

데이터베이스 설계 Database Design

강의주제: 2장 관계데이터모델

2023년 1학기

안문석

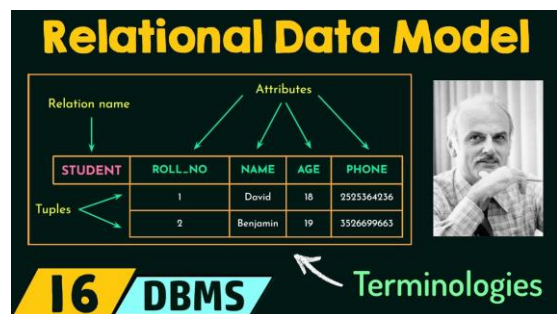
2023-04-24

Database Design

1

목차

- 관계 데이터 모델의 개념
- 무결성 제약조건
- 관계대수



2023-04-24

Database Design

2

학습목표

- 관계 데이터 모델의 개념 이해
- 관계 데이터 모델의 제약조건
- 관계 데이터 모델 연산인 관계대수의 종류와 작성법

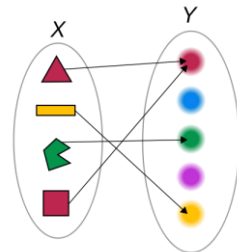
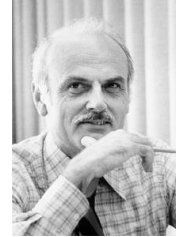
01. 관계 데이터 모델의 개념

- 릴레이션
- 릴레이션 스키마와 인스턴스
- 릴레이션의 특징
- 관계 데이터 모델

Relation

01. 관계 데이터 모델의 개념

- 관계 데이터 모델(relational data model)의 탄생
 - 1970년에 IBM의 E. F. Codd에 의해 제안
- 관계 데이터 모델의 특성
 - 수학의 **릴레이션(relation)**과 **집합(set)** 이론에 기초
 - 릴레이션: 일반 사용자는 **테이블(table)** 형태로 생각
 - 통상적인 테이블의 개념과는 다름
 - 관계 데이터 모델의 직관적인 이해에 도움
 - 테이블의 열(column) = 필드(field) 혹은 아이템(item)
 ≡ 관계 데이터 모델의 속성(attribute)
 - 테이블의 행(row) = 레코드(record)
 ≡ 관계 데이터 모델의 튜플(tuple)



2023-04-24

Database Design

5

1. 릴레이션(Relation: 관계)

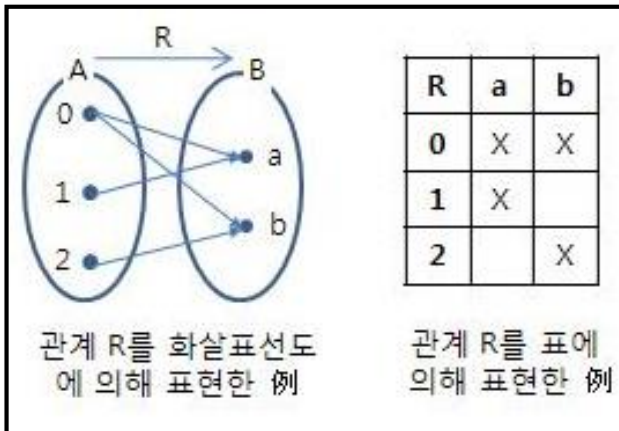
- 두 집합의 원소들 간을 관련시킨다는 수학 용어
 - 유사한 대상들을 관련시키는 성질들의 특정 조합
- 관계 표기 : R
 - $a R b$ ⇔ a 가 R 에 의해 b 와 관계되어 있음을 표현
- 관계 표현
 - 순서쌍에 의한 집합 형태로 표현 : $(a,b) \in R$
 - 순서쌍 집합에 속하면서 순서쌍을 이루는 원소들이 관계가 있다고 말함
 - 예) $A = \{1,2,3,4\}$, $B = \{1,2,3,4\}$ ($a \in A$, $b \in B$)
 - $a R b$ 에서 관계 R 이 **우항 b 보다 좌항 a 가 작다** 라는 관계이면,
 - $R = \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$
- 부분집합에 의한 표현
 - 두 집합 A , B 의 카티전 곱($A \times B$, 모든 순서쌍)들 중에서 부분집합에 의한 표현
 - 특정 순서쌍들이 카티전 곱의 부분집합을 이룬 형태

2023-04-24

Database Design

6

1. 릴레이션(Relation: 관계)



※ 이같은 표현들을 다음과 같이 말함
- 집합 A에서 집합 B로의 관계

1. 릴레이션(Relation) R

• 수학적 정의

- 릴레이션 R : 카티션 프로덕트(Cartesian product)의 부분집합
- $R \subseteq D_1 \times D_2 \times \dots \times D_n$, 단 D_i : i 번째 도메인
- 즉 n -튜플, $\langle d_1, d_2, \dots, d_n \rangle$ 의 집합
- $d_i \in D_i, i = 1, 2, \dots, n$
- n : R의 차수(degree : 1차, 2차, 3차, ..., n 차)
- 튜플의 수 : 카디널리티(cardinality)

• 개념적 정의

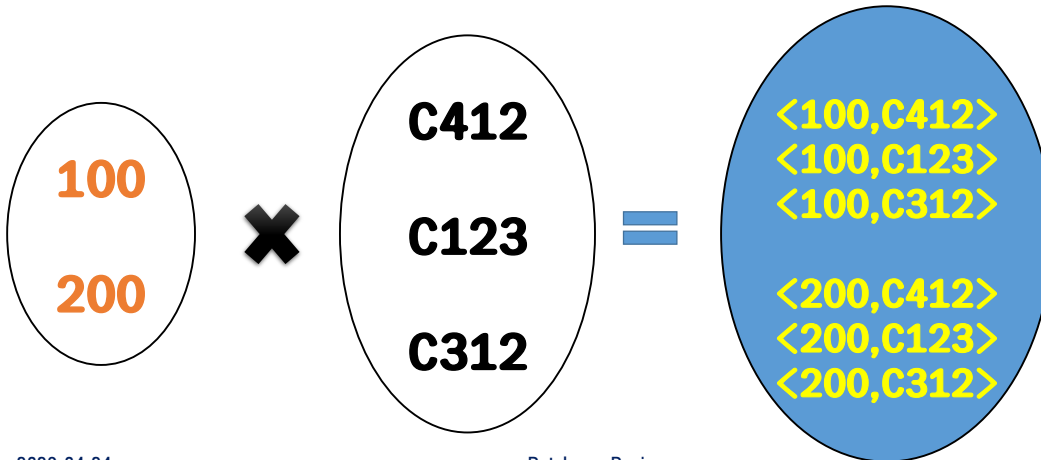
- 릴레이션 스키마 + 릴레이션 인스턴스

1. 릴레이션(Relation) R

학번(Sno)

과목번호(Cno)

학번 × 과목번호(Sno × Cno)



2023-04-24

Database Design

9

1. 릴레이션

• 릴레이션(relation) : 행과 열로 구성된 테이블

릴레이션
관련
한글 용어

용어	한글 용어	비고
relation	릴레이션, 테이블	“관계” 라고 하지 않음
relational data model	관계 데이터 모델	
relational database	관계 데이터베이스	
relational algebra	관계대수	
relationship	관계	

2023-04-24

Database Design

10

1. 릴레이션

학생(STUDENT) 테이블 : 릴레이션


학생 (STUDENT)	학번 (Sno)	이름 (Sname)	학년 (Year)	학과 (Dept)
	100	나수영	4	컴퓨터
	200	이찬수	3	전기
	300	정기태	1	컴퓨터
	400	송병길	4	컴퓨터
	500	박종와	2	산공

2023-04-24

Database Design

11

1. 릴레이션

도서 1, 축구의 역사, 쿼스포츠, 7000		도서번호	도서이름	출판사	가격
도서 2, 축구하는 여자, 나무수, 13000		1	축구의 역사	쿼스포츠	7000
도서 3, 축구의 이해, 대안미디어, 22000		2	축구 아는 여자	나무수	13000
도서 4, 골프 바이블, 대안미디어, 35000		3	축구의 이해	대안미디어	22000
도서 5, 피겨 교본, 쿼스포츠, 8000		4	골프 바이블	대안미디어	35000
		5	피겨 교본	쿼스포츠	8000

2023-04-24

Database Design

12

1. 릴레이션

데이터와 테이블(릴레이션)

도서번호 = {1,2,3,4,5}

도서이름 = {축구역사, 축구하는 여자, 축구의 이해, 골프 바이블, 피겨 교본}

출판사 = {쿠팡, 나무수, 대한미디어}

가격 = {7000, 13000, 22000, 35000, 8000}

→ 첫 번째 행(1, 축구역사, 쿠팡, 7000)의 경우

- 네 개의 집합에서 각각 원소 한 개씩 선택하여 만들어진 것
- 이 원소들이 관계(relationship)를 맺고 있다.

2023-04-24

Database Design

13

1. 릴레이션

• 관계(relationship)

- ① 릴레이션 내에서 생성되는 관계 : 릴레이션 내 데이터들의 관계
- ② 릴레이션 간에 생성되는 관계 : 릴레이션 간의 관계

도서(도서번호, 도서이름, 출판사, 가격)

주문(도서번호, 고객번호, 판매가격, 주문일자)

고객(고객번호, 이름, 주민번호, 주소, 핸드폰)

릴레이션 간의 관계

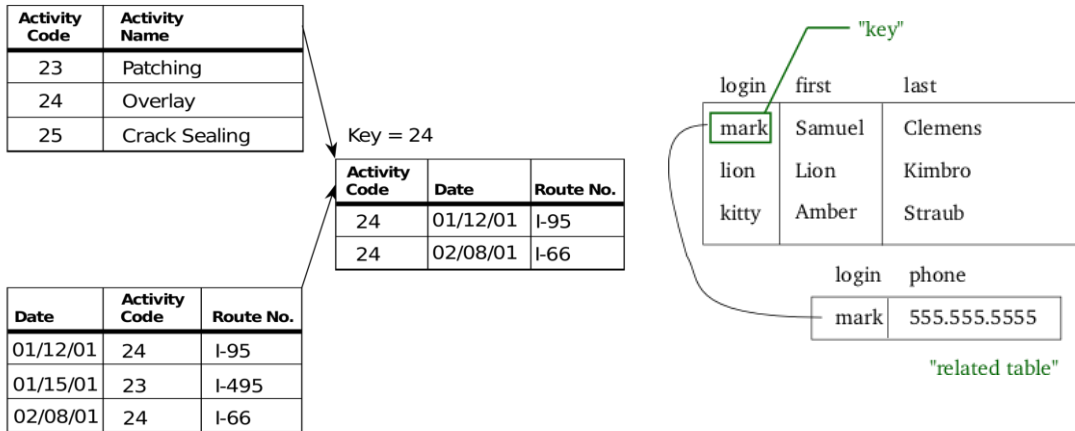
2023-04-24

Database Design

14

1. 릴레이션

Relational Model



2023-04-24

Database Design

15

2. 릴레이션 스키마와 인스턴스

스키마(내포)
Schema

속성(에트리뷰트), 열(column) 이라고도 함. (차수=4)

도서

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구 아는 여자	나무수	13000
3	축구의 이해	대안미디어	22000
4	골프 바이블	대안미디어	35000
5	피겨 교본	굿스포츠	8000

도서 릴레이션

튜플(tuple), 행(row) 이라고도 함. (카디널리티=5)

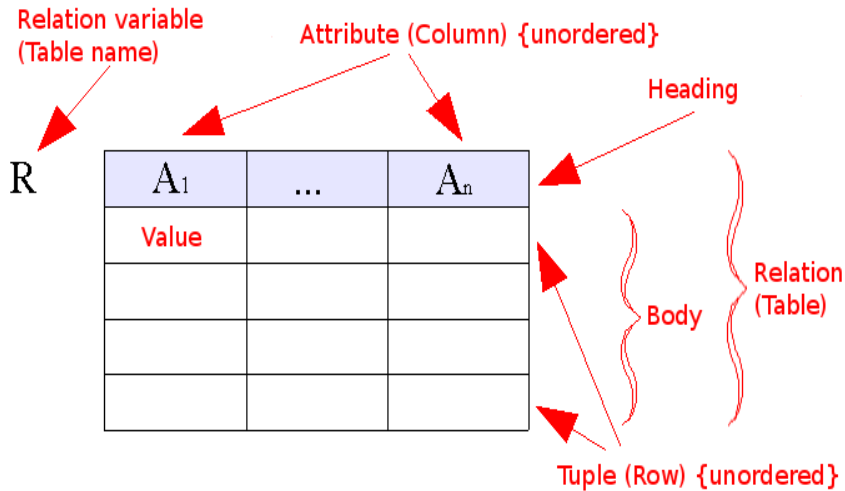
인스턴스(외연, Instance)

