

# DS108-05-05 - NoSQL - Lesson 5 Hands-On

## Final Project Information

Welcome to the last project for the NoSQL course! Great job making it this far! This hands-on will be different from the hands-on projects you have previously seen in a couple of different ways. You will be putting together the numerous topics you have learned into one large project. It is designed to mimic real problems which you may face in your career, so it may be a challenge for you and will also take several hours.

Take this project step-by-step and be aware that the project description below is written to be a bit less specific than previous Hands-Ons. The hands-on is supposed to challenge you to do some problem solving to figure out how to accomplish a task. You can always review past lessons or use a Google search if needed. Good luck!

[!Help] Additional Help! Before beginning this hands-on, you may want to watch this [recorded live workshop](#), "[Winnie the Pooh and Databases Too](#)," that goes over a similar example.

[!Caution] Caution! Do not submit your project until you have completed all requirements! You will not be able to resubmit.

[!info] To Submit Be sure to zip and submit your [Lesson5handson](#) 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.

---

## Requirements

For this hands on, you will be working through several real-life scenarios within new collections. This Hands-On is structured into *two* parts, and each part will 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 [Lesson5handson](#) . This way, you will be able to submit your answers to each part all at once.

---

# Part 1

You have just been hired at a startup company. They currently only have ten employees, but they need to be included in the database. So far, they have only been tracked within an excel sheet. Your boss would like you to create a new collection in Atlas named `employees` . Take a look at the following data and the notes listed below before inserting any data:

Name	Birthday	Address	City	State	Position Name	Remote	Full Time
Alison Davidson	04/05/75	874 W. Oak Place	Gary	Indiana	Customer Support	Yes	Yes
Henry Chapelton	09/29/80	9324 E. Vista Way	Tempe	Arizona	Customer Support	No	Yes
Alex Miller	11/22/83	244 Price Road	Mesa	Arizona	Customer Support	No	No
Carly Nielson	08/04/87	678 W. Westward Road	Phoenix	Arizona	Office Manager	No	Yes
Tom Talbot	12/30/89	12 Oakland Way	Chandler	Arizona	Inventory Manager	No	Yes
Mary Crawley	07/06/80	1010 Granite Way	Charlotte	North Carolina	Human Resources	Yes	Yes
Daisy Baxter	09/09/87	990 E. 84th St.	Tempe	Arizona	CEO	No	Yes
William Coyle	10/11/91	944 W. 16th St.	Phoenix	Arizona	Intern	No	No
Edith Bates	07/28/90	7 E. 20th Pl.	Chandler	Arizona	Customer Support	No	Yes
Gwen Harding	10/11/86	234 W. 48th. St.	Phoenix	Arizona	Office Assistant	No	Yes

## Notes:

- The `Birthday` field should have a data type of Date.
- The `Position Name` , `Remote` , and `Full Time` fields should be within an embedded document called `position` .
  - `position.PositionName`
  - `position.Remote`
  - `position.FullTime`
- `Remote` and `FullTime` fields should have boolean values.

## 1.1 Create `employees` database

```
db.employees.insertMany([
  {
    name: "Alison Davidson",
    birthday: new Date( "04/05/1975" ),
    address: "874 W. Oak Place",
    city: "Gary",
    state: "Indiana",
    position: { positionName: "Customer Support", remote: true, fullTime: true }
  },
  {
    name: "Henry Chapelton",
    birthday: new Date( "09/29/1980" ),
    address: "9324 E. Vista Way",
    city: "Tempe",
    state: "Arizona",
    position: { positionName: "Customer Support", remote: false, fullTime: true }
  },
  {
    name: "Alex Miller",
    birthday: new Date( "11/22/1983" ),
    address: "244 Price Road",
    city: "Mesa",
    state: "Arizona",
    position: { positionName: "Customer Support", remote: false, fullTime: false }
  },
  {
    name: "Carly Nielson",
    birthday: new Date( "08/04/1987" ),
    address: "678 W. Westward Road",
    city: "Phoenix",
    state: "Arizona",
    position: { positionName: "Office Manager", remote: false, fullTime: true }
  },
  {
    name: "Tom Talbot",
    birthday: new Date( "12/30/1989" ),
    address: "12 Oakland Way",
    city: "Chandler",
    state: "Arizona",
    position: { positionName: "Inventory Manager", remote: false, fullTime: true }
  },
  {
    name: "Mary Crawley",
    birthday: new Date( "07/06/1980" ),
    address: "1010 Granite Way",
    city: "Charlotte",
    state: "North Carolina",
    position: { positionName: "Human Resources", remote: true, fullTime: true }
  },
  {
    name: "Daisy Baxter",
    birthday: new Date( "09/09/1987" ),
    address: "990 E. 84th St.",
    city: "Tempe",
    state: "Arizona",
    position: { positionName: "CEO", remote: false, fullTime: true }
  },
  {
    name: "William Coyle",
    birthday: new Date( "10/11/1991" ),
    address: "944 W. 16th St.",
    city: "Phoenix",
    state: "Arizona",
    position: { positionName: "Intern", remote: false, fullTime: false }
  },
  {
    name: "Edith Bates",
    birthday: new Date( "07/28/1990" ),
    address: "7 E. 20th Pl.",
```

```

city: "Chandler",
state: "Arizona",
position: { positionName: "Customer Support", remote: false, fullTime: true }
},
{
name: "Gwen Harding",
birthday: new Date( "10/11/1986" ),
address: "234 W. 48th. St.",
city: "Phoenix",
state: "Arizona",
position: { positionName: "Office Assistant", remote: false, fullTime: true }
}
});

```

Get Started

Heather Walker - Lesson5HandsOn - Part 1 Code.mongoddb U

Student-Resources > DS108-Databases > Heather Walker - DS108-05-05 - NoSQL - Lesson 5 Hands-On > Heather Walker - Lesson

