릴레이션: 테이블 행: 레코드: 투플 열 : 필드 : 속성 정규화 : 모든 릴레이션(테이블)이 단일 주제를 갖도록 하는 것

용어

ACTIVITY (SID, Activity, Fee) 그림 4-5 Sample Data ACTIVITY 릴레이션

Activity SID 200 Skiing 100 50 Swimming 150 Squash 50 175 Swimming 200 50 1차 정규형 : 모든 속성은 하나의 값을 가져야 함 설명: SID 100 번 학생은 스키활동을 한다. 스키 비용은 200 달러이다 문제점: deletion abnormal: 100번 학생의 스키활동을 삭제하면..

Fee

insertion abnormal : scuba diving 이 175 달러라는 것을 등록하려고 하면...

문제의 원인: 하나의 릴레이션에 두개의 상이한 주제를 포함해서 발생 1. 각 활동에 참가하는 학생들

2. 각 활동의 참가비

ACTIVITIES (SID, Activity, Fee)

ACTIVITIES 릴레이션 200 Skiing 100 65 100 Golf 50 Swimming 150 50 Squash 175 50 Swimming 175 50 Swimming 200 Golf 65 200 2차 정규형 : 키가 아닌 모든 속성이 키 전체에 종속되는 릴레이션 문제: Activity -> Fee 에 종속됨. 부분종속이라고 함

STU-ACT (SID, Activity)

SID

Activity

ACT-COST (Activity, Fee)

Fee

200

50

50

Activity

Swimming

Skiing

Squash

SID → Dorm → Fee

SID Activity ACTIVITY를 나누어서 만든 두 릴레이션 100 Skiing 150 Swimming

175 Squash Swimming 200

그림 4-6

그림 4-9

(a) 이행적 종속을 가진 릴

이행적 종속의 제거

레이션

개선된 릴레이션

(b) 이행적 종속이 제거된 SID Dorm Fee 릴레이션 100 Randolph 3200 150 Ingersoll 3100 Randolph 200 3200 250 Pitkin 3100 Randolph 300 3200 (a) STU-HOUSING (SID, Dorm) BLDG-FEE (Dorm, Fee) Dorm Dorm Fee SID Randolph 3200 Randolph 100 Ingersoll Ingersoll 3100 150 Randolph Pitkin 200 3100 Pitkin 250 Randolph 300 (b) 3차 정규형: 1차 정규형 이면서 이행 종속이 존재하지 않는 릴레이션

HOUSING (SID, Dorm, Fee)

dependencies: Dorm → Fee

Kev: SID

Functional

Cauchy 100 Jung 150

한 학생은 하나의 기숙사에 거주할 수 있다. 기숙사의 사용료는 일정하다

SID -> Dorm -> Fee 이므로 간접적으로 SID -> Fee 가 성립함

이를 개선하면 그림 4-9(b) 와 같이 된다

그림 4-10

(a) 3차 정규형이면서 보이

(b) 보이스-코드 정규형 릴

따라서 이 조합들 모두 키가 될 수 있다.

말함

스-코드 정규형이 아닌

보이스-코드 정규형

릴레이션

레이션

100번 학생은 randolph 기숙사를 사용한다. 사용료는 3200 달러 이다.(연간 사용료겠지?)

ADVISER (SID, Major, Fname)

Key (candidate): (SID, Fname)

dependencies: Fname → Major

Math

Math

Math

Major

Psychology

Fname

Cauchy

Riemann

Cauchy

Jung

Functional

SID

100 150

200

250

즉 SID 는 Fee 속성을 Dorm 속성을 통해 결정하게 되는데 이를 이행 종속이라고 한다

Psychology Peris 300 Math 300 Riemann (a) ADV-SUBJ (Fname, Subject) STU-ADV (SID, Fname) Fname Subject Fname SID Math Cauchy Psychology Jung Riemann Math Riemann 200 Psycholog Peris Cauchy 250 Peris 300 Riemann 300 (b) BCNF(보이스 코드 정규형) : 모든 결정자가 후보키인 릴레이션을 말한다 한 학생은 하나 이상의 전공을 가진다. 한 전공은 여러 교수를 지도교수로 한다. 한 교수는 한 전공분야만 지도한다. 모든 교수의 이름은 서로 다르다. 교수의 이름을 결정하는 것은 SID, Major 조합이다 Major 를 결정하는 것은 SID, Fname 조합이다

