Projet base de données évoluées

Abdelillah EL KHOTRI

Université de Nantes, Département Informatique, Nantes, France
abdelillah.el-khotri@univ-nantes.fr
https://github.com/abdelillah48/BDD_Avancee_Project

1 Introduction

Ce projet porte sur l'analyse et la manipulation d'un ensemble de données spécifique à l'aide de requêtes OLAP-SQL et l'implémentation de contrôles d'accès. Pour ce faire, nous avons sélectionné un jeu de données publiques lié aux annonces Airbnb à travers différents pays. Ces données sont fournies par Inside Airbnb et sont disponibles sous la licence CC 0 1.0.Le dataset comprend 494 954 enregistrements répartis en 89 attributs variés, qui incluent des informations détaillées sur les logements, les hôtes, les localisations, et plus encore. Pour la gestion et l'exécution des requêtes sur ces données, nous avons choisi PostgreSQL comme système de gestion de base de données.

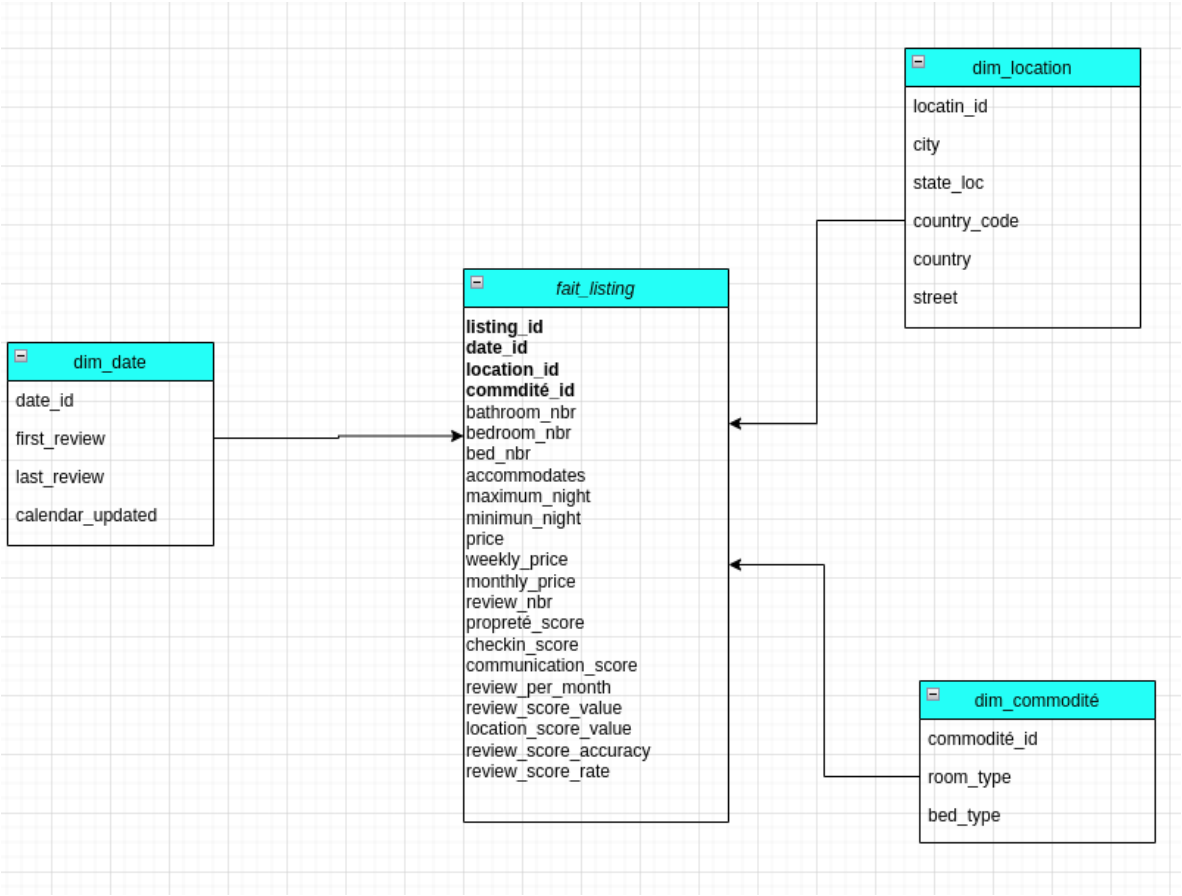
2 Schéma de l'entrepôt de données

Étant donné l'ampleur de notre dataset initial, comprenant 494 954 enregistrements et 89 attributs, une étape cruciale a été la préparation et le nettoyage des données. Pour rendre le dataset plus maniable et pertinent pour notre analyse, nous avons procédé à la suppression des colonnes qui n'étaient pas nécessaires à notre étude. De plus, toutes les lignes