

---

## Mise en Place et Optimisation d'un Entrepôt de Données et Analyse avec Power B

---

Yassine Essadi  
(ECODE)

# Introduction

Dans le paysage commercial en constante évolution d'aujourd'hui, les données sont devenues le moteur des organisations, orientant les décisions cruciales et les connaissances essentielles dans diverses industries. La gestion efficace et l'utilisation des données sont devenues primordiales, en particulier dans le domaine du commerce électronique, où d'énormes quantités d'informations sont générées quotidiennement. Ce projet se lance dans un voyage visant à optimiser la gestion des données au sein d'une plateforme de E-commerce, en utilisant un ensemble robuste d'outils et de techniques pour exploiter la puissance des données.

L'objectif central de ce projet est de transformer les données brutes en informations exploitables, améliorant les processus de prise de décision, la compréhension des tendances du marché et l'évaluation de la performance des fournisseurs. Pour ce faire, nous exploitons des technologies de pointe telles que Talend pour les processus d'Extraction, de Transformation et de Chargement (ETL), SQL Server comme solution de gestion de données, et Power BI pour l'analyse des données. De plus, notre approche intègre des mesures rigoureuses de sécurité des données et de conformité avec le Règlement Général sur la Protection des Données (RGPD) pour protéger les informations sensibles.

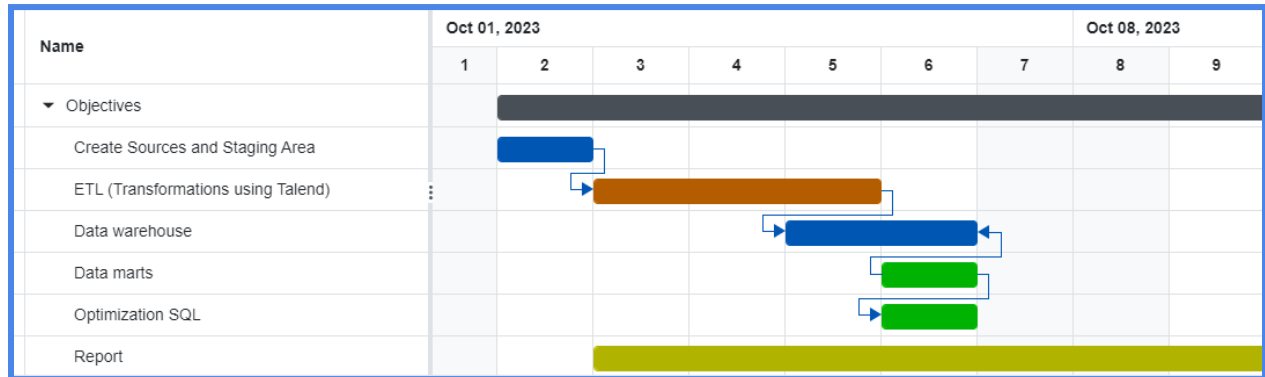
Avant de nous plonger dans les différentes phases de ce projet, nous commençons par appliquer un script qui facilite la segmentation de notre jeu de données en formats JSON et CSV, préparant ainsi le terrain pour le traitement ultérieur des données.

## Objectifs

- ETL
  - ETL
  - Politiques RGPD pour les Données Sensibles
- Schéma de Constellation Rapide (Fact Constellation Schema)
- SCD (Type 1)
- Data Marts Physiques
- Les Rôles
- Validation de la Logique de Transformation
- Optimisation
- Analytique avec Power BI
- Conclusion

# Planification

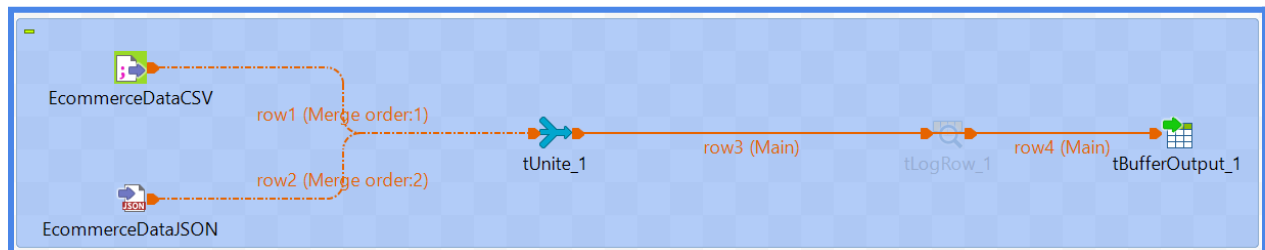
Avant de commencer le véritable travail, permettez-moi de vous présenter ce que nous allons aborder. Je vais afficher les tâches à accomplir et le temps que j'ai passé sur chaque tâche en utilisant un diagramme de Gantt.



