

# Create a collection called restaurants

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure under 'Local', including 'MongoHW', 'admin', 'local', 'project\_db', and 'quiz\_app'. The main panel shows the 'Databases' tab with a table listing the databases and their details.

Database Name	Storage Size	Collections	Indexes
MongoHW	4.1KB	1	1
admin	0.0B	0	0
local	0.0B	7	0
project_db	143.4KB	6	6
quiz_app	12.3KB	3	3

At the bottom, a terminal window shows the command to create a collection:

```
>_MONGOSH
> db.createCollection("restaurants")
< { ok: 1 }
Atlas atlas-9ori8g-shard-0 [primary] MongoHW>
```

insert one resturant into the resturants collection

The screenshot shows the MongoDB Compass interface. On the left, the 'Local' sidebar lists databases and collections. The 'MongoHW' database is expanded, showing a 'restaurants' collection. The main panel displays the 'MongoHW.restaurants' collection with one document. The document details are as follows:

DOCUMENTS	TOTAL SIZE	AVG. SIZE
1	67B	67B

INDEXES	TOTAL SIZE	AVG. SIZE
1	4.1KB	4.1KB

The document content is:

```
{ "_id": ObjectId("619b4d0d093488a8bf290d23"), "name": "Albaik", "city": "Jeddah", "score": 10 }
```

Below the document list, the MongoDB shell output is visible:

```
>_MONGOSH
> db.restaurants.insertOne ( { "name": "Albaik", "city": "Jeddah", "score": 10 } )
< { acknowledged: true,
  insertedId: ObjectId("619b4d0d093488a8bf290d23") }
Atlas atlas-9ori8g-shard-0 [primary] MongoHW>
```

insert many restaurants into the restaurants collection,

The screenshot shows the MongoDB Compass web interface. On the left, the 'Local' sidebar lists the 'MongoHW' database and its 'restaurants' collection. The main panel shows the 'restaurants' collection with 1 document, a total size of 67B, and an average size of 67B. The document is displayed in JSON format:

