

## VIT-AP UNIVERSITY, ANDHRA PRADESH

### Lab Sheet 6: MongoDB Basic commands

**Branch/ Class:** B.Tech/M.Tech

**Faculty Name:** Prof. S.Gopikrishnan

**Student name:** G Sai Siva Mani

**Date:** 26-02-2026

**School:** SCOPE

**Reg. no.:** 23BCE9808

1. Use MongoDB to implement the following DB operations

1. Create a database called 'vehicles' and *write* a MongoDB query to select database as "vehicles".

Code:

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists databases: admin, config, local, test, and vehicles. Under 'vehicles', there are two collections: four\_wheelers and two\_wHEELERS. The main panel displays the MongoDB shell. The command > use vehicles has been run, and the response < switched to db vehicles > is shown. The title bar of the application window reads 'MongoDB Compass - localhost:27017/Shell'.

2. Write a MongoDB query to display all the databases.

Code:

3. Create a collection called 'two\_wHEELERS'. (use capping) and Create a collection called 'four\_wHEELERS'.

Code:

4. 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.

Code:

```
MongoDB Compass - localhost:27017/Shell
Connections Edit View Help
Compass
My Queries
CONNECTIONS (1)
localhost:27017
+ ...
Search connections
admin
config
local
test
vehicles
four_wheelers
two_wheelers
>_ mongosh: localhost:27017 >_ mongosh: localhost:27017 +
>_MONGOSH
> db.two_wheelers.insertMany([
  {
    bike_name: "Hero Honda",
    model: "gear",
    category: "125cc",
    colors_available: ["red", "black", "blue"],
    manufacturer: "Honda",
    performance: 8,
    timestamp: new Date("2015-05-12"),
    price: 112000
  },
  {
    bike_name: "TVS Jupiter",
    model: "gearless",
    category: "125cc",
    colors_available: ["white", "black", "gray"],
    manufacturer: "TVS",
    performance: 7,
    timestamp: new Date("2017-03-18"),
    price: 110000
  },
  {
    bike_name: "Pulsar",
    model: "gear",
    category: "100cc",
    colors_available: ["red", "black", "blue"],
    manufacturer: "Bajaj",
    performance: 8,
    timestamp: new Date("2018-07-20"),
    price: 98000
  }
])
< {
  acknowledged: true,
  insertedIds: [
    '_0': ObjectId('699fc945615ac525cbcb8db4'),
    '_1': ObjectId('699fc945615ac525cbcb8db5'),
    '_2': ObjectId('699fc945615ac525cbcb8db6')
  ]
}
ENG IN 10:15 26-02-2026
```

```
MongoDB Compass - localhost:27017/Shell
Connections Edit View Help
Compass
My Queries
CONNECTIONS (1)
localhost:27017
+ ...
Search connections
admin
config
local
test
vehicles
four_wheelers
two_wheelers
>_ mongosh: localhost:27017 >_ mongosh: localhost:27017 +
>_MONGOSH
> db.two_wheelers.insertMany([
  {
    price: 98000
  },
  {
    bike_name: "Yamaha R15",
    model: "gear",
    category: "200cc",
    colors_available: ["red", "blue"],
    manufacturer: "Yamaha",
    performance: 9,
    timestamp: new Date("2021-01-10"),
    price: 175000
  },
  {
    bike_name: "shine",
    model: "gear",
    category: "100cc",
    colors_available: ["black", "grey"],
    manufacturer: "Honda",
    performance: 9,
    timestamp: new Date("2015-08-15"),
    price: 112000
  }
])
< {
  acknowledged: true,
  insertedIds: [
    '_0': ObjectId('699fc945615ac525cbcb8db4'),
    '_1': ObjectId('699fc945615ac525cbcb8db5'),
    '_2': ObjectId('699fc945615ac525cbcb8db6')
  ]
}
ENG IN 10:16 26-02-2026
```

```

>_MONGOOSH
  model: "gear",
  category: "200cc",
  colors_available: ["red", "blue"],
  manufacturer: "Yamaha",
  performance: 9,
  timestamp: new Date("2021-01-10"),
  price: 17500
},
{
  bike_name: "shine",
  model: "gear",
  category: "100cc",
  colors_available: ["black", "grey"],
  manufacturer: "Honda",
  performance: 9,
  timestamp: new Date("2015-08-15"),
  price: 112000
}
)
< {
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('699fc945615ac525cbcb3db4'),
    '1': ObjectId('699fc945615ac525cbcb3db5'),
    '2': ObjectId('699fc945615ac525cbcb3db6'),
    '3': ObjectId('699fc945615ac525cbcb3db7'),
    '4': ObjectId('699fc945615ac525cbcb3db8')
  ]
}

```

5. 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.

Code:

Code:

```

>_MONGOOSH
> db.four_wHEELERS.insertMany([
  {
    vehicle_name: "Maruti Swift",
    model: "own",
    category: "car",
    variants: ["petrol", "diesel"],
    manufacturer: "Maruti",
    performance: 9,
    timestamp: new Date("2017-02-14"),
    price: 780000
  },
  {
    vehicle_name: "Tata Ace",
    model: "commercial",
    category: "mini truck",
    variants: ["diesel"],
    manufacturer: "Tata",
    performance: 9,
    timestamp: new Date("2018-06-10"),
    price: 690000
  },
  {
    vehicle_name: "Ashok Leyland",
    model: "commercial",
    category: "heavy truck",
    variants: ["diesel"],
    manufacturer: "Ashok Leyland",
    performance: 8,
    timestamp: new Date("2014-10-01"),
    price: 800000
  }
])

```

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

Search connections

localhost:27017

- admin
- config
- local
- test
- vehicles
  - four\_wheelers
  - two\_wheelers

\_MONGOSH

```
>_ price: 2500000
  },
  {
    vehicle_name: "Volvo Bus",
    model: "commercial",
    category: "bus",
    variants: ["diesel"],
    manufacturer: "Volvo",
    performance: 9,
    timestamp: new Date("2021-11-21"),
    price: 8500000
  },
  {
    vehicle_name: "Mahindra tractor",
    model: "own",
    category: "tractor",
    variants: ["diesel"],
    manufacturer: "Mahindra",
    performance: 7,
    timestamp: new Date("2016-04-25"),
    price: 550000
  }
]
< {
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('699fcaa8615ac525cbbc8db9'),
    '1': ObjectId('699fcaa8615ac525cbbc8db8'),
    '2': ObjectId('699fcaa8615ac525cbbc8db1'),
    '3': ObjectId('699fcaa8615ac525cbbc8dbc'),
    '4': ObjectId('699fcaa8615ac525cbbc8dbd')
  ]
}
```

ENG IN 10:23 26-02-2026

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

Search connections

localhost:27017

- admin
- config
- local
- test
- vehicles
  - four\_wheelers
  - two\_wheelers

\_MONOOSH

```
>_ model: "commercial",
  category: "bus",
  variants: ["diesel"],
  manufacturer: "Volvo",
  performance: 9,
  timestamp: new Date("2021-11-21"),
  price: 8500000
},
{
  vehicle_name: "Mahindra tractor",
  model: "own",
  category: "tractor",
  variants: ["diesel"],
  manufacturer: "Mahindra",
  performance: 7,
  timestamp: new Date("2016-04-25"),
  price: 550000
}
]
< {
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('699fcaa8615ac525cbbc8db9'),
    '1': ObjectId('699fcaa8615ac525cbbc8db8'),
    '2': ObjectId('699fcaa8615ac525cbbc8db1'),
    '3': ObjectId('699fcaa8615ac525cbbc8dbc'),
    '4': ObjectId('699fcaa8615ac525cbbc8dbd')
  ]
}
```

ENG IN 10:23 26-02-2026

6. Write a MongoDB query to display all documents available in two\_wheelers and four\_wheelers.

Code:

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS ()

localhost:27017

- admin
- config
- local
- test
- vehicles
  - four\_wHEELERS
  - two\_wHEELERS

Search connections

\_MONGOSH

```
>_ db.four_wheelers.find()
< [
  {
    _id: ObjectId('699fcfaa8615ac525cbbc8db9'),
    vehicle_name: 'Maruti Swift',
    model: 'own',
    category: 'car',
    variants: [
      'petrol',
      'diesel'
    ],
    manufacturer: 'Maruti',
    performance: 9,
    timestamp: 2017-02-14T00:00:00.000Z,
    price: 780000
  },
  {
    _id: ObjectId('699fcfaa8615ac525cbbc8dba'),
    vehicle_name: 'Tata Ace',
    model: 'commercial',
    category: 'mini truck',
    variants: [
      'diesel'
    ],
    manufacturer: 'Tata',
    performance: 9,
    timestamp: 2018-06-10T00:00:00.000Z,
    price: 690000
  }
]
```

