

6. 정규형

1. 좋은 릴레이션과 나쁜 릴레이션 [212]

개요

데이터베이스 모델링

개념적 데이터 모델링

- ER 모델을 이용하여 데이터의 구조와 데이터 간의 관계를 정의하여 현실 세계의 업무 표현

논리적 데이터 모델링

- 개념적 데이터 모델링에서 도출된 ERD를 특정 DBMS에서 사용하는 구현 모델에 맞게 변환

정규화가 안된 릴레이션의 문제

- 데이터 중복 저장으로 데이터베이스의 저장 공간을 낭비

- 삭제, 갱신 등의 과정에서 이상(anomaly) 현상이 발생할 수 있음

정규화(normalization)

- 관계형 모델에서 논리 스키마를 효과적으로 모델링

- 릴레이션의 정규형(normal form)을 분석함으로써 릴레이션의 스키마가 얼마나 효율적으로 실세계를 반영하고 있는지를 평가

효용성

- 한 릴레이션 내의 컬럼들 간 관계 표현

- 불필요한 데이터의 종속과 중복 제거

- 새로운 컬럼 추가 시 기존 컬럼과의 관계 수정을 최소화

장점

- 갱신 이상(update anomaly)의 발생 가능성 차단

- 정보 삽입, 삭제, 갱신 시의 불일치로 데이터 변동 상황에 따른 이상 현상

삽입 이상

- 릴레이션에 레코드를 추가할 때, 필요한 컬럼의 값이 없이는 테이블에 새로운 레코드를 삽입하지 못하는 경우

삭제 이상

- 키 값의 삭제 시 의도하지 않았던 전혀 다른 데이터가 삭제될 때 발생

- ex) NEW 등급 할인 삭제할 경우 윤봉길 고객까지 삭제

수정 이상

- 하나의 릴레이션이 불필요한 중복된 레코드들을 포함하여 데이터베이스의 일관성이 훼손

- ex) 8% 갱신하다가 중간에 실패하면 어떤 VIP 고객은 할인을 8%, 다른 VIP는 10%

2. 함수적 종속성 [216]

1. 함수적 종속성의 정의

함수적 종속성(functional dependency)

- 속성들 간의 값의 연관관계를 표현

- [정의 6-1] 임의의 릴레이션 스키마 R의 인스턴스에 포함되는 서로 다른 두 레코드 r1, r2와 속성(컬럼) 집합 X와 Y에 대해, $r1[X] = r2[X]$ 일 때, $r1[Y] = r2[Y]$ 이면 함수적 종속성 $X \rightarrow Y$ 가 성립한다.

- 특정 시점의 릴레이션 인스턴스를 분석하여 파악

- ex) 모든 레코드에 대해 등급의 속성값이 같다면 할인을 역시 같음

- {등급} \rightarrow {할인율}

- '등급이 할인율을 함수적으로 결정한다.'

- '할인율은 등급에 함수적으로 종속된다.'

- 결정자(determinant)

- {등급}

- 종속자(dependent)

- {할인율}

- 적법한 릴레이션(legal relation) 상태

- 적법하지 않은 릴레이션(illegal relation) 상태

- ex) 특정 회원에게만 VIP 할인을 다르게 적용하는 상태

2. 함수적 종속성의 추론 규칙과 카노니컬 커버

릴레이션의 클로저

- 명시적으로 주어진 함수적 종속성과 그로부터 추론될 수 있는 모든 함수적 종속성 집합

- ex) 함수적 종속성 {고객번호} \rightarrow {고객명}과 {고객명} \rightarrow {전화번호}가 성립하면 {고객번호} \rightarrow {전화번호}도 성립

