

Projet SQL

Première partie Description en langage naturel + Modèle UML

Description

Pour ce projet nous avons choisis de gérer les commandes d'une boutique d'alimentation passée par des clients. Cette boutique possède un site internet sur lequel les clients peuvent commander des articles auxquels chaque article est défini par un son type ainsi que son fournisseur.

tout cela est décrit ci après :

les Clients:

Le client est identifié par un numéro client unique. On représente aussi leur nom, leur prénom, adresse, email ainsi que leur nombre de commandes effectué. Un client a passé 0 ou plusieurs commandes dans la boutique pour s'inscrire.

les Commandes:

Les commandes sont identifiées par un numéro de commande unique lié à un client, une date d'achat ainsi que la quantité de produits achetés par commande. Les commandes peuvent être liées à 1 ou plusieurs articles commandés par le client. Un article est commandé, s'il est disponible avant la commande.

les Articles:

Chaque article est identifié par un code unique et est caractérisé par un son type ainsi que son statut correspondant à "disponible" ou "commandé".

Plusieurs articles peuvent être attribués à un type d'article. Un article peut être lié à un numéro de commande s'il est commandé.

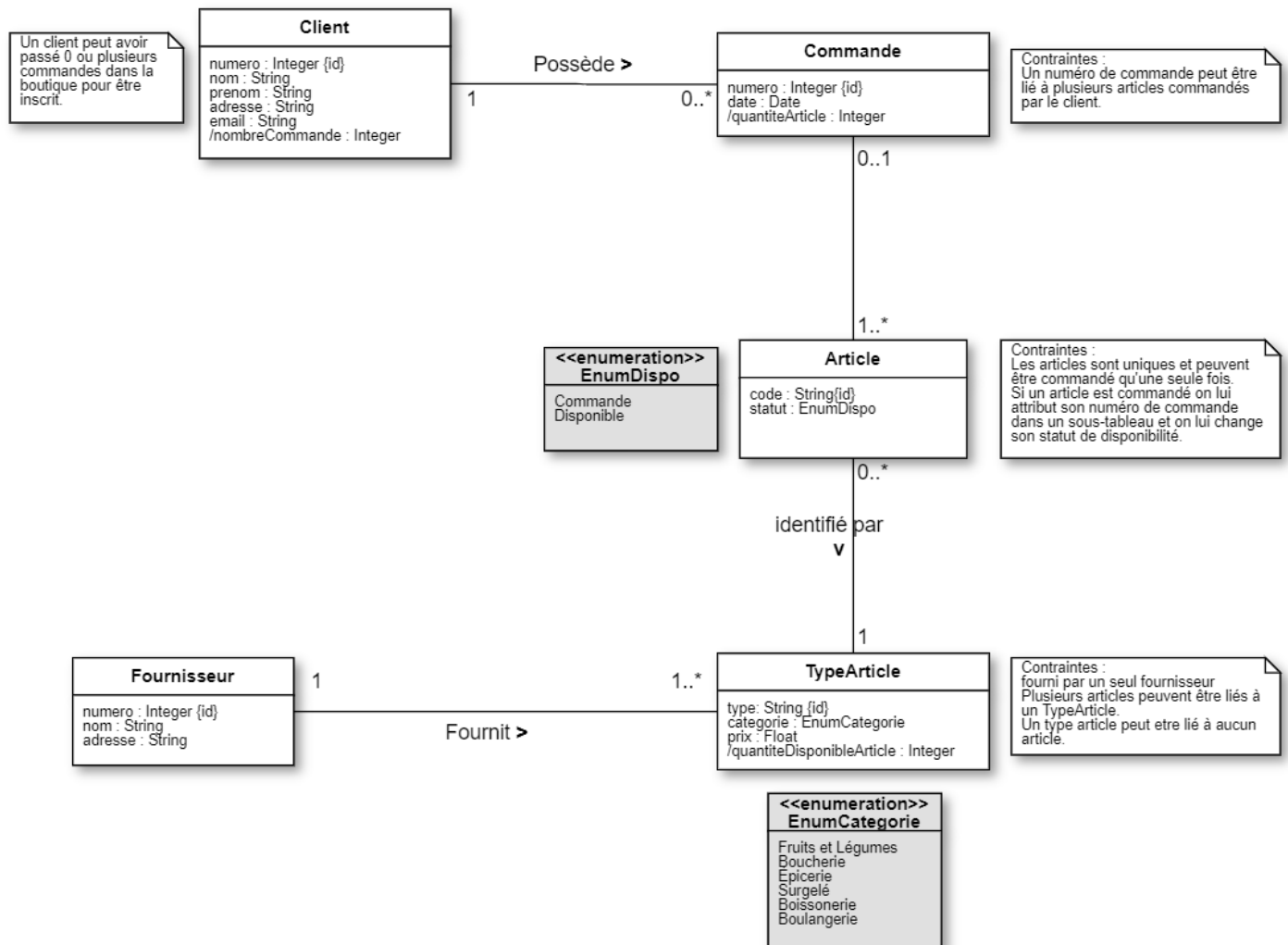
les TypesArticles:

