

# **Bloc -1**

# **BDD**

**Le langage S.Q.L :  
Interrogation de la  
base**

# Les différents opérateurs de l'algèbre relationnelle

Opérateurs

**Relationnels**

**Sélection**

**Sélection**

**Jointure**

Opérateurs

**Ensemblistes**

**Union**

**Intersection**

**Difference**

# Opérateurs relationnels

## La projection

C'est une opération qui consiste à afficher un certain nombre de colonnes.

NUMECLIE	NOMCLIE	ADRECLIE	CODPCLIE	VILLCLIE	TELECLIE	NUMEREPR
100001	Digit Equipement	25, rue des Pins	06200	Valbonne	0493564578	2
100002	F.D.T	10, Impasse de la Poste	06000	Nice	0493564545	2
100010	Ecole Pigier	10, av J Médecin	06200	Nice	0493656562	2



NUMECLIE	NOMCLIE	VILLCLIE
100001	Digit Equipement	Valbonne
100002	F.D.T	Nice
100010	Ecole Pigier	Nice

Pour obtenir le numéro, le nom et la ville des clients, on projette ces 3 colonnes (champs) à partir de la table CLIENTS

# Opérateurs relationnels

## La sélection

C'est une opération qui permet d'obtenir toutes les occurrences répondant à un ou plusieurs critères..

NUMCLIE	NOMCLIE	ADRECLIE	CODPCLIE	VILLCLIE	TELECLIE	NUMEREPR
100001	Digit Equipement	25, rue des Pins	06200	Valbonne	0493564578	2
100002	F.D.T	10, Impasse de la Poste	06000	Nice	0493564545	2
100010	Ecole Pigier	10, av J Médecin	06200	Nice	0493656562	2
100011	Forum Immobilier	5, Av de Grasse	06500	Valbonne	0493252525	2
100020	Habitable	2, av de Valbonne	06200	Nice	0493656566	2
200005	Lycée Victor Hugo	Bd Desplaces	13001	Marseille	0491525269	3
300010	A.R.T	20, rue de la mer	83000	Toulon	0494562154	1



NUMCLIE	NOMCLIE	ADRECLIE	CODPCLIE	VILLCLIE	TELECLIE	NUMEREPR
100002	F.D.T	10, Impasse de la Poste	06000	Nice	0493564545	2
100010	Ecole Pigier	10, av J Médecin	06200	Nice	0493656562	2
100020	Habitable	2, av de Valbonne	06200	Nice	0493656566	2

Si l'on veut visualiser les clients habitant Nice, on sélectionne dans la table CLIENT tous ceux dont le champ Villclie = « Nice »

# Opérateurs relationnels

## La jointure

C'est une opération permettant de mettre en relation deux ou plusieurs tables grâce à leurs colonnes communes correspondant respectivement aux clé primaire et clé étrangère)  
Une jointure est nécessaire lorsque les données recherchées se trouvent dans plusieurs tables.

NUMCLIE	NOMCLIE	ADRECLIE	CODPCLIE	VILLCLIE	TELECLIE	NUMEREPR
100001	Digit Equipement	25, rue des Pins	06200	Valbonne	0493564578	2
100002	F.D.T	10, Impasse de la Poste	06000	Nice	0493564545	2
100010	Ecole Pigier	10, av J Médecin	06200	Nice	0493656562	2
100011	Forum Immobilier	5, Av de Grasse	06500	Valbonne	0493252525	2
100020	Habitable	2, av de Valbonne	06200	Nice	0493565656	2
200005	Lycée Victor Hugo	Bd Desplaces	13001	Marseille	0491525269	3
300010	A.R.T	20, rue de la mer	83000	Toulon	0494562154	1

NUMEREPR	NOMRREPR	PRENREPR	ADREREPR	CODPREPR	VILLREPR
1	Durant	Patrick	56, Av du Lycée	83000	Fréjus
2	Dupont	Eric	45, Av du Collège	06000	Nice
3	Richard	Bernard	12, Av de l'Ecole	13001	Marseille



