TP final : Création d'application pour le Cloud

Yasmeen TRifiss - KilLian VENEZIA - Tara Zhong

Développement d'application pour le Cloud | 29-12-2017

1.1 Jeux de données

Pour réaliser cette application, nous utiliserons la base de données Sakila : <https://relational.fit.cvut.cz/dataset/Sakila>. Cette base de données est une base de données fictive qui simule la gestion d’une cinémathèque (lieu où l’on peut louer des DVDs)

# 1.2 Spécification des besoins

Pour pouvoir orienter les choix des étapes suivantes, il va falloir étudier les besoins de l’application :

— Définir, en langage courant, 4 types d’interrogations sur votre jeu de données. On estimera que celles-ci sont effectuées très fréquemment. Positionnez-vous comme un utilisateur standard de l’application.

* Quels sont les titres de films dont la note est supérieure à « » et qui sont disponibles
* Quels sont les titres de films dont le genre est « » et qui sont disponibles
* Quels sont les titres de films dont l'acteur est « » et qui sont disponibles
* Donner la liste de tous les films disponibles.

*Pour savoir si un DVD est disponible, il faut regarder la date de retour «*return\_date » *dans la table rental et regarder si la date de retour «*rental\_duration » dans la table film *par rapport à la date de l’emprunt «*rental\_date »

— Définir, en langage courant, 2 tymopes d’interrogations lourdes sur votre jeu de données. Ces requêtes seront effectuées moins fréquemment. Positionnez-vous comme un analyste ou un décisionnaire de l’application.

* Trier la liste de films les plus loués (avec le nombre de fois) par ordre décroissant
* Pour chaque genre donner le nombre de notes et la moyenne des notes par ordre décroissant.

— Définir des indicateurs de statistiques pertinents sur votre jeu de données. Positionnez-vous comme administrateur de la base de données.

Les indicateurs sont les catégories, acteurs, clients.

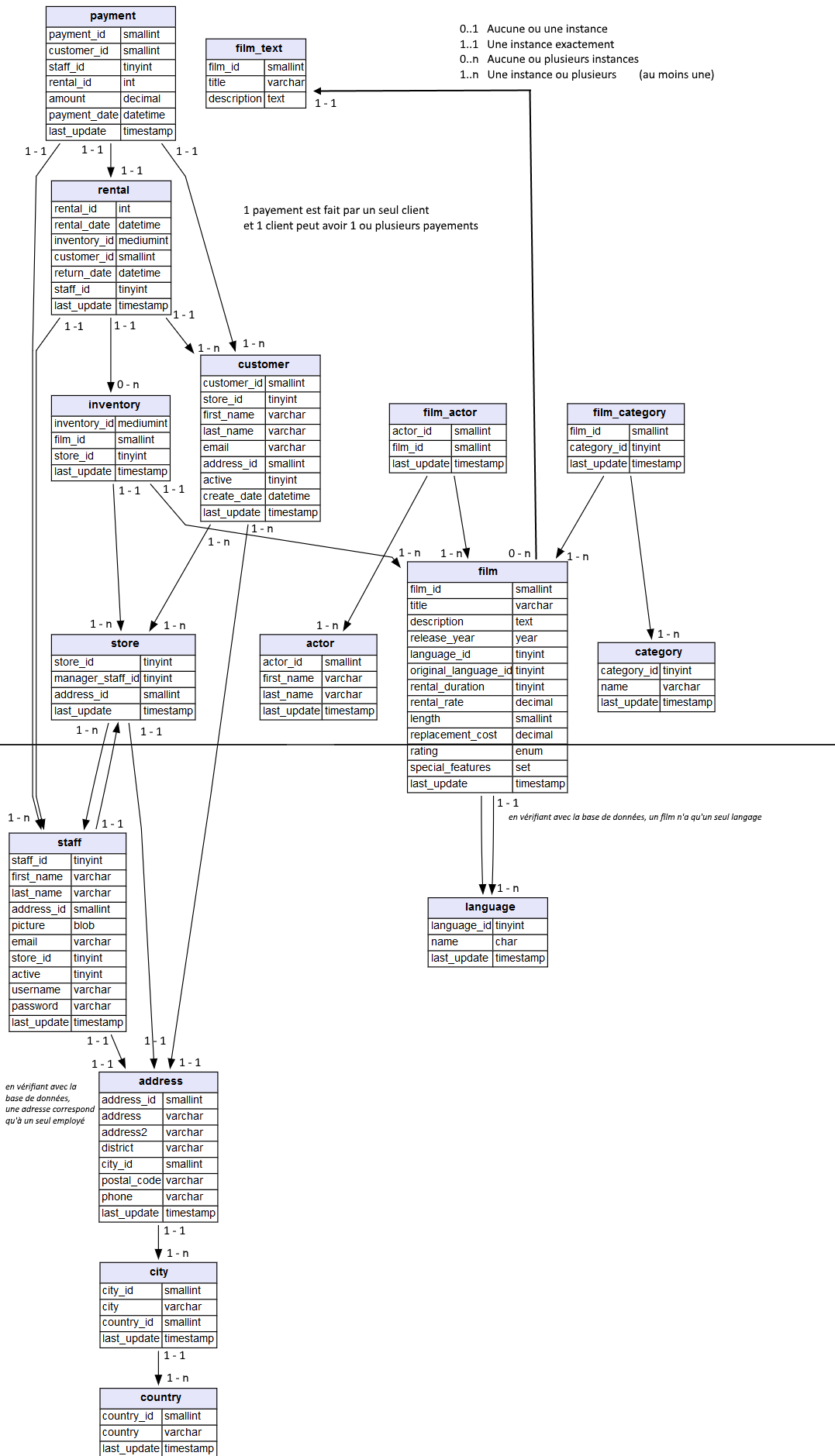
Ex de requêtes : Quels sont les clients qui n’ont retourné leur DVD.

