

Projet 3

OpenClassrooms

Méthodologie

Samuel OSENAT - Octobre 2024



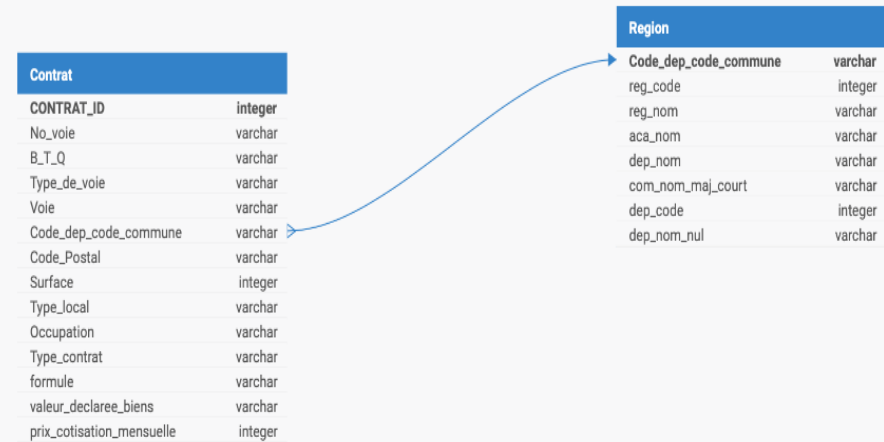
Dictionnaire de données

- Nous avons commencé par prendre connaissances des données en créant un dictionnaire de données recensant les différentes variables des tables, avec leur type et leurs potentielles contraintes.
- On a pu identifier les clés primaires et étrangères

	Nom des colonnes	Type de données	Taille	Clé	Description
CONTRAT.CSV	Contrat_ID	INT	6	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	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	VARCHAR			Code postal pour l'adresse du logement assuré
	Surface	INT			Surface du logement
	Type_local	VARCHAR			type de logement (appartement ou maison)
	Occupation	VARCHAR			Par qui le logement est occupé (locataire ou propriétaire)
	Type_contrat	VARCHAR			quelle est l'utilisation du logement (mise en location, résidence principale ou secondaire)
	Formule	VARCHAR			Quelle formule est utilisée par l'occupant
	valeur_declaree_biens	VARCHAR			A quelle tranche de prix le bien est valorisé
REGION.CSV	prix_cotisation_mensuel	INT			montant de la cotisation mensuelle
	Code_dep_code_commune	VARCHAR		Clé primaire	Concaténation du code département et code commune pour avoir une clé unique
	reg_code	INT			Code Insee de la région du logement
	reg_nom	VARCHAR			Nom de la région
	aca_nom	VARCHAR			Nom de l'académie
	dep_nom	VARCHAR			Nom du département
	com_nom_maj_court	VARCHAR			Nom de la commune en majuscule
	dep_code	INT			Code insee du département
	dep_nom_num	VARCHAR			Nom du département + numéro du département

Schéma relationnel

- Ensuite, grâce au dictionnaire de données, nous avons créé un modèle relationne entre les tables



Generated with app.erdlab.io

Base de données

- L'étape suivante consistait en la création de la base de données avec les deux tables contrat et région :

Data Output

Messages

Notifications

Add row

	contrat_id integer	no_voie integer	b_t_q character varying	type_de_voie character varying	voie character varying	code_dep_code_commune character varying	code_pc character
1	100601	190	A	RUE	CENTRALE	1350	1370
2	100602	347	[null]	RUE	DU CHATEAU	1103	1170
3	100603	58	[null]	AV	DU MONT BLANC	1143	1220
4	100604	140	[null]	RUE	DE L'ABBE JOLIVET	1288	1630
5	100605	39	[null]	RUE	BUFFON	1033	1200
6	100606	8	[null]	RUE	DE GENEVE	1354	1630
7	100607	2	[null]	RUE	DU RECULET	1354	1630
8	100608	1403	[null]	RUE	JEAN DE GINGINS	1143	1220

Total rows: 1000 of 30335

Query complete 00:00:00.206

Ln 20, Col

Data Output

Messages

Notifications

	code_dep_code_commune [PK] character varying (255)	reg_code character varying (255)	reg_nom character varying (255)	aca_nom character varying (255)	dep_nom character varying (255)	comm charac
1	1001	84	Auvergne-Rhône-Alpes	Lyon	Ain	L ABE
2	1002	84	Auvergne-Rhône-Alpes	Lyon	Ain	L ABE
3	1003	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMAF
4	1004	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBE
5	1005	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBE
6	1006	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBL
7	1007	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBR
8	1008	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBL

Total rows: 1000 of 38916

Query complete 00:00:00.205

Ln 33, Col

Requêtes :

Requête 1 :

```
4
5 SELECT contrat_id, surface
6 FROM contrat
7 WHERE code_postal = '14000'
```

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

Requête 2 :

```
4
5 SELECT contrat_id, type_contrat, formule
6 FROM contrat
7 WHERE code_dep_code_commune LIKE '71%' AND type_local='Maison';
```

Data Output				Messages	Notifications
	contrat_id [PK] integer	type_contrat character varying (255)	formule character varying (255)		
1	102358	Residence principale	Classique		
2	102385	Residence secondaire	Integral		
3	102395	Residence principale	Classique		
4	102411	Residence principale	Integral		
5	114768	Residence principale	Integral		
6	114779	Residence principale	Classique		
7	114782	Residence principale	Classique		
8	114812	Residence principale	Integral		

Requêtes :

Requête 3 :

```
9  select distinct reg_nom as "Nom des régions"
10 from region
11 ;
```

