

Carnet de bord de traitement de données

projet BiblioMoMA

Bassaisteguy Sidonie

Batard Camille

Benguigui Juliette

Dulaurent Adonis

28 février 2025

Table des matières

Introduction	4
1 Décembre	5
1.1 Semaine 2-8 décembre 2024	5
1.1.1 Jalon	5
1.1.2 Recherche	5
1.2 Semaine 9 - 14 décembre 2024	5
1.2.1 Recherche	5
1.3 Semaine 16 - 22 décembre 2024	5
1.3.1 Jalon	5
1.3.2 Recherche et livrable	6
1.4 Vacances de Noël 23 décembre 2024 - 5 janvier 2025	6
1.4.1 Recherche	6
1.4.2 Utilisation	7
2 Janvier	7
2.1 Semaine 6 - 12 Janvier 2024	7
2.1.1 Recherche	7
2.1.2 La table Artistes	7
2.1.3 La table Artworks	8
2.1.4 Recherche	8
2.2 Semaine 20 - 26 janvier 2025	8
2.2.1 Requêtage	8
2.2.2 Semaine 27 janvier - 2 février 2025	10
3 Février	10
3.1 Semaine 3 - 9 février 2025	10
3.1.1 Compte rendues de la réunion interne du 8 février 2025	10
3.1.2 Préparation	11
3.1.3 Création des tables Genres et Artists_Genres	12
3.2 Semaine 10 - 16 février 2025	13
3.2.1 Jalon	13
3.2.2 Requêtage	13
3.3 Semaine 17 - 23 février 2025	15
3.3.1 Requêtage	15
3.3.2 Préparation :	16

3.3.3	Création des différentes tables à partir du jeu de données récupéré sur OpenLibrary :	17
3.3.4	Exportation :	17
3.4	Semaine 24 février - 2 mars 2025	17
3.4.1	Documentation	17
Conclusion		18

Introduction

BiblioMoMA est un projet de visite bibliographique à travers les collections du Museum of Modern Art. BiblioMoMA permet à l'utilisateur de visualiser des informations relatives à la vie et au travail d'un artiste lorsqu'il ou elle recherche ou clique sur une œuvre, ainsi qu'une courte bibliographie académique. Simple d'accès, BiblioMoMA se veut ouvert à tous les publics, du collégien au chercheur en histoire de l'art.

Une partie de ce projet concerne l'UE 3 : il est attendu la réalisation d'un traitement de données et la production d'une datavisualisation. Pour cela, nous avons utilisé les jeux de données du MoMA que nous avons enrichis avec Wikidata.

Ce carnet de bord est accompagné de l'export du flux Dataiku, des fichiers CSV retravaillés et du modèle logique de données (MCD) de la base qui sera utilisée dans l'application. Il est organisé de manière chrono-thématique, ce qui permet un suivi détaillé et organisé des activités. Il est divisé en trois grandes périodes correspondant aux mois de décembre, janvier et février. Chaque mois est ensuite découpé en semaines numérotées. Ces semaines regroupent des entrées thématiques : recherche, requêtage, préparation, exportation, utilisation et documentation. Certains jalons clés, comme les réunions, sont intégrés pour marquer les moments de synthèse et de validation des travaux réalisés.

1 Décembre

1.1 Semaine 2-8 décembre 2024

1.1.1 Jalon

Le mercredi 4 décembre, lors du cours de Modélisation et Requêtage avec Mme Bermès, nous avons découvert les groupes et certains des jeux de données proposés.

1.1.2 Recherche

Le projet est encore au stade de conception. Nous évoquons plusieurs projets qui nous tentent. Le MoMA attire déjà notre attention, mais nous sommes également intéressés par un projet autour des bibliothèques parisiennes.

1.2 Semaine 9 - 14 décembre 2024

1.2.1 Recherche

Nous envisageons à ce stade une étude sur les fréquentations des salles de cinéma par secteur ou par type de film pour aider les médiathèques à acheter des films pertinents pour leurs collections. Devant la difficulté à trouver des jeux de données clairs (le CNC communique notamment les siens par fichiers Excel non conformes), et après avoir discuté de l'intérêt de travailler sur les collections du MoMA, nous avons abandonné les salles de cinéma pour nous concentrer sur le musée new-yorkais.

