



جامعة سيدي محمد بن عبد الله
كلية العلوم دهر المهرار

جامعة سيدي محمد بن عبد الله
كلية العلوم دهر المهرار

Université Sidi Mohamed Ben Abdellah
Faculté des Sciences Dhar Mahraz



Suite Rapport TP2

Base de données

Réalisé par :

- Adil EL Alami
- Ayoub Elbouri

Fonctions de groupe

Exercices :

1. Afficher en arrondissant:

- le plus grand salaire
- le plus petit salaire
- la somme de tous les salaires
- la moyenne des salaires

```
SQL> SELECT MAX(salary),MIN(salary),SUM(salary),AUG(salary)
2 FROM employees;

MAX(SALARY) MIN(SALARY) SUM(SALARY) AUG(SALARY)
-----
24000      2100      691416  6461,83178
```

2. Afficher le minimum, le maximum, la somme et la moyenne des salaires pour chaque job

```

SQL> column JOB_TITLE format a20;
SQL> SELECT j.job_title,MIN(e.salary),MAX(e.salary),SUM(e.salary),AVG(e.salary)
  2 FROM employees e,jobs j
  3 WHERE e.job_id=j.job_id
  4 GROUP BY j.job_title;

```

JOB_TITLE	MIN(E.SALARY)	MAX(E.SALARY)	SUM(E.SALARY)	AVG(E.SALARY)
Accounting Manager	12008	12008	12008	12008
Programmer	4200	9000	28800	5760
Public Relations Rep representative	10000	10000	10000	10000
Purchasing Clerk	2500	3100	13900	2780
Sales Representative	6100	11500	250500	8350
Administration Vice President	17000	17000	34000	17000
Marketing Representa tive	6000	6000	6000	6000
Stock Manager	5800	8200	36400	7280
Administration Assis tant	4400	4400	4400	4400
President	24000	24000	24000	24000
Finance Manager	12008	12008	12008	12008
Purchasing Manager	11000	11000	11000	11000
Human Resources Repr esentative	6500	6500	6500	6500
Accountant	6900	9000	39600	7920
Shipping Clerk	2500	4200	64300	3215
Stock Clerk	2100	3600	55700	2785
Public Accountant	8300	8300	8300	8300
Marketing Manager	13000	13000	13000	13000
Sales Manager	10500	14000	61000	12200

19 rows selected.

3. Afficher le nombre de personnes ayant le même travail (et ce travail)

```
SQL> SELECT j.job_title,COUNT(e.last_name)
2 FROM employees e,jobs j
3 WHERE e.job_id=j.job_id
4 GROUP BY j.job_title;
```

JOB_TITLE	COUNT(E.LAST_NAME)
-----------	--------------------

Accounting Manager	1
Sales Representative	30
Purchasing Clerk	5
Programmer	5
Public Relations Rep representative	1

Administration Vice President	2
----------------------------------	---

Stock Manager	5
---------------	---

JOB_TITLE	COUNT(E.LAST_NAME)
-----------	--------------------

Marketing Representa tive	1
------------------------------	---

President	1
Administration Assis tant	1

Finance Manager	1
Purchasing Manager	1
Human Resources Repr esentative	1

JOB_TITLE	COUNT(E.LAST_NAME)
-----------	--------------------

Accountant	5
Stock Clerk	20
Shipping Clerk	20
Public Accountant	1
Sales Manager	5
Marketing Manager	1

19 rows selected.

Exercices :

4. Afficher le nombre de personnes ayant un manager

```
SQL> SELECT COUNT(last_name)
2 FROM employees
3 WHERE manager_id IS NOT NULL;
```

COUNT(LAST_NAME)

106

5. Afficher la différence entre le plus grand et le plus petit salaire

```
SQL> SELECT MIN(salary),MAX(salary),MAX(salary)-MIN(salary) AS "Différence_Sal"
2 FROM employees;
```

MIN(SALARY)	MAX(SALARY)	Différence_Sal
2100	24000	21900

6. Afficher le numéro du manager et le plus petit salaire des employés sous sa direction.

- Exclure les employés sans manager.
- Exclure les employés ayant un salaire < 1000
- Trier par ordre décroissant de salaire

```
SQL> SELECT manager_id,MIN(salary)
2 FROM employees
3 WHERE manager_id IS NOT NULL AND salary > 1000
4 GROUP BY manager_id
5 ORDER BY MIN(salary);
```

MANAGER_ID	MIN(SALARY)
121	2100
122	2200
120	2200
123	2500
124	2500
114	2500
103	4200
101	4400
100	5800
201	6000
148	6100
149	6200
147	6200
108	6900
145	7000
146	7000
205	8300
102	9000

18 rows selected.

