University Institute of Engineering

Department of Computer Science & Engineering

EXPERIMENT: 10

NAME : Johnson Kumar **UID** : 23BCS12654

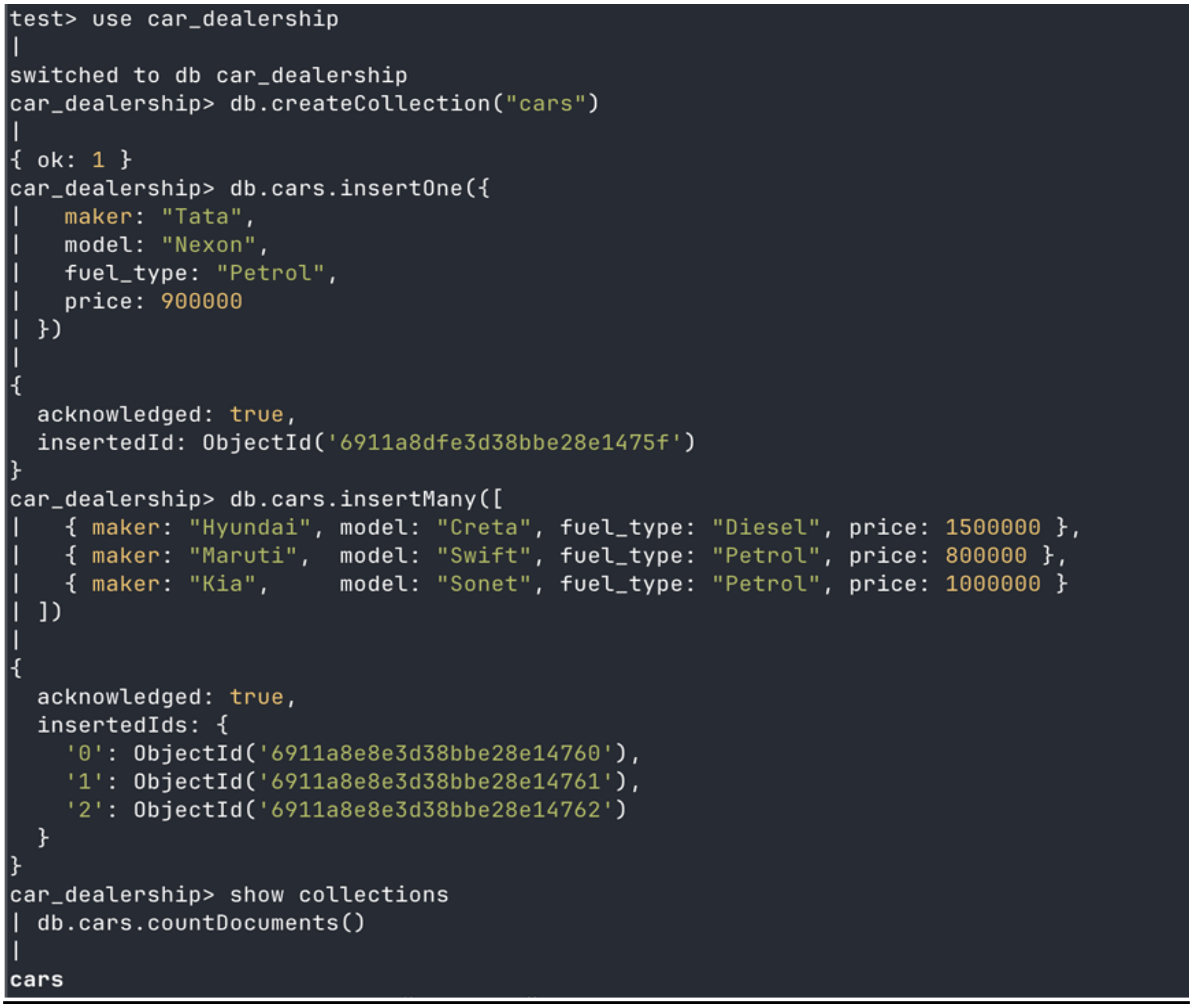
SECTION : KRG_2A **SEMESTER**: 5TH

