MAPÚA UNIVERSITY
SCHOOL OF ELECTRICAL, ELECTRONICS, AND COMPUTER ENGINEERING

Experiment 6:
NoSQL database models

CPE106L (Software Design Laboratory)

Member 1 (Cromuel Pangilinan)
Member 2 (Karl Ignatius G. Gavino)
Member 3 (Nicole Allyson B. Torres)
Member 3 (Xavier Alonzo)

Group No.: **9**
Section: **B3**

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

Readings, Insights, and Reflection:

- Torres, Nicole Allyson B.
    - Chapter 9 in Lingras' book, "Building Cross-Platform Mobile and Web Apps for Engineers and Scientists: An Active Learning Approach," is a must-read for anyone serious about developing successful apps. The chapter covers critical aspects of app development from Section 9.1 to 9.6, offering insights and strategies that can help take your app to the next level. Starting with user interface (UI) design, Section 9.1 emphasizes the pivotal role of intuitive layouts and interactive elements in engaging users effectively. Lingras encourages engineers and scientists to prioritize functionality while maintaining a user-friendly aesthetic, which is crucial for modern app success. Moving on to navigation patterns and data organization in Section 9.2, Lingras highlights the significance of logical navigation paths and efficient data grouping to ensure smooth user experiences and information retrieval.

      Sections 9.3 and 9.4 delve into input controls and flexibility, guiding readers on implementing diverse input methods and responsive feedback mechanisms. Understanding different input scenarios and validation techniques is critical to creating versatile and user-centric apps. The exploration of multimedia integration in Section 9.5 highlights strategies for seamlessly incorporating images, videos, and audio elements, enhancing app content and engagement across platforms.Regarding accessibility considerations (Section 9.6), Lingras emphasizes the importance of inclusive design practices. We can learn about creating accessible user interfaces (UIs), including text alternatives, keyboard navigation support, and contrast considerations. T

- Pangilinan, Cromuel
    - o

- Gavino, Karl Ignatius G.
    - o

- Alonzo, Xavier
    - o

InLab

Objectives .

1. Create and modify tables for the database.
2. Using DB Browser for SQLite
3. Check if SQLite is installed in the system using the Linux terminal.

- **Tools Used**
  - Anaconda
  - Microsoft Visual Studio Code 2022
  - DB Browser for SQLite

- **Procedure.**

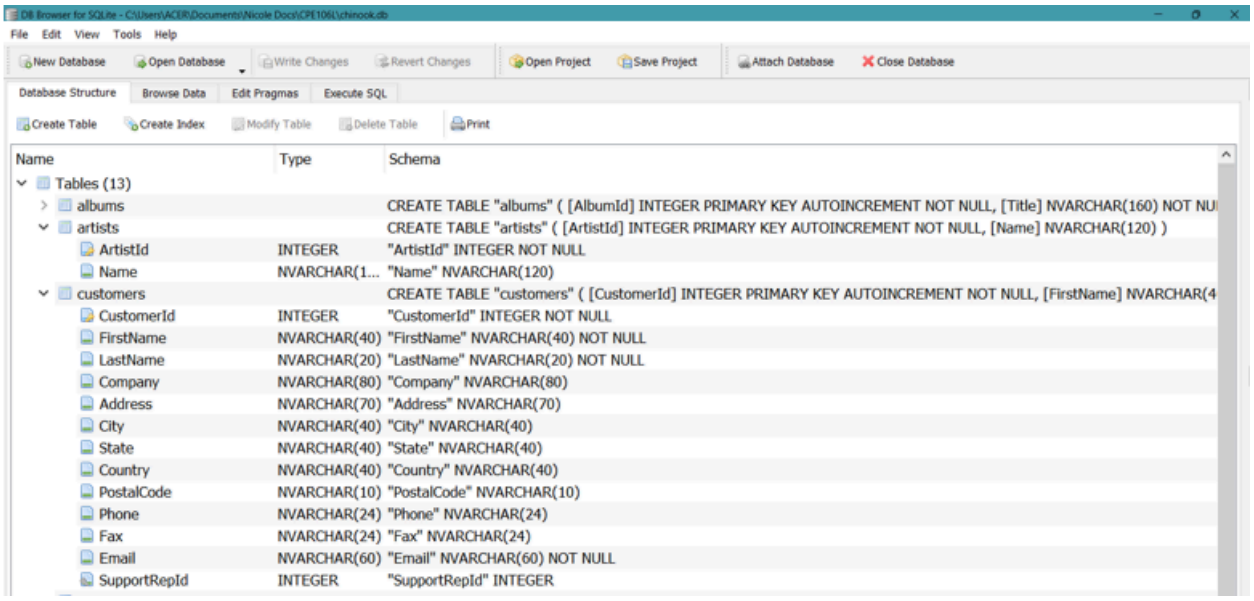
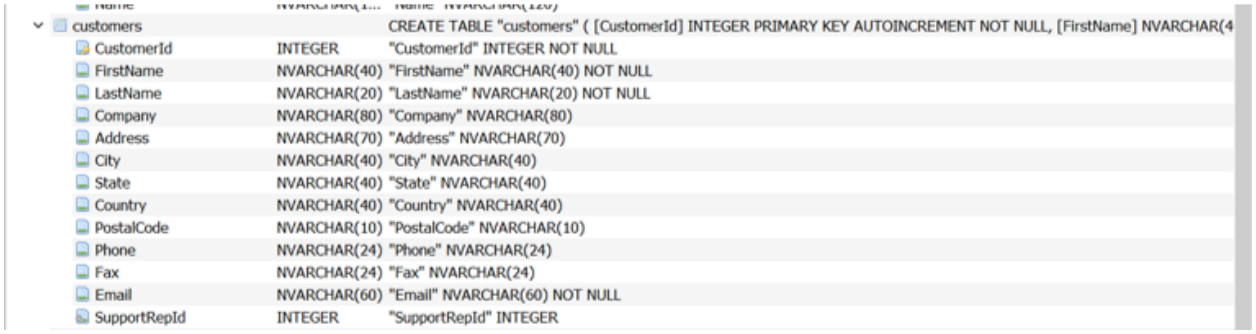