Exercices :

7. Créer une requête affichant:

- le nom du département
- sa localisation
- le nombre d'employés
- la moyenne salariale de tous les employés de chaque département

```

SQL> column CITY format a20;
SQL> column DEPARTMENT_NAME format a20;
SQL> SELECT d.department_name,l.city,COUNT(e.last_name),AUG(e.salary)
2 FROM employees e,departments d,locations l
3 WHERE e.department_id=d.department_id AND d.location_id=l.location_id
4 GROUP BY department_name,city;

```

DEPARTMENT_NAME	CITY	COUNT(E.LAST_NAME)	AUG(E.SALARY)
IT	Southlake	5	5760
Public Relations	Munich	1	10000
Sales	Oxford	34	8955,88235
Accounting	Seattle	2	10154
Shipping	South San Francisco	45	3475,55556
Administration	Seattle	1	4400
Purchasing	Seattle	6	4150
Executive	Seattle	3	19333,3333
Human Resources	London	1	6500
Finance	Seattle	6	8601,33333
Marketing	Toronto	2	9500

```

11 rows selected.

```

8. Créer une requête affichant:

- le nombre total d'employés
- le nombre d'employés engagés en 1981, 1982 et 1983

```

SQL> SELECT hire_date,COUNT (last_name) "Nb emp engagés",<SELECT COUNT(last_name
> FROM employees> "Total"
2 FROM employees
3 WHERE hire_date LIKE '%81' OR hire_date LIKE '%82' OR hire_date LIKE '%83'
4 GROUP BY hire_date;

```

no rows selected

En remplaçant les années 1981, 1982, 1983 par 2004, 2005, 2006 :

```
SQL> SELECT hire_date,COUNT (<last_name> "Nb emp engagés",<SELECT COUNT(<last_name
> FROM employees> "Total"
2 FROM employees
3 WHERE hire_date LIKE '%04' OR hire_date LIKE '%05' OR hire_date LIKE '%06'
4 GROUP BY hire_date;
```

HIRE_DAT	Nb emp engagés	Total
30/01/05	1	107
04/03/04	1	107
14/06/05	1	107
03/03/05	1	107
03/01/06	1	107
18/07/04	1	107
28/09/06	1	107
26/08/06	1	107
29/01/05	1	107
30/03/06	1	107
15/12/05	1	107

HIRE_DAT	Nb emp engagés	Total
27/01/04	1	107
04/02/04	1	107
23/05/06	1	107
07/03/06	1	107
10/10/05	1	107
06/04/06	1	107
09/07/06	1	107
01/10/04	1	107
10/03/05	2	107
01/08/04	1	107
24/06/06	1	107

Sous-requêtes (subqueries)

Exercices :

1. Afficher le nom et la date d'engagement de tous les employés du même département que celui de 'BLAKE', celui-ci étant exclu

```
SQL> SELECT last_name,hire_date,department_id FROM employees
2 WHERE department_id = <SELECT department_id FROM employees WHERE last_name
=' BLAKE'>
3 AND last_name <> 'BLAKE'
4 ORDER BY last_name;

no rows selected
```

Pour le nom : 'Perkins'

```
SQL> SELECT last_name,hire_date,department_id FROM employees
2 WHERE department_id = (SELECT department_id FROM employees WHERE last_name
='Perkins')
3 AND last_name <> 'Perkins'
4 ORDER BY last_name;
```

LAST_NAME	HIRE_DAT	DEPARTMENT_ID
Atkinson	30/10/05	50
Bell	04/02/04	50
Bissot	20/08/05	50
Bull	20/02/05	50
Cabrio	07/02/07	50
Chung	14/06/05	50
Davies	29/01/05	50
Dellinger	24/06/06	50
Dilly	13/08/05	50
Everett	03/03/05	50
Feeney	23/05/06	50

LAST_NAME	HIRE_DAT	DEPARTMENT_ID
Fleaur	23/02/06	50
Fripp	10/04/05	50
Gates	11/07/06	50
Gee	12/12/07	50
Geoni	03/02/08	50
Grant	13/01/08	50
Jones	17/03/07	50
Kaufling	01/05/03	50
Ladwig	14/07/03	50
Landry	14/01/07	50
Mallin	14/06/04	50

