

Document technique Projet 3 :

Requêtez une base de données avec SQL

Sommaire :

- Introduction
- Exploration des types de données et dictionnaires des données
- Schéma relationnel et création de la base de données
 - Schéma relationnel
 - Code SQL pour la création des tables
- Chargements des données
- Analyses des requêtes

Introduction

Dans le cadre de ce projet, nous accompagnons une entreprise du secteur de l'assurance souhaitant mieux comprendre le marché des assurances habitation afin d'améliorer l'accompagnement de ses clients. Pour cela, plusieurs jeux de données ont été mis à disposition : les données des contrats clients ainsi qu'un référentiel géographique des régions françaises provenant de data.gouv.fr.

L'objectif du projet est de structurer ces données dans une base relationnelle robuste, permettant de faciliter leur analyse. À partir des fichiers sources, il s'agit de comprendre la nature des variables, d'établir un dictionnaire des données, de concevoir un modèle relationnel, puis de créer la base SQL correspondante. Enfin, les données devront être chargées dans les tables afin de réaliser les premières requêtes d'analyse.

Nous allons dans un premier temps analyser nos fichiers csv afin d'identifier les différents types de variables, les contraintes et ainsi construire notre dictionnaire de données

Exploration des types de données et dictionnaires des données

Nous allons tout d'abord analyser nos deux fichiers csv afin de pouvoir déterminer les types des variables et les contraintes présentes.

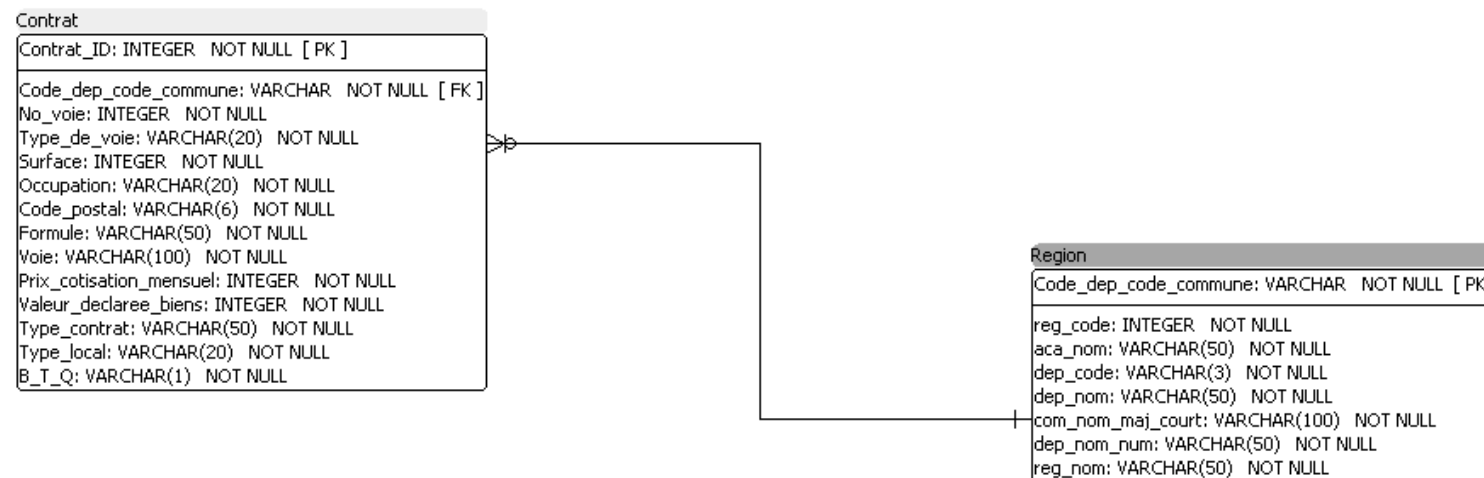
Nous avons deux fichiers qui vont correspondre chacun à une table, nous listons les différentes variables selon leur table dans un fichier Excel et nous procédons à l'analyse des différents types de variables possibles, nous insérons une colonne afin d'expliquer le choix du typage de la variable.