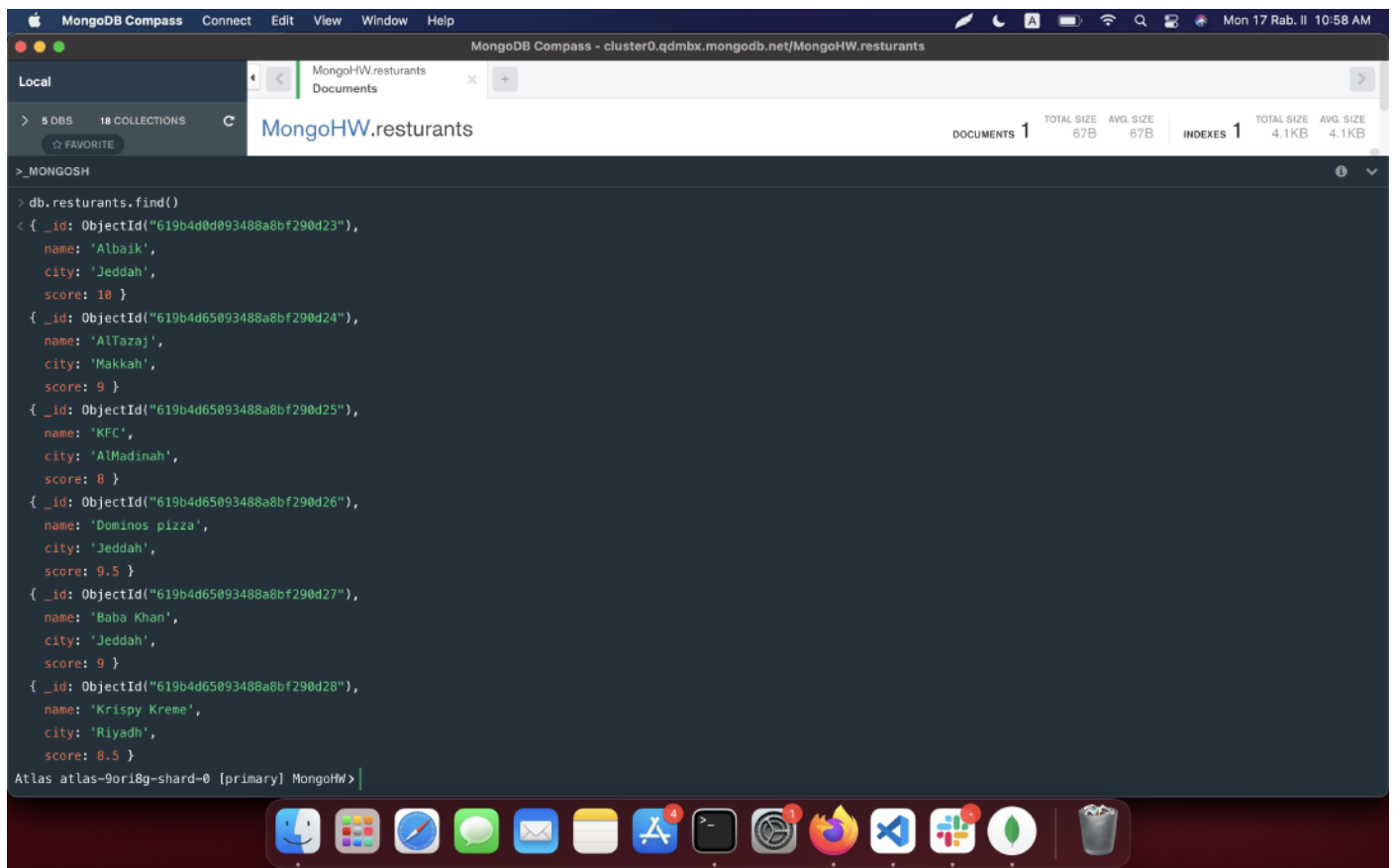
```
{
  "city": "Jeddah",
  "score": 10
}
```

Below the document, the MongoDB shell output is visible, showing the execution of the `insertMany` command and the resulting `insertedIds`:

```
> db.restaurants.insertMany([
  { "name": "AlTazaj", "city": "Makkah", "score": 9 },
  { "name": "KFC", "city": "AlMadinah", "score": 8 },
  { "name": "Dominos pizza", "city": "Jeddah", "score": 9.5 },
  { "name": "Baba Khan", "city": "Jeddah", "score": 9 },
  { "name": "Krispy Kreme", "city": "Riyadh", "score": 8.5 }
])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("619b4d65093488a8bf290d24"),
    '1': ObjectId("619b4d65093488a8bf290d25"),
    '2': ObjectId("619b4d65093488a8bf290d26"),
    '3': ObjectId("619b4d65093488a8bf290d27"),
    '4': ObjectId("619b4d65093488a8bf290d28")
  }
}
```

The terminal output also shows the shard information: `Atlas atlas-9or18g-shard-0 [primary] MongoHW>`

Display all the documents in the collection restaurants.



Display all the documents in the collection restaurants, but exclude the field `_id`.

The screenshot shows the MongoDB Compass interface. On the left, the 'Local' sidebar lists databases and collections. The main panel displays the 'MongoHW.restaurants' collection. A filter is applied: `{ field: 'value' }`. The document list shows three documents, with the first two visible. The command window at the bottom shows the following query and results:

```
> db.restaurants.find({}, {_id:0})
< { name: 'Albaik', city: 'Jeddah', score: 10 }
  { name: 'AlTazaj', city: 'Makkah', score: 9 }
  { name: 'KFC', city: 'AlMadinah', score: 8 }
  { name: 'Dominos pizza', city: 'Jeddah', score: 9.5 }
  { name: 'Baba Khan', city: 'Jeddah', score: 9 }
  { name: 'Krispy Kreme', city: 'Riyadh', score: 8.5 }
Atlas atlas-9or18g-shard-0 [primary] MongoHW>
```

Display all the restaurants which is in Jeddah city.