LAST_NAME	HIRE_DAT	DEPARTMENT_ID
Markle	08/03/08	50
Marlow	16/02/05	50
Matos	15/03/06	50
McCain	01/07/06	50

- Afficher le numéro et le nom de tous les employés qui ont un salaire supérieur au salaire moyen; trier par ordre de salaire décroissant

```
SQL> SELECT employee_id,last_name FROM employees
2 WHERE salary > (SELECT AVG(salary) FROM employees)
3 ORDER BY salary DESC;
```

EMPLOYEE_ID	LAST_NAME
-------------	-----------

100	King
101	Kochhar
102	De Haan
145	Russell
146	Partners
201	Hartstein
108	Greenberg
205	Higgins
147	Errazuriz
168	Ozer
174	Abel

EMPLOYEE_ID	LAST_NAME
-------------	-----------

114	Raphaely
148	Cambrault
162	Uishney
149	Zlotkey
150	Tucker
169	Bloom
156	King
204	Baer
170	Fox
151	Bernstein
163	Greene

EMPLOYEE_ID	LAST_NAME
-------------	-----------

157	Sully
109	Faviet
103	Hunold
152	Hall
159	M-Fess

3. Afficher le numéro et le nom de tous les employés qui travaillent dans un département contenant au moins un employé dont le nom contient 't'

```
SQL> SELECT employee_id,last_name FROM employees
2 WHERE department_id = ANY(SELECT department_id FROM employees WHERE last_name LIKE '<?tz?>') ;
```

EMPLOYEE_ID	LAST_NAME
-------------	-----------

199	Grant
198	OConnell
197	Feeney
196	Walsh
195	Jones
194	McCain
193	Everett
192	Bell
191	Perkins
190	Gates
189	Dilly

EMPLOYEE_ID	LAST_NAME
-------------	-----------

188	Chung
187	Cabrio
186	Dellinger
185	Bull
184	Sarchand
183	Geoni
182	Sullivan
181	Fleaur
180	Taylor
144	Vargas
143	Matos

4. Afficher le nom de l'employé, le numéro de département et le job de tous les employés dont la localisation du département est DALLAS

```
SQL> SELECT e.last_name,d.department_id,e.job_id
  2 FROM employees e,departments d
  3 WHERE e.department_id=d.department_id
  4 AND d.location_id = ANY (SELECT location_id FROM locations WHERE city='DALL
AS');
no rows selected
```

Remplaçons 'DALLAS' par 'Oxford' :

```
SQL> SELECT e.last_name,d.department_id,e.job_id
  2 FROM employees e,departments d
  3 WHERE e.department_id=d.department_id
  4 AND d.location_id = ANY (SELECT location_id FROM locations WHERE city='Oxfo
rd');

LAST_NAME          DEPARTMENT_ID JOB_ID
-----
Russell            80 SA_MAN
Partners           80 SA_MAN
Errazuriz          80 SA_MAN
Cambrault          80 SA_MAN
Zlotkey            80 SA_MAN
Tucker             80 SA_REP
Bernstein          80 SA_REP
Hall               80 SA_REP
Olsen              80 SA_REP
Cambrault          80 SA_REP
Tuvault            80 SA_REP

LAST_NAME          DEPARTMENT_ID JOB_ID
-----
King               80 SA_REP
Sully              80 SA_REP
McEwen             80 SA_REP
Smith              80 SA_REP
Doran              80 SA_REP
Sewall             80 SA_REP
Uishney            80 SA_REP
Greene             80 SA_REP
Marvins            80 SA_REP
Lee                80 SA_REP
Ande               80 SA_REP

LAST_NAME          DEPARTMENT_ID JOB_ID
-----
Banda              80 SA_REP
Ozer               80 SA_REP
Bloom              80 SA_REP
Fox                80 SA_REP
Smith              80 SA_REP
Bates              80 SA_REP
Kumar              80 SA_REP
Abel               80 SA_REP
Hutton             80 SA_REP
Taylor             80 SA_REP
Livingston         80 SA_REP

LAST_NAME          DEPARTMENT_ID JOB_ID
-----
Johnson           80 SA_REP

34 rows selected.
```

5. Afficher le nom et le salaire des employés dirigés par KING

```
SQL> SELECT last_name,salary FROM employees
2  WHERE manager_id =ANY (SELECT employee_id FROM employees WHERE last_name='K
ing' );
```

LAST_NAME	SALARY
Kochhar	17000
De Haan	17000
Raphaely	11000
Weiss	8000
Fripp	8200
Kaufling	7900
Ullman	6500
Mourgos	5800
Russell	14000
Partners	13500
Errazuriz	12000

LAST_NAME	SALARY
Cambrault	11000
Zlotkey	10500
Hartstein	13000

14 rows selected.

