**REPUBLIQUE DU SENEGAL**

Un Peuple, Un But, Une foi





Rigueur et persévérance

Data Engineering

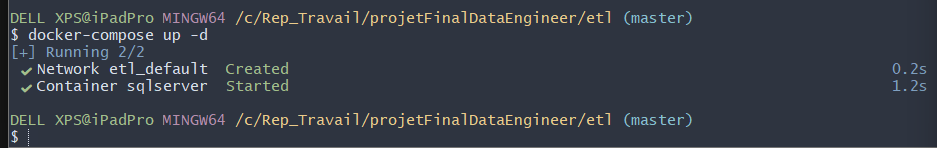
**Final Projet**

**Spark-ETL-Power BI**

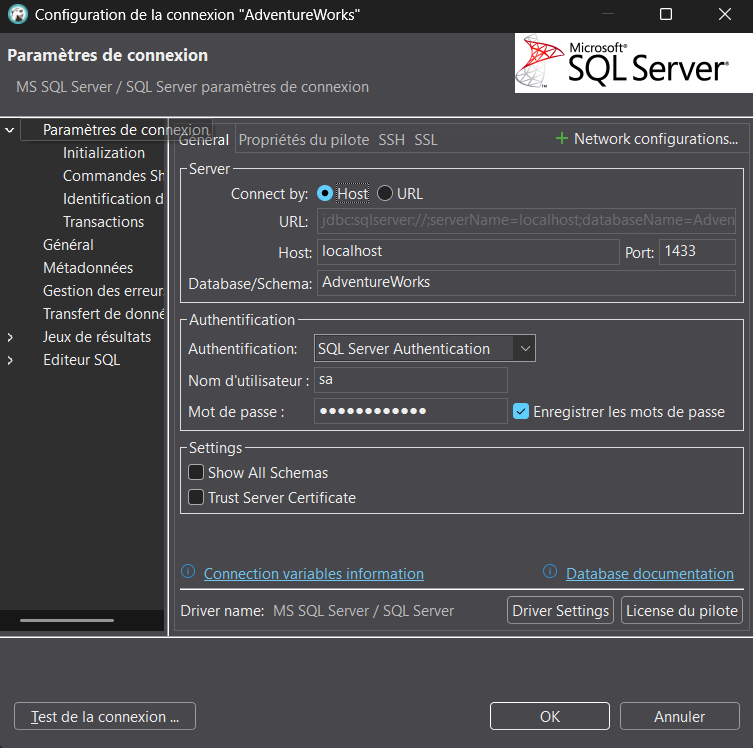
|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Auteur | Commentaires |
| 0.1 | 15.06.2025 | Mouhamadou Mourtala MBOW | Version initiale |

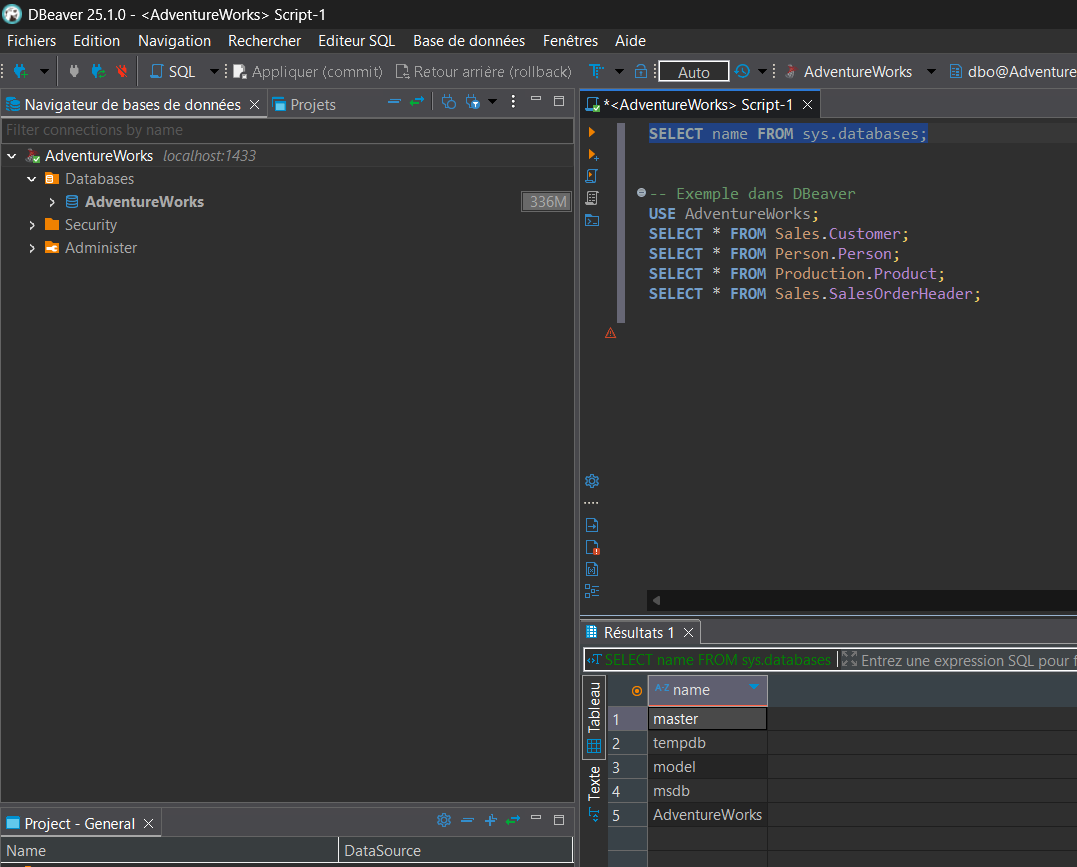
1. **Etape 1 : Préparation de l’environnement :**
2. Cloner le dépôt GitHub et lancer le conteneur MSSQL :

