



BestMarket

DOCUMENTATION TECHNIQUE CREATION BASE DE DONNEES

Titre du projet	Customer Data Feedback
Contact Business	Olivier, Responsable du Service Clients

Version	Auteur	Description	Date
V1	Frédéric NALLET	Business Intelligence Analyst	01.12.2023

Contexte de la documentation

Procédure de la création de la base de données par étapes et des différents éléments qui la compose :

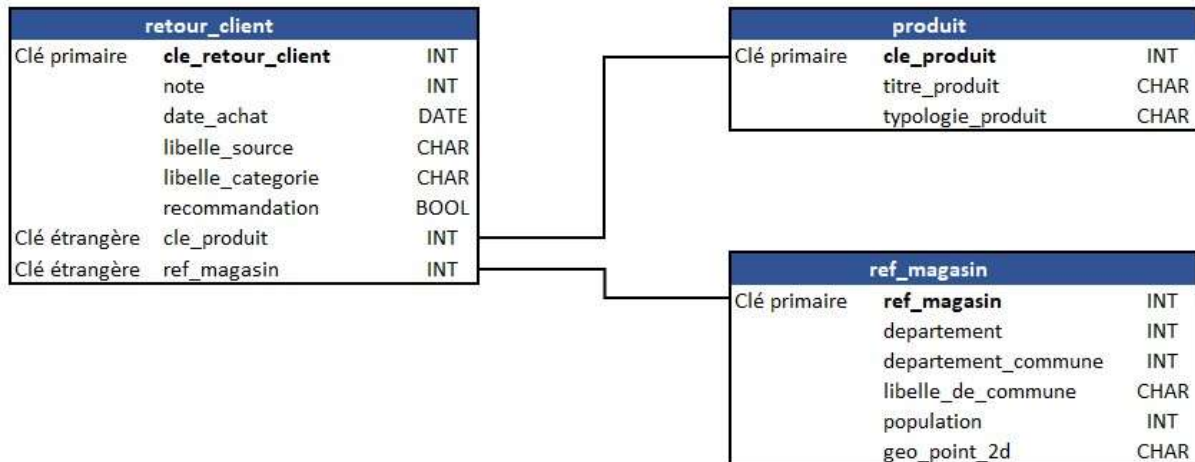
- Schema des tables avec leur jointure
- Definition du dictionnaire
- Création de la base
- Création des tables
- Chargement des données



Création de la base de données

1> Le schéma UML complet

Schema bdd



2> Dictionnaire des données

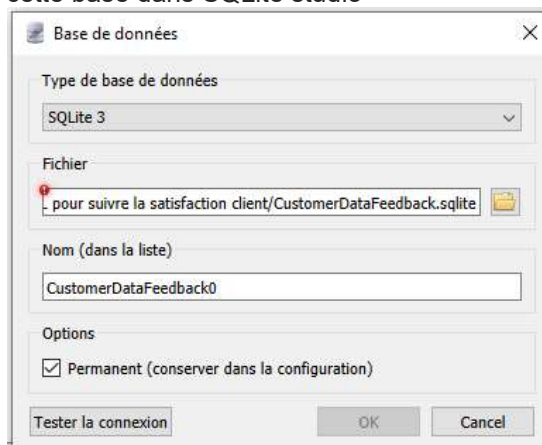
Dictionnaire données

Nom du champs	Type de données	Taille	Contrainte	Description	Table
cle_retour_client	INT		Clé primaire	Id unique pour les retours clients	retour_client
note	INT			Note donnée par le client, comprise entre 0 et 10, la note est la réponse à la question	
date_achat	DATE			Date à laquelle l'achat du client a eu lieu	
libelle_source	CHAR	50		Libellé de la source d'où provient le retour client (Réseaux sociaux, téléphone, email)	
libelle_categorie	CHAR	50		Libellé de la catégorie du retour client (Drive, service après-vente, qualité produit,	
recommandation	BOOL			Recommandation laissée par le client à la question 'Recommandez vous l'entreprise?'	
cle_produit	INT		Clé étrangère	Id unique pour les produits	produit
ref_magasin	INT			Id unique pour chaque magasin	
cle_produit	INT		Clé primaire		
titre_produit	CHAR	50		Libellé des produits	
typologie_produit	CHAR	50		Typologie des produits (Alimentaire, High-tech etc...)	ref_magasin
ref_magasin	INT		Clé primaire	Id unique pour chaque magasin	
departement	INT	3		departement	
departement_commune	INT	5		Code postal	
libelle_de_commune	CHAR	50		Commune	
population	INT			Nombre d'habitants	
geo_point_2d	CHAR			Coordonnées géographiques	

3> Création de la base de données à partir d'un document txt et le renommer avec une extension SQLite.

