

### 3. 제 1,2,3 정규형

// 제1정규형(1NF; first normal form)

제1정규형은 릴레이션의 모든 도메인이 원자값으로 구성되어 있다.

이 정의는 비정규형 릴레이션을 정규형으로 정규화만 하면 제1정규형에 속한다는 뜻이다.

다음 수강지도 릴레이션은 제1정규형에 해당한다.

수강지도

학번	과목	점수	지도교수	학과
1	디비	90	P1	전산
1	소공	50	P1	전산
2	디비	70	P2	토목
3	디비	80	P1	전산
3	소공	80	P1	전산
3	컴일	60	P1	전산
4	디비	70	P3	기계
4	소공	80	P3	기계

수강지도(학번, 과목, 점수, 지도교수, 학과)

기본키 = {학번, 과목}

↓ 함수종속

{학번, 과목} → 점수

학번 → 지도교수

학번 → 학과

지도교수 → 학과

// 함수종속의 의미

- 학번과 과목의 조합으로 점수를 식별할 수 있고
- 각 학생은 한명의 지도교수를 가질 수 있고, 하나의 학과에만 속한다.
- 각 지도교수는 하나의 학과에만 속한다.

// 수강지도 릴레이션에서 중복된 표현 존재

- 학번 1의 지도교수는 P1이다. / 학번 1의 학과는 전산이다.
- 학번 4의 지도교수는 P3이다. / 학번 4의 학과는 기계이다. 등

// 수강지도 릴레이션은 불필요하게 데이터 중복이 여러 곳에서 존재하므로 이상 현상이 발생한다.

삽입 이상	<ul style="list-style-type: none"> <li>• 학생이 어떤 과목을 등록할 때까지 학생의 지도교수를 삽입할 수 없다.</li> <li>• 예 : 학번 5인 학생의 지도교수가 P4라는 사실만을 삽입할 수 없다.(과목이 null이 되므로)</li> </ul>
삭제 이상	<ul style="list-style-type: none"> <li>• 학번 2인 학생이 등록을 취소하면, 학생의 지도교수가 P2라는 사실을 잃어 버린다.</li> <li>• 학번 2인 튜플은 하나만 존재하므로</li> <li>• 학생의 등록 취소는 학생의 지도교수가 P2라는 것을 삭제하려는 것이 아니다.</li> </ul>
갱신 이상	<ul style="list-style-type: none"> <li>• 여러 번 중복된 학번 4인 학생의 지도교수가 P3에서 P4로 변경되면,</li> <li>• 학번 4인 학생이 포함된 모든 튜플의 지도교수를 P3에서 P4로 변경해야한다.</li> </ul>

2 <http://cafe.daum.net/pass365>(홍재연)

수강지도 릴레이션에서 이상이 발생하는 세부적인 요인은 기본키가 아닌 속성들이 기본키에 **완전함수종속**의 관계가 되지 못하고, **부분함수종속**이 되기 때문이다. 따라서, 수강 릴레이션을 2개의 릴레이션으로 분해하여 부분함수종속을 제거하면 이상 발생을 해결할 수 있다.

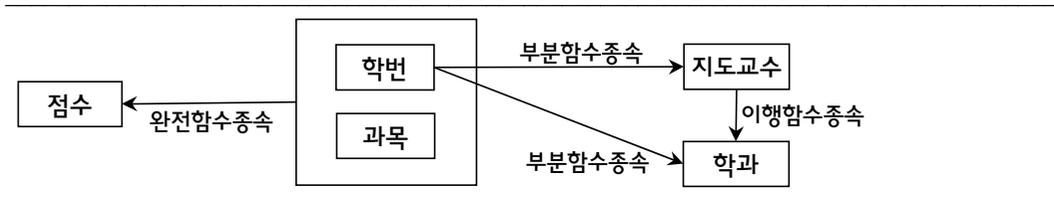
수강지도

학번	과목	점수	지도교수	학과
1	디비	90	P1	전산
1	소공	50	P1	전산
2	디비	70	P2	토목
3	디비	80	P1	전산
3	소공	80	P1	전산
3	컴일	60	P1	전산
4	디비	70	P3	기계
4	소공	80	P3	기계

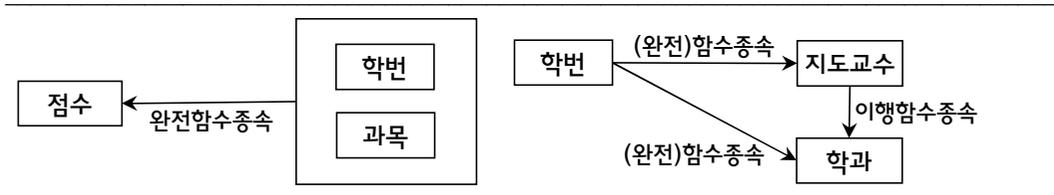
수강지도(학번, 과목, 점수, 지도교수, 학과)  
 기본키 = {학번, 과목}

↓ 함수종속  
 ↓  
 {학번, 과목} → 점수  
 학번 → 지도교수  
 학번 → 학과  
 지도교수 → 학과

↓ 수강지도 릴레이션에 존재하는 함수종속의 관계를 그림으로 분석

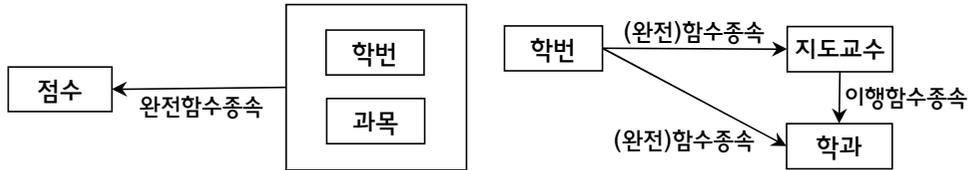


↓ 수강지도 릴레이션에서 부분함수종속 제거



- 부분함수종속이 제거되었을 때, 제2정규형이라고 한다.
- 그러면, 모든 제2정규형에서 삽입, 삭제, 갱신 연산에서 이상이 발생되지 않을까?
- 답은 아니다! 제2정규형 중에서 이상이 발생하는 릴레이션이 있을 수 있다.
- 구체적으로 살펴보기로 한다.

다음은 수강지도 릴레이션이 수강과 지도 릴레이션으로 분할된 모습이다.



수강

학번	과목	점수
1	디비	90
1	소공	50
2	디비	70
3	디비	80
3	소공	80
3	컴일	60
4	디비	70
4	소공	80

지도

학번	지도교수	학과
1	P1	전산
2	P2	토목
3	P1	전산
4	P3	기계

수강지도 릴레이션에서 부분함수종속을 제거함으로써 제1정규형에서 발생하는 문제점을 해결하였다.

즉, 릴레이션 수강과 지도는 모두 제2정규형에 해당한다.

어떤 릴레이션이 1NF이고,  
기본키에 속하지 않은 모든 속성이 기본키에 완전함수종속이면 제2정규형에 속한다.

