

5. 관계 대수와 관계 해석

❖ 관계 데이터 연산

- ◆ 연산과 데이터 언어
 - 연산 : 시스템 입장
 - 데이터 언어 : 사용자 입장
- ◆ 관계 데이터 언어
 - i . 관계 대수(relational algebra)
 - ◆ 절차 언어 : how, what
 - ii . 관계해석(relational calculus)
 - ◆ 비 절차 언어 : what
 - ◆ 튜플 관계해석
 - ◆ 도메인 관계해석
- ◆ 관계 해석과 관계 대수는 표현이나 기능 면에서 동등

❖ 관계대수(Relational Algebra)

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

▶ 일반 집합 연산자(1)

i . 합집합 (union, \cup)

$$R \cup S = \{ t \mid t \in R \vee t \in S \}$$

$$|R \cup S| \leq |R| + |S|$$

ii . 교집합 (intersect, \cap)

$$R \cap S = \{ t \mid t \in R \wedge t \in S \}$$

$$|R \cap S| \leq \min\{ |R|, |S| \}$$

iii . 차집합 (difference, $-$)

$$R - S = \{ t \mid t \in R \wedge t \notin S \}$$

$$|R - S| \leq |R|$$

iv . 카티션 프로덕트 (cartesian product, \times)

$$R \times S = \{ r \cdot s \mid r \in R \wedge s \in S \}$$

\cdot : 접속(concatenation)

$$|R \times S| = |R| \times |S|$$

차수(degree) = R의 차수 + S의 차수

▶ 일반 집합 연산자(2)

◆ Note

- 합병가능(union-compatible)한 릴레이션
 - ◆ $\cup, \cap, -$ 연산의 피연산자들은
 - i. 차수가 같아야 함
 - ii. 대응 애트리뷰트 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
$$R \cup S = S \cup R$$
$$R \cap S = S \cap R$$
$$R \times S = S \times R$$
- 결합적, 교환적 성질은 관계 데이터 처리시 최적화에 이용됨

▶ 순수 관계 연산자

- ◆ 릴레이션 R 의 애트리뷰트 집합 : $X = \{A_1, \dots, A_n\}$
- ◆ 릴레이션 : $R(X) = R(A_1, \dots, A_n)$
- ◆ R 의 튜플 : $r = \langle a_1, \dots, a_n \rangle$
- ◆ 튜플 r 에 대한 애트리뷰트 A_i 의 값
 - $r.A_i$ 또는 a_i
 - $r.A_i = r[A_i] = a_i$
- ◆ 일반적으로
 - $\langle r.A_1, r.A_2, \dots, r.A_n \rangle = \langle r[A_1], r[A_2], \dots, r[A_n] \rangle = r[A_1, A_2, \dots, A_n] = r[X]$

실렉트 (SELECT, σ)(1)

- ◆ A, B가 릴레이션 R의 애트리뷰트일 때,

$$\sigma_{A\theta v}(R) = \{ r \mid r \in R \wedge r.A\theta v \}$$

$$\sigma_{A\theta B}(R) = \{ r \mid r \in R \wedge \underbrace{r.A\theta r.B}_{\text{조건식(predicate)}} \}$$

조건식(predicate)

단, $\theta(\text{theta}) = \{ <, >, \leq, \geq, =, \neq \}$

v : 상수

- ◆ 조건식을 참으로 만드는 튜플 선택하는 것 (일원 연산자)
- ◆ 릴레이션의 수평적 부분집합 (horizontal subset)

실렉트 (2)

◆ example

$\sigma_{\text{학과} = \text{'컴퓨터'}} (\text{학생})$

$\sigma_{\text{학번} = 300 \wedge \text{과목번호} = \text{'C312'}} (\text{등록})$

$\sigma_{\text{중간성적} < \text{기말성적}} (\text{등록})$

$$\begin{aligned} \sigma_{\text{조건2}}(\sigma_{\text{조건1}}(R)) &= \sigma_{\text{조건1}}(\sigma_{\text{조건2}}(R)) \\ &= \sigma_{\text{조건1} \wedge \text{조건2}} (R) \end{aligned}$$

