Rapport

Pipeline Complet de Business Intelligence : l'OLTP à Power BI

Réalisé par :

Mouj Salma

Formation
Big Data - 4ème année

le 27/11/2025

Table des matières

Résumé exécutif

Vue d'ensemble du projet
Ce projet avait pour objectif principal de mettre en place une chaîne complète de Business Intelligence (BI) moderne, allant de l'extraction des données opérationnelles jusqu'à la visualisation des analyses pour la prise de décision. Le travail a été réalisé dans le contexte de l'entreprise e-commerce fictive **TechStore**, spécialisée dans la vente de produits électroniques.
L'architecture adoptée repose sur le principe de la séparation des préoccupations (OLTP vs. OLAP), garantissant que l'analyse ne pénalise pas les performances du système transactionnel quotidien.

L'architecture technique déployée est la suivante :

1. **Source :** Base de données OLTP MySQL (`ventes_oltp`).
2. **Transformation :** Outil ETL Pentaho Data Integration (PDI).
3. **Cible :** Data Warehouse MySQL (`ventes_dwh`) modélisé en schéma en étoile.
4. **Reporting :** Outil de visualisation Power BI.

Le pipeline ETL a permis de traiter un volume significatif de données synthétiques, notamment 100 000 lignes de commandes, 20 000 commandes et 10 000 clients, générées pour simuler un environnement de production réaliste.

Objectifs du Projet
Les objectifs du projet étaient doubles : techniques et métier.

Objectifs Techniques

— Modéliser un Data Warehouse (`ventes_dwh`) utilisant le schéma en étoile, optimisé pour les requêtes d'agrégation OLAP.
— Concevoir et orchestrer le processus ETL complet via Pentaho PDI pour extraire les données de l'OLTP, les transformer (nettoyage, jointure, calculs) et les charger dans le DWH.
— Établir la connectivité entre le Data Warehouse et l'outil de reporting (Power BI).

Objectifs Métier (Problématique TechStore)
L'objectif fondamental était de fournir à la direction les outils nécessaires pour répondre à des questions analytiques clés, notamment :

— Identifier le Chiffre d'affaires (CA) par ville (performance géographique).
— Analyser les performances par catégorie de produit.
— Déterminer l'évolution et la saisonnalité des ventes (analyse temporelle).
— Identifier les produits stars (Top 10) et le comportement d'achat des clients.

Résultats Clés et Livrables
Le projet a abouti à la mise en service réussie de l'intégralité de la chaîne BI, fournissant des informations stratégiques immédiates :

1. **Data Warehouse Fonctionnel :** La base `ventes_dwh` a été créée et structurée avec succès en schéma en étoile (`FactVentes`, `DimClient`, `DimProduit`, `DimDate`). Les données transformées (100 000 faits de ventes) sont chargées de manière cohérente.
2. **Automatisation ETL :** Un Job d'orchestration Pentaho complet (`job_etl_complet.kjb`) a été créé et exécuté avec succès.
3. **Capacité Analytique OLAP :** Le Data Warehouse permet d'exécuter des requêtes OLAP complexes en temps record.

4. **Livrable Final : Tableau de Bord Interactif Power BI :** Un dashboard dynamique (`Dashboard_Ventes_TechStore.pbix`) affiche les KPI majeurs et des visualisations interactives permettant une exploration intuitive des données grâce aux segments (Slicers).

En conclusion, ce projet fournit à TechStore une plateforme analytique robuste et évolutive, transformant les données transactionnelles brutes en informations stratégiques exploitables pour orienter les décisions marketing et de gestion des stocks.

Introduction

Contexte de l'entreprise

Ce projet de Business Intelligence (BI) est réalisé dans le cadre de l'entreprise e-commerce fictive **TechStore**, spécialisée dans la vente en ligne de produits électroniques et informatiques.

L'entreprise utilise actuellement une base de données opérationnelle de type OLTP (*Online Transaction Processing*) pour gérer l'ensemble de ses activités quotidiennes. Ce système enregistre toutes les transactions courantes : inscriptions des clients, création de produits, passation et suivi des commandes. Conformément aux bonnes pratiques OLTP, cette base est hautement normalisée (forme normale 3FN) afin de garantir l'intégrité des données, d'éliminer la redondance et d'optimiser les opérations d'insertion, de mise à jour et de suppression (`INSERT`, `UPDATE`, `DELETE`). Elle est donc parfaitement adaptée à un volume élevé de transactions courtes et concurrentes.

Problématique métier

Bien que la base OLTP soit très performante pour les opérations quotidiennes, elle n'est pas conçue pour répondre efficacement aux besoins analytiques complexes de la direction. Son schéma fortement normalisé et ses indexes orientés transaction engendrent des requêtes d'agrégation (OLAP) très lentes, parfois impossibles à exécuter en temps acceptable sans dégrader les performances du système opérationnel.

Pour orienter sa stratégie commerciale, optimiser ses stocks et ses campagnes marketing, la direction de TechStore a besoin de répondre rapidement et de façon fiable à des questions analytiques stratégiques. L'absence actuelle d'un environnement dédié à l'analyse conduit à la problématique métier suivante :

Comment transformer les données transactionnelles brutes en insights stratégiques actionnables, sans impacter les performances du système opérationnel ?

Objectifs d'analyse spécifiques

Afin de résoudre cette problématique, le projet vise à construire un pipeline complet de Business Intelligence reposant sur un Data Warehouse dédié (OLAP). Les analyses ciblées portent sur quatre grands axes :

- **Performance géographique** : identifier le chiffre d'affaires réalisé par ville afin d'évaluer la répartition géographique des ventes.
- **Performance produit** : déterminer quelles catégories de produits génèrent le plus de revenus et identifier les références les plus rentables.
- **Saisonnalité et tendance** : analyser l'évolution des ventes mois par mois pour détecter les périodes de pic d'activité et les tendances saisonnières.
- **Comportement client** : repérer les 10 produits les plus vendus (produits stars), calculer le panier moyen et comprendre les habitudes d'achat des clients.

La solution retenue consiste à mettre en œuvre un **Data Warehouse** modélisé selon le **schéma en étoile**, couplé à un processus ETL robuste et à un outil de reporting moderne (Power BI).

1 Architecture technique

1.1 Diagramme de l'Architecture Globale du Projet

L'architecture déployée est un pipeline complet de Business Intelligence (BI) qui sépare rigoureusement l'environnement opérationnel (OLTP) de l'environnement analytique (OLAP). Cette chaîne garantit que les requêtes analytiques complexes ne perturbent pas les systèmes transactionnels quotidiens.

Le flux de données suit quatre étapes séquentielles :

Flux de Données	**Outil**	**Type d'Opération**
Extraction des données brutes	Base OLTP MySQL (`ventes_oltp`)	Transactionnel
Chargement des données transformées	Pentaho PDI (ETL)	Extract-Transform-Load
Connexion et analyse	Data Warehouse MySQL (`ventes_dwh`)	Schéma en Étoile / Analytique
Reporting	Power BI	Dashboards Interactifs

TABLE 1 – Pipeline de données du projet BI

Visualisation du Pipeline :

Extraction des données brutes → Transformation → Chargement des données transformées → Connexion et analyse

1.2 Technologies Utilisées

Le projet s'appuie sur quatre technologies principales, toutes éprouvées et spécifiques à leur rôle dans la chaîne BI :

1.2.1 Base de Données Opérationnelle (OLTP Source)

- **Technologie :** MySQL
- **Rôle :** Sert de source de données brutes (`ventes_oltp`). Cette base est optimisée pour l'enregistrement et la mise à jour rapide des transactions. Sa structure est hautement normalisée (3FN) pour assurer l'intégrité des données.

1.2.2 Outil ETL (Transformation)

- **Technologie :** Pentaho Data Integration (PDI), également connu sous le nom de Kettle.
- **Rôle :** Assure le processus Extract, Transform, Load (ETL). PDI est utilisé pour extraire les données de l'OLTP, les transformer (nettoyer, calculer le montant total de la ligne, joindre des informations et renommer les colonnes), puis les charger dans le Data Warehouse. Des Jobs d'orchestration ont été créés pour exécuter les transformations dans le bon ordre.

1.2.3 Base de Données Analytique (Data Warehouse Cible)

- **Technologie :** MySQL

— **Rôle :** Sert de Data Warehouse (`ventes_dwh`). Ce système est conçu pour l'analyse OLAP. Il est structuré en schéma en étoile (FactVentes au centre, reliée aux dimensions DimClient, DimProduit, DimDate) pour optimiser la performance des requêtes d'agrégation complexes.

1.2.4 Outil de Reporting et de Visualisation

— **Technologie :** Power BI (Microsoft)
— **Rôle :** Se connecte directement au Data Warehouse pour générer des rapports visuels dynamiques et des tableaux de bord interactifs. Power BI permet de répondre aux questions métier (CA par ville, saisonnalité) en utilisant les dimensions et les faits stockés.

1.3 Technologies de Support

Python (avec les librairies Faker et Pandas) : Utilisé en amont pour générer les jeux de données synthétiques réalistes (10 000 clients, 100 000 lignes de commandes) afin de simuler un environnement de production pour le TP.

