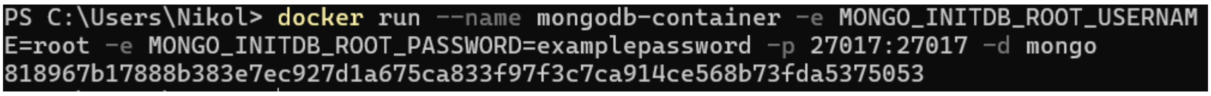
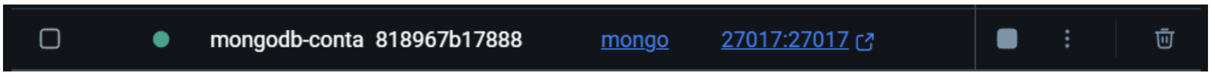
Setup:

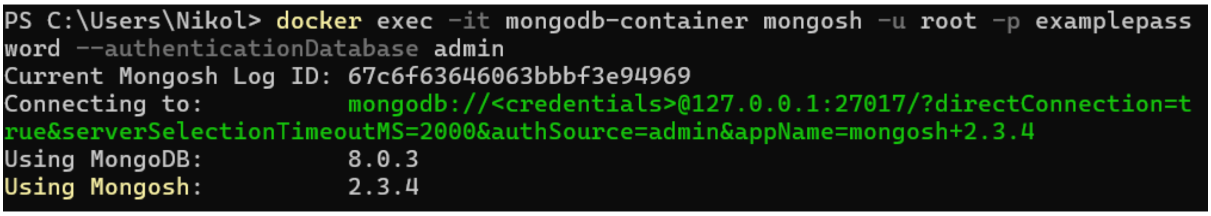


```
PS C:\Users\Nikol> docker run --name mongodb-container -e MONGO_INITDB_ROOT_USERNAM
E=root -e MONGO_INITDB_ROOT_PASSWORD=examplepassword -p 27017:27017 -d mongo
818967b17888b383e7ec927d1a675ca833f97f3c7ca914ce568b73fda5375053
```



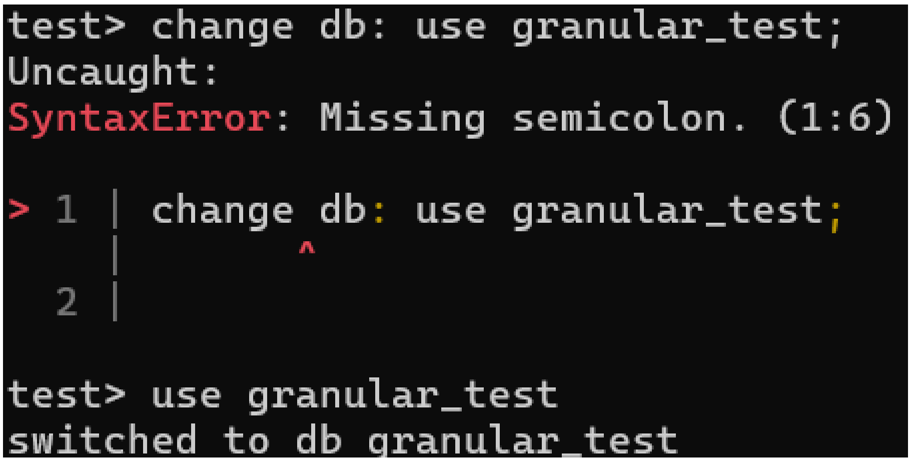
```
□   ●   mongodb-conta  818967b17888   mongo   27017:27017
```

Connection to DB with admin account:



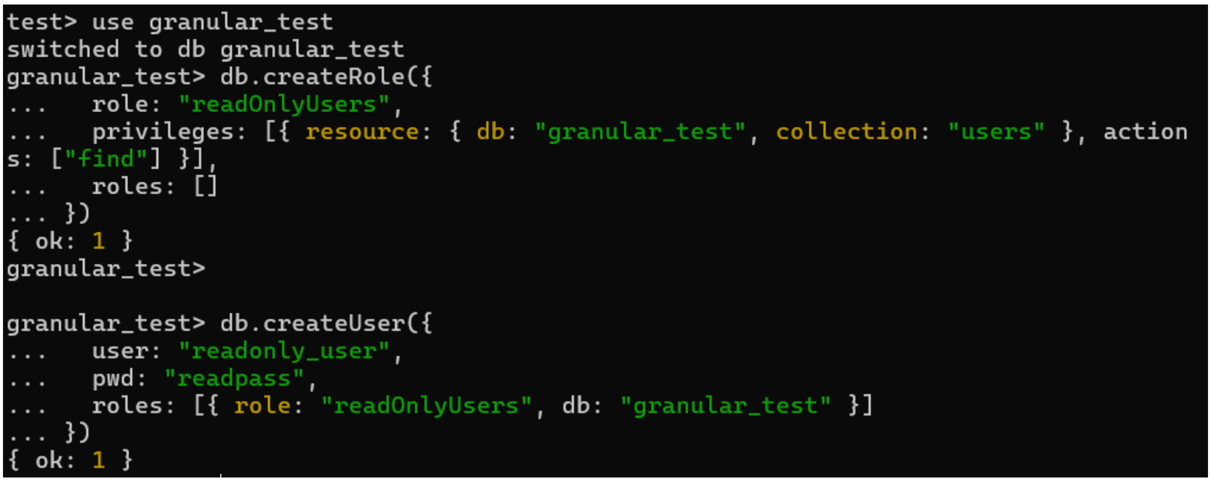
```
PS C:\Users\Nikol> docker exec -it mongodb-container mongosh -u root -p examplepass
word --authenticationDatabase admin
Current Mongosh Log ID: 67c6f63646063bbbf3e94969
Connecting to:          mongodb://<credentials>@127.0.0.1:27017/?directConnection=t
rue&serverSelectionTimeoutMS=2000&authSource=admin&appName=mongosh+2.3.4
Using MongoDB:          8.0.3
Using Mongosh:          2.3.4
```

Setup for DB content, commands unclear:



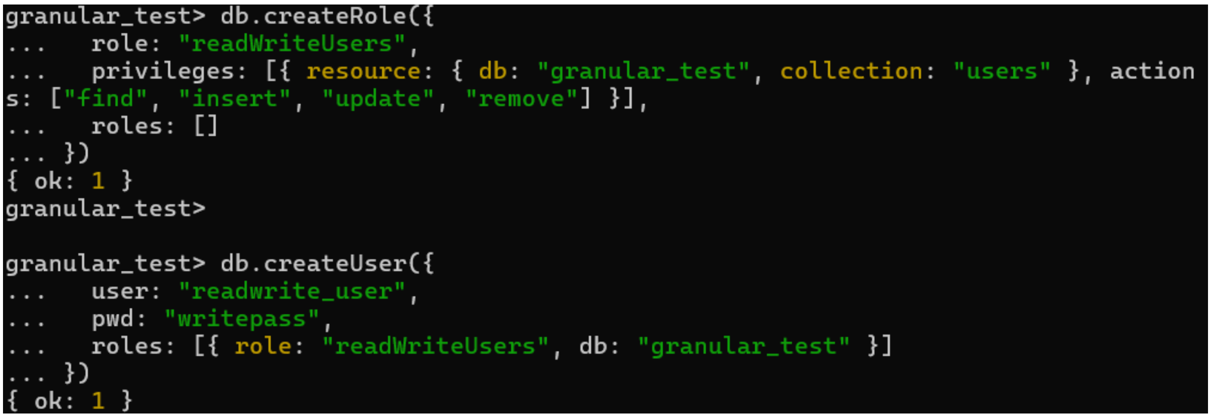
```
test> change db: use granular_test;
Uncaught:
SyntaxError: Missing semicolon. (1:6)

> 1 | change db: use granular_test;
    |       ^
  2 |

test> use granular_test
switched to db granular_test
```

Create Read-Only Role:



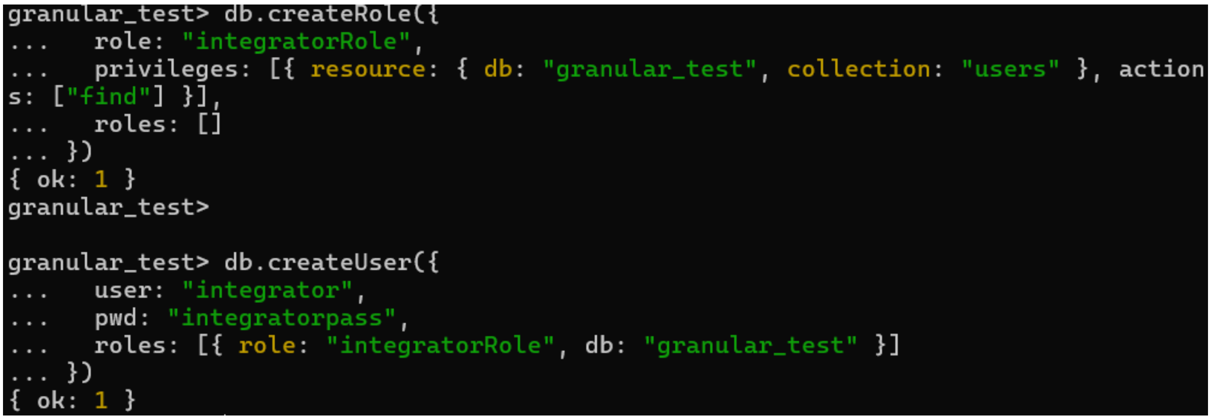
```
test> use granular_test
switched to db granular_test
granular_test> db.createRole({
...    role: "readOnlyUsers",
...    privileges: [{ resource: { db: "granular_test", collection: "users" }, action
s: ["find"] }],
...    roles: []
... })
{ ok: 1 }
granular_test>

granular_test> db.createUser({
...    user: "readonly_user",
...    pwd: "readpass",
...    roles: [{ role: "readOnlyUsers", db: "granular_test" }]
... })
{ ok: 1 }
```

Create Read-Write Role:



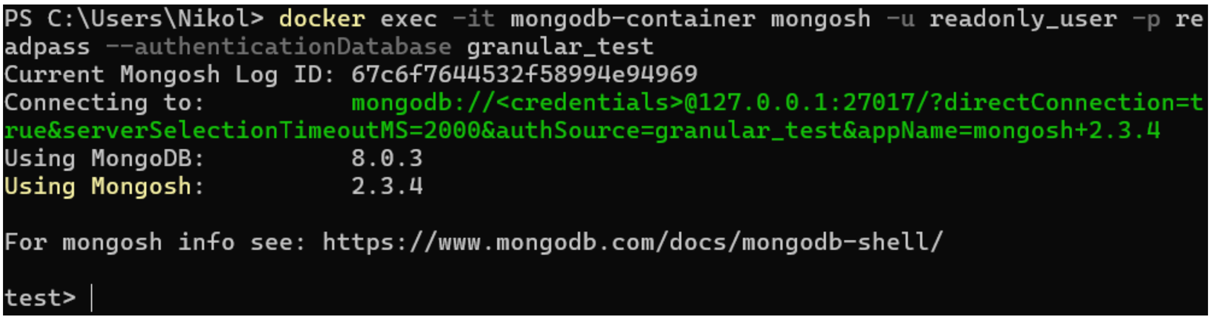
```
granular_test> db.createRole({
...   role: "readWriteUsers",
...   privileges: [{ resource: { db: "granular_test", collection: "users" }, action
s: ["find", "insert", "update", "remove"] }],
...   roles: []
... })
{ ok: 1 }
granular_test>

granular_test> db.createUser({
...   user: "readwrite_user",
...   pwd: "writepass",
...   roles: [{ role: "readWriteUsers", db: "granular_test" }]
... })
{ ok: 1 }
```

Create Integrator Role:

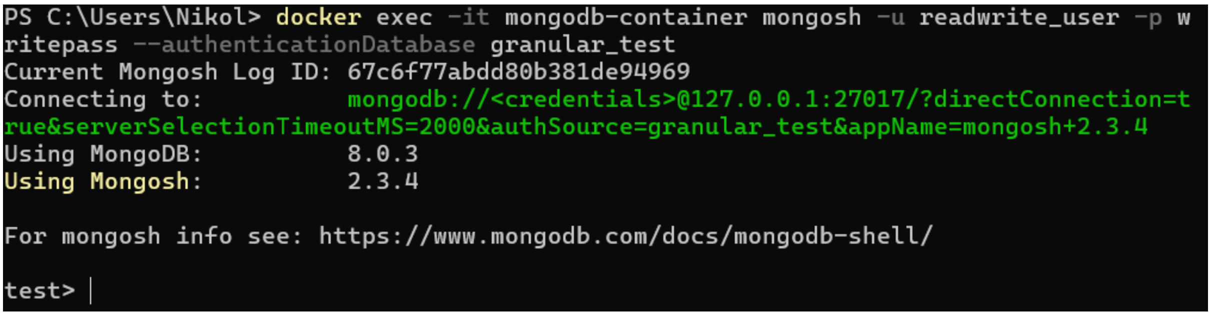


```
granular_test> db.createRole({
...   role: "integratorRole",
...   privileges: [{ resource: { db: "granular_test", collection: "users" }, action
s: ["find"] }],
...   roles: []
... })
{ ok: 1 }
granular_test>

granular_test> db.createUser({
...   user: "integrator",
...   pwd: "integratorpass",
...   roles: [{ role: "integratorRole", db: "granular_test" }]
... })
{ ok: 1 }
```

