

MONDAY, OCTOBER 9, 2023

OPTIMISATION DE LA GESTION DES DONNÉES DANS UNE PLATEFORME

E-COMMERCE

RAPPORT

AYMANE SABRI

DEVELOPER DE DONNÉES

Objectif du projet :

L'objectif principal de ce projet est de développer une compréhension approfondie des concepts fondamentaux de l'[ETL](#) en utilisant les services [Microsoft Azure](#). Nous nous sommes engagés à mettre en œuvre un scénario d'extraction, de transformation et de chargement de données.

I. Introduction :

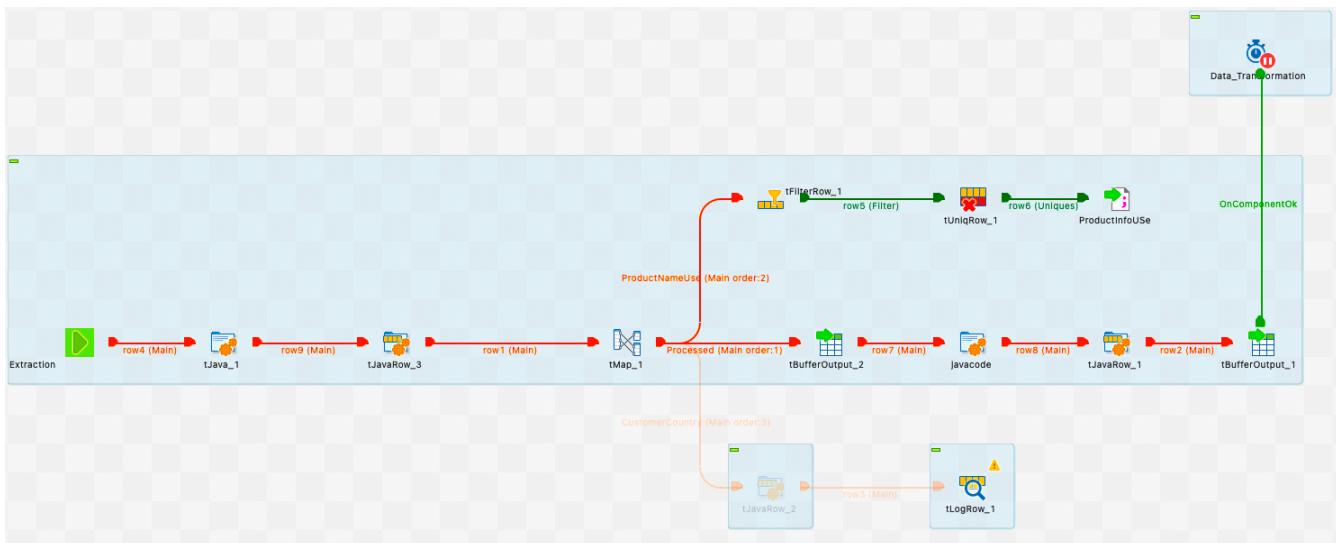
1. Contexte du Projet :

Le secteur du commerce électronique est en constante évolution, et la gestion efficace des données est essentielle pour rester compétitif. Dans ce contexte, ce projet a été initié pour optimiser la gestion des données d'une plateforme e-commerce.

