

## Ordre SELECT

---

<b>SELECT</b> liste de colonnes du résultat	<i>PROJECTIONS, AGREGATS</i>
<b>FROM</b> liste des tables utilisées	<i>TABLES UTILES, JOINTURES</i>
[ <b>WHERE</b> critère de sélection des lignes ]	<i>SELECTIONS</i>
[ <b>GROUP BY</b> liste des colonnes de regroupement ]	<i>REGROUP. D'AGREGATS</i>
[ <b>HAVING</b> critère de sélection des lignes pour les valeurs agrégées ]	<i>SELECTIONS</i>
[ <b>ORDER BY</b> critère de classement des lignes du résultat ] ;	

### L'ordre SELECT : obligatoire

---

définit la liste des colonnes qui devront être retournées comme résultat final : c'est l'équivalent de l'opération de projection.

### La clause FROM : obligatoire

---

La clause FROM de l'ordre SELECT définit la table ou la liste des tables jointes nécessaire à la fourniture de la réponse.

### La clause WHERE : seulement en cas de sélection

---

La clause WHERE de l'ordre SELECT définit le critère de choix des lignes à conserver dans le résultat final : c'est l'équivalent de l'opération de sélection.

### La clause ORDER BY : seulement si un classement est demandé

---

La clause ORDER BY de l'ordre SELECT définit le classement des lignes dans le résultat final (n'existe pas en algèbre relationnel)

**Exemple de table**

Soit la relation (ou table) personnel (numero, nom, prenom, ville, salaire, entree)

numero	nom	prenom	ville	salaire	entree
1	dupont	max	arras	1000	01/01/07
2	durant	tim	aix	1500	15/03/07
3	lambert	betty	pau	1350	20/04/07
4	bradford	jean	arras	1250	30/09/07

**Projections : SELECT, choisir les colonnes du résultat**

Requete1 : lister les nom et prénom de personnel

```
SELECT nom, prenom
FROM personnel;
```

numero	nom	prenom	ville	salaire	entree
1	dupont	max	arras	1000	01/01/07
2	durant	tim	aix	1500	15/03/07
3	lambert	betty	pau	1350	20/04/07
4	bradford	jean	arras	1250	30/09/07



nom	prenom
dupont	max
durant	tim
lambert	betty
bradford	jean

Requete2 : lister les villes de personnel

```
SELECT ville
FROM personnel;
```

par défaut, la projection SQL ne supprime pas les doublons de lignes, le défaut est ALL :

```
SELECT ALL ville
FROM personnel ;
```

numero	nom	prenom	ville	salaire	entree
1	dupont	max	arras	1000	01/01/07
2	durant	tim	aix	1500	15/03/07
3	lambert	betty	pau	1350	20/04/07
4	bradford	jean	arras	1250	30/09/07



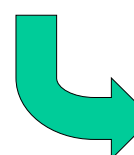
ville
arras
aix
pau
arras

**Projection avec suppression des doublons de lignes : DISTINCT**

Requete2B : lister les différentes villes de personnel

```
SELECT DISTINCT ville
FROM personnel ;
```

numero	nom	prenom	ville	salaire	entree
1	dupont	max	arras	1000	01/01/07
2	durant	tim	aix	1500	15/03/07
3	lambert	betty	pau	1350	20/04/07
4	bradford	jean	arras	1250	30/09/07

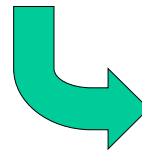


ville
arras
aix
pau

**Projections avec alias de colonne : AS**lister les nom, prenom et salaire mensuel (*en renommant l'intitulé de salaire*)

```
SELECT nom, prenom,
       salaire AS salaireMensuel
FROM personnel;
```

numero	nom	prenom	ville	salaire	entree
1	dupont	max	arras	1000	01/01/07
2	durant	tim	aix	1500	15/03/07
3	lambert	betty	pau	1350	20/04/07
4	bradford	jean	arras	1250	30/09/07



nom	prenom	salaireMensuel
dupont	max	1000
durant	tim	1500
lambert	betty	1350
bradford	jean	1250

**Projections avec colonne calculée et alias de colonne**

Des colonnes calculées peuvent être ajoutées au résultat. Les opérateurs de calcul sont les opérateurs arithmétiques : +, -, \* et /. L'utilisation des parenthèses est recommandé afin de définir précisément la priorité des calculs.

lister les nom, prenom et salaire annuel (*en renommant l'intitulé de salaire*)

```
SELECT nom, prenom,
       (salaire * 12)
       AS salaireAnnuel,
       (salaire / (35 * 4) )
       AS tauxHoraire
FROM personnel;
```

nom	prenom	salaireAnnuel	tauxHoraire
dupont	max	12000.00	7.142857
durant	tim	18000.00	10.714286
lambert	betty	16200.00	9.642857
bradford	jean	15000.00	8.928571