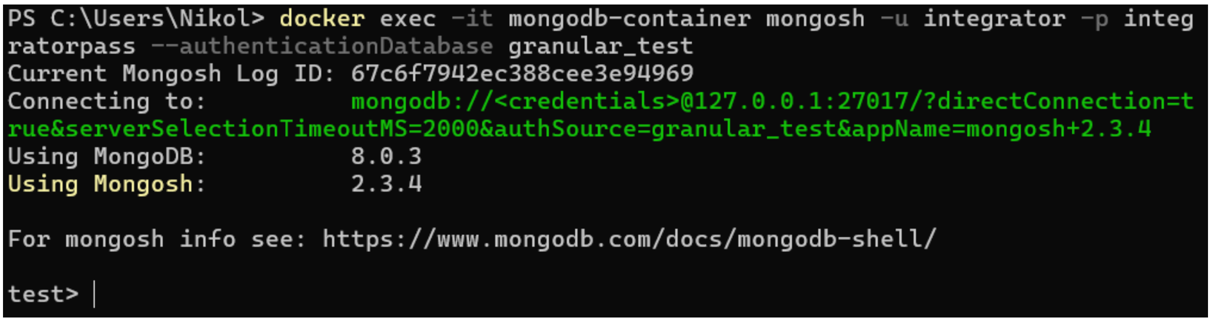
Testing Login:



```
PS C:\Users\Nikol> docker exec -it mongodb-container mongosh -u readonly_user -p re
adpass --authenticationDatabase granular_test
Current Mongosh Log ID: 67c6f7644532f58994e94969
Connecting to:          mongodb://<credentials>@127.0.0.1:27017/?directConnection=t
rue&serverSelectionTimeoutMS=2000&authSource=granular_test&appName=mongosh+2.3.4
Using MongoDB:          8.0.3
Using Mongosh:          2.3.4

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

test> 
```



```
PS C:\Users\Nikol> docker exec -it mongodb-container mongosh -u readwrite_user -p w
ritepass --authenticationDatabase granular_test
Current Mongosh Log ID: 67c6f77abdd80b381de94969
Connecting to:          mongodb://<credentials>@127.0.0.1:27017/?directConnection=t
rue&serverSelectionTimeoutMS=2000&authSource=granular_test&appName=mongosh+2.3.4
Using MongoDB:          8.0.3
Using Mongosh:          2.3.4

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

test> 
```



