

# Rapport de Projet de Business Intelligence

## Pipeline complet : OLTP → ETL Pentaho → Data Warehouse → Reporting Power BI

FORMATION : 4<sup>e</sup> année d'ingénierie Big Data – UEMF

Promo : BD 2025-2026

Réalisée par :

**Ahlam OUBOUAZZA**

Supervisé par :

**Ahmed AMAMOU**

# Sommaire

## 1. Introduction

- Contexte du projet
- Objectifs et buts du projet

## 2. Technologies utilisées

## 3. Implémentation

- Backend (FastAPI)
- Frontend (React)
- Structure du projet

## 4. Résultats

4.1. Évaluation du modèle

4.2. Prédictions de prix

## 5. Discussion

5.1. Analyse des résultats obtenus

5.2. Limites du modèle

## 6. Conclusion

- Résumé des conclusions
- Impact des résultats obtenus

# 1. Introduction

**TechStore** est une entreprise e-commerce fictive spécialisée dans les produits électroniques. Ce projet répond au besoin de transformer ses données transactionnelles brutes en informations analysables pour la prise de décision.

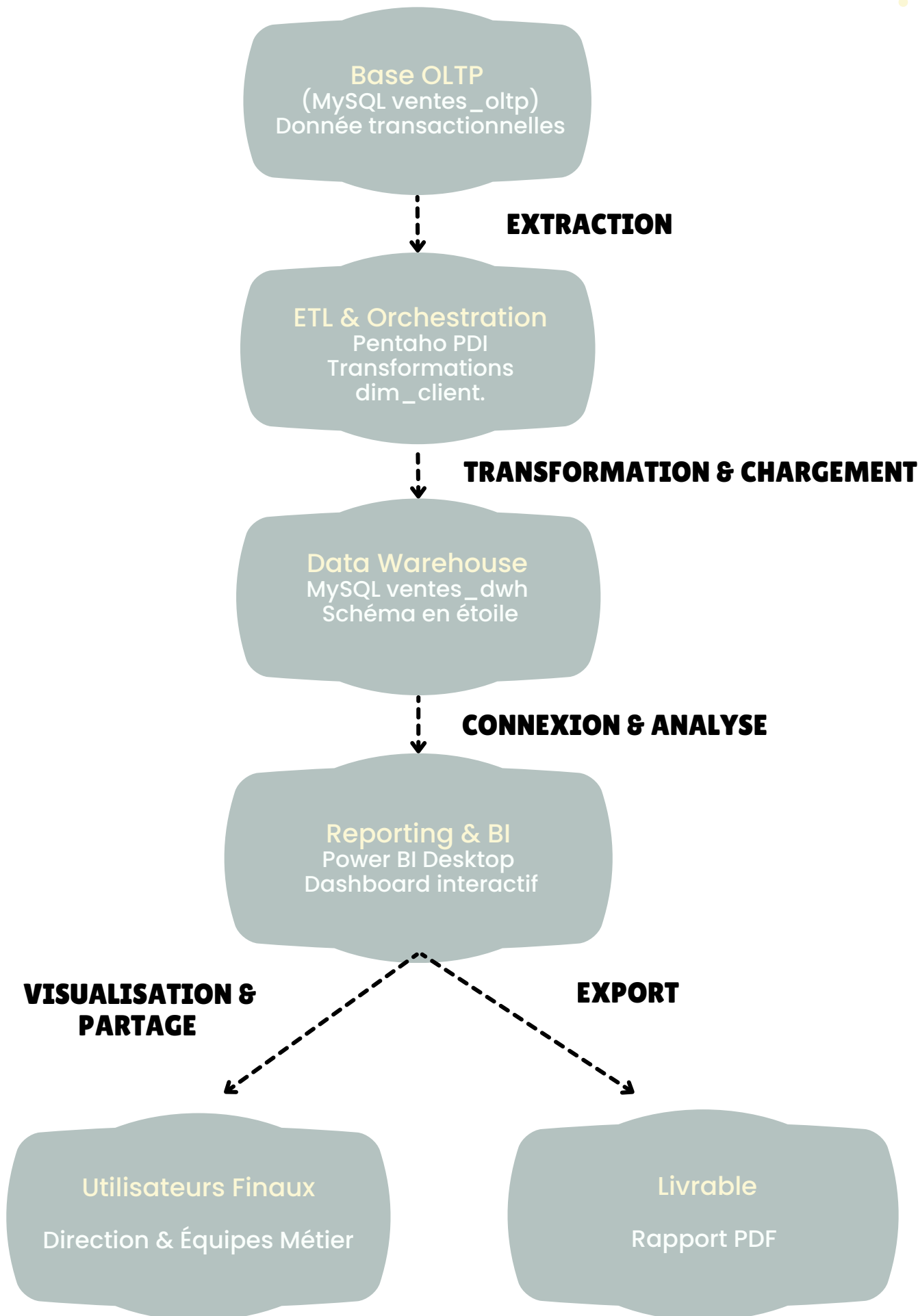
## Objectifs :

- Modéliser une base OLTP et générer des données réalistes.
- Concevoir un Data Warehouse avec un schéma en étoile.
- Automatiser le flux ETL avec Pentaho PDI.
- Créer un tableau de bord interactif avec Power BI.

