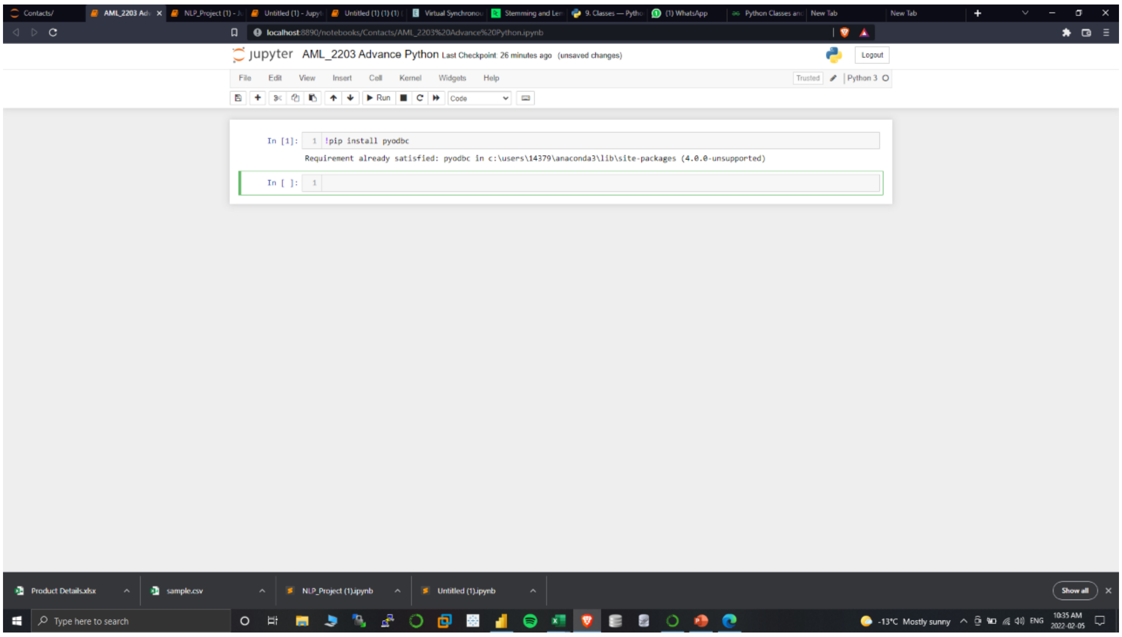
AML 2203 – Advanced Python AI and ML Tools

