EXPERIMENT- 10

Student Name: Ruchi Sharma **UID: 23BCS10713**

Branch: BE-CSE **Section/Group: KRG 1(B)**

Semester: 05 **Date of Performance: 28/10/25**

Subject Name: ADBMS **Subject Code: 23CSP-333**

1. Aim:
To perform CRUD operations and aggregation using **MongoDB**, a NoSQL document-based database.

2. Objective:
- Learn creation of databases and collections in MongoDB.
- Execute Insert, Read, Update, and Delete operations.

3. Tools / Software
- MongoDB
- Mongo Shell
- Sample Dataset: Car Dealership Data

4. Program:

```
PS C:\Users\ruchi> mongosh
Current Mongosh Log ID: 6901eb68e8ffe9c747cebea3
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2
.5.8
Using MongoDB:          8.2.1
Using Mongosh:          2.5.8

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-10-28T10:47:21.504+05:30: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
------
```

-- show dbs

```
test> show dbs
admin           40.00 KiB
carDealership    8.00 KiB
config          96.00 KiB
local           40.00 KiB
```

-- use car_dealership

```
test> use car_dealership
switched to db car_dealership
car_dealership>
```

INSERTION OPERATION:

```
db.createCollection("cars")
db.cars.insertMany([
  { maker: "Hyundai", model: "i20", fuel_type: "Petrol" },
  { maker: "Tata", model: "Nexon", fuel_type: "Diesel" },
  { maker: "Kia", model: "Seltos", fuel_type: "Petrol" },
  { maker: "Maruti", model: "Swift", fuel_type: "CNG" }
])
```

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6901ec50e8ffe9c747cebea4'),
    '1': ObjectId('6901ec50e8ffe9c747cebea5'),
    '2': ObjectId('6901ec50e8ffe9c747cebea6'),
    '3': ObjectId('6901ec50e8ffe9c747cebea7')
  }
}
```

READ OPERATION:

```
db.cars.find()
db.cars.find({ fuel_type: "Petrol" })
db.cars.find({}, { model: 1, _id: 0 })
```

```
[
  { model: 'i20' },
  { model: 'Nexon' },
  { model: 'Seltos' },
  { model: 'Swift' }
]
```

UPDATE OPERATION:

```
db.cars.updateOne({ model: "i20" }, { $set: { fuel_type: "Hybrid" } })
db.cars.updateMany({}, { $set: { color: "White" } })
db.cars.updateOne({ model: "Nexon" }, { $push: { features: "Sunroof" } })
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

DELETE OPERATION:

db.cars.deleteOne({ model: "Swift" })

```
car_dealership> db.cars.deleteOne({ model: "Swift" })
{ acknowledged: true, deletedCount: 1 }
```

AGGREGATION:

db.cars.aggregate([{ $group: { _id: "$maker", totalCars: { $sum: 1 } } }])

```
[
  { _id: 'Kia', totalCars: 1 },
  { _id: 'Hyundai', totalCars: 1 },
  { _id: 'Tata', totalCars: 1 }
]
```