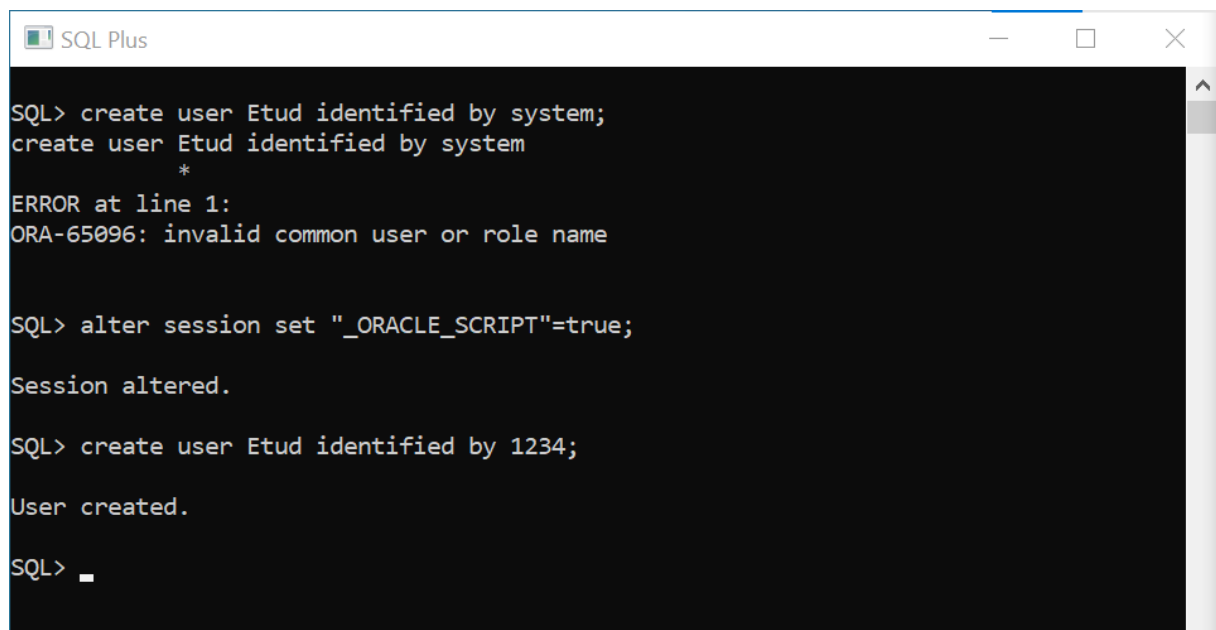


Base de données et modélisation

TP2

LDD :

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



```
SQL> create user Etud identified by system;
create user Etud identified by system
      *
ERROR at line 1:
ORA-65096: invalid common user or role name

SQL> alter session set "_ORACLE_SCRIPT"=true;

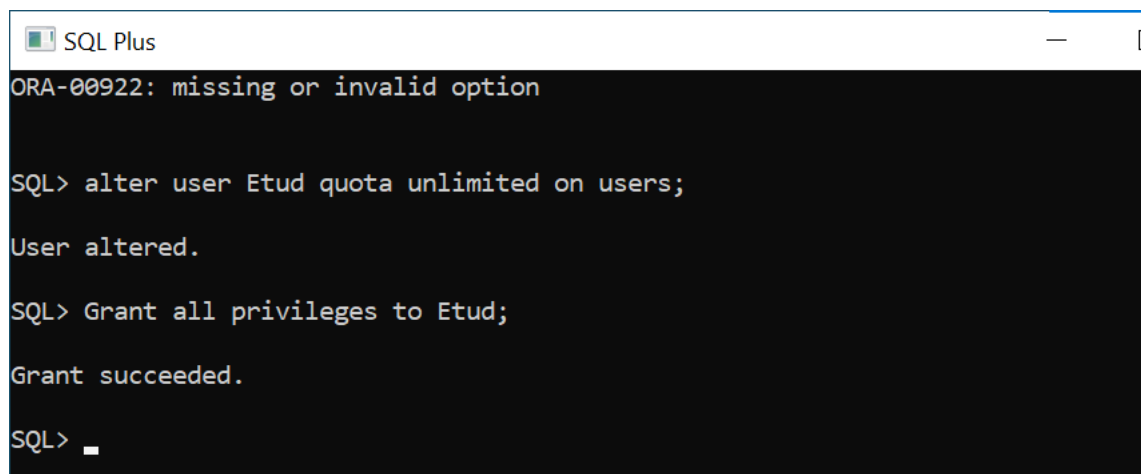
Session altered.

SQL> create user Etud identified by 1234;

User created.

SQL> _
```

