Chapter 07

정규화

목차

학습목표

- 데이터베이스 설계 과정에서 발생할 수 있는 이상현상의 종류와 원인을 알아본다.
- 함수 종속성의 개념을 이해하고 관련 규칙을 알아본다.
- 함수 종속성을 이용한 정규화 과정을 알아본다.

01 이상현상

1. 이상현상의 개념과 종류
2. 이상현상의 예

1. 이상현상의 개념과 종류

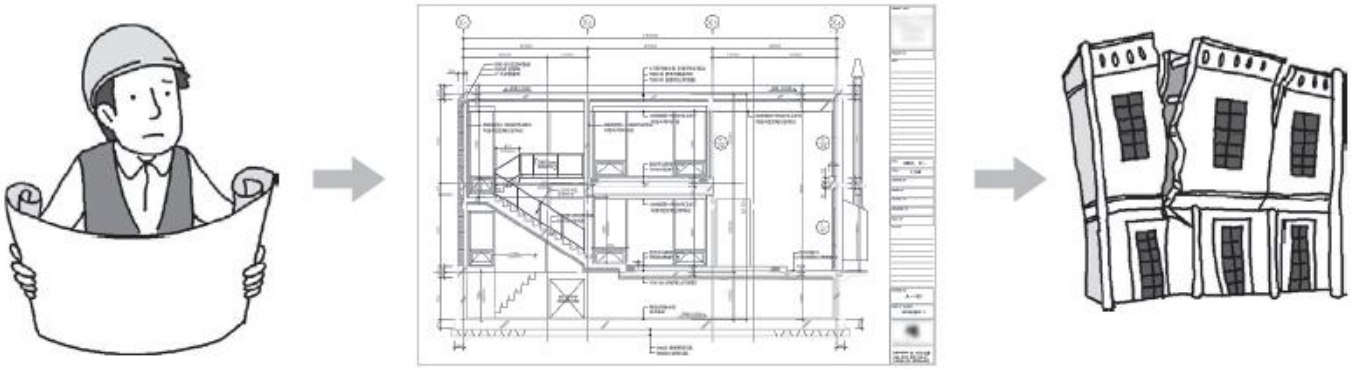
❖ 이상현상(anomaly)

- 데이터 조작 작업에 따라 테이블의 일관성을 훼손하여 데이터의 무결성을 깨뜨리는 현상을 말함
- 잘못 설계된 테이블로 삽입, 삭제, 수정 등의 데이터 조작을 하면 이상현상이 발생할 수 있음

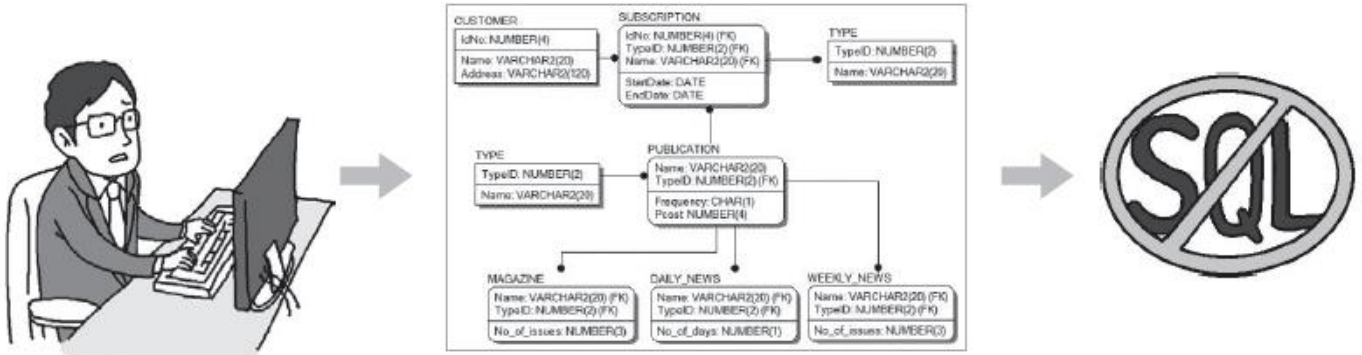
❖ 주요 이상현상

- 삽입 시 이상현상
- 삭제 시 이상현상
- 수정 시 이상현상

1. 이상현상의 개념과 종류



잘못된 건축 설계



잘못된 데이터베이스 설계

그림 7-1 건축과 데이터베이스의 설계 이상

1. 이상현상의 개념과 종류

❖ 잘못 설계된 데이터베이스가 어떤 이상현상을 일으키는지 예

- [그림 7-2]는 대학의 학생 정보와 수강 정보를 저장하는 학생수강 테이블

학생수강

학생번호	학생이름	학과	주소	강좌이름	강의실
501	박지성	컴퓨터학과	영국 맨체스터	데이터베이스	공학관 110
401	김연아	체육학과	대한민국 서울	데이터베이스	공학관 110
402	김연경	체육학과	대한민국 경기도	스포츠경영학	체육관 103
502	추신수	컴퓨터학과	미국 클리블랜드	자료구조	공학관 111
501	박지성	컴퓨터학과	영국 맨체스터	자료구조	공학관 111

그림 7-2 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

1. 이상현상의 개념과 종류

❖ 삭제이상

- 투플 삭제 시 같이 저장된 다른 정보까지 연쇄적으로 삭제되는 현상(그림 7-3] ❶ 참고)

  → **연쇄삭제 triggered deletion 문제 발생**

❖ 삽입이상

- 투플 삽입 시 특정 속성에 해당하는 값이 없어 NULL 값을 입력해야 하는 현상([그림 7-3] ❷ 참고)

  → **NULL 값 문제 발생**

❖ 수정이상

- 투플 수정 시 중복된 데이터의 일부만 수정되어 데이터의 불일치 문제가 일어나는 현상([그림 7-3] ❸ 참고)

  → **일관성 없음(Inconsistency) 문제 발생**

1. 이상현상의 개념과 종류

학생번호	학생이름	학과	주소	강좌이름	강의실
501	박지성	컴퓨터학과	영국 맨체스터	데이터베이스	공학관 110
401	김연아	체육학과	대한민국 서울	데이터베이스	공학관 110
402	김연경	체육학과	대한민국 경기도	스포츠경영학	체육관 103
502	추신수	컴퓨터학과	미국 클리블랜드	자료구조	공학관 111
501	박지성	컴퓨터학과	영국 맨체스터	자료구조	공학관 111
403	박세리	체육학과	대한민국 대전	NULL	NULL

❶ DELETE
- 김연경 학생 삭제
- 연쇄삭제 문제

❷ UPDATE
- 박지성 학생 주소 변경
- 불일치 문제

❸ INSERT
- 박세리 학생 삽입
- NULL 값 문제

그림 7-3 데이터 조작과 이상현상

2. 이상현상의 예

❖ 잘못 설계된 계절학기 수강 테이블

- [그림 7-4]는 계절학기 수강 정보를 저장하는 Summer 테이블 (한 학생은 계절학기 수강은 한 과목만 신청할 수 있다)

Summer

sid	class	price
100	JAVA	20000
150	PYTHON	15000
200	C	10000
250	JAVA	20000

그림 7-4 Summer 테이블

2. 이상현상의 예

- Summer 테이블을 생성하고 데이터를 삽입하는 SQL 문

파일명: 7_code01.sql

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

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

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

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

sid	class	price
100	JAVA	20000
150	PYTHON	15000
200	C	10000
250	JAVA	20000

2. 이상현상의 예

- 마당대학에서는 Summer 테이블을 이용하여 다음과 같은 질의를 처리함

표 7-1 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 LIKE '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;

