



Création et utilisation de la base de données

POIRIER PIERRE-MARIE



Laplace Immo

Contexte du Projet



Contexte

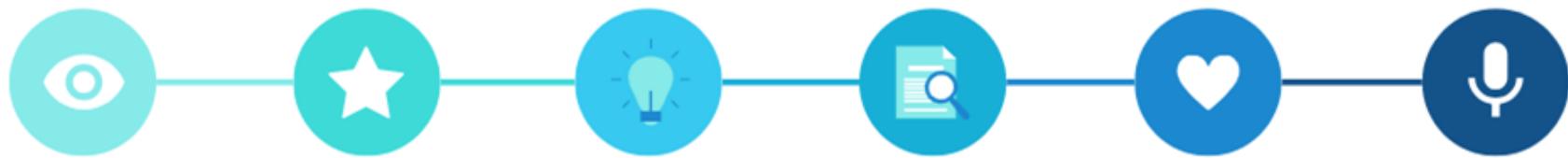
LaPlace Immo réseau d'agence immobilière

Objectif : Exploiter les données pour prendre un avantage concurrentiel:

Projet DATAIMMO : Création d'un modèle prédictif des prix de vente des biens immobiliers

Etude des transactions immobilières et foncière en France pour aider les différentes agences régionales à mieux accompagner leurs clients.

Timeline



Exploration des données

Modification du schéma relationnel

Création de la BDD

Import et contrôle de l'intégrité de la BDD

Travail de requêtage pour répondre aux besoins

Support de présentation

Les données initiales

3 jeux de données :

- Données communes issu de l'INSEE
- Référentiel géographique provenant de Data.gouv
- Demande des Valeurs Foncieres issu de Open Data

La stratégie de sauvegarde et la conformité RGPD

Il s'agit de reconstruire les données avec les informations strictement nécessaire

- Valeur foncière : on supprime les infos personnels (nom de l'acquéreur) ainsi que toutes les colonnes ne présentant pas d'intérêt. Puis on l'a découpe en 2 tables : bien et vente en créant 2 ids
- Commune : on joint valeur foncière à commune on garde seulement la population total l'id_codedep_codecommune le code_dep et le nom de la commune.
- Référentielle géographique on récupère la liste des régions et leurs ids

On vérifie que les tables ne contiennent plus d'informations personnelles pour respecter la RGPD

L'extrait du dictionnaire des données

DICTIONNAIRE DES DONNÉES - Vente

| CODE | SIGNIFICATION | TYPE | LONGUEUR | Clé | NATURE | RÈGLE DE GESTION | RÈGLE DE CALCUL |
|----------|---------------------------|---------|----------|----------------|-------------|----------------------|-----------------|
| id_vente | numero unique de la vente | INTEGER | | Clé primaire | Elémentaire | Ne doit pas être nul | |
| id_bien | numero unique du bien | INTEGER | | Clé secondaire | Elémentaire | Ne doit pas être nul | |
| date | date de la vente | date | | | Elémentaire | Ne doit pas être nul | |
| valeur | valeur du bien | Integer | | | Elémentaire | | |

DICTIONNAIRE DES DONNÉES - Bien

| CODE | SIGNIFICATION | TYPE | LONGUEUR | Clé | NATURE | RÈGLE DE GESTION | RÈGLE DE CALCUL |
|------------------------|--|---------|----------|----------------|-------------|----------------------|-----------------------|
| id_bien | numero unique du bien | INTEGER | | Clé primaire | Elémentaire | Ne doit pas être nul | |
| id_codedep_codecommune | Concaténation du code département et code commune pour a | Varchar | | Clé secondaire | Concaténer | Ne doit pas être nul | Colonne U + Colonne V |
| no_voie | Numéro des rues | Integer | | | Elémentaire | | |
| type_de_voie | Plusieurs valeurs (rue, avenue, chemin, etc.) | Varchar | 50 | | Elémentaire | | |
| voie | Nom de la rue | Varchar | 50 | | Elémentaire | | |
| nb_piece | Nombre de pièce du bien | Integer | | | Elémentaire | Ne doit pas être nul | |
| surface_carrez | Taille en m ² du bien | INTEGER | | | Elémentaire | Ne doit pas être nul | |
| type_local | Type du bien : Maison ou appartement | Varchar | 20 | | Elémentaire | Ne doit pas être nul | |

