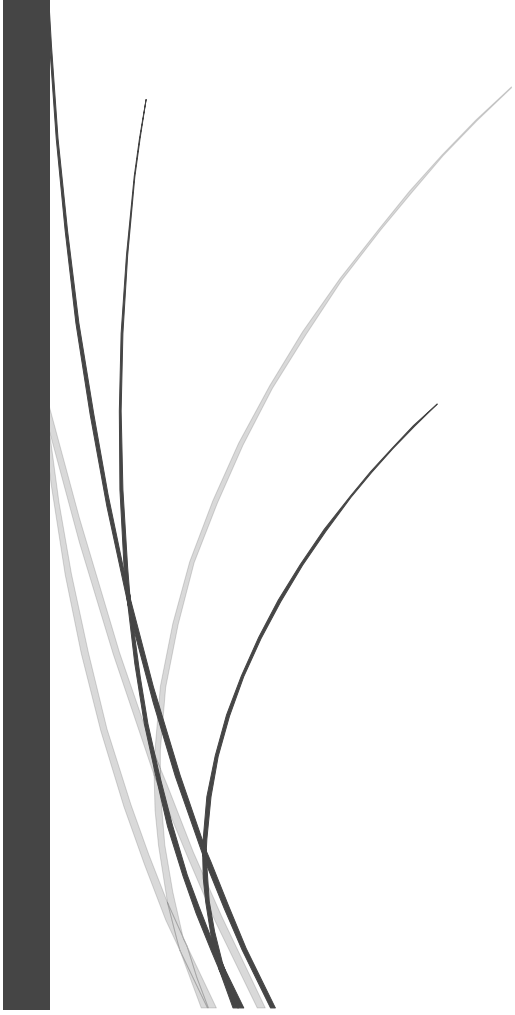
20/01/2018

Rapport de Projet

Développement d'Application Cloud



DURIEZ FABIEN
FENIRI LOUNES
HAIMEUR SHEHRAZADE

Table des matières

1. Jeux de données

Le jeu de données que nous avons sélectionné est Sakila. C'est une base de données de films, qui regroupe des informations sur les chaque film en soi (acteurs, catégories, etc…) mais également des information sur des magasin qui louent ces films (informations sur la localisation, les stocks, les clients ou encore le personnel).

2. Spécification des besoins

a. Interrogation utilisateur standards

- « La liste des films comportant un titre donné et se trouvant dans un magasin donné.»
- « La liste des films appartenant à une catégorie donnée et un magasin donné »
- « La liste des films dans lequel un acteur donné a joué par magasin »
- « Les 10 films les plus loué des 30 derniers jours par magasin »

b. Interrogation lourdes pour analyste

- « Le nombre de location effectué pour un mois donné »
- « Le nombre de film loué par catégorie pour un store donné »

c. Indicateur pour l'administrateur de base de données.

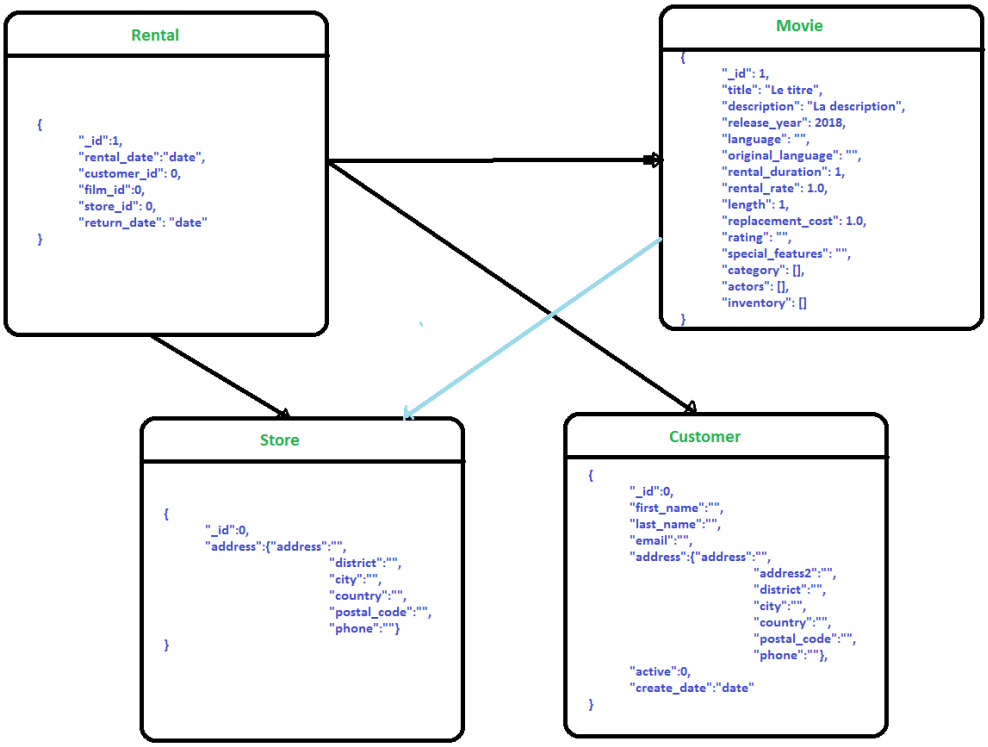
- « Je veux l'ensemble des informations concernant l'état du cluster »

