BDD (fiche 2) – Langage SQL - CORRIGE

**Exercice 1**

**CREATE TABLE** annuaire(nom VARCHAR(100) NOT NULL, prenom VARCHAR(100) NOT NULL, **numero** VARCHAR(20) **PRIMARY KEY);**

**Exercice 2**

**Veiller à créer la table classe en premier**

**CREATE TABLE** classe(nomclasse VARCHAR(20) **PRIMARY KEY);**

**CREATE TABLE** eleve(nom VARCHAR(100), prenom VARCHAR(100), nomclasse VARCHAR(20) **REFERENCES** **classe**(nomclasse),**ideleve** VARCHAR(20) **PRIMARY KEY);**

**Exercice 3**

**Veiller à créer la table note en dernier**

**CREATE TABLE** eleve(nom VARCHAR(100), prenom VARCHAR(100), **ideleve** VARCHAR(20) **PRIMARY KEY);**

**CREATE TABLE** matiere(intitule VARCHAR(100), **idmat** INT **PRIMARY KEY);**

**CREATE TABLE** note(note DECIMAL(4,2), **idmat** INT **REFERENCES** **matiere**(idmat), **ideleve** VARCHAR(20) **REFERENCES** **eleve**(**ideleve**), **PRIMARY\_KEY(ideleve,idmat),   
CHECK**(note>=0 and note<=20) **);**

**Impossible de créer supprimer d’abord eleve ou matiere car elles sont référencées dans note. Donc on supprime d’abord les deux autres.**

DROP TABLE note  
DROP TABLE matiere  
DROP TABLE eleve

**Exercice 4**

**Ne pas oublier les check**

**CREATE TABLE** categorie(libelle VARCHAR(20), **idcategorie** INT **PRIMARY KEY);**

**CREATE TABLE** magasin(nom VARCHAR(20), adresse VARCHAR(200), **idmagasin** INT **PRIMARY KEY);**

**CREATE TABLE** article(nom VARCHAR(20), poids INT, **idcategorie** INT **REFERENCES** **categorie**(idcategorie), **idarticle** INT **PRIMARY KEY, CHECK**(poids >=0));

**CREATE TABLE** stock(quantite INT, **idarticle** INT **REFERENCES** **article**(idarticle), **idmagasin** INT **REFERENCES** **magasin**(**idmagasin**), **PRIMARY KEY(idarticle,idmagasin),   
CHECK**(quantite>=0) **);**

**Exercice 5**

1) La table client n'existe pas donc on ne peut pas la supprimer

2) A la seconde requête, on se réfère à la table produit qui n'a pas encore été créée

3) Dans la table commande il y a une clef etrangere qui pointe sur l’attribut nom de la table produit. Or ce n’est pas la clef primaire.

4) Tout d'abord, le mot clef VALUES est manquant  
 D'autre part la quatrième requête :  
 INSERT INTO commande(0,0,'2020-03-02')  
 crée une commande en se référant au client 0 et au produit 0.  
 OR ces deux éléments n'ont pas encore été créés.

**Exercice 6**

On considère les deux tables suivantes

CREATE TABLE joueur (jid INT PRIMARY KEY, nom VARCHAR(100) NOT NULL);

CREATE TABLE partie (j1 INT REFERENCES joueur(jid), j2 INT REFERENCES joueur(jid), score1 INT NOT NULL,   
 score2 INT NOT NULL, CHECK( (j1<>j2) );

Cette BDD comporte une table "joueur" avec une clef primaire jid (entier) et un attribut nom de type string qui ne doit pas être NULL. Elle comporte une seconde table partie comportant 4 attributs j1, j2 1 score 1 et score2. Enfin dans la cette table, les joueurs j1 et j2 saisis dans chaque enregistrement doivent être évidemment distincts.

Les contraintes d'entité sont des règles qui doivent être respectées par les BDD. Voir chapitre 1.

INSERT INTO joueur VALUES (1,"toto")  
INSERT INTO joueur VALUES (1,"titi")  
**Violation contrainte d'entité**

INSERT INTO joueur VALUES(1,NULL)  
**Violation contrainte de domaine**

INSERT INTO partie VALUES(10,11,5,7)  
On suppose qu'il n'existe aucun joueur 11  
**Violation contrainte de référence**

INSERT INTO partie VALUES(1,1,5,7)  
Erreur car j1<>j2  
**Violation contrainte utilisateur**

**Exercice 7**

CREATE TABLE joueur (jid INT PRIMARY KEY, nom VARCHAR(100) NOT NULL);

CREATE TABLE partie (**date DATE NOT NULL**, j1 INT REFERENCES joueur(jid), j2 INT REFERENCES joueur(jid), score1 INT NOT NULL,   
 score2 INT NOT NULL, **CHECK(score1>=0 and score2>=0)** , **UNIQUE(j1,j2,date)** );