

2024 | Par : Kévin Maure

# Projet 3 : Requêtez une base de données avec SQL

Présentation de la méthodologie



# Exploration des données

## Dictionnaire des données

	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	VARCHAR			Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ...
	Voie	VARCHAR			Libellé de la voie pour l'adresse du logement assuré
	Code_dep_code_commune	INT		Clé secondaire	Concaténation du code département et code commune pour avoir une clé unique
	Code_postal	INT	97470		Code postal pour l'adresse du logement assuré
	Commune	VARCHAR			Libellé de la commune de l'adresse du logement
	Code_departement	INT	974		Code du département de l'adresse du logement assuré
	Surface	INT			Surface du logement assuré
	Type_local	VACHAR	50		Type du logement (Appartement ou maison)
	Occupation	VACHAR	50		Par qui est occupé le logement assuré (propriétaire ou locataire)
	Type_contrat	VACHAR	50		Type de contrat d'assurance selon le bien est une résidence principale, une mise en location ou une résidence secondaire
	Formule	VACHAR	50		Formule du contrat d'assurance (Classique ou Integral)
REGION.CSV	Valeur_declaree_biens	VACHAR	50		La tranche de valeurs qui a été déclarée pour le logement assuré
	Prix_cotisation_mensuel	MONEY			Prix mensuel des cotisations pour l'assurance du logement
	Code_dep_code_commune	INT		Clé primaire	Concaténation du code département et code commune pour avoir une clé unique
	reg_code	INT	94		Code de la region
	reg_nom	VACHAR	50		Libellé de la region
	aca_nom	VACHAR	50		Libellé de la zone académique
	dep_nom	VACHAR	50		Libellé du département
	com_nom_maj_court	VACHAR	50		Libellé de la commune en majuscule
	dep_code	VACHAR	50		Code du département
	dep_nom_num	VACHAR	50		Libellé et code du département

Comprendre la  
signification des  
données

Contraintes:

