

Rapport de Projet de Business Intelligence

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

FORMATION : 4^e 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.

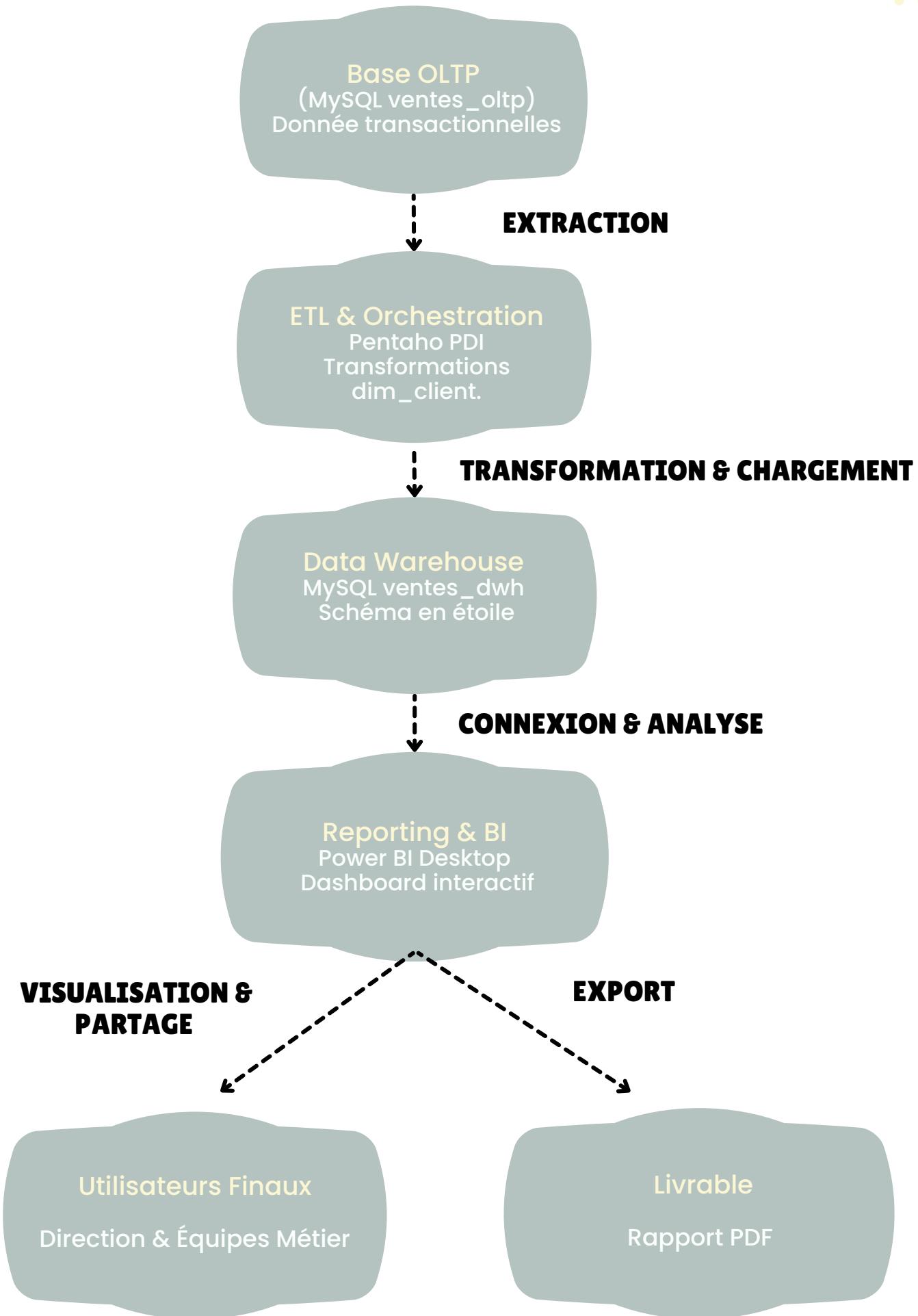
Objectifs :