Currently connected to cluster0.6hu1ng.mongodb.net. Click here to change connection.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

db.employees.insertMany([

{

name: "Alison Davidson",

birthday: new Date( "04/05/1975" ),

address: "874 W. Oak Place",

city: "Gary",

state: "Indiana",

position: { positionName: "Customer Support", remote: true, fullTime: true }

},

{

name: "Henry Chapelton",

birthday: new Date( "09/29/1988" ),

address: "9324 E. Vista Way",

city: "Tempe",

state: "Arizona",

position: { positionName: "Customer Support", remote: false, fullTime: true }

},

{

name: "Alex Miller",

birthday: new Date( "11/22/1983" ),

address: "244 Price Road",

city: "Mesa",

state: "Arizona",

position: { positionName: "Customer Support", remote: false, fullTime: false }

},

{

name: "Carly Nielson",

birthday: new Date( "08/04/1987" ),

address: "678 W. Westward Road",

city: "Phoenix",

state: "Arizona",

position: { positionName: "Office Manager", remote: false, fullTime: true }

},

{

name: "Tom Talbot",

birthday: new Date( "12/30/1989" ),

address: "12 Oakland Way",

city: "Chandler",

state: "Arizona",

position: { positionName: "Inventory Manager", remote: false, fullTime: true }

},

{

name: "Mary Crawley",

birthday: new Date( "07/06/1988" ),

address: "1010 Granite Way",

city: "Charlotte",

state: "North Carolina",

position: { positionName: "Human Resources", remote: true, fullTime: true }

},

{

name: "Daisy Baxter",

birthday: new Date( "09/09/1987" ),

address: "990 E. 84th St.",

city: "Tempe",

state: "Arizona",

position: { positionName: "CEO", remote: false, fullTime: true }

},

{

name: "William Coyle",

birthday: new Date( "10/11/1991" ),

address: "944 W. 16th St.",

city: "Phoenix",

state: "Arizona",

position: { positionName: "Intern", remote: false, fullTime: false }

},

{

name: "Edith Bates",

birthday: new Date( "07/28/1990" ),

address: "7 E. 20th Pl.",

city: "Chandler",

state: "Arizona",

position: { positionName: "Customer Support", remote: false, fullTime: true }

},

{

name: "Gwen Harding",

birthday: new Date( "10/11/1986" ),

address: "234 W. 48th. St.",

city: "Phoenix",

state: "Arizona",

position: { positionName: "Office Assistant", remote: false, fullTime: true }

},

]);

Playground Result

Include Import Statements | Include Driver Syntax

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

{

"acknowledged": true,

"insertedIds": {

"gw": {

"\$oid": "631d2ac6c73a17c4f1967d8c"

},

"1": {

"\$oid": "631d2ac6c73a17c4f1967d8d"

},

"2": {

"\$oid": "631d2ac6c73a17c4f1967d8e"

},

"3": {

"\$oid": "631d2ac6c73a17c4f1967d8f"

},

"4": {

"\$oid": "631d2ac6c73a17c4f1967d90"

},

"5": {

"\$oid": "631d2ac6c73a17c4f1967d91"

},

"6": {

"\$oid": "631d2ac6c73a17c4f1967d92"

},

"7": {

"\$oid": "631d2ac6c73a17c4f1967d93"

},

"8": {

"\$oid": "631d2ac6c73a17c4f1967d94"

},

"9": {

"\$oid": "631d2ac6c73a17c4f1967d95"

