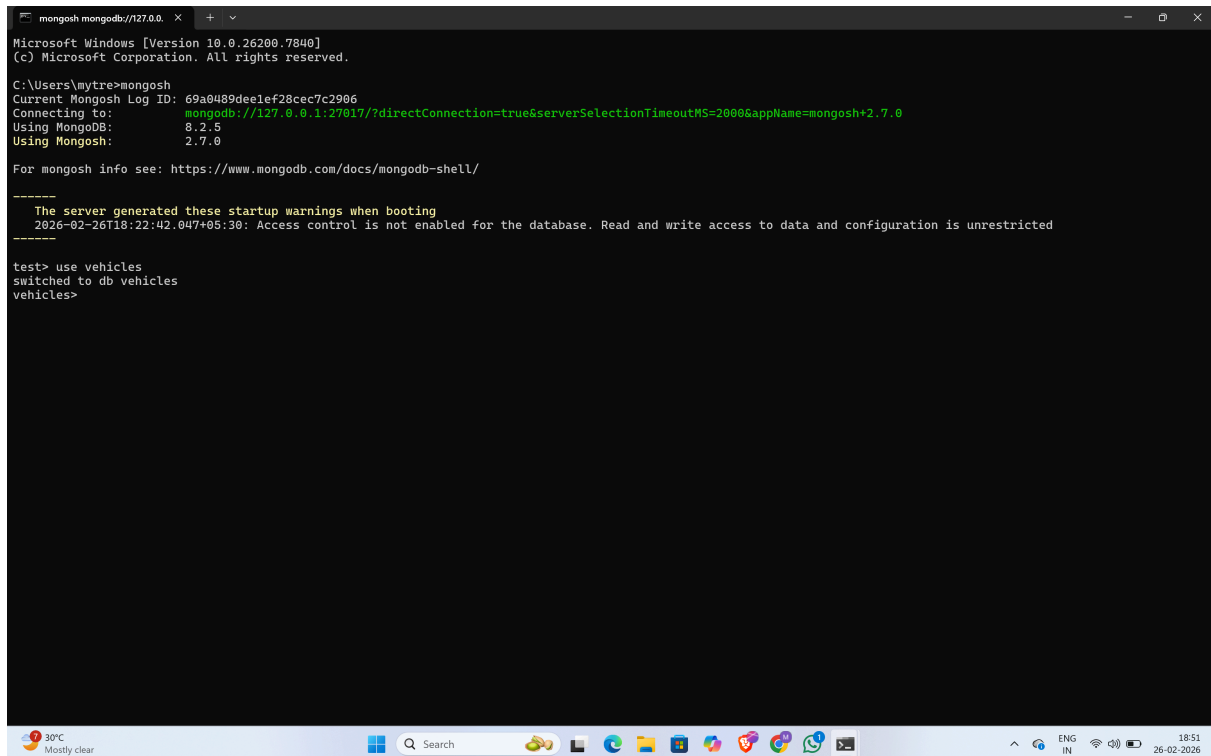


MONGODB - COMMANDS

Vehicles - MongoDB commands

1. Use MongoDB to implement the following DB operations
 - Create a database called 'vehicles' and write a MongoDB query to select database as "vehicles".

A screenshot of a Windows terminal window titled 'mongosh mongodb://127.0.0.1:27017'. The terminal shows the following text:

```
Microsoft Windows [Version 10.0.26200.7840]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mytre>mongosh
Current Mongosh Log ID: 69a0489dee1ef28cec7c2906
Connecting to:  mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Using MongoDB:  8.2.5
Using Mongosh:  2.7.0

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

-----
The server generated these startup warnings when booting
2026-02-26T18:22:42.047+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> use vehicles
switched to db vehicles
vehicles>
```

The terminal window is set against a dark background. At the bottom, the Windows taskbar is visible, showing the date and time as 18:51 on 26-02-2026, along with system icons for network, volume, and battery.

- Write a MongoDB query to display all the databases.

```
mongosh mongodb://127.0.0.1:27017/
Microsoft Windows [Version 10.0.26200.7840]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mytre>mongosh
Current Mongosh Log ID: 69a0489deef28cec7c2906
Connecting to:  mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Using MongoDB:  8.2.5
Using Mongosh:  2.7.0

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

-----
The server generated these startup warnings when booting
2026-02-26T18:22:42.047+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> use vehicles
switched to db vehicles
vehicles> show dbs
admin      40.00 KiB
config     92.00 KiB
local      40.00 KiB
vehicles> |
```

