

Exercice 1 :

Les fichiers finaux (creation_final.sql et insert_final.sql représente le script avec les modifications apporté suite aux questions de cet exercice

Q1.1). Changement dans le fichier creation.sql

```
drop table IF EXISTS Articles cascade;  
drop table IF EXISTS fournisseurs cascade;  
drop table IF EXISTS catalogue cascade ;  
drop type IF EXISTS colors cascade;
```

Les clés primaires n'ont pas de auto-increment, cela pose des problème lors de l'insertion de plusieurs articles par exemple.

Dans la table **Catalogue** le couple (fid,aid) n'est pas unique.

Dans la table Fournisseurs aucune clé primaire n'est définie.

Q1.2). Dans creation.sql, ajouté:

```
drop table IF EXISTS Articles cascade;  
drop table IF EXISTS fournisseurs cascade;  
drop table IF EXISTS catalogue cascade;  
drop type IF EXISTS colors cascade;
```

```
CREATE TYPE colors as  
ENUM('rouge','noir','argente','opaque','cyan','magenta','vert','superjaune', 'rose');
```

```
CREATE TABLE Articles(  
aid serial primary key, /* De type serial pour l'auto-increment */  
    anom varchar(30) NOT NULL, /* Un articles a forcement un nom */,  
    acoul Color NOT NULL /* Un articles a forcement une couleur */  
);
```

```
CREATE TABLE Catalogue(  
    fid bigint,  
    aid bigint,  
    prix numeric(8,2) NOT NULL, /* Un catalogue a forcement un prix*/  
    primary key(fid, aid) /* le couple fournisseurs - article doit être unique, un  
fournisseur ne peut proposer qu'une fois le même article*/  
);
```

```
CREATE TABLE Fournisseurs(  
    fid serial primary key /* De type serial pour l'auto-increment */,  
    fnom varchar(30) NOT NULL /* Un fournisseur a forcément un nom */,  
    fad varchar(70) NOT NULL /* Un fournisseur a forcément une adresse */
```

);

Q1.3). Q1.4).

```
ALTER TABLE Catalogue
  ADD CONSTRAINT fk_Catalogue_to_Articles FOREIGN KEY (aid)
  REFERENCES Articles(aid)
  ON DELETE CASCADE
  ON UPDATE CASCADE;
```

Si on enlève un article alors il ne plus être présent dans le catalogue.

```
ALTER TABLE Catalogue
  ADD CONSTRAINT fk_Catalogue_to_Fournisseurs FOREIGN KEY (fid)
  REFERENCES Fournisseurs(fid)
  ON DELETE CASCADE
  ON UPDATE CASCADE;
```

Si on enlève un fournisseur alors tous ses articles présent dans le catalogue sont supprimés.

Q1.5). Il faut insérer les valeurs dans un ordre précis, Article puis Fournisseurs et Catalogue ou Fournisseurs puis Article et Catalogue, sinon nous obtenons des problèmes à cause de la table Catalogue qui vérifie l'existence de ces clés étrangères.

Q1.6). *INSERT INTO articles (aid, anom, acoul) VALUES (NULL, 'Rubber duck', 'superjaune');*

Q1.7). *INSERT INTO catalogue (fid, aid, prix) VALUES ('1', '17', '1000000.00');*

Q1.8).

Grâce au type **SERIAL**, la requête suivant fait planter l'insertion, même identifiant:
INSERT INTO articles (aid, anom, acoul) VALUES (1, 'Fail', 'rouge');

Q1.10).

Avant :

```
INSERT INTO catalogue (fid, aid, prix) VALUES ('1', '19', '-10');
```

L'insertion c'est déroulée sans aucune erreur

Après :

```
ALTER TABLE Catalogue
  ADD CONSTRAINT prix_catalogue_positive
  CHECK (prix>0);
```

```
INSERT INTO `catalogue` (`fid`, `aid`, `prix`) VALUES ('1', '19', '-10');
```

Plus possible !

Q1.11). Quand je supprime un article, tous ceux présents dans le catalogue sont supprimés.

Quand je supprime un fournisseur, tous ses articles présents dans le catalogue sont supprimés.

Exemple de requête:

```
DELETE FROM articles WHERE articles.aid = 1;
```

```
DELETE FROM fournisseurs WHERE fournisseurs.fid = 1;
```

Exercice 2 :

Exécuter le fichier request.sql