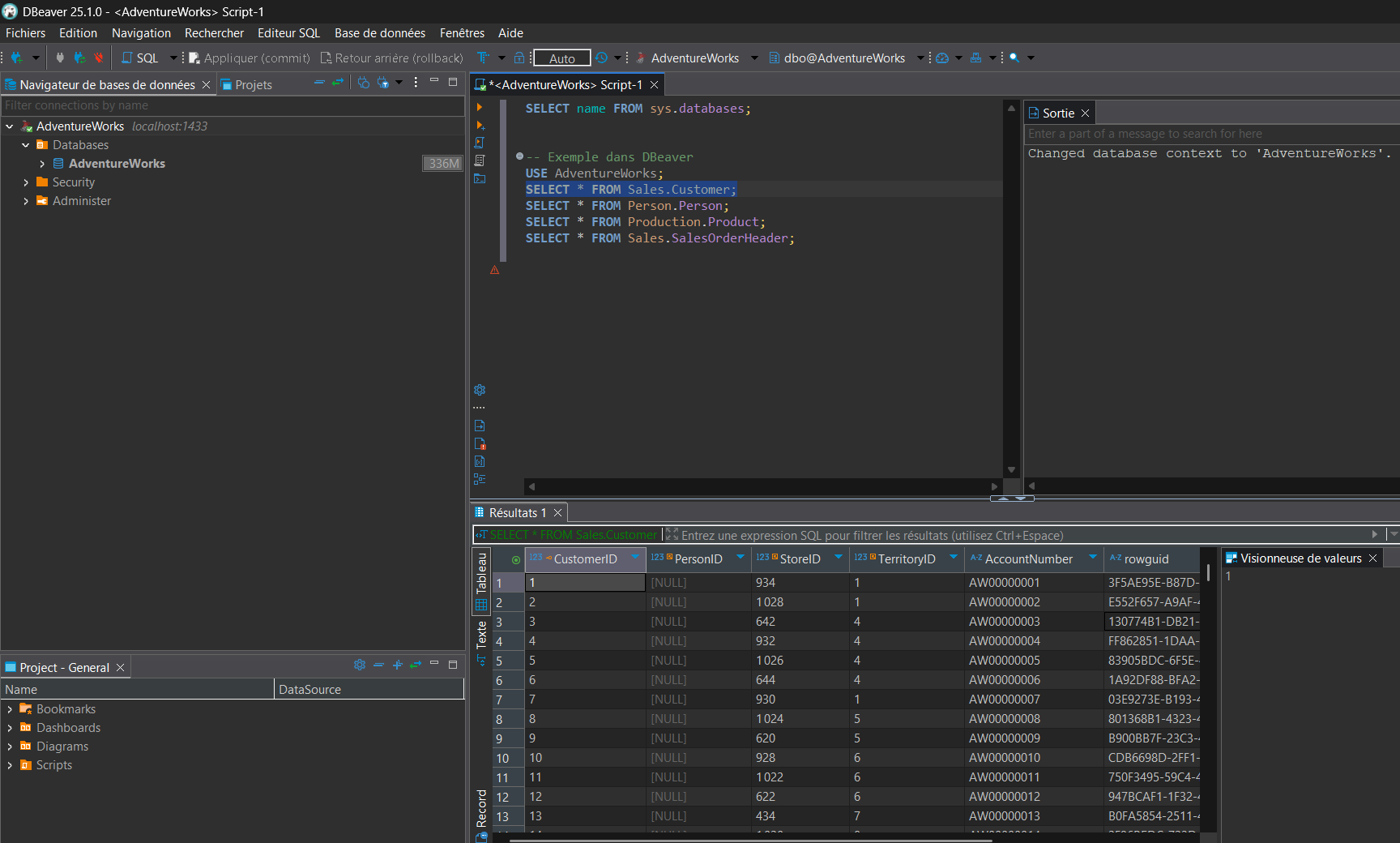
* git clone <https://github.com/sopeKhadim/etl.git>
* cd etl
* docker-compose up -d



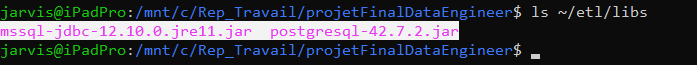
* DBeaver : Pour visualiser la base AdventureWorks
* **Configuration dans DBeaver :**
* **Type de base** : SQL Server
* **Host** : localhost
* **Port** : 1433
* **Database** : AdventureWorks (ou master au début)
* **User** : sa
* **Password** : password123?

****

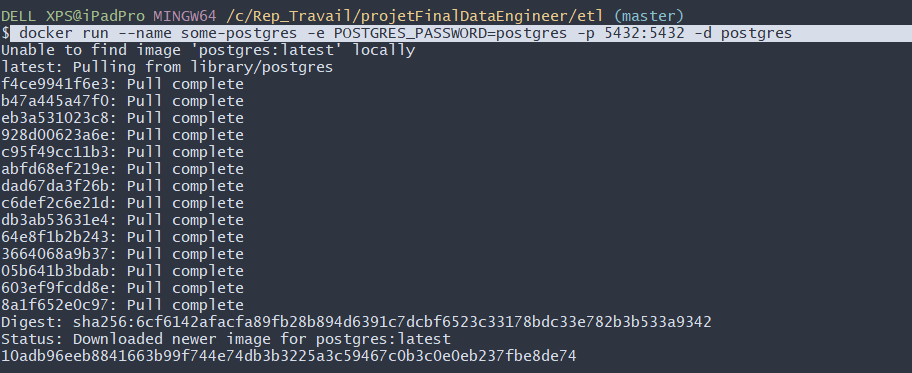
****

****

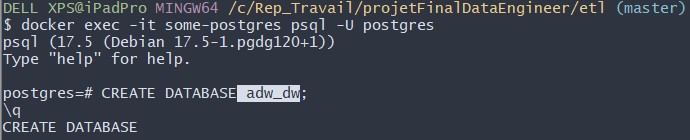
* Téléchargeons et Copions les drivers **mssql-jdbc-12.10.0.jre11.jar et postgresql-42.7.2.jar** dans ton dossier etl/libs



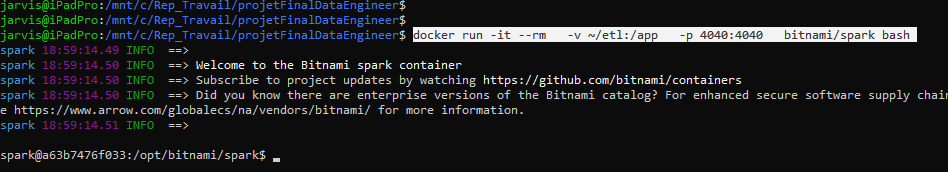
* Lançons le conteneur **PostgreSQL**
* docker run --name some-postgres -e POSTGRES\_PASSWORD=postgres -p 5432:5432 -d postgres



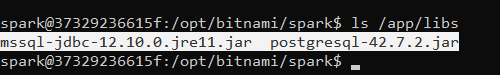
* Créons une base de données **PostgreSQL** nommée **adw\_dw** en se connectant au conteneur pour y charger les données transformées :



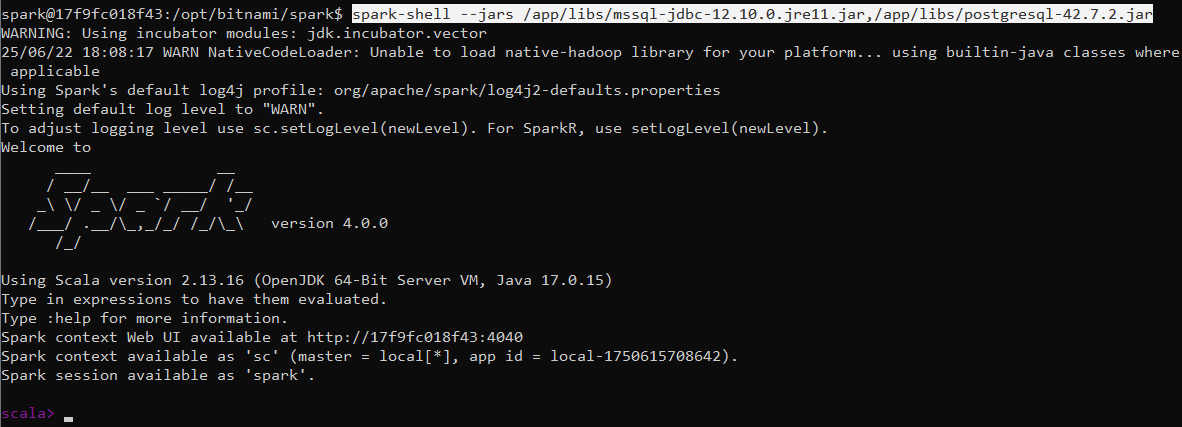
* Lançons un conteneur **Spark** avec montage du volume
* docker run -it --rm -v ~/etl:/app -p 4040:4040 bitnami/spark bash