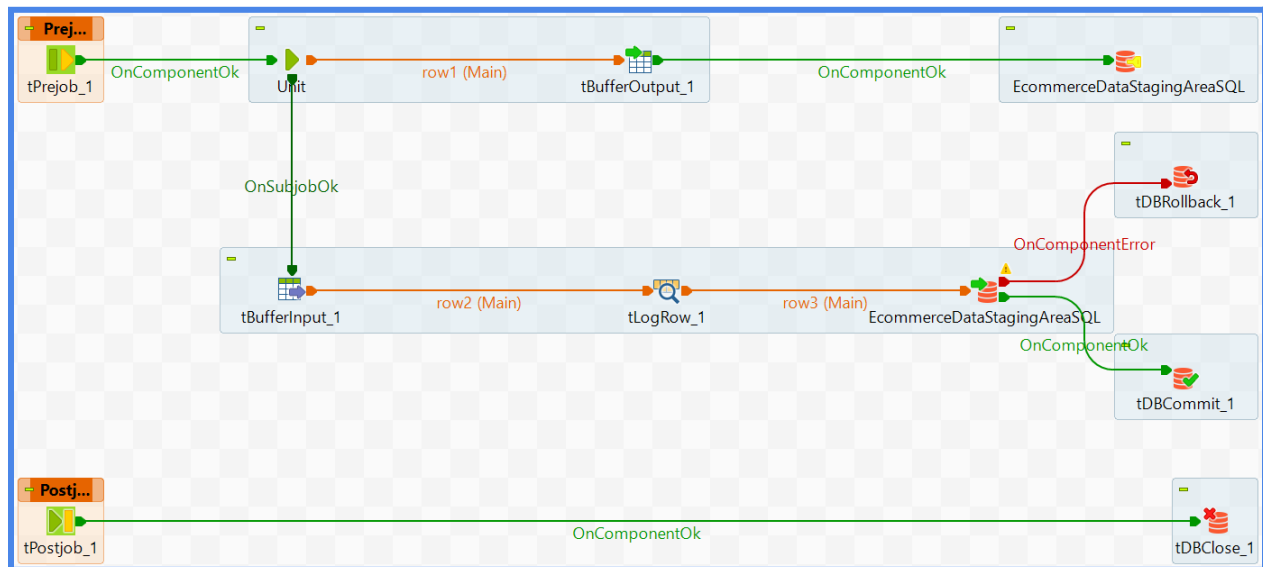
## ETL :

Dans cette étape du projet, préalablement à la transformation des données, j'ai mis en place la zone de staging pour stocker les diverses sources de données (CSV, JSON) à l'état brut, en vue de leur chargement dans le processus ETL de Talend pour la transformation.

Tout d'abord, unissez les fichiers en utilisant le composant tUnit:



## Travail de la Zone de Staging :



Après avoir terminé la partie de la zone de préparation, je vais maintenant commencer à récupérer ces données de la zone de préparation pour effectuer quelques transformations. Tout d'abord, je vais commencer par créer une connexion entre Talend et SQL Server pour récupérer ces informations, comme vous pouvez le voir dans l'image ci-dessous:

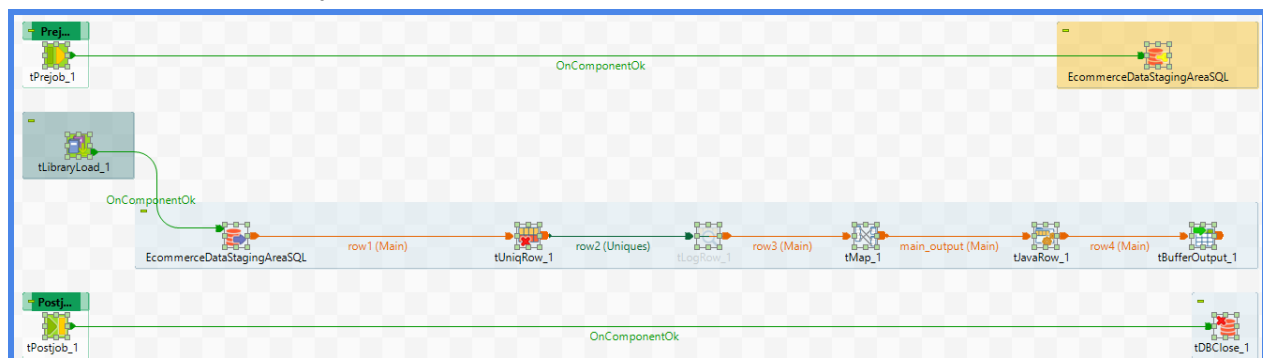
## Transformation :

Dans l'étape de transformation, j'ai transformé quelques colonnes :

**Colonne Date** : J'ai corrigé la date au format uniforme "dd-MM-yyyy".

**Suppliers contact** : J'ai remplacé les valeurs nulles par "Unkown".

**Colonne ProductPrice**: J'ai effectué des corrections en se basant sur les colonnes 'TotalAmount' et 'QuantitySold', en divisant le montant total par la quantité vendue.



