Création et utilisation de la base de données immobilières

Marie G.

19/02/2025

Laplace Immo

Création d’une base de données immobilières

À partir des données gouvernementales de transactions immobilières du 1er semestre 2020 en France

1. Données initiales et conformité RGPD
2. Définition du schéma relationnel et du dictionnaire de données
3. Définition de la stratégie d’enregistrement des données
4. Implémentation et chargement de la base de données
5. Vérification de son bon fonctionnement : requêtes SQL et résultats

1. Données initiales et conformité RGPD

valeurs_foncières.xlsx
34 169 lignes

Nom de colonne (non vides)	Exemple de valeurs
No disposition	1
Date mutation	**2020/01/08**
Nature mutation	Vente
Valeur fonciere	**283000**
No voie	**123**
B/T/Q	**B**
Code type de voie	1
Type de voie	**AV**
Code voie	905
Voie	**SAINT MICHEL**
Code ID commune	1357
Code postal	45160
Commune	OLIVET
Code departement	**45**
Code commune	**232**
Section	BK
No plan	281
1er lot	123
Surface Carrez du 1er lot	**116,72**
Nombre de lots	1
Code type local	1
Type local	**Maison**
Surface reelle bati	**109**
Nombre pieces principales	**4**
Nature culture	AG
Nature culture speciale	IMM
Surface terrain	1112
Nom de l'acquereur	**DUPONT**

RGPD : Nom acquéreur à supprimer des données

3 tables **issues des données gouvernementales** :

- **'Valeurs_foncières'**, décrivant les transactions immobilières enregistrées par notaires (ici sur le 1er semestre 2020) avec
  - le prix et la date de vente et le nom de l'acquéreur
  - les caractéristiques de construction et d'urbanisme du bien (ou du 1er lot de chaque vente en cas de vente en bloc)
- **'Référentiel_géographique'**, décrivant de multiples données géographiques et administratives pour chaque commune
- **'Données_communes'**, décrivant :
  - les principaux codes administratifs de chaque commune
  - sa population

➔ **Présélection des données à récupérer dans notre base de données :**

- Données de la vente
- Données caractéristiques du bien influençant la valeur
- Données géographiques

données_communes.xlsx *34 991 lignes*

Nom de colonne	Exemple de valeurs
CODREG	84
CODDEP	**01**
CODARR	02
CODCAN	08
CODCOM	**001**
COM	**L'Abergement-Clémenciat**
PMUN	779
PCAP	19
PTOT	**798**

référentiel_géographique.xlsx
38 916 lignes

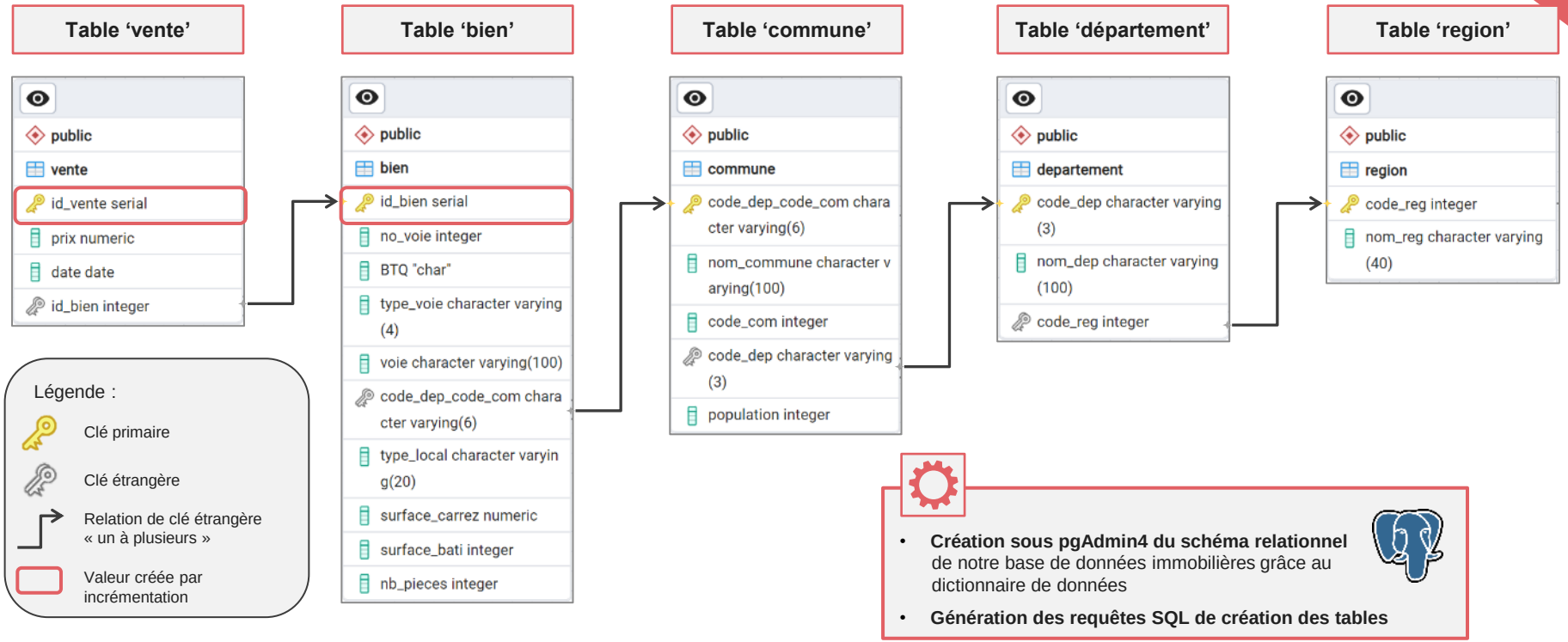
Nom de colonne	Exemple de valeurs
regrgp_nom	Province
reg_nom	**Auvergne-Rhône-Alpes**
reg_nom_old	Rhône-Alpes
aca_nom	Lyon
dep_nom	**Ain**
com_code	**01004**
com_code1	1004
com_code2	1004
com_id	C01004
com_nom_maj_court	AMBERIEU EN BUGEY
com_nom_maj	AMBERIEU-EN-BUGEY
com_nom	**Ambérieu-en-Bugey**
uu_code	1303
uu_id	UU01303
uucr_id	UU01303
uucr_nom	Ambérieu-en-Bugey
ze_id	ZE8201
dep_code	1
dep_id	D001
dep_nom_num	Ain (01)
dep_num_nom	01 - Ain
aca_code	10
aca_id	A10
reg_code	**84**
reg_id	R84
reg_code_old	82
reg_id_old	R82
fd_id	FD111
fr_id	FR11
fe_id	FE1
uu_id_99	UU01303
au_code	2
au_id	AU002
auc_id	AU002
auc_nom	Lyon
uu_id_10	UU01302
geolocalisation	45.9608475114,5.3729257777