},

}

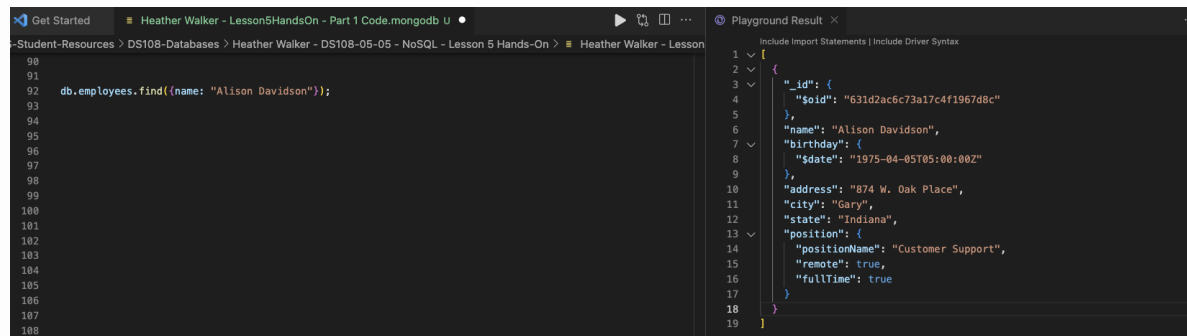
```
db.employees.find();
```

```
Get Started Heather Walker - Lesson5HandsOn - Part 1 Code.mongod... Playground Result X
Student-Resources > DS108-Databases > Heather Walker - DS108-05-05 - NoSQL - Lesson 5 Hands-On > Heather Walker - Lesson
Currently connected to cluster0.6hu1ngc.mongodb.net. Click here to change connection.
1 db.employees.insertMany([
2   {
3     name: "Alison Davidson",
4     birthday: new Date( "04/05/1975" ),
5     address: "874 W. Oak Place",
6     city: "Gary",
7     state: "Indiana",
8     position: { positionName: "Customer Support", remote: true, fullTime: true }
9   },
10  {
11    name: "Henry Chapelon",
12    birthday: new Date( "09/29/1980" ),
13    address: "9324 E. Vista Way",
14    city: "Tempe",
15    state: "Arizona",
16    position: { positionName: "Customer Support", remote: false, fullTime: true }
17  },
18  {
19    name: "Alex Miller",
20    birthday: new Date( "11/22/1983" ),
21    address: "244 Price Road",
22    city: "Mesa",
23    state: "Arizona",
24    position: { positionName: "Customer Support", remote: false, fullTime: false }
25  },
26  {
27    name: "Carly Nielson",
28    birthday: new Date( "08/04/1987" ),
29    address: "678 W. Westward Road",
30    city: "Phoenix",
31    state: "Arizona",
32    position: { positionName: "Office Manager", remote: false, fullTime: true }
33  },
34  {
35    name: "Tom Talbot",
36    birthday: new Date( "12/30/1989" ),
37    address: "112 Oakland Way",
38    city: "Chandler",
39    state: "Arizona",
40    position: { positionName: "Inventory Manager", remote: false, fullTime: true }
41  },
42  {
43    name: "Mary Crawley",
44    birthday: new Date( "07/06/1980" ),
45    address: "1010 Granite Way",
46    city: "Charlotte",
47    state: "North Carolina",
48    position: { positionName: "Human Resources", remote: true, fullTime: true }
49  },
50  {
51    name: "Daisy Baxter",
52    birthday: new Date( "09/09/1987" ),
53    address: "990 E. 84th St.",
54    city: "Tempe",
55    state: "Arizona",
56    position: { positionName: "CEO", remote: false, fullTime: true }
57  },
58  {
59    name: "William Coyle",
60    birthday: new Date( "10/11/1991" ),
61    address: "944 W. 16th St.",
62    city: "Phoenix",
63    state: "Arizona",
64    position: { positionName: "Intern", remote: false, fullTime: false }
65  },
66  {
67    name: "Edith Bates",
68    birthday: new Date( "07/28/1990" ),
69    address: "7 E. 20th Pl.",
70    city: "Chandler",
71    state: "Arizona",
72    position: { positionName: "Customer Support", remote: false, fullTime: true }
73  },
74  {
75    name: "Gwen Harding",
76    birthday: new Date( "10/11/1986" ),
77    address: "234 W. 48th. St.",
78    city: "Phoenix",
79    state: "Arizona",
80    position: { positionName: "Office Assistant", remote: false, fullTime: true }
81  },
82  ]);
83
84
85 db.employees.find();
86
87
88
89
90
1  {
2    "_id": {
3      "$oid": "631d2ac6c73a17c4f1967d8c"
4    },
5    "name": "Alison Davidson",
6    "birthday": {
7      "$date": "1975-04-05T05:00:00Z"
8    },
9    "address": "874 W. Oak Place",
10   "city": "Gary",
11   "state": "Indiana",
12   "position": {
13     "positionName": "Customer Support",
14     "remote": true,
15     "fullTime": true
16   }
17 },
18 {
19   "_id": {
20     "$oid": "631d2ac6c73a17c4f1967d8d"
21   },
22   "name": "Henry Chapelon",
23   "birthday": {
24     "$date": "1980-09-29T05:00:00Z"
25   },
26   "address": "9324 E. Vista Way",
27   "city": "Tempe",
28   "state": "Arizona",
29   "position": {
30     "positionName": "Customer Support",
31     "remote": false,
32     "fullTime": true
33   }
34 },
35 {
36   "_id": {
37     "$oid": "631d2ac6c73a17c4f1967d8e"
38   },
39   "name": "Alex Miller",
40   "birthday": {
41     "$date": "1983-11-22T06:00:00Z"
42   },
43   "address": "244 Price Road",
44   "city": "Mesa",
45   "state": "Arizona",
46   "position": {
47     "positionName": "Customer Support",
48     "remote": false,
49     "fullTime": false
50   }
51 },
52 {
53   "_id": {
54     "$oid": "631d2ac6c73a17c4f1967d8f"
55   },
56   "name": "Carly Nielson",
57   "birthday": {
58     "$date": "1987-08-04T05:00:00Z"
59   },
60   "address": "678 W. Westward Road",
61   "city": "Phoenix",
62   "state": "Arizona",
63   "position": {
64     "positionName": "Office Manager",
65     "remote": false,
66     "fullTime": true
67   }
68 },
69 {
70   "_id": {
71     "$oid": "631d2ac6c73a17c4f1967d90"
72   },
73   "name": "Tom Talbot",
74   "birthday": {
75     "$date": "1989-12-30T06:00:00Z"
76   },
77   "address": "112 Oakland Way",
78   "city": "Chandler",
79   "state": "Arizona",
80   "position": {
81     "positionName": "Inventory Manager",
82     "remote": false,
83     "fullTime": true
84   }
85 },
86 {
87   "_id": {
88     "$oid": "631d2ac6c73a17c4f1967d91"
89   },
90   "name": "Mary Crawley",
91   "birthday": {
92     "$date": "1980-07-06T05:00:00Z"
93   },
94   "address": "1010 Granite Way",
95   "city": "Charlotte",
96   "state": "North Carolina",
97   "position": {
98     "positionName": "Human Resources",
99     "remote": true,
100    "fullTime": true
101  }
102 },
103 {
104   "_id": {
105     "$oid": "631d2ac6c73a17c4f1967d92"
106   },
107   "name": "Daisy Baxter",
108   "birthday": {
109     "$date": "1987-09-09T05:00:00Z"
110   },
111   "address": "990 E. 84th St.",
112   "city": "Tempe",
113   "state": "Arizona",
114   "position": {
115     "positionName": "CEO",
116     "remote": false,
117     "fullTime": true
118   }
119 },
120 {
121   "_id": {
122     "$oid": "631d2ac6c73a17c4f1967d93"
123   },
124   "name": "William Coyle",
125   "birthday": {
126     "$date": "1991-10-11T05:00:00Z"
127   },
128   "address": "944 W. 16th St.",
129   "city": "Phoenix",
130   "state": "Arizona",
131   "position": {
132     "positionName": "Intern",
133     "remote": false,
134     "fullTime": false
135   }
136 },
137 {
138   "_id": {
139     "$oid": "631d2ac6c73a17c4f1967d94"
140   },
141   "name": "Edith Bates",
142   "birthday": {
143     "$date": "1990-07-28T05:00:00Z"
144   },
145   "address": "7 E. 20th Pl.",
146   "city": "Chandler",
147   "state": "Arizona",
148   "position": {
149     "positionName": "Customer Support",
150     "remote": false,
151     "fullTime": true
152   }
153 },
154 {
155   "_id": {
156     "$oid": "631d2ac6c73a17c4f1967d95"
157   },
158   "name": "Gwen Harding",
159   "birthday": {
160     "$date": "1986-10-11T05:00:00Z"
161   },
162   "address": "234 W. 48th. St.",
163   "city": "Phoenix",
164   "state": "Arizona",
165   "position": {
166     "positionName": "Office Assistant",
167     "remote": false,
168     "fullTime": true
169   }
170 }
```

## 1.2 Change document for Alison Davidson

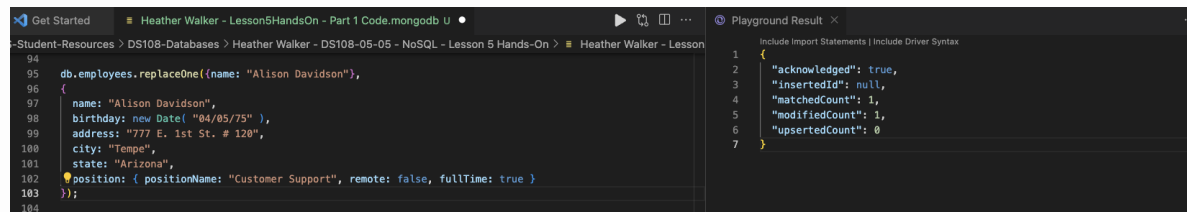
It's been about a month since you have inserted all employees into the database. There have been a couple of changes to the company. The CEO decided that he no longer wants remote employees, so they have transferred the remote employees and they are now living in Arizona. - Alison Davidson now lives at 777 E. 1st St. # 120 Tempe, AZ

```
db.employees.find({name: "Alison Davidson"});
```



