

Les Données de Base sur les Bases de Données

Système d'Information (SI) & Bases de Données (BD)

Le modèle Relationnel L'algèbre Relationnelle

Le langage SQL



M. Faouzi BOUFARES

boufares@lipn.univ-paris13.fr

<http://www.lipn.univ-paris13.fr/~boufares/>

Bureau A109

Laboratoire d'Informatique de Paris Nord

LIPN – UMR 7030 du CNRS

Université Paris 13, Sorbonne Paris Cité

Institut Galilée

99, Avenue J.B. Clément

F-93430 Villetaneuse FRANCE

www.univ-paris13.fr

Aujourd'hui, j'arrête de fumer !

Le TABAC t'ABAT !

FUMER NUIT GRAVEMENT À TA SANTÉ ET
À CELLE DE TON ENTOURAGE !

M T Dents !

SMILE and the World SMILES with you !



Le Langage SQL



SQL : Structured Query Language

Le langage SQL est composé de quelques **mots clés**.

C'est un outil très puissant et facile d'accès.

Il permet **d'interroger une Base de Données** relationnelles (**faire des requêtes**) afin **d'exploiter les données** stockées, afficher des listes ou encore établir des :

- **analyses de données ;**
- **outils d'aide à la décision ;**
- **tableaux de bord.**



Les mots clés, du langage SQL, sont résumés dans le tableau ci-dessous :

Mots Clés du langage SQL	 Signification du mot clé
SELECT	Quoi ??? Champ(s)/Colonne(s)...Fonction(s) [ce que l'on veut voir]
FROM	Où ??? Table(s) [à partir de quelle(s) table(s)]
WHERE	Selon quel(s) critères??? [Filtrer les lignes dans les tables]
GROUP BY	Faire des sous totaux (groupements) !
HAVING	avec des critères??? [Filtrer les résultats dans les sous-totaux]
ORDER BY	Classer/Ordonner le résultat ! [dans un ordre croissant ou décroissant]
;	Un point virgule termine la requête ;





Une requête SQL peut donc être formulée comme suit :

SELECT ...
FROM ...

Ces **DEUX** commandes sont **OBLIGATOIRES**.
Elles permettent de désigner les **COLONNES** à afficher en précisant les **TABLES** concernées !

WHERE ...

N'est nécessaire que s'il y a des **CRITERES/CONDITIONS/FILTRES** à exprimer.
Des conditions de **Sélections** et/ou de **jointures** !

GROUP BY ...
HAVING ...

N'est nécessaire que s'il y a des **GROUPEMENTS/SOUS-TOTAUX** à effectuer.
Le **HAVING** ne peut exister qu'avec **GROUP BY** !

ORDER BY ...

N'est nécessaire que s'il y a des **CLASSEMENTS/TRIS** à effectuer.
L'ordre peut être **ASCending** ou **DESCending** !



Exemples :

1	SELECT ... FROM ... ;
2	SELECT ... FROM ... WHERE ... ;
3	SELECT ... FROM ... ORDER BY ... ;
4	SELECT ... FROM ... WHERE ... ORDER BY ... ;
5	SELECT ... FROM ... GROUP BY ... ;
6	SELECT ... FROM ... WHERE ... GROUP BY ... ;
7	SELECT ... FROM ... WHERE ... GROUP BY ... HAVING... ;
8	SELECT ... FROM ... WHERE ... GROUP BY ... HAVING... ORDER BY ... ;
9	SELECT ... FROM ... WHERE ... UNION SELECT ... FROM ... WHERE ... ;
10	SELECT ... FROM ... WHERE ... IN (SELECT ... FROM ... WHERE ...);
11	SELECT ... FROM ... WHERE ... NOT IN (SELECT ... FROM ... WHERE ...);



Le système d'information de l'entreprise **iINFO** est articulé autour de la base de données « **Gestion des stocks : GESTOCK** », composée des tables **Articles**, **Stocks** et **Dépôts**. Elle est gérée sous **ACCESS**. Sa structure (son schéma, son architecture) est décrite ci-dessous :

Table des articles :

ARTICLES (↔ **RefArt**, NomArt, PUArt)

La clé primaire ↔ est : **RefArt**

Table des Dépôts :

DEPOTS (↔ **NumDep**, NomDep, ..., MailDep)

La clé primaire ↔ est : **NumDep**

Table des stocks :

STOCKS (**RefArt**↑ ⊞, QteStk, **NumDep**↑ ⊞)

La clé primaire ↔ est implicite. Elle est composée des deux colonnes : **RefArt** et **NumDep**

La table des stocks a deux clés étrangères ↑ ⊞ :

RefArt et **NumDep**.

Pour la gestion de ses stocks, les articles sont identifiés de manière unique par la référence (**RefArt**). Les dépôts sont retrouvés grâce au numéro (**NumDep**). Un même article peut être stocké dans plusieurs dépôts. Il peut aussi être en rupture de stock !

