**MONGO DB CRUD IN PYTHON**

**CREATE CODE**

*# INSERT PYMONGO FROM THE PACKAGE  
# import pymongo  
  
# MongoClient will do all the database operations*from pymongo import MongoClient  
  
*# PROVIDE THE VALUE OF localhost and port number - WHERE THE MONGODB IS RUNNING*client = MongoClient('localhost','27107') *# DEFAULT PORT FOR MONGODB  
  
#USE THE DATABASE*database = client['augustaMongoDb'] *# IF DB NOT AVAILABLE, IT WILL CREATE IT.*print("Database Created !!!!!")  
  
*# COLLECTION IS SIMILAR LIKE TABLE*collection= database['product'] *# PRODUCT IS LIKE TABLE NAME IN SQL*print("Collection Created !!!!!")  
  
*# MONGO WILL CREATE A DATABASE / COLLECTIONS WHEN ITS FIRST REALLY USED  
# SO TO ACTIVATE IT, WE HAVE TO PUT SOME DATA IN TO IT.*prod1={  
 "name":"IPhone 15",  
 "price":1499  
}  
  
*#TWO WAYS OF INSERT  
# INSERT\_ONE WILL INSERT ONLY 1 RECORD, WHICH WILL RETURN A RESULT*result = collection.insert\_one(prod1)  
print(result.inserted\_id) *# MONGO DB WILL CREATE AN UNIQUE ID - LIKE PRIMARY KEY - AND THIS WILL BE DISPLAYED  
  
# INSERT\_ALL WILL INSERT MULTIPLE RECORDS*prod2=[{  
 "name":"IPhone 15",  
 "price":1099  
},  
{  
 "name":"IPhone 15 max",  
 "price":1299  
},  
{  
 "name":"IPhone 15 pro",  
 "price":1499  
},  
 {  
 "name": "IPhone 15 pro max",  
 "price": 1699  
 }  
]  
result = collection.insert\_many(prod2)  
*# MONGO DB WILL CREATE AN UNIQUE ID - LIKE PRIMARY KEY - THIS WILL BE PRINTED FOR ALL THE RECORDS ADDED ABOVE*print(result.inserted\_ids)

**READ CODE**

*# MongoClient will do all the database operations*from pymongo import MongoClient  
  
*# PROVIDE THE VALUE OF localhost and port number - WHERE THE MONGODB IS RUNNING*client = MongoClient('localhost','27107') *# DEFAULT PORT FOR MONGODB  
  
#USE THE DATABASE (AS ITS ALREADY CREATED IN mongoCreate.py PROGRAME, NOW WE WILL USE ONLY*database = client['augustaMongoDb']  
  
*# COLLECTION WILL BE USED NOW, NOT CREATED*collection= database['product']  
  
*#TO RETRIEVE THE FIRST RECORD*print(collection.find\_one())  
  
*# TO RETRIEVE ONE RECORD , VALUES ARE CASESENSITIVE, SO HAVE TO PROVIDE AS THE SAME*cursor = collection.find({"name":"IPhone 15"})  
for eachrec in cursor:  
 print(eachrec)  
  
*# TO RETRIEVE ALL RECORDS*cursor = collection.find()  
for eachrec in cursor:  
 print(eachrec)

**UPDATE CODE**

*# MongoClient will do all the database operations*from pymongo import MongoClient  
  
*# PROVIDE THE VALUE OF localhost and port number - WHERE THE MONGODB IS RUNNING*client = MongoClient('localhost','27107') *# DEFAULT PORT FOR MONGODB  
  
#USE THE DATABASE (AS ITS ALREADY CREATED IN mongoCreate.py PROGRAME, NOW WE WILL USE ONLY*database = client['augustaMongoDb']  
  
*# COLLECTION WILL BE USED NOW, NOT CREATED*collection= database['product']  
  
*# SINGLE RECORD UPDATE*filter= {"name":"IPhone 15"}  
collection.update\_one(filter, {"$set":"999"}) *#CHANGING THE PRICE FOR THE FILTERED RECORD  
  
# MULTIPLE RECORD UPDATE*filter= {"name":"IPhone 15"} *# IF THE COLLECTION HAS MANY WITH NAME IPHONE 15 THEN IT CAN BE USED.*result = collection.update\_many(filter, {"$set":"999"}) *#CHANGING THE PRICE FOR THE FILTERED RECORD FOR ABOVE IF HAVE MORE THAN 1*print(result.modified\_count) *# WILL TELL US THE NOW OF ROWS UPDATED*

**DELETE CODE**

*# MongoClient will do all the database operations*from pymongo import MongoClient  
  
*# PROVIDE THE VALUE OF localhost and port number - WHERE THE MONGODB IS RUNNING*client = MongoClient('localhost','27107') *# DEFAULT PORT FOR MONGODB  
  
#USE THE DATABASE (AS ITS ALREADY CREATED IN mongoCreate.py PROGRAME, NOW WE WILL USE ONLY*database = client['augustaMongoDb']  
  
*# COLLECTION WILL BE USED NOW, NOT CREATED*collection= database['product']  
  
*# SINGLE RECORD DELETE*filter= {"name":"IPhone 15"}  
collection.delete\_one(filter, {"$set":"999"}) *#CHANGING THE PRICE FOR THE FILTERED RECORD  
  
# MULTIPLE RECORD DELETE*filter= {"name":"IPhone 15"} *# IF THE COLLECTION HAS MANY WITH NAME IPHONE 15 THEN IT CAN BE USED.*result = collection.delete\_many({filter}) *# DELETE ALL THE RECORDS WITH NAME = IPHONE 15*print(result.modified\_count) *# WILL TELL US THE NOW OF ROWS UPDATED*