CODING CHALLENGE – UNDERSTANDING RELATIONSHIPS

Using Database: codingChallenge

```
> use codingChallenge
< switched to db codingChallenge
```

1. **One-to-One Relationships with Embedded Documents:**

   Collection: student

   Documents: One document that contains the details of a student

```
> db.student.insertOne({
    StudentName: "GeeksA",
    StudentId: "g_f_g_1209",
    Branch: "CSE",
    PermanentAddress: {
      permaAddress: "XXXXX",
      City: "Delhi",
      PinCode: 202333
    }
  })
< {
    acknowledged: true,
    insertedId: ObjectId('6883051eda2b0205ce0d0782')
  }
```

   Output:

```
> db.student.find().pretty()
< {
    _id: ObjectId('6883051eda2b0205ce0d0782'),
    StudentName: 'GeeksA',
    StudentId: 'g_f_g_1209',
    Branch: 'CSE',
    PermanentAddress: {
      permaAddress: 'XXXXX',
      City: 'Delhi',
      PinCode: 202333
    }
  }
codingChallenge>
```

Displaying the address of the student:

```
> db.student.find({StudentName:"GeeksA"},{"PermanentAddress.permaAddress":1}).pretty()
< {
    _id: ObjectId('6883051eda2b0205ce0d0782'),
    PermanentAddress: {
      permaAddress: 'XXXXX'
    }
  }
codingChallenge>
```

2. **One-to-Many Relationships with Embedded Documents:**

Collection: student

Documents: One document that contains the details of a student

```
> db.student.insertOne({
      StudentName: "GeeksA",
      StudentId: "g_f_g_1209",
      Branch: "CSE",
      Address: [
          {
              PermanentAddress: "XXXXXXX",
              City: "Delhi",
              PinCode: 202333
          },
          {
              CurrentAddress: "PPPPPPP",
              City: "Mumbai",
              PinCode: 334509
          }
      ]
  })
< {
    acknowledged: true,
    insertedId: ObjectId('688309dcda2b0205ce0d0786')
  }
```

Output:

```
> db.student.find().pretty()
< {
    _id: ObjectId('688309dcda2b0205ce0d0786'),
    StudentName: 'GeeksA',
    StudentId: 'g_f_g_1209',
    Branch: 'CSE',
    Address: [
      {
        PermanentAddress: 'XXXXXXX',
        City: 'Delhi',
        PinCode: 202333
      },
      {
        CurrentAddress: 'PPPPPPP',
        City: 'Mumbai',
        PinCode: 334509
      }
    ]
  }
codingChallenge>
```

Displaying all the addresses of the student

```
> db.student.find({StudentName:"GeeksA"},
                  {"Address.PermanentAddress":1,
                   "Address.CurrentAddress":1}).pretty()
< {
    _id: ObjectId('688309dcda2b0205ce0d0786'),
    Address: [
      {
        PermanentAddress: 'XXXXXXX'
      },
      {
        CurrentAddress: 'PPPPPPP'
      }
    ]
  }
codingChallenge>
```

3. **One-to-Many relationships with the document reference:**

Inserting teacher document:

Collection: teacher

Documents: One document that contains the details of a teacher

```
> db.teacher.insertOne({
    teacherName: "Sunita",
    TeacherId: "g_f_g_1209",
    classIds: ["C_123", "C_234"]
  })
< {
    acknowledged: true,
    insertedId: ObjectId('68830c00da2b0205ce0d078f')
  }
codingChallenge>
```

Inserting class document:

Collection: class

Documents: One document that contains the details of a teacher assigned with class

```
> db.class.insertMany([
    {
      TeacherId: "g_f_g_1209",
      ClassName: "GeeksA",
      ClassId: "C_123",
      Studentcount: 23,
      Subject: "Science"
    },
    {
      TeacherId: "g_f_g_1209",
      ClassName: "GeeksB",
      ClassId: "C_234",
      Studentcount: 33,
      Subject: "Maths"
    }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('68830ba9da2b0205ce0d078d'),
      '1': ObjectId('68830ba9da2b0205ce0d078e')
    }
  }
codingChallenge>
```

Output:

```
> db.teacher.find().pretty()
< {
    _id: ObjectId('68830ca2da2b0205ce0d0790'),
    className: 'GeeksB',
    studentCount: 33,
    subject: 'Maths'
  }
  {
    _id: ObjectId('68830ca2da2b0205ce0d0791'),
    className: 'GeeksA',
    studentCount: 23,
    subject: 'Science'
  }
  {
    _id: ObjectId('68830ca2da2b0205ce0d0792'),
    name: 'Sunita',
    TeacherId: 'g_f_g_1209',
    classId: [
      ObjectId('60263b49f19652db63812e9a'),
      ObjectId('60263b49f19652db63812e9b')
    ]
  }
codingChallenge>
```

Displaying the values of classId field

```
> db.teacher.findOne({name:"Sunita"}, {classId:1})
< {
    _id: ObjectId('68830ca2da2b0205ce0d0792'),
    classId: [
      ObjectId('60263b49f19652db63812e9a'),
      ObjectId('60263b49f19652db63812e9b')
    ]
  }
codingChallenge>
```

4. Many-to-Many Relationships with Embedded Documents:

Embedded Teachers inside Classes:

```
> db.classes.insertMany([
    {
      className: "GeeksA",
      subject: "Science",
      studentCount: 23,
      teachers: [
        {
          name: "Sunita",
          teacherId: "T_001"
        }
      ]
    },
    {
      className: "GeeksB",
      subject: "Maths",
      studentCount: 33,
      teachers: [
        {
          name: "Sunita",
          teacherId: "T_001"
        },
        {
          name: "Rajesh",
          teacherId: "T_002"
        }
      ]
    }
  ])
```

```
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('68831369da2b0205ce0d0793'),
      '1': ObjectId('68831369da2b0205ce0d0794')
    }
  }
codingChallenge>
```

Output:

```
> db.classes.find()
< {
    _id: ObjectId('68831369da2b0205ce0d0793'),
    className: 'GeeksA',
    subject: 'Science',
    studentCount: 23,
    teachers: [
      {
        name: 'Sunita',
        teacherId: 'T_001'
      }
    ]
  }
  {
    _id: ObjectId('68831369da2b0205ce0d0794'),
    className: 'GeeksB',
    subject: 'Maths',
    studentCount: 33,
    teachers: [
      {
        name: 'Sunita',
        teacherId: 'T_001'
      },
      {
        name: 'Rajesh',
        teacherId: 'T_002'
      }
    ]
```