L'extrait du dictionnaire des données

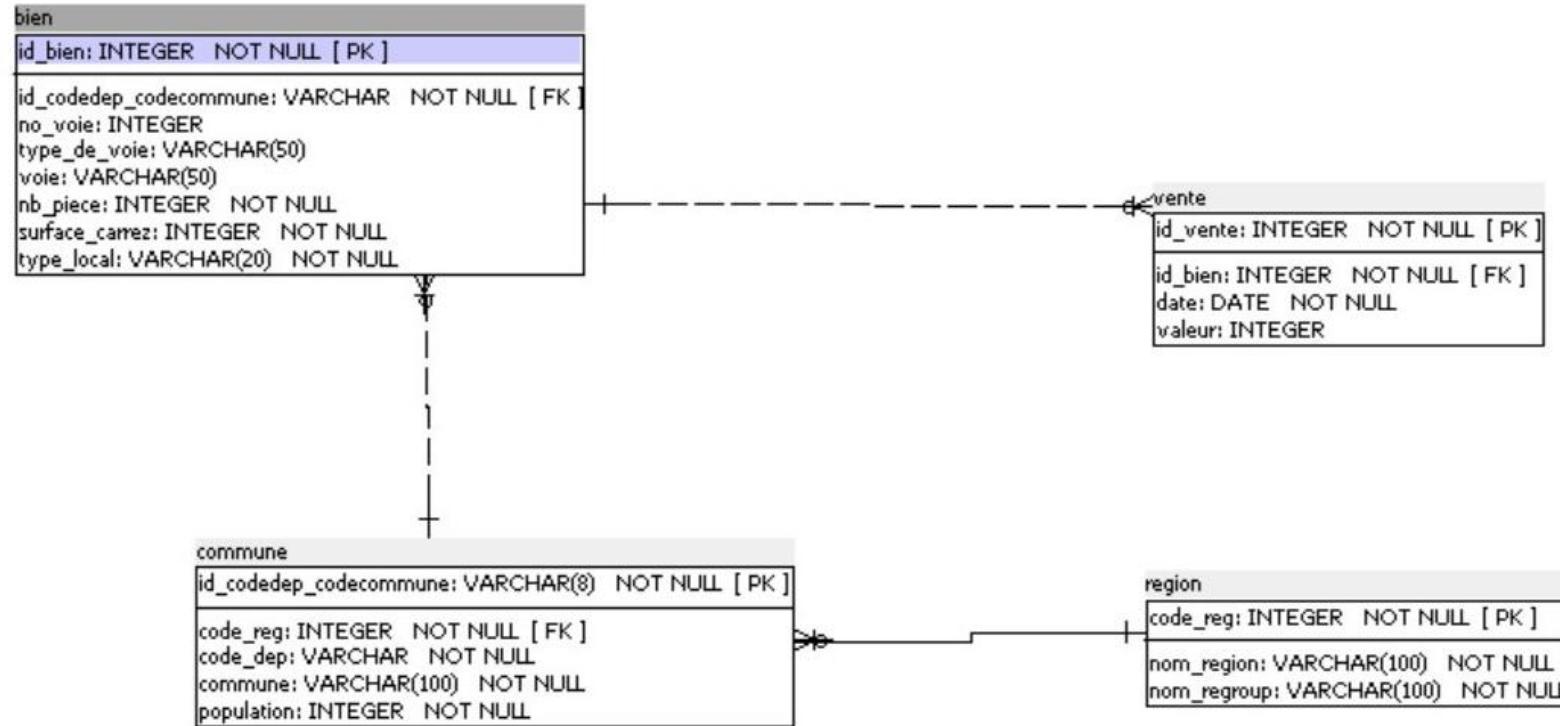
Dictionnaire des données - Commune

| CODE | SIGNIFICATION | TYPE | LONGUEUR | Clé | NATURE | RÈGLE DE GESTION | RÈGLE DE CALCUL |
|-------------------|---|---------|----------|----------------|-------------|----------------------|-----------------|
| id_codedep_codecc | Concaténation du code département et code commune pour avoir une clé unique | Varchar | 8 | Clé primaire | Elémentaire | Ne doit pas être nul | |
| code_reg | code de la région | INTEGER | | Clé secondaire | Elémentaire | Ne doit pas être nul | |
| code_dep | code du département | Varchar | | | Elémentaire | Ne doit pas être nul | |
| commune | nom de la commune | Varchar | 100 | | Elémentaire | Ne doit pas être nul | |
| population | Population de la commune | Integer | | | Elémentaire | Ne doit pas être nul | |

