



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	2
1.1	Contexte du projet	2
1.2	Objectifs	2
2	Sources de données	2
2.1	Vue d'ensemble	2
2.2	Source SQL Server	2
2.3	Sources Excel	3
3	Conception de l'entrepôt de données	3
3.1	Choix du schéma en étoile	3
3.2	Table de faits : fact_sales	4
3.3	Dimensions	5
3.3.1	dim_time	5
3.3.2	dim_customer	5
3.3.3	dim_product	5
3.3.4	dim_employee	5
3.3.5	dim_shipper	5
4	Implémentation de la chaîne ETL	5
4.1	Environnement technique	5
4.2	Connexion à SQL Server	6
4.3	Processus ETL	6
4.3.1	Extraction	6
4.3.2	Transformation	7
4.3.3	Chargement	7
4.4	Validation des données	7
5	Tableau de bord BI Streamlit	10
5.1	Objectifs du dashboard	10
5.2	Indicateurs clés (KPI)	10
5.3	Visualisations interactives	10
5.4	Filtres et interactivité	12
5.5	Export des données	12
6	Architecture technique	12
6.1	Flux de données	12
6.2	Avantages de l'architecture	12
7	Tests et validation	13
7.1	Validation de l'ETL	13
7.2	Qualité des données	13
8	Conclusion	13
8.1	Réalisations	13
8.2	Compétences acquises	13

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.

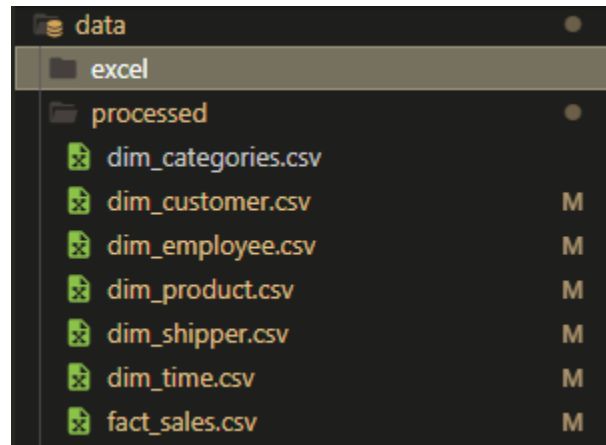


FIGURE 1 – Structure des fichiers CSV générés par l'ETL

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.

