

Analyse des Tendances des plateformes de streaming musical

Analyse des Tendances des plateformes de streaming musical

Louis ROCCA, Nicolas BELLANGER, Yanis ARAR

Master 1 Développeur FullStack PAR03 Groupe 1 - Big Data

Introduction

Contexte et motivation

L'industrie du streaming musical est en constante évolution. Les plateformes telles que Spotify, Apple Music et YouTube Music jouent un rôle essentiel dans la diffusion de la musique auprès de millions d'utilisateurs. Analyser leurs données permet de comprendre les préférences des utilisateurs, de suivre les artistes et genres populaires, et d'anticiper les futures tendances.

Objectifs du projet

Le projet a pour objectif de :

1. Créer un pipeline pour centraliser les données issues de plusieurs plateformes de streaming musical.
2. Nettoyer, transformer et organiser ces données dans un environnement analytique structuré.
3. Exploiter ces données pour produire des visualisations et des insights sur les artistes, genres et tendances musicales.

Afin de répondre aux questions suivantes :

- Quels artistes et genres dominent les tendances actuelles ?
- Comment ces tendances évoluent-elles au fil du temps ?

Définitions des concepts clés

1. DataLake

Un DataLake est un système de stockage conçu pour accueillir des volumes massifs de données brutes, dans divers formats (CSV, JSON, Parquet, etc.). Contrairement à un datawarehouse, un DataLake conserve les données dans leur forme originale. Il est idéal pour les projets nécessitant une flexibilité dans la collecte et le traitement des données.

2. Bronze-Silver-Gold Architecture

Cette architecture est une méthodologie utilisée pour organiser et transformer les données dans un DataLake :

- Bronze : Stockage brut des données. Les fichiers sont simplement ingérés dans leur format initial (ex. CSV).
- Silver : Transformation et nettoyage. Les données sont enrichies, normalisées et préparées pour l'analyse.
- Gold : Données prêtes à l'exploitation analytique, souvent utilisées pour des tableaux de bord ou des visualisations.

3. DataWarehouse

Un DataWarehouse est une base de données optimisée pour l'analyse. Contrairement au DataLake, il contient des données structurées, nettoyées et prêtes à être utilisées pour des visualisations ou des analyses avancées.

4. Iceberg Tables

Apache Iceberg est un format de table open source conçu pour gérer de grandes quantités de données dans les environnements Big Data. Il permet des opérations efficaces comme le partitionnement, le suivi des versions et les mises à jour incrémentielles. Dans ce projet, Iceberg a été utilisé pour gérer les tables dans le DataLake.

Données associées

Spotify API

Données disponibles : Top tracks, artistes, genres populaires, audio features (tempo, énergie, acoustique, etc.). [Spotify Dataset](#)

YouTube Music

Données disponibles : Vidéos les plus vues, playlists populaires, artistes tendance.

[Youtube Music Dataset](#)

Apple Music [Apple Music Dataset](#)

Format des données

Les données ont été collectées sous format CSV et stockées dans le DataLake de Microsoft Azure.

Architecture utilisée



1. Étape Bronze : Les fichiers CSV bruts des différentes plateformes sont ingérés dans le DataLake.
2. Étape Silver : Les données sont nettoyées, enrichies et organisées dans des tables intermédiaires grâce à des traitements SQL.
3. Étape Gold : Les données finales sont transformées en tables analytiques prêtes pour le DataWarehouse, où elles sont utilisées pour des visualisations et des analyses.

Pipeline de traitement

Technologies utilisées

1. Microsoft Azure : Datalake pour le stockage des fichiers CSV.
2. Databricks : Environnement d'exécution des pipelines ETL (Extract, Transform, Load).
3. Apache Iceberg : Format de table pour optimiser les requêtes et gérer les données dans le DataLake.
4. Utilisation de la librairie Maven. Il suffit de l'installer directement dans votre cluster : `org.apache.iceberg:iceberg-spark-runtime-3.2_2.12:1.3.1`

Étapes de traitement

Étape Bronze : Stockage des Données Brutes

Objectif de l'étape Bronze

L'objectif de cette étape est de centraliser les fichiers CSV bruts dans le DataLake Azure, en préservant leur format original. Ces données serviront de base pour les transformations ultérieures.

