데이터베이스

정규화와 제1, 2정규형

학습목표

- 이상 상태의 원인을 파악하고 정규화의 원칙을 설명할 수 있다.
- 함수 종속과 정규형과의 관계를 설명할 수 있다.

학습내용

- 이상 상태와 정규화의 원칙
- 제1, 2정규형

이상 상태와 정규화의 원칙

이상 상태

01 데이터의 논리적인 표현

실제 DB운영 시 직면하는 문제

Q 대량의 운영 데이터를 어떻게 조직하고 효율적으로 관리할 것인가?

- 데이터의 논리적 구조와 밀접한 관련이 있음
- 즉, DB 설계, DB의 논리적 설계와 관련되어 있음

Q 관계형 모델을 이용하여 어떻게 실세계를 표현할 것인가?

- 논리적 구조를 결정하는 것

Q DB 내에 어떤 릴레이션들이 필요하며,
각 릴레이션은 어떤 속성들로 구성할 것인가?

이상 상태와 정규화의 원칙

이상 상태

02 논리적 데이터베이스 설계

- 논리적 데이터베이스 설계의 유의점
  - 필요로 하는 속성, 개체, 관계성을 파악하고 관련된 속성들을 릴레이션으로 묶음
  - 속성들 사이의 관계성 고려
    - 관계성 고려를 하지 못할 경우 실제 데이터 처리 시에 이상 상태(Anomaly)의 부작용 발생
- 좋은 관계 스키마를 만들기 위한 속성들의 그룹핑

☑ 2단계 관계형 스키마

"User View" Level

"Base Relation" Level ➡ 설계와 연관

이상 상태와 정규화의 원칙

이상 상태

03 좋은 기반 릴레이션이란?

가이드 라인

01 하나의 릴레이션 속한 각 튜플은 실 세계에 존재하는 하나의 개체(Entity)만을 표현하여 함

- 개체(Entity): 꼭 물질적으로 존재하지 않아도 되며, 단순히 표현하고자 하는 정보의 대상을 의미

02 릴레이션의 각 튜플은 하나의 개체 인스턴스나 관계 인스턴스를 표현해야만 함

03 서로 다른 개체들의 속성은 하나의 테이블에 섞여서 나타나면 안됨

04 다른 개체를 참조하기 위해서는 외래키를 사용해야만 함

05 개체와 관계는 가능한 서로 분리시켜야 함

이상 상태와 정규화의 원칙

이상 상태

04 중복된 데이터의 저장

- 릴레이션에 속한 한 튜플이 복수 개의 개체들을 표현한다는 것은 데이터의 중복(Redundant) 발생을 유발

데이터의 중복 ➡ 저장 공간의 낭비 및 이상 상태 발생

- 삽입 이상(Insert Anomaly)
- 삭제 이상(Delete Anomaly)
- 변경 이상(Modification Anomaly) 또는 갱신 이상

이상 상태와 정규화의 원칙

이상 상태

05 이상 상태의 예

수강(학번, 과목번호, 성적, 학년)

- 기본키: 학번, 과목번호

학번	과목번호	성적	학년
100	C1	A	4
100	D1	C	4
200	C1	B	2
300	C2	C	1
300	E1	A	1

삭제 이상(Deletion Anomaly)

- 200번 학생이 'C1'과목 등록을 취소
- 2학년이라는 정보도 함께 삭제

학번	과목번호	성적	학년
100	C1	A	4
100	D1	C	4
200	C1	B	2
300	C2	C	1
300	E1	A	1

연쇄 삭제(Triggered Deletion)에 의한 정보 손실 (Loss of Information)

이상 상태와 정규화의 원칙

이상 상태

05 이상 상태의 예

삽입 이상(Insert Anomaly)

- 400번 학생이 2학년이라는 사실을 삽입 불가
- 어떤 과목을 등록하지 않는 한 삽입은 불가

원하지 않는 정보의 강제 삽입

학번	과목번호	성적	학년
100	C1	A	4
100	D1	C	4
200	C1	B	2
300	C2	C	1
300	E1	A	1

갱신 이상(Update Anomaly)