2 Modélisation OLTP

Le système opérationnel de TechStore est basé sur une architecture OLTP (Online Transaction Processing). Ce type de base de données est optimisé pour les opérations quotidiennes rapides (INSERT, UPDATE, DELETE), telles que l'enregistrement d'une nouvelle commande ou la mise à jour d'un stock. Pour garantir la cohérence et l'intégrité des données, la structure est fortement normalisée (3FN), ce qui implique une séparation des données en plusieurs tables liées par des clés.

2.1 Description détaillée des tables OLTP

La base de données opérationnelle (`ventes_oltp`) contient quatre tables principales nécessaires à l'enregistrement des transactions de vente :

Table	Contenu	Colonne Clé Primaire (PK)	Colonnes Clés Étrangères (FK)	Détails Importants
clients	Informations sur les clients inscrits.	`id_client` (INT)	Aucune	Contient également `email` (unique), `ville`, et `date_inscription`.
produits	Catalogue des produits vendus.	`id_produit` (INT)	Aucune	Contient `nom_produit`, `categorie`, et `prix_unitaire`.
commandes	En-tête de chaque commande.	`id_commande` (INT)	`id_client`	Assure la traçabilité de la commande à un unique client.
lignes_commandes	Détail des produits commandés au sein d'une commande.	`id_ligne` (INT)	`id_commande`, `id_produit`	Contient la `quantite` et le `prix_unitaire` au moment de la commande.

TABLE 2 – Description des tables OLTP

2.2 Diagramme de Relations d'Entités (ERD)

Le modèle OLTP est relationnel et met en évidence la normalisation et les relations de type un-à-plusieurs (1 :N).

Le diagramme montre les relations suivantes :

— **CLIENTS (1) → (0..N) COMMANDES :** Un client peut passer zéro ou plusieurs commandes. Chaque commande appartient à un seul client.

— **COMMANDES (1) → (1..N) LIGNES_COMMANDES :** Une commande contient au moins une ligne de commande. Chaque ligne appartient à une seule commande.

— **PRODUITS (1) → (0..N) LIGNES_COMMANDES :** Un produit peut se trouver dans zéro ou plusieurs lignes de commande. Chaque ligne concerne un seul produit.

Structure Visuelle (Représentation de l'ERD) :

CLIENTS (`id_client` PK)	**COMMANDES** (`id_commande` PK)	**LIGNES_COMMANDES** (`id_ligne` PK)	**PRODUITS** (`id_produit` PK)
→ 1 :N →	`id_client` (FK) → 1 :N →	`id_commande` (FK) ← N :1 ← `id_produit` (FK)	← N :1 ←

TABLE 3 – Structure visuelle des relations ERD

2.3 Justification des choix de modélisation OLTP

La modélisation de la base `ventes_oltp` a été réalisée en privilégiant la normalisation (3FN). Ce choix est essentiel pour l'environnement transactionnel :

1. **Intégrité des Données :** La séparation des tables garantit que chaque donnée n'est stockée qu'une seule fois. Par exemple, les informations d'une ville (`ville`) n'existent

que dans la table `clients`. Si un client change de ville, la modification est effectuée à un seul endroit, assurant la cohérence.

2. **Performance Opérationnelle :** L'architecture est optimisée pour des requêtes courtes et des écritures rapides. Lorsqu'une nouvelle ligne de commande est insérée, il n'est pas nécessaire de dupliquer les informations du client ou du produit, car elles sont déjà référencées via les clés étrangères (`id_client`, `id_produit`).
3. **Flexibilité :** L'utilisation de clés étrangères garantit que les relations entre les entités (client, commande, produit) sont respectées, ce qui est crucial pour le bon déroulement des processus métier (par exemple, pour s'assurer qu'une commande référence un client existant).

Ce modèle est idéal pour l'enregistrement quotidien des transactions, mais, comme mentionné dans l'introduction, il n'est pas conçu pour l'analyse globale et requiert un processus ETL pour transférer et dénormaliser les données vers le Data Warehouse (DWH).

3 Génération des données

Afin de simuler un environnement de production réaliste pour TechStore et de valider l'efficacité du pipeline ETL sur un volume significatif, un jeu de données synthétiques a été généré.

3.1 Méthodologie et Outils Utilisés

La génération des données a été entièrement réalisée à l'aide du langage de programmation Python, en s'appuyant sur deux bibliothèques principales :

1. **Faker :** Utilisée pour créer des données réalistes mais fictives (noms, prénoms, adresses email, dates d'inscription).
2. **Pandas :** Utilisée pour structurer les données générées en DataFrames, effectuer des calculs de cohérence (comme le montant total des commandes), et exporter les résultats sous forme de fichiers CSV.

Le processus de génération garantit la cohérence des données : par exemple, le `montant_total` dans la table `commandes` est calculé en faisant la somme des montants des `lignes_commandes` correspondantes. Les identifiants clients et produits sont correctement référencés entre les différentes tables.

3.2 Volumes du Dataset Généré

Le jeu de données a été conçu pour être conséquent, permettant de tester le processus ETL sur une charge de travail significative. Les volumes de données générés pour la base OLTP (`ventes_oltp`) sont les suivants :

Table OLTP	Contenu	Volume Généré	Période de Temps
clients	Informations sur les clients inscrits.	10 000 lignes	Données générées.
produits	Catalogue des produits vendus.	500 lignes	100 produits pour chacune des 5 catégories.
commandes	En-tête des transactions.	20 000 lignes	Réparties entre le 1er janvier 2022 et le 31 décembre 2024.
lignes_ commandes	Détail des produits dans les commandes (faits).	100 000 lignes	Correspond à la source principale des faits pour le DWH.

TABLE 4 – Volumes du dataset généré

3.3 Répartition des Données et Cohérence

Les données générées présentent une répartition réaliste nécessaire à l'analyse métier :

1. **Répartition Géographique :** Les clients ont été répartis aléatoirement entre 12 grandes villes françaises (ex. : Paris, Lyon, Marseille, Toulouse), ce qui est essentiel pour l'analyse du chiffre d'affaires par ville (Requête 1).
2. **Répartition Catégorielle :** Les 500 produits sont divisés en cinq grandes catégories (Ordinateurs, Téléphones, Tablettes, Accessoires, Montres). Cela permet l'analyse des performances par catégorie (Requête 2).
3. **Répartition Temporelle :** Les 20 000 commandes sont réparties sur une période de trois ans (2022-2024). Cette répartition sur une période longue est cruciale pour l'analyse de la saisonnalité et des tendances temporelles (Requête 3).

L'ensemble de ces fichiers CSV (`clients.csv`, `produits.csv`, `commandes.csv`, `lignes_commandes.csv`) est ensuite importé dans la base de données source OLTP (`ventes_oltp`) via MySQL Workbench ou la commande `LOAD DATA LOCAL INFILE`.

4 Modélisation du Data Warehouse (DWH)

4.1 Le Schéma en Étoile (Star Schema)

La base de données analytique de TechStore, nommée `ventes_dwh`, a été conçue selon le modèle du Schéma en Étoile (Star Schema). Ce modèle de données est la structure dénormalisée privilégiée pour l'analyse OLAP (Online Analytical Processing).

Le Schéma en Étoile est composé d'une table de faits centrale reliée directement à plusieurs tables de dimensions, formant une structure simple et intuitive.

4.2 Description des Composants du DWH

Table	Type	Contenu	Rôle
FactVentes	Table de Faits	Contient les métriques quantitatives (quantité, prix unitaire, montant total).	Cœur du DWH, stocke les valeurs à agréger.
DimClient	Dimension	Contexte descriptif des clients (nom complet, email, ville).	Permet l'analyse des performances par client ou par région géographique.
DimProduit	Dimension	Contexte descriptif des produits (nom du produit, catégorie).	Permet l'analyse des performances par catégorie de produit.
DimDate	Dimension	Contexte temporel (année, trimestre, mois, nom du mois, jour).	Essentielle pour l'analyse de la saisonnalité et des tendances.

TABLE 5 – Composants du Data Warehouse

4.3 Relations et Clés

Toutes les dimensions (DimClient, DimProduit, DimDate) sont reliées à la table de faits FactVentes par des relations un-à-plusieurs (1 :N).

- Les dimensions utilisent des clés surrogates (clés artificielles auto-incrémentées comme `id_client_dim`) comme clés primaires pour garantir l'unicité et simplifier la gestion.
- La table FactVentes contient des clés étrangères (`id_client_dim`, `id_produit_dim`, `id_date_dim`) qui pointent vers les clés primaires des dimensions.
- La dimension DimDate est une table artificielle qui n'existe pas dans l'OLTP, mais qui est générée pour enrichir l'analyse temporelle (ex. : nom du mois, trimestre).

