

## Corrigés EXERCICES SQL 1ère partie

### Extraire des données

1) La commande SELECT suivante est exécutée avec succès :

```
SELECT last_name, job_id, salary AS Sal  
FROM employees;
```

**Vrai**

2) L'instruction SELECT suivante est exécutée avec succès :

```
SELECT *  
FROM job_grades;
```

**Faux** la table job\_grades n'existe pas

3) L'instruction suivante présente quatre erreurs de codage. Pouvez-vous les identifier ?

```
SELECT employee_id, last_name  
sal x 12 ANNUAL SALARY  
FROM employees;
```

Les 4 erreurs

1. Il manque une virgule après last\_name
2. la table employees n'a pas de colonne nommée sal
3. l'opérateur de multiplication est \* et non x
4. l'alias ANNUAL SALARY contient une espace et doit être entourée de guillemets

4) Déterminer la structure de la table Departments

```
DESCRIBE departments
```

5) Afficher les données de la table DEPARTMENTS :

```
SELECT * FROM departments ;
```

6) Affichez la structure de la table EMPLOYEES.

```
DESCRIBE employees
```

7) Affichez le nom, l'ID de poste, la date d'embauche et l'ID d'employé de chaque employé, l'ID d'employé apparaissant en premier. Associez l'alias STARTDATE à la colonne HIRE\_DATE.

```
SELECT employee_id, last_name, job_id, hire_date StartDate  
FROM employees;
```

8) Affichez tous les ID de poste unique de la table EMPLOYEES.

```
SELECT DISTINCT job_id  
FROM employees ;
```

9) Affichez le nom concaténé avec l'ID de poste (en séparant les deux par une virgule et un espace) et intitulez la colonne Employee and Title.

```
SELECT last_name || ', ' || job_id "Employee and Title"  
FROM employees ;
```

## Restreindre et trier les données

1. Affichez le nom et le salaire des employés qui gagnent plus de 12000

```
SELECT last_name, salary
FROM employees
WHERE salary > 12000;
```

2. Créez un état affichant le nom et le numéro de département correspondant à l'ID d'employé 176.

```
SELECT last_name, department_id
FROM employees
WHERE employee_id=176;
```

3. Affichez les noms, salaires des employés qui gagnent entre 6000 et 10000

```
SELECT last_name, salary
FROM employees
WHERE salary BETWEEN 6000 AND 10000;
```

4. Affichez le nom, l'ID de poste et la date d'embauche des employés nommés Matos et Taylor. Triez les données par ordre croissant en fonction de la date d'embauche.

```
SELECT last_name, job_id, hire_date
FROM employees
WHERE last_name IN ('Matos', 'Taylor')
ORDER BY hire_date;
```

5. Affichez le nom et le numéro de département de tous les employés du département 20 ou 50 par ordre alphabétique croissant, en fonction du nom.

```
SELECT last_name, department_id
FROM employees
WHERE department_id IN (20, 50)
ORDER BY last_name ASC;
```

6. Affichez le nom et le salaire des employés qui gagnent entre 5000 et 12000, et travaillent dans le département 20 ou 50.

Intitulez respectivement les colonnes Employee et Monthly Salary. Enregistrez

```
SELECT last_name "Employee", salary "Monthly Salary"
FROM employees
WHERE salary BETWEEN 5000 AND 12000
AND department_id IN (20, 50);
```

7. Affichez le nom et la date d'embauche de tous les employés embauchés en 1994.

```
SELECT last_name, hire_date
FROM employees
WHERE hire_date LIKE '%94';
```

8. Créez un état affichant le nom et l'intitulé de poste de tous les employés qui n'ont pas de manager.

```
SELECT last_name, job_id
FROM employees
WHERE manager_id IS NULL;
```

9. Affichez le nom et le salaire des employés qui gagnent plus qu'un montant saisi par l'utilisateur en réponse à une invite.

```
SELECT last_name, salary
```

```
FROM employees
WHERE salary > &montant;
```

10. Affichez le nom de tous les employés dont le nom comporte un "a" et un "e".

```
SELECT last_name
FROM employees
WHERE last_name LIKE '%a%'
AND last_name LIKE '%e%';
```

## Fonctions monolignes

1. Ecrivez une interrogation permettant d'afficher la date système. Intitulez la colonne Date.

```
SELECT sysdate "Date"
FROM dual;
```

2. Afficher le numéro d'employé, le nom, le salaire et le salaire augmenté de 15,5 % (exprimé sous la forme d'un nombre entier) pour chaque employé. Nommez la colonne *New Salary*

```
SELECT employee_id, last_name, salary,
ROUND(salary * 1.155, 0) "New Salary"
FROM employees;
```

