

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Chapitre 4.4

Langage sql

Langage d'interrogation de données

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

L'interrogation des données se fera à partir du modèle relationnel ci-dessous.

EMPLOYE (mat, nom, fonction, dateEmbauche, salaire, commission, numServEmploye)

clé primaire : mat

clé étrangère : numServEmploye en référence à num de la relation Service

SERVICE (num, nom, lieu)

clé primaire : num

clé étrangère : num en référence à numServEmploye de la relation EMPLOYE

Relation : EMPLOYE

mat	nom	fonction	dateEmbauche	salaire	commission	numServEmploye
26691	DUPONT	commercial	14/04/1992	2 500	300	20
15155	GARDARIN	ingénieur	07/08/1998	2 400	0	10
16712	MARTIN	commercial	23/05/1990	3 000	1 000	30
17574	DUPONT		05/03/1995	900	0	30

Relation : SERVICE

num	nom	lieu
10	Recherche	Batiment A
20	Vente	Batiment B
30	Direction	Batiment C
40	Fabrication	Batiment D - E
50	Service après ventes	Batiment F

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

1° Interrogation d'une table simple

1.1 Projection

L'instruction **SELECT** recense les champs à projeter à l'aide de tables précisées dans la clause **FROM**.

Syntaxe	
SELECT	nomChamp1, nomChamp2, ... * si tous les champs de la table sont projetés
FROM	nomTable1, nomTable2, ... ; (le point virgule (;) signale la fin de la requête)

Exemple 1

Quels sont les différents services (nom) de l'entreprise et où sont-ils implantés ?

```
SELECT nom, lieu
FROM SERVICE ;
```

nom	lieu
Recherche	Batiment A
Vente	Batiment B
Direction	Batiment C
Fabrication	Batiment D – E
Service après ventes	Batiment F

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Exemple 2

Toutes les informations sur les employés de l'entreprise

```
SELECT * (projette tous les champs de la table EMPLOYE)
FROM EMPLOYE ;
```

mat	nom	fonction	dateEmbauche	salaire	commission	numServEmploye
15155	GARDARIN	ingénieur	1998-08-07	2400.00	0.00	10
16712	MARTIN	commercial	1990-05-23	3000.00	1000.00	30
17574	DUPONT	NULL	1995-03-05	900.00	0.00	30
26691	DUPONT	commercial	1992-04-14	2500.00	300.00	20

L'instruction SELECT est associée au mot-clé **ALL** par défaut (et de ce fait jamais spécifié), ce qui signifie que tous les enregistrements y compris ceux identiques, sont projetés.

La clause **DISTINCT** permet de ne projeter que les enregistrements différents.

Exemple 3

Quelles sont les fonctions occupées dans l'entreprise ?

```
SELECT DISTINCT fonction (projette tous les champs de la table EMPLOYE)
FROM EMPLOYE ;
```

fonction
ingénieur
commercial
NULL

Plusieurs employés pouvant occuper la même fonction, la clause **DISTINCT** permet de n'afficher qu'une seule fois chaque fonction.

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

1.2 Restriction de lignes

La clause **WHERE** permet de spécifier des critères de restriction. Elle est suivie d'une condition de sélection qui sera évaluée pour chaque enregistrement de la table. Seuls les enregistrements pour lesquels le résultat, booléen, de la condition de sélection est VRAI seront projetés.

