
EXERCICE SUR COMMANDES SQL (EXERCICE PAGE 9-10-11-12-13-14-15)

RESTREINDRE ET TRIER DES DONNEES

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

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

LAST_NAME	SALARY
King	24000
Kochhar	17000
De Haan	17000
Greenberg	12008
Russell	14000
Partners	13500
Hartstein	13000
Higgins	12008

+ 8 lignes sélectionnées.

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

```
SELECT department_name, department_id
FROM departments
WHERE department_id = 176;
```

NB : Cette numéro de département n'existe, donc aucune ligne sélectionnée !

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

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

NB: On a 42 lignes sélectionnées !

4. Affichons le nom, l'ID de post et la date d'embauche de Matos et Taylor tel que la date d'embauche est en ordre croissant :

```
SELECT last_name, job_id, hire_date
FROM employees
WHERE last_name = 'Matos'
      OR last_name = 'Taylor'
ORDER BY hire_date ASC;
```

LAST_NAME	JOB_ID	HIRE_DAT
Taylor	SH_CLERK	24/01/06
Matos	ST_CLERK	15/03/06
Taylor	SA_REP	24/03/06

5. Affichons le nom et le numéro de département de tous les employés du département 20 ou 50 par ordre alphabétique croissant:

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

NB: 47 lignes sélectionnées !

6. Affichons le nom et salaire des employés qui gagnent entre 5000 et 12000 dans le département 20 ou 50 avec colonnes Employee et Monthly Salary :

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

Employee	Monthly Salary
-----	-----
Weiss	8000
Fripp	8200
Kaufling	7900
Vollman	6500
Mourgos	5800
Fay	6000

6 lignes sélectionnées.

7. Affichons 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 'DD/Mon/94';
```

NB : Aucune ligne sélectionnée !

8. Créons 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;
```

LAST_NAME	JOB_ID
-----	-----
King	AD_PRES

9. Affichons le nom et 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. Affichons 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%';

NB : 82 lignes sélectionnées !

FONCTIONS MONOLIGNES

1. Ecriture d'une interrogation pour afficher la date du système avec son intitulation :

```
SELECT sysdate AS "DATE"
```

```
FROM dual;
```

DATE

17/03/22

2. Affichons le numéro d'employé, nom, salaire et salaire augmenté de 15.5% pour chaque employé avec nom New Salary :

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

```
FROM employees;
```

NB : 107 lignes sélectionnées !

3. Ecrivons un interrogation pour afficher le nom (Premier lettre en Majuscule et les en minuscule) et la longueur de nom de tous les employés dont le nom commence par les lettres « J », « A », « M » avec chaque colonne avec des libellé. Avec triage d e nom des employés :

```
SELECT INITCAP (last_name) AS "NAME",
```

```
LENGTH (UPPER(INSTR(last_name, 'A%')))
```

```
FROM employees
```

```
ORDER BY NAME ASC;
```

4. Une interrogation permettant d'afficher le nom et salaire de tous les employés avec format de 15 caractères :

```
SELECT last_name, LPAD(salary,15,'$') AS "SALARY"
```

```
FROM employees;
```

FONCTIONS DE CONVERSION ET DES EXPRESSIONS CONDITIONNELLES

1. Affichons le nom, la date d'embauche et le jours de la semaine où l'employé a commencé avec l'intituler colonne DAY et avec triage en fonction de la date :

```
SELECT last_name, hire_date "Date",  
       TO_CHAR(hire_date, 'DD')  
       AS day  
FROM employees  
ORDER BY hire_date ASC;
```

NB: 107 lignes sélectionnées !

2. Une interrogation qui affiche le nom et montant de la commission de chaque employé, si un employé ne perçoit pas de commission, on indique « No Commission » et la colonne est COMM :

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

NB: 107 lignes sélectionnées !

3. Avec la fonction de DECODE, écrivons une interrogation qui affiche le niveau de tous employés sur la base de valeur de la colonne JOB_ID a l'aide de données donné dans le tableau :

```
SELECT last_name,  
       DECODE (job_id, 'AD_PRES', 'A',  
              'ST_MAN', 'B',  
              'IT_PROG', 'C',  
              'SA_REP', 'D',  
              'ST_CLERK', 'E',  
              0)  
       AS GRADE_EMPLOYER
```

```
FROM employees;
```

NB: 107 lignes sélectionnées !

