FINAL PROJECT REPPORT – NoSQL

Project – CINE STORE

Professor: Fatah Ioualitene

Submitting by
Yao Junior

Paris, 11 July 2020

Contents

Background

A NoSQL (originally referring to "non-SQL" or "non-relational") database provides a mechanism for storage and retrieval of data that is modelled in means other than the tabular relations used in relational databases. MongoDB Inc. (formerly 10gen) is an American software company that develops and provides commercial support for the open source database MongoDB, a NoSQL database that stores data in JSON-like documents with flexible schemas.

Thanks to EPITA, we have chance to explore the software MongoDB through the subject of NoSQL. Cine Store is the idea to develop a simple application which helps to manage the film information database by using MongoDB as the back end.

The Application

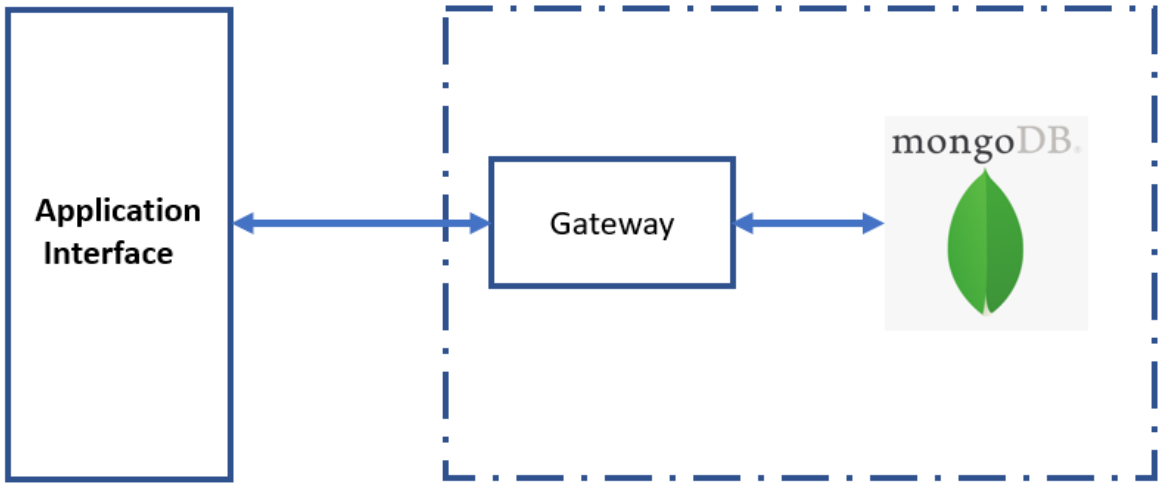
Data Flow

The application allows us to store any movies information into the MongoDB database. The application is structure into different section the add movie section is used to take the movie details and creation a collection into the database, to update any information from the section delete and update acts as interface

The deletion of any information from the database is based on the name of the movie and the language are the passing value to delete a movie information.

Updating or changing any information value uses the name of the movie as a key, and the description and release data of the movie can be change or update

The application can also read data from the mongo DB and display it into the list all section.



Creating Information

Insertion of new Movies information

Insertion SCHEMA

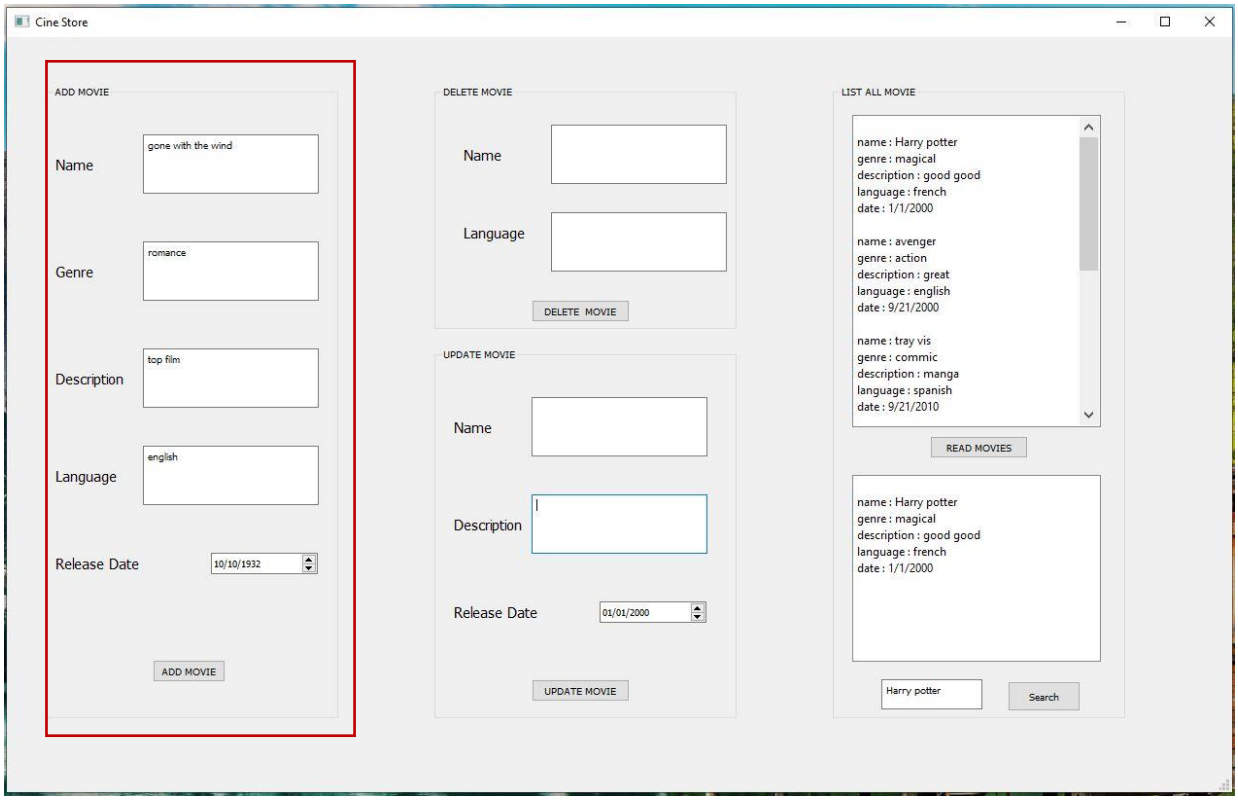
NAME: “take the name of the movie as input”

