

3 - Méthodologie



Parcours Data Analyst - Marie G.
Projet 3 : Requêtez une base de données avec SQL
19/02/2025

1. Exploration des données

2. Schéma relationnel de la base de données

3. Création des tables

4. Chargement des données

5. Interrogation des tables par des requêtes SQL

1. Exploration des données

2 fichiers csv sont fournis dans le cadre du projet pour construire notre base de données :

- Contrat+(4).csv : Liste des contrats d'assurance logement de l'entreprise (30 335 lignes de contrats)
- Region+(7).csv : Liste des communes de France, dont les données sont issues d'un site gouvernemental (38 916 lignes de communes) Certains codes communes issus d'anciens découpages administratifs sont obsolètes, comme ceux de l'ancien département de la Seine comprenant la ville de Paris et les communes de petite couronne.

Le dictionnaire de données suivant est construit en analysant sous Excel les données des deux fichiers, après création de tableaux pour une meilleure vision de l'ensemble des données pouvant être stockée dans chaque colonne.

Il reprend les éléments qui seront nécessaire pour construire la base de donnée SQL, ainsi qu'une description en documentation annexe

	Nom des colonnes	Type de données	Taille	Clé	Description
CONTRAT.CSV	Contrat_ID	INT	5	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	4		Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ...
	Voie	VARCHAR	100		Libellé de la voie pour l'adresse du logement assuré
	Code_dep_code_commune	VARCHAR	6	Clé secondaire	Concaténation du code département et code commune pour avoir une clé commune unique
	Code_postal	VARCHAR	5		Code postal pour l'adresse du logement assuré
	Surface	INT	4		Surface du logement assuré (m²)
	Type_local	VARCHAR	100		Type de logement (appartement ou maison)
	Occupation	VARCHAR	100		Occupant (locataire ou propriétaire)
	Type_contrat	VARCHAR	100		Occupation (Résidence principale, résidence secondaire ou mise en location)
	Formule	VARCHAR	100		Formule choisie comme assurance (classique ou intégrale)
	Valeur_declaree_biens	VARCHAR	100		Valeur des biens déclarés à l'assurance (€)
REGION.CSV	Prix_cotisation_mensuel	INT	4		Cotisation mensuelle de l'assuré (€)
	Code_dep_code_commune	VARCHAR	6	Clé primaire	Concaténation du code département et code commune pour avoir une clé commune unique
	reg_code	INT	2		Code région
	reg_nom	VARCHAR	100		Nom de la région
	aca_nom	VARCHAR	100		Nom de l'académie
	dep_nom	VARCHAR	100		Nom du département
	com_nom_maj_court	VARCHAR	100		Nom de la commune
	dep_code	VARCHAR	3		Code département
	dep_nom_num	VARCHAR	100		Nom et code du département

Dictionnaire des données de la future base de données

2. Schéma relationnel de la base de données

Le dictionnaire de données présenté à la page précédente permet de modifier le schéma relationnel fourni dans le logiciel **SQL Power Architect**



Points d'attention :

1. SQL Power Architect ne permet pas de donner une taille aux données de type INTEGER. La taille des données renseignée dans le dictionnaire construit devra donc être ajoutée dans la commande de création de tables extraite par SQL Power Architect
2. Les colonnes doivent respecter l'ordre des colonnes fichiers csv prévus pour l'import de données