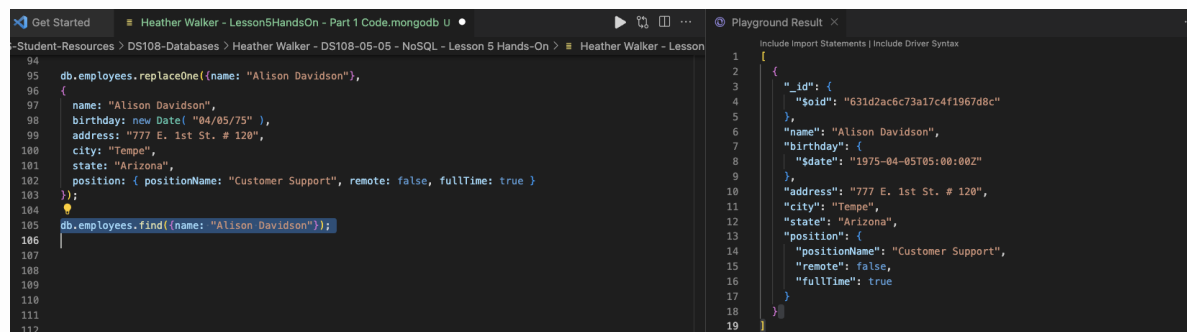
The screenshot shows a MongoDB Playground interface. On the left, the code editor contains the query `db.employees.find({name: "Alison Davidson"});`. On the right, the 'Playground Result' pane shows the JSON document for Alison Davidson, including fields for `_id`, `$oid`, `name`, `birthday`, `$date`, `address`, `city`, `state`, and `position` (with sub-fields `positionName`, `remote`, and `fullTime`).

```
db.employees.replaceOne({name: "Alison Davidson"},
{
  name: "Alison Davidson",
  birthday: new Date( "04/05/75" ),
  address: "777 E. 1st St. # 120",
  city: "Tempe",
  state: "Arizona",
  position: { positionName: "Customer Support", remote: false, fullTime: true }
});
```



The screenshot shows the MongoDB Playground after executing the `replaceOne` query. The left pane shows the query `db.employees.replaceOne({name: "Alison Davidson"}, { ... })`. The right pane shows the result, which is an object with counts: `"acknowledged": true`, `"insertedCount": null`, `"matchedCount": 1`, `"modifiedCount": 1`, and `"upsertedCount": 0`.

```
db.employees.find({name: "Alison Davidson"});
```

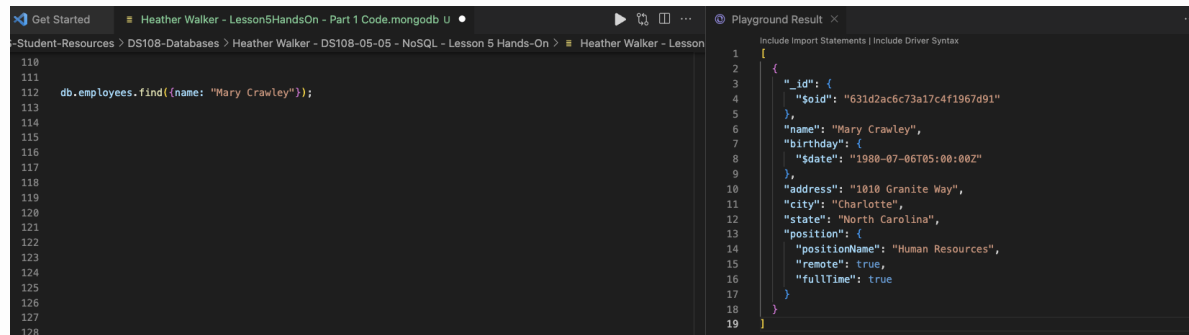


The screenshot shows the MongoDB Playground after executing the `find` query again. The left pane shows the query `db.employees.find({name: "Alison Davidson"});`. The right pane shows the updated JSON document for Alison Davidson, where the `remote` field is now `false` and the `state` is `"Arizona"`.

## 1.3 Change document for Mary Crawley

- Mary Crawley now lives at 8322 W. Vista Pl. Scottsdale, AZ

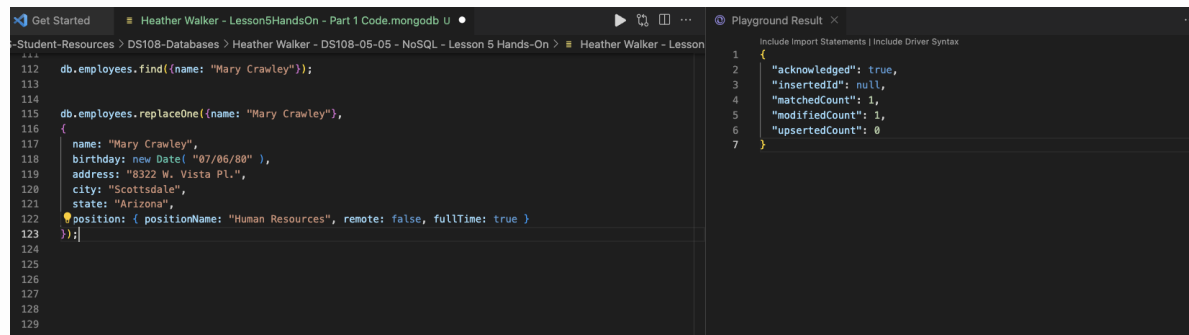
```
db.employees.find({name: "Mary Crawley"});
```



The screenshot shows the MongoDB Playground interface. The left pane contains the command `db.employees.find({name: "Mary Crawley"});`. The right pane, titled "Playground Result", shows the JSON document for Mary Crawley:

```
{
  "_id": {
    "$oid": "631d2ac6c73a17c4f1967d91"
  },
  "name": "Mary Crawley",
  "birthday": {
    "$date": "1980-07-06T05:00:00Z"
  },
  "address": "1810 Granite Way",
  "city": "Charlotte",
  "state": "North Carolina",
  "position": {
    "positionName": "Human Resources",
    "remote": true,
    "fullTime": true
  }
}
```

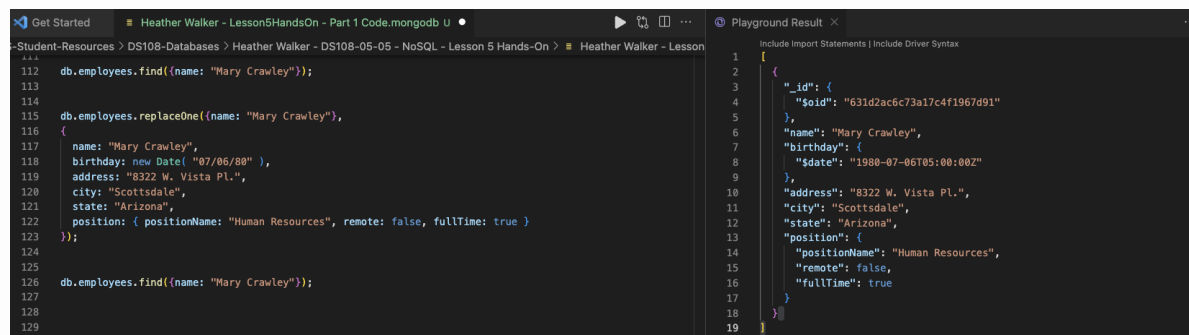
```
db.employees.replaceOne({name: "Mary Crawley"},
{
  name: "Mary Crawley",
  birthday: new Date( "07/06/80" ),
  address: "8322 W. Vista Pl.",
  city: "Scottsdale",
  state: "Arizona",
  position: { positionName: "Human Resources", remote: false, fullTime: true }
});
```