칙 키 : 투플에 대한 유일한 식별자

Key: SID

SID

도메인: 주어진 속성에 대해 허용된 값을 기술하는 것

4-10(a) 릴레이션은 1,2,3차 정규형을 만족하지만 변경 이상 현상은 발생한다

ex) 300번 학생을 삭제하면 Math 의 Riemann 교수도 삭제

Constraints: Dorm → Fee

STUDENT (SID. GradeLevel, Dorm, Fee)

SID must not begin with digit 1

Values

4 decimal digits, first digit not 1

도메인 / 키 정규형 : 릴레이션에 대한 모든 제약이 키와 도메인의 정의에 다른 논리적 결과인 릴레이션을

제약 : 시간 종속적인 제약을 제외한 속성들의 정적인 값의 참 여부를 결정할 수 있는 모든 규

그림 4-15 Domain Definitions: **Attribute**

예 1의 도메인 키 정의

Dorm: 기숙사

Fee: 기숙사 거주비

4-14 가 4-15 가 되는게 맞는지 생각해보자

그림 4-16

그림 4-17

예 2의 도메인 키 정의

DK/NF의 예 3

함)

그림 4-19

여 3의 도메인 키 정의

DK/NF의 예 2

SID: 학생 아이디 GradeLevel: 학년

그림 4-14

DK/NF의 예 1

{'FR', 'SO', 'JR', 'SN', 'GR'} GradeLevel StudentYear Dorm BuildingNames Char(4) StudentFees Fee Any currency value Relation and Key Definitions:

PROFESSOR (FID, Fname, Class, SID, Sname)

FID → Fname Fname → FID

SID -FID SID → Fname SID → Sname

Domain Name

ClassNames

PersonNames

FacultyID PersonNames

StudentID

FID → → Class | SID Fname → → Class | SID

FID must start with 1; SID must not start with 1

4 decimal digits, first digit is 1

4 decimal digits, first digit is not 1

Char(10); values {list of valid course names}

Values

Char(50)

Char(50)

Constraints:

Domain Definitions:

Attribute

FID

Fname

Sname

Key: SID

GradFacultyStatus : 학부/대학원 지도교수 속성값

Domain Definitions:

Attribute

Fname

GSID

UGSID

Sname

Gfname

FID

Constraints:

Class

SID

STUDENT (SID, GradeLevel, Dorm)

BLDG-FEE (Dorm, Fee)

StudentID

Domain Name

Relation and Key Definitions: FACULTY (FID, Fname) Candidate key: Fname PREPARATION (Fname, Class) STUDENT (SID, Sname, Fname) FID : 교수 **아이디** Fname: 교수이름 SID: 학생 아이디 Class: 학급 Sname: 학생 이름 4-16 가 4-17 가 되는게 맞는지 생각해보자 faculty: 교수 정보 preparation : 교수-학급 student: 학생 정보 STU-ADVISER (SID, Sname, FID, Fname, GradFacultyStatus) 그림 4-18

FID → Fname

Fname -- FID

FID begins with 1

대학원교수는 대학원 학생만 가르칠 수 있다(이 말은 학부 교수는 학부생만 가르칠 수 있다는 의미이기도

Domain Name

PersonNames

GradStudentID

PersonNames

PersonNames

FacultyID

GradFacultyStatus FacultyStatus

Relation and Key Definitions:

릴레이션 정의*

다른 속성 추가 규칙

N:1 속성 관계: A 가 B 를 결정하고, B 가 A 를 결정하지 않는다.(투자자: 상품)

ex) 한 교수가 여러 학급을 가르치고 한 학급이 여러 교수에게 배울 수 있다

● A 또는 B에 의해 함수적으로 결정되는 속성만을 R에 추가할 수 있다. ● A 또는 B에 의해 함수적으로 결정되지 않는 속성은 R에 추가할 수 없다.

● A와 B는 R에서는 함께 나타나야 하지만, 다른 릴레이션에서는 함께 나타나지 않아도 된다. ● A 또는 B는 R이 아닌 릴레이션 내에서는 쌍을 표현하도록 일관성 있게 사용해야 한다.

종속성

71

M:N 속성 관계: A, B 모두 서로를 결정하지 않는다

• A 또는 B 중 하나가 R의 키가 되어야 한다.

● C에 의해 결정되는 속성만을 S에 추가할 수 있다. • C에 의해 결정되지 않는 속성은 S에 추가할 수 없다.

