

Projet : Intégration et Entrepôt de Données

Membres du groupe

Danneaux Lucas
El Hmidi Yasmina
Mohamed Nafyssata
Oulboub Safaa

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 réalisées 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 situées 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 réalisée 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 créées, les données y sont insérées en utilisant des fichiers .ctl qui manipulent les données contenu dans les fichiers .csv et qui sont exécutables depuis un terminal de commande avec la commande suivante : `sqlplus /nolog`
`userid=utilisateur/mdp@nom_bd control=script/Dim_Customer.ctl` (Exemple pour le fichier [Dim_Customer.ctl](#))

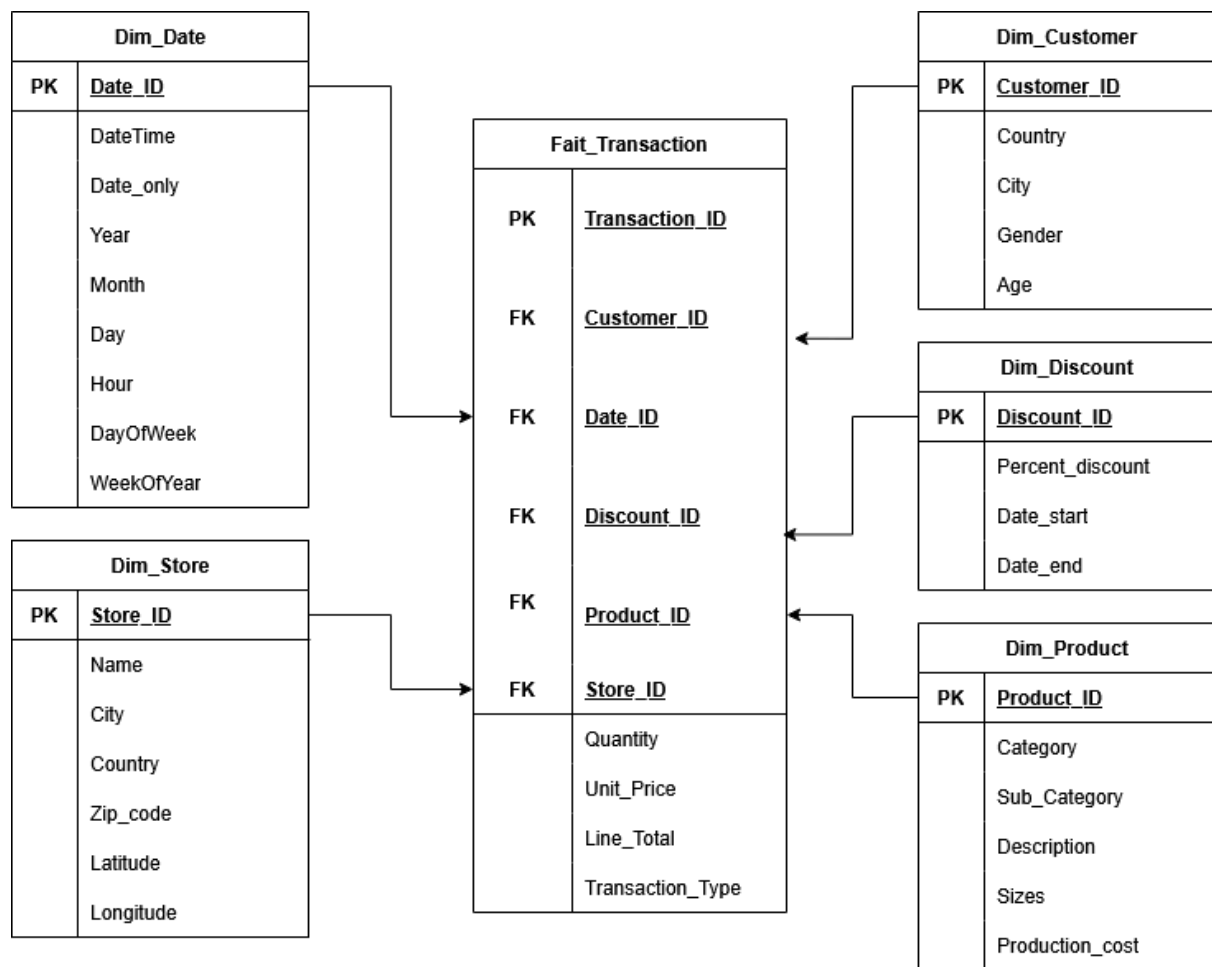
Concernant l'exécution des fichiers .ctl, l'ordre des dimensions importe peu mais il est obligatoire d'insérer les données de la table de fait avec le fichier [Fait_Transaction.ctl](#) en dernier pour éviter de rencontrer des erreurs à cause des clefs étrangères.

