

데이터베이스

강의 노트

제 13 회차
릴레이션 정규화

❖ 학습목표

- 갱신 이상이 무엇인지 설명할 수 있다.
- 함수적 종속이 무엇인지 설명할 수 있다.
- 정규화가 무엇인지 설명할 수 있다.
- 정규화의 단계를 설명할 수 있다.
- 역정규화가 필요한 이유를 설명할 수 있다.

❖ 학습내용

- 갱신 이상과 함수적 종속
- 단계별 정규화와 역정규화

갱신 이상과 함수적 종속

1. 갱신 이상
2. 함수적 종속
3. 완전 함수적 종속과 부분
함수적 종속
4. 이행적 함수적 종속

1. 갱신 이상

1) 갱신 이상(Update Anomaly)

갱신 이상(Update Anomaly)이란?

관계 DB를 조작할 때 데이터의 중복으로 인해 발생하는 문제점을 의미

갱신 이상의 종류

삽입 이상 (Insertion Anomaly)

- 불필요한 데이터를 함께 삽입하지 않으면 어떤 데이터를 삽입하는 것이 불가능한 문제점
- 즉, 데이터를 삽입할 때 원하지 않는 불필요한 데이터를 같이 삽입해야 하는 것

수정 이상 (Modification Anomaly)

- 중복된 데이터 가운데 일부만 수정되어 데이터의 불일치가 발생하는 문제점

삭제 이상 (Deletion Anomaly)

- 어떤 데이터를 삭제하면 유용한 데이터도 함께 삭제되는 문제점

1. 갱신 이상

2) 갱신 이상 사례

사례 1: 수강 릴레이션

학생이 수강한 교과목 정보와 성적 및 학년 정보 저장

기본 키 : {학번, 교과목번호}


수강

학번	교과목번호	성적	학년
100	C413	A	4
100	E412	A	4
200	C123	B	3
300	C312	A	1
300	C324	C	1
300	C413	A	1
400	C312	A	4
400	C324	A	4
400	C413	B	4
400	E412	C	4
500	C312	B	2


수강 릴레이션에서 여러 가지 갱신 이상이 발생하는 이유는?

같은 학번에 대해 학년 정보가 중복되어 있기 때문


삽입 이상 (Insertion Anomaly)

- 600번 학생이 2학년이라는 사실을 삽입하려는데, 교과목번호에 대한 정보가 없어서 불가능함
 기본 키를 구성하는 '교과목번호'에 임의의 값을 지정해서 삽입해야 함
- 원하지 않는(불필요한) 정보를 강제로 삽입해야 함

수정 이상 (Modification Anomaly)

- C312 교과목을 수강한 400번 학생의 학년을 4에서 3으로 변경하면, 나머지 3개 데이터의 학년 정보와 불일치가 발생함
 학번이 400번인 데이터를 모두 찾아서 전부 변경해야 함
- 중복된 데이터의 일부 갱신으로 인해 정보의 모순이 발생함

삭제 이상 (Deletion Anomaly)

- 200번 학생이 'C123'의 수강을 취소해서 삭제시키면, 200번 학생이 3학년이라는 정보도 함께 삭제됨
 유용한 정보가 사라짐
- 연쇄 삭제(Triggered deletion)로 인해 유용한 정보가 손실됨

1. 갱신 이상

2) 갱신 이상 사례

사례 2: 사원정보 릴레이션

사원과 부서 정보를 함께 저장함 (기본 키는 사원번호와 부서번호의 조합)

단, 한 사원이 여러 부서에 근무할 수 있는 상황임

사원이름	사원번호	주소	전화번호	부서번호	부서이름
김길동	1001	서울시 강남구	02-111-2222	1	영업
박영숙	1002	서울시 동작구	02-222-3333	2	개발
박영숙	1002	서울시 동작구	02-333-4444	3	기획
이창민	1003	천안시 성정동	041-444-5555	2	개발
최수민	1004	서울시 강북구	02-555-6666	1	영업

정보 중복

정보 중복

- 각 사원이 속한 부서 수만큼 동일한 사원의 정보가 존재해야 하므로 사원이름, 사원번호, 주소, 전화번호 등이 중복 저장되어 저장 공간이 낭비됨

수정 이상 (Modification Anomaly)

- 만일 어떤 부서의 이름이 바뀔 때 이 부서에 근무하는 일부 사원 정보의 부서이름만 변경하면 데이터의 불일치가 발생함

삽입 이상 (Insertion Anomaly)

- 만일 어떤 부서를 신설했는데 아직 사원이 한 명도 배정되지 않았다면, 이 부서에 관한 정보를 입력할 수 없음

삭제 이상 (Deletion Anomaly)

- 만일 단 한 명만 근무하는 부서가 있는 경우, 그 사원 정보를 삭제하면 그 사원이 속한 부서에 관한 정보도 함께 삭제됨

1. 갱신 이상

3) 갱신 이상의 원인과 해결 방법

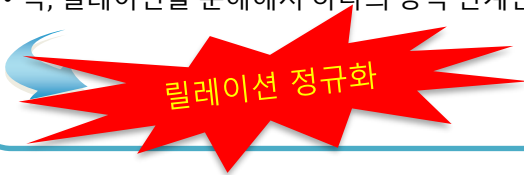
갱신 이상의 원인

- 릴레이션 스키마를 생성할 때 **속성들 간의 종속성**을 충분히 고려하지 않아서 불필요한 **데이터가 중복**되었기 때문임
- 즉, 한 개체의 속성들 간에 존재하는 여러 개의 종속 관계를 하나의 릴레이션으로 표현했기 때문임



갱신 이상의 해결 방법

- 릴레이션 스키마를 변환해서 속성들 간의 여러 종속 관계를 분해(Decomposition)함
- 즉, 릴레이션을 분해해서 하나의 종속 관계는 하나의 릴레이션으로 표현함



2. 함수적 종속

1) 함수적 종속 이해

함수적 종속 (Functional Dependency: FD) 이란?

