

# Introduction aux Bases de données

## *Cours 7 : Requêtes d'agrégation en SQL*

UFR 919 – Licence  
2<sup>e</sup> année

# SQL : Fonctions d'agrégation

Jusqu'à présent les requêtes retournent des n-uplets (multi-ensemble)

Les fonctions d'agrégation permettent de synthétiser le contenu des données en terme de statistiques descriptives :

- COUNT(att) ou COUNT(\*) : nombre de valeurs ou n-uplets,
- SUM(att) : somme des valeurs,
- MAX(att) : valeur maximale,
- MIN(att) : valeur minimale,
- AVG(att) : moyenne des valeurs

Aggrégation du multi-ensemble désigné par att

**Appliquées aux valeurs NON NULLES seulement!**

# Exemple

PNO	Pname	Budget	City
01	Dev web	100k	Paris
02	Dev web	200k	Lyon
03	Rech info	120k	Paris
04	Maintenance	100k	Pau
05	Assurances	80k	Lyon

**Project**

count(\*)

count(city)

**count**

5

count(distinct city)

**count**

3

# SQL : requêtes d'agrégation

On applique des fonctions de calcul sur le résultat d'une requête :

**SELECT** Agg1 ( $A_i$ ) , ... , AggN ( $A_j$ )

**FROM**  $R_1$  , ... ,  $R_m$

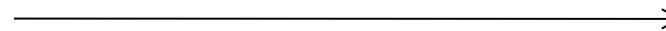
**WHERE** *Conditions*

- On exécute la requête (clause from et where).
- On applique les différentes fonctions Aggk sur l'ensemble du résultat.

PNO	Pname	Budget	City
01	Dev web	100k	Paris
02	Dev web	200k	Lyon
03	Rech info	120k	Paris
04	Maintenance	100k	Pau
05	Assurances	80k	Lyon

**Project**

**Select** city, count(\*) , sum(budget)  
**From** Project  
**Where** city='Paris';



City	count	sum
Paris	2	220k

# Exemples d'agrégation

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Nombre d'employés parisiens ?

Plus grand salaire et plus petit salaire de toutes les professions?

# Exemples d'agrégation

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Total des budgets des projets de Paris?

Nombre de villes où il y a un projet avec l'employé E4?

# Requêtes mal-formées

## Agrégation dans le WHERE

```
SELECT TITLE  
FROM Pay  
WHERE Salary=MAX(Salary) ;  
Problème?
```

## Agrégation dans le SELECT

```
SELECT TITLE,MAX(Salary)  
FROM Pay;  
Problème?
```

Comment faire pour connaître alors la profession payant le plus ?

# Fonctions d'agrégation et imbrication

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Pour utiliser des attributs de tables du FROM et une agrégation, il faut souvent passer par une sous-requête :

Noms des professions qui payent le plus (et les salaires correspondant) ?

```
SELECT Title, Salary
FROM Pay
WHERE Salary = (SELECT MAX(Salary)
                  FROM Pay) ;
```



# Fonctions d'agrégation et imbrication

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Noms des projets dont le budget est supérieur au budget moyen?

← Singleton  
absence de any/all

Remarque : dans le cas où la requête imbriquée retourne le résultat d'un agrégat (donc un seul résultat), pas besoin d'utiliser un IN, ANY ou ALL après l'opérateur =, <, >, <=, >=

# Agrégations et partitions

- ◆ Par défaut, agrégation de toutes les valeurs
- ◆ Possibilité d'appliquer les fonctions d'agrégation sur des **groupes de valeurs** → **partitionnement**

PNO	Pname	Budget	City
01	Dev web	100k	Paris
02	Dev web	200k	Lyon
03	Rech info	120k	Paris
04	Maintenance	100k	Pau
05	Assurances	80k	Lyon

**Project**

Grouper  
par city  
→

PNO	Pname	Budget	City
01	Dev web	100k	Paris
03	Rech info	120k	
02	Dev web	200k	Lyon
05	Assurances	80k	
04	Maintenance	100k	Pau

# Requêtes de groupement : GROUP BY

Pour *partitionner* les n-uplets résultats en fonction des valeurs de certains attributs :

