

M2 - 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. Find name of Employee where age is less than 30 and salary more than 50000. 2. Create a new document if no document in the employee collection contains {Designation: "Tester", Company_name: "TCS", Age: 25} 3. Selects all documents in the collection where the field age has a value less than 30 or the value of the salary field is greater than 40000. 4. Find documents where Designation is not equal to "Developer". 5. Find _id, Designation, Address and Name from all documents where Company_name is "Infosys". 6. Display only FName and LName of all Employees

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: 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"}]
},
{
  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"}]
  },
  {
    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. Final name of Employee where age is less than 30 and salary more than 50000.

```

db.Employee.find(
{
  Age: { $lt: 30 },
  Salary: { $gt: 50000 }
}
)

```

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: 35000,
  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. Selects all documents in the collection where the field age has a value less than 30 or the value of the salary field is greater than 40000.

```
db.Employee.find(  
  { $or:  
    [  
      { Age: { $lt: 30 } },  
      { Salary: { $gt: 40000 } }  
    ]  
  }  
)
```

4. Find documents where Designation is not equal to “Developer”.

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

5. Find __id, Designation, Address and Name from all documents where Company__name is “Infosys”.

```
db.Employee.find(  
  { Company: "Infosys" },  
  { _id: 1, Designation: 1, Address: 1, Name: 1 }  
)
```

6. Display only FName and LName of all Employees.

```
db.Employee.find(  
  {},  
  {"Name.FName": 1, "Name.LName": 1}  
)
```
