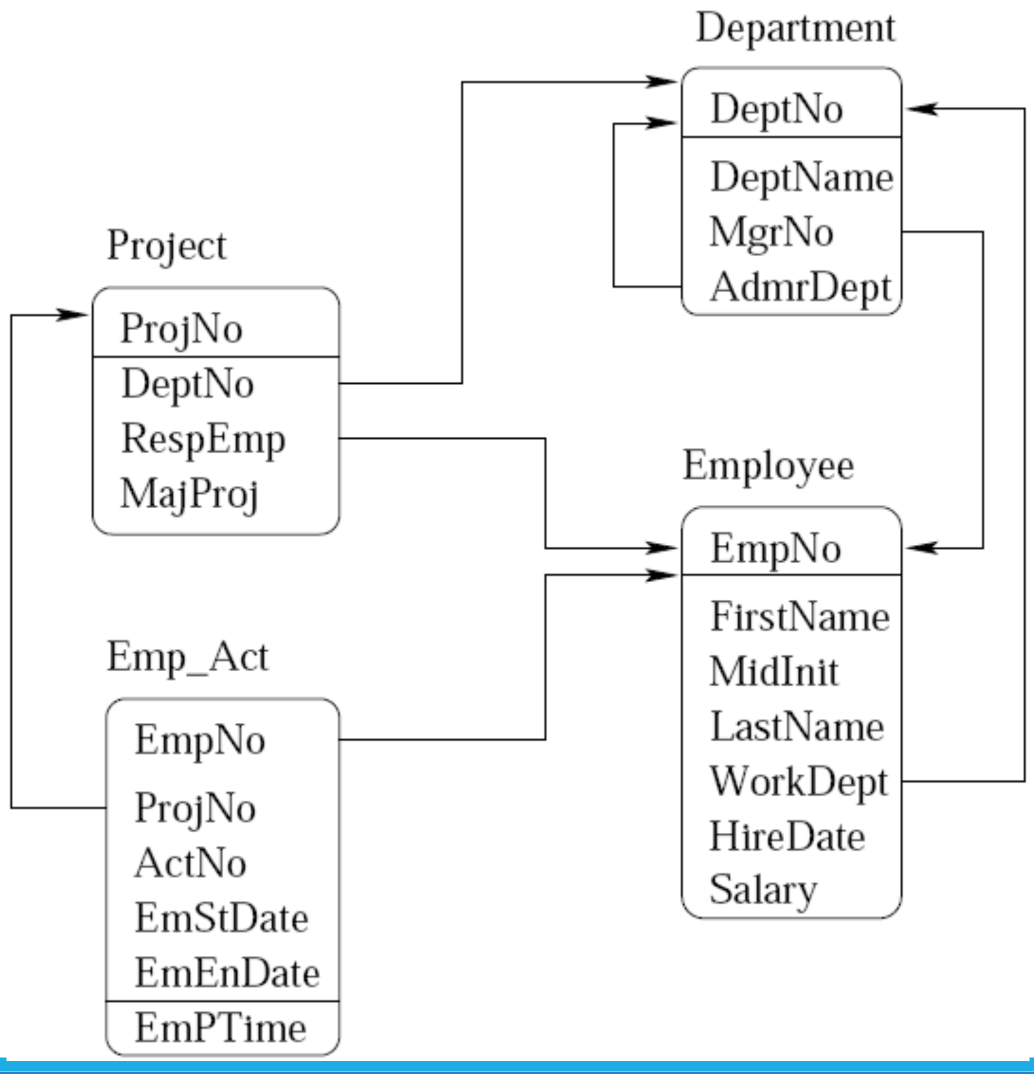
Relational Algebra

EGCI321: LECTURE06 (WEEK 3)

Relational Database



Relational Algebra

The relational algebra consists of a set of *operator*

- Each operator takes one or more input relations
- Relational operators can be composed of form expressions that define new relations in terms of existing relations.

Notation:

R is a relation name;

E is a relational algebra expression

Primary Relational Operators

Relation Name: R

Selection: $\sigma_{condition}(E)$

- Result schema is the same as E's
- Result instance includes the subset of the tuples of E that each satisfies the condition

Projection: $\pi_{attributes}(E)$

- Result schema includes only the specified attributes
- Result instance could have as many tuples as E, except that duplicates are eliminated

Primary Relational Operators (cont.)