**Projections avec colonne calculée avec fonctions ou variables de la BD**

De nombreuses fonctions sont disponibles, parmi lesquelles des fonctions de date:

fonction	ce qu'elle retourne
EXTRACT(YEAR FROM une date) ou YEAR(une date)	retourne l'année de la date passée en argument
EXTRACT(MONTH FROM une date) ou MONTH(une date)	retourne le mois de la date passée en argument
EXTRACT(DAY FROM une date) ou DAY(une date)	retourne le jour du mois de la date passée en arguments
WEEKDAY(une date)	retourne le numéro du jour de la semaine (0=lundi, 1=mardi, etc ?)
CURRENT_DATE	donne la date du jour (c'est une variable de la BD)

lister les nom, prenom et année d'entrée (*en renommant l'année d'entrée*)

```
SELECT nom, prenom,
       YEAR(entree) AS "Entree en"
FROM personnel;
```

nom	prenom	Entree en
dupont	max	2007
durant	tim	2007
lambert	betty	2007
bradford	jean	2007

Sélections : WHERE

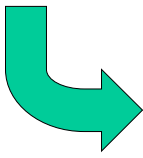
Requete1 : lister les informations de personnel si le numéro est égal à 2 ;

numero	nom	prenom	ville	salaire	entree
1	dupont	max	arras	1000	01/01/07
2	durant	tim	aix	1500	15/03/07
3	lambert	betty	pau	1350	24/04/07
4	bradford	jean	arras	1250	30/09/07

```

SELECT *
FROM personnel
WHERE numero = 2;
    
```



numero	nom	prenom	ville	salaire	entree
2	durant	tim	aix	1500	15/03/07

Les opérateurs de comparaison :

=	égal	!=	différent
>	supérieur	>=	supérieur ou égal
<	inférieur	<=	inférieur ou égal

Requete : lister les nom et prénom **et date d'entrée** de personnel si la date d'entrée est antérieur au 1<sup>er</sup> avril 2007;

```

SELECT nom, prenom, entree
FROM personnel
WHERE entree < "2007-04-01";
    
```

nom	prenom	entree
dupont	max	2007-01-01
durant	tim	2007-03-15

Requete : lister les nom et prénom de personnel si le prénom est Betty

```

SELECT nom
FROM personnel
WHERE prenom = "betty";
    
```

nom
lambert

Les connecteurs logiques :

Ils permettent de construire des conditions plus complexes :

AND	et	OR	ou
NOT	non		

Requete : lister les nom et prénom de personnel si le salaire est compris entre 1200 et 1400 euros (inclus);

```

SELECT nom, prenom
FROM personnel
WHERE (salaire >= 1200)
AND (salaire <= 1400);
    
```

nom	prenom
lambert	betty
bradford	jean

Requete : lister les nom et prénom de personnel si le salaire est inférieur à 1200 ou supérieur à 1400 euros

```

SELECT nom, prenom
FROM personnel
WHERE (salaire < 1200)
OR (salaire > 1400);
    
```

nom	prenom
dupont	max
durant	tim

Requete : lister les nom et prénom de personnel si le salaire n'est pas (est inférieur à 1200 ou supérieur à 1400) euros

<pre>SELECT nom, prenom FROM personnel WHERE NOT ((salaire &lt; 1200) OR (salaire &gt; 1400));</pre>	<b>nom</b>	<b>prenom</b>	
	lambert	betty	
	bradford	jean	

### Les opérateurs de comparaison spécifiques à SQL :

valeur IN (liste de valeurs)	tester si une valeur se trouve dans une liste de valeurs
valeur NOT IN (liste de valeurs)	tester si une valeur ne se trouve pas dans une liste de valeurs
valeur BETWEEN valeur1 AND valeur2	tester si une valeur se trouve dans l'intervalle [valeur1, valeur2]
valeur NOT BETWEEN valeur1 AND valeur2	tester si une valeur ne se trouve pas dans l'intervalle [valeur1, valeur2]
valeur LIKE "modele"	tester si une valeur ressemble à un modèle
valeur NOT LIKE "modele"	tester si une valeur ne ressemble pas à un modèle
modèle	est une chaîne de caractère utilisant les caractères jokers : % pour remplacer 0 à plusieurs caractères _ (tiret du 8) pour remplacer un seul caractère
valeur IS NULL	tester si une valeur est nulle (non renseignée)
valeur IS NOT NULL	tester si une valeur n'est pas nulle (renseignée)

### Opérateur IN / NOT IN

Requete : lister les numero, nom et prénom de personnel si la ville est Arras ou Pau			
<pre>SELECT numero, nom, prenom FROM personnel WHERE ville IN ( 'arras', 'aix');</pre>	<b>numero</b>	<b>nom</b>	<b>prenom</b>
	1	dupont	max
	2	durant	tim
	4	bradford	jean
Requete : lister les numero, nom et prénom de personnel si la ville n'est pas Arras ou Pau			
<pre>SELECT numero, nom, prenom FROM personnel WHERE ville NOT IN ( 'arras', 'aix');</pre>	<b>numero</b>	<b>nom</b>	<b>prenom</b>
	3	lambert	betty