Le **dictionnaire de données** de la BD GESTOCK est le suivant :

Champ Colonne Attribut	Type de données Domaine Contrainte	Description Signification Commentaire
MailDep	Caractères/Texte, 50	EEmail du dépôt
NomArt	Caractères/Texte, 30	Nom (désignation) de l'article
NomDep	Caractères/Texte, 20	Nom du dépôt
NumDep	Caractères/Textes, 10	Numéro du dépôt (Unique par dépôt)
PUArt	Réel, 10 dont 2 décimales	Prix unitaire de l'article
QteStk	Entier, 3, >0	Quantité en stock de l'article dans un dépôt donné
RefArt	Caractères/Texte, 8	Référence de l'article (Unique par article)
TelDep	Caractères/Texte, 15	Téléphone du dépôt
VilDep	Caractères/Texte, 50	Adresse du dépôt (N° + Rue/Avenue ...)

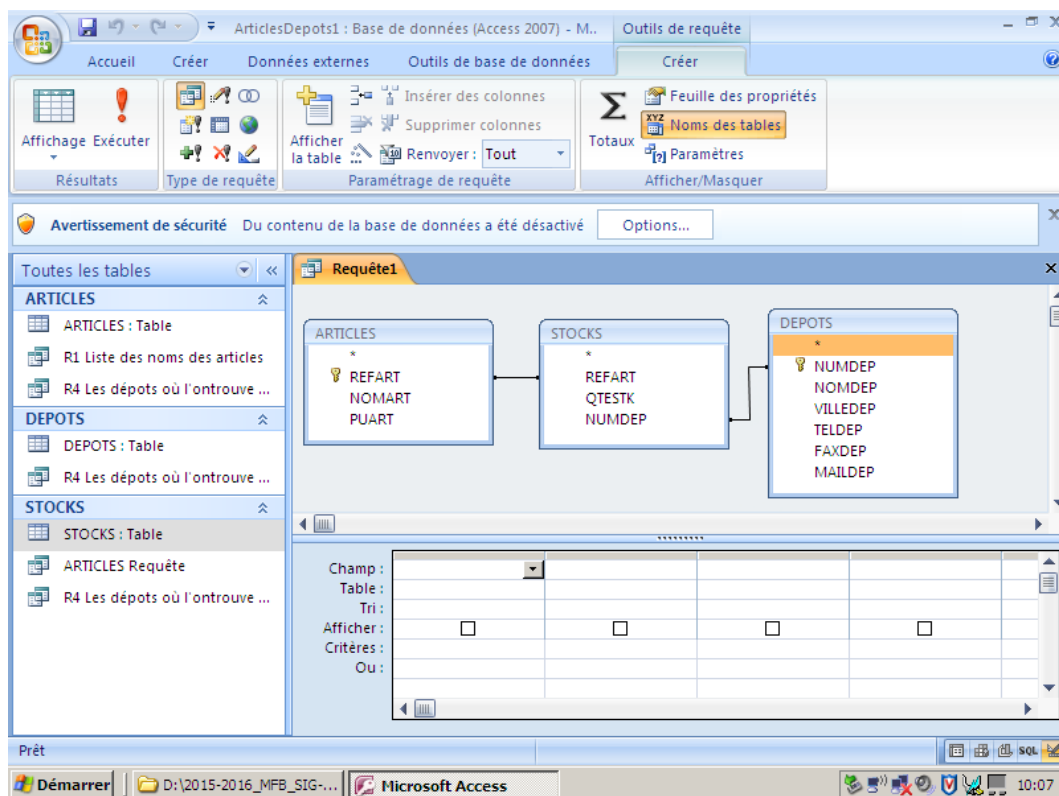
Remarques :

- Une colonne (un champ) déclaré **clé primaire** n'admet pas de valeurs en double (**doublons interdits**)
- Une colonne (un champ) **déclaré clé primaire** n'admet pas de valeurs vides (**vides interdits**)
- Une table définie avec une clé primaire n'a pas de lignes en double
- Une table ne peut avoir deux colonnes qui portent le même nom.



Etant donné la base de données « **Gestion de stocks** », composée des tables Articles, Stocks et Dépôts.

La base de données « **Gestion de stocks** » est gérée sous ACCESS, sa structure (son schéma, son architecture) est donnée ci-dessous :




L'exploitation des données va être faite moyennant les exemples de requêtes ci-dessous (A, B, C, D, E, F) :



A : Exemples de requêtes (de projection) :

Projection : Garder quelques colonnes à l'affichage →→→

```
SELECT ... FROM ... ;
```

Mots Clés du langage SQL	 Signification du mot clé
SELECT	Pour mentionner le(s) nom(s) des colonne(s) demandées à l'affichage Les noms des colonnes sont séparés par des virgules
FROM	Pour mentionner le(s) nom(s) des tables qui contiennent les colonnes demandées Les noms des tables sont séparés par des virgules
;	...

A01 : Afficher les noms des articles (NOMART), on demander d'afficher une seule colonne

```
SELECT ARTICLES.NOMART  
FROM ARTICLES;
```

```
SELECT NOMART  
FROM ARTICLES;
```



A02 : Afficher les noms et les prix unitaires des articles (NOMART, PUART), on demande d'afficher deux colonnes

```
SELECT ARTICLES.NOMART, ARTICLES.PUART  
FROM ARTICLES;
```

```
SELECT NOMART, PUART  
FROM ARTICLES;
```

```
SELECT NOMART, PUART FROM ARTICLES;
```

```
select NOMART, PUART from ARTICLES;
```

```
select nomart, puart from articles;
```

Remarque :

- **Pas de différence Minuscule/Majuscule en SQL**



A03 : Afficher toutes les colonnes de la table des articles

```
SELECT *  
FROM ARTICLES;
```

Remarque : **SELECT *** veut dire que l'on demande toutes les colonnes

A04 : Afficher toutes les références (la colonne REFART) des articles en stock (sans doublon)

```
SELECT DISTINCT STOCKS.REFART  
FROM STOCKS;
```

```
SELECT DISTINCT REFART FROM STOCKS;
```

*A05 : Afficher les références et les noms des articles dans une SEULE colonne à l'écran de nom **DescriptionArt** (**REFART, NOMART**).*

On demande d'afficher deux colonnes de la table dans une seule colonne à l'écran.

C'est la concaténation (assemblage de colonnes) de plusieurs colonnes avec l'opérateur &. Chaque partie du texte à afficher est composée par exemple de :

Un texte fixe "L'article de référence : " suivi par une colonne de la BD suivie par un texte fixe " a comme nom : " suivi par une colonne de la BD.

Requête A05 → Concaténation de plusieurs colonnes des tables	
DescriptionArticle	Expr10000 !
L'article de référence : R01 a comme nom :	Téléviseur Samtung MFB1
L'article de référence : R02 a comme nom :	Ail Phone 17
L'article de référence : R03 a comme nom :	Câble Coaxial 15
L'article de référence : R04 a comme nom :	Téléviseur Somy E2
L'article de référence : R05 a comme nom :	Clé USB Momo 16
L'article de référence : R06 a comme nom :	Blanc Berry
L'article de référence : R07 a comme nom :	HaWhy

SELECT

"L'article de référence : " & ARTICLES.REFART

& " a comme nom : " & ARTICLES.NOMART

AS DescriptionArticle ← Renommer une colonne à l'affichage


FROM

ARTICLES;

B : Exemples de requêtes (de sélection) :

Sélection : Garder quelques lignes à l'affichage selon des critères →→→

```
SELECT ... FROM ... WHERE... ;
```

Mots Clés du langage SQL	 Signification du mot clé
SELECT	<p>Pour mentionner le(s) nom(s) des colonne(s) demandées à l'affichage</p> <p>Les noms des colonnes sont séparés par des virgules</p>
FROM	<p>Pour mentionner le(s) nom(s) des tables qui contiennent les colonnes demandées</p> <p>Les noms des tables sont séparés par des virgules</p>
WHERE	<p>Pour mentionner les conditions/critères de sélections dans des tables (pour filtrer dans les tables)</p> <p>Les conditions/critères de sélections sont combinées par des opérateurs logiques (tels que : AND, OR, NOT)</p> <p>Les conditions/critères de sélections des opérateurs de comparaisons (tels que <, <=, =, >=, >, <>, IN, NOT IN, BETWEEN, NOT BETWEEN)... LIKE</p>
;	



B01 : Afficher les noms et les prix unitaires des articles (NOMART, PUART) qui vérifient le(s) critère(s) :

le prix unitaire est supérieur à 230€

```
SELECT  
ARTICLES.NOMART, ARTICLES.PUART  
FROM  
ARTICLES  
WHERE  
((ARTICLES.PUART)>230)) ;
```

```
SELECT NOMART, PUART  
FROM  
ARTICLES  
WHERE PUART > 230 ; ← Numérique !
```

*B02 : Afficher le prix unitaire de l'articles (PUART)
qui vérifie le(s) critère(s) :
le nom est égal à Blanc Berry*

```
SELECT ARTICLES.PUART  
FROM ARTICLES  
WHERE  
(((ARTICLES.NOMART)="Blanc Berry")) ;
```

```
SELECT PUART FROM ARTICLES  
WHERE NOMART = "Blanc Berry" ; ← Texte
```



B03 : Afficher les noms et les prix unitaires des articles (NOMART, PUART) qui vérifient le(s) critère(s) :

le prix unitaire est supérieur à 100€

et

le prix unitaire est inférieur à 300€

```
SELECT  
ARTICLES.NOMART, ARTICLES.PUART  
FROM  
ARTICLES  
WHERE  
((ARTICLES.PUART)>=100) AND  
((ARTICLES.PUART)<=300) ;
```

```
SELECT NOMART, PUART  
FROM ARTICLES  
WHERE (PUART>=100) AND (PUART<=300) ;
```

```
SELECT NOMART, PUART  
FROM ARTICLES  
WHERE PUART BETWEEN 100 AND 300 ;
```

B04 : Afficher les noms et les prix unitaires des articles (NOMART, PUART) qui vérifient le(s) critère(s) :

le prix unitaire est égal à 100€ ou le prix unitaire est égal à 200€

ou

le prix unitaire est égal à 300€ ou le prix unitaire est égal à 400€

```
SELECT NOMART, PUART  
FROM ARTICLES  
WHERE (PUART=100) OR (PUART=200)  
OR (PUART=300) OR (PUART=400) ;
```

```
SELECT NOMART, PUART  
FROM ARTICLES  
WHERE PUART IN (100, 200, 300, 400) ;
```



B05 : Afficher les noms et les prix unitaires des articles (NOMART, PUART) qui vérifient le(s) critère(s) :

le prix unitaire n'est égal ni à 100€ ni à 200€ ni à 300€ ni à 400€

```
SELECT NOMART, PUART
FROM ARTICLES
WHERE (PUART <> 100) AND (PUART <> 200)
AND (PUART <> 300) AND (PUART <> 400) ;
```

```
SELECT NOMART, PUART
FROM ARTICLES
WHERE PUART NOT IN (100, 200, 300, 400) ;
```

B06 : Afficher les noms et les prix unitaires des articles (NOMART, PUART) qui vérifient le(s) critère(s) :

le nom commence par Ail

```
SELECT  
ARTICLES.NOMART, ARTICLES.PUART  
FROM  
ARTICLES  
WHERE  
(ARTICLES.NOMART) LIKE "Ail*" ;
```

```
SELECT NOMART, PUART  
FROM ARTICLES  
WHERE NOMART LIKE "Ail*" ;
```

Remarque :

*commence par →→→ l'étoile * à la fin*

B07 : Afficher les noms et les prix unitaires des articles (NOMART, PUART) qui vérifient le(s) critère(s) :

le nom se termine par ung

```
SELECT
ARTICLES.NOMART, ARTICLES.PUART
FROM
ARTICLES
WHERE
(ARTICLES.NOMART) LIKE "*ung" ;
```

```
SELECT NOMART, PUART
FROM ARTICLES
WHERE NOMART LIKE "*ung") ;
```

Remarque :

*Se termine par →→→ l'étoile * au début*

B08 : Afficher les noms et les prix unitaires des articles (NOMART, PUART) qui vérifient le(s) critère(s) :

le nom contient ph

```
SELECT NOMART, PUART
FROM ARTICLES
WHERE NOMART LIKE "*ph*") ;
```

Remarque :

contient →→→ *l'étoile * au début et à la fin*

B09 : Afficher les noms et les prix unitaires des articles (NOMART, PUART) qui vérifient le(s) critère(s) :

le nom commence par Ail

et

le prix unitaire est supérieur à 230€

```
SELECT
ARTICLES.NOMART, ARTICLES.PUART
FROM
ARTICLES
WHERE
(((ARTICLES.NOMART) LIKE "Ail*")
AND
((ARTICLES.PUART)>230));
```

```
SELECT NOMART, PUART
FROM ARTICLES
WHERE
(NOMART LIKE "Ail*")
AND
(PUART > 230) ;
```

B10 : Afficher les numéros, les noms et les téléphones des dépôts pour lesquels on ne connaît pas le mail (NUMDEP, NOMDEP, TELDEP) qui vérifient le(s) critère(s) :

le mail est vide (inconnu, nul)

```
SELECT NUMDEP, NOMDEP, TELDEP
FROM DEPOTS
WHERE MAILDEP IS NULL ;
```

```
SELECT NUMDEP, NOMDEP, TELDEP
FROM DEPOTS
WHERE MAILDEP = '' ;
```




Mots Clés du langage SQL	 Signification du mot clé																					
WHERE	<p>Pour mentionner les conditions/critères de sélections dans des tables (pour filtrer dans les tables)</p> <p>Les conditions/critères de sélections sont combinées par des opérateurs logiques (tels que AND, OR, NOT)</p> <p>Les conditions/critères de sélections des opérateurs de comparaisons (tels que <, <=, =, >=, >, <>, IN, BETWEEN)... LIKE</p>																					
<p>Les critères de sélection dans la clause WHERE utilisent les opérateurs suivants :</p>	<table border="0"> <tr> <td>AND</td> <td>OR</td> <td>NOT</td> </tr> <tr> <td>BETWEEN</td> <td>NOT BETWEEN</td> <td></td> </tr> <tr> <td>IN (...)</td> <td>NOT IN (...)</td> <td></td> </tr> <tr> <td>LIKE →→→ *</td> <td>NOT LIKE</td> <td></td> </tr> <tr> <td>IS NULL = '''</td> <td>IS NOT NULL</td> <td></td> </tr> <tr> <td>=</td> <td><=</td> <td><</td> </tr> <tr> <td>>=</td> <td>></td> <td><></td> </tr> </table> <p>Les textes →→Guillemets ''' →→→ "letexte"</p> <p>Les dates →→Dièses ## →→→ #mm/jj/aaaa#</p> <p>Les numériques →→→→→ rien</p>	AND	OR	NOT	BETWEEN	NOT BETWEEN		IN (...)	NOT IN (...)		LIKE →→→ *	NOT LIKE		IS NULL = '''	IS NOT NULL		=	<=	<	>=	>	<>
AND	OR	NOT																				
BETWEEN	NOT BETWEEN																					
IN (...)	NOT IN (...)																					
LIKE →→→ *	NOT LIKE																					
IS NULL = '''	IS NOT NULL																					
=	<=	<																				
>=	>	<>																				
;																						



C : Exemples de requêtes (de tri/de classement) :

Trier : Classer/Trier/Ordonner les lignes selon des colonnes →→→

```
SELECT ... FROM ... WHERE... ORDER BY ... ;
```

Mots Clés du langage SQL	 Signification du mot clé
SELECT	Pour mentionner le(s) nom(s) des colonne(s) demandées à l'affichage Les noms des colonnes sont séparés par des virgules
FROM	Pour mentionner le(s) nom(s) des tables qui contiennent les colonnes demandées Les noms des tables sont séparés par des virgules
WHERE	Pour mentionner les conditions/critères de sélections dans des tables (pour filtrer dans les tables)
ORDER BY	Pour mentionner les colonnes de tri ainsi que l'ordre croissant (par défaut) ou décroissant Ordre croissant : ASC (ASCending) par défaut ↑ Ordre décroissant : DESC (DESCending) ↓
;	



*C01 : Classer/Trier/Ordonner les articles par ordre croissant sur le prix unitaire : (NOMART, PUART)
PUART croissant ↑)*

```
SELECT ARTICLES.NOMART,  
ARTICLES.PUART  
FROM ARTICLES  
ORDER BY ARTICLES.PUART ;
```

```
SELECT ARTICLES.NOMART,  
ARTICLES.PUART  
FROM ARTICLES  
ORDER BY ARTICLES.PUART ASC ;
```

```
SELECT NOMART, PUART  
FROM ARTICLES  
ORDER BY PUART ASC ;
```

```
SELECT NOMART, PUART FROM ARTICLES  
ORDER BY 2 ASC ;
```

C02 : Classer/Trier/Ordonner les articles par ordre décroissant sur le prix unitaire : (NOMART, PUART) PUART décroissant ↓

```
SELECT NOMART, PUART FROM ARTICLES  
ORDER BY PUART DESC ;
```

```
SELECT NOMART, PUART FROM ARTICLES  
ORDER BY 2 DESC ;
```

Remarque : **2** désigne le numéro de la colonne dans l'ordre d'apparition à l'affichage !

C03 : Classer/Trier/Ordonner les articles, dont le prix unitaire est inférieur à 300€, par ordre croissant sur le prix unitaire : (NOMART, PUART) qui vérifient le(s) critère(s) :

*PUART < 300€
PUART croissant)*

```
SELECT ARTICLES.NOMART, ARTICLES.PUART  
FROM ARTICLES  
WHERE ((ARTICLES.PUART)<300))  
ORDER BY 2 ;
```

```
SELECT ARTICLES.NOMART AS Article,  
ARTICLES.PUART AS PrixUnitaire  
FROM ARTICLES  
WHERE ((ARTICLES.PUART)<300))  
ORDER BY 2 ;
```

```
SELECT NOMART AS Article,  
PUART AS PrixUnitaire  
FROM ARTICLES  
WHERE ARTICLES.PUART<300  
ORDER BY 2 ;
```



C04 : Classer/Trier/Ordonner les articles, dont le prix unitaire est inférieur à 200€, par ordre croissant sur le nom et par ordre décroissant sur le prix unitaire : (NOMART, PUART) qui vérifient le(s) critère(s) :

PUART < 200€

NOMART croissant et PUART décroissant

```
SELECT NOMART AS Article,  
PUART AS PrixUnitaire  
FROM ARTICLES  
WHERE PUART < 200  
ORDER BY NOMART , PUART DESC ;
```


```
SELECT NOMART AS Article,  
PUART AS PrixUnitaire  
FROM ARTICLES  
WHERE PUART < 200  
ORDER BY 1 ASC, 2 DESC ;
```

D : Exemples de calculs (avec les fonctions de calcul : **somme, moyenne, minimum, maximum, comptage**) :

Calculer/Fonction/Function : (Somme **SUM**, Moyenne **AVG**, Minimum **MIN**, Maximum **MAX**, Dénombrer **COUNT**)

Le calcul se fait sur les lignes selon des colonnes →→→

```
SELECT FUNCTION(...) ... FROM ...WHERE ... ;
```

Mots Clés du langage SQL	 Signification du mot clé
SELECT	<p>Pour mentionner la ou les fonctions de calcul demandées sur de(s) nom(s) des colonne(s) demandées à l’affichage</p> <p>Les fonctions() sont séparés par des virgules et renommées avec AS le nom du résultat FUNCTION(NomDeColonne) : SUM(...) ou AVG(...), ou MIN(...) ou MAX(...) ou COUNT(...) qui peut être écrit COUNT(*)</p>
FROM	<p>Pour mentionner le(s) nom(s) des tables qui contiennent les colonnes demandées</p> <p>Les noms des tables sont séparés par des virgules</p>
WHERE	<p>Pour mentionner les conditions/critères de sélections dans des tables (pour filtrer dans les tables)</p>
;	



D01 : Calculer la moyenne du prix unitaire :

AVG(PUART)

```
SELECT  
AVG(ARTICLES.PUART) AS MoyenneDePUART  
FROM  
ARTICLES;
```

```
SELECT  
AVG(ARTICLES.PUART) AS MoyPU  
FROM  
ARTICLES;
```

```
SELECT AVG(PUART) AS MoyPU  
FROM ARTICLES;
```

```
SELECT AVG(PUART)  
FROM ARTICLES;
```


D02 : Calculer/Chercher le prix unitaire minimal :
MIN(PUART)

```
SELECT MIN(PUART) AS [Min PU]  
FROM ARTICLES;
```

D03 : Calculer/Chercher le prix unitaire maximal :
MAX(PUART)

```
SELECT MAX(PUART) AS MaxPU  
FROM ARTICLES;
```

D04 : Calculer le total (la somme) des quantités en stock :
SUM(QTESTK)

```
SELECT SUM(QTESTK) AS TotalQteStk  
FROM STOCKS;
```

D05 : Calculer le nombre (compter) d'articles :

COUNT(*)

```
SELECT COUNT(*) AS NbrArticles  
FROM ARTICLES;
```

```
SELECT COUNT(REFART) AS NbrArticles  
FROM ARTICLES;
```

```
SELECT COUNT(NOMART) AS NbrArticles  
FROM ARTICLES;
```

```
SELECT COUNT(PUART) AS NbrArticles  
FROM ARTICLES;
```

Remarque : Pour compter le nombre de lignes dans une table, le nom de la colonne utilisé dans la fonction **COUNT n'a pas d'importance. C'est pour cette raison que l'on peut utiliser le caractère ***** qui désigne **n'importe quel nom de colonne**.**

*D06 : Calculer le nombre (compter) d'articles dont le prix est inférieur à 100€ : **COUNT(*)** qui vérifient le(s) critère(s) :*

PUART < 100€

```
SELECT  
COUNT(ARTICLES.PUART) AS NbrArt  
FROM  
ARTICLES  
WHERE  
ARTICLES.PUART < 100 ;
```

```
SELECT COUNT(*) AS NbrArt  
FROM ARTICLES  
WHERE PUART < 100 ;
```



*D07 : Calculer/Chercher la moyenne des prix unitaires des articles ainsi que les prix minimum et maximum (statistiques avec plusieurs fonctions à la fois) : **AVG(PUART)**, **MIN(PUART)**, **MAX(PUART)***

```
SELECT  
AVG(PUART) AS MoyPU,  
MIN(PUART) AS MinPU,  
MAX(PUART) AS MaxPU  
FROM ARTICLES ;
```

Remarque : Cinq fonctions de calcul

AVG : permet de calculer la moyenne (**AVerage**) des valeurs qui se trouvent dans une colonne

MIN : permet de calculer la plus petite (**MINimum**) des valeurs qui se trouvent dans une colonne

MAX : permet de calculer la plus grande (**MAXimum**) des valeurs qui se trouvent dans une colonne


SUM : permet de calculer la **somme** (le total) des valeurs qui se trouvent dans une colonne

COUNT : permet de calculer le **nombre** de valeurs qui se trouvent dans une colonne

E : Exemples de calculs avec **groupement (des sous totaux de groupement avec les fonctions de calcul : somme, moyenne, minimum, maximum, comptage) :**

Calculer : (Somme SUM, Moyenne AVG, Minimum MIN, Maximum MAX, Dénombrer COUNT) les lignes selon des colonnes →→→

```
SELECT Colonne(s), FUNCTION(...) ...
FROM ...WHERE ...ORDER BY...
GROUP BY Colonne(s) HAVING ... ;
```

Mots Clés du langage SQL	 Signification du mot clé
SELECT	Pour mentionner la ou les fonctions de calcul demandées sur de(s) nom(s) des colonne(s) demandées à l’affichage Deux colonnes au moins doivent être affichées !
FROM	Pour mentionner le(s) nom(s) des tables concernées
WHERE	Cette clause peut ne pas exister (pas de sélection dans les lignes des tables de la Base de Données)
ORDER BY	Cette clause peut ne pas exister (pas de tri)
GROUP BY	
HAVING	
;	



E01 : Calculer le total des quantité en stock par article :

REFART, SUM(QTESTK)

```
SELECT  
STOCKS.REFART,  
SUM(STOCKS.QTESTK) AS SommeDeQTESTK  
FROM STOCKS  
GROUP BY  
STOCKS.REFART;
```

```
SELECT REFART, SUM(QTESTK) AS TotalStk  
FROM STOCKS  
GROUP BY REFART;
```



E02 : Calculer le nombre d'articles différents par dépôt :

NUPDEP, COUNT(*)

```
SELECT  
STOCKS.NUMDEP AS Dépôt,  
COUNT(*) AS NbrArtDiff  
FROM STOCKS  
GROUP BY  
STOCKS.NUMDEP;
```

```
SELECT  
NUMDEP AS Dépôt, COUNT(*) AS NbrArtDiff  
FROM STOCKS  
GROUP BY NUMDEP ;
```

E03 : Calculer le nombre d'articles différents par dépôt, n'afficher que les dépôts presque vides (dont le nombre d'articles différents, qui est en cours de calcul, est < 4 :

NUPDEP, COUNT(*)

nombre d'articles différents est < 4

```
SELECT  
STOCKS.NUMDEP AS Dépôt,  
COUNT(*) AS NbrArtDiff  
FROM STOCKS  
GROUP BY STOCKS.NUMDEP  
HAVING COUNT(*) < 4;
```


```
SELECT  
NUMDEP AS Dépôt, COUNT(*) AS NbrArtDiff  
  
FROM STOCKS  
  
GROUP BY NUMDEP HAVING COUNT(*) < 4;
```



F_Join : Exemples de requêtes sur **plusieurs tables à la fois** (Jointure+Projection+sélection=JOIN) :

Sélection : Garder quelques lignes à l'affichage selon des critères →→→

SELECT ... FROM ... WHERE... ;

Mots Clés du langage SQL	 Signification du mot clé
SELECT	Pour mentionner le(s) nom(s) des colonne(s) demandées à l'affichage
FROM	Pour mentionner le(s) nom(s) des tables qui contiennent les colonnes demandées INNER JOIN ... ON ...
WHERE	Pour mentionner les conditions/critères de jointure(s) et de sélection(s) dans des tables (pour filtrer dans les tables)
;	



F_Join_01: Afficher les noms des articles en stock, leurs prix unitaires, leurs quantités en stock et les dépôts dans lesquels ils se trouvent :

NOMART, PUART, QTESTK, NUMDEP

Access

```
SELECT  
ARTICLES.NOMART, ARTICLES.PUART,  
STOCKS.QTESTK, STOCKS.NUMDEP  
FROM ARTICLES INNER JOIN STOCKS  
ON ARTICLES.REFART = STOCKS.REFART;
```

Standard

```
SELECT  
ARTICLES.NOMART, ARTICLES.PUART,  
STOCKS.QTESTK, STOCKS.NUMDEP  
FROM ARTICLES , STOCKS  
WHERE  
ARTICLES.REFART = STOCKS.REFART;
```

```
SELECT  
NOMART, PUART, QTESTK, NUMDEP  
FROM ARTICLES , STOCKS  
WHERE  
ARTICLES.REFART = STOCKS.REFART;
```



Remarques sur la jointure :

- La jointure de la **table 1** avec la **table 2** est équivalente (donne le même résultat) que la jointure de la **table 2** avec la **table 1**.
- L'ordre des tables n'est pas important pour construire le résultat.
- On peut joindre une table avec elle-même.

```
SELECT  
ARTICLES.NOMART, ARTICLES.PUART, STOCKS.QTESTK,  
STOCKS.NUMDEP  
FROM ARTICLES INNER JOIN STOCKS  
ON ARTICLES.REFART = STOCKS.REFART;
```

Est équivalent à :

```
SELECT  
ARTICLES.NOMART, ARTICLES.PUART, STOCKS.QTESTK,  
STOCKS.NUMDEP  
FROM STOCKS INNER JOIN ARTICLES  
ON STOCKS.REFART= ARTICLES.REFART;
```

```
SELECT  
NOMART, PUART, QTESTK, NUMDEP  
FROM ARTICLES , STOCKS  
WHERE ARTICLES.REFART = STOCKS.REFART;
```

Est équivalent à :

```
SELECT  
NOMART, PUART, QTESTK, NUMDEP  
FROM STOCKS , ARTICLES  
WHERE STOCKS.REFART = ARTICLES.REFART;
```



F_Join_02: Afficher les noms des articles en stock, leurs prix unitaires, leurs quantités en stock et les dépôts dans lesquels ils se trouvent triés par article et par dépôt :

NOMART, PUART, QTESTK, NUMDEP

```
SELECT  
ARTICLES.NOMART, ARTICLES.PUART,  
STOCKS.QTESTK, STOCKS.NUMDEP  
FROM ARTICLES INNER JOIN STOCKS  
ON ARTICLES.REFART = STOCKS.REFART  
ORDER BY ARTICLES.NOMART,  
STOCKS.NUMDEP;
```

```
SELECT  
NOMART, PUART, QTESTK, NUMDEP  
FROM ARTICLES INNER JOIN STOCKS  
ON ARTICLES.REFART = STOCKS.REFART  
ORDER BY NOMART , NUMDEP ;
```

```
SELECT NOMART, PUART, QTESTK, NUMDEP  
FROM ARTICLES , STOCKS  
WHERE ARTICLES.REFART = STOCKS.REFART  
ORDER BY 1, 4;
```

F_Join_03: Afficher la quantité totale en stock pour tous les articles en stock, tout dépôt confondu :

Article=NOMART, TotalEnStock= SUM(QTESTK)

```
SELECT  
ARTICLES.NOMART AS Article,  
SUM(STOCKS.QTESTK) AS TotalEnStock  
FROM ARTICLES INNER JOIN STOCKS  
ON ARTICLES.REFART = STOCKS.REFART  
GROUP BY ARTICLES.NOMART ;
```

```
SELECT  
NOMART AS Article,  
SUM(STOCKS.QTESTK) AS TotalEnStock  
FROM ARTICLES INNER JOIN STOCKS  
ON ARTICLES.REFART = STOCKS.REFART  
GROUP BY NOMART;
```

```
SELECT  
NOMART AS Article,  
SUM(STOCKS.QTESTK) AS TotalEnStock  
FROM ARTICLES, STOCKS  
WHERE ARTICLES.REFART = STOCKS.REFART  
GROUP BY NOMART;
```

SQL offre plusieurs types d'opérations sur **plusieurs tables :**


- **La Jointure,**
- **L'Union,**
- **L'Intersection,**
- **La différence.**



F_Union : Exemples de requêtes sur **plusieurs tables** à la fois (**Union**+Projection+sélection) :