SUBJECT CODE: 23CSP-339 **SUBJECT** : ADBMS

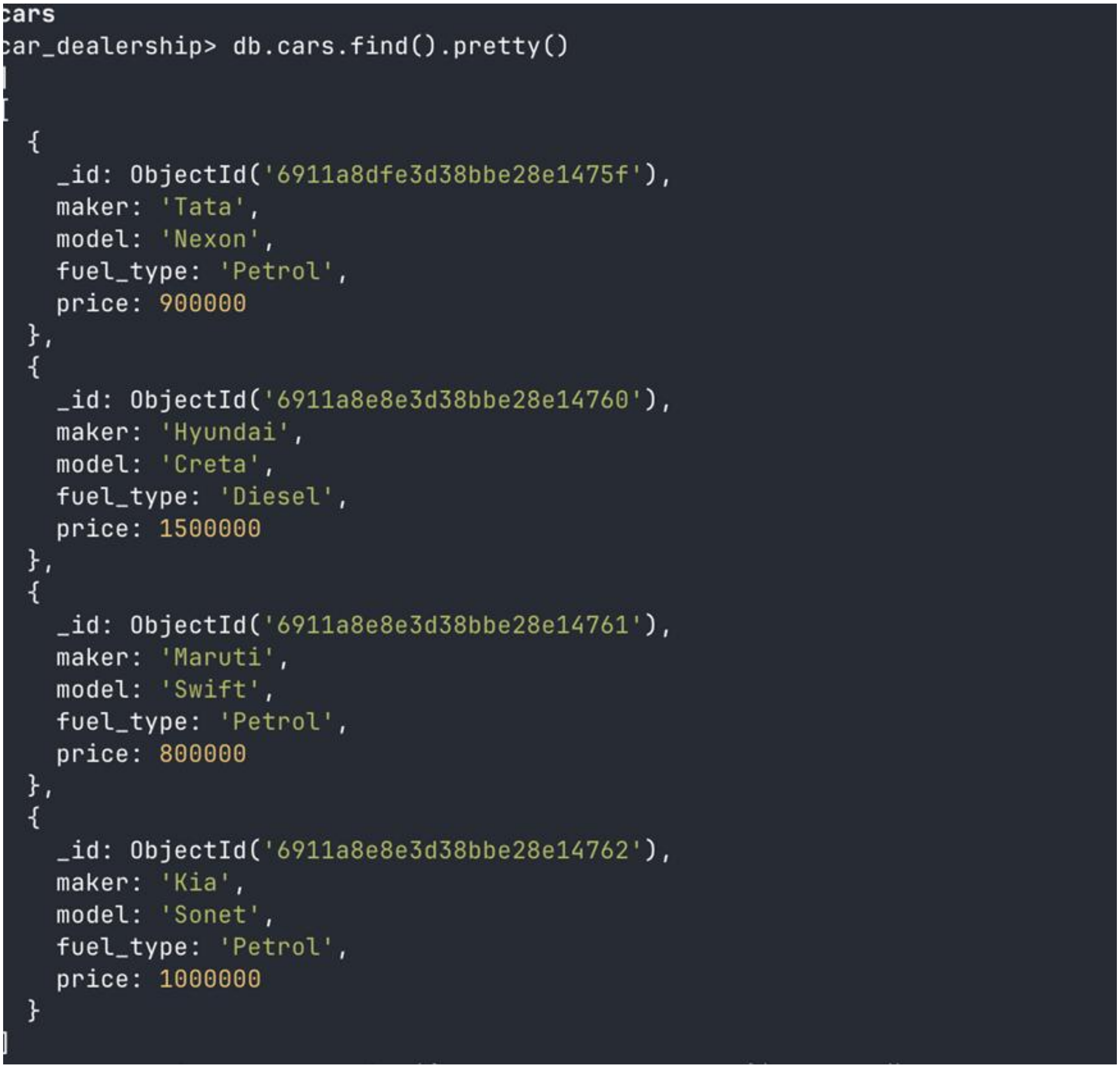
Problem Statement:

Implementation of MongoDB CRUD and Aggregation Operations Using Mongo Shell

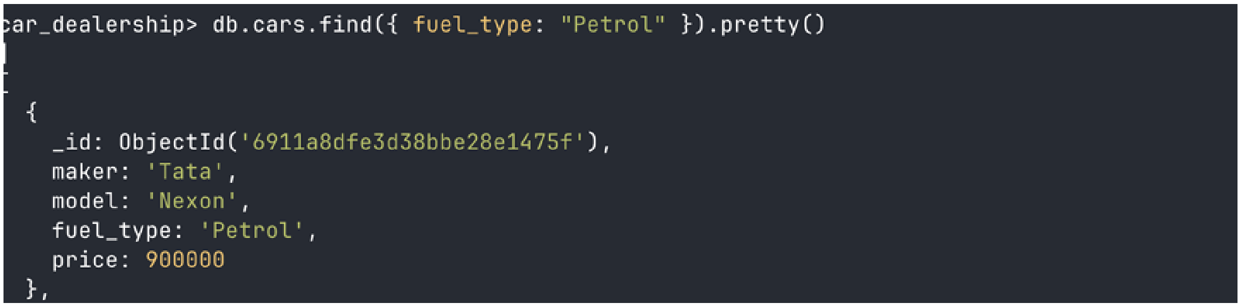
Solution:



