

# Langage SQL 3

## Travaux pratiques

<http://www.centraliup.fr.st>

1. Donner le total des salaires et des commissions des vendeurs.

```
SELECT Sum(sal) ,Sum(comm) FROM scott.emp
WHERE job='SALESMAN';
```

```
SUM(SAL)  SUM(COMM)
-----  -
5600      2200
```

2. Donner l'écart maximal de salaires dans l'entreprise.

```
SELECT Max(sal) -Min(sal) FROM scott.emp;
```

```
MAX(SAL) -MIN(SAL)
-----
4200
```

3. Donner le nombre d'employés susceptibles de recevoir une commission.

```
SELECT Count(ename) FROM scott.emp
WHERE comm>=0;
```

```
COUNT(ENAME)
-----
4
```

4. Donner la liste des employés avec leur salaire qui gagnent le plus d'argent.

```
SELECT ename, sal FROM scott.emp
WHERE sal>=(SELECT Max(sal) FROM scott.emp);
```

```
ENAME          SAL
-----
KING            5000
```

5. Donner les salaires moyens par départements.

```
SELECT deptno, AVG(sal) FROM scott.emp
GROUP BY deptno;
```

```
DEPTNO  AVG(SAL)
-----
10 2916.66667
20 2175
30 1566.66667
```

- 6.** Donner le nombre d'employés ainsi que leur salaire moyen annuel, pour chaque job de chaque département.

```
SELECT deptno, job, 12*AVG(sal) FROM scott.emp
GROUP BY deptno, job;
```

| DEPTNO | JOB       | 12*AVG(SAL) |
|--------|-----------|-------------|
| 10     | CLERK     | 15600       |
| 10     | MANAGER   | 29400       |
| 10     | PRESIDENT | 60000       |
| 20     | ANALYST   | 36000       |
| 20     | CLERK     | 11400       |
| 20     | MANAGER   | 35700       |
| 30     | CLERK     | 11400       |
| 30     | MANAGER   | 34200       |
| 30     | SALESMAN  | 16800       |

9 rows selected.

- 7.** Donner la liste des départements ayant au moins deux clercs.

```
SELECT deptno, count(job) FROM scott.emp
WHERE job='CLERK'
GROUP BY deptno
HAVING count(job) >=2;
```

| DEPTNO | COUNT(JOB) |
|--------|------------|
| 20     | 2          |

- 8.** Donner la liste des jobs ayant un salaire moyen supérieur à celui des managers.

```
SELECT job, AVG(sal) FROM scott.emp
GROUP BY job
HAVING AVG(sal) > (SELECT AVG(sal) FROM scott.emp WHERE job='MANAGER');
```

| JOB       | AVG(SAL) |
|-----------|----------|
| ANALYST   | 3000     |
| PRESIDENT | 5000     |

- 9.** Donner la liste des salaires annuels des employés du département 30 ; le titre de la colonne doit être « Salaire annuel ».

```
SELECT 12*sal "Salaire annuel" FROM scott.emp
WHERE deptno=30
ORDER BY sal DESC;
```

| Salaire annuel |
|----------------|
| 34200          |
| 19200          |
| 18000          |
| 15000          |
| 15000          |
| 11400          |

6 rows selected.

**10.** Donner l'ensemble des employés directement ou indirectement sous les ordres de Jones.

```
SELECT RPAD(LPAD('   ',2*(LEVEL-1)) || ename,20) "Hierarchie" FROM scott.emp
START WITH ename='JONES'
CONNECT BY PRIOR empno=mgr;
```

```
Hierarchie
-----
JONES
  SCOTT
    ADAMS
  FORD
    SMITH
```

**11.** Donner tous les employés au-dessus de Smith dans l'organigramme de la société.

```
SELECT RPAD(LPAD('   ',2*(4-LEVEL)) || ename,20) "Hierarchie" FROM scott.emp
START WITH ename='SMITH'
CONNECT BY PRIOR mgr=empno;
```

```
Hierarchie
-----
      SMITH
     FORD
    JONES
   KING
```

**12.** Donner l'ensemble des employés travaillant pour Jones, sauf Scott et les employés travaillant pour lui.

```
SELECT RPAD(LPAD('   ',2*(LEVEL-1)) || ename,20) "Hierarchie" FROM scott.emp
WHERE ename!=ALL(SELECT ename FROM scott.emp
                 START WITH ename='SCOTT'
                 CONNECT BY PRIOR empno=mgr)
START WITH ename='JONES'
CONNECT BY PRIOR empno=mgr;
```

```
Hierarchie
-----
JONES
  FORD
    SMITH
```

**13.** Donner l'ensemble des employés travaillant pour Jones, sauf Scott (et c'est tout).

```
SELECT RPAD(LPAD('   ',2*(LEVEL-1)) || ename,20) "Hierarchie" FROM scott.emp
WHERE ename!='SCOTT'
START WITH ename='JONES'
CONNECT BY PRIOR empno=mgr;
```

```
Hierarchie
-----
JONES
  ADAMS
  FORD
    SMITH
```

## 14. Quel est le grade où il y a le plus de personnes du même département ?

```
SELECT grade FROM scott.emp, scott.salgrade
WHERE scott.emp.sal BETWEEN scott.salgrade.losal AND scott.salgrade.hisal
GROUP BY grade, deptno
HAVING count(ename) >= (SELECT max(count(ename)) FROM scott.emp, scott.salgrade
WHERE scott.emp.sal BETWEEN scott.salgrade.losal AND scott.salgrade.hisal
GROUP BY grade, deptno);
```

```
-----
GRADE
-----
4
```

## 15. Créer les tables suivantes : Avion, Pilote et Vol.

```
CREATE TABLE Avion
(
    numav NUMBER (4,0) NOT NULL,
    nomav VARCHAR2(10) NOT NULL,
    capav NUMBER(4,0),
    locav VARCHAR2(20) NOT NULL,
    CONSTRAINT pkavion PRIMARY KEY(numav),
    CONSTRAINT nombrecapav CHECK (capav BETWEEN 2 AND 500)
);
```

Table created.

```
CREATE TABLE Pilote
(
    numpil NUMBER (4,0) NOT NULL,
    nompil VARCHAR2(10) NOT NULL,
    adrpil VARCHAR2(20) NOT NULL,
    CONSTRAINT pkpilote PRIMARY KEY(numpil),
    CONSTRAINT nompiluniq UNIQUE(NOMPIL)
);
```

Table created.

```
CREATE TABLE Vol
(
    numvol NUMBER (4,0) NOT NULL,
    numpil NUMBER (4,0),
    numav NUMBER (4,0),
    vd VARCHAR2(20) NOT NULL,
    va VARCHAR2(20) NOT NULL,
    hd NUMBER(4,2),
    ha NUMBER(4,2),
    CONSTRAINT pkvol PRIMARY KEY(numvol),
    CONSTRAINT dom_heur_dep CHECK (hd BETWEEN 00.00 AND 23.59),
    CONSTRAINT dom_heur_arr CHECK (ha BETWEEN 00.00 AND 23.59)
);
```

Table created.