

CER 1.0

Lundi 10 Juin 2025

# BACKEND ET DATA ENGINEERING



**base** for music



BENLKHIR MARWAN

1re Année cycle Ingénieur

# Table des matières

<b>Mots-clés</b>	<b>1</b>
<b>Mots à définir</b>	<b>1</b>
<b>Contexte</b>	<b>1</b>
<b>Problématique</b>	<b>1</b>
<b>Livrables</b>	<b>2</b>
<b>Besoins / Contraintes</b>	<b>2</b>
<b>Plan d'action</b>	<b>2</b>
<b>Planning réel</b>	<b>2</b>
<b>Généralisation</b>	<b>2</b>
<b>Ressources</b>	<b>3</b>
<b>Réalisation du plan d'action</b>	<b>3</b>
<b>Bilan personnel</b>	<b>8</b>
<b>Remarques</b>	<b>8</b>
<b>Pistes d'amélioration</b>	<b>8</b>

## Mots-clés

- Backend
- Data Engineering
- Python
- Github
- Jupyter Notebook
- Anaconda
- PostgreSQL
- Diagramme
- Statistique

## Mots à définir

- Diagramme relationnel (ERD) : C'est un diagramme de relations d'entités, une représentation visuelle de la structure d'une base de données

## Contexte

Le test technique consiste en l'analyse et la modélisation des données relatives à l'audience du catalogue musical donné

## Problématique

Comment traiter les données et proposer une modélisation et une analyse pertinente pour une approche data-driven ?

## Livrables

- Schéma de BDD (ERD)
- Compte rendu et analyse
- Notebook Jupyter

## Besoins / Contraintes

- Pandas
- Python
- 7 jours
- PostgreSQL

## Hypothèses

- Les bibliothèques Python offrent-elles un vrai gain de temps ? Oui même s'il faudra privilégier des requêtes SQL et une base de données car elles sont plus optimisées pour ce type d'opération.
- La source de données propose-t-elle vraiment des données brutes ? Oui, les données sont à réorganiser.
- Le diagramme relationnel ERD est similaire au MCD, MLD et MPD ? Ils sont similaires à quelques notions près. Les MCD, MLD et MPD sont, toutefois, plus précis.

## Plan d'action

- I. Traitement des données
  - A. Analyse de la structure des données brutes
- II. BDD
  - A. Proposer un diagramme relationnel
  - B. Implémentation sur PostgreSQL
- III. Modélisation et enrichissement
  - A. Proposer des visualisations, statistiques et métriques
  - B. Proposer des moyennes et corrélations
- IV. Analyse
  - A. Interprétation des résultats

## Planning réel

Ouvrir le fichier Planning réel.pdf

## Généralisation

Science des données

## Ressources

### Outils :

- [Eraser.io](https://eraser.io/) : Conception et diagramme
- Jupyter Notebook
- Anaconda
- Github
- VSCode

- json format
- csv beautify
- PostgreSQL
- pgAdmin 4
- [sqlformat.org](https://sqlformat.org) : Formateur de requête SQL

Liens :

- [stackoverflow.com](https://stackoverflow.com) : Forum programmation
- [pandas.pydata.org](https://pandas.pydata.org) : Documentation Pandas
- [numpy.org](https://numpy.org) : Documentation numpy
- [claude.ai](https://claude.ai) : Chatbot
- [sqlalchemy.org](https://sqlalchemy.org) : Documentation SQLAlchemy
- [matplotlib.org](https://matplotlib.org) : Documentation Matplotlib

## Réalisation du plan d'action

### **I. Traitement des données**

#### **A. Analyse de la structure des données brutes**

Cette première étape consiste à comprendre, analyser les données brutes et observer leur structure dans un environnement test afin de conserver une version saine. La première ligne donne des informations précieuses sur la nature des données. Le fichier est composé des colonnes :

- song\_id
- title
- artist
- release\_date
- image\_url
- timeSeries
- summaries'

L'extension "rainbow csv" disponible sur Visual Studio Code permet de manière graphique d'avoir un aperçu de la structure en mettant en couleur chaque colonne d'un fichier csv. Notons qu'il existe une limite de ligne pour appliquer la couleur pour des raisons de performance. La modification du paramètre "max tokenization line length" résout ce problème.