### Opérateur BETWEEN / NOT BETWEEN

Requete : lister les numero, nom et prénom de personnel si le salaire est compris entre 1200 et 1400			
<pre>SELECT numero, nom, prenom FROM personnel WHERE salaire BETWEEN 1200 AND 1400;</pre>	<b>numero</b>	<b>nom</b>	<b>prenom</b>
	3	lambert	betty
	4	bradford	jean
Requete : lister les numero, nom et prénom de personnel si le salaire n'est pas compris entre 1200 et 1400			
<pre>SELECT numero, nom, prenom FROM personnel WHERE salaire NOT BETWEEN 1200 AND 1400;</pre>	<b>numero</b>	<b>nom</b>	<b>prenom</b>
	1	dupont	max
	2	durant	tim

**Opérateur LIKE / NOT LIKE**

Requete : lister les informations de personnel si le prenom commence par 'm'						
<b>SELECT *</b> <b>FROM personnel</b> <b>WHERE prenom LIKE 'm%';</b>	<b>numero</b>	<b>nom</b>	<b>prenom</b>	<b>ville</b>	<b>salaire</b>	<b>entree</b>
	1	dupont	max	arras	1000.00	2007-01-01
Requete : lister les informations de personnel si le prenom se termine par 'm'						
<b>SELECT *</b> <b>FROM personnel</b> <b>WHERE prenom LIKE '%m';</b>	<b>numero</b>	<b>nom</b>	<b>prenom</b>	<b>ville</b>	<b>salaire</b>	<b>entree</b>
	2	durant	tim	aix	1500.00	2007-03-15
Requete : lister les informations de personnel si le nom contient 'd'						
<b>SELECT *</b> <b>FROM personnel</b> <b>WHERE nom LIKE '%d%';</b>	<b>numero</b>	<b>nom</b>	<b>prenom</b>	<b>ville</b>	<b>salaire</b>	<b>entree</b>
	1	dupont	max	arras	1000.00	2007-01-01
	2	durant	tim	aix	1500.00	2007-03-15
	4	bradford	jean	arras	1250.00	2007-09-04
Requete : lister les informations de personnel si le prenom contient 'e' en 2eme position de caractère						
<b>SELECT *</b> <b>FROM personnel</b> <b>WHERE prenom LIKE '_e%';</b>	<b>numero</b>	<b>nom</b>	<b>prenom</b>	<b>ville</b>	<b>salaire</b>	<b>entree</b>
	3	lambert	betty	pau	1350.00	2007-04-20
	4	bradford	jean	arras	1250.00	2007-09-04

**Opérateur IS NULL / IS NOT NULL**

Requete : lister les informations de personnel si la date d'entrée est renseignée (n'est pas nulle)						
<b>SELECT *</b> <b>FROM personnel</b> <b>WHERE entrée IS NOT NULL;</b>	<b>numero</b>	<b>nom</b>	<b>prenom</b>	<b>ville</b>	<b>salaire</b>	<b>entree</b>
	1	dupont	max	arras	1000.00	2007-01-01
	2	durant	tim	aix	1500.00	2007-03-15
	3	lambert	betty	pau	1350.00	2007-04-20
	4	bradford	jean	arras	1250.00	2007-09-04
Requete : lister les informations de personnel si la date d'entrée n'est pas renseignée (est nulle)						
<b>SELECT *</b> <b>FROM personnel</b> <b>WHERE entree IS NULL;</b>	<b>numero</b>	<b>nom</b>	<b>prenom</b>	<b>ville</b>	<b>salaire</b>	<b>entree</b>
	(empty set : résultat vide)					

## Projection, sélection, classement

**Classement du résultat final : ORDER BY**

La clause ORDER BY permet le classement du résultat final. Plusieurs colonnes de classement peuvent être nommées (séparées par des virgules), avec pour chacune un critère de classement.

Les critères de classement sont :

- Le classement croissant : ASC, pour ASCending (valeur par défaut)
- Le classement décroissant : DESC, pour DESCending

Requete : lister les informations de personnel classées par nom

```
SELECT *
FROM personnel
ORDER BY nom ASC;
```

numero	nom	prenom	ville	salaire	entree
4	bradford	jean	arras	1250.00	2007-09-04
1	dupont	max	arras	1000.00	2007-01-01
2	durant	tim	aix	1500.00	2007-03-15
3	lambert	betty	pau	1350.00	2007-04-20

Requete : lister les informations de personnel classées par ville, puis par salaire décroissant

```
SELECT *
FROM personnel
ORDER BY ville ASC,
salaire DESC;
```

numero	nom	prenom	ville	salaire	entree
2	durant	tim	aix	1500.00	2007-03-15
4	bradford	jean	arras	1250.00	2007-09-04
1	dupont	max	arras	1000.00	2007-01-01
3	lambert	betty	pau	1350.00	2007-04-20

**Une requête complète**

Requete : lister les nom et prenom de personnel qui sont entrés au mois de mars ou avril classées par salaire

```
SELECT nom, prenom
FROM personnel
WHERE MONTH(entree) IN (3,4)
ORDER BY salaire ASC;
```

numero	nom	prenom	ville	salaire	entree
3	lambert	betty	pau	1350.00	2007-04-20
2	durant	tim	aix	1500.00	2007-03-15