

# PROJECT PHASE 4

Serra Yakupoğlu - 31106  
Osman Berk An - 28849  
Yağmur Dolunay - 29323  
Ezgi Duman - 27900

At the beginning, the program says Welcome and we input a user id. The first, we started by creating an empty collection with selection 1 named Film collection :

The screenshot shows the MongoDB Atlas interface. On the left sidebar, under 'DEPLOYMENT', 'Database' is selected. A 'Create Database' button is visible. In the main area, 'AtlasCluster' is selected under 'Data Services'. Under 'Collections', 'Films' is selected. A sub-section titled 'AtlasCluster.Films' shows storage details: STORAGE SIZE: 4KB, LOCAL DATA SIZE: 0B, TOTAL DOCUMENTS: 0, INDEXES TOTAL SIZE: 4KB. It includes tabs for 'Find', 'Indexes', 'Schema Anti-Patterns', 'Aggregation', and 'Search Indexes'. A 'Filter' field contains the query '{ field: 'value' }'. Below it, 'QUERY RESULTS: 0' is displayed.

Then, we insert 3 different films by inputting manually and with the `insert_into_collection` method in Python code with selection 4 and can see the films at Mongo:

The screenshot shows the same MongoDB Atlas interface after inserting three documents. The 'Films' collection now has 3 documents. The 'QUERY RESULTS: 1-3 OF 3' section displays the following data:

```
id: ObjectId('65941daa989ab93fc033e0b#')
Title: "The Dark Knight"
Releasedate: "2008-07-18"
Genre: "Sci-Fi"
Runtime: "150"

id: ObjectId('65941daa989ab93fc033e0b#')
Title: "The Dark Knight"
Releasedate: "2008-07-18"
Genre: "Action"
Runtime: "150"
```

Then we read all the data with the `read_all` method with selection 2 and we can see the films also as an output in our Python code :

A screenshot of a Mac OS X desktop environment. The top bar shows the system status with icons for battery, signal, and network. The date and time are displayed as '5 Ocak Cum 10:08'. The main window is a terminal application with a dark theme. The title bar says 'Welcome' and the tab is 'phase4.py'. The terminal content is a Python script named 'phase4.py' which handles user input for selecting a collection ('Films' or 'CrewMembers') and reading data from it. The script uses MongoDB's PyMongo library. Below the terminal is a code editor window showing the same 'phase4.py' file. The code editor has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. A sidebar on the right lists recent files like 'zsh', 'Python: p...', and 'Python'. The bottom of the screen shows the Mac OS X dock with various application icons.

```
Users > osmabrkan > phase4.py > ...
244
245     selected_option = input("Selected option: ")
246
247     if selected_option == '1':
248         collection_name = input("Enter collection name: ")
249         review_collection = collection_name
250         createCollection(db, collection_name)
251
252     elif selected_option == '2':
253         if review_collection is None:
254             print("Please create or select a collection first.")
255         else:
256             print("Please select the collection to read:")
257             print("1-Films")
258             print("2-CrewMembers")
259
260             read_option = input("Selected option:")
261
262             if(read_option == '1'):
263                 review_collection = "Films"
264                 read_all_data(db,review_collection)
265             elif(read_option == '2'):
266                 review_collection = "CrewMembers"
267                 read_all_data(db,review.collection)
268
269     elif selected_option == '3':
270         if review_collection is None:
271             print("Please select the collection to read:
272             1-Films
273             2-CrewMembers
274 Selected option: 2
275 Please select the collection to read:
276             1-Films
277             2-CrewMembers
278 Selected option: 1
279
280             {'_id': ObjectId('55881aef0beaa68f210aa3e4')}, 'Title': 'Inception', 'ReleaseDate': '2010-07-16', 'Genre': 'Sci-Fi', 'Runtime': '158'}
281             {'_id': ObjectId('55881af0beaa68f210aa3e5')}, 'Title': 'The Dark Knight', 'ReleaseDate': '2008-07-18', 'Genre': 'Action', 'Runtime': '150'}
282             {'_id': ObjectId('55981b14beaa68f210aa3e6')}, 'Title': 'The Matrix', 'ReleaseDate': '1999-03-31', 'Genre': 'Sci-Fi', 'Runtime': '150'}
283
284 Please pick the option that you want to proceed:
285             1- Create a collection.
```

Then, we made a filtered read with selection 3 and we filtered it with a Runtime. The user gives an input Runtime and the program gives the film output with the selected Runtime.

