Jordan Mitchell

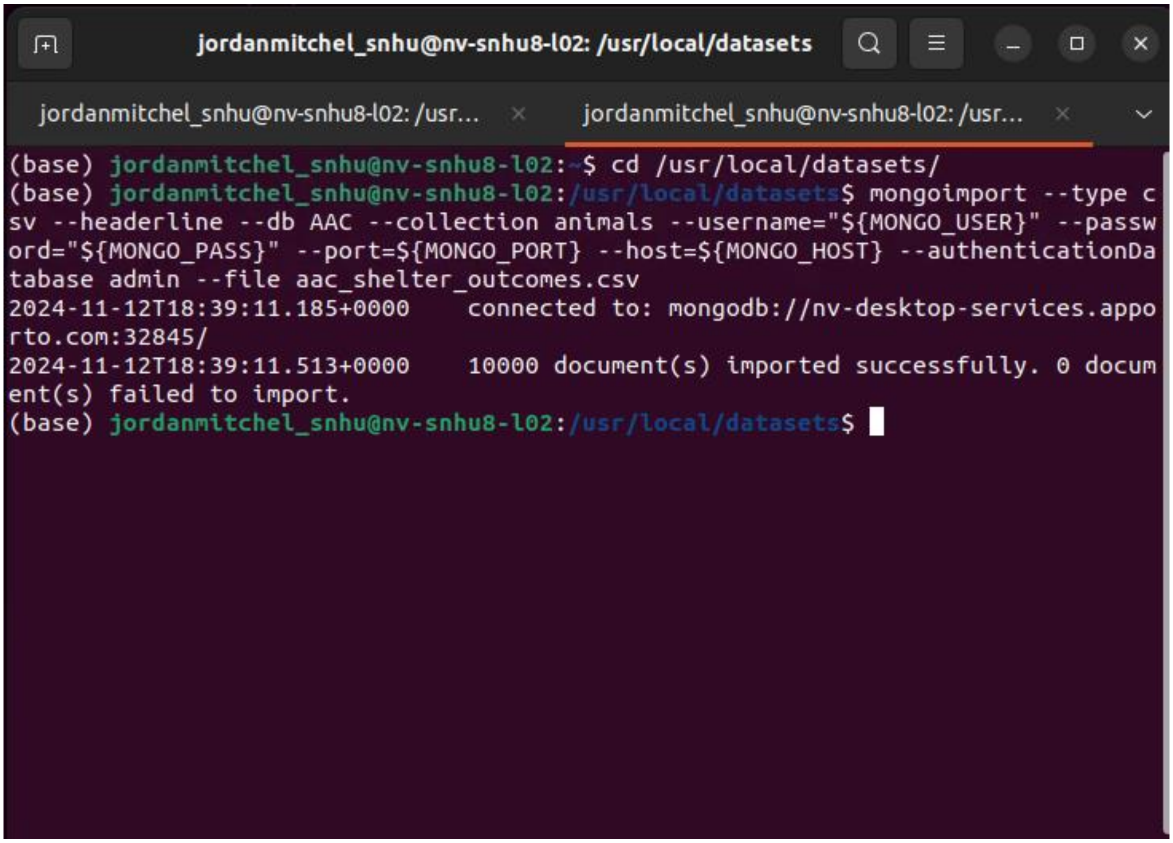
11/10/2024

Professor Morris

Milestone: Database Indexing and Authentication

Part One: Importing and Indexing a Data Set

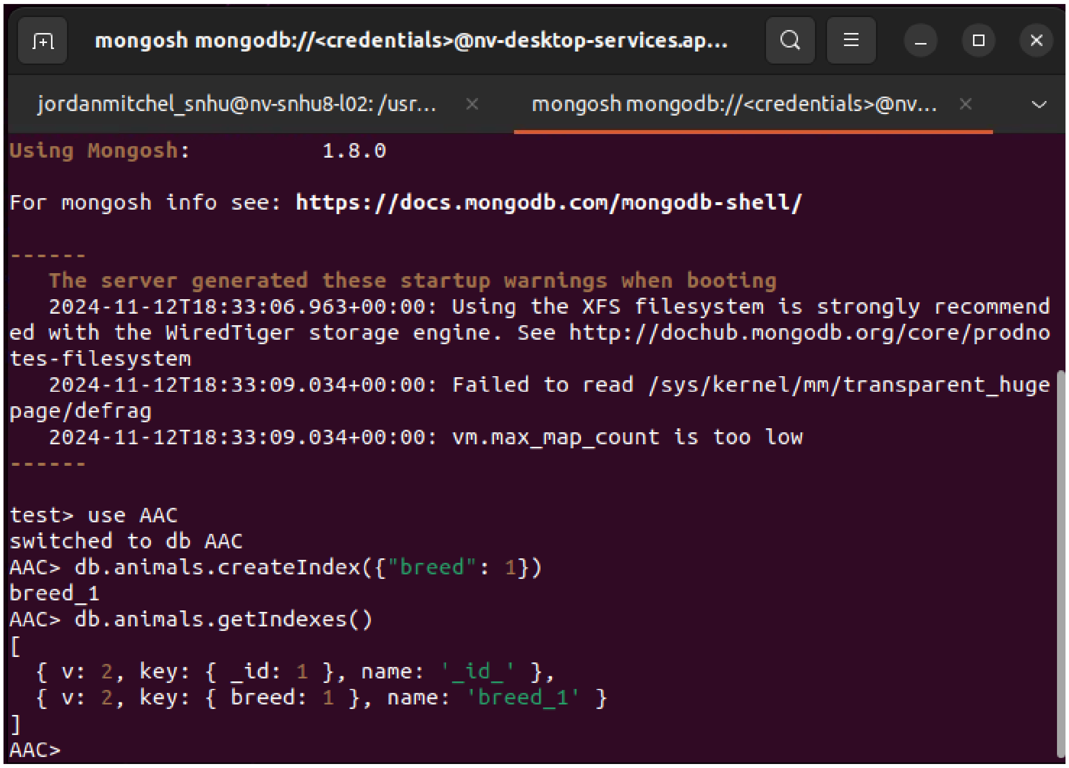
1. In Apporto, open the terminal window to access the Linux shell. Upload the Austin Animal Center (AAC) Outcomes data set into MongoDB by **importing a CSV file using the appropriate MongoDB import tool**. Use the database name “AAC” and collection name “animals.” Complete the import using the mongoimport tool, and **take screenshots** of both the import command and its execution.



jordanmitchel_snhu@nv-snhu8-l02: /usr/local/datasets
jordanmitchel_snhu@nv-snhu8-l02: /usr...
