use movie

db.createCollection("actor")

db.createCollection("film")

db.actor.insertOne({name:"SRk",age:"50",mail:"srk@gmail.com",Address:"Mumbai"})

db.actor.insertMany([{name:"Tom Holland",age:"27",mail:"TomH@gmail.com",Address:"Los Angeles"},{name:"Priyanka Chopra",age:"38",mail:"pcj@gmail.com",Address:"Mumbai-LA"},{name:"Prabhas",age:"40",mail:"prabhas@gmail.com",Address:"Hyderabad"},{name:"Siddharth",age:"35",mail:"siddharth@gmail.com",Address:"Mumbai"},{name:"Shraddha kapoor",age:"32",mail:"shraddha@gmail.com",Address:"Mumbai"},{name:"Aditya Roy Kapoor",age:"34",mail:"ark@gmail.com"},{name:"sushant singh",age:"35",mail:"ssr@gmail.com",Address:"Mumbai"},{name:"Katrina kaif",age:"38",mail:"kk@gmail.com",Address:"Mumbai"},{name:"zendya",age:"25",mail:"zendya@gmail.com",Address:"Los Angeles"}])

db.film.insertMany([{Fname:"Tarzen",Release:"2008",Actors:"Ajay Devgan",Rating:"4.5",genre:"action-romantic",Director:"ASP"},{Fname:"Kabir Singh",Release:"2019",Actors:"Shahid-Kiara",Rating:"5",genre:"romantic",Director:"Sandeep Reddy"},{Fname:"Golmal",Release:"2015",Actors:"Ajay Devgan\_Karina Kapoor",Rating:"4.1",genre:"comedy-drama",Director:"Rohit Shetty"},{Fname:"Ek tha Tiger",Release:"2011",Actors:"Salman Khan",Rating:"4.6",genre:"action-romantic",Director:"Rohit Shetty"},{Fname:"Maine Pyar kia",Release:"1995",Actors:"Salman Khan",Rating:"4.8",genre:"romantic",Director:"Sooraj"},{Fname:"Bhul Bhulaiyya 2",Release:"2022",Actors:"Kartik Aryan-Kiara Advani",Rating:"5",genre:"comedy-drama",Director:"Amit Dutta"},{Fname:"Dear Zindagi",Release:"2015",Actors:"Aliya Bhatt",Rating:"4.3",genre:"drama",Director:"steve smith"},{Fname:"Bahubali",Release:"2017",Actors:"Prabhas-Anushka shetty",Rating:"5",genre:"action-romantic",Director:"Rajmouli"},{Fname:"Dangal",Release:"2018",Actors:"Amir khan",Rating:"5",genre:"action",Director:"G Dutta"},{Fname:"Ramleela",Release:"2014",Actors:"Ranveer-Deepika",Rating:"4.6",genre:"action-romantic",Director:"Rajmouli"}])

db.film.find()

db.actor.find()

db.film.updateMany({Fname:/^T/},{$set:{Rating:"4.7"}})

db.actor.insertOne({name:"Shahid Kapoor",age:"39",mail:"shahidk@gmail.com",Address:"Mumbai"})

db.film.insertOne({Fname:"Kismat Konection",Release:"2018",Actors:"Shahid Kapoor",Rating:"3.8",genre:"action-romantic",Director:"A Dutta"})

db.film.find()

db.actor.find()

db.film.deleteOne({Fname:"Kismat Konection"})

db.actor.deleteOne({name:"Shahid Kapoor"})

db.actor.deleteMany({age:{$gt:50}})

db.actor.updateOne({name:"Shraddha kapoor"},{$set:{Address:"Mumbai-South"}})

db.film.updateMany({Director:"steve smith"},{$set:{genre:"about life"}})

db.film.find().sort({Fname:1,Release:-1})

db.actor.find().sort({age:1})

db.film.find({Actors:"Salman Khan"}).sort({Release:-1}).limit(2)

db.film.find({Actors:"Shahid-Kiara"},{Fname:1})

db.film.find({genre:"comedy-drama"}).sort({Release:-1})

**2**

// Step 1: Create a database with name 'Company'

use Company

// Step 2: Create collection "Employee"

db.createCollection("Employee")

// Step 3: Create collection "Transaction"

db.createCollection("Transaction")