3. Ecrivez une interrogation permettant d'afficher le nom (la première lettre en majuscule et toutes les autres lettres en minuscules) et la longueur du nom de tous les employés dont le nom commence par les lettres "J", "A" ou "M". Attribuez à chaque colonne un libellé approprié. Triez les résultats en fonction du nom des employés.

```
SELECT INITCAP(last_name) "Name", LENGTH(last_name) "Length"
FROM employees
WHERE last_name LIKE 'J%'
OR last_name LIKE 'M%'
OR last_name LIKE 'A%'
ORDER BY last_name ;
```

4. Créez une interrogation permettant d'afficher le nom et le salaire de tous les employés. Formatez le salaire de sorte qu'il contienne 15 caractères et soit complété à gauche par le symbole \$. Intitulez la colonne SALARY.

```
SELECT last_name,
LPAD(salary, 15, '$') SALARY
FROM employees;
```

5. Créez une interrogation permettant d'afficher le nom et le salaire de tous les employés. Formatez le salaire de sorte qu'il contienne 15 caractères et soit complété à gauche par le symbole \$. Intitulez la colonne SALARY

```
SELECT last_name,
LPAD(salary, 15, '$') SALARY
FROM employees;
```

## Fonctions de conversion et des expressions conditionnelles

1. Affichez le nom, la date d'embauche et le jour de la semaine où l'employé a commencé. Intitulez la colonne DAY. Triez les résultats en fonction de la date

```
SELECT last_name, hire_date,  
TO_CHAR(hire_date, 'DAY') DAY  
FROM employees  
ORDER BY hire_date ;
```

2. Créez une interrogation qui affiche le nom et le montant de la commission de chaque employé. Si un employé ne perçoit pas de commission, indiquez "No Commission". Intitulez la colonne COMM.

```
SELECT last_name,  
NVL(TO_CHAR(commission_pct), 'No Commission') COMM  
FROM employees;
```

3. A l'aide de la fonction DECODE, écrivez une interrogation qui affiche le niveau de tous les employés sur la base de la valeur de la colonne JOB\_ID, à l'aide des données suivantes :

Job	Grade
AD_PRES	A
ST_MAN	B
IT_PROG	C
SA_REP	D
ST_CLERK	E
<i>Sinon</i>	0

```
SELECT job_id, decode (job_id,  
'ST_CLERK', 'E',  
'SA_REP', 'D',  
'IT_PROG', 'C',  
'ST_MAN', 'B',  
'AD_PRES', 'A',  
'0')GRADE  
FROM employees
```

4. Réécrivez l'instruction dans l'exercice précédent à l'aide de la syntaxe CASE.

```
SELECT job_id, CASE job_id  
WHEN 'ST_CLERK' THEN 'E'  
WHEN 'SA_REP' THEN 'D'  
WHEN 'IT_PROG' THEN 'C'  
WHEN 'ST_MAN' THEN 'B'  
WHEN 'AD_PRES' THEN 'A'  
ELSE '0' END GRADE  
FROM employees;
```

## Fonctions de groupe

1. Les fonctions de groupe opèrent sur plusieurs lignes et produisent un résultat par groupe.

Vrai/Faux

**VRAI**

2. Les fonctions de groupe prennent en compte les valeurs NULL dans les calculs.

Vrai/Faux

**FAUX**

3. La clause WHERE limite les lignes avant inclusion dans un calcul de groupe.

Vrai/Faux

**VRAI**

4..Déterminez le salaire le plus élevé, le salaire le plus bas, le salaire cumulé et le salaire moyen pour tous les employés. Intitulez respectivement les colonnes Maximum, Minimum, Sum et Average. Arrondissez les résultats à l'entier le plus proche

```
SELECT ROUND(MAX(salary),0) "Maximum",  
ROUND(MIN(salary),0) "Minimum",  
ROUND(SUM(salary),0) "Sum",  
ROUND(AVG(salary),0) "Average"  
FROM employees;
```

5. Ecrivez une interrogation permettant d'afficher le nombre de personnes occupant le même poste.

```
SELECT job_id, COUNT(*)  
FROM employees  
GROUP BY job_id;
```

6. Déterminez le nombre de managers sans répertorier ceux-ci. Intitulez la colonne "Number of Managers".

```
SELECT COUNT(DISTINCT manager_id) "Number of Managers"  
FROM employees;
```

7. Trouvez la différence entre le salaire le plus élevé et le salaire le plus bas. Intitulez la colonne DIFFERENCE.

```
SELECT MAX(salary) - MIN(salary) DIFFERENCE  
FROM employees ;
```