Assignment – 1b

Aadarsha Chapagain
Ganesh chaulagain
Piyush Bhatia
Rishi Phaneendra Varma

<u>Installation</u>

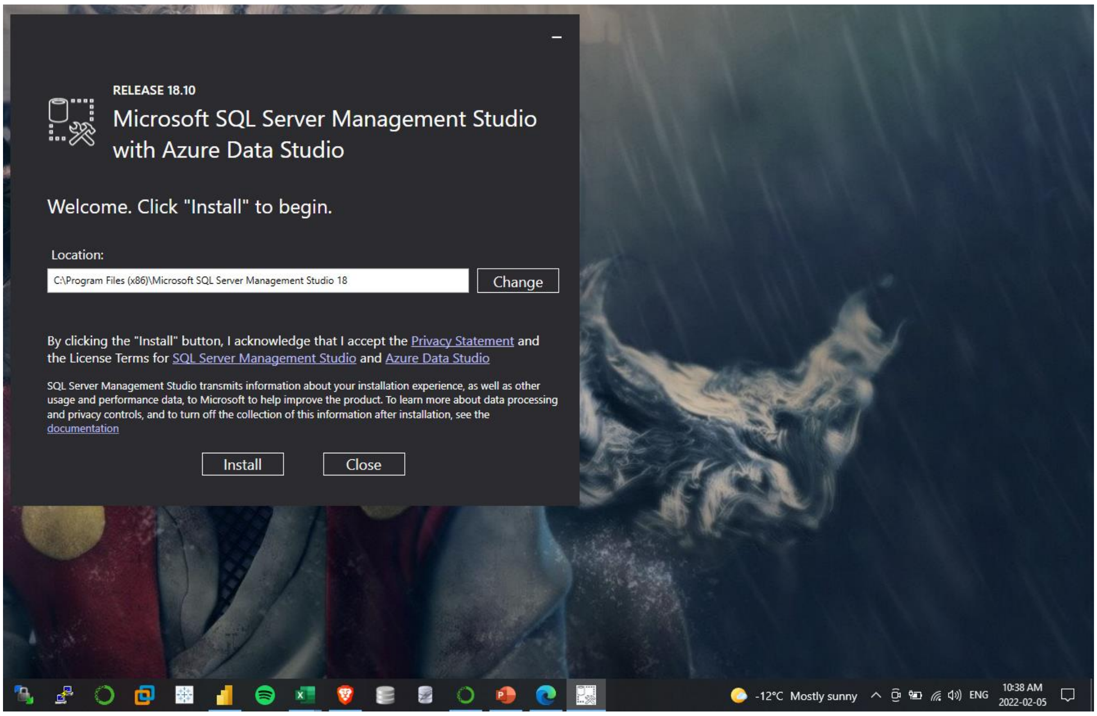
1. **Installing "pyodbc".**



jupyter AML_2203 Advance Python Last Checkpoint: 26 minutes ago (unsaved changes)

```
In [1]: !pip install pyodbc