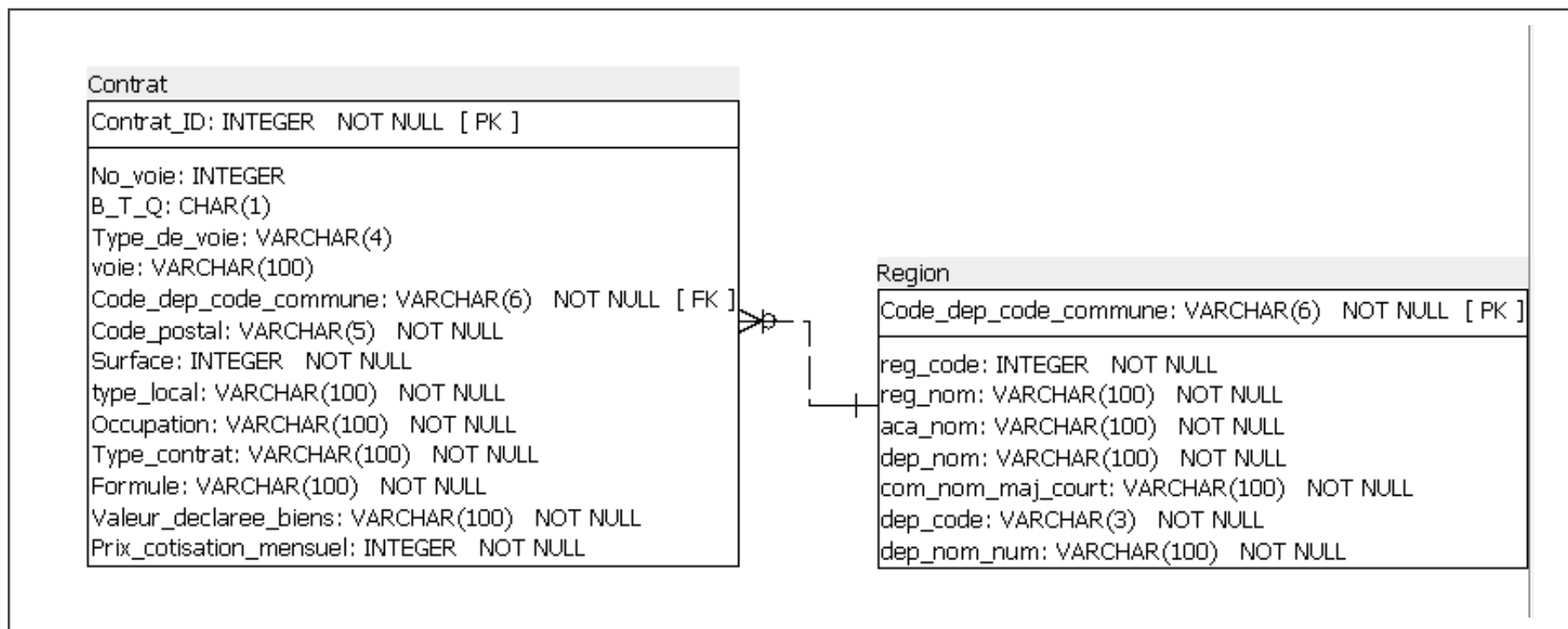


Schéma relationnel de la base de données

3. Création des tables

SQL Power Architect permet de générer le code ci-dessous sous le choix de format MySQL, après :

- Ajout de la taille des champs de format INTEGER
- Mise en forme sous <https://www.dpriver.com/pp/sqlformat.htm>
- Intégration de l'instruction « FOREIGN KEY » directement dans la création de la table « contrat »

```
CREATE TABLE region
(
  code_dep_code_commune2 VARCHAR(6) NOT NULL,
  reg_code               INT NOT NULL,
  dep_nom_num            VARCHAR(100) NOT NULL,
  dep_code               VARCHAR(3) NOT NULL,
  com_nom_maj_court      VARCHAR(100) NOT NULL,
  dep_nom                VARCHAR(100) NOT NULL,
  reg_nom                VARCHAR(100) NOT NULL,
  aca_nom                VARCHAR(100) NOT NULL,
  PRIMARY KEY (code_dep_code_commune2)
);

CREATE TABLE contrat
(
  contrat_id             INT NOT NULL,
  type_local             VARCHAR(100) NOT NULL,
  valeur_declaree_biens  VARCHAR(100) NOT NULL,
  formule                VARCHAR(100) NOT NULL,
  no_voie                INT,
  surface                INT NOT NULL,
  b_t_q                 CHAR(1) NOT NULL,
  code_dep_code_commune2 VARCHAR(6) NOT NULL,
  type_de_voie           VARCHAR(4),
  voie                  VARCHAR(100) NOT NULL,
  type_contrat           VARCHAR NOT NULL,
  occupation             VARCHAR(100) NOT NULL,
  code_postal            VARCHAR(5) NOT NULL,
  prix_cotisation_mensuel INT NOT NULL,
  PRIMARY KEY (contrat_id)
  FOREIGN KEY (code_dep_code_commune2) REFERENCES region (
    code_dep_code_commune2)
);
```