Rename: $\rho(R(\overline{F}),E)$

- F *is a list of terms of the form oldname* $\rightarrow$ *newname*
- Returns the result of E with columns renamed according to $\overline{F}$
- Remembers the result as R for future expressions

Product: $E_1 \times E_2$

- Result schema has all of the attributes of E_1 and all of the attributes of E_2
- Result instance includes one tuple for every pair of tuples (one from each expression result) in E_1 and E_2
- Sometimes called *cross-product* or *Cartesian product*
- Renaming in needed when E_1 and E_2 have common attributes

Cross Product Example

R

AAA	BBB
a_1	b_1
a_2	b_2
a_3	b_3

S

CCC	DDD
c_1	d_1
c_2	d_2

$R \times S$

AAA	BBB	CCC	DDD
a_1	b_1	c_1	d_1
a_2	b_2	c_1	d_1
a_3	b_3	c_1	d_1
a_1	b_1	c_2	d_2
a_2	b_2	c_2	d_2
a_3	b_3	c_2	d_2

Select, Project, Product Examples

Note: Use *Emp* to mean the Employee relation, *Proj* the project relation

Find the last names and hire date of employees who make more than $100000

$$\pi_{LastName,\ HireDate}(\sigma_{Salary>100000}(Emp))$$

For each project for which department E_{21} is responsible, find the name of the employee in charge of that project.

$$\pi_{ProjNo, LastName}(\sigma_{DeptNo=E21}(\sigma_{RespEmp=EmpNo}(Emp \times Proj)))$$

Joins

Conditional Join: $E_1 \bowtie_{condition} E_2$

- Equivalent to $\sigma_{condition}(E_1 \times E_2)$
- Special case *equijoin*

$$Proj \bowtie_{(\mathrm{RespEmp=EmpNo})} E$$

Natural join $(E_1 \bowtie E_2)$

- The result of $(E_1 \bowtie E_2)$ can be formed by the following steps
  1. Form the cross-product $E1$ and $E2$
  2. Eliminate from the cross product any tuples that do not have matching value for *all pairs* of attributes common to schemas $E1$ and $E2$
  3. Project out duplicate attributes
- If no attributes in common, this is just a product

Example:

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12.0	London
P2	Bolt	Green	17.0	Paris
P3	Screw	Blue	17.0	Rome
P4	Screw	Red	14.0	London
P5	Cam	Blue	12.0	Paris
P6	Cog	Red	19.0	London

Natural-join

S#	SNAME	STATUS	CITY	P#	PNAME	COLOR	WEIGHT
S1	Smith	20	London	P1	Nut	Red	12.0
S1	Smith	20	London	P4	Screw	Red	14.0
S1	Smith	20	London	P6	Cog	Red	19.0
S2	Jones	10	Paris	P2	Bolt	Green	17.0
S2	Jones	10	Paris	P5	Cam	Blue	12.0
S3	Blake	30	Paris	P2	Bolt	Green	17.0
S3	Blake	30	Paris	P5	Cam	Blue	12.0
S4	Clark	20	London	P1	Nut	Red	12.0
S4	Clark	20	London	P4	Screw	Red	14.0
S4	Clark	20	London	P6	Cog	Red	19.0

Example: Natural Join

Consider the natural join of the Project and Department tables, which have attribute DeptNo in common

- The schema of the result will include attributes ProjName, DeptNo, ResEmp, MajProj, DeptName, MgrNo, and AdmrDept
- The resulting relation will include one tuple for each tuple in the Project relation

Set-Based Relational Operators

Union $(R \cup S)$:

- Schemas of R and S must be “union compatible”
- Result includes all tuples that appear either in *R* or in *S* or in both

Difference $(R - S)$

- Schemas of *R* and *S* must be “union compatible”
- Result includes all tuples that appear in *R* and that do not appear in *S*

Intersection $(R \cap S)$: same type

- Schemas of *R* and *S* must be “union compatible”
- Result includes all tuples that appear in both *R* and *S*

Union Compatible:

- Same number of fields
- ‘Corresponding’ fields have the same type

Relational Division

X

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_2
a_2	b_1	c_1
a_2	b_1	c_2
a_2	b_2	c_2
a_2	b_3	c_3
a_3	b_1	c_1