-Taille

-NOT NULL

-DEFAULT

-Type de donnée

-CHECK

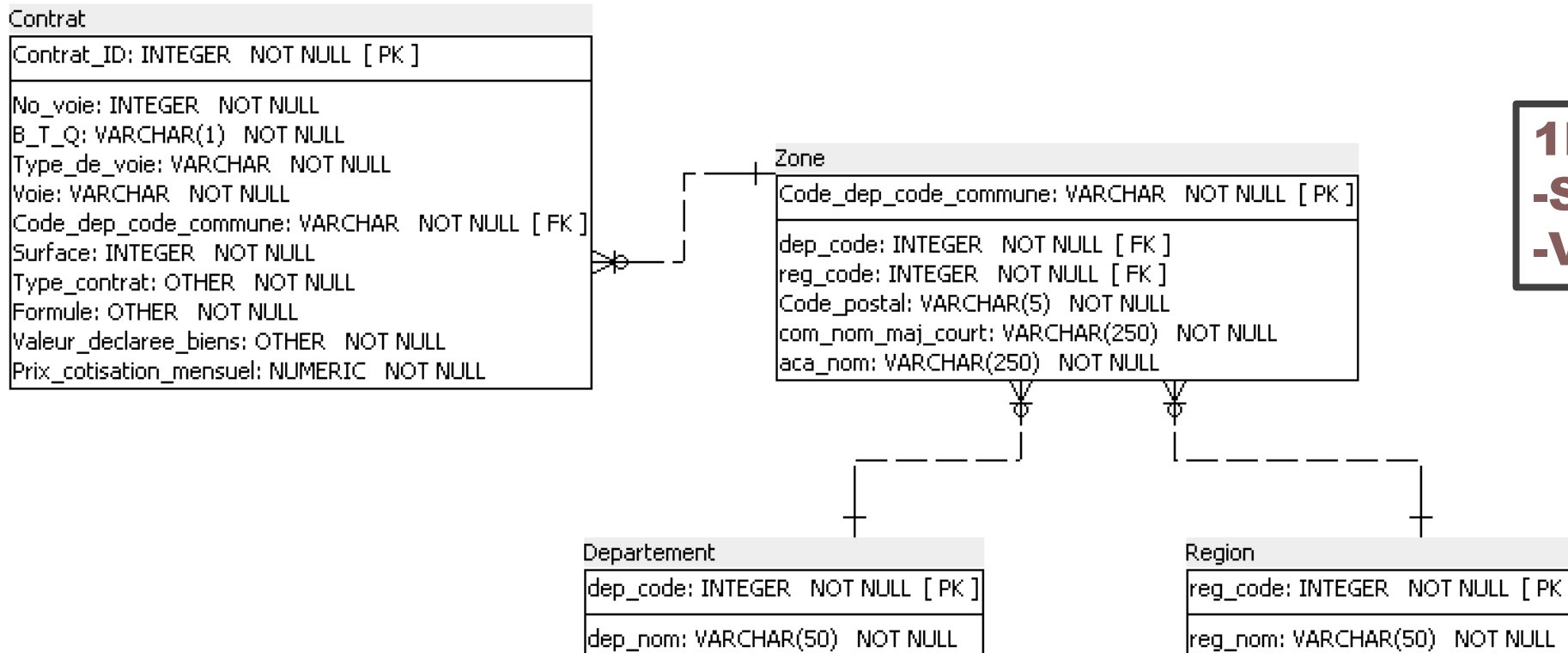
-UNIQUE

-PRIMARY KEY

-FOREIGN KEY

# Conception de schéma relationnel

Schéma relationnel normalisé en 3NF

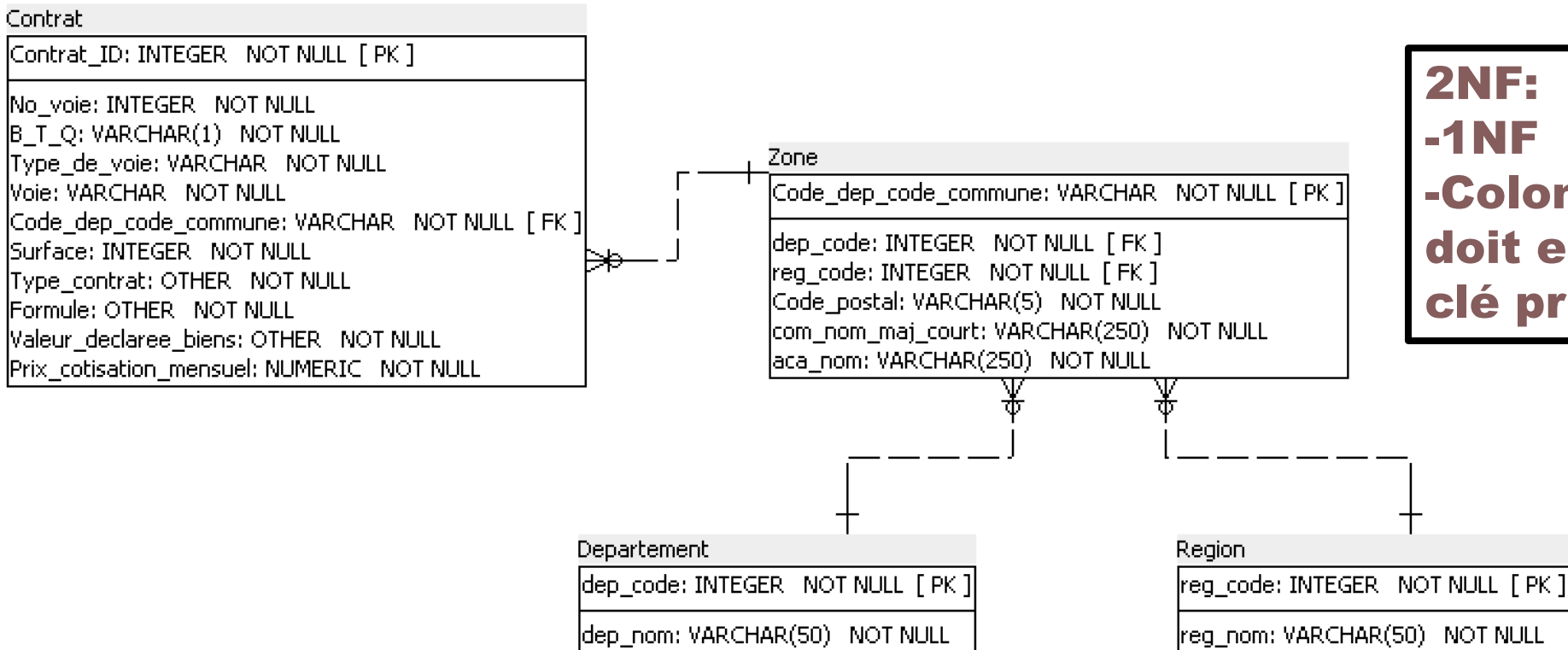


**1NF:**

**-Structure tabulaire**  
**-Valeurs atomiques**

# Conception de schéma relationnel

Schéma relationnel normalisé en 3NF



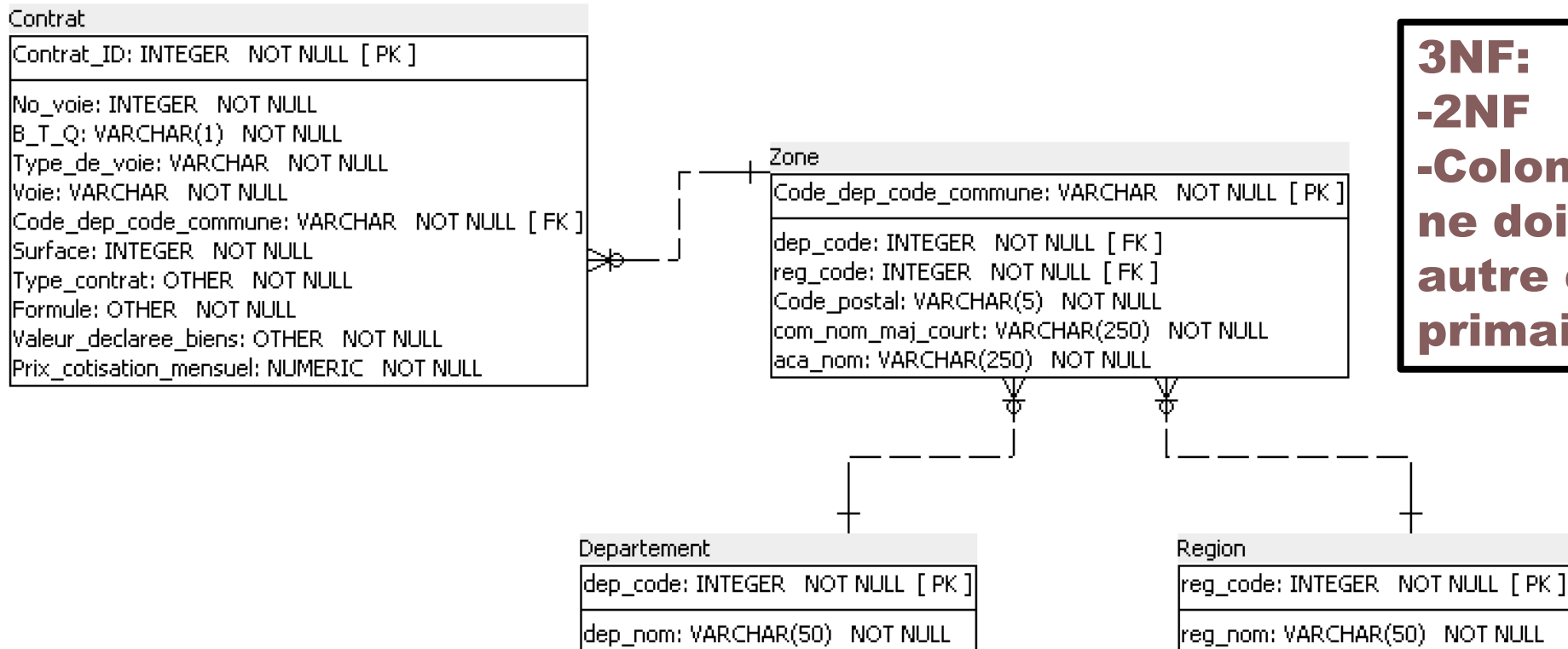
**2NF:**

**-1NF**

**-Colonne (hors clé primaire)  
doit est dépendant de la  
clé primaire**

# Conception de schéma relationnel

Schéma relationnel normalisé en 3NF



**3NF:**

**-2NF**

**-Colonne (hors clé primaire)  
ne doit pas dépendre d'une  
autre colonne (hors clé  
primaire)**