// Step 4: Insert 5 documents into 'Employee' collection

db.Employee.insertMany([

{

\_id: 101,

firstName: "John",

lastName: "Doe",

designation: "Supervisor",

salary: 5000,

address: "123 Main St"

},

{

\_id: 102,

firstName: "Jane",

lastName: "Smith",

designation: "Clerk",

salary: 4000,

address: "456 Elm St"

},

{

\_id: 103,

firstName: "Alice",

lastName: "Johnson",

designation: "Manager",

salary: 7000,

address: "789 Oak St"

},

{

\_id: 104,

firstName: "Robert",

lastName: "Brown",

designation: "Technician",

salary: 5500,

address: "321 Pine St"

},

{

\_id: 105,

firstName: "Emily",

lastName: "Davis",

designation: "Assistant",

salary: 3500,

address: "654 Cedar St"

}

])

// Step 5: Insert 10 documents into 'Transaction' collection

db.Transaction.insertMany([

{

transactionId: 201,

amount: 100,

paymentType: "Cash",

remark: "Payment received"

},

{

transactionId: 202,

amount: 200,

paymentType: "Credit Card",

remark: "Pending payment"

},

{

transactionId: 203,

amount: 150,

paymentType: "Cash",

remark: "Payment complete"

},

{

transactionId: 204,

amount: 300,

paymentType: "PayPal",

remark: "Payment processing"

},

{

transactionId: 205,

amount: 250,

paymentType: "Credit Card",

remark: "Payment confirmed"

},

{

transactionId: 206,

amount: 180,

paymentType: "Cash",

remark: "Payment received"

},

{

transactionId: 207,

amount: 400,

paymentType: "Credit Card",

remark: "Payment pending"

},

{

transactionId: 208,

amount: 220,

paymentType: "Cash",

remark: "Payment in progress"

},

{

transactionId: 209,

amount: 350,

paymentType: "PayPal",

remark: "Payment completed"

},

{

transactionId: 210,

amount: 280,

paymentType: "Cash",

remark: "Payment received"

}

])

// Step 6: Display all the documents of both the collections in a formatted manner

print("Employee Collection:")

db.Employee.find().pretty()

print("Transaction Collection:")

db.Transaction.find().pretty()

// Step 7: Update salary of all employees by giving an increment of Rs. 4000

db.Employee.updateMany({}, { $inc: { salary: 4000 } })

// Step 8: Update the remark for transaction id 201

db.Transaction.updateOne({ transactionId: 201 }, { $set: { remark: "Updated remark" } })

// Step 9: Update designation of an employee named "John" from supervisor to manager

db.Employee.updateOne({ firstName: "John" }, { $set: { designation: "Manager" } })

// Step 10: Update designation of an employee having Employee Id as "101"

db.Employee.updateOne({ \_id: 101 }, { $set: { designation: "New Designation" } })

// Step 11: Change the address of an employee having Employee Id as "101"

db.Employee.updateOne({ \_id: 101 }, { $set: { address: "New Address" } })

// Step 12: Delete transaction made by "raj" employee on the given date

db.Transaction.deleteMany({ employeeName: "raj", date: "2023-05-29" })

// Step 13: Delete all the employees whose first name starts with 'K'

db.Employee.deleteMany({ firstName: /^K/ })

// Step 14: Find employees having designation as either 'manager' or 'floor supervisor'

db.Employee.find({ designation: { $in: ["manager", "floor supervisor"] } }).pretty()

// Step 15: Find an employee whose name ends with " " and display it

db.Employee.find({ lastName: / $/ }).pretty()

// Step 16: Display the name of an employee whose salary is greater than "5000" using a MongoDB cursor

var cursor = db.Employee.find({ salary: { $gt: 5000 } })

while (cursor.hasNext()) {

var employee = cursor.next()

print(employee.firstName + " " + employee.lastName)

}

// Step 17: Sort the employees in the descending order of their designation

db.Employee.find().sort({ designation: -1 }).pretty()

// Step 18: Count the total number of employees in a collection

db.Employee.count()

// Step 19: Calculate the sum of total amount paid for all the transaction documents

db.Transaction.aggregate([{ $group: { \_id: null, totalAmount: { $sum: "$amount" } } }])

// Step 20: Calculate the sum of total amount paid for each payment type

db.Transaction.aggregate([{ $group: { \_id: "$paymentType", totalAmount: { $sum: "$amount" } } }])