L'avantage de celui-ci est qu'il propose des données conformes sur ses collections en accès libre sur GitHub. Nous évoquons l'idée de lier les données du MoMA à OpenAlex, un catalogue numérique d'articles scientifiques, d'auteurs, et d'institutions en open access afin de créer une application qui ressemblerait à une « promenade bibliographique » dans le musée. Nous envisageons déjà l'idée d'offrir une bibliographie sous chaque œuvre à son propos et/ou en lien avec son auteur et le mouvement, s'il y en a, où elle est attachée. Reste à savoir comment concrétiser toutes ces idées.

1.3 Semaine 16 - 22 décembre 2024

1.3.1 Jalon

À l'approche de la date de rendu de la note de cadrage le 20 décembre, nous organisons une nouvelle réunion.

— Lundi 16 décembre : réunion de cadrage et de lancement

- Nous discutons de la rédaction de la note de cadrage, de la répartition des tâches et du planning.
- Le projet est officiellement lancé : nous partons sur la promenade bibliographique.

1.3.2 Recherche et livrable

Nous rédigeons la note de cadrage détaillant le jeu de données choisi et l’objectif de réalisation, l’équipe avec les rôles de chacun, et l’organisation sous la forme d’un planning prévisionnel.

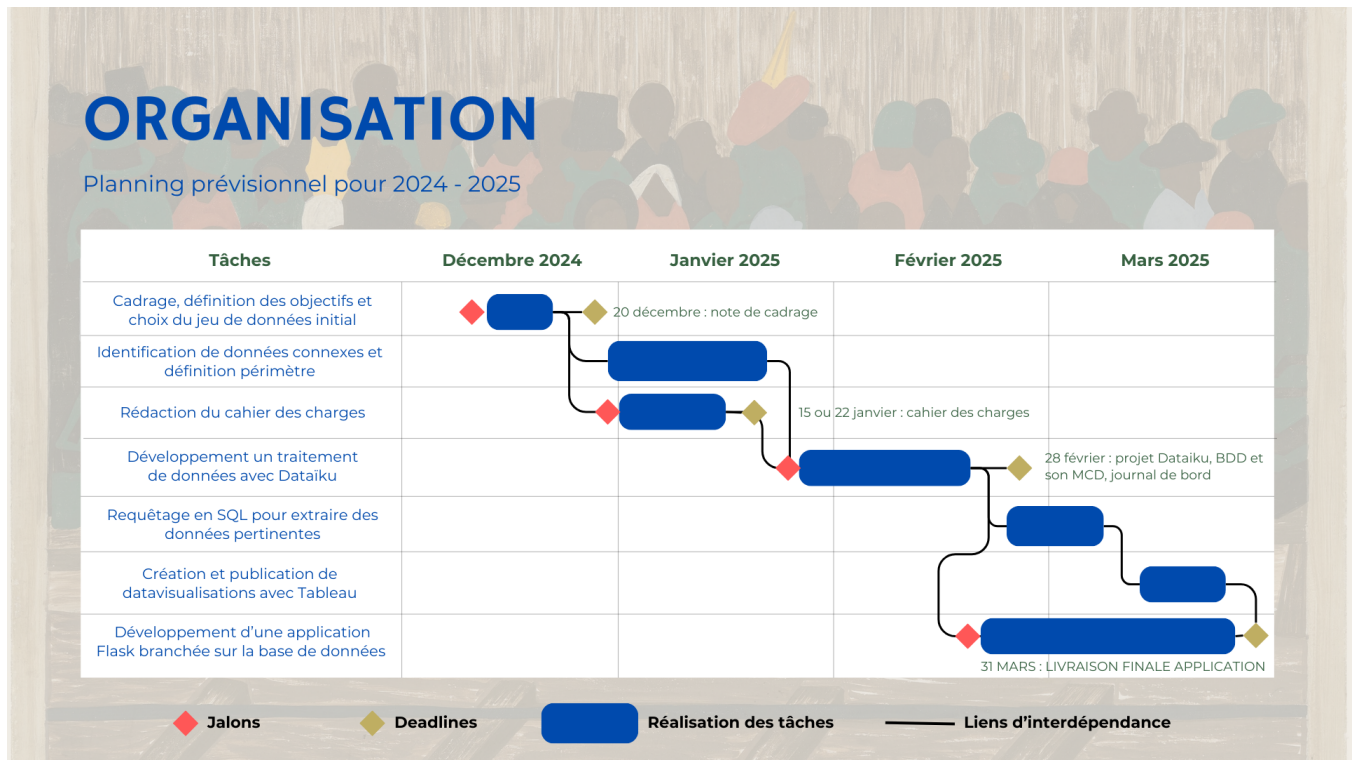


FIGURE 1 – Extrait de la note de cadrage visualisant notre planning

1.4 Vacances de Noël 23 décembre 2024 - 5 janvier 2025

1.4.1 Recherche

Durant ces deux semaines, nous avons réfléchi à quel requêtage pertinent nous pouvions faire de l’API d’OpenAlex (mais cela concerne l’application Flask et ne sera donc pas développé ici) et comment nous pouvons enrichir les jeux de données du MoMA grâce à Wikidata.

L’idée était d’obtenir, grâce à l’API Wikidata, des informations pour enrichir les œuvres et / ou artistes du MoMA grâce à des catégories “englobantes” permettant de les regrouper par groupe et d’avoir un nouvel axe de tri. Wikidata propose les relations mouvement (P135) et genre (P136) qui permettent d’obtenir ces informations et nous choisissons donc cette méthode d’enrichissement.

1.4.2 Utilisation

Cet enrichissement nous permet d'envisager également plusieurs visualisations de données comme :

- Courbes pleines superposées pour visualiser la chronologie des mouvements artistiques et voir lesquels ont dominé à différentes périodes.
- Diagramme radar / ou *heat map* pour visualiser les différentes affiliations des genres et mouvements par artistes et comprendre les liens entre eux

2 Janvier

2.1 Semaine 6 - 12 Janvier 2024

2.1.1 Recherche