A screenshot of a Mac OS X desktop environment. The top bar shows the system status with icons for battery, signal, and network. The title bar of the active window reads "phase4.py x". The main area contains a code editor with Python code for a movie database application. Below the code editor is a terminal window showing the execution of the script and its output. At the bottom, the Dock displays various application icons, and the status bar at the bottom right shows the date and time.

```
phase4.py x
Users > phase4.py ...
244     selected_option = input("Selected option: ")
245
246     if selected_option == '1':
247         collection_name = input("Enter collection name: ")
248         review_collection = collection_name
249         createCollection(db, collection_name)
250
251     elif selected_option == '2':
252         if review_collection is None:
253             print("Please create or select a collection first.")
254         else:
255             print("Please select the collection to read:")
256             print("1-Films")
257             print("2-CrewMembers")
258
259             read_option = input("Selected option:")
260
261             if(read_option == '1'):
262                 review_collection = "Films"
263                 read_all_data(db, review_collection)
264             elif(read_option == '2'):
265                 review_collection = "CrewMembers"
266                 read_all_data(db, review_collection)
267
268     elif selected_option == '3':
269         if review_collection is None:
270             print("Please select the filtering option:
271             1- Find by Runtime
272             2- Filter by CrewMember Role
273             Selected option: 1
274             Enter Runtime: 150
275
276             {'_id': ObjectId('65981ae7bead6bf210aa3e4'), 'Title': 'Inception', 'ReleaseDate': '2010-07-16', 'Genre': 'Sci-Fi', 'Runtime': '150'}
277             {'_id': ObjectId('65981ae7bead6bf210aa3e5'), 'Title': 'The Dark Knight', 'ReleaseDate': '2008-07-18', 'Genre': 'Action', 'Runtime': '150'}
278             {'_id': ObjectId('65981b14eaad6bf210aa3e6'), 'Title': 'The Matrix', 'ReleaseDate': '1999-03-31', 'Genre': 'Sci-Fi', 'Runtime': '150'}
279
280             Please pick the option that you want to proceed:
281             1- Create a collection.
282             2- Read all data in a collection.
```

Then, there is a deletion(Selection5). The user inputs an ObjectId a film contains that id and that film is deleted from the collection, here, we inputted the first film's object id and that film("Inception") was deleted:

```

_id: ObjectId('65981afbbeaa668f210aa3e5')
Title: "The Dark Knight"
ReleaseDate: "2008-07-18"
Genre: "Action"
Runtime: "150"

_id: ObjectId('65981b14bbeaa668f210aa3e6')
Title: "The Matrix"
ReleaseDate: "1999-03-31"
Genre: "Sci-Fi"
Runtime: "130"

```

Lastly, the last part is the update part(Selection6). User again input an ObjectId that he/she wants to update that film in collection. Then again user input a new runtime for that related film and that films runtime is updated, Here we put new runtime = 111 for The Dark Knight film runtime is updated:

```

_id: ObjectId('65981afbbeaa668f210aa3e5')
Title: "The Dark Knight"
ReleaseDate: "2008-07-18"
Genre: "Action"
Runtime: "111"

_id: ObjectId('65981b14bbeaa668f210aa3e6')
Title: "The Matrix"
ReleaseDate: "1999-03-31"
Genre: "Sci-Fi"
Runtime: "130"

```

Then we created a second empty collection named CrewMembers(Selection1):

The screenshot shows the MongoDB Atlas interface. On the left sidebar, under the 'Database' section, 'AtlasCluster' is expanded, and 'CrewMembers' is selected. The main panel displays the 'AtlasCluster.CrewMembers' collection. It shows storage details: STORAGE SIZE: 4KB, LOGICAL DATA SIZE: 0B, TOTAL DOCUMENTS: 0, and INDEXES TOTAL SIZE: 4KB. Below these details is a search bar with the placeholder 'Type a query: { field: 'value' }'. At the bottom of the main panel, it says 'QUERY RESULTS: 0'.

