

Projet DATAImmo

Présentation des résultats

LAPLACE IMMO

Contexte du projet

Mission :

- Notre réseau souhaite se démarquer de la concurrence en prévoyant le prix de vente des biens immobiliers.
- Projet stratégique dirigé par Clara Daucourt (CTO).
- Modifications de la base de données pour collecter les transactions immobilières et foncières en France.

Objectifs :

- 🍷 Analyser le marché immobilier.
- 🍷 Aider les agences régionales à mieux accompagner leurs clients.

Stratégie de sauvegarde et vérification de la conformité RGPD

Stratégie de Sauvegarde :

Sauvegardes régulières des données sur des serveurs sécurisés.

Utilisation de technologies de redondance pour garantir la disponibilité.

Tests périodiques des sauvegardes

Conformité RGPD :

Globalement, les tables ne contiennent pas de Donnée Personnel Identifiable.

Néanmoins, certaines données doivent être traitées avec prudence lorsqu'elles sont associées avec d'autres données (Numéro de voie, Voie, Code postal)

Les données initiales

Sources de données utilisées :

Sources libres, disponibles, origines étatiques, régulièrement mises à jour contenant les transactions immobilières (DVF), les données démographiques (INSEE) et les données géographiques (data.gouv) :

- fr-esr-referentiel-geographique : Il fournit les différents niveaux d'agrégats géographiques associés aux communes françaises
- Valeurs-foncières : Il recense les transactions immobilières intervenues en France au cours des cinq dernières années. Les données proviennent des actes notariés et des informations cadastrales.
- Communes-France : Il rassemble les communes de France métropolitaine et des départements d'outre-mer (DROM). Les informations proviennent de diverses sources, telles que l'INSEE, l'IGN et Natural Earth.

Dictionnaire des Données

Proposition dans le cadre du projet

Table Bien :

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
Id_bien	ID dans la base de données	Varchar	15	Elémentaire	Ne doit pas être nul	
Id_codedep_codecommune		VARCHAR	15	Elémentaire	Non null	
No_voie	Numéro des rues	Varchar	5	Elémentaire	Ne doit pas être nul	
BTQ	Complément N° de rue	Varchar	15	Elémentaire		
Type_de_voie	Plusieurs valeurs (rue, avenue, chemin, etc.)	Varchar	50	Elémentaire	Ne doit pas être nul	
Voie	Nom de la rue	Varchar	150	Elémentaire	Ne doit pas être nul	
Total_pièce	Nombre de pièce dans le bien	Integer				
Surface_Carrez	Surface Carrez du bien	Integer		Elémentaire	Ne doit pas être nul	
Surface_local	Surface du local	Integer		Elémentaire	Ne doit pas être nul	
Type_local	Maison ou appartement	Varchar	15	Elémentaire	Ne doit pas être nul	

Table Vente :

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
Id_vente	ID dans la base de données	Varchar	15	Elémentaire	Ne doit pas être nul	
Id_bien	ID dans la base de données	Varchar	15	Elémentaire	Ne doit pas être nul	
Date de mutation	Date de mutation du bien	Date		Elémentaire	Ne doit pas être nul yyyymmdd	
Valeur foncière	Valeur foncière du bien	Integer		Elémentaire		

Table Commune :

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
Id_codedep_codecommune	Clé primaire	VARCHAR	15	Elémentaire	Non null	
Id_region	Id de la région	VARCHAR	15	Elémentaire	Non null	
CODDEP	Numéro de département	VARCHAR	3	Elémentaire	Non null	
CODCOM	Code commune	VARCHAR	5	Elémentaire	Non null	
Code_postal	Code postal	VARCHAR	5	Elémentaire	Non null	
COM	Nom de la commune	VARCHAR	150	Elémentaire	Non null	
Nbre_habitant	Nombre d'habitant en 2019	VARCHAR	15	Elémentaire	Non null	

Table Région :

CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
Id_region	Id de la région	VARCHAR	15	Elémentaire	Non null	
Nom_region	Nom des regions	VARCHAR	3	Elémentaire	Non null	
Nom_regoup	Nom des regions regroupées	VARCHAR	5	Elémentaire	Non null	

Schéma Relationnel Normalisé

Les associations et cardinalités :

Un bien peut être vendu plusieurs fois mais chaque vente concerne un bien unique.

Une commune peut contenir plusieurs biens mais un bien ne peut se trouver que dans une seule commune.

Une région regroupe plusieurs communes mais une commune ne fait partie que d'une seule région.