Le Museum of Modern Art (MoMA) propose sur son GitHub deux jeux de données : l'un concernant ses œuvres et l'autre ses artistes. Ces fichiers sont disponibles aux formats CSV (encodés en UTF-8) et JSON. Le MoMA précise que ces données n'ont pas été vérifiées par des conservateurs et sont mises à disposition "en l'état" à des fins de recherche, avec des mises à jour hebdomadaires. Les données ont été téléchargées le 12 janvier, et nous travaillons donc avec les données en date du 07/01/2025.

2.1.2 La table Artistes

On constate que les données sont relativement bien formatées. Cependant, conserver la colonne ArtistBio semble peu pertinent, car elle répète des informations déjà présentes dans les colonnes Nationality, BeginDate et EndDate. Un nettoyage de la colonne EndDate s'avère également nécessaire. En effet, pour les artistes encore en vie, la valeur 0 est actuellement renseignée. Il serait plus pertinent de laisser cette cellule vide dans ce cas précis. Deux identifiants sont à notre disposition :

- Le Wiki QID, qui correspond à un identifiant Wikidata.
- L'ULAN (Union List of Artist Names) de Getty

Nous avons décidé de privilégier Wikidata par rapport à l'ULAN, car les données de Wikidata nous semblent plus complètes. De plus, Wikidata offre 2 % d'identifiants supplémentaires par rapport à l'ULAN.

2.1.3 La table Artworks

À l’approche de la date de rendu du cahier des charges de l’application Flask le 22 janvier, nous organisons une nouvelle réunion.

- Lundi 13 janvier, point d’équipe :
 - Nous discutons des fonctionnalités requises (produit minimum viable, MVP) et des fonctionnalités souhaitées.
 - Nous parlons également de l’aspect graphique du site, du nombre de pages et de leur contenu.

2.1.4 Recherche

Le Modèle Conceptuel des Données est élaboré à cette occasion. Il sera plus tard modifié afin d’incorporer Wikidata pour l’enrichissement des données.

2.2 Semaine 20 - 26 janvier 2025

2.2.1 Requêtage

Pour enrichir la table Artists avec Wikidata, il y avait deux possibilités :

1. La table contient un attribut “WikiID” qui correspond à l’identifiant Wikidata de l’artiste, et cela pour 3247 enregistrements. Il est donc possible de faire un script Python qui itère sur ces 3247 valeurs et récupérer les entités ayant pour propriété Movement ou Genre avec l’entité Personne requêtée.
2. Il existe pour les artistes la propriété “*has works in the collection*” (P6379) qui peut être utilisée pour récupérer les mouvements et genres associés aux artistes du MoMA grâce à des requêtes SPARQL sur le SPARQL Endpoint de Wikidata. Cette méthode permet d’en identifier 9209. Le fait qu’il y en 3 fois plus peut peut-être s’expliquer par le fait que le jeux de données du MoMA est toujours un travail en cours et que les données sont fournies “telles quelles”, comme le précise le README du GitHub du MoMA, donc avec des failles éventuelles. Nous choisissons cette méthode pour récupérer les données Wikidata.