Dictionnaire des données - Région

| CODE | SIGNIFICATION | TYPE | LONGUEUR | Clé | NATURE | RÈGLE DE GESTION | RÈGLE DE CALCUL |
|-------------|------------------------|---------|----------|--------------|-------------|----------------------|-----------------|
| code_reg | code région | INTEGER | | Clé Primaire | Elémentaire | Ne doit pas être nul | |
| nom_region | nom de la région | Varchar | 100 | | Elémentaire | Ne doit pas être nul | |
| nom_regroup | typologie de la région | Varchar | 100 | | Elémentaire | Ne doit pas être nul | |

Le schéma relationnel normalisé



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

VENTE

| 1 select * from vente | | | | |
|------------------------------------|-----------------------|-----------------|------------|----------------|
| Data Output Messages Notifications | | | | |
| | id_vente [PK] integer | id_bien integer | date date | valeur integer |
| 1 | 1 | 1 | 2020-01-02 | 165000 |
| 2 | 2 | 2 | 2020-01-02 | 355680 |
| 3 | 3 | 3 | 2020-01-02 | 229500 |

BIEN

| 1 select * from bien | | | | | | | | |
|------------------------------------|----------------------|--|-----------------|-------------------------------------|-----------------------------|------------------|------------------------|-----------------------------------|
| Data Output Messages Notifications | | | | | | | | |
| | id_bien [PK] integer | id_codedep_codecommune character varying | no_voie integer | type_de_voie character varying (50) | voie character varying (50) | nb_piece integer | surface_carrez integer | type_local character varying (20) |
| 1 | 1 | 1103 | 347 | RUE | DU CHATEAU | 3 | 48 | Appartement |
| 2 | 2 | 6004 | 4 | BD | EDOUARD BAUDOIN | 1 | 39 | Appartement |
| 3 | 3 | 6088 | 20 | RUE | MARCEAU | 3 | 80 | Appartement |

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

COMMUNE

The screenshot shows a SQL interface with a query window containing the command `select * from commune`. Below the query window is a toolbar with various icons for file operations and data export. The main area displays the results of the query in a tabular format. The table has five columns: `id_codedep_codecommune`, `code_reg`, `code_dep`, `commune`, and `population`. The data rows are:

| | <code>id_codedep_codecommune</code> [PK] character varying (8) | <code>code_reg</code> integer | <code>code_dep</code> character varying | <code>commune</code> character varying (100) | <code>population</code> integer |
|---|---|----------------------------------|--|---|------------------------------------|
| 1 | 1103 | 84 | 01 | CHEVRY | 2196 |
| 2 | 1004 | 84 | 01 | AMBERIEU-EN-BUGEY | 14514 |
| 3 | 1014 | 84 | 01 | ARBENT | 3442 |

REGION

The screenshot shows a SQL interface with a query window containing the command `select * from region`. Below the query window is a toolbar with various icons for file operations and data export. The main area displays the results of the query in a tabular format. The table has three columns: `code_reg`, `nom_region`, and `nom_regroup`. The data rows are:

| | <code>code_reg</code> [PK] integer | <code>nom_region</code> character varying (100) | <code>nom_regroup</code> character varying (100) |
|---|---------------------------------------|--|---|
| 1 | 0 | Collectivités d'outre-mer | DROM-COM |
| 2 | 1 | Guadeloupe | DROM-COM |
| 3 | 2 | Martinique | DROM-COM |

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

Requête Historique

```
1 ▾ SELECT
2     (SELECT COUNT(*) FROM BIEN) AS nb_Bien,
3     (SELECT COUNT(*) FROM VENTE) AS nb_Vente,
4     (SELECT COUNT(*) FROM COMMUNE) AS nb_Commune,
5     (SELECT COUNT(*) FROM REGION) AS nb_Region
```

Data Output Messages Notifications

≡+ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ ⌂ SQL

| | nb_bien bigint | nb_vente bigint | nb_commune bigint | nb_region bigint |
|---|-------------------|--------------------|----------------------|---------------------|
| 1 | 34169 | 34169 | 3125 | 19 |



Requêtes SQL et résultats

R1 : Nombre total d'appartements vendus au 1er semestre 2020.

