

DS108-01-12 - NoSQL - Lesson 1

Practice Hands-On

Requirements

This Hands-On is structured into *two* parts, and each part may ask you to run multiple queries. After each query, please take a screenshot and add it to a text document (or an equivalent) and name this file NoSQL-HandsOn1. This way, you will be able to submit your answers to each part all at once. You will continue working with the appusers collection in Atlas that you created during the lesson. Good luck!

[!caution] Caution! Be sure to take screenshots of each of your queries.

[!info] Info Be sure to zip and submit your NoSQL-HandsOn1 text document when finished! You will not be able to re-submit, so be sure the screenshots to each part are located within this document.

Part 1

1.1 Create 3 more users of your choosing

- Create three more users of your choosing
- Using the `insertOne()` query, and add them to the `appusers` collection.
- Include at least the `firstName`, `lastName`, and `age` fields.
- Feel free to also include any other fields you think would be useful data for an app user.

[!caution] Caution! Please take a screenshot of each query you run (since you are inserting each user one-by-one using `insertOne()`, you will run at least three queries) and save it to your NoSQL–HandsOn1 text document.

```
db.appusers.insertOne({lastName:"Walker", firstName:"Heather",  
middleName:"Lee", age:37, favoriteColor:"blue"})
```



The screenshot shows the MongoDB Playground interface. The command tab contains the query: `db.createCollection("appusers");` followed by `db.appusers.insertOne({lastName:"Walker", firstName:"Heather", middleName:"Lee", age:37, favoriteColor:"blue"})`. The result tab shows the output: `{ "acknowledged": true, "insertedId": "6315088f3434ec78dcdcd" }`.

```
db.appusers.insertOne({lastName:"Walker", firstName:"Cinder",  
middleName:"SweetBoi", age:6, breed: "Standard Poodle",  
favoriteFood:"lamb"})
```



The screenshot shows the MongoDB Playground interface. The command tab contains the query: `db.appusers.insertOne({lastName:"Walker", firstName:"Heather", middleName:"Lee", age:37, favoriteColor:"blue"})` followed by `db.appusers.insertOne({lastName:"Walker", firstName:"Cinder", middleName:"SweetBoi", age:6, breed: "Standard Poodle", favoriteFood:"lamb"})`. The result tab shows the output: `{ "acknowledged": true, "insertedId": "631501498427d645d243a6d" }`.

```
db.appusers.insertOne({lastName:"Walker", firstName:"Jarrod",  
middleName:"Thomas", age:39, favoriteFood:"burgers"})
```



The screenshot shows the MongoDB Playground interface. The command tab contains the query: `db.appusers.insertOne({lastName:"Walker", firstName:"Heather", middleName:"Lee", age:37, favoriteColor:"blue"})` followed by `db.appusers.insertOne({lastName:"Walker", firstName:"Cinder", middleName:"SweetBoi", age:6, breed: "Standard Poodle", favoriteFood:"lamb"})` followed by `db.appusers.insertOne({lastName:"Walker", firstName:"Jarrod", middleName:"Thomas", age:39, favoriteFood:"burgers"})`. The result tab shows the output: `{ "acknowledged": true, "insertedId": "631501955796be56b72a531" }`.

1.2 Run a basic find() query

- Next, run a basic find() query to see all of your documents within your database through the Mongo Shell.

 Playground Result 

Include Import Statements | Include Driver Syntax

```
1  [
2  {
3      "_id": 1,
4      "lastName": "Pumpernickel",
5      "firstName": "Georgina",
6      "middleName": "Sasquatch",
7      "age": 27,
8      "gender": "f"
9  },
10 {
11     "_id": 2,
12     "lastName": "Wilson",
13     "firstName": "Coderboy",
14     "age": 18,
15     "favoriteColor": "blue"
16 },
17 {
18     "_id": {
19         "$oid": "6314f6b8eceb3a0220524963"
20     },
21     "lastName": "Anstruther",
22     "firstName": "Jimmy",
23     "age": 21
24 },
25 {
26     "_id": {
27         "$oid": "6314f6b8eceb3a0220524964"
28     },
29     "lastName": "Stevens",
30     "firstName": "Amelia",
31     "age": 28
32 },
33 {
34     "_id": {
35         "$oid": "6314f6b8eceb3a0220524965"
36     },
37     "lastName": "Hopkins",
38     "firstName": "Fred",
```

```
39   "age": 35,
40   "favoriteColor": "Green"
41 },
42 {
  Edit Document
43   "_id": {
44     "$oid": "6315008fce345ecf70dcdced"
45   },
46   "lastName": "Walker",
47   "firstName": "Heather",
48   "middleName": "Lee",
49   "age": 37,
50   "favoriteColor": "blue"
51 },
52 {
  Edit Document
53   "_id": {
54     "$oid": "631501490427d645d2423a6d"
55   },
56   "lastName": "Walker",
57   "firstName": "Cinder",
58   "middleName": "SweetBoi",
59   "age": 6,
60   "breed": "Standard Poodle",
61   "favoriteFood": "lamb"
62 },
63 {
  Edit Document
64   "_id": {
65     "$oid": "631501955796be56bd72a531"
66   },
67   "lastName": "Walker",
68   "firstName": "Jarrod",
69   "middleName": "Thomas",
70   "age": 39,
71   "favoriteFood": "burgers"
72 }
73 ]
```

