Université Chouaib Doukkali
Ecole Nationale des Sciences Appliquées d'El Jadida
Département Télécommunications, Réseaux et Informatique

RAPPORT DE TRAVAUX PRATIQUES

Filière : 2ITE

Niveau : 2ère Année

Analyse des données avec le serveur **Mondrian**

Réalisé par :

Marcel Fassou HABA

Encadré par :

Pr. Hanine

Année Universitaire : 2025-2026

INTRODUCTION

L'analyse de données occupe une place essentielle dans la prise de décision des organisations. Avec la croissance des volumes d'informations, il devient nécessaire d'utiliser des outils performants pour préparer, structurer et explorer les données. Les systèmes **OLAP (Online Analytical Processing)** répondent à ce besoin en permettant une analyse multidimensionnelle efficace.

Le **serveur OLAP Mondrian** s'inscrit dans cette logique. C'est un moteur open source développé en Java, capable d'interroger des entrepôts de données via le langage **MDX**. Intégré à des outils comme **JasperReport**, il facilite la création de rapports et tableaux de bord interactifs, permettant une exploration dynamique des données selon plusieurs axes d'analyse.

Ce **travail pratique** vise à illustrer l'utilisation du moteur **Mondrian** pour la préparation, l'analyse et l'exploration d'une grande quantité de données, à travers son intégration avec **JasperReport**. L'objectif principal de ce TP est savoir comment utiliser le serveur OLAP Mondrian pour la préparation de l'analyse et l'exploration d'une large quantité de données.

Le TP se divise en deux parties :

- **Première partie** : présentation de l'outil Mondrian et de son intégration dans un environnement BI.
- **Seconde partie** : réalisation pratique incluant la **création de la base de données**, la **configuration de JasperReport**, et l'**intégration du moteur Mondrian** pour l'analyse multidimensionnelle.

Ce travail permettra de mieux appréhender le processus complet d'analyse de données et l'apport du serveur OLAP Mondrian dans la prise de décision.

PRÉSENTATION DE L'OUTIL UTILISÉ

Le **serveur OLAP Mondrian** est un moteur **d'analyse multidimensionnelle** open source développé en **Java**, largement utilisé dans le domaine de la **Business Intelligence (BI)**. Il permet d'explorer, d'agréger et d'analyser de grandes quantités de données issues d'une **base relationnelle**, en les organisant sous forme de **cubes OLAP**. Cette représentation multidimensionnelle facilite la compréhension des phénomènes complexes en offrant une vision globale et détaillée selon différents axes d'analyse (temps, produit, région, etc.).

Mondrian repose sur le langage **MDX (Multidimensional Expressions)**, spécialement conçu pour interroger et manipuler les cubes OLAP. Grâce à lui, les utilisateurs peuvent effectuer des requêtes analytiques avancées, réaliser des comparaisons temporelles, ou encore observer l'évolution des indicateurs de performance.

JasperReport est un puissant outil de **reporting et de visualisation de données**, également open source. Il permet de créer des **rapports dynamiques, interactifs et visuellement attractifs** à partir de diverses sources de données. Son intégration avec Mondrian renforce considérablement la capacité d'analyse en combinant la **puissance de calcul OLAP** et la **richesse visuelle** du reporting.

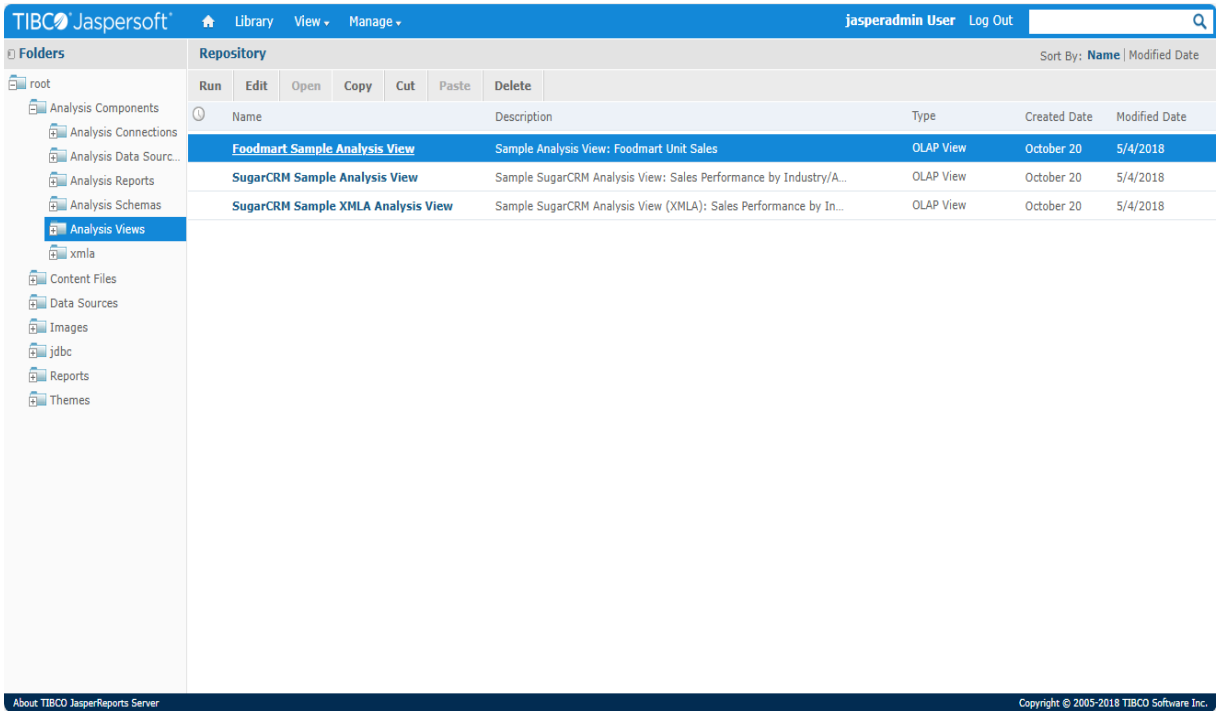
Dans cette intégration :

- **Mondrian** assure la **préparation et l'analyse multidimensionnelle** des données.
- **JasperReport** fournit une **interface graphique conviviale**, permettant de présenter les résultats sous forme de **rapports, tableaux de bord et graphiques interactifs**.

Cette collaboration offre un environnement complet où l'utilisateur peut **naviguer librement dans les données**, explorer les relations entre variables, effectuer des

analyses en profondeur (drill-down, roll-up, slice, dice), et produire des rapports clairs et pertinents pour la prise de décision.

La figure ci-dessous représente l'interface de JasperReport à travers JasperServer.



Atouts de Mondrian et JasperReport

- **Open source**, multiplateforme et extensible.
- Intégration fluide entre le moteur d'analyse OLAP et le module de reporting.
- Grande **puissance analytique** pour traiter et agréger de larges volumes de données.
- Visualisation interactive facilitant la compréhension des tendances et des indicateurs clés.

