



Département Informatique
ADIA-IISE/2023-2024

Module Bases de données et modélisation
TP 2

Travaux pratiques 1
Langage de manipulation des données

Réaliser par :
TARZOUKT Abderrahim
EL RHYATI Aymane

Encadré par :
M.OUKDACH Yassine

LDD

[Q1] Construire un utilisateur nommé Etud avec quota dans l'espace de table users.

```
SQL> CREATE USER Etud IDENTIFIED BY abdou ;
User created.
SQL> ALTER USER Etud QUOTA UNLIMITED ON users;
User altered.
```

[Q2] Accorder à cet utilisateur les privilèges 'all privileges'.

```
SQL> GRANT ALL PRIVILEGES TO Etud;
Grant succeeded.
```

[Q3] Indiquer les clés primaires et les clés étrangères de chaque table.

Table Etudiant : Clés primaires :CODEETUDIANT
Clés étrangères : -

Table RESULTAT: Clés primaires :CODEETUDIANT - CODECOURS
Clés étrangères :CODEETUDIANT - CODECOURS

Table COURS: Clés primaires :CODECOURS
Clés étrangères : -

Table ENSEIGNANT : Clés primaires :CODEENSEIGNANT
Clés étrangères :-

Table CHARGE: Clés primaires : CODEENSEIGNANT - CODECOURS
Clés étrangères : CODEENSEIGNANT - CODECOURS

[Q4] Construire les tables dans le schéma de Etud.

- Table etudiant

```
SQL> CREATE TABLE etudiant ( codeetudiant varchar(10) PRIMARY KEY , nomeetudiant varchar(20) , prenom  
2  
SQL> create table etudiant ( codeetudiant varchar(10) PRIMARY KEY , nomeetudiant varchar(20) , prenom  
etudiant varchar(20), datenaissance date , ville varchar(10) ) ;  
Table created.
```

- Table resultat

```
SQL> CREATE TABLE resultat (  
2     codeetudiant VARCHAR(10),  
3     codecours VARCHAR(10),  
4     note NUMBER(5,2),  
5     PRIMARY KEY (codeetudiant, codecours),  
6     CONSTRAINT fk_codeetudiant FOREIGN KEY (codeetudiant) REFERENCES etudiant(codeetudiant),  
7     CONSTRAINT fk_codecours FOREIGN KEY (codecours) REFERENCES cours(codecours)  
8 );  
Table created.
```

- Table cours

```
SQL> CREATE TABLE cours (  
2     codecours VARCHAR(10) PRIMARY KEY,  
3     intitule VARCHAR2(20) ,  
4     nbheures number(2)  
5 );  
Table created.
```

- Table charge

```
SQL> CREATE TABLE charge (  
2     codecours VARCHAR(10),  
3     codeenseignant VARCHAR(10),  
4     PRIMARY KEY (codecours, codeenseignant),  
5     CONSTRAINT fk_codecours FOREIGN KEY (codecours) REFERENCES cours(codecours),  
6     CONSTRAINT fk_codeenseignant FOREIGN KEY (codeenseignant) REFERENCES enseignant(codeenseig  
nant)  
7 );  
Table created.
```

- Table enseignant

```
SQL> CREATE TABLE enseignant (  
2     codeenseignant VARCHAR(10) PRIMARY KEY,  
3     nomenseignant VARCHAR(20),  
4     prenomenseignant VARCHAR(20),  
5     specialite VARCHAR(20)  
6 );  
Table created.
```

LMD

[Q5] Insérer des données dans les tables en utilisant la commande Insert (insérer 5 lignes au minimum).

- Table etudiant

```
SQL> INSERT INTO etudiant (codeetudiant, nometudiant, prenometudiant, datenaissance, ville)
2 VALUES
3 ('E001', 'Dupont', 'Jean', TO_DATE('1990-01-15', 'YYYY-MM-DD'), 'Agadir');

1 row created.

SQL> INSERT INTO etudiant (codeetudiant, nometudiant, prenometudiant, datenaissance, ville)
2 VALUES
3 ('E002', 'Martin', 'Sophie', TO_DATE('1992-05-22', 'YYYY-MM-DD'), 'Marrakech'),
4
SQL>
SQL>
SQL>
SQL>
SQL>
SQL>
SQL> INSERT INTO etudiant (codeetudiant, nometudiant, prenometudiant, datenaissance, ville)
2 VALUES
3 ('E003', 'Martin', 'Sophie', TO_DATE('1992-05-22', 'YYYY-MM-DD'), 'Marrakech');

1 row created.

SQL> INSERT INTO etudiant (codeetudiant, nometudiant, prenometudiant, datenaissance, ville)
2 VALUES
3 ('E003', 'Lefevre', 'Pierre', TO_DATE('1988-11-03', 'YYYY-MM-DD'), 'casa');

1 row created.

SQL> INSERT INTO etudiant (codeetudiant, nometudiant, prenometudiant, datenaissance, ville)
2 VALUES
3 ('E004', 'Leclerc', 'Marie', TO_DATE('1995-07-18', 'YYYY-MM-DD'), 'fes');

1 row created.

SQL> INSERT INTO etudiant (codeetudiant, nometudiant, prenometudiant, datenaissance, ville)
2 VALUES
3 ('E005', 'Dubois', 'Alice', TO_DATE('1993-09-28', 'YYYY-MM-DD'), 'Errachidia');

1 row created.
```