jordanmitchel_snhu@nv-snhu8-l02: /usr...

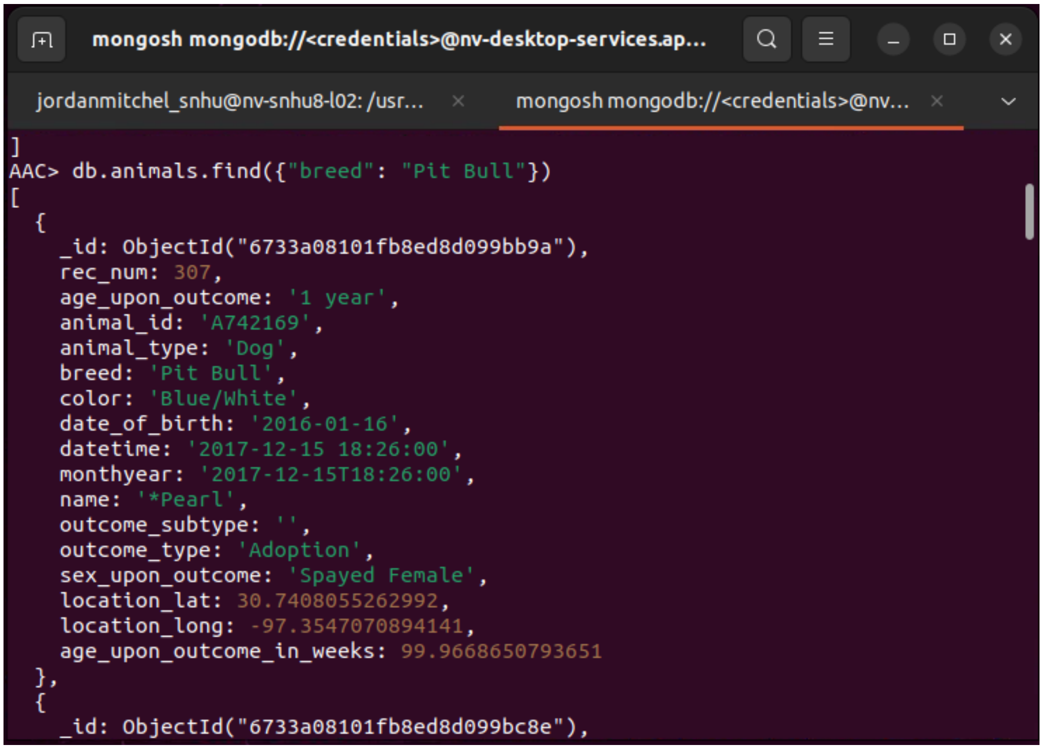
```
(base) jordanmitchel_snhu@nv-snhu8-l02:~$ cd /usr/local/datasets/
(base) jordanmitchel_snhu@nv-snhu8-l02:/usr/local/datasets$ mongoimport --type c
sv --headerline --db AAC --collection animals --username="${MONGO_USER}" --passw
ord="${MONGO_PASS}" --port=${MONGO_PORT} --host=${MONGO_HOST} --authenticationDa
tabase admin --file aac_shelter_outcomes.csv
2024-11-12T18:39:11.185+0000    connected to: mongodb://nv-desktop-services.appo
rto.com:32845/
2024-11-12T18:39:11.513+0000    10000 document(s) imported successfully. 0 docum
ent(s) failed to import.
(base) jordanmitchel_snhu@nv-snhu8-l02:/usr/local/datasets$ 
```

