

# Projet 3 -Requêtez une base de données avec SQL

## -Document Technique : Création d'une BDDR

### I. Exploration des données :

#### A. Les supports des données : fichiers CSV :

Dans ce projet les données sont fournies dans des **fichiers .CSV** , ouvrable en version texte **avec un outil Note**, ou ouvrable en version Tableur **avec Excel** (ou Google Sheets).

##### 1. Le fichier Contrat.csv :

Ouvert par Bloc-Note :

```
Contrat_ID;No_voie;B_T_Q;Type_de_voie;Voie;Code_dep_code_commune;Code_postal;Surface;Type_local;Occupation;Type_contrat;Formule;Valeur_declaree_biens;Prix_cotisation_mensuel
100601;190;A;RUE;CENTRALE;1350;1370;50;Appartement;Locataire;Residence principale;Integral;0-25000;25
100602;347;;RUE;DU CHATEAU;1103;1170;48;Appartement;Locataire;Residence principale;Classique;0-25000;30
100603;58;;AV;DU MONT BLANC;1143;1220;131;Appartement;Proprietaire;Residence principale;Integral;25000-50000;57
100604;140;;RUE;DE L'ABBE JOLIVET;1288;1630;109;Maison;Locataire;Residence principale;Integral;25000-50000;43
100605;39;;RUE;BUFFON;1033;1200;109;Appartement;Locataire;Residence principale;Classique;0-25000;33
```

La première ligne, correspond à la ligne des attributs.

Chaque attribut est séparé par un délimiteur « ; ».

Chaque ligne, à partir de la deuxième, représente un instance.

Avec ces informations, notamment du séparateur, on peut ouvrir le fichier avec Excel :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Contrat_ID	No_voie	B_T_Q	Type_de_voie	Voie	Code_dep_cc	Code_postal	Surface	Type_local	Occupation	Type_contrat	Formule	Valeur_decla	Prix_cotisation_mensuel	
2	100601	190	A	RUE	CENTRALE	1350	1370	50	Appartement	Locataire	Residence pr	Integral	0-25000	25	
3	100602	347		RUE	DU CHATEAU	1103	1170	48	Appartement	Locataire	Residence pr	Classique	0-25000	30	
4	100603	58		AV	DU MONT BL	1143	1220	131	Appartement	Proprietaire	Residence pr	Integral	25000-50000	57	
5	100604	140		RUE	DE L'ABBE JO	1288	1630	109	Maison	Locataire	Residence pr	Integral	25000-50000	43	
6	100605	39		RUE	BUFFON	1033	1200	109	Appartement	Locataire	Residence pr	Classique	0-25000	33	

On peut ainsi Filtrer sur les colonnes, et observer les différentes valeurs présentes dans chacune des colonnes :

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Contrat_ID	No_voie	B.T.Q	Type_de_v	Voie	Code_dep	Code_pos	Surface	Type_loca	Occupati	Type_cont	Formule	Valeur_de	Prix_cotis	on_mensuel
2	100601					1350	1370	50	Appartement	Locataire	Residence pr	Integral	0-25000	25	
3	100602					1103	1170	48	Appartement	Locataire	Residence pr	Classique	0-25000	30	
4	100603					1143	1220	131	Appartement	Proprietaire	Residence pr	Integral	25000-50000	57	
5	100604					1288	1630	109	Maison	Locataire	Residence pr	Integral	25000-50000	43	
6	100605					1033	1200	109	Appartement	Locataire	Residence pr	Classique	0-25000	33	
7	100606					1354	1630	53	Appartement	Proprietaire	Residence pr	Classique	0-25000	19	
8	100607					1354	1630	59	Appartement	Proprietaire	Residence pr	Integral	0-25000	15	
9	100608					1143	1220	93	Maison	Proprietaire	Mise en loca	Integral	25000-50000	34	
10	100609					1354	1630	117	Maison	Proprietaire	Residence pr	Classique	25000-50000	32	
11	100610					1288	1630	36	Appartement	Proprietaire	Residence pr	Integral	25000-50000	22	
12	100611					1283	1100	138	Appartement	Proprietaire	Residence se	Classique	0-25000	11	
13	100612					1173	1170	45	Appartement	Locataire	Residence pr	Classique	0-25000	16	
14	100613					1033	1200	83	Appartement	Locataire	Residence pr	Classique	0-25000	14	
15	100614					1143	1220	88	Appartement	Locataire	Residence pr	Integral	25000-50000	34	
16	100615					1004	1500	165	Appartement	Locataire	Residence pr	Classique	25000-50000	24	
17	100616					1071	1170	42	Appartement	Proprietaire	Residence pr	Classique	0-25000	17	
18	100617					1396	1150	68	Appartement	Proprietaire	Residence se	Classique	0-25000	10	
19	100618					1160	1210	83	Appartement	Proprietaire	Residence pr	Classique	0-25000	20	
20	100619					1103	1170	30	Appartement	Proprietaire	Residence se	Classique	0-25000	12	
21	100620					1399	1170	25	Appartement	Locataire	Residence pr	Integral	0-25000	11	
22	100621					1202	1150	53	Appartement	Locataire	Residence pr	Integral	0-25000	12	
23	100622					1173	1170	46	Appartement	Proprietaire	Residence se	Classique	0-25000	12	
24	100623					1354	1630	84	Appartement	Proprietaire	Residence pr	Classique	0-25000	20	
25	100624					1004	1500	68	Appartement	Locataire	Residence pr	Integral	0-25000	13	
26	100625	269		CHE	DE CHARIGNI	1034	1300	62	Appartement	Proprietaire	Residence pr	Classique	0-25000	11	
27	100626	222		DUE	DE BRETONN	1204	1210	112	Maison	Proprietaire	Residence pr	Integral	25000-50000	22	

