



Mise en place Base de données SQL & Analyse de marché

Icart Matthieu



Laplace Immo

Introduction

Laplace Immo est un réseau immobilier national.

Il vise à créer une base de données pour **collecter les transactions immobilières et foncières en France**, tout **en respectant les normes du RGPD** (*Règlement Général sur la Protection des Données*).

Cette initiative permettra notamment d'**analyser le marché** et de **prévoir au mieux les prix de vente des biens immobiliers**.



Stratégie de sauvegarde des données et conformité RGPD

1 . Identification des données personnelles :

Recensement précis des informations personnelles collectées, stockées ou traitées.

3 . Gestion des consentements :

Si le consentement est la base légale pour la collecte, s'assurer que celui-ci est obtenu de manière explicite, libre, et informée.

5 . Durée de conservation des données :

Déterminer une durée de conservation raisonnable.

2 . Raison et légalité du traitement des données :

Établissement clair de la raison pour laquelle chaque donnée est collectée.

4 . Sécurité et protection des données :

Mettre en place des mesures de sécurité adéquates pour protéger les données contre les accès non autorisés ou les fuites.

Sources des données

- **Référentiel géographique**

- Données issues de ***data.gouv*** sur les communes, unités urbaines, aires urbaines, départements, académies, régions, etc...

- **Référentiel démographique**

- Données de **l'INSEE** contenant les résultats des recensements de la population pour les 34 991 communes françaises.

- **Valeurs foncières**

- Données extraites depuis la plateforme open data des *Demandes de valeurs foncières (DVF)*.

Lexique des données

Un dictionnaire de données permet de définir de manière structurée et détaillée toutes les informations clés sur les données à collecter et à analyser. En identifiant les sources de données, en spécifiant les formats et les types de données, ainsi qu'en décrivant les relations entre les différents éléments, il assure une cohérence et une compréhension claires des informations.

Cela facilite non seulement la création et la maintenance de la base de données, mais aussi l'interprétation précise des résultats lors de l'analyse de marché, garantissant ainsi des décisions stratégiques basées sur des données fiables et bien organisées.

Exemple : Dictionnaire des données de la table « Commune »