4. Réécrivons l'instruction avec l'instruction CASE :

```
SELECT last_name,  
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 TO_CHAR (0) END AS GRADE  
FROM employees;
```

FONCTIONS DE GROUPE

1. C'est vrai, les fonctions de groupe opèrent sur plusieurs lignes et produisent un résultat par groupe
2. C'est faux, les fonctions de groupe ne prennent pas en compte les valeurs NULL dans un Calculs.
3. C'est vrai, le clause **WHERE** limite les lignes avant inclusion dans un calculs.
4. Déterminons le salaire le plus élevé, le salaire le plus bas, le salaire cumulé et le salaire moyen pour tous les employés :

```
SELECT MAX(salary) "Maximun",  
MIN(salary) "Minimum",  
SUM(salary) "Sum",  
ROUND(AVG(salary),1)"Average"  
FROM employees;
```

Maximun	Minimum	Sum	Average
24000	2100	691416	6461,8

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

```
SELECT job_id, COUNT (DISTINCT employee_id ) AS "NB EMPLOYE"
FROM employees
GROUP BY job_id;
```

6. Le nombre de manager :

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

Number of Managers
106

7. Trouvons la différence entre le salaire le plus élevé et le salaire le plus le plus bas :

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

DONNEES PROVENANT DE PLUSIEUR TABLE

1. Ecrivons une interrogation produisant l'adresse de tous les départements en utilisez les tables LOCATIONS et COUNTRIES tout en affichant dans les résultats l'ID de lieu, la rue, la ville, le département et le pays :

```
SELECT l.country_id COUNTRIE, l.street_address RUE, l.city CITY, department_name DEP,
c.country_name COUNTRY
FROM countries c
JOIN locations l
ON c.country_id = l.country_id
JOIN departments d
ON d.location_id = l.location_id;
```

2. Une interrogation permettant d'afficher le nom, ainsi que le numéro et le nom de départements, pour tous les employés :

```
SELECT last_name NAME, employee_id NUMERO , department_name DEPARTEMENT  
FROM employees NATURAL JOIN departments ;
```

3. Affichons le nom, le poste, ainsi que le numéro et le nom de département, pour tous les employés qui travaillent à Toronto :

```
SELECT e.last_name NAME, e.job_id POSTE, d.location_id "NUM DEP", d.department_name DEP  
FROM employees e  
JOIN departments d  
ON d.department_id = e.department_id  
JOIN locations l  
ON l.city ='Toronto';
```

4. Créons un état permettant d'afficher le nom et l'ID des employés, ainsi que le nom et l'ID de leur manager. Intitulez respectivement les colonnes Employee, Emp#, Manager et Mgr#.

```
SELECT e.last_name "EMPLOYEE", e.employee_id "EMP#", e.manager_id "MGR#"  
FROM employees e;
```

SOUS-INTERROGATION

1. Ecrivons une interrogation pour afficher le nom et la date d'embauche de tous les employés travaillant dans le même département que l'employé indiqué :

```
SELECT last_name AS "NOM",hire_date "DATE", department_id  
FROM employees  
WHERE department_id =  
    (SELECT department_id  
    FROM employees  
    WHERE last_name = '&&NON')  
AND last_name <> (SELECT last_name  
    FROM employees  
    WHERE last_name='&NON');
```


2. Créons un état qui affiche le numéro d'employé, le nom et le salaire de tous les employés qui gagnent plus que le salaire moyen et trions les résultats par ordre croissant sur la base du salaire :

```
SELECT employee_id "NUM", last_name "NAME", salary
FROM employees
WHERE salary >
(SELECT AVG(salary)
FROM employees);
```

NB : 51 lignes sélectionnées !

3. Une interrogation qui affiche le numéro d'employé et le nom de tous les employés qui travaillent dans un département comprenant un employé dont le nom contient la lettre « u » :
4. Créons un état affichant la liste de tous les employés dont le salaire est plus que le salaire de tout employé du département 60 :

```
SELECT last_name
FROM employees
WHERE salary >
(SELECT SUM(salary)
FROM employees
WHERE department_id=60);
```

OPERATEURS ENSEMBLISTES

1. Affichons les listes des ID des départements qui ne contiennent pas l'ID de poste ST_CLERK en utilisant les opérateurs ensemblistes pour créer cet état

```
SELECT department_id AS "ID"
FROM departments
UNION
SELECT department_id
FROM employees
WHERE job_id <> 'ST_CLERK';
```

NB : 28 lignes sélectionnées !

2. Production de la liste des postes des départements 10, 50 et 20 dans l'ordre donné au tableau :

3. Affichons l'ID et le nom des pays sans départements avec ensembles :

```
SELECT country_id, country_name
```

```
FROM locations
```

```
MINUS
```

```
SELECT country_id, city
```

```
FROM
```

SQL