## RGPD :

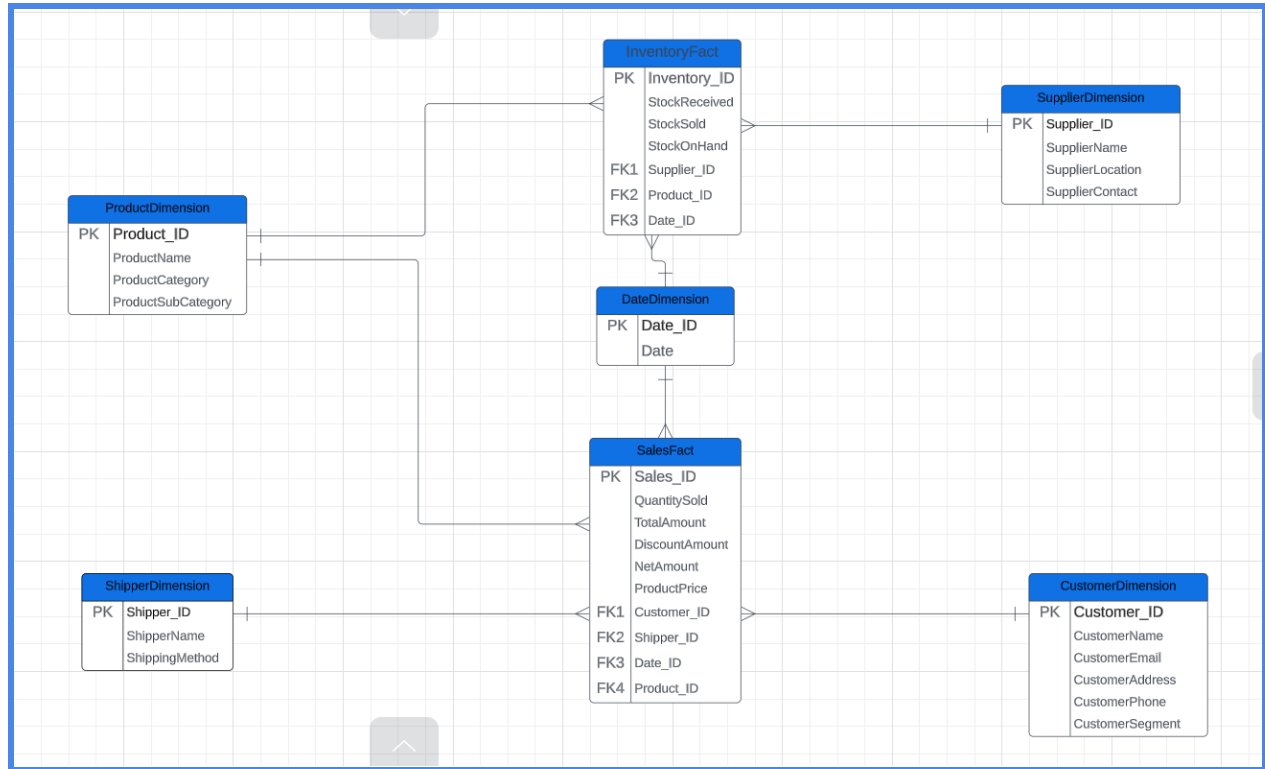
Pour les colonnes sensibles telles que customerEmail, customerAddress, SupplierContact, SupplierLocation, CustomerPhone, j'ai appliqué la norme RGPD. J'ai crypté les colonnes pour des raisons de sécurité.

```
StandardPBESStringEncryptor encryptor = new StandardPBESStringEncryptor();
encryptor.setAlgorithm("PBEWithMD5AndDES");
encryptor.setPassword("Essadi");

output_row.Date = input_row.Date;
output_row.ProductName = input_row.ProductName;
output_row.ProductCategory = input_row.ProductCategory;
output_row.ProductSubCategory = input_row.ProductSubCategory;
output_row.ProductPrice = (input_row.TotalAmount / input_row.QuantitySold);
output_row.CustomerName = input_row.CustomerName;
output_row.CustomerEmail = encryptor.encrypt(input_row.CustomerEmail);
output_row.CustomerAddress = encryptor.encrypt(input_row.CustomerAddress);
output_row.CustomerPhone = encryptor.encrypt(input_row.CustomerPhone);
output_row.CustomerSegment = input_row.CustomerSegment;
output_row.SupplierName = input_row.SupplierName;
output_row.SupplierLocation = encryptor.encrypt(input_row.SupplierLocation);
output_row.SupplierContact = encryptor.encrypt(input_row.SupplierContact);
output_row.ShipperName = input_row.ShipperName;
output_row.ShippingMethod = input_row.ShippingMethod;
output_row.QuantitySold = input_row.QuantitySold;
output_row.TotalAmount = input_row.TotalAmount;
output_row.DiscountAmount = input_row.DiscountAmount;
output_row.NetAmount = input_row.NetAmount;
output_row.StockReceived = input_row.StockReceived;
output_row.StockSold = input_row.StockSold;
output_row.StockOnHand = input_row.StockOnHand;
```

## Schéma de Constellation Rapide (Fact Constellation Schema)

Après avoir terminé la transformation, nous devons modéliser des données en utilisant la constellation de faits.

