CS 336 Recitation
Relational Algebra & SQL Query I

Fei Deng

Mon 10AM – 11AM, Hill 202

fei.deng@rutgers.edu

Relational Algebra

- operations on relations
- results are new relations

- relation ≠ relationship
- relation = table = collection of tuples
- collection can be
  - set: no duplicates
  - bag: allow duplicates, more efficient

each row is a tuple

X	Y
1	‘a’
2	‘b’
1	‘c’
4	‘b’
5	‘e’

set

X	Y
1	‘a’
2	‘b’
1	‘a’
2	‘b’
5	‘e’

bag

Relational Algebra

- Remove parts of a relation
  - projection π_L , selection σ_C
- Combine two relations
  - product $\times$, theta-join $\bowtie_C$, natural join $\bowtie$
- Set operations
  - union $\cup$, intersection $\cap$, difference $-$
- Renaming ρ

Projection

- remove some columns

R

X	Y
1	‘a’
2	‘b’
1	‘c’
4	‘b’
5	‘e’

projection can produce duplicates

Y
‘a’
‘b’
‘c’
‘b’
‘e’

```
SELECT Y
FROM R;
```

$\pi_Y(R)$

Y
‘a’
‘b’
‘c’
‘e’

```
SELECT DISTINCT Y
FROM R;
```

Selection

- remove some rows

R

X	Y
1	‘a’
2	‘b’
1	‘c’
4	‘b’
5	‘e’

$\sigma_{X>1}(R)$

X	Y
2	‘b’
4	‘b’
5	‘e’

```
SELECT *
FROM R
WHERE X > 1;
```

Selection + Projection

R

X	Y
1	‘a’
2	‘b’
1	‘c’
4	‘b’
5	‘e’

$\sigma_{X>1}(R)$

X	Y
2	‘b’
4	‘b’
5	‘e’

```
SELECT *
FROM R
WHERE X > 1;
```

$\pi_Y(\sigma_{X>1}(R))$

Y
‘b’
‘e’

```
SELECT DISTINCT Y
FROM R
WHERE X > 1;
```

Selection + Projection

- Don't use `DISTINCT` too often! It's expensive!

R

X	Y
1	'a'
2	'b'
1	'c'
4	'b'
5	'e'

$\sigma_{X>1}(R)$

X	Y
2	'b'
4	'b'
5	'e'

```
SELECT *
FROM R
WHERE X > 1;
```

$\pi_Y(\sigma_{X>1}(R))$

Y
'b'
'b'
'e'

result as a bag

```
SELECT Y
FROM R
WHERE X > 1;
```

Product

- pair tuples from two relations

R

X	Y
1	‘a’
2	‘b’

S

Y	Z
2018	3.36
2019	3.36

need to resolve name conflict

$R \times S$

X	R.Y	S.Y	Z
1	‘a’	2018	3.36
1	‘a’	2019	3.36
2	‘b’	2018	3.36
2	‘b’	2019	3.36

```
SELECT *
FROM R, S;
```

shorthand: `R JOIN S;`

Theta-Join: Product + Selection

- pair tuples from two relations based on some condition

R

X	Y
1	‘a’
2	‘b’

S

Y	Z
2018	3.36
2019	3.36

$R \bowtie_{R.Y=‘a’ \; AND \; Z=3.36} S$

X	R.Y	S.Y	Z
1	‘a’	2018	3.36
1	‘a’	2019	3.36

$$R \bowtie_C S = \sigma_C(R \times S)$$

```
SELECT *
FROM R, S
WHERE R.Y = ‘a’ AND Z = 3.36;
```

shorthand:

```
R JOIN S ON
    R.Y = ‘a’ AND Z = 3.36;
```

Natural Join: Special Theta-Join + Projection

- special condition: common attributes take same values
- projection: remove redundant columns for common attributes

R

X	Y	Z
1	2	3
4	5	6
7	5	6

S

Y	Z	W
2	3	4
2	3	5
5	6	7

$R \bowtie S$

X	Y	Z	W
1	2	3	4
1	2	3	5
4	5	6	7
7	5	6	7

$$R \bowtie S = \pi_{X,R.Y,R.Z,W}(R \bowtie_{R.Y=S.Y \text{ AND } R.Z=S.Z} S)$$

only one copy for each common attribute

```
R NATURAL JOIN S;
```

Set Operations

- must have same number of attributes and same data types
- usually apply projection before set operations
- eliminate duplicates by default

R

X	Y
1	‘a’
2	‘b’

