

REPUBLIQUE DU SENEGAL

Un Peuple, Un But, Une foi



UNIVERSITE GASTON BERGER

L'excellence au service du développement

**INSTITUT DE POLYTECHNIQUE
DE SAINT-LOUIS**

Rigueur et persévérance

DATA ENGINEERING

Final Projet

Spark-ETL-Power BI

Version	Date	Auteur	Commentaires
0.1	15.06.2025	Mouhamadou Mourtala MBOW Momar Talla Salla	Version initiale

I. Etape 1 : Préparation de l'environnement :

1. Cloner le dépôt GitHub et lancer le conteneur MSSQL :

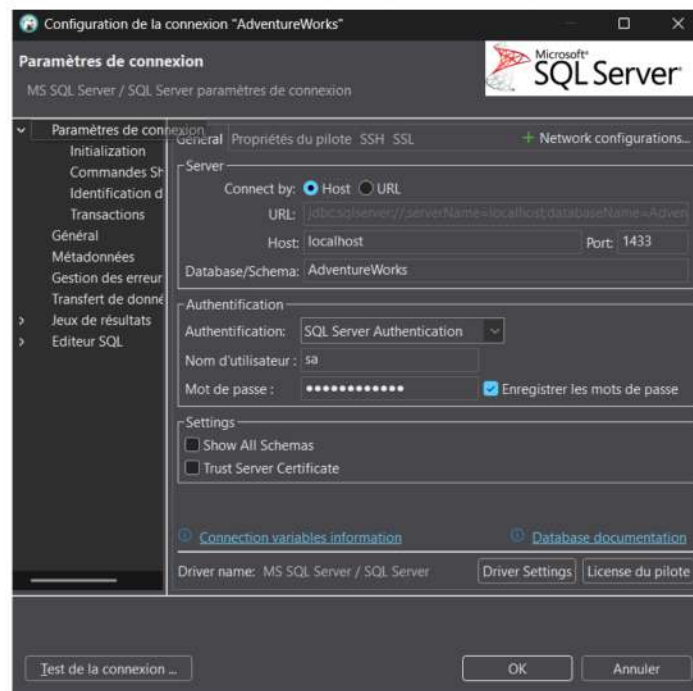
- ➔ git clone https://github.com/sopeKhadim/etl.git
- ➔ cd etl
- ➔ docker-compose up -d

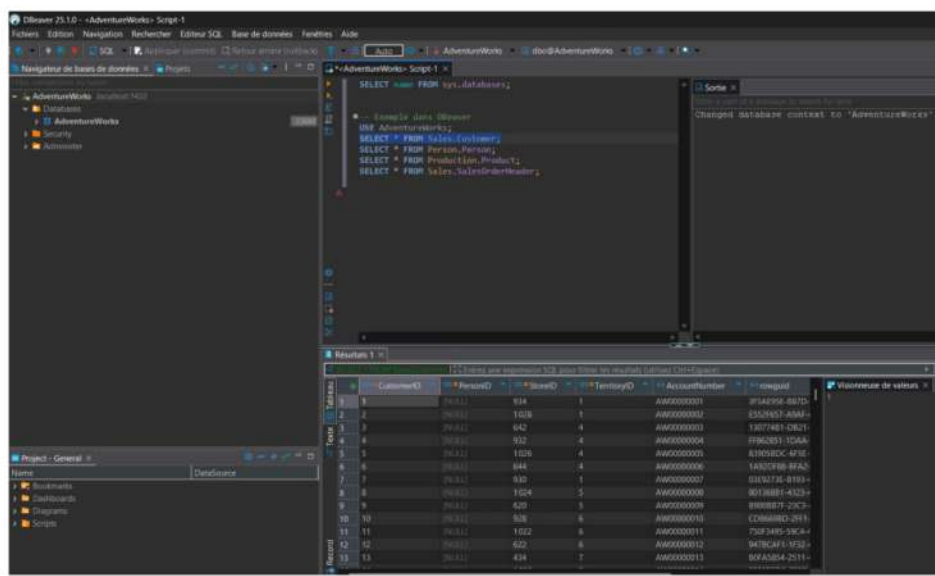
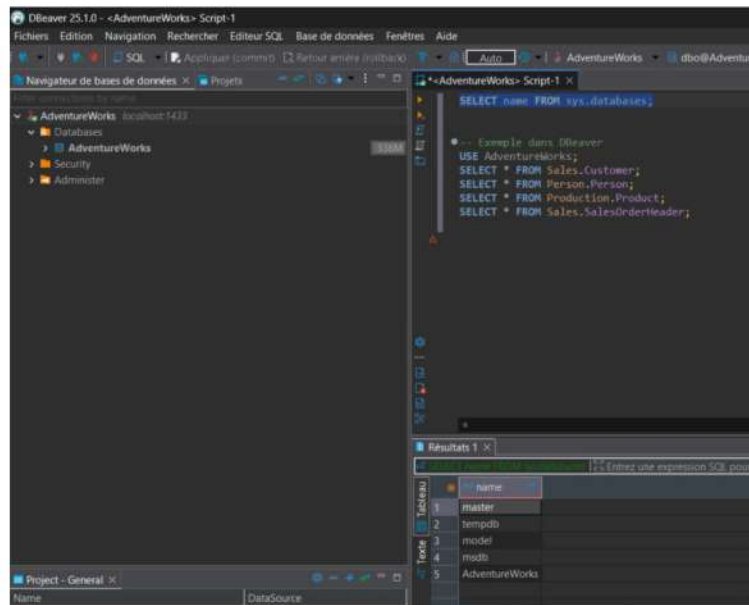
```
DELL XPS@iPadPro MINGW64 /c/Rep_Travail/projetFinalDataEngineer/etl (master)
$ docker-compose up -d
[+] Running 2/2
  ✓ Network etl_default Created                                0.2s
  ✓ Container sqlserver Started                                1.2s

DELL XPS@iPadPro MINGW64 /c/Rep_Travail/projetFinalDataEngineer/etl (master)
$
```

