

## Projet Data Analyst

Aidez une entreprise de l'assurance  
en analysant le marché  
des assurances habitation



Créer et Requêter  
une base de données  
relationnelle  
avec SQL

# Sommaire



Etape 1-Découvertes des données

Etape 2-Conception du schema relationnel

Etape 3-Création de la base de données

Etape 4-Rédaction de requêtes SQL

Méthodologie dans le cadre d'une requête complexe et à ajuster

Etape 4-Rédaction des premières requêtes SQL

Etape 5-Requêtes SQL

Annexes

# Etape 1-Découvertes des données

## Les Fichiers source au format CSV :

### - Contrat.csv

```
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
```

### - Region.csv

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

## Le dictionnaire des données :

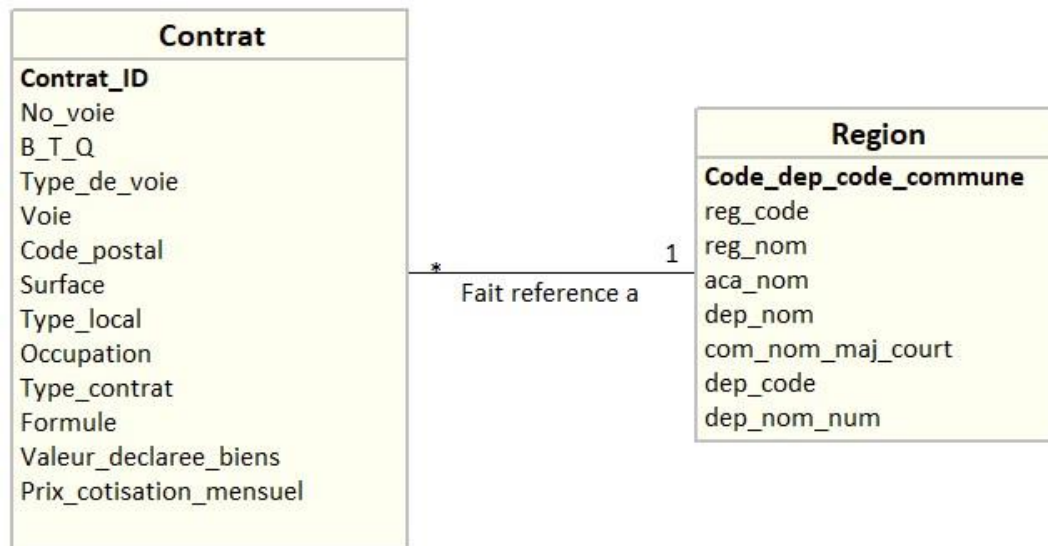
	Nom des colonnes	Type de données	Taille
CONTRAT.CSV	Contrat_ID	INT	6
	No_voie	INT	5
	B_T_Q	CHAR	1
	Type_de_voie	VARCHAR	50
	Voie	VARCHAR	200
	Code_dep_code_commune	VARCHAR	6
	Code_postal	INT	5
	Surface	INT	5
	Type_local	VARCHAR	50
	Occupation	VARCHAR	50
	Type_contrat	VARCHAR	50
	Formule	VARCHAR	50
	Valeur_declaree_biens	VARCHAR	50
REGION.CSV	Prix_cotisation_mensuel	FLOAT, NUMERIC(5,2)	7
	Code_dep_code_commune	VARCHAR	6
	reg_code	INT	2
	reg_nom	VARCHAR	100
	aca_nom	VARCHAR	100
	dep_nom	VARCHAR	100
	com_nom_maj_court	VARCHAR	100
	dep_code	VARCHAR	3
	dep_nom_num	VARCHAR	100