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



```
ORA-00922: missing or invalid option

SQL> alter user Etud quota unlimited on users;

User altered.

SQL> Grant all privileges to Etud;

Grant succeeded.

SQL> _
```

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

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

Creation de la table Etudiant :

```
SQL> Create table etudiant(  
2  codeetudiant varchar(10) primary key,  
3  nometudiant varchar(20),  
4  prenometudiant varchar(20),  
5  datenaissance date,  
6  ville varchar(10)  
7  );
```

Table created.

Creation de la 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.

Creation de la table cours :

```
SQL Plus  
4  
SQL> Create table cours(  
2  codecours varchar(10) primary key,  
3  intitule varchar(20),  
4  nbheures number(2)  
5  );
```

Table created.

Creation de la table resultat :

```
SQL> Create table resultat(  
2  codeetudiant varchar(10),  
3  codecours varchar(10),  
4  constraint pk_resultat primary key (codeetudiant,codecours),  
5  constraint fk_resultat1 foreign key (codeetudiant) references etudiant(codeetudiant),  
6  constraint fk_resultat2 foreign key (codecours) references cours(codecours),  
7  note number(5,2)  
8  );
```

Table created.

Creation de la table Charge :

```
SQL> Create table charge(
  2  codecours varchar(10),
  3  codeenseignant varchar(10),
  4  constraint pk_charge primary key (codecours,codeenseignant),
  5  constraint fk_charge1 foreign key (codecours) references cours(codecours),
  6  constraint fk_charge2 foreign key (codeenseignant) references enseignant(codeenseignant)
  7  );

Table created.
```

LMD :

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

Insertion de la table Etudiant :

```
SQL> INSERT INTO Etudiant (codeEtudiant, nomEtudiant, prenomEtudiant, dateNaissance, ville)
  2  VALUES
  3  ('E001', 'Dupont', 'Alice', TO_DATE('1998-05-15', 'YYYY-MM-DD'), 'Paris');

1 row created.
```

```
SQL>
SQL> INSERT INTO Etudiant (codeEtudiant, nomEtudiant, prenomEtudiant, dateNaissance, ville)
  2  VALUES
  3  ('E002', 'Martin', 'Bob', TO_DATE('1997-09-22', 'YYYY-MM-DD'), 'Lyon');

1 row created.
```

```
SQL> INSERT INTO Etudiant (codeEtudiant, nomEtudiant, prenomEtudiant, dateNaissance, ville)
  2  VALUES
  3  ('E003', 'Garcia', 'Claire', TO_DATE('1999-03-10', 'YYYY-MM-DD'), 'Marseille');

1 row created.
```

```
SQL> INSERT INTO Etudiant (codeEtudiant, nomEtudiant, prenomEtudiant, dateNaissance, ville)
  2  VALUES
  3  ('E004', 'Lefevre', 'David', TO_DATE('1996-11-28', 'YYYY-MM-DD'), 'Toulouse');

1 row created.
```

```
SQL> INSERT INTO Etudiant (codeEtudiant, nomEtudiant, prenomEtudiant, dateNaissance, ville)
  2  VALUES
  3  ('E005', 'Fournier', 'Emma', TO_DATE('1998-07-07', 'YYYY-MM-DD'), 'Nice');

1 row created.
```

