## **COMPTE RENDU**

BASES DE DONNÉES



**Encadrer par:** 

M.Moutachaouik M.Hamime



Réalisée par

El Ouardi Mohamed

#### 1. Afficher la date courante

SELECT SYSDATE AS "DATE COURANTE" FROM DUAL;

```
SQL> SELECT SYSDATE AS "Date Courante" FROM DUAL;

Date Cour
-----
16-NOV-24

SQL> |
```

2. Afficher la structure de la table DEPT. Sélectionner toutes les données de la table DEPT.

SELECT \* FROM DEPT;

```
SQL> SELECT * FROM DEPT;
    DEPTNO DNAME
                          LOC
        10 ACCOUNTING
                          NEW YORK
                          DALLAS
        20 RESEARCH
        30 SALES
                          CHICAGO
        40 OPERATIONS
                          BOSTON
        50 Finance
                          Casa
        60 RH
        70 Info
7 rows selected.
```

3. Afficher la structure de la table EMP. Créer une requête pour afficher le matricule, le nom, le poste et la date d'embauche de chaque employé, avec le matricule en premier.

SELECT EMPNO AS "MATRICULE", ENAME AS "NOM", JOB AS "POSTE", HIREDATE AS

"DATE D'EMBAUCHE" FROM EMP;

```
SQL> SELECT EMPNO AS "Matricule", ENAME AS "Nom", JOB AS "Poste", HIR
EDATE AS "Date d'Embauche'
 2 FROM EMP;
 Matricule Nom
                       Poste
                                  Date d'Em
      7369 SMITH
      7499 ALLEN
                        SALESMAN 20-FEB-11
      7521 WARD
                        SALESMAN
                                  22-FEB-11
      7566 JONES
7654 MARTIN
                        MANAGER
                                  02-APR-07
                        SALESMAN
                                  28-SEP-10
      7698 BLAKE
                        MANAGER
                                  01-MAY-10
      7782 CLARK
                        MANAGER
                                  09-JUN-09
      7788 SCOTT
                        ANALYST
                                  09-DEC-12
                       PRESIDENT 17-NOV-07
SALESMAN 11-SEP-12
      7839 KING
      7844 TURNER
                                   12-JAN-13
      7876 ADAMS
                        CLERK
```

4. Afficher le nom de chaque employé concaténé avec le salaire, en nommant la colonne "Employe avec son salaire" :

SELECT ENAME || ' A UN SALAIRE DE ' || SAL AS "EMPLOYE AVEC SON SALAIRE" FROM EMP;

```
SQL> SELECT ENAME || ' a un salaire de ' || SAL AS "Employe avec son salaire" FROM EMP;

Employe avec son salaire

SMITH a un salaire de 800

ALLEN a un salaire de 1600

WARD a un salaire de 1250

JONES a un salaire de 2975

MARTIN a un salaire de 1250

BLAKE a un salaire de 2850

CLARK a un salaire de 2450

SCOTT a un salaire de 3000

KING a un salaire de 5000

TURNER a un salaire de 1500

ADAMS a un salaire de 1100
```

5. Afficher toutes les données de la table DEPT dans une seule colonne, nommée "THE\_OUTPUT" :

```
SELECT DEPTNO || ', ' || DNAME || ', ' || LOC AS "THE_OUTPUT" FROM DEPT;
```

```
SQL> select deptno ||','||dname||','||loc as THE_OUTPUT
2 FROM dept
3;

THE_OUTPUT

10,ACCOUNTING,NEW YORK
20,RESEARCH,DALLAS
30,SALES,CHICAGO
40,OPERATIONS,BOSTON
```



## SÉLECTION ET TRI DES LIGNES RETOURNÉES



1. Afficher le nom, le poste et la date d'entrée des employés embauchés entre le 20 février 2010 et le 1 mai 2012, triés par date d'embauche :

select ename, job, hiredate from emp

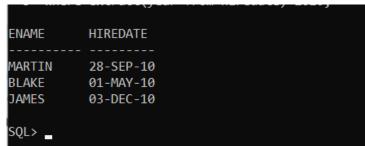
where hiredate between to\_date('20-feb-10', 'dd-mon-yy') and to\_date('01-may-12', 'dd-mon-yy')

order by hiredate;

ENAME	JOB	HIREDATE		
BLAKE	MANAGER	01-MAY-10		
MARTIN	SALESMAN	28-SEP-10		
JAMES	CLERK	03-DEC-10		
ALLEN	SALESMAN	20-FEB-11		
VARD	SALESMAN	22-FEB-11		
ORD	ANALYST	03-DEC-11		
MILLER	CLERK	23-JAN-12		
rows se	lected.			