6. Afficher le numéro de département, le nom et le job de tous les employés du département SALES

```
SQL> SELECT department_id,last_name,job_id FROM employees
2 WHERE department_id =ALL (SELECT department_id FROM departments WHERE depar
tment_name='Sales');

DEPARTMENT_ID LAST_NAME                JOB_ID
-----
80 Russell                               SA_MAN
80 Partners                             SA_MAN
80 Errazuriz                             SA_MAN
80 Cambrault                             SA_MAN
80 Zlotkey                               SA_MAN
80 Tucker                               SA_REP
80 Bernstein                             SA_REP
80 Hall                                 SA_REP
80 Olsen                                 SA_REP
80 Cambrault                             SA_REP
80 Tuvault                               SA_REP

DEPARTMENT_ID LAST_NAME                JOB_ID
-----
80 King                                 SA_REP
80 Sully                                 SA_REP
80 McEwen                                SA_REP
80 Smith                                 SA_REP
80 Doran                                 SA_REP
80 Sewall                                SA_REP
80 Vishney                               SA_REP
80 Greene                                SA_REP
80 Marvins                              SA_REP
80 Lee                                  SA_REP
80 Ande                                 SA_REP

DEPARTMENT_ID LAST_NAME                JOB_ID
-----
80 Banda                               SA_REP
80 Ozer                                 SA_REP
80 Bloom                                SA_REP
80 Fox                                  SA_REP
80 Smith                                SA_REP
80 Bates                                SA_REP
80 Kumar                                SA_REP
80 Abel                                 SA_REP
80 Hutton                               SA_REP
80 Taylor                               SA_REP
80 Livingston                           SA_REP

DEPARTMENT_ID LAST_NAME                JOB_ID
-----
80 Johnson                             SA_REP

34 rows selected.
```

7. Afficher le numéro, le nom et le salaire de tous les employés qui ont un salaire supérieur au salaire moyen et qui travaillent dans le département des employés dont le nom contient la lettre 'T'

```
SQL> SELECT department_id,last_name,salary FROM employees
2 WHERE salary >ALL (SELECT AVG(salary) FROM employees )
3 AND employee_id =ANY (SELECT employee_id FROM employees WHERE last_name LIK
E ('%T%'));

DEPARTMENT_ID LAST_NAME                SALARY
-----
80 Tucker                               100000
80 Tuvault                               70000
80 Taylor                               86000
```

SQL*PLUS

Gestion des données

<u>Exercices :</u>

1- Exécuter le script LAB1.sql

2- Ajouter les 5 lignes suivantes:

- 1 PATEL • RALPH -rpatel • 795
- 2 DANES -BETTY• bdanes • 860
- 3 Biri -Ben -bbiri -1100
- 4 Newman -Chad -cnewman • 750
- 5 Ropeburn • Audry -aropebur -1550

```
SQL> INSERT INTO my_employee (id, last_name, first_name, userid, salary) VALUES
(1,'PATEL','RALPH','rpatel',795);
1 row created.
SQL> INSERT INTO my_employee (id, last_name, first_name, userid, salary) VALUES
(2,'DANES','BETTY','bdanes',860);
1 row created.
SQL> INSERT INTO my_employee (id, last_name, first_name, userid, salary) VALUES
(3,'Biri','Ben','bbiri',1100);
1 row created.
SQL> INSERT INTO my_employee (id, last_name, first_name, userid, salary) VALUES
(4,'Newman','Chad','cnewman',750);
1 row created.
SQL> INSERT INTO my_employee (id, last_name, first_name, userid, salary) VALUES
(5,'Ropeburn','Audry','aropebur',1550);
1 row created.
SQL>
```

3- Vérifier que les nouvelles données sont bien MAJ

```
SQL> SELECT * FROM my_employee;
```