3. Dénormalisation

a. Schéma

Vous trouverez ici le schéma initial.

Afin de répondre aux besoins ci-dessus, nous avons choisis de dénormaliser notre base de la façon suivante :



Dans notre cas d'étude, notre idée principale était de regrouper au maximum les informations car le nombre de table initial est assez élevé.

Nous avons regroupé les tables « Acteurs », « Catégorie » et « Langues » dans un document « Film » car pour nos requêtes ces informations sont complémentaires au film et ne doivent pas nécessiter une jointure trop coûteuse pour leur valeur ajoutée. Nous avons aussi intégré « Inventaire » dans « Film » car pour les requêtes il nous permet de faire le lien entre le document « Film » et « Store » sans faire de jointures.

En ce qui concerne les « Adresses » nous avons regroupé toutes les informations directement dans les «Stores » et « Customer ».

Un document « Store » a été créé afin d'avoir l'adresse des magasins.

Enfin un nous avons mis dans un document « Rental » les informations concernant les clients, les films loués ainsi que le magasin.

Au vus du besoin définit, les tables « Payment » et « Staff » n'ont pas été prises en comptes.

b. Transformation

La méthode que nous avons choisis pour la transformation est la seconde soit :

« Stocker les données dans une base de données relationnelle. Les documents destinations seront produits à l'aide de requêtes SQL (et un formattage) »

Nous avons dans un premier temps intégré les données dans une base MySQL puis grâce à des requêtes SQL nous avons formatté nos données pour produire des documents JSON. Ensuite nous avons créé nos collections dans MongoDB en important ces documents.

4. Vues.

a. Utilisateur Standard

Requête 1	Requête 2	Requête 3	Requête 4
Input user: Store_Id et Titre de film Output request: Les films (Id, Nom, Desc, Langage, Category,...)	Input user: Store_Id et Category_Id Output request: Les films (Id, Nom, Desc, Langage, Category,...)	Input user: Store_Id et un nom d'acteur Output request: Les films (Id, Nom, Desc, Langage, Category,...)	Input user: Aucune entrée Output request: Les films (Id, Nom, Desc, Langage, Category,...)
Implémentation Input_user : Un champ à remplir et une liste déroulante avec les Stores existants. Output : Sous forme de tableau	**Implémentation** Input_user : Deux listes déroulante avec les Stores et Category existants. Output : Sous forme de liste	**Implémentation** Input_user : Un champ à remplir et une liste déroulante avec les Stores existants. Output : Sous forme de liste	**Implémentation** Input_user : Output : Sous forme de liste

b. Analyste/Décisionnaire

Requête 1

Input user:
Mois et Année souhaitée

Output request:
Nombre de location

Requête 2

Input user: Store_Id et Category_Id

Output request:
Category par store

Implémentation

Input_user : Deux une liste déroulante avec les Mois et années.

Output : Texte avec la valeur

Implémentation

Input_user : Une liste déroulante avec les stores.