Les colonnes “timeSeries” et “summaries” sont des tableaux, mais leur format est encore inconnu bien qu’il s’agit sans doute de JSON. Dans le Notebook, un test de chargement en json sera effectué (Chapitre “Analysis and observation” du Notebook). Il s’agit bien de données JSON. En isolant dans un fichier au format JSON la cellule “timeSeries” d’une ligne et avec l’extension Visual Studio Code “beautify json”, sa structure prend forme avec une indentation correcte. La même méthode sera appliquée à une cellule “summaries”. Ces étapes nous permettront de concevoir l’ERD.

## B. Nettoyage des données

Grâce à l’analyse, l’étape de nettoyage consiste soit à supprimer, soit à réparer des lignes qui pourraient poser problème. La dernière ligne du fichier est incomplète. Elle se finit par “spotify-st” et n’est pas conforme à la structure du fichier. L’extension “rainbow csv” rend très visuelle la détection d’erreur de structure.

La dernière ligne n'étant ni complète, ni correctement structurée, la meilleure décision est de la supprimer. Cette version du dataset (clean-dataset.csv) est disponible via le lien :

<https://drive.google.com/file/d/1yIoZSmt1HPfkknfw-8MEm5RymT2RAZma/view>

Elle est nécessaire à l'exécution du Notebook.

Ensuite, pour chaque ligne, l'index et de multiples tab sont interprétés comme faisant partie du "song\_id". Alors la colonne sera traitée pour enlever ces erreurs d'interprétation (Chapitre "Cleaning" du Notebook).

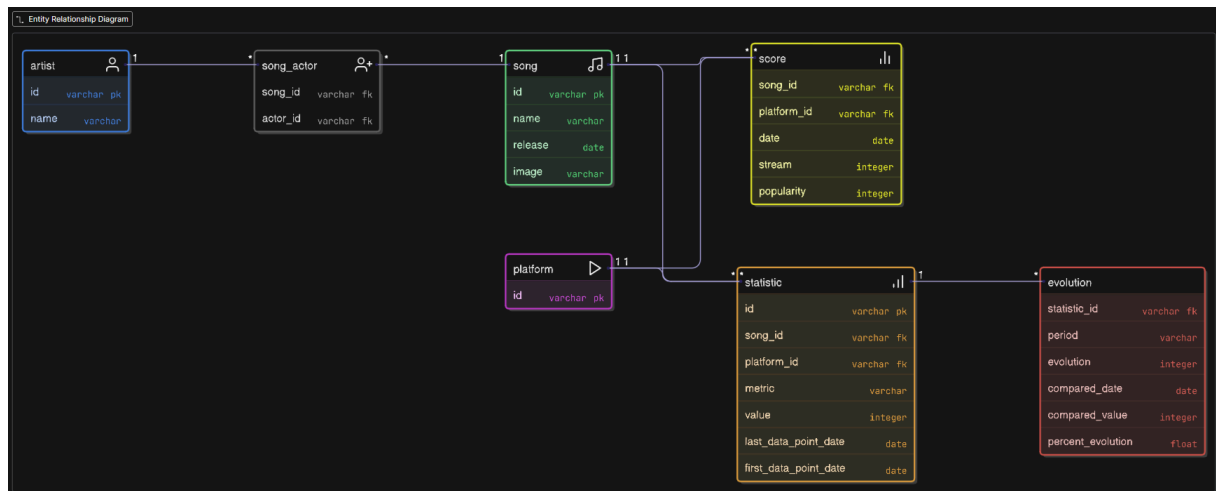
## II. Base de données

### A. Proposer un diagramme relationnel

L'analyse a permis de déterminer la nature et le type de chaque colonne du dataset. Ces informations vont déterminer les paramètres des attributs de L'ERD. La création d'un ID est obligatoire pour la table "artist". En revanche, le nom de la plateforme sera utilisé en tant qu'ID. Des tables ont une clé primaire, d'autres utilisent des clés étrangères et certaines les deux. La colonne "summaries" sera divisée en deux tables : "statistics" et "evolution". "statistics" sera construite à partir de toutes les informations de "summaries" sauf celles dépendantes de "period". La table "evolution" sera composée de l'ID de la statistique qui lui correspond et de toutes les mesures sur la période. Les attributs et noms de table sont uniformisés sans majuscule et les noms composés séparés par un underscore. L'ERD est conçu pour être ouvert à de futures augmentations de données telles que des plateformes, différentes statistiques. Il est aussi ouvert à de futures améliorations de données telles que l'ajout d'acteurs sur un morceau grâce à la table "song\_actor".