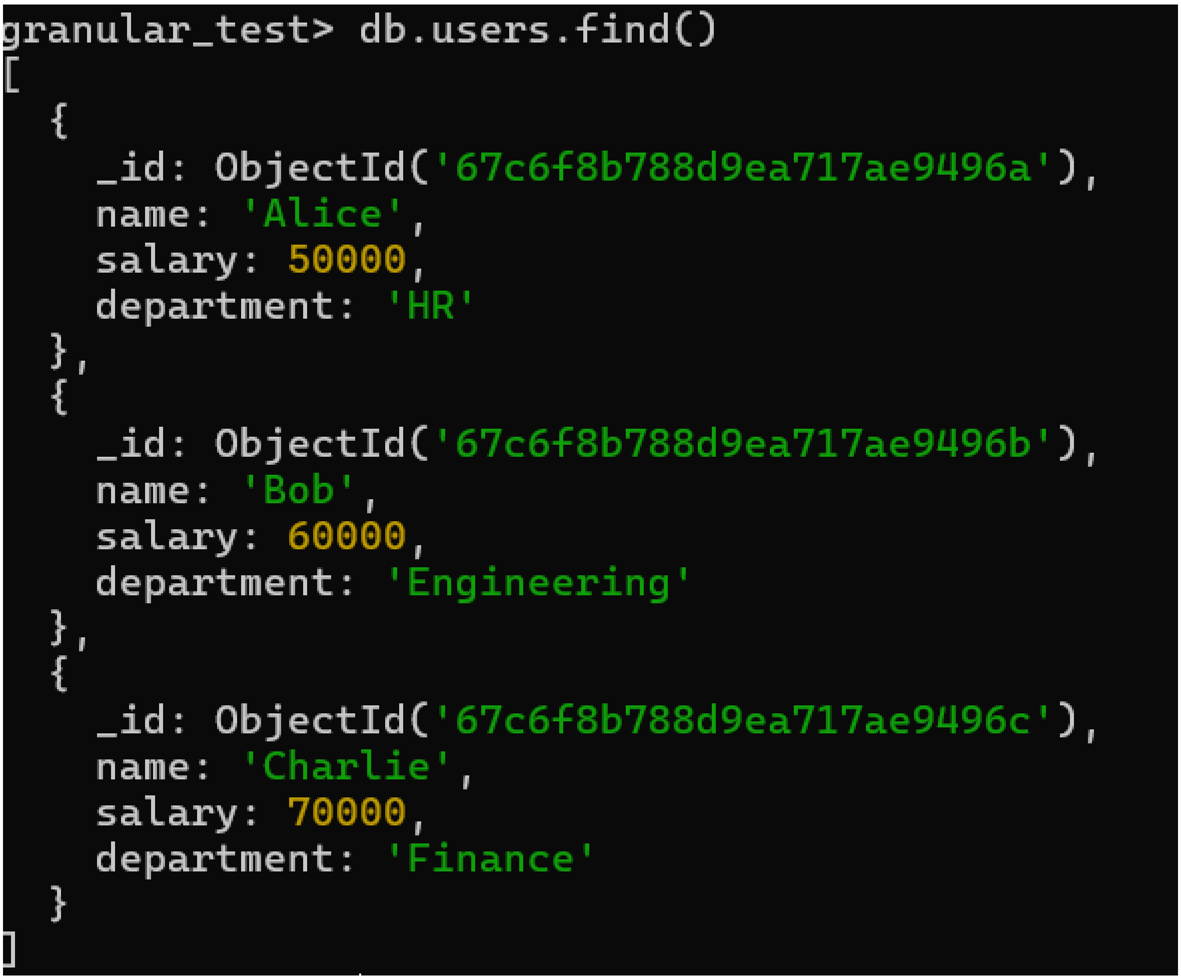
```
PS C:\Users\Nikol> docker exec -it mongodb-container mongosh -u integrator -p integ
ratorpass --authenticationDatabase granular_test
Current Mongosh Log ID: 67c6f7942ec388cee3e94969
Connecting to:          mongodb://<credentials>@127.0.0.1:27017/?directConnection=t
rue&serverSelectionTimeoutMS=2000&authSource=granular_test&appName=mongosh+2.3.4
Using MongoDB:          8.0.3
Using Mongosh:          2.3.4

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

test>
```

Read-Only Tests:

Test 1:

Expected Result: Only `id` and `name` fields visible.

Actual Result:

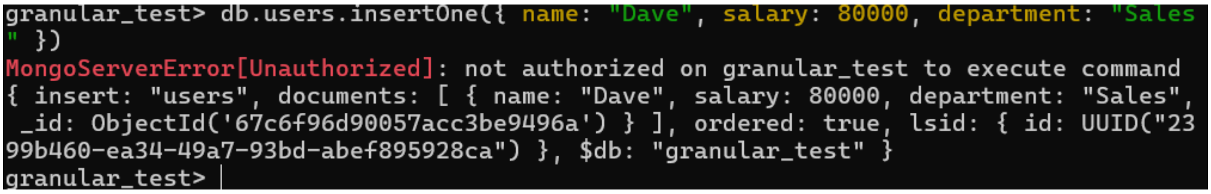


```
granular_test> db.users.find()
[
  {
    _id: ObjectId('67c6f8b788d9ea717ae9496a'),
    name: 'Alice',
    salary: 50000,
    department: 'HR'
  },
  {
    _id: ObjectId('67c6f8b788d9ea717ae9496b'),
    name: 'Bob',
    salary: 60000,
    department: 'Engineering'
  },
  {
    _id: ObjectId('67c6f8b788d9ea717ae9496c'),
    name: 'Charlie',
    salary: 70000,
    department: 'Finance'
  }
]
```

Test 2:

Expected: Unauthorized

Result:



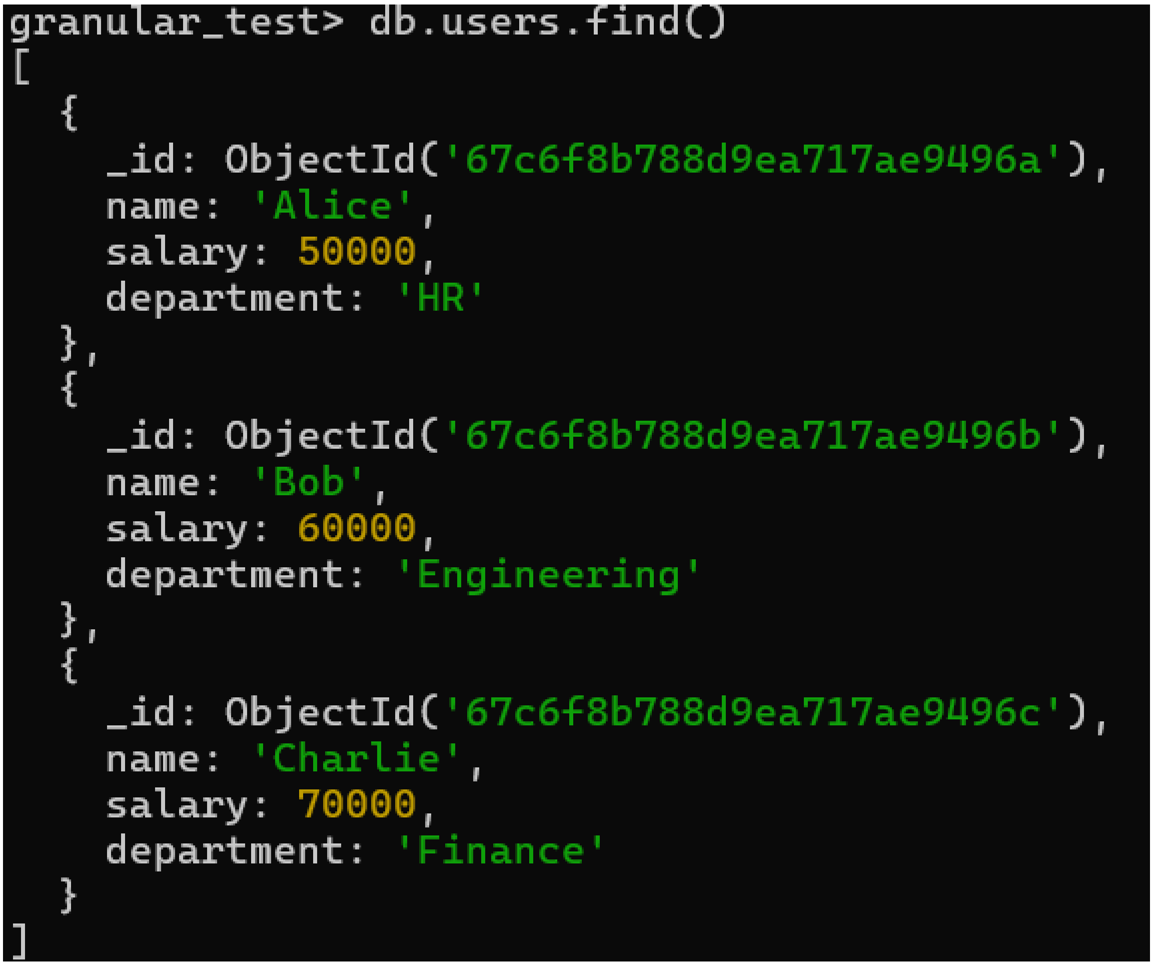
```
granular_test> db.users.insertOne({ name: "Dave", salary: 80000, department: "Sales
" })
MongoServerError[Unauthorized]: not authorized on granular_test to execute command
{ insert: "users", documents: [ { name: "Dave", salary: 80000, department: "Sales",
 _id: ObjectId('67c6f96d90057acc3be9496a') } ], ordered: true, lsid: { id: UUID("23
99b460-ea34-49a7-93bd-abef895928ca") }, $db: "granular_test" }
granular_test> 
```

Read-Write Tests:

Test 1:

Expected Result: All fields visible.