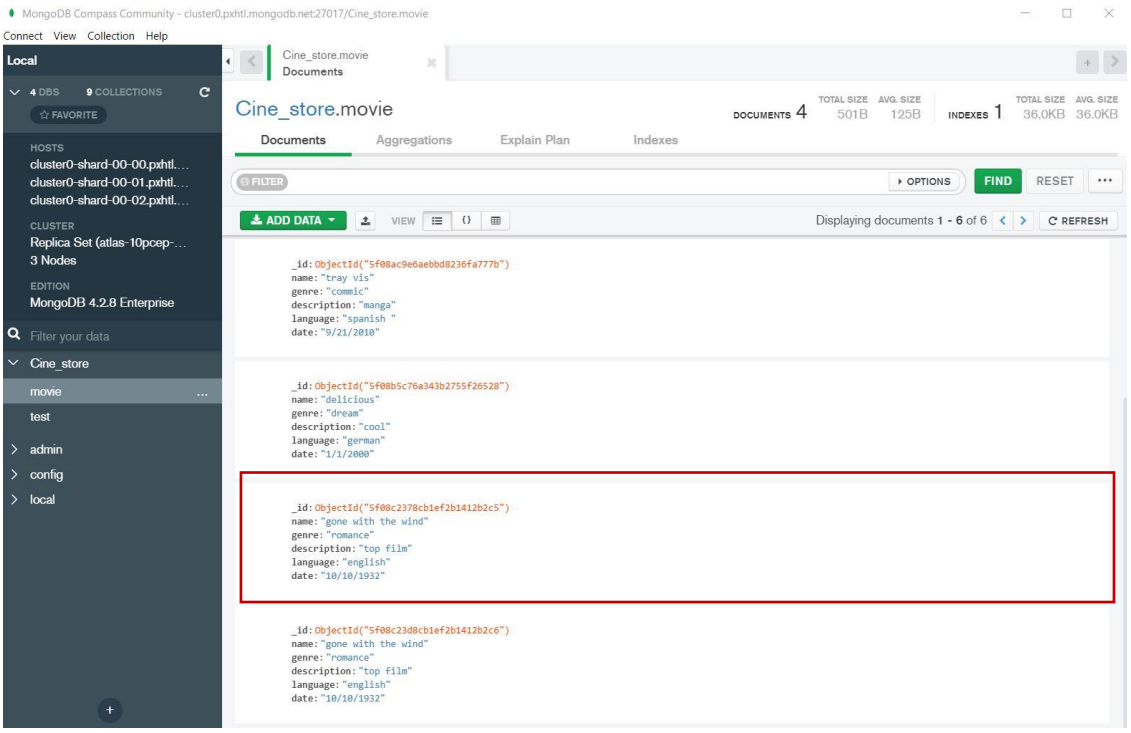
GENRE: “take the type of movie (action)”

DESCRIPTION: “any description of the movie”

LANGUAGE: “insert movie language”

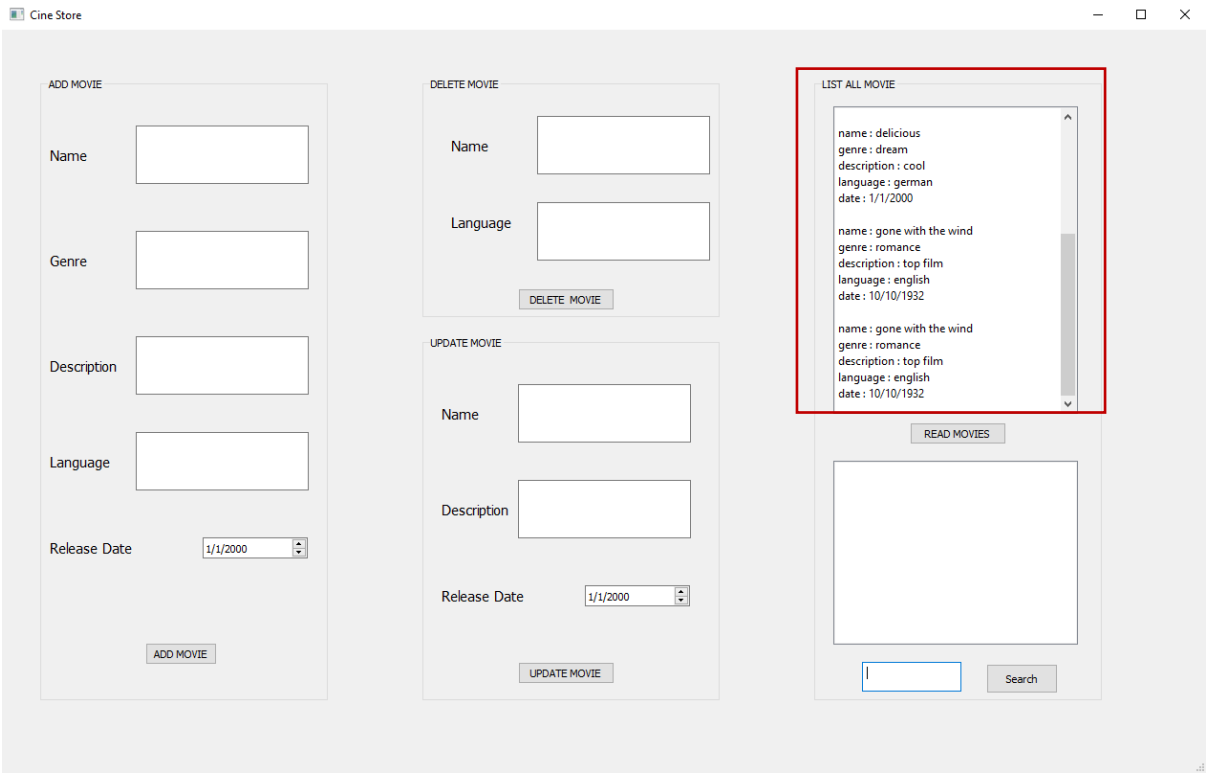
DATE:” insert the release date of the movie”