2. Définition du dictionnaire de données

- Définition des tables qui constitueront notre base de données, avec une attention portée au respect de la norme 3NF : **pas de dépendances fonctionnelles entre colonnes au sein d'une table (hors PK)**
- Définition des colonnes de chaque table, avec définition des clés primaires **:**
  - **Par incrémentation d'id_vente (table 'vente')**
  - **Par incrémentation d'id_bien (table 'bien)**
  - **'code_dep_code_com'** : créé par concaténation du code département et du code commune
- Identification des clés étrangères
- Identification par analyse des données des fichiers d'origine :
  - du type des données,
  - de la longueur des 'varchar'
  - des règles de gestion et de calcul

DICTIONNAIRE DES DONNÉES						
CODE	SIGNIFICATION	TYPE	LONGUEUR	NATURE	REGLE DE GESTION	REGLE DE CALCUL
Table 'vente'						
id_vente	ID dans la base de données	Integer	NC	Elémentaire	Clé primaire - Not null	Incrémentation
prix	Valeur de la vente	Float	NC	Elémentaire	Not null	
date	date de la vente	Date	NC	Elémentaire	Not null - jj/mm/aaaa	
id_bien	ID du bien dans la table bien	Integer	NC	Elémentaire	Clé étrangère (bien) - Not null	
Table 'bien'						
id_bien	ID du bien dans la table	Integer	NC	Elémentaire	Clé primaire - Not null	Incrémentation
no_voie	Numéro de voie du bien	Integer	NC	Elémentaire		
BTQ	bis/ter…	Char	1	Elémentaire		
type_voie	Type de la voie (rue, avenue…)	Varchar	4	Elémentaire		
voie	Nom de la voie	Varchar	100	Elémentaire		
code_dep_code_com	Code commune unique dans la table commune	Varchar	6	Elémentaire	Clé étrangère (commune) - Not null	Code_dep + code_com
type_local	Type de local (appartement, maison..)	Varchar	20	Elémentaire		
surface_carrez	Surface loi carrez du bien	Float	NC	Elémentaire	Not null	
surface_bati	Surface totale du bâti	Integer	NC	Elémentaire		
nb_pieces	Nombre de pièces du bien	Integer	NC	Elémentaire		
Table 'commune'						
code_dep_code_com	Code unique commune formé du code département et du code commune	Varchar	6	Concaténer	Clé primaire - Not null	Code_dep + code_com
nom_commune	Nom de la commune	Varchar	100	Elémentaire	Not null	
code_com	Code de la commune	Integer	NC	Elémentaire	Not null	
code_dep	Code du département de la commune	Varchar	3	Elémentaire	Clé étrangère (département) - Not null	
population	Population totale	Integer	NC	Elémentaire	Not null	
Table 'département'						
code_dep	Code département	Varchar	3	Elémentaire	Clé primaire - Not null	
nom_dep	Nom du département	Varchar	100	Elémentaire	Not null	
code_reg	Code de la région du département	Integer	NC	Elémentaire	Clé étrangère (région) - Not null	
Table 'région'						
code_reg	Code région	Integer	NC	Elémentaire	Clé primaire - Not null	
nom_reg	Nom de la région	Varchar	40	Elémentaire	Not null	

