Multisets and Aggregation

Dr Paolo Guagliardo

THE UNIVERSITY *of* EDINBURGH informatics

Fall 2020 (v20.1.1)

Duplicates

R

A	B
a1	b1
a2	b2
a1	b2

$\pi_A(R)$

A
a1
a2

SELECT A **FROM** R

A
a1
a2
a1

- We considered relational algebra on **sets**
- SQL uses **bags**: sets with duplicates

Multisets (a.k.a. bags)

Sets where the **same element** can occur **multiple times**

The number of occurrences of an element is called its multiplicity

Notation

$a \in_k B$: a occurs k times in bag B

$a \in B$: a occurs in B with multiplicity ≥ 1

$a \notin B$: a does not occur in B (that is, $a \in_0 B$)

Relational algebra on bags

Relations are **bags of tuples**

Projection

Keeps duplicates

$$\pi_A \left(\begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 2 & 3 \\ 1 & 1 \\ 2 & 2 \end{array} \right) = \begin{array}{c} \textbf{A} \\ \hline 2 \\ 1 \\ 2 \end{array}$$

Relational algebra on bags

Cartesian product

Concatenates tuples as many times as they occur

$$\begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 1 & 1 \\ & \end{array} \times \begin{array}{c} \textbf{C} \\ \hline 2 \\ 2 \end{array} = \begin{array}{ccc} \textbf{A} & \textbf{B} & \textbf{C} \\ \hline 1 & 1 & 2 \\ 1 & 1 & 2 \end{array}$$

Relational algebra on bags

Selection

Takes all occurrences of tuples satisfying the condition:

$$\text{If } \bar{a} \in_k R, \quad \text{then } \begin{cases} \bar{a} \in_k \sigma_\theta(R) & \text{if } \bar{a} \text{ satisfies } \theta \\ \bar{a} \notin \sigma_\theta(R) & \text{otherwise} \end{cases}$$

Example

$$\sigma_{A>1}\left(\begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 2 & 3 \\ 1 & 2 \\ 2 & 3 \end{array}\right) = \begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 2 & 3 \\ 2 & 3 \end{array}$$

Relational algebra on bags

Duplicate elimination ε

New operation that removes duplicates:

$$\text{If } \bar{a} \in R, \quad \text{then } \bar{a} \in_1 \varepsilon(R)$$

Example

$$\varepsilon\left(\begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 2 & 3 \\ 1 & 2 \\ 2 & 3 \end{array}\right) = \begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 2 & 3 \\ 1 & 2 \end{array}$$

Relational algebra on bags

Union

Adds multiplicities:

$$\text{If } \bar{a} \in_k R \text{ and } \bar{a} \in_n S, \quad \text{then } \bar{a} \in_{k+n} R \cup S$$

Example

$$\begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 1 & 2 \\ 1 & 2 \\ 1 & 3 \\ & \\ & \\ & \end{array} \cup \begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ & \\ & \\ & \end{array} = \begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 1 & 2 \\ 1 & 2 \\ 1 & 3 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{array}$$

Relational algebra on bags

Intersection

Takes the **minimum** multiplicity:

$$\text{If } \bar{a} \in_k R \text{ and } \bar{a} \in_n S, \quad \text{then } \bar{a} \in_{\min\{k,n\}} R \cap S$$

Example

$$\begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 1 & 2 \\ 1 & 2 \\ 1 & 3 \end{array} \quad \cap \quad \begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{array} \quad = \quad \begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 1 & 2 \\ 1 & 3 \end{array}$$

Relational algebra on bags

Difference

Subtracts multiplicities up to zero:

$$\text{If } \bar{a} \in_k R \text{ and } \bar{a} \in_n S, \text{ then } \begin{cases} \bar{a} \in_{k-n} R - S & \text{if } k > n \\ \bar{a} \notin \quad R - S & \text{otherwise} \end{cases}$$