Part 2

2.1 Create a new collection

- This new collection should be named cars.

```

23 db.createCollection("appusers");
24
25 db.createCollection("cars");

```

2.2 Insert 5 cars into this collection

- Include the following fields: `make`, `model`, `color`, `year`, `fourDoor`, and `fourWheelDrive`.
- The last two fields, `fourDoor` and `fourWheelDrive`, should be of type `Boolean` (i.e. `true` or `false`).

```
db.cars.insertMany([
  {make: "Toyota", model: "Corolla", color: "white", year: 2003,
    fourDoor: true, fourWheelDrive: false },
  {make: "Scion", model: "Xb", color: "black", year: 2014, fourDoor:
    true, fourWheelDrive: false },
  {make: "Mitsubishi", model: "Delica", color: "blue grey", year: 1993,
    fourDoor: true, fourWheelDrive: true },
  {make: "Subaru", model: "Impreza", color: "white", year: 2001,
    fourDoor: true, fourWheelDrive: true },
  {make: "Dodge", model: "Magnum", color: "red", year: 2007, fourDoor:
    true, fourWheelDrive: false }])
```

```

23 // Get Started
24 // MongoDB Playground
25 // db.createCollection("appusers");
26 // db.createCollection("appusers");
27 db.createCollection("cars");
28
29 db.cars.insertMany([
30   {make: "Toyota", model: "Corolla", color: "white", year: 2003, fourDoor: true, fourWheelDrive: false},
31   {make: "Scion", model: "xB", color: "black", year: 2014, fourDoor: true, fourWheelDrive: false},
32   {make: "Mitsubishi", model: "Delica", color: "blue grey", year: 1993, fourDoor: true, fourWheelDrive: true},
33   {make: "Subaru", model: "Impreza", color: "white", year: 2001, fourDoor: true, fourWheelDrive: true},
34   {make: "Dodge", model: "Magnum", color: "red", year: 2007, fourDoor: true, fourWheelDrive: false}]
35 )
36
37
38
39

```

2.3 Run a find() query

- Once that is done, run a `find()` query to see your newly created documents.

