Université des Sciences et de la Technologie Houari Boumediene

USTHB

Faculté d'Informatique

Projet Business Intelligence

Analyse des Ventes - Base Northwind

Réalisé par :

Ines Souai

Année Universitaire 2024/2025

Table des matières

1 Introduction

1.1 Contexte du projet

Dans le cadre du module Business Intelligence (BI), l'objectif est de mettre en œuvre une chaîne décisionnelle complète sur un cas d'étude réel : la base de données Northwind. Cette base simule l'activité commerciale d'une société de distribution incluant la gestion des clients, commandes, produits, employés et expéditions.

Le projet consiste à :

— Exploiter plusieurs sources de données hétérogènes (fichiers Excel et SGBD SQL Server)
— Concevoir et implémenter un processus ETL en Python
— Construire un schéma en étoile adapté à l'analyse des ventes
— Développer un tableau de bord interactif avec Streamlit
— Documenter l'ensemble de la démarche technique

1.2 Objectifs

Les objectifs principaux du projet sont :

1. Comprendre le cycle complet d'un projet BI : de la source opérationnelle (OLTP) au reporting décisionnel
2. Mettre en pratique les concepts de schémas dimensionnels (étoile), dimensions, faits et granularité
3. Maîtriser les indicateurs de performance (KPI) pour l'aide à la décision
4. Implémenter un ETL en Python sans outil graphique
5. Développer un dashboard web interactif pour l'analyse des ventes

2 Sources de données

2.1 Vue d'ensemble

Le projet repose sur deux familles de sources complémentaires :

— Une base de données SQL Server Northwind (source principale)
— Une série de fichiers Excel dans le dossier `data/excel/`

Cette approche multi-source répond au cahier des charges en exploitant des données hétérogènes.

2.2 Source SQL Server

La base Northwind est installée sur `localhost\SQLEXPRESS`. Elle contient les tables relationnelles suivantes :

— **Orders** et **Order Details** : Commandes et détails
— **Customers** : Informations clients
— **Products** et **Categories** : Catalogue produits
— **Employees** : Personnel commercial

— **Shippers** : Transporteurs

Ces tables représentent le système transactionnel (OLTP) de l'entreprise et constituent la source principale pour l'entrepôt de données.

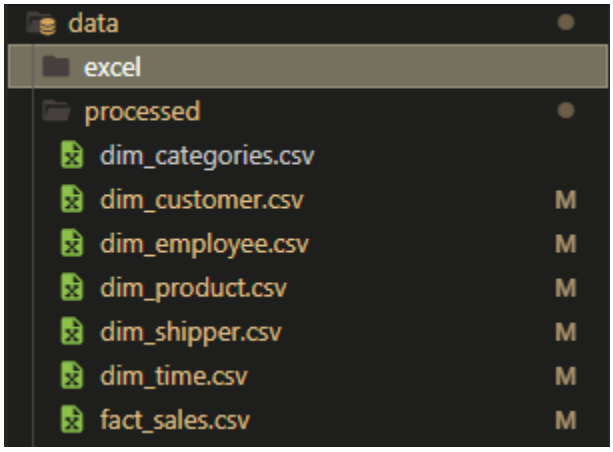

FIGURE 1 – Structure des fichiers CSV générés par l'ETL

2.3 Sources Excel

Les fichiers Excel dans `data/excel/` comprennent :

Fichier	Lignes	Colonnes
Orders.xlsx	48	20
Order_Details.xlsx	58	10
Customers.xlsx	29	18
Products.xlsx	45	14
Employees.xlsx	9	18
Shippers.xlsx	3	18

TABLE 1 – Fichiers Excel sources

Ces fichiers servent à :

— Illustrer le cas réel de données métier externes
— Valider les traitements ETL à petite échelle
— Démontrer la capacité de fusion multi-sources

3 Conception de l'entrepôt de données

3.1 Choix du schéma en étoile

Pour répondre aux objectifs d'analyse (ventes dans le temps, par client, produit, employé, etc.), un schéma en étoile a été adopté avec :

— Une table de faits centrale : `fact_sales`
— Six dimensions : `dim_time`, `dim_customer`, `dim_product`, `dim_employee`, `dim_shipper`, `dim_categories`

La granularité de la table de faits est définie au niveau de la commande (OrderID), avec agrégation des détails.

