Université de Nantes, Master 1 Informatique

Projet : Intégration et Entrepôt de Données

Membres du groupe

Danneaux Lucas
El Hmidi Yasmina
Mohamed Nafyssata
Oulboub Safaa

Année universitaire : 2024-2025

Introduction

Ce projet a pour but de mettre en œuvre un entrepôt de données à partir de données ouvertes, permettant d'explorer et d'analyser le secteur de la mode via un ensemble de requêtes décisionnelles et des mécanismes de contrôle d'accès avancés.

Sujet du Projet

Le sujet de ce projet est l'**analyse des ventes mondiales dans le secteur de la mode**.Nous avons choisi ce thème car le secteur de la mode représente un marché dynamique, compétitif et en constante évolution. L'étude des ventes au détail à l'échelle mondiale permet d'identifier des tendances de consommation, de mieux comprendre le comportement des clients selon les zones géographiques, et d'optimiser les stratégies commerciales. Le projet aura dans un premier temps été commencé sur des données aériennes avant d'abandonner ce sujet pour intervenir sur celui de la mode.

L'intégralité du projet est disponible au lien suivant : https://gitlab.univ-nantes.fr/E240578Y/airport-database

Sources de Données, Licences et Droits d'Utilisation

- **Nom du dataset :** Global Fashion Retail Sales
- **Format :** CSV
- **Source :** https://www.kaggle.com/datasets/ricgomes/global-fashion-retail-stores-dataset
- **Description :** Ce dataset contient des informations détaillées sur les clients, en mettant l'accent sur leurs attributs personnels, leurs coordonnées et leur localisation géographique. Il s'agit d'une simulation de ventes sur deux ans pour une marque multinationale.
- **Licence :** Attribution 4.0 International (CC BY 4.0) , cette licence permet l'utilisation, la modification et la distribution du dataset, y compris à des fins commerciales, à condition de créditer l'auteur original.

Aucun besoin de vérifier la cohérence des licences car notre projet ne se base que sur un seul dataset accessible par Kaggle.

Intégration des Données

L'ensemble du développement du projet a été réalisé sur des installations locales de Oracle en version 11 à cause de nombreux problèmes rencontrés lors du début du développement sur les installations de l'Université (Notamment des problèmes d'espace).

Le traitement des données récupérées sur Kaggle a été réalisé à l'aide du notebook python traitement.ipynb. Pour pouvoir exécuter ce notebook, il faut au préalable télécharger les données en format CSV depuis Kaggle et les placer dans un répertoire "Old_Data" dans le projet (Impossible de les ajouter dans le dépôt Gitlab du projet à cause de leurs volumes)

Le notebook permet de réaliser le traitement des données pour générer dans le répertoire "Data" les fichiers .csv qui serviront pour l'insertion des données dans la base de données Oracle.

Résumé des opérations réalisations dans le notebook traitement.ipynb :

- Lecture des fichiers .csv du répertoire “Old_Data”.
- Sélection des champs qui sont présents dans le schéma en étoile et abandon des autres champs inutiles.
- Calcul de l’âge des clients à partir de leur date de naissance.
- Traduction des caractères sinogrammes pour les villes situé en Chine (Utilisation de ChatGPT pour aider à la traduction)
- Remplacement des caractères spéciaux par des caractères ASCII pour éviter des erreurs lors des insertions.
- Calcul de la valeur de la clé primaire pour la dimension Discount, Date et la table de Fait.
- Création de la dimension date à partir de toutes les dates récupérées depuis le fichier transactions.csv.
- Création de nouveaux champs dans la dimension Date en extrayant des informations de la date récupérée.
- Réalisation de jointure entre la table de Fait et les Dimensions Date, Products et Discounts.
- Suppression de toutes les lignes ne correspondant pas à une période de Promotion.