Chaque type d'article est lié à 0 ou plusieurs articles auquel on leur attribue un prix ainsi que la quantité de produit disponible non commandé . Un type article est identifié par un numéro de fournisseur ainsi qu'une catégorie.

les Fournisseurs:

Les fournisseurs sont identifiés par un numéro unique de fournisseur, ainsi que du nom/origine et adresse. Un fournisseur peut fournir 1 ou plusieurs types de produits différents.

l'UML traduisant notre description du problème



Deuxième partie Traduction du modèle UML au relationnel + implémentation en SQL

Modèle relationnel

Règles de traduction de noms

Nom classe en singulier → nom table en pluriel

CamelCase → snake_case

att → att_nomclasse

Schéma des données et spécifications

Clients_base (numero_client, nom_client, prenom_client, adresse_client, email_client)

/* <nc, n, p, a, e, nbc> ∈ Clients ⇔ le client est identifié par son numéro nc dont le nom est n, le prénom p est son adresse mail e et postal a, (ainsi que le nombre nbc de commandes effectuées (vu dynamique)). */

Commandes_base (numero_commande, numero_client, date_commande)

/* <nco, nc, d, q> ∈ Commandes ⇔ la commande est définie par un numéro de commande nco et un numéro, effectué à la date d, (ainsi que la quantité de produit q achetée (vu dynamique)). */

ArticlesCommandes (code_article, numero_commande)

/* <ca, nco> ∈ ArticlesCommandes ⇔ l'Article Commandé est définie par un code article ca avec un numéro de commande nco */

Articles (code_article, type_article, statut_article)

/* <ca, ct, d> ∈ Articles ⇔ Un Article est identifié par un code d'article ca, son type ct et son statut de disponibilité d. Si le produit est commandé, on lui attribue le statut commandé. */

TypesArticles_base (type_article, categorie_type_article, prix_type_article, numero_fournisseur)

/* <ct, c, p, nf, nbd> ∈ TypesArticles ⇔ Un typeArticle est défini par un nom de type d'article ct, attribué à une catégorie c, avec un prix p, fournie par un fournisseur identifié par un un numéro de fournisseur nf, (ainsi que sa quantité de produit nbd disponible (vu dynamique)) */

Fournisseurs (numero_fournisseur, nom_fournisseur, adresse_fournisseur)

/* <nf, nomf, af> ∈ Fournisseurs ⇔ Un fournisseur est définie par son numéro de fournisseur nf. Il a un nom nomf ainsi qu'une adresse af */

(View) Commandes (numero_commande, numero_client, date_commande, quantite_article)

(View) TypesArticles

(type_article, categorie_type_article, prix_type_article, numero_fournisseur, quantite_disponible_article)

(View) Clients (numero_client, nom_client, prenom_client, adresse_client, email_client, nombre_commande_client)