## Une ligne du dictionnaire de données :

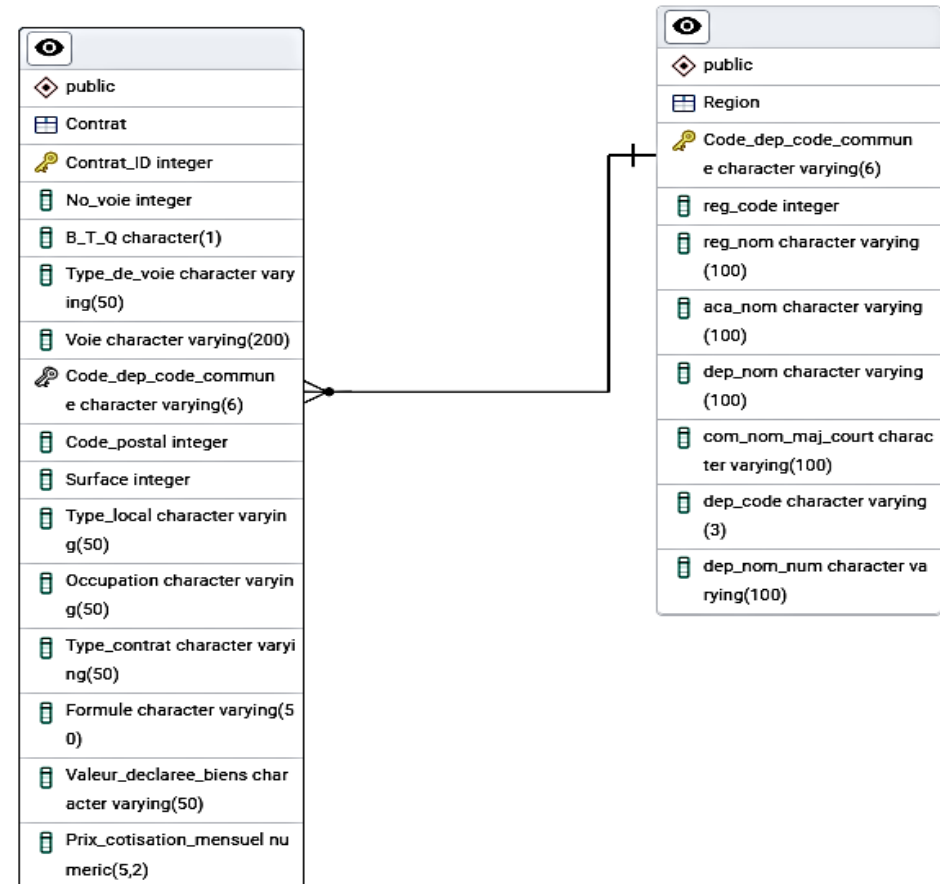
Nom des colonnes	Type de données	Taille	Clé	Contraintes	Description
Contrat_ID	INT	6	Clé primaire	NOT NULL, UNIQUE	Id unique pour les contrats

# Etape 2-Conception du schema relationnel

## MCD –UML-avec Looping



## MLD-avec PostgreSQL



# Etape 3-Création de la base de données

## Choix du SGBDR :



### Avantages de PostgreSQL :

- Intègre MLD, script SQL
- Open Source et Gratuit
- Conforme aux standards SQL
- type de données variées
- Extension possible
- Fiabilité
- Support données massives
- Compatibilité
- utilisé en entreprise

## Script SQL généré par PGSQL ajusté :

```
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;
```

# Etape 3-Création de la base de données

## Les tables générées :



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	

## Données chargées à partir des fichiers CSV :

```
1 SELECT * FROM Contrat;
```

Data Output Messages Notifications

	contrat_id [PK] integer	no_voie integer	b_t_q character (1)	type_de_voie character varying (50)	voie character varying (200)
1	100601	190	A	RUE	CENTRALE
2	100602	347	[null]	RUE	DU CHATEAU

Total rows: 1000 of 30335 Query complete 00:00:01.487 Ln 1, Col 23

```
1 SELECT * FROM Region;
```

Data Output Messages Notifications

	code_dep_code_commune [PK] character varying (6)	reg_code integer	reg_nom character varying (100)	aca_nom character varying (100)
1	1001	84	Auvergne-Rhône-Alpes	Lyon
2	1002	84	Auvergne-Rhône-Alpes	Lyon

Total rows: 1000 of 38916 Query complete 00:00:00.959 Ln 1, Col 22



# Etape 4-Rédaction de requêtes SQL

Méthodologie dans le cadre d'une requête complexe et à ajuster

## Exemple Requête 7 :

Quel est le nombre de formules "integral" sur la région Pays de la Loire ?

