

# Bases de données (3)

## *Algèbre relationnelle*

Stéphane Gonnord, Laurent Jouhet

Lycée du parc - Lyon

# Plan

SQL vs Maths

Projections et sélections

Jointures

Agrégation

Autres opérateurs

## Vous comprenez ça ?

$\gamma_{MAX(c)} (dep \gamma_{COMPTAGE:c} (departements \bowtie_{J_{CD}} (\sigma_{pop \geq 10^4} communes)))$

## Vous comprenez ça ?

$$\gamma_{MAX(c)}(\text{dep} \gamma_{COMPTAGE:c}(\text{departements} \bowtie_{J_{CD}} (\sigma_{pop \geq 10^4} \text{communes})))$$

```
SELECT max(c)
```

```
FROM
```

```
(SELECT dep, COUNT(*) AS c
```

```
FROM departements JOIN communes
```

```
ON departements.id=communes.dep
```

```
WHERE pop >= 10000
```

```
GROUP BY dep)
```

## Vous comprenez ça ?

$$\gamma_{MAX(c)} \left( \text{dep} \gamma_{COMPTAGE:c} \left( \text{departements} \bowtie_{J_{CD}} \left( \sigma_{pop \geq 10^4} \text{communes} \right) \right) \right)$$

```
SELECT max(c)
FROM
  (SELECT dep, COUNT(*) AS c
   FROM departements JOIN communes
   ON departements.id=communes.dep
   WHERE pop>=10000
   GROUP BY dep)
```

Quel est le nombre maximal de communes de plus de 10000 habitants par département ?

## L'objet de l'algèbre relationnelle

- Manipuler des *relations* (ensembles de tuples *typés*).
- Des opérateurs pour construire des relations à partir d'autres relations.
- Binaires : union, intersection, ... produit, jointure (*et division*).
- Unaires : projections, sélections.
- Des opérateurs d'agrégation.

# Exemples

Trois relations différentes

<b>communes</b>			
<u>id</u>	dep	nom	pop
...	...	...	...
69023	69	Lyon	484344
...	...	...	...
2B050	2B	Calvi	5394
...	...	...	...

<b>departements</b>		
<u>id</u>	reg	nom
...	...	...
69	82	Rhône
...	...	...
2B	94	Haute-Corse
...	...	...

<b>regions</b>	
<u>id</u>	nom
...	...
82	Rhône-Alpes
...	...
94	Corse
...	...

# Exemples

Et trois de plus

<b>eleves</b>		
<u>ide</u>	nom	prenom
0	Lions	Jacques-Louis
1	Laurent	Jean
...	...	...

<b>profs</b>		
<u>idp</u>	nom	prenom
0	Théron	Pierre
1	Brun	Jules
...	...	...

<b>colles</b>			
prof	eleve	semaine	note
2	8	1	16
1	0	6	19
...	...	...	...



# Projections

- Permet de choisir des *colonnes*
- Analogue de `SELECT`, attention !
- $\pi_{A_1, \dots, A_k} R$  ou  $\pi_{A_1, \dots, A_k}(R)$  : on ne prend que les attributs  $A_1, \dots, A_k$

# Projections

- Permet de choisir des *colonnes*
- Analogue de `SELECT`, attention !
- $\pi_{A_1, \dots, A_k} R$  ou  $\pi_{A_1, \dots, A_k}(R)$  : on ne prend que les attributs  $A_1, \dots, A_k$
- Exemple :
  - En français : «Donner les élèves (identifiants) et les notes pour toutes les colles.»
  - En SQL : `SELECT eleve, note FROM colles.`
  - En algèbre relationnelle :  $\pi_{eleve, note}(colles)$ .

<i>colles</i>					$\pi_{eleve, note}(colles)$	
prof	eleve	semaine	note		eleve	note
2	8	1	16	→	8	16
1	0	6	19		0	19
...	...	...	...		...	...

## Sélections/restrictions

- Permet de choisir des *lignes*
- Conditions de type `WHERE . . .`
- $\sigma_C R$  ou  $\sigma_C(R)$ , avec  $C$  la condition (formule logique portant sur les attributs)



## On peut composer !

- Exemple : nom des villes de plus de 100000 habitants : traduire en SQL et algèbre relationnelle.

## On peut composer !

- Exemple : nom des villes de plus de 100000 habitants : traduire en SQL et algèbre relationnelle.
- `SELECT nom FROM communes WHERE pop >= 100000`
- $\pi_{nom} \sigma_{pop \geq 10^5}(\textit{communes})$ .

## On peut composer !

- Exemple : nom des villes de plus de 100000 habitants : traduire en SQL et algèbre relationnelle.
- `SELECT nom FROM communes WHERE pop>=100000`
- $\pi_{nom}\sigma_{pop\geq 10^5}(communes)$ .

