

OPENCLASSROOMS



# Parcours Data Analyst

Projet 3



# Projet DATAImmo

Alhassane AHMED



1	INTRO .....	01
2	MODÉLISATION DE LA BDD .....	02
3	IMPLÉMENTATION DE LA BDD .....	06
4	REQUÊTES ET RÉSULTATS .....	10
5	BILAN .....	20
6	CONCLUSION .....	22

## Projet DATAImmo



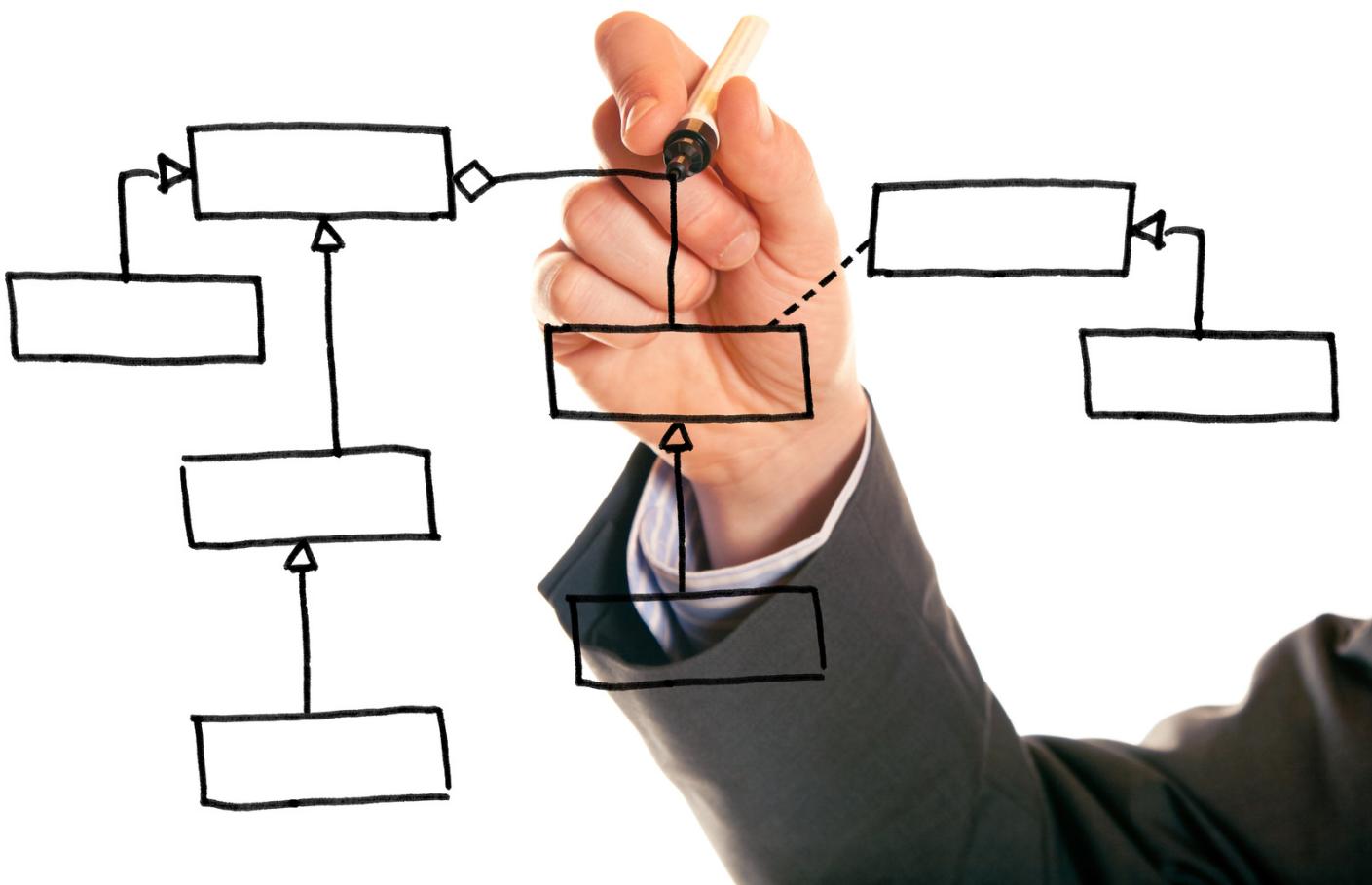
**Laplace Immo** est un réseau national d'agences immobilières

Il s'agit de créer un modèle à partir du Data set open source «Demandes de valeurs foncières» publié par la direction générale des finances publiques.