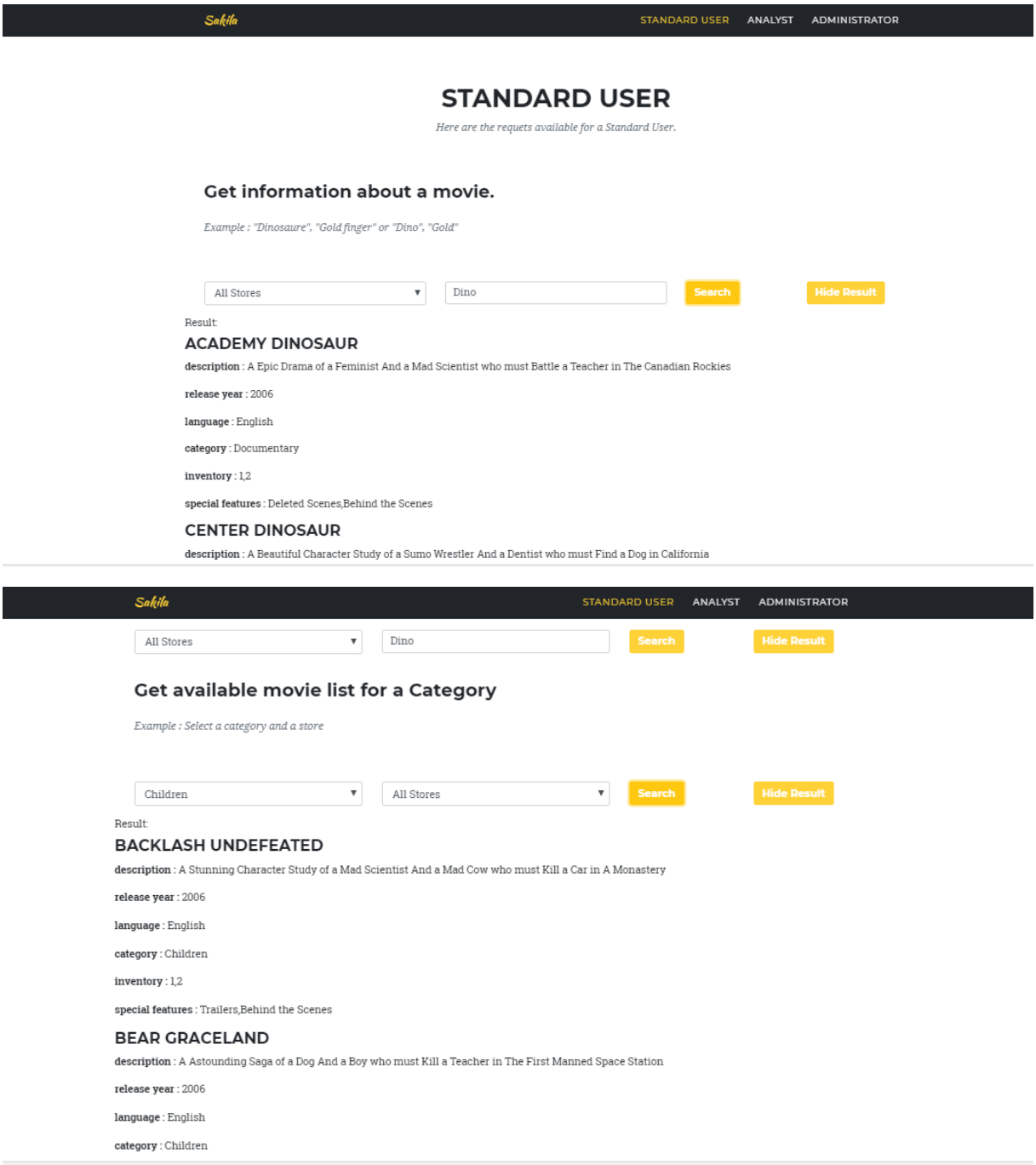
Output : Sous forme de liste

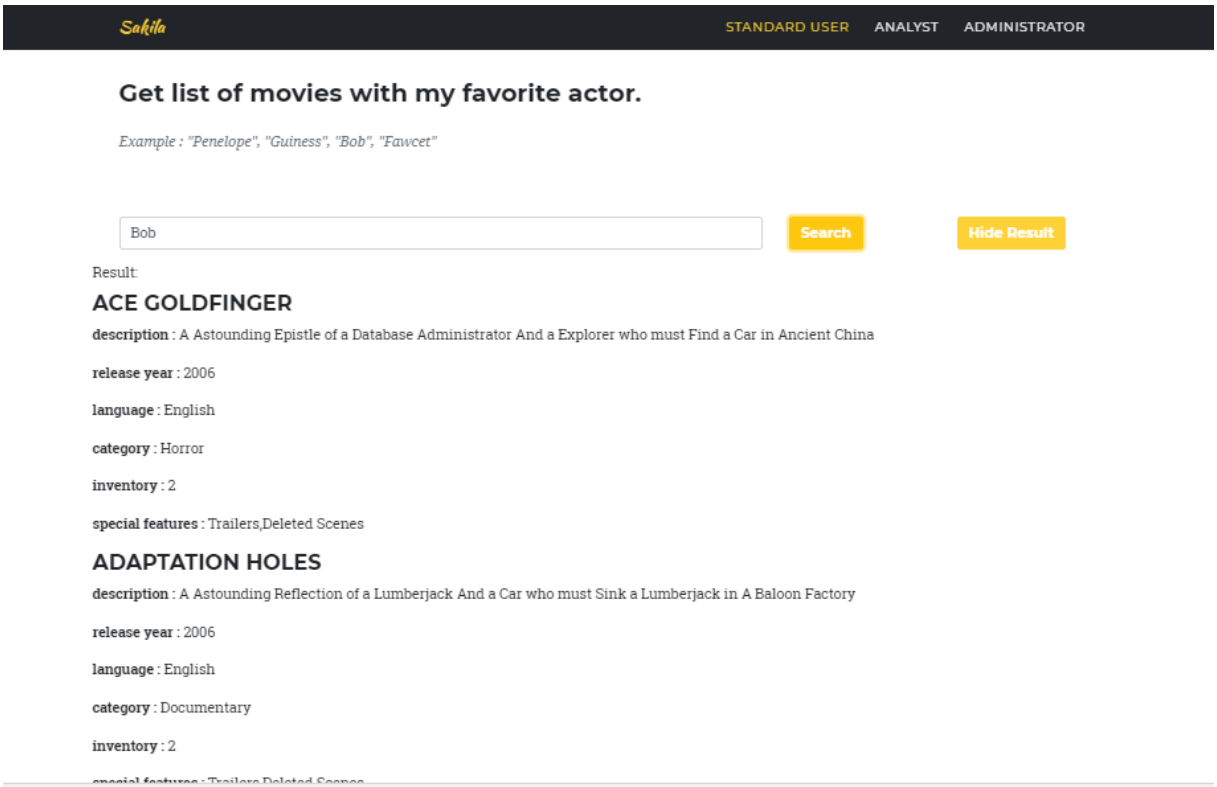
c. Administrateur

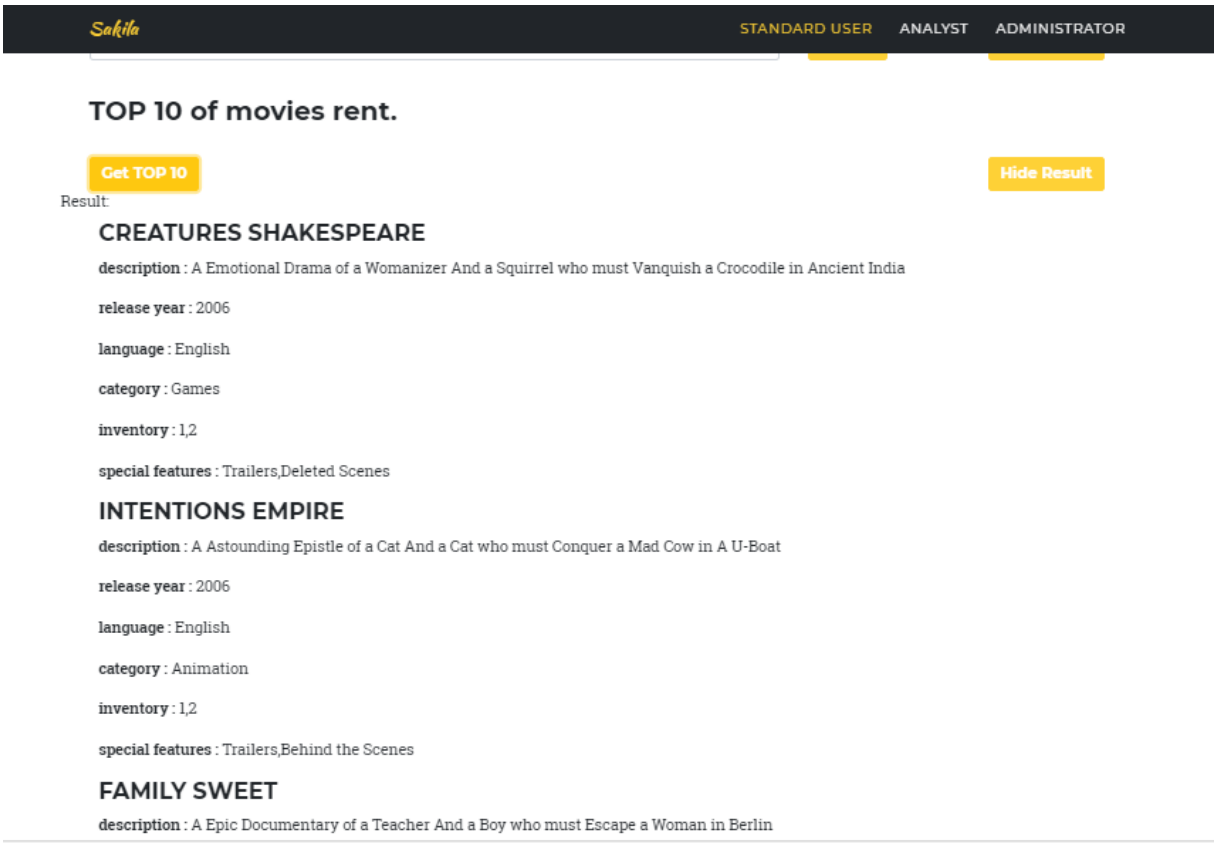
