Rapport de TP

Pipeline OLTP → ETL Pentaho PDI → Data Warehouse → Reporting Power BI

Fait par :**Khadija Nachid Idrissi**
:**Rajae Fdili**
:**Aya Hamim**

Année Universitaire : 2024–2025

Table des matières

0.1 Introduction 4
0.1.1 OLTP vs OLAP 4
0.1.2 Rôle de Pentaho Data Integration (ETL) 4
0.1.3 Rôle du Data Warehouse 4
0.1.4 Rôle de Power BI 5
0.2 Modèle OLTP 6
0.2.1 Contexte 6
0.2.2 Structure de la base OLTP 6
0.2.3 Script SQL de création de la base 6
0.3 Vérification de la structure de la base OLTP 7
0.3.1 Affichage des tables 7
0.3.2 Description des tables 7
0.3.3 Vérification de la génération des fichiers CSV 9
0.3.4 Vérification de la génération des fichiers CSV 9
0.4 Importation des CSV dans MySQL 11
0.4.1 Étapes d'importation 11
0.4.2 Vérification de l'importation 11
0.5 Création du Data Warehouse 12
0.5.1 Tables créées 12
0.5.2 Vérification de la création des tables 12
0.5.3 Remarques 14
0.6 Pentaho PDI – Configuration et Transformations 14
0.6.1 Configuration de Pentaho PDI 14
0.6.2 Vérification des connexions 14
0.7 Transformation 1 : Chargement de la dimension `DimClient` 15
0.7.1 Création de la transformation 15
0.7.2 Extraction et chargement 15
0.7.3 Résultat obtenu 16
0.8 Transformation 2 : Chargement de la dimension `DimProduit` 16

0.8.1 Création de la transformation 17
0.8.2 Extraction et renommage des colonnes 17
0.8.3 Chargement dans la dimension 17
0.8.4 Résultat obtenu 17
0.9 Transformation 3 : Génération de la dimension `DimDate` 18
0.9.1 Création de la transformation 18
0.9.2 Génération des dates et calcul des composantes 19
0.9.3 Chargement dans la dimension 19
0.9.4 Résultat obtenu 19
0.10 Transformation 4 : Chargement de la table de faits `FactVentes` 20
0.10.1 Création de la transformation 20
0.10.2 Extraction des données OLTP 20
0.10.3 Lookup des dimensions 20
0.10.4 Calcul du montant total 21
0.10.5 Sélection des champs finaux et chargement 21
0.10.6 Résultat obtenu 21
0.11 Création du Job d'Orchestration Pentaho 22
0.11.1 Structure du Job 22
0.11.2 Étapes de création 23
0.11.3 Capture d'écran du Job 23
0.12 Requêtes OLAP sur le Data Warehouse 23
0.12.1 Requête 1 : Chiffre d'affaires par ville 23
0.12.2 Requête 2 : Chiffre d'affaires par catégorie de produit 24
0.12.3 Requête 3 : Évolution mensuelle des ventes 24
0.12.4 Requête 4 : Top 10 des produits les plus vendus 25
0.12.5 Requête 5 : Analyse croisée trimestre–catégorie 25
0.12.6 Avantages du schéma en étoile 26
0.13 Power BI - Reporting et Visualisation 26
0.13.1 Prérequis 26
0.13.2 Étape 1 : Se connecter à MySQL depuis Power BI 26
0.13.3 Étape 2 : Importer les tables du Data Warehouse 28
0.13.4 Étape 3 : Vérifier et configurer le modèle de données 28
0.14 Étape 4 : Créer les visualisations 29
0.14.1 Visuel 1 : Chiffre d'affaires par ville (Graphique à barres) 29
0.14.2 Visuel 2 : Chiffre d'affaires par catégorie (Graphique en secteurs) . 30
0.14.3 Visuel 3 : Évolution des ventes mensuelles (Graphique en courbes) . 31
0.14.4 Visuel 4 : Top 10 des produits (Graphique à barres horizontales) . . 31

0.15 Étape 5 : Ajouter des segments (Slicers) pour l'interactivité 32
0.15.1 5.1 - Ajouter un segment Année 32
0.15.2 5.2 - Ajouter un segment Catégorie 33
0.15.3 5.3 - Tester l'interactivité 34
0.16 Étape 6 : Créer des mesures DAX (optionnel mais recommandé) 34
0.16.1 6.1 - Mesure "CA Total" 34
0.16.2 6.2 - Mesure "Nombre de ventes" 35
0.16.3 6.3 - Mesure "Panier Moyen" 35
0.16.4 6.4 - Mesure "Clients Uniques" 36
0.16.5 6.5 - Utiliser les mesures 36
0.17 Étape 9 : Créer une page de détails (Drill-through) 37
0.17.1 9.1 - Créer une nouvelle page 37
0.17.2 9.3 - Ajouter des visuels détaillés 38
0.17.3 9.4 - Tester le drill-through 39

0.1 Introduction

Ce rapport présente la construction complète d'un pipeline décisionnel allant d'une base de données opérationnelle (OLTP) à la création d'un Data Warehouse alimenté par un processus ETL réalisé dans Pentaho PDI, puis analysé dans Power BI.

L'objectif de ce TP est de mettre en place une architecture permettant à l'entreprise fictive **TechStore** de transformer ses données brutes en informations utiles pour la prise de décision.

0.1.1 OLTP vs OLAP

OLTP (Online Transaction Processing). Les systèmes OLTP gèrent les opérations quotidiennes :

— gestion des clients,
— enregistrement des commandes,
— mise à jour des produits.

Ils sont fortement normalisés pour éviter la redondance et garantir l'intégrité.

OLAP (Online Analytical Processing). Les systèmes OLAP sont utilisés pour l'analyse :

— tendances de ventes,
— performance des produits,
— comportement client.

Ils s'appuient sur un **Data Warehouse**, structuré en schémas en étoile.

0.1.2 Rôle de Pentaho Data Integration (ETL)

Pentaho PDI permet :

— l'extraction des données OLTP,
— le nettoyage, la transformation, l'harmonisation,
— le chargement vers le Data Warehouse.

0.1.3 Rôle du Data Warehouse

Le Data Warehouse regroupe des données :

— intégrées,
— nettoyées,
— historisées,
— optimisées pour l'analyse.

Il constitue la base des rapports Power BI.

0.1.4 Rôle de Power BI

Power BI permet :

— la connexion au DWH,

— la création de tableaux de bord,

— une visualisation dynamique des indicateurs.

0.2 Modèle OLTP

0.2.1 Contexte

TechStore est une entreprise spécialisée dans la vente de matériel électronique. La base OLTP constitue la source initiale des données, représentant les transactions quotidiennes : clients, produits, commandes et lignes de commande.

0.2.2 Structure de la base OLTP

La base de données `ventes_oltp` contient quatre tables principales :

— **clients**,

— **produits**,

— **commandes**,

— **lignes_commandes**.

Ces tables sont reliées par des clés étrangères pour former un modèle relationnel normalisé.

0.2.3 Script SQL de création de la base

0.3 Vérification de la structure de la base OLTP

Après la création des tables, il est essentiel de vérifier leur existence ainsi que leur structure.