4.4 Justification et Avantages du Schéma en Étoile

Le choix du schéma en étoile est fondamental pour atteindre les objectifs analytiques de TechStore. Il offre plusieurs avantages cruciaux par rapport à la structure normalisée de l'OLTP :

1. **Performance des Requêtes (Vitesse) :** Le modèle réduit considérablement le nombre de jointures nécessaires pour interroger les données. Chaque requête analytique complexe (OLAP) ne nécessite généralement que 1 ou 2 jointures (entre FactVentes et les dimensions), ce qui rend les requêtes très rapides malgré les 100 000 lignes de faits.
2. **Simplicité et Compréhension :** La structure est intuitive pour les utilisateurs métier et les analystes. Elle facilite la construction des requêtes SQL et simplifie la navigation dans les données.
3. **Analyse Multidimensionnelle (OLAP) :** Le schéma en étoile est parfaitement adapté pour répondre aux questions analytiques de TechStore (CA par ville, CA par catégorie, évolution mensuelle). Les dimensions enrichissent naturellement l'analyse (ex. : les noms de mois, les catégories, les villes sont disponibles directement).
4. **Dénormalisation et Accélération :** Les tables de dimensions sont légèrement dénormalisées (par exemple, la colonne `ville` se trouve dans `DimClient`). Cette redondance intentionnelle dans les dimensions permet d'accélérer l'exécution des requêtes.
5. **Compatibilité avec les Outils BI :** Ce modèle est la structure optimale pour les outils de Business Intelligence comme Power BI, Tableau ou QlikView, facilitant grandement la connexion et la création de rapports visuels dynamiques.

5 Processus ETL avec Pentaho

Le processus ETL (Extract, Transform, Load) est la phase centrale du projet, assurant la migration et la préparation des données brutes de la base OLTP (`ventes_oltp`) vers le Data Warehouse (`ventes_dwh`). Cet ensemble d'opérations a été réalisé à l'aide de l'outil open-source Pentaho Data Integration (PDI).

Le processus total comprend quatre transformations (une par table du DWH) orchestrées par un Job unique.

5.1 Configuration des Connexions

Avant de démarrer les transformations, deux connexions à la base de données MySQL ont été configurées dans Pentaho Spoon :

1. **OLTP_MySQL :** Connexion à la base source `ventes_oltp`.
2. **DWH_MySQL :** Connexion à la base cible `ventes_dwh`.

5.2 Transformations des Dimensions

Les transformations des dimensions (DimClient, DimProduit, DimDate) doivent être exécutées en premier lieu, car la table de faits (FactVentes) dépend de leurs clés surrogates.

5.2.1 Transformation 1 : Charger DimClient (dim_client.ktr)

Objectif : Alimenter la table DimClient avec les informations des clients en y ajoutant le nom complet.

Étapes Clés	**Description et Justification**
Lecture_Clients_OLTP	`Table Input` : Extraction des données brutes de la table OLTP `clients`. La requête SQL effectue une concaténation pour créer la colonne `nom_complet` (prénom + nom).
Renommer_Colonnes	`Select values` : Le champ `id_client` est renommé en `id_client_source` pour distinguer la clé naturelle de la clé surrogate future.
Ecriture_DimClient	`Table Output` : Chargement des 10 000 lignes dans la table DimClient du DWH. L'option Truncate table est utilisée pour assurer la réinitialisation de la dimension avant chaque chargement complet.

TABLE 6 – Étapes de transformation DimClient

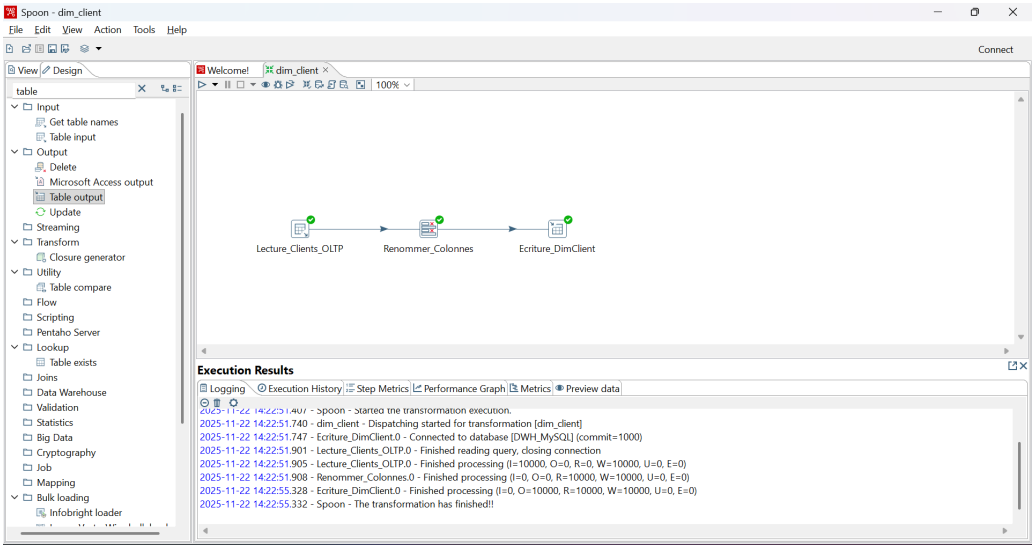

FIGURE 1 – Charger DimClient

5.2.2 Transformation 2 : Charger DimProduit (dim_produit.ktr)

Objectif : Alimenter la table DimProduit avec les 500 produits du catalogue OLTP.

Étapes Clés	Description et Justification
Lecture_Produits_OLTP	`Table Input` : Extraction de `id_produit`, `nom_produit` et `categorie` de la table OLTP `produits`.
Renommer_Colonnes	`Select values` : Le champ `id_produit` est renommé en `id_produit_source`.
Ecriture_DimProduit	`Table Output` : Chargement des 500 lignes dans DimProduit, après avoir vidé la table.

TABLE 7 – Étapes de transformation DimProduit

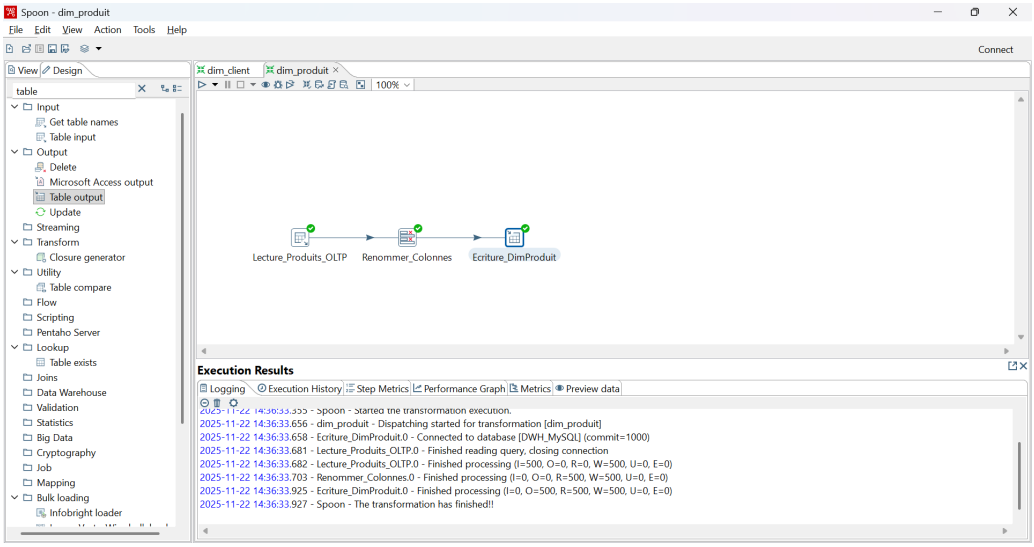

FIGURE 2 – Charger DimProduit

5.2.3 Transformation 3 : Générer DimDate (dim_date.ktr)

Objectif : Créer la dimension temporelle de 1096 jours (couvrant 2022-2024), qui est essentielle pour l'analyse de la saisonnalité.

Étapes Clés	Description et Justification
Generer_Lignes	`Generate Rows` : Génère 1096 lignes pour couvrir la période du 1er janvier 2022 au 31 décembre 2024.
Ajouter_Sequence	`Add sequence` : Ajoute un champ `numero_jour` incrémenté de 0 à 1095.
Calculer_Date	`Modified Java Script Value` : Utilise le `numero_jour` pour dériver la `date_complete` et toutes les composantes analytiques (annee, mois, trimestre, nom_mois, etc.). Il calcule également la clé surrogate `id_date_dim` au format YYYYMMDD.
Ecriture_DimDate	`Table Output` : Chargement des 1096 lignes dans DimDate.

