THE RELATIONAL DATA MODEL

Chapter 6 (Part 2):
Relational Algebra Example Queries

Outline of Chapter 6

1. Relational Algebra

2. Relational Algebra Example Queries

3. Relational Calculus

2.1. Remarks

- In general a query can be stated in different ways using the various operators.

- The presence of integrity constraints affects the way a query is expressed in relational algebra.

2.2. Example queries (1)

- *Retrieve the name and address of all employees who work for the 'Research' department.*

$\text{RES\_DEPT} \leftarrow \sigma_{\text{DNAME='Research'}}(\text{DEPARTMENT})$

$\text{RES\_EMPS} \leftarrow \text{RES\_DEPT} \rhd\!\lhd_{\text{DNUMBER=DNO}} \text{EMPLOYEE}$

$\text{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME, ADDRESS}}(\text{RES\_EMPS})$

2.2. Example queries (2)

- *Retrieve the name and address of all employees who work for the 'Research' department.*

  Alternative rewriting of the query (a natural join operator is used combined with an attribute renaming operator)

$\text{RES\_DEPT} \leftarrow \sigma_{\text{DNAME='Research'}}(\text{DEPARTMENT})$

$\text{RES\_EMPS} \leftarrow \text{EMPLOYEE} * \rho_{\text{DNUMBER} \rightarrow \text{DNO}}(\text{RES\_DEPT})$

$\text{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME, ADDRESS}}(\text{RES\_EMPS})$

2.2. Example queries (3)

- *Retrieve the name and address of all employees who work for the 'Research' department.*

  Alternative rewriting of the query (the selection and join operations are reversed)

$\text{EMP\_DEPT} \leftarrow \text{EMPLOYEE} \bowtie_{\text{DNUMBER=DNO}} \text{DEPARTMENT}$

$\text{RES\_EMPS\_DEPTS} \leftarrow \sigma_{\text{DNAME='Research'}} (\text{EMP\_DEPT})$

$\text{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME, ADDRESS}}(\text{RES\_EMPS\_DEPTS})$

2.2. Example queries (4)

- *For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address and birthdate.*

$\text{STAF\_PROJS} \leftarrow \sigma_{\text{PLOCATION='Stafford'}}(\text{PROJECT})$

$\text{CONT\_DEPT} \leftarrow \text{STAF\_PROJS} \rhd\lhd_{\text{DNUM=DNUMBER}} \text{DEPARTMENT}$

$\text{PR\_DEPT\_MGR} \leftarrow \text{CONT\_DEPT} \rhd\lhd_{\text{MGRSSN=SSN}} \text{EMPLOYEE}$

$\text{RESULT} \leftarrow \Pi_{\text{PNUMBER, DNUM, LNAME, ADDRESS, BDATE}}(\text{PR\_DEPT\_MGR})$

2.2. Example queries (5)

- *Find the names of employees who work on all the projects.*

$\text{ALL\_PROJS} \leftarrow \Pi_{\text{PNUMBER}} (\text{PROJECT})$

$\text{EMP\_PROJ} \leftarrow \rho_{\text{PNO} \rightarrow \text{PNUMBER}} (\Pi_{\text{ESSN, PNO}} (\text{WORKS\_ON}))$

$\text{RES\_ESSN} \leftarrow \text{EMP\_PROJ} \div \text{ALL\_PROJS}$

$\text{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME}} (\text{RES\_ESSN} \triangleright\triangleleft_{\text{ESSN=SSN}} \text{EMPLOYEE})$

2.2. Example queries (6)

- *Make a list of project numbers that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project.*

SMITHS $\leftarrow \rho_{\text{SSN} \rightarrow \text{ESSN}}(\Pi_{\text{SSN}}(\sigma_{\text{LNAME='Smith'}}(\text{EMPLOYEE})))$

SMITH_WORKS $\leftarrow \Pi_{\text{PNO}}(\text{WORKS\_ON} * \text{SMITHS})$

MGRS $\leftarrow \Pi_{\text{LNAME, DUMBER}}(\text{EMPLOYEE} \bowtie_{\text{SSN=MGRSSN}} \text{DEPARTMENT})$

SMITH_MGR $\leftarrow \rho_{\text{DNUMBER} \rightarrow \text{DNUM}}(\Pi_{\text{DNUMBER}}(\sigma_{\text{LNAME='Smith'}}(\text{MGRS})))$

