

DATAImmo

Modèle de prévision de prix immobiliers



Laplace Immo



Objectifs

Un projet stratégique pour
Laplace Immo



BASE DE DONNÉES NATIONALE

Créer un système d'informations à l'échelle nationale regroupant l'ensemble des transactions immobilières effectuées sur le territoire français.

MÉTHODOLOGIE

- Définition du dictionnaire de données, du diagramme de classes et du modèle physique de données (MPD)
- Traitement et intégration des données DVF

RÉSULTATS

A travers quelques requêtes "exemples, possibilités

DVF - Une base de données nationale

Source d'informations

Propos liminaires (*)

Conformément au décret n° 2018-1350 du 28 décembre 2018 relatif à la publication sous forme électronique des informations portant sur les valeurs foncières déclarées à l'occasion des mutations immobilières, le présent fichier DVF est désormais disponible en open data.

Qu'est-ce que DVF ? (*)

Le présent jeu de données « Demandes de valeurs foncières », publié et produit par la direction générale des finances publiques, permet de connaître les transactions immobilières intervenues au cours des cinq dernières années sur le territoire métropolitain et les DOM-TOM, à l'exception de l'Alsace, de la Moselle et de Mayotte. Les données contenues sont issues des actes notariés et des informations cadastrales.

* Source : DVF

<https://www.data.gouv.fr/fr/datasets/demandes-de-valeurs-foncieres/>



data.gouv.fr

Données

Réutilisations

Organisations

Actualités

A propos

Nous contacter

Accueil > Jeux de données > DVF

Demandes de valeurs foncières DVF

Mis à jour le 20 octobre 2021 — [Licence Ouverte / Open Licence version 2.0](#)

Producteur

Métadonnées

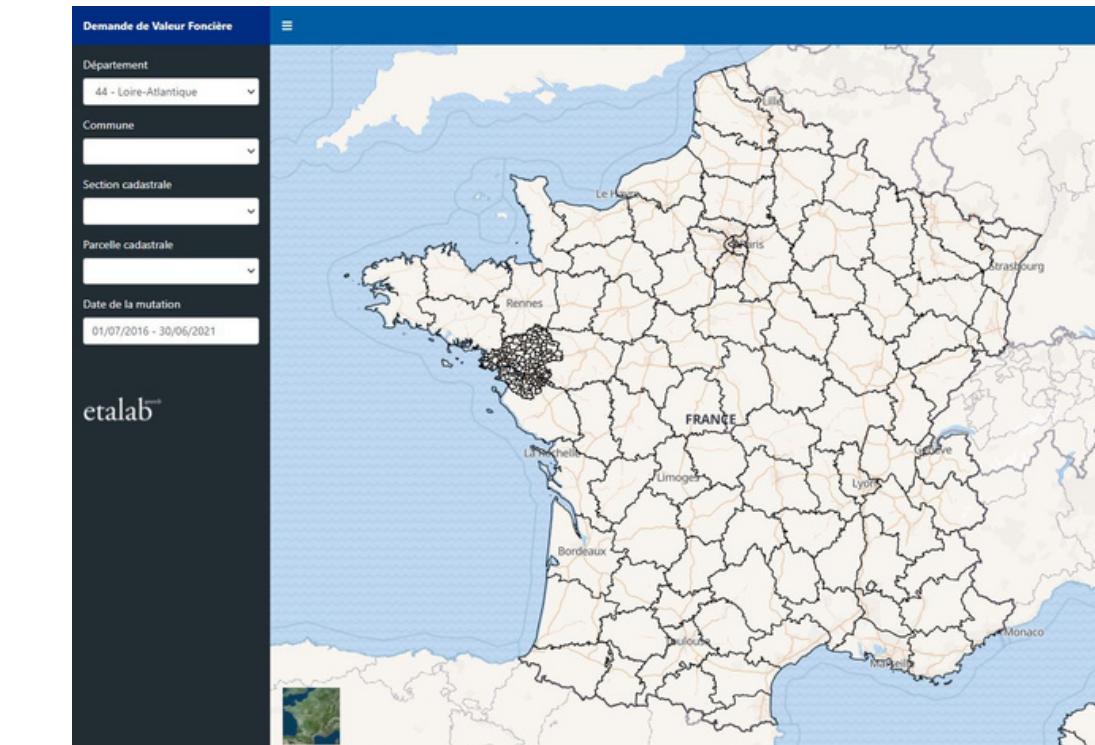
Actions



[Ministère de l'économie, des finances et de la relance](#)

Le ministère de l'économie, des finances et de la relance est chargé de mettre en œuvre la politique du Gouvernement en matière économique, financière, budgétaire et fiscale. Ses missions recouvrent des champs d'action variés tels que l'industrie, le commerce, les services, l'innovation mais aussi...

