

## DI

DEVELOPPEUR INFORMATIQUE

### SQL

## Langage de Manipulation de Données

Dossier de travaux pratiques

<u>Module</u>	1
<u>Séquence</u>	6
<u>Capacité</u>	1

Pour chaque requête demandée, vous présenterez:

- L'analyse de la requête à l'aide des opérations relationnelles et des opérations complémentaires utilisées par SQL
- La requête SQL

Les requêtes qui suivent concernent 2 tables que vous devez au préalable décrire et remplir:

Table EMP

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEV-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEV-81	1250	500	30
7566	JONES	MANAGER	7839	02-AVR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAI-81	2850		30
7782	CLARK	MANAGER	7839	09-JUI-81	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

Table DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

## Première partie

1. Donner nom, job, numéro et salaire de tous les employés,  
puis seulement des employés du département 10

```
SELECT * FROM emp
```

```
SELECT empno, ename, job, sal FROM `emp` WHERE deptno = 10
```

2. Donner nom, job et salaire des employés de type MANAGER dont le salaire est supérieur à 2800

```
SELECT ENAME, JOB, SAL FROM `emp` WHERE JOB = "manager" AND SAL > 2800
```

3. Donner la liste des MANAGER n'appartenant pas au département 30

```
SELECT ename, job FROM `emp` WHERE JOB='manager' AND DEPTNO != 30
```

4. Liste des employés de salaire compris entre 1200 et 1400

```
SELECT * FROM `emp` WHERE SAL BETWEEN 1200 AND 1400
```

5. Liste des employés des départements 10 et 30 classés dans l'ordre alphabétique

```
SELECT * FROM `emp` WHERE DEPTNO IN (10, 30) ORDER BY ENAME ASC
```

6. Liste des employés du département 30 classés dans l'ordre des salaires croissants

```
SELECT * FROM `emp` WHERE DEPTNO = 10 ORDER BY SAL ASC
```

7. Liste de tous les employés classés par emploi et salaires décroissants

```
SELECT * FROM `emp` ORDER BY JOB DESC , SAL DESC *****
```

8. Liste des différents emplois

```
SELECT DISTINCT JOB FROM `emp`
```

9. Donner le nom du département où travaille ALLEN

```
SELECT DNAME FROM dept WHERE DEPTNO IN (SELECT DEPTNO FROM emp WHERE ENAME = "ALLEN")
```

```
SELECT DNAME FROM dept WHERE DEPTNO = (SELECT DEPTNO FROM emp WHERE ENAME = "ALLEN")
```

```
SELECT dept.dname FROM emp INNER JOIN dept ON emp.deptno WHERE ename = "ALLEN"
```

10. Liste des employés avec nom du département, nom, job, salaire classés par noms de départements et par salaires décroissants.

```
SELECT dname, ename, job, sal FROM emp, dept WHERE emp.DEPTNO = dept.DEPTNO ORDER BY dname, sal DESC
```

11. Liste des employés vendeurs (SALESMAN) avec affichage de nom, salaire, commissions, salaire + commissions

```
SELECT ename, sal, comm, (sal + comm) AS 'salaire complet' FROM emp WHERE job='salesman'
```

12. Liste des employés du département 20: nom, job, date d'embauche sous forme VEN 28 FEV 1997'

```
SELECT ename, job, HIREDATE, dname FROM emp INNER JOIN dept ON dept.DEPTNO=emp.DEPTNO WHERE emp.DEPTNO=20  
*****
```

13. Donner le salaire le plus élevé par département

```
SELECT MAX(SAL) AS 'Salaire le plus eleve', DNAME FROM emp RIGHT JOIN dept ON dept.DEPTNO = emp.DEPTNO GROUP BY DNAME  
SELECT dname, max(sal) as 'salaire max' from emp, dept where dept.DEPTNO=emp.DEPTNO GROUP by dname
```