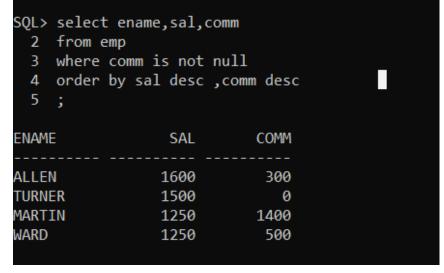
2. Afficher le nom et la date d'embauche de chaque employé entré en 2010 :

select ename, hiredate from emp where extract(year from hiredate) = 2010;



3. Afficher le nom, le salaire et la commission des employés qui perçoivent des commissions, triés par salaire et commission décroissants :

select ename, sal, comm from emp where comm is not null order by sal desc, comm desc;



4. Afficher le nom des employés dont le nom contient deux "t" et travaillant dans le département 20 ou avec le manager 7566 :

select ename

from emp

where (ename like '%t%t%' and deptno = 20) or mgr = 7566;

```
SQL> select ename from emp where (ename like '%t%t%' and deptno = 20) or mgr = 7566;

ENAME
------SCOTT
FORD
```

5. Nom des vendeurs (salesman) triés par ordre décroissant du ratio commission/salaire :

```
select ename, comm/sal as ratio
from emp
where job = 'salesman'
order by ratio desc;
```

```
SQL> select ename, comm/sal as ratio
2 from emp
3 where job = 'salesman'
4 order by ratio desc;
no rows selected
```

6. Nom des vendeurs dont la commission est inférieure à 25% de leur salaire :

select ename

from emp

where job = 'salesman' and comm < 0.25 \* sal;

```
SQL> select ename
2 from emp
3 where comm <= 0.25*sal
4 ;

ENAME
------
ALLEN
TURNER
```

7. Liste unique de tous les postes du département 30 :

```
SQL> select distinct job from emp where deptno = 30;

select distinct job

from emp

SALESMAN

CLERK

MANAGER

SQL>
```

8. Afficher le nom, le poste et le salaire des employés clerk ou account avec un salaire différent de 1000, 3000 ou 5000 :

```
select ename, job, sal from emp where job in ('clerk', 'account') and sal not in (1000, 3000, 5000);
```

```
SQL> select ename , job ,s<mark>a</mark>l
 2 from emp
 3 where job in ('CLERK', 'ACCOUNT') and sal not in (1000,3000,5000)
ENAME
           JOB
                              SAL
           CLERK
SMITH
                              800
ADAMS
           CLERK
                             1100
JAMES
           CLERK
                              950
MILLER
           CLERK
                             1300
```

9. Afficher le nom, le salaire et la commission des employés dont la commission est supérieure de plus de 10% au salaire :

```
select ename, sal, comm
from emp
where comm > 1.1 * sal;
```

**FONCTIONS MONO-LIGNE** 



1. Afficher le matricule, le nom, le salaire et le salaire augmenté de 15%, nommé "new salary" :

select empno, ename, sal, round(sal \* 1.15) as new\_salary

from emp;

SQL>	select	empno, ename	, sal, ro	ound(sal *	1.15) as	new_salary	from emp;
	EMPNO	ENAME	SAL	NEW_SALAR	1		
	7369	SMITH	800	920	9		
ıt	7499	ALLEN	1600	1846	)		
	7521	WARD	1250	1438	3		
	7566	JONES	2975	3421	Ĺ		
	7654	MARTIN	1250	1438	3		
	7698	BLAKE	2850	3278	3		
	7782	CLARK	2450	2818	3		
1	7788	SCOTT	3000	3450	)		
	7839	KING	5000	5750	)		