440 jeux de données 1 réutilisations



Méthodologie 1/4

Sélection des données

DAN-P3-data.xlsx

	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Date mutation	Nature mutation	Valeur fonciere	No voie	B/T/Q	Code type de voie	Type de voie	Code voie	Voie	Code ID commune	Code postal	Commune	Code département	Code commune
2	2020-01-03 00:00:00	C Vente	56000	190 A		0 RUE	5 CENTRALE	0 1370.0	SAINTE-ETIENNE-DU-BOIS	1	350			
3	2020-01-02 00:00:00	C Vente	165000	347		0 RUE	20 DU CHATEAU	1 1170.0	CHEVRY	1	103			
4	2020-01-08 00:00:00	C Vente	720000	58		1 AV	527 DU MONT BLANC	2 1220.0	DIVONNE-LES-BAINS	1	143			
5	2020-01-06 00:00:00	C Vente	429250	140		0 RUE	2 DE L'ABBE JOUVET	3 1630.0	PERON	1	288			
6	2020-01-07 00:00:00	C Vente	220900	39		0 RUE	110 BUFFON	4 1200.0	VALSERHONE	1	33			
7	2020-01-21 00:00:00	C Vente	42000	28		1 AV	179 JEAN FALCONNIER	5 1350.0	CULOZ	1	138			
8	2020-01-07 00:00:00	C Vente	262000	8		0 RUE	120 DU GENOVE	6 1630.0	ST-GENIS-POUILLY	1	354			
9	2020-01-08 00:00:00	C Vente	190000	2		0 RUE	210 DU RECULET	6 1630.0	ST-GENIS-POUILLY	1	354			
10	2020-01-16 00:00:00	C Vente	563130	1403		0 RUE	465 JEAN DE GINGINS	2 1220.0	DIVONNE-LES-BAINS	1	143			
11	2020-01-17 00:00:00	C Vente	535000	226		2 ALL	36 DES CAPUCINES	6 1630.0	ST-GENIS-POUILLY	1	354			
12	2020-01-16 00:00:00	C Vente	330000	276		3 RTE	210 DE POUIGNY	3 1630.0	PERON	1	288			
13	2020-01-27 00:00:00	C Vente	110600	79		4 CR	1490 DE VERDUN	7 1100.0	OYONNAZ	1	283			
14	2020-01-30 00:00:00	C Vente	50000	77 B		0 RUE	20 DU COMMERCE	8 1130.0	ST-GERMAIN-DE-JOUX	1	357			
15	2020-01-09 00:00:00	C Vente	212000	240		0 RUE	650 DE PRE BAILLY	9 1170.0	GEX	1	173			
16	2020-01-15 00:00:00	C Vente	160000	3		0 RUE	541 TURENN	4 1200.0	VALSERHONE	1	33			
17	2020-01-20 00:00:00	C Vente	561550	44		2 ALL	695 DU SQUARE DE LAUSANNE	2 1220.0	DIVONNE-LES-BAINS	1	143			
18	2020-01-28 00:00:00	C Vente	358000	59		0 RUE	40 ALEXANDRE BERARD	10 1500.0	AMBERIEU-EN-BUGEY	1	4			
19	2020-01-22 00:00:00	C Vente	220000	282		5 CHE	130 DES LONGES RAVES	11 1170.0	CESSY	1	71			
20	2020-01-27 00:00:00	C Vente	79300	54		6 GR	45 GRANDE RUE	12 1150.0	SAULT-BRENAT	1	396			
21	2020-01-06 00:00:00	C Vente	285000	1		0 RUE	160 DE GEX	13 1210.0	FERNEY-VOLTAIRE	1	160			
22	2020-01-27 00:00:00	C Vente	123100	17		7 LOT	A036 LES JARDINS DE CHEVRY	1 1170.0	CHEVRY	1	103			
23	2020-01-16 00:00:00	C Vente	107500	329		0 RUE	25 DES CAR	14 1170.0	SEGNY	1	399			
24	2020-01-30 00:00:00	C Vente	128810	369		1 AV	55 CHATEAU D'ALLE	15 1150.0	LAGNIEU	1	202			
25	2020-02-05 00:00:00	C Vente	91000	154		0 RUE	27	16 1130.0	LE POIZAT-LALLEYRIAT	1	204			
26	2020-01-21 00:00:00	C Vente	125000	6		0 RUE	74	9 1170.0	GEX	1	173			

