

République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la
Recherche Scientifique
Université Benyoucef Benkhedda Alger 1



Faculté de Sciences

Département Informatique

Master 1 Ingénierie des systèmes d'information intelligents ISII

Rapport de Projet de TP

Module : Entrepôt de données

Enseignante : Mme Menai

Sujet

Analyse des tendances de visionnage de contenus audiovisuels.

Réalisé par :

- BELABBAS Rania

Année universitaire : 2024/2025

Table des matières

| | |
|---|-----------|
| Introduction générale | 8 |
| 1 Modélisation Multidimensionnelle | 9 |
| 1.1 Introduction | 9 |
| 1.2 Présentation du thème | 9 |
| 1.3 Identification des Faits | 9 |
| 1.4 Identification des Dimensions | 10 |
| 1.5 Création du Schéma en Étoile | 11 |
| 1.6 Définition des Métadonnées | 11 |
| 1.7 Conclusion | 12 |
| 2 Intégration des Données (Processus ETL) | 13 |
| 2.1 Introduction | 13 |
| 2.2 Extraction des données | 14 |
| 2.3 Transformations | 21 |
| 2.4 Chargement des données transformées | 24 |
| 2.5 Installation du code SQL de la base de données PostgreSQL | 28 |
| 2.6 Conclusion | 29 |
| 3 Création du cube | 30 |
| 3.1 Introduction | 30 |
| 3.2 Création du cube OLAP | 30 |
| 3.2.1 Définition d'un cube OLAP | 30 |
| 3.2.2 Installation de Pentaho Schema Workbench | 30 |
| 3.2.3 Connexion à la base de données | 30 |
| 3.2.4 Création du cube | 31 |
| 3.2.5 Test du cube avec des requêtes MDX | 35 |
| 3.2.6 Enregistrement du schéma OLAP | 37 |
| 3.3 Conclusion | 38 |
| 4 Analyse et Visualisation des Résultats | 39 |
| 4.1 Introduction | 39 |
| 4.2 Connexion à la base de données MySQL | 39 |
| 4.2.1 Étapes de connexion | 40 |
| 4.3 Création des graphiques de visualisation | 43 |
| 4.3.1 Création des mesures | 43 |
| 4.3.2 Analyse démographique | 44 |
| 4.3.3 Analyse par plateforme | 47 |
| 4.3.4 Tendances temporelles | 48 |

TABLE DES MATIÈRES

| | | |
|-------|--------------------------------------|-----------|
| 4.3.5 | Analyse du contenu | 49 |
| 4.3.6 | Tableau de bord global | 51 |
| 4.4 | Conclusion | 51 |
| | Conclusion générale | 52 |

Table des figures

| | | |
|------|---|----|
| 1.1 | Schéma en étoile du modèle de visionnage | 11 |
| 2.1 | Logo de Talend. | 13 |
| 2.2 | Logo de PostgreSQL. | 14 |
| 2.3 | Source de données | 14 |
| 2.4 | Création d'un nouveau Job dans Talend | 15 |
| 2.5 | Installation ou activation des composants nécessaires | 15 |
| 2.6 | Création métadonnée : fichier excel | 16 |
| 2.7 | Nommer la métadonnée | 16 |
| 2.8 | Selectionner sheet | 17 |
| 2.9 | Configuration de métadonnée : en-têtes | 17 |
| 2.10 | Configuration des métadonnées : types et clés | 18 |
| 2.11 | Métadonnée créée | 18 |
| 2.12 | Composant <code>tFileInputExcel</code> préconfiguré | 18 |
| 2.13 | Glisser-déposer de la métadonnée dans le Job | 19 |
| 2.14 | Connexion de <code>tFileInputExcel</code> à <code>tLogRow</code> | 19 |
| 2.15 | Configuration du <code>tLogRow</code> en mode <i>Table</i> | 19 |
| 2.16 | Exécution du Job dans Talend | 20 |
| 2.17 | Affichage des données extraites dans la console | 20 |
| 2.18 | Ajout du composant <code>tUniqRow</code> pour supprimer les doublons | 21 |
| 2.19 | Configuration <code>tUniqRow</code> | 21 |
| 2.20 | Connexion de <code>tUniqRow</code> à <code>tLogRow</code> pour vérification | 22 |
| 2.21 | Connexion de <code>tUniqRow</code> à <code>tLogRow</code> pour vérification | 22 |
| 2.22 | Résultat après suppression des doublons | 22 |
| 2.23 | Lignes supprimées | 23 |
| 2.24 | Création de la base PostgreSQL dans pgAdmin | 24 |
| 2.25 | Connexion PostgreSQL dans Talend | 24 |
| 2.26 | Vérifier Connexion PostgreSQL dans Talend | 25 |
| 2.27 | Configuration Table de fait | 25 |
| 2.28 | Configuration Tables de Dimensions | 25 |
| 2.29 | Configuration Table de fait | 26 |
| 2.30 | Configuration Tables de Dimensions | 26 |
| 2.31 | Job complet de chargement dans PostgreSQL | 27 |
| 2.32 | Tables chargées dans PostgreSQL visibles via pgAdmin | 27 |
| 2.33 | Ajout de PostgreSQL au PATH dans Windows | 28 |
| 2.34 | Ajout de PostgreSQL au PATH dans Windows | 28 |
| 3.1 | Connexion réussi | 31 |
| 3.2 | Schéma | 31 |
| 3.3 | Nom du schéma | 31 |

| | | |
|------|---|----|
| 3.4 | Création du cube Visonnage_Cube | 31 |
| 3.5 | Création de la table de fait | 32 |
| 3.6 | Ajout des mesures | 32 |
| 3.7 | Ajout de la mesure Durée de visionnage | 32 |
| 3.8 | Ajout de la mesure Note | 32 |
| 3.9 | Création de la table de dimension Utilisateur | 33 |
| 3.10 | Création de la table de dimension Utilisateur | 33 |
| 3.12 | Structure de la table Utilisateur | 33 |
| 3.13 | Structure de la table Contenu Audiovisuel | 34 |
| 3.14 | Structure de la table Plateforme | 35 |
| 3.15 | Structure de la table Date | 35 |
| 3.16 | MDX Query | 35 |
| 3.17 | Requête MDX (1) et son résultat | 36 |
| 3.18 | Requête MDX (2) et son résultat | 36 |
| 3.19 | Requête MDX (3) et son résultat | 37 |
| 3.20 | Le fichier XML | 37 |
| 4.1 | Logo de Power Bi. | 39 |
| 4.2 | Obtention des données | 40 |
| 4.3 | Insertion des infos de connexion | 40 |
| 4.4 | Sélection des dimensions | 41 |
| 4.5 | Sélection des dimensions | 41 |
| 4.6 | Chargement et importation des données | 42 |
| 4.7 | Chargement et importation des données | 42 |
| 4.8 | Ajout d'une mesure personnalisée dans Power BI | 43 |
| 4.9 | Résultat des mesures | 43 |
| 4.10 | Ajout d'une colonne personnalisée dans Power BI | 44 |
| 4.11 | Visionnages par tranche d'âge | 45 |
| 4.12 | Visionnages par sexe | 45 |
| 4.13 | Visionnages par pays d'origine | 46 |
| 4.14 | Localisation des utilisateurs par pays | 46 |
| 4.15 | Préférences de contenu selon les tranches d'âge | 47 |
| 4.16 | Visionnages par plateforme | 47 |
| 4.17 | Visionnages par plateforme au fil des années | 48 |
| 4.18 | Visionnages selon les saisons | 48 |
| 4.19 | Répartition par type de contenu | 49 |
| 4.20 | Contenus les mieux notés (note ≥ 4) | 50 |
| 4.21 | Répartition par genre | 50 |
| 4.22 | Tableau de bord général | 51 |

Liste des tableaux

| | | |
|-----|--|----|
| 1.1 | Structure de la table des faits Visionnage | 11 |
| 1.2 | Structure de la dimension Utilisateur | 12 |
| 1.3 | Structure de la dimension Contenu_Audiovisuel | 12 |
| 1.4 | Structure de la dimension Plateforme | 12 |
| 1.5 | Structure de la dimension Date | 12 |

Listings

| | | |
|-----|--|----|
| 2.1 | Chemin de PostgreSQL | 28 |
| 2.2 | Commande pour vérifier la version de PostgreSQL | 29 |
| 2.3 | Commande pour exécuter le script SQL dans PostgreSQL | 29 |
| 4.1 | DAX code to count number of users | 43 |
| 4.2 | DAX code to calculate average age | 43 |
| 4.3 | DAX code for age group classification | 44 |
| 4.4 | DAX code for continent classification | 44 |

Introduction générale

les plateformes de streaming accumulent un volume massif de données utilisateurs, il devient essentiel de structurer ces informations pour en extraire des indicateurs pertinents. L'objectif principal de ce projet est **la mise en œuvre d'un système d'aide à la décision** reposant sur l'analyse des données de visionnage de contenus audiovisuels.

Ce travail s'articule autour des différentes étapes du processus **ETL (Extraction, Transformation, Chargement)** en s'appuyant sur des outils spécialisés tels que **Talend**.

Le projet se divise en plusieurs étapes principales :

- **Analyse des besoins et modélisation multidimensionnelle** : structurer les données sous forme de tables de faits et de dimensions, représentées dans un **schéma en étoile**.
- **Intégration des données** : extraire, transformer et charger les données brutes dans l'entrepôt, en garantissant leur qualité et leur cohérence.
- **Analyse OLAP** : effectuer des analyses interactives, générer des rapports et des visualisations répondant aux problématiques métiers identifiées.

Ce projet illustre ainsi l'ensemble du processus décisionnel, depuis l'intégration des données jusqu'à leur valorisation analytique, en passant par la modélisation, l'ETL et l'analyse en ligne.

Chapitre 1

Modélisation Multidimensionnelle

1.1 Introduction

La réussite de toute étude dépend de la qualité de son départ. De ce fait, ce chapitre est consacré à l'analyse des besoins et à la modélisation multidimensionnelle.

La modélisation multidimensionnelle consiste à considérer un sujet analysé comme un point dans un espace à plusieurs dimensions. Les données sont organisées de manière à mettre en évidence le sujet analysé et les différentes perspectives de l'analyse.

Dans ce chapitre, nous allons tout d'abord présenter le thème, identifier les faits et dimensions pertinents, créer le schéma en étoile, et définir les métadonnées associées.

1.2 Présentation du thème

Le thème de ce projet s'intitule : **Analyse des tendances de visionnage de contenus audiovisuels**.

Il vise à étudier les habitudes de consommation de séries, films, animes, etc., en fonction de critères tels que les plateformes utilisées, le profil des spectateurs et les préférences de contenu, afin de générer des conclusions pertinentes utiles aux problématiques métier.

1.3 Identification des Faits

La table des faits : nommée **Visionnage**, est le sujet à analyser. Elle contient :

- **Les mesures** : données quantitatives stockées dans la table des faits. Elles permettent d'effectuer des analyses et des agrégations.
 - **note** : note attribuée par l'utilisateur (de 1 à 5), utile pour évaluer la satisfaction.
 - **duree_visionnage** : durée du visionnage en heures, utile pour mesurer l'engagement.
 - **Les clés étrangères** : sont des champs qui établissent des liens avec les clés primaires des tables de dimensions.
- ❖ **Table de fait Visionnage** :
- `id_visionnage` (Clé primaire)
 - `utilisateur_id` (Clé étrangère)

- contenu_id (Clé étrangère)
- plateforme_id (Clé étrangère)
- date_id (Clé étrangère)
- note (Mesure)
- duree_visionnage (Mesure)

1.4 Identification des Dimensions

Une table de dimension représente les axes d'analyse d'un entrepôt de données. Elle stocke des attributs utilisés pour filtrer et analyser les données de la table des faits.

Dimensions associées à la table des faits :

❖ **Dimension Utilisateur :**

- utilisateur_id (Clé primaire)
- Nom
- Âge
- Sexe
- Pays

❖ **Dimension Contenu Audiovisuel :**

- contenu_id (Clé primaire)
- Titre
- Type (Film, Série, Anime, etc.)
- Genre
- Date_sortie
- Langue

❖ **Dimension Plateforme :**

- plateforme_id (Clé primaire)
- Nom de la plateforme

❖ **Dimension Date :**

- date_id (Clé primaire)
- Jour
- Mois
- Année
- Saison (Hiver, Printemps, Été, Automne)

1.5 Création du Schéma en Étoile

Un schéma en étoile est un modèle de données multidimensionnel utilisé dans les entrepôts de données. Il permet d'organiser les données afin de faciliter leur compréhension et leur analyse.

Il est composé d'une table de faits centrale entourée de plusieurs tables de dimensions.

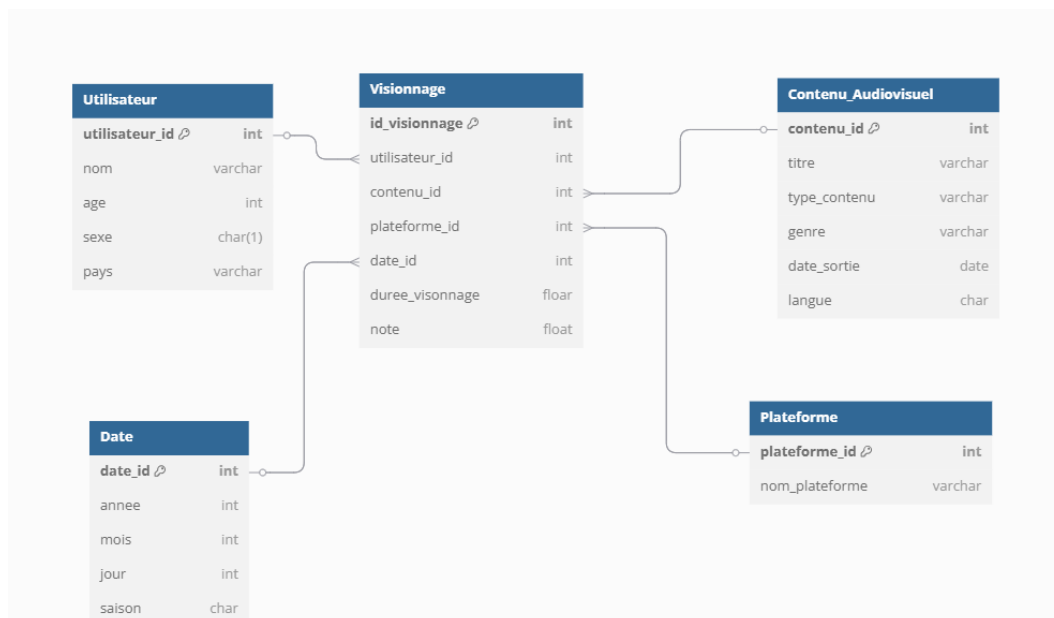


FIG. 1.1 : Schéma en étoile du modèle de visionnage

1.6 Définition des Métadonnées

Les métadonnées sont les données qui servent de répertoire pour d'autres données. Il aide les utilisateurs à comprendre les données à un niveau supérieur.

Table : Visionnage

| Champ | Type | Description |
|------------------|-------|--|
| id_visionnage | INT | Identifiant unique du fait de visionnage |
| utilisateur_id | INT | Référence vers la table Utilisateur |
| contenu_id | INT | Référence vers la table Contenu_Audiovisuel |
| plateforme_id | INT | Référence vers la table Plateforme |
| date_id | INT | Référence vers la table Date |
| duree_visionnage | FLOAT | Durée du visionnage du contenu en heures |
| note | FLOAT | Note attribuée par l'utilisateur (1 à 5) |

TAB. 1.1 : Structure de la table des faits **Visionnage**

Table : Utilisateur

| Champ | Type | Description |
|----------------|--------|-------------------------------------|
| utilisateur_id | INT | Identifiant unique de l'utilisateur |
| nom | STRING | Nom ou pseudonyme |
| age | INT | Âge de l'utilisateur |
| sexe | STRING | Sexe (Masculin, Féminin, Autre) |
| pays | STRING | Pays de résidence |

TAB. 1.2 : Structure de la dimension **Utilisateur****Table : Contenu_Audiovisuel**

| Champ | Type | Description |
|--------------|--------|---|
| contenu_id | INT | Identifiant unique du contenu |
| titre | STRING | Titre du contenu |
| type_contenu | STRING | Type (Film, Série, Anime, etc.) |
| genre | STRING | Genre principal (Action, Comédie, etc.) |
| date_sortie | DATE | Date de sortie |
| langue | STRING | Langue principale |

TAB. 1.3 : Structure de la dimension **Contenu_Audiovisuel****Table : Plateforme**

| Champ | Type | Description |
|----------------|--------|--|
| plateforme_id | INT | Identifiant unique de la plateforme |
| nom_plateforme | STRING | Nom de la plateforme (ex : Netflix, Prime Video) |

TAB. 1.4 : Structure de la dimension **Plateforme****Table : Date**

| Champ | Type | Description |
|---------|--------|---|
| date_id | INT | Identifiant unique de la date |
| jour | STRING | Jour de la semaine |
| mois | INT | Mois (numérique) |
| annee | INT | Année |
| saison | STRING | Saison (Hiver, Printemps, Été, Automne) |

TAB. 1.5 : Structure de la dimension **Date**

1.7 Conclusion

Dans ce chapitre, nous avons présenté la table de fait, les tables de dimensions et les métadonnées ainsi que le schéma en étoile structurant notre entrepôt de données. Cette modélisation constitue une base solide et cohérente pour les chapitres suivants.

