TD 1

Dans cet exemple, nous allons créer une base de données simple pour suivre les informations sur les étudiants et les cours auxquels ils sont inscrits. La base de données aura deux tables principales : etudiants et cours, avec une table de jonction appelée inscription pour représenter les inscriptions des étudiants aux cours.

Création de la base de données :

```
CREATE DATABASE IF NOT EXISTS universite;
USE universite;
```

Création de la table etudiants :

```
CREATE TABLE IF NOT EXISTS etudiants (
                    id_etudiant INT PRIMARY KEY,
                     nom VARCHAR(50),
                     prenom VARCHAR(50),
                    date_naissance DATE );
```

Création de la table cours :

```
CREATE TABLE IF NOT EXISTS cours (
                    id_cours INT PRIMARY KEY,
                     nom_cours VARCHAR(50),
                     credit INT );
```

Création de la table inscription pour les inscriptions :

```
CREATE TABLE IF NOT EXISTS inscription (
                    id_inscription INT PRIMARY KEY,
                    id_etudiant INT,
                    id_cours INT,
                    FOREIGN KEY (id_etudiant) REFERENCES etudiants(id_etudiant),
                    FOREIGN KEY (id_cours) REFERENCES cours(id_cours) );
```

Insertion de données dans les tables :

```
INSERT INTO etudiants VALUES
 (1, 'AHMED', 'BERA', '1990-01-01');
INSERT INTO etudiants VALUES
(2, 'GHALI', 'OUTMAN', '1992-05-15');
```

```
INSERT INTO cours VALUES
 (101, 'Mathématiques', 3);
 INSERT INTO cours VALUES
 (102, 'Informatique', 4);
```

```
INSERT INTO inscription VALUES
(1, 1, 101);
INSERT INTO inscription VALUES
(2, 1, 102);
 INSERT INTO inscription VALUES
(3, 2, 101);
```

Jointure interne pour obtenir les cours auxquels chaque étudiant est inscrit :

```
SELECT etudiants.nom, etudiants.prenom, cours.nom_cours
 FROM etudiants
 JOIN inscription ON etudiants.id_etudiant = inscription.id_etudiant
 JOIN cours ON inscription.id_cours = cours.id_cours;
```

Jointure externe pour obtenir tous les étudiants, même s'ils ne sont pas inscrits à un cours :

```
SELECT etudiants.nom, etudiants.prenom, cours.nom_cours
FROM etudiants
LEFT JOIN inscription ON etudiants.id_etudiant = inscription.id_etudiant
LEFT JOIN cours ON inscription.id_cours = cours.id_cours;
```

Jointure externe droite (RIGHT JOIN) pour obtenir tous les cours, même s'ils n'ont pas d'étudiants inscrits :

```
SELECT cours.nom_cours, etudiants.nom, etudiants.prenom
FROM cours
 RIGHT JOIN inscription ON cours.id_cours = inscription.id_cours
RIGHT JOIN etudiants ON inscription.id_etudiant = etudiants.id_etudiant;
```

Jointure croisée (CROSS JOIN) pour obtenir toutes les combinaisons possibles d'étudiants et de cours (produit cartésien) :

```
SELECT etudiants.nom, etudiants.prenom, cours.nom_cours
FROM etudiants
CROSS JOIN cours;
```

Jointure avec condition supplémentaire (JOIN avec WHERE) pour obtenir les étudiants inscrits à un cours spécifique :

```
SELECT etudiants.nom, etudiants.prenom, cours.nom_cours
 FROM etudiants
 JOIN inscription ON etudiants.id_etudiant = inscription.id_etudiant
 JOIN cours ON inscription.id_cours = cours.id_cours
 WHERE cours.nom_cours = 'Mathématiques';
```

Jointure auto (SELF JOIN) pour obtenir les étudiants qui partagent la même date de naissance :

```
SELECT e1.nom, e1.prenom, e2.nom AS nom_copain,
        e2.prenom AS prenom_copain,
        e1.date_naissance
FROM etudiants e1
 JOIN etudiants e2 ON e1.date_naissance = e2.date_naissance
                    AND e1.id_etudiant != e2.id_etudiant;
```

Jointure avec agrégation (utilisation de GROUP BY et fonctions d'agrégation) pour obtenir le nombre d'étudiants inscrits à chaque cours :

```
SELECT cours.nom_cours, COUNT(inscription.id_etudiant) AS nombre_etudiants_inscrits
FROM cours
LEFT JOIN inscription ON cours.id_cours = inscription.id_cours
GROUP BY cours.nom_cours;
```

Jointure avec condition complexe pour obtenir les étudiants inscrits à plus d'un cours :

```
SELECT etudiants.nom, etudiants.prenom,
COUNT(DISTINCT inscription.id_cours) AS nombre_cours_inscrits
FROM etudiants
JOIN inscription ON etudiants.id_etudiant = inscription.id_etudiant
GROUP BY etudiants.nom, etudiants.prenom
HAVING nombre_cours_inscrits > 1;
```