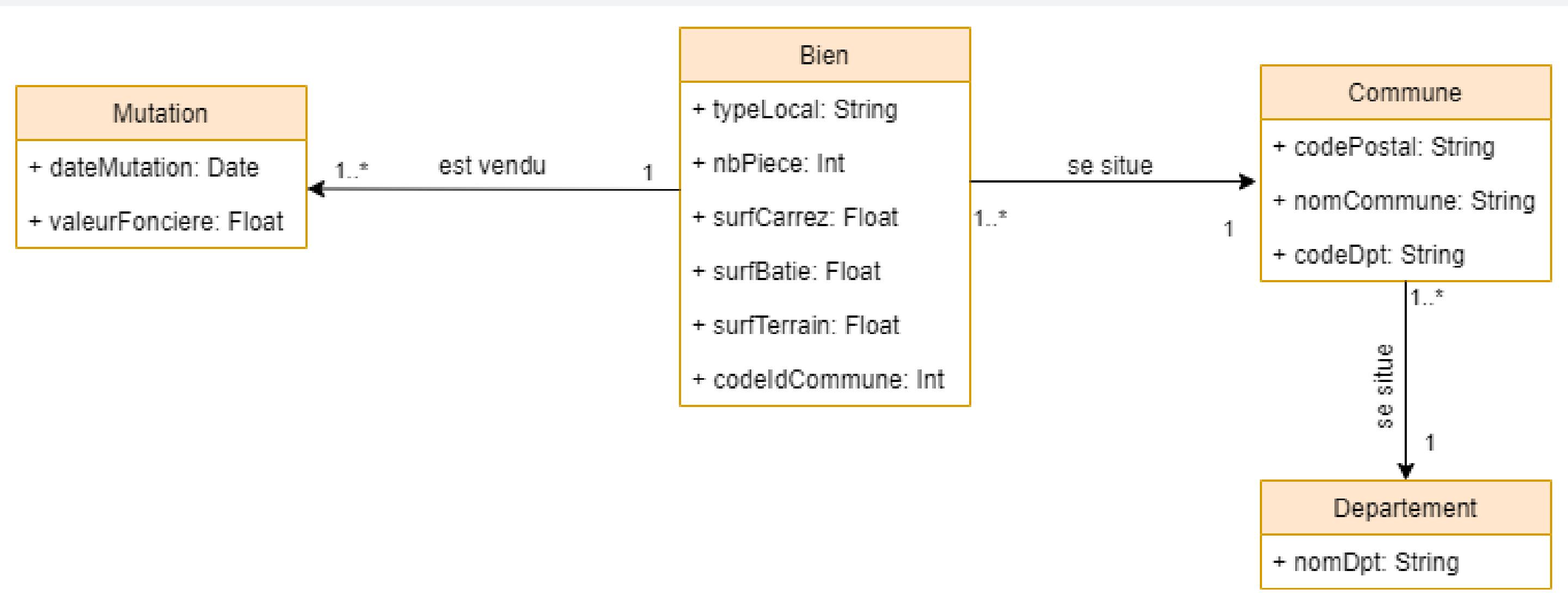
- Choix de données non exhaustif
- Nb colonnes, nb lignes, semestre, POC...
- Vision "macro" à l'échelle de la commune
- Ensemble des données à retenir pour notre modèle (cf. en annexe)

Numéro	Code propriété	Signification	Type	Observation
1	dat_mut	Date de la vente	Date	Format AAAA-MM-JJ
2	prix_mut	Montant de la vente	Numérique	Float
3	code_id_commune	Code INSEE de la commune	Numérique	Integer
4	code_postal	Code postal	Texte	Longueur : 5
5	nom_commune	Nom de la commune	Texte	Longueur : 50
6	code_dpt	Numéro du département / référence cadastrale	Alphanumérique	Longueur : 3
7	type_local	Typologie du bien immobilier (appartement, maison...)	Texte	Longueur : 20
8	nb_piece	Nombre de pièces principales	Numérique	Integer
9	id	Identifiant composite du bien créé par concaténation des données cadastrales + N° 1er lot	Alphanumérique	Longueur : 15

Méthodologie 2/4

Diagramme de classes (UML)