Chapitre 2

Intégration des Données (Processus ETL)

2.1 Introduction

Le processus d'intégration des données, couramment appelé ETL (Extract, Transform, Load), est une étape cruciale dans la gestion des données. Il permet de collecter des données provenant de sources diverses, de les transformer pour les rendre cohérentes et exploitables, puis de les charger dans une base de données cible ou un entrepôt de données (data warehouse).

Le processus ETL se compose de trois étapes principales :

- **Extraction** : récupération des données depuis différentes sources (fichiers plats, bases de données, API, etc.).
- **Transformation** : nettoyage, enrichissement, formatage, et consolidation des données extraites afin de garantir leur qualité et cohérence.
- **Chargement** : insertion des données transformées dans la base cible, généralement un entrepôt de données, pour permettre l'analyse et le reporting.

Dans ce projet, nous utiliserons l'outil **Talend**, une plateforme d'intégration de données open source, pour automatiser et orchestrer ce processus ETL. Talend offre une interface graphique intuitive qui facilite la conception des flux d'extraction, transformation et chargement des données.



FIG. 2.1 : Logo de Talend.

Pour la phase de chargement, la base de données cible choisie est **PostgreSQL**, un système de gestion de base de données relationnelle robuste et performant, bien adapté à la gestion des entrepôts de données.



FIG. 2.2 : Logo de PostgreSQL.

2.2 Extraction des données

Pour extraire les données à partir d'un fichier Excel, nous avons suivi les étapes suivantes dans l'outil d'intégration Talend :

1. **Fichier de données initiales** : Les données initiales sont dans un fichier Excel contenant 113 lignes. Ces données sont organisées en plusieurs colonnes : Nom de l'utilisateur, age, sexe, pays, titre, type de contenu, genre, plateforme, date de sortie, langue, durée de visionnage par heure, note d'évaluation sur 5, date de visionnage, saison et les IDs.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
|----|----------------|-----------------|-----|------|----------|------------|-------------------|--------------|-----------------|------------|---------------|---------|-------------|----------|----------------|------|------|------|-------|-----------|
| 1 | utilisateur_id | nom_utilisateur | age | sexe | pays | contenu_id | titre | type_contenu | genre | plateforme | plateforme_id | date_id | date_sortie | langue | uree_visionnag | note | jour | mois | annee | saison |
| 2 | 1 | Nabil | 56 | F | France | 1 | Extraction | Film | Action | 1 | Netflix | 2E+07 | 2021-05-11 | Arabe | 1.42 | 1.8 | 11 | 5 | 2021 | Printemps |
| 3 | 2 | Walid | 33 | F | Sénégal | 2 | Baki | Anime | Action | 1 | Netflix | 2E+07 | 2010-01-27 | Arabe | 2.47 | 2.6 | 27 | 1 | 2010 | Hiver |
| 4 | 3 | Sarah | 60 | M | Sénégal | 3 | Demon Slayer | Anime | Fantastique | 1 | Netflix | 2E+07 | 2019-09-25 | Espagnol | 1.19 | 4.3 | 25 | 9 | 2019 | Automne |
| 5 | 4 | Lina | 46 | F | Sénégal | 4 | Aggretsuko | Anime | Comédie | 1 | Netflix | 2E+07 | 2020-02-17 | Anglais | 2.12 | 4.1 | 17 | 2 | 2020 | Hiver |
| 6 | 5 | Amine | 20 | F | Égypte | 5 | Don't Look Up | Film | Comédie | 1 | Netflix | 2E+07 | 2002-04-05 | Arabe | 1.53 | 2.1 | 5 | 4 | 2002 | Printemps |
| 7 | 6 | Reda | 64 | F | Maroc | 6 | Emily in Paris | Série | Comédie | 1 | Netflix | 2E+07 | 2019-03-20 | Japonais | 0.89 | 1.1 | 20 | 3 | 2019 | Printemps |
| 8 | 7 | Walid | 61 | M | Italie | 7 | Emily in Paris | Série | Comédie | 1 | Netflix | 2E+07 | 2010-12-24 | Espagnol | 0.82 | 1.9 | 24 | 12 | 2010 | Hiver |
| 9 | 8 | Anis | 60 | F | Maroc | 8 | Extraction | Film | Action | 1 | Netflix | 2E+07 | 2003-06-15 | Espagnol | 1.09 | 1.2 | 15 | 6 | 2003 | Été |
| 10 | 9 | Reda | 51 | M | Maroc | 9 | Reasons Why | Série | Drame | 1 | Netflix | 2E+07 | 2001-07-21 | Japonais | 0.77 | 4.9 | 21 | 7 | 2001 | Été |
| 11 | 10 | Nora | 46 | F | Algérie | 10 | Jujutsu Kaisen | Anime | Action | 2 | Crunchyroll | 2E+07 | 2019-12-19 | Anglais | 0.54 | 2.1 | 19 | 12 | 2019 | Hiver |
| 12 | 11 | Lina | 41 | M | Canada | 11 | Aggretsuko | Anime | Comédie | 1 | Netflix | 2E+07 | 2008-07-03 | Français | 1.01 | 2.5 | 3 | 7 | 2008 | Été |
| 13 | 12 | Lina | 26 | M | Belgique | 12 | One Piece | Anime | Aventure | 2 | Crunchyroll | 2E+07 | 2024-03-18 | Japonais | 1.3 | 4.1 | 18 | 3 | 2024 | Printemps |
| 14 | 13 | Karim | 63 | F | Belgique | 13 | Marriage Story | Film | Drame | 1 | Netflix | 2E+07 | 2023-12-22 | Anglais | 0.58 | 3.8 | 22 | 12 | 2023 | Hiver |
| 15 | 14 | Yasmine | 28 | F | Sénégal | 14 | The Witcher | Série | Fantastique | 1 | Netflix | 2E+07 | 2007-10-27 | Français | 1.63 | 2 | 27 | 10 | 2007 | Automne |
| 16 | 15 | Bilal | 34 | M | Égypte | 15 | Bird Box | Film | Thriller | 1 | Netflix | 2E+07 | 2022-04-11 | Arabe | 1.31 | 1.3 | 11 | 4 | 2022 | Printemps |
| 17 | 16 | Fatima | 11 | F | France | 16 | The Social Dilemm | Documentaire | Technologie | 1 | Netflix | 2E+07 | 2018-03-02 | Arabe | 1.07 | 3.5 | 2 | 3 | 2018 | Printemps |
| 18 | 17 | Lella | 31 | M | Italie | 17 | Game of Thrones | Série | Fantastique | 3 | Prime Video | 2E+07 | 2004-12-22 | Espagnol | 1.96 | 5 | 22 | 12 | 2004 | Hiver |
| 19 | 18 | Lella | 52 | M | Maroc | 18 | Demon Slayer | Anime | Fantastique | 1 | Netflix | 2E+07 | 2018-09-19 | Arabe | 2.23 | 1 | 19 | 9 | 2018 | Automne |
| 20 | 19 | Kenza | 37 | M | Italie | 19 | Money Heist | Série | Action | 1 | Netflix | 2E+07 | 2019-09-29 | Français | 1.49 | 1.3 | 29 | 9 | 2019 | Automne |
| 21 | 20 | Amine | 49 | M | Italie | 20 | Titanic | Film | Romance | 3 | Prime Video | 2E+07 | 2000-02-29 | Anglais | 0.57 | 3.4 | 29 | 2 | 2000 | Hiver |
| 22 | 21 | Nabil | 16 | M | Maroc | 21 | One Piece | Anime | Aventure | 2 | Crunchyroll | 2E+07 | 2011-04-13 | Anglais | 2.08 | 2.1 | 13 | 4 | 2011 | Printemps |
| 23 | 22 | Rania | 41 | M | Tunisie | 22 | Reasons Why | Série | Drame | 1 | Netflix | 2E+07 | 2004-07-08 | Japonais | 0.87 | 1.1 | 8 | 7 | 2004 | Été |
| 24 | 23 | Bilal | 60 | M | Canada | 23 | Parasite | Film | Thriller | 3 | Prime Video | 2E+07 | 2011-08-13 | Espagnol | 2.44 | 4.2 | 13 | 8 | 2011 | Été |
| 25 | 24 | Sofiane | 45 | M | France | 24 | Beastars | Anime | Drame | 1 | Netflix | 2E+07 | 2020-10-27 | Anglais | 0.71 | 5 | 27 | 10 | 2020 | Automne |
| 26 | 25 | Fatima | 59 | M | Belgique | 25 | Black Mirror | Série | Science-Fiction | 1 | Netflix | 2E+07 | 2015-08-27 | Anglais | 1.51 | 1.8 | 27 | 8 | 2015 | Été |
| 27 | 26 | Walid | 15 | M | France | 26 | Marriage Story | Film | Drame | 1 | Netflix | 2E+07 | 2008-07-16 | Français | 2.3 | 3.9 | 16 | 7 | 2008 | Été |
| 28 | 27 | Nora | 21 | M | France | 27 | Emily in Paris | Série | Comédie | 1 | Netflix | 2E+07 | 2012-04-30 | Français | 2.48 | 3.6 | 30 | 4 | 2012 | Printemps |

FIG. 2.3 : Source de données

2. **Création d'un Job Talend** : Nous avons commencé par créer un nouveau Job dans Talend afin d'organiser le processus d'extraction.

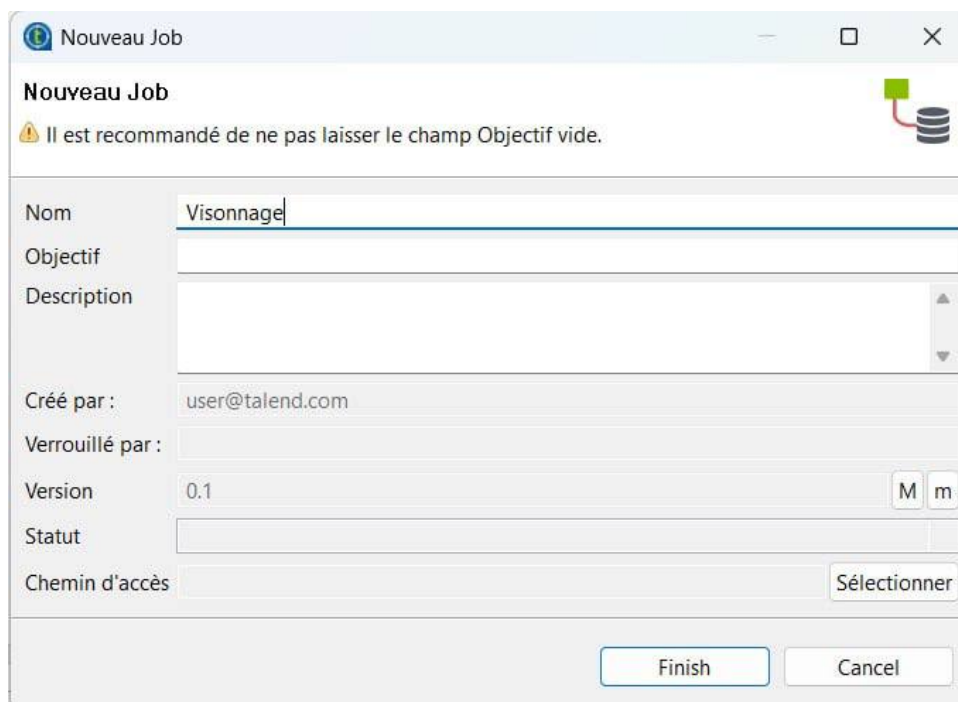


FIG. 2.4 : Création d'un nouveau Job dans Talend

3. **Installation des packages nécessaires** : Les composants requis pour manipuler les fichiers Excel (tels que `tFileInputExcel` et `tLogRow`) ont été installés ou activés dans le studio Talend.

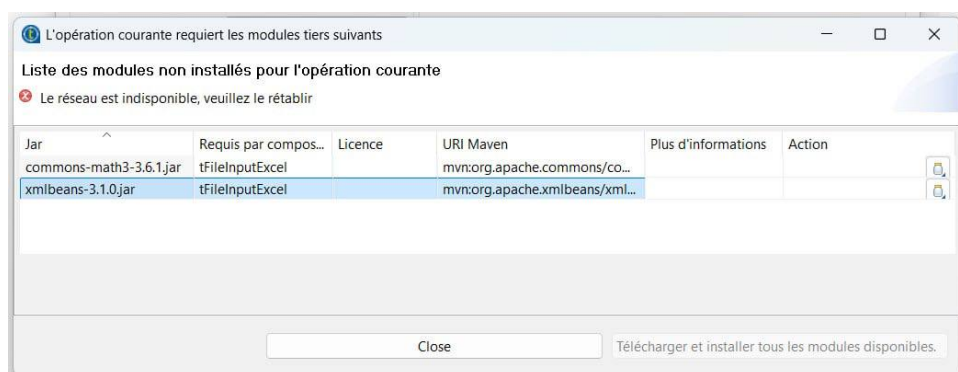


FIG. 2.5 : Installation ou activation des composants nécessaires

4. **Création de la métadonnée** : Nous avons configuré la métadonnée pour inclure la première ligne comme en-tête, défini les types de données des colonnes, et spécifié les clés primaires nécessaires pour les futures transformations.

-Dans Métadonnées -> clique droit dans Fichier Excel -> sélectionner "Créer un fichier Excel"