Approche utilisée

Les fichiers bruts (CSV) des trois plateformes de streaming (Spotify, YouTube Music et Apple Music) ont été chargés dans le DataLake Azure sans modification. Ils sont organisés dans un répertoire dédié au niveau Bronze de notre architecture.

Résultat de l'étape Bronze

Les données brutes sont accessibles depuis le DataLake dans un format structuré par source et par date d'ingestion (par exemple : `/ds-bronze/spotify/current/`).

Ces fichiers contiennent toutes les informations initiales, telles qu'importées depuis Kaggle.

Étape Silver : Nettoyage et Validation des Données

Objectif de l'étape Silver

L'étape Silver consiste à :

1. Nettoyer les données (suppression des doublons, gestion des valeurs nulles, uniformisation des formats).
2. Valider les relations et référentiels grâce aux fichiers Init et Reject Referential.
3. Créer des tables intermédiaires prêtes pour les analyses exploratoires.

Approche utilisée

Nous avons utilisé des scripts SQL dans Databricks, notamment le fichier

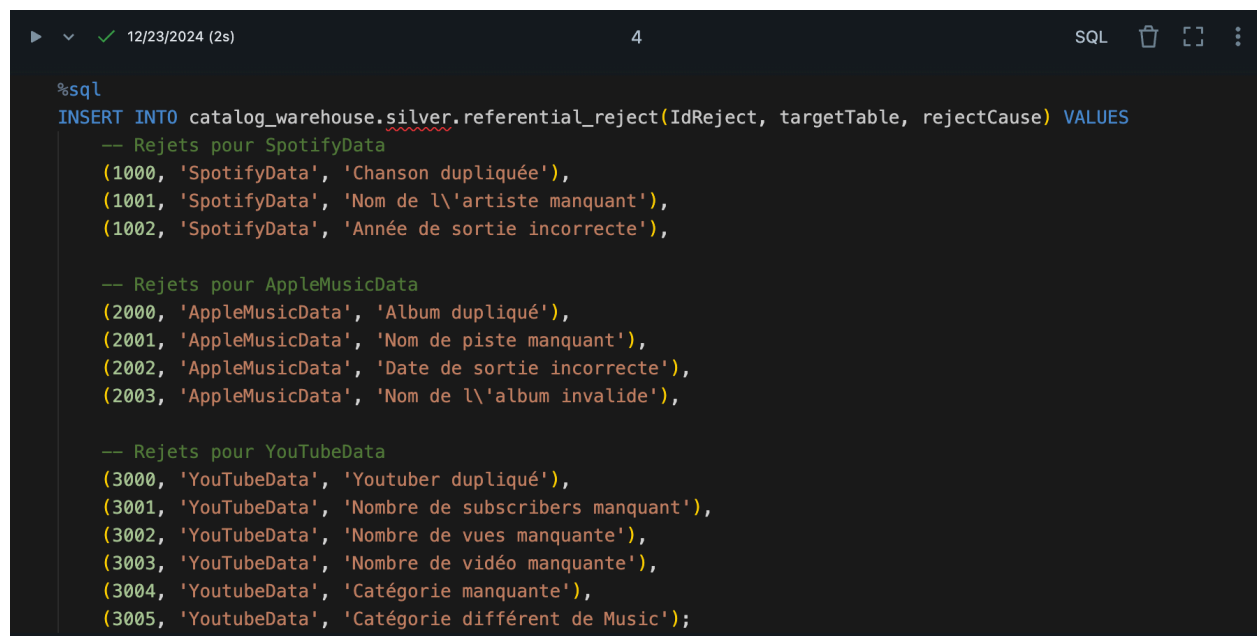
[2.1_Init_Reject_Referential](#), pour :

- Charger les données brutes depuis le DataLake.
- Identifier et rejeter les lignes invalides ou incohérentes.
- Appliquer des transformations nécessaires pour normaliser les données.