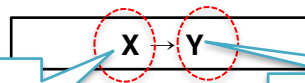
- 속성들 간의 대응 관계를 보다 수학적으로 규명하기 위해 E. F. Codd가 제안한 이론
- 정규화의 핵심적인 개념
- 릴레이션이 갖고 있는 갱신 이상을 발견하기 위한 하나의 수단
- 속성들 간의 관계(Relationship)에 대한 제약 조건에 해당함

함수적 종속의 정의 및 표현 방법

속성 X의 값이 속성 Y의 값을 결정지으면,
즉 속성 X의 각각의 값에 대해 속성 Y의 값이 오직 하나만 연관되면



속성 Y는 속성 X에 함수적으로 종속된다



X - 결정자(Determinant)

: 주어진 릴레이션에서 다른 속성(또는 속성의 집합)을 고유하게 결정하는 하나 이상의 속성들

Y - 종속자(Dependent)

: 결정자에 의해서 값이 결정되는 하나 이상의 속성들

2. 함수적 종속

2) 예제

예제 1 : 학생 릴레이션의 함수적 종속 관계 분석

학생 (학번, 이름, 학년, 학과)

학번	이름	학년	학과
20113910	김철수	2	컴퓨터
20123929	이영희	1	전기
20103932	민동규	3	전자
20113934	장영주	2	컴퓨터
20123900	김철수	1	정통

- ① 이름이 같은 학생이 여러 명 있을 수 있으므로, 이름을 알아도 학번이나 학년, 학과를 알 수 없음
 - ▶ 이름이 학번이나 학년, 학과를 함수적으로 결정하지 않음
- ② 모든 학생의 학번이 서로 다르므로, 학번을 알면 이름과 학년, 학과를 알 수 있음
 - ▶ 학번을 알면 이름과 학년, 학과를 함수적으로 결정함



학생 릴레이션의 함수적 종속 관계 표현

학번 → 이름
 학번 → 학년
 학번 → 학과

또는

학번 → (이름, 학년, 학과)

또는

학번 → 이름 | 학년 | 학과

릴레이션 R에서 속성 X가 기본 키이면, R의 모든 속성 Y에 대해 $X \rightarrow Y$ 가 성립

2. 함수적 종속

2) 예제

예제 2 : 사원 릴레이션의 함수적 종속 관계 분석

사원 (사원번호, 부서번호, 부서이름, 이름, 직책, 주소, 연락처)

사원번호	부서번호	부서이름	이름	직책	주소	연락처
3214	100	기획팀	김철수	팀장	서울	010-1111-2222
2456	200	개발팀	이영호	프로그래머	천안	010-2222-3333
2456	300	연구팀	이영호	설계자	천안	010-3333-4444
4602	100	기획팀	박민희	사원	서울	010-4444-5555
3722	300	연구팀	김철수	팀장	천안	010-5555-6666

* 단, 한 사원이 2개 이상의 부서에 소속되어 근무할 수 있고, 각 부서에서의 직책이 다를 수 있음



사원 릴레이션의 함수적 종속 관계 표현

종속 관계 분석	함수적 종속 표현
① 사원번호만 알면 사원이름과 주소, 전화번호를 알 수 있음 사원번호가 사원이름과 주소, 전화번호를 함수적으로 결정	사원번호 → (사원이름, 주소, 전화번호)
② 부서번호를 알면 부서이름을 알 수 있음 부서번호가 부서이름을 함수적으로 결정	부서번호 → 부서이름
③ 부서이름을 알면 부서번호를 알 수 있음 (사람 이름과는 달리 일반적으로 동일한 이름을 갖는 부서가 없으므로) 부서이름이 부서번호를 함수적으로 결정	부서이름 → 부서번호
④ 사원번호와 부서번호를 모두 알아야 직책을 알 수 있음 {사원번호, 부서번호}가 직책을 함수적으로 결정 {사원번호, 부서번호}가 기본 키이므로 나머지 모든 속성의 결정자 역할을 하지만, 여기에서는 완전 함수 종속 관계만을 표현한 것임	{사원번호, 부서번호} → 직책
⑤ 사원이름을 알아도 사원번호나 부서번호, 주소 등을 알 수 없음 사원이름은 나머지 속성들을 함수적으로 결정하지 X	

2. 함수적 종속

3) 함수적 종속 다이어그램

함수적 종속 다이어그램(Functional Dependency Diagram : FDD)이란?

- 함수적 종속 관계를 그림으로 표현하는 방법
- 결정자와 종속자를 모두 직사각형으로 표시하고, 화살표로 연결해서 표시함

2. 함수적 종속

4) 예제

예제 1 : 교과목 릴레이션의 FDD 작성

교과목(교과목번호, 교과목명, 학점, 학과, 담당교수)

교과목번호	교과목명	학점	학과	담당교수
C123	컴퓨터개론	3	컴퓨터	1102
C234	프로그래밍	3	전자	1200
C257	자료구조	3	컴퓨터	1322
C156	프로그래밍	2	전기	1565
C198	데이터베이스	3	컴퓨터	1422

함수적 종속 관계 분석

결정자 역할을 하는 속성은 기본 키인 교과목번호 밖에 없음

- 교과목번호가 나머지 모든 속성들을 함수적으로 결정함

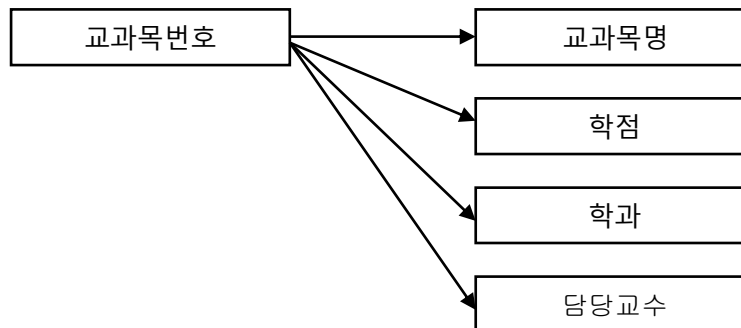


함수적 종속 표현 : 교과목번호 → (교과목명, 학점, 학과, 담당교수)

교과목명을 알아도 교과목번호나 학점, 학과, 담당교수를 알 수 없으므로
교과목명 어떤 속성도 함수적으로 결정하지 않음

- 동일한 교과목명을 갖는 강의가 서로 다른 학과에서 개설되어,
다른 교과목번호와 학점 등으로 표현될 수 있기 때문

FDD 작성



2. 함수적 종속

4) 예제

예제 2 : 사원 릴레이션의 FDD 작성

사원(사원번호, 부서번호, 부서이름, 이름, 직책, 주소, 연락처)

사원번호	부서번호	부서이름	사원이름	직책	주소	전화번호
3214	100	기획팀	김철수	팀장	서울	010-1111-2222
2456	200	개발팀	이영호	프로그래머	천안	010-2222-3333
2456	300	연구팀	이영호	설계자	천안	010-3333-4444
4602	100	기획팀	박민희	사원	서울	010-4444-5555
3722	300	연구팀	김철수	팀장	천안	010-5555-6666

단, 한 사원이 2개 이상의 부서에 소속되어 근무할 수 있고, 각 부서에서의 직책이 다를 수 있다.