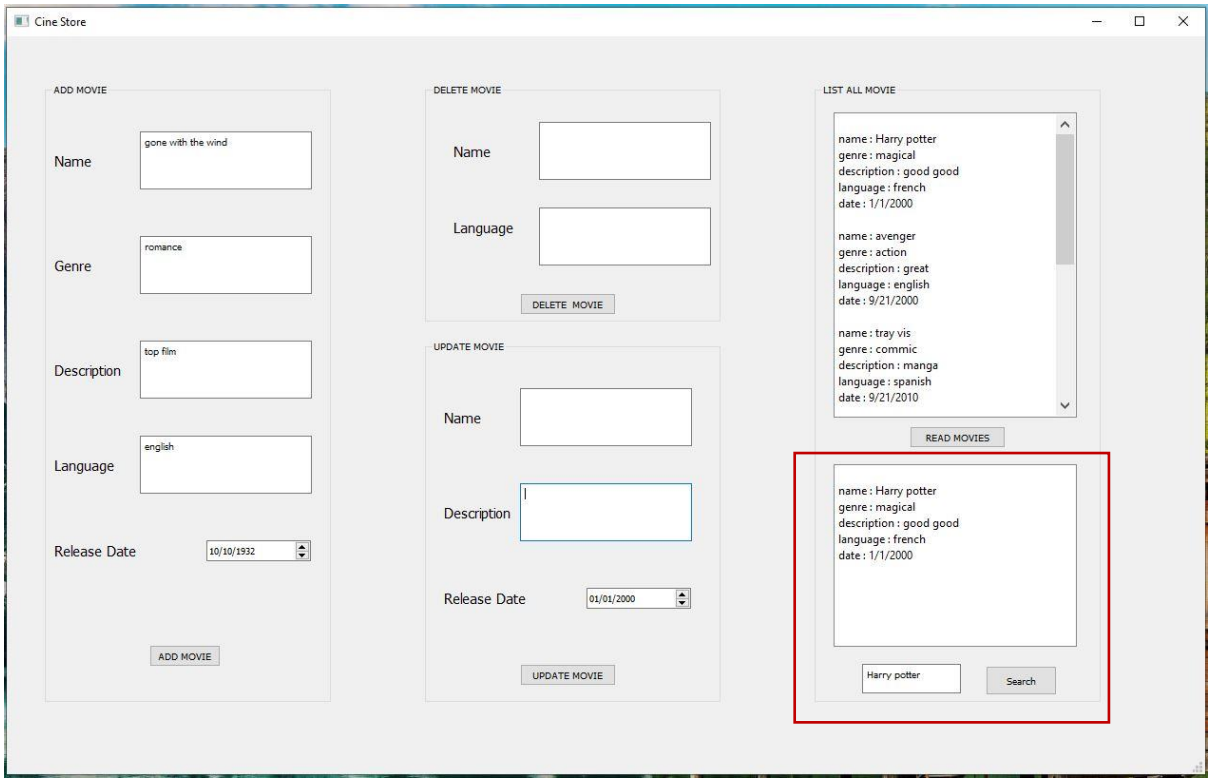




When we insert a new movie with its information including: Name, Genre, Description, Language and Date, this data will be added into MongoDB database.