* Vérifions dans le conteneur que les JAR sont bien là :
* ls /app/libs

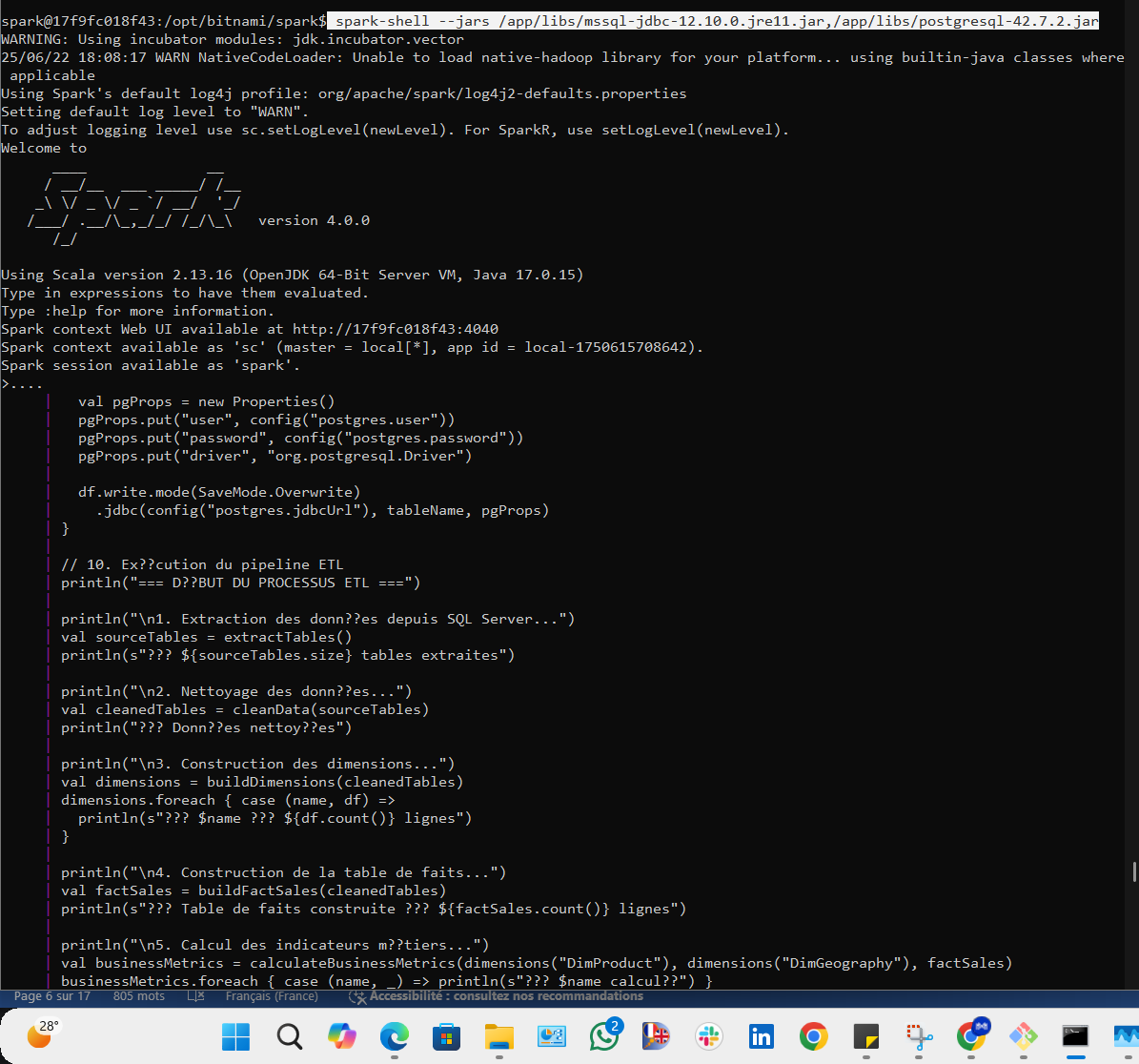


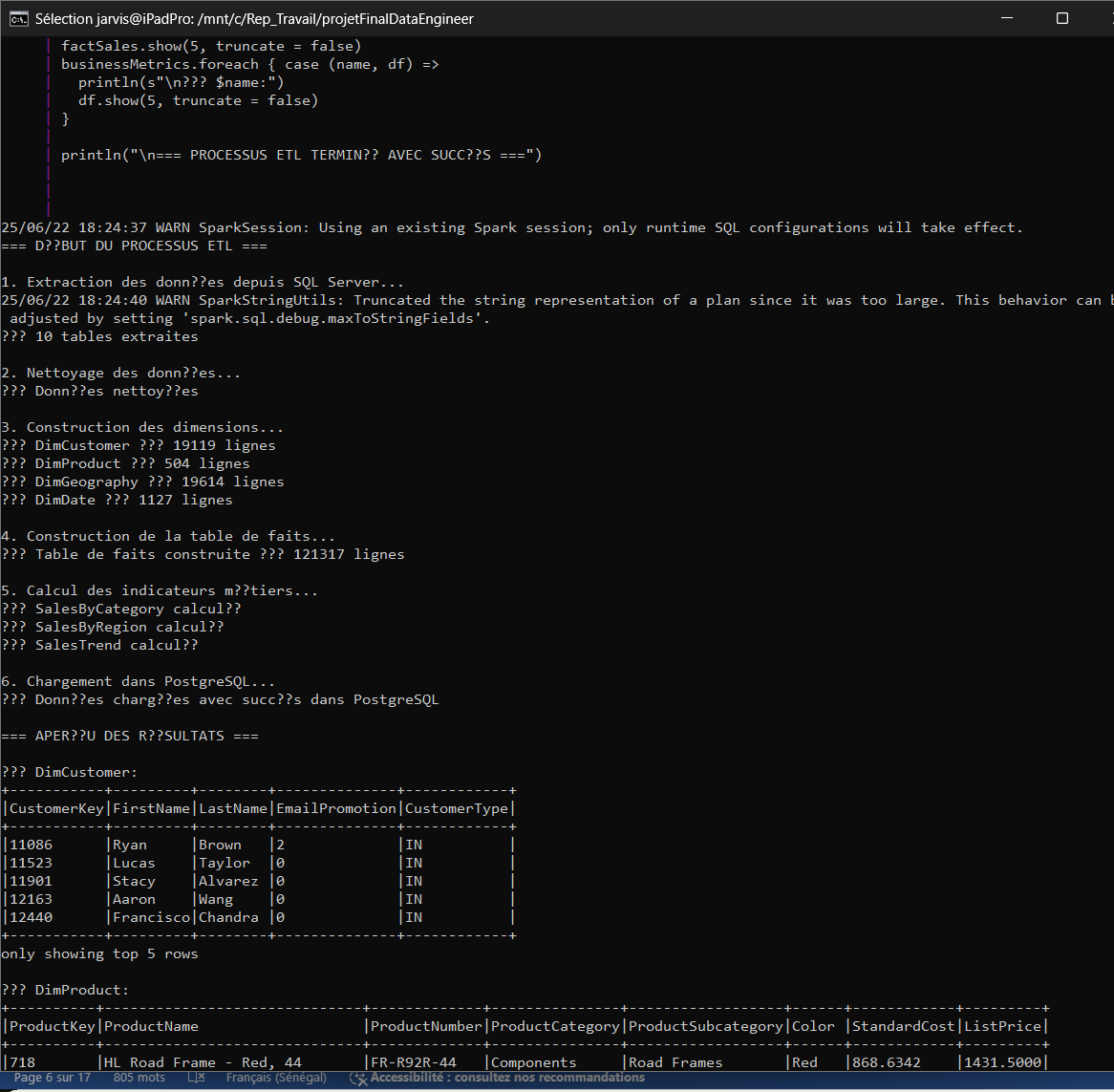
* Lancer spark-shell avec le bon chemin vers le JAR :
* spark-shell --jars /app/libs/**mssql-jdbc-12.10.0.jre11.jar**,/app/libs/**postgresql-42.7.2.jar**

