Université Sorbonne Paris Nord

2022-2023

SAE 3.01 – Base de données de films

Analyse de la BDD

Team DraCorporation

Analyse de la base de données d'IMDB

I Présentation de la Base de données

Nous devons utiliser une base de données à l'issue de notre SAE rédigé par M. JCD de l'IUT Villetaneuse

La base utilisée est la base de données d'IMDB disponible sur le lien suivant : https://www.imdb.com/interfaces/

Suite à l'analyse du cadre juridique, nous nous apercevons que nous ne pouvons pas modifier la base de données mais nous devons utiliser directement la structure de données fournie par IMDB.

Nous devons créer notre propre base de données en récupérant les données d'IMDB et les afficher via PHP sur un site internet que nous allons créer.

II Structure de la base de données

Nous avons pris les données sur le lien suivant : https://datasets.imdbws.com/

La structure de données est un fichier = une table

Table TitleBasics :

Nom de la colonne	Type	Fonction	Null accepté ?	Commentaire
Tconst	text	Clef primaire	Non	
titleType	text	Type ou Format du titre	Oui	
primaryTitle	text		Oui	
originalTitle	boolean	Indique si c'est 18+	Non	0: non-adult 1: adult
startYear	smallint	La date de réalisation	Oui	
endYear	smallint	La date de fin de diffusion	Oui	Null sauf pour série TV
runtimeMinutes	integer	La durée en min	Oui	
genres	text[]	3 genres associés	Oui	

Table TitleAkas:

Nom de la colonne	Type	Fonction	Null accepté ?	Commentaire
titleId	text	Clef primaire composée	Non	Référence au tconst de la table TitleBasics
ordering	integer	Clef primaire composée	Non	
title	text	Le titre selon la région	Non	
region	text	la région du titre	Oui	
language	text	la langue	Oui	
types	text[]	attribut classique associé	Oui	
attributes	text[]	attribut additionnel	Oui	
isOriginalTitle	boolean	Indique si le titre est dans la langue originale	Oui	

Table NameBasics:

Nom de la colonne	Type	Fonction	Null accepté ?	Commentaire
nconst	text	clé primaire	non	
primaryName	text	nom de la personne	non	
birthYear	smallint	année de naissance	Oui	
deathYear	smallint	année de décès	Oui	Null si il n’est pas mort
primaryProfession	text[]	Profession	Oui	
knownForTitles	text[]	Titre dont il est connu pour	Oui	

Table TitlePrincipals:

Nom de la colonne	Type	Fonction	Null accepté ?	Commentaire
tconst	text	Clef primaire composée	Non	Référence TitleBasics
ordering	integer	Clef primaire composée	Non	
nconst	text	Clef primaire composée	Non	Référence NameBasics
category	text	La catégorie du métier de la personne	Oui	
job	text	le métier de la personne	Oui	null si non applicable
characters	text	le rôle joué	Oui	null si il ne joue pas

Table TitleEpisode:

Nom de la colonne	Type	Fonction	Null accepté ?	Commentaire
tconst	text	Clef primaire composée	Non	Référence TitleBasics
parentTconst	text	id du parent tconst	Non	Référence TitleBasics
seasonNumber	smallint	Numéro de la saison	Oui	
episodeNumber	integer	Numéro de l'épisode	Oui	

