

ASSIGNMENT 6

```
import pymongo
from urllib.parse import quote_plus

username = "t23cs008"
password = "Aby@0210"
cluster_url = "cluster0.nnxad22.mongodb.net"

# Escape and quote the username and password
escaped_username = quote_plus(username)
escaped_password = quote_plus(password)

# Create the MongoDB connection string
connection_string = f"mongodb+srv://{escaped_username}:{escaped_password}@{cluster_url}/test?retryWrites=true&w=majority"

# Establish the connection
client = pymongo.MongoClient(connection_string)

# Access the 'test' database
db = client.test
```

[2] ✓ 1.7s

```
db=client['mongodb']
```

[3] ✓ 0.0s

Q1-Design a MongoDB schema for a "Student" collection with the following fields:

```
coll_create=db['Student']
```

[3] ✓ 0.0s

+ Code + Markdown Python

Q2- Insert the following student data in the collection

```
student_data=[
{"RollNum": 43, "FirstName": "John","LastName":"Doe","Age": 20,"Department":"Computer Science","Mark": 78},
{"RollNum": 67, "FirstName": "Alice","LastName":"Smith","Age": 22,"Department":"Physics","Mark": 59},
{"RollNum": 23, "FirstName": "Bob","LastName":"Johnson","Age": 21,"Department":"Computer Science","Mark": 81},
{"RollNum": 18, "FirstName": "Eve","LastName":"Adams","Age": 19,"Department":"Mathematics","Mark": 56},
{"RollNum": 84, "FirstName": "Mike","LastName":"Brown","Age": 23,"Department":"Physics","Mark": 92}
]
coll_create.insert_many(student_data)
```

[5] ✓ 0.6s Python

... InsertManyResult([ObjectId('65490afd1d0bc40daf734a9d'), ObjectId('65490afd1d0bc40daf734a9e'), ObjectId('65490afd1d0bc40daf734a9f'), ObjectId('65490afd1

```
for i in coll_create.find():
    print(i)
```

[6] ✓ 0.0s Python

... {'_id': ObjectId('65490afd1d0bc40daf734a9d'), 'RollNum': 43, 'FirstName': 'John', 'LastName': 'Doe', 'Age': 20, 'Department': 'Computer Science', 'Mark': 78},
{'_id': ObjectId('65490afd1d0bc40daf734a9e'), 'RollNum': 67, 'FirstName': 'Alice', 'LastName': 'Smith', 'Age': 22, 'Department': 'Physics', 'Mark': 59},
{'_id': ObjectId('65490afd1d0bc40daf734a9f'), 'RollNum': 23, 'FirstName': 'Bob', 'LastName': 'Johnson', 'Age': 21, 'Department': 'Computer Science', 'Mark': 81},
{'_id': ObjectId('65490afd1d0bc40daf734aa0'), 'RollNum': 18, 'FirstName': 'Eve', 'LastName': 'Adams', 'Age': 19, 'Department': 'Mathematics', 'Mark': 56},
{'_id': ObjectId('65490afd1d0bc40daf734aa1'), 'RollNum': 84, 'FirstName': 'Mike', 'LastName': 'Brown', 'Age': 23, 'Department': 'Physics', 'Mark': 92}

Q4-Write a MongoDB query to find all students in the "Computer Science" department.

```
for i in coll_create.find({'Department':'Computer Science'}):
    print(i)
```

[8] ✓ 0.0s Python

... {'_id': ObjectId('65490afd1d0bc40daf734a9d'), 'RollNum': 43, 'FirstName': 'John', 'LastName': 'Doe', 'Age': 20, 'Department': 'Computer Science', 'Mark': 78},
{'_id': ObjectId('65490afd1d0bc40daf734a9f'), 'RollNum': 23, 'FirstName': 'Bob', 'LastName': 'Johnson', 'Age': 21, 'Department': 'Computer Science', 'Mark': 81}

Q5-Write a MongoDB query to find all students whose age is greater than or equal to 20.

