TRAVAILLE A RENDRE

TP1 : BASE DE DONNEES **NOM ET PRENOM :IZIKKI Hajar**

PROF ENCADRANT : M. OUKDACH

Création des tables :

```
CREATE TABLE Etudiant (
    codeEtudiant VARCHAR(10) PRIMARY KEY,
    nomEtudiant VARCHAR(20),
    prenomEtudiant VARCHAR(20),
    dateNaissance DATE,
    ville VARCHAR(20));
CREATE TABLE Cours (
    codeCours VARCHAR(10) PRIMARY KEY,
    intitule VARCHAR(20),
    nbHeures DECIMAL(4, 2)
);
CREATE TABLE Enseignant (
    codeEnseignant VARCHAR(10) PRIMARY KEY,
    nomEnseignant VARCHAR(20),
    prenomEnseignant VARCHAR(20),
    specialite VARCHAR(20)
);
CREATE TABLE Resultat (
    codeEtudiant VARCHAR(10),
    codeCours VARCHAR(10),
    note DECIMAL(5, 2),
    FOREIGN KEY (codeEtudiant) REFERENCES Etudiant(codeEtudiant),
    FOREIGN KEY (codeCours) REFERENCES Cours(codeCours)
);
CREATE TABLE Charge (
    codeCours VARCHAR(10),
    codeEnseignant VARCHAR(10),
    FOREIGN KEY (codeCours) REFERENCES Cours(codeCours),
    FOREIGN KEY (codeEnseignant) REFERENCES Enseignant(codeEnseignant)
);
```

Pour créer un utilisateur :

```
CREATE USER ETu IDENTIFIED BY Hajar1234;
```

Accéder les privilèges à l'utilisateur :

```
SQL> GRANT ALL PRIVILEGES TO ETu;

Autorisation de privilÞges (GRANT) acceptÚe.

SQL>
```

Insertion de données :

```
SQL> INSERT INTO Etudiant (codeEtudiant, nomEtudiant, prenomEtudiant, dateNaissance, ville)
  2  SELECT 'E001', 'Dupont', 'Jean', TO_DATE('2000-01-15', 'YYYY-MM-DD'), 'Paris' FROM DUAL
  3  UNION ALL
  4  SELECT 'E002', 'Martin', 'Sophie', TO_DATE('1999-08-22', 'YYYY-MM-DD'), 'Lyon' FROM DUAL
  5  UNION ALL
  6  SELECT 'E003', 'Dubois', 'Pierre', TO_DATE('2001-03-05', 'YYYY-MM-DD'), 'Marseille' FROM DUAL
  7  UNION ALL
  8  SELECT 'E004', 'Leroy', 'Marie', TO_DATE('2002-06-10', 'YYYY-MM-DD'), 'Toulouse' FROM DUAL
  9  UNION ALL
 10  SELECT 'E005', 'Moreau', 'Luc', TO_DATE('1998-12-28', 'YYYY-MM-DD'), 'Bordeaux' FROM DUAL;

5 ligne(s) crÚÚe(s).
```

```
SQL> INSERT INTO Cours (codeCours, intitule, nbHeures)
  2  SELECT 'C001', 'Mathématiques', 60 FROM DUAL
  3  UNION ALL
  4  SELECT 'C002', 'Histoire', 45 FROM DUAL
  5  UNION ALL
  6  SELECT 'C003', 'Physique', 75 FROM DUAL
  7  UNION ALL
  8  SELECT 'C004', 'Anglais', 30 FROM DUAL
  9  UNION ALL
 10  SELECT 'C005', 'Chimie', 50 FROM DUAL;

5 ligne(s) crÚÚe(s).
```

```
SQL> INSERT INTO Enseignant (codeEnseignant, nomEnseignant, prenomEnseignant, specialite)
  2  SELECT 'ENS001', 'Garcia', 'Marie', 'Mathematiques' FROM DUAL UNION ALL
  3  SELECT 'ENS002', 'Lefevre', 'Antoine', 'Histoire' FROM DUAL UNION ALL
  4  SELECT 'ENS003', 'Durand', 'Sophie', 'Physique' FROM DUAL UNION ALL
  5  SELECT 'ENS004', 'Leclerc', 'Luc', 'Anglais' FROM DUAL UNION ALL
  6  SELECT 'ENS005', 'Petit', 'Julie', 'Chimie' FROM DUAL;

5 ligne(s) crÚÚe(s).
```

```
SQL> INSERT INTO Resultat (codeEtudiant, codeCours, note)
  2  SELECT 'E001', 'C001', 15.5 FROM DUAL
  3  UNION ALL
  4  SELECT 'E002', 'C002', 18 FROM DUAL
  5  UNION ALL
  6  SELECT 'E003', 'C003', 16.75 FROM DUAL
  7  UNION ALL
  8  SELECT 'E004', 'C004', 14 FROM DUAL
  9  UNION ALL
 10  SELECT 'E005', 'C005', 17.25 FROM DUAL;

5 ligne(s) crÚÚe(s).
```

```
SQL> INSERT INTO Charge (codeCours, codeEnseignant)
  2  SELECT 'C001', 'ENS001' FROM DUAL UNION ALL
  3  SELECT 'C002', 'ENS002' FROM DUAL UNION ALL
  4  SELECT 'C003', 'ENS003' FROM DUAL UNION ALL
  5  SELECT 'C004', 'ENS004' FROM DUAL UNION ALL
  6  SELECT 'C005', 'ENS005' FROM DUAL;

5 ligne(s) crÚÚe(s).
```