- ➔ 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

```
jarvis@iPadPro:/mnt/c/Rep_Travail/projetFinalDataEngineer$ ls ~/etl/libs
mssql-jdbc-12.10.0.jre11.jar  postgresql-42.7.2.jar
jarvis@iPadPro:/mnt/c/Rep_Travail/projetFinalDataEngineer$
```

➔ Lançons le conteneur **PostgreSQL**

➔ `docker run --name some-postgres -e POSTGRES_PASSWORD=postgres -p 5432:5432 -d postgres`

```
DELL_XPS@iPadPro MINGW64 /c/Rep_Travail/projetFinalDataEngineer/etl (master)
$ docker run --name some-postgres -e POSTGRES_PASSWORD=postgres -p 5432:5432 -d postgres
Unable to find image 'postgres:latest' locally
latest: Pulling from library/postgres
f4ce9941f6e3: Pull complete
b47a445a47f0: Pull complete
eb3a531023c8: Pull complete
928d00623a6e: Pull complete
c95f49cc11b3: Pull complete
abfd68ef219e: Pull complete
dad67da3f26b: Pull complete
c6def2c6e21d: Pull complete
db3ab53631e4: Pull complete
64e8f1b2b243: Pull complete
3664068a9b37: Pull complete
05b641b3bdab: Pull complete
603ef9fcd8e: Pull complete
8a1f652e0c97: Pull complete
Digest: sha256:6cf6142afacfa89fb28b894d6391c7dcbf6523c33178bdc33e782b3b533a9342
Status: Downloaded newer image for postgres:latest
10adb96eeb8841663b99f744e74db3b3225a3c59467c0b3c0e0eb237f8e8de74
```

➔ Créons une base de données **PostgreSQL** nommée **adw_dw** en se connectant au conteneur pour y charger les données transformées :

```
DELL_XPS@iPadPro MINGW64 /c/Rep_Travail/projetFinalDataEngineer/etl (master)
$ docker exec -it some-postgres psql -U postgres
psql (17.5 (Debian 17.5-1.pgdg120+1))
Type "help" for help.

postgres=# CREATE DATABASE adw_dw;
\q
CREATE DATABASE
```

➔ Lançons un conteneur **Spark** avec montage du volume

➔ `docker run -it --rm -v ~/etl:/app -p 4040:4040 bitnami/spark bash`

```
jarvis@iPodPro:/mnt/c/Rep_Travail/projetFinalDataEngineer$
jarvis@iPodPro:/mnt/c/Rep_Travail/projetFinalDataEngineer$
jarvis@iPodPro:/mnt/c/Rep_Travail/projetFinalDataEngineer$ docker run -it --rm -v ~/etl:/app -p 4040:4040 bitnami/spark bash
spark 18:59:14.49 INFO ==>
spark 18:59:14.50 INFO ==> Welcome to the Bitnami spark container
spark 18:59:14.50 INFO ==> Subscribe to project updates by watching https://github.com/bitnami/containers
spark 18:59:14.50 INFO ==> Did you know there are enterprise versions of the Bitnami catalog? For enhanced secure software supply chain
e https://www.arrow.com/globalecs/na/vendors/bitnami/ for more information.
spark 18:59:14.51 INFO ==>
spark@a63b7476f033:/opt/bitnami/spark$
```

➔ Vérifions dans le conteneur que les JAR sont bien là :

➔ `ls /app/libs`

```
spark@37329236615f:/opt/bitnami/spark$ ls /app/libs
mssql-jdbc-12.10.0.jre11.jar postgresql-42.7.2.jar
spark@37329236615f:/opt/bitnami/spark$
```

