**DATASET:** [**https://data.world/alexandra/music-composers**](https://data.world/alexandra/music-composers)

**TASK 1: Connecting to mongodb server using cmd.**

**Command** : mongosh "mongodb+srv://cluster0.ezjbfwh.mongodb.net/" --apiVersion 1 --username praketpatitiwarimath20

Enter your username

****

**TASK 2: Switching to the required database.**

**Command** : use Lab.

Enter the name of desired database

****

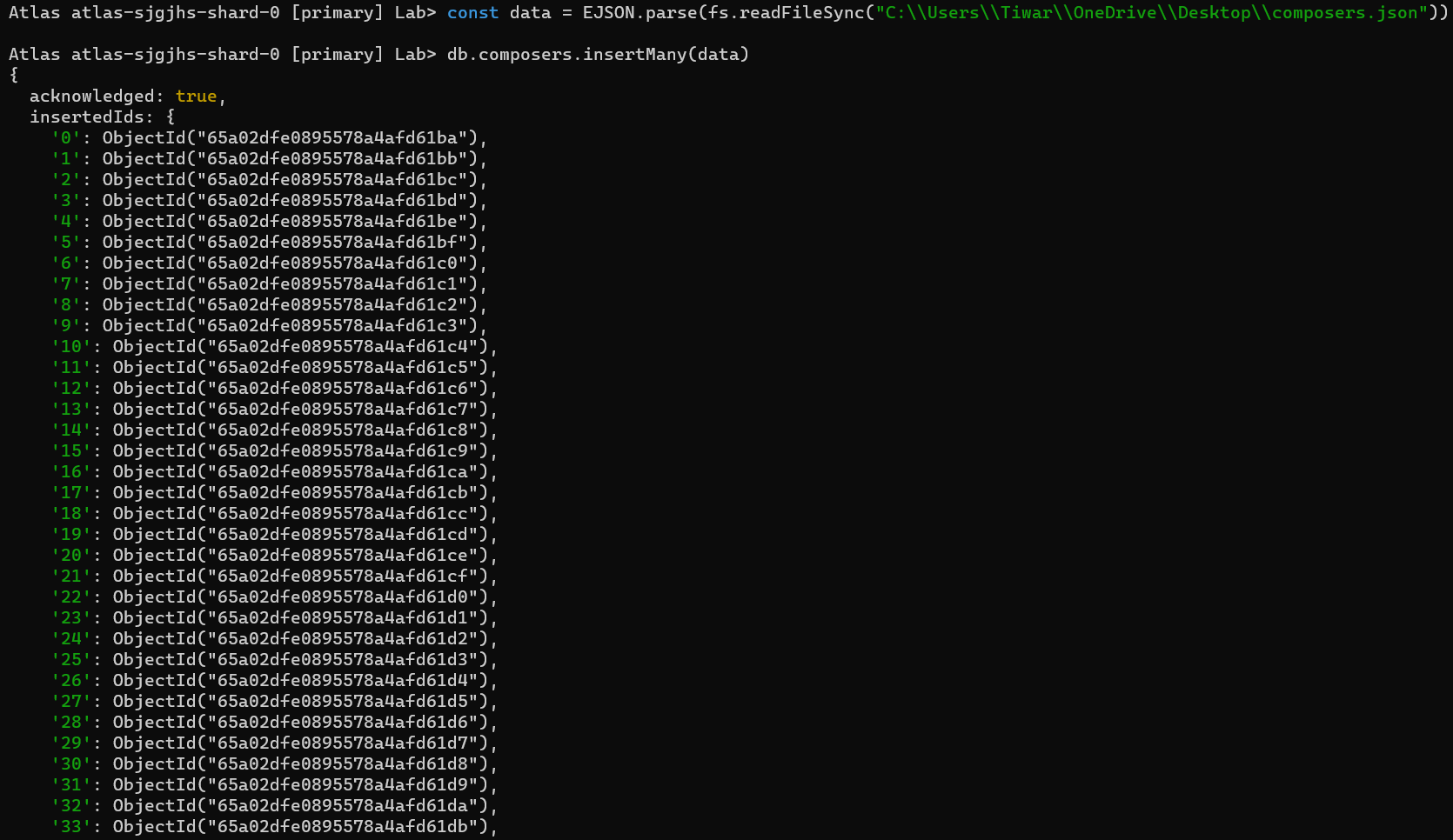
**TASK 3: Loading JSON file, containing data , into the database**