        Requirement already satisfied: pyodbc in c:\users\14379\anaconda3\lib\site-packages (4.0.0-unsupported)

In [ ]:
```

2. **Downloading Microsoft SQL Server Management Studio.**



RELEASE 18.10

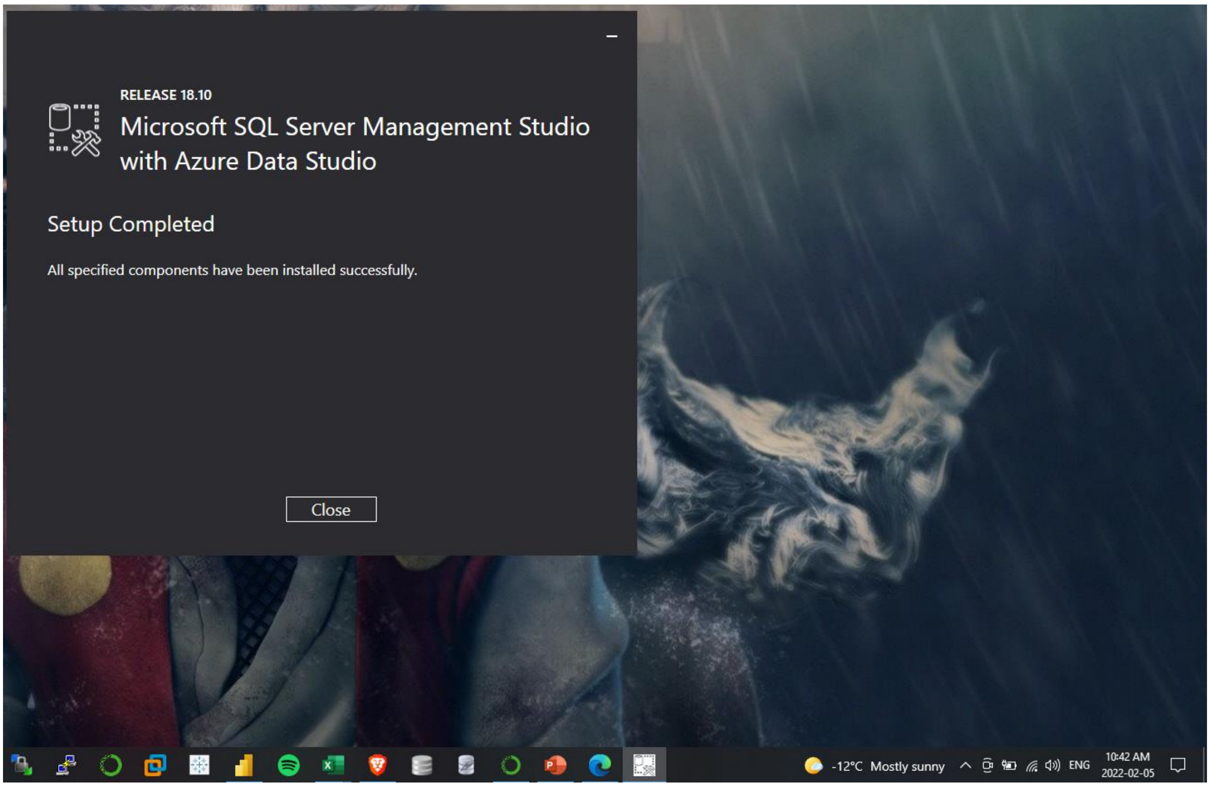
Microsoft SQL Server Management Studio with Azure Data Studio