Query Query History

```
1 < SELECT
2       COUNT(*) AS NB_APPARTEMENTS_VENDUS
3   FROM
4     PUBLIC.VENTE AS V
5   LEFT JOIN BIEN AS B ON V.ID_BIEN = B.ID_BIEN
6 WHERE
7   TYPE_LOCAL = 'Appartement'
8   AND V.DATE BETWEEN '2020-01-01' AND '2020-06-30'
```

Data Output Messages Notifications

nb_appartements_vendus
bigint

| | nb_appartements_vendus | bigint |
|---|------------------------|--------|
| 1 | | 31378 |

R2 : Le nombre de ventes d'appartement par région pour le 1er semestre 2020

```
1 ✓ SELECT commune.code_reg, region.nom_region, count(DISTINCT bien.id_bien) AS nb_apparte_vendu
2   FROM bien
3   LEFT JOIN vente ON bien.id_bien = vente.Id_bien
4   LEFT JOIN commune ON bien.id_codedep_codecommune = commune.id_codedep_codecommune
5   LEFT JOIN region ON commune.code_reg = region.code_reg
6   WHERE Type_local = 'Appartement' AND date BETWEEN '2020/01/01' AND '2020/06/30'
7   GROUP BY commune.code_reg, region.nom_region
8   ORDER BY 3 DESC
```

Data Output Messages Notifications

SQL

| | code_reg integer | nom_region character varying (100) | nb_apparte_vendu bigint |
|----|---------------------|---------------------------------------|----------------------------|
| 1 | 11 | Ile-de-France | 13995 |
| 2 | 93 | Provence-Alpes-Côte d'Azur | 3649 |
| 3 | 84 | Auvergne-Rhône-Alpes | 3253 |
| 4 | 75 | Nouvelle-Aquitaine | 1932 |
| 5 | 76 | Occitanie | 1640 |
| 6 | 52 | Pays de la Loire | 1357 |
| 7 | 32 | Hauts-de-France | 1254 |
| 8 | 44 | Grand Est | 984 |
| 9 | 53 | Bretagne | 983 |
| 10 | 28 | Normandie | 862 |
| 11 | 24 | Centre-Val de Loire | 696 |
| 12 | 27 | Bourgogne-Franche-Comté | 376 |
| 13 | 94 | Corse | 223 |
| 14 | 2 | Martinique | 94 |
| 15 | 4 | La Réunion | 44 |
| 16 | 3 | Guyane | 34 |
| 17 | 1 | Guadeloupe | 2 |

R3 : Proportion des ventes d'appartements par le nombre de pièces

```
1 ✓ SELECT
2     b.nb_piece,
3     round(COUNT(v.id_vente) / SUM(COUNT(v.id_vente)) OVER ()*100,2) AS proportion_vente
4 FROM vente AS v
5 LEFT JOIN bien AS b ON v.id_bien = b.id_bien
6 WHERE b.type_local = 'Appartement'
7 GROUP BY 1
8 ORDER BY 1;
```

Data Output Messages Notifications

| | nb_piece integer | proportion_vente numeric |
|----|---------------------|-----------------------------|
| 1 | 0 | 0.10 |
| 2 | 1 | 21.48 |
| 3 | 2 | 31.18 |
| 4 | 3 | 28.57 |
| 5 | 4 | 14.21 |
| 6 | 5 | 3.55 |
| 7 | 6 | 0.65 |
| 8 | 7 | 0.17 |
| 9 | 8 | 0.05 |
| 10 | 9 | 0.03 |
| 11 | 10 | 0.01 |
| 12 | 11 | 0.00 |

R4 : Liste des 10 départements où le prix du mètre carré est le plus élevé

```
1 v SELECT
2      c.code_dep as departement,
3      ROUND(AVG(v.valeur/b.surface_carrez),2) as prix_m2_moyen
4  FROM
5      vente AS v
6      LEFT JOIN bien AS b ON v.id_bien = b.id_bien
7      LEFT JOIN commune AS c ON b.id_codedep_codecommune = c.id_codedep_codecommune
8  WHERE b.surface_carrez !=0 and c.code_dep IS not null and v.valeur is not null
9  group by 1
10 order by 2 desc
11 LIMIT 10
```

