MAPÚA UNIVERSITY
SCHOOL OF ELECTRICAL, ELECTRONICS, AND COMPUTER ENGINEERING

Experiment 6:

NoSQL Database Models

CPE106L (Software Design Laboratory)

Member 1: ASIDAO, Shane Eowyn R.
Member 2: GONZALES, Benedick B.
Member 3: LULU, Neal Axel M.
Member 4: VARGAS, Imar Colt S.

Group No.: **4**
Section: **E04**

PreLab

Readings, Insights, and Reflection

- [Chapter 9, Section 9.1 to 9.6] Lingras, P. (2016-01-01). Building Cross-Platform Mobile and Web Apps for Engineers and Scientists: An Active Learning Approach. [[VitalSource Bookshelf version]]. Retrieved from vbk://9781305855892
- https://www.mongodb.com/docs/manual/introduction/
- https://www.sqlitetutorial.net/wp-content/uploads/2018/03/sqlite-sample-database-diagram-color.pdf

Reflecting on the materials from *Building Cross-Platform Mobile and Web Apps for Engineers and Scientists: An Active Learning Approach* by P. Lingras (2016), along with the documentation from MongoDB and the SQLite sample database diagram, I've gained valuable insights into mobile and web application development, especially from an engineering and scientific standpoint.

Cross-platform development has become an essential aspect of modern engineering and scientific applications. Lingras highlights the importance of ensuring that web and mobile applications function seamlessly across various operating systems. As mobile technology continues to advance, I recognize the necessity of engaging with cross-platform solutions that enhance accessibility and usability. The book's active learning approach resonated with me—it reinforced how hands-on experience with coding and application development is key to mastering modern computational tools.

One of the biggest takeaways for me was the role of database management in application development. Comparing SQL and NoSQL databases helped me understand their unique strengths and limitations. MongoDB, as a NoSQL database, stands out for its flexibility and scalability, making it perfect for handling unstructured or semi-structured data. Its dynamic schema is a game-changer for applications requiring real-time updates and large-scale data management. On the other hand, SQLite

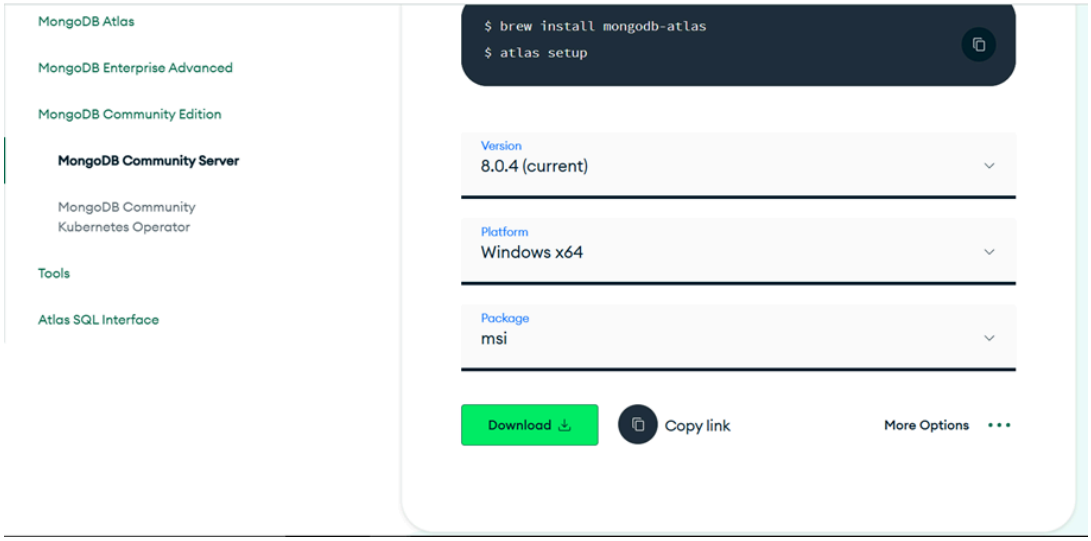
follows a structured, relational database model that is lightweight and ideal for mobile applications needing offline functionality. Looking at the SQLite sample database diagram, I could see how efficiently it organizes structured data with minimal setup—making it a solid choice for applications prioritizing local storage.

Understanding both NoSQL and SQL databases has been eye-opening for me as I think about mobile and web application development. Choosing between MongoDB and SQLite really depends on the specific needs of an application—whether it requires a scalable, flexible solution or a lightweight, structured database. Additionally, cross-platform development demands careful attention to database compatibility, performance, security, and data synchronization across different environments.

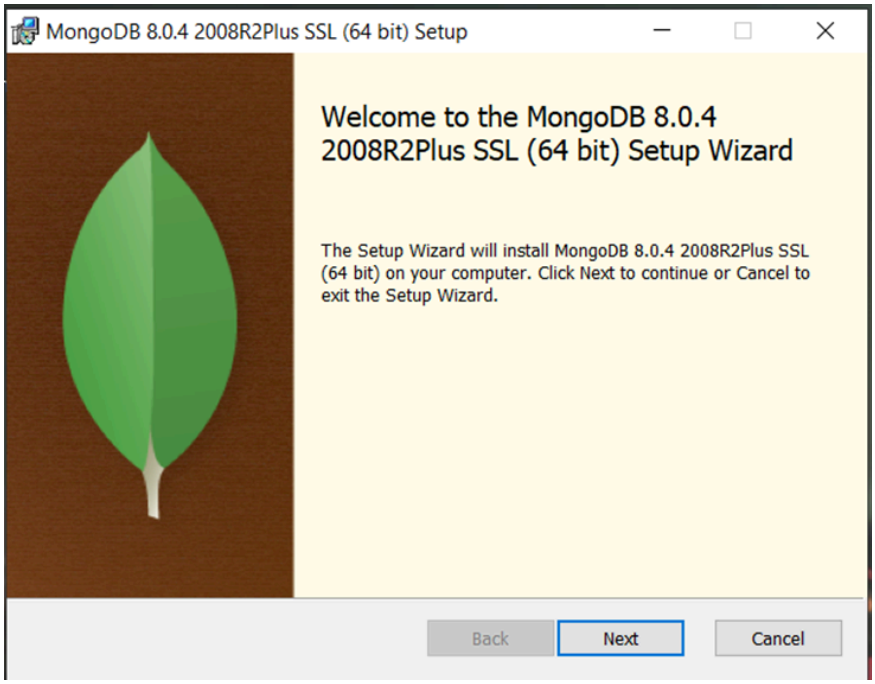
Ultimately, integrating mobile and web applications into engineering and scientific work significantly enhances data accessibility and computational efficiency. Lingras' emphasis on active learning, along with a deep dive into database structures, has given me a strong foundation for developing robust applications. By weighing the advantages of both MongoDB and SQLite, I feel more equipped to make informed decisions about database implementation, ensuring efficiency and adaptability in real-world scenarios

InLab

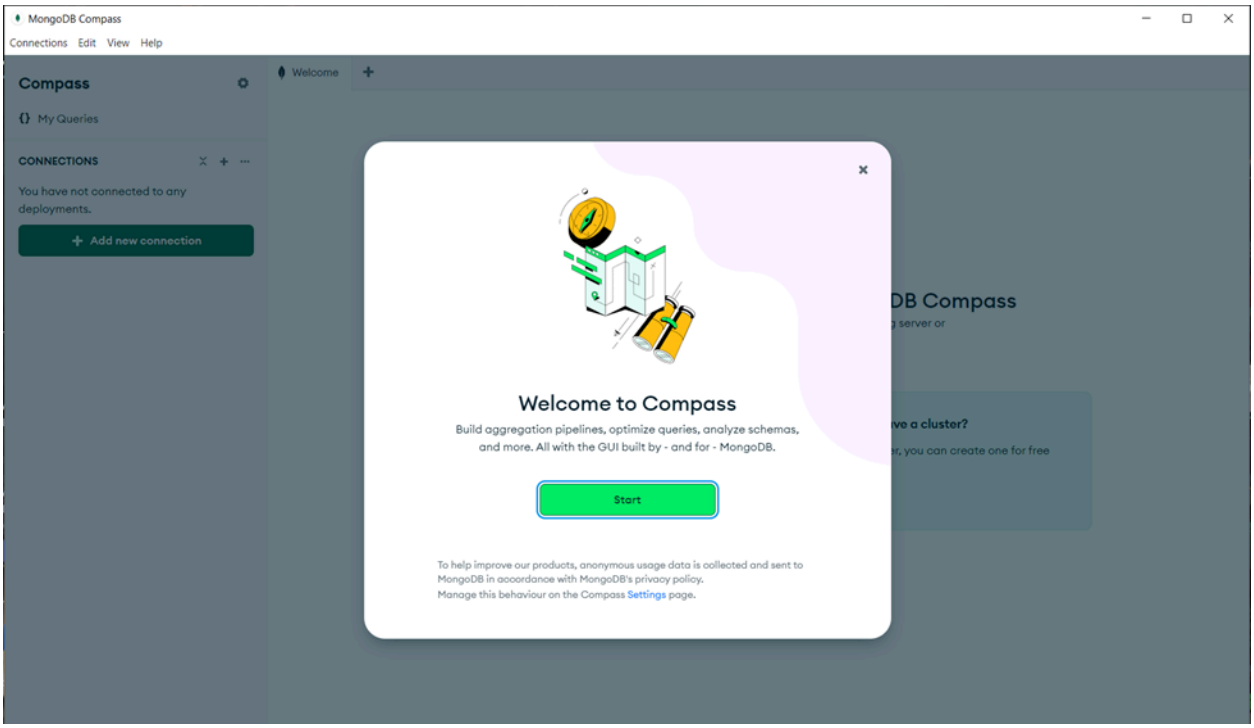
- Install MongoDB Compass. Download here: MongoDB Compass | MongoDB. For Ubuntu 64 bit OS, Download here >> https://downloads.mongodb.com/compass/mongodb-compass_1.41.0_amd64.deb Installation:

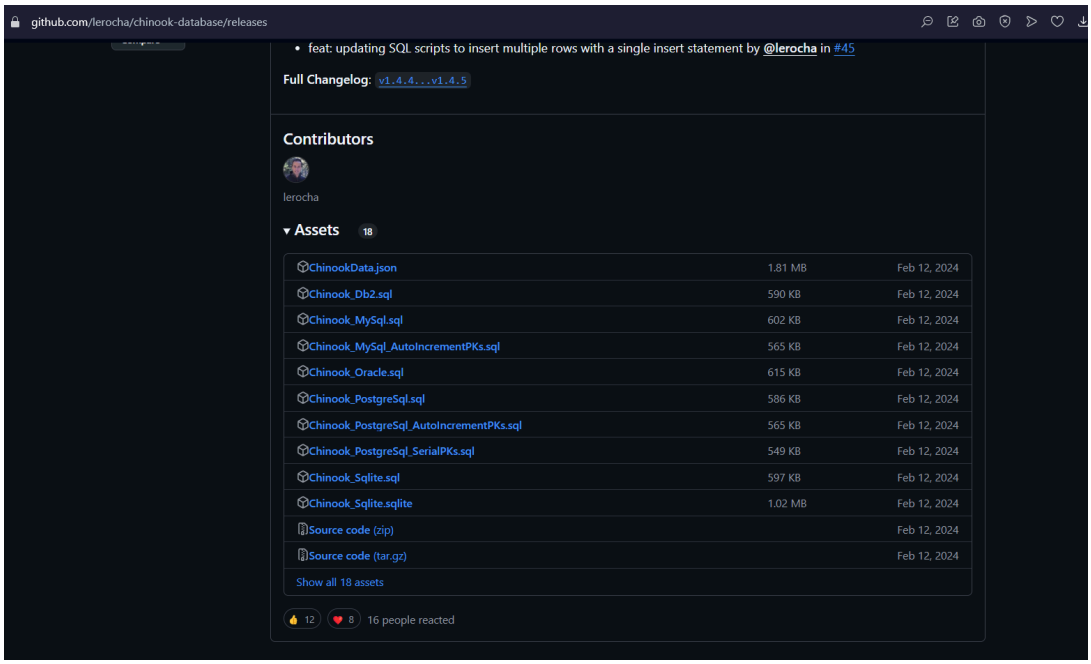


We downloaded MongoDB first. Once MongoDB had been downloaded, we installed it.

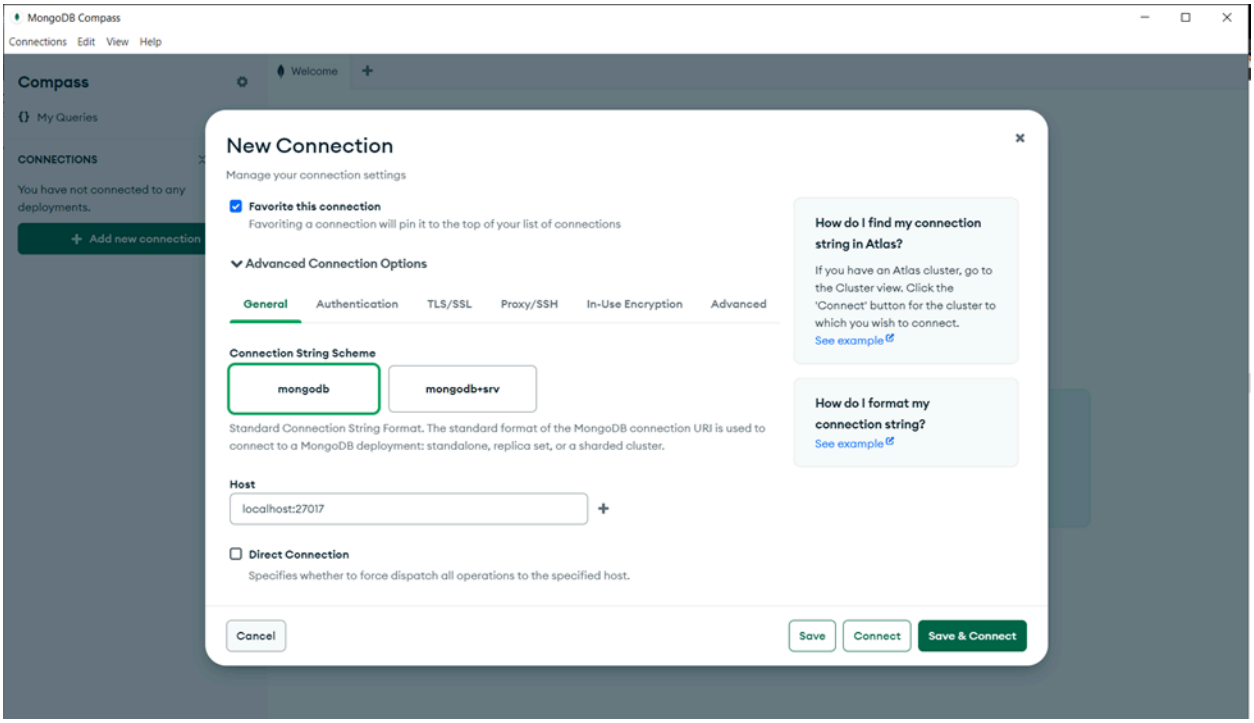


To verify the configuration, we connected to MongoDB, installed the appropriate Python version, and imported *chinook.db* into MongoDB Compass using SQLite.