함수적 종속 관계 분석

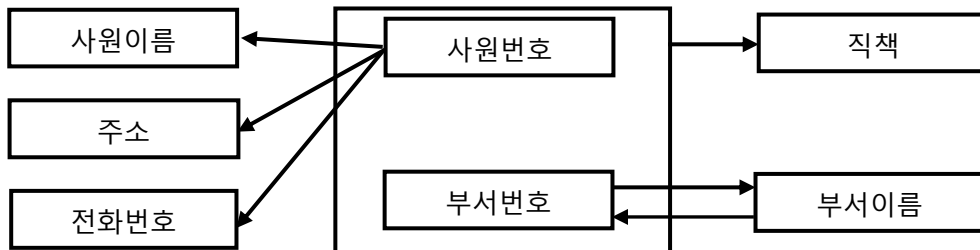
사원번호 → (사원이름, 주소, 전화번호)

부서번호 → 부서이름

부서이름 → 부서번호

{사원번호, 부서번호} → 직책

FDD 작성



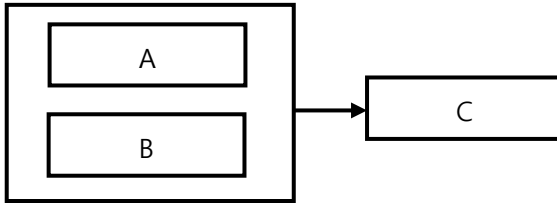
3. 완전 함수적 종속과 부분 함수적 종속

1) 완전 함수적 종속

완전 함수적 종속(Full Functional Dependency)이란?

- 복합 속성 X 에 대해서 $X \rightarrow Y$ 가 성립할 때, $X' \subset X$ 이고 $X' \rightarrow Y$ 를 만족하는 속성 X' 이 존재하지 않으면, 속성 Y 는 복합 속성 X 에 완전 함수적 종속이라 칭함

즉, 결정자인 복합 속성에 포함되는 속성 가운데 **단독으로 결정자 역할을 하는 속성이 없으면**, 종속자가 결정자에 완전 함수적 종속이라고 칭함



[완전 함수적 종속 다이어그램]

종속자를 완전 또는 부분 함수적 종속으로 구분 할 때는 어떤 경우인가요?

결정자가 복합 속성인 경우

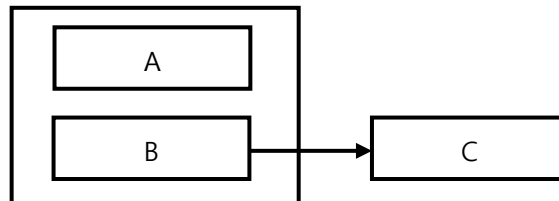
복합 속성인 $\{A, B\}$ 가 C 의 결정자이며, A 나 B 만으로는 C 를 결정할 수 없음

2) 부분 함수적 종속

부분 함수적 종속(Partial Functional Dependency)이란?

- 복합 속성 X 에 대하여 $X \rightarrow Y$ 가 성립할 때, $X' \subset X$ 이고 $X' \rightarrow Y$ 를 만족하는 속성 X' 이 존재하면, 속성 Y 는 복합 속성 X 에 부분 함수적 종속이라 칭함

즉, 결정자인 복합 속성에 포함되는 속성 가운데 **단독으로 결정자 역할을 하는 속성이 있으면**, 종속자가 결정자에 부분 함수적 종속이라고 칭함



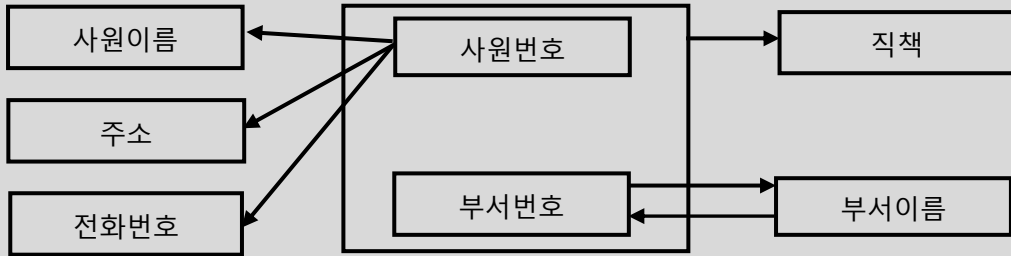
[부분 함수적 종속 다이어그램]

복합 속성인 $\{A, B\}$ 가 C 의 결정자이지만, B 만으로도 C 를 결정할 수 있음

3. 완전 함수적 종속과 부분 함수적 종속

3) 예제

예제 1 : 사원 릴레이션의 완전 및 부분 함수적 종속 관계 분석



복합속성 $X = \{\text{사원번호}, \text{부서번호}\}$ 에 대해
 $\{\text{사원번호}, \text{부서번호}\} \rightarrow (\text{직책}, \text{사원이름}, \text{주소}, \text{전화번호}, \text{부서이름})$ 가 성립



완전 및 부분 함수적 종속 관계 분석

- ① **직책**의 결정자는 복합속성 X 이고, 사원번호나 부서번호만으로는 직책을 결정하지 못하므로 직책은 결정자 X 에 **완전 함수적 종속**
- ② **사원이름, 주소, 전화번호**의 결정자는 사원번호이고, 사원번호는 복합속성 X 에 포함되므로 사원이름, 주소, 전화번호는 결정자 X 에 **부분 함수적 종속**
- ③ **부서이름**의 결정자는 부서번호이고, 부서번호는 복합속성 X 에 포함되므로 부서이름은 결정자 X 에 **부분 함수적 종속**

3. 완전 함수적 종속과 부분 함수적 종속

3) 예제

예제 2 : 수강 릴레이션의 완전 및 부분 함수적 종속 관계 분석

수강(학번, 교과목번호, 성적, 학년)

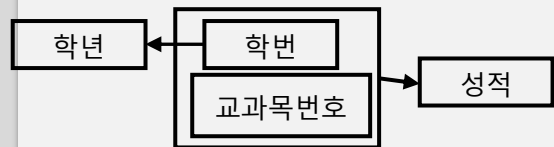
학번	교과목번호	성적	학년
100	C413	A	4
100	E412	A	4
200	C123	B	3
300	C312	A	1
300	C324	C	1
300	C413	A	1
400	C312	A	4
400	C324	A	4
400	C413	B	4
400	E412	C	4
500	C312	B	2

FD

(학번, 교과목번호) → 성적

학번 → 학년

FDD



- ① 성적 : {학번, 교과목번호}에 완전 함수적 종속
- ② 학년 : {학번, 교과목번호}에 부분 함수적 종속

4. 이행적 함수적 종속

1) 이행적 함수적 종속

이행적 함수적 종속(Transitive Functional Dependency)이란?