## 2. Le Fichier Region.csv :

Il en est de même pour le fichier Region.csv :

 Region.csv - Bloc-notes

Fichier Edition Format Affichage Aide

```
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 CLEMENCIAT;1;Ain (01)
1002;84;Auvergne-Rhône-Alpes;Lyon;Ain;L ABERGEMENT DE VAREY;1;Ain (01)
```

	A	B	C	D	E	F	G	H	I
1	Code_dep_cc	reg_code	reg_nom	aca_nom	dep_nom	com_nom_m	dep_code	dep_nom_num	
2	1001	84	Auvergne-Rh	Lyon	Ain	LABERGEME	1	Ain (01)	
3	1002	84	Auvergne-Rh	Lyon	Ain	LABERGEME	1	Ain (01)	

	A	B	C	D	E	F	G	H	
1	Code_dep	reg_code	reg_nom	aca_nom	dep_nom	com_nom	dep_code	dep_nom	m
2	1001	84	Auvergne-Rh	Lyon				Ain (01)	
3	1002	84	Auvergne-Rh	Lyon				Ain (01)	
4	1003	84	Auvergne-Rh	Lyon				Ain (01)	
5	1004	84	Auvergne-Rh	Lyon				Ain (01)	
6	1005	84	Auvergne-Rh	Lyon				Ain (01)	
7	1006	84	Auvergne-Rh	Lyon				Ain (01)	
8	1007	84	Auvergne-Rh	Lyon				Ain (01)	
9	1008	84	Auvergne-Rh	Lyon				Ain (01)	
10	1009	84	Auvergne-Rh	Lyon				Ain (01)	
11	1010	84	Auvergne-Rh	Lyon				Ain (01)	
12	1011	84	Auvergne-Rh	Lyon				Ain (01)	
13	1012	84	Auvergne-Rh	Lyon				Ain (01)	
14	1013	84	Auvergne-Rh	Lyon				Ain (01)	
15	1014	84	Auvergne-Rh	Lyon				Ain (01)	
16	1015	84	Auvergne-Rh	Lyon				Ain (01)	
17	1016	84	Auvergne-Rh	Lyon				Ain (01)	
18	1017	84	Auvergne-Rh	Lyon				Ain (01)	
19	1018	84	Auvergne-Rh	Lyon				Ain (01)	
20	1019	84	Auvergne-Rh	Lyon				Ain (01)	
21	1020	84	Auvergne-Rh	Lyon				Ain (01)	
22	1021	84	Auvergne-Rh	Lyon				Ain (01)	
23	1022	84	Auvergne-Rh	Lyon				Ain (01)	
24	1023	84	Auvergne-Rh	Lyon				Ain (01)	

A Z ↓ Trier du plus petit au plus grand  
 Z A ↓ Trier du plus grand au plus petit  
 Trier par couleur >  
 Affichage du tableau >  
 Effacer le filtre de « dep\_code »  
 Filtrer par couleur >  
 Filtres numériques >  
 Rechercher  
☒ 977  
☒ 978  
☒ 984  
☒ 986  
☒ 987  
☒ 988  
☒ 989  
☒ 2A  
☒ 2B  
 OK Annuler

On peut constater par exemple que l'attribut dep\_code contient à la fois des nombres à 1 chiffre, à 2 chiffres, à 3 chiffres, voire inclure des lettres.

## B. Le dictionnaire des données :

On peut donc **lister les données (attributs)** dont on dispose pour notre étude pour chacune des entités Region et Contrat (qui se dégagent), et **voir le format (le type)** de ces données. On les rassemble alors dans ce qu'on appelle **le dictionnaire des données** suivant :