10:24  
ENG IN 26-02-2026

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS ()

localhost:27017

- admin
- config
- local
- test
- vehicles
  - four\_wHEELERS
  - two\_wHEELERS

Search connections

\_MONGOSH

```
>_ db.vehicles.find()
< [
  {
    _id: ObjectId('699fcfaa8615ac525cbbc8dbb'),
    vehicle_name: 'Ashok Leyland',
    model: 'commercial',
    category: 'heavy truck',
    variants: [
      'diesel'
    ],
    manufacturer: 'Ashok Leyland',
    performance: 8,
    timestamp: 2014-10-01T00:00:00.000Z,
    price: 2500000
  },
  {
    _id: ObjectId('699fcfaa8615ac525cbbc8dbc'),
    vehicle_name: 'Volvo Bus',
    model: 'commercial',
    category: 'bus',
    variants: [
      'diesel'
    ],
    manufacturer: 'Volvo',
    performance: 9,
    timestamp: 2021-11-21T00:00:00.000Z,
    price: 8500000
  },
  {
    _id: ObjectId('699fcfaa8615ac525cbbc8dbd'),
    vehicle_name: 'Mahindra tractor',
    model: 'commercial',
    category: 'tractor',
    variants: [
      'diesel'
    ],
    manufacturer: 'Mahindra',
    performance: 10,
    timestamp: 2022-01-15T00:00:00.000Z,
    price: 12000000
  }
]
```

10:24  
ENG IN 26-02-2026

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

**Compass**

My Queries

CONNECTIONS (1)

localhost:27017

Search connections

\_MONGOSH

```
>_ mongo:localhost:27017 >_ mongo:localhost:27017 +  
>_ MONGOSH  
model: 'own',  
category: 'tractor',  
variants: [  
    'diesel'  
,  
    manufacturer: 'Mahindra',  
    performance: 7,  
    timestamp: 2016-04-25T00:00:00.000Z,  
    price: 550000  
,  
> db.two_wheelers.find()  
< {  
    _id: ObjectId('699fc945615ac525cbbc8db4'),  
    bike_name: 'Hero Honda',  
    model: 'gear',  
    category: '125cc',  
    colors_available: [  
        'red',  
        'black',  
        'blue'  
,  
        manufacturer: 'Honda',  
        performance: 8,  
        timestamp: 2015-05-12T00:00:00.000Z,  
        price: 112000  
,  
{  
    _id: ObjectId('699fc945615ac525cbbc8db5'),  
    bike_name: 'TVS Jupiter',
```

2 2026 10:25 ENG IN

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

**Compass**

My Queries

CONNECTIONS (1)

localhost:27017

Search connections

\_MONGOSH

```
>_ mongo:localhost:27017 >_ mongo:localhost:27017 +  
>_ MONGOSH  
model: 'gearless',  
category: '125cc',  
colors_available: [  
    'white',  
    'black',  
    'gray'  
,  
    manufacturer: 'TVS',  
    performance: 7,  
    timestamp: 2017-03-18T00:00:00.000Z,  
    price: 110000  
,  
{  
    _id: ObjectId('699fc945615ac525cbbc8db6'),  
    bike_name: 'Pulsar',  
    model: 'gear',  
    category: '180cc',  
    colors_available: [  
        'red',  
        'black',  
        'blue'  
,  
        manufacturer: 'Bajaj',  
        performance: 8,  
        timestamp: 2018-07-20T00:00:00.000Z,  
        price: 98000  
,  
{  
    _id: ObjectId('699fc945615ac525cbbc8db7'),
```

2 2026 10:25 ENG IN

```

>_ mongo: localhost:27017 >_ mongosh: localhost:27017 + 
>_ MONGOSH
{
  "_id": ObjectId("699fc945615ac525cbbc8db7"),
  "bike_name": "Yamaha R15",
  "model": "gear",
  "category": "200cc",
  "colors_available": [
    "red",
    "blue"
  ],
  "manufacturer": "Yamaha",
  "performance": 9,
  "timestamp": 2021-01-10T00:00:00.000Z,
  "price": 175000
}
{
  "_id": ObjectId("699fc945615ac525cbbc8db8"),
  "bike_name": "shine",
  "model": "gear",
  "category": "100cc",
  "colors_available": [
    "black",
    "grey"
  ],
  "manufacturer": "Honda",
  "performance": 9,
  "timestamp": 2015-08-15T00:00:00.000Z,
  "price": 112000
}

```

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

Code:

```

>_ mongo: localhost:27017 >_ mongosh: localhost:27017 + 
>_ MONGOSH
{
  "bike_name": "Yamaha R15",
  "price": 175000
}
{
  "bike_name": "shine",
  "price": 112000
}
> db.four_wheelers.find({}, {vehicle_name: 1, price: 1, _id: 0})
< [
  {
    "vehicle_name": "Maruti Swift",
    "price": 780000
  },
  {
    "vehicle_name": "Tata Ace",
    "price": 690000
  },
  {
    "vehicle_name": "Ashok Leyland",
    "price": 2500000
  },
  {
    "vehicle_name": "Volvo Bus",
    "price": 8500000
  },
  {
    "vehicle_name": "Mahindra tractor",
    "price": 550000
  }
]

```

8. Write a MongoDB query to display two\_wheelers from a particular company

Code:

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

**Compass**

My Queries

CONNECTIONS (1)

localhost:27017

admin config local test vehicles

four\_wheelers two\_wheelers

>\_MONGOSH

```
>_MONGOSH
  'black',
  'grey'
],
manufacturer: 'Honda',
performance: 9,
timestamp: 2015-08-15T00:00:00.000Z,
price: 112000
}
)
> db.two_wheelers.find({}, {bike_name:1, price:1, _id:0})
< [
  {
    bike_name: 'Hero Honda',
    price: 112000
  },
  {
    bike_name: 'TVS Jupiter',
    price: 110000
  },
  {
    bike_name: 'Pulsar',
    price: 98000
  },
  {
    bike_name: 'Yamaha R15',
    price: 175000
  },
  {
    bike_name: 'shine',
    price: 112000
  }
]
```

Search connections

10:27 26-02-2026 ENG IN

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

**Compass**

My Queries

CONNECTIONS (1)

localhost:27017

admin config local test vehicles

four\_wheelers two\_wheelers

>\_MONGOSH

```
>_MONGOSH
{
  bike_name: 'Yamaha R15',
  price: 175000
}
{
  bike_name: 'shine',
  price: 112000
}
> db.four_wheelers.find({}, {vehicle_name:1, price:1, _id:0})
< [
  {
    vehicle_name: 'Maruti Swift',
    price: 780000
  },
  {
    vehicle_name: 'Tata Ace',
    price: 690000
  },
  {
    vehicle_name: 'Ashok Leyland',
    price: 2500000
  },
  {
    vehicle_name: 'Volvo Bus',
    price: 8500000
  },
  {
    vehicle_name: 'Mahindra tractor',
    price: 550000
  }
]
```

Search connections

10:28 26-02-2026 ENG IN

9. Write a MongoDB query to display four\_wheelers available in diesel variants

Code:

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

**Compass**

My Queries

CONNECTIONS (1)

localhost:27017

Search connections

admin config local test vehicles

four\_wHEELERS two\_wHEELERS

>\_MONGOSH

```
> db.four_wheelers.find({ variants: "diesel" })
```

```
< [
```

```
  {
```

```
    _id: ObjectId('699fcaa8615ac525cbcc8db9'),
```

```
    vehicle_name: 'Maruti Swift',
```

```
    model: 'own',
```

```
    category: 'car',
```

```
    variants: [
```

```
      'petrol',
```

```
      'diesel'
```

```
    ],
```

```
    manufacturer: 'Maruti',
```

```
    performance: 9,
```

```
    timestamp: 2017-02-14T00:00:00.000Z,
```

```
    price: 780000
```

```
  }
```

```
  {
```

```
    _id: ObjectId('699fcaa8615ac525cbcc8dba'),
```

```
    vehicle_name: 'Tata Ace',
```

```
    model: 'commercial',
```

```
    category: 'mini truck',
```

```
    variants: [
```

```
      'diesel'
```

```
    ],
```

```
    manufacturer: 'Tata',
```

```
    performance: 9,
```

```
    timestamp: 2018-06-10T00:00:00.000Z,
```

```
    price: 690000
```

```
  }
```

```
]
```

