2. 데이터베이스 활용과 클라우드 서버 환경 준비

2 . SQL 심화 / 데이터 모델링 / 데이터베이스연동

01. 이상현상

1. 이상현상의 개념
2. 이상현상의 예

이상현상 , 정규화

목차

1. 이상현상의 개념

- **잘못 설계된 데이터베이스가 어떤 이상현상(anomaly)을 일으키는지 알아보기**

학생번호	학생이름	학과	주소	강좌이름	강의실
501	박지성	컴퓨터과	영국 맨체스타	데이터베이스	공학관 110
401	김연아	체육학과	대한민국 서울	데이터베이스	공학관 110
402	장미란	체육학과	대한민국 강원도	스포츠경영학	체육관 103
502	추신수	컴퓨터과	미국 클리블랜드	자료구조	공학관 111
501	박지성	컴퓨터과	영국 맨체스타	자료구조	공학관 111

학생수강 테이블

- **삭제이상(deletion anomly)** : 투플 삭제 시 같이 저장된 다른 정보까지 연쇄적으로 삭제되는 현상 → 연쇄삭제(triggered deletion) 문제 발생
- **삽입이상(insertion anomly)** : 투플 삽입 시 특정 속성에 해당하는 값이 없어 NULL 값을 입력해야 하는 현상 → NULL 값 문제 발생
- **수정이상(update anomly)** : 투플 수정 시 중복된 데이터의 일부만 수정되어 데이터의 불일치 문제가 일어나는 현상 → 불일치(inconsistency) 문제 발생

2. 이상현상의 예

학생번호	학생이름	학과	주소	강좌이름	강의실
501	박지성	컴퓨터과	영국 맨체스타	데이터베이스	공학관 110
401	김연아	체육학과	대한민국 서울	데이터베이스	공학관 110
402	장미란	체육학과	대한민국 강원도	스포츠경영학	체육관 103
502	추신수	컴퓨터과	미국 클리블랜드	자료구조	공학관 111
501	박지성	컴퓨터과	영국 맨체스타	자료구조	공학관 111
403	박세리	체육학과	대한민국 대전	NULL	NULL

DELETE
- 장미란 학생 삭제
- 연쇄삭제 문제

UPDATE
- 박지성 학생 주소 변경
- 불일치 문제

INSERT
- 박세리 학생 삽입
- NULL 값 문제

데이터 조작과 이상현상

2. 이상현상의 예

❖ 잘못 설계된 계절학기 수강 테이블

■ Summer 테이블을 생성하고 데이터를 삽입하는 SQL 문

```
DROP TABLE IF EXISTS Summer; /* 기존 테이블이 있으면 삭제 */

CREATE TABLE Summer
( sid    INTEGER,
  class VARCHAR(20),
  price INTEGER
);

INSERT INTO Summer VALUES (100, 'FORTRAN', 20000);
INSERT INTO Summer VALUES (150, 'PASCAL', 15000);
INSERT INTO Summer VALUES (200, 'C', 10000);
INSERT INTO Summer VALUES (250, 'FORTRAN', 20000);

/* 생성된 Summer 테이블 확인 */
SELECT *
FROM    Summer;
```

2. 이상현상의 예

❖ 잘못 설계된 계절학기 수강 테이블

■ 각 질의에 대한 SQL문을 직접 실습해보기

Summer 테이블을 이용하여 처리하는 질의와 SQL 문

질의	SQL 문
계절학기를 듣는 학생의 학번과 수강하는 과목은?	SELECT sid, class FROM Summer;
C 강좌의 수강료는?	SELECT price FROM Summer WHERE class='C';
수강료가 가장 비싼 과목은?	SELECT DISTINCT class FROM Summer WHERE price = (SELECT max(price) FROM Summer);
계절학기를 듣는 학생 수와 수강료 총액은?	SELECT COUNT(*), SUM(price) FROM Summer;

2. 이상현상의 예

❖ 잘못 설계된 계절학기 수강 테이블

■ 삭제이상

질의 200번 학생의 계절학기 수강신청을 취소하시오.

```
/* C 강좌 수강료 조회 */
SELECT      price "C 수강료"
FROM        Summer
WHERE       class='C';
```

C 수강료
10000

```
/* 200번 학생의 수강신청 취소 */
DELETE FROM  Summer
WHERE  sid=200;
```

```
/* C 강좌 수강료 다시 조회 */ => C 수강료 조회 불가능!!
SELECT      price "C 수강료"
FROM        Summer
WHERE       class='C';
```

C 수강료

```
/* 다음 실습을 위해 200번 학생 자료 다시 입력 */
INSERT INTO Summer VALUES (200, 'C', 10000);
```

2. 이상현상의 예

❖ 잘못 설계된 계절학기 수강 테이블

■ 삽입이상

질의 계절학기에 새로운 자바 강좌를 개설하시오.

```
/* 자바 강좌 삽입 */ => NULL을 삽입해야 한다. NULL 값은 문제가 있을 수 있다.
INSERT INTO Summer VALUES (NULL, 'JAVA', 25000);

/* Summer 테이블 조회 */
SELECT      *
FROM        Summer;
```

sid	class	price
100	FORTRAN	20000
150	PASCAL	15000
250	FORTRAN	20000
200	C	10000
NULL	JAVA	25000

```
/* NULL 값이 있는 경우 주의할 질의 : 투플은 다섯 개지만 수강학생은 총 네 명임 */
SELECT      COUNT(*) "수강인원"
FROM        Summer;
```

수강 인원
5

```
SELECT      COUNT(sid) "수강인원"
FROM        Summer;
```

수강 인원
4

```
SELECT      count(*) "수강인원"
FROM        Summer
WHERE       sid IS NOT NULL;
```

수강 인원
4

2. 이상현상의 예

❖ 잘못 설계된 계절학기 수강 테이블

■ 수정이상

질의 FORTRAN 강좌의 수강료를 20,000원에서 15,000원으로 수정하시오.

```
/* FORTRAN 강좌 수강료 수정 */
UPDATE      Summer
SET         price=15000
WHERE       class='FORTRAN';

SELECT      *
FROM        Summer;
```

sid	class	price
100	FORTRAN	15000
150	PASCAL	15000
250	FORTRAN	15000
200	C	10000
NULL	JAVA	25000

```
SELECT      DISTINCT price "FORTRAN 수강료"
FROM        Summer
WHERE       class='FORTRAN';
```

FORTRAN 수강료
15000

```
/* 다음 실습을 위해 FORTRAN 강좌의 수강료를 다시 20,000원으로 복구 */
UPDATE      Summer
SET         price=20000
WHERE       class='FORTRAN';

/* 만약 UPDATE 문을 다음과 같이 작성하면 데이터 불일치 문제가 발생함 */
UPDATE      Summer
SET         price=15000
WHERE       class='FORTRAN' AND sid=100;
```