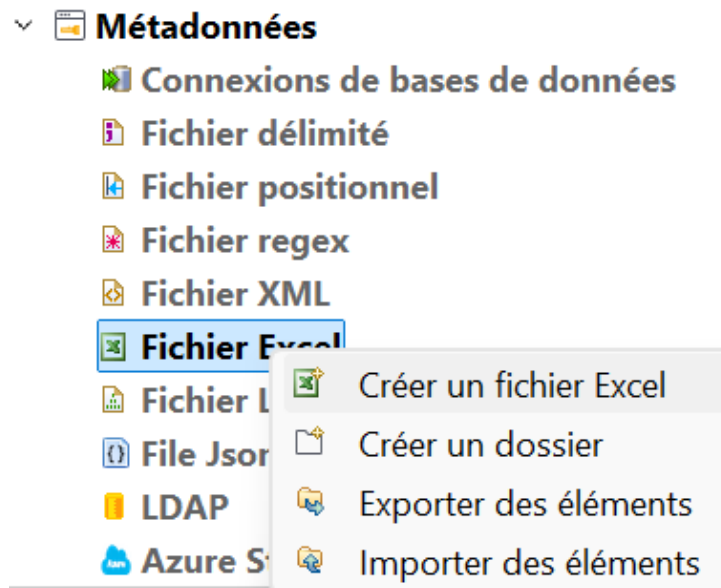


FIG. 2.6 : Création métadonnée : fichier excel

-Donner un nom

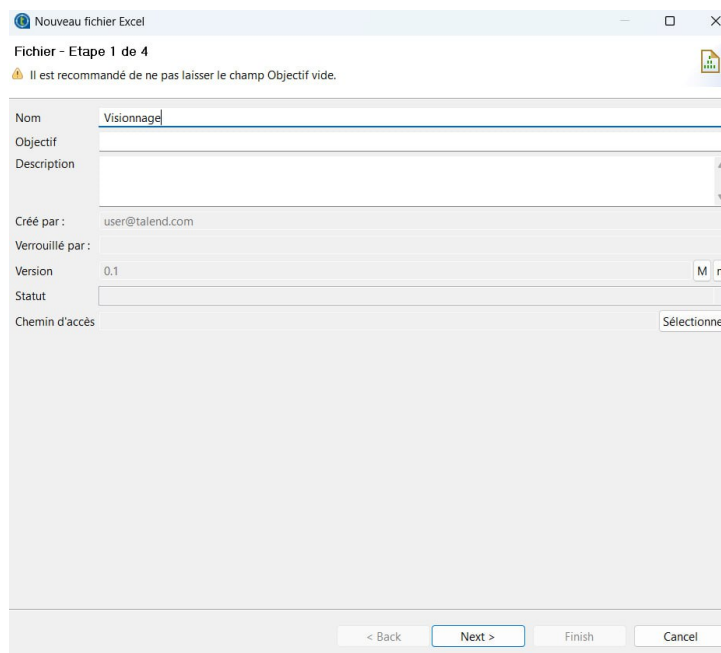


FIG. 2.7 : Nommer la métadonnée

-Sélectionner le sheet concerné

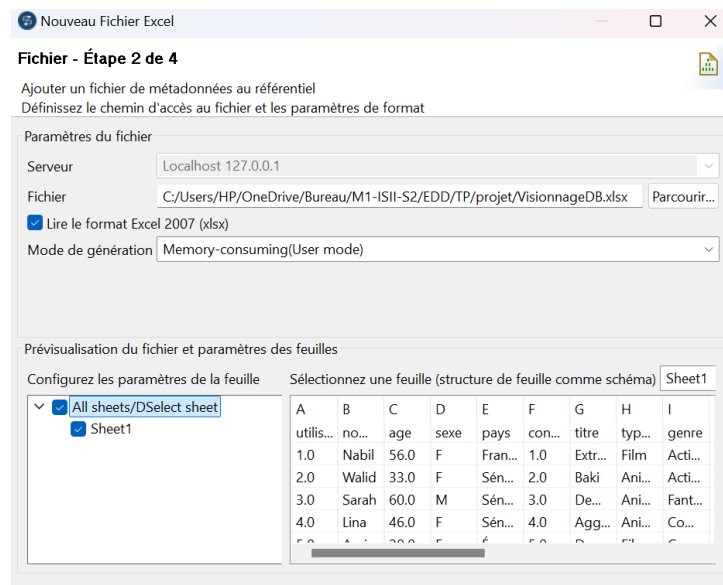


FIG. 2.8 : Sélectionner sheet

-Inclure l'en-tête et on a aussi un aperçu de la métadonnée

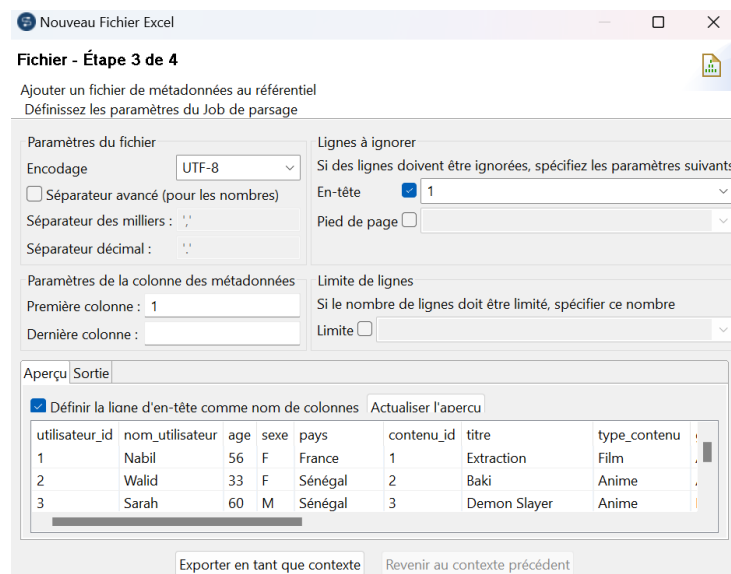


FIG. 2.9 : Configuration de métadonnée : en-têtes

-Configurer les types des colonnes et les clés


| Description du schéma | | | | |
|--|-------------------------------------|-----------|-------------------------------------|--|
| Colonne | Clé | Type | ☑ Nullable | |
|  utilisateur_id | <input checked="" type="checkbox"/> | int | <input type="checkbox"/> | |
| nom_utilisateur | <input type="checkbox"/> | String | <input checked="" type="checkbox"/> | |
| age | <input type="checkbox"/> | Integer | <input checked="" type="checkbox"/> | |
| sexe | <input type="checkbox"/> | Character | <input checked="" type="checkbox"/> | |

FIG. 2.10 : Configuration des métadonnées : types et clés

-Métadonnée créée avec success

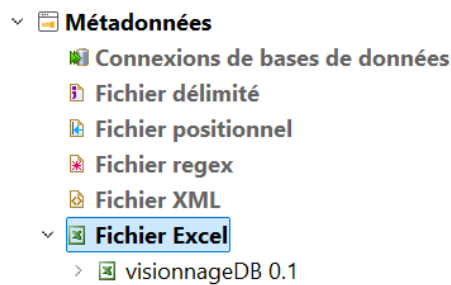


FIG. 2.11 : Métadonnée créée

5. **Glisser la métadonnée dans le Job** : La métadonnée définie précédemment a été glissée directement dans l'espace de travail du Job, ce qui a généré automatiquement un composant `tFileInputExcel` préconfiguré.

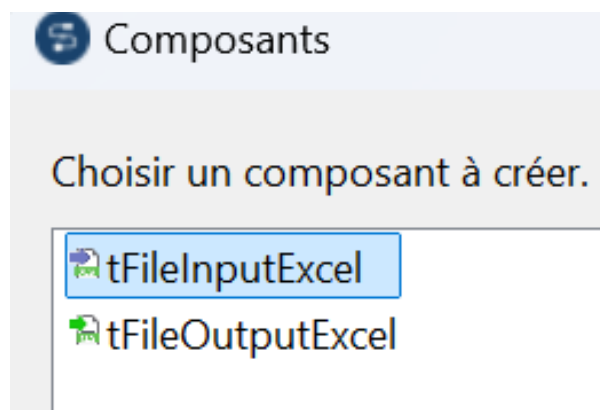


FIG. 2.12 : Composant `tFileInputExcel` préconfiguré

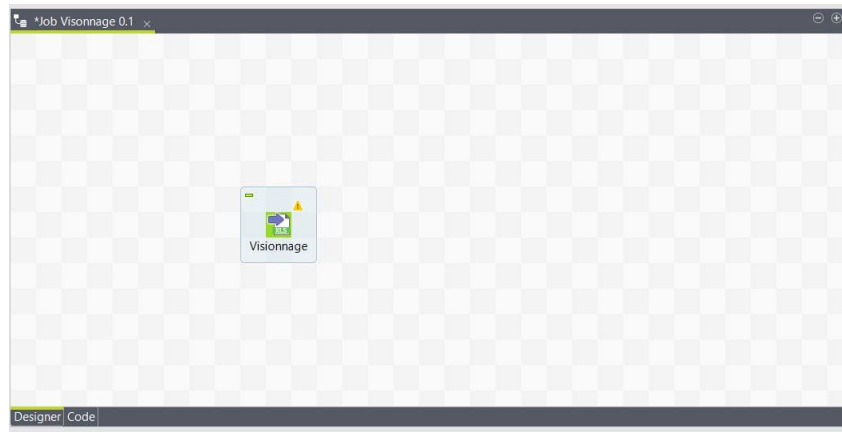


FIG. 2.13 : Glisser-déposer de la métadonnée dans le Job

6. **Connexion avec tLogRow** : Le composant de lecture a été relié à un tLogRow, permettant d’afficher le contenu du fichier en format tabulaire dans la console Talend.

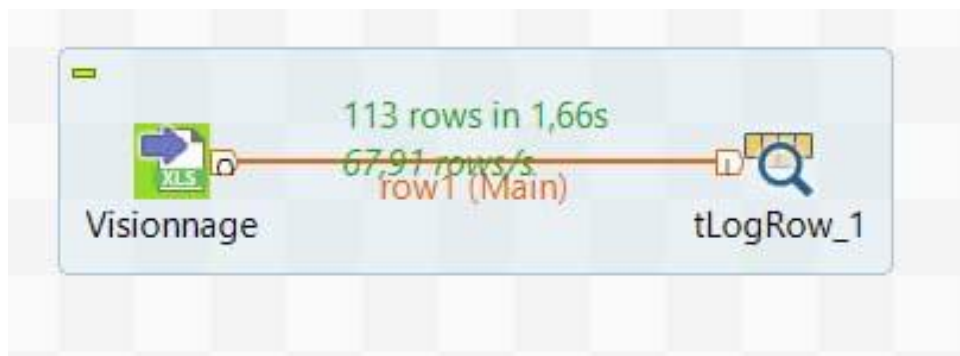


FIG. 2.14 : Connexion de tFileInputExcel à tLogRow

7. **Configurer tLogRow en mode tableau** : Le composant tLogRow a été configuré pour afficher les résultats en format "Table" afin d’améliorer la lisibilité des données en sortie.

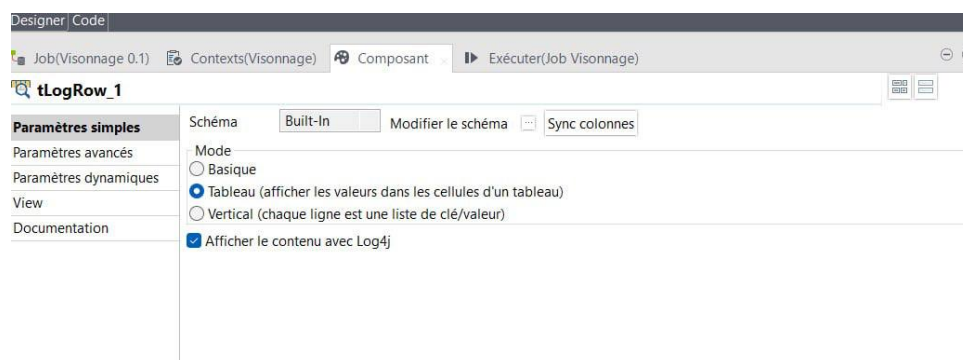


FIG. 2.15 : Configuration du tLogRow en mode *Table*

8. **Exécution du Job** : Le Job a été exécuté pour valider le bon fonctionnement du processus d'extraction.

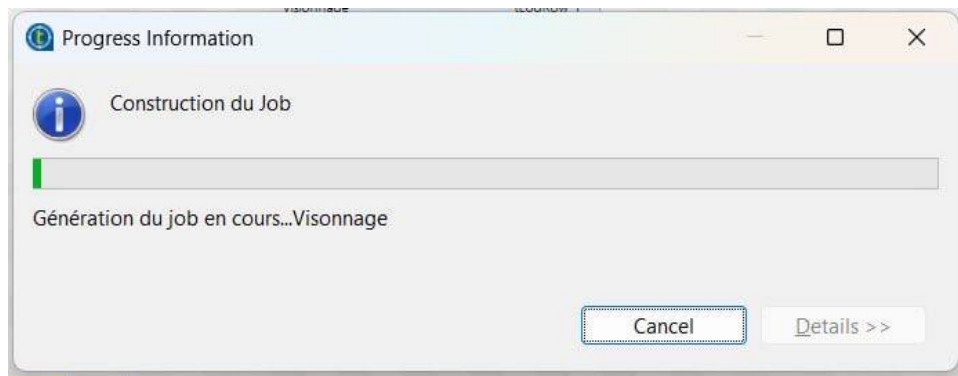


FIG. 2.16 : Exécution du Job dans Talend

9. **Observation des résultats** : Les données extraites ont été affichées correctement dans la console, confirmant que le fichier Excel a été lu avec succès.

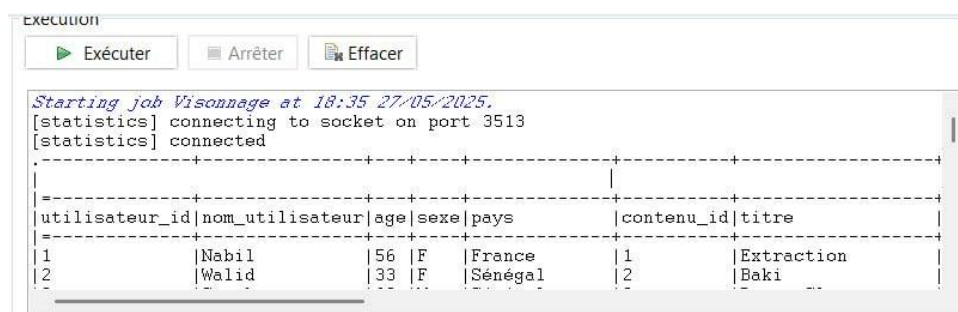


FIG. 2.17 : Affichage des données extraites dans la console

Cette étape constitue la base du pipeline ETL, permettant de préparer les données pour les étapes de transformation et de chargement.

2.3 Transformations

Après l'extraction des données, une phase de transformation est nécessaire pour nettoyer et structurer les données avant leur chargement. Dans Talend, nous avons appliqué les transformations suivantes :

1. **Ajout du composant tUniqRow** : Ce composant permet de supprimer les doublons à partir d'une ou plusieurs colonnes clés. Nous l'avons glissé dans le Job depuis la palette de composants.

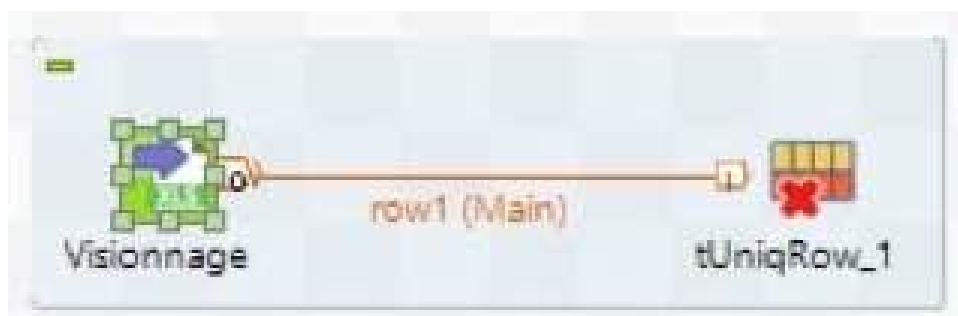


FIG. 2.18 : Ajout du composant tUniqRow pour supprimer les doublons

2. **Configuration du tUniqRow** : Nous avons défini les colonnes servant à l'identification des doublons, en particulier les identifiants uniques des entités (utilisateurs, contenus, etc.).

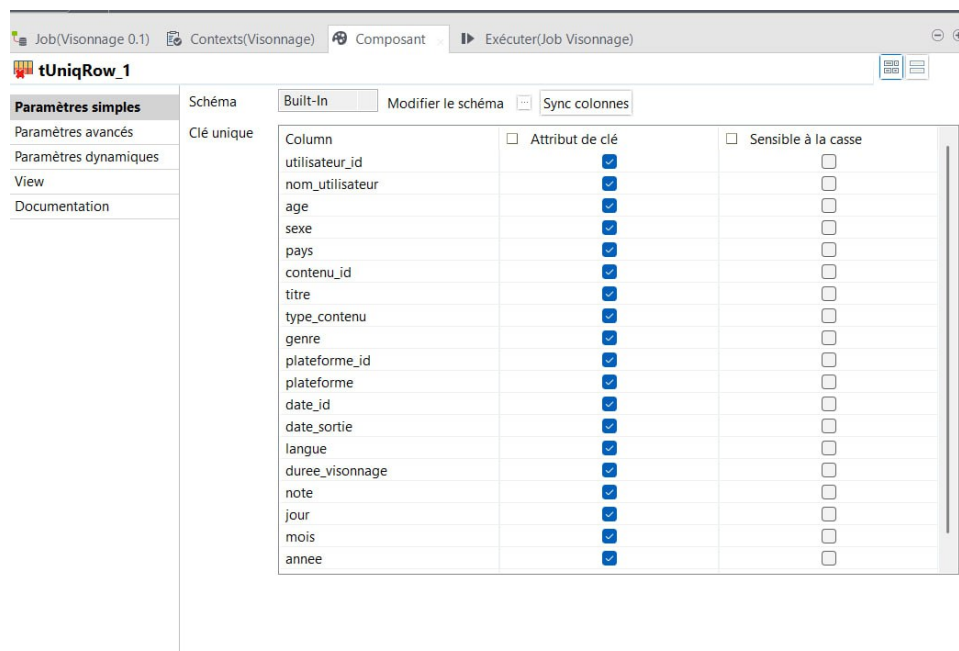


FIG. 2.19 : Configuration tUniqRow

3. **Connexion à tLogRow** : Pour vérifier le résultat de la suppression des doublons, nous avons connecté la sortie du tUniqRow à un tLogRow.

