

27/12/2024

Documentation technique de la base de données « Requêtez une base de données avec SQL »



Antoine Vatin

Table des matières

Table des illustrations	2
Création du dictionnaire de données	3
Mise en place d'un schéma relationnel normalisé	4
Relations entre les tables et rôle des clés étrangères	5
Code SQL de création des tables	6
La base de données	7

Table des illustrations

<i>Figure 1 Dictionnaire de données</i>	3
<i>Figure 2 Schéma relationnel</i>	4
<i>Figure 3 Vue de la table Région dans pgAdmin</i>	7
<i>Figure 4 Vue de la table Contrat dans pgAdmin</i>	7

Création du dictionnaire de données

Un **dictionnaire de données** est un document ou un outil qui décrit et structure les informations sur les données d'un système. Il détaille les attributs, types, relations, et règles associées à chaque élément de données (tables, champs, etc.) dans une base de données ou un projet.

Un dictionnaire de données a été créé pour documenter les tables, colonnes, et relations de notre base de données. Voici une capture d'écran représentant les attributs des tables principales :

	Nom des colonnes	Type de données	Taille	Clé	Description
CONTRAT.CSV	Contrat_ID	INT		Clé primaire	Id unique pour les contrats
	No_voie	INT			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	CVARCHAR			Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ...
	Voie	CVARCHAR			Libellé de la voie pour l'adresse du logement assuré
	Code_dep_code_commune	CVARCHAR		Clé secondaire	Concaténation du code département et code commune pour avoir une clé unique
	Code_postal	INT			Code postal pour l'adresse du logement assuré
	Surface	INT			Surface du logement assuré
	Type_local	CVARCHAR			Type de logement (Maison, Appartement)
	Occupation	CVARCHAR			Statut des habitants (Locataire, Propriétaire)
	Type_contrat	CVARCHAR			Type de contrat du logement (Résidence principale, secondaire ou mise en location)
	Formule	CVARCHAR			Formule d'assurance (Classique ou Intégral)
	Valeur_declaree_biens	CVARCHAR			Valorisation des biens assurés
	Prix_cotisation_mensuel	NUMERIC			Prix de la cotisation d'assurance du client
REGION.CSV	Code_dep_code_commune	CVARCHAR		Clé primaire	Concaténation du code département et code commune pour avoir une clé unique
	reg_code	INT			Code de la région
	reg_nom	CVARCHAR			Nom de la région
	aca_nom	CVARCHAR			Nom de la ville dépendant l'académie
	dep_nom	CVARCHAR			Nom du département
	com_nom_maj_court	CVARCHAR			Nom complet de la commune
	dep_code	CVARCHAR			Code du département
	dep_nom_num	CVARCHAR			Concaténation du nom du département et code département

Figure 1 Dictionnaire de données

Fonctions principales :

1. Documentation : Fournit une référence claire sur la structure et la signification des données.
2. Normes : Assure la cohérence et standardisation des données.
3. Communication : Facilite la compréhension entre les équipes (développeurs, analystes, décideurs).
4. Support au développement : Aide à concevoir, modifier, ou auditer une base de données ou un système d'information.
5. Traçabilité : Documente les origines et transformations des données pour les audits ou analyses.

Mise en place d'un schéma relationnel normalisé

Outil utilisé : SQL Power Architects

Un **schéma relationnel** est une représentation graphique ou textuelle de la structure d'une base de données relationnelle. Il décrit les **tables** (relations), leurs **champs** (attributs), et les **liens** (clés primaires et étrangères) entre elles.

Nous avons conçu un schéma relationnel basé sur les principes de la 3ème forme normale (3NF). Voici les caractéristiques principales :

- **Clés primaires** : Chaque table possède un identifiant unique.
- **Clés étrangères** : Les relations entre les tables sont assurées par des clés étrangères (par ex. Code_dep_code_commune).