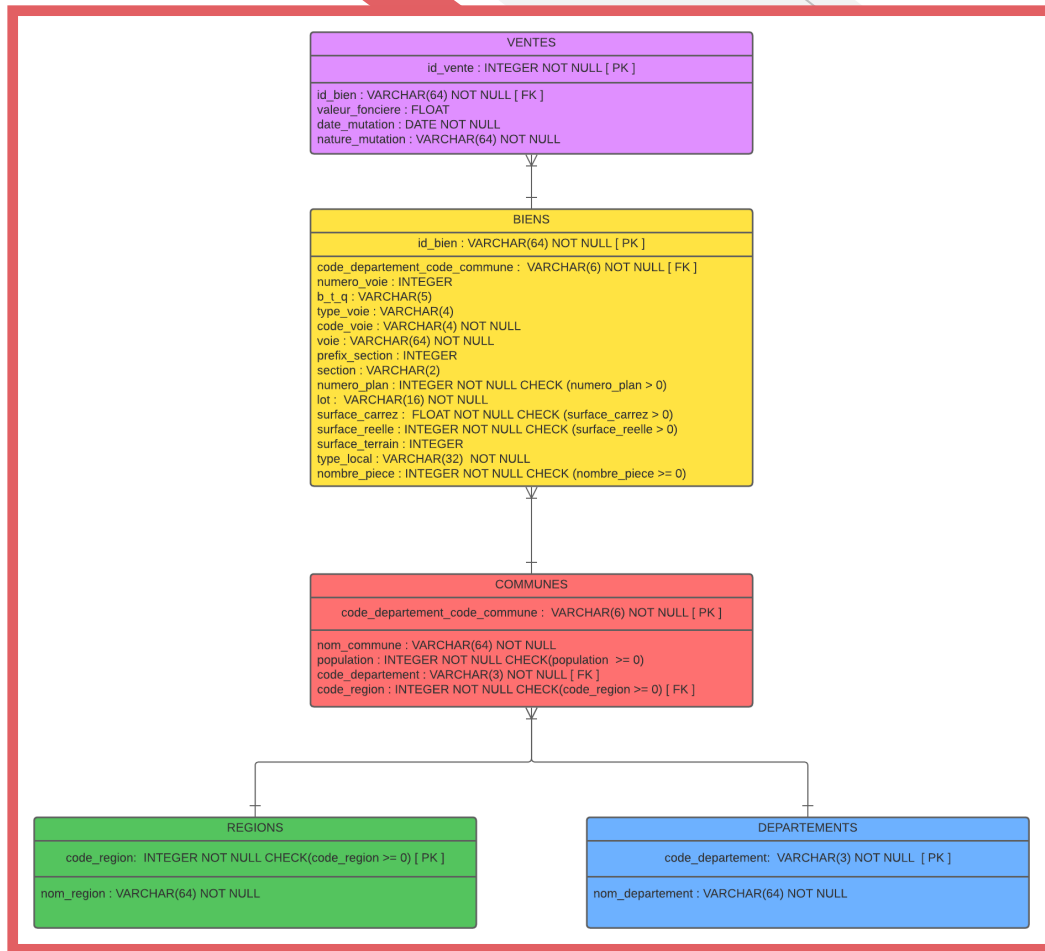
| CODE | SIGNIFICATION | TYPE | LONGUEUR | NATURE | CONTRAINTES | REGLES DE GESTIONS | REGLES DE CALCULS | UNITE DE MESURE | KEY |
|-------------------------------|---|---------|----------|-------------|-------------|--------------------|---------------------------------|-----------------|-------------|
| code_departement_code_commune | Concaténation code département + code commune | VARCHAR | 6 | COMPOSITION | | NOT NULL | Code_departement + Code_commune | | PRIMARY KEY |
| nom_commune | Nom de la commune | VARCHAR | 64 | ELEMENTAIRE | | NOT NULL | | | |
| population | Population totale | INTEGER | NC | ELEMENTAIRE | >= 0 | NOT NULL | | | |
| code_departement | Code du departement | VARCHAR | 3 | ELEMENTAIRE | | NOT NULL | | | FOREIGN KEY |
| code_region | Code de la région | INTEGER | NC | ELEMENTAIRE | >= 0 | NOT NULL | | | FOREIGN KEY |

Schéma relationnel normalisé

Un schéma définit la structure et les relations entre les différentes tables de la base de données, en spécifiant les clés primaires, les clés étrangères et les attributs de chaque entité.

Il permet de visualiser comment les données seront organisées et interconnectées, facilitant ainsi la gestion efficace des informations.

Cela garantit une base de données bien conçue et optimisée, essentielle pour des prévisions précises et une analyse approfondie.





Analyse de marché

Requêtes SQL et résultats

Mise en place de la base de donnée SQL

Mise en place structurelle de la base de données et définitions des contraintes, des différentes entités et de leurs attributs.

Exemple : Script SQL de la table « Commune »

```
*****
*****
CREATE TABLE Communes (
    code_departement_code_commune VARCHAR(6) NOT NULL PRIMARY KEY,
    nom_commune VARCHAR(64) NOT NULL,
    population INTEGER NOT NULL CHECK(population >= 0),
    code_departement VARCHAR(3) NOT NULL,
    FOREIGN KEY (code_departement) REFERENCES
Departements(code_departement)
    code_region INTEGER NOT NULL CHECK(code_region >= 0)
    FOREIGN KEY (code_region) REFERENCES Regions(code_region)
);
*****
*****
```


Requêtes et résultats

Formulation de requêtes SQL pour sélectionner les données pertinentes aux besoins spécifiques de l'analyse et extraire de précieux insights.

Exemple de requête SQL

```
*****  
*****
```

12 / Top 20 des communes enregistrant le plus de transactions
pour 1000 habitants, pour les communes de plus de 10 000 habitants.


```
SELECT nom_commune, COUNT( id_vente ) AS nombre_transactions, population,  
ROUND( ( COUNT( id_vente ) * 1000.00 / population), 2 ) AS  
Transaction_pour_1000_habitants  
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien  
JOIN communes ON communes.code_departement_code_commune =  
biens.code_departement_code_commune  
WHERE population > '10000'  
GROUP BY nom_commune, population ORDER BY Transaction_pour_1000_habitants  
DESC LIMIT 20
```

```
*****  
*****
```