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

**Compass**

My Queries

CONNECTIONS (1)

localhost:27017

Search connections

admin config local test vehicles

four\_wHEELERS two\_wHEELERS

>\_MONGOSH

```
< [
```

```
  {
```

```
    _id: ObjectId('699fcaa8615ac525cbcc8dbb'),
```

```
    vehicle_name: 'Ashok Leyland',
```

```
    model: 'commercial',
```

```
    category: 'heavy truck',
```

```
    variants: [
```

```
      'diesel'
```

```
    ],
```

```
    manufacturer: 'Ashok Leyland',
```

```
    performance: 8,
```

```
    timestamp: 2014-10-01T00:00:00.000Z,
```

```
    price: 2500000
```

```
  }
```

```
  {
```

```
    _id: ObjectId('699fcaa8615ac525cbcc8dbc'),
```

```
    vehicle_name: 'Volvo Bus',
```

```
    model: 'commercial',
```

```
    category: 'bus',
```

```
    variants: [
```

```
      'diesel'
```

```
    ],
```

```
    manufacturer: 'Volvo',
```

```
    performance: 9,
```

```
    timestamp: 2021-11-21T00:00:00.000Z,
```

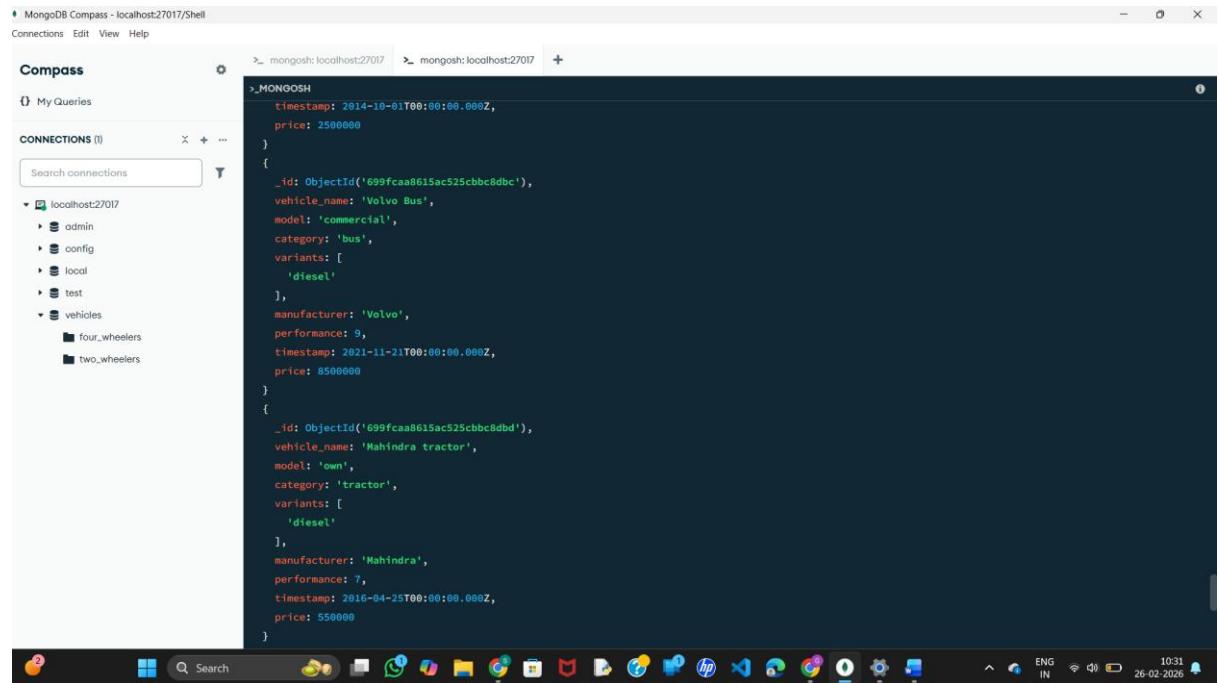
```
    price: 8500000
```

```
  }
```

```
  {
```

```
    _id: ObjectId('699fcaa8615ac525cbcc8dbd'),
```

```
    vehicle_name: 'Mahindra tractor',
```



The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'localhost:27017' with databases 'admin', 'config', 'local', 'test', and 'vehicles'. Under 'vehicles', there are two sub-folders: 'four\_wHEELERS' and 'two\_wHEELERS'. The main pane displays a query result for the 'MONGOSH' database, specifically from the 'two\_wHEELERS' collection. The results show three documents representing different vehicle models:

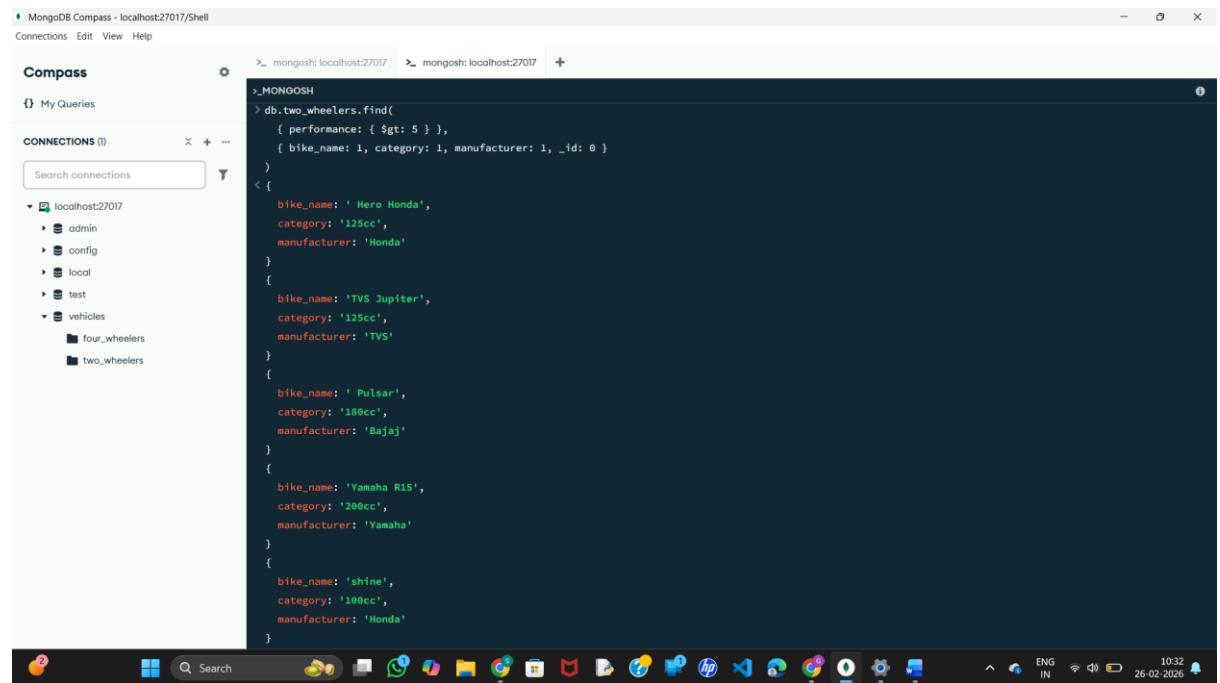
```

>_MONGOSH
{
  timestamp: 2014-10-01T00:00:00.000Z,
  price: 250000
}
{
  _id: ObjectId('699fcaa8615ac525cbbc8dbc'),
  vehicle_name: 'Volvo Bus',
  model: 'commercial',
  category: 'bus',
  variants: [
    'diesel'
  ],
  manufacturer: 'Volvo',
  performance: 9,
  timestamp: 2021-11-21T00:00:00.000Z,
  price: 850000
}
{
  _id: ObjectId('699fcaa8615ac525cbbc8dbd'),
  vehicle_name: 'Mahindra tractor',
  model: 'own',
  category: 'tractor',
  variants: [
    'diesel'
  ],
  manufacturer: 'Mahindra',
  performance: 7,
  timestamp: 2016-04-25T00:00:00.000Z,
  price: 550000
}