Fichiers clés

1. 2.1_Init_Reject_Referential

- Valide l'intégrité des données et détecte les incohérences.
- Stocke les lignes invalides dans une table dédiée.



```
%sql
INSERT INTO catalog_warehouse.silver.referential_reject(IdReject, targetTable, rejectCause) VALUES
-- Rejets pour SpotifyData
(1000, 'SpotifyData', 'Chanson dupliquée'),
(1001, 'SpotifyData', 'Nom de l\'artiste manquant'),
(1002, 'SpotifyData', 'Année de sortie incorrecte'),

-- Rejets pour AppleMusicData
(2000, 'AppleMusicData', 'Album dupliqué'),
(2001, 'AppleMusicData', 'Nom de piste manquant'),
(2002, 'AppleMusicData', 'Date de sortie incorrecte'),
(2003, 'AppleMusicData', 'Nom de l\'album invalide'),

-- Rejets pour YouTubeData
(3000, 'YouTubeData', 'Youtuber dupliqué'),
(3001, 'YouTubeData', 'Nombre de subscribers manquant'),
(3002, 'YouTubeData', 'Nombre de vues manquante'),
(3003, 'YouTubeData', 'Nombre de vidéo manquante'),
(3004, 'YouTubeData', 'Catégorie manquante'),
(3005, 'YouTubeData', 'Catégorie différent de Music');
```

2. Fichiers Ingest

- Gèrent l'ingestion des données nettoyées vers les tables intermédiaires Silver.

Exemple d'ingestion des rejets pour Spotify

```
12/23/2024 (28s) 6 SQL

INSERT INTO spotify_reject_catalog.silver.spotify_reject
WITH
-- Rejets pour track_name NULL (1000)
spotify_reject_1000 AS (
    SELECT
        1000 AS IdCauseRejet,
        CURRENT_DATE AS insertionDate,
        ROWID AS rowNumber,
        TRY_CAST(track_name AS VARCHAR(255)) AS track_name,
        TRY_CAST(`artist(s)_name` AS VARCHAR(255)) AS artist_name,
        TRY_CAST(released_year AS INT) AS released_year,
        TRY_CAST(released_month AS INT) AS released_month,
        TRY_CAST(released_day AS INT) AS released_day,
        TRY_CAST(streams AS BIGINT) AS streams
    FROM catalog_warehouse.silver.spotify_deduplicated
    WHERE RowId > 1
),
-- Rejets pour artist_name NULL (1001)
spotify_reject_1001 AS (
    SELECT
        1001 AS IdCauseRejet,
        CURRENT_DATE AS insertionDate,
        ROWID AS rowNumber,
        TRY_CAST(track_name AS VARCHAR(255)) AS track_name,
        TRY_CAST(`artist(s)_name` AS VARCHAR(255)) AS artist_name,
        TRY_CAST(released_year AS INT) AS released_year,
        TRY_CAST(released_month AS INT) AS released_month,
        TRY_CAST(released_day AS INT) AS released_day,
        TRY_CAST(streams AS BIGINT) AS streams
    FROM catalog_warehouse.silver.spotify_deduplicated
    WHERE track_name IS NULL
```

Étape Gold : Tables Analytiques

Objectif de l'étape Gold

L'objectif principal de cette étape est de transformer les données nettoyées et validées (Silver) en un modèle analytique structuré et optimisé pour les visualisations et analyses approfondies. Les tables issues de cette étape constituent le **DataWarehouse**.

Approche utilisée : Modèle en Étoile

Pour cette étape, nous avons conçu un **modèle en étoile**. Ce modèle organise les données autour d'une **table de faits** principale (fact table) et de plusieurs **tables de dimensions** connectées par des clés étrangères.

Le modèle en étoile permet :

1. **Une organisation efficace des données** pour des requêtes analytiques rapides.
2. **Une navigation intuitive** entre les dimensions (artistes, albums, genres, etc.) et les faits (streams, vues, etc.).