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

```
SELECT COUNT ( id_vente ) AS ventes_appartements_1er_Semetre_2020  
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien  
WHERE type_local ILIKE 'appartement'  
AND date_mutation BETWEEN '2020/01/01' AND '2020/06/30'
```

Results

| | ventes_appartements_1er_semestre_2020  |
|---|---|
| 1 | 31378 |

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

```
SELECT nom_region, COUNT( id_vente ) AS nombre_de_ventes  
  
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien  
JOIN communes ON communes.code_departement_code_commune =  
biens.code_departement_code_commune  
JOIN regions ON regions.code_region = communes.code_region  
  
WHERE type_local ILIKE 'appartement'  
AND date_mutation BETWEEN '2020/01/01' AND '2020/06/30'  
GROUP BY nom_region ORDER BY nombre_de_ventes DESC
```

Results

| | nom_region character varying (64) | nombre_de_ventes bigint |
|----|--------------------------------------|----------------------------|
| 1 | Ile-de-France | 13995 |
| 2 | Provence-Alpes-Cote d'Azur | 3649 |
| 3 | Auvergne-Rhone-Alpes | 3253 |
| 4 | Nouvelle-Aquitaine | 1932 |
| 5 | Occitanie | 1640 |
| 6 | Pays de la Loire | 1357 |
| 7 | Hauts-de-France | 1254 |
| 8 | Grand Est | 984 |
| 9 | Bretagne | 983 |
| 10 | Normandie | 862 |
| 11 | Centre-Val de Loire | 696 |
| 12 | Bourgogne-Franche-Comte | 376 |
| 13 | Corse | 223 |
| 14 | Martinique | 94 |
| 15 | La Reunion | 44 |
| 16 | Guyane | 34 |
| 17 | Guadeloupe | 2 |

3. Proportion des ventes d'appartements en fonction du nombre de pièces.

```
WITH Total AS ( SELECT COUNT ( id_vente ) AS Total_des_ventes
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
WHERE type_local ILIKE 'appartement' )
SELECT nombre_piece, ROUND ( COUNT ( id_vente ) * 100.00 / ( SELECT
Total_des_ventes FROM Total ), 2 ) AS Pourcentage_des_ventes
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
WHERE type_local ILIKE 'appartement'
GROUP BY nombre_piece ORDER BY nombre_piece
```

Results

| | nombre_piece integer | pourcentage_des_ventes 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 |

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

```
SELECT nom_departement,  
ROUND ( AVG ( valeur_fonciere / surface_carrez ) ) AS Prix_moyen_m2  
  
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien  
JOIN communes ON communes.code_departement_code_commune =  
biens.code_departement_code_commune  
JOIN departements ON departements.code_departement =  
communes.code_departement  
  
GROUP BY nom_departement  
ORDER BY Prix_moyen_m2 DESC LIMIT 10
```

Results

| | nom_departement character varying (64) 🔒 | prix_moyen_m2 double precision 🔒 |
|----|---|-------------------------------------|
| 1 | Paris | 12053 |
| 2 | Hauts-de-Seine | 7219 |
| 3 | Val-de-Marne | 5343 |
| 4 | Alpes-Maritimes | 4700 |
| 5 | Haute-Savoie | 4667 |
| 6 | Seine-Saint-Denis | 4345 |
| 7 | Yvelines | 4225 |
| 8 | Rhone | 4059 |
| 9 | Corse-du-Sud | 4027 |
| 10 | Gironde | 3764 |

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

```
SELECT ROUND( AVG ( valeur_fonciere / surface_carrez ) ) AS  
prix_moyen_m2_maison_IDF
```

```
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien  
JOIN Communes ON communes.code_departement_code_commune =  
biens.code_departement_code_commune  
JOIN Regions ON regions.code_region = communes.code_region
```

```
WHERE type_local ILIKE 'maison' AND nom_region ILIKE 'ile-de-France'
```

Results

| | prix_moyen_m2_maison_idf double precision |
|---|--|
| 1 | 3745 |

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

SELECT Ventes.id_bien, valeur_fonciere, surface_carrez, nom_region

FROM Ventes **JOIN** biens **ON** biens.id_bien = ventes.id_bien

JOIN communes **ON** communes.code_departement_code_commune =
biens.code_departement_code_commune