2. 이상현상의 예

❖ 잘못 설계된 계절학기 수강 테이블

```
/* Summer 테이블을 조회하면 FORTRAN 강좌의 수강료가 한 건만 수정되었음 */
SELECT      *
FROM        Summer;
```

sid	class	price
100	FORTRAN	15000
150	PASCAL	15000
250	FORTRAN	20000
200	C	10000
NULL	JAVA	25000

```
/* FORTRAN 수강료를 조회하면 두 건이 나옴(데이터 불일치 문제 발생) */
SELECT      price "FORTRAN 수강료"
FROM        Summer
WHERE       class='FORTRAN';
```

FORTRAN 수강료
15000
20000

```
/* 다음 실습을 위해 FORTRAN 강좌의 수강료를 다시 20,000원으로 복구 */
UPDATE      Summer
SET         price=20000
WHERE       class='FORTRAN';
```

```
/* 다음 실습을 위해 sid가 NULL인 투플 삭제 */
DELETE FROM    Summer
WHERE    sid IS NULL;
```

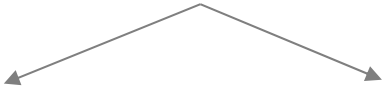
2. 이상현상의 예

❖ 수정된 계절학기 수강 테이블

- 테이블의 구조를 수정하여 이상현상이 발생하지 않는 사례

Summer(sid, class, price)

sid	class	price
100	FORTRAN	20000
150	PASCAL	15000
200	C	10000
250	FORTRAN	20000



SummerPrice(class, price)

class	price
FORTRAN	20000
PASCAL	15000
C	10000

SummerEnroll(sid,class)

sid	class
100	FORTRAN
150	PASCAL
200	C
250	FORTRAN

Summer 테이블의 분리

2. 이상현상의 예

❖ 수정된 계절학기 수강 테이블

■ SummerPrice 테이블과 SummerEnroll 테이블을 생성하는 SQL 문

```
/* 기존 테이블이 있으면 삭제하고 새로 생성하기 위한 준비 */
DROP TABLE SummerPrice;
DROP TABLE SummerEnroll;

/* SummerPrice 테이블 생성 */
CREATE TABLE SummerPrice
(   class VARCHAR2(20),
    price NUMBER
);

INSERT INTO SummerPrice VALUES ('FORTRAN', 20000);
INSERT INTO SummerPrice VALUES ('PASCAL', 15000);
INSERT INTO SummerPrice VALUES ('C', 10000);

SELECT * FROM SummerPrice;

/* SummerEnroll 테이블 생성 */
CREATE TABLE SummerEnroll
(   sid NUMBER,
    class VARCHAR2(20)
);

INSERT INTO SummerEnroll VALUES (100, 'FORTRAN');
INSERT INTO SummerEnroll VALUES (150, 'PASCAL');
INSERT INTO SummerEnroll VALUES (200, 'C');
INSERT INTO SummerEnroll VALUES (250, 'FORTRAN');

SELECT * FROM SummerEnroll;
```

class	price
FORTRAN	20000
PASCAL	15000
C	10000

class	price
FORTRAN	20000
PASCAL	15000
C	10000

2. 이상현상의 예

❖ 수정된 계절학기 수강 테이블

■ 각 질의에 대한 SQL문 실습하기

SummerPrice 테이블과 SummerEnroll 테이블을 이용하여 처리하는 질의와 SQL 문

질의	SQL 문
계절학기를 듣는 학생의 학번과 수강하는 과목은?	SELECT sid, class FROM SummerEnroll;;
C 강좌의 수강료는?	SELECT price FROM SummerPrice WHERE class='C';
수강료가 가장 비싼 과목은?	SELECT DISTINCT class FROM SummerPrice WHERE price = (SELECT max(price) FROM SummerPrice);
계절학기를 듣는 학생 수와 수강료 총액은?	SELECT COUNT(*), SUM(price) FROM SummerPrice, SummerEnroll WHERE SummerPrice.class=SummerEnroll.class;

2. 이상현상의 예

❖ 수정된 계절학기 수강 테이블

■ 삭제이상 없음

질의 200번 학생의 계절학기 수강신청을 취소하시오.

```
/* C 강좌 수강료 조회 */
SELECT      price "C 수강료"
FROM        SummerPrice
WHERE       class='C';
```

C 수강료
10000

```
DELETE
FROM        SummerEnroll
WHERE       sid=200;
```

```
SELECT      *
FROM        SummerEnroll;
```

sid	class
100	FORTRAN
150	PASCAL
250	FORTRAN

```
/* C 강좌의 수강료가 존재하는지 확인 */ => 삭제이상 없음!!
SELECT      price "C 수강료"
FROM        SummerPrice
WHERE       class='C';
```

C 수강료
10000

2. 이상현상의 예

❖ 수정된 계절학기 수강 테이블

■ 삽입이상 없음

질의 계절학기에 새로운 자바 강좌를 개설하시오.

```
/* 자바 강좌 삽입, NULL 값을 입력할 필요 없음 */
INSERT INTO SummerPrice VALUES ('JAVA', 25000);

SELECT      *
FROM        SummerPrice;
```

class	price
FORTRAN	20000
PASCAL	15000
C	10000
JAVA	25000

```
/* 수강신청 정보 확인 */
SELECT      *
FROM        SummerEnroll;
```

sid	class
100	FORTRAN
150	PASCAL
250	FORTRAN

2. 이상현상의 예

❖ 수정된 계절학기 수강 테이블

■ 수정이상 없음

질의 FORTRAN 강좌의 수강료를 20,000원에서 15,000원으로 수정하시오.

```
UPDATE    SummerPrice
SET       price=15000
WHERE     class='FORTRAN';
```

```
SELECT    price "FORTRAN 수강료"
FROM      SummerPrice
WHERE     class='FORTRAN';
```

FORTRAN 수강료
15000

02. 함수 종속성

1. 함수 종속성의 개념
2. 함수 종속성 다이어그램
3. 함수 종속성 규칙
4. 함수 종속성 기본키
5. 이상현상과 결정자
6. 함수 종속성 예제

1. 함수 종속성의 개념

- 학생수강성적 릴레이션의 각 속성 사이에는 의존성이 존재함
- 어떤 속성 A의 값을 알면 다른 속성 B의 값이 유일하게 정해지는 의존 관계를 '속성 B는 속성 A에 종속한다(dependent)' 혹은 '속성 A는 속성 B를 결정한다(determine)'라고 함
- 'A → B'로 표기하며, A를 B의 결정자라고 함

학생수강성적

학생번호	학생이름	주소	학과	학과사무실	강좌이름	강의실	성적
501	박지성	영국 맨체스타	컴퓨터과	공학관101	데이터베이스	공학관 110	3.5
401	김연아	대한민국 서울	체육학과	체육관101	데이터베이스	공학관 110	4.0
402	장미란	대한민국 강원도	체육학과	체육관101	스포츠경영학	체육관 103	3.5
502	추신수	미국 클리블랜드	컴퓨터과	공학관101	자료구조	공학관 111	4.0
501	박지성	영국 맨체스타	컴퓨터과	공학관101	자료구조	공학관 111	3.5