NUMCLIE	NOMCLIE	ADRECLIE	CODPCLIE	VILLCLIE	TELECLIE	NUMEREPR	NOMRREPR	PRENREPR	ADREREPR	CODPREPR	VILLREPR
100001	Digit Equipement	25, rue des Pins	06200	Valbonne	0493564578	2	Dupont	Eric	45, Av du Collège	06000	Nice
100002	F.D.T	10, Impasse de la Poste	06000	Nice	0493564545	2	Dupont	Eric	45, Av du Collège	06000	Nice
100010	Ecole Pigier	10, av J Médecin	06200	Nice	0493656562	2	Dupont	Eric	45, Av du Collège	06000	Nice
100011	Forum Immobilier	5, Av de Grasse	06500	Valbonne	0493252525	2	Dupont	Eric	45, Av du Collège	06000	Nice
100020	Habitable	2, av de Valbonne	06200	Nice	0493565656	2	Dupont	Eric	45, Av du Collège	06000	Nice
200005	Lycée Victor Hugo	Bd Desplaces	13001	Marseille	0491525269	3	Richard	Bernard	12, Av de l'Ecole	13001	Marseille
300010	A.R.T	20, rue de la mer	83000	Toulon	0494562154	1	Durant	Patrick	56, Av du Lycée	83000	Fréjus

Si l'on veut visualiser pour chaque client les coordonnées des représentants auxquels ils sont rattachés, il faut joindre les tables CLIENT et REPRESENTANT à l'aide de leur champ commun : NumRepr.

# Opérateurs ensemblistes

Les opérateurs ensemblistes ne peuvent s'appliquer que sur des tables à structure identique.

## L'Union

L'union permet de faire la somme de deux ou plusieurs relations.



# Opérateurs ensemblistes

Les opérateurs ensemblistes ne peuvent s'appliquer que sur des tables à structure identique.

## L' Intersection

L'intersection permet de déterminer les occurrences communes à deux relations.

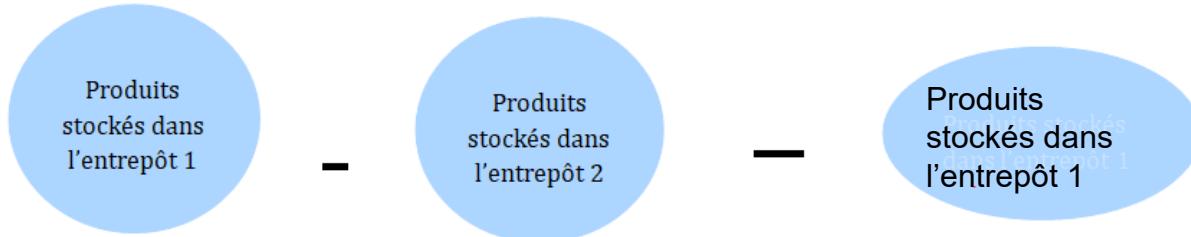


# Opérateurs ensemblistes

Les opérateurs ensemblistes ne peuvent s'appliquer que sur des tables à structure identique.

## La Différence

Faire la différence entre une relation R1 et une relation R2 consiste à prélever de la relation R1 les occurrences identiques à celles de R2 afin de ne conserver que les occurrences spécifiques à R1.



# Le processus de construction d'une requête d'interrogation.

1. Observer le schéma relationnel et repérer les liens éventuels entre les tables relationnelles (clé étrangère et clé primaire).
2. Analyser la requête et déterminer les opérations algébriques à réaliser (sélection, projection, jointure)

ATTENTION : Si  $N$  tables sont nécessaires pour réaliser la requête, il y aura  $N-1$  jointures. Pensez à minimiser le nombre de tables dans la réalisation de la requête.