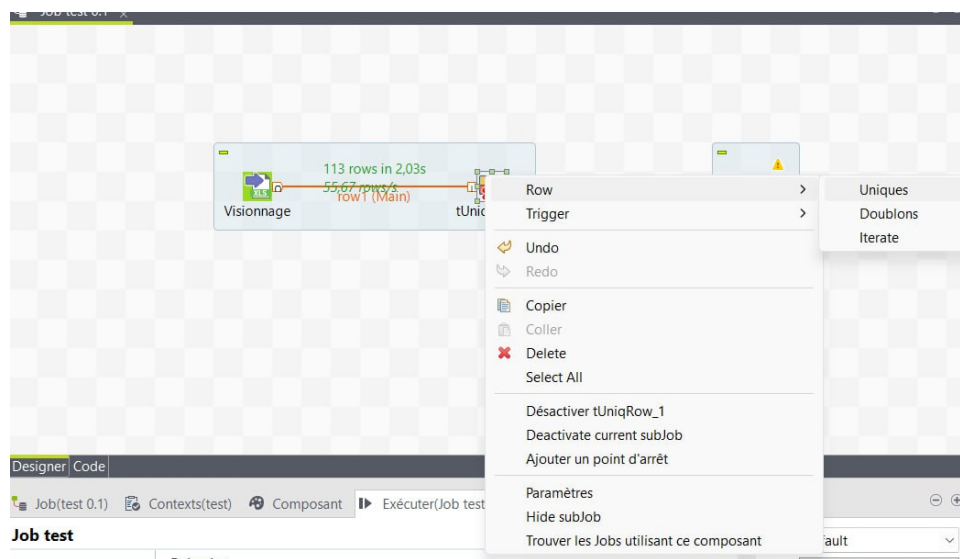


FIG. 2.20 : Connexion de tUniqRow à tLogRow pour vérification

4. **Affichage du résultat via tLogRow** : Nous avons connecté tUniqRow à tLogRow pour visualiser les lignes filtrées et confirmer le bon fonctionnement de la suppression des doublons. On reli tUniqRow à tLogRow avec la ligne "Uniques" -> 10 lignes ont été supprimées car elles sont identiques.

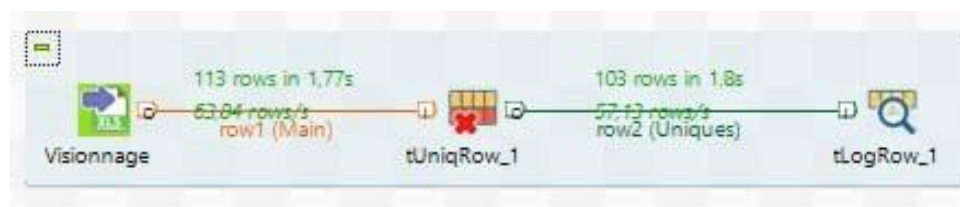


FIG. 2.21 : Connexion de tUniqRow à tLogRow pour vérification

Job test

Exécution simple

Exécution Debug

Paramètres avancés

Exéc distante

Exéc. test mémoire

Exécution

Exécuter Arrêter Effacer

| | | | | | |
|-----|---------|----|---|---------|------|
| 100 | Bilal | 59 | M | France | 100 |
| 101 | Yasmine | 28 | F | Maroc | 1001 |
| 102 | Omar | 35 | M | Tunisie | 1002 |
| 103 | Aiko | 22 | F | Japon | 1003 |

[statistics] disconnected

Job test ended at 22:57 29/05/2025. [exit code = 0]

☐ Nombre limite de lignes 100 ☐ Retour automatique à la ligne

FIG. 2.22 : Résultat après suppression des doublons

5. **Affichage des lignes supprimées via tLogRow** : Nous avons connecté tUniqRow à tLogRow pour visualiser les lignes supprimées. On reli tUniqRow à tLogRow avec la ligne "Doublons".

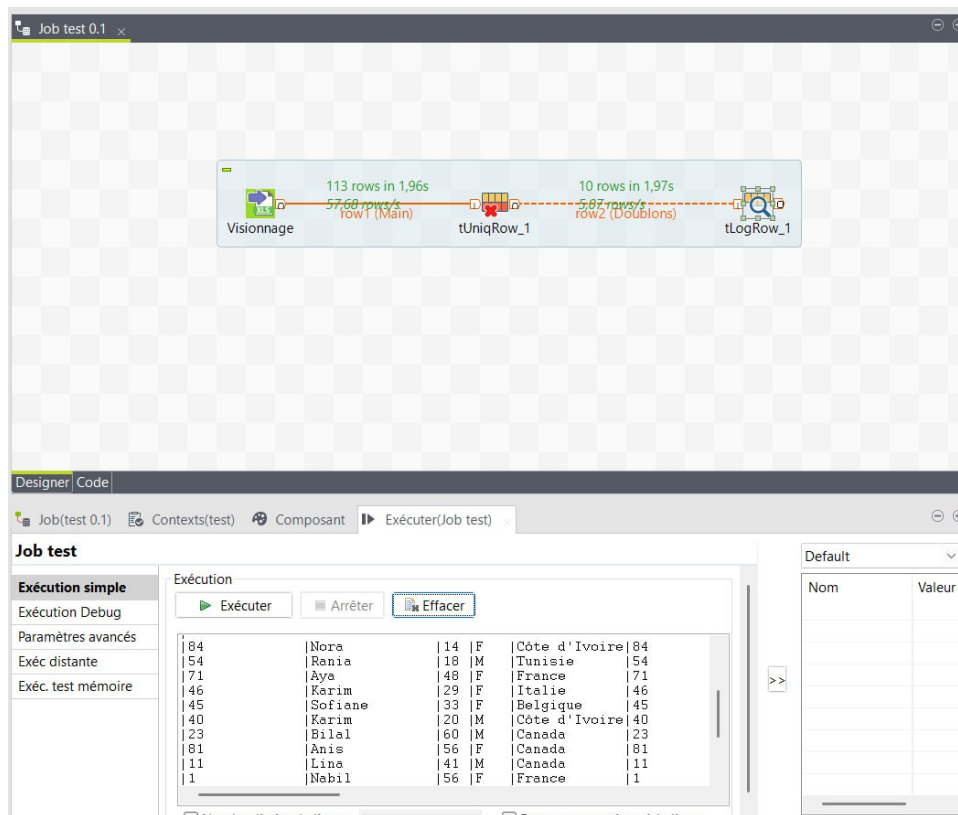


FIG. 2.23 : Lignes supprimées

Ces transformations assurent la qualité, la cohérence et la pertinence des données avant leur chargement final dans la base de données.

2.4 Chargement des données transformées

Après nettoyage et transformation, les données ont été chargées dans une base de données **PostgreSQL** via Talend. Voici les étapes réalisées :

1. **Installation de PostgreSQL** : Nous avons installé PostgreSQL localement afin de disposer d'un SGBD fiable pour accueillir notre entrepôt de données.
2. **Création d'une base vide** : Une base nommée `visionnagedb_etl` a été créée à l'aide de pgAdmin, interface graphique officielle de PostgreSQL.

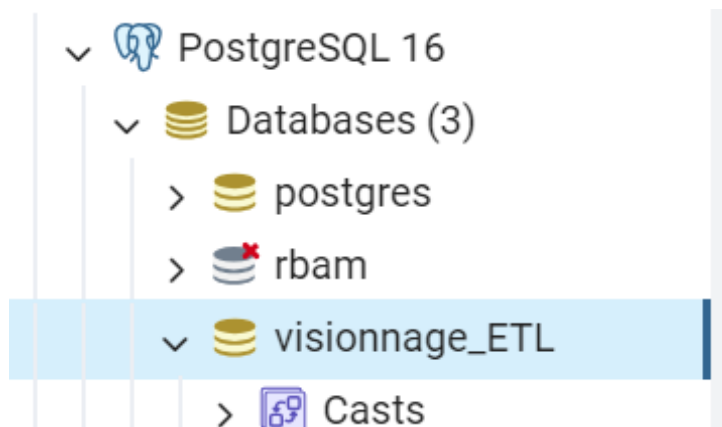


FIG. 2.24 : Création de la base PostgreSQL dans pgAdmin

3. **Connexion PostgreSQL dans Talend** : Depuis Talend, une connexion PostgreSQL a été définie via l'option *"Créer une connexion de base de données"*. Les paramètres renseignés étaient :

- Hôte : `localhost`
- Port : `5432`
- Base : `visionnagedb_etl`
- Utilisateur : `postgres`
- Mot de passe : `*****`

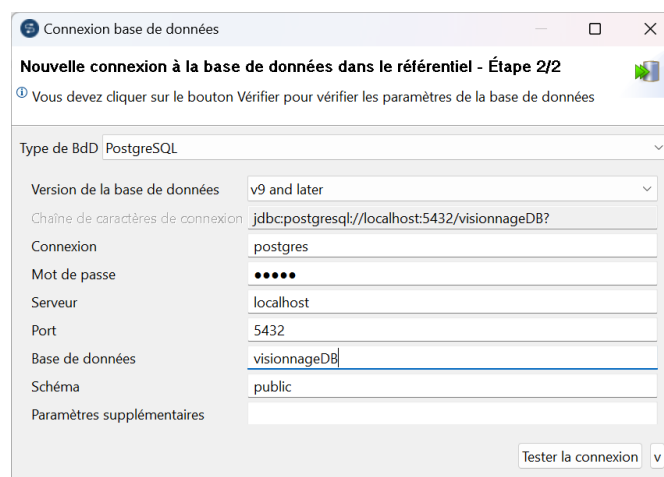


FIG. 2.25 : Connexion PostgreSQL dans Talend

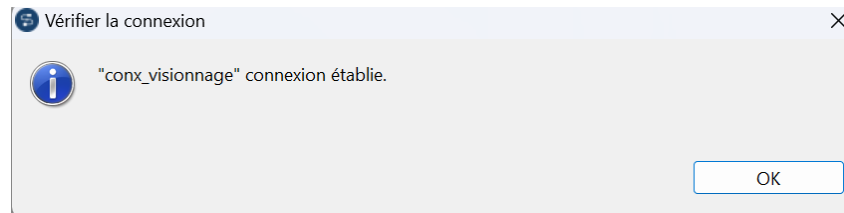


FIG. 2.26 : Vérifier Connexion PostgreSQL dans Talend

4. **Séparation des dimensions avec plusieurs tMap** : Pour chaque dimension (utilisateur, contenu, plateforme, date), un composant tMap a été utilisé pour extraire les colonnes pertinentes à partir du fichier source.
5. **Dé-duplication avec tUniqRow** : Chaque flux de dimension a été connecté à un composant tUniqRow pour garantir l'unicité des clés primaires avant insertion dans la base.

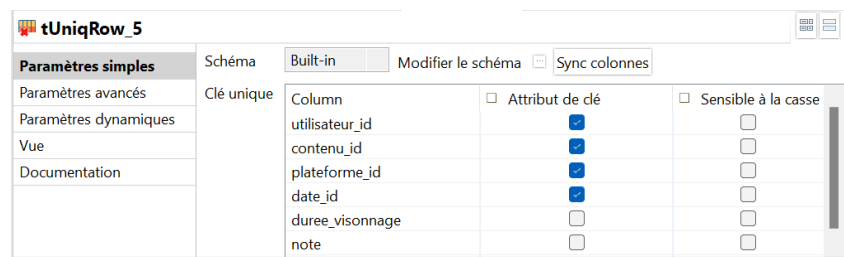
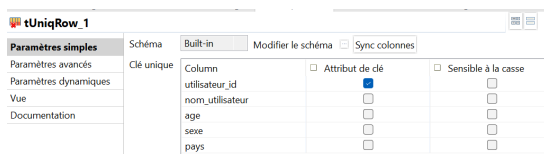
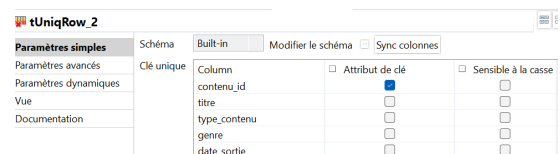


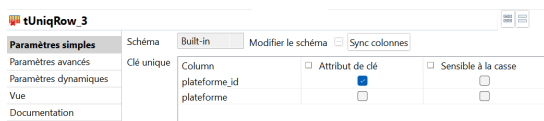
FIG. 2.27 : Configuration Table de fait



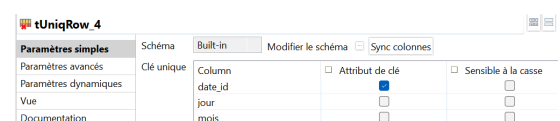
(a) Configuration Dimension Utilisateur



(b) Configuration Dimension Contenu



(c) Configuration Dimension Plateforme



(d) Configuration Dimension Date

FIG. 2.28 : Configuration Tables de Dimensions

6. **Chargement dans PostgreSQL via tPostgresqlOutput** : Chaque flux final (dimension ou fait) a été dirigé vers un composant tPostgresqlOutput. Les options suivantes ont été configurées :

- *Action sur la table* : Supprimer la table si elle existe et la recréer
- *Action sur les données* : Insérer

The screenshot shows the configuration for the tPostgresqlOutput component. The 'Database' is set to 'PostgreSQL' and 'Appliquer' is clicked. The 'Type de propriété' is 'Référentiel' and 'Bases de données (POSTGRES)' is selected. The 'Utiliser une connexion existante' checkbox is unchecked. The 'Version de la base de données' is 'V9 et plus'. The 'Hôte' is 'localhost', 'Port' is '5432', 'Base de données' is 'visionnage_ETL', 'Schéma' is 'public', 'Utilisateur' is 'postgres', and 'Mot de passe' is '*****'. The 'Table' is 'fait_visionnage'. The 'Action sur la table' is 'Supprimer la table si elle existe et la créer' and 'Action sur les données' is 'Insert'. The 'Schéma' is 'Built-in', 'Modifier le schéma' is unchecked, and 'Sync colonnes' is checked.

FIG. 2.29 : Configuration Table de fait

The screenshot shows the configuration for the tPostgresqlOutput component for a dimension table. The 'Database' is 'PostgreSQL', 'Appliquer' is clicked, and 'Type de propriété' is 'Référentiel'. The 'Version de la base de données' is 'V9 et plus'. The 'Hôte' is 'localhost', 'Port' is '5432', 'Base de données' is 'visionnage_ETL', 'Schéma' is 'public', 'Utilisateur' is 'postgres', and 'Mot de passe' is '*****'. The 'Table' is 'dim_utilisateur'. The 'Action sur la table' is 'Supprimer la table si elle existe et la créer' and 'Action sur les données' is 'Insert'. The 'Schéma' is 'Built-in', 'Modifier le schéma' is unchecked, and 'Sync colonnes' is checked.

(a) Configuration Dimension Utilisateur

The screenshot shows the configuration for the tPostgresqlOutput component for a dimension table. The 'Database' is 'PostgreSQL', 'Appliquer' is clicked, and 'Type de propriété' is 'Référentiel'. The 'Version de la base de données' is 'V9 et plus'. The 'Hôte' is 'localhost', 'Port' is '5432', 'Base de données' is 'visionnage_ETL', 'Schéma' is 'public', 'Utilisateur' is 'postgres', and 'Mot de passe' is '*****'. The 'Table' is 'dim_contenu'. The 'Action sur la table' is 'Supprimer la table si elle existe et la créer' and 'Action sur les données' is 'Insert'. The 'Schéma' is 'Built-in', 'Modifier le schéma' is unchecked, and 'Sync colonnes' is checked.

(b) Configuration Dimension Contenu

The screenshot shows the configuration for the tPostgresqlOutput component for a dimension table. The 'Database' is 'PostgreSQL', 'Appliquer' is clicked, and 'Type de propriété' is 'Référentiel'. The 'Version de la base de données' is 'V9 et plus'. The 'Hôte' is 'localhost', 'Port' is '5432', 'Base de données' is 'visionnage_ETL', 'Schéma' is 'public', 'Utilisateur' is 'postgres', and 'Mot de passe' is '*****'. The 'Table' is 'dim_plateforme'. The 'Action sur la table' is 'Supprimer la table si elle existe et la créer' and 'Action sur les données' is 'Insert'. The 'Schéma' is 'Built-in', 'Modifier le schéma' is unchecked, and 'Sync colonnes' is checked.

(c) Configuration Dimension Plateforme

The screenshot shows the configuration for the tPostgresqlOutput component for a dimension table. The 'Database' is 'PostgreSQL', 'Appliquer' is clicked, and 'Type de propriété' is 'Référentiel'. The 'Version de la base de données' is 'V9 et plus'. The 'Hôte' is 'localhost', 'Port' is '5432', 'Base de données' is 'visionnage_ETL', 'Schéma' is 'public', 'Utilisateur' is 'postgres', and 'Mot de passe' is '*****'. The 'Table' is 'dim_date'. The 'Action sur la table' is 'Supprimer la table si elle existe et la créer' and 'Action sur les données' is 'Insert'. The 'Schéma' is 'Built-in', 'Modifier le schéma' is unchecked, and 'Sync colonnes' is checked.

