

# TP Cloud : Pipeline Moderne

Modern Data Stack avec AWS, Snowflake et dbt

**Étudiant :** Aymane EL MKADMI

**Filière :** Data Analytics

**Date :** 6 janvier 2026

# 1. Introduction au Modern Data Stack

## 1.1 L'évolution vers le Cloud

Les organisations migrent massivement leurs infrastructures vers le cloud, bénéficiant ainsi d'une scalabilité élastique, d'une optimisation des coûts (pay-as-you-go), d'une maintenance réduite et d'un accès rapide aux innovations technologiques.

## 1.2 Architecture du Modern Data Stack

Le Modern Data Stack combine des outils cloud-natifs modulaires : PostgreSQL (source OLTP), Amazon S3 (Data Lake), Snowflake (Data Warehouse), dbt (transformations), Airflow (orchestration) et Power BI (visualisation).

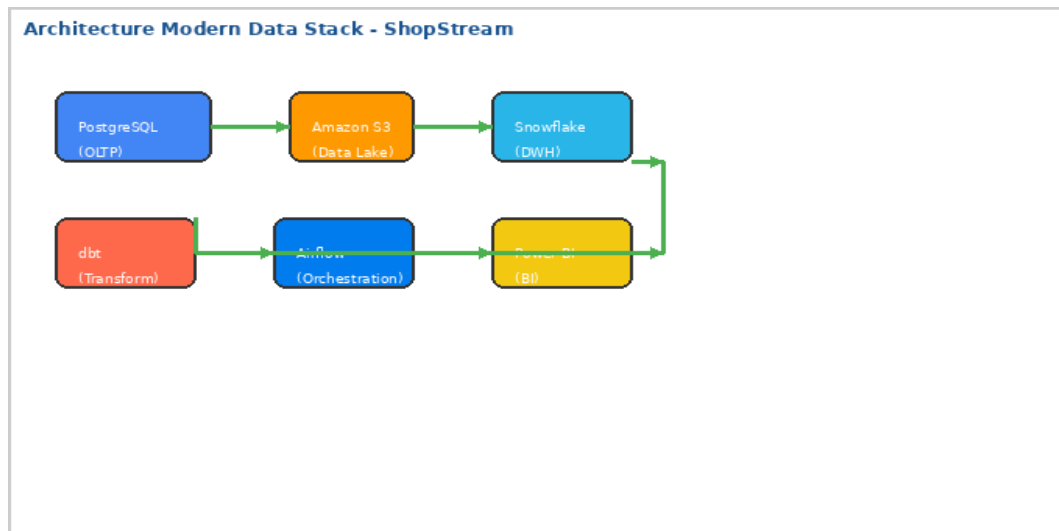


Figure 1 – Architecture du Modern Data Stack

## 1.3 Configuration de PostgreSQL

PostgreSQL sert de base de données transactionnelle source. La base shopstream est créée avec les tables : users, products, orders, order\_items, events et crm\_contacts.

**pgAdmin 4 - Création de la base shopstream**

**Nouvelle base de données**

Nom de la base : shopstream

Propriétaire : postgres

Créer

Figure 2 – Création de la base shopstream



Figure 3 – Tables OLTP créées

## 1.4 Génération et vérification des données

Le script `generate_data.py` génère automatiquement des données de test (utilisateurs, produits, commandes, événements). La vérification dans pgAdmin confirme l'insertion correcte des enregistrements.

```
Terminal - Génération des données

$ python generate_data.py

Génération des utilisateurs... ✓ 1000 users créés
Génération des produits... ✓ 200 products créés
Génération des commandes... ✓ 5000 orders créés
Génération des événements... ✓ 15000 events créés
Génération des contacts CRM... ✓ 800 contacts créés

✓ GENERATION TERMINEE AVEC SUCCES
```

Figure 4 – Génération des données

## pgAdmin - Vérification table users

```
SELECT * FROM users LIMIT 10;
```

id	full_name	email	country	created_at
1	Alice Martin	alice.m@mail.com	France	2024-01-15
2	Bob Smith	bob.smith@mail.com	USA	2024-01-16
3	Claire Dubois	c.dubois@mail.com	France	2024-01-17

Figure 5 – Vérification table users

## 1.5 Création du Data Lake sur AWS S3

Un bucket S3 (shopstream-datalake-aymane) est créé dans la région Europe (Paris) avec chiffrement activé. Une structure de dossiers hiérarchique organise les données brutes par source.