	Nom des colonnes	Type de données	Taille	Clé	Description	Choix du type de variable
CONTRAT.CSV	Contrat_ID	INT		Clé primaire	Id unique pour les contrats	Int correspondant à un numéro de contrat
	No_voie	INT			Numéro dans la voie pour l'adresse du logement assuré	Numéro de voie = Int
	B_T_Q	CHAR	1		Indicateur éventuel de répétition pour l'adresse du logement assuré sur un caractère	Répétition d'adresse indiquée par une lettre donc Char(1)
	Type_de_voie	VARCHAR	20		Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ...	Indication texte courte pour le type de voie
	Voie	VARCHAR	100		Libellé de la voie pour l'adresse du logement assuré	Nom de la rue en chaîne de caractère, taille 100 pour les adresses plus longues
	Code_dep_code_commune	Char	6	Clé étrangère	Concaténation du code département et code commune pour avoir une clé unique	Code Postal en chaîne de caractère notamment pour la Corse 2A/2b de taille 6 (dom/tom) / clé étrangère liant les tables
	Code_postal	Char	6		Code postal pour l'adresse du logement assuré	Idem que pour la variable Code_dep_code_commune
	Surface	int			Surface du logement	Indiqué en M² donc par un nombre
	type_local	VARCHAR	20		Type du Logement	"appartement" / "Maison" donc chaîne de caractère de taille 20
	Occupation	VARCHAR	20		Type de l'occupation (ex: "Locataire", "Propriétaire")	Chaîne de caractère de taille 20
	Type_contrat	VARCHAR	50		Type de contrat (Résidence principale, mise en location)	Chaîne de caractère de taille 50
	Formule	VARCHAR	50		type de la formule (intégrale ou classique)	Chaîne de caractère de taille 50
	Valeur_declaree_biens	VARCHAR	50		Valeur des biens déclarés dans le logement	Chaîne de caractère de taille 50
	Prix_cotisation_mensuel	int			Montant de la cotisation	Chiffre exprimé en euro donc Int
REGION.CSV	Code_dep_code_commune	CHAR	6	Clé primaire	Concaténation du code département et code commune pour avoir une clé unique	Code Postal en chaîne de caractère notamment pour la Corse 2A/2b de taille 6 (dom/tom)
	reg_code	int			Code région	Code de région exprimé en numéro donc INT
	reg_nom	VarChar	50		Nom complet de la région	Nom complet de la région désigné par une chaîne de caractères
	aca_nom	VARCHAR	50		Préfecture de région	Préfecture de région désignée par une chaîne de caractères
	dep_nom	VarChar	50		Nom département	Nom département donc chaîne de caractères
	com_nom_maj_court	VarChar	100		Nom commune	Chaîne de caractère pour le nom de Commune et taille 100 pour les communes avec un long nom
	dep_code	Char	3		numéro département	Chaîne de caractère pour prendre en charge le code de la Corse (2A) et (2b) et de taille 3 pour les Dom/Tom
	dep_nom_num	VarChar	50		concaténation nom département et code département	concaténation de deux chaînes de caractère donc chaîne de caractère

Schéma relationnel et création de la base de données

1. Schéma relationnel.

Maintenant que nous avons établis notre dictionnaire de données, nous allons pouvoir établir un schéma relationnel de notre base de données avec nos deux tables et leurs différents caractéristiques et contraintes. Pour ce faire nous allons utiliser le logiciel SQL Power Architect.



Code SQL pour la création de la base de données.

Nous allons maintenant créer notre base de données sur le SGBD que nous avons choisis (MySQL Workbench).

Nous commençons par la création de la base avec un CREATE DATABASE puis nous nous occupons de créer les tables avec les champs et leur type et contraintes que nous avons établis précédemment.