TABLE 8 – Étapes de transformation DimDate

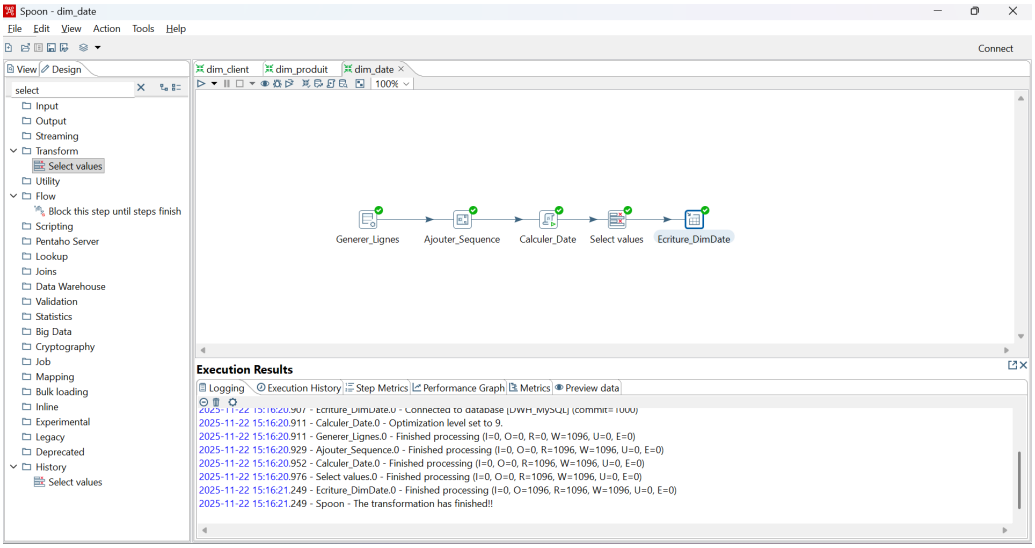

FIGURE 3 – Générer DimDate

5.3 Transformation de la Table de Faits

5.3.1 Transformation 4 : Charger FactVentes (fact_ventes.ktr)

Objectif : Créer la table de faits centrale en joignant les données de transaction OLTP et en remplaçant les clés naturelles par les clés surrogates du DWH. Cette transformation traite 100 000 lignes de commandes.

Étapes Clés	Description et Justification
Lecture_Lignes_ Commandes	`Table Input` : Extraction principale. Jointure SQL entre `lignes_commandes` (pour les quantités et prix) et `commandes` (pour l'id client et la date) de l'OLTP.
Lookup_Client	`Database lookup` : Recherche la clé surrogate `id_client_dim` dans DimClient en utilisant le champ `id_client` (OLTP).
Lookup_Produit	`Database lookup` : Recherche la clé surrogate `id_produit_dim` dans DimProduit en utilisant le champ `id_produit` (OLTP).
Lookup_Date	`Database lookup` : Recherche la clé surrogate `id_date_dim` dans DimDate en utilisant le champ `date_commande` (OLTP).
Calculer_Montant	`Calculator` : Ajoute la métrique `montant_total` en multipliant la `quantite` par le `prix_unitaire`.
Ecriture_FactVentes	`Table Output` : Écrit les 100 000 lignes de faits dans FactVentes, n'incluant que les clés surrogates (`id_client_dim`, `id_produit_dim`, `id_date_dim`) et les métriques (quantité, prix, montant total).

TABLE 9 – Étapes de transformation FactVentes

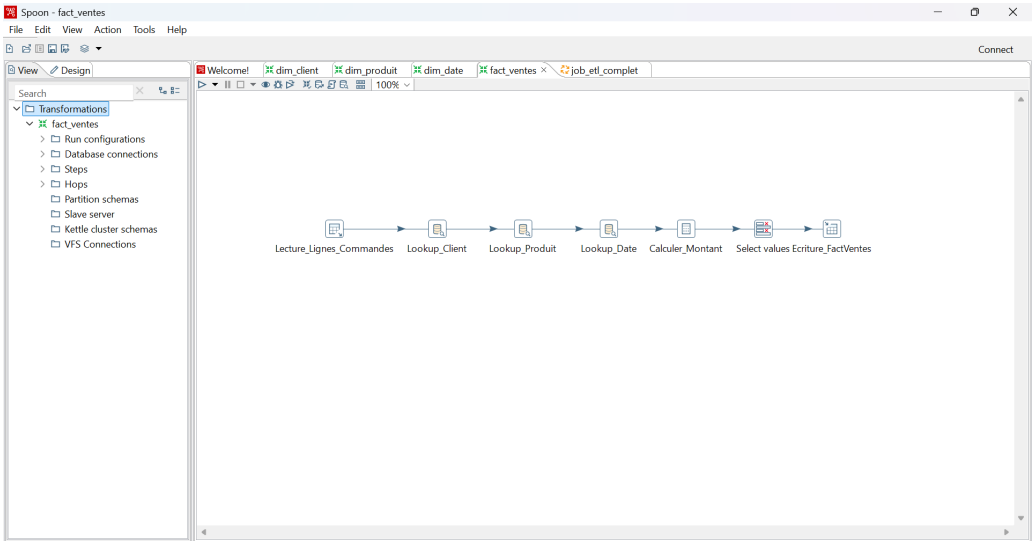

FIGURE 4 – Charger FactVentes

5.4 Job d'Orchestration

Le Job `job_etl_complet.kjb` assure l'exécution correcte et séquentielle de toutes les transformations.

1. **Logique Séquentielle :** Le Job démarre avec l'entrée START et enchaîne les transformations dans l'ordre de dépendance : Charger_DimClient → Charger_DimProduit → Charger_DimDate → Charger_FactVentes.
2. **Gestion des Dépendances :** Le Job utilise l'option Follow when result is true pour s'assurer que l'exécution de l'étape suivante n'a lieu que si la précédente transformation s'est terminée avec succès.

3. **Finalisation :** Un message de succès (`Write to log`) est affiché à la fin pour confirmer la mise à jour complète du Data Warehouse.

L'exécution réussie de ce Job permet de garantir que le DWH est rempli de 100 000 lignes de faits et est prêt à être interrogé par Power BI.

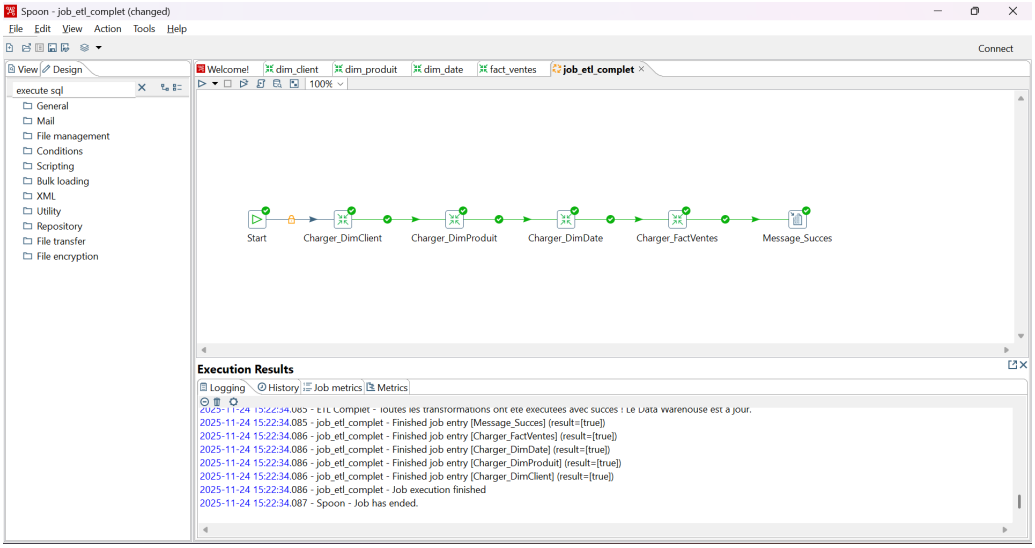

FIGURE 5 – Job d'Orchestration

6 Analyses OLAP

Le Data Warehouse (`ventes_dwh`), désormais rempli des 100 000 lignes de faits via le processus ETL Pentaho, est optimisé pour les requêtes analytiques OLAP (Online Analytical Processing). Contrairement au système OLTP, les requêtes suivantes sont exécutées rapidement grâce à l'efficacité du schéma en étoile.

Toutes les requêtes ci-dessous sont exécutées dans MySQL Workbench sur la base `ventes_dwh`.

6.1 Requête 1 : Chiffre d'affaires par ville

Cette analyse répond à la problématique de la performance commerciale par dimension géographique.

Interprétation Métier : Cette requête agrège le chiffre d'affaires (CA) en utilisant la dimension `DimClient` et le fait `montant_total`. Elle permet à TechStore d'identifier les villes les plus rentables. Par exemple, si l'analyse révèle que Paris génère 35% du CA, cela justifie de concentrer les efforts marketing ou d'investir dans une présence physique (comme un showroom) dans cette région.

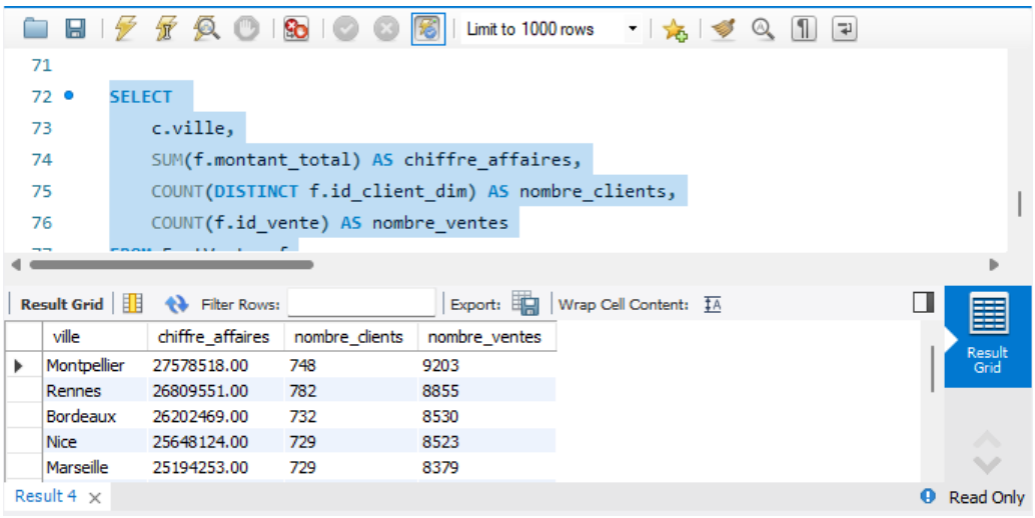

FIGURE 6 – Chiffre d'affaires par ville

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

Cette analyse identifie les produits générant le plus de revenus, répondant ainsi à l'objectif de performance par catégorie de produit.

