

MISE EN PLACE ET AMÉLIORATION D'UN ENTREPÔT DE DONNÉES POUR L'ANALYSE AVEC POWER BI



PRÉSENTÉ PAR :
Mouflla Faissal

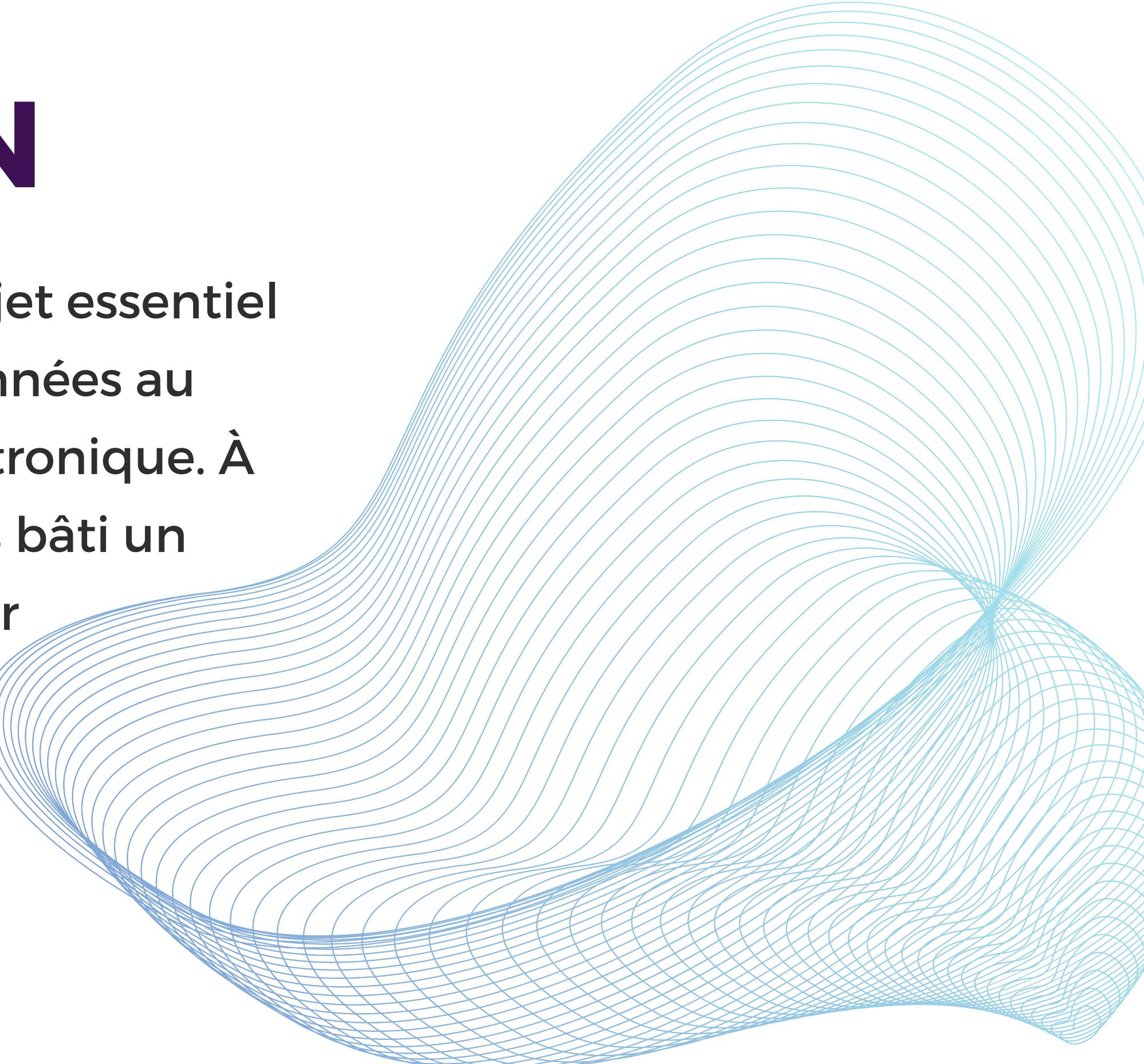
PRÉSENTÉ À M :
YOUCODE TEAM

PLAN

- Introduction
- Planification
- Exploration des données
- Application des Politiques RGPD sur les Données Sensibles
- Modélisation du DataWarehouse
- ETL en utilisant Talend
- Data Marts Physiques
- Analytique avec Power BI
- Optimisation
- Validation de la Logique de Transformation
- Autorisation
- Conclusion

INTRODUCTION

Aujourd'hui nous plongeons dans un projet essentiel qui vise à révolutionner la gestion des données au sein d'une plateforme de commerce électronique. À l'aide de SQL Server et Talend, nous avons bâti un entrepôt de données efficace pour stocker et traiter nos informations. De plus, l'analyse approfondie de ces données avec Power BI offre un regard perspicace sur notre activité.



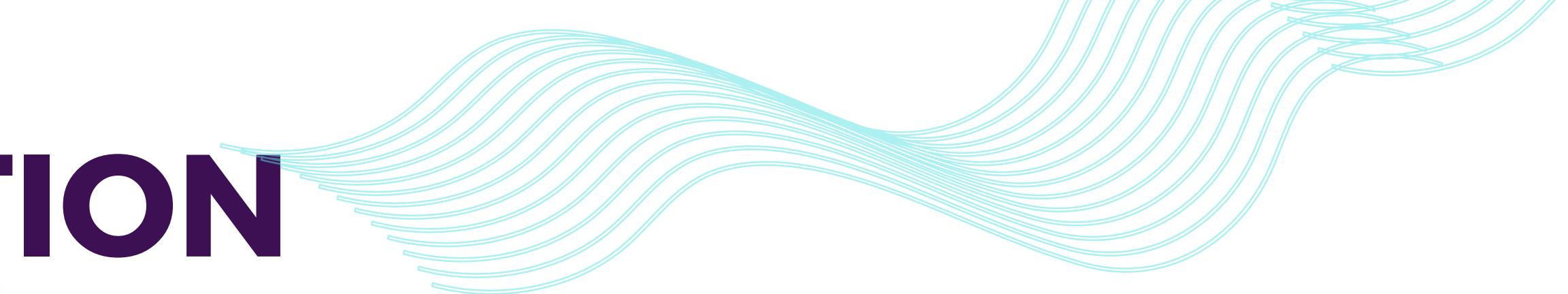
PLANIFICATION

The image shows a GitHub project board with three columns: Todo, In Progress, and Done. Each column has a header with a status icon and a count of items. Below each header is a brief description of the status. The Todo column contains one item: "Task 11: Livrables". The In Progress column contains two items: "Task 10: Autorisation" and "Task 9: Valider la Logique de Transformation". The Done column contains eight items: "Task 1 : Exploration", "Task 2: Create a Fast Constellation Schema", "Task 3: RGPD", "Task 4: Data Splitter", "Task 5: ETL using Talend", "Task 6: Create Data Marts", "Task 7: Analytique avec Power BI", and "Task 8: Finalisation". At the bottom of each column, there is a "+ Add item" button.

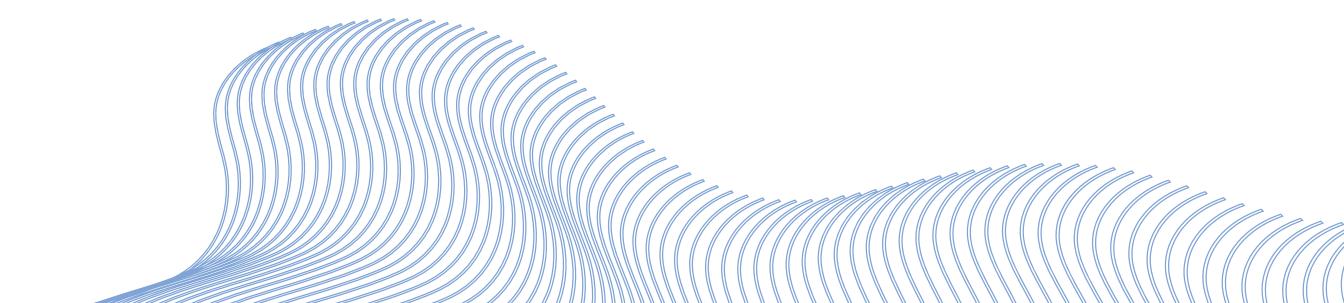
Todo	In Progress	Done
This item hasn't been started	This is actively being worked on	This has been completed
Draft Task 11: Livrables	Draft Task 10: Autorisation	Draft Task 1 : Exploration
	Draft Task 9: Valider la Logique de Transformation	Draft Task 2: Create a Fast Constellation Schema
		Draft Task 3: RGPD
		Draft Task 4: Data Splitter
		Draft Task 5: ETL using Talend
		Draft Task 6: Create Data Marts
		Draft Task 7: Analytique avec Power BI
		Draft Task 8: Finalisation

Avant de plonger dans les détails de notre projet, il est essentiel de bien l'organiser. Pour cela, j'ai utilisé GitHub comme un outil de gestion de projet. Cela nous a permis de planifier, suivre et collaborer efficacement tout au long de notre parcours. Une planification bien pensée est la première étape vers le succès de notre entreprise.

EXPLORATION



La phase d'exploration des données avec Python a joué un rôle crucial dans la préparation de nos données pour l'entrepôt de données. Grâce à cette étape, nous avons acquis une compréhension approfondie de nos données. En utilisant des bibliothèques puissantes telles que Pandas, nous avons identifié plusieurs aspects clés dans nos données



- Repérage de valeurs manquantes dans certaines colonnes, notamment **CustomerEmail** (507 valeurs manquantes) et **SupplierContact** (484 valeurs manquantes).

```
print(df.isnull().sum(axis=0))
```

Date	0
ProductName	0
ProductCategory	0
ProductSubCategory	0
ProductPrice	0
CustomerName	0
CustomerEmail	507
CustomerAddress	0
CustomerPhone	0
CustomerSegment	0
SupplierName	0
SupplierLocation	0
SupplierContact	484
ShipperName	0
ShippingMethod	0
QuantitySold	0
TotalAmount	0

- Identification de 10 doublons dans nos données, nécessitant une attention particulière.

```
duplicates = df[df.duplicated(keep='first')].shape[0]  
duplicates
```

```
10
```

- Observation de certaines valeurs de la colonne ProductName contenant "NonExistentProduct".

