Travail Pratique

Data Warehouse Moderne

PySpark – PostgreSQL – SCD Type 2

Réalisé par : Hanae TALEBI
Année : 2025–2026

Table des matières

Chapitre 1

Introduction

Ce TP a pour objectif de mettre en place un Data Warehouse moderne en utilisant PostgreSQL et PySpark, avec implémentation du Slowly Changing Dimension de type 2 afin de conserver l'historique des données.

Chapitre 2

Installation de PostgreSQL

PostgreSQL est utilisé comme base de données source OLTP.

2.1 Téléchargement

`https://www.postgresql.org/download/`

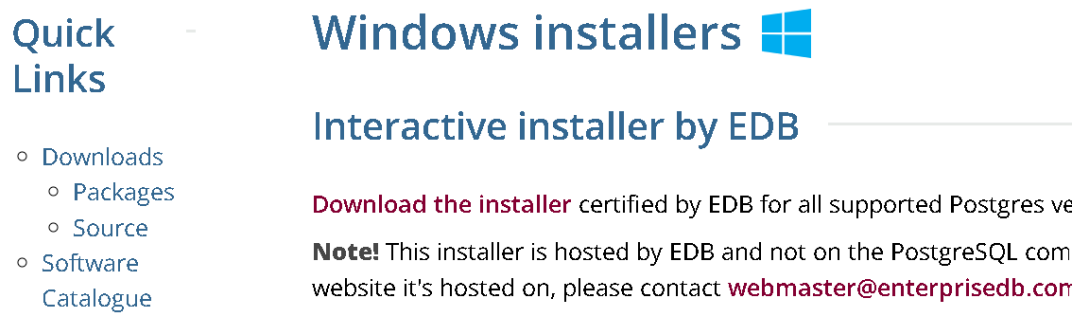


FIGURE 2.1 – Téléchargement de PostgreSQL

2.2 Installation et configuration

Choix du port 5432 et création du mot de passe pour l'utilisateur postgres.

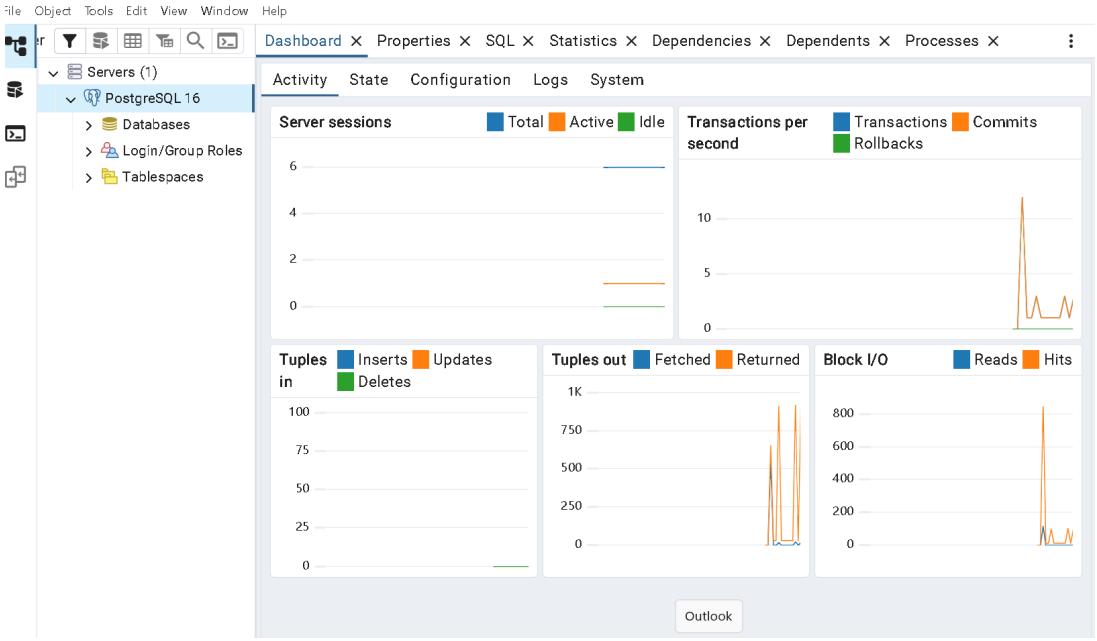


FIGURE 2.2 – Installation de PostgreSQL

Chapitre 3

Installation de Python

3.1 Téléchargement

`https://www.python.org/downloads/`

3.2 Installation

La case **Add Python to PATH** est cochée.