La Création des différentes tables de la base de données est ensuite réalisé à l’aide du fichier creation_table.sql qui est exécutable avec la commande suivante depuis sqlplus (seulement si sqlplus est lancé depuis un terminal qui est dans la racine du projet): `@script/creation_table.sql`

Une fois les tables crées, les données y sont insérés en utilisant des fichiers .ctl qui manipulent les données contenu dans les fichiers .csv et qui sont exécutable depuis un terminal de commande avec la commande suivante : `sqlldr userid=utilisateur/mdp@nom_bd control=script/Dim_Customer.ctl` (Exemple pour le fichier Dim_Dustomer.ctl)

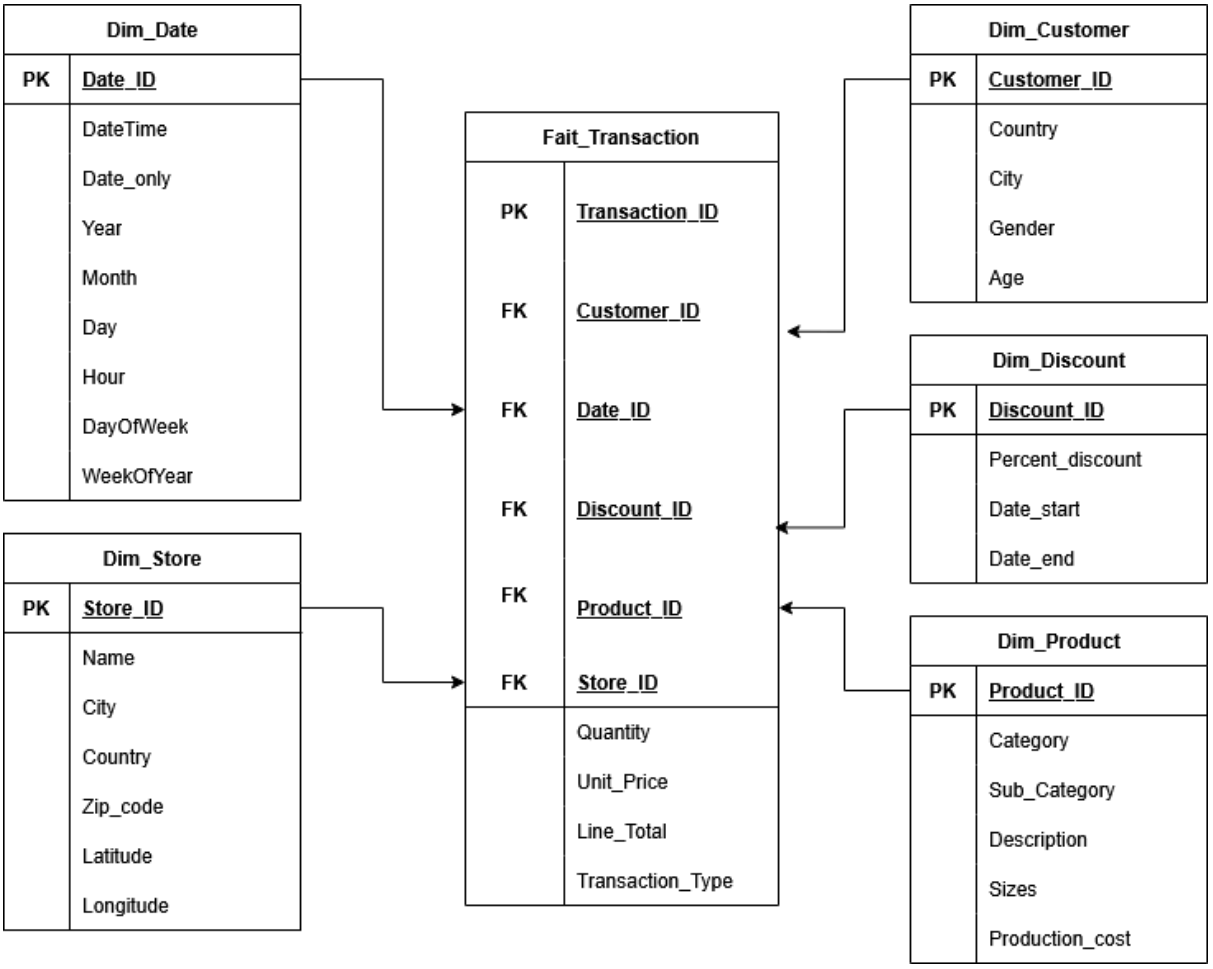
Concernant l’exécution des fichiers .ctl, l’ordre des dimensions importe peu mais il est obligatoire d’insérer les données de la table de fait avec le fichier Fait_Transaction.ctl en dernier pour éviter de rencontrer des erreurs à cause des clefs étrangères.

