MongoDB

What is MongoDB?

MongoDB is a **NoSQL database** that stores data in a **flexible, JSON-like format (called BSON)** instead of traditional rows and columns like in SQL databases. It is designed for **high performance**, **scalability**, and **ease of development**.

Key Features of MongoDB:

- **Document-Oriented**: Stores data in collections of documents (similar to JSON).
- **Schema-less**: Documents in a collection can have different fields.
- **High Performance**: Optimized for fast read/write operations.
- **Horizontal Scaling**: Supports sharding and distributed systems.
- **Flexible Queries**: Rich query language, supports filtering, sorting, and aggregation.

Example:

A document in MongoDB looks like:

```
{
  "name": "Alice",
  "age": 25,
  "city": "Chennai"
}
```

MongoDB Installation (Windows)

Step-by-Step Installation:

1. **Download MongoDB Community Edition**

   - Visit: https://www.mongodb.com/try/download/community
   - Choose:
     - ✓ Version: MongoDB Community Server
     - ✓ OS: Windows
     - ✓ Package: .msi installer

2. **Install MongoDB**

   - Run the downloaded .msi file.
   - Choose **Complete Installation**.

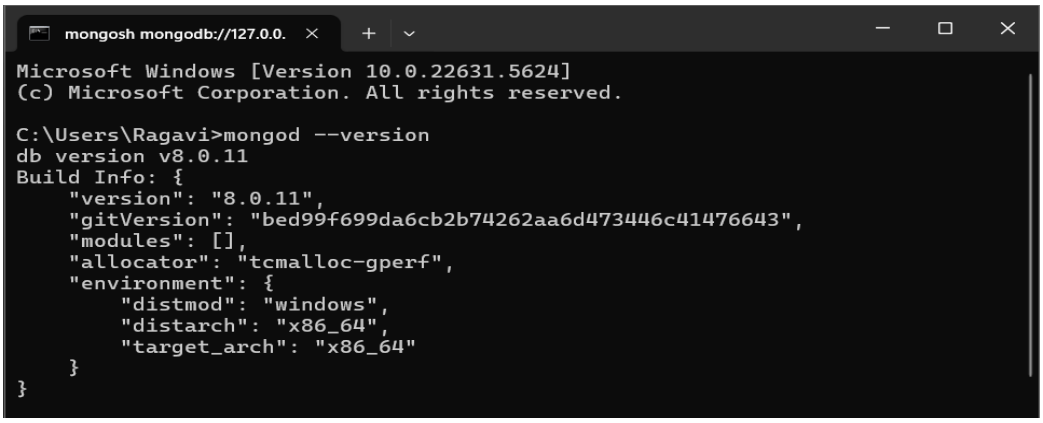
- Select **Install MongoDB as a Service** (recommended).

3. **Install MongoDB Shell (mongosh)**

- mongosh is the new MongoDB shell used to interact with the database.
- It is usually bundled with the installer. If not, download separately from: https://www.mongodb.com/try/download/shell

4. **Checking MongoDB Version**

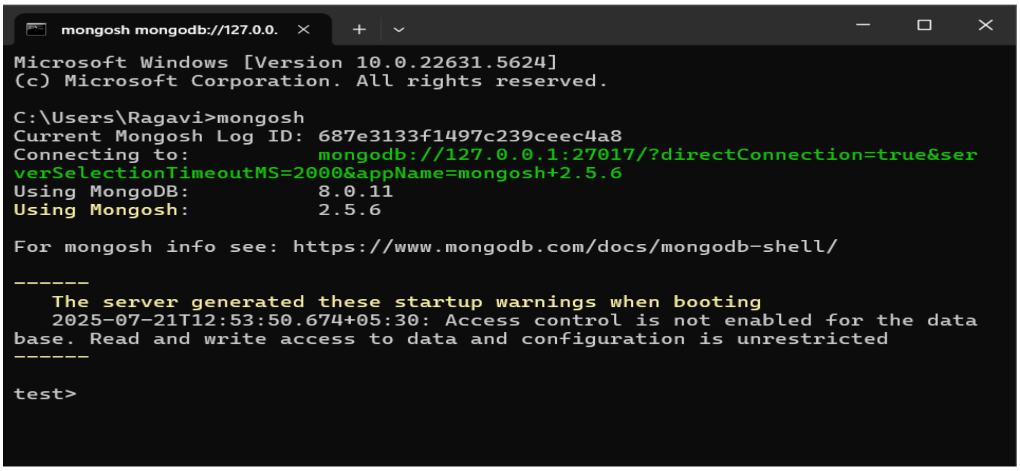
- To verify that MongoDB is correctly installed and to check the version of the MongoDB server, you can use the following command:



```
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ragavi>mongod --version
db version v8.0.11
Build Info: {
    "version": "8.0.11",
    "gitVersion": "bed99f699da6cb2b74262aa6d473446c41476643",
    "modules": [],
    "allocator": "tcmalloc-gperf",
    "environment": {
        "distmod": "windows",
        "distarch": "x86_64",
        "target_arch": "x86_64"
    }
}
```

5. **Verify Installation in Command Prompt**

- Open **Command Prompt** and run:

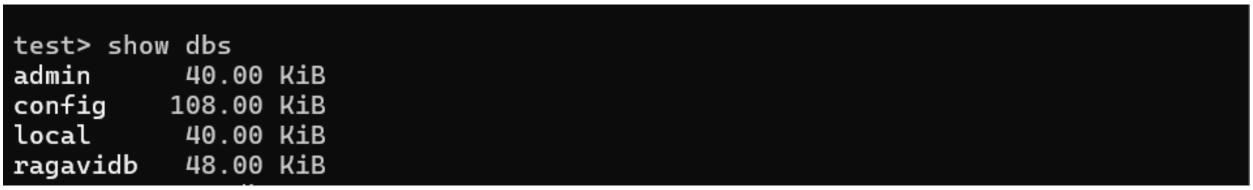


```
Microsoft Windows [Version 10.0.22631.5624]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Ragavi>mongosh
Current Mongosh Log ID: 687e3133f1497c239ceec4a8
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&ser
verSelectionTimeoutMS=2000&appName=mongosh+2.5.6
Using MongoDB:          8.0.11
Using Mongosh:          2.5.6

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

------
   The server generated these startup warnings when booting
   2025-07-21T12:53:50.674+05:30: Access control is not enabled for the data
base. Read and write access to data and configuration is unrestricted
------

test>
```

- If installed correctly, it opens the MongoDB shell.

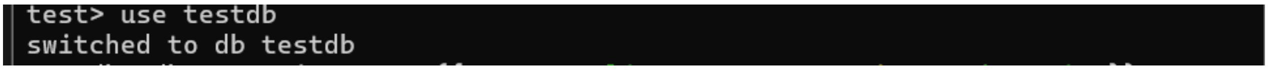
MongoDB Commands

1. Show All Databases



```
test> show dbs
admin      40.00 KiB
config    108.00 KiB
local      40.00 KiB
ragavidb   48.00 KiB
```

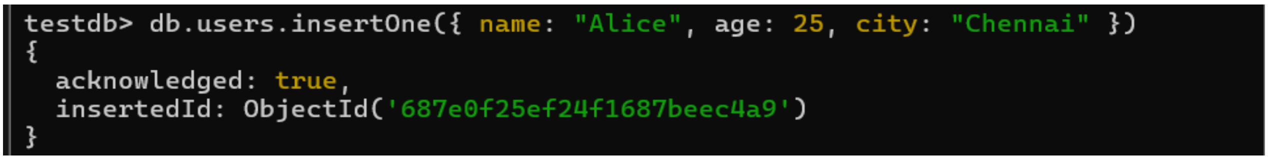
2. Create or Switch to a Database



```
test> use testdb
switched to db testdb
```

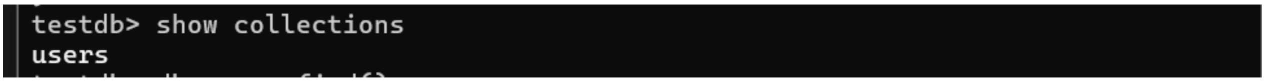
This will switch to testdb. If it doesn't exist, it'll be created when you insert data.