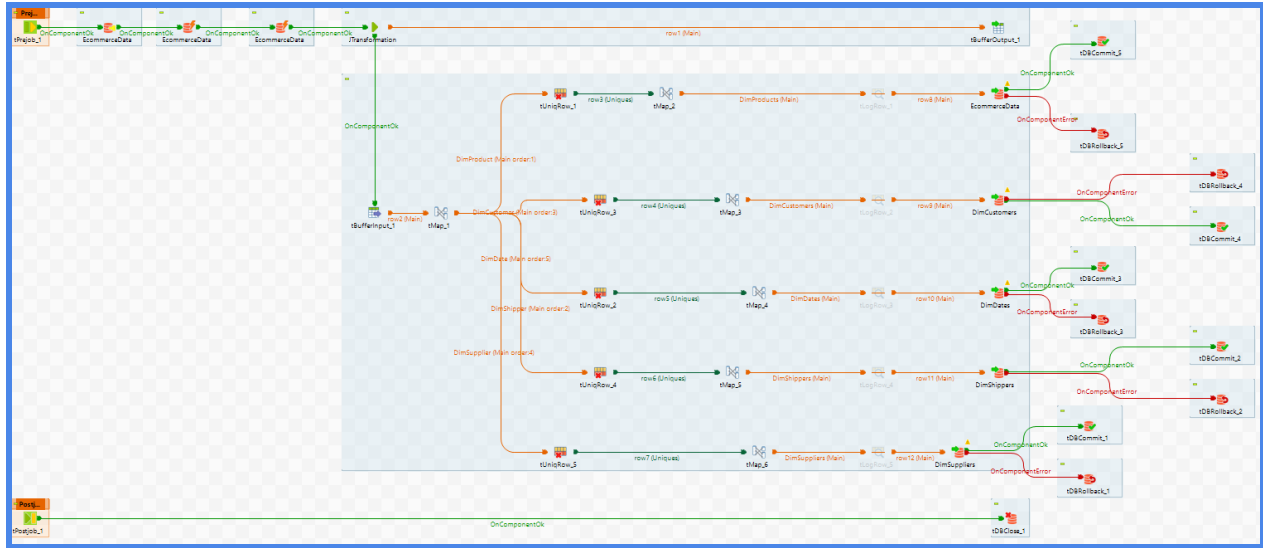


Dans le cadre de ce projet, j'ai réalisé la modélisation d'un schéma de constellation , une structure de base de données complexe qui permet d'organiser efficacement et de relier des données provenant de différentes sources. Les tables de dimension que j'ai créées comprennent la "DateDimension," la "ProductDimension," la "CustomerDimension," la "SupplierDimension," et la "ShipperDimension." Ces tables de dimension servent à stocker des informations clés sur les dates, les produits, les clients, les fournisseurs et les expéditeurs, respectivement.

En complément, j'ai également élaboré des tables de fait, notamment la "SalesFact" et la "InventoryFact." La table "SalesFact" enregistre des données liées aux ventes, telles que la date de la transaction, le produit vendu, le client, le montant total, les réductions, et bien plus encore. Quant à la table "InventoryFact," elle conserve des informations essentielles sur la gestion des stocks, telles que les quantités de produits reçues, vendues et en stock à différentes périodes.

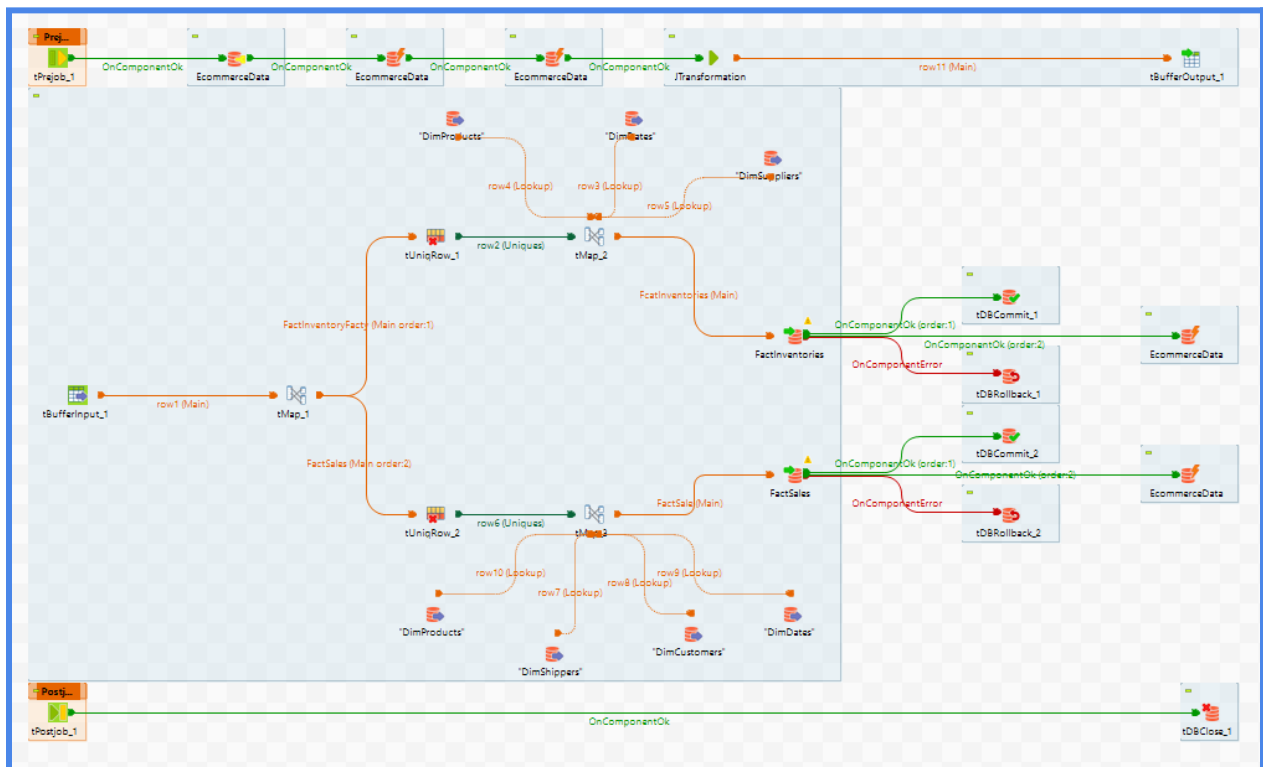
### Separation DimSubJob :

Sur la base du diagramme Fact Constellation, j'ai créé un "Job" pour séparer les tables de dimension et générer des clés primaires pour chaque table de dimension, ainsi qu'un "Job" pour les tables de faits.



## Separation FactJob :

Ce Job consiste à séparer la table des faits en fonction des tables de dimensions que nous avons déjà insérées dans la base de données en récupérant ces tables depuis le data warehouse et en essayant d'établir une relation entre ces tables, comme vous pouvez le voir dans l'image ci-dessous.