The screenshot shows the MongoDB Playground interface. The left pane contains the `replaceOne` command. The right pane shows the result of the update operation:

```
{
  "acknowledged": true,
  "insertedId": null,
  "matchedCount": 1,
  "modifiedCount": 1,
  "upsertedCount": 0
}
```

```
db.employees.find({name: "Mary Crawley"});
```



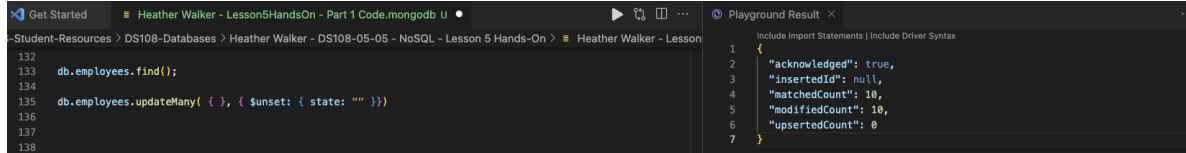
The screenshot shows the MongoDB Playground interface. The left pane contains the `find` command. The right pane shows the updated JSON document for Mary Crawley:

```
{
  "_id": {
    "$oid": "631d2ac6c73a17c4f1967d91"
  },
  "name": "Mary Crawley",
  "birthday": {
    "$date": "1980-07-06T05:00:00Z"
  },
  "address": "8322 W. Vista Pl.",
  "city": "Scottsdale",
  "state": "Arizona",
  "position": {
    "positionName": "Human Resources",
    "remote": false,
    "fullTime": true
  }
}
```

## 1.4 Remove **state** field

Since all employees now all live in Arizona, there is no need to have a field named “state” within this collection, so please remove it.

```
db.employees.updateMany( {}, { $unset: { state: "" } });
```



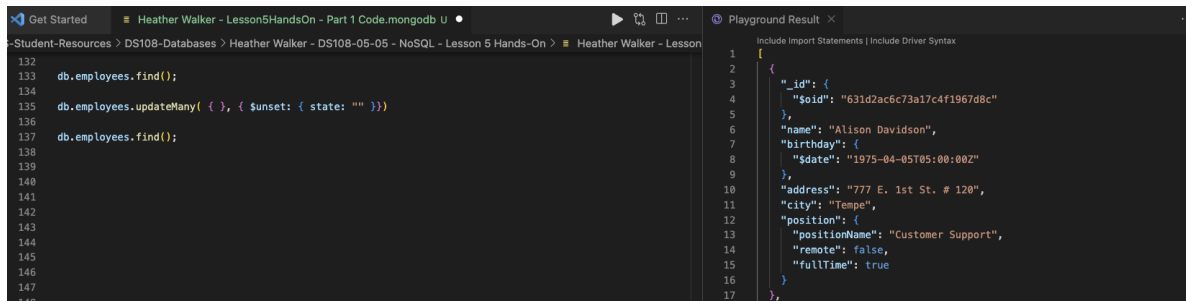
The screenshot shows the MongoDB Playground interface. On the left, the code editor contains the following commands:

```
132  
133 db.employees.find();  
134  
135 db.employees.updateMany( {}, { $unset: { state: "" } })  
136  
137  
138
```

On the right, the 'Playground Result' panel shows the output of the update operation:

```
1 Include Import Statements | Include Driver Syntax  
2 {  
3   "acknowledged": true,  
4   "insertedId": null,  
5   "matchedCount": 10,  
6   "modifiedCount": 10,  
7   "upsertedCount": 0  
}
```

```
db.employees.find();
```



The screenshot shows the MongoDB Playground interface. On the left, the code editor contains the following commands:

```
132  
133 db.employees.find();  
134  
135 db.employees.updateMany( {}, { $unset: { state: "" } })  
136  
137 db.employees.find();  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148
```

On the right, the 'Playground Result' panel shows the output of the find operation:

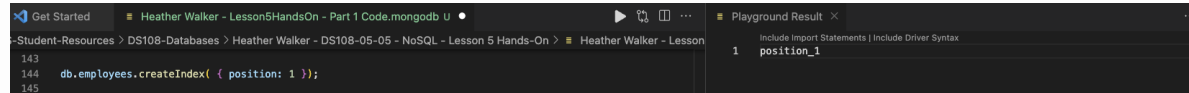
```
1 Include Import Statements | Include Driver Syntax  
2 {  
3   "_id": {  
4     "$oid": "631d2ac6c73a17c4f1967d8c"  
5   },  
6   "name": "Alison Davidson",  
7   "birthday": {  
8     "$date": "1975-04-05T05:00:00Z"  
9   },  
10  "address": "777 E. 1st St. # 120",  
11  "city": "Tempe",  
12  "position": {  
13    "positionName": "Customer Support",  
14    "remote": false,  
15    "fullTime": true  
16  },  
17 }
```



## 1.5 Create index on **position** field / document

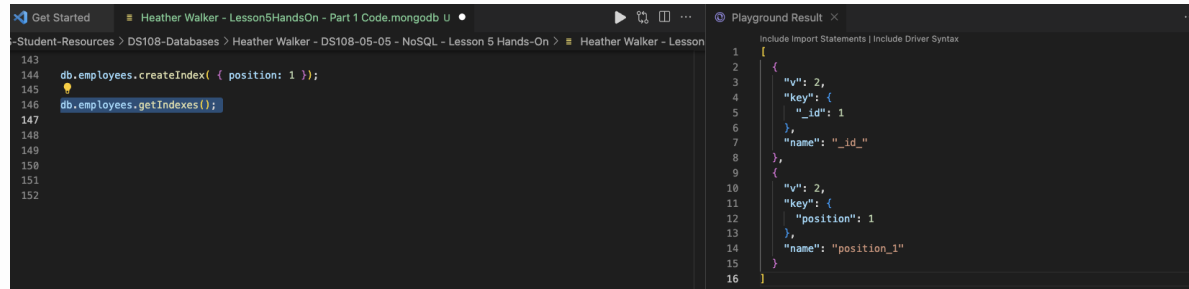
Lastly, they would like very efficient searching using the “position” field (remember that field includes a document with three other fields).

```
db.employees.createIndex( { position: 1 } );
```



The screenshot shows the MongoDB Playground interface. The left pane contains the code `db.employees.createIndex( { position: 1 } );` at line 144. The right pane, titled "Playground Result", shows the output: `position_1`.

```
db.employees.getIndexes();
```



The screenshot shows the MongoDB Playground interface. The left pane contains the code `db.employees.createIndex( { position: 1 } );` at line 144 and `db.employees.getIndexes();` at line 146. The right pane, titled "Playground Result", shows the output of the `getIndexes` command as a JSON array:

```
[{"v": 2, "key": {"_id": 1}, "name": "_id_"}, {"v": 2, "key": {"position": 1}, "name": "position_1"}]
```

## Part 2

You are currently working for a company who wants to build an app similar to Spotify. Below is a list of data for different songs. Please insert this data into a new collection named `songs` .