3. Create a Collection and Insert a Document



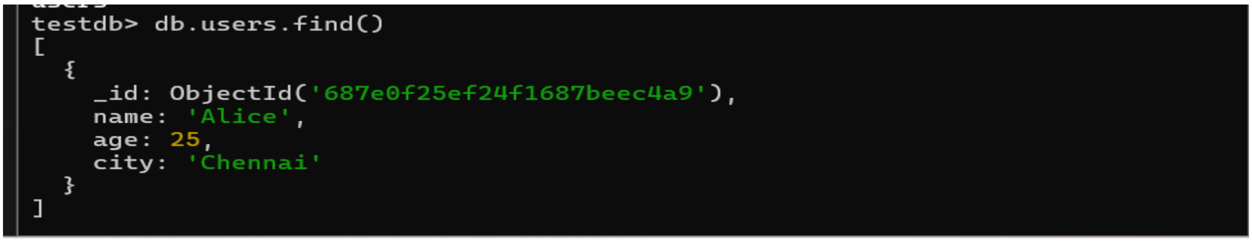
```
testdb> db.users.insertOne({ name: "Alice", age: 25, city: "Chennai" })
{
  acknowledged: true,
  insertedId: ObjectId('687e0f25ef24f1687beec4a9')
}
```

4. View Collections in Current DB



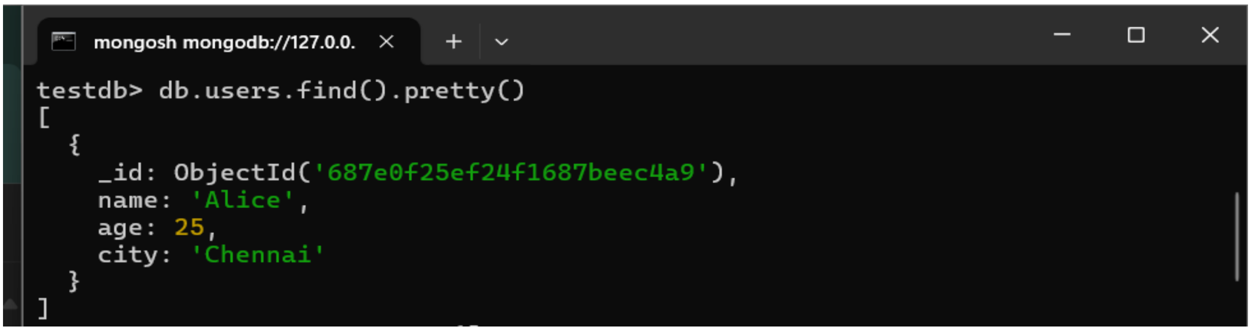
```
testdb> show collections
users
```

5. Find All Documents in a Collection



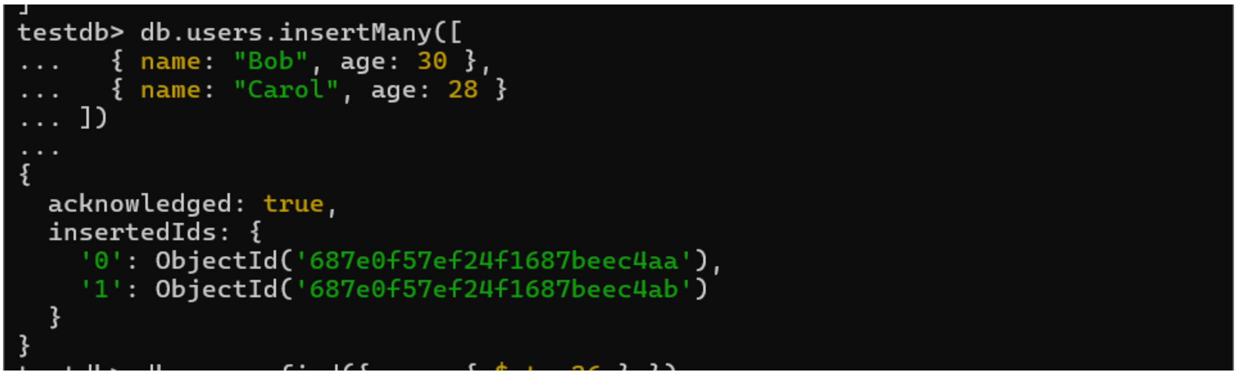
```
testdb> db.users.find()
[
  {
    _id: ObjectId('687e0f25ef24f1687beec4a9'),
    name: 'Alice',
    age: 25,
    city: 'Chennai'
  }
]
```

6. Pretty Print Output



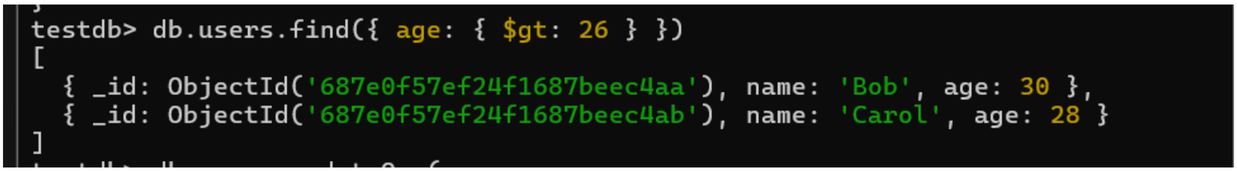
```
testdb> db.users.find().pretty()
[
  {
    _id: ObjectId('687e0f25ef24f1687beec4a9'),
    name: 'Alice',
    age: 25,
    city: 'Chennai'
  }
]
```

7. Insert Multiple Documents



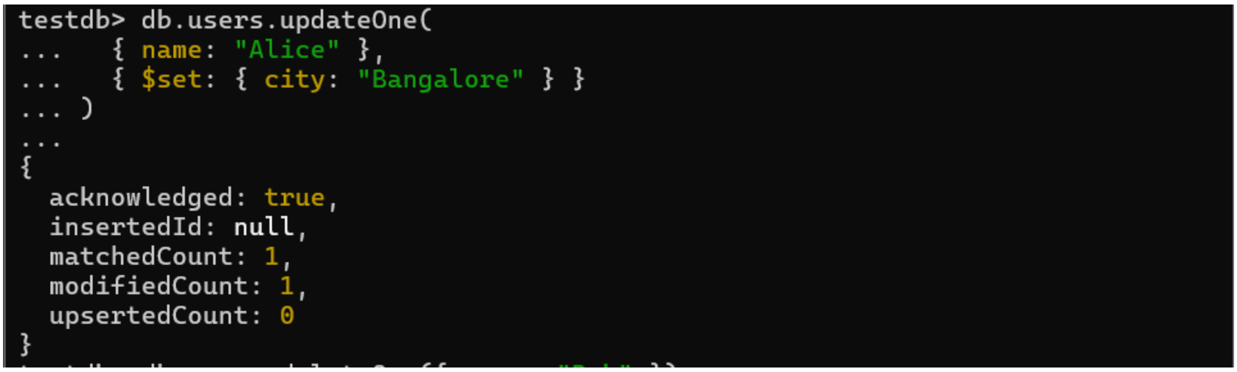
```
testdb> db.users.insertMany([
...   { name: "Bob", age: 30 },
...   { name: "Carol", age: 28 }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('687e0f57ef24f1687beec4aa'),
    '1': ObjectId('687e0f57ef24f1687beec4ab')
  }
}
```

8. Query with a Condition



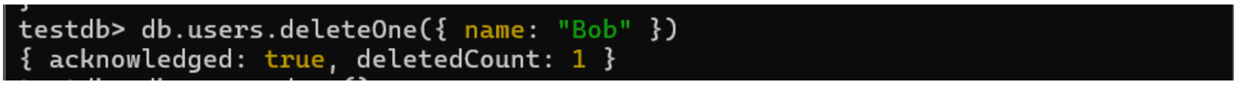
```
testdb> db.users.find({ age: { $gt: 26 } })
[
  { _id: ObjectId('687e0f57ef24f1687beec4aa'), name: 'Bob', age: 30 },
  { _id: ObjectId('687e0f57ef24f1687beec4ab'), name: 'Carol', age: 28 }
]
```

