



데이터베이스

관계 대수





학습목표

- ➔ 관계 대수와 관계 해석의 차이를 설명할 수 있다.
- ➔ 관계 연산자에 대한 분류를 할 수 있다.



학습내용

- ➔ 관계 대수와 관계 해석
- ➔ 관계 연산자의 분류



릴레이션을 조작하기 위한 연산

관계 해석

(Relational Calculus)

질의에 대한
검색 조건을 기술

무엇(What)에 대한
선언적 명시

관계 대수

(Relational Algebra)

질의 수행에 대하여
어떻게 하는지(수행 절차)를 기술

무엇(What)을
어떻게(How)할지를 명시



관계 해석

01 관계 해석(Relational Calculus)

- 관계 대수로 명시할 수 있는 모든 검색 요구는 관계 해석으로도 명시할 수 있으며, 그 역도 성립됨이 증명됨
 - 두 언어의 표현력(Expression Power)은 동등
- 관계형 데이터베이스의 표준 질의 언어: SQL
 - 관계 해석(튜플 관계 해석)을 기반으로 만들어짐
 - ▶ 무엇(What)만을 기술
- 관계형 데이터베이스 내에서 SQL을 처리하는 방법
 - 관계 대수를 기반으로 함
 - ▶ 어떤 순서로 처리하는지가 중요해 짐 → 질의 최적화
- 관계 해석의 기반: 수학의 술어해석(Predicate Calculus)
- 관계 해석의 제안: 데이터베이스 언어의 기초로 사용되기 위하여
코드(E.F. Codd)박사에 의하여 제안

튜플 관계 해석

(Tuple Relational Calculus)

튜플 기반의 해석 → 행 기반

VS

도메인 관계 해석

(Domain Relational Calculus)

도메인 기반의 해석 → 열 기반



관계 해석

02 튜플 관계 해석

- 튜플 변수(Tuple Variable)를 명시
- 튜플 변수의 범위: 릴레이션
 - 튜플 변수는 주어진 릴레이션의 **어떤 튜플도 값으로 가질 수 있음**
- 형태

$\{t \mid \text{COND}(t)\}$



- t : 튜플 변수
- $\text{COND}(t)$: t 를 포함하는 조건식

예 | 성적이 3.5 이상인 학생을 구하는 튜플 해석식

$\{t \mid \text{학생}(t) \text{ AND } t.\text{성적} > 3.5\}$

- 튜플 관계 해석식에서 명시해야 하는 것
 - 각 튜플 **변수 t 의 범위** 릴레이션
 - 튜플들을 특정 조합들로 **선택**하기 위한 조건
 - 검색 속성들의 집합

일반적인 튜플 관계 해석식

$\{t_1.a_i, t_2.a_k, \dots, t_n.a_m \mid \text{COND}(t_1, t_2, \dots, t_{n+m})\}$

- $t_i.a_j$: t_i 의 범위 릴레이션의 속성
- t_1, t_2, \dots, t_{n+m} : 튜플 변수
- $\text{COND}()$: **WFF(Well Formed Formula)**



관계 해석



02 튜플 관계 해석

| WFF

- 원자(Atom)들로 구성된 식

| 원자(Atom)

- $R(t)$ 형태: 튜플 변수와 대응되는 범위 릴레이션

$t_1.A \text{ op } t_2.B$ \rightarrow op는 비교 연산자(즉, $=, >, <, \leq, \geq, \neq$)

$t.A \text{ op } c$ \rightarrow c는 상수, op는 비교 연산자

| WFF

- 원자는 WFF
- F가 WFF이면 (F) 와 $\neg F$ 도 WFF
- F와 G가 WFF이면 $F \text{ and } G$ 와 $F \text{ or } G$ 도 WFF
- $F(t)$ 가 WFF이면 $\forall t(F(t))$ 와 $\exists t(F(t))$ 도 WFF



관계 해석

02 튜플 관계 해석

자유 변수(Free Variable)와 한정 변수(Bounded Variable)



- **속박 변수**: 정량자로 한정된 변수
- **자유 변수**: 정량자로 한정되지 않은 변수

자유 변수(Free Variable)와 한정 변수(Bounded Variable)

- 자유 변수

{t | 학생(t) AND t.성적 > 3.5}

- ▶ 조건을 만족하는 각 튜플이 **t**에 연동됨
- ▶ 학생 릴레이션에 속한 튜플들 중 **조건(t.성적 > 3.5)**에 만족하는 튜플들이 **연속적으로 연동됨**



관계 해석

02

튜플 관계 해석

자유 변수(Free Variable)와 한정 변수(Bounded Variable)

