



Projet 3 - Créez et utilisez une base de données immobilière avec SQL

Valérie Chadeau

Formation Data Analyst – OpenClassrooms

Sommaire

- 1. Contexte**
- 2. Dictionnaire des données**
- 3. Modèle conceptuel des données**
- 4. Modèle conceptuel des données - Aller plus loin**
- 5. Schéma relationnel normalisé en 3^{ème} forme normale**
- 6. Création de la base de données PostgreSQL**
- 7. Alimentation de la base de données**
- 8. Requêtes SQL**

1. CONTEXTE

Data analyst chez Laplace Immo, un réseau national d'agences immobilières, je participe au projet « DATAImmo », projet de création d'un modèle pour mieux prévoir le prix de vente des biens immobiliers.

Je dois :

- * **créer un dictionnaire de données** à partir de données extraites du site open data des Demandes de valeurs foncières (DVF)
- * **proposer un modèle conceptuel des données**
- * **proposer un schéma relationnel normalisé en 3ème forme normale**
- * **créer la base de données à partir d'un script**
- * **alimenter la base de données**
- * **créer des requêtes SQL**



Laplace Immo

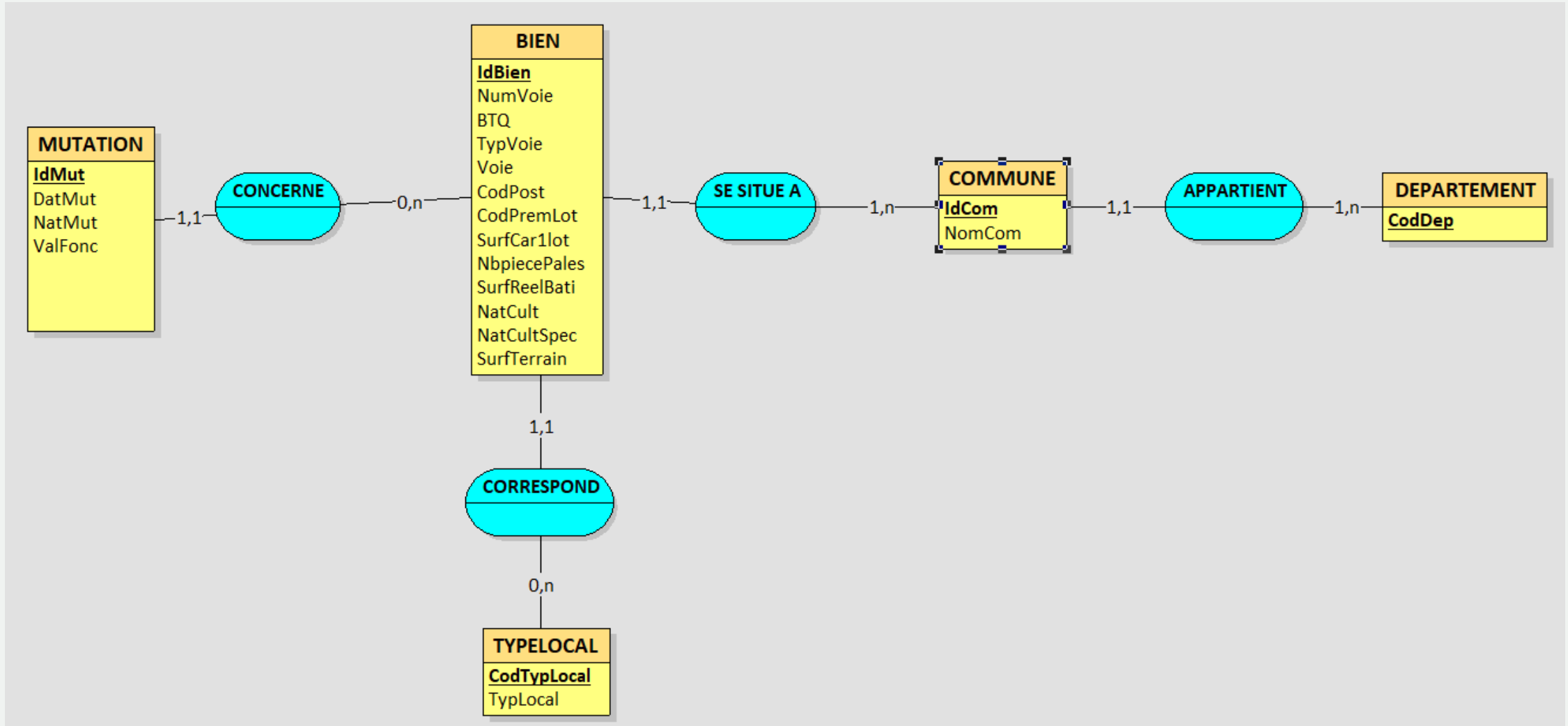
2. Dictionnaire des données

A partir du fichier des demandes de valeurs foncières, récupération des données importantes à stocker :

Numéro ▼	Code propriété ▼	Signification ▼	Type ▼	Observation ▼
1	IdMut	Identifiant mutation	Numérique	Identifiant
2	DatMut	Date mutation	Date	
3	NatMut	Nature mutation	Texte	Longueur : 10
4	ValFonc	Valeur fonciere	Monétaire	
5	IdBien	Identifiant du bien	Numérique	Identifiant
6	NumVoie	Numéro de voie	Numérique	
7	BTQ	Suffixe du numéro de l'adresse	Texte	Longueur : 5
8	TypVoie	Type de voie	Texte	Longueur : 5
9	Voie	Nom de la voie	Texte	Longueur : 50
10	CodPost	Code postal de la commune	Numérique	
11	NomCom	Nom de la commune	Texte	Longueur : 50
12	CodDep	Code du département	Texte	Longueur : 5
13	IdCom	Code Id de la commune	Numérique	Identifiant
14	CodPremLot	Code du premier lot	Numérique ou alphanumérique	
15	SurfCar1lot	SurfCar1lot	Numérique	Décimal
16	CodTypLocal	Code type local	Numérique	Identifiant
17	TypLocal	Type de local	Texte	Longueur : 15
18	SurfReelBati	Surface réelle bati	Numérique	
19	NbPiecePales	Nombre de pieces principales	Numérique	
20	NatCult	Nature culture	Texte	Longueur : 5
21	NatCultSpec	Nature culture speciale	Texte	Longueur : 6
22	SurfTerrain	Surface Terrain	Numérique	

