

용어
릴레이션 : 테이블
행 : 레코드 : 튜플
열 : 필드 : 속성

정규화 : 모든 릴레이션(테이블)이 단일 주제를 갖도록 하는 것

그림 4-5

ACTIVITY 릴레이션

ACTIVITY (SID, Activity, Fee)
Sample Data

SID	Activity	Fee
100	Skiing	200
150	Swimming	50
175	Squash	50
200	Swimming	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..
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

그림 4-8

ACTIVITIES 릴레이션

ACTIVITIES (SID, Activity, Fee)

SID	Activity	Fee
100	Skiing	200
100	Golf	65
150	Swimming	50
175	Squash	50
175	Swimming	50
200	Swimming	50
200	Golf	65

2차 정규형 : 키가 아닌 모든 속성이 키 전체에 종속되는 릴레이션

문제 : Activity -> Fee 에 종속됨. 부분종속이라고 함

그림 4-6

ACTIVITY를 나누어서 만든 두 릴레이션

STU-ACT (SID, Activity)
ACT-COST (Activity, Fee)

SID	Activity
100	Skiing
150	Swimming
175	Squash
200	Swimming

Activity	Fee
Skiing	200
Swimming	50
Squash	50

그림 4-9

이행적 종속의 제거

(a) 이행적 종속을 가진 릴레이션
(b) 이행적 종속이 제거된 릴레이션

HOUSING (SID, Dorm, Fee)
Key: SID
Functional dependencies: Dorm → Fee
SID → Dorm → Fee

SID	Dorm	Fee
100	Randolph	3200
150	Ingersoll	3100
200	Randolph	3200
250	Pitkin	3100
300	Randolph	3200

(a)

STU-HOUSING (SID, Dorm)
BLDG-FEE (Dorm, Fee)

SID	Dorm
100	Randolph
150	Ingersoll
200	Randolph
250	Pitkin
300	Randolph

Dorm	Fee
Randolph	3200
Ingersoll	3100
Pitkin	3100

(b)

3차 정규형 : 1차 정규형 이면서 이행 종속이 존재하지 않는 릴레이션

한 학생은 하나의 기숙사에 거주할 수 있다. 기숙사의 사용료는 일정하다
100번 학생은 randolph 기숙사를 사용한다. 사용료는 3200 달러 이다.(연간 사용료겠지?)

SID -> Dorm -> Fee 이므로 간접적으로 SID -> Fee 가 성립함
즉 SID 는 Fee 속성을 Dorm 속성을 통해 결정하게 되는데 이를 이행 종속이라고 한다

이를 개선하면 그림 4-9(b) 와 같이 된다

그림 4-10

보이스-코드 정규형

(a) 3차 정규형이면서 보이스-코드 정규형이 아닌 릴레이션
(b) 보이스-코드 정규형 릴레이션

ADVISER (SID, Major, Fname)
Key (candidate): (SID, Fname)
Functional dependencies: Fname → Major

SID	Major	Fname
100	Math	Cauchy
150	Psychology	Jung
200	Math	Riemann
250	Math	Cauchy
300	Psychology	Perls
300	Math	Riemann

(a)

STU-ADV (SID, Fname)
ADV-SUBJ (Fname, Subject)

SID	Fname
100	Cauchy
150	Jung
200	Riemann
250	Cauchy
300	Perls
300	Riemann

Fname	Subject
Cauchy	Math
Jung	Psychology
Riemann	Math
Perls	Psychology

(b)

BCNF(보이스 코드 정규형) : 모든 결정자가 후보키인 릴레이션을 말한다

한 학생은 하나 이상의 전공을 가진다. 한 전공은 여러 과수를 지도교수로 한다. 한 과수는 한 전공분야만 지도한다. 모든 과수의 이름은 서로 다르다.

과수의 이름을 결정하는 것은 SID, Major 조합이다
Major 를 결정하는 것은 SID, Fname 조합이다
따라서 이 조합들 모두 키가 될 수 있다.