	Date	ProductName
1	2023-02-11	NonExistentProduct

- Détection de certaines valeurs de la colonne ProductCategory avec "InvalidCategory".

	Date	ProductName	ProductCategory
2	2021-11-12	Angela	InvalidCategory

- Ainsi que de certaines valeurs de la colonne ProductPrice avec "InvalidPrice".

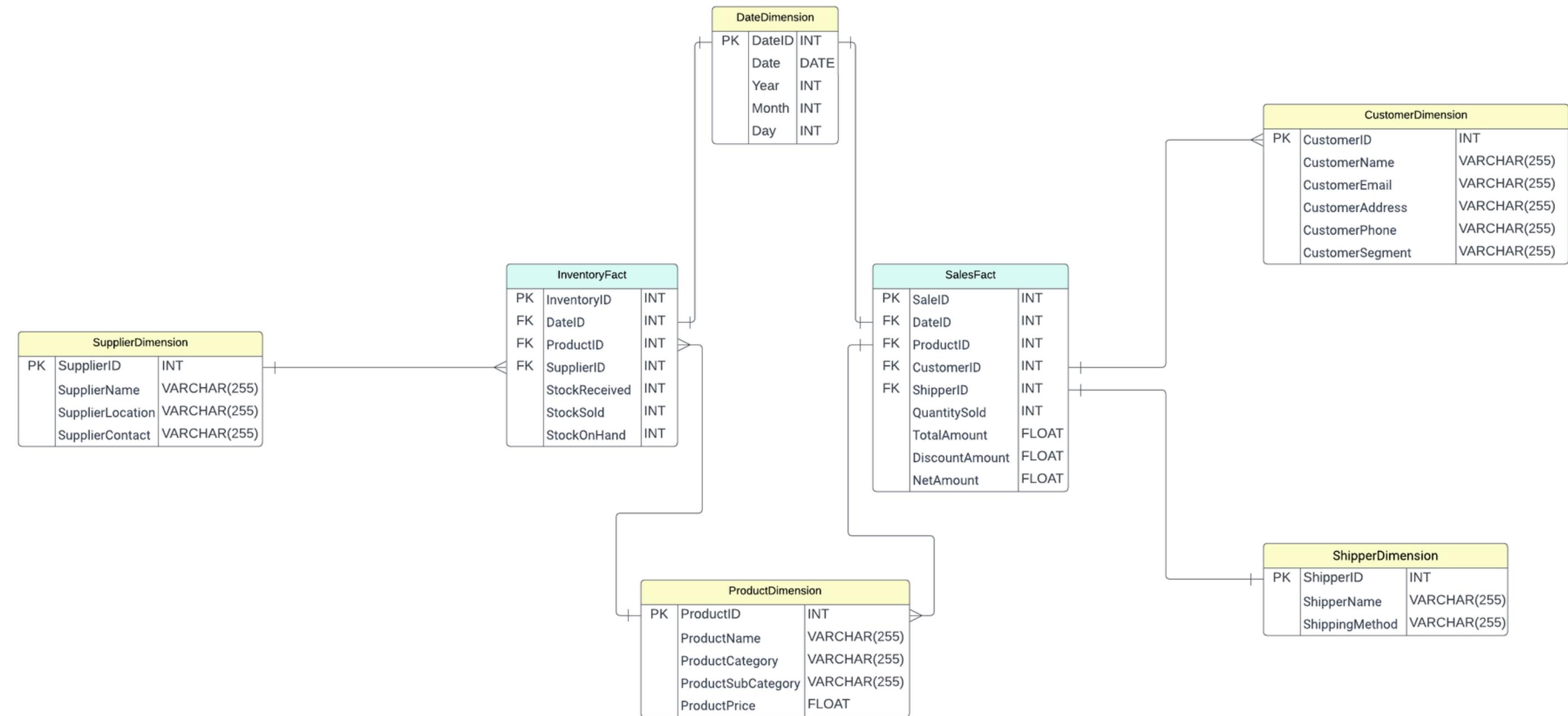
	Date	ProductName	ProductCategory	ProductSubCategory	ProductPrice
23	2022-04-19	NonExistentProduct	Home & Garden	Decor	InvalidPrice

RGPD

Après une exploration minutieuse de nos données avec Python, nous avons identifié les colonnes contenant des informations sensibles nécessitant une protection en conformité avec le RGPD. Cela inclut des données personnelles telles que le nom des clients, leur adresse e-mail, leur adresse physique, leur numéro de téléphone, ainsi que la localisation de nos fournisseurs. Pour assurer la confidentialité de ces données et respecter les normes du RGPD, nous avons choisi d'appliquer le chiffrement.

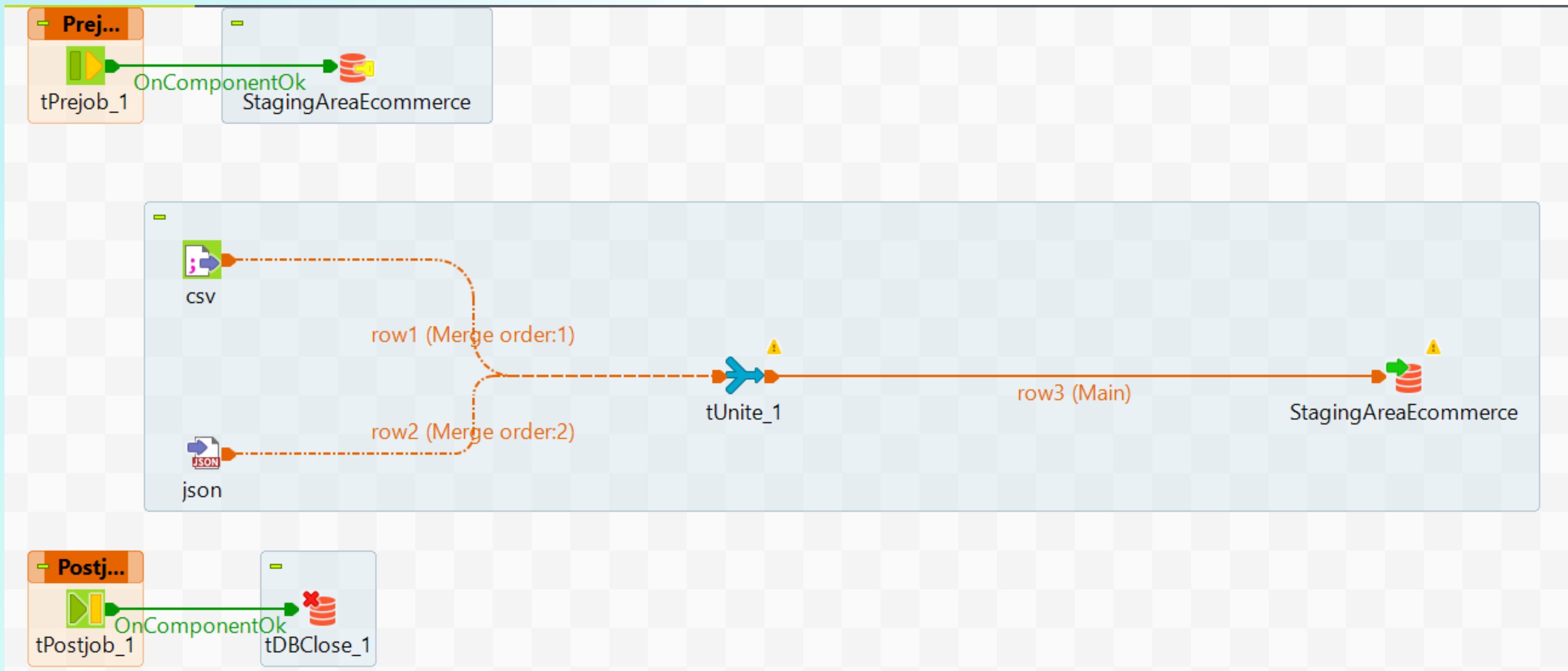
MODÉLISATION

La phase de modélisation est cruciale pour la mise en place d'un entrepôt de données efficace. Pour cette étape, nous avons conçu un schéma de base de données qui prend en compte la structure de nos données.



ETL EN UTILISANT TALEND

Extraction des Données



Transformation des Données

+Gestion des données manquantes :

-J'ai remplacé personnellement les valeurs manquantes dans les colonnes 'CustomerEmail' par "Not exist" et 'SupplierContact' par "NaN".

+Gestion des doublons :

-J'ai utilisé le composant TUnique dans Talend pour supprimer les doublons dans nos données.

+Traitement des valeurs "NonExistentProduct" dans 'ProductName' :

-J'ai pris en charge le remplacement des valeurs "NonExistentProduct" par les valeurs correspondantes dans la colonne 'ProductSubCategory'.

+Traitement des valeurs "InvalidCategory" dans 'ProductCategory' :

-J'ai effectué le remplacement des valeurs "InvalidCategory" par les valeurs correspondantes dans la colonne 'ProductName'.

+Traitement des valeurs "InvalidPrice" dans 'ProductPrice' :

-J'ai recalculé personnellement les valeurs "InvalidPrice" en utilisant la formule TotalAmount divisé par QuantitySold, et j'ai veillé à la conversion du résultat en type float.