학생, 등록 릴레이션에 대한 실렉트 연산 예

$\sigma_{\text{학과}='컴퓨터'}(\text{학생})$

학번	이름	학년	학과
100	나수영	4	컴퓨터
300	정기태	1	전기
400	송병길	4	컴퓨터

$\sigma_{\text{학번}=300 \wedge \text{과목번호}='C312'}(\text{등록})$

학번	과목번호	성적	중간성적	기말성적
300	C312	A	90	95

$\sigma_{\text{중간성적} < \text{기말성적}}(\text{등록})$

학번	과목번호	성적	중간성적	기말성적
100	C413	A	90	95
300	C312	A	90	95
400	C312	A	90	95
400	C413	B	80	85
400	E412	C	65	75

프로젝트 (PROJECT, Π)

- ◆ 릴레이션 $R(X)$ 에서
 $Y \subseteq X$ 이고 $Y = \{B_1, B_2, \dots, B_m\}$ 이면,
 $\Pi_Y(R) = \{ \langle r.B_1, \dots, r.B_m \rangle \mid r \in R \}$
- ◆ example
 - 학생(학번, 이름, 학년)에서 $\Pi_{\text{이름}}(\text{학생})$
- ◆ 릴레이션의 수직적 부분집합(vertical subset)
- ◆ 생성된 중복 튜플은 제거

학생, 과목 릴레이션에 대한 프로젝트 연산 예

$\Pi_{\text{이름, 학과}}(\text{학생})$

이름	학과
나수영	컴퓨터
이찬수	전기
정기태	컴퓨터
송병길	컴퓨터
박종화	컴퓨터

$\Pi_{\text{과목이름, 담당교수}}(\text{과목})$

과목이름	담당교수
프로그래밍	김성국
자료구조	황수관
파일구조	이규찬
데이터베이스	이일로
반도체	홍봉진

조인 (JOIN, ⋈)(1)

◆ 세타 조인 (theta join)

$R(X), S(Y), A \in X, B \in Y$ 에 대하여

$$R \bowtie_{A \theta B} S = \{ r \cdot s \mid r \in R \wedge s \in S \wedge (r.A \theta s.B) \}$$

- A, B : 조인 애트리뷰트(joining attribute)
- 결과 차수 = R 의 차수 + S 의 차수
- 원 소속 릴레이션의 이름을 애트리뷰트 앞에 한정어로 붙여 일관성 유지
 - ◆ 예) 학생 $\bowtie_{\text{학번}=\text{학번}}$ 등록의 결과 릴레이션 에서
 - 학생.학번, 학생.이름, 학생.학년, 학생.학과, 등록.학번, 등록.과목번호, 등록.성적, 등록.중간성적, 등록.기말성적

동일 조인 (equi join)

- ◆ 세타 조인에서 θ 가 " $=$ "인 경우

$$R \bowtie_{A=B} S = \{ r \cdot s \mid r \in R \wedge s \in S \wedge (r.A = s.B) \}$$

- ◆ 릴레이션 학생과 등록의 동일 조인 예

학생 $\bowtie_{\text{학번}=\text{학번}}$ 등록

학생.학번	이름	학년	학과	등록.학번	과목번호	성적	중간성적	기말성적
100	나수영	4	컴퓨터	100	C413	A	90	95
100	나수영	4	컴퓨터	100	E412	A	95	95
200	이찬수	3	전기	200	C123	B	85	80
300	정기태	1	컴퓨터	300	C312	A	90	95
300	정기태	1	컴퓨터	300	C324	C	75	75
300	정기태	1	컴퓨터	300	C413	A	95	95
400	송병길	4	컴퓨터	400	C312	A	90	95
400	송병길	4	컴퓨터	400	C324	A	95	90
400	송병길	4	컴퓨터	400	C413	B	80	85
400	송병길	4	컴퓨터	400	E412	C	65	75
500	박종화	2	산공	500	C312	B	85	80

조인 (2)

- ◆ 자연조인 (natural join, \bowtie_N)

