

관계 데이터 연산

- 관계 데이터 언어 (사용자)
 - 관계 대수 (Relational Algebra)
 - 절차언어 - How, What
 - 관계 해석 (Relational Calculus)
 - 비절차 언어 - What
 - 튜플 관계 해석
 - 도메인 관계 해석
- 관계 해석과 관계 해석은 표현/기능 면에서 동등
- 상용 관계 데이터 언어
 - 관계 대수와 관계 해석을 기초로 함
 - SQL, QUEL, QBE

관계 대수

- 릴레이션 조작을 위한 **연산의 집합**
- 폐쇄 성질
 - 피연산자와 연산 결과가 모두 릴레이션
 - 중첩된 수식의 표현이 가능함
- 구성
 - 릴레이션 : 튜플의 집합
 - 일반 집합 연산자 : UNION / INTERSECTION / DIFFERENCE / CARTESIAN PRODUCT
 - 순수 관계 연산자 : SELECT / PROJECT / JOIN / DIVISION

일반 집합 연산자

- 합집합(UNION)
 - 정의 : $R \cup S = \{t \mid t \in R \vee t \in S\}$
 - Cardinality : $\max \{|R|, |S|\} \leq |R \cup S| \leq |R| + |S|$
- 교집합(INTERSECT)
 - 정의 : $R \cap S = \{t \mid t \in R \wedge t \in S\}$
 - Cardinality : $0 \leq |R \cap S| \leq \min \{|R|, |S|\}$
- 차집합(DIFFERENCE)
 - 정의 : $R - S = \{t \mid t \in R \wedge t \notin S\}$
 - Cardinality : $0 \leq |R - S| \leq |R|$
- 카티션 프로덕트(CARTESIAN PRODUCT)
 - 정의 : $R \times S = \{r \circ s \mid r \in R \wedge s \in S\}$ // \circ 는 접속(Concatenation)을 의미
 - Cardinality : $|R \times S| = |R| * |S|$
 - 차수 : R의 차수 + S의 차수 -> 애트리뷰트의 개수를 의미함
- 합병 가능한 릴레이션 : UNION / INTERSECT / DIFFERENCE의 피연산자들은

- 차수가 같아야 하고
- 대응 애트리뷰트 별로 도메인이 같아야 함
- 결합적 연산 : UNION / INTERSECT / CARTESIAN PRODUCT
 - 결합법칙이 적용됨 $(A \cup B) \cup C = A \cup (B \cup C)$
 - 교환법칙이 적용됨 $A \cup B = B \cup A$

순수관계 연산자

- Symbolic Notations
 - 릴레이션 $R : R(X), X = \{A_1 \sim A_n\} \rightarrow R(A_1, \dots, A_n)$
 - R 의 튜플 $r : r \in R, r = \langle a_1, \dots, a_n \rangle$
 - 튜플 r 의 애트리뷰트 A_i 의 값 : $A_i = r[A_i] = a_i$
- SELECT(σ)
 - 정의 : A, B 가 릴레이션 $R(X)$ 의 애트리뷰트 일 때
 - $\sigma_A \Theta v(R) = \{r \mid r \in R \wedge r.A \Theta v\}$
 - $\sigma_{A \Theta B}(R) = \{r \mid r \in R \wedge r.A \Theta r.B\}$
 - $\Theta = \{<, >, \leq, \geq, =, \neq\}$, v 는 상수
 - 선택 조건을 만족하는 릴레이션의 수평적 부분집합
 - (예시) σ 학과 = '컴퓨터' (학생)
 - 학생 릴레이션에서 학과 애트리뷰트가 컴퓨터인 튜플들의 릴레이션
 - (예시) σ 학번 = 300 \wedge 과목번호 = 'C312' (등록)
 - 등록 릴레이션에서 학번이 300 또는 과목번호가 C312인 튜플들의 릴레이션
 - (예시) σ 중간성적 < 기말성적 (등록)
 - 등록 릴레이션에서 중간성적 < 기말성적인 튜플들의 릴레이션
- 데이터 언어식에서의 표현 : WHERE
- 교환법칙 성립
- 선택도 : 조건식에 의해 선택된 튜플들의 비율

