



# Création et utilisation de la base de données

Rouhling Jules



Laplace Immo

# Contexte du projet

## « DATAImmo »

Créer un modèle de prédiction du prix de vente des biens immobiliers à l'échelle nationale

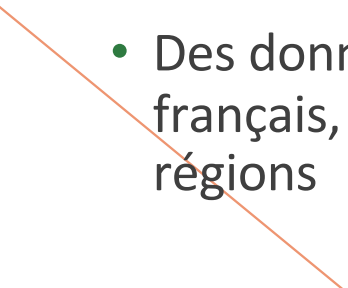


# La stratégie de sauvegarde et la conformité RGPD

- **Ne collectez que les données vraiment nécessaires à votre objectif**  
=> noms acquéreurs supprimés, adresses logements utilisées uniquement pour création Id\_bien
- **Soyez transparent**  
=> INSEE et data.gouv = données publiques, DVF = pas de ré-identification possible des acheteurs
- **Organisez et facilitez l'exercice des droits des personnes**  
=> les acquéreurs peuvent contacter directement Laplace Immo via le formulaire
- **Fixez des durées de conservation**  
=> les données seront archivées une fois les requêtes exécutées
- **Sécurisez les données et identifiez les risques**  
=> les données sont publiques
- **Inscrivez la mise en conformité dans une démarche continue**  
=> les procédures de mise en conformité seront scrutées et ajustées régulièrement

# Les données initiales



- Des données extraites du site open data des Demandes de valeurs foncières (DVF)
  - Des données de l'INSEE avec les résultats des recensements de la population
  - Des données de data.gouv sur les régions, avec le référentiel géographique français, communes, unités urbaines, aires urbaines, départements, académies, régions
- 

# Nettoyage des données

## Table DVF

=> création table **Biens** (1 bien = 1 id) en **supprimant doublons**

=> création table **Vente** (1 vente = 1 id) en **ajoutant Id\_bien avec RECHERCHEV** comme clé étrangère

## Table Commune

=> **concaténation code département et code commune** pour clé primaire

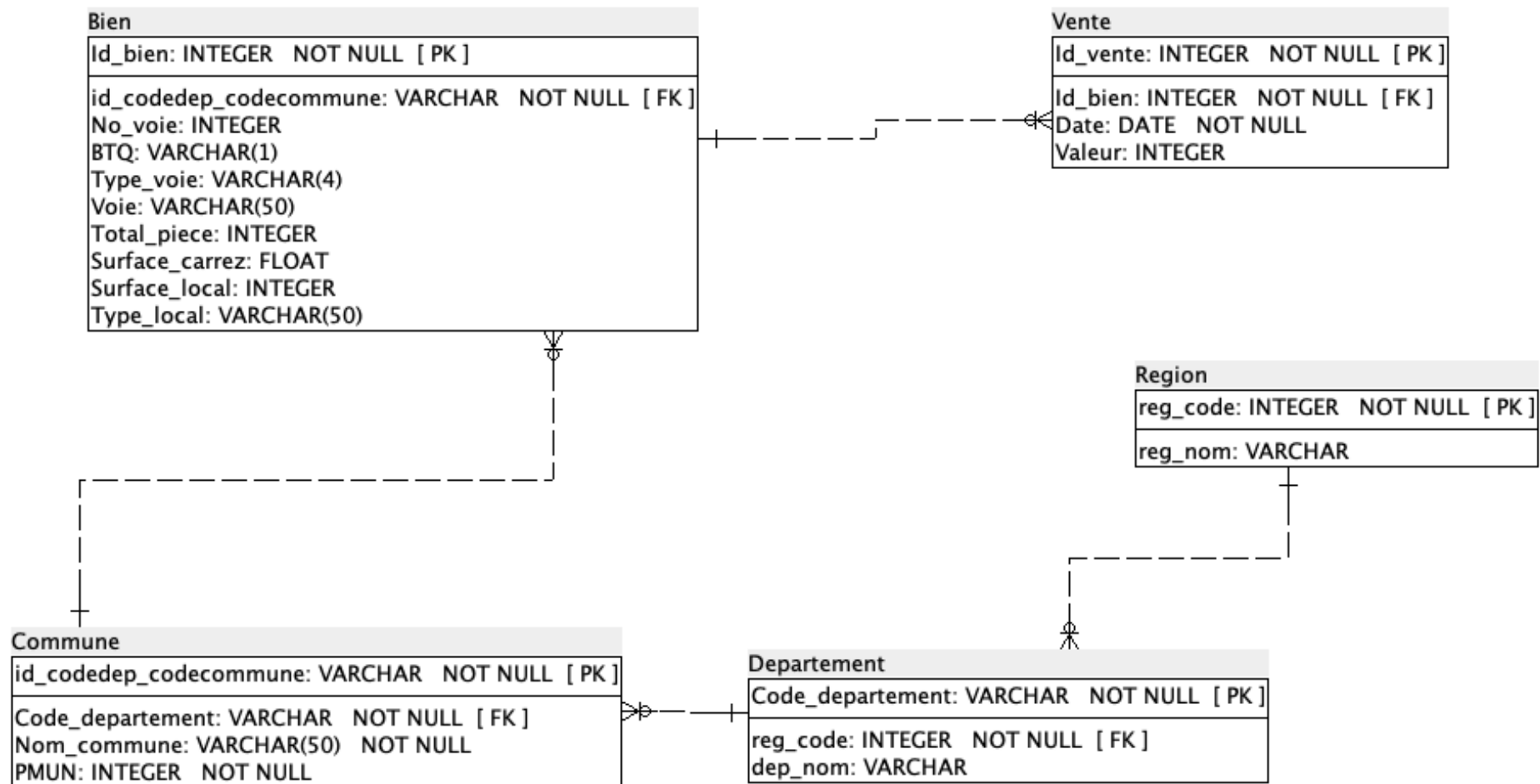
=> **ajoutant des populations (INSEE)** avec **RECHERCHEV**

