TP 1 : REDIS & MongoDB

NO SQL

BOUIRDI Salma

2025/2026

Partie 1 : REDIS

INSTALATION :

```
Status: Downloaded newer image for redis:latest
452f884cc9d4132117537a274c5fe074e993cfd11eaf324af9b5f14d02911c60
ing@ing:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS                      PORTS
NAMES
452f884cc9d4   redis:latest   "docker-entrypoint.s…"   About a minute ago   Up About a minute           0.0.0.0:6379->6379/tcp, [::]:6379->6379/tcp
redis
a106be510bde   hello-world    "/hello"                 10 minutes ago       Exited (0) 10 minutes ago
unruffled_black
ing@ing:~$ sudo docker exec -it redis redis-cli
127.0.0.1:6379> PING
PONG
```

```
ing@ing:~$ sudo docker exec -it redis redis-cli
127.0.0.1:6379> PING
PONG
127.0.0.1:6379> set demo  "bonjour"
OK
127.0.0.1:6379> set demo  user:1234 "Salma"
(error) ERR syntax error
127.0.0.1:6379> set user:1234 "Salma"
OK
127.0.0.1:6379> fet user:1234
(error) ERR unknown command 'fet', with args beginning with: 'user:1234'
127.0.0.1:6379> get user:1234
"Salma"
127.0.0.1:6379> del user:1234
(integer) 1
127.0.0.1:6379> get user:1234
(nil)
127.0.0.1:6379> set 1mar 0
OK
127.0.0.1:6379> incr 1mar
(integer) 1
127.0.0.1:6379> del 1mar
```

```
127.0.0.1:6379> get 1mar
"1"
127.0.0.1:6379> set mac maval
OK
127.0.0.1:6379> ttl mac
(integer) -1
127.0.0.1:6379> expire macle 126
(integer) 0
127.0.0.1:6379>
127.0.0.1:6379> exipre mac 120
(error) ERR unknown command 'exipre', with args beginning with: 'mac' '120'
127.0.0.1:6379> expire mac 120
(integer) 1
127.0.0.1:6379> expire macle 126
(integer) 0
127.0.0.1:6379> del mac
(integer) 1
127.0.0.1:6379> get mac
(nil)
127.0.0.1:6379> get 1mar
"1"
127.0.0.1:6379> del 1mar
```

```
127.0.0.1:6379> RPUSH mesCr "nosql"
(integer) 1
127.0.0.1:6379> RPUSH mesCr "APD"
(integer) 2
127.0.0.1:6379> LRANGE mesCr 0 -1
1) "nosql"
2) "APD"
127.0.0.1:6379> LRANGE mesCr 0 -2
1) "nosql"
127.0.0.1:6379> LRANGE mesCr 0 -5
(empty array)
127.0.0.1:6379> LRANGE mesCr 1 0
(empty array)
127.0.0.1:6379> LRANGE mesCr 1 2
1) "APD"
127.0.0.1:6379> LRANGE mesCr 1 5
1) "APD"
```

depuis le 1er terminal

```
1) "APD"
127.0.0.1:6379> LRANGE mesCr 1 5
1) "APD"
127.0.0.1:6379> SUBSCRIBE mescours user:1
1) "subscribe"
2) "mescours"
3) (integer) 1
1) "subscribe"
2) "user:1"
3) (integer) 2
1) "message"
2) "mescours"
3) "hlllo"
127.0.0.1:6379(subscribed mode)>
127.0.0.1:6379(subscribed mode)>
127.0.0.1:6379> PSUBSCRIBE mes*
1) "psubscribe"
2) "mes*"
3) (integer) 1
1) "pmessage"
2) "mes*"
3) "mesnotes"
4) "helllo"
Reading messages... (press Ctrl-C to quit or any key to type command)
```

depuis le 2ème terminal

```
ing@ing:~$ docker exec -it redis redis-cli
permission denied while trying to connect to the docker API at unix:///var/run/docker.sock
ing@ing:~$ sudo docker exec -it redis redis-cli
[sudo] Mot de passe de ing :
127.0.0.1:6379> PUBLISH mescours "hlllo"
(integer) 1
127.0.0.1:6379> PUBLISH mesnotes "helllo"
(integer) 1
127.0.0.1:6379> PUBLISH otes "helllo"
(integer) 0
127.0.0.1:6379> keys *
1) "mesCr"
2) "demo"
127.0.0.1:6379> select 1
OK
127.0.0.1:6379[1]>
```