Interprétation Métier : En se basant sur la dimension `DimProduit`, cette requête détermine la contribution relative de chaque catégorie au CA total. Si la catégorie « Ordinateurs » représente la majorité du CA (par exemple, 45%), cela justifie d'élargir continuellement cette gamme de produits. Inversement, les catégories peu performantes pourraient faire l'objet d'une promotion différente ou être abandonnées pour optimiser le stock.

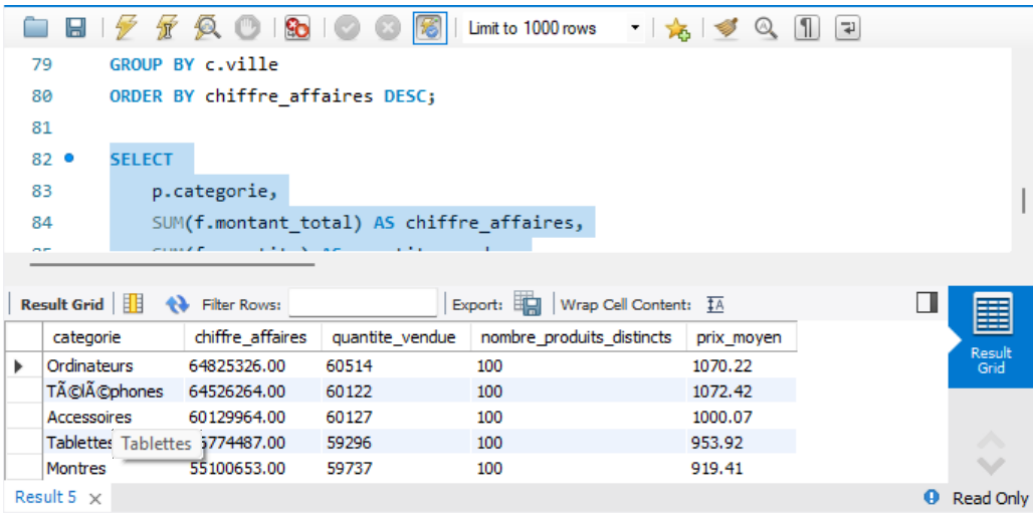

FIGURE 7 – Chiffre d'affaires par catégorie de produit

6.3 Requête 3 : Évolution des ventes par mois

Cette analyse temporelle utilise la dimension `DimDate` pour déceler la saisonnalité des ventes.

Interprétation Métier : Les résultats de cette requête révèlent la saisonnalité des ventes et les tendances annuelles. L'identification d'un pic de ventes récurrent en décembre (effet des fêtes de fin d'année) est un insight crucial, suggérant la nécessité d'augmenter les niveaux de stock et de lancer les campagnes publicitaires ciblées dès novembre.

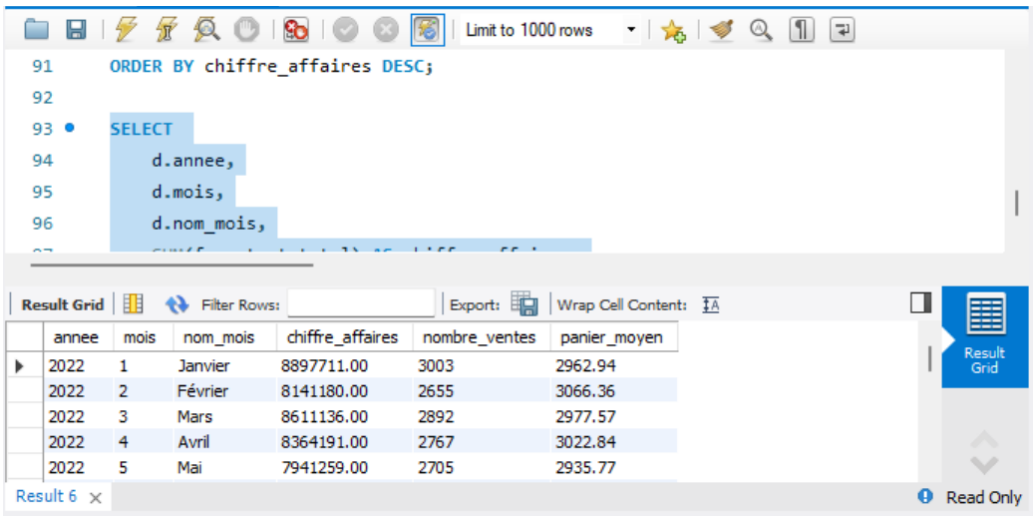

FIGURE 8 – Évolution des ventes par mois

6.4 Requête 4 : Top 10 des produits les plus vendus

Cette analyse se concentre sur le comportement d'achat et permet d'identifier les produits stars du catalogue.

Interprétation Métier : L'identification des 10 produits les plus vendus est vitale, car ils contribuent souvent de manière disproportionnée au CA (principe de Pareto). TechStore doit s'assurer que ces produits sont toujours disponibles en stock et qu'ils bénéficient d'une mise en avant stratégique sur le site web pour maximiser les ventes.

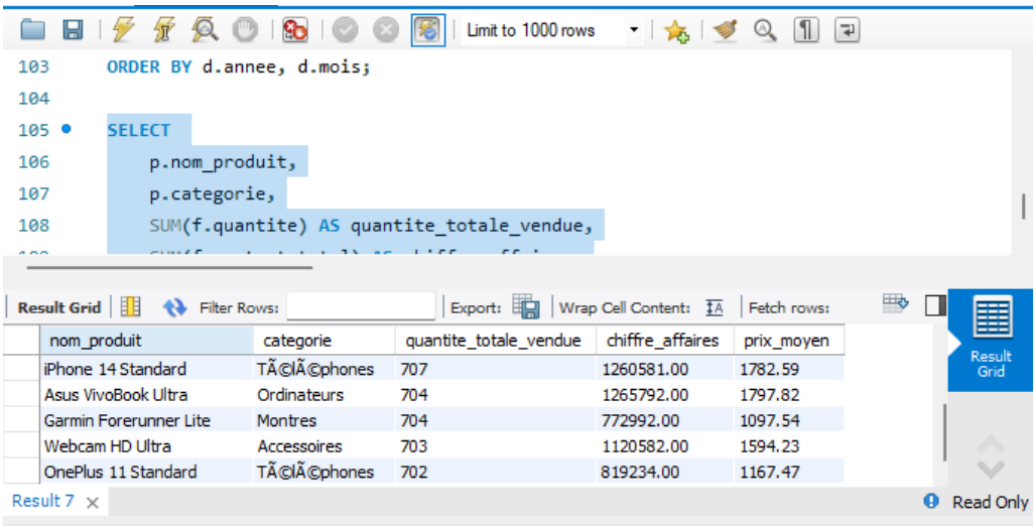

FIGURE 9 – Top 10 des produits les plus vendus

6.5 Requête 5 : Analyse croisée par trimestre et catégorie (Analyse multidimensionnelle)

Cette requête combine plusieurs dimensions (`DimDate` et `DimProduit`) pour une analyse approfondie.

Interprétation Métier : Ce type d'analyse révèle des tendances fines. Elle permet de savoir quelles catégories performent le mieux à quel moment précis de l'année. Par exemple, si les « Tablettes » se vendent majoritairement au Q3, c'est probablement lié à la rentrée scolaire.

Les « Montres » pourraient avoir leur pic au Q4 en préparation des cadeaux de Noël. Ces informations affinent les stratégies promotionnelles.

