

2023 - 03 - 21 날씨 ☀

관계모델

관계모델의 3단계

1. 개념단계
 - 개념스키마 : 현실세계의 객체를 식별
2. 논리단계
3. 물리단계

릴레이션

- 관계 모델의 데이터 표현
- 구성
 - 릴레이션 스키마
 - 릴레이션 인스턴스

릴레이션 스키마

- 데이터 베이스의 구조와 제약조건에 관한 전반적인 명세
- 개체, 속성, 관계, 제약조건을 기술한 것
- 릴레이션 스키마 명세
 - 릴레이션 이름
 - 필드 또는 열 또는 속성의 이름
 - 각 속성의 도메인 이름

학생(학번(속성이름):문자열(도메인 이름), 이름 : 문자열, 로그인: 문자열, 나이: 정수, 평점 평균 : 실수)

TIP 릴레이션 스키마

- 🌈 릴레이션을 정의 하는 청사진

릴레이션 인스턴스

- 레코드라고도 불리는 튜플의 집합
- 튜플
 - 릴레이션 스키마에서 정의된 각 속성(필드)로 구성되며, 하나의 데이터 묶음을 나타냄
 - 릴레이션 인스턴스에서 각 튜플을 하나의 행으로 구성
- 릴레이션 인스턴스의 튜플 수 : 카디널리티
- 릴레이션 인스턴스의 속성 수 : 차수
- 인스턴스의 튜플 순서는 중요하지 않음

도메인

- 하나의 속성이 가질 수 있는 동일한 유형의 원자값들의 집합
- 릴레이션 스키마는 릴레이션 인스턴스의 각 필드의 도메인을 명세(도메인 제약조건)
- 프로그래밍 언어 관점에서 필드의 데이터 타입을 의미
 - $R(f_1:D_1, \dots, f_n:D_n)$ 을 릴레이션 스키마라고 하고, 각 $f_i, 1 \leq i \leq n$ 에 대한 이름이 D_i 인 도메인에 대한 집합을 Dom_i 라고 할 때, 도메인 제약조건을 만족하는 R 의 인스턴스는 다음과 같이 n 개의 필드를 가진 튜플의 한 집합이 됨

$$\{ \langle f_1: d_1, \dots, f_n: d_n \rangle \mid d_1 \in Dom_1, \dots, d_n \in Dom_n \}$$

- domain tip
 - 사용자가 도메인을 직접 만들수 있다.

관계 데이터베이스

- 관계 데이터베이스 : 서로 다른 릴레이션 이름을 가진 릴레이션들의 모임
- 관계 데이터베이스 스키마 : 데이터베이스에 속한 릴레이션의 스키마들의 집합

관계형 데이터베이스의 무결성 제약조건

무결성 제약조건

[참고1](#), [참고2](#), [참고3 SQL로 제약조건](#)

- 저장된 정보의 품질에 따라 데이터 베이스의 품질이 결정
- DBMS에 부정확한 정보가 입력되는걸 막기 위한 수단
- 무결성 제약조건
 - 스키마에 명세된 조건
 - 인스턴스에 저장될 수 있는 데이터를 제한
- 적법한 인스턴스
 - 데이터 베이스 스키마에 명세된 무결성 제약조건들을 모두 만족하는 데이터베이스 인스턴스
- dbms는 무결성 제약조건을 집행해 적합하게 만듦

1. 개체 무결성

- 한 개체(튜플)은 다른 개체들과 명확히 구분되어야함.
- 모든 릴레이션에 속한 필드들의 최소 부분집합이 각 튜플에 대해 고유 식별자가 되어야함
- 키 제약 조건으로 개체 무결성 유지
 - 릴레이션에 속한 필드들의 최소 부분 집합이 각 튜플에 고유한 식별자가 되어야함.
 - 키 제약 조건에서 튜플을 식별하는 부분집합을 키(key)라고 부름.
- 키
 - 키는 유일해야함
 - 키를 구성하는 필드 집합의 어떠한 부분집합도 튜플에 대해 유일한 식별자가 될 수없음(키를 구성하는 속성이 여러개라면 그중 일부만으론 식별자가 될 수 없다.)
- 수퍼키
 - 유일성을 만족하는, 필드들의 부분집합
- 후보키
 - 유일성과 최소성을 만족하는 수퍼키 집합의 부분집합
- 기본키

- 특정 투플을 구별하기 위해 후보 키에서 선택된 고유한 식별자

학번, 전화번호, 이메일 등등

수퍼키를 기준으로 하면, 데이터 삽입 삭제에 어려움이 생길수 도 있다.
키는 요구사항에 따라 달라질수도 있다.

후보키는 유일성과 최소성을 만족할때 기본키를 고르는 법!

- 관계를 고려하여 기본키를 선택한다.
- 릴레이션에 적합한 의미를 가진 키를 고른다
- 키 길이가 짧고 간결
- 키의 사용빈도로 파악한다. 예를 들어 조인 할때

❌ 불필요한 정보는 사회적 문제를 일으킬 수 있고 공간 낭비를 줌

2. 참조 무결성

- 한 릴레이션에 저장된 정보와 다른 릴레이션에 저장된 정보가 연결되는 경우가 대부분
- 참조 관계에 있는 두 릴레이션의 데이터는 항상 일관된 값으로 유지되어야함
 - 한 릴레이션의 데이터가 수정되면 참조관계에 있는 데이터 역시 수정되어야함
 - 한 릴레이션의 데이터가 삭제되면 참조관계에 있는 데이터 역시 삭제되어야함
- 외래 키 제약조건으로 참조 무결성 유지
 - 한 릴레이션의 필드가 다른 릴레이션의 키를 참조
 - 참조하는 릴레이션의 데이터 변경에 따른 무결성을 유지
- 외래 키
 - 한 릴레이션의 키 중에서 다른 릴레이션의 투플을 유일하게 식별 할 수 있는 키
 - 한 릴레이션의 투플에서 다른 릴레이션의 투플을 참조하기 위해 사용