Structure du modèle en étoile

Le modèle comprend les éléments suivants :

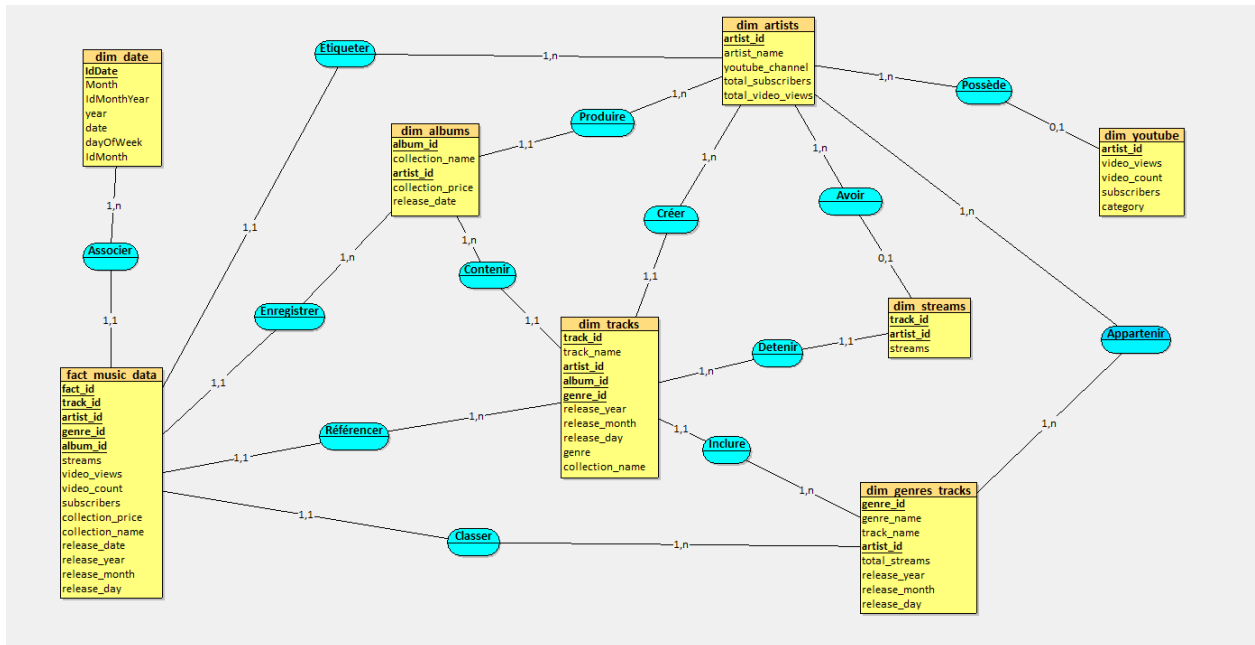
Table de faits : `fact_music_data`

La table de faits centralise les métriques analytiques comme le nombre de streams, de vues YouTube ou d'abonnés. Elle est connectée aux dimensions pour fournir un contexte détaillé aux analyses.

Tables de dimensions :

1. `dim_date` : Gère les informations temporelles (année, mois, jour).
2. `dim_artists` : Stocke les informations des artistes, comme leur nom et leur popularité sur YouTube.
3. `dim_albums` : Contient les données des albums (nom, artiste associé, date de sortie, prix).
4. `dim_genres_tracks` : Regroupe les genres musicaux, les morceaux associés et les streams par genre.
5. `dim_tracks` : Décrit chaque morceau (titre, genre, album, artiste).
6. `dim_streams` : Suit le nombre de streams des morceaux par artiste.
7. `dim_youtube` : Fournit des métriques YouTube (vues, vidéos, abonnés, catégorie).

Modèle Conceptuel de Données



Les relations entre les entités :

- **Étiqueter :**

Associe les données temporelles (**dim_date**) aux analyses dans la table de faits (**fact_music_data**).

- **Associer :**

Lie la table de faits (**fact_music_data**) à des morceaux spécifiques (**dim_tracks**).

- **Produire :**

Connecte les artistes (`dim_artists`) aux albums qu'ils ont produits (`dim_albums`).

- **Créer :**

Indique quel artiste a créé un morceau donné dans la table `dim_tracks`.