Les requêtes sont les suivantes :

1. Mouvement

```
1 SELECT ?artiste ?artisteLabel ?mouvement ?mouvementLabel
2 WHERE {
3     ?artiste wdt:P6379 wd:Q188740;
4     wdt:P135 ?mouvement.
```



```

5 SERVICE wikibase:label {
6     bd:serviceParam wikibase:language "en".
7 }
8 }

```

Nombre de résultats 1926

2. Genre

```

1 SELECT ?artiste ?artisteLabel ?genreLabel
2 WHERE {
3     ?artiste wdt:P6379 wd:Q188740;
4     wdt:P136 ?genre.
5     SERVICE wikibase:label {
6         bd:serviceParam wikibase:language "en".
7     }
8 }

```

Nombre de résultats 2060

Pour collecter les genres et mouvements artistiques associés aux œuvres d'arts du MoMA sur Wikidata, nous considérons que les œuvres du MoMA sont présentes sur Wikidata :

- Soit via la relation **"owned by"** (P127) avec le MoMA (Q188740) s'il n'y a pas *Point in time* **"end time"** (P582). Cela nous permet de récupérer les œuvres d'arts qui sont actuellement propriété du MoMA.
- oit via la relation **"collection"** (P195) avec comme objet le MoMA pour trouver les œuvres référencées comme dans ses collections.
- Soit via la relation **"location"** (P276) avec comme objet le MoMA pour trouver les œuvres situées au MoMA.

Nous ajoutons ensuite :

1. La relation **"movement"** (P135) aux entités trouvées pour récupérer les mouvements artistiques des entités du MoMA, ce qui nous donne une première requête :

```

1
2 SELECT ?oeuvre ?oeuvreLabel ?mouvement ?mouvementLabel
3 WHERE {
4     { ?oeuvre wdt:P127 wd:Q188740. } # Propriété "owned by"
5     UNION
6     { ?oeuvre wdt:P195 wd:Q188740. } # Propriété "collection"
7     UNION
8     { ?oeuvre wdt:P276 wd:Q188740. } # Propriété "location"

```

```

9
10 ?oeuvre wdt:P135 ?mouvement. # Mouvements artistiques
11
12 SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
13 }

```

On obtient ici 399 résultats.

2. La relation “genre” (P136) qui nous donne les genres artistiques des œuvres.

```

1
2 SELECT ?oeuvre ?oeuvreLabel ?genre ?genreLabel
3 WHERE {
4   { ?oeuvre wdt:P127 wd:Q188740. } # Propriété "owned by"
5   UNION
6   { ?oeuvre wdt:P195 wd:Q188740. } # Propriété "collection"
7   UNION
8   { ?oeuvre wdt:P276 wd:Q188740. } # Propriété "location"
9
10  ?oeuvre wdt:P136 ?genre. # Genres artistiques
11
12  SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
13 }

```

On obtient alors 3104 résultats.

2.2.2 Semaine 27 janvier - 2 février 2025

RAS

3 Février

3.1 Semaine 3 - 9 février 2025

3.1.1 Compte rendues de la réunion interne du 8 février 2025

- **Point d’avancement du projet** : État des lieux avec l’équipe afin de vérifier l’avancement des travaux et identifier d’éventuels points bloquants.
- **Nettoyage des données** : Réflexions aux méthodes de nettoyage des données afin d’améliorer et d’optimiser leur qualité et leur exploitation.

3.1.2 Préparation

Préparation des données :

À l'issue de la réunion, il a été décidé de supprimer **17 colonnes** du jeu de données Artworks. Ces colonnes ont été identifiées comme non pertinentes ou redondantes, et leur suppression permettra d'optimiser la qualité et la gestion des données. Les colonnes supprimées sont classées selon les critères suivants :

Répétition avec le CSV Artist	Duplication au sein du même CSV	Colonne vide	Peu pertinente
ArtistBio	Circumference	OnView	Cataloged
Nationality	Height		ObjectID
Gender	Weight		
BeginDate	Width		
EndDate	Seat Height		
	Duration		

Un travail de nettoyage et de formatage a été réalisé pour les colonnes relatives aux dates de création des œuvres. Deux nouveaux attributs ont été créés :

- **BeginningDate** : Date de début de création de l'œuvre.
- **EndDate** : Date de fin de création de l'œuvre.