Reading Information

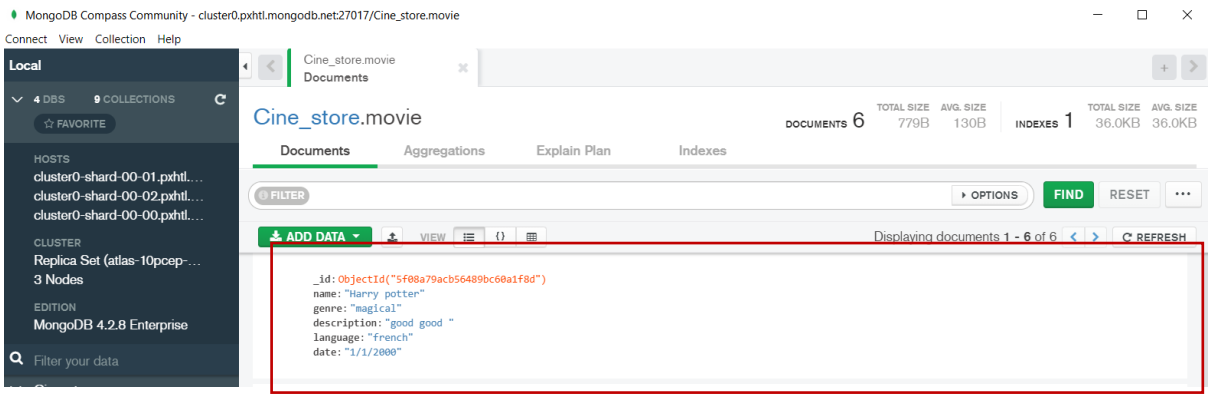




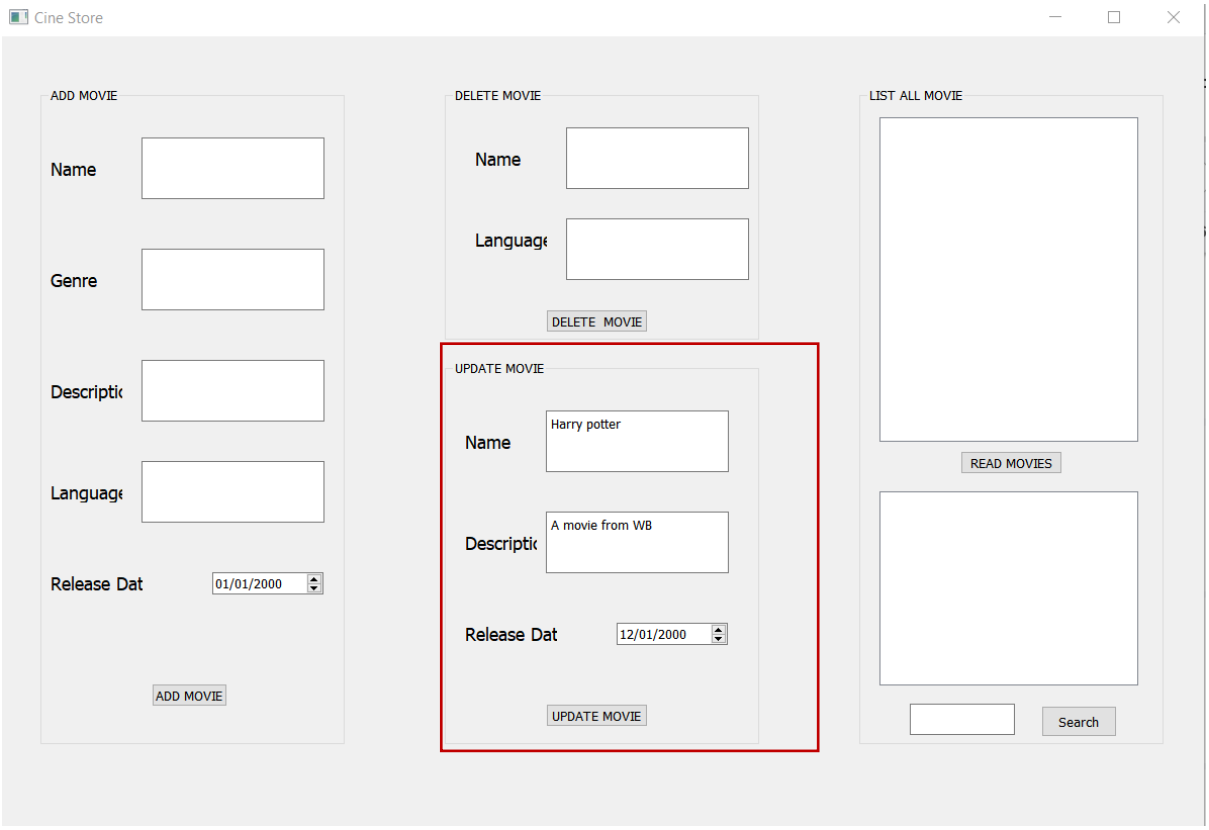
Once clicking on the Read Movies, the interface will show the list of all of film with related information. On the Reading session, we can use the search interface to find the information of a particular movie.

Updating Information

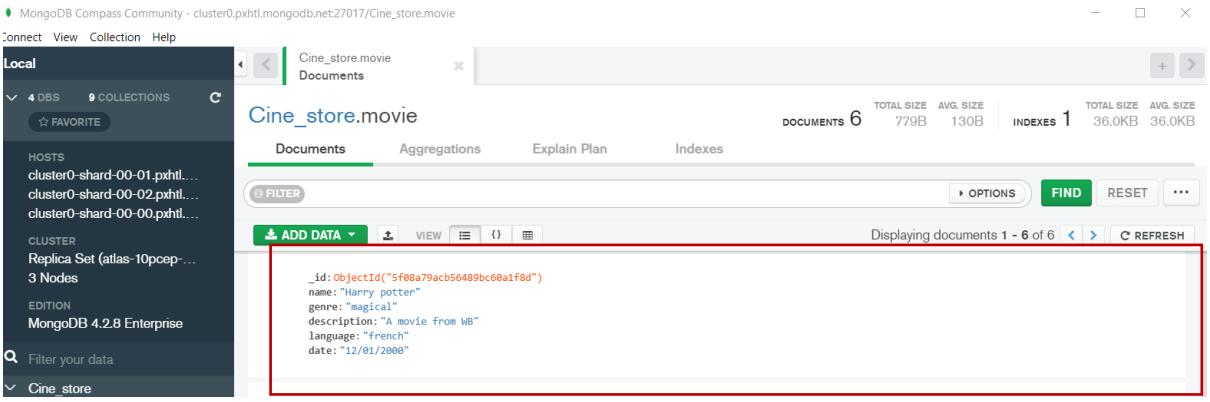
This is the Database of “Harry Potter” before updating.