5. Application

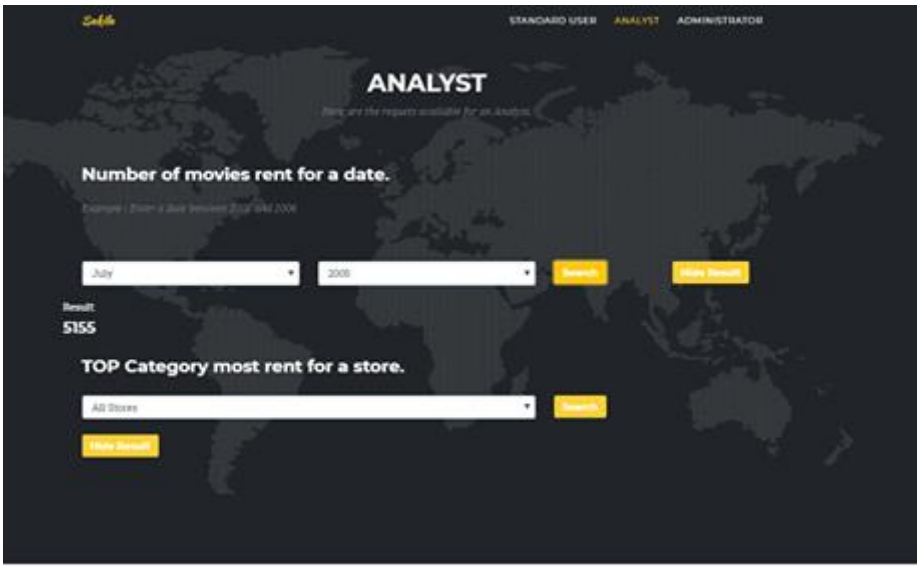
Nous avons choisis de développer cette application grâce à NodeJs et une interface en HTML. Vous trouverez joint à ce dossier un fichier README ainsi que le code de l'application. Vous trouverez en annexe des impressions de l'application web afin de compléter la partie « Vue ».