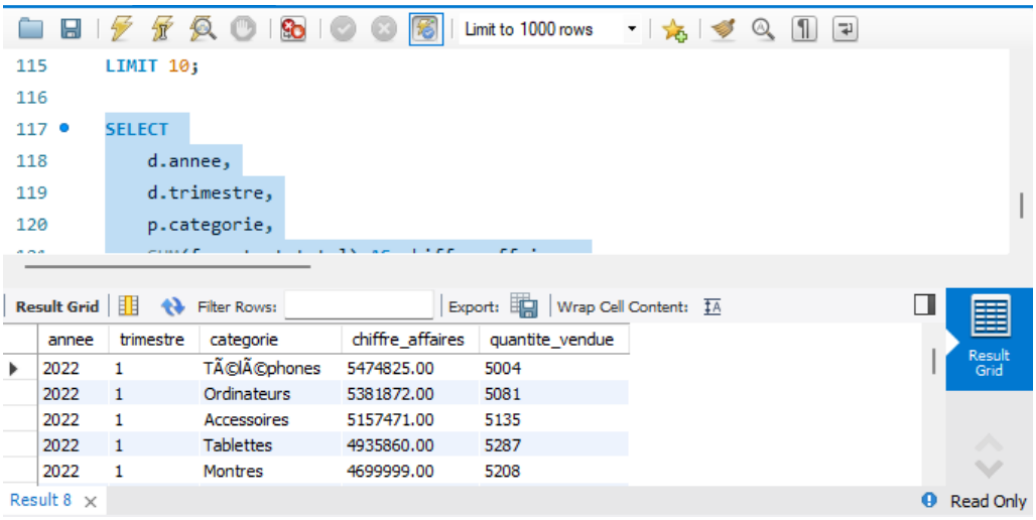

FIGURE 10 – Analyse croisée par trimestre et catégorie

6.6 Avantages du Schéma en Étoile pour ces Analyses

Toutes ces requêtes complexes sont exécutées très rapidement malgré les 100 000 lignes de faits. Cette performance est due à la structure du schéma en étoile, qui nécessite seulement 1 ou 2 jointures entre la table de faits et les dimensions (DimClient, DimProduit, DimDate) pour obtenir toutes les informations analytiques nécessaires (ville, catégorie, nom du mois, etc.).

7 Visualisations Power BI

L'étape finale du projet consiste à utiliser Power BI pour transformer les données du Data Warehouse (DWH) en un tableau de bord interactif. Power BI se connecte directement à la base `ventes_dwh`, tirant parti du schéma en étoile optimisé pour l'analyse OLAP.

Le rapport final, nommé Dashboard Ventes TechStore, est structuré autour de cartes KPI et de visualisations dynamiques permettant aux décideurs d'explorer les données sans utiliser SQL.

7.1 Cartes KPI (Indicateurs Clés de Performance)

Quatre indicateurs clés sont affichés en haut du tableau de bord grâce à des mesures DAX calculées sur la table de faits `FactVentes` :

Mesure DAX	Calcul	Rôle Métier
CA Total	`SUM(FactVentes[montant_total])`	Représente le revenu global généré par TechStore.
Nombre de Ventes	`COUNT(FactVentes[id_vente])`	Indique le volume total des transactions.
Clients Uniques	`DISTINCTCOUNT(FactVentes_[id_client_dim])`	Mesure la taille de la base de clients actifs ayant réalisé un achat.
Panier Moyen	`DIVIDE([CA Total], [Nombre de Ventes], 0)`	Essentiel pour évaluer la valeur moyenne des commandes et l'efficacité des stratégies de vente croisée.

TABLE 10 – Mesures DAX pour les KPI

7.2 Visualisations Détaillées et Interprétations Métier

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

— **Configuration :** Utilise la dimension `DimClient` (champ `ville`) pour l'axe Y et la mesure `montant_total` de `FactVentes` pour les valeurs.

— **Insight Métier :** Ce visuel est conçu pour répondre à la question "Quel est le chiffre d'affaires par ville ?". Il permet d'identifier instantanément les zones géographiques les plus rentables. Si Paris et Lyon dominent, la direction peut justifier l'allocation de budgets marketing spécifiques ou l'ouverture d'un showroom physique dans ces zones.

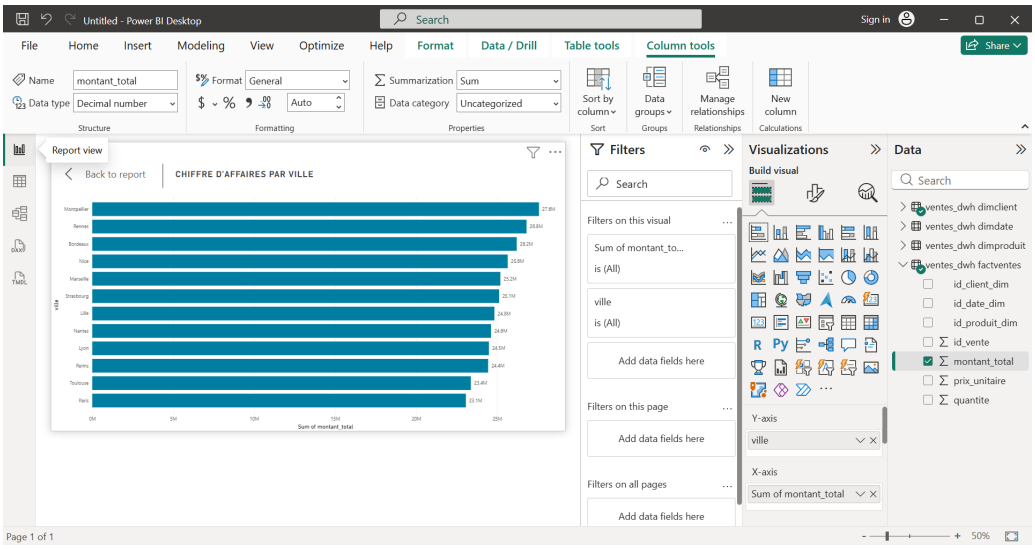

FIGURE 11 – Chiffre d'affaires par ville

7.2.2 Visuel 2 : Répartition du CA par catégorie (Graphique en secteurs)

— **Configuration :** Utilise `DimProduit.categorie` pour la Légende et `FactVentes._montant_total` pour les Valeurs.

— **Insight Métier :** Ce graphique répond à la question "Quelles catégories de produits génèrent le plus de revenus ?". Il révèle la contribution relative de chaque gamme au CA total. Par exemple, si la catégorie "Ordinateurs" représente 21.51% du camembert, c'est la catégorie la plus stratégique à surveiller pour éviter les ruptures de stock et pour justifier l'élargissement de la gamme.

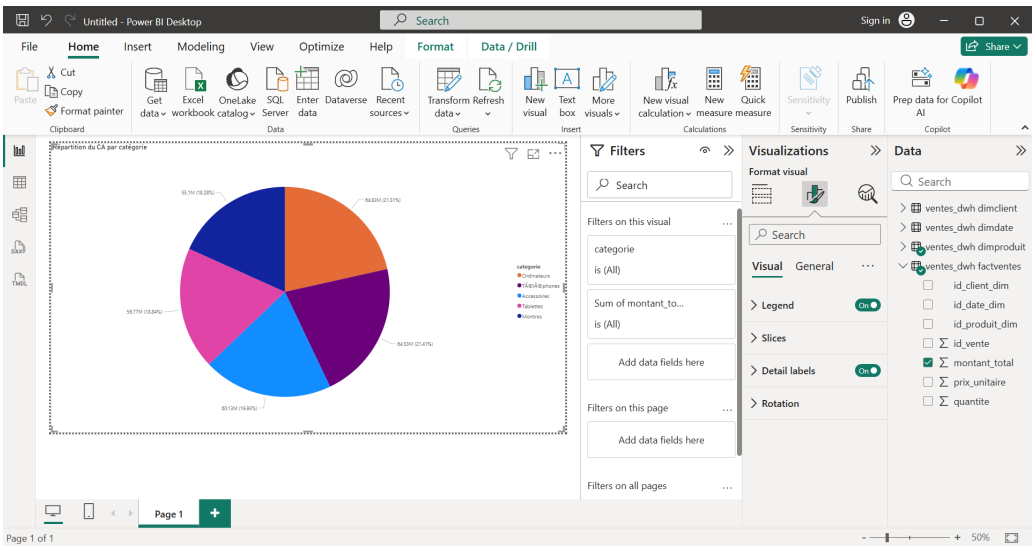

FIGURE 12 – Répartition du CA par catégorie

7.2.3 Visuel 3 : Évolution mensuelle du chiffre d'affaires (Graphique en courbes)

— **Configuration :** Utilise la hiérarchie temporelle de `DimDate` (niveau Mois) pour l'Axe X et `FactVentes.montant_total` pour les Valeurs.

— **Insight Métier :** Ce visuel répond à l'objectif de l'analyse de tendance temporelle. Il est essentiel pour déceler la saisonnalité des ventes. L'identification de pics réguliers (par exemple, en décembre) est un indice clair (effet des fêtes de fin d'année) qui doit conduire à des décisions proactives sur le stock et le lancement des campagnes publicitaires.