0.3.1 Affichage des tables

Commande exécutée :

```
SHOW TABLES;
```

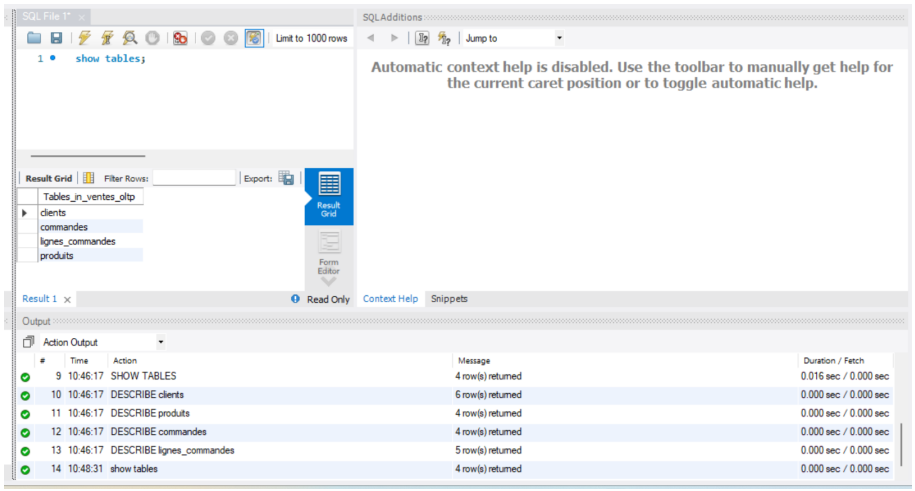


FIGURE 1 – Résultat de la commande SHOW TABLES

0.3.2 Description des tables

Table : clients

```
DESCRIBE clients;
```

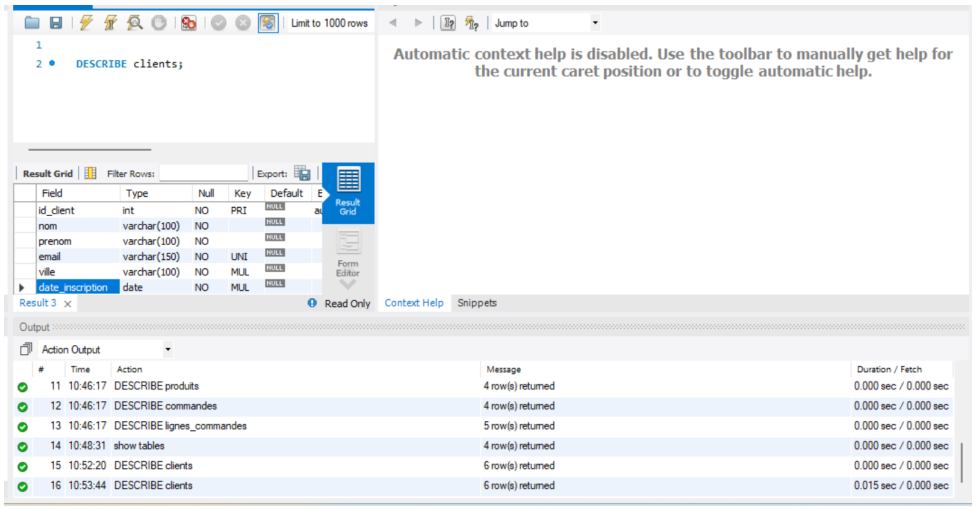


FIGURE 2 – Structure de la table clients

Table : produits

```
DESCRIBE produits;
```

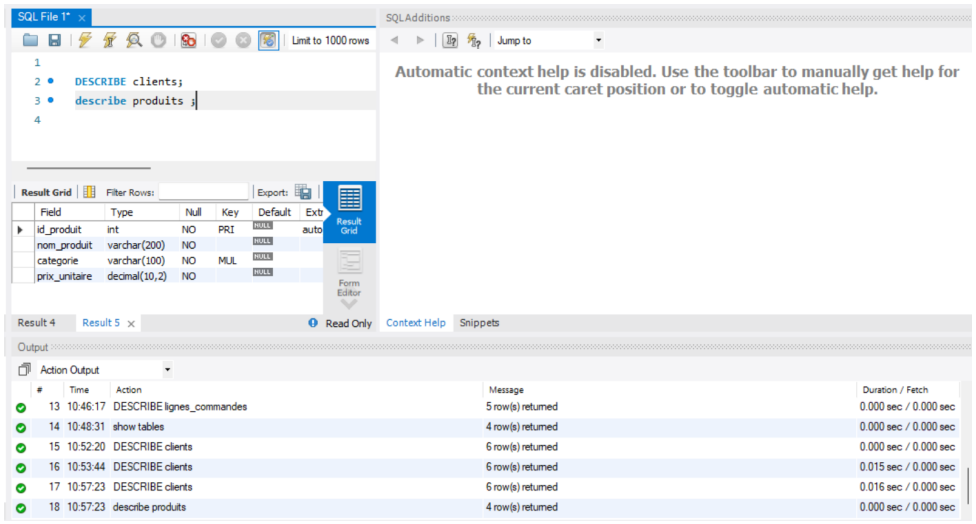


FIGURE 3 – Structure de la table produits

Table : commandes

```
DESCRIBE commandes;
```

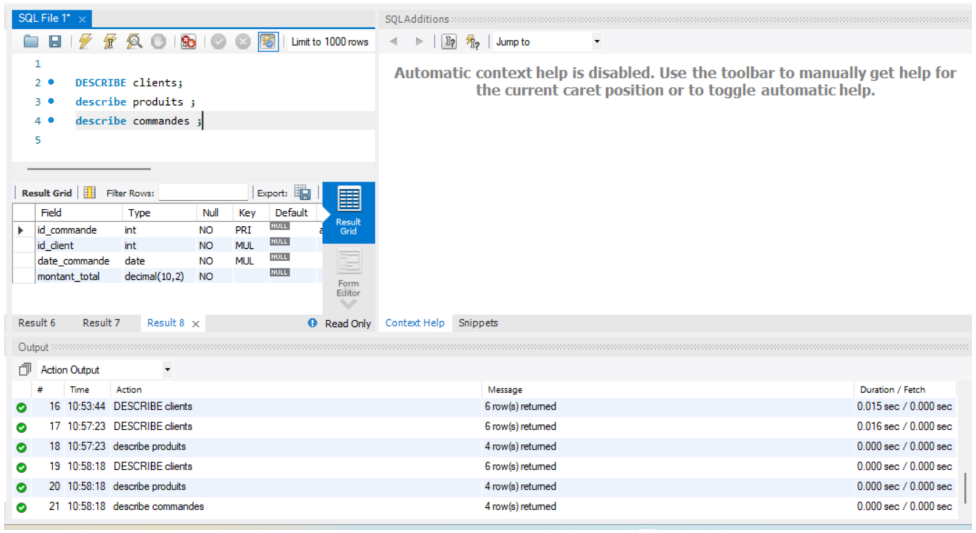


FIGURE 4 – Structure de la table commandes

Table : lignes_commandes

```
DESCRIBE lignes_commandes;
```

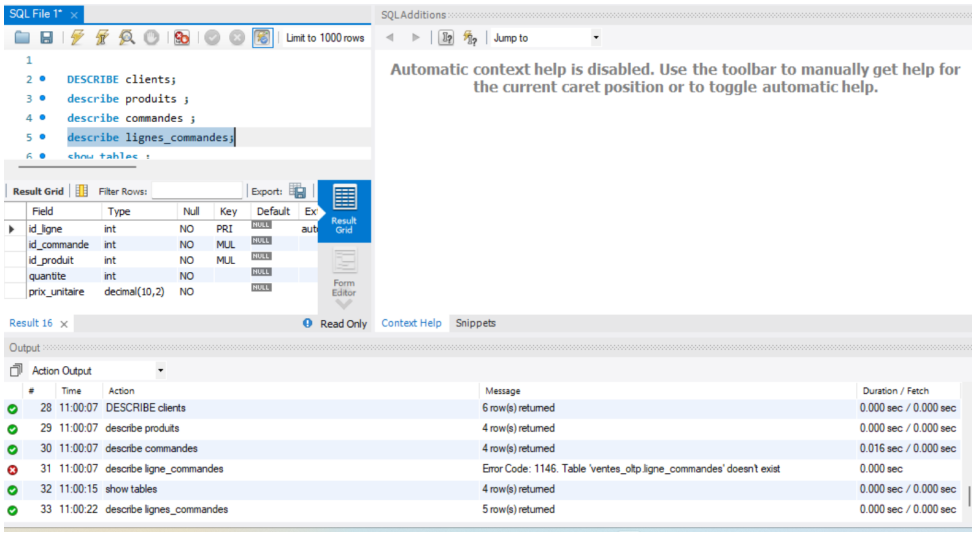


FIGURE 5 – Structure de la table lignes_commandes

0.3.3 Vérification de la génération des fichiers CSV

Après l'exécution du script, quatre fichiers CSV doivent être présents : `clients.csv`, `produits.csv`, `commandes.csv`, `lignes_commandes.csv`.

0.3.4 Vérification de la génération des fichiers CSV

Après l'exécution du script, quatre fichiers CSV doivent être présents : `clients.csv`, `produits.csv`, `commandes.csv`, `lignes_commandes.csv`.

Vérification dans Colab / terminal

Pour vérifier que les fichiers ont bien été créés, on peut exécuter la commande suivante dans Colab ou dans un terminal :

```
!ls
```

ou simplement vérifier dans l'explorateur de fichiers de Colab.

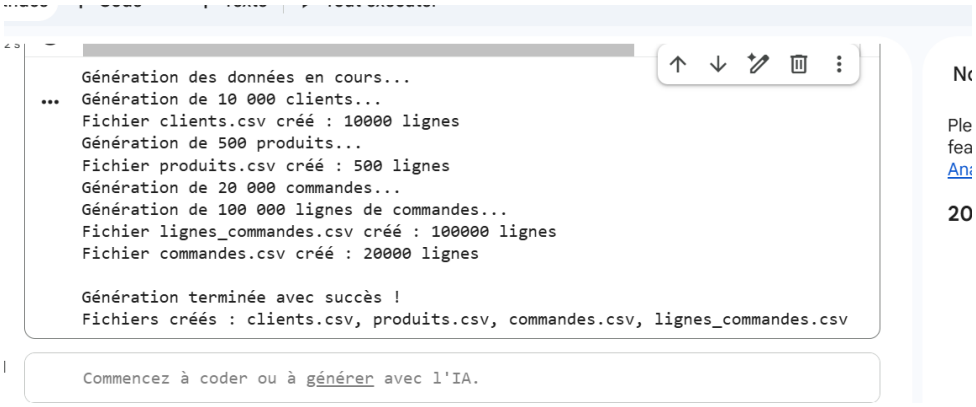


FIGURE 6 – Fichiers générés dans Colab après l'exécution du script

Aperçu des données générées

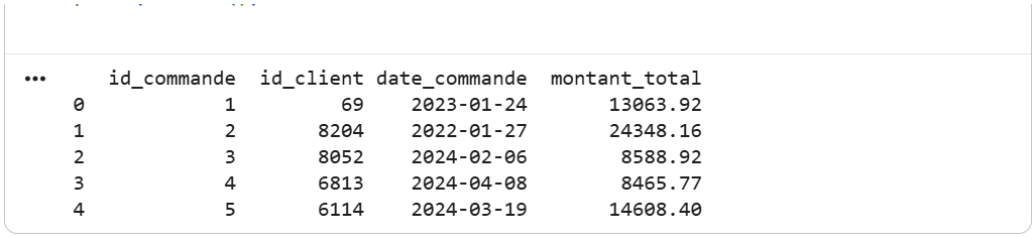
Une fois les fichiers créés, on peut visualiser les premières lignes pour vérifier la cohérence des données. Voici quelques exemples :

	id_client	nom	prenom	email	ville	date_inscription
0	1	Riou	Frédéric	client1@voila.fr	Rennes	2025-02-10
1	2	Moulin	Maurice	client2@laposte.net	Lyon	2025-02-05
2	3	Labbé	Nicole	client3@hotmail.fr	Paris	2024-08-29
3	4	David	Antoine	client4@hotmail.fr	Reims	2023-07-18
4	5	Bodin	Josette	client5@voila.fr	Nice	2024-07-28

FIGURE 7 – Aperçu des 5 premières lignes de clients.csv

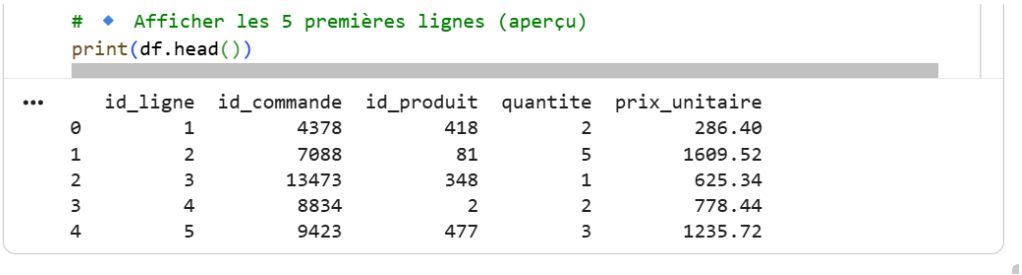
	id_produit	nom_produit	categorie	prix_unitaire
0	1	MacBook Pro Ultra	Ordinateurs	100.63
1	2	MacBook Pro Standard	Ordinateurs	778.44
2	3	MacBook Pro Ultra	Ordinateurs	349.86
3	4	MacBook Pro Standard	Ordinateurs	1669.86
4	5	Lenovo ThinkPad Standard	Ordinateurs	1275.30

FIGURE 8 – Aperçu des 5 premières lignes de produits.csv



	id_commande	id_client	date_commande	montant_total
0	1	69	2023-01-24	13063.92
1	2	8204	2022-01-27	24348.16
2	3	8052	2024-02-06	8588.92
3	4	6813	2024-04-08	8465.77
4	5	6114	2024-03-19	14608.40

FIGURE 9 – Aperçu des 5 premières lignes de commandes.csv



```
# 🔹 Afficher les 5 premières lignes (aperçu)
print(df.head())
```

	id_ligne	id_commande	id_produit	quantite	prix_unitaire
0	1	4378	418	2	286.40
1	2	7088	81	5	1609.52
2	3	13473	348	1	625.34
3	4	8834	2	2	778.44
4	5	9423	477	3	1235.72

FIGURE 10 – Aperçu des 5 premières lignes de lignes$_c$*ommandes.csv*

Remarques importantes

- Les identifiants des clients et des produits (`id_client`, `id_produit`) référencent correctement les tables correspondantes.
- Les dates des commandes sont réparties sur une période de 3 ans (2022-2024).
- Le `montant_total` des commandes correspond à la somme des lignes de commandes.
- Les fichiers CSV sont prêts à être utilisés pour l'ETL avec Pentaho.

0.4 Importation des CSV dans MySQL

Les fichiers CSV générés (`clients.csv`, `produits.csv`, `commandes.csv`, `lignes_commandes.csv`) doivent maintenant être importés dans la base `ventes_oltp`.

0.4.1 Étapes d'importation

0.4.2 Vérification de l'importation

Après l'importation, il est recommandé de vérifier que les fichiers ont bien été chargés et que le nombre de lignes correspond aux données générées.

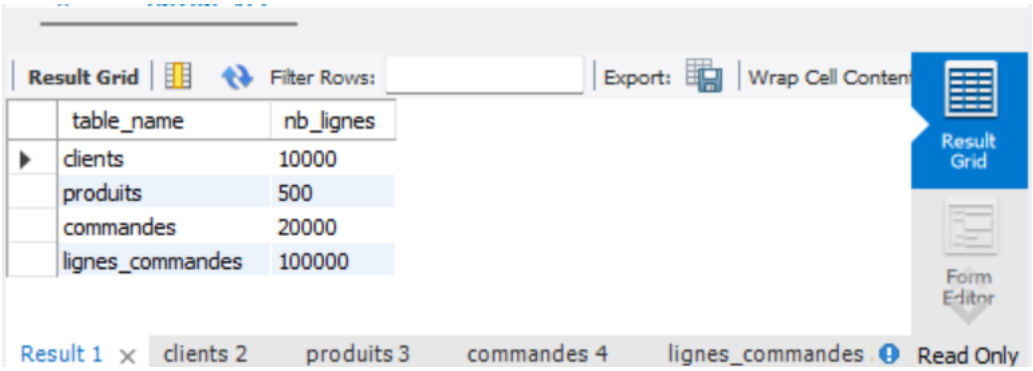


FIGURE 11 – Vérification de l'importation dans MySQL Workbench

0.5 Création du Data Warehouse

Nous avons maintenant créé la base de données `ventes_dwh` avec un schéma en étoile. Cette étape prépare les tables qui seront ensuite alimentées via Pentaho.

0.5.1 Tables créées

La base `ventes_dwh` contient les quatre tables suivantes :

— **DimClient** : dimension clients avec clé surrogate et clé source
— **DimProduit** : dimension produits avec clé surrogate et clé source
— **DimDate** : dimension date permettant l'analyse temporelle
— **FactVentes** : table de faits contenant les métriques de ventes

0.5.2 Vérification de la création des tables

Après l'exécution du script SQL, nous avons vérifié que les tables existaient et leur structure correspondait aux attentes.