```

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

Code:



The screenshot shows the MongoDB Compass interface. The 'two\_wHEELERS' collection is selected. A query is run to find documents where the 'performance' field is greater than 5, including fields 'bike\_name', 'category', 'manufacturer', and '\_id':

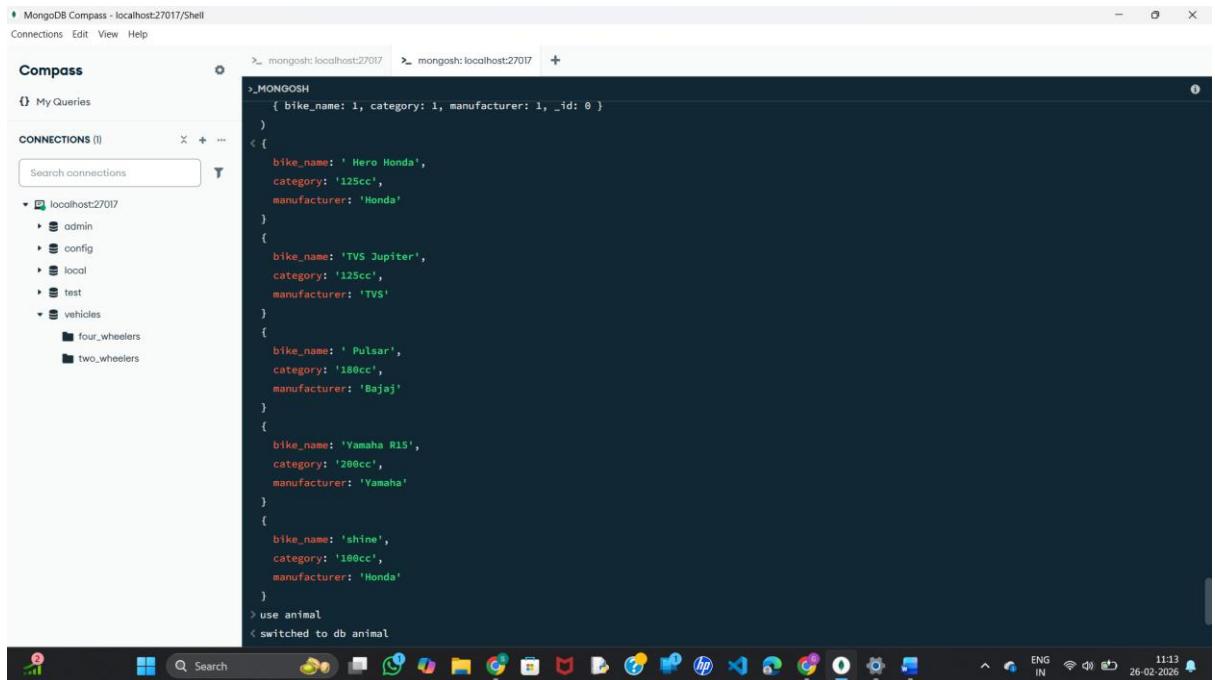
```

>_MONGOSH
> db.two_wHEELERS.find(
  { performance: { $gt: 5 } },
  { bike_name: 1, category: 1, manufacturer: 1, _id: 0 }
)
< [
  {
    bike_name: 'Hero Honda',
    category: '125cc',
    manufacturer: 'Honda'
  },
  {
    bike_name: 'TVS Jupiter',
    category: '125cc',
    manufacturer: 'TVS'
  },
  {
    bike_name: 'Pulsar',
    category: '180cc',
    manufacturer: 'Bajaj'
  },
  {
    bike_name: 'Yamaha R15',
    category: '200cc',
    manufacturer: 'Yamaha'
  },
  {
    bike_name: 'shine',
    category: '100cc',
    manufacturer: 'Honda'
  }
]

```

2. Use MongoDB to implement the following DB operations for a Zoo

1. Create a database called 'animal' and write a MongoDB query to select database as 'animal'.



MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

localhost:27017

- admin
- config
- local
- test
- vehicles
  - four\_wheelers
  - two\_wHEELERS

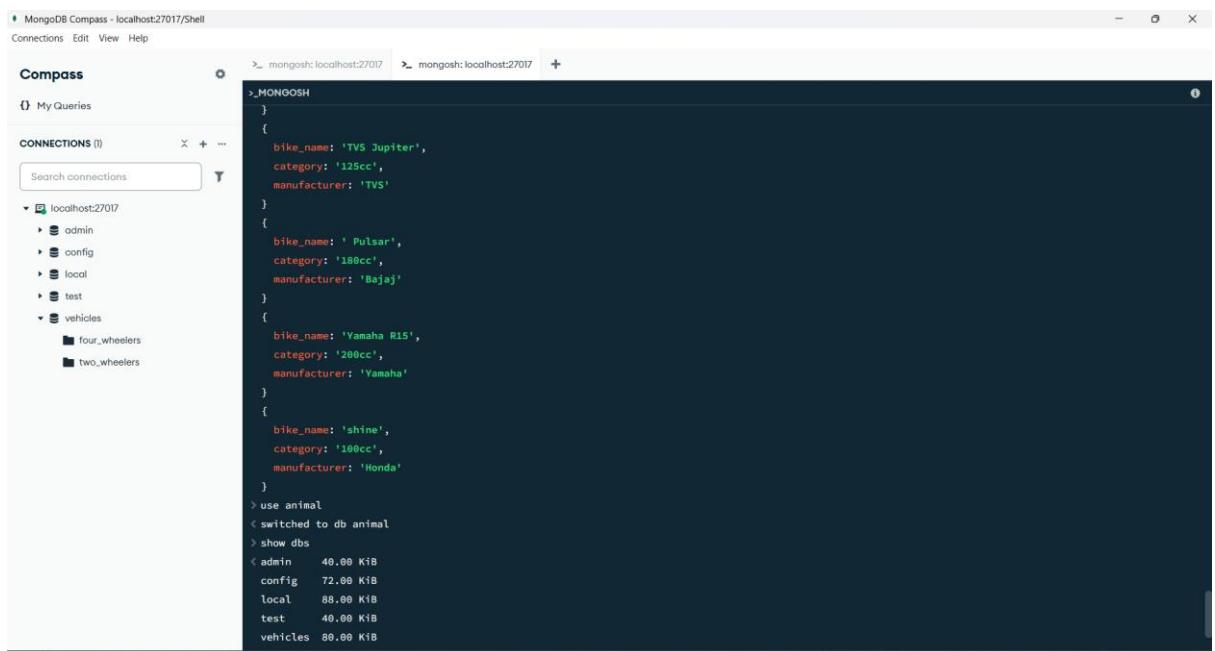
>\_MONGOSH

```
>_MONGOSH
{
  <!
    bike_name: 'Hero Honda',
    category: '125cc',
    manufacturer: 'Honda'
  >
  {
    bike_name: 'TVS Jupiter',
    category: '125cc',
    manufacturer: 'TVS'
  }
  {
    bike_name: 'Pulsar',
    category: '180cc',
    manufacturer: 'Bajaj'
  }
  {
    bike_name: 'Yamaha R15',
    category: '200cc',
    manufacturer: 'Yamaha'
  }
  {
    bike_name: 'shine',
    category: '100cc',
    manufacturer: 'Honda'
  }
> use animal
< switched to db animal
```

Search connections

11:13 26-02-2026 ENG IN

## 2. Write a MongoDB query to display all the databases.



MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

localhost:27017

- admin
- config
- local
- test
- vehicles
  - four\_wHEELERS
  - two\_wHEELERS

>\_MONGOSH

```
>_MONGOSH
{
  <!
    bike_name: 'TVS Jupiter',
    category: '125cc',
    manufacturer: 'TVS'
  >
  {
    bike_name: 'Pulsar',
    category: '180cc',
    manufacturer: 'Bajaj'
  }
  {
    bike_name: 'Yamaha R15',
    category: '200cc',
    manufacturer: 'Yamaha'
  }
  {
    bike_name: 'shine',
    category: '100cc',
    manufacturer: 'Honda'
  }
> use animal
< switched to db animal
> show dbs
< admin   40.00 KiB
 config  72.00 KiB
 local   88.00 KiB
 test   40.00 KiB
 vehicles 80.00 KiB
```

Search connections

## 3. Create a collection called 'wild\_animals'.(use capping) and Create a collection called 'domestic\_animals'.

The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'localhost:27017' with databases 'admin', 'config', 'local', 'test', and 'vehicles'. Under 'vehicles', there are sub-folders 'four\_wHEELERS' and 'two\_wHEELERS'. The main panel displays the mongo shell command history:

```

>_MONGOSH
{
  bike_name: 'shine',
  category: '100cc',
  manufacturer: 'Honda'
}
> use animal
< switched to db animal
> show dbs
< admin   40.00 KiB
config   72.00 KiB
local    88.00 KiB
test     40.00 KiB
vehicles 80.00 KiB
> show dbs
< admin   40.00 KiB
animals  8.00 KiB
config   72.00 KiB
local    88.00 KiB
test     40.00 KiB
vehicles 80.00 KiB
> db.createCollection("wild_animals", {
  capped: true,
  size: 100000,
  max: 100
})
< { ok: 1 }
> db.createCollection("domestic_animals")
< { ok: 1 }
animal>

```

The status bar at the bottom shows 'ENG IN' and the date '26-02-2026'.