2. After importing your data set, open the mongo shell. **Create a simple index** on the key “breed.” Show an example query using this index, and use the **explain function** to verify that the index will be used. **Take screenshots** of your example query.


mongosh mongodb://<credentials>@nv-desktop-services.ap...
jordanmitchel_snhu@nv-snhu8-l02: /usr...
mongosh mongodb://<credentials>@nv...

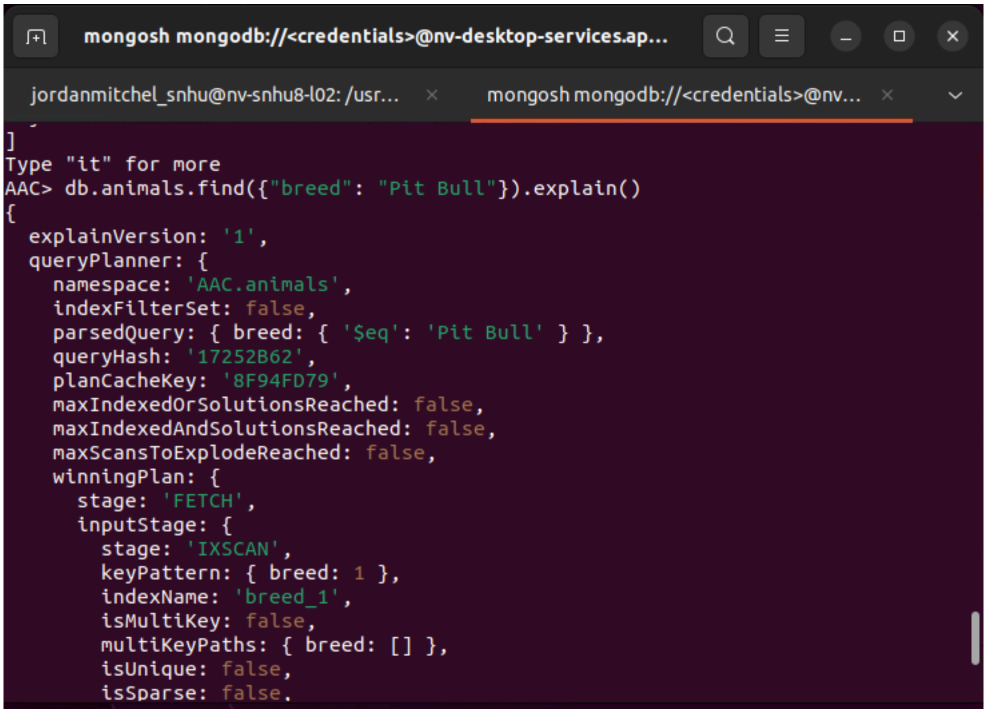
```
Using Mongosh:          1.8.0

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

------
   The server generated these startup warnings when booting
   2024-11-12T18:33:06.963+00:00: Using the XFS filesystem is strongly recommend
ed with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodno
tes-filesystem
   2024-11-12T18:33:09.034+00:00: Failed to read /sys/kernel/mm/transparent_huge
page/defrag
   2024-11-12T18:33:09.034+00:00: vm.max_map_count is too low
------

test> use AAC
switched to db AAC
AAC> db.animals.createIndex({"breed": 1})
breed_1
AAC> db.animals.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { breed: 1 }, name: 'breed_1' }
]
AAC>
```

Pit bulls used for example query


mongosh mongodb://<credentials>@nv-desktop-services.ap...
jordanmitchel_snhu@nv-snhu8-l02: /usr...
mongosh mongodb://<credentials>@nv...