- PROJECT(Π)

- 정의

릴레이션 $R(X), X = \{A_1, A_2, \dots, A_n\}$ 에서
애트리뷰트 집합 Y 가 $Y \subseteq X$, $Y = \{B_1, B_2, \dots, B_m\}$, $n \geq m$ 이면,

$$\Pi_Y(R) = \{ \langle r.B_1, \dots, r.B_m \rangle \mid r \in R \}$$

- 학생(학번, 이름, 학년)에서 Π _이름, 학번(학생)
 - 학생 테이블에서 이름, 학번 애트리뷰트 값만 가져온 릴레이션
- 릴레이션의 수직적 부분집합
- 생성된 중복 튜플은 제거
- $\Pi_y(\Pi_x(R)) = \Pi_y(R)$
- (예시) 학생 테이블에서 CS학과 3학년의 학번과 이름을 검색
 - Π 이름, 학번(σ 학과 = 'CS' \wedge 학년 = 3 (학생))
 - SELECT부터 하고 PROJECT 해야 함 순서가 바뀌면, SELECT의 조건식을 적용 할 수 없음

- JOIN(\bowtie)

- 세타 조인

$R(X), S(Y), A \in X, B \in Y$ 에 대하여

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

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

- 결과 차수 = R의 차수 + S의 차수
- A와 B는 조인 애트리뷰트, $A \Theta B$ 는 조인 조건식

- 동일 조인

세타조인에서 Θ 가 " $=$ "인 경우

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

- A와 B는 조인 애트리뷰트, $A=B$ 는 조인 조건식
- 결과 차수 = R의 차수 + S의 차수
- 결과의 Cardinality : $0 \leq |R \bowtie_{A=B} S| \leq |R| \times |S|$
- (예시) 학생 $\bowtie_{\text{학번}=\text{학번}}$ 등록

▪ 학생 릴레이션의 튜플의 학번 = 등록 릴레이션의 튜플의 학번인 애들을 합쳐서 릴레이션 짬뽕

- 자연 조인(\bowtie_N)

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

$$R \bowtie_N S = \{ \langle r \cdot s \rangle [X \cup Y] \mid r \in R \wedge s \in S \wedge r[Z] = s[Z] \} \\ = \Pi_{X \cup Y} (\sigma_{Z=Z}(R \times S)) \\ = \Pi_{X \cup Y} (R \bowtie_{Z=Z} S)$$

- 동일조인의 결과 릴레이션에서 애트리뷰트의 중복을 제거
- 결과 차수 = R의 차수 + S의 차수 - $|X \cap Y|$
- $Z (= X \cap Y)$ 는 조인 애트리뷰트, $Z=Z$ 는 조인 조건식

- DIVIDE(\div)

- 정의

릴레이션 $R(X), S(Y)$ 에 대하여

$Y \subseteq X$ 이고 $X-Y=Z$ 이면, $R(X)=R(Z,Y)$ 로 표현 가능.

따라서 $R(Z,Y), S(Y)$ 에 대하여

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

- 나눠지는 애가 가지고 있는 애트리뷰트 값들을 보유하고 있는 튜플들
- 차수는 $X-Y$ 가 됨

- RENAME(ρ)

- 중간결과 릴레이션에 이름을 붙이거나 애트리뷰트 이름을 변경할 때 사용
 - (예시) $\rho_s(E)$: 관계 대수식 E의 결과 릴레이션의 이름을 s로 지
 - (예시) $\rho_s(B_1, \dots, B_m)(E)$: 관계 대수식 E의 결과 릴레이션의 이름을 S로 하면서 애트리뷰트의 이름을 $B_1 \sim B_m$ 으로 지정