2. Ajouter une colonne indiquant l'augmentation, nommée "increase" :

select empno, ename, sal, round(sal \* 1.15) as new\_salary, round(sal \* 1.15 - sal) as increase

from emp;

```
select empno, ename, sal, round(sal * 1.15) as new_salary ,round(sal*1.15 - sal) as increase
from emp;
EMPNO ENAME
                         SAL NEW SALARY
                                           INCREASE
 7369 SMITH
                                     920
                                                 120
                         800
 7499 ALLEN
                         1600
                                    1840
                                                 240
 7521 WARD
                         1250
                                    1438
                                                 188
  7566 JONES
                                    3421
                                                 446
  7654 MARTIN
                         1250
                                    1438
                                                 188
 7698 BLAKE
                         2850
                                    3278
                                                 428
  7782 CLARK
                         2450
                                    2818
                                                 368
  7788 SCOTT
                         3000
                                    3450
                                                 450
                                    5750
  7839 KING
                         5000
                                                 750
  7844 TURNER
                         1500
                                                 225
  7876 ADAMS
                         1100
                                    1265
                                                 165
```

3. Afficher le nom de chaque employé et le nombre de mois travaillés depuis l'embauche, nommé "months\_worked", trié par ancienneté :

select ename, round(months between(sysdate, hiredate)) as months worked

from emp

order by months\_worked desc;

```
SQL> select ename , round(months_between(sysdate,hiredate)) as MONTHS_WORKED
    from emp
    order by MONTHS_WORKED DESC;
           MONTHS_WORKED
ENAME
JONES
KING
                      204
SMITH
CLARK
BLAKE
MARTIN
                      170
JAMES
                      168
ALLEN
                      165
WARD
                      165
FORD
                      156
MILLER
                      154
```

4. Afficher "<nom de l'employé> gagne <salaire> par mois, mais veut <3 fois son salaire>", nommé "salaires de rêve" :

select ename || ' gagne ' || sal || ' par mois, mais veut ' || (sal \* 3) as salaires\_de\_rêve

from emp;

```
SALAIRES_DE_REVE

SMITH gagne800 par mois , mais veut 2400
ALLEN gagne1600 par mois , mais veut 4800
MARD gagne1250 par mois , mais veut 8925
MARTIN gagne1250 par mois , mais veut 3750
BLAKE gagne2850 par mois , mais veut 8550
LARK gagne2450 par mois , mais veut 7350
SCOTT gagne3000 par mois , mais veut 9000
CING gagne5000 par mois , mais veut 15000
TURNER gagne1100 par mois , mais veut 4500
ADAMS gagne1100 par mois , mais veut 3300
```

5. Afficher le nom et le salaire avec 15 caractères, complété par

des \$, nommé "salary" :

select ename, lpad(sal, 15, '\$') as salary from emp;

```
SQL> select ename, lpad(sal,15,'$')as salary
 2 from emp;
ENAME
           SALARY
SMITH
           $$$$$$$$$$$$800
ALLEN
           $$$$$$$$$$$1600
WARD
           $$$$$$$$$$$1250
JONES
           $$$$$$$$$$$2975
MARTIN
           $$$$$$$$$$$1250
BLAKE
           $$$$$$$$$$$$2850
CLARK
           $$$$$$$$$$$2450
SCOTT
           $$$$$$$$$$$3000
KING
           $$$$$$$$$$$5000
TURNER
           $$$$$$$$$$$1500
ADAMS
           $$$$$$$$$$$1100
```

6. Afficher le nom et la commission de chaque employé, remplacer les null par "no commission" :

select ename, nvl(to\_char(comm), 'no commission') as comm

from emp;

```
SQL> select ename , nvl(to_char(comm),'no commission') as comm
  2 from emp;
ENAME
           COMM
SMITH
           no commission
ALLEN
          300
WARD
          500
JONES
          no commission
MARTIN
          1400
BLAKE
          no commission
CLARK
          no commission
          no commission
SCOTT
KING
          no commission
TURNER
          0
ADAMS
          no commission
```

LES JOINTURES



## 1. Afficher le nom, le numéro de département et le nom de département de tous les employés :

select emp.ename, emp.deptno, dept.dname from emp, dept where emp.deptno = dept.deptno;

ENAME	DEPTNO	DNAME
SMITH	20	RESEARCH
ALLEN	30	SALES
WARD	30	SALES
JONES	20	RESEARCH
MARTIN	30	SALES
BLAKE	30	SALES
CLARK	10	ACCOUNTING
SCOTT	20	RESEARCH
KING	10	ACCOUNTING
TURNER	30	SALES
adams	20	RESEARCH

2. Créer une liste unique de tous les postes du département 30 :

select distinct job from emp where deptno = 30;