(d) Configuration Dimension Date

FIG. 2.30 : Configuration Tables de Dimensions

7. Connexion complète du flux Talend :

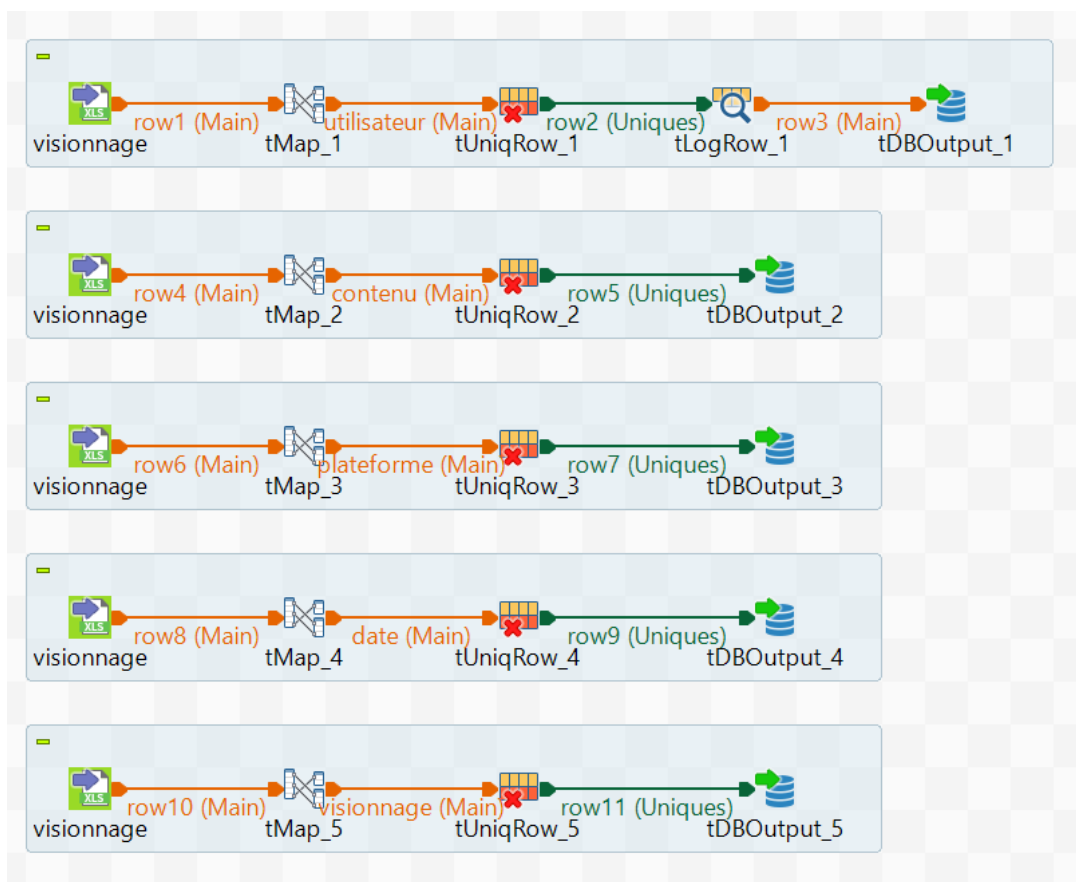


FIG. 2.31 : Job complet de chargement dans PostgreSQL

8. **Exécution et vérification** : Le Job Talend a été exécuté avec succès, et les tables remplies ont été vérifiées dans pgAdmin.

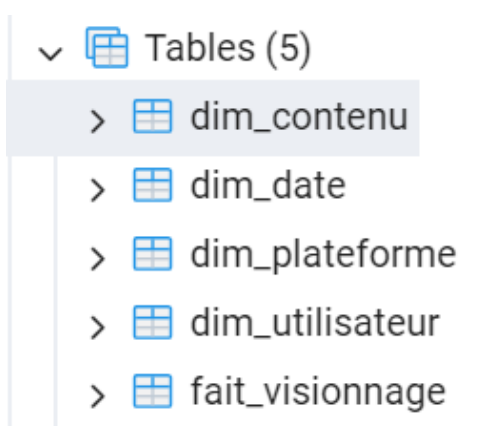


FIG. 2.32 : Tables chargées dans PostgreSQL visibles via pgAdmin

Cette étape de chargement finalise le processus ETL en rendant les données prêtes pour l'analyse dans un outil de visualisation comme Power BI.

2.5 Installation du code SQL de la base de données PostgreSQL

Afin de partager et réutiliser la base de données créée, nous avons généré un fichier SQL contenant l'ensemble des commandes de création des tables et d'insertion des données. Ce fichier peut être réinstallé sur n'importe quelle machine disposant de PostgreSQL.

1. Création d'une variable d'environnement pour PostgreSQL :

Sur Windows, il est nécessaire d'ajouter le chemin du dossier `bin` de PostgreSQL dans la variable d'environnement `PATH` pour accéder aux commandes `psql` et `pg_dump` depuis n'importe quel terminal.

- Aller dans Paramètres > Variables d'environnement
- Dans la section "Path", cliquer sur "Modifier" puis "Nouveau"
- Ajouter le chemin suivant (selon le dossier et la version installée) :

```
1 C:\Program Files\PostgreSQL\16\bin
2 C:\Program Files (x86)\PostgreSQL\16\bin
```

Listing 2.1: Chemin de PostgreSQL

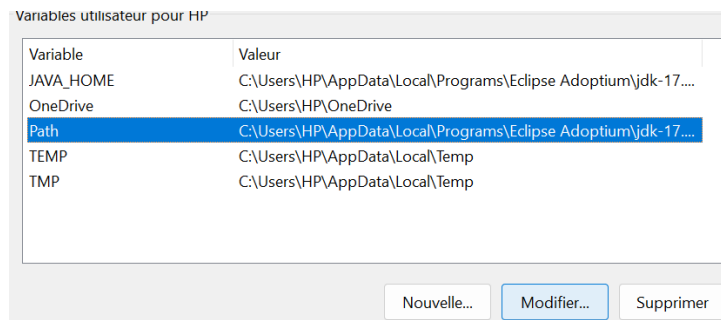


FIG. 2.33 : Ajout de PostgreSQL au PATH dans Windows

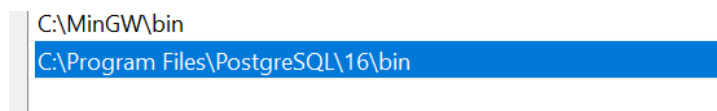


FIG. 2.34 : Ajout de PostgreSQL au PATH dans Windows

2. Vérification de l'installation :

Pour confirmer que PostgreSQL est accessible depuis le terminal, on tape l'une des commandes suivantes dans l'invite de commandes (CMD) :

```
1 psql --version
2 pg_dump --version
```

Listing 2.2: Commande pour vérifier la version de PostgreSQL

Si l'installation est correcte, la version de PostgreSQL s'affiche et on peut maintenant faire l'export.

3. Commande finale pour exporter la base en script SQL :

On tape la commande suivante dans l'invite de commandes (CMD) :

```
1 pg_dump -U postgres -d visionnagedb_etl -F p -f "C:\Users\HP\Documents
  \visionnagedb_etl.sql"
```

Listing 2.3: Commande pour exécuter le script SQL dans PostgreSQL

- -U : nom de l'utilisateur PostgreSQL
- -d : nom de la base de données existante (à créer avant si nécessaire)
- -f : chemin vers le fichier SQL contenant le code d'installation
- Taper mot de passe quand demandé.

4. Résultat attendu :

Après l'exécution de la commande, toutes les tables et leurs données sont automatiquement générées. On obtiendra un fichier complet visionnagedb_etl.sql dans le dossier Documents, prêt à être partagé ou versionné..

2.6 Conclusion

Les processus ETL représentent un pilier fondamental de la gestion moderne des données. Dans ce chapitre, nous avons utilisé Talend pour extraire, transformer et charger les données dans une base de données relationnelle.

Chapitre 3

Création du cube

3.1 Introduction

Dans ce chapitre nous allons créer une structure multidimensionnelle permettant de faciliter l'analyse avancée des données selon plusieurs axes.

L'outil utilisé : **Pentaho Schema Workbench**, est un logiciel qui permet de concevoir des cubes OLAP (structures multidimensionnelles) pour l'analyse avancée.

3.2 Création du cube OLAP

3.2.1 Définition d'un cube OLAP

le cube OLAP est une base de données multidimensionnelle organisée en tables qui permet de traiter et d'analyser plusieurs dimensions de données beaucoup plus rapidement et efficacement qu'une base de données relationnelle.

3.2.2 Installation de Pentaho Schema Workbench

- Installation de Pentaho Schema Workbench.
- Lancer pentaho avec le fichier : workbench.bat

3.2.3 Connexion à la base de données

- Connexion à la base de données **visionnagedb_etl** via le panneau de configuration de Pentaho :
 - Nom : CubeOlap.
 - Type : MySQL.
 - Hôte : localhost.
 - Nom de la base : la base de données **visionnagedb_etl** réalisée dans le chapitre précédent.
 - Identifiant et mot de passe : Les identifiants de connexion configurés dans MySQL.

- Tester la connexion en cliquant sur le bouton Test :

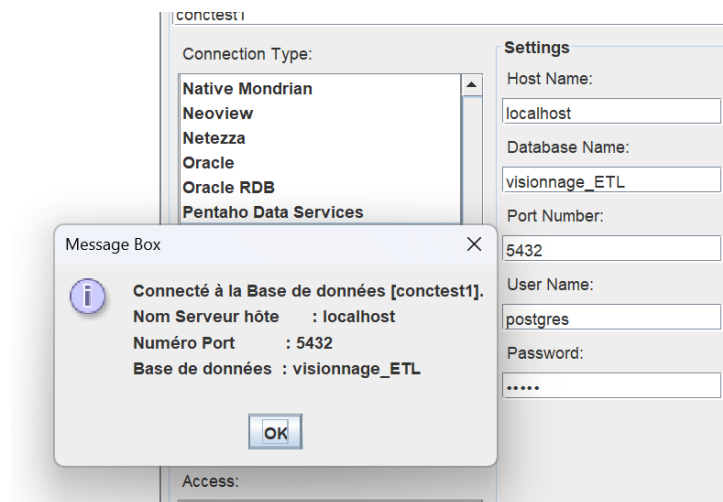


FIG. 3.1 : Connexion réussi

3.2.4 Création du cube

- Création d'un nouveau schéma qui servira de base pour définir les cubes, les dimensions, les mesures et les relations avec la base de données.

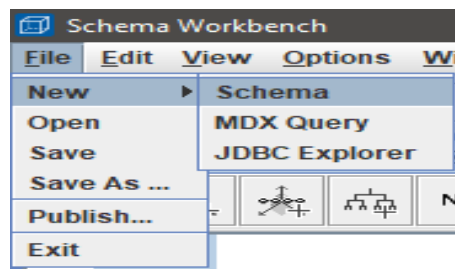


FIG. 3.2 : Schéma

- Attribuer un nom au schéma

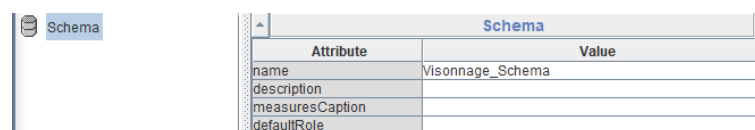


FIG. 3.3 : Nom du schéma

- À partir du schéma, nous créons le cube OLAP et lui attribuons le nom **Vison-nage_Cube**.

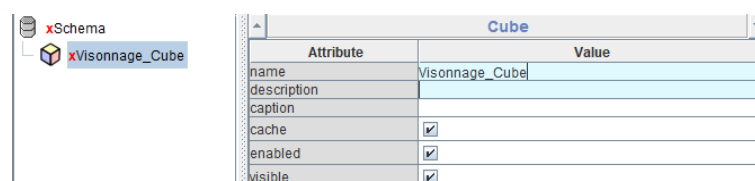
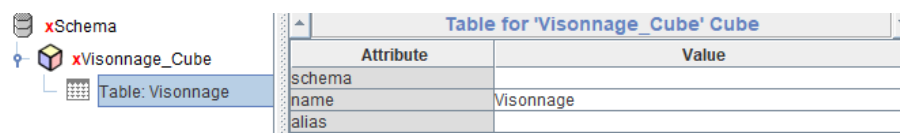


FIG. 3.4 : Création du cube Visonnage_Cube

1. Définition de la table de fait :



| Attribute | Value |
|-----------|-----------|
| schema | |
| name | Visonnage |
| alias | |

FIG. 3.5 : Création de la table de fait

2. Ajout des mesures :

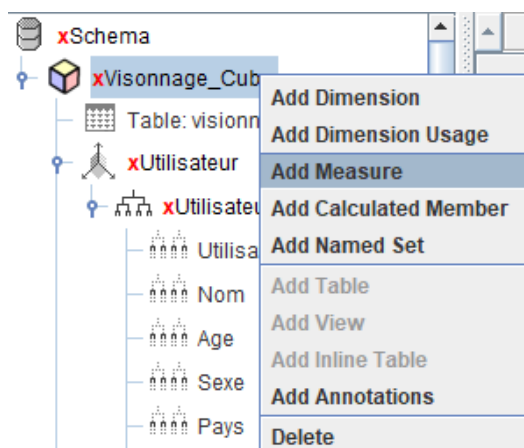
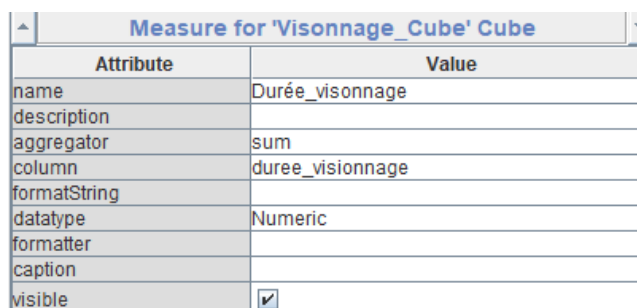


FIG. 3.6 : Ajout des mesures

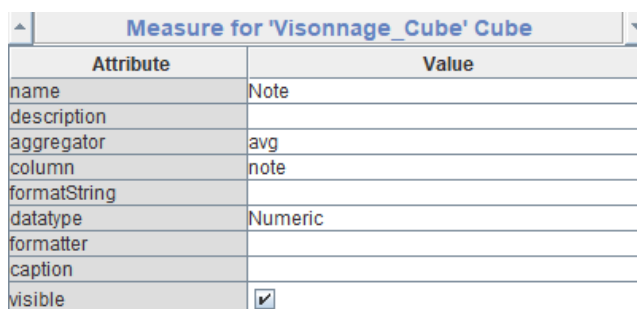
- **Durée de visionnage** : permet d'estimer le temps total passé sur un contenu. Nous avons utilisé l'agrégateur SUM pour l'additionner dans le cube.



| Attribute | Value |
|--------------|-------------------------------------|
| name | Durée_visionnage |
| description | |
| aggregator | sum |
| column | duree_visionnage |
| formatString | |
| datatype | Numeric |
| formatter | |
| caption | |
| visible | <input checked="" type="checkbox"/> |

FIG. 3.7 : Ajout de la mesure Durée de visionnage

- **Note** : est évaluée par les utilisateurs afin de représenter une appréciation globale et équilibrée, nous avons utilisé l'agrégateur AVG pour obtenir la note moyenne.



| Attribute | Value |
|--------------|-------------------------------------|
| name | Note |
| description | |
| aggregator | avg |
| column | note |
| formatString | |
| datatype | Numeric |
| formatter | |
| caption | |
| visible | <input checked="" type="checkbox"/> |

FIG. 3.8 : Ajout de la mesure Note

| Attribute | Value |
|-----------------|-------------------------------------|
| name | Age |
| description | |
| table | utilisateur |
| column | age |
| nameColumn | age |
| parentColumn | |
| nullParentValue | |
| ordinalColumn | |
| type | Numeric |
| internalType | |
| uniqueMembers | <input type="checkbox"/> |
| levelType | Regular |
| hideMemberIf | |
| approxRowCount | |
| caption | |
| captionColumn | |
| formatter | |
| visible | <input checked="" type="checkbox"/> |

(a) Attribut Age

| Attribute | Value |
|-----------------|-------------------------------------|
| name | Sexe |
| description | |
| table | utilisateur |
| column | sexe |
| nameColumn | sexe |
| parentColumn | |
| nullParentValue | |
| ordinalColumn | |
| type | String |
| internalType | |
| uniqueMembers | <input type="checkbox"/> |
| levelType | Regular |
| hideMemberIf | Never |
| approxRowCount | |
| caption | |
| captionColumn | |
| formatter | |
| visible | <input checked="" type="checkbox"/> |

(b) Attribut Sexe

| Attribute | Value |
|-----------------|-------------------------------------|
| name | Pays |
| description | |
| table | utilisateur |
| column | pays |
| nameColumn | pays |
| parentColumn | |
| nullParentValue | |
| ordinalColumn | |
| type | String |
| internalType | |
| uniqueMembers | <input type="checkbox"/> |
| levelType | Regular |
| hideMemberIf | |
| approxRowCount | |
| caption | |
| captionColumn | |
| formatter | |
| visible | <input checked="" type="checkbox"/> |

(c) Attribut Pays

| Attribute | Value |
|-----------------|-------------------------------------|
| name | Nom |
| description | |
| table | utilisateur |
| column | nom_utilisateur |
| nameColumn | nom_utilisateur |
| parentColumn | |
| nullParentValue | |
| ordinalColumn | |
| type | String |
| internalType | |
| uniqueMembers | <input type="checkbox"/> |
| levelType | Regular |
| hideMemberIf | Never |
| approxRowCount | |
| caption | |
| captionColumn | |
| formatter | |
| visible | <input checked="" type="checkbox"/> |

(d) Attribut nom

FIG. 3.12 : Structure de la table Utilisateur

Après avoir défini la première dimension, nous avons répétés les mêmes étapes pour les tables restantes, tout en adaptant les paramètres à la structure spécifique de chaque table.

