

# Persistence des données I

## DON2

Denis Boigelot, Geneviève Cuvelier, Selim Rexhep, Yannick Voglaire



Haute École Bruxelles-Brabant  
École Supérieure d'Informatique

Année académique 2020 / 2021

## Plan du cours

- 0 – Présentation
- 1 – Introduction
- 2 – Dépendance fonctionnelle
- 3 – Schéma conceptuel
- 4 – Projection et sélection
- 5 – Jointure
- 6 – Agrégat
- 7 – Sous-requête
- 8 – Fichiers

## 6 – Agrégat

- 1 Principe
- 2 Les fonctions agrégatives
- 3 La clause GROUP BY
- 4 Algèbre relationnelle
- 5 SQL - Regroupement
- 6 SQL - Sélection de groupes
- 7 SQL - Regroupement - Exemples
- 8 SQL - Ordre des lignes

# Principe

Dans ce chapitre, nous allons présenter des requêtes permettant de considérer des **groupes** de lignes.

Exemple : Comment pourrions-nous faire pour compter le nombre de lignes d'une table ? Pour compter le nombre de lignes qui satisfont certaines conditions ? Pour connaître la valeur maximum d'une colonne d'un certain groupe de ligne ?

# Principe

Vous avez déjà vu les **fonctions agrégatives** : COUNT, SUM, MIN, MAX, AVG qui permettent de calculer le nombre de valeurs différentes, la somme/minimum/maximum/moyenne d'une colonne donnée (ou plus généralement : d'une expression algébrique portant sur une ou plusieurs colonnes.)

Nous allons maintenant introduire la clause **GROUP BY** qui permet de grouper les lignes d'une table selon certains critères et d'afficher, pour chaque groupe, une (ou plusieurs) valeurs données.

# Principe

Voici quelques exemples d'application :

Vous avez vu comment obtenir : Combien de clients sont-ils enregistrés dans ma base de donnée ? Combien de commandes ? Quelle est la valeur moyenne du compte d'un client enregistré ?

Nous allons voir : Comment afficher, pour chaque localité (Namur, Poitiers, ...) le nombre de clients présents dans cette localité ?

- 1 Principe
- 2 Les fonctions agrégatives**
- 3 La clause GROUP BY
- 4 Algèbre relationnelle
- 5 SQL - Regroupement
- 6 SQL - Sélection de groupes
- 7 SQL - Regroupement - Exemples

# Les fonctions agrégatives

Rappel : Les fonctions d'agrégat les plus courantes :

- ◇ COUNT
- ◇ SUM
- ◇ MIN
- ◇ MAX
- ◇ AVG

Les valeurs nulles **ne sont pas** reprises par les fonctions d'agrégation.



- 1 Principe
- 2 Les fonctions agrégatives
- 3 La clause GROUP BY**
- 4 Algèbre relationnelle
- 5 SQL - Regroupement
- 6 SQL - Sélection de groupes
- 7 SQL - Regroupement - Exemples

# Grouper des lignes

LOCALITES			
NCLI	NOM	LOCALITE	COMPTE
F400	JACOB	Bruxelles	0
B332	MONTI	Genève	0
K111	VANBIST	Lille	720
S127	VANDERKA	Namur	-4580
L422	FRANCK	Namur	0
C123	MERCIER	Namur	-2300
B062	GOFFIN	Namur	-3200
S712	GUILLAUME	Paris	0
F010	TOUSSAINT	Poitiers	0
B112	HANSENNE	Poitiers	1250
C400	FERARD	Poitiers	350
C003	AVRON	Toulouse	-1700
B512	GILLET	Toulouse	-8700
F011	PONCELET	Toulouse	0
K729	NEUMAN	Toulouse	0
D063	MERCIER	Toulouse	-2250

→ le groupe des clients de Genève

→ le groupe des clients de Namur

→ le groupe des clients de Poitiers

## Grouper des lignes

```
SELECT localite, COUNT(*) as NOMBRE_CLIENTS,
       AVG(COMPTE) as MOYENNE_COMPTE
FROM client
GROUP BY localite ;
```

LOCALITE	NOMBRE_CLIENTS	MOYENNE_COMPTE	
Bruxelles	1	0.00	
Geneve	1	0.00	→ le groupe des clients de Genève
Lille	1	720.00	
Namur	4	-2520.00	→ le groupe des clients de Namur
Paris	1	0.00	
Poitiers	3	533.33	→ le groupe des clients de Poitiers
Toulouse	5	-2530.00	

On s'intéresse aux **localités** et non plus aux clients

- 1 Principe
- 2 Les fonctions agrégatives
- 3 La clause GROUP BY
- 4 Algèbre relationnelle**
- 5 SQL - Regroupement
- 6 SQL - Sélection de groupes
- 7 SQL - Regroupement - Exemples

## Algèbre relationnelle - Introduction

Comment pourrions-nous exprimer un GROUP BY au moyen d'une expression d'algèbre relationnelle ?

