CODING CHALLENGE - 2 (MongoDB)

TRAINEE NAME: Swathi Baskaran

1. One-to-One Relationship

A person has one job profile

```
admin> use CodingChallengeMongoDB
switched to db CodingChallengeMongoDB
CodingChallengeMongoDB> db.createCollection("employeeProfile")
{ ok: 1 }
CodingChallengeMongoDB> db.employeeProfile.insertOne({name: "Swathi", age: 22, course: "B.Tech Information Technology",
 jobProfile: {company: "Hexaware Technologies", location: "Siruseri", salaryPackage: 4}})
{
  acknowledged: true,
  insertedId: ObjectId('68830524666aae8435cb0ce2')
}
CodingChallengeMongoDB> db.employeeProfile.find()
[
  {
    _id: ObjectId('68830524666aae8435cb0ce2'),
    name: 'Swathi',
    age: 22,
    course: 'B.Tech Information Technology',
    jobProfile: {
      company: 'Hexaware Technologies',
      location: 'Siruseri',
      salaryPackage: 4
    }
  }
]
```

2. One-to-Many Relationship

A single project title is linked to several members

```
CodingChallengeMongoDB> db.createCollection("projectTitle")
{ ok: 1 }
CodingChallengeMongoDB> db.projectTitle.insertOne({_id: 1, title: "CARS - Crime Analysis and Reporting System", numberO
fMembers: 3, batch: "Python"})
{ acknowledged: true, insertedId: 1 }
CodingChallengeMongoDB> db.projectTitle.find()
[
  {
    _id: 1,
    title: 'CARS - Crime Analysis and Reporting System',
    numberOfMembers: 3,
    batch: 'Python'
  }
]
```

```
CodingChallengeMongoDB> db.createCollection("projectMembers")
{ ok: 1 }
CodingChallengeMongoDB> db.projectMembers.insertMany([{project_id: 1, memberName: "Swathi Baskaran", age: 22}, {project
_id: 1, memberName: "Poojashree", age: 22}, {project_id: 1, memberName: "Shakthishree", age: 22}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68830661666aae8435cb0ce3'),
    '1': ObjectId('68830661666aae8435cb0ce4'),
    '2': ObjectId('68830661666aae8435cb0ce5')
  }
}
CodingChallengeMongoDB> db.projectMembers.find()
[
  {
    _id: ObjectId('68830661666aae8435cb0ce3'),
    project_id: 1,
    memberName: 'Swathi Baskaran',
    age: 22
  },
  {
    _id: ObjectId('68830661666aae8435cb0ce4'),
    project_id: 1,
    memberName: 'Poojashree',
    age: 22
  },
  {
    _id: ObjectId('68830661666aae8435cb0ce5'),
    project_id: 1,
    memberName: 'Shakthishree',
    age: 22
  }
]
```

3. Many-to-One Relationship

Many employees assigned to a single manager

```
CodingChallengeMongoDB> db.createCollection("managers")
{ ok: 1 }
CodingChallengeMongoDB> db.managers.insertOne([{_id: 101, name: "Sharmila", age: 45, domain: "Cloud Technologies"}, {_i
d: 102, name: "Preethi", age: 34, domain: "Data Analysis"}])
{
  acknowledged: true,
  insertedId: ObjectId('6883074d666aae8435cb0ce6')
}
```

```
CodingChallengeMongoDB> db.createCollection("employees")
{ ok: 1 }
CodingChallengeMongoDB> db.employees.insertMany([{employee_id: 1, name: "Swathi", age: 22, role: "Data Engineer", repor
ting_Manager_id: 102}, {employee_id: 2, name: "Anjali", age: 25, role: "Data Analyst", reporting_Manager_id: 102}, {emp
loyee_id: 3, name: "Geetha", age: 23, role: "Data Scientist", reporting_Manager_id: 102}, {employee_id: 4, name: "Rithi
ka", age: 22, role: "Cloud solutions architect", reporting_Manager_id: 101}, {employee_id: 5, name: "Caren", age: 28, r
ole: "Cloud Engineer", reporting_Manager_id: 101}, {employee_id: 6, name: "Sahana", age: 32, role: "Cloud Consultant",
reporting_Manager_id: 101}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('688308dd666aae8435cb0ce7'),
    '1': ObjectId('688308dd666aae8435cb0ce8'),
    '2': ObjectId('688308dd666aae8435cb0ce9'),
    '3': ObjectId('688308dd666aae8435cb0cea'),
    '4': ObjectId('688308dd666aae8435cb0ceb'),
    '5': ObjectId('688308dd666aae8435cb0cec')
  }
}
CodingChallengeMongoDB> db.employees.find()
[
  {
    _id: ObjectId('688308dd666aae8435cb0ce7'),
    employee_id: 1,
    name: 'Swathi',
    age: 22,
    role: 'Data Engineer',
    reporting_Manager_id: 102
  },
  {
    _id: ObjectId('688308dd666aae8435cb0ce8'),
    employee_id: 2,
    name: 'Anjali',
```

```
  {
    _id: ObjectId('688308dd666aae8435cb0ce9'),
    employee_id: 3,
    name: 'Geetha',
    age: 23,
    role: 'Data Scientist',
    reporting_Manager_id: 102
  },
  {
    _id: ObjectId('688308dd666aae8435cb0cea'),
    employee_id: 4,
    name: 'Rithika',
    age: 22,
    role: 'Cloud solutions architect',
    reporting_Manager_id: 101
  },
  {
    _id: ObjectId('688308dd666aae8435cb0ceb'),
    employee_id: 5,
    name: 'Caren',
    age: 28,
    role: 'Cloud Engineer',
    reporting_Manager_id: 101
  },
  {
    _id: ObjectId('688308dd666aae8435cb0cec'),
    employee_id: 6,
    name: 'Sahana',
    age: 32,
    role: 'Cloud Consultant',
    reporting_Manager_id: 101
  }
]
```

4. Many-to-Many Relationship

Students taking up many courses, and courses being chosen by multiple students

```
CodingChallengeMongoDB> db.students.insertMany([{_id: 1, name: "Swathi", course_ids: [101, 103, 104]}, {_id: 2, name: "
Saai Krishna", course_ids: [101, 102, 104]}])
{ acknowledged: true, insertedIds: { '0': 1, '1': 2 } }
CodingChallengeMongoDB> db.students.find()
[
  { _id: 1, name: 'Swathi', course_ids: [ 101, 103, 104 ] },
  { _id: 2, name: 'Saai Krishna', course_ids: [ 101, 102, 104 ] }
]
CodingChallengeMongoDB> db.createCollection("courses")
{ ok: 1 }
CodingChallengeMongoDB> db.courses.insertMany([{_id: 101, courseName: "Mathematics"}, {_id: 102, courseName: "Physics"}
, {_id: 103, courseName: "Biology"}, {_id: 104, courseName: "Chemistry"}])
{
  acknowledged: true,
  insertedIds: { '0': 101, '1': 102, '2': 103, '3': 104 }
}
CodingChallengeMongoDB> db.courses.find()
[
  { _id: 101, courseName: 'Mathematics' },
  { _id: 102, courseName: 'Physics' },
  { _id: 103, courseName: 'Biology' },
  { _id: 104, courseName: 'Chemistry' }
]
```