Figure 1.1. Opening chinook Database

In *Figure 1.1* we tried to navigate through the Chinook database using DB browser for SQlite.



customers		CREATE TABLE "customers" ([CustomerId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, [FirstName] NVARCHAR(4
CustomerId	INTEGER	"CustomerId" INTEGER NOT NULL
FirstName	NVARCHAR(40)	"FirstName" NVARCHAR(40) NOT NULL
LastName	NVARCHAR(20)	"LastName" NVARCHAR(20) NOT NULL
Company	NVARCHAR(80)	"Company" NVARCHAR(80)
Address	NVARCHAR(70)	"Address" NVARCHAR(70)
City	NVARCHAR(40)	"City" NVARCHAR(40)
State	NVARCHAR(40)	"State" NVARCHAR(40)
Country	NVARCHAR(40)	"Country" NVARCHAR(40)
PostalCode	NVARCHAR(10)	"PostalCode" NVARCHAR(10)
Phone	NVARCHAR(24)	"Phone" NVARCHAR(24)
Fax	NVARCHAR(24)	"Fax" NVARCHAR(24)
Email	NVARCHAR(60)	"Email" NVARCHAR(60) NOT NULL
SupportRepId	INTEGER	"SupportRepId" INTEGER

Figure 1.2. Viewing Customer Table

In *Figure 1.2,* we opened the table named “Customer” and reviewed the values in each fields inside the table.