	Nom des colonnes	Type de données	Taille	Clé	Contraintes	Description
CONTRAT.CSV	Contrat_ID	INT	6	Clé primaire	NOT NULL, UNIQUE	Id unique pour les contrats
	No_voie	INT	5			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	200		NOT NULL	Libellé de la voie pour l'adresse du logement assuré
	Code_dep_code_commune	VARCHAR	6	Clé étrangère	NOT NULL	Concaténation du code département et code commune pour avoir une clé unique (2A , 971... possible)
	Code_postal	INT	5		NOT NULL	Code postal pour l'adresse du logement assuré (pas de 2A possible transformé en 20)
	Surface	INT	5			Surface du bien concerné par ce contrat
	Type_local	VARCHAR	50			Type de bien (Appartement, Maison...)
	Occupation	VARCHAR	50			Statut de l'occupant du titulaire du contrat : Propriétaire ou locataire
	Type_contrat	VARCHAR	50			Type de contrat (Rés principale, Res secondaire, Mise en location)
	Formule	VARCHAR	50			Formule du contrat choisie (Integral, Classique )
	Valeur_declaree_biens	VARCHAR	50			Fourchette correspondant à la valeur déclarée des biens couverts
	Prix_cotisation_mensuel	FLOAT, NUMERIC(5,2)	7			Prix de la cotisation
REGION.CSV	Code_dep_code_commune	VARCHAR	6	Clé primaire	NOT NULL, UNIQUE	Concaténation du code département et code commune pour avoir une clé unique (2A, 971... possible)
	reg_code	INT	2		NOT NULL	Code à 2 chiffres du département (même Corse 2A c'est 94, et 978 c'est 00)
	reg_nom	VARCHAR	100		NOT NULL	Nom de la région
	aca_nom	VARCHAR	100		NOT NULL	Nom de l'académie
	dep_nom	VARCHAR	100		NOT NULL	Nom du département
	com_nom_maj_court	VARCHAR	100		NOT NULL	Nom de la commune
	dep_code	VARCHAR	3		NOT NULL	Code à 2 ou 3 caractères du département (ex 2A, 978, 75,...)
	dep_nom_num	VARCHAR	100		NOT NULL	Nom du département associé à son code



## II. Modélisation des données :

Afin de pouvoir créer une base de données fonctionnelle, il faut voir comment nous pouvons organiser les données.

Ici nous avons des instances qui ont **des données bien structurées**, et dont les attributs sont bien renseignés.

(Nous ne faisons pas face à un ensemble de données peu structurées, et avec beaucoup d'attributs vides).

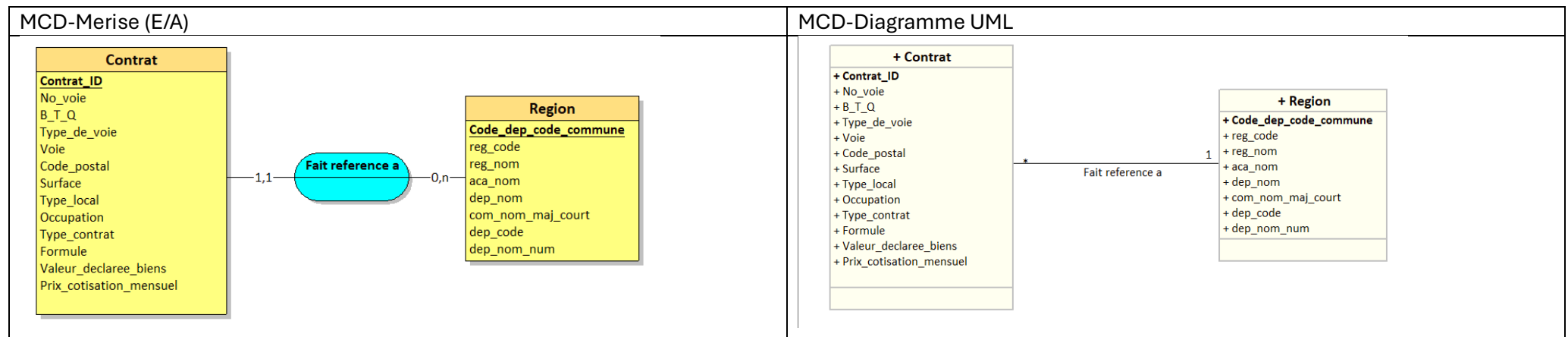
Nous pouvons donc modéliser cette situation selon un **modèle relationnel**, pour l'introduire dans une **base de données relationnelles**.

Etablissons le Modèle Conceptuel de données avec Looping. (Pour des raisons d'incompatibilité Java, je n'ai pu installer le logiciel SQL Power Architect).

### A. Modèle Conceptuel de Données (MCD) :

Nous effectuons une première ébauche de Modèle de données avec les données dont nous disposons :

