**PROGRAMS AND QUERIES USING MONGODB**

**Question 1**

> use workdb

//switched to db workdb

> db.createCollection("customerSale", {

    validator: {

       $jsonSchema: {

          bsonType: "object",

          required: [ "customer", "item", "sale" ],

          properties: {

            customer: {

                bsonType: "object",

                required: [ "customer\_id", "customer\_name" ],

                properties: {

                   customer\_id: {

                      bsonType: "string",

                      description: "must be a string and required"

                   },

                   customer\_name: {

                      bsonType: "string",

                      description: "must be a string and is required"

                   }

                }

             },

             item: {

                bsonType: "object",

                required: [ "item\_id", "item\_name", "item\_price" ],

                properties: {

                   item\_id: {

                      bsonType: "string",

                      description: "must be a string and is required"

                   },

                   item\_name: {

                      bsonType: "string",

                      description: "must be a string and is required"

                   },

                   item\_price: {

                      bsonType: "double",

                      description: "must be a double and is required"

                   }

                }

             },

             sale: {

                bsonType: "object",

                required: [ "bill\_no", "bill\_date" ],

                properties: {

                   bill\_no: {

                      bsonType: "string",

                      description: "must be a string and is required"

                   },

                   bill\_date: {

                      bsonType: "date",

                      description: "must be a string and is required"

                   }

                }

             }

          }

       }

    }

 })

//{"ok" : 1}

> db.customerSale.insert( {

     customer: {

         customer\_id:"FMP0000021",

         customer\_name:"Frimpong Michael"

     },

     item:{

         item\_id:"DRNK00034",

         item\_name:"Coca Cola",

         item\_price:40.00

     },

     sale:{

         bill\_no:"RCPT0000021",

         bill\_date: new Date(2021-09-12)

     }

 })

 //WriteResult({ "nInserted" : 1 })

//Result: Our schema for Customer-sale scenario is tested and working successfully.

**Question 2**

> use schooldb

//switched to db schooldb

> db.createCollection("studentLibrary", {

    validator: {

       $jsonSchema: {

          bsonType: "object",

          required: [ "student","book", "iss\_rec" ],

          properties: {

             student: {

                bsonType: "object",

                required: [ "student\_id", "student\_name" ],

                properties: {

                   student\_id: {

                      bsonType: "string",

                      description: "must be a string and is required"

                   },

                   student\_name: {

                      bsonType: "string",

                      description: "must be a string and is required"

                   },

                   student\_phone: {

                      bsonType: "string",

                      description: "must be an string if it exist"

                   }

                }

             },

             book: {

               bsonType: "object",

               required: [ "book\_id", "book\_name", "author" ],

               properties: {

                  book\_id: {

                     bsonType: "string",

                     description: "must be a string and is required"

                  },

                  book\_name: {

                     bsonType: "string",

                     description: "must be a string and is required"

                  },

                  author: {

                     bsonType: "string",

                     description: "must be a string and is required"

                  }

               }

             },

             iss\_rec: {

               bsonType: "object",

               required: [ "iss\_no", "iss\_date" ],

               properties: {

                  iss\_no: {

                     bsonType: "string",

                     description: "must be a string and is required"

                  },

                  iss\_date: {

                     bsonType: "date",

                     description: "must be a date and is required"

                  },

               }

             },

             date\_of\_collection: {

               bsonType: "date",

               description: "must be a date and is required"

             },

             date\_of\_return: {

               bsonType: "date",

               description: "must be a date and is required"

             },

          }

       }

    }

 })

 //{"ok" : 1}

> db.studentLibrary.insert( {

   student:{

      student\_id:"ABD20210001",

      student\_name:"Michael Frimpong",

      student\_phone:"8923126743"

   },

   book:{

      book\_id:"SCI201034",

      book\_name:"BioTechnology",

      author:"Dr Brethney Trainor"

   },

   iss\_rec:{

      iss\_no:"ISS015500010N",

      iss\_date: new Date("2021-10-06")

   },

   date\_of\_collection: new Date("2021-10-06"),

   date\_of\_return: new Date("2021-10-20")

})

//WriteResult({ "nInserted" : 1 })

//Result: Our Schema for Student-library Scenario is tested and working successfully.