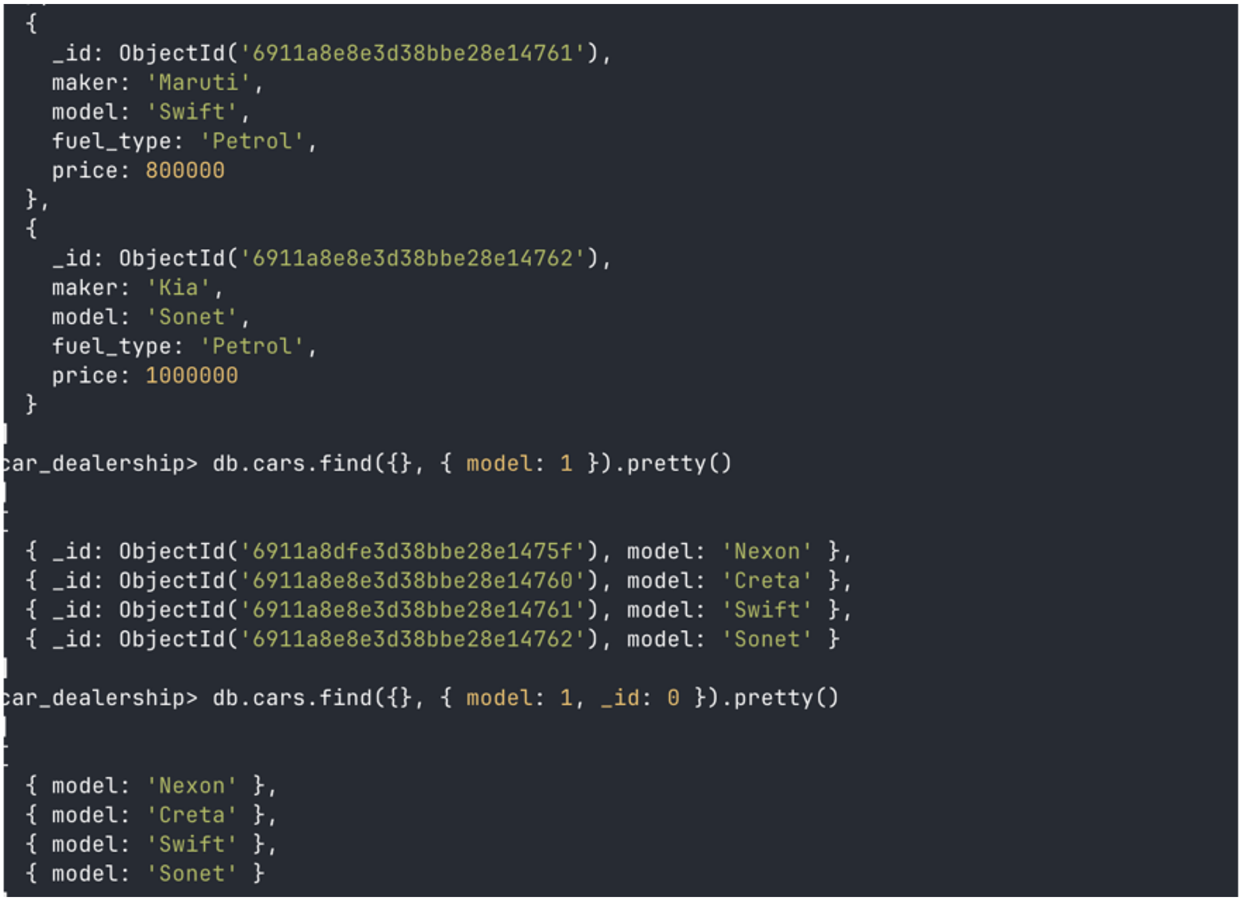
```
test> use car_dealership
|
switched to db car_dealership
car_dealership> db.createCollection("cars")
|
{ ok: 1 }
car_dealership> db.cars.insertOne({
|   maker: "Tata",
|   model: "Nexon",
|   fuel_type: "Petrol",
|   price: 900000
| })
|
{
  acknowledged: true,
  insertedId: ObjectId('6911a8dfe3d38bbe28e1475f')
}
car_dealership> db.cars.insertMany([
|   { maker: "Hyundai", model: "Creta", fuel_type: "Diesel", price: 1500000 },
|   { maker: "Maruti",  model: "Swift", fuel_type: "Petrol", price: 800000 },
|   { maker: "Kia",     model: "Sonet", fuel_type: "Petrol", price: 1000000 }
| ])
|
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6911a8e8e3d38bbe28e14760'),
    '1': ObjectId('6911a8e8e3d38bbe28e14761'),
    '2': ObjectId('6911a8e8e3d38bbe28e14762')
  }
}
car_dealership> show collections
| db.cars.countDocuments()
|
cars
```