근원연산과 복합 연산

- 근원연산 : UNION, DIFFERENCE, CARTESIAN PRODUCT, SELECT, PROJECT
- 복합연산 : INTERSECT, JOIN, DIVISION
- 복합연산은 근원연산으로 표현 가능함

$$\begin{aligned} R \cap S &= R - (R - S) = S - (S - R) \\ &= (R \cup S) - ((R - S) \cup (S - R)) \\ R \bowtie_{A \theta B} S &= \sigma_{A \theta B} (R \times S) \\ R \bowtie_{A=B} S &= \sigma_{A=B} (R \times S) \\ R \bowtie_N S &= \Pi_{X \cup Y} (\sigma_{Z=Z} (R \times S)) = \Pi_{X \cup Y} (R \bowtie_{Z=Z} S), \text{ 단 } Z=X \cap Y \\ R(Z, Y) \div S(Y) &= R[Z] - ((R[Z] \times S) - R)[Z] \end{aligned}$$

관계대수의 확장

- SEMIJOIN(\bowtie, \bowtie_x)

- 정의

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

$$R \bowtie S = R \bowtie_N (\Pi_{X \cap Y} (S)) = \Pi_X (R \bowtie_N S)$$

- S 와 자연조인을 할 수 있는 R 의 튜플(기호가 뒤집히면 반대)
 - 특징

$$\begin{aligned} R \bowtie S &\neq S \bowtie R \\ R \bowtie_N S &= (R \bowtie S) \bowtie_N S = (S \bowtie R) \bowtie_N R \\ &= (R \bowtie S) \bowtie_N (S \bowtie R) \end{aligned}$$

- 분산환경($R =$ 서울, $S =$ 부산)

$$R \bowtie_N S = (R \bowtie S) \bowtie_N S = (R \bowtie_N (\Pi_{X \cap Y} (S))) \bowtie_N S$$

- ◆ 질의: 부산

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

- ◆ 질의: 서울

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

- ◆ 질의: 제3의 장소

- OUTERJOIN($\bowtie+$)

- 정의 : 조인시 한 릴레이션에 있는 튜플이 조인할 상대 릴레이션에 대응되는 튜플이 없을 경우, 상대를 **NULL 튜플로 만들어** 결과 릴레이션에 포함
 - 두 조인 릴레이션의 모든 튜플들이 결과 릴레이션에 포함됨
 - Left Outer Join : R 기준, Right Outer Join : S 기준, Full Outer Join : 둘다 포함시킴

- OUTER-UNION($\cup+$)

- 정의 : 합병가능하지 않은(부분적으로만 가능한) 두 릴레이션을 차수를 확장시켜 합집합으로 만듬
- 집단연산
 - 수학적 집단 연산 : SUM, AVG, MAX, MIN, COUNT
 - 중복값이 있더라도 제외하지 않고 그대로 적용
 - NULL값의 처리에 주의
- 그룹연산
 - Group 연산
 - 주어진 기준 애트리뷰트의 값이 같은 튜플들끼리 그루핑
 - 내부적으로는 애트리뷰트 값을 기준으로 정렬한 것과 같은 효과
 - GROUP_학년(학생) : 학생 릴레이션을 학년 기준으로 그룹 -> 정렬
- 일반형식

일반형식: $F_B(GROUP_A(E))$

- ◆ E : 관계 대수식
- ◆ G : 그룹연산
- ◆ F : 집단 함수 (SUM, AVG, MAX, MIN, COUNT)
- ◆ B : 집단 함수의 적용 대상 애트리뷰트
- ◆ A : 그룹 함수가 적용할 기준 애트리뷰트

먼저 그룹 연산을 적용한후, 각 그룹에 대해 집단 함수를 적용.

- ◆ $AVG_{나이}(GROUP_{학년}(학생))$
그룹을 연속적으로 적용 가능
- ◆ $AVG_{나이}(GROUP_{학년}(GROUP_{소속학부}(학생)))$