II. Mise en Place de l'ETL avec Talend :

1. Extraction des Données :

1.1. Extraction des Données CSV et JSON :



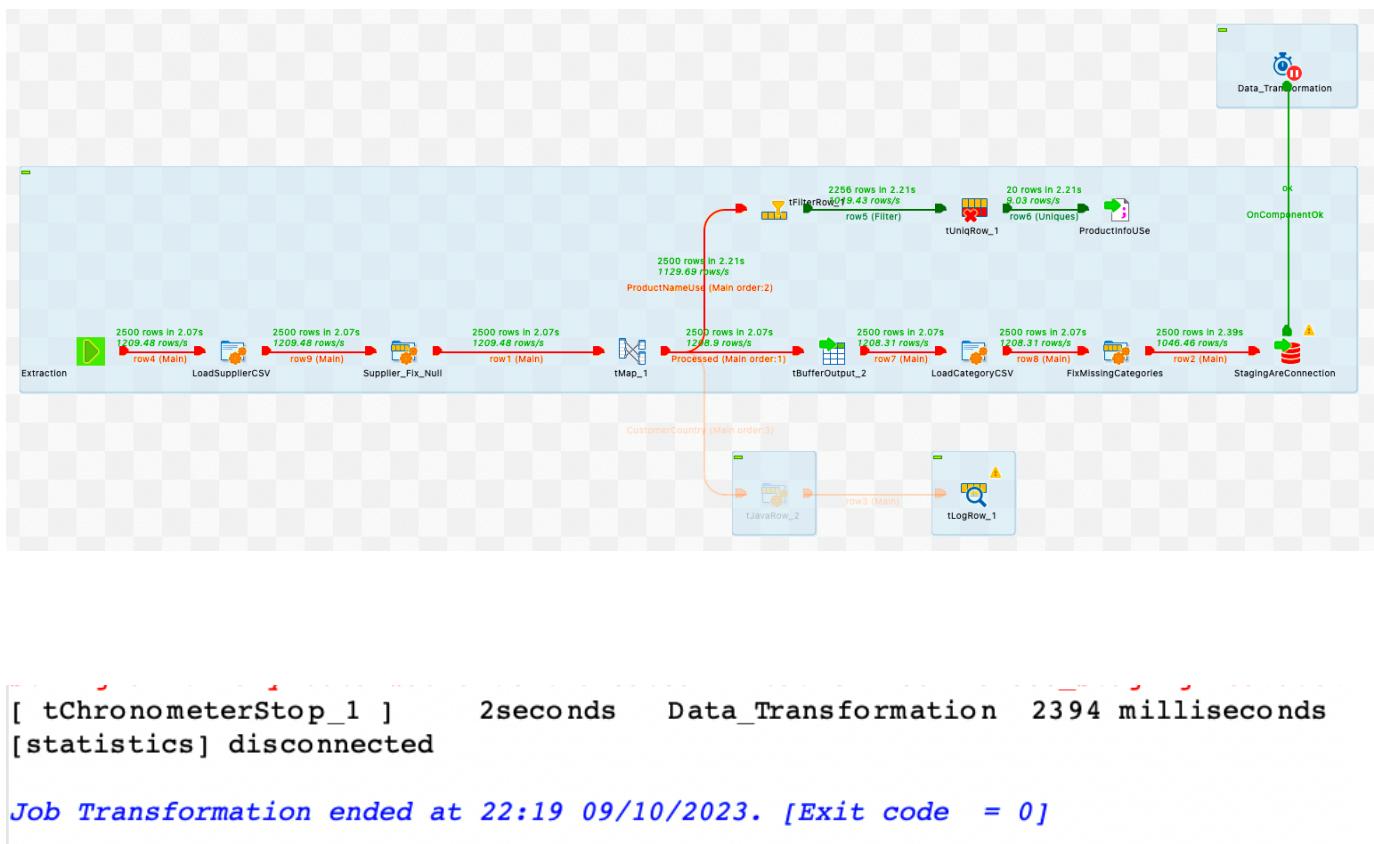
```
Starting job Extraction at 22:15 09/10/2023.  
[statistics] connecting to socket on port 3928  
[statistics] connected  
[ tChronometerStop_1 ]      1seconds    DataExtraction  1910 milliseconds  
[statistics] disconnected  
  
Job Extraction ended at 22:15 09/10/2023. [Exit code = 0]
```

Après avoir établi les connexions, nous avons procédé à l'extraction des données à partir des fichiers CSV et JSON. Cette étape a impliqué l'utilisation des capacités d'extraction de Talend pour extraire ces données brutes et les préparer pour les préparer à l'insertion dans la base de données (staging area).