- 300번 학생의 학년은 1에서 2로 변경
- 학번 300에 대한 2개 튜플 모두 변경 필요

중복 데이터의 일부 갱신에 따른 불일치(Inconsistency)가 발생할 수 있음

학번	과목번호	성적	학년
100	C1	A	4
100	D1	C	4
200	C1	B	2
300	C2	C	1
300	E1	A	1

이상 상태와 정규화의 원칙

이상 상태

06 이상 상태의 원인

- 하나의 릴레이션의 한 튜플이 여러 개의 개체 정보를 나타내고 있음
- 하나의 릴레이션 내에 여러 개의 속성들 간의 연관성이 중복되어 나타남

» 예 | 수강(학번, 과목번호, 성적, 학년) 릴레이션의 경우

X학생이 C과목에서의 성적 개체와
X학생 몇 학년이란 개체가 혼재되어 나타남

07 이상 상태의 해결

속성들 간의 여러 연관 관계를 분해하여
별개의 릴레이션으로 표현

☑ 하나의 연관 관계는 하나의 릴레이션에서만 나타나도록 설계

➡ 정규화(Normalization)
(스키마 변환을 통하여 정규형을 만드는 것)

이상 상태와 정규화의 원칙

정규화의 원칙

01 정규형(Normal Form)

일련의 제약조건을 만족하는 릴레이션

제1정규형, 제2정규형, 제3정규형, BCNF

- 함수 종속을 기반한 정규화 과정을 다룸

제4정규형, 제5정규형

- 고급 정규화로 다치 종속과 조인 종속에 기반

02 원칙

01 정보 표현의 무손실

- 하나의 스키마에서 다른 스키마로 변환할 때 정보의 손실이 있어서는 안됨
- 변환된 스키마가 포함하고 있는 정보는 변환 전 스키마가 포함하고 있는 모든 정보를 포함하고 있어야 함

02 최소의 데이터 중복

03 분리의 원칙

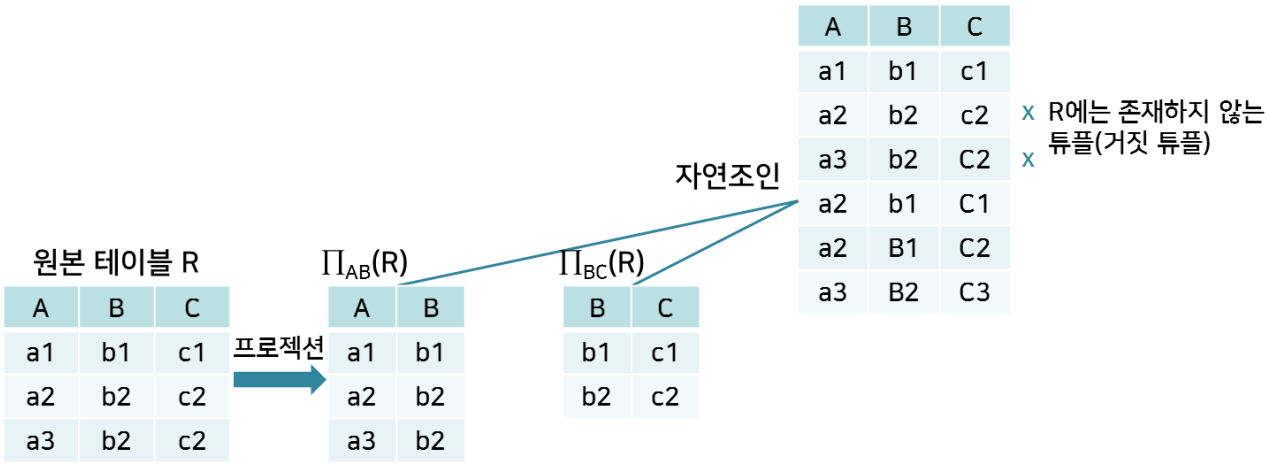
- 하나의 독립된 관계성은 별도의 분리된 릴레이션으로 표현함
- 이는 각 릴레이션을 독립적으로 처리할 수 있는 기초가 됨

이상 상태와 정규화의 원칙

정규화의 원칙

03 무손실 분해/무손실 조인

손실 분해/조인



릴레이션 R = {R1, R2, ..., Rn}은 다음 조건을 만족하는 경우

$$R = \Pi_{R1}(R) \bowtie \Pi_{R2}(R) \bowtie \ \ldots \ \bowtie \Pi_{Rn}(R)$$

- R은 분해된 테이블들로부터 복구 가능
- 분해된 테이블들을 모두 합쳐 자연 조인했을 때 추가적인 튜플이 생성되지 않아야 함
- 원본 릴레이션에서 얻을 수 있는 정보는 분해된 릴레이션들로부터도 얻을 수 있어야 함