Data Output Messages Notifications

| | departement character varying | prix_m2_moyen numeric |
|----|----------------------------------|--------------------------|
| 1 | 75 | 11987.53 |
| 2 | 92 | 7216.01 |
| 3 | 94 | 5369.69 |
| 4 | 06 | 4705.40 |
| 5 | 74 | 4696.79 |
| 6 | 93 | 4345.36 |
| 7 | 78 | 4236.66 |
| 8 | 69 | 4057.60 |
| 9 | 2A | 4027.12 |
| 10 | 33 | 3764.21 |

R5 : Prix moyen du mètre carré d'une maison en Île-de-France

```
1 v select
2     r.nom_region as nom_region,
3     b.type_local,
4     round(avg(v.valeur / b.surface_carrez), 2) as prix_m2_moyen
5 from
6     bien as b
7     left join vente as v on b.id_bien = v.id_bien
8     left join commune as c on b.id_codedep_codecommune = c.id_codedep_codecommune
9     left join region as r on c.code_reg = r.code_reg
10 where b.type_local = 'Maison'
11     and r.nom_region = 'Île-de-France'
12     and v.valeur is not null
13 group by
14     1,
15     2
16 order by
17     prix_m2_moyen desc;
```

Data Output Messages Notifications

SQL

| | nom_region character varying (100) | type_local character varying (20) | prix_m2_moyen numeric |
|---|---------------------------------------|--------------------------------------|--------------------------|
| 1 | Île-de-France | Maison | 3744.30 |

R6 : Liste des 10 appartements les plus chers avec la région et le nombre de mètres carrés

```
select
    b.id_bien,
    b.surface_carrez,
    b.type_local,
    v.valeur,
    r.nom_region as nom_region
from
    bien as b
    left join vente as v on b.id_bien = v.id_bien
    left join commune as c on b.id_codedep_codecommune = c.id_codedep_
    left join region as r on c.code_reg = r.code_reg
where
    b.type_local = 'Appartement'
    and v.valeur is not null
order by
    v.valeur desc
limit 10;
```

Output Messages Notifications

| id_bien | surface_carrez | type_local | valeur | nom_region |
|---------|----------------|-------------|---------|---------------|
| 30603 | 9 | Appartement | 9000000 | Île-de-France |
| 5261 | 64 | Appartement | 8600000 | Île-de-France |
| 3625 | 21 | Appartement | 8577713 | Île-de-France |
| 7602 | 43 | Appartement | 7620000 | Île-de-France |
| 9988 | 253 | Appartement | 7600000 | Île-de-France |
| 17823 | 140 | Appartement | 7535000 | Île-de-France |
| 410 | 361 | Appartement | 7420000 | Île-de-France |
| 16357 | 595 | Appartement | 7200000 | Île-de-France |
| 1924 | 123 | Appartement | 7050000 | Île-de-France |
| 19161 | 79 | Appartement | 6600000 | Île-de-France |

R7 : Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020

```
1 v with trim1 as (
2     select round(count(*), 2) as nb_trim1
3     from vente
4     where date between '2020/01/01' and '2020/03/31'
5 ),
6 trim2 as (
7     select round(count(*), 2) as nb_trim2
8     from vente
9     where date between '2020/04/01' and '2020/06/30'
10 )
11 select round(((nb_trim2 - nb_trim1) / nb_trim1 * 100), 2) as taux_evolution
12 from trim1, trim2;
```

The screenshot shows a database query interface with the following components:

- SQL Editor:** Displays the SQL code for calculating the percentage change in sales volume between Q1 and Q2 2020.
- Data Output:** A table showing the result of the query. It has one row with the value 3.68.
- Messages:** No messages are present.
- Notifications:** No notifications are present.
- Toolbar:** Includes icons for saving, opening, closing, and running the query.
- Table Headers:** The table has a single header "taux_evolution" of type "numeric".
- Table Data:** One row with the value 3.68.