여기서 주목할 것은

수강	<ul style="list-style-type: none"> <li>• 수강 릴레이션은 단순한 구조로 더 이상 문제점이 없다.</li> <li>• 수강 릴레이션은 기본키와 무관한 속성은 모두 제거되었다.(완전함수종속)</li> <li>• 수강 릴레이션은 기본키에 의한 삽입, 삭제 갱신 이상은 발생되지 않는다.</li> <li>• 기본키에 의한 이상이 발생되지 않으므로 더 이상 분해하지 않아도 된다.</li> </ul>
지도	<ul style="list-style-type: none"> <li>• 지도 릴레이션은 내부적으로 복잡한 함수종속의 관계를 가지고 있다.(이상이 발생됨)</li> <li>• 복잡한 함수종속 관계는 내부적으로 이행함수종속이 존재하는 것이다.</li> <li>• 이행함수종속이 존재하면 이상이 발생할 수 있으므로 릴레이션을 분해해야 한다.</li> </ul>



탐구

릴레이션 분해 과정

수강지도

학번	과목	점수	지도교수	학과
1	디비	90	P1	전산
1	소공	50	P1	전산
2	디비	70	P2	토목
3	디비	80	P1	전산
3	소공	80	P1	전산
3	컴일	60	P1	전산
4	디비	70	P3	기계
4	소공	80	P3	기계

수강지도(학번, 과목, 점수, 지도교수, 학과)

기본키 = {학번, 과목}

↓ 함수종속

{학번, 과목} → 점수

학번 → 지도교수

학번 → 학과

지도교수 → 학과

↓ 수강지도 릴레이션 분해 과정을 살펴본다.

Step 1. 같은 결정자를 가지는 그룹으로 나눈다.

- {학번, 과목} → 점수
- 학번 → (지도교수, 학과)

Step 2. 각 그룹별로 릴레이션 스킴을 만든다.

- 수강(학번, 과목, 점수)  
기본키 : {학번, 과목}, 외래키 : {학번} → 외래키는 정보 손실 방지
- 지도(학번, 지도교수, 학과)  
기본키 : {학번}

Step 3. 릴레이션 스킴을 근거로 릴레이션을 작성한다.

수강

학번	과목	점수
1	디비	90
1	소공	50
2	디비	70
3	디비	80
3	소공	80
3	컴일	60
4	디비	70
4	소공	80

지도

학번	지도교수	학과
1	P1	전산
2	P2	토목
3	P1	전산
4	P3	기계

- 최소의 중복 : 외래키 '학번'을 '수강' 릴레이션에 넣으면 최소의 중복이 된다.
- 이유는 한 학생은 여러 과목을 수강할 수 있다.(학생 : 수강 = 1 대 다)

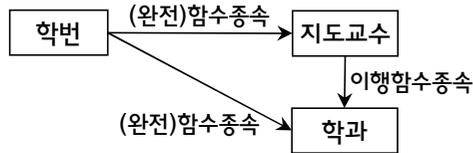
// 제2정규형(2NF; second normal form)

어떤 릴레이션이 1NF이고,  
기본키에 속하지 않은 모든 속성이 기본키에 완전함수종속이면 제2정규형에 속한다.

다음 지도 릴레이션은 제2정규형에 해당한다.

지도

학번	지도교수	학과
1	P1	전산
2	P2	토목
3	P1	전산
4	P3	기계



주어진 지도 릴레이션은 제2정규형은 만족하지만, 이상 현상이 발생할 수 있다.

삽입 이상	<ul style="list-style-type: none"> <li>어떤 지도교수가 어느 학과에 속한다는 사실만을 삽입할 수 없다.</li> <li>이유는 기본키인 학번이 null이 되므로</li> <li>삽입할 수 있기 위해서는 지도교수의 지도를 받는 학생이 존재해야 한다.</li> </ul>
삭제 이상	<ul style="list-style-type: none"> <li>학번 2인 학생이 지도교수 관계를 취소하여 튜플이 삭제되면</li> <li>지도교수 P2가 토목과에 속해 있다는 정보까지 같이 삭제된다.</li> <li>학생의 지도교수 취소는 지도교수 P2가 토목과에 속한다는 것을 삭제하려는 것이 아니다.</li> </ul>
갱신 이상	<ul style="list-style-type: none"> <li>어느 지도교수의 학과가 변경되면, 그 지도교수가 포함된 튜플의 학과는 모두 변경해야한다.</li> <li>지도교수 P1이 전산과에서 전기과로 변경되면, 학번 1과 3에 있는 학과를 모두 전기과로 변경해야한다.</li> <li>그렇지 않으면, 지도교수 P1은 전산과와 전기과라는 상이한 2개의 값을 가지는 모순 발생</li> <li>데이터베이스는 모순이 발생되지 않도록 해야 한다.</li> </ul>

↓  
↓ 주어진 지도 릴레이션에서 이상 현상이 발생하는 이유  
↓

- 상이한 2개의 종속 정보를 하나의 릴레이션에서 표현하고 있기 때문이다.
- 즉, 이행함수종속의 관계를 하나의 릴레이션에서 표현하고 있기 때문이다.
- 이행함수종속의 관계가 제거되도록 릴레이션을 분해하면, 이상 현상을 해결할 수 있다.

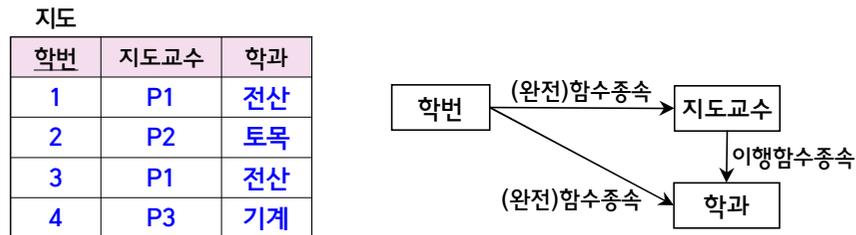
// 제3정규형(3NF; third normal form)

어떤 릴레이션이 2NF이고,  
기본키에 속하지 않은 모든 속성이 기본키에 이행함수종속이 아니면 제3정규형에 속한다.

// 이행함수종속(transitive FD; 전이함수종속)

- 함수종속 관계에서 'A→B일 때 B→C가 성립하면 A→C가 성립한다.'
- 이때, 속성 C는 속성 A에 '이행함수종속'이라고 한다.
- 예 : '학번→학과, 학과→등록금'이면 '학번→등록금'

다음 지도 릴레이션은 제2정규형에 해당한다. 제3정규형이 되도록 분해하기로 한다.



이행함수종속이 제거되도록 릴레이션을 분해하면



분해된, 학생지도 릴레이션과 교수학과 릴레이션은 모두 제3정규형에 해당한다.

**기출문제 분석**

1. 학과\_등록 테이블이 다음과 같을 때, 학과\_등록 테이블에 대한 설명으로 옳지 않은 것은? (단, 밑줄은 기본키이다) [2016년 국가 7급]

학과\_등록

학번	과목번호	학점	과목이름
100	A01	A	JAVA
101	B01	D	C#
101	B03	A	DB 시스템
100	B01	B	C#
200	A02	C	Direct X

- ① 과목이름을 수정하려 할 때 갱신이상이 발생할 수 있다.
- ② 제2정규형을 만족시킨다.
- ③ 학번이 200인 학생이 수강을 철회하면 삭제이상이 발생한다.
- ④ 아직 수강생이 없는 새로운 과목을 삽입할 때, 삽입이상이 발생한다.

