

# Création de tables sous SQLite3 - Contrainte de référence

---

## I. Création de tables

1. Sous SQLite3 créer une base de données nommée test\_tables.db.
2. Paramétrer le SGBD afin d'utiliser les contraintes d'intégrité référentielles.

```
PRAGMA foreign_keys = ON;
```

3. Créer la table groupes\_fournisseurs

```
CREATE TABLE groupes_fournisseurs (  
    id_groupe INTEGER PRIMARY KEY, #premier argument, ayant comme fonction d'être  
    la clé primaire  
    nom_groupe TEXT NOT NULL "deuxième argument de type text"  
);
```

4. Vérifier la création de la table
5. Quelle est sa clé primaire ?
  - La clé primaire de la table groupes\_fournisseurs est la colonne id\_groupe.
6. Créer la table fournisseurs

```
CREATE TABLE fournisseurs (  
    id_fournisseur INTEGER PRIMARY KEY, #Clé primaire de type INTEGER  
    nom_fournisseur TEXT NOT NULL, #argument de type TEXT non nulle.  
    id_groupe INTEGER, #Clé étrangère faisant référence à id_groupe dans la table  
    groupes_fournisseurs  
    FOREIGN KEY (id_groupe) REFERENCES groupes_fournisseurs (id_groupe)  
);
```

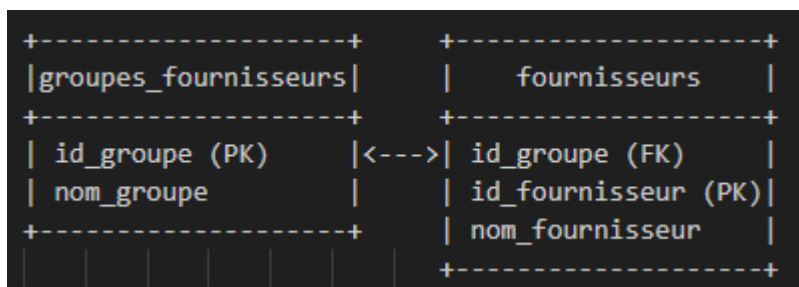
6. Quelle est sa clé étrangère ? À quelle clé primaire de quelle table fait-elle référence ?
  - La clé étrangère dans la table fournisseurs est id\_groupe. Elle fait référence à la clé primaire id\_groupe de la table groupes\_fournisseurs.
7. Schéma des deux tables
  - groupe\_fournisseurs :

id_groupe	nom_groupe
INTEGER (PK)	TEXT (NOT NULL)

- fournisseurs :

id_fournisseur	nom_fournisseur	id_groupe
INTEGER (PK)	TEXT (NOT NULL)	INTEGER (FK)

8. Représenter sous forme de diagramme les schémas des deux tables et leur relation



9. Insérer trois lignes dans la table groupes\_fournisseurs

```

INSERT INTO groupes_fournisseurs (id_groupe, nom_groupe)
VALUES
(1, 'mécanique'), #première ligne de la table groupe_fournisseurs
(2, 'électronique'), #deuxième ligne de la table groupe_fournisseurs
(3, 'optique'); #troisième ligne de la table groupe_fournisseurs
  
```

10. Lister le contenu de la table groupes\_fournisseurs

```

SELECT * FROM groupes_fournisseurs;
#suite à l'utilisation de cette instruction, toutes les lignes de la table
groupes_fournisseurs se sont affichées
  
```

11. Insérer une ligne dans la table fournisseurs

```

INSERT INTO fournisseurs (id_fournisseur, nom_fournisseur, id_groupe)
VALUES
(1, 'lextronic', 2), #première ligne de la table fournisseurs
(2, 'gotronic', 2); #deuxième ligne de la table fournisseurs
  
```

12. Vérifier son contenu

```
SELECT * FROM fournisseurs;  
SELECT * FROM groupes_fournisseurs;
```

13. Que retourne l'instruction suivante ? Expliquer

- Cette instruction retourne une erreur car il n'existe pas de groupe avec `id_groupe = 4` dans la table `groupes_fournisseurs`.

14. Ecrire et exécuter les instructions permettant d'insérer le fournisseur carrefour dans le groupe de fournisseurs 4 nommé `grandes_surfaces`. Justifier la démarche

```
INSERT INTO groupes_fournisseurs (id_groupe, nom_groupe)  
VALUES (4, 'grandes_surfaces');  
  
INSERT INTO fournisseurs (id_fournisseur, nom_fournisseur, id_groupe)  
VALUES (3, 'carrefour', 4);  
  
# D'abord, il faut créer le groupe "grandes_surfaces" avec id_groupe = 4, puis  
insérer "Carrefour" dans ce groupe.
```

15. Vérifier les contenus des tables.

16. Effacer une ligne dans la table `fournisseurs`

```
DELETE FROM fournisseurs #supprimer de la table fournisseurs  
WHERE id_fournisseur = 3; #celui qui possède l'identifiant "3"
```

17. Vérifier le contenu des tables

```
SELECT * FROM fournisseurs;  
SELECT * FROM groupes_fournisseurs;
```

18. Supprimer un groupe de la table `groupes_fournisseurs`

```
DELETE FROM groupes_fournisseurs  
WHERE id_groupe = 2;  
  
# Cette instruction retourne une erreur si la contrainte de clé étrangère est  
activée, car il existe des fournisseurs associés au groupe avec id_groupe = 2.
```

19. Supprimer un groupe de fournisseurs

Pour supprimer un groupe de fournisseurs, il faut d'abord supprimer les fournisseurs associés à cause de la relation entre les deux tables. :

```
DELETE FROM fournisseurs
WHERE id_groupe = 2;
DELETE FROM groupes_fournisseurs
WHERE id_groupe = 2;
```

## 20. Supprimer la table fournisseurs

```
DROP TABLE fournisseurs; #Supprime définitivement la table fournisseurs ainsi que
toutes ses données
```

## 21. Vérifier que c'est bien le cas.

```
SELECT name FROM sqlite_master WHERE type='table';
```

## 22. Cette table ne contenait plus de données. Peut-on supprimer une table contenant des données ?

- Oui, on peut supprimer une table contenant des données avec la commande DROP TABLE. Cependant, si la table contient des contraintes de clé étrangère qui sont violées par la suppression, SQLite peut empêcher cette action selon la configuration de la contrainte.

# Manipulation des tables en SQL

---

## I. La base de données

### 1. Quelles sont les tables contenues dans la base.

```
#manque de temps
```