contenant des valeurs manquantes ont été éliminées, afin de garantir l'intégrité et la précision de nos requêtes OLAP-SQL. Ces opérations de nettoyage ont été réalisées efficacement à l'aide de la bibliothèque Pandas, un outil puissant pour la manipulation de données en Python.

Pour structurer efficacement notre base de données en vue des requêtes OLAP, nous avons adopté un schéma en étoile, qui est illustré dans la figure ci-dessous. Ce schéma comprend trois tables de dimensions et une table de faits. Les tables de dimensions sont dédiées aux aspects suivants :

- **Location**: Cette table de dimension rassemble des informations précises sur les emplacements des propriétés listées.
- **Date**: Cette table capte les détails temporels associés à chaque annonce et réservation.
- **Commodité**: Elle catalogue les équipements et services proposés avec chaque annonce

La table de faits, quant à elle, correspond au listing des annonces et intègre les clés étrangères des tables de dimensions, permettant ainsi des analyses multidimensionnelles complexes et des agrégations efficaces.



3 Requêtes OLAP

3.1 Nombre de listing par pays

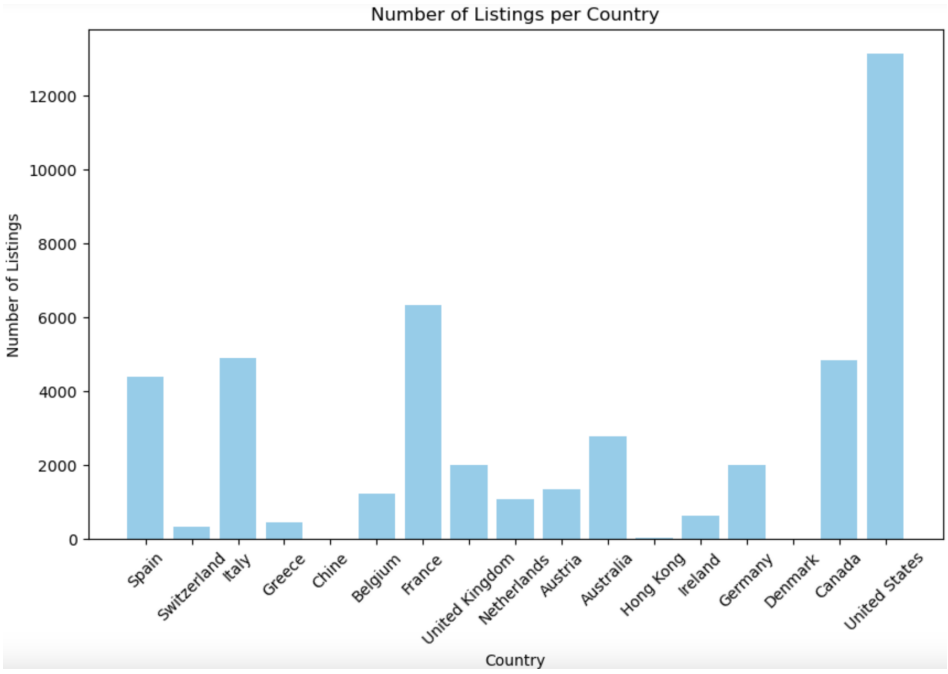
```sql
SELECT l.country, COUNT(*) AS number_of_listings
FROM fait_listing f
JOIN dim_location l ON f.location_id = l.location_id
GROUP BY l.country;
```