FIGURE 3.1 – Installation de Python

3.3 Vérification

```
python --version
```

FIGURE 3.2 – Vérification de Python

Chapitre 4

Création de l'environnement virtuel

```
python -m venv venv
venv\Scripts\activate
```

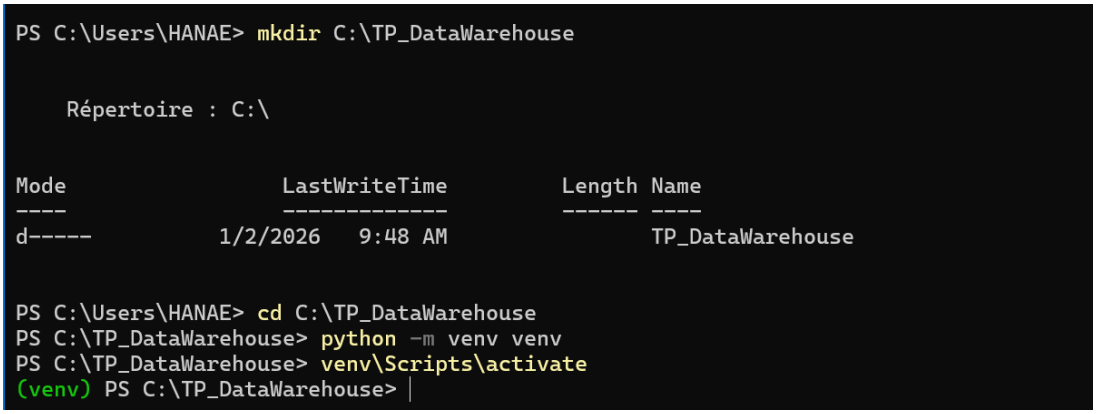


FIGURE 4.1 – Activation de l'environnement virtuel

Chapitre 5

Installation de PySpark et dépendances

```
pip install pyspark
pip install delta-spark
pip install psycopg2-binary
pip install pandas
```

```
(venv) PS C:\TP_DataWarehouse> python -c "import pyspark, psycopg2, pandas; from importlib.metadata import version; prin
t('pyspark', pyspark.__version__); print('delta-spark', version('delta-spark')); print('psycopg2', psycopg2.__version__)
; print('pandas', pandas.__version__)"
pyspark 3.5.0
delta-spark 3.0.0
psycopg2 2.9.9 (dt dec pq3 ext lo64)
pandas 2.1.4
(venv) PS C:\TP_DataWarehouse>
```

FIGURE 5.1 – Installation des bibliothèques

Chapitre 6

Téléchargement du driver JDBC

Le driver JDBC PostgreSQL est nécessaire pour permettre la connexion entre PySpark et PostgreSQL.