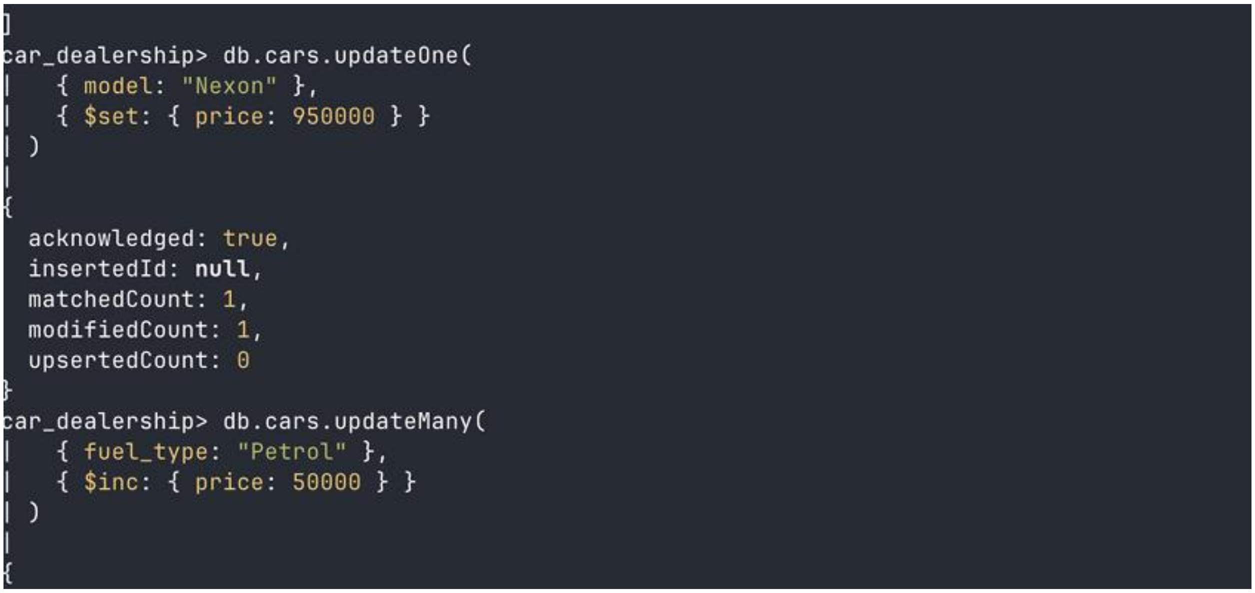
```
cars
car_dealership> db.cars.find().pretty()
[
  {
    _id: ObjectId('6911a8dfe3d38bbe28e1475f'),
    maker: 'Tata',
    model: 'Nexon',
    fuel_type: 'Petrol',
    price: 900000
  },
  {
    _id: ObjectId('6911a8e8e3d38bbe28e14760'),
    maker: 'Hyundai',
    model: 'Creta',
    fuel_type: 'Diesel',
    price: 1500000
  },
  {
    _id: ObjectId('6911a8e8e3d38bbe28e14761'),
    maker: 'Maruti',
    model: 'Swift',
    fuel_type: 'Petrol',
    price: 800000
  },
  {
    _id: ObjectId('6911a8e8e3d38bbe28e14762'),
    maker: 'Kia',
    model: 'Sonet',
    fuel_type: 'Petrol',
    price: 1000000
  }
]
```