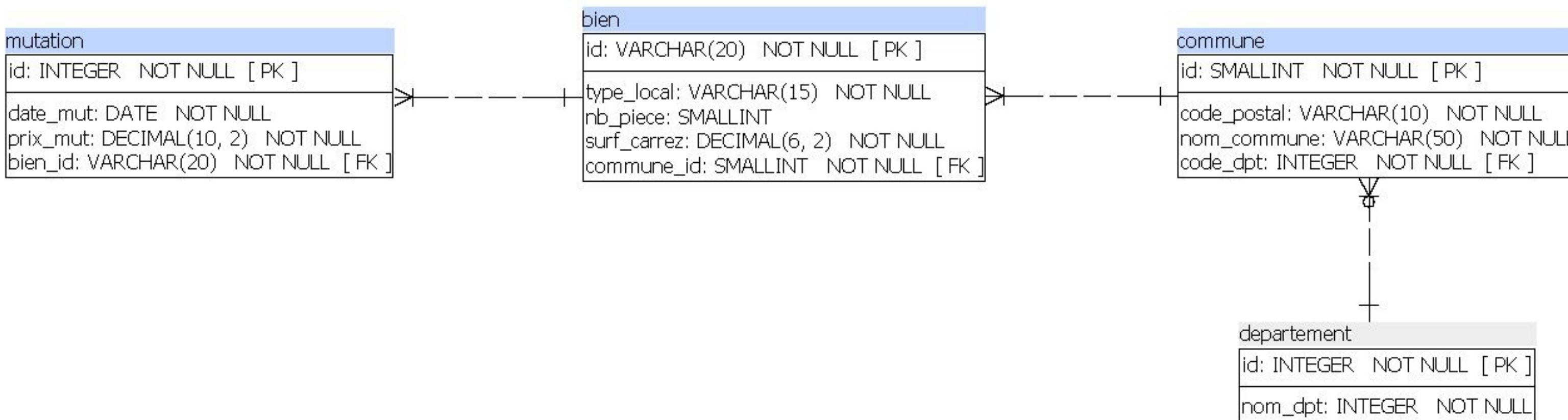
Un diagramme de classe de type UML décrit un système en visualisant les différents types d'objets au sein d'un système et les types de relations statiques qui existent entre eux. Il illustre également les opérations et les attributs des classes.



Méthodologie 3/4

Modèle Physique de Données (MPD)

Le modèle physique des données (MPD) d'une base de données permet d'avoir une représentation graphique de la structure d'une base de données et ainsi de mieux comprendre les relations entre les différents tables. Il permet d'avoir un point de vue global sur l'ensemble de la base de données.



Méthodologie 4/4

Création de la base de données

Importation des données



Création de la base de données à partir du MPD via SQL Power Architect



Import du dataframe dans un Jupyter Notebook



Traitement des données (outliers, valeurs nulles...)