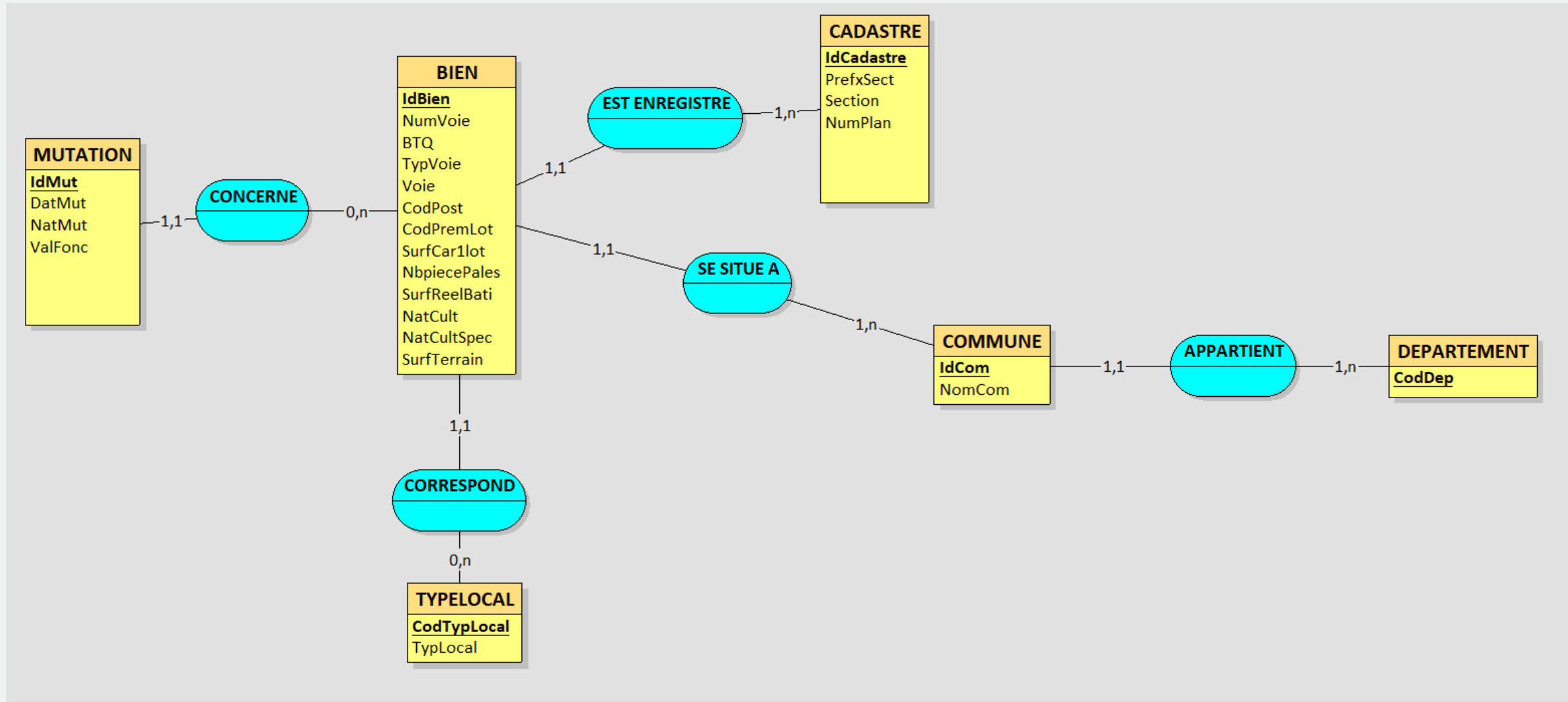
3. Modèle conceptuel des données

- *Elaboration du MCD à partir de l'outil « looping »*
- *Choix des tables réalisé en fonction des informations demandées pour les requêtes SQL*
- *Une autre version plus complète du MCD a été réalisée mais ne fera pas l'objet d'une base de données car elle va au-delà de ce qui est demandé (slide suivante)*