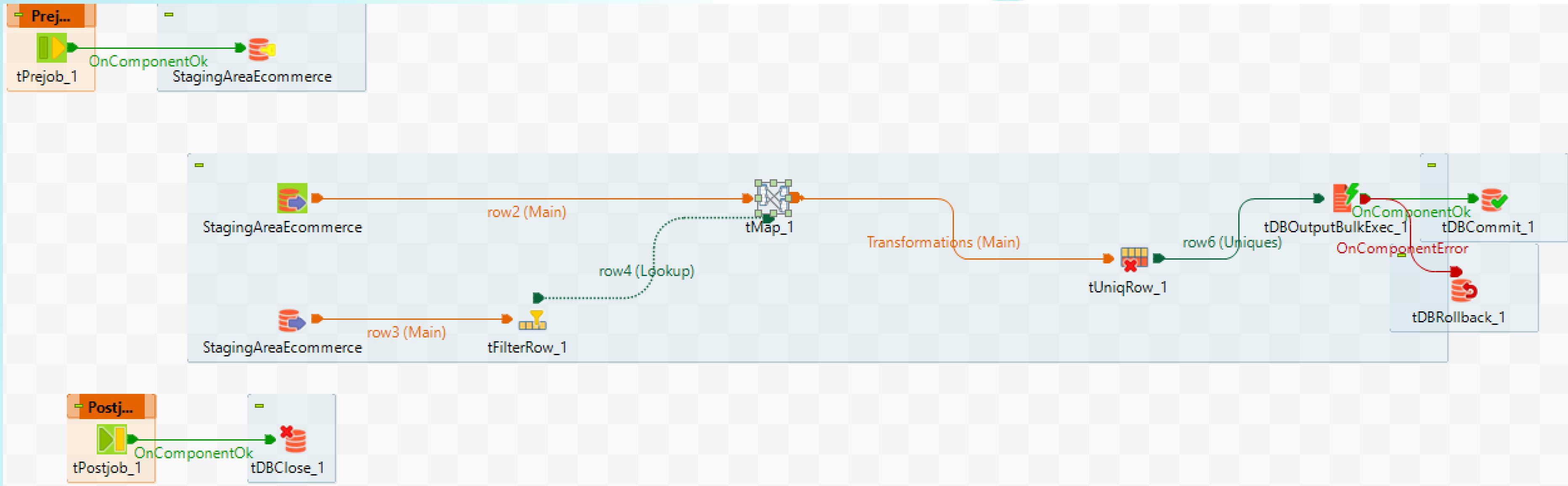
+Normalisation des colonnes de type chaîne de caractères :

-J'ai personnellement effectué la transformation de toutes les valeurs des colonnes de type chaîne de caractères en minuscules (lowercase).

+Chiffrement des colonnes sensibles :

-J'ai mis en œuvre des techniques de chiffrement pour protéger les données sensibles, notamment les noms des clients, les adresses e-mail, les adresses physiques, les numéros de téléphone, et la localisation des fournisseurs, conformément aux réglementations RGPD.

CustomerName	CustomerEmail	CustomerAddress
KkDc2J94AGk8Lzh6j4Na7epWCti6tRURJqUN9UUUnMYU=	pn798UboHBbyYirdZfsRVnFUbo20kSBMhcBKr16QZV8=	zWhLNlqxeFkOeo/cq19tjDF0xNapHTV4OZ64hEErAm/VfgF
ozwzcW5zzG6Cr/WBxHsGJA==	1TSmroGTdwHEHpdrP7/cQon0MNK9SauR7AQP7bmu5Ko=	+r0gEKX2Wnqx/akr0LOQm/JRKQ1tQhAhhm5cr+0sUyHkybE
SR3c/4x8lq69n0baDejFyw==	SmieZWVwSL2pHSKnI22qwobdlGALAJeE+sTso+9utyU=	awKfsXpTZmgVzUBkbGHhWeQDHJMSa9Q9C1h+gDBEuZ
mjh5cGgh3a8ql4H8sRcfKA==	hCZ7FJQsmuu0KZKtd/TgjlbdlGALAJeE+sTso+9utyU=	Y8E9ilb6zbFQrcP5ubRrd5bSwURJqDYapuCuFKLWGP3+w
uxWKBwj3OoOnjdmlrbIzMA==	QBg6KWdljmLo9al0GRLmgobdlGALAJeE+sTso+9utyU=	vrl06nMFmzqqqmM7rwIMbCY13Fili6dI6rHLzW4VJJbganjnL



Find :

row2

Column

- Date
- ProductName
- ProductCategory
- ProductSubCategory
- ProductPrice
- CustomerName
- CustomerEmail
- CustomerAddress
- CustomerPhone
- CustomerSegment
- SupplierName
- SupplierLocation
- SupplierContact
- ShipperName
- ShippingMethod
- QuantitySold
- TotalAmount
- DiscountAmount

Var

Transformations

Expression	Column
TalendDate.isDate(row2.Date, "yyyy-MM-dd")	Date
!row2.ProductName.equals("NonExistent")	ProductName
row2.ProductCategory.equals("InvalidCategory")	ProductCategory
StringHandling.DOWNCASE(row2.ProductSubCategory)	ProductSubCategory
row2.ProductPrice.equals("InvalidPrice")	ProductPrice
Encryption.encrypt(row2.CustomerName)	CustomerName
Encryption.encrypt(row2.CustomerEmail)	CustomerEmail
Encryption.encrypt(row2.CustomerAddress)	CustomerAddress
Encryption.encrypt(row2.CustomerPhone)	CustomerPhone
StringHandling.DOWNCASE(row2.CustomerSegment)	CustomerSegment
StringHandling.DOWNCASE(row2.SupplierName)	SupplierName
StringHandling.DOWNCASE(row2.SupplierLocation)	SupplierLocation
Encryption.encrypt(row2.SupplierContact)	SupplierContact
StringHandling.DOWNCASE(row2.ShippingMethod)	ShippingMethod
StringHandling.DOWNCASE(row2.TotalAmount)	TotalAmount
row2.QuantitySold	QuantitySold
row2.DiscountAmount	DiscountAmount

Éditeur de schéma Éditeur d'expression

row2

Transformations

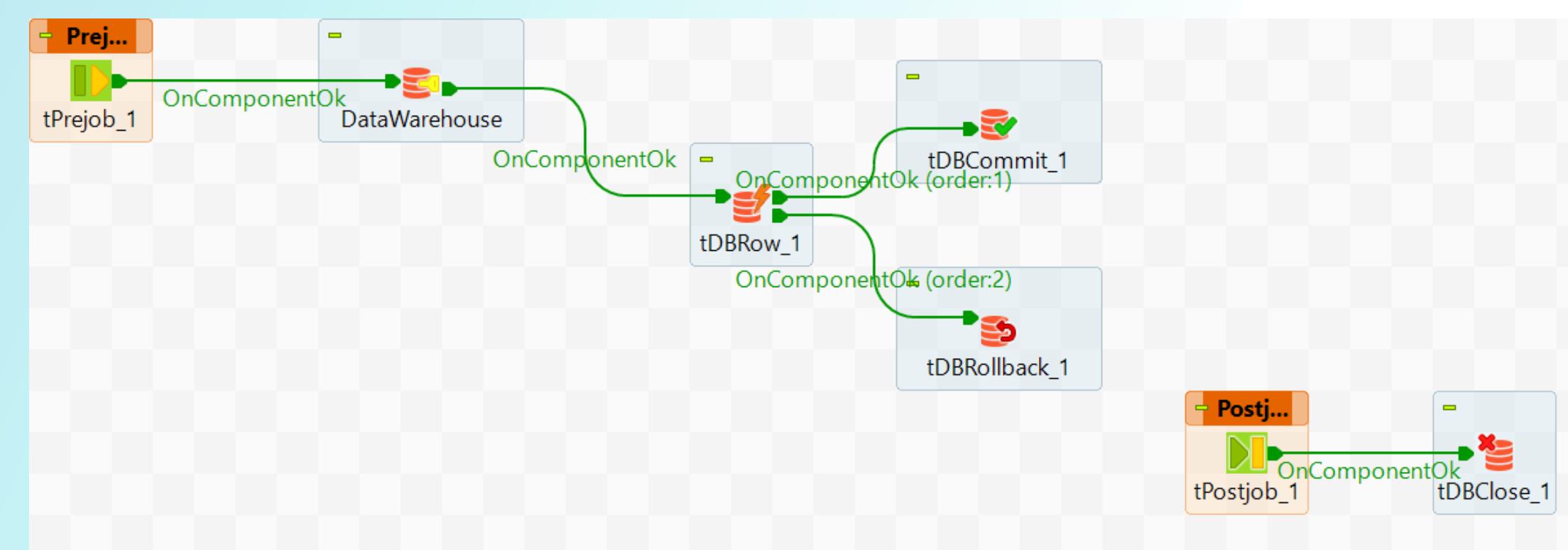
Colonne	C...	Type	<input checked="" type="checkbox"/> N.	Modèle de dat...	Longueur	Précision	Par d...	Comme...
Date	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		
ProductName	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		
ProductCategory	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		
ProductSubCategory	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		

Colonne	C...	Type	<input checked="" type="checkbox"/> N.	Modèle de dat...	Longueur	Précision	Par d...	Comme...
Date	<input type="checkbox"/>	Date	<input checked="" type="checkbox"/>	"yyyy-MM-dd"	10	0		
ProductName	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		
ProductCategory	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		
ProductSubCategory	<input type="checkbox"/>	String	<input checked="" type="checkbox"/>		255	0		

Appliquer Ok Annuler

Suppression des Clés Étrangères

Un job dédié, "DeleteForeignKeys", a été créé en utilisant TDBRow pour supprimer les clés étrangères de l'entrepôt de données.