한 릴레이션의 속성 A, B, C가 있을 때 다음과 같은 필요 충분 조건을 만족하는 경우,

$$A \rightarrow B \wedge B \rightarrow C$$

속성 C가 속성 A에 이행적(또는 추이적) 함수적 종속이라고 함

2) 예제

예제

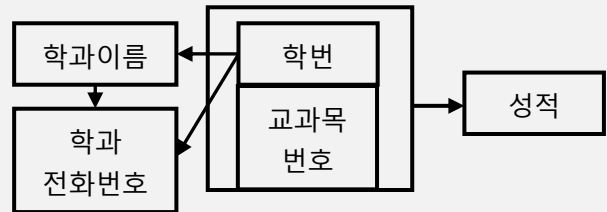
학생 릴레이션의 이행적 함수적 종속 관계 분석

학생(학번, 교과목번호, 성적, 학과이름, 학과전화번호)

FD

{학번, 교과목번호} → 성적
학번 → (학과이름, 학과전화번호)
학과이름 → 학과전화번호

FDD



“학번 → 학과이름, 학과이름 → 학과전화번호”라는
2개의 필요 충분 조건을 만족하므로 학과전화번호는 학번에 이행적 함수적 종속 관계

단계별 정규화와 역정규화

1. 정규화 개념
2. 정규형 구분
3. 단계별 정규화
4. 역정규화

1. 정규화 개념

1) 정규화 개념

정규화(Normalization)란?

- 스키마 변환을 통해서 일련의 제약 조건을 만족하는 릴레이션을 만드는 과정
- 속성들 간의 종속성을 분석해서 하나의 릴레이션에는 하나의 종속성만 갖도록 릴레이션을 분해하는 과정
- 갱신 이상의 발생 요인인 데이터 중복을 최소화하기 위해서 일종의 제약조건을 기초로 릴레이션을 분해하는 과정
- 불필요한 데이터 중복을 제거해서 논리적 데이터 모델을 단순화시키는 과정

2) 정규화의 원칙

1

데이터의 중복을 최소화함

2

정보가 사라지지 않아야 함 (정보의 무손실)

- 같은 의미의 정보를 유지하면서 더 바람직한 구조로 변환해야 함

3

다음과 같은 원칙에 따라 분해함

- 독립적인 관계는 별개의 릴레이션으로 표현함
- 릴레이션 각각에 대해 독립적인 조작이 가능해야 함

1. 정규화 개념

3) 정규화되지 않은 릴레이션의 문제점

- 1 데이터 중복으로 인해 갱신 이상 현상이 발생함
- 2 특정 정보를 표현하는 것이 불가능할 수 있음
- 3 널(NULL) 값을 많이 포함하게 됨
- 4 저장된 정보가 부정확할 수 있음

4) 정규화의 효과 및 문제점

- 1 중복된 데이터가 제거되어 갱신 이상이 감소함
- 2 데이터의 일관성 유지가 가능함
- 3 데이터 모델이 단순해짐
- 4 무결성 제약조건을 만족시키기 위해서 필요한 프로그램 코드 양이 감소함

- 즉, DBMS가 자동으로 처리해 주므로 응용 프로그램에서 처리해야 할 일의 양이 감소함

정규화의 문제점!!

: 릴레이션(테이블) 분해로 인해 많은 조인(Join)* 연산이 발생하므로 질의에 대한 응답 시간이 느려질 수 있음

두 릴레이션과 관련된 튜플을 하나의 튜플로 결합하는 연산

1. 정규화 개념

4) 정규화의 효과 및 문제점



여기서 잠깐!

조인(Join) 연산

1) 조인 연산이란?

- 두 릴레이션과 관련된 튜플을 하나의 튜플로 결합하는 연산으로, 두 릴레이션의 각 튜플들을 1:1로 대응시킨 다음, 조건에 맞는 튜플들만 구하는 연산임
- 세타 조인, 동일 조인, 자연 조인 등 여러 가지 유형의 조인 연산이 있음
- 관계 대수에서는 조인 연산자로 \bowtie 기호를 사용함

2) 조인 연산 예제

학생 릴레이션과 교수 릴레이션의 조인 (조건: 학생.지도교수 = 교수.교수번호)

학생 릴레이션

학번	주민등록번호	이름	주소	휴대폰	지도교수
98201002	800321	김정욱	천안	010-1111-2222	01223
99202012	811123	이기원	서울	010-2222-3333	02355
98201025	810923	박문구	대전	010-3333-4444	01223

교수 릴레이션

교수번호	이름	연구실	휴대폰
01223	장경구	A112	010-1111-2222
02355	김민숙	B344	010-2222-3333
05378	이장수	A322	010-3333-4444

학생 릴레이션의 지도교수 속성의 값과 교수 릴레이션의 교수번호 속성의 값이 같은 것끼리 조인하라는 의미

➔ 학생 \bowtie 지도교수=교수번호 교수 :

학생과 교수 릴레이션을 조인한 릴레이션

학번	주민등록번호	이름	주소	휴대폰	지도교수	교수번호	이름	연구실	휴대폰
98201002	800321	김정욱	천안	010-1111-2222	01223	01223	장경구	A112	010-1111-2222
99202012	811123	이기원	서울	010-2222-3333	02355	02355	김민숙	B344	010-2222-3333
98201025	810923	박문구	대전	010-3333-4444	01223	05378	이장수	A322	010-3333-4444