When we update the movie with the description: “Harry potter” with the Description: “ A movie from WB”.

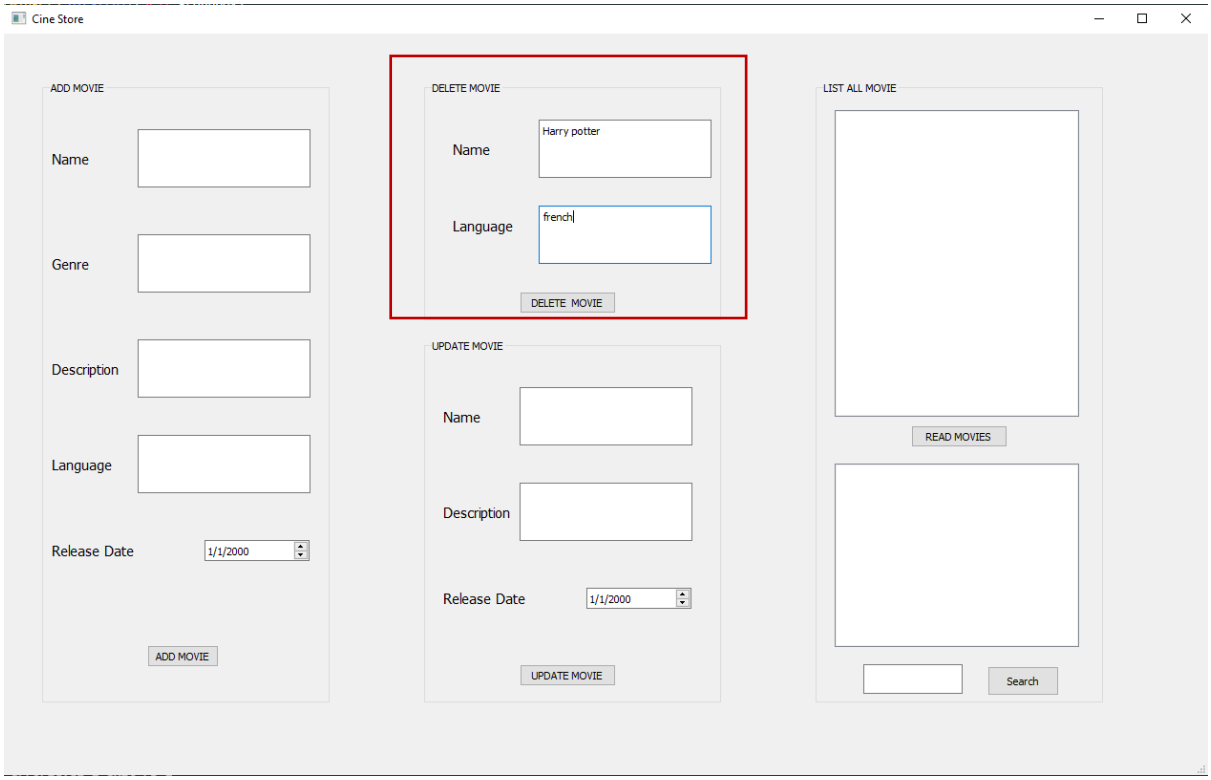


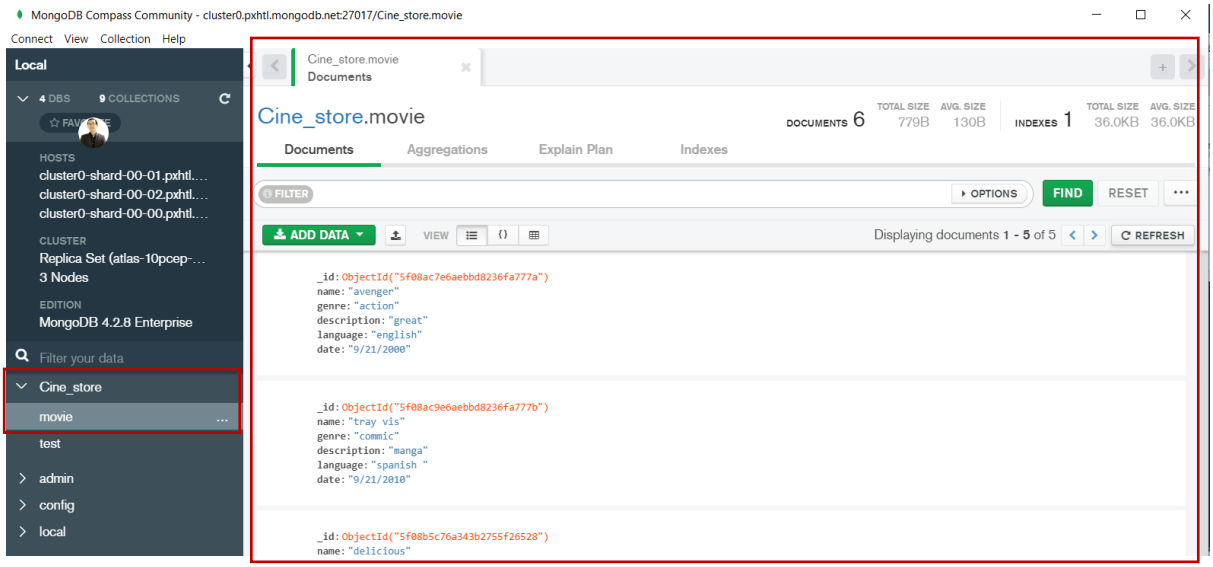
Then, the film “Harry potter” has been updated with the details “A movie from WB”



Deleting Information

We tried to delete the movie “Harry potter” by using Delete. The date of “Harry potter” was removed according my from the mongoDB.

