4.Problem Statement: Create a collection named **rating** that contain 5 documents of the following prototype and solve the following Queries.

```
{
  movie_id: 123,
  user_id: 12,
  title: Toy Story(1995),
  status: 'A'
}
```

```
> db.movie.insert({movie_id:123,user_id:12,title:"Toy Story(1995)",status:'A',rating:1});
WriteResult({ "nInserted" : 1 })
> db.movie.insert({movie_id:1,user_id:2,title:"Horror(1920)",status:'A',rating:2});
WriteResult({ "nInserted" : 1 })
> db.movie.insert({movie_id:2,user_id:2,title:"Horror(1920)",status:'A',rating:2});
WriteResult({ "nInserted" : 1 })
> db.movie.insert({movie_id:3,user_id:4,title:"Comedy(1820)",status:'B',rating:5});
WriteResult({ "nInserted" : 1 })
> db.movie.insert({movie_id:4,user_id:5,title:"Heropanti(2000)",status:'C',rating:5});
WriteResult({ "nInserted" : 1 })
> db.movie.find().pretty();
{
        "_id" : ObjectId("6538e3d7d2224cb9b39f105f"),
        "movie_id" : 123,
        "user_id" : 12,
        "title" : "Toy Story(1995)",
        "status" : "A",
        "rating" : 1
}
{
        "_id" : ObjectId("6538e3f8d2224cb9b39f1060"),
        "movie_id" : 1,
        "user_id" : 2,
        "title" : "Horror(1920)",
        "status" : "A",
        "rating" : 2
}
{
        "_id" : ObjectId("6538e444d2224cb9b39f1061"),
        "movie_id" : 2,
        "user_id" : 2,
        "title" : "Horror(1920)",
        "status" : "A",
        "rating" : 2
}
{
        "_id" : ObjectId("6538e45bd2224cb9b39f1062"),
        "movie_id" : 3,
        "user_id" : 4,
        "title" : "Comedy(1820)",
        "status" : "B",
        "rating" : 5
```

```
}
{
        "_id" : ObjectId("6538e46dd2224cb9b39f1063"),
        "movie_id" : 4,
        "user_id" : 5,
        "title" : "Heropanti(2000)",
        "status" : "C",
        "rating" : 5
}
>
```

a.creating an index on movie_id and sorts the keys in the index in ascending order.Verify query plan.

```
> db.rating.ensureIndex({movie_id:1},{unique:true});
{
        "numIndexesBefore" : 3,
        "numIndexesAfter" : 3,
        "note" : "all indexes already exist",
        "ok" : 1
}

> db.rating.find().explain();
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "library.rating",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "winningPlan" : {
                        "stage" : "COLLSCAN",
                        "direction" : "forward"
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "comp-ThinkCentre-M920q",
                "port" : 27017,
                "version" : "3.6.8",
                "gitVersion" : "8e540c0b6db93ce994cc548f000900bdc740f80a"
        },
        "ok" : 1
}
```

b.Show various indexes created on movie collection.

```
> db.rating.ensureIndex({movie_id:1,unique:true});
{
        "ok" : 0,
        "errmsg" : "Values in v:2 index key pattern cannot be of type bool. Only numbers > 0, numbers < 0, and strings are allowed.",
        "code" : 67,
```

```
        "codeName" : "CannotCreateIndex"
}


> db.rating.getIndexes();
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_",
                "ns" : "library.rating"
        },
        {
                "v" : 2,
                "key" : {
                        "movie_id" : 1,
                        "user_id" : 2,
                        "rating" : -1
                },
                "name" : "movie_id_1_user_id_2_rating_-1",
                "ns" : "library.rating"
        },
        {
                "v" : 2,
                "key" : {
                        "movie_id" : 1
                },
                "name" : "movie_id_1",
                "ns" : "library.rating"
        }
]



> db.rating.ensureIndex({movie_id:1});
{
        "numIndexesBefore" : 3,
        "numIndexesAfter" : 3,
        "note" : "all indexes already exist",
        "ok" : 1
}
```