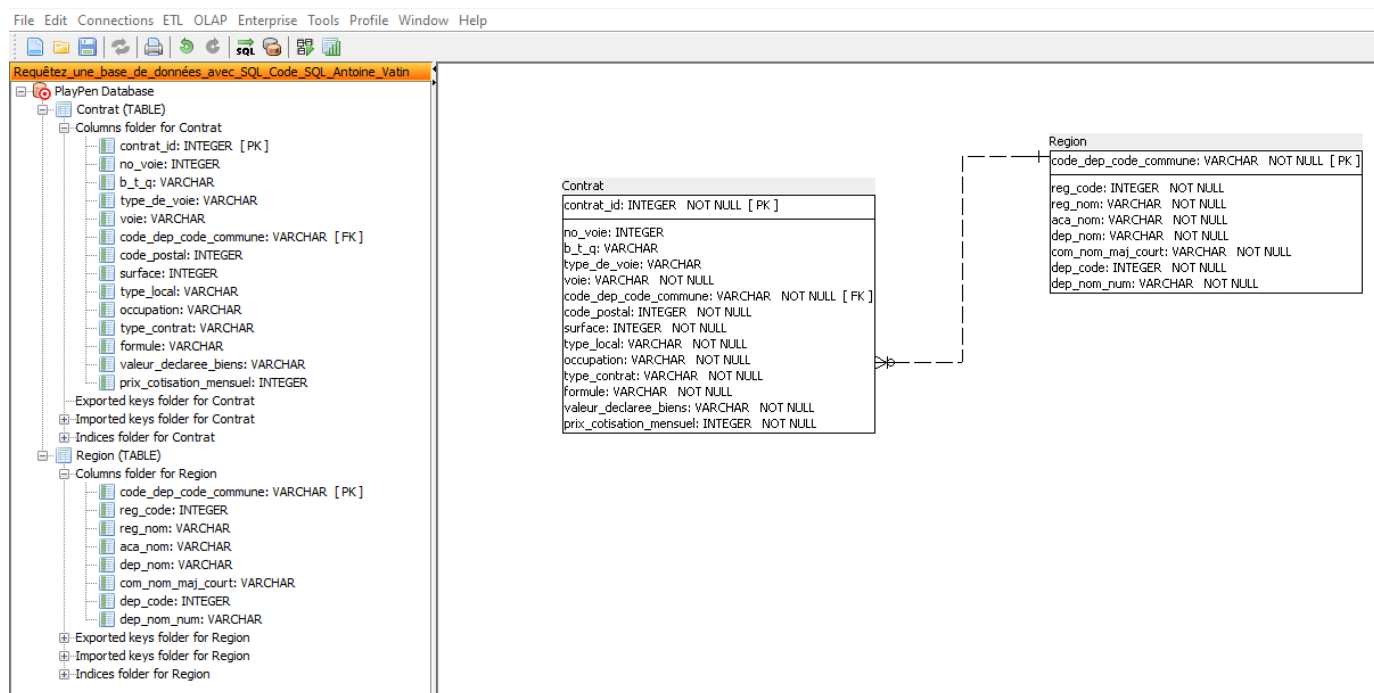


Figure 2 Schéma relationnel

Fonctions principales :

1. **Modélisation** : Représente la structure logique des données et leurs relations.
2. **Conception** : Sert de base pour créer la base de données physique.
3. **Visualisation** : Facilite la compréhension des interactions entre tables.
4. **Documentation** : Guide pour les développeurs, analystes, et administrateurs.
5. **Optimisation** : Aide à identifier les relations mal définies ou les redondances dans la structure.

Relations entre les tables et rôle des clés étrangères

La structure de la base de données repose sur deux tables principales : **Region** et **Contrat**, reliées par une relation clé étrangère.

1. Relation entre Region et Contrat :

- La table Contrat inclut une colonne appelée Code_dep_code_commune, qui agit comme une **clé étrangère** pointant vers la clé primaire de la table Region (également appelée Code_dep_code_commune).
- Cette relation permet de lier chaque contrat à une commune spécifique, contenue dans la table Region.