2. … et du schéma relationnel



3. Définition de la stratégie d'enregistrement des données

Rédaction et utilisation d'un programme Python permettant de :

- charger les données sélectionnées dans des tables 'vente', 'bien', 'commune', 'departement' et 'region'
- transformer et relier les données pour obtenir le format et le type défini dans le dictionnaire de données
- ajouter les index id_bien et id_vente
- extraire les tables sous format csv destinées à l'import SQL

➔ Réutilisable avec de nouvelles données de transactions immobilières

données_communes

Nom de colonne	Exemple de valeurs
CODREG	84
CODDEP	**01**
CODARR	02
CODCAN	08
CODCOM	**001**
COM	L'Abergement-Clémenciat
PMUN	779
PCAP	19
PTOT	**798**

référentiel_géographique

Nom de colonne	Exemple de valeurs
regrgp_nom	Province
reg_nom	**Auvergne-Rhône-Alpes**
reg_nom_old	Rhône-Alpes
aca_nom	Lyon
dep_nom	**Ain**
com_code	**01004**
com_code1	1004
com_code2	1004
com_id	C01004
com_nom_maj_court	AMBERIEU EN BUGEY
com_nom_maj	AMBERIEU-EN-BUGEY
com_nom	**Ambérieu-en-Bugey**
uu_code	1303
uu_id	UU01303
uucr_id	UU01303
uucr_nom	Ambérieu-en-Bugey
ze_id	ZE8201
dep_code	**1**
dep_id	D001
dep_nom_num	Ain (01)
dep_num_nom	01 - Ain
aca_code	10
aca_id	A10
reg_code	**84**
reg_id	R84
reg_code_old	82
reg_id_old	R82
fd_id	FD111
fr_id	FR11
fe_id	FE1
uu_id_99	UU01303
au_code	2
au_id	AU002
auc_id	AU002
auc_nom	Lyon
uu_id_10	UU01302
geolocalisation	45.9608475114,5.3729257777

valeurs_foncières

Nom de colonne (non vides)	Exemple de valeurs
No disposition	1
Date mutation	**2020/01/08**
Nature mutation	Vente
Valeur fonciere	**283000**
No voie	**123**
B/T/Q	**B**
Code type de voie	1
Type de voie	**AV**
Code voie	905
Voie	**SAINT MICHEL**
Code ID commune	1357
Code postal	45160
Commune	OLIVET
Code departement	**45**
Code commune	**232**
Section	BK
No plan	281
1er lot	123
Surface Carrez du 1er lot	**116,72**
Nombre de lots	1
Code type local	1
Type local	**Maison**
Surface reelle bati	**109**
Nombre pieces principales	**4**
Nature culture	AG
Nature culture speciale	IMM
Surface terrain	1112

Tables de la base de données

Vente
- id_vente
- Date mutation
- Valeur foncière
- Id_bien

Bien
- id_bien
- no_voie
- BTQ
- type_voie
- Voie
- code_dep_code_com
- type_local
- surface_carrez
- surface_bati
- nb_pieces

Commune
- code_dep_code_com
- nom_commune
- code_com
- code_dep
- population

Département
- dep_nom
- dep_code
- reg_code

Région
- reg_nom
- reg_code

4. Implémentation et chargement de la base de données (1/2)

Sous pgAdmin4 :

1. Création d'une base de donnée PostGreSQL locale « Base_immobiliere_Laplace »
2. Création des tables sous l'outil « Query » grâce au code généré par l'outil ERD de pgAdmin4 (extrait du code)
3. Importation sous l'outil « Query » dans la base de données des données csv préparées précédemment sous Python

1. Création des tables (extrait)

```
-- Création des tables
CREATE TABLE IF NOT EXISTS PUBLIC.bien (
    id_bien SERIAL NOT NULL,
    no_voie  INTEGER,
    "BTQ" "CHAR",
    type_voie character VARYING(4) COLLATE pg_catalog."default",
    voie character VARYING(100) COLLATE pg_catalog."default",
    code_dep_code_com character VARYING(6) COLLATE pg_catalog."def
ault" NOT NULL,
    type_local character VARYING(20) COLLATE pg_catalog."default",
    surface_carrez numeric NOT NULL,
    surface_bati   INTEGER,
    nb_pieces       INTEGER,
    PRIMARY KEY (id_bien)
     );

CREATE TABLE IF NOT EXISTS PUBLIC.vente (
    id_vente SERIAL,
    prix NUMERIC,
    date DATE,
    id_bien  INTEGER,
    PRIMARY KEY (id_vente)
     );

ALTER TABLE IF EXISTS PUBLIC.bien
    ADD CONSTRAINT "bien_Code_dep_code_com_fkey"
    FOREIGN KEY (code_dep_code_com) REFERENCES PUBLIC.commune (cod
e_dep_code_com) match simple
    ON UPDATE no action
    ON DELETE no action NOT valid;

ALTER TABLE IF EXISTS PUBLIC.vente
    ADD FOREIGN KEY (id_bien) REFERENCES PUBLIC.bien (id_bien) mat
ch simple
    ON UPDATE no action
    ON DELETE no action NOT valid;
```