PARTIE 2 : mongodb

Partie 1 – Filtrer et projeter les données :

```
ing@ing:~/Documents/NoSql/tp1$ docker cp /home/ing/Documents/NoSql/tp1/sampledat
a.archive mongodb:/sampledata.archive
Successfully copied 0B to mongodb:/sampledata.archive
permission denied while trying to connect to the docker API at unix:///var/run/d
ocker.sock
ing@ing:~/Documents/NoSql/tp1$ sudo sudo docker cp /home/ing/Documents/NoSql/tp1
/sampledata.archive mongodb:/sampledata.archive
Successfully copied 370MB to mongodb:/sampledata.archive
ing@ing:~/Documents/NoSql/tp1$
```

```
ing@ing:~/Documents/NoSql/tp1$ sudo docker exec -it mongodb mongosh
Current Mongosh Log ID: 6928465deef1ab1df09dc29c
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverS
electionTimeoutMS=2000&appName=mongosh+2.5.9
Using MongoDB:          8.2.2
Using Mongosh:          2.5.9

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-11-27T09:10:20.257+00:00: Using the XFS filesystem is strongly recommend
ed with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodno
tes-filesystem
   2025-11-27T09:10:20.347+00:00: Access control is not enabled for the database
. Read and write access to data and configuration is unrestricted
   2025-11-27T09:10:20.347+00:00: Soft rlimits for open file descriptors too low
   2025-11-27T09:10:20.348+00:00: For customers running the current memory alloc
ator, we suggest changing the contents of the following sysfsFile
   2025-11-27T09:10:20.348+00:00: We suggest setting the contents of sysfsFile t
o 0.
```

```
test> show dbs
admin                40.00 KiB
config               80.00 KiB
local                40.00 KiB
sample_airbnb        51.77 MiB
sample_analytics      9.39 MiB
sample_geospatial   980.00 KiB
sample_guides        40.00 KiB
sample_mflix         94.48 MiB
sample_restaurants    5.61 MiB
sample_supplies     968.00 KiB
sample_training      40.16 MiB
sample_weatherdata    2.38 MiB
test>
```

1. Afficher les 5 films sortis depuis 2015.

```
ıple_mflix> db.movies.find({ year: { $gte: 2015 } }).limit(5)

[
  _id: ObjectId('573a13adf29313caabd2b765'),
  plot: "A new theme park is built on the original site of Jurassic Park. Everything is going well until the park's newest attraction--a geneticall
ɔdified giant stealth killing machine--escapes containment and goes on a killing spree.",
  genres: [ 'Action', 'Adventure', 'Sci-Fi' ],
  runtime: 124,
  metacritic: 59,
  rated: 'PG-13',
  cast: [
    'Chris Pratt',
    'Bryce Dallas Howard',
    'Irrfan Khan',
    "Vincent D'Onofrio"
  ],
  num_mflix_comments: 0,
  poster: 'https://m.media-amazon.com/images/M/MV5BNzQ3OTY4NjAtNzM5OS00N2ZhLWJlOWUtYzYwZjNmOWRiMzcyXkEyXkFqcGdeQXVyMTMxODk2OTU@._V1_SY1000_SX677_AL
ɔg',
  title: 'Jurassic World',
  fullplot: '22 years after the original Jurassic Park failed, the new park (also known as Jurassic World) is open for business. After years of stu
ıg genetics the scientists on the park genetically engineer a new breed of dinosaur. When everything goes horribly wrong, will our heroes make it o
the island?',
  languages: [ 'English' ],
  released: ISODate('2015-06-12T00:00:00.000Z'),
  directors: [ 'Colin Trevorrow' ],
  writers: [
    'Rick Jaffa (screenplay)',
    'Amanda Silver (screenplay)',
    'Colin Trevorrow (screenplay)',
```