`https://jdbc.postgresql.org/download/`

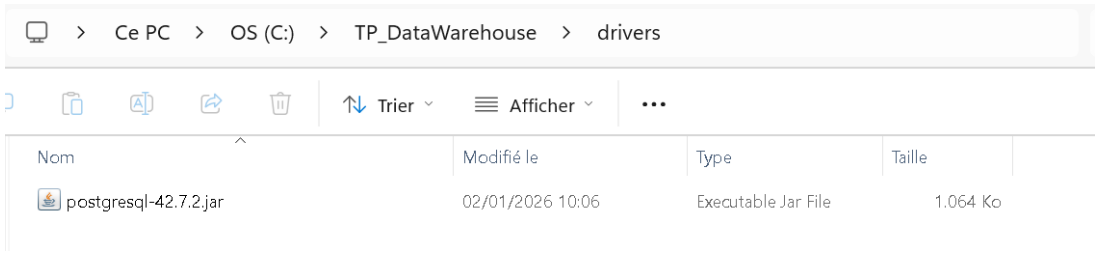


FIGURE 6.1 – Driver JDBC PostgreSQL

Chapitre 7

Connexion PySpark à PostgreSQL

Un script PySpark est utilisé pour tester la connexion JDBC.

```
(venv) PS C:\TP_DataWarehouse> .\venv\Scripts\python.exe test_connexion_pyspark.py
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Session Spark créée avec succès
Version de Spark : 3.4.1
Tentative de connexion à PostgreSQL...
Connexion réussie !
Nombre de lignes : 3

Schéma :
root
 |-- datname: string (nullable = true)
 |-- datdba: decimal(20,0) (nullable = true)
 |-- encoding: integer (nullable = true)


Aperçu des données :
+---------+------+--------+
|  datname|datdba|encoding|
+---------+------+--------+
| postgres|    10|       6|
|template1|    10|       6|
|template0|    10|       6|
+---------+------+--------+

Session Spark arrêtée
```

FIGURE 7.1 – Connexion PySpark à PostgreSQL

Chapitre 8

Création des tables sources

Les tables clients, produits et ventes sont créées dans PostgreSQL.

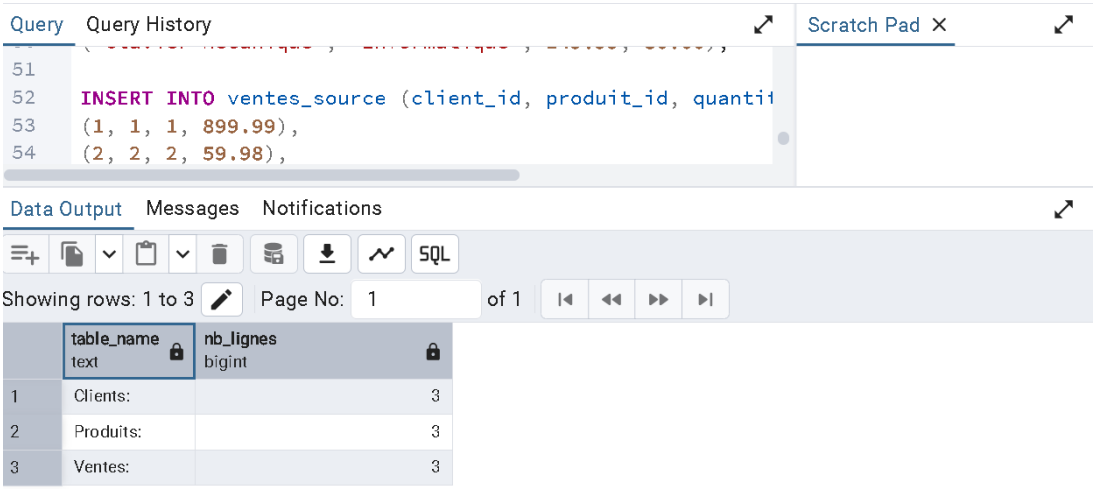


FIGURE 8.1 – Création des tables sources

Chapitre 9

Implémentation du SCD Type 2

Une table dimensionnelle est créée afin de gérer l'historique des clients.