ID	LAST_NAME	FIRST_NAME	USERID

SALARY			

1	PATEL	RALPH	rpatel
795			
2	DANES	BETTY	bdanes
860			
3	Biri	Ben	bbiri
1100			

ID	LAST_NAME	FIRST_NAME	USERID

SALARY			

4	Newman	Chad	cnewman
750			
5	Ropeburn	Audry	aropebur
1550			

4- Rendre ces modifications permanentes

```
SQL> commit;
Commit complete.
SQL>
```

5- Modifier le champ « last_name » de l'employé 3 en Drexler

```
SQL> UPDATE my_employee SET last_name='Drexler'
2 WHERE id=3;
1 row updated.
SQL> SELECT last_name FROM my_employee WHERE id=3;
LAST_NAME
-----
Drexler
```

6- Modifier le salaire de tous les employés ayant moins de 900;
vérifier les changements

```
SQL> UPDATE my_employee SET salary=950
2 WHERE salary < 900;

3 rows updated.

SQL> SELECT salary FROM my_employee;

SALARY
-----
950
950
1100
950
1550
```

7- Effacer « Betty Danes »; rendre ce changement permanent

```
SQL> DELETE FROM my_employee
2 WHERE last_name='DANES';

1 row deleted.

SQL> COMMIT;

Commit complete.

SQL> SELECT * FROM my_employee;

ID LAST_NAME FIRST_NAME USERID
-----
SALARY
-----
1 PATEL RALPH rpatel
950
3 Drexler Ben bbiri
1100
4 Newman Chad cnewman
950

ID LAST_NAME FIRST_NAME USERID
-----
SALARY
-----
5 Ropeburn Audry aropebur
1550
```

8- Afficher le contenu de la table; définir un point intermédiaire de transaction; vider entièrement la table; vérifier que la table est bien vide; annuler ce changement; vérifier que les données sont bien là; rendre les changements définitifs.

```
SQL> SELECT * FROM my_employee;
```

ID	LAST_NAME	FIRST_NAME	USERID

SALARY			

1	PATEL	RALPH	rpatel
950			
3	Drexler	Ben	bbiri
1100			
4	Newman	Chad	cnewman
950			
ID	LAST_NAME	FIRST_NAME	USERID

SALARY			

5	Ropeburn	Audry	aropebur
1550			

```
SQL> SAVEPOINT update_done;
```

```
Savepoint created.
```

```
SQL> DELETE FROM my_employee;
```

```
4 rows deleted.
```

```
SQL> SELECT * FROM my_employee;
```

```
no rows selected
```

```
SQL> ROLLBACK TO update_done;
```

```
Rollback complete.
```

```
SQL> SELECT * FROM my_employee;
```

ID	LAST_NAME	FIRST_NAME	USERID

SALARY			

1	PATEL	RALPH	rpatel
950			
3	Drexler	Ben	bbiri
1100			
4	Newman	Chad	cnewman
950			
ID	LAST_NAME	FIRST_NAME	USERID

SALARY			

5	Ropeburn	Audry	aropebur
1550			

Gestion des tables

Exercices :

1. Créer la table *DEPARTEMENT* comme suit;

- vérifier que la table est bien créée
- *ID* *NUMBER*(7)
- *NAME* *VARCHAR2*(25)

```
SQL> CREATE TABLE DEPARTEMENT
2 < id          NUMBER(7)
3   CONSTRAINT DEPARTEMENT_id_nn NOT NULL,
4   name        VARCHAR2(25)>;

Table created.

SQL> DESCRIBE DEPARTEMENT;
Name                                     Null?      Type
-----
ID                                     NOT NULL   NUMBER(7)
NAME                                               VARCHAR2(25)
```

2. Remplir cette table avec les données de la table *DEPT*; utiliser uniquement les champs nécessaires

```
SQL> INSERT INTO DEPARTEMENT (id,name)
2 SELECT department_id,department_name FROM departments;

27 rows created.

SQL> SELECT * FROM DEPARTEMENT;

   ID NAME
-----
   10 Administration
   20 Marketing
   30 Purchasing
   40 Human Resources
   50 Shipping
   60 IT
   70 Public Relations
   80 Sales
   90 Executive
  100 Finance
  110 Accounting

   ID NAME
-----
  120 Treasury
  130 Corporate Tax
  140 Control And Credit
  150 Shareholder Services
  160 Benefits
  170 Manufacturing
  180 Construction
  190 Contracting
  200 Operations
  210 IT Support
  220 NOC

   ID NAME
-----
  230 IT Helpdesk
  240 Government Sales
  250 Retail Sales
  260 Recruiting
  270 Payroll

27 rows selected.
```