♣ 정규형

- 
- ① 과목이름을 수정하려 할 때 갱신이상이 발생할 수 있다.(○)  
→ 과목이름 C#을 한군데만 수정하면 데이터 불일치가 발생한다.
  - ② 제2정규형을 만족시킨다.(×)  
→ 기본키는 (학번, 과목번호)이다.  
→ 과목번호->과목이름은 부분함수종속 관계이다.  
→ 부분함수종속이 존재한다.  
→ 제2정규형을 만족시키지 못한다.(제2정규형은 부분함수종속 제거되어야 한다)
  - ③ 학번이 200인 학생이 수강을 철회하면 삭제이상이 발생한다.(○)  
→ 학번이 200인 학생이 수강을 철회하면, Direct X 과목이 삭제된다.
  - ④ 아직 수강생이 없는 새로운 과목을 삽입할 때, 삽입이상이 발생한다.(○)  
→ 수강생이 없는 새로운 과목 삽입은 불가능하다.(학번이 기본키)  
→ 기본키는 null이 될 수 없다. 해서, 삽입이 불가능하다.

2. 함수적 종속성과 정규화에 대한 설명으로 옳지 않은 것은? [2018년 국가 7급]

- ① 함수적 종속성  $X \rightarrow Y$ 는 릴레이션 내 임의의 두 튜플에서 속성 X의 값이 같을 경우, 속성 Y의 값도 항상 같음을 의미한다.
- ② 함수적 종속성은 릴레이션 상태(외연)의 특성이다.
- ③ 제2정규형에서 릴레이션의 모든 비주요 속성들은 릴레이션의 기본키에 완전하게 함수적으로 종속한다.
- ④ 정규화는 함수적 종속성과 기본키를 기반으로 주어진 릴레이션 스키마를 분석하는 과정이다.

☞ 함수적 종속성과 정규화

- 함수적 종속성은 릴레이션 상태(외연)의 특성이다.(x)  
→ 함수적 종속성은 튜플을 구성하는 속성들 사이의 특성이다.(속성들 사이의 종속 관계)
- 예 : {직원 ID} → {직원 생일}  
{직원 ID}는 정확히 하나의 {직원 생일}을 가진다.

정답 : ②

3. 다음 릴레이션 R에서 함수종속성이 성립하는 경우를 연결한 것으로 가장 옳은 것은? [2022년 군무원 7급]

R		
A	B	C
a	b	c
d	e	f
g	e	f
d	a	h

- ㉠  $A \rightarrow B$
- ㉡  $B \rightarrow C$
- ㉢  $(B, C) \rightarrow A$
- ㉣  $(A, B) \rightarrow C$

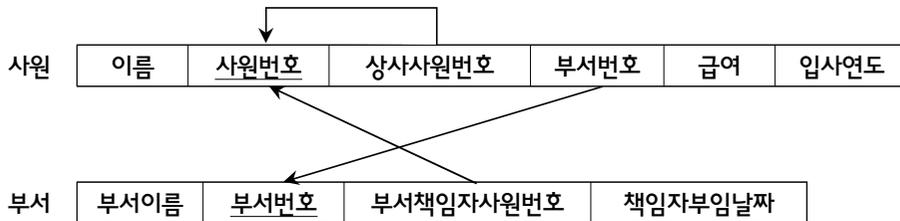
- ① ㉠ ㉡
- ② ㉡ ㉢
- ③ ㉠ ㉣
- ④ ㉡ ㉣

☞ 함수종속성 :  $X \rightarrow Y$ 에서 X의 각각에 대해 시간과 무관하게 항상 속성 Y의 값이 오직 하나만 연관

- ㉠  $A \rightarrow B$  :  $d \rightarrow \{e, a\}$ 이 있으므로 함수종속성이 성립하지 않음
- ㉡  $B \rightarrow C$  :  $b \rightarrow c, e \rightarrow f, a \rightarrow h$ 이므로 함수종속성이 성립
- ㉢  $(B, C) \rightarrow A$  :  $ef \rightarrow \{d, g\}$ 이 있으므로 함수종속성이 성립하지 않음
- ㉣  $(A, B) \rightarrow C$  :  $ab \rightarrow c, de \rightarrow f, ge \rightarrow f, da \rightarrow h$ 이므로 함수종속성이 성립

정답 : ④

4. 다음 데이터베이스 스키마에 대한 설명으로 옳지 않은 것은? (단, 밑줄이 있는 속성은 그 릴레이션의 기본키를, 화살표는 외래키 관계를 의미한다) [2015년 지방 9급]



- ① 외래키는 동일한 릴레이션을 참조할 수 있다.
- ② 사원 릴레이션의 부서번호는 부서 릴레이션의 부서번호 값 중 하나 혹은 null이어야 한다는 제약조건은 참조무결성을 의미한다.
- ③ 신입사원을 사원 릴레이션에 추가할 때 그 사원의 사원번호는 반드시 기존 사원의 사원번호와 같지 않아야 한다는 제약조건은 제1정규형의 원자성과 관계있다.
- ④ 부서 릴레이션의 책임자부임날짜는 반드시 그 부서책임자의 입사연도 이후이어야 한다는 제약조건을 위해 트리거(trigger)와 주장(assertion)을 사용할 수 있다.

☞ 데이터베이스 스키마

- 신입사원을 사원 릴레이션에 추가할 때 그 사원의 사원번호는 반드시 기존 사원의 사원번호와 같지 않아야 한다는 제약조건은 제1정규형의 원자성과 관계있다.(x)
- 주어진 내용은 개체 무결성과 관계가 있다.
- 기본키(Primary Key) : Null을 허용하지 않고, 중복된 값도 허용하지 않는다.(유일성)
- 제1정규형은 릴레이션의 모든 속성이 원자값으로 구성되어 있다.

// 트리거(trigger)와 주장(assertion)

- 먼저, 트리거와 주장은 데이터베이스 무결성 제약조건을 위해 사용할 수 있다.
- 트리거는 데이터베이스가 갱신될 때마다 DBMS가 자동으로 수행하는 프로시저이다.
- 트리거는 데이터베이스 무결성을 유지하기 위한 일반적이고 강력한 도구이다.
- 트리거는 제약조건을 위반했을 때 수행할 동작을 명시하는 것이다.
- 주장은 제약조건을 위반하는 연산은 수행되지 않도록 하는 것이다.
- 주장은 트리거보다 좀 더 일반적인 무결성 제약조건에 적용한다.
- 두 개 이상의 테이블에 영향을 미치는 제약조건을 명시하기 위해 사용될 수 있다.