3. Ecrire la requête dans le langage demandé.
4. Exécuter et vérifier la cohérence du résultat obtenu

# Le SGBDR MySQL

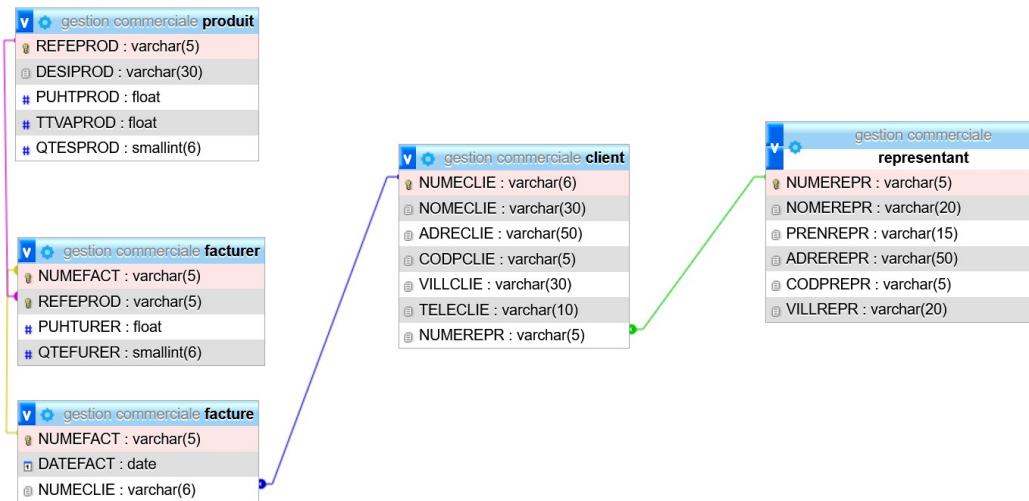


Figure 1 : Schéma relationnel sous MySQL

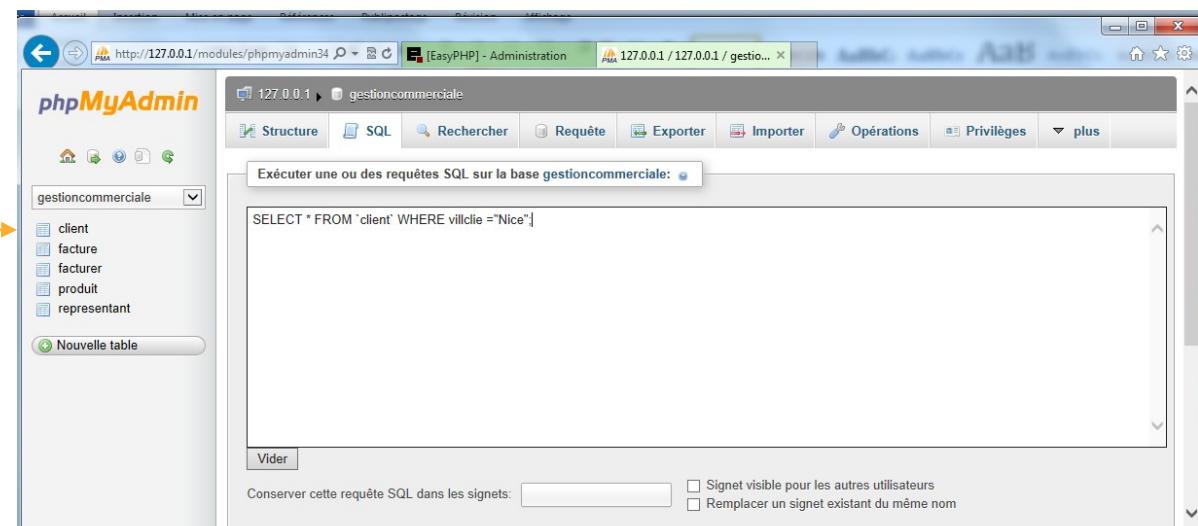


Figure 2 : Requête dans MySQL

# Le langage S.Q.L

## Comment questionner la base de données ?

Exemple : *Liste des clients (numéro, nom, ville) domiciliés à Nice et nom de leur représentant,*

Quelles sont des données à afficher (projection) ?	numéro, nom, ville du client et nom du representant
Dans quelle(s) table(s) se trouvent ces données ?	Client et representant
Y-a-t-il des critères de sélection à respecter ?	Ville du client = « Nice »

Une requête SQL se construit de la façon suivante :

**PROJECTION,**

**(JOINTURES),**

**(SELECTION)**

# Le langage S.Q.L

## Comment questionner la base de données ?

La requête s'écrira:

```
SELECT NuméClie, NomeClie, VillClie, NomeRepr
FROM REPRESENTANT INNER JOIN CLIENT ON
REPRESENTANT.NuméRepr = CLIENT.NuméRepr
WHERE VillClie='Nice' ;
```

**Ou**

```
SELECT NuméClie, NomeClie, VillClie, NomeRepr
FROM REPRESENTANT ,CLIENT
WHERE REPRESENTANT.NuméRepr = CLIENT.NuméRepr
AND VillClie='Nice' ;
```

# Le langage S.Q.L

## Structures de base d'une requête d'interrogation

Forme générale d'une requête SQL:

```
SELECT Colonne1 [AS Alias1], [Colonne2 [AS Alias2],.....  
FROM Table1 INNER JOIN Table2 ON Table1.Colonne3 =  
Table2.Colonne3 [INNER JOIN Table3 ON Table3.Colonne4 =  
Table2.Colonne4...]
```

Ou

```
FROM Table1,Table2, Table3  
WHERE Table1.Colonne3 = Table2.Colonne3  
AND Table3.Colonne4 = Table2.Colonne4...
```

```
[WHERE Condition1][AND Condition2 | OR Condition3]
```

```
[ORDER BY Colonne5 [ASC]DESC, Colonne6 [ASC]DESC...];
```

# Le langage S.Q.L

## Opérateurs relationnels

### La projection

***Afficher la liste des clients avec le numéro, le nom et la ville.***

```
SELECT NumeClie, NomeClie, VillClie  
FROM CLIENT
```