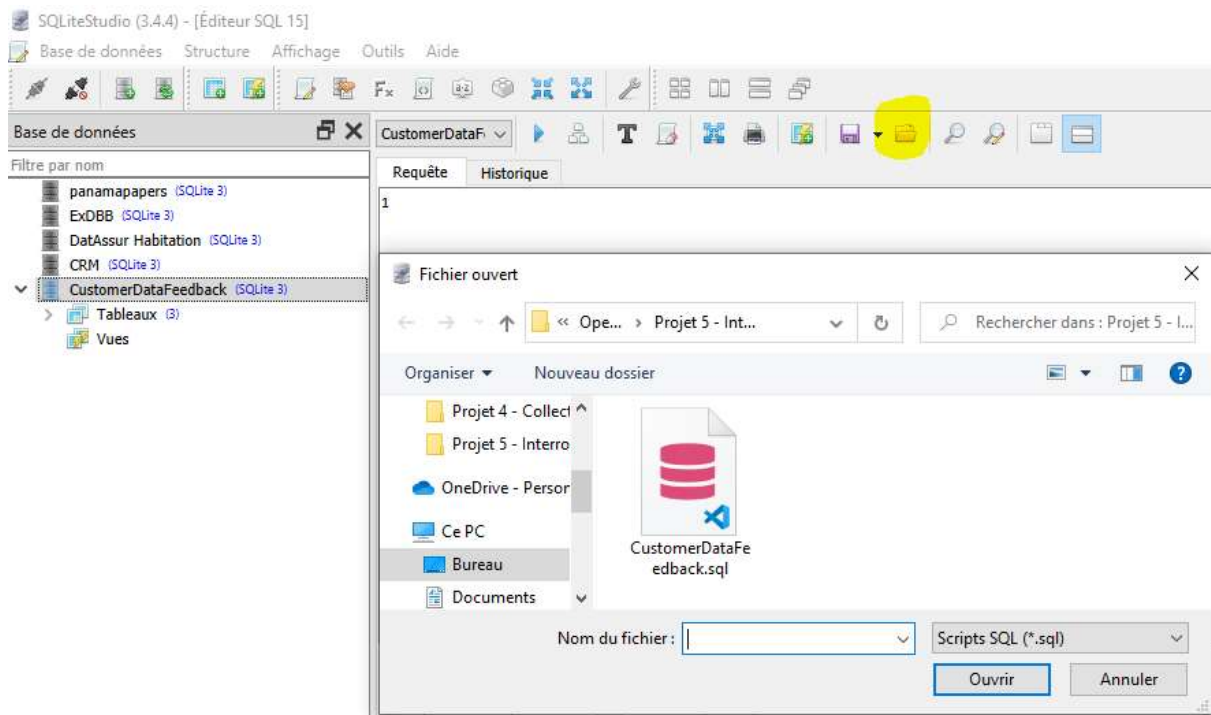
CustomerDataFeedback.txt CustomerDataFeedback.sqlite

4> Charger le fichier de cette base dans SQLite studio





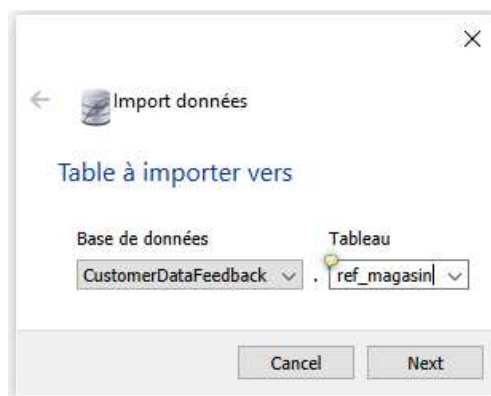
5> Importation du fichier SQL dans la base de données



6> Création des différents attributs de la table « ref_magasin » avec leur typologie et autres caractéristiques.