# 2. Architecture du Projet

## Technologies utilisées :

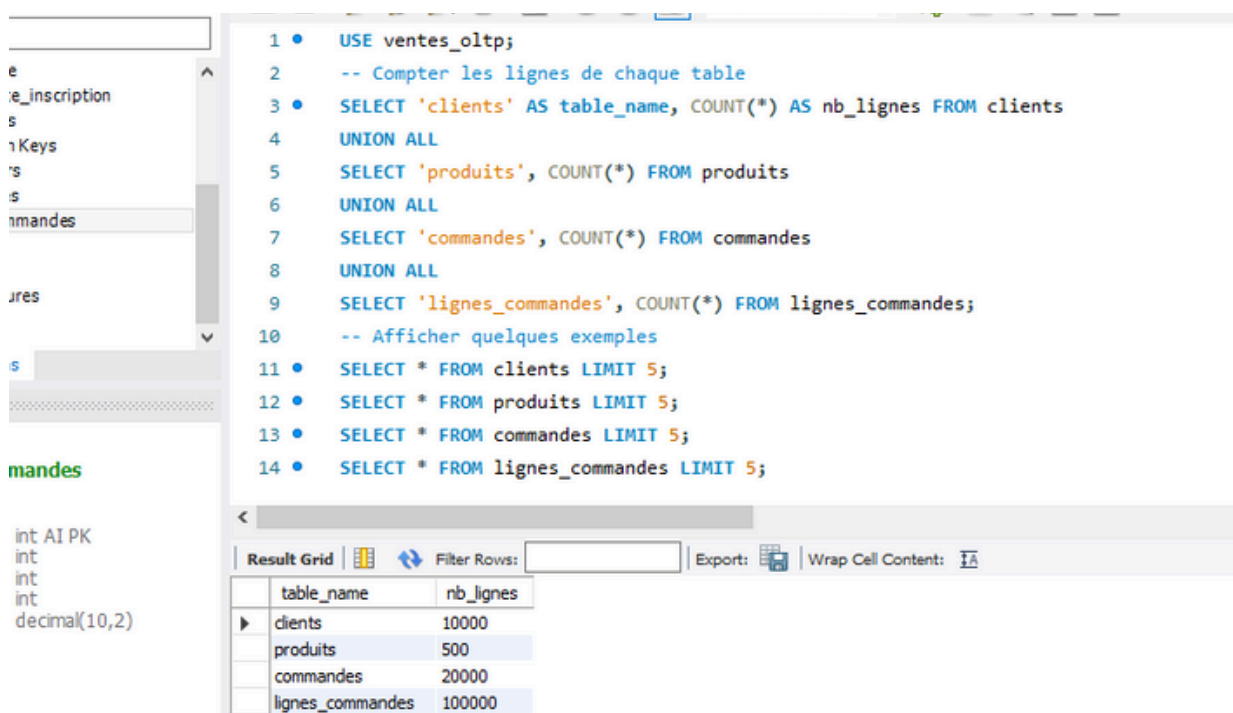
- SGBD : MySQL 8.0
- ETL : Pentaho Data Integration (PDI / Kettle) 9.x
- Visualisation : Microsoft Power BI Desktop
- Langages : SQL, Python (pour la génération de données)
- Système : Windows



### 3. Modélisation OLTP

La base **ventes\_oltp** contient 4 tables normalisées :

- clients : 10 000 clients
- produits : 500 produits
- commandes : 20 000 commandes
- lignes\_commandes : 100 000 lignes



The screenshot shows a database management interface. On the left, a tree view displays the database structure, including tables like 'clients', 'produits', 'commandes', and 'lignes\_commandes'. The main area displays a SQL script with the following queries:

```
1 • USE ventes_oltp;
2   -- Compter les lignes de chaque table
3 • SELECT 'clients' AS table_name, COUNT(*) AS nb_lignes FROM clients
4   UNION ALL
5   SELECT 'produits', COUNT(*) FROM produits
6   UNION ALL
7   SELECT 'commandes', COUNT(*) FROM commandes
8   UNION ALL
9   SELECT 'lignes_commandes', COUNT(*) FROM lignes_commandes;
10  -- Afficher quelques exemples
11 • SELECT * FROM clients LIMIT 5;
12 • SELECT * FROM produits LIMIT 5;
13 • SELECT * FROM commandes LIMIT 5;
14 • SELECT * FROM lignes_commandes LIMIT 5;
```

Below the script, a 'Result Grid' shows the output of the queries:

table_name	nb_lignes
clients	10000
produits	500
commandes	20000
lignes_commandes	100000

**Justification des choix :** Le modèle relationnel (3FN) a été choisi pour son efficacité dans la gestion des transactions (INSERT, UPDATE, DELETE), la minimisation des redondances et le maintien de l'intégrité référentielle via des clés étrangères.

## 4. Génération des Données

Pour simuler une activité réaliste, un jeu de données conséquent a été généré via un script Python personnalisé.

**Méthodologie** : Utilisation des bibliothèques Faker (pour des données réalistes) et pandas. Les données ont été générées en respectant les contraintes d'intégrité référentielle (ex: un id\_client dans la table commandes doit exister dans la table clients).

### Volumes et Répartition :

- Période couverte : 3 ans (2022-2024)
  - Clients : 10 000, répartis sur 12 villes françaises.
  - Produits : 500, répartis en 5 catégories (Ordinateurs, Téléphones, Tablettes, Accessoires, Montres).
  - Commandes : 20 000, générées aléatoirement sur la période.
- L-ignes de commande : 100 000, assurant un lien cohérent entre commandes et produits.



clients.csv - Microsoft Excel (Échec									
Fichier Accueil Insertion Mise en page Formules Données Révision Affichage PDFement									
Calibri 11 A A G I S Police Alignement									
fx id_client,nom,prenom,email,ville,date_inscription									
	A	B	C	D	E	F	G	H	I
1	id_client,nom,prenom,email,ville,date_inscription								
2	1	Riou,FrÃ©dÃ©ric	client1@voila.fr	Rennes	2024-01-22				
3	2	Moulin,Maurice	client2@laposte.net	Lyon	2023-04-29				
4	3	Labbe,Nicole	client3@hotmail.fr	Paris	2025-05-26				
5	4	Briand,Arthur	client4@voila.fr	Reims	2023-04-15				
6	5	Carpentier,Nicolas	client5@wanadoo.fr	Nice	2025-06-13				
7	6	Michel,Gabrielle	client6@sfr.fr	Toulouse	2025-08-26				
8	7	Boutin,Ã%milie	client7@noos.fr	Toulouse	2023-10-29				
9	8	Pons,Jacqueline	client8@bouygtel.fr	Marseille	2022-12-01				
10	9	Thierry,Susan	client9@free.fr	Reims	2025-11-09				
11	10	Briand,Zacharie	client10@bouygtel.fr	Lyon	2023-07-20				
12	11	Fabre,Paulette	client11@dbmail.com	Rennes	2023-04-28				

## 5. Modélisation du Data Warehouse (DWH)

Pour l'analyse, un schéma en étoile dénormalisé a été conçu dans la base ventes\_dwh.

### Structure du Schéma en Étoile :

-Table de Faits : FactVentes (100 000 lignes). Contient les mesures (quantité, prix, montant) et des clés étrangères vers les dimensions.

Dimensions :

-DimClient : Dérivée de la table clients.

-DimProduit : Dérivée de la table produits.

-DimDate : Table de dimension temporelle artificielle et pré-calculée (1 096 jours).

### Justification et Avantages :

Ce modèle a été choisi pour :

-Performance : Réduction du nombre de jointures pour les requêtes analytiques.

-Simplicité : Structure intuitive pour les utilisateurs métier et les outils de BI.

-Intégration : Optimisation pour les requêtes d'agrégation et de filtrage multidimensionnel typiques des outils comme Power BI.