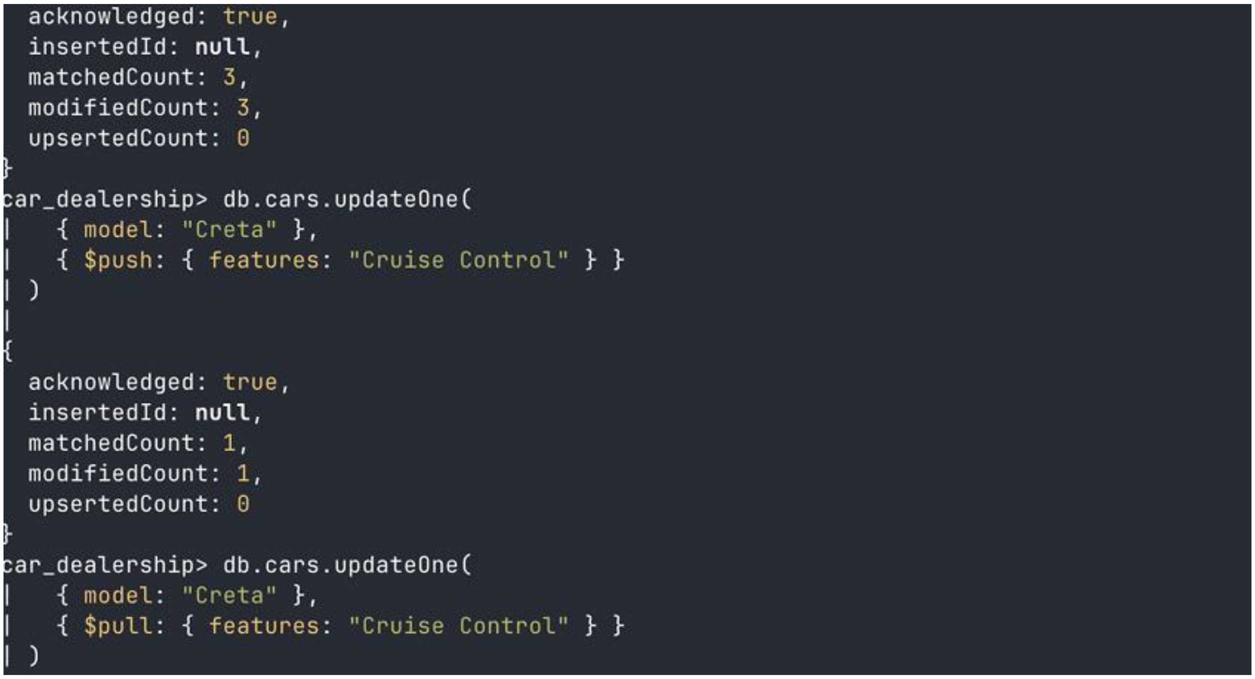
```
car_dealership> db.cars.find({ fuel_type: "Petrol" }).pretty()
[
  {
    _id: ObjectId('6911a8dfe3d38bbe28e1475f'),
    maker: 'Tata',
    model: 'Nexon',
    fuel_type: 'Petrol',
    price: 900000
  },
```



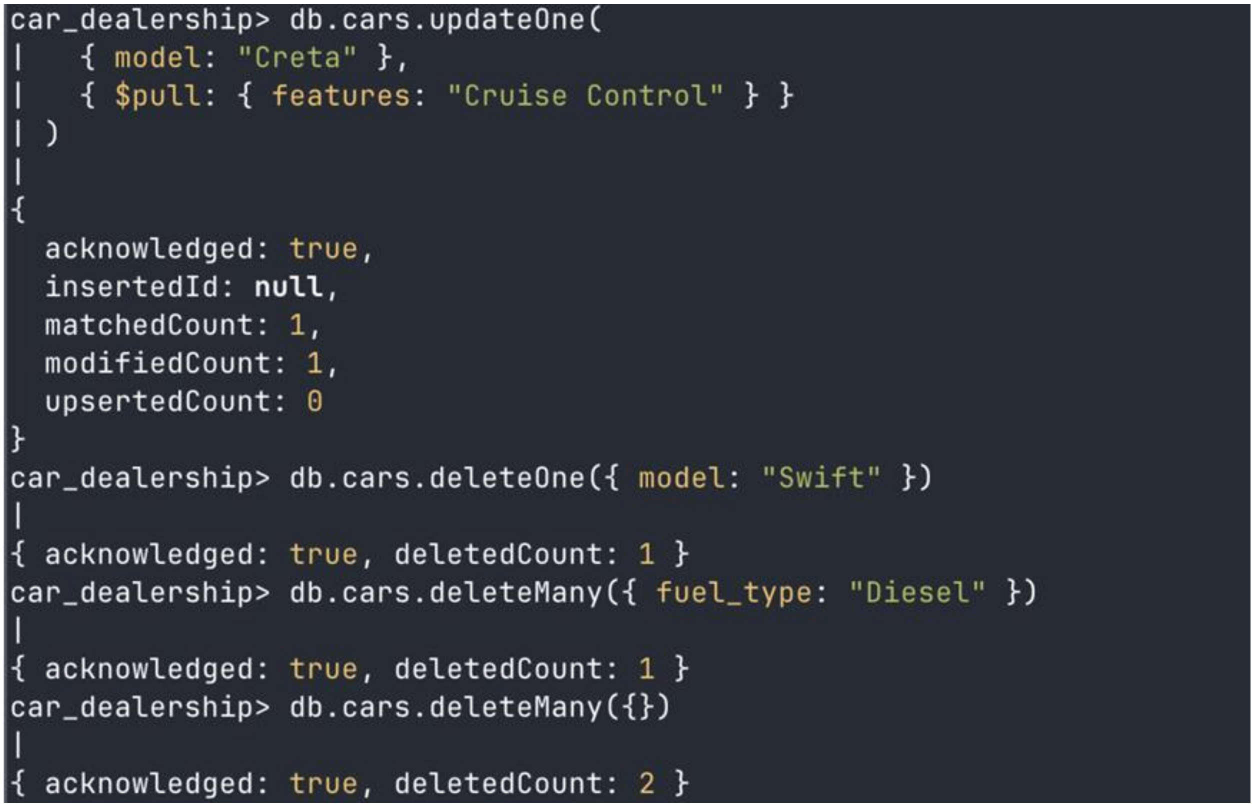
```
  {
    _id: ObjectId('6911a8e8e3d38bbe28e14761'),
    maker: 'Maruti',
    model: 'Swift',
    fuel_type: 'Petrol',
    price: 800000
  },
  {
    _id: ObjectId('6911a8e8e3d38bbe28e14762'),
    maker: 'Kia',
    model: 'Sonet',
    fuel_type: 'Petrol',
    price: 1000000
  }
]
car_dealership> db.cars.find({}, { model: 1 }).pretty()
[
  { _id: ObjectId('6911a8dfe3d38bbe28e1475f'), model: 'Nexon' },
  { _id: ObjectId('6911a8e8e3d38bbe28e14760'), model: 'Creta' },
  { _id: ObjectId('6911a8e8e3d38bbe28e14761'), model: 'Swift' },
  { _id: ObjectId('6911a8e8e3d38bbe28e14762'), model: 'Sonet' }
]
car_dealership> db.cars.find({}, { model: 1, _id: 0 }).pretty()
[
  { model: 'Nexon' },
  { model: 'Creta' },
  { model: 'Swift' },
  { model: 'Sonet' }
```