Database Microsoft SQL Server Appliquer
Utiliser une connexion existante Liste des composants tDBConnection_1 - DataWarehouse*

Schéma Built-in Modifier le schéma

Nom de la table ""

Type de requête Built-in Guess Query

Requête

```
-- Drop foreign keys from SalesFact
ALTER TABLE SalesFact
DROP CONSTRAINT IF EXISTS fk_SalesFact_CustomerDimension;

ALTER TABLE SalesFact
DROP CONSTRAINT IF EXISTS fk_SalesFact_ShipperDimension;

ALTER TABLE SalesFact
DROP CONSTRAINT IF EXISTS fk_SalesFact_DateDimension;

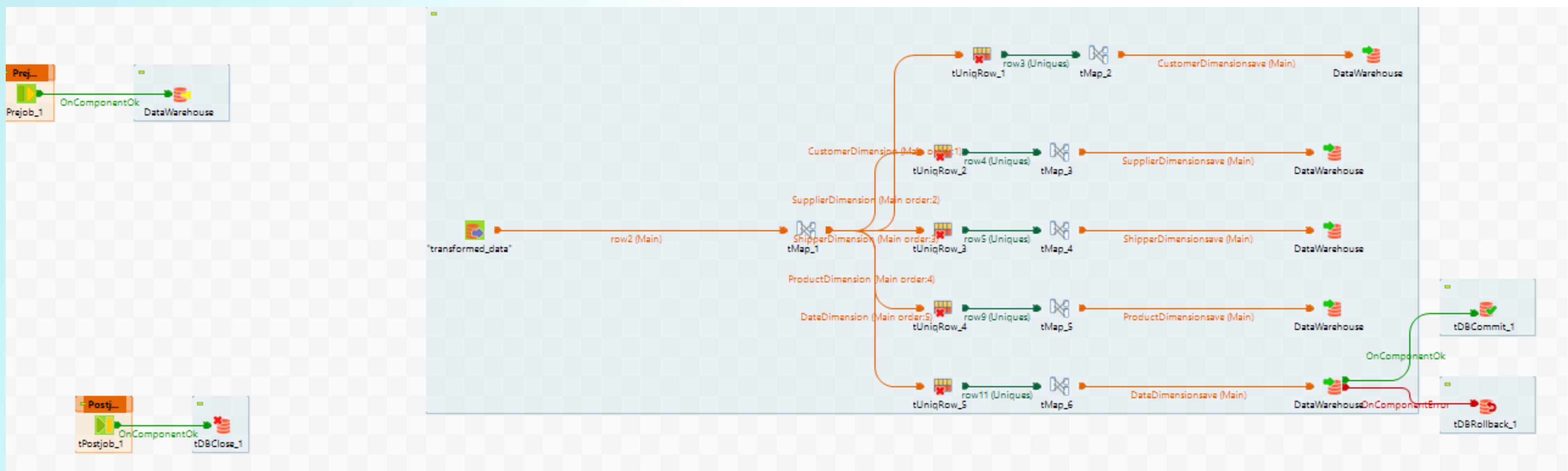
ALTER TABLE SalesFact
DROP CONSTRAINT IF EXISTS fk_SalesFact_ProductDimension;

-- Drop foreign keys from InventoryFact
ALTER TABLE InventoryFact
DROP CONSTRAINT IF EXISTS fk_InventoryFact_ProductDimension;

ALTER TABLE InventoryFact
DROP CONSTRAINT IF EXISTS fk_InventoryFact_DateDimension;
```

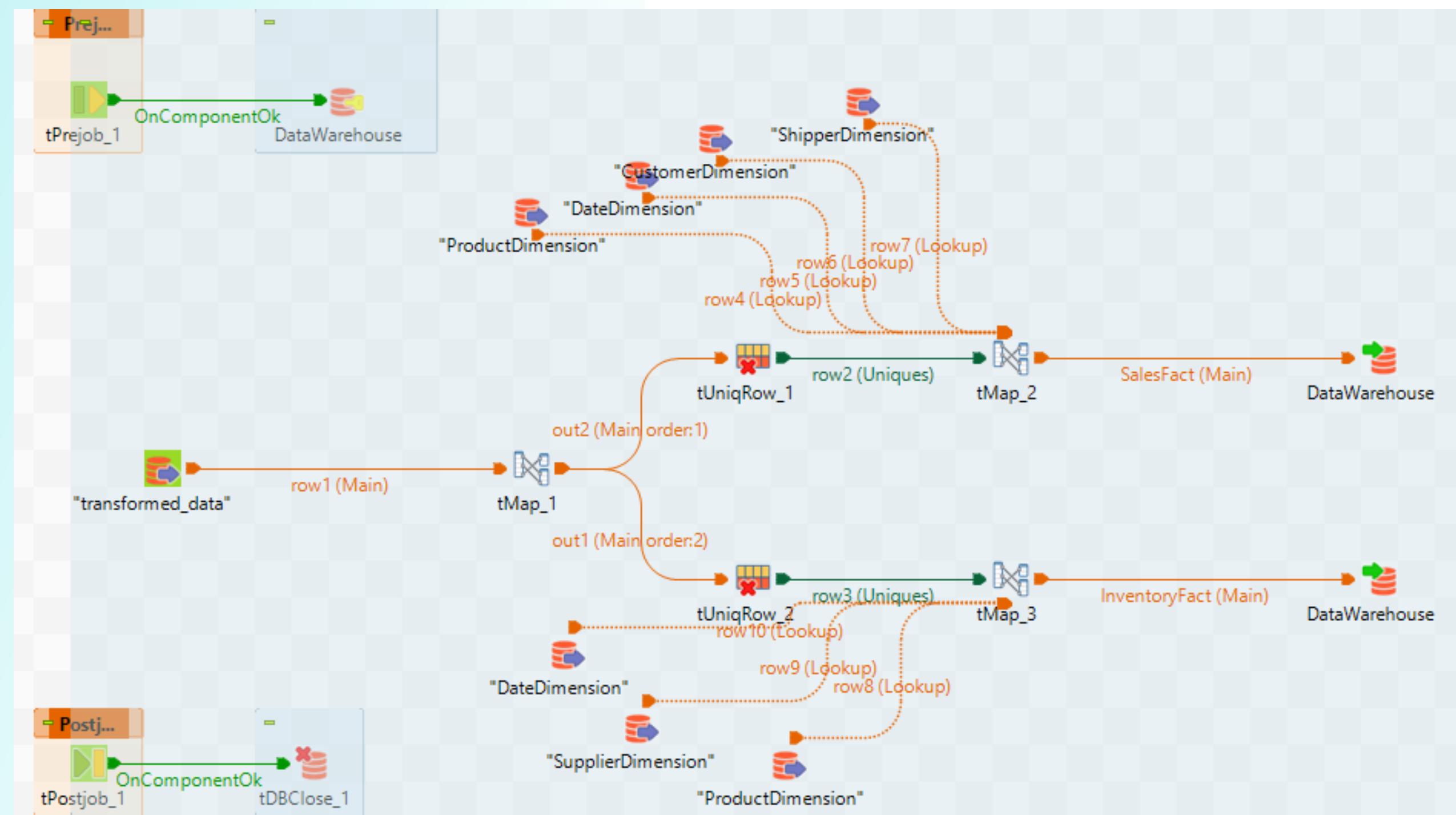
Tables de Dimensions

Le job "Dimensions" a divisé les "transformed_data", conservé les enregistrements uniques, généré des identifiants uniques et les a insérés dans les tables de dimension respectives de la base de données "Datawarehouse".



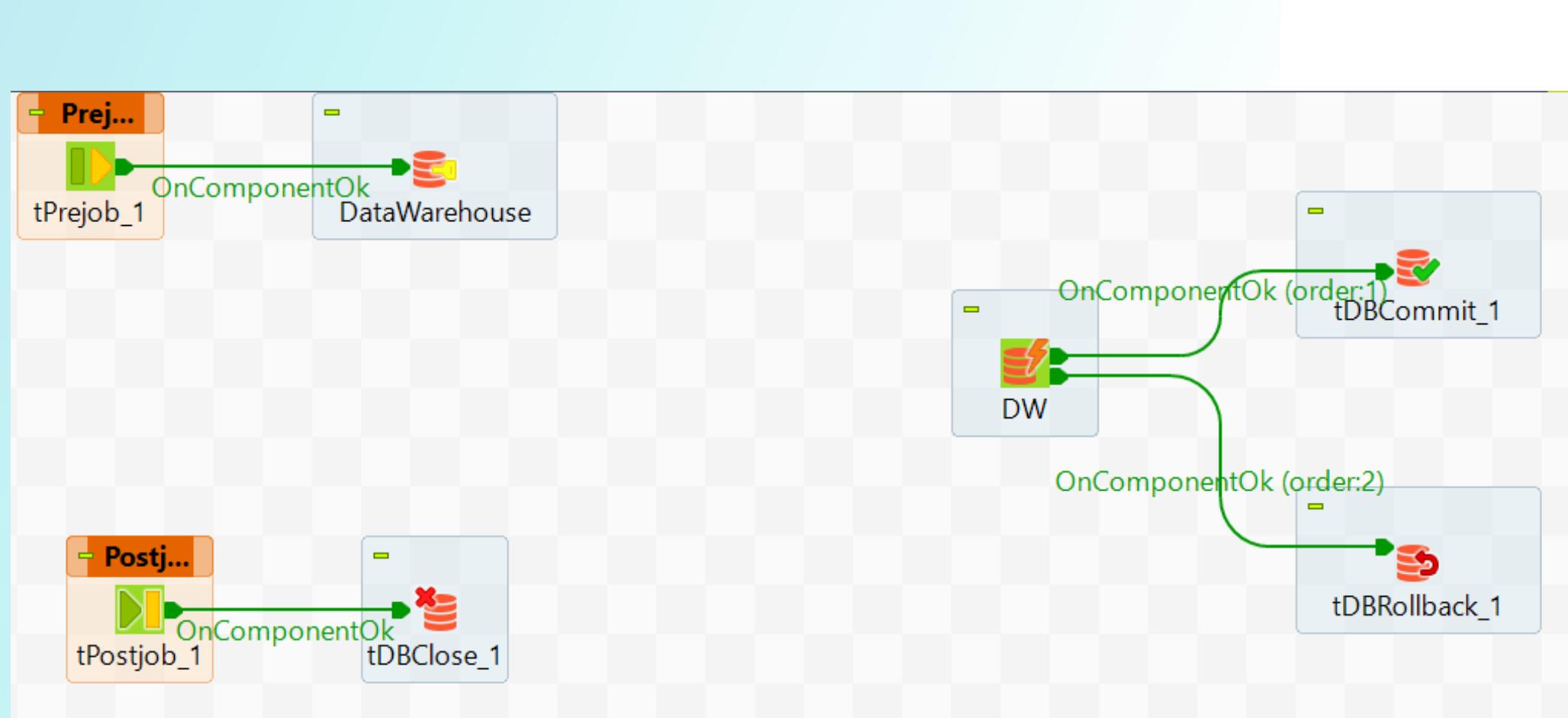
Tables de Faits

Un job "Facts" a été développé pour importer les tables de dimension et les insérer dans les tables de faits de l'entrepôt de données



Ajout des Clés Étrangères

À l'aide d'un autre job basé sur TDBRow appelé "AddForeignKeys", les clés étrangères ont été réintroduites dans l'entrepôt de données pour restaurer les relations entre différentes tables, garantissant que les données demeuraient liées pour l'analyse.