- [표 7-1]과 같은 조회 작업은 SELECT 문을 이용해 별문제 없이 처리할 수 있음
- 그러나 데이터를 조작하는 작업의 경우 이상현상이 발생함

2. 이상현상의 예

❖ 삭제이상

질의 7-1

200번 학생의 계절학기 수강신청을 취소하시오.

```
/* C 강좌 수강료를 조회하면 10,000원으로 나온다 */
SELECT price 'C 수강료'
FROM   Summer
WHERE  class LIKE 'C';
```

C 수강료
10000

```
/* 200번 학생의 수강신청을 취소하였다 */
SET  SQL_SAFE_UPDATES=0; /* DELETE, UPDATE 연산에 필요한 설정문 */
DELETE   FROM Summer
WHERE    sid=200;
```

```
/* C 강좌 수강료를 다시 조회하면 조회되지 않는다 */
SELECT   price 'C 수강료'
FROM     Summer
WHERE    class LIKE 'C';
```

C 수강료

연쇄삭제로 조회불가능

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

2. 이상현상의 예

❖ 삽입이상

질의 7-2

계절학기에 새로운 강좌 C++을 개설하시오.

```
/* C++ 강좌를 삽입한다 */
INSERT INTO Summer VALUES (NULL, 'C++', 25000);

/* Summer 테이블 전체를 조회해 본다 */
SELECT   *
FROM     Summer;
```

NULL을 삽입해야함

sid	class	price
100	JAVA	20000
150	PYTHON	15000
250	JAVA	20000
200	C	10000
NULL	C++	25000

```
/* NULL 값이 있는 경우 질의에 주의한다
   - 투플은 5개이지만 수강 학생은 총 4명이다 */
SELECT   COUNT(*) '수강 인원'
FROM     Summer;
```

수강 인원
5

```
SELECT   COUNT(sid) '수강 인원'
FROM     Summer;
```

수강 인원
4

2. 이상현상의 예

```
SELECT   COUNT(*) '수강 인원'
FROM     Summer
WHERE    sid IS NOT NULL;
```

수강 인원
4

```
/* 다음 실습을 위해 C++ 강좌를 삭제한다 */
DELETE FROM Summer WHERE class LIKE 'C++';
```

2. 이상현상의 예

❖ 수정이상

질의 7-3

JAVA 강좌의 수강료를 20,000원에서 15,000원으로 수정하시오.

```
/* JAVA 강좌 수강료를 수정하면 JAVA 강좌 수강료 전체가 정상적으로 수정된다 */
UPDATE   Summer
SET      price=15000
WHERE    class='JAVA';

SELECT   *
FROM     Summer;
```

sid	class	price
100	JAVA	15000
150	PYTHON	15000
250	JAVA	15000
200	C	10000

```
/* JAVA 강좌를 조회하면 같은 값이 2번 조회되므로 DISTINCT 문을 사용한다 */
SELECT   DISTINCT price 'JAVA 수강료'
FROM     Summer
WHERE    class LIKE 'JAVA';
```

JAVA 수강료
15000

2. 이상현상의 예

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

/* UPDATE 문을 다음과 같이 조건부로 작성하면 데이터 불일치 문제가 발생한다 */
UPDATE   Summer
SET      price=15000
WHERE    class LIKE 'JAVA' AND sid=100;

/* Summer 테이블을 조회해 보면 JAVA 강좌의 수강료가 1건만 수정되었다 */
SELECT   *
FROM     Summer;
```

sid	class	price
100	JAVA	15000
150	PYTHON	15000
250	JAVA	20000
200	C	10000

수정시 JAVA 수강료가 2가지가 생길수있음

2. 이상현상의 예

```
/* JAVA 수강료를 조회하면 두 건이 나와 데이터 불일치 문제가 발생하였다 */
SELECT   price 'JAVA 수강료'
FROM     Summer
WHERE    class LIKE 'JAVA';
```

JAVA 수강료
15000
20000

```
/* 다음 실습을 위해 JAVA 강좌의 수강료를 다시 20,000원으로 복구한다. */
UPDATE   Summer
SET      price=20000
WHERE    class LIKE 'JAVA';
```

2. 이상현상의 예

❖ 수정된 계절학기 수강 테이블

- 테이블의 구조를 고쳐서 이상현상이 발생하지 않도록 만들어보기

Summer(sid, class, price)

sid	class	price
100	JAVA	20000
150	PYTHON	15000
200	C	10000
250	JAVA	20000

테이블을 분해하여 저장

SummerPrice(class, price)

class	price
JAVA	20000
PYTHON	15000
C	10000

SummerEnroll(sid, class)

sid	class
100	JAVA
150	PYTHON
200	C
250	JAVA

그림 7-5 Summer 테이블의 분리

2. 이상현상의 예

- SummerPrice 테이블과 SummerEnroll 테이블을 생성하는 SQL 문

파일명: 7_code02.sql

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

/* SummerPrice 테이블 생성 */
CREATE TABLE SummerPrice (
       class VARCHAR(20),
       price INTEGER
);
```

```
INSERT INTO SummerPrice VALUES ('JAVA', 20000);
INSERT INTO SummerPrice VALUES ('PYTHON', 15000);
INSERT INTO SummerPrice VALUES ('C', 10000);

SELECT   *
FROM     SummerPrice;
```

class	price
JAVA	20000
PYTHON	15000
C	10000

2. 이상현상의 예

```
/* SummerEnroll 테이블 생성 */
CREATE TABLE SummerEnroll (
       sid INTEGER,
       class VARCHAR(20)
);

INSERT INTO SummerEnroll VALUES (100, 'JAVA');
INSERT INTO SummerEnroll VALUES (150, 'PYTHON');
INSERT INTO SummerEnroll VALUES (200, 'C');
INSERT INTO SummerEnroll VALUES (250, 'JAVA');

SELECT   *
FROM     SummerEnroll;
```

sid	class
100	JAVA
150	PYTHON
200	C
250	JAVA

2. 이상현상의 예

- 분해된 테이블을 이용하여 이전과 같은 질의를 처리해보기

표 7-2 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 LIKE '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

❖ 삭제이상 없음

질의 7-4

200번 학생의 계절학기 수강신청을 취소하시오.

```
/* C 강좌 수강료를 조회하면 10,000원으로 나온다 */
SELECT   price 'C 수강료'
FROM     SummerPrice
WHERE    class LIKE 'C';
```

C 수강료
10000

```
/* 200번 학생이 수강신청을 취소하였다. SummerEnroll 테이블에서 삭제한다 */
SET SQL_SAFE_UPDATES=0; /* DELETE, UPDATE 연산에 필요한 설정문 */
DELETE
FROM     SummerEnroll
WHERE    sid=200;

SELECT   *
FROM     SummerEnroll;
```

sid	class
100	JAVA
150	PYTHON
250	JAVA

2. 이상현상의 예

```
/* C 강좌의 수강료가 존재하는지 확인한다. SummerPrice 테이블에서 조회한다 */
SELECT   price 'C 수강료'
FROM     SummerPrice
WHERE    class LIKE 'C';
```

C 수강료
10000

연쇄삭제 현상이 발생하지 않고 데이터가 조회됨

```
/* 다음 실습을 위해 200번 학생의 자료를 다시 입력한다 */
INSERT INTO SummerEnroll VALUES (200, 'C');
```

❖ 삽입이상 없음

질의 7-5

계절학기에 새로운 강좌 C++를 개설하시오.