- Create database Soutenance;

- Use Soutenance;

- CREATE TABLE Region (

```
Code_dep_code_commune VARCHAR(10) NOT NULL, |
```

```
reg_code INT NULL,
```

```
reg_nom VARCHAR(50) NULL,
```

```
aca_nom VARCHAR(100) NULL,
```

```
dep_nom VARCHAR(50) NULL,
```

```
com_nom_maj_court VARCHAR(100) NULL,
```

```
dep_code INT NULL,
```

```
dep_nom_num VARCHAR(20) NULL,
```

```
PRIMARY KEY (Code_dep_code_commune)
```

```
);
```

- CREATE TABLE Contrat (

```
Contrat_ID INT NOT NULL,
```

```
No_voie INT NULL,
```

```
B_T_Q CHAR(1) NULL,
```

```
Type_de_voie VARCHAR(50) NULL,
```

```
Voie VARCHAR(100) NOT NULL,
```

```
Code_dep_code_commune VARCHAR(10) NULL,
```

```
Code_postal INT NULL,
```

```
Surface INT NULL,
```

```
Type_local VARCHAR(20) NULL,
```

```
Occupation VARCHAR(20) NULL,
```

```
Type_contrat VARCHAR(50) NULL,
```

```
Formule VARCHAR(50) NULL,
```

```
Valeur_declaree_biens VARCHAR(50) NULL,
```

```
Prix_cotisation_mensuel INT NULL,
```

```
PRIMARY KEY (Contrat_ID),
```

```
CONSTRAINT fk_contrat_region FOREIGN KEY (Code_dep_code_commune)
```

```
REFERENCES Region(Code_dep_code_commune)
```

```
ON UPDATE CASCADE
```

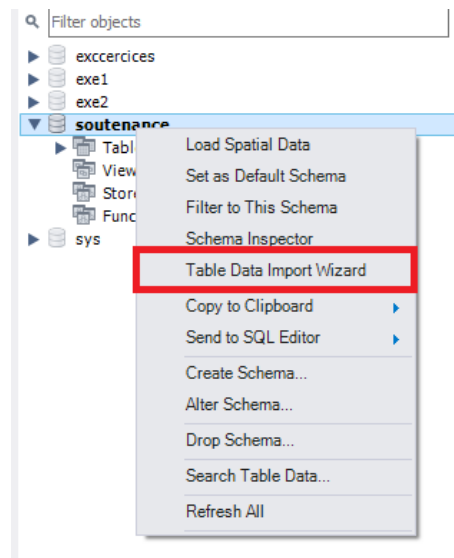
```
ON DELETE SET NULL
```

```
);
```

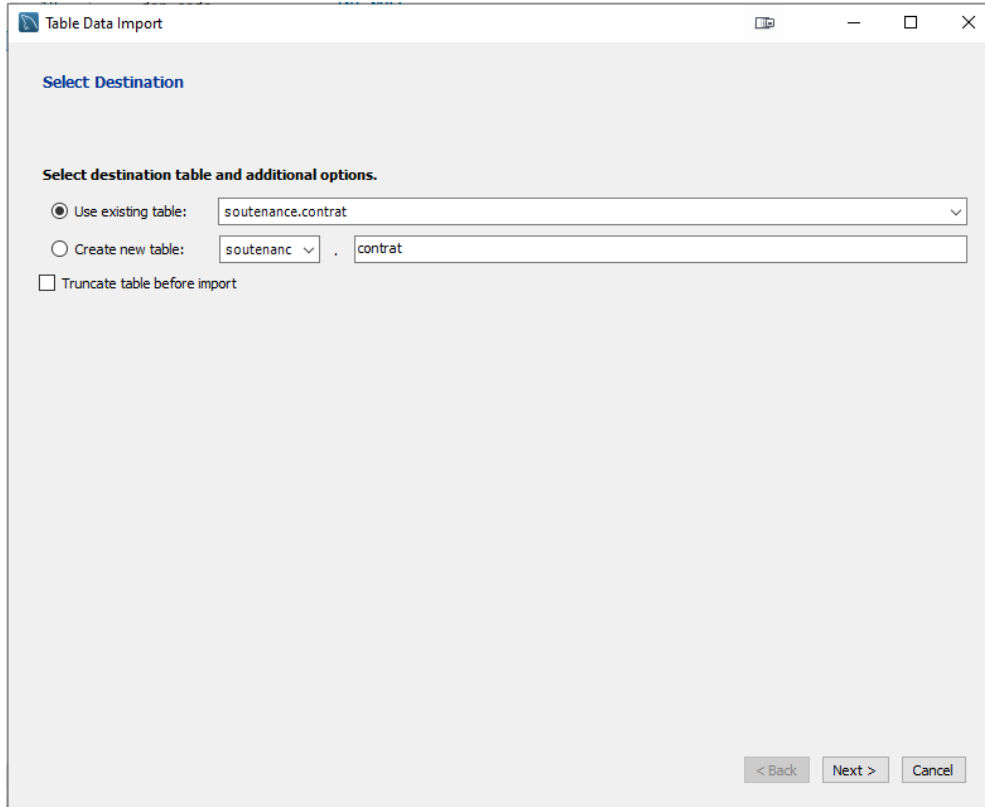
Chargement des données dans My SQL Workbench