3. Créer la table *EMPLOYEE* comme suit;

- vérifier que la table est bien créée
- ID NUMBER(7)
- LAST_NAME VARCHAR2(25)
- FIRST_NAME VARCHAR2(25)
- DEPTID NUMBER(7)

```
SQL> CREATE TABLE EMPLOYEE
2  (id          NUMBER(7)
3   CONSTRAINT EMPLOYEE_id_nn NOT NULL,
4   last_name   VARCHAR2(25),
5   first_name  VARCHAR2(25),
6   deptid     NUMBER(7));

Table created.

SQL> DESCRIB EMPLOYEE;
Name                               Null?    Type
-----
ID                                NOT NULL NUMBER(7)
LAST_NAME                        VARCHAR2(25)
FIRST_NAME                       VARCHAR2(25)
DEPTID                           NUMBER(7)
```

4. Augmenter la taille du champ LAST_NAME à 50 caractères;
confirmer la modification

```
SQL> ALTER TABLE EMPLOYEE
2  MODIFY (last_name VARCHAR2(50));

Table altered.
```

5. Examiner le DATA DICTIONARY pour voir que les tables sont bien créées

```
SQL> SELECT TABLE_NAME FROM USER_TABLES;

TABLE_NAME
-----
REGIONS
LOCATIONS
DEPARTMENTS
JOBS
EMPLOYEES
JOB_HISTORY
MY_EMPLOYEE
DEPARTEMENT
EMPLOYEE
COUNTRIES

10 rows selected.
```

6. Créer la table EMPLOYEE2 basée sur la structure de la table EMP mais ne reprendre que les colonnes suivantes en changeant leur nom:
- EMPNO ID
 - ENAME LAST_NAME
 - DEPTNO DEPTID

```
SQL> CREATE TABLE EMPLOYEE2 (LAST_NAME, ID, DEPTID)
2 AS SELECT last_name, employee_id, department_id FROM employees;
Table created.
SQL> DESC EMPLOYEE2;
```

Name	Null?	Type
LAST_NAME	NOT NULL	VARCHAR2(25)
ID		NUMBER(6)
DEPTID		NUMBER(4)

7. Effacer la table EMPLOYEE

```
SQL> DROP TABLE EMPLOYEE;
Table dropped.
```

8. Renommer la table EMPLOYEE2 en EMPLOYEE

```
SQL> RENAME EMPLOYEE2 TO EMPLOYEE ;
Table renamed.
```

Contraintes

Exercices :

1. Définir le champ ID de la table EMPLOYEE comme PK

```
SQL> ALTER TABLE EMPLOYEE ADD CONSTRAINT employee_id_pk PRIMARY KEY <id>;
Table altered.
```

2. Définir également le champ ID de la table DEPARTMENT comme PK

```
SQL> ALTER TABLE DEPARTEMENT ADD CONSTRAINT department_id_pk PRIMARY KEY <id>;
Table altered.
```

3. Ajouter une FK sur la table EMPLOYEE qui assure qu'un employé appartient nécessairement à un département existant

```
SQL> ALTER TABLE EMPLOYEE
  2 ADD CONSTRAINT employee_deptid_fk FOREIGN KEY(deptid)
  3 REFERENCES DEPARTEMENT(id);
Table altered.
```

4. Visualiser les contraintes créées dans le DATA DICTIONARY

```
SQL> SELECT constraint_name, constraint_type, search_condition
  2 FROM user_constraints;

CONSTRAINT_NAME          C SEARCH_CONDITION
-----
REGION_ID_NN              C "REGION_ID" IS NOT N
                           ULL
REG_ID_PK                 P
COUNTRY_ID_NN             C "COUNTRY_ID" IS NOT
                           NULL
COUNTRY_C_ID_PK           P
COUNTR_REG_FK             R
LOC_CITY_NN               C "CITY" IS NOT NULL
LOC_ID_PK                 P
CONSTRAINT_NAME          C SEARCH_CONDITION
-----
LOC_C_ID_FK               R
DEPT_NAME_NN              C "DEPARTMENT_NAME" IS
                           NOT NULL
```