# L'extrait du dictionnaire des données

conditionnelle forme de tableau cellule

L34

# Le schéma relationnel normalisé





# La base de données avec les tables créées et les données chargées

Database: Ventures immo

- Tables (5)
  - Bien
    - Columns (10)
      - Id\_bien
      - id\_codedep\_codecom...
      - No\_voie
      - BTQ
      - Type\_voie
      - Voie
      - Total\_piece
      - Surface\_carrez
      - Surface\_local
      - Type\_local
    - Indexes
    - Triggers
  - Commune
    - Columns (4)
      - id\_codedep\_codecom...
      - Code\_departement
      - Nom\_commune
      - PMUN
    - Indexes
    - Triggers
- Departement
  - Columns (3)
    - Code\_departement
    - reg\_code
    - dep\_nom
  - Indexes
  - Triggers
- Region
  - Columns (2)
    - reg\_code
    - reg\_nom
  - Indexes
  - Triggers
- Vente
  - Columns (4)
    - Id\_vente
    - Id\_bien
    - Date
    - Valeur
  - Indexes
  - Triggers

```
1 SELECT * FROM Vente
```

Total rows loaded: 34169

	Id_vente	Id_bien	Date	Valeur
1	1	1	2020/01/02	165000
2	2	2	2020/01/02	355680
3	3	3	2020/01/02	229500



Les requêtes ou screenshot qui permettent de démontrer le bon chargement des données

fr-esr-referentiel-geographique

Nb (non vides) : 38917

```
1 SELECT * FROM Commune
```



1



Total rows loaded: 38916

	id_codedep_codecommune	Code_departement	Nom_commune	PMUN
1	01001	01	L'Abergement-Clémenciat	779



## Requêtes SQL et résultats

# Requête 1

Nombre total d'appartements vendus au 1er semestre 2020

```
SELECT count(v.Id_vente) AS nb_ventes_appartements  
FROM Vente v  
NATURAL JOIN Bien b  
WHERE b.Type_local = "Appartement"  
AND v.Date between '2020/01/01' and '2020/07/01'
```

nb_ventes_appartements
------------------------

31378
-------

# Requête 2

Le nombre de ventes d'appartement par région pour le 1er semestre 2020

```
SELECT count(v.Id_vente) AS nb_ventes, r.reg_nom  
FROM Vente v  
NATURAL JOIN Bien b  
NATURAL JOIN Departement  
NATURAL JOIN Commune  
NATURAL JOIN Region r  
WHERE b.Type_local = "Appartement"  
AND v.Date between '2020/01/01' and '2020/07/01'  
GROUP BY r.reg_nom  
ORDER BY count(v.Id_vente) DESC
```

	nb_ventes	reg_nom
1	13995	Ile-de-France
2	3649	Provence-Alpes-Côte d'Azur
3	3253	Auvergne-Rhône-Alpes
4	1932	Nouvelle-Aquitaine
5	1640	Occitanie
6	1357	Pays de la Loire
7	1254	Hauts-de-France
8	984	Grand Est
9	983	Bretagne
10	862	Normandie
11	696	Centre-Val de Loire
12	376	Bourgogne-Franche-Comté
13	223	Corse
14	94	Martinique
15	44	La Réunion