# 1.3 Dénormalisation

## 1.3.1 Schéma

Afin de réaliser la dénormalisation, nous avons définis les cardinalités de chaque table pour regrouper entre elles les plus pertinentes en observant les données dans la base de données

**Schéma relationnel initial Sakila**



Nous avons supprimé les redondances rencontrées (par ex la table film\_text comporte les mêmes informations que dans la table film (title et description)).

Les tables finales sont :

**Film**: qui contient la table language convertie en un seul string (dans la base de données on a observé qu’une seule unique langue), la table actor convertie en liste des acteurs, et la table category convertie en liste des catégories en plus. (cf structure json)

**Customer** : qui contient la table adresse du customer convertie en attributs (address, district,city, postal\_code, country et phone), les attributs « city » de la table city et « country » de la table country ont directement été placé à côté de l’adresse. (cf structure json)

**Rental** : qui contient la table payment en plus convertie en attributs. (cf structure json)

**Store** : qui contient la table adresse du store convertie en attributs (address, district,city, postal\_code, country et phone), les attributs « city » de la table city et « country » de la table country ont directement été placé à côté de l’adresse. (cf structure json)

**Staff** : qui contient la table adresse du staff convertie en attributs (address, district,city, postal\_code, country et phone), les attributs « city » de la table city et « country » de la table country ont directement été placé à côté de l’adresse. (cf structure json)

**Inventory** : aucune modification sur cette table. (cf structure json)

## 1.3.1 Transformation

Pour la transformation nous avons stocké les données dans une base de données relationnelle. Les documents destinations sous format csv ont été produits à l’aide de requêtes SQL.

Puis nous avons importé les collections et converti les fichiers csv en fichiers json sous MongoDB

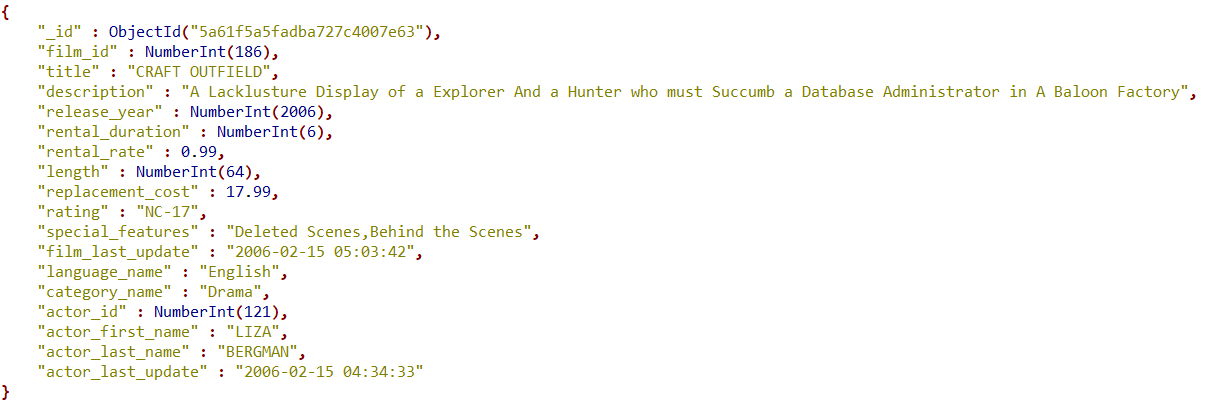
**Après transformation voici nos structures pour chaque table** :

Exemple d’objet de la table **film**

Exemple d’objet de la table **customer**



Exemple d’objet de la table **rental**



Exemple d’objet de la table **store**

Exemple d’objet de la table **staff**