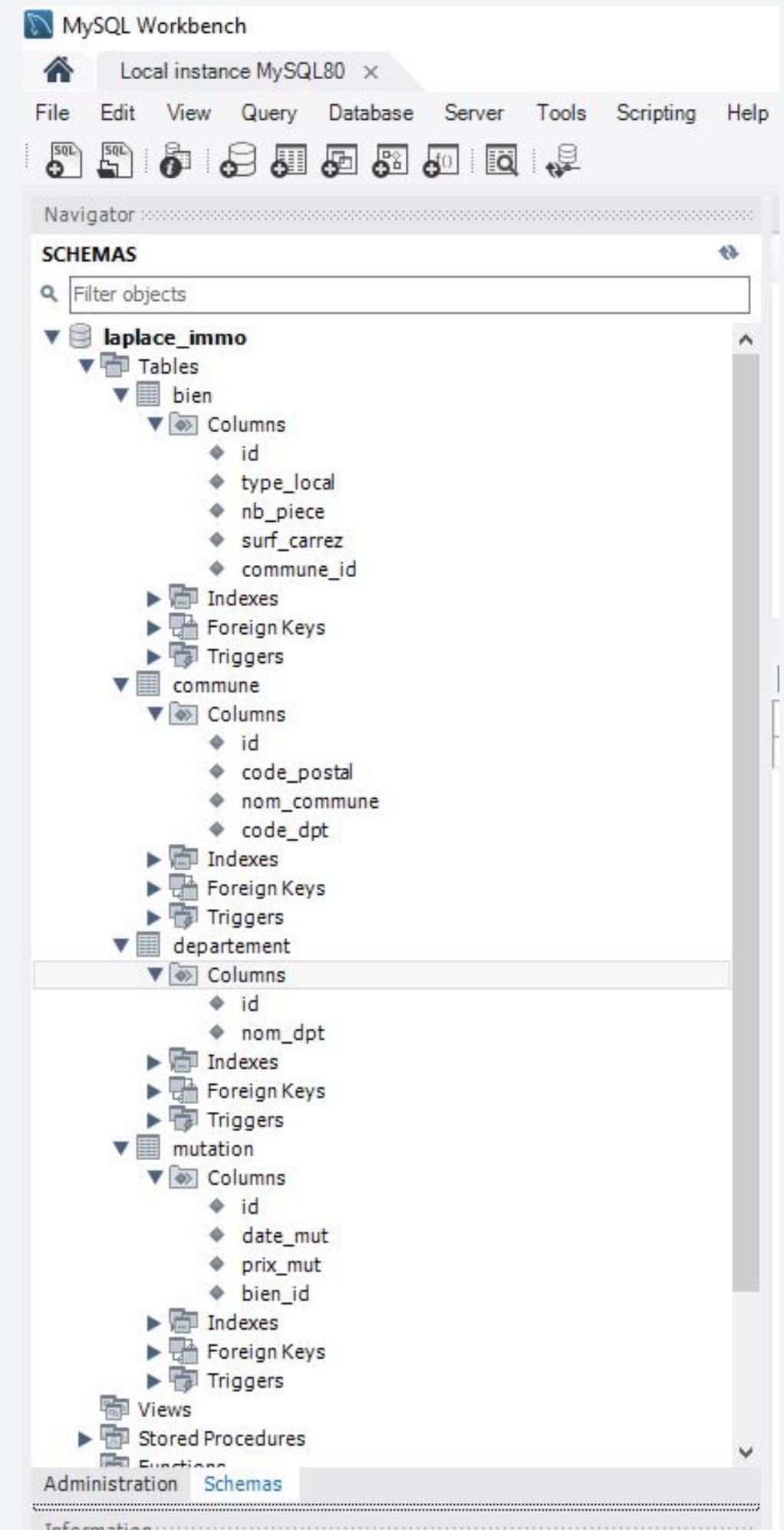


Création des fichiers csv et export pour intégration dans la base de données

Résultats

Création de la base de données

SGBD-R utilisé : MySQL



Résultats - exemple 1/9

The screenshot shows a MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a text area containing a SQL query:

```
1 -- Nombre total d'appartements vendus au 1er semestre 2020
2
3 • select COUNT(mutation.id) AS nb_vente_appart
4 FROM mutation
5 LEFT JOIN bien
6 ON mutation.bien_id = bien.id
7 WHERE bien.type_Local = "Appartement"
8 AND mutation.date_mut BETWEEN "2020-01-01" AND "2020-06-30";
```

At the bottom, there's a results grid labeled "Result Grid". It has one row with one column. The column header is "nb_vente_appart" and the value is "31362".

nb_vente_appart
31362

31362 appartements ont été vendus au 1er semestre 2020

Résultats - exemple 2/9

```
1 -- Proportion des ventes d'appartements par le nombre de pièces.  
2  
3 • SET @req = ( -- Nombre total d'appartements vendus  
4     SELECT COUNT(mutation.id)  
5     FROM mutation  
6     JOIN bien ON bien.id = mutation.bien_id  
7     WHERE type_local = "Appartement"  
8     AND type_local = "Appartement");  
9  
10 • SELECT nb_piece, COUNT(mutation.id) as nb_ventes, COUNT(mutation.id) / @req * 100 AS proportion_vente  
11    FROM mutation  
12    JOIN bien ON bien.id = mutation.bien_id  
13    WHERE type_local = "Appartement"  
14    GROUP BY nb_piece  
15    ORDER BY nb_piece  
16
```

Result Grid			
	nb_piece	nb_ventes	proportion_vente
▶	0	30	0.0957
	1	6736	21.4782
	2	9773	31.1619
	3	8966	28.5887
	4	4458	14.2147
	5	1114	3.5521
	6	203	0.6473
	7	54	0.1722
	8	17	0.0542
	9	8	0.0255
	10	2	0.0064
	11	1	0.0032

