21. 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. Get ratings for the movie “ICE AGE(2005)” using the descending ordered index on movie\_id and explain.
2. Rebuild all indexes for the ratings collection.
3. Drop index on rating collection.
4. Create an index on movie\_id and ratings fields together with movie\_id (ascending order sorted) and rating (descending order sorted).
5. Create a descending order index on movie\_id to get ratings related to “Toy Story (1995)” verify the query plan.

// 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 a descending order index on movie\_id

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

// Query ratings for the movie "ICE AGE (2005)" using the index and explain

db.rating.find({ title: "ICE AGE (2005)" }).sort({ movie\_id: -1 }).explain("executionStats");

// Rebuild all indexes for the "ratings" collection

db.rating.reIndex();

// Drop the index on movie\_id

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

// Create a compound index on movie\_id and ratings

db.rating.createIndex({ movie\_id: 1, ratings: -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");