- Add 5 wild\_animal details to the collection named 'wild\_animals'. Each document consists of following fields as animal\_name, nature (harm or harmless), favorite\_foods (meat, rabbits, deer etc) as array, care\_taker\_name, life span (in years), timestamp (when the animal registered at the Zoo) and expenses.

Code:

The screenshot shows the mongo shell command history for inserting documents into the 'wild\_animals' collection:

```

>_MONGOOSH
> db.wild_animals.insertMany([
  {
    animal_name: "Lion",
    nature: "harm",
    favorite_foods: ["meat", "deer"],
    care_taker_name: "Siva",
    life_span: 15,
    timestamp: new Date("2017-01-15"),
    expenses: 55000
  },
  {
    animal_name: "Tiger",
    nature: "harm",
    favorite_foods: ["meat", "rabbits"],
    care_taker_name: "Kanth",
    life_span: 14,
    timestamp: new Date("2018-06-20"),
    expenses: 60000
  },
  {
    animal_name: "Elephant",
    nature: "harmless",
    favorite_foods: ["grass", "fruits"],
    care_taker_name: "Mahesh",
    life_span: 65,
    timestamp: new Date("2004-03-10"),
    expenses: 90000
  },
  {
    animal_name: "Giraffe",
    nature: "harmless",
    favorite_foods: ["leaves", "fruits"],
    care_taker_name: "Rajesh",
    life_span: 45,
    timestamp: new Date("2019-09-20"),
    expenses: 75000
  }
])

```

The status bar at the bottom shows 'ENG IN' and the date '26-02-2026'.

```

>_MONGOSH
{
  "animal_name": "Bear",
  "nature": "harm",
  "favorite_foods": ["fish", "honey"],
  "care_taker_name": "prashant",
  "life_span": 25,
  "timestamp": new Date("2016-11-05"),
  "expenses": 50000
},
{
  "animal_name": "Giraffe",
  "nature": "harmless",
  "favorite_foods": ["leaves"],
  "care_taker_name": "Kiran",
  "life_span": 26,
  "timestamp": new Date("2009-08-12"),
  "expenses": 45000
}
< {
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('699fe128615ac525cbbc8dbe'),
    '1': ObjectId('699fe128615ac525cbbc8dbf'),
    '2': ObjectId('699fe128615ac525cbbc8dc0'),
    '3': ObjectId('699fe128615ac525cbbc8dc1'),
    '4': ObjectId('699fe128615ac525cbbc8dc2')
  ]
}

```

5. Add 5 domestic-animal details to the collection named 'domestic\_animals'. Each document consists of following fields as animal\_name, gender (male or female), favorite\_foods (meat, rabbits, deer etc) as array, animal\_petname, life span (in years), timestamp (when the animal registered at the Zoo) and expenses.

Code:

```

>_MONGOSH
> db.domestic_animals.insertMany([
  {
    animal_name: "Dog",
    gender: "female",
    favorite_foods: ["meat", "rice"],
    animal_petname: "puppy",
    life_span: 15,
    timestamp: new Date("2019-02-10"),
    expenses: 12000
  },
  {
    animal_name: "Cat",
    gender: "female",
    favorite_foods: ["fish", "milk"],
    animal_petname: "Kitty",
    life_span: 15,
    timestamp: new Date("2018-07-18"),
    expenses: 8000
  },
  {
    animal_name: "Cow",
    gender: "female",
    favorite_foods: ["grass", "hay"],
    animal_petname: "lakshmi",
    life_span: 25,
    timestamp: new Date("2019-03-25"),
    expenses: 15000
  },
  {
    animal_name: "Lion",
    gender: "male",
    favorite_foods: ["meat", "bones"],
    animal_petname: "king",
    life_span: 20,
    timestamp: new Date("2018-01-01"),
    expenses: 20000
  }
])

```

The screenshot shows the MongoDB Compass interface. On the left, the 'Compass' sidebar displays 'My Queries' and a 'CONNECTIONS' section with a dropdown menu and a search bar. Below these are connection entries for 'localhost:27017': 'admin', 'config', 'local', 'test', and 'vehicles'. Under 'vehicles', there are two sub-folders: 'four\_wheelers' and 'two\_wHEELERS'. The main area is a terminal window titled '\_MONGOSH' showing the following code:

```
_MONGOSH
{
  animal_name: "Goat",
  gender: "male",
  favorite_foods: ["leaves", "grass"],
  animal_petsname: "puri",
  life_span: 12,
  timestamp: new Date("2022-01-05"),
  expenses: 7000
},
{
  animal_name: "Horse",
  gender: "male",
  favorite_foods: ["grass", "grains"],
  animal_petsname: "badsha",
  life_span: 25,
  timestamp: new Date("2016-09-14"),
  expenses: 33000
}
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fe489615ac525cbcc8dc3'),
    '1': ObjectId('699fe489615ac525cbcc8dc4'),
    '2': ObjectId('699fe489615ac525cbcc8dc5'),
    '3': ObjectId('699fe489615ac525cbcc8dc6'),
    '4': ObjectId('699fe489615ac525cbcc8dc7')
  }
}
```

6. Write a MongoDB query to display all documents available in wild\_animals and domestic\_animals.

Code:

The screenshot shows the MongoDB Compass interface. The 'Compass' sidebar and 'CONNECTIONS' section are identical to the previous screenshot. The main terminal window is titled '\_MONGOSH' and contains the following code:

```
_MONGOSH
> db.wild_animals.find()
< [
  {
    _id: ObjectId('699fe120615ac525cbbc8dbe'),
    animal_name: 'Lion',
    nature: 'harm',
    favorite_foods: [
      'meat',
      'deer'
    ],
    care_taker_name: 'Siva',
    life_span: 15,
    timestamp: 2017-01-15T00:00:00.000Z,
    expenses: 55000
  },
  {
    _id: ObjectId('699fe120615ac525cbbc8dbf'),
    animal_name: 'Tiger',
    nature: 'harm',
    favorite_foods: [
      'meat',
      'rabbits'
    ],
    care_taker_name: 'Kanth',
    life_span: 14,
    timestamp: 2018-06-20T00:00:00.000Z,
    expenses: 60000
  },
  {
    _id: ObjectId('699fe120615ac525cbbc8dc0'),
    animal_name: 'Leopard',
    nature: 'harm',
    favorite_foods: [
      'meat',
      'small mammals'
    ],
    care_taker_name: 'Ravi',
    life_span: 18,
    timestamp: 2019-03-10T00:00:00.000Z,
    expenses: 50000
  }
]
```

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

**Compass**

My Queries

**CONNECTIONS (1)**

localhost:27017

admin config local test vehicles

four\_wHEELERS two\_wHEELERS

**\_MONGOSH**

```
{  
  "_id": ObjectId('699fe120615ac525cbcb8dc0'),  
  "animal_name": "Elephant",  
  "nature": "harmless",  
  "favorite_foods": [  
    "grass",  
    "fruits"  
  ],  
  "care_taker_name": "Mahesh",  
  "life_span": 65,  
  "timestamp": 2004-03-10T00:00:00.000Z,  
  "expenses": 90000  
}  
  
{  
  "_id": ObjectId('699fe120615ac525cbcb8dc1'),  
  "animal_name": "Bear",  
  "nature": "harm",  
  "favorite_foods": [  
    "fish",  
    "honey"  
  ],  
  "care_taker_name": "prashant",  
  "life_span": 25,  
  "timestamp": 2016-11-05T00:00:00.000Z,  
  "expenses": 50000  
}  
  
{  
  "_id": ObjectId('699fe120615ac525cbcb8dc2'),  
  "animal_name": "Giraffe",  
  "nature": "harmless",  
  "favorite_foods": [  
    "leaves"  
  ],  
  "care_taker_name": "Kiran",  
  "life_span": 26,  
  "timestamp": 2009-08-12T00:00:00.000Z,  
  "expenses": 45000  
}  
animal>|
```