Augmenter de 5% toutes les notes des étudiant :

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

5 ligne(s) mise(s) Ó jour.
```

Changer le nom du cours 2 par ‘SGBD’ :

```
SQL> UPDATE Cours
  2  SET intitule = 'SGBD'
  3  WHERE codeCours = 'C002';

1 ligne mise Ó jour.
```

Supprimer le cours n° 5 :

```
SQL> DELETE FROM Charge
  2  WHERE codeCours = 'C005';

1 ligne supprimÚe.
```

Affichage des tables déjà créées :

```
SQL> DESCRIBE Etudiant;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ----------------------------
 CODEETUDIANT                              NOT NULL VARCHAR2(10)
 NOMETUDIANT                                        VARCHAR2(20)
 PRENOMETUDIANT                                     VARCHAR2(20)
 DATENAISSANCE                                      DATE
 VILLE                                              VARCHAR2(20)

SQL> DESCRIBE Cours;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ----------------------------
 CODECOURS                                 NOT NULL VARCHAR2(10)
 INTITULE                                           VARCHAR2(20)
 NBHEURES                                           NUMBER(4,2)

SQL> DESCRIBE Resultat;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ----------------------------
 CODEETUDIANT                                       VARCHAR2(10)
 CODECOURS                                          VARCHAR2(10)
 NOTE                                               NUMBER(5,2)

SQL> DESCRIBE Enseignant;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ----------------------------
 CODEENSEIGNANT                            NOT NULL VARCHAR2(10)
 NOMENSEIGNANT                                      VARCHAR2(20)
 PRENOMENSEIGNANT                                   VARCHAR2(20)
 SPECIALITE                                         VARCHAR2(20)

SQL> DESCRIBE Charge;
 Nom                                       NULL ?   Type
 ----------------------------------------- -------- ----------------------------
 CODECOURS                                          VARCHAR2(10)
 CODEENSEIGNANT                                     VARCHAR2(10)

SQL>
```

Affichage de nom et la ville des étudiants :

```
SQL> SELECT nomEtudiant, ville
  2  FROM Etudiant;

NOMETUDIANT          VILLE
-------------------- --------------------
Dupont               Paris
Martin               Lyon
Dubois               Marseille
Leroy                Toulouse
Moreau               Bordeaux

SQL>
```

Affichage de nom et de la ville des étudiants ordonnés par nom :

```
SQL> SELECT nomEtudiant, ville
  2  FROM Etudiant
  3  ORDER BY nomEtudiant;

NOMETUDIANT          VILLE
-------------------- --------------------
Dubois               Marseille
Dupont               Paris
Leroy                Toulouse
Martin               Lyon
Moreau               Bordeaux

SQL>
```

Affichage de nom des étudiants des villes agadir et Tiznit :

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

aucune ligne sÚlectionnÚe

SQL>
```

Affichage de 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;

aucune ligne sÚlectionnÚe

SQL>
```

Affichage de nom et l'âge des étudiants ordonnés par ville et par âge descendant :

```
SQL> SELECT nomEtudiant, ville, FLOOR(MONTHS_BETWEEN(SYSDATE, dateNaissance) / 12) AS age
  2  FROM Etudiant
  3  ORDER BY ville, age DESC;

NOMETUDIANT          VILLE                    AGE
-------------------- -------------------- ----------
Moreau               Bordeaux                 24
Martin               Lyon                     24
Dubois               Marseille                22
Dupont               Paris                    23
Leroy                Toulouse                 21

SQL>
```

Affichage de nom et l'âge des étudiants dont l'âge est compris entre 17 et 20, et ordonnés par âge descendant :

```
SQL> SELECT nomEtudiant, ville, FLOOR(MONTHS_BETWEEN(SYSDATE, dateNaissance) / 12) AS age
  2  FROM Etudiant
  3  WHERE FLOOR(MONTHS_BETWEEN(SYSDATE, dateNaissance) / 12) BETWEEN 17 AND 20
  4  ORDER BY age DESC;

aucune ligne sÚlectionnÚe

SQL>
```

Affichage de 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');

aucune ligne sÚlectionnÚe

SQL>
```

Lister les étudiants dont le nom contient ‘Ben’ ou ‘oui’ :

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

aucune ligne sÚlectionnÚe
```