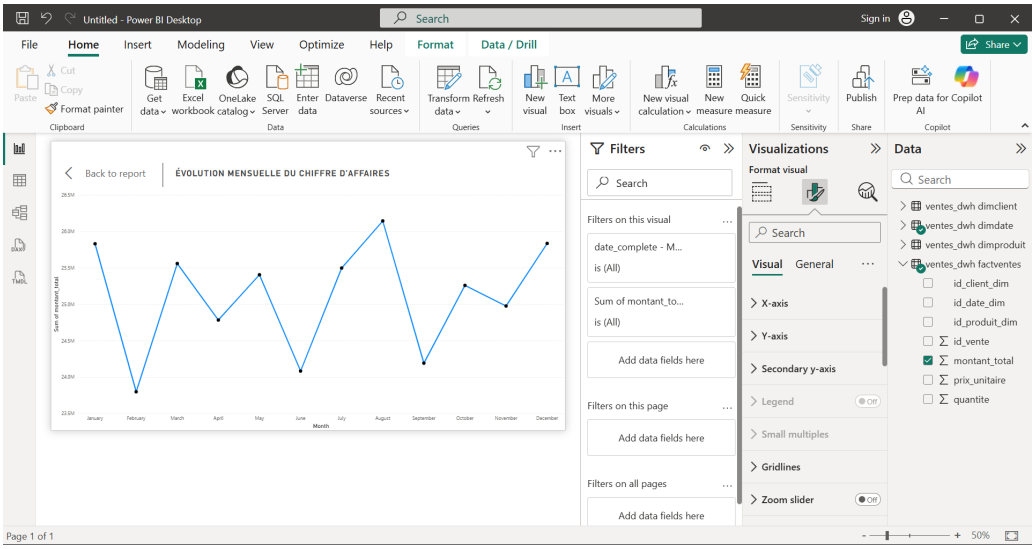

FIGURE 13 – Évolution mensuelle du chiffre d'affaires

7.2.4 Visuel 4 : Top 10 des produits les plus vendus (Graphique à barres horizontales)

— **Configuration :** Utilise `DimProduit.nom_produit` pour l'Axe Y et `FactVentes.quantite` pour les Valeurs, avec un filtre Top N appliqué sur la `Somme de quantite` pour ne garder que les 10 meilleurs.

— **Insight Métier :** Ce graphique permet d'identifier les produits stars du catalogue. En accord avec le principe de Pareto (80/20), ces produits contribuent souvent de manière

disproportionnée au CA. Ils doivent bénéficier d'une mise en avant stratégique sur la page d'accueil du site pour maximiser leur potentiel de vente.

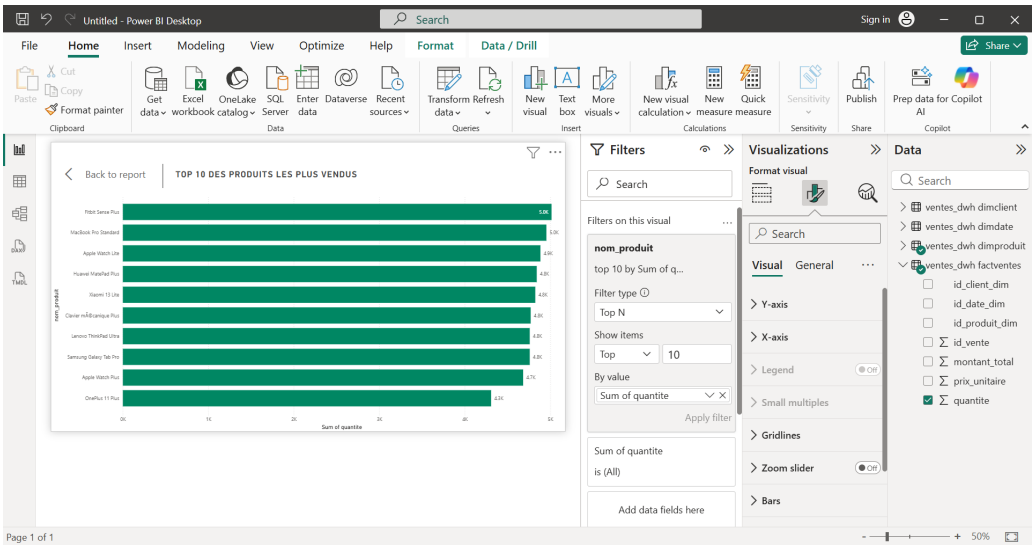

FIGURE 14 – Top 10 des produits les plus vendus

7.3 Interactivité du Rapport

L'une des forces de Power BI est l'interactivité. Des segments (Slicers) ont été ajoutés pour permettre aux utilisateurs de filtrer dynamiquement l'ensemble du rapport par Année (`DimDate.annee`) et par Catégorie (`DimProduit.categorie`).

De plus, la fonctionnalité de Drill-through (ou exploration en profondeur) a été implémentée. Elle permet aux utilisateurs de cliquer sur un élément (comme un produit du Top 10) et d'accéder à une page détaillée, filtrée sur cet élément, sans surcharger le tableau de bord principal. Cette interactivité bidirectionnelle (cross-filtering) permet une exploration intuitive des données.

L'ensemble de ces visualisations permet à TechStore d'avoir une vue unifiée et dynamique des KPI et de transformer les données complexes du Data Warehouse en informations stratégiques exploitables.

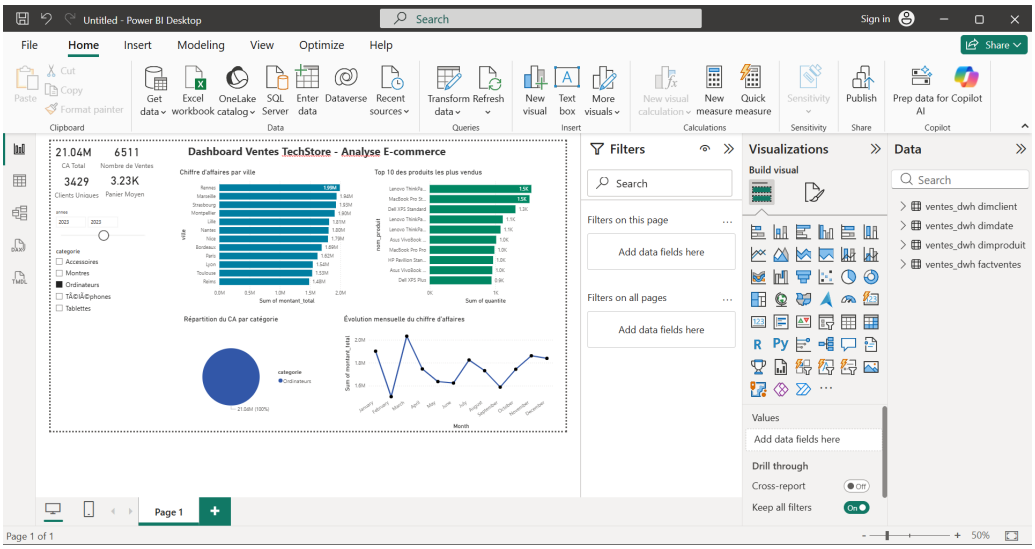

FIGURE 15 – Dashboard Complet

Conclusions et Recommandations

Ce projet de Business Intelligence (BI) pour TechStore a mis en place une chaîne complète afin de migrer les données transactionnelles normalisées (OLTP) depuis la base MySQL `ventes_oltp` vers une structure analytique (OLAP). La base de données source, qui contient un jeu de données synthétique de 100 000 lignes de commandes générées sur trois ans, est structurée autour de quatre tables relationnelles. Le processus ETL, orchestré par Pentaho PDI, a été conçu pour nettoyer, transformer et charger les données dans un Data Warehouse (DWH) MySQL `ventes_dwh`. Ce DWH est modélisé en Schéma en Étoile, composé de la table de faits `FactVentes` et des dimensions `DimClient`, `DimProduit`, et `DimDate`, afin de garantir des requêtes d'agrégation rapides. Ces requêtes OLAP ont validé le DWH en répondant aux questions métier critiques, notamment l'identification du chiffre d'affaires par ville et par catégorie, et l'analyse de la saisonnalité. Enfin, les résultats ont été visualisés et rendus accessibles aux décideurs via un tableau de bord Power BI interactif. Pour une mise en production future, il est recommandé d'ajouter un mécanisme de chargement incrémental des faits et d'implémenter les Slowly Changing Dimensions (SCD).