5. 릴레이션  $R(A, B, C, D)$ 에 대해 <보기 1>이 성립하고, 이를 <보기 2>와 같이 분해하였을 때, (가) 릴레이션  $R$ 의 정규형과 분해 시 (나) 정보 손실 여부를 옳게 짝지은 것은? [2022년 서울 7급]

----<보기 1>-----

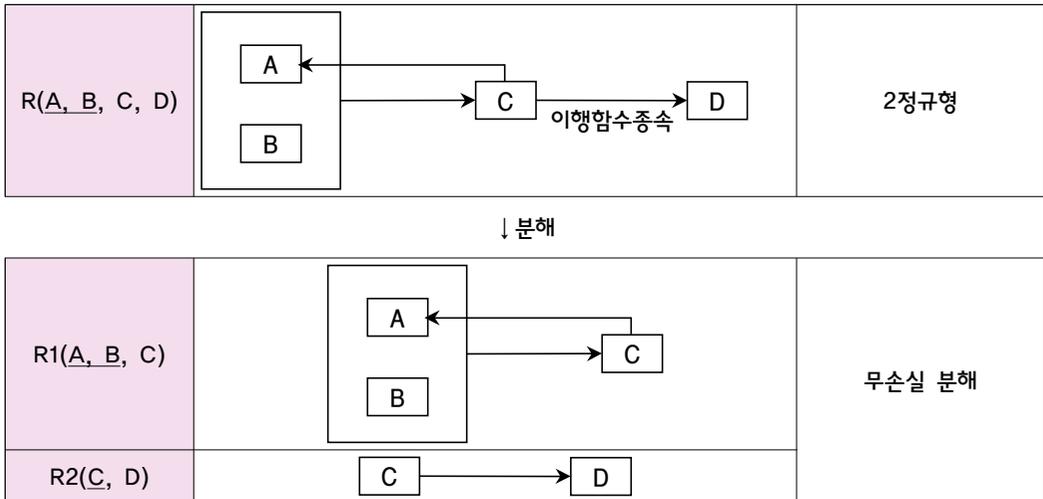
함수종속 :  $AB \rightarrow C, C \rightarrow A, C \rightarrow D$

----<보기 2>-----

분해 :  $R1(A, B, C), R2(C, D)$

- |        |        |
|--------|--------|
| (가)    | (나)    |
| ① 2정규형 | 무손실 분해 |
| ② 2정규형 | 손실 분해  |
| ③ 3정규형 | 무손실 분해 |
| ④ 3정규형 | 손실 분해  |

♣ 정규화



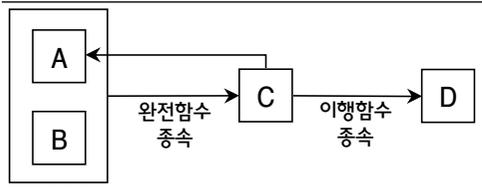
- 릴레이션  $R$ 은 **이행함수종속**이 존재하므로 제2정규형이다.
- 분해 결과는 **무손실 분해**이다. 이유는 속성  $C$ 가 무손실 분해를 보장한다.
- 분해 결과 :  $R1$ 과  $R2$ 는 제3정규형을 만족한다.

6. 릴레이션 R(A, B, C, D)는 다음과 같은 함수종속성이 성립한다. 다음 설명 중 가장 옳지 않은 것은? [2022년 군무원 7급]

$AB \rightarrow C, C \rightarrow A, C \rightarrow D$

- ① 릴레이션 전체를 결정하는 후보키는 AB와 BC이다.
- ② 릴레이션 R은 제3정규형이다.
- ③ 릴레이션 R은 R1(A, B, C), R2(C, D)와 같이 무손실 분해가 가능하다.
- ④ AB가 C를 결정하고, C가 D를 결정하면 AB는 CD를 결정한다.

☞ 정규형



- 릴레이션 R은 제2정규형이다.(이행함수종속이 있음)
- 후보키는 AB와 BC이다.(폐포를 이용한 후보키 찾기)
- {A, B}의 폐포 = {A, B}<sup>+</sup> = {A, B, C, D}
- {B, C}의 폐포 = {B, C}<sup>+</sup> = {A, B, C, D}

↓ 무손실 분해

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

정답 : ②

7. 다음은 민원인이 관공서에 발급 신청한 서류의 접수대장을 저장하기 위한 테이블이다. 이 테이블이 충족하는 가장 높은 정규형은?(단, 밑줄은 기본키이다) [2013년 국가 7급]

접수대장(접수일련번호, 민원인주민등록번호, 민원인성명, 신청서류, 수량, 접수일자)

- ① 제1정규형                      ② 제2정규형
- ③ 제3정규형                      ④ BCNF

☞ 정규형

- 이행함수종속이 존재한다.(제3정규형은 불가)  
 접수일련번호 → 민원인주민등록번호 → 민원인성명
- 부분함수종속은 없다.(제2정규형 가능)  
 → 기본키는 접수일련번호이다.(하나의 속성으로 구성)  
 → 해서, 이 테이블이 충족하는 가장 높은 정규형은 제2정규형이다.

정답 : ②

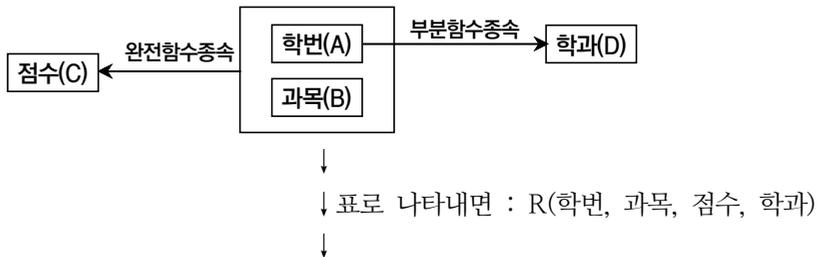
8. 다음은 속성(attribute) A, B, C, D와 4개의 튜플(tuple)로 구성되고 두 개의 함수종속  $AB \rightarrow C$ ,  $A \rightarrow D$ 를 만족하는 릴레이션을 나타낸다. ㉠과 ㉡에 들어갈 수 있는 속성 값이 옳게 짝지어진 것은? (단, A 속성의 도메인은 {a1, a2, a3, a4}이고, D 속성의 도메인은 {d1, d2, d3, d4, d5}이다) [2017년 지방 9급]

A	B	C	D
a1	b1	c1	d1
a1	b2	c2	㉠
㉡	b1	c1	d3
a4	b1	c1	d4

- ㉠                      ㉡
- ① d1                      a1
- ② d1                      a2 또는 a3
- ③ d5                      a2 또는 a4
- ④ d4                      a4

☞ 정규형

• 먼저, 주어진 문제는 다음 그림과 같은 함수종속을 가지는 릴레이션 구조이다.



학번(A)	과목(B)	점수(C)	학과(D)
a1	컴일	90	d1
a1	디비	80	㉠
㉡	컴일	90	d3
a4	컴일	90	d4

• 해서

- ㉠에 들어갈 수 있는 속성 값은 학과 d1이다. 학번 a1의 학과는 d1이므로
- ㉡에 들어갈 수 있는 속성 값은 학번 a2 또는 a3이다. 학과가 서로 다른 d3이므로  
→ 학번(A) 속성의 도메인은 {a1, a2, a3, a4}이고

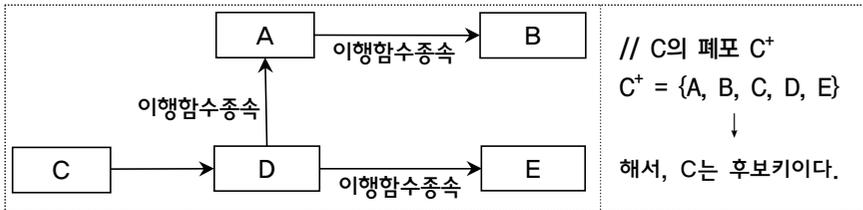
9. <보기>와 같은 함수적 종속성 집합 F를 가지는 릴레이션 R(A, B, C, D, E)을 여러 릴레이션으로 분해하고자 할 때 이에 대한 설명으로 가장 옳지 않은 것은? [2020년 서울 7급]