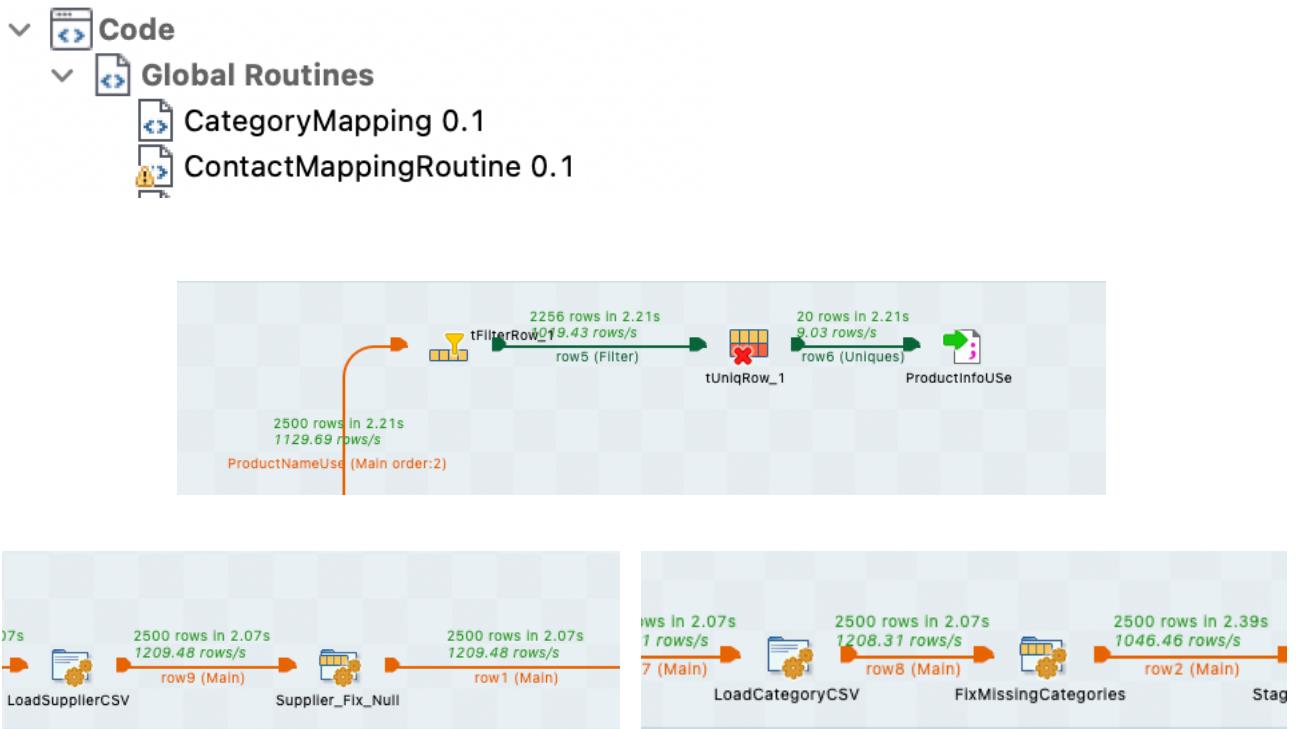
2. Transformation des Données :

2.1. Nettoyage des Données :

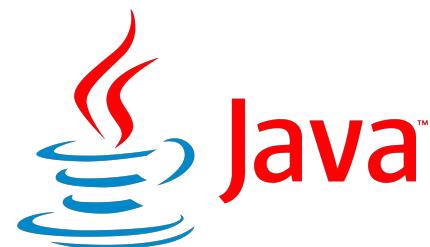
Une fois les données extraites, elles ont été soumises à un processus de nettoyage approfondi. Cela incluait l'**identification** et la gestion des valeurs **manquantes**, la suppression des doublons et la résolution des incohérences. Le nettoyage des données était essentiel pour garantir la qualité et la cohérence des données avant leur chargement dans la base de données.



- Supplier Contact , Product Category



J'ai choisi d'utiliser des routines et de créer du code Java pour effectuer les transformations nécessaires, en particulier pour remplacer les valeurs nulles par des valeurs appropriées. Cette approche repose sur la correspondance des autres valeurs d'un objet afin de distinguer les données manquantes et de les remplacer de manière adéquate.