The screenshot displays a SQL IDE interface. On the left, a schema diagram shows a table 'clients' with columns 'id\_client', 'nom', 'prenom', 'email', 'ville', and 'date\_inscription'. Below it, a table 'lignes\_commandes' is shown with columns 'id\_client', 'id\_produit', 'date', 'quantite', 'prix', and 'montant'. The main window shows a SQL query file with the following content:

```
55 • SHOW TABLES;
56 • DESCRIBE DimClient;
57 • DESCRIBE DimProduit;
58 • DESCRIBE DimDate;
59 • DESCRIBE FactVentes;
```

The 'Result Grid' shows the output of the queries, listing tables in the 'ventes\_dwh' database: 'dimclient', 'dimdate', 'dimproduit', and 'factventes'. Below the grid, a table structure is displayed for 'dimclient':

Field	Type	Null	Key	Default	Extra
id_client_dim	int	NO	PRI	NULL	auto_increment
id_client_source	int	NO	UNI	NULL	
nom_complet	varchar(200)	NO		NULL	
email	varchar(150)	NO		NULL	
ville	varchar(100)	NO	MUL	NULL	

SQL File 1\* x

Limit to 1000 rows

```

53 ) ENGINE=InnoDB;
54
55 • SHOW TABLES;
56 • DESCRIBE DimClient;
57 • DESCRIBE DimProduit;
58 • DESCRIBE DimDate;
59 • DESCRIBE FactVentes;
60
61

```

Result Grid

Field	Type	Null	Key	Default	Extra
id_vente	int	NO	PRI	HULL	auto_increment
id_client_dim	int	NO	MUL	HULL	
id_produit_dim	int	NO	MUL	HULL	
id_date_dim	int	NO	MUL	HULL	
quantite	int	NO		HULL	
prix_unitaire	decimal(10,2)	NO		HULL	
montant_total	decimal(10,2)	NO		HULL	

Result 43 x

Output

Action Output

SQL File 1\* x

Limit to 1000 rows

```

53 ) ENGINE=InnoDB;
54
55 • SHOW TABLES;
56 • DESCRIBE DimClient;
57 • DESCRIBE DimProduit;
58 • DESCRIBE DimDate;
59 • DESCRIBE FactVentes;
60

```

Result Grid

Field	Type	Null	Key	Default	Extra
id_produit_dim	int	NO	PRI	HULL	auto_increment
id_produit_source	int	NO	UNI	HULL	
nom_produit	varchar(200)	NO		HULL	
categorie	varchar(100)	NO	MUL	HULL	

Result 41 x

Output

SQL File 1\* x

Limit to 1000 rows

```

53 ) ENGINE=InnoDB;
54
55 • SHOW TABLES;
56 • DESCRIBE DimClient;
57 • DESCRIBE DimProduit;
58 • DESCRIBE DimDate;

```

Result Grid

Field	Type	Null	Key	Default	Extra
id_date_dim	int	NO	PRI	HULL	
date_complete	date	NO	UNI	HULL	
annee	int	NO	MUL	HULL	
trimestre	int	NO	MUL	HULL	
mois	int	NO	MUL	HULL	
nom_mois	varchar(20)	NO		HULL	
jour	int	NO		HULL	
jour_semaine	int	NO		HULL	
nom_jour	varchar(20)	NO		HULL	

Result 42 x

Output

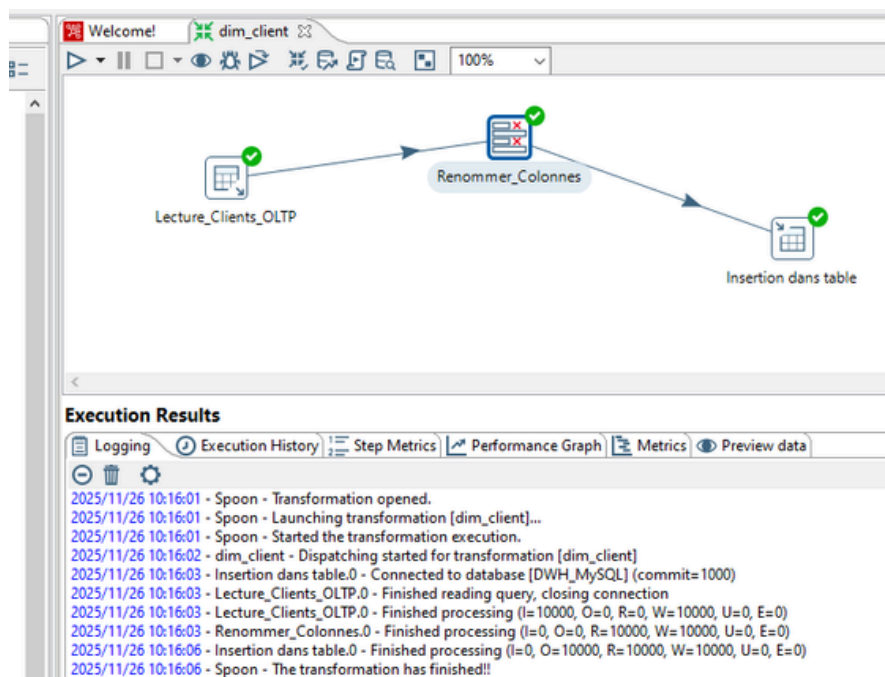


## 6. Processus ETL avec Pentaho

Le processus ETL a été conçu dans Pentaho Data Integration (Spoon) pour migrer et transformer les données de l'OLTP vers le DWH de manière automatisée et reproductible.

### Transformations développées :

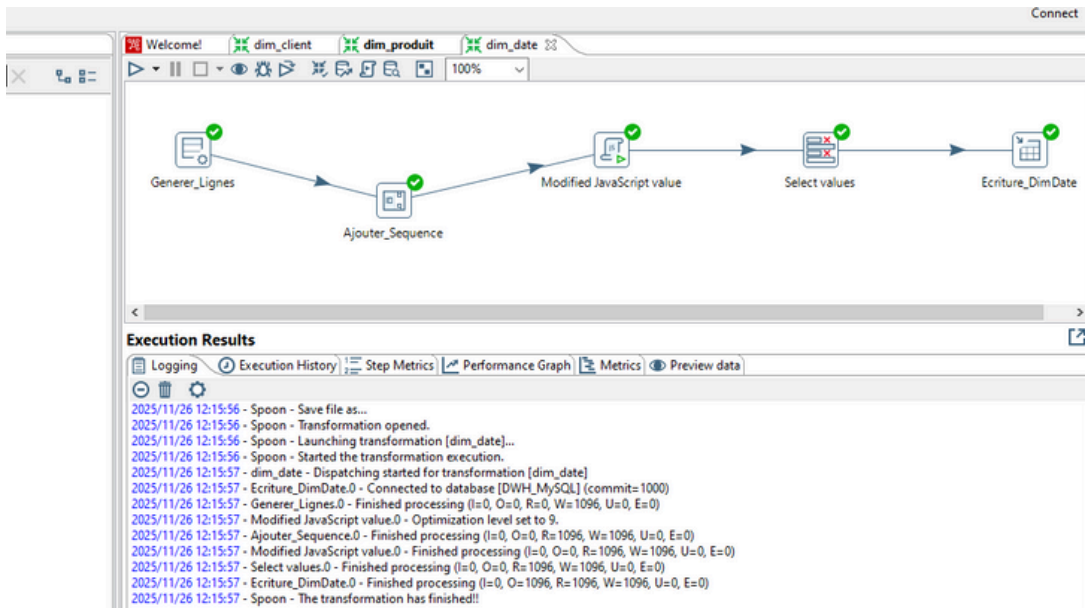
**1-dim\_client.ktr :** Lit la table clients, concatène le nom et prénom, et charge DimClient.