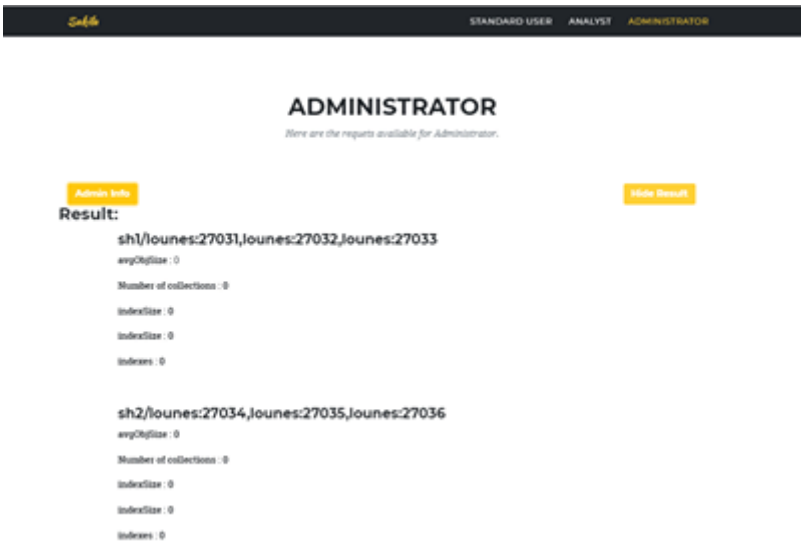
6. Annexe

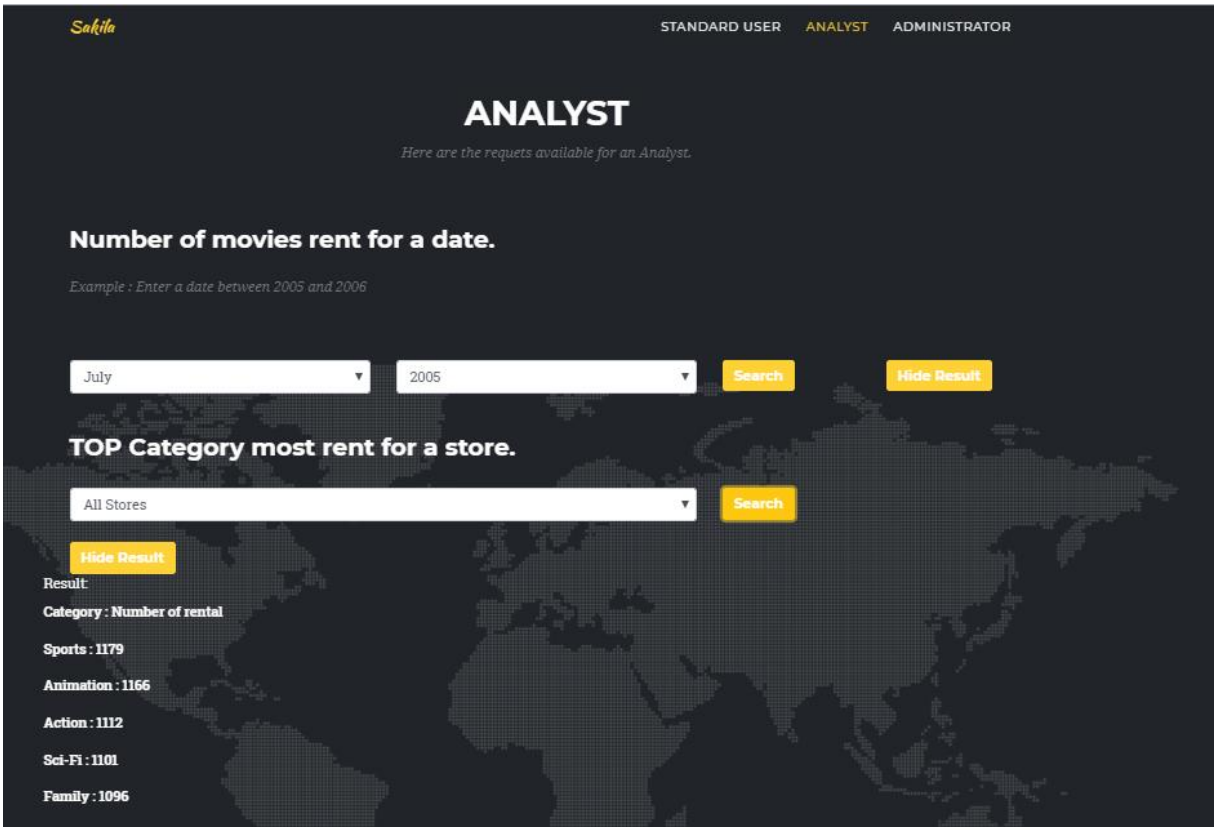

Sakila
STANDARD USER
ANALYST
ADMINISTRATOR
STANDARD USER
Here are the requets available for a Standard User.
Get information about a movie.
Example : "Dinosaure", "Gold finger" or "Dino", "Gold"
All Stores
Dino
Search
Hide Result
Result:
ACADEMY DINOSAUR
description : A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Canadian Rockies
release year : 2006
language : English
category : Documentary
inventory : 1,2
special features : Deleted Scenes,Behind the Scenes
CENTER DINOSAUR
description : A Beautiful Character Study of a Sumo Wrestler And a Dentist who must Find a Dog in California
Sakila
STANDARD USER
ANALYST
ADMINISTRATOR
All Stores
Dino
Search
Hide Result
Get available movie list for a Category
Example : Select a category and a store
Children
All Stores
Search
Hide Result
Result:
BACKLASH UNDEFEATED
description : A Stunning Character Study of a Mad Scientist And a Mad Cow who must Kill a Car in A Monastery
release year : 2006
language : English
category : Children
inventory : 1,2
special features : Trailers,Behind the Scenes
BEAR GRACELAND
description : A Astounding Saga of a Dog And a Boy who must Kill a Teacher in The First Manned Space Station
release year : 2006
language : English
category : Children


Sakila
STANDARD USER
ANALYST
ADMINISTRATOR
Get list of movies with my favorite actor.
Example : "Penelope", "Guiness", "Bob", "Fawcet"
Bob
Search
Hide Result
Result:
ACE GOLDFINGER
description : A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient China
release year : 2006
language : English
category : Horror
inventory : 2
special features : Trailers,Deleted Scenes
ADAPTATION HOLES