**Objectif du projet:** mieux prévoir le prix de vente des biens immobiliers.

**Source des données :** [data.gouv.fr](http://data.gouv.fr)

**Données traitées :** premier semestre 2020 dans un premier temps.



## MODÉLISATION DE LA BDD

1. Dictionnaire des données
2. Modèle conceptuel des données
3. Schéma relationnel normalisé 3FN

## 1. DICTIONNAIRE DES DONNÉES

Numéro ▾	Attribut ▾	Signification ▾	Type ▾	Observation ▾
1	typeLocal	Type du local	Texte	Longueur : 20
2	nbrDePieces	Nombre de pièces	Numérique	Longueur : 5
3	surfaceRelleBati	Surface réelle bati	Numérique	Longueur : 10
4	surfaceCarrezDuLot	Surface carrez du 1er lot	Numérique	Longueur : 10
5	surfaceTerrain	Surface du terrain	Numérique	Longueur : 10
6	numeroVoie	Numéro Voie	Texte	Longueur : 10
7	typeVoie	Type de voie	Texte	Longueur : 20
8	nomVoie	Voie	Texte	Longueur : 50
9	codePostal	Code Postal	Numérique	Longueur : 10
10	codeDepartement	Département	Texte	Longueur : 10
11	nomCommune	Nom de la commune	Texte	Longueur : 50
12	natureTransaction	Nature de la mutation	Texte	Longueur : 20
13	dateTransaction	Date de la mutation	Date	AAAA/MM/DD
14	valeurFonciere	Valeur foncière	Monétaire	

Choix des concepts sur la base de l'objectif du projet et des requêtes de l'énoncé.

# /04

## 2. MODÈLE CONCEPTUEL DES DONNÉES

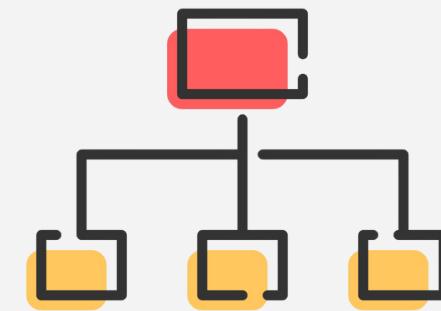
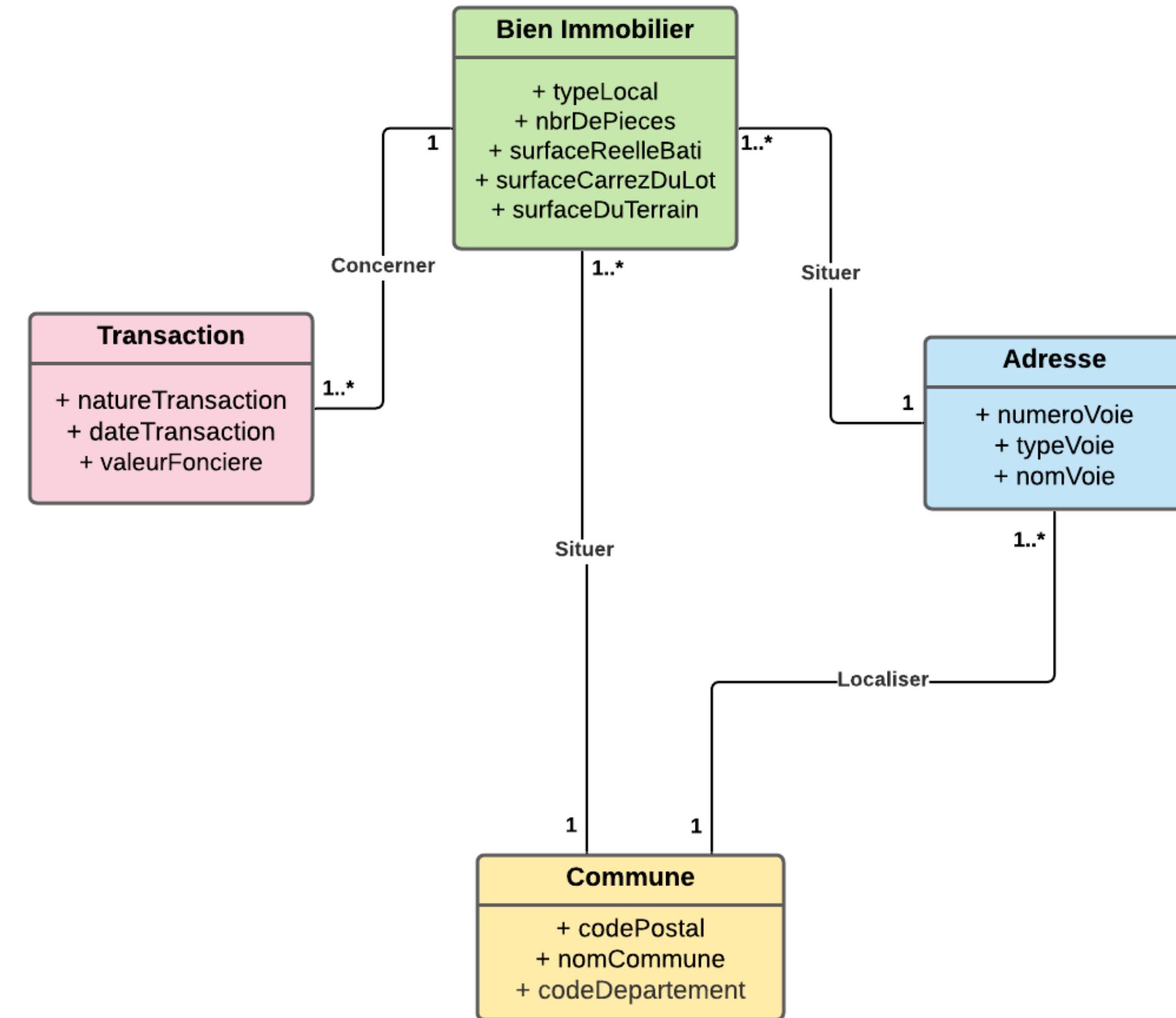