SongId	Title	Artist	Album	ReleaseYear
1	Girls Just Want To Have Fun	Cyndi Lauper	She's So Unusual	1983
2	Hips Don't Lie	Shakira feat. Wyclef Jean	Oral Fixation Vol. 2	2006
3	Poker Face	Lady Gaga	The Fame	2008
4	Wannabe	Spice Girls	Spice	1996
5	California Gurls	Katy Perry feat. Snoop Dogg	Teenage Dream	2010
6	Bye, Bye, Bye	NSYNC	No Strings Attached	2000
7	I Will Always Love You	Whitney Houston	I Will Always Love You: The Best of Whitney Houston	2012
8	Baby One More Time	Britney Spears	Baby One More Time	1999
9	Vogue	Madonna	I'm Breathless	1990
10	Rolling in the Deep	Adele	21	2011
11	1234	Feist	The Reminder	2007
12	Elastic Heart	Sia	The Hunger Games: Catching Fire Soundtrack	2015
13	Oops! I Did It Again	Britney Spears	Oops! I Did It Again	2000
14	Bad Romance	Lady Gaga	The Fame Monster	2009
15	Lose Control	Missy Elliot	The Cookbook	2005
16	U Can't Touch This	MC Hammer	Please Hammer, Don't Hurt 'Em	1990
17	Thriller	Michael Jackson	Thriller	1982
18	Single Ladies	Beyonce	I am... Sasha Fierce	2008
19	Rhythm Nation	Janet Jackson	Janet Jackson's Rhythm Nation 1814	1989

### Notes:

- The `artist` , `album` , and `releaseYear` fields should be an embedded document named `details` .
- Be sure that the `songId` and `releaseYear` fields have a type of number.

## 2.1 Create **songs** database

```
db.songs.insertMany([
  {
    "songId": 1,
    "title": "Girls Just Want To Have Fun",
    "details": { "artist" : "Cyndi Lauper",
    "album": "She's So Unusual",
    "releaseYear": 1983
    }
  },
  {
    "songId": 2,
    "title": "Hips Don't Lie",
    "details": { "artist" : "Shakira feat. Wyclef Jean",
    "album": "Oral Fixation Vol. 2",
    "releaseYear": 2006
    }
  },
  {
    "songId": 3,
    "title": "Poker Face",
    "details": { "artist" : "Lady Gaga",
    "album": "The Fame",
    "releaseYear": 2008
    }
  },
  {
    "songId": 4,
    "title": "Wannabe",
    "details": { "artist" : "Spice Girls",
    "album": "Spice",
    "releaseYear": 1996
    }
  },
  {
    "songId": 5,
    "title": "California Gurls",
    "details": { "artist" : "Katy Perry feat. Snoop Dogg",
    "album": "Teenage Dream",
    "releaseYear": 2010
    }
  },
  {
    "songId": 6,
    "title": "Bye, Bye, Bye",
    "details": { "artist" : "NSYNC",
    "album": "No Strings Attached",
    "releaseYear": 2000
    }
  },
  {
    "songId": 7,
    "title": "I Will Always Love You",
    "details": { "artist" : "Whitney Houston",
    "album": "I Will Always Love You: The Best of Whitney Houston",
    "releaseYear": 2012
    }
  },
  {
    "songId": 8,
    "title": "Baby One More Time",
    "details": { "artist" : "Britney Spears",
    "album": "Baby One More Time",
    "releaseYear": 1999
    }
  },
  {
    "songId": 9,
    "title": "Vogue",
    "details": { "artist" : "Madonna",
    "album": "I'm Breathless",
    "releaseYear": 1990
    }
  },
  {
    "songId": 10,
    "title": "Rolling in the Deep",
    "details": { "artist" : "Adele",
```

```

        "album": "21",
        "releaseYear": 2011
    }}

    {
        "songId": 11,
        "title": "1234",
        "details": { "artist": "Feist",
        "album": "The Reminder",
        "releaseYear": 2007
        }}

    {
        "songId": 12,
        "title": "Elastic Heart",
        "details": { "artist": "Sia",
        "album": "The Hunger Games: Catching Fire Soundtrack",
        "releaseYear": 2015
        }}

    {
        "songId": 13,
        "title": "Oops! I Did It Again",
        "details": { "artist": "Britney Spears",
        "album": "Oops! I Did It Again",
        "releaseYear": 2000
        }}

    {
        "songId": 14,
        "title": "Bad Romance",
        "details": { "artist": "Lady Gaga",
        "album": "The Fame Monster",
        "releaseYear": 2009
        }}

    {
        "songId": 15,
        "title": "Lose Control",
        "details": { "artist": "Missy Elliot",
        "album": "The Cookbook",
        "releaseYear": 2005
        }}

    {
        "songId": 16,
        "title": "U Can't Touch This",
        "details": { "artist": "MC Hammer",
        "album": "Please Hammer, Don't Hurt 'Em",
        "releaseYear": 1990
        }}

    {
        "songId": 17,
        "title": "Thriller",
        "details": { "artist": "Michael Jackson",
        "album": "Thriller",
        "releaseYear": 1982
        }}

    {
        "songId": 18,
        "title": "Single Ladies",
        "details": { "artist": "Beyonce",
        "album": "I am... Sasha Fierce",
        "releaseYear": 2008
        }}

    {
        "songId": 19,
        "title": "Rhythm Nation",
        "details": { "artist": "Janet Jackson",
        "album": "Janet Jackson's Rhythm Nation 1814",
        "releaseYear": 1989
        }}
    );

