

데이터베이스

휴먼지능정보공학전공

학습 목차

- 데이터베이스 기초 이론: 데이터베이스, DBMS, 데이터베이스 시스템
- 데이터 모델과 연산: 데이터 모델링의 개념과 데이터 모델의 역할, 관계 데이터 모델, 관계 데이터의 주요 연산
- 데이터베이스 언어 SQL: SQL의 주요 기능, 테이블 생성과 데이터 검색 및 조작을 위해 SQL로 질의문

학습 목차

- 관계 데이터 연산
- SQL실습기초

관계 데이터 연산

- 데이터 모델 = 데이터 구조 + 연산 + 제약조건

관계 데이터 연산

- 관계 데이터 연산(relational data operation)
 - 관계 데이터 모델의 연산
 - 원하는 데이터를 얻기 위해 릴레이션에 필요한 처리 요구를 수행하는 것
 - 관계 대수와 관계 해석
 - 기능과 표현력 측면에서 동등한 능력

관계 데이터 연산

- 관계 대수와 관계 해석의 역할
 - 데이터 언어의 유용성을 검증하는 기준
 - 관계 대수나 관계 해석으로 기술할 수 있는 모든 질의를 기술할 수 있는

데이터 언어 → 관계적으로 완전(relationally complete)하다고 판단

- 질의(query) : 데이터에 대한 처리 요구

관계 데이터 연산

- 관계 대수(relational algebra)의 개념
 - 절차 언어(procedural language)
 - 원하는 결과를 얻기 위해 릴레이션의 처리 과정을 순서대로 기술하는 언어
 - 릴레이션을 처리하는 연산자들의 모임
 - 대표 연산자 8개
 - 일반 집합 연산자와 순수 관계 연산자로 분류
- 폐쇄 특성(closure property)
 - 피연산자도 릴레이션이고 연산의 결과도 릴레이션

관계 데이터 연산

- 관계 대수 연산자
 - 일반 집합 연산자
 - 합집합 , 교집합, 차집합, 카티션프로덕트
 - 순수 관계 연산자
 - 선택, 프로젝트, 조인, 디비전

관계 데이터 연산

- 일반 집합 연산자(set operation)
 - 릴레이션이 튜플의 집합이라는 개념을 이용하는 연산자
 - 합집합: 릴레이션 A와 B의 합집합 반환
 - 교집합: 릴레이션 A와 B의 교집합 반환
 - 차집합: 릴레이션 A와 B의 차집합 반환
 - 카티션프로젝트: 릴레이션 A의 각 튜플과 릴레이션 Sdml 각 튜플을 모두 연결하여 만든 새로운 튜플을 반환

관계 데이터 연산

- 일반 집합 연산자의 특성
 - 피연산자가 2개 필요
 - 2개의 릴레이션을 대상으로 연산 수행
 - 합집합, 교집합, 차집합은 피연산자인 두 릴레이션이 합병 가능해야 함
 - 합병 가능(union-compatible) 조건
 - 두 릴레이션의 차수가 같아야 함
 - 두 릴레이션에서 서로 대응되는 속성의 도메인이 같아야 함

관계 데이터 연산

- 일반 집합 연산자 – 합집합(union)
 - 합병 가능한 두 릴레이션 R과 S의 합집합 : $R \cup S$
 - 릴레이션 R에 속하거나 릴레이션 S에 속하는 모든 튜플로 결과 릴레이션 구성
 - 결과 릴레이션의 특성
 - 차수는 릴레이션 R과 S의 차수와 같음
 - 카디널리티는 릴레이션 R과 S의 카디널리티를 더한 것과 같거나 적어짐
 - 교환적 특징이 있음
 - $R \cup S = S \cup R$
 - 결합적 특징이 있음
 - $(R \cup S) \cup T = R \cup (S \cup T)$

관계 데이터 연산

- 일반 집합 연산자 – 교집합(intersection)
 - 합병 가능한 두 릴레이션 R과 S의 교집합 : $R \cap S$
 - 릴레이션 R과 S에 공통으로 속하는 튜플로 결과 릴레이션 구성
 - 결과 릴레이션의 특성
 - 차수는 릴레이션 R과 S의 차수와 같음
 - 카디널리티는 릴레이션 R과 S의 어떤 카디널리티보다 크지 않음(같거나 적음)
 - 교환적 특징이 있음
 - $R \cap S = S \cap R$
 - 결합적 특징이 있음
 - $(R \cap S) \cap T = R \cap (S \cap T)$