The screenshot shows the 'SCHEMAS' tab in Pentaho Spoon. On the left, the 'ventes\_dwh' schema is expanded, showing the 'dimclient' table. The table's columns are listed: 'id\_client\_dim' (int, AI, PK), 'id\_client\_source' (int), 'nom\_complet' (varchar(200)), 'email' (varchar(150)), and 'ville' (varchar(100)). On the right, the 'Result Grid' shows the data being loaded from the 'ventes\_dwh.dimclient' table. The data is displayed in a table with 16 rows.

id_client_dim	id_client_source	nom_complet	email	ville
10001	1	FRÃ©dÃ©ric Rou	client1@voila.fr	Rennes
10002	2	Maurice Moulin	client2@laposte.net	Lyon
10003	3	Nicole LabbÃ©	client3@hotmail.fr	Paris
10004	4	Arthur Briand	client4@voila.fr	Reims
10005	5	Nicolas Carpenter	client5@wanadoo.fr	Nice
10006	6	Gabrielle Michel	client6@sfr.fr	Toulouse
10007	7	Ã‰milie Boutin	client7@noos.fr	Toulouse
10008	8	Jacqueline Pons	client8@bouygtele.fr	Marseille
10009	9	Susan Thierry	client9@free.fr	Reims
10010	10	Zacharie Briand	client10@bouygtele.fr	Lyon
10011	11	Paulette Fabre	client11@dbmail.com	Rennes
10012	12	Alex Morin	client12@yahoo.fr	Reims
10013	13	Josette Barre	client13@bouygtele.fr	Bordeaux
10014	14	CÃ©line Chauveau	client14@yahoo.fr	Lyon
10015	15	FranÃ©oise Chev...	client15@club-intern...	Lille
10016	16	AgnÃ©s Maurice	client16@sefr.fr	Strasbo...

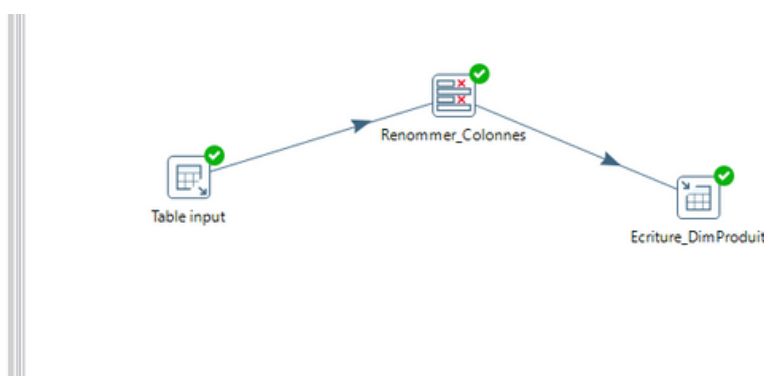
**2-dim\_date.ktr** : Génère dynamiquement la table de temps DimDate pour la période 2022-2024.



The screenshot shows the 'Result Grid' for the '2-dim\_date.ktr' workflow. The grid displays a list of dates from 2022-01-01 to 2022-01-16, along with their corresponding day of the week and month.

id_date_dim	date_complete	annee	trimestre	mois	nom_mois	jour	jour_semaine	nom_jour
20220101	2022-01-01	2022	1	1	Janvier	1	6	Samedi
20220102	2022-01-02	2022	1	1	Janvier	2	7	Dimanche
20220103	2022-01-03	2022	1	1	Janvier	3	1	Lundi
20220104	2022-01-04	2022	1	1	Janvier	4	2	Mardi
20220105	2022-01-05	2022	1	1	Janvier	5	3	Mercredi
20220106	2022-01-06	2022	1	1	Janvier	6	4	Jeudi
20220107	2022-01-07	2022	1	1	Janvier	7	5	Vendredi
20220108	2022-01-08	2022	1	1	Janvier	8	6	Samedi
20220109	2022-01-09	2022	1	1	Janvier	9	7	Dimanche
20220110	2022-01-10	2022	1	1	Janvier	10	1	Lundi
20220111	2022-01-11	2022	1	1	Janvier	11	2	Mardi
20220112	2022-01-12	2022	1	1	Janvier	12	3	Mercredi
20220113	2022-01-13	2022	1	1	Janvier	13	4	Jeudi
20220114	2022-01-14	2022	1	1	Janvier	14	5	Vendredi
20220115	2022-01-15	2022	1	1	Janvier	15	6	Samedi
20220116	2022-01-16	2022	1	1	Janvier	16	7	Dimanche

**3-dim\_produit.ktr** : Lit la table produits et charge DimProduit.



Filter objects

- ventes\_dwh
  - Tables
    - dimclient
    - dimdate
    - dimproduit
    - factventes
  - Views
  - Stored Procedures
  - Functions
  - ventes\_olp
- Administration
- Schemas

Table: **dimproduit**

Columns:

Column Name	Column Type	Column Properties
id_produit_dim	int	AI PK
id_produit_source	int	
nom_produit	varchar(200)	
categorie	varchar(100)	

1 • SELECT \* FROM ventes\_dwh.dimproduit;