ENG IN 11:48 26-02-2026

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

**Compass**

My Queries

**CONNECTIONS (1)**

localhost:27017

admin config local test vehicles

four\_wHEELERS two\_wHEELERS

**\_MONGOSH**

```
timestamp: 2004-03-10T00:00:00.000Z,  
expenses: 90000  
}  
  
{  
  "_id": ObjectId('699fe120615ac525cbcb8dc1'),  
  "animal_name": "Bear",  
  "nature": "harm",  
  "favorite_foods": [  
    "fish",  
    "honey"  
  ],  
  "care_taker_name": "prashant",  
  "life_span": 25,  
  "timestamp": 2016-11-05T00:00:00.000Z,  
  "expenses": 50000  
}  
  
{  
  "_id": ObjectId('699fe120615ac525cbcb8dc2'),  
  "animal_name": "Giraffe",  
  "nature": "harmless",  
  "favorite_foods": [  
    "leaves"  
  ],  
  "care_taker_name": "Kiran",  
  "life_span": 26,  
  "timestamp": 2009-08-12T00:00:00.000Z,  
  "expenses": 45000  
}  
animal>|
```

ENG IN 11:48 26-02-2026

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

**Compass**

My Queries

**CONNECTIONS** (1)

localhost:27017

- admin
- config
- local
- test
- vehicles
  - four\_wheelers
  - two\_wheelers

Search connections

```
_MONGOSH
> db.domestic_animals.find()
< [
  {
    _id: ObjectId('699fe489615ac525cbcb8dc3'),
    animal_name: 'Dog',
    gender: 'female',
    favorite_foods: [
      'meat',
      'rice'
    ],
    animal_petsname: 'puppy',
    life_span: 15,
    timestamp: 2019-02-10T00:00:00.000Z,
    expenses: 12000
  },
  {
    _id: ObjectId('699fe489615ac525cbcb8dc4'),
    animal_name: 'Cat',
    gender: 'female',
    favorite_foods: [
      'fish',
      'milk'
    ],
    animal_petsname: 'Kitty',
    life_span: 15,
    timestamp: 2018-07-16T00:00:00.000Z,
    expenses: 8000
  },
  {
    _id: ObjectId('699fe489615ac525cbcb8dc5'),
    animal_name: 'Cow',
    gender: 'female',
    favorite_foods: [
      'grass',
      'hay'
    ],
    animal_petsname: 'lakshmi',
    life_span: 25,
    timestamp: 2019-03-25T00:00:00.000Z,
    expenses: 15000
  },
  {
    _id: ObjectId('699fe489615ac525cbcb8dc6'),
    animal_name: 'Goat',
    gender: 'male',
    favorite_foods: [
      'leaves',
      'grass'
    ],
    animal_petsname: 'puri',
    life_span: 12,
    timestamp: 2022-01-05T00:00:00.000Z,
    expenses: 7000
  },
  {
    _id: ObjectId('699fe489615ac525cbcb8dc7'),
    animal_name: 'Horse',
    gender: 'male',
    favorite_foods: [
      'grass',
      'leaves'
    ],
    animal_petsname: 'Shiva',
    life_span: 30,
    timestamp: 2022-01-05T00:00:00.000Z,
    expenses: 20000
  }
]
```

ENG IN 11:55 26-02-2026

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

**Compass**

My Queries

**CONNECTIONS** (1)

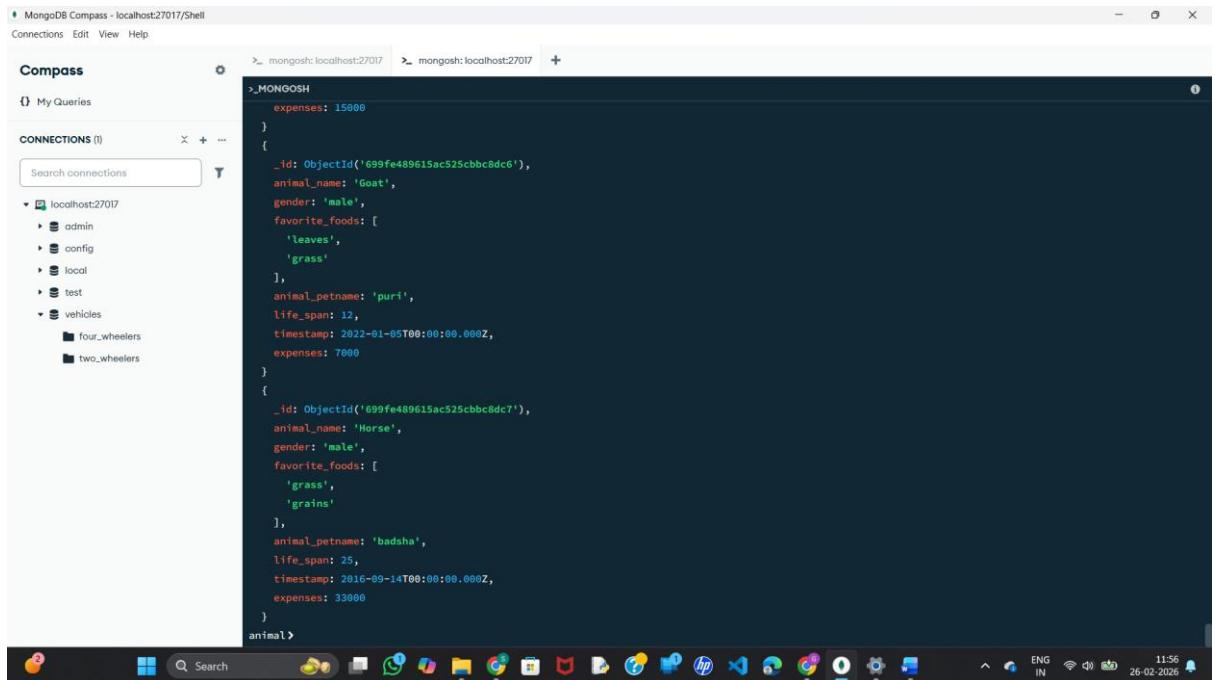
localhost:27017

- admin
- config
- local
- test
- vehicles
  - four\_wheelers
  - two\_wheelers

Search connections

```
_MONGOSH
> db.domestic_animals.find()
< [
  {
    _id: ObjectId('699fe489615ac525cbcb8dc5'),
    animal_name: 'Cow',
    gender: 'female',
    favorite_foods: [
      'grass',
      'hay'
    ],
    animal_petsname: 'lakshmi',
    life_span: 25,
    timestamp: 2019-03-25T00:00:00.000Z,
    expenses: 15000
  },
  {
    _id: ObjectId('699fe489615ac525cbcb8dc6'),
    animal_name: 'Goat',
    gender: 'male',
    favorite_foods: [
      'leaves',
      'grass'
    ],
    animal_petsname: 'puri',
    life_span: 12,
    timestamp: 2022-01-05T00:00:00.000Z,
    expenses: 7000
  },
  {
    _id: ObjectId('699fe489615ac525cbcb8dc7'),
    animal_name: 'Horse',
    gender: 'male',
    favorite_foods: [
      'grass',
      'leaves'
    ],
    animal_petsname: 'Shiva',
    life_span: 30,
    timestamp: 2022-01-05T00:00:00.000Z,
    expenses: 20000
  }
]
```