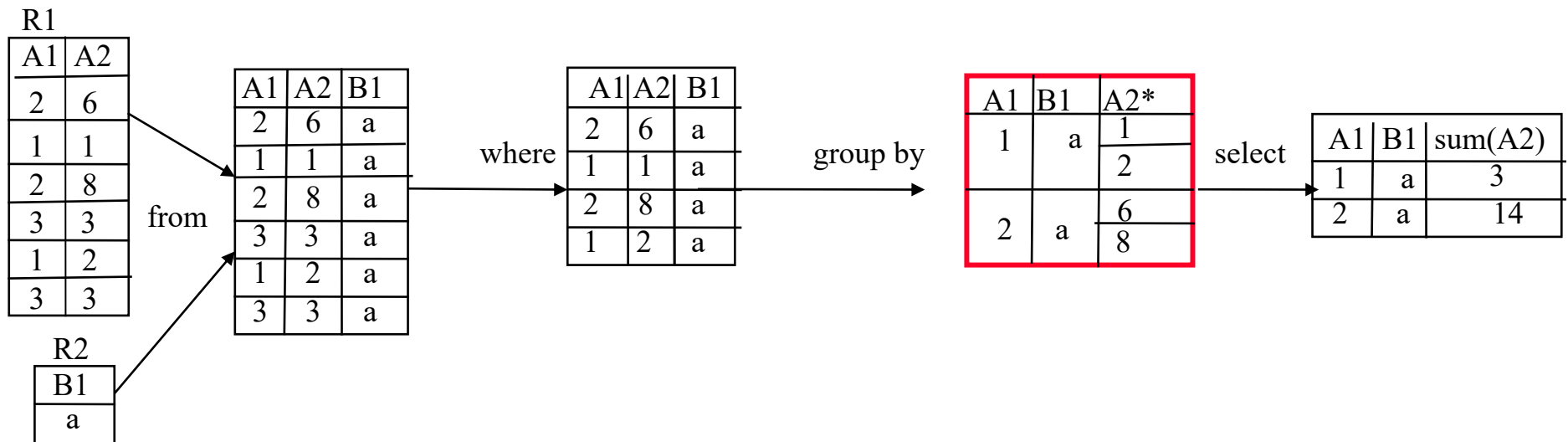
```
SELECT  $A_i$ , ...,  $A_n$ , agg1, agg2, ...  
FROM  $R_1$ , ...,  $R_m$   
WHERE Cond  
GROUP BY  $A_j$  ...,  $A_k$  ;
```

Principes :

- On évalue la requête FROM-WHERE
- On regroupe le résultat en paquets d'enregistrements en plaçant dans un paquet tous les enregistrements ayant la même valeur pour  $A_j$  ...,  $A_k$
- On fait le select en appliquant les agrégations à chaque paquet

# GROUP BY

**SELECT**  $A1, B1, \text{sum}(A2)$   
**FROM**  $R_1, R_2$   
**WHERE**  $A1 < 3$   
**GROUP BY**  $A1, B1$  ;



# GROUP BY : problèmes courants

```
SELECT  A1, B1, sum(A2)
FROM    R1, R2
WHERE   A1 < 3
GROUP BY A1, B1 ;
```

Group by

A1	B1	A2*
1	a	1
		2
2	a	6
		8

A1	B1	sum(A2)
1	a	3
2	a	14

les attributs projetés doivent être dans le group by

R1

A1	B1	A2
1	-2	1
	-3	2
2	1	6
	0	8

```
SELECT  A1, B1, sum(A2)
FROM    R1, R2
WHERE   A1 < 3
GROUP BY A1 ;
```

Résultat R1

pas de sens

Sauf si Agg(atti) et pas de partitionnement sur atti

R2

```
SELECT  A1, min(B1), sum(A2)
FROM    R1, R2
WHERE   A1 < 3
GROUP BY A1 ;
```

Résultat R2

A1	min	sum
1	-3	3
2	0	14

Group by

# Exemples de groupement

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Numéros des projets avec leurs effectifs ?

Pour chaque ville, nombre d'employés par profession ?

# Prédicats sur des groupes

Pour *garder (éliminer) les groupes (partitions)* qui satisfont (ne satisfont) pas une certaine condition :

```
SELECT  $A_i, \dots, A_n, agg1, agg2, \dots$   
FROM  $R_1, \dots, R_m$   
WHERE  $P$   
GROUP BY  $A_j \dots, A_k$   
HAVING  $Q$  ;
```

Principe : une fois la requête FROM-WHERE exécutée, on regroupe en agrégat avec le GROUP BY et on ne garde ensuite que les agrégats satisfaisants le HAVING

# Remarques

Ordre d'exécution des clauses :

5 SELECT

1 FROM

2 WHERE

3 GROUP BY

4 HAVING

6 ORDER BY

Conséquence : à partir de 3 on manipule des agrégats donc pour 4, 5 et 6 on ne manipule que *des fonctions d'agrégations ou des attributs apparaissant dans le GROUP BY*



# Exemples de groupement (1)

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Villes dans lesquelles habitent plus de 2 employés?

Projets demandant plus de 1000 jours/homme?

# Exemples de groupement (2)

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Numéros et noms des projets ayant des employés venant de plus de 5 villes différentes ?

Liste des employés triés par temps total de travail décroissants ?

# Imbrication de fonctions d'agrégation

- On peut imbriquer dans le SELECT deux fonctions d'agrégation :

agg1(agg2 [DISTINCT] att)

Où agg1 : MIN, MAX ou AVG

- Il faut que l'agrégat interne soit associé à un GROUP BY
- L'agrégat externe agg1 s'applique au résultat que retournerait la requête avec uniquement agg2 et le GROUP BY.
- Agg1 n'est donc pas associé au GROUP BY
- Attention : pas implanté dans tous les systèmes !

# Exemples de groupement (3)

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Effectif maximal des projets

Nombre moyen d'employés provenant de chaque ville ?