F의 클로저(closure), F^*

릴레이션 스키마 R의 함수적 종속성 집합을 F라고 할 때, F로부터 유추할 수 있는 모든 종속성 집합

확장된 종속성 집합으로부터 논리적 모순, 오류를 배제하고 새로운 종속성을 찾아내기 위한 추론규칙

규칙 1. 재귀성 규칙: $X \supset Y$ 이면, $X \rightarrow Y$ 이다. 규칙 2. 부가성 규칙: $X \rightarrow Y$ 이면, $XZ \rightarrow YZ$ 이다. 규칙 3. 이행성 규칙: $X \rightarrow Y$ 이고 $Y \rightarrow Z$ 이면, $X \rightarrow Z$ 이다. 규칙 4. 분해 규칙: $X \rightarrow YZ$ 이면, $X \rightarrow Y$ 이다. 규칙 5. 합집합 규칙: $X \rightarrow Y$ 이고 $X \rightarrow Z$ 이면, $X \rightarrow YZ$ 이다. 규칙 6. 의사 이행성 규칙: $X \rightarrow Y$ 이고 $WY \rightarrow Z$ 이면, $WX \rightarrow Z$ 이다.

규칙 1, 2, 3은 암스트롱의 추론 규칙(Armstrong's inference rule)이며, 나머지 규칙은 암스트롱 추론 규칙으로부터 유도된 규칙

부가성 규칙의 예

카노니컬 커버(canonical cover)

클로저에는 자명한 종속성($A \rightarrow A$)과 중복된 종속성($A \rightarrow BC$ 와 $A \rightarrow B$, $A \rightarrow C$ 등)이 포함되는데, 불필요한 함수적 종속성을 제거하여 종속 정보에 대한 손실 없이 간소화된 함수적 종속성 집합

[정의 6-3] $F_c(F$ 의 카노니컬 커버)는 F^* 에 존재하는 모든 함수적 종속성을 커버할 수 있는 최소한의 함수적 종속성들로만 이루어진 집합이다.

커버(cover)

[정의 6-2] 주어진 함수적 종속성 집합 E에 대해 E가 함수적 종속성 집합 F^* (F의 클로저)에 포함되면 E의 모든 함수적 종속성들이 F로부터 추론될 수 있으며, 이때 F가 E를 커버(cover)한다고 표현한다.

조건

① F의 모든 함수적 종속성의 오른쪽 속성은 단일 속성이다.

② F에 포함된 종속성 $X \rightarrow A$ 의 결정자를 X의 진부분집합 Y를 사용한 $Y \rightarrow A$ 로 교체했을 때, 그 집합이 F와 동등한 집합이 될 수 있다.

③ F에서 어떤 함수적 종속성을 제거했을 때, 그 집합이 F와 동등한 집합이 될 수 없다.

3. 기본 정규형 [221]

개요

제1정규형에서 제5정규형으로 수준이 높아질수록 릴레이션의 구조화 조건이 더욱 까다로워지며 그 결과 데이터 중복이 최소화된다.

정규화의 목적

1. 어떠한 릴레이션이라도 데이터베이스 내에서 표현할 수 있도록 만든다. 2. 좀 더 간단한 관계 연산에 기초하여 검색 알고리즘을 효과적으로 만든다. 3. 릴레이션에서 바람직하지 않은 삽입, 갱신, 삭제 등의 이상을 제거한다. 4. 새로운 형태의 데이터가 삽입될 때 릴레이션을 재구성할 필요성을 줄인다.

제1정규형 (1NF: First Normal Form)

어떤 릴레이션 스키마에서 정의된 모든 속성의 도메인이 원자(atomic) 값을 가진다

정규형 만족

제2정규형 (2NF)

릴레이션이 제1정규형을 만족하고 기본키가 아닌 속성들이 기본키에 완전 함수 종속되면 제2정규형

기본키를 구성하는 속성들에 대하여 부분종속성을 제거

함수적 종속

부분 함수적 종속(Partial Functional Dependency)

속성 집합 Y가 속성 집합 X의 전체가 아닌 일부분에 함수적으로 종속

완전 함수적 종속(Full Functional Dependency)

속성 집합 Y가 속성 집합 X 전체에 함수적으로 종속

함수적 종속성 다이어그램(FDD, Functional Dependency Diagram)

예시

'도서관리자'는 기본키의 일부분인 '도크번호'에 부분 함수 종속됨

무손실 분해(lossless)