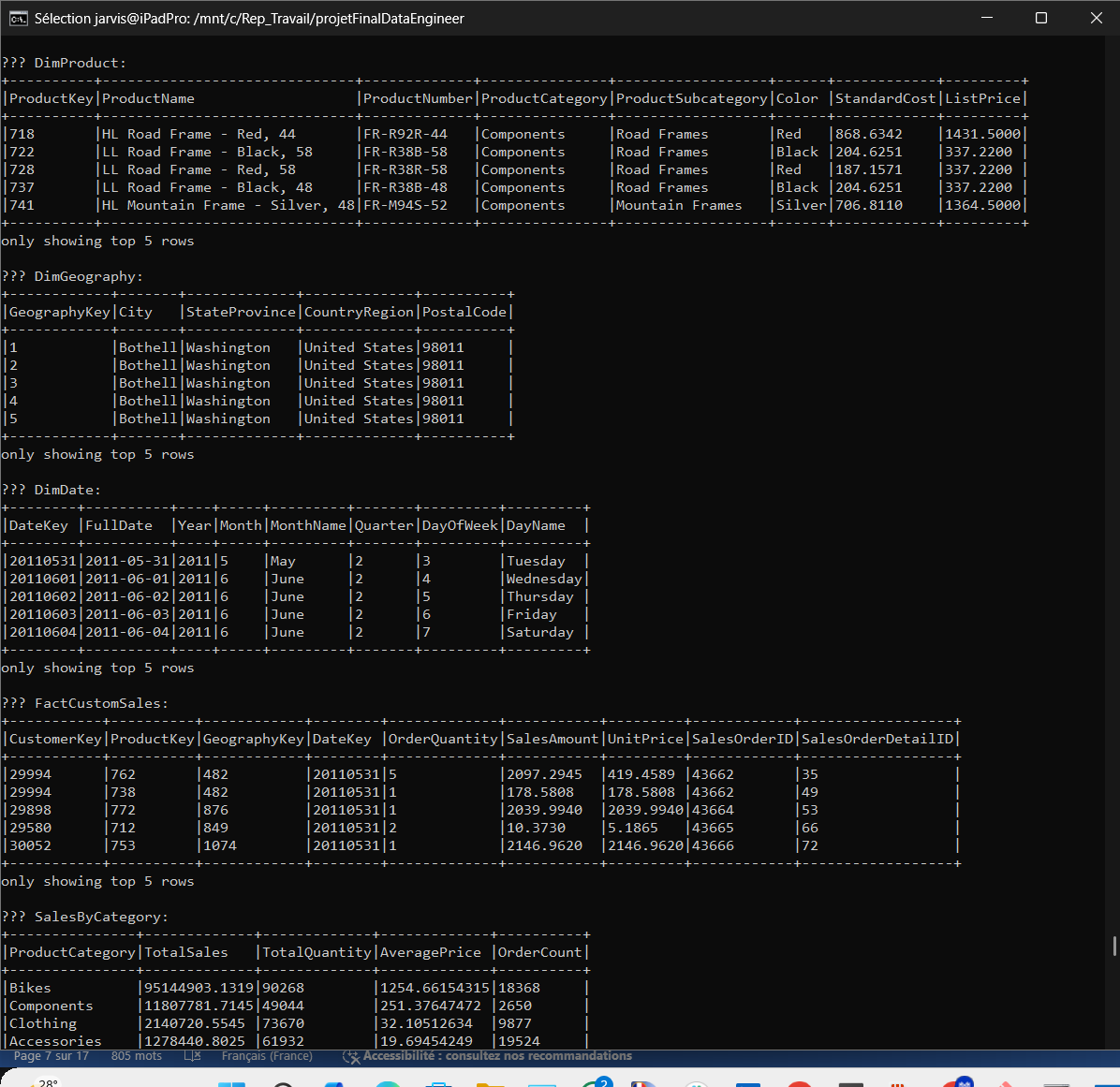
****

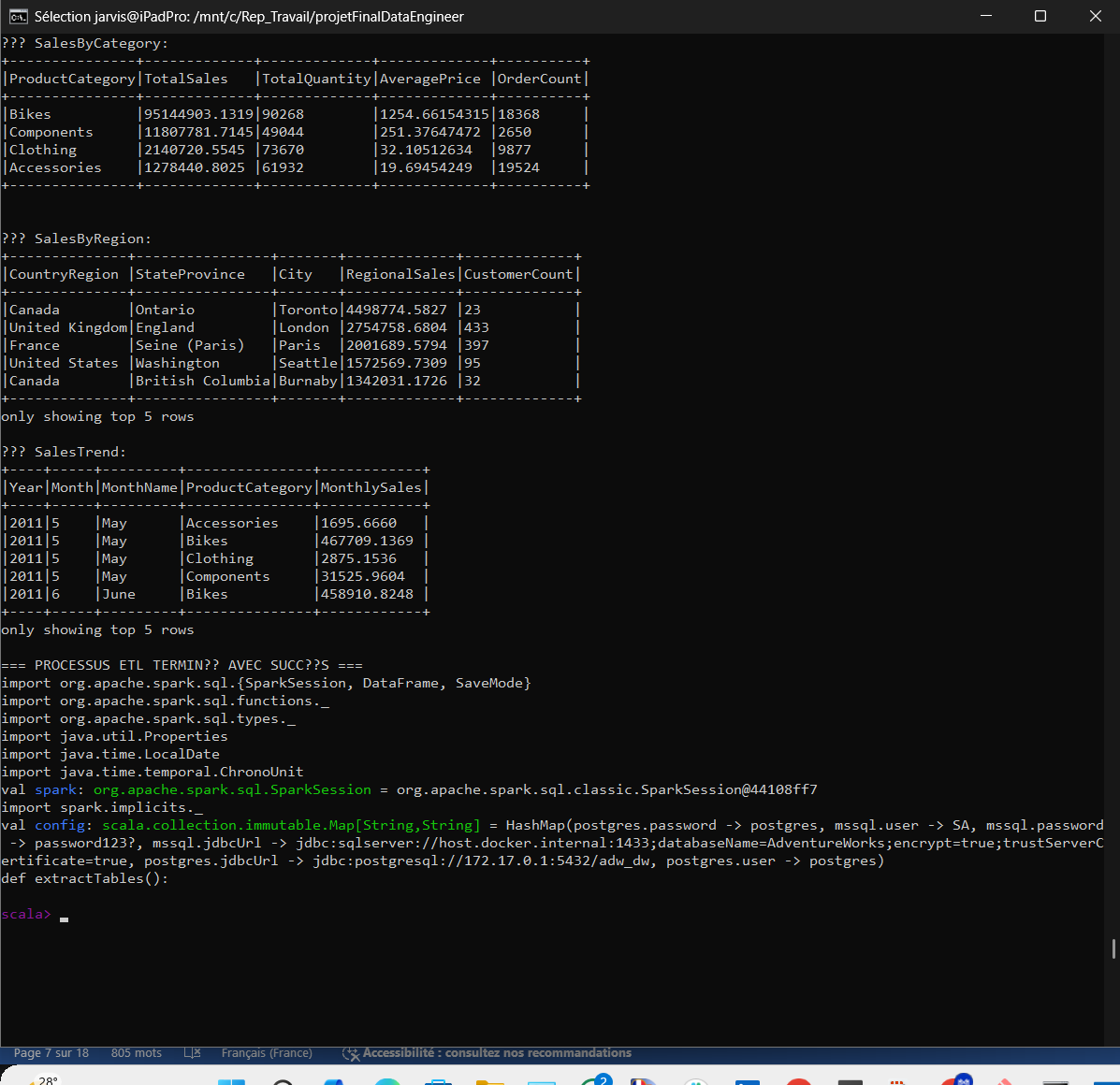
1. **Code ETL avec Spark : Script Scala/Spark pour extraire, transformer et charger les données transformées dans PostgreSQL**