- 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 the MySQL Workbench interface. On the left, there's a tree view of database objects: 'e', 'e_inscription', 's', 'sKeys', 't', 'tmandes', 'tres', 's', and 'tmandes'. Below the tree, there's some schema information: 'int AI PK', 'int', 'int', 'int', and 'decimal(10,2)'. In the center, a query editor window displays the following SQL code:

```
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 query editor is a 'Result Grid' window showing the results of the first query:

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.

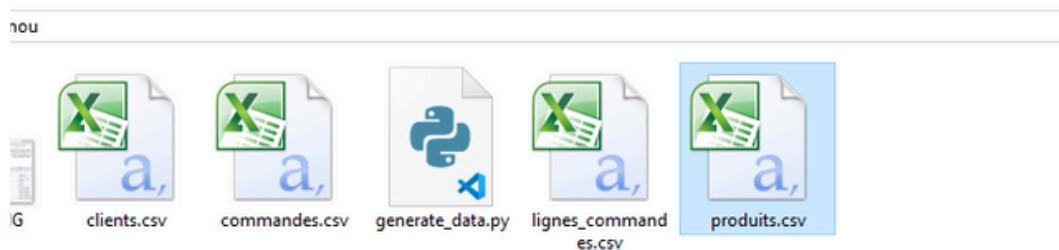
4. Génération des Données

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.
 - Lignes de commande : 100 000, assurant un lien cohérent entre commandes et produits.



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 shows the MySQL Workbench interface with two main panes. The left pane displays the database structure with tables like clients, factventes, and dimdate. The right pane shows the SQL editor with the following code:

```
55 • SHOW TABLES;
56 • DESCRIBE DimClients;
57 • DESCRIBE DimProduit;
58 • DESCRIBE DimDate;
59 • DESCRIBE FactVentes;
60
61
```

Below the code, the results grid shows the structure of the DimClients table:

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

SQL File 1* ×

```

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

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

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

Result 43 ×

Output

Action Output

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

SQL File 1* ×

```

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

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

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

Result 41 ×

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

SQL File 1* ×

```

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

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

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

Result 42 ×

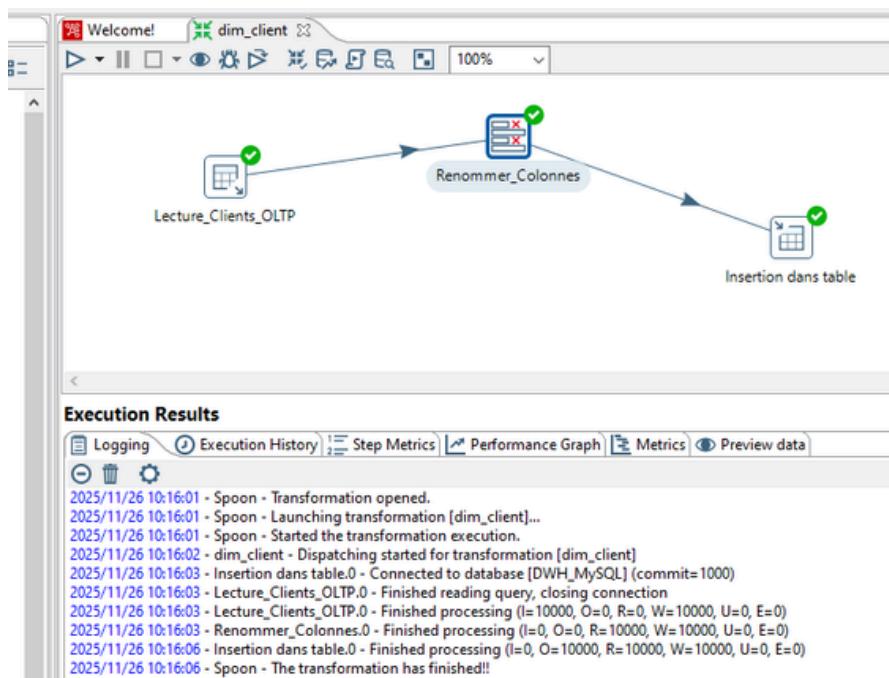
Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

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.