2. Trouver tous les films dont le genre est "Comedy" .

```
sample_mflix>  db.movies.find({ genres: "Comedy" })
[
  {
    _id: ObjectId('573a1390f29313caabcd4803'),
    plot: 'Cartoon figures announce, via comic strip balloons, that they will move - and move they do, in a wildly exaggerated style.',
    genres: [ 'Animation', 'Short', 'Comedy' ],
    runtime: 7,
    cast: [ 'Winsor McCay' ],
    num_mflix_comments: 0,
    poster: 'https://m.media-amazon.com/images/M/MV5BYzg2NjNhNTctMjUxMi00ZWU4LWI3ZjYtNTI0NTQxNThjZTk2XkEyXkFqcGdeQXVyNzg5OTk2OA@@._V1_SY1000_SX
.jpg',
    title: 'Winsor McCay, the Famous Cartoonist of the N.Y. Herald and His Moving Comics',
    fullplot: 'Cartoonist Winsor McCay agrees to create a large set of drawings that will be photographed and made into a motion picture. The j
ires plenty of drawing supplies, and the cartoonist must also overcome some mishaps caused by an assistant. Finally, the work is done, and ever
n see the resulting animated picture.',
    languages: [ 'English' ],
    released: ISODate('1911-04-08T00:00:00.000Z'),
    directors: [ 'Winsor McCay', 'J. Stuart Blackton' ],
    writers: [
      'Winsor McCay (comic strip "Little Nemo in Slumberland")',
      'Winsor McCay (screenplay)'
    ],
    awards: { wins: 1, nominations: 0, text: '1 win.' },
    lastupdated: '2015-08-29 01:09:03.030000000',
    year: 1911,
    imdb: { rating: 7.3, votes: 1034, id: 1737 },
    countries: [ 'USA' ],
    type: 'movie',
    tomatoes: {
      viewer: { rating: 3.4, numReviews: 89, meter: 47 },
      lastUpdated: ISODate('2015-08-20T18:51:24.000Z')
    }
  },
  {
    _id: ObjectId('573a1390f29313caabcd50e5'),
    plot: 'The cartoonist, Winsor McCay, brings the Dinosaurus back to life in the figure of his latest creation, Gertie the Dinosaur.',
    genres: [ 'Animation', 'Short', 'Comedy' ],
    runtime: 12
```

3. Afficher les sortis entre entre 2000 et 2005.

```
sample_mflix> db.movies.find({year: {$gte: 2000, $lte: 2005}}, {title: 1, year:
... 1}).pretty()
[
  {
    _id: ObjectId('573a1393f29313caabcdcb42'),
    title: 'Kate & Leopold',
    year: 2001
  },
  {
    _id: ObjectId('573a1398f29313caabceb1fe'),
    title: 'Crime and Punishment',
    year: 2002
  },
  {
    _id: ObjectId('573a139af29313caabcf0718'),
    year: 2001,
    title: 'Glitter'
  },
  {
    _id: ObjectId('573a139af29313caabcf0782'),
    year: 2000,
    title: 'In the Mood for Love'
  },
  {
    _id: ObjectId('573a139af29313caabcf0809'),
    year: 2003,
    title: 'The Manson Family'
  },
  {
    _id: ObjectId('573a139af29313caabcf0869'),
    title: 'The Dancer Upstairs',
    year: 2002
  },
  {
    _id: ObjectId('573a139af29313caabcf0d0b'),
    year: 2000,
    title: 'State and Main'
```