2023-04-24

Database Design

16

2. 릴레이션 스키마와 인스턴스



2023-04-24

Database Design

17

2.1 릴레이션 스키마

- 릴레이션 내포 (relation intension)
- 릴레이션 스킴 (relation scheme)
 - 릴레이션 이름 + 속성 이름
 - $R(A_1, A_2, \dots, A_n), \quad A_i \Leftrightarrow D_i$
 - $\equiv R(\{A_1, A_2, \dots, A_n\})$
- 정적 성질
 - 시간에 무관
 - 시간에 따라 변경되지 않음
 - 릴레이션 타입과 같은 의미



2023-04-24

Database Design

18

2.1 릴레이션 스키마

- 스키마의 요소
 - 속성(attribute) : 릴레이션 스키마의 열
 - 도메인(domain) : 속성이 가질 수 있는 값의 집합
 - 차수(degree) : 속성의 개수
- 스키마의 표현
 - 릴레이션 이름(속성1 : 도메인1, 속성2 : 도메인2 ...)
 - 예) 도서(도서번호, 도서이름, 출판사, 가격)

2.2 릴레이션 인스턴스

- 릴레이션 외연 (relation **extension**)이라고도 함
- An instance is an occurrence or a copy of an object.
- 릴레이션 R의 인스턴스
 - 어느 한 시점에 릴레이션 R이 포함하고 있는 튜플들의 집합
 - $\{ \langle V_1, V_2, \dots, V_n \rangle \mid V_i \in D_i \}$
 - 릴레이션의 내용, 상태, snapshot
- 튜플: $\{ (attr_1=V_1, attr_2=V_2, \dots, attr_n=V_n) \}$
- 동적 성질
 - 삽입, 삭제, 갱신으로 시간에 따라 변함
 - 릴레이션 값(보통 릴레이션)

2.2 릴레이션 인스턴스

• 인스턴스 요소

- 튜플(tuple) : 릴레이션의 행
- 카디널리티(cardinality) : 튜플의 수

릴레이션 구조와 관련된 용어

릴레이션 용어	동일 의미로 통용되는 용어	파일 시스템 용어
릴레이션(relation)	테이블(table)	파일(file)
스키마(schema)	내포(intension)	헤더(header)
인스턴스(instance)	외연(extension)	데이터(data)
튜플(tuple)	행(row)	레코드(record)
속성(attribute)	열(column)	필드(field)

2023-04-24

Database Design

21

2.2 릴레이션 인스턴스

• 인스턴스 요소

- 튜플(tuple) : 릴레이션의 행
- 카디널리티(cardinality) : 튜플의 수

릴레이션 구조와 관련된 용어

릴레이션 용어	동일 의미로 통용되는 용어	파일 시스템 용어
릴레이션(relation)	테이블(table)	파일(file)
스키마(schema)	내포(intension)	헤더(header)
인스턴스(instance)	외연(extension)	데이터(data)
튜플(tuple)	행(row)	레코드(record)
속성(attribute)	열(column)	필드(field)

2023-04-24

Database Design

22

→ 튜플이 가지는 속성의 개수는 릴레이션 스키마의 개수와 동일하고,
→ 릴레이션 내의 모든 튜플들은 서로 중복되지 않아야 함

3. 릴레이션의 특징

- 속성은 단일 값을 가짐
- 속성은 서로 다른 이름을 가짐
- 한 속성의 값은 모두 같은 도메인 값을 가짐
- **속성의 순서는 상관없음**
 - 속성의 순서가 달라도 릴레이션 스키마는 같음
 - 예) 릴레이션 스키마에서 (이름, 주소) 순으로 속성을 표시하거나 (주소, 이름) 순으로 표시하여도 상관없음
- 릴레이션 내의 중복된 튜플은 허용하지 않음
- **튜플의 순서는 상관없음**
 - 튜플의 순서가 달라도 같은 릴레이션임

2023-04-24

Database Design

23

3. 릴레이션의 특징

릴레이션의 특징에 위배된 경우

도서번호	도서이름	출판사	가격
1	축구역의 역사	국스포스	7000
2	축구 아는 여자	나무수	13000
3	축구의 이해	대안미디어	22000
4	골프 바이블	대안미디어	35000
5	피겨 교본	국스포스	8000
5	피겨 교본	국스포스	8000
6	피겨 교본, 피겨 기초	국스포스	8000

속성의 값은 단일 값이어야 함

동일한 튜플이 중복되면 안 됨

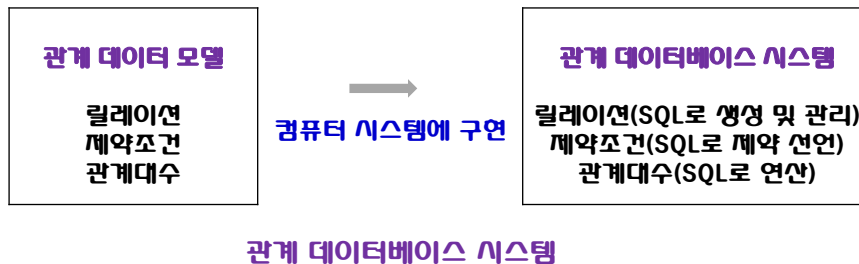
2023-04-24

Database Design

24

4. 관계 데이터 모델

- 데이터를 2차원 테이블 형태인 **릴레이션**으로 표현
- 릴레이션에 대한 **제약조건(constraints)**과 관계 연산을 위한 **관계대수(relational algebra)**를 정의함



연습문제 풀이

1. 다음 중 관계 데이터 모델의 릴레이션에 대한 설명 중 옳지 않은 것은?

- ① 릴레이션은 릴레이션 스키마와 릴레이션 인스턴스로 구성된다.
- ② 릴레이션 스키마를 릴레이션 외연(extension)이라고 한다.
- ③ 릴레이션의 스키마는 정적인 성질을 가진다.
- ④ 릴레이션 인스턴스는 동적인 성질을 가진다.

2. 릴레이션의 특징으로 알맞은 것은?

- ① 중복된 투플이 존재한다.
- ② 투플 간의 순서가 정의된다.
- ③ 속성 간의 순서가 정의된다.
- ④ 모든 속성 값은 원자값이다.

3. 하나의 속성이 가질 수 있는 값을 총칭하여 무엇이라 하는가?

- ① 투플
- ② 릴레이션
- ③ 도메인
- ④ 엔티티

02. 무결성 제약조건

- 키
- 무결성 제약조건
- 무결성 제약조건의 수행

1. 키

- 특정 튜플을 식별할 때 사용하는 속성 혹은 속성의 집합
- 릴레이션은 중복된 튜플을 허용하지 않음
 - 각각의 튜플에 포함된 속성들 중 어느 하나(혹은 하나 이상)는 값이 달라야 함.
 - 즉 키가 되는 속성(혹은 속성의 집합)은 반드시 값이 달라서 튜플들을 서로 구별할 수 있어야 함
- 키는 릴레이션 간의 관계를 맺는 데도 사용됨

1. 키



자동차 1 대당 키는 단 하나

2023-04-24

Database Design

29

1. 키: 마당서점 데이터베이스

고객

고객번호	이름	주민번호	주소	핸드폰
1	박지성	810101-1111111	영국 맨체스터	000-5000-0001
2	김연아	900101-2222222	대한민국 서울	000-6000-0001
3	장미란	830101-2333333	대한민국 강원도	000-7000-0001
4	주신수	820101-1444444	미국 클리블랜드	000-8000-0001

도서

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구는 여자	나무수	13000
3	축구의 이해	대안미디어	22000
4	골프 바이블	대안미디어	35000
5	피겨 교본	굿스포츠	8000

주문

고객번호	도서번호	판매가격	주문일자
1	1	7000	2014-07-01
1	2	13000	2014-07-03
2	5	8000	2014-07-03
3	2	13000	2014-07-04
4	4	35000	2014-07-05
1	3	22000	2014-07-07
4	3	22000	2014-07-07

2023-04-24

Database Design

30

1.1 슈퍼키(Super Key)

- 튜플을 **유일하게** 식별할 수 있는 하나의 속성 혹은 속성의 집합
- 고객 릴레이션의 예
 - 고객번호 : 고객별로 유일한 값이 부여되어 있기 때문에 튜플을 식별할 수 있음
 - 이름 : 동명이인이 있을 경우 튜플을 유일하게 식별할 수 없음
 - 주민번호 : 개인별로 유일한 값이 부여되어 있기 때문에 튜플을 식별할 수 있음

1.1 슈퍼키

- 주소 : 가짜끼리는 같은 정보를 사용하므로 튜플을 식별할 수 없음
- 핸드폰 :
 - 한 사람이 여러 개의 핸드폰 사용 가능
 - 핸드폰을 사용하지 않는 사람이 있을 수 있기 때문에 튜플을 식별할 수 없음
- 고객 릴레이션은 고객번호와 주민번호를 포함한 모든 속성의 집합이 슈퍼키
 - (주민번호), (주민번호, 이름), (주민번호, 이름, 주소), (주민번호, 이름, 핸드폰)
 - (고객번호), (고객번호, 이름, 주소), (고객번호, 이름, 주민번호, 주소, 핸드폰)

1.2 후보키

- 튜플을 **유일하게** 식별 가능한 속성의 **최소 집합**
- 주문 릴레이션의 예
 - 고객번호
 - 한 명의 고객이 여러 권의 도서를 구입할 수 있으므로 후보키가 될 수 없음.
 - 고객번호가 1인 박지성 고객은 세 번의 주문 기록이 있으므로 튜플을 유일하게 식별할 수 없음

1.2 후보키

- 도서번호
 - 도서번호가 2인 '축구 아는 여자' 는 두 번의 주문 기록이 있으므로 튜플을 유일하게 식별할 수 없음
- 주문 릴레이션의 후보키
 - 2개의 속성을 합한 **(고객번호, 도서번호)**
 - 2개 이상의 속성으로 이루어진 키: **복합키(composite key)**

1.3 기본키

- 여러 후보키 중 하나를 선정하여 대표로 삼는 키
- 기본키 선정 시 고려사항
 - 릴레이션 내 튜플을 식별할 수 있는 **고유한 값**을 가져야 함
 - **NULL 값은 허용하지 않음**
 - 키 값의 변동이 일어나지 않아야 함
 - **최대한 적은 수의 속성**을 가진 것이어야 함
 - 향후 키를 사용하는 데 있어서 문제 발생 소지가 없어야 함
- 릴레이션 스키마를 표현할 때 기본키는 밑줄로 표시
 - 릴레이션 이름(속성1, 속성2, ..., 속성N)
 - 고객(고객번호, 이름, 주민번호, 주소, 핸드폰)
 - 도서(도서번호, 도서이름, 출판사, 가격)

1.4 대리키

- 대리키(surrogate key) 혹은 인조키(artificial key)
- 필요성
 - 기본키가 보안을 요함
 - 여러 개의 속성으로 구성되어 복잡
 - 마땅한 기본키가 없을 때
 - 일련번호 같은 가상의 속성을 만들어 기본키로 삼는 경우
- DBMS나 관련 소프트웨어에서 임의로 생성하는 값
- 사용자가 직관적으로 그 값의 의미를 알 수 없음

1.4 대리키(Surrogate Key)

주문

주문번호	고객번호	도서번호	판매가격	주문일자
1	1	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	3	2	13000	2014-07-04
5	4	4	35000	2014-07-05
6	1	3	22000	2014-07-07
7	4	3	22000	2014-07-07

대리키를 사용하도록 변경된 주문 릴레이션

1.5 대체키

- 대체키(alternate key)
 - 기본키로 선정되지 않은 후보키
- 고객 릴레이션
 - 고객번호와 주민번호 중 고객번호를 기본키로 정하면 주민번호가 대체키가 됨