The screenshot shows the MongoDB Compass interface. On the left, the 'Local' sidebar lists the database 'MongoHW' and the collection 'restaurants'. The main panel displays the 'MongoHW.restaurants' collection with a filter bar containing the query `{ field: 'value' }`. Below the filter bar, a document is shown with the fields `city: "Jeddah"` and `score: 10`. The bottom panel shows the MongoDB shell with the command `db.restaurants.find({city: 'Jeddah'})` and its output, which lists three restaurants: 'Albaik', 'Dominos pizza', and 'Baba Khan', all located in Jeddah. The status bar at the bottom indicates the connection is to 'Atlas atlas-9ori8g-shard-0 [primary] MongoHW'.

MongoDB Compass - cluster0.qdmbx.mongodb.net/MongoHW.restaurants

Local

5 DBS 18 COLLECTIONS

Filter your data

MongoHW

restaurants

admin

MongoHW.restaurants Documents

DOCUMENTS 1 TOTAL SIZE 67B AVG. SIZE 67B INDEXES 1 TOTAL SIZE 4.1KB AVG. SIZE 4.1KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 6 of 6

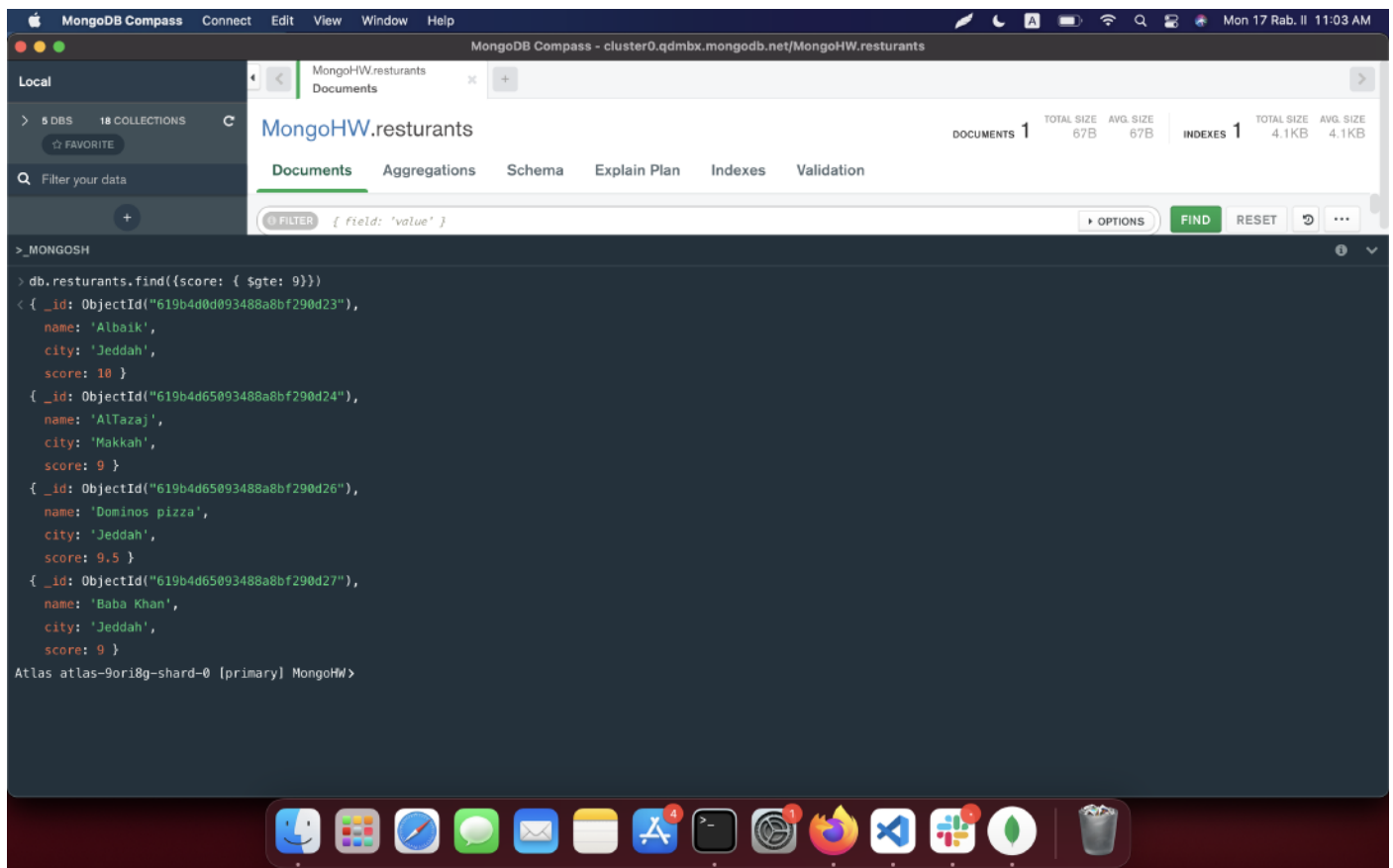
city: "Jeddah"  
score: 10

>\_MONGOSH

```
db.restaurants.find({city: 'Jeddah'})
< { _id: ObjectId("619b4d0d093488a8bf290d23"),
  name: 'Albaik',
  city: 'Jeddah',
  score: 10 }
{ _id: ObjectId("619b4d65093488a8bf290d26"),
  name: 'Dominos pizza',
  city: 'Jeddah',
  score: 9.5 }
{ _id: ObjectId("619b4d65093488a8bf290d27"),
  name: 'Baba Khan',
  city: 'Jeddah',
  score: 9 }
```

Atlas atlas-9ori8g-shard-0 [primary] MongoHW>

Display all the restaurants which is achieved a score grater than or equal to 9.