$R(X)$, $S(Y)$ 의 조인 애트리뷰트를 $Z(=X \cap Y)$ 라 하면

$$R \bowtie_N S$$

$$= \{ \langle r \cdot s \rangle [X \cup Y] \mid r \in R \wedge s \in S \wedge r[Z] = s[Z] \}$$

$$= \Pi_{X \cup Y}(\sigma_{Z=Z}(R \times S))$$

$$= \Pi_{X \cup Y}(R \bowtie_{Z=Z} S)$$

즉 동일조인의 결과 릴레이션에서 애트리뷰트의 중복을 제거함

릴레이션 학생과 등록의 자연 조인 예

학생 ⋈ N등록

학번	이름	학년	학과	과목번호	성적	중간성적	기말성적
100	나수영	4	컴퓨터	C413	A	90	95
100	나수영	4	컴퓨터	E412	A	95	95
200	이찬수	3	전기	C123	B	85	80
300	정기태	1	컴퓨터	C312	A	90	95
300	정기태	1	컴퓨터	C324	C	75	75
300	정기태	1	컴퓨터	C413	A	95	95
400	송병길	4	컴퓨터	C312	A	90	95
400	송병길	4	컴퓨터	C324	A	95	90
400	송병길	4	컴퓨터	C413	B	80	85
400	송병길	4	컴퓨터	E412	C	65	75
500	박종화	2	산공	C312	B	85	80

디비전 (DIVISION, \div)(1)

◆ 릴레이션 $R(X), S(Y)$ 에 대하여

$Y \subseteq X$ 이고 $Z = X - Y$ 이면 $R(X) = R(Z, Y)$

$R \div S$

$= \{ t \mid t \in \Pi_Z(R) \wedge t \cdot s \in R \text{ for all } s \in S \}$

디비전(2)

◆ example

학과목(SC)

학번 (Sno)	과목번호 (Cno)
100	C413
100	E412
200	C123
300	C312
300	C324
300	C413
400	C312
400	C324
400	C413
400	E412
500	C312

과목1(C1)

과목번호 (Cno)
C413

SC ÷ C1

학번 (Sno)
100
300
400

과목2(C2)

과목번호 (Cno)
C312
C413

SC ÷ C2

학번 (Sno)
300
400

과목3(C3)

과목번호 (Cno)
C312
C413
E412

SC ÷ C3

학번 (Sno)
400

개명 연산 (RENAME, ρ)

- ◆ 중간 결과 릴레이션에 이름을 지정하거나 애트리뷰트 이름을 변경할 때 사용

① $\rho_S(E)$

관계 대수식 E 의 결과 릴레이션의 이름을 S 로 지정

② $\rho_{S(B_1, B_2, \dots, B_m)}(E)$

관계 대수식 E 의 결과 릴레이션의 이름을 S 로 지정하면서
애트리뷰트 이름을 각각 B_1, B_2, \dots, B_m 으로 변경

③ $\rho_{(B_1, B_2, \dots, B_m)}(R)$

릴레이션 R 의 애트리뷰트 이름을 각각 B_1, B_2, \dots, B_m 으로 변경

▶ 근원 연산과 복합 연산

◆ 근원 연산 (primitive operations)

- 하나의 논리적 기능을 수행, 다른 연산을 이용하여 표현할 수 없음
 - ◆ 합집합, 차집합, 카티션 프로덕트, 프로젝트, 실렉트

◆ 복합 연산 (composite operations)

- 근원 연산을 이용하여 표현할 수 있음
 - ◆ 교집합, 조인, 디비전

$$\begin{aligned} R \cap S &= R - (R - S) = S - (S - R) \\ &= (R \cup S) - ((R - S) \cup (S - R)) \end{aligned}$$

$$R \bowtie_{A \theta B} S = \sigma_{A \theta B} (R \times S)$$

- 연산력 보다는 표현력 증대

▶ 관계 대수의 확장(1)

i . 세미 조인 (Semijoin, \bowtie)

- ◆ $R(X), S(Y)$ 의 조인 애트리뷰트를 $Z=X \cap Y$ 라 하면
 $R \bowtie S = R \bowtie_N (\Pi_Z(S)) = \Pi_X(R \bowtie_N S)$