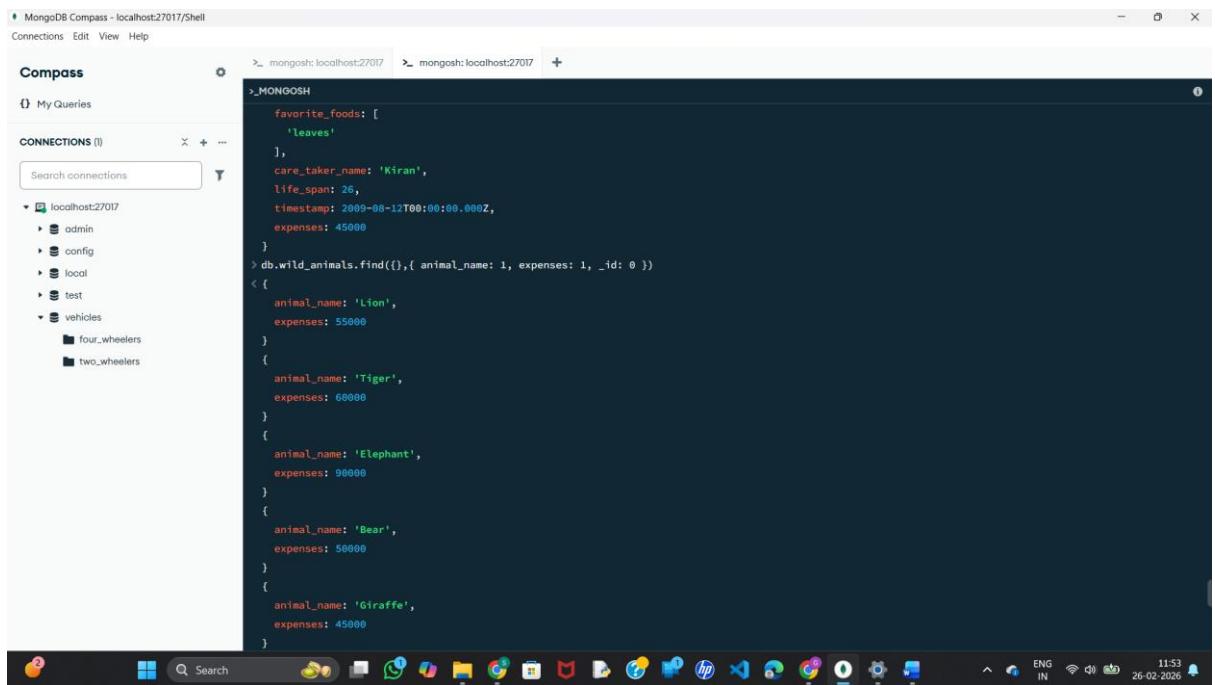
ENG IN 11:55 26-02-2026



The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists 'localhost:27017' with databases 'admin', 'config', 'local', 'test', and 'vehicles'. Under 'vehicles', there are sub-folders 'four\_wHEELERS' and 'two\_wHEELERS'. The main panel displays the contents of the '\_MONGOSH' collection. Two documents are shown:

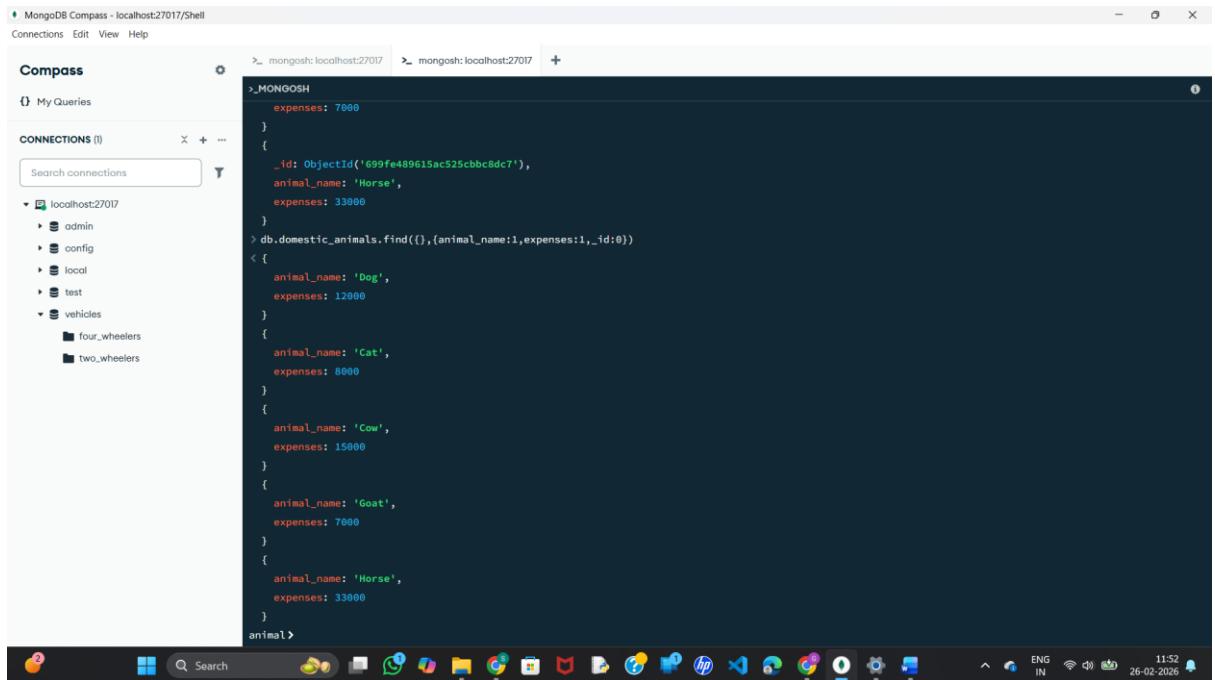
```
_MONGOSH
expenses: 15000
}
{
_id: ObjectId('699fe489615ac525cbbc8dc6'),
animal_name: 'Goat',
gender: 'male',
favorite_foods: [
'Leaves',
'grass',
],
animal_petsname: 'puri',
life_span: 12,
timestamp: 2022-01-05T00:00:00.000Z,
expenses: 7000
}
{
_id: ObjectId('699fe489615ac525cbbc8dc7'),
animal_name: 'Horse',
gender: 'male',
favorite_foods: [
'grass',
'grains'
],
animal_petsname: 'badsha',
life_span: 25,
timestamp: 2016-09-14T00:00:00.000Z,
expenses: 33000
}
animal>
```

7. Write a MongoDB query to display only animal name and expenses in all the collection of the database



The screenshot shows the MongoDB Compass interface. The 'Connections' sidebar is identical to the previous one. The main panel shows a query being run in the shell:

```
>_MONGOSH
favorite_foods: [
'Leaves'
],
care_taker_name: 'Kiran',
life_span: 26,
timestamp: 2009-08-12T00:00:00.000Z,
expenses: 45000
}
>db.wild_animals.find({}, { animal_name: 1, expenses: 1, _id: 0 })
<[
{
animal_name: 'Lion',
expenses: 55000
},
{
animal_name: 'Tiger',
expenses: 60000
},
{
animal_name: 'Elephant',
expenses: 90000
},
{
animal_name: 'Bear',
expenses: 50000
},
{
animal_name: 'Giraffe',
expenses: 45000
}]
```



MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

localhost:27017

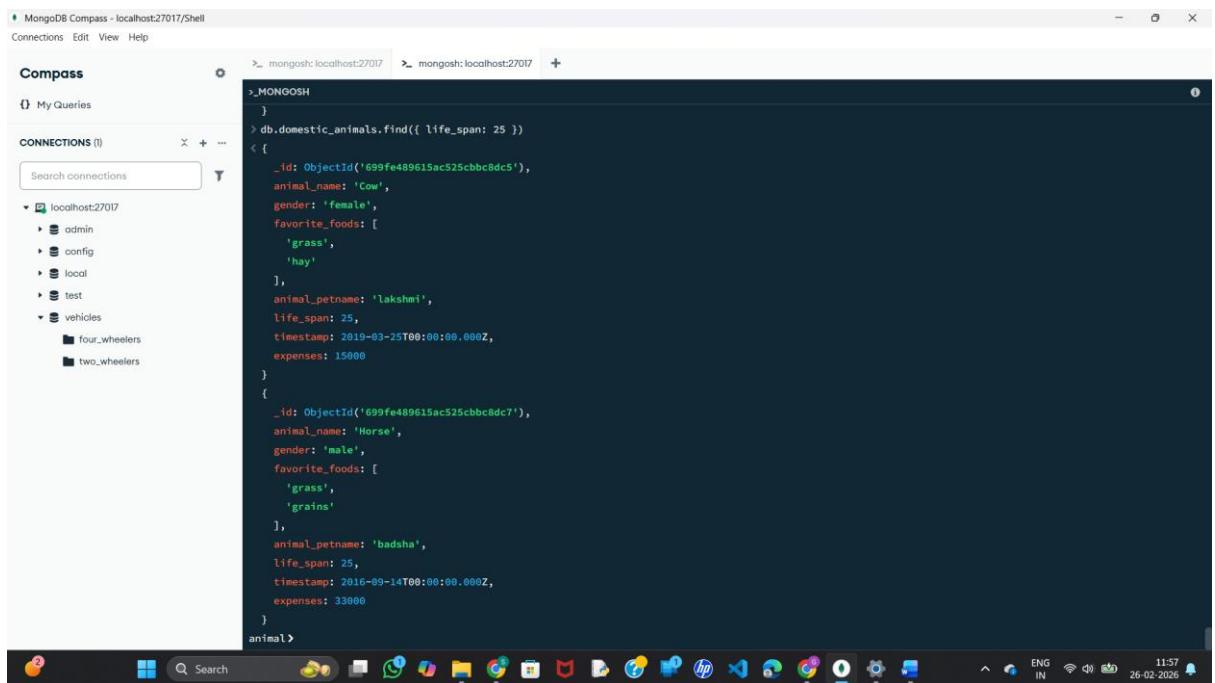
admin config local test vehicles

four\_wHEELERS two\_wHEELERS

\_MONGOSH

```
>_MONGOSH
expenses: 7000
}
{
_id: ObjectId('699fe489615ac525cbbc8dc7'),
animal_name: 'Horse',
expenses: 33000
}
>db.domestic_animals.find({}, {animal_name:1, expenses:1, _id:0})
<[
  {
    animal_name: 'Dog',
    expenses: 12000
  },
  {
    animal_name: 'Cat',
    expenses: 8000
  },
  {
    animal_name: 'Cow',
    expenses: 15000
  },
  {
    animal_name: 'Goat',
    expenses: 7000
  },
  {
    animal_name: 'Horse',
    expenses: 33000
  }
]
animal>
```

8. Write a MongoDB query to display domestic\_animals whose life is a particular year



MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

localhost:27017

admin config local test vehicles

four\_wHEELERS two\_wHEELERS

\_MONGOSH

```
>_MONGOSH
}
>db.domestic_animals.find({ life_span: 25 })
<[
  {
    _id: ObjectId('699fe489615ac525cbbc8dc5'),
    animal_name: 'Cow',
    gender: 'female',
    favorite_foods: [
      'grass',
      'hay'
    ],
    animal_petname: 'lakshmi',
    life_span: 25,
    timestamp: 2019-03-25T00:00:00.000Z,
    expenses: 15000
  },
  {
    _id: ObjectId('699fe489615ac525cbbc8dc7'),
    animal_name: 'Horse',
    gender: 'male',
    favorite_foods: [
      'grass',
      'grains'
    ],
    animal_petname: 'badsha',
    life_span: 25,
    timestamp: 2016-09-14T00:00:00.000Z,
    expenses: 33000
  }
]
animal>
```

9. Write a MongoDB query to display wild\_animals available under a particular care\_taker

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

localhost:27017

admin config local test vehicles

four\_wHEELERS two\_wHEELERS

\_id: ObjectId('699fe489615ac525cbbc8dc7'), animal\_name: 'Horse', gender: 'male', favorite\_foods: [ 'grass', 'grains' ], animal\_petrname: 'badsha', life\_span: 25, timestamp: 2016-09-14T00:00:00.000Z, expenses: 33000 } > db.wild\_animals.find({ care\_taker\_name: "siva" }) < > db.wild\_animals.find({ care\_taker\_name: "Siva" }) < { \_id: ObjectId('699fe120615ac525cbbc8dbe'), animal\_name: 'Lion', nature: 'harm', favorite\_foods: [ 'meat', 'deer' ], care\_taker\_name: 'Siva', life\_span: 15, timestamp: 2017-01-15T00:00:00.000Z, expenses: 55000 } animal>

This screenshot shows the MongoDB Compass interface. On the left, there's a sidebar with 'Compass' at the top, followed by 'My Queries', 'CONNECTIONS (1)', and a dropdown menu for 'localhost:27017'. Under 'localhost:27017', there are connections for 'admin', 'config', 'local', 'test', and 'vehicles'. Within 'vehicles', there are two sub-folders: 'four\_wHEELERS' and 'two\_wHEELERS'. The main pane displays a mongo shell session. The command `db.wild\_animals.find({ life\_span: { \$gt: 5 } })` is run, and the results are shown. There are two documents returned. The first document is for a 'Horse' with an ID of '699fe489615ac525cbbc8dc7'. It has a life span of 25 years. The second document is for a 'Lion' with an ID of '699fe120615ac525cbbc8dbe'. It has a life span of 15 years. Both documents include fields for 'animal\_name', 'gender', 'favorite\_foods', 'animal\_petrname', 'care\_taker\_name', 'life\_span', 'timestamp', and 'expenses'. The interface includes a search bar and a toolbar with various icons.

10. Write a MongoDB query to display animal name, favorite\_foods and expenses details whose lifespan is more than 5 years.

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

localhost:27017

admin config local test vehicles

four\_wHEELERS two\_wHEELERS

\_id: ObjectId('699fe120615ac525cbbc8dbe'), animal\_name: 'Lion', favorite\_foods: [ 'meat', 'deer' ], expenses: 55000 } < { animal\_name: 'Tiger', favorite\_foods: [ 'meat', 'rabbits' ], expenses: 60000 } & { animal\_name: 'Elephant', favorite\_foods: [ 'grass', 'fruits' ], expenses: 90000 } & { animal\_name: 'Bear', favorite\_foods: [ 'fish' ], expenses: 45000 }

This screenshot shows the MongoDB Compass interface. The setup is identical to the previous one, with the 'localhost:27017' connection selected. The mongo shell session shows the command `db.wild\_animals.find({ life\_span: { \$gt: 5 } })` being run again. The results are displayed in the main pane. This time, three documents are returned: a 'Lion' (ID: '699fe120615ac525cbbc8dbe'), a 'Tiger' (ID: '699fe120615ac525cbbc8dbe'), and an 'Elephant' (ID: '699fe120615ac525cbbc8dbe'). Each document contains fields for 'animal\_name', 'favorite\_foods', and 'expenses'. The 'Lion' has expenses of 55000, the 'Tiger' has expenses of 60000, and the 'Elephant' has expenses of 90000. The interface includes a search bar and a toolbar with various icons.

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

localhost:27017

Search connections

mongosh: localhost:27017 | mongosh: localhost:27017 | +

```
_MONGOSH
{
  animal_name: 'Giraffe',
  favorite_foods: [
    'leaves'
  ],
  expenses: 45000
}
> db.domestic_animals.find({ life_span: { $gt: 5 } }, { animal_name: 1, favorite_foods: 1, expenses: 1, _id: 0 })
< [
  {
    animal_name: 'Dog',
    favorite_foods: [
      'meat',
      'rice'
    ],
    expenses: 12000
  },
  {
    animal_name: 'Cat',
    favorite_foods: [
      'fish',
      'milk'
    ],
    expenses: 8000
  },
  {
    animal_name: 'Cow',
    favorite_foods: [
      'grass',
      'hay'
    ],
    expenses: 15000
  },
  {
    animal_name: 'Goat',
    favorite_foods: [
      'leaves',
      'grass'
    ],
    expenses: 7000
  },
  {
    animal_name: 'Horse',
    favorite_foods: [
      'grass',
      'grains'
    ],
    expenses: 33000
  }
]
animal>
```

MongoDB Compass - localhost:27017/Shell

Connections Edit View Help

Compass

My Queries

CONNECTIONS (1)

localhost:27017

Search connections

mongosh: localhost:27017 | mongosh: localhost:27017 | +

```
_MONGOSH
{
  'milk'
},
expenses: 8000
}
{
  animal_name: 'Cow',
  favorite_foods: [
    'grass',
    'hay'
  ],
  expenses: 15000
}
{
  animal_name: 'Goat',
  favorite_foods: [
    'leaves',
    'grass'
  ],
  expenses: 7000
}
{
  animal_name: 'Horse',
  favorite_foods: [
    'grass',
    'grains'
  ],
  expenses: 33000
}
animal>
```