- ➔ 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`

```
spark@17f9fc018f43:/opt/bitnami/spark$ spark-shell --jars /app/libs/mssql-jdbc-12.10.0.jre11.jar,/app/libs/postgresql-42.7.2.jar
WARNING: Using incubator modules: jdk.incubator.vector
25/06/22 18:08:17 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j2-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to

  ____      _
 / ___|    / \
| |  | |  / _ \
| |  | | / ___ \
| |  | || |___) |
| |  | || |___) |
| |  | || |___) |
|_|  |_| \____/

version 4.0.0

Using Scala version 2.13.16 (OpenJDK 64-Bit Server VM, Java 17.0.15)
Type in expressions to have them evaluated.
Type :help for more information.
Spark context Web UI available at http://17f9fc018f43:4040
Spark context available as 'sc' (master = local[*], app id = local-1750615708642).
Spark session available as 'spark'.

scala> _
```


II. 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**>

```
spark@17f9cf018f43:/opt/bitnami/spark$ spark-shell --master spark://10.10.10.10:7077 --jars /opt/bitnami/postgresql-42.7.0.jar
WARNING: Using incubator modules: jdk.incubator.vector
25/06/22 18:08:17 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              _
     / ___|  _ __| | | | | |
    | |___| | '_ \| |_| | |_| |
    |___|_|_|_||_|_|_|_|_|_|_|
Spark version 4.0.0

Using Scala version 2.13.10 (OpenJDK 64-Bit Server VM, Java 17.0.15)
Type in expressions to have them evaluated.
Type :help for more information.
Spark context Web UI available at http://17f9cf018f43:4040
Spark context available as 'sc' (master = local[*], app id = local-1750615708642).
Spark session available as 'spark'.
=====
val pgProps = new Properties()
pgProps.put("user", config("postgres.user"))
pgProps.put("password", config("postgres.password"))
pgProps.put("driver", "org.postgresql.Driver")

df.write.mode(SaveMode.Overwrite)
  .jdbc(config("postgres.jdbcUrl"), tableName, pgProps)

// 10. Exécution du pipeline ETL
println("=== DÉBUT DU PROCESSUS ETL ===")

println("\n1. Extraction des données depuis SQL Server...")
val sourceTables = extractTables()
println(s"??? $sourceTables.size() tables extraites")

println("\n2. Nettoyage des données...")
val cleanedTables = cleanData(sourceTables)
println(s"??? Données nettoyées")

println("\n3. Construction des dimensions...")
val dimensions = buildDimensions(cleanedTables)
dimensions.foreach { case (name, df) =>
  println(s"??? $name ??? ${df.count()} lignes")
}

println("\n4. Construction de la table de faits...")
val factSales = buildFactSales(dimensions)
println(s"??? Table de faits construite ??? ${factSales.count()} lignes")

println("\n5. Calcul des indicateurs métier...")
val businessMetrics = calculateBusinessMetrics(dimensions("DimProduct"), dimensions("DimGeography"), factSales)
businessMetrics.foreach { case (name, df) => println(s"??? $name calcul ???") }
```

```
Selection javis@iPadPro: /mnt/c/Rep_Travail/projetFinalDataEngineer
factSales.show(5, truncate = false)
businessMetrics.foreach { case (name, df) =>
  println(s"\n??? $name:")
  df.show(5, truncate = false)
}
println("\n=== PROCESSUS ETL TERMINÉ AVEC SUCCÈS ===")

25/06/22 18:24:37 WARN SparkSession: Using an existing Spark session; only runtime SQL configurations will take effect.
=== DÉBUT DU PROCESSUS ETL ===

1. Extraction des données depuis SQL Server...
25/06/22 18:24:40 WARN SparkStringUtils: Truncated the string representation of a plan since it was too large. This behavior can
be adjusted by setting 'spark.sql.debug.maxToStringFields'.
??? 10 tables extraites

2. Nettoyage des données...
??? Données nettoyées

3. Construction des dimensions...
??? DimCustomer ??? 19119 lignes
??? DimProduct ??? 504 lignes
??? DimGeography ??? 19614 lignes
??? DimDate ??? 1127 lignes

4. Construction de la table de faits...
??? Table de faits construite ??? 121317 lignes

5. Calcul des indicateurs métier...
??? SalesByCategory calcul??
??? SalesByRegion calcul??
??? SalesTrend calcul??

6. Chargement dans PostgreSQL...
??? Données chargées avec succès dans PostgreSQL

=== APÉRÇU DES RÉSULTATS ===

??? DimCustomer:
-----
|CustomerKey|FirstName|LastName|Email|Promotion|CustomerType|
-----
|11086      |Ryan     |Brown   |      |          |             |
|11523      |Lucas    |Taylor  |      |          |             |
|11904      |Stacy    |Alvarez |      |          |             |
|12163      |Aaron    |Hane    |      |          |             |
|12640      |Francisco|Chandra |      |          |             |