3. Afficher les sortis entre entre 2000 et 2005.

```
sample_mflix> db.movies.find({genres: {$all: ["Drama", "Romance"]}}, {title: 1, genres:
... 1})
[
  {
    _id: ObjectId('573a1391f29313caabcd70b4'),
    genres: [ 'Drama', 'Romance', 'War' ],
    title: 'The Four Horsemen of the Apocalypse'
  },
  {
    _id: ObjectId('573a1391f29313caabcd7a34'),
    genres: [ 'Drama', 'Romance' ],
    title: 'A Woman of Paris: A Drama of Fate'
  },
  {
    _id: ObjectId('573a1391f29313caabcd7b98'),
    genres: [ 'Drama', 'Romance', 'Thriller' ],
    title: 'He Who Gets Slapped'
  },
  {
    _id: ObjectId('573a1391f29313caabcd7da6'),
    genres: [ 'Drama', 'Romance' ],
    title: 'Wild Oranges'
  },
  {
    _id: ObjectId('573a1391f29313caabcd89aa'),
    genres: [ 'Drama', 'Romance', 'War' ],
    title: 'Wings'
  },
  {
    _id: ObjectId('573a1391f29313caabcd91d7'),
    genres: [ 'Drama', 'Romance', 'Thriller' ],
    title: 'The Big House'
  },
  {
    _id: ObjectId('573a1391f29313caabcd9264'),
    genres: [ 'Romance', 'Drama' ],
    title: 'The Divorcee'
```

5. Afficher les films sans champ rated .

```
sample_mflix> db.movies.find({rated: {$exists: false}}, {title: 1})
[
  {
    _id: ObjectId('573a1390f29313caabcd4803'),
    title: 'Winsor McCay, the Famous Cartoonist of the N.Y. Herald and His Moving Comics'
  },
  {
    _id: ObjectId('573a1390f29313caabcd50e5'),
    title: 'Gertie the Dinosaur'
  },
  {
    _id: ObjectId('573a1390f29313caabcd516c'),
    title: 'In the Land of the Head Hunters'
  },
  {
    _id: ObjectId('573a1390f29313caabcd5293'),
    title: 'The Perils of Pauline'
  },
  { _id: ObjectId('573a1390f29313caabcd5a93'), title: 'Civilization' },
  {
    _id: ObjectId('573a1390f29313caabcd6223'),
    title: 'The Poor Little Rich Girl'
  },
  {
    _id: ObjectId('573a1390f29313caabcd6377'),
    title: 'Wild and Woolly'
  },
  { _id: ObjectId('573a1390f29313caabcd63d6'), title: 'The Blue Bird' },
  { _id: ObjectId('573a1391f29313caabcd6ea2'), title: 'The Saphead' },
  {
    _id: ObjectId('573a1391f29313caabcd71e3'),
    title: 'Miss Lulu Bett'
  },
  {
    _id: ObjectId('573a1391f29313caabcd72f0'),
    title: "Tol'able David"
  },
```

6. Afficher le nombre de films par année.

```
sample_mflix> db.movies.aggregate([
...   {$group: {_id: "$year", total: {$sum: 1}}},
...   {$sort: {_id: 1}}
... ])
[
  { _id: 1896, total: 2 }, { _id: 1903, total: 1 },
  { _id: 1909, total: 1 }, { _id: 1911, total: 2 },
  { _id: 1913, total: 1 }, { _id: 1914, total: 3 },
  { _id: 1915, total: 2 }, { _id: 1916, total: 2 },
  { _id: 1917, total: 2 }, { _id: 1918, total: 1 },
  { _id: 1919, total: 1 }, { _id: 1920, total: 4 },
  { _id: 1921, total: 5 }, { _id: 1922, total: 3 },
  { _id: 1923, total: 2 }, { _id: 1924, total: 6 },
  { _id: 1925, total: 3 }, { _id: 1926, total: 6 },
  { _id: 1927, total: 4 }, { _id: 1928, total: 8 }
]
Type "it" for more
sample_mflix>
```