**Command line1**: const data = EJSON.parse(fs.readFileSync("C:\\Users\\Tiwar\\OneDrive\\Desktop\\composers.json"))

Path to file

**Command line2**: db.composers.insertMany(data)

Collection name





**TASK 4: Displaying the dataset**

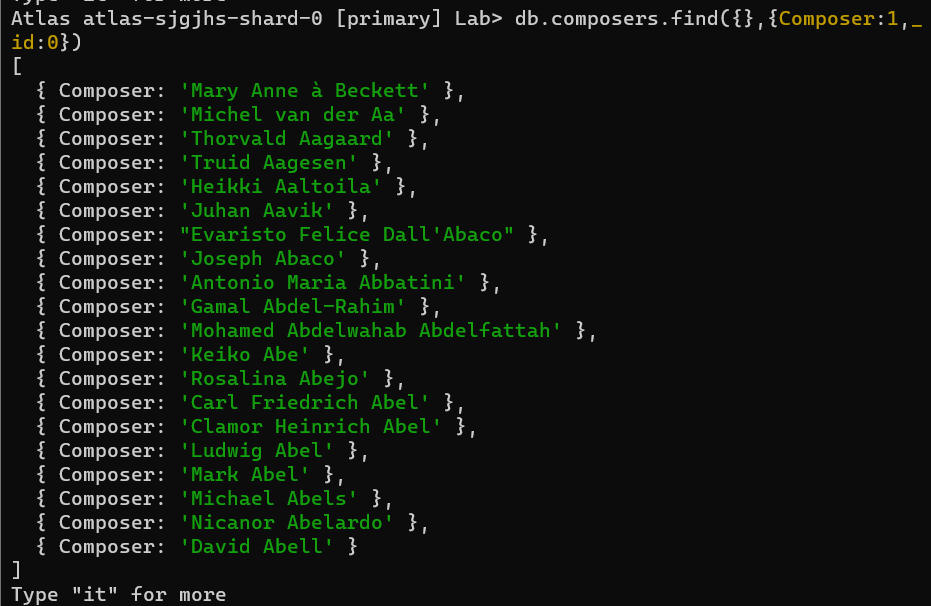
**Command** : db.composers.find()

****

**TASK 5: Projecting specific column.**

**Command** : db.composers.find({},{Composer:1,\_id:0})

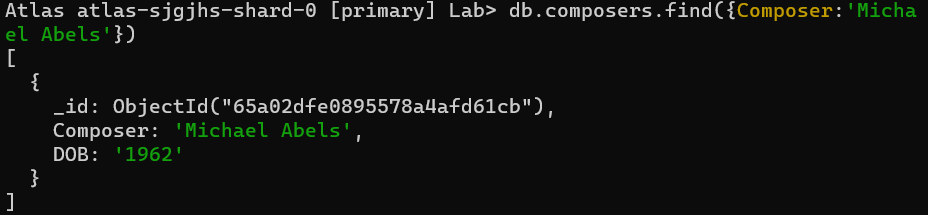
Composer to be displayed, id to be not



**TASK 6: Searching for specific data entry with single constraint.**

**Command** : db.composers.find({Composer:'Michael Abels'})

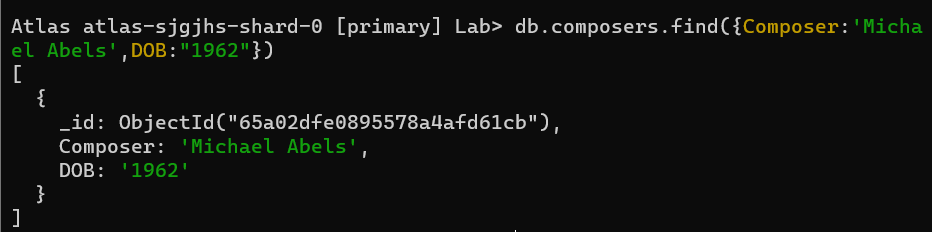
Searching for composer name ‘Michael Abels’