Pour rappel, on décide de représenter une table comme une relation  $R$ . La projection est représentée par le symbole  $\pi$ , la sélection par le symbole  $\sigma$ , et nous allons introduire un nouveau symbole pour l'agrégation :  $\gamma$ .

# Algèbre relationnelle - Introduction

La relation R :

x	y	z	t
1	a	12	x
2	a	12	x
3	a	4	x
4	a	3	y
5	a	12	z
6	a	6	z
7	b	1	x
8	b	12	y
9	c	4	x
10	d	5	t
11	d	5	u
12	x	12	t
13	y	4	t

# Algèbre relationnelle - Agrégat

$\gamma_{\{ \text{liste d'attributs pour le groupement} ; \text{liste d'expressions affichées} \}} (Relation)$

R :

x	y	z	t
1	a	12	x
2	a	12	x
3	a	3	x
4	a	3	y
5	a	12	z
6	a	6	z
7	b	1	x
8	b	12	y
9	c	4	x
10	d	5	t
11	d	5	u
12	x	12	t
13	y	4	t

$\gamma_{\{y; \max(z)\}} (R) =$

y	Max(z)
a	12
b	12
c	4
d	5
x	12
y	4

# Algèbre relationnelle - Groupement des nulls

$\gamma_{\{ \text{liste d'attributs pour le groupement} ; \text{liste d'expressions affichées} \}} (Relation)$

R :

x	y	z	t
1	a	12	x
2	a	12	x
3	a	3	x
4	a	3	y
5	a	12	z
6	a	6	z
7	b	1	x
8	b	12	y
9	c	4	x
10	d	5	t
11	d	5	u
12	x	12	t
13	y	4	t

$$\gamma_{\{t, \max(z)\}} (R) =$$

t	Max(z)
t	12
u	5
x	12
y	12
z	12



# Algèbre relationnelle - La relation = un groupe

$\gamma_{\{ \text{liste d'attributs pour le groupement} ; \text{liste d'expressions affichées} \}} (Relation)$

R :

x	y	z	t
1	a	12	x
2	a	12	x
3	a	3	x
4	a	3	y
5	a	12	z
6	a	6	z
7	b	1	x
8	b	12	y
9	c	4	x
10	d	5	t
11	d	5	u
12	x	12	t
13	y	4	t

$$\gamma_{\{ \text{max}(z) \}} (R) =$$

Max(z)
12

# Algèbre relationnelle - Sous groupes

$\gamma_{\{ \text{liste d'attributs pour le groupement} ; \text{liste d'expressions affichées} \}} (Relation)$

R :

x	y	z	t
1	a	12	x
2	a	12	x
3	a	3	x
4	a	3	y
5	a	12	z
6	a	6	z
7	b	1	x
8	b	12	y
9	c	4	x
10	d	5	t
11	d	5	u
12	x	12	t
13	y	4	t

$$\gamma_{\{y, t; y, t, \text{avg}(z)\}} (R) =$$

y	t	Avg(z)
a	x	9
a	y	3
a	z	9
b	x	1
b	y	12
c	x	4
d	t	5
d	u	5
x	t	12
y	t	4

# Algèbre relationnelle - Sous groupes

$\gamma_{\{ \text{liste d'attributs pour le groupement} ; \text{liste d'expressions affichées} \}} (Relation)$

R :

x	y	z	t
1	a	12	x
2	a	12	x
3	a	3	x
4	a	3	y
5	a	12	z
6	a	6	z
7	b	1	x
8	b	12	y
9	c	4	x
10	d	5	t
11	d	5	u
12	x	12	t
13	y	4	t

$\gamma_{\{y;y,t,count(*)\}} (R)$

Erreur de syntaxe

et même de sens !