Diagramme conçu en **UML**

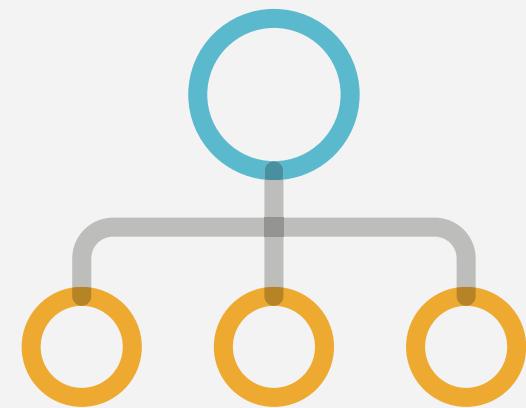
**LucidChart**



/05

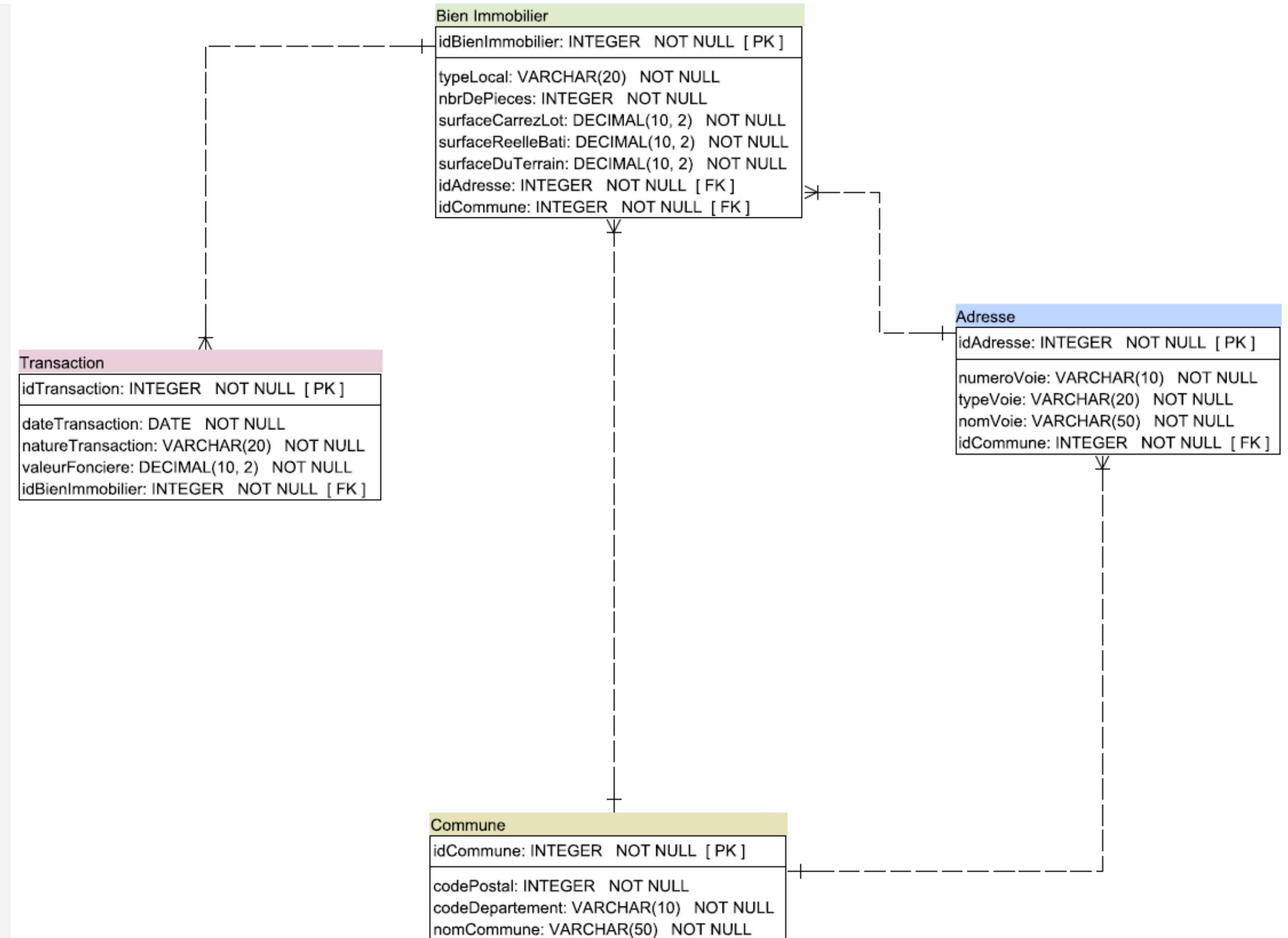
### 3. SCHÉMA RELATIONNEL NORMALISÉ

3FN



MPD

SQL Power Architect



## IMPLÉMENTATION DE LA BDD



1. SDBG, Création de la BDD
2. Préparation des données
3. Transfert des données

## 1. SDBG



- 2e du Top10 des SDBG sur DB-engines.com
- Simple à configurer
- Simple à utiliser
- Gratuit

## Aperçu de la BDD

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' tree view displays several databases: 'data\_immo', 'dataimmo' (selected), 'foodly', 'moviz', 'sakila', 'sys', and 'world'. Under 'dataimmo', there are 'Tables' (adresse, bien\_immobilier, commune, transaction) and other objects like Views, Stored Procedures, and Functions. At the bottom left, a status bar shows 'Administration' and 'Schemas' tabs, with 'Schema: dataimmo' selected. The main pane on the right is titled 'Aperçu de la BDD' and contains a generated SQL script:

```
1
2 • CREATE TABLE Commune (
3     idCommune INT AUTO_INCREMENT NOT NULL,
4     codePostal INT NOT NULL,
5     nomCommune VARCHAR(50) NOT NULL,
6     codeDepartement VARCHAR(10) NOT NULL,
7     PRIMARY KEY (idCommune)
8 );
9
10
11 • CREATE TABLE Adresse (
12     idAdresse INT AUTO_INCREMENT NOT NULL,
13     numeroVoie VARCHAR(10) NOT NULL,
14     typeVoie VARCHAR(20) NOT NULL,
15     nomVoie VARCHAR(50) NOT NULL,
16     idCommune INT NOT NULL,
17     PRIMARY KEY (idAdresse)
18 );
19
20
21 • CREATE TABLE Bien_Immobilier (
22     idBienImmobilier INT AUTO_INCREMENT NOT NULL,
23     ...
24 );
```

Script généré sur Power Architect

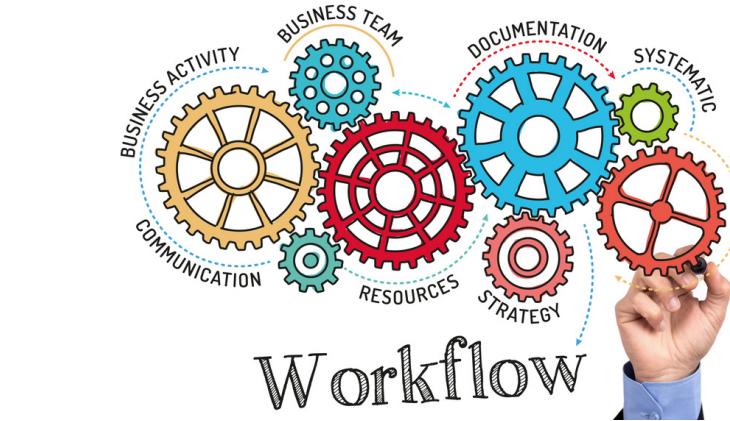
## 2. PRÉPARATION DES DONNÉES

Cette phase a permis de :

- Corriger les erreurs
- Gérer les doublons
- Obtenir des données de grande qualité



Prises de décisions plus avisées.



