EXERCISE: 9

Innovating with Data: MongoDB Atlas Cluster Creation & Python Interaction:

Aim:

• To create a free MongoDB cluster on MongoDB Atlas.

• To set up a user with privileges for database access.

• To create a database, collection, and document within the MongoDB cluster.

• To interact with the MongoDB cluster using Python code for basic operations.

Procedure:

• Sign Up/Login to MongoDB Atlas:

• Visit the MongoDB Atlas website.
Sign up for a new account or log in if you already have one.
Create a Free Cluster:

• Once logged in, click on "Try Free" or "Build a Cluster".
Click "Create Cluster".
Create a Database User:

• Navigate to the "Database Access" under the "Security" tab in the left sidebar.
Click "Add New Database User".
Click "Add User" to create the user.

• Go to "Network Access" under the "Security" tab.
Click "Add IP Address" or "Add IP Address to List".
Choose "Allow Access from Anywhere" or specify an IP range.
Save the changes.
Create a Database and Collection:

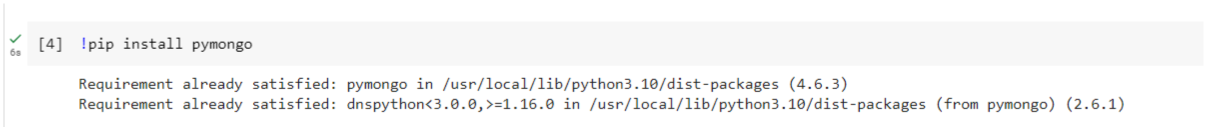
• After your cluster is ready, go to the "Clusters" tab and click on "Collections".
Insert a Document:

• With the "students_records" collection selected, click "Insert Document".
Input the document data in JSON format (e.g., {"name": "John Doe", "age": 20}).
Click "Insert" and verify and utilize:

CODE:

B) Interacting with the MongoDB Cluster using Python Code:

1. Install the pymongo library for interacting with MongoDB.
pip install pymongo



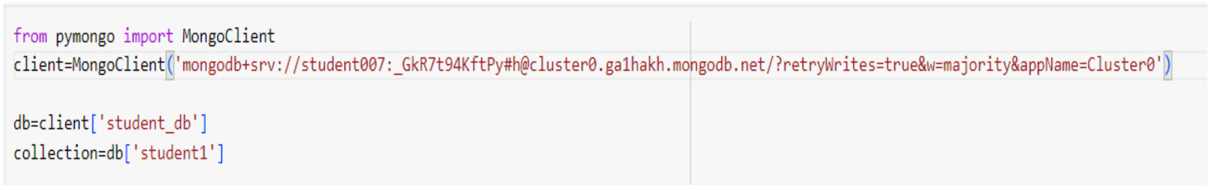
```
[4] !pip install pymongo

    Requirement already satisfied: pymongo in /usr/local/lib/python3.10/dist-packages (4.6.3)
    Requirement already satisfied: dnspython<3.0.0,>=1.16.0 in /usr/local/lib/python3.10/dist-packages (from pymongo) (2.6.1)
```

2. Import MongoClient from the pymongo library in your Python code.
from pymongo import MongoClient

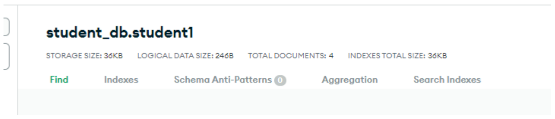
```
from pymongo import MongoClient
```

3. Connect to the "Student_db" database using its connection string obtained from MongoDB Atlas.

client = MongoClient("Connection String")
db = client.Student_db



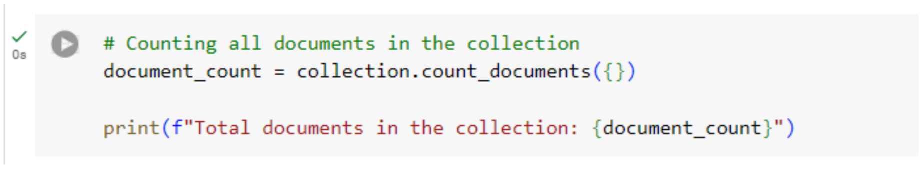
```
from pymongo import MongoClient
client=MongoClient('mongodb+srv://student007:_GkR7t94KftPy#h@cluster0.ga1hakh.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0')

db=client['student_db']
collection=db['student1']
```



student_db.student1

STORAGE SIZE: 36KB   LOGICAL DATA SIZE: 246B   TOTAL DOCUMENTS: 4   INDEXES TOTAL SIZE: 36KB

4. Perform the following operations
i) Count documents
document_count = db.students_records.count_documents({})
print("Total documents:", document_count)



```
# Counting all documents in the collection
document_count = collection.count_documents({})

print(f"Total documents in the collection: {document_count}")
```