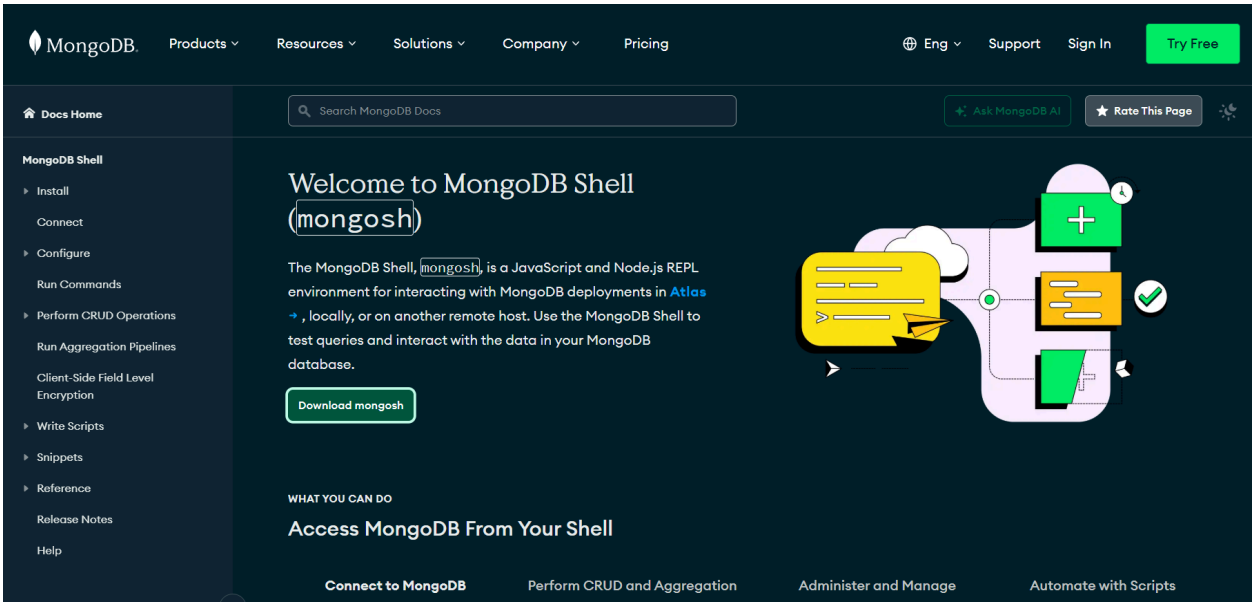




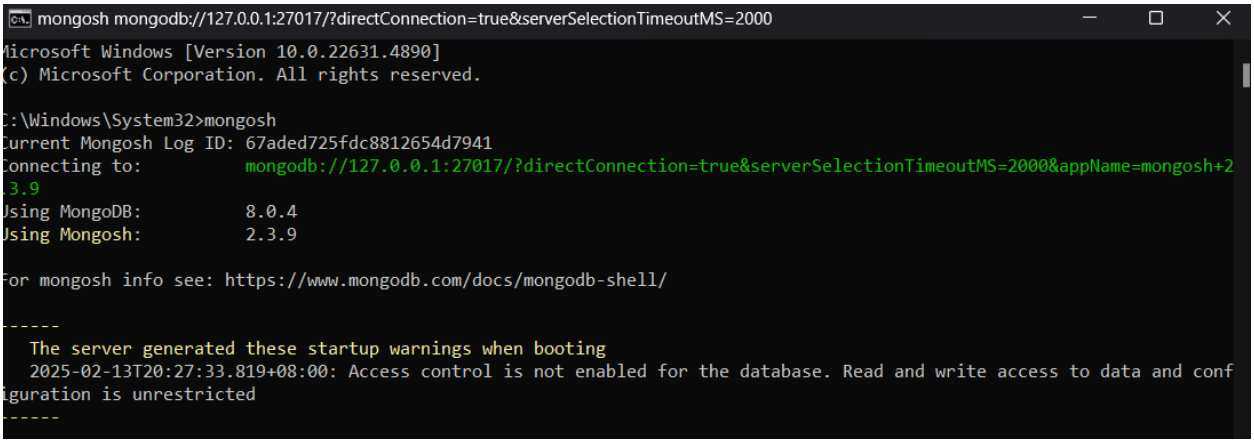
Install Chinook_Sqlite.sqlite



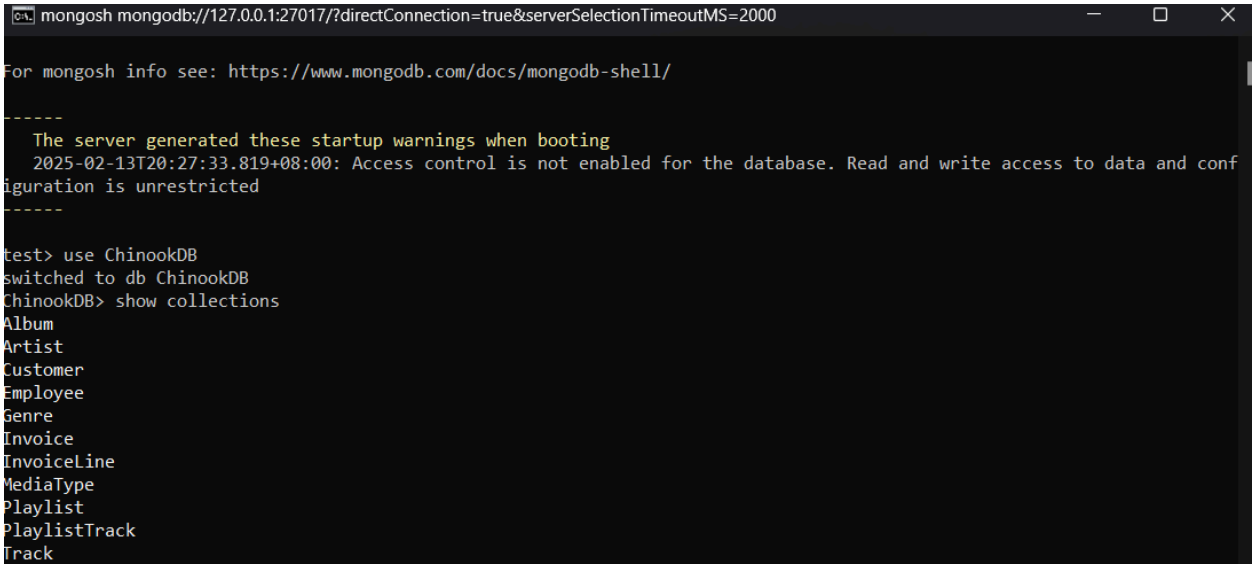
Creating a database connection



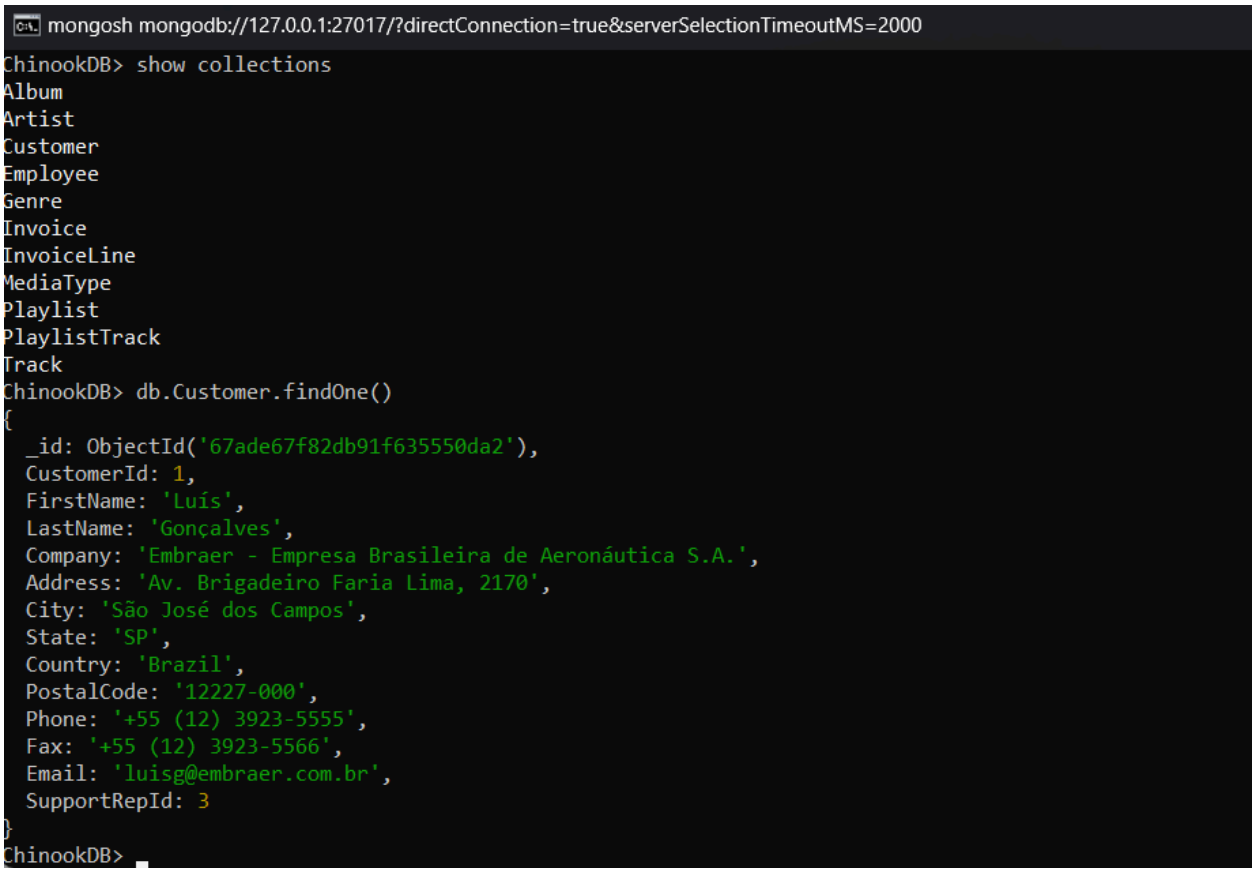
Install mongosh and extract the zip file to your desired directory