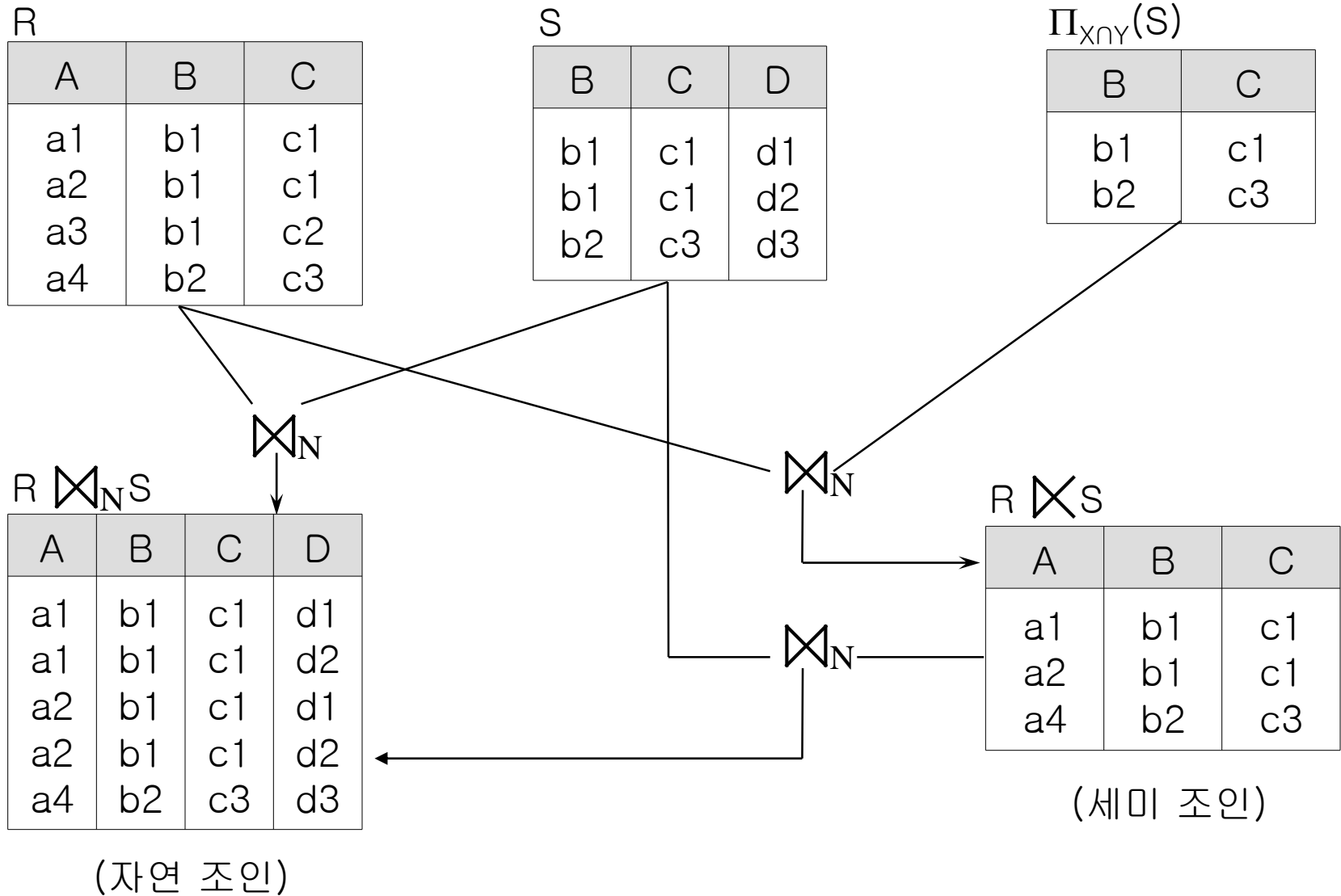
- S 와 자연조인을 할 수 있는 R 의 튜플

- ◆ 특징

- $R \bowtie S \neq S \bowtie R$
- $R \bowtie_N S = (R \bowtie S) \bowtie_N S = (S \bowtie R) \bowtie_N R$

– 처리해야 될 데이터의 양이 다름

※ 자연조인과 세미조인



▶ 관계대수의 확장(2)

ii. 외부 조인 (Outerjoin, \bowtie^+)

- ◆ 한 릴레이션에 있는 튜플이 조인할 상대 릴레이션에 대응되는 튜플이 없을 경우, 상대를 널(null) 튜플로 만들어 결과 릴레이션에 포함
- ◆ 두 조인 릴레이션의 모든 튜플들이 결과 릴레이션에 포함됨

※ 자연조인과 외부조인

R

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

S

B	C	D
b1	c1	d1
b1	c1	d2
b2	c3	d3
b3	c3	d3

\bowtie^+

$R \bowtie^+ S$

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a2	b1	c1	d1
a2	b1	c1	d2
a3	b1	c2	
a4	b2	c3	d3
	b3	c3	d3