Welcome. Click "Install" to begin.

Location:
C:\Program Files (x86)\Microsoft SQL Server Management Studio 18

By clicking the "Install" button, I acknowledge that I accept the Privacy Statement and the License Terms for SQL Server Management Studio and Azure Data Studio

SQL Server Management Studio transmits information about your installation experience, as well as other usage and performance data, to Microsoft to help improve the product. To learn more about data processing and privacy controls, and to turn off the collection of this information after installation, see the documentation

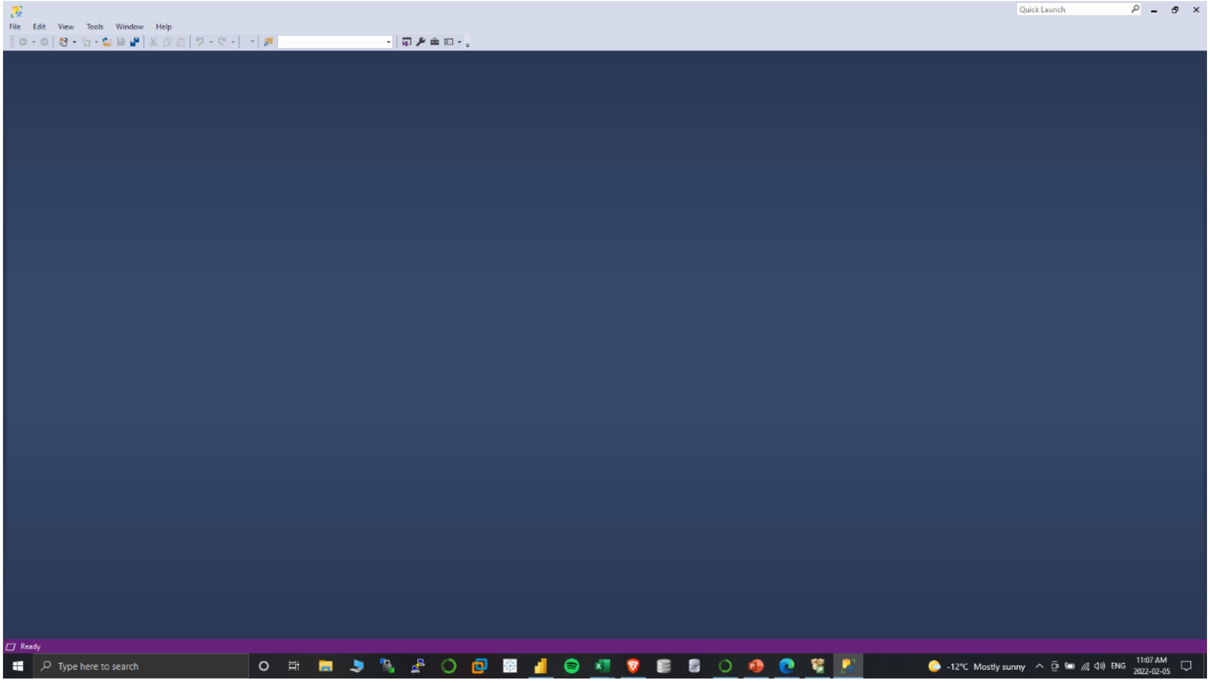


RELEASE 18.10

Microsoft SQL Server Management Studio with Azure Data Studio

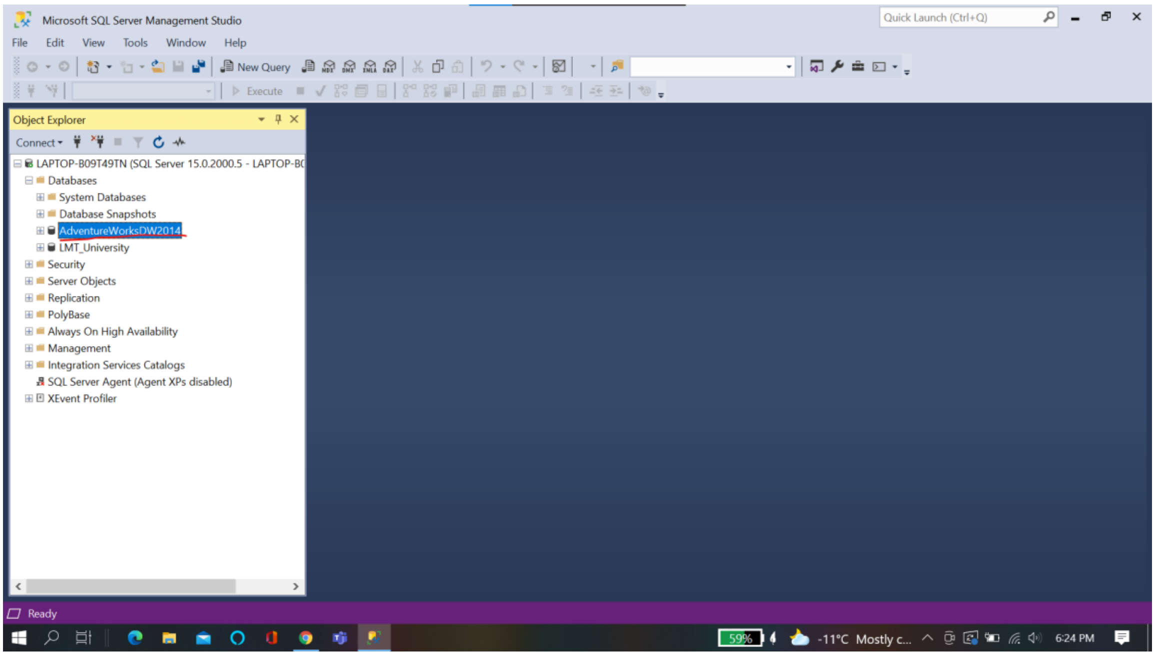
Setup Completed

All specified components have been installed successfully.