Check if MongoDB and Mongosh are installed



Check the tables inside of the chinook database



pymongo_test.py ● mongo_test2.py ×

D: > LR5 > chinookdb > mongo_test2.py > ...

```python
import sqlite3
from pymongo import MongoClient

# Connect to SQLite database
sqlite_conn = sqlite3.connect("C:\\chinookdb\\chinook.db")  # Update path if needed
sqlite_cursor = sqlite_conn.cursor()

# Connect to MongoDB
mongo_client = MongoClient("mongodb://localhost:27017/")
mongo_db = mongo_client["ChinookDB"]

# Get all table names from SQLite
sqlite_cursor.execute("SELECT name FROM sqlite_master WHERE type='table';")
tables = [table[0] for table in sqlite_cursor.fetchall()]

for table in tables:
    print(f"Importing {table}...")

    # Fetch all rows from the table
    sqlite_cursor.execute(f"SELECT * FROM {table}")
    rows = sqlite_cursor.fetchall()

    # Get column names
    column_names = [desc[0] for desc in sqlite_cursor.description]

    # Convert rows to dictionaries
    documents = [dict(zip(column_names, row)) for row in rows]

    # Insert into MongoDB
    if documents:
        mongo_db[table].insert_many(documents)
        print(f"Inserted {len(documents)} records into {table}")

# Close connections
sqlite_conn.close()
mongo_client.close()
```

Importing chinook sqlite database file using VS Code

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  QUERY RESULTS (PREVIEW)  SQL CONSOLE

Inserted 8 records into Employee
Importing Genre...
Inserted 25 records into Genre
Importing Invoice...
Inserted 412 records into Invoice
Importing InvoiceLine...
Inserted 2240 records into InvoiceLine
Importing MediaType...
Inserted 5 records into MediaType
Importing Playlist...
Inserted 18 records into Playlist
Importing PlaylistTrack...
Inserted 8715 records into PlaylistTrack
Importing Track...
Inserted 3503 records into Track
Data import completed successfully!
PS C:\Users\itski>
0 ⚠ 0   Connect
```

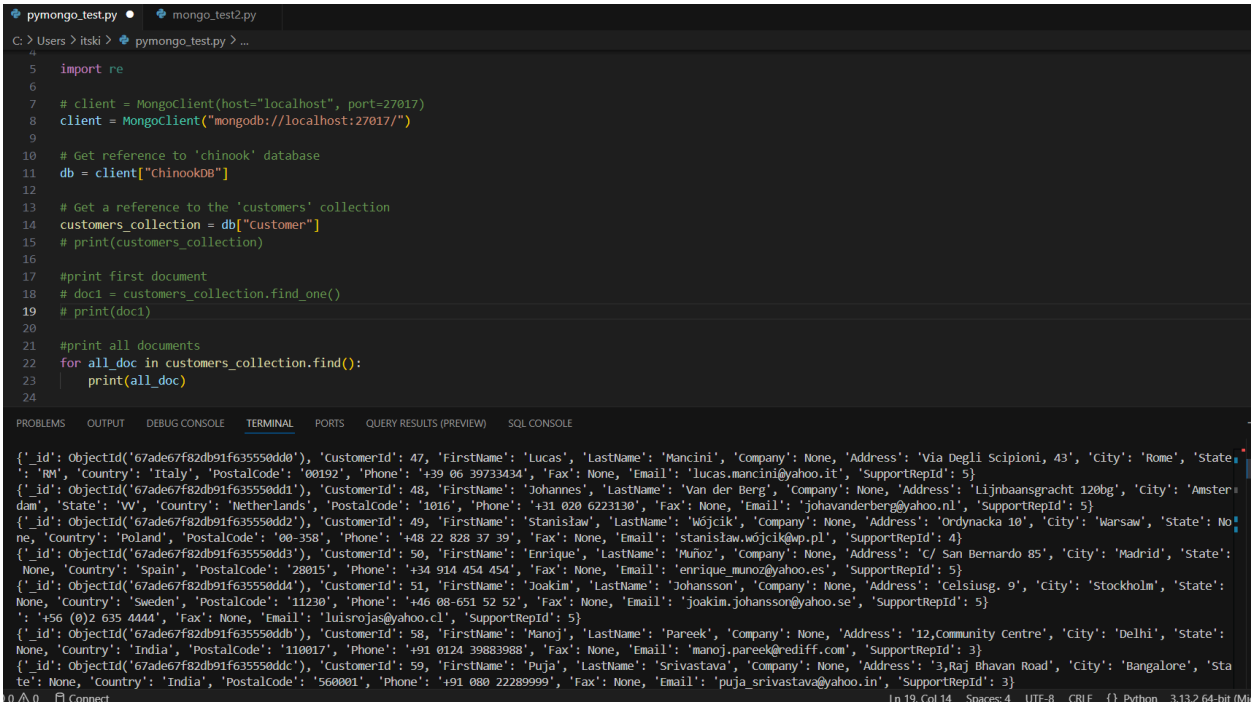
pymongo_test.py | mongo_test2.py

C: > Users > itski > pymongo_test.py > ...

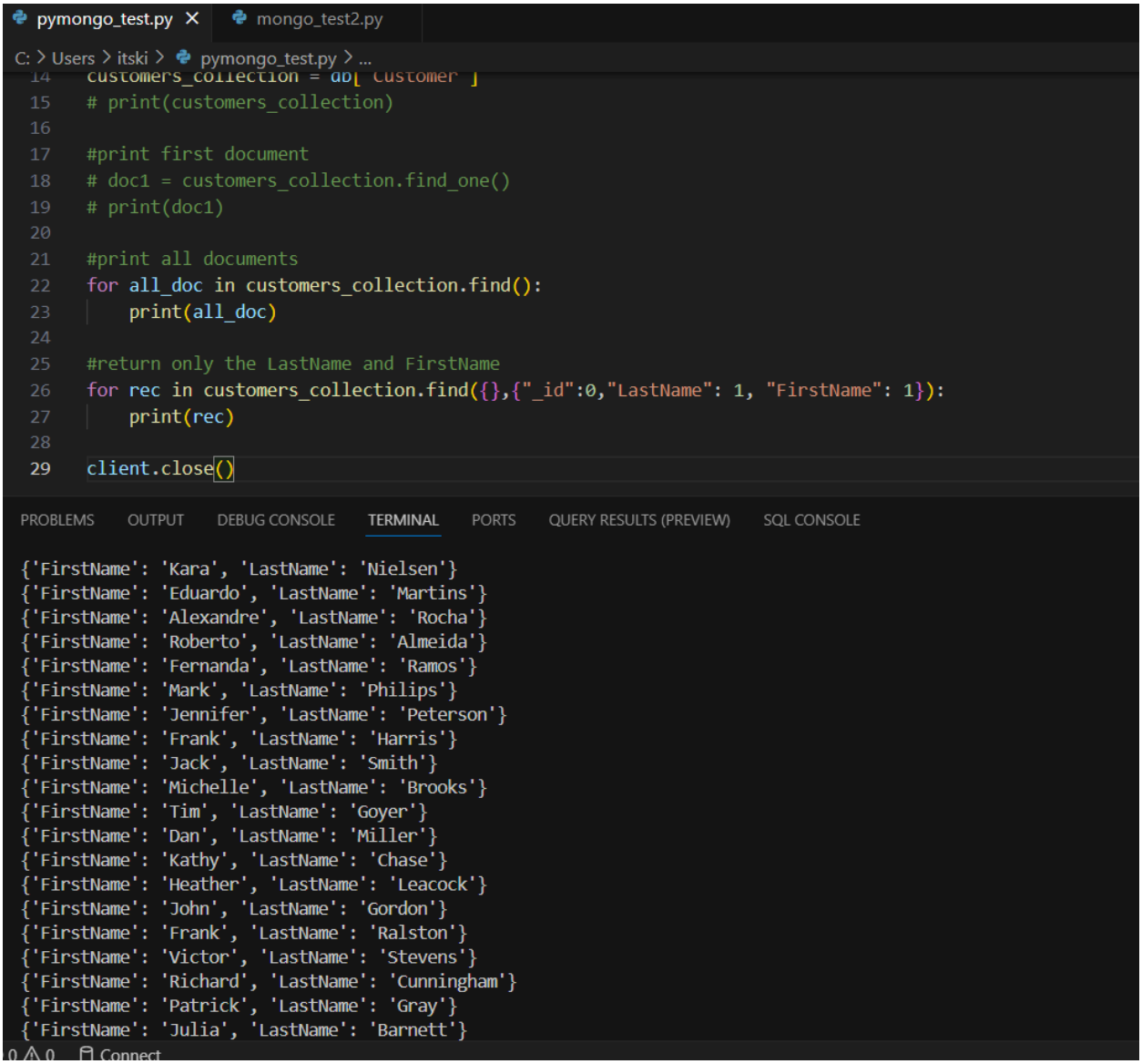
```python
from pymongo import MongoClient

import pprint

import re

# client = MongoClient(host="localhost", port=27017)
client = MongoClient("mongodb://localhost:27017/")

# Get reference to 'chinook' database
db = client["ChinookDB"]

# Get a reference to the 'customers' collection
customers_collection = db["Customer"]
# print(customers_collection)

#print all documents
doc1 = customers_collection.find_one()
print(doc1)

client.close()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  QUERY RESULTS (PREVIEW)  SQL CONSOLE

