I. L’exploration des types de données avec la capture d’écran du dictionnaire des données

L’exploration des types de données est une étape cruciale dans la conception d'une base de données. Elle permet de définir les caractéristiques de chaque colonne, de choisir les types de données appropriés, et d'établir des contraintes pour assurer l'intégrité des données. Voici le dictionnaire des données pour les tables "contrat" et "région" avec les informations requises :

	Nom des colonnes	Type de données	Taille	Clé	Description
CONTRAT.CSV	Contrat_ID	INT	10	Clé primaire	Id unique pour les contrats
	No_voie	INT	10		Numéro dans la voie pour l'adresse du logement assuré
	B_T_Q	CHAR	10		Indicateur éventuel de répétition pour l'adresse du logement assuré sur un caractère
	Type_de_voie	VARCHAR	10		Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ...
	Voie	VARCHAR	30		Libellé de la voie pour l'adresse du logement assuré
	Code_dep_code_commune	INT	10	Clé secondaire	Concaténation du code département et code commune pour avoir une clé unique
	Code_postal	INT	10		Code postal pour l'adresse du logement assuré
	Commune	VARCHAR	30		Libellé de la commune de l'adresse du logement
	Code_departement	INT	10		Libellé de la code officiel géographique de l'addresse du logement
	Surface	INT	10		Les dimentions de l'espace de vie du logement
	Type_local	VARCHAR	20		Typel de local (ex: appartement, maison...)
	Occupation	VARCHAR	20		Occupation du logement (ex: propriétaire, locataire, etc...)
	Type_contrat	VARCHAR	20		Type de contrats d'assurance
	Formule	VARCHAR	10		Formule d'assurance souscrite
	Valeur_declaree_biens	INT	10		Valeur décalrée des biens assurés
	Prix_cotisation_mensuel	INT	10		Prix de la cotisation mensuelle
REGION.CSV	Code_dep_code_commune	INT	10	Clé primaire	Concaténation du code département et code commune pour avoir une clé unique
	reg_code	INT	10		Code de le région
	reg_nom	VARCHAR	30		Nom de la région
	aca_nom	VARCHAR	30		Nom de l'académie
	dep_nom	VARCHAR	50		Nom du département
	com_nom_maj_court	VARCHAR	40		Nom court en majuscules de la commune
	dep_code	INT	10		Code du département
	dep_nom_num	VARCHAR	50		Nom numérique du département

Table "contrat" :

La table "contrat" comprend l'ensemble des colonnes provenant du fichier CSV associé aux contrats. Elle est structurée pour stocker des informations spécifiques liées aux contrats d'assurance.

Table "région" :

La table "région" contient toutes les colonnes du fichier CSV associé aux régions. Elle est conçue pour stocker des informations géographiques et administratives relatives aux différentes régions.

Dans ce dictionnaire, j'ai ajouté des indications sur les clés primaires et secondaires, ainsi que des descriptions pour chaque colonne afin de faciliter la compréhension des données. Les types de données et les tailles ont également été spécifiés en fonction des informations fournies. J'ai incorporé des contraintes dans le dictionnaire des données lorsque nécessaire : la clé primaire ne peut être vide.

II. Le schéma relationnel modifié de la base de données respectant la norme 3NF

Pour créer un schéma relationnel modifié respectant la norme 3NF à partir des informations fournies, je vais définir deux tables distinctes : "Contrat" et "Region". J'utiliserai les colonnes pertinentes pour chaque table et veillerai à respecter les principes de la 3NF, notamment l'atomicité des données et les dépendances des données.

Dans ce schéma, la table "Contrat" est définie avec la colonne "Code_dep_code_commune" en tant que clé étrangère (FK) reliant la table "Region". Ainsi, la relation entre les deux tables est établie. La clé primaire de la table "Contrat" est "Contrat_ID".