4. Modèle conceptuel des données – Aller plus loin

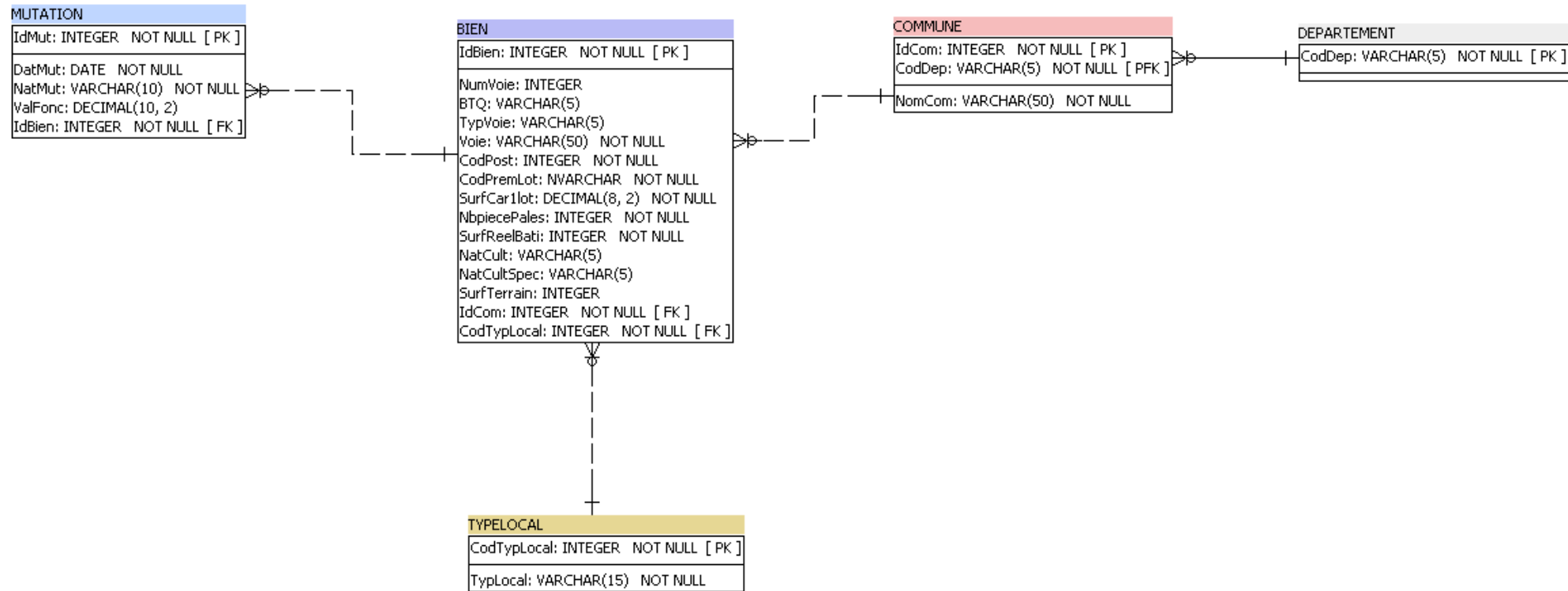
- Optimisation des recherches immobilières par ajout d'informations concernant le cadastre (préfixe, section, numéro de plan)
- Permet la recherche par parcelle cadastrale (cf site <https://app.dvf.etalab.gouv.fr/>)



5. Schéma relationnel normalisé en 3^{ème} forme normale 1/3

Afin de réaliser le schéma relationnel normalisé en 3^{ème} forme normale :

→ *Création du modèle physique de données sous SQL Power Architect*



5. Schéma relationnel normalisé en 3^{ème} forme normale 2/3

→ Passage au schéma relationnel normalisé en 3^{ème} forme normale

Il est nécessaire que :

- *Chaque attribut soit atomique (non constitué par un ensemble de valeurs)*
- *Chaque attribut d'une table soit totalement dépendant de la clé*
- *Tout attribut n'appartenant pas à une clé ne dépend pas d'un attribut non clé*

En **rouge** : clé primaire

En **#bleu** : clé étrangère

MUTATION(***IdMut***, *DatMut*, *NatMut*, *ValFonc*, ***#IdBien***)

BIEN (***IdBien***, *NumVoie*, *BTQ*, *TypVoie*, *Voie*, *CodPost*, *CodPremLot*, *SurfCar1lot*, *NbpiecePales*, *SurfReelBati*, *NatCult*, *NatCultSpec*, *SurfTerrain*, ***#IdCom***, ***#CodTypLocal***, ***#CodTypVoie***, ***#CodVoie***)

COMMUNE (***IdCom***, *NomCom*, ***#CodDep***)

DEPARTEMENT (***CodDep***)

TYPELOCAL (***CodTypLocal***, *TypLocal*)

5. Schéma relationnel normalisé en 3^{ème} forme normale 3/3

Nb : Cependant on note que la table DEPARTEMENT ne contient que la clé car le nom du département n'est pas présent dans les data fournis, ce qui n'est pas souhaitable pour une table.

Pour des raisons d'optimisation de temps d'exécution des requêtes, j'ai « dénormalisé » le schéma, la 2^{ème} FN n'est plus respectée, et j'ai intégré la clé « Cod Dep » à la table « COMMUNE » qui devient donc :

COMMUNE (~~IdCom~~, ~~NomCom~~, ~~#CodDep~~)

DEPARTEMENT (~~CodDep~~)



COMMUNE (~~IdCom~~, ~~NomCom~~, CodDep)

6. Création de la base de données PostgreSQL 1/3

- Le système de gestion de base de données choisi ici est PostgreSQL.
- Script généré à partir du modèle physique de données sous SQL Power Architect et intégré via la plateforme d'administration PgAdmin :

```
CREATE TABLE public.COMMUNE (  
    IdCom INTEGER NOT NULL,  
    NomCom VARCHAR(50) NOT NULL,  
    CodDep VARCHAR(5) NOT NULL,  
    CONSTRAINT commune_pk PRIMARY KEY (IdCom)  
);  
COMMENT ON COLUMN public.COMMUNE.IdCom IS 'Code Id commune';  
COMMENT ON COLUMN public.COMMUNE.NomCom IS 'Nom commune';  
COMMENT ON COLUMN public.COMMUNE.CodDep IS 'Code département';
```

```
CREATE TABLE public.TYPELOCAL (  
    CodTypLocal INTEGER NOT NULL,  
    TypLocal VARCHAR(15) NOT NULL,  
    CONSTRAINT typelocal_pk PRIMARY KEY (CodTypLocal)  
);  
COMMENT ON COLUMN public.TYPELOCAL.CodTypLocal IS 'Code type local';
```