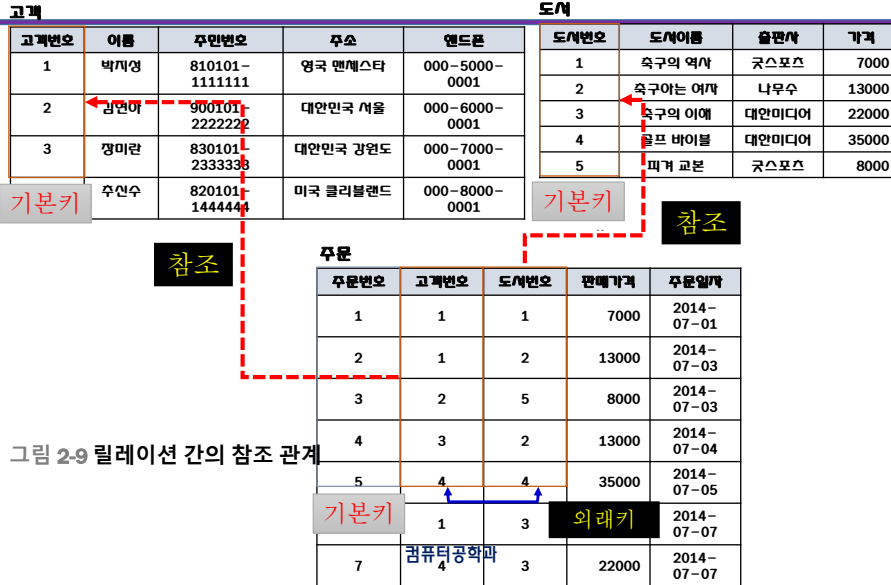
1.6 외래키

- 다른 릴레이션의 기본키를 참조하는 속성
- 다른 릴레이션의 기본키를 참조하여 관계 데이터 모델의 특징인 릴레이션 간의 관계(relationship)를 표현
- 외래키의 특징
 - 관계 데이터 모델의 릴레이션 간의 관계를 표현함
 - 다른 릴레이션의 기본키를 참조하는 속성임

1.6 외래키

- 참조하고(외래키) 참조되는(기본키) 양쪽 릴레이션의 도메인은 서로 같아야 함
- 참조되는(기본키) 값이 변경되면 참조하는(외래키) 값도 변경됨
- NULL 값과 중복 값 등이 허용됨
- 자기 자신의 기본키를 참조하는 외래키도 가능함
- 외래키가 기본키의 일부가 될 수 있음

1.6 외래키



2023-04-24

41

1.6 외래키

- 외래키 사용 시 참조하는 릴레이션과 참조되는 릴레이션이 꼭 다른 릴레이션일 필요는 없음
- 즉 자기 자신의 기본키를 참조할 수도 있음

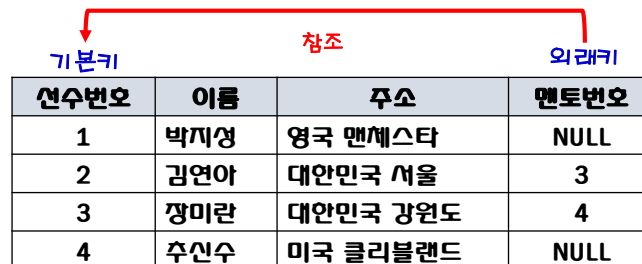
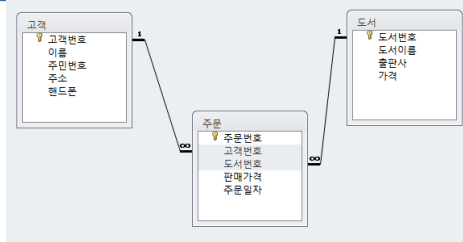


그림 2-10 멘토 릴레이션
컴퓨터공학과

2023-04-24

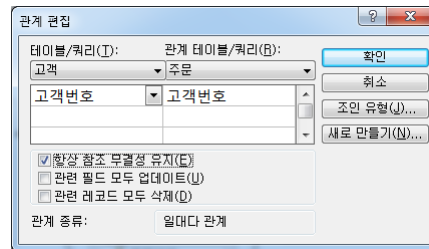
42

1.6 외래키



A key used to link two tables together.

A field (or collection of fields) in one table that refers to the **PRIMARY KEY** in another table.



2023-04-24

컴퓨터공학과

43

1. 키 - 내용 요약



우보키(두표를 식별할 수 있는 속성의 최소 집합)

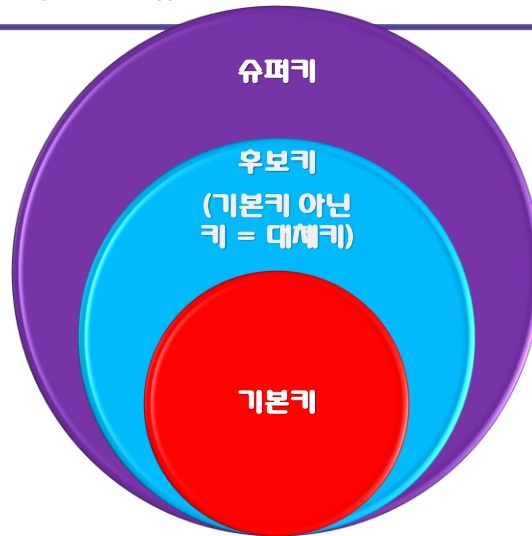
키의 포함 관계

2023-04-24

컴퓨터공학과

44

1. 키 – 키들의 관계



2023-04-24

컴퓨터공학과

45

11. 키 – 예제

English Monarchs

Monarch Name	Monarch Number	Royal House
Edward	II	Plantagenet
Edward	III	Plantagenet
Richard	III	Plantagenet
Henry	IV	Lancaster

처음에, 모든 애트리뷰트 집합들 열거:

- {~~Monarch-Name~~}
- {~~Monarch-Number~~}
- {~~Royal-House~~}
- {Monarch Name, Monarch Number}
- {~~Monarch-Name~~, ~~Royal-House~~}
- {~~Monarch-Number~~, ~~Royal-House~~}
- {Monarch Name, Monarch Number, Royal House}

두 번째, 슈퍼키 요구조건(유일성)을 충족 못하는 모든 집합들 제거:

{~~Monarch Name~~, ~~Royal House~~}는 슈퍼키가 아님 - (~~Edward~~, ~~Plantagenet~~)가 중복

서로 구별할 수 있는 두 개의 튜플:

(Edward, ~~II~~, Plantagenet)

(Edward, ~~III~~, Plantagenet)

제거 후 마지막으로 남아 있는 애트리뷰트 집합은 가능한 슈퍼키만 남음:

{~~Monarch Name~~, ~~Monarch Number~~} (**Candidate Key**)

{Monarch Name, Monarch Number, Royal House}

2023-04-24

컴퓨터공학과

46

Start lecture from here

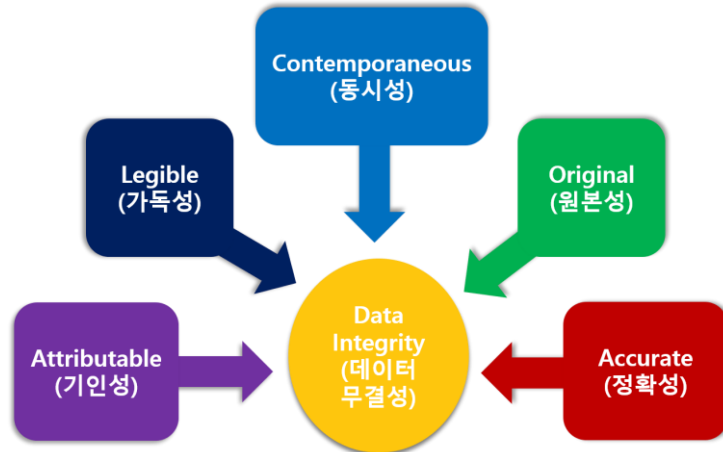


- 데이터베이스설계
- (9주-1) 강의
- 2022년 4월 28일

2. 무결성 제약조건

- 데이터 무결성(**data integrity**, 無缺性)
 - DB에 저장된 데이터의 **완전성, 일관성, 정확성** 지키는 것
 - 원본 데이터의 변경이나 파괴 없이 보존되는 특성을 내포
 - **ALCOA**
 - 완전하고 일관되며 정확한 데이터는 출처가 있어야 하며(attributable)
 - 읽을 수 있고(legible)
 - 동시에 기록되어야 하며(contemporaneously recorded)
 - 원본 또는 진정한 사본이어야 하며(original or a true copy)
 - 정확(accurate)해야 함

2. 무결성 제약조건



2. 무결성 제약조건

- Domain Integrity
- Entity Integrity
- Referential Integrity
- User-Defined Integrity

2. 무결성 제약조건

• 도메인 무결성 제약조건

- 도메인 제약(domain constraint)
- 릴레이션 내의 튜플들이 각 속성의 도메인에 지정된 값만을 가져야 한다는 조건
- SQL 문에서 데이터 형식(type) 널(null/not null) 기본 값(default), 체크

Employee_id	Name	Salary	Age
1	Andrew	486522	25
2	Angel	978978	30
3	Anamika	697abc	35

This value is out of domain(not INTEGER)so it is not acceptable.

DOMAIN INTEGRITY

2023-04-24

컴퓨터공학과

51

2. 무결성 제약조건

• 개체 무결성 제약조건

- 기본키 제약(primary key constraint)
- 기본키는 NULL 값을 가져서는 안되며 릴레이션 내에 오직 하나의 값만 존재

ID	Customer_Name	Age
1	Andrew	18
2	Angel	20
	Angel	20

Primary Key

This value cannot be NULL as we will not be able to identify customers uniquely

ENTITY INTEGRITY

2023-04-24

컴퓨터공학과

52

2. 무결성 제약조건

• 참조 무결성 제약조건

- 외래키 제약(**foreign key constraint**)
- 릴레이션 간의 참조 관계를 선언하는 제약조건
- 자식 릴레이션의 외래키는 부모 릴레이션의 기본키와 **도메인이 동일**해야 함
- 자식 릴레이션의 값이 변경될 때 부모 릴레이션의 제약을 받는다는 것임

2. 무결성 제약조건

Student (First Table)

Roll_no	Student_name	Age	Course_id
1	Andrew	18	78
2	Angel	19	16
3	Priya	20	56
4	Analisa	21	

Primary
Key

Foreign
Key

This value is not allowed because this value is not defined as a primary key in the course table.

The value can be NULL as the student(Analisa) may not have taken any course.

Course (Second Table)

Course_id	Course_name	Duration (months)
78	Big Data	4
56	Algorithm	2

Primary
Key

REFERENTIAL INTEGRITY

2. 무결성 제약조건

• 사용자 정의 무결성

- 세 가지 무결성으로 데이터 무결성을 유지하지 않을 때

- 트리거 및 저장 프로시저를 통해 구현

- 사용자 정의 제약 조건

- 직원의 ID가 항상 'AfterAcademy'와 같은 특정 문자로 시작하고 그 뒤에 숫자가 와야 한다.

EMPLOYEE TABLE

Emp_id	Emp_name	Job_name	Salary	Mobile_no
AfterAcademy001	John	Engineer	100000	9111037890
AfterAcademy002	Adam	Analyst	500000	9587569214
Academy003	Kande	Manager	890000	7895212355

This is not following the user-defined constraint so it is not acceptable.

USER DEFINED INTEGRITY

2. 무결성 제약조건

EMPLOYEE TABLE

Emp_id	Emp_name	Job_name	Salary	Mobile_no
AfterAcademy001	John	Engineer	100000	9111037890
AfterAcademy002	Adam	Analyst	500000	9587569214
Academy003	Kande	Manager	890000	7895212355

This is not following the user-defined constraint so it is not acceptable.