Schemas

- ventes_dwh
 - Tables
 - dimclient
 - dimdate
 - dimproduit
 - factventes
 - Views
 - Stored Procedures
 - Functions
- ventes_olap

Table: dimclient

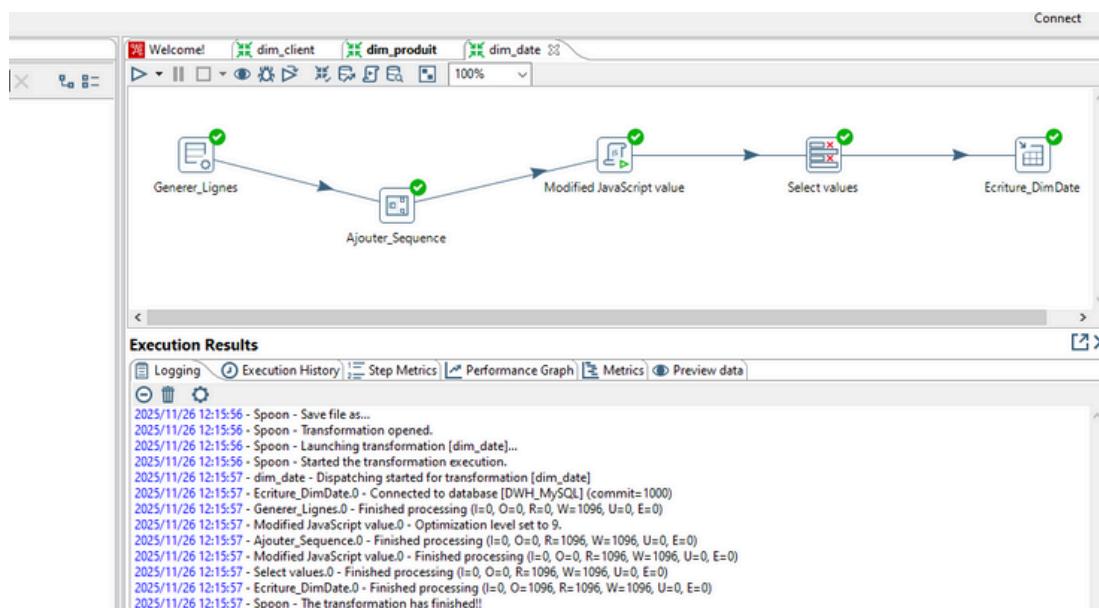
Columns:

Column	Type
id_client_dim	int AI PK
id_client_source	int
nom_complet	varchar(200)
email	varchar(150)
ville	varchar(100)

Result Grid

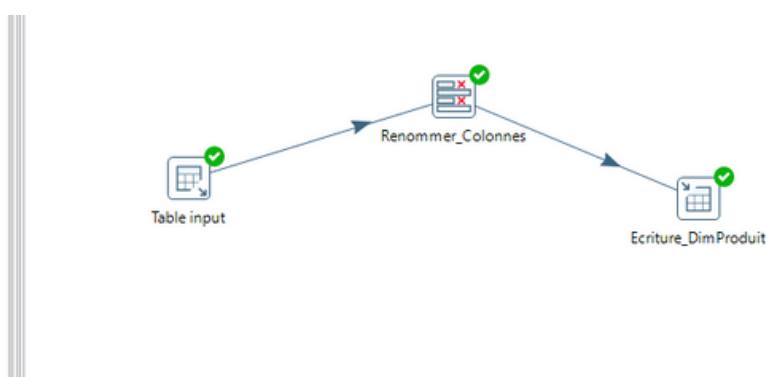
	id_client_dim	id_client_source	nom_complet	email	ville
1	1		Frédéric Riou	client1@volo.fr	Rennes
2	2		Maurice Moulin	client2@laposte.net	Lyon
3	3		Nicole Lebâth	client3@hotmail.fr	Paris
4	4		Arthur Briand	client4@volo.fr	Reims
5	5		Nicolas Carpenter	clients5@wanadoo.fr	Nice
6	6		Gabrielle Michel	clients6@sfr.fr	Toulouse
7	7		Ã‰milie Bouth	client7@noos.fr	Toulouse
8	8		Jacqueline Pons	client8@bouytel.fr	Marseille
9	9		Susan Thierry	client9@free.fr	Reims
10	10		Zacharie Briand	client10@bouytel.fr	Lyon
11	11		Paulette Fabre	client11@gmail.com	Rennes
12	12		Alex Morin	client12@yahoo.fr	Reims
13	13		Josette Barre	client13@bouytel.fr	Bordeaux
14	14		CÃ©line Chauveau	client14@yahoo.fr	Lyon
15	15		FranÃ§oise Chev...	client15@club-intern...	Lille
16	16		AgnÃ©s Maurice	client16@sfr.fr	Strasbourg

2-dim_date.ktr : Génère dynamiquement la table de temps DimDate pour la période 2022-2024.