Pour plus de détails sur l’ensemble des commandes pour pouvoir exécuter l'ensemble du projet, ces dernières sont listées dans le README.md du projet avec plus de détails sur le réglage des différents paramètres.

Conception de l'Entrepôt de Données

Nous avons opté pour une modélisation en étoile, avec une table de faits centrale contenant les transactions de vente (Faits_transactions) , et des tables de dimensions pour les produits (Dim_product) , les magasins (Dim_store) ,les promotions (Dim_discount) et la date (Dim_date). Une dernière table de dimension sur les clients (Dim_Client) a été ajouté au cours du projet

Le schéma réalisé et exploité au cours de ce projet est le suivant :



Description des champs de l'entrepôt

Table de Dimension sur le Client

Attribut	**Type dans Oracle**	**Description**
country	VARCHAR2(20)	pays du client EX : United States
city	VARCHAR2(50)	ville du client EX : New York
gender	VARCHAR2(1)	Sexe du client EX : M ou F
age	NUMBER(2)	age du client EX : 27

Table de Dimension sur les Promotions

Attribut	Type dans Oracle	Description
date_start	Date	date de début de la promotion EX : 2020-01-01
date_end	Date	date de fin de la promotion EX : 2020-01-10
percent_discount	Number(3,2)	Pourcentage de la promotion EX : 0.25 (pour 25% de réduction)

Table de Dimension sur les Magasins

Attribut	Type dans Oracle	Description
store_name	VARCHAR2(30)	Nom du magasin EX : Store Berlin
city	VARCHAR2(20)	Ville dans laquelle se situe le magasin EX : Berlin
country	VARCHAR2(20)	Pays dans lequel se situe le pays EX : Deutschland
zip_code	VARCHAR2(20)	Code postal de la ville EX : 10115 (pour Berlin) ou L1 1AA (pour Liverpool)
latitude	NUMBER(9,6)	Latitude de la position du magasin EX : 52.5125 (pour Berlin)
longitude	NUMBER(9,6)	Longitude de la position du magasin EX : 13.3903 (pour Berlin)

Table de Dimension sur la Date

Attribut	**Type dans Oracle**	**Description**
datetime	Timestamp	date et heure de l'enregistrement EX : 2023-01-01 00:00:00
date_only	Date	seulement la date EX : 2023-01-01
year	Number(4)	Annee de la date EX : 2023
month	Number(2)	Mois de la date EX : 1 (pour janvier)
day	Number(2)	Jour de la date EX : 13
hour	Number(2)	Heure de la date EX : 4
dayofweek	Number(1)	Numéro du jour EX : 1 (pour lundi)
weekofyear	Number(2)	Numéro de la semaine

Table de Dimension sur les Produits

Attribut	**Type dans Oracle**	**Description**
category	VARCHAR2(20)	Catégorie de personne à laquelle correspond ce produit EX : Masculine, Feminine ou Children
sub_category	VARCHAR2(50)	Sous catégorie de produit EX : Suits and Sets (pour des costumes et ensembles)
description	VARCHAR2(100)	Description courte du produit en français EX : Chemisier En Coton De Base
sizes	VARCHAR2(30)	Ensemble des tailles dans laquelle le produit est disponible EX : S|M|L|XL
production_cost	NUMBER(5,2)	Coût de production du produit EX : 15.8