7. Afficher la moyenne des notes IMDb par genre.

```
sample_mflix> db.movies.aggregate([
...   {$unwind: "$genres"},
...   {$group: {_id: "$genres", moyenne: {$avg: "$imdb.rating"}}},
...   {$sort: {moyenne: -1}}
... ])
[
  { _id: 'Film-Noir', moyenne: 7.397402597402598 },
  { _id: 'Short', moyenne: 7.377574370709382 },
  { _id: 'Documentary', moyenne: 7.365679824561403 },
  { _id: 'News', moyenne: 7.252272727272728 },
  { _id: 'History', moyenne: 7.1696100917431185 },
  { _id: 'War', moyenne: 7.128591954022989 },
  { _id: 'Biography', moyenne: 7.087984189723319 },
  { _id: 'Talk-Show', moyenne: 7 },
  { _id: 'Animation', moyenne: 6.89669603524229 },
  { _id: 'Music', moyenne: 6.883333333333334 },
  { _id: 'Western', moyenne: 6.823553719008264 },
  { _id: 'Drama', moyenne: 6.803377338624768 },
  { _id: 'Sport', moyenne: 6.749041095890411 },
  { _id: 'Crime', moyenne: 6.688585405625764 },
  { _id: 'Musical', moyenne: 6.665831435079727 },
  { _id: 'Romance', moyenne: 6.6564272782136396 },
  { _id: 'Mystery', moyenne: 6.527425044091711 },
  { _id: 'Adventure', moyenne: 6.493680884676145 },
  { _id: 'Comedy', moyenne: 6.450214658080344 },
  { _id: 'Fantasy', moyenne: 6.3829847908745245 }
]
Type "it" for more
```

8. Afficher le nombre de films par pays.

```
sample_mflix> db.movies.aggregate([
...   {$unwind: "$countries"},
...   {$group: {_id: "$countries", total: {$sum: 1}}},
...   {$sort: {total: -1}}
... ])
[
  { _id: 'USA', total: 10921 },
  { _id: 'UK', total: 2652 },
  { _id: 'France', total: 2647 },
  { _id: 'Germany', total: 1494 },
  { _id: 'Canada', total: 1260 },
  { _id: 'Italy', total: 1217 },
  { _id: 'Japan', total: 786 },
  { _id: 'Spain', total: 675 },
  { _id: 'India', total: 564 },
  { _id: 'Australia', total: 470 },
  { _id: 'Sweden', total: 402 },
  { _id: 'Belgium', total: 364 },
  { _id: 'Hong Kong', total: 357 },
  { _id: 'Netherlands', total: 337 },
  { _id: 'Finland', total: 322 },
  { _id: 'Denmark', total: 308 },
  { _id: 'Russia', total: 290 },
  { _id: 'South Korea', total: 278 },
  { _id: 'China', total: 275 },
  { _id: 'West Germany', total: 246 }
]
Type "it" for more
```

9. Afficher les top 5 réalisateurs.

```
sample_mflix> db.movies.aggregate([
...   {$unwind: "$directors"},
...   {$group: {_id: "$directors", total: {$sum: 1}}},
...   {$sort: {total: -1}},
...   {$limit: 5}
... ])
[
  { _id: 'Woody Allen', total: 40 },
  { _id: 'Martin Scorsese', total: 32 },
  { _id: 'Takashi Miike', total: 31 },
  { _id: 'Sidney Lumet', total: 29 },
  { _id: 'Steven Spielberg', total: 29 }
]
```

10. Afficher les films triés par note IMDb.

```
sample_mflix> db.movies.aggregate([
...   {$sort: {"imdb.rating": -1}},
...   {$project: {title: 1, "imdb.rating": 1}}
... ])
[
  {
    _id: ObjectId('573a13d7f29313caabda5079'),
    title: 'The Deposit',
    imdb: { rating: '' }
  },
  {
    _id: ObjectId('573a13eff29313caabdd87c5'),
    title: 'Learning to Ride',
    imdb: { rating: '' }
  },
  {
    _id: ObjectId('573a13e0f29313caabdbb1ea'),
    title: 'Ad Inexplorata',
    imdb: { rating: '' }
  },
  {
    _id: ObjectId('573a13ecf29313caabdd1fc2'),
    title: 'American Hostage',
    imdb: { rating: '' }
  },
  {
    _id: ObjectId('573a13f3f29313caabdde93d'),
    title: 'The Birth of Sakè',
    imdb: { rating: '' }
  },
  {
    _id: ObjectId('573a13edf29313caabdd41ff')
```

Partie 3 – Mises à jour

11. Ajouter un champ etat :

```
sample_mflix> db.movies.updateOne({title: "Jaws"}, {$set: {etat: "culte"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

12. Incrémenter les votes IMDb de 100, par exemple.

