Relational Algebra

Operators

- **Set operators**
  - Union
  - Intersection
  - Difference
  - Cartesian product

- **Relational operators**
  - Selection
  - Projection
  - Join
  - Division

Examples (1): Select (\(\sigma\)) and Project (\(\pi\))

EMP (EMPNO, NAME, DNO, JOB, MGR, SAL, COMMISSION)
DEPT (DNO, DNAME, LOC)

Q1. Find the subset of employees who are in dept. 50.
\(\sigma_{DNO=50}(EMP)\)

Q2. Find the subset of employees who are in dept. 25, 47 or 53.
\(\sigma_{DNO=25 \text{ or } DNO=47 \text{ or } DNO=53}(EMP)\)

Q3. List names and jobs of all employees.
\(\pi_{NAME, JOB}(EMP)\)

Examples (2):

More Complex Select (\(\sigma\)) and Project (\(\pi\))

EMP (EMPNO, NAME, DNO, JOB, MGR, SAL, COMMISSION)
DEPT (DNO, DNAME, LOC)

Q4. List names and jobs of employees in departments 25, 47 or 53.
\(\pi_{NAME, JOB}(\sigma_{DNO=25 \text{ or } DNO=47 \text{ or } DNO=53}(EMP))\)

D_EMPS ← \(\sigma_{DNO=25 \text{ or } DNO=47 \text{ or } DNO=53}(EMP)\)
RESULT ← \(\pi_{NAME, JOB}(D_{EMP})\)

Examples (3): Union (\(\cup\)) and Difference (\(-\))

EMP (EMPNO, NAME, DNO, JOB, MGR, SAL, COMMISSION)
DEPT (DNO, DNAME, LOC)

Q5. List the numbers of those departments which have an employee named ‘Smith’ or are located in ‘Columbus’.
\(\pi_{DNO}(\sigma_{NAME='Smith'}(EMP)) \cup \pi_{DNO}(\sigma_{LOC='Columbus'}(DEPT))\)

Q6. List the DNO for departments that have no employees.
\(\pi_{DNO}(DEPT) - \pi_{DNO}(EMP)\)

Examples (4): Join: Natural (\(\ast\)) and Theta (\(\bowtie\))

EMP (EMPNO, NAME, DNO, JOB, MGR, SAL, COMMISSION)
DEPT (DNO, DNAME, LOC)

Q7. List the names of all employees and the locations of their departments.
\(\pi_{NAME, LOC}(EMP * DEPT)\) * Natural join
\(\pi_{NAME, LOC}(EMP \bowtie_{EMP.DNO=DEPT.DNO}(DEPT))\) \(\bowtie\) Theta-join (\(\bowtie\) Join)
\(\theta_{\bowtie}(\bowtie)\)
Examples (5): Select, project and join

EMP(EMPNO, NAME, DNO, JOB, MGR, SAL, COMMISSION)
DEPT(DNO, DNAME, LOC)

Q8. Find the names of employees who work in Ann Arbor.

\[ \pi \text{NAME} \quad (\text{EMP} \ast (\sigma_{\text{LOC} = \text{Ann Arbor}}(\text{DEPT}))) \quad \text{Natural join} \ast \]

\[ \pi \text{NAME} \quad (\text{EMP} \ast|\text{EMP.DNO = DEPT.DNO} \quad \text{equi-join} \ast) \quad \text{when } \theta = \text{"="} \]

Examples (6): Not Equal (≠) in Select

EMP(EMPNO, NAME, DNO, JOB, MGR, SAL, COMMISSION)
DEPT(DNO, DNAME, LOC)

Q9. Find the names of employees who do not work in Ann Arbor.

\[ \pi \text{NAME} \quad (\text{EMP} \ast (\sigma_{\text{LOC} \neq \text{Ann Arbor}}(\text{DEPT}))) \quad \text{Natural join} \ast \]

\[ \pi \text{NAME} \quad (\text{EMP} \ast|\text{EMP.DNO = DEPT.DNO} \quad \theta \text{ Join} \ast) \quad (\theta) \]

Examples (7): A “classic”, equi-join as product (×) and select (σ)

EMP(EMPNO, NAME, DNO, JOB, MGR, SAL, COMMISSION)
EMP(EMPNO, NAME, DNO, JOB, MGR, SAL, COMMISSION)
DEPT(DNO, DNAME, LOC)

Q10. For each employee whose salary exceeds his or her manager’s salary, list the employee’s name and the manager’s name.

\[ X \leftarrow \text{EMP} \quad Y \leftarrow \text{EMP} \quad \pi \text{X.NAME, Y.NAME} \quad (\sigma_{\text{X.SAL} > \text{Y.SAL} \land \text{Y.EMPNO} = \text{X.MGR}}(\text{X} \times \text{Y})) \]

Examples (8): A new kind of query

EMP(EMPNO, NAME, DNO, JOB, MGR, SAL, COMMISSION)
DEPT(DNO, DNAME, LOC)

Q11. Find the numbers of those departments that have employees who can do some job that is done by an employee in department D3.