- Table resultat

```
SQL> INSERT INTO Resultat (codeEtudiant, codeCours, note)
  2 VALUES ('E001', 'C0001', 85.5);
```

1 row created.

```
SQL> INSERT INTO Resultat (codeEtudiant, codeCours, note)
  2 VALUES ('E002', 'C0002', 45);
```

1 row created.

```
SQL> INSERT INTO Resultat (codeEtudiant, codeCours, note)
  2 VALUES ('E003', 'C0003', 34);
```

1 row created.

```
SQL> INSERT INTO Resultat (codeEtudiant, codeCours, note)
  2 VALUES ('E004', 'C0004', 89);
```

1 row created.

```
SQL> INSERT INTO Resultat (codeEtudiant, codeCours, note)
  2 VALUES ('E005', 'C0005', 100);
```

1 row created.

- Table cours

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2 VALUES ('C0001', 'Mathematics', 60);
```

1 row created.

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2 VALUES ('C0002', 'Physics', 45);
```

1 row created.

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2 VALUES ('C0003', 'Computer Science', 75);
```

1 row created.

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2 VALUES ('C0004', 'JAVA', 23);
```

1 row created.

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2 VALUES ('C0005', 'BD', 48);
```

1 row created.

SQL>

- Table charge

```
SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2 VALUES ('C0001', 'ENS001');
```

1 row created.

```
SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2 VALUES ('C0002', 'ENS002');
```

1 row created.

```
SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2 VALUES ('C0003', 'ENS003');
```

1 row created.

```
SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2 VALUES ('C0004', 'ENS002');
```

1 row created.

```
SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2 VALUES ('C0005', 'ENS001');
```

1 row created.

```
SQL> COMMIT;
```

Commit complete.

- Table enseignant

```
SQL> INSERT INTO enseignant (codeenseignant, nomenseignant, prenomenseignant, specialite)
  2 VALUES ('ENS001', 'Martin', 'Jean', 'Mathématiques');
```

1 row created.

```
SQL> INSERT INTO enseignant (codeenseignant, nomenseignant, prenomenseignant, specialite)
  2 VALUES ('ENS002', 'Dupont', 'Sophie', 'Physique');
```

1 row created.

```
SQL> INSERT INTO enseignant (codeenseignant, nomenseignant, prenomenseignant, specialite)
  2 VALUES ('ENS004', 'Lefevre', 'Pierre', 'Informatique');
```

1 row created.

```
SQL> INSERT INTO enseignant (codeenseignant, nomenseignant, prenomenseignant, specialite)
  2 VALUES ('ENS003', 'Lefevre', 'Pierre', 'Informatique');
```

1 row created.

```
SQL> INSERT INTO enseignant (codeenseignant, nomenseignant, prenomenseignant, specialite)
  2 VALUES ('ENS005', 'Lefevre', 'Pierre', 'Informatique');
```

1 row created.

[Q6] Faire les mises à jour suivantes (n'oubliez pas de valider les mises à jour par la commande commit) :

6.1. Augmenter de 5% toutes les notes des étudiant.

```
SQL> UPDATE Resultat
  2 SET note = note * 1.05;
```

5 rows updated.

```
SQL> commit ;
```

Commit complete.

6.2. Changer le nom du cours 2 par 'SGBD'.

```
SQL> UPDATE Cours
  2 SET intitule = 'SGBD'
  3 WHERE codeCours = 'C0002';
```

1 row updated.

```
SQL> commit ;
```

Commit complete.

6.3. Supprimer le cours n° 5.

```
SQL> DELETE FROM resultat
  2 WHERE codecours = 'C0005';
```

1 row deleted.

```
SQL> DELETE FROM cours
  2 WHERE codecours = 'C0005';
```

```
DELETE FROM cours
```

```
*
```

ERROR at line 1:

ORA-02292: integrity constraint (E
found

```
SQL> DELETE FROM charge
  2 WHERE codecours = 'C0005';
```

1 row deleted.