제1, 2정규형

제1정규형

01 정의

- 모든 속성의 도메인은 원자값으로 이루어진 릴레이션
- 모든 관계형 릴레이션은 제1정규형을 만족
  - 릴레이션의 특성

튜플의 유일성

속성의 무순서성

튜플의 무순서성

속성의 원자성

제1, 2정규형

제2정규형

01 정의

- 1NF이고 키에 속하지 않은 속성들이 모두 기본키에 완전 함수 종속된 경우
  - 제2정규형의 특징
    - 제1정규형 만을 만족하는 릴레이션에서 이상상태(Anomaly)가 발생할 수 있음

02 제2정규형의 예

- 기본키: {학번, 과목번호}

학번	지도교수	학과	과목번호	성적
100	P1	컴공	C100	A
100	P1	컴공	C200	A
200	P2	메카	M100	B
300	P3	컴공	C100	A
300	P3	컴공	C200	A
400	P1	컴공	C300	C
400	P1	컴공	C200	A

삽입 이상

- 학번 500인 신입생
- 과목 신청이 안되면 등록 불가

제1, 2정규형

제2정규형

02 제2정규형의 예

기본키: {학번, 과목번호}

학번	지도교수	학과	과목번호	성적
100	P1	컴공	C100	A
100	P1	컴공	C200	A
200	P2	메카	M100	B
300	P3	컴공	C100	A
300	P3	컴공	C200	A
400	P1	컴공	C300	C
400	P1	컴공	C200	A

삭제 이상

- 200번 학생이 과목 M100 철회 시, 지도교수 P2라는 정보도 같이 삭제

학번	지도교수	학과	과목번호	성적
100	P1	컴공	C100	A
100	P1	컴공	C200	A
200	P2	메카	M100	B
300	P3	컴공	C100	A
300	P3	컴공	C200	A
400	P1	컴공	C300	C
400	P1	컴공	C200	A

갱신 이상

- 400번 학생이 지도교수를 P3으로 변경 할 경우, 2개의 튜플을 모두 고쳐야 함

제1, 2정규형

제2정규형

03 이상 상태의 원인

- 기본키로 식별되는 개체와는 무관한 정보들의 존재
- 한 튜플 내에 여러 개체들이 혼재되어 나타남
  - 튜플 내에 여러 개의 속성들 간의 연관성이 있음

함수 종속(Funtional Dependency: FD)

04 함수 종속(Functional Dependency: FD)의 정의

- 속성들 간의 연관성을 의미
  - 어떤 릴레이션 R에서 속성 X의 값에 대하여 속성 Y의 값이 하나만 연관될 때, 속성 Y는 속성 X에 함수 종속이라 하고, X→Y로 표현

[속성 X: 결정자(Determinant)]

[속성 Y: 종속자(Dependent)]

제1, 2정규형

제2정규형

05 함수 종속(Functional Dependency: FD)의 특징

01 릴레이션에서 속성 X가 키이면, R의 모든 속성 Y에 대하여 X→Y 성립

- 학번 값이 정해지면 해당 학번에 대한 학생 정보(이름, 주소, 학년 등)는 오직 하나만 나옴

02 함수 종속 X →Y에서 X가 꼭 키가 아니어도 됨

» 예 | 1자녀 정책의 경우: 보호자 명이 정해지면 학생이름이 하나만 나옴
보호자 → 학생명

03 X의 한 값에 대응되는 Y의 값을 가지는 튜플이 두 개 이상일 수 있음

- 같은 값을 가짐, 즉 속성값은 하나만 나옴

06 완전 함수 종속과 부분 함수 종속

함수 종속 X→Y에 대하여

완전 함수 종속	X의 부분 집합 X'에 대하여 X'→Y를 만족하는 X'이 존재하지 않을 때
부분 함수 종속	X의 부분 집합 X'에 대하여 X'→Y를 만족하는 X'이 존재할 때

제1, 2정규형

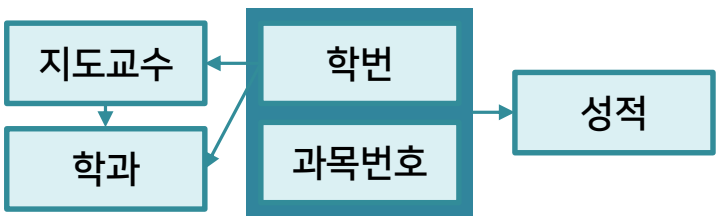
제2정규형

07 함수 종속(Functional Dependency: FD)의 파악

완전 함수 종속

- 기본키: {학번, 과목번호} → 성적{지도교수} → 학과

학번	지도교수	학과	과목번호	성적
100	P1	컴공	C100	A
100	P1	컴공	C200	A
200	P2	메카	M100	B
300	P3	컴공	C100	A
300	P3	컴공	C200	A
400	P1	컴공	C300	C
400	P1	컴공	C200	A