-----
only showing top 5 rows

??? DimProduct:
-----
|ProductKey|ProductName          |ProductNumber|ProductCategory|ProductSubcategory|Color  |StandardCost|ListPrice|
-----
|718       |IH Road Frame - Red 44|FR-R92R-44   |Components     |Road Frames       |Red    |868.6342  |1431.5000|
-----
Page 6 sur 11 - 403 mots - 127 lignes de code - 128 caractères de code - 128 caractères de code
```

```
Selection jarvis@iPadPro: /mnt/c/Rep_Travail/projetfinalDataEngineer

??? DimProduct:
-----
|ProductKey|ProductName|ProductNumber|ProductCategory|ProductSubcategory|Color|StandardCost|ListPrice|
-----
|718|HL Road Frame - Red, 44|FR-R02R-44|Components|Road Frames|Red|868.6342|1431.5000|
|722|LL Road Frame - Black, 58|FR-R38B-58|Components|Road Frames|Black|204.6251|337.2200|
|728|LL Road Frame - Red, 58|FR-R38B-58|Components|Road Frames|Red|187.1971|337.2200|
|737|LL Road Frame - Black, 48|FR-R38B-48|Components|Road Frames|Black|204.6251|337.2200|
|741|HL Mountain Frame - Silver, 48|FR-M94S-52|Components|Mountain Frames|Silver|706.8110|1364.5000|
-----
only showing top 5 rows

??? DimGeography:
-----
|GeographyKey|City|StateProvince|CountryRegion|PostalCode|
-----
|1|Bothell|Washington|United States|98011|
|2|Bothell|Washington|United States|98011|
|3|Bothell|Washington|United States|98011|
|4|Bothell|Washington|United States|98011|
|5|Bothell|Washington|United States|98011|
-----
only showing top 5 rows

??? DimDate:
-----
|DateKey|FullDate|Year|Month|MonthName|Quarter|DayOfWeek|DayName|
-----
|20110531|2011-05-31|2011|5|May|2|3|Tuesday|
|20110601|2011-06-01|2011|6|June|2|4|Wednesday|
|20110602|2011-06-02|2011|6|June|2|5|Thursday|
|20110603|2011-06-03|2011|6|June|2|6|Friday|
|20110604|2011-06-04|2011|6|June|2|7|Saturday|
-----
only showing top 5 rows

??? FactCustomSales:
-----
|CustomerKey|ProductKey|GeographyKey|DateKey|OrderQuantity|SalesAmount|UnitPrice|SalesOrderID|SalesOrderDetailID|
-----
|29994|762|482|20110531|5|2097.2945|419.4589|43662|35|
|29994|738|482|20110531|1|178.5808|178.5808|43662|49|
|29898|772|876|20110531|1|2039.9940|2039.9940|43664|53|
|30580|712|840|20110531|2|110.3730|55.1665|43665|66|
|30052|753|1074|20110531|1|2146.9620|2146.9620|43666|72|
-----
only showing top 5 rows

??? SalesByCategory:
-----
|ProductCategory|TotalSales|TotalQuantity|AveragePrice|OrderCount|
-----
|Bikes|95144903.1319|90268|1254.66154315|18368|
|Components|11807781.7145|49044|251.37647472|2650|
|Clothing|2140720.5545|73670|32.10512634|9877|
|Accessories|1278440.8025|61932|19.69454249|19524|
-----
Page 7 sur 18 BOS mots L25 (français / français) Accessibilité : consultez nos recommandations
```

```
Selection jarvis@iPadPro: /mnt/c/Rep_Travail/projetfinalDataEngineer

??? SalesByCategory:
-----
|ProductCategory|TotalSales|TotalQuantity|AveragePrice|OrderCount|
-----
|Bikes|95144903.1319|90268|1254.66154315|18368|
|Components|11807781.7145|49044|251.37647472|2650|
|Clothing|2140720.5545|73670|32.10512634|9877|
|Accessories|1278440.8025|61932|19.69454249|19524|
-----

??? SalesByRegion:
-----
|CountryRegion|StateProvince|City|RegionalSales|CustomerCount|
-----
|Canada|Ontario|Toronto|4488774.5827|23|
|United Kingdom|England|London|2754758.6804|433|
|France|Seine (Paris)|Paris|2001689.5794|397|
|United States|Washington|Seattle|1572569.7309|95|
|Canada|British Columbia|Burnaby|1342031.1726|32|
-----
only showing top 5 rows

??? SalesTrend:
-----
|Year|Month|MonthName|ProductCategory|MonthlySales|
-----
|2011|5|May|Accessories|1695.6660|
|2011|5|May|Bikes|467709.1369|
|2011|5|May|Clothing|2875.1536|
|2011|5|May|Components|31525.9604|
|2011|6|June|Bikes|458910.8248|
-----
only showing top 5 rows

=== PROCESSUS ETL TERMIN?? AVEC SUCC??S ===
import org.apache.spark.sql.{SparkSession, DataFrame, SaveMode}
import org.apache.spark.sql.functions._
import org.apache.spark.sql.types._
import java.util.Properties
import java.time.LocalDate
import java.time.temporal.ChronoUnit
val spark: org.apache.spark.sql.SparkSession = org.apache.spark.sql.classic.SparkSession@44108ff7
import spark.implicits._
val config: scala.collection.immutable.Map[String, String] = HashMap(postgres.password -> postgres, mssql.user -> SA, mssql.password -> password123?, mssql.jdbcUrl -> jdbc:sqlserver://host.docker.internal:1433;databaseName=AdventureWorks;encrypt=true;trustServerCertificate=true, postgres.jdbcUrl -> jdbc:postgresql://172.17.0.1:5432/adw_dw, postgres.user -> postgres)
def extractTables():
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;`