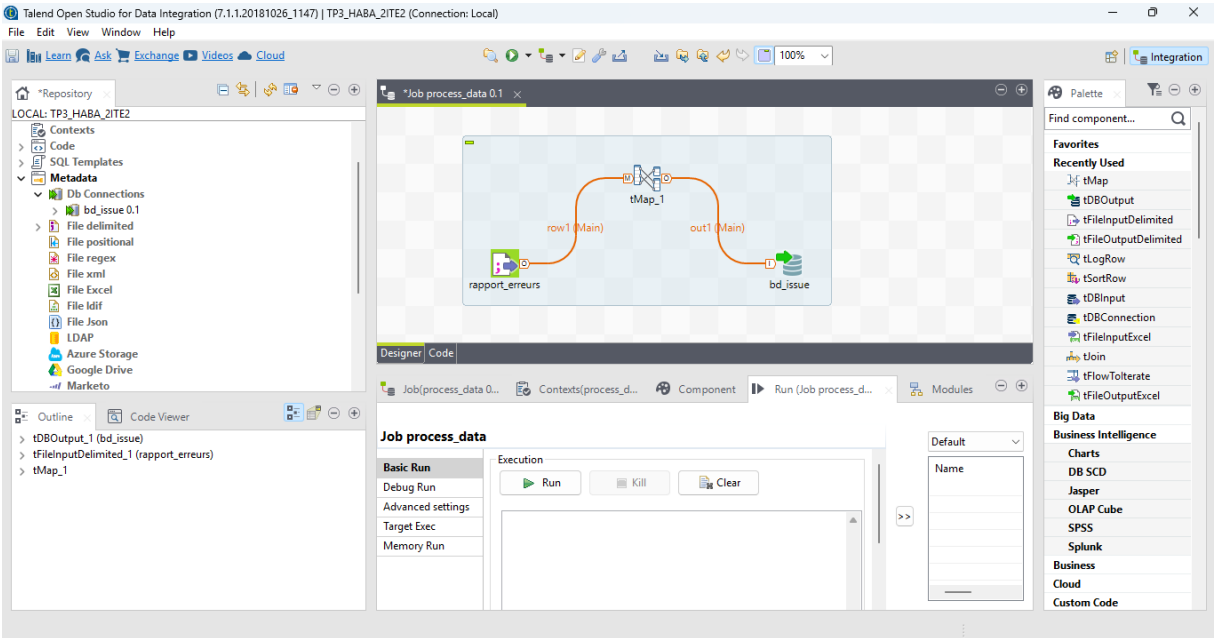
Ainsi, la combinaison **Mondrian–JasperReport** constitue une solution BI complète, offrant à la fois la **puissance de l'analyse multidimensionnelle** et la **clarté du reporting interactif**, indispensable pour explorer efficacement les données et soutenir les décisions stratégiques.

RÉALISATION ET RESULTATS

Création de la base de données

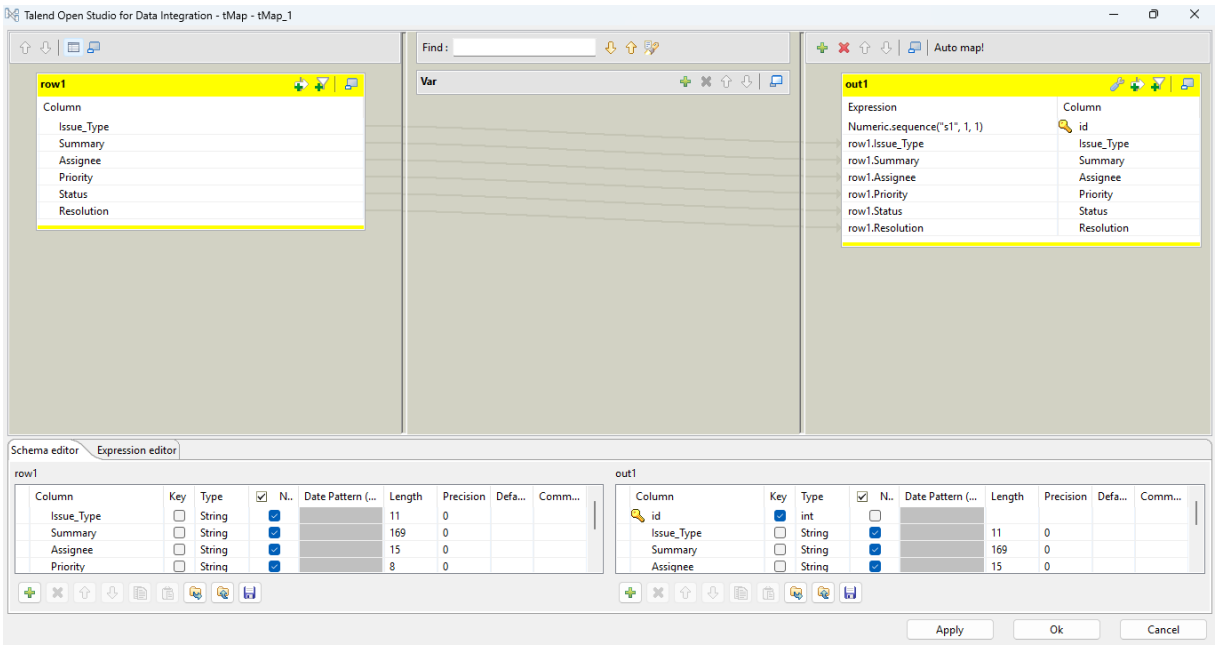
Dans cette partie, considérer comme la première étape, nous allons créer une base de données, en utilisant Talend Data Intégration, un outil que nous avons déjà utilisé dans le précédent TP. Nous allons copier les données du fichier délimité Rapport_Erreurs.csv dans une base de données mysql. Pour cela nous allons créer un job, ce job va nous permettre de mettre en place un pipeline de la préparation des sources de données à leur chargement dans la base de données (Processus ETL).

Cette image représente le job dédié à ce processus.



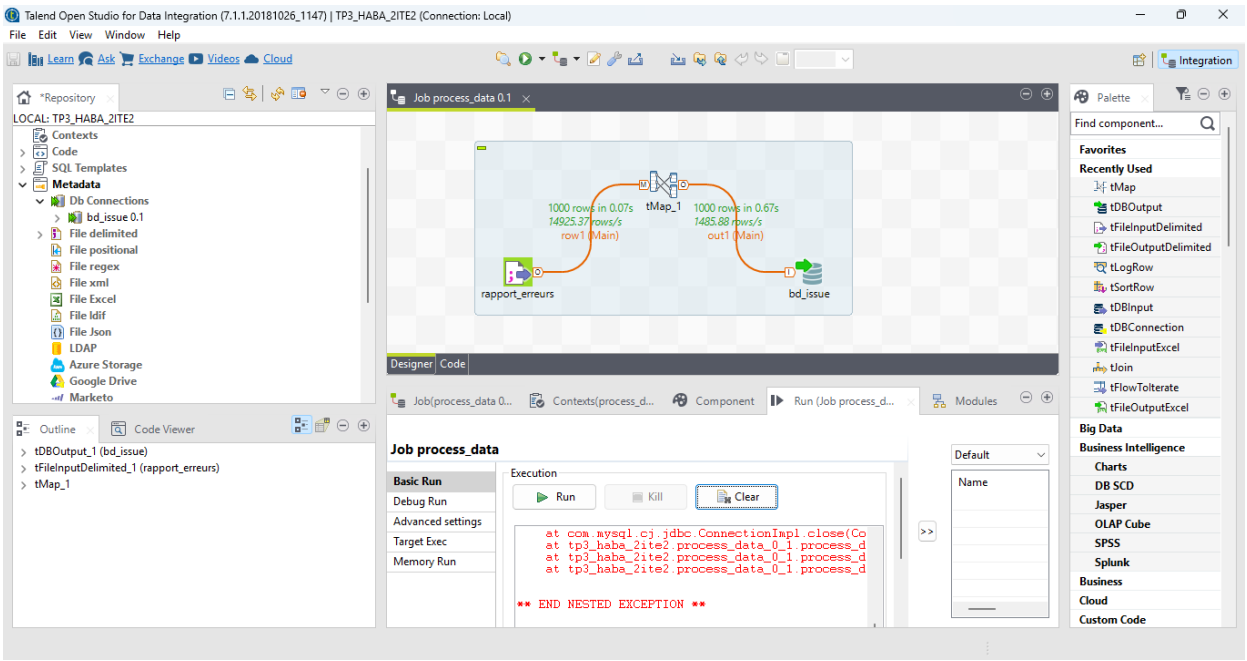
Etant donné que les données de départ manquent une colonne qui l'id, on utilise le composant **tMap** pour permettre l'ajout d'une nouvelle colonne id, qui sera généré automatiquement.

La figure ci-après présente la configuration du composant **tMap**.