Requête SQL de génération des tables



On utilise le logiciel SQLiteStudio pour créer les tables vides à partir de la requête SQL :

Structure										
Données										
Contraintes										
Index										
Déclencheurs										
DDL										
bdd_mg_p3		Nom de la table : region		<input type="checkbox"/> WITHOUT ROWID		<input type="checkbox"/> STRICT				
	Nom	Type de données	Clé primaire	Clé étrangère	Unique	Contrôle	Non NULL	Collecter	Généré	Valeur par défaut
1	code_dep_code_commune2	VARCHAR (6)	🔑				🚫			NULL
2	reg_code	INTEGER (2)					🚫			NULL
3	reg_nom	VARCHAR (100)					🚫			NULL
4	aca_nom	VARCHAR (100)					🚫			NULL
5	dep_nom	VARCHAR (100)					🚫			NULL
6	com_nom_maj_court	VARCHAR (100)					🚫			NULL
7	dep_code	VARCHAR (3)					🚫			NULL
8	dep_nom_num	VARCHAR (100)					🚫			NULL
Type	Nom	Détails								
1	PRIMARY KEY	region_pk	(code_dep_code_commune2)							

Structure

Données

Contraintes

Index

Déclencheurs

DDL

bdd_mg_p3

Nom de la table : contrat

WITHOUT ROWID

STRICT

	Nom	Type de données	Clé primaire	Clé étrangère	Unique	Contrôle	Non NULL	Collecter	Généré	Valeur par défaut
1	contrat_id	INTEGER (5)	🔑				🚫			NULL
2	no_voie	INTEGER (4)					🚫			NULL
3	b_t_q	CHAR (1)					🚫			NULL
4	type_de_voie	VARCHAR (4)					🚫			NULL
5	voie	VARCHAR (100)					🚫			NULL
6	code_dep_code_commune	VARCHAR (6)		🔗			🚫			NULL
7	code_postal	VARCHAR (5)					🚫			NULL
8	surface	INTEGER (4)					🚫			NULL
9	type_local	VARCHAR (100)					🚫			NULL
10	occupation	VARCHAR (100)					🚫			NULL
11	type_contrat	VARCHAR (100)					🚫			NULL
12	formule	VARCHAR (100)					🚫			NULL
13	valeur_declaree_biens	VARCHAR (100)					🚫			NULL
14	prix_cotisation_mensuel	INTEGER (4)					🚫			NULL

Type	Nom	Détails
1 🔑 PRIMARY KEY	contrat_pk	(contrat_id)
2 🔗 FOREIGN KEY		(code_dep_code_commune) REFERENCES region (code_dep_code_commune)

4. Chargement des données

1. Avec l'outil d'importation de données au format CSV de SQLiteStudio, on charge en premier lieu les données de la table « régions » car elle contient la donnée « code_dep_code_commune », qui est une clé secondaire de la table « contrats »
2. En chargeant la table « contrats » de la même manière, on constate que la clé secondaire n'est pas connue pour certains contrats :

[16:31:22] Erreur lors de l'import des données : FOREIGN KEY constraint failed

En effectuant une recherche sous Excel (par exemple en ajoutant dans la table « contrat » une colonne qui vérifie si la valeur « code_dep_code_commune » figure dans la table « région », on repère les lignes dont le **code commune** (code_dep_code_commune) a mal été renseigné : ici, le code postal a été renseigné au lieu du code commune sur des contrats situés à la Réunion. On enregistre alors le fichier csv avec les valeurs rectifiées grâce à une recherche sur internet. On parvient ensuite à charger la table « contrats » rectifiée.