Insertion de table Enseignant :

```
SQL> INSERT INTO Enseignant (codeEnseignant, nomEnseignant, prenomEnseignant, specialite)
  2  VALUES
  3  ('E001', 'Smith', 'John', 'Mathematics');

1 row created.

SQL> INSERT INTO Enseignant (codeEnseignant, nomEnseignant, prenomEnseignant, specialite)
  2  VALUES
  3  ('E002', 'Johnson', 'Emma', 'Computer Science');

1 row created.

SQL> INSERT INTO Enseignant (codeEnseignant, nomEnseignant, prenomEnseignant, specialite)
  2  VALUES
  3  ('E003', 'Anderson', 'Michael', 'Physics');

1 row created.

SQL> INSERT INTO Enseignant (codeEnseignant, nomEnseignant, prenomEnseignant, specialite)
  2  VALUES
  3  ('E004', 'Brown', 'Sophie', 'History');

1 row created.

SQL> INSERT INTO Enseignant (codeEnseignant, nomEnseignant, prenomEnseignant, specialite)
  2  VALUES
  3  ('E005', 'Taylor', 'David', 'French Literature');

1 row created.

SQL>
```

Insertion de table cours :

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2 VALUES
  3     ('C001', 'Mathematics', 48);
```

1 row created.

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2 VALUES
  3     ('C002', 'Computer Science', 36),
  4
```

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2 VALUES
  3     ('C002', 'Computer Science', 36);
```

1 row created.

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2 VALUES
  3     ('C003', 'History', 30);
```

1 row created.

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2 VALUES
  3     ('C004', 'Physics', 42);
```

1 row created.

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2 VALUES
  3     ('C005', 'English Literature', 45);
```

1 row created.

Insertion de la table Resultat :

```
SQL> INSERT INTO Resultat (codeEtudiant, codeCours, note)
  2  VALUES
  3  ('E001', 'C001', 15);
```

1 row created.

```
SQL> INSERT INTO Resultat (codeEtudiant, codeCours, note)
  2  VALUES
  3  ('E002', 'C002', 12);
```

1 row created.

```
SQL> INSERT INTO Resultat (codeEtudiant, codeCours, note)
  2  VALUES
  3  ('E003', 'C003', 18);
```

1 row created.

```
SQL> INSERT INTO Resultat (codeEtudiant, codeCours, note)
  2  VALUES
  3  ('E004', 'C004', 19);
```

1 row created.

```
SQL> INSERT INTO Resultat (codeEtudiant, codeCours, note)
  2  VALUES
  3  ('E005', 'C005', 17);
```

1 row created.

SQL>

Insertion de la table Charge :

```
SQL Plus

1 row created.

SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2 VALUES
  3 ('C001', 'E001');

1 row created.

SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2 VALUES
  3 ('C002', 'E002');

1 row created.

SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2 VALUES
  3 ('C003', 'E003');

1 row created.

SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2 VALUES
  3
SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2 VALUES
  3 ('C004', 'E004');

1 row created.

SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2 VALUES
  3 ('C005', 'E005');

1 row created.
```

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

- Augmentation de 5% de toutes les notes des étudiant :

```
SQL> update resultat SET note=note*1.05;

5 rows updated.

SQL> COMMIT;
```

- Changement de nom du cours 2 par 'SGBD' :

```
SQL> UPDATE cours SET intitule='SGBD' WHERE codecours = 'C002';
1 row updated.

SQL> commit
  2  commit;
commit
*
ERROR at line 2:
ORA-02185: a token other than WORK follows COMMIT

SQL> COMMIT;

Commit complete.
```

- Suppression du cours n° 5 :

```
SQL> DELETE FROM cours WHERE codecours='C005';
1 row deleted.

SQL> COMMIT;

Commit complete.
```


LID :