```
jarvis@iPadPro:/mnt/c/Rep_Travail/projetFinalDataEngineers$
jarvis@iPadPro:/mnt/c/Rep_Travail/projetFinalDataEngineers$ docker exec -it some-postgres psql -U postgres -d adw_dw
psql (17.5 (Debian 17.5-1.pgdg120+1))
Type "help" for help.

adw_dw=# SELECT * FROM dimcustomer LIMIT 15;
 CustomerKey | FirstName | LastName | EmailPromotion | CustomerType
-----+-----+-----+-----+-----
      11082 | Angela   | Butler   |              2 | IN
      11086 | Ryan     | Brown    |              2 | IN
      11201 | Amanda   | Foster    |              0 | IN
      11272 | Lauren   | Martinez  |              0 | IN
      11282 | Christian| Thomas    |              2 | IN
      11315 | Hailey   | Ward      |              1 | IN
      11317 | Victoria | Russell   |              0 | IN
      11339 | Dennis   | She       |              0 | IN
      11505 | Jasmine  | Powell    |              2 | IN
      11523 | Lucas    | Taylor    |              0 | IN
      11575 | Jackson  | Campbell  |              1 | IN
      11615 | Dwayne   | Navarro   |              0 | IN
      11671 | Jessica  | Brown     |              2 | IN
      11694 | Jonathon | Ortega    |              1 | IN
      11700 | Xavier   | Richardson|              2 | IN
(15 rows)

adw_dw=#
```

➔ `SELECT * FROM dimproduct LIMIT 3;`

```
adw_dw=# SELECT * FROM dimproduct LIMIT 5;
 ProductKey | ProductName | ProductNumber | ProductCategory | ProductSubcategory | Color | StandardCost | ListPrice
-----+-----+-----+-----+-----+-----+-----+-----
          1 | Adjustable Race | AR-5381 | Uncategorized | Uncategorized | N/A |      0.0000 |      0.0000
          2 | Bearing Ball | BA-8327 | Uncategorized | Uncategorized | N/A |      0.0000 |      0.0000
          3 | BB Ball Bearing | BE-2349 | Uncategorized | Uncategorized | N/A |      0.0000 |      0.0000
          4 | Headset Ball Bearings | BE-2908 | Uncategorized | Uncategorized | N/A |      0.0000 |      0.0000
         316 | Blade | BL-2036 | Uncategorized | Uncategorized | N/A |      0.0000 |      0.0000
(5 rows)

adw_dw=#
```

➔ `SELECT * FROM dimgeography LIMIT 3;`

```
adw_dw=# SELECT * FROM dimgeography LIMIT 3;
 GeographyKey | City | StateProvince | CountryRegion | PostalCode
-----+-----+-----+-----+-----
          1 | Bothell | Washington | United States | 98011
          2 | Bothell | Washington | United States | 98011
          3 | Bothell | Washington | United States | 98011
(3 rows)

adw_dw=# |
```


➔ SELECT * FROM dimdate LIMIT 3;

```
adw_dw=#
adw_dw=# SELECT * FROM dimdate LIMIT 3;
 DateKey | FullDate | Year | Month | MonthName | Quarter | DayOfWeek | DayName
-----+-----+-----+-----+-----+-----+-----+-----
 20120307 | 2012-03-07 | 2012 | 3 | March | 1 | 4 | Wednesday
 20120308 | 2012-03-08 | 2012 | 3 | March | 1 | 5 | Thursday
 20120309 | 2012-03-09 | 2012 | 3 | March | 1 | 6 | Friday
(3 rows)

adw_dw=#
```

➔ SELECT * FROM factcustomsales LIMIT 3;

```
adw_dw=#
adw_dw=# SELECT * FROM factcustomsales LIMIT 3;
 CustomerKey | ProductKey | GeographyKey | DateKey | OrderQuantity | SalesAmount | UnitPrice | SalesOrderID | SalesOrderDetailID
-----+-----+-----+-----+-----+-----+-----+-----+-----
      29994 |         762 |          482 | 20110531 |             5 | 2097.2945 | 419.4589 |         43662 |             35
      29994 |         738 |          482 | 20110531 |             1 | 178.5808 | 178.5808 |         43662 |             49
      29898 |         772 |          876 | 20110531 |             1 | 2039.9940 | 2039.9940 |         43664 |             53
(3 rows)