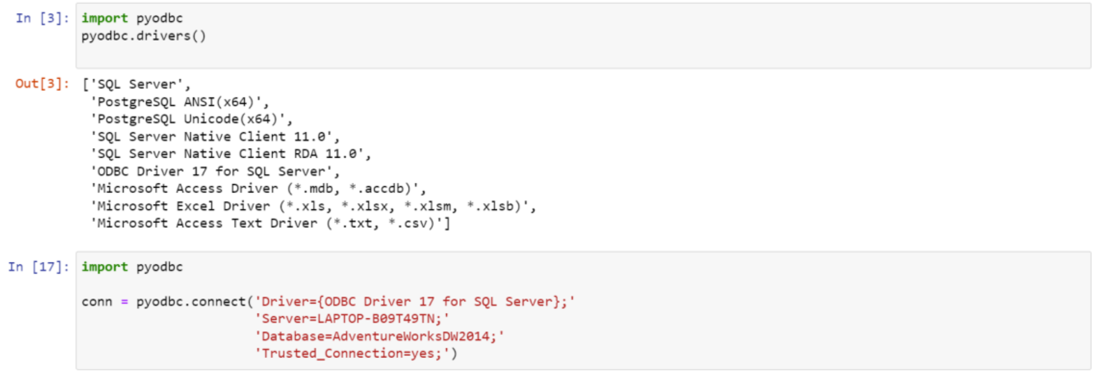


Execution

Apply CRUD operations on the adventureworks2019 database file using pyodbc.



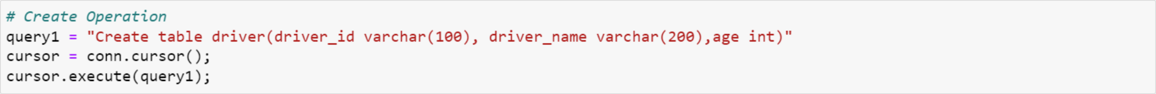
1. SQL database connection:



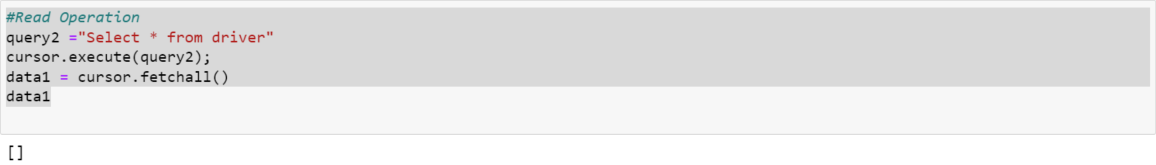
```
In [3]: import pyodbc
        pyodbc.drivers()

Out[3]: ['SQL Server',
         'PostgreSQL ANSI(x64)',
         'PostgreSQL Unicode(x64)',
         'SQL Server Native Client 11.0',
         'SQL Server Native Client RDA 11.0',
         'ODBC Driver 17 for SQL Server',
         'Microsoft Access Driver (*.mdb, *.accdb)',
         'Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)',
         'Microsoft Access Text Driver (*.txt, *.csv)']
```

```
In [17]: import pyodbc

         conn = pyodbc.connect('Driver={ODBC Driver 17 for SQL Server};'
                               'Server=LAPTOP-B09T49TN;'
                               'Database=AdventureWorksDW2014;'
                               'Trusted_Connection=yes;')
```

2. Create Operation:

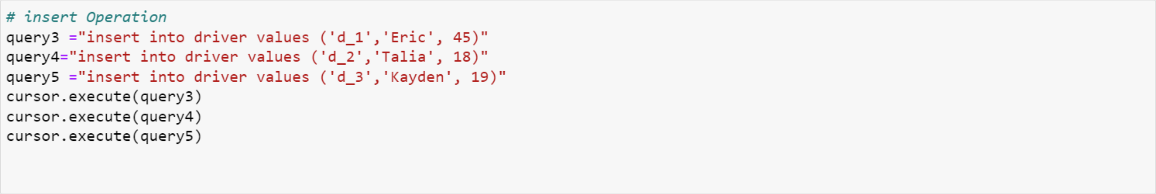


```
# Create Operation
query1 = "Create table driver(driver_id varchar(100), driver_name varchar(200),age int)"
cursor = conn.cursor();
cursor.execute(query1);
```



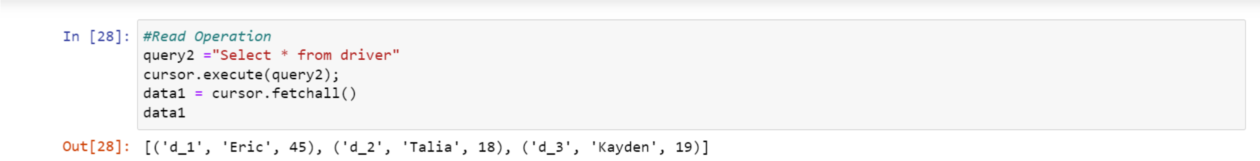
```
#Read Operation
query2 ="Select * from driver"
cursor.execute(query2);
data1 = cursor.fetchall()
data1
```

```
[]
```