Une condition de sélection est construite à partir :

- d'opérateurs arithmétiques (+, -, *, /)
- d'opérateurs de comparaison : =, >, <, <=, >=, <>(différent) ou != dans certains SGBDR
- d'opérateurs SQL :
 - ✓ BETWEEN ⇒ BETWEEN valeur1 AND valeur2 (intervalle entre valeur1 ET valeur2)
 - ✓ AND ⇒ ET
 - ✓ OR ⇒ OU
 - ✓ IN ⇒ champ IN (valeur1, valeur2,...) ⇒ recherche des valeurs du champ égales à l'une des valeurs entre parenthèses
 - ✓ LIKE ⇒ champ LIKE (chaine) ⇒ valeur égale à la valeur de chaine, qui peut contenir des caractères jokers ('_' pour un seul caractère, '%' pour plusieurs caractères)

Syntaxe	
SELECT	nomChamp1, nomChamp2,...
FROM	nomTable1, nomTable2, ...
WHERE	conditionDeSelection ;

Exemple 4

Les noms des salariés qui occupent la fonction de "commercial"?	
SELECT	nom
FROM	EMPLOYE
WHERE	fonction='commercial';
	

Le critère de restriction de type texte est placé entre quotes ⇒ '.

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Exemple 5:

Les noms des salariés embauchés avant 1993.

```
SELECT    nom
FROM      EMPLOYE
WHERE     dateEmbauche <= '1993-01-01';
```

nom
MARTIN
DUPONT

Le critère "avant 1993" est exprimé sous forme de date (dans Mysql les dates utilisent le forma aaaa-mm-jj). La date est placée entre quotes ⇒ '.

Exemple 6

Les noms des salariés dont la rémunération (salaire + commission) dépasse 3 500 €.

```
SELECT    nom
FROM      EMPLOYE
WHERE     (salaire + commission) > 3500 ;
```

nom
MARTIN

Une valeur numérique n'est pas placée entre quotes.

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Exemple 7 (clause BETWEEN)

Les noms des salariés embauchés en 1998.

```
SELECT nom
FROM EMPLOYE
WHERE dateEmbauche BETWEEN '1998-01-01' AND '1998-12-31';
```

nom
GARDARIN

Exemple 8

Les noms des salariés dont la commission représente plus de 20% du salaire.

```
SELECT nom
FROM EMPLOYE
WHERE commission > (salaire * 0.2);
```

nom
MARTIN

Exemple 9 (clause IN)

Les noms et fonctions des salariés qui sont soit commercial, soit ingénieur.

```
SELECT nom, fonction
FROM EMPLOYE
WHERE fonction IN ('commercial', 'ingénieur');
```

nom	fonction
GARDARIN	ingénieur
MARTIN	commercial
DUPONT	commercial

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

(clause OR)

Les noms et fonctions des salariés qui sont soit commercial, soit ingénieur.

```
SELECT nom, fonction
FROM EMPLOYE
WHERE fonction = 'commercial' OR fonction = 'ingénieur' ;
```

nom	fonction
GARDARIN	ingénieur
MARTIN	commercial
DUPONT	commercial

Exemple 10 (clause LIKE)

Les noms des salariés dont le nom commence par M.

```
SELECT nom
FROM EMPLOYE
WHERE nom LIKE 'M%' ;
```

nom	fonction
MARTIN	commercial

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Exemple 11 :

Il est possible de renommer, temporairement pour une requête, une expression, un champ ou une table à l'aide de la clause facultative **AS** et d'un alias (entre quotes s'il comporte des espaces).

Le nom et la rémunération brute (salaire + commission) des salariés.

```
SELECT nom, (salaire + commission) AS remunerationBrute
FROM EMPLOYE ;
```

nom	remunerationBrute
GARDARIN	2400.00
MARTIN	4000.00
DUPONT	900.00
DUPONT	2800.00

Exemple 12

Les conditions de sélection composées combinent plusieurs conditions de sélection simples grâce aux opérateurs logiques :

- ✓ **AND** (et),
- ✓ **OR** (ou inclusif : l'un ou l'autre, ou les deux),
- ✓ **XOR** (ou exclusif : ou l'un ou l'autre mais pas les deux).

L'opérateur **NOT** placé devant une condition de sélection en inverse le sens.

Des parenthèses peuvent être utilisées pour imposer une priorité dans l'évaluation de la condition de sélection, ou tout simplement pour rendre plus claire l'expression logique.

Les noms des salariés du service 30, ayant un salaire brut d'au moins 4 000 €.

```
SELECT nom
FROM EMPLOYE
WHERE numServEmploye = 30
AND salaire >= 4000 ;
```

nom
MARTIN

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Exemple 13

Les noms des salariés qui sont soit commercial travaillant dans le service 20, soit ingénieur..

```

SELECT    nom
FROM      EMPLOYE
WHERE     fonction 'commercial' AND numServEmploye = 20
OR        fonction = 'ingénieur' ;

```

nom
GARDARIN
DUPONT

Exemple 14

Les noms des salariés qui sont commercial ou ingénieur, et travaillant dans le service 20.

```

SELECT    nom
FROM      EMPLOYE
WHERE     fonction IN ('commercial', 'ingénieur')
AND       numServEmploye = 20 ;

```

nom
DUPONT

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Exemple 15

Une valeur **NULL** est une valeur non renseignée (vide) pour un champ où cela est autorisé. La valeur NULL est différente de la valeur zéro ou d'un texte vide (' ') qui sont des valeurs définies.

L'opérateur **IS NULL** permet de tester la valeur NULL.

L'expression de la forme **NULL+ valeur** renvoie NULL comme résultat quelle que puisse être la valeur.

Les noms des salariés qui n'ont pas encore de fonction.

```
SELECT    nom
FROM      EMPLOYE
WHERE     fonction IS NULL ;
```

nom
DUPONT

Exemple 16

Les noms des salariés qui ont obtenu une commission.

```
SELECT    nom
FROM      EMPLOYE
WHERE     commission != 0 ;
```

nom
MARTIN
DUPONT

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

1.3 Trier le résultat d'une interrogation

Les enregistrements projetés par l'instruction **SELECT** peuvent être triés par la clause **ORDER BY**.

L'ordre des tris a lieu selon l'ordre des champs ou des alias spécifiés dans la clause **ORDER BY**.

Pour chaque critère de tri, le classement peut être croissant (**ASC** par défaut) ou décroissant (**DESC** à spécifier).

Quel que soit le sens de tri, des valeurs **NULL** sont projetées en premier.

Syntaxe	
SELECT	nomChamp1, nomChamp2, nomChamp3...
FROM	nomTable1, nomTable2, ...
ORDER BY	nomChamp2 DESC , nomChamp3 ASC ;

Tri par ordre décroissant du nomChamp2 et enfin par ordre croissant du nomChamp3. Pour l'ordre croissant, la clause **ASC** est facultative puisque c'est la valeur par défaut.

Exemple17

Les noms des salariés, classés par fonction puis par salaire décroissant.

```
SELECT nom, fonction, salaire
FROM EMPLOYE
ORDER BY fonction, salaire DESC ;
```

nom	fonction	salaire
DUPONT	NULL	900.00
MARTIN	commercial	3000.00
DUPONT	commercial	2500.00
GARDARIN	ingénieur	2400.00



Un tri ne peut se faire que sur un champ projeté par **SELECT**.

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Exemple 18

Les noms des salariés touchant une commission, classés par rémunération brute (salaire + commission) décroissante.

```
SELECT nom, (salaire + commission) AS remunerationBrute
FROM EMPLOYE
WHERE commission != 0
ORDER BY remunerationBrute DESC ;
```

nom	remunerationBrute
MARTIN	4000.00
DUPONT	2800.00

2° Les jointures

2.1 Principe

Les jointures permettent de mettre en relation plusieurs tables.

L'établissement d'une jointure est nécessaire dès lors que la projection rend nécessaire la présence de plusieurs tables dans la clause FROM. Il faut alors spécifier les critères de jointure à l'aide de la clause **WHERE**.

Lorsque plusieurs tables sont utilisées, il est nécessaire de préfixer par le nom de table (nomTable.nomChamp), les champs pour lesquels une ambiguïté existerait (deux champs portant le même nom dans deux tables différentes).

On peut éviter le préfixage en spécifiant des noms différents pour tous les champs lors de la conception du schéma relationnel, y compris pour les clés étrangères, en leur donnant un nom significatif.

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

2.2 Exemples d'une jointure

Exemple 19

Afficher le nom des salariés avec le service dans lequel il travaille.

```
SELECT EMPLOYE.nom, SERVICE.nom
FROM EMPLOYE, SERVICE
WHERE EMPLOYE.numServEmploye = SERVICE.num ;    (critère de jointure)
```

nom	nom
GARDARIN	Recherche
DUPONT	Vente
MARTIN	Direction
DUPONT	Direction

La jointure est matérialisée par le critère « EMPLOYE.numServEmploye = SERVICE.num », celle-ci a pour but de relier « physiquement » la table EMPLOYE avec la table SERVICE.

À chaque enregistrement de la table EMPLOYE sera associé l'enregistrement de la table SERVICE pour lequel numServEmploye sera égal à num.

Exemple 20

Afficher tous les champs des salariés avec le lieu où ils travaillent.

```
SELECT EMPLOYE.*, SERVICE.lieu
FROM EMPLOYE, SERVICE
WHERE EMPLOYE.numServEmploye = SERVICE.num ;
```

mat	nom	fonction	dateEmbauche	salaire	commission	numServEmploye	lieu
15155	GARDARIN	ingénieur	1998-08-07	2400.00	0.00	10	Batiment A
26691	DUPONT	commercial	1992-04-14	2500.00	300.00	20	Batiment B
16712	MARTIN	commercial	1990-05-23	3000.00	1000.00	30	Batiment C
17574	DUPONT	NULL	1995-03-05	900.00	0.00	30	Batiment C

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

2.2 Exemple d'une jointure oubliée

En l'absence de critère de jointure, le nombre de résultats projetés sera égal au produit cartésien des enregistrements sélectionnés des deux tables.

Exemple 21

Afficher tous les champs des salariés avec le lieu où ils travaillent (ne pas mettre de jointure).

```
SELECT EMPLOYE.*, SERVICE.lieu
FROM EMPLOYE, SERVICE ;
```

Oubli volontaire du critère de jointure

mat	nom	fonction	dateEmbauche	salaire	commission	numServEmploye	nom
15155	GARDARIN	ingénieur	1998-08-07	2400.00	0.00	10	Recherche
16712	MARTIN	commercial	1990-05-23	3000.00	1000.00	30	Recherche
17574	DUPONT	NULL	1995-03-05	900.00	0.00	30	Recherche
26691	DUPONT	commercial	1992-04-14	2500.00	300.00	20	Recherche
15155	GARDARIN	ingénieur	1998-08-07	2400.00	0.00	10	Vente
16712	MARTIN	commercial	1990-05-23	3000.00	1000.00	30	Vente
17574	DUPONT	NULL	1995-03-05	900.00	0.00	30	Vente
26691	DUPONT	commercial	1992-04-14	2500.00	300.00	20	Vente
15155	GARDARIN	ingénieur	1998-08-07	2400.00	0.00	10	Direction
16712	MARTIN	commercial	1990-05-23	3000.00	1000.00	30	Direction
17574	DUPONT	NULL	1995-03-05	900.00	0.00	30	Direction
26691	DUPONT	commercial	1992-04-14	2500.00	300.00	20	Direction
15155	GARDARIN	ingénieur	1998-08-07	2400.00	0.00	10	Fabrication
16712	MARTIN	commercial	1990-05-23	3000.00	1000.00	30	Fabrication
17574	DUPONT	NULL	1995-03-05	900.00	0.00	30	Fabrication
26691	DUPONT	commercial	1992-04-14	2500.00	300.00	20	Fabrication
15155	GARDARIN	ingénieur	1998-08-07	2400.00	0.00	10	Service après ventes
16712	MARTIN	commercial	1990-05-23	3000.00	1000.00	30	Service après ventes
17574	DUPONT	NULL	1995-03-05	900.00	0.00	30	Service après ventes
26691	DUPONT	commercial	1992-04-14	2500.00	300.00	20	Service après ventes

La table EMPLOYE comporte 4 enregistrements (donc 4 employés), la table SERVICE en compte 5. Le résultat juste devrait faire apparaître 4 enregistrements.

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder



L'oubli du critère de jointure va donc fausser le résultat. Le produit cartésien des deux tables sera affiché, c'est-à-dire $4 * 5 = 20$ (toutes les combinaisons possibles pour chaque employé) alors même qu'un employé n'appartient qu'à un seul service

3° Les fonctions d'agrégat SQL

3.1 Liste des fonctions d'agrégat sql

Fonction d'agrégat SQL	Description
SUM(nomChamp)	Somme
AVG(nomChamp)	Moyenne
MIN(nomChamp)	Minimum
MAX(nomChamp)	Maximum
COUNT(nomChamp)	Compte le nombre d'enregistrements d'une sélection (sur une clé, en général)

Les fonctions d'agrégat **SUM**, **AVG**, **MIN** et **MAX** portent sur des valeurs numériques ou date/heure.

3.2 Distinction entre une opération avec fonction d'agrégat et calcul

Il est possible d'introduire des calculs dans l'instruction SELECT ou dans les clauses WHERE ou ORDER BY mais il s'agit alors de calculs répétés de manière indépendante pour chaque enregistrement (calcul en ligne).

Une opération avec fonction d'agrégat est au contraire réalisée « en colonne » et concerne simultanément tous les enregistrements issus des jointures et restrictions.

Une opération avec fonction d'agrégat effectuée sans regroupement délivrera un seul enregistrement-résultat.

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

3.3 Exemples d'utilisation des fonctions d'agrégat

Exemple 22

Déterminer le salaire moyen des employés de la société

```
SELECT    AVG(salaire) AS salaireMoyen
FROM      EMPLOYE ;
```

salaireMoyen 2200.000000

La clause **AS** permet de nommer le champ qui affichera le résultat de AVG(salaire).

Exemple 23

Indiquer la date d'embauche du salarié du service 30 ayant le plus d'ancienneté.

```
SELECT    MIN(EMPLOYE.dateEmbauche)
FROM      EMPLOYE
WHERE     .numServEmploye = 30 ;
```

MIN(dateEmbauche) 1990-05-23
--

Exemple 24

Calculer l'ancienneté en années du salarié en poste depuis le plus longtemps

```
SELECT    ( YEAR(CURDATE()) - MAX (YEAR(dateEmbauche)) ) AS salarieLePlusAncien
FROM      EMPLOYE ;
```

salarieLePlusAncien 13

CURDATE() ⇒ Renvoie la date du jour

YEAR(CURDATE()) ⇒ Renvoie l'année de la date du jour (2012 pour le 03/01/2012)

YEAR(nomChamp) ⇒ Renvoie l'année du champ sélectionné

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Exemple 25

Calculer le montant total des commissions du service 30.

```
SELECT SUM(commission) AS totalCommissionsService30
FROM EMPLOYE
WHERE numServEmploye = 30 ;
```

totalCommissionsService30 1000.00

Exemple 26

Déterminer le nombre de membres du service 30.

```
SELECT COUNT(mat)
FROM EMPLOYE
WHERE numServEmploye = 30 ;
```

COUNT(mat) 2

Exemple 27

Calculer le montant total des commissions du service Vente.

```
SELECT SUM(commission) AS totalCommissionsServiceVente
FROM EMPLOYE, SERVICE
WHERE EMPLOYE.numServEmploye = SERVICE.num
AND SERVICE.nom = 'vente' ;
```

totalCommissionsServiceVente 300.00

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Exemple 28

Calculer le montant total des rémunérations brutes.

```
SELECT SUM(salaire + commission) AS totalRemunerationsBrutes
FROM EMPLOYE ;
```

totalRemunerationsBrutes 10100.00

Exemple 29

Déterminer le salaire le plus élevé de l'entreprise.

```
SELECT MAX(salaire)
FROM EMPLOYE ;
```