9. Update a Document



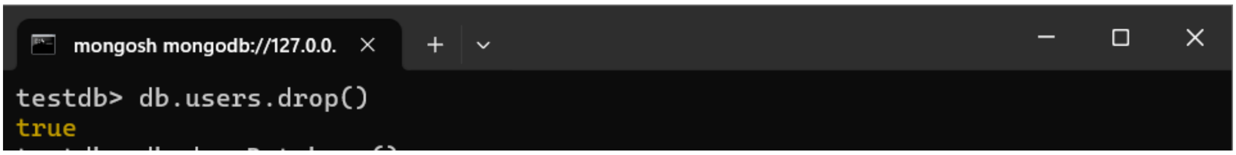
```
testdb> db.users.updateOne(
...   { name: "Alice" },
...   { $set: { city: "Bangalore" } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

10. Delete a Document



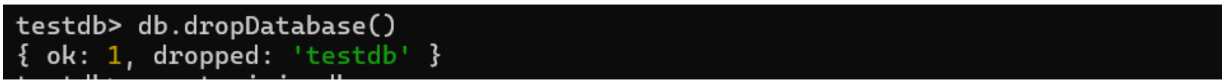
```
testdb> db.users.deleteOne({ name: "Bob" })
{ acknowledged: true, deletedCount: 1 }
```

11. Drop a Collection