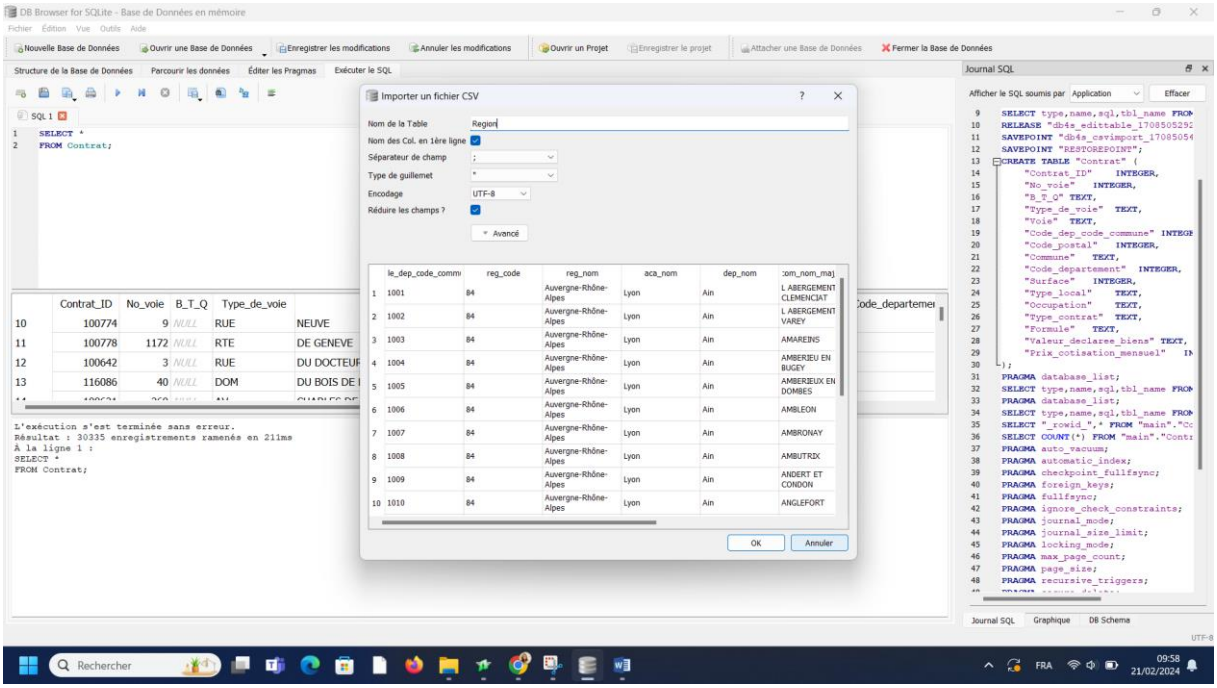
En respectant la 3NF, j'ai évité les redondances de données en séparant les informations liées aux contrats et aux régions dans des tables distinctes. Les colonnes sont définies avec des types de données appropriés, et la clé étrangère assure l'intégrité référentielle entre les deux tables.

J'ai utilisé ce schéma pour créer ce schéma de base de données dans MySQL Power Architect.

III. Le code SQL générant les tables :

La création des tables dans la base de données a été effectuée en important les données depuis deux fichiers CSV distincts. Pour importer des données depuis des fichiers CSV dans DB Browser for SQLite, je suivi ces étapes :

1. Ouvrir DB Browser for SQLite : Lancez l'application DB Browser for SQLite.
2. Créer ou Ouvrir une Base de Données : Créez une nouvelle base de données ou ouvrez une base de données existante dans laquelle vous souhaitez importer les données CSV.
3. Accéder à l'onglet "Fichier" : Allez dans l'onglet "Fichier" dans la barre de menu en haut de la fenêtre.
4. Choisir "Importer" : Dans le menu déroulant de l'onglet "Fichier", sélectionnez l'option "Importer".
5. Sélectionner le Fichier CSV : Choisissez le fichier CSV que vous souhaitez importer dans la base de données. Assurez-vous que le fichier est bien formaté et que les colonnes sont correctement délimitées.
6. Configurer les Paramètres d'Importation : DB Browser for SQLite vous demandera de configurer les paramètres d'importation. Cela inclut généralement la sélection de la table de destination, la configuration du délimiteur de colonnes, et d'autres options spécifiques à votre fichier CSV .
7. Configurer les Colonnes : Vérifiez et configurez les colonnes pour vous assurer que chaque champ est correctement mappé avec la colonne correspondante dans la base de données.
8. Lancer l'Importation : Une fois que tous les paramètres sont configurés, lancez le processus d'importation.
9. Vérifier les Résultats : Après l'importation, vérifiez la base de données pour vous assurer que les données ont été ajoutées correctement.



Lorsque j'ai importé des données depuis un fichier CSV dans DB Browser for SQLite, les tables ne sont pas créées à l'aide d'un code SQL que vous générez manuellement. Au lieu de cela, DB Browser for SQLite génère automatiquement le code SQL nécessaire pour créer les tables en fonction des colonnes et des données dans le fichier CSV.

Code génère table Contrat	**Code génère table Region**
CREATE TABLE "Contrat" ("Contrat_ID" INTEGER, "No_voie" INTEGER, "B_T_Q" TEXT, "Type_de_voie" TEXT, "Voie" TEXT, "Code_dep_code_commune" INTEGER, "Code_postal"INTEGER, "Commune" TEXT, "Code_departement"INTEGER, "Surface" INTEGER, "Type_local" TEXT, "Occupation" TEXT, "Type_contrat" TEXT, "Formule" TEXT, "Valeur_declaree_biens" TEXT, "Prix_cotisation_mensuel" INTEGER);	CREATE TABLE "Region" ("Code_dep_code_commune" INTEGER, "reg_code" INTEGER, "reg_nom" TEXT, "aca_nom" TEXT, "dep_nom" TEXT, "com_nom_maj_court" TEXT, "dep_code" INTEGER, "dep_nom_num" TEXT);

IV. La capture d’écran de votre base de données chargée

La base de données comprend deux tables principales : "Contrat" avec 30,335 lignes et "Région" avec 38,916 lignes.