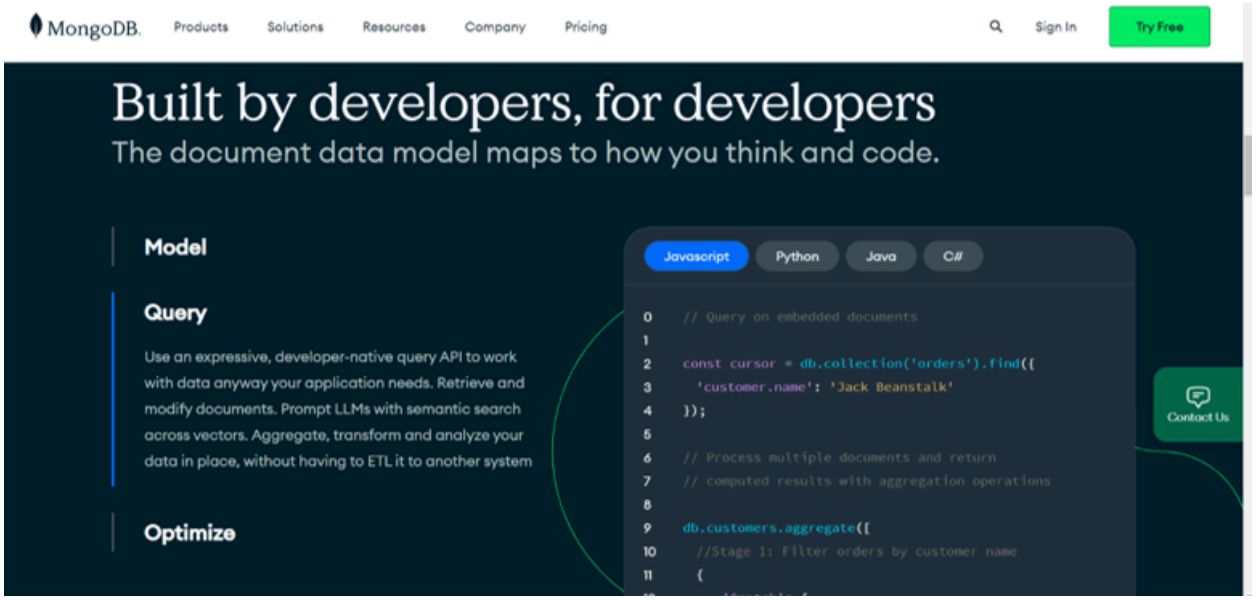


Figure 1.3. Installing MongoDB

Figure 1.3 shows the installation of MongoDB in the system to be used in creating databases and other collections.

Figure 1.4 Creating chinook database

In *Figure 1.4* we created a database chinook with a collection named as “Customer” in mongoDB. We then import the chinook database file to our new database in MongoDB.

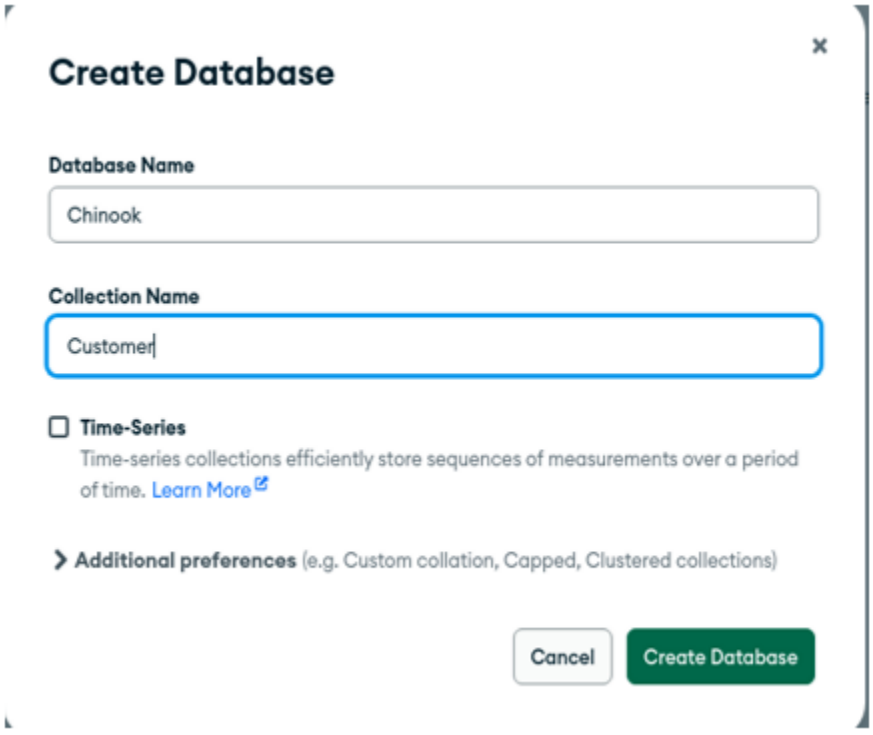


Figure 1.5 Creating collection “Customer”

In Figure 1.5, after we created the database chinook, we also created the collection named “Customer” inside the chinook.