학생수강성적 릴레이션

1. 함수 종속성의 개념

- **학생수강성적 릴레이션에서 종속관계에 있는 예**

  학생번호 → 학생이름

  학생번호 → 주소

  강좌이름 → 강의실

  학과 → 학과사무실

- **종속하지 않는 예**

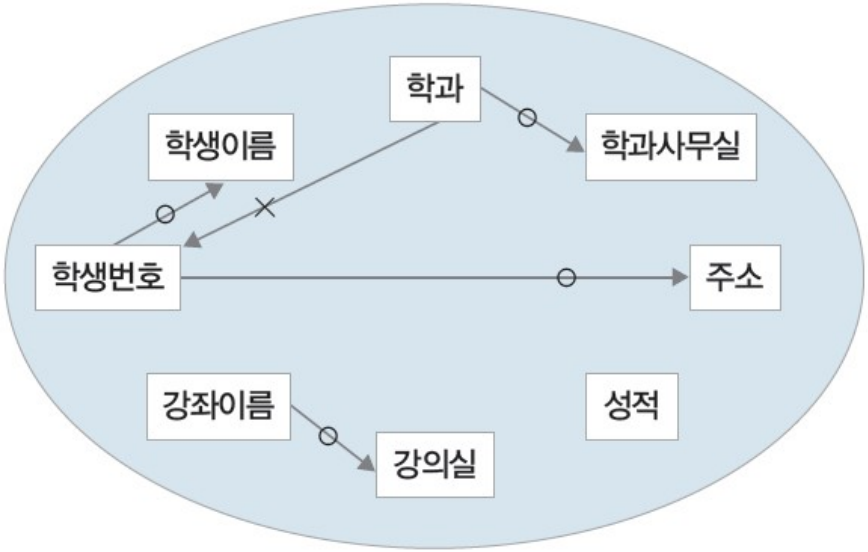
  학생이름 → 강좌이름

  학과 → 학생번호

- **종속하는 것처럼 보이지만 주의 깊게 보면 그렇지 않은 예**

  학생이름 → 학과

1. 함수 종속성의 개념



학생수강성적 릴레이션의 종속관계

함수 종속성(FD, Functional Dependency)

릴레이션 R과 R에 속하는 속성의 집합 X, Y가 있을 때, X 각각의 값이 Y의 값 한 개와 대응이 될 때 'X는 Y를 함수적으로 결정한다'라고 하고 X→Y로 표기함. 이때 X를 결정자(determinant)라고 하고, Y를 종속 속성(dependent attribute)이라고 함. 함수 종속성은 보통 릴레이션 설계 때 속성의 의미로부터 정해짐.

2. 함수 종속성 다이어그램

❖ 함수 종속성 다이어그램(functional dependency diagram)은 함수 종속성을 나타내는 표기법

- 릴레이션의 속성 : 직사각형
- 속성 간의 함수 종속성 : 화살표
- 복합 속성 : 직사각형으로 묶어서 그림

학생수강성적 릴레이션의 함수 종속성 다이어그램

3. 함수 종속성 규칙

함수 종속성 규칙(functional dependency rule)
X, Y, Z가 릴레이션 R에 포함된 속성의 집합이라고 할 때,
함수 종속성에 관한 다음과 같은 규칙이 성립

부분집합(Subset) 규칙 : if $Y \subseteq X$, then $X \rightarrow Y$
증가(Augmentation) 규칙 : If $X \rightarrow Y$, then $XZ \rightarrow YZ$
이행(Transitivity) 규칙 : If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$

위 세 가지 규칙으로부터 부가적으로 다음의 규칙을 얻을 수 있음

결합(Union) 규칙 : If $X \rightarrow Y$ and $X \rightarrow Z$, then $X \rightarrow YZ$
분해(Decomposition) 규칙 : If $X \rightarrow YZ$, then $X \rightarrow Y$ and $X \rightarrow Z$
유사이행(Pseudotransitivity) 규칙 : If $X \rightarrow Y$ and $WY \rightarrow Z$, then $WX \rightarrow Z$

3. 함수 종속성 규칙

학생수강성적 릴레이션에 함수 종속성 규칙을 적용한 예

적용 규칙	사례	설명
부분집합 규칙 if Y ⊆ X, then X → Y	(학과, 주소) → 학과	학과는 (학과, 주소)의 부분집합 속성이므로, '(학과, 주소) → 학과' 성립
증가 규칙 If X → Y, then XZ → YZ	(학생번호, 강좌이름) → (학생이름, 강좌이름)	'학생번호 → 학생이름'이므로 강좌이름을 추가하여, '(학생번호, 강좌이름) → (학생이름, 강좌이름)' 성립
이행 규칙 : If X → Y and Y → Z, then X → Z	학생번호 → 학과사무실	'학생번호 → 학과', '학과 → 학과사무실'이므로 이행 규칙을 적용하여, '학생번호 → 학과사무실' 성립
결합 규칙 If X → Y and X → Z, then X → YZ	학생번호 → (학생이름, 주소)	'학생번호 → 학생이름', '학생번호 → 주소'이므로 결합 규칙을 적용하여, '학생번호 → (학생이름, 주소)' 성립
분해 규칙 If X → YZ, then X → Y and X → Z	학생번호 → 학생이름, 학생번호 → 주소	'학생번호 → (학생이름, 주소)'이므로 분해하여, '학생번호 → 학생이름', '학생번호 → 주소' 성립
유사이행 규칙 If X → Y and WY → Z, then WX → Z	(강좌이름, 학생이름) → 성적	'학생이름 → 학생번호'(학생이름이 같은 경우가 없다고 가정한다), '(강좌이름, 학생번호) → 성적'이므로 유사이행 규칙을 적용하여, '(강좌이름, 학생이름) → 성적' 성립

4. 함수 종속성과 기본키

- **릴레이션의 함수 종속성을 파악하기 위해서는 우선 기본키를 찾아야 함**
- **기본키가 함수 종속성에서 어떤 역할을 하는지 알면 이상현상을 제거하는 정규화 과정을 쉽게 이해할 수 있음**

함수 종속성과 기본키

릴레이션 R(K, A1, A2, A3, ..., An)에서 K가 기본키면, K → R이 성립.

즉 기본키는 릴레이션의 모든 속성에 대해 결정자(determinant)임.

예) 이름이 같은 학생이 없다고 가정하면, '이름 → 학과, 이름 → 주소, 이름 → 취득학점'이므로 '이름 → 이름, 학과, 주소, 취득학점'이 성립한다. 즉 이름 속성이 학생 릴레이션의 전체를 결정함.