(외부조인)

\bowtie_N

$R \bowtie_N S$

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a2	b1	c1	d1
a2	b1	c1	d2
a4	b2	c3	d3

(자연조인)

▶ 관계대수의 확장(3)

iii. 외부 합집합 (Outer-union, \cup^+)

- ◆ 합병 가능하지 않은(부분적으로 합병 가능한) 두 릴레이션을 차수를 확장시켜 합집합으로 만듦

※ 외부 합집합

R

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

S

B	C	D
b1	c1	d1
b1	c1	d2
b2	c2	d3

U⁺



A	B	C	D
a1	b1	c1	
a2	b1	c1	
a3	b1	c2	
a4	b2	c3	
	b1	c1	d1
	b1	c1	d2
	b2	c2	d3

▶ 관계대수의 확장(4)

iv. 집단 연산

- $AVG_{성적}(등록)$
 - ◆ 등록 릴레이션에 있는 성적 애트리뷰트 값들에 대해 평균값 계산
- $GROUP_{학년}(학생)$
 - ◆ 학생 릴레이션의 튜플들을 학년 값에 따라 그룹 짓게 함
- $GROUP_{과목번호}AVG_{성적}(등록)$
 - ◆ 과목별 그룹에 대한 평균성적
- 일반 형식 : $G_A F_B(E)$
 - ◆ E : 관계 대수식
 - ◆ F : 집단 함수 (SUM, AVG, MAX, MIN, COUNT)
 - ◆ B : 집단 함수의 적용 대상 애트리뷰트
 - ◆ G : 그룹 함수 GROUP
 - ◆ A : 그룹 함수가 적용할 애트리뷰트

▶ 관계대수의 질의문 표현(1)

- ◆ 모든 학생의 이름과 학과를 검색하라.

$\Pi_{\text{이름, 학과}}(\text{학생})$

- ◆ 과목번호가 C413인 과목에 등록한 학생의 이름과 성적은 무엇인가?

$\Pi_{\text{이름, 성적}}(\sigma_{\text{과목번호}='C413'}(\text{학생} \bowtie_N \text{등록}))$

- ◆ '화일구조' 과목을 가르치는 교수의 이름을 검색하라.

$\Pi_{\text{담당교수}}(\sigma_{\text{과목이름}='화일구조'}(\text{과목}))$

▶ 관계대수의 질의문 표현(2)

- ◆ 모든 과목에 수강하고 있는 학생의 학번과 이름을 검색하라

$$\Pi_{\text{학번,이름}}((\Pi_{\text{학번,과목번호}}(\text{등록}) \div \Pi_{\text{과목번호}}(\text{과목})) \bowtie_N \text{학생})$$

- ◆ 학번이 600, 이름이 '김영호', 학년이 4, 학과가 컴퓨터인 학생을 삽입하라.

$$\text{학생} \cup \{ \langle 600, \text{'김영호'}, 4, \text{'컴퓨터'} \rangle \}$$

- ◆ 과목 '데이터베이스'를 삭제하라.

$$\text{과목} - (\sigma_{\text{과목이름}=\text{'데이터베이스'}}(\text{과목}))$$

❖ 관계 해석 (Relational Calculus)

- ◆ predicate calculus에 기반
 - predicate : a function whose value is true or false
- ◆ 관계 데이터 모델의 연산 표현 방법
- ◆ 비절차적(non-procedural)
 - 원하는 정보가 무엇이라는 것만 선언
 - 튜플 관계 해석(tuple relational calculus)
 - 도메인 관계 해석(domain relational calculus)

▶ 튜플 관계해석(1)

- ◆ 원하는 릴레이션을 튜플해석식(tuple calculus expression)으로 정의할 수 있는 표기법
- ◆ 튜플 해석식의 구성 요소
 - i . 튜플 변수(tuple variable) 또는 범위변수(range variable): t
 - ◆ $R(t)$: t 의 범위식(range formula)
 - ◆ R : t 의 범위 릴레이션(range relation)
 - ii. 한정 애트리뷰트(qualified attribute) : $t.A$ 또는 $t[A]$
 - ◆ 튜플 변수 t 가 나타내는 튜플의 어떤 애트리뷰트 A 의 값
 - Student(s)
 $s.Sno$