Bien

Vente

Bien

Vente

2. Importation des données

```
-- Import des données
COPY region
FROM 'C:\Users\LENOVO\Desktop\Data\region.csv'
delimiter ';'
csv header;

COPY departement
FROM 'C:\Users\LENOVO\Desktop\Data\departement.csv'
delimiter ';'
csv header;

COPY commune
FROM 'C:\Users\LENOVO\Desktop\Data\commune.csv'
delimiter ';'
csv header;

COPY bien
FROM 'C:\Users\LENOVO\Desktop\Data\bien.csv'
delimiter ';' csv header;

COPY vente
FROM 'C:\Users\LENOVO\Desktop\Data\vente.csv'
delimiter ';'
csv header;
```

Région

Département

Commune

Bien

Vente

4. Implémentation et chargement de la base de données (2/2)

Aperçu des tables SQL chargées

Région

	code_reg [PK] integer	nom_reg character varying (40)
1	84	Auvergne-Rhône-Alpes
2	32	Hauts-de-France
3	93	Provence-Alpes-Côte d'Azur
4	44	Grand Est
5	76	Occitanie
6	28	Normandie
7	75	Nouvelle-Aquitaine
8	24	Centre-Val de Loire
9	27	Bourgogne-Franche-Comté
10	53	Bretagne

Total rows: 19 of 19 Query complete 00:00:0

Département

	code_dep [PK] character varying (3)	nom_dep character varying (100)	code_reg integer
1	1	Ain	84
2	2	Aisne	32
3	3	Allier	84
4	4	Alpes-de-Haute-Provence	93
5	5	Hautes-Alpes	93
6	6	Alpes-Maritimes	93
7	7	Ardèche	84
8	8	Ardennes	44
9	9	Ariège	76
10	10	Aube	44

Total rows: 109 of 109 Query complete 00:00:00.286

Commune

	code_dep_code_com [PK] character varying (6)	nom_commune character varying (100)	code_com integer	code_dep character varying (3)	population integer
1	1001	L'Abergement-Clémenciat	1	1	798
2	1002	L'Abergement-de-Varey	2	1	257
3	1004	Ambérieu-en-Bugey	4	1	14514
4	1005	Ambérieux-en-Dombes	5	1	1776
5	1006	Ambléon	6	1	118
6	1007	Ambronay	7	1	2915
7	1008	Ambutrix	8	1	777
8	1009	Andert-et-Condon	9	1	335
9	1010	Anglefort	10	1	1122
10	1011	Apremont	11	1	379

Total rows: 1000 of 34991 Query complete 00:00:00.381

Bien

	id_bien [PK] integer	no_voie integer	BTQ "char"	type_voie character varying (4)	voie character varying (100)	code_dep_code_com character varying (6)	type_local character varying (20)	surface_carrez numeric	surface_bati integer	nb_pieces integer
1	0	347	[null]	RUE	du chateau	1103	Appartement	48.22	48	3
2	1	4	[null]	BD	edouard baudoin	6004	Appartement	39.11	40	1
3	2	20	B	RUE	marceau	6088	Appartement	80.25	82	3
4	3	550	[null]	RTE	des vespins rn7	6123	Appartement	27.51	27	1
5	4	9300	[null]	RES	les arpeges bd des aba	13005	Appartement	47.33	47	2
6	5	27	[null]	RUE	du grand madier	13028	Appartement	25.31	24	1
7	6	360	[null]	AV	du prado	13208	Appartement	70.84	70	3
8	7	5076	F	PARC	dessuard	13212	Appartement	67.19	66	3
9	8	1194	[null]	RUE	de normandie	14338	Appartement	18.89	19	1
10	9	30	[null]	ALL	des noisetiers	14366	Maison	105.37	99	4

Total rows: 1000 of 34169 Query complete 00:00:00.804

Vente

	id_vente [PK] integer	prix numeric	date date	id_bien integer
1	0	165000.0	2020-01-02	0
2	1	355680.0	2020-01-02	1
3	2	229500.0	2020-01-02	2
4	3	125000.0	2020-01-02	3
5	4	90000.0	2020-01-02	4
6	5	93000.0	2020-01-02	5
7	6	298100.0	2020-01-02	6
8	7	163500.0	2020-01-02	7
9	8	53000.0	2020-01-02	8
10	9	136000.0	2020-01-02	9