Actual Result:

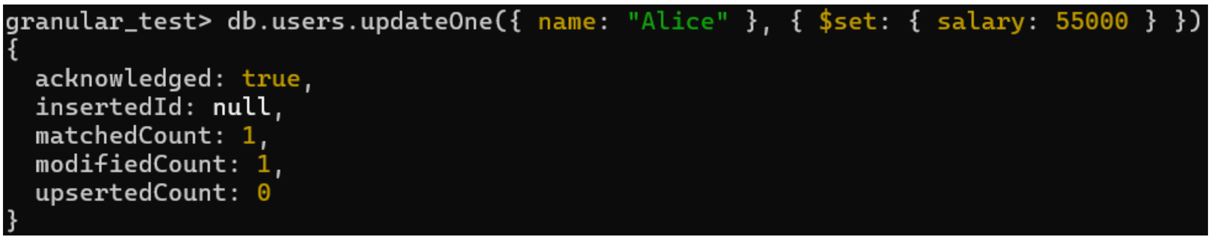


```
granular_test> db.users.find()
[
  {
    _id: ObjectId('67c6f8b788d9ea717ae9496a'),
    name: 'Alice',
    salary: 50000,
    department: 'HR'
  },
  {
    _id: ObjectId('67c6f8b788d9ea717ae9496b'),
    name: 'Bob',
    salary: 60000,
    department: 'Engineering'
  },
  {
    _id: ObjectId('67c6f8b788d9ea717ae9496c'),
    name: 'Charlie',
    salary: 70000,
    department: 'Finance'
  }
]
```

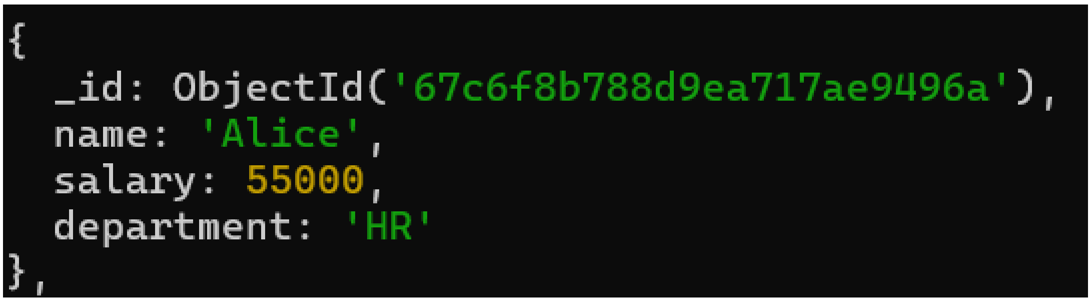
Test 2:

Expected Result: Salary updated successfully.