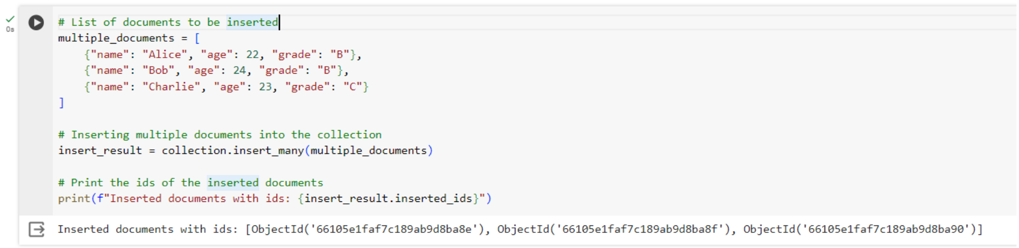
ii) Create a new document

new_document = {"name": "John Doe", "age": 25, "grade": "A"}
db.students_records.insert_one(new_document)

```
document={"name": "John Doe", "age": 25, "grade": "A"}
insert_doc=collection.insert_one(document)

print("inserted Document Successfully")
```

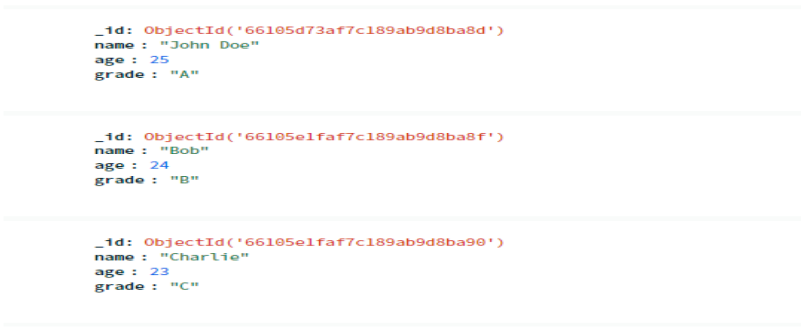
iii) Insert single and multiple documents:

For Insert Single document:
new_document = {"name": "John Doe", "age": 25, "grade": "A"}
db.students_records.insert_one(new_document)
For Insert Multiple documents:
multiple_documents = [
{"name": "Alice", "age": 22, "grade": "B"},
{"name": "Bob", "age": 24, "grade": "B"},
{"name": "Charlie", "age": 23, "grade": "C"}
]
db.students_records.insert_many(multiple_documents)



```
# List of documents to be inserted
multiple_documents = [
    {"name": "Alice", "age": 22, "grade": "B"},
    {"name": "Bob", "age": 24, "grade": "B"},
    {"name": "Charlie", "age": 23, "grade": "C"}
]

# Inserting multiple documents into the collection
insert_result = collection.insert_many(multiple_documents)

# Print the ids of the inserted documents
print(f"Inserted documents with ids: {insert_result.inserted_ids}")
```

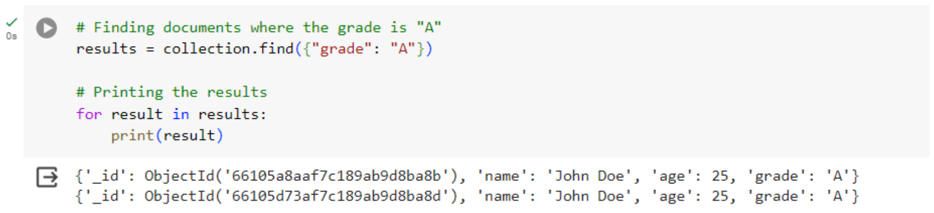
```
Inserted documents with ids: [ObjectId('66105e1faf7c189ab9d8ba8e'), ObjectId('66105e1faf7c189ab9d8ba8f'), ObjectId('66105e1faf7c189ab9d8ba90')]
```



```
_id: ObjectId('66105d73af7c189ab9d8ba8d')
name : "John Doe"
age : 25
grade : "A"

_id: ObjectId('66105e1faf7c189ab9d8ba8f')
name : "Bob"
age : 24
grade : "B"

_id: ObjectId('66105e1faf7c189ab9d8ba90')
name : "Charlie"
age : 23
grade : "C"
```

