M3 - Crud Operations

Problem Statement: Design and Develop MongoDB Queries using CRUD operations: Create Employee collection by considering following Fields: i. Emp_id : Number ii. Name: Embedded Doc (FName, LName) iii. Company Name: String iv. Salary: Number v. Designation: String vi. Age: Number vii. Expertise: Array viii. DOB: String or Date ix. Email id: String x. Contact: String xi. Address: Array of Embedded Doc (PAddr, LAddr) Insert at least 5 documents in collection by considering above attribute and execute following queries: 1. Creates a new document if no document in the employee collection contains {Designation: "Tester", Company_name: "TCS", Age: 25} 2. Finds all employees working with Company_name: "TCS" and increase their salary by 2000. 3. Matches all documents where the value of the field Address is an embedded document that contains only the field city with the value "Pune" and the field Pin_code with the value "411001". 4. Find employee details who are working as "Developer" or "Tester". 5. Drop Single documents where designation="Developer". 6. Count number of documents in employee collection.

Creating database & collection:

```
use empDB3
db.createCollection("Employee")
```

Inserting data:

```
db.Employee.insertMany([
{
  Name: {FName: "Ayush", LName: "Kalaskar"},
  Company: "TCS",
  Salary: 45000,
  Designation: "Programmer",
  Age: 24,
  Expertise: ['Docker', 'Linux', 'Networking', 'Politics'],
  DOB: new Date("1998-03-12"),
  Email: "ayush.k@tcs.com",
  Contact: 9972410427,
  Address: [{PAddr: "Kokan, Maharashtra"}, {LAddr: "Lohegaon, Pune", Pin_code: 411014}]
},
{
  Name: {FName: "Mehul", LName: "Patil"},
  Company: "MEPA",
  Salary: 55000,
  Designation: "Tester",
  Age: 20,
  Expertise: ['HTML', 'CSS', 'Javascript', 'Teaching'],
```

```
  DOB: new Date("1964-06-22"),
  Email: "mehul.p@mepa.com",
  Contact: 9972410426,
  Address: [{PAddr: "NDB, Maharashtra"}, {LAddr: "Camp, Pune", Pin_code: 411001}]
},
{
  Name: {FName: "Himanshu", LName: "Patil"},
  Company: "Infosys",
  Salary: 85000,
  Designation: "Developer",
  Age: 67,
  Expertise: ['Mongodb', 'Mysql', 'Cassandra', 'Farming'],
  DOB: new Date("1957-04-28"),
  Email: "himanshu.p@infosys.com",
  Contact: 9972410425,
  Address: [{PAddr: "NDB, Maharashtra"}, {LAddr: "Camp, Pune", Pin_code: 411001}]
}
])
```

Queries

1. Creates a new document if no document in the employee collection contains `{Designation: "Tester", Company_name: "TCS", Age: 25}`

```
db.Employee.updateOne(
  {Designation: "Tester", Company: "TCS", Age: 25},
  { $setOnInsert: {
      Name: {FName: "Karan", LName: "Salvi"},
      Salary: 67500,
      Expertise: ['Blockchain', 'C++', 'Python', 'Fishing'],
      DOB: new Date("1999-11-01"),
      Email: "karan.s@tcs.com",
      Contact: 9972410424,
      Address: [{PAddr: "Kolhapur, Maharashtra"}, {LAddr: "Viman Nagar, Pune", Pin_code: 411
    }
  },
  { upsert: true }
)
```

2. Finds all employees working with Company_name: “TCS” and increase their salary by 2000.

```
db.Employee.updateMany(
  { Company: "TCS" },
  { $inc: { Salary: 2000 } }
)
```

3. Matches all documents where the value of the field Address is an embedded

document that contains only the field city with the value “Pune” and the field Pin_code with the value “411001”.

```
db.Employee.find(
  { $or: [
      {
        "Address.Pin_code": 411001,
        "Address.LAddr": { $regex: /Pune/i }
      },
      {
        "Address.Pin_code": 411001,
        "Address.PAddr": { $regex: /Pune/i }
      }
    ]
  }
)
```

4. Find employee details who are working as “Developer” or “Tester”.

```
db.Employee.find(
  { $or: [
      { Designation: "Developer" },
      { Designation: "Tester" }
    ]
  }
)
```

5. Drop Single documents where Designation=“Developer”

```
db.Employee.deleteOne( { Designation: "Developer" } )
```

6. Count number of documents in employee collection.

```
db.Employee.countDocuments();
```