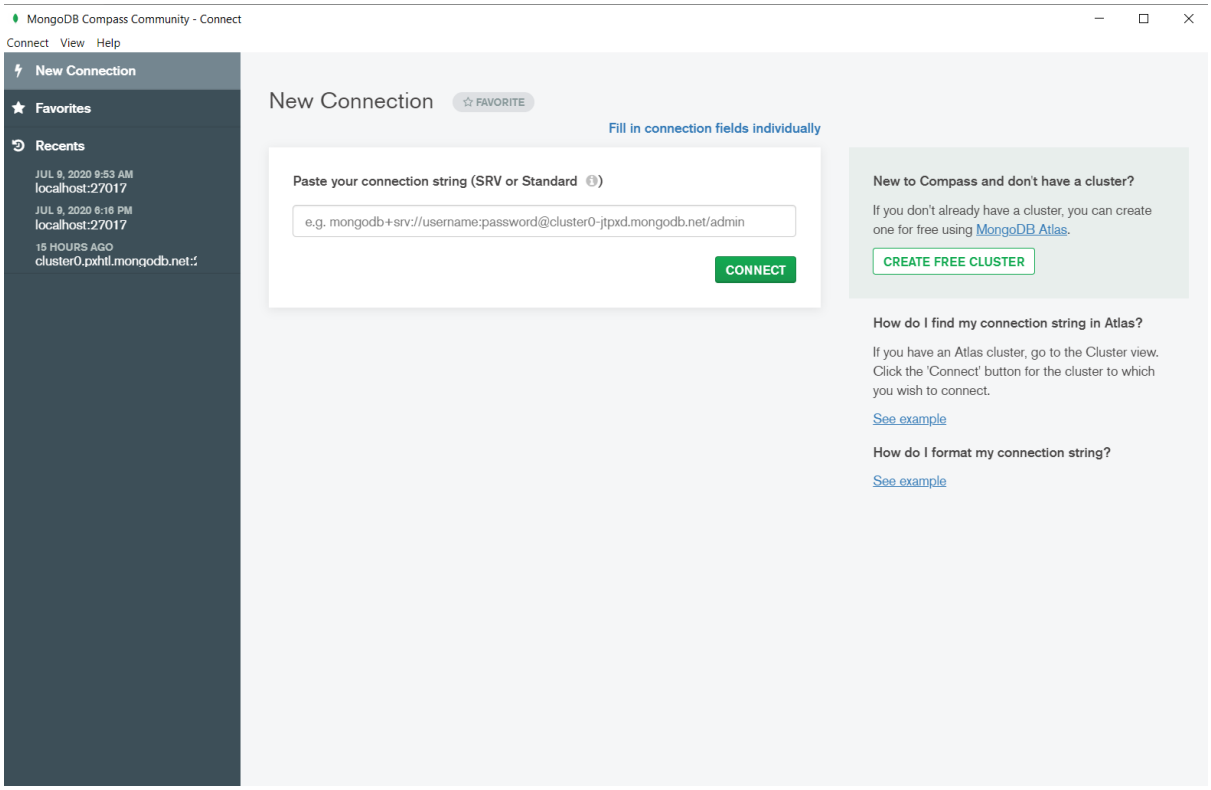




The database had removed the “Harry potter” movie.

