

DIALLO Mamadou Saliou
DIALLO Ibrahima
FALADE Irving
LE MEUR Gurban

2020/2021
Université de Nantes
M1 Informatique

RAPPORT PROJET BDE

Lien du git : <https://github.com/saliou673/movie-datawarehouse>

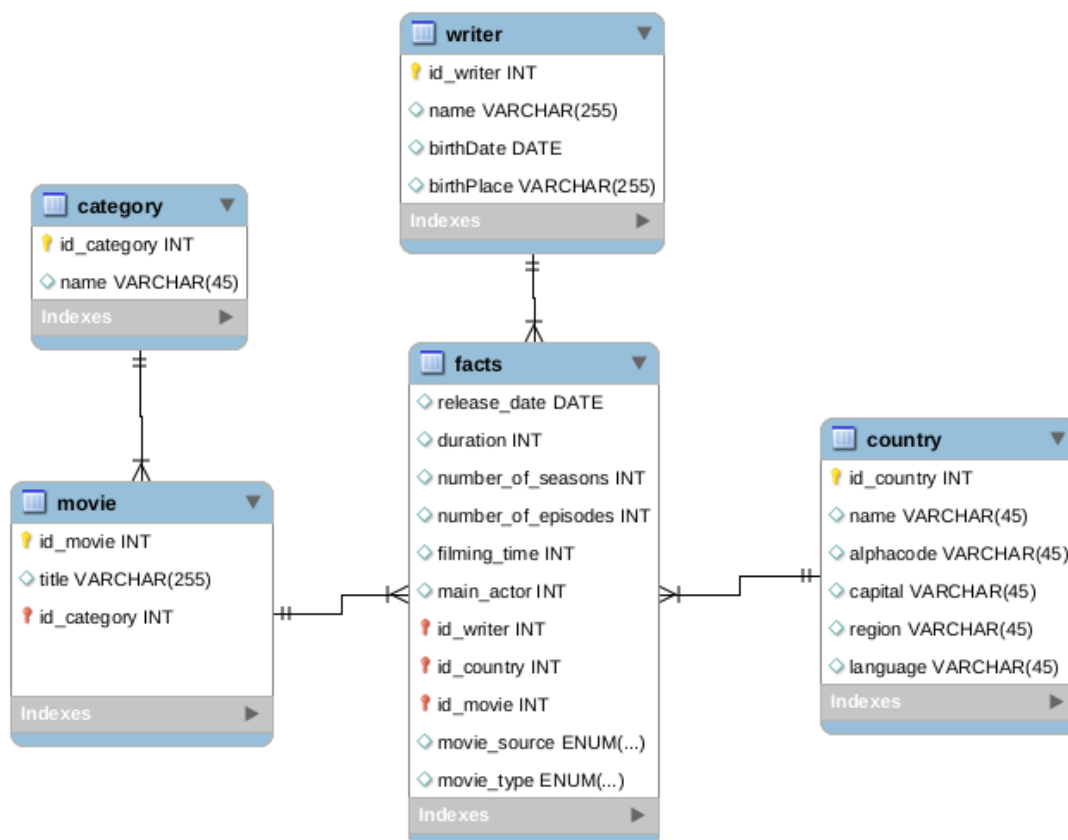
Table des matières :

Table des matières :	1
Introduction :	2
Modèle de l'entrepôt :	2
Requêtes :	3
Requête 1 :	3
Requête 2 :	4
Méthode d'intégration :	4
Nettoyage des données :	5
Difficultés rencontrées :	5
Conclusion :	5

Introduction :

Le but de ce projet était la création d'un entrepôt de données qu'il a fallu collecter à travers des ensembles de données disponibles en open source sur le web et qui se rejoignent vers un thème commun pour l'analyse du processus d'entreprise. Ici, ce dernier se base sur les productions cinématographiques hébergées par différentes plateformes de streaming les plus connues comme Netflix, Disney et Amazon Prime Video. Ainsi, les ensembles de données étaient tous choisis, car il y en avait une pour chacune des plateformes. Avec ces données on peut classer les films ou séries par pays, catégories et réalisateurs, et en tirer des informations intéressantes sur l'exploitation de ces plateformes pour produire des films/séries.