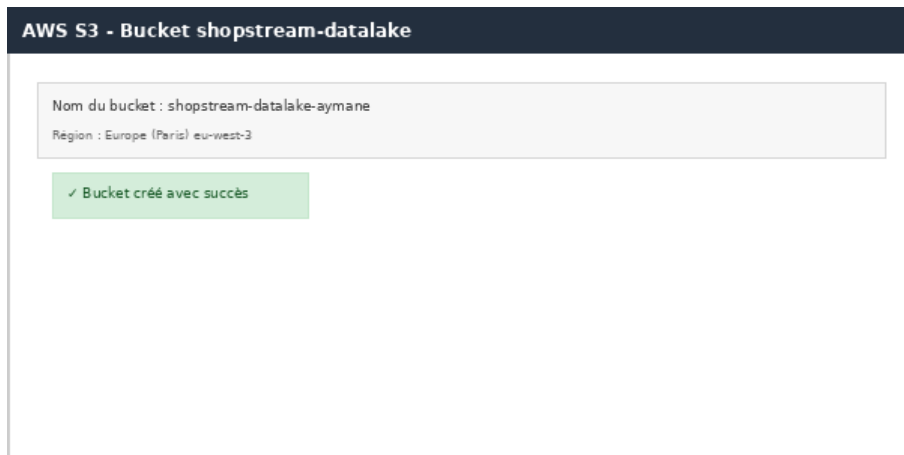


Figure 6 – Création du bucket S3



Figure 7 – Structure des dossiers S3

## 1.6 Configuration IAM et AWS CLI

Un utilisateur IAM (shopstream-s3-user) avec la stratégie AmazonS3FullAccess est créé. Les clés d'accès sont générées puis configurées via AWS CLI pour permettre l'accès programmatique au bucket.



Figure 8 – Utilisateur IAM et clés d'accès



Figure 9 – Configuration AWS CLI

## 1.7 Export PostgreSQL vers S3

Le script `export_to_s3.py` extrait les données de PostgreSQL et les charge au format CSV dans le bucket S3, préparant ainsi les données pour l'ingestion dans Snowflake.