En résultat, nous disposons d'une table qui illustre clairement ces informations:

```
postgres=# GROUP BY l.country;
    country     | number_of_listings
----------------+--------------------
 Spain          |               4388
 Switzerland    |                331
 Italy          |               4886
 Greece         |                440
 China          |                  3
 Belgium        |               1231
 France         |               6320
 United Kingdom |               1986
 Netherlands    |               1072
 Austria        |               1336
 Australia      |               2785
 Hong Kong      |                 12
 Ireland        |                613
 Germany        |               1984
 Denmark        |                  9
 Canada         |               4834
 United States  |              13125
(17 rows)

postgres=# c
```

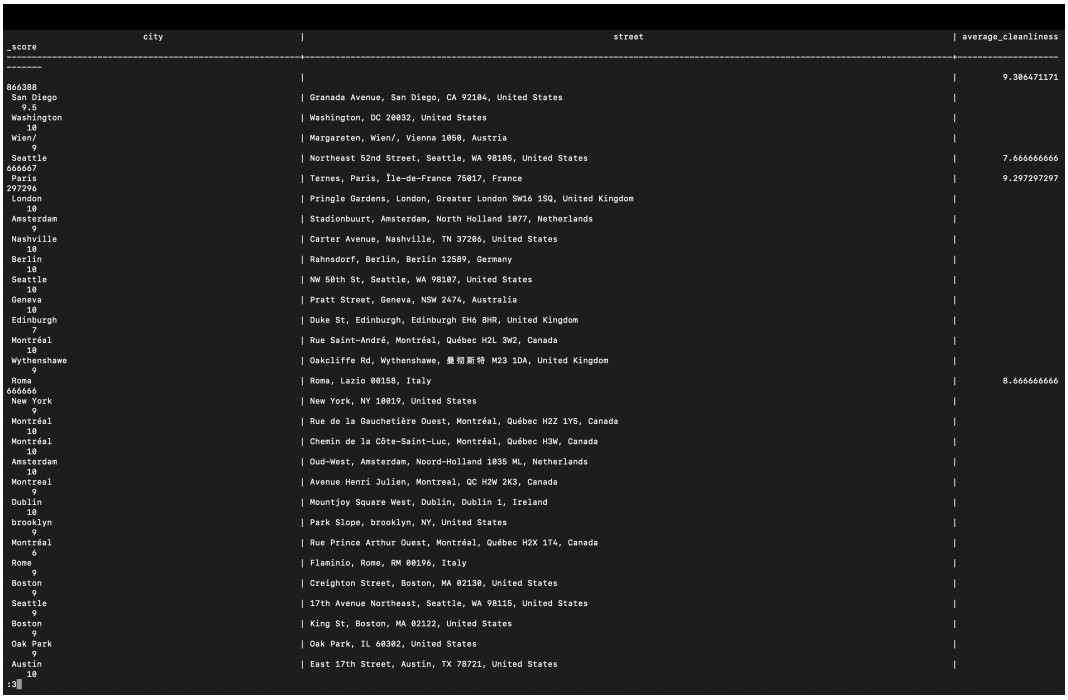
Cette requête a été conçue pour identifier les pays où des annonces Airbnb sont disponibles, ainsi que pour comptabiliser le nombre d'annonces dans chaque pays. L'histogramme ci-dessous illustre clairement ces résultats, fournissant une représentation visuelle efficace de la distribution des annonces par pays.



