CSc 134 Database Management Systems

5. Relational Algebra

Ying Jin
Computer Science Department
California state University, Sacramento

Relational Algebra

- A set of operations for the relational model.
- Enable a user to specify basic retrieval requests.
- The **algebra operations** produce new relations.
  - The result of a retrieval is a new relation.
- A sequence of relational algebra operations forms a **relational algebra expression**
  - result
    - a relation
    - represents the result of a database query.

Topics on relational algebra

- Select
- Project
- Union
- Intersection
- Minus
- Cartesian product
- Join
- Natural join

The SELECT Operation

- $\sigma_{\text{<selection condition>}}(R)$
- Filter - only those tuples that satisfy a qualifying condition appear in the result.
- Result: *subset* of the tuples
- Examples
- The $\leftarrow$ symbol is an assignment operator

EMPLOYEE	FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John		Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin		Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia		Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer		Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh		Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce		English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad		Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James		Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

The SELECT Operation (Cont.)

- Commutative

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

- A cascaded SELECT operation *may be applied in any order*

$\sigma_{<condition1>}(\sigma_{<condition2>} (\sigma_{<condition3>} (R))$

$= \sigma_{<condition2>} (\sigma_{<condition3>} (\sigma_{<condition1>} (R)))$

- Cascade of SELECT operations into a single SELECT operation

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

$\sigma_{<cond1> \text{ and } <cond2> \text{ and} \cdots \text{ and } <condn>}(R)$

The Project Operation

- This operation selects certain *columns* from the table and discards the other columns.
- Creates a vertical partitioning –
  - one with the needed columns (attributes) containing results of the operation
  - other containing the discarded Columns.
- $\pi_{\text{<list>}}(R)$
- Example

The Project Operation (Cont.)

- π *removes any duplicate tuples*
- The result of π is a set of tuples – a valid relation
- $\pi_{\textbf{sex, salary}}$(EMPLOYEE)
- The number of tuples in the result of projection $\pi_{<list>}(R)$ is always less or equal to the number of tuples in R.
- $\pi_{<list1>}(\pi_{<list2>}(R)) = \pi_{<list1>}(R)$

Sequence of Operations

- Relational algebra expression

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

- Intermediate results

e.g. $TEMP \leftarrow \sigma_{DNO=5}(EMPLOYEE)$

$RESULT \leftarrow \pi_{FNAME, LNAME, SALARY}(TEMP)$

Rename Operator: ρ

- $\rho_{S\ (B1,\ B2,\ \ldots,\ Bn\ )}(R)$ changes both:
  - the relation name to S, *and*
  - the column (attribute) names to B1, B1, ...Bn
- $\rho_{S}(R)$ changes:
  - the *relation name* only to S
- $\rho_{(B1,\ B2,\ \ldots,\ Bn\ )}(R)$ changes:
  - the *column (attribute) names* only to B1, B1, ...Bn

Rename (cont.)

- $R \leftarrow \pi_{FNAME,\ LNAME,\ SALARY}$ (EMPLOYEE)
- $\rho_{S\ (FN,\ LN, SAL\ )}(R)$
- $\rho_{(FN,\ LN, SAL\ )}(R)$
- $\rho_{S}\ (R)$

UNION

- $R \cup S$
- includes all tuples that are either in R or in S or in both R and S.
- Duplicate tuples are eliminated.
- **Example:** To retrieve the social security numbers of all employees who either work in department 5 or directly supervise an employee who works in department 5:

Union Example

RESULT1	SSN
	123456789
	333445555
	666884444
	453453453

RESULT2	SSN
	333445555
	888665555

RESULT	SSN
	123456789
	333445555
	666884444
	453453453
	888665555

Union Compatibility

- The operand relations $R_1(A_1, A_2, ..., A_n)$ and $R_2(B_1, B_2, ..., B_n)$ must
  - have the same number of attributes, AND
  - the domains of corresponding attributes must be compatible: $dom(A_i)=dom(B_i)$ for $i=1, 2, ..., n$.

Intersection

- $R \cap S$
- includes all tuples that are in both R and S
- The two operands must be Union compatible

Set Difference (MINUS)

- R – S
- The two operands must be Union compatible
- Result: a relation that includes all tuples that are in R but not in S

Commutative and associative

- Union and Intersection are *commutative operations*

  R ∪ S = S ∪ R, and R ∩ S = S ∩ R

- Union and intersection are *associative operations*

  R ∪ (S ∪ T) = (R ∪ S) ∪ T, and (R ∩ S) ∩ T = R ∩ (S ∩ T)

- The minus operation is *not commutative*

  R - S ≠ S – R

Cartesian Product

- R X S
- Combine tuples from two relations in a combinatorial fashion
- $Q(A_1, A_2, \ldots, A_n, B_1, B_2, \ldots, B_m)$ <- $R(A_1, A_2, \ldots, A_n) \times S(B_1, B_2, \ldots, B_m)$
  - m+n attributes
  - if R has n_R tuples (denoted as $|R| = n_R$), and S has n_S tuples, then

    Q have $n_R * n_S$ tuples.