c.sort movie_id in descending order

```
> db.rating.find({title:"Toy Story(1995)"},{rating:1}).sort({movie_id:-1});
{ "_id" : ObjectId("6538e2afd2224cb9b39f105a"), "rating" : 1 }
>
```

d.Create a descending order index on movie_id to get ratings related to “Toy Story (1995)” verify the query plan.

```
> db.rating.find({title:"Toy Story(1995)"},{rating:1}).sort({movie_id:-1});
{ "_id" : ObjectId("6538e2afd2224cb9b39f105a"), "rating" : 1 }
>

> db.rating.find().explain();
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "library.rating",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "winningPlan" : {
                        "stage" : "COLLSCAN",
                        "direction" : "forward"
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "comp-ThinkCentre-M920q",
                "port" : 27017,
                "version" : "3.6.8",
                "gitVersion" : "8e540c0b6db93ce994cc548f000900bdc740f80a"
        },
        "ok" : 1
}

> db.rating.find().explain();
{
        "queryPlanner" : {
                "plannerVersion" : 1,
                "namespace" : "library.rating",
                "indexFilterSet" : false,
                "parsedQuery" : {

                },
                "winningPlan" : {
                        "stage" : "COLLSCAN",
                        "direction" : "forward"
                },
                "rejectedPlans" : [ ]
        },
        "serverInfo" : {
                "host" : "comp-ThinkCentre-M920q",
                "port" : 27017,
                "version" : "3.6.8",
                "gitVersion" : "8e540c0b6db93ce994cc548f000900bdc740f80a"
        },
        "ok" : 1
}
```

e. Limit the number of items in the result of above query.

```
> db.rating.find({title:"Toy Story(1995)"},{rating:1}).sort({movie_id:-1}).limit(5);
{ "_id" : ObjectId("6538e2afd2224cb9b39f105a"), "rating" : 1 }
```

f. Get ratings for the movie “ICE AGE(2005)” using the descending ordered index on movie_id and explain.

```
> db.rating.find({title:"Horror(1920)"},{rating:1}).sort({movie_id:-1});
{ "_id" : ObjectId("6538e30bd2224cb9b39f105c"), "rating" : 2 }
{ "_id" : ObjectId("6538e2e4d2224cb9b39f105b"), "rating" : 2 }
>
```

g. Rebuild all indexes for the ratings collection

```
> db.rating.reIndex();
{
        "nIndexesWas" : 3,
        "nIndexes" : 3,
        "indexes" : [
                {
                        "v" : 2,
                        "key" : {
                                "_id" : 1
                        },
                        "name" : "_id_",
                        "ns" : "library.rating"
                },
                {
                        "v" : 2,
                        "key" : {
                                "movie_id" : 1,
                                "user_id" : 2,
                                "raing" : 2
                        },
                        "name" : "movie_id_1_user_id_2_raing_2",
                        "ns" : "library.rating"
                },
                {
                        "v" : 2,
                        "key" : {
                                "movie_id" : 1,
                                "reting" : -1
                        },
                        "name" : "movie_id_1_reting_-1",
                        "ns" : "library.rating"
                }
```

```
        ],
        "ok" : 1
}
```

h. Drop index on rating collection.

```
> db.rating.dropIndexes();
{
        "nIndexesWas" : 3,
        "msg" : "non-_id indexes dropped for collection",
        "ok" : 1
}
```

i.Create an index on movie_id and ratings fields together with movie_id (ascending order sorted) and rating (descending order sorted)

```
> db.rating.ensureIndex({movie_id:1,reting:-1});
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "ok" : 1
}
```

j. A compound index for movie_id, rating, and user_id.

```
> db.rating.ensureIndex({movie_id:1,user_id:2,rating:-1});
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
```