- **Contenir :**

Indique quels morceaux sont inclus dans un album spécifique.

- **Inclure et Détenir :**

Relient les morceaux (`dim_tracks`) à leurs genres respectifs (`dim_genres_tracks`), ainsi qu'aux statistiques de diffusion (`dim_streams`).

- **Appartenir :**

Connecte les statistiques YouTube d'un artiste avec les informations détaillées dans `dim_youtube`.

- **Enregistrer et Référencer :**

Relié à la table de faits, ces relations permettent de synthétiser les données analytiques entre les morceaux, albums et genres.

Utilité de ce modèle

Ce modèle structure les données pour permettre une analyse avancée de l'industrie musicale. Par exemple :

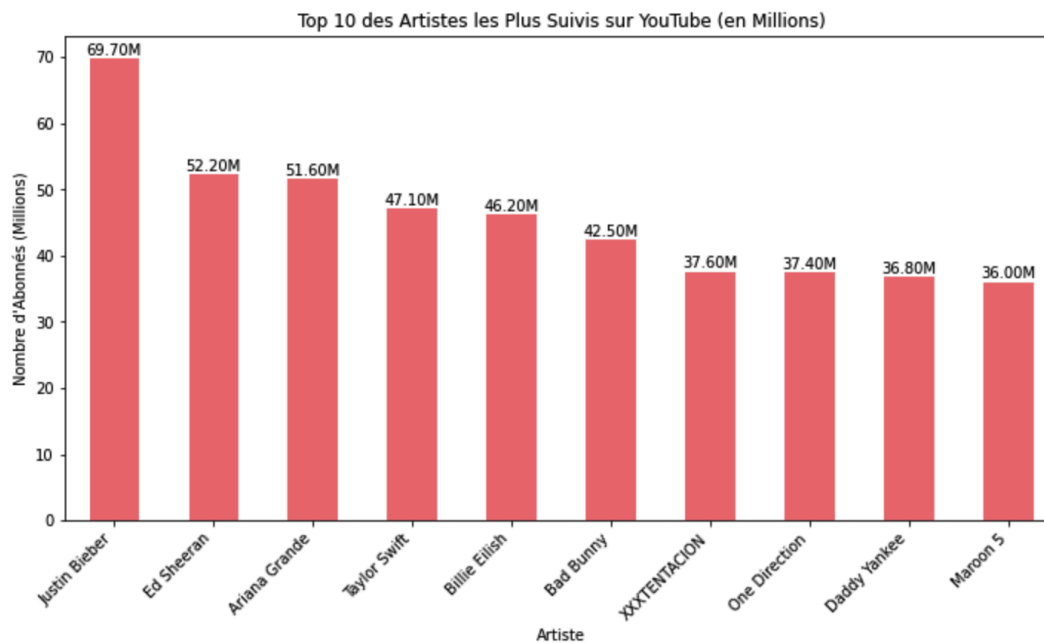
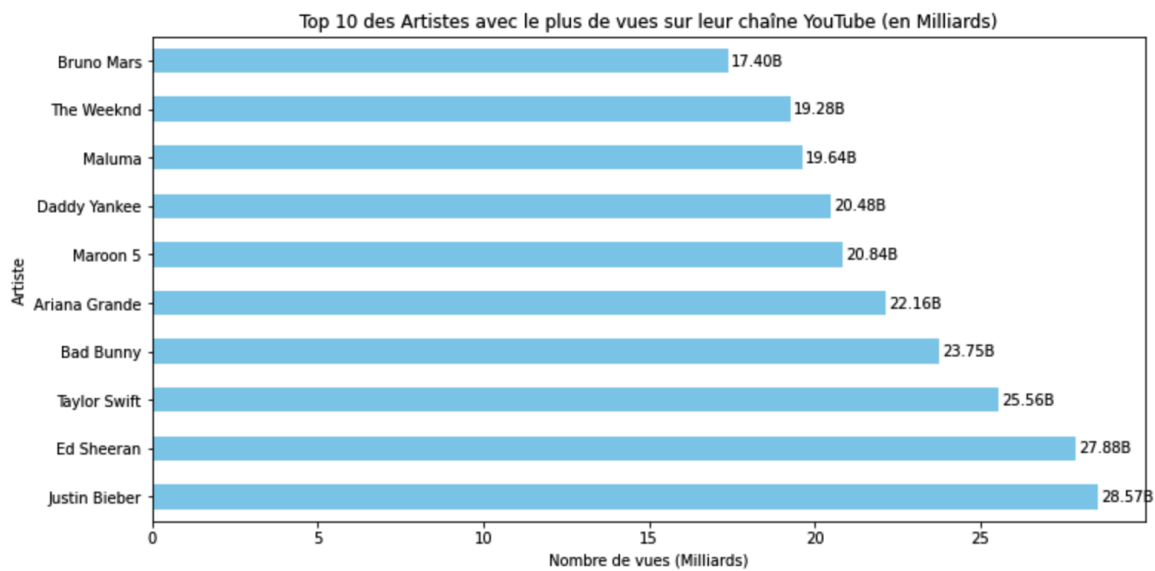
- **Analyse temporelle :** Étudier les tendances par année, mois ou jour.