#### 1. Etablissement du MCD :

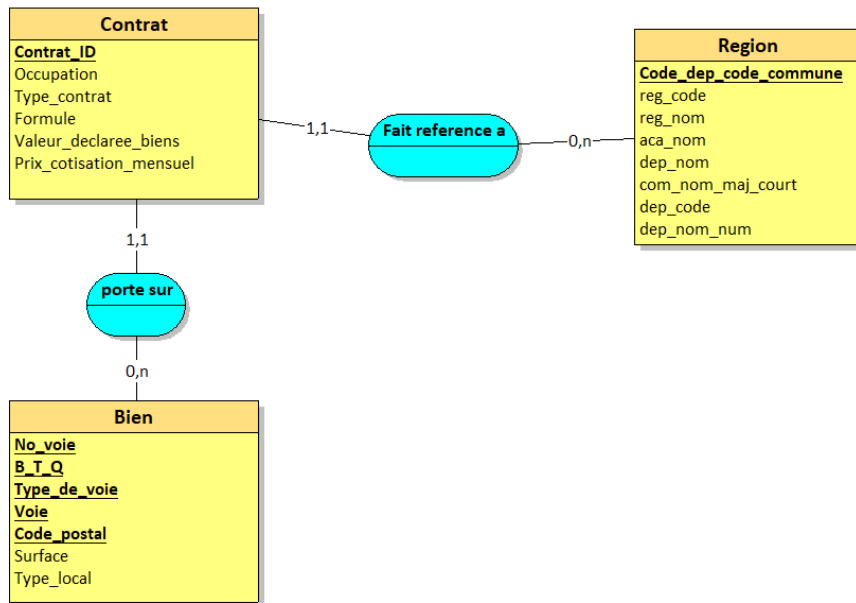


(Le logiciel Looping permet facilement de basculer d'une convention Merise à UML, et inversement)

#### 2. Réflexion sur le MCD :

Je remarque que plusieurs contrats pourraient appeler plusieurs fois le même bien (par exemple si le bien est assuré par le propriétaire et le locataire).

Afin de préserver l'unicité du Bien, nous pourrions créer une entité Bien :



La clé de l'entité Bien serait donc (No\_voie , B\_T\_Q , Type\_de\_voie , Voie , Code\_postal )

Cependant je remarque que selon les données telles qu'elles ont été indiquées **2 appartements au même endroits n'ont pas été bien distingués** par l'adresse (avec par exemple un numéro d'appartement dans l'attribut B\_T\_Q) :

Contrat_ID	No_voie	B_T_Q	Type_de_voie	Voie	Code_dep	Code_pos	Surface	Type_local	Occupation	Type_contrat	Formule	Valeur_de	Prix_cotis
100896	5028		CHE	DE STE ANNE	4094	4800	103	Appartement	Proprietaire	Residence se	Classique	0-25000	17
100951	5028		CHE	DE STE ANNE	4094	4800	25	Appartement	Locataire	Residence pr	Classique	0-25000	10

Il n'y a **pas non plus d'identifiant Bien qui permettrait de les distinguer**.

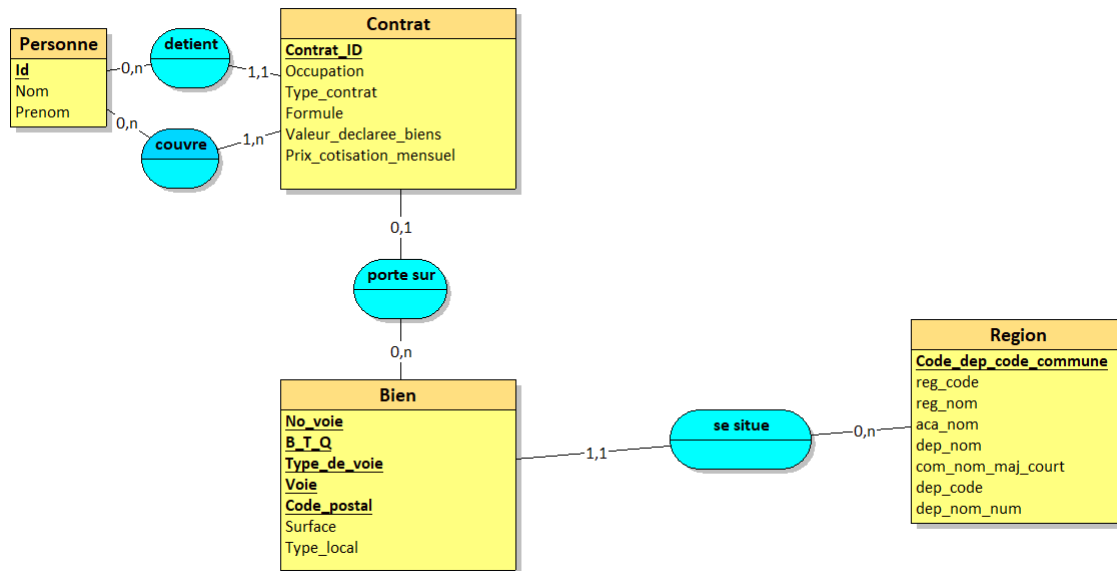
Avec les données qui nous sont fournis, nous ne pouvons donc pas faire un Modèle ainsi car nous ne pourrions pas distinguer ces 2 appartements.

Nous pourrions faire une distinction en ajoutant la surface dans la clé, pour distinguer ces 2 appartements, mais on pourrait se retrouver avec 2 appartements dans un même immeuble ayant la même surface (assez facilement si tous les étages d'un immeuble sont conçus de la même manière).