**Question 3**

//create the database

> use schooldb

//create collection

db.createCollection("studentDetails")

//insert some documents into your created collection

> db.studentDetails.insertMany(

    [

        {

            First\_Name: "Radhika",

            Last\_Name: "Sharma",

            Date\_Of\_Birth: "1995-09-26",

            e\_mail: "radhika\_sharma.123@gmail.com",

            phone: "7034587542"

        },

        {

            First\_Name: "Rachel",

            Last\_Name: "Christopher",

            Date\_Of\_Birth: "1990-02-16",

            e\_mail: "Rachel\_Christopher.123@gmail.com",

            phone: "8543210009"

        },

        {

            First\_Name: "Fathima",

            Last\_Name: "Sheik",

            Date\_Of\_Birth: "1990-02-16",

            e\_mail: "Fathima\_Sheik.123@gmail.com",

            phone: "8702805432"

        },

        {

            First\_Name: "Daniel",

            Last\_Name: "Wellbeck",

            Date\_of\_Birth: "1996-10-21",

            e\_mail: "DanielWellbeck.123@gmail.com",

            phone: "9256900321"

        },

        {

            First\_Name: "Manoj",

            Last\_Name: "Kumar",

            Date\_of\_Birth: "1993-04-11",

            e\_mail: "ManojKumar.123@gmail.com",

            phone: "8732690456"

        }

    ]

)

//write queries to retrieve your data.

> db.studentDetails.find().pretty()

> db.studentDetails.findOne()

> db.studentDetails.findOne({First\_Name:"Fathima"})

> db.studentDetails.find({$and:[{First\_Name:"Fathima"},{e\_mail:"Fathima\_Sheik.123@gmail.com"}]}).pretty()

> db.studentDetails.find({$or:[{First\_Name:"Fathima"},{First\_Name:"Manoj"}]}).pretty()

> db.studentDetails.find({$nor:[{First\_Name:"Manoj"},{Last\_Name:"Kumar"}]}).pretty()

//Result: Therefore data retrieval of data from the mongo database is successful.

**Question 4**

//create the database

> use schooldb

//create collection

db.createCollection("studentDetails")

//insert some documents into your created collection

> db.studentDetails.insertMany(

    [

        {

            First\_Name: "Radhika",

            Last\_Name: "Sharma",

            Date\_Of\_Birth: "1995-09-26",

            e\_mail: "radhika\_sharma.123@gmail.com",

            phone: "7034587542"

        },

        {

            First\_Name: "Rachel",

            Last\_Name: "Christopher",

            Date\_Of\_Birth: "1990-02-16",

            e\_mail: "Rachel\_Christopher.123@gmail.com",

            phone: "8543210009"

        },

        {

            First\_Name: "Fathima",

            Last\_Name: "Sheik",

            Date\_Of\_Birth: "1990-02-16",

            e\_mail: "Fathima\_Sheik.123@gmail.com",

            phone: "8702805432"

        },

        {

            First\_Name: "Daniel",

            Last\_Name: "Wellbeck",

            Date\_of\_Birth: "1996-10-21",

            e\_mail: "DanielWellbeck.123@gmail.com",

            phone: "9256900321"

        }

    ]

)

//Updating the records

> db.mycol.update({First\_Name:"Rachel"},{$set:{First\_Name:"Diva", Last\_Name:"Kumar"}})

> db.mycol.update({First\_Name:"Diva"},{$set:{Date\_Of\_Birth:"1995-10-20"}})

//Deleting a records

> db.studentDetails.remove({First\_Name:"Diva"})

//Viewing the records

db.studentDetails.find().sort({First\_Name:1}).pretty()

db.studentDetails.find().sort({First\_Name:-1}).pretty()

**Question 5**

//create a database

> use companydb

//switched to db salesCompanydb//

//create collection

> db.createCollection("users")

//creating embeded relationship

> db.users.insert({

        \_id:ObjectId("52ffc33cd85242f436000001"),

        contact: "987654321",

        dob: "01-01-1991",

        name: "Tom Benzamin",

        address: [

            {

                building: "22A, Eachinari Apt",

                pincode: 123456,

                city: "Coimbatore",

                state: "Tamil Nadu"

            },

            {

                building: "170A, Anna Apt",

                pincode: 456789,

                city: "Kochi",

                state: "Kerala"

            }

        ]

})

//inserting two document address

> db.users.insert({

    "\_id":ObjectId("52ffc4a5d85242602e000000"),

    "name": "Tom Hanks",

    "contact": "987654321",

    "dob": "01-01-1991"

 })

> db.users.insert({

    "\_id":ObjectId("52ffc4a5d85242602e000001"),

    "building": "22A, Eachinari Apt",

    "pincode": 123456,

    "city": "Coimbatore",

    "state": "Tamil Nadu"

 } )

//insert document with reference

> db.users.insert({

    "\_id":ObjectId("52ffc33cd85242f436000002"),

    "contact": "987654321",

    "dob": "01-01-1991",

    "name": "Tom Benzamin",

    "address\_ids": [

       ObjectId("52ffc4a5d85242602e000000"),

       ObjectId("52ffc4a5d85242602e000001")

    ]

 })

 //Result: Therefore the program was successfully executed.