# Requête 3

Proportion des ventes d'appartements par le nombre de pièces

```
WITH Vente_appartement AS (SELECT count(Id_vente) AS  
total_ventes  
FROM Vente v  
NATURAL JOIN Bien b  
WHERE b.Type_local = "Appartement")  
SELECT b.Total_piece as nb_pieces,  
ROUND(CAST(count(v.Id_vente) AS FLOAT) * 100 / total_ventes,  
2) AS total_PCT, count(v.Id_vente) as nb_ventes  
FROM Vente_appartement, Vente v  
NATURAL JOIN Bien b  
WHERE b.Type_local = "Appartement"  
GROUP BY b.Total_piece  
ORDER BY b.Total_piece
```

	nb_pieces	total_PCT	nb_ventes
1	0	0.1	30
2	1	21.48	6739
3	2	31.18	9783
4	3	28.57	8966
5	4	14.21	4460
6	5	3.55	1114
7	6	0.65	204
8	7	0.17	54
9	8	0.05	17
10	9	0.03	8
11	10	0.01	2
12	11	0	1

## Requête 4

Liste des 10 départements où le prix du mètre carré est le plus élevé

```
SELECT c.Code_departement AS departement,  
ROUND(AVG(v.Valeur / b.Surface_carrez)) AS  
prix_m2  
FROM Vente v  
NATURAL JOIN Bien b  
NATURAL JOIN Commune c  
GROUP BY c.Code_departement  
ORDER BY prix_m2 DESC LIMIT 10
```

	departement	prix_m2
1	75	12084
2	92	7300
3	94	5427
4	74	4781
5	06	4755
6	93	4385
7	78	4275
8	69	4100
9	2A	4062
10	33	3807

# Requête 5

Prix moyen du mètre carré d'une maison en Île-de-France

```
SELECT ROUND(AVG(v.Valeur /  
b.Surface_carrez)) AS prix_moyen_m2, r.reg_nom  
FROM Vente v  
NATURAL JOIN Bien b  
NATURAL JOIN Commune  
NATURAL JOIN Departement  
NATURAL JOIN Region r  
WHERE upper(r.reg_nom) = "ILE-DE-FRANCE"  
AND b.Type_local = "Maison"
```

prix_moyen_m2	reg_nom
3764	Ile-de-France



## Requête 6

Liste des 10 appartements les plus chers avec la région et le nombre de mètres carrés

```
SELECT b.Id_bien, v.Valeur AS montant,  
CAST(b.Surface_carrez AS INT) AS surface,  
r.reg_nom  
FROM Vente v  
NATURAL JOIN Bien b  
NATURAL JOIN Commune  
NATURAL JOIN Departement  
NATURAL JOIN Region r  
WHERE b.Type_local = "Appartement"  
ORDER BY CAST(v.Valeur AS FLOAT) DESC  
LIMIT 10
```

	Id_bien	montant	surface	reg_nom
1	30591	9000000	9	Ile-de-France
2	5261	8600000	64	Ile-de-France
3	3625	8577713	20	Ile-de-France
4	7602	7620000	42	Ile-de-France
5	9988	7600000	253	Ile-de-France
6	17821	7535000	139	Ile-de-France
7	410	7420000	360	Ile-de-France
8	16355	7200000	595	Ile-de-France
9	1924	7050000	122	Ile-de-France
10	19158	6600000	79	Ile-de-France

# Requête 7

Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020

```
WITH Ventes_1 AS (  
  SELECT count(Id_vente) AS ventes_trim_1  
  FROM Vente  
  WHERE Date between '2020/01/01' and '2020/03/31'),  
Ventes_2 AS (  
  SELECT count(Id_vente) AS ventes_trim_2  
  FROM Vente  
  WHERE Date between '2020/04/01' and '2020/06/30')  
SELECT ventes_trim_1, ventes_trim_2,  
  ROUND((CAST(ventes_trim_2 AS FLOAT) /  
  CAST(ventes_trim_1 AS FLOAT)-1)*100,2) AS PCT_evolution  
FROM Ventes_1, Ventes_2
```

ventes_trim_1	ventes_trim_2	PCT_evolution
16776	17393	3.68

# Requête 8

Le classement des régions par rapport au prix au mètre carré des appartement de plus de 4 pièces

```
SELECT r.reg_nom, ROUND(AVG(v.Valeur /  
b.Surface_carrez)) AS prix_moyen_m2  
FROM Vente v  
NATURAL JOIN Bien b  
NATURAL JOIN Commune  
NATURAL JOIN Departement  
NATURAL JOIN Region r  
WHERE b.Type_local = "Appartement"  
AND b.Total_piece > 4  
GROUP BY r.reg_nom  
ORDER BY prix_moyen_m2 DESC
```



reg_nom	prix_moyen_m2
Ile-de-France	8806
La Réunion	3659
Provence-Alpes-Côte d'Azur	3616
Corse	3117
Auvergne-Rhône-Alpes	2903
Nouvelle-Aquitaine	2476
Bretagne	2427
Pays de la Loire	2329
Hauts-de-France	2199
Occitanie	2107
Normandie	2026