Affichage des tables

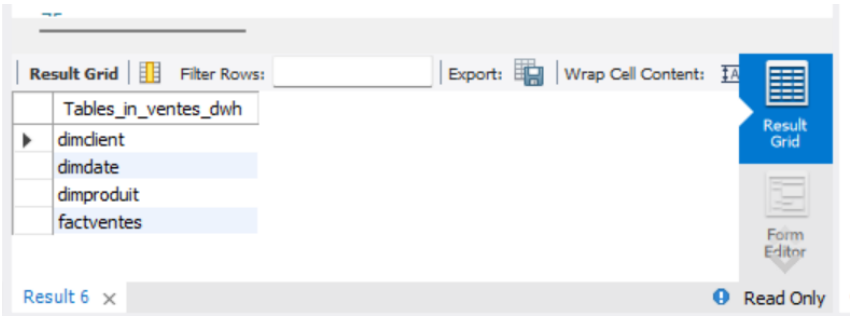


FIGURE 12 – Résultat de la commande `SHOW TABLES` dans ventes_dwh

Description des tables

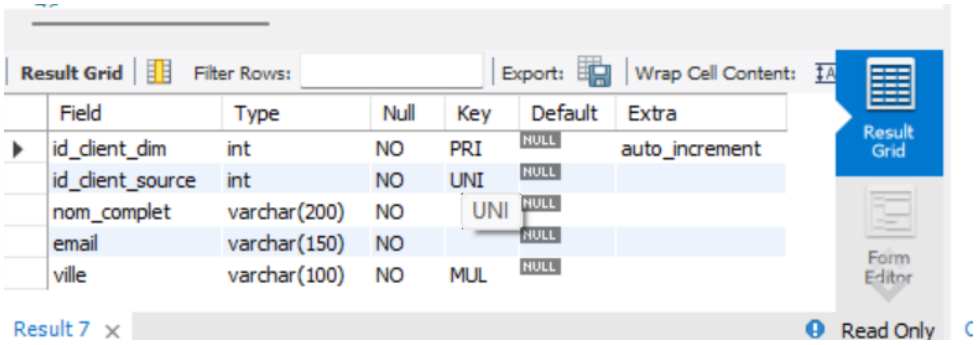


FIGURE 13 – Structure de la table DimClient

Table DimClient

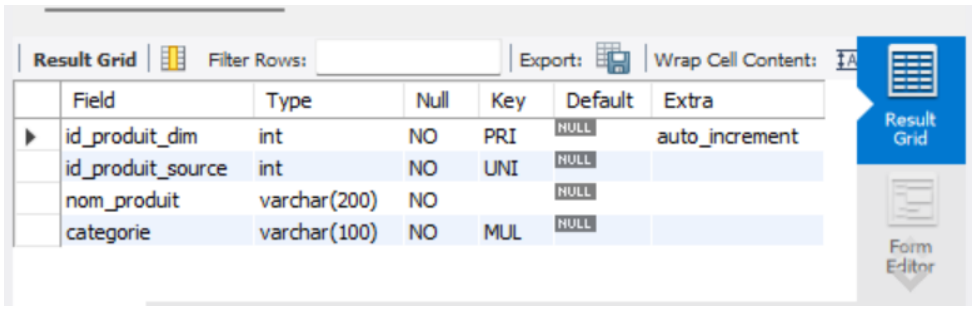


FIGURE 14 – Structure de la table DimProduit

Table DimProduit

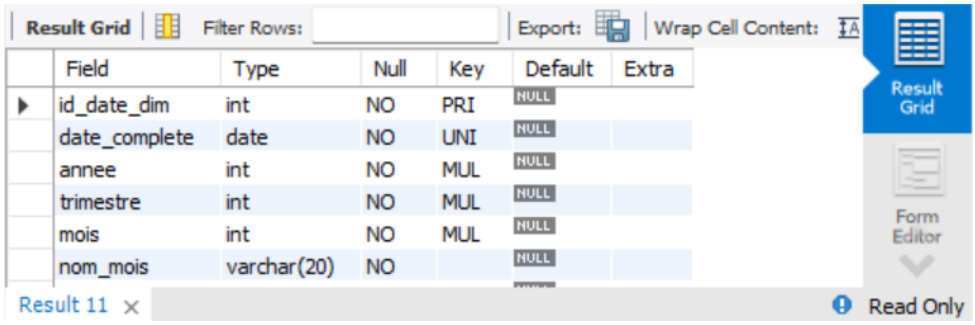


FIGURE 15 – Structure de la table DimDate

Table DimDate

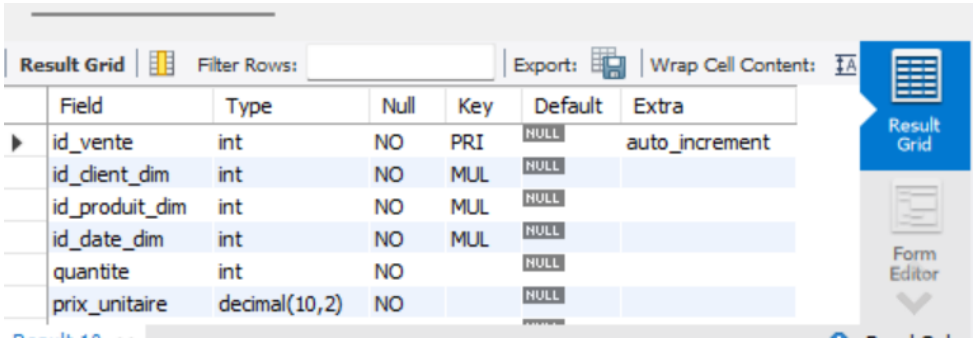


FIGURE 16 – Structure de la table FactVentes

Table FactVentes

0.5.3 Remarques

— Toutes les tables sont vides mais prêtes à recevoir les données transformées depuis l'OLTP via Pentaho PDI.

— Les clés surrogates permettent de gérer l'historique et les relations avec les faits.

— Les indices sont créés pour optimiser les requêtes analytiques dans le Data Warehouse.

0.6 Pentaho PDI – Configuration et Transformations

Cette partie présente la configuration initiale de Pentaho Data Integration (PDI) et la mise en place des connexions nécessaires entre la base OLTP et le Data Warehouse.

0.6.1 Configuration de Pentaho PDI

Avant de commencer les transformations, nous devons configurer les connexions aux deux bases MySQL. Après avoir installé Pentaho Data Integration, nous lançons l'outil **Spoon**.

0.6.2 Vérification des connexions

Après validation, les deux connexions apparaissent dans la liste *Database connections*. La figure suivante illustre l'écran de configuration dans PDI.

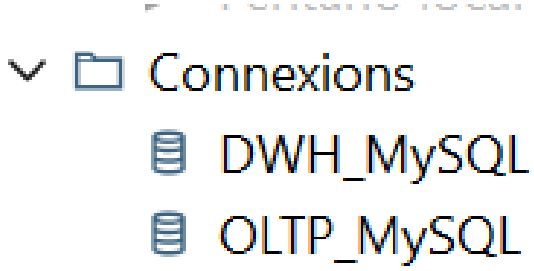


FIGURE 17 – Vérification des connexions OLTP et DWH dans Pentaho PDI

0.7 Transformation 1 : Chargement de la dimension `DimClient`

Cette première transformation a pour objectif d'extraire les données clients depuis la base OLTP, de renommer certains champs, puis de charger le résultat dans la table `DimClient` du Data Warehouse.

0.7.1 Création de la transformation

Une nouvelle transformation `dim_client.ktr` a été créée dans Pentaho PDI. La figure ci-dessous illustre l'architecture générale composée de trois étapes principales :

- **Table Input** : extraction des clients depuis la base OLTP.
- **Select Values** : renommage et sélection des colonnes nécessaires.
- **Table Output** : chargement des données dans la table `DimClient` du DWH.

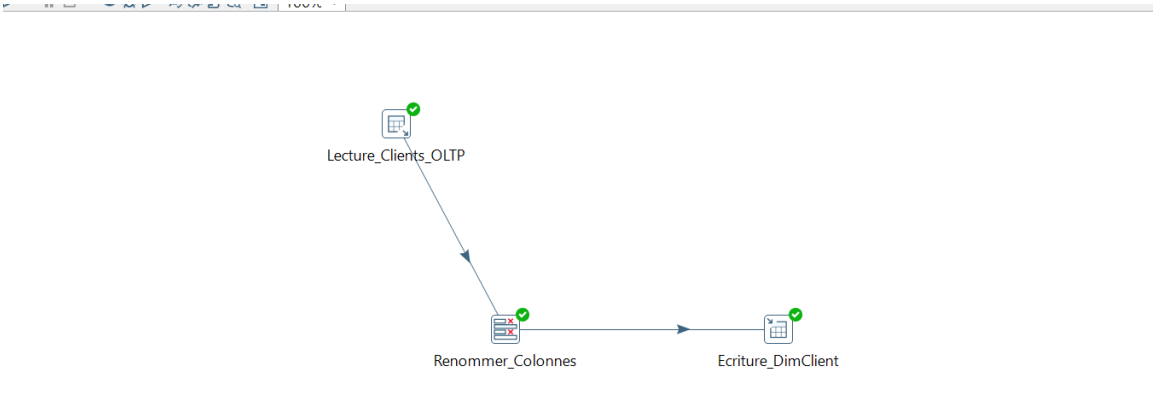


FIGURE 18 – Architecture de la transformation DimClient dans Pentaho PDI

0.7.2 Extraction et chargement

L'étape **Table Input** utilise la connexion `OLTP_MySQL` avec la requête SQL :

[language=SQL] SELECT id$_client, CONCAT(prenom, '', nom) AS nom_complet, email, ville FROM c$

Les colonnes sont ensuite renommées via **Select Values**, puis les données sont insérées dans la table `DimClient` à l'aide de l'étape **Table Output**, reliée au schéma `DWH_MySQL`.

0.7.3 Résultat obtenu

Après exécution de la transformation, 10 000 lignes sont chargées dans la dimension. La figure suivante montre la vérification dans MySQL Workbench :

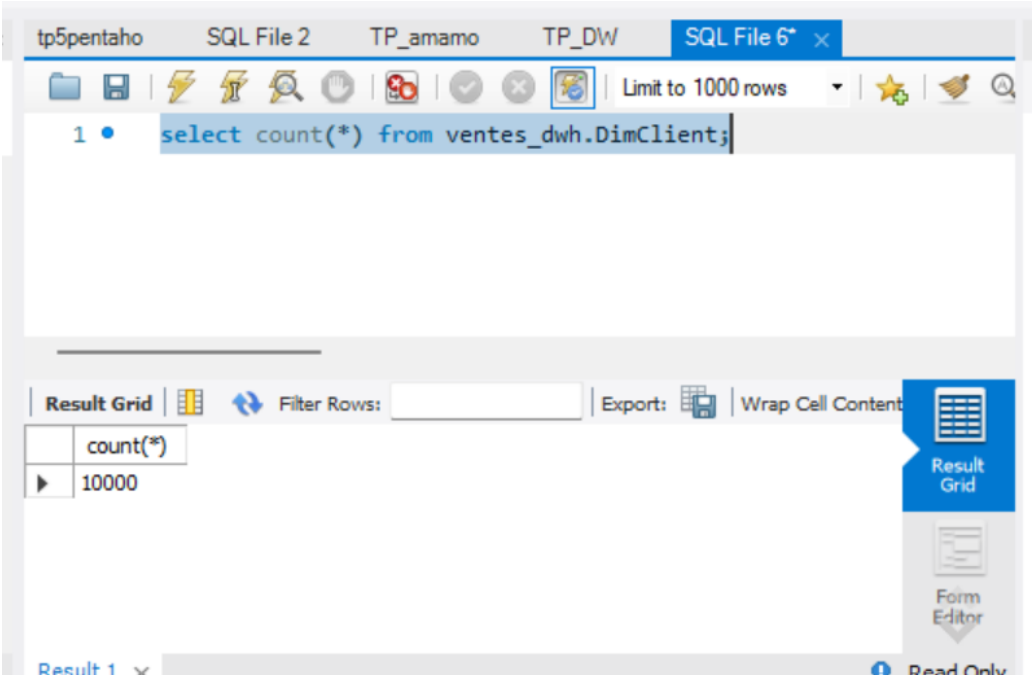


FIGURE 19 – Vérification du chargement de DimClient dans MySQL Workbench

La commande utilisée pour vérifier est :

[language=SQL] SELECT COUNT(*) FROM ventes$_dwh.DimClient$;

Le résultat obtenu est bien de **10 000 lignes**, ce qui confirme le bon fonctionnement de la transformation.

0.8 Transformation 2 : Chargement de la dimension `DimProduit`

Cette transformation a pour objectif d'extraire les données produits depuis la base OLTP, de renommer certains champs, puis de charger le résultat dans la table `DimProduit` du Data Warehouse.

0.8.1 Création de la transformation

Une nouvelle transformation `dim_produit.ktr` a été créée dans Pentaho PDI. La figure ci-dessous illustre l'architecture générale composée de trois étapes principales :

— **Table Input** : extraction des produits depuis la base OLTP.

— **Select Values** : renommage de la colonne `id_produit`.

— **Table Output** : chargement des données dans la table `DimProduit` du DWH.

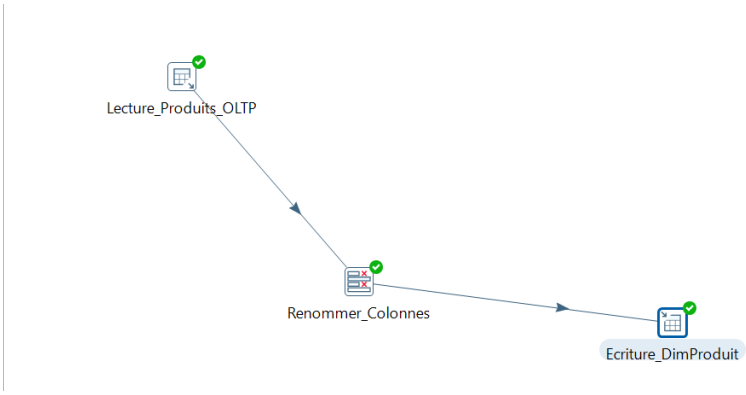


FIGURE 20 – Architecture de la transformation DimProduit dans Pentaho PDI

0.8.2 Extraction et renommage des colonnes

L'étape **Table Input** utilise la connexion `OLTP_MySQL` avec la requête SQL suivante :
[language=SQL] SELECT id$_produit, nom_produit, categorie FROM produits$;
Ensuite, l'étape **Select Values** permet de renommer la colonne :

— `id_produit` → `id_produit_source`

0.8.3 Chargement dans la dimension

Les données sont ensuite envoyées vers l'étape **Table Output**, configurée avec :

— Connexion : `DWH_MySQL`

— Table cible : `DimProduit`

— Option : **Truncate table** activée

0.8.4 Résultat obtenu

Après exécution, la transformation charge correctement les données dans la dimension. La figure suivante montre la vérification dans MySQL Workbench :

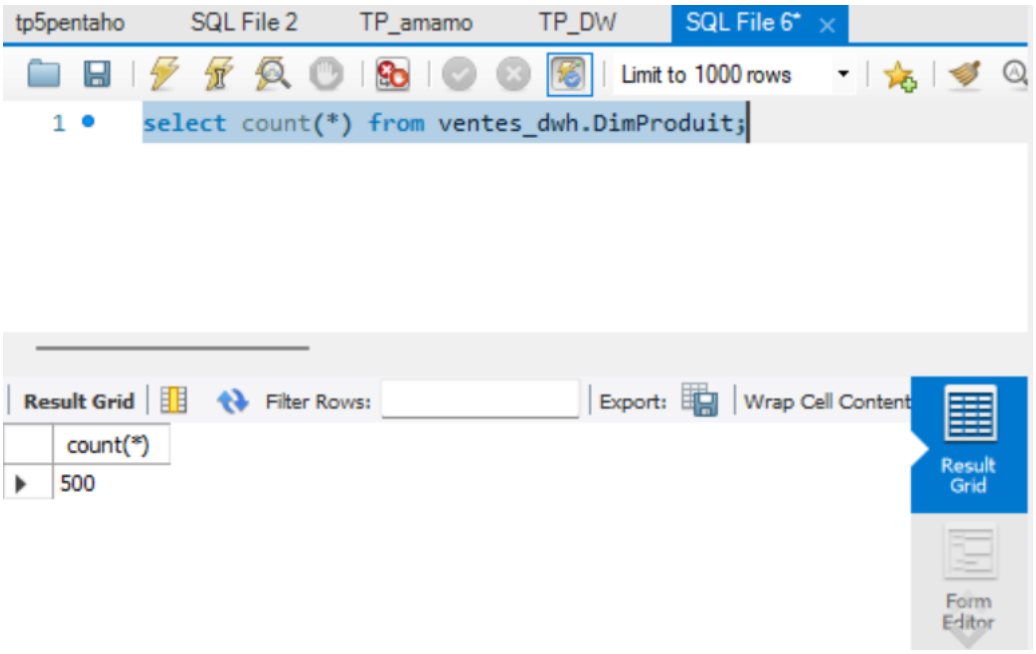


FIGURE 21 – Vérification du chargement de DimProduit dans MySQL Workbench

La commande SQL utilisée pour vérifier est :

[language=SQL] SELECT COUNT(*) FROM ventes$_dwh.DimProduit$;

Le résultat obtenu est de **500 lignes**, ce qui confirme que la transformation a fonctionné comme prévu.