La base de données étant maintenant créée ainsi que les tables et leurs dépendances, nous devons maintenant insérer les données présentes dans nos fichiers csv afin de pouvoir commencer notre travail d'analyse.

Pour ce faire nous allons utiliser l'outil proposé par MySQL WorkBench : Table Data Import Wizard, c'est un assistant graphique qui simplifie l'import de donnée dans les tables sélectionnées.



Une fois le fichier csv choisis nous pouvons ou de choisir de créer une nouvelle table ou utilisé une table déjà existante, nous choisissons d'utiliser les tables que nous venons de créer.



The screenshot shows a window titled "Table Data Import" with a standard Windows title bar (minimize, maximize, close buttons). The main content area is titled "Select Destination" in blue text. Below this, there is a section titled "Select destination table and additional options." in bold. This section contains two radio button options: "Use existing table:" (selected) and "Create new table:". The "Use existing table:" option has a text box containing "soutenance.contrat" with a dropdown arrow. The "Create new table:" option has two text boxes: the first contains "soutenanc" with a dropdown arrow, followed by a period, and the second contains "contrat". Below these options is a checkbox labeled "Truncate table before import" which is currently unchecked. At the bottom right of the window, there are three buttons: "< Back", "Next >", and "Cancel".

Table Data Import

Select Destination

Select destination table and additional options.

☒ Use existing table: soutenance.contrat

☐ Create new table: soutenanc . contrat

☐ Truncate table before import

< Back Next > Cancel

Ensuite nous prenons le soin de vérifier la correspondance de nos champs de nos table SQL avec les champs présents dans les fichiers csv.

Table Data Import

Configure Import Settings

Detected file format: csv

Encoding: utf-8

Columns:

Source Column	Dest Column
<input checked="" type="checkbox"/> Contrat_ID	Contrat_ID
<input checked="" type="checkbox"/> No_voie	No_voie
<input checked="" type="checkbox"/> B_T_Q	B_T_Q
<input checked="" type="checkbox"/> Type_de_voie	Type_de_voie
<input checked="" type="checkbox"/> Voie	Voie
<input checked="" type="checkbox"/> Code_dep_code_commune	Code_dep_code_commune

Contrat_ID	No_voie	B_T_Q	Type_de_...	Voie	Code_dep...	Code_postal	Surface	Type_local	Occupai
100601	190	A	RUE	CENTRALE	1350	1370	50	Appartement	Locatai
100602	347		RUE	DU CHATEAU	1103	1170	48	Appartement	Locatai
100603	58		AV	DU MONT...	1143	1220	131	Appartement	Proprie
100604	140		RUE	DE L'ABBE...	1288	1630	109	Maison	Locatai

< Back Next > Cancel

Nous obtenons à la suite de ses insertions nos 2 table remplis avec nos données.


```
Use Exercice;  
select count(*) from contrat;
```

```
Use Exercice;  
select count(*) from regions;
```

Result Grid
count(*)
30335

Result Grid
count(*)
38916

Nos données étant maintenant correctement chargées nous allons maintenant passer à l'analyse de données selon les besoins établis.