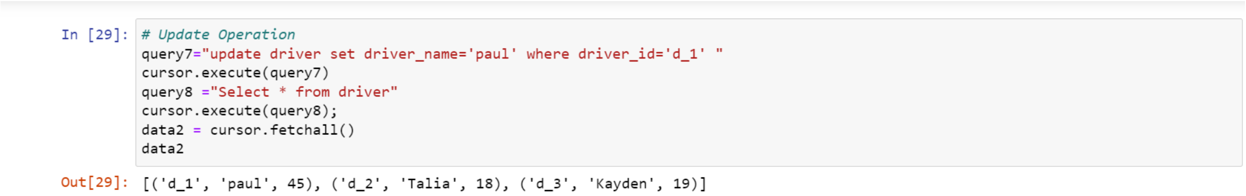
```
# insert Operation
query3 ="insert into driver values ('d_1','Eric', 45)"
query4="insert into driver values ('d_2','Talia', 18)"
query5 ="insert into driver values ('d_3','Kayden', 19)"
cursor.execute(query3)
cursor.execute(query4)
cursor.execute(query5)
```

3. Read Operation:



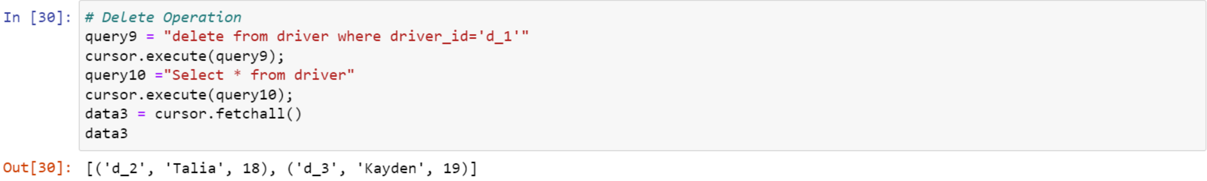
```
In [28]: #Read Operation
         query2 ="Select * from driver"
         cursor.execute(query2);
         data1 = cursor.fetchall()
         data1

Out[28]: [('d_1', 'Eric', 45), ('d_2', 'Talia', 18), ('d_3', 'Kayden', 19)]
```

4. Update Operation:



```
In [29]: # Update Operation
         query7="update driver set driver_name='paul' where driver_id='d_1' "
         cursor.execute(query7)
         query8 ="Select * from driver"
         cursor.execute(query8);
         data2 = cursor.fetchall()
         data2

Out[29]: [('d_1', 'paul', 45), ('d_2', 'Talia', 18), ('d_3', 'Kayden', 19)]
```