```
testdb> db.users.drop()
true
```

12. Drop the Database



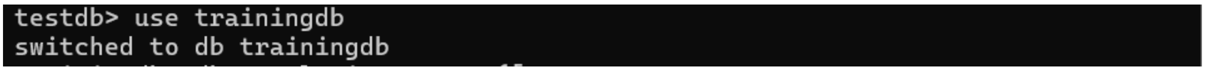
```
testdb> db.dropDatabase()
{ ok: 1, dropped: 'testdb' }
```

Prerequisites:

Before you run any query, ensure:

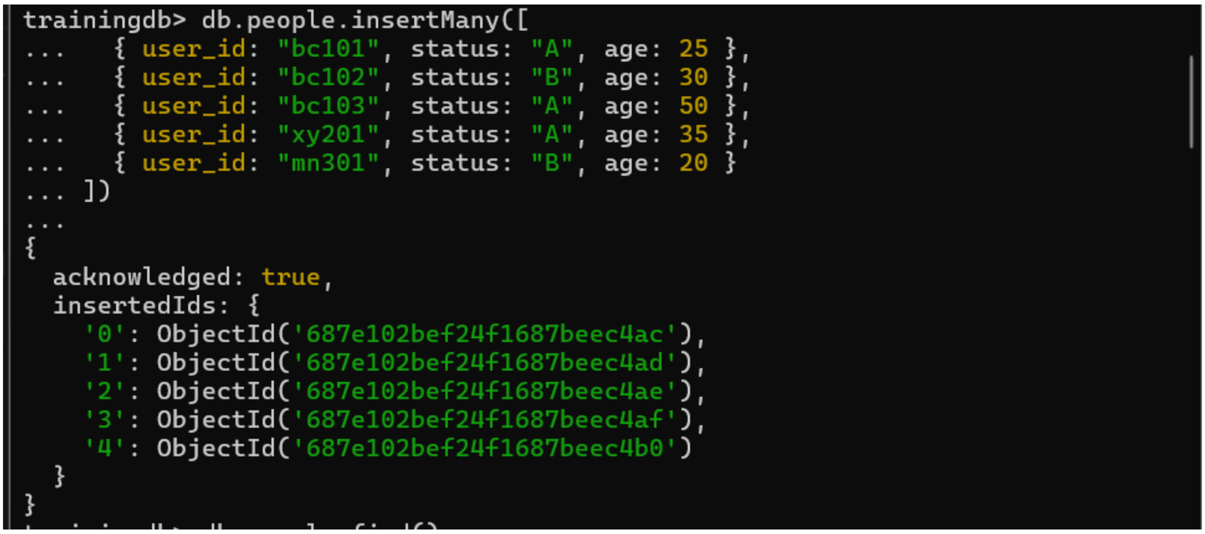
- ✓ You have started MongoDB (mongod)
- ✓ You've opened the shell (mongosh)
- ✓ You're using the database:

Switch to (or create) your database



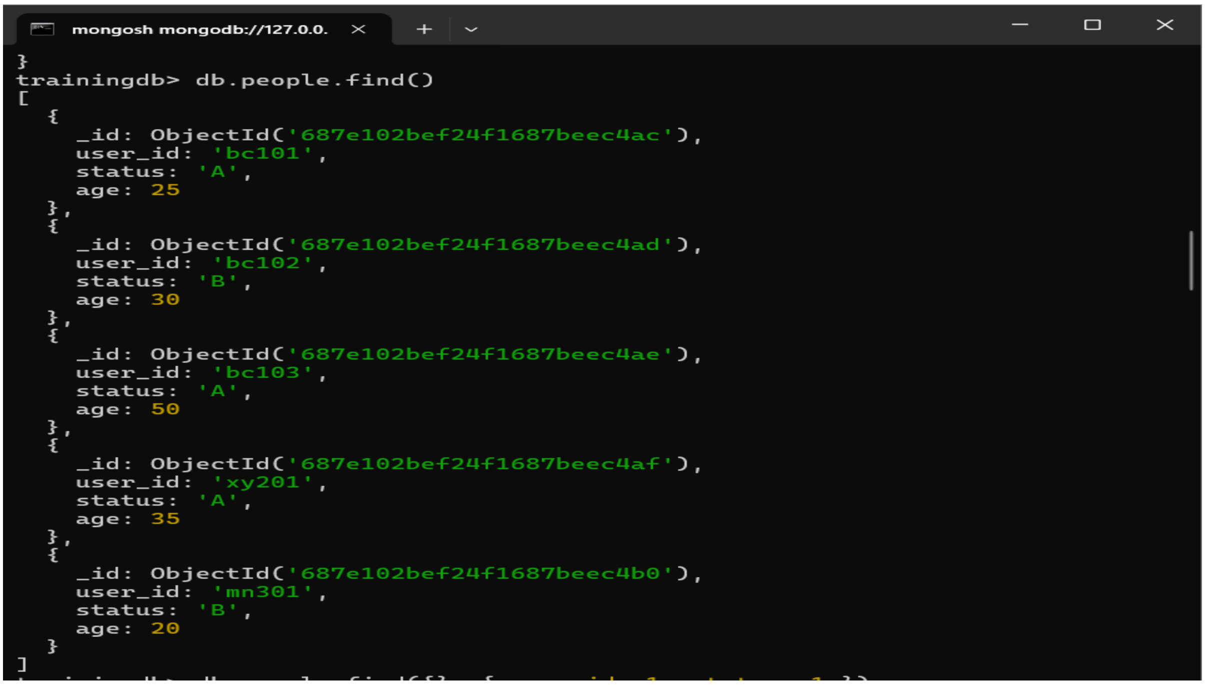
```
testdb> use trainingdb
switched to db trainingdb
```

Create sample data in people collection



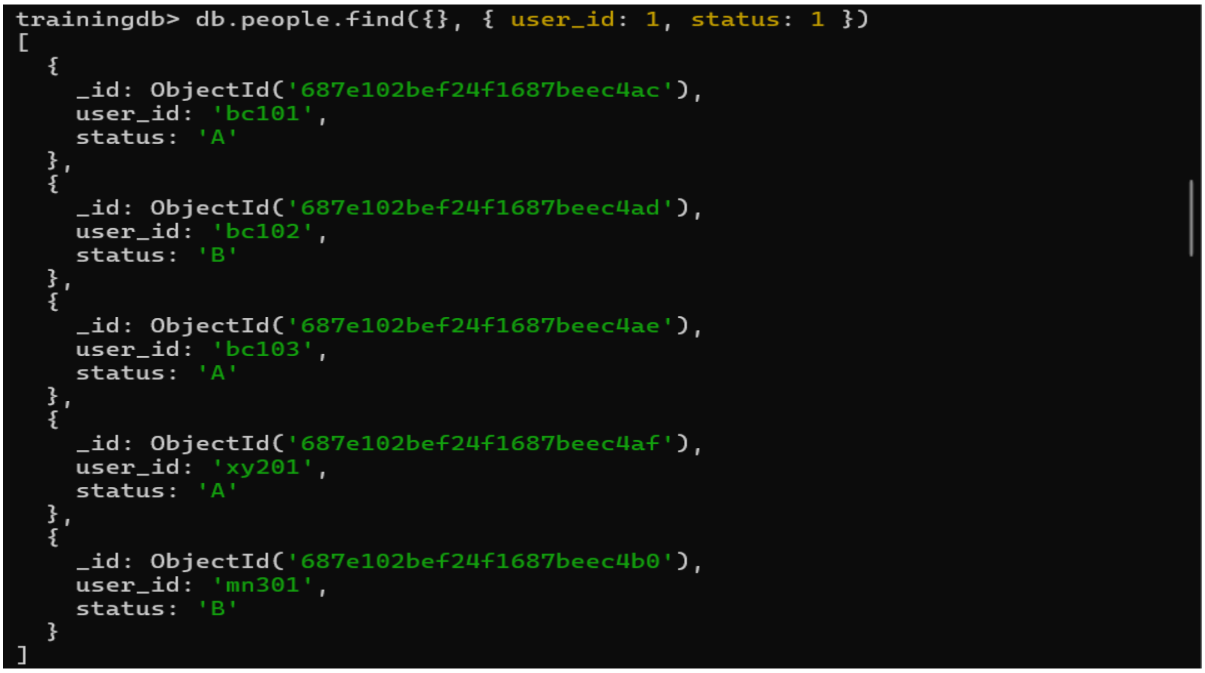
```
trainingdb> db.people.insertMany([
...    { user_id: "bc101", status: "A", age: 25 },
...    { user_id: "bc102", status: "B", age: 30 },
...    { user_id: "bc103", status: "A", age: 50 },
...    { user_id: "xy201", status: "A", age: 35 },
...    { user_id: "mn301", status: "B", age: 20 }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('687e102bef24f1687beec4ac'),
    '1': ObjectId('687e102bef24f1687beec4ad'),
    '2': ObjectId('687e102bef24f1687beec4ae'),
    '3': ObjectId('687e102bef24f1687beec4af'),
    '4': ObjectId('687e102bef24f1687beec4b0')
  }
}
```

Query 1: Display All Records



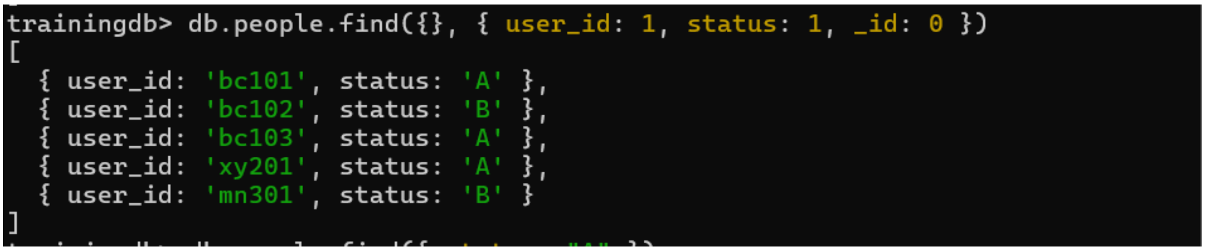
```
}
trainingdb> db.people.find()
[
  {
    _id: ObjectId('687e102bef24f1687beec4ac'),
    user_id: 'bc101',
    status: 'A',
    age: 25
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ad'),
    user_id: 'bc102',
    status: 'B',
    age: 30
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ae'),
    user_id: 'bc103',
    status: 'A',
    age: 50
  },
  {
    _id: ObjectId('687e102bef24f1687beec4af'),
    user_id: 'xy201',
    status: 'A',
    age: 35
  },
  {
    _id: ObjectId('687e102bef24f1687beec4b0'),
    user_id: 'mn301',
    status: 'B',
    age: 20
  }
]
```

Query 2: Display Specific Fields



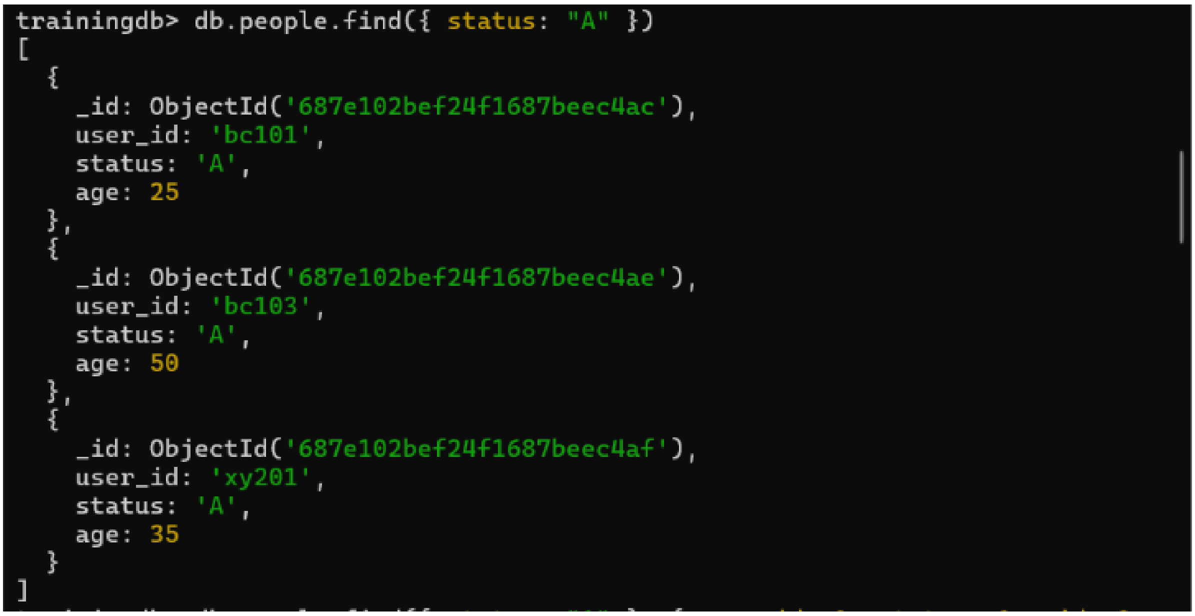
```
trainingdb> db.people.find({}, { user_id: 1, status: 1 })
[
  {
    _id: ObjectId('687e102bef24f1687beec4ac'),
    user_id: 'bc101',
    status: 'A'
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ad'),
    user_id: 'bc102',
    status: 'B'
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ae'),
    user_id: 'bc103',
    status: 'A'
  },
  {
    _id: ObjectId('687e102bef24f1687beec4af'),
    user_id: 'xy201',
    status: 'A'
  },
  {
    _id: ObjectId('687e102bef24f1687beec4b0'),
    user_id: 'mn301',
    status: 'B'
  }
]
```

Query 3: Select id, user_id, status from people

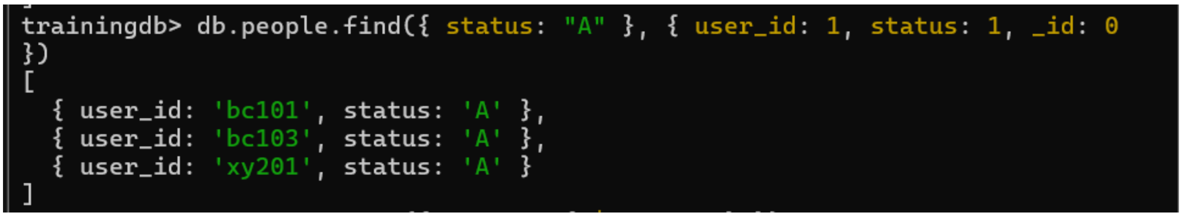