이름	학과	주소	취득학점
박지성	컴퓨터과	영국 맨체스타	92
김연아	체육학과	대한민국 서울	95
장미란	체육학과	대한민국 강원도	98
추신수	컴퓨터과	미국 클리블랜드	99

학생 릴레이션

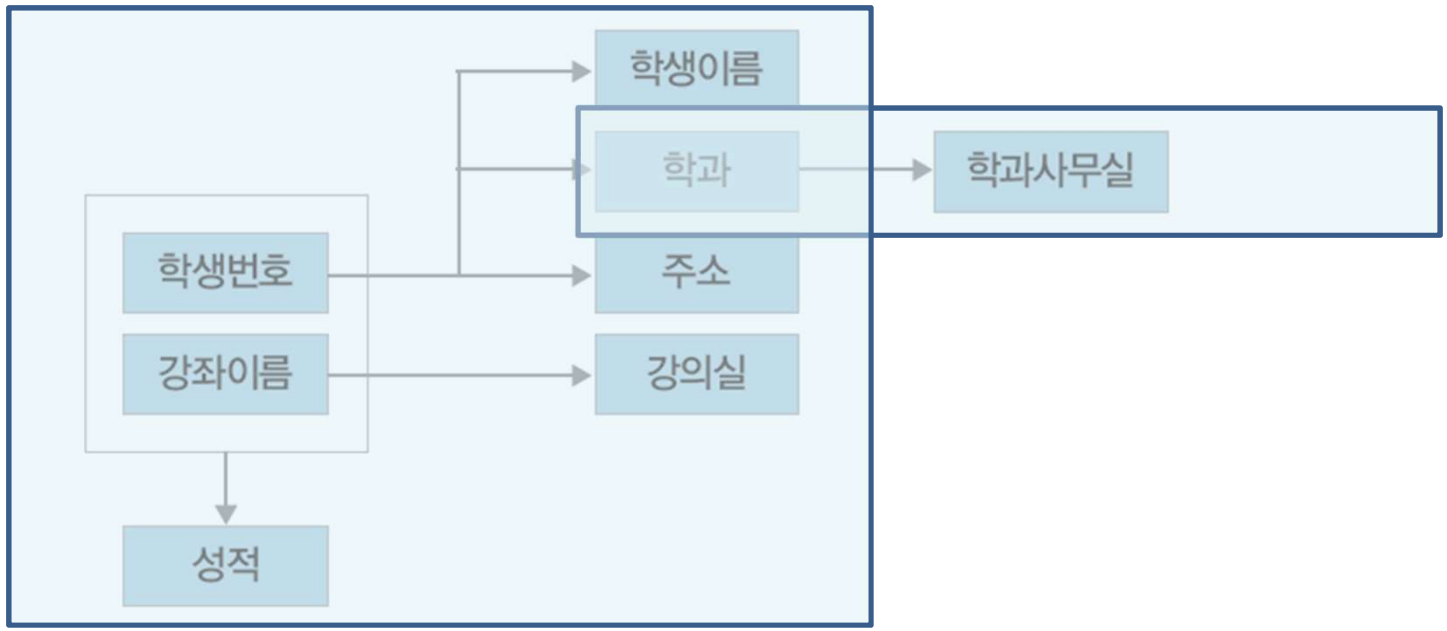
5. 이상현상과 결정자

- 이상현상은 한 개의 릴레이션에 두 개 이상의 정보가 포함되어 있을 때 나타남
  기본키가 아니면서 결정자인 속성이 있을 때 발생함
- 학생수강성적 릴레이션의 경우 학생 정보(학생번호, 학생이름, 주소, 학과)와 강좌 정보(강좌이름, 강의실)가 한 릴레이션에 포함되어서 이상현상이 나타남
  (학과, 학생번호, 강좌이름은 기본키가 아니면서 결정자인 예이다)

학생수강성적 릴레이션의 함수 종속성 다이어그램

5. 이상현상과 결정자

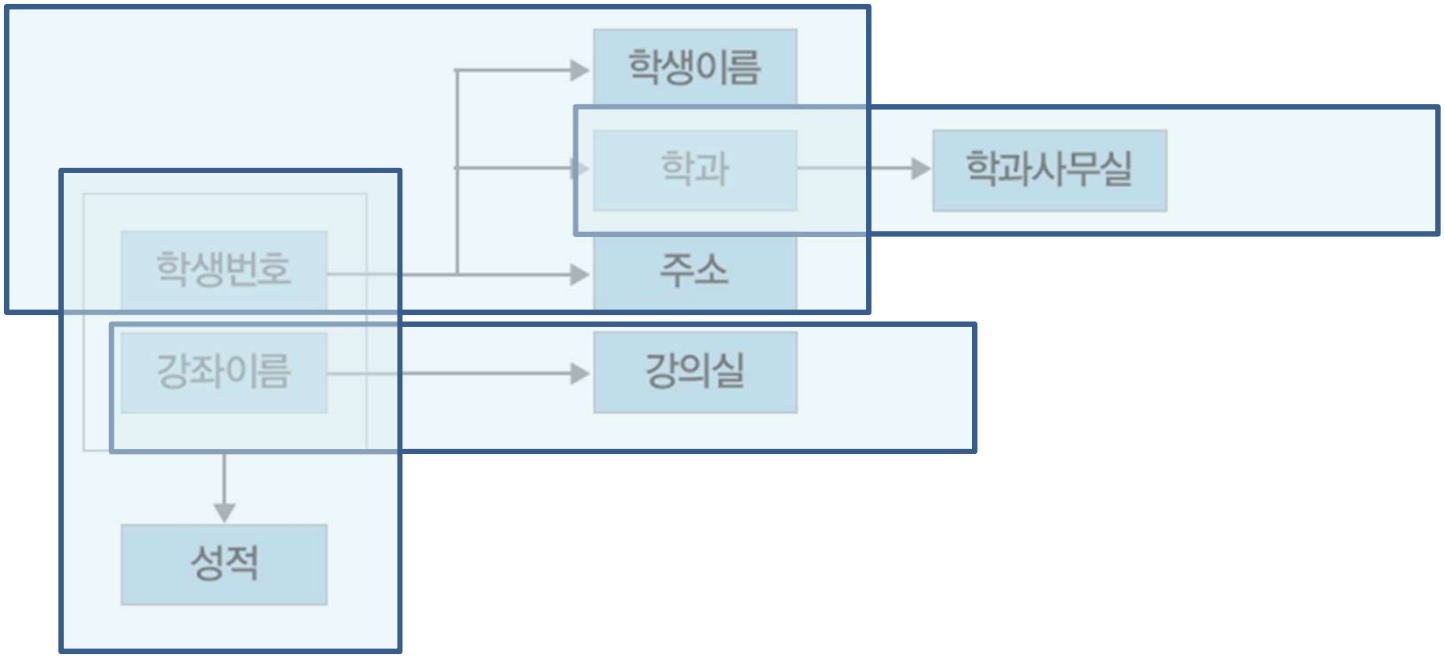
- 이상현상을 없애려면 릴레이션을 분해한다.
- (학과, 학과사무실) 속성을 학생수강성적 릴레이션에서 분리하는 예