description : A Astounding Reflection of a Lumberjack And a Car who must Sink a Lumberjack in A Baloon Factory
release year : 2006
language : English
category : Documentary
inventory : 2


Sakila
STANDARD USER
ANALYST
ADMINISTRATOR
TOP 10 of movies rent.
Get TOP 10
Hide Result
Result:
CREATURES SHAKESPEARE
description : A Emotional Drama of a Womanizer And a Squirrel who must Vanquish a Crocodile in Ancient India
release year : 2006
language : English
category : Games
inventory : 1,2
special features : Trailers,Deleted Scenes
INTENTIONS EMPIRE
description : A Astounding Epistle of a Cat And a Cat who must Conquer a Mad Cow in A U-Boat
release year : 2006
language : English
category : Animation
inventory : 1,2
special features : Trailers,Behind the Scenes
FAMILY SWEET
description : A Epic Documentary of a Teacher And a Boy who must Escape a Woman in Berlin


STANDARD USER
ANALYST
ADMINISTRATOR
ANALYST
Number of movies rent for a date.
July
2005
Search
Hide Result
Result:
5155
TOP Category most rent for a store.
All Stores
Search
Hide Result


Sakila
STANDARD USER
ANALYST
ADMINISTRATOR
ADMINISTRATOR
Here are the requets available for Administrator.
Admin Info
Hide Result
Result:
sh1/lounes:27031,lounes:27032,lounes:27033
avgObjSize : 0
Number of collections : 0
indexSize : 0
indexSize : 0
indexes : 0
sh2/lounes:27034,lounes:27035,lounes:27036
avgObjSize : 0
Number of collections : 0
indexSize : 0
indexSize : 0
indexes : 0


Sakila
STANDARD USER
ANALYST
ADMINISTRATOR
ANALYST
Here are the requets available for an Analyst.
Number of movies rent for a date.
Example : Enter a date between 2005 and 2006
July
2005
Search
Hide Result
TOP Category most rent for a store.
All Stores
Search
Hide Result
Result:
Category : Number of rental
Sports : 1179
Animation : 1166
Action : 1112
Sci-Fi : 1101
Family : 1096