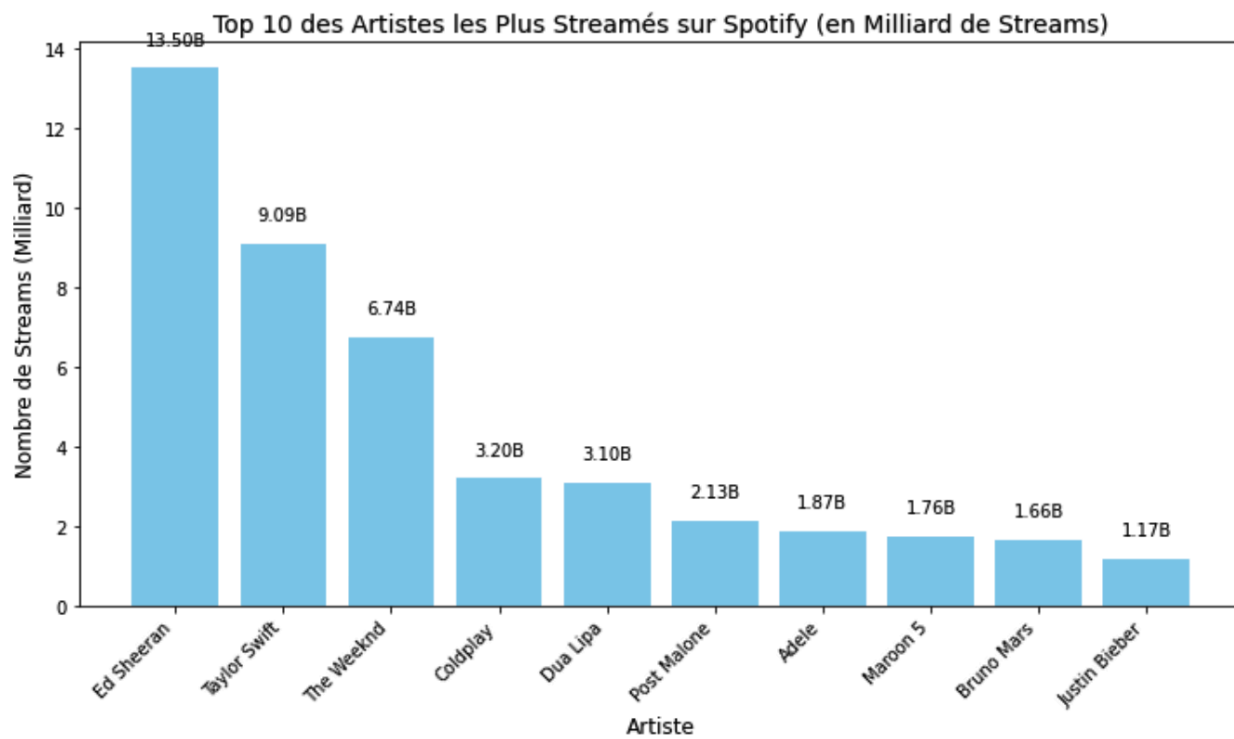
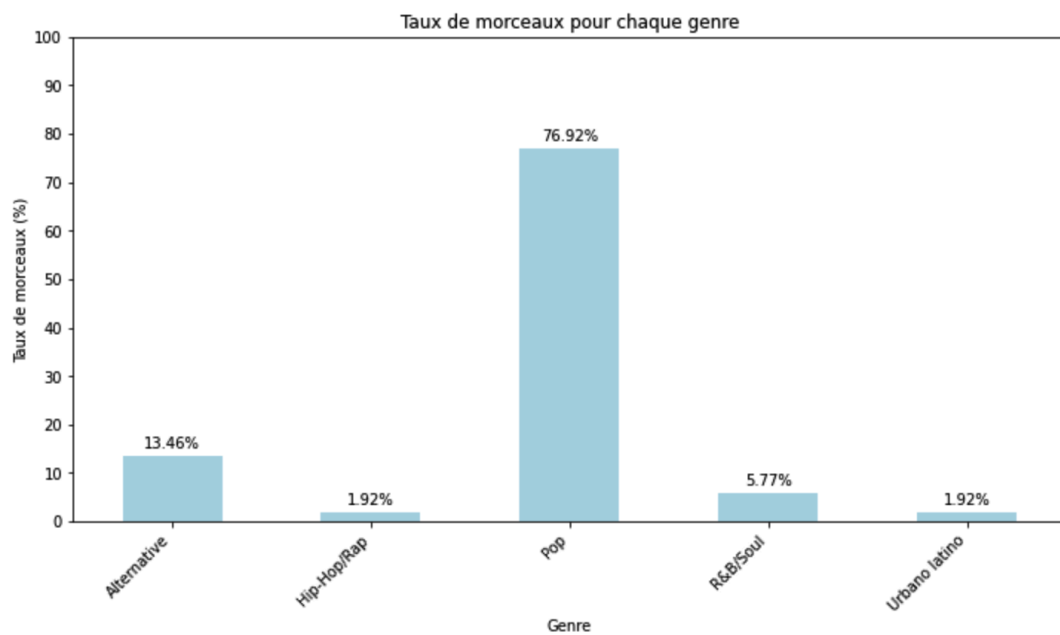
- **Statistiques des artistes** : Suivre les performances YouTube, les streams et les ventes d'albums/morceaux.
- **Relations entre entités** : Comprendre les contributions d'artistes à des albums ou des morceaux spécifiques, ainsi que l'impact de certains genres sur les performances.