Parce qu'une image vaut mille mots, voici l'ERD, disponible sur le lien suivant (un compte est nécessaire) : <https://app.eraser.io/workspace/zGdBMSC0RzIjWtiEtCWE?origin=share>

Sinon, le voici en image accompagné de la représentation des liens entre les tables et d'une légende.



```

artist < song_actor

song_actor > song

platform < statistic

platform < score

song < score

song < statistic

statistic < evolution

```

Légende :

- < : One to many
- pk : primary key
- fk : foreign key

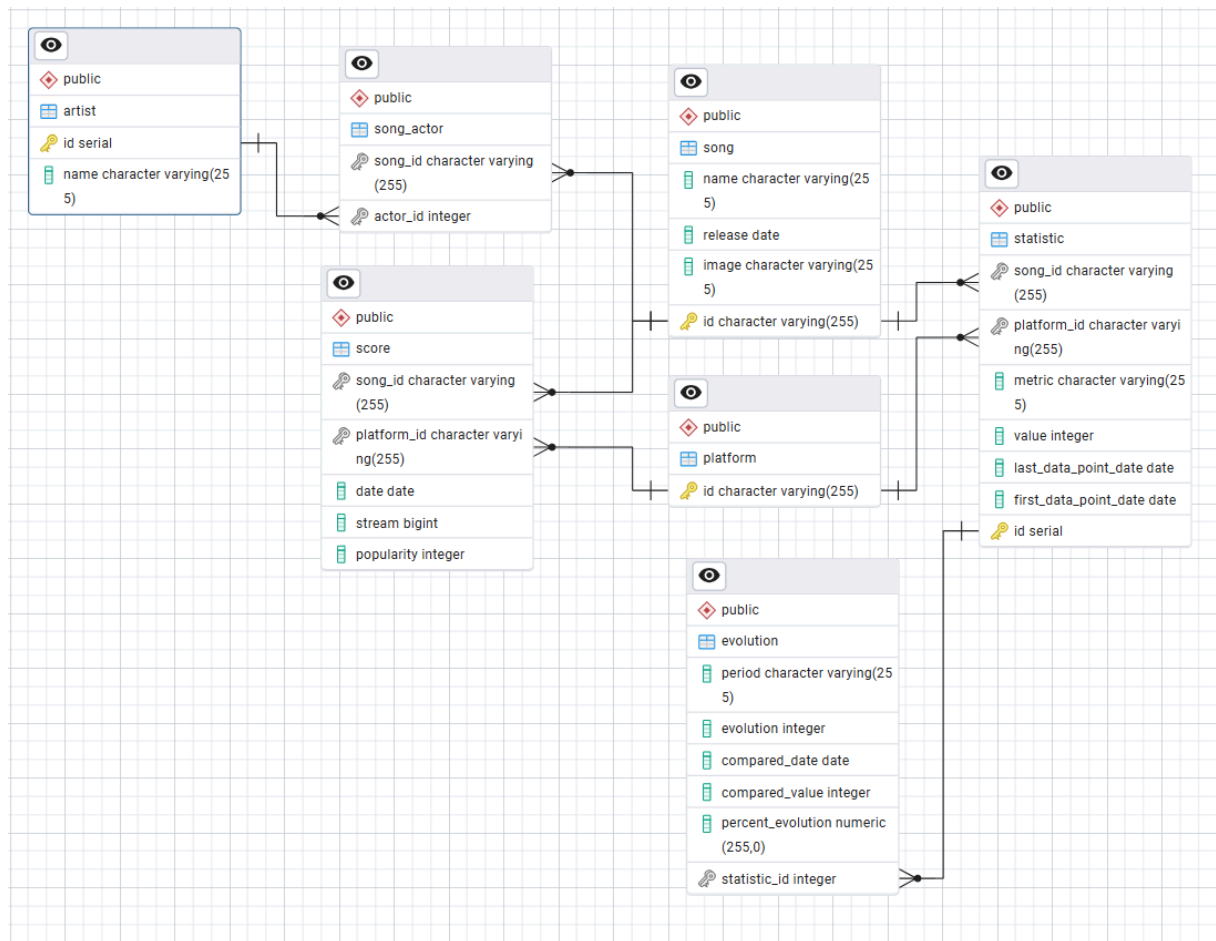
## B. Création de la base sur PostgreSQL

Le dump de la structure et ses contraintes est disponible dans le fichier dump\_pre-data.sql .