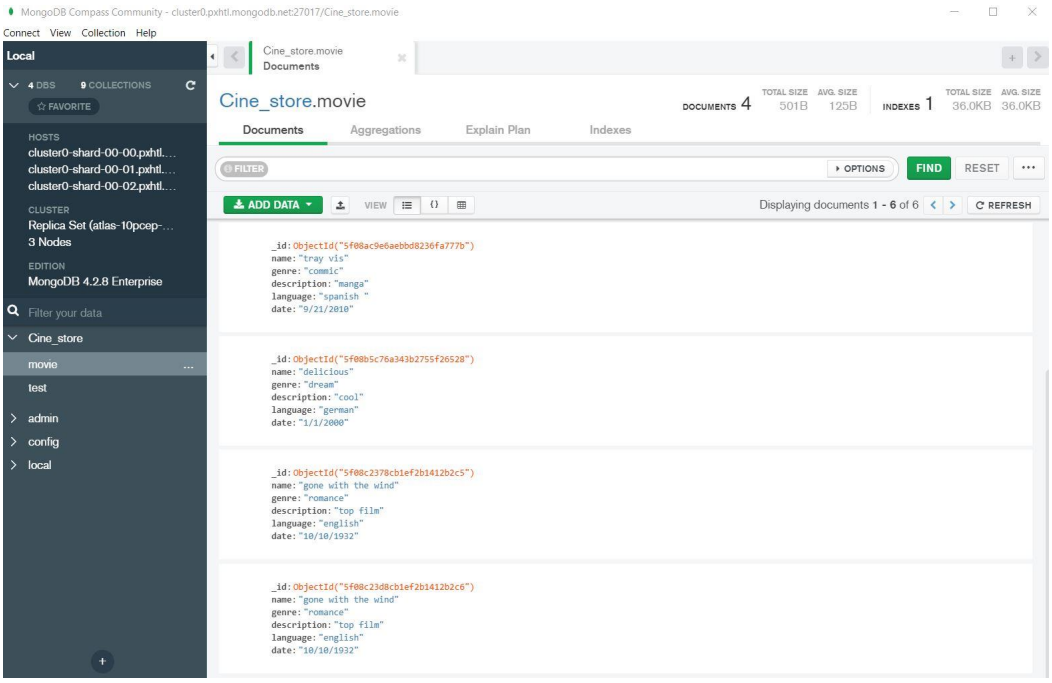
MongoDB

We used the newest version of MongoDB compass community which is 4.2



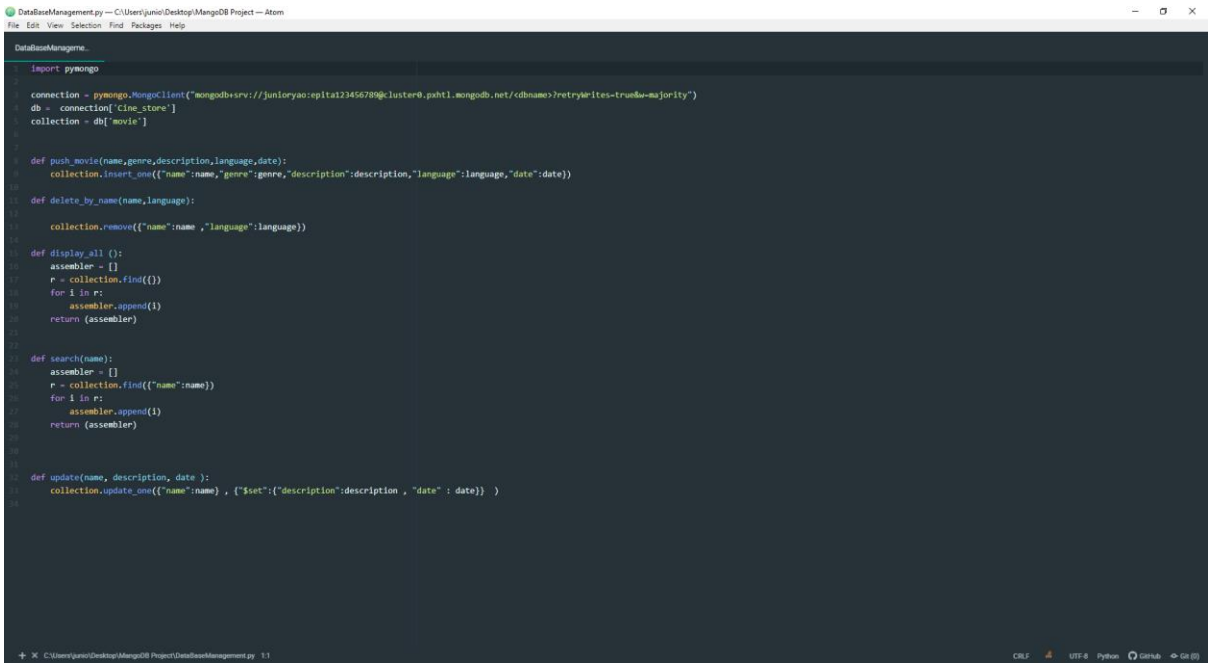
String connection that we have created:
mongodb+srv://junioryao:epita123456789@cluster0.pxhtl.mongodb.net/test

Here is the Database of Cine_store in MongoDB that we have used for the Movie application python based.