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

3-dim_produit.ktr : Lit la table produits et charge DimProduit.



Filter objects

```
1 • SELECT * FROM ventes_dwh.dimproduit;
```

Tables

- dimclient
- dimdate
- dimproduit
- factventes

Views

Stored Procedures

Functions

Imministration Schemas

formation

Table: dimproduit

Columns:

id_produit_dim	int AI PK
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16

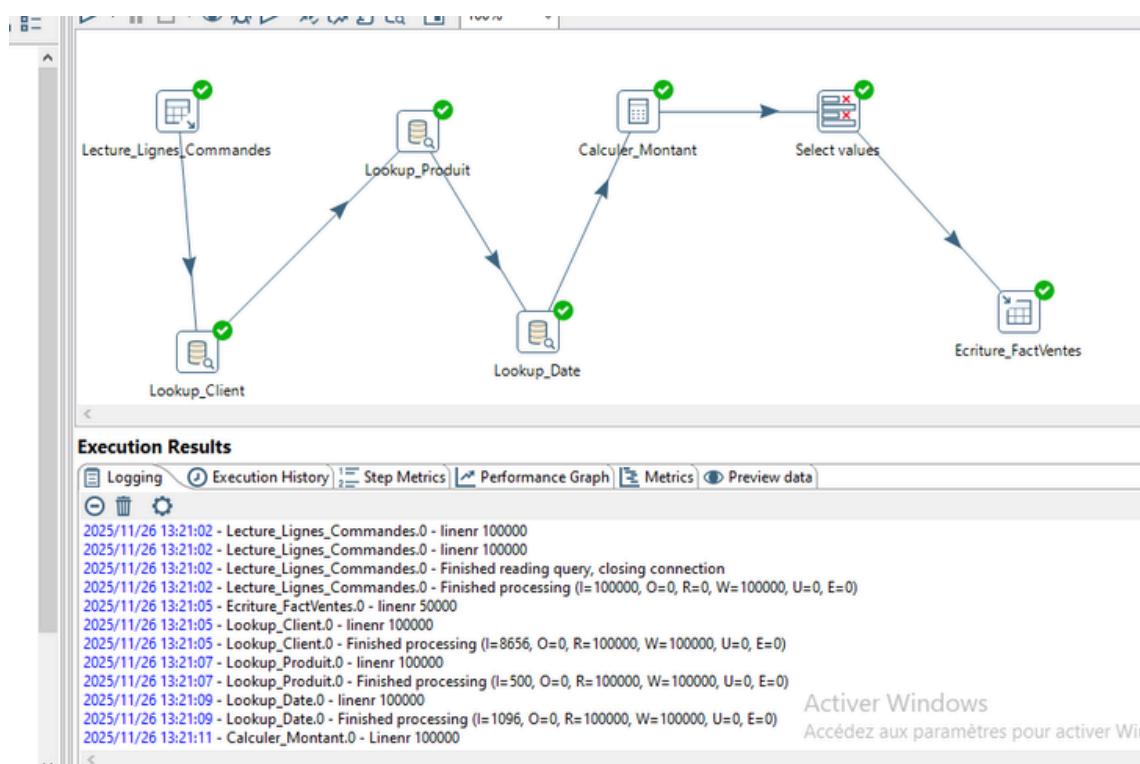
nom_produit

categorie

dimproduit1

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.



Spoon - Job has ended.