2.2. Conversion des Types de Données , Application de la Logique Métier :

Expression	Column
row1.Date.split("-")[0].length() == 4 ? // Check if the year part has 4 characters ... (row1.Date.split(''	Date
row1.ProductName.replaceAll("(?i)NonExistentProduct", "UNKNOWN").toUpperCase();	ProductName
row1.ProductCategory.toUpperCase()	ProductCategory
row1.ProductSubCategory.toUpperCase();	ProductSubCategory
(float)(row1.TotalAmount / row1.QuantitySold)	ProductPrice
row1.CustomerName.toUpperCase()	CustomerName
!row1.CustomerEmail.isEmpty() ? row1.CustomerEmail : "Unknown"	CustomerEmail
row1.CustomerAddress	CustomerAddress
row1.CustomerPhone	CustomerPhone
row1.CustomerSegment.toUpperCase()	CustomerSegment
row1.SupplierName.toUpperCase()	SupplierName
row1.SupplierLocation.toUpperCase()	SupplierLocation
row1.SupplierContact	SupplierContact
row1.ShipperName.toUpperCase()	ShipperName
row1.ShippingMethod.toUpperCase()	ShippingMethod
row1.QuantitySold	QuantitySold
row1.TotalAmount	TotalAmount
row1.DiscountAmount	DiscountAmount
row1.NetAmount	NetAmount
row1.StockReceived	StockReceived
row1.StockSold	StockSold
row1.StockOnHand	StockOnHand

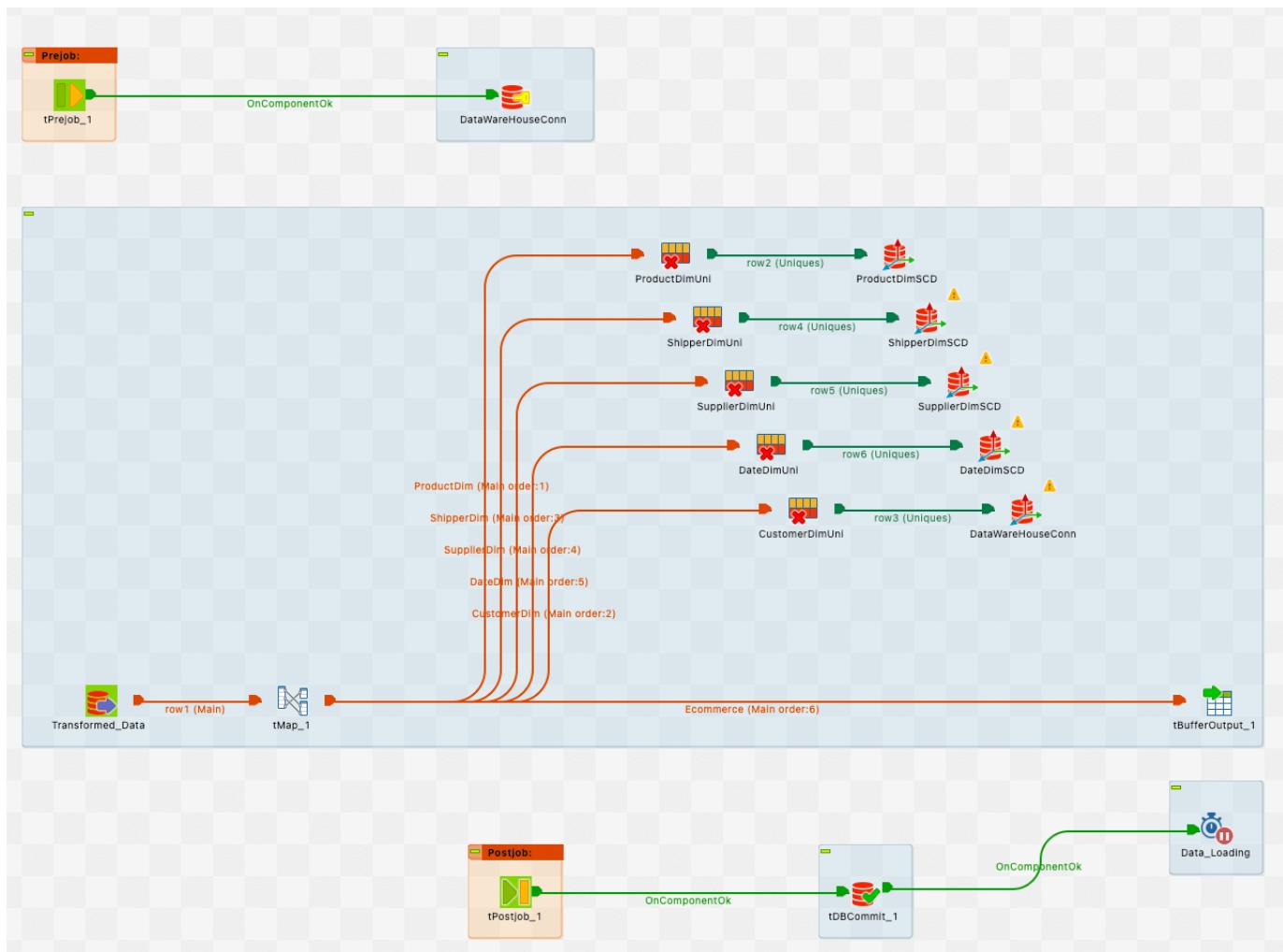
L'objectif était de préparer les données de manière à ce qu'elles fournissent des informations exploitables. Pour y parvenir, nous avons effectué des transformations essentielles, notamment :