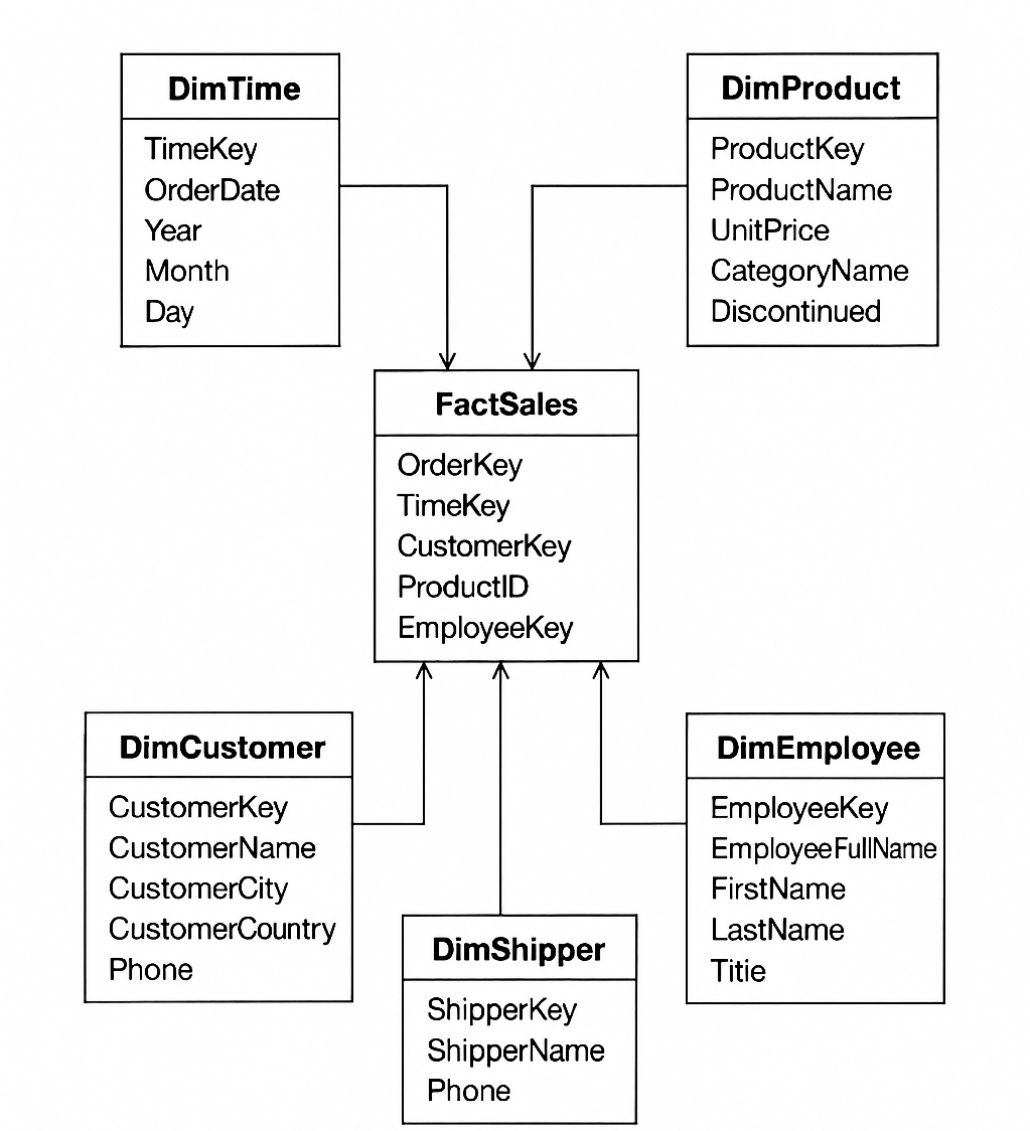

FIGURE 2 – Schéma en étoile de l'entrepôt Northwind

3.2 Table de faits : fact_sales

La table de faits contient les mesures agrégées et les clés étrangères :
Clés d'analyse :
— OrderKey, TimeKey, CustomerKey
— ProductKey, EmployeeKey, ShipperKey
Mesures agrégées :