Actual Result:



```
granular_test> db.users.updateOne({ name: "Alice" }, { $set: { salary: 55000 } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



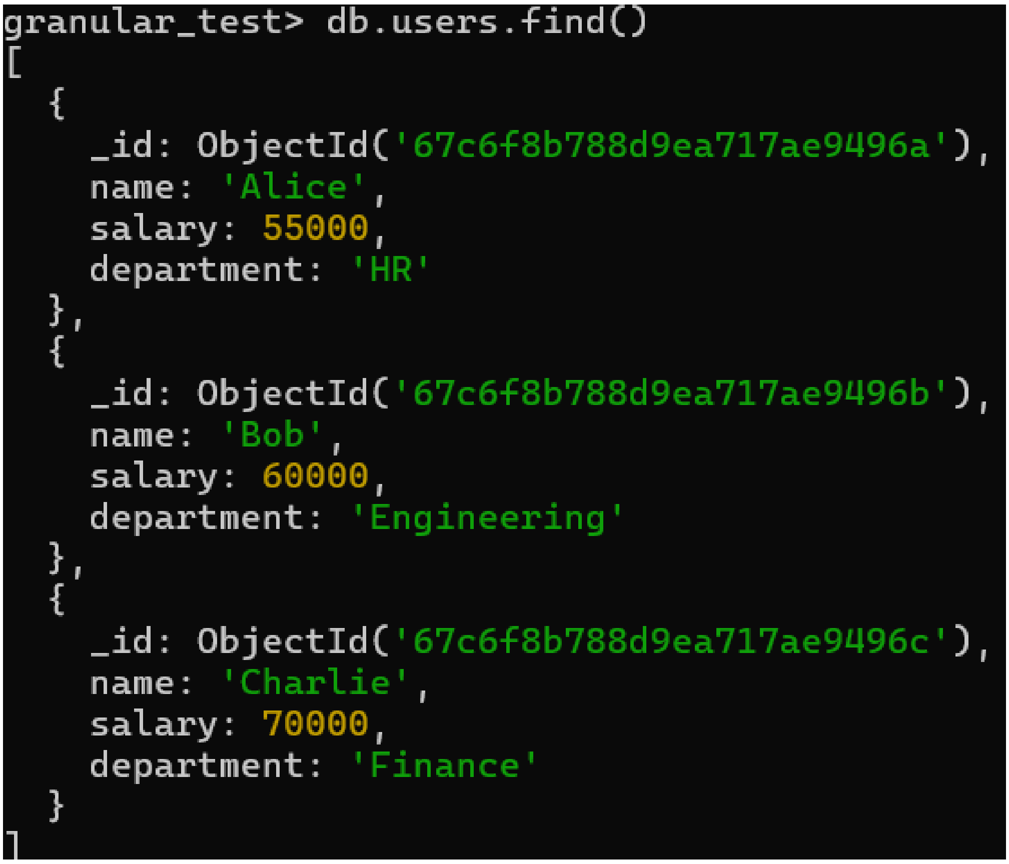
```
{
  _id: ObjectId('67c6f8b788d9ea717ae9496a'),
  name: 'Alice',
  salary: 55000,
  department: 'HR'
},
```

Integrator Tests:

Test 1:

Expected Result: Only `id` and `department` fields for **Engineering** department rows visible.

Actual Result:

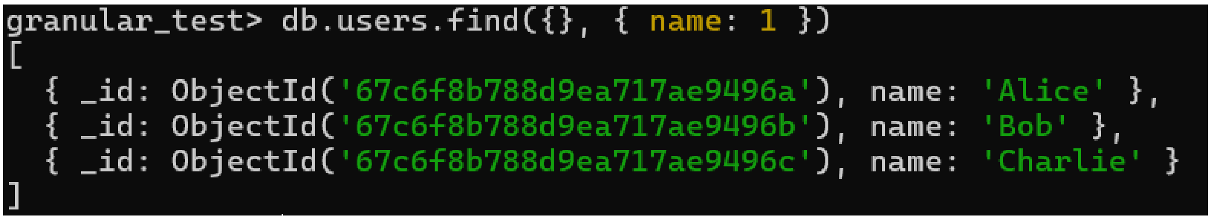


```
granular_test> db.users.find()
[
  {
    _id: ObjectId('67c6f8b788d9ea717ae9496a'),
    name: 'Alice',
    salary: 55000,
    department: 'HR'
  },
  {
    _id: ObjectId('67c6f8b788d9ea717ae9496b'),
    name: 'Bob',
    salary: 60000,
    department: 'Engineering'
  },
  {
    _id: ObjectId('67c6f8b788d9ea717ae9496c'),
    name: 'Charlie',
    salary: 70000,
    department: 'Finance'
  }
]
```

Test 2:

Expected Error: **Unauthorized access to name field**

Actual Result:

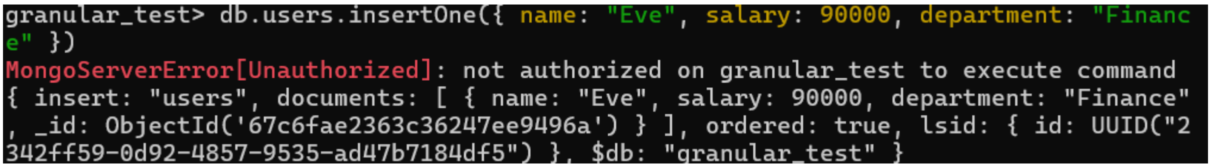


```
granular_test> db.users.find({}, { name: 1 })
[
  { _id: ObjectId('67c6f8b788d9ea717ae9496a'), name: 'Alice' },
  { _id: ObjectId('67c6f8b788d9ea717ae9496b'), name: 'Bob' },
  { _id: ObjectId('67c6f8b788d9ea717ae9496c'), name: 'Charlie' }
]
```

Test 3:

Expected Error: **Unauthorized**

Actual Result:



```
granular_test> db.users.insertOne({ name: "Eve", salary: 90000, department: "Financ
e" })
MongoServerError[Unauthorized]: not authorized on granular_test to execute command
{ insert: "users", documents: [ { name: "Eve", salary: 90000, department: "Finance"
, _id: ObjectId('67c6fae2363c36247ee9496a') } ], ordered: true, lsid: { id: UUID("2
342ff59-0d92-4857-9535-ad47b7184df5") }, $db: "granular_test" }
```