Contrat_ID	No_voie	B_T_Q	Type_de_voie	Voie	Code_dep	Code_pos	Surface	Type_local	Occupation	Type_contrat	Formule	Valeur_de	Prix_cotis
120999	9001		RES	VOIE DU SUD	91345	91160	84	Appartement	Locataire	Residence pr	Integral	0-25000	12
121071	9001		RES	VOIE DU SUD	91345	91160	44	Appartement	Proprietaire	Residence pr	Classique	0-25000	12
121269	9001		RES	VOIE DU SUD	91345	91160	43	Appartement	Proprietaire	Residence pr	Integral	0-25000	10
Contrat_ID	No_voie	B_T_Q	Type_de_voie	Voie	Code_dep	Code_pos	Surface	Type_local	Occupation	Type_contrat	Formule	Valeur_de	Prix_cotis
123348	103		AV	VERDIER	92049	92120	23	Appartement	Proprietaire	Mise en loca	Integral	0-25000	15
123677	103		AV	VERDIER	92049	92120	23	Appartement	Proprietaire	Residence pr	Classique	0-25000	16

Nous ne pouvons donc pas ici nous orienter vers la création d'une entité Bien, car il y aurait un incident avec des logements distincts mais qui auraient la même clé !

Pour aller plus loin, nous aurions pu modéliser la situation de la manière suivante :

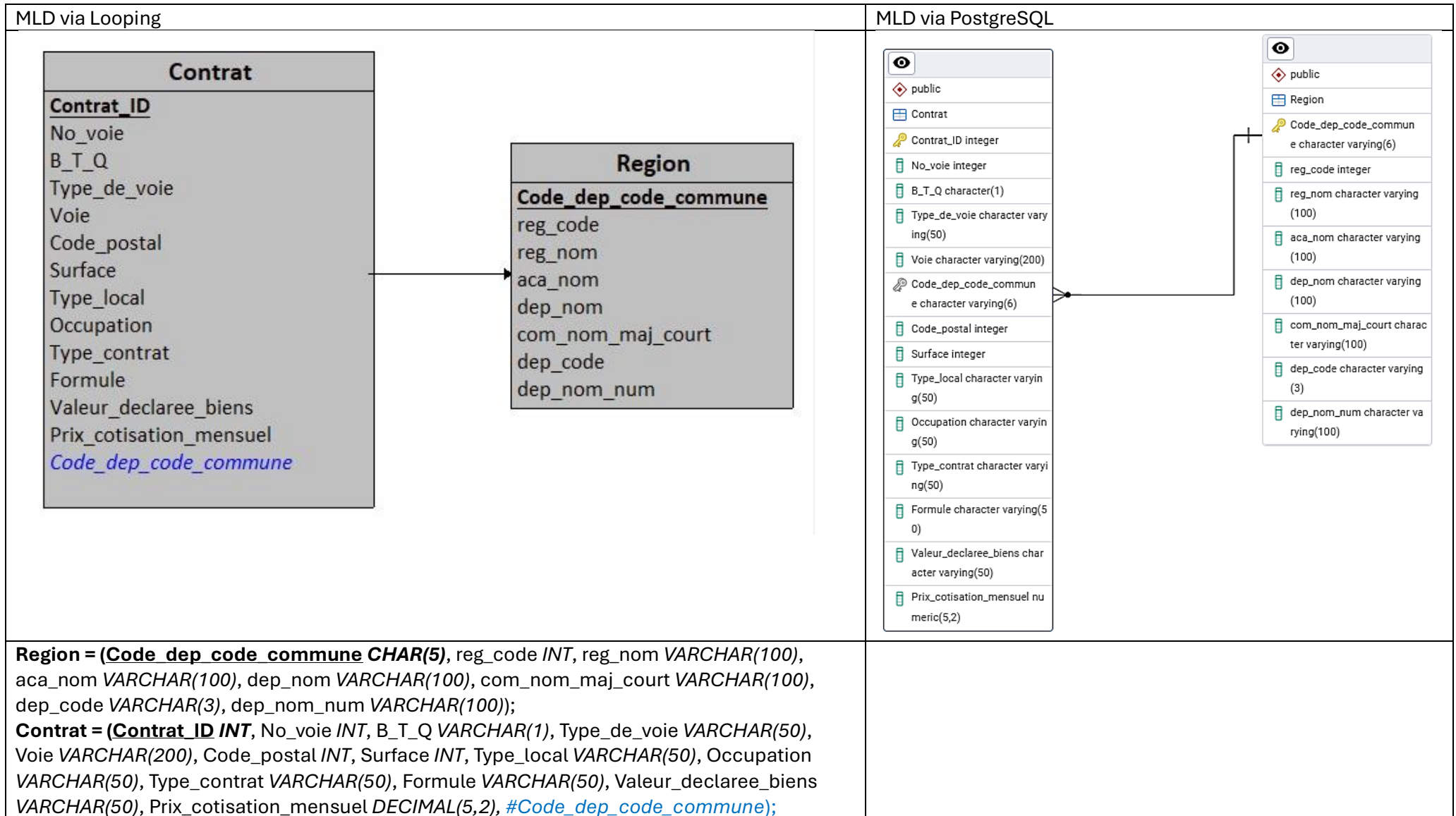


Cependant nous aurions toujours le problème d'unicité de l'entité Bien, en l'absence d'un identifiant pour l'entité Bien. Et les données ne nous permettent pas d'avoir des entités personne.

Tenant compte des données que nous avons à disposition, dans les fichiers Contrat.csv et Region.csv , nous resterons donc sur les MCD établis au 1.