-----<보기>-----  
 F = {A→B, C→D, D→AE}

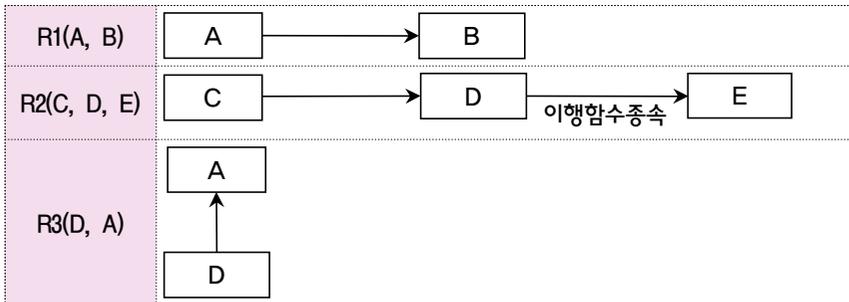
- ① R에서 모든 속성은 C에 함수적으로 종속될 수 있다.
- ② R1(A, B), R2(C, D, E), R3(D, A)로 분해할 경우, R1, R2, R3는 제3정규형을 만족한다.
- ③ R1(D, A, B), R2(C, D, E)로 분해할 경우, R1, R2는 제2정규형을 만족한다.
- ④ R1(C, D), R2(D, A, E), R3(A, B)로 분해할 경우, R1, R2, R3는 BCNF를 만족한다.

☞ 릴레이션 분해

// 함수종속 관계를 **함수종속 다이어그램**으로 그리면 다음과 같다.



↓  
 ② R1(A, B), R2(C, D, E), R3(D, A)로 분해할 경우  
 ↓



• 분해 결과 : R2(C, D, E)에 **이행함수종속**이 존재하므로 제3정규형을 만족하지 않는다.

- ④ R1(C, D), R2(D, A, E), R3(A, B)로 분해할 경우, R1, R2, R3는 BCNF를 만족한다.
  - R1(C, D)의 결정자는 C이다. (C가 후보키)
  - R2(D, A, E)의 결정자는 D이다. (D가 후보키)
  - R3(A, B)의 결정자는 A이다. (A가 후보키)

10. 스키마가 사원(사원번호, 직급, 보너스)인 사원 테이블의 인스턴스가 <보기 1>과 같을 때, <보기 2>의 ㉠, ㉡에 들어갈 말로 옳게 짝 지은 것은? [2020년 국가 7급]

---<보기 1>---

사원

사원번호	직급	보너스
1000	과장	500
1001	과장	500
1002	사원	200

---<보기 2>---

이 사원 테이블에서 보너스는 직급별로 결정된다. 이때 직급과 보너스 속성은 모두 기본키인 사원번호에 의해 결정되지만 사원번호가 직급을 결정하고 직급이 보너스를 결정하는 ( ㉠ ) 관계를 가지고 있으므로, 이 테이블은 ( ㉡ )이라고 할 수 있다.

㉠

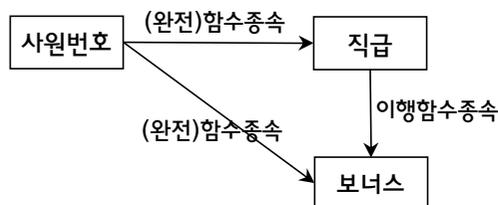
- ① 완전함수종속
- ② 완전함수종속
- ③ 이행함수종속
- ④ 이행함수종속

㉡

- 제2정규형
- 제3정규형
- 제2정규형
- 제3정규형

☞ 정규형

// 함수종속(FD)의 관계를 그림으로 그리면 다음과 같다.



- 함수종속 FD : 사원번호→직급, 직급→보너스
- 이행함수종속이 존재하므로 제3정규형이 아니고, 제2정규형에 속한다.

// 제3정규형(3NF; third normal form)

어떤 릴레이션이 2NF이고,  
기본키에 속하지 않은 모든 속성이 기본키에 이행함수종속이 아니면 제3정규형에 속한다.

11. <보기>와 같이 릴레이션 R과 함수종속성 F가 정의되어 있다. 제3정규형(3NF)을 만족하도록 분해한 릴레이션들을 나타낸 것으로 가장 옳은 것은? [2018년 서울 7급]

----<보기>-----

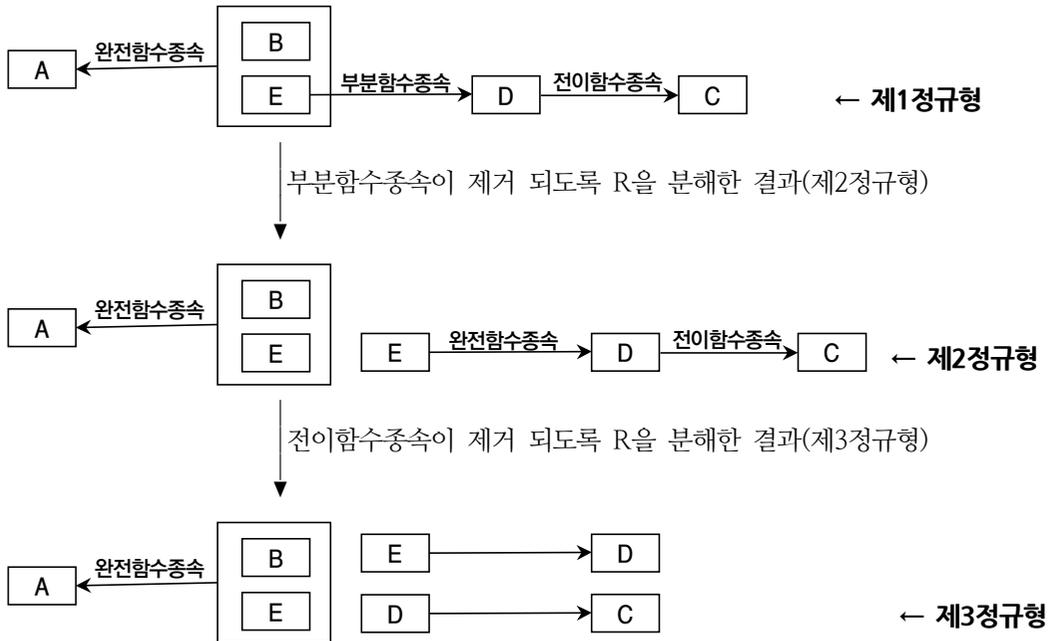
R(A, B, C, D, E)

F = {BE→A, E→D, D→C}

- ① (D, C), (E, D), (B, E, A)
- ② (D, C), (B, E, A, D)
- ③ (D, C), (B, E, A), (A, E)
- ④ (E, D, C), (B, E, A)

☞ 제3정규형(3NF)

// 함수종속 관계를 그림으로 나타내면



// 제3정규형(3NF)을 만족하도록 분해한 릴레이션들

- (D, C)
- (E, D)
- (B, E, A)