****

**TASK 7: Searching for specific data entry with 2 constraints.**

**Command** : db.composers.find({Composer:'Michael Abels',DOB:"1962"})

Entry with Composer ‘Michael Abels’ and DOB: 1962

****

**TASK 8: Getting number of entries in dataset.**

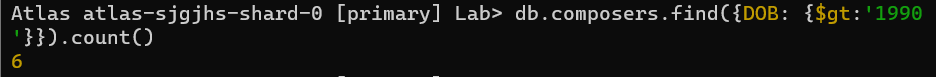
**Command** : db.composers.find().count()



**TASK 9: Getting number of specific occurences in dataset.**

**Command** : db.composers.find({DOB: {$gt:'1990'}}).count()

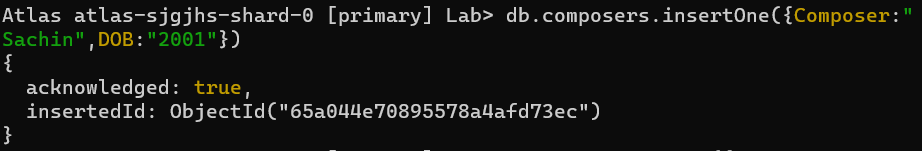
Count of entries with DOB>1990

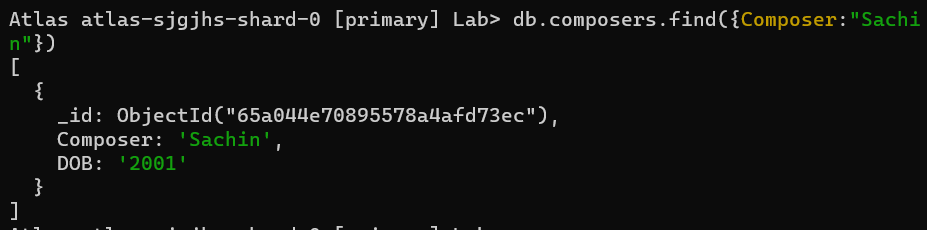