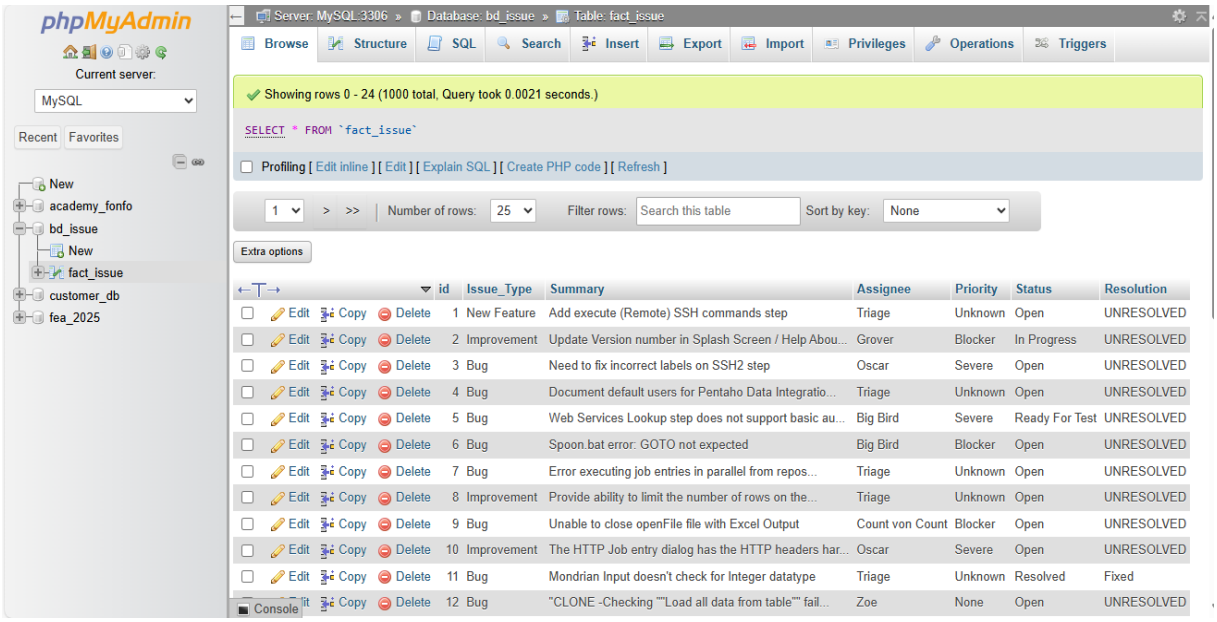


Après cette configuration, nous passons à l'exécution du workflow, on extrait les données, on les transforme et on les charge dans une base de données MySQL.

Ces images ci-après montrent la réussite de l'exécution du workflow et la présence des données dans la base de données.



A travers cette image, nous constatons la réussite de l'opération, vérifions dans la table créée si les données sont bien présentes.



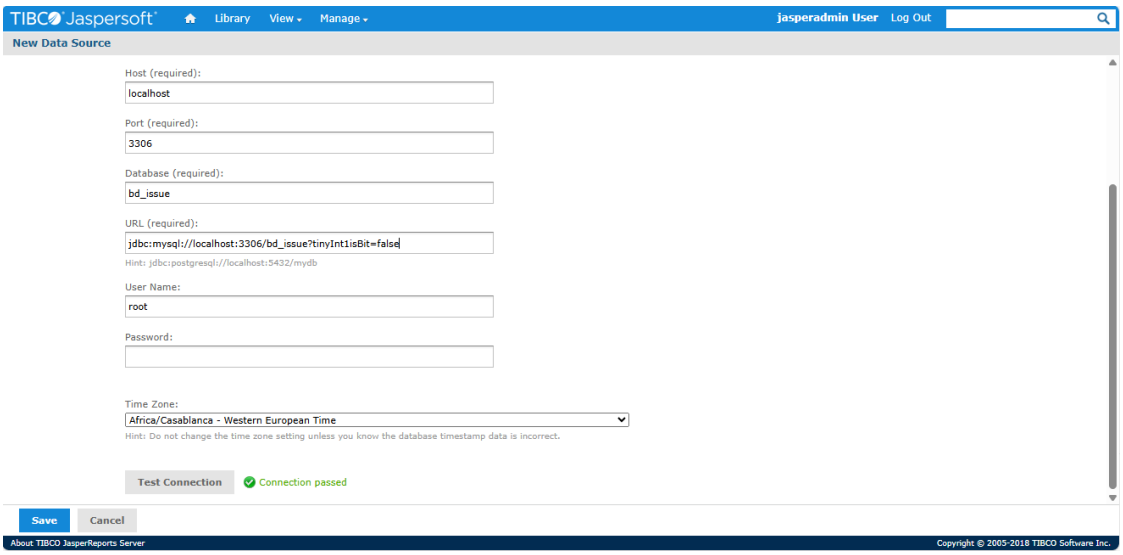
A partir de cette image, on peut bien constater que la table est remplie et que les données y sont présentes.

Configuration de JasperReport

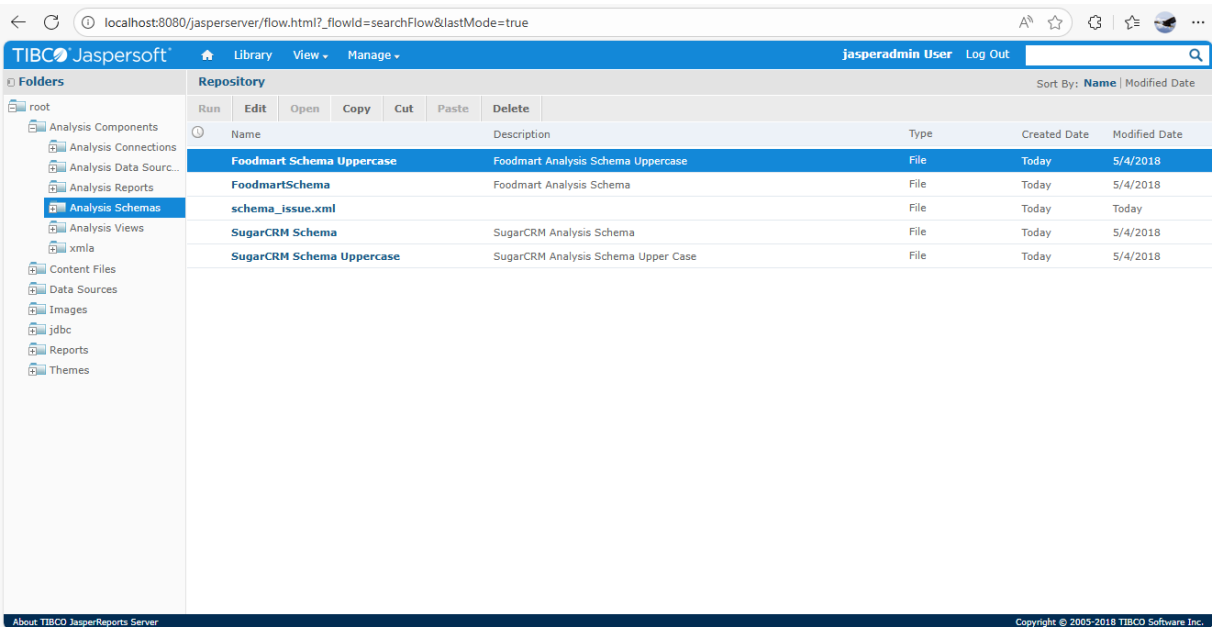
Pour cette partie nous utiliser JasperReport pour l'analyse des données multidimensionnelles. Nous allons travailler avec MySQL, base de données que nous avons utilisé dès le départ.

Commençons d'abord comme indiqué, à ajouter une nouvelle source de données dans le répertoire Analysis Data Sources.

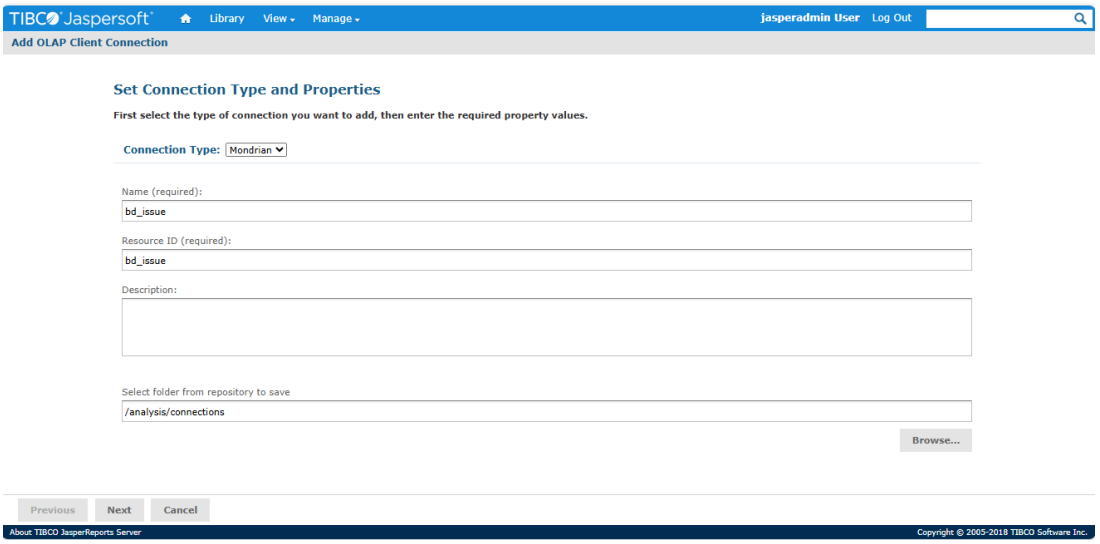
La figure ci-dessous illustre la réussite de cette opéreation.

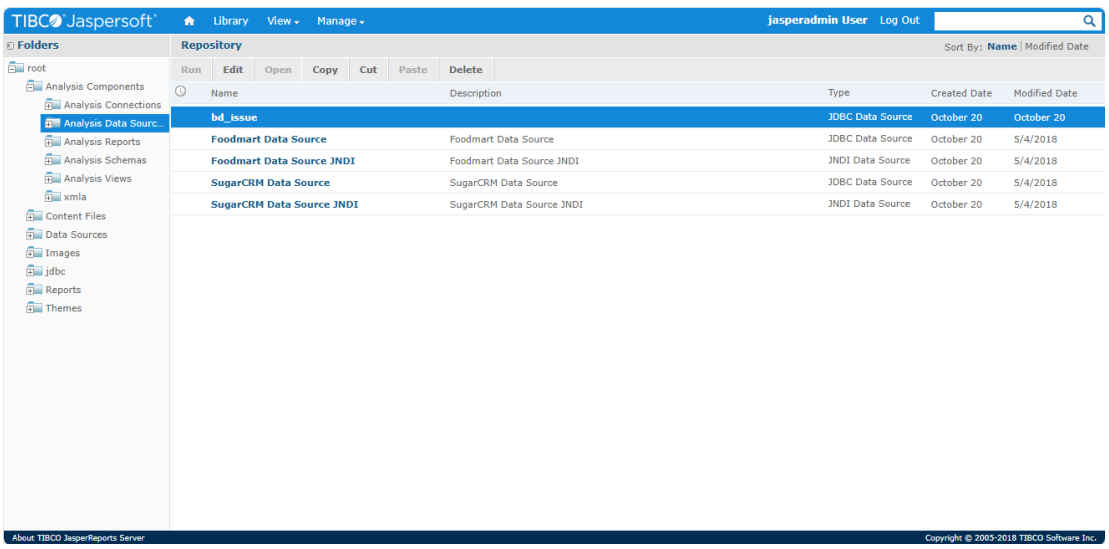


Après l'ajout de la source de données, il faut bien sur, ajouté le schéma OLAP qui représente la structure du cube OLAP. Donc nous allons ajouter cela.



Nous allons faire une connexion à la base de données bd_issue, en définissant une connexion de type Mondrian.

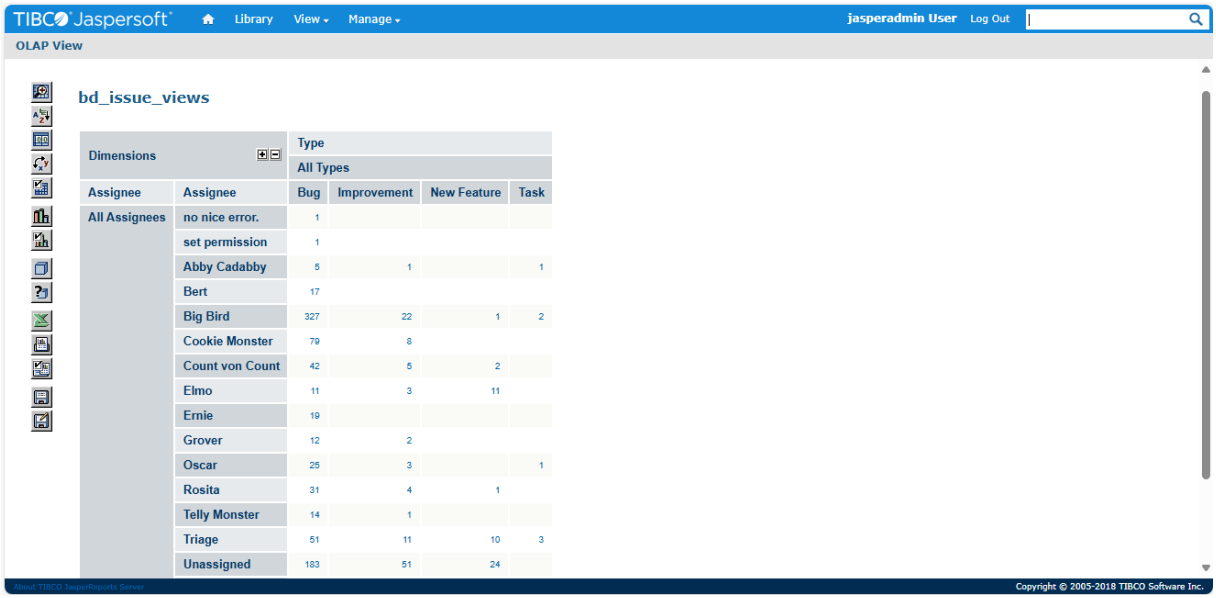




Une fois cela effectué, nous créons une vue, il faut noter que la vue représente une requête particulière. Nous allons exécuter la requête MDX suivante :

SELECT Type.Children ON COLUMNS, Assignee.Children ON ROWS FROM Issue

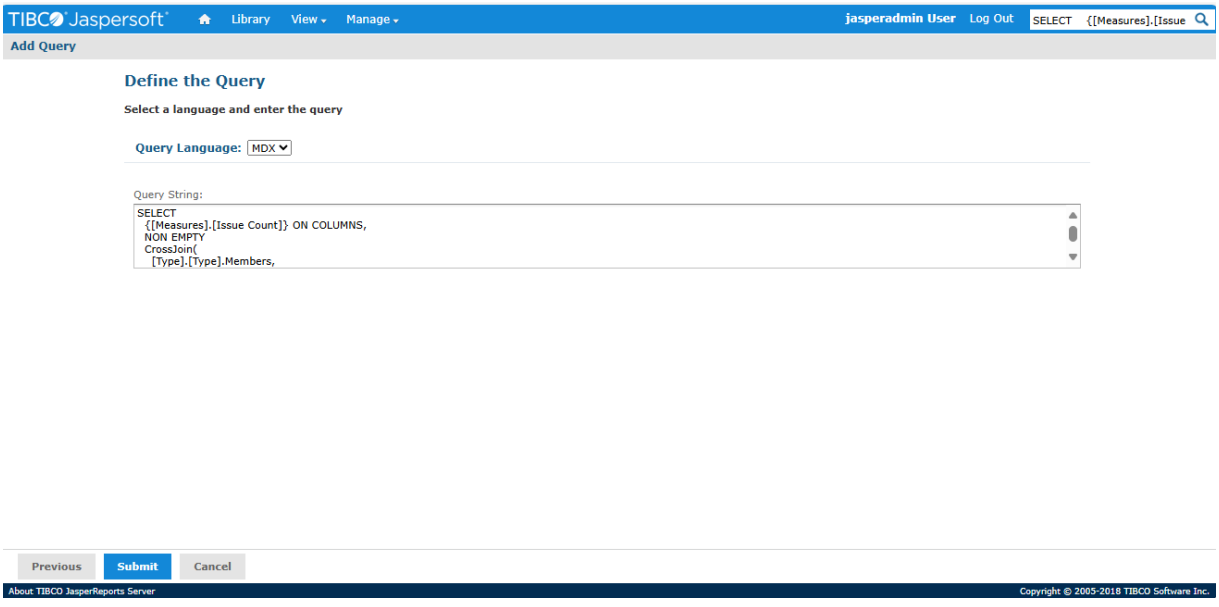
L'exécution de cette requête nous donne ceci : (voir l'image ci-dessous)