3. Afficher le nom, le nom du département et la localisation de tous les employés qui touchent une commission :

select emp.ename, dept.dname, dept.loc from emp, dept where emp.deptno = dept.deptno and emp.comm is not null;

2 from 3 wher	emp , dep	no=dept.deptno
ENAME	DNAME	LOC
TURNER	SALES	CHICAGO
MARTIN	SALES	CHICAGO
WARD	SALES	CHICAGO
ALLEN	SALES	CHICAGO

## 4. Afficher le nom et le nom du département pour tous les employés dont le nom contient la lettre "a" :

select emp.ename, dept.dname from emp, dept where emp.deptno = dept.deptno and emp.ename like '%a%';

SALES SALES SALES						
SALES						
SALES						
O'HEED						
SALES						
ACCOUNTING						
RESEARCH						
SALES						
SALES						
ected.						
	ACCOUNTING RESEARCH SALES	ACCOUNTING RESEARCH SALES	ACCOUNTING RESEARCH SALES	ACCOUNTING RESEARCH SALES	ACCOUNTING RESEARCH SALES	ACCOUNTING RESEARCH SALES

5. Afficher le nom, le poste, le numéro de département et le nom du département de tous les employés basés à dallas :

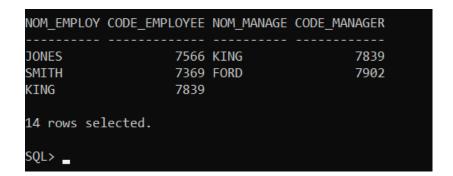
select emp.ename, emp.job, emp.deptno, dept.dname from emp, dept where emp.deptno = dept.deptno and dept.loc = 'dallas';

ENAME	JOB	DEPTNO	DNAME
SMITH	CLERK	20	RESEARCH
JONES	MANAGER		RESEARCH
SCOTT	ANALYST	20	RESEARCH
ADAMS	CLERK	20	RESEARCH
FORD	ANALYST	20	RESEARCH
SOL >			

6. Afficher le nom et le matricule des employés et de leur manager, avec des colonnes nommées "nom\_employee", "code\_employee", "nom\_manager" et "code\_manager" :

select e.ename as nom\_employee, e.empno as code\_employee, m.ename as nom\_manager, m.empno as code\_manager

from emp e, emp m
where e.mgr = m.empno(+);



7. Afficher les noms et dates d'embauche des employés et de leur manager, pour tous les employés ayant été embauchés avant leur manager, avec des colonnes nommées "employee", "emp\_hiredate", "manager" et "mgr\_hiredate":

select e.ename as employee, e.hiredate as emp\_hiredate,
 m.ename as manager, m.hiredate as mgr\_hiredate

from emp e, emp m

where e.mgr = m.empno

and e.hiredate < m.hiredate;

```
EMPLOYEE EMP_HIRED MANAGER MGR_HIRED

JONES 02-APR-07 KING 17-NOV-07
SMITH 17-DEC-08 FORD 03-DEC-11

SQL> _
```

LES FONCTIONS DE GROUPE



#### 1. Déterminez si les affirmations suivantes sont vraies ou fausses :

- 1. Déterminez si les affirmations suivantes sont vraies ou fausses :
- a. Les fonctions de groupe agissent sur plusieurs lignes pour produire un seul résultat.
- Les fonctions de groupe comme sum, avg, min, max calculent une valeur agrégée sur plusieurs lignes.
- b. Les fonctions de groupe intègrent les valeurs null dans leurs calculs.
- Les valeurs null sont ignorées par les fonctions de groupe (sauf count(\*) qui compte toutes les lignes, y compris celles contenant des null).
- c. La clause where restreint les lignes avant qu'elles soient incluses dans un calcul de groupe. vrai
- La clause where filtre les lignes avant qu'un regroupement soit effectué.