2. 이상현상의 예

```
/* C++ 강좌를 삽입한다. SummerPrice 테이블에 NULL 값 없이 입력할 수 있다 */
INSERT INTO SummerPrice VALUES ('C++', 25000);

SELECT   *
FROM     SummerPrice;
```

class	price
JAVA	20000
PYTHON	15000
C	10000
C++	25000

NULL 값을 삽입하지 않아도 됨

```
/* 수강신청 정보를 SummerEnroll 테이블에서 확인한다 */
SELECT   *
FROM     SummerEnroll;
```

sid	class
100	JAVA
150	PYTHON
200	C
250	JAVA

```
/* 다음 실습을 위해 C++ 강좌를 삭제한다 */
DELETE FROM SummerPrice WHERE class LIKE 'C++';
```

2. 이상현상의 예

❖ 수정이상 없음

질의 7-6

JAVA 강좌의 수강료를 20,000원에서 15,000원으로 수정하시오.

```
/* JAVA 강좌 수강료를 수정한다 */
UPDATE   SummerPrice
SET      price=15000
WHERE    class LIKE 'JAVA';

/* JAVA 강좌 수강료를 조회해 보니 불일치 문제가 없다 */
SELECT   price 'JAVA 수강료'
FROM     SummerPrice
WHERE    class LIKE 'JAVA';
```

JAVA 수강료
15000

데이터 중복이 없어서 데이터 불일치가 발생하지 않음

```
/* 실습 후 JAVA 강좌의 수강료를 다시 20,000원으로 복구한다 */
UPDATE   SummerPrice
SET      price=20000
WHERE    class LIKE 'JAVA';
```

연습문제 (Q7.1)

01 정규화의 필요성으로 거리가 먼 것은?

① 데이터 구조의 안정성 최대화

② 중복 데이터의 활성화

③ 데이터 수정, 삭제 시 이상현상의 최소화

④ 테이블 불일치 위험의 최소화

02 관계 데이터베이스의 정규화에 대한 설명으로 옳지 않은 것은?

① 정규화를 거치지 않으면 여러 가지 상이한 종류의 정보가 하나의 릴레이션에 표현되기 때문에 릴레이션을 조작할 때 이상현상이 발생할 수 있다.

② 정규화의 목적은 각 릴레이션에 분산된 종속성을 하나의 릴레이션에 통합하는 것이다.

③ 이상현상은 속성 간에 존재하는 함수 종속성이 원인이다.

④ 정규화가 잘못되면 데이터의 불필요한 중복을 야기하여 릴레이션을 조작할 때 문제가 된다.

03 정규화 과정에서 발생하는 이상현상에 관한 설명으로 옳지 않은 것은?

① 이상현상은 속성 간에 존재하는 여러 종류의 종속관계가 하나의 릴레이션에 표현되어 있을 때 발생한다.

② 속성 간의 종속관계를 분석하여 여러 개의 릴레이션을 하나로 결합하여 이상현상을 해결한다.

③ 삭제이상, 삽입이상, 수정이상이 있다.

④ 정규화는 이상현상을 제거하기 위해서 중복성 및 종속성을 배제시키는 방법을 사용한다.

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 맨체스터	컴퓨터학과	공학관 101	데이터베이스	공학관 110	3.5
401	김연아	대한민국 서울	체육학과	체육관 101	데이터베이스	공학관 110	4.0
402	김연경	대한민국 경기도	체육학과	체육관 101	스포츠경영학	체육관 103	3.5
502	추신수	미국 클리블랜드	컴퓨터학과	공학관 101	자료구조	공학관 111	4.0
501	박지성	영국 맨체스터	컴퓨터학과	공학관 101	자료구조	공학관 111	3.5

그림 7-6 학생수강성적 릴레이션

1. 함수 종속성의 개념

❖ 정규화(mormalization)

- 이상현상이 있는 테이블을 수정하여 문제를 해결하는 과정
- 정규화하려면 먼저 테이블을 분석하여 기본키와 함수 종속성을 파악해야 함

- 다음 그림은 마당대학의 학생 정보와 수강 정보를 저장하는 학생수강성적 릴레이션

학생수강성적

학생번호	학생이름	주소	학과	학과사무실	강좌이름	강의실	성적
501	박지성	영국 맨체스터	컴퓨터학과	공학관 101	데이터베이스	공학관 110	3.5
401	김연아	대한민국 서울	체육학과	체육관 101	데이터베이스	공학관 110	4.0
402	김연경	대한민국 경기도	체육학과	체육관 101	스포츠경영학	체육관 103	3.5
502	추신수	미국 클리블랜드	컴퓨터학과	공학관 101	자료구조	공학관 111	4.0
501	박지성	영국 맨체스터	컴퓨터학과	공학관 101	자료구조	공학관 111	3.5

그림 7-6 학생수강성적 릴레이션

- 학생수강성적 릴레이션의 각 속성 사이에는 의존성이 존재함
- 강좌이름이 '데이터베이스'인 경우 강의실은 '공학관 110'호 한 곳뿐임
- 학생이름이 '박지성'인 경우 강좌이름은 '데이터베이스'도 있고 '자료구조'도 있음

1. 함수 종속성의 개념

- 어떤 속성 A의 값을 알면 다른 속성 B의 값이 유일하게 정해지는 의존 관계를 '속성 B는 속성 A에 종속한다(dependent)' 혹은 '속성 A는 속성 B를 **결정**한다(determine)'라고 함
- 이 관계를 'A → B'로 표기하며, A는 B의 **결정자**라고 함
- [그림 7-7]은 학생수강성적 릴레이션의 종속관계를 나타냄
  (a)는 함수종속성 성립, (b)는 성립하지 않는 예, (c)는 상황에 따라 다름

(a) 학생번호 → 학생이름
학생번호 → 주소
강좌이름 → 강의실
학과 → 학과사무실

(b) 학생이름 → 강좌이름
학과 → 학생번호

(c) 학생이름 → 학과

그림 7-7 학생수강성적 릴레이션의 종속관계 예

1. 함수 종속성의 개념

- [그림 7-8]은 학생수강성적 릴레이션의 종속관계 여부를 일부 표시한 것
  - 종속관계인 경우 연결하는 선에 ○를, 그렇지 않은 경우 ×를 표시함