- La correction des valeurs des prix des produits. Cette correction a été effectuée en divisant le "[Total Amount](#)" par la "[Quantity Sold](#)".
- Formattage de la column date vers le format appropriés .

IV. Schéma de Fast Constellation :

1. Crédation des Tables de Dimension :

- DateDimension
- ProductDimension
- CustomerDimension
- SupplierDimension
- ShipperDimension



```
Starting job DimensionsTablesCreation at 22:36 09/10/2023.  
[statistics] connecting to socket on port 4064  
[statistics] connected  
[ tChronometerStop_1 ]      0seconds    DimensionnTableCreation  517 milliseconds  
[statistics] disconnected  
  
Job DimensionsTablesCreation ended at 22:36 09/10/2023. [Exit code = 0]
```

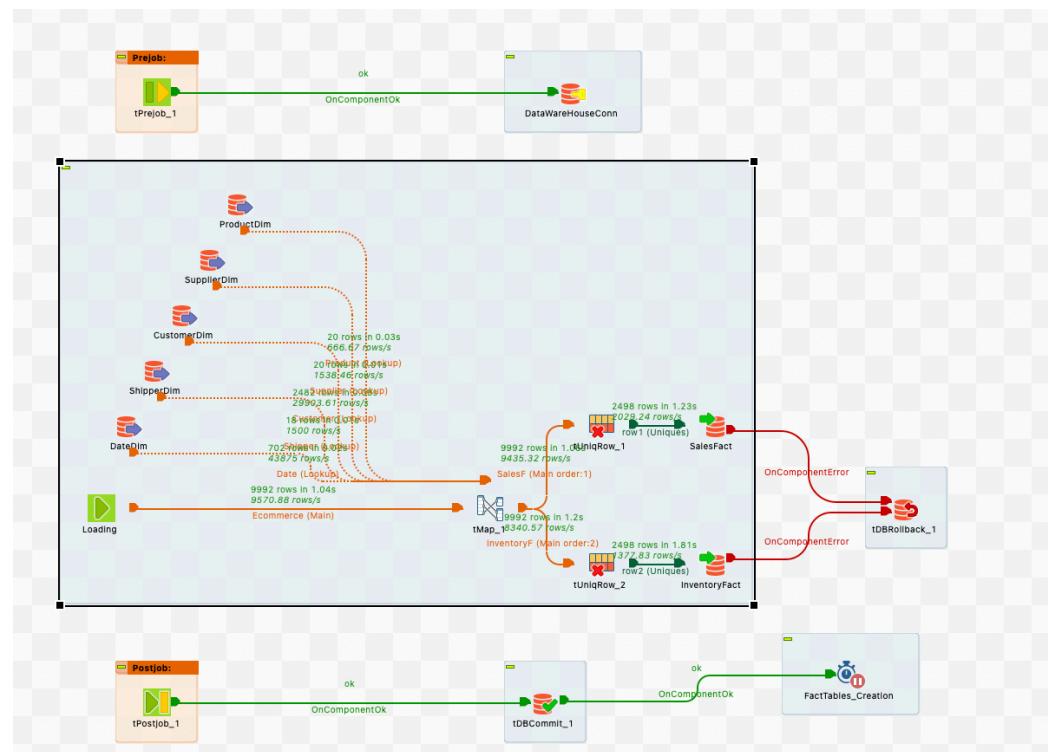
2. Crédation des Tables de Faits :