JOIN regions **ON** regions.code_region = communes.code_region

WHERE valeur_fonciere **IS NOT NULL**

ORDER BY valeur_fonciere **DESC LIMIT 10**

Results

| | id_bien character varying (64) 🔒 | valeur_fonciere double precision 🔒 | surface_carrez double precision 🔒 | nom_region character varying (64) 🔒 |
|----|-------------------------------------|---------------------------------------|--------------------------------------|--|
| 1 | 7511669107DD1853 | 9000000 | 9.1 | Ile-de-France |
| 2 | 9117416850AX16322 | 8600000 | 64 | Ile-de-France |
| 3 | 75107104620AK34100 | 8577713 | 20.55 | Ile-de-France |
| 4 | 75117335503CO10015 | 7620000 | 42.77 | Ile-de-France |
| 5 | 7510672499AU14104 | 7600000 | 253.3 | Ile-de-France |
| 6 | 7510188641AZ8679 | 7535000 | 139.9 | Ile-de-France |
| 7 | 75116364141DV465 | 7420000 | 360.95 | Ile-de-France |
| 8 | 7511623794CN2411 | 7200000 | 595 | Ile-de-France |
| 9 | 75101261449BC672 | 7050000 | 122.56 | Ile-de-France |
| 10 | 751011088635AT4181 | 6600000 | 79.38 | Ile-de-France |

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

```
WITH Trimestre_1 AS (  
    SELECT COUNT( id_vente ) AS trimestre_1 FROM ventes  
    WHERE date_mutation BETWEEN '2020/01/01' AND '2020/03/31' ),  
  
    Trimestre_2 AS (  
    SELECT COUNT( id_vente ) AS trimestre_2 FROM ventes  
    WHERE date_mutation BETWEEN '2020/04/01' AND '2020/06/30' )  
  
SELECT Trimestre_1.trimestre_1 AS ventes_trimestre_1,  
    Trimestre_2.trimestre_2 AS ventes_trimestre_2,  
    ROUND( ( Trimestre_2.trimestre_2 - Trimestre_1.trimestre_1 ) * 100.00 /  
    Trimestre_1.trimestre_1, 2 )  
AS Taux_Evolution FROM Trimestre_1, Trimestre_2
```

Results

| | ventes_trimestre_1 bigint | ventes_trimestre_2 bigint | taux_evolution numeric |
|---|------------------------------|------------------------------|---------------------------|
| 1 | 16776 | 17393 | 3.68 |

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

```
SELECT nom_region,  
ROUND( AVG ( valeur_fonciere / surface_carrez ) ) AS prix_moyen_m2  
  
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien  
JOIN Communes ON communes.code_departement_code_commune =  
biens.code_departement_code_commune  
JOIN Regions ON regions.code_region = communes.code_region  
  
WHERE type_local ILIKE 'appartement' AND nombre_piece > 4  
GROUP BY nom_region ORDER BY prix_moyen_m2 DESC
```

Results

| | nom_region character varying (64) | prix_moyen_m2 double precision |
|----|--------------------------------------|-----------------------------------|
| 1 | Ile-de-France | 8770 |
| 2 | La Reunion | 3642 |
| 3 | Provence-Alpes-Cote d'Azur | 3588 |
| 4 | Corse | 3105 |
| 5 | Auvergne-Rhone-Alpes | 2891 |
| 6 | Nouvelle-Aquitaine | 2465 |
| 7 | Bretagne | 2412 |
| 8 | Pays de la Loire | 2316 |
| 9 | Hauts-de-France | 2190 |
| 10 | Occitanie | 2097 |
| 11 | Normandie | 2016 |
| 12 | Grand Est | 1541 |
| 13 | Centre-Val de Loire | 1453 |
| 14 | Bourgogne-Franche-Comte | 1251 |
| 15 | Martinique | 573 |

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

```
SELECT nom_commune, COUNT ( ventes.id_vente ) AS nombre_de_ventes
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien
JOIN communes ON communes.code_departement_code_commune =
biens.code_departement_code_commune
WHERE date_mutation BETWEEN '2020/01/01' AND '2020/03/31'
GROUP BY nom_commune HAVING COUNT( ventes.id_vente ) >= 50
ORDER BY nom_commune ASC
```