# Algèbre relationnelle - Sous groupes

$\gamma_{\{ \text{liste d'attributs pour le groupement} ; \text{liste d'expressions affichées} \}} (Relation)$

R :

x	y	z	t
1	a	12	x
2	a	12	x
3	a	3	x
4	a	3	y
5	a	12	z
6	a	6	z
7	b	1	x
8	b	12	y
9	c	4	x
10	d	5	t
11	d	5	u
12	x	12	t
13	y	4	t

$$\gamma_{\{y, t; y, count(*)\}} (R) =$$

y	Count(*)
a	3
a	2
a	1
b	1
c	1
d	1
x	1
y	1

Certaines lignes représentent plusieurs groupes

# Algèbre relationnelle - Agrégat et autres opérations

$\gamma_{\{liste\ d'attributs\ pour\ le\ groupement : liste\ d'expressions\ affichées\}} (Relation)$

R :

x	y	z	t
1	a	12	x
2	a	12	x
3	a	3	x
4	a	3	y
5	a	12	z
6	a	6	z
7	b	1	x
8	b	12	y
9	c	4	x
10	d	5	t
11	d	5	u
12	x	12	t
13	y	4	t

$$\gamma_{\{y; y, count(*)\}} (\sigma_{\{z=12\}}(R)) =$$

y	Count(*)
a	3
b	1
x	1

# Algèbre relationnelle - Exercices

Etudiant					
EtuNo	EtuNom	EtuPnom	EtuSec	EtuAn	etuTel
32345	Dupont	Marc	R	2	null
33568	Durant	Pierre	R	1	047857456
38514	Dupont	François	G	1	null

Section	
SecId	SecLibelle
I	Industrielle
R	Réseaux
G	Gestion

Donnez une expression relationnelle fournissant

- 1 par id et libelle de section, le nombre d'étudiants régulièrement inscrits
- 2 par id et libelle de section, le nombre d'étudiants dont le nom est 'Dupont'
- 3 les id et libellés de sections dans lesquelles au moins 3 étudiants ont le même nom.

- 1 Principe
- 2 Les fonctions agrégatives
- 3 La clause GROUP BY
- 4 Algèbre relationnelle
- 5 SQL - Regroupement**
- 6 SQL - Sélection de groupes
- 7 SQL - Regroupement - Exemples

# SQL - Ordre de lecture

- 6 : **select** liste d'expressions
- 1 : **from** table(s) et jointures
- 2 : **where** conditions sur les lignes
- 3 : **group by** liste d'expressions pour le regroupement



## SQL - Regroupement

```
SELECT localite, COUNT(*), AVG(compte)  
FROM client  
GROUP BY localite ;
```

```
SELECT localite, cat, COUNT(*), AVG(compte)  
FROM client  
GROUP BY localite, cat ;
```

```
SELECT ncli, COUNT(*)  
FROM client  
GROUP BY ncli ;
```

Cette dernière requête n'a pas beaucoup de sens !

- 1 Principe
- 2 Les fonctions agrégatives
- 3 La clause GROUP BY
- 4 Algèbre relationnelle
- 5 SQL - Regroupement
- 6 SQL - Sélection de groupes**
- 7 SQL - Regroupement - Exemples

# SQL - Ordre de lecture

- 6 : **select** liste d'expressions
- 1 : **from** table(s) et jointures
- 2 : **where** conditions sur les lignes
- 3 : **group by** liste d'expressions pour le regroupement
- 4 : **having** conditions sur les groupes

## SQL - Sélection de groupes

```
SELECT localite, COUNT(*), AVG(compte)
FROM client
GROUP BY localite
HAVING COUNT(*)  $\geq$  3 ;
```

LOCALITE	count(*)	avg(COMPTE)
Namur	4	-2520.00
Poitiers	3	533.33
Toulouse	5	-2530.00

```
SELECT ncli, COUNT(*)
FROM commande
GROUP BY ncli
HAVING COUNT(*)  $\geq$  2 ;
```

- 1 Principe
- 2 Les fonctions agrégatives
- 3 La clause GROUP BY
- 4 Algèbre relationnelle
- 5 SQL - Regroupement
- 6 SQL - Sélection de groupes
- 7 SQL - Regroupement - Exemples**

# SQL - WHERE != HAVING

```
SELECT ncli, COUNT(*)  
  FROM commande  
 WHERE ncom > 30185  
 GROUP BY ncli  
 HAVING COUNT(*) >= 2 ;
```

**WHERE** : Sélection des lignes

**HAVING** : Sélection de groupes

```
SELECT ncli, COUNT(*)  
  FROM commande  
 GROUP BY ncli  
 HAVING ncli > 'C400' ;
```

Cette dernière requête n'est pas optimisée

Toujours privilégier le WHERE si possible !

