M1 - Crud Operations

Problem Statement: Design and Develop MongoDB Queries using CRUD operations: Create Employee collection by considering following Fields: i. Name: Embedded Doc (FName, LName) ii. Company Name: String iii. Salary: Number iv. Designation: String v. Age: Number vi. Expertise: Array vii. DOB: String or Date viii. Email id: String ix. Contact: String x. Address: Array of Embedded Doc (PAddr, LAddr)

Insert at least 5 documents in collection by considering above attribute and execute following queries: 1. Select all documents where the Designation field has the value "Programmer" and the value of the salary field is greater than 30000. 2. Creates a new document if no document in the employee collection contains {Designation: "Tester", Company_name: "TCS", Age: 25} 3. Increase salary of each Employee working with "Infosys" 10000. 4. Finds all employees working with "TCS" and reduce their salary by 5000. 5. Return documents where Designation is not equal to "Tester". 6. Find all employee with Exact Match on an Array having Expertise: ['Mongodb','Mysql','Cassandra']

Creating database & collection:

```
use empDB1
db.createCollection("Employee")
```

Inserting data:

```
db.Employee.insertMany([
{
  Name: {FName: "Ayush", LName: "Kalaskar"},
  Company: "TCS",
  Salary: 45000,
  Designation: "Programmer",
  Age: 55,
  Expertise: ['Docker', 'Linux', 'Networking', 'Politics'],
  DOB: new Date("1969-03-12"),
  Email: "ayush.k@tcs.com",
  Contact: 9972410427,
  Address: [{PAddr: "Kokan, Maharashtra"}, {LAddr: "Lohegaon, Pune"}]
},
{
  Name: {FName: "Mehul", LName: "Patil"},
  Company: "MEPA",
  Salary: 55000,
  Designation: "Tester",
  Age: 60,
```

```
  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"}]
},
{
  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"}]
}
])
```

Queries

1. Select all documents where the Designation field has the value "Programmer" and the value of the salary field is greater than 30000.

```
db.Employee.find(
{ Designation: "Programmer", Salary: { $gt: 30000 } }
)
```

2. 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"}]
  }
},
{ upsert: true }
)
```

3. Increase salary of each Employee working with “Infosys” 10000.

```
db.Employee.updateMany(
{ Company: "Infosys" },
{ $inc: { Salary: 10000 } }
)
```

4. Finds all employees working with “TCS” and reduce their salary by 5000.

```
db.Employee.updateMany(
{ Company: "TCS" },
{ $inc: { Salary: -5000 } }
)
```

5. Return documents where Designation is not equal to “Tester”.

```
db.Employee.find(
{ Designation: { $ne: "Tester"} }
)
```

6. Find all employee with Exact Match on an Array having Expertise: `['Mongodb','Mysql','Cassandra']`

```
db.Employee.find(
{ Expertise: { $all: ['Mongodb', 'Mysql', 'Cassandra'] } }
)
```