adw_dw=#
```

➔ Et cetera

➔ \dt

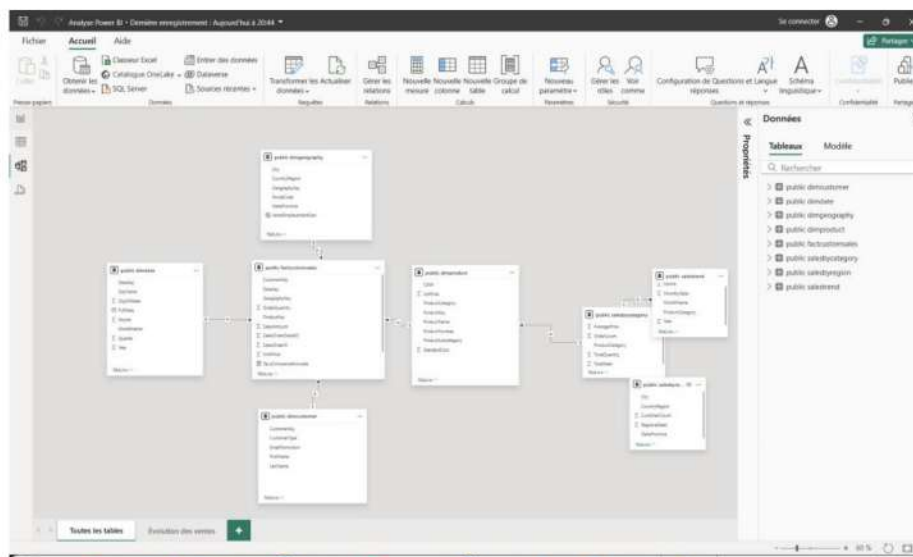
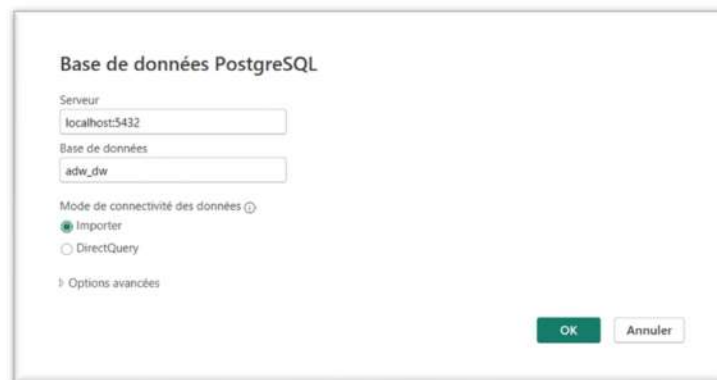
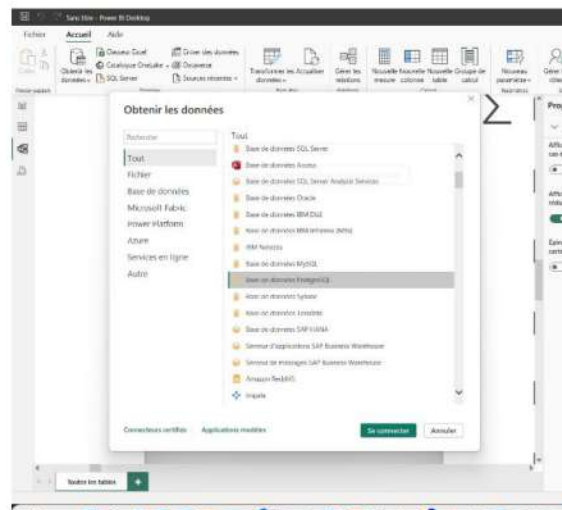
```
adw_dw=#
adw_dw=#
adw_dw=#
adw_dw=# \dt
          List of relations
 Schema | Name | Type | Owner
-----+-----+-----+-----
 public | dimcustomer | table | postgres
 public | dimdate | table | postgres
 public | dimgeography | table | postgres
 public | dimproduct | table | postgres
 public | factcustomsales | table | postgres
 public | salesbycategory | table | postgres
 public | salesbyregion | table | postgres
 public | salestrend | table | postgres
(8 rows)

adw_dw=#
```

La base de données adw_dw.sql et les vues matérialisées (agrégations)

✓ Maintenant l'extraction, la transformation et le chargement dans PostgreSQL est terminé !!

III. Modèle de Données : Schéma relationnel entre les tables PostgreSQL :



La base de données adw_dw.sql et les vues matérialisées aggregation (salesbycategory, salesbyregion, salestrend)

MODÉLISATION EN ÉTOILE RÉUSSIE.....

- On peut récupérer les données chargées dans la base de données postgresSQL **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`

