

Méthodologie projet 3

Requêtez une base de données avec SQL

A hand is pointing towards a central circular graphic that contains the text 'SQL'. The background is dark blue with a network of glowing nodes and lines, suggesting a database or project management theme.

SQL

Étape 1 - Découvrez les différents types de données

La méthode

1. Consultation des différents type de données référencées sur le site [Microsoft](#)
2. Renseignement du dictionnaire des données en accord avec les fichiers CSV fournis
3. Choix du type de données
4. Choix des valeurs qui peuvent être nulles ou non
5. Choix des contraintes en s'appuyant sur un filtrage des colonnes et l'ajout de colonnes comptant le nombre de caractères de chaque lignes.
6. Les clés étaient déjà renseignées, elles me convenaient, je n'ai pas jugé nécessaire de les modifier
7. Renseignement de la colonne 'Description' en m'appuyant sur le filtrage que j'avais effectué et sur ce [document](#) trouvé sur la page du gouvernement qui fournit le jeu de données.

Ce que j'ai appris

1. Ce qu'était un charachter et un varchar et leur différence, même si ce n'est pas très clair (codage en bit?)
2. Je m'interroge sur le nombre de caractère à choisir pour les variables dont la longueur n'est pas fixe
3. Je m'interroge sur la pertinence de typer Contrat_id en integer, pour les autres codes également
4. En ayant suivi cette logique, je me suis trouvé confronté à un problème sur la colonne Code_dep_code_commune. J'ai du modifié le type après chargement alors que SQLite m'a autorisé à charger la table avec cette colonne typée en Integer alors qu'elle comportait des lignes 2A et 2B.... Je ne les avais pas remarquée lors de l'exploration des données.

Étape 1 - Le dictionnaire des données

	Nom des colonnes	Type de données	Taille	Clé	Description
CONTRAT.CSV	Contrat_ID	INT NOT NULL	6	Clé primaire	Id unique pour les contrats
	No_voie	INT	4		Numéro dans la voie pour l'adresse du logement assuré
	B_T_Q	CHAR	1		Indicateur éventuel de répétition pour l'adresse du logement assuré sur un caractère
	Type_de_voie	VARCHAR	50		Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ...
	Voie	VARCHAR	255		Libellé de la voie pour l'adresse du logement assuré
	Code_dep_code_commune	VARCHAR NOT NULL	6	Clé secondaire	Concaténation du code département et code commune pour avoir une clé unique
	Code_postal	INT	5		Code postal pour l'adresse du logement assuré
	Surface	INT NOT NULL	3		Surface du logement assuré en M2
	Type_local	VARCHAR NOT NULL	20		Type de local. Peut prendre deux valeurs: appartement ou maison
	Occupation	VARCHAR NOT NULL	20		Type d'occupation. Peut prendre deux valeurs: locataire ou propriétaire.
	Type_contrat	VARCHAR NOT NULL	50		Type de contrat. Peut prendre trois valeurs: mise en location, résidence principale, résidence secondaire.
	Formule	VARCHAR NOT NULL	20		Formule du contrat. Peut prendre deux valeurs: classique ou intégral
	Valeur_declaree_biens	VARCHAR NOT NULL	20		Valeur déclarée du bien assuré. Peut prendre 4 valeurs: 0-25000, 25000-50000, 50000-100000, 100000+
	Prix_cotisation_mensuel	INT NOT NULL	3		Montant de la cotisation mensuelle en €
REGION.CSV	Code_dep_code_commune	INT NOT NULL	6	Clé primaire	Concaténation du code département et code commune pour avoir une clé unique
	Reg_code	INT NOT NULL	2		Code de la région sur laquelle se trouve la commune
	Reg_nom	VARCHAR NOT NULL	50		Libellé de la région sur laquelle se trouve la commune
	Aca_nom	VARCHAR	50		Libellé de l'académie à laquelle est rattachée la commune
	Dep_nom	VARCHAR NOT NULL	50		Libellé du département sur lequel se trouve la commune
	Com_nom_maj_court	VARCHAR NOT NULL	50		Libellé court de la commune en majuscule
	Dep_code	VARCHAR NOT NULL	3		Code du département sur lequel se trouve la commune
	Dep_nom_num	VARCHAR	50		Libellé et code du département sur lequel se trouve la commune.