NumeClie	NomeClie	VillClie
100001	Digit Equipement	Valbonne
100002	F.D.T	Nice
100010	Ecole Pugier	Nice
100011	Forum Immobilier	Valbonne
100020	Habitable	Nice
200005	Lycée Victor Hugo	Marseille
300010	A.R.T	Toulon

# Le langage S.Q.L

## Opérateurs relationnels

### La sélection

Il existe plusieurs façons d'exprimer une condition dans le cadre d'une sélection (**restriction**)

Avec le « = »

***Afficher le numéro, le nom et le numéro de téléphone des clients de Nice***

```
SELECT NumeClie, NomeClie, TeleClie  
FROM CLIENT  
WHERE VillClie='Nice';
```

NumeClie	NomeClie	TeleClie
100002	F.D.T	0493564545
100010	Ecole Pugier	0493656562
100020	Habitable	0493565656

# Le langage S.Q.L

## Opérateurs relationnels

### La sélection

Il existe plusieurs façons d'exprimer une condition dans le cadre d'une sélection (**restriction**)

Avec "<>, <, >, >=, <=

***Afficher les produits (référence et désignation) dont le prix ht dépasse 150 €***

```
SELECT RefeProd, DesiProd  
FROM PRODUIT  
WHERE PuhtProd >150;
```

RefeProd	DesiProd
20001	Home Jet 1
20002	Home Jet 2
20005	Home Laser 4/600
20010	Home Laser 4/300
30003	Home Power 500
30005	Home Power 1000
30006	Home Power 1100 8/500
30007	Home Power 1000 16/800
30009	Home Power 3000

# Le langage S.Q.L

## Opérateurs relationnels

### La Jointure

Il y a 2 façons de formuler une jointure entre 2 tables.

Dans le **FROM .....****WHERE..... [AND]**

**Afficher les clients (numéro, nom, ville) de Nice avec le numéro et le nom de leur représentant.**

```
SELECT NumeClie, NomeClie, VillClie, REPRESENTANT.NumRepr, NomeRepr
FROM REPRESENTANT, CLIENT
WHERE REPRESENTANT.NumRepr = CLIENT.NumRepr
AND CLIENT.VillClie='Nice';
```

NumeClie	NomeClie	VillClie	NumRepr	NomeRepr
100002	F.D.T	Nice	2	Dupont
100010	Ecole Pugier	Nice	2	Dupont
100020	Habitable	Nice	2	Dupont

# Le langage S.Q.L

## Opérateurs relationnels

### La Jointure

Il y a 2 façons de formuler une jointure entre 2 tables.

Dans le "FROM... ....INNER JOIN.....ON ....."

**Afficher les clients (numéro, nom, ville) de Nice avec le numéro et le nom de leur représentant.**

```
SELECT NumeClie, NomeClie, VillClie, REPRESENTANT.NumeRepr ,NomeRepr
FROM REPRESENTANT INNER JOIN CLIENT ON REPRESENTANT.NumeRepr = CLIENT.NumeRepr
WHERE VillClie='Nice';
```

NumeClie	NomeClie	VillClie	NumeRepr	NomeRepr
100002	F.D.T	Nice	2	Dupont
100010	Ecole Pugier	Nice	2	Dupont
100020	Habitable	Nice	2	Dupont

# Le langage S.Q.L

## Opérateurs ensemblistes

### L'UNION

Avec "UNION"

***Qui sont les clients habitant de Nice ou Valbonne?***

```
SELECT * FROM client WHERE VillClie='Nice'  
UNION  
SELECT * FROM client WHERE VillClie='Valbonne' ;
```

NUMECLIE	NOMECLIE	ADRECLIE	CODPCLIE	VILLCLIE	TELECLIE	NUMEREPR
100002	F.D.T	10, Impasse de la Poste	06000	Nice	0493564545	2
100010	Ecole Pugier	10, av J Médecin	06200	Nice	0493656562	2
100020	Habitable	2, av de Valbonne	06200	Nice	0493565656	2
100001	Digit Equipement	25, rue des Pins	06200	Valbonne	0493564578	2
100011	Forum Immobilier	5, Av de Grasse	06500	Valbonne	0493252525	2

# Le langage S.Q.L

## Opérateurs ensemblistes

### L'UNION

Avec l'opérateur "OR"

***Qui sont les clients habitant de Nice ou Valbonne?***

```
SELECT *
FROM CLIENT
WHERE VilleClie='Nice' OR VilleClie='Valbonne';
```