### Exercice

Est-ce que ça commute ?

$$\pi_{nom}(\sigma_{pop\geq 10^5}(communes)) = ? \sigma_{pop\geq 10^5}(\pi_{nom}(communes))$$

$$\pi_{leve,note}(\sigma_{note\geq 19}(colles)) = ? \sigma_{note\geq 19}(\pi_{leve,note}(colles))$$

## Produit : beurk

- Deux tables :

<b>communes</b>	
nom	dep
Lyon	69
Calvi	2B
Corte	2B

<b>departements</b>	
<u>id</u>	nom
69	Rhône
2B	Haute-Corse



## Produit : beurk

- Deux tables :

<b>communes</b>	
nom	dep
Lyon	69
Calvi	2B
Corte	2B

<b>departements</b>	
<u>id</u>	nom
69	Rhône
2B	Haute-Corse

- Et leur produit :

<b>communes × départements</b>			
nom	dep	<u>id</u>	nom
Lyon	69	69	Rhône
Lyon	69	2B	Haute-Corse
Calve	2B	69	Rhône
Calvi	2B	2B	Haute-Corse
Corte	2B	69	Rhône
Corte	2B	2B	Haute-Corse

## Jointure : produit avec sélection

- Formellement :

$$R_1 \underset{C}{\bowtie} R_2 = \sigma_C(R_1 \times R_2)$$

## Jointure : produit avec sélection

- Formellement :

$$R_1 \bowtie_C R_2 = \sigma_C(R_1 \times R_2)$$

- `t1 JOIN t2 ON (C)` et `t1, t2 WHERE (C)` sont proches... mais différentes ? (Boîte noire) !

## Jointure : produit avec sélection

- Formellement :

$$R_1 \bowtie_C R_2 = \sigma_C(R_1 \times R_2)$$

- `t1 JOIN t2 ON (C)` et `t1, t2 WHERE (C)` sont proches... mais différentes ? (Boîte noire) !
- Par exemple :  $C = (dep = id)$

<b>communes</b> $\bowtie_C$ <b>departements</b>			
nom	dep	<u>id</u>	nom
Lyon	69	69	Rhône
Calvi	2B	2B	Haute-Corse
Corte	2B	2B	Haute-Corse

# Fonctions d'agrégation

- But : regrouper des lignes, et évaluer une fonction sur ces regroupements.

## Fonctions d'agrégation

- But : regrouper des lignes, et évaluer une fonction sur ces regroupements.
- Syntaxe strange :

$$A_1, \dots, A_k \gamma_{f_1(B_1), \dots, f_i(B_i)} table$$

- « Regroupe selon les attributs  $A_i$ , et calcule les valeurs  $f_i$  sur les attributs  $B_i$  »

## Fonctions d'agrégation

- But : regrouper des lignes, et évaluer une fonction sur ces regroupements.
- Syntaxe strange :

$$A_1, \dots, A_k \gamma_{f_1(B_1), \dots, f_i(B_i)} table$$

- « Regroupe selon les attributs  $A_j$ , et calcule les valeurs  $f_j$  sur les attributs  $B_j$  »
- SQL :

```
SELECT A1, ..., Ak, f1(B1), ..., fi(Bi)
FROM table
GROUP BY A1, ..., Ak
```

## Fonctions d'agrégation

- But : regrouper des lignes, et évaluer une fonction sur ces regroupements.
- Syntaxe strange :

$$A_1, \dots, A_k \gamma_{f_1(B_1), \dots, f_i(B_i)} table$$

- « Regroupe selon les attributs  $A_j$ , et calcule les valeurs  $f_j$  sur les attributs  $B_j$  »
- SQL :  

```
SELECT A1, ..., Ak, f1(B1), ..., fi(Bi)
FROM table
GROUP BY A1, ..., Ak
```
- Attention, les autres champs sont perdus ! *En particulier lors du calcul du maximum !*



# Exemples

- Population des différents départements :

## Exemples

- Population des différents départements :

*dep*  $\gamma$  *SOMME(pop)* (*communes*)

## Exemples

- Population des différents départements :

*dep*  $\gamma$  *SOMME*(*pop*)(*communes*)

- Avec le nom du département ?

## Exemples

- Population des différents départements :

$$\text{dep} \gamma_{\text{SOMME}(\text{pop})}(\text{communes})$$

- Avec le nom du département ?

$$\sigma_n(\text{dep, departements.nom:n} \gamma_{\text{SOMME}(\text{pop})}(R)),$$

où  $R = \text{communes} \bowtie \text{departements}$

## Exemples

- Population des différents départements :

$$dep \gamma_{SOMME(pop)}(communes)$$

- Avec le nom du département ?

$$\sigma_n (dep, departements.nom:n \gamma_{SOMME(pop)}(R)) ,$$

où  $R = communes \bowtie departements$

- Moyenne des différents élèves ? (sans les noms)

## Exemples

- Population des différents départements :

$$dep \Upsilon_{SOMME(pop)}(communes)$$

- Avec le nom du département ?

$$\sigma_n (dep, departements.nom:n \Upsilon_{SOMME(pop)}(R)) ,$$

où  $R = communes \bowtie departements$

- Moyenne des différents élèves ? (sans les noms)

$$eleve \Upsilon_{MOYENNE(note)}colles$$

## Exemples

- Population des différents départements :

$$dep \Upsilon_{SOMME(pop)}(communes)$$

- Avec le nom du département ?

$$\sigma_n (dep, departements.nom:n \Upsilon_{SOMME(pop)}(R)) ,$$

où  $R = communes \bowtie departements$

- Moyenne des différents élèves ? (sans les noms)

$$eleve \Upsilon_{MOYENNE(note)} colles$$

- Et enfin :  $\Upsilon_{MAX(c)} (dep \Upsilon_{COMPTAGE:c} (departements \bowtie_{J_{CD}} (\sigma_{pop \geq 10^4} communes)))$

- Union
- Intersection
- Différence

Uniquement quand ça a du sens !