Modèle de l'entrepôt :



Sur le modèle en flocon ci-dessus, on observe nos trois dimensions qui sont le movie, le country et le writer. Et on peut voir que les grains de processus, c'est à dire les aspects associés aux films selon les attributs des tables de dimensions, sont les suivants : le titre, la date de sortie, la durée, le nombre de saisons, le nombre d'épisodes, le temps de tournage, le nombre d'acteurs principaux, le type (film ou série) et la plateforme qui propose le contenu.

Requêtes :

Nous allons vous présenter 2 requêtes OLAP que nous avons effectuées sur notre base de données.

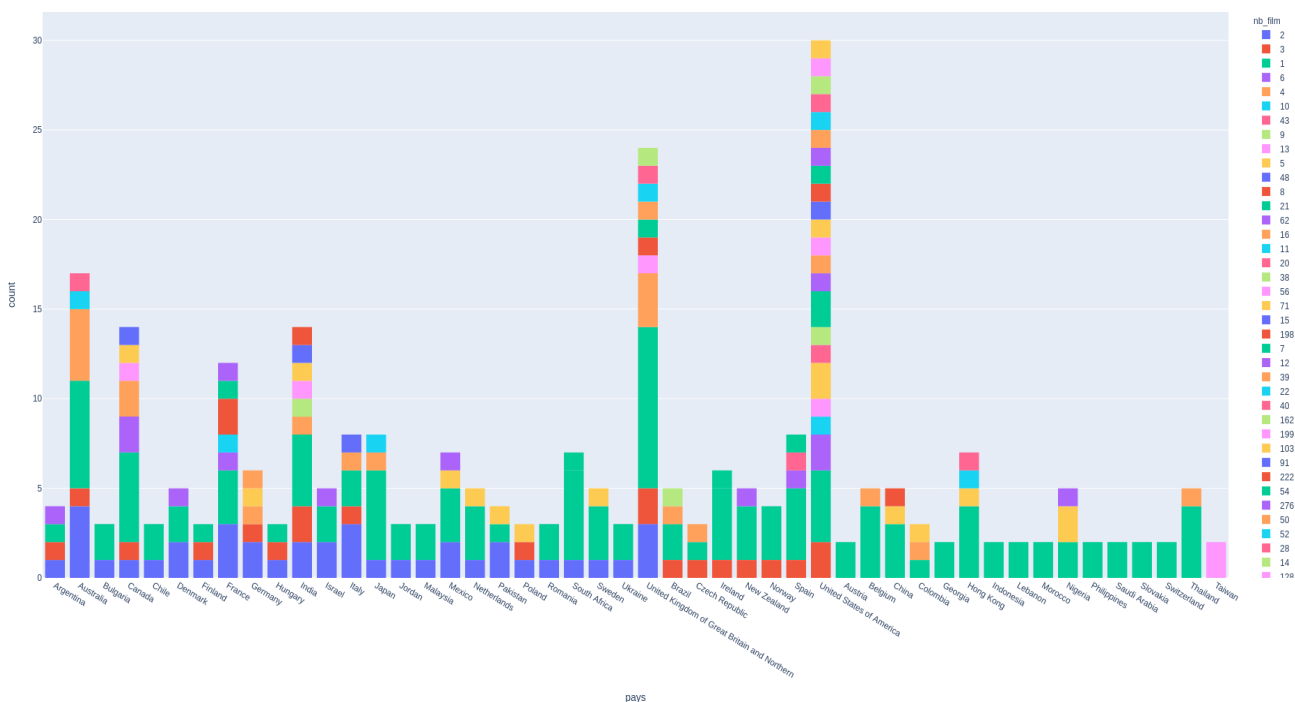
- Requête 1 :