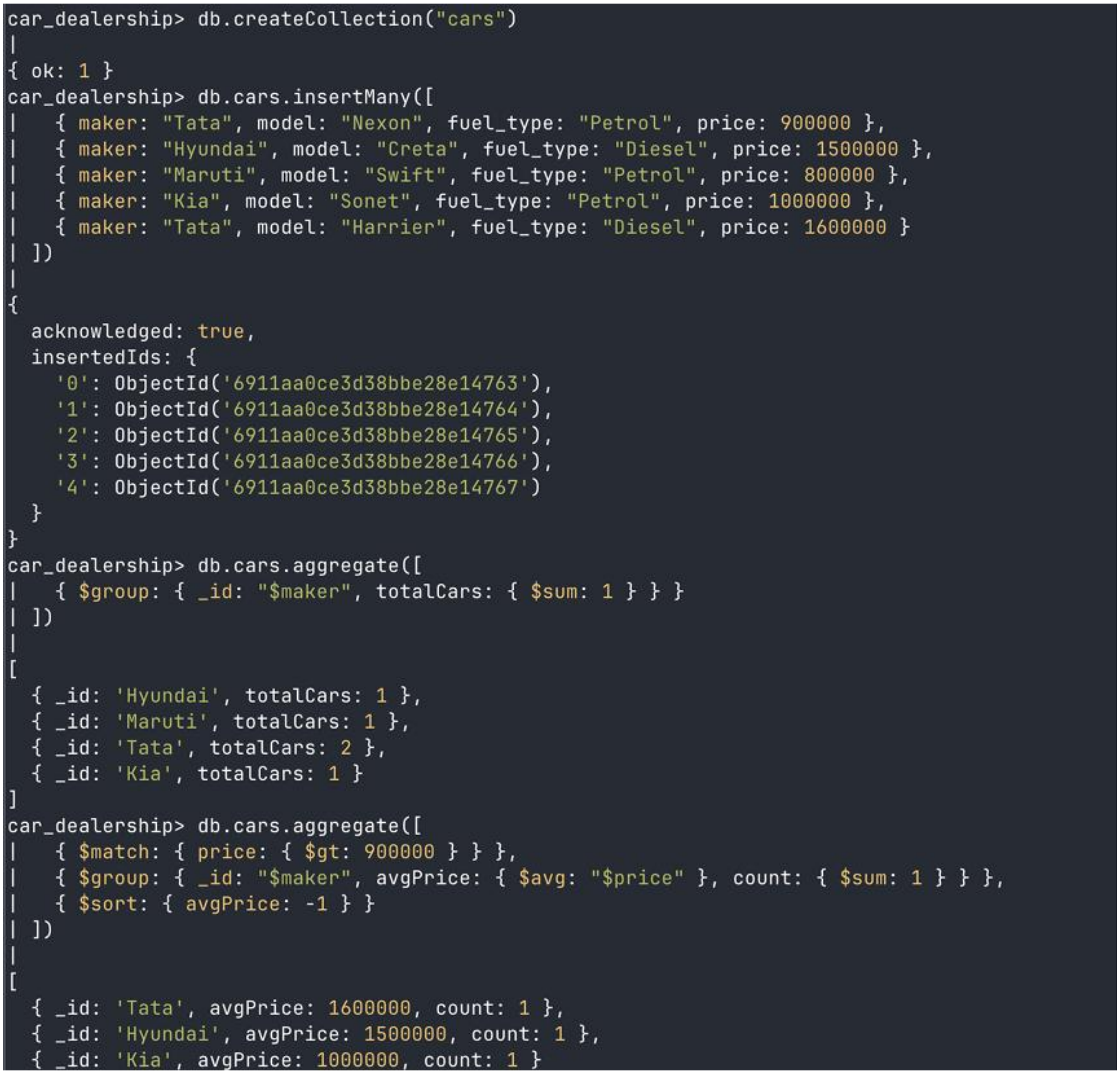
```
]
car_dealership> db.cars.updateOne(
|   { model: "Nexon" },
|   { $set: { price: 950000 } }
| )
|
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
car_dealership> db.cars.updateMany(
|   { fuel_type: "Petrol" },
|   { $inc: { price: 50000 } }
| )
|
{
```