Requête

```
"  
ALTER TABLE SalesFact  
ADD CONSTRAINT fk_SalesFact_CustomerDimension  
FOREIGN KEY (CustomerID)  
REFERENCES CustomerDimension(CustomerID)  
ON DELETE CASCADE;  
  
ALTER TABLE SalesFact  
ADD CONSTRAINT fk_SalesFact_ShipperDimension  
FOREIGN KEY (ShipperID)  
REFERENCES ShipperDimension(ShipperID)  
ON DELETE CASCADE;  
  
ALTER TABLE SalesFact  
ADD CONSTRAINT fk_SalesFact_DateDimension  
FOREIGN KEY (DateID)  
REFERENCES DateDimension(DateID)  
ON DELETE CASCADE;  
  
ALTER TABLE SalesFact
```

DATA MARTS PHYSIQUES

-Un job nommé "InSalesDataMart" a été créé pour importer les tables de dimension nécessaires et les insérer dans le data mart spécifique aux données de vente.

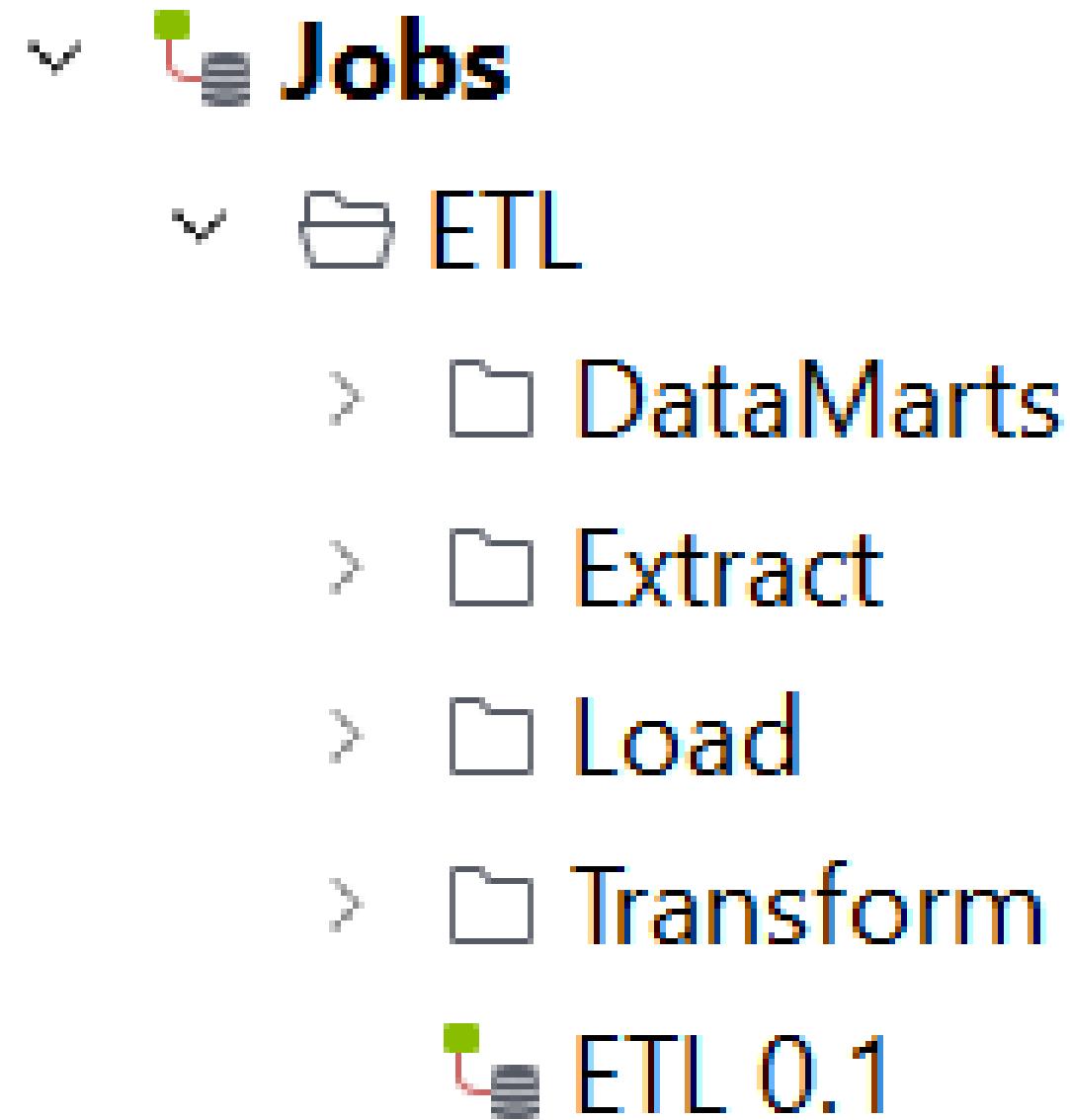
-De même, un job nommé "InInventoryDataMart" a été développé pour importer les tables de dimension requises et les insérer dans le data mart conçu pour les données d'inventaire.

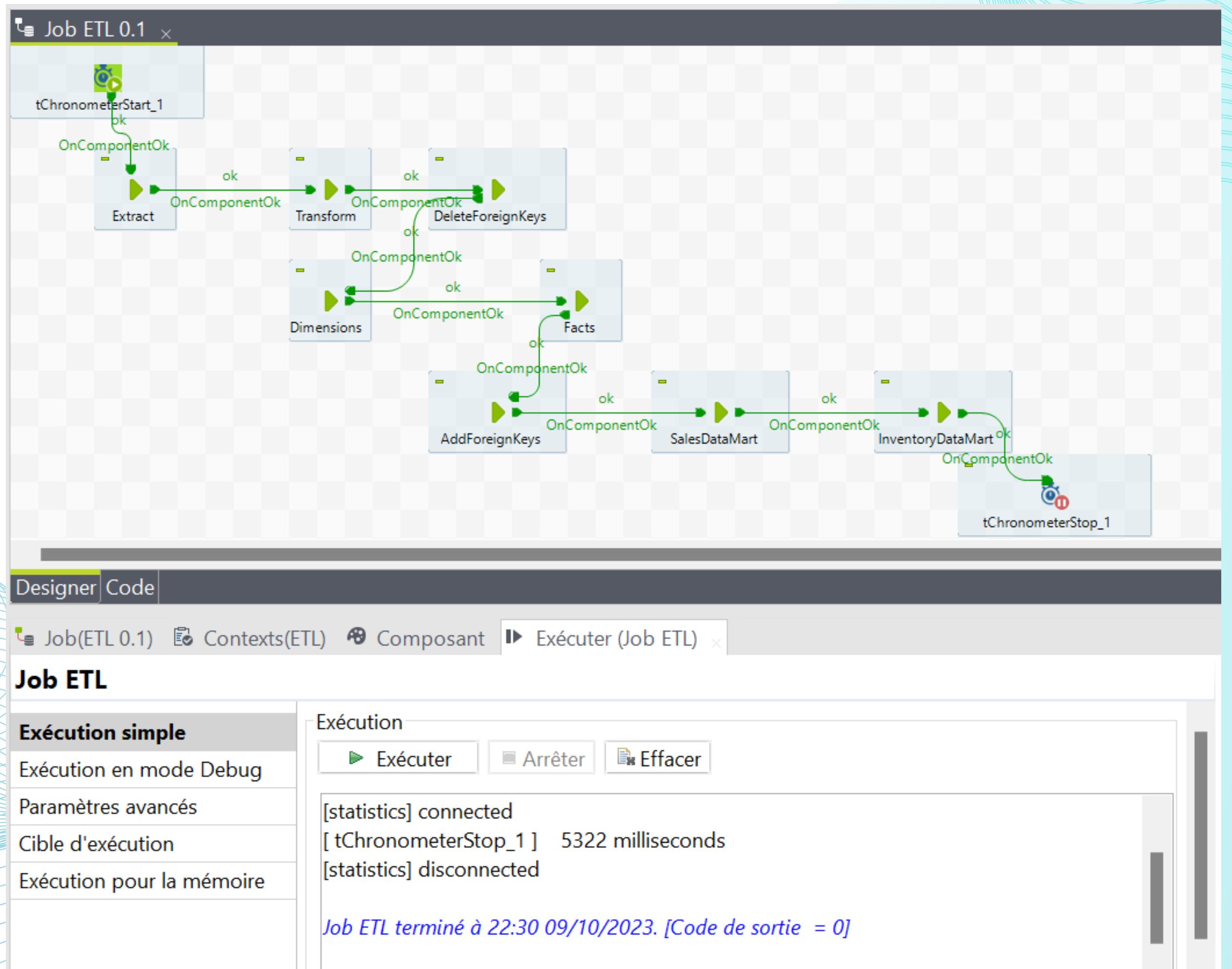




Job ETL Global

Tous ces jobs individuels ont été intégrés dans un job final appelé "ETL". Ce job principal orchestre l'ensemble du processus ETL, garantissant le déplacement efficace et ordonné des données de la source à la destination, tout en gérant diverses transformations, validations et insertions en cours de route. Le résultat est un entrepôt de données bien structuré, propre et organisé, avec des data marts associés, désormais prêts pour une analyse approfondie. Ce processus ETL complet constitue la base des capacités de gestion et d'analyse des données de ce projet.





