

Microsoft Windows [Version 10.0.19045.4291]

(c) Microsoft Corporation. All rights reserved.

C:\Users\computer>mongosh

Current Mongosh Log ID: 6673c8cff253bf437546b798

Connecting to:

mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.5

Using MongoDB: 7.0.9

Using Mongosh: 2.2.5

mongosh 2.2.9 is available for download: <https://www.mongodb.com/try/download/shell>

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

The server generated these startup warnings when booting

2024-06-19T17:50:12.651+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

Student> use Vehicle

switched to db Vehicle

Vehicle> db.createCollection("VehicleDetails")

{ ok: 1 }

Vehicle> show collections

VehicleDetails

*****A*****

Vehicle> db.VehicleDetails.insertMany([
{Reg_No:"BRG446",Make:"Ford",Model:"Meteor",Year:2009,Colour:"White"},{Reg_No:"VRG655",Ma
ke:"BMW",Model:"Delivery Van",Year:2008,Colour:"Blue"}])

{

acknowledged: true,

insertedIds: {

```

    '0': ObjectId('6673d256f253bf437546b799'),
    '1': ObjectId('6673d256f253bf437546b79a')
  }
}

Vehicle> db.VehicleDetails.insertOne({Reg_No:"NMT667",Make:"Madza",Model:"Delivery
Van",Year:2010,Colour:"Green"})

{
  acknowledged: true,
  insertedId: ObjectId('6673d307f253bf437546b79b')
}

Vehicle>
db.VehicleDetails.insertOne({Reg_No:"CCT899",Make:"Toyota",Model:"HiLuxe",Year:2009,Colour:"Re
d"})

{
  acknowledged: true,
  insertedId: ObjectId('6673d38bf253bf437546b79c')
}

Vehicle>
db.VehicleDetails.insertOne({Reg_No:"FGR122",Make:"Mitsubishi",Model:"Magna",Year:2007,Colour
:"Purple"})

{
  acknowledged: true,
  insertedId: ObjectId('6673d3adf253bf437546b79d')
}

*****B*****

Vehicle> db.VehicleDetails.find().pretty()

[
  {
    _id: ObjectId('6673d256f253bf437546b799'),
    Reg_No: 'BRG446',
    Make: 'Ford',
    Model: 'Meteor',

```

```
Year: 2009,
Colour: 'White'
},
{
  _id: ObjectId('6673d256f253bf437546b79a'),
  Reg_No: 'VRG655',
  Make: 'BMW',
  Model: 'Coupe',
  Year: 2008,
  Colour: 'Blue'
},
{
  _id: ObjectId('6673d307f253bf437546b79b'),
  Reg_No: 'NMT667',
  Make: 'Madza',
  Model: 'Delivery Van',
  Year: 2010,
  Colour: 'Green'
},
{
  _id: ObjectId('6673d38bf253bf437546b79c'),
  Reg_No: 'CCT899',
  Make: 'Toyota',
  Model: 'HiLuxe',
  Year: 2009,
  Colour: 'Red'
},
{
  _id: ObjectId('6673d3adf253bf437546b79d'),
  Reg_No: 'FGR122',
  Make: 'Mitsubishi',
```

```
Model: 'Magna',
Year: 2007,
Colour: 'Purple'
}
]
```

*****C*****

```
Vehicle> db.VehicleDetails.find({Make: { $regex: "^T" } })
[
  {
    _id: ObjectId('6673d38bf253bf437546b79c'),
    Reg_No: 'CCT899',
    Make: 'Toyota',
    Model: 'HiLuxe',
    Year: 2009,
    Colour: 'Red'
  }
]
```

*****D*****

```
Vehicle> db.VehicleDetails.updateOne({ Reg_No:"FGR122" },{$set:{Colour:"Blue"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Vehicle> db.VehicleDetails.find({Reg_No:"FGR122"})
[
  {
```

```
_id: ObjectId('6673d3adf253bf437546b79d'),
Reg_No: 'FGR122',
Make: 'Mitsubishi',
Model: 'Magna',
Year: 2007,
Colour: 'Blue'
}
]
```

*****E*****

```
Vehicle> db.VehicleDetails.updateMany({}, { $set: { EnNo: null } })
```

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 5,
  upsertedCount: 0
}
```

```
Vehicle> db.VehicleDetails.find().pretty()
```

```
[
  {
    _id: ObjectId('6673d256f253bf437546b799'),
    Reg_No: 'BRG446',
    Make: 'Ford',
    Model: 'Meteor',
    Year: 2009,
    Colour: 'White',
    EnNo: null
  },
  {
    _id: ObjectId('6673d256f253bf437546b79a'),
```

```
Reg_No: 'VRG655',
Make: 'BMW',
Model: 'Coupe',
Year: 2008,
Colour: 'Blue',
EnNo: null
},
{
  _id: ObjectId('6673d307f253bf437546b79b'),
  Reg_No: 'NMT667',
  Make: 'Madza',
  Model: 'Delivery Van',
  Year: 2010,
  Colour: 'Green',
  EnNo: null
},
{
  _id: ObjectId('6673d38bf253bf437546b79c'),
  Reg_No: 'CCT899',
  Make: 'Toyota',
  Model: 'HiLuxe',
  Year: 2009,
  Colour: 'Red',
  EnNo: null
},
{
  _id: ObjectId('6673d3adf253bf437546b79d'),
  Reg_No: 'FGR122',
  Make: 'Mitsubishi',
  Model: 'Magna',
  Year: 2007,
```

```
    Colour: 'Blue',
    EnNo: null
  }
]
```

*****F*****

```
db.VehicleDetails.aggregate({ $match:{Year:2009}}, { $group:{_id:"$Year",count : {$sum:1} }})
{ "_id" : 2009, "count" : 2 }
```

*****G*****

```
Vehicle> db.VehicleDetails.deleteOne( { Reg_No:"BRG446" } )
```

```
{ acknowledged: true, deletedCount: 1 }
```

```
Vehicle> db.VehicleDetails.find()
```

```
[
  {
    _id: ObjectId('6673d256f253bf437546b79a'),
    Reg_No: 'VRG655',
    Make: 'BMW',
    Model: 'Coupe',
    Year: 2008,
    Colour: 'Blue',
    EnNo: null
  },
  {
    _id: ObjectId('6673d307f253bf437546b79b'),
    Reg_No: 'NMT667',
    Make: 'Madza',
    Model: 'Delivery Van',
    Year: 2010,
    Colour: 'Green',
    EnNo: null
  }
]
```

```

    },
    {
      _id: ObjectId('6673d38bf253bf437546b79c'),
      Reg_No: 'CCT899',
      Make: 'Toyota',
      Model: 'HiLuxe',
      Year: 2009,
      Colour: 'Red',
      EnNo: null
    },
    {
      _id: ObjectId('6673d3adf253bf437546b79d'),
      Reg_No: 'FGR122',
      Make: 'Mitsubishi',
      Model: 'Magna',
      Year: 2007,
      Colour: 'Blue',
      EnNo: null
    }
  ]

```

*****H*****

```
db.createUser({user:"Ram",pwd:"ram123",roles:[ { role : "readWrite",db:"Vehicle"} ]})
```

Successfully added user: {

```

  "user" : "Ram",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "Vehicle"
    }
  ]
}
```