```
trainingdb> db.people.find({}, { user_id: 1, status: 1, _id: 0 })
[
  { user_id: 'bc101', status: 'A' },
  { user_id: 'bc102', status: 'B' },
  { user_id: 'bc103', status: 'A' },
  { user_id: 'xy201', status: 'A' },
  { user_id: 'mn301', status: 'B' }
]
```

Query 4: Filter by Condition - status = "A"

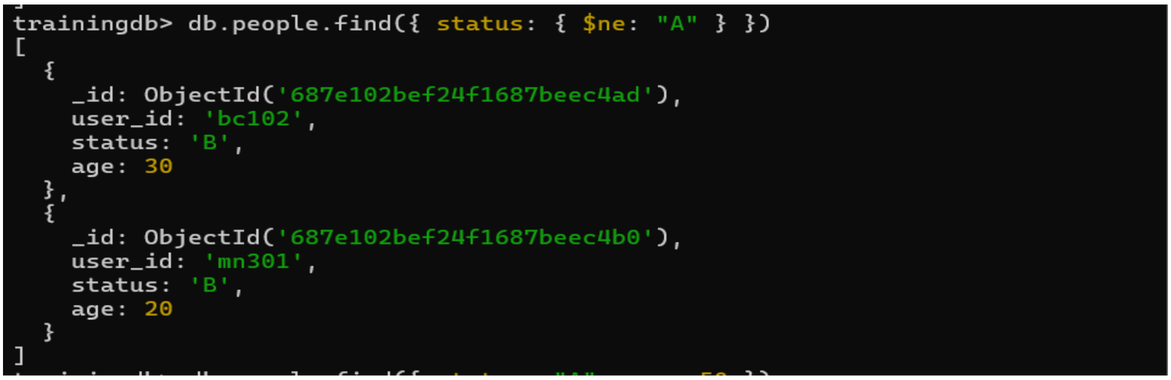