— DetailCount : Nombre de lignes de détail
— TotalQuantity : Quantité totale
— AverageDiscount : Remise moyenne
— TotalLineTotal : Chiffre d'affaires total
— Freight : Frais de transport

3.3 Dimensions

3.3.1 dim_time

Dimension temporelle avec :
— TimeKey (format YYYYMMDD)
— date, year, month, day

3.3.2 dim_customer

Informations clients :
— CustomerKey, CustomerName
— CustomerCity, CustomerCountry
— Phone

3.3.3 dim_product

Catalogue produits enrichi :
— ProductKey, ProductName
— UnitPrice, CategoryName
— Discontinued (statut)

3.3.4 dim_employee

Personnel commercial :
— EmployeeKey, EmployeeFullName
— FirstName, LastName, Title
— City, Country

3.3.5 dim_shipper

Transporteurs :
— ShipperKey, ShipperName
— Phone

4 Implémentation de la chaîne ETL

4.1 Environnement technique

Technologies utilisées :
— **Langage :** Python 3.13
— **Bibliothèques :** pandas, pyodbc, pathlib, streamlit

— **Base de données :** SQL Server Express

Structure du projet :

— `scripts/etl_northwind.py` : ETL principal
— `scripts/dashboard_northwind.py` : Dashboard Streamlit
— `scripts/test_etl.py` : Tests de validation
— `data/excel/` : Sources Excel
— `data/processed/` : Entrepôt CSV

4.2 Connexion à SQL Server

La fonction de connexion encapsule l'accès à SQL Server :

```
def get_sql_connection():
    try:
        conn = pyodbc.connect(
            "DRIVER={ODBC Driver 17 for SQL Server};"
            "SERVER=localhost\\SQLEXPRESS;"
            "DATABASE=Northwind;"
            "Trusted_Connection=yes;"
        )
        print("Connexion SQL OK")
        return conn
    except Exception as e:
        print("Erreur connexion SQL:", e)
        return None
```

Listing 1 – Connexion SQL Server

4.3 Processus ETL

4.3.1 Extraction

Deux fonctions de chargement sont implémentées :

```
def load_excel(table):
    """Charge un fichier Excel si disponible (accepte nom avec ou
   sans underscores)."""
    candidates = [
        EXCEL_DIR / f"{table}.xlsx",
        EXCEL_DIR / f"{table.replace(' ', '_')}.xlsx",
    ]
    for file in candidates:
        if file.exists():
            df = pd.read_excel(file)
            df.columns = [c.replace(" ", "") for c in df.columns]
            print(f"   Excel {table:<15}: {df.shape[0]:4} lignes, {
   df.shape[1]} colonnes")
            return df
    return None
```

Listing 2 – Chargement des sources

4.3.2 Transformation

Les transformations principales incluent :

1. **Nettoyage des dates :** Conversion et suppression des dates invalides
2. **Gestion des catégories :** Jointure avec la table Categories, remplacement des valeurs manquantes par "Unknown"
3. **Fusion multi-sources :** Combinaison Excel + SQL avec suppression des doublons
4. **Construction des dimensions :** Sélection et renommage des colonnes pertinentes

```
dim_customer = (
    customers[["CustomerID", "CompanyName", "City",
               "Country", "Phone"]]
    .rename(columns={
        "CustomerID": "CustomerKey",
        "CompanyName": "CustomerName",
        "City": "CustomerCity",
        "Country": "CustomerCountry"
    })
    .dropna(subset=["CustomerKey"])
    .drop_duplicates(subset=["CustomerKey"])
)
```

Listing 3 – Construction de dim_customer

4.3.3 Chargement

Les tables sont exportées en CSV :

```
dim_time.to_csv(PROCESSED_DIR / "dim_time.csv", index=False)
dim_customer.to_csv(PROCESSED_DIR / "dim_customer.csv",
                    index=False)
dim_product.to_csv(PROCESSED_DIR / "dim_product.csv",
                   index=False)
fact_sales.to_csv(PROCESSED_DIR / "fact_sales.csv",
                  index=False)
```