## SQL - Regroupement - Exemples avec jointures

```
SELECT m.ncli, COUNT(*), SUM(qcom)
FROM commande m
JOIN detail d ON m.ncom = d.ncom
WHERE npro = ' PA45'
GROUP BY m.ncli
HAVING COUNT(*) >= 2 ;
```

```
SELECT 'Montant dû par ', c.ncli, ' = ', SUM(qcom * prix)
FROM client c
JOIN commande m ON m.ncli = c.ncli
JOIN detail d ON m.ncom = d.ncom
JOIN produit p ON p.npro = d.npro
WHERE localite = ' Poitiers'
GROUP BY c.ncli ;
```

## SQL - Regroupement - Exemple d'erreur courante

```
SELECT p.npro, qstock - SUM(d.qcom) as solde  
FROM detail d  
      JOIN produit p ON p.npro = d.npro  
GROUP BY p.npro ;
```

Erreur !

On ne peut afficher que les éléments communs aux groupes

```
SELECT p.npro, qstock - SUM(d.qcom) as solde  
FROM detail d  
      JOIN produit p ON p.npro = d.npro  
GROUP BY p.npro, qstock ;
```

Solution - regrouper aussi sur QSTOCK



## SQL - Regroupement - Exemples

```
SELECT localite, p.npro, SUM(qcom * prix) as montant
FROM client c
    JOIN commande m ON m.ncli = c.ncli
    JOIN detail d ON m.ncom = d.ncom
    JOIN produit p ON p.npro = d.npro
GROUP BY localite, p.npro ;
```

```
SELECT  SUBSTR (cat,1,1) as CAT, COUNT(*) as N
FROM client
GROUP BY  SUBSTR (cat,1,1) ;
```

CAT	N
null	2
B	8
C	6

# SQL - Regroupement - Exemples

```

SELECT 'de ', FLOOR (compte/1000) *1000 as Min,
        ' à ', FLOOR (compte/1000) *1000 + 999 as Max,
        COUNT(*) as N
FROM client
GROUP BY FLOOR (compte/1000)
ORDER BY Min ;

```

de	Min	à	Max	N
de	-9000	à	-8001	1
de	-5000	à	-4001	1
de	-4000	à	-3001	1
de	-3000	à	-2001	2
de	-2000	à	-1001	1
de	0	à	999	9
de	1000	à	1999	1

## SQL - Limite de SQL2

Donnez pour chaque localité, la somme des comptes clients ainsi que le nombre de leurs commandes.

```
SELECT localite, SUM(compte), COUNT(*)
FROM client c
JOIN commande m ON c.ncli = m.ncli
GROUP BY localite ;
```

Erreur : La requête ne répond pas à la demande !

LOCALITE	sum(COMPTE)	count(*)
Lille	720.00	1
Namur	-4580.00	1
Poitiers	1050.00	3
Toulouse	-8700.00	2

FAUX (=350)

**Impossible en SQL2**

# SQL - Regroupement - Exercices

CLIENT					
NCLI	NOM	ADRESSE	LOCALITE	(CAT)	COMPTE
B032	GOFFIN	72, r. de la Gare	Namur	B2	-3200
B112	HANSENNE	23, r. Dumont	Poitiers	C1	1250
B332	MONTI	112, r. Neuve	Genève	B2	0
B512	GILLET	14, r. de l'Eté	Toulouse	B1	-8700
C003	AVIRON	8, r. de la Cure	Toulouse	B1	-1700
C123	MERCIER	25, r. Lemaitre	Namur	C1	-2300
C400	FERARD	65, r. du Tertre	Poitiers	B2	350
D063	MERCIER	201, bd Nord	Toulouse		-2250
F010	TOUSSAINT	5, r. Godefroid	Poitiers	C1	0
F011	PONCELET	17, Clos des Erables	Toulouse	B2	0
F400	JACOB	78, ch. du Moulin	Bruxelles	C2	0
K111	VANBIST	180, r. Floirmont	Lille	B1	720
K729	NEUMAN	40, r. Bransart	Toulouse		0
L422	FRANCK	60, r. de Wépion	Namur	C1	0
S127	VANDERKA	3, av. des Roses	Namur	C1	-4580
S712	GUILAUME	14a, ch. des Roses	Paris	B1	0