Results

| | nom_commune character varying (64) | nombre_de_ventes bigint | | nom_commune character varying (64) | nombre_de_ventes bigint |
|----|---------------------------------------|----------------------------|----|---------------------------------------|----------------------------|
| 1 | Ajaccio | 54 | 25 | Paris 14e Arrondissement | 146 |
| 2 | Angers | 64 | 26 | Paris 15e Arrondissement | 215 |
| 3 | Antibes | 77 | 27 | Paris 16e Arrondissement | 165 |
| 4 | Asnières-sur-Seine | 81 | 28 | Paris 17e Arrondissement | 228 |
| 5 | Bordeaux | 157 | 29 | Paris 18e Arrondissement | 209 |
| 6 | Boulogne-Billancourt | 99 | 30 | Paris 19e Arrondissement | 116 |
| 7 | Courbevoie | 80 | 31 | Paris 20e Arrondissement | 127 |
| 8 | Grenoble | 106 | 32 | Paris 2e Arrondissement | 61 |
| 9 | Issy-les-Moulineaux | 50 | 33 | Paris 3e Arrondissement | 79 |
| 10 | La Ciotat | 62 | 34 | Paris 4e Arrondissement | 60 |
| 11 | Levallois-Perret | 59 | 35 | Paris 5e Arrondissement | 79 |
| 12 | Lille | 67 | 36 | Paris 6e Arrondissement | 86 |
| 13 | Marseille 1er Arrondissement | 71 | 37 | Paris 7e Arrondissement | 87 |
| 14 | Marseille 4e Arrondissement | 72 | 38 | Paris 8e Arrondissement | 62 |
| 15 | Marseille 8e Arrondissement | 81 | 39 | Paris 9e Arrondissement | 106 |
| 16 | Marseille 9e Arrondissement | 66 | 40 | Puteaux | 53 |
| 17 | Montreuil | 65 | 41 | Rennes | 61 |
| 18 | Nantes | 119 | 42 | Rueil-Malmaison | 68 |
| 19 | Nice | 173 | 43 | Saint-Maur-des-Fossés | 56 |
| 20 | Nîmes | 63 | 44 | Sète | 62 |
| 21 | Paris 10e Arrondissement | 109 | 45 | Toulon | 59 |
| 22 | Paris 11e Arrondissement | 169 | 46 | Toulouse | 78 |
| 23 | Paris 12e Arrondissement | 110 | 47 | Versailles | 54 |
| 24 | Paris 13e Arrondissement | 94 | 48 | Vincennes | 68 |

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

```
WITH Appartement_2_pieces AS (  
  SELECT ROUND( AVG( valeur_fonciere / surface_carrez ) ) AS  
    prix_moyen_m2_2_pieces  
  FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien  
  WHERE type_local ILIKE 'appartement' AND nombre_piece = 2 ),
```

```
  Appartement_3_pieces AS (  
    SELECT ROUND( AVG( valeur_fonciere / surface_carrez ) ) AS  
      prix_moyen_m2_3_pieces  
    FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien  
    WHERE type_local ILIKE 'appartement' AND nombre_piece = 3 )
```

```
SELECT Appartement_2_pieces.prix_moyen_m2_2_pieces,  
Appartement_3_pieces.prix_moyen_m2_3_pieces,  
ROUND( ( Appartement_3_pieces.prix_moyen_m2_3_pieces -  
Appartement_2_pieces.prix_moyen_m2_2_pieces ) /  
Appartement_2_pieces.prix_moyen_m2_2_pieces * 100.00 )  
AS Pourcentage_difference  
FROM Appartement_2_pieces, Appartement_3_pieces
```

Results

| | prix_moyen_m2_2_pieces double precision | prix_moyen_m2_3_pieces double precision | pourcentage_difference double precision |
|---|--|--|--|
| 1 | 4909 | 4300 | -12 |

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

SELECT * FROM (

```
SELECT nom_departement,  
ROW_NUMBER() OVER ( PARTITION BY nom_departement  
ORDER BY ROUND( AVG ( valeur_fonciere ) ) DESC ) AS top3,  
nom_commune, ROUND( AVG( valeur_fonciere ) ) AS  
moyenne_valeur_fonciere  
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien  
JOIN communes ON communes.code_departement_code_commune  
= biens.code_departement_code_commune  
JOIN departements ON departements.code_departement =  
communes.code_departement  
WHERE departements.code_departement IN ('6', '13', '33', '59', '69')  
GROUP BY nom_departement, nom_commune )
```