CustomerDataF		Nom de la table :	ref_magasin							<input type="checkbox"/> WITHOUT ROWID	<input type="checkbox"/> STRICT
	Nom	Type de données	Clé primaire	Clé étrangère	Unique	Contrôle	Non NULL	Collecter	Généré		
1	ref_magasin	INT									
2	departement	INT (3)									
3	departement_commune	INT (5)									
4	libelle_de_commune	CHAR (50)									
5	population	INT									
6	geo_point_2d	CHAR									

7> Chargement des données de la table ref_magasin





BestMarket

← Import données

Source de données à importer de

Type de données source
CSV

Options

Fichier : se de données avec SQL pour suivre la satisfaction client/ref_magasin.csv

Texte codé : UTF-8

☐ Ignorer les erreurs

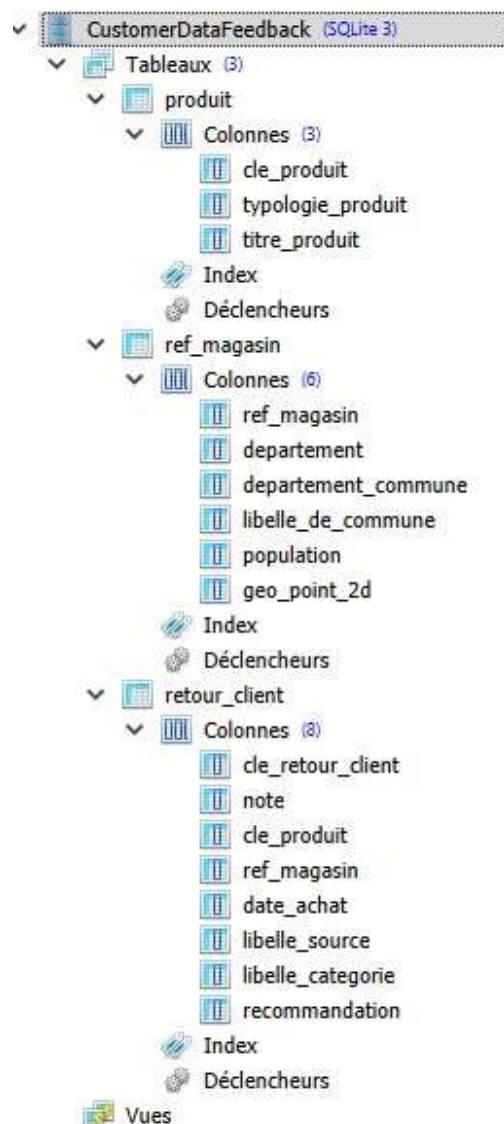
Options de source de données

☒ La première ligne représente les noms de colonnes CSV

Séparateur de colonnes : ; (point virgule)

Cancel Finish

Vue d'ensemble de la base avec ses tables





8> Vérification des doublons dans la base par une requête

```
1 SELECT COUNT(*) AS nombre_doublon, cle_produit, typologie_produit, titre_produit
2 FROM produit
3 GROUP BY cle_produit, typologie_produit, titre_produit
4 HAVING COUNT(*)>1
```

Table		Formulaire	
Nombre de lignes chargées : 0			
nombre_doublon	cle_produit	typologie_produit	titre_produit

Aucun doublon dans la table produit

```
1 SELECT COUNT(*) AS nombre_doublon, ref_magasin, departement, departement_commune, libelle_de_commune, population, geo_point_2d
2 FROM ref_magasin
3 GROUP BY ref_magasin, departement, departement_commune, libelle_de_commune, population, geo_point_2d
4 HAVING COUNT(*)>1
```

Table		Formulaire	
Nombre de lignes chargées : 0			
nombre_doublon	ref_magasin	departement	departement_commune
libelle_de_commune	population	geo_point_2d	

Aucun doublon dans la table ref_magasin

```
1 SELECT COUNT(*) AS nombre_doublon, cle_retour_client, note, cle_produit, ref_magasin, date_achat, libelle_source, libelle_categorie, recommandation
2 FROM retour_client
3 GROUP BY cle_retour_client, note, cle_produit, ref_magasin, date_achat, libelle_source, libelle_categorie, recommandation
4 HAVING COUNT(*)>1
```

Table		Formulaire	
Nombre de lignes chargées : 0			
nombre_doublon	cle_retour_client	note	cle_produit
ref_magasin	date_achat	libelle_source	libelle_categorie
recommandation			

Aucun doublon dans la table retour_client

9 > Vérification si certains attributs ont des données „vides“

```
1 SELECT *
2 FROM retour_client
3 WHERE cle_retour_client = '' OR note = '' OR cle_produit = '' OR ref_magasin = ''
4 OR date_achat = '' OR libelle_source = '' OR libelle_categorie = '' OR recommandation = ''
```

Table		Formulaire	
Nombre de lignes chargées : 674			
	cle_retour	note	cle_produ
	ref_maga	date_achat	libelle_source
	libelle_categorie	recommar	
1	157308	1	142
2	157381	1	94
3	157897	1	117

La table retour_client possèdent des champs vides sur 674 lignes dans la colonne „recommandation“.

Il faudra en tenir compte pour certains calculs.