Listing 4 – Export des tables

4.4 Validation des données

Un script de test vérifie :
— La présence de toutes les colonnes obligatoires
— La cohérence du nombre de lignes
— L'absence de valeurs critiques manquantes

FIGURE 3 – Exécution réussie du processus ETL

Résultats de validation :

Table	Nombre de lignes
dim_time	508
dim_customer	91
dim_product	77
dim_employee	9
dim_shipper	3
fact_sales	878

TABLE 2 – Volumétrie de l'entrepôt de données

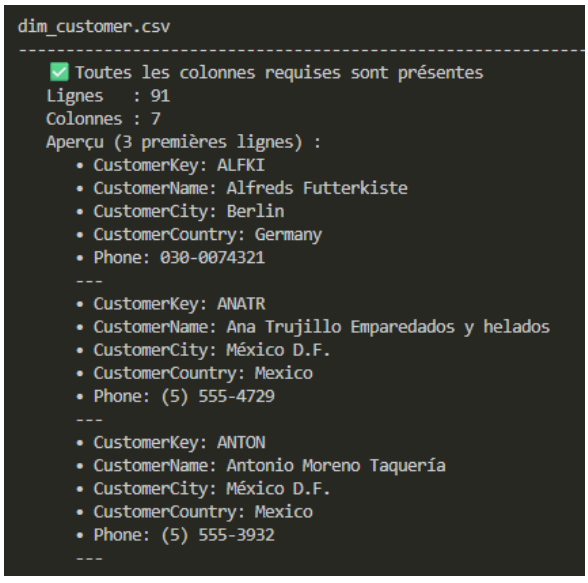

FIGURE 4 – Validation dim_customer

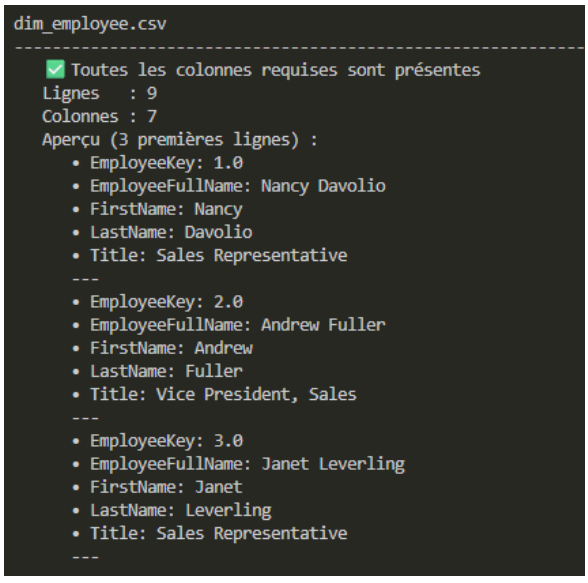

FIGURE 5 – Validation dim_employee

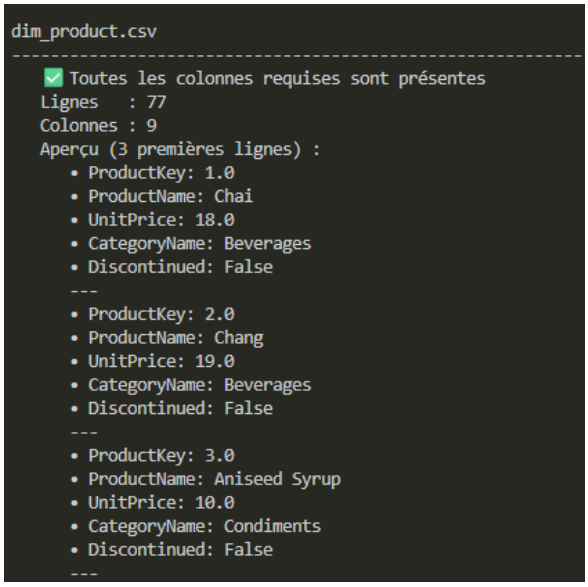

FIGURE 6 – Validation dim_product

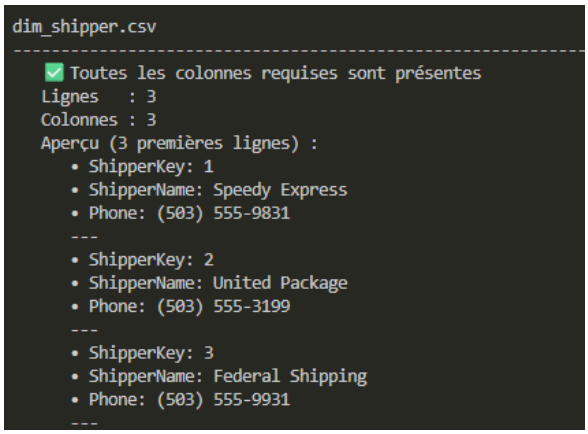

FIGURE 7 – Validation dim_shipper

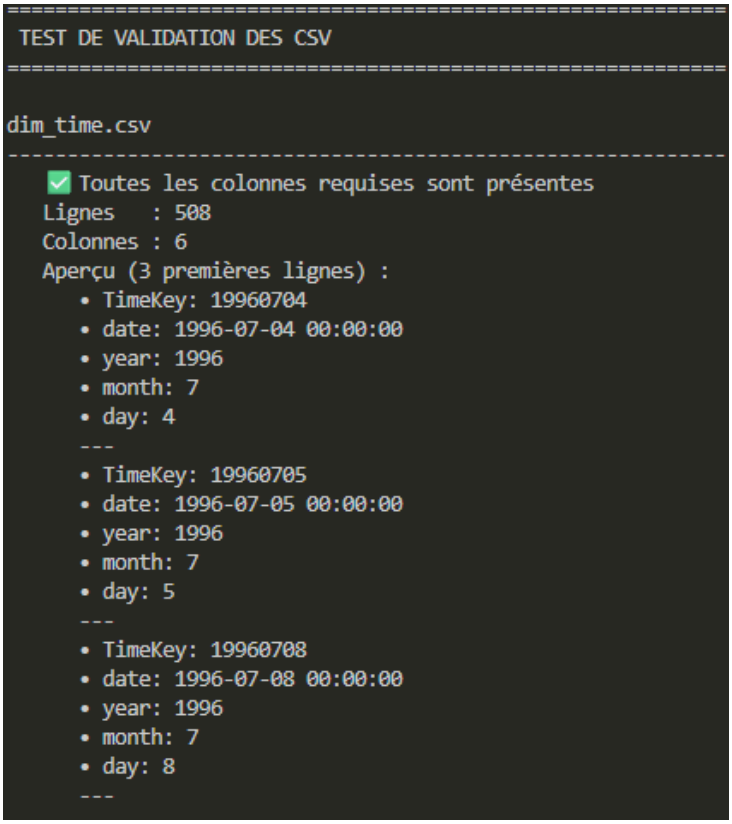

FIGURE 8 – Validation dim_time

5 Tableau de bord BI Streamlit

5.1 Objectifs du dashboard

Le tableau de bord développé avec Streamlit offre :
— Une vue synthétique de l'activité commerciale
— Des filtres interactifs (période, pays, commerciaux, transporteurs)
— Des visualisations dynamiques avec Plotly
— Un accès aux données détaillées

5.2 Indicateurs clés (KPI)

Les KPI affichés en temps réel :
— Chiffre d'affaires total
— Nombre de commandes
— Nombre de clients actifs
— Panier moyen

5.3 Visualisations interactives

Le dashboard propose plusieurs graphiques :

1. **Évolution mensuelle du CA** : Courbe temporelle montrant la tendance des ventes
2. **Top 10 pays** : Classement par chiffre d'affaires
3. **Meilleurs clients** : Top 10 des clients générateurs de revenus
4. **Performance commerciale** : Comparaison des commerciaux
5. **Répartition du fret** : Distribution par transporteur