2. 정규형 구분

1) 정규형의 구분

정규형(Normal Form)*

어떤 제약 조건을 만족하는 형태

1

'데이터의 불필요한 중복 감소'와 '응답시간 단축'이라는 상반된 목표를 달성하기 위함

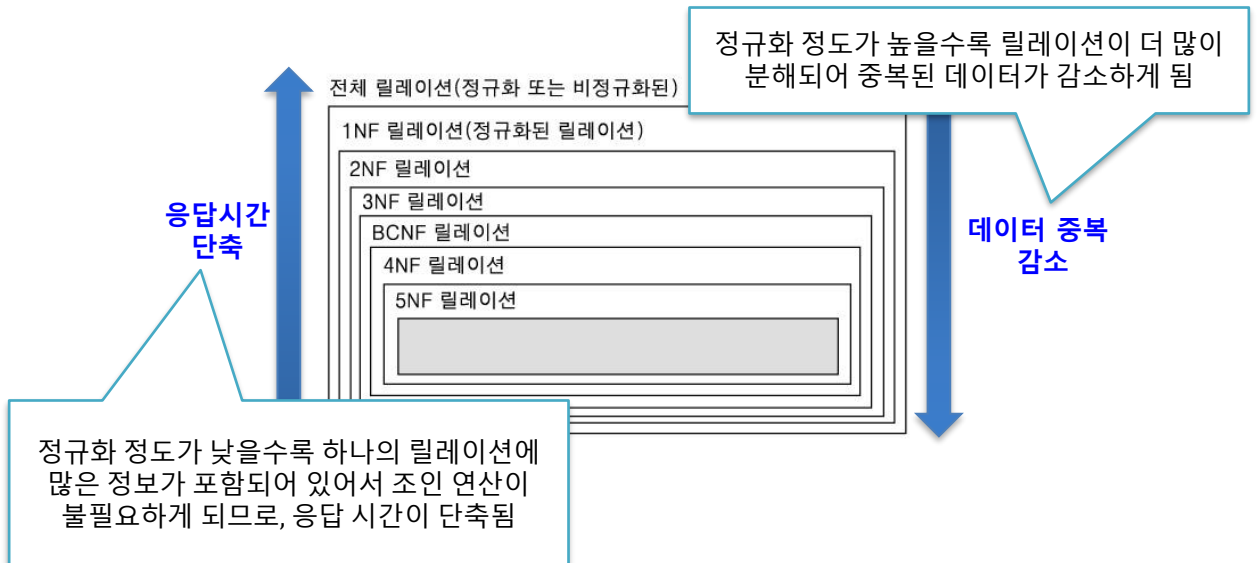
2

릴레이션이 만족해야 하는 제약 조건을 구분함

3

제약 조건에 따라 제1정규형 ~ 제5정규형 그리고 보이스코드 정규형으로 구분함

2) 정규형의 포함 관계





2. 정규형 구분

3) 단계별 정규형과 제약 조건

정규형 단계	제약 조건	정규화 방법
제 1 정규형 (1NF)	모든 속성의 도메인이 원자 값이어야 함	다중치 속성 및 중복 속성을 분리 (다중치와 반복 그룹 제거)
제 2 정규형 (2NF)	모든 속성이 기본 키에 완전 함수적 종속이어야 함	기본 키에 부분 함수적 종속인 속성을 분리 (부분 함수적 종속 제거)
제 3 정규형 (3NF)	속성들 간에 이행적 함수적 종속이 없어야 함	이행적 함수적 종속인 속성을 분리 (이행적 함수적 종속 제거)
보이스/코드 정규형 (BCNF)	후보 키가 아닌 결정자가 없어야 함	후보 키가 아니면서 결정자 역할을 하는 속성과 그 종속자를 분리 (모든 결정자가 후보 키)
제 4 정규형 (4NF)	다치 종속(Multi-valued Dependency)인 속성이 없어야 함	다치 종속인 속성을 분리
제 5 정규형 (5NF)	조인 종속(Join Dependency)인 속성이 없어야 함	조인 종속인 속성을 분리

4) 제 3 정규형이 가장 좋은 이유

 정규화의 **최종 목표**를 **3NF**나 **BCNF**로 정하는 것이 BEST! 

구분	문제점
1NF, 2NF	<ul style="list-style-type: none"> 3NF와 비교하면 불필요한 중복이 많음 불필요한 중복으로 인해 기억공간의 낭비가 큼
4NF, 5NF	<ul style="list-style-type: none"> 3NF와 비교하면 릴레이션 수가 지나치게 많아질 가능성이 있음 → DB를 생성하면 테이블이 과다하게 세분화되어, 조인(Join) 횟수가 늘어나서 질의에 대한 응답시간이 늦어짐 다치 종속과 조인 종속이라는 복잡한 형태의 종속성까지 고려해야 함

3. 단계별 정규화

1) 정규화 단계

일반적으로 제 1 정규형부터 보이스/코드(BCNF) 정규형까지 단계별로 진행함

실제 정규화 과정은 정규형 순서와 다를 수 있음

어떤 정규형은
건너뛸 수 있음

정규화는 실제 데이터 값이 아닌 개념적 측면에서 수행되어야 함



3. 단계별 정규화

2) 비정규형

다중치 속성을 포함하고 있는 비정규형

복수 개의 값 각각에 대해 하나씩의 튜플을 생성해서 새로운 릴레이션(제 1 정규형)을 만들

예) '동아리'라는 다중치 속성을 포함하고 있는 비정규형 테이블인 학생1 테이블 (기본 키: 학번)

학생1

학번	이름	학과	동아리
20013426	박하늘	컴퓨터학과	{영어회화반, 검도부}
20025714	홍길동	영문학과	{수화반, 합창반}
20038540	김동아	음악학과	미술반
99570264	이몽룡	사회복지학과	검도부
97456123	최푸름	국어국문과	축구부

동아리라는 다중치 속성이 존재하므로 제1정규형을 만족하지 않음

릴레이션의 모든 속성은 원자 값만을 가져야 하므로, 동아리 속성의 복수 개의 값 각각에 대해 하나씩의 튜플을 생성해서 학생2라는 새로운 릴레이션(기본 키: {학번, 동아리})을 만들

학생2

학번	이름	학과	동아리
20013426	박하늘	컴퓨터학과	영어회화반
20013426	박하늘	컴퓨터학과	검도부
20025714	홍길동	영문학과	수화반
20025714	홍길동	영문학과	합창반
20038540	김동아	음악학과	미술반
99570264	이몽룡	사회복지학과	검도부
97456123	최푸름	국어국문과	축구부

- 학생2 릴레이션은 제1정규형의 제약 조건(모든 속성의 도메인이 원자 값이어야 함)을 만족함
- 학생2 릴레이션의 기본 키는 {학번, 동아리}임
- 만약 학번만 기본 키로 정의하는 경우, 2번째, 4번째 튜플은 개체 무결성에 위반되므로 삽입될 수 없음

3. 단계별 정규화

2) 비정규형

중복된 속성을 포함하고 있는 비정규형

동일한 값이 중복되어 나타나는 반복 그룹을 분리해서
2개의 새로운 릴레이션(제 1 정규형)을 만들

예) 동일한 {학번, 이름, 학과} 값이 반복되는 학생2 릴레이션 (기본 키: {학번, 동아리})

학생2

학번	이름	학과	동아리
20013426	박하늘	컴퓨터학과	영어회화반
20013426	박하늘	컴퓨터학과	검도부
20025714	홍길동	영문학과	수화반
20025714	홍길동	영문학과	합창반
20038540	김동아	음악학과	미술반
99570264	이몽룡	사회복지학과	검도부
97456123	최푸름	국어국문과	축구부

