

3. 관계형 모델

1. 관계형 데이터베이스 구조[64]

들어가기 전에

관계형 모델(relational model)

- 술어 논리(predicate logic)와 집합론(set theory)가 이론적 배경

ER 모델

- 개체 집합과 개체 집합을 연결하는 관계 집합으로 현실세계의 업무를 추상화

관계형 모델

- 테이블, 컬럼, 기본키-외래키 참조 관계 등으로 현실 관계를 표현

1. 개요

표(table)

- 2차원 데이터 구조

열(column)

- 열에 저장되는 정보의 의미를 담고 있는 열 이름 기록

행(row)

- 열의 순서에 맞춰 일련의 데이터 집합을 입력

2. 릴레이션

관계형 모델(relation model)

- 행과 열의 관계로 이루어진 격자 구조의 수학적 개념 릴레이션(relation)

컬럼(column), 필드(field)

- 컬럼에 해당하는 값의 집합(열, 필드, 속성)

레코드(record), 튜플(tuple)

- 각 컬럼의 순서에 맞게 나열된 값의 집합(행, 튜플)

도메인(domain)

- 컬럼이 가질 수 있는 값의 범위

차수(degree)

- 릴레이션에 존재하는 속성의 개수

카디널리티(cardinality)

- 릴레이션에 존재하는 레코드의 개수

릴레이션의 특징

레코드의 유일성

- 하나의 릴레이션에는 중복되는 레코드가 존재할 수 없으며, 하나의 키 값으로 하나의 레코드를 유일하게 식별한다.

레코드의 무순서성

- 한 릴레이션에 포함된 레코드는 순서가 정해져 있지 않다.

컬럼의 무순서성

- 한 릴레이션을 구성하는 컬럼 사이에는 순서가 없으며, 이름과 값의 쌍으로 구성된다.

컬럼값의 원자성

- 컬럼은 여러 의미를 갖는 값으로 분해가 불가능하며 원자적(atomic)하다

키(key)와 키의 성질

키(key)

- 그 릴레이션의 레코드를 유일하게 식별하는 역할을 하는 컬럼의 부분집합

키의 성질

유일성(uniqueness)

- 키 컬럼의 값은 중복될 수 없다.

최소성(irreducibility)

- 2개 이상의 컬럼으로 구성된 키에서 어느 한 컬럼을 제거하면 유일한 식별성이 파괴된다.

- 키는 유일성을 유지하기 위한 최소의 컬럼으로만 구성된다.

- 후보키와 관련

키에 따른 컬럼 분류

- 어떤 릴레이션 내의 모든 컬럼을 묶어서 키로 설정하면 '유일성'을 만족하기 때문에 모든 레코드가 적어도 하나의 키를 갖고 있다는 사실은 명백하다.