Table « région » chargée

Base de données								
Structure								
Données								
Contraintes								
Index								
Déclencheurs								
DDL								
Table								
Formulaire								
Filtre de données								
Nombre de lignes chargées : 38916								
	code_dep_code_commune	reg_code	reg_nom	aca_nom	dep_nom	com_nom_maj_court	dep_code	dep_nom_num
1	1001	84	Auvergne-Rhône-Alpes	Lyon	Ain	L ABERGEMENT CLEMENCIAT	1	Ain (01)
2	1002	84	Auvergne-Rhône-Alpes	Lyon	Ain	L ABERGEMENT DE VAREY	1	Ain (01)
3	1003	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMAREINS	1	Ain (01)
4	1004	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBERIEU EN BUGEY	1	Ain (01)
5	1005	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBERIEUX EN DOMBES	1	Ain (01)
6	1006	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBLEON	1	Ain (01)
7	1007	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBRONAY	1	Ain (01)
8	1008	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBUTRIX	1	Ain (01)
9	1009	84	Auvergne-Rhône-Alpes	Lyon	Ain	ANDERT ET CONDON	1	Ain (01)
10	1010	84	Auvergne-Rhône-Alpes	Lyon	Ain	ANGLEFORT	1	Ain (01)
11	1011	84	Auvergne-Rhône-Alpes	Lyon	Ain	APREMONT	1	Ain (01)
12	1012	84	Auvergne-Rhône-Alpes	Lyon	Ain	ARANC	1	Ain (01)
13	1013	84	Auvergne-Rhône-Alpes	Lyon	Ain	ARANDAS	1	Ain (01)
14	1014	84	Auvergne-Rhône-Alpes	Lyon	Ain	ARBENT	1	Ain (01)
15	1015	84	Auvergne-Rhône-Alpes	Lyon	Ain	ARBIGNIEU	1	Ain (01)

Table « contrat » chargée

Base de données

Tableaux (2)

bdd_mg_P3 (SQLite 3)

contrat

region

Vues

Structure

Données

Contraintes

Index

Déclencheurs

DDL

Table

Formulaire

Filtre de données

Nombre de lignes chargées : 30335

	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_declare	prix_cotisation_mensuel
1	100601	190	A	RUE	CENTRALE	1350	1370	50	Appartement	Locataire	Residence principale	Integral	0-25000	25
2	100602	347	NULL	RUE	DU CHATEAU	1103	1170	48	Appartement	Locataire	Residence principale	Classique	0-25000	30
3	100603	58	NULL	AV	DU MONT BLANC	1143	1220	131	Appartement	Proprietaire	Residence principale	Integral	25000-50000	57
4	100604	140	NULL	RUE	DE L'ABBE JOLIVET	1288	1630	109	Maison	Locataire	Residence principale	Integral	25000-50000	43
5	100605	39	NULL	RUE	BUFFON	1033	1200	109	Appartement	Locataire	Residence principale	Classique	0-25000	33
6	100606	8	NULL	RUE	DE GENEVE	1354	1630	53	Appartement	Proprietaire	Residence principale	Classique	0-25000	19
7	100607	2	NULL	RUE	DU RECULET	1354	1630	59	Appartement	Proprietaire	Residence principale	Integral	0-25000	15
8	100608	1403	NULL	RUE	JEAN DE GINGINS	1143	1220	93	Maison	Proprietaire	Mise en location	Integral	25000-50000	34
9	100609	226	NULL	ALL	DES CAPUCINES	1354	1630	117	Maison	Proprietaire	Residence principale	Classique	25000-50000	32
10	100610	276	NULL	RTE	DE POUGVY	1288	1630	36	Appartement	Proprietaire	Residence principale	Integral	25000-50000	22
11	100611	79	NULL	CRS	DE VERDUN	1283	1100	138	Appartement	Proprietaire	Residence secondaire	Classique	0-25000	11
12	100612	240	NULL	RUE	DE PRE BAILLY	1173	1170	45	Appartement	Locataire	Residence principale	Classique	0-25000	16
13	100613	3	NULL	RUE	TURENNE	1033	1200	83	Appartement	Locataire	Residence principale	Classique	0-25000	14
14	100614	44	NULL	ALL	DU SQUARE DE LAUSANNE	1143	1220	88	Appartement	Locataire	Residence principale	Integral	25000-50000	34
15	100615	59	NULL	RUE	ALEXANDRE BERARD	1004	1500	165	Appartement	Locataire	Residence principale	Classique	25000-50000	24
16	100616	282	NULL	CHE	DES LONGES RAVES	1071	1170	42	Appartement	Proprietaire	Residence principale	Classique	0-25000	17
17	100617	54	NULL	GR	GRANDE RUE	1396	1150	68	Appartement	Proprietaire	Residence secondaire	Classique	0-25000	10
18	100618	1	NULL	RUE	DE GEX	1160	1210	83	Appartement	Proprietaire	Residence principale	Classique	0-25000	20
19	100619	17	NULL	LOT	LES JARDINS DE CHEVRY	1103	1170	30	Appartement	Proprietaire	Residence secondaire	Classique	0-25000	12
20	100620	329	NULL	RUE	DES CARPIERES	1399	1170	25	Appartement	Locataire	Residence principale	Integral	0-25000	11