Visualisation et Résultats

Les données consolidées dans le modèle en étoile ont été utilisées pour générer des graphiques et des tableaux de bord, tels que :

1. Diagramme des artistes les plus vus sur Youtube.
2. Diagramme des artistes les plus suivis sur Youtube.
3. Diagramme du taux de morceaux par genre.
4. Diagramme des 10 artistes les plus streamés sur Spotify.





Conclusion

Ce projet d'analyse des tendances musicales sur les plateformes de streaming a démontré l'efficacité d'une architecture **Bronze-Silver-Gold** pour centraliser, structurer et exploiter des données volumineuses et hétérogènes. Grâce à l'intégration des données provenant de Spotify, YouTube Music et Apple Music, il a été possible d'identifier les artistes, genres et morceaux les plus populaires, ainsi que d'anticiper les évolutions des tendances musicales.

Les visualisations générées, telles que les classements des artistes les plus suivis ou la répartition des genres musicaux, offrent des insights précieux pour les professionnels de l'industrie musicale et les plateformes de streaming. Ce travail met également en lumière l'importance des technologies Big Data, comme **Databricks**, **Microsoft Azure**, et **Apache Iceberg**, dans l'analyse et la gestion des données complexes.

Enfin, les résultats de ce projet ouvrent la voie à des développements futurs, tels que l'intégration de données supplémentaires (réseaux sociaux, événements en direct) et l'utilisation d'algorithmes de machine learning pour des prévisions encore plus précises des tendances musicales.