```

```
Heather Walker - Lesson5HandsOn - Part 2 Code.mongod... X
bases > Heather Walker - DS108-05-05 - NoSQL - Lesson 5 Hands-On > Heather Walker - Lesson5HandsOn - Part 2 Code.mongod...
Currently connected to cluster0.dhu1ngc.mongod...net. Click here to change connection.
1 db.songs.insertMany([
2 {
3   "songId": 1,
4   "title": "Girls Just Want To Have Fun",
5   "details": { "artist": "Cyndi Lauper",
6   "album": "She's So Unusual",
7   "releaseYear": 1983
8 },
9 {
10  "songId": 2,
11  "title": "Hips Don't Lie",
12  "details": { "artist": "Shakira feat. Wyclef Jean",
13  "album": "Oral Fixation Vol. 2",
14  "releaseYear": 2006
15 },
16 {
17  "songId": 3,
18  "title": "Poker Face",
19  "details": { "artist": "Lady Gaga",
20  "album": "The Fame",
21  "releaseYear": 2008
22 },
23 {
24  "songId": 4,
25  "title": "Wannabe",
26  "details": { "artist": "Spice Girls",
27  "album": "Spice",
28  "releaseYear": 1996
29 },
30 {
31  "songId": 5,
32  "title": "California Gurls",
33  "details": { "artist": "Katy Perry feat. Snoop Dogg",
34  "album": "Teenage Dream",
35  "releaseYear": 2010
36 },
37 {
38  "songId": 6,
39  "title": "Bye, Bye, Bye",
40  "details": { "artist": "NSYNC",
41  "album": "No Strings Attached",
42  "releaseYear": 2000
43 },
44 {
45  "songId": 7,
46  "title": "I Will Always Love You",
47  "details": { "artist": "Whitney Houston",
48  "album": "I Will Always Love You: The Best of Whitney Houston",
49  "releaseYear": 2012
50 },
51 {
52  "songId": 8,
53  "title": "Baby One More Time",
54  "details": { "artist": "Britney Spears",
55  "album": "Baby One More Time",
56  "releaseYear": 1999
57 },
58 {
59  "songId": 9,
60  "title": "Vogue",
61  "details": { "artist": "Madonna",
62  "album": "I'm Breathless",
63  "releaseYear": 1990
64 },
65 {
66  "songId": 10,
67  "title": "Rolling in the Deep",
68  "details": { "artist": "Adele",
69  "album": "21",
70  "releaseYear": 2011
71 },
72 {
73  "songId": 11,
74  "title": "1234",
75  "details": { "artist": "Feist",
76  "album": "The Reminder",
77  "releaseYear": 2007
78 },
79 {
80  "songId": 12,
81  "title": "Elastic Heart",
82  "details": { "artist": "Sia",
83  "album": "The Hunger Games: Catching Fire Soundtrack",
84  "releaseYear": 2015
85 },
86 {
87  "songId": 13,
88  "title": "Oops! I Did It Again",
89  "details": { "artist": "Britney Spears",
90  "album": "Oops! I Did It Again",
91  "releaseYear": 2000
92 }
93 ]
94 )
95 }
96 }
97 }
98 }
99 }
100 }
```

```
db.songs.find();
```

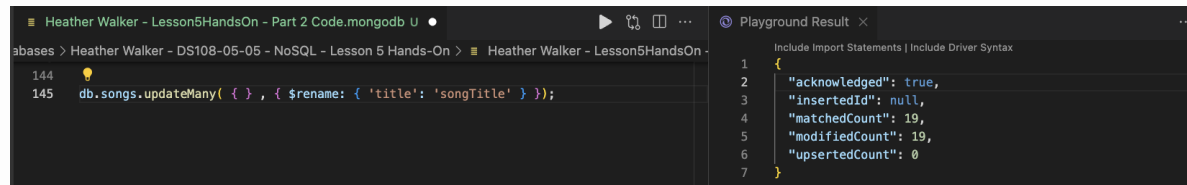
```
Heather Walker - Lesson5HandsOn - Part 2 Code.mongod...
base > Heather Walker - DS108-05-05 - NoSQL - Lesson 5 Hands-On > Heather Walker - Lesson5HandsOn - Part 2 Code.mongod...
51 {
52   "songId": 8,
53   "title": "Baby One More Time",
54   "details": { "artist": "Britney Spears",
55   "album": "Baby One More Time",
56   "releaseYear": 1999
57 },
58 {
59   "songId": 9,
60   "title": "Vogue",
61   "details": { "artist": "Madonna",
62   "album": "I'm Breathless",
63   "releaseYear": 1990
64 },
65 {
66   "songId": 10,
67   "title": "Rolling in the Deep",
68   "details": { "artist": "Adele",
69   "album": "21",
70   "releaseYear": 2011
71 },
72 {
73   "songId": 11,
74   "title": "1234",
75   "details": { "artist": "Feist",
76   "album": "The Reminder",
77   "releaseYear": 2007
78 },
79 {
80   "songId": 12,
81   "title": "Elastic Heart",
82   "details": { "artist": "Sia",
83   "album": "The Hunger Games: Catching Fire Soundtrack",
84   "releaseYear": 2015
85 },
86 {
87   "songId": 13,
88   "title": "Oops! I Did It Again",
89   "details": { "artist": "Britney Spears",
90   "album": "Oops! I Did It Again",
91   "releaseYear": 2000
92 },
93 {
94   "songId": 14,
95   "title": "Bad Romance",
96   "details": { "artist": "Lady Gaga",
97   "album": "The Fame Monster",
98   "releaseYear": 2009
99 },
100 {
101   "songId": 15,
102   "title": "Lose Control",
103   "details": { "artist": "Missy Elliott",
104   "album": "The Cookbook",
105   "releaseYear": 2005
106 },
107 {
108   "songId": 16,
109   "title": "U Can't Touch This",
110   "details": { "artist": "MC Hammer",
111   "album": "Please Hammer, Don't Hurt 'Em",
112   "releaseYear": 1990
113 },
114 {
115   "songId": 17,
116   "title": "Thriller",
117   "details": { "artist": "Michael Jackson",
118   "album": "Thriller",
119   "releaseYear": 1982
120 },
121 {
122   "songId": 18,
123   "title": "Single Ladies",
124   "details": { "artist": "Beyonce",
125   "album": "I am... Sasha Fierce",
126   "releaseYear": 2008
127 },
128 {
129   "songId": 19,
130   "title": "Rhythm Nation",
131   "details": { "artist": "Janet Jackson",
132   "album": "Janet Jackson's Rhythm Nation 1814",
133   "releaseYear": 1989
134 }
135 }):
136
137
138 db.songs.find();

1
2
3 {
4   "id": {
5     "$oid": "631d3554d5662f827732ca4f"
6   },
7   "songId": 1,
8   "title": "Girls Just Want To Have Fun",
9   "details": {
10     "artist": "Cyndi Lauper",
11     "album": "She's So Unusual",
12     "releaseYear": 1983
13   },
14 {
15   "id": {
16     "$oid": "631d3554d5662f827732ca58"
17   },
18   "songId": 2,
19   "title": "Hips Don't Lie",
20   "details": {
21     "artist": "Shakira feat. Wyclef Jean",
22     "album": "Oral Fixation Vol. 2",
23     "releaseYear": 2006
24   },
25 },
26 {
27   "id": {
28     "$oid": "631d3554d5662f827732ca51"
29   },
30   "songId": 3,
31   "title": "Poker Face",
32   "details": {
33     "artist": "Lady Gaga",
34     "album": "The Fame",
35     "releaseYear": 2008
36   },
37 },
38 {
39   "id": {
40     "$oid": "631d3554d5662f827732ca52"
41   },
42   "songId": 4,
43   "title": "Wannabe",
44   "details": {
45     "artist": "Spice Girls",
46     "album": "Spice",
47     "releaseYear": 1996
48   },
49 },
50 {
51   "id": {
52     "$oid": "631d3554d5662f827732ca53"
53   },
54   "songId": 5,
55   "title": "California Gurls",
56   "details": {
57     "artist": "Katy Perry feat. Snoop Dogg",
58     "album": "Teenage Dream",
59     "releaseYear": 2010
60   },
61 },
62 {
63   "id": {
64     "$oid": "631d3554d5662f827732ca54"
65   },
66   "songId": 6,
67   "title": "Bye, Bye, Bye",
68   "details": {
69     "artist": "NSYNC",
70     "album": "No Strings Attached",
71     "releaseYear": 2000
72   },
73 },
74 {
75   "id": {
76     "$oid": "631d3554d5662f827732ca55"
77   },
78   "songId": 7,
79   "title": "I Will Always Love You",
80   "details": {
81     "artist": "Whitney Houston",
82     "album": "I Will Always Love You: The Best of Whitney Houston",
83     "releaseYear": 2012
84   },
85 },
86 {
87   "id": {
88     "$oid": "631d3554d5662f827732ca56"
89   },
90   "songId": 8,
91   "title": "Baby One More Time",
92   "details": {
```