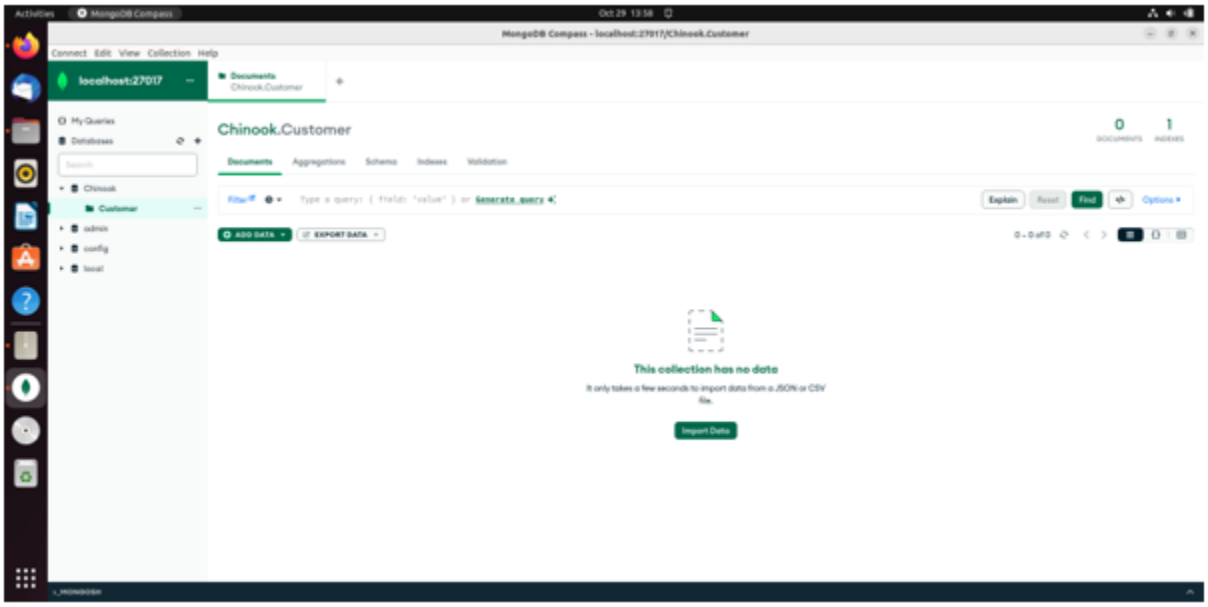


Figure 1.6 Importing the database in MongoDB

In *figure 1.6* we imported the chinook database file inside the MongoDB and specifically access the “Customer” table.

PostLab

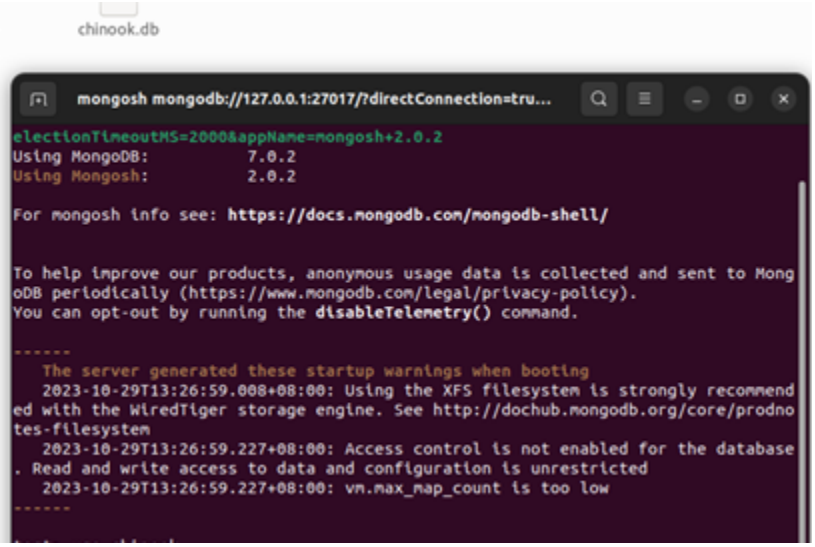


Figure 2.1 mongosh

Figure 2.1 shows the terminal when executing the “mongosh” command

Figure 2.2 Using Chinook as Database

In *Figure 2.2* we tested if we are able to access the chinook database.

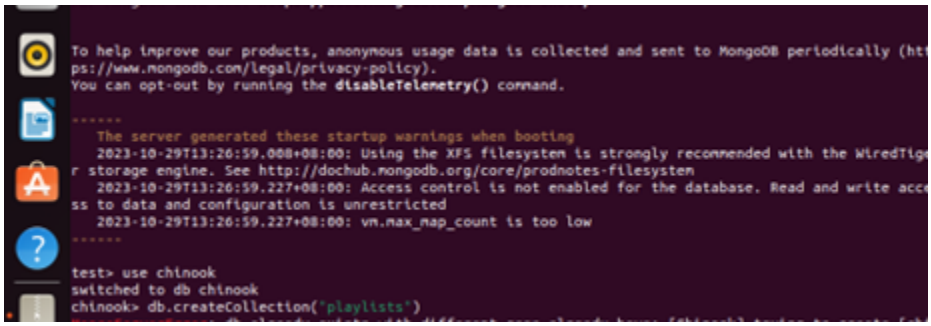


Figure 2.3

Figure 2.3 demonstrates the creation of the collection inside our new database “Chinook”. We created the collection named “Playlists” and added values in each such as, “playlistsID”, “composer”, “genre” and many more.