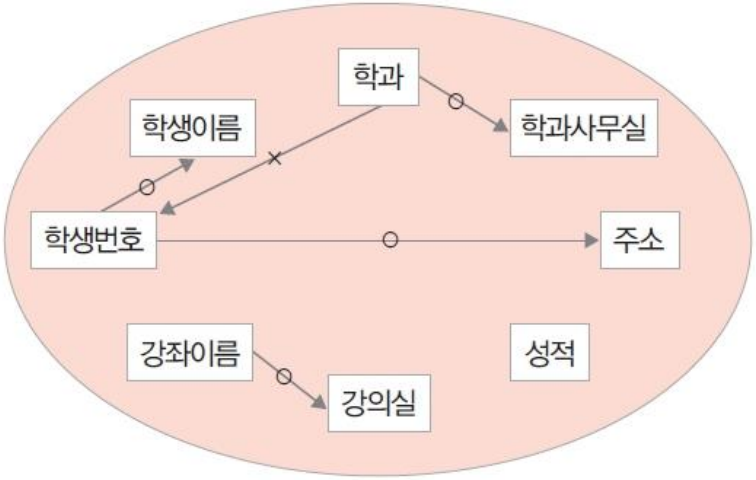


그림 7-8 학생수강성적 릴레이션의 종속관계

- '학생번호 → 주소'와 같이 왼쪽 속성의 각 값에 대하여 오른쪽 속성의 값이 유일하게 결정될 때 '함수적으로 종속한다'라고 함
- 릴레이션의 속성 간에 함수적으로 종속하는 성질을 '함수 종속성' 혹은 원어 그대로 해석하여 '함수적 종속성'이라고 함

정의 | 함수 종속성(FD, Functional Dependency)

릴레이션 R과 R에 속하는 속성의 집합 X, Y가 있을 때, X 각각의 값이 Y 값 한 개와 대응되면 'X는 Y를 함수적으로 결정한다'라고 하고 X → Y로 표기한다. 이때 X를 결정자(determinant)라고 하고, Y를 종속 속성(dependent attribute)이라고 한다. 함수 종속성은 보통 릴레이션 설계 때 속성의 의미를 바탕으로 정해진다.

2. 함수 종속성 다이어그램

❖ 함수 종속성 다이어그램

- 릴레이션의 속성을 직사각형, 속성 간의 함수 종속성을 화살표로 나타냄
- 학생수강성적 릴레이션에서 함수 종속성을 찾아 정리하면 [그림 7-9]의 (a)와 같음

학생번호 → 학생이름
학생번호 → 학과
학생번호 → 주소
학과 → 학과사무실
강좌이름 → 강의실
(학생번호, 강좌이름) → 성적

(a) 함수 종속성

(b) 함수 종속성 다이어그램

그림 7-9 학생수강성적 릴레이션의 함수 종속성과 다이어그램

3. 함수 종속성 규칙

- 함수 종속성 규칙에 대한 정의

정의 | 함수 종속성 규칙(functional dependency rule)

X, Y, Z가 릴레이션 R에 포함된 속성의 집합이라고 할 때, 함수 종속성에 관한 다음과 같은 규칙이 성립한다.

- 부분집합(subset) 규칙: if $Y \subseteq X$, then $X \rightarrow Y$, 사소한(trivial) 함수적 종속이라고도 한다.
- 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 있다.

- 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 릴레이션에서 유도할 수 있는 종속관계 [표 7-3]

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

3. 함수 종속성 규칙

- 함수 종속성 부가적 규칙 유도

(결합 규칙) X → Y, X → Z이면 X → YZ

(부분집합규칙) X ⊆ X이므로 X → X,

(증가규칙) X → Y이면 X → XY

X → Z이면 XY → YZ

(이행규칙) X → XY, XY → YZ이면 X → YZ

(분해 규칙) X → YZ이면 X → Y, X → Z

(부분집합규칙) Y ⊆ YZ, Z ⊆ YZ이므로 YZ → Y, YZ → Z

(이행규칙) X → YZ, YZ → Y이면 X → Y

X → YZ, YZ → Z이면 X → Z

(유사이행 규칙) X → Y, WY → Z이면 WX → Z

(증가규칙) X → Y이면 WX → WY

(이행규칙) WX → WY, WY → Z이면 WX → Z

4. 함수 종속성과 기본키

- 릴레이션의 함수 종속성을 파악하기 위해서는 우선 기본키를 찾아야 함

> **정의 | 함수 종속성과 기본키**
>
> 릴레이션 R(K, A1, A2, A3, ⋯, An)에서 K가 기본키이면, K → R이 성립한다. 즉, 기본키는 릴레이션의 모든 속성에 대한 결정자(determinant)다.

- 예) 이름이 같은 학생이 없다고 가정하면, '이름 → 학과, 이름 → 주소, 이름 → 취득학점'이므로 '이름 → 이름, 학과, 주소, 취득학점'이 성립함
- 즉, 이름 속성이 학생 릴레이션의 전체를 결정

학생

이름	학과	주소	취득학점
박지성	컴퓨터학과	영국 맨체스터	92
김연아	체육학과	대한민국 서울	95
김연경	체육학과	대한민국 경기도	98
추신수	컴퓨터학과	미국 클리블랜드	99

그림 7-10 학생 릴레이션

5. 이상현상과 결정자

- 학생수강성적 릴레이션의 경우 학생 정보(학생번호, 학생이름, 주소, 학과)와 강좌 정보(강좌이름, 강의실)가 한 릴레이션에 포함되어 있기 때문에 이상현상이 나타남

학생수강성적

학생번호	학생이름	주소	학과	학과사무실	강좌이름	강의실	성적
501	박지성	영국 맨체스터	컴퓨터학과	공학관 101	데이터베이스	공학관 110	3.5
401	김연아	대한민국 서울	체육학과	체육관 101	데이터베이스	공학관 110	4.0
402	김연경	대한민국 경기도	체육학과	체육관 101	스포츠경영학	체육관 103	3.5
502	추신수	미국 클리블랜드	컴퓨터학과	공학관 101	자료구조	공학관 111	4.0
501	박지성	영국 맨체스터	컴퓨터학과	공학관 101	자료구조	공학관 111	3.5

그림 7-11 학생수강성적 릴레이션

그림 7-12 학생수강성적 릴레이션의 함수 종속성 다이어그램

5. 이상현상과 결정자

- 이상현상은 기본키가 아니면서 결정자인 속성(비후보키 결정자 속성)이 릴레이션에 존재할 때 발생한다.
- 릴레이션을 결정자인 속성이 기본키가 되도록 분해하여 이상현상을 제거한다.
- 학생수강성적 릴레이션에서 부분 릴레이션을 분해해보기

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

강의실(강좌이름, 강의실)

(a) 1단계 분해: 학생수강성적 릴레이션에서 (강좌이름, 강의실) 분리

5. 이상현상과 결정자

학생수강성적2(학생번호, 학생이름, 학과, 주소, 강좌이름, 성적)

학과(학과, 학과사무실)

강의실(강좌이름, 강의실)