USER DEFINED INTEGRITY

2. 무결성 제약조건

제약조건의 정리

구분	도메인	키	
	도메인 무결성 제약조건	개체 무결성 제약조건	참조 무결성 제약조건
제약 대상	속성	투플	속성과 투플
같은 용어	도메인 제약 (Domain Constraint)	기본키 제약 (Primary Key Constraint)	외래키 제약 (Foreign Key Constraint)
연관되는 키	-	기본키	외래키
NULL 값 허용 여부	여용	불가	여용
릴레이션 내 제약조건의 개수	속성의 개수와 동일	1개	0~여러 개
기타	• 투플 삽입, 수정 시 제약 사항 우선 확인	• 투플 삽입/수정 시 제약 사항 우선 확인	• 투플 삽입/수정 시 제약/사항 우선 확인 • 부모 릴레이션의 투플 수정/삭 제 시 제약/사항 우선 확인

2023-04-24

컴퓨터공학과

57

3.1 개체 무결성 제약조건 수행

- 삽입
 - 기본키 값이 같으면 삽입이 금지됨
- 수정
 - 기본키 값이 같거나 NULL로도 수정이 금지됨
- 삭제
 - 특별한 확인이 필요하지 않으며 즉시 수행함

2023-04-24

컴퓨터공학과

58

3.1 개체 무결성 제약조건

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

학생 릴레이션

(501, 남슬찬, 1001)



삽입 거부

(NULL, 남슬찬, 1001)



삽입 거부

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

개체 무결성 제약조건 수행 예(기본키 중첩 및 NULL 값 삽입)

2023-04-24

컴퓨터공학과

59

3.2 참조 무결성 제약조건 수행

• 삽입

- 예: 학생 릴레이션에 다음을 삽입하고자 할 때: (601, 박세리, 3001)
- 학생(자식 릴레이션)
 - 참조받는 테이블에 외래키 값이 없으므로 삽입이 금지됨
- 학과(부모 릴레이션)
 - 투플 삽입한 후 수행하면 정상적으로 진행됨

(601, 박세리, 3001)

Insert

학생(자식 릴레이션)

학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	추신수	1001

학과(부모 릴레이션)

학과코드	학과명
1001	컴퓨터학과
2001	체육학과

참조

그림 2-14 학생관리 데이터베이스

컴퓨터공학과

2023-04-24

60

3.2 참조 무결성 제약조건

• 삭제

- 학과(부모 릴레이션)
 - 참조하는 테이블을 같이 삭제할 수 있어서 금지 또는 다른 추가작업이 필요
- 학생(자식 릴레이션) : 바로 삭제 가능
- 부모 릴레이션의 튜플을 삭제할 경우 참조 무결성 조건 수행 고려사항
 - 즉시 작업을 중지
 - 자식 릴레이션의 관련 튜플을 삭제
 - 초기에 설정된 다른 어떤 값으로 변경
 - NULL 값으로 설정

• 수정

- 삭제와 삽입 명령 연속 수행됨
- 부모 릴레이션의 수정이 일어날 경우
 - 삭제 옵션에 따라 처리된 후 문제가 없으면 다시 삽입 제약조건에 따라 처리됨

2023-04-24

컴퓨터공학과

61

3.2 참조 무결성 제약조건

참조 무결성 제약조건의 옵션(부모 릴레이션에서 튜플을 삭제할 경우)

명령어	의미	예
RESTRICTED	자식 릴레이션에서 참조하고 있을 경우 부모 릴레이션의 삭제 작업을 거부함	학과 릴레이션의 튜플 삭제 거부
CASCADE	자식 릴레이션의 관련 튜플을 같이 삭제 처리함	학생 릴레이션의 관련 튜플을 삭제
DEFAULT	자식 릴레이션의 관련 튜플을 미리 설정해둔 값으로 변경함	학생 릴레이션의 학과가 다른 학과로 자동 배경
NULL	자식 릴레이션의 관련 튜플을 NULL 값으로 설정함(NULL 값을 허용하는 경우)	학과 릴레이션의 학과가 NULL 값으로 변경

2023-04-24

컴퓨터공학과

62

3.2 참조 무결성 제약조건

② 참조 확인 (2001, 체육학과) ① 삭제 요청

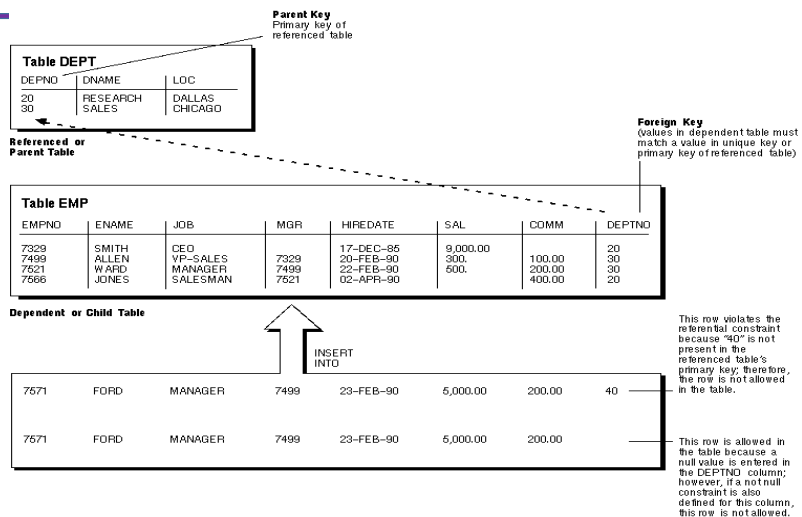
학번	이름	학과코드
501	박지성	1001
401	김연아	2001
402	장미란	2001
502	주연수	1001

학과코드	학과명
1001	컴퓨터학과
2001	체육학과

- ① **RESTRICTED** : 요청한 삭제 작업중지(에러 처리)
- ② **CASCADE** : 학생 릴레이션의 해당 튜플을 같이 연쇄적으로 삭제 (CASCADE)
- ③ 기본값으로 변경(미리 설정한 값, **DEFAULT**)
- ④ **NULL** 값으로 설정

참조 무결성 제약조건에서 부모 테이블의 주를 삭제할 경우

참조무결성의 예



참조무결성 위반 사례

artist_id	artist_name
1	Bono
2	Cher
3	Nuno Bettencourt

artist_id	album_id	album_name
3	1	Schizophonic
4	2	Eat the rich
3	3	Crave (single)

2023-04-24

컴퓨터공학과

65

연습문제 풀이

4. 외래키(FK, Foreign Key)에 대한 설명으로 옳은 것은?

- ① 릴레이션 R1에 속한 속성 집합 FK가 다른 릴레이션 R2의 기본키인 것을 말한다.
- ② 외래키와 기본키가 정의된 도메인은 다를 수도 있다.
- ③ 외래키는 NULL 값을 가질 수 없다.
- ④ 둘 이상의 후보키 중에서 하나를 선정하여 대표로 삼은 키를 말한다.

5. 한 릴레이션의 기본키를 구성하는 어떠한 속성 값도 NULL 값이나 중복값을 가질 수 없다는 것을 의미하는 제약조건은?

- ① 개체 무결성 제약조건
- ② 참조 무결성 제약조건
- ③ 보안 무결성 제약조건
- ④ 정보 무결성 제약조건

11. 다음은 릴레이션에서 더는 삽입할 수 없는 데이터이다. 다음 물음에 답하시오.

- (1) 릴레이션 R과 S의 후보키를 찾아라.
- (2) 릴레이션 R과 S의 기본키를 찾아라.

R

A	B	C
a1	b1	c1
a2	b1	c1
a3	b1	c2
a4	b2	c3

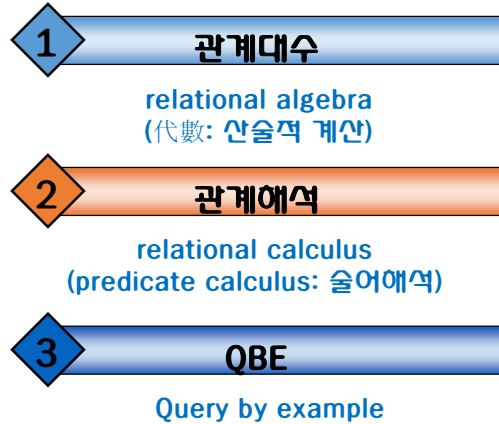
C	D	E
c1	d2	e1
c1	d1	e2
c2	d3	e3
c3	d3	e3

2023-04-24

컴퓨터공학과

66

관계연산



관계 데이터 연산

- 연산(operation)과 데이터 언어(data language)
 - 연산 : 명령 수행, 시스템 입장
 - 데이터 언어 : 사용자 입장
- 관계 데이터 언어(relational data language)
 - 형언어(formal language): a set of strings of symbols
 - i. 관계 대수(relational algebra)
 - 절차 언어 : **how, what**
 - ii. 관계 해석(relational calculus)
 - 비 절차 언어 : **what**
 - Two calculi
 - 튜플 관계 해석(tuple relational calculus)
 - 도메인 관계 해석 (domain relational calculus)
- 관계 해석과 관계 대수는 표현과 기능 면에서 **동등**
 - **Relationally complete**

관계해석 예

- 튜플 관계 해석(tuple relational calculus)
 - $\{t \mid \text{EMPLOYEE}(t) \text{ and } t.\text{SALARY} > 50000\}$
 - $\{t \mid P(t)\}$
- 도메인 관계 해석 (domain relational calculus)
 - $\{uv \mid (\exists q) (\exists r) (\exists s)$
 $(\text{EMPLOYEE}(qrstuvwxyz) \text{ and } q = \text{'John'}$
 $\text{and } r = \text{'B' and } s = \text{'Smith' })\}$

2023-04-24

컴퓨터공학과

69

05 QBE의 예

- A query method implemented in most DBSs
- A graphical query language
- 사용자는 예제 테이블을 만들어 쿼리를 작성
- 명령 P.

<i>Sailors</i>	<i>sid</i>	<i>sname</i>	<i>rating</i>	<i>age</i>
P.				

Select * From Sailors;

<i>Sailors</i>	<i>sid</i>	<i>sname</i>	<i>rating</i>	<i>age</i>
P.			10	

DRC
 $\{\langle I, N, 10, A \rangle \mid \langle I, N, 10, A \rangle \in \text{Sailors}\}$

Select *
 From Sailors
 Where age = 10;

2023-04-24

컴퓨터공학과

70

03. 관계대수

- 관계대수
- 선택연산과 투영연산
- 집합연산
- 조인
- 디비전
- 관계대수 예제

1. 관계대수

- Relational algebra
- 關係代數
- 릴레이션을 처리하기 위한 연산의 집합
- 릴레이션에서 원하는 결과를 얻기 위해 수학의 대수와 같은 연산을 이용하여 질의하는 방법을 기술하는 언어

1. 관계대수

• 관계대수와 관계해석

• 관계대수

- 어떤 데이터를 어떻게 찾는지에 대한 처리 절차를 명시하는 **절차적인 언어**이며, **DBMS 내부의 처리 언어**로 사용됨

• 관계해석(relational calculus, 關係解析)

- 어떤 데이터를 찾는지만 명시하는 선언적인 **비절차적 언어**로 **SQL의 이론적인 기반**을 제공함

→ 모두 관계 데이터 모델의 중요한 언어

→ 실제 동일한 표현 능력을 가지고 있음

Relationally complete

1.1 관계의 수학적 의미

• 예) $A = \{2, 4\}$, $B = \{1, 3, 5\}$ 일 때

- $A \times B = \{(2,1), (2,3), (2,5), (4,1), (4,3), (4,5)\}$

• 릴레이션 R은 카티전 프로덕트의 부분집합

- 예) $R1 = \{(2,1), (4,1)\}$, $R2 = \{(2,1), (2,3), (2,5)\}$, $R3 = \{(2,3), (2,5), (4,3), (4,5)\}$

• 원소 개수가 n인 집합 S의 부분집합의 개수: 2^n

- 카티전 프로덕트 $A \times B$ 의 부분집합의 개수:

$$2^{|A| \times |B|}$$

1.1 관계의 수학적 의미

- 카티전 프로덕트의 기초 집합 A, B
- 각각이 가질 수 있는 값의 범위: 도메인 (domain)
 - 집합 A의 도메인은 {2, 4}
- 릴레이션 역시 집합
 - 가능한 연산
 - 합집합(\cup), 교집합(\cap), 카티전 프로덕트(\times)
 - $R1 \cup R2 = \{(2, 1), (4, 1), (2, 3), (2, 5)\}$
 - $R1 \cap R2 = \{(2, 1)\}$

1.1 관계의 연산 세계 적용

- 예: 학번={2, 4}, 과목={데이터베이스, 자료구조, 프로그래밍}
- 두 집합의 카티전 프로덕트
 - 학번 \times 과목은 학번 원소와 과목 원소의 순서쌍의 집합
 - 즉, 학번 \times 과목 = {(2, 데이터베이스), (2, 자료구조), (2, 프로그래밍), (4, 데이터베이스), (4, 자료구조), (4, 프로그래밍)}
- 학번 \times 과목의 각 원소
 - 학생이 과목을 수강할 수 있는 모든 경우
 - 수강 = {(2, 데이터베이스), (2, 자료구조), (4, 프로그래밍)}
 - 카티전 프로덕트 학번 \times 과목의 부분집합
 - 수강 집합 안 원소는 하나의 릴레이션 인스턴스

1.1 관계의 현실 세계 적용

- 수강 릴레이션의 투플은 위에서 나열한 여섯 개 원소 중 하나
- 아래 수강 테이블을 데이터베이스에서는 릴레이션(relation)이라고 함
수강 릴레이션

수강

학번	과목
2	데이터베이스
2	자료구조
4	프로그래밍

1.2 관계대수 연산자

- 기본 연산
 - 일반 집합 연산자
 - 합집합(UNION, \cup)
 - 교집합(INTERSECT, \cap)
 - 차집합(DIFFERENCE, $-$)
 - 카티션 프로덕트(CARTESIAN PRODUCT, \times)
 - 순수 관계 연산자
 - 선택(SELECT, σ)
 - 프로젝션(PROJECT, π)
 - 조인(JOIN, \bowtie)
 - 디비전(DIVISION, \div)
- 폐쇄 성질 (closure property)
 - 피연산자와 그들의 연산 결과가 모두 릴레이션
- 중첩(nested)된 수식의 표현이 가능