SoftwareFoundation.Python.3.13_qbz5n2kfra8p0/python.exe c:/Users/itski/pymongo_test.py
None
PS C:\Users\itski> & C:/Users/itski/AppData/Local/Microsoft/WindowsApps/PythonSoftwareFoundation.Python.3.13_qbz5n2kfra8p0/python.exe c:/Users/itski/pymongo_test.py
{'_id': ObjectId('67ade67f82db91f635550da2'), 'CustomerId': 1, 'FirstName': 'Luís', 'LastName': 'Gonçalves', 'Company': 'Embraer - Empresa Brasileira de Aeronáutica S.A.', 'Address': 'Av. Brigadeiro Faria Lima, 2170', 'City': 'São José dos Campos', 'State': 'SP', 'Country': 'Brazil', 'PostalCode': '12227-000', 'Phone': '+55 (12) 3923-5555', 'Fax': '+55 (12) 3923-5566', 'Email': 'luisg@embraer.com.br', 'SupportRepId': 3}
PS C:\Users\itski>
```

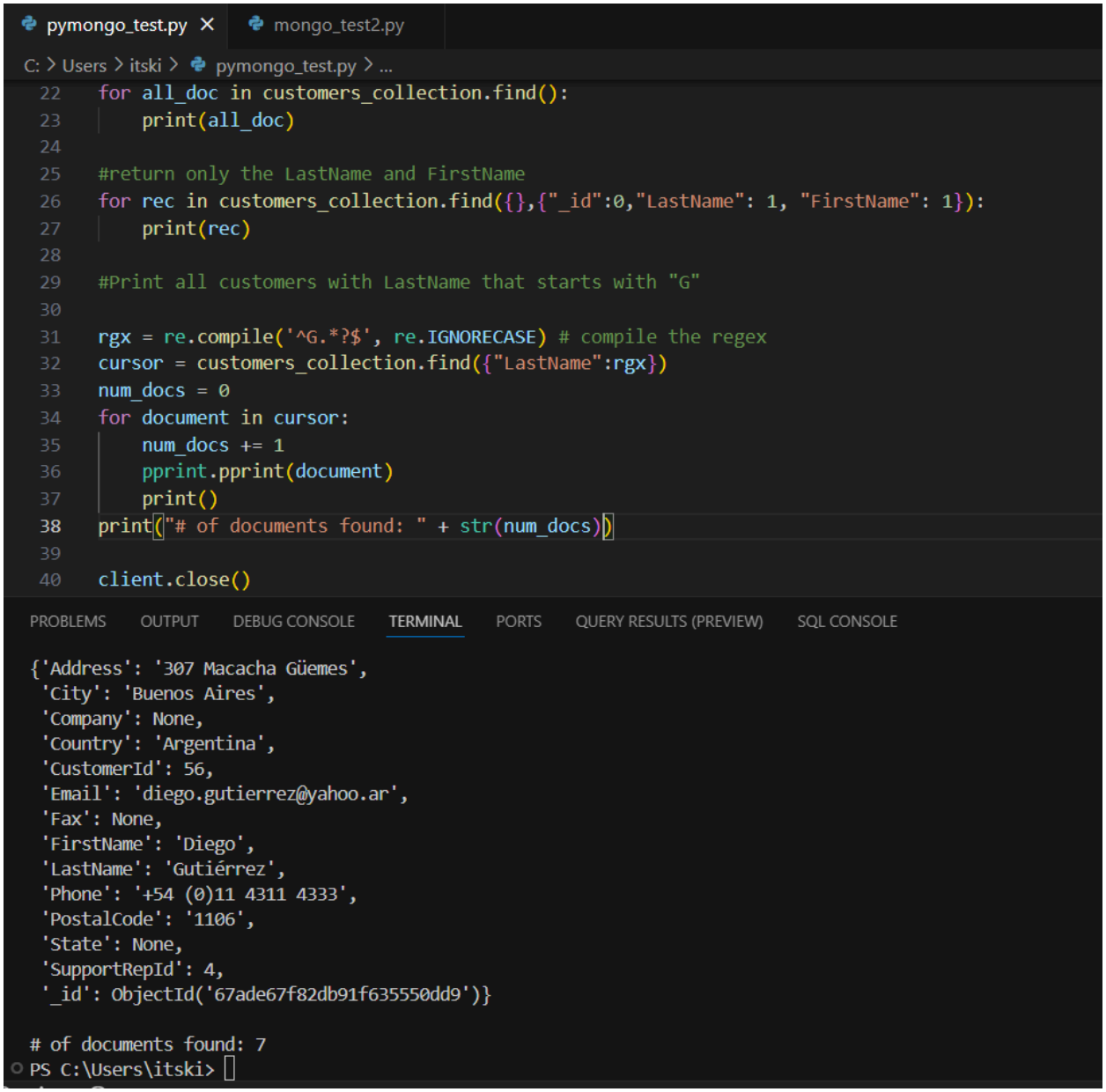
Print all documents of the customer's collection



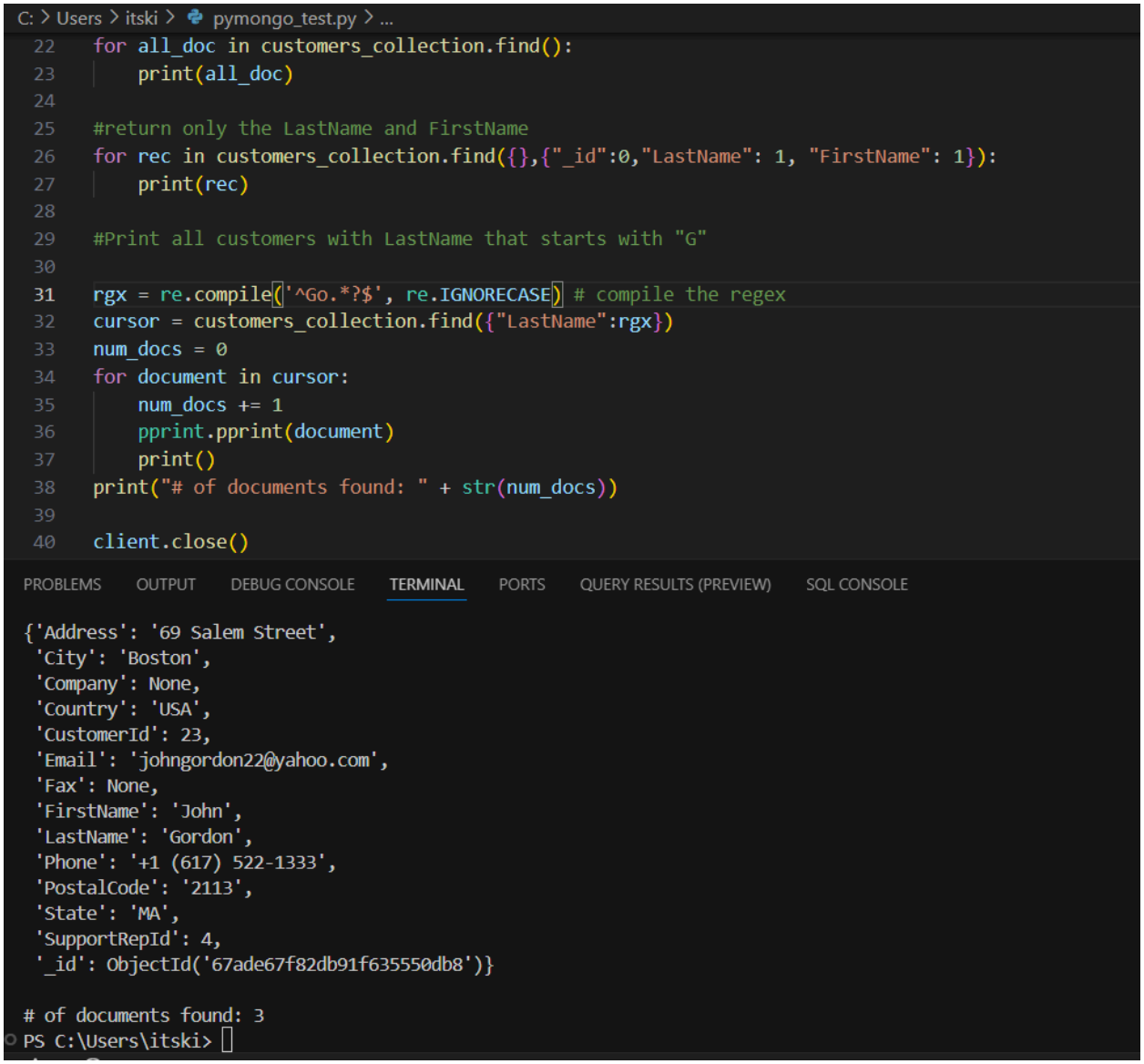
Printing all documents



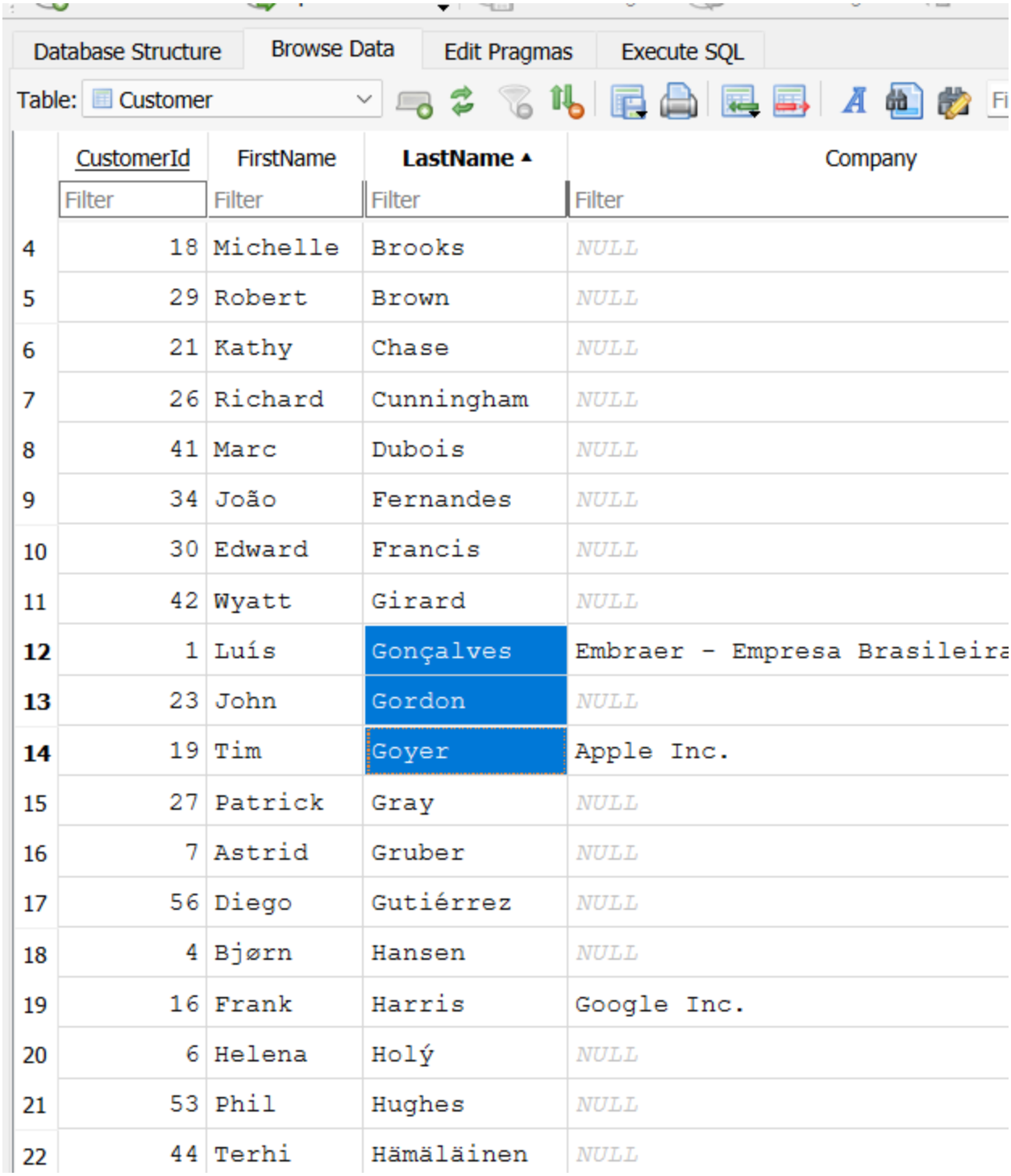
Return only the last name and first name



Print all customers with LastName that starts with “G”



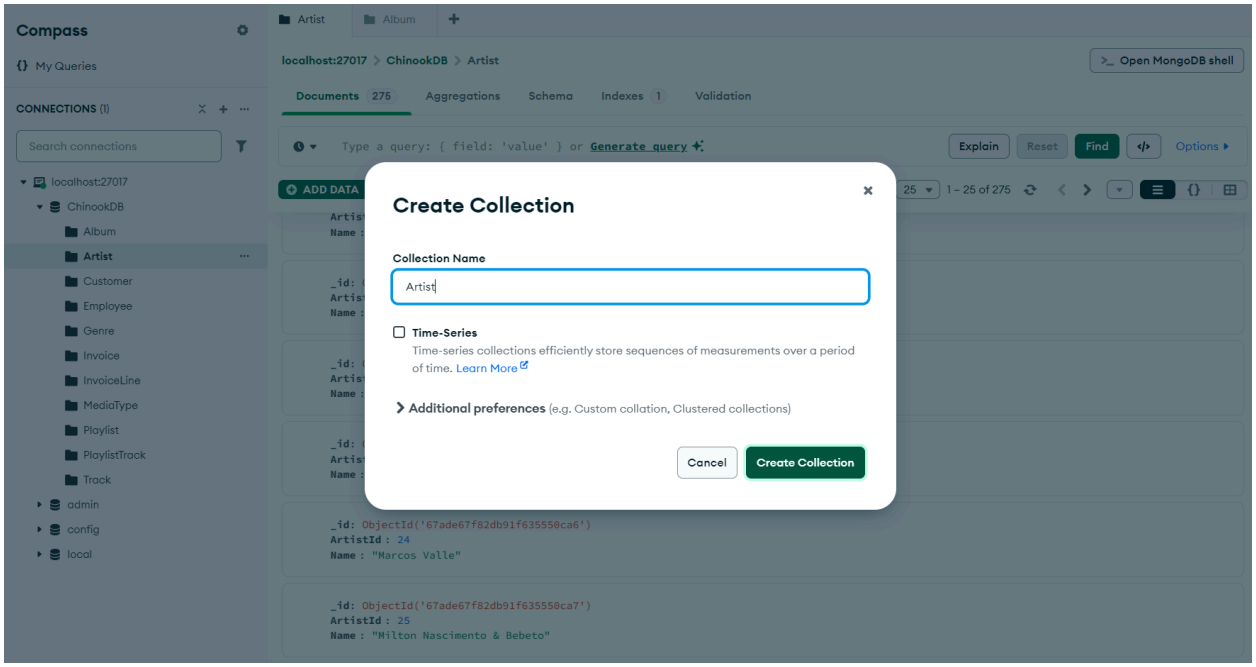
Print all customers with LastName that starts with “Go”



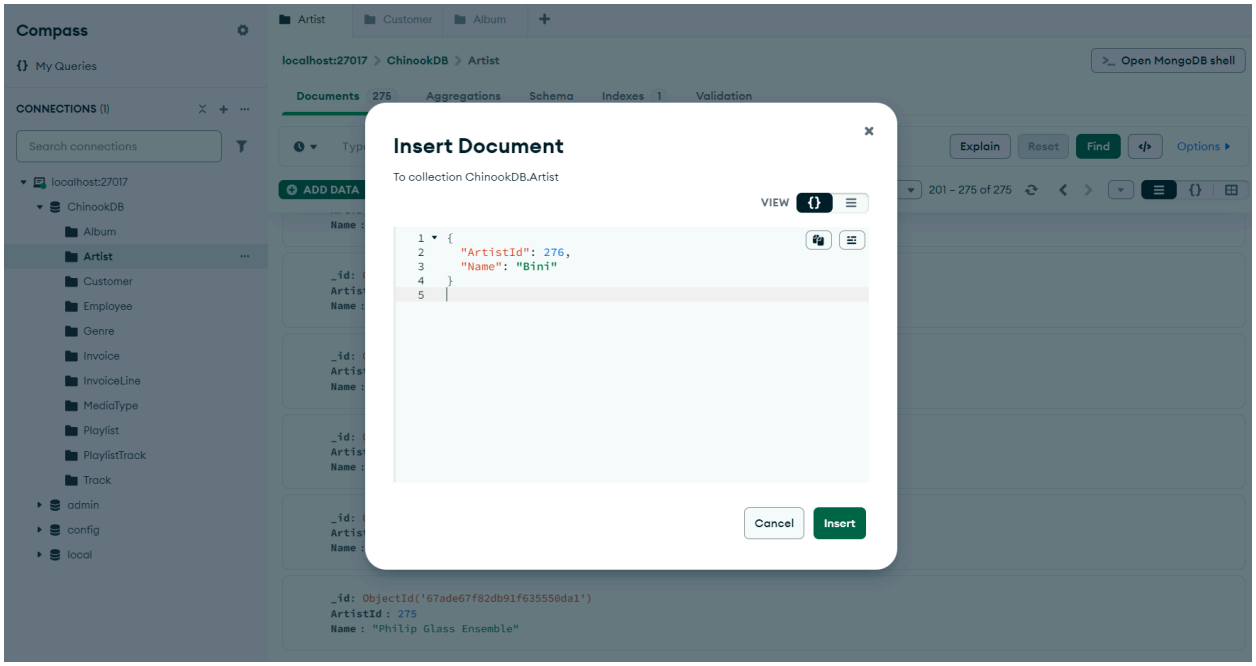
	CustomerId	FirstName	LastName	Company
