

1

이항 관계 연산

01 이항 관계 연산

1 이항 연산 개요

- 피 연산자가 두 개인 관계 연산
- 조인 연산
- 세타, 동등, 자연 조인
- 완전 집합
- 디비젼 연산

2

조인 연산과 종류

02 조인 연산과 종류

1 조인 연산

- 두 릴레이션으로부터 관련 있는 투플들을 결합하여 하나의 투플로 생성함
- 관련성의 여부를 조건으로 표시하며 이를 조인 조건이라고 함
- 연산 형식 : $R1 \triangleright \triangleleft_{\langle \text{join condition} \rangle} R2$

02 조인 연산과 종류

1 조인 연산



- DEPT $\triangleright \triangleleft$ MGRSSN=SSN EMP

[DEPT]

| DNAME | DNUMBER | MGRSSN | MGRSTARTDATE |
|----------------|---------|-----------|--------------|
| Research | 5 | 333445555 | 22-MAY-78 |
| Administration | 4 | 987654321 | 01-JAN-85 |
| Headquarters | 1 | 888665555 | 19-JUN-71 |

※ 출처 : 데이터베이스 시스템 6판, Elmasri, Navathe 저, 황규영 외 역, 흥릉과학출판사, 2016년

02 조인 연산과 종류

1 조인 연산



- DEPT $\triangleright \triangleleft$ MGRSSN=SSN EMP

[EMP]

| FNAME | MINIT | LNAME | SSN | BDATE | ADDRESS | SEX | SALARY | SUPERSSN | DNO |
|----------|-------|---------|-----------|-----------|--------------------------|-----|--------|-----------|-----|
| John | B | Smith | 123456789 | 09-JAN-55 | 731 Fondren, Houston, TX | M | 30000 | 333445555 | 5 |
| Franklin | T | Wong | 333445555 | 08-DEC-45 | 638 Voss, Houston, TX | M | 40000 | 888665555 | 5 |
| Alicia | J | Zelaya | 999887777 | 19-JUL-58 | 3321 Castle, Spring, TX | F | 25000 | 987654321 | 4 |
| Jennifer | S | Wallace | 987654321 | 20-JUN-31 | 291 Berry, Bellaire, TX | F | 43000 | 888665555 | 4 |
| Ramesh | K | Narayn | 666884444 | 15-SEP-52 | 975 Fire Oak, Humble, TX | M | 38000 | 333445555 | 5 |
| Joyce | A | English | 453453453 | 31-JUL-62 | 5631 Rice, Houston, TX | F | 25000 | 333445555 | 5 |
| Ahmad | V | Jabbar | 987987987 | 29-MAR-59 | 980 Dallas, Houston, TX | M | 25000 | 987654321 | 4 |
| James | E | Borg | 888665555 | 10-NOV-27 | 450 Stone, Houston, TX | M | 55000 | null | 1 |

※ 출처 : 데이터베이스 시스템 6판, Elmasri, Navathe 저, 황규영 외 역, 흥릉과학출판사, 2016년

02 조인 연산과 종류

2 세타, 동등, 자연, 자체 조인

조인 조건

- <조건> AND <조건> AND … AND <조건>
- 각 조건의 형태는 $A_i \Theta B_j$
- 조인 조건에 사용되는 속성 A_i 와 B_j 를
조인 속성이라고 부름

세타 조인(Theta Join)

- 일반적인 조인 조건(>, =, < 등)을 가진 조인 연산

02 조인 연산과 종류

2 세타, 동등, 자연, 자체 조인

동등 조인(Equi. Join)

- 조인 조건에서 동등 비교(Equality comparison)
만을 사용하는 조인
- 동등 조인 사용 예
: 모든 부서의 부서명과 관리자의 이름을 검색하라
 - $T \leftarrow DEPT \bowtie MGRSSN=SSN EMP$
 - $RESULT \leftarrow \Pi_{DNAME,FNAME,LNAME}(T)$

02 조인 연산과 종류

2 세타, 동등, 자연, 자체 조인

○ 자연 조인(Natural Join)

- 동등 조인의 결과에는 두 조인 속성의 값이 중복되어 나타남
- 조인 결과에서 조인 속성 하나를 제거하여 중복된 값이 나타나지 않도록 한 조인을 자연조인이라고 함
- 표시법 : $R \leftarrow R1 *_{(R1의 조인 속성),(R2의 조인 속성)} R2$

02 조인 연산과 종류

2 세타, 동등, 자연, 자체 조인

○ 자연 조인(Natural Join)

- 예제)