5. *Ajouter dans la table EMPLOYEE une colonne SALARY de type NUMBER(7,2)*

```
SQL> ALTER TABLE EMPLOYEE  
2 ADD (salary NUMBER(7,2));  
Table altered.
```

6. *Y associer une contrainte vérifiant que le salaire n'est pas inférieur à 500 et ne dépasse pas 5000*

```
SQL> ALTER TABLE EMPLOYEE  
2 ADD CONSTRAINT employee_salary_ck CHECK (salary BETWEEN 500 AND 5000);  
Table altered.
```

Vues

Exercices :

1. Créer une vue EMP_VU basée sur le n° et le nom de l'employé et son numéro de département

```
SQL> CREATE VIEW EMP_VU
  2 AS SELECT employee_id,last_name,department_id
  3 FROM employees;

View created.

SQL> DESC emp_vu;
Name                               Null?    Type
-----
EMPLOYEE_ID                        NOT NULL NUMBER(6)
LAST_NAME                          NOT NULL VARCHAR2(25)
DEPARTMENT_ID                      NUMBER(4)
```

2. Changer l'en-tête des noms des employés en EMPLOYEE

```
SQL> CREATE OR REPLACE VIEW emp_vu
  2 AS SELECT employee_id,last_name employee,department_id
  3 FROM employees;

View created.

SQL> DESC emp_vu;
Name                               Null?    Type
-----
EMPLOYEE_ID                        NOT NULL NUMBER(6)
EMPLOYEE                          NOT NULL VARCHAR2(25)
DEPARTMENT_ID                      NUMBER(4)
```

3. Afficher le contenu de cette vue

```
SQL> SELECT * FROM emp_vu;
```

EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
105	Austin	60
106	Pataballa	60
107	Lorentz	60
108	Greenberg	100
109	Faviet	100
110	Chen	100
EMPLOYEE_ID	EMPLOYEE	DEPARTMENT_ID
111	Sciarra	100
112	Urman	100
113	Popp	100
114	Raphaely	30
115	Khoo	30

4. Voir cette vue dans le DATA DICTIONARY

```
SQL> SELECT view_name
2 FROM USER_VIEWS;
```

VIEW_NAME
EMP_DETAILS_VIEW
EMP_VU

5. Utiliser la vue pour extraire le nom des employés et leur numéro de département

```
SQL> SELECT employee, department_id
2 FROM emp_vu;
```

EMPLOYEE	DEPARTMENT_ID
King	90
Kochhar	90
De Haan	90
Hunold	60
Ernst	60
Austin	60
Pataballa	60
Lorentz	60
Greenberg	100
Faviet	100
Chen	100
EMPLOYEE	DEPARTMENT_ID
Sciarra	100
Urman	100
Popp	100
Raphaely	30
Khoo	30
Baida	30
Tobias	30
Himuro	30
Colmenares	30
Weiss	50
Frapp	50
EMPLOYEE	DEPARTMENT_ID
Kaufling	50
Uollman	50
Mourgos	50

6. Créer une vue DEPT20 contenant pour les employés du département 20:

- le numéro (libeller EMPID)
- le nom (libeller EMPNAME)
- le n° de département (libeller DEPTID)
- Interdire de pouvoir déplacer un employé dans un autre département au travers de cette vue

```
SQL> CREATE VIEW DEPT20
  2 AS SELECT employee_id EMPID, last_name EMPNAME, department_id DEPTID
  3 FROM employees
  4 WHERE department_id=20 WITH CHECK OPTION CONSTRAINT DEPT20_ck;

View created.

SQL> DESC DEPT20;
Name                               Null?    Type
-----
EMPID                               NOT NULL NUMBER(6)
EMPNAME                             NOT NULL VARCHAR2(25)
DEPTID                              NOT NULL  NUMBER(4)
```

7. Utiliser la vue et essayer de déplacer un employé dans le département 30

```
SQL> SELECT * FROM DEPT20;

  EMPID EMPNAME      DEPTID
-----
    201 Hartstein    20
    202 Fay           20

SQL> UPDATE DEPT20
  2 SET DEPTID=30 WHERE EMPID=202;
UPDATE DEPT20
*
ERROR at line 1:
ORA-01402: view WITH CHECK OPTION where-clause violation
```