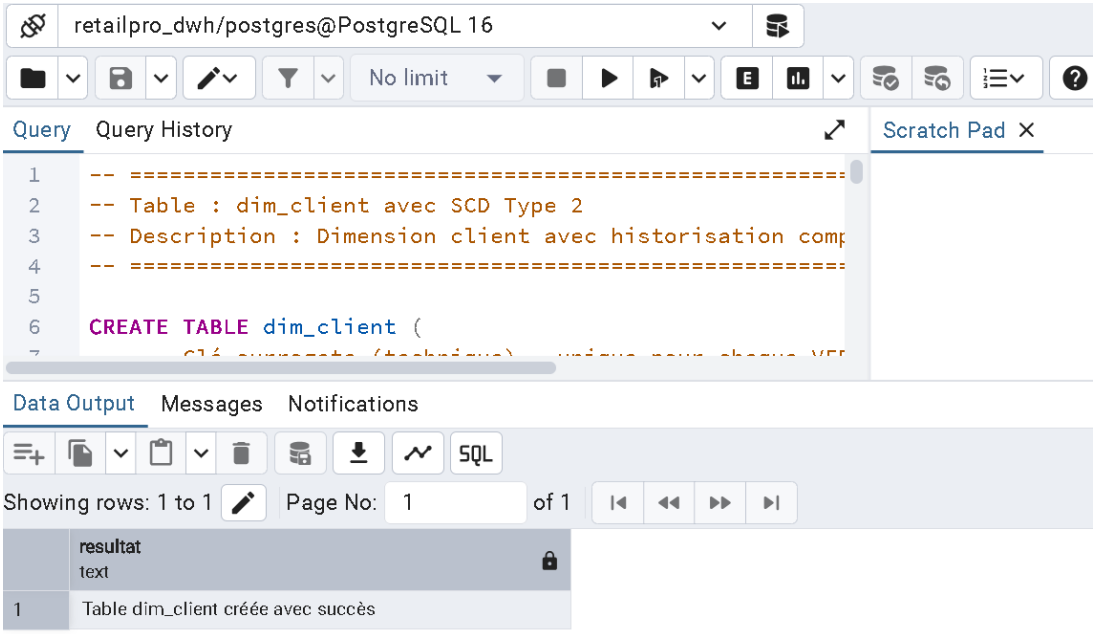


FIGURE 9.1 – Table dimension SCD Type 2

Chapitre 10

Chargement des données SCD Type 2

Un script Python permet de détecter les changements et créer de nouvelles versions.

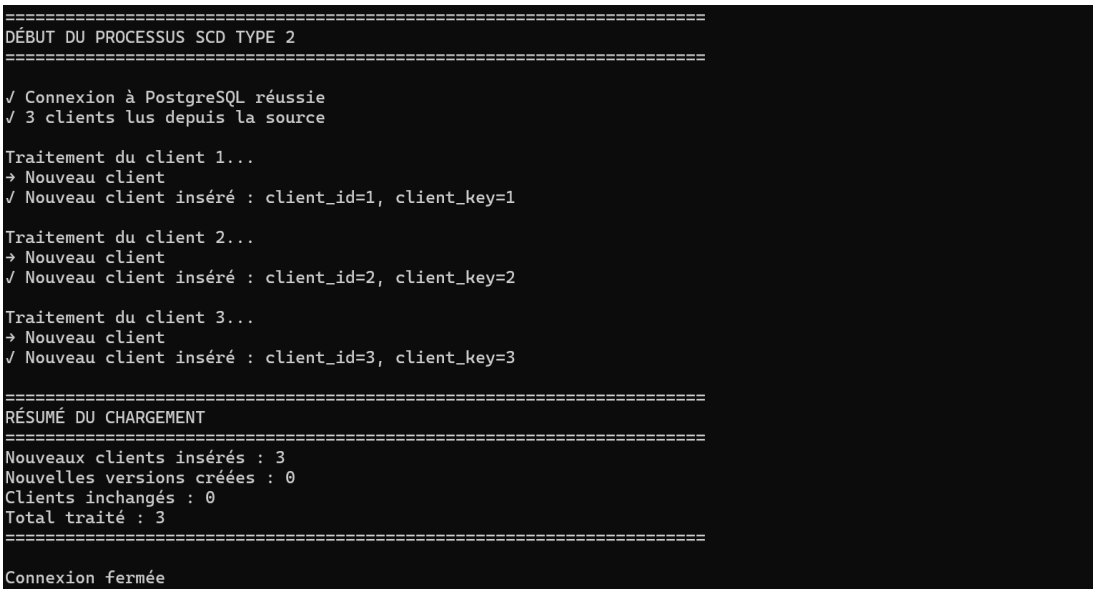


FIGURE 10.1 – Chargement SCD Type 2

Chapitre 11

Test avec modification des données

Des modifications sont effectuées dans la table source afin de vérifier l'historisation.

