Relational Algebra Examples – Round 2

Maximum Search

Students(Name, Date of Birth, Town, Grade, Performance)

Name	Date of Birth	Town	Grade	Performance
Schmidt János	1985	Arad	5	94%
Hoffmann Géza	1986	Tata	4	84%
Péter Lajos	1985	Szombathely	5	90%

Question: Give the relational algebra expression which selects the student with maximum performance.

Maximum Search

Name	Date of Birth	Town	Grade	Performance
Schmidt János	1985	Arad	5	94%
Hoffmann Géza	1986	Tata	4	84%
Péter Lajos	1985	Szombathely	5	90%

- We only need the name and performance of the students.

  $\Pi_{\text{name, performance}}(\text{Students})$

- We make a copy of our table

  $\rho_{\text{New}}(\Pi_{\text{name, performance}}(\text{Students}))$

Name	Performance
Schmidt János	94%
Hoffmann Géza	84%
Péter Lajos	90%

→

New_Name	New_Performance
Schmidt János	94%
Hoffmann Géza	84%
Péter Lajos	90%

Maximum Search

Name	Date of Birth	Town	Grade	Performance
Schmidt János	1985	Arad	5	94%
Hoffmann Géza	1986	Tata	4	84%
Péter Lajos	1985	Szombathely	5	90%

- Let us make the cartesian product of the two table

$\Pi_{name,\ performance}(Students) \times \rho_{New}(\Pi_{name,\ performance}(Students))$

Name	Performance	New_Name	New_Performance
Schmidt J.	94 %	Schmidt J.	94 %
Schmidt J.	94 %	Hoffmann G.	84 %
Schmidt J.	94 %	Péter L.	90 %
Hoffmann G.	84 %	Schmidt J.	94 %
Hoffmann G.	84 %	Hoffmann G.	84 %
Hoffmann G.	84 %	Péter L.	90 %
Péter L.	90 %	Schmidt J.	94 %
Péter L.	90 %	Hoffmann G.	84 %
Péter L.	90 %	Péter L.	90 %

Maximum Search

Name	Date of Birth	Town	Grade	Performance
Schmidt János	1985	Arad	5	94%
Hoffmann Géza	1986	Tata	4	84%
Péter Lajos	1985	Szombathely	5	90%

- Selct those rows, where the performance is greater than new performance

$$\sigma_{\text{performance} > \text{new\_performance}}(\Pi_{\text{name, performance}}(\text{Students}) \times \rho_{\text{New}}(\Pi_{\text{name, performance}}(\text{Students})))$$

Name	Performance	New_Name	New_Performance
Schmidt J.	94 %	Schmidt J.	94 %
Schmidt J.	94 %	Hoffmann G.	84 %
Schmidt J.	94 %	Péter L.	90 %
Hoffmann G.	84 %	Schmidt J.	94 %
Hoffmann G.	84 %	Hoffmann G.	84 %
Hoffmann G.	84 %	Péter L.	90 %
Péter L.	90 %	Schmidt J.	94 %
Péter L.	90 %	Hoffmann G.	84 %
Péter L.	90 %	Péter L.	90 %

Maximum Search

Name	Date of Birth	Town	Grade	Performance
Schmidt János	1985	Arad	5	94%
Hoffmann Géza	1986	Tata	4	84%
Péter Lajos	1985	Szombathely	5	90%

- We get the result subtracting the rows with the columns from the new table from the original table:

$$\Pi_{\text{name, performance}}(\text{Students}) - \Pi_{\text{new\_name, new\_performance}}(\sigma_{\text{performance} > \text{new\_performance}}(\Pi_{\text{name, performance}}(\text{Students}) \times \rho_{\text{New}}(\Pi_{\text{name, performance}}(\text{Students}))))$$

Name	Performance
Schmidt János	94%
Hoffmann Géza	84%
Péter Lajos	90%

—

New_Name	New_Performance
Hoffmann Géza	84%
Péter Lajos	90%

Result: Schmidt János – 94%!

Maximum Search

- *Steps (summary):*
  - Rename
  - Cross product
  - Selection with a given condition
  - Projection of the names
  - Difference of the original set and the set we get after the above operations

---

$\Pi_{name,\ performance}(Students)$-

$\Pi_{new\_name,\ new\_performance}(\sigma_{performance > new\_performance}($
$\Pi_{name,\ performance}(Students) \times \rho_{new}(\Pi_{name,\ performance}(Students)))$

If you change the part highlighted with green text, it will give you:

1. $\Pi_{new\_name,\ new\_performance}(\sigma_{performance > new\_performance}$ → **max**imum
2. $\Pi_{name,\ performance}(\sigma_{performance>new\_performance}$ → **min**imum
3. $\Pi_{new\_name,\ new\_performance}(\sigma_{performance < new\_performance}$ → **min**imum
4. $\Pi_{name,\ performance}(\sigma_{performance<new\_performance}$ → **max**imum

Assignment Operator

$\pi_{SSN,\ Accidents}$ (Drivers)- $\pi_{Driver.SSN,\ Drivers.Accidents}$
($\sigma_{Driver.\ Accidents\ >NEW.\ Accidents}$ ($\pi_{Drivers.SSN,\ Drivers.\ Accidents}$ (Drivers)
$\times\rho_{\ R\ \leftarrow NEW}$($\pi_{Drivers.SSN,\ Drivers.\ Accidents}$ (Drivers))))

