

Les regroupements (GROUP BY)

Permet de supprimer les doublons dans les lignes ramenées et de réaliser des calculs sur les colonnes sélectionnées.

Sur une colonne sans faire de calcul, le GROUP BY remplace le DISTINCT.

```
SELECT DISTINCT Prix FROM Tarifs;
```

est équivalent à :

```
SELECT Prix FROM Tarifs GROUP BY Prix;
```

Fonctions	Description
AVG (attribut)	Calcule la moyenne des valeurs dans l'attribut.
SUM (attribut)	Calcule la somme des valeurs dans l'attribut.
MIN (attribut)	Détermine la plus petite valeur dans l'attribut.
MAX (attribut)	Détermine la plus grande valeur dans l'attribut.
COUNT (attribut)	Compte le nombre d'occurrences dans l'attribut.
VARIANCE(attribut)	Variance de l'attribut
STDDEV(attribut)	Ecart type de l'attribut

- calculer la somme des prix par type ?
 - trouver le tarif le plus cher ?
- trouver le tarif le plus cher et le moins cher par type de chambre ?
 - le prix moyen des chambres par type de chambre ?

```
SELECT typeChambre, SUM(Prix) AS  
PRIX_CUMULE  
FROM Tarifs  
GROUP BY typeChambre;
```

```
SELECT MAX(Prix) AS PRIX_MAX  
FROM Tarifs;
```

```
SELECT TypeChambre, Description, MIN(Prix)  
AS PRIX_MIN, MAX(Prix)  
AS PRIX_MAX, COUNT(1) AS NbTarifs  
FROM Tarifs INNER JOIN TypesChambre ON  
TypesChambre.idTypeChambre =  
Tarifs.typeChambre  
GROUP BY TypeChambre, Description;
```

```
SELECT TypeChambre, Description,  
MIN(Prix) AS PRIX_MIN,  
MAX(Prix) AS PRIX_MAX,  
AVG(prix) AS PrixMoyen  
FROM Tarifs INNER JOIN TypesChambre ON  
TypesChambre.idTypeChambre  
= Tarifs.typeChambre  
GROUP BY TypeChambre, Description;
```

Sans le inner join cela donne quoi ?

La clause de restriction sur un regroupement (HAVING)

C'est un filtre supplémentaire qui travaille sur le résultat d'un GROUP BY.

À l'instar du WHERE qui filtre les lignes ramenées par le SELECT, le HAVING va filtrer le résultat des fonctions utilisées.

Moyenne des prix par type chambres uniquement
sur les prix de chambres de plus de 70 € :

```
SELECT TypeChambre, Description, MIN(Prix)  
AS PRIX_MIN, MAX(Prix)  
AS PRIX_MAX, AVG(prix) AS PrixMoyen  
FROM Tarifs INNER JOIN TypesChambre ON  
TypesChambre.idTypeChambre =  
Tarifs.typeChambre  
GROUP BY TypeChambre, Description  
HAVING MIN(Prix) > 70;
```


Sélection des types de chambres et de leur description qui sont présentes moins de 4 fois.

```
SELECT typechambre, description,  
COUNT(IdChambre) AS NbChambres  
FROM Chambres CH  
INNER JOIN TypesChambre TYP  
HAVING COUNT(IdChambre) < 4;
```

A vous de jouer, si vous acceptez cette mission
voici la base de données (vous la connaissez
déjà) :

Réalisateur(num_realisateur, nom, prenom, *nationalite*)

Film(ident_film, titre, genre, recette, date_sortie,
pays, nb_entree, date_saisie, resume,
num_real, *distributeur*)

Acteur(ident_acteur, nom, prenom, date_naissance,
nationalite)

Casting(ident_film, ident_acteur, *role*
nb_jour_tournage)

Statistique(budget, ident_film, nb_entree_france, n
b_entree, recette_usa, recette_monde

Pays(ident_pays, libelle)