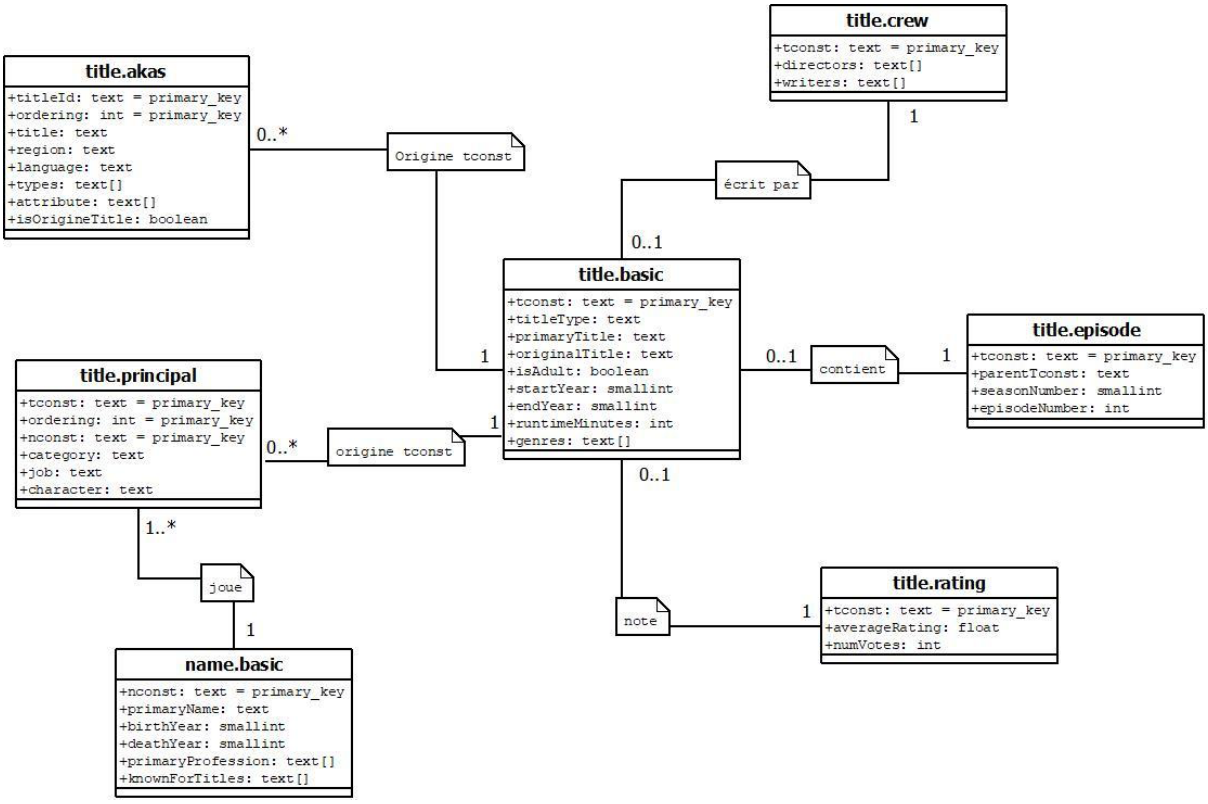
Table TitleCrew:

Nom de la colonne	Type	Fonction	Null accepté ?	Commentaire
tconst	text	Clef primaire composée	Non	Référence TitleBasics
directors	text[]	directeurs	Oui	
writers	text[]	écrivains	Oui	

Table TitleRatings:

Nom de la colonne	Type	Fonction	Null accepté ?	Commentaire
tconst	text	Clef primaire composée	Non	Référence TitleBasics
averageRating	float	Une note sur 10	Oui	
NumVotes	integer	Nombre de vote	Oui	

Nous avons établit le diagramme UML suivant:



D’après les dépendances des tables, nous devons d’abord créer la table TitleBasic et NameBasics puis nous pouvons ensuite créer les autres tables.

III Soucis de la base de données fournie

Lors de l'importation, nous avons remarqué quelques problèmes dans le contenu de la base de données fournie par IMDB:

De nombreuses incohérences ont été remarquées dans la base de données fournie par IMDb. Certaines contraintes d'intégrité que nous avions appliqué aux tables n'étaient pas respectées, par exemple la date de mort d'un acteur devait être supérieure ou égale à sa date de naissance mais nous avons trouvé des acteurs qui sont morts avant leur naissance.

Exemple :

```
psycopg2.errors.CheckViolation: ERREUR:  la nouvelle ligne de la relation « namebasics » viole la contrainte de vérification « namebasics_check »
DETAIL:  La ligne en échec contient (nm0202552, C. Daveillans, 1962, 1936, writer, tt14793342,tt0183025).
CONTEXT:  COPY namebasics, ligne 192279 : « nm0202552   C. Daveillans   1962    1936    writer  tt14793342,tt0183025 »
```

Ici on peut voir un acteur qui est né en 1962 et est mort en 1936.

D'autres problèmes ont également été relevés lors de l'importation de toutes les données. Beaucoup de "tconst" étaient manquants.

Exemple :

```
Connection Opened!
Starting data injection ...
Started title.principals
Finished title.principals in 13756.628390073776 seconds with a total of 13655 errors.
```

Pendant l'insertion de "titleprincipals", nous nous sommes rendu compte qu'il y avait 13 655 lignes en erreur sur la cinquantaine de millions.

Nous avons également eu des problèmes de conception liés au noms de champs données par IMDb, notamment l'identifiant unique pour un film qui est appelé "tconst" dans toutes les tables sauf "titleakas". La clé ordering qui existe dans "titleakas" et "titleprincipals" sous le même nom bien qu'il n'y ait aucun lien.

Le nombre de lignes des tables nous à également posé des problèmes en termes de temps. L'insertion était longue et nous étions donc obligés d'attendre pour voir nos erreurs et réessayez d'insérer les données.

Le type des variables nous a également posé un petit problème comme avec le nombre d'épisodes d'une série, nous avions utilisé des smallint en pensant que celui-ci ne dépasserait pas 32 767, or certaines séries dépassent ce nombre et nous avons donc dû changer le code dé création de table.

Cependant, un des problèmes majeurs est que les arrays PostgreSQL commencent et finissent par des accolades qui n'étaient pas présentes dans les fichiers '.tsv" de IMDb. Ce qui nous a obligé à réaliser des millions d'insertion sur les tables.