NUMECLIE	NOMECLIE	ADRECLIE	CODPCLIE	VILLCLIE	TELECLIE	NUMEREPR
100001	Digit Equipement	25, rue des Pins	06200	Valbonne	0493564578	2
100002	F.D.T	10, Impasse de la Poste	06000	Nice	0493564545	2
100010	Ecole Pugier	10, av J Médecin	06200	Nice	0493656562	2
100011	Forum Immobilier	5, Av de Grasse	06500	Valbonne	0493252525	2
100020	Habitable	2, av de Valbonne	06200	Nice	0493565656	2

# Le langage S.Q.L

## Opérateurs ensemblistes

### L'UNION

Avec l'opérateur “IN”

***Qui sont les clients habitant de Nice ou Valbonne?***

```
SELECT *
FROM CLIENT
WHERE VilleClie IN('Nice', 'Valbonne');
```

NUMECLIE	NOMECLIE	ADRECLIE	CODPCLIE	VILLCLIE	TELECLIE	NUMEREPR
100001	Digit Equipement	25, rue des Pins	06200	Valbonne	0493564578	2
100002	F.D.T	10, Impasse de la Poste	06000	Nice	0493564545	2
100010	Ecole Pugier	10, av J Médecin	06200	Nice	0493656562	2
100011	Forum Immobilier	5, Av de Grasse	06500	Valbonne	0493252525	2
100020	Habitable	2, av de Valbonne	06200	Nice	0493565656	2

# Le langage S.Q.L

## Opérateurs ensemblistes

### L'intersection

Avec "INTERSECT"

***Quelles sont les factures éditées en 2018 ?***

```
SELECT * FROM Facture WHERE DateFact>= '01/01/2018'  
INTERSECT  
SELECT * FROM Facture WHERE DateFact<= '31/12/2018'
```

# Le langage S.Q.L

## Opérateurs ensemblistes

### L'intersection

Avec l'opérateur "AND"

***Quelles sont les factures éditées en 2018 ?***

```
SELECT *  
FROM FACTURE  
WHERE DateFact>= '01/01/2018' And DateFact <='31/12/2018';
```

# Le langage S.Q.L

## Opérateurs ensemblistes

### L'intersection

Le “BETWEEN ..... AND.....”.

**Quelles sont les factures éditées en 2018 ?**

```
SELECT *  
FROM FACTURE  
WHERE DateFact Between '01/01/2018' And '31/12/2018';
```

# Le langage S.Q.L

## Opérateurs ensemblistes

### Le Différence

Avec "EXCEPT"

***Qui sont les clients qui n'habitent pas Nice ou Valbonne ?***

```
SELECT * FROM client  
EXCEPT  
SELECT * FROM client WHERE VillClie='Nice' OR VillClie='Valbonne';
```

# Le langage S.Q.L

## Opérateurs ensemblistes

### Le Différence

Avec "NOT IN"

***Qui sont les clients qui n'habitent pas Nice ou Valbonne ?***

```
SELECT *  
FROM CLIENT  
WHERE VilleClie NOT IN ('Nice','Valbonne');
```

# Le langage S.Q.L

## Les spécificités de SQL

### Le surnom de table

Il permet abréger le nom d'une table en une seule lettre et ainsi de raccourcir l'écriture d'une requête.

*Afficher les clients (numéro, nom, ville) de Nice avec le nom de leur représentant.*

```
SELECT C.NumeClie, C.NomClie, C.VillClie, R.NomRepr
FROM REPRESENTANT R, CLIENT C
WHERE R.NumeRepr = C.NumeRepr
AND C.VillClie='Nice';
```

# Le langage S.Q.L

## Les spécificités de SQL

### Surnom ou alias de champs avec "AS"

Il s'agit de renommer certains champs afin de modifier le titre de la colonne lors de l'affichage du résultat de la requête.

*Afficher les clients (numéro, nom, ville) de Nice avec le nom de leur représentant.*

```
SELECT NomClie as Nom, NomRepr as Représentant
FROM REPRESENTANT, CLIENT
WHERE REPRESENTANT.NumeRepr = CLIENT.NumeRepr
AND CLIENT.VillClie='Nice';
```

# Le langage S.Q.L

## Les spécificités de SQL

Surnom ou alias de champs avec "AS"

**Le surnom est souvent utilisé pour nommer un champ calculé.**

*Afficher pour chaque produit, sa désignation, son prix HT, et son prix TTC*

```
SELECT NomProd as Désignation,pxuht as Prix_HT ,(pxuht *(1+txtva)) As Prix_TTC  
FROM PRODUIT ;
```