모든 사원의 이름과
그가 소속된 부서의 이름을 검색하라

- $T \leftarrow EMP *_{(DNO), (DNUMBER)} DEPT$
- $RESULT \leftarrow \Pi_{FNAME, LNAME, DNAME} (T)$
- 두 조인 속성이 동일한 이름을 갖는다면
간단히 $R1 * R2$ 라고 표시함

02 조인 연산과 종류

2 세타, 동등, 자연, 자체 조인

○ 자체 조인(Self Join)

- 하나의 릴레이션에 대한 조인
- 자체 조인은 한 릴레이션의 서로 다른 두 사본을 조인하는 것으로 간주함
- 이 경우, 사본 릴레이션에서는 원본 애트리뷰트 이름을 재명명(Renaming)하는 것이 유용함

02 조인 연산과 종류

2 세타, 동등, 자연, 자체 조인

○ 자체 조인(Self Join)

- 예제)

모든 사원의 이름과 그들의 상사 이름을 검색하라

- $\text{SUPERVISOR}(\text{SSSN}, \text{SFN}, \text{SLN}) \leftarrow \Pi_{\text{SSN}, \text{FNAME}, \text{LNAME}} (\text{EMP})$
- $T \leftarrow \text{EMP} \triangleright \triangleleft_{\text{SUPERSSN}=\text{SSSN}} \text{SUPERVISOR}$
- $\text{RST} \leftarrow \Pi_{\text{FNAME}, \text{LNAME}, \text{SFN}, \text{SLN}} (T)$

02 조인 연산과 종류

3 관계대수 연산의 완전 집합

- 🔍 최소한의 연산자 집합
: 지금까지 소개한 모든 연산자는
 σ , π , \cup , $-$, \times 만의 조합으로 표현할 수 있음
- 🔍 연산자 집합 $\{\sigma, \pi, \cup, -, \times\}$ 를 관계대수
연산자의 완전 집합(Complete set)이라 부름
- 🔍 이 연산자 집합과 동등한 모든 질의 언어들은
관계적으로 완전하다(Relationally complete)라고
정의함

02 조인 연산과 종류

3 관계대수 연산의 완전 집합

🔍 추가적으로 유용한 연산자들

- 디비젼 연산
- 집계 함수와 그룹화 연산
- 외부 조인과 외부 합집합

③

디비젼 연산

03 디비젼 연산

1 디비젼(Division)

- 기본 연산자는 아니지만
- A 는 x 와 y, B 는 y 로 구성되었을 때
“모든 y 에 해당되는 x 를 구하라” 같은 경우에
디비젼을 사용
- 즉, A / B 는 B의 모든 $\langle y \rangle$ 에 대하여,
 A 내에 $\langle x, y \rangle$ 가 존재하는 그러한 x 들의 모임임

03 디비젼 연산

1 디비젼(Division)

🔍 예제) division 연산

| sno | pno |
|-----|-----|
| s1 | p1 |
| s1 | p2 |
| s1 | p3 |
| s1 | p4 |
| s2 | p1 |
| s2 | p2 |
| s3 | p2 |
| s4 | p2 |
| s4 | p4 |

A

| pno |
|-----|
| p2 |

B1

| pno |
|-----|
| p2 |
| p4 |

B2

| pno |
|-----|
| p1 |
| p2 |

B3

| sno |
|-----|
| s1 |
| s2 |
| s3 |
| s4 |

A/B1

| sno |
|-----|
| s1 |
| s2 |
| s3 |

A/B2

| sno |
|-----|
| s1 |

A/B3

※ 출처 : 데이터베이스 시스템 3판, Ramakrishnan, Gehrke 저, 김수희 외 역, 한티미디어, 2003년

03 디비전 연산

2 추가적인 연산들

🔍 집계 함수(Aggregate Functions)

- SUM, COUNT, AVERAGE, MIN, MAX 함수를 의미함
- 표준 관계 대수로는 표현할 수 없음
- 다음과 같이 표현하며 그룹화 속성들은 선택적임
- <그룹화 속성들> F <함수 리스트> (R)

03 디비전 연산

2 추가적인 연산들

○ 예제 1)

모든 사원의 평균 봉급을 검색하라(그룹화 불필요)

$R(\text{AVGSAL}) \leftarrow F_{\text{AVERAGE SALARY}} (\text{EMPLOYEE})$

○ 예제 2)

각 부서에 대하여, 부서 번호와 부서별 사원 수와
평균 봉급을 검색하라(부서별 그룹화)

$R(\text{DNO}, \text{NUMEMPS}, \text{AVGSAL}) \leftarrow$
 $DNO F_{\text{COUNT SSN, AVERAGE SALARY}} (\text{EMPLOYEE})$