Cartesian Product Example

- Retrieve a list of names each female employee’s dependents (employee’s first name, last name, dependent’s name

FEMALE_EMPS	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle,Spring,TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry,Bellaire,TX	F	43000	888665555	4
	Joyce	A	English	453453453	1972-07-31	5631 Rice,Houston,TX	F	25000	333445555	5

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

EMPNAMES	FNAME	LNAME	SSN
	Alicia	Zelaya	999887777
	Jennifer	Wallace	987654321
	Joyce	English	453453453

EMP_DEPENDENTS	FNAME	LNAME	SSN	ESSN	DEPENDENT_NAME	SEX	BDATE	• • •
	Alicia	Zelaya	999887777	333445555	Alice	F	1986-04-05	• • •
	Alicia	Zelaya	999887777	333445555	Theodore	M	1983-10-25	• • •
	Alicia	Zelaya	999887777	333445555	Joy	F	1958-05-03	• • •
	Alicia	Zelaya	999887777	987654321	Abner	M	1942-02-28	• • •
	Alicia	Zelaya	999887777	123456789	Michael	M	1988-01-04	• • •
	Alicia	Zelaya	999887777	123456789	Alice	F	1988-12-30	• • •
	Alicia	Zelaya	999887777	123456789	Elizabeth	F	1967-05-05	• • •
	Jennifer	Wallace	987654321	333445555	Alice	F	1986-04-05	• • •
	Jennifer	Wallace	987654321	333445555	Theodore	M	1983-10-25	• • •
	Jennifer	Wallace	987654321	333445555	Joy	F	1958-05-03	• • •
	Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28	• • •
	Jennifer	Wallace	987654321	123456789	Michael	M	1988-01-04	• • •
	Jennifer	Wallace	987654321	123456789	Alice	F	1988-12-30	• • •
	Jennifer	Wallace	987654321	123456789	Elizabeth	F	1967-05-05	• • •
	Joyce	English	453453453	333445555	Alice	F	1986-04-05	• • •
	Joyce	English	453453453	333445555	Theodore	M	1983-10-25	• • •
	Joyce	English	453453453	333445555	Joy	F	1958-05-03	• • •
	Joyce	English	453453453	987654321	Abner	M	1942-02-28	• • •
	Joyce	English	453453453	123456789	Michael	M	1988-01-04	• • •
	Joyce	English	453453453	123456789	Alice	F	1988-12-30	• • •
	Joyce	English	453453453	123456789	Elizabeth	F	1967-05-05	• • •

ACTUAL_DEPENDENTS	FNAME	LNAME	SSN	ESSN	DEPENDENT_NAME	SEX	BDATE
	Jennifer	Wallace	987654321	987654321	Abner	M	1942-02-28

RESULT	FNAME	LNAME	DEPENDENT_NAME
	Jennifer	Wallace	Abner

JOIN

- Example

EMP_DEPENDENTS $\leftarrow$ EMPNAMES x DEPENDENT

ACTUAL_DEPENDENTS $\leftarrow \sigma_{SSN=ESSN}$(EMP_DEPENTS)

Replace with a single JOIN operation

ACTUAL_DEPENDENTS $\leftarrow$ EMPNAMES $\bowtie_{SSN=ESSN}$ DEPENDENT

JOIN (Cont.)

- a join operation on two relations $R(A_1, A_2, ..., A_n)$ and $S(B_1, B_2, ..., B_m)$ is:

  $R \bowtie_{\text{<join condition>}} S$

  where R and S can be any relations that result from general *relational algebra expressions.*
- <condition> AND <condition> AND ... AND <condition>
- Each condition: Ai Ө Bj
  - Ai: an attribute of R
  - Bj: an attribute of S
  - Ai and Bj have the same domain
  - Ө: =, < ,>, ≠, ≥, ≤

EQUIJOIN

- The join conditions with "=" only
- e.g.

$$\textbf{DEPT\_MGR} \leftarrow \textbf{DEPARTMENT} \bowtie_{\textbf{MGRSSN=SSN}} \textbf{EMPLOYEE}$$

- The result of an EQUIJOIN:
  - Always have one or more pairs of attributes that have *identical values* in every tuple

Natural join

- *
- A equijoin without superfluous attributes
- Any two join attributes have the same name in both relations.
- Join attributes
- Equating all attributes pairs that have the same name in the two relations.
- Rename when necessary before applying nature join
- e.g. **DEPT_LOCS ← DEPARTMENT * DEPT_LOCATIONS**

Join Selectivity

$R \bowtie_{\text{<join condition>}} S$

- R has n_R tuples, S has n_S tuples
- Result:
  - min:empty relation with 0 tuples
    - No combination of tuples satisfies the join condition
  - max: $n_R * n_S$

Complete Set of Relational Operations

- $\sigma, \pi, \cup, -, X$
- Any other relational algebra expression can be expressed by a combination of these five operations
- Examples

$$R \cap S = (R \cup S) - ((R - S) \cup (S - R))$$

$$R \bowtie_{\langle \text{join condition} \rangle} S = \sigma_{\langle \text{join condition} \rangle} (R \text{ X } S)$$

Examples of queries in relational algebra - 1

- Retrieve the name (fname,lname) and address of all employees who work for the ‘Research’ department.

Examples of queries in relational algebra - 2

- Retrieve the names (fname, lname) of employees who have no dependents.

These slides are is based on the textbook:

R. Elmaseri and S. Navathe, *Fundamentals of Database Systems*, 7th Edition, Addison-Wesley.