3.2 Analyse des scores de propreté par ville et quartier

```sql
SELECT
    city,
    street,
    AVG(propreté_score) AS average_cleanliness_score
FROM fait_listing
JOIN dim_location ON fait_listing.location_id = dim_location.location_id
GROUP BY ROLLUP (city, street);
```

En résultat, nous disposons d'une table qui illustre clairement ces informations:



city	street	average_cleanliness_score
		9.386471171866388
San Diego	Granada Avenue, San Diego, CA 92104, United States	9.5
Washington	Washington, DC 20032, United States	10
Wien/	Margareten, Wien/, Vienna 1050, Austria	9
Seattle	Northeast 52nd Street, Seattle, WA 98105, United States	7.666666666666667
Paris	Ternes, Paris, Île-de-France 75017, France	9.297297297297296
London	Pringle Gardens, London, Greater London SW16 1SQ, United Kingdom	10
Amsterdam	Stadionbuurt, Amsterdam, North Holland 1077, Netherlands	9
Nashville	Carter Avenue, Nashville, TN 37206, United States	10
Berlin	Rahnsdorf, Berlin, Berlin 12589, Germany	10
Seattle	NW 50th St, Seattle, WA 98107, United States	10
Geneva	Pratt Street, Geneva, NSW 2474, Australia	10
Edinburgh	Duke St, Edinburgh, Edinburgh EH6 8HR, United Kingdom	7
Montréal	Rue Saint-André, Montréal, Québec H2L 3W2, Canada	10
Wythenshawe	Oakcliffe Rd, Wythenshawe, 曼彻斯特 M23 1DA, United Kingdom	9
Roma	Roma, Lazio 00158, Italy	8.666666666666666
New York	New York, NY 10019, United States	9
Montréal	Rue de la Gauchetière Ouest, Montréal, Québec H2Z 1Y5, Canada	10
Montréal	Chemin de la Côte-Saint-Luc, Montréal, Québec H3W, Canada	10
Amsterdam	Oud-West, Amsterdam, Noord-Holland 1035 ML, Netherlands	10
Montreal	Avenue Henri Julien, Montreal, QC H2W 2K3, Canada	9
Dublin	Mountjoy Square West, Dublin, Dublin 1, Ireland	10
brooklyn	Park Slope, brooklyn, NY, United States	9
Montréal	Rue Prince Arthur Ouest, Montréal, Québec H2X 1T4, Canada	6
Rome	Flaminio, Rome, RM 00196, Italy	9
Boston	Creighton Street, Boston, MA 02130, United States	9
Seattle	17th Avenue Northeast, Seattle, WA 98115, United States	9
Boston	King St, Boston, MA 02122, United States	9
Oak Park	Oak Park, IL 60302, United States	9
Austin	East 17th Street, Austin, TX 78721, United States	10

:3

Cette requête SQL calcule le score moyen de propreté pour chaque rue et ville dans la table des listings, en utilisant une agrégation avec ROLLUP pour inclure des totaux à chaque niveau. Le résultat fournit des moyennes de propreté pour les rues individuelles ainsi que des moyennes globales pour chaque ville.