Proportion des ventes
d'appartements Par
nombre de pièces

Résultats - exemple 3/9

```
1  -- Liste des 10 départements où le prix du mètre carré est le plus élevé.  
2  
3 • SELECT code_dpt, nom_dpt, ROUND(SUM(prix_mut) / SUM(surf_carrez), 2) AS prix_m2  
4   FROM mutation  
5   JOIN bien ON mutation.bien_id = bien.id  
6   JOIN commune ON bien.commune_id = commune.id  
7   JOIN departement ON departement.id = commune.code_dpt  
8   GROUP BY code_dpt  
9   ORDER BY prix_m2 DESC  
10  LIMIT 10;
```

Result Grid			
	code_dpt	nom_dpt	prix_m2
▶	75	Paris	11912.46
	92	Hauts-de-Seine	7238.19
	94	Val-de-Marne	4832.75
	6	Alpes-Maritimes	4591.20
	74	Haute-Savoie	4159.22
	93	Seine-Saint-Denis	4074.47
	78	Yvelines	4002.16
	69	Rhône	3893.53
	2A	Corse-du-Sud	3764.55
	33	Gironde	3567.93

Les 10 départements au prix du m² le plus cher

Résultats - exemple 4/9

```
1 -- Prix moyen du mètre carré d'une maison en Île-de-France.  
2  
3 • SELECT ROUND(SUM(prix_mut) / SUM(surf_carrez), 2) AS prix_m2_moy_maison_IdF  
4 FROM mutation  
5 JOIN bien ON mutation.bien_id = bien.id  
6 JOIN commune ON bien.commune_id = commune.id  
7 JOIN departement ON departement.id = commune.code_dpt  
8 WHERE type_local = "Maison" AND code_dpt IN (75, 77, 78, 91, 92, 93, 94, 95)
```

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

	prix_m2_moy_maison_IdF
▶	3675.71

Prix moyen au m²
d'une maison en île de
France

Résultats - exemple 5/9

```
1 -- Liste des 10 appartements les plus chers avec le département et le nombre de
2
3 • SELECT bien.id, bien.surf_carrez, mutation.prix_mut AS prix_mut, commune.code_
4 FROM bien
5 JOIN mutation ON mutation.bien_id = bien.id
6 JOIN commune ON bien.commune_id = commune.id
7 JOIN departement ON departement.id = commune.code_dpt
8 WHERE bien.type_local = "Appartement"
9 ORDER BY mutation.prix_mut DESC
10 LIMIT 10
```

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

	id	surf_carrez	prix_mut	code_dpt	nom_dpt
1	75116nanDD1853	9.10	9000000.00	75	Paris
2	91174nanAX16322	64.00	8600000.00	91	Essonne
3	75107nanAK34100	20.55	8577713.00	75	Paris
4	75117nanCO10015	42.77	7620000.00	75	Paris
5	75106nanAU14104	253.30	7600000.00	75	Paris
6	75101nanAZ8679	139.90	7535000.00	75	Paris
7	75116nanDV465	360.95	7420000.00	75	Paris
8	75116nanCN2411	595.00	7200000.00	75	Paris
9	75101nanBC672	122.56	7050000.00	75	Paris
10	75101nanAT4181	79.38	6600000.00	75	Paris

Liste des 10 appartements les plus chers

Résultats - exemple 6/9

```
1  -- Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020
2
3  • t1 AS (          -- 1er trimestre
4      SELECT COUNT(id) as nb_vente_t1
5      FROM mutation
6      WHERE date_mut BETWEEN "2020-01-01" AND "2020-03-31"),
7  ⊖ t2 AS (          -- 2ème trimestre
8      SELECT COUNT(id) as nb_vente_t2
9      FROM mutation
10     WHERE date_mut BETWEEN "2020-04-01" AND "2020-06-30")
11    SELECT ROUND((nb_vente_t2 - nb_vente_t1) / nb_vente_t1 * 100, 2) as taux_evolution_t1t2
12    FROM t1, t2;
```