- 이 때 '최소성'을 갖고 있는 부분집합을 찾는 것이 중요하다.
 - 주속성(prime attribute)
 - 릴레이션 R의 키 중 적어도 한 키에 포함되는 컬럼
 - 비주속성(non-prime attribute)
 - 주속성에 포함되지 않는 속성
 - 키의 분류
 - 수퍼키(superkey)
 - 한 릴레이션 내부에는 레코드를 고유하게 구별할 수 있는 컬럼 집합이 여러 개 있을 수 있음. 이러한 컬럼의 집합을 수퍼키(superkey)라고 한다.
 - 후보키(candidate key)
 - 수퍼키 중에서도 최소한의 컬럼으로 구성된 수퍼키
 - 최소성 위배
 - {학과이름}과 {단과대학, 주소}는 수퍼키이다. 하지만 {단과대학, 주소}의 경우 {주소}가 이미 후보키이기 때문에 최소성 위배로 후보키가 될 수 없다.
 - 기본키(PK: Primary Key)
 - 한 릴레이션에서 여러 후보키가 존재하며, 데이터베이스 설계자가 레코드를 구분하기 위해 지정한 후보키 중 하나를 기본키(PK: Primary Key)라고 한다.
 - 개체 무결성 제약조건
 - 키는 널(NULL) 값을 가질 수 없다. 그러나 외래키나 일반 컬럼은 널 값을 가질 수 있다.
1. 관계형 데이터베이스 구조(2) [68]
- 3. 릴레이션 스키마
 - 릴레이션 스키마(schema)
 - 어떠한 테이블에서 사용되는 컬럼과 컬럼이 지니는 데이터 타입(data type)을 정의한 것
 - 테이블의 컬럼과 컬럼에 저장될 데이터의 유형을 정의하는 것과 유사
 - 릴레이션 인스턴스(instance)
 - 특정 시점에 스키마에 따라 저장된 실제 튜플(tuple)들의 집합
 - 튜플의 묶음으로, 테이블의 행(row)들에 해당
 - 분리되어 있는 서로 다른 릴레이션 간의 관련성이 데이터의 복사를 통해 이루어짐 (FK)
 - 4. 관계형 모델의 제약조건
 - 제약조건(constraints)
 - 릴레이션의 모든 인스턴스들이 만족해야 하는 조건으로, DBMS는 여러 유형의 제약조건을 스키마에 기술할 수 있는 기능을 제공한다.
 - 유형
 - 영역 제약조건(domain constraints)
 - 특정 컬럼이 가질 수 있는 값에 대한 제약조건, 컬럼에 해당되는 값이 사전에 정의된 가능한 값의 범위인 영역(domain)에 포함되며 원자적(atomic)이어야 하는 조건
 - 키 제약조건(key constraints)
 - 릴레이션에 저장된 각 레코드를 구별하기 위해 특정 컬럼의 값들이 릴레이션 내부의 레코드를 고유하게 식별할 수 있어야 한다는 조건
 - 적어도 하나의 키가 모든 레코드를 고유하게 식별
 - 개체 무결성 제약조건(entity integrity constraints)
 - 기본키 값으로 널(NULL) 값을 저장할 수 없다
 - 두 개 이상의 기본키 값이 널이면 그들을 서로 구별할 수 없어 무결성이 훼손될 뿐만 아니라, 기본키를 추후에 입력한다고 하더라도 레코드를 특정지어 수정을 지시할 수 없어 무결성을 회복할 수 있는 방법도 없다.
 - 참조 무결성 제약조건(referential integrity constraints)
 - 한 릴레이션에 있는 레코드가 다른 릴레이션에 있는 레코드를 참조하려면 반드시 참조되는 레코드가 해당 릴레이션 내에 있어야 하는 제약조건을 명시
 - 외래키(FK: Foreign Key)
 - 두 릴레이션의 스키마 R_1 , R_2 사이의 다음 두 조건을 만족
 - ① R_1 의 외래키의 컬럼들은 R_2 의 기본키(PK)의 속성들과 동일한 영역을 가진다.
 - R_1 의 외래키는 R_2 의 기본키 값을 사용
 - ② R_1 의 한 레코드의 외래키(FK)는 R_2 의 어떤 컬럼의 기본키(PK)와 일치하거나 널 값을 가져야 한다.

외래키가 NULL이라는 의미는 해당 레코드가 R2의 어떤 레코드와도 연관성이 없다는 것

참조 무결성을 만족하기 위해 수강 릴레이션에서 참조하는 학생과 과목 릴레이션의 레코드는 반드시 존재해야 하며, 만약 학생 릴레이션이나 과목 릴레이션의 레코드가 삭제되는 경우에는 삭제된 기본키값을 참조하고 있는 수강 릴레이션의 레코드도 같이 삭제되어야 한다.

2. 논리적 데이터 모델링[72]

개요

ER 모델은 데이터의 구조와 데이터 간의 관계를 정의한 개념적 데이터 모델이다

상용 DBMS에 데이터베이스를 구축할 때 개념적 데이터 모델을 바로 적용할 수 없기 때문에 개념적 데이터 모델을 상용 DBMS에 맞는 논리적 데이터 모델로 변환하는 작업이 필요하다.

1. ER 다이어그램의 관계형 모델 변환

관계형 모델은 데이터베이스를 릴레이션(relation)의 집합으로 구성하는 것이다. 따라서 ERD(ER 다이어그램)에 포함된 개체 집합과 관계 집합 등 표현된 모든 내용을 관계형 모델의 릴레이션으로 변환하는 작업이 필요하다.

속성

컬럼

개체 집합

릴레이션

개체

레코드

ERD의 모든 표현은 의미 훼손 없이 관계형 모델의 스키마로 변환되어야 한다.

단계[73]

단계 1. 개체 집합: 개체 집합은 릴레이션으로 변환된다. 개체 집합의 키 속성 중 하나를 릴레이션의 기본키로 지정한다.

단계 2. 약한 개체 집합: 약한 개체 집합이 독자적으로 존재할 수 없는 개체 집합을 의미하며, 약한 개체 집합을 릴레이션으로 변환할 경우 지배 개체인 강한 개체 집합의 키 속성을 포함시킨다. 강한 개체란 다른 개체의 도움 없이 독자적으로 존재할 수 있는 개체이며, 변환된 릴레이션의 기본키는 강한 개체 집합의 키와 약한 개체 집합의 부분 키를 묶어서 지정한다.

단계 3. 일대일 관계: 일대일의 사상수를 가지는 이항 관계는, 관계 집합에 참여하는 두 릴레이션 중에서 어느 하나를 다른 릴레이션의 기본키를 참조하는 외래키 컬럼으로 변환한다.