(b) 2단계 분해: 학생수강성적1 릴레이션에서 (학과, 학과사무실) 분리

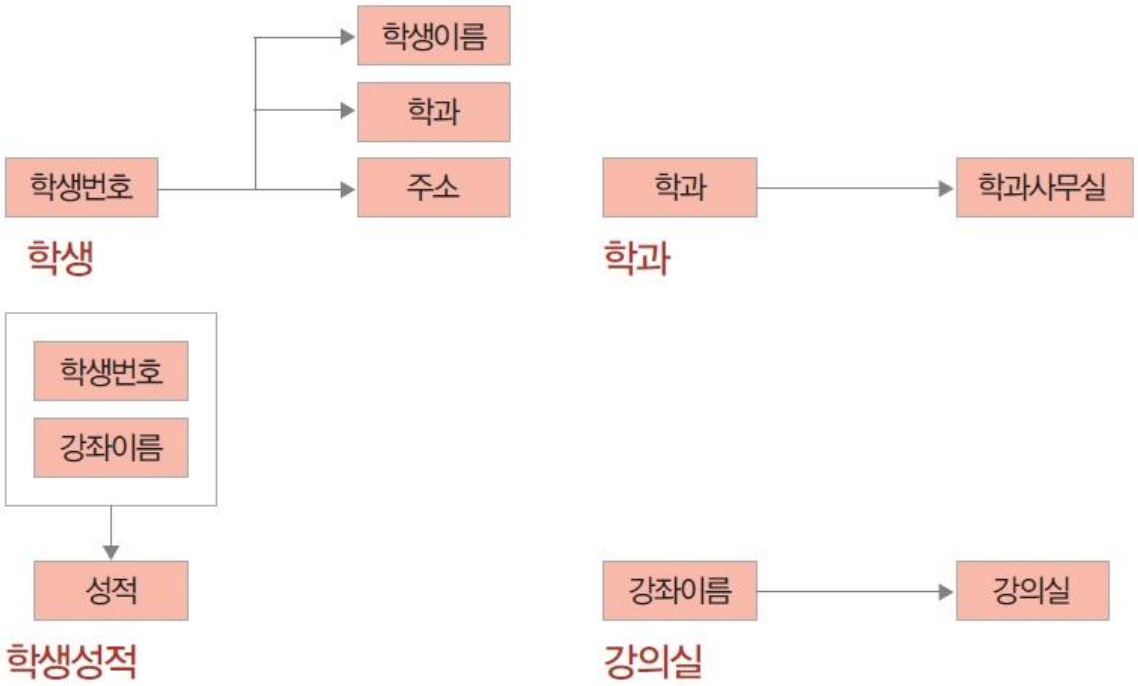
5. 이상현상과 결정자

학생성적(학생번호, 강좌이름, 성적)

학생(학생번호, 학생이름, 학과, 주소)

학과(학과, 학과사무실)

강의실(강좌이름, 강의실)



(c) 최종 분해 결과: 학생수강성적2 릴레이션에서 (학생번호, 학생이름, 학과, 성적) 분리

그림 7-13 학생수강성적의 이상현상을 없애기 위한 단계적 분해

6. 함수 종속성 예제

- 속성의 의미가 주어지지 않은 릴레이션에서 함수 종속성을 찾아보기 위해 다음 예제를 살펴보기

예제 7-1

릴레이션 R에서 오른쪽 함수 종속성이 성립하는지 살펴보시오.

R

A	B	C
b	c	h
e	i	f
g	i	f
e	b	a

[함수 종속성]

① $A \rightarrow B$

② $B \rightarrow C$

③ $(B, C) \rightarrow A$

④ $(A, B) \rightarrow C$

① A → B: 성립하지 않는다. A의 e에 대해 B의 i와 b가 대응한다.

② B → C: 성립한다. B 값에 대하여 C의 값이 한 개씩만 대응한다.

③ (B, C) → A: 성립하지 않는다. (i, f) 값에 대하여 e와 g가 대응한다.

④ (A, B) → C: 성립한다. 모든 투플의 (A, B) 값이 다르다.

6. 함수 종속성 예제

예제 7-2

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

결정자가 한 개인 경우: B → C, C → B, D → A, D → B, D → C

결정자가 두 개인 경우: AB → C, AB → D, AC → B, AC → D, AD → B, AD → C,
BD → A, BD → C, CD → A, CD → B

결정자가 세 개인 경우: ABC → D, ABD → C, ACD → B, BCD → A

- 정답

B → C, C → B, D → A, D → B, D → C, AB → D, AC → D

연습문제 (Q7.2)

11 다음 릴레이션 R을 보고, 오른쪽 함수 종속성 중에서 성립하는 것을 모두 고르시오.

R

A	B	C	D
1	1	1	1
2	2	2	2
2	1	3	1
4	3	4	3

A → B, A → C, A → D, B → A, B → C, B → D,
C → A, C → B, C → D, D → A, D → B, D → C

12 함수 종속성에 관한 다음 물음에 답하시오.

(1) 다음 릴레이션 R을 보고, 아래의 함수 종속성이 성립하는지 답하시오. 그 이유도 설명하시오.

R

A	B	C
b	c	h
e	i	f
g	i	f
e	b	a

① A → C
② B → A
③ C → B
④ C → A
⑤ (A, B, C) → C
⑥ (A, C) → B

(2) 릴레이션 R(A, B, C, D, E)에서 다음과 같은 함수 종속성이 성립한다고 한다. 아래의 종속성 중 성립하지 않는 것은?

함수 종속성: A → B, A → C, CD → E, B → D, E → A

① CD → AC
② BD → CD
③ BC → CD
④ AC → BC

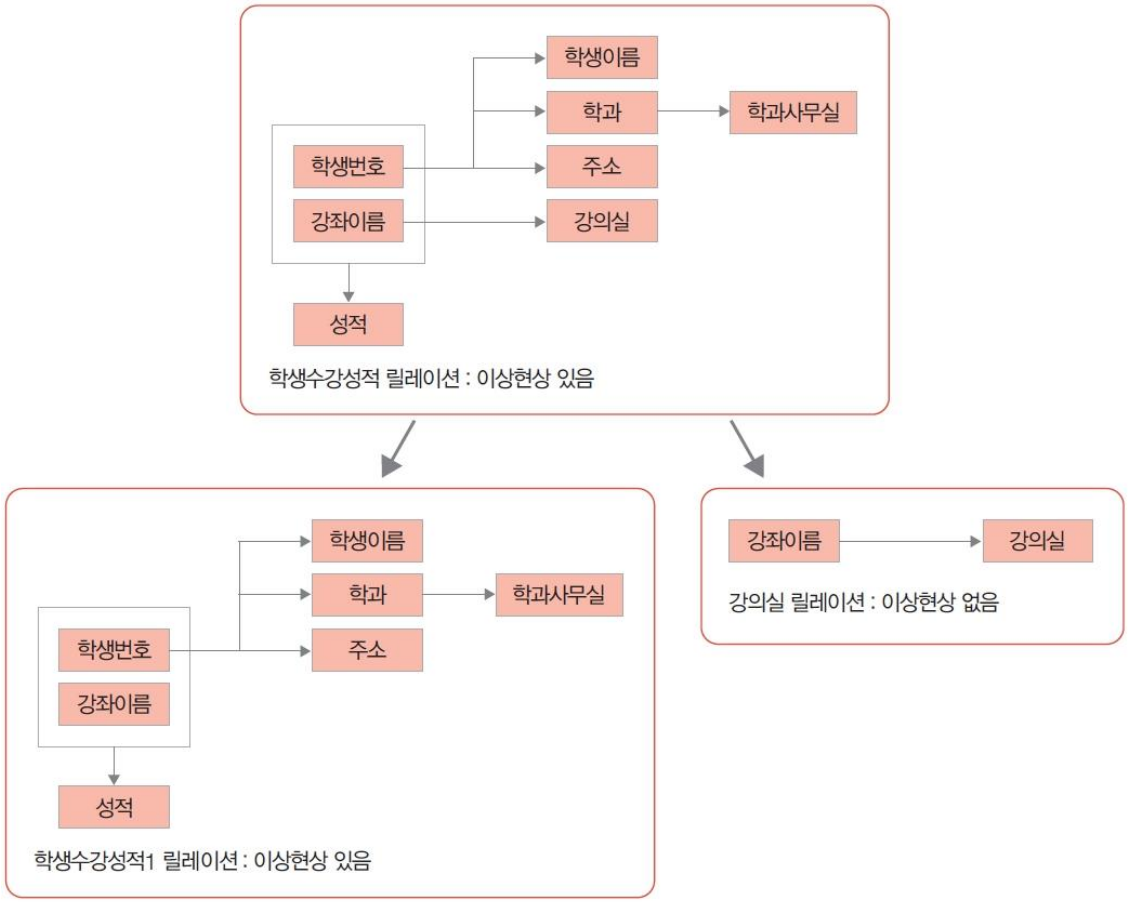
03 정규화

1. 정규화 과정
2. 무손실 분해
3. 정규화 정리

릴레이션의 분해 과정

❖ 정규화(normalization)

