

Les commandes SQL (Exercice page 9-10-11-12)

Restreindre et trier les données :

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

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

2. Etat 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. Les noms, salaires des employés qui gagnent entre 6000 et 10000

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

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

```
SELECT last_name, job_id, hire_date  
FROM employees  
WHERE last_name='Matos' or last_name='Taylor';
```

5. 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=20 OR department_id=50
ORDER BY last_name ASC;
```

6. Le nom et le salaire des employés qui gagnent entre 5000 et 12000, et travaillent dans le département 20 ou 50. Intitule respective des colonnes Employee et Monthly Salary

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

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

```
SELECT last_name, hire_date
FROM employees
WHERE hire_date= 'DD/MM/94';
```

8. Création d'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. 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. 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%'
      OR last_name LIKE '%e%';
```

Fonctions monolignes :

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

```
SELECT sysdate "DATE"
FROM employees;
```

2. 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. Nommons la colonne New Salary

```
SELECT employee_id, last_name, salary, salary+(salary*155)/1000 "New Salary"
FROM employees ;
```

3. 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". Attribution à chaque colonne un libellé approprié. Trie des résultats en fonction du nom des employés.

```
SELECT last_name, LENGTH('last_name') AS LENGHT
FROM employees
WHERE last_name LIKE 'J%'
      OR last_name LIKE 'A%'
      OR last_name LIKE 'M%'
ORDER BY last_name;
```

5. Création d' une interrogation permettant d'afficher le nom et le salaire de tous les employés. Formatage de 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 . 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,
      TO_CHAR(hire_date, 'DD/MM/YY')
      AS DAY
FROM employees
ORDER BY hire_date;
```

2. Création d'une interrogation qui affiche le nom et le montant de la commission de chaque employé. Si un employé ne perçoit pas de commission, indiquons "No Commission". Intitule de la colonne COMM.

```
SELECT last_name,  
       DECODE(commission_pct, 'null', 'No Commission') "COMM"  
FROM employees;
```

3.

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

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

```
SELECT job_id,  
       CASE job_id WHEN 'AD_PRES' THEN 'A'  
                 WHEN 'ST_MAN' THEN 'B'  
                 WHEN 'IT_PROG' THEN 'C'  
                 WHEN 'SA_REP' THEN 'D'  
                 WHEN 'ST_CLERK' THEN 'E'  
                 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

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

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

4. Le salaire le plus élevé, le salaire le plus bas, le salaire cumulé et le salaire moyen pour tous les employés. Intitule respective des colonnes Maximum, Minimum, Sum et Average. Arrondissement des résultats à l'entier le plus proche

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

5. Interrogation permettant d'afficher le nombre de personnes occupant le même poste

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

6. Le nombre de managers sans répertorier ceux-ci. Intitule de la colonne "Number of Managers".

```
SELECT COUNT(manager_id) AS "Number of managers"  
FROM employees ;
```

7. La différence entre le salaire le plus élevé et le salaire le plus bas. Intitulons la colonne DIFFERENCE

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