Il s'agit d'un dump pre-data (la structure sans donnée) et post-data (avec les contraintes de clé étrangères notamment) au format plain (SQL lisible et non compressé) en UTF8, sans propriétaire, ni privilège.

Voici l'ERD composé de ses précisions comme présenté dans pgAdmin.





### C. Insertion des données dans la base

Cette étape consiste à manipuler “clean-dataset.csv” avec du code python en utilisant les librairies pandas, numpy et json pour leur donner la forme souhaitée avant une insertion dans la base de données avec les librairies pycpg2 et sqlalchemy. Le code pour y parvenir est présent dans le Notebook au chapitre “Data insertion”. Chaque table nécessite son code afin de préparer les données comme il convient dans l’ERD. Chaque code est commenté. Il l’est sur l’ensemble du Notebook et en anglais. La manipulation est très délicate et demande beaucoup de précaution pour ne pas fausser les données.

Le dump complet de la base de données avec les données insérées est disponible dans le lien suivant :

[https://drive.google.com/file/d/1YlarfbUwPvjpEI9rFlk7qcw06PMRZuIN/view?usp=drive\\_link](https://drive.google.com/file/d/1YlarfbUwPvjpEI9rFlk7qcw06PMRZuIN/view?usp=drive_link)

Il s’agit d’un dump pre-data (la structure sans donnée), data (avec les données) et post-data (avec les contraintes de clé étrangères notamment) au format plain (SQL lisible et non compressé) en UTF8, sans propriétaire, ni privilège.

### III. Modélisation et enrichissement

Toutes les informations suivantes sont présentes sur le Notebook. Sinon un export du Notebook est disponible dans le fichier Export\_Notebook.html .

Ces informations ont été produites par des requêtes SQL car les base de données sont spécialisées pour les opérations de sélection et autres, atteignant des performances inégalables.

## **A. Proposer des visualisations, statistiques et métriques**

### **1. Tendances générales**

- Graphique linéaire des streams dans le temps
- Graphique linéaire des streams de 2022 à 2024
- Graphique linéaire des sorties dans le temps
- Nuage de points des streams en fonction de la popularité
- Tracé hexagonal
- Graphique linéaire de la moyenne de streams en fonction du nombre d'artistes présents un morceau

### **2. Les tops**

- Graphique à barres des 5 artistes les plus streamés
- Graphique à barres des 5 artistes les plus populaires

La popularité moyenne a été utilisée.

- Graphique à barres des 5 artistes avec le plus de morceaux
- Graphique à barres des 5 artistes avec le plus de collaborations
- Graphique à barres des 5 morceaux les plus streamés
- Graphique à barres des 5 morceaux les plus populaires

### **3. Étude particulière**

#### **a) Par son**

- Graphique linéaire d'un morceau
- Graphique linéaire de la popularité d'un morceau

#### **b) Par artiste**

- Graphique linéaire des streams des morceaux d'un artiste

## **B. Proposer des corrélations et moyennes**

### **1. Statistiques générales**

- Nombre de morceaux
- Moyenne de streams par morceau
- Écart-type de streams
- Nombre de streams minimum
- 25% de streams
- 50% de streams
- 75% de streams
- Nombre de streams maximum
- Médiane des streams
- Boîtes à moustaches des streams
- Moyenne de sons par artiste
- Médiane de sons par artiste
- Moyenne de streams par artiste

- Médiane de streams par artiste
- Moyenne d'artiste par morceau
- Médiane d'artiste par morceau

## **2. Corrélations**

- Entre le nombre de streams et popularité
- Entre le nombre de streams et le nombre de morceaux d'un artiste
- Entre le nombre de streams et le nombre d'artiste sur le morceau

## **IV. Interprétation des résultats**

L'interprétation n'est pas demandée dans l'exercice mais me permet de revenir sur certains points et d'ajouter des précisions.

Le nombre de streams augmente considérablement à partir de mai 2024 possiblement dû à l'arrivée de plus d'artistes chez Base for Music. Peut-être aussi par le nombre de morceaux sortis dans la même période.