- $R \leftarrow \pi_{Drivers.SSN,\ Drivers.\ Accidents}$ *(Drivers)*
- $NEW \leftarrow \rho_{R2}$ *(R)*
- $TEMP1 \leftarrow \sigma_{R.Accidents\ >NEW.\ Accidents}$ (R $\times$ *NEW*).
- $R - \pi_{R.SSN,\ R.\ Accidents}$ *(TEMP1)*

Division - Example

R:

A	B
a1	b1
a1	b2
a1	b3
a1	b4
a2	b1
a2	b2
a3	b2
a4	b2
a4	b4

S:

B
b2
b4

R÷S:

A
a1
a4

Division Example 2 – self work

Question: Who completed all Database Tasks?

Completed:

Student	Task
Fred	Database1
Fred	Database2
Fred	Compiler1
Eugene	Database1
Eugene	Compiler1
Sara	Database1
Sara	Database2

DBProject:

Task
Database1
Database2

Completed ÷ DBProject?

Student
Fred
Sara

Exact Row Counting

- Problem: Give those rows in a table which are exactly exists only x-times in it.
- In this example we search those pilots, who can drive exactly TWO types of airplanes!

Exact Row Counting

- Steps:
  - Rename
  - Cross product
  - Selection -> Projection
  - Cross product
  - Selection -> Projection
  - Difference

N ≥ 2

N ≥ 3

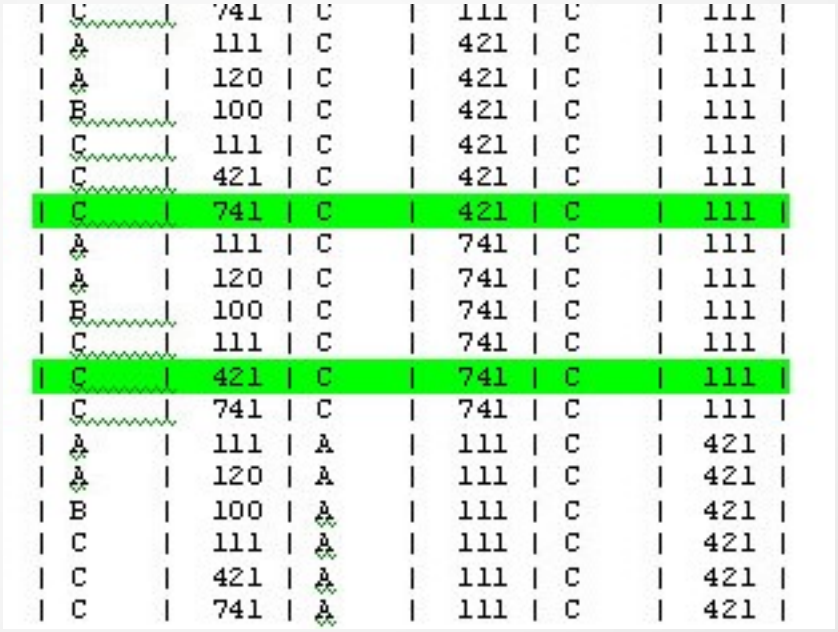
„Exact row counting“

Pilot:

Name	Type
A	111
A	120
B	100
C	111
C	421
C	741

Solution

n	t	un	ut
A	111	A	111
A	120	A	111
B	100	A	111
C	111	A	111
C	421	A	111
C	741	A	111
A	111	A	120
A	120	A	120
B	100	A	120



C	741	C	111	C	111		
A	111	C	421	C	111		
A	120	C	421	C	111		
B	100	C	421	C	111		
C	111	C	421	C	111		
C	421	C	421	C	111		
C	741	C	421	C	111		
A	111	C	741	C	111		
A	120	C	741	C	111		
B	100	C	741	C	111		
C	111	C	741	C	111		
C	421	C	741	C	111		
C	741	C	741	C	111		
A	111	A	111	C	421		
A	120	A	111	C	421		
B	100	A	111	C	421		
C	111	A	111	C	421		
C	421	A	111	C	421		
C	741	A	111	C	421		

$$\rho_{New\_Pilot(new\_name,new\_type)}(Pilot)$$

$$\rho_{New2\_Pilot(new2\_name,new2\_type)}(Pilot)$$

$$\pi_{name}(\sigma_{name=new\_name \text{ AND } type\neq new\_type}(Pilot \times New\_Pilot) -$$

$$\pi_{name}(\sigma_{name=new\_name=new2\_name \text{ AND } type\neq new\_type\neq new2\_type}((Pilot \times New\_Pilot) \times New2\_Pilot))$$

Problem 1. - continued

Consider the following schema:

- **Suppliers** (sid, sname, address)
- **Parts** (pid, pname, color)
- **Catalog** (sid, pid, cost)

Write the following queries in relational algebra:

5. Find the sids of suppliers who supply every part.
6. Find the sids of suppliers who supply every red part.
7. Find the sids of suppliers who supply every red and green part.
8. Find the sids of suppliers who supply every red part or supply every green part.
9. Find pids of parts that are supplied by at least two different suppliers
10. Find the pids of the most expensive parts supplied by “Yosemite Sham”