2. Objectifs des relations :

- **Éviter les redondances** : Les informations sur les communes (nom, code postal, etc.) sont stockées uniquement dans la table Region, tandis que la table Contrat se concentre sur les informations spécifiques aux contrats.
- **Assurer l'intégrité référentielle** : La clé étrangère garantit qu'aucun contrat ne peut exister sans correspondre à une région valide.
- **Faciliter les jointures** : Grâce à cette relation, des analyses combinant données géographiques et informations contractuelles peuvent être réalisées.

3. Avantages de cette modélisation relationnelle :

- **Clarté** : La séparation des tables permet de structurer les données de manière logique.
- **Performance** : Les requêtes exploitent des clés indexées, ce qui améliore leur rapidité.
- **Flexibilité** : Il est facile d'ajouter ou de modifier des données dans l'une des tables sans impact sur l'autre, tant que les contraintes sont respectées.

Code SQL de création des tables

Les requêtes SQL présentées dans cette section ont été générées à partir des spécifications définies dans le dictionnaire de données. Ce dernier a permis de structurer les attributs des tables, leurs types de données, et les relations entre elles, garantissant ainsi une correspondance rigoureuse avec le modèle conceptuel. Les contraintes définies, telles que les clés primaires et étrangères, assurent l'intégrité référentielle et la cohérence de la base de données. Ces requêtes respectent également les bonnes pratiques en matière de normalisation et de structuration relationnelle.

```
CREATE TABLE public.region (  
    code_dep_code_commune VARCHAR NOT NULL,  
    reg_code INTEGER NOT NULL,  
    reg_nom VARCHAR NOT NULL,  
    aca_nom VARCHAR NOT NULL,  
    dep_nom VARCHAR NOT NULL,  
    com_nom_maj_court VARCHAR NOT NULL,  
    dep_code INTEGER NOT NULL,  
    dep_nom_num VARCHAR NOT NULL,  
    CONSTRAINT region_pk PRIMARY KEY (code_dep_code_commune)  
)  
CREATE TABLE public.contrat (  
    contrat_id INTEGER NOT NULL,  
    no_voie INTEGER,  
    b_t_q VARCHAR,  
    type_de_voie VARCHAR, -- Accepte les valeurs NULL  
    voie VARCHAR NOT NULL,  
    code_dep_code_commune VARCHAR NOT NULL,  
    code_postal INTEGER NOT NULL,  
    surface INTEGER NOT NULL,  
    type_local VARCHAR NOT NULL,  
    occupation VARCHAR NOT NULL,  
    type_contrat VARCHAR NOT NULL,  
    formule VARCHAR NOT NULL,  
    valeur_declaree_biens INTEGER NOT NULL,  
    prix_cotisation_mensuel INTEGER NOT NULL,  
    CONSTRAINT contrat_pk PRIMARY KEY (contrat_id)  
)  
ALTER TABLE public.contrat ADD CONSTRAINT region_contrat_fk  
FOREIGN KEY (code_dep_code_commune)  
REFERENCES public.region (code_dep_code_commune);
```

La base de données

La base de données a été implémentée à l'aide de l'outil **pgAdmin**. Les captures d'écran des tables Région et Contrat illustrent leur structure, conformes aux spécifications du dictionnaire de données et au schéma relationnel. Ces tables ont été conçues pour optimiser les performances des requêtes et assurer la flexibilité nécessaire à l'évolution future du système. Les relations définies entre les tables facilitent les analyses croisées et répondent aux besoins fonctionnels identifiés.