```
jarvis@iPadPro:/mnt/c/Rep_Travail/projetFinalDataEngineer$ docker exec -t some-postgres pg_dump -U postgres -F p adw_dw > /mnt/c/Rep_Travail/projetFinalDataEngineer/adw_dw.sql
jarvis@iPadPro:/mnt/c/Rep_Travail/projetFinalDataEngineer$ ls /mnt/c/Rep_Travail/projetFinalDataEngineer/adw_dw.sql
/mnt/c/Rep_Travail/projetFinalDataEngineer/adw_dw.sql
```

La base de données **adw_dw** est sur mon Git

IV. 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

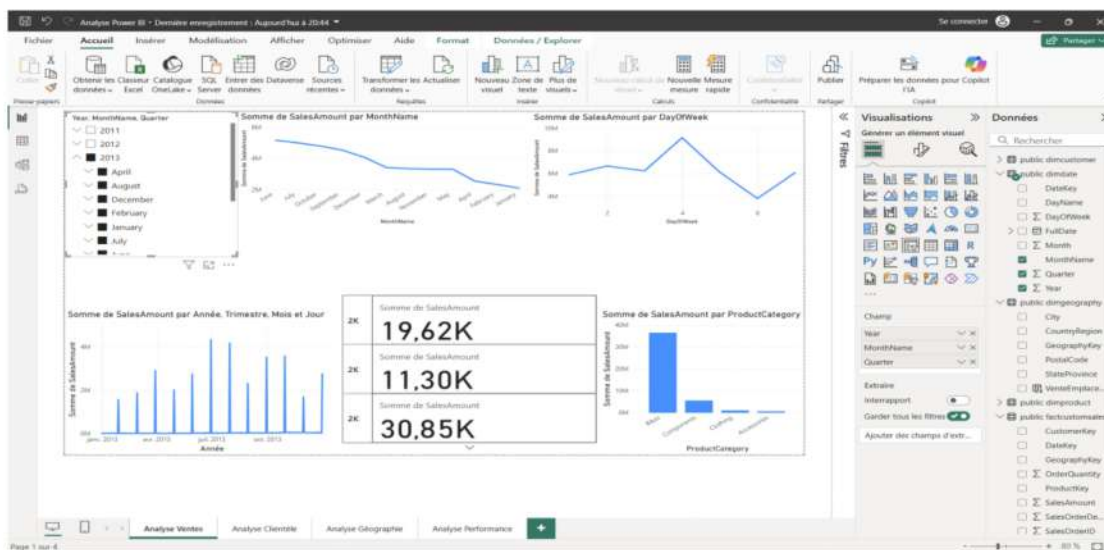
1. Navigation générale

- 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

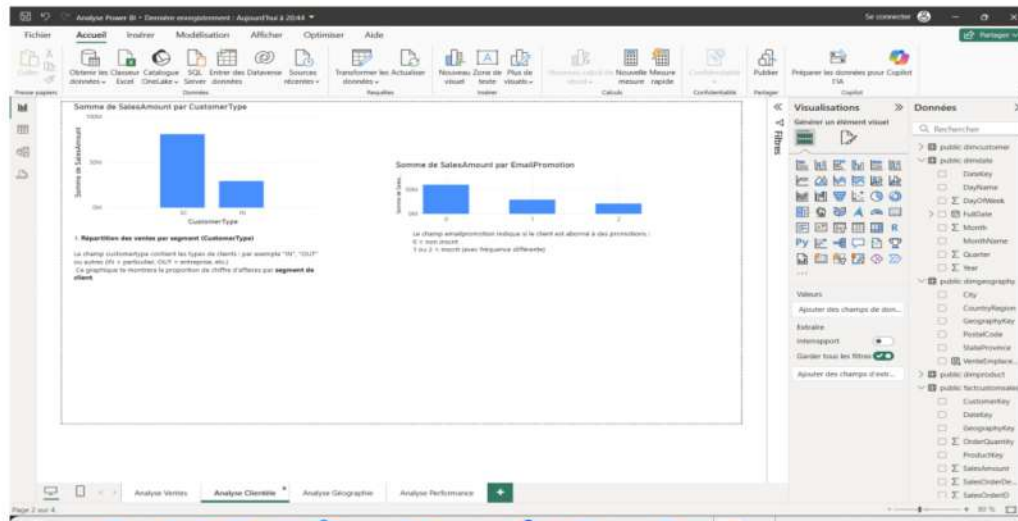
- Permet de suivre les ventes par **année**, **trimestre**, **mois** ou **jour**.