▶ 튜플 관계해석(2)

◆ 튜플 해석식의 구성 요소(con't)

iii. 원자(atom)

① $R(t)$

t : 튜플 변수

R : t 의 범위 릴레이션

② $t.A \theta u.B$

t, u : 튜플 변수

A, B : t 와 u 에 대한 한정 애트리뷰트

θ : 비교 연산자($=, \neq, <, \leq, >, \geq$)

③ $t.A \theta c$

A : 튜플 변수 t 에 대한 한정 애트리뷰트

c : 상수

- 원자의 실행 결과는 반드시 참(True) 또는 거짓(False)

▶ 튜플 관계해석(3)

◆ 튜플 해석식의 구성 요소(con't)

iv. 정형식(WFF, Well-formed formula)

- 원자, 불리언 연산자(\wedge, \vee, \neg), 정량자 (\forall, \exists)가 다음 규칙에 따라 결합된 식
 - ① 모든 원자는 WFF
 - ② F 가 WFF이면, (F) 와 $\neg F$ 도 WFF
 - ③ F 와 G 가 WFF이면, $F \wedge G$ 와 $F \vee G$ 도 WFF
 - ④ 튜플 변수 t 가 자유변수로 사용된 $F(t)$ 가 WFF이면, $\forall t(F(t))$ 와 $\exists t(F(t))$ 도 WFF
 - ⑤ 위의 규칙만을 반복 적용해서 만들어진 식은 WFF
- 정형식의 예
 - $s.Sno = 100$
 - $c.Cno \neq e.Cno$
 - $s.Sno = e.Sno \wedge e.Cno \neq c.Cno$
 - $(\exists e)(e.Sno = s.Sno \wedge e.Cno = 'C413')$

Note

- ◆ 자유변수(free variable)
 - 정량자로 한정되지 않는 튜플 변수
 - \forall : 전칭 정량자(Universal quantifier) – “for all”
 - \exists : 존재 정량자(Existential quantifier) – “there exists”
- ◆ 속박변수(bound variable)
 - 정량자로 한정된 튜플 변수
 - ◆ $\forall t$
 - ◆ $\exists t$

튜플 해석식

◆ 형식

$$\{ t_1.A_1, t_2.A_2, \dots, t_n.A_n \mid F(t_1, \dots, t_n, t_{n+1}, \dots, t_{n+m}) \}$$

- t_i : 튜플 변수
- $F(t_1, \dots, t_n, t_{n+1}, \dots, t_{n+m})$: t_i 가 연관된 정형식으로 조건을 명세
- 막대 (|) 왼편에 나온 한정 애트리뷰트들은 목표리스트로서 막대 (|) 오른편에 명세된 조건을 만족하는 결과로 추출 됨

◆ example

- $\{ s.Sname \mid STUDENT(s) \}$
- $\{ s.Sname \mid STUDENT(s) \wedge s.Dept = \text{'컴퓨터'} \}$
- $\{ s.Sname, s.Dept \mid STUDENT(s) \wedge (\exists e)(ENROL(e) \wedge s.Sno = e.Sno \wedge e.Grade = \text{'A'}) \}$

▶ 튜플 해석식의 질의문 표현(1)

- ◆ 과목 C413에서 성적이 A인 학생의 학번을 모두 검색하라

$$\{ e.Sno \mid ENROL(e) \wedge e.Cno='C413' \wedge e.Grade='A' \}$$

- ◆ 과목 C413을 등록한 학생의 이름과 학과를 모두 검색하라

$$\{ s.Sname, s.Dept \mid STUDENT(s) \wedge \exists e(ENROL(e) \wedge s.Sno=e.Sno \wedge e.Cno='C413') \}$$

▶ 튜플 해석식의 질의문 표현(2)

- ◆ 모든 과목에 등록한 학생의 이름을 전부 검색하라.

$$\{ s.Sname \mid STUDENT(s) \wedge (\forall c)(\exists e)(COURSE(c) \wedge ENROL(e) \wedge e.Sno=s.Sno \wedge e.Cno=c.Cno) \}$$

- ◆ 과목 C413에 등록하지 않은 학생의 이름 전부를 검색하라.

$$\{ s.Sname \mid STUDENT(s) \wedge (\neg \exists e)(ENROL(e) \wedge s.Sno=e.Sno \wedge e.Cno='C413') \}$$