1 • SELECT * FROM ventes_dwh.factventes;

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Fetch rows: | Result Grid | Form Editor | Field Types | Query Stats | Context

factventes 1 x

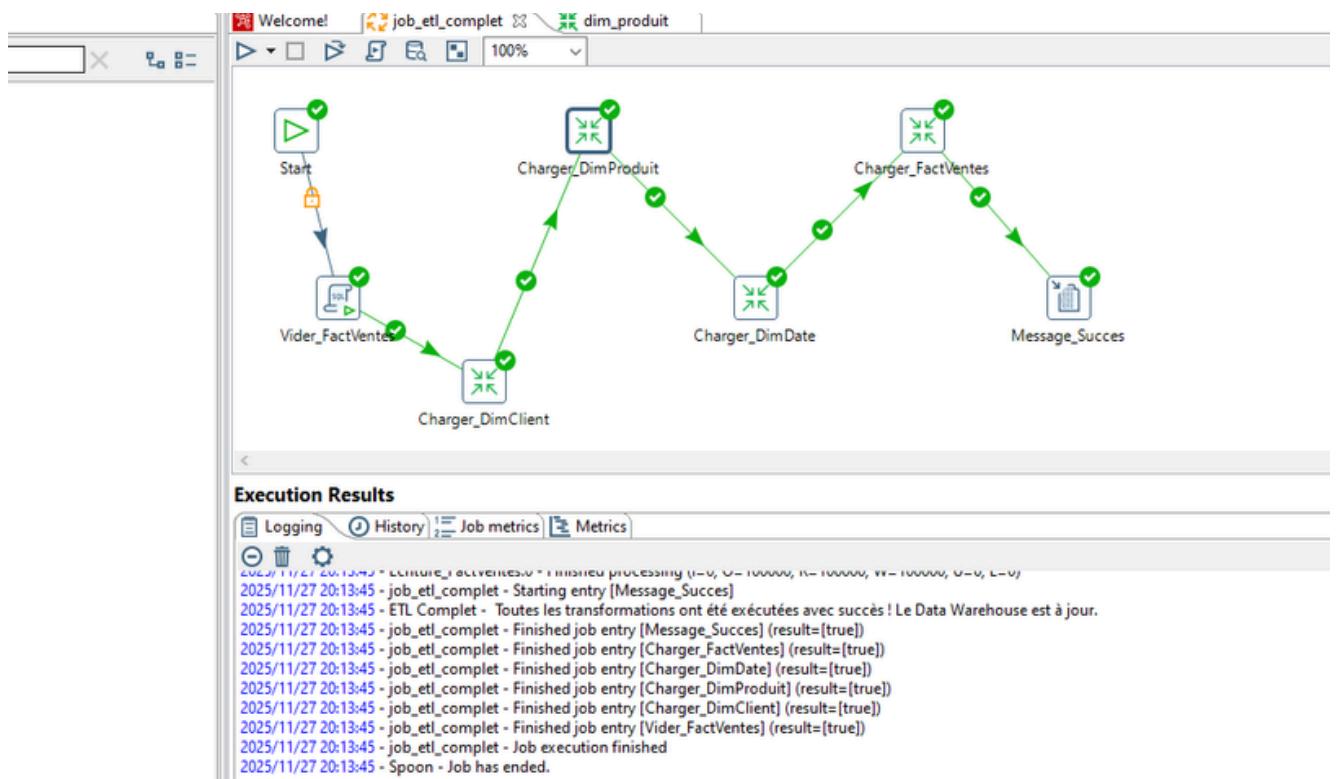
Output

Action Output

Activer Windows

Accédez aux paramètres p

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 labeled "03_queries_olap". The query itself is:

```
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, there is a "Result Grid" table with four columns: "ville", "chiffre_affaires", "nombre_clients", and "nombre_ventes". The data is as follows:

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

```

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 | Filter Rows: Export: Wrap Cell Content:

	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

```

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 | Filter Rows: Export: Wrap Cell Content:

	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

```

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 | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

nom_produit	categorie	quantite_totale_vendue	chiffre_affaires	prix_moyen
Apple Watch Plus	Montres	751	1334527.00	1776.81
Clavier mécanique Ultra	Accessoires	740	1277240.00	1726.43
Huawei MatePad Lite	Tablettes	734	136524.00	186.37
Xiaomi 13 Plus	Téléphones	725	1046175.00	1442.98
HP Pavilion Plus	Ordinateurs	714	369852.00	518.16
iPhone 14 Standard	Téléphones	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	Téléphones	702	819234.00	1167.47

Result 4 x

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 | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

annee	trimestre	categorie	chiffre_affaires	quantite_vendue
2022	1	Téléphones	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	Téléphones	5493563.00	5091
2022	2	Accessoires	4660607.00	4617
2022	2	Tablettes	4392008.00	4516

