19.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'

}

1. Creating an index on movie\_id and sorts the keys in the index in ascending order.

Verify the query plan

1. Show various indexes created on movie collection.
2. Sort movie\_id in descending order.
3. Create a descending order index on movie\_id to get ratings related to “Toy Story (1995)” verify the query plan.
4. Limit the number of items in the result of above query.

// Use your MongoDB database and collection names use your\_database\_name;

// Create the "rating" collection db.createCollection("rating");

// Insert 5 documents into the "rating" collection db.rating.insert([

{

movie\_id: 123, user\_id: 12, title: "Toy Story (1995)", status: 'A'

}

]);

// Create an index on movie\_id

db.rating.createIndex({ movie\_id: 1 });

// Verify the query plan

db.rating.find().sort({ movie\_id: 1 }).explain("executionStats");

// Show indexes on the "rating" collection db.rating.getIndexes();

// Sort movie\_id in descending order db.rating.find().sort({ movie\_id: -1 });

// Create a descending order index on movie\_id db.rating.createIndex({ movie\_id: -1 });

// Verify the query plan for "Toy Story (1995)"

db.rating.find({ title: "Toy Story (1995)" }).sort({ movie\_id: -1 }).explain("executionStats");

// Limit the number of items in the result

db.rating.find({ title: "Toy Story (1995)" }).sort({ movie\_id: -1 }).limit(5);