Cette préparation a été effectuée à l'aide du logiciel Dataiku et de l'outil Find and Replace.

Les expressions régulières suivantes ont été utilisées pour standardiser les formats de date :

- Transformation de "1976-77" vers "1977" :

```
(\d{2})-
```

- Pour passer de "1985-1988" à "1988" :

```
(\d{4})-
```

- Pour enlever "c." au début de "c.1991" :

```
c\.\s*
```

- Pour enlever le "." à la fin de "1998." :

```
\.$
```

- Pour enlever "published" :

```
\bpublished\b
```

- Pour enlever les deux chiffres suivis d'une virgule :

```
\d{2},
```

- Pour enlever le "s" à la fin d'une date :

```
(?<=\d{4})s
```

- Pour enlever les termes "before" et "after" :

```
[A-Za-z]
```

- Pour enlever les parenthèses :

```
\([^()]*\)
```

3.1.3 Création des tables Genres et Artists_Genres

Afin d'organiser les données de manière optimale, nous avons créé deux nouvelles tables :

- **Genres** : Une table unique listant tous les genres associés aux artistes et aux œuvres.
- **ArtistsGenres** : Une table relationnelle permettant de lier les artistes à leurs genres respectifs.

Source des données :

Nous sommes partis des tables `artists_genre` et `artworks_genre` issues de Wikidata. Ces deux tables ont été fusionnées (stackées) pour constituer la base de notre travail.

Préparation de la table ArtistsGenres

1. Suppression des colonnes inutiles :

- *genderLabel*
- *oeuvrelabel*

2. Nettoyage des identifiants Wikidata :

- Les URL Wikidata ont été remplacées par des valeurs vides afin de ne conserver que les identifiants Wikidata.

3. Renommage des colonnes :

- La colonne **Oeuvre** a été renommée en **ArtistID**.
- La colonne **Genre** a été renommée en **GenreID**.

Préparation de la table Genres :

1. Suppression des colonnes inutiles :

- *oeuvreLabel*
- *oeuvre*

2. Renommage des colonnes :

- La colonne **Genre** a été renommée en **WikiID**.
- La colonne **GenreLabel** a été renommée en **Label**.

3. Suppression des doublons :

- La formule **DISTINCT** a été utilisée pour éliminer les doublons et garantir l'unicité des genres.

3.2 Semaine 10 - 16 février 2025

3.2.1 Jalon

12 février midi , point bilan avec Mme Bermès, Mme Lemosquet, M. Roche-Dioré et le soir, réunion de groupe

— Point Bilan :

- Il a été porté à notre attention que nos données n'étaient, en l'état, pas enrichies, ou du moins pas assez. Nous avons alors évoqué l'idée d'enrichir les données grâce à Wikidata et OpenLibrary.
- Dataviz

— Réunion de groupe :

- Nous avons discuté plus en détail de l'enrichissement de données grâce à Wikidata et OpenLibrary et de la façon dont nous pouvions l'implémenter (voir ci-dessous),
- Nous avons également parlé de la structure Flask de l'application et décidé de la répartition du travail pour les templates HTML.

3.2.2 Requêtage

Wikidata :

Le site Wikidata propose une propriété intitulée *described by source*, qui référence les sources bibliographiques associées à chaque artiste. Afin de récupérer ces informations, une requête SPARQL a été écrite et exécutée.

```

1 SELECT ?artwork ?artist ?artistLabel ?bibliography ?bibliographyTitle ?
   bibliographyAuthor ?bibliographyAuthorLabel ?bibliographyDate ?
   authorSex WHERE {
2   ?artwork wdt:P31/wdt:P279* wd:Q3305213 .
3
4   #?artwork est une instance de ou une sous-classe de "oeuvre d'art"
5
6   ?artwork wdt:P170 ?artist .
7   # ?artwork a pour créateur ?artist