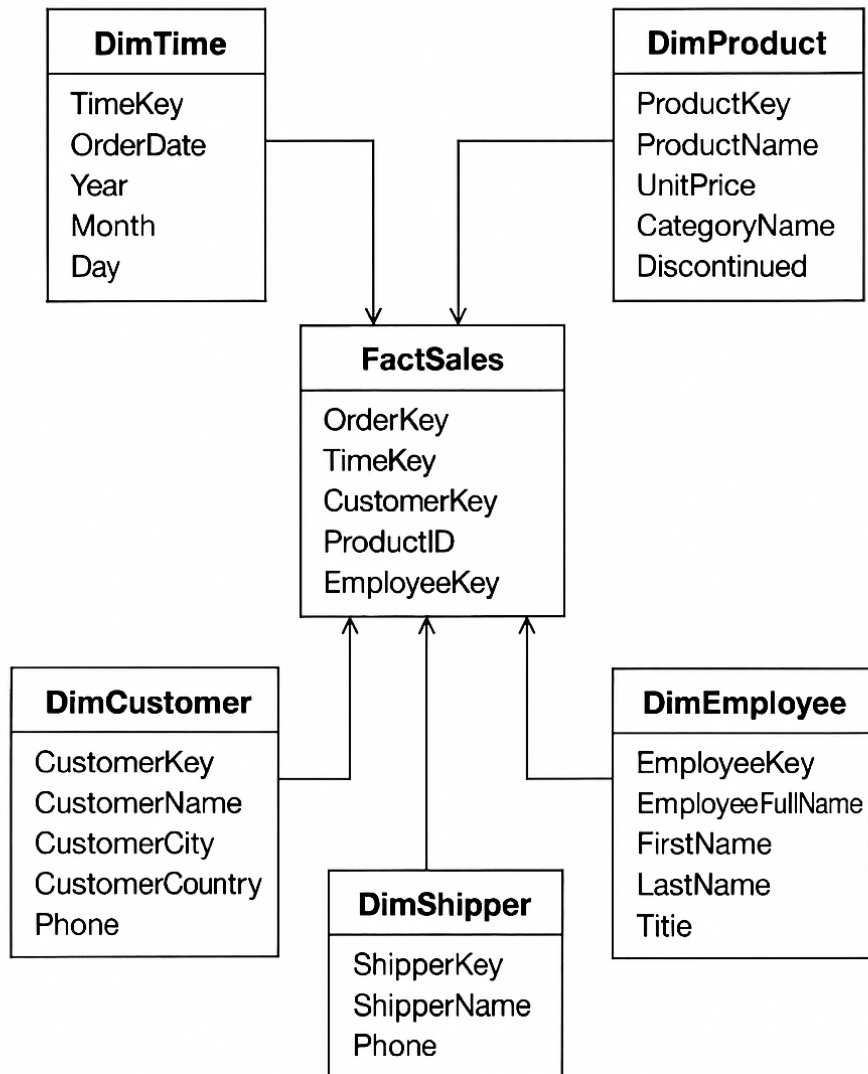


FIGURE 2 – Schéma en étoile de l'entrepôt Northwind

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 :

```

1 def get_sql_connection():
2     try:
3         conn = pyodbc.connect(
4             "DRIVER={ODBC Driver 17 for SQL Server};"
5             "SERVER=localhost\\SQLEXPRESS;"
6             "DATABASE=Northwind;"
7             "Trusted_Connection=yes;"
8         )
9         print("Connexion SQL OK")
10        return conn
11    except Exception as e:
12        print("Erreur connexion SQL:", e)
13        return None

```

Listing 1 – Connexion SQL Server

4.3 Processus ETL

4.3.1 Extraction

Deux fonctions de chargement sont implémentées :

```

1 def load_excel(table):
2     """Charge un fichier Excel si disponible (accepte nom avec ou
3     sans underscores)."""
4     candidates = [
5         EXCEL_DIR / f"{table}.xlsx",
6         EXCEL_DIR / f"{table.replace(' ', '_')}.xlsx",
7     ]
8     for file in candidates:
9         if file.exists():
10            df = pd.read_excel(file)
11            df.columns = [c.replace(" ", "") for c in df.columns]
12            print(f"    Excel {table[:15]:<15}: {df.shape[0]:4} lignes, {
13            df.shape[1]} colonnes")
14            return df
15        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

```
1 dim_customer = (  
2     customers[["CustomerID", "CompanyName", "City",  
3               "Country", "Phone"]]  
4     .rename(columns={  
5         "CustomerID": "CustomerKey",  
6         "CompanyName": "CustomerName",  
7         "City": "CustomerCity",  
8         "Country": "CustomerCountry"  
9     })  
10    .dropna(subset=["CustomerKey"])  
11    .drop_duplicates(subset=["CustomerKey"])  
12 )
```

Listing 3 – Construction de dim_customer

4.3.3 Chargement

Les tables sont exportées en CSV :

```
1 dim_time.to_csv(PROCESSED_DIR / "dim_time.csv", index=False)  
2 dim_customer.to_csv(PROCESSED_DIR / "dim_customer.csv",  
3                     index=False)  
4 dim_product.to_csv(PROCESSED_DIR / "dim_product.csv",  
5                    index=False)  
6 fact_sales.to_csv(PROCESSED_DIR / "fact_sales.csv",  
7                   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


```
(.venv) PS C:\Users\PC2000\OneDrive\Desktop\Northwind> python scripts\etl_northwind_sqlserver.py
>> python scripts\test_etl.py
>> streamlit run scripts\dashboard_northwind.py
? Connexion à SQL Server...
✓ Connecté.
? Chargement des tables source...
C:\Users\PC2000\OneDrive\Desktop\Northwind\scripts\etl_northwind_sqlserver.py:26: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or datab
ase string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider using SQLAlchemy.
  return pd.read_sql(query, conn)
? Construction DimTime...
? Construction DimCustomer...
? Construction DimProduct...
? Construction DimEmployee...
? Construction DimShipper...
? Construction FactSales...
? Sauvegarde des CSV dans: C:\Users\PC2000\OneDrive\Desktop\Northwind\data\processed
✓ ETL terminé.
```

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

```
dim_customer.csv
-----
✓ Toutes les colonnes requises sont présentes
Lignes : 91
Colonnes : 7
Aperçu (3 premières lignes) :
  • CustomerKey: ALFKI
  • CustomerName: Alfreds Futterkiste
  • CustomerCity: Berlin
  • CustomerCountry: Germany
  • Phone: 030-0074321
  ---
  • CustomerKey: ANATR
  • CustomerName: Ana Trujillo Emparedados y helados
  • CustomerCity: México D.F.
  • CustomerCountry: Mexico
  • Phone: (5) 555-4729
  ---
  • CustomerKey: ANTON
  • CustomerName: Antonio Moreno Taquería
  • CustomerCity: México D.F.
  • CustomerCountry: Mexico
  • Phone: (5) 555-3932
  ---
```