Table de Fait sur les Transactions

Attribut	Type dans Oracle	Description
quantity	NUMBER(2)	Quantité de présence de l'article dans la transaction EX : 1
unit_price	NUMBER(6,2)	Prix de l'article au moment de la vente EX : 75.0
line_total	NUMBER(6,2)	Coût total de la transaction EX : 45.0
transaction_type	VARCHAR2(10)	Type de la transaction EX : Sale ou Return

Contrôle d'accès avec VPD (Virtual Private Database)

Pour renforcer la sécurité et la confidentialité des données dans notre entrepôt, nous avons mis en place des politiques de contrôle d'accès basées sur la technologie VPD (Virtual Private Database). L'objectif est de restreindre dynamiquement l'accès aux données en fonction du rôle de l'utilisateur, sans dupliquer les tables.

- Rôles définis et politiques associées

	Dim_product	Dim_Store	Dim_Discount	Dim_Customer	Facts_transactions
Directeur régional	SELECT	SELECT	NONE	NONE	SELECT (via VPD Store_ID)
Analyste	SELECT (VPD pour le pays donné)	SELECT (VPD pour le pays donné)	SELECT (VPD pour le pays donné)	SELECT (VPD pour le pays donné)	SELECT (VPD pour le pays donné)
Client	NONE	NONE	NONE	VPD (consulter ses infos)	SELECT (VPD sur ses transactions uniquement)

Trois rôles ont été définis pour simuler différents profils d'accès :

- `directeur régional` : Accès au transaction du magasin auquel il est associé à l'aide d'un VPD.
- `analyste` : accès aux données des magasins appartenant à un seul pays à l'aide d'un VPD mis en place sur le pays qui est associé à l'utilisateur.
- `client` : accès uniquement à ses propres données à l'aide d'un VPD mis en place sur seulement ces propres données.

La mise en place de ces rôles permet de restreindre l'accès aux données de la base et ainsi de garantir la sécurité des données aux personnes concernées.

Les politiques de sécurités ont étés défini dans le fichier controlAccessVPD.sql qui peut être exécuté depuis sqlplus (seulement si sqlplus est lancé depuis un terminal qui est dans la racine du projet): @script/controlAccessVPD.sql

Initialement, ces politiques VPD (Virtual Private Database) n'étaient pas pleinement fonctionnelles. Après correction de la fonction vpd_analyste_country, nous avons validé que les politiques s'appliquent désormais correctement. En particulier, nous avons mis en place un filtrage dynamique sur la colonne country à l'aide du contexte d'application Oracle (SYS_CONTEXT).

Cela permet de restreindre automatiquement les lignes visibles pour chaque utilisateur en fonction de son rôle et de son pays.

Par exemple :

1. L'utilisateur analyste_FRANCE ne voit que les transactions associées aux magasins situés en France : 82 766 lignes (voir figure1).

```
SQL> BEGIN
  2    ADMIN27.set_app_ctx_pkg.set_context;
  3  END;
  4  /

ProcÚdure PL/SQL terminÚe avec succÞs.

SQL> SELECT SYS_CONTEXT('app_ctx', 'country') AS country,
  2         SYS_CONTEXT('app_ctx', 'role') AS role
  3  FROM dual;

COUNTRY
--------------------------------------------------------------------
----------------
ROLE
--------------------------------------------------------------------
----------------
FRANCE
ANALYSTE


SQL> SELECT count(*) FROM ADMIN27.FAIT_TRANSACTION;

  COUNT(*)
----------
     82766
```

Figure 1 : Contexte analyste_FRANCE avec 82 766 lignes visibles

2. L'utilisateur analyste_CHINA ne voit que les transactions associées aux magasins situés en Chine : 203 870 lignes (voir figure 2).