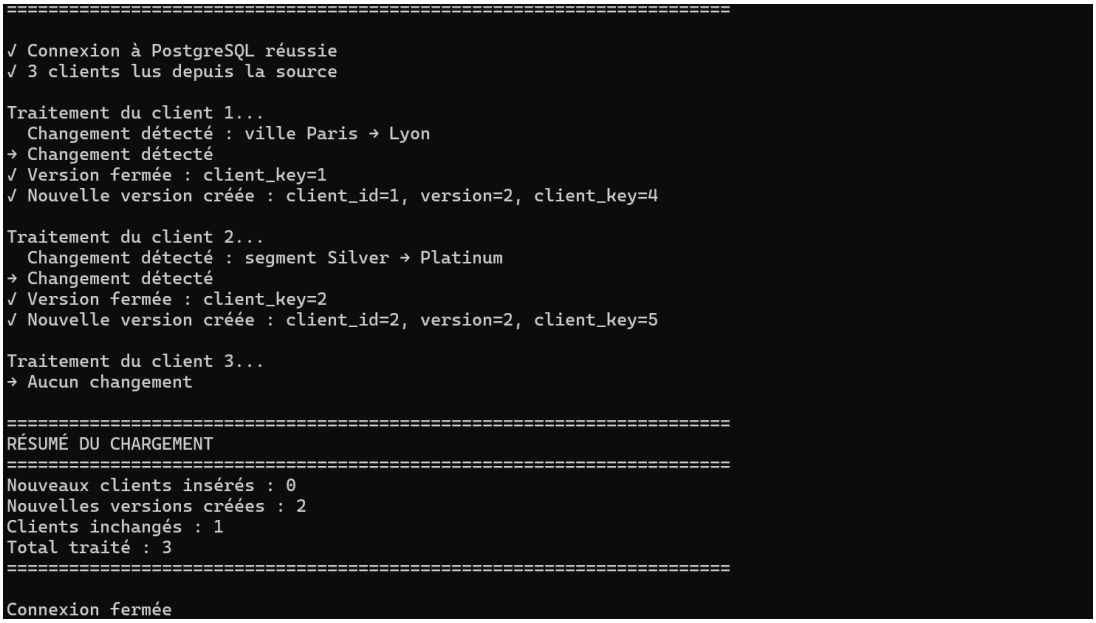


FIGURE 11.1 – Test de modification des données

Chapitre 12

Data Vault 2.0 - Modélisation Agile

12.1 Créer les tables Data Vault

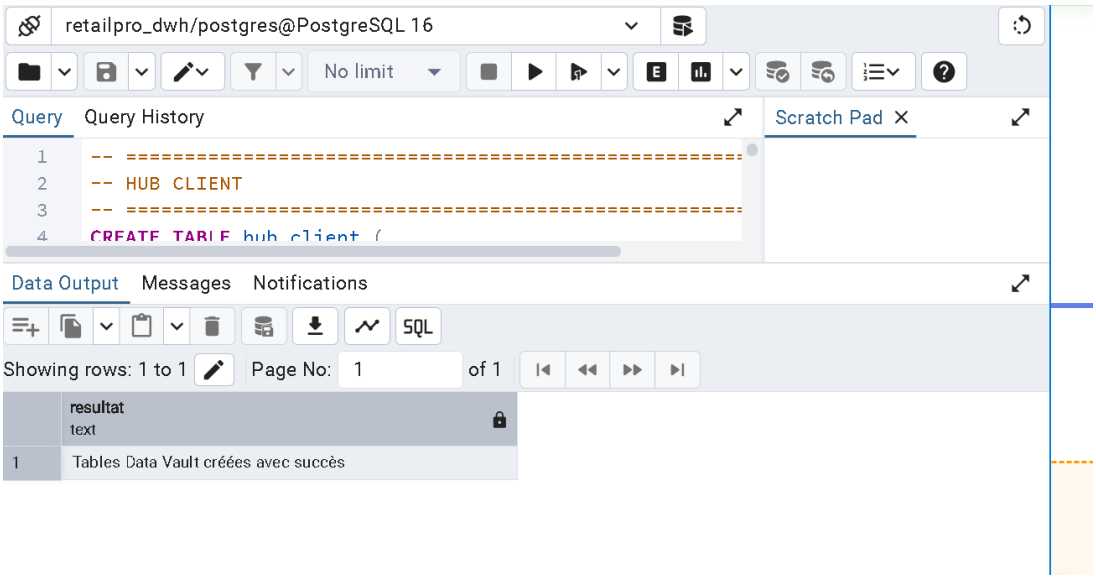


FIGURE 12.1 – Création des tables Data Vault

Chapitre 13

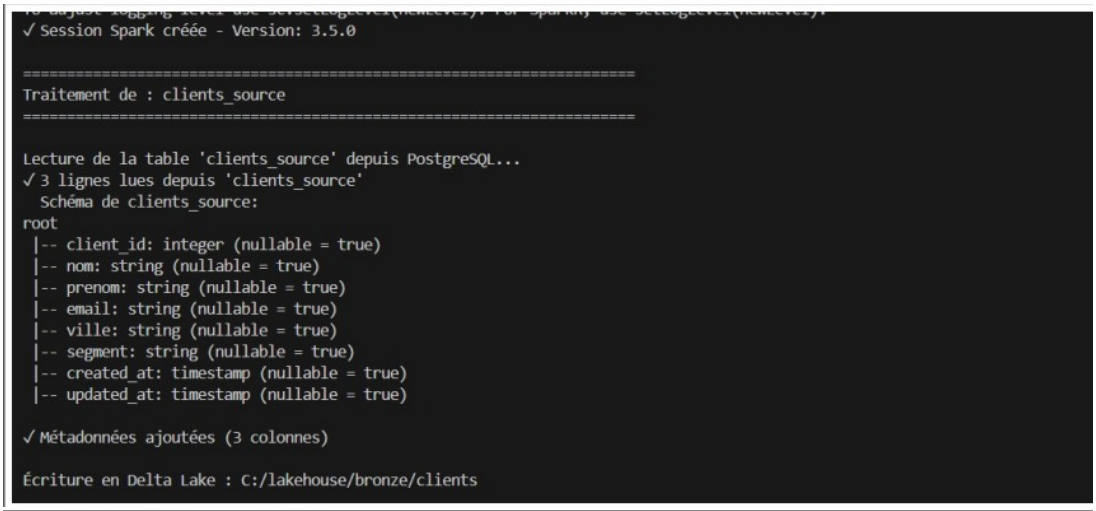
Architecture Lakehouse avec PySpark et Delta Lake

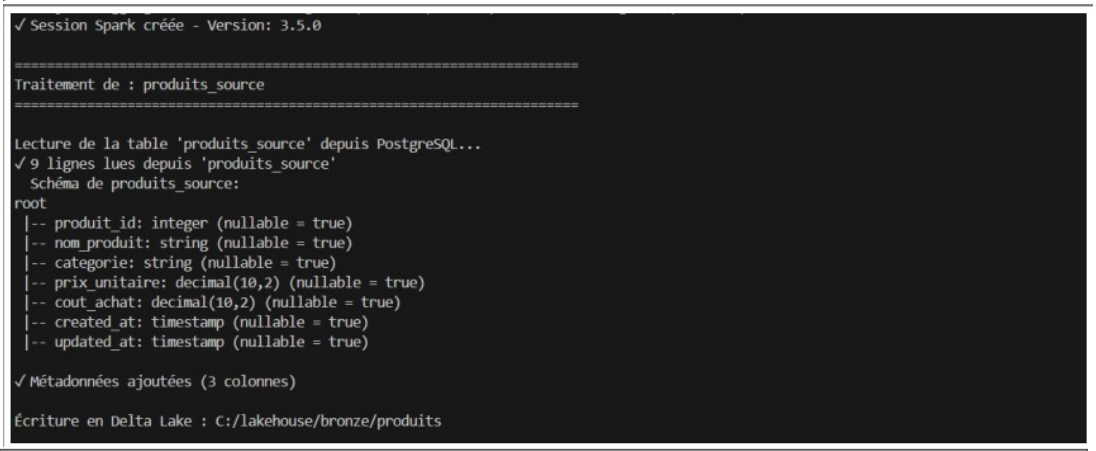
13.1 Configuration de l'Environnement Lakehouse

```
(venv) PS C:\TP_DataWarehouse> mkdir C:\lakehouse

    Répertoire : C:\

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          1/2/2026   1:38 PM                lakehouse

(venv) PS C:\TP_DataWarehouse> mkdir C:\lakehouse\bronze

    Répertoire : C:\lakehouse

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----          1/2/2026   1:38 PM                bronze

(venv) PS C:\TP_DataWarehouse> mkdir C:\lakehouse\silver

    Répertoire : C:\lakehouse

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
```