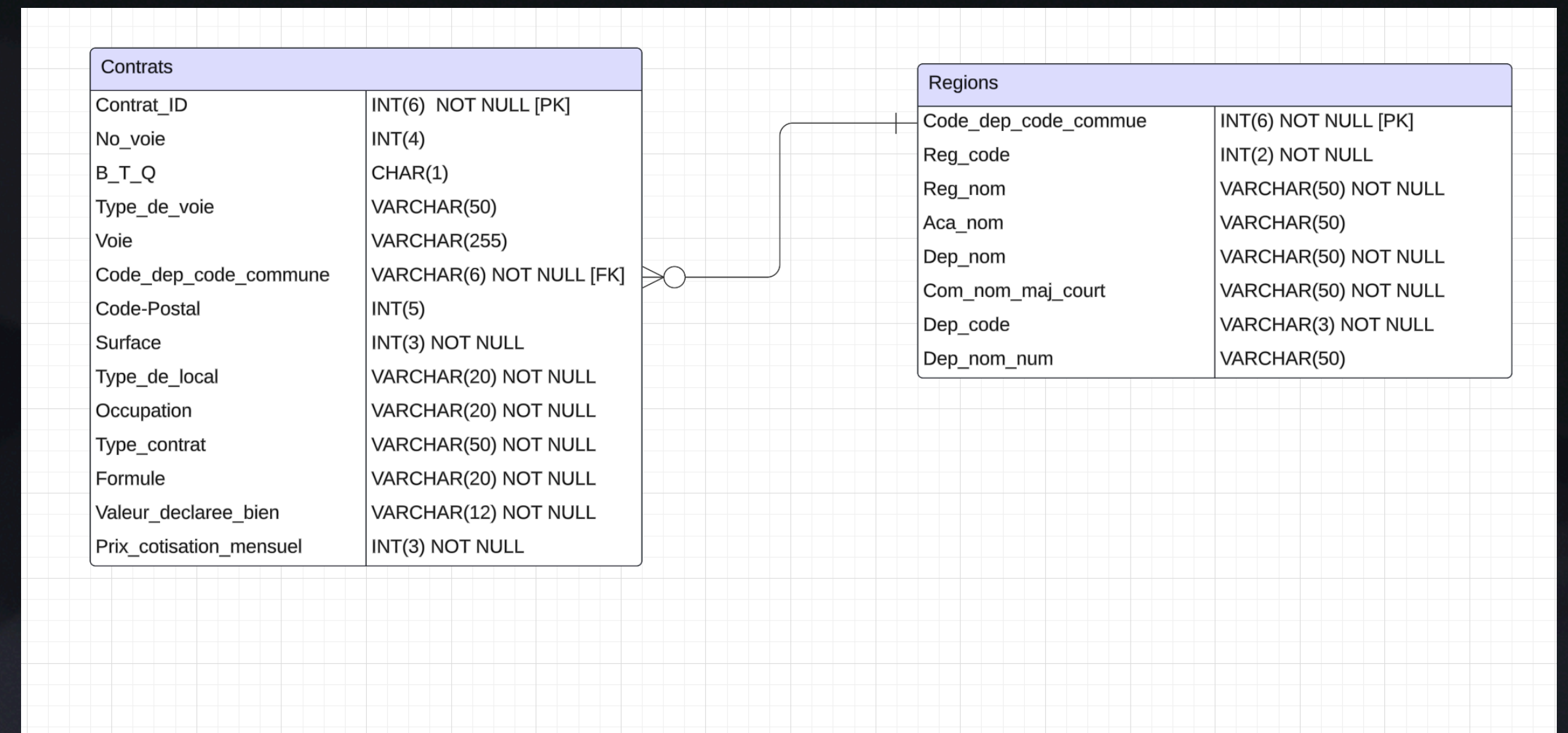
Étape 2- Découvrez la conception du schéma relationnel

La méthode

1. Suivi et résumé (cheat sheet) du cours 'Modélisez vos bases de données en entier'
2. Recherche d'un outil capable de faire le schéma et de générer le code SQL, SQL Power Architect n'a jamais voulu se lancer sur ma machine....
3. Prise en main de l'outil
4. Recopie des lignes renseignées dans le dictionnaire des données sur la plateforme pour la création du schéma et des deux tables
5. Export du schéma et du code associé