2.1. SalesFact :

La table SalesFact était au cœur de notre schéma de constellation, enregistrant les données liées aux transactions de vente. Elle contenait des informations telles que les clés de dimension, les quantités vendues, les montants de vente, les réductions appliquées et les dates de transaction. Cette table était essentielle pour l'analyse des ventes et des performances.

2.2. InventoryFact :

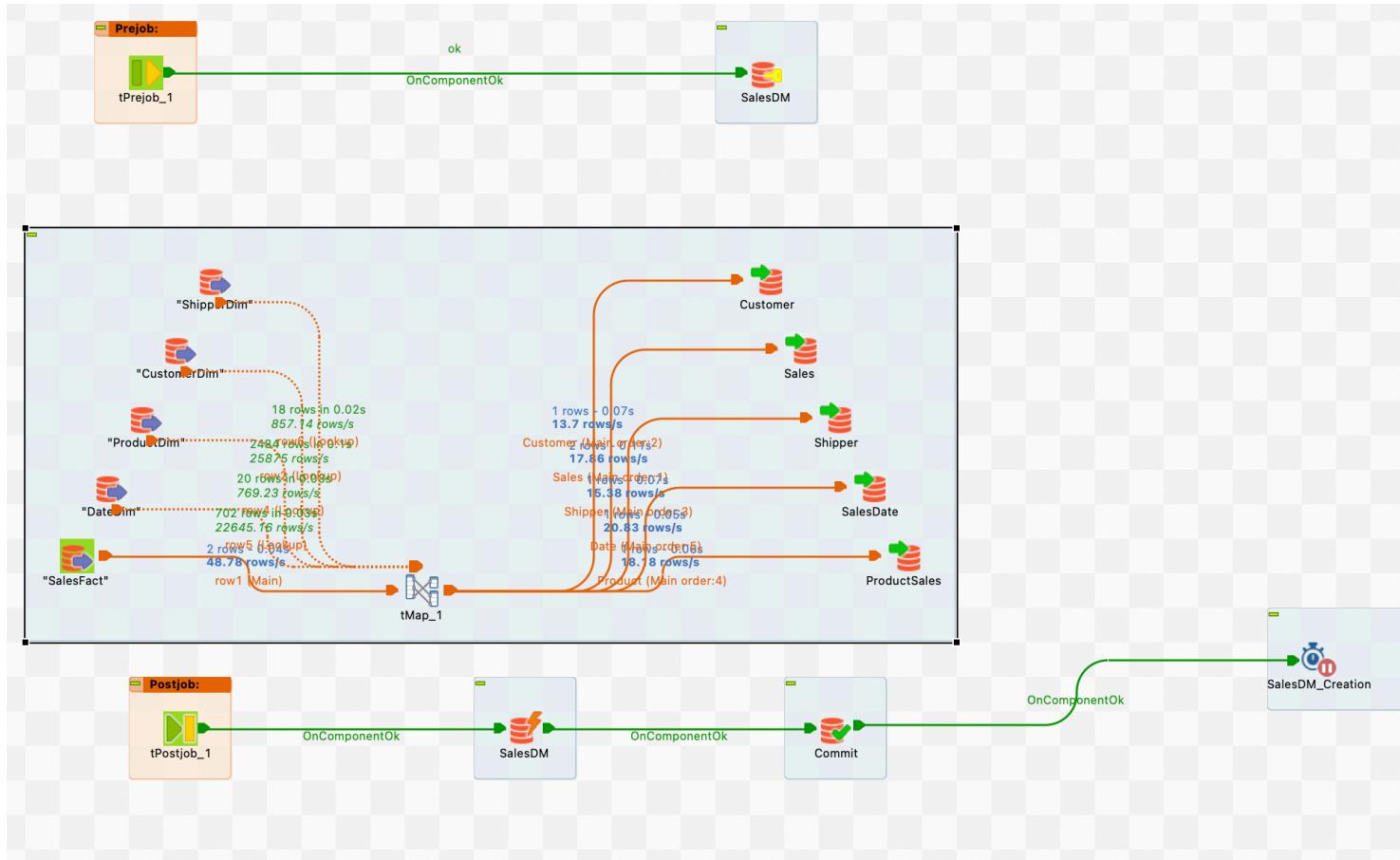
La table InventoryFact a été créée pour suivre les données relatives à l'inventaire des produits. Cela incluait les niveaux de stock, les dates d'inventaire et les mouvements de stock. Cette table a été utilisée pour surveiller la disponibilité des stocks et évaluer la performance des fournisseurs.