정보 손실이 없으며 조인 시 원래의 릴레이션으로 복원 가능한 분해

제3정규형

릴레이션이 제2정규형을 만족하고, 기본키가 아닌 속성들이 어떤 키에도 이행적으로 종속되지 않을 때 제3정규형

기본키가 아닌 속성이 이행적으로 종속되어 발생하는 중복을 제거

이행적(transitive) 종속성

$A \rightarrow B$, $B \rightarrow C$ 일 때 $A \rightarrow C$ 인 함수적 종속성

예시

{도크번호, 입항시간} \rightarrow {목적}, {목적} \rightarrow {담당 도선사}

- 무손실 분해

BC 정규형(BCNF: Boyce-Codd Normal Form)

- 릴레이션이 제3정규형을 만족하고 릴레이션에서 성립하는 ' $X \rightarrow Y$ ' 형태의 모든 함수적 종속성에 대하여 X 가 수퍼키이면 BC정규형이다.

- 수퍼키 : 최소성과는 무관한 고유키

- 예시

- $\{\text{도크번호, 입항시간}\} \rightarrow \{\text{목적}\}$ 과 $\{\text{도크번호, 입항시간}\} \rightarrow \{\text{출항시간}\}$ 의 결정자가 모두 수퍼키

- BUT** $\{\text{목적}\} \rightarrow \{\text{도크번호}\}$ 에 대해 $\{\text{목적}\}$ 은 수퍼키가 아님.

- 기본키가 아닌 속성 목적이 도크번호를 결정한다.

- 모든 함수적 종속성들의 결정자가 수퍼키가 되도록 만들어야 한다.

- 도크번호가 분해 대상이며, 이 경우 기존의 기본키를 유지할 수 없어 기본키를 변경해야 한다.

- 목적 속성이 도크번호 속성을 결정하기 때문에 기존 릴레이션에서 $\{\text{도크번호, 입항시간}\}$ 대신에 $\{\text{목적, 입항시간}\}$ 을 기본키로 지정한다.

- 그 이후 기존 릴레이션을 분해하여 목적과 도크번호 속성으로 이루어진 새로운 릴레이션을 생성한다. 새로운 릴레이션의 기본키는 함수적 종속성에 의해 목적 속성이 기본키가 된다. (무손실 분해)

- 최종 FDD

- [그림 6-20]의 FD 다이어그램으로부터 도크 스케줄 릴레이션의 유일한 함수적 종속성 $\{\text{목적, 입항시간}\} \rightarrow \{\text{출항시간}\}$ 의 결정자가 수퍼키이며, 도크 릴레이션에서도 함수적 종속성 $\{\text{목적}\} \rightarrow \{\text{도크번호}\}$ 의 결정자가 수퍼키이므로 BC 정규형 조건을 만족한다.

- 정규화 정리

4. 역정규화

- 개요

- 정규화에 의해 연관된 모든 릴레이션에 각각 접근하는 디스크 접근 비용 및 테이블 간의 조인 비용이 추가적으로 발생하여, 질의 응답(query response)의 속도 지연으로 이어질 수 있음

- 역정규화(denormalizing)

- 정규화의 반대 과정으로 정규화를 통해 분리되었던 릴레이션을 통합하는 구조의 재조정을 통하여 정보의 부분적 중복을 허용하지만 데이터 접근 성능을 개선

- 역정규화의 방법

- 1. 릴레이션의 병합

- 1:N 관계를 갖는 두 릴레이션의 경우 N쪽의 모든 데이터를 1에 통합하여 저장

- 질의 처리 시 빈번하게 읽어 들이는 속성만 별도로 릴레이션에 병합하여 저장

- 2. 릴레이션의 수직 분할

- 특정 릴레이션 내에서 빈번하게 접근하는 속성과 접근이 일어나지 않는 속성을 두 릴레이션으로 나누어 저장

- 3. 릴레이션의 수평 분할

- 속성이 아닌 레코드를 기준으로 분할

- ex) 학생 레코드를 학과별로 수평분할하여 물리적으로 저장

- 4. 유도 속성의 추가

- '생년월일'만 관리하는 릴레이션에 현재 시간을 이용하여 '나이' 속성을 도출

- 역정규화의 장단점

- 읽기 성능(SELECT) 향상

- 중복된 데이터 저장을 위한 추가 공간의 요구

- 중복된 데이터의 일관성을 유지하기 위해 쓰기 성능(I, U, D) 속도는 저하될 수 있음