한정 변수

$$(\exists d)(d.snum = 50)$$

- ▶ 튜플 변수 d 는 $(d.snum = 50)$ 라는 식에 속박됨
- ▶ $d.snum = 50$ 을 만족하는 튜플이 존재하는가 존재하지 않는가 만을 평가함
 - ➡ 튜플 변수 d 는 위의 식 밖에서는 의미가 없음
- 예 | 모든 학생이 성적이 1.0이상이면 참이 되는가 아닌가 만을 평가

$$(\forall s)(\text{학생}(s) \text{ and } s.\text{성적} > 1.0)$$

$$\{t \mid \text{COND}(t)\}$$

- ▶ 위의 식의 바($()$)를 기준으로 왼편에는 자유변수만 올 수 있음



관계 해석

02

튜플 관계 해석

예시 1: 학번이 100번인 학생의 이름, 주소를 검색하시오

$$\{t.이름, t.주소 \mid \text{학생}(t) \text{ and } t.학번 = 100\}$$

- 해당 질의에 대한 튜플 해석식이 오직 하나만 있는 것은 아님
- 같은 결과를 얻을 수 있는 많은 식이 있음

예 | $1+1 = 1-1+2 = 1/1+1 = \dots$

$$\{t.이름, t.주소 \mid \text{학생}(t) \text{ and } (\exists s)(\text{학생}(s) \text{ and } s.학번 = 100 \text{ and } t.학번 = s.학번)\}$$

예시 2: 컴퓨터공학과에 속한 모든 학생의 이름과 학번을 구하시오

주1) 해당 질의에 학생 릴레이션은 학과번호를 속성으로 가짐

주2) 학과 릴레이션에 학과명과 학과번호가 있음

$$\{s.이름, s.학번 \mid \text{학생}(s) \text{ and } (\exists d)(\text{학과}(d) \text{ and } d.학과명 = \text{'컴퓨터공학과'} \text{ and } d.학과번호 = s.학과번호)\}$$



관계 해석

03 도메인 관계 해석

- 튜플 관계 해석을 기반으로 한 언어인 SQL은 IBM 산호세연구소에서 개발
- 도메인 관계 해석을 기반으로 한 QBE(Query By Example)가 IBM 왓슨연구소에서 개발
- 도메인 해석과 튜플 해석은 사용되는 변수의 유형만 다를 뿐 유사함
- 도메인 해석에서 변수의 범위는 튜플이 아니라 속성의 도메인에 속한 값

➡ 질의 결과로 차수가 n 인 릴레이션을 생성하기 위해서는
 각 속성 마다 하나씩 총 n 개의 도메인 변수(Domain Variable)이 필요

SQL

(Structured Query Language)

- 해석 기반: 튜플 관계 해석
- 개발: IBM 산호세연구소

VS

QBE

(Query By Example)

- 해석 기반: 도메인 관계 해석
- 개발: IBM 왓슨연구소

- 도메인 해석과 튜플 해석: 사용되는 변수의 유형만 다를 뿐 유사함
- 도메인 해석에서 변수의 범위: 속성의 도메인에 속한 값

➡ 질의 결과로 차수가 n 인 릴레이션을 생성하기 위해서는
 각 속성 마다 하나씩 총 n 개의 도메인 변수(Domain Variable)가 필요함



관계 해석

03

도메인 관계 해석

도메인 관계 해석식

$$\{x_1, x_2, \dots, x_n \mid \text{COND}(x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_{n+m})\}$$

- $x_1, x_2, \dots, x_n, x_{n+1}, \dots, x_{n+m}$: 속성의 도메인을 범위로 가지는 도메인 변수
- $\text{COND}()$: 도메인 관계 해석의 조건

예 |

{학번 | (\exists 학번)(학생(학번, 이름, 주소) and 학번 = 100)}

원자

$R(x_1, x_2, \dots, x_j) \rightarrow$

R은 차수가 j인 릴레이션, x_i 는 도메인 변수

- 이 원자식은 x_i 가 튜플의 i번째 속성 값일 때 $\langle x_1, x_2, \dots, x_j \rangle$ 가 릴레이션 R에 속함을 의미

$x_i \text{ op } x_j$

\rightarrow

op는 비교 연산자(=, >, <, \geq , \leq , \neq)

$x_i \text{ op } c$

\rightarrow

c는 상수, op는 비교 연산자



관계 해석

03

도메인 관계 해석

WFF(Well Formed Formula)