0.9 Transformation 3 : Génération de la dimension `DimDate`

Cette transformation a pour objectif de générer artificiellement la dimension date, couvrant la période 2022-2024, car elle n'existe pas dans l'OLTP.

0.9.1 Création de la transformation

Une nouvelle transformation `dim_date.ktr` a été créée dans Pentaho PDI. La figure ci-dessous illustre l'architecture générale composée de quatre étapes principales :

- **Generate Rows** : création de 1096 lignes correspondant aux jours de 2022 à 2024.
- **Add Sequence** : génération d'un numéro de jour unique pour chaque ligne.
- **Modified Java Script Value** : calcul de la date complète et des composantes associées (année, mois, jour, trimestre, jour de la semaine, noms).
- **Table Output** : chargement des données dans la table `DimDate` du DWH.

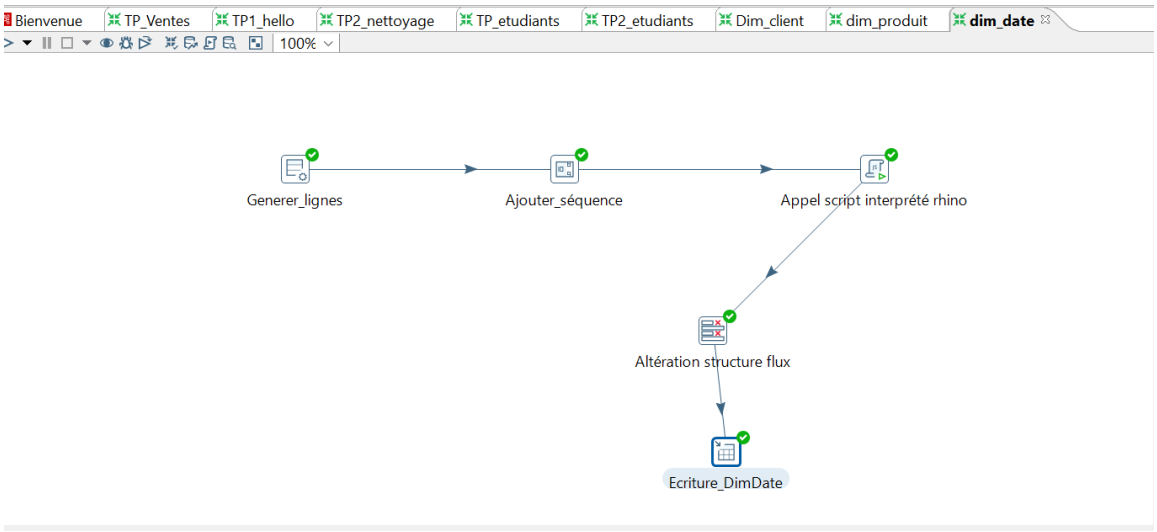


FIGURE 22 – Architecture de la transformation DimDate dans Pentaho PDI

0.9.2 Génération des dates et calcul des composantes

Le script JavaScript utilisé dans l'étape **Modified Java Script Value** est le suivant :

0.9.3 Chargement dans la dimension

L'étape **Table Output** est configurée avec :

— Connexion : `DWH_MySQL`

— Table cible : `DimDate`

— Option : **Truncate table** activée

0.9.4 Résultat obtenu

Après exécution, la transformation charge correctement les 1096 dates dans la dimension. La figure suivante montre la vérification dans MySQL Workbench :

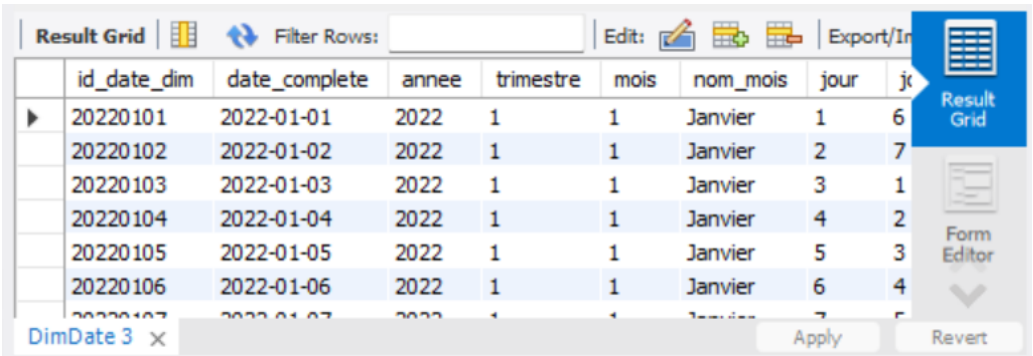


FIGURE 23 – Vérification du chargement de DimDate dans MySQL Workbench

La commande SQL utilisée pour vérifier est :

[language=SQL] SELECT * FROM ventes$_d wh.DimDateORDERBY\,date_c ompleteLIMIT10;

0.10 Transformation 4 : Chargement de la table de faits `FactVentes`

Cette transformation crée la table de faits en effectuant des jointures entre les données OLTP et les dimensions du DWH.

0.10.1 Création de la transformation

Une nouvelle transformation `fact_ventes.ktr` a été créée dans Pentaho PDI. L'architecture générale est illustrée ci-dessous :

- **Table Input** : extraction des lignes de commandes depuis l'OLTP.
- **Database Lookup** : récupération des identifiants des dimensions `DimClient`, `DimProduit` et `DimDate`.
- **Modified Java Script Value** : calcul du montant total.
- **Select Values** : sélection des champs finaux.
- **Table Output** : chargement des données dans la table `FactVentes`.

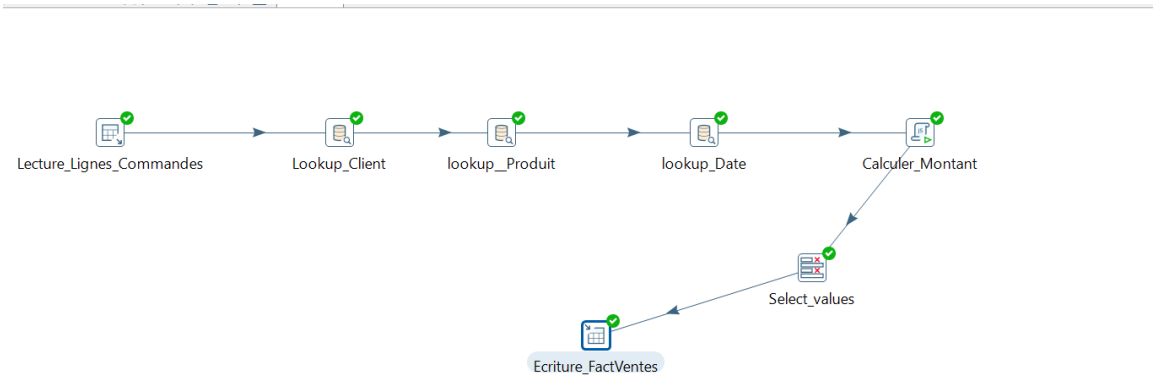


FIGURE 24 – Architecture de la transformation FactVentes dans Pentaho PDI

0.10.2 Extraction des données OLTP

Le step **Table Input** utilise la connexion `OLTP_MySQL` avec la requête SQL suivante :

[language=SQL] SELECT lc.id$_l igne, lc.id_c ommande, lc.id_p roduit, lc.quantite, lc.prix_u nitaire, c.id_c li
c.id_c ommande;

0.10.3 Lookup des dimensions

- **Lookup Client** : récupération de `id_client_dim` depuis `DimClient`.

— **Lookup Produit** : récupération de `id_produit_dim` depuis `DimProduit`.

— **Lookup Date** : récupération de `id_date_dim` depuis `DimDate`.

0.10.4 Calcul du montant total

Le step **Modified Java Script Value** effectue le calcul :

[language=JavaScript] var montant$_t$$otal = quantite * prix_u nitaire$;

0.10.5 Sélection des champs finaux et chargement

Les champs sélectionnés dans **Select Values** sont :

— `id_client_dim`

— `id_produit_dim`

— `id_date_dim`

— `quantite`

— `prix_unitaire`

— `montant_total`

L'étape **Table Output** est configurée avec :

— Connexion : `DWH_MySQL`

— Table cible : `FactVentes`

— Option : **Truncate table** activée

— Commit size : 1000

0.10.6 Résultat obtenu

Après exécution, la transformation charge correctement les 100 000 lignes dans la table de faits. La vérification peut se faire avec :

[language=SQL] SELECT COUNT(*) FROM ventes$_d$$wh.FactVentes$; Le résultat attendu est de **100 000 lignes**.

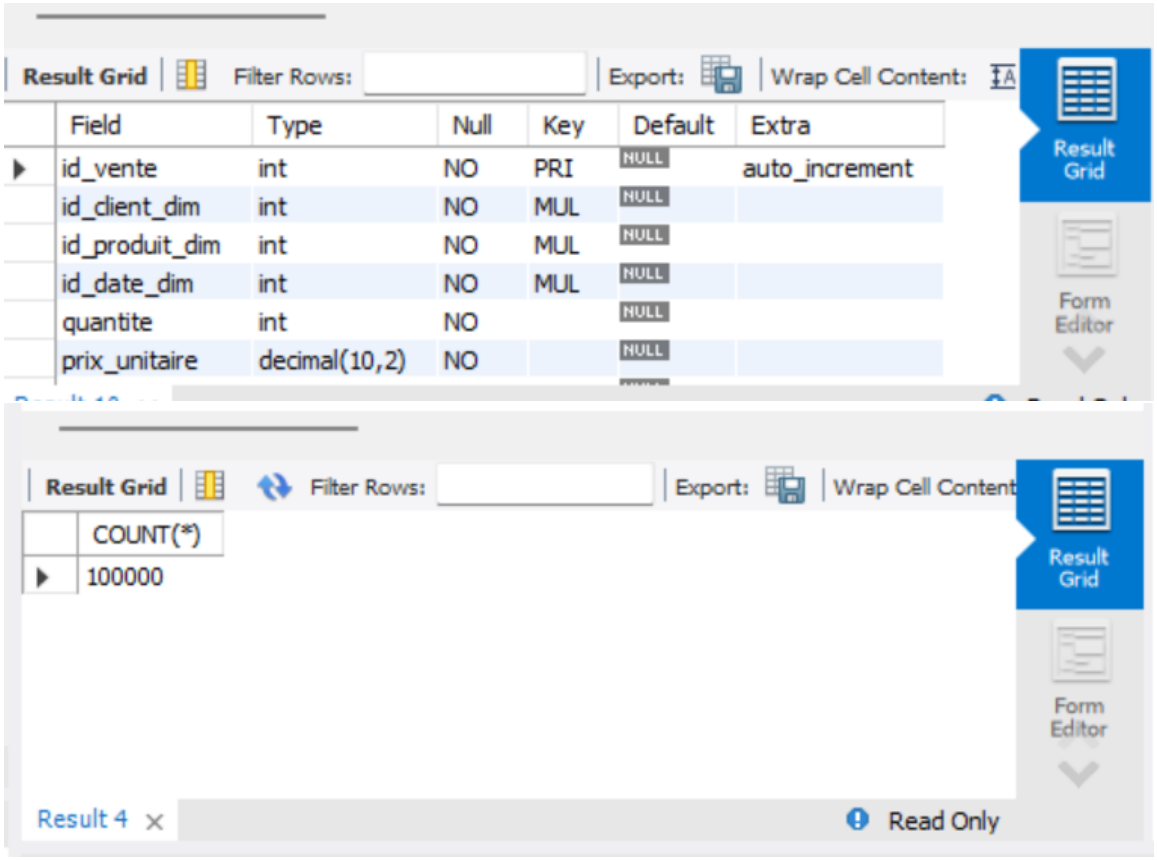


FIGURE 25 – Vérification du chargement de FactVentes dans MySQL Workbench

0.11 Création du Job d'Orchestration Pentaho

Après la création des quatre transformations (*DimClient*, *DimProduit*, *DimDate* et *FactVentes*), nous avons mis en place un Job Pentaho permettant d'orchestrer l'exécution complète du processus ETL. L'objectif d'un Job est d'exécuter les transformations dans le bon ordre, de gérer les dépendances et d'assurer un contrôle d'erreur global.