S

B	C
b_1	c_1
b_1	c_2
b_2	c_2

X/S

A
a_1
a_2

Division is the Inverse of Product

R

A
a_1
a_2

S

B	C
b_1	c_1
b_1	c_2
b_2	c_2

$R \times S$

A	B	C
a_1	b_1	c_1
a_1	b_1	c_2
a_1	b_2	c_2
a_2	b_1	c_1
a_2	b_1	c_2
a_2	b_2	c_2

$(R \times S)/S$

A
a_1
a_2

Summary of Relational Operators

$$
\begin{array}{lcl}
E & ::= & R \\
 & | & \sigma_{condition}(E) \\
 & | & \pi_{attributes}(E) \\
 & | & \rho(R(\overline{F}), E) \\
 & | & E_1 \times E_2 \\
 & | & E_1 \bowtie_{condition} E_2 \\
 & | & E_1 \bowtie E_2 \\
 & | & E_1 \cup E_2 \\
 & | & E_1 \cap E_2 \\
 & | & E_1 - E_2 \\
 & | & E_1 \,/\, E_2
\end{array}
$$

Algebraic Equivalences

This

$$\pi_{ProjNo,LastName}(\sigma_{DeptNo=E21}(\sigma_{RespEmp=EmpNo}(E \times P)))$$

Equivalent to this:

$$\pi_{ProjNo,LastName}(\sigma_{DeptNo=E21}(E \bowtie_{RespEmp=EmpNo} P))$$

Equivalent to this:

$$\pi_{ProjNo,LastName}(E \bowtie_{RespEmp=EmpNo} \sigma_{DeptNo=E21}(P))$$

Equivalent to this:

$$\pi_{ProjNo,LastName}(\quad (\quad \pi_{LastName,EmpNo}(E)) \bowtie_{RespEmp=EmpNo} (\quad \pi_{ProjNo,RespEmp}(\sigma_{DeptNo=E21}(P))))$$

Example I: Instances (Sailors and Reserve)

R1

sid	bid	day
22	101	10/10/96
58	103	11/12/96

S1

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0

S2

sid	sname	rating	age
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

Example I: Projection

- Deletes attributes that are not in projection list.
- Schema of result contains exactly the fields in the projection list, with the same names that they had in the (only) input relation.
- Projection operator has to eliminate duplicates!
  - Note: real systems typically don't do duplicate elimination unless the user explicitly asks for it.

sname	rating
yuppy	9
lubber	8
guppy	5
rusty	10

$$\pi_{sname,rating}(S2)$$

age
35.0
55.5

$$\pi_{age}(S2)$$

Example I: Selection

- Selects rows that satisfy selection condition.
- No duplicates in result!
- Schema of result identical to schema of (only) input relation.
- Result relation can be the input for another relational algebra operation! (Operator composition.)

sid	sname	rating	age
28	yuppy	9	35.0
58	rusty	10	35.0

$$\sigma_{rating>8}(S2)$$

sname	rating
yuppy	9
rusty	10

$$\pi_{sname,rating}(\sigma_{rating>8}(S2))$$

Example I: Union, Intersection, and Set-Difference

All of these operations take two input relations, which must be union-compatible:

- Same number of fields.
- `Corresponding' fields have the same type.

sid	sname	rating	age
22	dustin	7	45.0
31	lubber	8	55.5
58	rusty	10	35.0
44	guppy	5	35.0
28	yuppy	9	35.0

$S1 \cup S2$

sid	sname	rating	age
22	dustin	7	45.0

$S1 - S2$

sid	sname	rating	age
31	lubber	8	55.5
58	rusty	10	35.0

$S1 \cap S2$

Example I: Cross Product

- Each row of S1 is paired with each row of R1.
- Result schema has one field per field of S1 and R1, with field names `inherited' if possible.
  - Conflict: Both S1 and R1 have a field called sid.

(sid)	sname	rating	age	(sid)	bid	day
22	dustin	7	45.0	22	101	10/10/96
22	dustin	7	45.0	58	103	11/12/96
31	lubber	8	55.5	22	101	10/10/96
31	lubber	8	55.5	58	103	11/12/96
58	rusty	10	35.0	22	101	10/10/96
58	rusty	10	35.0	58	103	11/12/96

Renaming operator:

$$\rho\,(C(1 \rightarrow sid1, 5 \rightarrow sid2)\,,\; S1 \times R1)$$

Practice: Write the Relational Algebra

List of Tables

- Flights(flno, from, to, distance, departs)
- Aircraft(aid, aname, range)
- Certified(eid, aid)
- Employees(eid, ename, salary)

Query Statement

1. Find eid’s of pilots who are certified on “Boeing777”.
   (Boeing777 is aid)
2. Find eid of employee(s) with the salary > 100,000.\
3. Find employee’s name and aircraft’s name who are certified on “Airbus 600”
   (Airbus 600 is aid)

Reference

1. Ramakrishnan R, Gehrke J., Database management systems, 3rd ed., New York (NY): McGraw-Hill, 2003.