학생수강성적 릴레이션의 함수 종속성 다이어그램

5. 이상현상과 결정자

- 릴레이션의 분해



학생수강성적 릴레이션의 함수 종속성 다이어그램

5. 이상현상과 결정자

- **학생수강성적 릴레이션에서 부분 릴레이션을 분해하기**

  분해할 때 부분 릴레이션의 결정자는 원래 릴레이션에 남겨두어야 함. 그래야 분해된 부분 릴레이션이 원래 릴레이션과 관계를 형성할 수 있음

- **[1단계] 학생수강성적 릴레이션에서 (강좌이름, 강의실)을 분리**

  학생수강성적1(학생번호, 학생이름, 학과, 주소, 강좌이름, 성적, 학과사무실)

  강의실(강좌이름, 강의실)

- **[2단계] 학생수강성적1 릴레이션에서 (학생번호, 강좌이름, 성적)을 분리**

  학생학과(학생번호, 학생이름, 학과, 주소, 학과사무실)

  학생성적(학생번호, 강좌이름, 성적)

  강의실(강좌이름, 강의실)

- **[3단계] 학생학과 릴레이션에서 (학과, 학과사무실)을 분리**

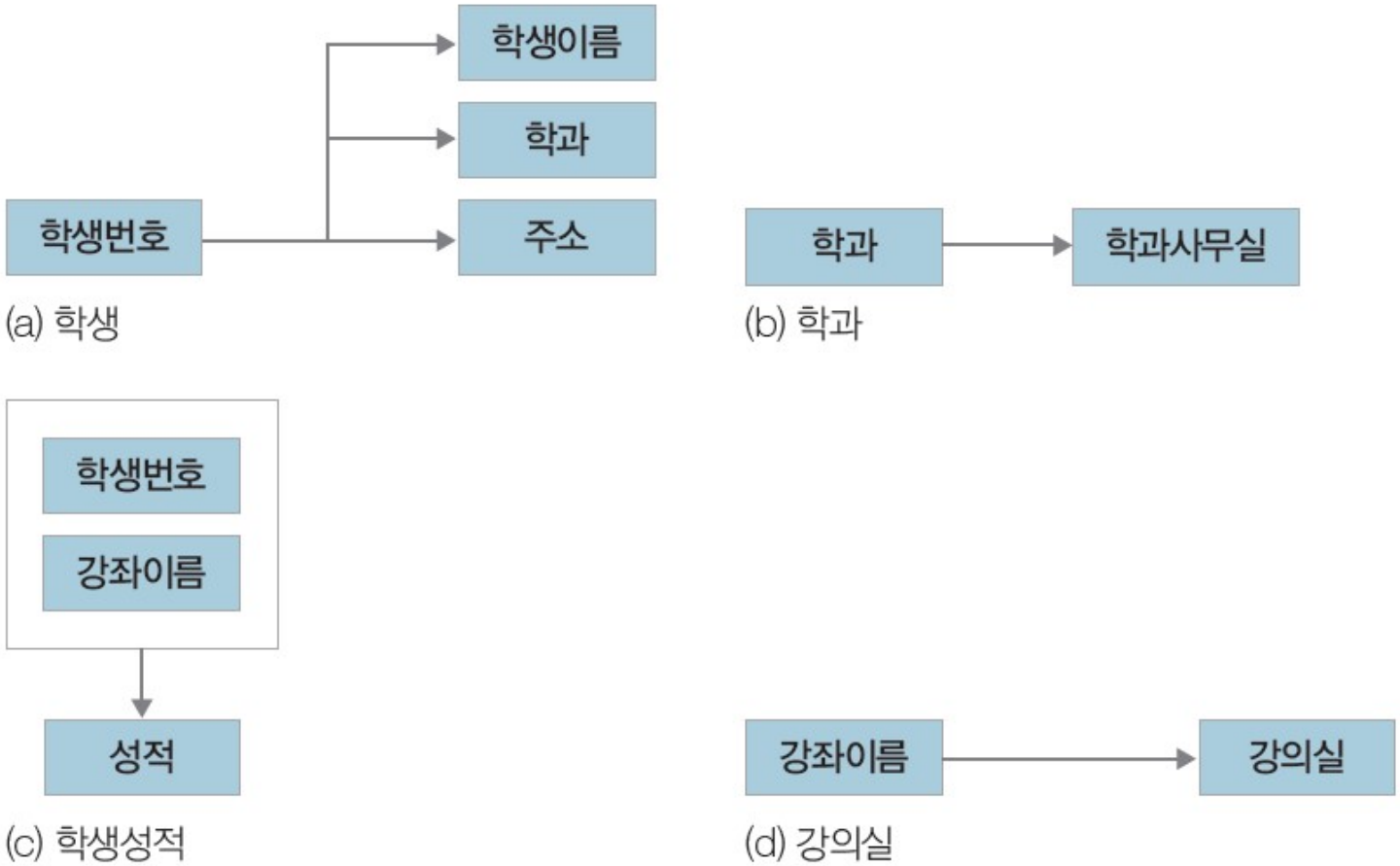
  학생(학생번호, 학생이름, 학과, 주소)

  학과(학과, 학과사무실)

  학생성적(학생번호, 강좌이름, 성적)

  강의실(강좌이름, 강의실)

5. 이상현상과 결정자



학생수강성적 릴레이션을 분해한 결과

6. 함수 종속성 예제

- **함수 종속성은 보통 릴레이션을 설계할 때 속성의 의미로부터 정해지지만, 역으로 릴레이션에 저장된 속성 값으로부터 추정할 수 있음.**

예제 다음 릴레이션 R에서 아래 함수 종속성이 성립하는지 살펴보시오.

R

A	B	C
2	3	8
5	9	6
7	9	6
5	2	2

[함수 종속성]

① A → B

② B → C

③ (B, C) → A

④ (A, B) → C

6. 함수 종속성 예제

예제 다음 릴레이션 R에서 성립하는 함수 종속성을 모두 찾아보시오.