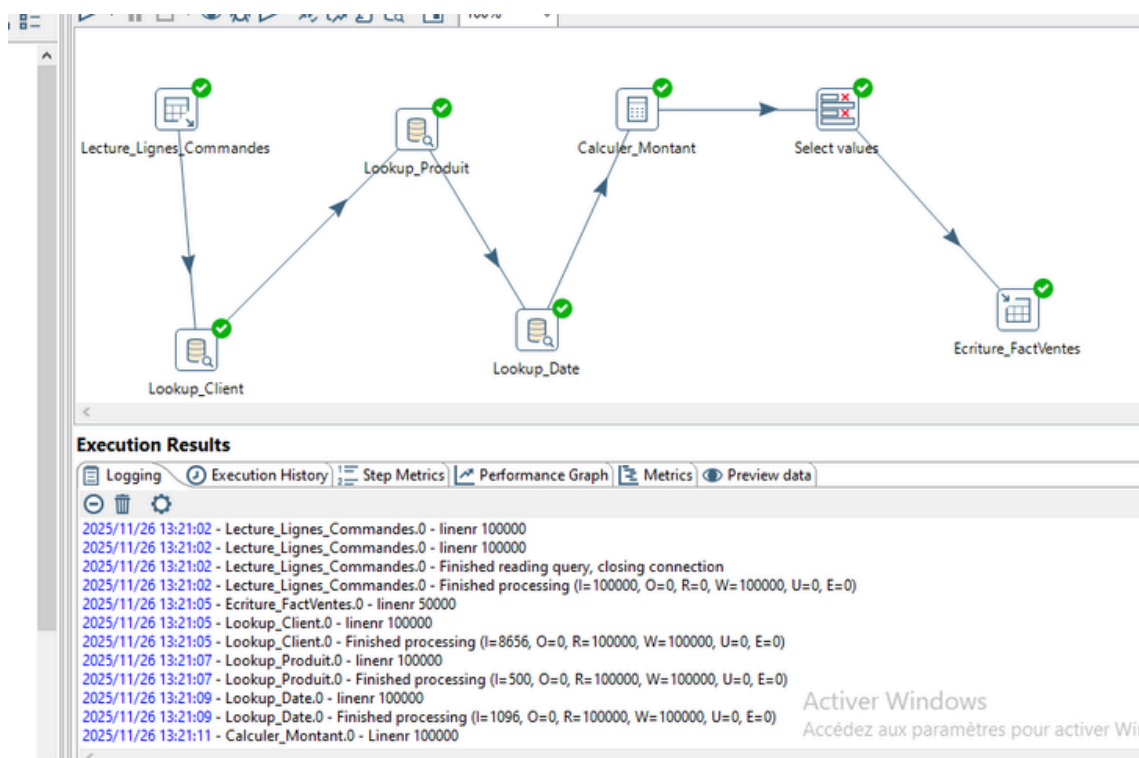
Result Grid

	id_produit_dim	id_produit_source	nom_produit	categorie
1	1		MacBook Pro Ultra	Ordinateurs
2	2		MacBook Pro Standard	Ordinateurs
3	3		MacBook Pro Ultra	Ordinateurs
4	4		MacBook Pro Standard	Ordinateurs
5	5		Lenovo ThinkPad Standard	Ordinateurs
6	6		HP Pavilion Lite	Ordinateurs
7	7		HP Pavilion Pro	Ordinateurs
8	8		Lenovo ThinkPad Ultra	Ordinateurs
9	9		Asus VivoBook Plus	Ordinateurs
10	10		MacBook Pro Ultra	Ordinateurs
11	11		MacBook Pro Pro	Ordinateurs
12	12		Asus VivoBook Standard	Ordinateurs
13	13		HP Pavilion Plus	Ordinateurs
14	14		Lenovo ThinkPad Pro	Ordinateurs
15	15		Dell XPS Plus	Ordinateurs
16	16		Lenovo ThinkPad Ultra	Ordinateurs

dimproduit1 x

**4-fact ventes.ktr** : La transformation principale. Elle :

- Joint les tables lignes\_commandes et commandes.
- Effectue trois "lookups" pour récupérer les clés de substitution (id\_client\_dim, id\_produit\_dim, id\_date\_dim) depuis les dimensions du DWH.
- Calcule le montant\_total (quantité \* prix).
- Charge les faits dans la table FactVentes.



Limit to 1000 rows

1 •

SELECT \* FROM ventes\_dwh.factventes;

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Contents:

Fetch rows:

	id_vente	id_client_dim	id_produit_dim	id_date_dim	quantite	prix_unitaire	montant_total
1	10069	401	20230124	3	1227.23	3681.00	
2	10069	221	20230124	4	1692.76	6772.00	
3	10069	278	20230124	3	170.90	513.00	
4	10069	209	20230124	3	574.86	1725.00	
5	10069	59	20230124	1	373.91	374.00	
6	18204	479	20220127	5	176.00	880.00	
7	18204	42	20220127	2	1266.43	2532.00	
8	18204	346	20220127	3	1874.22	5622.00	
9	18204	213	20220127	4	1576.62	6308.00	
10	18204	136	20220127	2	1974.56	3950.00	
11	18204	73	20220127	4	1264.26	5056.00	
12	18052	166	20240206	3	327.17	981.00	
13	18052	433	20240206	3	201.29	603.00	
14	18052	435	20240206	2	472.51	946.00	
15	18052	37	20240206	3	1548.86	4647.00	
16	18052	225	20240206	1	1411.94	1412.00	