Answer: D1 and D2

Q12. Find the numbers of those departments that have employees who can do all the jobs that are done by an employee in department D3.

Examples (8): continued

Q11. Find the numbers of those departments that have employees who can do some job that is done by an employee in department D3.

Answer: D1 and D2

\[ \text{EMP}(\text{EMPNO}, \ldots, \text{DNO, JOB,} \ldots) \]

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>DNO</th>
<th>JOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>D3</td>
<td>electrician</td>
</tr>
<tr>
<td>200</td>
<td>D3</td>
<td>plumber</td>
</tr>
<tr>
<td>300</td>
<td>D3</td>
<td>electrician</td>
</tr>
<tr>
<td>400</td>
<td>D1</td>
<td>electrician</td>
</tr>
<tr>
<td>500</td>
<td>D1</td>
<td>plumber</td>
</tr>
<tr>
<td>600</td>
<td>D1</td>
<td>carpenter</td>
</tr>
<tr>
<td>700</td>
<td>D2</td>
<td>electrician</td>
</tr>
<tr>
<td>800</td>
<td>D2</td>
<td>carpenter</td>
</tr>
<tr>
<td>900</td>
<td>D2</td>
<td>electrician</td>
</tr>
</tbody>
</table>
Examples (8): continued

Q12. Find the numbers of those departments that have employees who can do all the jobs that are done by an employee in department D3. Answer: D1, but not D2

\[
\begin{array}{|c|c|c|}
\hline
\text{EMP( EMPNO, ..., DNO, JOB, ...)} & \text{Electrician} & \text{Plumber} \\
\hline
100 & D3 & \text{Electrician} \\
200 & D3 & \text{Plumber} \\
300 & D3 & \text{Electrician} \\
400 & D1 & \text{Electrician} \\
500 & D1 & \text{Plumber} \\
600 & D1 & \text{Carpenter} \\
700 & D2 & \text{Electrician} \\
800 & D2 & \text{Carpenter} \\
900 & D2 & \text{Electrician} \\
\hline
\end{array}
\]

Examples (9): divide (÷)

\[
\begin{array}{c}
\text{SP( S#, P#)} \\
S1 P1 \\
S1 P2 \\
S1 P3 \\
S1 P4 \\
S2 P1 \\
S2 P2 \\
S2 P3 \\
S3 P2 \\
S4 P1 \\
S4 P2 \\
S4 P3 \\
\end{array}
\]

\[
\begin{array}{c}
\text{SP } ÷ \text{ P ( S#)} \\
S1 P1 \\
S1 P2 \\
S1 P3 \\
S1 P4 \\
S2 P1 \\
S2 P2 \\
S2 P3 \\
S3 P2 \\
S4 P1 \\
S4 P2 \\
S4 P3 \\
\end{array}
\]

Examples (8): Divide (÷), for all (∀)

Q12. Find the numbers of those departments that have employees who can do all the jobs that are done by an employee in department D3. Answer: D1, but not D2

\[
\begin{array}{|c|c|c|}
\hline
\text{EMP( EMPNO, DNO, JOB, ...)} & \text{Electrician} & \text{Plumber} \\
\hline
100 & D3 & \text{Electrician} \\
200 & D3 & \text{Plumber} \\
300 & D3 & \text{Electrician} \\
400 & D1 & \text{Electrician} \\
500 & D1 & \text{Plumber} \\
600 & D1 & \text{Carpenter} \\
700 & D2 & \text{Electrician} \\
800 & D2 & \text{Carpenter} \\
900 & D2 & \text{Electrician} \\
\hline
\end{array}
\]

\[
\begin{array}{c}
\text{D3JOBS } ← \pi_{\text{DNO = D3}}(\text{EMP}) \\
\text{DEPT_JOBS } ← \pi_{\text{DNO, JOB}}(\text{EMP}) \\
\text{GOOD_DEPTS } ← \text{DEPT_JOBS } ÷ \text{D3JOBS} \\
\end{array}
\]

\[
\begin{array}{c}
\pi_{\text{DNO, JOB}}(\text{EMP}) ÷ \\
\pi_{\text{DNO = D3}}(\text{EMP}) \\
\end{array}
\]
Examples (10): Functions and groups

Q13. Find the average salary of all employees.