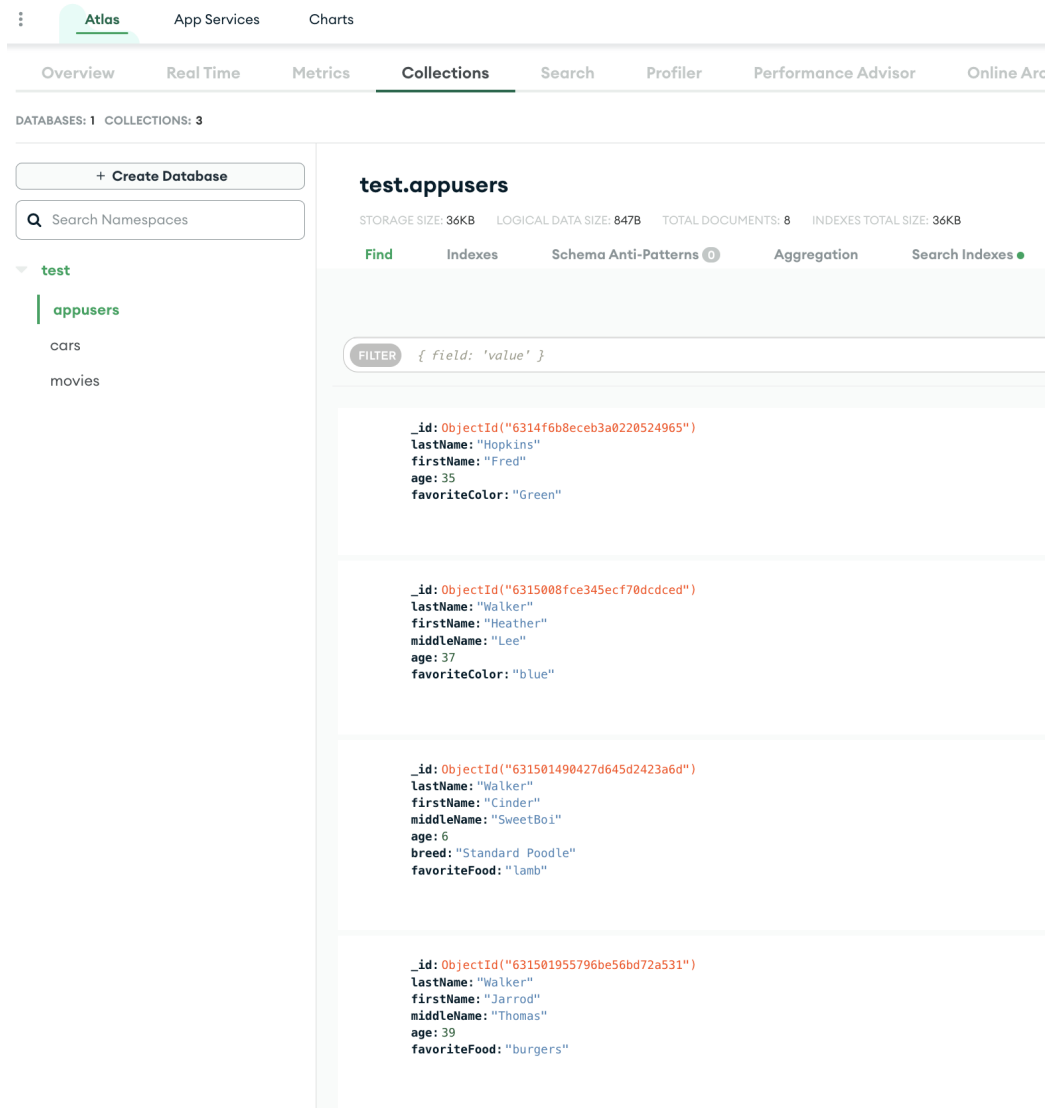
db.cars.find({})

```
Get Started // MongoDB Playground - db.createCollection("appusers"); - Playground Result X
23
24 db.createCollection("cars");
25
26 db.cars.insertMany([
27   {make: "Toyota", model: "Corolla", color: "white", year: 2003, fourDoor: true, fourWheelDrive: false },
28   {make: "Scion", model: "XP", color: "black", year: 2014, fourDoor: true, fourWheelDrive: false },
29   {make: "Mitsubishi", model: "Delica", color: "blue grey", year: 1993, fourDoor: true, fourWheelDrive: true },
30   {make: "Subaru", model: "Impreza", color: "white", year: 2001, fourDoor: true, fourWheelDrive: true },
31   {make: "Dodge", model: "Magnum", color: "red", year: 2007, fourDoor: true, fourWheelDrive: false }])
32
33
34
35
36
37
38
39 db.cars.find({})
40
```

```
1
2 {
3   "_id": "633526f67c0e3a7071c043b",
4   "make": "Toyota",
5   "model": "Corolla",
6   "color": "white",
7   "year": 2003,
8   "fourDoor": true,
9   "fourWheelDrive": false
10 },
11 {
12   "_id": "633526f67c0e3a7071c043c",
13   "make": "Scion",
14   "model": "xp",
15   "color": "black",
16   "year": 2014,
17   "fourDoor": true,
18   "fourWheelDrive": false
19 },
20 {
21   "_id": "633526f67c0e3a7071c043d",
22   "make": "Mitsubishi",
23   "model": "Delica",
24   "color": "blue grey",
25   "year": 1993,
26   "fourDoor": true,
27   "fourWheelDrive": true
28 },
29 {
30   "_id": "633526f67c0e3a7071c043e",
31   "make": "Subaru",
32   "model": "Impreza",
33   "color": "white",
34   "year": 2001,
35   "fourDoor": true,
36   "fourWheelDrive": true
37 },
38 {
39   "_id": "633526f67c0e3a7071c043f",
40   "make": "Dodge",
41   "model": "Magnum",
42   "color": "red",
43   "year": 2007,
44   "fourDoor": true,
45   "fourWheelDrive": false
46 }
47
48
```

2.4 View your collection and documents in MongoDB Atlas

- Lastly, return to Atlas and view your collection and documents there.



⋮

Atlas

App Services

Charts

Overview

Real Time

Metrics

Collections

Search

Profiler

Performance Advisor

Online Arc

DATABASES: 1COLLECTIONS: 3

+ Create Database

Q

Search Namespaces

test

appusers

cars

movies

test.cars

STORAGE SIZE: 20KBLOGICAL DATA SIZE: 562BTOTAL DOCUMENTS: 5INDEXES TOTAL SIZE: 20KB

Find

Indexes

Schema Anti-Patterns 0

Aggregation

Search Indexes ●

FILTER

{ field: 'value' }

QUERY RESULTS: 1-5 OF 5

>

_id: ObjectId("631526fd7c60e3a7671cd43b")

make: "Toyota"

model: "Corolla"

color: "white"

year: 2003

fourDoor: true

fourWheelDrive: false

_id: ObjectId("631526fd7c60e3a7671cd43c")

make: "Scion"

model: "Xb"

color: "black"

year: 2014

fourDoor: true

fourWheelDrive: false

_id: ObjectId("631526fd7c60e3a7671cd43d")

make: "Mitsubishi"

model: "Delica"

color: "blue grey"

year: 1993

fourDoor: true

fourWheelDrive: true

_id: ObjectId("631526fd7c60e3a7671cd43e")

make: "Subaru"

model: "Impreza"