### Écriture de requêtes connexes pour assurer la compréhension :

```
--REQUETE 7
SELECT Formule, COUNT(contrat_id) AS
"Nombre de Formule en Pays de la Loire"
FROM Contrat C, Region R
WHERE C.Code_dep_code_commune
=R.Code_dep_code_commune
AND reg_nom='Pays de la Loire'
--AND Formule='integral'
GROUP BY Formule;
```

	formule character varying (50) 🔒	Nombre de Formule en Pays de la Loire bigint 🔒
1	Integral	589
2	Classique	607

### Écriture de requêtes connexes pour assurer la compréhension (2):

```
--REQUETE 7bis
SELECT Formule, COUNT(contrat_id) AS
"Nombre de Formule en Pays de la Loire"
FROM Contrat C, Region R
WHERE C.Code_dep_code_commune
=R.Code_dep_code_commune
AND reg_nom='Pays de la Loire'
AND Formule='Integral'
GROUP BY Formule;
```

	formule character varying (50) 🔒	Nombre de Formule en Pays de la Loire bigint 🔒
1	Integral	589

### Écriture de requêtes connexes pour assurer la compréhension (2):

```
--REQUETE 7ter
SELECT COUNT(contrat_id) AS "Nombre de
Formule en Pays de la Loire"
FROM Contrat C, Region R
WHERE C.Code_dep_code_commune
=R.Code_dep_code_commune
AND reg_nom='Pays de la Loire'
AND Formule='Integral'
GROUP BY Formule;
```

	Nombre de Formule en Pays de la Loire bigint 🔒
1	589

# Etape 4-Rédaction de requêtes SQL

Méthodologie dans le cadre d'une requête complexe et à ajuster

Exemple Requête 7 : Quel est le nombre de formules "integral" sur la région Pays de la Loire ?

## Écriture de variantes :

```
--REQUETE 7quar avec un INNER JOIN
SELECT COUNT(contrat_id) AS "Nombre de
Formule en Pays de la Loire"
FROM Contrat C
INNER JOIN Region R ON
C.Code_dep_code_commune
=R.Code_dep_code_commune
AND reg_nom='Pays de la Loire'
AND Formule='Integral'
GROUP BY Formule
;
```

	Nombre de Formule en Pays de la Loire	
	bigint	
1	589	

## Verification proposition Chat GPT :

```
--REQUETE 7//ChatGPT
SELECT COUNT(c.Contrat_ID) AS
nombre_formules_integral
FROM Contrat c
JOIN Region r ON c.Code_dep_code_commune
= r.Code_dep_code_commune
WHERE c.Formule = 'Integral' AND r.reg_nom =
'Pays de la Loire';
```

	nombre_formules_integral	
	bigint	
1	589	

## Analyse proposition Chat GPT et correction si nécessaire :

```
--REQUETE 7quar Final avec un INNER JOIN
SELECT COUNT(contrat_id) AS "Nombre de
Formule Integral en Pays de la Loire"
FROM Contrat C
INNER JOIN Region R ON
C.Code_dep_code_commune
=R.Code_dep_code_commune
WHERE reg_nom='Pays de la Loire'
AND Formule='Integral'
--GROUP BY Formule;
```

	Nombre de Formule Integral en Pays de la Loire	
	bigint	
1	589	

## ANALYSE :

- Ici on peut retirer le **GROUP BY** car, n'affichant plus la colonne Formule il n'y a plus besoin de s'assurer que les Formules soient regroupées. C'est le critère Formule='Integral' qui l'assure. Valable pour les versions 7quar et 7 ter
- La syntaxe AND fonctionne aussi, mais WHERE respecte mieux les standards.
- INNER JOIN et JOIN sont équivalents



# Etape 4-Rédaction des premières requêtes SQL

## Requête 1

Lister les numéros de contrats (contrat\_ID) avec leur surface pour la commune de Caen

```
SELECT Contrat_ID, Surface
FROM Contrat C, Region R
WHERE C.Code_dep_code_commune
=R.Code_dep_code_commune
AND com_nom_maj_court='CAEN'
;
```

	contrat_id [PK] integer	surface integer
1	103791	35
2	103792	99
3	103793	40
4	103794	20

## Requête 2

Lister les numéros de contrats avec le type de contrat et leur formule pour les maisons du département 71.

```
SELECT Contrat_ID , Type_contrat , Formule
FROM Contrat C, Region R
WHERE C.Code_dep_code_commune
=R.Code_dep_code_commune
AND dep_code='71'
AND Type_local='Maison'
;
```

	contrat_id [PK] integer	type_contrat character varying (50)	formule character varying (50)
1	114768	Residence principale	Integral
2	114779	Residence principale	Classique
3	114782	Residence principale	Classique
4	114812	Residence principale	Integral