Figure 2 : Contexte analyste_CHINA avec 203 870 lignes visibles

Ce comportement est entièrement transparent pour l'utilisateur : il peut exécuter une requête simple du type SELECT * FROM Fait_Transaction, et le filtre est injecté dynamiquement par Oracle.

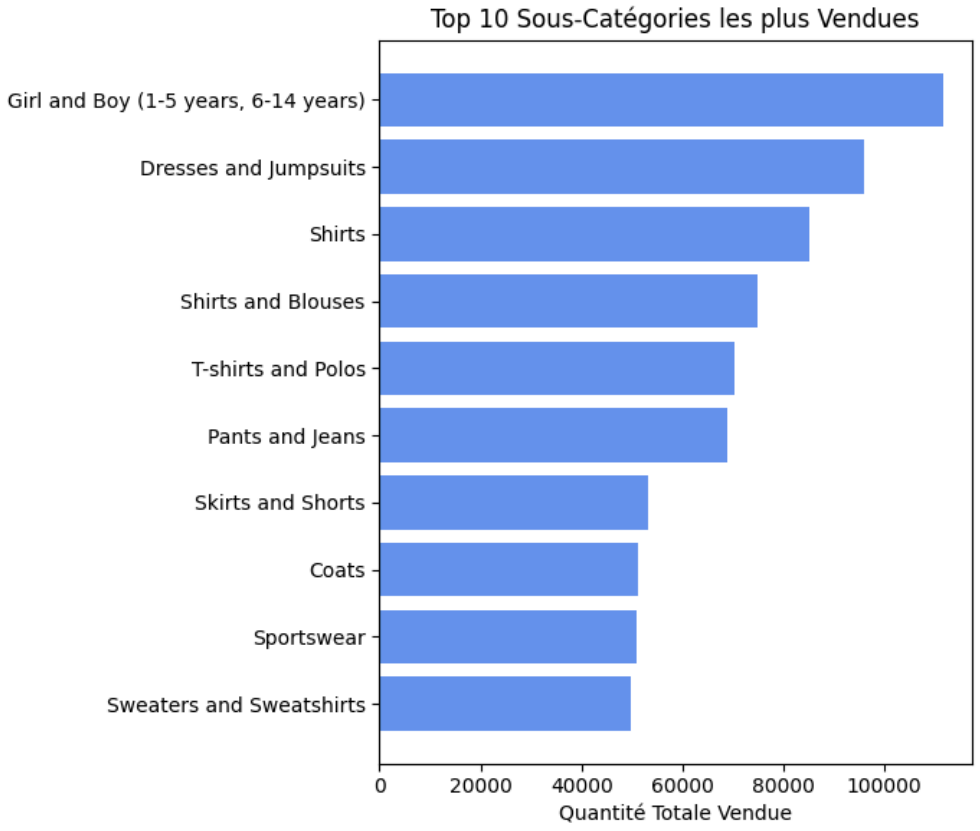
Requêtes OLAP et analyses

Les trois premières requêtes SQL utilisées pour les visualisations sont contenues dans le fichier connection_to_db.ipynb. Les autres requêtes du projet sont regroupées dans le fichier requete.sql, exécutable depuis SQL*Plus (seulement si sqlplus est lancé depuis un terminal qui est dans la racine du projet): @script/requete.sql

Top 10 des sous-catégories les plus vendues

```sql
SELECT p.Sub_Category , SUM(f.Quantity) AS Total_Quantite
FROM Fait_Transaction f
JOIN Dim_Product p ON f.Product_ID = p.Product_ID
GROUP BY p.Sub_Category
ORDER BY Total_Quantite DESC
FETCH FIRST 10 ROWS ONLY
```

Le graphique obtenu (voir figure ci-dessous) met en évidence les sous-catégories les plus populaires. On remarque notamment que la catégorie "Girl and Boy (1-5 years, 6-14 years)" se démarque nettement avec un volume supérieur à 100 000 unités.