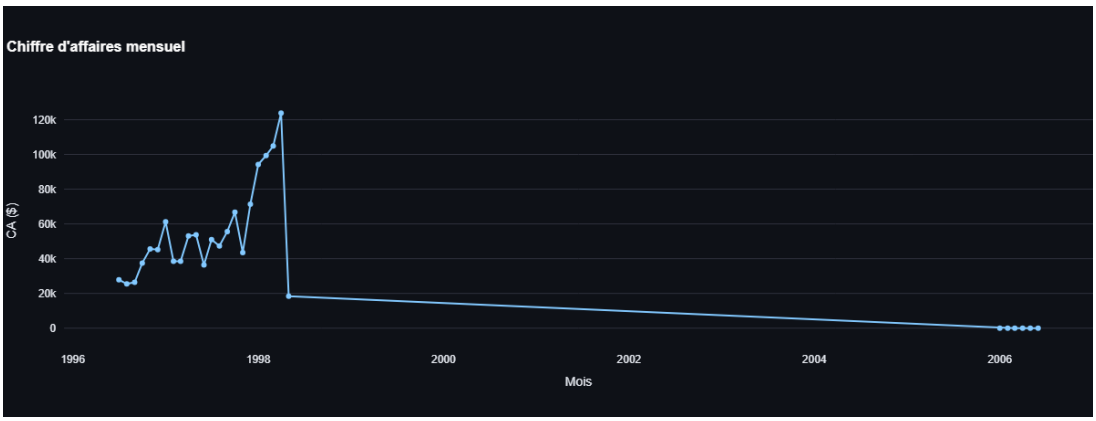

FIGURE 9 – Évolution du chiffre d'affaires mensuel

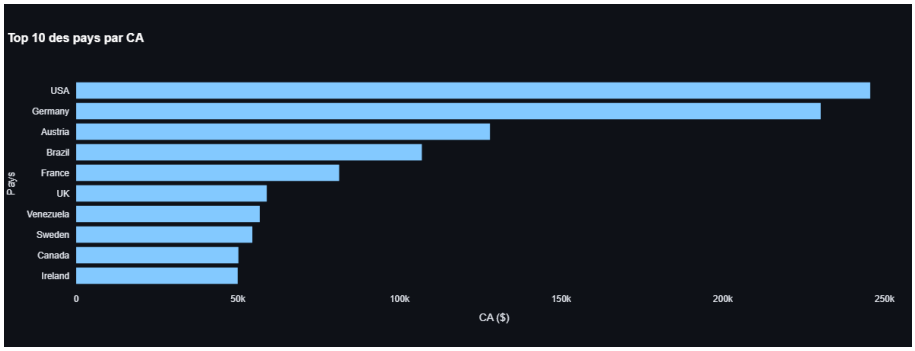

FIGURE 10 – Top 10 des pays par CA

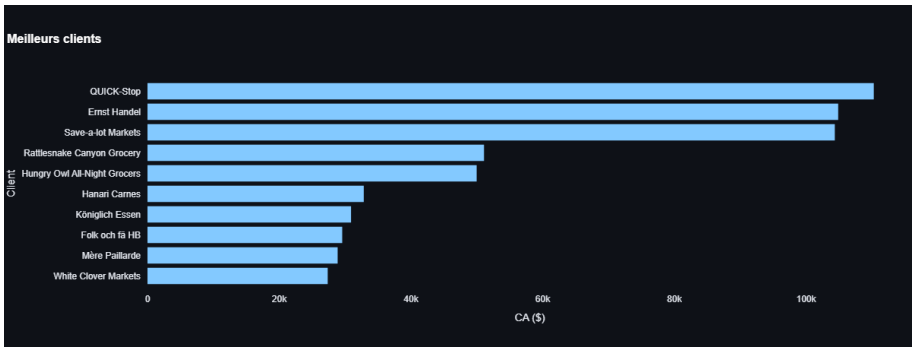

FIGURE 11 – Meilleurs clients

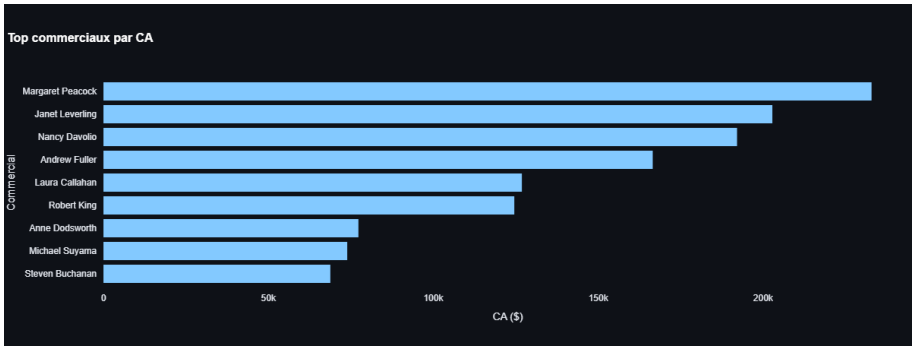

FIGURE 12 – Performance des commerciaux

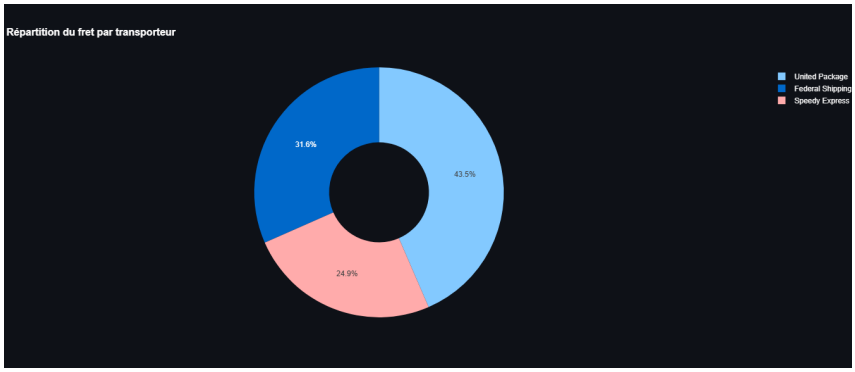

FIGURE 13 – Répartition du fret par transporteur

5.4 Filtres et interactivité

Le dashboard propose des filtres dans la barre latérale :
— **Période** : Sélection de plage de dates
— **Pays** : Multi-sélection des pays clients
— **Commerciaux** : Filtrage par employé
— **Transporteurs** : Sélection des shippers
Ces filtres s'appliquent dynamiquement à toutes les visualisations et KPI.

5.5 Export des données

Un bouton de téléchargement permet d'exporter les données filtrées au format CSV pour des analyses complémentaires.

6 Architecture technique

6.1 Flux de données

Le flux global du projet suit ce schéma :

1. **Sources** : SQL Server + Excel
2. **ETL Python** : Extraction, transformation, chargement
3. **Entrepôt CSV** : Tables dimensionnelles et de faits
4. **Dashboard Streamlit** : Visualisation et analyse

6.2 Avantages de l'architecture

— **Modularité** : Séparation claire ETL / Dashboard
— **Flexibilité** : Format CSV portable et réutilisable
— **Performance** : Agrégation dans l'ETL, lecture rapide dans le dashboard
— **Évolutivité** : Ajout facile de nouvelles dimensions ou mesures