# Conception de schéma relationnel

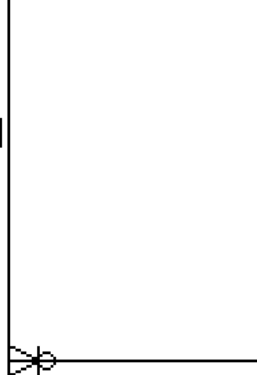
Schéma relationnel utilisé pour la suite du projet

Contrat

contrat_ID: INTEGER NOT NULL [ PK ]
no_voie: INTEGER NOT NULL
b_t_q: VARCHAR(1) NOT NULL
type_de_voie: VARCHAR NOT NULL
voie: VARCHAR NOT NULL
code_dep_code_commune: VARCHAR NOT NULL [ FK ]
code_postal: VARCHAR(5) NOT NULL
commune: VARCHAR NOT NULL
code_departement: INTEGER NOT NULL
surface: INTEGER NOT NULL
type_local: VARCHAR(50) NOT NULL
occupation: VARCHAR(50) NOT NULL
type_contrat: VARCHAR(50) NOT NULL
formule: VARCHAR(50) NOT NULL
valeur_declaree_biens: VARCHAR(50) NOT NULL
prix_cotisation_mensuel: NUMERIC NOT NULL

Region

code_dep_code_commune: VARCHAR NOT NULL [ PK ]
reg_code: INTEGER NOT NULL
reg_nom: VARCHAR(50) NOT NULL
aca_nom: VARCHAR(250) NOT NULL
dep_nom: VARCHAR(50) NOT NULL
com_nom_maj_court: VARCHAR(250) NOT NULL
dep_code: INTEGER NOT NULL
dep_nom_num: VARCHAR(50) NOT NULL



# Création de la base de données

Code SQL générant les tables de la base de données

**SGBD:**



```
CREATE TABLE Region (  
    code_dep_code_commune VARCHAR NOT NULL,  
    reg_code INT NOT NULL,  
    reg_nom VARCHAR(50) NOT NULL,  
    aca_nom VARCHAR(250) NOT NULL,  
    dep_nom VARCHAR(50) NOT NULL,  
    com_nom_maj_court VARCHAR(250) NOT NULL,  
    dep_code INT NOT NULL,  
    dep_nom_num VARCHAR(50) NOT NULL,  
    PRIMARY KEY (code_dep_code_commune)  
);
```