ANALYTIQUE AVEC POWER BI

Concernant l'analytique du Data Mart des Ventes, nous avons réalisé plusieurs analyses essentielles, notamment :

- Analyse des Tendances des Ventes
- Analyse des Meilleurs Produits & Catégories
- Segmentation des Clients
- Analyse de l'Impact des Réductions
- Performance des Transporteurs

Dans le cadre de l'analyse du Data Mart de l'Inventaire, nos efforts ont été axés sur les aspects suivants :

- Surveillance du Niveau d'Inventaire
- Analyse de la Disponibilité des Stocks
- Évaluation de la Performance des Fournisseurs
- Prévision de la Demande de Produits

Sales Dashboard

E-COMMERCE SALES DASHBOARD

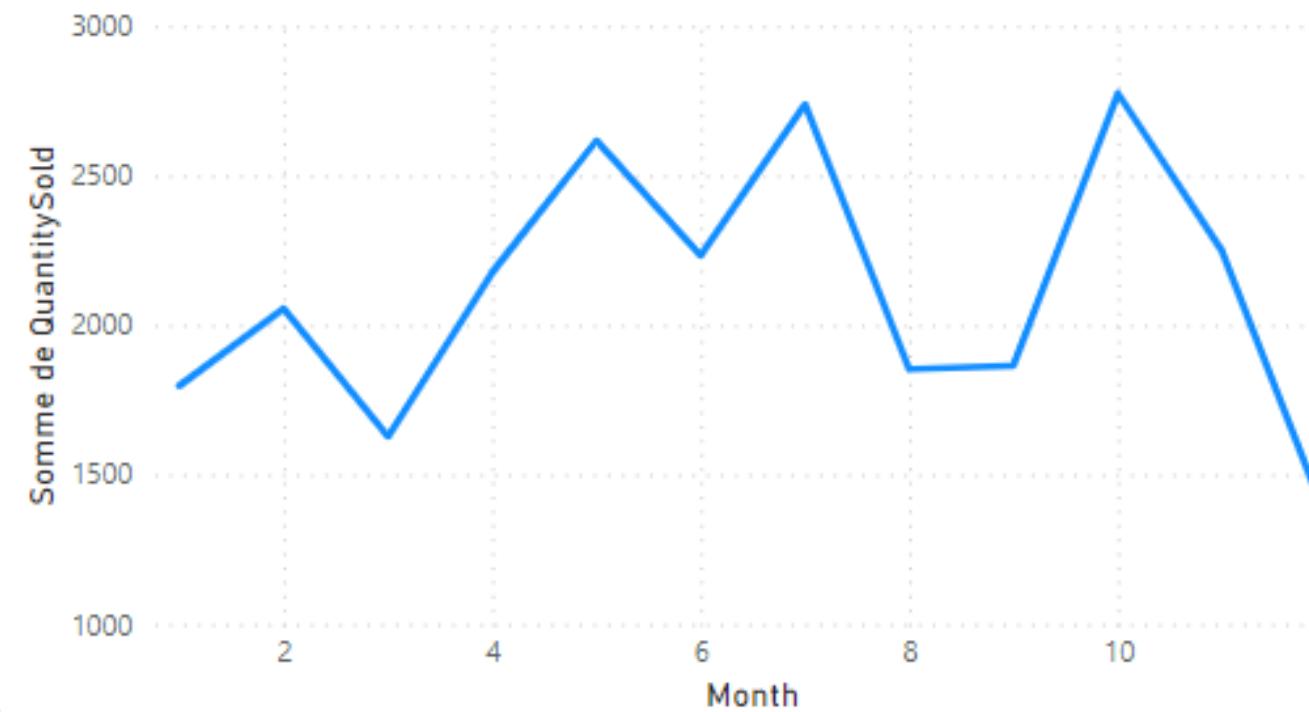
Total Sales

516

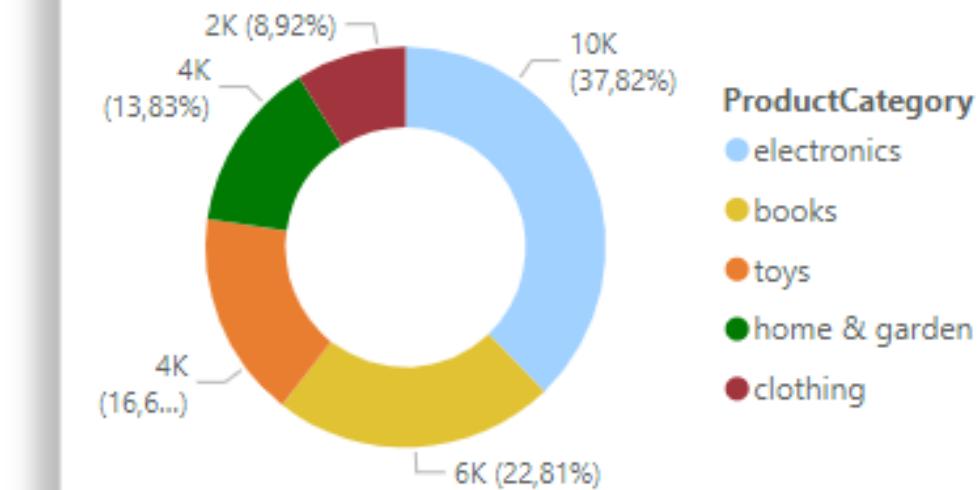
CustomerSegment ▾

- bronze
- gold
- silver

Sales Trend Analysis



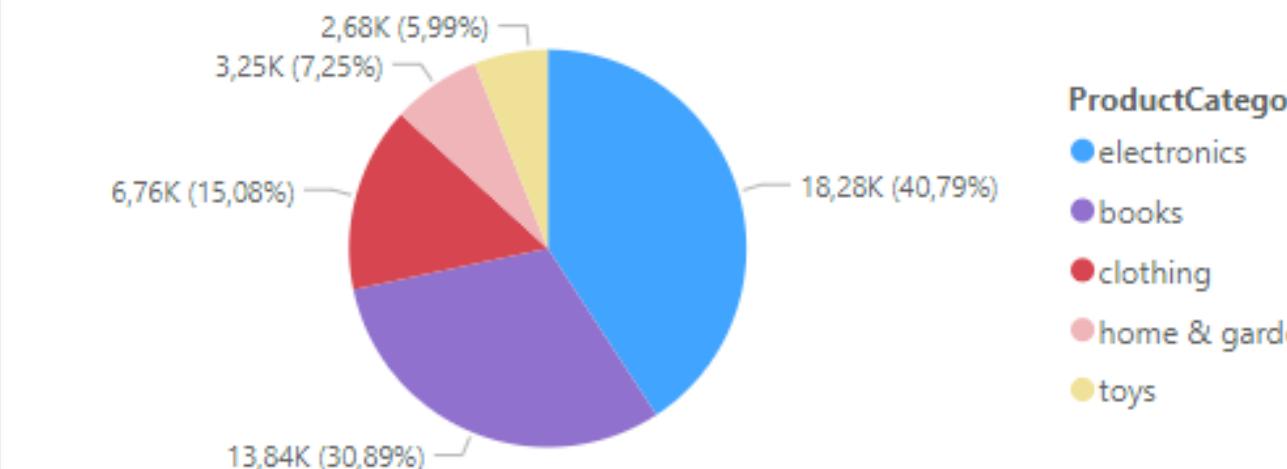
Analysis of the Best Products & Categories