● (E, F)에 의해 결정되는 속성만을 T에 추가할 수 있다. ● (E, F)에 의해 결정되지 않는 속성은 T에 추가할 수 없다.

• C는 S의 키가 되어야 한다.

• T의 키는 (E, F)이어야 한다.

릴레이션 구축을 위한 규칙 요약

일 대 일 관계의 경우

다 대 일 관계의 경우

다 대 다 관계의 경우

한다고 하자.

4.8.1 부자연스러운 정규화

다음의 릴레이션을 고려해보자.

다면 역정규화가 더 좋기도 하다

가/아닌가?

UG_ADV (UGSID, Sname, Fname)

SID must not begin with 1

FID and Fname - GradFacultyStatus

SID of graduate student begins with 9

Values

Char(50)

Values {0, 1}

4 decimal digits, first digit is 1

4 decimal digits, first digit is 9

Values (FACULTY. Fname where

다대일

S(C,D)

C→D

D++C

C

C→E

다대다

T(E,F)

E+→F

F → E

(E,F)

(E,F) → G

UnderGradStudentID 4 decimal digits, first digit is not 1 and not 9

GradFacultyStatus = 1 }

Only graduate faculty can advise graduate students

GradFacultyStatus = 10 for undergraduate faculty

1 for graduate faculty

FACULTY (FID, Fname, GradFacultyStatus) Candidate key: Fname G_ADV (GSID, Sname, Gfname)

그림 4-21

세 가지 유형의 속성 관계의 요약

* 이 릴레이션 정의에 사용된 문자들은 그림 4-22에서와 동일 1:1 속성 관계 : A 가 B 를 결정하고, B 도 A 를 결정한다.(기혼자의 배우자 관계)

그림 4-22

● 일 대 일 관계를 가지는 속성들은 최소한 하나의 릴레이션 내에 함께 나타나야 한다. 릴레이션을 R, 속성을 A와 B라 하자.

● 다 대 일 관계를 가지는 속성들은 하나의 릴레이션 내에 함께 존재할 수 있다. 릴레이션 S에서 C가 D를 결정한다고 하자.

● 다 대 다 관계를 가지는 속성들은 하나의 릴레이션 내에 함께 존재할 수 있다. 이러한 속성 E와 F가 릴레이션 T에 함께 존재

속성 관계의 유형

일대일

R(A,B)

A→B

B→A

Either A or B

Either A or B → C

• 새로운 속성 G의 추가시 키가 (E. F. G)로 확장된다면, 릴레이션의 주제가 바뀌게 된다. G를 T에 속하지 않게 하든지, 새로 운 주제를 반영하기 위해 T의 이름을 변경하여야 한다.

이 릴레이션은 키인 CustNumber에 의해 암시되지 않는 함수 종속 Zip → (City, State) 를 포함하고 있으므로 DK/NF가 아니다. 즉 도메인과 키의 정의에 의해서 암시되지 않는 제약이 존재한다. 보통의 정규화 과정을 따르면 이 릴레이션을 다음과 같이 2개의 DK/NF 릴레이션으로 분해할 수 있다. CUSTOMER(CustNumber, CustName, Zip) CODES(Zip, City, State)

CUSTOMER(CustNumber, CustName, City, State, Zip)

조 무결성 제약이 성립한다. 이 두 테이블은 도메인/키 정규형이지만 좋은 설계 결과라고 보기는 어렵다. 사용자가 고객의 시(city)와 주(state)를 알고자 할 때 마다 CODES 테이블을 조사(즉 해당 시와 주 데이터를 가진 행을 판독)해야 한다. 이 설계가 정규화되지 않았다면 시와 주 데이터가 고 객의 나머지 데이터와 같이 저장되어 있을 것이므로 추가적인 조사는 필요하지 않게 된다. 또한 시와 주 데이터를 중복시키는 단점이 심각하지는 않을 것이다.

역정규화: 정규화된 설계가 부자연스럽거나 불편하거나 데이터 사이즈에 따른 성능상이 이슈를 만들어 낸

ex1) 고객 정보에 거주지와 관련된 정보를 저장. 행정구역이 바뀌는 경우 기존 주소를 업데이트 해야 하는

정규화가 되어있다면 고객 정보에 기존 주소정보를 저장하지 않는 한 주소정보가 CODES 에만 존재하게

이 두 릴레이션에는 CUSTOMER.Zip의 값이 CODES.Zip의 값으로 존재해야 하는 참