Q352

Write commands to perform the following tasks on an employee collection having fields

name, age & joiningDate:

(1) Insert 3-4 records in the collection.

(2) List all employees who joined before 1st January 2010.

(3) Update the name of the employee to "WWW" whose joiningDate is "05-05-2015"

(1) Insert 3-4 records in the collection

db.employee.insertMany([

    { name: "Dwij", age: 19, joiningDate: "2004-02-31" },

    { name: "Shubham", age: 20, joiningDate: "2007-05-05" },

    { name: "Jyoti", age: 21, joiningDate: "2009-05-05" },

    { name: "Brijesh", age: 22, joiningDate: "2015-05-05" }

])

(2) List all employees who joined before 1st January 2010

db.employee.find({ "joiningDate": { $lt: "2010-01-01" } })

(3) Update the name of the employee to "WWW" whose joiningDate is "05-05-2015"

db.employee.updateOne({ 'joiningDate': "2015-05-05" }, { $set: { name: "WWW" } })

Q353

Write commands to perform following tasks on employee collection having fields having

name,age & joiningDate:

(1) Delete all records having joiningDate before 1st January, 2010.

(2) List all employees having age>50 years.

(3) List only 1st employee having age>60 years.

db.employee.insertMany([

    { name: "Omdev", age: 19, joiningDate: "2004-02-31" },

    { name: "Jainam", age: 20, joiningDate: "2007-05-05" },

    { name: "Jatin", age: 21, joiningDate: "2009-05-05" },

    { name: "Amisha", age: 50, joiningDate: "2015-05-05" },

    { name: "Mosam", age: 60, joiningDate: "2017-09-05" },

    { name: "Rohan", age: 70, joiningDate: "2019-05-05" },

    { name: "Divyam", age: 55, joiningDate: "2019-10-31" }

])

(1) Delete all records having joiningDate before 1st January, 2010.

db.employee.deleteMany({ "joiningDate": { $lt: "2010-01-01" } })

(2) List all employees having age>50 years.

db.employee.find({ "age": { $gt: 50 } })

(3) List only 1st employee having age>60 years.

db.employee.find({ "age": { $gt: 60 } }).limit(1)

Q354

Write commands to perform following tasks on employee collection having fields having

name,age & joiningDate:

(1) Update the name="Senior citizen" having age>60 years.

(2) Update the name="JKL" having age=20 years. Insert this record, if it is not found.

(3) Retire all employees by deleting senior citizens from collection.

(1) Update the name="Senior citizen" having age>60 years.

db.employee.updateMany({ "age": { $gt: 60 } }, { $set: { name: "Senior citizen" } })

(2) Update the name="JKL" having age=20 years. Insert this record, if it is not found.

db.employee.updateOne({ "age": 20 }, { $set: { name: "JKL" } }, { upsert: true })

(3) Retire all employees by deleting senior citizens from collection.

db.employee.deleteMany({ "age": { $gt: 60 } })

Q355

Write commands to perform following tasks on employee collection having fields having

name,age & joiningDate:

(1) Count no. of employees having age>=60 years.

(2) List all employees in descending order of names having names "ABC", "PQR", "XYZ".

(3) List all employees whose age lies between 25 to 50 years excluding all rest of the fields

(1) Count no. of employees having age>=60 years.

db.employee.find({ "age": { $gte: 60 } }).count()

(2) List all employees in descending order of names having names "ABC", "PQR", "XYZ".

db.employee.insertMany([

    { name: "ABC", age: 19, joiningDate: "2004-02-31" },

    { name: "PQR", age: 20, joiningDate: "2007-05-05" },

    { name: "XYZ", age: 21, joiningDate: "2009-05-05" },

])

db.employee.find({ name: { $in: ["ABC", "PQR", "XYZ"] } }).sort({ name: -1 })

(3) List all employees whose age lies between 25 to 50 years excluding all rest of the fields

db.employee.find({ "age": { $gte: 25, $lte: 50 } }, { name: 1, age: 1, joiningDate: 1, \_id: 0 })