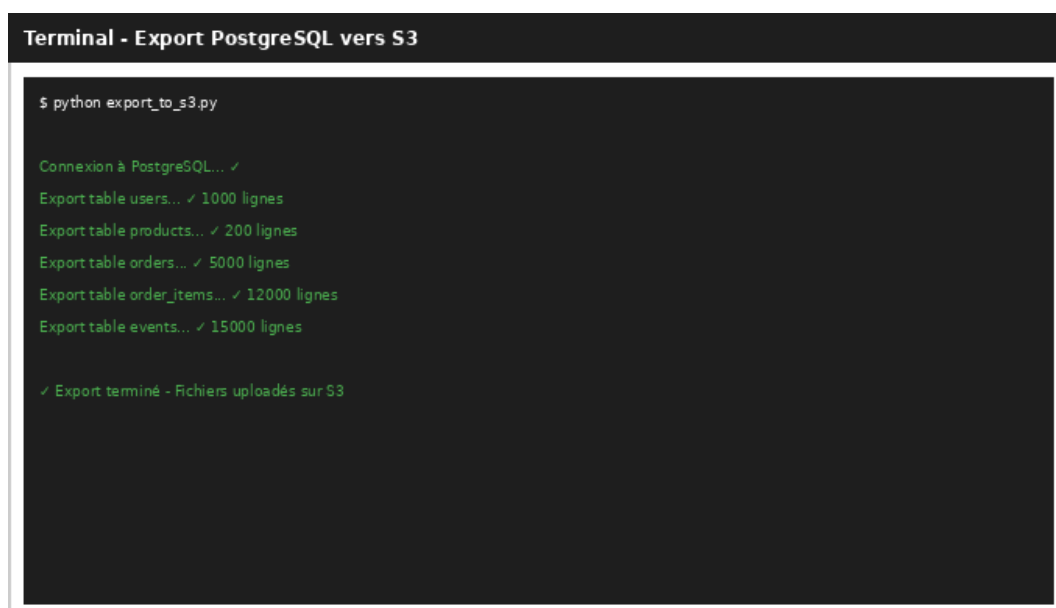


Figure 10 – Export vers S3

## 1.8 Configuration de Snowflake

Dans Snowflake, quatre schémas sont créés (RAW, STAGING, CORE, MARTS) ainsi que trois Virtual Warehouses (LOADING\_WH, TRANSFORM\_WH, BI\_WH) pour gérer les différentes charges de travail.