(Sur la base du schéma relationnel normalisé)

- ✓ Préparation des attributs
- ✓ Création des tables
- ✓ Fusion des tables et création des clés
- ✓ Nettoyage des données
- ✓ Transformation des données
- ✓ Exportation des tables en CSV.

# /09

## 3. IMPORT DES DONNÉS DANS LE SDBG

4 tables importées avec succès

- Commune : 3215 lignes
- Adresse : 30696 lignes
- Bien\_immobilier : 34145 lignes
- Transaction : 34169 lignes

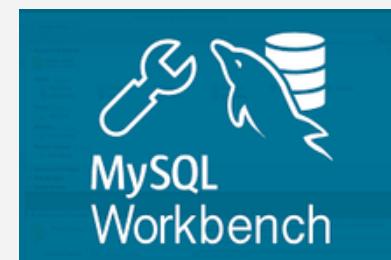


TABLE DATA IMPORT WIZARD

## Aperçu de la BDD après l'import

mysql> use dataimmo	mysql> show columns from bien_immobilier;
Database changed	+-----+-----+-----+-----+-----+-----+
mysql> show tables;	Field   Type   Null   Key   Default   Extra
+-----+	+-----+-----+-----+-----+-----+-----+
Tables_in_dataimmo	idBienImmobilier   int   NO   PRI   NULL   auto_increment
+-----+	+-----+-----+-----+-----+-----+-----+
adresse	typeLocal   varchar(20)   NO     NULL
bien_immobilier	nbrDePieces   int   NO     NULL
commune	surfaceCarrezLot   decimal(10,2)   NO     NULL
transaction	Surface_Bati   decimal(10,2)   NO     NULL
+-----+	Surface_Terrain   decimal(10,2)   NO     NULL
	idAdresse   int   NO   MUL   NULL
	idCommune   int   NO   MUL   NULL
	+-----+-----+-----+-----+-----+-----+

Result Grid | Filter Rows: | Edit: | Export/Import:

SCHEMAS					
Filter objects					
▶ data_immo					
▶ dataimmo					
▶ Tables					
▶ ▶ adresse					
▶ ▶ bien_immobilier					
▶ ▶ commune					
▶ ▶ transaction					
	idTransaction	dateTransaction	natureTransaction	valeurFonciere	idBienImmobilier
	34148	2020-06-22	Vente	1200000.00	34124
	34149	2020-06-26	Vente	500000.00	7538
	34150	2020-06-22	Vente	270100.00	34126
	34151	2020-06-25	Vente	849000.00	34127
	34152	2020-06-15	Vente	750000.00	34128
	34164	2020-06-15	Vente	518300.00	34140
	34165	2020-06-30	Vente	1290000.00	34141
	34166	2020-06-30	Vente	726000.00	26783
	34167	2020-06-25	Vente	420000.00	34143
	34168	2020-06-25	Vente	725000.00	34144
	34169	2020-06-11	Vente	433000.00	34145



**REQUÊTES  
ET  
RÉSULTATS**

## 1. Nombre total d'appartements vendus au 1er semestre 2020.

```
1  /*  
2   1. J'applique COUNT sur ibTransaction afin de compter le nombre de ventes.  
3   2. Je JOIN la table bien_immobilier.  
4   3. J'applique un filtre sur la colonne typeLocal pour limiter mes résultats aux Appartements.  
5   4. Je rajoute un 2e filtre pour déterminer la période concernée. */  
6  
7 • SELECT COUNT(DISTINCT tr.idTransaction) AS "Total des appartements vendus au 1er semestre 2020"  
8   FROM transaction tr  
9   INNER JOIN bien_immobilier bi ON (bi.idBienImmobilier = tr.idBienImmobilier)  
10  WHERE bi.typeLocal = "Appartement"  
11  AND tr.dateTransaction BETWEEN "2020-01-01" AND "2020-06-30";
```

WHERE bi.typeLocal LIKE "%partemen%"

