

Experiment10

Code:

Last login: Fri Nov 7 09:32:01 on console

[~]\$ mongosh

Current Mongosh Log ID: 6911a8a2e3d38bbe28e1475e

Connecting to:

mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.6

Using MongoDB: 8.0.12

Using Mongosh: 2.5.6

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

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (<https://www.mongodb.com/legal/privacy-policy>).

You can opt-out by running the `disableTelemetry()` command.

The server generated these startup warnings when booting

2025-11-07T09:32:14.795+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

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

car_dealership> db.cars.aggregate([
| { $group: { _id: "$fuel_type", avgPrice: { $avg: "$price" } } }
| ])
|

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> 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>
```