# Le langage S.Q.L

## Les spécificités de SQL

### Le tri : ORDER BY

Permet d'afficher le résultat d'une requête selon un ordre précis comme par exemple l'ordre alphabétique, du plus grand au plus petit

*Afficher les clients et leur représentant respectif dans l'ordre alphabétique des clients.*

```
SELECT NumeClie, NomClie, VillClie, NomRepr
FROM REPRESENTANT INNER JOIN CLIENT ON REPRESENTANT.NumeRepr =
CLIENT.NumeRepr
WHERE CLIENT.VillClie='Nice'
ORDER BY NomClie;
```

#### Remarque :

Par défaut, un tri s'effectue par ordre croissant. Pour trier dans un ordre décroissant, il faut préciser **DESC** après le champ concerné.

**Exemple :** ORDER BY NomeClie **DESC**

# Le langage S.Q.L

## Les spécificités de SQL

"Like" ou "Not Like".

***Donner le numéro, le nom, la ville des clients domiciliés dans le Var***

```
SELECT NumeClie, NomeClie, VillClie  
FROM CLIENT  
WHERE CodpClieLike '83%';
```

% (ou \* selon les SGBDR) remplace toute chaîne comprenant entre zéro et un nombre quelconque de caractères.

# Le langage S.Q.L

## Les spécificités de SQL

### Le "is Null"

Il indique l'absence de valeur d'une colonne de table autrement dit un champ non renseigné.

*Qui sont les clients qui ne nous ont pas communiqué leur numéro de téléphone ?*

```
SELECT NumeClie, NomeClie  
FROM CLIENT  
WHERE TeleClie is Null;
```

# Le langage S.Q.L

## Les spécificités de SQL

### Distinct

Le résultat de certaines requêtes affiche plusieurs fois les mêmes données sur des lignes différentes.

Pour éviter cet affichage multiple et inutile, il convient de rajouter DISTINCT après le SELECT.

#### *Lister les produits qui ont été facturés*

```
SELECT DISTINCT PRODUIT.RefeProd ,PRODUIT.DesiProd
FROM FACTURER INNER JOIN PRODUIT ON FACTURER.RefeProd = PRODUIT.RefeProd;
```

# Le langage S.Q.L

## Les spécificités de SQL

### Les fonctions de regroupement

Les fonctions de regroupement permettent d'effectuer des calculs portant sur un ou des ensembles de données.

Il existe cinq principales fonctions de regroupements : **MAX**, **MIN**, **COUNT** pour déterminer le nombre, **AVG** pour calculer la moyenne et **SUM** pour calculer une somme.

***Combien y- a- t'il de clients dans la base ?***

```
SELECT COUNT(*)  
FROM CLIENT;
```

# Le langage S.Q.L

## Les spécificités de SQL

### Les fonctions de regroupement

Les fonctions de regroupement permettent d'effectuer des calculs portant sur un ou des ensembles de données.

Il existe cinq principales fonctions de regroupements : **MAX**, **MIN**, **COUNT** pour déterminer le nombre, **AVG** pour calculer la moyenne et **SUM** pour calculer une somme.

***Combien y- a- t'il de clients dans la base ?***

```
SELECT COUNT(*)  
FROM CLIENT;
```

***Quel est le prix moyen d'un produit ?***

```
SELECT AVG(PuhtProd)  
FROM PRODUIT;
```

# Le langage S.Q.L

## Les spécificités de SQL

### Les fonctions de regroupement

#### Le “GROUP BY”

La clause *Group by* permet d'effectuer des calculs, non pas sur toute la table, mais sur des sous-ensembles de table.

***Donner pour chacune des factures, le nombre de lignes de facture.***

```
SELECT Numefact, Count(*)  
FROM FACTURER  
GROUP BY NumeFact;
```

***Donner le nombre de clients habitant dans chacune des différentes villes.***

```
SELECT VillClie, Count(*)  
FROM CLIENT  
GROUP BY VillClie;
```

# **Le langage S.Q.L**

## **Les spécificités de SQL**

### **Les fonctions de regroupement**

#### **Le “GROUP BY”**

Le ou les champs qui apparaissent au niveau du ‘Select’ et qui ne sont pas utilisés dans une fonction de regroupement doivent impérativement se retrouver au niveau du GROUP BY.

# Le langage S.Q.L

## Les spécificités de SQL

### Les fonctions de regroupement

#### Le “HAVING”

La clause *HAVING* agit comme le filtre *WHERE* mais permet de filtrer les résultats calculés dans les opérations de regroupement : pas de **having** sans son **groupby** !