ShippingMethod ▾

- air
- ground
- sea

Analysis of the Impact of Reductions



TotalAmount

14,46M

Inventory Dashboard

Ecommerce Inventory Dashboard

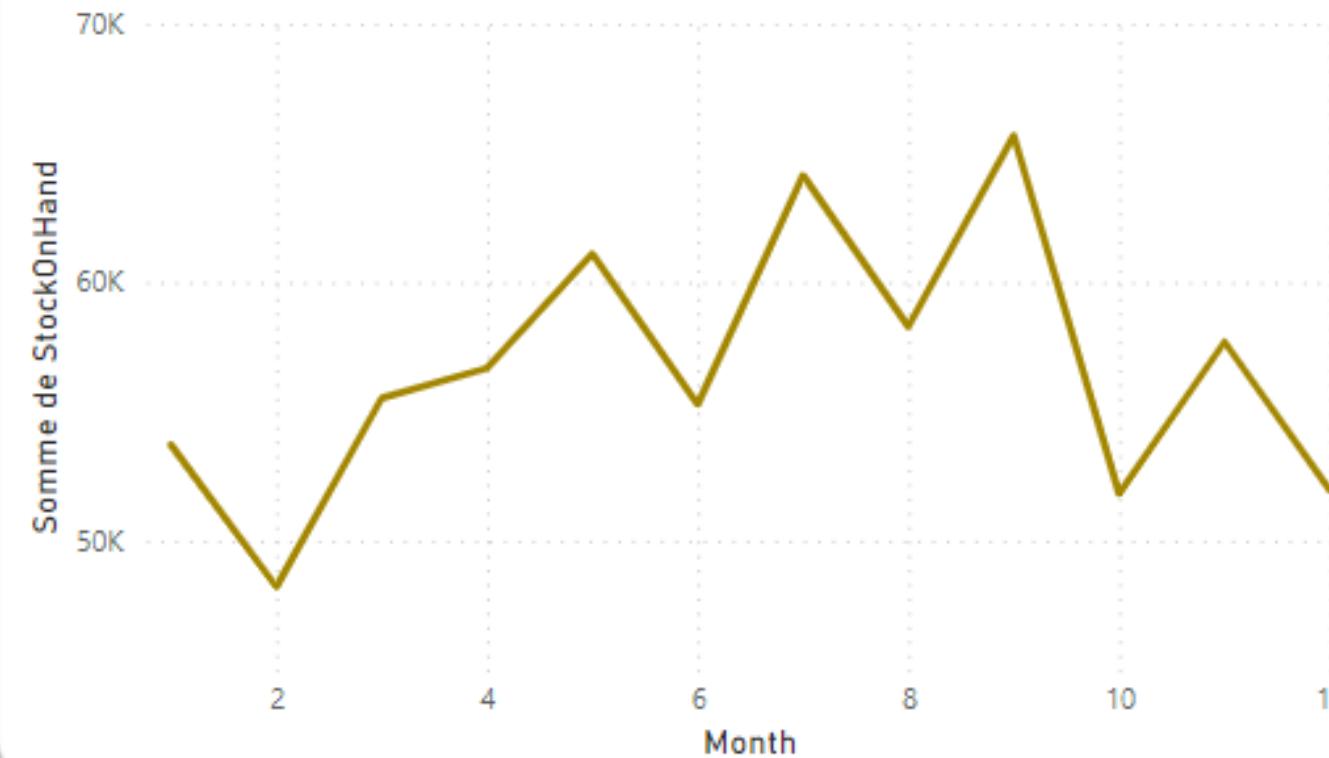
Total StockSold

679K

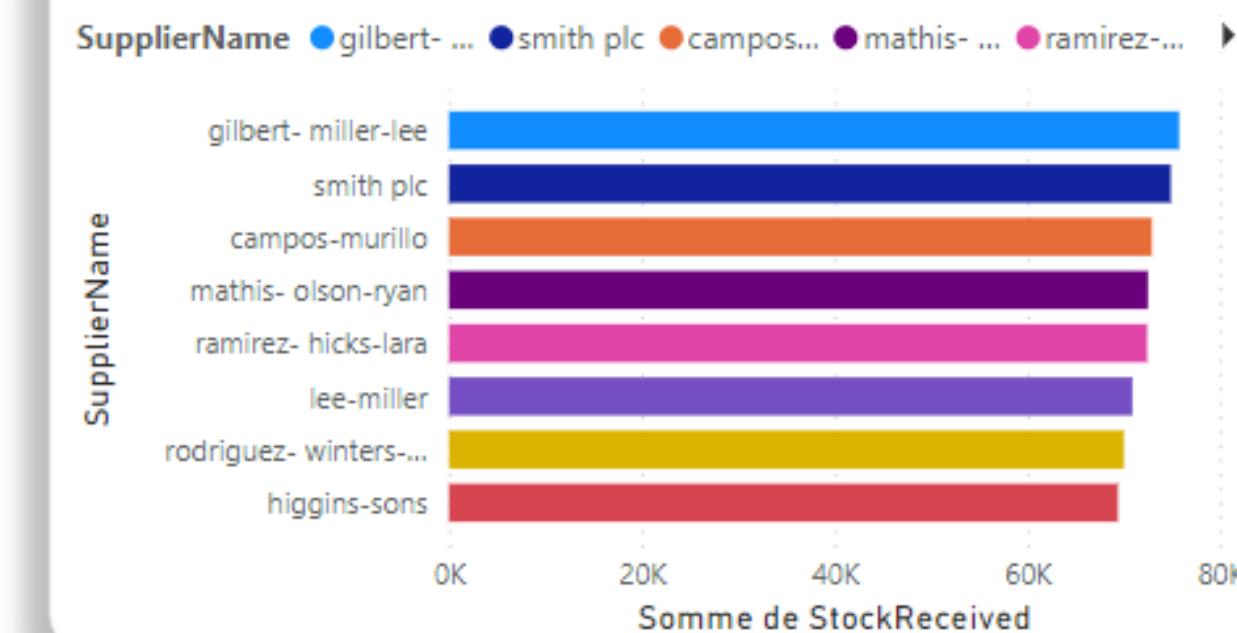
Stock Availability Analysis

Somme de StockOnHand	ProductCategory
166643	books
72277	clothing
230649	electronics
103403	home & garden
107099	toys
680071	

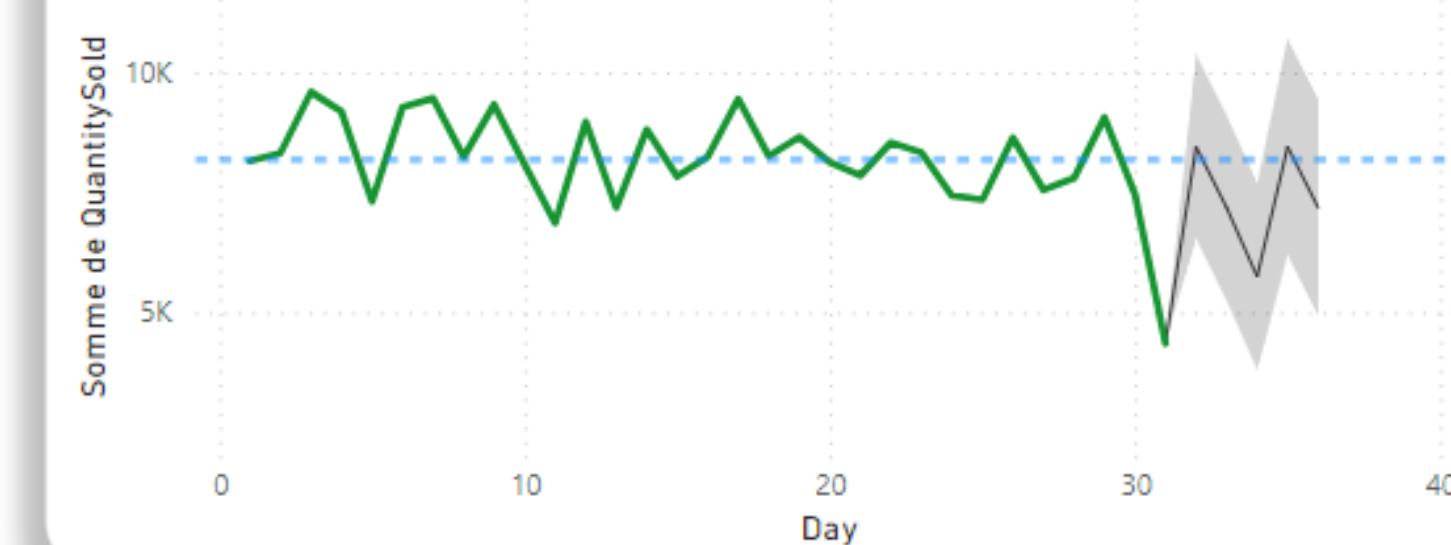
Inventory Level Monitoring



Supplier Performance Assessment



Quantity Sold Forecasting

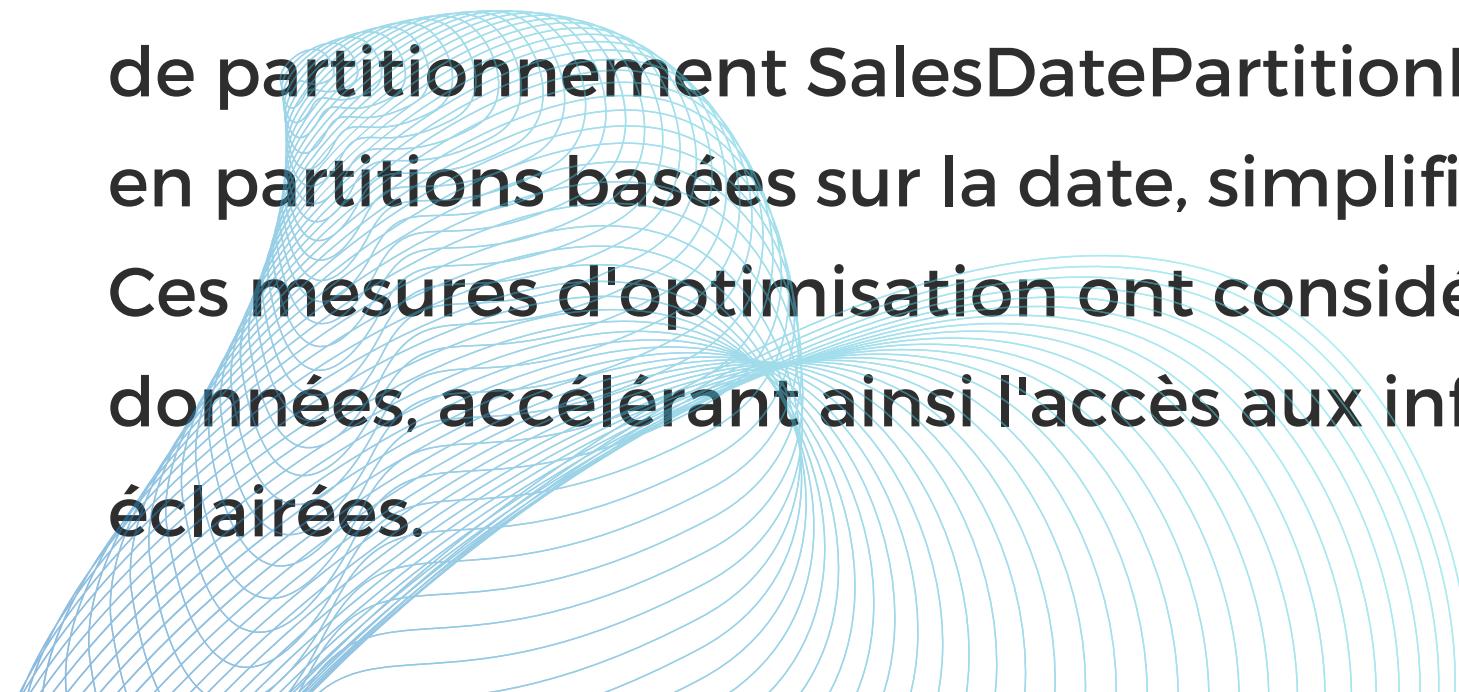


OPTIMISATION

Dans notre démarche d'optimisation, nous avons déployé des stratégies clés pour améliorer les performances de notre entrepôt de données. Nous avons utilisé des techniques d'indexation et de partitionnement pour atteindre cet objectif.

Tout d'abord, nous avons créé des index non cluster sur des colonnes fréquemment interrogées, notamment celles des tables de dimensions telles que DateDimension, ProductDimension et SupplierDimension. Ces index accélèrent les requêtes et optimisent la recherche de données.

Nous avons également adopté une stratégie de partitionnement basée sur la date en utilisant la fonction de partitionnement SalesDatePartitionFunction. Cette approche a divisé notre table de faits, SalesFact, en partitions basées sur la date, simplifiant la gestion et l'accès aux données historiques. Ces mesures d'optimisation ont considérablement renforcé les performances de notre entrepôt de données, accélérant ainsi l'accès aux informations, un élément essentiel pour des prises de décision éclairées.



VALIDATION LA LOGIQUE DE TRANSFORMATION

Dans la phase de validation de la logique de transformation, nous avons mené des requêtes et des tests pour confirmer la précision et la cohérence des données stockées dans notre entrepôt de données.

Tout d'abord, nous avons exécuté une requête pour extraire des données spécifiques de la table SalesFactPartitioned, confirmant ainsi que les données correspondent à la plage de dates attendue. Ensuite, une autre requête nous a permis d'effectuer des calculs agrégés sur les ventes de produits électroniques en 2022 et 2023, garantissant l'exactitude de nos calculs.

Enfin, un test de procédure stockée, nommé "CategoryAndNameTest," a été mis en place pour vérifier la qualité de nos données en s'assurant qu'aucun enregistrement ne comporte de noms de produits non valides ou de catégories de produits non valides.

Ces requêtes et tests renforcent la fiabilité de notre entrepôt de données, assurant ainsi des données de qualité pour nos futures analyses et rapports.