## B. Modèle Logique de Données -MLD :

Tenant compte du Modèle Conceptuel de Données retenue vu au A.1 , selon les règles de conversions, nous transformons ce MCD en MLD aboutissant au résultat suivant :



(Le logiciel Looping permet facilement de basculer du MCD au MLD graphique et textuel)



### III. Construction de la Base de Données dans le SGBDR :

#### A. Création des tables :

##### 1. Le script SQL de création de tables

Les 2 outils permettent de générer le code SQL permettant de créer les tables associées au MLD :

Script SQL généré par Looping	Script SQL généré par PostgreSQL
<pre>CREATE TABLE Region(   Code_dep_code_commune CHAR(5),   reg_code INT NOT NULL,   reg_nom VARCHAR(100) NOT NULL,   aca_nom VARCHAR(100) NOT NULL,   dep_nom VARCHAR(100) NOT NULL,   com_nom_maj_court VARCHAR(100) NOT NULL,   dep_code VARCHAR(3) NOT NULL,   dep_nom_num VARCHAR(100) NOT NULL,   PRIMARY KEY(Code_dep_code_commune) );  CREATE TABLE Contrat(   Contrat_ID INT,   No_voie INT,   B_T_Q VARCHAR(1),   Type_de_voie VARCHAR(50),   Voie VARCHAR(200) NOT NULL,   Code_postal INT NOT NULL,   Surface INT,   Type_local VARCHAR(50),   Occupation VARCHAR(50),   Type_contrat VARCHAR(50),   Formule VARCHAR(50),   Valeur_declaree_biens VARCHAR(50),   Prix_cotisation_mensuel DECIMAL(5,2),   Code_dep_code_commune CHAR(5) NOT NULL,   PRIMARY KEY(Contrat_ID),   FOREIGN KEY(Code_dep_code_commune) REFERENCES Region(Code_dep_code_commune) );</pre>	<pre>BEGIN;  CREATE TABLE IF NOT EXISTS public."Contrat" (   "Contrat_ID" integer NOT NULL,   "No_voie" integer,   "B_T_Q" "char",   "Type_de_voie" character varying(50),   "Voie" character varying(200) NOT NULL,   "Code_dep_code_commune" character varying(6),   "Code_postal" integer NOT NULL,   "Surface" integer,   "Type_local" character varying(50),   "Occupation" character varying(50),   "Type_contrat" character varying(50),   "Formule" character varying(50),   "Valeur_declaree_biens" character varying(50),   "Prix_cotisation_mensuel" numeric(5, 2),   PRIMARY KEY ("Contrat_ID") );  CREATE TABLE IF NOT EXISTS public."Region" (   "Code_dep_code_commune" character varying(6) NOT NULL,   reg_code integer NOT NULL,   reg_nom character varying(100) NOT NULL,   aca_nom character varying(100) NOT NULL,   dep_nom character varying(100) NOT NULL,   com_nom_maj_court character varying(100) NOT NULL,   dep_code character varying(3) NOT NULL,   dep_nom_num character varying(100) NOT NULL,   PRIMARY KEY ("Code_dep_code_commune") );  ALTER TABLE IF EXISTS public."Contrat"   ADD CONSTRAINT "Code_dep_code_commune [FK]" FOREIGN KEY ("Code_dep_code_commune")   REFERENCES public."Region" ("Code_dep_code_commune") MATCH SIMPLE   ON UPDATE CASCADE   ON DELETE SET NULL   NOT VALID;  END;</pre>

J'ai utilisé le Script PostgreSQL que j'ai débarrassé notamment des nombreux "" autour des attributs, qui sont des artefacts inutiles et qui par la suite nuisaient aux requêtes car il fallait appeler les attributs selon la forme "Code\_dep\_code\_commune" par exemple au lieu de Code\_dep\_code\_commune .

Script SQL modifié et utilisé :

```
-- SCRIPT MODIFIED
BEGIN;

CREATE TABLE IF NOT EXISTS public.Contrat
(
    Contrat_ID integer NOT NULL,
    No_voie integer,
    B_T_Q character(1),
    Type_de_voie character varying(50),
    Voie character varying(200) NOT NULL,
    Code_dep_code_commune character varying(6),
    Code_postal integer NOT NULL,
    Surface integer,
    Type_local character varying(50),
    Occupation character varying(50),
    Type_contrat character varying(50),
    Formule character varying(50),
    Valeur_declaree_biens character varying(50),
    Prix_cotisation_mensuel numeric(5, 2),
    PRIMARY KEY (Contrat_ID)
);


















CREATE TABLE IF NOT EXISTS public.Region
(
    Code_dep_code_commune character varying(6) NOT NULL,
    reg_code integer NOT NULL,
    reg_nom character varying(100) NOT NULL,
    aca_nom character varying(100) NOT NULL,
    dep_nom character varying(100) NOT NULL,
    com_nom_maj_court character varying(100) NOT NULL,
    dep_code character varying(3) NOT NULL,
    dep_nom_num character varying(100) NOT NULL,
    PRIMARY KEY (Code_dep_code_commune)
);











ALTER TABLE IF EXISTS public.Contrat
    ADD CONSTRAINT Code_dep_code_commune_FK FOREIGN KEY (Code_dep_code_commune)
    REFERENCES public.Region (Code_dep_code_commune) MATCH SIMPLE
    ON UPDATE CASCADE
    ON DELETE SET NULL
    NOT VALID;

END;
```