Création des nouvelles vues pour chaque travail :

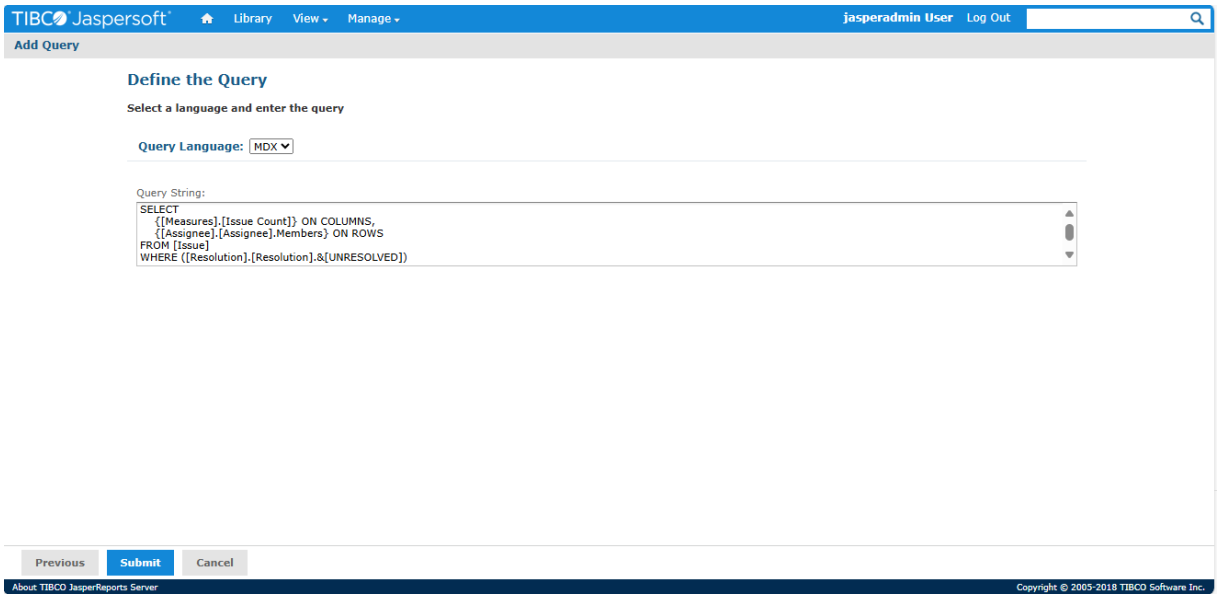
- **Nombre de statuts par type de problèmes**
  La requête MDX à utiliser est donnée par :

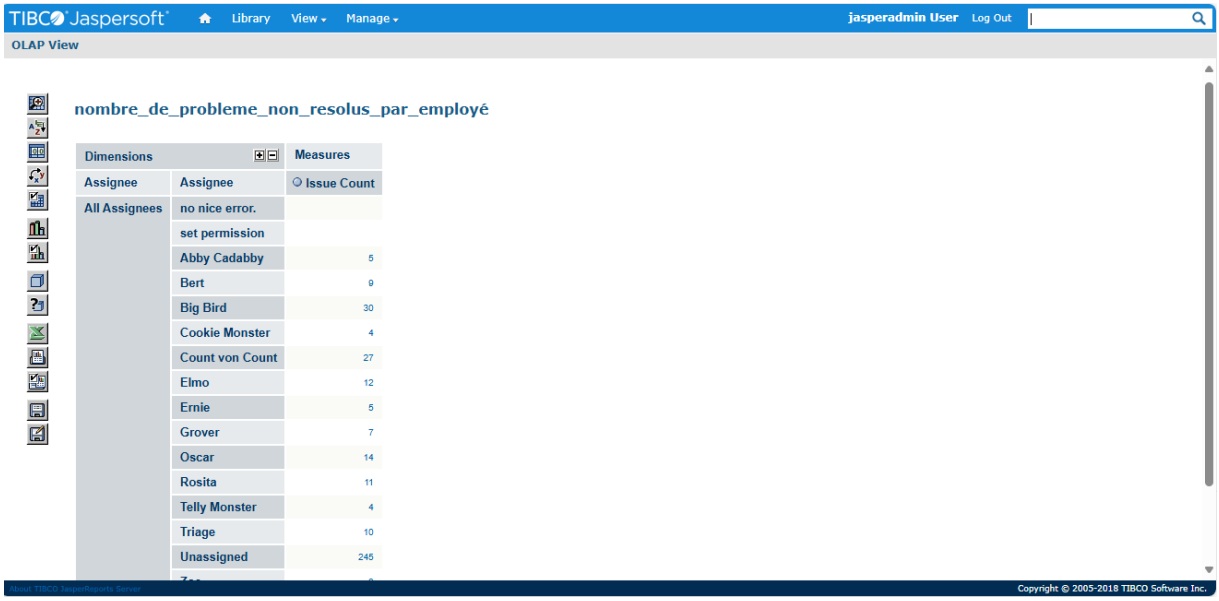
  SELECT {[Measures].[Issue Count]} ON COLUMNS, NON EMPTY CrossJoin([Type].[Type].Members,[Status].[Status].Members) ON ROWS FROM [Issue]



L'image suivante représente la vue associée à cette requête.