R8 : Le classement des régions par rapport au prix au mètre carré des appartement de plus de 4 pièces

```
1 v select r.nom_region, round(avg(v.valeur / b.surface_carrez), 0) as prix_m2
2   from bien as b
3   left join vente as v on b.id_bien = v.id_bien
4   left join commune as c on b.id_codedep_codecommune = c.id_codedep_codecommune
5   left join region as r on c.code_reg = r.code_reg
6   where b.type_local = 'Appartement' and b.surface_carrez != 0 and b.nb_piece > 4
7   group by r.nom_region
8   order by round(avg(valeur / surface_carrez), 2) desc;
```

Data Output Messages Notifications

SQL

| | nom_region character varying (100) | prix_m2 numeric |
|----|---------------------------------------|--------------------|
| 1 | Ile-de-France | 8757 |
| 2 | La Réunion | 3636 |
| 3 | Provence-Alpes-Côte d'Azur | 3587 |
| 4 | Corse | 3106 |
| 5 | Auvergne-Rhône-Alpes | 2890 |
| 6 | Nouvelle-Aquitaine | 2465 |
| 7 | Bretagne | 2409 |
| 8 | Pays de la Loire | 2315 |
| 9 | Hauts-de-France | 2189 |
| 10 | Occitanie | 2097 |
| 11 | Normandie | 2015 |
| 12 | Grand Est | 1532 |
| 13 | Centre-Val de Loire | 1453 |
| 14 | Bourgogne-Franche-Comté | 1250 |
| 15 | Martinique | 574 |

R9 : Liste des communes ayant eu au moins 50 ventes au 1er trimestre

```
1 v with vente1 as
2   (select c.commune,count(*) as nb_vente from vente as v  left join bien as b on v.id_bien = b.id_bien left join commune as c on b.id_codedep_codecommune = c.id_codedep_codecommune
3   where v.date between '2020-01-01' and '2020-04-01' group by c.commune)
4 select commune, nb_vente
5 from vente1
6 where nb_vente > 50
7 order by commune
```

Data Output Messages Notifications

| | commune | nb_vente |
|----|----------------------|----------|
| 1 | AJACCIO | 54 |
| 2 | ANGERS | 64 |
| 3 | ANTIBES | 77 |
| 4 | ASNIERES-SUR-SEINE | 81 |
| 5 | BORDEAUX | 157 |
| 6 | BOULOGNE-BILLANCOURT | 99 |
| 7 | COURBEVOIE | 80 |
| 8 | GRENOBLE | 106 |
| 9 | LA CIOTAT | 63 |
| 10 | LEVALLOIS-PERRET | 60 |
| 11 | LILLE | 67 |
| 12 | MARSEILLE 1ER | 71 |
| 13 | MARSEILLE 4EME | 73 |
| 14 | MARSEILLE 8EME | 81 |
| 15 | MARSEILLE 9EME | 66 |
| 16 | MONTRÉUIL | 65 |
| 17 | NANTES | 120 |
| 18 | NICE | 173 |
| 19 | NIMES | 63 |
| 20 | PARIS 02 | 61 |
| 21 | PARIS 03 | 79 |

✓ Exécution réussie. Temps total : 101 msec. 47 lignes affectées. ✘

Total de lignes: 47 sur 47 Requête terminée 00:00:00.101

Lgn 3, Col

R10 : Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un de 3 pièces.

```
1  with
2      prix_2p as (
3          select round(avg(v.valeur / b.surface_carrez), 2) as prix_moyen_2p from vente as v left join bien as b on v.id_bien = b.id_bien
4          where type_local = 'Appartement' and b.nb_piece = 2 and v.valeur is not null),
5      prix_3p as (select round(avg(v.valeur / b.surface_carrez), 2) as prix_moyen_3p from vente as v left join bien as b on v.id_bien = b.id_bien
6          where type_local = 'Appartement' and b.nb_piece = 3 and v.valeur is not null
7      )
8  select *, concat(round(((prix_moyen_3p - prix_moyen_2p) / prix_moyen_2p) * 100, 2), '%') as difference
9  from prix_2p, prix_3p;
10
```

Data Output Messages Notifications

SQL

| | prix_moyen_2p numeric | prix_moyen_3p numeric | difference text |
|---|--------------------------|--------------------------|--------------------|
| 1 | 4908.06 | 4300.35 | -12.38% |