[Q7] Afficher séparément les tables déjà créées :

```
SQL> desc cours
```

| Name | Null? | Type |
|-----------|----------|--------------|
| CODECOURS | NOT NULL | VARCHAR2(10) |
| INTITULE | | VARCHAR2(20) |
| NBHEURES | | NUMBER(2) |

```
SQL> desc resultat
```

| Name | Null? | Type |
|--------------|----------|--------------|
| CODEETUDIANT | NOT NULL | VARCHAR2(10) |
| CODECOURS | NOT NULL | VARCHAR2(10) |
| NOTE | | NUMBER(5,2) |

```
SQL> desc enseignant
```

| Name | Null? | Type |
|------------------|----------|--------------|
| CODEENSEIGNANT | NOT NULL | VARCHAR2(10) |
| NOMENSEIGNANT | | VARCHAR2(20) |
| PRENOMENSEIGNANT | | VARCHAR2(20) |
| SPECIALITE | | VARCHAR2(20) |

```
SQL> desc charge
```

| Name | Null? | Type |
|----------------|----------|--------------|
| CODECOURS | NOT NULL | VARCHAR2(10) |
| CODEENSEIGNANT | NOT NULL | VARCHAR2(10) |

```
SQL> desc etudiant
```

| Name | Null? | Type |
|----------------|----------|--------------|
| CODEETUDIANT | NOT NULL | VARCHAR2(10) |
| NOMETUDIANT | | VARCHAR2(20) |
| PRENOMETUDIANT | | VARCHAR2(20) |
| DATENAISSANCE | | DATE |
| VILLE | | VARCHAR2(10) |

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

```
SQL> SELECT VILLE FROM Etudiant;

VILLE
-----
Agadir
Tiznit
Casa
Taroudant
Safi
Casa
Paris
Lyon
Marseille
Toulouse
Nice

11 rows selected.
```

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

```
SQL> SELECT NOMETUDIANT, VILLE FROM etudiant ORDER BY NOMETUDIANT;

NOMETUDIANT      VILLE
-----
Adim             Tiznit
Ben              Agadir
Daoui            Taroudant
Dupont           Paris
Fournier         Nice
Garcia           Marseille
Idrissi          Casa
Lefevre          Toulouse
Martin           Lyon
kab              Casa
nadi             Safi

11 rows selected.
```

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

```
SQL> SELECT NOMETUDIANT FROM ETUDIANT WHERE VILLE IN ('Agadir','Tiznit');

NOMETUDIANT
-----
Ben
Adim
```

[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 FROM ETUDIANT WHERE VILLE IN ('Agadir','Tiznit','Taroudant')
2      ORDER BY VILLE, NOMETUDIANT;
```

| NOMETUDIANT | VILLE |
|-------------|-----------|
| Ben | Agadir |
| Daoui | Taroudant |
| Adim | Tiznit |

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

```
SQL> SELECT EXTRACT(YEAR FROM CURRENT_DATE)-EXTRACT(YEAR FROM dateNaissance) AS Age, nomEtudiant
2      FROM ETUDIANT
3      ORDER BY Ville, Age DESC;
```

| AGE | NOMETUDIANT |
|-----|-------------|
| 23 | Ben |
| 25 | Idrissi |
| 17 | kab |
| 26 | Martin |
| 24 | Garcia |
| 25 | Fournier |
| 25 | Dupont |
| 16 | nadi |
| 18 | Daoui |
| 20 | Adim |
| 27 | Lefevre |

11 rows selected.

[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 NOMETUDIANT, EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM dateNaissance) AS age
2      FROM etudiant
3      WHERE EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM dateNaissance) BETWEEN 17 AND 20
4      ORDER BY age DESC;
```

| NOMETUDIANT | AGE |
|-------------|-----|
| Adim | 20 |
| Daoui | 18 |
| kab | 17 |

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

```
SQL> SELECT nomEtudiant, ville FROM Etudiant WHERE Ville IN ('Casa', 'Taroudant', 'Safi');
```

| NOMETUDIANT | VILLE |
|-------------|-----------|
| Idrissi | Casa |
| Daoui | Taroudant |
| nadi | Safi |
| kab | Casa |

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

```
SQL> SELECT nomEtudiant, prenomEtudiant  
2 FROM ETUDIANT  
3 WHERE nomEtudiant LIKE '%Ben%' OR nomEtudiant LIKE '%oui%';
```

| NOMETUDIANT | PRENOMETUDIANT |
|-------------|----------------|
| Ben | Amine |
| Daoui | Asma |