R

A	B	C	D
a1	b4	c1	d6
a1	b2	c4	d5
a2	b4	c1	d4
a2	b2	c4	d3
a2	b3	c2	d2

03. 정규화

1. 정규화 과정
2. 무손실 분해
3. 정규화 정리

정규화

- 이상현상이 발생하는 릴레이션을 분해하여 이상현상을 없애는 과정
- 이상현상이 있는 릴레이션은 이상현상을 일으키는 함수 종속성의 유형에 따라 등급을 구분 가능
- 릴레이션은 정규형 개념으로 구분하며, 정규형이 높을수록 이상현상은 줄어듦

오토바이 1등급 | 자동차 2등급 | 기차 3등급 | 비행기 4등급

▲ 이동수단의 유형에 따른 안전도 등급의 구분 : 등급이 높을수록 빠르고 안전하다

▲ 함수 종속성의 유형에 따른 등급의 구분 : 정규형이 높을수록 이상현상은 줄어든다

이동수단과 릴레이션의 등급 구분

1. 정규화 과정

❖ 제 1정규형

- **릴레이션 R의 모든 속성 값이 원자값을 가지면 제 1정규형이라고 함**
- **제 1정규형으로 변환**

  고객취미들(이름, 취미들) 릴레이션을 고객취미(이름, 취미) 릴레이션으로 바꾸어 저장하면 제 1정규형을 만족함

고객취미들(이름, 취미들)

이름	취미들
김연아	인터넷
추신수	영화, 음악
박세리	음악, 쇼핑
장미란	음악
박지성	게임

고객취미(이름, 취미)

이름	취미
김연아	인터넷
추신수	영화
추신수	음악
박세리	음악
박세리	쇼핑
장미란	음악
박지성	게임

속성 값이 원자값을 갖도록 분해

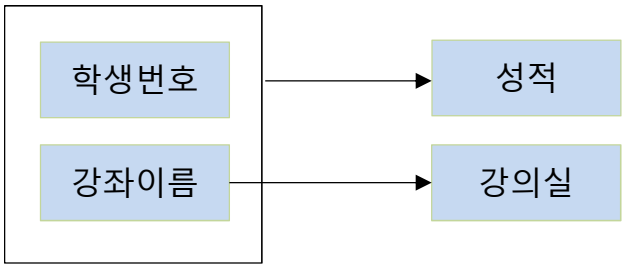
1. 정규화 과정

❖ 제 2정규형

- 릴레이션 R이 제 1정규형이고 기본키가 아닌 속성이 기본키에 완전 함수 종속일 때 제 2정규형이라고 함.
- 완전 함수 종속(full functional dependency) : A와 B가 릴레이션 R의 속성이고 A → B 종속성이 성립할 때, B가 A의 속성 전체에 함수 종속하고 부분 집합 속성에 함수 종속하지 않을 경우 완전 함수 종속라고 함.

수강강좌

학생번호	강좌이름	강의실	성적
501	데이터베이스	공학관 110	3.5
401	데이터베이스	공학관 110	4.0
402	스포츠경영학	체육관 103	3.5
502	자료구조	공학관 111	4.0
501	자료구조	공학관 111	3.3



수강강좌 릴레이션

* 후보키는 무엇인가?

1. 정규화 과정

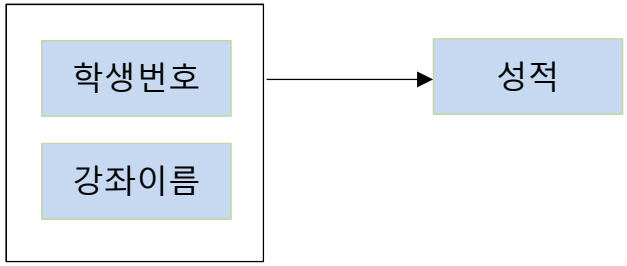
❖ 제 2정규형

■ 제 2정규형으로 변환

수강강좌 릴레이션에서 이상현상을 일으키는(강좌이름, 강의실)을 분해함

수강

학생번호	강좌이름	성적
501	데이터베이스	3.5
401	데이터베이스	4.0
402	스포츠경영학	3.5
502	자료구조	4.0
501	자료구조	3.3



강의실

강좌이름	강의실
데이터베이스	공학관 110
스포츠경영학	체육관 103
자료구조	공학관 111

수강강좌 릴레이션을 수강, 강의실 릴레이션으로 분해

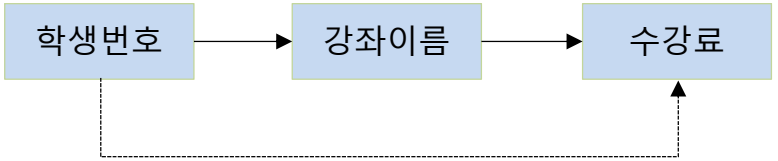
1. 정규화 과정

❖ 제 3정규형

- 릴레이션 R이 제 2정규형이고 기본키가 아닌 속성이 기본키에 비이행적(non-transitive)으로 종속할 때(직접 종속) 제 3정규형이라고 함
- 이행적 종속이란 A → B, B → C가 성립할 때 A → C가 성립되는 함수 종속성

계절학기

학생번호	강좌이름	수강료
501	데이터베이스	20000
401	데이터베이스	20000
402	스포츠경영학	15000
502	자료구조	25000



계절학기 릴레이션

* 계절학기 강좌는 학생은 한 강좌만 신청할 수 있다고 가정한다.

* 후보키는 무엇인가?

1. 정규화 과정

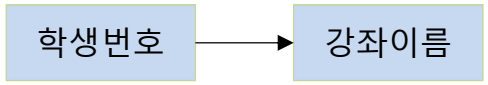
❖ 제 3정규형

■ 제 3정규형으로 변환

계절학기 릴레이션에서 이상현상을 일으키는 (강좌이름, 수강료)를 분해함

계절수강

학생번호	강좌이름
501	데이터베이스
401	데이터베이스
402	스포츠경영학
502	자료구조



수강료

강좌이름	수강료
데이터베이스	20000
스포츠경영학	15000
자료구조	25000

계절학기 릴레이션을 계절수강, 수강료 릴레이션으로 분해

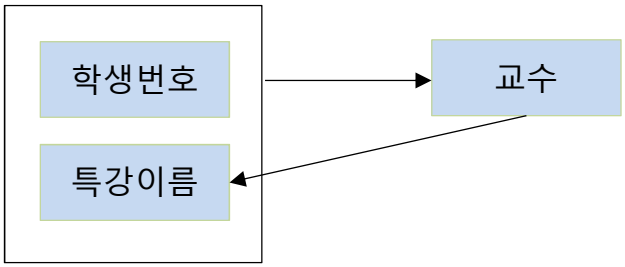
1. 정규화 과정

❖ BCNF

- 릴레이션 R에서 함수 종속성 X → Y가 성립할 때 모든 결정자 X가 후보키이면 BCNF 정규형이라고 함

특강수강

학생번호	특강이름	교수
501	소셜네트워크	김교수
401	소셜네트워크	김교수
402	인간과 동물	승교수
502	창업전략	박교수
501	창업전략	홍교수



특강수강 릴레이션

* 교수는 1개의 특강만을 담당한다.
* 학생은 같은 이름의 특강을 1개만 신청할 수 있다
* 후보키는 무엇인가?