factventes 1 x

Apply

Revert

Context

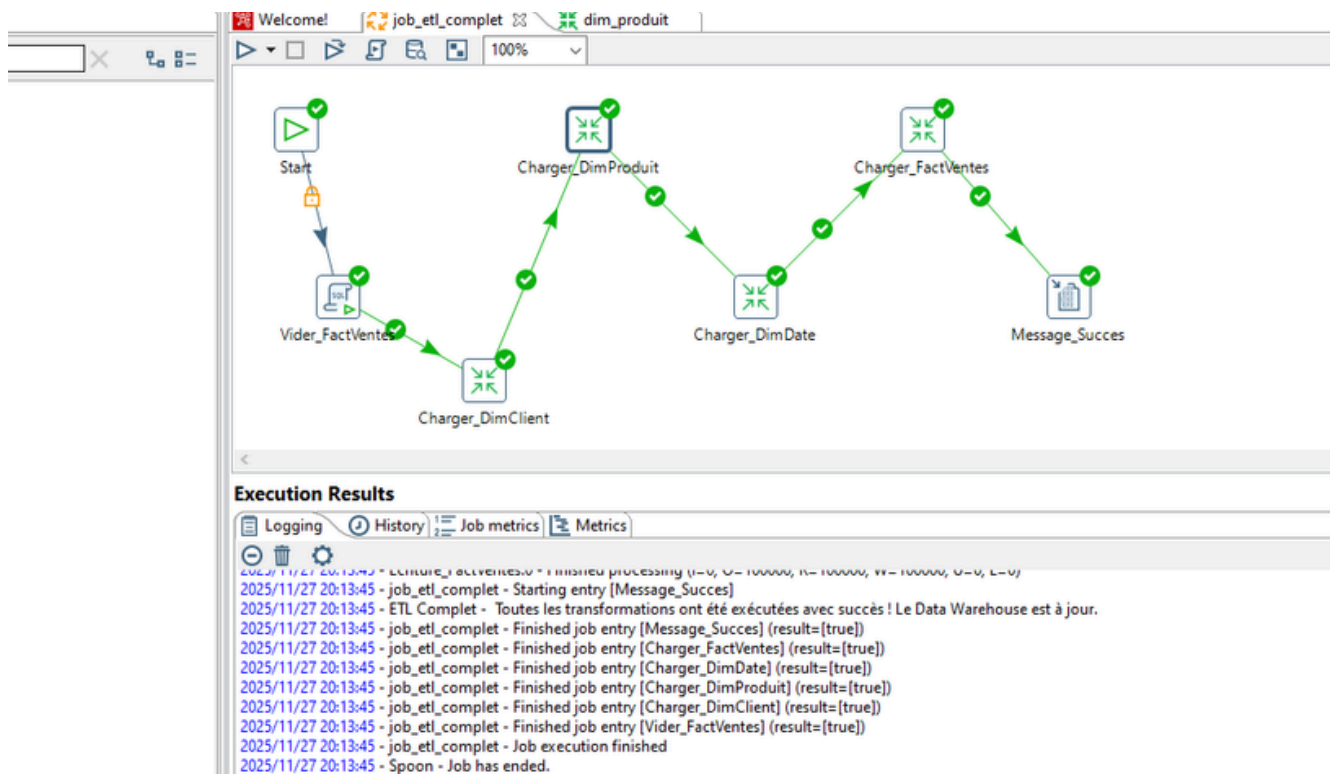
Output

Action Output

Activer Windows

Accédez aux paramètres

**Orchestration** : Un job principal (job\_etl\_complet.kjb) a été créé pour exécuter les transformations dans l'ordre logique (dimensions d'abord, faits ensuite) et gérer les dépendances.

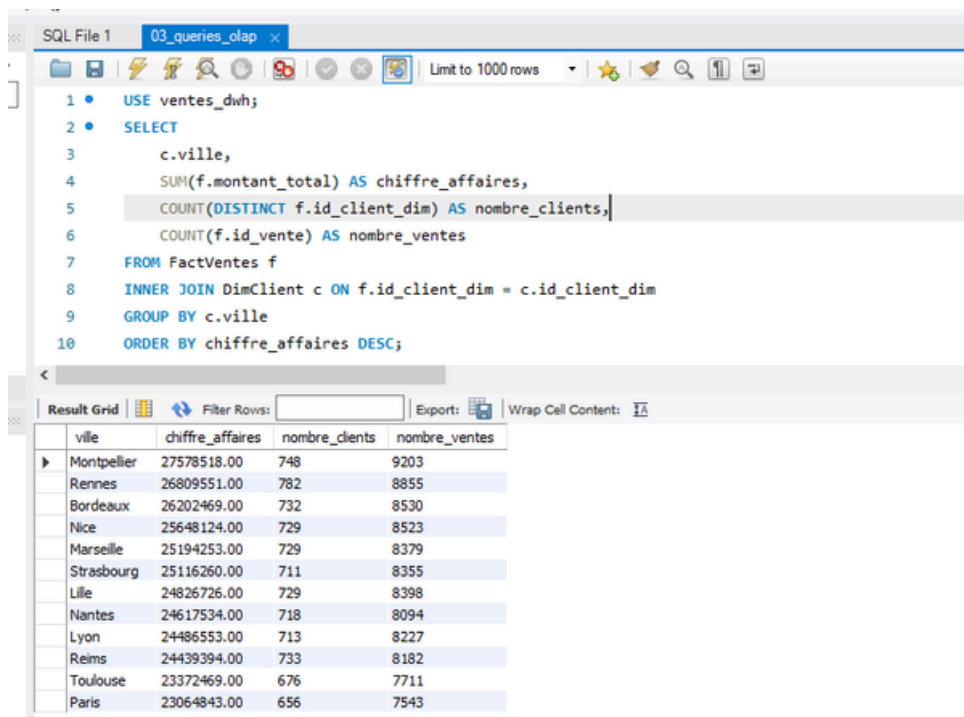


## 7- Analyses OLAP

Avec le DWH en place, des requêtes analytiques complexes deviennent simples et rapides.

### Requêtes et Interprétation Métier :

-CA par Ville : Montpellier, Rennes et Bordeaux arrivent en tête. Cela valide l'opportunité de cibler les campagnes marketing sur ces zones à fort potentiel.



The screenshot shows a SQL query editor window titled "SQL File 1" with a tab "03\_queries\_olap". The query is as follows:

```
1 • USE ventes_dwh;
2 • SELECT
3     c.ville,
4     SUM(f.montant_total) AS chiffre_affaires,
5     COUNT(DISTINCT f.id_client_dim) AS nombre_clients,
6     COUNT(f.id_vente) AS nombre_ventes
7 FROM FactVentes f
8 INNER JOIN DimClient c ON f.id_client_dim = c.id_client_dim
9 GROUP BY c.ville
10 ORDER BY chiffre_affaires DESC;
```

Below the query editor is a "Result Grid" showing the results of the query. The grid has four columns: ville, chiffre\_affaires, nombre\_clients, and nombre\_ventes. The results are sorted by chiffre\_affaires in descending order.

ville	chiffre_affaires	nombre_clients	nombre_ventes
Montpellier	27578518.00	748	9203
Rennes	26809551.00	782	8855
Bordeaux	26202469.00	732	8530
Nice	25648124.00	729	8523
Marseille	25194253.00	729	8379
Strasbourg	25116260.00	711	8355
Lille	24826726.00	729	8398
Nantes	24617534.00	718	8094
Lyon	24486553.00	713	8227
Reims	24439394.00	733	8182
Toulouse	23372469.00	676	7711
Paris	23064843.00	656	7543

-CA par Catégorie : La catégorie "Ordinateurs" génère à elle seule près de 45% du chiffre d'affaires, confirmant son statut de moteur principal des ventes.

SQL File 1 03\_queries\_olap x

Limit to 1000 rows

```

11 • SELECT
12     p.categorie,
13     SUM(f.montant_total) AS chiffre_affaires,
14     SUM(f.quantite) AS quantite_vendue,
15     COUNT(DISTINCT f.id_produit_dim) AS nombre_produits_distincts,
16     ROUND(AVG(f.prix_unitaire), 2) AS prix_moyen
17 FROM FactVentes f
18 INNER JOIN DimProduit p ON f.id_produit_dim = p.id_produit_dim
19 GROUP BY p.categorie
20 ORDER BY chiffre_affaires DESC;

```

Result Grid

	categorie	chiffre_affaires	quantite_vendue	nombre_produits_distincts	prix_moyen
►	Ordinateurs	64825326.00	60514	100	1070.22
	Téléphones	64526264.00	60122	100	1072.42
	Accessoires	60129964.00	60127	100	1000.07
	Tablettes	56774487.00	59296	100	953.92
	Montres	55100653.00	59737	100	919.41

Évolution Mensuelle : Une saisonnalité marquée est observée avec des pics systématiques en novembre-décembre (période des fêtes). Une recommandation serait d'anticiper ces pics par des préparatifs logistiques.