관계 데이터 연산

- 일반 집합 연산자 – 차집합(difference)
 - 합병 가능한 두 릴레이션 R과 S의 차집합 : $R-S$
 - 릴레이션 R에는 존재하지만 릴레이션 S에는 존재하지 않는 튜플로 결과 릴레이션 구성
 - 결과 릴레이션의 특성
 - 차수는 릴레이션 R과 S의 차수와 같음
 - $R-S$ 의 카디널리티는 릴레이션 R의 카디널리티와 같거나 적음
 - $S-R$ 의 카디널리티는 릴레이션 S의 카디널리티와 같거나 적음
- 교환적, 결합적 특징이 없음

관계 데이터 연산

- 일반 집합 연산자 – 카티션 프로덕트(cartesian product)
 - 두 릴레이션 R과 S의 카티션 프로덕트 : $R \times S$
 - 릴레이션 R에 속한 각 튜플과 릴레이션 S에 속한 각 튜플을 모두 연결하여 만들어진 새로운 튜플로 결과 릴레이션을 구성
 - 결과 릴레이션의 특성
 - 차수는 릴레이션 R과 S의 차수를 더한 것과 같음
 - 카디널리티는 릴레이션 R과 S의 카디널리티를 곱한 것과 같음
 - 교환적 특징이 있음
 - $R \times S = S \times R$
 - 결합적 특징이 있음
 - $(R \times S) \times T = R \times (S \times T)$

관계 데이터 연산

- 순수 관계 연산자(relational operation)
 - 릴레이션의 구조와 특성을 이용하는 연산자
 - 선택: 릴레이션 R에서 조건을 만족하는 튜플들을 반환
 - 프로젝트: 릴레이션 R에서 주어진 속성들의 값으로만 구성된 튜플들을 반환
 - 조인: 공통 속성을 이용해 릴레이션 R과 S의 튜플들을 연결하여 만든 새로운 튜플들을 반환
 - 디비전: 릴레이션 S의 모든 튜플과 관련이 있는 릴레이션 R의 튜플들을 반환

관계 데이터 연산

- 순수 관계 연산자 – 선택(select)
 - 릴레이션에서 조건을 만족하는 튜플만 선택하여 결과 릴레이션을 구성
 - 수평적 연산자 : 결과 릴레이션은 연산 대상 릴레이션의 수평적 부분집합
 - 하나의 릴레이션을 대상으로 연산을 수행
 - 수학적 표현 : $\sigma_{\text{조건식}}(\text{릴레이션})$
 - 데이터 언어적 표현 : 릴레이션 where 조건식
 - 조건식
 - 비교식, 프레디캣(predicate)이라고도 함
 - 속성과 상수의 비교나 속성들 간의 비교로 표현
 - 비교 연산자(>, ≥, <, ≤, =, ≠)와 논리 연산자(∧, ∨, ¬)를 이용해 작성

관계 데이터 연산

- 순수 관계 연산자 – 프로젝트(project)
 - 릴레이션에서 선택한 속성의 값으로 결과 릴레이션을 구성
 - 수직적 연산자 : 결과 릴레이션은 연산 대상 릴레이션의 수직적 부분집합
 - 하나의 릴레이션을 대상으로 연산을 수행
 - 수학적 표현 : $\pi_{\text{속성리스트}}(\text{릴레이션})$
 - 데이터 언어적 표현 : 릴레이션[속성리스트]

관계 데이터 연산

- 순수 관계 연산자 – 조인(join)
 - 조인 속성을 이용해 두 릴레이션을 조합하여 결과 릴레이션을 구성
 - 조인 속성의 값이 같은 튜플만 연결하여 생성된 튜플을 결과 릴레이션에 포함
 - 조인 속성 : 두 릴레이션이 공통으로 가지고 있는 속성
 - 표현법 : 릴레이션1 \bowtie 릴레이션2
 - 동등 조인(equi-join)이라고도 함