## 2. Affichez le salaire maximum, minimum, la somme des salaires et le salaire moyen de tous les employés.

round(max(sal), 0) as maximum, round(min(sal), 0) as minimum, round(sum(sal), 0) as sum,

round(avg(sal), 0) as average from emp;

MAXIMUM	MINIMUM	SUM	AVERAGE	
5000	800	29025	2073	
SQL>				

## 3. Afficher le salaire maximum, minimum, la somme des salaires et le salaire moyen pour chaque type de poste :

select

select

job,

round(max(sal), 0) as maximum, round(min(sal), 0) as minimum, round(sum(sal), 0) as sum, round(avg(sal), 0) as average

from emp

group by job;

ЈОВ	MAXIMUM	MINIMUM	SUM	AVERAGE
CLERK	1300	800	4150	1038
SALESMAN	1600	1250	5600	1400
PRESIDENT	5000	5000	5000	5000
Manager	2975	2450	8275	2758
ANALYST	3000	3000	6000	3000
SQL> S_	3000	3000	0000	3000

### 4. Afficher la différence entre le salaire maximum et le salaire minimum.

```
select
  max(sal) - min(sal) as difference
from emp;
```

### 5. Affichez le matricule des différents managers et le salaire minimum le plus bas de leurs employés.

```
select

mgr as manager_id,

min(sal) as minimum_salary

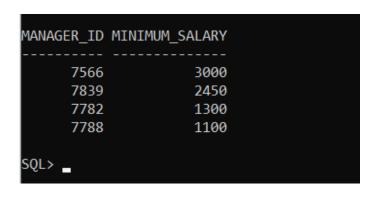
from emp

where mgr is not null

group by mgr

having min(sal) >= 1000

order by minimum_salary desc;
```



## 6. Affichez le département, le nombre d'employés et le salaire moyen pour chaque département.

```
select

deptno as code_departement,

count(*) as nombre_d_employes,

round(avg(sal), 0) as salaire_moyenne

from emp

group by deptno;
```

ODE_DEPARTEMENT	NOMBRE_D_EMPLOYES	SALAIRE_MOYENNE
30	6	1567
20	5	2175
10	3	2917

LES SOUS-REQUÊTES



#### 1. Afficher le département qui ne comprend aucun employé :

select deptno, dname from dept where deptno not in (select deptno from emp);

```
SQL> select deptno, dname
2 from dept
3 where deptno not in (select deptno from emp);

DEPTNO DNAME

40 OPERATIONS

SQL>
```

2. Afficher le nom et la date d'embauche de tous les employés travaillant dans le même département que Blake, à l'exclusion de Blake :

select ename, hiredate
from emp
where deptno = (select deptno from emp where ename = 'BLAKE')
and ename != 'BLAKE';

ENAME	HIREDATE
ALLEN	20-FEB-11
WARD	22-FEB-11
MARTIN	28-SEP-10
TURNER	11-SEP-12
JAMES	03-DEC-10
SQL> _	

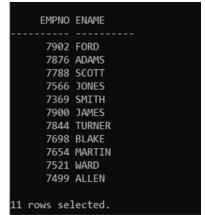
3. Afficher le matricule et le nom de tous les employés qui gagnent plus que le salaire moyen. Triez les résultats par ordre décroissant des salaires :

select empno, ename, sal
from emp
where sal > (select avg(sal) from emp)
order by sal desc;

	•	•	
EMPNO	ENAME	SAL	
7839	KING	5000	
7902	FORD	3000	
7788	SCOTT	3000	
7566	JONES	2975	
7698	BLAKE	2850	
7782	CLARK	2450	
6 rows sele	ected.		

4. Afficher le matricule et le nom de tous les employés qui travaillent dans le même département que tout employé dont le nom contient un T :

select empno, ename
from emp
where deptno in (select deptno
from emp
where ename like '%T%');



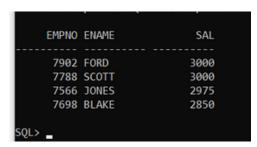
5. Afficher le matricule, le nom et le salaire de tous les employés qui gagnent plus que le salaire moyen et qui travaillent dans un département avec tout employé dont le nom contient un T :

```
select empno, ename, sal

from emp

where sal > (select avg(sal) from emp)

and deptno in (select deptno from emp where ename like '%T%');
```

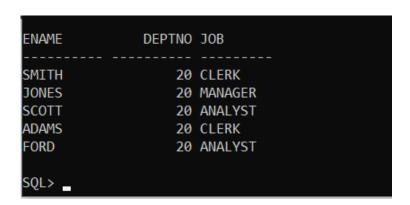


6. Afficher le nom, le numéro de département et le poste de tous les employés dont le département est situé à DALLAS :

```
select ename, deptno, job

from emp

where deptno = (select deptno from dept where loc = 'DALLAS');
```



## 7. Afficher le nom et le salaire de tous les employés dont le manager est KING :

select ename, sal from emp where mgr = (select empno from emp where ename = 'KING');

