The Relational Algebra

CS157A
Chris Pollett
Oct. 3, 2005.

Outline

- Overview of the Relation Algebra
- Select Operations
- Project Operations
- Composition and Rename Operations
- Union, Intersection and Minus
- Cartesian Product
- Join

Overview of the Relation Algebra

- We now discuss how retrieval of data stored according to the relational model can be done.
- There are actually two approaches: the relational algebra (functional) and the relational calculus (logic-based, so order of operation somewhat less emphasized).
- The relational algebra is vaguely behind SQL’s query language.
- The relational calculus is vaguely behind schemes like Query By Example.

Select Operations

- Used to select a subset of the tuples from a relation that satisfy a **selection condition**.

$$\sigma_{DNO=4}(EMPLOYEE)$$

$$\sigma_{(DNO=4 \text{ AND } SALARY>25000) \text{ OR } DNO=5}(EMPLOYEE)$$

- Notice atomic condition of the form:
  <attribute name><comparison op><constant value>
  <attribute name><comparison op><attribute name>
- More complicated expressions can be built from these using AND, OR, NOT.

Properties of Select

$$\sigma_{<cond1>}(\sigma_{<cond2>}(R)) = \sigma_{<cond2>}(\sigma_{<cond1>}(R))$$

(Commutative)

$$\sigma_{<cond_1>}(\sigma_{<cond_2>}(...\sigma_{<cond_n>}(R)...)) = \sigma_{<cond_1> \text{ AND } ... \text{ AND} <cond_n>}(R)$$

(Cascade)

Project Operations

- This operation selects certain columns from the table and discards all other columns.

$$\pi_{LNAME,FNAME,SALARY}(EMPLOYEE)$$

$$\pi_{<attribute\ list>}(R)$$

- Note; if project on non-key attributes, duplicate tuples might occur. Project, however, gets rid of duplicates. (**Duplicate elimination**).

$\pi_{<list1>}(\pi_{<list2>}(R)) = \pi_{<list1>}(R)$ if $< list1 >$ is contained in $< list2 >$.

Composition and Rename Operations

- We can create relational algebra expressions from our relational value operations using composition:

$$\pi_{FNAME,LNAME,SALARY}(\sigma_{DNO=5}(EMPLOYEE))$$

- Alternatively, we can explicitly show intermediate results:

$$DEP5\_EMPS \leftarrow \sigma_{DNO=5}(EMPLOYEE)$$
$$RESULT \leftarrow \pi_{FNAME,LNAME,SALARY}(DEP5\_EMPS)$$

- We can do renaming of columns either via the intermediate table way or with a RENAME operation:

$$R(FIRST, LAST, SAL) \leftarrow \pi_{FNAME,LNAME,SALARY}(EMP)$$
$$\rho_{R(FIRST,LAST,SAL)}(EMP)$$

Union, Intersection and Minus

- The relational algebra also allows certain set theoretic operations:
  - Union: $R \cup S$ returns in a relation those tuples which are either in R or in S
  - Intersection: $R \cap S$ returns in a relation those tuples which are in both R and S.
  - Difference: $R \not S$ returns is a relation those tuples of R which are not in S.
- To work, the relations R and S must have compatible attributes.
- Union and Intersection are commutative. i.e., $R \cup S = S \cup R$ and $R \cap S = S \cap R$.
- Set difference is not commutative.

Cartesian Product

- We now consider the binary operation $R(A_1 \ldots A_n) \times S(B_1, \ldots, B_m)$.
- This relation contains all tuples of the form: $(t[A_1], \ldots, t[A_n], s[B_1], \ldots, s[B_m])$ where t is a tuple in the instance of R and s is a tuple from the instance of S.

Join

- Join is a useful combination of both a select operation and a cartesian product operation:

$$R \bowtie_{<join\ condition>} S := \sigma_{<join\ condition>} R \times S$$

- Implementation ways these two operations can often be done faster together.
- The typical condition is usual an equality between attributes: $DEPT \bowtie_{MGRSSN=SSN} EMP$
- If the join involves a more general selection then it is called a theta-join.

Equijoins and Natural Joins

- If the join condition involves only equalities of attributes, it is called an equijoin.
- If we delete the duplicate columns in the result of an equijoin, we get a join called a natural join.
- We write R*S for the natural join of R and S.
- Notice if we don't list the joined attributes, it is assumed we are joining attributes with the same name in both relations.