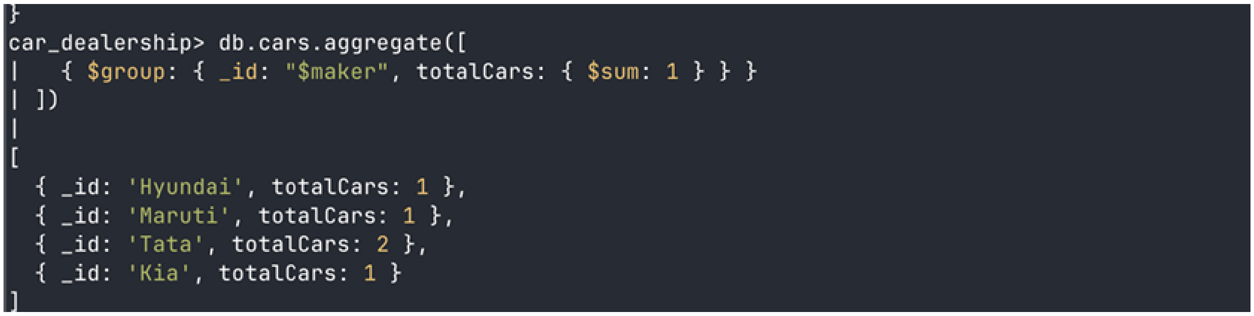
```
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
car_dealership> db.cars.updateOne(
|   { model: "Creta" },
|   { $push: { features: "Cruise Control" } }
| )
|
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
car_dealership> db.cars.updateOne(
|   { model: "Creta" },
|   { $pull: { features: "Cruise Control" } }
| )
```