- **Montrer le nombre de problèmes non résolus (UNRESOLVED) par employé**

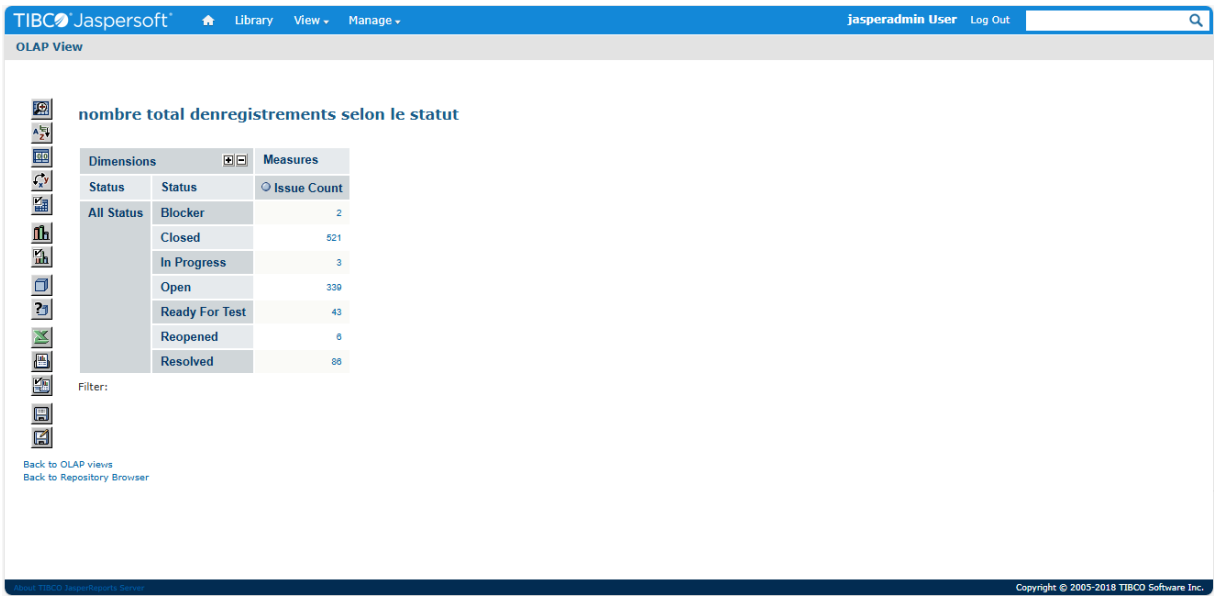




- **Montrons le nombre d'enregistrements dans la table par statut**

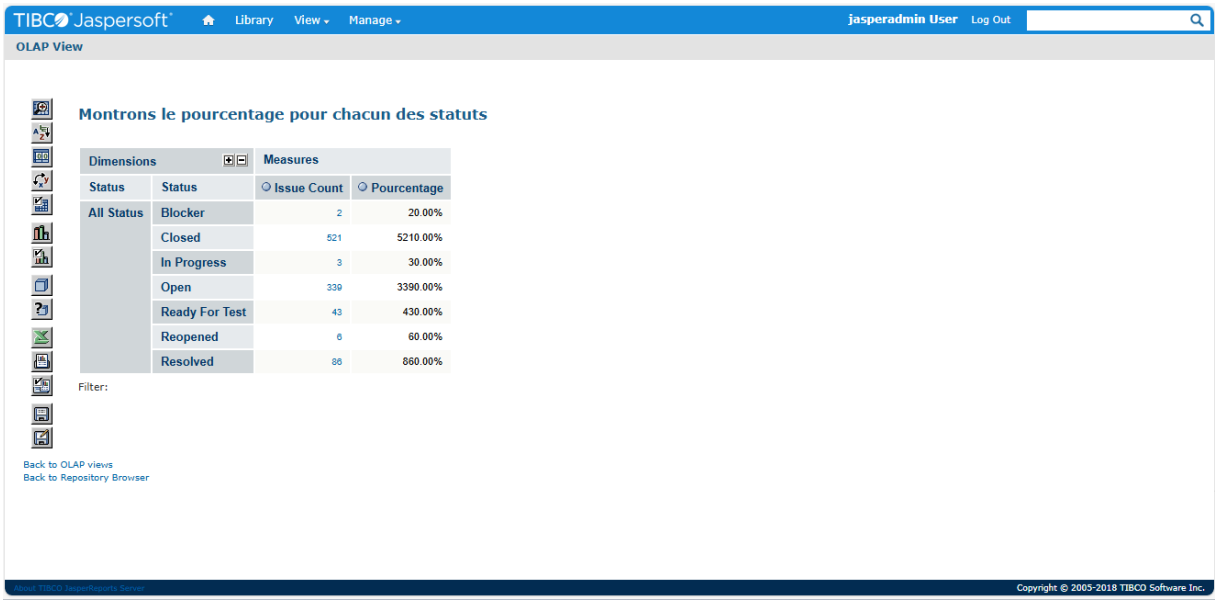
SELECT {[Measures].[Issue Count]} ON COLUMNS, {[Status].[Status].Members} ON ROWS FROM [Issue]

Dans la page suivant nous montrerons la vue associée à cette requête .



- **Montrons le pourcentage pour chacun des statuts**

```
WITH
    MEMBER [Measures].[Pourcentage] AS
  ([Measures].[Issue Count]) / ([Measures].[Issue Count], [Status].[All Status])
  * 100, FORMAT_STRING = "Percent"
SELECT
  {[Measures].[Issue Count], [Measures].[Pourcentage]} ON COLUMNS,
  [Status].[Status].Members ON ROWS
FROM [Issue]
```

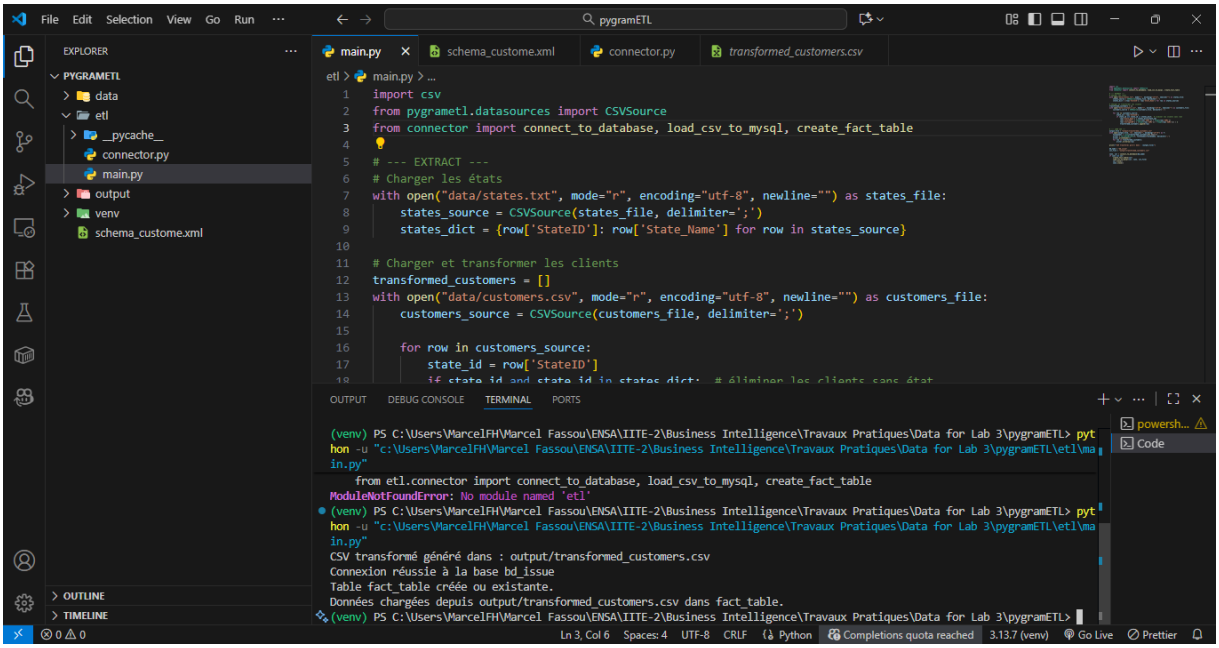


Utilisation du Framework PygramETL pour charger deux fichiers

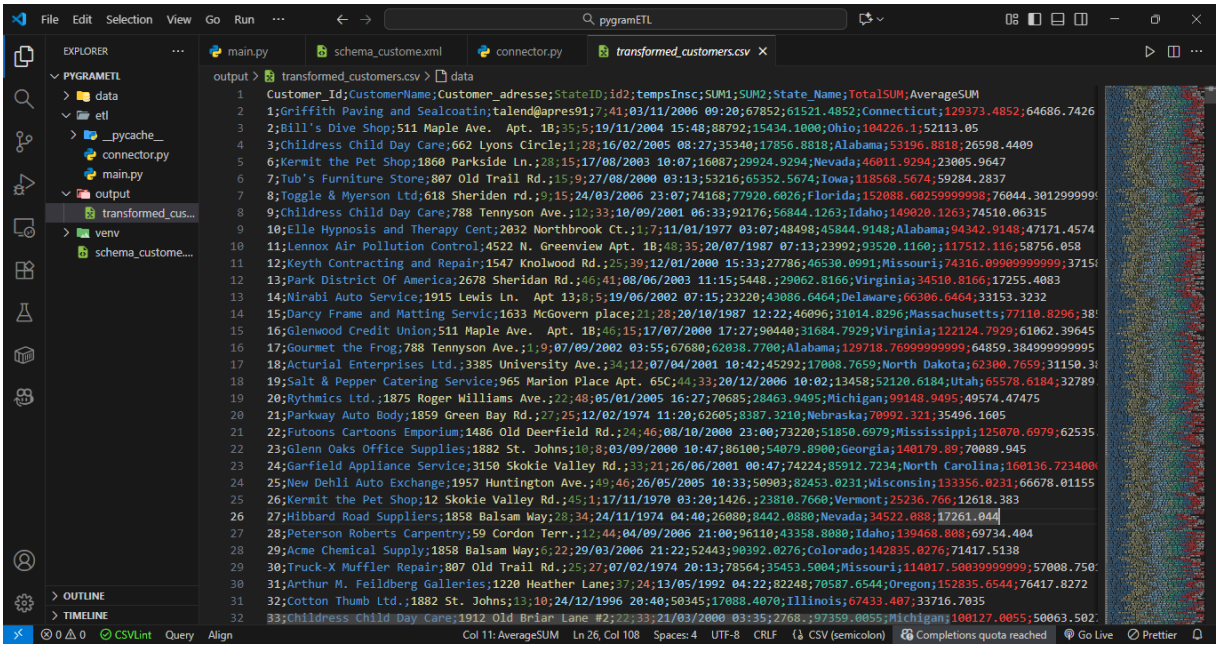
PygramETL est une **bibliothèque Python open source** conçue pour faciliter la **mise en œuvre de processus ETL (Extract, Transform, Load)**. Elle permet d'extraire des données depuis différentes sources, de les transformer selon des règles spécifiques, puis de les charger dans une base de données cible, le tout de manière **structurée et automatisée**.