단계 4. 일대다 혹은 다대일 관계: 일대다 이항관계는 '다' 쪽 릴레이션의 외래키 컬럼으로 변환하며, '일' 쪽의 기본키를 참조하도록 외래키를 지정한다.

단계 5. 다대다 관계: 다대다 이항관계는 별도의 릴레이션, 즉 관계 릴레이션을 별도로 생성하고, 관계에 참여하는 두 릴레이션의 기본키를 각각 참조하는 외래키로 복합키 형태의 컬럼을 구성한다.

단계 6. 다중값 속성: 기존 릴레이션의 기본키와 다중값 속성을 별도의 릴레이션으로 구성하고, 이 둘을 새로운 릴레이션의 기본키로 지정한다.

단계 7. 관계 집합의 속성: 각각의 속성을 외래키가 위치한 릴레이션의 컬럼으로 삽입한다.

단계 8. n항 관계:n개 이상의 개체 집합이 참여한 관계의 경우, 관계에 참여하는 n개의 릴레이션의 키들로 구성하는 관계 릴레이션으로 변환한다. 관계 릴레이션의 컬럼들은 참여 릴레이션의 기본키를 참조하는 외래키와 관계 컬럼으로 구성한다.

2. 관계형 모델 변환의 예[74-77]

3. 데이터 조작 [77]

관계형 모델은 스키마로부터 특정 데이터를 추출하는 데이터 연산 방법을 관계 대수(relational algebra)를 통해 제공함

1. 관계 대수의 이해

SQL은 선언적 언어

관계 대수는 질의에 대한 처리 과정을 단계적으로 표현할 수 있는 절차적(procedural) 언어

DBMS는 주어진 SQL에 대한 실행 계획 또는 처리 절차를 나타내기 위해 관계 대수를 DBMS 실행 엔진의 내부 언어로 사용함

사용자가 의도하는 데이터 결과를 얻기 위해 복수 개의 연산자를 적용하여 처리 과정을 절차적으로 표현할 수 있다.

폐쇄적 특성(closure property)

모두 1개 이상의 릴레이션을 입력으로 받아 새로운 릴레이션을 생성하는 폐쇄적 특성을 갖추고 있어 여러 연산자를 중첩하여 하나의 완전한 관계 대수식을 구성

2. 기본 연산자

항의 개수에 따른 분류

단항 연산자(unary operator)

한 개의 릴레이션을 대상으로 하는 연산자

선택, 프로젝트, 리네임 연산자

이항 연산자(binary operator)

두 개의 릴레이션을 필요로 함

합집합, 차집합, 카티션 프로덕트 연산자

(1) 선택(select)

- 릴레이션에서 조건(predicate)을 만족하는 레코드들을 선택
- 시그마(σ) 기호
- 조건은 아래 첨자로 표현

(2) 프로젝트(project)

- 입력된 릴레이션의 특정 컬럼만을 추출하여 새로운 릴레이션으로 재구성
- 그리스어의 대문자 파이(Π)
- 입력 릴레이션은 소괄호내부에 위치

(3) 집합 연산자

- 관계형 모델은 집합론(set theory)을 이론적 배경으로 하기 때문에 집합 연산자인 합집합, 교집합 그리고 차집합 연산자를 적용할 수 있다.

(다른 관계 대수 연산과 달리) 호환 가능한 릴레이션(compatible relation)간에만 적용할 수 있음

조건

- 릴레이션 R과 S는 가지고 있는 컬럼의 수가 같아야 한다.
- 모든 i에 대해 릴레이션 R의 i번째 컬럼의 도메인과 릴레이션 S의 i번째 컬럼의 도메인이 서로 같아야 한다.

1) 합집합 연산자

- 두 릴레이션이 포함된 모든 레코드를 갖는 릴레이션을 결과값으로 가진다.

2) 차집합 연산자

- 한 릴레이션에는 포함되지만 다른 릴레이션에는 포함되지 않는 레코드를 가지는 릴레이션을 결과로 반환하는 연산자
- 연산 기호 '-'

3) 교집합 연산자

- 기호 \cap
- 양쪽 릴레이션에 동시에 포함되는 레코드의 집합을 생성

(4) 리네임(rename) 연산자

- 소문자 ρ (rho)로 표기
- 관계 대수식에 새로운 이름을 부여하는 연산자
- 계 대수식 R의 결과에 n개의 컬럼이 있다면, - 왼쪽 식: $\rho_x(R)$ - R의 결과 릴레이션에 x라는 이름만 부여 - 오른쪽 식: $\rho_{x(A_1, A_2, \dots, A_n)}(R)$ - x가 보유한 n개의 컬럼에 대해 각각 A_1, A_2, \dots, A_n 이라는 새로운 이름을 부여

(5) 카티션 프로덕트(Cartesian product)

- 카티션 프로덕트 연산자는 \times 로 표시
- 두 릴레이션의 레코드 간에 모든 조합을 취하여 결합한 레코드를 생산하는 연산자이다. 릴레이션 R과 S에 대해 카티션 프로덕트 연산자는 아래와 같이 정의된다.
- 만약 R에 m개의 컬럼과 a개의 레코드, S에 n개의 컬럼과 b개의 레코드가 존재한다면 R과 S의 카티션 프로덕트 연산의 결과로 m+n개의 컬럼과 a×b개의 레코드가 저장된 릴레이션이 만들어진다.