```

```

8
9 ?artwork wdt:P195/wdt:P361* wd:Q188740 .
10 #?artwork est conservé dans ou fait partie de "Museum of Modern Art"
11 # Récupérer les références bibliographiques pour l'artiste
12
13 OPTIONAL {
14     ?artist wdt:P1343 ?bibliography .
15     #?bibliography est une source qui décrit l'artiste
16     OPTIONAL {
17         ?bibliography wdt:P1476 ?bibliographyTitle .
18         #?bibliographyTitle est le titre de la bibliographie
19     }
20     OPTIONAL {
21         ?bibliography wdt:P50 ?bibliographyAuthor .
22         #?bibliographyAuthor est l'auteur de la bibliographie
23         OPTIONAL {
24             ?bibliographyAuthor wdt:P569 ?bibliographyDate .
25             #?bibliographyDate est la date de naissance de l'auteur de la
26                 bibliographie
27         }
28         OPTIONAL {
29             ?bibliographyAuthor wdt:P21 ?authorSex .
30             #?authorGender est le genre de l'auteur de la bibliographie
31         }
32     }
33 }
34 SERVICE wikibase:label { bd:serviceParam wikibase:language "[
35     AUTO_LANGUAGE],en". }

```

Nous obtenons 4639 résultats.

OpenLibrary

: Le site OpenLibrary propose une [documentation](#) permettant de récupérer des JSON. Nous utilisons un [script python](#) pour interroger l'API, sur le thème art. Nous obtenons un JSON qui contient 101 545 livres.

3.3 Semaine 17 - 23 février 2025

3.3.1 Requêtage

Requêtage de liens vers des images des artistes du MoMA Les Artworks ont déjà un lien vers une image, mais nous pouvons requêter en SPARQL, de la même manière que pour les Mouvements et Genres, les url des images d’artistes ayant exposé au MoMA. Cela sera surtout utile pour l’application Flask donc nous n’entrons pas dans les détails.

```
1 SELECT ?artiste ?artisteLabel ?image ?imageLabel
2 WHERE
3 {
4     ?artiste wdt:P6379 wd:Q188740;
5     wdt:P18 ?image.
6     SERVICE wikibase:label { bd:serviceParam wikibase:language "en". }
7 }
```

Nous obtenons 3376 résultats.

Croisement OpenLibrary / Wikidata :

Nous avons ensuite enrichi le dataset d’OpenLibrary en y ajoutant des identifiants Wikidata pour les mouvements artistiques et les artistes correspondants. Pour cela, nous avons écrit un [notebook Python](#) pour croiser les données d’OpenLibrary avec deux datasets que nous avons déjà : Movements.csv (contenant les mouvements artistiques et leurs IDs Wikidata) et Artists.csv (avec les noms d’artistes et leurs ID Wikidata). Nous avons d’abord converti ces fichiers en dictionnaires pour faciliter la recherche des correspondances, puis parcouru le JSON d’OpenLibrary et cherché les correspondances :

- Les mouvements artistiques correspondant à des sujets d’OpenLibrary ont été remplacés par leurs WikiId : permet de dire avec cette certitude, que si ce mouvement est présent dans la liste des sujets des références d’OpenLibrary, alors cette référence peut être considérée comme liée au mouvement. Remplacer l’intitulé par le WikiID permet alors de faire une jointure sur la table Mouvement. Nous n’avons pas fait de même pour les Genres car ceux-ci sont trop génériques (“fiction”, “drama” par exemple) et créent de mauvaises associations et du bruit.
- Les artistes du MoMA mentionnés dans les titres d’OpenLibrary ont été signalés par l’ajout leur Wikidata ID dans le dictionnaire des références d’OpenLibrary, sous condition qu’ils comportent un espace (nom complet, cela évite le bruit pour les artistes qui ont un nom très court, par exemple “Chi”).

Ces données ont ensuite été sauvegardées dans un [csv](#).

Associer les genres et dates de naissance aux auteurs des livres d'OpenLibrary

L'objectif de cette requête était d'enrichir les données des auteurs des livres du dataset d'OpenLibrary, notamment leur genre, pays de citoyenneté, date de naissance et éventuellement date de décès. Wikidata dispose de toutes ces informations, et nous pouvons les récupérer grâce à la relation "Open Library ID" (P648) qui indique l'identifiant OpenLibrary d'un auteur où d'une œuvre. Ici nous n'avons pas trouvé de point de liaison pour faire une requête SPARQL dans Wikidata directement (par exemple, il n'était pas possible de faire une requête du type "toutes les personnes qui ont un Open Library ID et qui ont écrit des livres sur le thème art" car ce lien n'existe pas).