## Requête 3

Lister le nom des régions de France

```
SELECT DISTINCT reg_nom FROM region ;
```

reg_nom character varying (100)	10	Guyane
1	11	Bourgogne-Franche-Comté
2	12	Provence-Alpes-Côte d'Azur
3	13	Martinique
4	14	Guadeloupe
5	15	Normandie
6	16	Nouvelle-Aquitaine
7	17	Occitanie
8	18	Pays de la Loire
9	19	Centre-Val de Loire
Collectivités d'outre-mer	Total rows: 19 of 19    Query com	

# Etape 5-Requêtes SQL

## Requête 4

Quels sont les 5 contrats qui ont les surfaces les plus élevées ?

```
SELECT Contrat_ID, Surface FROM Contrat
ORDER BY Surface DESC
LIMIT 5 ;
```

	contrat_id [PK] integer	surface integer
1	104211	815
2	105463	742
3	130878	595
4	100822	570
5	109872	559

## Requête 5

Quel est le prix moyen de la cotisation mensuelle ?

```
SELECT AVG(Prix_cotisation_mensuel) AS "Prix
moyen de la cotisation mensuelle"
FROM Contrat ;
```

	Prix moyen de la cotisation mensuelle numeric
1	19.3286962254821164

## Requête 6

Quel est le nombre de contrats pour chaque catégorie de prix de la valeur déclarée des biens ?

```
SELECT valeur_declaree_biens As "Valeur déclarée
des biens", COUNT (contrat_id) AS "Nombre de
contrats" FROM Contrat
GROUP BY valeur_declaree_biens;
```


	Valeur déclarée des biens character varying (50)	Nombre de contrats bigint
1	50000-100000	696
2	100000+	104
3	25000-50000	6815
4	0-25000	22720

# Etape 5-Requêtes SQL

## Requête 7

Quel est le nombre de formules  
“integral” sur la région Pays de la Loire ?




```
SELECT COUNT(contrat_id) AS "Nombre de Formule  
Integral en Pays de la Loire"  
FROM Contrat C  
INNER JOIN Region R ON  
C.Code_dep_code_commune  
=R.Code_dep_code_commune  
WHERE reg_nom='Pays de la Loire'  
AND Formule='Integral';
```

	Nombre de Formule Integral en Pays de la Loire 
1	589

## Requête 8

Lister les numéros de contrats avec le  
type de contrat et leur formule pour les  
maisons du département 71.


```
SELECT Contrat_ID , Type_contrat , Formule  
FROM Contrat C  
INNER JOIN Region R ON  
C.Code_dep_code_commune  
=R.Code_dep_code_commune  
WHERE dep_code='71'  
AND Type_local='Maison';
```

	contrat_id  [PK] integer	type_contrat  character varying (50)	formule  character varying (50)
1	114768	Residence principale	Integral
2	114779	Residence principale	Classique
3	114782	Residence principale	Classique
4	114812	Residence principale	Integral

## Requête 9

Quelle est la surface moyenne des  
contrats à Paris ?

```
SELECT AVG(Surface) AS "Surface Moyenne des  
Contrats parisiens"  
FROM Contrat C  
INNER JOIN Region R ON  
C.Code_dep_code_commune  
=R.Code_dep_code_commune  
WHERE dep_code='75';
```

	Surface Moyenne des Contrats parisiens 
1	51.7695498859157851

# Etape 5-Requêtes SQL

## Requête 10

Classement des 10 départements où le prix moyen de la cotisation est le plus élevé.

```
SELECT dep_nom_num AS Département,
AVG(prix_cotisation_mensuel) AS "Prix moyen de cotisation"
FROM Contrat C
JOIN Region R ON C.Code_dep_code_commune
=R.Code_dep_code_commune
GROUP BY dep_nom_num
ORDER BY AVG(prix_cotisation_mensuel) DESC
LIMIT 10;
```

	département character varying (100)	Prix moyen de cotisation numeric
1	Paris (75)	36.3990873262808546
2	Hauts-de-Seine (92)	26.2693584070796460
3	Val-de-Marne (94)	19.8241639697950378
4	Yvelines (78)	18.8880697050938338
5	Rhône (69)	18.4855182926829268
6	Ain (01)	18.2388888888888889
7	Alpes-Maritimes (06)	18.1376228775692583
8	Charente-Maritime (17)	17.3173076923076923
9	Haute-Savoie (74)	17.1487341772151899
10	Corse-du-Sud (2A)	17.0743801652892562