# SCD (Type 1)

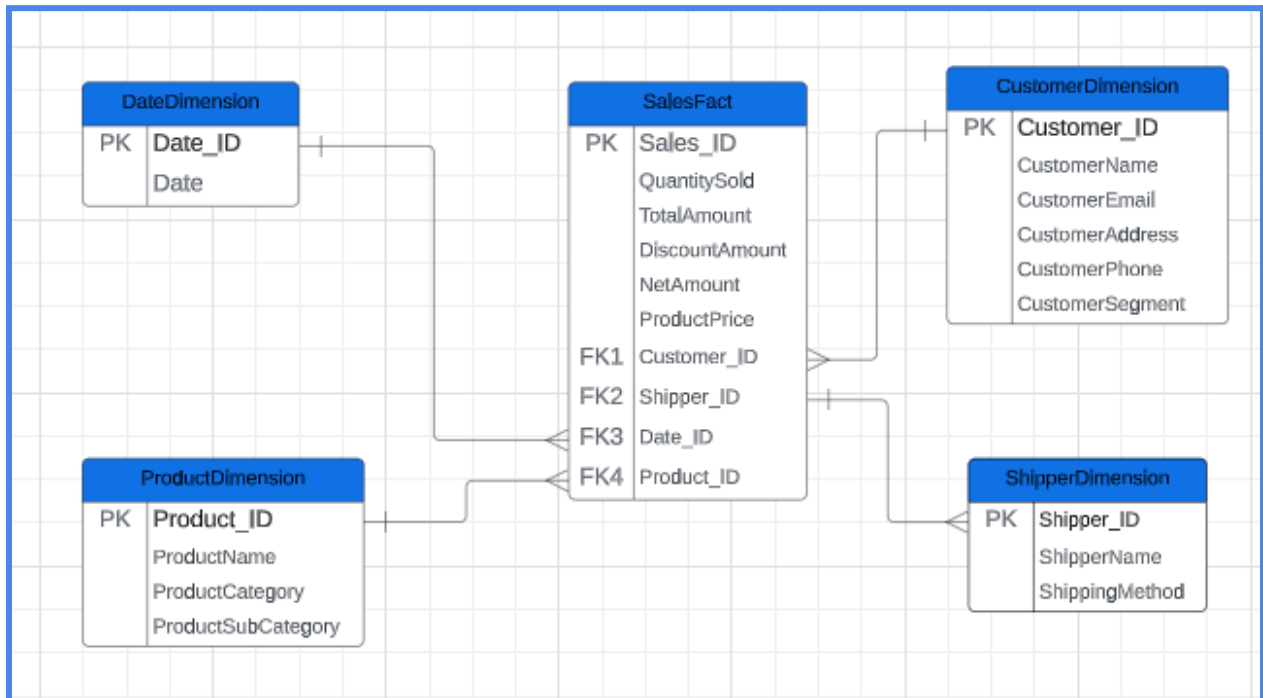
Dans ce type, les anciennes données sont simplement écrasées par de nouvelles données, ce qui signifie que les changements historiques ne sont pas préservés, le rendant adapté aux données non historiques.

Username	"Yushin"	Password	*****	
Table	"FactInventories"			
Action on table	Create table if does not exist	<input type="checkbox"/> Turn on identity insert	Action on data	Update or insert
<input type="checkbox"/> Specify identity field				
Schema	Built-In	Edit schema	<input type="checkbox"/> Sync columns	

## Data Marts Physiques

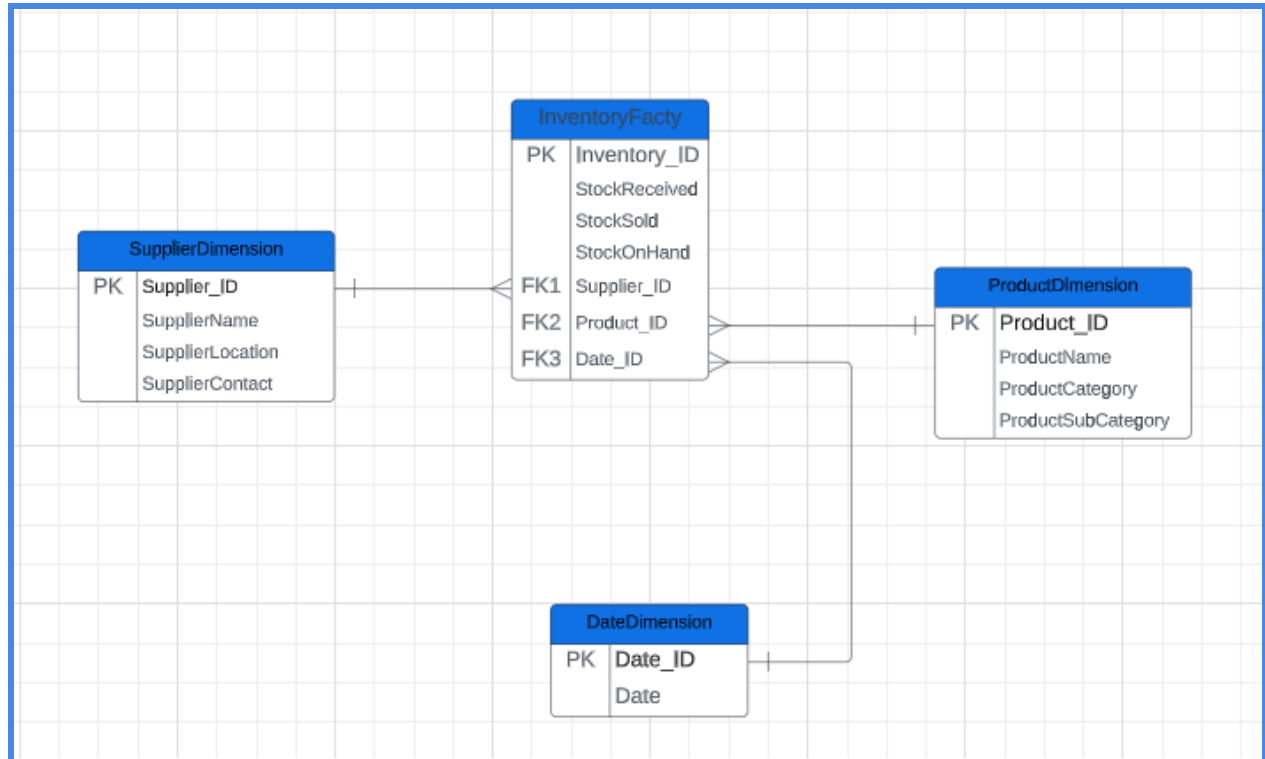
Après avoir terminé la création du data warehouse à ce stade, nous passerons aux datamarts, étant donné qu'ils reposent sur le data warehouse. Dans ce cas, nous allons créer un modèle de données pour servir de point de départ à notre travail.