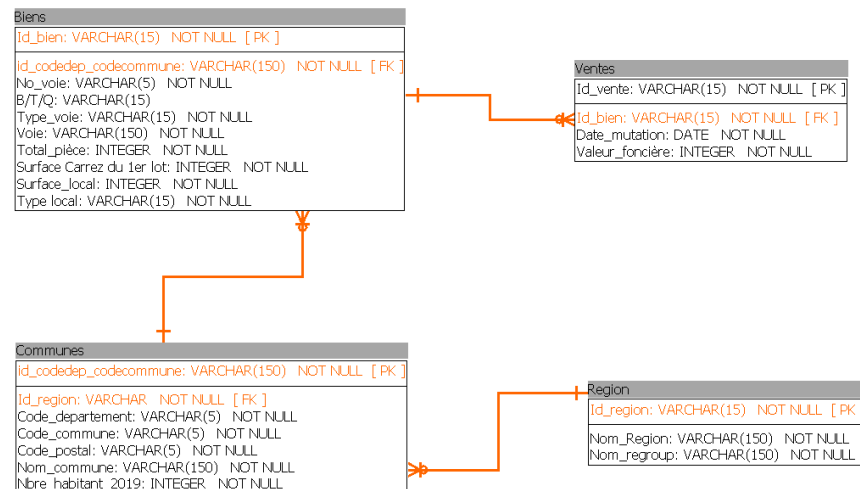
Les trois formes normales :

1NF : Les tables contiennent des valeurs atomiques dans chaque colonne.

2NF : Les tables ont des clés primaires et leurs attributs dépendent uniquement de ces clés.

3NF : Chaque attribut non-clé est directement lié à la clé primaire de sa table.

Schéma relationnel normalisé :



Base de Données

Tables créées et données chargées

```
1 SELECT * FROM Bien
```

Id_bien	Id_codedep_c...	No_v...	BTQ	Type...	Voie	Total...	Surfa...	Surfa...	Type_local
BI0000001	01103	347	NULL	RUE	DU C...	3	48.22	48	Appartement
BI0000002	06004	4	NULL	BD	EDOU...	1	39.11	40	Appartement

```
1 SELECT COUNT(*) FROM Bien
```

```
1 COUNT(*)
```

```
34169
```

```
1 SELECT * FROM Commune
```

Id_codedep_codecommune	Id_region	CODDEP	CODCOM
01001	REG00001	01	001
01002	REG00001	01	002

```
1 SELECT COUNT(*) FROM Commune
```

```
1 COUNT(*)
```

```
34991
```

```
1 SELECT * FROM Region
```

Id_region	Nom_region	Nom_regoup
REG00001	Auvergne-Rhône-Alpes	Province
REG00002	Hauts-de-France	Province

```
1 SELECT COUNT(*) FROM Region
```

```
1 COUNT(*)
```

```
19
```

```
1 SELECT * FROM Vente
```

Id_vente	Id_bien	Date_mutation	Valeur_foncière
VT00000001	BI0000001	2020-01-02	165000
VT00000002	BI0000002	2020-01-02	355680

```
1 SELECT COUNT(*) FROM Vente
```

```
1 COUNT(*)
```

```
34169
```

Code SQL et Résultats des Requêtes

Requêtes 1 : Nombre total d'appartements vendus au 1er semestre 2020

```
1 SELECT COUNT(v.Id_vente) AS "Nombre d'appartements vendus au 1er semestre 2020"  
2 FROM Vente v  
3 JOIN Bien b ON v.Id_bien = b.Id_bien  
4 WHERE b.Type_local = 'Appartement'  
5 AND v.Date_mutation BETWEEN '2020-01-01' AND '2020-06-30'
```

⌵ Nombre d'appartements vendus au 1er semestre 2020

31378

Code SQL et Résultats des Requêtes

Requêtes 2 : Le nombre de ventes d'appartement par région pour le 1er semestre 2020.

```
1 SELECT r.Nom_region ,COUNT(v.Id_vente) AS "Nombre d'appartements vendus au 1er semestre 2020"
2 FROM Vente v
3 JOIN Bien b ON v.Id_bien = b.Id_bien
4 JOIN Commune c ON c.Id_codedep_codecommune = b.Id_codedep_codecommune
5 JOIN Region r ON r.Id_region = c.Id_region
6 WHERE b.Type_local = 'Appartement'
7 AND v.Date_mutation BETWEEN '2020-01-01' AND '2020-06-30'
8 GROUP BY r.Nom_region
```