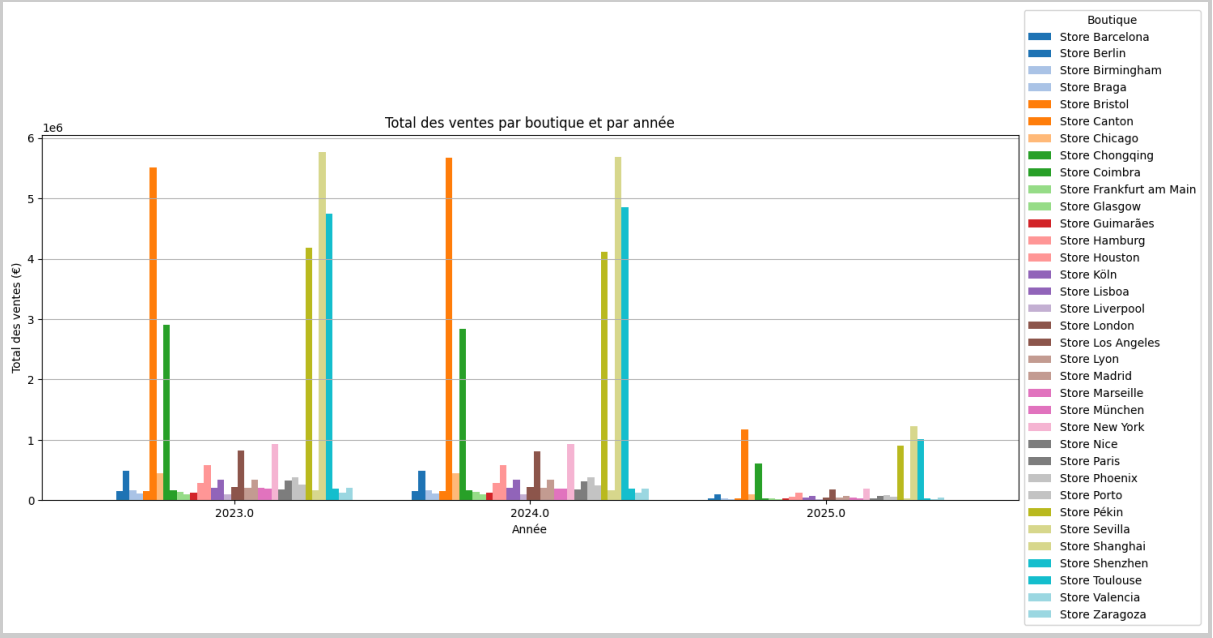


Total des ventes par boutique et par année avec ROLLUP

```sql
SELECT
    S.Name AS Boutique,
    D.Year AS Annee,
    SUM(F.Line_Total) AS Total_Ventes
FROM
    Fait_Transaction F
JOIN
    Dim_Store S ON F.Store_ID = S.Store_ID
JOIN
    Dim_Date D ON F.Date_ID = D.Date_ID
WHERE
    F.Transaction_Type = 'Sale'
GROUP BY
    ROLLUP(S.Name, D.Year)
```

Le graphique présente le montant total des ventes pour chaque boutique, réparti par année. On remarque que certaines boutiques comme Canton, Pékin et Shenzhen génèrent un chiffre d'affaires nettement supérieur aux autres, notamment en 2023 et 2024.

Cette visualisation permet d'identifier les meilleures boutiques en termes de ventes et de suivre leur évolution d'une année à l'autre.