	Filter	Filter	Filter	Filter
4	18	Michelle	Brooks	NULL
5	29	Robert	Brown	NULL
6	21	Kathy	Chase	NULL
7	26	Richard	Cunningham	NULL
8	41	Marc	Dubois	NULL
9	34	João	Fernandes	NULL
10	30	Edward	Francis	NULL
11	42	Wyatt	Girard	NULL
12	1	Luís	Gonçalves	Embraer - Empresa Brasileira
13	23	John	Gordon	NULL
14	19	Tim	Goyer	Apple Inc.
15	27	Patrick	Gray	NULL
16	7	Astrid	Gruber	NULL
17	56	Diego	Gutiérrez	NULL
18	4	Bjørn	Hansen	NULL
19	16	Frank	Harris	Google Inc.
20	6	Helena	Holý	NULL
21	53	Phil	Hughes	NULL
22	44	Terhi	Hämäläinen	NULL

PostLab

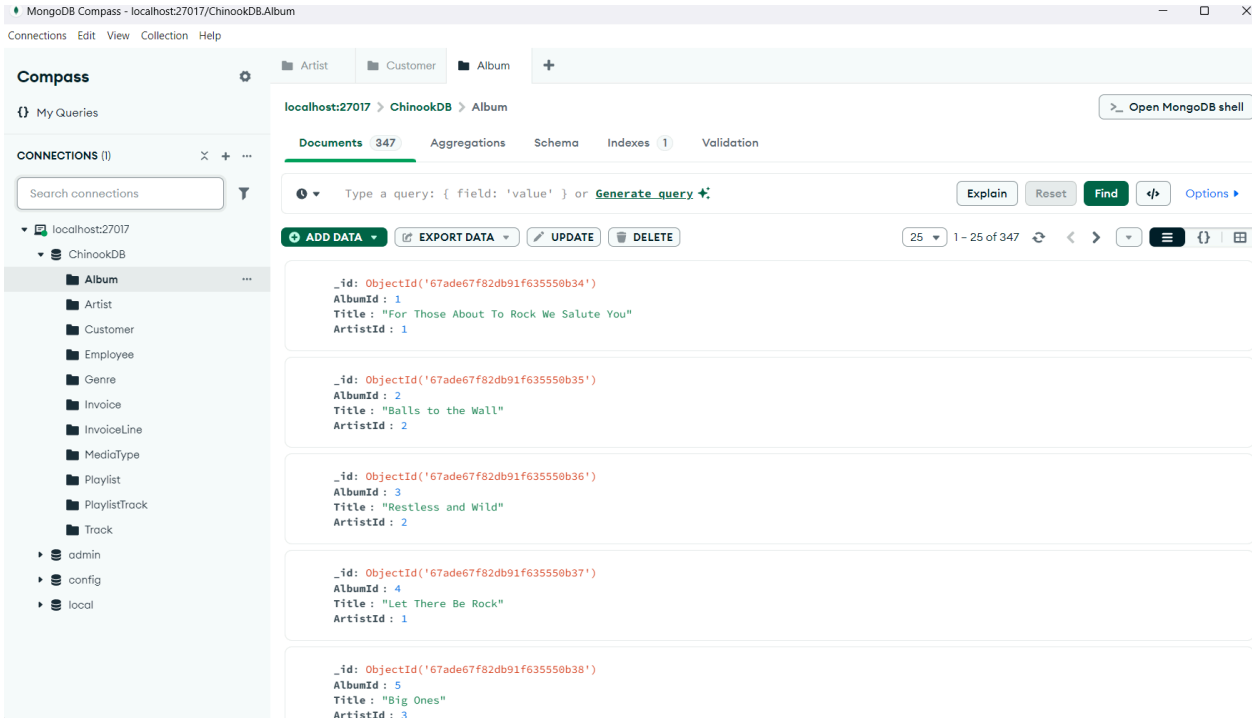