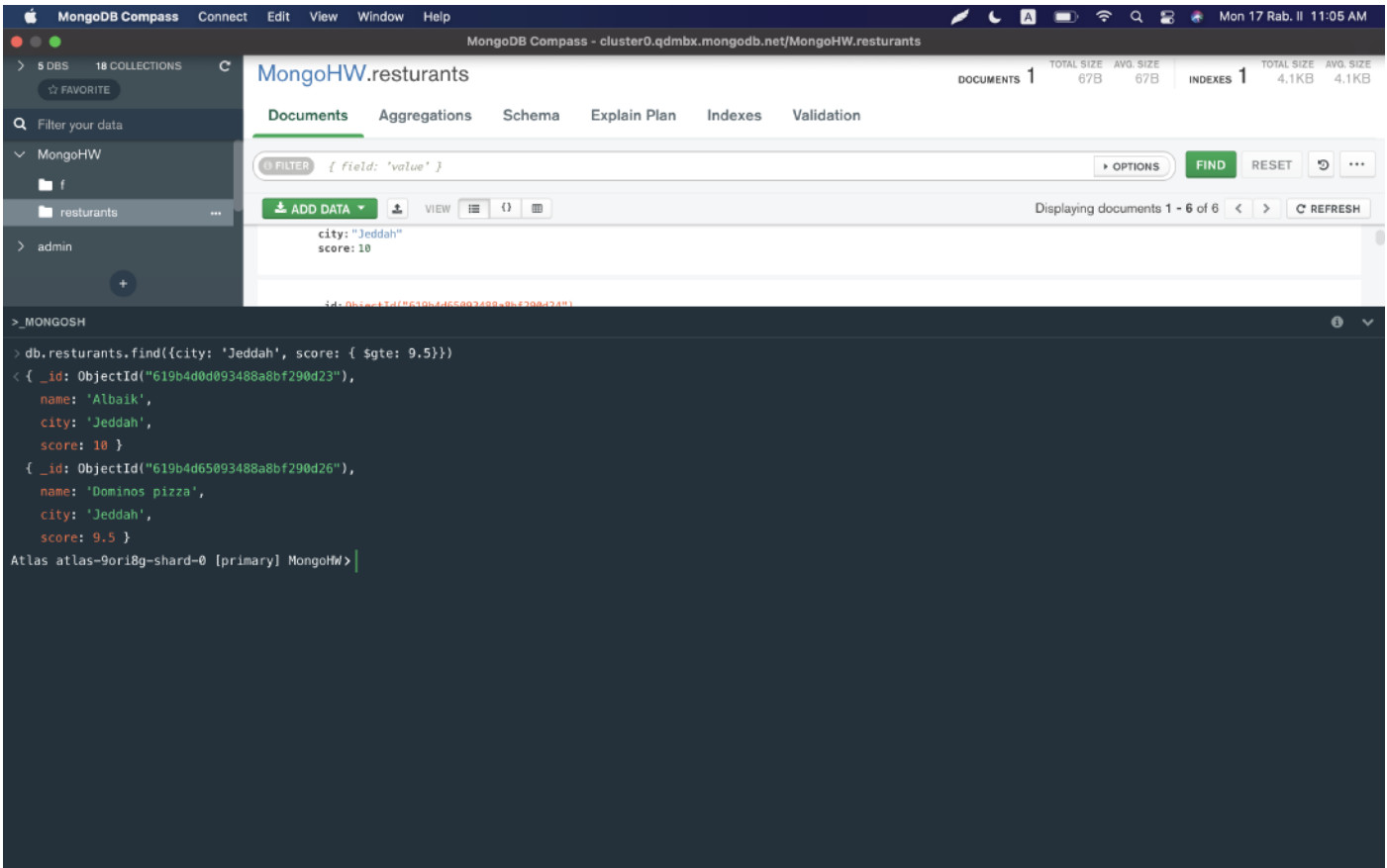
Display all the restaurants which is achieved a score grater than or equal to 8 but less than 9.5.

The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'cluster0.qdmbx.mongodb.net/MongoHW.restaurants'. The left sidebar shows the database structure with 'MongoHW' containing a collection 'f' and a folder 'admin'. The main panel displays the 'Documents' tab for the 'MongoHW.restaurants' collection. A filter is applied: `{ field: 'value' }`. Below the filter, a single document is shown: `{ city: 'Jeddah', score: 10 }`. The bottom panel shows the MongoDB shell with the following query and results:

```
> db.restaurants.find({score: { $gte: 8, $lt: 9.5}})
< { _id: ObjectId("619b4d65093488a8bf290d24"),
  name: 'AlTazaj',
  city: 'Makkah',
  score: 9 }
  { _id: ObjectId("619b4d65093488a8bf290d25"),
  name: 'KFC',
  city: 'AlMadinah',
  score: 8 }
  { _id: ObjectId("619b4d65093488a8bf290d27"),
  name: 'Baba Khan',
  city: 'Jeddah',
  score: 9 }
  { _id: ObjectId("619b4d65093488a8bf290d28"),
  name: 'Krispy Kreme',
  city: 'Riyadh',
  score: 8.5 }
Atlas atlas-9ori8g-shard-0 [primary] MongoHW>
```



Display all the restaurants which is in Jeddah and achieved a score grater than or equal to 9.5.



Display all the restaurants which is contain 'Al' as first two letters of its name.