FIGURE 13.1 – Configuration de l'environnement Lakehouse

13.2 BRONZE : Ingestion depuis PostgreSQL - Expliqué ligne par ligne





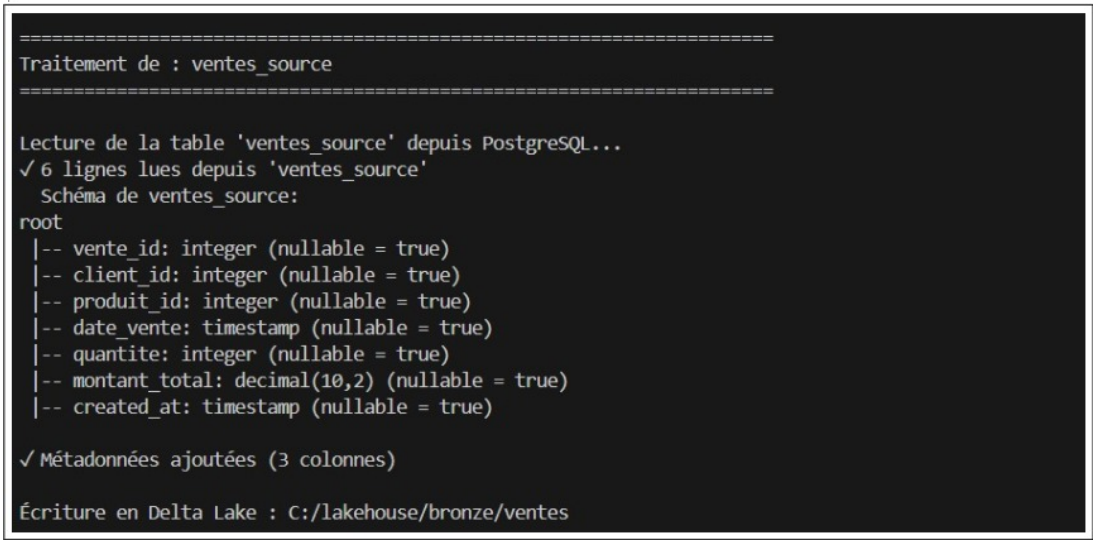


FIGURE 13.2 – Ingestion des données Bronze depuis PostgreSQL

Chapitre 14

Projet Final - Intégration Complète

14.1 Pipeline Complet

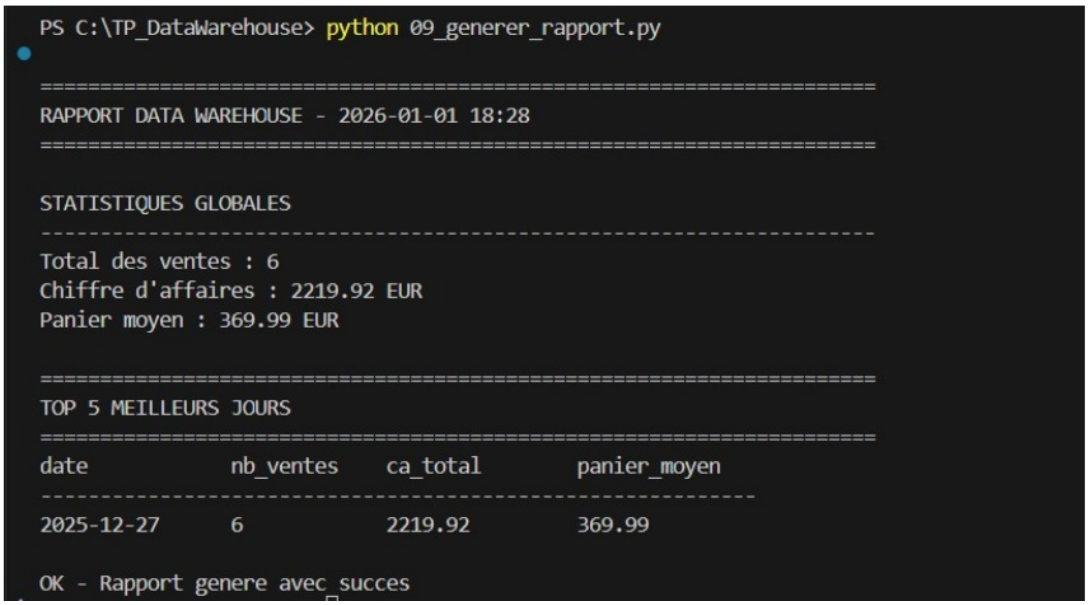


FIGURE 14.1 – Pipeline complet d'intégration

Chapitre 15

Conclusion

Ce TP a permis de comprendre la mise en place d'un Data Warehouse moderne et l'importance du SCD Type 2 pour conserver l'historique des données.