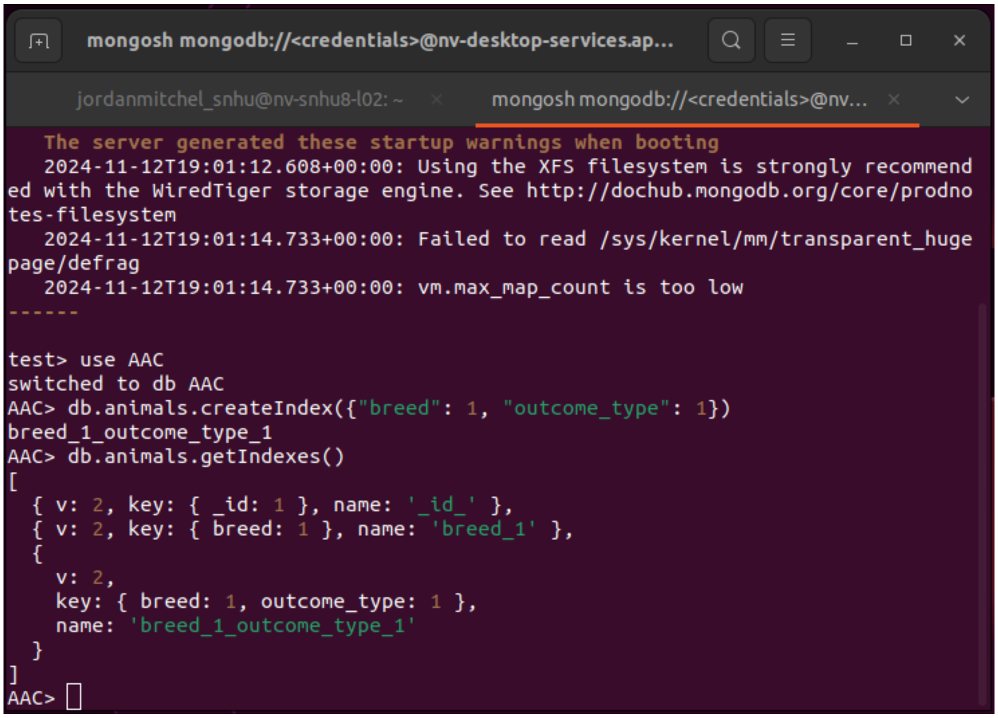
```
]
AAC> db.animals.find({"breed": "Pit Bull"})
[
  {
    _id: ObjectId("6733a08101fb8ed8d099bb9a"),
    rec_num: 307,
    age_upon_outcome: '1 year',
    animal_id: 'A742169',
    animal_type: 'Dog',
    breed: 'Pit Bull',
    color: 'Blue/White',
    date_of_birth: '2016-01-16',
    datetime: '2017-12-15 18:26:00',
    monthyear: '2017-12-15T18:26:00',
    name: '*Pearl',
    outcome_subtype: '',
    outcome_type: 'Adoption',
    sex_upon_outcome: 'Spayed Female',
    location_lat: 30.7408055262992,
    location_long: -97.3547070894141,
    age_upon_outcome_in_weeks: 99.9668650793651
  },
  {
    _id: ObjectId("6733a08101fb8ed8d099bc8e"),
```

explain() function


mongosh mongodb://<credentials>@nv-desktop-services.ap...
jordanmitchel_snhu@nv-snhu8-l02: /usr...
mongosh mongodb://<credentials>@nv...