Ce que j'ai appris

1. Comment modéliser une table
2. Utiliser Vertabelo



Étape 3- Découvrez la création et le chargement de base de données

La méthode

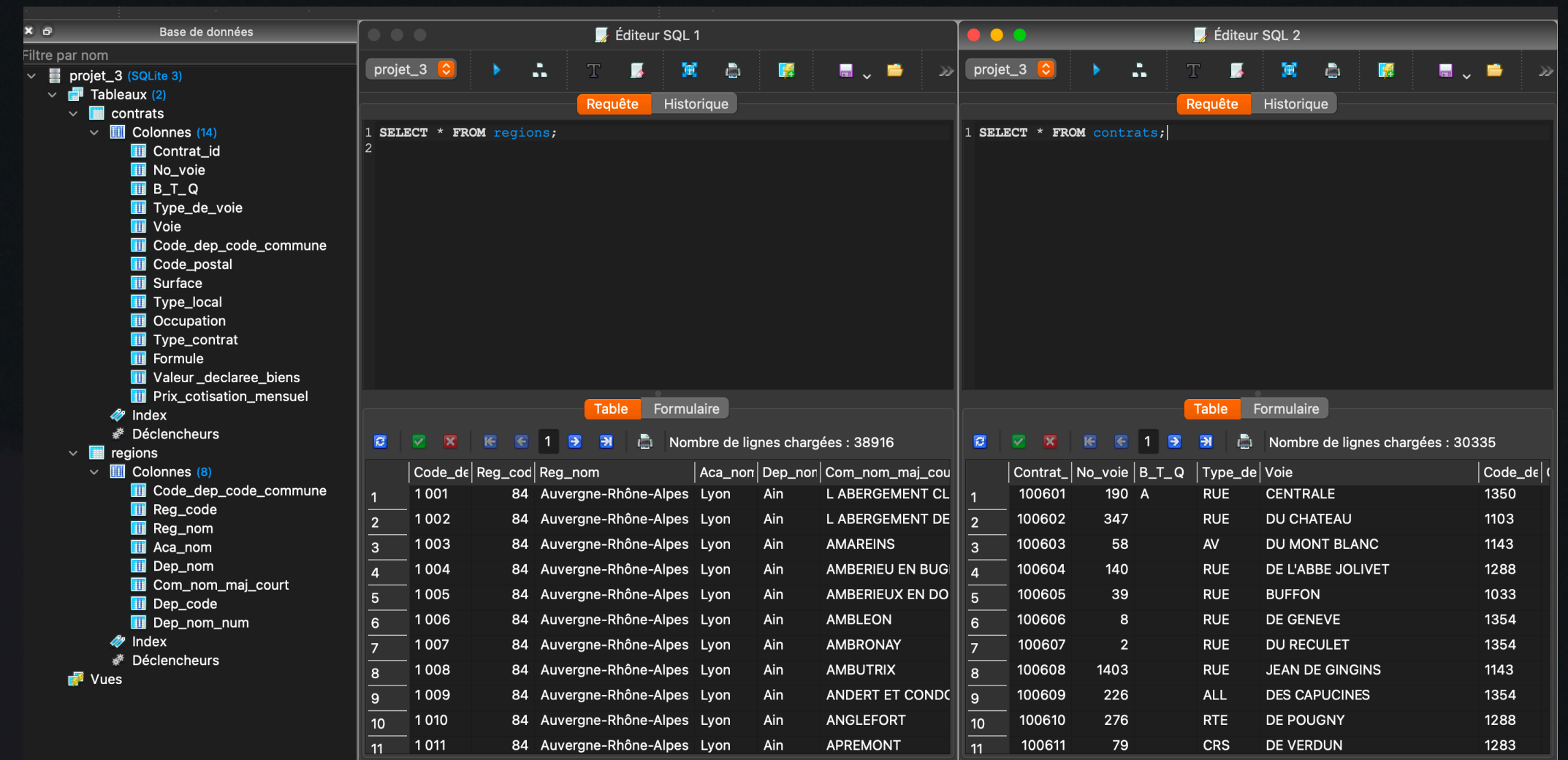
1. Recherche sur les commandes SQL nécessaires pour répondre aux questions
2. Création de la base de données en ligne de commande
3. Je n'ai pas réussi à charger les fichiers .csv dans la base de données en ligne de commandes. Je ne suis pas à l'aise avec le terminal
4. Je les ai chargées avec SQLite Studio. Peu de difficultés pour la table contrats, mais de grosses difficultés pour la table régions à cause du séparateur de milliers
5. Avant d'en arriver au séparateur de milliers, j'ai ausculté le fichier .csv à la loupe. Je l'ai transformé en .xlsx pour utiliser les filtres et mieux voir les données des colonnes.
6. J'ai pris la décision de supprimer toute la partie de la Corse qui m'obligeait à typer cette colonne en varchar alors que je l'avais typée en integer dans la première table. Je craignais un conflit.
7. J'ai trouvé deux cellules dans Com_nom_maj_court qui devait considérée comme booléen (faux). J'ai trouvé leur véritable valeur sur internet.
8. Pour m'assurer d'avoir le bon type de données, j'ai usé des CNUM et CTXT

