**MongoDB Skills Essential Exercise**

[MongoDB](https://docs.mongodb.com/manual/introduction/) is probably the most prevalent NoSQL database choice for the backend of web applications. It is an open source and relatively easy to use. It works well with the modern web applications written in Node.js or Python Django. Enterprise-level web application platforms like Sitecore (works on .NET) also use MongoDB as the backend database.

Working with MongoDB is quite different from relational databases as its query language is completely different from SQL. Good news is that it is not as complicated as it seems. It does the same CURD operation, flat file upload functions, conditional queries and aggregation functions as the relational world. With MongoDB, you don’t need to worry about joining (because it does no join!). If you spend a few hours, you will become pretty good at it.

I created a series of exercises that cover the basic operations that you can execute with the [Mongo Shell](https://docs.mongodb.com/manual/mongo/). It will give you the solid foundation to develop applications on MongoDB.

**EXERCISE**

1. **Create Database**

Create a database called usermanaged, drop it and create it again. Check which database you are currently in.

1. **Create a Collection & Insert a Record**

Create a collection called customers in usermanaged created in Exercise 1 and insert the document below. Check if the document is inserted correctly.

{"firstName":"John",

"lastName":"West",

"email":"john.west@mail.com",

"phone":"032345432134",

"BusinessType": ["Sell", "Sugar", "Drinks"],

"Reference":100,

"Company":"Coca-Cola"}

1. **Bulk Load JSON File**

3-1. Create a collection called transactions in usermanaged (drop if it already exists) and bulk load the data from a json file, transactions.json (see the data at the end of the questions).

3-2. Append the records with the same file, transactions.json

3-3. Upsert the record from the new file called transactions\_upsert.json (see the data at the end of the questions.

1. **Bulk Load CSV File**

Create a collection and load data from a CSV file will multiple rows. Define the keys from the header row.

1. **Query MongoDB with Conditions**

This question uses the collection (transactions) created in Exercise 3.

5-1. Find any record where Name is Tom

5-2. Find any record where total payment amount (Payment.Total) is 400.

5-3. Find any record where price (Transaction.price) is greater than 400.

5-4. Find any record where Note is null or the key itself is missing.

5-5. Find any record where Note exists and its value is null.

5-6. Find any record where the Note key does not exist.

1. **Aggregation with MongoDB**

This question uses the collection (transactions) created in Exercise 3.

6-1. Calculate the total transaction amount by adding up Payment.Total in all records.

6-2. Get the total price per record by adding up the price values in the Transaction array (Transaction.price).

6-3. Calculate total payments (Payment.Total) for each payment type (Payment.Type).

6-4. Find the max Id.

6-5. Find the max price (Transaction.price).

1. **CURD Operations**

This question uses the collection (transactions) created in Exercise 3.

7-1. Insert a record below.

7-2. Updating the new inserted record above. Make Name='Updated Record' & Note='Updated!'

7-3. Delete the record inserted above by using Id.

1. **User Creation**

8-1. Create a read only user who can query records from collections from all databases.

8-2. Create a writer user who can create collections and do CURD operations in any collections.

8-3. Create a usermanaged user who can do the writer operation in the usermanaged database and read only for the rest of the databases.

**JSON Files**

Here is the transaction JSON for insert in Exercise 3-1.

Here is the JSON file for upsert in Exercise 3-3.

**ANSWERS**

1. **Create Database**

Use db\_name will create a database if it doesn't exist. You can drop the database with dropDatabase().

[cc lang="bash" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

1. **Create a Collection & Insert a Record**

It really doesn't matter if you put double quote or not. Either format works.

[cc lang="bash" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

1. **Bulk Load JSON File**

You can import data from json, csv or tsv files with mongoimport and export data with mongoexport. Note that these are not for backups as you loose some data type information by writing it to flat files.

You need to quit the Mongo Shell session and use Terminal.

For multiple records, you need jsonArray option. Otherwise you get the error message below.

**Failed: error unmarshaling bytes on document #0: JSON decoder out of sync - data changing underfoot?**

[cc lang="bash" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

Once imported, you can log into the database and check the data.

[cc lang="bash" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

1. **Bulk Load CSV File**

Keys will be created from the header row. --headerline and --type csv are required.

[cc lang="bash" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

1. **Query MongoDB with Conditions**

[cc lang="bash" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

1. **Aggregation with MongoDB**

[cc lang="bash" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

1. **CURD Operations**

[cc lang="bash" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

1. **User Creation**

See [**User Management with MongoDB**](https://www.mydatahack.com/user-management-with-mongodb/).

**REFERENCE**

* [mongoimport](https://docs.mongodb.com/manual/reference/program/mongoimport/)
* [Query and Projection Operators](https://docs.mongodb.com/manual/reference/operator/query/#query-selectors)
* [Greater than operator](https://docs.mongodb.com/manual/reference/operator/query/gt/#op._S_gt)
* [Query for null or missing fields](https://docs.mongodb.com/v3.2/tutorial/query-for-null-fields/)
* [Aggregation Sum Reference](https://docs.mongodb.com/manual/reference/operator/aggregation/sum/)
* [Aggregation Max Reference](https://docs.mongodb.com/manual/reference/operator/aggregation/max/)