(6) 조인(join)

- 가능한 모든 레코드 조합을 결과에 포함시키는 카티션 프로덕트 연산과 유사하지만, 조인 연산은 특정 조건을 만족하는 레코드 조합만 결과에 포함시킨다.
- 조인 조건(join condition)을 만족하는 레코드 조합만 결과에 포함시킨다.
- 세타 조인(θ)
 - 관계 연산자 θ 로 정의되는 조인 조건을 가진다. A라는 속성을 갖는 릴레이션 R과 B라는 속성을 갖는 릴레이션 S에 대해 세타 조인($\bowtie_{A \theta B}$)은 다음과 같이 정의된다.
 - 두 릴레이션을 결합할 때 단순 동등 비교(equi-join)뿐 아니라 $<, >, \leq, \geq, \neq$ 등 임의의 비교 연산자(θ)를 조건으로 사용할 수 있는 일반화된 조인 연산
- 동등 조인(이퀄 조인, equi join)
 - θ 이 '='인 경우 특별히 동등 조인이라 한다.

3. 데이터 조작(2) [89]

3. 확장 연산자

- 확장 연산자는 질의 표현 시 자주 이용되는 기본 연산자들의 조합을 압축적으로 표현하는 연산자
- 기본 연산자가 표현할 수 없는 질의를 표현하기 위한 연산자가 포함된다.
- 자연 조인 연산자, 할당 연산자, 외부 조인(왼쪽, 오른쪽, 완전) 연산자 및 집계 연산자가 있다.

(1) 자연 조인 연산자

자연 조인(\bowtie)

- 동등 조인을 보다 간결하게 표현할 수 있는 연산자

연산 과정

- ① R과 S에 대한 카티션 프로덕트 생성
- ② R과 S에 동일한 이름의 컬럼의 값이 같은 레코드만 선택
- ③ 중복되는 컬럼을 한 번만 표현한 결과를 반환

예제

(2) 할당 연산자(\leftarrow)

- 할당 연산자는 \leftarrow 나타내며, 관계 대수식의 결과를 임시적으로 릴레이션 변수에 저장시키는 연산자이다

- 왼쪽 릴레이션 변수에 임시적으로 저장시킬 뿐, 특별히 릴레이션을 변화시키지는 않음

(3) 집계 함수 연산 ($g(\text{caligraph } G)$)

- $g(\text{caligraph } G)$ 로 표기되는 집계 함수(aggregate function)는 집계를 통하여 얻은 수치를 릴레이션에 포함시키는 역할을 한다.

집계함수 종류

sum(summation)	값의 총합을 반환	avg(average)	값의 평균값을 반환	count	집합을
구성하는 원소들의 개수를 반환		max(maximum)	집합 내 가장 큰 값을 반환	min(minimum)	집합 내 가장 작은 값을 반환

그룹 기준에 따른 분류

- 그룹 기준이 없는 집계 연산

- 전체 릴레이션에 포함된 모든 레코드를 하나의 그룹으로 인식하고 집계 함수를 적용

- 그룹 기준이 있는 집계 연산

- 기준이 되는 컬럼의 값이 같은 레코드를 그룹화하고 각 그룹별로 집계 함수를 적용

구문 형식과 예제

- 계산에 앞서 중복된 값을 제거해야 하는 경우

- "-distinct" 키워드 추가

그룹화할 속성이 여럿일 때

- 일반적으로 관계 대수식 R에 대해 그룹화할 속성 G_1, G_2, \dots, G_n 과 속성 A_i 에 적용할 함수 $F_i(1 \leq i \leq m)$ 에 대한 집계 함수 g 는 다음과 같이 기술한다.

- $G_1, G_2, \dots, G_n, g, F_1(A_1), F_2(A_2), \dots, F_m(A_m)(R)$

그룹화하는 방식

- ① 그룹 안에 있는 모든 레코드는 그룹화 속성 G_1, G_2, \dots, G_n 에 대하여 같은 값을 가진 레코드들이어야 한다.
- ② 서로 다른 그룹 안에 존재하는 레코드들은 대상 컬럼 G_1, G_2, \dots, G_n 에 대한 값이 하나 이상 달라야 한다.
- ③ 만약 G_1, G_2, \dots, G_n 이 명시되지 않은 경우, 모든 레코드를 포함하는 단 하나의 그룹이 생성된다.