```
SQL> DELETE FROM cours
  2 WHERE codecours = 'C0005';
```

1 row deleted.

LID

[Q7] Afficher séparément les tables déjà créées.

- Table etudiant

```
SQL> SELECT * FROM etudiant;
```

CODEETUDIA	NOMETUDIANT	PRENOMETUDIANT	DATENAISS	VILLE
E001	Dupont	Jean	15-JAN-90	Agadir
E002	Martin	Sophie	22-MAY-92	Marrakech
E003	Lefevre	Pierre	03-NOV-88	casa
E004	Leclerc	Marie	18-JUL-95	fes
E005	Dubois	Alice	28-SEP-93	Errachidia

- Table resultat

```
SQL> SELECT * FROM resultat;
```

CODEETUDIA	CODECOURS	NOTE
E001	C0001	94.27
E002	C0002	49.61
E003	C0003	37.49
E004	C0004	98.12

- Table cours

```
SQL> SELECT * FROM cours;
```

CODECOURS	INTITULE	NBHEURES
C0001	Mathematics	60
C0002	SGBD	45
C0003	Computer Science	75
C0004	JAVA	23

- Table charge

```
SQL> SELECT * FROM charge;
```

CODECOURS	CODEENSEIG
C0001	ENS001
C0002	ENS002
C0003	ENS003
C0004	ENS002

- Table enseignant


```
SQL> SELECT * FROM enseignant;
```

CODEENSEIG	NOMENSEIGNANT	PRENOMENSEIGNANT	SPECIALITE
ENS001	Martin	Jean	Mathématiques
ENS002	Dupont	Sophie	Physique
ENS004	Lefevre	Pierre	Informatique
ENS003	Lefevre	Pierre	Informatique
ENS005	Lefevre	Pierre	Informatique

[Q8] Afficher le nom et la ville des étudiants.

```
SQL> SELECT nometudiant, ville  
2 FROM ETUDIANT;
```

NOMETUDIANT	VILLE
Dupont	Agadir
Martin	Marrakech
Lefevre	casa
Leclerc	fes
Dubois	Errachidia

[Q9] Afficher le nom et la ville des étudiants ordonnés par nom.

```
SQL> SELECT nometudiant, ville  
2 FROM ETUDIANT  
3 ORDER BY nometudiant;
```

NOMETUDIANT	VILLE
Dubois	Errachidia
Dupont	Agadir
Leclerc	fes
Lefevre	casa
Martin	Marrakech

[Q10] Afficher le nom des étudiants des villes agadir et tiznit.

```
SQL> SELECT nometudiant  
2 FROM ETUDIANT  
3 WHERE ville IN ('Agadir', 'Tiznit');
```

NOMETUDIANT
Dupont

[Q11] Afficher le nom et la ville des étudiants des villes agadir, tiznit

et taroudant ordonnés par ville et par nom.

```
SQL> SELECT nometudiant, ville
2 FROM ETUDIANT
3 WHERE ville IN ('Agadir', 'Tiznit', 'Taroudant')
4 ORDER BY ville, nometudiant;
```

NOMETUDIANT	VILLE
Dupont	Agadir

[Q12] Afficher le nom et l'âge des étudiants ordonnés par ville et par âge descendant.

```
SQL> SELECT NOMETUDIANT, DATENAissance, ville
2 FROM ETUDIANT
3 ORDER BY ville, DATENAissance DESC;
```

NOMETUDIANT	DATENAissance	VILLE
Dupont	15-JAN-90	Agadir
Dubois	28-SEP-93	Errachidia
Martin	22-MAY-92	Marrakech
Lefevre	03-NOV-88	casa
Leclerc	18-JUL-95	fes

SQL>

[Q13] Afficher le nom et l'âge des étudiants dont l'âge est compris entre 17 et 20, et ordonnés par âge descendant.

```
SQL> select nom, dateNaissance from Etudiant
2 where dateNaissance between '01-01-2002' and '31-12-2005'
3 order by dateNaissance desc;
```

[Q14] Afficher le nom et la ville des étudiants venants de Casa, Taroudant ou Safi.

```
SQL> SELECT nometudiant, ville
2 FROM ETUDIANT
3 WHERE ville IN ('Casa', 'Taroudant', 'Safi');
```

no rows selected

[Q15] Lister les étudiants dont le nom contient 'Ben' ou 'oui'.

```
SQL> SELECT nometudiant, ville  
2 FROM ETUDIANT  
3 WHERE nometudiant LIKE '%Ben%' OR nometudiant LIKE '%oui%';  
  
no rows selected
```