- Create a collection called 'two_wheelers'. (use capping) and Create a collection called 'four_wheelers'.

```
mongosh mongodb://127.0.0.1:27017/
Microsoft Windows [Version 10.0.26200.7840]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mytre>mongosh
Current Mongosh Log ID: 69a0489deef28cec7c2906
Connecting to:  mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Using MongoDB:  8.2.5
Using Mongosh:  2.7.0

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

-----
The server generated these startup warnings when booting
2026-02-26T18:22:42.047+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> use vehicles
switched to db vehicles
vehicles> show dbs
admin      40.00 KiB
config     92.00 KiB
local      40.00 KiB
vehicles> db.createCollection("two_wheelers", { capped: true, size: 500000 })
{ ok: 1 }
vehicles> db.createCollection("four_wheelers")
{ ok: 1 }
vehicles> |
```

- Add 5 two-wheeler details to the collection named 'two_wheelers'. Each document consists of following fields as bike_name, model (gear or gearless), category (100cc, 125cc, 150cc, 200cc), colors_available (red, black, blue, sport red etc) as array,

manufacturer, performance (out of 10), timestamp (date and year release) and price.

```
mongosh mongodb://127.0.0.1
vehicles> db.two_wheelers.insertMany([
  {
    bike_name: "Honda Shine",
    model: "gear",
    category: "115cc",
    colors_available: ["red", "black", "blue"],
    manufacturer: "Honda",
    performance: 8,
    timestamp: new Date("2020-05-10"),
    price: 85000
  },
  {
    bike_name: "TVS Jupiter",
    model: "gearless",
    category: "115cc",
    colors_available: ["white", "black"],
    manufacturer: "TVS",
    performance: 7,
    timestamp: new Date("2019-03-15"),
    price: 75000
  },
  {
    bike_name: "Yamaha R15",
    model: "gear",
    category: "155cc",
    colors_available: ["sport red", "blue"],
    manufacturer: "Yamaha",
    performance: 9,
    timestamp: new Date("2021-07-20"),
    price: 150000
  },
  {
    bike_name: "Bajaj Pulsar",
    model: "gear",
    category: "160cc",
    colors_available: ["black", "red"],
    manufacturer: "Bajaj",
    performance: 8,
    timestamp: new Date("2018-06-11"),
    price: 120000
  },
  {
    bike_name: "Hero Splendor",
    model: "gear",
    category: "105cc",
    colors_available: ["black", "blue"],
    manufacturer: "Hero",
    performance: 6,
    timestamp: new Date("2017-02-01"),
    price: 70000
  }
])
acknowledged: true,
insertedIds: {
  0: ObjectId('60a0938ae1f28ec7c2987'),
  1: ObjectId('60a0938ae1f28ec7c2988'),
  2: ObjectId('60a0938ae1f28ec7c2989'),
  3: ObjectId('60a0938ae1f28ec7c298a'),
  4: ObjectId('60a0938ae1f28ec7c298b')
}
vehicles> |
```

- Add 5 four-wheeler details to the collection named 'four_wheelers'. Each document consists of following fields as vehicle_name, model (commercial or own), category (car, lorry, bus, mini truck, heavy truck, containers), variants (vxi, zxi, petrol, diesel etc) as array, manufacturer, performance (out of 10), timestamp (date and year release) and price.

```
mongosh mongodb://127.0.0.1
vehicles> db.four_wheelers.insertMany([
  {
    vehicle_name: "Swift",
    model: "omni",
    category: "car",
    variants: ["vdi", "xdi", "petrol"],
    manufacturer: "Maruti",
    performance: 8,
    timestamp: new Date("2020-01-10"),
    price: 700000
  },
  {
    vehicle_name: "Innova",
    model: "omni",
    category: "car",
    variants: ["diesel", "xdi"],
    manufacturer: "Toyota",
    performance: 9,
    timestamp: new Date("2019-04-20"),
    price: 1500000
  },
  {
    vehicle_name: "Tata Ace",
    model: "commercial",
    category: "mini truck",
    variants: ["diesel"],
    manufacturer: "Tata",
    performance: 7,
    timestamp: new Date("2018-08-12"),
    price: 500000
  },
  {
    vehicle_name: "Ashok Leyland Bus",
    model: "commercial",
    category: "bus",
    variants: ["diesel"],
    manufacturer: "Ashok Leyland",
    performance: 8,
    timestamp: new Date("2016-09-05"),
    price: 2500000
  },
  {
    vehicle_name: "Eicher Truck",
    model: "commercial",
    category: "heavy truck",
    variants: ["diesel"],
    manufacturer: "Eicher",
    performance: 6,
    timestamp: new Date("2015-12-18"),
    price: 3000000
  }
])
acknowledged: true,
insertedIds: {
  0: ObjectId('69a04971ee1f28cec7c298e'),
  1: ObjectId('69a04971ee1f28cec7c298d'),
  2: ObjectId('69a04971ee1f28cec7c298f'),
  3: ObjectId('69a04971ee1f28cec7c298c'),
  4: ObjectId('69a04971ee1f28cec7c2910')
}
vehicles> |
```