6. Création de la base de données PostgreSQL 2/3

```
CREATE SEQUENCE public.bien_idbien_seq;
```

```
CREATE TABLE public.BIEN (  
    IdBien INTEGER NOT NULL DEFAULT nextval('public.bien_idbien_seq'),  
    NumVoie INTEGER,  
    BTQ VARCHAR(5),  
    TypVoie VARCHAR(5),  
    Voie VARCHAR(50) NOT NULL,  
    CodPost INTEGER NOT NULL,  
    CodPremLot NVARCHAR NOT NULL,  
    SurfCar1lot NUMERIC(8,2) NOT NULL,  
    NbpiecePales INTEGER NOT NULL,  
    SurfReelBati INTEGER NOT NULL,  
    NatCult VARCHAR(5),  
    NatCultSpec VARCHAR(5),  
    SurfTerrain INTEGER,  
    IdCom INTEGER NOT NULL,  
    CodTypLocal INTEGER NOT NULL,  
    CONSTRAINT bien_pk PRIMARY KEY (IdBien)  
);  
COMMENT ON COLUMN public.BIEN.NumVoie IS 'Numéro de voie';  
COMMENT ON COLUMN public.BIEN.BTQ IS 'Suffixe du numéro de l"adresse';  
COMMENT ON COLUMN public.BIEN.CodPost IS 'Code postal';  
COMMENT ON COLUMN public.BIEN.CodPremLot IS 'Code du premier lot';  
COMMENT ON COLUMN public.BIEN.SurfCar1lot IS 'Surface Carrez lot 1';  
COMMENT ON COLUMN public.BIEN.NbpiecePales IS 'Nombre de pièces principales';  
COMMENT ON COLUMN public.BIEN.SurfReelBati IS 'Surface réelle du bâti';  
COMMENT ON COLUMN public.BIEN.NatCult IS 'Nature culture';  
COMMENT ON COLUMN public.BIEN.NatCultSpec IS 'Nature culture speciale';  
COMMENT ON COLUMN public.BIEN.SurfTerrain IS 'Surface Terrain';  
COMMENT ON COLUMN public.BIEN.IdCom IS 'Code Id commune';  
COMMENT ON COLUMN public.BIEN.CodTypLocal IS 'Code type local';
```

6. Création de la base de données PostgreSQL 3/3

```
ALTER SEQUENCE public.bien_idbien_seq OWNED BY public.BIEN.IdBien;
```

```
CREATE SEQUENCE public.mutation_idmut_seq;
```

```
CREATE TABLE public.MUTATION (  
    IdMut INTEGER NOT NULL DEFAULT nextval('public.mutation_idmut_seq'),  
    DatMut DATE NOT NULL,  
    NatMut VARCHAR(10) NOT NULL,  
    ValFonc NUMERIC(10,2),  
    IdBien INTEGER NOT NULL,  
    CONSTRAINT mutation_pk PRIMARY KEY (IdMut)  
);  
COMMENT ON COLUMN public.MUTATION.DatMut IS 'Date de mutation';  
COMMENT ON COLUMN public.MUTATION.NatMut IS 'Nature mutation';
```

```
ALTER SEQUENCE public.mutation_idmut_seq OWNED BY public.MUTATION.IdMut;
```

```
ALTER TABLE public.BIEN ADD CONSTRAINT commune_bien_fk  
FOREIGN KEY (IdCom)  
REFERENCES public.COMMUNE (IdCom)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

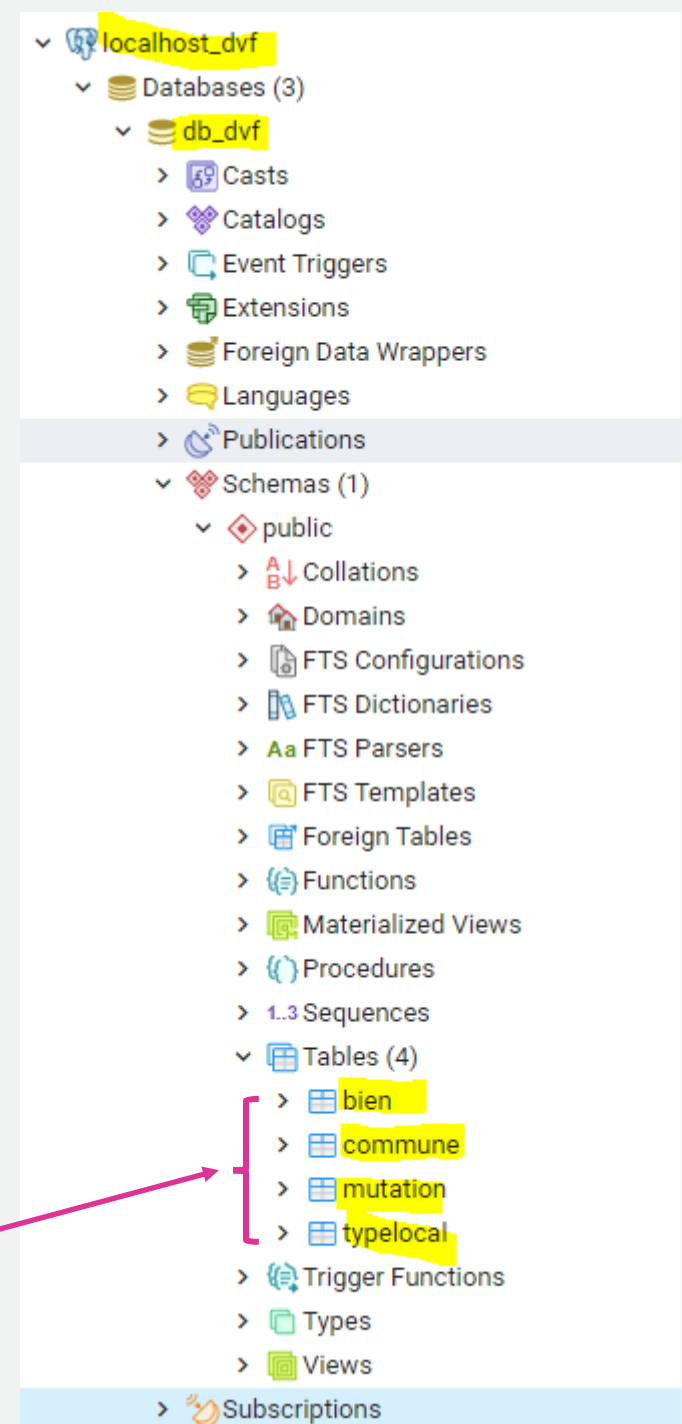
```
ALTER TABLE public.BIEN ADD CONSTRAINT typelocal_bien_fk  
FOREIGN KEY (CodTypLocal)  
REFERENCES public.TYPELOCAL (CodTypLocal)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

```
ALTER TABLE public.MUTATION ADD CONSTRAINT bien_mutation_fk  
FOREIGN KEY (IdBien)  
REFERENCES public.BIEN (IdBien)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
NOT DEFERRABLE;
```