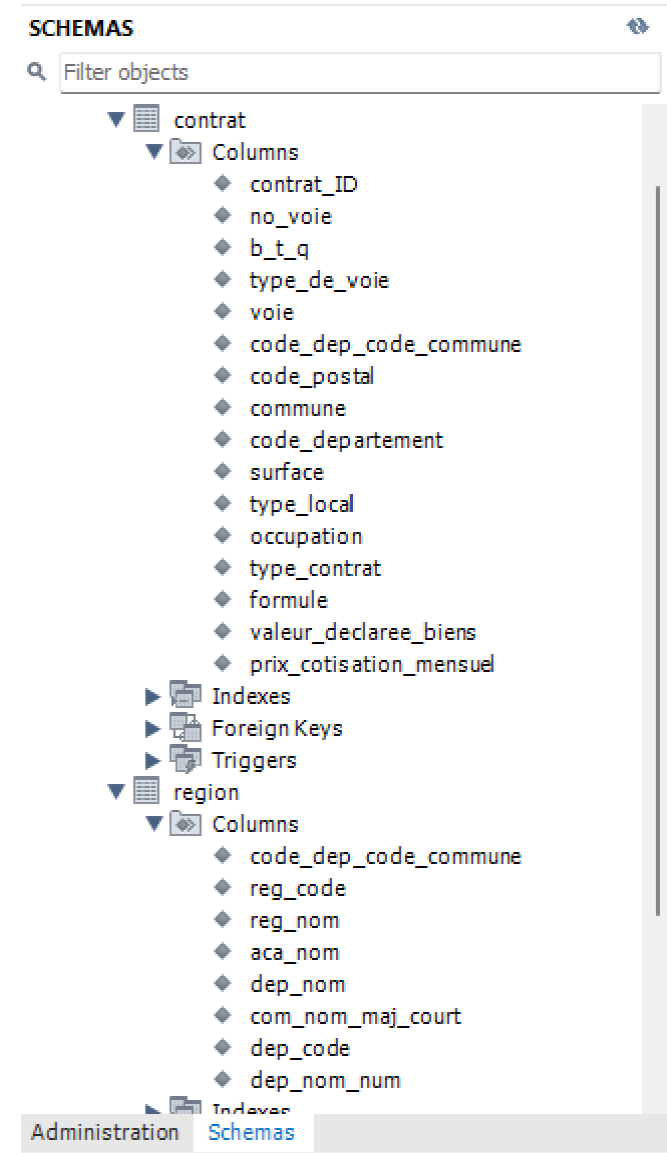
```
CREATE TABLE Contrat (  
    contrat_ID INT NOT NULL,  
    no_voie INT NOT NULL,  
    b_t_q VARCHAR(1) NOT NULL,  
    type_de_voie VARCHAR NOT NULL,  
    voie VARCHAR NOT NULL,  
    code_dep_code_commune VARCHAR NOT NULL,  
    code_postal VARCHAR(5) NOT NULL,  
    commune VARCHAR NOT NULL,  
    code_departement INT NOT NULL,  
    surface INT NOT NULL,  
    type_local VARCHAR(50) NOT NULL,  
    occupation VARCHAR(50) NOT NULL,  
    type_contrat VARCHAR(50) NOT NULL,  
    formule VARCHAR(50) NOT NULL,  
    valeur_declaree_biens VARCHAR(50) NOT NULL,  
    prix_cotisation_mensuel NUMERIC NOT NULL,  
    PRIMARY KEY (contrat_ID)  
);
```

```
ALTER TABLE Contrat ADD CONSTRAINT region_contrat_1_fk  
FOREIGN KEY (code_dep_code_commune)  
REFERENCES Region (code_dep_code_commune);
```

# Création de la base de données

Résultat du code SQL générant les tables de la base de données

**SGBD:**





# Importation des données dans la base de données

## Vérification de l'importation

contrat_ID	no_voie	b_t_q	type_de_voie	voie	code_dep_code_commune	code_postal	commune	code_departement
100601	190	A	RUE	CENTRALE	1350	1370	SAINT-ETIENNE-DU-BOIS	1
100602	347		RUE	DU CHATEAU	1103	1170	CHEVRY	1
100603	58		AV	DU MONT BLANC	1143	1220	DIVONNE-LES-BAINS	1
100604	140		RUE	DE L'ABBE JOLIVET	1288	1630	PERON	1
100605	39		RUE	BUFFON	1033	1200	VALSERHÔNE	1
100606	8		RUE	DE GENEVE	1354	1630	ST GENIS POUILLY	1
100607	2		RUE	DU RECULET	1354	1630	ST GENIS POUILLY	1
100608	1403		RUE	JEAN DE GINGINS	1143	1220	DIVONNE-LES-BAINS	1
100609	226		ALL	DES CAPUCINES	1354	1630	ST GENIS POUILLY	1
100610	276		RTE	DE POUIGNY	1288	1630	PERON	1
100611	79		CRS	DE VERDUN	1283	1100	OYONNAX	1
100612	240		RUE	DE PRE BAILLY	1173	1170	GEX	1