- Utilise les **segments** (filtres en haut ou sur le côté) pour :
 - Sélectionner une année, un mois ou un trimestre spécifique.
 - Comparer des périodes entre elles.
- Visualise les ventes par catégorie de produit.
- 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

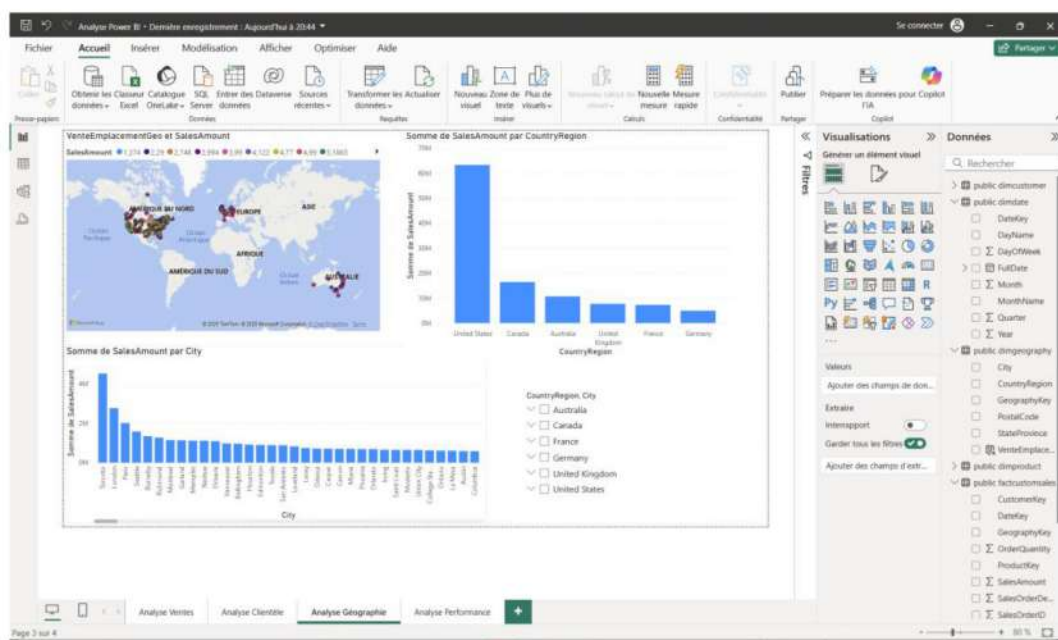
- Affiche la part des ventes par **type de client** (ex : Regular, VIP...).
- Mesure la **fidélité** via le nombre de commandes ou le volume d'achat.

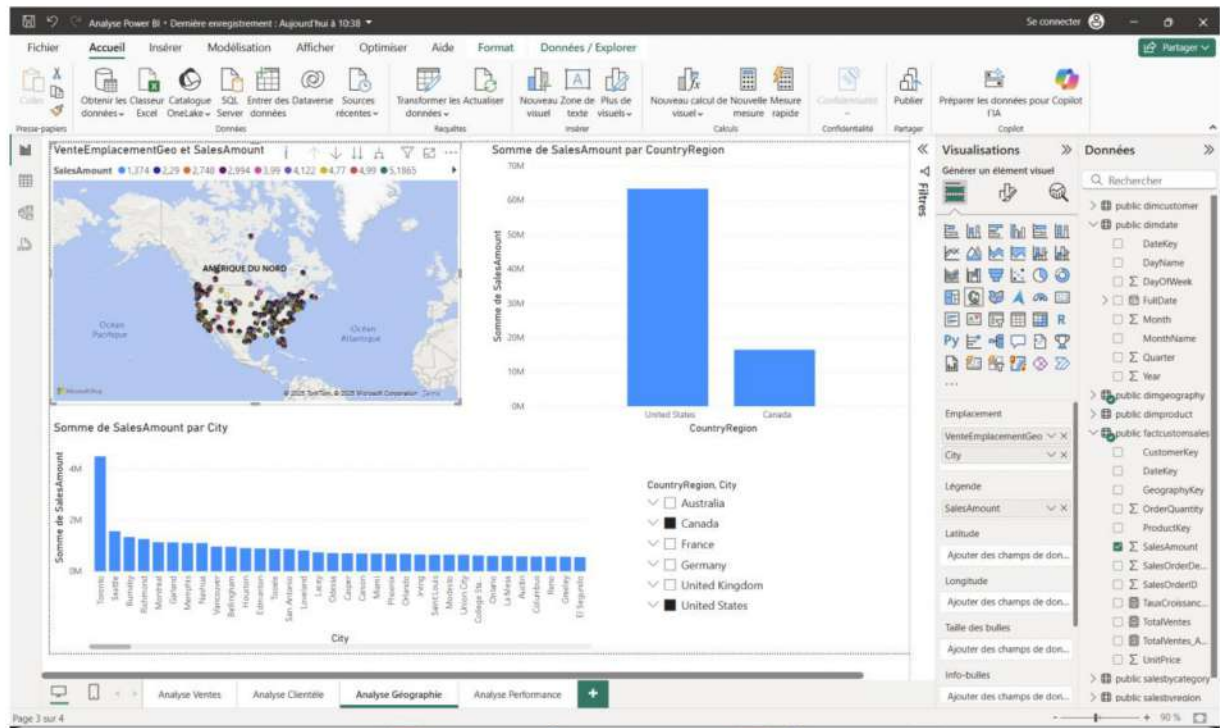


4. Analyse géographique

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 :

➔ $\text{TauxCroissance} = \frac{[\text{TotalVentes}] - [\text{TotalVentes_AnneePrecedente}]}{[\text{TotalVentes_AnneePrecedente}]}$