JF Dubroca

Étape 3- Découvrez la création et le chargement de base de données

Ce que j'ai appris

1. Que SQL s'exécutait une ligne de commande depuis le terminal
2. À créer une base de données
3. Que SQLite Studio que l'on nous a fait charger pour un cours et l'interface graphique de SQLite
4. Que le séparateur de milliers que j'aime beaucoup peut-être une véritable plaie!



Étape 4 et 5 - Découvrez la rédaction des requêtes SQL

La méthode

1. J'avais déjà fait le cours 'Requêtez une base de données avec SQL' et une cheat sheet.
2. Je n'hésite pas à essayer des requêtes en dehors de celles demandées
3. Je cherche à comprendre les erreurs et les messages d'erreur
4. J'ai eu recours à ChatGPT pour comprendre pourquoi certaines requêtes ne fonctionnent pas, mais toujours après avoir cherché la solution par moi même.
5. J'ai refait plusieurs fois les requêtes, avec WHERE et avec une jointure. J'aime bien la jointure, que je trouve finalement plus simple.

Ce que j'ai appris

1. Les requêtes SQL et leurs structures
2. À essayer d'organiser mon code de manière à ce qu'il soit le plus lisible possible
3. À comprendre la logique des requêtes même si c'est encore un point à développer.
4. Qu'après une fonction d'agrégation il faut mettre un GROUP BY, après le WHERE et avant le HAVING, même si SQLite permet quelques fois de ne pas l'utiliser.

Étape 4 et 5 - Quelques requêtes

Toutes les requêtes sont disponibles sur un document avec les captures d'écran

```
SELECT
  Contrat_id,
  Com_nom_maj_court AS Commune
FROM
  contrats
LEFT JOIN regions
  ON regions.Code_dep_code_commune = contrats.Code_dep_code_commune
WHERE Com_nom_maj_court LIKE 'CAEN'
;
```

Cette requête joint les tables contrats et régions en faisant correspondre les colonnes Code_dep_code_commune de chaque table. Elle retournera les colonnes Contrat_id et Com_nom_maj_court renommée en Commune pour les lignes dans lesquelles CAEN est inscrit dans la colonne Com_nom_maj_court

Étape 4 et 5 - Quelques requêtes

```
SELECT
  Dep_nom as Departement,
  ROUND(AVG(Prix_cotisation_mensuel),2) AS Prix_moyen
FROM regions
LEFT JOIN contrats
  ON regions.Code_dep_code_commune =
  contrats.Code_dep_code_commune
GROUP BY Departement
ORDER BY Prix_moyen DESC
LIMIT 10
;
```

Cette requête joint les tables contrats et régions en faisant correspondre les colonnes Code_dep_code_commune de chaque table. Elle retournera les 10 premiers résultats des colonnes Dep_nom nommée Département et la moyenne du prix de la cotisation arrondi à 2 décimales pour chaque département, nommée Prix_moyen classé par ordre décroissant

```
SELECT
  COUNT (Contrat_id) as Nombre_contrat,
  Valeur_declaree_biens
FROM
  contrats
GROUP BY Valeur_declaree_biens
ORDER BY Nombre_contrat DESC
;
```

Cette requête retourne les colonnes valeur déclarée des biens et le nombre de contrat pour chacune de ces valeurs, nommée Nombre_contrat