```
trainingdb> db.people.find({ status: "A" })
[
  {
    _id: ObjectId('687e102bef24f1687beec4ac'),
    user_id: 'bc101',
    status: 'A',
    age: 25
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ae'),
    user_id: 'bc103',
    status: 'A',
    age: 50
  },
  {
    _id: ObjectId('687e102bef24f1687beec4af'),
    user_id: 'xy201',
    status: 'A',
    age: 35
  }
]
```



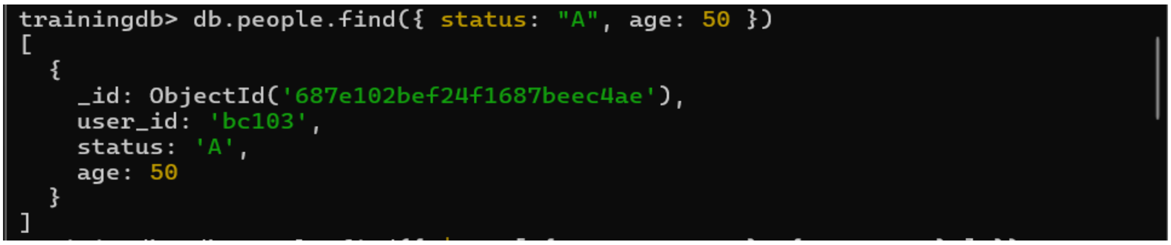
```
trainingdb> db.people.find({ status: "A" }, { user_id: 1, status: 1, _id: 0
})
[
  { user_id: 'bc101', status: 'A' },
  { user_id: 'bc103', status: 'A' },
  { user_id: 'xy201', status: 'A' }
]
```

Query 5: Exclude status = "A"



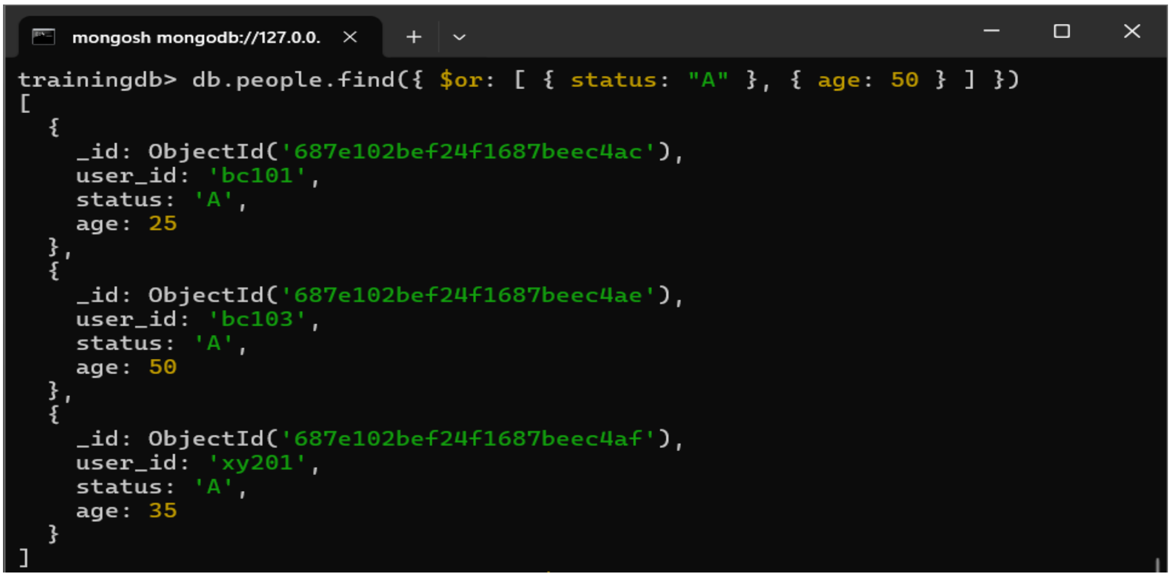
```
trainingdb> db.people.find({ status: { $ne: "A" } })
[
  {
    _id: ObjectId('687e102bef24f1687beec4ad'),
    user_id: 'bc102',
    status: 'B',
    age: 30
  },
  {
    _id: ObjectId('687e102bef24f1687beec4b0'),
    user_id: 'mn301',
    status: 'B',
    age: 20
  }
]
```

Query 6: status = "A" AND age = 50



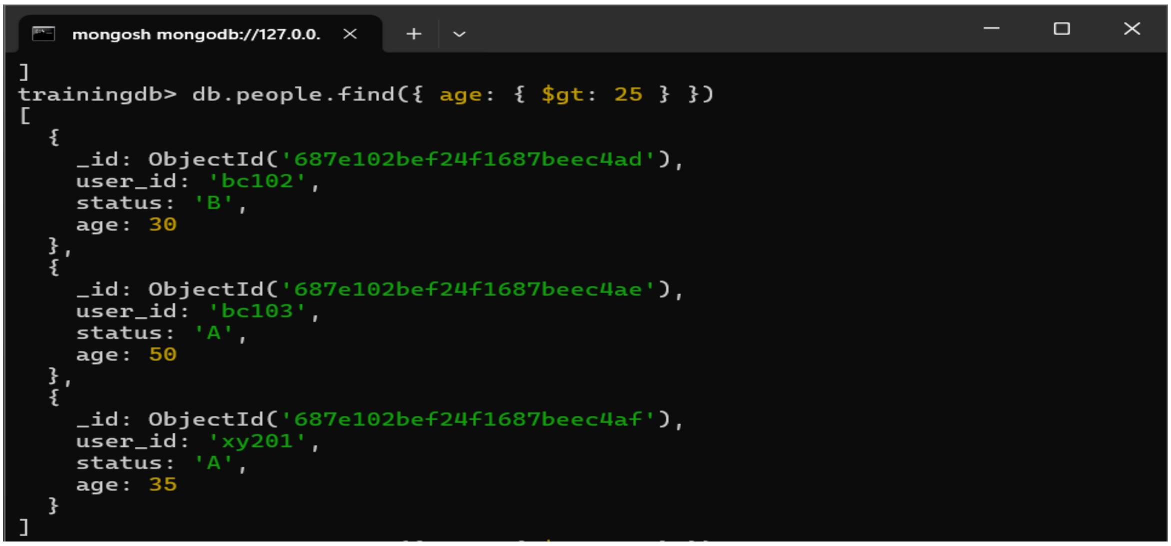
```
trainingdb> db.people.find({ status: "A", age: 50 })
[
  {
    _id: ObjectId('687e102bef24f1687beec4ae'),
    user_id: 'bc103',
    status: 'A',
    age: 50
  }
]
```

Query 7: status = "A" OR age = 50



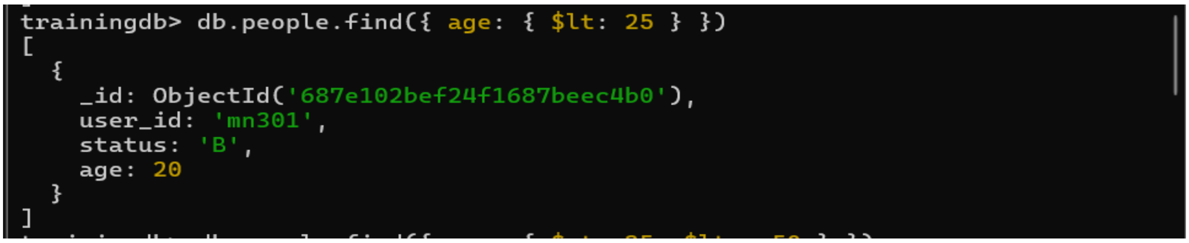
```
trainingdb> db.people.find({ $or: [ { status: "A" }, { age: 50 } ] })
[
  {
    _id: ObjectId('687e102bef24f1687beec4ac'),
    user_id: 'bc101',
    status: 'A',
    age: 25
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ae'),
    user_id: 'bc103',
    status: 'A',
    age: 50
  },
  {
    _id: ObjectId('687e102bef24f1687beec4af'),
    user_id: 'xy201',
    status: 'A',
    age: 35
  }
]
```

Query 8: age > 25



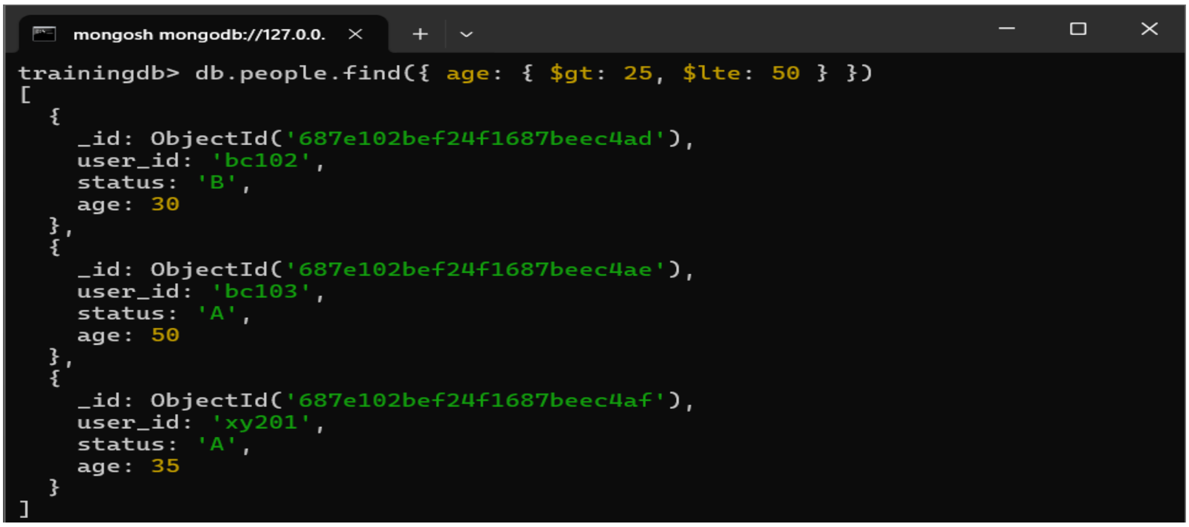
```
]
trainingdb> db.people.find({ age: { $gt: 25 } })
[
  {
    _id: ObjectId('687e102bef24f1687beec4ad'),
    user_id: 'bc102',
    status: 'B',
    age: 30
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ae'),
    user_id: 'bc103',
    status: 'A',
    age: 50
  },
  {
    _id: ObjectId('687e102bef24f1687beec4af'),
    user_id: 'xy201',
    status: 'A',
    age: 35
  }
]
```

Query 9: age < 25



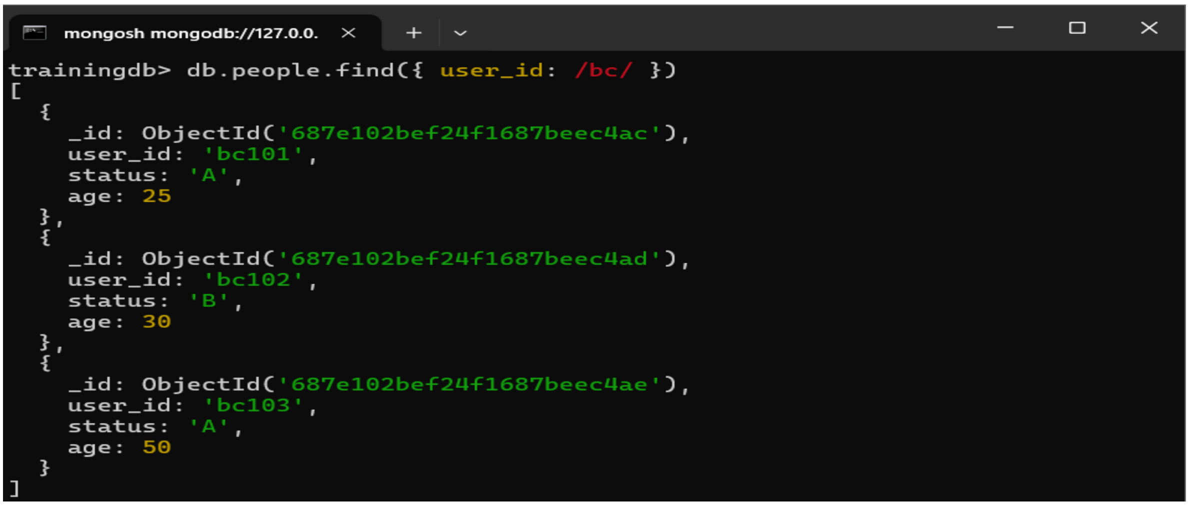
```
trainingdb> db.people.find({ age: { $lt: 25 } })
[
  {
    _id: ObjectId('687e102bef24f1687beec4b0'),
    user_id: 'mn301',
    status: 'B',
    age: 20
  }
]
```

Query 10: age > 25 AND age <= 50



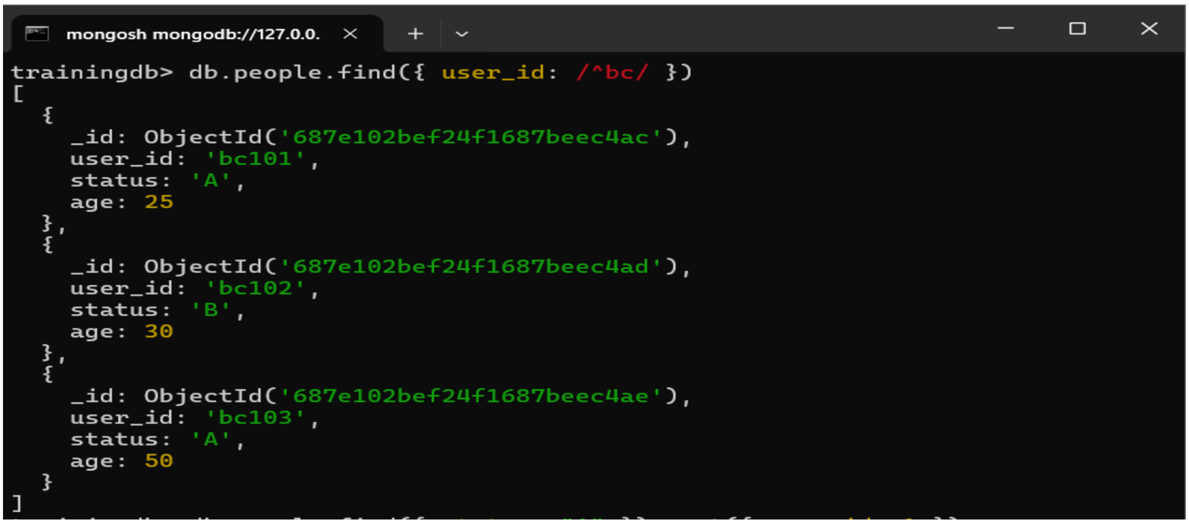
```
trainingdb> db.people.find({ age: { $gt: 25, $lte: 50 } })
[
  {
    _id: ObjectId('687e102bef24f1687beec4ad'),
    user_id: 'bc102',
    status: 'B',