제1, 2정규형

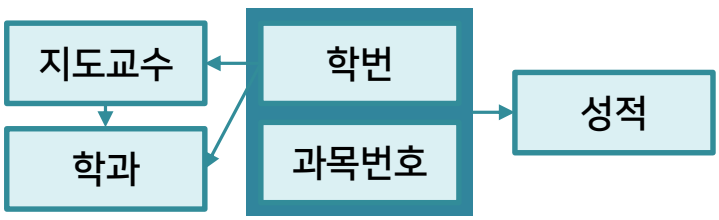
제2정규형

07 함수 종속(Functional Dependency: FD)의 파악

부분 함수 종속

- 기본키: {학번, 과목번호} → 지도교수 → {학번} → 지도교수

학번	지도교수	학과	과목번호	성적
100	P1	컴공	C100	A
100	P1	컴공	C200	A
200	P2	메카	M100	B
300	P3	컴공	C100	A
300	P3	컴공	C200	A
400	P1	컴공	C300	C
400	P1	컴공	C200	A



제1, 2정규형

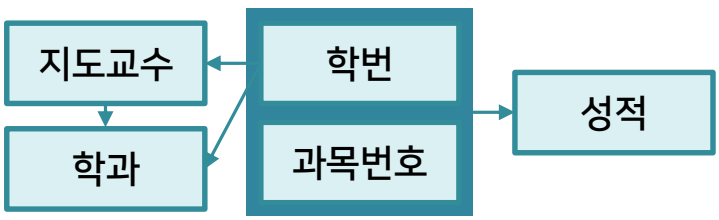
제2정규형

07 함수 종속(Functional Dependency: FD)의 파악

부분 함수 종속

- 기본키: {학번, 과목번호} → 학과 → {학번} → 학과

학번	지도교수	학과	과목번호	성적
100	P1	컴공	C100	A
100	P1	컴공	C200	A
200	P2	메카	M100	B
300	P3	컴공	C100	A
300	P3	컴공	C200	A
400	P1	컴공	C300	C
400	P1	컴공	C200	A

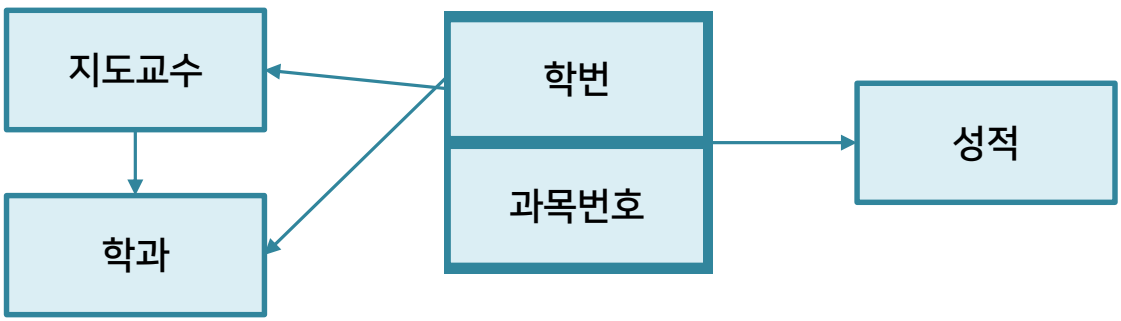


제1, 2정규형

제2정규형

08 1NF의 이상 상태 원인

- 기본키에 부분 함수 종속된 속성들 존재



기본키로 식별되는 개체와는 무관한 정보들 존재

제1, 2정규형

제2정규형

09 무손실 분해/조인의 조건

제1정규형인 릴레이션을 프로젝션 연산자을 통해서 분해하기 위한 방법

- 하나의 릴레이션을 2개로 쪼갤 때 (즉, R=R1∪R2)