## 2. Les tables créées dans PostgreSQL :

On peut alors constater la création des tables suivantes, dans notre Système de Gestion de Base de Données SGBDR, PostgreSQL :

▼  Tables (2)	
▼  contrat	
▼  Columns (14)	
	contrat_id
	no_voie
	b_t_q
	type_de_voie
	voie
	code_dep_code_commune
	code_postal
	surface
	type_local
	occupation
	type_contrat
	formule
	valeur_declaree_biens
	prix_cotisation_mensuel

▼  region	
▼  Columns (8)	
	code_dep_code_commune
	reg_code
	reg_nom
	aca_nom
	dep_nom
	com_nom_maj_court
	dep_code
	dep_nom_num

## B. Import des données :

### 1. Import des données :

Dans PostgreSQL, il est possible de charger les données des instances, à partir des fichiers .csv en indiquant le format du fichier (notamment le séparateur ; ). Et via une requête SQL, on peut vérifier le bon chargement de l'ensemble des instances, montré ci-dessous :

Table Contrat

24

25

SELECT \* FROM Contrat ;

Data Output

Messages

Notifications

contrat

Columns (14)

contrat\_id

no\_voie

b\_t\_q

type\_de\_voie

voie

code\_dep\_code\_commune

code\_postal

surface

type\_local

occupation

type\_contrat

formule

valeur\_declaree\_biens

prix\_cotisation\_mensuel

	contrat_id	no_voie	b_t_q	type_de_voie
	[PK] integer	integer	character (1)	character varyii
1	100601	190	A	RUE
2	100602	347	[null]	RUE
3	100603	58	[null]	AV
4	100604	140	[null]	RUE
5	100605	39	[null]	RUE
6	100606	8	[null]	RUE
7	100607	2	[null]	RUE
8	100608	1403	[null]	RUE
9	100609	226	[null]	Al I

Total rows: 1000 of 30335    Query complete 00:00:00.328

Table Region

25

SELECT \* FROM Region ;

Data Output

Messages

Notifications

region

Columns (8)

code\_dep\_code\_commune

reg\_code

reg\_nom

aca\_nom

dep\_nom

com\_nom\_maj\_court

dep\_code

dep\_nom\_num

Constraints

Indexes

RLS Policies

Rules

Triggers

Trigger Functions

	code_dep_code_commune	reg_code	reg_nom
	[PK] character varying (6)	integer	character varying (1
1	1001	84	Auvergne-Rhône-Al
2	1002	84	Auvergne-Rhône-Al
3	1003	84	Auvergne-Rhône-Al
4	1004	84	Auvergne-Rhône-Al
5	1005	84	Auvergne-Rhône-Al
6	1006	84	Auvergne-Rhône-Al
7	1007	84	Auvergne-Rhône-Al
8	1008	84	Auvergne-Rhône-Al
9	1009	84	Auvergne-Rhône-Al

Total rows: 1000 of 38916    Query complete 00:00:00.150

25

SELECT \* FROM Region ;

Data Output

Messages

Notifications

region

Columns (8)

code\_dep\_code\_commune

reg\_code

reg\_nom

aca\_nom

dep\_nom

com\_nom\_maj\_court

dep\_code

dep\_nom\_num

Constraints

Indexes

RLS Policies

Rules

Triggers

Trigger Functions

	code_dep_code_commune	reg_code	reg_nom
	[PK] character varying (6)	integer	character varying (1
1	1001	84	Auvergne-Rhône-Al
2	1002	84	Auvergne-Rhône-Al
3	1003	84	Auvergne-Rhône-Al
4	1004	84	Auvergne-Rhône-Al
5	1005	84	Auvergne-Rhône-Al
6	1006	84	Auvergne-Rhône-Al
7	1007	84	Auvergne-Rhône-Al
8	1008	84	Auvergne-Rhône-Al
9	1009	84	Auvergne-Rhône-Al

Total rows: 1000 of 38916    Query complete 00:00:00.150

### 2. Retour sur les difficultés lors du chargement des données :

#### 1. Obligation de charger les données du Fichier Region.csv avant les données Contrat.csv :

Lors du premier Chargement du Fichier Contrat, j'ai rencontré un message d'erreur explicite.

En effet le Fichier Contrat incluant la clé étrangère Code\_dep\_code\_commune de Regionn le SGBD, PostgreSQL a indiqué les données Code\_dep\_code\_commune n'étaient pas existante dans la Table Region (en effet car elle n'avait pas encore été alimenté du fichier CSV).

**Conclusion :** avant de charger les données dans une table ayant des clés étrangères, il faut bien s'assurer avant d'avoir chargé les données correspondantes dans les Tables ou sont référencées ces clés étrangères.