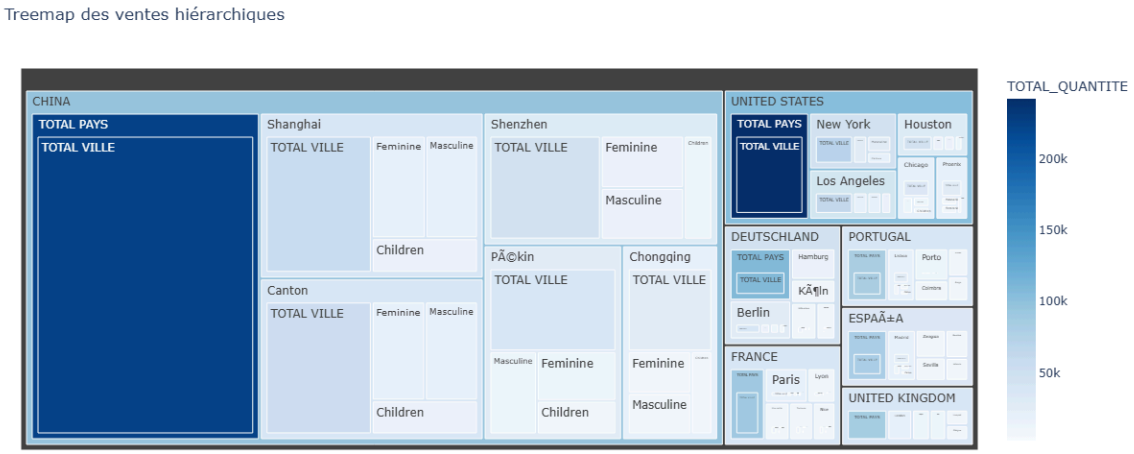


Ventes hiérarchisées par pays, ville et catégorie

Cette requête regroupe les ventes par pays, ville et catégorie, et met en évidence les zones géographiques les plus actives. La Chine se démarque nettement en tête.

```sql
SELECT
    s.Country,
    s.City,
    p.Category,
    SUM(f.Quantity) AS Total_Quantite,
    SUM(f.Line_Total) AS Chiffre_affaires
FROM Fait_Transaction f
JOIN Dim_Store s ON f.Store_ID = s.Store_ID
JOIN Dim_Product p ON f.Product_ID = p.Product_ID
WHERE s.Country IS NOT NULL
GROUP BY ROLLUP(s.Country, s.City, p.Category)
ORDER BY 1,2,3
```

La figure ci dessous représente :Treemap des ventes (quantité) par pays, ville et catégorie de produit



Classement des 10 villes sans magasins avec le plus de client

COUNTRY	CITY	Nombre de Ventes
China	Guangzhou	36352
China	Dongguan	1850
China	Huizhou	1832

On remarque que la ville de Guangzhou en Chine est la ville avec le plus de transactions alors qu'elle ne possède pas de magasin. On en déduit qu'il pourrait être judicieux de réaliser une expertise plus poussée pour ajouter une boutique dans cette ville. Ne peut pas être réalisé par l'un des rôles créés dans ce projet.

Groupement des boutiques par année en fonction de la somme de leurs ventes

NAME	YEAR	Montant Gagne	VENTES
Store Koln	2023	265722,95	8595
Store Koln	2024	174021,7	5714
Store Koln	2025	22108,15	801
Store Koln		461852,8	15110
Store Lyon	2023	176130,65	5673
Store Lyon	2024	238584,66	7702
Store Lyon	2025	41133,74	1465
Store Lyon		455849,05	14840

Cette requête permet de visualiser les ventes et montant gagné par chaque boutique chaque année et ainsi suivre les fluctuations des ventes chaque année. Ne peut pas être réalisé par l'un des rôles créés dans ce projet.

Conclusion

Ce projet nous a permis d'appliquer de manière concrète les concepts d'intégration, de modélisation en étoile, de requêtes OLAP et de contrôle d'accès. Nous avons acquis une expérience pratique complète dans la construction d'un entrepôt de données sécurisé et interrogeable efficacement.