$$R1 \cap R2 \rightarrow R1-R2 \quad or \quad R1 \cap R2 \rightarrow R2-R1$$

- R(A, B, C)에서 함수 종속 B→C가 존재할 경우
  - R1(A, B)와 R2(B, C)로 분해하면 무손실 분해가 됨

$$-\{A, B\} \cap \{B, C\} \rightarrow C(=\{B, C\}-\{A, B\})$$

원본 테이블 R

A	B	C
a1	b1	c1
a2	b2	c2
a3	b2	c2

프로젝션 → $\prod_{AB}(R)$

A	B
a1	b1
a2	b2
a3	b2

$\prod_{BC}(R)$

B	C
b1	c1
b2	c2

자연조인

A	B	C
a1	b1	c1
a2	b2	c2
a3	b2	C2

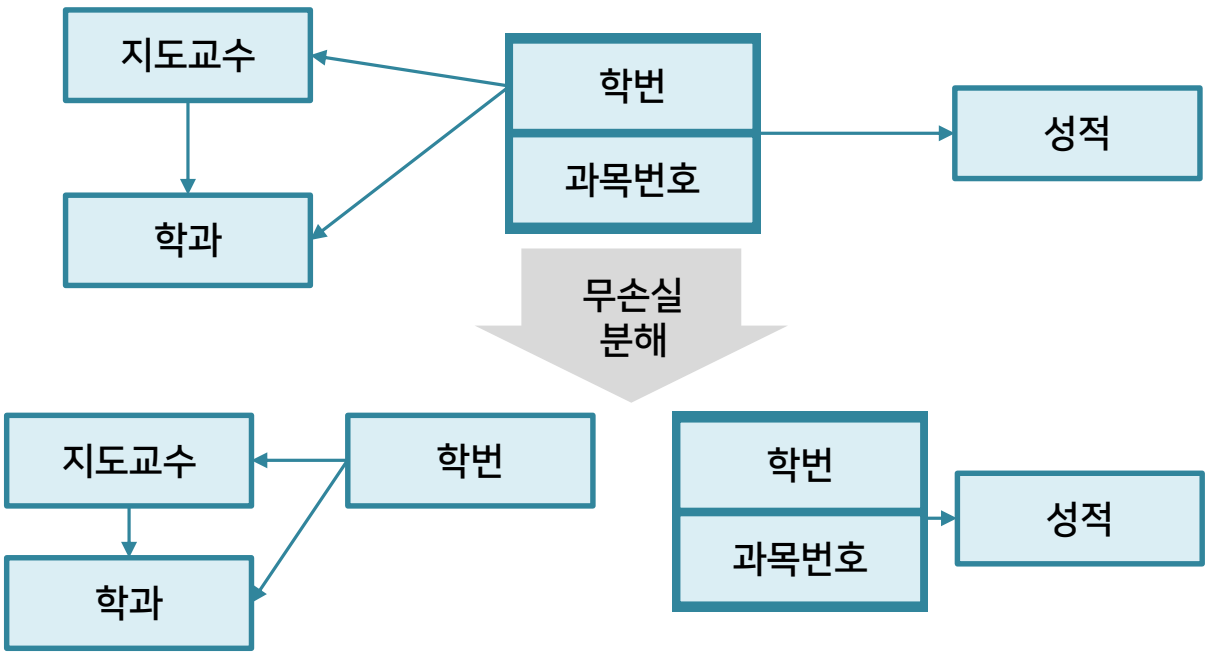
B→C인 함수 종속이 존재하므로
(A, B)와 (B, C)로 나누면 무손실 분해/조인

제1, 2정규형

제2정규형

10 무손실 분해/조인의 조건의 예

릴레이션의 무손실 분해/무손실 조인



1 이상 상태와 정규화의 원칙

✓ **데이터베이스 설계**

- 좋은 데이터베이스의 설계 기준
- 릴레이션의 각 튜플은 하나의 개체 인스턴스나 관계 인스턴스를 표현해야만 함

✓ **정규화**

- 이상 상태: 데이터 변경시 발생될 수 있는 문제점
- 하나의 객체에 속한 속성들 간에 존재하는 여러 개의 종속관계(Dependency)를 하나의 테이블에 표현되어 있기 때문
- 정규형: 일련의 제약 조건을 만족하는 릴레이션

2 제1, 2정규형

✓ **제1정규형: 속성값이 원자값으로만 구성**

✓ **제2정규형: 기본키를 제외한 속성들은 기본키에 완전 함수 종속**

✓ **함수 종속: 속성들 간의 연관 관계를 함수의 특성을 활용하여 파악하는 것**