Les domaines associés sont :

domaine (numero_client) = domaine (numero_commande) = domaine (numero_fournisseur)
= domaine (quantite_article) = domaine (nombre_commande_client) = domaine
(quantite_disponible_article) = entiers >0

domaine (nom_client) = domaine (numero_fournisseurs) = domaine (prenom_client) =
domaine (adresse_client)
= domaine (adresse_fournisseur) = domaine (email_client) = domaine (code_article)
= domaine (type_article) = domaine (categorie_type_article) = domaine (statut_article)
= chaîne de caractère

domaine (prix_type_article) = float >0
domaine (date_commande) = date

Les contraintes d'intégrité référentielles

Commandes_base [numero_client] \subseteq Clients_base [numero_client]

*/*Un client peut ne pas posséder de commande*/*

Articles [type_article] \subseteq TypesArticles_base [type_article]

*/*Car correspond à la cardinalité, un Type Article peut avoir 0 ou plusieurs Articles*/*

ArticlesCommandes [numero_commande] = Commandes_base [numero_commande]

*/*Car correspond à la cardinalité, une Commandes peut avoir 1 ou plusieurs ArticlesCommandes*/*

ArticlesCommandes [code_article] \subseteq Articles[code_article]

*/*Un Article peut ne pas être commandé */*

TypesArticles_base [numero_fournisseur] = Fournisseurs [numero_fournisseur]

*/*Un seul fournisseur fournit plusieurs types d'articles*/*

categorie \in {Fruits et Légumes, Boucherie,Épicerie,Boissonerie,Surgelé,Boulangerie}

statut \in {disponible, commande}

Autres contraintes

On ne peut pas attribuer automatiquement un code barre à un produit, on est obligé de vérifier les codes déjà pris dans les tables. (En fait si, on a réussi avec python)

(Traduction en annexe)

Troisième partie Fonctionnalités simplifiées d'une application Python et SQLite

Introduction

Pour cette partie nous avons implémenté une interface python pour utiliser la base de données de notre boutique. Cette interface comprend plusieurs menus à différents niveaux permettant aux utilisateurs de passer d'un menu à l'autre en entrant dans le terminal ce qui est indiqué par le jeu d'affichage. L'utilisateur peut effectuer différentes tâches dans chaque menu et ainsi suivre les liens logiques de notre interface liant tous les menus du programme, lui permettant ainsi de faire et refaire ces tâches comme bon lui semble.

Requêtes utilisées

Les 3 types de requêtes (sélection-projection / opérateurs ensemblistes / jointure-agrégation) sont utilisés tout au long du programme, il suffit de voir le code python des différentes fonctions pour les voir de manière plus précise.

SELECT/JOIN :

- Utilisés dans la plupart des fonctions
- Tous les affichages
- Les vérifications des inputs

INSERT :

- Inscription de nouveaux clients
- Passage de nouvelles commandes
- Ajout fournisseur/types articles/n nombre articles/... (côté admin)

DELETE :

- Supprimer clients/article/commande (côté admin)

UPDATE :

- pour la modification des prix (pas eu le temps de coder d'autres update)

Fonctionnement de l'interface

Pour exécuter notre programme, il suffit seulement de rentrer une seule commande

python main.py

Tout le reste est codé dans des sous-fonctions appelées dans la fonction main selon les choix entrés par l'utilisateur.

Menu Principal

Nous commençons tout d'abord par l'affichage d'un Menu Principal, correspondant à la racine de notre boutique, c'est ici que nous séparons le côté client et admin.

Normalement, seul le côté client est librement disponible pour les utilisateurs, mais nous avons ajouté une option admin sécurisée par un mot de passe, permettant d'effectuer d'autres requêtes importantes au gérant de la boutique.

Côté Client

Du côté client, d'autres sous-menus s'ouvriront selon les choix du client, où il aura tout d'abord la possibilité de s'inscrire, se connecter, (pour voir les produits il faut d'abord se connecter) ,... qui correspondent aux choix libres et accessibles à tous les utilisateurs.