- 이상현상이 발생하는 릴레이션을 분해하여 이상현상을 없애는 과정



▲ 릴레이션의 이상현상을 없애려면 함수 종속성을 분석하여 테이블을 단계적으로 분해해야 한다

그림 7-14 릴레이션의 분해 과정

1. 정규화 과정

- 이동 수단의 유형에 따라 안전도 등급을 구분할 수 있는 것과 비슷하게 릴레이션을 '정규형'이라는 개념으로 구분함
- 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

그림 7-15 이동 수단과 릴레이션의 등급 구분

1. 정규화 과정

❖ 제1정규형

정의 | 제1정규형(1NF, First Normal Form)

릴레이션 R의 모든 속성값이 원자값을 가지면 제1정규형이라고 한다.

원문 A relation in which the intersection of each row and column contains one and only one value.

- 제1정규형으로 변환
  - 관계 데이터베이스에서 릴레이션의 속성값은 반드시 원자값이어야 함
  - 고객취미들(이름, 취미) 릴레이션을 고객취미(이름, 취미) 릴레이션으로 바꾸어 저장하면 제1정규형을 만족함

고객취미들(이름, 취미)
→ 고객취미(이름, 취미)

고객취미들(이름, 취미)

이름	취미
김연아	인터넷
추신수	영화, 음악
박세리	음악, 쇼핑
김연경	음악
박지성	게임

→

고객취미(이름, 취미)

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

그림 7-16 속성값이 원자값을 갖도록 분해

1. 정규화 과정

❖ 제2정규형

정의 | 제2정규형(2NF, Second Normal Form)

릴레이션 R이 제1정규형이고, 기본키가 아닌 속성이 기본키에 완전 함수 종속일 때 제2정규형이라고 한다.

원문 A relation that is in first normal form and every non–primary key attribute is fully functionally dependent on the primary key.

정의 | 완전 함수 종속

A와 B가 릴레이션 R의 속성이고 A → B 종속성이 성립할 때, B가 A의 속성 전체에 함수 종속하고 부분집합 속성에 함수 종속하지 않을 경우 완전 함수 종속(full functional dependency)이라고 한다.

- A → B 종속성에서 A의 속성 일부를 제거해도 종속성이 여전히 성립하는 경우 불완전 함수 종속 혹은 부분 함수 종속이라고 함

1. 정규화 과정

- 수강강좌 릴레이션은 수강 정보와 강의실 정보를 저장함

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	3.5

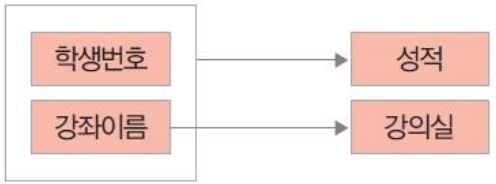


그림 7-17 수강강좌 릴레이션

❖ 이상현상

- 삭제이상
  - 402번 학생이 수강을 취소하면 스포츠경영학 과목의 강의실에 대한 정보가 사라짐
- 삽입이상
  - 컴퓨터입문 과목이 개설되어 공학관 112호를 사용하게 되었는데 아직 신청한 학생이 없음. 이 경우 수강강좌 릴레이션에 학생번호와 성적을 NULL 값으로 삽입해야 하는 문제가 발생한
- 수정이상
  - 데이터베이스 강의실을 공학관 113호로 변경할 경우 데이터 불일치가 발생할 가능성이 있음

1. 정규화 과정

❖ 이상현상의 원인

- 기본키가 아닌 속성이 기본키에 완전 함수 종속이 아닌 불완전 함수 종속되어 있으면 이상현상이 발생함

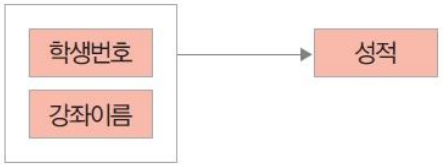
❖ 제2정규형으로 변환

- 수강강좌 릴레이션에서 이상현상을 일으키는 (강좌이름, 강의실)을 분해

수강강좌(학생번호, 강좌이름, 강의실, 성적)
→ 수강(학생번호, 강좌이름, 성적), 강의실(강좌이름, 강의실)

수강강좌

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



강의실

강좌이름	강의실
데이터베이스	공학관 110
스포츠경영학	체육관 103
자료구조	공학관 111

그림 7-18 수강강좌 릴레이션을 수강, 강의실 릴레이션으로 분해

1. 정규화 과정

❖ 제3정규형

정의 | 제3정규형(3NF, Third Normal Form)

릴레이션 R이 제2정규형이고, 기본키가 아닌 속성이 기본키에 비이행적(non-transitive)으로 종속할 때(직접 종속) 제3정규형이라고 한다. 이행적 종속이란 A → B, B → C가 성립할 때 A → C가 성립되는 함수 종속성을 말한다.

원문 A relation that is in first and second normal form and in which no non-primary key is transitively dependent on the primary key.

- 계절학기 릴레이션은 수강 정보와 수강료 정보를 저장. 단, 계절학기이기 때문에 학생은 한 강좌만 신청할 수 있다고 가정, 후보키를 먼저 찾아보자.

계절학기

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

그림 7-19 계절학기 릴레이션

1. 정규화 과정

❖ 이상현상

- 삭제이상
  - 402번 학생이 수강을 취소하면 스포츠경영학 과목의 수강료에 대한 정보가 사라짐
- 삽입이상
  - 컴퓨터입문 과목이 개설되어 수강료 15,000원을 삽입해야 하는데, 아직 신청한 학생이 없어 학생번호를 NULL 값으로 삽입해야 하는 문제가 발생한
- 수정이상
  - 데이터베이스 수강료를 15,000원으로 변경할 경우 데이터 불일치가 발생할 가능성이 있음

❖ 이상현상의 원인

- 이상현상의 원인은 계절학기 릴레이션의 함수 종속성 다이어그램을 보면 알 수 있음
- (학생번호 → 강좌이름), (강좌이름 → 수강료)로 수강료는 기본키에 이행적으로 종속되어 있음. 이때 이상현상이 발생함

1. 정규화 과정

❖ 제3정규형으로 변환

계절학기(학생번호, 강좌이름, 수강료)
→ 계절수강(학생번호, 강좌이름), 수강료(강좌이름, 수강료)

계절수강

학생번호	강좌이름
501	데이터베이스
401	데이터베이스
402	스포츠경영학
502	자료구조

수강료

강좌이름	수강료
데이터베이스	20000
스포츠경영학	15000
자료구조	25000

그림 7-20 계절학기 릴레이션을 계절수강, 수강료 릴레이션으로 분해

1. 정규화 과정

❖ BCNF

정의 | BCNF 정규형(Boyce Codd Normal Form)

릴레이션 R에서 함수 종속성 X → Y가 성립할 때 모든 결정자 X가 후보키이면 BCNF 정규형이라고 한다.