La méthode que nous avons choisi passe par un [script Python](#) qui exécute la requête SPARQL suivante pour tous les identifiants OpenLibrary :

```
1 SELECT ?artist ?artistLabel ?birth ?death ?gender ?genderLabel ?country ?
   countryLabel ?ol_id
2 WHERE {{
3     VALUES ?ol_id {{ { values } }}
4     ?artist wdt:P648 ?ol_id;
5     wdt:P21 ?gender;
6     wdt:P569 ?birth.
7     OPTIONAL {{ ?artist wdt:P570 ?death. }}
8     OPTIONAL {{ ?artist wdt:P27 ?country. }}
9     SERVICE wikibase:label {{ bd:serviceParam wikibase:language "en". }}
10 }}
```

Ici "values" est la variable remplacée par les identifiants OpenLibrary des auteurs.

3.3.2 Préparation :

Nettoyage de BiographieArtists créer à partir de Wikidata :

- Suppression des colonnes suivantes : artwork, artistLabel, BiographyAuthor.
- Remplacement des identifiants Wikidata : Q6581097 par "male"; et Q6581072 par "female".
- Suppression des lignes où les colonnes BiographyID et BiographyTitle.

3.3.3 Création des différentes tables à partir du jeu de données récupéré sur Open-Library :

Création de la table OLArtists :

- Suppression des colonnes suivantes : (WikidataSubjectID, FirstPublishYear, AuthorName, AuthorKey, Title).
- Renommage de la colonne OLKey en OLKeyBook.
- Utilisation de la formule distinct pour éliminer les doublons.

Création de la table relative OLkey_AuthorKey :

- Suppression des colonnes suivantes : (WikidataArtistID, WikidataSubjectID, FirstPublishYear, AuthorName, Title).
- Renommage des colonnes : OLKey en OLKeyBook et AuthorKey en OLKeyAuthor.

Création la table OLBooks :

- Suppression des colonnes (WikidataArtistID, WikidataSubjectID, AuthorName, AuthorKey) .
- Renommage des colonnes : OLKey en OLKeyBook et Title en BookTitle.
- Utilisation de la formule distinct pour éliminer les doublons.

Création de la table OLMouvement :

- Suppression des colonnes suivantes : (WikidataArtistID, FirstPublishYear, AuthorName, AuthorKey, Title)
- Renommage des colonnes : OLKey en OLKeyBook et WikidataSubjectID en WikidataMovementID.

3.3.4 Exportation :

Nous avons exporté les fichiers depuis Dataiku en .csv. Nous avons ensuite créé une base de données SQLITE dans DBBrowser et importé les csv.

3.4 Semaine 24 février - 2 mars 2025

3.4.1 Documentation

Cette semaine constitue une première étape de finalisation pour ce projet. Notre objectif étant de faciliter la compréhension du travail effectué jusqu'à ce jour et de garder une trace précise de l'ensemble des actions menées, nous avons repris le journal de bord et avons structuré,

clarifié et organisé les données collectées tout au long du projet selon une approche chronothématique.

Conclusion :

Pour aller plus loin : des requêtes SQL

Nous proposons ici quelques exemples de requêtes SQL pour montrer des utilisations possibles de notre base de données.

Une requête SQL pour récupérer les artistes et leurs genres associés s'ils en ont :

```
1 SELECT Artworks.Title, Artists.DisplayName, Movements.Label
2 FROM Artworks
3 INNER JOIN Artists ON Artworks.Artist = Artists.DisplayName
4 INNER JOIN ArtistsMovements ON Artists.WikiID = ArtistsMovements.ArtistID
5 INNER JOIN Movements ON ArtistsMovements.MovementID = Movements.WikiID;
```

Une requête SQL pour récupérer les ouvrages concernant les artistes :

```
1 SELECT Artists.DisplayName, BiographiesArtists.biographyTitle,
   BiographiesArtists.BiographyAuthorLabel, OLBooks.BookTitle
2 FROM BiographiesArtists
3 INNER JOIN Artists ON BiographiesArtists.ArtistID = Artists.WikiID
4 INNER JOIN OpenLibraryArtists ON Artists.WikiID = OpenLibraryArtists.
   WikidataArtistID
5 INNER JOIN OLBooks ON OpenLibraryArtists.OLKeyBook = OLBooks.OLKeyBook;
```