– **Table Contenu Audiovisuel :**

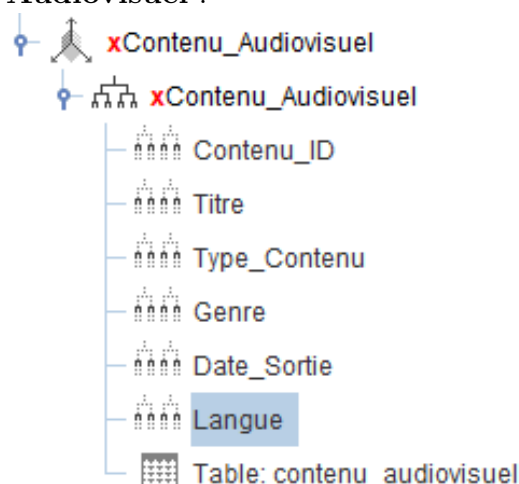


FIG. 3.13 : Structure de la table Contenu Audiovisuel

– **Table Plateforme :**

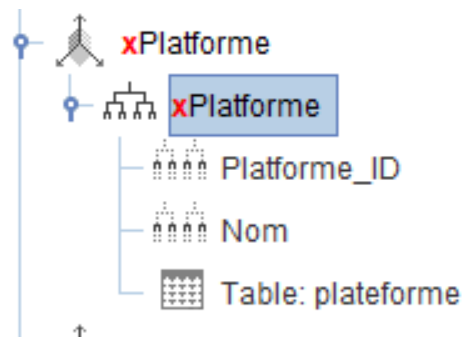


FIG. 3.14 : Structure de la table Platforme

– Table Date :

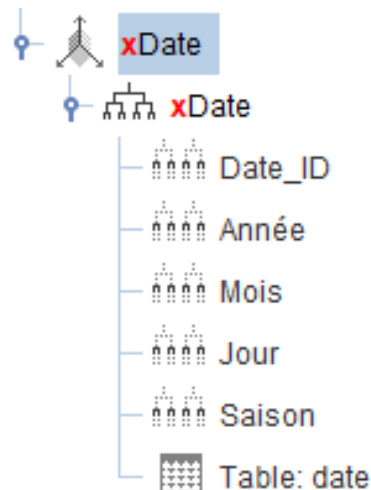


FIG. 3.15 : Structure de la table Date

3.2.5 Test du cube avec des requêtes MDX

Les requêtes MDX (MultiDimensional eXpressions) sont l'équivalent des requêtes SQL, mais pour interroger un cube OLAP. Elles permettent de faire des analyses multi-dimensionnelles sur les données modélisées dans le schéma OLAP.

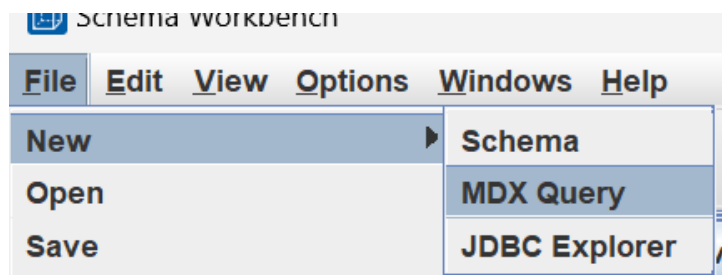


FIG. 3.16 : MDX Query

1. Durée totale de visionnage par plateforme

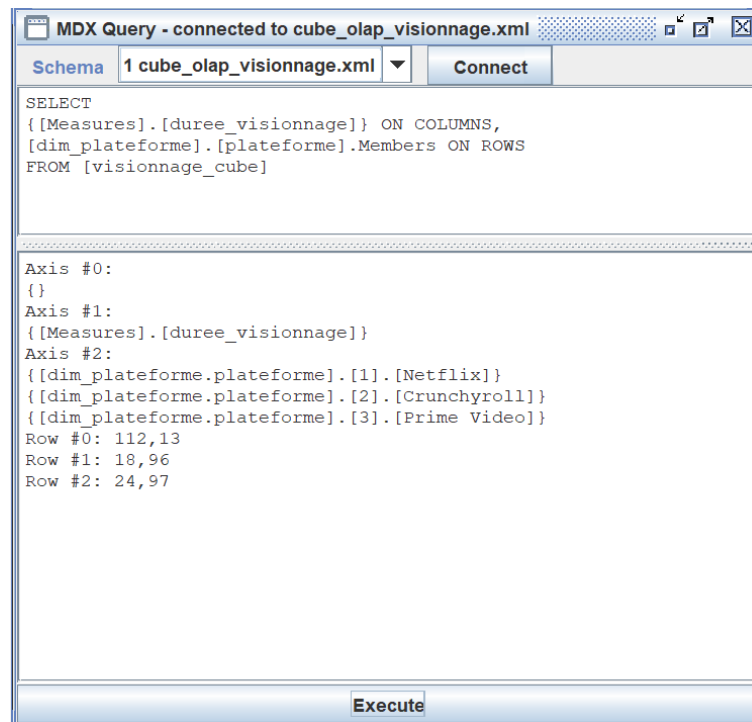


FIG. 3.17 : Requête MDX (1) et son résultat

Remarque : Cette requête permet de visualiser les plateformes les plus utilisées en termes de temps cumulé de visionnage et on observe que netflix est largement au dessus.

2. Top 5 contenus les mieux notés

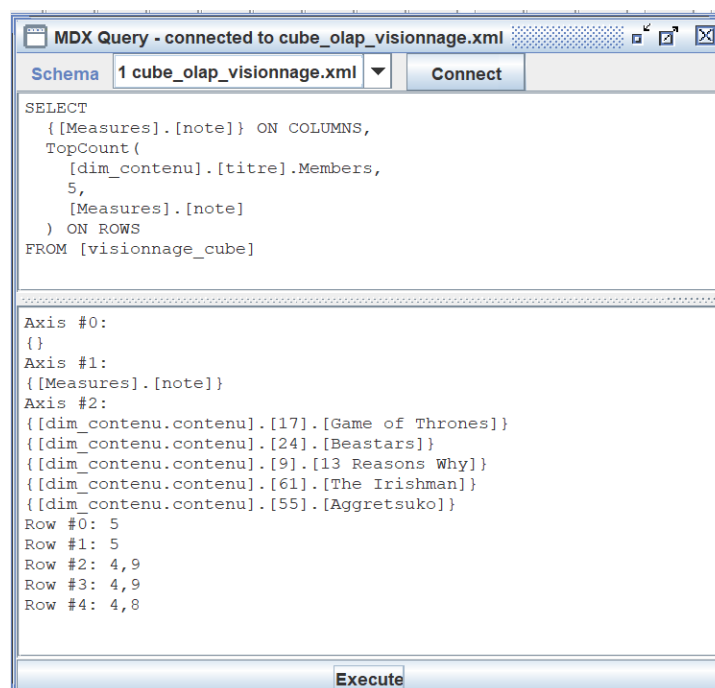


FIG. 3.18 : Requête MDX (2) et son résultat

Remarque : Cette requête permet de lister les contenus audiovisuels ayant obtenu les meilleures évaluations.

3. 5 utilisateurs ayant la plus petite durée de visionnage totale

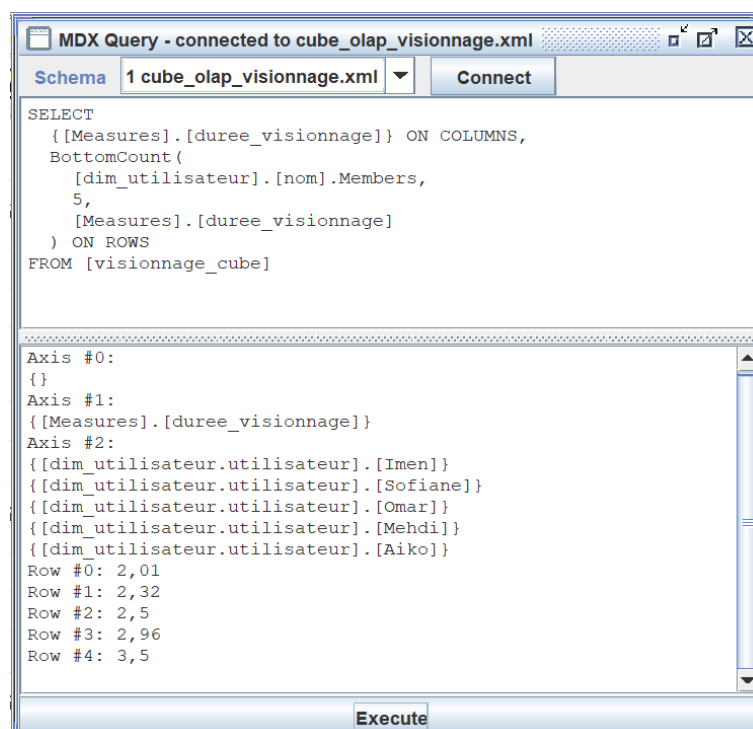


FIG. 3.19 : Requête MDX (3) et son résultat

Remarque : Cette requête permet de retourner les 5 utilisateurs avec la plus petite valeur de durée de visionnage.

3.2.6 Enregistrement du schéma OLAP

Après avoir finalisé la définition du cube, nous avons procédé à l'enregistrement du schéma OLAP sous forme d'un fichier XML.

Cette étape marque la finalisation technique du modèle OLAP, et permet son exploitation dans les étapes d'analyse.

```
<Schema name="Visonage_Schema">
  <Cube name="Visonage_Cube" visible="true" cache="true" enabled="true">
    <Table name="visionnage" alias=""></Table>
    <Dimension type="StandardDimension" visible="true" foreignKey="utilisateur_id" name="Utilisateur">
      <Hierarchy name="Utilisateur" visible="true" hasAll="true" primaryKey="utilisateur_id">
        <Table name="utilisateur" alias=""></Table>
        <Level name="utilisateur_id" visible="true" table="utilisateur" column="utilisateur_id" nameColumn="utilisateur_id" type="Numeric" uniqueMembers="false" levelType="Regular"></Level>
        <Level name="Nom" visible="true" table="utilisateur" column="nom_utilisateur" nameColumn="nom_utilisateur" type="String" uniqueMembers="false" levelType="Regular"></Level>
        <Level name="Age" visible="true" table="utilisateur" column="age" nameColumn="age" type="Numeric" uniqueMembers="false" levelType="Regular"></Level>
        <Level name="Sexe" visible="true" table="utilisateur" column="sexe" nameColumn="sexe" type="String" uniqueMembers="false" levelType="Regular"></Level>
        <Level name="Pays" visible="true" table="utilisateur" column="pays" nameColumn="pays" type="String" uniqueMembers="false" levelType="Regular"></Level>
      </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="contenu_id" name="Contenu_Audiovisuel">
      <Hierarchy name="Contenu_Audiovisuel" visible="true" hasAll="true" primaryKey="contenu_id">
        <Table name="contenu_audiovisuel" alias=""></Table>
        <Level name="Contenu_ID" visible="true" table="contenu_audiovisuel" column="contenu_id" nameColumn="contenu_id" type="Numeric" uniqueMembers="false" levelType="Regular"></Level>
        <Level name="Titre" visible="true" table="contenu_audiovisuel" column="titre" nameColumn="titre" type="String" uniqueMembers="false" levelType="Regular"></Level>
        <Level name="Type_contenu" visible="true" table="contenu_audiovisuel" column="type_contenu" nameColumn="type_contenu" type="String" uniqueMembers="false" levelType="Regular"></Level>
        <Level name="Genre" visible="true" table="contenu_audiovisuel" column="genre" nameColumn="genre" type="String" uniqueMembers="false" levelType="Regular"></Level>
        <Level name="Date_Sortie" visible="true" table="contenu_audiovisuel" column="date_sortie" nameColumn="date_sortie" type="Timestamp" uniqueMembers="false" levelType="Regular"></Level>
        <Level name="Langue" visible="true" table="contenu_audiovisuel" column="langue" nameColumn="langue" type="String" uniqueMembers="false" levelType="Regular"></Level>
      </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="plateforme_id" name="Plateforme">
      <Hierarchy name="Plateforme" visible="true" hasAll="true" primaryKey="plateforme_id">
        <Table name="plateforme" alias=""></Table>
        <Level name="plateforme_id" visible="true" table="plateforme" column="plateforme_id" nameColumn="plateforme_id" type="Numeric" uniqueMembers="false" levelType="Regular"></Level>
        <Level name="Nom" visible="true" table="plateforme" column="plateforme" nameColumn="plateforme" type="String" uniqueMembers="false" levelType="Regular"></Level>
      </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" visible="true" foreignKey="date_id" name="Date">
      <Hierarchy name="Date" visible="true" hasAll="true" primaryKey="date_id">
        <Table name="date" alias=""></Table>
        <Level name="Date_ID" visible="true" table="date" column="date_id" nameColumn="date_id" type="Numeric" uniqueMembers="false" levelType="Regular"></Level>
        <Level name="Année" visible="true" table="date" column="annee" nameColumn="annee" type="Numeric" uniqueMembers="false" levelType="Regular"></Level>
      </Hierarchy>
    </Dimension>
  </Cube>
</Schema>
```

FIG. 3.20 : Le fichier XML

3.3 Conclusion

Ce chapitre a donc permis de créer un cube OLAP en définissant la table de faits, les dimensions et les mesures associées.

Le modèle a ensuite été testé à l'aide de requêtes MDX afin de vérifier sa cohérence et sa capacité à répondre aux besoins d'analyse.

Chapitre 4

Analyse et Visualisation des Résultats

4.1 Introduction

Dans cette section, nous présentons l'analyse des données issues de la base *visionnagedb_etl*, en mettant l'accent sur les comportements des utilisateurs en matière de consommation de contenus audiovisuels. Pour mener à bien cette analyse, nous avons utilisé ces outils principaux :

- **Power BI** : outil de Business Intelligence développé par Microsoft, permettant la création de rapports interactifs et de tableaux de bord dynamiques à partir de sources de données diverses.



FIG. 4.1 : Logo de Power Bi.

L'objectif de cette analyse est de dégager des tendances de consommation, de comprendre les préférences des utilisateurs selon leurs caractéristiques démographiques (âge, sexe, pays), et d'évaluer les performances des plateformes de diffusion.

4.2 Connexion à la base de données MySQL

Avant de créer les visualisations, il a été nécessaire d'établir une connexion entre Power BI et le système de gestion de base de données relationnelle **MySQL**, qui contient la base *visionnagedb_etl*.

4.2.1 Étapes de connexion

1. **Installation du connecteur MySQL pour Power BI** : il a fallu d'abord installer le pilote ODBC MySQL (MySQL Connector/NET) compatible avec Power BI pour établir la connexion.
2. **Connexion dans Power BI** :
 - Depuis Power BI Desktop, cliquer sur *Accueil > Obtenir les données > Plus > Base de données > MySQL*.