Then, we inserted 4 crew members to the CrewMember collection with the related data fields (Selection 4):

The screenshot shows the same MongoDB Atlas interface as before, but now with 4 documents in the 'AtlasCluster.CrewMembers' collection. The 'QUERY RESULTS' section at the bottom of the main panel shows '1-4 OF 4'. The first document is:

```
_id: ObjectId("65981f5d8caad68f210aa3e7")
Crew_Member_Name: "Osman Berk An"
Crew_Member_Role: "Director"
Film_Id: "1"
```

The second document is:

```
_id: ObjectId("65981f5f8caad68f210aa3e8")
Crew_Member_Name: "Yagmur Dolunay"
Crew_Member_Role: "Editor"
Film_Id: "2"
```

The third document is:

```
_id: ObjectId("65981ff08caad68f210aa3e9")
Crew_Member_Name: "Hakan Duman"
Crew_Member_Role: "Editor"
Film_Id: "3"
```

Then, we make a read all(Selection 2) and we see the all crew members in the collection as a output at the python code also we can see at the mongodb:

```

Code File Edit Selection View Go Run Terminal Window Help
Welcome phase4.py ...
Users > osmanbrkan > phase4.py ...
    # Insert data.
    # Delete data.
    # Update data.
    Selected option: 2
Please select the collection to read:
1-Films
2-CrewMembers
Selected option: 2
{
    "_id": ObjectId('65981f4beaad68f210aa3e2'), "Crew_Member_Name": "Osman Berk An", "Crew_Member_Role": "Director", "Film_id": '1'
    {"_id": ObjectId('65981f54beaad68f210aa3e1'), "Crew_Member_Name": "Yagmur Dolunay", "Crew_Member_Role": "Editor", "Film_id": '2'}
    {"_id": ObjectId('65981f57beaad68f210aa3e3'), "Crew_Member_Name": "Ezgi Duman", "Crew_Member_Role": "Editor", "Film_id": '3'}
    {"_id": ObjectId('65981f58beaad68f210aa3e4'), "Crew_Member_Name": "Serra Yakupoglu", "Crew_Member_Role": "Costume Designer", "Film_id": '4'}
}
Please pick the option that you want to proceed:
1- Create a collection.
2- Read all data in a collection.
2- Read all data in a collection.
Ln 10, Col 96 Spaces: 4 UTF-8 LF ↵ Python 3.11.3 64-bit