Optimisation-Part...-1US3GU3J\Youcode)* ✎ X

```
SELECT
    p.partition_number AS partition_number,
    f.name AS file_group,
    p.rows AS row_count
FROM sys.partitions p
JOIN sys.destination_data_spaces dds ON p.partition_number = dds.destination_id
JOIN sys.filegroups f ON dds.data_space_id = f.data_space_id
WHERE OBJECT_NAME(OBJECT_ID) = 'SalesFactPartitioned'
order by partition_number;
```

79 %

Results Messages

	partition_number	file_group	row_count
1	1	PRIMARY	0
2	2	FG_sales_2021	646
3	3	FG_sales_2022	2436
4	4	FG_sales_2023	1918

MAX: 1

MIN: 1

AVG: 1

SUM: 1

COUNT: 1

DISTINCT: 1

Optimisation-Part...-1US3GU3J\Youcode)* ✎ X

```
-- Select aggregated data from the SalesFactPartitioned table
SELECT
    dd.Year,
    pd.ProductCategory,
    SUM(sf.QuantitySold) AS TotalQuantitySold,
    SUM(sf.TotalAmount) AS TotalSalesAmount
FROM
    SalesFactPartitioned sf
JOIN
    DateDimension dd ON sf.DateID = dd.DateID
JOIN
    ProductDimension pd ON sf.ProductID = pd.ProductID
WHERE
    dd.Year IN (2022, 2023)
    AND pd.ProductCategory = 'Electronics'
GROUP BY
    dd.Year, pd.ProductCategory;
```

79 %

Results Messages

	Year	ProductCategory	TotalQuantitySold	TotalSalesAmount
1	2022	electronics	43448	26810269.99
2	2023	electronics	33141	21067208.59

MAX: 2022 MIN: 2022 AVG: 2022 SUM: 2022 COUNT: 1 DISTINCT: 1

SQLQuery1.sql - L..1US3GU3J\Youcode)*

Test List Manager

UnitTesting.sql -...P-1US3GU3J\Youcode)

```
-- Comments here are associated with the test.
-- For test case examples, see: http://tsqlt.org/user-guide/tsqlt-tutorial/
ALTER PROCEDURE Product.[test CategoryAndNameTest]
AS
BEGIN

    -- Act: Query the table for invalid product names and categories
    DECLARE @invalidProductNames INT;
    DECLARE @invalidProductCategories INT;

    SELECT @invalidProductNames = COUNT(*)
    FROM [WarehouseEcommerce].[dbo].[ProductDimension] pd
    WHERE [ProductName] = 'NonExistentProduct';

    SELECT @invalidProductCategories = COUNT(*)
    FROM [WarehouseEcommerce].[dbo].[ProductDimension] pd
    WHERE [ProductCategory] = 'InvalidCategory';

    -- Assert: Verify that there are no occurrences of invalid product names or categories
    EXEC tSQLt.AssertEquals 0, @invalidProductNames;
    EXEC tSQLt.AssertEquals 0, @invalidProductCategories;

END;
```

78 %

Connected. (1/1)

LAPTOP-1US3GU3J\SQLEXPRESS ... | LAPTOP-1US3GU3J\Youcod... | WarehouseEcommerce | 00:00:00 | 0 rows

Test Results

Status	Test Name	Class Name	Error Message	Execution Time
✓ Succeeded	CategoryAndNameTest	Product		0 ms

Failed: 0; Passed: 1; Total: 1; Checked: 0; Connection: 'LAPTOP-1US3GU3J\SQLEXPRESS' Status: Connected Database: WarehouseEcommerce

AUTORISATION

Dans la phase d'autorisation, nous avons établi des règles de sécurité pour garantir que seuls les utilisateurs autorisés ont accès aux données de l'entrepôt. Nous avons créé des logins et des utilisateurs, associé ces utilisateurs à des rôles pertinents, et attribué des autorisations spécifiques. Par exemple, le Data Engineer a des autorisations pour gérer les données de vente, tandis que le Data Analyst a des autorisations limitées à l'analyse des données. Ces mesures de sécurité garantissent que seules les personnes appropriées ont accès aux données de l'entrepôt, tout en maintenant l'intégrité des données.

-- Use the target database

USE WarehouseEcommerce;

-- Create server-level logins with passwords

CREATE LOGIN DataEngineerLogin WITH PASSWORD = 'DataEngineer2004';

CREATE LOGIN DataAnalystLogin WITH PASSWORD = 'DataAnalyst2004';

-- Create database users

CREATE USER DataEngineerUser FOR LOGIN DataEngineerLogin;

CREATE USER DataAnalystUser FOR LOGIN DataAnalystLogin;

-- Create database roles for Data Engineer and Data Analyst

CREATE ROLE DataEngineerRole;

CREATE ROLE DataAnalystRole;

-- Add users to their respective roles

ALTER ROLE DataEngineerRole ADD MEMBER DataEngineerUser;

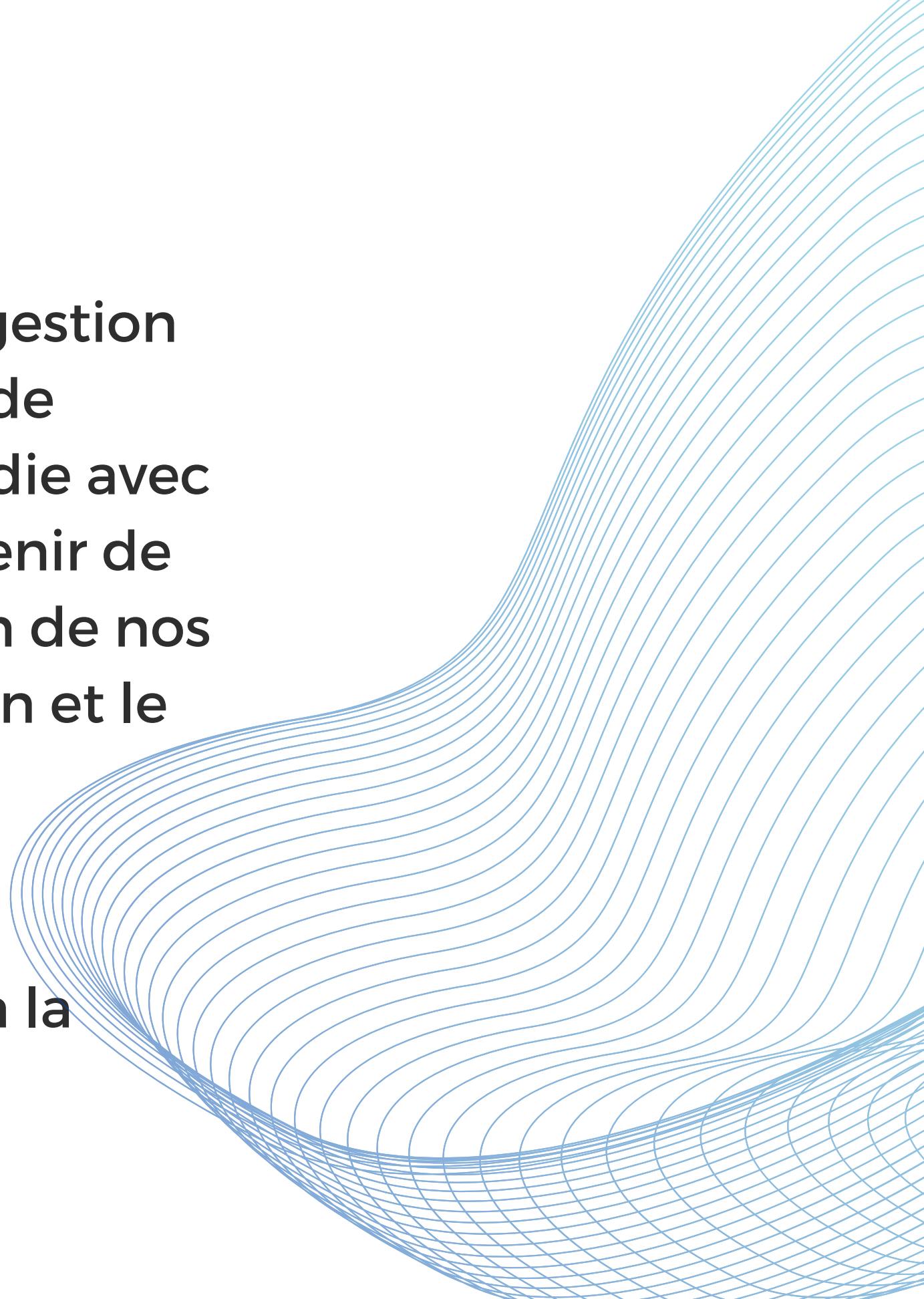
ALTER ROLE DataAnalystRole ADD MEMBER DataAnalystUser;

```
-- Grant permissions to DataEngineerRole  
GRANT SELECT ON SalesFact TO DataEngineerRole;  
GRANT INSERT ON SalesFact TO DataEngineerRole;  
GRANT UPDATE ON SalesFact TO DataEngineerRole;  
GRANT DELETE ON SalesFact TO DataEngineerRole;  
  
GRANT SELECT ON SupplierDimension TO DataEngineerRole;  
GRANT SELECT ON ProductDimension TO DataEngineerRole;  
GRANT SELECT ON ShipperDimension TO DataEngineerRole;  
GRANT SELECT ON DateDimension TO DataEngineerRole;
```

```
-- Grant permissions to DataAnalystRole  
GRANT SELECT ON SalesFact TO DataAnalystRole;  
GRANT SELECT ON SupplierDimension TO DataAnalystRole;  
GRANT SELECT ON ProductDimension TO DataAnalystRole;  
GRANT SELECT ON ShipperDimension TO DataAnalystRole;  
GRANT SELECT ON DateDimension TO DataAnalystRole;
```

CONCLUSION

En conclusion, ce projet représente une réalisation remarquable dans notre parcours visant à améliorer la gestion des données. Notre entrepôt de données, nos schémas de constellation, nos data marts et notre analyse approfondie avec Power BI forment une combinaison puissante pour obtenir de nouvelles perspectives sur nos opérations. L'optimisation de nos data marts grâce à des techniques telles que l'indexation et le partitionnement souligne notre engagement envers la performance et l'efficacité. De plus, notre focalisation sur la sécurité des données et la conformité aux réglementations du RGPD renforce notre dévouement à la protection des informations sensibles.



Merci
pour votre Attention



(212) 07 72 10 47 62



faissalmouflla@outlook.fr



Faissal Mouflla