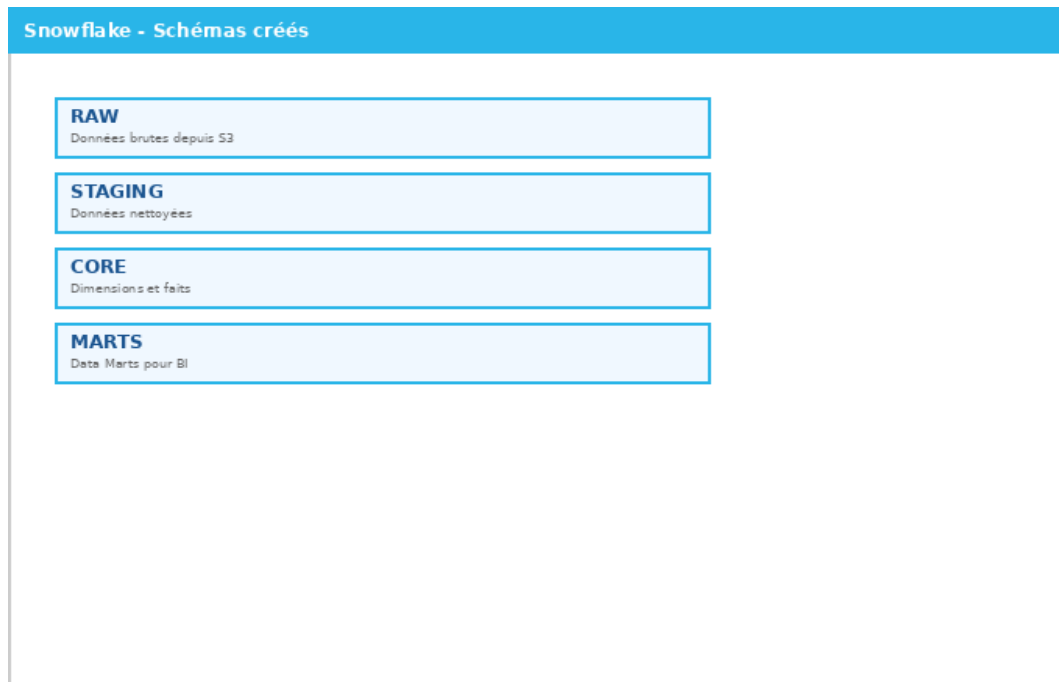


Figure 11 – Schémas Snowflake

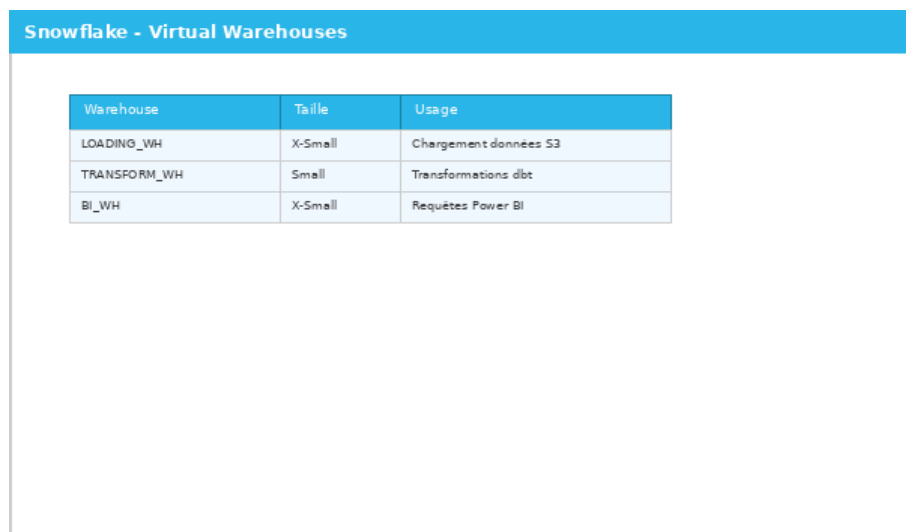


Figure 12 – Virtual Warehouses

### 1.8.1 Storage Integration et chargement des données

Une Storage Integration permet à Snowflake de se connecter au bucket S3. Un Stage externe pointe vers S3 et les tables STAGING sont créées puis chargées via la commande COPY INTO.

### Snowflake - Storage Integration avec S3

1. Etape A : Création d'un rôle IAM AWS pour Snowflake
2. Etape B : Création de la Storage Integration dans Snowflake
3. Etape C : Configuration de la relation de confiance dans AWS

Figure 13 – Storage Integration S3

### Snowflake - Stage externe vers S3

```
CREATE STAGE s3_stage
  STORAGE_INTEGRATION = s3_integration
  URL = 's3://shopstream-datalake/raw/';
```

✓ Stage créé avec succès

Figure 14 – Stage externe S3

### Snowflake - Tables STAGING

❑ stg\_users

❑ stg\_products

❑ stg\_orders

❑ stg\_order\_items

❑ stg\_events

❑ stg\_cm\_contacts

Figure 15 – Tables STAGING



## Snowflake - Chargement depuis S3

```
COPY INTO stg_users FROM @s3_stage/postgres/users/  
FILE_FORMAT = (TYPE = 'CSV' SKIP_HEADER = 1);
```

- ✓ 1000 lignes chargées dans stg\_users
- ✓ 200 lignes chargées dans stg\_products
- ✓ 5000 lignes chargées dans stg\_orders
- ✓ 12000 lignes chargées dans stg\_order\_items

Figure 16 – Chargement données S3

## 1.9 dbt : Transformations de données

dbt (data build tool) transforme les données STAGING en dimensions, faits et data marts optimisés pour l'analyse. Installation via pip, puis initialisation du projet et configuration de la connexion Snowflake.

```
Terminal - Installation de dbt

$ pip install dbt-core dbt-snowflake

Collecting dbt-core...
Collecting dbt-snowflake...
Successfully installed dbt-core-1.7.0 dbt-snowflake-1.7.0

$ dbt --version
installed version: 1.7.0
latest version: 1.7.0
✓ Up to date!
```

Figure 17 – Installation dbt

```
Terminal - Test connexion dbt

$ dbt debug

Running with dbt=1.7.0
dbt version: 1.7.0
python version: 3.11.0
Configuration:
  profiles.yml file [OK found and valid]
  dbt_project.yml file [OK found and valid]
Required dependencies:
  - snowflake-connector-python [OK found]
Connection test: [OK connection ok]

✓ All checks passed!
```