FIGURE 4 – Validation dim_customer

```
dim_employee.csv
-----
✓ Toutes les colonnes requises sont présentes
Lignes : 9
Colonnes : 7
Aperçu (3 premières lignes) :
  • EmployeeKey: 1.0
  • EmployeeFullName: Nancy Davolio
  • FirstName: Nancy
  • LastName: Davolio
  • Title: Sales Representative
  ---
  • EmployeeKey: 2.0
  • EmployeeFullName: Andrew Fuller
  • FirstName: Andrew
  • LastName: Fuller
  • Title: Vice President, Sales
  ---
  • EmployeeKey: 3.0
  • EmployeeFullName: Janet Leverling
  • FirstName: Janet
  • LastName: Leverling
  • Title: Sales Representative
  ---
```

FIGURE 5 – Validation dim_employee

```

dim_product.csv
-----
✅ Toutes les colonnes requises sont présentes
Lignes : 77
Colonnes : 9
Aperçu (3 premières lignes) :
  • ProductKey: 1.0
  • ProductName: Chai
  • UnitPrice: 18.0
  • CategoryName: Beverages
  • Discontinued: False
  ---
  • ProductKey: 2.0
  • ProductName: Chang
  • UnitPrice: 19.0
  • CategoryName: Beverages
  • Discontinued: False
  ---
  • ProductKey: 3.0
  • ProductName: Aniseed Syrup
  • UnitPrice: 10.0
  • CategoryName: Condiments
  • Discontinued: False
  ---

```

FIGURE 6 – Validation dim_product

```

dim_shipper.csv
-----
✅ Toutes les colonnes requises sont présentes
Lignes : 3
Colonnes : 3
Aperçu (3 premières lignes) :
  • ShipperKey: 1
  • ShipperName: Speedy Express
  • Phone: (503) 555-9831
  ---
  • ShipperKey: 2
  • ShipperName: United Package
  • Phone: (503) 555-3199
  ---
  • ShipperKey: 3
  • ShipperName: Federal Shipping
  • Phone: (503) 555-9931
  ---

```

FIGURE 7 – Validation dim_shipper

```

=====
TEST DE VALIDATION DES CSV
=====

dim_time.csv
-----
✅ Toutes les colonnes requises sont présentes
Lignes : 508
Colonnes : 6
Aperçu (3 premières lignes) :
  • TimeKey: 19960704
  • date: 1996-07-04 00:00:00
  • year: 1996
  • month: 7
  • day: 4
  ---
  • TimeKey: 19960705
  • date: 1996-07-05 00:00:00
  • year: 1996
  • month: 7
  • day: 5
  ---
  • TimeKey: 19960708
  • date: 1996-07-08 00:00:00
  • year: 1996
  • month: 7
  • day: 8
  ---