1.2 관계대수 연산자

연산자 종류	대상	연산자 이름	기호	설명
기본	단항	선택연	σ	릴레이션에서 조건에 만족하는 튜플을 선택
기본	단항	프로젝션	π	릴레이션의 속성을 선택
추가	단항	개명	ρ	릴레이션이나 속성의 이름을 변경
유도	이항	디비전	\div	분모 릴레이션에 포함된 튜플의 값을 모두 갖고 있는 튜플을 분자 릴레이션에서 추출
기본	이항	합집합	\cup	두 릴레이션의 합집합
기본	이항	차집합	$-$	두 릴레이션의 차집합
유도	이항	교집합	\cap	두 릴레이션의 교집합

2023-04-24

컴퓨터공학과

79

1.2 관계대수 연산자

연산자 종류	대상	연산자 이름	기호	설명
기본	이항	캐디전 프로덕트	\times	두 릴레이션에 속한 모든 튜플의 교차집합
유도	이항	조인	세타	\bowtie_{θ} 두 릴레이션 간의 비교 조건에 만족하는 집합
			동등	\bowtie 두 릴레이션 간의 같은 값을 가진 집합
			자연	\bowtie_N 동등 조인에서 중복 속성을 제거
			세미	left \ltimes 자연 조인 후 오른쪽 속성을 제거
				right \rtimes 자연 조인 후 왼쪽 속성을 제거
			외부	left $\ltimes\!\!\!\bowtie$ 자연 조인 후 각각 왼쪽(left), 오른쪽(right), 양쪽(full)의 모든 값을 결과로 추출
				right $\rtimes\!\!\!\bowtie$ 자연 조인 후 각각 왼쪽(left), 오른쪽(right), 양쪽(full)의 모든 값을 결과로 추출
			full	$\ltimes\!\!\!\bowtie\!\!\!\rtimes$ 자연 조인 후 각각 왼쪽(left), 오른쪽(right), 양쪽(full)의 모든 값을 결과로 추출

2023-04-24

컴퓨터공학과

80

1.3 관계대수식

- 릴레이션 간 연산을 통해 결과 릴레이션을 찾는 절차 기술 언어
 - 관계대수식(relational algebra expression): 연산 수행식
- 구성: 대상이 되는 릴레이션, 연산자 / 결과: 릴레이션으로 반환
 - 반환된 릴레이션은 릴레이션의 모든 특징을 따름
- 관계대수식 형태
 - 단항 연산자 : 연산자 <조건> 릴레이션
 - 이항 연산자 : 릴레이션1 연산자 <조건> 릴레이션2

R1			R2		
A	B	C	A	B	C
a1	b1	c1	a1	b1	c1
a2	b3	c3	a2	b3	c3
a3	b4	c2	a3	b3	c1

관계대수식을 이해하기 위한 예제 데이터

관계대수식 사용 예

R1	A	B	C	R2	A	B	C
	a1	b1	c1		a1	b1	c1
	a2	b3	c3		a2	b3	c3
	a3	b4	c2		a3	b3	c1

주요 연산자	사용 예	결과	설명									
선택연(σ)	$\sigma_{A=a1 \text{ or } A=a2}(R1)$	<table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>a1</td><td>b1</td><td>c1</td></tr><tr><td>a2</td><td>b3</td><td>c3</td></tr></table>	A	B	C	a1	b1	c1	a2	b3	c3	R1에서 조건에 맞는 튜플을 추출함
A	B	C										
a1	b1	c1										
a2	b3	c3										
프로젝션(π)	$\pi_{A,B}(R2)$	<table><tr><th>A</th><th>B</th></tr><tr><td>a1</td><td>b1</td></tr><tr><td>a2</td><td>b3</td></tr><tr><td>a3</td><td>b3</td></tr></table>	A	B	a1	b1	a2	b3	a3	b3	R2에서 조건에 맞는 속성만을 추출함	
A	B											
a1	b1											
a2	b3											
a3	b3											

일반 집합 연산자-Note

• 합병가능(union-compatible)한 릴레이션

- \cup , \cap , $-$ 연산의 피연산자들은
 1. 차수가 같아야 함
 2. 대응 속성 쌍 별로 도메인이 같아야 함

• \cup , \cap , \times 연산은 결합적(associative)임

- $R \cup S \cup T = (R \cup S) \cup T = R \cup (S \cup T)$
- $R \cap S \cap T = (R \cap S) \cap T = R \cap (S \cap T)$
- $R \times S \times T = (R \times S) \times T = R \times (S \times T)$

• \cup , \cap , \times 연산은 교환적(commutative)임

- $R \cup S = S \cup R$, $R \cap S = S \cap R$, $R \times S = S \times R$

관계대수식 사용 예

R1			R2		
A	B	C	A	B	C
a1	b1	c1	a1	b1	c1
a2	b3	c3	a2	b3	c3
a3	b4	c2	a3	b3	c1

주요 연산자	사용 예	결과	설명															
합집합(\cup)	$R1 \cup R2$	<table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>a1</td><td>b1</td><td>c1</td></tr><tr><td>a2</td><td>b3</td><td>c3</td></tr><tr><td>a3</td><td>b4</td><td>c2</td></tr><tr><td>a3</td><td>b3</td><td>c1</td></tr></table>	A	B	C	a1	b1	c1	a2	b3	c3	a3	b4	c2	a3	b3	c1	R1과 R2의 합집합을 구함
A	B	C																
a1	b1	c1																
a2	b3	c3																
a3	b4	c2																
a3	b3	c1																
차집합($-$)	$R1 - R2$	<table><tr><th>A</th><th>B</th><th>C</th></tr><tr><td>a3</td><td>b4</td><td>c2</td></tr></table>	A	B	C	a3	b4	c2	R1과 R2의 차집합을 구함									
A	B	C																
a3	b4	c2																

관계대수식 사용 예

R1

A	B	C
a1	b1	c1
a2	b3	c3
a3	b4	c2

R2

A	B	C
a1	b1	c1
a2	b3	c3
a3	b3	c1

주요 연산자	사용 예	결과	설명																								
조인	$R1 \bowtie_{R1.C=R2.C} R2$	<table><tr><th>R1.A</th><th>R1.B</th><th>R1.C</th><th>R2.A</th><th>R2.B</th><th>R2.C</th></tr><tr><td>a1</td><td>b1</td><td>c1</td><td>a1</td><td>b1</td><td>c1</td></tr><tr><td>a1</td><td>b1</td><td>c1</td><td>a3</td><td>b3</td><td>c1</td></tr><tr><td>a2</td><td>b3</td><td>c3</td><td>a2</td><td>b3</td><td>c3</td></tr></table>	R1.A	R1.B	R1.C	R2.A	R2.B	R2.C	a1	b1	c1	a1	b1	c1	a1	b1	c1	a3	b3	c1	a2	b3	c3	a2	b3	c3	R1과 R2의 카티잔 프로덕트를 구하여 조건에 맞는 튜플을 추출함
R1.A	R1.B	R1.C	R2.A	R2.B	R2.C																						
a1	b1	c1	a1	b1	c1																						
a1	b1	c1	a3	b3	c1																						
a2	b3	c3	a2	b3	c3																						

2023-04-24

컴퓨터공학과

85

2.1 셀렉션(selection)

- 릴레이션의 튜플 추출 연산
- 하나의 릴레이션을 대상으로 하는 단항 연산자
- 찾고자 하는 튜플을 조건(predicate)을 명시하고 그 조건에 만족하는 튜플을 반환함
- 형식 : $\sigma_{\langle \text{조건} \rangle} (R)$ (R은 릴레이션)

2023-04-24

컴퓨터공학과

86

2.1 셀렉션(selection)

- A, B가 릴레이션 R의 속성일 때,
 - $\sigma_{A\theta v}(R) = \{ r \mid r \in R \wedge r.A \theta v \}$
 - $\sigma_{A\theta B}(R) = \{ r \mid r \in R \wedge r.A \theta r.B \}$
 - θ (theta) : 비교 연산자, $\{ <, >, \leq, \geq, =, \neq \}$
 - v: 상수, r: 튜플
- 셀렉트(σ)의 결과
 - 선택 조건을 만족하는 릴레이션의 수평적 부분집합
 - **Horizontal subset**

2023-04-24

컴퓨터공학과

87

2.1 셀렉션(selection)

질의 2-1 마당서점에서 판매하는 도서 중 8,000원 이하인 도서를 검색하시오.

$\sigma_{\text{가격} \leq 8000}(\text{도서})$

도서

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구는 여자	나무수	13000
3	축구의 이해	대안미디어	22000
4	골프 바이블	대안미디어	35000
5	피겨 교본	굿스포츠	8000



도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
5	피겨 교본	굿스포츠	8000

2023-04-24

컴퓨터공학과

88

2.2 프로젝션(projection)

- PROJECT: II: π
- 릴레이션의 속성 추출 연산
- 단항 연산자
- 형식 : $\pi \langle \text{속성리스트} \rangle (R)$
- 릴레이션의 수직적 부분집합(vertical subset)
- 결과에 투플이 중복되는 경우에는 제거
 - 결과도 투플의 집합(릴레이션) – 폐쇄성(closure)

2023-04-24

컴퓨터공학과

89

2.2 프로젝션(projection)

질의 2-2 신간도서 안내를 위해 고객의 (이름, 주소, 핸드폰)이 적힌 카탈로그 주소록을 만드시오.

π 이름, 주소, 핸드폰 (고객)

고객번호	이름	주민번호	주소	핸드폰
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001
2	김연아	900905-2222222	대한민국 서울	000-6000-0001
3	강미란	831009-2333333	대한민국 강원도	000-7000-0001
4	주인수	820713-1444444	미국 클리블랜드	000-8000-0001

이름	주소	핸드폰
박지성	영국 맨체스터	000-5000-0001
김연아	대한민국 서울	000-6000-0001
강미란	대한민국 강원도	000-7000-0001
주인수	미국 클리블랜드	000-8000-0001

2023-04-24

컴퓨터공학과

90

3.1 합집합

- 두 개의 릴레이션을 합하여 하나의 릴레이션을 반환
 - 두 개의 릴레이션은 서로 같은 속성 순서와 도메인을 가져야 함
- 연식 : $R \cup S$

질의 2-3 마당서점은 지점A와 지점B가 있다. 두 지점의 도서는 각 지점에서 관리하며 릴레이션 이름은 각각 도서A, 도서B다. 마당서점의 도서를 하나의 릴레이션으로 보이시오.

도서A \cup 도서B

3.1 합집합

도서A

도서번호	도서이름	출판사	가격
1	축구와 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대안미디어	22000

\cup

도서B

도서번호	도서이름	출판사	가격
1	축구와 역사	굿스포츠	7000
4	골프 바이블	대안미디어	35000
5	피겨 교본	굿스포츠	8000



도서번호	도서이름	출판사	가격
1	축구와 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대안미디어	22000
4	골프 바이블	대안미디어	35000
5	피겨 교본	굿스포츠	8000

3.2 교집합

- **입력 가능한 두 릴레이션을 대상으로 함**
- 두 릴레이션이 공통으로 가지고 있는 튜플을 반환함
- **영식 : $R \cap S$**

질의 2-4 마당서점의 두 지점에서 동일하게 보유하고 있는 도서 목록을 보이시오.

도서A \cap 도서B

도서A

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대안미디어	22000

\cap

도서B

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
4	골프 바이블	대안미디어	35000
5	피겨 교본	굿스포츠	8000



도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000

2023-04-24

컴퓨터공학과

93

3.3 차집합

- 첫 번째 릴레이션에는 속하고 두 번째 릴레이션에는 속하지 않는 튜플 반환
- **영식 : $R - S$**

질의 2-5 마당서점 두 지점 중 지점 A에서만 보유하고 있는 도서 목록을 보이시오.