\[
\text{EMP (EMPNO, DNO, SAL, ...)}
\]

\[
\begin{array}{ccc}
100 & D3 & 66,000 \\
200 & D3 & 55,000 \\
300 & D3 & 66,000 \\
400 & D1 & 65,000 \\
500 & D1 & 55,000 \\
600 & D1 & 60,000 \\
700 & D2 & 66,000 \\
800 & D2 & 60,000 \\
900 & D2 & 66,000 \\
\end{array}
\]

\[
\text{average sal} \ (\text{EMP}) \rightarrow 62,000
\]

Examples (10): Functions and groups continued

Q14. List the departments (DNO) and the average salary of each.

\[
\text{EMP (EMPNO, DNO, SAL, ...)}
\]

\[
\begin{array}{ccc}
100 & D3 & 66,000 \\
200 & D3 & 55,000 \\
300 & D3 & 66,000 \\
400 & D1 & 66,000 \\
500 & D1 & 55,000 \\
600 & D1 & 60,000 \\
700 & D2 & 66,000 \\
800 & D2 & 60,000 \\
900 & D2 & 66,000 \\
\end{array}
\]

\[
\text{DNO} \rightarrow \text{average sal} \ (\text{EMP}) \rightarrow 62,000, 60,333, 64,000
\]

Examples (11): Outer joins

Q15a. For each part list all suppliers in the same city.

\[
P (P#, PNAME, CITY) \quad S (S#, SNAME, CITY)
\]

\[
P (P1, Nut, London) \quad S (S1, Smith, London)
\]

Q15b. For each part list all suppliers in the same city.

\[
P (P#, PNAME, S#, SNAME, CITY) \quad S (S#, SNAME, CITY)
\]

\[
P (P1, Nut, S1, Smith, London) \quad S (S1, Smith, London)
\]

Q15c. For each part list all suppliers in the same city.

\[
P (P#, PNAME, S#, SNAME, CITY) \quad S (S#, SNAME, CITY)
\]

\[
P (P1, Nut, S1, Smith, London) \quad S (S1, Smith, London)
\]

Q15d. For each part list all suppliers in the same city.

\[
P (P#, PNAME, S#, SNAME, CITY) \quad S (S#, SNAME, CITY)
\]

\[
P (P3, Screw, Rome) \quad S (S5, Adams, Athens)
\]

Examples (12): Recursive closure

Q16. List all the superiors of EMPNO 500.

\[
\begin{array}{ccc}
100 & 200 & 800 \\
700 & 800 & 950 \\
850 & 980 & 980 \\
\end{array}
\]

Q17. List all those supervised by EMPNO 900.

\[
\begin{array}{ccc}
100 & 200 & 300 \\
700 & 800 & 950 \\
980 & 980 & 980 \\
\end{array}
\]

Examples (12): Recursive closure continued

Q16. List all the superiors of EMPNO 500.

\[
\begin{array}{ccc}
100 & 700 \\
200 & 800 \\
300 & 950 \\
400 & 700 \\
500 & 800 \\
600 & 950 \\
700 & 800 \\
800 & 950 \\
900 & 980 \\
950 & 980 \\
980 & 980 \\
\end{array}
\]

Q17. List all those supervised by EMPNO 900.

\[
\begin{array}{ccc}
100 & 200 & 300 \\
700 & 800 & 950 \\
980 & 980 & 980 \\
\end{array}
\]

Can’t express these queries? WHY?
Q 16. List all the superiors of EMPNO 500. SUPERIORS (EMPNO, MGR)

<table>
<thead>
<tr>
<th>EMPNO</th>
<th>MGR</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>700</td>
</tr>
<tr>
<td>200</td>
<td>700</td>
</tr>
<tr>
<td>300</td>
<td>800</td>
</tr>
<tr>
<td>400</td>
<td>900</td>
</tr>
<tr>
<td>500</td>
<td>900</td>
</tr>
<tr>
<td>600</td>
<td>950</td>
</tr>
<tr>
<td>700</td>
<td>900</td>
</tr>
<tr>
<td>800</td>
<td>900</td>
</tr>
<tr>
<td>850</td>
<td>950</td>
</tr>
<tr>
<td>900</td>
<td>980</td>
</tr>
<tr>
<td>950</td>
<td>980</td>
</tr>
<tr>
<td>980</td>
<td>980</td>
</tr>
</tbody>
</table>

Can’t express these queries? WHY?

---

Important Concepts

- Relational completeness
  - Language is sufficiently powerful.
  - Is a new language this powerful?
    - Must be able to translate any relational algebra statement to the new language.

Important Concepts, Continued

- Fully relational database
  - Relational structure
    - Domain
    - Integrity rules
      - Key
      - Entity
      - Referential
  - Language as powerful as relational algebra