1. 정규화 과정

❖ BCNF

■ BCNF 정규형으로 변환

특강수강 릴레이션에서 이상현상을 일으키는 (교수, 특강이름)을 분해함

특강신청

학생번호	교수
501	김교수
401	김교수
402	승교수
502	박교수
501	홍교수

특강교수

특강이름	교수
소셜네트워크	김교수
인간과 동물	승교수
창업전략	박교수
창업전략	홍교수

특강수강 릴레이션을 특강신청, 특강교수 릴레이션으로 분해

2. 무손실 분해

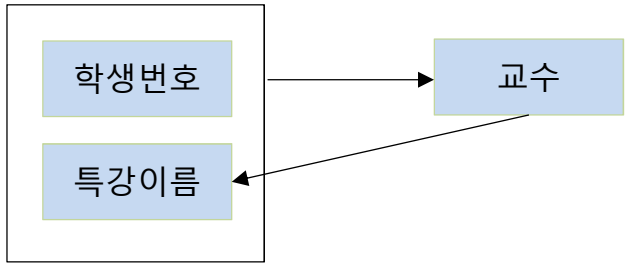
❖ BCNF

- 릴레이션 R을 릴레이션 R1과 R2로 분해할 때, R1 ⋈ R2 = R이면 무손실 분해 (lossless-join decomposition)라고 함
- R1 ∩ R2 → R1 혹은 R1 ∩ R2 → R2 중 하나를 만족해야 함

특강수강

학생번호	특강이름	교수
501	소셜네트워크	김교수
401	소셜네트워크	김교수
402	인간과 동물	승교수
502	창업전략	박교수
501	창업전략	홍교수

특강수강 릴레이션



2. 무손실 분해

❖ BCNF

특강수강 릴레이션의 분해 – 2가지 방법 비교

구분	릴레이션 분해	무손실 분해 여부
[분해1]	특강수강(학생번호, 특강이름, 교수) → R1(학생번호, 교수), R2(교수, 특강이름)	R1과 R2의 공통 속성은 교수이며, 교수는 R2의 키 → 무손실 분해 규칙을 만족
[분해2]	특강수강(학생번호, 특강이름, 교수) → R3(학생번호, 특강이름), R4(교수, 특강이름)	R3와 R4의 공통 속성은 특강이름이지만, 특강이름은 R3나 R4의 키가 아님. → 무손실 분해 규칙을 만족하지 않음

2. 무손실 분해

[분해1]의 경우 R1, R2를 다시 조인하면 원래 릴레이션이 됨

[분해2]의 경우 R3, R4 릴레이션

R3

학생번호	특강이름
501	소셜네트워크
401	소셜네트워크
402	인간과 동물
502	창업전략
501	창업전략

R4

특강이름	교수
소셜네트워크	김교수
인간과 동물	승교수
창업전략	박교수
창업전략	홍교수

특강수강 릴레이션을 R3, R4 릴레이션으로 분해

2. 무손실 분해

R3, R4 릴레이션을 다시 조인하면 의미없는 투플이 생김

-> 무손실 분해 조건을 만족하지 못하고 손실(loss) 분해되었기 때문

특강수강

학생번호	특강이름	교수
501	소셜네트워크	김교수
401	소셜네트워크	김교수
402	인간과 동물	승교수
502	창업전략	박교수
501	창업전략	홍교수

≠

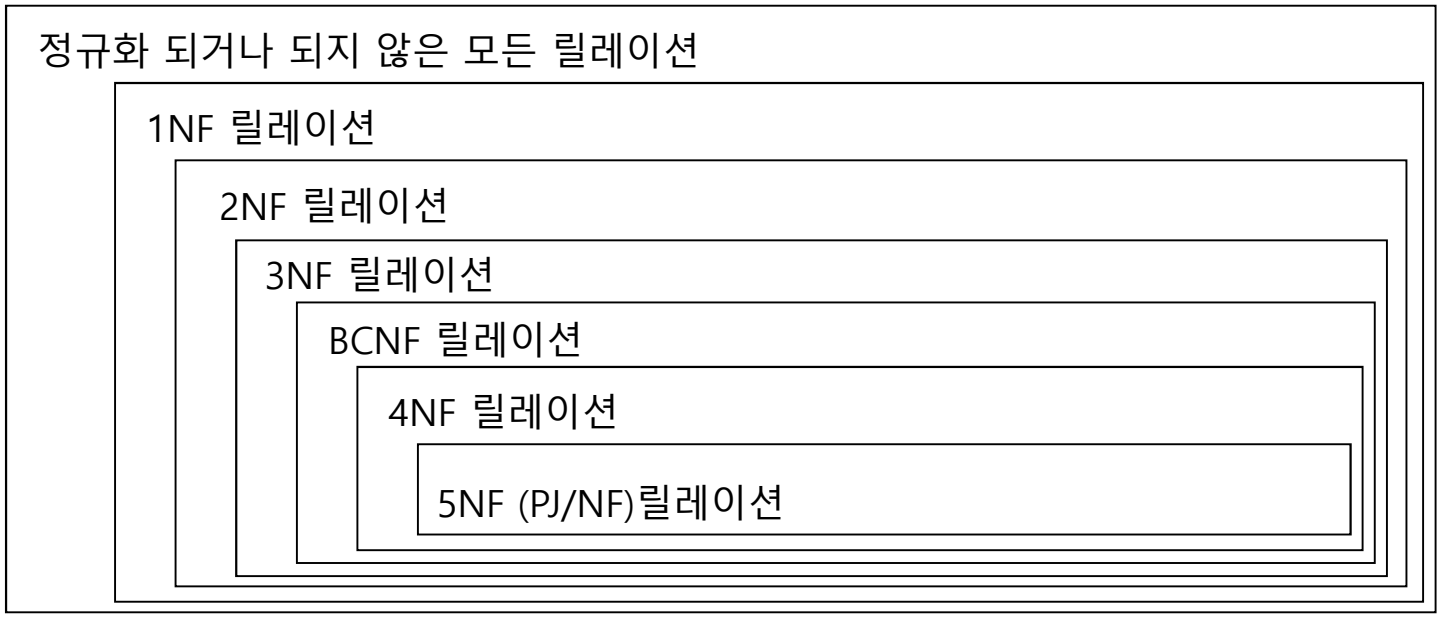
R3 ⋈ R4

학생번호	특강이름	교수
501	소셜네트워크	김교수
401	소셜네트워크	김교수
402	인간과 동물	승교수
502	창업전략	박교수
502	창업전략	홍교수
501	창업전략	박교수
501	창업전략	홍교수

특강수강 릴레이션과 R3 R4 릴레이션의 비교

3. 정규화 정리

- **대부분의 릴레이션은 BCNF까지 정규화하면 실제적인 이상현상이 없어지기 때문에 보통 BCNF까지 정규화를 진행함.**



정규형의 포함 관계

3. 정규화 정리

예제 1 릴레이션R(A, B, C, D)는 다음과 같은 함수 종속성이 성립한다. 아래의 물음에 답하시오.