III. Data Marts Physiques:

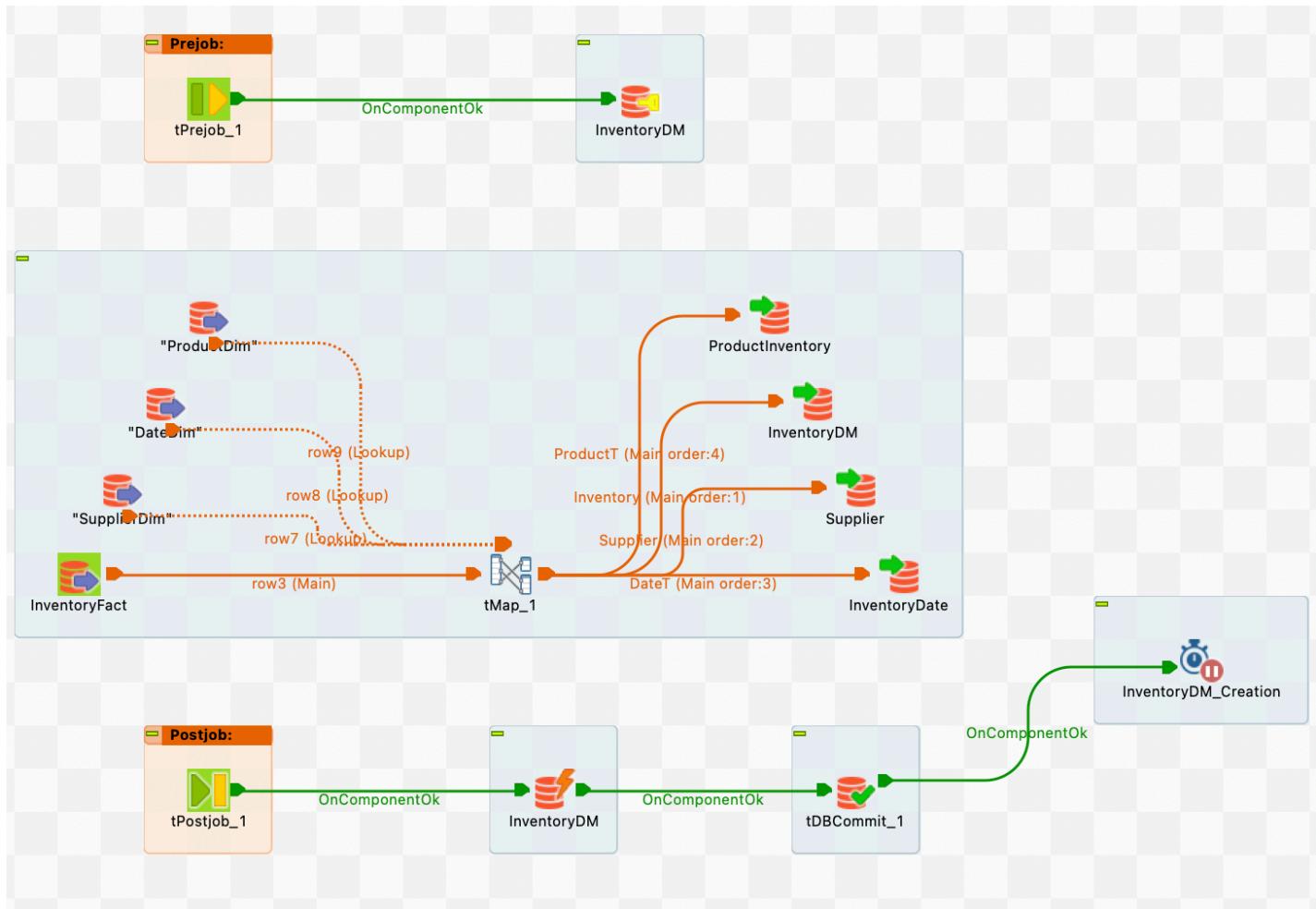
1. Data Mart des Ventes :