Result Grid	
	Total des appartements vendus au 1er semestre 2020
▶	31378

## 2. Proportion des ventes d'appartements par le nombre de pièces.

```
1  /*
2   * 1. Je calcule le nombre total d'appartement vendus.
3   * 2. Je divise le COUNT des transactions par la première requête (requêtes imbriquées).
4   * 3. Je multiplie par 100 pour les pourcentages.
5   * 4. Je formate l'affichage des nombres décimaux avec Round.*/
6
7 • SELECT nbrDePieces AS "Nombre de pièces",
8   ROUND((COUNT(idTransaction)/(
9     SELECT COUNT(bi.idBienImmobilier) AS NbrTotalAppart
10    FROM bien_immobilier bi
11    INNER JOIN transaction tr USING (idBienImmobilier)
12    WHERE typeLocal = "Appartement")*100),2)
13    AS "Proportion des ventes des appartements par le no
14  FROM transaction tr
15  INNER JOIN bien_immobilier bi USING (idBienImmobilier)
16  WHERE bi.typeLocal = "Appartement"
17  GROUP BY bi.nbrDePieces
18  ORDER BY bi.nbrDePieces
```

	Nombre de pièces	Proportion des ventes des appartements par le nombre de pièces (en %)
►	0	0.10
	1	21.48
	2	31.18
	3	28.57
	4	14.21
	5	3.55
	6	0.65
	7	0.17
	8	0.05
	9	0.03
	10	0.01
	11	0.00

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

```
1  /*  
2   1. Je SELECT la colonne code département et Prix du mètre carré après l'avoir calculée.  
3   2. j'écarte les valeurs foncières nulles afin de s'assurer d'avoir une vraie moyenne.  
4   3. je group par codeDepartement  
5   4. Je tri le prix du mètre carré en ordre décroissant (du plus élevé au plus petit).  
6   Je limite l'affichage des résultats aux 10 premières lignes. */  
7  
8 • SELECT DISTINCT cm.codeDepartement AS "Les 10 départements les plus cher au prix du mètre carré"  
9   ROUND(AVG(tr.valeurFonciere/bi.surfaceCarrezLot),2) AS "Prix du mètre Carré"  
10  FROM commune cm  
11  JOIN bien_immobilier bi USING (idCommune)  
12  JOIN transaction tr USING (idBienImmobilier)  
13  WHERE bi.surfaceCarrezLot != 0  
14  AND tr.valeurFonciere != 0  
15  GROUP BY 1  
16  ORDER BY 2 DESC  
17  LIMIT 10;
```

Les 10 départements les plus cher au prix du mètre Carré	Prix du mètre Carré
75	12223.09
92	7259.11
94	5499.46
74	4894.57
6	4742.34
93	4477.42
78	4393.84
73	4247.75
69	4207.95
5	4047.90

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

```
1  /*
2   1. Je SELECT le prix moyen du mètre carré d'un maison (en le calculant)
3   2. J'applique un filtre Maison.
4   3. Je rajoute des filtres sur tous les départements IDF. */
5
6 •  SELECT ROUND(AVG(tr.valeurFonciere/bi.surfaceCarrezLot),2) AS "Prix moyen du mètre carré d'une maison en IDF"
7   FROM transaction tr
8   JOIN bien_immobilier bi USING (idBienImmobilier)
9   JOIN commune cm USING (idCommune)
10  WHERE bi.typeLocal = "Maison"
11  AND bi.surfaceCarrezLot != 0
12  AND tr.valeurFonciere != 0
13  AND cm.codeDepartement IN ('75', '77', '78', '91', '92', '93', '94', '95');
```

Result Grid	
	Prix moyen du mètre carré d'une maison en IDF
▶	3746.77

## 5. Liste des 10 appartements les plus chers avec le département et le nombre de mètres carrés.

```

1  /*
2   1. Je SELECT les colonnes pertinentes.alter
3   2. Je JOIN les tables concernées.
4   3. J'applique le filtre WHERE pour limiter mes résultats au typeLocal Appartement.
5   4. Je tri en ordre décroissant : de la valeur foncière la plus élevée à la plus petite.
6   5. Je limite l'affichage des résultats aux 10 premières lignes.*/
7
8 • SELECT
9     bi.idBienImmobilier AS "Référence",
10    cm.codeDepartement AS "Département",
11    bi.surfaceCarrezLot AS "Nombre de mètres carrés de l'appartement",
12    tr.valeurFonciere AS "Valeur (en euros)"
13  FROM
14    commune cm
15  JOIN bien_immobilier bi USING (idCommune)
16  JOIN transaction tr USING (idBienImmobilier)
17  WHERE bi.typeLocal = "Appartement"
18  ORDER BY tr.valeurFonciere DESC
19  LIMIT 10;

```

	Référence	Département	Nombre de mètres carrés de l'appartement	Valeur (en euros)
•	32252	75	9.10	9000000.00
	21817	91	64.00	8600000.00
	29778	75	20.55	8577713.00
	7771	34	42.77	7620000.00
	29829	75	253.30	7600000.00
	29501	75	139.90	7535000.00
	31950	75	360.95	7420000.00
	32112	75	595.00	7200000.00
	29332	75	122.56	7050000.00
	29492	75	79.38	6600000.00