14. Donner département par département masse salariale, nombre d'employés, salaire moyen par type d'emploi.

```
SELECT dname, job, SUM(sal)+ SUM(comm) AS 'mass salaire', COUNT(*) AS 'nb employe', AVG(SAL) AS 'sal moyen' FROM emp, dept WHERE  
dept.DEPTNO=emp.DEPTNO GROUP BY dname, job ORDER BY dname
```

\*\*\*\*\*

15. Même question mais on se limite aux sous-ensembles d'au moins 2 employés

```
SELECT dname, job, SUM(sal)+ SUM(comm) AS 'mass salaire', COUNT(*) AS 'nb employe', AVG(SAL) AS 'sal moyen' FROM emp, dept WHERE  
dept.DEPTNO=emp.DEPTNO GROUP BY dname, job HAVING COUNT(*)>=2
```

\*\*\*\*\*

16. Liste des employés (Nom, département, salaire) de même emploi que JONES

```
select ename, deptno, sal from emp where job = (select job from emp where ename ="jones") and ename != 'jones'
```

17. Liste des employés (nom, salaire) dont le salaire est supérieur à la moyenne globale des salaires

```
SELECT ENAME, SAL FROM emp WHERE SAL> (SELECT AVG(SAL) FROM emp)
```

18. Création d'une table PROJET avec comme colonnes numéro de projet (3 chiffres), nom de projet(5 caractères), budget. Entrez les valeurs suivantes:

```
101, ALPHA,    96000  
102, BETA,     82000  
103, GAMMA,    15000
```

19. Ajouter l'attribut numéro de projet à la table EMP et affecter tous les vendeurs du département 30 au projet 101, et les autres au projet 102

20. Créer une vue comportant tous les employés avec nom, job, nom de département et nom de projet

21. A l'aide de la vue créée précédemment, lister tous les employés avec nom, job, nom de département et nom de projet triés sur nom de département et nom de projet

22. Donner le nom du projet associé à chaque manager

## Deuxième partie

1. Afficher la liste des managers des départements 20 et 30

```
SELECT * FROM emp WHERE job = 'manager' AND DEPTNO IN (20 , 30)
```

2. Afficher la liste des employés qui ne sont pas manager et qui ont été embauchés en 81

```
SELECT * FROM `emp` WHERE job != 'manager' and HIREDATE LIKE '1981%'
```

3. Afficher la liste des employés ayant une commission

```
SELECT * FROM `emp` WHERE comm is NOT null AND comm != 0
```

4. Afficher la liste des noms, numéros de département, jobs et date d'embauche triés par Numero de Département et JOB les derniers embauches d'abord.

5. Afficher la liste des employés travaillant à DALLAS

6. Afficher les noms et dates d'embauche des employés embauchés avant leur manager, avec le nom et date d'embauche du manager.

7. Lister les numéros des employés n'ayant pas de subordonné.

8. Afficher les noms et dates d'embauche des employés embauchés avant BLAKE.

9. Afficher les employés embauchés le même jour que FORD.

10. Lister les employés ayant le même manager que CLARK.

11. Lister les employés ayant même job et même manager que TURNER.

12. Lister les employés du département RESEARCH embauchés le même jour que quelqu'un du département SALES.

13. Lister le nom des employés et également le nom du jour de la semaine correspondant à leur date d'embauche.
14. Donner, pour chaque employé, le nombre de mois qui s'est écoulé entre leur date d'embauche et la date actuelle.
15. Afficher la liste des employés ayant un M et un A dans leur nom.
16. Afficher la liste des employés ayant deux 'A' dans leur nom.
17. Afficher les employés embauchés avant tous les employés du département 10.
18. Sélectionner le métier où le salaire moyen est le plus faible.
19. Sélectionner le département ayant le plus d'employés.
20. Donner la répartition en pourcentage du nombre d'employés par département selon le modèle ci-dessous

Département	Répartition en %
-----	-----
10	21.43
20	35.71
30	42.86

## Quelques Fonctions SQL Server

CONVERT: Effectue des conversions de types de données. Permet notamment le formatage de dates

SUBSTRING: Extrait une partie d'une chaîne de caractères

DATENAME, DATEDIFF ...: Permet la manipulation de date