관계 데이터 연산

- 세타 조인(theta-join, θ -join)
 - 동등 조인에 비해 더 일반화된 조인
 - 주어진 조인 조건을 만족하는 두 릴레이션의 모든 튜플을 연결하여 생성된 새로운 튜플로 결과 릴레이션을 구성
 - 결과 릴레이션의 차수는 두 릴레이션의 차수를 더한 것과 같음
 - 표현법 : 릴레이션1 $\bowtie_{A\theta B}$ 릴레이션2
 - θ 는 비교 연산자(>, \geq , <, \leq , =, \neq)를 의미

관계 데이터 연산

- 동등 조인(equi-join)
 - θ 연산자가 "="인 세타 조인을 의미
 - 표현법 : 릴레이션1 $\bowtie_{A=B}$ 릴레이션2

관계 데이터 연산

- 자연 조인(natural join)
 - 동등 조인의 결과 릴레이션에서 조인 속성이 한 번만 나타나게 하는 연산
 - 표현법 : 릴레이션1 \bowtie_N 릴레이션2

관계 데이터 연산

- 확장된 관계 대수 연산자
 - 기본 연산자를 확장한 연산자들이 제안됨
 - 예) 자연 조인 연산을 확장한 세미 조인과 외부 조인

관계 데이터 연산

- 확장된 관계 대수 연산자 – 세미 조인(semi-join)
 - 조인 속성으로 프로젝트 연산을 수행한 릴레이션을 이용하는 조인
 - 표현법 : 릴레이션1 \bowtie 릴레이션2
 - 릴레이션2를 조인 속성으로 프로젝트 연산한 후, 릴레이션1에 자연 조인하여 결과 릴레이션을 구성
 - 불필요한 속성을 미리 제거하여 조인 연산 비용을 줄이는 장점이 있음
 - 교환적 특징이 없음
 - $R \bowtie S \neq S \bowtie R$

관계 데이터 연산

- 확장된 관계 대수 연산자 – 외부 조인(outer-join)
 - 자연 조인 연산에서 제외되는 튜플도 결과 릴레이션에 포함시키는 조인
 - 결과 릴레이션에서 속성 값이 없는 경우는 널 값으로 처리
- 분류
 - 왼쪽(left) 외부 조인 / 오른쪽(right) 외부 조인 / 완전(full) 외부 조인
 - 모든 튜플을 결과 릴레이션으로 가져오는 릴레이션이 무엇이냐에 따라 분류
 - 자연 조인 연산에서 제외되는 튜플도 포함

관계 데이터 연산

- 확장된 관계 대수 연산자 – 외부 조인(outer-join)
 - 오른쪽 외부 조인
 - 표현법 : 릴레이션1 릴레이션2
 - 오른쪽에 있는 릴레이션2에 존재하는 모든 튜플을 결과 릴레이션에 포함시킴

관계 데이터 연산

- 확장된 관계 대수 연산자 – 외부 조인(outer-join)
 - 완전 외부 조인
 - 표현법 : 릴레이션1 릴레이션2
 - 두 릴레이션에 있는 모든 튜플을 결과 릴레이션에 포함시킴

관계 데이터 연산

- 관계 해석(relational calculus)
 - 처리를 원하는 데이터가 무엇인지만 기술하는 언어
 - 비절차 언어(nonprocedural language)
 - 수학의 프레디킷 해석(predicate calculus)에 기반을 두고 있음
 - 기능과 표현력 측면에서 관계 대수와 동등한 능력을 가짐
- 분류
 - 튜플 관계 해석(tuple relational calculus)
 - 도메인 관계 해석(domain relational calculus)

관계 데이터 연산

- 관계 해석(relational calculus)
 - 처리를 원하는 데이터가 무엇인지만 기술하는 언어
 - 비절차 언어(nonprocedural language)
 - 수학의 프레디킷 해석(predicate calculus)에 기반을 두고 있음
 - 기능과 표현력 측면에서 관계 대수와 동등한 능력을 가짐
- 분류
 - 튜플 관계 해석(tuple relational calculus)
 - 도메인 관계 해석(domain relational calculus)

학습 내용 요약

- 관계 데이터 연산
- SQL실습기초