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

```
2   1. Je calcule les ventes du 1er trimestre 2020.  
3   2. Je calcule les ventes du 2e trimestre 2020.  
4   3. Je calcule le taux d'évolution avec les ventes de chaque trimestre. */  
5  
6 • WITH  
7   ⊖ Table1 AS (SELECT COUNT(tr.idTransaction) AS ventesTrismestre1  
8     FROM transaction tr  
9    WHERE tr.dateTransaction BETWEEN "2020-01-01" AND "2020-03-31"),  
10  
11   ⊖ Table2 AS (SELECT COUNT(tr.idTransaction) AS ventesTrismestre2  
12     FROM transaction tr  
13    WHERE tr.dateTransaction BETWEEN "2020-04-01" AND "2020-06-30")  
14  
15   SELECT t1.ventesTrismestre1 AS "Nombre de ventes du premier trimestre 2020",  
16         t2.ventesTrismestre2 AS "Nombre de ventes du second trimestre 2020",  
17         round(((ventesTrismestre2 - ventesTrismestre1)/ventesTrismestre1*100),2)  
18         AS "Taux d'évolution des ventes entre les 2 trimestres (en %)"  
19   FROM Table1 t1, Table2 t2;
```

	Nombre de ventes du premier trimestre 2020	Nombre de ventes du second trimestre 2020	Taux d'évolution des ventes entre les 2 trimestres (en %)
▶	16776	17393	3.68

## 7. Liste des communes où le nombre de ventes a augmenté d'au moins 20% entre le premier et le second trimestre de 2020

```

6
7 • WITH
8   table1 AS (SELECT cm.nomCommune, COUNT(tr.idTransaction) AS trim1
9     FROM commune cm
10    INNER JOIN bien_immobilier bi USING (idCommune)
11    INNER JOIN transaction tr USING (idBienImmobilier)
12   WHERE tr.dateTransaction BETWEEN "2020-01-01" AND "2020-03-31"
13   GROUP BY 1),
14
15   table2 AS (SELECT cm.nomCommune, count(tr.idTransaction) as trim2
16     FROM commune cm
17    INNER JOIN bien_immobilier bi USING (idCommune)
18    INNER JOIN transaction tr USING (idBienImmobilier)
19   WHERE tr.dateTransaction BETWEEN "2020-04-01" AND "2020-06-30"
20   GROUP BY 1)
21
22   SELECT t1.nomCommune AS "Commune",
23         t1.trim1 AS "Nombre de ventes du premier trimestre 2020",
24         t2.trim2 AS "Nombre de ventes du second trimestre 2020" ,
25         ROUND(((trim2- trim1)/trim1)*100),2 AS "Taux d'évolution ent
26   FROM table1 t1
27   INNER JOIN table2 t2 USING (nomCommune)
28   WHERE ((trim2- trim1)/trim1) >= 0.2
29   ORDER BY 4

```

599 row(s) returned

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
Commune	Nombre de ventes du premier trimestre 2020	Nombre de ventes du second trimestre 2020	Taux d'évolution entre le premier et le second trimestre 2020
TOUQUES	5	6	20.00
LAMBALLE-ARMOR	5	6	20.00
LE COUDRAY	5	6	20.00
MAINVILLIERS	5	6	20.00
CASTELGINEST	5	6	20.00
CROIX	10	12	20.00
ARACHES LA FRASSE	5	6	20.00
THORIGNY SUR MARNE	5	5	20.00
MANTES-LA-VILLE	5	6	20.00
CARPENTRAS	5	6	20.00
GOLBEY	5	6	20.00
LINAS	5	6	20.00
BRETIGNY SUR ORGE	15	18	20.00
MENTON	58	70	20.69
VALENCE	33	40	21.21
TASSIN-LA-DEMI-LUNE	14	17	21.43
VERNOUILLET	9	11	22.22
SAINTE-FARGEAU-PON...	9	11	22.22