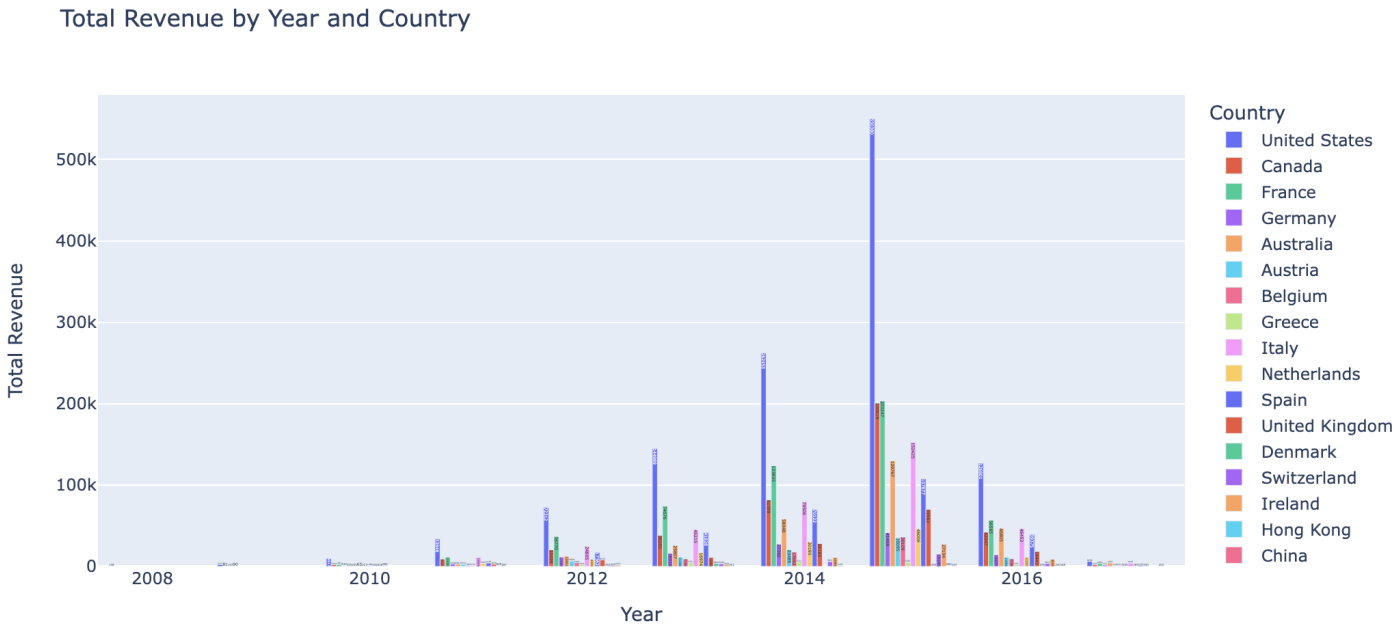
3.3 Analyse des Reviews Airbnb : Une Vue Agrégée par Année et Pays

```sql
SELECT
    EXTRACT(YEAR FROM first_review) AS year,
    country,
    SUM(price * review_nbr) AS total_revenue
FROM fait_listing
JOIN dim_date ON fait_listing.date_id = dim_date.date_id
JOIN dim_location ON fait_listing.location_id = dim_location.location_id
GROUP BY CUBE (EXTRACT(YEAR FROM first_review), country);
```

En résultat, nous disposons d'une table qui illustre clairement ces informations:

year	country	total_revenue
		146427718
2016	France	646520
2016	Greece	50486
2015	Germany	983721
2011	Germany	230535
2016	Netherlands	176626
2017	United States	30217
2011	Denmark	75933
2010	Australia	40968
2014	Australia	2534118
2010	United Kingdom	65774
2010	Netherlands	36609
2016	Switzerland	14531
2012	Australia	744417
2013	France	4035488
2014	Austria	862114
2015	Italy	5518793
2016	Denmark	5934
2017	France	8864
2017	Austria	2348
2011	Austria	258323

La requête SQL déployant la fonction CUBE effectue une agrégation du revenu total des annonces Airbnb, basée sur le prix multiplié par le nombre de critiques, avec des données regroupées par année de première critique et par pays. Cela inclut des totaux pour chaque année, pays, et leurs combinaisons. Le graphique, intitulé **"Total Revenue by Year and Country"**, traduit visuellement ces résultats, fournissant une analyse claire de l'évolution et de la distribution des revenus d'Airbnb.