```

Then, we make a filtered read(Selection3). Filter is a crew member role. User inputs an crew member role and we can see the crew members with that related role as an output, For example here we input role as Editor and we see the crew members their role is Editor:

```

Code File Edit Selection View Go Run Terminal Window Help
Welcome phase4.py ...
Users > osmanbrkan > phase4.py ...
    # Insert data.
    # Delete data.
    # Update data.
    Selected option: 3
Please select the filtering option:
1- Find by Crew Member Name
2- Find by Crew Member Role
Selected option: 2
Enter Crew Member Role: Editor
{
    {"_id": ObjectId('65981f54beaad68f210aa3e2'), "Crew_Member_Name": "Yagmur Dolunay", "Crew_Member_Role": "Editor", "Film_id": '2'}
    {"_id": ObjectId('65981f57beaad68f210aa3e3'), "Crew_Member_Name": "Ezgi Duman", "Crew_Member_Role": "Editor", "Film_id": '3'}
}
Please pick the option that you want to proceed:
1- Create a collection.
2- Read all data in a collection.
3- Read some part of the data while filtering.
4- Insert data.
5- Delete data.
6- Update data.
Selected option: 3
Please select the filtering option:
1- Find by Crew Member Name
2- Find by Crew Member Role
Selected option: 2
Enter Crew Member Role: Editor
{
    {"_id": ObjectId('65981f54beaad68f210aa3e2'), "Crew_Member_Name": "Yagmur Dolunay", "Crew_Member_Role": "Editor", "Film_id": '2'}
    {"_id": ObjectId('65981f57beaad68f210aa3e3'), "Crew_Member_Name": "Ezgi Duman", "Crew_Member_Role": "Editor", "Film_id": '3'}
}
Please pick the option that you want to proceed:
1- Create a collection.
2- Read all data in a collection.
3- Read some part of the data while filtering.
4- Insert data.
5- Delete data.
6- Update data.
Selected option: 3
Please select the filtering option:
1- Find by Crew Member Name
2- Find by Crew Member Role
Selected option: 2
Enter Crew Member Role: Editor
{
    {"_id": ObjectId('65981f54beaad68f210aa3e2'), "Crew_Member_Name": "Yagmur Dolunay", "Crew_Member_Role": "Editor", "Film_id": '2'}
    {"_id": ObjectId('65981f57beaad68f210aa3e3'), "Crew_Member_Name": "Ezgi Duman", "Crew_Member_Role": "Editor", "Film_id": '3'}
}
Please pick the option that you want to proceed:
Ln 10, Col 96 Spaces: 4 UTF-8 LF ↵ Python 3.11.3 64-bit

```

Then, we made a deletion(Selection5). We again input ObjectId and the crew member with that id is deleted, now we have 3 crew members:

The screenshot shows the MongoDB Atlas Cluster interface. On the left, the navigation sidebar includes Project 0, Data Services (selected), App Services, Charts, Overview, Deployment (AtlasCluster selected), Database (Films), Services, Security, and Advanced. The system status is shown as "All Good". The main area displays the "AtlasCluster.CrewMembers" collection. It shows a storage size of 36KB, logical data size of 307B, total documents of 3, and index sizes of 36KB. A search bar at the top says "Search Namespaces". Below it, there are tabs for Find, Indexes, Schema Anti-Patterns, Aggregation, and Search Indexes. An "INSERT DOCUMENT" button is at the top right. A "Filter" field contains the query: { field: 'value' }. Three documents are listed:

```

_id: ObjectId('65981f4abaea668f210aa3e7')
Crew_Member_Name: "Osman Berk An"
Crew_Member_Role: "Director"
Film_Id: "1"

_id: ObjectId('65981f5fbbeaa668f210aa3e9')
Crew_Member_Name: "Ezgi Duman"
Crew_Member_Role: "Editor"
Film_Id: "2"

_id: ObjectId('65981f6dbeaa668f210aa3ea')
Crew_Member_Name: "Serra Yakupoglu"
Crew_Member_Role: "Costume Designer"

```

Lastly, we made an update(Selection6). User gives input Object id. And also the user gives a new role input. Crew members with related Object\_id are updated. Its role is updated. Here Crew\_Member\_Role of Serra is updated to "Barista":

This screenshot is identical to the previous one, showing the MongoDB Atlas Cluster interface. The navigation sidebar and collection details are the same. The main area shows the "AtlasCluster.CrewMembers" collection with three documents. The third document, which had a role of "Costume Designer", has been updated to "Barista".

```

_id: ObjectId('65981f5fbbeaa668f210aa3e9')
Crew_Member_Name: "Ezgi Duman"
Crew_Member_Role: "Editor"
Film_Id: "2"

_id: ObjectId('65981f6dbeaa668f210aa3ea')
Crew_Member_Name: "Serra Yakupoglu"
Crew_Member_Role: "Barista"
Film_Id: "4"

```