원문 A relation is in BCNF if and only if every determinant is a candidate key.

- 한 학생은 한 개 이상의 특강을 신청할 수 있고, 교수는 특강을 하나만 담당할 수 있다고 가정, 후보키를 먼저 찾아보자.

특강수강

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

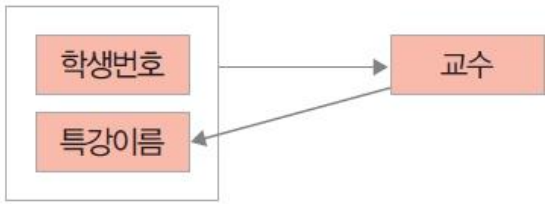


그림 7-21 특강수강 릴레이션

1. 정규화 과정

❖ 이상현상

- 삭제이상
  - 402번 학생이 수강을 취소하면 '인간과 동물' 특강을 담당하는 교수 정보가 사라짐
- 삽입이상
  - 새로운 특강을 개설하여 최교수가 담당하게 되면, 아직 신청한 학생이 없어 학생번호를 NULL 값으로 삽입해야 하는 문제가 발생함
- 수정이상
  - 김교수의 특강 제목을 '소셜네트워크'에서 '소셜네트워크 분석'으로 변경할 경우 데이터 불일치가 발생할 가능성이 있음

❖ 이상현상의 원인

- 특강수강 릴레이션의 함수 종속성 다이어그램을 보면 알 수 있음
- 결정자이면서 후보키가 아닌 속성이 존재하면 이상현상이 발생함

1. 정규화 과정

❖ BCNF 정규형으로 변환

특강수강(학생번호, 특강이름, 교수)
→ 특강신청(학생번호, 교수), 특강교수(특강이름, 교수)

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

그림 7-22 특강수강 릴레이션을 특강신청, 특강교수 릴레이션으로 분해

2. 무손실 분해

❖ 무손실 분해

- 릴레이션 R을 두 개의 릴레이션 R1과 R2로 분해했을 때, 다시 조인하면 원래의 릴레이션 R이 만들어지는 것을 의미

정의 | 무손실 분해

릴레이션 R을 릴레이션 R1과 R2로 분해할 때, $R1 \bowtie R2 = R$이면 무손실 분해(lossless-join decomposition)라고 한다. 무손실 분해를 위한 조건은 $R1 \cap R2 \rightarrow R1$이나 $R1 \cap R2 \rightarrow R2$ 중 하나를 만족해야 한다.

특강수강

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

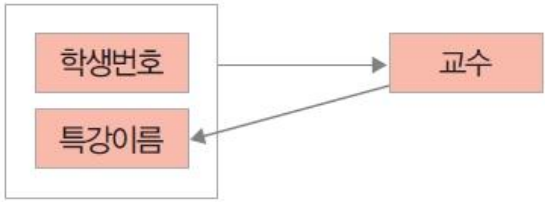


그림 7-23 특강수강 릴레이션

2. 무손실 분해

- 특강수강 릴레이션은 모든 결정자가 후보키가 아니므로 이상현상이 발생. 따라서 [표 7-4]와 같이 분해함

표 7-4 특강수강 릴레이션의 분해

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

[분해1]의 경우 R1, R2를 다시 조인하면 원래 릴레이션이 됨

[분해2]의 경우 R3, R4 릴레이션

- [분해 2]의 경우 R3, R4 릴레이션은 [그림

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

그림 7-24 특강수강 릴레이션을 R3, R4 릴레이션으로 분해

2. 무손실 분해

[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

그림 7-24 특강수강 릴레이션을 R3, R4 릴레이션으로 분해

R3, R4 릴레이션을 다시 조인하면 [그림 7-25]와 같음

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

그림 7-25 특강수강 릴레이션과 R3 ⋈ R4 릴레이션의 비교

3. 정규화 정리

❖ 정규형

- [그림 7-26]과 같이 정규화되거나 되지 않은 모든 릴레이션(예를 들어 속성값이 원자값이 아니어서 릴레이션의 정의를 만족하지 못하는 경우 등을 말하며, 이 경우 릴레이션이라고 부르지 않는다)부터 제1정규형, 제2정규형, 제3정규형, BCNF, 제4정규형, 제5정규형까지 존재
- 대부분의 릴레이션은 BCNF까지 정규화하면 실제적인 이상현상은 없어짐
- 실무에서는 보통 제3정규형이나 BCNF까지 정규화를 진행함

그림 7-26 정규형의 포함 관계

3. 정규화 정리

예제 7-3

릴레이션 R(A, B, C, D)는 다음과 같은 함수 종속성이 성립한다. 물음에 답하시오.

$A \rightarrow B, B \rightarrow C, C \rightarrow D$

① 릴레이션 R의 후보키를 찾아보시오.

② 릴레이션 R은 어떤 정규형인가?

③ 릴레이션을 다음과 같이 분해한다면 무손실 분해에 해당하는가?

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

① A가 B를 결정하고(A → B) B가 C를 결정하면(B → C) A는 BC를 결정한다(A → BC).

② 이행적 종속성이 있으므로 제3정규형은 만족하지 못하지만 제2정규형은 만족한다.

③ R1과 R2의 공통 속성은 C이다. 그리고 C는 R2의 후보키이므로 무손실 분해다.

3. 정규화 정리

예제 7-4

릴레이션 R(A, B, C)는 다음과 같은 함수 종속성이 성립한다. 물음에 답하시오.

AB → C, C → A

① 릴레이션 R의 후보키를 찾아보시오.

② 릴레이션 R은 어떤 정규형인가?

③ 릴레이션을 다음과 같이 분해한다면 무손실 분해에 해당하는가?

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

① AB → C이므로 AB → ABC도 만족한다. 따라서 AB는 릴레이션 전체를 결정하는 후보키다. 같은 방식으로 생각해 보면 BC도 릴레이션 전체를 결정하는 후보키가 된다.

② 제2정규형과 제3정규형을 만족한다. 그러나 C → A에서 결정자 C는 후보키가 아니므로 BCNF 조건은 만족하지 못한다. 따라서 제3정규형이다.

③ R1과 R2의 공통 속성은 C다. 그리고 C는 R1의 후보키이므로 무손실 분해이다.

3. 정규화 정리

예제 7-5

릴레이션 R(A, B, C, D)는 다음과 같은 함수 종속성이 성립한다. 물음에 답하시오.

AB → C, C → A, C → D

① 릴레이션 R의 후보키를 찾아보시오.

② 릴레이션 R은 어떤 정규형인가?

③ 릴레이션을 다음과 같이 분해한다면 무손실 분해에 해당하는가?

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

① AB가 C를 결정하고(AB → C), C가 D를 결정하면(C → D) AB는 CD를 결정한다(AB → CD). 즉 AB → ABCD이므로 AB는 릴레이션 전체를 결정하는 후보키다. 같은 방식으로 풀어보면 BC도 릴레이션 전체를 결정하는 후보키가 된다.

② 제2정규형은 만족하지만, AB → C, C → D의 이행적 종속성이 있으므로 제3정규형을 만족하지 못한다. 따라서 제2정규형이다.

③ R1과 R2의 공통 속성은 C다. 그리고 C는 R2의 후보키이므로 무손실 분해다.