0.11.1 Structure du Job

Le Job complet suit la séquence suivante :

- **START**
- **Charger_DimClient**
- **Charger_DimProduit**
- **Charger_DimDate**
- **Charger_FactVentes**
- **Message_Succes**

Chaque transformation n'est exécutée que si la précédente s'est terminée avec succès. À la fin du processus, un message de validation est écrit dans les logs indiquant que le

Data Warehouse est à jour.

0.11.2 Étapes de création

1. Création d'un nouveau Job *job_ etl_ complet.kjb*.
2. Ajout de l'entrée **START**.
3. Ajout des transformations et configuration du chemin d'exécution conditionnel *(succes uniquement)*.
4. Ajout d'un message de fin pour confirmer l'exécution complète.

0.11.3 Capture d'écran du Job

La Figure ci-dessous présente la structure finale du Job d'orchestration :

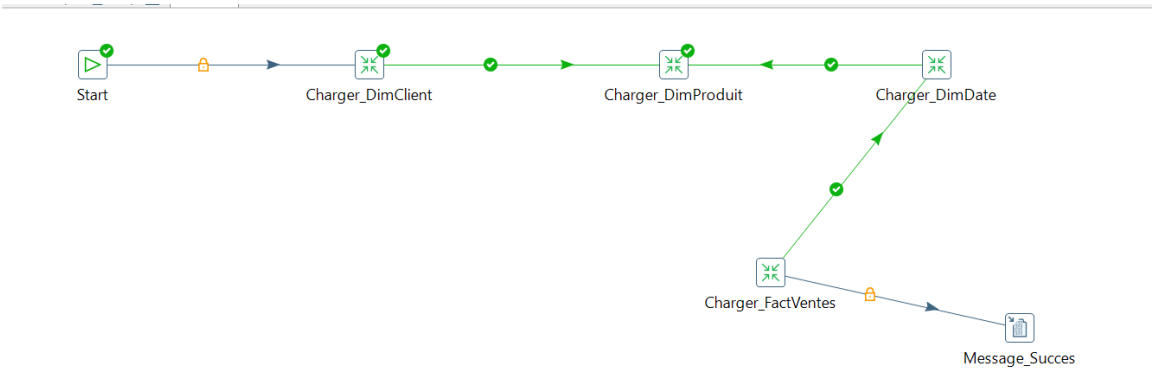


FIGURE 26 – Structure du Job d'orchestration Pentaho

0.12 Requêtes OLAP sur le Data Warehouse

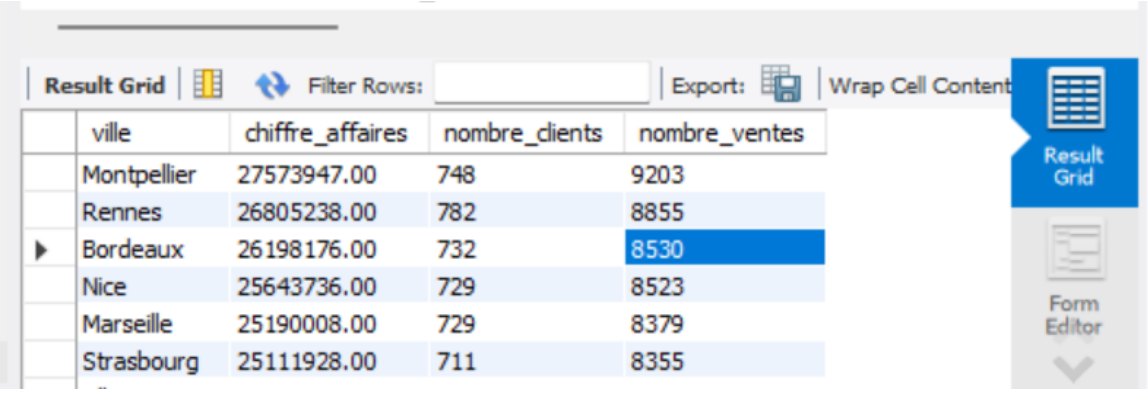
Après le chargement complet du Data Warehouse *ventes_ dwh*, nous avons exécuté plusieurs requêtes OLAP dans MySQL Workbench afin d'analyser les ventes selon différentes dimensions. Les requêtes ont été exécutées après sélection de la base :

```
USE ventes_dwh;
```

0.12.1 Requête 1 : Chiffre d'affaires par ville

Cette requête permet d'identifier les villes les plus rentables.

Interprétation : permet de repérer les zones géographiques générant le plus de chiffre d'affaires.



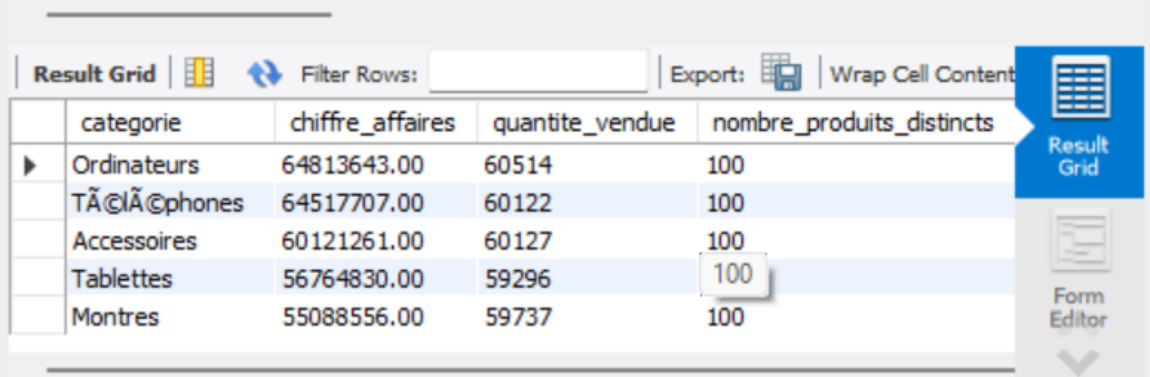
ville	chiffre_affaires	nombre_clients	nombre_ventes
Montpellier	27573947.00	748	9203
Rennes	26805238.00	782	8855
Bordeaux	26198176.00	732	8530
Nice	25643736.00	729	8523
Marseille	25190008.00	729	8379
Strasbourg	25111928.00	711	8355

FIGURE 27 – Résultat de la requête OLAP : CA par ville

0.12.2 Requête 2 : Chiffre d'affaires par catégorie de produit

Permet d'identifier les catégories les plus performantes.

Interprétation : permet d'identifier les gammes de produits les plus rentables.



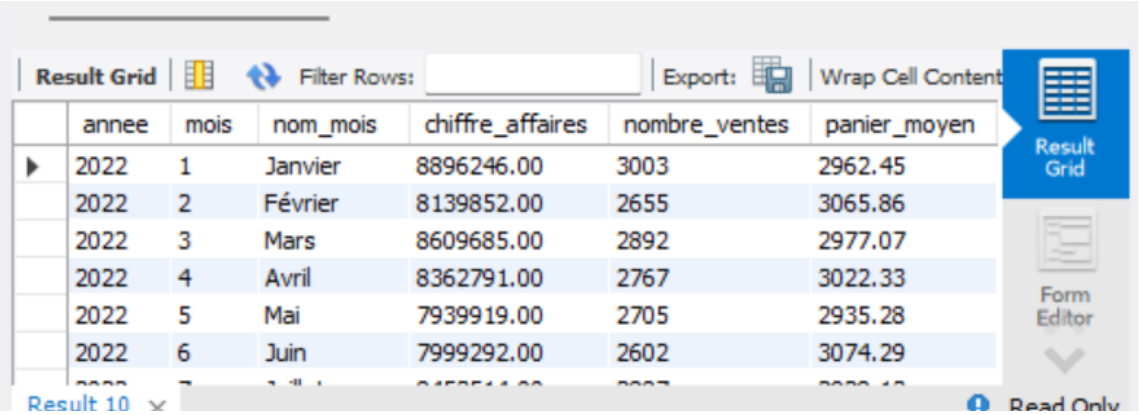
categorie	chiffre_affaires	quantite_vendue	nombre_produits_distincts
Ordinateurs	64813643.00	60514	100
TÃ©lÃ©phones	64517707.00	60122	100
Accessoires	60121261.00	60127	100
Tablettes	56764830.00	59296	100
Montres	55088556.00	59737	100

FIGURE 28 – Résultat : CA par catégorie

0.12.3 Requête 3 : Évolution mensuelle des ventes

Analyse de la saisonnalité et de la tendance mensuelle.

Interprétation : permet d'observer les hausses et baisses des ventes au fil de l'année.



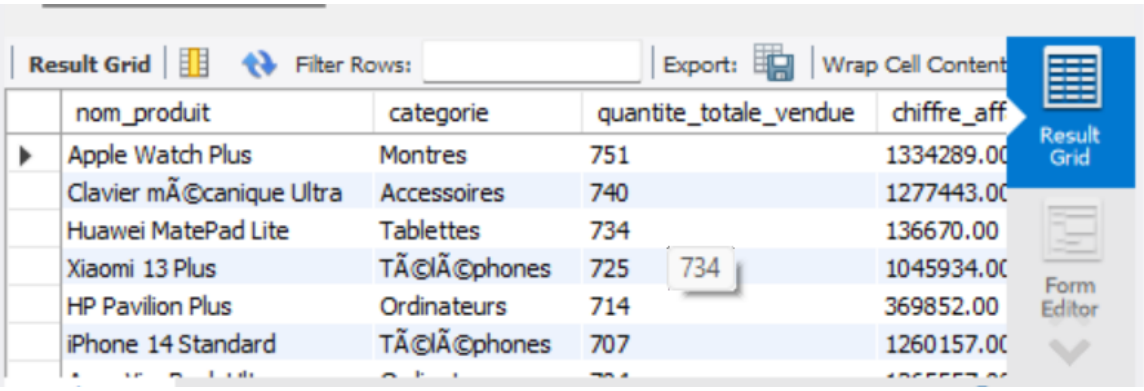
annee	mois	nom_mois	chiffre_affaires	nombre_ventes	panier_moyen
2022	1	Janvier	8896246.00	3003	2962.45
2022	2	Février	8139852.00	2655	3065.86
2022	3	Mars	8609685.00	2892	2977.07
2022	4	Avril	8362791.00	2767	3022.33
2022	5	Mai	7939919.00	2705	2935.28
2022	6	Juin	7999292.00	2602	3074.29

FIGURE 29 – Résultat : évolution mensuelle des ventes

0.12.4 Requête 4 : Top 10 des produits les plus vendus

Identifie les produits leaders.

Interprétation : permet d'identifier les produits essentiels à maintenir en stock.



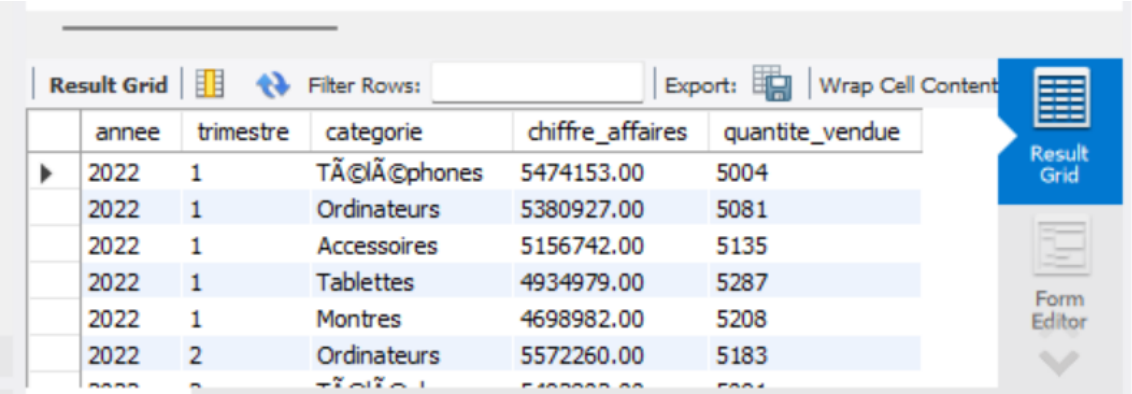
nom_produit	categorie	quantite_totale_vendue	chiffre_aff
Apple Watch Plus	Montres	751	1334289.00
Clavier mÃ©canique Ultra	Accessoires	740	1277443.00
Huawei MatePad Lite	Tablettes	734	136670.00
Xiaomi 13 Plus	TÃ©lÃ©phones	725	1045934.00
HP Pavilion Plus	Ordinateurs	714	369852.00
iPhone 14 Standard	TÃ©lÃ©phones	707	1260157.00

FIGURE 30 – Résultat : Top 10 des produits

0.12.5 Requête 5 : Analyse croisée trimestre–catégorie

Analyse multidimensionnelle croisant le temps et la catégorie.