+ Code

+ Markdown

```
for i in coll_create.find({"Age": {"$gte": 20}}):  
    print(i)
```

Python

[9] ✓ 0.0s

```
... {'_id': ObjectId('65490afd1d0bc40daf734a9d'), 'RollNum': 43, 'FirstName': 'John', 'LastName': 'Doe', 'Age': 20, 'Department': 'Computer Science', 'Mark': 78}  
{'_id': ObjectId('65490afd1d0bc40daf734a9e'), 'RollNum': 67, 'FirstName': 'Alice', 'LastName': 'Smith', 'Age': 22, 'Department': 'Physics', 'Mark': 59}  
{'_id': ObjectId('65490afd1d0bc40daf734a9f'), 'RollNum': 23, 'FirstName': 'Bob', 'LastName': 'Johnson', 'Age': 21, 'Department': 'Computer Science', 'Mark': 81}  
{'_id': ObjectId('65490afd1d0bc40daf734aa1'), 'RollNum': 84, 'FirstName': 'Mike', 'LastName': 'Brown', 'Age': 23, 'Department': 'Physics', 'Mark': 92}
```

Q6-Write a MongoDB query to find all students whose mark is less than 60.

```
for i in coll_create.find({"Mark": {"$lt": 60}}):  
    print(i)
```

Python

[10] ✓ 0.0s

```
... {'_id': ObjectId('65490afd1d0bc40daf734a9e'), 'RollNum': 67, 'FirstName': 'Alice', 'LastName': 'Smith', 'Age': 22, 'Department': 'Physics', 'Mark': 59}  
{'_id': ObjectId('65490afd1d0bc40daf734aa0'), 'RollNum': 18, 'FirstName': 'Eve', 'LastName': 'Adams', 'Age': 19, 'Department': 'Mathematics', 'Mark': 56}
```

Q7-Write a MongoDB query to show the first name and Mark of all students in the "Physics" department.

```
for i in coll_create.find({"Department": "Physics"}, {"FirstName": 1, "Mark": 1}):  
    print(i)
```

Python

[15] ✓ 0.0s

```
... {'_id': ObjectId('65490afd1d0bc40daf734a9e'), 'FirstName': 'Alice', 'Mark': 59}  
{'_id': ObjectId('65490afd1d0bc40daf734aa1'), 'FirstName': 'Mike', 'Mark': 92}
```

Q8-Write a MongoDB query to find all students in the descending order of Mark.

```
for i in coll_create.find().sort({"Mark": -1}):  
    print(i)
```

Python

[18] ✓ 0.0s

```
... ObjectId('65490afd1d0bc40daf734aa1'), 'RollNum': 84, 'FirstName': 'Mike', 'LastName': 'Brown', 'Age': 23, 'Department': 'Physics', 'Mark': 92}  
ObjectId('65490afd1d0bc40daf734a9f'), 'RollNum': 23, 'FirstName': 'Bob', 'LastName': 'Johnson', 'Age': 21, 'Department': 'Computer Science', 'Mark': 81}  
ObjectId('65490afd1d0bc40daf734a9d'), 'RollNum': 43, 'FirstName': 'John', 'LastName': 'Doe', 'Age': 20, 'Department': 'Computer Science', 'Mark': 78}  
ObjectId('65490afd1d0bc40daf734a9e'), 'RollNum': 67, 'FirstName': 'Alice', 'LastName': 'Smith', 'Age': 22, 'Department': 'Physics', 'Mark': 59}  
ObjectId('65490afd1d0bc40daf734aa0'), 'RollNum': 18, 'FirstName': 'Eve', 'LastName': 'Adams', 'Age': 19, 'Department': 'Mathematics', 'Mark': 56}
```

Q9-Write a MongoDB query to find the youngest student.

```
for i in coll_create.find().sort({"Age": 1}).limit(1):  
    print(i)
```

Python

[21] ✓ 0.0s

```
... {'_id': ObjectId('65490afd1d0bc40daf734aa0'), 'RollNum': 18, 'FirstName': 'Eve', 'LastName': 'Adams', 'Age': 19, 'Department': 'Mathematics', 'Mark': 56}
```

Q10-Write a MongoDB query to find all students in the "Physics" department whose RollNum is greater than or equal to 70.

```
for i in coll_create.find({"Department": "Physics", "RollNum": {"$gte": 70}}):  
    print(i)
```

Python

[22] ✓ 0.0s

```
... {'_id': ObjectId('65490afd1d0bc40daf734aa1'), 'RollNum': 84, 'FirstName': 'Mike', 'LastName': 'Brown', 'Age': 23, 'Department': 'Physics', 'Mark': 92}
```

+ Code

+ Markdown