Nom_region	Nombre d'appartements vendus au 1er semestre 2020
Auvergne-Rhône-Alpes	3246
Bourgogne-Franche-Comté	529

Code SQL et Résultats des Requêtes

Requêtes 3 : Proportion des ventes d'appartements par le nombre de pièces.

```
1 WITH Nbre_piece_par_logement AS (  
2   SELECT total_pièce AS "Nombre de pièces", COUNT(id_codedep_codecommune) AS "Nombre_logements"  
3   FROM Bien  
4   WHERE type_local = 'Appartement'  
5   GROUP BY total_pièce  
6 ),  
7 Logements AS (  
8   SELECT COUNT(Bien.Id_bien) AS deux  
9   FROM Bien  
10  WHERE type_local = 'Appartement'  
11 )  
12  
13 SELECT  
14   Nbre_piece_par_logement."Nombre de pièces",  
15   Nbre_piece_par_logement."Nombre_logements",  
16   round(CAST(Nbre_piece_par_logement."Nombre_logements" AS FLOAT) / Logements.deux * 100, 2) || '%' AS "Proportion de logement"  
17 FROM Nbre_piece_par_logement, Logements;
```

! Nombre de pièces	Nombre_logements	Proportion de logement
0	30	0.1%
1	6739	21.48%

Code SQL et Résultats des Requêtes

Requêtes 4 : Liste des 10 départements où le prix du mètre carré est le plus élevé.

```
1 SELECT coddep AS "Département", round(valeur_foncière/surface_carrez,2) AS "Prix moyen"
2 FROM Bien, Vente, Commune
3 WHERE Bien.Id_codedep_codecommune = Commune.Id_codedep_codecommune
4 AND Bien.Id_bien = Vente.Id_bien
5 GROUP BY coddep
6 ORDER BY valeur_foncière/surface_carrez DESC
7 LIMIT 10
```

Département	Prix moyen
75	11880.2
06	9094.35

Code SQL et Résultats des Requêtes

Requêtes 5 : Prix moyen du mètre carré d'une maison en Île-de-France.

```
1 SELECT b.Type_local, round(AVG(v.valeur_foncière / b.Surface_Carrez),2) AS 'Prix moyen / M² maison Ile-de-France'
2 FROM Vente v
3 JOIN Bien b ON v.Id_bien = b.Id_bien
4 JOIN Commune c ON b.Id_codedep_codecommune = c.Id_codedep_codecommune
5 JOIN Region r ON c.Id_region = r.Id_region
6 WHERE r.Nom_region = 'Ile-de-France'
7 AND b.Type_local = 'Maison';
```

Type_local	Prix moyen / M² maison Ile-de-France
Maison	3763.89

Code SQL et Résultats des Requêtes

Requêtes 6 : Liste des 10 appartements les plus chers avec la région et le nombre de mètres carrés.

```
1 SELECT v.Id_bien, r.nom_region, b.surface_carrez, v.valeur_foncière
2 FROM Vente v
3 JOIN Bien b ON v.Id_bien = b.Id_bien
4 JOIN Commune c ON b.Id_codedep_codecommune = c.Id_codedep_codecommune
5 JOIN Region r ON c.Id_region = r.Id_region
6 WHERE b.type_local = 'Appartement'
7 ORDER BY v.valeur_foncière DESC
8 LIMIT 10
```

Id_bien	Nom_region	Surface_Carrez	Valeur_foncière
BI0030603	Ile-de-France	9.1	9000000
BI0005261	Ile-de-France	64	8600000

Code SQL et Résultats des Requêtes

Requêtes 7 : Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020.

```
1 WITH Q1_2020 AS (  
2     SELECT COUNT(v.Id_vente) AS total_sales_q1  
3     FROM Vente v  
4     WHERE date_mutation BETWEEN '2020-01-01' AND '2020-03-31'  
5 ),  
6 Q2_2020 AS (  
7     SELECT COUNT(v.Id_vente) AS total_sales_q2  
8     FROM Vente v  
9     WHERE date_mutation BETWEEN '2020-04-01' AND '2020-06-30'  
10 )  
11 SELECT  
12     round((((Q2_2020.total_sales_q2 - Q1_2020.total_sales_q1) * 100.0 / Q1_2020.total_sales_q1),2) || '%' AS "Taux d'évolution"  
13 FROM  
14     Q1_2020,  
15     Q2_2020;
```