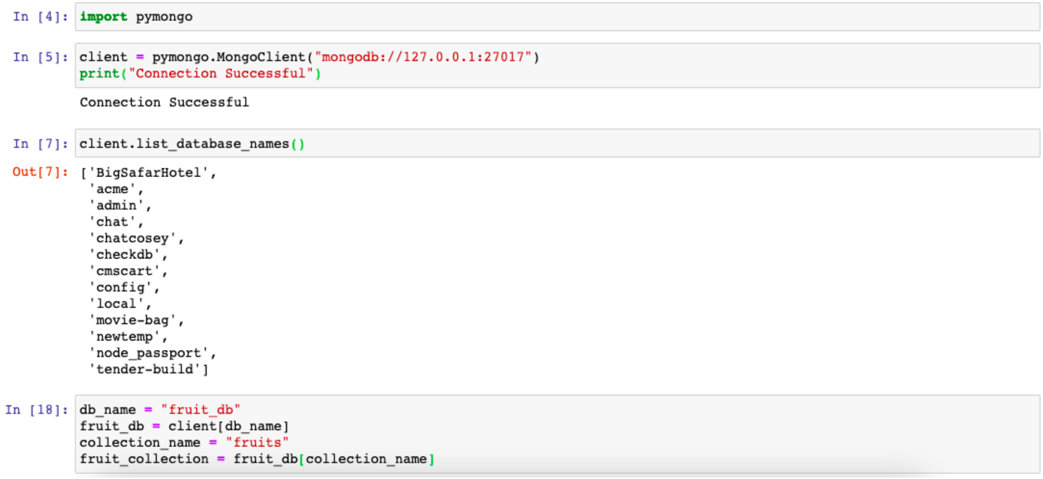
5. Delete Operation:



```
In [30]: # Delete Operation
         query9 = "delete from driver where driver_id='d_1'"
         cursor.execute(query9);
         query10 ="Select * from driver"
         cursor.execute(query10);
         data3 = cursor.fetchall()
         data3

Out[30]: [('d_2', 'Talia', 18), ('d_3', 'Kayden', 19)]
```

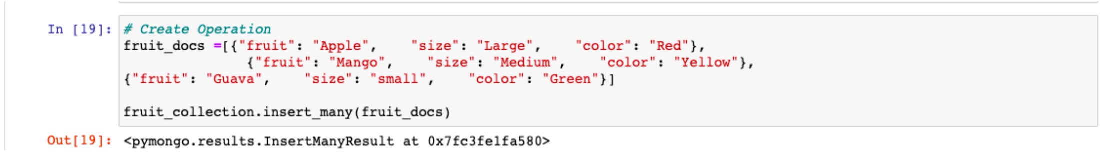
Apply CRUD operations on the list of documents(collection) using pymango

1. Mongo Database Connection



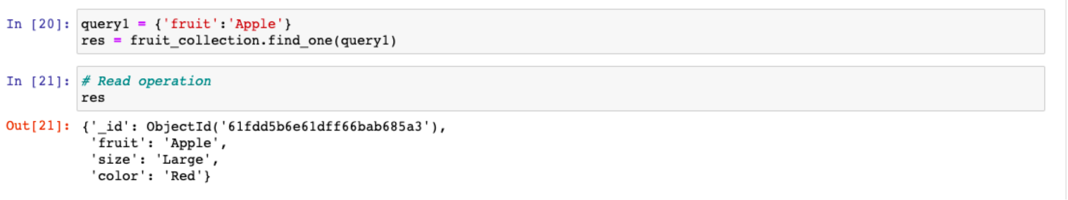
```
In [4]: import pymongo

In [5]: client = pymongo.MongoClient("mongodb://127.0.0.1:27017")
        print("Connection Successful")

        Connection Successful

In [7]: client.list_database_names()

Out[7]: ['BigSafarHotel',
         'acme',
         'admin',
         'chat',
         'chatcosey',
         'checkdb',
         'cmscart',
         'config',
         'local',
         'movie-bag',
         'newtemp',
         'node_passport',
         'tender-build']

In [18]: db_name = "fruit_db"
         fruit_db = client[db_name]
         collection_name = "fruits"
         fruit_collection = fruit_db[collection_name]
```

2. Create



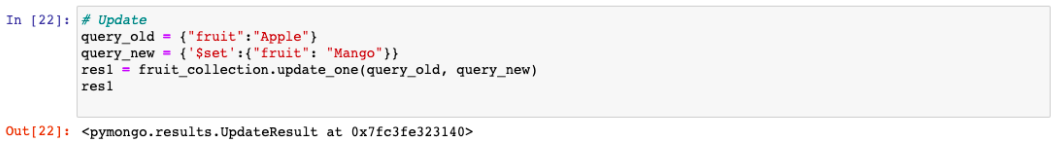
```
In [19]: # Create Operation
         fruit_docs =[{"fruit": "Apple",    "size": "Large",    "color": "Red"},
                        {"fruit": "Mango",    "size": "Medium",    "color": "Yellow"},
         {"fruit": "Guava",    "size": "small",    "color": "Green"}]

         fruit_collection.insert_many(fruit_docs)

Out[19]: <pymongo.results.InsertManyResult at 0x7fc3fe1fa580>
```