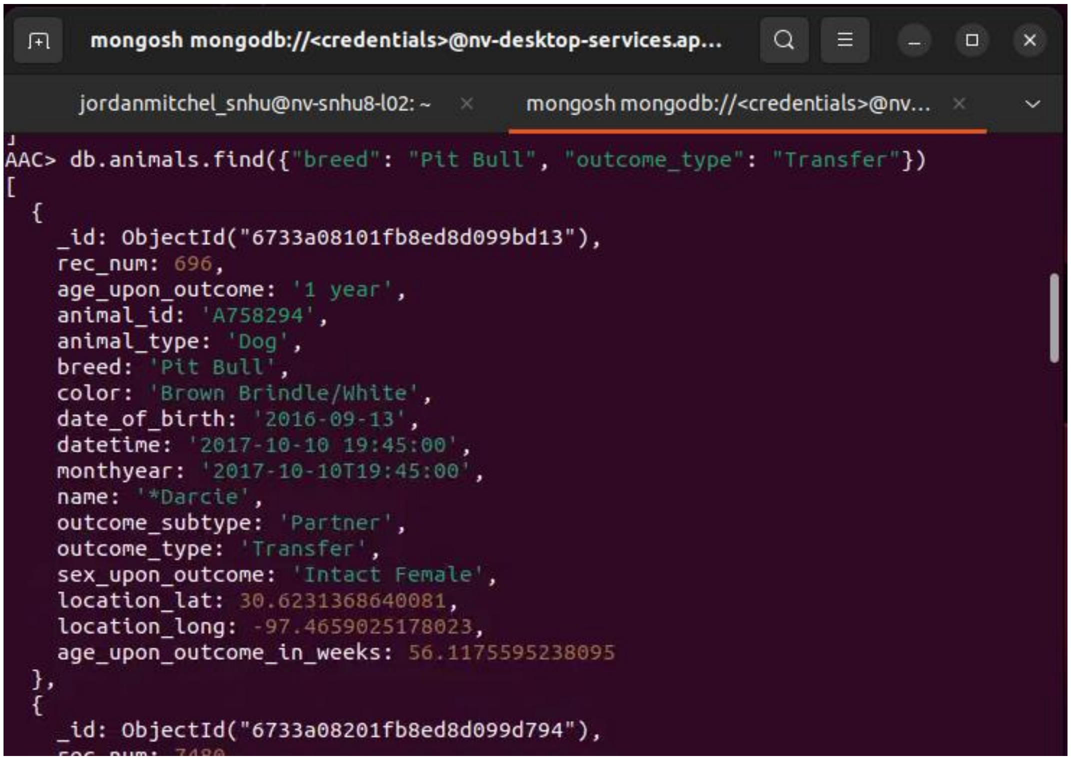
```
]
Type "it" for more
AAC> db.animals.find({"breed": "Pit Bull"}).explain()
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'AAC.animals',
    indexFilterSet: false,
    parsedQuery: { breed: { '$eq': 'Pit Bull' } },
    queryHash: '17252B62',
    planCacheKey: '8F94FD79',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { breed: 1 },
        indexName: 'breed_1',
        isMultiKey: false,
        multiKeyPaths: { breed: [] },
        isUnique: false,
        isSparse: false,
```

3. **Create a compound index** that will improve the performance of queries looking for breeds that have an "outcome_type" of "Transfer." Show an example query using this compound index, and use the **explain function** to confirm the index will be used. **Take screenshots** of your example query.