- Write a MongoDB query to display all documents available in two_wheelers and four_wheelers.

```
mongosh mongodb://127.0.0.1
vehicles> db.two_wheelers.find()
{
  '_id': ObjectId('69a04971ee1f28cec7c298e'),
  'vehicle_name': 'Swift',
  'model': 'omni',
  'category': 'car',
  'variants': [ 'vdi', 'xdi', 'petrol' ],
  'manufacturer': 'Maruti',
  'performance': 8,
  'timestamp': ISODate('2020-01-10T00:00:00.000Z'),
  'price': 700000
},
{
  '_id': ObjectId('69a04971ee1f28cec7c298d'),
  'vehicle_name': 'Innova',
  'model': 'omni',
  'category': 'car',
  'variants': [ 'diesel', 'xdi' ],
  'manufacturer': 'Toyota',
  'performance': 9,
  'timestamp': ISODate('2019-04-20T00:00:00.000Z'),
  'price': 1500000
},
{
  '_id': ObjectId('69a04971ee1f28cec7c298f'),
  'vehicle_name': 'Tata Ace',
  'model': 'commercial',
  'category': 'mini truck',
  'variants': [ 'diesel' ],
  'manufacturer': 'Tata',
  'performance': 7,
  'timestamp': ISODate('2018-08-12T00:00:00.000Z'),
  'price': 500000
},
{
  '_id': ObjectId('69a04971ee1f28cec7c298c'),
  'vehicle_name': 'Ashok Leyland Bus',
  'model': 'commercial',
  'category': 'bus',
  'variants': [ 'diesel' ],
  'manufacturer': 'Ashok Leyland',
  'performance': 8,
  'timestamp': ISODate('2016-09-05T00:00:00.000Z'),
  'price': 2500000
},
{
  '_id': ObjectId('69a04971ee1f28cec7c2910'),
  'vehicle_name': 'Eicher Truck',
  'model': 'commercial',
  'category': 'heavy truck',
  'variants': [ 'diesel' ],
  'manufacturer': 'Eicher',
  'performance': 6,
  'timestamp': ISODate('2015-12-18T00:00:00.000Z'),
  'price': 3000000
}
vehicles> |
```

- Write a MongoDB query to display only vehicle name and price in all the collection of the database

```
mongosh mongodb://127.0.0.1:27000
performance: 9,
timestamp: ISODate('2019-04-20T00:00:00.000Z'),
price: 1500000
},
{
  _id: ObjectId('69a04971ee1ef28cec7c290e'),
  vehicle_name: 'Tata Ace',
  model: 'commercial',
  category: 'mini truck',
  variants: [ 'diesel' ],
  manufacturer: 'Tata',
  performance: 7,
  timestamp: ISODate('2018-08-12T00:00:00.000Z'),
  price: 500000
},
{
  _id: ObjectId('69a04971ee1ef28cec7c290f'),
  vehicle_name: 'Ashok Leyland Bus',
  model: 'commercial',
  category: 'bus',
  variants: [ 'diesel' ],
  manufacturer: 'Ashok Leyland',
  performance: 8,
  timestamp: ISODate('2016-09-05T00:00:00.000Z'),
  price: 2500000
},
{
  _id: ObjectId('69a04971ee1ef28cec7c2910'),
  vehicle_name: 'Eicher Truck',
  model: 'commercial',
  category: 'heavy truck',
  variants: [ 'diesel' ],
  manufacturer: 'Eicher',
  performance: 6,
  timestamp: ISODate('2015-12-18T00:00:00.000Z'),
  price: 3000000
}
]
vehicles> db.two_wheelers.find({}, {bike_name:1, price:1, _id:0})
[ db.four_wheelers.find({}, {vehicle_name:1, price:1, _id:0})
[
  { vehicle_name: 'Swift', price: 700000 },
  { vehicle_name: 'Innova', price: 1500000 },
  { vehicle_name: 'Tata Ace', price: 500000 },
  { vehicle_name: 'Ashok Leyland Bus', price: 2500000 },
  { vehicle_name: 'Eicher Truck', price: 3000000 }
]
vehicles> |
```