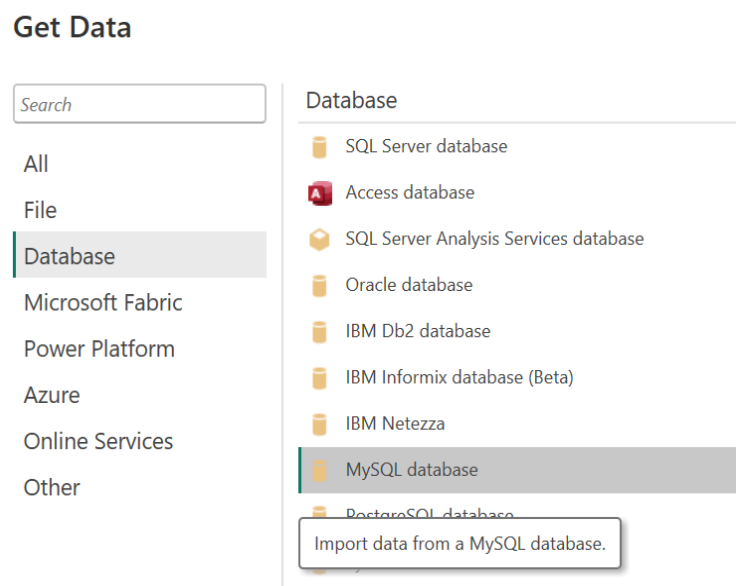


FIG. 4.2 : Obtention des données

- Renseigner les informations de connexion :
 - Serveur : adresse IP ou nom d'hôte (ex. : `localhost` ou `127.0.0.1`).
 - Base de données : `visionnagedb_etl`.
 - Identifiants : nom d'utilisateur et mot de passe MySQL.



FIG. 4.3 : Insertion des infos de connexion

- Cliquer sur *Se connecter*, puis sélectionner les tables souhaitées.

Navigator

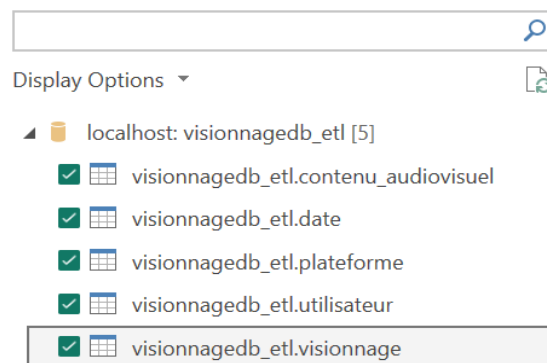


FIG. 4.4 : Sélection des dimensions

3. Chargement des données :

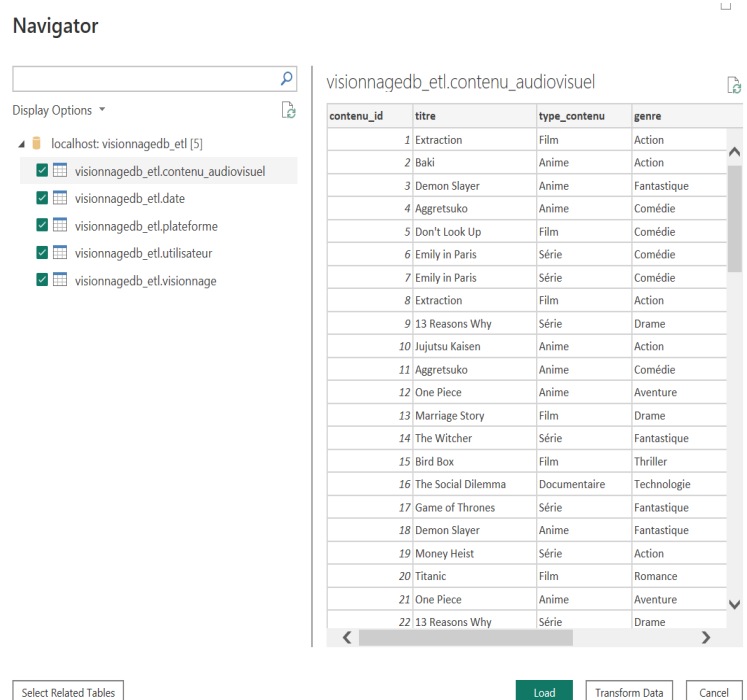


FIG. 4.5 : Sélection des dimensions

Load

- visionnagedb_etl contenu_audiovisuel
Waiting for other queries...
- visionnagedb_etl date
Waiting for other queries...
- visionnagedb_etl plateforme
Waiting for other queries...
- visionnagedb_etl utilisateur
Waiting for other queries...
- visionnagedb_etl visionnage
Waiting for other queries...

FIG. 4.6 : Chargement et importation des données

Les tables sélectionnées ont été chargées dans Power BI, prêtes à être modélisées et analysées.

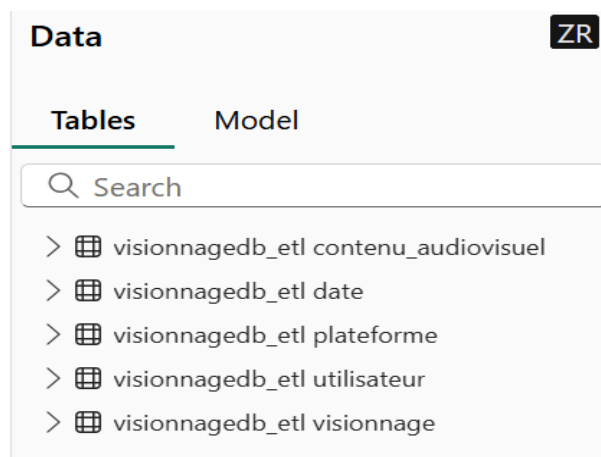


FIG. 4.7 : Chargement et importation des données

Cette connexion directe à MySQL permet d'automatiser les mises à jour des données dans Power BI à chaque actualisation.

4.3 Création des graphiques de visualisation

Pour représenter visuellement les données, plusieurs graphiques et tableaux ont été créés afin de répondre aux principales problématiques d'analyse. Voici les étapes et résultats principaux.

4.3.1 Création des mesures

Cette analyse met en lumière les comportements de visionnage en fonction des profils utilisateurs.

- Cliquer sur la table Visionnage.
- Cliquer sur "Modélisation" ou "Modeling" en haut, puis sur "Nouvelle mesure" ou "New measure".

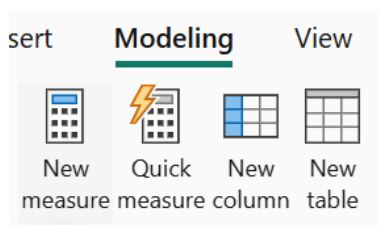


FIG. 4.8 : Ajout d'une mesure personnalisée dans Power BI

- Mesure : **Nombre total des utilisateurs.**

```
1 Nombre d'utilisateurs =
2 COUNT('visionnagedb_etl_visionnage'[utilisateur_id])
```

Listing 4.1: DAX code to count number of users

- Mesure : **Age moyen des utilisateurs.**

```
1 Age moyen des utilisateurs =
2 AVERAGE('visionnagedb_etl_utilisateur'[age])
```

Listing 4.2: DAX code to calculate average age

- Résultat des mesures :

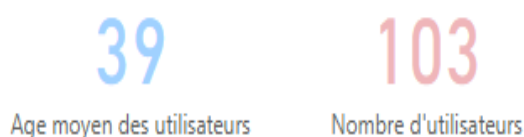


FIG. 4.9 : Résultat des mesures

4.3.2 Analyse démographique

Cette analyse met en lumière les comportements de visionnage en fonction des profils utilisateurs.

- **Création des colonnes calculées :**

Avant de pouvoir représenter les tranches d'âge ou les continents dans les visualisations, il est nécessaire d'ajouter des colonnes personnalisées à la table `utilisateur`. Voici les étapes à suivre :

1. **Sélectionner la bonne table**

Dans la table `utilisateur` on crée les colonnes.

- Cliquer sur "Modélisation" en haut, puis sur "Nouvelle colonne" ou "New column".

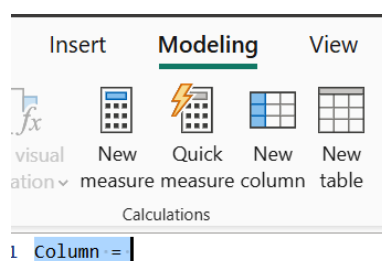


FIG. 4.10 : Ajout d'une colonne personnalisée dans Power BI

2. **Ajouter ce code pour créer la colonne `TrancheAge` :**

```

1 TrancheAge =
2 SWITCH(
3     TRUE(),
4     'visionnagedb_etl utilisateur'[age] < 18, -"017",
5     'visionnagedb_etl utilisateur'[age] < 35, -"1834",
6     'visionnagedb_etl utilisateur'[age] < 60, -"3559",
7     "60+"
8 )

```

Listing 4.3: DAX code for age group classification

3. **Ajouter ce code pour créer la colonne `Continent` :**

```

1 Continent =
2 SWITCH(
3     TRUE(),
4     'visionnagedb_etl utilisateur'[pays] IN {"France", "Belgique",
5         "Italie"}, "Europe",
6     'visionnagedb_etl utilisateur'[pays] IN {"Égypte", "Maroc", "
7         Algérie", "Tunisie", "Côte d'Ivoire", "Sénégal"}, "Afrique",
8     'visionnagedb_etl utilisateur'[pays] IN {"Canada"}, "Amérique
9     du Nord",
10    'visionnagedb_etl utilisateur'[pays] IN {"Japon"}, "Asie",
11    "Autre"
12 )

```

Listing 4.4: DAX code for continent classification

Deux colonnes ont ainsi été ajoutées à la table `utilisateur` :

- `TrancheAge` : pour regrouper les utilisateurs selon leur groupe d'âge (0–17, 18–34, 35–59, 60+).
- `Continent` : pour regrouper les utilisateurs selon leur pays d'origine.

- **Répartition par tranche d'âge :**

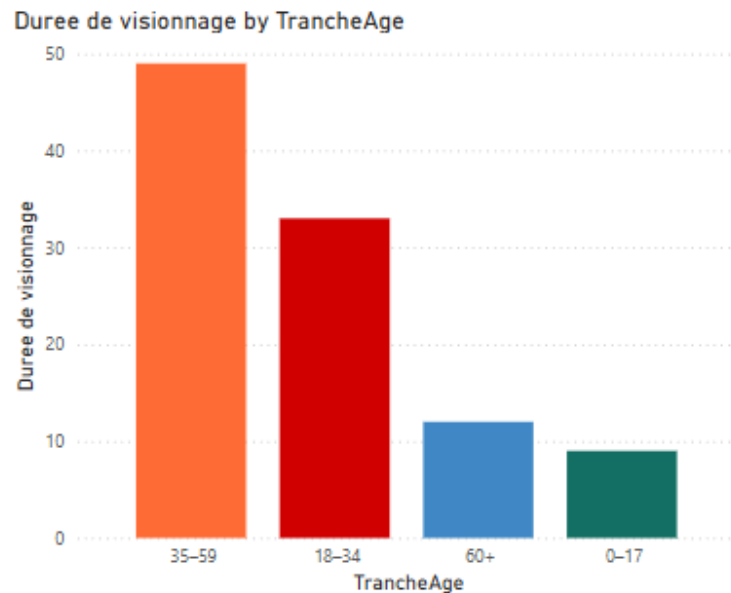


FIG. 4.11 : Visionnages par tranche d'âge

Graphique utilisé : Histogramme à colonnes empilées (vertical)

Interprétation : Montre la répartition des visionnages selon les tranches d'âge.

Résultat : Les 35-59 ans sont les plus actifs, représentant le public cible principal.

- **Répartition par sexe :**

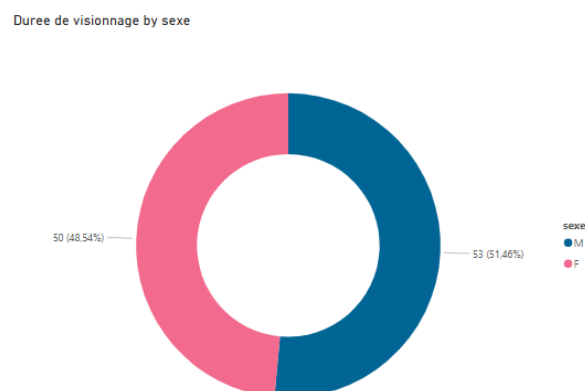


FIG. 4.12 : Visionnages par sexe

Graphique utilisé : Diagramme en anneau

Interprétation : Répartition des visionnages selon le sexe des utilisateurs.

Résultat : L'écart est faible, mais une légère dominance masculine est observée.

- Répartition par pays :

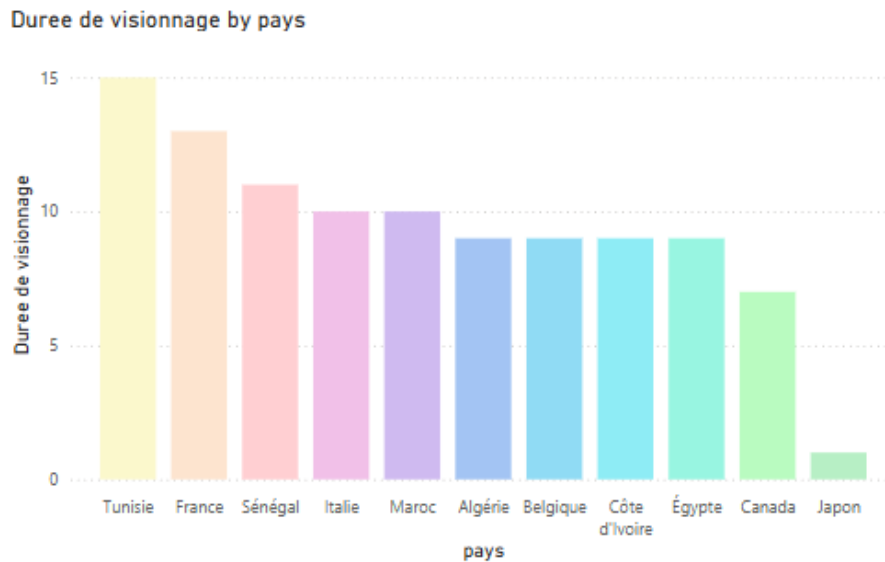


FIG. 4.13 : Visionnages par pays d'origine

Graphique utilisé : Histogramme à colonnes empilées (vertical)

Interprétation : Affiche les durées de visionnage selon les pays d'origine des utilisateurs.

Résultat : La Turquie est le pays le plus représenté et le Japon le moins représenté.

- Carte géographique (via Bing Maps) :

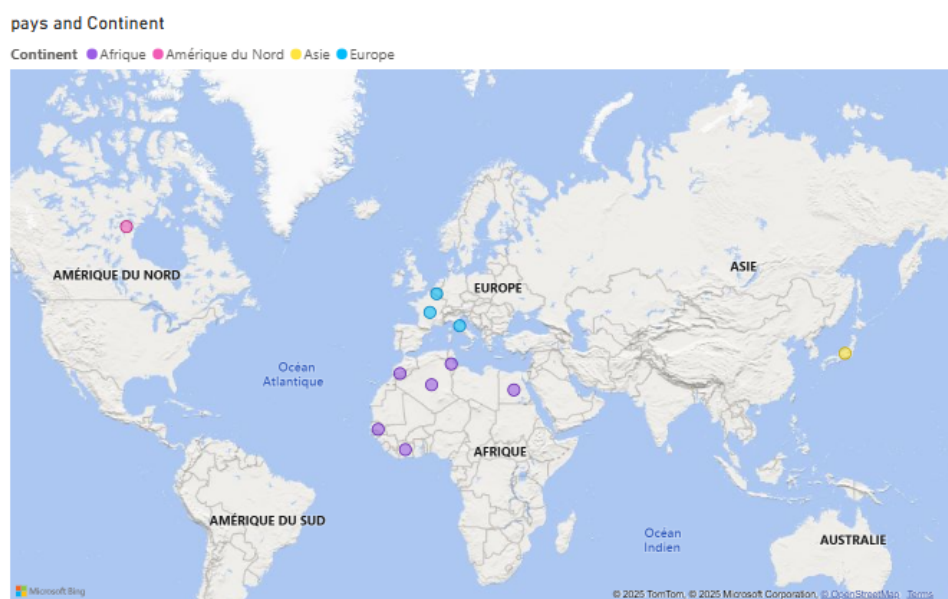


FIG. 4.14 : Localisation des utilisateurs par pays

Graphique utilisé : Carte géographique (Bing Maps)

Interprétation : Localise les utilisateurs par pays.

Résultat : Concentration des utilisateurs en Afrique et Europe.