mongosh mongodb://<credentials>@nv-desktop-services.ap...
jordanmitchel_snhu@nv-snhu8-l02: ~
mongosh mongodb://<credentials>@nv...

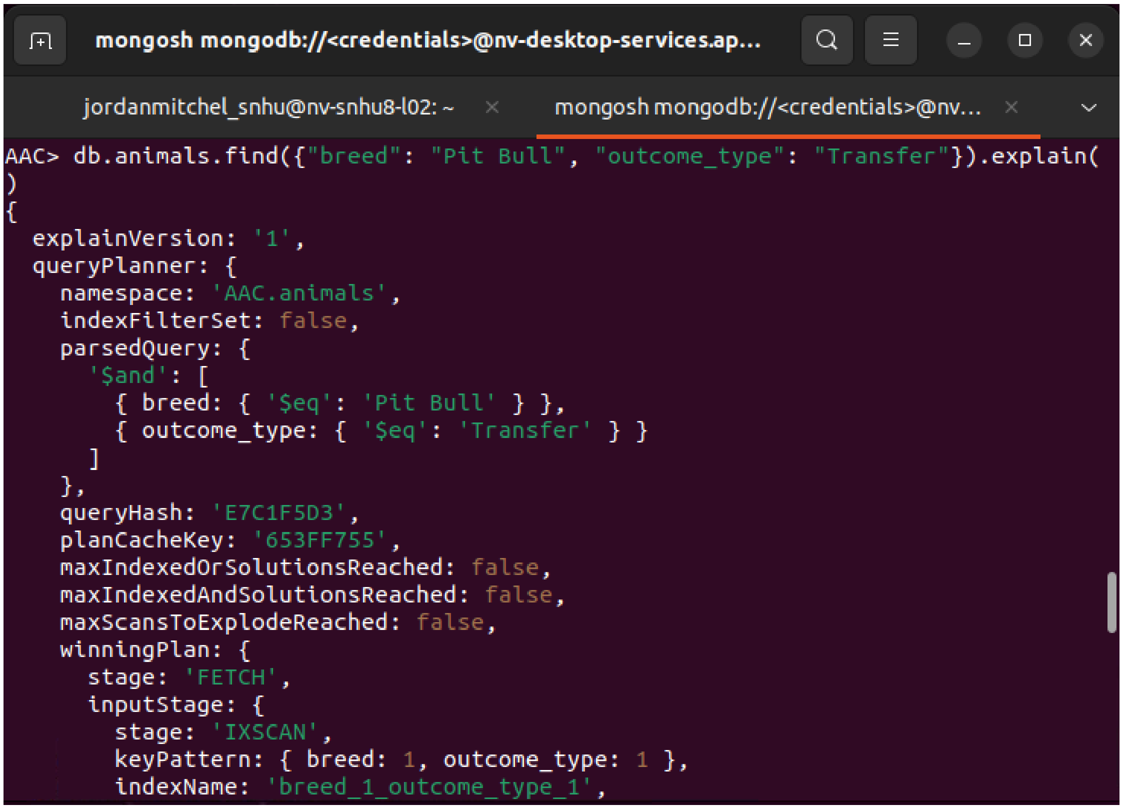
```
   The server generated these startup warnings when booting
   2024-11-12T19:01:12.608+00:00: Using the XFS filesystem is strongly recommend
ed with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodno
tes-filesystem
   2024-11-12T19:01:14.733+00:00: Failed to read /sys/kernel/mm/transparent_huge
page/defrag
   2024-11-12T19:01:14.733+00:00: vm.max_map_count is too low
------

test> use AAC
switched to db AAC
AAC> db.animals.createIndex({"breed": 1, "outcome_type": 1})
breed_1_outcome_type_1
AAC> db.animals.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { breed: 1 }, name: 'breed_1' },
  {
    v: 2,
    key: { breed: 1, outcome_type: 1 },
    name: 'breed_1_outcome_type_1'
  }
]
AAC> 
```