****

**TASK 10: Inserting 1 Entry in Dataset.**

**Command** : db.composers.insertOne({Composer:"Sachin",DOB:"2001"})

Inserting data with Composer: Sachin and DOB:2001

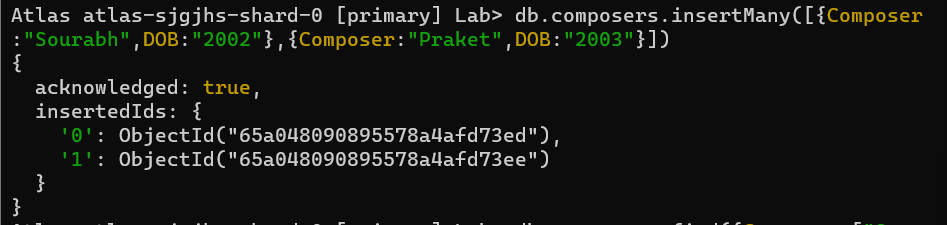




**TASK 11: Inserting more than 1 Entry in Dataset.**

**Command** : db.composers.insertMany([{Composer:"Sourabh",DOB:"2002"},{Composer:"Praket",DOB:"2003"}])

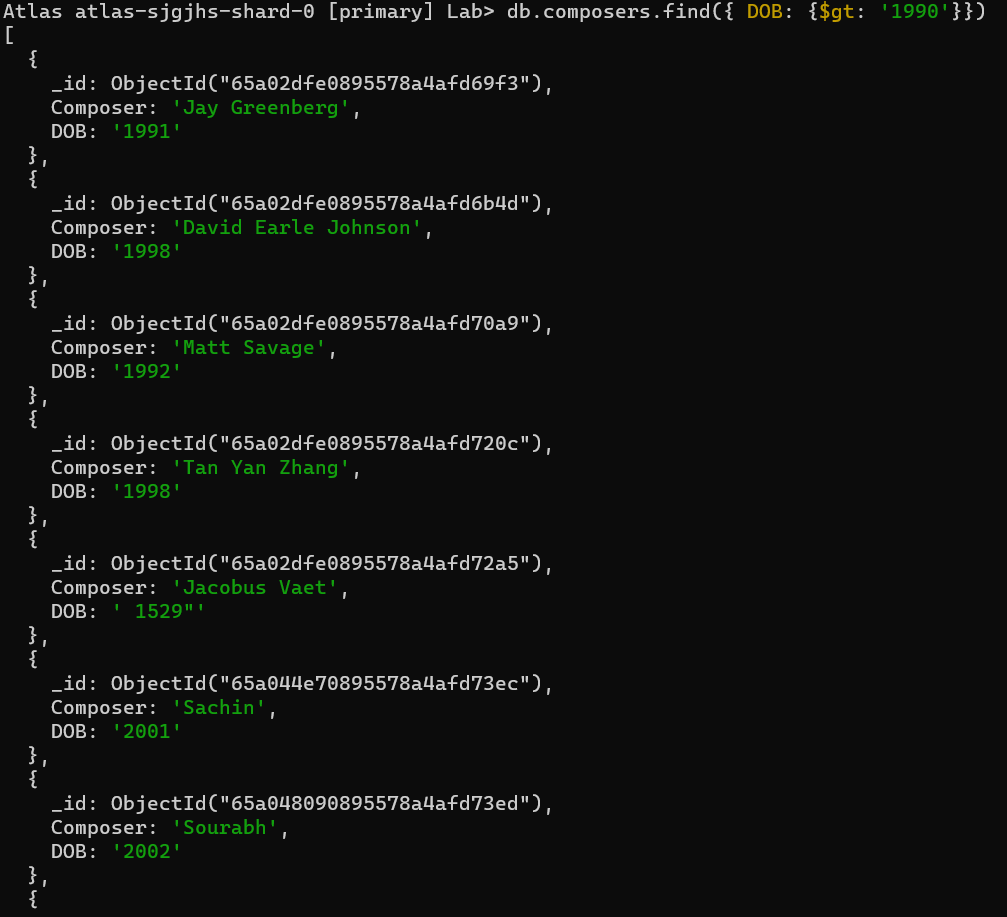
Inserting 2 Entries.



**TASK 12: Selecting data with DOB > 1990.**

**Command** : db.composers.find({ DOB: {$gt: '1990'}})

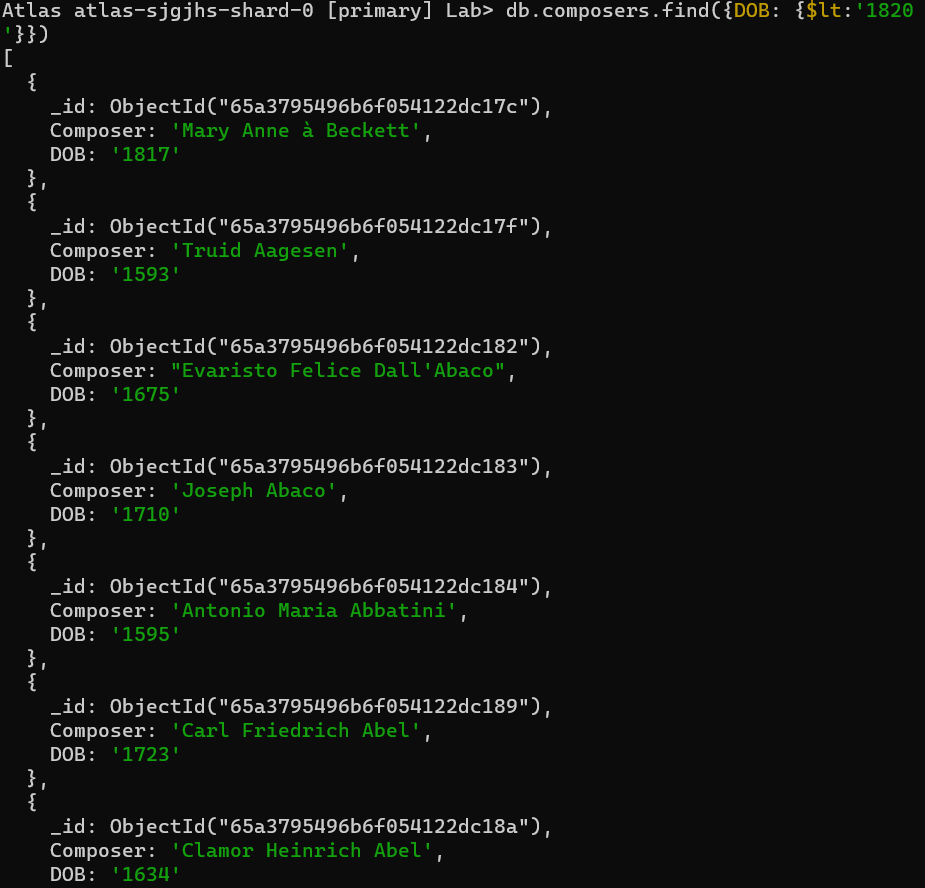
All entries with DOB > 1990. ‘gt’ means greater than



