

Chapter 1

Document database

In this chapter we will discuss of the organization of the document database and how we handled the replicas.

We decided to use MongoDB as DBMS for the document database for the purpose of storing the main information for movies, users, reviews and personnel. In MongoDB we created the following three collection:

- movie
- user

Inside the movie collection there are embedded the documents for the reviews and personnel

1 Collection composition

1.1 Movie collection

The following is the composition of the Movie collection.

```
1 {
2   "_id" : ObjectId(<<id_field>>),
3   "primaryTitle": "The first ever movie",
4   "year": 1989,
5   "runtimeMinutes": 70,
6   "genres": ["Crime", "Drama", "Romantic"],
7   "productionCompany": "Dingo Picture Production" ,
8   "personnel": [
9     {
10      "name": "John Doe",
11      "category": "producer",
12      "job": "writer"
13    },
14    {
15      "name": "Christopher Lee",
16      "category": "actor",
17      "character": ["The one"]
18    },
19    ...
20  ],
21  "top_critic_fresh_count": "4",
22  "top_critic_rotten_count": 0,
```

```

23     "user_fresh_count": 14,
24     "user_rotten_count": 1,
25     "top_critic_rating": 100,
26     "top_critic_status": "Fresh",
27     "user_rating": 93,
28     "user_status": "Fresh",
29     "review":
30     [
31         {
32             "critic_name": "AntonyE",
33             "review_date": "2018-12-07",
34             "review_type": "Rotten",
35             "top_critic": false,
36             "review_content": "I really didn't liked it!",
37             "review_score": "1/10"
38         },
39         {
40             "critic_name": "AntonyE",
41             "review_date": "2015-12-07",
42             "review_type": "Fresh",
43             "top_critic": true,
44             "review_content": "I really liked it!",
45             "review_score": "A-"
46         },
47         ...
48     ]
49 }

```

As previously stated the field personnel and review are arrays of embedded documents.

1.2 User collection

The following is the composition of the User collection.

```

1  {
2      "_id"                : ObjectId(<<id_field>>),
3      "username"           : "AntonyE",
4      "password"           : "hashed_password",
5      "firstName"          : "Anton",
6      "lastName"           : "Ego",
7      "registrationreview_date" : "2019-06-29",
8      "review_date of birth"    : "2002-07-16",
9      "last3Reviews":
10     [
11         {
12             "_id"                : ObjectId(<<id_field>>),
13             "primaryTitle"       : "Star Wars: A new Hope",
14             "review_date"        : "2018-12-07",
15             "review_type"       : "fresh",
16             "top_critic"        : false,
17             "review_content"     : "I really liked it!",
18             "vote"              : "8/10"

```

```

19     },
20     {
21         "_id"                : ObjectId(<<id_field>>),
22         "primaryTitle"       : "Ratatouille",
23         "review_date"        : "2019-02-17",
24         "review_type"        : "rotten",
25         "top_critic"         : false,
26         "review_content"     : "The director was also
↪ controlled by a rat",
27         "vote"               : "D+"
28     },
29     {
30         "_id"                : ObjectId(<<id_field>>),
31         "primaryTitle"       : "300",
32         "review_date"        : "2020-02-15",
33         "review_type"        : "fresh",
34         "top_critic"         : false,
35         "review_content"     : "Too much slow motion",
36         "vote"               : "3.5/5"
37     }
38 ],
39 "reviews":[
40     {
41         "movie_id" : ObjectId(<<id_field>>),
42         "primaryTitle" : "Evidence",
43         "review_index": 11
44     },
45     ...
46 ]
47
48 }

```

Is important to notice that in this collection we have two fields for the reviews, *last3Reviews* and *reviews* which present different ways of storing data for the same underling entity. As suggested by the name itself *last3Reviews* contains the last three review in the form of an embedded document meanwhile *reviews* contains all the reviews made by a user in the form of document linking towards the movie for which they were written; in particular, the link is formed by the id of the movie, the title and the position in which the review can be found in the array *review* of the movie document. This structure was chosen so that we could avoid a full replica of the reviews in both collections and, at the same time, avoiding to perform join operation for those reviews that are more frequently checked, which are the most recent ones. The idea of creating a separated collection for reviews was immediately discarded because it would have resorted in a design for the document database that resemble a third normal form of relational database.

2 Indexes

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of

the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

3 Partition and replicas

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.