La majorité des morceaux vont de 0 à 500 millions de streams avec une popularité extrêmement diverse. L'analyse de densité nous révèle que la grande majorité des morceaux sont peu streamés, peu populaires. 75% des morceaux sont en dessous de 700 000 streams. On observe une forte discontinuité de streams entre 1 milliard et 1,5 milliard, puis 2 milliards et 3,5 milliards. On distingue alors 3 principales catégories de streams. Les streams sont très hétérogènes. Les morceaux extrêmement streamés sont rares, mais généralement avec une popularité élevée. La corrélation entre les streams et la popularité d'un morceau est de 0.39 qui s'explique par le fait qu'un morceau très streamé est populaire et que la réciproque ne fonctionne pas. La popularité est déterminée par des facteurs complexes tels que les tendances récentes, les algorithmes de recommandation, les nouveautés.

La partie concernant les tops permet d'avoir des informations concernant le catalogue ou du moins ces données. Par exemple, Bad bunny et les morceaux de son dernier album ont vraiment atteint des chiffres impressionnants. D'ailleurs, ces morceaux à grand nombre de streams impactent fortement les différentes statistiques de moyenne. Dans ce contexte la médiane permet d'avoir un second point de vue. Le nombre de stream moyen est d'environ 10 millions et la médiane d'environ 56 000. Sur ces données, les artistes les plus populaires ne sont pas ceux avec le plus de streams. Ce top prend en compte les morceaux disponibles dans les données et est calculé avec la popularité moyenne.

Les featurings ne sont pas forcément à privilégier alors que le nombre moyen d'artistes sur un morceau est de 1.66 et que la corrélation montre qu'il n'y a aucun rapport sur son impact sur les streams. Toutefois, les duos et les featurings impliquant deux artistes enregistrent la meilleure moyenne de streams

A sa mesure, la productivité des artistes peut avoir un impact sur les scores enregistrés.

Enfin, il est important de garder en tête que ces interprétations sont relatives aux données

## [Bilan personnel](#)

J'ai adoré ce test technique. Je dois avouer que c'est le premier du genre que j'ai eu à réaliser pour une candidature. Il me conforte dans l'idée que je veux faire de ces compétences mon métier et que j'ai une chance incroyable de les employer dans l'industrie musicale.

## Questions

Durant la réalisation de ce projet, je me suis heurté à plusieurs interrogations et je souhaiterais mieux comprendre leur origine ou logique :

- J'ai trouvé des doublons sur certains "song\_id". S'agit-il d'une particularité spécifique ? Plusieurs versions ou relevés ?
- Certains relevés de stream et popularité sont "null" ? Est-ce dû à un manque de données ou sont-ils à considérer comme 0 ?
- J'ai également observé que les valeurs de streams ne sont pas systématiquement croissantes dans le temps. Cela peut-il être lié à des corrections ou une autre logique d'enregistrement ?
- Quelques dates sont très anciennes, parfois datées de l'an 1. S'agit-il de valeurs par défaut ?
- Je n'ai malheureusement pas réussi à différencier les duos des featurings. Existait-il dans ce test technique un moyen de les distinguer ?
- Je n'ai pas pu regrouper les occurrences multiples d'un même artiste sous des noms légèrement différents. Est-ce que des informations et données m'auraient permis de les rassembler ?

## Pistes d'amélioration

Plusieurs pistes d'amélioration sont à envisager.

D'abord, l'optimisation des requêtes SQL des parties modélisation et enrichissement. Ces requêtes sont une première version. Pour les rendre plus performantes il est nécessaire de placer les opérations de sélection et de limitation au plus bas niveau de l'imbrication, puis, de faire les opérations de jointure au plus haut niveau. Enfin d'imbriquer le plus possible ces opérations.

Deuxièmement, utiliser Luna Modeler pour construire un ERD. Ce dernier est un outil intuitif et graphique dédié à la modélisation de données pour les bases de données relationnelles.

Troisièmement, modifier la table "song\_actor" en définissant le rôle tels que auteur, compositeur, interprète, topliner, ingénieur du son, ingénieur du mixage, ingénieur du mastering.

Dernièrement, trouver des manières pertinentes d'utiliser la boîte à moustaches, le KDE plot, l'area plot, le diagramme circulaire, le nuage de points et le hexbin plot. Ce sont des moyens efficaces de comprendre des données complexes.