```
sample_mflix> db.movies.updateOne({title: "Inception"}, {$inc: {"imdb.votes": 10
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

13. Supprimer le le champ poster

```
sample_mflix> db.movies.updateMany({}, {$unset: {poster: ""}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 21349,
  modifiedCount: 18044,
  upsertedCount: 0
}
```

14. Modifier le réalisateur.

```
sample_mflix> db.movies.updateOne({ title: "Titanic" }, { $set: { directors: ["James Cameron"] } })
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Partie 4 : – Requêtes complexes

15. Afficher les films les mieux notés par décennie.

```
...   {
...     $group: {
...       _id: "$decade",
...       bestMovie: { $first: "$title" },
...       maxRating: { $first: "$rating" }
...     }
...   },
...
...   // Trier par décennie
...   { $sort: { _id: 1 } }
... ])
...
[
  { _id: 1890, bestMovie: 'The Kiss', maxRating: 5.9 },
  { _id: 1900, bestMovie: 'The Great Train Robbery', maxRating: 7.4 },
  { _id: 1910, bestMovie: 'The Perils of Pauline', maxRating: 7.6 },
  { _id: 1920, bestMovie: 'One Week', maxRating: 8.3 },
  { _id: 1930, bestMovie: 'Modern Times', maxRating: 8.6 },
  { _id: 1940, bestMovie: "It's a Wonderful Life", maxRating: 8.6 },
  { _id: 1950, bestMovie: 'Rear Window', maxRating: 8.6 },
  { _id: 1960, bestMovie: 'Father of a Soldier', maxRating: 8.8 },
  { _id: 1970, bestMovie: 'The Godfather', maxRating: 9.2 },
  { _id: 1980, bestMovie: 'Cosmos', maxRating: 9.3 },
  { _id: 1990, bestMovie: 'The Civil War', maxRating: 9.4 },
  { _id: 2000, bestMovie: 'Band of Brothers', maxRating: 9.6 },
  {
    _id: 2010,
    bestMovie: 'A Brave Heart: The Lizzie Velasquez Story',
    maxRating: 9.4
  }
]
```

16. Afficher les films dont le titre commence par “Star”.

```
sample_mflix> db.movies.find({title: /^Star/}, {title: 1})
[
  { _id: ObjectId('573a1395f29313caabce10cc'), title: 'Stars' },
  { _id: ObjectId('573a1396f29313caabce37ff'), title: 'Star!' },
  {
    _id: ObjectId('573a1396f29313caabce4248'),
    title: 'Start the Revolution Without Me'
  },
  { _id: ObjectId('573a1396f29313caabce57f7'), title: 'Stardust' },
  {
    _id: ObjectId('573a1397f29313caabce68f6'),
    title: 'Star Wars: Episode IV - A New Hope'
  },
  { _id: ObjectId('573a1397f29313caabce7509'), title: 'Starcrash' },
  { _id: ObjectId('573a1397f29313caabce750b'), title: 'Starting Over' },
  {
    _id: ObjectId('573a1397f29313caabce7546'),
    title: 'Star Trek: The Motion Picture'
  },
  {
    _id: ObjectId('573a1397f29313caabce77d9'),
    title: 'Star Wars: Episode V - The Empire Strikes Back'
  },
  {
    _id: ObjectId('573a1397f29313caabce7b29'),
    title: 'Stardust Memories'
  },
  {
    _id: ObjectId('573a1397f29313caabce8744'),
    title: 'Star Trek II: The Wrath of Khan'
```

17. afficher les films avec plus de 2 genres.