- 원자는 WFF
- F 가 WFF이면 (F) 와 $\neg F$ 도 WFF
- F 와 G 가 WFF이면 $F \text{ and } G$ 와 $F \text{ or } G$ 도 WFF
- $F(x)$ 가 WFF이면 $\forall x(F(x))$ 와 $\exists x(F(x))$ 도 WFF

예시: 컴퓨터공학과에 속한 모든 학생의 이름과 학번을 구하시오

주1) 학생 릴레이션은 학과번호를 속성으로 가짐

주2) 학과 릴레이션에 학과명과 학과번호가 있음

{이름, 학번 | 학생(이름, 학번, 학과번호) and
 (\exists 학과번호1)(\exists 학과명)(학과(학과명, 학과번호1)
 and 학과명 = '컴퓨터공학과'
 and 학과번호1 = 학과번호)}



QBE

01 QBE(Query By Example)

2차원 그래픽 터미널 이용

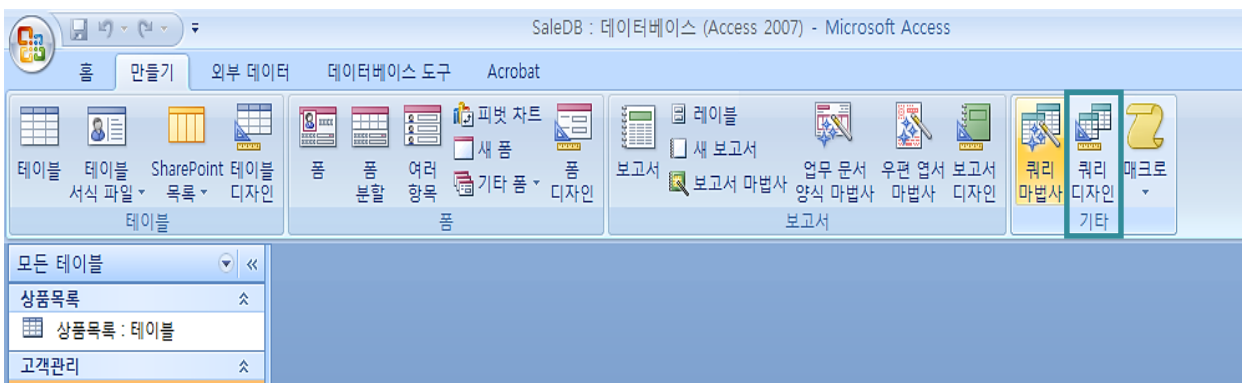
학생	학번	이름	학년	학과
	P._STX		2	

MS-ACCESS에서 QBE를 지원

Microsoft사에서 제공되는 데이터베이스 관리용 애플리케이션

- 간단하고 편리한 기능들을 제공
- **초보 사용자**가 데이터베이스를 구축, 관리하기 용이함

02 MS-ACCESS의 쿼리 디자인 기능





QBE

02 MS-ACCESS의 쿼리 디자인 기능

고객관리 테이블에서 성별이 "남"인 고객의 정보를 검색

드래그앤드랍

조건 입력

성별
고객관리테이블
= '남'



도구 모음의 '보기' 단추나 실행 단추 클릭

질의 결과

고객번호	고객명	성별	우편 번호	주소	전화번호	전자 메일 주소
1	이순제	남	123-345	대전 서구	(123)4354-324	abc@kut.ac.kr
2	정준하	남	123-345	대전 서구	(123)4354-324	bdc@kut.ac.kr
4	정문술	남	432-433	서울 성북구	(02)4342-213	ccc@fred.net
8	김민수	남	343-841	충남 천안	(041)1234-5322	kimms@kut.ac.kr
* (새 항목)						



관계 대수와 관계 해석



QBE

02 MS-ACCESS의 쿼리 디자인 기능

고객관리 테이블에서 성별이 '여'이고 고객번호가 3보다 큰 고객 정보를 검색

필드:	고객번호	고객명	성별	우편 번호	주소
이름:	고객관리	고객관리	고객관리	고객관리	고객관리
정렬:					
표시:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
조건:	>3		= '여'		
또는:					



QBE

03

다중 테이블 검색

■ 상품목록 테이블과 판매관리 테이블을 이용하여 각 판매정보에 상품명을 표시

The screenshot shows a database query editor with two tables: '판매관리' (Sales Management) and '상품목록' (Product List). The '판매관리' table has fields: 판매번호, 판매일자, 고객번호, 상품번호, and 겹수. The '상품목록' table has fields: 상품번호, 상품명, and 가격. A join is established between the two tables. A dialog box titled '테이블 표시' (Table Display) is open, showing a list of tables: 가입동기, 고객관리, 당첨번호, 당첨번호연결시험, 부서정보, 상품목록, 직원관리, 직원정보, 판매관리, and 회원. The '상품목록' and '판매관리' tables are highlighted.