***Donner les numéros de factures qui ont un nombre de lignes supérieur à 2.***

```
SELECT NumeFact, COUNT(*)AS NbreLigneFacture
FROM FACTURER
GROUP BY NumeFact
HAVING COUNT(*)>2;
```

# Le langage S.Q.L

## Les spécificités de SQL

### Forme générale d'une requête

```
SELECT [DISTINCT|ALL] Colonne1 [AS Alias1], [DISTINCT|ALL]  
Colonne2 [AS Alias2,...],[FonctionDeRegroupement(Colonne|*)]  
  
FROM Table1 [Alias1] [, Table2 [Alias2]...]| Table1 [Alias1] INNER  
JOIN Table2 [Alias2] ON Table1.Colonne3 = Table2.Colonne3 [INNER  
JOIN Table3 ON Table3.Colonne4 = Table2.Colonne4...]  
  
[WHERE Condition1][AND Condition2 | OR Condition3]  
  
GROUP BY Colonne1 [, Colonne2]...  
  
[HAVING Condition1][AND Condition2 | OR Condition3]  
  
[ORDER BY Colonne5 [ASC|DESC], Colonne6 [ASC|DESC]...];
```

# Le langage S.Q.L

## Les spécificités de SQL

### Les sous requêtes

Une sous-requête est une requête qui est imbriquée dans autre une requête, ou dans une autre sous-requête. On utilise une sous interrogation à chaque fois qu'un attribut (un champ) de la requête principale doit récupérer sa valeur dans une liste définie par un autre bloc SELECT... FROM...WHERE

**Sous requêtes non corrélées**  
Sous requêtes de base retournant une ou plusieurs valeurs

#### Avec le signe « = »

*Quel est le produit le plus cher ?*

```
SELECT Refeprod, desiprod  
from produit  
WHERE puhtprod= (SELECT MAX(Puhtprod) from PRODUIT);
```

# Le langage S.Q.L

## Les spécificités de SQL

### Les sous requêtes

**Avec le « IN » ou « NOT IN »**

*Donner l'ensemble des informations des représentants qui suivent des clients qui habitent la ville de “Nice”.*

```
SELECT *  
FROM REPRESENTANT  
WHERE NumeRepr IN (SELECT NumeRepr FROM client WHERE VillClie ='Nice');
```

# Le langage S.Q.L

## Les spécificités de SQL

### Les sous requêtes

#### Sous requêtes renvoyant une table (dans le « FROM »)

Donner le montant moyen d'une facture parmi l'ensemble des factures existantes.

	NUMFACT	REFEPROD	PURTURER	QTEFURER
<input type="checkbox"/> Supprimer	96001	10001	152.45	1
<input type="checkbox"/> Supprimer	96001	10003	91.47	2
<input type="checkbox"/> Supprimer	96002	20005	1372.04	1
<input type="checkbox"/> Supprimer	96003	10003	91.47	2
<input type="checkbox"/> Supprimer	96003	20002	609.8	2
<input type="checkbox"/> Supprimer	96003	30006	2439.18	2
<input type="checkbox"/> Supprimer	96004	30007	2858.8	1
<input type="checkbox"/> Supprimer	96005	10001	121.96	5
<input type="checkbox"/> Supprimer	96005	30003	1219.59	5
<input type="checkbox"/> Supprimer	96006	10001	121.96	15
<input type="checkbox"/> Supprimer	96007	30009	4116.12	5
<input type="checkbox"/> Supprimer	96008	20005	1372.04	3
<input type="checkbox"/> Supprimer	96008	20010	914.69	3
<input type="checkbox"/> Supprimer	96008	30003	1219.59	3
<input type="checkbox"/> Supprimer	96008	30009	3811.23	3
<input type="checkbox"/> Supprimer	96009	20010	914.69	5
<input type="checkbox"/> Supprimer	96010	10001	121.96	1
<input type="checkbox"/> Supprimer	96010	20002	609.8	1

*La requête va se faire à partir de la table FACTURER*

*Il va falloir calculer le montant de chaque facture et ensuite calculer la moyenne de ces montants.*

# Le langage S.Q.L

## Les spécificités de SQL

### Les sous requêtes

#### Sous requêtes renvoyant une table (dans le « FROM »)

On va décomposer le travail

1. on calcule dans une première requête, le montant de chaque facture (MtFact) et on nomme cette requête TOTALFACTURES

```
SELECT NumeFact, SUM(QteFurer *  
PuhtUrer) as Mtfact  
FROM FACTURER  
GROUP BY NumeFact
```

NUMFACT	MtFact
96001	335.38999938964844
96002	1372.0400390625
96003	6280.899841308594
96004	2850.800048828125
96005	6707.749824523926
96006	1829.3999062670898
96007	20580.600585375
96008	21952.649963378906
96009	4573.450012207031
96010	731.7599868774414