Sample query


mongosh mongodb://<credentials>@nv-desktop-services.ap...
jordanmitchel_snhu@nv-snhu8-l02: ~
mongosh mongodb://<credentials>@nv...

```
]
AAC> db.animals.find({"breed": "Pit Bull", "outcome_type": "Transfer"})
[
  {
    _id: ObjectId("6733a08101fb8ed8d099bd13"),
    rec_num: 696,
    age_upon_outcome: '1 year',
    animal_id: 'A758294',
    animal_type: 'Dog',
    breed: 'Pit Bull',
    color: 'Brown Brindle/White',
    date_of_birth: '2016-09-13',
    datetime: '2017-10-10 19:45:00',
    monthyear: '2017-10-10T19:45:00',
    name: '*Darcie',
    outcome_subtype: 'Partner',
    outcome_type: 'Transfer',
    sex_upon_outcome: 'Intact Female',
    location_lat: 30.6231368640081,
    location_long: -97.4659025178023,
    age_upon_outcome_in_weeks: 56.1175595238095
  },
  {
    _id: ObjectId("6733a08201fb8ed8d099d794"),
    rec_num: 7480
```

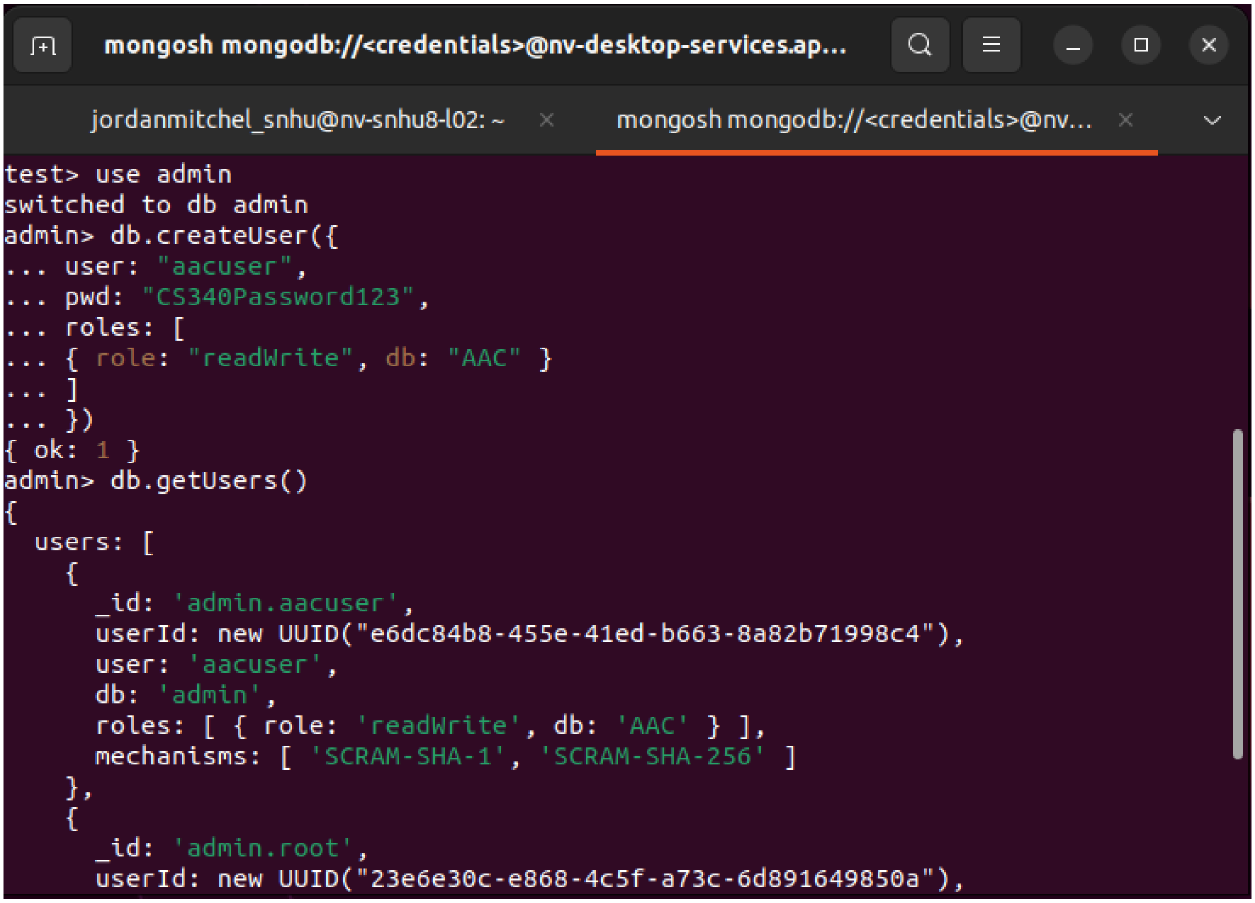
explain() function


mongosh mongodb://<credentials>@nv-desktop-services.ap...
jordanmitchel_snhu@nv-snhu8-l02: ~
mongosh mongodb://<credentials>@nv...

```
AAC> db.animals.find({"breed": "Pit Bull", "outcome_type": "Transfer"}).explain(
)
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'AAC.animals',
    indexFilterSet: false,
    parsedQuery: {
      '$and': [
        { breed: { '$eq': 'Pit Bull' } },
        { outcome_type: { '$eq': 'Transfer' } }
      ]
    },
    queryHash: 'E7C1F5D3',
    planCacheKey: '653FF755',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { breed: 1, outcome_type: 1 },
        indexName: 'breed_1_outcome_type_1',
```