5. Interrogation des tables par des requêtes SQL 1/4

Requête 1 : Lister les numéros de contrats (contrat_ID) avec leur surface pour la commune de Caen

```
SELECT contrat_id,
       surface,
       r.com_nom_maj_court
FROM   contrat c
       INNER JOIN region r
         ON c.code_dep_code_commune = r.code_dep_code_commune
WHERE  r.com_nom_maj_court = 'CAEN';
```

- Sélectionner pour chaque contrat son identifiant, sa surface et sa commune depuis la table « contrat »
- Jointure de la table « region » pour associer le nom de la commune à chaque contrat
- Filtrer les contrats dont la communes est Caen

	contrat_ID	surface	com_nom_maj_court
1	103791	35	CAEN
2	103792	99	CAEN
3	103793	40	CAEN
4	103794	20	CAEN

Requête 3 : Lister le nom des régions de France

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

- Sélectionner depuis la table « region » le nom des régions de chaque commune
- Commande « Distinct » pour n'afficher chaque nom de région qu'une seule fois

reg_nom	
1 Auvergne-Rhône-Alpes	11 Pays de la Loire
2 Hauts-de-France	12 Ile-de-France
3 Provence-Alpes-Côte d'Azur	13 Guadeloupe
4 Grand Est	14 Martinique
5 Occitanie	15 Guyane
6 Normandie	16 La Réunion
7 Nouvelle-Aquitaine	17 Collectivités d'outre-mer
8 Centre-Val de Loire	18 Mayotte
9 Bourgogne-Franche-Comté	19 Corse
10 Bretagne	

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
       INNER JOIN region r
         ON c.code_dep_code_commune = r.code_dep_code_commune
WHERE  r.dep_code = 71 ;
```

- Sélectionner pour chaque contrat son identifiant, le type de contrat et la formule depuis la table « contrat »
- Jointure de la table « region » pour associer le numéro de département à chaque contrat
- Filtrer les contrats dont le département est le 71

	contrat_Id	type_contrat	Formule
1	114767	Residence secondaire	Classique
2	114768	Residence principale	Integral
3	114769	Residence principale	Classique
4	114770	Residence principale	Classique
5	114771	Residence principale	Integral
6	114772	Residence principale	Classique
7	114773	Residence principale	Integral
8	114774	Residence principale	Classique
9	114775	Residence principale	Integral
10	114776	Residence principale	Classique
11	114777	Residence principale	Integral
12	114778	Residence principale	Classique
13	114779	Residence principale	Classique
14	114780	Residence principale	Integral
15	114781	Residence secondaire	Integral
16	114782	Residence principale	Classique
17	114783	Mise en location	Integral
18	114784	Residence principale	Classique
19	114785	Residence principale	Integral
20	114786	Residence principale	Classique

	contrat_Id	type_contrat	Formule
21	114787	Residence secondaire	Classique
22	114788	Residence principale	Classique
23	114789	Residence principale	Classique
24	114790	Residence principale	Integral
25	114791	Residence principale	Integral
26	114792	Mise en location	Integral
27	114793	Residence principale	Classique
28	114794	Residence secondaire	Classique
29	114795	Residence secondaire	Integral
30	114796	Residence principale	Classique
31	114797	Residence principale	Classique
32	114803	Residence principale	Integral
33	114804	Residence secondaire	Classique
34	114805	Mise en location	Integral
35	114806	Residence principale	Classique
36	114807	Residence principale	Integral
37	114808	Residence principale	Integral
38	114809	Residence principale	Classique
39	114810	Residence principale	Integral
40	114811	Residence principale	Integral

41	114812	Residence principale	Integral
42	114813	Mise en location	Classique
43	115182	Residence principale	Classique
44	115207	Residence principale	Integral
45	115221	Residence principale	Integral
46	115223	Residence principale	Classique
47	115256	Residence principale	Classique
48	115259	Residence principale	Classique

5. Interrogation des tables par des requêtes SQL 2/4

Requête 4 : 5 contrats qui ont les surfaces les plus élevées