MAX(salaire) 3000.00

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

4° Le regroupement des résultats

Les opérations avec fonction d'agrégat exécutées dans l'instruction SELECT délivrent un seul enregistrement-résultat. La clause **GROUP BY** permet d'effectuer des opérations avec fonction d'agrégat par groupes d'enregistrements et donc de délivrer plusieurs enregistrements-résultat.

Exemple 30

Indiquer le nombre de salariés par service.

```
SELECT SERVICE.nom, COUNT(EMPLOYEE.mat)
FROM EMPLOYEE, SERVICE
WHERE EMPLOYEE.numServEmploye = SERVICE.num
GROUP BY Service.nom ;
```

nom	COUNT(EMPLOYEE.mat)
Direction	2
Recherche	1
Vente	1

Exemple 31

Indiquer le nombre de salariés par lieu.

```
SELECT SERVICE.lieu, COUNT(mat) AS nombreDeSalariesParLieu
FROM EMPLOYEE, SERVICE
WHERE EMPLOYEE.numServEmploye = SERVICE.num
GROUP BY SERVICE.lieu ;
```

nom	COUNT(EMPLOYEE.mat)
Direction	2
Recherche	1
Vente	1

Le nom du service doit être affiché pour permettre l'interprétation des résultats affichés.

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Exemple 32

Classer en ordre décroissant les salaires moyens par service.

```
SELECT SERVICE.nom, AVG(EMPLOYE.salaire) AS salaireMoyen
FROM EMPLOYE, SERVICE
WHERE EMPLOYE.numServEmploye = SERVICE.num
GROUP BY SERVICE.nom
ORDER BY salaireMoyen DESC ;
```

nom	salaireMoyen
Vente	2500.000000
Recherche	2400.000000
Direction	1950.000000

L'utilisation possible de la fonction d'agrégat ou de son alias dans la clause **ORDER BY** dépend du SGBDR.

Exemple 33

La clause **HAVING** permet une restriction d'affichage sur les résultats d'une opération avec fonction d'agrégat. Elle ne doit pas être confondue avec la clause **WHERE** qui induira une restriction sur les enregistrements auxquels appliquer l'opération. **HAVING** intervient après l'opération (après les clauses **WHERE** et **GROUP BY**).

Afficher les salaires moyens des services pour lesquels ce salaire moyen dépasse 2 000 €.

```
SELECT SERVICE.nom, AVG(EMPLOYE.salaire) AS salaireMoyen
FROM EMPLOYE, SERVICE
WHERE EMPLOYE.numServEmploye = SERVICE.num
GROUP BY SERVICE.nom
HAVING salaireMoyen > 2000 ;
```

nom	salaireMoyen
Recherche	2400.000000
Vente	2500.000000

Question 4	Comment peut-on produire de l'information à partir de données contenues dans une table ?	
Chapitre 4.4	<i>Base de données sql</i>	Bac STMG SIG
Cours	<i>Langage d'interrogation de données</i>	Lycée J. Feyder

Exemple 34

Calculer les salaires moyens par services des employés dont le salaire dépasse 2000 € (moyenne des « hauts salaires »).

```

SELECT SERVICE.nom, AVG(EMPLOYE.salaire) AS moyenneDesHautsSalaires
FROM EMPLOYE, SERVICE
WHERE EMPLOYE.numServEmploye = SERVICE.num
AND EMPLOYE.salaire > 2000
GROUP BY SERVICE.nom ;

```

nom	moyenneDesHautsSalaires
Direction	3000.000000
Recherche	2400.000000
Vente	2500.000000