Figure 18 – Test connexion dbt

### 1.9.1 Exécution des modèles dbt

La commande dbt run compile et exécute tous les modèles dans l'ordre des dépendances, créant les tables stg\_\*, dim\_customers, dim\_products, fact\_orders et mart\_sales\_overview dans Snowflake.

```
Terminal - Exécution dbt run

$ dbt run

Running with dbt=1.7.0
Found 7 models, 12 tests, 0 snapshots

Compiling model stg_users
Compiling model stg_products
Compiling model dim_customers
Compiling model dim_products
Compiling model fact_orders
Compiling model mart_sales_overview

✓ Completed successfully

Done. PASS=7 WARN=0 ERROR=0 SKIP=0 TOTAL=7
```

Figure 19 – Exécution dbt run

### 1.9.2 Tests de qualité et documentation

dbt test vérifie la qualité des données (not\_null, unique, relations). dbt docs génère automatiquement la documentation complète du projet avec le lineage graph visualisant les dépendances entre modèles.

```
Terminal - Exécution dbt test

$ dbt test

Running with dbt=1.7.0
Found 7 models, 12 tests

PASS not_null_stg_users_id
PASS unique_stg_users_email
PASS not_null_stg_products_id
PASS relationships_orders_user_id
PASS accepted_values_status

✓ Completed successfully

Done. PASS=12 WARN=0 ERROR=0 SKIP=0 TOTAL=12
```

Figure 20 – Tests dbt

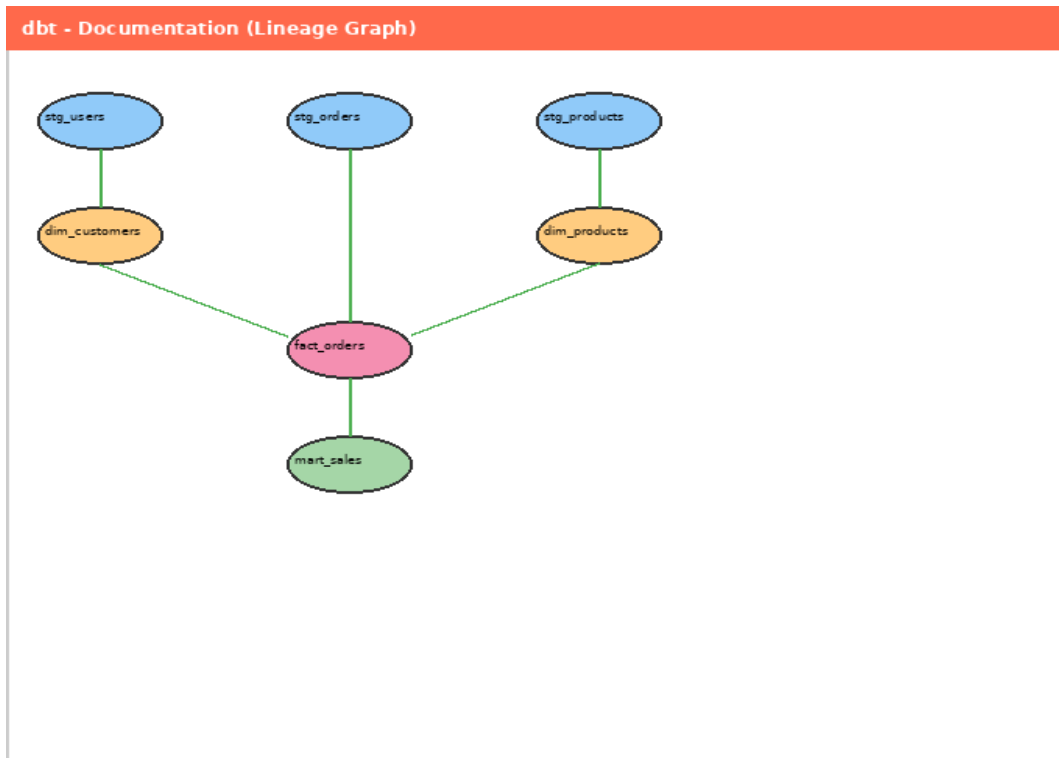


Figure 21 – Documentation dbt (lineage)