7. Alimentation de la base de données 1/6

Elle est réalisée en plusieurs étapes :

- **Génération des fichiers** d'alimentation des tables via Excel (4 fichiers : 1 par table)
- **Correction des fichiers** :
 - Modification format code postal (suppression des décimales)
 - Modification séparateur de décimale (« , » en « . ») pour prise en compte Postgresql
 - Modification du format de date (sans l'heure) → modification format date cellule JJ/MM/AAAA
 - ***nom de voie erroné** : valeur « 11/11/1918 00:00:00 » au lieu de « 11 novembre 1918 » et « 01/06/1907 00:00:00 » au lieu de « 01 juin 1907 » (Excel a transformé le nom de voie en date)*
 - ***code postal erroné** : valeur « nan » au lieu de 20090 suite à vérification sur le site France Adresse (lié au code ID commune erroné)*
- **Conversion des fichiers** au format « csv » pour intégration dans pgadmin
- **Pré-requis** à la création des tables dans pgadmin :
 - création serveur (localhost_dvf)
 - création base de données (db_dvf)
- **Intégration des fichiers** dans PgAdmin pour alimentation des 4 tables



7. Alimentation de la base de données 2/6

→ Table « bien »

▼ Tables (4)

▼ bien

▼ Columns (15)

idbien
numvoie
btq
typvoie
voie
codpost
codpremlot
surfcar1lot
nbpiecepales
surfreelbati
natcult
natcultspec
surfterrain
idcom
codtyplocal

> Constraints

> Indexes

db_dvf/adm_dvf@localhost_dvf

Query Editor Query History

```
1 select count(*) from bien;
```

Data Output Explain Messages Notifications

	count bigint	
1	34169	

7. Alimentation de la base de données 3/6

➔ Table « bien »

db_dvf/adm_dvf@localhost_dvf

Query EditorQuery History

1select * from bien;

Data Output

Explain

Messages

Notifications

	idbien [PK] integer	numvoie integer	btq character varying (5)	typvoie character varying (5)	voie character varying (50)	codpost integer	codpremi character v	surfcartlot numeric (10,2)	nbpiecepales integer	surfreelbati integer	natcult character varying (5)	natcultspec character varying (5)	surfterrain integer	idcom integer	codtyplocal integer
1	1	347	[null]	RUE	DU CHATEAU	1170	12	48.22	3	48	[null]	[null]	[null]	1	2
2	2	20	B	RUE	MARCEAU	6000	99	80.25	3	82	[null]	[null]	[null]	142	2
3	3	4	[null]	BD	EDOUARD BAUDOIN	6160	132	39.11	1	40	[null]	[null]	[null]	205	2
4	4	550	[null]	RTE	DES VESPINS RN7	6700	242	27.51	1	27	[null]	[null]	[null]	228	2
5	5	9300	[null]	RES	LES ARPEGES BD DES ABA	13400	218	47.33	2	47	[null]	[null]	[null]	326	2
6	6	360	[null]	AV	DU PRADO	13008	112	70.84	3	70	[null]	[null]	[null]	327	2
7	7	27	[null]	RUE	DU GRAND MADIER	13600	2	25.31	1	24	[null]	[null]	[null]	331	2
8	8	5076	F	PARC	DESSUARD	13012	1080	67.19	3	66	[null]	[null]	[null]	343	2
9	9	30	[null]	ALL	DES NOISETIERS	14100	32	105.37	4	99	[null]	[null]	[null]	366	1
10	10	1194	[null]	RUE	DE NORMANDIE	14510	155	18.89	1	19	[null]	[null]	[null]	372	2
11	11	11	[null]	RUE	ROUGET DE L ISLE	17000	6	31.99	2	34	[null]	[null]	[null]	390	2
12	12	13	[null]	RUE	BERTHE MORISOT	25000	6	96.21	5	100	[null]	[null]	[null]	480	1
13	13	1	[null]	RUE	DU POHER	29000	305	30.86	1	31	[null]	[null]	[null]	595	2
14	14	2	[null]	RUE	DES JARDINS	29290	3	66.21	3	67	[null]	[null]	[null]	602	2
15	15	5	[null]	AV	DU COMMANDANT TAILLANDIER	31500	9	45.58	2	46	[null]	[null]	[null]	732	2
16	16	9	[null]	RUE	DE BELFORT	33160	207	51.40	2	51	[null]	[null]	[null]	782	2
17	17	15	[null]	RUE	PAUL DENUCE	33800	8	23.20	1	27	[null]	[null]	[null]	796	2
18	18	176	[null]	RUE	SAINTE CATHERINE	33000	12	13.10	1	12	[null]	[null]	[null]	783	2
19	19	822	G	[null]	LA BAYNASSE SUD	33121	89	29.00	2	29	[null]	[null]	[null]	877	2
20	20	183	[null]	RUE	MARC ANTOINE MENARD	34400	1	16.79	1	18	[null]	[null]	[null]	907	2
21	21	8	[null]	PL	DES ETATS DU LANGUEDOC	34120	3	68.22	3	85	[null]	[null]	[null]	948	2
22	22	50	[null]	AV	ANDRE CHASSEFIERE	34340	19	51.40	2	51	[null]	[null]	[null]	949	2