03 디비전 연산

2 추가적인 연산들

[DNO 별로 EMPLOYEE 투플들을 그룹화]

The diagram illustrates the grouping of employee tuples by DNO. On the left, a source table 'EMPLOYEE' lists employees with their first name (FNAME), middle initial (MINIT), last name (LNAME), Social Security Number (SSN), salary (SALARY), super SSN (SUPERSSN), and department number (DNO). The DNO column contains values 5, 5, 5, 5, 4, 4, 4, and 1 respectively. A vertical ellipsis between the 5th and 6th rows indicates that there are more rows in the source table. On the right, a result table shows the grouped data. It has three columns: DNO, COUNT(*), and AVG(SALARY). The result table shows four groups: one group for DNO 5 with 4 employees and an average salary of 33250, another for DNO 4 with 3 employees and an average salary of 31000, and two groups for DNO 1, each with 1 employee and an average salary of 55000.

| EMPLOYEE | FNAME | MINIT | LNAME | SSN | ... | SALARY | SUPERSSN | DNO |
|----------|----------|-------|---------|-----------|-----|--------|-----------|-----|
| | John | B | Smith | 123456789 | | 30000 | 333445555 | 5 |
| | Franklin | T | Wong | 333445555 | | 40000 | 888665555 | 5 |
| | Ramesh | J | Narayn | 666884444 | | 38000 | 333445555 | 5 |
| | Joyce | S | English | 453453453 | ... | 25000 | 333445555 | 5 |
| | Alicia | K | Zelaya | 999887777 | | 25000 | 987654321 | 4 |
| | Jennifer | A | Wallace | 987654321 | | 43000 | 888665555 | 4 |
| | Ahmad | V | Jabbar | 987987987 | | 25000 | 987654321 | 4 |
| | James | E | Borg | 888665555 | | 55000 | null | 1 |

| DNO | COUNT(*) | AVG(SALARY) |
|-----|----------|-------------|
| 5 | 4 | 33250 |
| 4 | 3 | 31000 |
| 1 | 1 | 55000 |

※ 출처 : 데이터베이스 시스템 6판, Elmasri, Navathe 저, 황규영 외 역, 홍릉과학출판사, 2016년

03 디비전 연산

2 추가적인 연산들

🔍 순환적 폐포(Recursive Closure) 연산

- 동일한 테이블에서 투플들 간의 순환적 관계를 질의하는데 사용됨
- 관계 대수로서는 표현할 수 없음
- 예) Employee 테이블에서
특정 사원의 모든 상사 직원을 모두 검색하시오
- 이러한 질의는 반복 구조(루프)을 사용하여
1단계 위 상사들의 집합을 구하고, 이를 바탕으로
2단계 위 상사를 구하며, 이러한 과정을 더 이상의
상사 집합이 없을 때까지 구해나가야 하므로 반복
처리가 필요하게 됨

03 디비전 연산

2 추가적인 연산들

외부 조인(Outer Join) 연산

- 동등 조인이나 자연 조인 연산에서 조인 조건을 만족하지 않은 투플들은 결과 릴레이션에 나타나지 않음
- 조인에 참여하는 릴레이션의 모든 투플들을 조인의 여부와 관계없이 결과 릴레이션에 나타내고 싶은 경우 외부 조인을 사용함
- 외부 조인에서는 상대방 릴레이션에 대응되는 투플이 없으면 빈 속성에 NULL을 채워서 결과에 포함시킴

03 디비전 연산

2 추가적인 연산들

외부 조인(Outer Join) 연산

- LEFT OUTER JOIN
 - : R1 \bowtie R2는 R1의 모든 투플들이 결과 릴레이션에 나타나도록 함
- RIGHT OUTER JOIN)
 - : R1 \bowtie R2는 R2의 모든 투플들이 결과 릴레이션에 나타나도록 함
- FULL OUTER JOIN)
 - : R1 \bowtie R2는 R1과 R2의 모든 투플들이 결과 릴레이션에 나타나도록 함

03 디비전 연산

2 추가적인 연산들

외부 합집합(Outer Union) 연산

- 호환성이 없는 두 릴레이션을 합집합하는데 사용됨
- 예제)

STUDENT(Name, SSN, Department, Advisor)와
FACULTY(Name, SSN, Department, Rank)의
OUTER UNION 은 T(Name, SSN, Department,
Advisor, Rank)

- T 에서 STUDENT 투플의 Rank 값은 null 이고
- FACULTY 투플의 Advisor 값도 null 임