▶ 도메인 관계해석 (1)

- ◆ 원하는 릴레이션을 도메인 해석식 (domain calculus expression)으로 표현하는 방법
- ◆ 도메인 해석식의 구성요소
 - i . 도메인 변수(domain variable)
 - ◆ 지정된 애트리뷰트의 도메인의 한 원소만을 값으로 취하는 변수
 - xSno, xSname, ...
 - ◆ 범위식
 - STUDENT(xSno, xSname, xDept, xYear)

▶ 도메인 관계해석 (2)

◆ 도메인 해석식의 구성요소(con't)

ii. 원자(atom)

① $R(x_1, x_2, \dots, x_n)$

x_i : 도메인 변수

R : x_i 의 범위 릴레이션

$\langle x_1, x_2, \dots, x_n \rangle$ 에 해당하는 값의 리스트는 릴레이션 R 의 튜플

② $x \theta y$

x, y : 도메인 변수

θ : 비교 연산자($=, \neq, <, \leq, >, \geq$)

③ $x \theta c$

x : 도메인 변수

θ : 비교 연산자

c : x 가 정의된 도메인 값의 상수

- 원자의 실행 결과는 반드시 참(True) 또는 거짓(False)

▶ 도메인 관계해석 (3)

◆ 도메인 해석식의 구성요소(con't)

iii. 정형식(WFF, Well-formed formula)

- ◆ 원자, 불리언 연산자(\wedge, \vee, \neg), 정량자(\forall, \exists)가 다음 규칙에 따라 결합되어 표현된 식

- ① 모든 원자는 WFF
- ② F 가 WFF이면, (F) 와 $\neg F$ 도 WFF
- ③ F 와 G 가 WFF이면, $F \wedge G$ 와 $F \vee G$ 도 WFF
- ④ 도메인 변수 x 가 자유변수로 사용된 $F(x)$ 가 WFF이면, $(\forall x)(F(x))$ 와 $(\exists x)(F(x))$ 도 WFF
- ⑤ 위의 규칙만을 반복 적용해서 만들어진 식은 WFF

도메인 해석식

◆ 형식

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

◆ x_i : 도메인 변수

◆ $F(x_1, \dots, x_n, x_{n+1}, \dots, x_{n+m})$: x_i 에 대한 정형식

- 막대 (|) 왼편에 나온 도메인 변수들은 목표 리스트로서 막대 (|) 오른편에 명세된 조건을 만족하는 도메인 값으로 만들어지는 튜플

● example

- ① $\{ xSname \mid STUDENT(xSno, xSname, xYear, xDept) \}$
- ② $\{ xSname \mid (\exists xDept)(STUDENT(xSno, xSname, xYear, xDept) \wedge xDept = \text{'컴퓨터'}) \}$
- ③ $\{ xSno, xDept \mid STUDENT(xSno, xSname, xYear, xDept) \wedge (\exists xxSno)(\exists xGrade)(ENROL(xxSno, xCno, xGrade, xMidterm, xFinal) \wedge xSno = xxSno \wedge xGrade = \text{'A'}) \}$

▶ 도메인 해석식의 질의문 표현(1)

- ◆ 컴퓨터학과 3,4 학년의 이름을 검색하라.

$$\{xSname \mid (\exists xYear)(\exists xDept)(STUDENT(xSno, xSname, xYear, xDept) \wedge xYear \geq 3 \wedge xDept='컴퓨터') \}$$

- ◆ 과목 C413에서 성적이 A인 학생의 학번을 모두 검색하라

$$\{xSno \mid (\exists xCno)(\exists xGrade)(ENROL(xSno, xCno, xGrade, xMidterm, xFinal) \wedge xCno='C413' \wedge xGrade='A') \}$$

▶ 도메인 해석식의 질의문 표현(2)

- ◆ 기말 성적이 90점 이상인 학생의 학번과 이름을 검색하라.

$$\{xSno, xSname \mid (STUDENT(xSno, xSname, xYear, xDept) \wedge (\exists xFinal)(\exists xxSno) (ENROL(xxSno, xCno, xGrade, xMidterm, xFinal) \wedge xSno=xxSno \wedge xFinal \geq 90)) \}$$