The screenshot shows the same database query editor with the '테이블 표시' dialog box closed. The query results are displayed in a table. The table has three columns: '판매관리.*', '상품명', and '상품번호'. The '판매관리.*' column contains the value '판매관리.*'. The '상품명' column contains the value '상품목록'. The '상품번호' column contains the value '상품번호'. The table also has a '조건' (Condition) column with a checkmark in the first row and a condition '[판매관리].[상품번호]' in the second row.

필드:	판매관리.*	상품명	상품번호
테이블:	판매관리	상품목록	상품목록
정렬:			
표시:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
조건:			[판매관리].[상품번호]
또는:			



관계 대수

01 관계 대수(Relational Algebra)

- 관계 해석은 **질의**를 나타내기 위한 **선언적 표기법**
- 관계형 모델에서의 **기본적인 연산들의 집합**
- 관계 대수의 **중요성**

01 관계형 모델의 연산을 위한 공식적인 토대를 제공

02 DBMS에서 질의를 구현하고 최적화 하기 위한 기반

03 관계 대수의 몇 가지 개념은 SQL 표준 질의에 반영됨



관계형 모델에 기반한 분류

수학적 집합 연산

수학적 집합 이론에 근거

합집합, 교집합, 차집합,
카티션 곱

순수 관계 연산

2차원 테이블 형태의 구조 (형식적
모형)

선택, 프로젝트, 조인 등

피연산자의 수에 따른 분류

단항 연산자

연산 대상이 되는 릴레이션을
하나만 가지는 연산자

선택, 프로젝트,
개명연산 등

이항 연산자

연산 대상이 되는 릴레이션을
두 개 가지는 연산자

카티션 프로젝트, 조인,
합집합, 교집합, 차집합 등



관계 연산자의 분류



단항 연산자

01

선택 연산

- 릴레이션에 선택 조건을 기술하여 **조건을 만족하는 튜플들을 선택**하는데 사용하는 연산

02

프로젝트 연산

- 릴레이션에서 **특정 속성들만을 선택**하는 연산



이항 연산자

01

합집합, 교집합, 차집합, 카티션 곱

- 릴레이션은 튜플들의 집합
- 따라서 **집합 연산자를 사용 가능**

02

조인

- 두 릴레이션으로부터 **관련된 튜플들을 결합**하여 하나의 튜플로 만듦



관계 연산자의 분류



폐쇄 성질

01 폐쇄 성질(Closure Property)

피연산자와 연산자의 결과가 같은 자료형

피연산자(Operand)

연산자(Operator)

〔 연산에 참여하는 자료 〕

VS

〔 자료를 처리하는 방법 〕

모든 연산자가 폐쇄 성질을 가지는 것은 아님

예1 | 정수 + 정수 = 정수

▶ 더하기 연산자는 정수형에 폐쇄 성질이 있음

예2 | 정수/정수 = 정수 또는 실수

▶ 나누기 연산자는 정수형에 대하여 폐쇄 성질이 없음

02 특징

관계 대수의 모든 연산은 릴레이션에 대하여 폐쇄 성질을 지님

- 어떤 릴레이션에 관계 대수 연산을 수행하면 그 결과도 릴레이션(튜플의 집합) 임

관계 대수는 **질의 처리의 대상(What)**과 **절차(How)**를 나타냄

- 폐쇄 성질이 없으면 절차를 자유롭게 표현하기 어려움

폐쇄 성질에 따라서 **연산자의 중첩(Nesting) 순서**를 표현

- 연산자 3(연산자2(연산자1(릴레이션))) → 릴레이션



1 관계 대수와 관계 해석

- ✓ 관계 해석: 질의에 대한 검색 조건 기술, 무엇에 대한 선언적 명시
- ✓ 관계 대수: 질의 수행에 대하여 어떻게 하는지를 기술, 즉 무엇을 어떻게 할지를 명시

2 관계 연산자의 분류

- ✓ 수학적 집합 연산: 관계형 모델이 집합 이론에 근거
 - 합집합, 교집합, 차집합, 카티션 곱
- ✓ 순수 관계 연산자: 릴레이션이 2차원 테이블 구조
 - 선택(σ), 프로젝트(π), 조인(\bowtie) 등
- ✓ 폐쇄 성질
 - 관계 대수의 모든 연산은 릴레이션에 대하여 폐쇄 성질을 지님
 - 어떤 릴레이션에 관계 대수 연산을 수행하면 그 결과 또한 릴레이션임