Using the ERD shown here >> Chinook DB ERD , create the artists-albums-tracks database in MongoDB compass



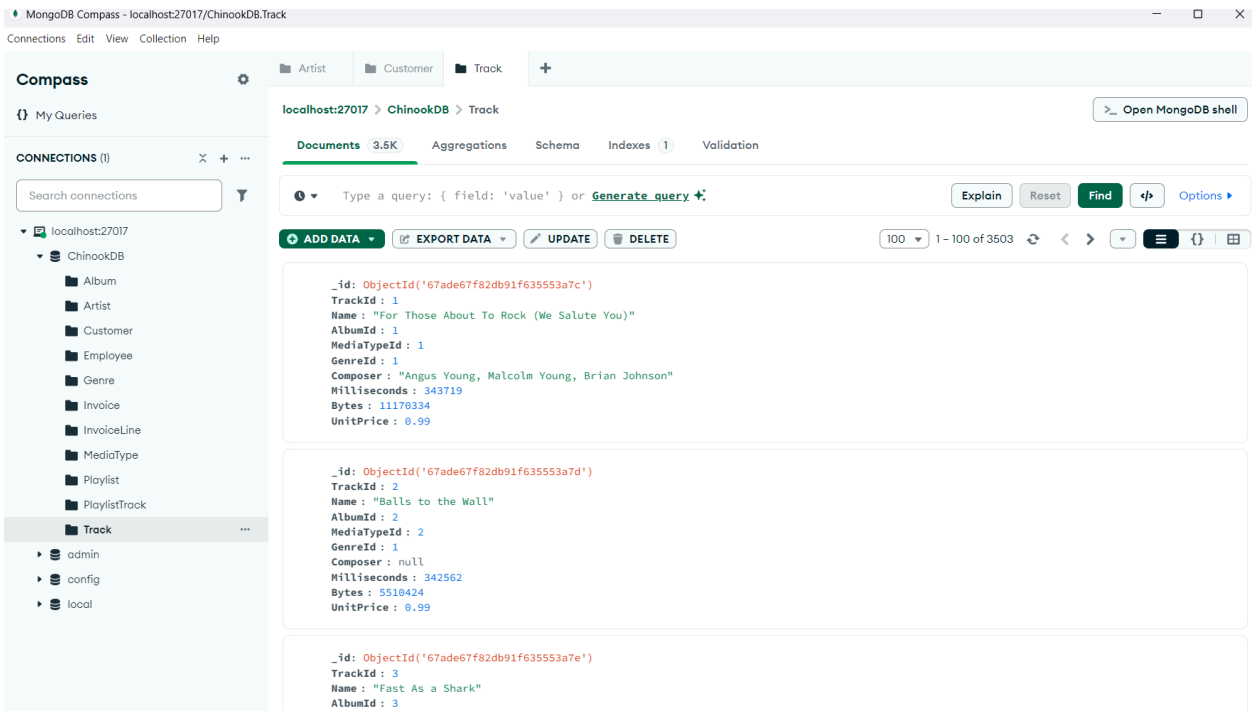
Creating the database “Artist.” The same process was used with creating the collection “Albums” and “Tracks”



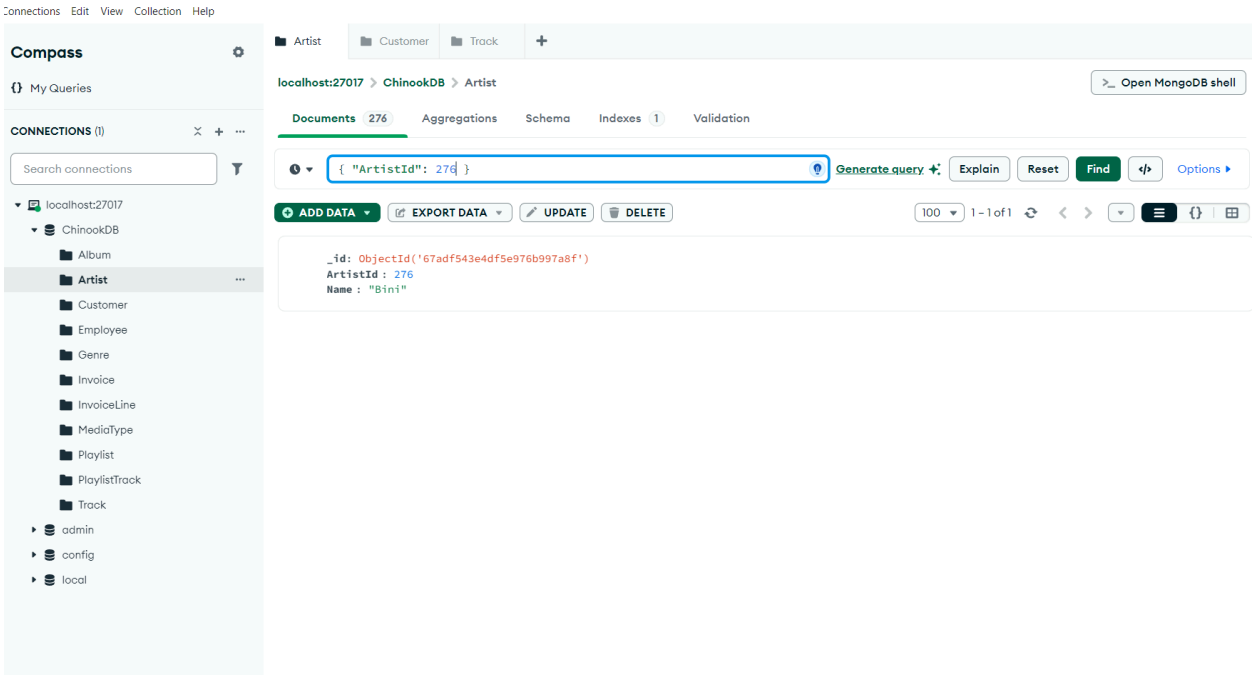
Adding sample data to the collection "Artist"



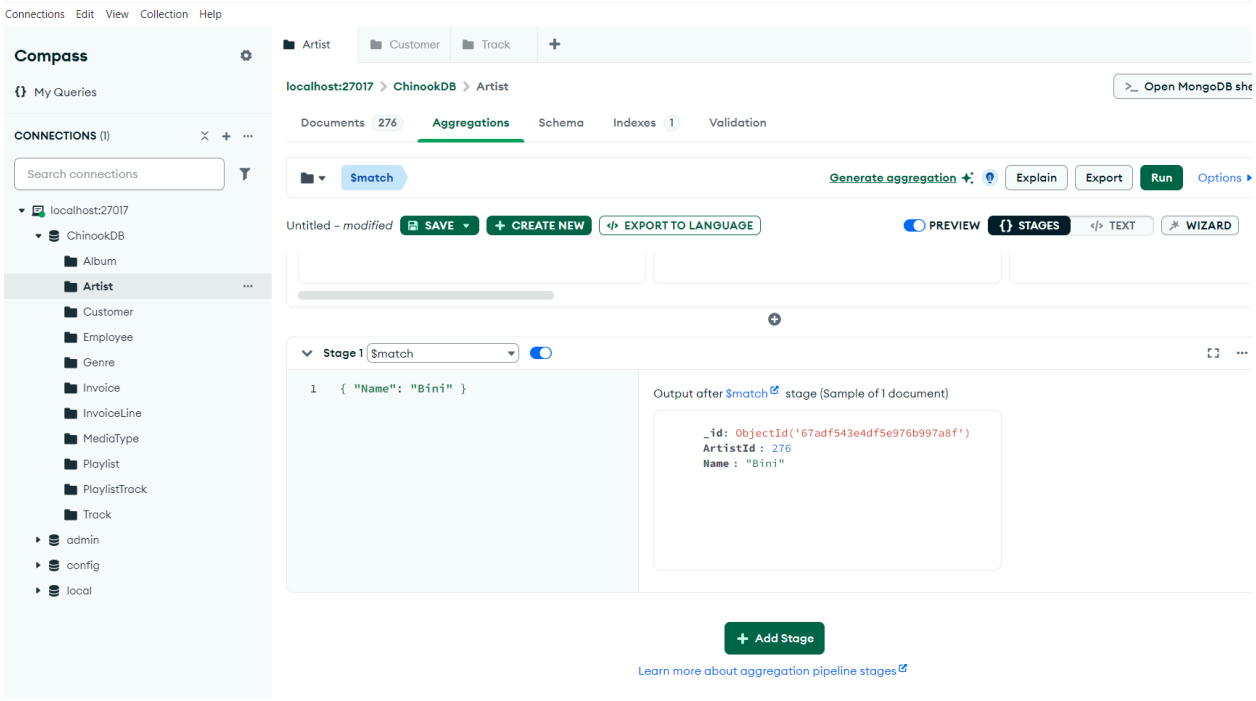
Database "Album"



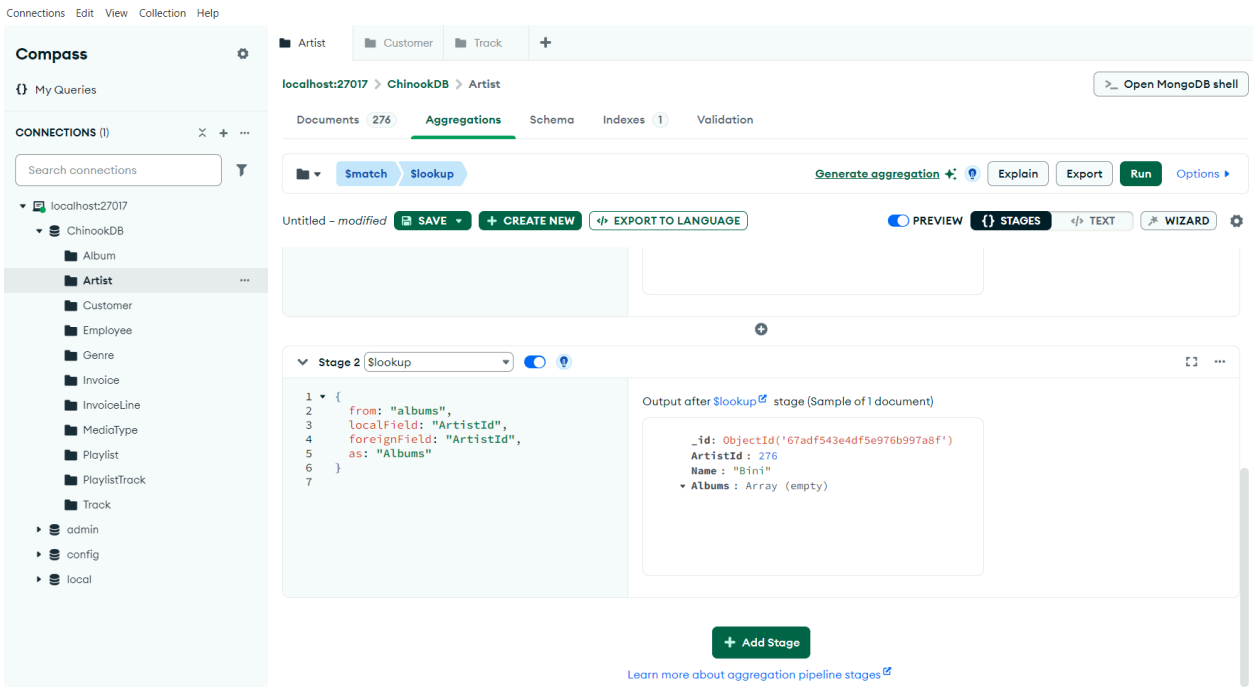
Database “Track”



Sample Query



Data relationship verification: Aggregation query (Stage 1)



Data relationship verification: Aggregation query (Stage 2)