Example

$$\begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 1 & 2 \\ 1 & 2 \\ 1 & 3 \end{array} \quad - \quad \begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 1 & 2 \\ 1 & 3 \\ 1 & 3 \\ 1 & 4 \end{array} \quad = \quad \begin{array}{cc} \textbf{A} & \textbf{B} \\ \hline 1 & 2 \end{array}$$

RA on sets vs. RA on bags

Equivalences of RA on sets **do not** necessarily hold on **bags**

Example

On bags $\sigma_{\theta_1 \vee \theta_2}(R) \not\equiv \sigma_{\theta_1}(R) \cup \sigma_{\theta_2}(R)$

R	**A**
	2

$\sigma_{A>1 \vee A<3}(R)$	**A**
	2

$\sigma_{A>1}(R) \cup \sigma_{A<3}(R)$	**A**
	2
	2

$\varepsilon\big(\sigma_{\theta_1 \vee \theta_2}(R)\big) \equiv \varepsilon\big(\sigma_{\theta_1}(R) \cup \sigma_{\theta_2}(R)\big)$ holds

Basic SQL queries revisited

$$
\begin{aligned}
Q := {} & \texttt{SELECT}\ [\texttt{DISTINCT}]\ \alpha\ \texttt{FROM}\ \tau\ \texttt{WHERE}\ \theta \\
& \mid Q_1\ \texttt{UNION}\ [\texttt{ALL}]\ Q_2 \\
& \mid Q_1\ \texttt{INTERSECT}\ [\texttt{ALL}]\ Q_2 \\
& \mid Q_1\ \texttt{EXCEPT}\ [\texttt{ALL}]\ Q_2
\end{aligned}
$$

SQL and RA on bags

SQL	RA on bags
SELECT α ...	$\pi_\alpha(\cdot)$
SELECT DISTINCT α ...	$\varepsilon\big(\pi_\alpha(\cdot)\big)$
Q_1 **UNION ALL** Q_2	$Q_1 \cup Q_2$
Q_1 **INTERSECT ALL** Q_2	$Q_1 \cap Q_2$
Q_1 **EXCEPT ALL** Q_2	$Q_1 - Q_2$
Q_1 **UNION** Q_2	$\varepsilon(Q_1 \cup Q_2)$
Q_1 **INTERSECT** Q_2	$\varepsilon(Q_1 \cap Q_2)$
Q_1 **EXCEPT** Q_2	$\varepsilon(Q_1) - Q_2$

Duplicates and aggregation (1)

Customer

ID	Name	City	Age
1	John	Edinburgh	31
2	Mary	London	37
3	Jane	London	22
4	Jeff	Cardiff	22

Average age of customers: **avg**$\big(\pi_{\text{Age}}(\text{Customer})\big)$

- If we remove duplicates we get $\frac{31+37+22}{3} = 30$ **(wrong)**

SQL keeps duplicates by default:

```
SELECT AVG(age)
FROM   Customer ;
```

Duplicates and aggregation (2)

Account

Number	Branch	CustID	Balance
111	London	1	1330.00
222	London	2	1756.00
333	Edinburgh	1	450.00

Number of branches: $\left|\varepsilon\left(\pi_{\text{Branch}}(\text{Account})\right)\right|$

- If we keep duplicates we get 3 **(wrong)**

In SQL:

```
SELECT COUNT(DISTINCT branch)
FROM   Account ;
```

Aggregate functions in SQL

COUNT	number of elements in a column
AVG	average value of elements in a column
SUM	adds up all elements in a column
MIN	minimum value of elements in a column
MAX	maximum value of elements in a column

- Using **DISTINCT** with **MIN** and **MAX** makes no difference
- **COUNT**(*) counts all rows in a table
- **COUNT**(**DISTINCT** *) is **illegal**

  To count all distinct rows of a table T use

```
SELECT COUNT(DISTINCT T.*)
FROM   T ;
```

Aggregation and empty tables

Suppose table T has a column (of numbers) called A

```
SELECT MIN(A),MAX(A),AVG(A),SUM(A),COUNT(A),COUNT(*)
FROM   T
WHERE  1=2 ;
```

```
 min | max | sum | avg | count | count
-----+-----+-----+-----+-------+-------
     |     |     |     |     0 |     0
```