

Rapport sur la fusion de base

Étape 1

Pour fusionner deux bases de données il faut d'abord télécharger un minimum de deux bases.

Une fois les bases acquises, créer un fichier SQL sur VScode.

Étape 2

Pour commencer il faut supprimer et recréer les clefs étrangères en les mettant en :

```
ADD CONSTRAINT  
ON UPDATE "CASCADE"  
ON DELETE "CASCADE".
```

Ont commencé par supprimer les ancienne clef étrangères en mettant le nom de la table touché puis la clef à supprimer :

```
ALTER TABLE Film DROP FOREIGN KEY film_ibfk_1;
```

Ici on ajoute la table contenant la clef à modifier et le nom de la clef :

```
ALTER TABLE Film  
  
ADD CONSTRAINT films_ibfk_1  
FOREIGN KEY (...) REFERENCES ...(...)  
ON DELETE CASCADE  
ON UPDATE CASCADE,
```

Puis, le nom de la clef étrangère à modifier :

```
ALTER TABLE Film

ADD CONSTRAINT ...

FOREIGN KEY (idPersonne) REFERENCES ...(...)

ON DELETE CASCADE

ON UPDATE CASCADE,
```

Ensuite, le nom de la table que la clef étrangère référencer:

```
ALTER TABLE Film

ADD CONSTRAINT ...

FOREIGN KEY (idPersonne) REFERENCES Personne(...)

ON DELETE CASCADE

ON UPDATE CASCADE,
```

Enfin, on modifie le nom de la colonne que la clef vise:

```
ALTER TABLE Film

ADD CONSTRAINT ...

FOREIGN KEY (idPersonne) REFERENCES Personne(Personne)

ON DELETE CASCADE

ON UPDATE CASCADE,
```

Répéter cette action pour toutes les clef étrangère à modifier dans les autres tables.

Étape 3

Changer les id de la deuxième bases avec la requête SQL suivante:

```
UPDATE ...  
SET ...= ...+ 100;
```

Remplacer les point par la table a modifier comme dans l'exemple suivant:

```
UPDATE Film  
SET ...= ...+ 100;
```

Puis, la colonne id à modifier (ici: idFilm):

```
UPDATE Film  
SET idFilm = idFilm + 100;
```

Changer aussi les autres id des colonnes que vous souhaitez insérer.

Étape 3: Insérer le contenu des tables dans la table finale grâce à la requête SQL suivante:

```
INSERT INTO ...  
SELECT * FROM ... WHERE ... > ...;
```

Remplacer les point par la table a modifier comme dans l'exemple suivant:

```
INSERT INTO Film  
SELECT * FROM ... WHERE ... > ...;
```

Puis, le “nom de la base.le nom de la table”:

```
INSERT INTO Film
SELECT * FROM cinemaNathan.Film WHERE ... > ...;
```

Après le WHERE, le nom de la colonne à insérer dans la base nommée juste avant (ici: idFfilm):

```
INSERT INTO Film
SELECT * FROM cinemaNathan.Film WHERE idFfilm > ...;
```

Pour finir, insérer l'index juste avant le numéro de l'index de la première colonne à insérer comme dans l'exemple (ici nous souhaitons insérer toutes les colonnes dont l'index est > 164):

```
INSERT INTO Film
SELECT * FROM cinemaNathan.Film WHERE idFfilm > 164;
```

Répéter cette action pour toutes les colonnes à insérer dans la base principale.

Étape 4

Nous finissons par supprimer tout les doublons avec la requête suivante.

En premier, nous mettons le nom de la table à modifier:

```
DELETE FROM Personne
WHERE ... IN (
    SELECT ... FROM (
        SELECT
            ...,
            ROW_NUMBER() OVER (PARTITION BY ... ORDER BY ...) AS rn
        FROM ...
    ) AS doublons
    WHERE rn > 1
);
```

Puis, nous récupérons l'id ciblé:

```
DELETE FROM Personne
WHERE idPersonne IN (
    SELECT idPersonne FROM (
        SELECT
            idPersonne,
            ROW_NUMBER() OVER (PARTITION BY ... ORDER BY ...) AS rn
        FROM ...
    ) AS doublons
    WHERE rn > 1
);
```

Ensuite, nous mettons les colonnes qu'on souhaite vérifier (ici nous voulons vérifier si les colonnes prénom ou/et nom contiennent des doublons):

```
DELETE FROM Personne
WHERE idPersonne IN (
    SELECT idPersonne FROM (
        SELECT
            idPersonne,
            ROW_NUMBER() OVER (PARTITION BY nom, prenom ORDER BY ...) AS
rn
        FROM ...
    ) AS doublons
    WHERE rn > 1
);
```

Ensuite, l'ordre selon l'id:

```
DELETE FROM Personne
WHERE idPersonne IN (
    SELECT idPersonne FROM (
        SELECT
            idPersonne,
            ROW_NUMBER() OVER (PARTITION BY nom, prenom ORDER BY
idPersonne) AS rn
        FROM ...
    ) AS doublons
    WHERE rn > 1
);
```

Pour finir, nous remettons la table ciblée comme choisi précédemment (en vert):

```
DELETE FROM Personne
WHERE idPersonne IN (
    SELECT idPersonne FROM (
        SELECT
            idPersonne,
            ROW_NUMBER() OVER (PARTITION BY nom, prenom ORDER BY
idPersonne) AS rn
        FROM Personne
    ) AS doublons
    WHERE rn > 1
);
```

Une fois ces étapes terminées, nous avons réussi à fusionner plusieurs bases de données.