Interprétation : montre quelles catégories fonctionnent le mieux selon les périodes de l'année.



annee	trimestre	categorie	chiffre_affaires	quantite_vendue
2022	1	TÃ©lÃ©phones	5474153.00	5004
2022	1	Ordinateurs	5380927.00	5081
2022	1	Accessoires	5156742.00	5135
2022	1	Tablettes	4934979.00	5287
2022	1	Montres	4698982.00	5208
2022	2	Ordinateurs	5572260.00	5183

FIGURE 31 – Résultat : analyse trimestre–catégorie

0.12.6 Avantages du schéma en étoile

— Requêtes rapides même sur de gros volumes.
— SQL simple (peu de jointures).
— Dimensions enrichies permettant une analyse flexible.
— Idéal pour l'analyse OLAP.

0.13 Power BI - Reporting et Visualisation

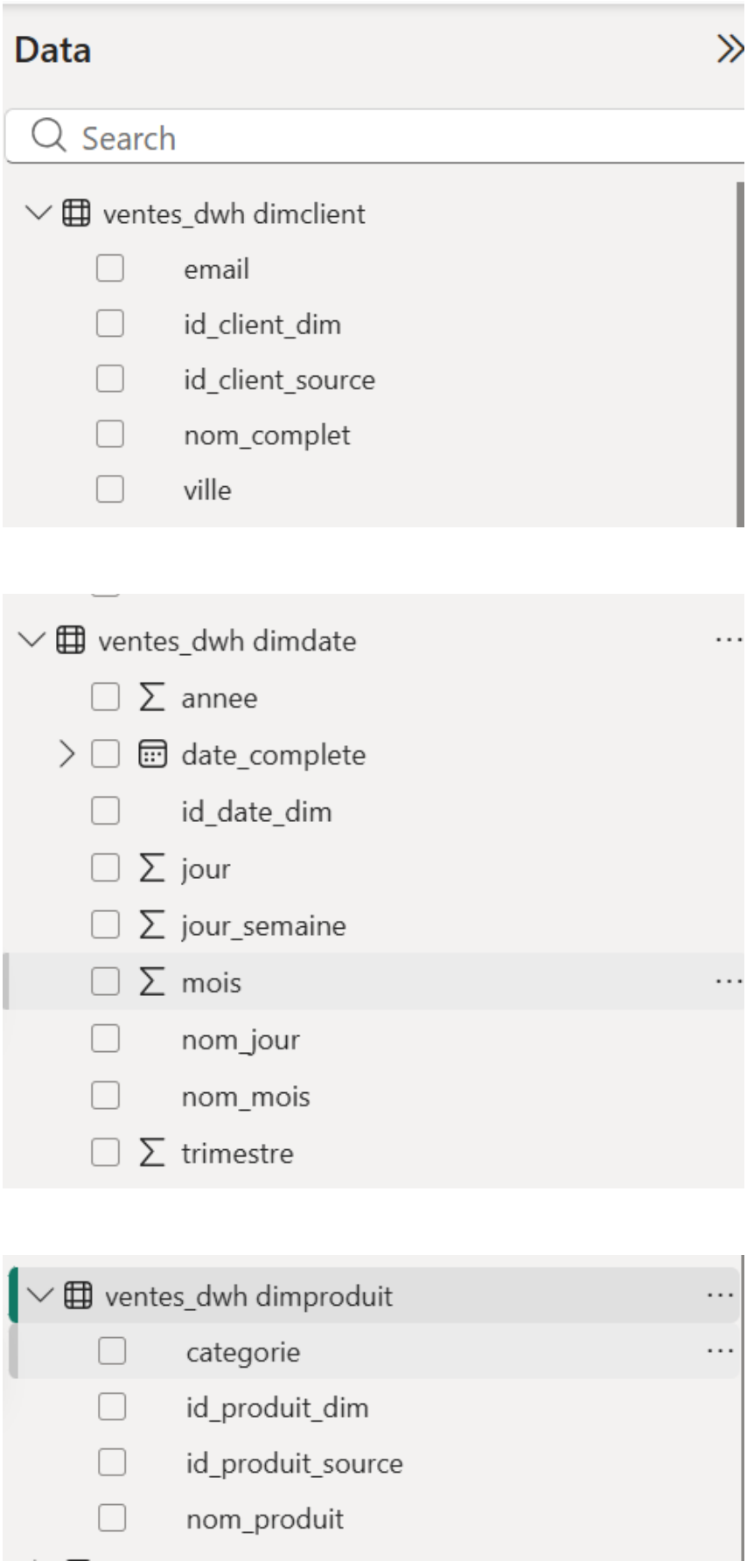
Nous allons créer un tableau de bord interactif dans Power BI pour visualiser nos données du Data Warehouse.

0.13.1 Prérequis

— Télécharger et installer Power BI Desktop (gratuit) depuis le site de Microsoft.
— Assurez-vous d'avoir le connecteur MySQL pour Power BI.

0.13.2 Étape 1 : Se connecter à MySQL depuis Power BI

1. **Lancer Power BI Desktop.**
2. **Obtenir les données :** MySQL Database.
3. **Configurer la connexion.** Remplissez :
   — Serveur : `localhost`
   — Base de données : `ventes_dwh`
   — Nom d'utilisateur : `root`
   — Mot de passe : votre mot de passe
   Cliquez sur **Connexion**.


Data
Search
ventes_dwh dimclient
email
id_client_dim
id_client_source
nom_complet
ville
ventes_dwh dimdate
annee
date_complete
id_date_dim
jour
jour_semaine
mois
nom_jour
nom_mois
trimestre
ventes_dwh dimproduit
categorie
id_produit_dim
id_produit_source
nom_produit

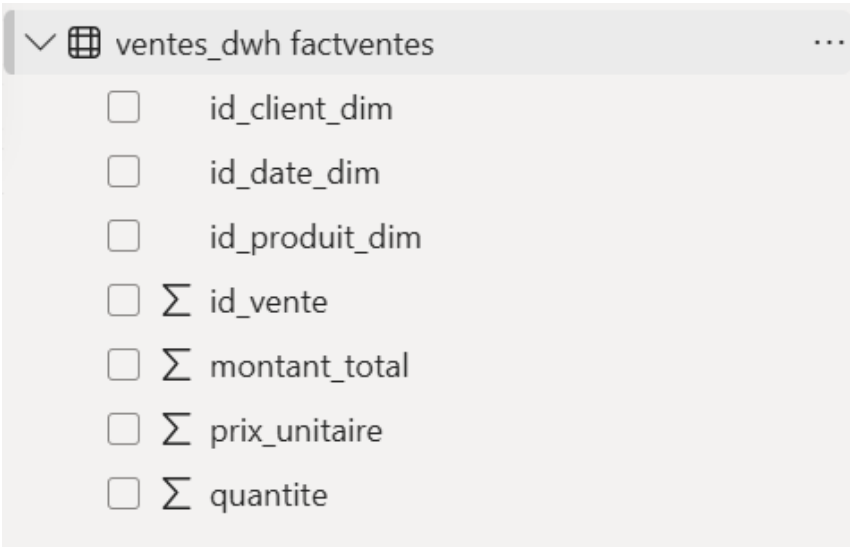


FIGURE 32 – Connexion à la base MySQL dans Power BI

0.13.3 Étape 2 : Importer les tables du Data Warehouse

1. Sélectionnez les tables `DimClient`, `DimProduit`, `DimDate`, `FactVentes` et cliquez sur **Charger**.

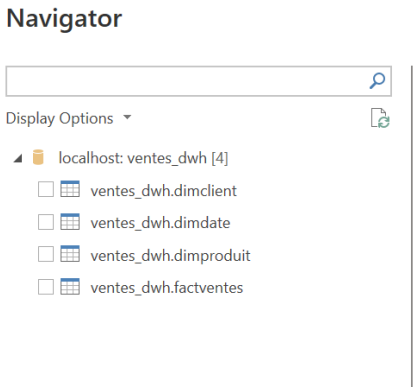


FIGURE 33 – Tables importées dans Power BI

0.13.4 Étape 3 : Vérifier et configurer le modèle de données

1. Ouvrir la vue Modèle.

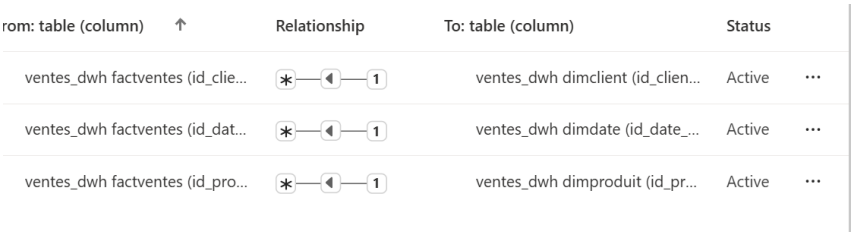


FIGURE 34 – Relations entre les tables dans Power BI

2. Vérifier les relations automatiques ou créer manuellement si nécessaire.

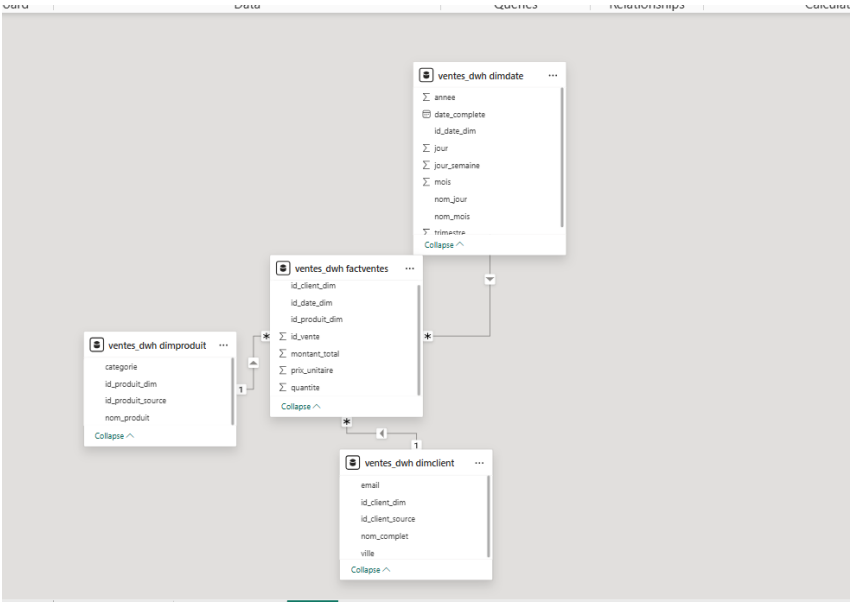


FIGURE 35 – Diagramme en étoile avec FactVentes au centre et dimensions autour

0.14 Étape 4 : Créer les visualisations

Revenons à la vue **Rapport** (icône graphique à gauche) pour créer nos visuels.

0.14.1 Visuel 1 : Chiffre d'affaires par ville (Graphique à barres)

1. Ajouter un graphique à barres groupées.
2. Configurer les champs :
   — Axe : `DimClient.ville`
   — Valeurs : `FactVentes.montant_total`
3. Personnaliser :
   — Titre : "Chiffre d'affaires par ville"
   — Activer les étiquettes de données
   — Modifier les couleurs si souhaité
4. Interprétation : montre quelles villes génèrent le plus de revenus.

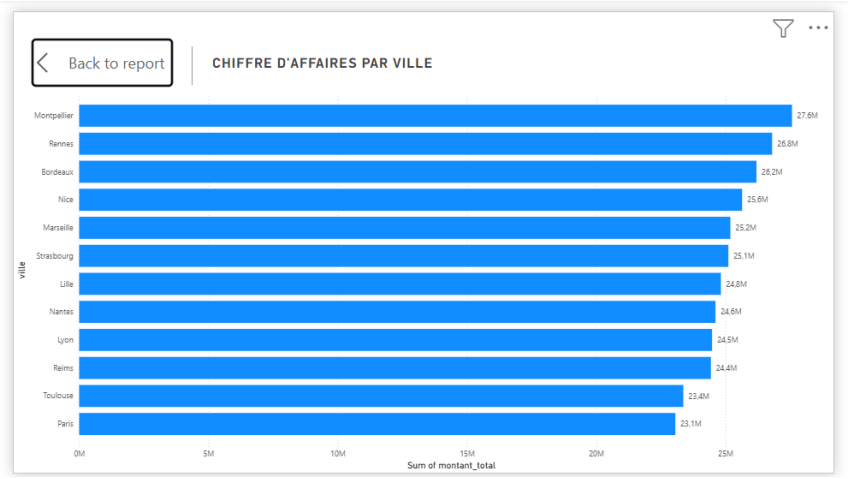


FIGURE 36 – Chiffre d'affaires par ville

0.14.2 Visuel 2 : Chiffre d'affaires par catégorie (Graphique en secteurs)

1. Ajouter un graphique en secteurs (Pie chart).
2. Configurer les champs :
   — Légende : `DimProduit.categorie`
   — Valeurs : `FactVentes.montant_total`
3. Personnaliser :
   — Titre : "Répartition du CA par catégorie"
   — Activer les étiquettes de détails
   — Choisir des couleurs différentes pour chaque catégorie
4. Interprétation : contribution relative de chaque catégorie au CA.