The screenshot shows the MongoDB Compass interface. On the left, the database structure is visible: 5 DBS, 18 COLLECTIONS. The 'MongoHW' database is selected, and the 'restaurants' collection is highlighted. The main panel displays the 'MongoHW.restaurants' collection with a filter bar containing the query `{ field: 'value' }`. Below the filter bar, there are buttons for 'ADD DATA', 'VIEW', and 'REFRESH'. The document list shows three documents:

- `city: "Jeddah"`  
`score: 10`
- `_id: ObjectId("619b4d65093488a8bf290d24")`  
`name: "AlTazaj"`  
`city: "Makkah"`  
`score: 9`
- `_id: ObjectId("619b4d65093488a8bf290d25")`  
`name: "KFC"`  
`city: "AlMadinah"`  
`score: 8`

At the bottom, the MongoDB shell shows the command `db.restaurants.find({name: {$regex: /^Al/}})` and its output, which lists the documents for 'Albaik', 'AlTazaj', and 'KFC'.

```
>_MONGOSH
> db.restaurants.find({name: {$regex: /^Al/}})
< { _id: ObjectId("619b4d0d093488a8bf290d23"),
  name: 'Albaik',
  city: 'Jeddah',
  score: 10 }
{ _id: ObjectId("619b4d65093488a8bf290d24"),
  name: 'AlTazaj',
  city: 'Makkah',
  score: 9 }
Atlas atlas-9ori8g-shard-0 [primary] MongoHW>
```

Display all the restaurants which is contains 'k' somewhere in its name, either k is big or small.

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: 5 DBS, 18 COLLECTIONS, and a tree view under 'MongoHW' containing 'f' and 'restaurants'. The main panel is titled 'MongoHW.restaurants' and shows a filter bar with the query `{ field: 'value' }`. Below the filter, there are buttons for 'ADD DATA', 'VIEW', and 'REFRESH'. The document list shows two documents: one with `city: 'Jeddah'` and `score: 10`, and another with `_id: ObjectId('619b4d65093488a8bf290d24')`, `name: 'AlTazaj'`, `city: 'Makkah'`, and `score: 9`. The bottom panel shows the MongoDB shell with the command `> db.restaurants.find({name: {$regex: /[k/i}})` and its output, which lists four restaurants: 'Albaik' (Jeddah, score 10), 'KFC' (AlMadinah, score 8), 'Baba Khan' (Jeddah, score 9), and 'Krispy Kreme' (Riyadh, score 8.5).

MongoDB Compass - cluster0.qdmbx.mongodb.net/MongoHW.restaurants

5 DBS 18 COLLECTIONS

Filter your data

MongoHW

restaurants

admin

local

project\_db

quiz\_app

MongoHW.restaurants

DOCUMENTS 1 TOTAL SIZE 67B AVG. SIZE 67B INDEXES 1 TOTAL SIZE 4.1KB AVG. SIZE 4.1KB

Documents Aggregations Schema Explain Plan Indexes Validation

{ field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 6 of 6

city: "Jeddah"  
score: 10

\_id: ObjectId("619b4d65093488a8bf290d24")  
name: "AlTazaj"  
city: "Makkah"  
score: 9

>\_MONGOSH

```
> db.restaurants.find({name: {$regex: /[k/i}})
{ _id: ObjectId("619b4d0d093488a8bf290d23"),
  name: 'Albaik',
  city: 'Jeddah',
  score: 10 }
{ _id: ObjectId("619b4d65093488a8bf290d25"),
  name: 'KFC',
  city: 'AlMadinah',
  score: 8 }
{ _id: ObjectId("619b4d65093488a8bf290d27"),
  name: 'Baba Khan',
  city: 'Jeddah',
  score: 9 }
{ _id: ObjectId("619b4d65093488a8bf290d28"),
  name: 'Krispy Kreme',
  city: 'Riyadh',
  score: 8.5 }
```

Atlas atlas-9ori8g-shard-0 [primary] MongoHW>

Display all the restaurants by arranged the name of the restaurants in ascending order.