**TASK 13: Selecting data with DOB < 1820 .**

**Command** : db.composers.find({DOB: {$lt:'1820'}})

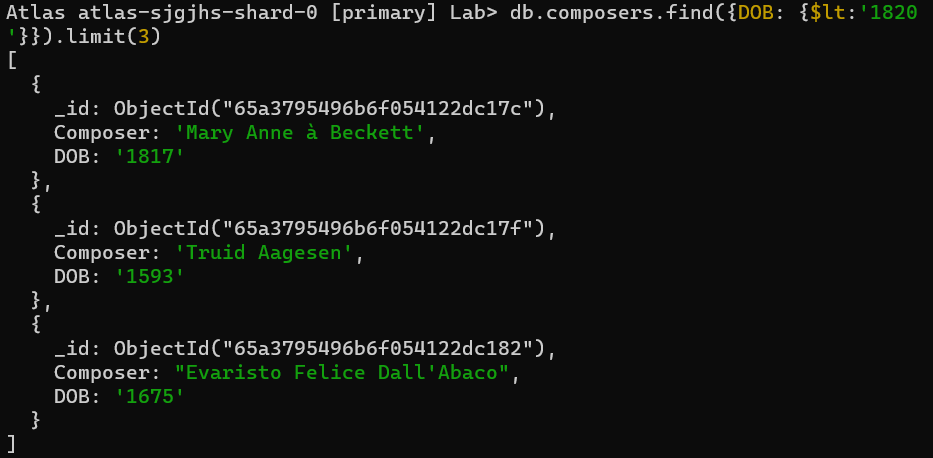
All entries with DOB < 1820. ‘lt’ means less than



**TASK 14: Selecting data with DOB < 1820 and displaying only first 3 matching results.**

**Command** : db.composers.find({DOB: {$lt:'1820'}}).limit(3)

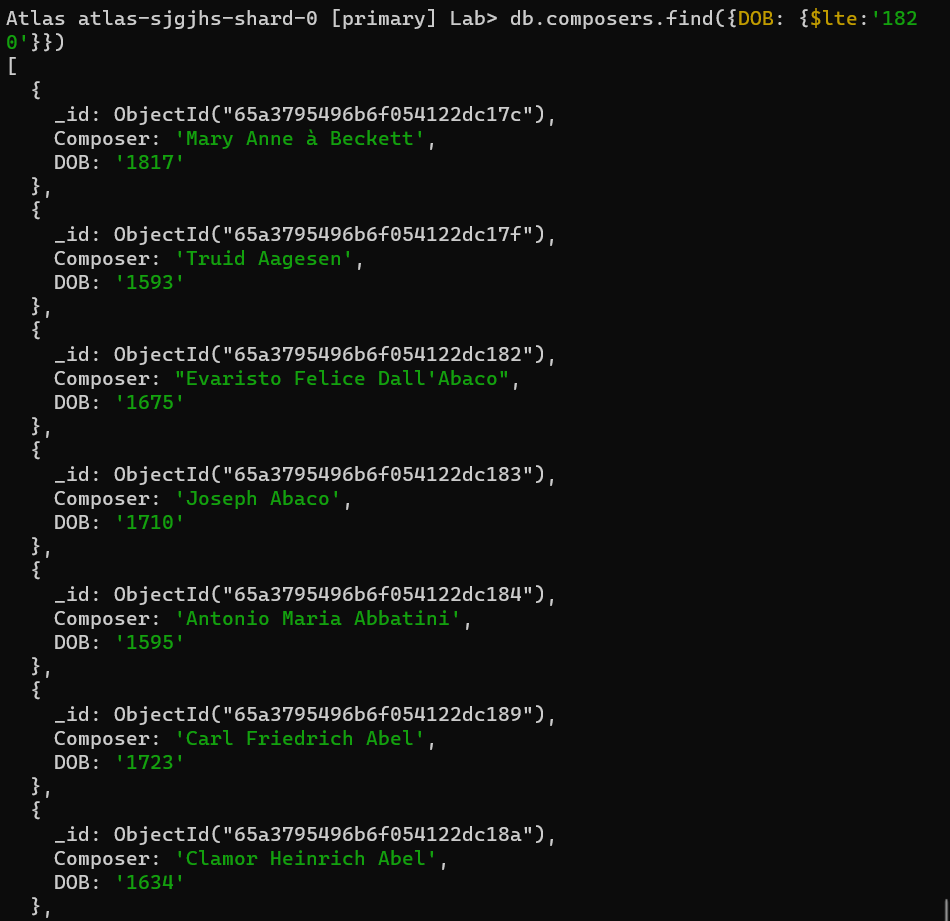
limit() command limits the number of results displayed