R11 : Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69

```
1 v with prix_par_ville as (
2     select c.commune, c.code_dep, round(avg(v.valeur)) as moyenne, row_number() over (partition by c.code_dep order by avg(v.valeur) desc) as rank
3     from vente as v left join bien as b on v.id_bien = b.id_bien left join commune as c on b.id_codedep_codecommune = c.id_codedep_codecommune
4     where c.code_dep in ('06', '13', '33', '59', '69') and v.valeur is not null
5     group by c.commune, c.code_dep
6 )
7 select commune, code_dep as département, moyenne as prix_moyen
8 from prix_par_ville
9 where rank <= 3
10 order by code_dep, moyenne desc
```

Data Output Messages Notifications

SQL

| | commune | département | prix_moyen |
|----|-----------------------|-------------|------------|
| 1 | SAINT-JEAN-CAP-FERRAT | 06 | 968750 |
| 2 | EZE | 06 | 655000 |
| 3 | MOUANS-SARTOUX | 06 | 476898 |
| 4 | GIGNAC-LA-NERTHE | 13 | 330000 |
| 5 | SAINT SAVOURNIN | 13 | 314425 |
| 6 | CASSIS | 13 | 313417 |
| 7 | LEGE-CAP-FERRET | 33 | 549501 |
| 8 | VAYRES | 33 | 335000 |
| 9 | ARCACHON | 33 | 307436 |
| 10 | BERSEE | 59 | 433202 |
| 11 | CYSOING | 59 | 408550 |
| 12 | HALLUIN | 59 | 322250 |
| 13 | VILLE SUR JARNIOUX | 69 | 485300 |
| 14 | LYON 2EME | 69 | 455217 |
| 15 | LYON 6EME | 69 | 426968 |

R12 : 20 communes avec le plus de vente pour 1000 habitants pour les communes \geq 10 000 habitants.

```
1 v WITH top_commune AS (
2   SELECT code_dep, commune,id_codedep_codecommune, population
3   FROM commune
4   WHERE population > 10000
5 )
6   SELECT
7     c.commune, c.population, COUNT(*) AS nb_vente, round(round((COUNT(v.id_bien)*1000),2) / c.population,2) AS nb_vente_pour_1000_habitants
8   FROM VENTE AS v LEFT JOIN bien AS b ON v.id_bien = b.id_bien
9   INNER JOIN top_commune AS c ON b.id_codedep_codecommune = c.id_codedep_codecommune --pour éviter les valeurs nulls
10  GROUP BY c.commune, c.population
11  ORDER BY nb_vente_pour_1000_habitants DESC
12  LIMIT 20;
```

Data Output Messages Notifications

| commune | population | nb_vente | nb_vente_pour_1000_habitants |
|--------------------------|------------|----------|------------------------------|
| 2 PARIS 01 | 16055 | 79 | 4.92 |
| 3 PARIS 03 | 34306 | 161 | 4.69 |
| 4 ARCACHON | 11898 | 55 | 4.62 |
| 5 LA BAULE-ESCOUBLAC | 16797 | 77 | 4.58 |
| 6 PARIS 04 | 29390 | 120 | 4.08 |
| 7 ROQUEBRUNE CAP MA... | 13041 | 52 | 3.99 |
| 8 PARIS 08 | 36250 | 139 | 3.83 |
| 9 SANARY SUR MER | 17160 | 60 | 3.50 |
| 10 PARIS 09 | 60563 | 208 | 3.43 |
| 11 LA LONDE LES MAURES | 10776 | 37 | 3.43 |
| 12 PARIS 06 | 41171 | 139 | 3.38 |
| 13 SAINT-CYR-SUR-MER | 11725 | 38 | 3.24 |
| 14 CHANTILLY | 11178 | 35 | 3.13 |
| 15 SAINT-MANDE | 22576 | 69 | 3.06 |
| 16 PORNICHET | 11440 | 35 | 3.06 |
| 17 PARIS 10 | 86863 | 264 | 3.04 |
| 18 MENTON | 30981 | 91 | 2.94 |
| 19 SAINT-HILAIRE-DE-RIEZ | 11501 | 33 | 2.87 |
| 20 VINCENNES | 50230 | 141 | 2.81 |

Total de lignes: 20 sur 20 Requête terminée 00:00:00.068

Q/A

Thanks you