Figure 3 shows the 'region' table in pgAdmin. The table structure is as follows:

code_dep_code_commune	reg_code	reg_nom	aca_nom	dep_nom	com_nom_maj_court	dep_code	dep_nom_num
1001	84	Auvergne-Rhône-Alpes	Lyon	Ain	L ABERGEMENT CLEMENCIAI	1	Ain (01)
1002	84	Auvergne-Rhône-Alpes	Lyon	Ain	L ABERGEMENT DE VAREY	1	Ain (01)
1003	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMARENS	1	Ain (01)
1004	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBERIEUX EN BUGEY	1	Ain (01)
1005	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBERIEUX EN DOMBES	1	Ain (01)
1006	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBLEON	1	Ain (01)
1007	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBRONAY	1	Ain (01)
1008	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBUTRIX	1	Ain (01)
1009	84	Auvergne-Rhône-Alpes	Lyon	Ain	ANDERT ET CONDON	1	Ain (01)
1010	84	Auvergne-Rhône-Alpes	Lyon	Ain	ANGLEFORT	1	Ain (01)
1011	84	Auvergne-Rhône-Alpes	Lyon	Ain	APREMONT	1	Ain (01)
1012	84	Auvergne-Rhône-Alpes	Lyon	Ain	ARANC	1	Ain (01)
1013	84	Auvergne-Rhône-Alpes	Lyon	Ain	ARANDAS	1	Ain (01)
1014	84	Auvergne-Rhône-Alpes	Lyon	Ain	AREBENT	1	Ain (01)
1015	84	Auvergne-Rhône-Alpes	Lyon	Ain	ARBIGNIEUX	1	Ain (01)
1016	84	Auvergne-Rhône-Alpes	Lyon	Ain	ARBIGNY	1	Ain (01)
1017	84	Auvergne-Rhône-Alpes	Lyon	Ain	ARGIS	1	Ain (01)

Figure 3 Vue de la table Région dans pgAdmin

Figure 4 shows the 'contrat' table in pgAdmin. The table structure is as follows:

contrat_id	no_voie	b.t.q	type_de_voie	voie	code_dep_code_commune	code_postal	surface	type_local	occupation	type_contrat	formule
100001	190	A	RUE	CENTRALE	1350	1370	50	Appartement	Locataire	Residence principale	Integral
100002	347	[null]	RUE	DU CHATEAU	1103	1170	48	Appartement	Locataire	Residence principale	Classique
100003	58	[null]	AV	DU MONT BLANC	1143	1220	131	Appartement	Propriétaire	Residence principale	Integral
100004	140	[null]	RUE	DE CABBÉ JOLIVET	1288	1630	109	Maison	Locataire	Residence principale	Integral
100005	39	[null]	RUE	BUFFON	1093	1200	109	Appartement	Locataire	Residence principale	Classique
100006	8	[null]	RUE	DE GENÈVE	1354	1630	53	Appartement	Propriétaire	Residence principale	Classique
100007	2	[null]	RUE	DU RECOULET	1354	1630	59	Appartement	Propriétaire	Residence principale	Integral
100008	1403	[null]	RUE	JEAN DE BRINGS	1143	1220	93	Maison	Propriétaire	Mise en location	Integral
100009	226	[null]	ALL	DES CAPUCINES	1354	1630	117	Maison	Propriétaire	Residence principale	Classique
100010	276	[null]	RTE	DE POUJOY	1288	1630	36	Appartement	Propriétaire	Residence principale	Integral
100011	79	[null]	GRD	DE VERDUN	1283	1100	138	Appartement	Propriétaire	Residence principale	Classique
100012	240	[null]	RUE	DE PRE DALLY	1173	1170	45	Appartement	Locataire	Residence principale	Classique
100013	3	[null]	RUE	TURNÈRE	1033	1200	83	Appartement	Locataire	Residence principale	Classique
100014	44	[null]	ALL	DU SQUARE DE LAUSANNE	1143	1220	88	Appartement	Locataire	Residence principale	Integral
100015	59	[null]	RUE	ALEXANDRE BERNARD	1004	1500	165	Appartement	Locataire	Residence principale	Classique
100016	282	[null]	CHE	DES LONGES RAVES	1071	1170	42	Appartement	Propriétaire	Residence principale	Classique
100017	54	[null]	GR	GRANDE RUE	1396	1150	68	Appartement	Propriétaire	Residence secondai...	Classique

Figure 4 Vue de la table Contrat dans pgAdmin