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

	taux_evolution_t1t2
▶	3.66

Taux d'évolution des ventes d'appartement entre T1 et T2 2020

Résultats - exemple 7/9

```
1  -- Liste des communes où le nombre de ventes a augmenté d'au moins 20% entre le premier et le second trimestre de 2020
2
3 • WITH t1 AS (          -- 1er trimestre
4     SELECT commune_id, nom_commune, COUNT(mutation.id) as nb_vente_t1
5     FROM bien
6     JOIN mutation ON bien.id = mutation.bien_id
7     JOIN commune ON bien.commune_id = commune.id
8     WHERE date_mut BETWEEN "2020-01-01" AND "2020-03-31"
9     GROUP BY commune_id), -- Arrondissements, communes ?
10    t2 AS (           -- 2ème trimestre
11        SELECT commune_id, COUNT(mutation.id) as nb_vente_t2
12        FROM bien
13        JOIN mutation ON bien.id = mutation.bien_id
14        WHERE date_mut BETWEEN "2020-04-01" AND "2020-06-30"
15        GROUP BY commune_id) -- Arrondissements, communes ?
16        SELECT t1.commune_id, t1.nom_commune, (nb_vente_t2 - nb_vente_t1) / nb_vente_t1 *100 as taux_evolution,
17            nb_vente_t1, nb_vente_t2
18        FROM t1
19        JOIN t2 USING (commune_id)
20        HAVING taux_evolution > 20
```

	commune_id	nom_commune	taux_evolution	nb_vente_t1	nb_vente_t2
	301	LEUCATE	20.6897	29	35
	205	ANTIBES	20.8333	24	29
	2885	ETAMPES	20.8333	24	29
	729	TOULOUSE	21.4286	14	17
	2416	LE CHESNAY-ROCQUENCOURT	21.4286	14	17
	3037	ORLY	22.2222	9	11
	3044	BOISSY-SAINT-LEGER	22.2222	9	11
	3164	FORT DE FRANCE	22.2222	9	11
	3197	PARIS 08	24.1935	62	77
	513	VALENCE	25.0000	28	35
	811	CADAUJAC	25.0000	4	5
	1176	SOLITONS	25.0000	2	10



Résultats - exemple 8/9

```
1  -- Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et
2
3 • WITH t2 AS (          -- prix m2 d'un T2
4     SELECT nb_piece, SUM(prix_mut) / SUM(surf_carrez) as prix_m2_t2
5     FROM mutation
6     JOIN bien ON bien.id = mutation.bien_id
7     WHERE type_local = "Appartement"
8     AND nb_piece = 2),
9 • t3 AS (          -- prix m2 d'un T3
10    SELECT nb_piece, SUM(prix_mut) / SUM(surf_carrez) as prix_m2_t3
11    FROM mutation
12    JOIN bien ON bien.id = mutation.bien_id
13    WHERE type_local = "Appartement"
14    AND nb_piece = 3)
15    SELECT ROUND((prix_m2_t3 - prix_m2_t2) / prix_m2_t2 * 100,2) AS prix_m2_t3_par_rapport_t2
16    FROM t2, t3;
```

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

	prix_m2_t3_par_rapport_t2
▶	-12.61

Définition de la requête SQL :

La requête calcule la différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces (T2) et un appartement de 3 pièces (T3). Elle utilise deux sous-requêtes avec des alias t2 et t3 pour calculer le prix au m² pour chaque type d'appartement. Ensuite, elle calcule la différence entre ces deux prix et l'exprime en pourcentage par rapport au prix du T2.

Annotation manuscrite :

Définition de la requête SQL :

La requête calcule la différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces (T2) et un appartement de 3 pièces (T3).

Résultats - exemple 9/9

```
1 -- Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69
2
3 • WITH valeur_par_ville AS (          -- Calcul des valeurs foncières moyennes par commune
4     SELECT code_dpt, nom_commune, avg(prix_mut) as valeur
5     FROM mutation
6     JOIN bien ON mutation.bien_id = bien.id
7     JOIN commune ON bien.commune_id = commune.id
8     WHERE code_dpt IN (6,13,33,59,69)
9     GROUP BY code_dpt, nom_commune)
10    SELECT code_dpt, nom_commune, round(valeur,2) AS prix_moyen
11  FROM (
12      SELECT code_dpt, nom_commune, valeur,
13          RANK() OVER (PARTITION BY code_dpt ORDER BY valeur DESC) AS rang
14      FROM valeur_par_ville) AS resultat
15  WHERE rang <= 3;
```

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

code_dpt	nom_commune	prix_moyen
13	GIGNAC-LA-NERTHE	330000.00
13	SAINT SAVOURNIN	314425.00
13	CASSIS	313416.88
33	LEGE-CAP-FERRET	549500.64
33	VAYRES	335000.00
33	ARCACHON	307435.93
59	BERSEE	433202.00
59	CYSOING	408550.00
59	HALLUIN	322250.00
6	SAINT-JEAN-CAP-FERRAT	968750.00
6	EZE	655000.00
6	MOUANS-SARTOUX	476898.10
69	VILLE SUR JARNIOUX	485300.00
69	LYON 2EME	455217.27
69	LYON 6EME	426968.25