도서A - 도서B

도서A

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구 하는 여자	나무수	13000
3	축구의 이해	대안미디어	22000

-

도서B

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
4	골프 바이블	대안미디어	35000
5	피겨 교본	굿스포츠	8000



도서번호	도서이름	출판사	가격
2	축구 하는 여자	나무수	13000
3	축구의 이해	대안미디어	22000

2023-04-24

컴퓨터공학과

94

3.4 카티전 프로덕트(cartesian product)

- 두 릴레이션을 연결시켜 하나로 합칠 때 사용
- 결과 릴레이션은 첫 번째 릴레이션의 오른쪽에 두 번째 릴레이션의 모든 튜플을 순서대로 배열하여 반환함
- 결과 릴레이션의 차수: 두 릴레이션의 차수의 합
- 카디널리티: 두 릴레이션의 카디널리티의 곱

• **영식 : R** **질의 2-6** 고객 릴레이션과 주문 릴레이션의 카티전 프로덕트를 구하시오(결과가 많으므로 튜플을 일부 삭제한 릴레이션을 사용한다).

고객 × 주문

2023-04-24

컴퓨터공학과

95

3.4 카티전 프로덕트(cartesian product)

고객				주문				
고객번호	이름	주소	핸드폰	주문번호	고객번호	도시코드	판매가격	주문일자
1	박지영	영국 맨체스터	000-5000-0001	1	2	1	7000	2014-07-01
2	김연아	대한민국 서울	000-6000-0001	2	1	2	13000	2014-07-03
3	강미란	대한민국 강원도	000-7000-0001	3	2	5	8000	2014-07-03
				4	1	2	13000	2014-07-04

그림 2-23 카티전 프로덕트의 예

고객번호	이름	주소	핸드폰	주문번호	고객번호	도시코드	판매가격	주문일자
1	박지영	영국 맨체스터	000-5000-0001	1	2	1	7000	2014-07-01
1	박지영	영국 맨체스터	000-5000-0001	2	1	2	13000	2014-07-03
1	박지영	영국 맨체스터	000-5000-0001	3	2	5	8000	2014-07-03
1	박지영	영국 맨체스터	000-5000-0001	4	1	2	13000	2014-07-04
2	김연아	대한민국 서울	000-6000-0001	1	2	1	7000	2014-07-01
2	김연아	대한민국 서울	000-6000-0001	2	1	2	13000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	3	2	5	8000	2014-07-03
2	김연아	대한민국 서울	000-6000-0001	4	1	2	13000	2014-07-04
3	강미란	대한민국 강원도	000-7000-0001	1	2	1	7000	2014-07-01
3	강미란	대한민국 강원도	000-7000-0001	2	1	2	13000	2014-07-03
3	강미란	대한민국 강원도	000-7000-0001	3	2	5	8000	2014-07-03
3	강미란	대한민국 강원도	000-7000-0001	4	1	2	13000	2014-07-04

2023-04-24

컴퓨터공학과

96

4. 조인(join)

- 두 릴레이션의 공통 속성을 기준으로 속성 값이 같은 튜플을 **수평 결합**하는 연산
- 두 릴레이션의 조인에 참여하는 속성이 서로 동일한 **도메인**으로 구성
- 조인 연산의 결과: 공통 속성의 속성 값이 동일한 튜플 만을 반환
- 형식 : $R \bowtie_C S = \sigma_C(R \times S)$
 - R과 S는 릴레이션, C는 조인조건
- 조인 연산의 구분
 - 기본연산 : 세타조인(\bowtie_θ), 동등조인(\bowtie), 자연조인(\bowtie_N)
 - 확장된 조인 연산 : 세미조인(\ltimes, \bowtie), 외부조인($\bowtie \bowtie$)

4.1 세타조인과 동등조인

- 세타조인(theta join, θ)
 - 조인에 참여하는 두 릴레이션의 속성 값을 비교하여 조건을 만족하는 튜플만 반환함
 - 세타조인의 조건은 $\{=, \neq, \leq, \geq, <, >\}$ 중 하나가 됨
 - 형식 : $R \bowtie_{r \text{ 조건 } s} S$
 - R과 S는 릴레이션, r은 R의 속성, s는 S의 속성
- 동등조인(equi join)
 - 세타조인에서 = 연산자를 사용한 조인
 - 보통 조인 연산이라고 하면 동등조인을 지칭함
 - 형식 : $R \bowtie_{r=s} S$

질의 2-7 고객과 고객의 주문 사항을 모두 보이시오.

고객 \bowtie 고객.고객번호=주문.고객번호 주문

4.1 세타조인과 동등조인

고객				주문				
고객번호	이름	주소	핸드폰	주문번호	고객번호	도시코드	판매가격	주문일자
1	박지성	영국 맨체스터	000-5000-0001	1	2	1	7000	2014-07-01
2	김연아	대한민국 서울	000-6000-0001	2	1	2	13000	2014-07-03
3	강미란	대한민국 강원도	000-7000-0001	3	2	5	8000	2014-07-03
4	주인수	미국 클리블랜드	000-8000-0001	4	1	2	13000	2014-07-04
				5	4	4	35000	2014-07-05
				6	5	3	22000	2014-07-07
				7	4	3	22000	2014-07-07

그림 2-24 동등조인의 예

고객 \bowtie 고객.고객번호=주문.고객번호 주문

고객번호	이름	주소	핸드폰	주문번호	고객번호	도시코드	판매가격	주문일자
1	박지성	영국 맨체스터	000-5000-0001	2	1	2	13000	2014-07-03
1	박지성	영국 맨체스터	000-5000-0001	4	1	2	13000	2014-07-04
2	김연아	대한민국 서울	000-6000-0001	1	2	1	7000	2014-07-01
2	김연아	대한민국 서울	000-6000-0001	3	2	5	8000	2014-07-03
4	주인수	미국 클리블랜드	000-8000-0001	5	4	4	35000	2014-07-05
4	주인수	미국 클리블랜드	000-8000-0001	7	4	3	22000	2014-07-07

4.2 자연조인(natural join)

- 동등조인에서 조인에 참여한 속성이 두 번 나오지 않도록 두 번째 속성을 제거한 결과를 반환함
- 형식 : $R \bowtie_{N(r,s)} S$

질의 2-8 고객과 고객의 주문 사항을 모두 보여주되 같은 속성은 한 번만 표시하시오.

고객 $\bowtie_{N(\text{고객.고객번호}, \text{주문.고객번호})}$ 주문

4.2 자연조인(natural join)

고객				주문				
고객번호	이름	주소	핸드폰	주문번호	고객번호	도시코드	판매가격	주문일자
1	박지성	영국 맨체스터	000-5000-0001	1	2	1	7000	2014-07-01
2	김연아	대한민국 서울	000-6000-0001	2	1	2	13000	2014-07-03
3	강미란	대한민국 강원도	000-7000-0001	3	2	5	8000	2014-07-03
4	주인수	미국 클리블랜드	000-8000-0001	4	1	2	13000	2014-07-04
				5	4	4	35000	2014-07-05
				6	5	3	22000	2014-07-07
				7	4	3	22000	2014-07-07

그림 2-25 자연조인의 예

고객 ⋈ (고객, 고객번호-주문, 고객번호) 주문

고객번호	이름	주소	핸드폰	주문번호	도시코드	판매가격	주문일자
1	박지성	영국 맨체스터	000-5000-0001	2	2	13000	2014-07-03
1	박지성	영국 맨체스터	000-5000-0001	4	2	13000	2014-07-04
2	김연아	대한민국 서울	000-6000-0001	1	1	7000	2014-07-01
2	김연아	대한민국 서울	000-6000-0001	3	5	8000	2014-07-03
4	주인수	미국 클리블랜드	000-8000-0001	5	4	35000	2014-07-05
4	주인수	미국 클리블랜드	000-8000-0001	7	3	22000	2014-07-07

2023-04-24

컴퓨터공학과

101

자연 조인 예

Employee			Dept	
Name	EmpId	DeptName	DeptName	Manager
Harry	3415	Finance	Finance	George
Sally	2241	Sales	Sales	Harriet
George	3401	Finance	Production	Charles
Harriet	2202	Sales		

One common attribute

Employee ⋈ Dept

Name	EmpId	DeptName	Manager
Harry	3415	Finance	George
Sally	2241	Sales	Harriet
George	3401	Finance	George
Harriet	2202	Sales	Harriet

2023-04-24

컴퓨터공학과

102

4.3 외부조인과 세미조인

• 외부조인(outer join)

- 자연조인 시 조인에 실패한 튜플을 모두 보여주되 값이 없는 대응 속성에는 **NULL** 값을 채워서 반환
- 모든 속성을 보여주는 기준 릴레이션 위치에 따라
 - 왼쪽(left) 외부조인, 오른쪽(right) 외부조인, 완전(full) 외부조인으로 분류
- 형식 :
 - 왼쪽(left) 외부조인 – R $(r, s) \bowtie^L C$
 - 완전(full) 외부조인 – R $(r, s) \bowtie^F C$
 - 오른쪽(right) 외부조인 – R $(r, \bowtie^R S$

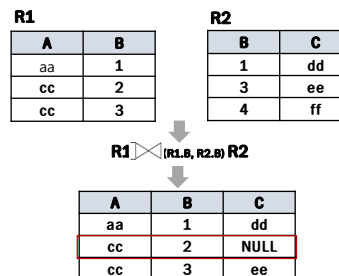
4.3 외부조인과 세미조인

질의 2-9 마당세장의 고객과 고객의 주문 내역을 보이시오.

- ① 고객 기준으로 주문내역이 없는 고객도 모두 보이시오.
- ② 주문내역이 없는 고객과, 고객 릴레이션에 고객번호가 없는 주문을 모두 보이시오.
- ③ 주문내역 기준으로 고객 릴레이션에 고객번호가 없는 주문도 모두 보이시오.

- ① 고객 \bowtie^L (고객,고객번호, 주문,고객번호) 주문
- ② 고객 \bowtie^F (고객,고객번호, 주문,고객번호) 주문
- ③ 고객 \bowtie^R (고객,고객번호, 주문,고객번호) 주문

그림 2-26 왼쪽 외부조인의 예



4.3 외부조인과 세미조인



그림 2-27 외부조인의 예
컴퓨터공학과

2023-04-24

105

4.3 외부조인과 세미조인

• 세미조인(semi join)

- 자연조인을 한 후 두 릴레이션 중 안쪽 릴레이션의 결과만 반환하며, 기호에서 달인 쪽 릴레이션의 튜플만 반환함
- 형식 : $R \ltimes_{(r,s)} S$

질의 2-10 마당서점의 고객 중 주문 내역이 있는 고객의 고객 정보를 보이시오.

고객 $\ltimes_{(\text{고객.고객번호}, \text{주문.고객번호})}$ 주문

2023-04-24

컴퓨터공학과

106

4.3 외부조인과 세미조인

고객

고객번호	이름	주소	핸드폰
1	박지성	영국 맨체스터	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	주인수	미국 클리블랜드	000-8000-0001

주문

주문번호	고객번호	도/세번호	판매가격	주문일자
1	2	1	7000	2014-07-01
2	1	2	13000	2014-07-03
3	2	5	8000	2014-07-03
4	1	2	13000	2014-07-04
5	4	4	35000	2014-07-05
6	5	3	22000	2014-07-07
7	4	3	22000	2014-07-07



고객 ⋈ (고객, 고객번호, 주문, 고객번호) 주문

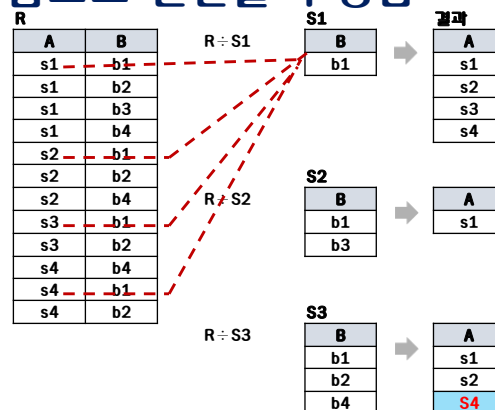
고객번호	이름	주소	핸드폰
1	박지성	영국 맨체스터	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
4	주인수	미국 클리블랜드	000-8000-0001

그림 2-28 세미조인(왼쪽이 닫힌 경우)의 예

5. 디비전(division)

- 릴레이션의 속성 값의 집합으로 연산을 수행함
- 형식 : $R \div S$

그림 2-29 디비전 연산의 예



6.1 선택, 프로젝션, 집합연산의 복합 사용

• 마당서점의 지점이 하나인 경우: π 도서이름, 출판사 (σ 가격 < = 8000 도서)

질의 2-11 마당서점의 도서 중 가격이 8,000원 이하인 도서이름과 출판사를 보이시오.