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

```

1  /*
2   1. Je calcule le prix du mètre carré d'un T2.
3   2. Je calcule le prix du mètre carré d'un T3.
4   3. Je calcule le taux de variation du prix d'un T2 vers un T3 */
5
6 • WITH
7   table1 AS (SELECT ROUND(AVG((tr.valeurFonciere)/bi.surfaceCarrezLot),2) AS prixm2T2
8     FROM bien_immobilier bi
9       INNER JOIN transaction tr ON ( bi.idBienImmobilier = tr.idBienImmobilier)
10      WHERE bi.typeLocal = "Appartement"
11        AND bi.nbrDePieces = "2"),
12
13   table2 AS (SELECT ROUND(AVG((tr.valeurFonciere)/bi.surfaceCarrezLot),2) AS prixm2T3
14     FROM bien_immobilier bi
15       INNER JOIN transaction tr ON ( bi.idBienImmobilier = tr.idBienImmobilier)
16      WHERE bi.typeLocal = "Appartement"
17        AND bi.nbrDePieces = "3")
18
19   SELECT
20     t1.prixm2T2 AS "Prix du mètre d'un T2",
21     t2.prixm2T3 AS "Prix du mètre d'un T3",
22     ROUND(((t2.prixm2T3 - t1.prixm2T2)/t1.prixm2T2)*100,2) AS "Différen
23   FROM table1 t1, table2 t2

```

<span style="float: left; margin-right: 10px;">Result Grid</span> <span>Filter Rows:</span> <span>Export:</span> <span>Wrap Cell Content:</span>			
	Prix du mètre d'un T2	Prix du mètre d'un T3	Différence du prix au mètre carré entre un T2 et un T3 (en %)
▶	4903.56	4299.90	-12.31

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

```

1  /*
2   1. Je calcule la moyenne des valeurs foncières des communes des départements indiqués.
3   2. Je RANK par département par ordre de valeur foncière décroissante.
4   3. Je rajoute un filtre pour limiter le rank par catégorie en dessous de 4.
5  */
6
7 • WITH
8   rankvaleurcommune AS(
9     SELECT
10       cm.codeDepartement,
11       cm.nomCommune,
12       ROUND(AVG(tr.valeurFonciere),2) AS valeurmoyenne,
13       RANK() OVER (PARTITION BY codeDepartement ORDER BY avg(valeurFonciere) DESC) AS ranking
14     FROM commune cm
15     INNER JOIN bien_immobilier bi USING (idCommune)
16     INNER JOIN transaction tr USING (idBienImmobilier)
17     WHERE codeDepartement IN (6, 13, 33, 59, 69)
18     GROUP BY 1, 2)
19
20   SELECT
21     rvc.codeDepartement AS "Département",
22     rvc.nomCommune AS "Commune",
23     rvc.valeurmoyenne AS "Moyenne des valeurs foncières",
24     rvc.ranking AS "Classement"
25   FROM rankvaleurcommune rvc
26   WHERE rvc.ranking < 4

```

	Département	Commune	Moyenne des valeurs foncières	Classement
▶	13	GIGNAC-LA-NERTHE	330000.00	1
	13	SAINT SAVOURNIN	314425.00	2
	13	CASSIS	310292.63	3
	33	LEGE-CAP-FERRET	549500.64	1
	33	VAYRES	335000.00	2
	33	LE TAILLAN MEDOC	303700.00	3
	59	BERSEE	433202.00	1
	59	CYSOING	408550.00	2
	59	MOUVAUX	331727.14	3
	6	EZE	655000.00	1
	6	SAINT-JEAN-CAP-F...	582088.89	2
	6	MOUANS-SARTOUX	476898.10	3
	69	VILLE SUR JARNIOUX	485300.00	1
	69	LYON 2EME	458667.22	2
	69	LYON 6EME	439409.97	3



## BILAN

- Temps de réalisation : 3 semaines
- Cours OC Algèbre relationnelle
- Cours OC Modélisation UML
- Webinar OC Transfert fichiers Excel vers SDBG

1. Difficultés rencontrées

2. Compétences développées

## 1. DIFFICULTÉS RENCONTRÉES

- ⚠ Aucune connaissance en SQL et BDD au début du projet
- ⚠ Phase de DATAPREP fastidieuse (format date, cellules vides, séparateur de milliers etc...)
- ⚠ COURS OpenClassrooms très longs
- ⚠ Requête 2 ( principe des requêtes imbriquées)
- ⚠ Requête 9 (Rank Over partition et tous les alias qui viennent avec...)

## 2. COMPÉTENCES DÉVELOPÉES

- ✓ Modélisation UML, méthode Mérise
- ✓ Dépendances fonctionnelles
- ✓ Normalisation
- ✓ CRUD
- ✓ Requêtes avancées
- ✓ MySQL, PostgreSQL, SQLite
- ✓ Lucidchart, Draw.io, SQL Power Architect

# CONCLUSION



Découverte de l'univers des BDD

Approche orientée objet

Logique SQL



De Zéro à Héros grâce à la réalisation des livrables,  
une sacrée dose de persévérance et surtout un  
**super mentor.**

**N'hésitez pas à m'envoyer un  
message pour toute question ou tout  
commentaire**

**Merci !**