Nombre de film par catégorie et par pays :

```
select co.name as pays, c.name as category, count(m.id_movie) as  
nb_film, GROUPING(co.name,c.name) as groupin
```

```
from facts f, category c, country co, movie m
```

```
where f.id_movie=m.id_movie and f.id_country=co.id_country and  
m.id_category=c.id_category group by co.name, c.name with rollup;
```

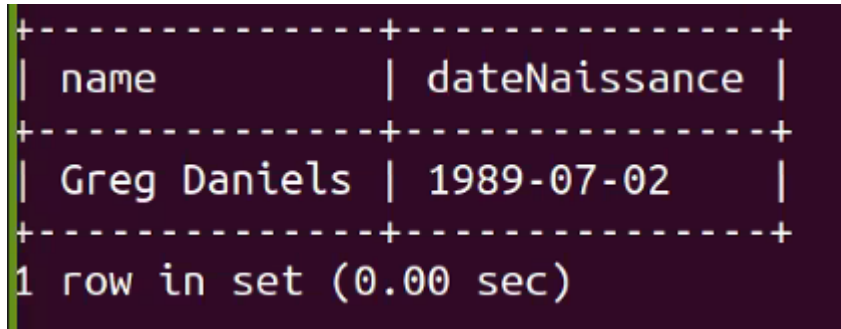


On peut voir, selon le résultat de la requête, que les États-Unis sont sans grande surprise, le pays qui produit le plus de contenu avec la plus grande variété de catégories sur les plateformes de streaming. Les différentes couleurs sur une barre représentent une catégorie.

- Requête 2 :

Le nom et la date de naissance du réalisateur le plus jeune :

```
select w.name, MAX(w.birthdate) as dateNaissance
from facts f, movie m, writer w
where f.id_writer=w.id_writer and f.id_movie=m.id_movie;
```



```
+-----+-----+
| name          | dateNaissance |
+-----+-----+
| Greg Daniels  | 1989-07-02    |
+-----+-----+
1 row in set (0.00 sec)
```

On peut voir que le réalisateur le plus jeune s'appelle Greg Daniels et est né en 1989. On utilise la fonction MAX sur l'attribut birthdate car plus la date est récente, et donc plus le réalisateur est jeune, plus l'année sera grande.

Le reste des requêtes se trouve directement sur ce lien :

<https://github.com/saliou673/movie-datawarehouse/blob/master/src/migrations/query/requete.sql>

Méthode d'intégration :

Pour l'intégration des données des différents datasets, nous avons modélisé la base de données en écrivant des scripts MySql. Nous avons ensuite procédé au mapping entre les champs des datasets et ceux de la base de données. Puis nous avons intégré automatiquement les données via un script python.

Nous avons aussi utilisé des requêtes SPARQL pour compléter les informations incomplètes dans les datasets car un champ existant dans un dataset pourrait ne pas exister dans un autre.

Nettoyage des données :

Pour le nettoyage des données, nous avons d'abord procédé par la suppression des tuples ne possédant pas de catégories. Ensuite nous avons pris en compte que la première catégorie dans le cadre d'un film qui avait plusieurs catégories. Et enfin nous avons supprimé les films dupliqués (les films se trouvant à la fois dans 2 ou nos 3 datasets).

Difficultés rencontrées :

La première difficulté a été d'intégrer les données qui était parfois trop hétérogène entre les datasets. Nous avons donc dû les compléter à l'aide de requêtes sparql prélevant les sources directement depuis dbpedia.

La deuxième était de ne pas pouvoir utiliser les commandes CUBE et GROUPING SET avec mySql, il a fallu écrire des requêtes équivalentes avec ROLLUP.

Conclusion :

Le projet était intéressant et demandait de savoir se débrouiller selon les situations étant donné que nous étions libres de choisir nos ensemble de données et vers quoi notre analyse allait se porter. L'entrepôt peut s'étendre en ajoutant plus de films ou séries ou même en ajoutant des dimensions, voir même au delà des plateformes de streaming ajouter des contenus de plateforme hébergeant des vidéos comme Youtube ou des films qui ne sortent qu'en salle de cinéma. Cela se rejoint tous à du contenu audiovisuel.