CS3DB3/SE4DB3/SE6DB3 TUTORIAL

Xiao Jiao Wang

Feb 27, 2016

SQL and relational algebra

- **SELECT** A1, A2, …, An
  FROM R1, R2, …, Rm
  WHERE P

is equivalent to the multiset relational algebra expression

Don't forget the parenthesis since σ has a higher Precedence than [×, ⋈]

$$\prod_{A1, A2, \cdots, An} (\sigma_P (R1 \times R2 \times \cdots \times Rm))$$

SQL and relational algebra (Cont.)

- Example 1

  Takes (id, course_id, semester, year, grade)

  Teaches(name, course_id, semester, year)

  - Find IDs of all students who were taught by an instructor named Jones.
  - SQL
    - **SELECT** id

      FROM Takes, Teaches

      WHERE name = 'Jones' AND Takes.course_id = Teaches.course_id;
  - Relation algebra
    - WAY 1: $\Pi_{id}(\sigma_{name='Jones'}(Takes \bowtie Teaches))$

      Because of common attribute
    - WAY 2: $\Pi_{Teaches.course\_id}(\sigma_{name='Jones' \wedge Takes.course\_id = Teaches.course\_id}(Takes \times Teaches))$

SQL and relational algebra (Cont.)

- Example 2
  - **Works** (pname, cname, salary)
  - Find the names of all employees who earn more than every employee of "First Bank".
  - SQL

```
SELECT pname
FROM Works
WHERE salary >ALL (SELECT salary
                   FROM Works
                   WHERE cname= 'First Bank');
```

  - Relational algebra

$$R1 := \Pi_{w1.pname}(\rho_{w1}(Works) \bowtie_{w1.salary \leqslant w2.salary \wedge w2.cname='First\ Band'} \rho_{w2}(Works))$$

Assignment:
create temporary relation names

$$Result := \Pi_{pname}(Works) - R1$$

SQL and relational algebra (Cont.)

- **SELECT** A1, A2, **AGG**(A3) AS AGG3
  FROM R1, R2,…, Rm
  WHERE P
  GROUP BY A1, A2
  - Is equivalent to the multiset relational algebra expression

$$\gamma_{A1,A2,AGG(A3)\rightarrow AGG3}(\sigma_P(R1\times R2\times\ldots\times Rm))$$

- If only display attribute A1 and AGG3, then

$$\Pi_{A1,AGG3}(\gamma_{A1,A2,AGG(A3)\rightarrow AGG3}(\sigma_P(R1\times R2\times\ldots\times Rm)))$$

SQL and relational algebra (Cont.)

- Example 3
  - **Takes (student_id, course_id, semester, year, grade)**
  - Find the enrollment of each course that was offered in Fall 2009.
  - SQL

```
SELECT course_id, count(*) as enrollment
FROM Takes
WHERE year=2009 AND semester='Fall'
GROUP BY course_id;
```

  - Relational Algebra

$$\gamma_{\text{course\_id, count(*)}\rightarrow\text{enrollment}}\left(\sigma_{\text{year=2009} \wedge \text{semester="Fall"}}(\text{Takes})\right)$$

SQL and relational algebra (Cont.)

- Example 4
  - **Takes (student_id, course_id, semester, year, grade)**
  - Find the maximum enrollment in Fall 2009.
  - SQL

```
SELECT MAX(enrollment)
FROM (SELECT course_id, count(*) as enrollment
        FROM Takes
        WHERE year=2009 AND semester='Fall'
        GROUP BY course_id);
```

  - Relational Algebra

$$R := \gamma_{\text{course\_id, count(*)} \rightarrow \text{enrollment}} (\sigma_{\text{year}=2009 \wedge \text{semester}=\text{“Fall”}}(\text{Takes}))$$
$$\text{Result} := \gamma_{\text{max(enrollment)}}(R)$$