surface	type_local	occupation	type_contrat	formule	valeur_declaree_biens	prix_cotisation_mensuel
50	Appartement	Locataire	Residence principale	Integral	0-25000	25
48	Appartement	Locataire	Residence principale	Classique	0-25000	30
131	Appartement	Proprietaire	Residence principale	Integral	25000-50000	57
109	Maison	Locataire	Residence principale	Integral	25000-50000	43
109	Appartement	Locataire	Residence principale	Classique	0-25000	33
53	Appartement	Proprietaire	Residence principale	Classique	0-25000	19
59	Appartement	Proprietaire	Residence principale	Integral	0-25000	15
93	Maison	Proprietaire	Mise en location	Integral	25000-50000	34
117	Maison	Proprietaire	Residence principale	Classique	25000-50000	32
36	Appartement	Proprietaire	Residence principale	Integral	25000-50000	22
138	Appartement	Proprietaire	Residence secondaire	Classique	0-25000	11
45	Appartement	Locataire	Residence principale	Classique	0-25000	16

# Importation des données dans la base de données

Vérification de l'importation

**Nombre de lignes dans la table region**

```
66 • SELECT COUNT(code_dep_code_commune) FROM region;
```

Result Grid	Filter Rows:	Export:	Wrap Cell C
COUNT(code_dep_code_commune)			
▶ 38916			

**Nombre de lignes dans la table contrat**

```
66 • SELECT COUNT(contrat_ID) FROM contrat;
```

Result Grid	Filter Rows:	Export:
COUNT(contrat_ID)		
▶ 30335		

# Analyser les données grâce aux requêtes SQL

Requête 1 : *Lister les numéros de contrats (contrat\_ID) avec leur surface pour la commune de Caen.*

```
2      #Requête 1 : Lister les numéros de contrats (contrat_ID) avec leur surface pour la commune de Caen.  
3      • SELECT contrat_ID, surface  
4      FROM contrat  
5      WHERE commune LIKE "Caen";  
6  
7      /*
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |



	contrat_ID	surface
▶	103791	35
	103792	99
	103793	40
	103794	20
*	NULL	NULL

# Analyser les données grâce aux requêtes SQL

Requête 2 : *Lister les numéros de contrats avec le type de contrat et leur formule pour les maisons du départements 71*

#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
WHERE code_departement = 71;
```

Result Grid   Filter Rows: <input type="text"/>			
contrat_ID	type_contrat	formule	
114770	Residence principale	Classique	
114771	Residence principale	Integral	
114772	Residence principale	Classique	
114773	Residence principale	Integral	
114774	Residence principale	Classique	
114775	Residence principale	Integral	
114776	Residence principale	Classique	
114777	Residence principale	Integral	
114778	Residence principale	Classique	
114779	Residence principale	Classique	
114780	Residence principale	Integral	
114781	Residence secondaire	Integral	
114782	Residence principale	Classique	
114783	Mise en location	Integral	
114784	Residence principale	Classique	
114785	Residence principale	Integral	
114786	Residence principale	Classique	
114787	Residence secondaire	Classique	
114788	Residence principale	Classique	
114789	Residence principale	Classique	
114790	Residence principale	Integral	
114791	Residence principale	Integral	
114792	Mise en location	Integral	
114793	Residence principale	Classique	
114794	Residence secondaire	Classique	
114795	Residence secondaire	Integral	
114796	Residence principale	Classique	
114797	Residence principale	Classique	
114803	Residence principale	Integral	
114804	Residence secondaire	Classique	
114805	Mise en location	Integral	
114806	Residence principale	Classique	
114807	Residence principale	Integral	
114808	Residence principale	Integral	
114809	Residence principale	Classique	
114810	Residence principale	Integral	
114811	Residence principale	Integral	
114812	Residence principale	Integral	
114813	Mise en location	Classique	