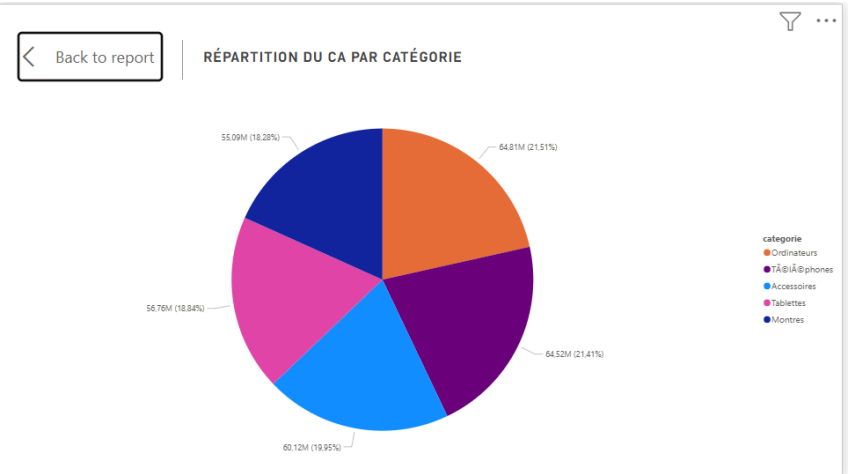


FIGURE 37 – Répartition du chiffre d'affaires par catégorie

0.14.3 Visuel 3 : Évolution des ventes mensuelles (Graphique en courbes)

1. Ajouter un graphique en courbes (Line chart).
2. Configurer les champs :
   — Axe X : `DimDate.date_complete`
   — Valeurs : `FactVentes.montant_total`
3. Hiérarchie temporelle : sélectionner Année, Trimestre ou Mois selon besoin.
4. Personnaliser :
   — Titre : "Évolution mensuelle du chiffre d'affaires"
   — Activer les marqueurs
   — Choisir une couleur distinctive (ex : bleu foncé)
5. Interprétation : montre la saisonnalité et les tendances.

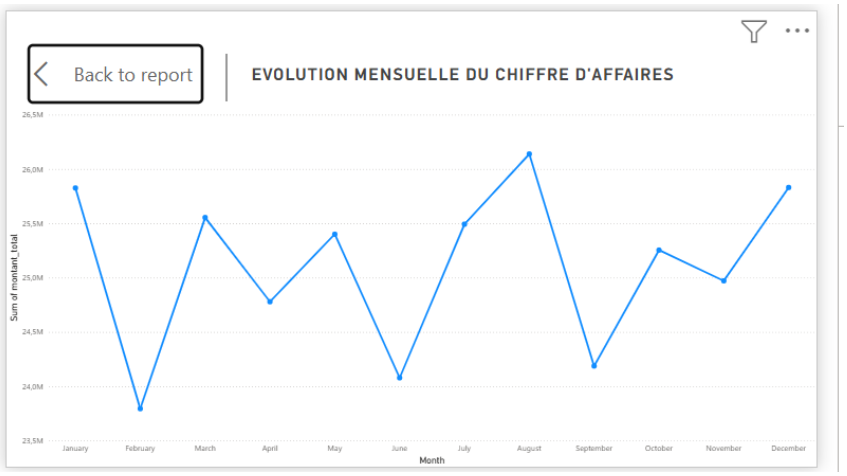


FIGURE 38 – Évolution mensuelle du chiffre d'affaires

0.14.4 Visuel 4 : Top 10 des produits (Graphique à barres horizontales)

1. Ajouter un graphique à barres horizontales.
2. Configurer les champs :
   — Axe Y : `DimProduit.nom_produit`
   — Valeurs : `FactVentes.quantite`
3. Limiter au Top 10 :
   — Dans Filtres, choisir Top N
   — Afficher les 10 premiers par Somme de `quantite`
4. Personnaliser :

— Titre : “Top 10 des produits les plus vendus”

— Trier les barres par ordre décroissant

— Activer les étiquettes de données

5. Interprétation : identifie les produits phares pour les promotions et mises en avant.

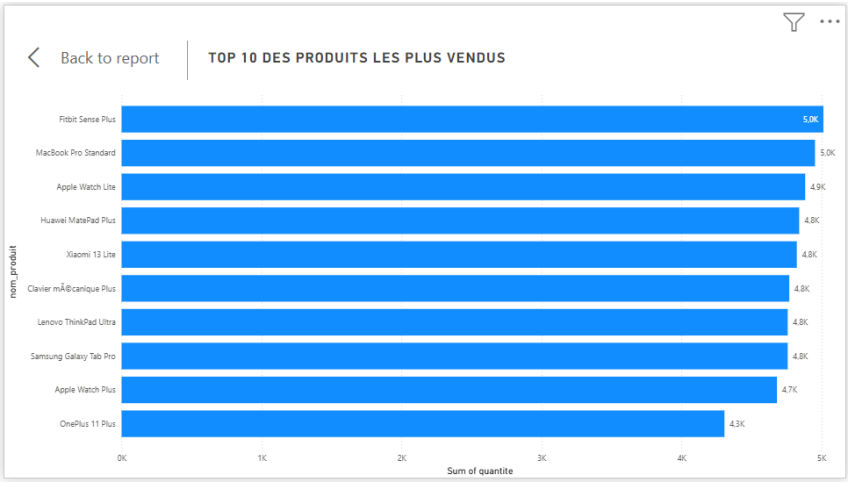


FIGURE 39 – Top 10 des produits les plus vendus

0.15 Étape 5 : Ajouter des segments (Slicers) pour l’interactivité

Les segments permettent de filtrer dynamiquement tous les visuels du rapport.

0.15.1 5.1 - Ajouter un segment Année

1. Cliquez sur une zone vide.
2. Sélectionnez l’icône **Segment (Slicer)** dans le panneau Visualisations.
3. Glissez `DimDate.annee` vers le champ du segment.
4. Positionnez ce segment en haut à gauche du rapport.

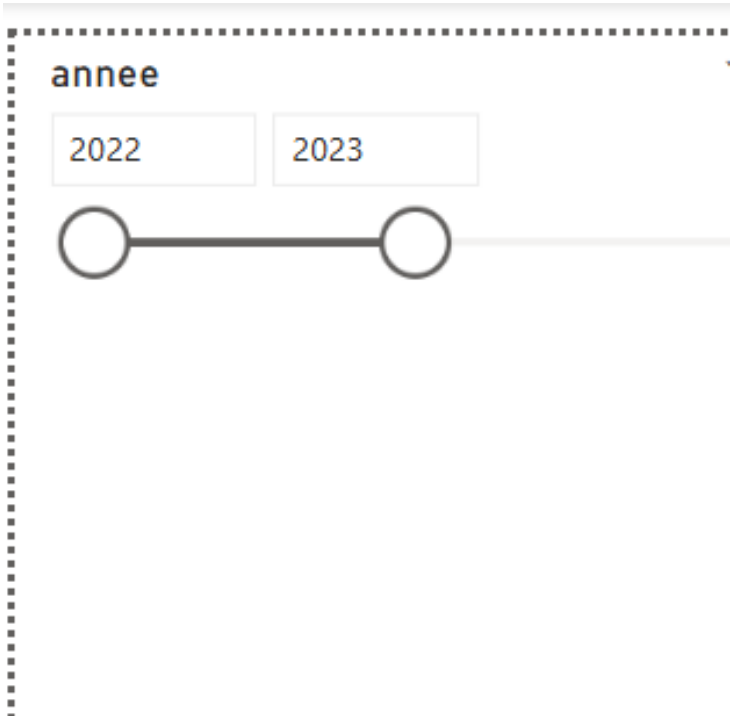


FIGURE 40 – Segment Année pour filtrer les visuels

0.15.2 5.2 - Ajouter un segment Catégorie

1. Ajoutez un autre segment.
2. Glissez `DimProduit.categorie` vers le champ du segment.
3. Positionnez-le à côté du segment Année.

FIGURE 41 – Segment Catégorie pour filtrer les visuels

0.15.3 5.3 - Tester l'interactivité

— Cliquer sur "2023" dans le segment Année : tous les visuels se mettent à jour pour afficher uniquement les données 2023.
— Cliquer sur "Ordinateurs" dans le segment Catégorie : les visuels se filtrent encore plus.
— Cliquer sur une barre du graphique "CA par ville" : tous les autres visuels se filtrent pour cette ville.

Note : Cette interactivité bidirectionnelle (cross-filtering) est l'une des forces majeures de Power BI. Elle permet aux utilisateurs d'explorer les données de manière intuitive sans écrire de SQL.

0.16 Étape 6 : Créer des mesures DAX (optionnel mais recommandé)

Pour enrichir l'analyse, nous créons quelques mesures calculées avec le langage DAX.

0.16.1 6.1 - Mesure "CA Total"

1. Dans le panneau **Champs**, clic droit sur `FactVentes` → **Nouvelle mesure**.
2. Tapez la formule :

$$CATotal = SUM(FactVentes[montant\_total])$$

3. Appuyez sur Entrée.

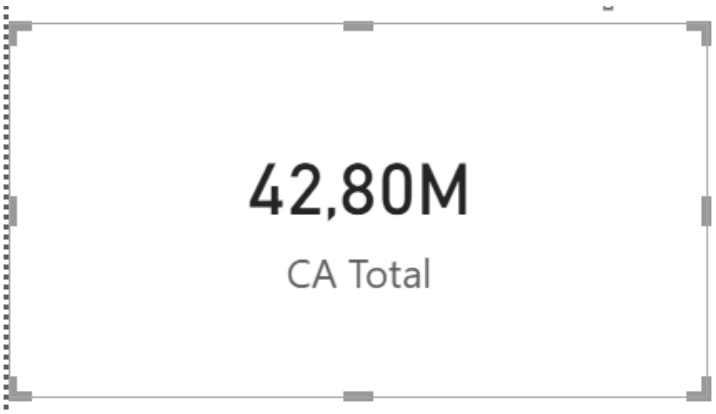


FIGURE 42 – Création de la mesure "CA Total"

0.16.2 6.2 - Mesure “Nombre de ventes”

1. Nouvelle mesure sur `FactVentes` :

$$NombredeVentes = COUNT(FactVentes[id\_vente])$$

FIGURE 43 – Création de la mesure “Nombre de Ventes”

0.16.3 6.3 - Mesure “Panier Moyen”

1. Nouvelle mesure :

$$PanierMoyen = DIVIDE([CATotal], [NombredeVentes], 0)$$

2. La fonction `DIVIDE` gère automatiquement les divisions par zéro.

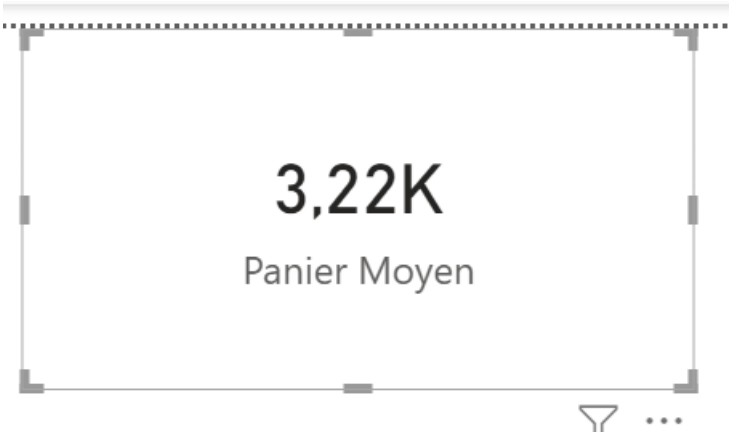


FIGURE 44 – Création de la mesure “Panier Moyen”

0.16.4 6.4 - Mesure “Clients Uniques”

1. Nouvelle mesure :

$$ClientsUniques = DISTINCTCOUNT(FactVentes[id\_client\_dim])$$

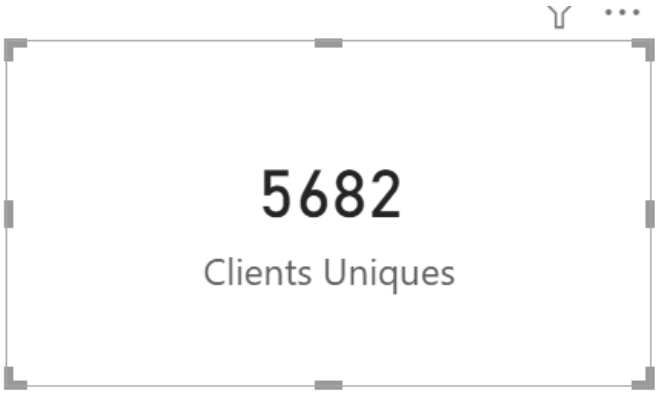


FIGURE 45 – Création de la mesure “Clients Uniques”

0.16.5 6.5 - Utiliser les mesures

— Créez des cartes (**Card**) pour afficher :
  — **CA Total**
  — **Panier Moyen**
  — **Nombre de Ventes**
  — **Clients Uniques**
— Positionnez ces 4 cartes en haut du rapport comme des KPI.