## Requête 9

Liste des communes ayant eu au moins 50 ventes au 1er trimestre

```
SELECT c.Nom_commune, count(v.Id_vente) AS  
nb_ventes  
FROM Vente v  
NATURAL JOIN Bien  
NATURAL JOIN Commune c  
WHERE v.Date between '2020/01/01' and  
'2020/03/31'  
GROUP BY c.Nom_commune  
HAVING nb_ventes > 50  
ORDER BY c.Nom_commune
```

Total rows loaded: 47

Nom_commune	nb_ventes
Ajaccio	54
Angers	64
Antibes	77
Asnières-sur-Seine	81
Bordeaux	157
Boulogne-Billancourt	99
Courbevoie	80
Grenoble	106
La Ciotat	62
Levallois-Perret	59
Lille	67
Marseille 4e	74

# Requête 10

Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces

```
WITH Deux AS (  
  SELECT ROUND(AVG(v.Valeur/b.Surface_carrez)) AS prix_2_pieces  
  FROM Vente v  
  NATURAL JOIN Bien b  
  WHERE b.Total_piece = 2 AND b.Type_local = "Appartement"),  
Trois AS (  
  SELECT ROUND(AVG(v.Valeur/b.Surface_carrez)) AS prix_3_pieces  
  FROM Vente v  
  NATURAL JOIN Bien b  
  WHERE b.Total_piece = 3 AND b.Type_local = "Appartement")  
SELECT prix_3_pieces, prix_2_pieces, ROUND((((prix_3_pieces/prix_2_pieces -  
1)*100),2) AS DIFF_PRIX_PCT  
FROM Deux, Trois
```

prix_3_pieces	prix_2_pieces	DIFF_PRIX_PCT
4335	4965	-12.69

# Requête 11

Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69

```
WITH cte AS(
SELECT c.Code_departement AS dep, c.Nom_commune,
RANK() OVER(PARTITION BY Code_departement ORDER BY
AVG(v.Valeur) DESC) AS rang, CAST(AVG(v.Valeur) AS INT)
AS val_moyenne
FROM Vente v
NATURAL JOIN Bien b
NATURAL JOIN Commune c
WHERE c.Code_departement IN ("06", "13", "33", "59", "69")
GROUP BY c.Nom_commune)
SELECT * FROM cte
WHERE rang <= 3
```

dep	Nom_commune	rang	val_moyenne
06	Saint-Jean-Cap-Ferrat	1	968750
06	Eze	2	655000
06	Mouans-Sartoux	3	476898
13	Gignac-la-Nerthe	1	330000
13	Saint-Savournin	2	314425
13	Cassis	3	313416
33	Lège-Cap-Ferret	1	549500
33	Vayres	2	335000
33	Arcachon	3	307435
59	Bersée	1	433202
59	Cysoing	2	408550
59	Halluin	3	322250
69	Ville-sur-Jarnioux	1	485300
69	Lyon 2e	2	455217
69	Lyon 6e	3	426968

# Requête 12

Les 20 communes avec le plus de transactions pour 1000 habitants pour les communes qui dépassent les 10 000 habitants

```
SELECT c.Nom_commune as commune, c.PMUN as  
population, ROUND((CAST(count(v.Id_vente) AS  
FLOAT) * 1000 / c.PMUN),2) as  
nb_transactions_1000_hab  
FROM Vente v  
LEFT JOIN Bien b ON v.Id_bien = b.Id_bien  
LEFT JOIN Commune c ON  
b.id_codedep_codecommune =  
c.id_codedep_codecommune  
GROUP BY c.Nom_commune  
HAVING c.PMUN >= 10000  
ORDER BY nb_transactions_1000_hab DESC LIMIT 20
```

	commune	population	nb_transactions_1000_hab
1	Paris 2e	21595	5.88
2	Paris 1er	15917	4.96
3	La Baule-Escoublac	16255	4.74
4	Paris 3e	34025	4.73
5	Arcachon	11630	4.73
6	Paris 4e	29131	4.12
7	Roquebrune-Cap-Martin	12839	4.05
8	Paris 8e	35655	3.9
9	Sanary-sur-Mer	16889	3.55





**Merci !**