A → B, B → C, C → D

① 릴레이션 R의 후보키는 무엇인가?
② 릴레이션 R은 몇 정규형인가?
③ 릴레이션을 다음과 같이 분해했을 때 무손실 분해인가?
R1(A, B, C), R2(C, D)

예제 2 릴레이션R(A, B, C)는 다음과 같은 함수 종속성이 성립한다. 아래의 물음에 답하시오.

AB → C, C → A

① 릴레이션 R의 후보키는 무엇인가?
② 릴레이션 R은 몇 정규형인가?
③ 릴레이션을 다음과 같이 분해했을 때 무손실 분해인가?
R1(B, C), R2(A, C)

3. 정규화 정리

예제 3　릴레이션R(A, B, C, D)는 다음과 같은 함수 종속성이 성립한다. 아래의 물음에 답하시오.

AB → C, C → A, C → D

① 릴레이션 R의 후보키는 무엇인가?

② 릴레이션 R은 몇 정규형인가?

③ 릴레이션을 다음과 같이 분해했을 때 무손실 분해인가?

R1(A, B, C), R2(C, D)

04. 정규화 연습(부동산 데이터베이스)

정규화 연습(부동산 데이터베이스)

❖ 부동산 릴레이션

■ **부동산(필지번호, 주소, 공시지가, 소유자이름, 주민등록번호, 전화번호)**

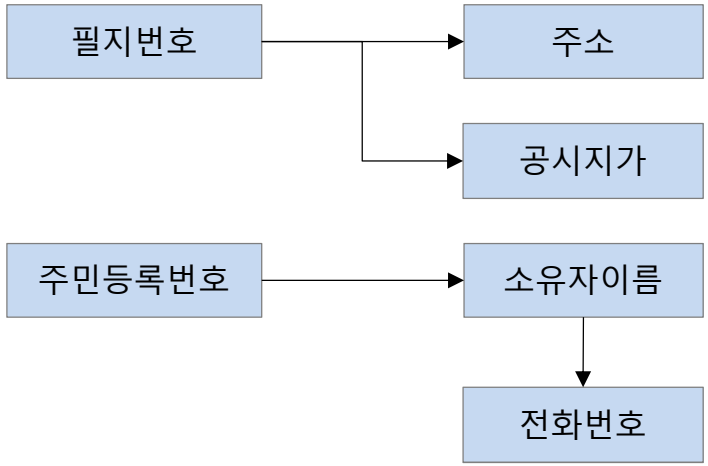
부동산 릴레이션의 함수 종속성

함수 종속성	설명
필지번호 → 주소, 공시지가	땅에 대한 고유의 번호이며 필지에 대하여 주소와 공시가격이 주어짐.
소유자이름 → 전화번호	소유자는 하나의 전화번호를 가짐.
소유자 이름 → 전화번호	소유자에 대하여 전화번호 하나가 주어짐.
주민등록번호 → 소유자이름	사람마다 고유한 주민등록번호가 있음.

정규화 연습(부동산 데이터베이스)

❖ 부동산 릴레이션

- **부동산(필지번호, 주소, 공시지가, 소유자이름, 주민등록번호, 전화번호)**

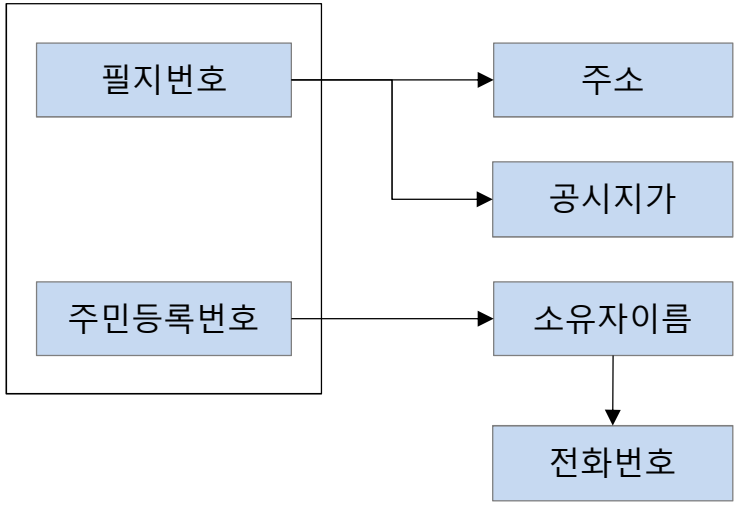


부동산 릴레이션의 함수 종속성 다이어그램

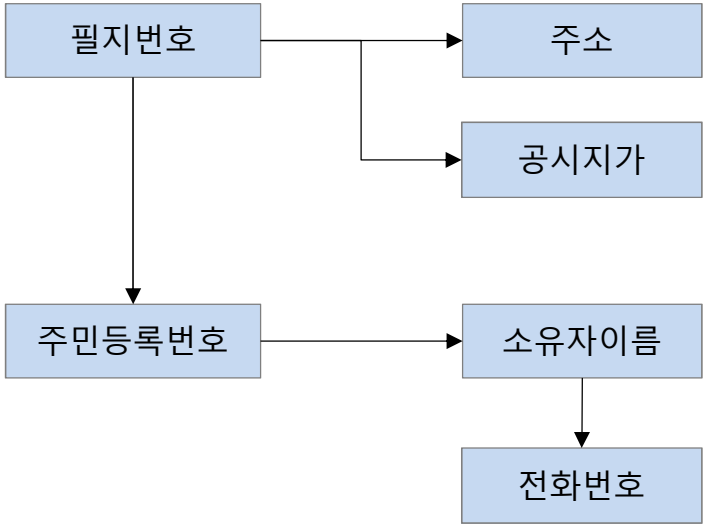
정규화 연습(부동산 데이터베이스)

[사례1] 공동 소유 – 한 필지를 두 사람 이상이 공동으로 소유하는 경우

[사례2] 단독 소유 – 한 필지를 한 사람만 소유하는 경우



[사례1] 부동산1 릴레이션
* 키 = (필지번호, 주민등록번호)



[사례2] 부동산2 릴레이션
* 키 = 필지번호

[사례1]과 [사례2]에 대한 함수 종속성 다이어그램

정규화 연습 (부동산 데이터베이스)

- **[사례1] - 공동 소유**
  - 부동산1 릴레이션은 다음과 같이 분해된다.

    부동산소유(필지번호, 주민등록번호)

    부동산필지(필지번호, 주소, 공시지가)

    소유자(주민등록번호, 소유자이름, 전화번호)

- **[사례2] – 단독 소유**
  - 부동산2 릴레이션은 다음과 같이 분해된다.

    부동산소유(필지번호, 주소, 공시지가, 주민등록번호)

    소유자(주민등록번호, 소유자이름, 전화번호)