3. Read



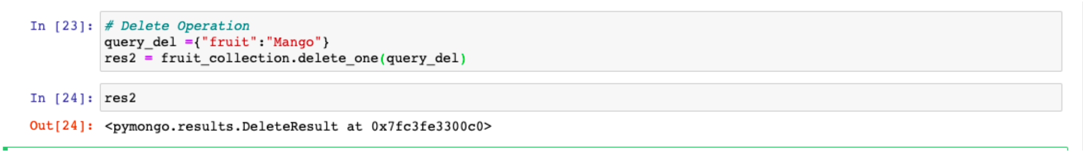
```
In [20]: query1 = {'fruit':'Apple'}
         res = fruit_collection.find_one(query1)

In [21]: # Read operation
         res

Out[21]: {'_id': ObjectId('61fdd5b6e61dff66bab685a3'),
          'fruit': 'Apple',
          'size': 'Large',
          'color': 'Red'}
```

4. Update

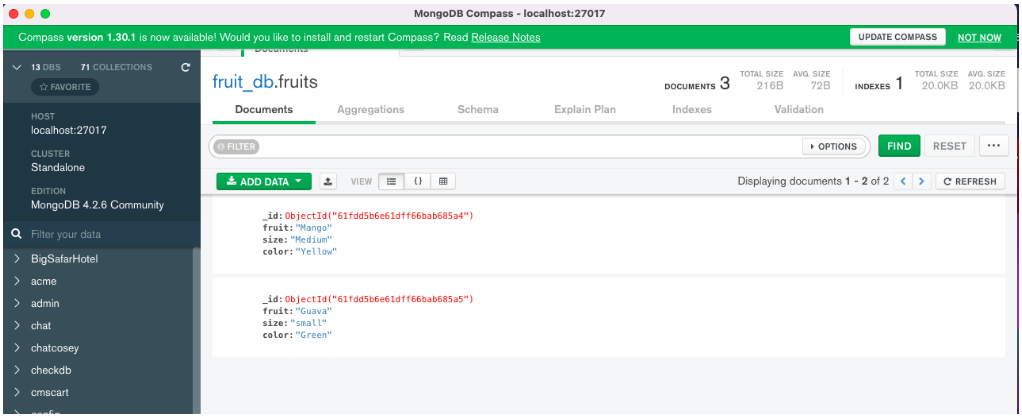


```
In [22]: # Update
         query_old = {"fruit":"Apple"}
         query_new = {'$set':{"fruit": "Mango"}}
         res1 = fruit_collection.update_one(query_old, query_new)
         res1

Out[22]: <pymongo.results.UpdateResult at 0x7fc3fe323140>
```

5. Delete



```
In [23]: # Delete Operation
         query_del ={"fruit":"Mango"}
         res2 = fruit_collection.delete_one(query_del)

In [24]: res2

Out[24]: <pymongo.results.DeleteResult at 0x7fc3fe3300c0>
```



MongoDB Compass - localhost:27017

Compass **version 1.30.1** is now available! Would you like to install and restart Compass? Read Release Notes

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 4.2.6 Community

DOCUMENTS 3 TOTAL SIZE 216B AVG. SIZE 72B INDEXES 1 TOTAL SIZE 20.0KB AVG. SIZE 20.0KB

Displaying documents 1 - 2 of 2

```
_id: ObjectId("61fdd5b6e61dff66bab685a4")
fruit: "Mango"
size: "Medium"
color: "Yellow"

_id: ObjectId("61fdd5b6e61dff66bab685a5")
fruit: "Guava"
size: "small"
color: "Green"
```