7. Alimentation de la base de données 4/6

→ Table « mutation »

Sequences

Tables (4)

bien

commune

mutation

Columns (5)

idmut

datmut

natmut

valfonc

idbien

Constraints

Indexes

RLS Policies

Rules

Triggers

typelocal

db_dvf/adm_dvf@localhost_dvf

Query EditorQuery History

1select * from mutation;

Data OutputExplainMessagesNotifications

	idmut [PK] integer	datmut date	natmut character varying (10)	valfonc numeric (10,2)	idbien integer
1	1	2020-01-02	Vente	165000.00	1
2	2	2020-01-02	Vente	229500.00	2
3	3	2020-01-02	Vente	355680.00	3
4	4	2020-01-02	Vente	125000.00	4
5	5	2020-01-02	Vente	90000.00	5
6	6	2020-01-02	Vente	298100.00	6
7	7	2020-01-02	Vente	93000.00	7
8	8	2020-01-02	Vente	163500.00	8
9	9	2020-01-02	Vente	136000.00	9
10	10	2020-01-02	Vente	53000.00	10
11	11	2020-01-02	Vente	125900.00	11
12	12	2020-01-02	Vente	234000.00	12
13	13	2020-01-02	Vente	46210.00	13
14	14	2020-01-02	Vente	129000.00	14
15	15	2020-01-02	Vente	122500.00	15
16	16	2020-01-02	Vente	156000.00	16
17	17	2020-01-02	Vente	86025.00	17
18	18	2020-01-02	Vente	79000.00	18
19	19	2020-01-02	Vente	81795.00	19
20	20	2020-01-02	Vente	139900.00	20
21	21	2020-01-02	Vente	108100.00	21
22	22	2020-01-02	Vente	163870.00	22
23	23	2020-01-02	Vente	205000.00	23

db_dvf/adm_dvf@localhost_dvf

Query EditorQuery History

1select count(*) from mutation;

Data OutputExplainMessagesNotifications

	count bigint
1	34169

7. Alimentation de la base de données 5/6

→ Table « commune »

> 1.3 Sequences

▼ Tables (4)

> bien

▼ commune

▼ Columns (3)

idcom

nomcom

coddep

> Constraints

> Indexes

> RLS Policies

> Rules

> Triggers

db_dvf/adm_dvf@localhost_dvf

Query EditorQuery History

1 select count(*) from commune;

Data OutputExplainMessagesNotifications

	count bigint
1	3214

db_dvf/adm_dvf@localhost_dvf

Query EditorQuery History

1 select * from commune;


Data OutputExplainMessagesNotifications

	idcom [PK] integer	nomcom character varying (50)	coddep character varying (5)
1	0	SAINT-ETIENNE-DU-BOIS	1
2	1	CHEVRY	1
3	2	DIVONNE-LES-BAINS	1
4	3	PERON	1
5	4	VALSERHONE	1
6	5	CULOZ	1
7	6	ST-GENIS-POUILLY	1
8	7	OYONNAX	1
9	8	ST-GERMAIN-DE-JOUX	1
10	9	GEX	1
11	10	AMBERIEU-EN-BUGEY	1
12	11	CESSY	1
13	12	SAULT-BRENAZ	1
14	13	FERNEY-VOLTAIRE	1
15	14	SEGNY	1
16	15	LAGNIEU	1
17	16	LE POIZAT-LALLEYRIAT	1
18	17	BELLEY	1
19	18	ORNEX	1
20	19	DORTAN	1
21	20	PREVESSIN-MOENS	1
22	21	BRENOD	1
23	22	SAINT-JEAN-LE-VIEUX	1

7. Alimentation de la base de données 6/6

→ Table « typelocal »



▼	Tables (4)
>	bien
>	commune
>	mutation
▼	typelocal
▼	Columns (2)
	codtyplocal
	typlocal
>	Constraints
>	Indexes
>	RLS Policies
>	Rules
>	Triggers
>	Trigger Functions


db_dvf/adm_dvf@localhost_dvf ▾

Query Editor
Query History

```
1 select * from typelocal;
```

Data Output
Explain
Messages
Notifications

	<div>codtyplocal</div> <div>[PK] integer </div>	<div>typlocal</div> <div>character varying (15) </div>	
1	1	Maison	
2	2	Appartement	

db_dvf/adm_dvf@localhost_dvf

Query Editor

Query History

1

select count(*) from typelocal;

Data Output

Explain

Messages

Notifications

count
bigint

1

2

8. Requêtes SQL 1/11

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

db_dvf/adm_dvf@localhost_dvf

Query EditorQuery History

```
1 SELECT COUNT(*)
2 FROM mutation M
3 INNER JOIN (
4     bien B INNER JOIN typelocal T ON B.codtyplocal = T.codtyplocal
5 ) ON M.idmut=B.idbien
6 WHERE datmut BETWEEN '01/01/2020' AND '30/06/2020'
7 AND T.typlocal ='Appartement';
8
```

Data OutputExplainMessagesNotifications

	count bigint
1	31378

8. Requêtes SQL 2/11

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

```
db_dvf/adm_dvf@localhost_dvf v
Query Editor  Query History

1  SELECT B.nbpiecepales "Nb pièces",
2  ROUND((100*COUNT(*)::numeric/(SELECT COUNT(*) FROM mutation M
3  INNER JOIN (
4  bien B INNER JOIN typelocal T ON B.codtyplocal = T.codtyplocal
5  ) ON M.idmut=B.idbien
6  WHERE T.typlocal ='Appartement')::numeric ),3) "Proportion de ventes
7  par nombre de pièces en %"
8  FROM mutation M
9  INNER JOIN (
10 bien B INNER JOIN typelocal T ON B.codtyplocal = T.codtyplocal
11 ) ON M.idmut=B.idbien
12 WHERE T.typlocal ='Appartement'
13 GROUP BY B.nbpiecepales
14 ORDER BY B.nbpiecepales;
15
16
```

