Exercice 1 :

On considère la base de données dont le schéma est le suivant :

ARTISTE(NumArt, Nom, Genre, Nationalite)
ALBUM(NumAlb, Titre, Annee)
CHANSON(NumChan, Titre, NumAlb, Duree)
PARTICIPE(NumAlb, NumArt)

L'année d'un album doit être comprise en 1900 et 2100; de plus sa valeur par défaut est 2019. La table PARTICIPE permet de stocker l'information relative à la participation d'un artiste à un album.

1. Identifiez les clés primaires et les clés étrangères de toutes les tables.

   ARTISTE(NumArt, Nom, Genre, Nationalite)

   ALBUM(NumAlb, Titre, Annee)

   CHANSON(NumChan, Titre, # NumAlb, Duree)

   PARTICIPE(#NumAlb,#NumArt)

2. Donnez le code de définition de l'attribut ANNEE de la table ALBUM avec les contraintes énoncées

```
Annee integer DEFAULT 2019 CHECK (Annee BETWEEN 1900 AND 2100)
```

3. Donnez le code de création de la table CHANSON

```
CREATE TABLE Chanson (
    NumChan serial PRIMARY KEY,
    Titre varchar(50),
    NumAlb integer,
    Duree numeric(4,2),
    FOREIGN KEY (NumAlb) REFERENCES Album(NumAlb)
);
```

Exercice 2 :

Soit la base suivante décrivant un supermarché :

RAYON(NOMR, ETAGE)
ARTICLE(REFERENCE, TYPE, DESCRIPTION, COULEUR)
DISPONIBILITE(#NOMR, #REFERENCE, QUANTITE)
EMPLOYÉ(NUMERO, NOME, SALAIRE, #RAYON, #RESPONSABLE)

Les rayons, identifiés par la clé NOMR, sont situés à un étage donné. Les articles sont identifiés par l'entier REFERENCE et décrits par un type, une description et une couleur. Ils sont disponibles dans une certaine quantité à chaque rayon. Un employé travaille à un rayon et a pour responsable un autre employé. L'attribut responsable représente donc un numéro d'employé.

Formulez en SQL les requêtes suivantes :

1. Les descriptions des articles de couleur ROUGE.

```
SELECT A.Description
FROM Article A
WHERE A.Couleur = 'ROUGE';
```

2. Les descriptions des articles, l'étage des rayons où ils sont disponibles, ainsi que la quantité.

```
SELECT A.Description, R.Etage, D.Quantite
FROM Article A
JOIN Disponibilite D ON D.Reference = A.Reference
JOIN Rayon R ON R.Nomr = D.Nomr;
```

3. Les descriptions des articles et le nom du rayon où ils sont disponibles, y compris les articles que ne sont plus disponibles.

```
SELECT A.Description, D.Nomr
FROM Article A
LEFT JOIN Description D ON D.Reference = A.Reference;
```

4. Les articles qui ne sont pas disponibles.

```
SELECT A.description
FROM Article A
WHERE A.reference NOT IN (SELECT D.reference
                          FROM Disponibilite D
                         );
```

5. Les descriptions des articles présents dans tous les rayons.

```
SELECT A.Description
FROM Article A
WHERE NOT EXISTS( SELECT 1
                  FROM Rayon R
                  WHERE NOT EXISTS( SELECT 1
                                    FROM Disponibilite D
                                    WHERE D.nomr = R.nomR
                                      AND D.reference = A.reference
                                  )
                );
```

6. Les noms des rayons et la quantité totale d'articles qu'ils contiennent

```
SELECT Nomr, SUM(Quantite)
FROM Disponibilite
GROUP BY Nomr;
```

7. Les noms des responsables de plus de 10 employés et le nombre d'employés dont ils sont responsables.

```
SELECT ER.Nome, count(*)
FROM Employe ER
JOIN Employe E ON E.Numero = ER.Responsable
GROUP BY ER.Numero, ER.Nome
HAVING count(*)>=10;
```

Question bonus : proposez des insertions de tuples (dans le bon ordre) de sorte que la requête 5 retourne au moins un tuple.