* **Collons le code ETL dans le prompt scala>**

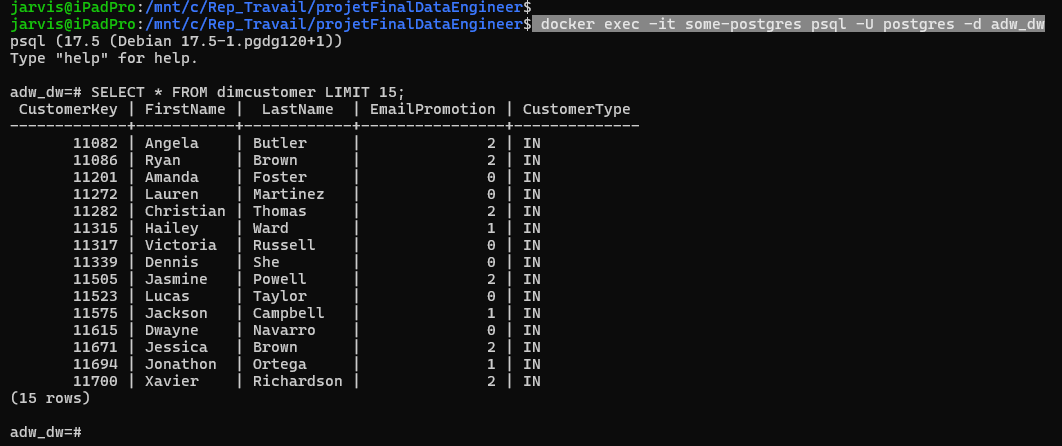




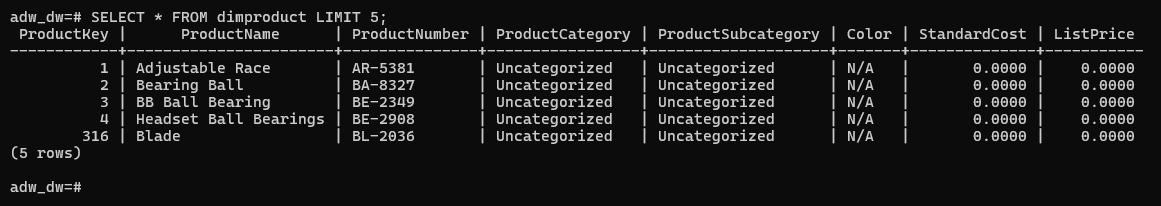




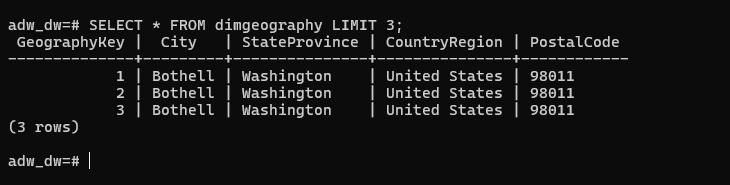
* Voici des commandes SQL à exécuter dans notre psql (PostgreSQL) pour vérifier l’existence et voir un aperçu de toutes les tables supposées chargées par notre ETL.
* Connectons-nous dans notre base PostgreSQL et faisons des requêtes:
* docker run --name some-postgres -e POSTGRES\_PASSWORD=postgres -p 5432:5432 -d postgres
* docker exec -it some-postgres psql -U postgres -d adw\_dw
* **SELECT \* FROM dimcustomer LIMIT 15;**



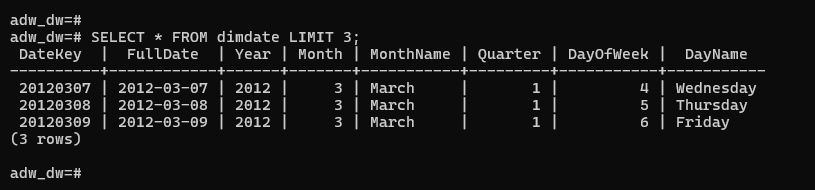
* **SELECT \* FROM dimcustomer LIMIT 3;**



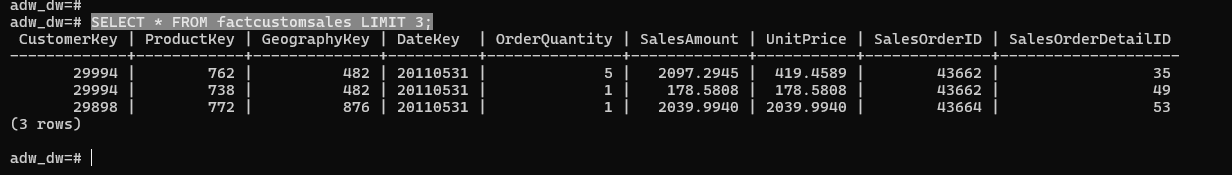
* **SELECT \* FROM dimgeography LIMIT 3;**