Data Output			Explain	Messages	Notifications
	Nb pièces integer	Proportion de ventes par nombre de pièces en % numeric			
1	0	0.096			
2	1	21.477			
3	2	31.178			
4	3	28.574			
5	4	14.214			
6	5	3.550			
7	6	0.650			
8	7	0.172			
9	8	0.054			
10	9	0.025			
11	10	0.006			
12	11	0.003			

8. Requêtes SQL 3/11

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

```
db_dvf/adm_dvf@localhost_dvf
Query Editor  Query History
1 SELECT C.coddep "code département", ROUND(AVG(M.valfonc/B.Surfcarlot),2)
2 FROM commune C
3 INNER JOIN (
4 Bien B INNER JOIN mutation M ON B.idbien = M.idmut
5 ) ON C.idcom = B.idcom
6 GROUP BY C.coddep
7 ORDER BY ROUND(AVG(M.valfonc/B.Surfcarlot),2) DESC
8 LIMIT 10;
9
```

Data Output		Explain	Messages	No
	code département character varying (5)		round numeric	
1	75		12052.89	
2	92		7219.39	
3	94		5343.28	
4	6		4700.33	
5	74		4667.13	
6	93		4344.78	
7	78		4225.25	
8	69		4059.31	
9	2A		4026.97	
10	33		3764.14	

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

Les départements Ile de France sont : 75, 77, 78, 91, 92, 93 et 94

db_dvf/adm_dvf@localhost_dvf

Query EditorQuery History

```
1 SELECT COALESCE('Région IDF') "Région", ROUND(AVG(M.valfonc/ B.Surfcarlot),2) "Prix m2"
2 FROM commune C
3 INNER JOIN (
4   Bien B INNER JOIN mutation M ON B.idbien = M.idmut
5         INNER JOIN typelocal T ON B.codtyplocal = T.codtyplocal
6 ) ON C.idcom = B.idcom
7 WHERE T.typlocal = 'Maison'
8 AND C.coddep IN ('75', '77', '78', '91', '92', '93','94');
9
10
11
```

Data Output		Explain	Messages	Notifications
	Région text	Prix m2 numeric		
1	Région IDF	3850.71		

8. Requêtes SQL

5/11

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

```
db_dvf/adm_dvf@localhost_dvf v
Query Editor  Query History

1  SELECT C.coddep "Code dépt",B.Surfcarl1ot "Surface", M.valfonc "Valeur foncière"
2  FROM mutation M
3  INNER JOIN (
4  Bien B  INNER JOIN commune C ON B.idcom = C.idcom
5      INNER JOIN typelocal T ON B.codtyplocal = T.codtyplocal
6  ) ON M.idmut=B.idbien
7  WHERE T.typlocal = 'Appartement'
8  AND M.valfonc IS NOT NULL
9  ORDER BY M.valfonc desc
10 LIMIT 10;
```

	Data Output	Explain	Messages	Notifications
	Code dépt character varying (5)		Surface numeric (10,2)	Valeur foncière numeric (10,2)
1	75		9.10	9000000.00
2	91		64.00	8600000.00
3	75		20.55	8577713.00
4	75		42.77	7620000.00
5	75		253.30	7600000.00
6	75		139.90	7535000.00
7	75		360.95	7420000.00
8	75		595.00	7200000.00
9	75		122.56	7050000.00
10	75		79.38	6600000.00

8. Requêtes SQL

6/11

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

```
db_dvf/adm_dvf@localhost_dvf
Query Editor  Query History

1  WITH
2  ventepremtrim (premtrim) AS(
3  SELECT COUNT(*)
4  FROM mutation M
5  INNER JOIN  bien B ON M.idmut=B.idbien
6  WHERE M.datmut BETWEEN '01/01/2020' AND '31/03/2020'
7  AND M.valfonc is not null
8  ),
9
10 ventedeuxtrim(deuxtrim) AS(
11 SELECT COUNT(*)
12 FROM mutation M
13 INNER JOIN  bien B ON M.idmut=B.idbien
14 WHERE M.datmut BETWEEN '01/04/2020' AND '30/06/2020'
15 AND M.valfonc is not null
16 )
17
18 SELECT premtrim "Premier trimestre", deuxtrim "Deuxième trimestre",
19 ROUND((deuxtrim-premtrim)*100/(premtrim*1.0),2) "Taux évolution en %"
20 FROM ventepremtrim, ventedeuxtrim;
21
```

	Data Output	Explain	Messages	Notifications
	Premier trimestre bigint		Deuxième trimestre bigint	Taux évolution en % numeric
1	16769		17382	3.66

8. Requêtes SQL

7/11

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

```
db_dvf/adm_dvf@localhost_dvf v
Query Editor  Query History

1  WITH
2  ventepremtrim (idcom,pretrim) AS(
3  SELECT b.idcom,count(*)
4  FROM mutation M
5  INNER JOIN
6  (bien B INNER JOIN commune C ON B.idcom = C.idcom)
7  ON M.idmut=B.idbien
8  WHERE M.datmut BETWEEN '01/01/2020' AND '31/03/2020'
9  AND M.valfonc IS NOT NULL
10 GROUP BY b.idcom
11 ORDER BY b.idcom
12 ),
13 ventedeuxtrim(idcom,deuxtrim) AS(
14 SELECT b.idcom,count(*)
15 FROM mutation M
16 INNER JOIN
17 (bien B INNER JOIN commune C ON B.idcom = C.idcom)
18 ON M.idmut=B.idbien
19 WHERE M.datmut BETWEEN '01/04/2020' AND '30/06/2020'
20 AND M.valfonc IS NOT NULL
21 GROUP BY b.idcom
22 ORDER BY b.idcom
23 )
```

```
25 SELECT C.nomcom "Nom commune",
26 ROUND((D.deuxtrim-P.pretrim)*100/P.pretrim::numeric,2) "Evolution"
27 FROM ventepremtrim P
28 INNER JOIN ventedeuxtrim D ON P.idcom= D.idcom
29 INNER JOIN commune C ON P.idcom = C.idcom
30 WHERE ROUND((D.deuxtrim-P.pretrim)*100/P.pretrim::numeric,2) >= 20
31 ORDER BY C.nomcom;
```

8. Requêtes SQL

8/11

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

Il y a eu 607 lignes en résultat.