```
SELECT contrat_id,  
       surface  
FROM   contrat  
ORDER BY surface DESC  
LIMIT 5;
```

- Sélectionner pour chaque contrat son identifiant et la formule depuis la table « contrat »
- Trier par surface décroissante
- Sélectionner les 5 premiers contrats

	Contrat_id	surface
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  
FROM   (SELECT DISTINCT contrat_id,  
                        prix_cotisation_mensuel  
        FROM   contrat);
```

- Sélectionner la moyenne de la colonne « prix_cotisation_mensuel » dans la table « contrat »
- Commande « Distinct » pour ne compter chaque contrat qu’une seule fois

	prix_moyen
1	19.32869622548212

Requête 6 : Quel est le nombre de contrats pour chaque catégorie de prix de la valeur déclarée des biens ?

```
SELECT valeur_declaree_biens,  
       Count(DISTINCT(contrat_id)) AS nombre_de_contrats  
FROM   contrat  
GROUP BY valeur_declaree_biens;
```

- Grouper les lignes de la table « contrat » par « valeur_declaree_biens »
- Compter le nombre de contrats distincts dans chaque groupe créé
- Sélectionner la donnée « valeur_declaree_biens » pour chaque décompte du nombre de contrats

	valeur_declaree_biens	nombre_de_contrats
1	0-25000	22720
2	100000+	104
3	25000-50000	6815
4	50000-100000	696

Requête 7 : Quel est le nombre de formule “integral” sur la région Pays de la Loire ?

```
SELECT r.reg_nom,  
       c.formule,  
       Count(DISTINCT( contrat_id )) AS nombre  
FROM   contrat c  
       INNER JOIN region r  
         ON c.code_dep_code_commune = r.code_dep_code_commune  
WHERE ( r.reg_nom = 'Pays de la Loire'  
       AND c.formule = 'Integral' );
```

- Jointure de la table « region » pour associer le nom de région à chaque contrat
- Filtrer les contrats de formules « integral » et les contrats de la région « Pays de la Loire »
- Compter les contrats distincts correspondants et afficher la région et la formule

	reg_nom	Formule	nombre
1	Pays de la Loire	Integral	589

5. Interrogation des tables par des requêtes SQL 3/4

Requête 8 : Lister les numéros de contrats avec le type de contrat et leur formule pour les maisons du département 71

```
SELECT r.dep_code,
       c.type_local,
       c.contrat_id,
       c.type_contrat,
       c.formule
FROM   contrat c
       INNER JOIN region r
           ON r.code_dep_code_commune = c.code_dep_code_commune
WHERE  ( r.dep_code = 71
        AND c.type_local = 'Maison' );
```

- Jointure de la table « region » pour associer le numéro de département à chaque ligne de la table « contrat »
- Filtrer les contrats du département 71 associés à des maisons
- Sélectionner les contrats distincts correspondants, et afficher le code département, le type de local, le type de contrat et leur formule

	dep_code	Type_local	contrat_Id	type_contrat	formule
1	71	Maison	114768	Residence principale	Integral
2	71	Maison	114779	Residence principale	Classique
3	71	Maison	114782	Residence principale	Classique
4	71	Maison	114812	Residence principale	Integral

Requête 9 : Quelle est la surface moyenne des contrats à Paris ?

```
SELECT r.dep_nom,
       Avg(surface) AS surface_moyenne
FROM   contrat c
       INNER JOIN region r
           ON r.code_dep_code_commune = c.code_dep_code_commune
WHERE  r.dep_nom = 'Paris';
```