Total rows: 1000 of 34169 Query complete 00:00:00.75

5. Vérification du bon fonctionnement

Requêtes SQL et affichage des résultats

5. Vérification du bon fonctionnement : requêtes SQL (1/7)

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

- Jointure sur les tables vente et bien
- Filtrer les ventes dont la date est dans le 1er semestre 2020
- Filtrer les biens de type appartement
- Compter le nombre d'entrées correspondant dans la table

```
SELECT Count(id_vente) AS nombre_ventes_S1_2020
FROM  vente v
NATURAL JOIN bien b
WHERE  v.date BETWEEN '2020-01-01' AND '2020-06-30'
AND b.type_local = 'Appartement';
```

	nombre_ventes_s1_2020 bigint
1	31378

2. Nombre de ventes d'appartement par région pour le 1er semestre 2020

- Jointure sur les tables vente, bien, commune, departement, region
- Filtrer les ventes dont la date est dans le 1er semestre 2020
- Filtrer les biens de type appartement
- Grouper par region
- Compter le nombre d'entrées par region

```
SELECT   r.code_reg,
         r.nom_reg,
         Count(*) AS nb_ventes
FROM     vente v
NATURAL JOIN     bien b
NATURAL JOIN     commune c natural
NATURAL JOIN     departement d natural
NATURAL JOIN     region r
WHERE    b.type_local = 'Appartement'
AND      v.date BETWEEN '2020-01-01' AND '2020-06-30'
GROUP BY r.nom_reg,
         r.code_reg;
```

	code_reg [PK] integer	nom_reg character varying (40)	nb_ventes bigint
1	4	La Réunion	44
2	2	Martinique	94
3	75	Nouvelle-Aquitaine	1932
4	27	Bourgogne-Franche-Comté	376
5	52	Pays de la Loire	1357
6	84	Auvergne-Rhône-Alpes	3253
7	11	Ile-de-France	13995
8	44	Grand Est	984
9	3	Guyane	34
10	32	Hauts-de-France	1254
11	53	Bretagne	983
12	93	Provence-Alpes-Côte d'Azur	3649
13	1	Guadeloupe	2
14	76	Occitanie	1640
15	28	Normandie	862
16	24	Centre-Val de Loire	696
17	94	Corse	223

A comparer au nombre d'appartements total de chaque région pour obtenir une mesure de l'activité immobilière de chaque région

5. Vérification du bon fonctionnement : requêtes SQL (2/7)

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

- CTE retournant le nombre total de ventes d'appartement
- Jointure des tables vente et bien
- Filtrer les biens de type appartement
- Grouper les ventes par nombre de pièces des biens
- Compter le nombre de ventes par nombre de pièces
- Retourner la proportion en divisant ce nombre par le nombre total d'appartements avec mise sous forme de pourcentage

```
WITH constantes AS
(
       SELECT Count(*) AS total_ventes
       FROM vente v
       NATURAL JOIN bien b
       WHERE  b.type_local = 'Appartement' )
SELECT b.nb_pieces,
      To_char(100*Cast(Count(v.id_vente) AS DECIMAL) / cs.total_ventes, '00D00%')
         AS prop_appartements,
      Count(v.id_vente) AS nb_ventes
FROM       bien b
NATURAL JOIN vente v
CROSS JOIN constantes cs
WHERE      b.type_local = 'Appartement'
GROUP BY   b.nb_pieces,
           cs.total_ventes
ORDER BY   b.nb_pieces;
```

30 appartements sans nombre de pièces renseigné

	nb_pieces integer	prop_appartements text	nb_ventes bigint
1	0	00,10%	30
2	1	21,48%	6739
3	2	31,18%	9783
4	3	28,57%	8966
5	4	14,21%	4460
6	5	03,55%	1114

	nb_pieces integer	prop_appartements text	nb_ventes bigint
7	6	00,65%	204
8	7	00,17%	54
9	8	00,05%	17
10	9	00,03%	8
11	10	00,01%	2
12	11	00,00%	1

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

- Jointure des tables vente, bien, commune et departement
- Filtrer les ventes de prix non nul
- Grouper par département
- Calculer la moyenne du prix au m² des ventes
- Trier par prix moyen décroissant
- Sélectionner les 10 premières lignes

```
SELECT   d.nom_dep,
         d.code_dep,
         Round(Avg(v.prix/b.surface_carrez)) AS prix_m2
FROM     departement d
NATURAL JOIN commune c
NATURAL JOIN bien b
NATURAL JOIN vente v
WHERE    v.prix notnull
GROUP BY d.nom_dep,
         d.code_dep
ORDER BY prix_m2 DESC limit 10;
```

	nom_dep character varying (100)	code_dep [PK] character varying (3)	prix_m2 numeric	nb_ventes bigint
1	Paris	75	12053	4840
2	Hauts-de-Seine	92	7219	1813
3	Val-de-Marne	94	5343	1926
4	Alpes-Maritimes	6	4700	1343
5	Haute-Savoie	74	4667	364
6	Seine-Saint-Denis	93	4345	1074
7	Yvelines	78	4225	1586
8	Rhône	69	4059	1412
9	Corse-du-Sud	2A	4027	245
10	Gironde	33	3764	1239