2. On calcule le montant moyen d'une facture

```
SELECT AVG(MtFact) As Moyenne  
FROM TOTALFACTURES
```

# Le langage S.Q.L

## Les spécificités de SQL

### Les sous requêtes

#### Sous requêtes renvoyant une table (dans le « FROM »)

On va décomposer le travail

En une seule requête, le tout s'écrira :

```
SELECT AVG(MtFact) As Moyenne  
FROM (SELECT SUM(QteFurer * PuhtUrer) as Mtfact FROM FACTURER  
GROUP BY NumeFact) TOTALFACTURES
```

# Le langage S.Q.L

## Les spécificités de SQL

### Les sous requêtes

#### Sous requêtes renvoyant une donnée calculée

Dans le SELECT

*Donner les numéros de facture, leurs montants HT respectifs et le montant global HT de toutes les factures.*

```
SELECT NumFact, SUM  
(PuhtUrer*QtefUrer) as MontantFacture,  
(SELECT SUM(PuhtUrer*QtefUrer) from  
FACTURER) As MontantTotalFactures  
FROM FACTURER  
GROUP BY NumFact
```

NumFact	MontantFacture	MontantTotalFactures
96001	335.3999339964844	67214.74028778076
96002	1372.0400396625	67214.74028778076
96003	6280.899841308594	67214.74028778076
96004	2850.8000468828125	67214.74028778076
96005	670749824523926	67214.74028778076
96006	1829.899862670898	67214.74028778076
96007	2080.6005859375	67214.74028778076
96008	2195.649963378906	67214.74028778076
96009	4570.450012207031	67214.74028778076
96010	7317599868774414	67214.74028778076

# Le langage S.Q.L

## Les spécificités de SQL

### Les sous requêtes

#### Sous requêtes renvoyant une donnée calculée

Dans le WHERE

*Donner les numéros et dates de facture dont les quantités de produits facturés sont supérieures à la moyenne des quantités facturées de l'ensemble des Factures.*

```
SELECT DISTINCT FACTURE.NumeFact, FACTURE.DateFact
from FACTURE INNER JOIN FACTURER ON
FACTURE.NumeFact=FACTURER.NumeFact
WHERE QteFurer>(SELECT AVG(QteFurer) from facturer)
```

+ Options	
NumeFact	DateFact
96005	2019-10-01
96006	2019-01-20
96007	2019-01-22
96009	2019-01-25

# Le langage S.Q.L

## Les spécificités de SQL

### Les sous requêtes

#### Sous requêtes renvoyant une donnée calculée

##### Dans le HAVING

**Donner les factures (numéros ,montant) dont le montant est supérieur à la moyenne du montant des factures.**

```
SELECT Numefact, sum(QtefUrer*PuhtUrer)
FROM FACTURER
GROUP BY Numefact
HAVING sum(QtefUrer*PuhtUrer)> (SELECT
AVG(MtFact) AS Moyenne from (Select
SUM(FACTURER.QteFurer* FACTURER.PuhtUrer) as
MtFact from FACTURER group by
Numefact) TOTALFACTURE)
```

+ Options		
Numefact	sum(QtefUrer*PuhtUrer)	
96007	20580.6005859375	
96008	21952.649963378906	

# Le langage S.Q.L

## Les spécificités de SQL

### Les vues

Une vue est une table virtuelle. Elle peut donc être utilisée pratiquement comme une table. Elle ne stocke aucune donnée mais uniquement le résultat d'une requête.

#### Création d'une vue

##### Forme générale.

```
CREATE VIEW NomVue AS OrdreSelect
```

L'Ordre Select est une requête qui permet la création de la vue

##### Exemple : Créer une vue affichant les clients et le nom de leur représentant

```
CREATE VIEW ClientsEtRepresentants as
    Select NumeClie, NomeClie, NomeRepr
    from Client,Representant
    WhereClient.Numerepr=Representant.Numerepr;
```

# Le langage S.Q.L

## Les spécificités de SQL

### Les vues

Une vue est une table virtuelle. Elle peut donc être utilisée pratiquement comme une table. Elle ne stocke aucune donnée mais uniquement le résultat d'une requête.

#### Suppression d'une vue

Forme générale	Exemple
<code>DROP VIEW <b>NomVue</b></code>	<code>DROP VIEW ClientsEtRepresentants</code>

# Le langage S.Q.L

**Applications**

**BLOC1**

**BDD**

# **Le langage S.Q.L : Interrogation de la base**

**FIN**