SQL File 1 03\_queries\_olap x

Limit to 1000 rows

```

21 • SELECT
22     d.annee,
23     d.mois,
24     d.nom_mois,
25     SUM(f.montant_total) AS chiffre_affaires,
26     COUNT(f.id_vente) AS nombre_ventes,
27     ROUND(AVG(f.montant_total), 2) AS panier_moyen
28 FROM FactVentes f
29 INNER JOIN DimDate d ON f.id_date_dim = d.id_date_dim
30 GROUP BY d.annee, d.mois, d.nom_mois
31 ORDER BY d.annee, d.mois;

```

Result Grid

	annee	mois	nom_mois	chiffre_affaires	nombre_ventes	panier_moyen
►	2022	1	Janvier	8897711.00	3003	2962.94
	2022	2	Février	8141180.00	2655	3066.36
	2022	3	Mars	8611136.00	2892	2977.57
	2022	4	Avril	8364191.00	2767	3022.84
	2022	5	Mai	7941259.00	2705	2935.77
	2022	6	Juin	8000701.00	2602	3074.83
	2022	7	Juillet	8454878.00	2887	2928.60
	2022	8	Août	8361499.00	2740	3051.64
	2022	9	Septembre	8268176.00	2717	3043.13
	2022	10	Octobre	8026616.00	2688	2986.09
	2022	11	Novembre	8328418.00	2760	3017.54
	2022	12	Décembre	8354504.00	2770	3016.07



-Top 10 Produits : 70% des ventes proviennent de seulement 10 références. Ces produits "stars" doivent être particulièrement surveillés en termes de stock et de mise en avant.

SQL File 1 03\_queries\_olap x

Limit to 1000 rows

```

32 • SELECT
33     p.nom_produit,
34     p.categorie,
35     SUM(f.quantite) AS quantite_totale_vendue,
36     SUM(f.montant_total) AS chiffre_affaires,
37     ROUND(AVG(f.prix_unitaire), 2) AS prix_moyen
38 FROM FactVentes f
39 INNER JOIN DimProduit p ON f.id_produit_dim = p.id_produit_dim
40 GROUP BY p.id_produit_dim, p.nom_produit, p.categorie
41 ORDER BY quantite_totale_vendue DESC
42 LIMIT 10;

```

Result Grid

	nom_produit	categorie	quantite_totale_vendue	chiffre_affaires	prix_moyen
▶	Apple Watch Plus	Montres	751	1334527.00	1776.81
	Clavier mÃ©canique Ultra	Accessoires	740	1277240.00	1726.43
	Huawei MatePad Lite	Tablettes	734	136524.00	186.37
	Xiaomi 13 Plus	TÃ©lÃ©phones	725	1046175.00	1442.98
	HP Pavilion Plus	Ordinateurs	714	369852.00	518.16
	iPhone 14 Standard	TÃ©lÃ©phones	707	1260581.00	1782.59
	Asus VivoBook Ultra	Ordinateurs	704	1265792.00	1797.82
	Garmin Forerunner Lite	Montres	704	772992.00	1097.54
	Webcam HD Ultra	Accessoires	703	1120582.00	1594.23
	OnePlus 11 Standard	TÃ©lÃ©phones	702	819234.00	1167.47

Result 4 x

Limit to 1000 rows

```

41 ORDER BY quantite_totale_vendue DESC
42 LIMIT 10;
43 • SELECT
44     d.annee,
45     d.trimestre,
46     p.categorie,
47     SUM(f.montant_total) AS chiffre_affaires,
48     SUM(f.quantite) AS quantite_vendue
49 FROM FactVentes f
50 INNER JOIN DimDate d ON f.id_date_dim = d.id_date_dim
51 INNER JOIN DimProduit p ON f.id_produit_dim = p.id_produit_dim
52 GROUP BY d.annee, d.trimestre, p.categorie
53 ORDER BY d.annee, d.trimestre, chiffre_affaires DESC;

```

Result Grid

	annee	trimestre	categorie	chiffre_affaires	quantite_vendue
▶	2022	1	TÃ©lÃ©phones	5474825.00	5004
	2022	1	Ordinateurs	5381872.00	5081
	2022	1	Accessoires	5157471.00	5135
	2022	1	Tablettes	4935860.00	5287
	2022	1	Montres	4699999.00	5208
	2022	2	Ordinateurs	5573268.00	5183
	2022	2	TÃ©lÃ©phones	5493563.00	5091
	2022	2	Accessoires	4660607.00	4617
	2022	2	Tablettes	4392008.00	4516

## **8. Visualisations Power BI**

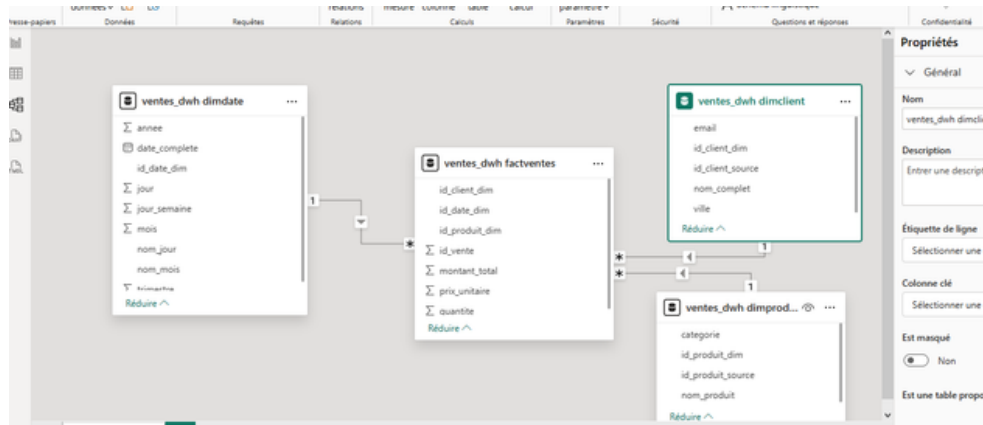


## 8. Visualisations Power BI

Le tableau de bord interactif synthétise l'ensemble des analyses pour une prise de décision rapide.