연습문제 (Q7.3)

05 제1정규형에서 제2정규형이 되기 위한 조건은?

① 모든 도메인이 원자값이어야 한다.

② 키가 아닌 모든 속성이 기본키에 이행적으로 함수 종속되지 않아야 한다.

③ 다치 종속이 제거되어야 한다.

④ 키가 아닌 모든 속성이 기본키에 완전 함수 종속되어야 한다.

06 제2정규형에서 제3정규형이 되기 위한 조건은?

① 이행적 함수 종속을 제거해야 한다.

② 부분 함수 종속을 제거해야 한다.

③ 다치 종속을 제거해야 한다.

④ 결정자가 후보키가 아닌 함수적 종속을 제거해야 한다.

07 제3정규형에서 보이스코드 정규형(BCNF)이 되기 위한 조건은?

① 원자값이 아닌 도메인을 분해한다.

② 부분 함수 종속을 제거해야 한다.

③ 이행적 함수 종속을 제거해야 한다.

④ 결정자가 후보키가 아닌 함수적 종속을 제거해야 한다.

04 정규화 연습(부동산 데이터베이스)

정규화 연습(부동산 데이터베이스)

- 부동산 정보를 저장하는 부동산 데이터베이스를 예로 들어 정규화를 연습해보기

(릴레이션)

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

표 7-5 부동산 릴레이션의 함수 종속성

함수 종속성	설명
필지번호 → 주소, 공시지가	땅에 대한 고유 번호이며 필지에 대하여 주소와 공시가격이 주어진다.
소유자이름 → 전화번호	소유자는 하나의 전화번호를 갖는다.
주민등록번호 → 소유자이름	사람마다 고유한 주민등록번호가 있다.

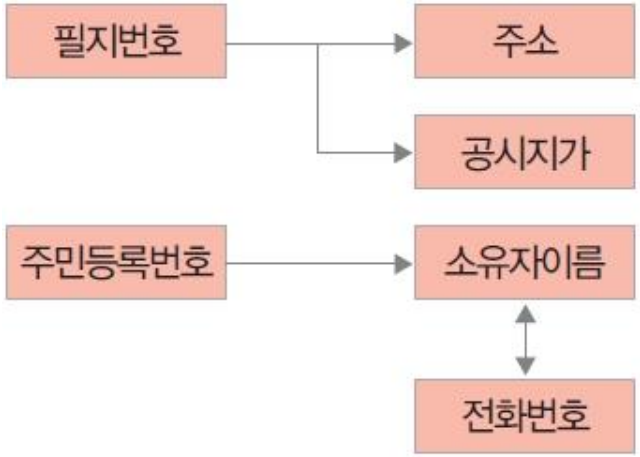


그림 7-27 부동산 릴레이션의 함수 종속성 다이어그램

정규화 연습(부동산 데이터베이스)

- 여기서 부동산 릴레이션은 다음의 두 가지 사례에 대하여 생각해 볼 수 있음

  공동 소유: 한 필지를 두 사람 이상이 공동으로 소유하는 경우

  단독 소유: 한 필지를 한 사람만 소유하는 경우

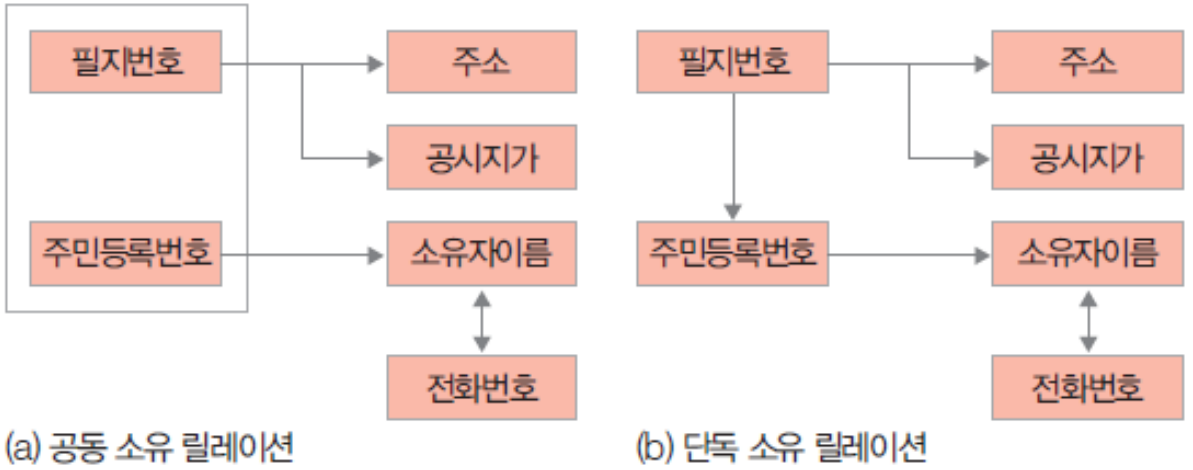


그림 7-28 부동산 릴레이션의 두 사례에 대한 함수 종속성 다이어그램

정규화 연습(부동산 데이터베이스)

❖ 공동 소유

- 한 필지를 두 사람 이상이 공동으로 소유하는 경우 이상현상이 발생
- 기본키인 필지번호와 주민등록번호를 보면 각각 단독으로 다른 속성을 결정하는 결정자
  - 즉, 부분 함수 종속성을 가져 제2정규형의 정의에 위배됨
- 결과적으로 공동 소유 릴레이션은 다음과 같이 분해됨

부동산소유(필지번호, 주민등록번호)

부동산필지(필지번호, 주소, 공시지가)

소유자(주민등록번호, 소유자이름, 전화번호)

❖ 단독 소유

- 한 필지를 한 사람만 소유하는 경우도 마찬가지로 이상현상이 발생
- 모든 속성이 기본키인 필지번호에 완전 함수 종속이므로 제2정규형은 만족하지만, 이행 종속이 남아 있어서 제3정규형은 만족하지 못함

부동산소유(필지번호, 주소, 공시지가, 주민등록번호)

소유자(주민등록번호, 소유자이름, 전화번호)

연습문제 (Q7.4)

15 다음 릴레이션 R에서 성립하는 최상위 정규형은 무엇인가?

(1) 릴레이션: R(H, I, J, K, L, M, N, O)
기본키: (H, I)
함수 종속성: (H, I) → J, K, L
J → M
K → N
L → O

(2) 릴레이션: R(D, O, N, T, C, R, Y)
함수 종속성: (D, O) → N, T, C, R, Y
(C, R) → D
D → N

(3) 릴레이션: R(A, B, C, D, E, F, G, H)
기본키: (A, D)
함수 종속성: CH → G
A → BC
B → CFH
E → A
F → EG

(4) 릴레이션: R(A, B, C, D, E)
함수 종속성: BD → A
AB → C
D → A
B → C
C → E

17 릴레이션 Book(booktitle, authorname, booktype, listprice, authorgroup, publisher)에서 함수 종속성은 다음과 같다. 물음에 답하시오.

booktitle, authorname → publisher
booktitle → booktype
booktype → listprice
authorname → authorgroup

(1) Book 릴레이션은 어떤 정규형인가? 그 이유를 설명하시오.
(2) 정규화를 수행하시오.

요약

1. 이상현상
2. 함수 종속성
3. 정규화
4. 무손실 분해