The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'cluster0.qdmbx.mongodb.net/MongoHW.restaurants'. The left sidebar shows the database structure with '5 DBS' and '18 COLLECTIONS'. The main panel displays the 'MongoHW.restaurants' collection with a filter of '{ field: 'value' }'. The 'Documents' tab is active, showing a list of restaurant documents sorted by name in ascending order. The documents are as follows:

_id	name	city	score
ObjectId("619b4d65093488a8bf290d24")	AlTazaj	Makkah	9
ObjectId("619b4d0d093488a8bf290d23")	Albaik	Jeddah	10
ObjectId("619b4d65093488a8bf290d27")	Baba Khan	Jeddah	9
ObjectId("619b4d65093488a8bf290d26")	Dominos pizza	Jeddah	9.5
ObjectId("619b4d65093488a8bf290d25")	KFC	AlMadinah	8
ObjectId("619b4d65093488a8bf290d28")	Krispy Kreme	Riyadh	8.5

The bottom panel shows the MongoDB shell with the command `db.restaurants.find({}).sort({name: 1})` and its output, which matches the documents displayed in the Compass interface. The status bar at the bottom indicates the connection is to 'Atlas atlas-9or18g-shard-0 [primary] MongoHW'.

Update Dominos pizza restaurant which is in Jeddah to be in AlMadinah, and the name to Dominos Pizza, and the score to 10

MongoDB Compass - cluster0.qdmbx.mongodb.net/MongoHW.restaurants

Documents 1 TOTAL SIZE 67B AVG. SIZE 67B INDEXES 1 TOTAL SIZE 4.1KB AVG. SIZE 4.1KB

Filter your data { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 6 of 6 REFRESH

```
{ "_id": ObjectId("619b4d05093488a8bf290d23"),  
  "name": "Albaik",  
  "city": "Jeddah",  
  "score": 10 }  
  
{ "_id": ObjectId("619b4d65093488a8bf290d24"),  
  "name": "AlTazaj",  
  "city": "Makkah",  
  "score": 9 }  
  
{ "_id": ObjectId("619b4d65093488a8bf290d25"),  
  "name": "KFC",  
  "city": "AlMadinah",  
  "score": 8 }
```

```
> _MONGOSH  
> db.restaurants.updateOne({name: 'Dominos pizza'}, {$set: {city:'AlMadinah', name:'Dominos Pizza', score:10}})  
< { acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0 }  
> db.find({name: 'Dominos Pizza'})  
✖ TypeError: db.find is not a function  
> db.restaurants.find({name: 'Dominos Pizza'})  
< { _id: ObjectId("619b4d65093488a8bf290d26"),  
  name: 'Dominos Pizza',  
  city: 'AlMadinah',  
  score: 10 }  
Atlas atlas-9or18g-shard-0 [primary] MongoHW>
```

# Delete the KFC restaurant

The screenshot shows the MongoDB Compass interface. On the left, the database structure is visible: 5 DBS, 18 COLLECTIONS. The 'MongoHW' database is expanded, showing a collection named 'restaurants'. The main panel displays the 'MongoHW.restaurants' collection with 1 document. The document is a JSON object with the following fields: `city: "Jeddah"`, `score: 10`, `_id: ObjectId("619b4d65093488a8bf290d24")`, `name: "AlTazaj"`, `city: "Makkah"`, and `score: 9`. The bottom panel shows the MongoDB shell with the command `db.restaurants.deleteOne({name: 'KFC'})` and the response `{ acknowledged: true, deletedCount: 1 }`. The status bar indicates the operation was successful on the primary shard.

MongoDB Compass - cluster0.qdmbx.mongodb.net/MongoHW.restaurants

5 DBS 18 COLLECTIONS

MongoHW

restaurants

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 6 of 6

city: "Jeddah"  
score: 10

\_id: ObjectId("619b4d65093488a8bf290d24")  
name: "AlTazaj"  
city: "Makkah"  
score: 9

\_id: ObjectId("619b4d65093488a8bf290d25")  
name: "KFC"  
city: "AlMadinah"  
score: 8

\_id: ObjectId("619b4d65093488a8bf290d26")

```
>_MONGOSH
> db.restaurants.deleteOne({name: 'KFC'})
< { acknowledged: true, deletedCount: 1 }
Atlas atlas-9ori8g-shard-0 [primary] MongoHW
```