* **SELECT \* FROM dimdate LIMIT 3;**

****

* **SELECT \* FROM factcustomsales LIMIT 3;**

****

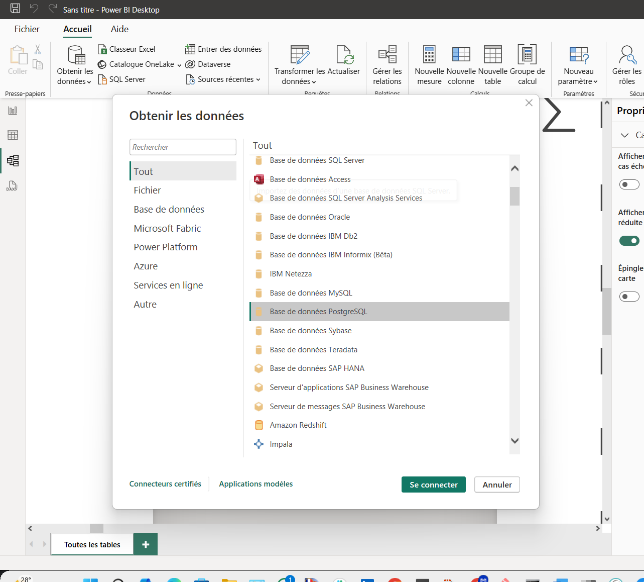
* **Et cetera**
* \dt

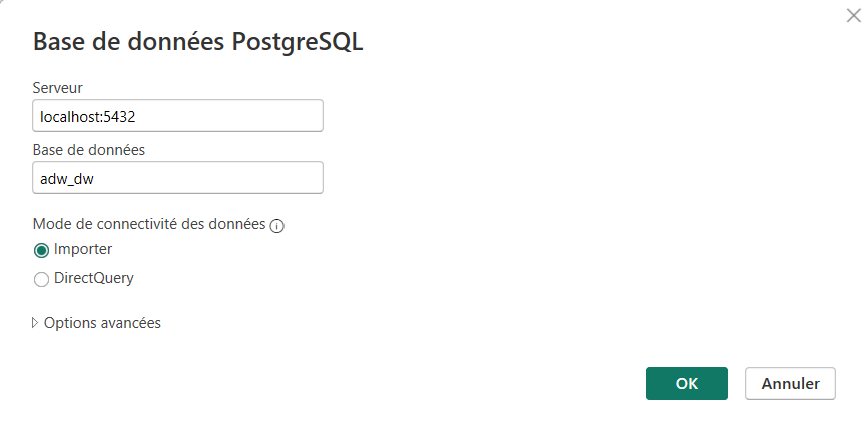


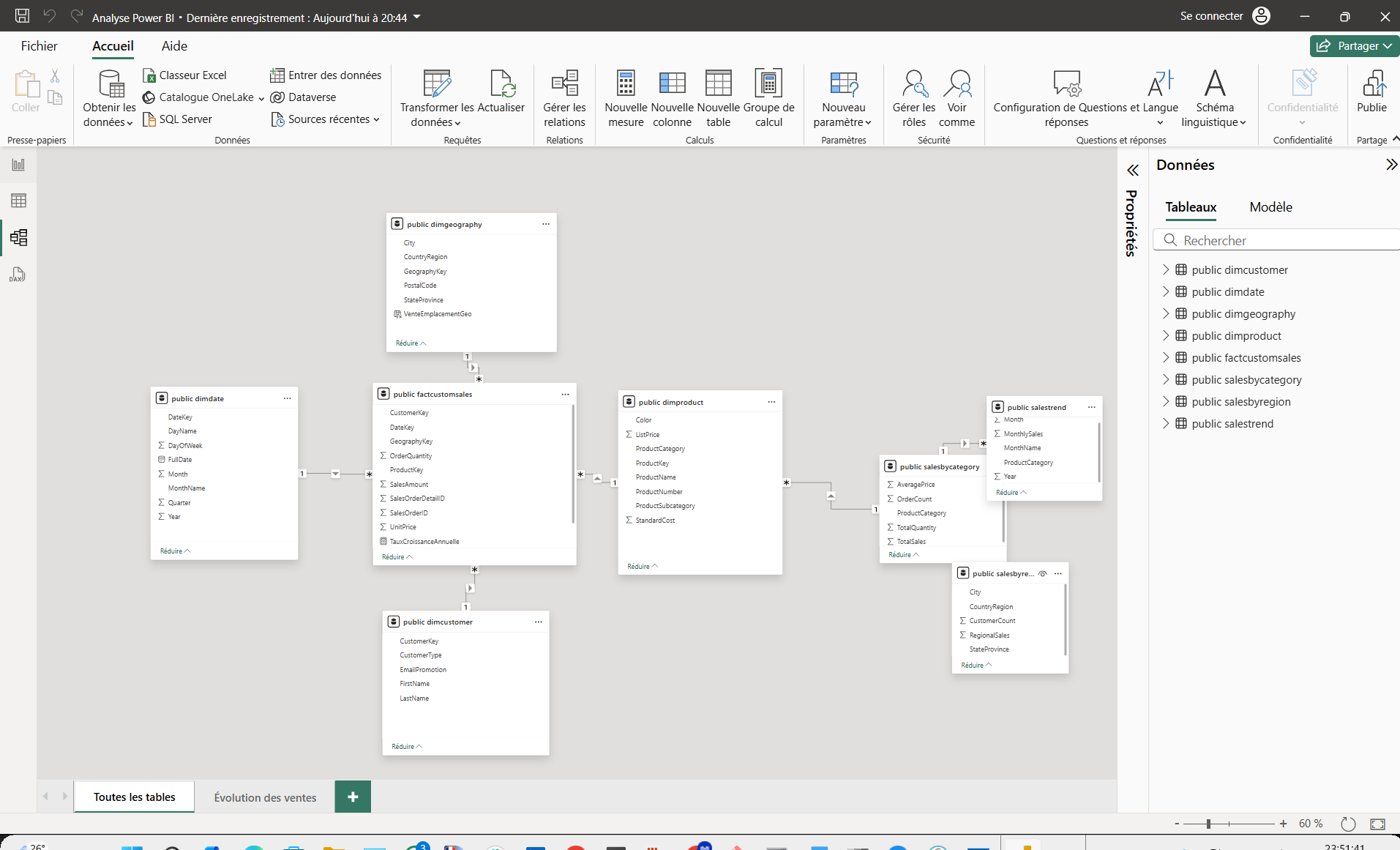
**La base de données adw\_dw.sql et les vues matérialisées (agrégations)**

* **Maintenant l’extraction, la transformation et le chargement dans PostgreSQL est terminé !!**

1. **Modèle de Données : Schéma relationnel entre les tables PostgreSQL :**



****

****

**La base de données adw\_dw.sql et les vues matérialisées agregation (salesbycategory, salesbyregion, salestrend)**

**MODÉLISATION EN ÉTOILE RÉUSSIE…….**

* On peut récupère les données chargées dans la base de données postgreSQL **adw\_dw** après l’ETL
* docker exec -t some-postgres pg\_dump -U postgres -F p adw\_dw > /mnt/c/Rep\_Travail/projetFinalDataEngineer/adw\_dw.sql

****

**La base de données adw\_dw est sur mon Git**

1. **Documentation : Instructions d’utilisation des tableaux de bord Power BI**

**🧩 Objectif**

Ce tableau de bord permet d’analyser les ventes de l’entreprise selon plusieurs dimensions :

* **Ventes dans le temps**
* **Répartition par catégorie et produit**
* **Analyse client**
* **Analyse géographique**
* **Performance et comparaison des ventes par période**