**Snowflake - Vérification mart\_sales\_overview**

```
SELECT * FROM MARTS.mart_sales_overview LIMIT 5;
```

sale_date	country	category	total_revenue	total_orders
2024-01-15	France	Electronics	15420.50	42
2024-01-15	USA	Clothing	8230.20	28
2024-01-16	France	Books	3450.80	15

Figure 22 – Vérification dans Snowflake

## 1.10 Création des Data Marts

Deux data marts supplémentaires sont créés : mart\_product\_performance (analyse ABC des produits) et mart\_customer\_ltv (Customer Lifetime Value avec segmentation RFM).

```
Terminal - dbt run mart_product_performance

$ dbt run --select mart_product_performance

Running with dbt=1.7.0
Found 1 model

Compiling model mart_product_performance
Starting execution...
Creating table MARTS.mart_product_performance

✓ Completed successfully in 2.4s

Done. PASS=1 WARN=0 ERROR=0 SKIP=0 TOTAL=1
```

Figure 23 – Data Mart Performance Produits

```
Terminal - dbt run mart_customer_ltv

$ dbt run --select mart_customer_ltv

Running with dbt=1.7.0
Found 1 model

Compiling model mart_customer_ltv
Starting execution...
Creating table MARTS.mart_customer_ltv
Calculating RFM scores...

✓ Completed successfully in 3.1s

Done. PASS=1 WARN=0 ERROR=0 SKIP=0 TOTAL=1
```

Figure 24 – Data Mart Customer LTV

## 1.11 Power BI : Visualisation et Dashboards

Power BI se connecte directement à Snowflake via le warehouse BI\_WH. Trois dashboards interactifs sont créés : Vue d'ensemble des ventes, Performance produits et Analyse clients.

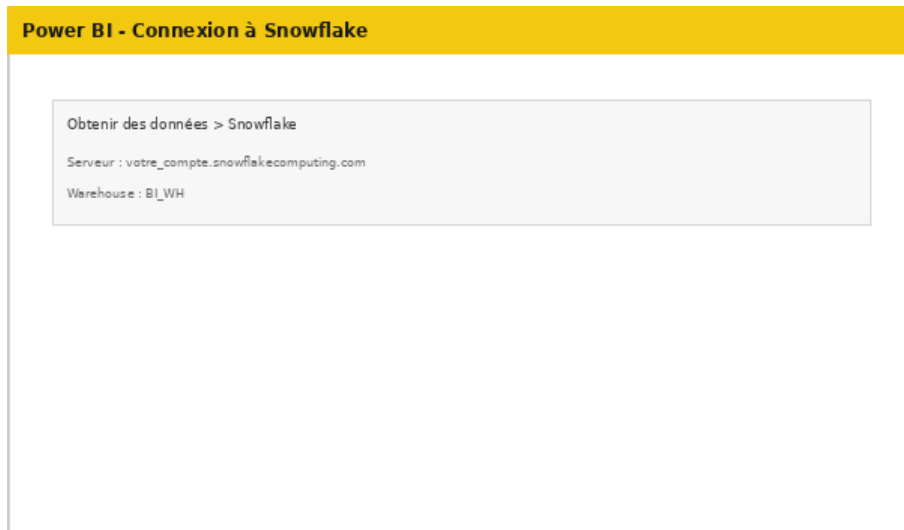


Figure 25 – Connexion Power BI à Snowflake

### 1.11.1 Dashboard Vue d'ensemble

Cartes KPI (chiffre d'affaires total, nombre de commandes), graphiques d'évolution temporelle du CA, répartition par pays et par catégorie de produits. Mesures DAX créées pour calculer le panier moyen et la croissance mensuelle.