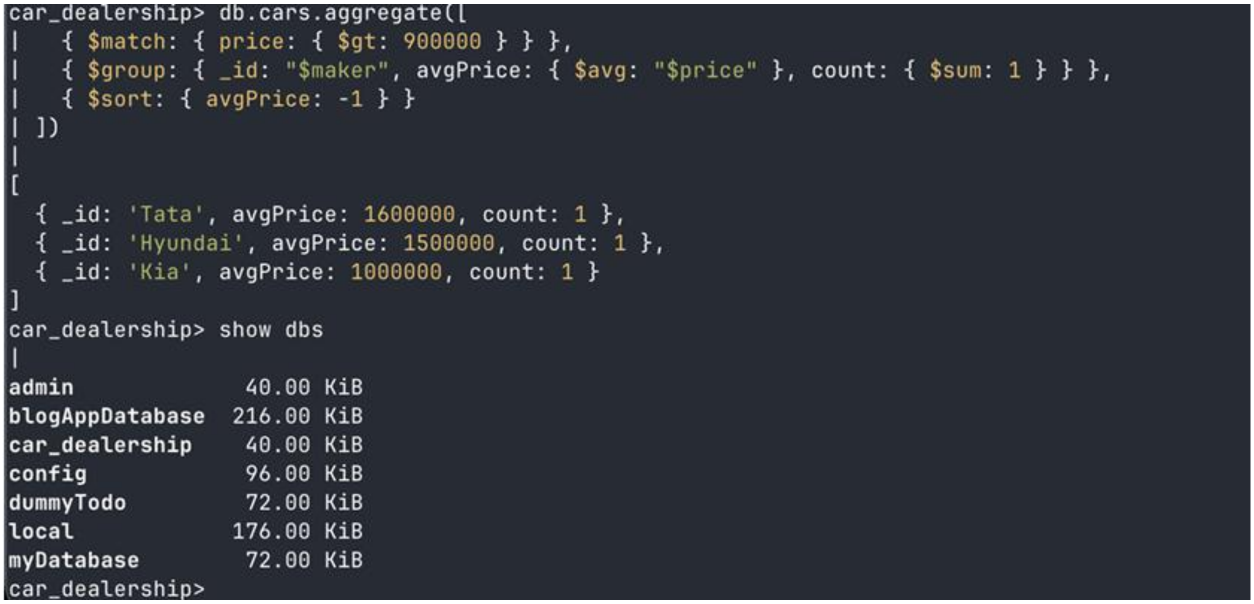
```
car_dealership> db.cars.updateOne(
|   { model: "Creta" },
|   { $pull: { features: "Cruise Control" } }
| )
|
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
car_dealership> db.cars.deleteOne({ model: "Swift" })
|
{ acknowledged: true, deletedCount: 1 }
car_dealership> db.cars.deleteMany({ fuel_type: "Diesel" })
|
{ acknowledged: true, deletedCount: 1 }
car_dealership> db.cars.deleteMany({})
|
{ acknowledged: true, deletedCount: 2 }
```



```
car_dealership> db.createCollection("cars")
|
{ ok: 1 }
car_dealership> db.cars.insertMany([
|   { maker: "Tata", model: "Nexon", fuel_type: "Petrol", price: 900000 },
|   { maker: "Hyundai", model: "Creta", fuel_type: "Diesel", price: 1500000 },
|   { maker: "Maruti", model: "Swift", fuel_type: "Petrol", price: 800000 },
|   { maker: "Kia", model: "Sonet", fuel_type: "Petrol", price: 1000000 },
|   { maker: "Tata", model: "Harrier", fuel_type: "Diesel", price: 1600000 }
| ])
|
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6911aa0ce3d38bbe28e14763'),
    '1': ObjectId('6911aa0ce3d38bbe28e14764'),
    '2': ObjectId('6911aa0ce3d38bbe28e14765'),
    '3': ObjectId('6911aa0ce3d38bbe28e14766'),
    '4': ObjectId('6911aa0ce3d38bbe28e14767')
  }
}
car_dealership> db.cars.aggregate([
|   { $group: { _id: "$maker", totalCars: { $sum: 1 } } }
| ])
|
[
  { _id: 'Hyundai', totalCars: 1 },
  { _id: 'Maruti', totalCars: 1 },
  { _id: 'Tata', totalCars: 2 },
  { _id: 'Kia', totalCars: 1 }
]
car_dealership> db.cars.aggregate([
|   { $match: { price: { $gt: 900000 } } },
|   { $group: { _id: "$maker", avgPrice: { $avg: "$price" }, count: { $sum: 1 } } },
|   { $sort: { avgPrice: -1 } }
| ])
|
[
  { _id: 'Tata', avgPrice: 1600000, count: 1 },
  { _id: 'Hyundai', avgPrice: 1500000, count: 1 },
  { _id: 'Kia', avgPrice: 1000000, count: 1 }
```



```
]
car_dealership> db.cars.aggregate([
|   { $group: { _id: "$maker", totalCars: { $sum: 1 } } }
| ])
|
[
  { _id: 'Hyundai', totalCars: 1 },
  { _id: 'Maruti', totalCars: 1 },
  { _id: 'Tata', totalCars: 2 },
  { _id: 'Kia', totalCars: 1 }
]
```



```
car_dealership> db.cars.aggregate([
|   { $match: { price: { $gt: 900000 } } },
|   { $group: { _id: "$maker", avgPrice: { $avg: "$price" }, count: { $sum: 1 } } },
|   { $sort: { avgPrice: -1 } }
| ])
|
[
  { _id: 'Tata', avgPrice: 1600000, count: 1 },
  { _id: 'Hyundai', avgPrice: 1500000, count: 1 },
  { _id: 'Kia', avgPrice: 1000000, count: 1 }
]
car_dealership> show dbs
|
admin              40.00 KiB
blogAppDatabase   216.00 KiB
car_dealership     40.00 KiB
config             96.00 KiB
dummyTodo          72.00 KiB
local             176.00 KiB
myDatabase         72.00 KiB
car_dealership>
```

Learning outcomes:

- I learned how to initialize and connect to a local MongoDB instance using mongosh, and understood basic shell commands.
- I practiced creating a new database and collection, and inserting documents using both insertOne() and insertMany() methods.
- I explored various query operations including find(), filtering by fields, and projecting specific attributes like model.
- I learned how to update documents using $set, $inc, $push, and $pull operators to modify fields and arrays.
- I performed deletion operations using deleteOne(), deleteMany(), and drop() to manage collection data.
- I applied aggregation techniques like $group, $match, and $sort to analyze data by maker, fuel_type, and price statistics.