Si le client décide de s'inscrire, il sera alors ajouté à la base de données et on lui attribuera un numéro client qui correspond à son code de connexion avec son nom et prénom (un numéro client dont il faut se souvenir !).

Si le client décide de se connecter, une vérification de son numéro client, nom et prénom sera d'abord effectuée puis plus d'options s'offriront à lui.
comme le fait de voir les produits/rayons/caractéristiques des produits de la boutique, passer des commandes en y ajoutant autant de produits qu'il souhaite, voir ses commandes. cette partie est plus restreinte et ne peut être vue que par le client, ou bien celui qui connaît les informations personnelles du client.

Côté Administrateur

L'administrateur est le chef de la boutique, c'est celui qui a accès à tout. Mais pour y accéder, il lui suffit d'entrer un mot de passe assez complexe pour éviter de se faire pirater la base de donnée, sur le menu principal.

Ici, le mot de passe de l'administrateur a été simplifié pour le test, c'est : **403**

Dans ce menu, l'administrateur peut quasiment tout faire, inscrire un client, ajouter des commandes/produits/fournisseurs/types articles, supprimer des clients/commandes/articles, annuler des produits commandés, modifier le prix des articles, et afficher toutes les bases de données de la boutique mais aussi réinitialisé la base de donnée.

Dans votre cas, si vous avez testé le menu en ayant fait beaucoup de modifications, et que vous souhaitez revenir à la base de donnée initiale de test, vous devez aller dans **admin>mdp:403>13>OUI** , pour réinitialiser complètement la base de données.

Conclusion projet

Nous avons pris beaucoup de plaisir à faire ce projet, depuis la conception au brouillon jusqu'à sa finalisation concrète en python. Nous n'avons pas réalisé tout ce que nous souhaitions, mais nous considérons tout de même ce projet terminé et fonctionnel
Merci pour nous avoir encadré

Traduction Sql (Annexe)