Annexes

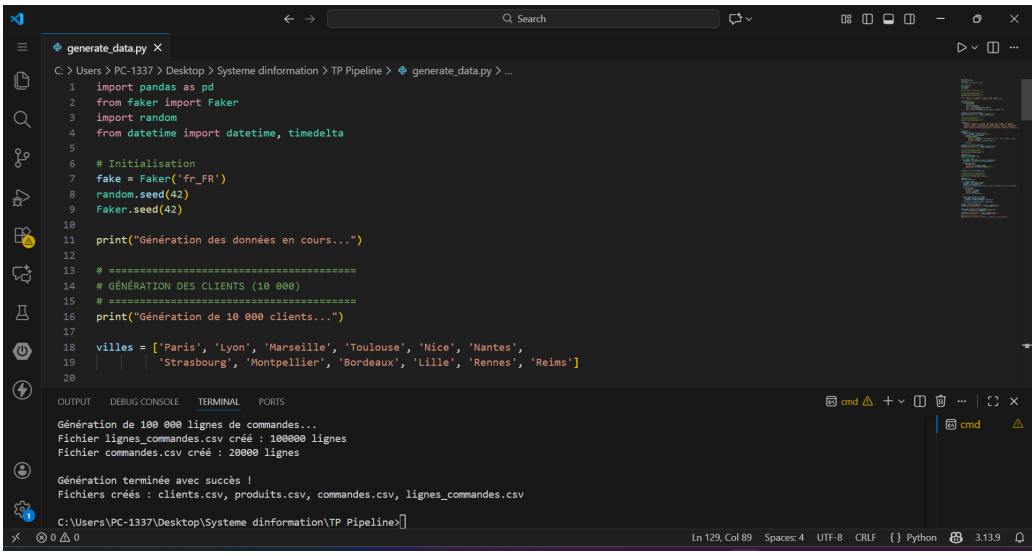

FIGURE 16 – Python

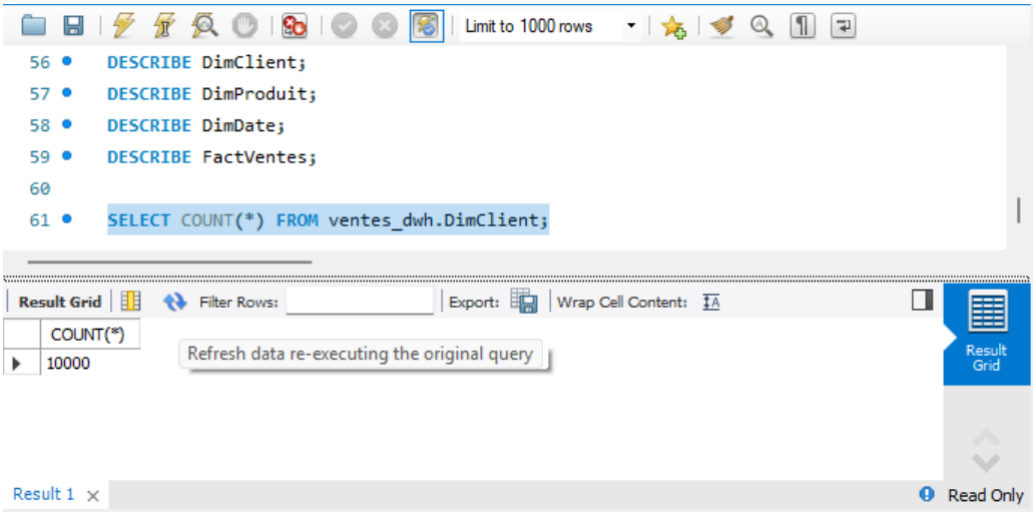

FIGURE 17 – Verification de DimClient

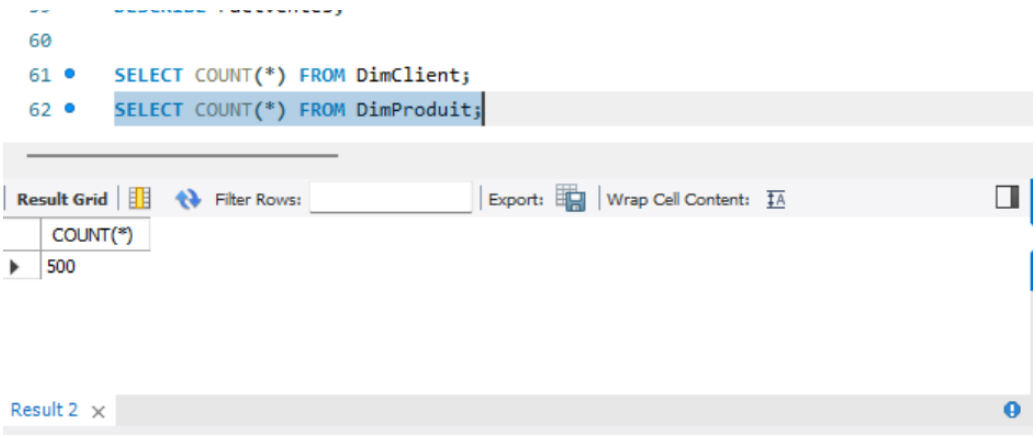

FIGURE 18 – Verification de DimProduit

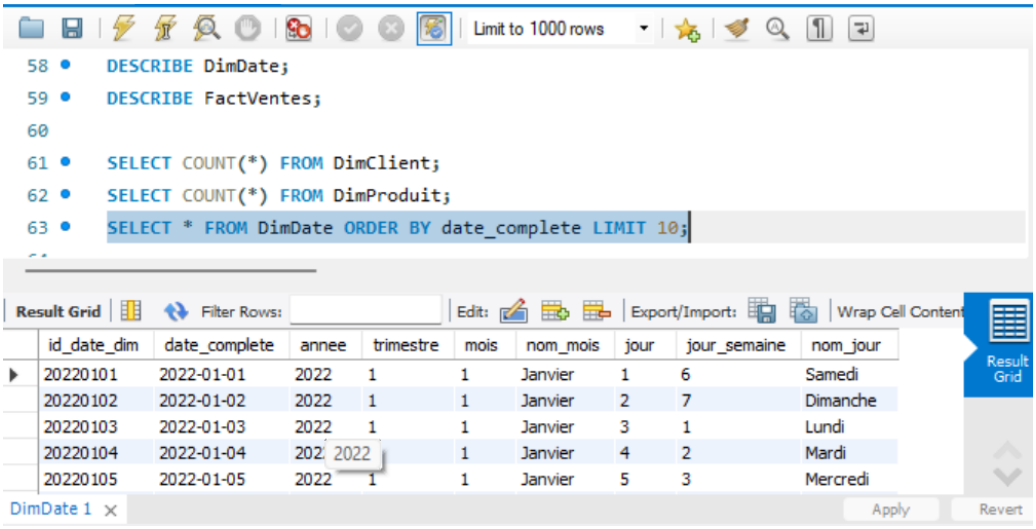

FIGURE 19 – Verification de DimDate

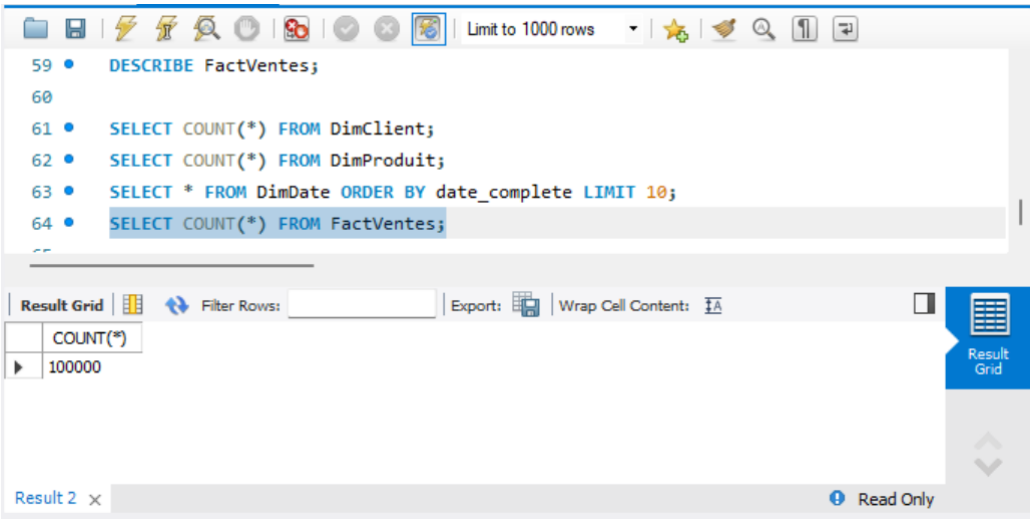

FIGURE 20 – Verification de FactVentes

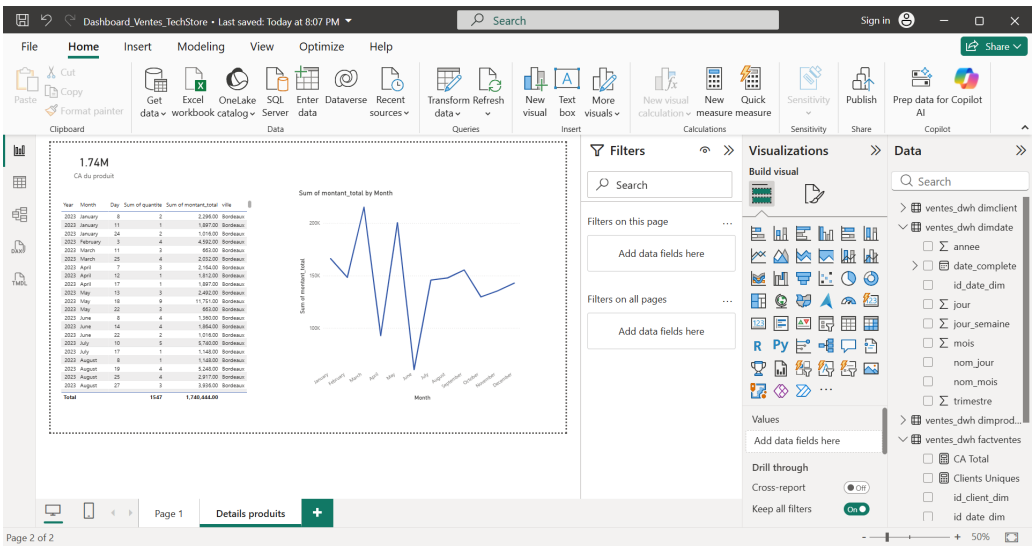

FIGURE 21 – Drill-Through