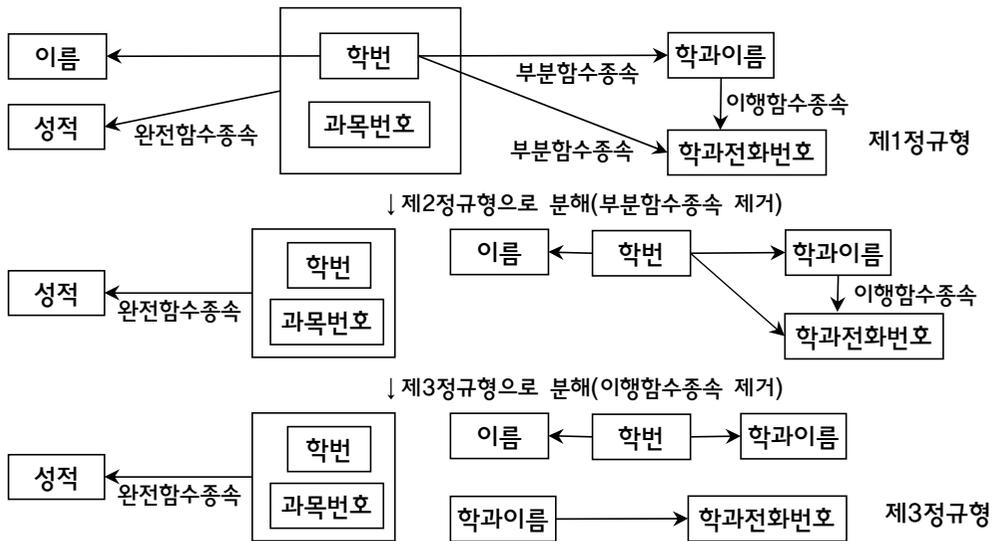
12. 다음 함수종속 관계가 있는 릴레이션 '학생(학번, 과목번호, 이름, 학과이름, 학과전화번호, 성적)'에 대한 설명으로 옳지 않은 것은? (단, 학생 릴레이션은 이미 제1정규형을 만족한다) [2022년 국가 7급]

- 학번 → {이름, 학과이름, 학과전화번호}
- 학과이름 → 학과전화번호
- {학번, 과목번호} → 성적

- ① 함수적 종속관계  $X \rightarrow Y$ 가 성립하더라도,  $Y \rightarrow X$ 가 항상 성립하는 것은 아니다.
- ② 학생 릴레이션을 분해하더라도 무손실분해(lossless decomposition)이어야 한다.
- ③ 학생 릴레이션이 제2정규형을 만족하도록 분해하면, (학번, 이름, 학과이름, 학과전화번호), (학번, 과목번호, 성적)이다.
- ④ 학생 릴레이션이 제3정규형을 만족하도록 분해하면, (학번, 이름), (학과이름, 학과전화번호), (학번, 과목번호, 성적)이다.

☞ 함수종속

릴레이션	학생(학번, 과목번호, 이름, 학과이름, 학과전화번호, 성적)
함수종속	학번 → {이름, 학과이름, 학과전화번호} 학과이름 → 학과전화번호 {학번, 과목번호} → 성적



• 제3정규형 : (학번, 과목번호, 성적), (학번, 이름, 학과이름), (학과이름, 학과전화번호)

13. <보기>의 릴레이션 R에 존재하는 함수종속성을 나타낸 것으로 가장 옳은 것은? [2018년 서울 7급]

----<보기>-----

R				
A	B	C	D	E
1	b	10	2	a
2	a	10	2	a
1	b	10	3	a
1	b	20	3	b

- ①  $A \rightarrow BC$                       ②  $C \rightarrow DE$   
 ③  $DE \rightarrow C$                       ④  $BC \rightarrow D$

☞ 함수종속성 - 오직 하나만 연관

//  $A \rightarrow B$ 가 함수종속성인 이유

- $1 \rightarrow b$
- $2 \rightarrow a$

//  $DE \rightarrow C$ 가 함수종속성인 이유

- $\{2, a\} \rightarrow 10$
- $\{3, a\} \rightarrow 10$
- $\{3, b\} \rightarrow 20$

//  $A \rightarrow BC$ 가 함수종속성이 아닌 이유

- $1 \rightarrow \{b, 10\}$  또는  $\{b, 20\}$

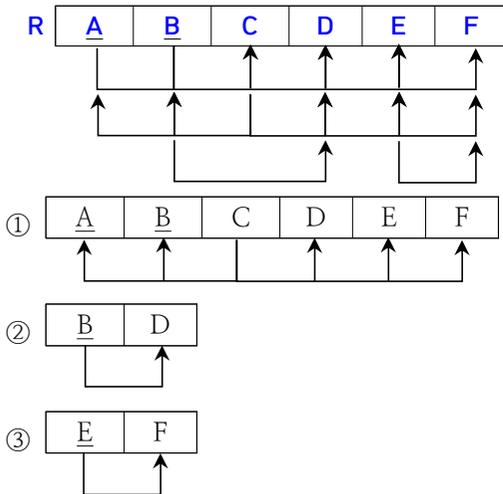
↳ 즉, A가 1일 때, BC가 가지는 것은 오직 하나가 아니다. 2개이다.

// 함수종속성(FD, Functional Dependency)

- 어떤 릴레이션 R에서 X, Y가 각각 R의 속성 집합에 대해 부분집합일 때, 속성 X의 값 각각에 대해 속성 Y의 값이 **오직 하나만 연관**되어 있을 때 Y는 X에 함수종속이라 한다.
- 속성 Y가 속성 X에 함수종속이면,  $X \rightarrow Y$ 처럼 표시한다.
- 여기서, X를 결정자(determinent), Y를 종속자(dependent)라 한다.
- 만일, R에서 X가 후보키(기본키)이면 R의 모든 Y는 반드시 X에 함수종속이어야 한다.

정답 : ③

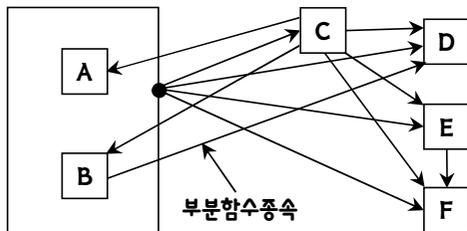
14. 다음 그림은 릴레이션 R과 그 함수적 종속성을 표현하고 있다. 속성 C는 릴레이션 R의 후보 키이며, 이 릴레이션은 이미 제1정규화를 수행하였다. 이 릴레이션을 후보키까지 고려하여 제2 정규화하였을 때, 분해된 릴레이션 중 기본키가 A, B가 아닌 릴레이션은? (단, 밑줄은 기본키를 의미한다) [2017년 국가 7급]



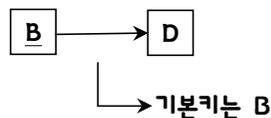
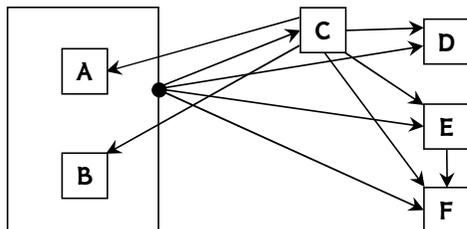
④ 제2정규화하더라도 변화 없음

☞ 정규화

• 제2정규화는 부분함수종속을 제거한 것이다.