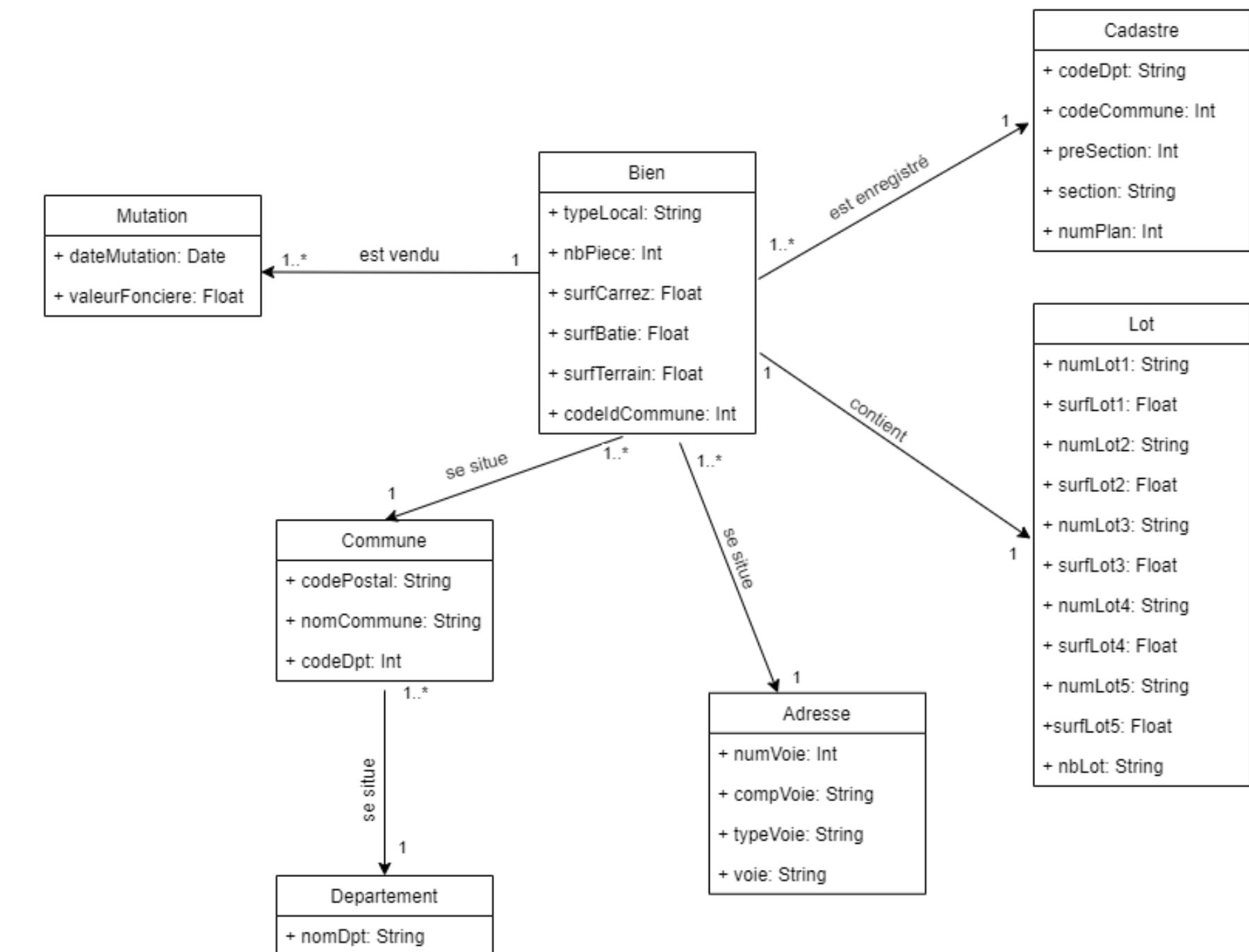
Top 3 des communes (valeur foncière moyenne) pour les départements 6, 13, 33, 59, 69

Annexe

Evolution de la base de données



Rien n'est figé, exhaustivité plus grande possible de la base de données pour avoir une vison "micro" (à l'échelle de l'adresse, du code cadastral) du prix des biens immobiliers



DATAImmo

Décembre 2021

Matthieu GIMBERT

matgimbert@gmail.com

07 83 70 67 29