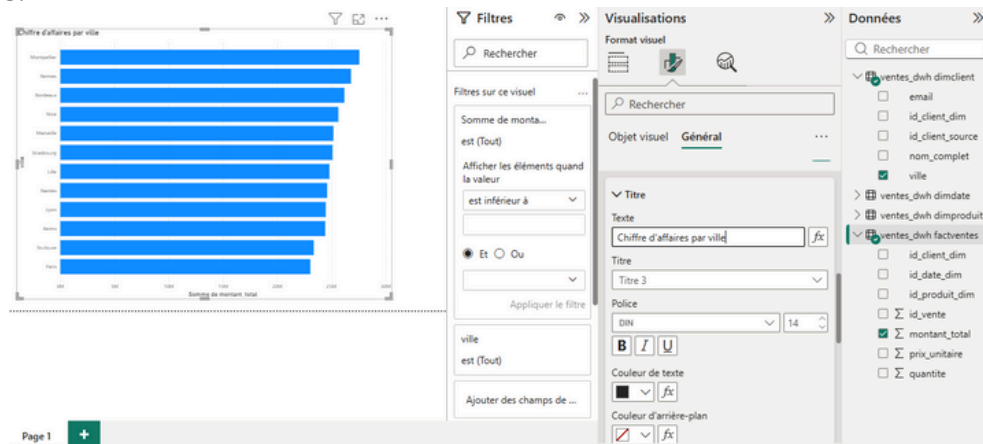
### Description du Dashboard :

Le rapport Power BI se connecte directement au DWH (ventes\_dwh) .

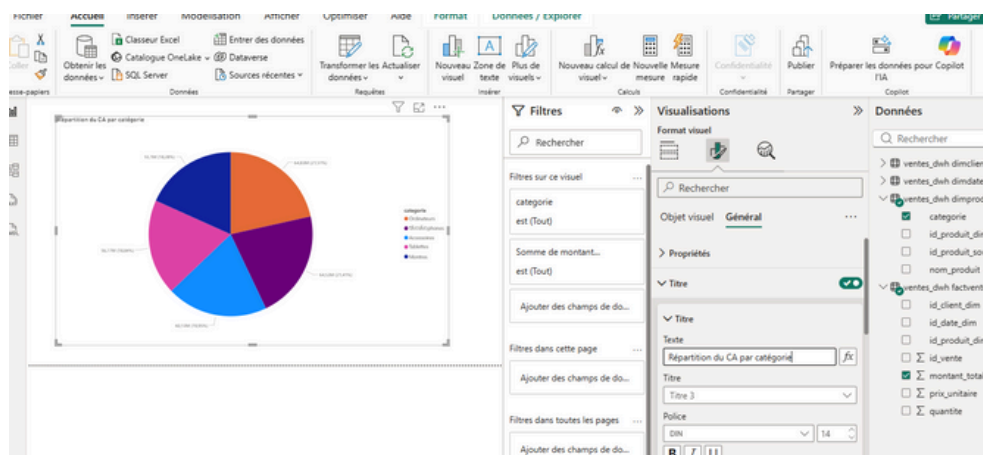


et présente :

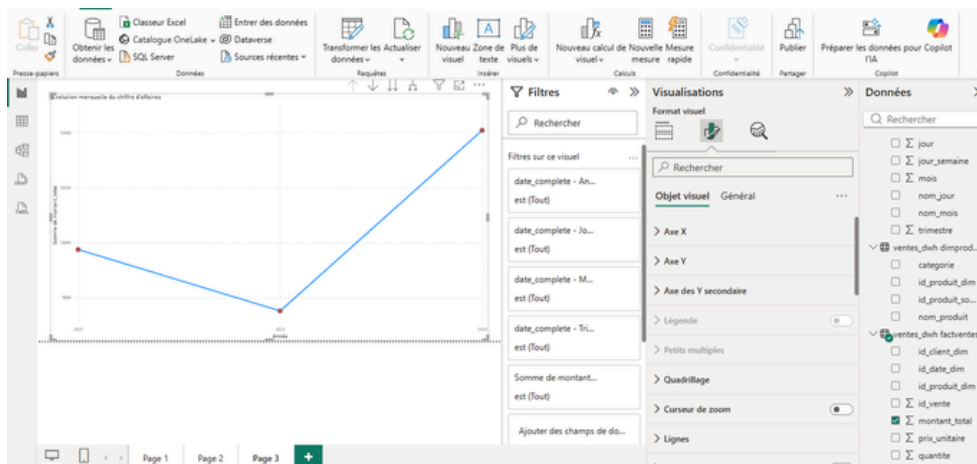
- Des indicateurs clés (KPI) : CA Total, Panier Moyen, Nombre de Ventes, Clients Uniques.
- Des segments (slicers) interactifs : Filtrage par Année et Catégorie.
- Quatre visualisations principales :
  - CA par Ville (Graphique à barres) : Permet d'identifier les marchés géographiques les plus rentables.



Répartition du CA par Catégorie (Graphique en secteurs) : Montre la contribution de chaque famille de produits.



-Évolution Mensuelle du CA (Graphique en courbes): Visualise les tendances et la saisonnalité.



### Insights Métier Tirés :

Le dashboard permet de constater en un clin d'œil que la performance de TechStore est tirée par les ventes d'ordinateurs à Montpellier pendant le dernier trimestre. Il facilite ainsi des décisions telles que l'ajustement des stocks, le ciblage des campagnes publicitaires, ou l'évaluation de l'opportunité d'ouvrir un point de vente physique.

## 9. Conclusions et Recommandations

**Synthèse :** Ce projet a permis de construire avec succès un pipeline BI opérationnel pour TechStore. L'architecture mise en place répond aux problématiques initiales en fournissant des insights rapides, fiables et actionnables à partir des données transactionnelles.

### Apprentissages :

- Maîtrise de la différence fondamentale entre modèles OLTP (normalisé) et OLAP (dénormalisé).
- Expérience pratique sur l'outil ETL Pentaho pour l'automatisation des flux.
- Conception d'un schéma en étoile adapté aux besoins analytiques.
- Création d'un tableau de bord interactif orienté métier avec Power BI.

### Améliorations Possibles :

1. Industrialisation : Planifier l'exécution du job Pentaho via Kitchen et un planificateur de tâches (cron/Windows Scheduler) pour un rafraîchissement quotidien automatique.
2. Enrichissement des Données : Intégrer des données externes (météo, jours fériés) dans la dimension DimDate pour affiner l'analyse des ventes.
3. Power BI Cloud : Publier le rapport sur le service Power BI pour le partage en ligne et l'accès mobile.
4. Gestion des Dimensions Lentes (SCD) : Implémenter dans Pentaho la gestion des changements historiques dans les dimensions (ex: suivi des changements d'adresse client).