Data Output		Messages	Notifications
	Nom des régions character varying (255) 🔒		
1	Hauts-de-France		
2	La Réunion		
3	Bretagne		
4	Mayotte		
5	Auvergne-Rhône-Alpes		
6	Ile-de-France		
7	Grand Est		
8	Collectivités d'outre-mer		

Requête 4 :

```
13 SELECT contrat_id, surface
14 FROM contrat
15 ORDER BY surface desc
16 LIMIT 5;
```

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

Requêtes :

Requête 5 :

```
17
18 SELECT AVG(prix_cotisation_mensuel) as "Cotisation mensuelle moyenne"
19 FROM contrat
20 ;
```

Data Output		Messages	Notifications
	Cotisation mensuelle moyenne		
	numeric		
1	19.328696225482116		

Requête 6 :

```
22 SELECT -- requete 6
23 valeur_declaree_biens,
24 COUNT(contrat_id) AS "Nombre de contrat"
25 FROM contrat
26 GROUP BY valeur_declaree_biens;
```

Data Output		Messages	Notifications
	valeur_declaree_biens		Nombre de contrat
	character varying (255)		bigint
1	50000-100000		696
2	100000+		104
3	25000-50000		6815
4	0-25000		22720

Requêtes :

Requête 7 :

```
41 WITH
42   reg_loire AS(
43     SELECT code_dep_code_commune, reg_nom
44     FROM region
45     WHERE reg_code = '52'
46   )
47 SELECT r."reg_nom", formule ,COUNT(formule) AS Nombre_formule
48 FROM contrat c
49 INNER JOIN reg_loire r ON c.code_dep_code_commune=r.code_dep_code_commune
50 GROUP BY reg_nom, formule ;
51
```

	reg_nom character varying (255)	formule character varying (255)	nombre_formule bigint
1	Pays de la Loire	Classique	607
2	Pays de la Loire	Integral	589

Requête 8 :

```
11 --Requete 8
12 SELECT c.contrat_id, c.type_contrat, c.formule, r.dep_code
13 FROM contrat c
14 INNER JOIN region r ON c.code_dep_code_commune=r.code_dep_code_commune
15 WHERE dep_code = '71' AND type_local = 'Maison' ;
16
```

	contrat_id integer	type_contrat character varying (255)	formule character varying (255)	dep_code character varying (255)
1	114768	Residence principale	Integral	71
2	114779	Residence principale	Classique	71
3	114782	Residence principale	Classique	71
4	114812	Residence principale	Integral	71

Requêtes :

Requête 9 :

```
67 --Requete 9
68 SELECT r.dep_nom, ROUND(AVG(Surface),2)AS "Surface Moyenne"
69 FROM contrat c
70 INNER JOIN region r ON c.code_dep_code_commune=r.code_dep_code_commune
71 WHERE dep_code = '75'
72 GROUP BY dep_nom;
73
```

Data Output			Messages	Notifications
	dep_nom character varying (255)	Surface Moyenne numeric		
1	Paris	51.77		

Requête 10 :

```
74 -- Requete 10
75 SELECT r.dep_nom AS "Nom Département", ROUND(AVG(c.prix_cotisation_mensuel),2)AS "Prix moyen"
76 FROM contrat c
77 INNER JOIN region r ON c.code_dep_code_commune=r.code_dep_code_commune
78 GROUP BY "Nom Département"
79 ORDER BY "Prix moyen" desc
80 LIMIT 10;
```

Data Output			Messages	Notifications
	Nom Département character varying (255)	Prix moyen numeric		
1	Paris	36.4		
2	Hauts-de-Seine	26.27		
3	Val-de-Marne	19.82		
4	Yvelines	18.89		
5	Rhône	18.49		
6	Ain	18.24		
7	Alpes-Maritimes	18.14		
8	Charente-Maritime	17.32		

Total rows: 10 of 10 Query complete 00:00:00.344 Ln

Requêtes :

Requête 11 :

```
1  --Requete 11
2  SELECT r.comm_nom_maj_court AS "Nom commune", COUNT(c.contrat_id) as nombre_contrat
3  FROM contrat c
4  JOIN region r
5  ON r.code_dep_code_commune=c.code_dep_code_commune
6  GROUP BY r.comm_nom_maj_court
7  HAVING COUNT(c.contrat_id) >= 150
8  ORDER BY nombre_contrat DESC
9  ;
```

Data Output		Messages	Notifications
	Nom commune character varying (255)	nombre_contrat bigint	
1	PARIS 18	515	
2	PARIS 17	468	
3	PARIS 15	407	
4	PARIS 16	394	
5	NICE	387	
6	PARIS 11	381	
7	BORDEAUX	302	
8	PARIS 20	302	
9	NANTES	291	
10	PARIS 19	266	
Total rows: 20 of 20		Query complete 00:00:00.870	Ln 8,1

Requête 12 :

```
94  --Requete 12
95  SELECT r.reg_nom AS "Nom region", COUNT(contrat_id) as Nombre_contrat
96  FROM contrat c
97  INNER JOIN region r ON c.code_dep_code_commune=r.code_dep_code_commune
98  GROUP BY "Nom region"
99  ORDER BY Nombre_contrat DESC
100 ;
```

Data Output		Messages	Notifications
	Nom region character varying (255)	nombre_contrat bigint	
1	Ile-de-France	14177	
2	Provence-Alpes-Côte d'Azur	3279	
3	Auvergne-Rhône-Alpes	3042	
4	Nouvelle-Aquitaine	2038	
5	Occitanie	1609	
6	Pays de la Loire	1196	
7	Hauts-de-France	1189	
8	Bretagne	947	
9	Normandie	824	
10	Grand Est	769	
11	Centre-Val de Loire	598	
Total rows: 16 of 16		Query complete 00:00:00.323	