## Requête 11

Liste des communes ayant eu au moins 150 contrats.

```
SELECT com_nom_maj_court AS Commune,
COUNT(contrat_id) AS "Nombre de Contrats"
FROM Contrat C
JOIN Region R ON C.Code_dep_code_commune
=R.Code_dep_code_commune
GROUP BY Commune
HAVING COUNT(contrat_id)>=150;
```

	commune character varying (100)	Nombre de Contrats bigint
1	PARIS 15	407
2	LILLE	161
3	COURBEVOIE	163
4	NICE	387
5	PARIS 16	394
6	PARIS 3	159
7	NANTES	291
8	TOULON	170
9	PARIS 14	222
10	TOULOUSE	187

11	PARIS 12	252
12	BORDEAUX	302
13	GRENOBLE	220
14	PARIS 10	263
15	PARIS 11	381
16	PARIS 18	515
17	PARIS 20	302
18	PARIS 19	266
19	PARIS 9	204
20	PARIS 17	468
Total rows: 20 of 20		Query complete 00:00:00.120

## Requête 12

Quel est le nombre de contrats pour chaque région ?

```
SELECT reg_nom AS "Nom de Région",
COUNT(contrat_id) AS "Nombre de Contrats"
FROM Contrat C
JOIN Region R ON C.Code_dep_code_commune
=R.Code_dep_code_commune
GROUP BY reg_nom;
```

	Nom de Région character varying (100)	Nombre de Contrats bigint
1	Hauts-de-France	1189
2	La Réunion	17
3	Bretagne	947
4	Auvergne-Rhône-Alpes	3042
5	Ile-de-France	14177
6	Grand Est	769
7	Guyane	37
8	Corse	247
9	Bourgogne-Franche-Comté	293
10	Provence-Alpes-Côte d'Azur	3279
11	Martinique	73
12	Normandie	824
13	Nouvelle-Aquitaine	2038
14	Occitanie	1609
15	Pays de la Loire	1196
16	Centre-Val de Loire	598

# Projet Data Analyst



# Merci !

Créer et Requêter  
une base de données  
relationnelle  
avec SQL

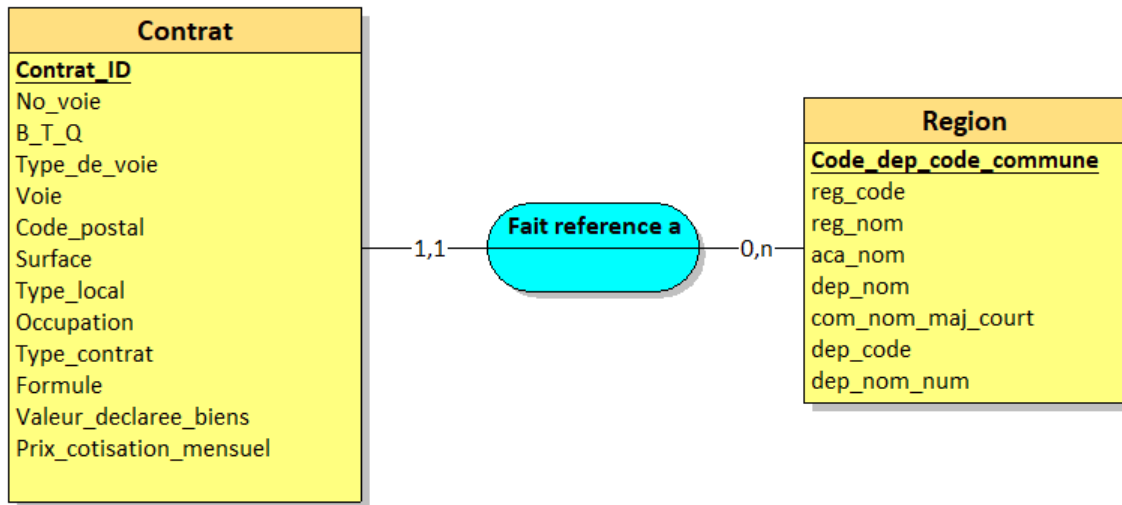
# Annexe-Etape 1-Le dictionnaire des données

	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 )
REGION.CSV	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
	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



# Annexe-Etape 2- Looping

## MCD –Merise-avec Looping



## MLD-avec Looping