Figure 26 – Carte KPI Chiffre d'affaires

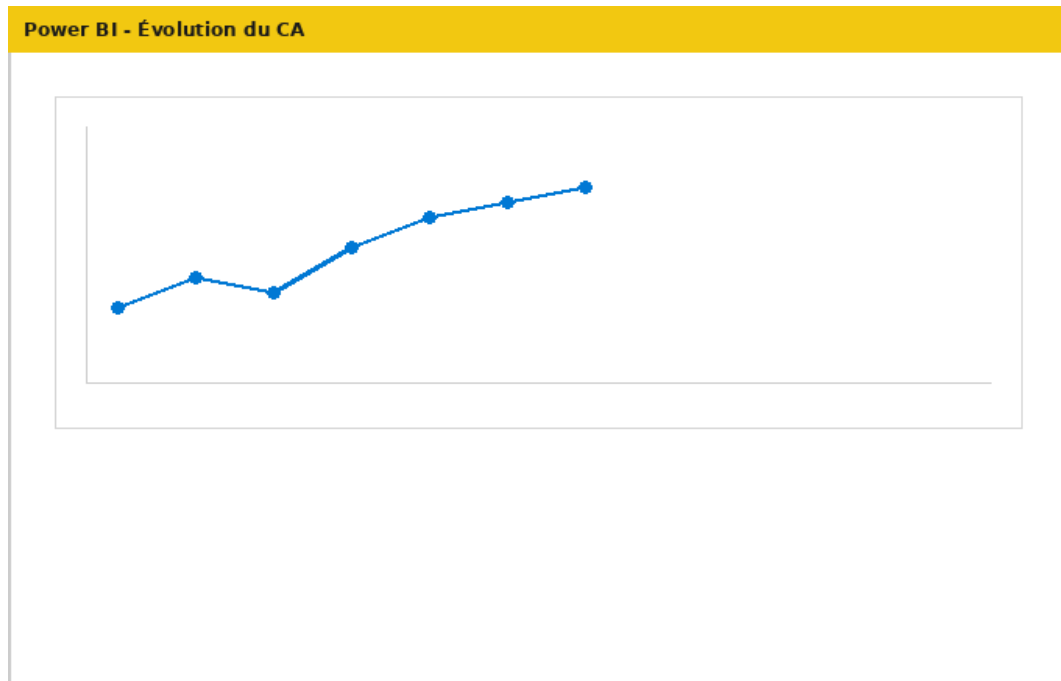


Figure 27 – Graphique évolution CA

### 1.11.2 Dashboards Produits et Clients

Le dashboard Performance Produits affiche un tableau détaillé avec classification ABC et tiers de performance. Le dashboard Analyse Clients présente la segmentation RFM, la valeur à vie par segment et un tableau des top clients.

**Power BI - Tableau Performance Produits**

Produit	Catégorie	Revenue	Commandes	ABC	Tier
Laptop Pro	Electronics	245,800 €	142	A	Top 10
Smartphone X	Electronics	198,400 €	256	A	Top 10
T-Shirt Blue	Clothing	45,200 €	892	B	Top 50

Figure 28 – Tableau Performance Produits

## 2. Synthèse et conclusions

Ce travail pratique a permis de mettre en œuvre un pipeline de données moderne de bout en bout, intégrant les meilleures pratiques de l'écosystème cloud actuel. Le projet démontre l'orchestration réussie de multiples technologies (PostgreSQL, AWS S3, Snowflake, dbt, Power BI) pour créer une architecture analytique scalable et performante.

Les compétences acquises incluent la création et gestion d'un Data Lake cloud, l'intégration de données depuis diverses sources, la modélisation dimensionnelle dans un Data Warehouse moderne, les transformations SQL déclaratives avec dbt, et la création de dashboards interactifs pour la prise de décision métier.

Cette approche modulaire et cloud-native représente l'état de l'art des architectures de données en entreprise et constitue une base solide pour l'évolution vers des cas d'usage plus complexes (streaming temps réel, Machine Learning, data science à grande échelle).