Taux d'évolution

3.68%

Code SQL et Résultats des Requêtes

Requêtes 8 : Le classement des régions par rapport au prix au mètre carré des appartements de plus de 4 pièces.

```
1 SELECT r.nom_region, round(AVG(v.valeur_foncière / b.Surface_Carrez),2) AS 'Prix moyen'
2 FROM Vente v
3 JOIN Bien b ON v.Id_bien = b.Id_bien
4 JOIN Commune c ON b.Id_codedep_codecommune = c.Id_codedep_codecommune
5 JOIN Region r ON c.Id_region = r.Id_region
6 WHERE type_local = 'Appartement'
7 AND total_piece > 4
8 GROUP BY r.Nom_region
9 ORDER BY round(AVG(v.valeur_foncière / b.Surface_Carrez),2) ASC
```

Nom_region	Prix moyen
Martinique	573.48
Grand Est	1664.05

Code SQL et Résultats des Requêtes

Requêtes 9 : Liste des communes ayant eu au moins 50 ventes au 1er trimestre.

```
1 SELECT c.COM AS 'Commune', COUNT(v.Id_vente) AS 'Nombre de vente'
2 FROM Vente v
3 JOIN Bien b ON v.Id_bien = b.Id_bien
4 JOIN Commune c ON b.Id_codedep_codecommune = c.Id_codedep_codecommune
5 WHERE v.Date_mutation BETWEEN '2020-01-01' AND '2020-03-31'
6 GROUP BY c.COM
7 HAVING COUNT(v.Id_vente) > 50
8 ORDER BY COUNT(v.Id_vente) DESC
```

Commune	Nombre de vente
Paris 17e Arrondissement	228
Paris 15e Arrondissement	215

Code SQL et Résultats des Requêtes

Requêtes 10 : Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces.

```
1 WITH PrixM22P AS (  
2     SELECT AVG(v.valeur_foncière / b.surface_carrez) AS PM2P  
3     FROM Vente v, Bien b  
4     WHERE total_pièce = 2  
5 ),  
6 PrixM23P AS (  
7     SELECT AVG(v.valeur_foncière / b.surface_carrez) AS PM3P  
8     FROM Vente v, Bien b  
9     WHERE total_pièce = 3  
10 )  
11 SELECT  
12     round((((PrixM23P.PM3P - PrixM22P.PM2P) * 100.0 / PrixM22P.PM2P),2) || '%' AS 'Différence Prix/m² 2 vs 3 Pièces'  
13 FROM  
14     PrixM22P,  
15     PrixM23P;
```

Différence Prix/m² 2 vs 3 Pièces

-34.33%

Code SQL et Résultats des Requêtes

Requêtes 11 : Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69.

```
1 SELECT c.COM AS Commune, round(AVG(valeur_foncière),2) AS VF_Moyenne
2 FROM Commune c
3 JOIN Bien b ON b.Id_codedep_codecommune = c.Id_codedep_codecommune
4 JOIN Vente v ON v.Id_bien = b.Id_bien
5 WHERE coddep = 6
6 OR coddep = 13
7 OR coddep = 33
8 OR coddep = 59
9 OR coddep = 69
10 GROUP BY c.COM
11 ORDER BY VF_Moyenne DESC
12 LIMIT 3
```

Commune	VF_Moyenne
Lège Cap Ferret	549500.64
Ville sur Jarnioux	485300
Lyon 2e Arrondissement	455217.27

Code SQL et Résultats des Requêtes

Requêtes 12 : Les 20 communes avec le plus de transactions pour 1000 habitants pour les communes qui dépassent les 10 000 habitants.

```
1 SELECT      c.COM AS Commune, c.PTOT AS Population_totale, round((COUNT(v.Id_vente) * 1.0 / c.PTOT * 1000),2) AS 'Transactions pour 1000 habitants'
2 FROM        Commune c
3 JOIN        Bien b ON b.Id_codedep_codecommune = c.Id_codedep_codecommune
4 JOIN        Vente v ON b.Id_bien = v.Id_bien
5 WHERE       c.PTOT > 10000
6 GROUP BY    c.COM, c.PTOT
7 ORDER BY    round((COUNT(v.Id_vente) * 1.0 / c.PTOT * 1000),2) DESC
8 LIMIT 20;
9
```

Commune	Population_totale	Transactions pour 1000 habitants
Paris 2e Arrondissement	21735	5.84
Paris 1er Arrondissement	16055	4.92