MongoDB Coding Assessment

One-to-One Relationships with Embedded Documents

```
> db.student.insertOne({_id: 1,name: "Tejashree",passport:{passport_no: "M1234567",country:"India",expiry: "2030-12-31"}})
< {
    acknowledged: true,
    insertedId: 1
  }
> db.student.find()
< {
    _id: 1,
    name: 'Tejashree',
    passport: {
      passport_no: 'M1234567',
      country: 'India',
      expiry: '2030-12-31'
    }
  }
```

Inserted One more value

```
> db.student.insertOne({_id: 2,name: "Shiva",passport:{passport_no: "M987654",country:"India",expiry: "2030-12-12"}})
< {
    acknowledged: true,
    insertedId: 2
  }
> db.student.find()
< {
    _id: 1,
    name: 'Tejashree',
    passport: {
      passport_no: 'M1234567',
      country: 'India',
      expiry: '2030-12-31'
    }
  }
  {
    _id: 2,
    name: 'Shiva',
    passport: {
      passport_no: 'M987654',
      country: 'India',
      expiry: '2030-12-12'
    }
  }
```

Displaying the passport number of the student

```
> db.student.find({name:"Shiva"},{"passport.passport_no":1}).pretty()
< {
    _id: 2,
    passport: {
      passport_no: 'M987654'
    }
  }
```

One-to-Many Relationships with Embedded Documents

Code

db.student.insertOne({_id:3,name:"Alica",
passports:[{old_passport_no:"G1234",country: "USA",expiry: "2030-08-21"},
{new_passport_no: "N9876543",country: "USA",expiry: "2020-05-01"}]})

```
> db.student.insertOne({_id:3,name:"Alica", passports:[{old_passport_no:"G1234",country: "USA",expiry: "2030-08-21"}, {new_passport_no: "N9876543",country: "USA",expiry: "2020-05-01"}]})
< {
    acknowledged: true,
    insertedId: 3
  }
student >
```

Output

```
< {
    acknowledged: true,
    insertedId: 3
  }
```

```
> db.student.find().pretty()
< {
    _id: 1,
    name: 'Tejashree',
    passport: {
      passport_no: 'M1234567',
      country: 'India',
      expiry: '2030-12-31'
    }
  }
  {
    _id: 2,
    name: 'Shiva',
    passport: {
      passport_no: 'M987654',
      country: 'India',
      expiry: '2030-12-12'
    }
  }
  {
    _id: 3,
    name: 'Alica',
    passports: [
      {
        old_passport_no: 'G1234',
        country: 'USA',
        expiry: '2030-08-21'
      },
      {
        new_passport_no: 'N9876543',
        country: 'USA',
        expiry: '2020-05-01'
      }
    ]
  }
```

Displaying the old and new passport number of the student

```
> db.student.find({name:"Alica"},{"passports.old_passport_no":1,"passports.new_passport_no":1}).pretty()
< {
    _id: 3,
    passports: [
      {
        old_passport_no: 'G1234'
      },
      {
        new_passport_no: 'N9876543'
      }
    ]
  }
```

Many-to-Many Relationships with Embedded Documents

A student can enroll in multiple courses, and a course can have multiple students.

```
> db.courses.insertMany([
    { _id: "C101", name: "Mathematics" },
    { _id: "C102", name: "Physics" },
    { _id: "C103", name: "Chemistry" }
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': 'C101',
      '1': 'C102',
      '2': 'C103'
    }
  }
```

```
> db.students_enrolled_data.insertMany([
    { _id: "S001", name: "Tejashree", enrolled_courses: ["C101", "C102"] },
    { _id: "S002", name: "Shiva", enrolled_courses: ["C102", "C103"] },
    { _id: "S003", name: "Alica", enrolled_courses: ["C101", "C103"] }
]);
< {
    acknowledged: true,
    insertedIds: {
      '0': 'S001',
      '1': 'S002',
      '2': 'S003'
    }
  }
```

Output

```
> db.students_enrolled_data.find().pretty()
< {
    _id: 'S001',
    name: 'Tejashree',
    enrolled_courses: [
      'C101',
      'C102'
    ]
  }
  {
    _id: 'S002',
    name: 'Shiva',
    enrolled_courses: [
      'C102',
      'C103'
    ]
  }
  {
    _id: 'S003',
    name: 'Alica',
    enrolled_courses: [
      'C101',
      'C103'
    ]
  }
```

```
> db.courses.find().pretty()
< {
    _id: 'C101',
    name: 'Mathematics'
  }
  {
    _id: 'C102',
    name: 'Physics'
  }
  {
    _id: 'C103',
    name: 'Chemistry'
  }
```

Many-to-One Relationships with Document References

Many contacts (multiple phone numbers, emails, etc.) belong to one person.

```
> db.persons.insertMany([
    { _id: "P001", name: "Mr. Kumar", company: "Tech Solutions" },
    { _id: "P002", name: "Ms. Priya", company: "Hexaware" }
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': 'P001',
      '1': 'P002'
    }
  }
```

```
> db.contacts.insertMany([
    { _id: 1, phone: "+91-9876543210", email: "kumar.work@gmail.com", person_id: "P001" },
    { _id: 2, phone: "+91-9123456789", email: "kumar.personal@gmail.com", person_id: "P001" },
    { _id: 3, phone: "+91-9988776655", email: "priya@gmail.com", person_id: "P002" }
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': 1,
      '1': 2,
      '2': 3
    }
  }
> db.persons.find().pretty()
```

```
> db.persons.find().pretty()
< {
    _id: 'P001',
    name: 'Mr. Kumar',
    company: 'Tech Solutions'
  }
  {
    _id: 'P002',
    name: 'Ms. Priya',
    company: 'Hexaware'
  }
```

```
> db.contacts.find().pretty()
< {
    _id: 1,
    phone: '+91-9876543210',
    email: 'kumar.work@gmail.com',
    person_id: 'P001'
  }
  {
    _id: 2,
    phone: '+91-9123456789',
    email: 'kumar.personal@gmail.com',
    person_id: 'P001'
  }
  {
    _id: 3,
    phone: '+91-9988776655',
    email: 'priya@gmail.com',
    person_id: 'P002'
  }
```