S

X	Z
2	20.0
3	30.0

$\pi_X(R) \cup \pi_X(S)$

X
1
2
3

```
(SELECT X
 FROM R)
UNION
(SELECT X
 FROM S);
```

$\pi_X(R) \cap \pi_X(S)$

X
2

```
(SELECT X
 FROM R)
INTERSECT
(SELECT X
 FROM S);
```

$\pi_X(R) - \pi_X(S)$

X
1

```
(SELECT X
 FROM R)
EXCEPT
(SELECT X
 FROM S);
```

Renaming

- useful when a query involves ≥2 tuples from one relation

Person(name, gender), ParentOf(paName, chName)
Query: find all (mother, son) pairs

```
SELECT Pa.name, Ch.name
FROM Person AS Pa, Person AS Ch, ParentOf
WHERE paName = Pa.name AND chName = Ch.name
    AND Pa.gender = 'female' AND Ch.gender = 'male';
```

$$\pi_{Pa.name,Ch.name}\left(\sigma_C(\rho_{Pa}(Person) \times \rho_{Ch}(Person) \times ParentOf)\right)$$

- Supplier (sid, sname)
- Part (pid, pname, color)
- Catalog (sid, pid, cost)

- Find the pname's of parts for which there is some supplier
- Find the sid's of suppliers who supply a red part AND a green part
- Find the sid's of suppliers who DO NOT supply any red part
- Find the sid's of suppliers who supply ONLY red parts
  (Hint: Find the sid's of suppliers who do not supply non-red parts)

- Supplier (sid, sname)
- Part (pid, pname, color)
- Catalog (sid, pid, cost)

- Find the pname's of parts for which there is some supplier

$$\pi_{pname}(Part \bowtie Catalog)$$

```
SELECT pname
FROM Part NATURAL JOIN Catalog;
```

use theta-join when primary key and foreign key do not have the same name

- Supplier (sid, sname)
- Part (pid, pname, color)
- Catalog (sid, pid, cost)

- Find the sid's of suppliers who supply a red part AND a green part

$$\pi_{sid}\left(\sigma_{color='red'}(Part \bowtie Catalog)\right) \cap \pi_{sid}\left(\sigma_{color='green'}(Part \bowtie Catalog)\right)$$

```
(SELECT sid
 FROM Part NATURAL JOIN Catalog
 WHERE color = 'red')
    INTERSECT
(SELECT sid
 FROM Part NATURAL JOIN Catalog
 WHERE color = 'green');
```

- Supplier (sid, sname)
- Part (pid, pname, color)
- Catalog (sid, pid, cost)

- Find the sid's of suppliers who DO NOT supply any red parts

$$\pi_{sid}(Supplier) - \pi_{sid}\left(\sigma_{color='red'}(Part \bowtie Catalog)\right)$$

```
(SELECT sid
 FROM Supplier)
    EXCEPT
(SELECT sid
 FROM Part NATURAL JOIN Catalog
 WHERE color = 'red');
```

- Supplier (sid, sname)
- Part (pid, pname, color)
- Catalog (sid, pid, cost)

- Find the sid's of suppliers who supply ONLY red parts
  (Hint: Find the sid's of suppliers who do not supply non-red parts)

$$\pi_{sid}(\sigma_{color='red'}(Part \bowtie Catalog)) - \pi_{sid}(\sigma_{color\neq'red'}(Part \bowtie Catalog))$$

```
(SELECT sid
 FROM Part NATURAL JOIN Catalog
 WHERE color = 'red')
    EXCEPT
(SELECT sid
 FROM Part NATURAL JOIN Catalog
 WHERE color != 'red');
```

- Supplier (sid, sname)
- Part (pid, pname, color)
- Catalog (sid, pid, cost)

- Find the sid's of suppliers who supply EVERY part

$$\pi_{sid,pid}(Catalog)/\pi_{pid}(Part)$$

Division

- $R(X, Y),\ \ S(Y)$
- $Q = R/S = \{x \in \pi_X(R) | \{x\} \times S \subseteq R\}$

R

X	Y
x1	y1
x1	y2
x2	y1

S

Y
y1
y2

$\pi_X(R)$

X
x1
x2

$\{x_1\} \times S$

X	Y
x1	y1
x1	y2

$\{x_1\} \times S \subseteq R$

$\{x_2\} \times S$

X	Y
x2	y1
x2	y2

$\{x_2\} \times S \nsubseteq R$

R/S

X
x1

- $Q \times S \subseteq R,\ \ (\pi_X(R) - Q) \times S \nsubseteq R$