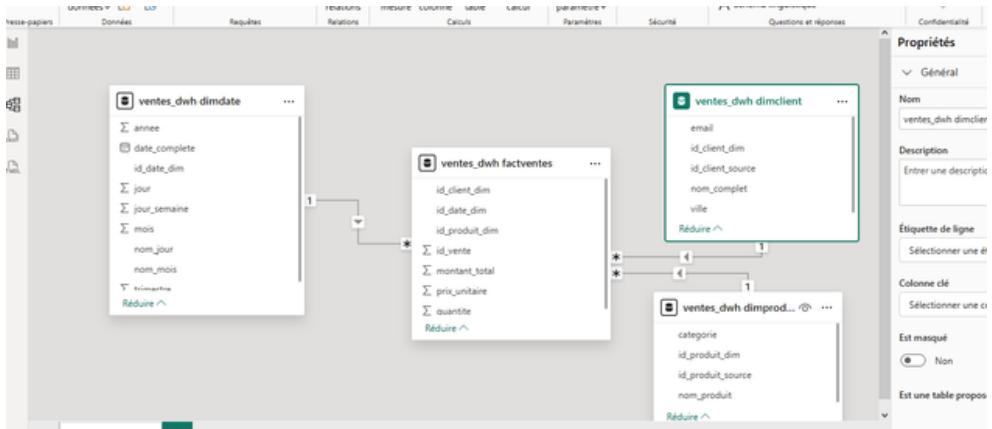
8. Visualisations Power BI

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.

Visualisation : Chiffre d'affaires par ville

Filtres :

- Somme de montant... est (Tout)
- Et / Ou
- ville est (Tout)

Format visuel :

- Titre : Chiffre d'affaires par ville
- Police : DIN, 14
- Couleur de texte : Noire
- Couleur d'arrière-plan : Blanche

Données :

- ventes_dwh.dimclient (checked): email, id_client_dim, id_client_source, nom_complet, ville
- ventes_dwh.dimdate
- ventes_dwh.dimproduit
- ventes_dwh.factventes (checked): id_vente, montant_total, prix_unitaire, quantité

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

Visualisation : Répartition du CA par catégorie

Filtres :

- categorie est (Tout)
- Somme de montant... est (Tout)

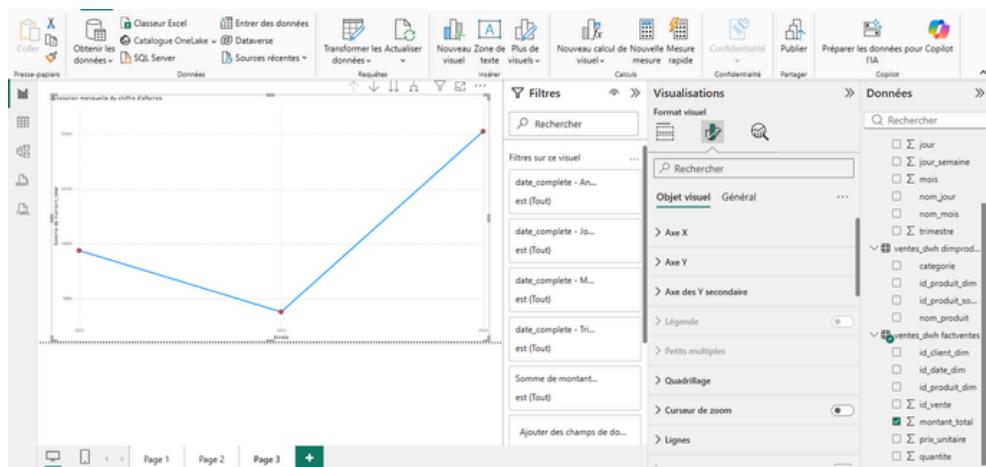
Format visuel :

- Titre : Répartition du CA par catégorie
- Police : DIN, 14

Données :

- ventes_dwh.dimclient
- ventes_dwh.dimdate
- ventes_dwh.dimproduit (checked): categorie, id_produit_dim, id_produit_source, nom_produit
- ventes_dwh.factventes (checked): id_vente, montant_total, prix_unitaire, quantité

-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).