학번, 이름, 학과 속성의
동일한 값이 반복됨

반복 그룹 {학번, 이름, 학과} 속성을 분리해서 학생3(기본 키: 학번) 릴레이션을 만들고, 나머지 속성인 동아리 속성과 기본 키인 학번을 분리해서 학생4(기본 키 : {학번, 동아리})라는 릴레이션을 만들

학생3

학번	이름	학과
20013426	박하늘	컴퓨터학과
20025714	홍길동	영문학과
20038540	김동아	음악학과
99570264	이몽룡	사회복지학과
97456123	최푸름	국어국문과

학생4

학번	동아리
20013426	영어회화반
20013426	검도부
20025714	수화반
20025714	합창반
20038540	미술반
99570264	검도부
97456123	축구부

학생3, 학생4
릴레이션은 모두 제
1 정규형의 제약
조건인 모든 속성의
도메인이 원자
값이어야 함을 만족

3. 단계별 정규화

3) 제 1 정규형 (1st Normal Form: 1NF)

제 1 정규형의 제약 조건

모든 속성의 도메인이 원자 값을 가져야 함

제 1 정규형의 갱신 이상의 원인

기본 키에 **부분 함수적 종속** 관계가 존재할 때 갱신 이상 현상이 발생



- 기본 키로 식별되는 개체와는 무관한 속성이 존재한다는 의미
- 즉, 두 가지 독립적인 정보가 하나의 릴레이션에 표현된 것

제 1 정규형의 갱신 이상 해결 방법

릴레이션을 분해해서 **부분 함수적 종속**을 제거하여 제 2 정규형을 만들

제 1 정규형의 갱신 이상 해결 예제 : 수강지도 릴레이션의 분해

- 수강지도(학번, 교과목번호, 지도교수, 학과, 성적)
- 기본 키 : {학번, 교과목번호}
- 함수적 종속 : {학번, 교과목번호} → 성적
 - 학번 → 지도교수
 - 학번 → 학과
 - 지도교수 → 학과
- FDD :

3. 단계별 정규화

3) 제 1 정규형 (1st Normal Form: 1NF)



릴레이션 확인

수강지도

학번	지도교수	학과	교과목번호	성적
100	P1	컴퓨터	C413	A
100	P1	컴퓨터	E412	A
200	P2	전기	C123	B
300	P3	컴퓨터	C312	A
300	P3	컴퓨터	C324	C
300	P3	컴퓨터	C413	A
400	P1	컴퓨터	C312	A
400	P1	컴퓨터	C324	A
400	P1	컴퓨터	C413	B
400	P1	컴퓨터	E412	C

3. 단계별 정규화

3) 제 1 정규형 (1st Normal Form: 1NF)

제 1 정규형의 갱신 이상 해결 예제 : 수강지도 릴레이션의 분해

제 1 정규형인 수강지도 릴레이션에 존재하는 갱신 이상 분석

삽입 이상	500번 학생의 지도교수가 P4라는 사실을 삽입하려 할 때, 어떤 과목을 수강하지 않는 한 삽입하는 것이 불가능함
삭제 이상	200번 학생이 C123의 수강을 취소해서 이 데이터가 삭제되는 경우, 지도교수가 P2라는 유용한 정보까지 삭제됨
수정 이상	400번 학생의 지도교수를 P1에서 P3로 변경하는 경우, 학번이 400인 4개 튜플의 지도교수를 모두 P3로 변경하지 않으면 데이터 불일치가 발생함

수강지도 릴레이션에 존재하는 갱신 이상 원인 분석

- 지도교수와 학과 속성이 기본 키인 {학번, 교과목번호}에 **부분 함수적 종속** 관계가 존재함

릴레이션 분해

- 부분 함수적 종속이 존재하지 않도록 다음과 같이 2개의 릴레이션으로 분해함

지도(학번, 지도교수, 학과)
수강(학번(FK), 교과목번호, 성적)

지도

<u>학번</u>	지도교수	학과
100	P1	컴퓨터
200	P2	전기
300	P3	컴퓨터
400	P1	컴퓨터

지도와 수강 릴레이션은 각각 제 2 정규형의 제약조건(모든 속성이 기본 키에 완전 함수적 종속이어야 함)을 만족

수강

<u>학번</u>	<u>교과목번호</u>	성적
100	C413	A
100	E412	A
200	C123	B
300	C312	A
300	C324	C
300	C413	A
400	C312	A
400	C324	A
400	C413	B
400	E412	C

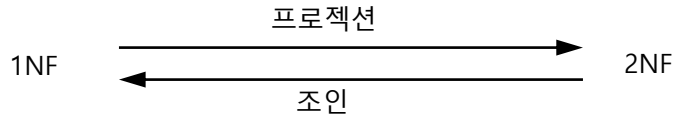
3. 단계별 정규화

3) 제 1 정규형 (1st Normal Form: 1NF)

무손실 분해(Nonloss Decomposition)

프로젝션(Projection)*으로 분해된 릴레이션들은 조인(Join)을 통해 원래의 릴레이션으로 복귀될 수 있어야 함

릴레이션의 특정 속성만으로 구성된 새로운 릴레이션을 구하는 연산



원래의 릴레이션에서 얻을 수 있는 정보는 분해된 릴레이션들로부터도 얻을 수 있으나, 그 역은 성립하지 않음

- 앞의 예에서, 500번 학생의 지도교수가 P4라는 정보는 2NF으로 분해된 지도 릴레이션에는 삽입할 수 있지만, 1NF인 수강지도 릴레이션에는 삽입할 수 없음



여기서 잠깐!

프로젝션(Projection) 연산

프로젝션 연산이란?

- 릴레이션의 특정 속성만으로 구성된 새로운 릴레이션을 구하는 연산
- 즉, 릴레이션의 수직적 부분 집합을 구하는 연산
- 관계 대수에서는 프로젝트 연산자로 Π 기호를 사용함

프로젝션 연산 예제

- 학생 릴레이션에서 학번, 이름, 휴대폰 속성만 프로젝트함

학생 릴레이션

학번	주민등록번호	이름	주소	휴대폰	지도교수
98201002	800321	김정욱	천안	010-1111-2222	01223
99202012	811123	이기원	서울	010-2222-3333	02355
98201025	810923	박문구	대전	010-3333-4444	01223

→ 학생 Π (학번, 이름, 휴대폰) :

학번	이름	휴대폰
98201002	김정욱	010-1111-2222
99202012	이기원	010-2222-3333
98201025	박문구	010-3333-4444

학생 릴레이션에서 학번, 이름, 휴대폰 속성만 프로젝트 하라는 의미

3. 단계별 정규화

4) 제 2 정규형 (2nd Normal Form: 2NF)