SMITH_PROJ $\leftarrow \rho_{\text{PNUMBER} \rightarrow \text{PNO}}(\Pi_{\text{PNUMBER}}(\text{SMITH\_MGR} * \text{PROJECT})))$

RESULT $\leftarrow$ SMITH_WORKS $\cup$ SMITH_PROJ

2.2. Example queries (7)

- *List the names of all employees with two or more dependents.*

$T1 \leftarrow {}_{\text{ESSN}}\,\mathfrak{F}_{\,\text{COUNT(DEPENDENT\_NAME)}}\,(\text{DEPENDENT})$

$T2 \leftarrow \rho_{\text{ESSN} \rightarrow \text{SSN, COUNT(DEPENDENT\_NAME)} \rightarrow \text{NO\_OF\_DEPS}}\,(T1)$

$T3 \leftarrow \sigma_{\text{NO\_OF\_DEPS} \geq 2}\,(T2)$

$\text{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME}}\,(\ T3 * \text{EMPLOYEE}\ )$

2.2. Example queries (8)

- *List the names of all employees with two or more dependents.*

  (Alternative solution without aggregate functions.)

$\text{DEPEN1} \leftarrow \Pi_{\text{ESSN, DEPENDENT\_NAME}}(\text{ DEPENDENT})$

$\text{DEPEN2} \leftarrow \rho_{\text{DEPENDENT\_NAME} \rightarrow \text{DEPENDENT\_NAME1}} (\text{DEPEN1})$

$\text{T} \leftarrow \sigma_{\text{DEPENDENT\_NAME} \neq \text{DEPENDENT\_NAME1}}(\text{DEPEN1 * DEPEN2})$

$\text{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME}} (\text{ T } \triangleright\triangleleft_{\text{ESSN=SSN}} \text{ EMPLOYEE })$

2.2. Example queries (9)

- *Retrieve the names of employees who have no dependents.*

$$\text{ALL\_EMPS} \leftarrow \Pi_{\text{SSN}} (\text{ EMPLOYEE })$$

$$\text{EMPS\_WITH\_DEPS} \leftarrow \rho_{\text{ESSN} \rightarrow \text{SSN}} (\Pi_{\text{ESSN}} (\text{ DEPENDENT }))$$

$$\text{EMPS\_WITHOUT\_DEPS} \leftarrow \text{ALL\_EMPS - EMPS\_WITH\_DEPS}$$

$$\text{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME}} (\text{EMPS\_WITHOUT\_DEPS * EMPLOYEE})$$

2.2. Example queries (10)

- *List the names of managers who have at least one dependent.*

$$\text{MGRS} \leftarrow \rho_{\text{MGRSSN} \rightarrow \text{SSN}} (\Pi_{\text{MGRSSN}} (\text{DEPARTMENT}))$$

$$\text{EMPS\_WITH\_DEPS} \leftarrow \rho_{\text{ESSN} \rightarrow \text{SSN}} (\Pi_{\text{ESSN}} (\text{DEPENDENT}))$$

$$\text{MGRS\_WITH\_DEPS} \leftarrow \text{MGRS} \cap \text{EMPS\_WITH\_DEPS}$$

$$\text{RESULT} \leftarrow \Pi_{\text{FNAME, LNAME}} (\text{MGRS\_WITH\_DEPS} * \text{EMPLOYEE})$$

2.2. Example queries (11)

- *List the names of managers who have at least one dependent (alternative solution).*

$$\text{MGRS\_DEPS} \leftarrow \text{DEPENDENT} \bowtie_{\text{ESSN}=\text{MGRSSN}} \text{DEPARTMENT}$$

$$\text{ESSN\_MGRS\_DEPS} \leftarrow \Pi_{\text{ESSN}} (\text{MGRS\_DEPS})$$

$$\text{RESULT} \leftarrow$$

$$\Pi_{\text{FNAME, LNAME}} (\text{ESSN\_MGRS\_DEPS} \bowtie_{\text{ESSN}=\text{SSN}} \text{EMPLOYEE})$$

2.3. Query optimization

- All DBMSs have a *query optimizer*.
- A relational algebra query may have *many equivalent expressions* and some of them may be *much more easier to evaluate (compute)*.
- One of the tasks of a query optimizer is to generate *alternative equivalent expressions* of a query in order to find *one that is more efficient to evaluate*.