↓ 부분함수종속 제거



• 분해된 릴레이션 B→D에서 기본키는 B이다.(A, B가 아님)

15. 다음은 릴레이션 R의 스키마와 이에 해당되는 함수적 종속성을 나타낸 것이다. 이상 현상을 없애기 위하여 릴레이션을 다음과 같이 분해하였을 경우 옳지 않은 것은? [2009년 국가 7급]

- 릴레이션 스키마 : R(A, B, C, D, E, F)
- 함수적 종속성 :  $A \rightarrow B, A \rightarrow C, A \rightarrow E, A \rightarrow F, C \rightarrow D, C \rightarrow E, C \rightarrow B$

- ① R1(A, B, C, F), R2(C, E, D)
- ② R1(A, C, E, F), R2(C, B, D)
- ③ R1(A, C, F), R2(B, C, D, E)
- ④ R1(A, B, E, F), R2(B, D, E)

☞ 이상 현상을 제거하기 위한 릴레이션 분해

④ R1(A, B, E, F), R2(B, D, E)

- 먼저, 항목 ④에서는 속성 C가 존재하지 않는다.
- 그리고, R2(B, D, E)에서 B, D, E는 모두 종속자로 결정자가 되지 못한다.
- 이상현상을 없애기 이전에, 릴레이션 분해가 잘못되었다.

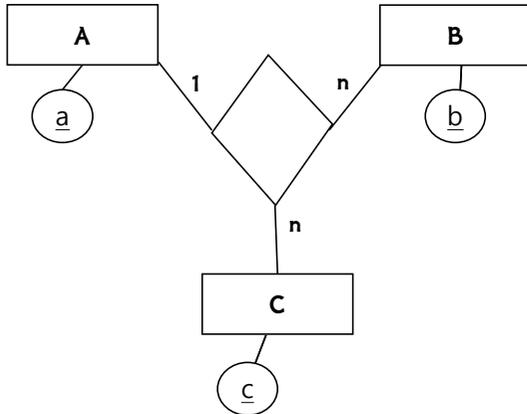
② R1(A, C, E, F), R2(C, B, D)에 대한 함수종속 다이어그램



- R1에는 이행함수종속이 존재한다.
- R1에는 이행함수종속이 존재하므로 2NF이고, 이상현상이 발생된다.
- 이상현상이 발생되지 않도록 하려면, R1은 R1\_1(A, C), R1\_2(C, E)로 분해되어야 한다.

- 주어진 문제에서 정답은 ④번으로 발표되었다.
- 그런데, ②번도 이상현상이 발생하므로 답이 될 수 있다.
- 문제가 릴레이션 분해가 잘못된 것을 고르는 것이면 정답은 ④번이다.
- 그런데, 문제는 이상현상을 없애기 위하여 릴레이션을 분해하는 것으로 되어 있다.

16. 다음의 ERD(entity-relationship diagram)에서 성립하는 함수종속성(functional dependency)으로 옳은 것은? (단, 개체집합(entity set) A, B, C의 기본키는 각각 a, b, c이다) [2014년 국가 7급]



- ①  $a \rightarrow b, a \rightarrow c$       ②  $b \rightarrow a, c \rightarrow a$   
 ③  $a \rightarrow bc$               ④  $bc \rightarrow a$

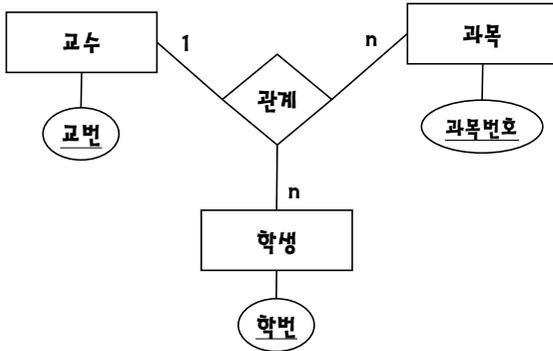
☞ 3진 관계 ERD에서 성립하는 함수종속성

- 릴레이션 A와 B는 1:n의 관계이다.
- 해서, 속성 a와 연결되는 속성 b의 값은 다수가 될 수 있다.  $a \rightarrow b$ 는 불가능하다.
- 릴레이션 A와 C는 1:n의 관계이다.
- 해서, 속성 a와 연결되는 속성 c의 값은 다수가 될 수 있다.  $a \rightarrow c$ 는 불가능하다.
- 릴레이션 B, C는 A와 n:1의 관계이다.
- 해서, bc에 연결되는 a의 값은 오직 하나밖에 없다.  $bc \rightarrow a$ 는 성립 가능하다.(함수종속성)

// 3진 관계

- 3진 관계에서는 관계타입에 대한 새로운 릴레이션을 생성한다.
- 관계타입에 대한 릴레이션에는 3개의 개체타입의 기본키를 추가한다.  
 → 기본키는 n측의 기본키로 구성된다. 복합키가 된다.(b, c)
- 스키마(릴레이션) 구성  
 A(a)  
 B(b)  
 C(c)  
 R(a, b, c) → 릴레이션 R의 기본키는 {b, c}이다.
- 기본키가 {b, c}이므로,  $bc \rightarrow a$ 는 함수종속성이 된다.

// 3진 관계 예제



밑줄은 기본키

↓ 스키마(릴레이션) 구성

- 교수(교번)
- 학생(학번)
- 과목(과목번호)
- 관계(학번, 과목번호, 교번) → 기본키는 {학번, 과목번호}이 된다.

관계

학번	과목번호	교번
1	DB_1	P1_홍재연
1	SE_1	P2_김유신
1	DB_2	P1_홍재연
2	DB_1	P1_홍재연
2	SE_2	P2_김유신
3	보안_1	P3_이삼오

- 학번 1인 학생은 교번 P1\_홍재연이 개설한 DB\_1을 수강한다.
- 학번 2인 학생은 교번 P2\_김유신이 개설한 SE\_2를 수강한다.
- {학번, 과목번호} → 교번
- 교번은 기본키 {학번, 과목번호}에 함수종속성을 가진다.
- 함수종속성 어떤 속성에 대해 다른 속성의 속성값이 오직 하나만 연관되는 것이므로

// 함수종속성(FD, Functional Dependency)

- 어떤 릴레이션 R에서 X, Y가 각각 R의 속성 집합에 대해 부분집합일 때, 속성 X의 값 각각에 대해 속성 Y의 값이 오직 하나만 연관되어 있을 때 Y는 X에 함수종속이라 한다.
- 속성 Y가 속성 X에 함수종속이면, X→Y처럼 표시한다.  
→ 여기서, X를 결정자(determinent), Y를 종속자(dependent)라 한다.
- 만일, R에서 X가 후보키(기본키)이면 R의 모든 Y는 반드시 X에 함수종속이어야 한다.

17. 릴레이션  $R(A, B, C, D, E)$ 에 대해 <보기>와 같이 함수 종속성(functional dependency)이 있을 때, 릴레이션  $R$ 을 릴레이션  $R_1$ 과  $R_2$ 로 나눈 것 중 무손실 분해(lossless decomposition) 성질을 만족하는 것은? [2021년 서울 7급]

-----  
보기 :  $A \rightarrow CD$   
-----

- ①  $R_1(A, C, D), R_2(A, B, E)$
- ②  $R_1(A, C, D), R_2(B, C, E)$
- ③  $R_1(A, D, E), R_2(A, B, C)$
- ④  $R_1(A, D, E), R_2(B, C, D)$

♣ 무손실 분해

// 무손실 조인(lossless-join) 분해

- 무손실 조인 분해는 분해된 릴레이션을 조인했을 때,
- 원래 분해 전의 릴레이션에 대해 어떠한 데이터 손실도 없는 분해를 의미한다.
- 릴레이션을 분해한 후, 다시 조인할 때 데이터 손실이 발생하면 분해는 잘못된 것이다.
- 다시 조인할 때, 원하지 않은 정보가 추가되어도 분해는 잘못된 것이다.

