CMPT 321
Fall 2017

Relational algebra queries

review

By *Marina Barsky*

General

- π followed by comma-separated list of columns (dimensions) to project into
- σ followed by Boolean conditions, multiple conditions connected with Boolean operators: and, or, not
- Boolean condition requiring for value in the same column to be both A and B, **cannot** be performed with selection σ – we look at each row in turn and we cannot know if both A and B occur in this column

Three special methods

1. Finding min/max
2. Every
3. At least k

Suppliers (sname, address)
Parts (pname, color)
Catalog (sname, pname, cost)
Substitute (pname, substitute)

Finding min/max:
find part(s) with a minimum price

- There cannot be *min* operator in RA: we look at each row one at a time, so there is no way to compare values in different rows.
- Our only tool is Cartesian Product – create a new table where the answer will be clear <u>from a single row</u>…

Finding min/max: find part(s) with a minimum price

Catalog	
pname	price
A	1
B	3
C	2

x

Catalog1	
pname1	price1
A	1
A	3
B	2

Suppliers (sname, address)

Parts (pname, color)

Catalog (sname, pname, cost)

Substitute (pname, substitute)

Catalog1 (pname1, price1) = ρ ($\pi_{pname,price}$(Catalog))

Product = Catalog x Catalog1

pname	price	pname1	price1
A	1	A	1
A	1	B	3
A	1	C	2
~~B~~	~~3~~	~~A~~	~~1~~
B	3	B	3
~~B~~	~~3~~	~~C~~	~~2~~
~~C~~	~~2~~	~~A~~	~~1~~
C	2	B	3
C	2	C	2

Finding rows where *pname* cannot be min – as it is > than some other product's *price1*

Eliminate = π_{pname} ($\sigma_{price>price1}$ (Product))

Catalog
A
B
C

-

Eliminate
B
C

Min = π_{pname} (Product) - Eliminate

Every color: find parts that are offered in every color

- Given set of all colors (say, there are only 2: red and green), find parts that appear in every color.
- If part A appears in green but not in red, it is not part of the answer. If B appears in both green and red, it is part of the answer.
- The parts that are outside of the requirements (say, B also can be non-colored) are of no interest to us.
- Again, we can read only a single row at a time, and when we see part B in red, there is no way to know that we have seen it also in green

Finding parts in **every** color

PartsOfInterest	
pname	color
A	red
B	red
B	green

Colors (set)
color
red
green

Colors = π_{color}(Parts)

PartsOfInterest = $\pi_{pname,color}$(Parts ⋈ Colors)

Suppliers (sname, address)

Parts (pname, color)

Catalog (sname, pname, cost)

Substitute (pname, substitute)

The only way to discriminate between parts of type A and of type B is to find parts of type A

ArtificialProduct	
pname	color
A	red
A	green
B	red
B	green

-

PartsOfInterest	
pname	color
A	red
B	red
B	green

=

NotEvery
A

NotEvery = π_{pname} (ArtificialProduct - PartsOfInterest)

Every
B

ArtificialProduct = π_{pname} (PartsOfInterest) x (Colors))

Every = π_{pname} (Parts) - NotEvery

At least *k*: find parts that are offered in at least *k* colors

- Again, we need to bring all the information into a single row by performing k Cartesian products of table with itself

Finding parts in at least 2 colors

Parts	
pname	color
A	green
A	red
B	red

x

Parts1	
pname1	color1
A	green
A	red
B	red

Suppliers (sname, address)

Parts (pname, color)

Catalog (sname, pname, cost)

Substitute (pname, substitute)

Parts1 (pname1, color1) = $\rho\ (\pi_{\text{pname,color}}(\text{Parts}))$

pname	color	pname1	color1
A	green	A	green
A	green	A	red
A	green	B	red
A	red	A	green
A	red	A	red
A	red	B	red
B	red	A	green
B	red	A	red
B	red	B	red

Product = Parts x Parts1

Finding rows where *pname* is the same, but color and color1 are different

AtLeast2
A

AtLeast2= $\boldsymbol{\pi_{\text{pname}}(\sigma_{\text{pname=pname1 AND ccolor<color1}}(\text{Product}))}$