5. Vérification du bon fonctionnement : requêtes SQL (3/7)

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

- Jointure des tables vente, bien, commune, departement et region
- Filtrer les biens de type maison de prix non nul
- Filtrer les biens situés dans la région Ile-de-France
- Calculer la moyenne des prix moyen au m² des ventes

```
SELECT    r.nom_reg,
          Round(Avg(v.prix/b.surface_carrez)) AS prix_m2
FROM      vente v
NATURAL JOIN bien b
NATURAL JOIN commune c
NATURAL JOIN departement d
NATURAL JOIN region r
WHERE     b.type_local = 'Maison'
AND       v.prix notnull
AND       r.nom_reg = 'Ile-de-France'
GROUP BY r.nom_reg;
```

	nom_reg character varying (40)	prix_m2 numeric
1	Ile-de-France	3745

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

- Jointure des tables vente, bien, commune, departement et region
- Filtrer les biens de type appartement de prix non nul
- Sélectionner la région et la surface de chaque bien
- Trier par prix décroissant
- Sélectionner les 10 premières lignes

```
SELECT    v.id_vente,
          To_char(v.prix/b.surface_carrez, '9G999G999') AS prix_m2,
          To_char(v.prix, '9G999G999')                  AS prix,
          b.surface_carrez,
          b.surface_bati,
          b.nb_pieces,
          c.nom_commune,
          r.nom_reg
FROM      vente v
NATURAL JOIN bien b
NATURAL JOIN commune c
NATURAL JOIN departement d
NATURAL JOIN region r
WHERE     b.type_local = 'Appartement'
AND       v.prix notnull
ORDER BY prix DESC limit 10;
```

	id_vente integer	prix_m2 text	prix text	surface_carrez numeric	surface_bati integer	nb_pieces integer	nom_commune character varying (100)	nom_reg character varying (40)
1	30602	989 011	9 000 000	9.1	10	1	Paris 16e	Ile-de-France
2	5260	134 375	8 600 000	64.0	62	3	Corbeil-Essonnes	Ile-de-France
3	3624	417 407	8 577 713	20.55	289	7	Paris 7e	Ile-de-France
4	7601	178 162	7 620 000	42.77	42	2	Paris 17e	Ile-de-France
5	9987	30 004	7 600 000	253.3	200	5	Paris 6e	Ile-de-France
6	17822	53 860	7 535 000	139.9	143	4	Paris 1er	Ile-de-France
7	409	20 557	7 420 000	360.95	357	8	Paris 16e	Ile-de-France
8	16356	12 101	7 200 000	595.0	241	8	Paris 16e	Ile-de-France
9	1923	57 523	7 050 000	122.56	310	7	Paris 1er	Ile-de-France
10	19160	83 144	6 600 000	79.38	76	3	Paris 1er	Ile-de-France

Données aberrantes
Prix, surface carrez, surface bâti et nombre de pièces incohérents : Le prix correspond peut-être parfois à l'ensemble de la vente et non à celle du 1er lot

Nettoyage de données à réaliser

5. Vérification du bon fonctionnement : requêtes SQL (4/7)

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

- CTE retournant dans la table nb_ventes_trimestre le nombre de ventes pour le T1 2020 et pour le T2 2020 :
  - Filtrer les ventes dont la date appartient au trimestre voulu
  - Compter le nombre d'entrées dans la table vente
- Calculer le taux d'évolution à partir de ces deux données
- Mettre ce taux sous forme de pourcentage

```
WITH nb_ventes_trimestre AS (
    SELECT
        Cast(
            (SELECT Count(*)
            FROM   vente
            WHERE  date BETWEEN '2020-01-01' AND '2020-03-31')
            AS DECIMAL)
        AS ventes_t1_2020,
        Cast(
            (SELECT Count(*)
            FROM   vente
            WHERE  date BETWEEN '2020-04-01' AND '2020-06-30')
            AS DECIMAL)
        AS ventes_t2_2020)
SELECT ventes_t1_2020,
       ventes_t2_2020,
       To_char(100 * ( ventes_t2_2020  ventes_t1_2020 ) /
           ventes_t1_2020, '0D0%') AS taux_evolution
FROM nb_ventes_trimestre;
```

	ventes_t1_2020 numeric	ventes_t2_2020 numeric	taux_evolution text
1	16776	17393	3,7%

Evolution positive malgré le confinement du 2ème trimestre 2020

8. Classement des régions par rapport au prix au mètre carré des appartement de plus de 4 pièces

- Jointure sur les tables region, departement, commune, bien, vente
- Filtrer les bien de type appartement de plus de 4 pièces de prix non nul
- Grouper par région
- Calculer le prix au m² de chaque région
- Calculer le rang de chaque région selon son prix au m²
- Afficher les données pour chaque région
- Trier par rang décroissant de la région

```
SELECT    Rank() OVER (ORDER BY Avg(v.prix / b.surface_carrez) DESC)
          AS rang_region,
          r.nom_reg,
          Round(Avg(v.prix/b.surface_carrez)) AS prix_m2_region_min_t4 ,
          Count(*) AS nb_ventes
FROM      region r
NATURAL JOIN departement d
NATURAL JOIN commune c
NATURAL JOIN bien b
NATURAL JOIN vente v
WHERE     b.type_local = 'Appartement'
AND       b.nb_pieces >= 4
AND       v.prix notnull
GROUP BY r.nom_reg
ORDER BY rang_region;
```