# Exemples de groupement (4)

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Comment connaître le nom du (des) projet(s) avec le plus de travailleurs?

```
SELECT Pno, MAX (COUNT (*))  
FROM Works  
GROUP BY Pno ;
```

=> INTERDIT ! Le GROUP BY sert pour le COUNT, pas de GROUP BY pour le MAX, donc pas Pno dans le SELECT

Solution ? Utiliser une requête imbriquée !

# Simuler l'imbrication d'agrégations sous H2

- En SQL : agg1(agg2 [DISTINCT] att)
- Impossible sous H2 car agrégation à un seul niveau
- Contourner au moyen de l'imbrication dans le FROM
  - Y placer une requête qui évalue l'agrégation *interne* agg2
  - Appliquer l'agrégation *externe* agg1 sur le résultat de la requête imbriquée dans le FROM

# Exemples de groupement sous H2

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Effectif maximal des projets

Nombre moyen d'employés provenant de chaque ville ?

# Division à l'aide de l'agrégation

**Emp** (Eno, Ename, Title, City)

**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)

**Works**(Eno, Pno, Resp, Dur)

On souhaite connaître les employés ayant travaillé sur tous les projets

Rappel : on calcule les

*employés pour lesquels il n'existe pas de projets auxquels ils n'ont pas participé*

Soit une **double-négation** !



# Exemples de division (1)

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Quels sont les employés ayant travaillé sur tous les projets?

```
SELECT Eno, Ename
FROM Emp
WHERE NOT EXISTS (SELECT *
                   FROM Project
                   WHERE NOT EXISTS (SELECT *
                                     FROM Work
                                     WHERE Work.Eno=Emp.Eno
                                     AND Work.Pno=Project.Pno) ) ;
```

Bien observer que la (les) table(s) faisant le lien entre les 2 autres tables se trouve(nt) dans la 2ème imbrication et réalise(nt) les 2 jointures

# Exemples de division (2)

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Quels sont les villes dans lesquelles tous les salaires d'employés sont représentés?

```
SELECT DISTINCT City
FROM Emp A
WHERE NOT EXISTS (SELECT *
                  FROM Pay P1
                  WHERE NOT EXISTS (SELECT *
                                    FROM Emp B, Pay P2
                                    WHERE B.Title=P2.Title
                                    AND P2.Salary=P1.Salary
                                    AND B.City=A.City) ) ;
```

# Exemples de division (3)

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Quels sont les projets parisiens pour lesquels toutes les professions gagnant plus de 3000 euros ont participé?

```
SELECT Pno, Pname
FROM   Projet
WHERE  City='Paris'
AND NOT EXISTS (SELECT *
                 FROM   Pay
                 WHERE  Salary>3000
                 AND NOT EXISTS (SELECT *
                                FROM   Work, Emp
                                WHERE  Work.Eno=Emp.Eno
                                AND    Work.Pno=Project.Pno
                                AND    Emp.Title=Pay.Title) ) ;
```

# Division : autre solution

Une autre solution pour faire la division repose sur les agrégats

On procède en 3 étapes :

1. On compte le nombre d'enregistrements correspondant à l'ensemble de référence
2. On compte pour chaque candidat potentiel à combien d'éléments de l'ensemble de référence il peut être associé
3. Si c'est égal au total, on retourne ce candidat

# Exemples de division avec agrégats(1)

**Emp** (Eno, Ename, Title, City)      **Project**(Pno, Pname, Budget, City)  
**Pay**(Title, Salary)                      **Works**(Eno, Pno, Resp, Dur)

Quels sont les employés ayant travaillé sur tous les projets?

```
SELECT Eno, Ename
FROM Emp
WHERE NOT EXISTS (SELECT *
                   FROM Project
                   WHERE NOT EXISTS (SELECT *
                                     FROM Work
                                     WHERE Work.Eno=Emp.Eno
                                     AND Work.Pno=Project.Pno) ) ;
```

```
SELECT Eno, Ename
FROM Emp, Work
WHERE Emp.Eno=Work.Eno
GROUP BY Eno, Ename
HAVING COUNT(DISTINCT Pno)=(SELECT COUNT(*)
                             FROM Project);
```

**utilisation indispensable du GROUP BY dans ce type de division**

# Exemples de division avec agrégats(2)

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Quels sont les villes dans lesquelles tous les salaires d'employés sont représentés?

```
SELECT DISTINCT City
FROM Emp A
WHERE NOT EXISTS (SELECT *
                  FROM Pay P1
                  WHERE NOT EXISTS (SELECT *
                                    FROM Emp B, Pay P2
                                    WHERE B.Title=P2.Title
                                    AND P2.Salary=P1.Salary
                                    AND B.City=A.City) ) ;
```

# Exemples de division avec agrégats(3)

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Quels sont les villes dans lesquelles tous les salaires d'employés sont représentés?

# Exemples de division avec agrégats(4)

**Emp** (Eno, Ename, Title, City)  
**Pay**(Title, Salary)

**Project**(Pno, Pname, Budget, City)  
**Works**(Eno, Pno, Resp, Dur)

Quels sont les projets parisiens pour lesquels toutes les professions gagnant plus de 3000 euros ont participé?