ENAME	SAL
JONES	2975
BLAKE	2850
CLARK	2450

8. Afficher le numéro de département, le nom et le poste de tous les employés travaillant dans le département des ventes (SALES) :

select deptno, ename, job
from emp
where deptno = (select deptno from dept where dname = 'SALES');

	DEPTNO	ENAME	ЗОВ
	30	ALLEN	SALESMAN
	30	WARD	SALESMAN
	30	MARTIN	SALESMAN
	30	BLAKE	MANAGER
);	30	TURNER	SALESMAN
<i>)</i> ,	30	JAMES	CLERK
	6 rows sele	ected.	
	601		

9. Afficher les employés qui perçoivent un salaire supérieur à tout employé dont le poste est CLERK. Triez le résultat par ordre décroissant des salaires :

select empno, ename, sal from emp where sal > all (select sal from emp where job = 'CLERK') order by sal desc;

EMPNO	ENAME	SAL
7839	KING	5000
7902	FORD	3000
7788	SCOTT	3000
7566	JONES	2975
7698	BLAKE	2850
7782	CLARK	2450
7499	ALLEN	1600
7844	TURNER	1500
8 rows sele	ected.	

MANIPULATION DES DONNÉES



#### 1. Insérer des données dans les tables DEPT et EMP :

a. Ajouter les lignes dans la table DEPT:

```
insert into dept values (50, 'Finance', 'Casa');
insert into dept values (60, 'Personnel', null);
insert into dept values (70, 'Info', null);
```

```
SQL> insert into dept values (60, 'Personnel', null);

1 row created.

SQL> insert into dept values (70, 'Info', null);

1 row created.

SQL> select * from dept

2 ;
```

#### <u>b. Ajouter les lignes dans la table EMP :</u>

insert into emp values (8100, 'Akram', 'Manager', 7839, to\_date('21/04/15', 'dd/mm/yy'), 2700, 100, 50); insert into emp values (8200, 'Laila', 'Clarck', 7566, to\_date('12/03/15', 'dd/mm/yy'), 1950, null, 60); insert into emp values (8300, 'Sara', 'Salesman', 7698, to\_date('18/03/15', 'dd/mm/yy'), 1800, null, 30);

```
SQL> insert into emp values (8100, 'Akram', 'Manager', 7839, to_date('21/04/15', 'dd/mm/yy'), 2700, 100, 50);

1 row created.

SQL> insert into emp values (8200, 'Laila', 'Clarck', 7566, to_date('12/03/15', 'dd/mm/yy'), 1950, null, 60);

1 row created.

SQL> insert into emp values (8300, 'Sara', 'Salesman', 7698, to_date('18/03/15', 'dd/mm/yy'), 1800, null, 30);

1 row created.

SQL>
```

#### 2. Vérifiez le contenu de chaque table :

```
select * from dept;
select * from emp;
```



#### 3. Changez le nom du département Personnel en RH:

```
update dept
set dname = 'RH'
where dname = 'Personnel';
```

```
SQL> update dept
2 set dname='RH'
3 where dname='Personnel'
4 ;
1 row updated.
```

#### 4. Remplacez le nom de l'employé 8100 par Karim :

```
update emp

set ename = 'Karim'

where empno = 8100;

SQL> update emp
2 set ename='KARIM'
3 where empno=8100;

1 row updated.
```

5. Saisissez un salaire de 1400 pour tous les employés ayant un salaire inférieur à 1200 :

```
update emp

SQL> update emp

2 set sal = 1400

3 where sal<=1200;

where sal < 1200;

3 rows updated.
```

#### 6. Vérifiez vos modifications:

select \* from emp;

o rows upuc	iccu.							
QL> select * from emp;								
EMPNO	ENAME	J0B	MGR	HIREDATE	SAL	COMM		
DEPTNO								
7369 20	SMITH	CLERK	7902	17-DEC-08	1400			
7499 30	ALLEN	SALESMAN	7698	20-FEB-11	1600	300		
7521 30	WARD	SALESMAN	7698	22-FEB-11	1250	500		
EMPNO	ENAME	J0B	MGR	HIREDATE	SAL	COMM		
DEPTNO								
7566 20	JONES	MANAGER	7839	02-APR-07	2975			
7654 30	MARTIN	SALESMAN	7698	28-SEP-10	1250	1400		
7698 30	BLAKE	Manager	7839	01-MAY-10	2850			
EMPNO DEPTNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM		