Dans cette partie, nous allons utiliser PygramETL pour charger deux fichiers clients.csv et états.txt vers une table **fact_custome** tout en respectant différentes consignes données.

Cette image illustre le plan de travail et données générées :

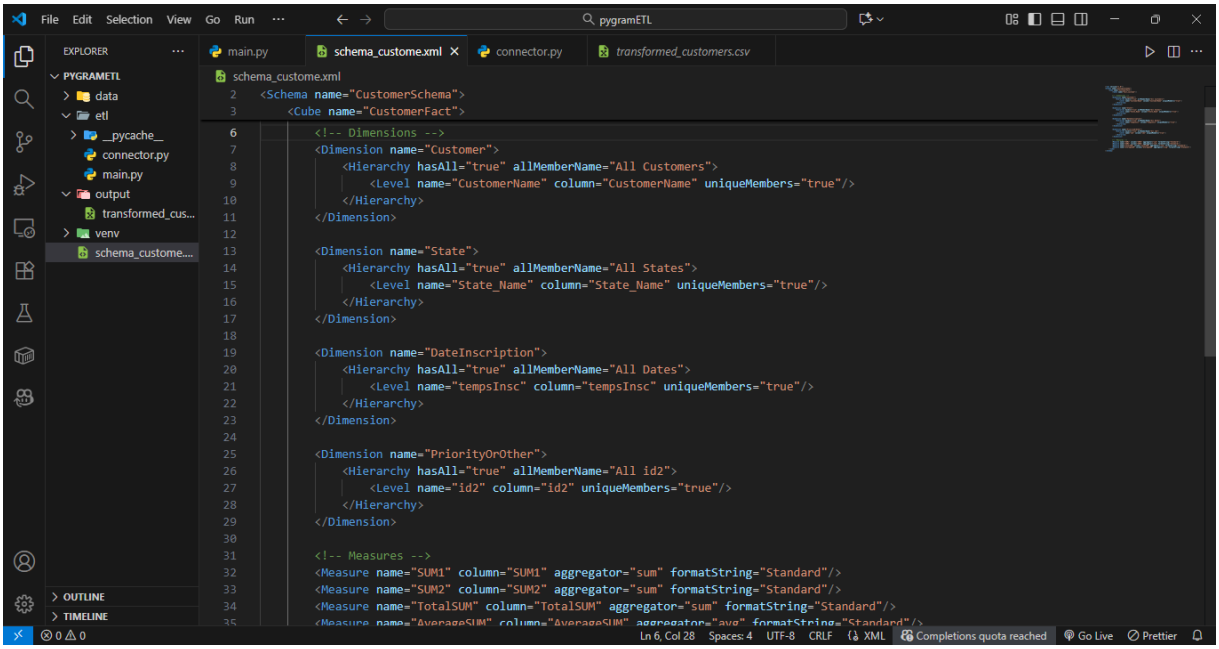


Les données générées sont :



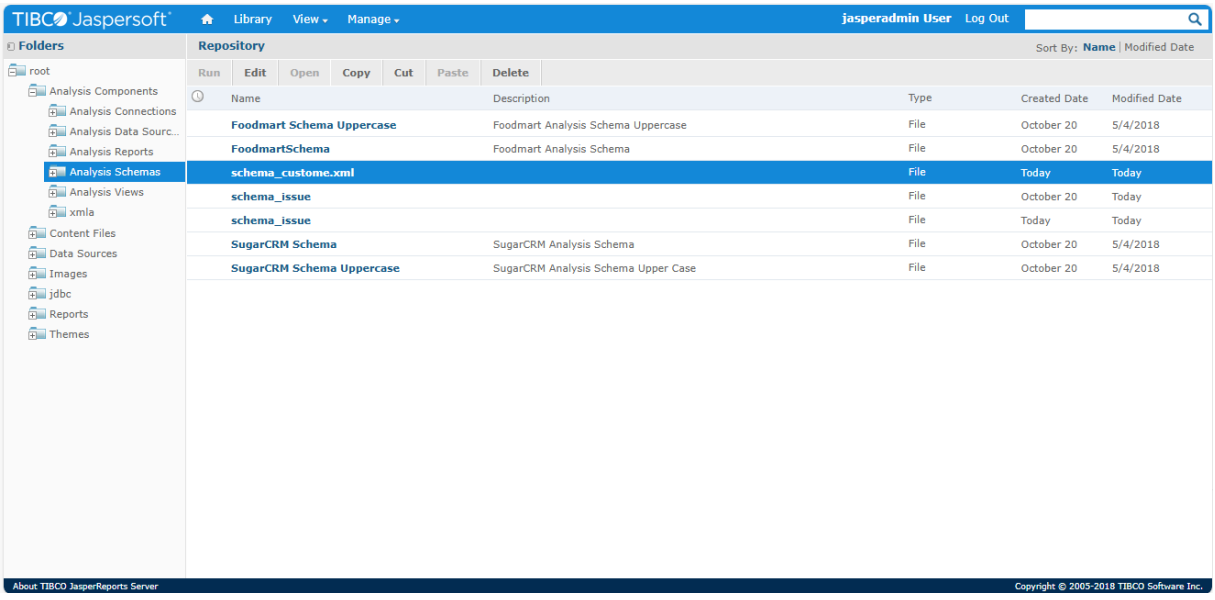
Le code source se trouve en lien ou en fichier zip au mail qui a été envoyé.

Et le schéma créé est :