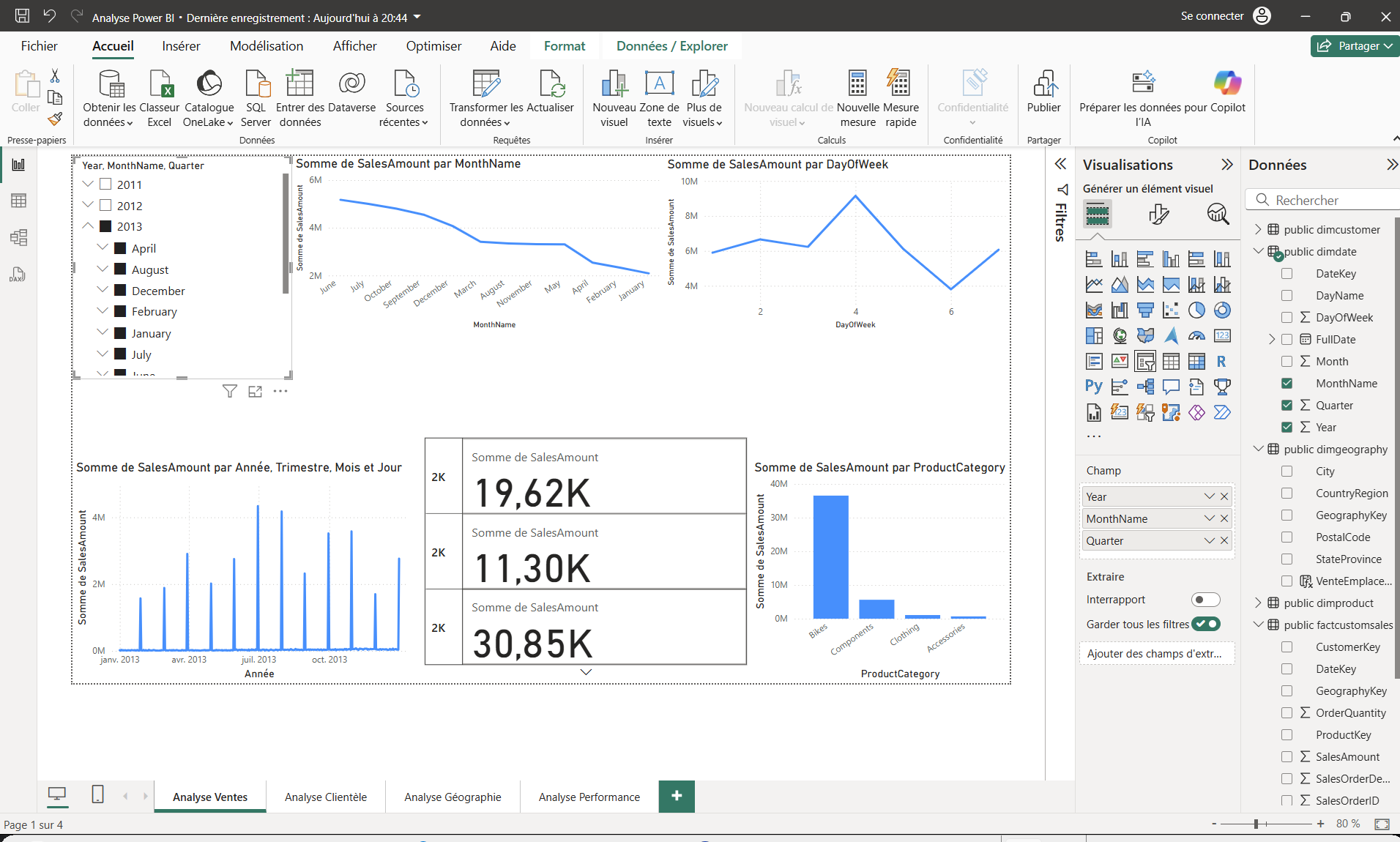
**🖥️ 1. Navigation générale**

* Le rapport se compose de **plusieurs pages** (onglets en bas).
* Chaque page contient des **graphiques dynamiques** et **filtres interactifs**.

**📅 2. Analyse temporelle des ventes**

**📊 Graphique : Évolution des ventes**

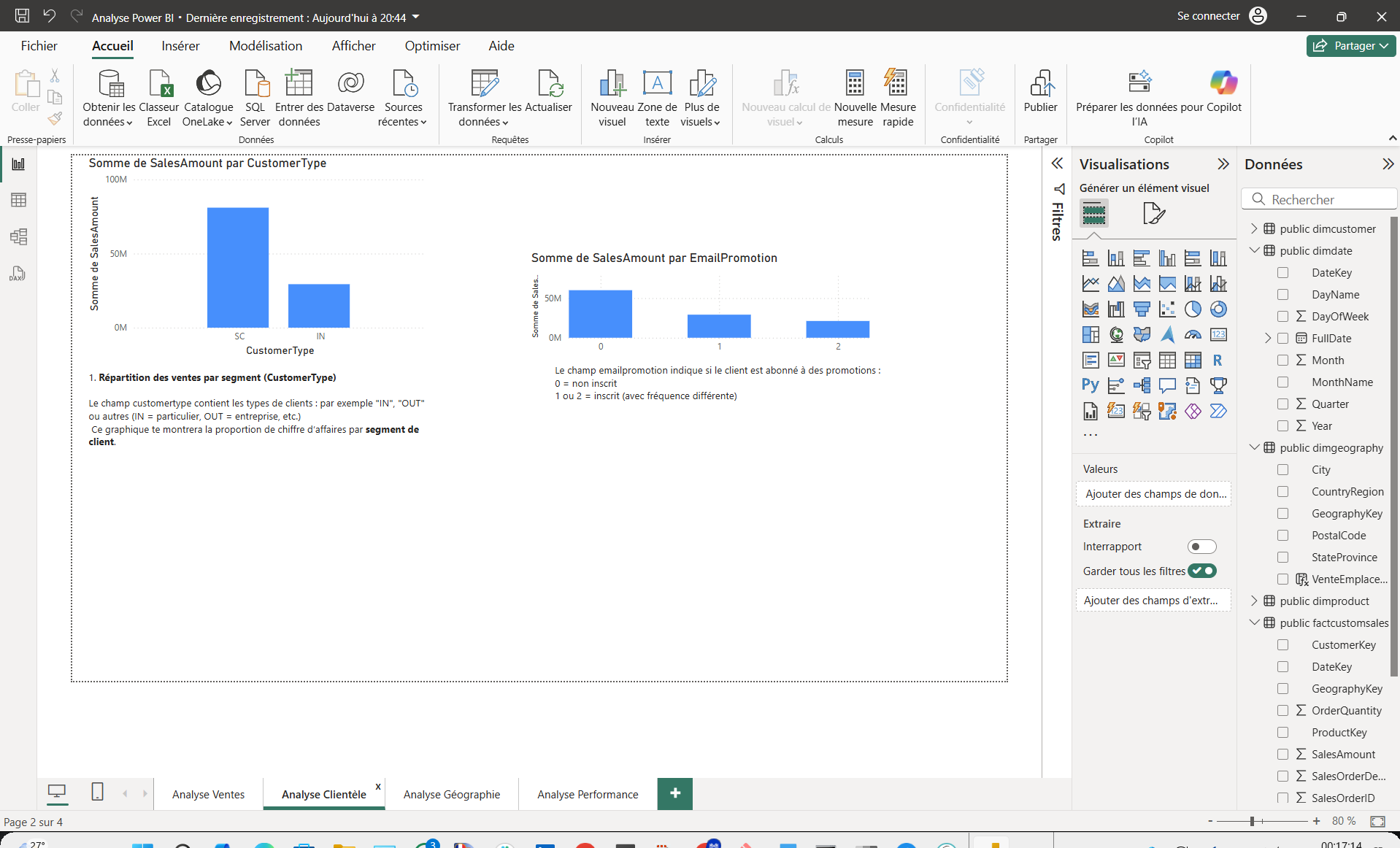
* Permet de suivre les ventes par **année**, **trimestre**, **mois** ou **jour**.
* Utilise les **segments** (filtres en haut ou sur le côté) pour :
  + Sélectionner une année, un mois ou un trimestre spécifique.
  + Comparer des périodes entre elles.
* Visualise les ventes par catégorie de produit.
* Passe la souris sur un élément pour voir le chiffre d'affaires total, la quantité vendue, et le prix moyen.