- Write a MongoDB query to display two_wheelers from a particular company

```
mongosh mongodb://127.0.0.1:27000
{
  _id: ObjectId('69a04971ee1ef28cec7c290f'),
  vehicle_name: 'Ashok Leyland Bus',
  model: 'commercial',
  category: 'bus',
  variants: [ 'diesel' ],
  manufacturer: 'Ashok Leyland',
  performance: 8,
  timestamp: ISODate('2016-09-05T00:00:00.000Z'),
  price: 2500000
},
{
  _id: ObjectId('69a04971ee1ef28cec7c2910'),
  vehicle_name: 'Eicher Truck',
  model: 'commercial',
  category: 'heavy truck',
  variants: [ 'diesel' ],
  manufacturer: 'Eicher',
  performance: 6,
  timestamp: ISODate('2015-12-18T00:00:00.000Z'),
  price: 3000000
}
]
vehicles> db.two_wheelers.find({}, {bike_name:1, price:1, _id:0})
[ db.four_wheelers.find({}, {vehicle_name:1, price:1, _id:0})
[
  { vehicle_name: 'Swift', price: 700000 },
  { vehicle_name: 'Innova', price: 1500000 },
  { vehicle_name: 'Tata Ace', price: 500000 },
  { vehicle_name: 'Ashok Leyland Bus', price: 2500000 },
  { vehicle_name: 'Eicher Truck', price: 3000000 }
]
vehicles> db.two_wheelers.find({manufacturer:"Honda"})
[
  {
    _id: ObjectId('69a04938ee1ef28cec7c2907'),
    bike_name: 'Honda Shine',
    model: 'gear',
    category: '125cc',
    colors_available: [ 'red', 'black', 'blue' ],
    manufacturer: 'Honda',
    performance: 8,
    timestamp: ISODate('2020-05-10T00:00:00.000Z'),
    price: 85000
  }
]
vehicles> |
```

- Write a MongoDB query to display four_wheelers available in diesel variants

```
mongosh mongodb://127.0.0.1:27000
vehicles> db.four_wheelers.find({variants:'diesel'})
[
  {
    _id: ObjectId('69a04971ee1ef28cec7c290d'),
    vehicle_name: 'Innova',
    model: 'own',
    category: 'car',
    variants: [ 'diesel', 'zxi' ],
    manufacturer: 'Toyota',
    performance: 9,
    timestamp: ISODate('2019-04-20T00:00:00.000Z'),
    price: 1500000
  },
  {
    _id: ObjectId('69a04971ee1ef28cec7c290e'),
    vehicle_name: 'Tata Ace',
    model: 'commercial',
    category: 'mini truck',
    variants: [ 'diesel' ],
    manufacturer: 'Tata',
    performance: 7,
    timestamp: ISODate('2018-08-12T00:00:00.000Z'),
    price: 500000
  },
  {
    _id: ObjectId('69a04971ee1ef28cec7c290f'),
    vehicle_name: 'Ashok Leyland Bus',
    model: 'commercial',
    category: 'bus',
    variants: [ 'diesel' ],
    manufacturer: 'Ashok Leyland',
    performance: 8,
    timestamp: ISODate('2016-09-05T00:00:00.000Z'),
    price: 2500000
  },
  {
    _id: ObjectId('69a04971ee1ef28cec7c2910'),
    vehicle_name: 'Eicher Truck',
    model: 'commercial',
    category: 'heavy truck',
    variants: [ 'diesel' ],
    manufacturer: 'Eicher',
    performance: 6,
    timestamp: ISODate('2015-12-18T00:00:00.000Z'),
    price: 3000000
  }
]
vehicles>
```

- Write a MongoDB query to display vehicles name, category and manufacturer details whose rating is more than 5.

```
mongosh mongodb://127.0.0.1:27000
category: 'bus',
variants: [ 'diesel' ],
manufacturer: 'Ashok Leyland',
performance: 8,
timestamp: ISODate('2016-09-05T00:00:00.000Z'),
price: 2500000
},
{
  _id: ObjectId('69a04971ee1ef28cec7c2910'),
  vehicle_name: 'Eicher Truck',
  model: 'commercial',
  category: 'heavy truck',
  variants: [ 'diesel' ],
  manufacturer: 'Eicher',
  performance: 6,
  timestamp: ISODate('2015-12-18T00:00:00.000Z'),
  price: 3000000
}
]
vehicles> db.two_wheelers.find(
  {performance:{>5}},
  {bike_name:1, category:1, manufacturer:1, _id:0}
)
db.four_wheelers.find(
  {performance:{>5}},
  {vehicle_name:1, category:1, manufacturer:1, _id:0}
)
[
  { vehicle_name: 'Swift', category: 'car', manufacturer: 'Maruti' },
  { vehicle_name: 'Innova', category: 'car', manufacturer: 'Toyota' },
  {
    vehicle_name: 'Tata Ace',
    category: 'mini truck',
    manufacturer: 'Tata'
  },
  {
    vehicle_name: 'Ashok Leyland Bus',
    category: 'bus',
    manufacturer: 'Ashok Leyland'
  },
  {
    vehicle_name: 'Eicher Truck',
    category: 'heavy truck',
    manufacturer: 'Eicher'
  }
]
vehicles>
```