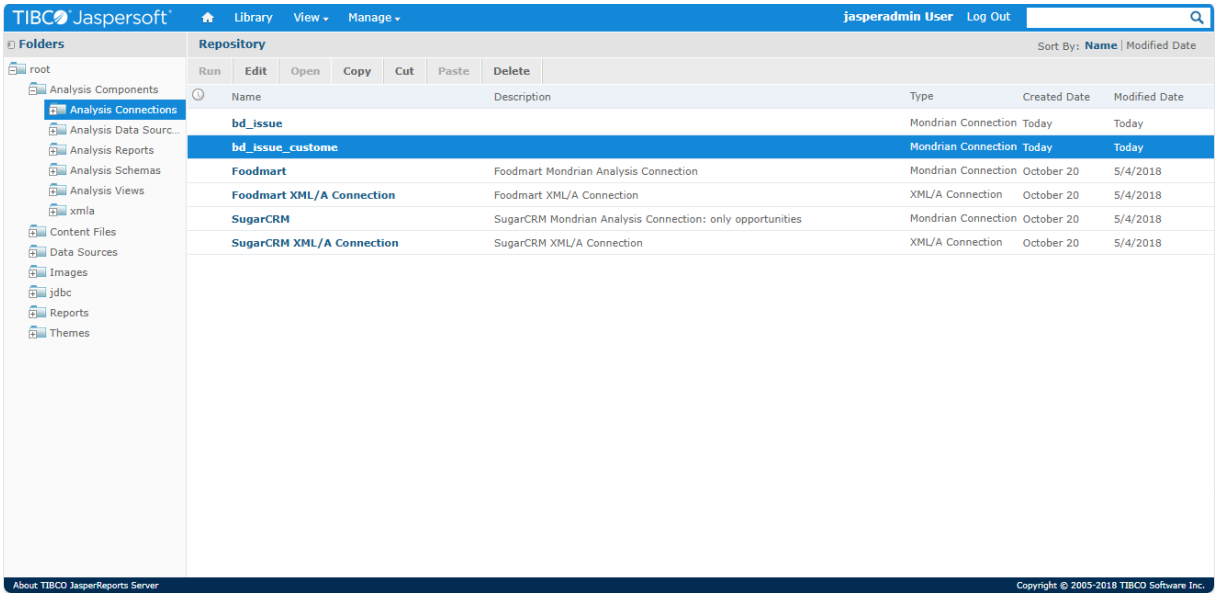


Dans la suite nous allons utiliser le serveur Mondrian pour essayer de présenter quelques vues.

Nous allons ajouter le schéma généré, puisque nous possédions déjà une connexion sur la base de données bd_issue dans laquelle nous avons ajouté la table fact_custome.



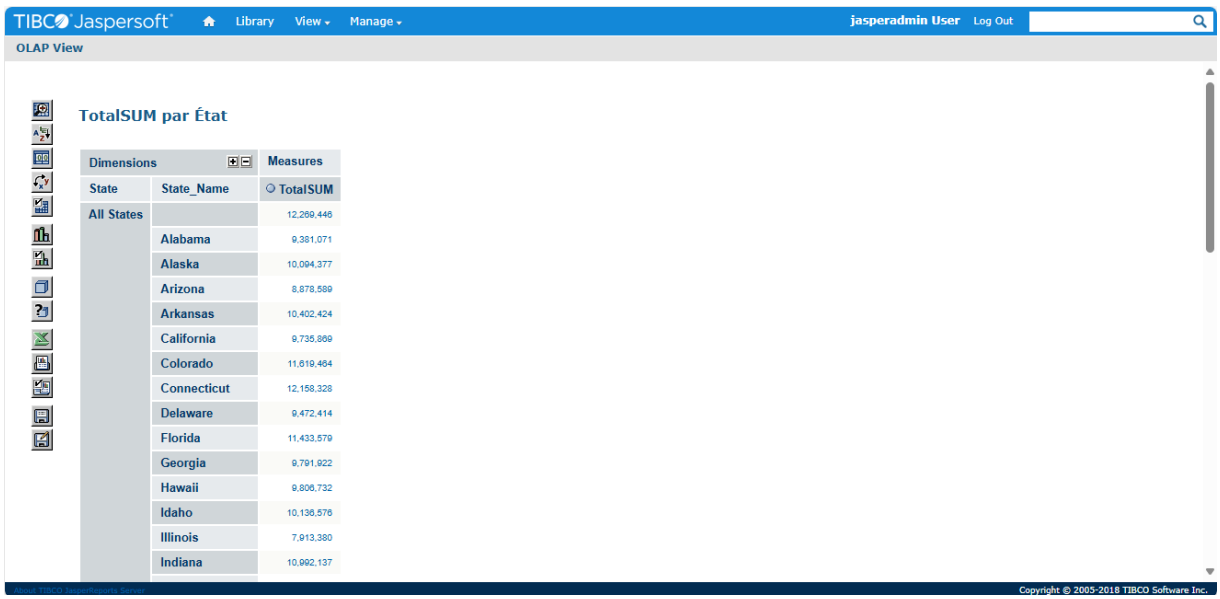
Nous allons créé une nouvelle connexion Mondrian, qui est illustrée par cette image (voir page suivante).



Maintenant qu'une nouvelle connexion est créé nous allons présenter quelques vues OLAP

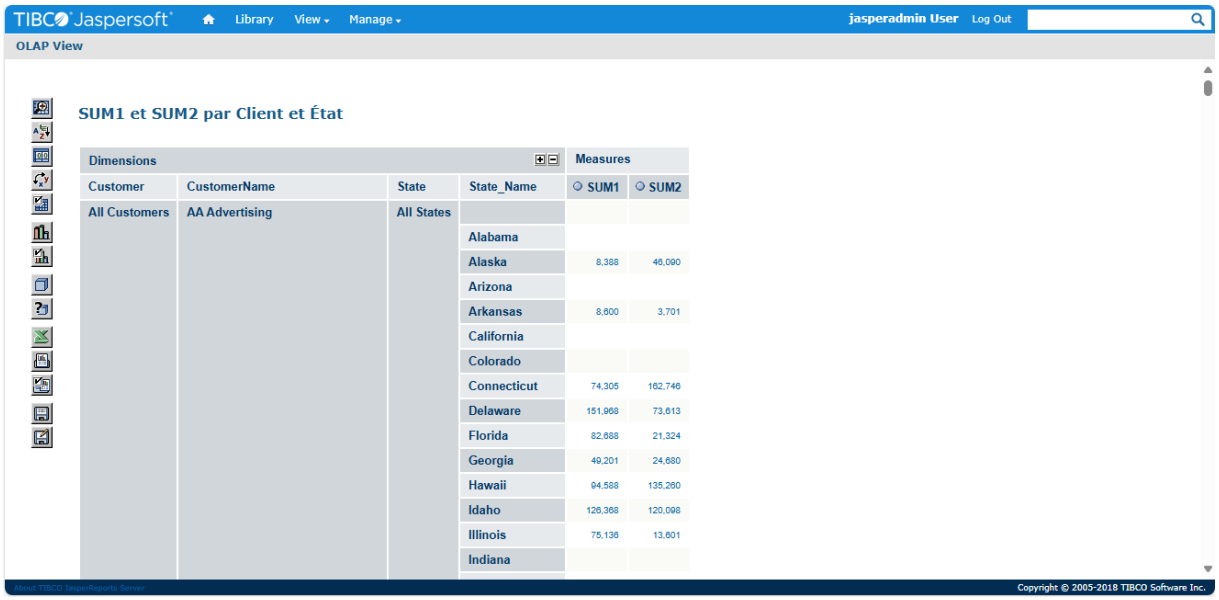
1. **Chiffre total par état pour voir quels états génèrent le plus de valeur**

```
SELECT
  {[Measures].[TotalSUM]} ON COLUMNS,
  [State].[State_Name].Members ON ROWS
FROM [CustomerFact]
```



2. **SUM1 et SUM2 par Client et État**

```
SELECT
  {[Measures].[SUM1], [Measures].[SUM2]} ON COLUMNS,
  CROSSJOIN([Customer].[CustomerName].Members, [State].[State_Name].Members)
ON ROWS
FROM [CustomerFact]
```



Conclusion :

En conclusion, ce TP a permis de mettre en œuvre un processus complet d'**ETL**, en extrayant les données depuis plusieurs sources, en les transformant (calcul de TotalSUM et AverageSUM) et en les chargeant dans la table de faits fact_customer de MySQL. Nous avons ensuite construit un **cube OLAP Mondrian**, avec des dimensions pertinentes (Customer, State, DateInscription, id2) et des mesures clés, pour faciliter l'analyse multidimensionnelle. Des requêtes **MDX** ont illustré différents axes de décision, comme l'analyse des totaux par état, des moyennes par date d'inscription ou encore la répartition des mesures par client et état. L'intégration avec **JasperReport** a permis de visualiser ces analyses sous forme de rapports clairs et interactifs. Ce TP a ainsi démontré comment combiner **intégration de données, modélisation OLAP et reporting** pour obtenir des informations exploitables. En résumé, il fournit une base solide pour la **prise de décision stratégique** à partir des données clients.