## 2. Erreur sur les données incluses dans le Fichier Contrat.csv :

Après le chargement, de la table Region, une nouvelle erreur est apparue lors du chargement des données de la Table Contrat :

ERREUR: une instruction insert ou update sur la table « Contrat » viole la contrainte de clé étrangère « Code\_dep\_code\_commune [FK] »

DETAIL: La clé (Code\_dep\_code\_commune)=(97460) n'est pas présente dans la table « Region ».

En effet le **Code\_dep\_code\_commune 97460 existe** dans la table **Contrat.csv**, alors qu'il **n'existe pas** dans la table **Region.csv**:

Table Contrat.csv										Table Region.csv									
	A	B	C	D	E	F	G	H	I		Code_dep	reg_code	reg_nom	aca_nom	dep_nom	com_nom	dep_code	dep_nom	jm
1	Contrat_ID	No_voie	B_T_Q	Type_de_v	Voie	Code_dep	Code_pos	Surface	Type_loca	12	974401	4	La Réunion	La Réunion	La Réunion	LES AVIRONS	974	La Réunion (974)	
25500	128054	2		RUE	JACQUOT	97460	97460	50	Appartem	13	974402	4	La Réunion	La Réunion	La Réunion	BRAS PANON	974	La Réunion (974)	
25503	128061	26 B		IMP	YLANG YLANG	97460	97460	84	Maison	14	974403	4	La Réunion	La Réunion	La Réunion	ENTRE DEUX	974	La Réunion (974)	
25511	128077	34		RUE	DE LA CAVERI	97460	97460	55	Appartem	15	974404	4	La Réunion	La Réunion	La Réunion	LE TANG SALI	974	La Réunion (974)	
25513	128082	7		RUE	DES SAPHIRS	97460	97460	59	Appartem	16	974405	4	La Réunion	La Réunion	La Réunion	PETITE ILE	974	La Réunion (974)	
30337										17	974406	4	La Réunion	La Réunion	La Réunion	LA PLAINE DE	974	La Réunion (974)	
30338										18	974407	4	La Réunion	La Réunion	La Réunion	LE PORT	974	La Réunion (974)	
30339										19	974408	4	La Réunion	La Réunion	La Réunion	LA POSSESSI	974	La Réunion (974)	
30340										20	974409	4	La Réunion	La Réunion	La Réunion	ST ANDRE	974	La Réunion (974)	
										21	974410	4	La Réunion	La Réunion	La Réunion	ST BENOIT	974	La Réunion (974)	
										22	974411	4	La Réunion	La Réunion	La Réunion	ST DENIS	974	La Réunion (974)	
										23	974412	4	La Réunion	La Réunion	La Réunion	ST JOSEPH	974	La Réunion (974)	
										24	974413	4	La Réunion	La Réunion	La Réunion	ST LEU	974	La Réunion (974)	
										25	974414	4	La Réunion	La Réunion	La Réunion	ST LOUIS	974	La Réunion (974)	
										26	974415	4	La Réunion	La Réunion	La Réunion	ST PAUL	974	La Réunion (974)	
										27	974416	4	La Réunion	La Réunion	La Réunion	ST PIERRE	974	La Réunion (974)	
										28	974417	4	La Réunion	La Réunion	La Réunion	ST PHILIPPE	974	La Réunion (974)	
										29	974418	4	La Réunion	La Réunion	La Réunion	STE MARIE	974	La Réunion (974)	
										30	974419	4	La Réunion	La Réunion	La Réunion	STE ROSE	974	La Réunion (974)	
										31	974420	4	La Réunion	La Réunion	La Réunion	STE SUZANNI	974	La Réunion (974)	
										32	974421	4	La Réunion	La Réunion	La Réunion	SALAZIE	974	La Réunion (974)	
										33	974422	4	La Réunion	La Réunion	La Réunion	LE TAMPON	974	La Réunion (974)	
										34	974423	4	La Réunion	La Réunion	La Réunion	LES TROIS BA	974	La Réunion (974)	
										35	974424	4	La Réunion	La Réunion	La Réunion	CILAOS	974	La Réunion (974)	
										36									

Code Postal 97460, ville et département 974

En effet le code postal 97460 correspond à la commune de Saint Paul : Le code postal 97460 correspond à la commune de Saint-Paul située dans le sud-ouest de

Et dans notre table **Region.csv** l'attribut **Code\_dep\_code\_commune** qui correspond à Saint Paul est **974415 et non 97460** (qui est le Code postal uniquement)

Afin de pouvoir poursuivre l'exercice nous rectifions le fichier Contrat csv avec le bon **Code\_dep\_code\_commune** de Saint Paul-CP 97460 qui est **974415**.

J'ai rencontré la difficulté pour les 3 **Code\_dep\_code\_commune** suivants dans le **fichier Contrat.csv** que j'ai corrigé :

le **bon Code\_dep\_code\_commune** de **Saint Paul-CP 97460** est **974415**

le **bon Code\_dep\_code\_commune** de **Trois Bassins-CP 97434** est **974423**

le **bon Code\_dep\_code\_commune** de **Saint Benoit-CP 97470** est **974410**

Conclusion : Il arrive que les données soient erronées. Lorsque c'est possible, sans ambiguïté, il faut les corriger.