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 ajoutée 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 été définies 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`

Cependant, les politiques de VPD réalisées ne sont pas pleinement fonctionnelles, il ne faut donc pas exécuter ce fichier pour pouvoir correctement exécuter les requêtes OLAP.

Requêtes OLAP et analyses

L'ensemble des requêtes rédigées dans le cadre de ce projet sont contenu dans le fichier [requete.sql](#) qui peut être exécuté depuis sqlplus (seulement si sqlplus est lancé depuis un terminal qui est dans la racine du projet): `@script/requete.sql`

Exemples de requêtes développés

Total des ventes par pays

```
SELECT s.Country, SUM(f.Line_Total) AS Total_Ventes
FROM Fait_Transaction f
JOIN Dim_Store s ON f.Store_ID = s.Store_ID
GROUP BY s.Country;
```

Objectif :

Cette requête a pour but de déterminer le chiffre d'affaires total généré dans chaque pays où la marque est implantée. Elle permet d'identifier les marchés les plus importants en termes de revenus

Total des ventes par pays et ville avec **ROLLUP**

```
SELECT s.Country, s.City, SUM(f.Line_Total) AS Total_Ventes
FROM Fait_Transaction f
JOIN Dim_Store s ON f.Store_ID = s.Store_ID
GROUP BY ROLLUP (s.Country, s.City);
```

Analyse : Pour chaque pays, les ventes sont réparties par ville avec un total cumulé par pays. Par exemple, en France, Paris réalise 1,65 million d'euros, tandis que le total France est de 5,86 millions.

Top 5 des produits les plus vendus (en quantité)

```
SELECT * FROM (
    SELECT p.Description, SUM(f.Quantity) AS Total_Qte,
           RANK() OVER (ORDER BY SUM(f.Quantity) DESC) AS Rang
    FROM Fait_Transaction f
    JOIN Dim_Product p ON f.Product_ID = p.Product_ID
```



```
GROUP BY p.Description
) AS classement
WHERE Rang <= 5;
```

Objectif : Identifier les produits les plus populaires.

Analyse : Les 5 produits les plus vendus sont mis en évidence selon le volume total commandé.

Classement des magasins par pays selon les ventes

```
SELECT s.Country, s.Name, SUM(f.Line_Total) AS Total,
       RANK() OVER (PARTITION BY s.Country ORDER BY SUM(f.Line_Total)
DESC) AS Rang
FROM Fait_Transaction f
JOIN Dim_Store s ON f.Store_ID = s.Store_ID
GROUP BY s.Country, s.Name;
```

Analyse : Pour chaque pays, les magasins les plus performants sont identifiés et classés en fonction du chiffre d'affaires généré. Cela permet de repérer facilement les points de vente les plus rentables à l'échelle locale.

11. Classement des 10 villes sans magasins avec le plus de client

COUNTRY		CITY

Nombre de Ventes		

China	36352	Guangzhou
China	1850	Dongguan
China	1832	Huizhou

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.

12. 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.