Création des tables + view + trigger

```
DROP TABLE IF EXISTS Commandes ;
DROP TABLE IF EXISTS ArticlesCommandes ;
DROP TABLE IF EXISTS Articles ;
DROP TABLE IF EXISTS TypesArticles ;
DROP TABLE IF EXISTS Fournisseurs ;
DROP TABLE IF EXISTS Clients ;
DROP VIEW IF EXISTS Commandes_View;
DROP VIEW IF EXISTS TypesArticles_View;
DROP VIEW IF EXISTS Clients_View;
DROP TRIGGER IF EXISTS update_status_commande;
DROP TRIGGER IF EXISTS update_status_disponible;
```

```
PRAGMA FOREIGN_KEYS=ON;
```

```
CREATE TABLE Clients (
    numero_client INTEGER NOT NULL,
    nom_client TEXT NOT NULL,
    prenom_client TEXT NOT NULL,
    adresse_client TEXT NOT NULL,
    email_client TEXT,
    CONSTRAINT pk_client_c00 PRIMARY KEY (numero_client)
);
```

```
CREATE TABLE Commandes(
    numero_commande INTEGER NOT NULL,
    numero_client INTEGER NOT NULL,
    date_commande DATE NOT NULL,
    CONSTRAINT pk_Commandes_c00 PRIMARY KEY (numero_commande),
    CONSTRAINT fk_Commandes_c01 FOREIGN KEY (numero_client) REFERENCES
Clients(numero_client)
);
```

```
CREATE TABLE Fournisseurs(
    numero_fournisseur INTEGER NOT NULL,
    nom_fournisseur TEXT NOT NULL,
    adresse_fournisseur TEXT NOT NULL,
    CONSTRAINT pk_Fournisseurs_c00 PRIMARY KEY (numero_fournisseur)
);
```

```
CREATE TABLE TypesArticles(
    type_article TEXT NOT NULL,
    categorie_type_article TEXT NOT NULL,
    prix_type_article REAL NOT NULL,
    numero_fournisseur INTEGER NOT NULL,
    CONSTRAINT pk_TypesArticles_c00 PRIMARY KEY (type_article),
    CONSTRAINT fk_TypesArticles_c01 FOREIGN KEY (numero_fournisseur) REFERENCES
Fournisseurs (numero_fournisseur),
    CONSTRAINT ck_TypesArticles_c03 CHECK (categorie_type_article in ("Fruits
et Legumes","Boissonerie","Surgele","Boucherie","Epicerie","Boulangerie")),
```

```
CONSTRAINT ck_TypeArticle_c04 CHECK (prix_type_article > 0)
);
```

```
CREATE TABLE Articles(
    code_article TEXT NOT NULL,
    type_article TEXT NOT NULL,
    statut_article TEXT NOT NULL DEFAULT 'disponible',
    CONSTRAINT pk_Articles_c00 PRIMARY KEY (code_article),
    CONSTRAINT ck_Articles_c01 CHECK(statut_article in
    ("disponible","commande")),
    CONSTRAINT fk_Articles_c02 FOREIGN KEY (type_article) REFERENCES
    TypesArticles(type_article)
);
```

```
CREATE TABLE ArticlesCommandes(
    code_article TEXT NOT NULL,
    numero_commande INTEGER NOT NULL,
    CONSTRAINT pk_ArticlesCommandes_c00 PRIMARY KEY (code_article),
    CONSTRAINT fk_ArticlesCommandes_c01 FOREIGN KEY (code_article) REFERENCES
    Articles(code_article),
    CONSTRAINT fk_ArticlesCommandes_c02 FOREIGN KEY (numero_commande) REFERENCES
    Commandes(numero_commande) ON DELETE CASCADE ON UPDATE CASCADE
);
```

```
CREATE VIEW Clients_View(
    numero_client,
    nom_client,
    prenom_client,
    adresse_client,
    email_client,
    nombre_commande_client
) AS
SELECT numero_client,
       nom_client,
       prenom_client,
       adresse_client,
       email_client,
       COUNT(numero_commande)AS nombre_commande_client
FROM Clients
JOIN Commandes
USING (numero_client)
GROUP BY numero_client
UNION
SELECT numero_client,
       nom_client,
       prenom_client,
       adresse_client,
       email_client,
       0 AS nombre_commande_client
FROM Clients
WHERE numero_client NOT IN(SELECT numero_client
                           FROM Commandes);
```



```

CREATE VIEW Commandes_View(
    numero_commande,
    numero_client,
    date_commande,
    quantite_article
) AS
SELECT numero_commande,
        numero_client,
        date_commande,
        COUNT(code_article ) AS quantite_article
FROM Commandes
JOIN ArticlesCommandes
USING (numero_commande)
GROUP BY numero_commande;

CREATE VIEW TypesArticles_View(
    type_article,
    categorie_type_article,
    prix_type_article,
    numero_fournisseur,
    quantite_disponible_article
) AS
SELECT type_article,
        categorie_type_article,
        prix_type_article,
        numero_fournisseur,
        COUNT(code_article) AS quantite_disponible_article
FROM Articles
JOIN TypesArticles
USING (type_article)
WHERE (statut_article="disponible")
GROUP BY type_article;

```

Insertion de données

```

--Insertions OK
INSERT INTO Fournisseurs VALUES (1,'SOCOPA','ZI La petite Marine, 84800
LIsle-sur-la-Sorgue');
INSERT INTO Fournisseurs VALUES (2,'Nawhals','Willem Elsschotstraat 15, 1800
Vilvoorde, Belgique');
INSERT INTO Fournisseurs VALUES (3,'Lays','420 rue d Estienne d Orves 92705
Colombes Cedex');
INSERT INTO Fournisseurs VALUES (4,'Metro','ZI l Argentière - 8, rue
François Blumet, Sassenage, 38360');
INSERT INTO Fournisseurs VALUES (5,'Picard','375 Avenue des Frères Lumière,
84700 Sorgues');
INSERT INTO Fournisseurs VALUES (6,'Boulangier','84 Cours Jean Jaurès, 38130
Échirolles');