    age: 30
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ae'),
    user_id: 'bc103',
    status: 'A',
    age: 50
  },
  {
    _id: ObjectId('687e102bef24f1687beec4af'),
    user_id: 'xy201',
    status: 'A',
    age: 35
  }
]
```

Query 11: user_id LIKE "%bc%"



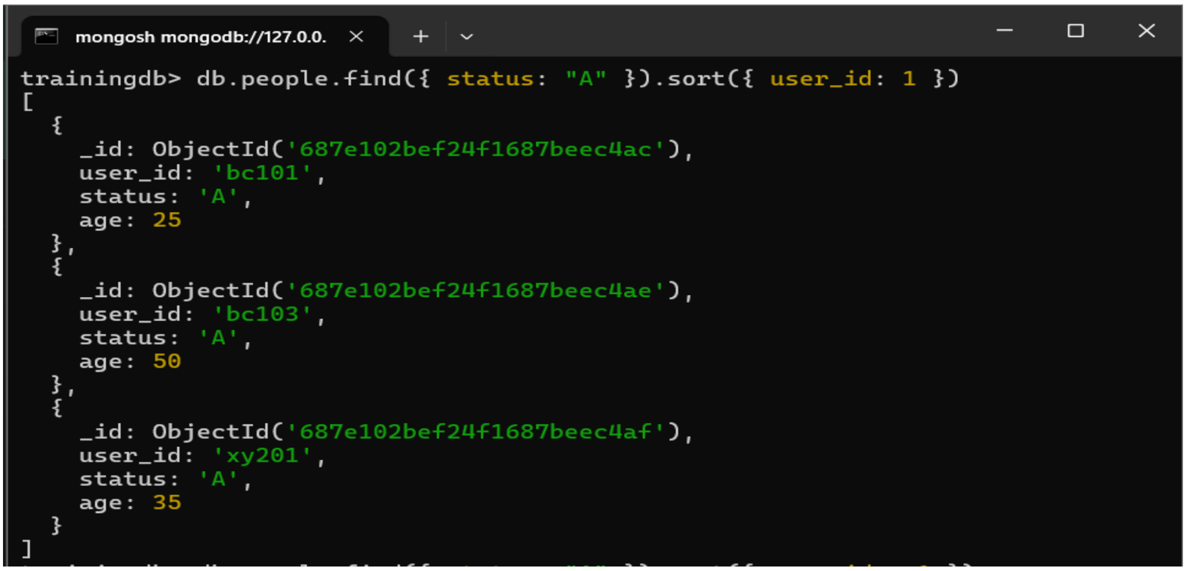
```
trainingdb> db.people.find({ user_id: /bc/ })
[
  {
    _id: ObjectId('687e102bef24f1687beec4ac'),
    user_id: 'bc101',
    status: 'A',
    age: 25
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ad'),
    user_id: 'bc102',
    status: 'B',
    age: 30
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ae'),
    user_id: 'bc103',
    status: 'A',
    age: 50
  }
]
```

Query 12: user_id LIKE "bc%"



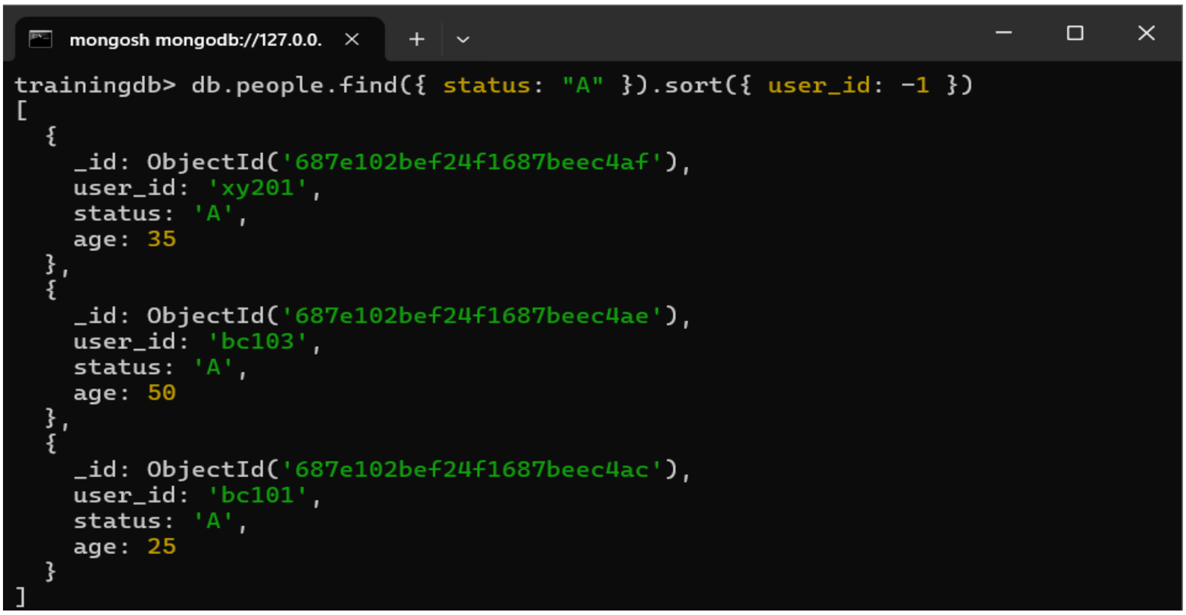
```
trainingdb> db.people.find({ user_id: /^bc/ })
[
  {
    _id: ObjectId('687e102bef24f1687beec4ac'),
    user_id: 'bc101',
    status: 'A',
    age: 25
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ad'),
    user_id: 'bc102',
    status: 'B',
    age: 30
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ae'),
    user_id: 'bc103',
    status: 'A',
    age: 50
  }
]
```

Query 13: Sort status = "A" by user_id ASC



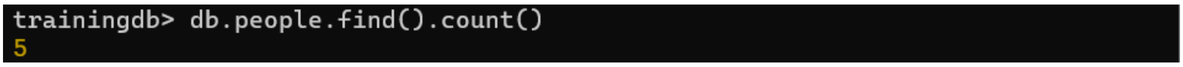
```
trainingdb> db.people.find({ status: "A" }).sort({ user_id: 1 })
[
  {
    _id: ObjectId('687e102bef24f1687beec4ac'),
    user_id: 'bc101',
    status: 'A',
    age: 25
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ae'),
    user_id: 'bc103',
    status: 'A',
    age: 50
  },
  {
    _id: ObjectId('687e102bef24f1687beec4af'),
    user_id: 'xy201',
    status: 'A',
    age: 35
  }
]
```

Query 14: Sort status = "A" by user_id DESC



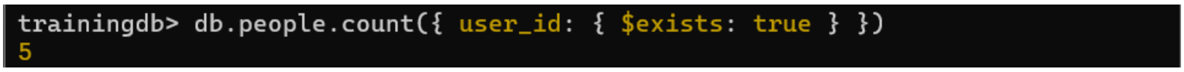
```
trainingdb> db.people.find({ status: "A" }).sort({ user_id: -1 })
[
  {
    _id: ObjectId('687e102bef24f1687beec4af'),
    user_id: 'xy201',
    status: 'A',
    age: 35
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ae'),
    user_id: 'bc103',
    status: 'A',
    age: 50
  },
  {
    _id: ObjectId('687e102bef24f1687beec4ac'),
    user_id: 'bc101',
    status: 'A',
    age: 25
  }
]
```

Query 15: Count All Records



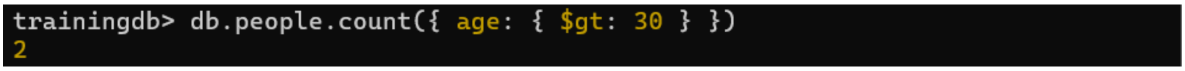
```
trainingdb> db.people.find().count()
5
```

Query 16: Count of Non-null user_id



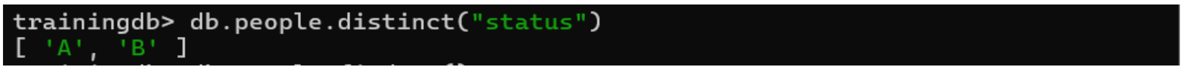
```
trainingdb> db.people.count({ user_id: { $exists: true } })
5
```

Query 17: Count with Condition – age > 30



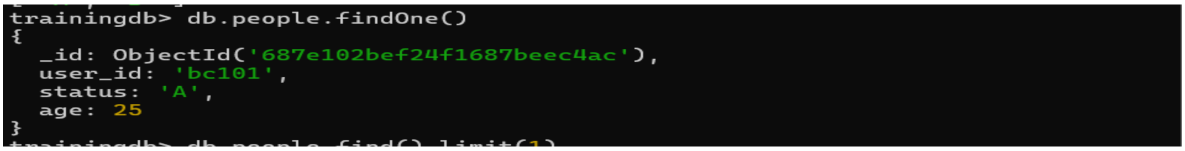
```
trainingdb> db.people.count({ age: { $gt: 30 } })
2
```

Query 18: Distinct Values of status

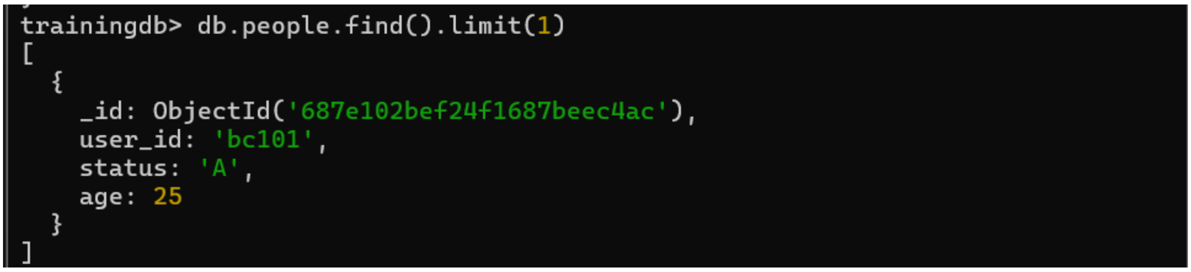