iv) Find documents

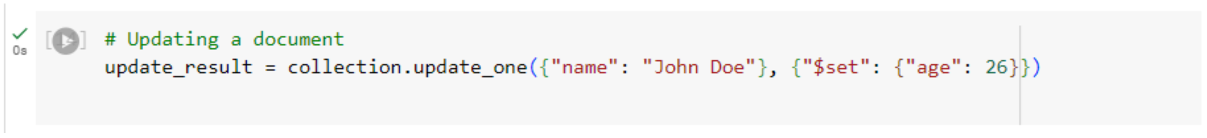
results = db.students_records.find({"grade": "A"})
for result in results:
print(result)



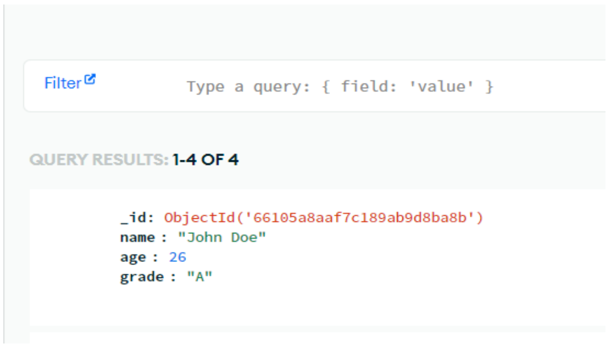
```
# Finding documents where the grade is "A"
results = collection.find({"grade": "A"})

# Printing the results
for result in results:
    print(result)

{'_id': ObjectId('66105a8aaf7c189ab9d8ba8b'), 'name': 'John Doe', 'age': 25, 'grade': 'A'}
{'_id': ObjectId('66105d73af7c189ab9d8ba8d'), 'name': 'John Doe', 'age': 25, 'grade': 'A'}
```

v) Update documents:

db.students_records.update_one({"name": "John Doe"}, {"$set": {"age": 26}})



```
# Updating a document
update_result = collection.update_one({"name": "John Doe"}, {"$set": {"age": 26}})
```



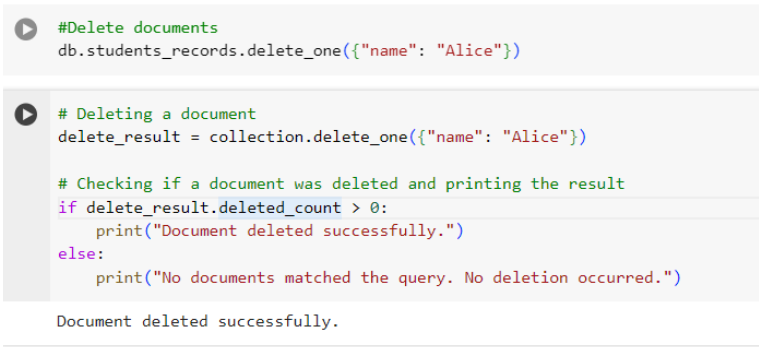
Filter Type a query: { field: 'value' }

QUERY RESULTS: 1-4 OF 4

```
_id: ObjectId('66105a8aaf7c189ab9d8ba8b')
name : "John Doe"
age : 26
grade : "A"
```

vi) Delete documents

db.students_records.delete_one({"name": "Alice"})



```
#Delete documents
db.students_records.delete_one({"name": "Alice"})
```

```
# Deleting a document
delete_result = collection.delete_one({"name": "Alice"})

# Checking if a document was deleted and printing the result
if delete_result.deleted_count > 0:
    print("Document deleted successfully.")
else:
    print("No documents matched the query. No deletion occurred.")

Document deleted successfully.
```

Result:

The above MongoDB Atlas, create a free cluster, and configure a user and network access and Create "Student_db" database, "students_records" collection, and insert documents via the Collections tab is executed and Output is Verified.