



## Experiment 10

**Student Name:** Anshum  
**Branch:** CSE  
**Semester:** 5th  
**Subject Name:** ADBMS

**UID:** 23BCS11330  
**Section/Group:** KRG 3\_A  
**Date of Performance:** 30/10/2025  
**Subject Code:** 23CSP-333

### **1. Aim:**

- Create and manage databases and collections.
- Perform CRUD operations — Create, Read, Update, and Delete documents.
- Use aggregation for grouping and summarizing data.
- Understand how to modify, query, and manage collections efficiently in MongoDB.
- Gain practical experience with real-world data handling through command-line operations.

### **2. DBMS script and output:**

#### **1. Create Database**

```
test> use local
switched to db local
local> db.createCollection("cars")
{ ok: 1 }
local> |
```

#### **2. Insert Operations**

```
local> db.cars.insertOne(
... {
...   "maker": "Tata",
...   "model": "Nexon",
...   "fuel_type": "Petrol",
...   "transmission": "Automatic",
...   "engine": {
...     "type": "Turbocharged",
...     "cc": 1199,
...     "torque": "170 Nm"
...   },
...   "features": [
...     "Touchscreen",
...     "Reverse Camera",
...     "Bluetooth Connectivity"
...   ],
...   "sunroof": false,
...   "airbags": 2
... }
{
  acknowledged: true,
  insertedId: ObjectId('690509469dc2419e22cb0ce2')
}
local> |
```

```
local> db.cars.insertMany(
... [
...   {
...     "maker": "Hyundai",
...     "model": "Creta",
...     "fuel_type": "Diesel",
...     "transmission": "Manual",
...     "engine": {
...       "type": "Naturally Aspirated",
...       "cc": 1493,
...       "torque": "250 Nm"
...     },
...     "features": [
...       "Sunroof",
...       "Leather Seats",
...       "Wireless Charging",
...       "Ventilated Seats",
...       "Bluetooth"
...     ]
...   },
... ]
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('690509ac9dc2419e22cb0ce3'),
    '1': ObjectId('690509ac9dc2419e22cb0ce4'),
    '2': ObjectId('690509ac9dc2419e22cb0ce5'),
    '3': ObjectId('690509ac9dc2419e22cb0ce6')
  }
}
local> |
```

### 3. Find Operations

```
local> db.cars.find()
[
  {
    _id: ObjectId('690509469dc2419e22cb0ce2'),
    maker: 'Tata',
    model: 'Nexon',
    fuel_type: 'Petrol',
    transmission: 'Automatic',
    engine: { type: 'Turbocharged', cc: 1199, torque: '170 Nm' },
    features: [ 'Touchscreen', 'Reverse Camera', 'Bluetooth Connectivity' ],
    sunroof: false,
    airbags: 2
  },
  {
    _id: ObjectId('690509ac9dc2419e22cb0ce3'),
    maker: 'Hyundai',
    model: 'Creta',
    fuel_type: 'Diesel',
    transmission: 'Manual',
    engine: { type: 'Naturally Aspirated', cc: 1493, torque: '250 Nm' },
    features: [
      'Sunroof',
      'Leather Seats',
      'Wireless Charging',
      'Ventilated Seats',
      'Bluetooth'
    ]
  }
],
```

```
local> db.cars.findOne()
{
  _id: ObjectId('690509469dc2419e22cb0ce2'),
  maker: 'Tata',
  model: 'Nexon',
  fuel_type: 'Petrol',
  transmission: 'Automatic',
  engine: { type: 'Turbocharged', cc: 1199, torque: '170 Nm' },
  features: [ 'Touchscreen', 'Reverse Camera', 'Bluetooth Connectivity' ],
  sunroof: false,
  airbags: 2
}
```

```
local> db.cars.find({}, {model:true})
[
  { _id: ObjectId('690509469dc2419e22cb0ce2'), model: 'Nexon' },
  { _id: ObjectId('690509ac9dc2419e22cb0ce3'), model: 'Creta' },
  { _id: ObjectId('690509ac9dc2419e22cb0ce4'), model: 'Baleno' },
  { _id: ObjectId('690509ac9dc2419e22cb0ce5'), model: 'XUV500' },
  { _id: ObjectId('690509ac9dc2419e22cb0ce6'), model: 'City' }
]
local> |
```

```
local> db.cars.find({}, {_id:false,model:true})
[
  { model: 'Nexon' },
  { model: 'Creta' },
  { model: 'Baleno' },
  { model: 'XUV500' },
  { model: 'City' }
]
local> |
```

```
local> db.cars.find({"fuel_type":"Petrol"})
[
  {
    _id: ObjectId('690509469dc2419e22cb0ce2'),
    maker: 'Tata',
    model: 'Nexon',
    fuel_type: 'Petrol',
    transmission: 'Automatic',
    engine: { type: 'Turbocharged', cc: 1199, torque: '170 Nm' },
    features: [ 'Touchscreen', 'Reverse Camera', 'Bluetooth Connectivity' ],
    sunroof: false,
    airbags: 2
  },
  {
    _id: ObjectId('690509ac9dc2419e22cb0ce4'),
    maker: 'Maruti Suzuki',
    model: 'Baleno',
    fuel_type: 'Petrol',
```

## 5 Update Operations

```
local> db.cars.updateOne(
... {model:"Nexon"},
... {$set:{color: "Red"}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
local> |
```

```
local> db.cars.updateOne( { model: "Nexon" }, { $push: { features: "Heated Seats" } } )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
local> |
```

```
local> db.cars.updateMany({fuel_type: "Diesel"}, {$set: {alloys: "yes" }})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
local> |
```

## 6. Delete Operations

```
local> db.cars.deleteOne({fuel_type: "Petrol"})
{ acknowledged: true, deletedCount: 1 }
local> |
```

```
local> db.cars.deleteOne({fuel_type: "Petrol"})
{ acknowledged: true, deletedCount: 1 }
local> db.cars.deleteMany({})
{ acknowledged: true, deletedCount: 4 }
local> |
```