4-10(a) 릴레이션은 1,2,3차 정규형을 만족하지만 변경 이상 현상은 발생한다
ex) 300번 학생을 삭제하면 Math 의 Riemann 교수도 삭제

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

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

키 : 튜플에 대한 유일한 식별자

도메인 : 주어진 속성에 대해 허용된 값을 기술하는 것

그림 4-14

DK/NF의 예 1

STUDENT (SID, GradeLevel, Dorm, Fee)
Key: SID
Constraints: Dorm → Fee
SID must not begin with digit 1

그림 4-15

예 1의 도메인 키 정의

Domain Definitions:

Attribute	Domain Name	Values
SID	StudentID	4 decimal digits, first digit not 1
GradeLevel	StudentYear	{ 'FR', 'SO', 'JR', 'SN', 'GR' }
Dorm	BuildingNames	Char(4)
Fee	BuildingFees	Any currency value

Relation and Key Definitions:

STUDENT (SID, GradeLevel, Dorm)
BLDG-FEE (Dorm, Fee)

Dorm : 기숙사
Fee : 기숙사 거주비
SID: 학생 아이디
GradeLevel : 학년

4-14 가 4-15 가 되는게 맞는지 생각해보자

그림 4-16

DK/NF의 예 2

PROFESSOR (FID, Fname, Class, SID, Sname)
Constraints: FID → Fname
Fname → FID
FID → Class 1 SID
Fname → Class 1 SID
SID → FID
SID → Fname
SID → Sname
FID must start with 1; SID must not start with 1

그림 4-17

예 2의 도메인 키 정의

Domain Definitions:

Attribute	Domain Name	Values
FID	FacultyID	4 decimal digits, first digit is 1
Fname	PersonNames	Char(50)
Class	ClassNames	Char(10); values (list of valid course names)
SID	StudentID	4 decimal digits, first digit is not 1
Sname	PersonNames	Char(50)

Relation and Key Definitions:

FACULTY (FID, Fname)
PREPARATION (Fname, Class)
STUDENT (SID, Sname, Fname)

Candidate key: Fname

faculty : 교수 정보
preparation : 교수-학급
student : 학생 정보

그림 4-18

DK/NF의 예 3

STU-ADVISER (SID, Sname, FID, Fname, GradFacultyStatus)
Key: SID
Constraints: FID → Fname
Fname → FID
FID and Fname → GradFacultyStatus
Only graduate faculty can advise graduate students
FID begins with 1
SID of graduate student begins with 9
GradFacultyStatus = 0 for undergraduate faculty
1 for graduate faculty

GradFacultyStatus : 학부/대학원 지도교수 속성값

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

그림 4-19

예 3의 도메인 키 정의

Domain Definitions:

Attribute	Domain Name	Values
FID	FacultyID	4 decimal digits, first digit is 1
Fname	PersonNames	Char(50)
GradFacultyStatus	FacultyStatus	Values {0, 1}
GSID	GradStudentID	4 decimal digits, first digit is 9
UGSID	UnderGradStudentID	4 decimal digits, first digit is not 1 and not 9
Sname	PersonNames	Char(50)
Gfname	PersonNames	Values {FACULTY Fname where GradFacultyStatus = 1 }

Relation and Key Definitions:

FACULTY (FID, Fname, GradFacultyStatus)
G_ADV (GSID, Sname, Gfname)
UG_ADV (UGSID, Sname, Fname)

Candidate key: Fname

그림 4-21

세 가지 유형의 속성 관계의 요약

속성 관계의 유형

	일 대 일	다 대 일	다 대 다
릴레이션 정의	R(A,B)	S(C,D)	T(E,F)
종속성	A → B B → A	C → D D → C	E → F F → E
키	Either A or B	C	(E,F)
다른 속성 추가 규칙	Either A or B → C	C → E	(E,F) → G

* 이 릴레이션 정의에 사용된 문자들은 그림 4-22에서와 동일

4.8.1 부자연스러운 정규화

다음의 릴레이션을 고려해보자.

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

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)

이 두 릴레이션에는 CUSTOMER.Zip의 값이 CODES.Zip의 값으로 존재해야 하는 참조 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 고객의 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 낸다면 역정규화가 더 좋기도 하다

ex1) 고객 정보에 거주지와 관련된 정보를 저장. 행정구역이 바뀌는 경우 기존 주소를 업데이트 해야 하는가/아닌가?

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