However if $a \to b$

then we can do:

$$|\gamma^{a} R| = |\gamma^{a,b} R| = |\Pi_{a,b}(R \bowtie \gamma^{a} R)|$$

i.e. all queries return the same number of tuples

And:

$$\gamma^{a,b} R = \Pi_{a,b}(R \bowtie \gamma^{a} R)$$

But more frequently you will need:

Assume $R(\underline{a,b,c})$, $S(\underline{a,d})$ (FK: a of R → a of S)

$$\gamma^{a,b}_{count(d)}(R \bowtie S) =$$

$$\Pi_{a,b,count(d)}(R \bowtie \gamma^{a}_{count(d)} S)$$

But only because $a \to b$!!

Ex: Find id and name of student and the number of courses she/he is registered in.

Aggregation.

Frequently it is necessary to summarize a set of tuples into only one.

Ex:

- How many tuples satisfy this condition?
- What is the average of this attribute?

γ group-by operator

In its simplest form $\gamma_{\langle seq.\ aggr.\ exps \rangle} R$ computes a sequence of aggregation expressions on a relation R.

Aggregation functions.

Given a set of tuples or attributes, compute a single value.

- count(*) Count number of tuples in set.
- count(att) Count number of tuples with attribute not NULL
- sum(att) Sums the value of attr.
- $avg(att) = \frac{sum(att)}{count(att)}$
- max(att), min(att).

Example

R(a, b, c)

a	b	c
7	a	⊥ ← NULL
2	x	-1
5	y	5

$\gamma_{count(*),\ sum(a),\ count(c)}\ R$

"count(*)"	"sum(a)"	"count(c)"
3	12	2

$\gamma_{\frac{sum(a)}{count(a)} \to aavg}\ R$ — rename attribute

aavg
4

Grouping

Sometimes we need to make summaries of different subsets of tuples.

Ex: How many courses is each student taking?

- What is the average price of each part?

Be careful:

$\gamma_b^a R$, $\Pi_b \gamma_b^a R$, $\Pi_b \gamma^a R$

are all illegal

Remember: the schema of γ does not contain attributes of R not listed in the grouping attributes $\gamma^{<list\ att>}$

---

mySQL allows this:

$\gamma_b^a R$

Value of b is non deterministic. Chosen at random from one tuple in grouping subset.

We don't like NON DETERMINISM Unless you know what you're doing.

Instead use:

$\Pi_{a,b}[R \bowtie \gamma^a R]$

We can combine operations:

$\Pi_{count(c)}\ \sigma_{count(c)>1}\ \gamma^{a}_{count(c)}\ \sigma_{b>3}\ R$

```
SELECT count(c) FROM
  ( SELECT a, count(c)
    FROM R
    WHERE b>3
    GROUP BY a ) AS X
WHERE count(c) > 1;
```

Any subquery requires a name

↑ selection on result of aggregation

$\Pi\ \sigma_q$ of $\gamma\ \sigma_p$ is so common that SQL has syntantic sugar for it:

```
SELECT count(c)
FROM R
WHERE b>3
GROUP BY a
HAVING count(c)>1.
```

Ex: Find the student id of students who are taking 3 or more carsers.

$\gamma^{<att\ list>}\ R$

Creates one tuple for each different value of the list of attributes.

Ex. R(a, b, c)

a	b	c
3	9	1
2	5	4
3	9	5
2	1	8

$\gamma^{a,b}\ R$

a	b
3	9
2	5
2	1

$\gamma^{a}\ R$

a
3
2

$\gamma^{c}\ R$

c
1
4
5
8

Warning: This is my notation.

In fact, our textbook does not even include γ in its RA chapter.

SQL

$\gamma_{count(*),\,count(a)}\,R$

Remember, it returns only one tuple

```
SELECT count(*), count(a)
FROM R;
```

This is not a $\Pi_{count(*),\,count(a)}$ But it can be interpreted as

$\Pi_{count(*),\,count(a)}\,\gamma_{count(*),\,count(a)}\,R$

Redundant in this case.

$\gamma^{a,b}\,R$

```
SELECT a,b FROM R
GROUP BY a,b
```

Yes, redundant but necessary

REMOVES DUPLICATES!!

Equivalent to:

```
SELECT DISTINCT a,b FROM R
```

$\Rightarrow \Pi_{a,b}\,R = \gamma^{a,b}\,R$

only in RA (relations are sets)

Combining both:

$\gamma^{\langle list\ of\ attr\rangle}_{\langle list\ of\ agg\ exp\rangle}\,R$

Computes the expressions on each subset of different values of attributes.

Ex:

$R(a,b,c)$

a	b	c
3	9	1
2	5	4
3	9	5
2	1	8

$\gamma^{a}_{avg(c),\,count(*)}\,R$

a	"avg(c)"	"count(*)"
3	5	2
2	6	2

```
SELECT count(c), count(*)
FROM R
GROUP BY a
```