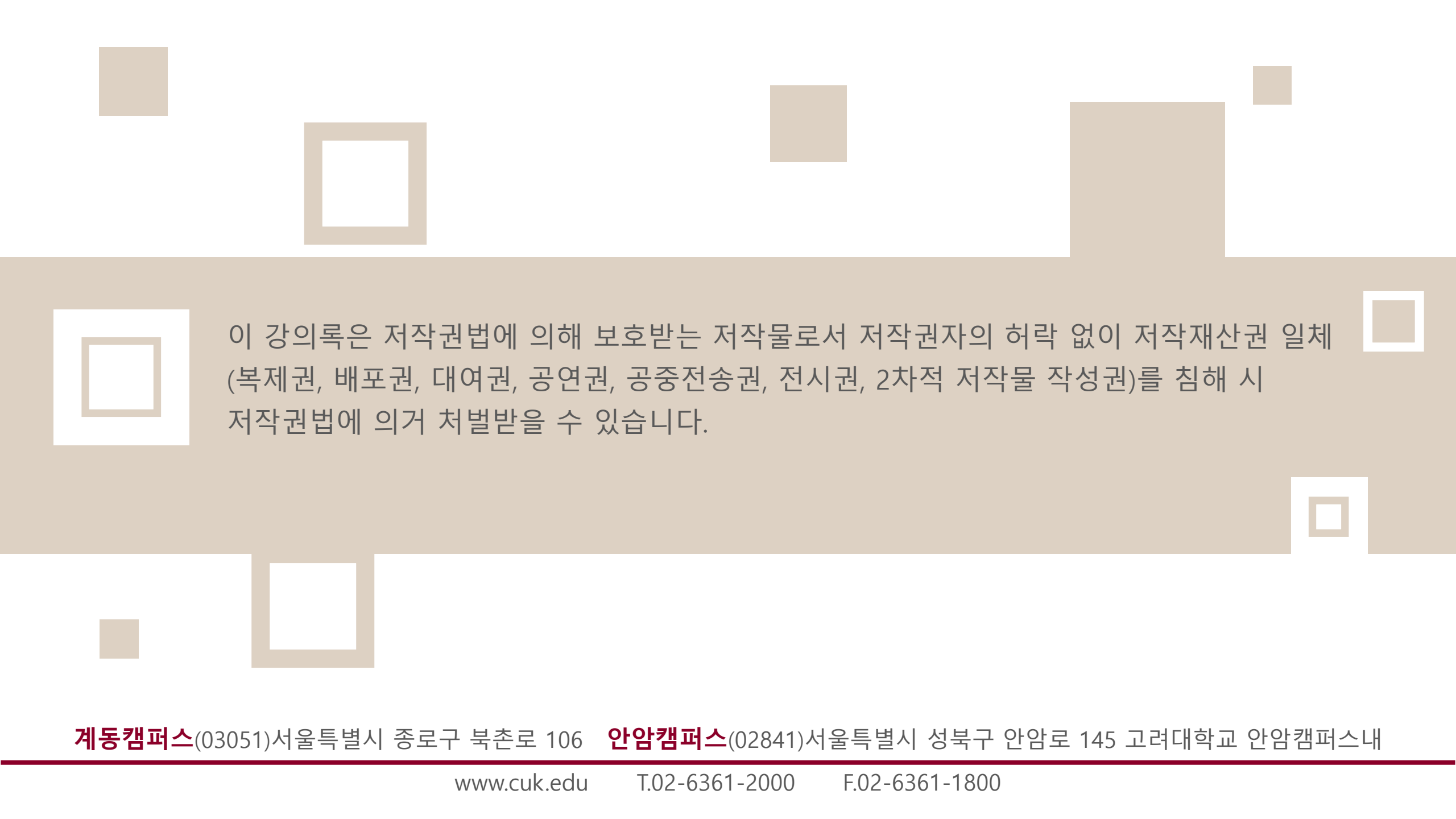
데이터베이스 시스템 7판,  
Ramez Elmasri , Shamkant B. Navathe  
지음, 황규영 등 옮김, 홍릉과학출판사,  
2018년 8월

---



[www.wikipedia.org](http://www.wikipedia.org)

---



이 강의록은 저작권법에 의해 보호받는 저작물로서 저작권자의 허락 없이 저작권재산권 일체 (복제권, 배포권, 대여권, 공연권, 공중전송권, 전시권, 2차적 저작물 작성권)를 침해 시 저작권법에 의거 처벌받을 수 있습니다.