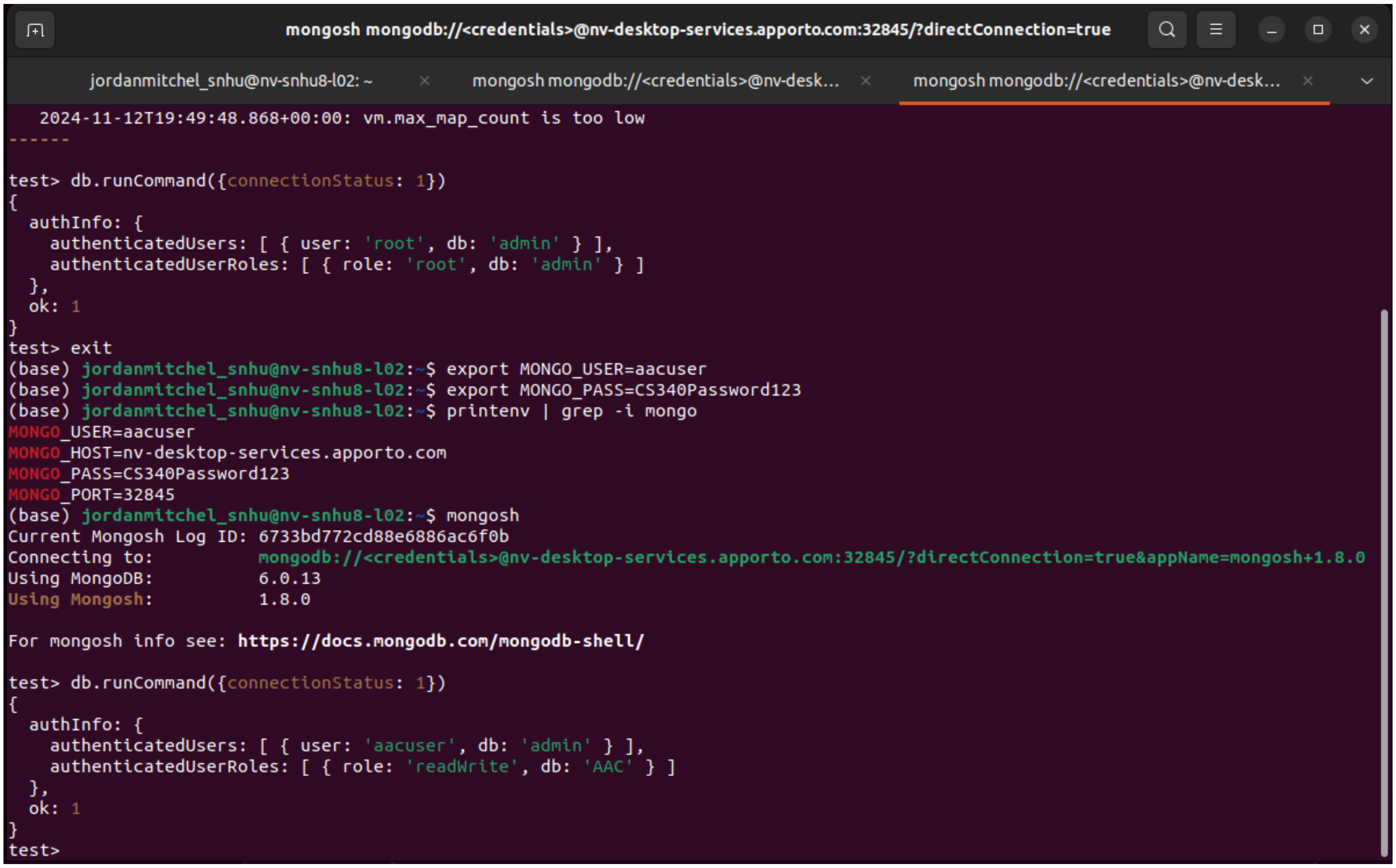
Part Two: User Authentication

1. **Create a new user account called "aacuser"** for the database AAC in the mongo shell


mongosh mongodb://<credentials>@nv-desktop-services.ap...
jordanmitchel_snhu@nv-snhu8-l02: ~
mongosh mongodb://<credentials>@nv...

```
test> use admin
switched to db admin
admin> db.createUser({
... user: "aacuser",
... pwd: "CS340Password123",
... roles: [
... { role: "readWrite", db: "AAC" }
... ]
... })
{ ok: 1 }
admin> db.getUsers()
{
  users: [
    {
      _id: 'admin.aacuser',
      userId: new UUID("e6dc84b8-455e-41ed-b663-8a82b71998c4"),
      user: 'aacuser',
      db: 'admin',
      roles: [ { role: 'readWrite', db: 'AAC' } ],
      mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
    },
    {
      _id: 'admin.root',
      userId: new UUID("23e6e30c-e868-4c5f-a73c-6d891649850a"),
```

2. Take a screenshot of your login process to MongoDB using the mongo shell. Be sure you can access MongoDB and list the databases using both the admin and aacuser accounts. This task will verify that your accounts are working. You should be able to include the login commands for both accounts in one screenshot, but if you cannot, include two screenshots to show both login commands.



mongosh mongodb://<credentials>@nv-desktop-services.apporto.com:32845/?directConnection=true
jordanmitchel_snhu@nv-snhu8-l02: ~
mongosh mongodb://<credentials>@nv-desk...
mongosh mongodb://<credentials>@nv-desk...

```
   2024-11-12T19:49:48.868+00:00: vm.max_map_count is too low
------

test> db.runCommand({connectionStatus: 1})
{
  authInfo: {
    authenticatedUsers: [ { user: 'root', db: 'admin' } ],
    authenticatedUserRoles: [ { role: 'root', db: 'admin' } ]
  },
  ok: 1
}
test> exit
(base) jordanmitchel_snhu@nv-snhu8-l02:~$ export MONGO_USER=aacuser
(base) jordanmitchel_snhu@nv-snhu8-l02:~$ export MONGO_PASS=CS340Password123
(base) jordanmitchel_snhu@nv-snhu8-l02:~$ printenv | grep -i mongo
MONGO_USER=aacuser
MONGO_HOST=nv-desktop-services.apporto.com
MONGO_PASS=CS340Password123
MONGO_PORT=32845
(base) jordanmitchel_snhu@nv-snhu8-l02:~$ mongosh
Current Mongosh Log ID: 6733bd772cd88e6886ac6f0b
Connecting to:          mongodb://<credentials>@nv-desktop-services.apporto.com:32845/?directConnection=true&appName=mongosh+1.8.0
Using MongoDB:          6.0.13
Using Mongosh:          1.8.0

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

test> db.runCommand({connectionStatus: 1})
{
  authInfo: {
    authenticatedUsers: [ { user: 'aacuser', db: 'admin' } ],
    authenticatedUserRoles: [ { role: 'readWrite', db: 'AAC' } ]
  },
  ok: 1
}
test>
```