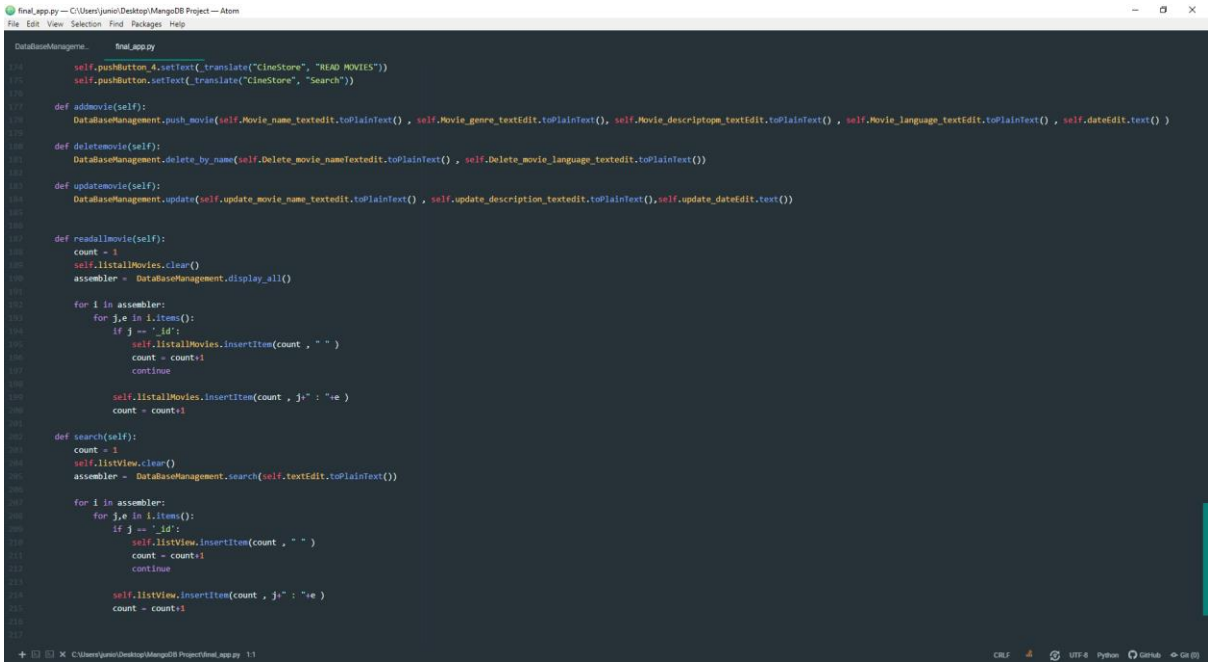


The Python Coding

In order to connect the application interface with the MangoDB , we were resquested to used PYMONGO ,DNSPYTHON packages , below are the querry to the mangoo DB server ;



```python
import pymongo

connection = pymongo.MongoClient("mongodb+srv://junioryao:epita123456789@cluster0.pxhtl.mongodb.net/<dbname>?retryWrites=true&w=majority")
db =  connection['Cine_store']
collection = db['movie']


def push_movie(name,genre,description,language,date):
    collection.insert_one({"name":name,"genre":genre,"description":description,"language":language,"date":date})

def delete_by_name(name,language):

    collection.remove({"name":name ,"language":language})

def display_all ():
    assembler = []
    r = collection.find({})
    for i in r:
        assembler.append(i)
    return (assembler)


def search(name):
    assembler = []
    r = collection.find({"name":name})
    for i in r:
        assembler.append(i)
    return (assembler)



def update(name, description, date ):
    collection.update_one({"name":name} , {"$set":{"description":description , "date" : date}}  )
```

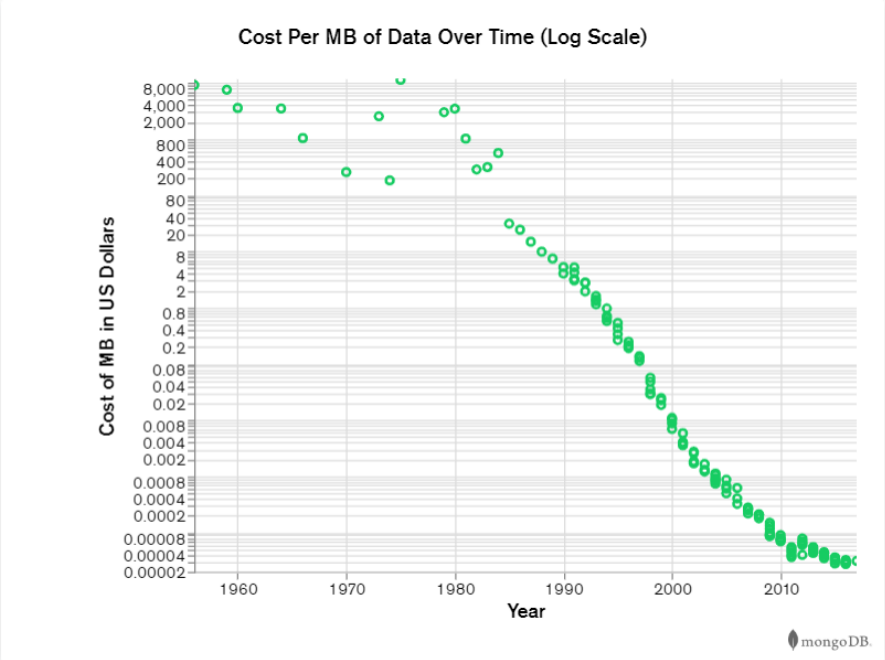


```python
        self.pushButton_4.setText(_translate("CineStore", "READ MOVIES"))
        self.pushButton.setText(_translate("CineStore", "Search"))

    def addmovie(self):
        DataBaseManagement.push_movie(self.Movie_name_textedit.toPlainText() , self.Movie_genre_textEdit.toPlainText(), self.Movie_descriptopm_textEdit.toPlainText() , self.Movie_language_textEdit.toPlainText() , self.dateEdit.text() )

    def deletemovie(self):
        DataBaseManagement.delete_by_name(self.Delete_movie_nameTextedit.toPlainText() , self.Delete_movie_language_textedit.toPlainText())

    def updatemovie(self):
        DataBaseManagement.update(self.update_movie_name_textedit.toPlainText() , self.update_description_textedit.toPlainText(),self.update_dateEdit.text())


    def readallmovie(self):
        count = 1
        self.listallMovies.clear()
        assembler =  DataBaseManagement.display_all()

        for i in assembler:
            for j,e in i.items():
                if j == '_id':
                    self.listallMovies.insertItem(count , " " )
                    count = count+1
                    continue

                self.listallMovies.insertItem(count , j+" : "+e )
                count = count+1

    def search(self):
        count = 1
        self.listView.clear()
        assembler =  DataBaseManagement.search(self.textEdit.toPlainText())

        for i in assembler:
            for j,e in i.items():
                if j == '_id':
                    self.listView.insertItem(count , " " )
                    count = count+1
                    continue

                self.listView.insertItem(count , j+" : "+e )
                count = count+1
```

Conclusion

NoSQL is a very powerful tool in database. Recently, the number of companies using NoSQL are increasing, from GAFA to any start-up around the world. The project is just a humble demonstration to see the useful and effective of MongoDB tool.

In fact, NoSQL databases emerged in the late 2000s as the cost of storage dramatically decreased. Gone were the days of needing to create a complex, difficult-to-manage data model simply for the purposes of reducing data duplication. Developers (rather than storage) were becoming the primary cost of software development, so NoSQL databases optimized for developer productivity.



Reference

1. https://en.wikipedia.org/wiki/NoSQL
2. https://www.mongodb.com/nosql-explained