릴레이션: Category

CategoryNo	CategoryName
1	Novel
3	History

릴레이션: Product

ProductNo	ModelNumber	ProductName	Price	CategoryNo
20101927	(null)	The Second World War	37800	3
97422537	(null)	Hobbit	28800	1
97422515	(null)	Lord of the Rings 1	28800	1

참조하는 릴레이션의 외래키 값은 널이 들어올 수 있다.

예를 들어 신입사원이라 부서가 정해지지 않은경우 부서번호는 널 값이어도 된다.

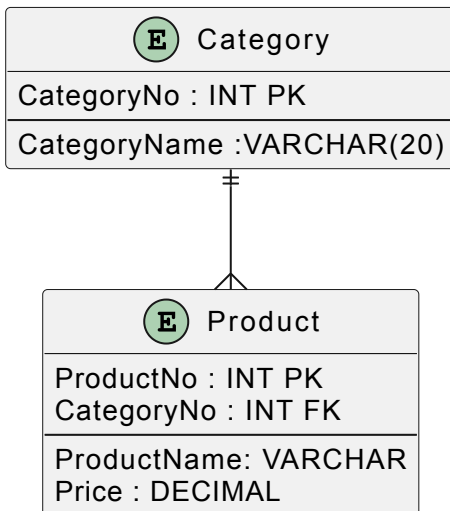
존재하는 부서중에선 참조관계를 변경해도 되지만, 없는 부서를 참조할 경우 외래키 제약조건에 위반한다.

3. 도메인 무결성

- 릴레이션의 각 속성은 필드에 일관된 데이터가 저장되어야함.
- 속성에 지정된 형식이 아닌 다른 형식의 데이터는 포함 될 수 없어야함.
- 도메인 제약조건으로 도메인 무결성 유지
- 릴레이션의 각 속성에 포함되는 데이터의 형식을 제약
 - 각 속성의 값은 반드시 원자값
 - 속성에 지정된 도메인의 범위 내에 해당되는 값
 - 속성의 기본 값과 널포함 가능 여부등에 대한 제약조건

무결성 제약조건 집행

- 제약조건은 릴레이션 스키마에 정의되고 릴레이션의 수정될 때 집행됨
- 도메인 무결성과 개체 무결성은 직관적으로 즉시 집행됨
 - 제약조건을 위배하는 삽입, 삭제, 갱신 명령은 즉시 거부
- 참조 무결성 집행 단계
 - 참조하는 릴레이션에 없는 값에 대한 삽입이 시도되는 경우
 - 명령이 거부됨.
 - 참조 릴레이션의 데이터가 삭제되는 경우
 - 외래 키가 존재하는 릴레이션의 참조 데이터를 모두 삭제
 - 참조 릴레이션의 데이터를 삭제할 수 없도록 명령을 거부
 - 외래 키가 존재하는 릴레이션의 해당 데이터를 모두 다른 값으로 갱신
 - 참조하는 릴레이션의 데이터가 변경되는 경우
 - 세가지 방법중 하나를 선택



거부 하는 경우

1. 카테고리 릴레이션에 없는 카테고리이름을 가진 상품투플이 삽입될 경우
2. 카테고리 릴레이션의 투플이 삭제되는 경우
 - 삭제될 카테고리 투플을 참조하는 모든 상품 투플을 삭제
 - 삭제될 카테고리 릴레이션을 참조하는 상품 투플이 있는 경우, 카테고리 릴레이션의 투플 삭제를 거부
 - 삭제될 카테고리 릴레이션을 참조하는 모든 상품 투플의 데이터를 모두 다른 값으로 갱신합니다.
3. 카테고리 릴레이션의 투플이 변경되는 경우
 - 변경될 카테고리 투플을 참조하는 모든 상품 투플을 삭제
 - 변경될 카테고리 릴레이션을 참조하는 상품 투플이 있는 경우, 카테고리 릴레이션의 투플 삭제를 거부
 - 변경될 카테고리 릴레이션을 참조하는 모든 상품 투플의 데이터를 모두 변경되는 값으로 갱신

SQL

- 구조적 쿼리 언어
- 관계형 데이터 베이스 관리 시스템의 데이터를 관리하기 위해 설계
- 특수 목적의 대화식 프로그래밍 언어
 - 캠퍼레인과 보이스가 개발
 - 오라클, infomix, Sybase, IBM등 자사의 관리 시스템에 맞는 SQL을 만들어 적용
 - 사용은 비슷하지만 내부 동작이 다름

제약조건 추가하기

```
ALTER TABLE Category ADD CONSTRAINT pk_category primary key(CategoryNo);  
ALTER TABLE Category //테이블 변경  
ADD CONSTRAINT pk_category //제약조건을 추가 pk_category는 제약조건 이름  
PRIMARY KEY(CategoryNo); // 카테고리번호를 기본키로 지정하겠다.
```

UPDATE의 조건문은 PRIMARY KEY로 지정해주는것이 안전

TIP

필요하지만 거의 쓰이지 않는 데이터는 새로운 테이블로 분리
스키마가 객체
튜플이 객체
인스턴스는 여러개의 튜플들이 들어있음