# Analyser les données grâce aux requêtes SQL

Requête 3 : *Lister le nom des régions de France.*

```
#Requête 3 : Lister le nom des régions de France.  
SELECT DISTINCT reg_nom  
FROM region;
```

Result Grid

Filter Rows:

	reg_nom
▶	Grand Est
	Auvergne-Rhône-Alpes
	Occitanie
	Provence-Alpes-Côte d'Azur
	Normandie
	Nouvelle-Aquitaine
	Centre-Val de Loire
	Hauts-de-France
	Bourgogne-Franche-Comté
	Bretagne
	Corse
	Pays de la Loire
	Ile-de-France
	Guadeloupe
	Martinique
	Guyane
	La Réunion
	Collectivités d'outre-mer
	Mayotte

# Analyser les données grâce aux requêtes SQL

Requête 4 : *Combien existe-t-il de contrats sur les résidences principales ?*

```
17      #Requête 4 : Combien existe-t-il de contrats sur les résidences principales ?
18 •    SELECT COUNT(contrat_ID) AS NbContratResidencePrincipale
19      FROM contrat
20      WHERE type_contrat LIKE "Residence principale";
21
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	NbContratResidencePrincipale
▶	25620

# Analyser les données grâce aux requêtes SQL

Requête 5 : *Quelle est la surface moyenne des logements avec un contrat à Paris ?*

```
27      #Requête 5 : Quelle est la surface moyenne des logements avec un contrat à Paris ?
28 •    SELECT ROUND(AVG(surface),2) AS MoyenneSurfaceLogementParis
29      FROM contrat
30      WHERE code_postal LIKE "75___";
31
32
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	MoyenneSurfaceLogementParis
▶	51.78

# Analyser les données grâce aux requêtes SQL

Requête 6 : *Quels sont les 5 contrats qui ont les surfaces les plus élevées ?*

```
31      #Requête 6 : Quels sont les 5 contrats qui ont les surfaces les plus élevées ?
32      •  SELECT contrat_ID, surface
33      FROM contrat
34      ORDER BY surface DESC
35      LIMIT 5;
36
```



Result Grid			Filter Rows:	Edit:	Export/Import:	Wrap Cell Content
	contrat_ID	surface				
▶	104211	815				
	105463	742				
	130878	595				
	100822	570				
	109872	559				



# Analyser les données grâce aux requêtes SQL

Requête 7 : *Quel est le prix moyen de la cotisation mensuelle ?*

```
48      #Requête 7 : Quel est le prix moyen de la cotisation mensuelle ?
49 •    SELECT ROUND(AVG(prix_cotisation_mensuel),2) AS PrixMoyenCotisationMensuelle
50      FROM contrat;
51
52
53  ☹  /*
```

Result Grid |   Filter Rows:  | Export:  | Wrap Cell Content: 

	PrixMoyenCotisationMensuelle
▶	19.33

# Analyser les données grâce aux requêtes SQL

Requête 8 : *Quel est le nombre de contrats pour chaque catégorie de prix de la valeur déclarée des biens ?*

```
41  #Requête 8 : Quel est le nombre de contrats pour chaque catégorie de prix de la valeur déclarée des biens ?
42  • SELECT COUNT(contrat_ID) AS NbContrat, valeur_declaree_biens
43  FROM Contrat
44  GROUP BY valeur_declaree_biens;
45
46  /*
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	NbContrat	valeur_declaree_biens			
▶	22720	0-25000			
	6815	25000-50000			
	104	100000+			
	696	50000-100000			

# Analyser les données grâce aux requêtes SQL

Requête 9 : *Quel est le classement des 10 départements où le prix moyen de la cotisation est le plus élevé ?*