```
sample_mflix> db.movies.find({$where: "this.genres.length > 2"}, {title: 1, genres: 1})
[
  {
    _id: ObjectId('573a1390f29313caabcd4803'),
    genres: [ 'Animation', 'Short', 'Comedy' ],
    title: 'Winsor McCay, the Famous Cartoonist of the N.Y. Herald and His Moving Comics
  },
  {
    _id: ObjectId('573a1390f29313caabcd50e5'),
    genres: [ 'Animation', 'Short', 'Comedy' ],
    title: 'Gertie the Dinosaur'
  },
  {
    _id: ObjectId('573a1390f29313caabcd587d'),
    genres: [ 'Biography', 'Crime', 'Drama' ],
    title: 'Regeneration'
  },
  {
    _id: ObjectId('573a1390f29313caabcd6223'),
    genres: [ 'Comedy', 'Drama', 'Family' ],
    title: 'The Poor Little Rich Girl'
  },
  {
    _id: ObjectId('573a1390f29313caabcd6377'),
    genres: [ 'Comedy', 'Western', 'Romance' ],
    title: 'Wild and Woolly'
  },
  {
    _id: ObjectId('573a1391f29313caabcd68d0'),
    genres: [ 'Comedy', 'Short', 'Action' ],
    title: 'From Hand to Mouth'
  },
  {
    _id: ObjectId('573a1391f29313caabcd6f98'),
    genres: [ 'Crime', 'Drama', 'Mystery' ],
    title: 'The Ace of Hearts'
```

```
Type "it" for more
sample_mflix> db.movies.find({directors: "Christopher Nolan"}, {title: 1, year: 1,
... "imdb.rating": 1})
[
  {
    _id: ObjectId('573a139df29313caabcf8dd4'),
    imdb: { rating: 7.6 },
    year: 1998,
    title: 'Following'
  },
  {
    _id: ObjectId('573a13a0f29313caabd05acc'),
    imdb: { rating: 8.5 },
    year: 2000,
    title: 'Memento'
  },
  {
    _id: ObjectId('573a13a5f29313caabd1605f'),
    imdb: { rating: 7.2 },
    year: 2002,
    title: 'Insomnia'
  },
  {
    _id: ObjectId('573a13aef29313caabd2c349'),
    title: 'Batman Begins',
    year: 2005,
    imdb: { rating: 8.3 }
  },
  {
    _id: ObjectId('573a13b5f29313caabd42722'),
    imdb: { rating: 9 },
    year: 2008,
    title: 'The Dark Knight'
  },
  {
    _id: ObjectId('573a13b6f29313caabd45b78'),
    imdb: { rating: 8.5 }
```

Partie 5 – Indexation

19. Créer un index sur year

```
]
sample_mflix> db.movies.createIndex({year: 1})
year_1
```

20. Vérifier les index existants.

```
sample_mflix> db.movies.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { _fts: 'text', _ftsx: 1 },
    name: 'cast_text_fullplot_text_genres_text_title_text',
    weights: { cast: 1, fullplot: 1, genres: 1, title: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  },
  { v: 2, key: { year: 1 }, name: 'year_1' }
]
```

21. Comparer deux requêtes avec et sans index (utiliser explain()).

```
sample_mflix> db.movies.find({year: 1995}).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'sample_mflix.movies',
    parsedQuery: { year: { '$eq': 1995 } },
    indexFilterSet: false,
    queryHash: '82E400C1',
    planCacheShapeHash: '82E400C1',
    planCacheKey: '5F2A8919',
    optimizationTimeMillis: 0,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { year: 1 },
        indexName: 'year_1',
        isMultiKey: false,
        multiKeyPaths: { year: [] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { year: [ '[1995, 1995]' ] }
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 372,
```

22. Supprimer l'index sur year .

```
sample_mflix> db.movies.dropIndex({year: 1})
{ nIndexesWas: 3, ok: 1 }
sample_mflix>
```

23. créer un index composé sur year et imdb.rating .

```
sample_mflix> db.movies.createIndex({year: 1, "imdb.rating": -1})
year_1_imdb.rating_-1
sample_mflix> db.movies.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  {
    v: 2,
    key: { _fts: 'text', _ftsx: 1 },
    name: 'cast_text_fullplot_text_genres_text_title_text',
    weights: { cast: 1, fullplot: 1, genres: 1, title: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  },
  {
    v: 2,
    key: { year: 1, 'imdb.rating': -1 },
    name: 'year_1_imdb.rating_-1'
  }
]
sample_mflix>
```