제 2 정규형의 제약 조건

모든 속성이 기본 키에 **완전 함수적 종속**이어야 함

제 2 정규형의 갱신 이상의 원인

이행적 함수적 종속 관계가 존재할 때 갱신 이상이 발생

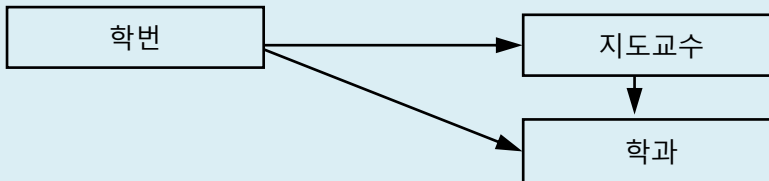
제 2 정규형의 갱신 이상 해결 방법

릴레이션을 분해해서 **이행적 함수적 종속을 제거**하여 제 3 정규형을 만들

제 2 정규형의 갱신 이상 해결 예제 : 지도 릴레이션의 분해

- 지도(학번, 지도교수, 학과)
- 기본 키 : 학번
- 함수적 종속 : 학번 \rightarrow 지도교수
학번 \rightarrow 학과
지도교수 \rightarrow 학과

▪ FDD :



릴레이션 확인

지도

학번	지도교수	학과
100	P1	컴퓨터
200	P2	전기
300	P3	컴퓨터
400	P1	컴퓨터

3. 단계별 정규화

4) 제 2 정규형 (2nd Normal Form: 2NF)

제 2 정규형의 갱신 이상 해결 예제 : 지도 릴레이션의 분해

제 2 정규형인 지도 릴레이션에 존재하는 갱신 이상 분석

삽입 이상	지도 학생 없이는 어떤 교수가 특정 학과에 속한다는 정보를 삽입하는 것이 불가능함
삭제 이상	300번 학생의 정보를 삭제하면, 지도교수 P3가 컴퓨터학과에 속한다는 유용한 정보까지 삭제됨
수정 이상	지도교수 P1의 소속을 컴퓨터에서 전기로 변경하는 경우, 학번이 100과 400인 2개 튜플의 학과를 모두 전기로 변경하지 않으면 데이터 불일치가 발생함 (실제 상황이라면 100과 200의 지도교수를 변경해야 함)

지도 릴레이션에 존재하는 갱신 이상의 원인 분석

- 학번, 지도교수, 학과라는 3개 속성 사이에 **이행적 함수적 종속** 관계가 존재

릴레이션 분해

- 이행적 함수적 종속이 존재하지 않도록 다음과 같이 2개의 릴레이션으로 분해함

학생지도(학번, 지도교수)

교수소속(교수, 학과)

학생지도와 교수소속 릴레이션은 모두 제 3 정규형의 제약조건(속성들 간에 이행적 함수적 종속이 없어야 함)을 만족

학생지도

<u>학번</u>	지도교수
100	P1
200	P2
300	P3
400	P1

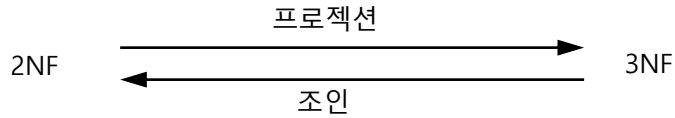
교수소속

<u>교수</u>	학과
P1	컴퓨터
P2	전기
P3	컴퓨터

3. 단계별 정규화

4) 제 2 정규형 (2nd Normal Form: 2NF)

무손실 분해(Nonloss Decomposition)



원래의 릴레이션에서 얻을 수 있는 정보는 분해된 릴레이션들로부터도 얻을 수 있으나,
그 역은 성립하지 않음



- 지도교수 P4가 전자에 속한다는 정보는 3NF으로 분해된 교수소속 릴레이션에는 삽입할 수 있지만, 2NF인 지도 릴레이션에는 삽입할 수 없음

3. 단계별 정규화

5) 제 3 정규형 (3rd Normal Form: 3NF)

제 3 정규형의 제약 조건

속성들 간에 **이행적 함수적 종속이 없어야 함**

제 3 정규형의 갱신 이상의 원인

후보 키가 아닌 결정자가 존재할 때 갱신 이상 현상이 발생

제 3 정규형의 갱신 이상 해결 방법

후보 키가 아니면서 결정자 역할을 하는 속성과 그 종속자를 분리해서, **모든 결정자가 후보 키**가 되는 보이스/코드 정규형(BCNF)을 만들

3NF의 약점

복수개의 후보 키를 갖고 있고, 후보 키들이 복합 속성들로 구성되고, 후보 키들이 서로 중첩되는 경우 → 3NF로는 불충분하므로 보다 일반적인 BCNF로 변환해야 함

제 3 정규형의 갱신 이상 해결 예제 : 수강과목 릴레이션의 분해

▪ 수강과목(학번, 교과목, 교수)

▪ 제약 조건

- ① 한 학생은 각 과목에 대해 오직 한 교수의 강의만 수강함
- ② **각 교수는 한 과목만 담당함**
- ③ 같은 과목을 여러 교수가 담당할 수 있음

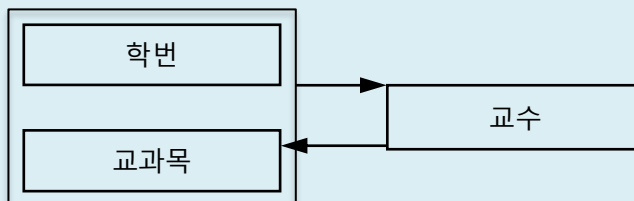
▪ 후보 키 : {학번, 교과목}, {학번, 교수}

▪ 기본 키 : {학번, 교과목}

▪ 함수적 종속 : {학번, 교과목} → 교수

교수 → 교과목

▪ FDD :



3. 단계별 정규화

5) 제 3 정규형 (3rd Normal Form: 3NF)



릴레이션 확인

수강과목

학번	교과목	교수
100	프로그래밍	P1
100	자료구조	P2
200	프로그래밍	P1
200	자료구조	P3
300	자료구조	P3
300	프로그래밍	P4

3. 단계별 정규화

5) 제 3 정규형 (3rd Normal Form: 3NF)

제 3 정규형의 갱신 이상 해결 예제 : 수강과목 릴레이션의 분해

제 3 정규형인 수강과목 릴레이션에 존재하는 갱신 이상 분석

삽입 이상	교수 P5가 자료구조 교과목을 담당한다는 정보의 삽입은 학번(수강 학생)이 없으면 불가능함
삭제 이상	100번 학생이 자료구조를 취소해서 데이터를 삭제하면, P2가 담당교수라는 유용한 정보까지 삭제됨
수정 이상	지도교수 P1이 프로그래밍 과목 대신 자료구조를 담당하게 되면, P1이 포함된 모든 튜플을 변경하지 않으면 데이터 불일치가 발생함