7 Tests et validation

7.1 Validation de l'ETL

Le script `test_etl.py` vérifie systématiquement :
— L'existence de tous les fichiers CSV
— La présence des colonnes obligatoires
— La cohérence des types de données
— L'absence de valeurs NULL critiques

7.2 Qualité des données

Actions de nettoyage effectuées :
— Suppression d'une ligne avec OrderDate manquante
— Élimination d'un produit avec ProductID invalide
— Remplacement des CategoryName NULL par "Unknown"
— Conversion et validation des valeurs numériques

8 Conclusion

8.1 Réalisations

Ce projet a permis de :
— Implémenter une chaîne ETL complète en Python
— Construire un entrepôt de données structuré en schéma étoile
— Développer un tableau de bord interactif professionnel
— Valider la cohérence des données via des tests automatisés
— Maîtriser l'intégration de sources hétérogènes

8.2 Compétences acquises

— Conception de modèles dimensionnels
— Manipulation avancée de données avec pandas
— Connexion et requêtage SQL Server
— Développement d'interfaces web avec Streamlit
— Visualisation de données avec Plotly
— Bonnes pratiques ETL et documentation technique

En conclusion, ce projet représente une mise en pratique complète du cycle décisionnel : depuis les données sources opérationnelles jusqu'à l'analyse visuelle pour l'aide à la décision. Il démontre la maîtrise des concepts fondamentaux de la Business Intelligence et des outils modernes d'analyse de données.