Commençons par les datamarts pour les ventes, en priorité:

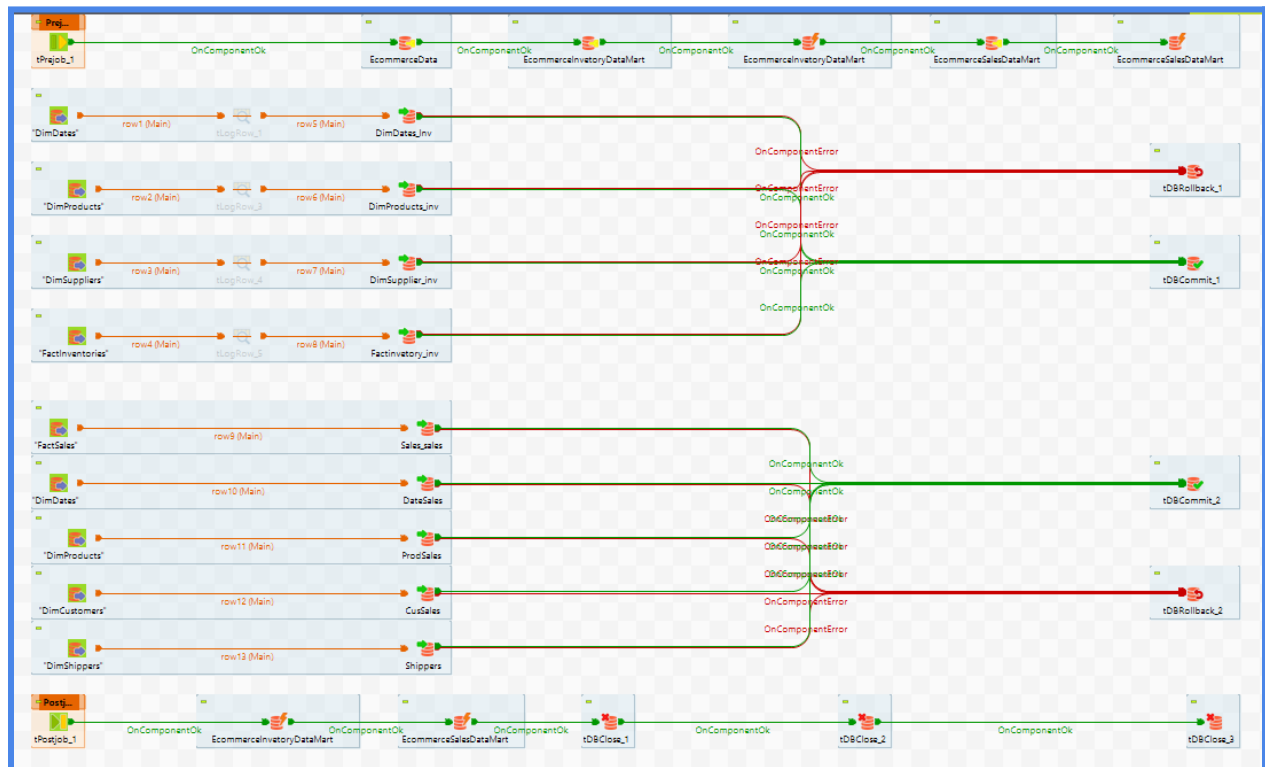


D'abord, commençons par les datamarts pour les inventaires:



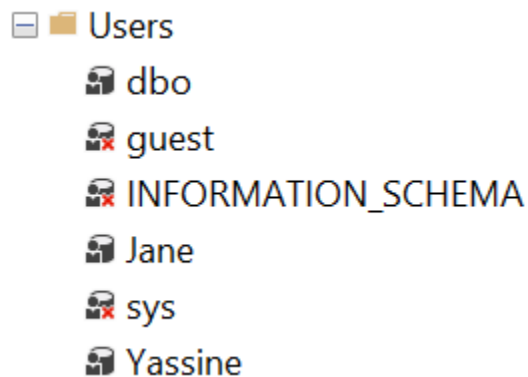


Maintenant que toutes les structures sont en place, voyons comment créer un datamart physique en utilisant la puissance de Talend, en nous basant sur ces schémas. Pour ce processus, j'ai créé un seul job appelé "Jloading\_Datamarts" pour gérer les deux schémas, comme vous pouvez le voir dans l'image ci-dessous.



# Les Rôles

Maintenant, dans cette partie, nous allons attribuer des rôles aux utilisateurs pour leur accorder des privilèges d'accès à ces bases de données."Le script est inclus dans le fichier SQL."