- Jointure de la table « region » pour associer le nom département à chaque contrat
- Filtrer les contrats du département de Paris
- Afficher la moyenne de la surface des contrats correspondants et le nom du département

	dep_nom	surface_moyenne
1	Paris	51.76954988591579

Requête 10 : Classements des 10 départements où le prix moyen de la cotisation est le plus élevé

```
SELECT r.dep_code,
       r.dep_nom,
       Avg(c.prix_cotisation_mensuel) AS cotisation_moyenne
FROM   region r
       INNER JOIN contrat c
           ON r.code_dep_code_commune = c.code_dep_code_commune
GROUP BY r.dep_code,
         r.dep_nom
ORDER BY cotisation_moyenne DESC
LIMIT 10;
```

- Jointure de la table « contrat » (réduite aux données prix_cotisation_mensuel et code_dep_code_commune, avec contrats distincts) pour associer le nom et code de département à chaque contrat
- Grouper les contrats par département
- Calculer le prix moyen de cotisation pour chaque département
- Trier par ordre décroissant et afficher les 10 premières valeurs de cotisation moyenne, avec nom et numéro de département correspondant

	dep_code	dep_nom	cotisation_moyenne
1	75	Paris	36.39908732628086
2	92	Hauts-de-Seine	26.26935840707965
3	94	Val-de-Marne	19.82416396979504
4	78	Yvelines	18.88806970509383
5	69	Rhône	18.48551829268293
6	1	Ain	18.23888888888889
7	6	Alpes-Maritimes	18.13762287756926
8	17	Charente-Maritime	17.31730769230769
9	74	Haute-Savoie	17.14873417721519
10	2A	Corse-du-Sud	17.07438016528926

5. Interrogation des tables par des requêtes SQL 4/4

Requête 11 : Liste des communes ayant eu au moins 150 contrats

```
SELECT r.code_dep_code_commune,
       r.com_nom_maj_court,
       Count(DISTINCT( contrat_id )) AS nombre_contrats
FROM   contrat c
       INNER JOIN region r
           ON r.code_dep_code_commune = c.code_dep_code_commune
GROUP BY r.code_dep_code_commune,
         r.com_nom_maj_court
HAVING nombre_contrats >= 150
ORDER BY nombre_contrats DESC
```

- Jointure de la table « region » pour associer le nom et le code de commune à chaque contrat
- Grouper les contrats par commune et compter les éléments distincts
- Filtrer les communes dont le nombre de contrat est supérieur à 150
- Trier par ordre décroissant

	code dep code com	com nom maj coi	nombre contr
1	75118	PARIS 18	515
2	75117	PARIS 17	468
3	75115	PARIS 15	407
4	75116	PARIS 16	394
5	6088	NICE	387
6	75111	PARIS 11	381
7	33063	BORDEAUX	302
8	75120	PARIS 20	302
9	44109	NANTES	291
10	75119	PARIS 19	266
11	75110	PARIS 10	263
12	75112	PARIS 12	252
13	75114	PARIS 14	222
14	38185	GRENOBLE	220
15	75109	PARIS 9	204
16	31555	TOULOUSE	187
17	83137	TOULON	170
18	92026	COURBEVOIE	163
19	59350	LILLE	161
20	75103	PARIS 3	159

Requête 12 : Quel est le nombre de contrats pour chaque région ?

```
SELECT r.reg_code,
       r.reg_nom,
       Count(DISTINCT( contrat_id )) AS nombre_contrat
FROM   region r
       INNER JOIN contrat c
           ON r.code_dep_code_commune = c.code_dep_code_commune
GROUP BY r.reg_code,
         r.reg_nom;
```

- Jointure de la table « region » pour associer le nom et le code de région à chaque contrat
- Grouper les contrats par région et compter les éléments distincts
- Afficher pour chaque région son code, son nom et le nombre de contrats

	reg_code	reg_nom	nombre_contrat
1	2	Martinique	73
2	3	Guyane	37
3	4	La Réunion	17
4	11	Ile-de-France	14177
5	24	Centre-Val de Loire	598
6	27	Bourgogne-Franche-Comté	293
7	28	Normandie	824
8	32	Hauts-de-France	1189
9	44	Grand Est	769
10	52	Pays de la Loire	1196
11	53	Bretagne	947
12	75	Nouvelle-Aquitaine	2038
13	76	Occitanie	1609
14	84	Auvergne-Rhône-Alpes	3042
15	93	Provence-Alpes-Côte d'Azur	3279
16	94	Corse	247