****

**TASK 15: Selecting data with DOB <= 1820 .**

**Command** : db.composers.find({DOB: {$lte:'1820'}})

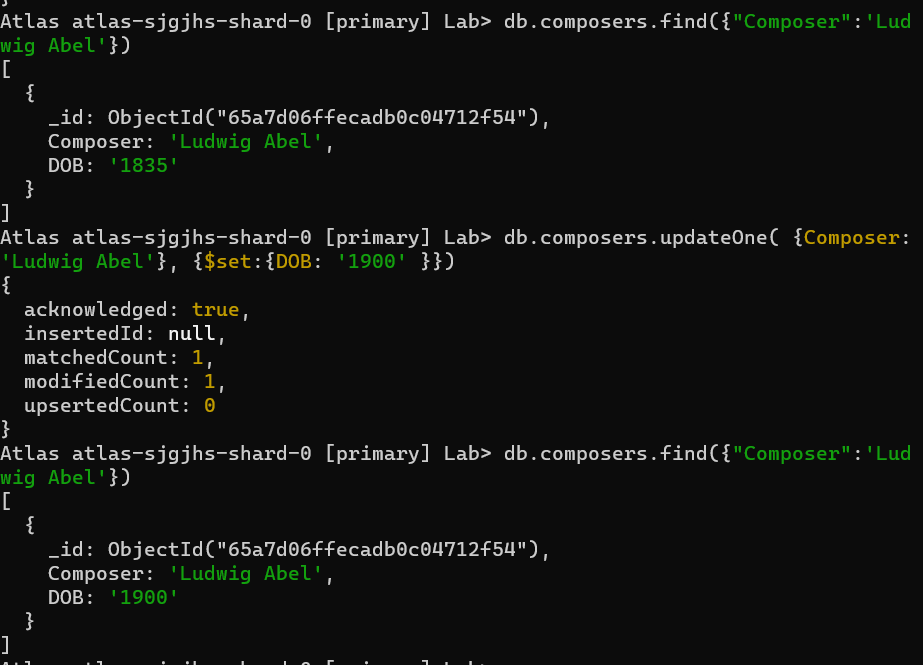
‘lte’ means less than equal to



**TASK 16: Updating only One entry.**

**Command** : db.composers.updateOne( {Composer:'Ludwig Abel'}, {$set:{DOB: '1900' }})

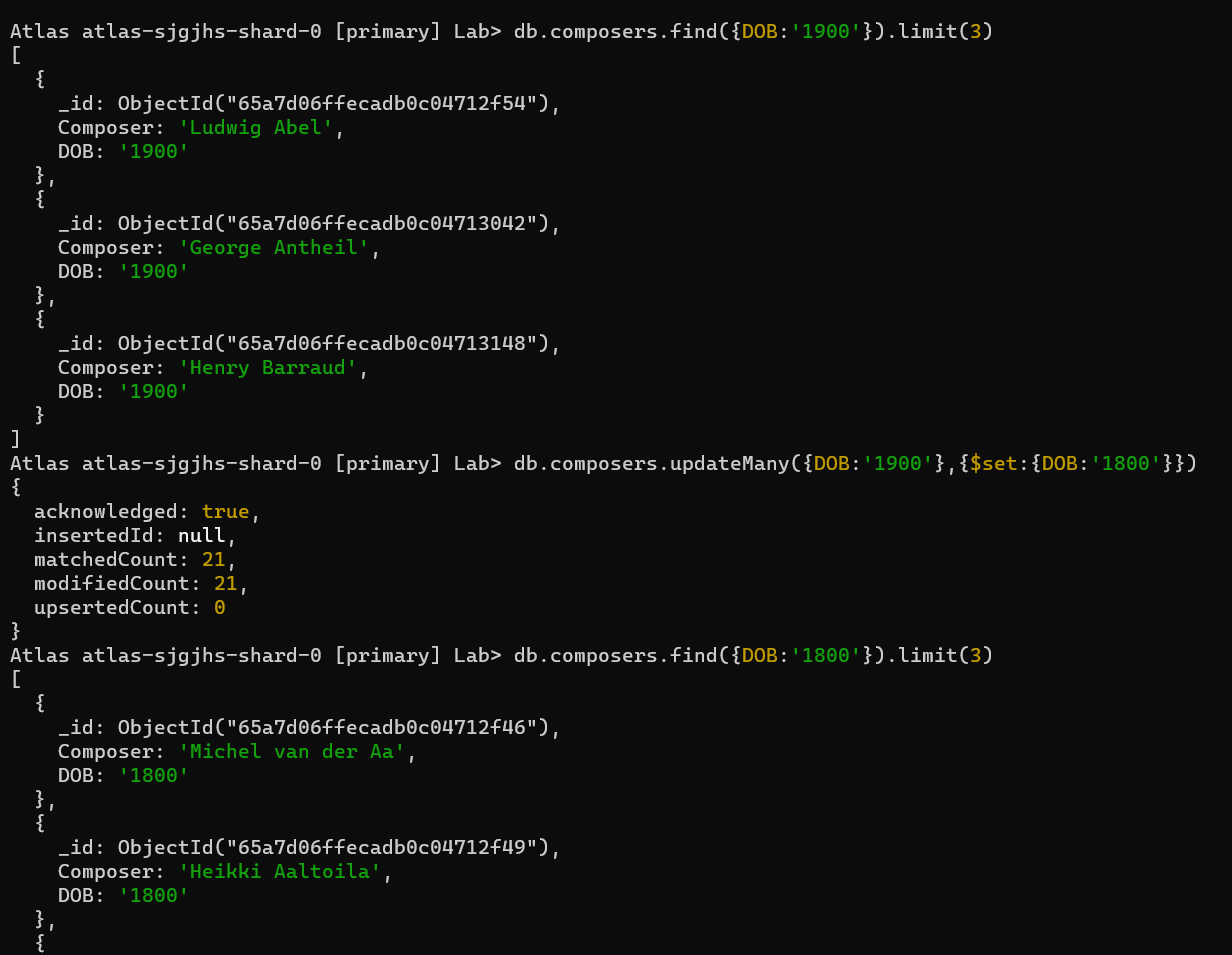
updateOnly updates only 1 entry at a time.

****

**TASK 17: Updating many entries at a time.**

**Command** : db.composers.updateMany({DOB:'1900'},{$set:{DOB:'1800'}})

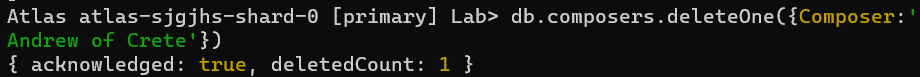
updateMany updates many entries at a time.

****

**TASK 18: Deleting 1 Entry from Dataset.**

**Command** : db.composers.deleteOne({Composer:'Andrew of Crete'})

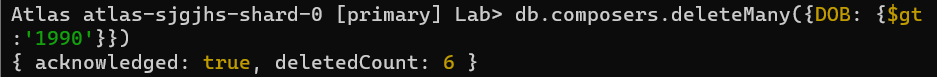
deleteOne() deletes only 1 entry



**TASK 19: Deleting more than 1 Entry from Dataset.**

**Command** : db.composers.deleteMany({DOB: {$gt:'1990'}})

deleteMany() deletes many entries at a time



**TASK 10: Selecting all the entries with names starting with M and getting its count.**

**Command** : db.composers.find({Composer:{$regex:"^M"}}).count()