도서

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대안미디어	22000
4	골프 바이블	대안미디어	35000
5	피겨 교본	굿스포츠	8000



σ 가격 < = 8000 (도서)

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
5	피겨 교본	굿스포츠	8000

π 도서이름, 출판사

도서이름	출판사
축구의 역사	굿스포츠
피겨 교본	굿스포츠

단일 릴레이션에서 선택, 프로젝션 연산의 복합 사용

2023-04-24

컴퓨터공학과

109

6.1 선택, 프로젝션, 집합연산의 복합 사용

마당서점의 지점이 둘 이상인 경우: π 도서이름, 출판사 ((σ 가격 < = 8000 도서A) \cup (σ 가격 < = 8000 도서B))

도서A

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
2	축구하는 여자	나무수	13000
3	축구의 이해	대안미디어	22000

σ 가격 < = 8000 (도서A)

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000

도서B

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
4	골프 바이블	대안미디어	35000
5	피겨 교본	굿스포츠	8000

σ 가격 < = 8000 (도서B)

도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
5	피겨 교본	굿스포츠	8000

\cup



도서번호	도서이름	출판사	가격
1	축구의 역사	굿스포츠	7000
5	피겨 교본	굿스포츠	8000



π 도서이름, 출판사

도서이름	출판사
축구의 역사	굿스포츠
피겨 교본	굿스포츠

2023-04-24

컴퓨터공학과

110

6.2 카티전 프로덕트 사용 연산과 조인 사용 연산

• 카티전 프로덕트를 사용한 연산

π 주문.주문번호, 고객.이름, 주문.판매가격 (σ 고객.고객번호=주문.고객번호 AND 고객.이름= '박지성' (고객 \times 주문))

질의 2-12 마당서점의 박지성 고객의 거래 내역 중 주문번호, 이름, 가격을 보이시오.

2023-04-24

컴퓨터공학과

111

고객번호	이름	주문번호	주소	핸드폰	주문번호	고객번호	도서번호	판매가격	주문일자
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001	1	1	1	7000	2014-07-01
2	김연애	900905-2222222	대한민국 서울	000-6000-0001	2	1	2	13000	2014-07-03
3	정미란	831009-2333333	대한민국 강원도	000-7000-0001	3	2	5	8000	2014-07-03
4	우연수	820713-1444444	미국 캘리포니아	000-8000-0001	4	3	2	13000	2014-07-04

고객 \times 주문

고객번호	이름	주문번호	주소	핸드폰	주문번호	고객번호	도서번호	판매가격	주문일자
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001	1	1	1	7000	2014-07-01
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001	2	1	2	13000	2014-07-03
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001	3	2	5	8000	2014-07-03
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001	4	3	2	13000	2014-07-04
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001	5	4	4	35000	2014-07-05
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001	6	1	3	22000	2014-07-07
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001	7	4	3	22000	2014-07-07
2	김연애	900905-2222222	대한민국 서울	000-6000-0001	1	1	1	7000	2014-07-01
4	우연수	820713-1444444	미국 캘리포니아	000-8000-0001	5	4	4	35000	2014-07-05
4	우연수	820713-1444444	미국 캘리포니아	000-8000-0001	6	1	3	22000	2014-07-07
4	우연수	820713-1444444	미국 캘리포니아	000-8000-0001	7	4	3	22000	2014-07-07

σ 고객.고객번호=주문.고객번호 AND 고객.이름= '박지성'

고객번호	이름	주문번호	주소	핸드폰	주문번호	고객번호	도서번호	판매가격	주문일자
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001	1	1	1	7000	2014-07-01
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001	2	1	2	13000	2014-07-03
1	박지성	810225-1111111	영국 맨체스터	000-5000-0001	6	1	3	22000	2014-07-07

π 주문.주문번호, 고객.이름, 주문.판매가격

카티전 프로덕트를 사용한 연산

컴퓨터공학과

2023-04-24

112

주문번호	이름	판매가격
1	박지성	7000
2	박지성	13000
6	박지성	22000

6.2 카티전 프로덕트 사용 연산과 조인 사용 연산

• 조인 사용 연산

- π 주문번호, 이름, 판매가격 (σ 이름 = '박지성' (고객 \bowtie 고객.고객번호=주문.고객번호 주문))

• 카티전 프로덕트를 사용한 연산과 비교

- π 주문.주문번호, 고객.이름, 주문.판매가격 (σ 고객.고객번호=주문.고객번호 AND 고객.이름 = '박지성' (고객 \times 주문))

2023-04-24

컴퓨터공학과

113



그림 2-33 조인을 사용한 연산

2023-04-24

컴퓨터공학과

114

Notation

Operation	My HTML	Symbol	Operation	My HTML	Symbol
Projection	PROJECT	π	Cartesian product	X	\times
Selection	SELECT	σ	Join	JOIN	\bowtie
Renaming	RENAME	ρ	Left outer join	LEFT OUTER JOIN	\ltimes
Union	UNION	\cup	Right outer join	RIGHT OUTER JOIN	\rtimes
Intersection	INTERSECTION	\cap	Full outer join	FULL OUTER JOIN	$\ltimes\rtimes$
Assignment	\leftarrow	\leftarrow	Semijoin	SEMIJOIN	\ltimes

Example: The relational algebra expression which I would here write as

`PROJECTNam (SELECTMedlemsnummer < 3 (Medlem))`

should actually be written

$\pi_{\text{Nam}} (\sigma_{\text{Medlemsnummer} < 3} (\text{Medlem}))$

2023-04-24

컴퓨터공학과

115

Projection

Example: The table **E** (for **EMPLOYEE**)

nr	name	salary
1	John	100
5	Sarah	300
7	Tom	100

SQL	Result	Relational algebra								
select salary from E	<table><tr><th>salary</th></tr><tr><td>100</td></tr><tr><td>300</td></tr></table>	salary	100	300	$\text{PROJECT}_{\text{salary}}(E)$					
salary										
100										
300										
select nr, salary from E	<table><tr><th>nr</th><th>salary</th></tr><tr><td>1</td><td>100</td></tr><tr><td>5</td><td>300</td></tr><tr><td>7</td><td>100</td></tr></table>	nr	salary	1	100	5	300	7	100	$\text{PROJECT}_{\text{nr, salary}}(E)$
nr	salary									
1	100									
5	300									
7	100									

Note that there are **no duplicate** rows in the result.

2023-04-24

컴퓨터공학과

116

Selection

Example: The table **E** (for **EMPLOYEE**)

nr	name	salary
1	John	100
5	Sarah	300
7	Tom	100

Selection

The same table **E** (for **EMPLOYEE**) as above.

SQL	Result	Relational algebra									
select * from E where salary < 200	<table> <tr><th>nr</th><th>name</th><th>salary</th></tr> <tr><td>1</td><td>John</td><td>100</td></tr> <tr><td>7</td><td>Tom</td><td>100</td></tr> </table>	nr	name	salary	1	John	100	7	Tom	100	SELECT salary < 200(E)
nr	name	salary									
1	John	100									
7	Tom	100									
select * from E where salary < 200 and nr >= 7	<table> <tr><th>nr</th><th>name</th><th>salary</th></tr> <tr><td>7</td><td>Tom</td><td>100</td></tr> </table>	nr	name	salary	7	Tom	100	SELECT salary < 200 and nr >= 7(E)			
nr	name	salary									
7	Tom	100									

Note!!

- SQL keyword **select**와 무관.
- 조건을 충족하는 릴레이션에서 해당 튜플을 반환

Relational Algebra Expressions

Example: The table **E** (for **EMPLOYEE**)

nr	name	salary
1	John	100
5	Sarah	300
7	Tom	100

SQL	Result	Relational algebra						
<pre>select name, salary from E where salary < 200</pre>	<table><tr><th>name</th><th>salary</th></tr><tr><td>John</td><td>100</td></tr><tr><td>Tom</td><td>100</td></tr></table>	name	salary	John	100	Tom	100	<p>$\text{PROJECT}_{\text{name, salary}} (\text{SELECT}_{\text{salary} < 200}(\text{E}))$</p> <p><i>or, step by step, using an intermediate result</i></p> <p>Temp <- $\text{SELECT}_{\text{salary} < 200}(\text{E})$ Result <- $\text{PROJECT}_{\text{name, salary}}(\text{Temp})$</p>
name	salary							
John	100							
Tom	100							

Renaming tables and columns

Example: The table **E** (for **EMPLOYEE**)

nr	name	dept
1	Bill	A
2	Sarah	C
3	John	A

Example: The table **D** (for **DEPARTMENT**)

nr	name
A	Marketing
B	Sales
C	Legal

SQL	Result	Relational algebra																				
<pre>select * from E as E(enr, ename, dept), D as D(dnr, dname) where dept = dnr</pre>	<table><tr><th>enr</th><th>ename</th><th>dept</th><th>dnr</th><th>dname</th></tr><tr><td>1</td><td>Bill</td><td>A</td><td>A</td><td>Marketing</td></tr><tr><td>2</td><td>Sarah</td><td>C</td><td>C</td><td>Legal</td></tr><tr><td>3</td><td>John</td><td>A</td><td>A</td><td>Marketing</td></tr></table>	enr	ename	dept	dnr	dname	1	Bill	A	A	Marketing	2	Sarah	C	C	Legal	3	John	A	A	Marketing	$(\text{RENAME}_{(\text{enr}, \text{ename}, \text{dept})}(\text{E})) \text{ JOIN}_{\text{dept} = \text{dnr}} (\text{RENAME}_{(\text{dnr}, \text{dname})}(\text{D}))$
enr	ename	dept	dnr	dname																		
1	Bill	A	A	Marketing																		
2	Sarah	C	C	Legal																		
3	John	A	A	Marketing																		
<pre>select * from E, D where dept = D.nr</pre>	<table><tr><th>nr</th><th>name</th><th>dept</th><th>nr</th><th>name</th></tr><tr><td>1</td><td>Bill</td><td>A</td><td>A</td><td>Marketing</td></tr><tr><td>2</td><td>Sarah</td><td>C</td><td>C</td><td>Legal</td></tr><tr><td>3</td><td>John</td><td>A</td><td>A</td><td>Marketing</td></tr></table>	nr	name	dept	nr	name	1	Bill	A	A	Marketing	2	Sarah	C	C	Legal	3	John	A	A	Marketing	$\text{E JOIN}_{\text{dept} = \text{D.nr}} \text{D}$
nr	name	dept	nr	name																		
1	Bill	A	A	Marketing																		
2	Sarah	C	C	Legal																		
3	John	A	A	Marketing																		

2023-04-24

컴퓨터공학과

119

Aggregate functions

Example: The table **E** (for **EMPLOYEE**)

nr	name	salary	dept
1	John	100	A
5	Sarah	300	C
7	Tom	100	A
12	Anne	null	C

Note:

Duplicates are **not** eliminated.

Null values are ignored.

SQL	Result	Relational algebra		
<pre>select sum(salary) from E</pre>	<table><tr><th>sum</th></tr><tr><td>500</td></tr></table>	sum	500	$F_{\text{sum(salary)}}(E)$
sum				
500				

2023-04-24

컴퓨터공학과

120

Aggregate functions

Employee

nr	name	salary	dept
1	John	100	A
5	Sarah	300	C
7	Tom	100	A
12	Anne	null	C

Note:

Count 함수는 NULL을 계산 안 함

SQL	Result	Relational algebra
select count(salary) from E	Result: count 3	$F_{\text{count(salary)}}(E)$
select count(distinct salary) from E	Result: count 2	$F_{\text{count(salary)}}(\text{PROJECT}_{\text{salary}}(E))$

2023-04-24

컴퓨터공학과

121

Aggregate functions

Employee

nr	name	salary	dept
1	John	100	A
5	Sarah	300	C
7	Tom	100	A
12	Anne	null	C

You can calculate aggregates "grouped by" something:

SQL	Result	Relational algebra
select sum(salary) from E group by dept	dept sum A 200 C 300	$F_{\text{sum(salary)}}(E)$

2023-04-24

컴퓨터공학과

122

Aggregate functions

Employee

nr	name	salary	dept
1	John	100	A
5	Sarah	300	C
7	Tom	100	A
12	Anne	null	C

Several aggregates simultaneously:

SQL	Result	Relational algebra									
<pre>select sum(salary), count(*) from E group by dept</pre>	<table> <tr> <th>dept</th><th>sum</th><th>count</th></tr> <tr> <td>A</td><td>200</td><td>2</td></tr> <tr> <td>C</td><td>300</td><td>1</td></tr> </table>	dept	sum	count	A	200	2	C	300	1	$\text{dept} \leftarrow \text{sum}(\text{salary}), \text{count}(\ast)(E)$
dept	sum	count									
A	200	2									
C	300	1									

Standard aggregate functions: sum, count, avg, min, max

Cartesian product

The *cartesian product* of two tables combines each row in one table with each row in the other table.