1.1. Tables Liées aux Transactions de Vente



2. Data Mart de l'Inventaire :

2.1. Tables Liées à l'Inventaire des Produits



V. Analyse avec Power BI:

1. Analytique du Data Mart des Ventes :

- 1.1. Analyse des Tendances des Ventes
- 1.2. Analyse des Meilleurs Produits & Catégories
- 1.3. Analyse de Segmentation des Clients
- 1.4. Analyse de l'Impact des Réductions
- 1.5. Performance des Transporteurs



2. Analytique du Data Mart des Ventes :

- 2.1. Surveillance du Niveau d'Inventaire
- 2.2. Analyse de la Disponibilité des Stocks
- 2.3. Évaluation de la Performance des Fournisseurs
- 2.4. Prévision de la Demande de Produits



V. Optimisation de la Base de Données SQL Server:

1. Indexation & Partitionnement :

```
-- Create a temporary table to hold the distinct years
CREATE TABLE #DistinctYears (YearValue INT);

-- Insert distinct years from your DateDim table
INSERT INTO #DistinctYears (YearValue)
SELECT DISTINCT [Year] FROM [dbo].[DateDim];

-- Create a partition function dynamically
DECLARE @PartitionFunctionSQL NVARCHAR(MAX) = 'CREATE PARTITION FUNCTION YearPartitionFunction (INT) AS RANGE LEFT FOR VALUES (';
DECLARE @Year INT;
DECLARE @FirstYear BIT = 1;
DECLARE @FilegroupSQL NVARCHAR(MAX);

-- Loop through distinct years to generate partition function SQL
WHILE EXISTS (SELECT * FROM #DistinctYears)
BEGIN
    SELECT TOP 1 @Year = YearValue FROM #DistinctYears;

    IF @FirstYear = 1
        SET @FirstYear = 0;
    ELSE
        SET @PartitionFunctionSQL += ', ';

    SET @PartitionFunctionSQL += CAST(@Year AS NVARCHAR(MAX));

    DELETE FROM #DistinctYears WHERE YearValue = @Year;
END
```

2. Test Unitaires :

2.1. Écriture & Exécution de Cas de Test SQL

V. Creation des utilisateurs :

1. Utilisateurs de la Base de Données & Rôles :

- 1.1. Création des Utilisateurs et Attribution de Rôles
- 1.2. Attribution de Permissions en Fonction des Responsabilités

```
CREATE LOGIN StagingAreaAdmin WITH PASSWORD = 'AllahSave.1234/' ;
CREATE LOGIN SalesDataMartAdmin WITH PASSWORD = 'AllahSave.1234/' ;
CREATE LOGIN InventoryDataMartAdmin WITH PASSWORD = 'AllahSave.1234/' ;
CREATE LOGIN DataWareHouseAdmin WITH PASSWORD = 'AllahSave.1234/' ;
```



```
CREATE USER StagingAreaAdminUser FOR LOGIN StagingAreaAdmin ;
CREATE USER SalesDataMartAdminUser FOR LOGIN SalesDataMartAdmin ;
CREATE USER InventoryDataMartAdminUser FOR LOGIN InventoryDataMartAdmin ;
CREATE USER DataWareHouseAdminUser FOR LOGIN DataWareHouseAdmin ;
```

V.