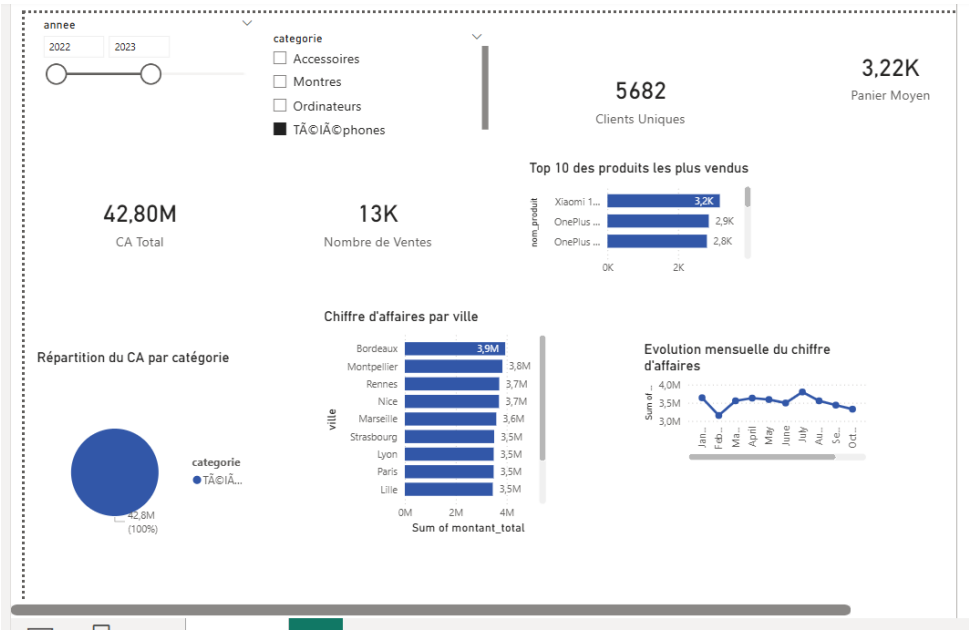


FIGURE 46 – Dashboard final : 4 KPI, 2 segments et 4 visualisations principales

Résultat attendu : un tableau de bord complet avec interactivité totale entre tous les visuels.

0.17 Étape 9 : Créer une page de détails (Drill-through)

Pour une analyse approfondie, nous créons une page de détails accessible en cliquant sur un élément.

0.17.1 9.1 - Créer une nouvelle page

1. En bas de l'écran, cliquez sur le + pour ajouter une page.
2. Renommez la page "Détails Produit".

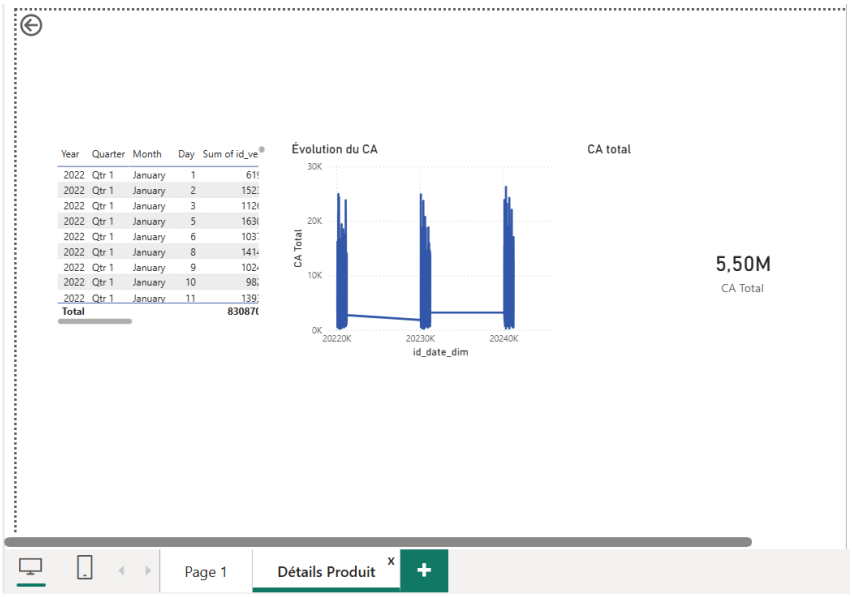


FIGURE 47 – Nouvelle page “Détails Produit”

0.17.2 9.3 - Ajouter des visuels détaillés

1. Tableau des ventes du produit sélectionné Ajoutez un tableau montrant toutes les ventes du produit sélectionné.

Year	Quarter	Month	Day	Sum of id_vente	Sum of montant_total	Sum of prix_unitaire	Sum of quantite	categorie
2022	Qtr 1	January	1	776656	28 781,00	15 961,20	28	Accessoires
2022	Qtr 1	January	1	1358052	67 900,00	24 503,61	79	Montres
2022	Qtr 1	January	1	961663	87 057,00	22 705,50	74	Ordinateurs
2022	Qtr 1	January	1	1093168	72 829,00	24 284,00	66	TÃ©lÃ©phones
2022	Qtr 1	January	1	1147539	65 861,00	23 149,66	63	Tablettes
2022	Qtr 1	January	2	796031	39 083,00	15 471,13	53	Accessoires
2022	Qtr 1	January	2	709063	42 251,00	12 090,94	56	Montres
2022	Qtr 1	January	2	741127	42 693,00	13 889,72	42	Ordinateurs
2022	Qtr 1	January	2	1513249	103 743,00	32 541,21	100	TÃ©lÃ©phones
2022	Qtr 1	January	2	1792067	93 281,00	28 541,45	87	Tablettes
2022	Qtr 1	January	3	583966	32 111,00	9 280,08	44	Accessoires
2022	Qtr 1	January	3	610851	35 971,00	15 863,69	35	Montres
2022	Qtr 1	January	3	815464	35 012,00	10 454,21	40	Ordinateurs
2022	Qtr 1	January	3	981427	42 556,00	12 834,88	46	TÃ©lÃ©phones
2022	Qtr 1	January	3	1121512	54 975,00	20 463,48	62	Tablettes
2022	Qtr 1	January	4	500186	32 576,00	10 454,06	38	Accessoires
2022	Qtr 1	January	4	475323	34 068,00	10 098,92	37	Montres
2022	Qtr 1	January	4	490207	47 562,00	12 948,76	34	Ordinateurs
Total				**5000050000**	**301 305 997,00**	**100 362 315,05**	**299796**	

FIGURE 48 – Tableau montrant toutes les ventes du produit sélectionné

2. Graphique en courbes de l’évolution des ventes Ajoutez un graphique en courbes montrant l’évolution des ventes de ce produit dans le temps.

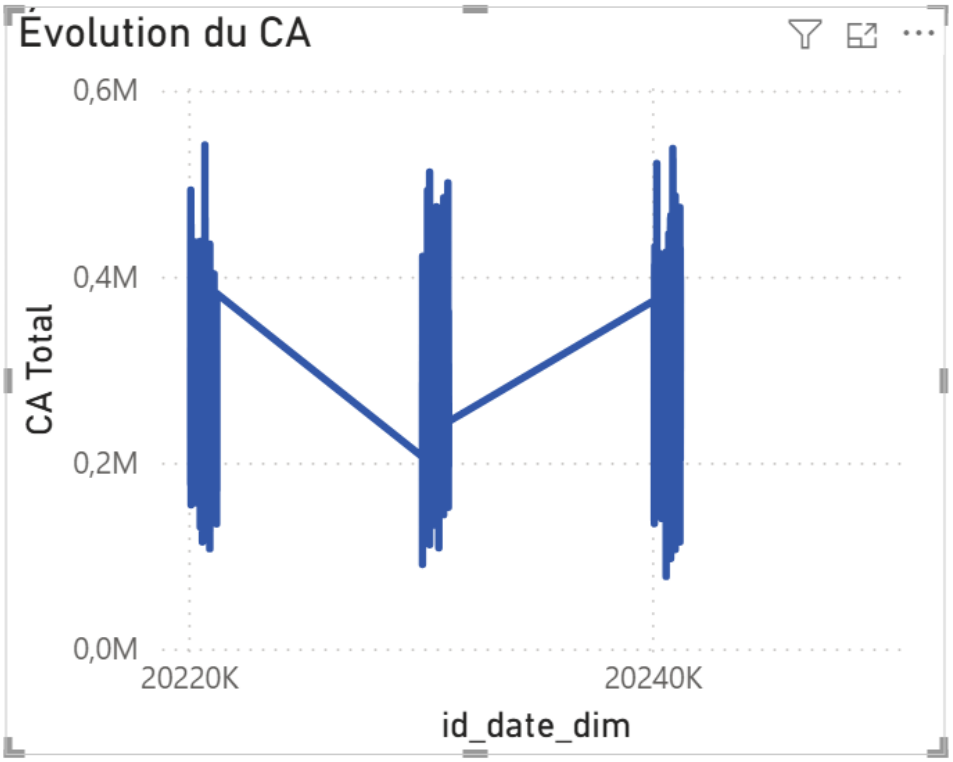


FIGURE 49 – Évolution des ventes du produit dans le temps

3. Carte du CA total du produit Ajoutez une carte montrant le chiffre d'affaires total du produit.

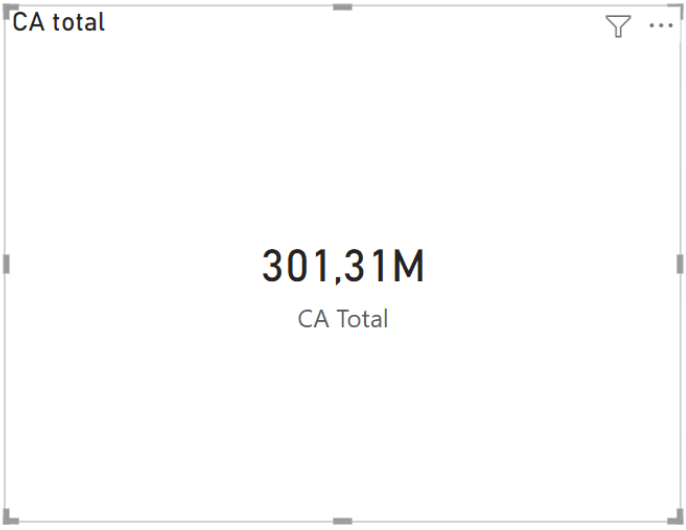


FIGURE 50 – Carte montrant le CA total du produit sélectionné

0.17.3 9.4 - Tester le drill-through

1. Revenir à la première page.
2. Clic droit sur une barre du graphique "Top 10 produits".

3. Sélectionner **Drill-through → Détails Produit**.
4. La page de détails s'affiche, filtrée sur le produit sélectionné.
5. Cliquez sur la flèche de retour en haut à gauche pour revenir.

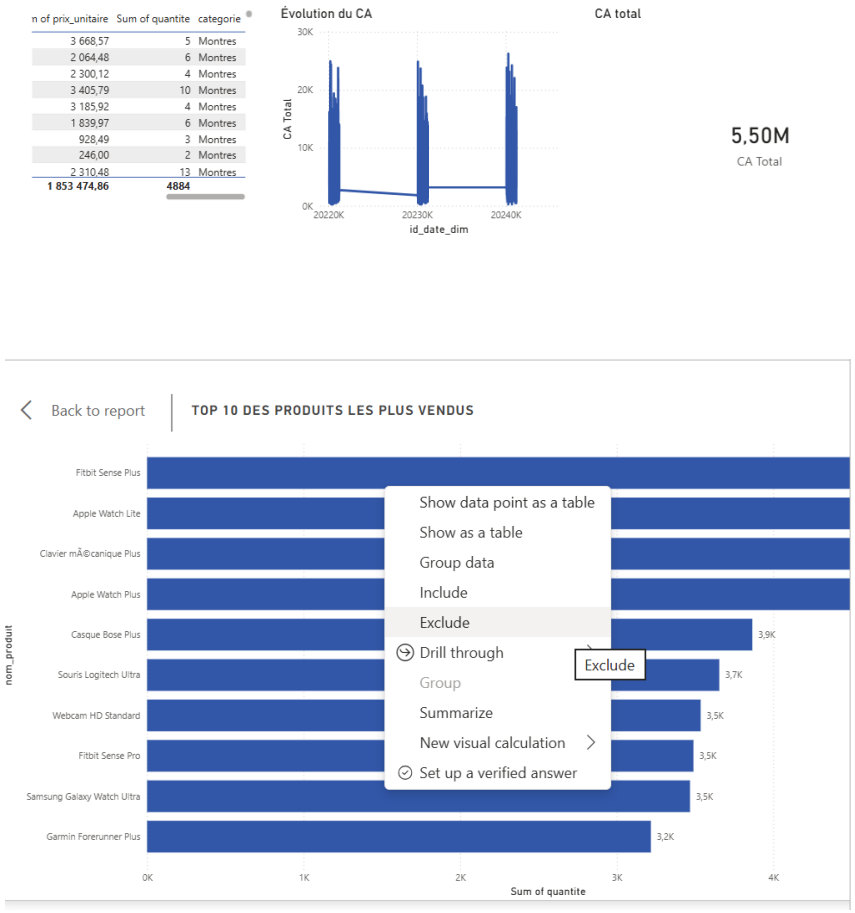


FIGURE 51 – Test du Drill-through : page filtrée sur le produit sélectionné

Remarque : Le drill-through permet aux utilisateurs d'explorer les données en profondeur sans surcharger le tableau de bord principal. C'est une fonctionnalité très appréciée par les analystes métier.