// 함수종속 보존 분해

- 릴레이션  $R$ 을  $R_1, R_2, \dots, R_n$ 으로 분해할 때, 함수종속 집합  $F$ 도 각 릴레이션에 속한 속성만을 포함하는 함수종속 집합  $F_1, F_2, \dots, F_n$ 으로 분해되는 것이다.
- 분해되었을 때,  $(F_1 \cup F_2 \cup \dots \cup F_n)^+ = F^+$ 이면 함수종속이 보존되는 분해가 된다.
- $F^+$ 는  $F$ 로부터 추론할 수 있는 가능한 모든 함수종속들의 집합이다.

[예] 릴레이션  $R(A, B, C)$ 의 함수종속집합  $F=(A \rightarrow B, B \rightarrow C)$ 인 경우,

↓ 무손실 조인 분해

- 릴레이션  $R(A, B, C)$ 을  $R_1(A, B)$ 과  $R_2(B, C)$ 로 분해했을 때
- $R_1 \cap R_2$ 의 교집합 연산 결과가  $R_1$  또는  $R_2$ 의 기본키이면 무손실 분해가 된다.
- $R_1 \cap R_2 = \{A, B\} \cap \{B, C\} = \{B\}$ 이다.  $B$ 는  $R_2$ 의 기본키이므로 무손실 분해가 된다.

①  $R_1(A, C, D), R_2(A, B, E)$  → 무손실 조인 분해

↓  
함수종속 보존 분해( $A \rightarrow CD$ )



19. 테이블은 주문 거래의 일부를 보여준다. 데이터의 의미가 그 아래와 같다고 가정할 때, 제3정규형으로 정규화한 결과로 가장 옳은 것은? (단, 기본키는 밑줄을 그어 표시하였다) [2021년 군무원 7급]

고객ID	고객이름	주문날짜	주문ID	상품ID	수량	단가
100	최인기	2021-3-2	N231	P002	2	500
100	최인기	2021-4-12	N302	P002	3	500
110	정수희	2021-4-23	N323	P002	1	500
120	김하나	2021-4-23	N325	P003	3	250
120	김하나	2021-4-23	N326	P001	2	300
100	최인기	2021-5-2	N399	P004	4	250

- 고객별 고유한 고객ID가 부여된다.
- 각 주문별 고유한 주문ID가 주어진다.
- 고객은 하루에 여러 번 주문할 수 있으나, 한 고객은 동일한 상품을 하루에 최대한 한 번만 주문할 수 있다.
- 각 상품ID별 단가는 고정되어 있다.

- 
- ① R1(주문ID, 고객ID, 상품ID, 주문날짜, 수량)  
R2(주문ID, 고객이름)  
R3(상품ID, 단가)
  - ② R1(주문ID, 고객ID, 상품ID, 주문날짜, 수량)  
R2(고객ID, 고객이름)  
R3(상품ID, 단가)
  - ③ R1(주문ID, 고객ID, 주문날짜, 수량)  
R2(고객ID, 고객이름)  
R3(주문ID, 상품ID, 단가)
  - ④ R1(고객ID, 상품ID, 주문날짜, 수량)  
R2(고객ID, 고객이름)  
R3(주문ID, 상품ID, 단가)

♣ 제3정규형으로 정규화한 결과

어떤 릴레이션이 2NF이고,  
기본키에 속하지 않은 모든 속성이 기본키에 이행함수종속이 아니면 제3정규형에 속한다.

- 이행함수종속 : 함수종속 관계에서 'A→B와 B→C가 성립하면 A→C가 성립한다.'

고객ID	고객이름	주문날짜	주문ID	상품ID	수량	단가
------	------	------	------	------	----	----

↓ 함수종속 관계를 분석하면

- **고객별 고유한 고객ID가 부여된다.**

고객ID → 고객이름

- **각 주문별 고유한 주문ID가 주어진다.**

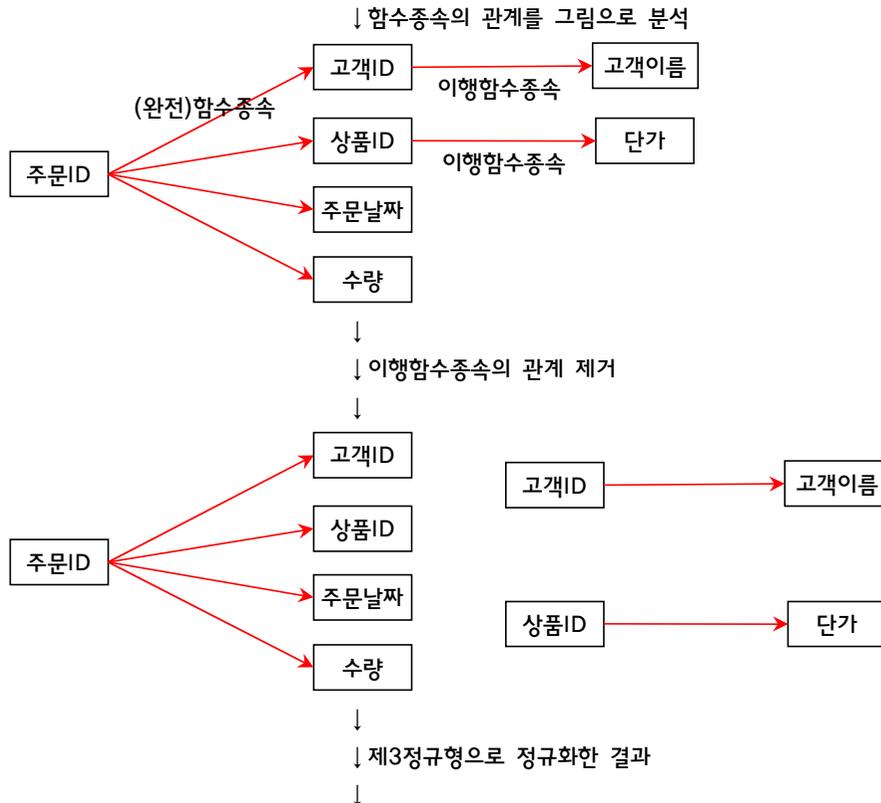
주문ID → 주문별(고객ID, 상품ID, 주문날짜, 수량)

- **고객은 하루에 여러 번 주문할 수 있으나, 한 고객은 동일한 상품을 하루에 최대한 한 번만 주문할 수 있다.**

주문ID → 고객ID, 상품ID, 주문날짜, 수량

- **각 상품ID별 단가는 고정되어 있다.**

상품ID → 단가



- ② R1(주문ID, 고객ID, 상품ID, 주문날짜, 수량)
- R2(고객ID, 고객이름)
- R3(상품ID, 단가)