수강과목 릴레이션에 존재하는 갱신 이상 원인 분석

- 교수 속성이 **후보 키가 아닌데 결정자 역할**을 함

3. 단계별 정규화

5) 제 3 정규형 (3rd Normal Form: 3NF)

릴레이션 분해

교수과목(교수, 교과목) ----- (ㄱ)
수강교수(학번, 교수(FK)) ----- (ㄴ)

- (ㄱ) 후보 키가 아니면서 결정자 역할을 하는 '교수' 속성과 그 결정자에 함수적으로 종속하는 속성인 '교과목'을 별도의 릴레이션(교수과목)으로 분해하고, 결정자인 교수 속성을 기본 키로 정의
교수과목

<u>교수</u>	교과목
P1	프로그래밍
P2	자료구조
P3	자료구조
P4	프로그래밍

- (ㄴ) 기존 릴레이션에 결정자(교수 속성)를 남겨서 기본 키의 구성 요소가 되도록 하고, 이 결정자가 새로운 릴레이션(교수과목)에 대한 외래 키 역할도 하도록 함

수강교수

<u>학번</u>	<u>교수</u>
100	P1
100	P2
200	P1
200	P3
300	P3
300	P4

교수과목과 수강교수 릴레이션은 모두 BCNF의 제약조건인 후보 키가 아닌 결정자가 없어야 함을 만족

3. 단계별 정규화

6) 보이스/코드 정규형 (Boyce/Codd Normal Form: BCNF)

BCNF의 제약 조건

후보 키가 아닌 결정자가 없어야 함

BCNF가 고안된 이유

기본 키가 둘 이상의 속성으로 구성된 복합 키이고, 복합 키가 둘 이상 존재할 경우에 발생하는 갱신 이상 현상을 방지하기 위해서 고안됨


BCNF의 특징

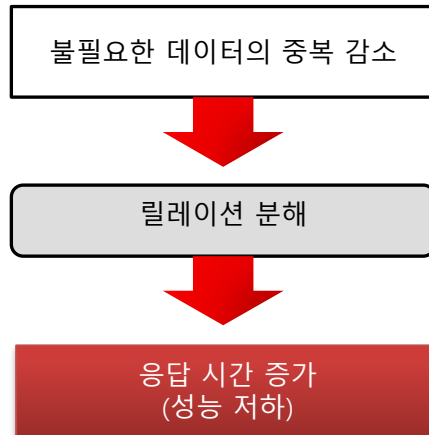
하나의 후보 키만 가진 릴레이션이 3NF이면 동시에 BCNF도 만족하며, 이를 **강한 3NF(Strong 3NF)**이라고도 칭함

4. 역정규화

1) 정규화의 문제점

정규화 단계가 진행될수록 릴레이션이 분해되므로 원하는 정보를 얻기 위해서 조인의 필요성이 증가되어 성능이 저하될 수 있음

 불필요한 데이터의 중복은 감소하지만 질의에 대한 응답 시간은 증가할 수 있음

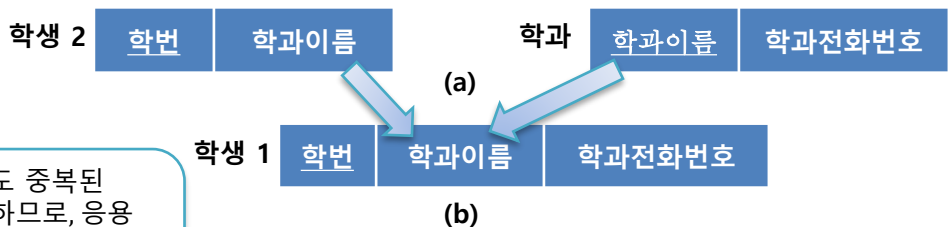


2) 역정규화(Denormalization)

역정규화(Denormalization)란?

- 성능에 대한 요구를 만족시키기 위해 데이터 중복과 갱신 이상을 대가로 치르면서 보다 낮은 정규형으로 되돌아가는 것을 의미
- 즉, 빈번하게 수행되는 검색 질의의 수행 속도를 높이기 위해서 이미 분해된 두 개 이상의 릴레이션을 합쳐서 하나의 릴레이션으로 만드는 작업임

예



역정규화 하더라도 중복된 데이터는 관리해야 하므로, 응용 프로그램에서 별도로 데이터 무결성을 보장하기 위한 비용이 발생될 수 있음

(a) : 제 3 정규형을 만족하도록 분해된 두 릴레이션
(b) : 역정규화된 릴레이션

심터

나는 배웠다
샤를르 드 푸코

나는 배웠다.
다른 사람으로 하여금 나를 사랑하게 만들 수 없다는 것을.
내가 할 수 있는 일은 사랑받을 만한 사람이 되는 것뿐임을.
사람은 사랑하는 사람의 선택에 달린 일.

삶은 무엇을 손에 쥐고 있는가가 아니라
누가 곁에 있는가에 달려 있음을 나는 배웠다.
우리의 매력이라는 것은 15분을 넘지 못하고
그 다음은 서로를 알아가는 것이 더 중요함을.

다른 사람의 최대치에 나를 비교하기 보다는
나 자신의 최대치에 나를 비교해야 함을 나는 배웠다.
삶은 무슨 사건이 일어나는가에 달린 것이 아니라
일어난 사건에 대해 어떻게 대처하는 가에 달린 것임을.

사랑하는 사람에게는 언제나
사람의 말을 남겨 놓아야 함을 나는 배웠다.
어느 순간이 우리의 마지막 시간이 될지
아는 사람은 아무도 없으므로.

나는 배웠다.
나에게도 분노할 권리는 있으나
타인에 대한 몰인정하고 잔인하게 대할 권리는 없음을.
내가 바라는 방식대로 나를 사랑해 주지 않는다 해서
내 전부를 다해 사랑하지 않아도 좋다는 것이 아님을.

그리고 나는 배웠다.
아무리 내 마음이 아프다 하더라도 이 세상은
내 슬픔때문에 운영을 중단하지 않는다는 것을.
타인의 마음에 상처를 주지 않는 것과
내가 믿는 것을 위해 내 입장을 분명히 하는 것,
이 두가지를 엄격하게 구분하는 일이 얼마나 어려운가를.

나는 배웠다.
사랑하는 것과 사랑받는 것을.