```
trainingdb> db.people.distinct("status")
[ 'A', 'B' ]
```

Query 19: Retrieve One Record

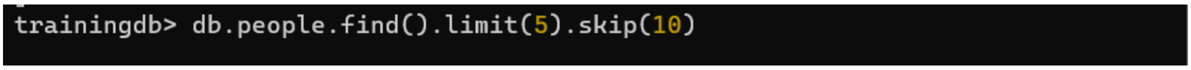


```
trainingdb> db.people.findOne()
{
  _id: ObjectId('687e102bef24f1687beec4ac'),
  user_id: 'bc101',
  status: 'A',
  age: 25
}
trainingdb> db.people.find().limit(1)
```



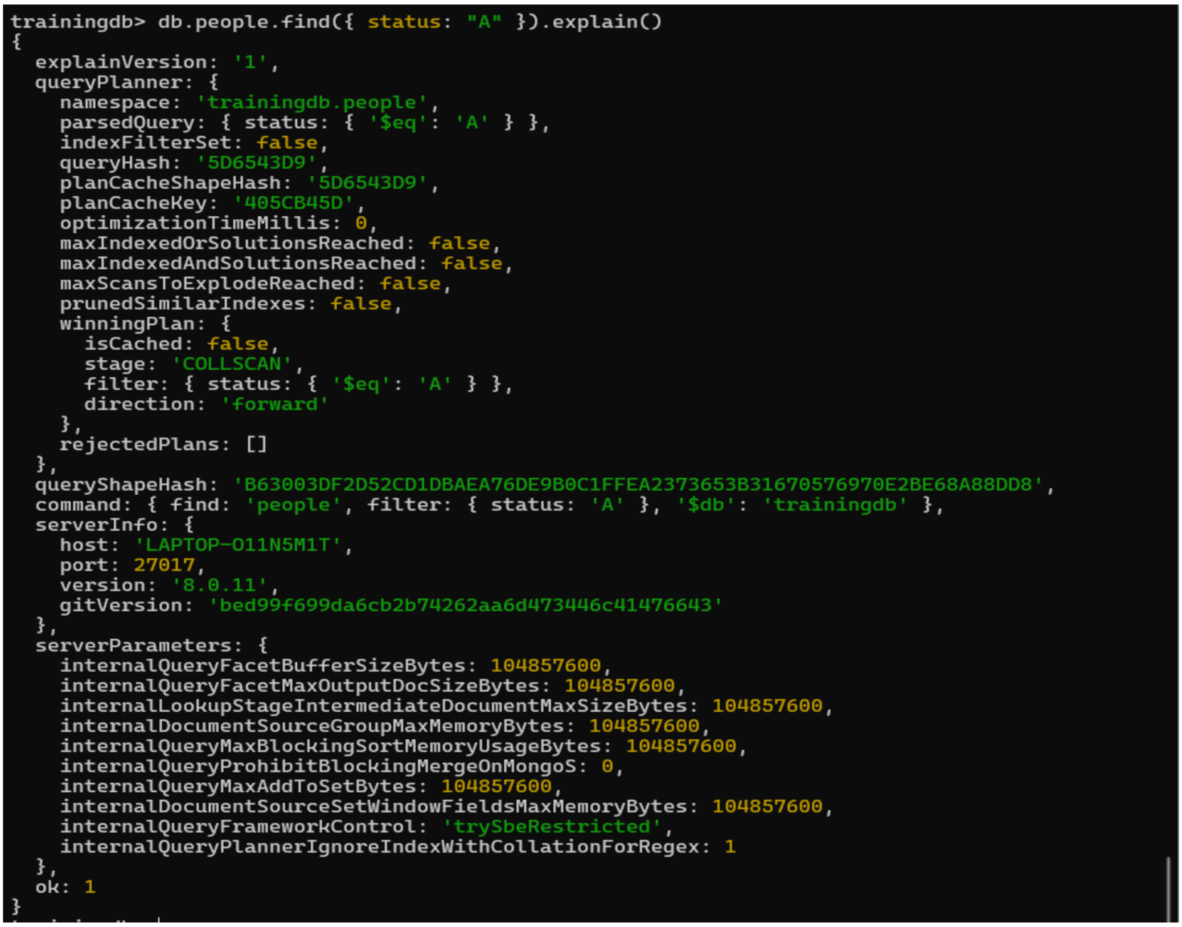
```
trainingdb> db.people.find().limit(1)
[
  {
    _id: ObjectId('687e102bef24f1687beec4ac'),
    user_id: 'bc101',
    status: 'A',
    age: 25
  }
]
```

Query 20: Limit and Skip



```
trainingdb> db.people.find().limit(5).skip(10)
```

Query 21: Explain Query Plan



```
trainingdb> db.people.find({ status: "A" }).explain()
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'trainingdb.people',
    parsedQuery: { status: { '$eq': 'A' } },
    indexFilterSet: false,
    queryHash: '5D6543D9',
    planCacheShapeHash: '5D6543D9',
    planCacheKey: '405CB45D',
    optimizationTimeMillis: 0,
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    prunedSimilarIndexes: false,
    winningPlan: {
      isCached: false,
      stage: 'COLLSCAN',
      filter: { status: { '$eq': 'A' } },
      direction: 'forward'
    },
    rejectedPlans: []
  },
  queryShapeHash: 'B63003DF2D52CD1DBAEA76DE9B0C1FFEA2373653B31670576970E2BE68A88DD8',
  command: { find: 'people', filter: { status: 'A' }, '$db': 'trainingdb' },
  serverInfo: {
    host: 'LAPTOP-O11N5M1T',
    port: 27017,
    version: '8.0.11',
    gitVersion: 'bed99f699da6cb2b74262aa6d473446c41476643'
  },
  serverParameters: {
    internalQueryFacetBufferSizeBytes: 104857600,
    internalQueryFacetMaxOutputDocSizeBytes: 104857600,
    internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,
    internalDocumentSourceGroupMaxMemoryBytes: 104857600,
    internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,
    internalQueryProhibitBlockingMergeOnMongoS: 0,
    internalQueryMaxAddToSetBytes: 104857600,
    internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,
    internalQueryFrameworkControl: 'trySbeRestricted',
    internalQueryPlannerIgnoreIndexWithCollationForRegex: 1
  },
  ok: 1
}
```