```

```
INSERT INTO Fournisseurs VALUES (7,'Le Jardin de Riad','Loin dans la
campagne, après Chambéry');
```

```
INSERT INTO TypesArticles VALUES ('Poulet','Boucherie',10.5,1);
INSERT INTO TypesArticles VALUES ('Merguez','Boucherie',6.95,1);
INSERT INTO TypesArticles VALUES ('Sauces bbq','Epicerie',3,2);
INSERT INTO TypesArticles VALUES ('Harissa','Epicerie',3.25,2);
INSERT INTO TypesArticles VALUES ('Chips','Epicerie',2,3);
INSERT INTO TypesArticles VALUES ('Bouga','Boissonerie',2.5,1);
INSERT INTO TypesArticles VALUES ('Leben','Boissonerie',1,1);
INSERT INTO TypesArticles VALUES ('Capri sun','Boissonerie',5,4);
INSERT INTO TypesArticles VALUES ('Poms','Boissonerie',4,4);
INSERT INTO TypesArticles VALUES ('Amigo','Surgele',2,5);
INSERT INTO TypesArticles VALUES ('Fiesta','Surgele',1.8,5);
INSERT INTO TypesArticles VALUES ('Baguette','Boulangerie',1,6);
INSERT INTO TypesArticles VALUES ('Pain','Boulangerie',1,6);
INSERT INTO TypesArticles VALUES ('Fraise','Fruits et Legumes',4,7);
INSERT INTO TypesArticles VALUES ('Banane','Fruits et Legumes',3,7);
INSERT INTO TypesArticles VALUES ('Dattes','Fruits et Legumes',5,7);
```

```
INSERT INTO Articles (code_article, type_article) VALUES
('Dattes1','Dattes');
INSERT INTO Articles (code_article, type_article) VALUES
('Dattes2','Dattes');
INSERT INTO Articles (code_article, type_article) VALUES
('Dattes3','Dattes');
INSERT INTO Articles (code_article, type_article) VALUES
('Dattes4','Dattes');
INSERT INTO Articles (code_article, type_article) VALUES
('Dattes5','Dattes');
INSERT INTO Articles (code_article, type_article) VALUES
('Dattes6','Dattes');
```

```
INSERT INTO Articles (code_article, type_article) VALUES
('Fraise1','Fraise');
INSERT INTO Articles (code_article, type_article) VALUES
('Fraise2','Fraise');
INSERT INTO Articles (code_article, type_article) VALUES
('Fraise3','Fraise');
INSERT INTO Articles (code_article, type_article) VALUES
('Fraise4','Fraise');
INSERT INTO Articles (code_article, type_article) VALUES
('Fraise5','Fraise');
INSERT INTO Articles (code_article, type_article) VALUES
('Fraise6','Fraise');
```

```
INSERT INTO Articles (code_article, type_article) VALUES
('Banane1','Banane');
INSERT INTO Articles (code_article, type_article) VALUES
('Banane2','Banane');
INSERT INTO Articles (code_article, type_article) VALUES
('Banane3','Banane');
```

```

INSERT INTO Articles (code_article, type_article) VALUES
('Banane4','Banane');
INSERT INTO Articles (code_article, type_article) VALUES
('Banane5','Banane');
INSERT INTO Articles (code_article, type_article) VALUES
('Banane6','Banane');
INSERT INTO Articles (code_article, type_article) VALUES
('Banane7','Banane');
INSERT INTO Articles (code_article, type_article) VALUES
('Banane8','Banane');

```

```

INSERT INTO Articles (code_article, type_article) VALUES
('Baguette1','Baguette');
INSERT INTO Articles (code_article, type_article) VALUES
('Baguette2','Baguette');
INSERT INTO Articles (code_article, type_article) VALUES
('Baguette3','Baguette');
INSERT INTO Articles (code_article, type_article) VALUES
('Baguette4','Baguette');
INSERT INTO Articles (code_article, type_article) VALUES
('Baguette5','Baguette');
INSERT INTO Articles (code_article, type_article) VALUES
('Baguette6','Baguette');
INSERT INTO Articles (code_article, type_article) VALUES
('Baguette7','Baguette');
INSERT INTO Articles (code_article, type_article) VALUES
('Baguette8','Baguette');
INSERT INTO Articles (code_article, type_article) VALUES
('Baguette9','Baguette');
INSERT INTO Articles (code_article, type_article) VALUES
('Baguette10','Baguette');