8. Créer une vue SAL_VU basée sur

- le nom des employés (libeller Employee)
- le nom de son département (libeller Department)
- son salaire (libeller Salary)
- le niveau de son salaire (libeller Grade)

```

SQL> CREATE VIEW SAL_UU
  2 AS SELECT e.last_name EMPLOYEE, d.department_name DEPARTMENT, e.salary Sala
ry, j.job_title Grade
  3 FROM employees e,departments d,jobs j
  4 WHERE e.department_id=d.department_id AND e.job_id=j.job_id;

View created.

SQL> DESC SAL_UU;

```

Name	Null?	Type
EMPLOYEE	NOT NULL	VARCHAR2(25)
DEPARTMENT	NOT NULL	VARCHAR2(30)
SALARY		NUMBER(8,2)
GRADE	NOT NULL	VARCHAR2(35)

9. Voir le contenu de cette vue

SQL> SELECT * FROM SAL_UU;

EMPLOYEE	DEPARTMENT	SALARY	GRADE
Whalen	Administration	4400	Administration Assistant
Fay	Marketing	6000	Marketing Representative
Hartstein	Marketing	13000	Marketing Manager
Raphaely	Purchasing	11000	Purchasing Manager
Colmenares	Purchasing	2500	Purchasing Clerk
Uimuro	Purchasing	2600	Purchasing Clerk
Tobias	Purchasing	2800	Purchasing Clerk
EMPLOYEE	DEPARTMENT	SALARY	GRADE
Khoo	Purchasing	3100	Purchasing Clerk
Baida	Purchasing	2900	Purchasing Clerk
Mauris	Human Resources	6500	Human Resources Representative
Weiss	Shipping	8000	Stock Manager
Fripp	Shipping	8200	Stock Manager
Kaufling	Shipping	7900	Stock Manager
Jollman	Shipping	6500	Stock Manager
Mourgos	Shipping	5800	Stock Manager
Seo	Shipping	2700	Stock Clerk
EMPLOYEE	DEPARTMENT	SALARY	GRADE
Stiles	Shipping	3200	Stock Clerk
Ladwig	Shipping	3600	Stock Clerk
Philtanker	Shipping	2200	Stock Clerk
Gee	Shipping	2400	Stock Clerk
Rogers	Shipping	2900	Stock Clerk
Mallin	Shipping	3300	Stock Clerk
Jargas	Shipping	2500	Stock Clerk
Marlow	Shipping	2500	Stock Clerk
Atkinson	Shipping	2800	Stock Clerk
Bissot	Shipping	3300	Stock Clerk
Markle	Shipping	2200	Stock Clerk
EMPLOYEE	DEPARTMENT	SALARY	GRADE
Landry	Shipping	2400	Stock Clerk
Mikkilineni	Shipping	2700	Stock Clerk
Mayer	Shipping	3200	Stock Clerk
Patel	Shipping	2500	Stock Clerk

Sécurité d'accès

Exercices :

1. Quelle commande faut-il utiliser pour changer votre mot de passe?

```
SQL> ALTER USER user1 IDENTIFIED 199994 ;
User altered.
SQL>
```

2. Donner les droits d'accès en lecture sur la table DEPT à l'utilisateur USER1

```
SQL> GRANT select ON departments TO user1;
Grant succeeded.
```

3. Comment afficher toutes les lignes de la table DEPT

```
SQL> disconnect;
Disconnected from Oracle Database 11g Express Edition Release 11.2.0.2.0 - Production
SQL> connect;
Enter user-name: user1
Enter password:
Connected.
SQL> SELECT * FROM hr.departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
100	Finance	108	1700
110	Accounting	205	1700

```
DEPARTMENT_ID DEPARTMENT_NAME MANAGER_ID LOCATION_ID
```

4. Créer un synonyme DEP sur la table DEPT

```
SQL> CREATE SYNONYM dep FOR departments;
Synonym created.
```

5. Afficher toutes les données de cette table en utilisant le synonyme

```
SQL> SELECT *  
2 FROM dep;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
100	Finance	108	1700
110	Accounting	205	1700

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
120	Treasury		1700
130	Corporate Tax		1700
140	Control And Credit		1700

6. Afficher les informations sur les tables auxquelles vous avez accès

```
SQL> SELECT table_name, privilege  
2 FROM user_tab_privs;
```

TABLE_NAME	PRIVILEGE
DBMS_STATS	EXECUTE

```
SQL>
```