Sélection : Garder quelques lignes à l'affichage selon des critères →→→

SELECT ... FROM ...WHERE...
UNION
 SELECT ... FROM ...WHERE... ;

Mots Clés du langage SQL	 Signification du mot clé
SELECT	Pour mentionner le(s) nom(s) des colonne(s) demandées à l'affichage
FROM	Pour mentionner le nom de la première table qui contient les colonnes demandées
WHERE	Cette clause peut ne pas exister (pas de sélection dans les lignes des tables de la Base de Données)
UNION	
SELECT	Pour mentionner le(s) mêmes nom(s) des colonne(s) demandées à l'affichage
FROM	Pour mentionner le nom de la première table qui contient les colonnes demandées
WHERE	Cette clause peut ne pas exister (pas de sélection dans les lignes des tables de la Base de Données)
;	



F_Union_01: Afficher les numéros des articles dont le prix unitaire est inférieur à 15€ ainsi que (UNION) les numéros des articles en stock au dépôt D3 :

REFART tel que **PUART < 15€**

à partir de la table des articles

AINSI QUE (UNION)

REFART tel que **NUMDEP = "D3"**

à partir de la table des stocks


```
SELECT ARTICLES.REFART  
FROM ARTICLES  
WHERE ARTICLES.PUART < 15  
UNION  
SELECT STOCKS.REFART  
FROM STOCKS  
WHERE STOCKS.NUMDEP = "D3"  
;
```

```
SELECT REFART FROM ARTICLES  
WHERE PUART < 15  
UNION  
SELECT REFART FROM STOCKS  
WHERE NUMDEP = "D3"  
;
```


F_Inter : Exemples de requêtes sur **plusieurs tables à la fois** (Intersection+Projection+sélection) :

Sélection : Garder quelques lignes à l'affichage selon des critères →→→

SELECT ... **FROM** ...
WHERE..IN
 (SELECT ... FROM ...);

Mots Clés du langage SQL	 Signification du mot clé
SELECT	Pour mentionner le(s) nom(s) des colonne(s) demandées à l'affichage
FROM	Pour mentionner le nom de la première table qui contient les colonnes demandées
WHERE	Pour mentionner la clé primaire à retrouver dans la deuxième table (pour faire l'intersection entre clés primaire et étrangère) CléPrimaire IN (SELECT CléEtrangère FROM ...);
;	



F_Inter_01 : Afficher les noms des articles en stock :
NOMART

```
SELECT ARTICLES.NOMART  
FROM ARTICLES  
WHERE ARTICLES.REFART ← Clé primaire  
IN  
(SELECT STOCKS.REFART ← Clé étrangère  
FROM STOCKS)  
;
```

```
SELECT NOMART FROM ARTICLES  
WHERE REFART  
IN  
(SELECT REFART FROM STOCKS)  
;
```

F_Inter_02 : Afficher les noms des articles en stock ainsi que leurs prix unitaires :

NOMART, PUART


```
SELECT ARTICLES.NOMART ,  
ARTICLES.PUART  
FROM ARTICLES  
WHERE ARTICLES.REFART  
IN  
(SELECT STOCKS.REFART  
FROM STOCKS)  
;
```

```
SELECT NOMART, PUART FROM ARTICLES  
WHERE REFART  
IN  
(SELECT REFART FROM STOCKS)  
;
```

F_Differ : Exemples de requêtes sur **plusieurs tables à la fois** (Différence+Projection+sélection) :

Sélection : Garder quelques lignes à l'affichage selon des critères →→→

SELECT ... **FROM** ...
WHERE...NOT IN
 (SELECT ... FROM ...);

Mots Clés du langage SQL	 Signification du mot clé
SELECT	Pour mentionner le(s) nom(s) des colonne(s) demandées à l'affichage
FROM	Pour mentionner le nom de la première table qui contient les colonnes demandées
WHERE	Pour mentionner la clé primaire à ne pas retrouver dans la deuxième table (pour faire la différence entre clés primaire et étrangère) CléPrimaire NOT IN (SELECT CléEtrangère FROM ...);
;	



F_Differ_01 : Afficher les noms des articles qui ne sont pas en stock :

NOMART

```
SELECT ARTICLES.NOMART
FROM ARTICLES
WHERE ARTICLES.REFART ← Clé primaire
NOT IN
(SELECT STOCKS.REFART ← Clé étrangère
FROM STOCKS)
;
```

```
SELECT NOMART FROM ARTICLES
WHERE REFART
NOT IN
(SELECT REFART FROM STOCKS)
;
```



G01 : Calculer/Chercher la valeur actuelle (le coût) du stock :

SUM(ARTICLES.PUART*STOCKS.QTESTK)

```
SELECT  
SUM(ARTICLES.PUART*STOCKS.QTESTK) AS V  
FROM  
ARTICLES INNER JOIN STOCKS  
ON ARTICLES.REFART = STOCKS.REFART;
```

```
SELECT  
SUM(ARTICLES.PUART*STOCKS.QTESTK) AS V  
FROM ARTICLES , STOCKS  
WHERE ARTICLES.REFART = STOCKS.REFART;
```