Certains échantillons sont trop faibles pour être représentatifs : France non métropolitaine & Corse dans une moindre mesure

Ratio probablement différent entre appartements et maisons

	rang_region bigint	nom_reg character varying (40)	prix_m2_region_min_t4 numeric	nb_ventes bigint
1	1	Ile-de-France	6702	2648
2	2	Corse	3256	38
3	3	Provence-Alpes-Côte d'Azur	3423	573
4	4	La Réunion	3149	4
5	5	Auvergne-Rhône-Alpes	2862	788
6	6	Nouvelle-Aquitaine	2591	314
7	7	Pays de la Loire	2421	225
8	8	Bretagne	2247	163
9	9	Hauts-de-France	2184	238
10	10	Normandie	2106	134
11	11	Martinique	2080	14
12	12	Occitanie	2032	236
13	13	Guyane	1789	2
14	14	Centre-Val de Loire	3342	133
15	15	Grand Est	1514	234
16	16	Bourgogne-Franche-Comté	1236	113

5. Vérification du bon fonctionnement : requêtes SQL (5/7)

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

- Jointure sur les tables vente, bien et commune
- Filtrer les ventes dont la date appartient au 1er trimestre
- Grouper par commune
- Compter le nombre de ventes par commune
- Filtrer les communes où le nombre de ventes est >= 50
- Afficher le nom de chaque commune et le nombre de ventes pour chacune

```
SELECT   c.nom_commune,
         Count(v.id_vente) AS nombre_ventes
FROM     vente v
NATURAL JOIN bien b
NATURAL JOIN commune c
WHERE    v.date BETWEEN '2020-01-01' AND '2020-03-31'
GROUP BY c.nom_commune
HAVING   Count(v.id_vente) >= 50
ORDER BY nombre_ventes DESC;
```

	nom_commune character varying (100)	nombre_ventes bigint
1	Paris 17e	228
2	Paris 15e	215
3	Paris 18e	209
4	Nice	173
5	Paris 11e	169
6	Paris 16e	165
7	Bordeaux	157
8	Paris 14e	146
9	Paris 20e	127
10	Nantes	119
11	Paris 19e	116
12	Paris 12e	110
13	Paris 10e	109
14	Grenoble	106
15	Paris 9e	106

	nom_commune character varying (100)	nombre_ventes bigint
16	Boulogne-Billancourt	99
17	Paris 13e	94
18	Paris 7e	87
19	Paris 6e	86
20	Marseille 8e	81
21	Asnières-sur-Seine	81
22	Courbevoie	80
23	Paris 5e	79
24	Paris 3e	79
25	Toulouse	78
26	Antibes	77
27	Marseille 4e	72
28	Marseille 1er	71
29	Vincennes	68
30	Rueil-Malmaison	68

	nom_commune character varying (100)	nombre_ventes bigint
36	Sète	62
37	La Ciotat	62
38	Paris 8e	62
39	Paris 2e	61
40	Rennes	61
41	Paris 4e	60
42	Levallois-Perret	59
43	Toulon	59
44	Saint-Maur-des-Fossés	56
45	Ajaccio	54
46	Versailles	54
47	Puteaux	53
48	Issy-les-Moulineaux	50

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

- 1ère CTE 'prix_m2_pieces' retournant les prix au m² des appartements en fonction du nombre de pieces :
  - Jointure des tables vente et bien
  - Filtrer les biens de type appartement de prix non nul
  - Filtrer les biens dont le nombre de pièces est 2 ou 3
  - Retourner le prix au m² comme la moyenne des prix au m² des ventes
- 2ème CTE 'constantes' retournant prix_m2_t2 et prix_m2_t3 depuis la CTE 'prix_m2_pieces'
- Afficher les prix au m² pour les T2 et les T3 et le pourcentage de différence calculé

```
WITH constantes AS (
      WITH prix_m2_pieces AS (
            SELECT nb_pieces,
            Avg(v.prix/b.surface_carrez) AS prix_moyen_m2
            FROM vente v
            NATURAL JOIN bien b
            WHERE    b.nb_pieces IN (2,3)
            AND b.type_local = 'Appartement'
            AND v.prix notnull
            GROUP BY nb_pieces)
      SELECT
            (SELECT prix_moyen_m2 FROM prix_m2_pieces WHERE nb_pieces = 2) AS prix_m2_f2,
            (SELECT prix_moyen_m2 FROM prix_m2_pieces WHERE nb_pieces = 3) AS prix_m2_f3
      )
SELECT
      round(prix_m2_f2) AS prix_m2_f2,
      round(prix_m2_f3) AS prix_m2_f3,
      to_char(100 * ( prix_m2_f2 - prix_m2_f3 ) / prix_m2_t3, '00D0%') AS diff_prix_f2_f3
FROM constantes;
```

