

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']

```
use Empdb
```

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

```
# Insert 10 documents in Employee collection
```

```
db.Employee.insertMany([
  {
    "Empid": 1,
    "Name": {
      "FName": "John",
      "LName": "Doe"
    },
    "CompanyName": "Infosys",
    "Salary": 50000,
    "Designation": "Developer",
    "Age": 30,
    "Expertise": ["Java", "Python", "MongoDB"],
    "DOB": "1992-08-05",
    "Email": "john.doe@infosys.com",
    "Contact": "9876543210",
    "Address": [
      {
        "PAddr": "123 Main Street",
        "LAddr": "Pune, Maharashtra"
      }
    ]
  },
  {
    "Empid": 2,
```

```

    "Name": {
      "FName": "Jane",
      "LName": "Doe"
    },
    "CompanyName": "TCS",
    "Salary": 40000,
    "Designation": "Tester",
    "Age": 25,
    "Expertise": ["Selenium", "Appium", "JIRA"],
    "DOB": "1997-06-15",
    "Email": "jane.doe@tcs.com",
    "Contact": "8765432109",
    "Address": [
      {
        "PAddr": "456 Elm Street",
        "LAddr": "Mumbai, Maharashtra"
      }
    ]
  },
  {
    "Empid": 3,
    "Name": {
      "FName": "Peter",
      "LName": "Parker"
    },
    "CompanyName": "Wipro",
    "Salary": 60000,
    "Designation": "Manager",
    "Age": 35
  }
])

```

// Query 1: Select all documents where the Designation field is "Programmer" and the salary is greater than 30000

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

// Query 2: Create a new document if no document in the "Employee" collection contains

```
//{Designation: "Tester", Company_name: "TCS", Age: 25}
```

```

db.Employee.updateOne(
  { "Designation": "Tester", "Company_Name": "TCS", "Age": 25 },
  { $setOnInsert: { "Designation": "Tester", "Company_Name": "TCS", "Age": 25 } },
  { upsert: true }
)

```

// Query 3: Increase salary of each Employee working with "Infosys" by 10000

```

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

```

// Query 4: Find all employees working with "TCS" and reduce their salary by

```

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

// Query 5: Return documents where Designation is not equal to "Tester"
db.Employee.find({ "Designation": { $ne: "Tester" } })

// Query 6: Find all employees with an Exact Match on an Array having Expertise:
['Mongodb','Mysql','Cassandra']
db.Employee.find({ "Expertise": { $all: ['Mongodb', 'Mysql', 'Cassandra'] } })

db.Employee.find({ Designation: "Programmer", Salary: { $gt: 30000 } })
db.Employee.insertOne({
  Designation: "Tester",
  CompanyName: "TCS",
  Age: 25
}, { upsert: true })

db.Employee.find({ $or: [ { Age: { $lt: 30 } }, { Salary: { $gt: 40000 } } ] })
db.Employee.find({ Address: { $elemMatch: { city: "Pune", Pin_code: "411001" } } })
db.Employee.updateMany({ CompanyName: "TCS" }, { $inc: { Salary: 2000 } })
db.Employee.find({ Designation: { $ne: "Developer" } })
db.Employee.find({ CompanyName: "Infosys" }, { _id: 1, Designation: 1, Address:
1, Name: 1 })
db.Employee.find({ Designation: { $in: ["Developer", "Tester"] } })
db.Employee.find({ Expertise: { $all: ["Mongodb", "Mysql", "Cassandra"] } })
db.Employee.deleteOne({ Designation: "Developer" })

```