# Validation de la Logique de Transformation

J'ai appliqué les tests unitaires essentiels pour notre entrepôt de données à l'aide de l'outil TSqlT. Ces tests unitaires ont été spécifiquement conçus pour évaluer la qualité et l'intégrité des données."Le script est inclus dans le fichier SQL."

Tests unitaires sur les inventaires.

```
+-----+
|Test Execution Summary|
+-----+

|No|Test Case Name                                     |Dur(ms)|Result |
+-----+-----+-----+-----+
|1 |[DatawareHouseTestingInventory].[test ForeignKeysAreValid]|  29663|Success|
+-----+-----+-----+-----+
Test Case Summary: 1 test case(s) executed, 1 succeeded, 0 skipped, 0 failed, 0 errored.
+-----+

Completion time: 2023-10-09T18:17:45.3370853+01:00
```

Tests unitaires sur les ventes.

```
(1 row affected)

+-----+
|Test Execution Summary|
+-----+

|No|Test Case Name                                     |Dur (ms)|Result |
+---+-----+-----+-----+
|1 |[DatawareHouseTestingInventory].[test ForeignKeysAreValid]| 3211|Success|
+---+-----+-----+-----+

Test Case Summary: 1 test case(s) executed, 1 succeeded, 0 skipped, 0 failed, 0 errored.
-----

Completion time: 2023-10-09T18:34:31.0330829+01:00
```

# Optimisation

## Indexation:







J'ai réalisé des opérations d'indexation essentielles au niveau de notre base de données SQL, en créant des index sur les colonnes "ProductName", "ProductCategory" et "ProductSubCategory", "Customer\_ID", "Sales\_ID". L'objectif était d'apporter une amélioration significative aux performances de nos requêtes de recherche et de filtrage de produits.

L'indexation de ProductName nous permet de rechercher rapidement des produits spécifiques par leur nom, facilitant ainsi l'accès aux informations pour nos utilisateurs et améliorant leur expérience globale. De manière similaire, l'indexation de ProductCategory et ProductSubCategory simplifie la catégorisation et la recherche de produits dans leurs catégories respectives, ce qui revêt une grande importance pour notre gestion des stocks et l'analyse des produits. "Le script est inclus dans le fichier SQL."

## Partitionnement:

Pendant ce projet, j'ai divisé nos DataMarts, à savoir "Ecom\_Inventory\_DM", en fonction des années 2021, 2022 et 2023. Cette stratégie de partitionnement a impliqué la séparation de nos données en trois segments distincts, chacun correspondant à une année particulière.

Le partitionnement revêt une importance stratégique pour notre entreprise, car il permet une gestion plus efficace des données et améliore les performances d'accès. En segmentant nos données par année, nous simplifions l'extraction et l'analyse d'informations spécifiques à chaque année. Cette approche est particulièrement bénéfique pour l'analyse des données.

  inventory_2021.ndf	10/8/2023 7:16 PM	SQL Server Database ...	2,048 KB
  inventory_2022.ndf	10/8/2023 7:16 PM	SQL Server Database ...	2,048 KB
  inventory_2023.ndf	10/8/2023 7:16 PM	SQL Server Database ...	2,048 KB

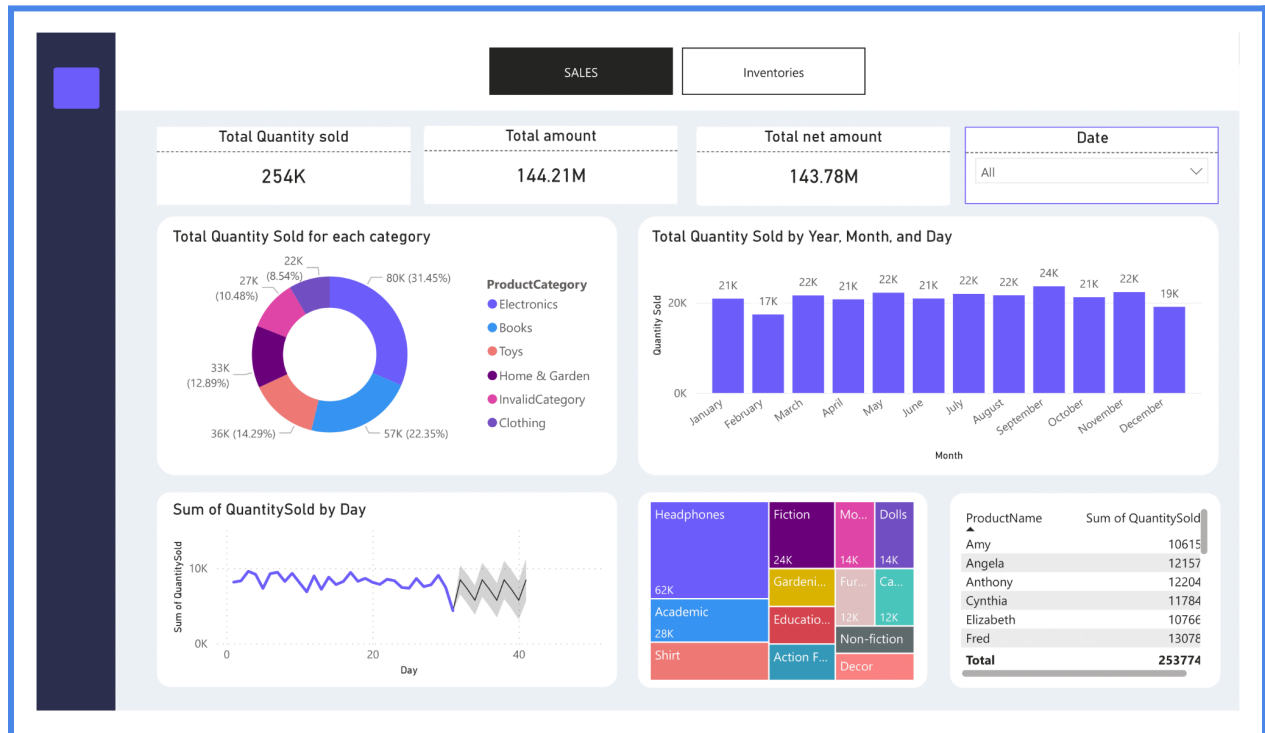
# Analytique avec Power BI:

Donc, après avoir achevé tous les processus nécessaires pour rendre ces données utiles, nous allons maintenant passer à l'utilisation de Power BI pour donner du sens à ces données.

Commençons.

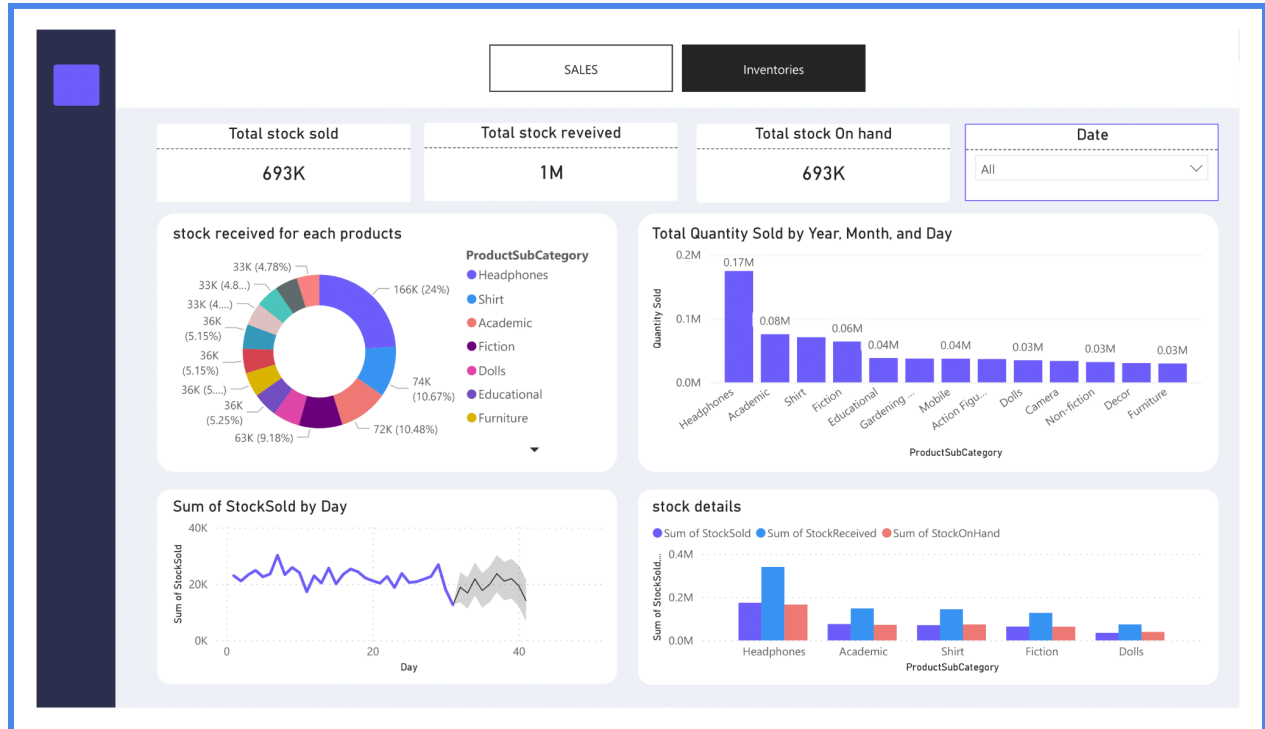
## **Dashboard des ventes:**

Ce **Dashboard représente** l'analyse des ventes en suivant les variations de la quantité vendue au fil du temps et en identifiant la catégorie la plus vendue en fonction de la quantité vendue.



## Dashboard des inventoriés:

Dans ce tableau de bord, les informations sur les inventaires telles que les stocks en main, les stocks reçus et les stocks vendus sont affichées, et ces changements sont suivis en fonction du temps.



## Conclusion :

En conclusion, ce projet a utilisé Talend pour des processus ETL efficaces, transformant et préparant les données provenant de diverses sources, notamment les stocks, les ventes, les clients, les transporteurs, les produits, les dates, les fournisseurs, etc., dans le domaine du commerce électronique. Nous avons utilisé un entrepôt de données robuste pour stocker et centraliser cette abondance d'informations, optimisant l'accessibilité et l'intégrité des données. En créant deux datamarts dédiés, l'un pour les ventes et l'autre pour les inventaires, nous avons adapté nos données à des besoins d'analyse spécifiques.

L'intégration de Power BI nous a fourni un outil puissant pour extraire et visualiser des données à partir de ces datamarts, permettant des visualisations perspicaces et interactives. Cette approche complète a non seulement amélioré nos capacités de gestion des données, mais nous a également permis de prendre des décisions basées sur les données. En résumé, ce projet illustre la valeur d'un pipeline de données bien structuré, jetant les bases pour des informations basées sur les données et des stratégies commerciales éclairées.