# Mongo-DB Shell commands

## 1. Mongo Database and collection creation:

On Docker log on:

docker exec -it mongo bash

**Log in to mongo with root user so that you can create collections**: mongo -u root -p root

**Create a Database** : use Patient

**Create collection**: db.createCollection("medical\_data")

**Insert some docs in the collection**:

| db.medical\_data.insertMany( [  { "Age" : 37, "Sex" : 1, "ChestPain" : 2, "RestBP" : 130, "Chol" : 250, "Fbs" : 0, "restecg" : 0, "MaxHR" : 187, "ExAng" : 0, "Oldpeak" : 3.5, "Slope" : 3, "ca" : 0, "Thal" : 2, "target" : 1} ,  { "Age" : 67, "Sex" : 1, "ChestPain" : 0, "RestBP" : 120, "Chol" : 229, "Fbs" : 0, "restecg" : 2, "MaxHR" : 129, "ExAng" : 1, "Oldpeak" : 2.6, "Slope" : 2, "ca" : 2, "Thal" : 3, "target" : 0 } ] )  **View the collection documents**: db.medical\_data.find( ) 2. Python code: reading dataset: ! pip install pymongo  ! pip install pandas  import pandas as pd  from pymongo import MongoClient  myclient = MongoClient('mongodb://%s:%s@localhost:27017/admin' % ('root', 'root')) ## login to mongodb with root user  patientdb = myclient['Patient']  medical\_col = patientdb['medical\_data']  data = pd.read\_csv('heart.csv')  for i in data.index :  X= data.iloc [i,: ]  medical\_col.insert\_one(X.to\_dict())  Check with Mango command lines that the database is created. |
| --- |
| 3. Create user groups and access control 1.Create super user: use admin;    db.createUser(  {  user: "admin",  pwd: 'admin',  roles: [  { role: "userAdminAnyDatabase", db: "admin" },  { role: "readWriteAnyDatabase", db: "admin" }  ]  }  )  2. configuration authentication: 1- Run : vim /etc/mongod.conf  2- add the [security.authorization](https://www.mongodb.com/docs/manual/reference/configuration-options/#mongodb-setting-security.authorization) configuration file setting :  security:  authorization: enabled  3.Authenticate as admin user : 1.mongo  2.use admin  3.db.auth (‘admin’, ‘admin’ )  4.Switch to Patient database: use Patient |

### 5. Create doctor user:

db.createUser(

{

user: "doctor",

pwd: "rotcod",

roles: [

{ role: "read", db: "Patient" }

]

})

### 6.Create nurse user:

db.createUser(

{

user: "nurse",

pwd: "esrun",

roles: [ ] } )

### 7. Create a view “ Nurse\_col “ from “medical\_data” collection:

This view is a subset of medical\_data table:

db.createView("Nurse\_col", 'medical\_data', [ { $project : { 'ca': 0, 'target' : 0 ,'restecg': 0 } } ] )

### 8.Create NurseRole :

db.createRole( {role : 'NurseRole', privileges: [ {resource : {db : "Patient", collection:'Nurse\_col' } ,actions :[ 'find' ] } ] , roles:[ ] } )

### 9.Grant the role to Nurse user :

db.grantRolesToUser ( 'nurse' , ['NurseRole'] )

### 10.Connect with a groupe f user (doctor or nurse) :

mongo -u <user> -p <user\_password> --authenticationDatabase <db\_name>

Example

mongo -u doctor -p rotcod --authenticationDatabase Patient

db.medical\_data.find( )

## References:

1. <https://www.mongodb.com/docs/manual/tutorial/insert-documents/>
2. <https://www.mongodb.com/docs/manual/reference/method/db.createView/#mongodb-method-db.createView>
3. <https://www.mongodb.com/docs/manual/tutorial/configure-scram-client-authentication/>
4. <https://www.mongodb.com/docs/manual/core/authorization/#std-label-roles>
5. dataset:<https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset>