12.1 Lesson Summary - Mastering MongoDB

A critical part of data analytics is being able to store and access stored data. When dealing with larger data sets with potentially hundreds of millions of records relational databases utilizing SQL have many benefits but they are not always ideal. If your data is unstructured or if you will need to quickly scale up to storing billions of records then a NoSQL database such as MongoDB may be preferable.

Concept: SQL groups records into tables whereas MongoDB groups records into **Collections**. Where SQL stores records in table rows MongoDB stores data in **Documents**. Document syntax resembles JavaScript objects and Python dictionaries. Documents can have complex structures containing arrays and nested objects just like JavaScript objects and Python dictionaries. To store data about books and authors into a SQL database you would probably create two tables, one for authors and one for books. To store data about books and authors in MongoDB you could create a books collection and add the following documents:

*{ "id": 1,*

*"Title": "The History of Blah",*

*"Author": {*

*"name": "Blah Matic",*

*"email": "blahston@gmail.com",*

*"phone": "901-555-5555"*

*},*

*"Published": 2010*

*},*

*{*

*"id": 2,*

*"Title": "The Chronicles of Blahrnia",*

*"Author": {*

*"name": "Sir Blahston",*

*"email": "blahby@gmail.com",*

*"phone": "212-555-5555"*

*},*

*"Published": 2015*

*}*

* Suppl link: <https://www.mongodb.com/>

Concept: To **view all** MongoDB databases you could use the following command:

*show dbs*

To **create** a MongoDB collection named 'destinations' you could use the following command:

*db.createCollection("destinations")*

To **see all** collections in a database you could use the following commands:

*show collections*

To **insert** a document into the *destinations* collection you could use the following command:

*db.destinations.insert({"continent": "Africa", "country": "Morocco",*

*"major\_cities": ["Casablanca", "Fez", "Marrakech"]})*

To **retrieve** data from the *destinations* collection you could use the following command:

*db.destinations.find()*

* Activity: 01-Ins\_MongoBasics, 02-Stu\_MongoClass

Concept: To **update** data already in a MongoDB database you can use the following commands:

*db.collectionName.update()*

*db.collectionName.updateMany()*

To **delete** data from a MongoDB database you can use the following code:

*db.destinations.remove({"country": "Morocco"}, {justOne: true})*

To remove an entire collection or database you could use the following commands:

*db.destinations.drop()*

*db.dropDatabase()*

* Activity: 03-Ins\_CrudMongo, 04-Stu\_DumpsterDB

Concept: MongoDB Compass offers an easy to use UI to view and edit MongoDB data.

* Suppl link: <https://www.mongodb.com/products/compass>

Concept: You are able to access data in a MongoDB database in Python by using the pymongo library. For example:

*import pymongo*

*conn = 'mongodb://localhost:27017'*

*client = pymongo.MongoClient(conn)*

*db = client.travel*

*collection = db.destinations*

*results = collection.find()*

*for result in results:*

*print(result)*

* Activity: 05-Ins\_PyMongo, 06-Stu\_MongGrove
* Suppl link: <https://www.w3schools.com/python/python_mongodb_getstarted.asp>