PRODUIT			
NPRO	LIBELLE	PRIX	QSTOCK
CS262	CHEV. SAPIN 200x6x2	75	45
CS364	CHEV. SAPIN 200x6x4	120	2690
CS464	CHEV. SAPIN 400x6x4	220	450
PA45	POINTE ACIER 45 (2K)	105	580
PA60	POINTE ACIER 60 (1K)	95	134
PH222	PL. HETRE 200x20x2	230	782
PS222	PL. SAPIN 200x20x2	195	1220

COMMANDE		
NCOM	NCLI	DATECOM
30178	K111	21/12/2008
30179	C400	22/12/2008
30182	S127	23/12/2008
30184	C400	23/12/2008
30195	F011	20/1/2009
30186	C400	20/1/2009
30188	B512	30/1/2009

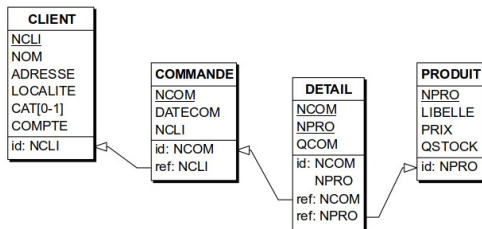
DETAIL		
NCOM	NPRO	QCOM
30178	CS464	25
30179	CS262	60
30179	PA60	20
30182	PA60	30
30184	CS464	120
30184	PA45	20
30185	CS464	260
30185	PA60	15
30185	PS222	600
30186	PA45	3
30188	CS464	180
30188	PA45	22
30188	PA60	70
30188	PH222	92

Donnez la sémantique de la requête suivante :

```

SELECT MIN(datecom), nom
FROM client c
      JOIN commande m ON c.ncli = m.ncli
GROUP BY c.ncli, nom ;
  
```

# SQL - Regroupement - Exercices



Donnez la sémantique de la requête suivante :

```

SELECT commande.ncom, client.ncli, nom
FROM commande
JOIN client ON client.ncli = commande.ncli
JOIN detail ON detail.ncom = commande.ncom
GROUP BY commande.ncom, client.ncli, nom
HAVING MAX(qcom) >= 100 ;
  
```

# SQL - Ordre de lecture

- 6 : **select** liste d'expressions
- 1 : **from** table(s) et jointures
- 2 : **where** conditions sur les lignes
- 3 : **group by** liste d'expressions pour le regroupement
- 4 : **having** conditions sur les groupes
- 5 : **order by** liste d'expressions pour le tri du résultat

## SQL - Ordre des lignes

```
SELECT ncli, nom, localite  
  FROM client  
  WHERE cat IN ('C1','C2')  
  ORDER BY localite ;
```

```
SELECT *  
  FROM client  
  ORDER BY localite, cat ;
```

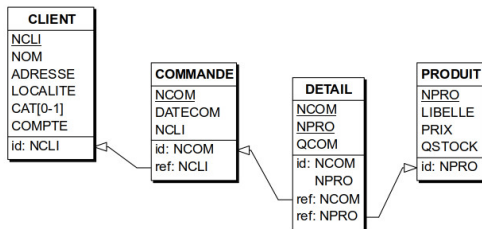
## SQL - Ordre des lignes

```
SELECT localite, COUNT(*) as population, SUM(compte)  
  FROM client  
  GROUP BY localite  
  ORDER BY population desc ;
```

```
SELECT ncom, npro, qcom  
  FROM detail d  
    JOIN produit p ON d.npro = p.npro  
  ORDER BY ncom, qcom*prix desc ;
```



# SQL - Ordre des lignes- Exercices



Donnez la sémantique de la requête suivante :

```

SELECT nom, commande.ncom, datecom, detail.npro, libelle, qcom*prix
FROM detail
JOIN commande ON detail.ncom = commande.ncom
JOIN client ON client.ncli = commande.ncli
JOIN produit ON produit.npro = detail.npro
ORDER BY commande.ncom, libelle ;
  
```