#### 7. Supprimer l'employé 8300 :

```
delete from emp
where empno = 8300;
```

```
SQL> delete from emp where empno = 8300;
1 row deleted.
SQL>
```

#### 8. Vérifiez le contenu de la table :

select \* from emp;



#### 9. Maintenant, videz entièrement la table EMP :

truncate table emp;



#### 10. Vérifiez que la table est vide :

```
select * from emp;
```

```
SQL> select * from emp;
no rows selected
SQL> _
```

#### 11. Annuler toutes les opérations :

rollback;

```
SQL> rollback
2 ;
Rollback complete.
```

CRÉATION DES TABLES / LES CONTRAINTES,



#### 1. Initialisez une session SQLPlus avec vos identifiants:

connect u2/u2

```
SQL> conn u2/u2
Connected.
SQL>
```

#### 2. Afficher le nom de l'utilisateur connecté :

show user;

);

```
SQL> conn u2/u2
Connected.
SQL> show user
USER is "U2"
SQL> _
```

#### 3. Afficher la date courante :

select sysdate from dual;

```
SQL> select sysdate from dual;

SYSDATE
------
15-DEC-24
```

#### 4. Afficher la liste de toutes les tables de votre base de données :

select table\_name from user\_tables;

```
SQL> select table_name from user_tables;

TABLE_NAME

DEPT
BONUS
SALGRADE
DUMMY
EMP
```

#### 5. Créer la table Service et vérifier sa création :

```
create table service (

id number(2),

nom varchar(25));

Table created.

SQL> create table service(id number(2),nom varchar(25));

Table created.

SQL> _
```

6. Créer la table Employe et vérifier sa création :

7. Modifier la table Employe pour allonger la taille des noms des employés et vérifier :

alter table employe modify nom varchar2(30);

8. Vérifiez que les tables SERVICE et EMPLOYE sont bien enregistrées :

select table\_name from user\_tables where table\_name in ('SERVICE',

'EMPLOYE');

```
SQL> select table_name from user_tables where table_name in ('SERVICE', 'EMPLOYE');

TABLE_NAME
------
EMPLOYE
SERVICE
SOL>
```

9. Ajouter une colonne Fonction dans la table Employe :

alter table employe add fonction varchar2(25);

```
SQL> alter table employe add fonction varchar2(25);
Table altered.
SOL>
```

10. Supprimer la colonne salaire de la table Employe et vérifier :

alter table employe drop column salaire;

```
SQL> alter table employe drop column salaire;
Table altered.
SQL>
```

#### 11. Créer la table Employe2 avec une structure basée sur la table Emp :

create table employe2 as select empno as id\_emp, ename as nom, deptno as id\_service from emp;

```
Table aftered.

SQL> create table employe2 as select empno as id_emp, ename as nom, deptno as id_service from emp;

Table created.
```

#### 12. Supprimer la table Employe:

drop table employe;

```
SQL> drop table employe;
Table dropped.
SQL> _
```

#### 13. Renommer la table Employe2 en Employe:

rename employe2 to employe;

```
SQL> rename employe2 to employe;
Table renamed.
```

## 14. Ajouter une contrainte PRIMARY KEY sur la colonne id\_emp dans la table Employe :

alter table employe add constraint pk\_employe primary key (id\_emp);

```
SQL> alter table employe add constraint pk_employe primary key (id_emp);
Table altered.
SOL> _
```

### 15. Ajouter une contrainte PRIMARY KEY sur la colonne id dans la table Service :

alter table service add constraint pk\_service primary key (id);

```
SQL> alter table service add constraint pk_service primary key (id);
Table altered.
SQL>
```

### 16. Ajouter une clé étrangère dans la table Employe pour contrôler les services :

alter table employe add constraint fk\_employe\_service foreign key (id\_service) references service(id);

```
SQL> alter table employe add constraint fk_employe_service foreign key (id_service) references service(id);
Table altered.

SQL> _
```

## 17. Ajouter une contrainte NOT NULL à la colonne nom dans la table Employe :

alter table employe modify nom varchar2(30) not null;

```
SQL> alter table employe modify nom varchar2(30) not null;
Table altered.
```

## **COMPTE RENDU**

BASES DE DONNÉES

**Encadrer par:** 

M.Moutachaouik M.Hamime



Réalisée par

El Ouardi Mohamed