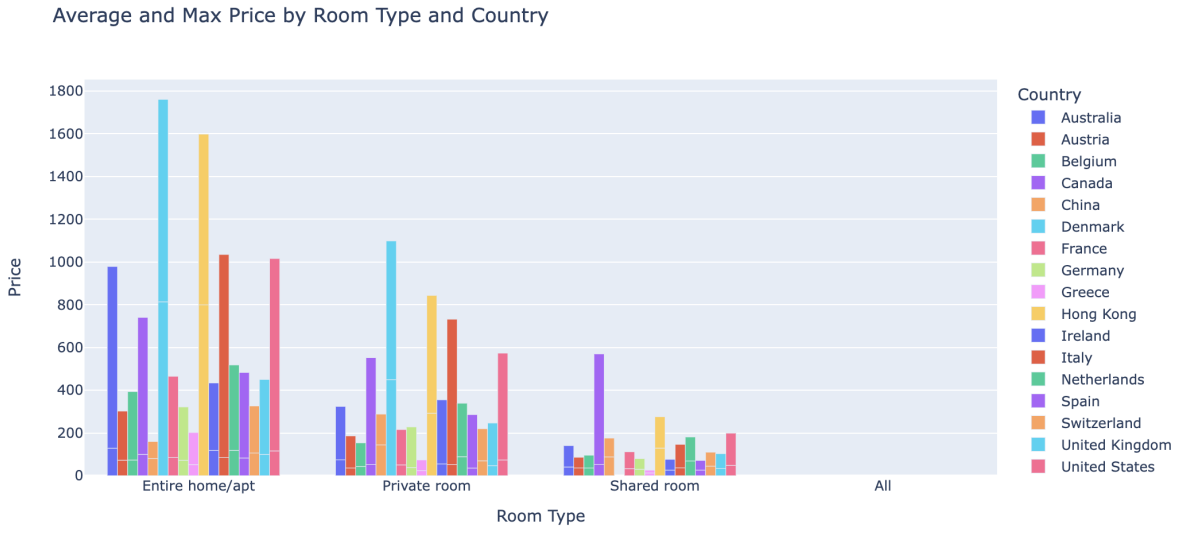
3.4 Analyse de Prix Airbnb : Moyennes et Maximums par Type de Chambre et Pays

```sql
SELECT
    room_type,
    country,
    AVG(price) AS average_price,
    MAX(price) AS max_price
FROM fait_listing
JOIN dim_location ON fait_listing.location_id = dim_location.location_id
JOIN dim_commodité ON fait_listing.commodité_id = dim_commodité.commodité_id
GROUP BY GROUPING SETS ((room_type, country), (room_type), (country));
```

En résultat, nous disposons d'une table qui illustre clairement ces informations:

room_type	country	average_price	max_price
Private room	United States	73.38459302325582	500
Private room	United Kingdom	46.631185807656394	200
Entire home/apt	Spain	83.01835893593106	400
Entire home/apt	Germany	71.92056856187291	250
Shared room	Belgium	36	60
Private room	Denmark	449.4	649
Entire home/apt	Belgium	73.17109826589595	320
Entire home/apt	Australia	128.57798833819243	850
Shared room	Ireland	26.4	50
Entire home/apt	Ireland	117.98220640569394	316
Private room	Spain	35.63139329805996	250
Shared room	Canada	52.659574468085104	517
Entire home/apt	France	84.88045406172401	380
Shared room	France	33.05714285714286	79
Entire home/apt	China	80	80
Private room	Ireland	54.86435331230284	300
Private room	Austria	36.01376146788991	150
Shared room	Australia	40.56	100
Entire home/apt	Austria	72.42356115107914	230
Shared room	Greece	13	13
Private room	China	144	144

Cette requête SQL vise à évaluer les prix des annonces Airbnb en se concentrant sur le type de chambre et le pays. Elle calcule à la fois le prix moyen et le prix maximal pour différentes combinaisons de groupements : par type de chambre et pays, par type de chambre uniquement, et par pays uniquement. Ces agrégations sont facilitées par l'utilisation de GROUPING SETS, permettant une analyse flexible et granulaire des prix sur le marché Airbnb.Le graphique intitulé **"Prix Moyen et Maximum par Type de Chambre et Pays"** nous offre une visualisation précise des résultats obtenus à la suite de l'exécution de la requête correspondante, mettant en lumière les dynamiques de prix sur la plateforme Airbnb.