Extrait des résultats :

	Data Output	Explain	Messages	Notifications
	 Nom commune character varying (50)			Evolution numeric
1	ABBEVILLE			350.00
2	ABLON-SUR-SEINE			160.00
3	AGDE			104.55
4	AIGUES-MORTES			83.33
5	AIRE-SUR-L'ADOUR			400.00
6	AJACCIO			20.00
7	ALENCON			100.00
8	AMPUIS			100.00
9	ANCENIS-SAINT-GEREON			100.00
10	ANDRESY			33.33
11	ANGLET			100.00
12	ANNECY			42.86
13	ANNECY			85.71
14	ANNECY			25.00
15	ANNEMASSE			100.00
16	ANTIBES			20.83
17	APT			250.00
18	ARCACHON			20.00
19	ARCUEIL			137.50
20	ARDON			33.33

21	ARGENTEUIL	68.97
22	ARMENTIERES	125.00
23	ARNAS	100.00
24	ARPAJON	122.22
25	AUBERGENVILLE	200.00
26	AUBERVILLIERS	48.00
27	AUCAMVILLE	50.00
28	AUCH	200.00
29	AULT	33.33
30	AUTUN	33.33
31	AUXERRE	77.78
32	AVIGNON	200.00
33	AVIGNON	152.00
34	AVRANCHES	200.00
35	AYTRE	100.00
36	BALLAN MIRE	100.00
37	BALMA	200.00
38	BARCELONNETTE	150.00
39	BARRE-DE-MONTS (LA)	100.00
40	BARSAC	100.00

8. Requêtes SQL

9/11

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

```
db_dvf/adm_dvf@localhost_dvf
Query Editor  Query History

1  WITH
2  appart2p (prixm22) AS(
3  SELECT ROUND(AVG(M.valfonc/B.surfcarllot)::numeric,2)
4  FROM bien B
5  INNER JOIN mutation M ON B.idbien=M.idmut
6  INNER JOIN typelocal T ON B.codtyplocal = T.codtyplocal
7  WHERE B.nbpiecepales = 2
8  AND T.typlocal = 'Appartement'
9  ),
10 appart3p (prixm23) AS(
11 SELECT ROUND(AVG(M.valfonc/B.surfcarllot)::numeric,2)
12 FROM bien B
13 INNER JOIN mutation M ON B.idbien=M.idmut
14 INNER JOIN typelocal T ON B.codtyplocal = T.codtyplocal
15 WHERE B.nbpiecepales = 3
16 AND T.typlocal = 'Appartement'
17 )
18 SELECT prixm22 "Prix au m2 appart T2",prixm23 "Prix au m2 appart T3",
19 ROUND((prixm23-prixm22)*100/prixm22::numeric,2) "Ecart en % prix au m2 T2 T3"
20 FROM appart2p, appart3p;
21
```

	Data Output	Explain	Messages	Notifications
	Prix au m2 appart T2 numeric	Prix au m2 appart T3 numeric	Ecart en % prix au m2 T2 T3 numeric	
1	4908.58	4299.90	-12.40	

8. Requêtes SQL

10/11

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




```
db_dvf/adm_dvf@localhost_dvf
Query Editor  Query History

1  WITH
2  selcom (idbien, coddep,nomcom,idcom) AS(
3  SELECT B.idbien, C.coddep,C.nomcom,C.idcom
4  FROM commune C
5  INNER JOIN bien B ON C.idcom = B.idcom
6  WHERE  c.coddep in ('6','13','33','59','69')
7  ORDER BY idbien
8  ),
9  valfonccom(coddep,nomcom,moyfonc) AS(
10 SELECT S.coddep,S.nomcom, ROUND(AVG(M.valfonc),2)
11 FROM selcom S
12 INNER JOIN mutation M ON S.idbien=M.idmut
13 GROUP BY S.coddep,S.nomcom
14 ORDER BY S.coddep,S.nomcom
15 ),
16 top3val AS
17 (SELECT ROW_NUMBER() OVER(PARTITION BY coddep ORDER BY moyfonc DESC)
18  AS POSITION, coddep, nomcom, moyfonc
19 FROM valfonccom
20 )
21
22 SELECT coddep "Code département", nomcom "Nom commune", moyfonc "Moyenne foncière"
23 FROM top3val
24 WHERE POSITION < 4
25 ORDER BY coddep,moyfonc desc;
26
```

8. Requêtes SQL

11/11

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

	Data Output	Explain	Messages	Notifications
	 Code département character varying (5)		Nom commune character varying (50)	 Moyenne foncière numeric
1	13		GIGNAC-LA-NERTHE	330000.00
2	13		SAINT SAVOURNIN	314425.00
3	13		CASSIS	313416.88
4	33		LEGE-CAP-FERRET	549500.64
5	33		VAYRES	335000.00
6	33		ARCACHON	307435.93
7	59		BERSEE	433202.00
8	59		CYSOING	408550.00
9	59		HALLUIN	322250.00
10	6		SAINT-JEAN-CAP-FERRAT	968750.00
11	6		EZE	655000.00
12	6		MOUANS-SARTOUX	476898.10
13	69		VILLE SUR JARNIOUX	485300.00
14	69		LYON 2EME	455217.27
15	69		LYON 6EME	426968.25