	prix_m2_t2 numeric	prix_m2_t3 numeric	diff_prix_t2_t3 text
1	4909	4300	14,2%

5. Vérification du bon fonctionnement : requêtes SQL (6/7)

11. Moyennes des valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69

- CTE retournant les prix moyens des communes demandées et leur rang dans le département :
  - Jointure des tables vente, bien, commune et departement
  - Filtrer les ventes de prix non nul
  - Filtrer les communes de code 6, 13, 33, 59, 69
  - Grouper par commune
  - Retourner le prix moyen au m² comme le rapport entre la somme des prix et la somme des surfaces
  - Retourner le rang de chaque commune dans son département à partir de ce prix moyen au m²
- Filtrer les communes dont le rang départemental est <= 3
- Retourner le nom de la commune, le département, et le prix moyen

```
WITH prix_moyen_commune AS
    (
    SELECT d.code_dep,
        c.nom_commune,
        Count(*) AS nb_ventes,
        To_char(Avg(v.prix), '999G999G999') AS prix_moyen,
        Rank() OVER(partition BY d.code_dep ORDER BY Avg(v.prix) DESC)
            AS rang_dep
    FROM vente v
    NATURAL JOIN bien b
    NATURAL JOIN commune c
    NATURAL JOIN departement d
    WHERE    d.code_dep
        IN ('6', '13', '33', '59', '69')
    AND      v.prix notnull
    GROUP BY c.nom_commune, d.code_dep)
SELECT   nom_commune,
         code_dep,
         rang_dep,
         prix_moyen
FROM     prix_moyen_commune
WHERE    rang_dep <= 3
ORDER BY prix_moyen DESC;
```

	nom_commune character varying (100)	code_dep character varying (3)	rang_dep bigint	prix_moyen text	nb_ventes bigint
1	Saint-Jean-Cap-Ferrat	6	1	968 750	4
2	Bersée	59	1	433 202	1
3	Gignac-la-Nerthe	13	1	330 000	1
4	Lège-Cap-Ferret	33	1	549 501	22
5	Ville-sur-Jarnioux	69	1	485 300	1
6	Cysoing	59	2	408 550	1
7	Saint-Savournin	13	2	314 425	2
8	Vayres	33	2	335 000	1
9	Eze	6	2	655 000	2
10	Lyon 2e	69	2	455 217	69
11	Lyon 6e	69	3	426 968	32
12	Mouans-Sartoux	6	3	476 898	8
13	Cassis	13	3	313 417	16
14	Halluin	59	3	322 250	4
15	Arcachon	33	3	307 436	55

Echantillonnage trop serré :
Plusieurs moyennes basée sur 1 ou 2 ventes

➔ **Travailler sur des communautés de commune, ou sur une période plus longue ?**

5. Vérification du bon fonctionnement : requêtes SQL (7/7)

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

- Jointure sur les tables commune, bien, vente
- Filtrer les communes de population >= 10 000
- Grouper par commune
- Calculer pour chaque commune le nombre de vente pour 1000 habitants
- Trier les communes par nombre de ventes pour 1000 habitants
- Sélectionner les 20 premières valeurs
- Afficher chaque commune et le nombre de ventes mis en forme en ‰

```
SELECT c.code_dep_code_com,
    c.nom_commune,
    To_char((1000*Cast(Count(*) AS DECIMAL) /c.population), '0D0 ‰' )
        AS nb_ventes_pour_1000_hab,
     c.population
FROM commune c
NATURAL JOIN bien b
NATURAL JOIN vente v
WHERE    c.population >= 10000
GROUP BY c.nom_commune,
         c.population,
         c.code_dep_code_com
ORDER BY nb_ventes_pour_1000_hab DESC limit 20;
```

	code_dep_code_com [PK] character varying (6)	nom_commune character varying (100)	nb_ventes_pour_1000_hab text	population integer
1	75102	Paris 2e	5,8 ‰	21735
2	75101	Paris 1er	4,9 ‰	16055
3	75103	Paris 3e	4,7 ‰	34306
4	44055	La Baule-Escoublac	4,6 ‰	16797
5	33009	Arcachon	4,6 ‰	11898
6	75104	Paris 4e	4,1 ‰	29390
7	6104	Roquebrune-Cap-Martin	4,0 ‰	13041
8	75108	Paris 8e	3,8 ‰	36250
9	83123	Sanary-sur-Mer	3,5 ‰	17160
10	75109	Paris 9e	3,4 ‰	60563
11	75106	Paris 6e	3,4 ‰	41171
12	83071	La Londe-les-Maures	3,4 ‰	10776
13	83112	Saint-Cyr-sur-Mer	3,2 ‰	11725
14	60141	Chantilly	3,1 ‰	11178
15	44132	Pornichet	3,1 ‰	11440
16	94067	Saint-Mandé	3,1 ‰	22576
17	75110	Paris 10e	3,0 ‰	86863
18	85226	Saint-Hilaire-de-Riez	2,9 ‰	11501
19	6083	Menton	2,9 ‰	30981
20	75117	Paris 17e	2,8 ‰	167975