```

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

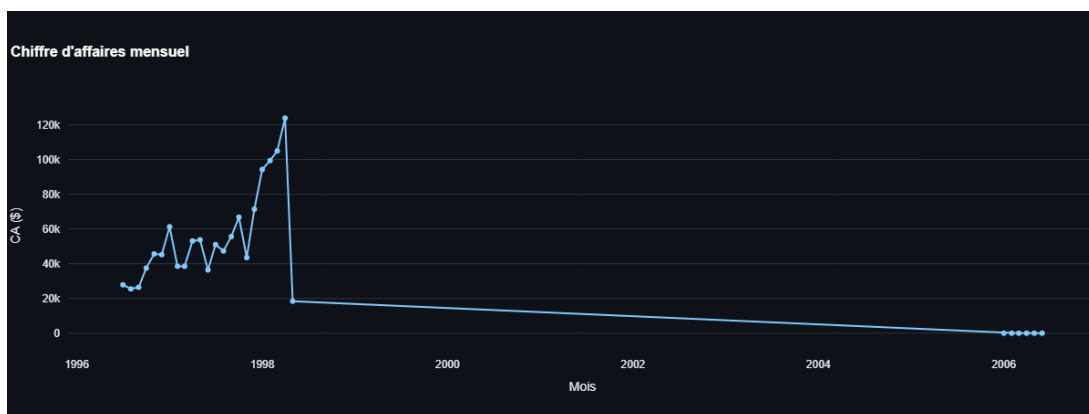


FIGURE 9 – Évolution du chiffre d'affaires mensuel

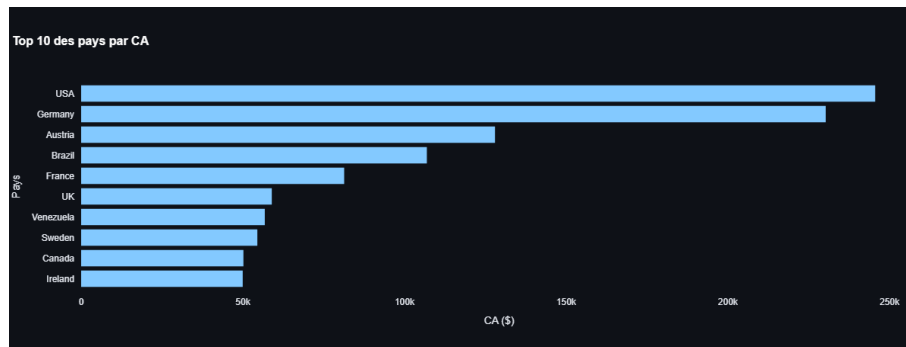


FIGURE 10 – Top 10 des pays par CA

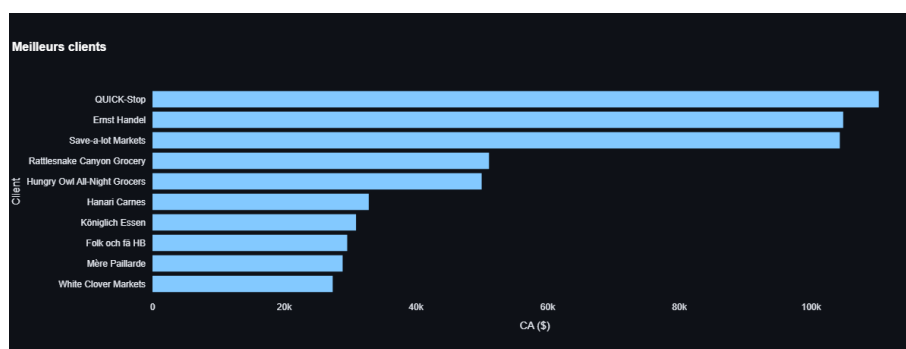


FIGURE 11 – Meilleurs clients

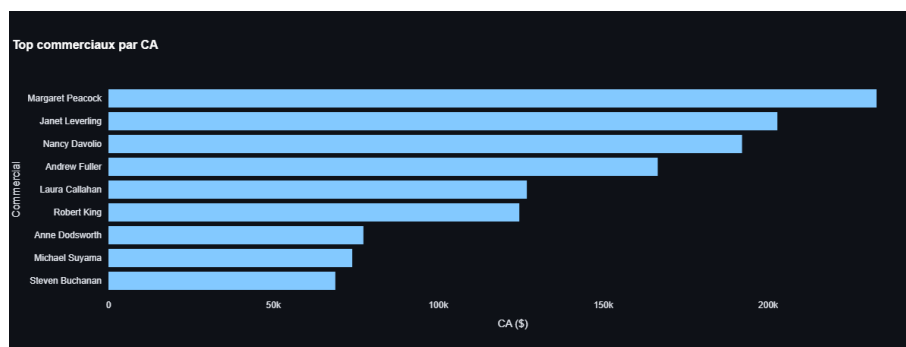


FIGURE 12 – Performance des commerciaux

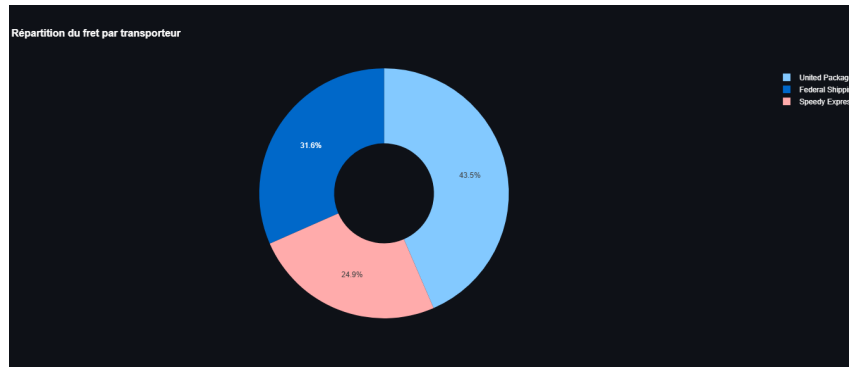


FIGURE 13 – Répartition du fret par transporteur

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.