**👥 3. Analyse clientèle**

**📊 Graphique : Répartition des ventes par segment**

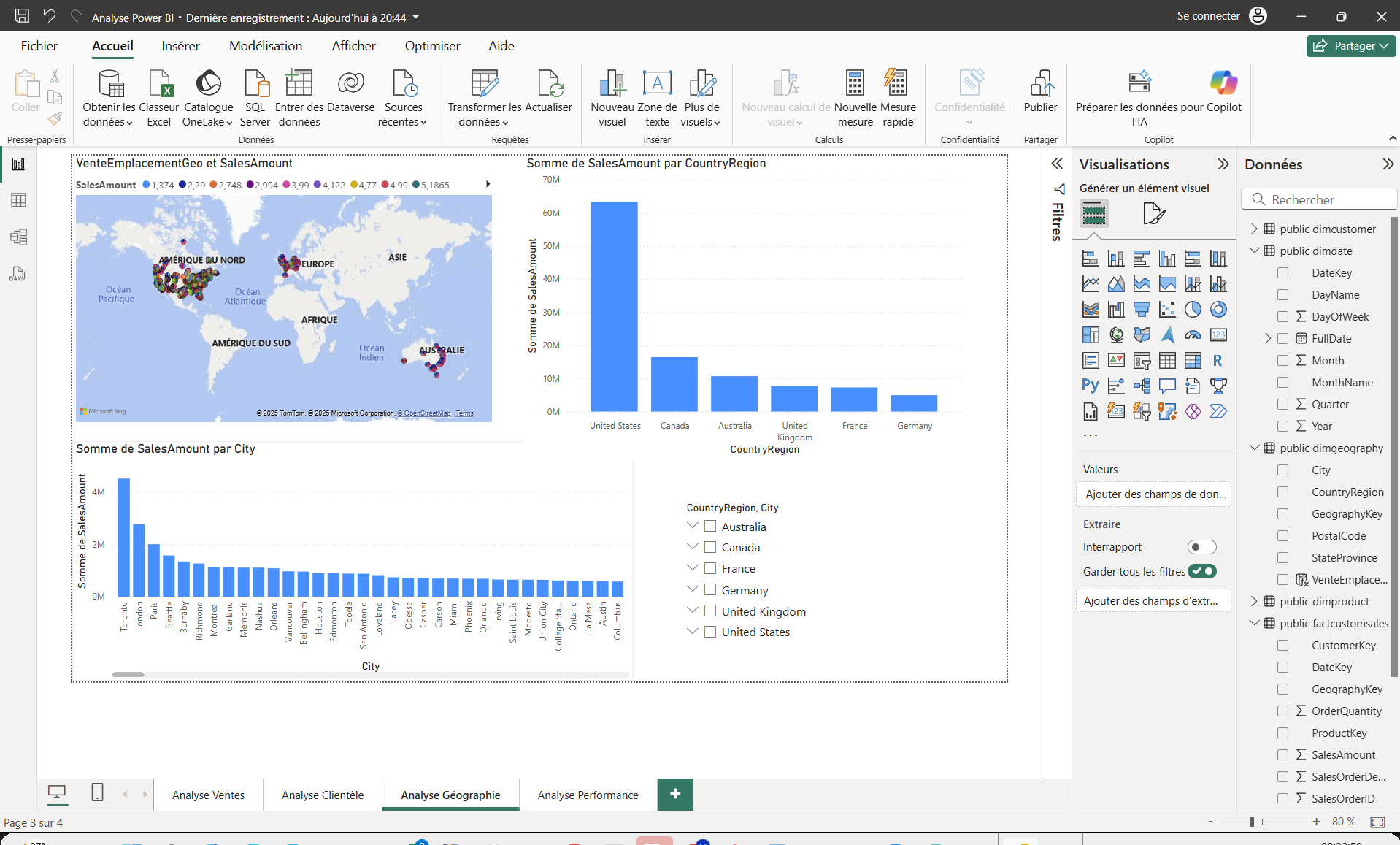
* Affiche la part des ventes par **type de client** (ex : Regular, VIP…).
* Mesure la **fidélité** via le nombre de commandes ou le volume d’achat.

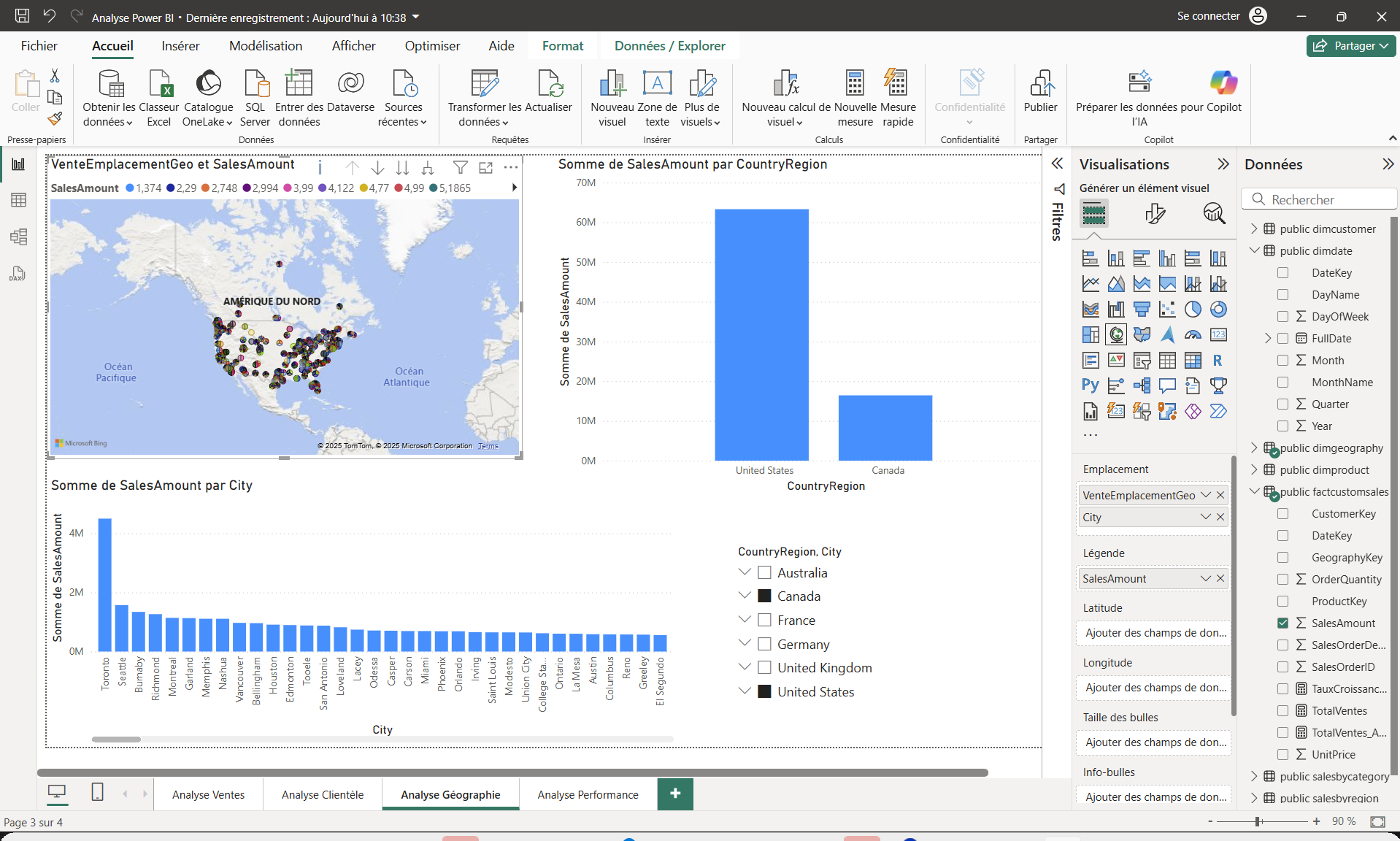


**🗺️ 4. Analyse géographique**

**📍Carte interactive**

* Affiche les **ventes par région et par ville**.
* Utilise la colonne calculée EmplacementGeo pour la géolocalisation.
* Clique sur une ville ou une région pour **filtrer tous les autres visuels**.





**📈 5. Performance commerciale**

**📊 Graphique : Taux de croissance annuelle**

* Compare les ventes de l’année sélectionnée avec l’année précédente.
* Affiche un **indicateur de croissance (%)** basé sur la formule DAX :
* TauxCroissance = DIVIDE([TotalVentes] - [TotalVentes\_AnneePrecedente], [TotalVentes\_AnneePrecedente])