AS ranked_data

```
WHERE top3 <= 3  
ORDER BY nom_departement, top3 ;
```

Results

| | nom_departement character varying (64) | top3 bigint | nom_commune character varying (64) | moyenne_valeur_fonciere double precision |
|----|---|----------------|---------------------------------------|---|
| 1 | Alpes-Maritimes | 1 | Saint-Jean-Cap-Ferrat | 968750 |
| 2 | Alpes-Maritimes | 2 | Eze | 655000 |
| 3 | Alpes-Maritimes | 3 | Mouans-Sartoux | 476898 |
| 4 | Bouches-du-Rhone | 1 | Gignac-la-Nerthe | 330000 |
| 5 | Bouches-du-Rhone | 2 | Saint-Savournin | 314425 |
| 6 | Bouches-du-Rhone | 3 | Cassis | 313417 |
| 7 | Gironde | 1 | Lège-Cap-Ferret | 549501 |
| 8 | Gironde | 2 | Vayres | 335000 |
| 9 | Gironde | 3 | Arcachon | 307436 |
| 10 | Nord | 1 | Bersée | 433202 |
| 11 | Nord | 2 | Cysoing | 408550 |
| 12 | Nord | 3 | Halluin | 322250 |
| 13 | Rhone | 1 | Ville-sur-Jarnioux | 485300 |
| 14 | Rhone | 2 | Lyon 2e Arrondissement | 455217 |
| 15 | Rhone | 3 | Lyon 6e Arrondissement | 426968 |

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

```
SELECT nom_commune, COUNT( id_vente ) AS nombre_transactions,  
population, ROUND( ( COUNT( id_vente ) * 1000.00 / population), 2 ) AS  
Transaction_pour_1000_habitants
```

```
FROM ventes JOIN biens ON biens.id_bien = ventes.id_bien  
JOIN communes ON communes.code_departement_code_commune =  
biens.code_departement_code_commune
```

```
WHERE population > '10000'  
GROUP BY nom_commune, population ORDER BY  
Transaction_pour_1000_habitants DESC LIMIT 20
```

Results

| | nom_commune character varying (64) | nombre_transactions bigint | population integer | transaction_pour_1000_habitants numeric |
|----|---------------------------------------|-------------------------------|-----------------------|--|
| 1 | Paris 2e Arrondissement | 127 | 21735 | 5.84 |
| 2 | Paris 1er Arrondissement | 79 | 16055 | 4.92 |
| 3 | Paris 3e Arrondissement | 161 | 34306 | 4.69 |
| 4 | Arcachon | 55 | 11898 | 4.62 |
| 5 | La Baule-Escoubac | 77 | 16797 | 4.58 |
| 6 | Paris 4e Arrondissement | 120 | 29390 | 4.08 |
| 7 | Roquebrune-Cap-Martin | 52 | 13041 | 3.99 |
| 8 | Paris 8e Arrondissement | 139 | 36250 | 3.83 |
| 9 | Sanary-sur-Mer | 60 | 17160 | 3.50 |
| 10 | Paris 9e Arrondissement | 208 | 60563 | 3.43 |
| 11 | La Londe-les-Maures | 37 | 10776 | 3.43 |
| 12 | Paris 6e Arrondissement | 139 | 41171 | 3.38 |
| 13 | Saint-Cyr-sur-Mer | 38 | 11725 | 3.24 |
| 14 | Chantilly | 35 | 11178 | 3.13 |
| 15 | Pornichet | 35 | 11440 | 3.06 |
| 16 | Saint-Mandé | 69 | 22576 | 3.06 |
| 17 | Paris 10e Arrondissement | 264 | 86863 | 3.04 |
| 18 | Menton | 91 | 30981 | 2.94 |
| 19 | Saint-Hilaire-de-Riez | 33 | 11501 | 2.87 |
| 20 | Vincennes | 141 | 50230 | 2.81 |



Thanks!

Do you have any questions?

matthieuicartpro@gmail.com