Example: The table E (for EMPLOYEE)

enr	ename	dept
1	Bill	A
2	Sarah	C
3	John	A

Example: The table D (for DEPARTMENT)

dnr	dname
A	Marketing
B	Sales
C	Legal

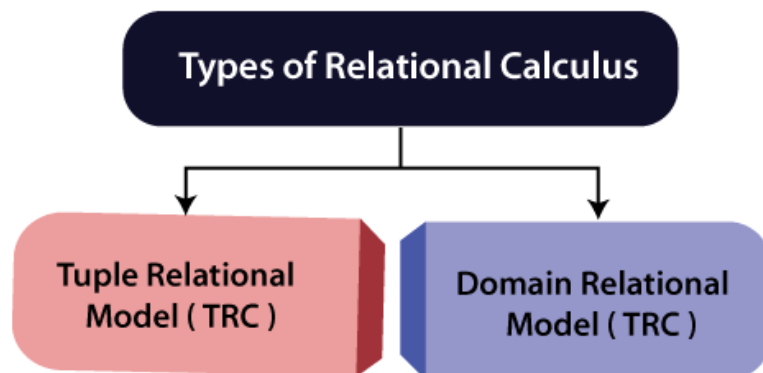
SQL	Result					Relational algebra
select * from E, D	enr	ename	dept	dnr	dname	E X D
	1	Bill	A	A	Marketing	
	1	Bill	A	B	Sales	
	1	Bill	A	C	Legal	
	2	Sarah	C	A	Marketing	
	2	Sarah	C	B	Sales	
	2	Sarah	C	C	Legal	
	3	John	A	A	Marketing	
	3	John	A	B	Sales	
3	John	A	C	Legal		

Join (called "inner join")

SQL	Result	Relational algebra																				
<pre>select * from E, D where dept = dnr</pre>	<table><tr><th>enr</th><th>ename</th><th>dept</th><th>dnr</th><th>dname</th></tr><tr><td>1</td><td>Bill</td><td>A</td><td>A</td><td>Marketing</td></tr><tr><td>2</td><td>Sarah</td><td>C</td><td>C</td><td>Legal</td></tr><tr><td>3</td><td>John</td><td>A</td><td>A</td><td>Marketing</td></tr></table>	enr	ename	dept	dnr	dname	1	Bill	A	A	Marketing	2	Sarah	C	C	Legal	3	John	A	A	Marketing	<p>$\text{SELECT}_{\text{dept} = \text{dnr}} (\text{E} \times \text{D})$</p> <p><i>or, using the equivalent join operation</i></p> <p>$\text{E JOIN}_{\text{dept} = \text{dnr}} \text{D}$</p>
enr	ename	dept	dnr	dname																		
1	Bill	A	A	Marketing																		
2	Sarah	C	C	Legal																		
3	John	A	A	Marketing																		

04 관계예식

Relational Calculus



04 관계해석

- 관계 해석
 - “어떻게 검색할 것인가” 보다 “무엇을 검색할 것인가” 만을 기술하는 선언적 표현법을 사용하는 비절차적 질의어
 - SQL을 포함한 많은 상업용 관계 언어들이 관계 해석에 기반함
 - 종류
 - 튜플 관계 해석(tuple relational calculus)
 - 도메인 관계 해석(domain relational calculus)
- 관계 대수와의 차이점
 - 선언적(declarative) 해석식으로 검색 질의를 명시
 - 비 절차적인 언어
 - 관계 대수는 연산들을 순차적으로 사용, 절차적
 - 두 언어의 표현력(expressive power)은 동등함

2023-04-24

컴퓨터공학과

127

04 튜플관계해석: TRC

- 튜플 변수
 - 릴레이션의 튜플들을 범위(range)로 가지는 변수
 - 예제: 봉급이 \$50,000를 넘는 모든 사원을 검색하라.
 - $\{t \mid \text{EMPLOYEE}(t) \text{ and } t.\text{SALARY} > 50000\}$
 - 여기서, EMPLOYEE(t)는 튜플 변수 t가 릴레이션 EMPLOYEE의 튜플들을 범위로 함
 - 튜플 t에 대하여 t.SALARY > 50000을 만족하는 튜플 만이 검색됨
 - 튜플 t의 모든 속성 값들이 반환됨

2023-04-24

컴퓨터공학과

128

04 튜플관계예식: TRC

• 프로젝션의 표현

- t 의 일부 속성 만을 위한 검색식
 - $\{t.FNAME, t.LNAME \mid EMPLOYEE(t) \text{ and } t.SALARY > 50000\}$
- 이는 다음 SQL 질의와 동일한 의미를 가진다.

```
SELECT      T.FNAME, T.LNAME
FROM        EMPLOYEE T           //T로 EMPLOYEE를 개명
WHERE       T.SALARY > 50000;
```

04 관계예식: TRC

• 정량자(quantifiers)를 식에 사용 가능

- 전체 정량자(universal quantifier) (\forall) (for all이라 읽음)
- 존재 정량자(existential quantifier) (\exists) (there exists라 읽음)

• 존재 정량자

- F 가 참이 되게 하는 어떤 튜플 t 가 “존재” 하면 $(\exists t)(F)$ 가 참이므로, \exists 를 **존재정량자**라 부름

• 전체 정량자

- “모든” 튜플들이 F 를 참이 되도록 해야 $(\forall t)(F)$ 가 참이므로, \forall 를 **전체 정량자**라 부름

04 관계해석: TRC 예

- 질의 1: Research 부서에서 일하는 모든 사원의 이름과 주소를 검색하라.
- Q1: $\{t.FNAME, t.LNAME, t.ADDRESS \mid EMPLOYEE(t) \text{ and } (\exists d) (DEPARTMENT(d) \text{ and } d.DNAME = \text{'Research'} \text{ and } d.DNUMBER = t.DNO))\}$
 - 막대 (|) 왼쪽: 관계 해석 식에서 **자유** 튜플 변수들만 표시
 - 막대 (|)는 “such that” 이라 읽음
 - $EMPLOYEE(t)$, $DEPARTMENT(d)$ 는 t 와 d 의 범위 릴레이션을 명시
 - $d.DNAME = \text{'Research'}$ 는 선택 조건(selection condition)
 - 관계 대수의 SELECT에 해당
 - $d.DNUMBER = t.DNO$ 는 조인 조건(join condition)
 - 관계 대수의 EQUI-JOIN과 유사한 목적으로 사용

04 도메인관계해석: DRC

- 튜플 변수 대신 **도메인 변수(domain variables)**를 사용하는 관계 해석
- 도메인 변수는 **한 속성 도메인을 범위로 가짐**
- 차수가 n 인 릴레이전의 경우 n 개의 도메인 변수를 사용함

04 도메인관계예석: DRC

• 예제:

- 질의: 이름이 'John B. Smith' 인 사원의 생일과 주소를 검색하라.

Q0: {uv | ($\exists q$) ($\exists r$) ($\exists s$) (EMPLOYEE(qrstuvwxyz)

and q = 'John' and r = 'B' and s = 'Smith')}

- EMPLOYEE의 각 속성들을 위한 열 개의 도메인 변수들: rstuvwxyz
- BDATE를 위한 변수 u, ADDRESS를 위한 v
- 조건에 참여하는 변수들 q(FNAME), r(MINIT), s(LNAME)
- 조건에 참여하는 변수들 (q, r, s)만 존재 정량자로 속박함

• 또 다른 표기법(QBE에서 사용):

- Q0' : {uv | EMPLOYEE('John' , 'B' , 'Smith' ,t,u,v,w,x,y,z)}

05 QBE

- A query method implemented in most database systems
- A graphical query language
 - Users can input commands into a table like conditions and example elements
- 사용자는 예제 테이블을 만들어 쿼리를 작성
- DRC에서와 같이 도메인 변수를 사용하여 예제 테이블을 생성

05 QBE

- 변수의 영역은 변수가 나타나는 열에 의해 결정
- 변수 기호에는 상수와 구별하기 위해 밑줄(_)이 접두사로 붙음
- 명령 P.는 인색을 의미
 - SQL 질의에서 SELECT 질의 대상 목록과 유사
- 다음 스키마를 고려
 - Sailors(sid: integer, *sname*: string, *rating*: integer, *age*: real)
 - Boats(bid: integer, *bname*: string, *color*: string)
 - Reserves(sid: integer, bid: integer, *day*: dates)

2023-04-24

컴퓨터공학과

135

05 QBE: 예제

- 모든 선원(sailor):

<i>Sailors</i>	<i>sid</i>	<i>sname</i>	<i>rating</i>	<i>age</i>
		P. <i>N</i>		P. <i>A</i>
- DRC query
 - $\{ \langle N, A \rangle \mid \exists I, T (\langle I, N, T, A \rangle \in \text{Sailors}) \}$
- 선원의 모든 필드

<i>Sailors</i>	<i>sid</i>	<i>sname</i>	<i>rating</i>	<i>age</i>
P. <i></i>				
- Select * From Sailors

2023-04-24

컴퓨터공학과

136

05 QBE: 예제

• Rating이 10인 선원

Sailors	sid	sname	rating	age
P.			10	

• DRC

• $\{ \langle I, N, 10, A \rangle \mid \langle I, N, 10, A \rangle \in \text{Sailors} \}$

• 1996년 8월 24일에 배를 예약하고 25세 이상 된 선원들을 찾아라.

Sailors	sid	sname	rating	age	Reserves	sid	bid	day
	Id	P..S		> 25		Id		'8/24/96'

2023-04-24

컴퓨터공학과

137

연습문제 풀이

6. 릴레이션에서 특정 속성에 해당하는 열을 선택하는 데 사용하며, 릴레이션의 수직적 부분 집합을 반환하는 관계대수 연산자는?

- ① projection ② join
③ division ④ selection

7. 릴레이션 C가 릴레이션 A와 B를 자연조인한 결과일 때 다음 중 맞는 설명을 모두 고르시오.

- ① C의 카디널리티는 A의 카디널리티보다 많다.
② C의 카디널리티는 A의 카디널리티보다 적다.
③ C의 차수는 A의 차수보다 많다.
④ C의 차수는 A의 차수보다 적다.
⑤ 모두 틀리다.

2023-04-24

컴퓨터공학과

138

연습문제 풀이

12. 다음 릴레이션에서 관계대수식의 결과를 작성하시오.

(1) $\sigma_{A=a2}(R)$

(2) $\pi_{A,B}(R)$

(3) $R \bowtie_{R.C=S.C} S$

R

A	B	C
a1	b1	c1
a2	b1	c1
a3	b1	c2
a4	b2	c3

S

C	D	E
c1	d2	e1
c1	d1	e2
c2	d3	e3
c3	d3	e3

13. 다음 수강신청 관련 릴레이션에 대한 질의문을 관계대수식으로 표현하시오.

학생(학번, 이름, 전공, 학년)

수강(과목코드, 학번, 수강학기, 성적)

과목(과목코드, 과목이름, 강의실, 요일, 담당교수)

- (1) 과목코드가 1234이고, 성적이 A인 모든 학생의 학번을 보이시오.
- (2) 과목코드가 1234인 과목을 등록한 학생의 이름과 전공을 보이시오.
- (3) 모든 과목에 등록된 학생의 이름을 보이시오.
- (4) 과목 1234에 등록하지 않은 학생의 이름을 보이시오.

2023-04-24

컴퓨터공학과

139

연습문제 풀이

15. [판매원 데이터베이스] 다음 릴레이션을 보고 질문에 답하시오. Salesperson은 판매원, Order는 주문, Customer는 고객을 나타낸다. 밑줄 친 속성은 기본키고 custname과 salesperson은 각각 Customer.name과 Salesperson.name을 참조하는 외래키다.

Salesperson(name, age, salary)

Order(number, custname, salesperson, amount)

Customer(name, city, industrytype)

- (1) 모든 판매원(Salesperson)의 이름을 보이시오.
- (2) 고객 '웅길동'의 주문을 수주한 판매원의 이름을 보이시오.
- (3) 주문이 있는 판매원의 이름을 보이시오.
- (4) 주문이 없는 판매원의 이름을 보이시오.
- (5) 고객 '웅길동'의 주문을 수주한 판매원의 나이를 보이시오.
- (6) 나이가 25살인 판매원에게 주문한 고객의 city 값을 보이시오.
- (7) 판매원의 이름과 그 판매원에게 주문을 한 고객의 이름을 보이시오. 단 주문이 없는 판매원도 포함되어 구한다.

2023-04-24

컴퓨터공학과

140

요약

- 릴레이션
- 릴레이션 스키마
- 릴레이션 인스턴스
- 관계 데이터베이스 시스템
- 키
- 무결성 제약조건
- 참조 무결성 제약조건의 옵션
- 관계대수
- 선택연산
- 프로젝션
- 집합연산
- 조인
- 디비전

Q & A