```
60  #Requête 9 : Quel est le classement des 10 départements où le prix moyen de la cotisation est le plus élevé ?
61  • SELECT ROUND(AVG(C.prix_cotisation_mensuel),2) AS PrixMoyenCotisationMensuelle, R.dep_nom
62  FROM Contrat C
63  JOIN Region R ON C.Code_dep_code_commune = R.Code_dep_code_commune
64  GROUP BY R.dep_nom
65  ORDER BY PrixMoyenCotisationMensuelle DESC
66  LIMIT 10;
67
68  /*
```

	PrixMoyenCotisationMensuelle	dep_nom
►	35.34	Paris
	26.49	Hauts-de-Seine
	20.00	Allier
	19.73	Val-de-Marne
	18.96	Rhône
	18.18	Yvelines
	18.12	Alpes-Maritimes
	17.95	Ain
	17.30	Charente-Maritime
	17.25	La Réunion

# Analyser les données grâce aux requêtes SQL

Requête 10 : *Quel est le nombre de contrats avec des formules « Intégral » pour la région Pays de la Loire ?*

```
77      #Requête 10 avec la condition "Intégral"
78 •    SELECT COUNT(contrat_ID) AS NbContrat
79      FROM Contrat
80      WHERE
81      formule LIKE "Integral" AND code_postal LIKE "53__"
82      OR formule LIKE "Integral" AND code_postal LIKE "44__"
83      OR formule LIKE "Integral" AND code_postal LIKE "49__"
84      OR formule LIKE "Integral" AND code_postal LIKE "72__"
85      OR formule LIKE "Integral" AND code_postal LIKE "85__"
86      ;
```

Result Grid		 Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	NbContrat			
▶	590			

# Analyser les données grâce aux requêtes SQL

Requête 11 : *Quelle est la liste des communes ayant au moins 150 contrats ?*

```
89  #Requête 11 : Quelle est la liste des communes ayant au moins 150 contrats ?
90  •  SELECT R.com_nom_maj_court
91     FROM region R
92     JOIN contrat C ON R.Code_dep_code_commune = C.Code_dep_code_commune
93     GROUP BY R.com_nom_maj_court
94     HAVING COUNT(C.contrat_ID) >=150
95     ORDER BY R.com_nom_maj_court;
96
```

Result Grid     Filter Row	
	com_nom_maj_court
▶	BORDEAUX
	COURBEVOIE
	GRENOBLE
	LILLE
	NANTES
	NICE
	PARIS 3
	PARIS 9
	PARIS 10
	PARIS 11
	PARIS 12
	PARIS 14
	PARIS 15
	PARIS 16
	PARIS 17
Result 49 ×	
	PARIS 17
	PARIS 18
	PARIS 19
	PARIS 20
	TOULON
	TOULOUSE
Result 49 ×	

# Analyser les données grâce aux requêtes SQL

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

```
67 #Requête 12 : Quel est le nombre de contrats pour chaque région ?
68 • SELECT COUNT(C.contrat_ID) AS NbContrat, R.reg_nom
69 FROM Contrat C
70 RIGHT JOIN Region R ON C.code_dep_code_commune = R.code_dep_code_commune
71 GROUP BY R.reg_nom;
72
```

Result Grid		Filter Rows:
	NbContrat	reg_nom
▶	806	Grand Est
	2972	Auvergne-Rhône-Alpes
	1837	Occitanie
	3287	Provence-Alpes-Côte d'Azur
	898	Normandie
	2097	Nouvelle-Aquitaine
	804	Centre-Val de Loire
	1327	Hauts-de-France
	402	Bourgogne-Franche-Comté
	945	Bretagne
	247	Corse
	1138	Pays de la Loire
	13474	Ile-de-France
	0	Guadeloupe
	60	Martinique
	37	Guyane
	4	La Réunion
	0	Collectivités d'outre-mer
	0	Mayotte