## 2.2 Rename `title` field

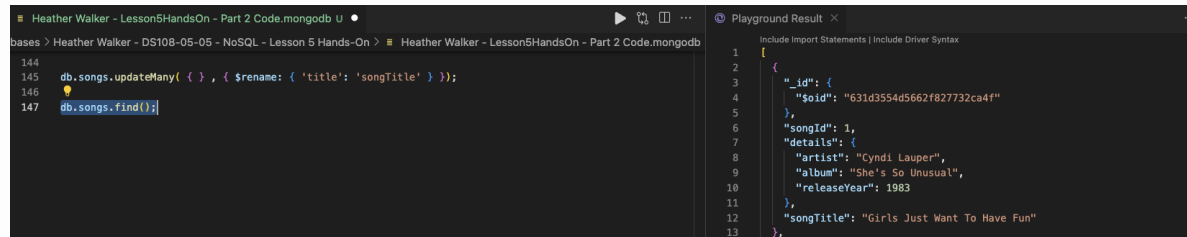
Next, your company has run into some things they would like to be changed within the database: - The `title` field needs to be renamed to `songTitle`, so it is clearer to the developers working with the data.

```
db.songs.updateMany( {}, { $rename: { 'title': 'songTitle' } } );
```



The screenshot shows the MongoDB Playground interface. On the left, the code editor contains the command `db.songs.updateMany( {}, { $rename: { 'title': 'songTitle' } } );` at line 145. On the right, the 'Playground Result' panel shows the result of the command: `{ "acknowledged": true, "insertedId": null, "matchedCount": 19, "modifiedCount": 19, "upsertedCount": 0 }`.

```
db.songs.find();
```

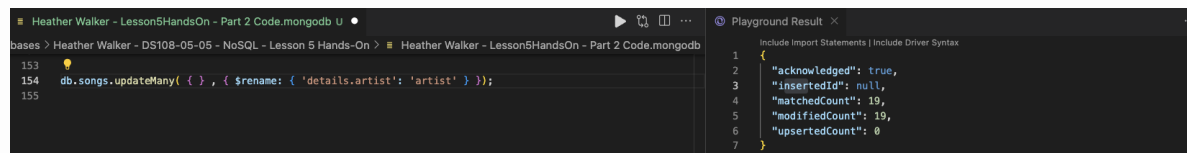


The screenshot shows the MongoDB Playground interface. On the left, the code editor contains the command `db.songs.find();` at line 147. On the right, the 'Playground Result' panel shows the result of the command: `{ "_id": { "$oid": "631d3554d5662f827732ca4f" }, "songId": 1, "details": { "artist": "Cyndi Lauper", "album": "She's So Unusual", "releaseYear": 1983 }, "songTitle": "Girls Just Want To Have Fun" }`.

## 2.3 Move `artist` field outside embedded `details` document

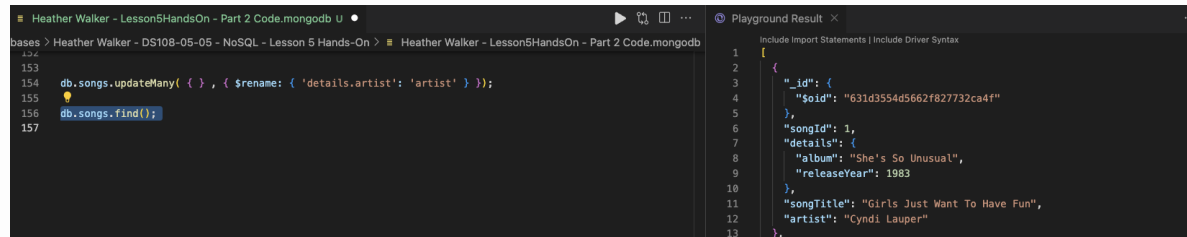
- They would like to have the `artist` field to be outside the `details` document but the `album` and `releaseYear` should stay within that document.

```
db.songs.updateMany( {}, { $rename: { 'details.artist': 'artist' } } );
```



The screenshot shows the MongoDB Playground interface. The left pane contains the command `db.songs.updateMany( {}, { $rename: { 'details.artist': 'artist' } } );` at line 154. The right pane, titled 'Playground Result', shows the result of the command: a document with `"acknowledged": true`, `"insertedId": null`, `"matchedCount": 19`, `"modifiedCount": 19`, and `"upsertedCount": 0`.

```
db.songs.find();
```



The screenshot shows the MongoDB Playground interface. The left pane contains the command `db.songs.find();` at line 156. The right pane, titled 'Playground Result', shows the result of the command: a document with `"_id": { "$oid": "631d3554d5662f827732ca4f" }`, `"songId": 1`, `"details": { "album": "She's So Unusual", "releaseYear": 1983 }`, `"songTitle": "Girls Just Want To Have Fun"`, and `"artist": "Cyndi Lauper"`.