- **Corrélation âge - type de contenu :**

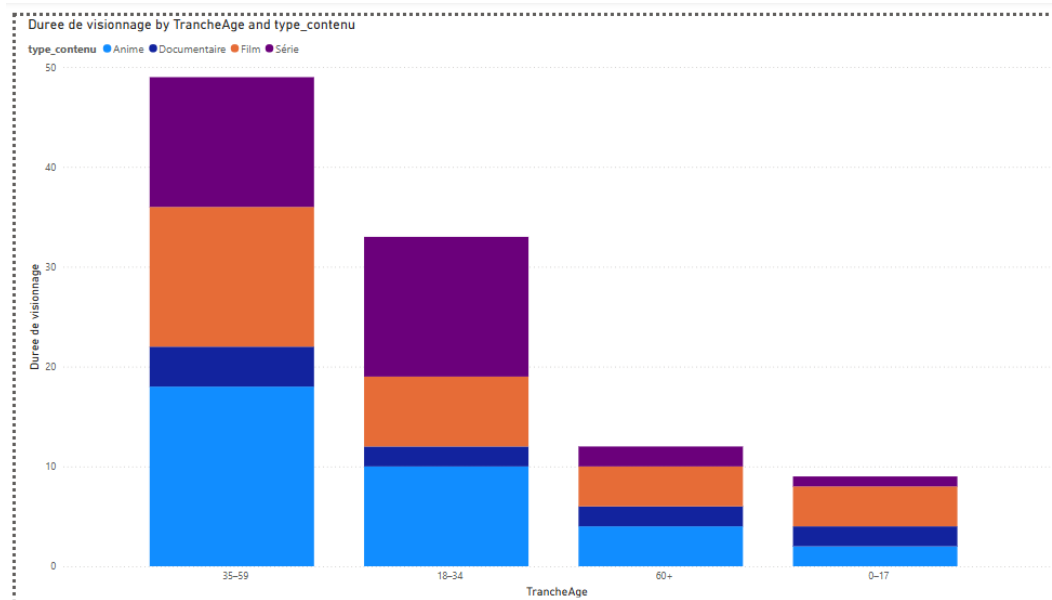


FIG. 4.15 : Préférences de contenu selon les tranches d'âge

Graphique utilisé : Histogramme à colonnes empilées (vertical)

Interprétation : Met en relation les types de contenus avec les tranches d'âge.

Résultat : Les plus jeunes préfèrent animes et séries, les plus âgés préfèrent les films.

4.3.3 Analyse par plateforme

Un tableau de bord spécifique a été dédié à l'étude des performances des plateformes (Netflix, Crunchyroll, Prime Video) :

- Diagrammes à barres : nombre de visionnages par plateforme.

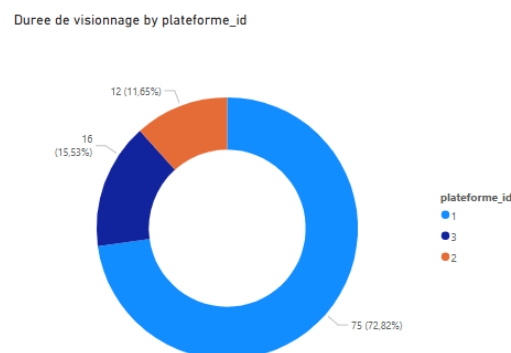


FIG. 4.16 : Visionnages par plateforme

Graphique utilisé : Diagramme en anneau

Interprétation : Compare le nombre de visionnages par plateforme.

Résultat : Netflix domine, suivi par Prime Video et Crunchyroll.

- Évolution des usages dans le temps par plateforme.

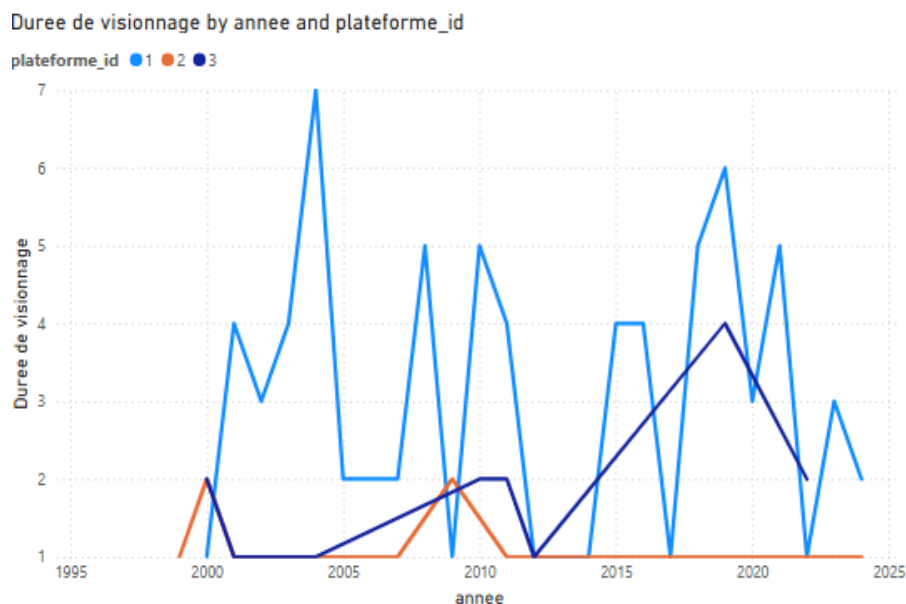


FIG. 4.17 : Visionnages par plateforme au fil des années

Graphique utilisé : Graphique en courbes

Interprétation : Montre l'évolution des usages des plateformes au fil du temps.

Résultat : Netflix est beaucoup plus utilisé contrairement à Crunchyroll au fil des années.

4.3.4 Tendances temporelles

Les données temporelles ont permis d'observer les variations de consommation :

- Répartition saisonnière (hiver, printemps, été, automne).

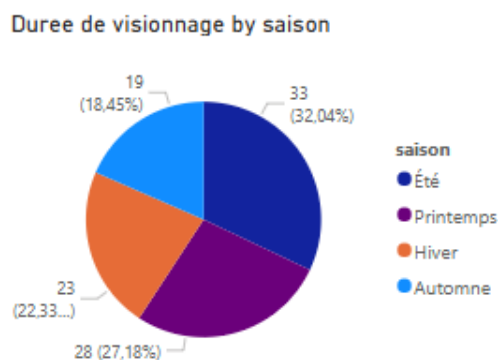


FIG. 4.18 : Visionnages selon les saisons

Graphique utilisé : Diagramme en secteurs

Interprétation : Indique les variations de consommation selon les saisons.

Résultat : Les pics de visionnage sont observés en été.

4.3.5 Analyse du contenu

Enfin, un dernier tableau de bord s'est focalisé sur les contenus eux-mêmes :

- Répartition par type de contenu (Film, Série, Anime, etc.).

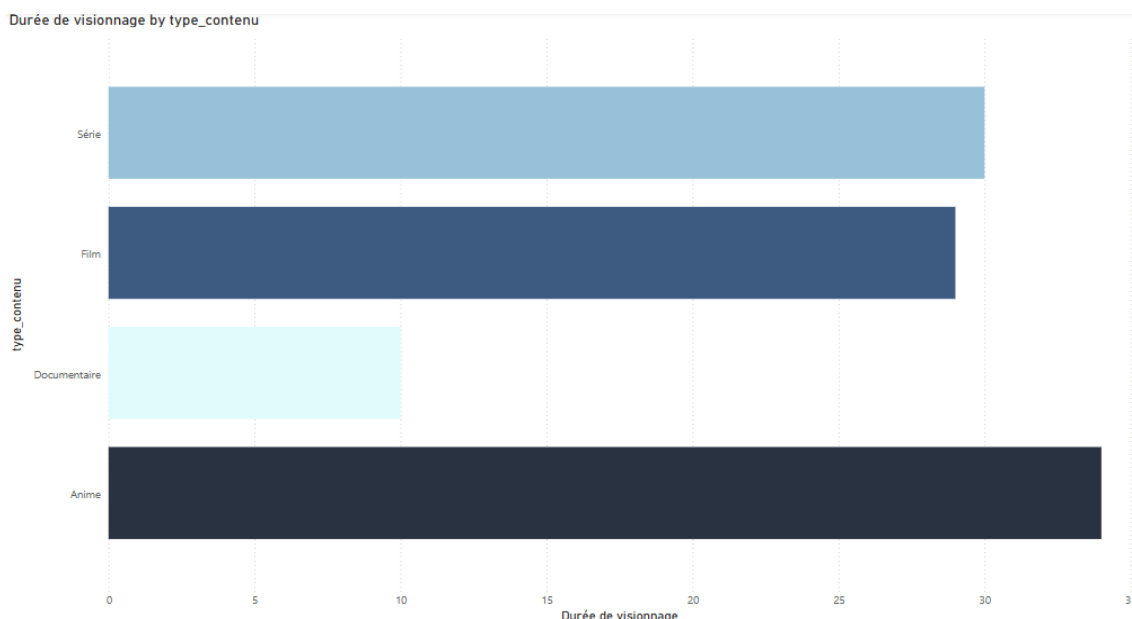


FIG. 4.19 : Répartition par type de contenu

Graphique utilisé : Histogramme à barres empilées (horizontal)

Interprétation : Ce graphique présente la fréquence des visionnages selon le type de contenu (film, série, anime, etc.).

Résultat : Les séries et les animes sont les plus populaires, indiquant un intérêt accru pour les contenus épisodiques et animés.

- Classement des contenus les mieux notés (note supérieure ou égale à 4).

Graphique utilisé : Tableau

Interprétation : Classe les contenus avec des notes supérieures ou égales à 4.

Résultat : Les séries et animes dominent ce classement qualitatif.

| titre | type_contenu | genre | note |
|------------------|--------------|------------------|------|
| Beastars | Anime | Drame | 5 |
| Game of Thrones | Série | Fantastique | 5 |
| 13 Reasons Why | Série | Drame | 4,9 |
| The Irishman | Film | Crime | 4,9 |
| Aggretsuko | Anime | Comédie | 4,8 |
| One Piece | Anime | Aventure | 4,8 |
| Parasite | Film | Thriller | 4,8 |
| Game of Thrones | Série | Fantastique | 4,7 |
| Stranger Things | Série | Science-Fiction | 4,7 |
| Devilman Crybaby | Anime | Horreur | 4,6 |
| Devilman Crybaby | Anime | Horreur | 4,5 |
| The Witcher | Série | Fantastique | 4,5 |
| Attack on Titan | Anime | Drame | 4,4 |
| Beastars | Anime | Drame | 4,4 |
| Aggretsuko | Anime | Comédie | 4,3 |
| Demon Slayer | Anime | Fantastique | 4,3 |
| 13th | Documentaire | Politique | 4,2 |
| Jujutsu Kaisen | Anime | Action | 4,2 |
| La La Land | Film | Comédie musicale | 4,2 |
| Money Heist | Série | Action | 4,2 |
| Parasite | Film | Thriller | 4,2 |
| Stranger Things | Série | Science-Fiction | 4,2 |
| The Office | Série | Comédie | 4,2 |
| Aggretsuko | Anime | Comédie | 4,1 |
| One Piece | Anime | Aventure | 4,1 |
| Jack Ryan | Série | Action | 4 |

FIG. 4.20 : Contenus les mieux notés (note ≥ 4)

- Répartition par genre.

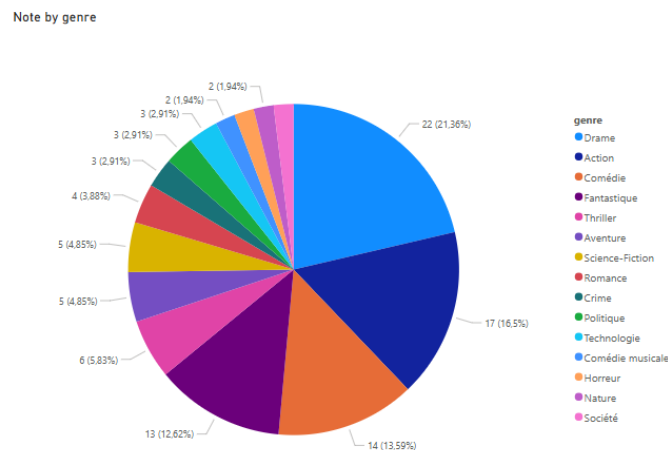


FIG. 4.21 : Répartition par genre

Graphique utilisé : Diagramme en secteurs

Interprétation : Répartition des contenus selon leur genre principal.

Résultat : Les genres les plus populaires sont drama, action et comédie.

4.3.6 Tableau de bord global

Voici le tableau de bord général conçu pour fournir une vue d'ensemble :

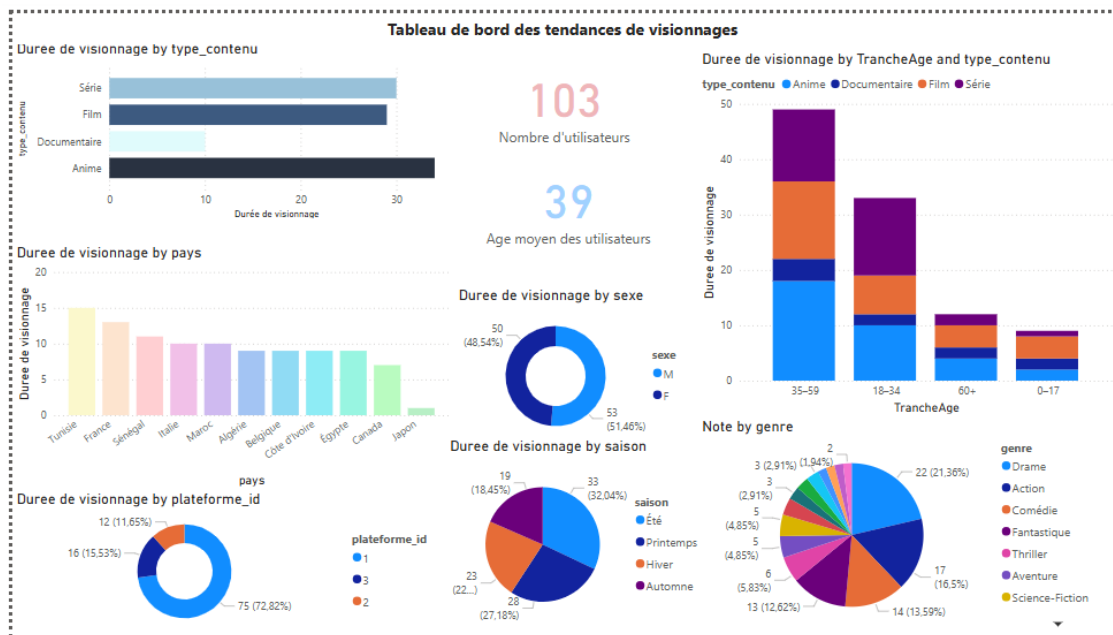


FIG. 4.22 : Tableau de bord général

4.4 Conclusion

Grâce à Power BI, nous avons pu transformer des données brutes en informations visuelles exploitables. Les visualisations ont permis d'identifier les contenus populaires, de détecter les préférences des utilisateurs selon leur profil, et de comparer les plateformes entre elles. Cette analyse visuelle constitue une étape essentielle dans la valorisation des données audiovisuelles, permettant une prise de décision éclairée tant pour les producteurs de contenu que pour les plateformes.

Conclusion générale

Ce projet a permis la mise en œuvre complète d'un **entrepôt de données dédié à l'analyse des animes**, en exploitant efficacement des outils essentiels du monde décisionnel.

Nous avons utilisé **Talend** pour le processus **ETL** (Extraction, Transformation et Chargement), assurant ainsi la qualité et l'intégration des données. Le **WAMP Server** a servi de support pour la gestion des bases de données, tandis que **Pentaho Schema Workbench** a été employé pour la création de **cubes multidimensionnels** facilitant l'analyse **OLAP**.

Pour la **visualisation des résultats**, nous avons opté pour **Microsoft Power BI**, qui a permis de générer des **graphiques interactifs** et des tableaux de bord dynamiques, rendant l'analyse plus intuitive.

Grâce à cette architecture, nous avons pu transformer des données brutes en **informations exploitables** : tendances de popularité par genre, par studio, par source ou encore par période. Ces résultats peuvent guider les **décisions stratégiques** dans la production et le marketing dans l'industrie des animes.

En résumé, ce projet illustre l'importance de :

- **Processus ETL robuste.**
- **Modélisation multidimensionnelle pertinente** via des cubes OLAP.
- **Solutions de Business Intelligence** pour exploiter tout le potentiel des données.

Il démontre comment un **Data Warehouse bien conçu** peut devenir un véritable outil de **pilotage et d'aide à la décision** pour les entreprises du secteur audiovisuel.