- ◆ 과목 C324에 등록하지 않은 학생의 이름을 검색하라.

$$\{ xSname \mid (\exists xSno)((STUDENT(xSname, xSno, xYear, xDept) \wedge (\neg \exists xxSno) (\exists xCno) (ENROL(xxSno, xCno, xGrade, xMidterm, xFinal) \wedge xSno=xxSno \wedge xCno='C324')) \}$$

❖ QBE (Query By Example)

- ◆ 도메인 관계 해석 사용
- ◆ 이차원 구문(two-dimensional syntax) 언어
- ◆ 테이블 형태 "Skeleton"
- ◆ 예(example)를 질의문 명세에 사용
 - 예제 원소(example element) : 도메인 변수

▶ 데이터 검색(1)

◆ 단순 조건 검색

- 컴퓨터학과 4학년 학생의 학번과 이름을 검색하라

STUDENT	Sno	Sname	Year	Dept
	P.	P.	4	컴퓨터

- 중복되는 것은 자동적으로 제거됨
- 'ALL'을 삽입하면 중복을 허용

STUDENT	Sno	Sname	Year	Dept
		P.ALL	4	컴퓨터

▶ 데이터 검색(2)

◆ 테이블 전체의 검색

- 학생 테이블의 모든 사항을 검색하라

STUDENT	Sno	Sname	Year	Dept
	P.	P.	P.	P.

- 간단한 방법 : 테이블 이름 밑에 P. 명세

STUDENT	Sno	Sname	Year	Dept
P.				

▶ 데이터 검색(3)

◆ 복수 조건 검색

- 'OR' 조건 : 두 개의 행, 다른 예제 원소

- ◆ 기말성적이 85점 이상이거나 과목번호 'C413'에 등록한 학생의 학번을 검색하라

ENROL	Sno	Cno	Final	Midterm
	P. P.	C413	≥85	

- 'AND' 조건 : 하나의 행, 같은 예제 원소

- ◆ 과목번호가 'C413'이고 기말성적이 85점 이상인 학생의 학번

ENROL	Sno	Cno	Final	Midterm
	P.	C413	≥85	

▶ 데이터 검색(4)

- ◆ 복수 조건 검색(con't)
 - 조건 상자(condition box)의 사용

ENROL	Sno	Cno	Final	Midterm
	P.	_EC	_EF	

CONDITIONS
_EC=C413 AND _EF ≥85

▶ 데이터 검색(5)

◆ 복수 테이블에서 검색

- 기말성적이 85점 이상이거나 과목 'C413'을 등록한 학생의 이름

ENROL	Sno	Cno	Final	Midterm
	_STX _STY	C413	≥85	

STUDENT	Sno	Sname	Year	Dept
	_STX _STY	P. P.		

▶ 데이터의 삽입

◆ 단순 레코드의 삽입

- 학번이 100이고 과목번호가 'C413'인 튜플 삽입

ENROL	Sno	Cno	Grade	Midterm	Final
I.	100	C413			

Note : 기본 키가 null이어서는 안됨

◆ 튜플 검색을 이용한 삽입

- 4학년 학생의 학번을 학생테이블로부터 검색해서 SENIOR 테이블에 삽입하라.

SENIOR	Sno
I.	_STX

STUDENT	Sno	Sname	Year	Dept
	_STX		4	

▶ 데이터의 삭제

◆ 한 테이블에서의 삭제

- 학번이 100인 학생을 학생 테이블에서 삭제

STUDENT	Sno	Sname	Year	Dept
D.	100			

◆ 복수 테이블에서의 레코드 삭제

- 기말성적이 60점 미만인 학생을 등록 테이블과 학생테이블에서 삭제

ENROL	Sno	Cno	Grade	Midterm	Final
D.	_STX _STX				<60

STUDENT	Sno	Sname	Year	Dept
D.	_STX			

▶ 데이터의 갱신

◆ 필드 값의 단순 갱신

- 학번이 300인 학생의 학년을 2로 변경

STUDENT	Sno	Sname	Year	Dept
	300		U.2	

STUDENT	Sno	Sname	Year	Dept
U.	300		2	

◆ 산술식을 이용한 갱신

- 과목 'C413'에 등록한 학생의 기말 성적(Final)에 5점을 가산

ENROL	Sno	Cno	Final	Midterm
U.		C413	_G+5 _G	