```

```

INSERT INTO Articles (code_article, type_article) VALUES ('Pain1','Pain');
INSERT INTO Articles (code_article, type_article) VALUES ('Pain2','Pain');
INSERT INTO Articles (code_article, type_article) VALUES ('Pain3','Pain');
INSERT INTO Articles (code_article, type_article) VALUES ('Pain4','Pain');
INSERT INTO Articles (code_article, type_article) VALUES ('Pain5','Pain');
INSERT INTO Articles (code_article, type_article) VALUES ('Pain6','Pain');
INSERT INTO Articles (code_article, type_article) VALUES ('Pain7','Pain');

```

```

INSERT INTO Articles (code_article, type_article) VALUES ('Capri
sun1','Capri sun');
INSERT INTO Articles (code_article, type_article) VALUES ('Capri
sun2','Capri sun');
INSERT INTO Articles (code_article, type_article) VALUES ('Capri
sun3','Capri sun');
INSERT INTO Articles (code_article, type_article) VALUES ('Capri
sun4','Capri sun');

```

```

INSERT INTO Articles (code_article, type_article) VALUES
('Fiestal','Fiesta');

```

```
INSERT INTO Articles (code_article, type_article) VALUES
('Fiesta2', 'Fiesta');
INSERT INTO Articles (code_article, type_article) VALUES
('Fiesta3', 'Fiesta');
```

```
INSERT INTO Articles (code_article, type_article) VALUES ('Chips1', 'Chips');
INSERT INTO Articles (code_article, type_article) VALUES ('Chips2', 'Chips');
INSERT INTO Articles (code_article, type_article) VALUES ('Chips3', 'Chips');
INSERT INTO Articles (code_article, type_article) VALUES ('Chips4', 'Chips');
INSERT INTO Articles (code_article, type_article) VALUES ('Chips5', 'Chips');
INSERT INTO Articles (code_article, type_article) VALUES ('Chips6', 'Chips');
INSERT INTO Articles (code_article, type_article) VALUES ('Chips7', 'Chips');
INSERT INTO Articles (code_article, type_article) VALUES ('Chips8', 'Chips');
INSERT INTO Articles (code_article, type_article) VALUES ('Chips9', 'Chips');
INSERT INTO Articles (code_article, type_article) VALUES
('Chips10', 'Chips');
```

```
INSERT INTO Articles (code_article, type_article) VALUES
('Poulet1','Poulet');
INSERT INTO Articles (code_article, type_article) VALUES
('Poulet2','Poulet');
INSERT INTO Articles (code_article, type_article) VALUES
('Poulet3','Poulet');
INSERT INTO Articles (code_article, type_article) VALUES
('Poulet4','Poulet');
INSERT INTO Articles (code_article, type_article) VALUES
('Poulet5','Poulet');
INSERT INTO Articles (code_article, type_article) VALUES
('Poulet6','Poulet');
INSERT INTO Articles (code_article, type_article) VALUES
('Poulet7','Poulet');
INSERT INTO Articles (code_article, type_article) VALUES
('Poulet8','Poulet');
INSERT INTO Articles (code_article, type_article) VALUES
('Poulet9','Poulet');
```

```

INSERT INTO Articles (code_article, type_article) VALUES
('Merguez1','Merguez');
INSERT INTO Articles (code_article, type_article) VALUES
('Merguez2','Merguez');
INSERT INTO Articles (code_article, type_article) VALUES
('Merguez3','Merguez');
INSERT INTO Articles (code_article, type_article) VALUES
('Merguez4','Merguez');
INSERT INTO Articles (code_article, type_article) VALUES
('Merguez5','Merguez');

INSERT INTO Articles (code_article, type_article) VALUES ('Sauces
bbq1','Sauces bbq');
INSERT INTO Articles (code_article, type_article) VALUES ('Sauces
bbq2','Sauces bbq');

INSERT INTO Articles (code_article, type_article) VALUES
('Harissa1','Harissa');
INSERT INTO Articles (code_article, type_article) VALUES
('Harissa2','Harissa');
INSERT INTO Articles (code_article, type_article) VALUES
('Harissa3','Harissa');

INSERT INTO Clients VALUES (0,'BOUISSA','Ilyass','4 rue de
lavenue','bouissailyass@gmail.com');
INSERT INTO Clients VALUES (1,'GUERBAA','Rayan','Rue de la
paix','bibiztoz@gmail.com');
INSERT INTO Clients VALUES (2,'Mahi','Riad','21 rue de l
ivrogne','bebou_le_pilo@Hotmail.com');
INSERT INTO Clients VALUES (3,'Bebou','Jessyca','17 rue la poste dla
desco','hot_watifan@laposte.por');
INSERT INTO Clients VALUES (4,'Gorur','Selim','rue de la
condescendance','jaiquel0@yahooo.rios');

INSERT INTO Commandes VALUES (1,0,'2021-04-13');
INSERT INTO Commandes VALUES (2,1,'2021-04-14');

INSERT INTO ArticlesCommandes VALUES ('Poulet1',1);
INSERT INTO ArticlesCommandes VALUES ('Poulet2',1);
INSERT INTO ArticlesCommandes VALUES ('Merguez1',1);
INSERT INTO ArticlesCommandes VALUES ('Merguez2',2);
INSERT INTO ArticlesCommandes VALUES ('Merguez3',2);

