Lecture 11

MULTISETS & AGGREGATION

MULTISETS (a.k.a. BAGS)

→ are sets where the same element can occur multiple times

R	
A	B
a1	b1
a2	b2
a1	b2

$\pi_A(R)$
A
a1
a2

We consider RA on **sets**

SELECT A FROM R
A
a1
a2
a1

SQL uses **bags**
↳ sets w/ duplicates

→ The no. 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 w/ multiplicity ≥ 1
$a \notin B$	a does not occur in B → $a \in_0 B$

RELATIONAL ALGEBRA ON BAGS

→ We now consider relations as bags of tuples

PROJECTION

- keeps duplicates

i.e.

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

CARTESIAN PRODUCT

- Concatenates tuples as many times as they occur

i.e.

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

↓

A	B	C
1	1	2
1	1	2

SELECTION

→ Takes all occurrences of tuples satisfying the condition

i.e. $\sigma_{A>1}$

A	B
2	3
1	2
2	3

$=$

A	B
2	3
2	3

DUPLICATE ELIMINATION

→ removes duplicates

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

i.e. ε

A	B
2	3
2	3
1	1

$=$

A	B
2	3
1	1

UNION

→ Adds multiplicities

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

i.e.

A	B
1	2
1	2
1	3

$\cup$

A	B
1	2
1	3
1	4

$=$

A	B
1	2
1	2
1	3
1	2
1	3
1	4

INTERSECTION

→ Takes the minimum multiplicity

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

i.e.

A	B
1	2
1	2
1	3

$\cap$

A	B
1	2
1	3
1	4

$=$

A	B
1	2
1	3

DIFFERENCE

→ Subtracts multiplicities up to zero —— If $\bar{a} \in_k R$ and $\bar{a} \in_n S$, then $\begin{cases} \bar{a} \in_{k-n} R-S & \text{if } k>n \\ \bar{a} \notin R-S & \text{otherwise} \end{cases}$

i.e.

A	B
1	2
1	2
1	3

$-$

A	B
1	2
1	3
1	3
1	4

$=$

A	B
1	2

RA ON SETS VS. RA ON BAGS

→ Equivalences of RA on sets do NOT necessarily hold on bags

→ i.e. 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

→ Notice that $\varepsilon(\sigma_{\theta_1 \vee \theta_2}(R)) \equiv \varepsilon(\sigma_{\theta_1}(R) \cup \sigma_{\theta_2}(R))$ holds

SQL and RA ON BAGS

keeps duplicates and need to explicitly say to remove them.

Q := SELECT [DISTINCT] α FROM τ WHERE θ

| Q_1 UNION [ALL] Q_2
| Q_1 INTERSECT [ALL] Q_2
| Q_1 EXCEPT [ALL] Q_2

} when you do these operations, the default is that duplicates are removed and you need to explicitly say when you want to keep them

SQL	RA on bags
SELECT α ...	$\pi_a(\cdot)$
SELECT DISTINCT α ...	$\varepsilon(\pi_a(\cdot))$
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	$\varepsilon(Q_1) - Q_2 \not\equiv \varepsilon(Q_1 - Q_2)$ $\equiv$ $\varepsilon(Q_1) - \varepsilon(Q_2)$

AGGREGATE FNs IN SQL

- COUNT no. of elems. in a column
- AVG avg. value of elems in a column
- SUM adds up all elems. in a column
- MIN min. value of elems. in a column
- MAX max. value of elems. in a column

} Using DISTINCT here makes no difference

- COUNT (*) ✓ counts all rows in a table

COUNT (DISTINCT *) ✗

↳ To count all distinct rows of a table T use:

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

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
NULL	NULL	NULL	NULL	0	0

There is one row and the value is NULL