3.5 Classement des Prix des Annonces Airbnb par Ville et Pays

```sql
SELECT
    l.location_id,
    l.country,
    l.city,
    f.price,
    RANK() OVER (PARTITION BY l.country ORDER BY f.price DESC) AS price_rank
FROM
    dim_location l
JOIN
    fait_listing f ON l.location_id = f.location_id
ORDER BY
    l.country,
    price_rank;
```

En résultat, nous disposons d'une table qui illustre clairement ces informations:

location_id	country	city	price	price_rank
33269	Australia	Zetland	850	1
18962	Australia	Cremorne	750	2
4527	Australia	Sandringham	750	2
18696	Australia	Potts Point	349	4
8215	Australia	Brighton East	300	5
16740	Australia	Clunes	270	6
3917	Australia	Mosman	260	7
18399	Australia	Redfern	250	8
9275	Australia	Kingscliff	250	8
13900	Australia	Yarraville	250	8
1589	Australia	St Kilda West	250	8
26474	Australia	Bondi Beach	250	8
35010	Australia	Dixons Creek	250	8
26448	Australia	Richmond	250	8
22431	Australia	Balgowlah	250	8
41773	Australia	North Balgowlah	250	8
18282	Australia	Waterloo	250	8
17403	Australia	Lovett Bay	250	8
39358	Australia	Redfern	250	8
42320	Australia	Vaucluse	249	20
43345	Australia	Watsons Bay	249	20

La requête sélectionne les annonces Airbnb les plus chères par pays, les classant par prix décroissant. Chaque pays a son classement séparé grâce à RANK(). Pour visualiser bien ces résultats, le graphique ci-dessous montre les deux villes ayant les annonces les plus onéreuses de chaque pays, facilitant l'identification des destinations les plus chères.

4 contrôle d'accès

4.1 Création des roles

Chaque rôle est conçu pour répondre aux besoins spécifiques de nos opérations, en alignant les permissions de manière à maximiser la sécurité et l'efficacité.

- **Role_admin (Administrateur de la base de données)**
  - *Permissions*:
    - Création et suppression de bases de données.
    - Gestion des rôles et des droits d'accès.
    - Surveillance et optimisation des performances.
    - Exécution de sauvegardes et de restaurations.
  - *Objectif*: Assurer l'intégrité, la sécurité et la performance de l'ensemble des systèmes de bases de données.
- **Role_analyste (Analyste de données)**
  - *Permissions*:
    - Accès en lecture à toutes les données nécessaires pour l'analyse.
    - Exécution de requêtes complexes pour générer des rapports.
    - Accès limité en écriture pour des modifications spécifiques, si nécessaire.
  - *Objectif*: Fournir des analyses de données qui aident à la prise de décision.
- **Role_gestionnaire (Gestionnaire de projet ou de département)**
  - *Permissions*:
    - Accès en lecture et écriture aux données relatives à leur domaine.
    - Capacité d'approuver ou de rejeter des modifications.
    - Accès à des fonctions administratives limitées.
  - *Objectif*: Superviser et gérer les opérations au sein de leur domaine de responsabilité.
- **Role_service_client (Agent de service à la clientèle)**
  - *Permissions*:
    - Accès en lecture à l'information client.
    - Mise à jour des dossiers des clients.
    - Création de nouvelles entrées sous supervision.
  - *Objectif*: Offrir un support client de qualité en protégeant la confidentialité des données clients.

5 Conclusion

Ce projet a été une véritable application pratique de nos connaissances acquises en cours. Nous avons débuté par l'importation méticuleuse d'un dataset, en tenant compte des licences appropriées. Ensuite, nous avons utilisé un script Python pour insérer habilement nos données dans une base de données PostgreSQL, en veillant à mettre en place des contrôles d'accès adéquats pour sécuriser nos informations.

La partie la plus captivante du projet a été l'application des requêtes SQL-OLAP avancées, telles que ROLLUP et CUBE, pour analyser nos données sous différents angles et niveaux d'agrégation.

Ce projet a été une opportunité inestimable pour consolider nos compétences en bases de données et pour mettre en pratique les concepts théoriques vus en cours et en travaux pratiques. Nous avons apprécié la méthodologie d'apprentissage adoptée, qui consistait à passer immédiatement des cours magistraux à des travaux pratiques pour renforcer notre compréhension.

De plus, le suivi hebdomadaire de l'avancement du projet a été très bénéfique pour nous maintenir sur la bonne voie et nous assurer que nous progressions de manière constante. En somme, ce projet a été une expérience enrichissante qui a mis en lumière l'importance de l'application pratique dans l'apprentissage des bases de données.