```

--Insertions NOK (après avoir exécuté OK)

```

--Fournisseurs
--Erreur pk fournisseur 1 déjà utilisé
INSERT INTO Fournisseurs VALUES (1,'AVS','4 Rue Lainerie, 69005 Lyon');

```

```

--TypesArticles
--Erreur ck enum Boulangerie n'existe pas dans l'enum
INSERT INTO TypesArticles VALUES ('Baguette','Boulangerie',1.1,1);
--Erreur pk Poulet existe déjà
INSERT INTO TypesArticles VALUES ('Poulet','Epicerie',10.5,1);
--Erreur fk Fournisseurs 5 et 3 n'existent pas
INSERT INTO TypesArticles VALUES ('Poivre','Epicerie',0.55,5);
INSERT INTO TypesArticles VALUES ('Steak','Boucherie',10.5,3);
--Erreur ck prix doit être >0
INSERT INTO TypesArticles VALUES ('Steak','Boucherie',-10,2);

--Articles
--Erreur pk P1 existe déjà
INSERT INTO Articles (code_article, type_article) VALUES ('P1','Poulet');
INSERT INTO Articles (code_article, type_article) VALUES ('P1','Poulet');
--Erreur fk type poivre n'existe pas
INSERT INTO Articles (code_article, type_article) VALUES ('Po1','Poivre');

--Clients
--Erreur pk
INSERT INTO Clients VALUES (0,'Mahi','Riad','21 rue de l
ivrogne','bebou_le_pilo@Hotmail.com');

--Commandes
--Erreur ck manque la date
INSERT INTO Commandes VALUES (3,0);
--Erreur pk 1 commande existe déjà
INSERT INTO Commandes VALUES (1,0,'2021-04-15');
--Erreur fk client 2 n'existe pas
INSERT INTO Commandes VALUES (1,2,'2021-04-15');

--ArticlesCommandes
--Erreur pk article p3 déjà commandé dans une autre commande
INSERT INTO ArticlesCommandes VALUES ('P2',2);
--Erreur fk p6 n'existe pas
INSERT INTO ArticlesCommandes VALUES ('P6',1);
--Erreur fk article p3 n'existe pas
INSERT INTO ArticlesCommandes VALUES ('P3',3);

```