Group Symbol: E

Student One Name: Eskandar Atrakchi

Subject: Advanced Databases

Submission Date: 15th/Nov/2023

Submission Time: Before 23:55

Project Weight: 40% Overall

Contents

[**Introduction** 3](#_Toc159453000)

[**Business Requirements** 3](#_Toc159453001)

[**Student One: Eskandar Atrakchi** 3](#_Toc159453002)

[**Dimensional Model** 3](#_Toc159453003)

[**DDL Script** 4](#_Toc159453004)

[**Data Loading** 7](#_Toc159453005)

[**DML Script** 7](#_Toc159453006)

[**Explaining the modification done to python code** 8](#_Toc159453007)

[**Visual Studio Code** 10](#_Toc159453008)

[**Using VSCode to make python libraries useable in jupyterlab** 10](#_Toc159453009)

[**Using VSCode to make the connection possible between jupyterlab and MySQL** 10](#_Toc159453010)

[**Queries** 10](#_Toc159453011)

[**Student One: Eskandar Atrakchi** 10](#_Toc159453012)

[**Optimizations** 14](#_Toc159453013)

[**Student One: Eskandar Atrakchi** 14](#_Toc159453014)

[**GitHub** 16](#_Toc159453015)

[**Conclusion** 18](#_Toc159453016)

# **Introduction**

In this project we will gather the business requirements and make a structure of our business database module, then will visualize it using Lucid.app to make it easy to code it using SQL then later with python to implement the business requirement on a plot.

We will use all the resources online to try to implement queries and will mention resources along the way.

This is done by using MySQL, VSCode, GitHub, and Jupyterlab.

# **Business Requirements**

Our business is on leisure and Training (Module – Students) so we were planning to establish five tables in total, four dimensional tables and one fact table.

## **Student One: Eskandar Atrakchi**

The business requirement is to retrieve the performance score for each student individually and compare students with each other, the business should have a requirement of performance score which it starts from 0% and ends at 100% students can achieve performance score only between or equal those two percentages.

The business should be able to compare the performance score of students themselves and show who is average, below average, and above average.

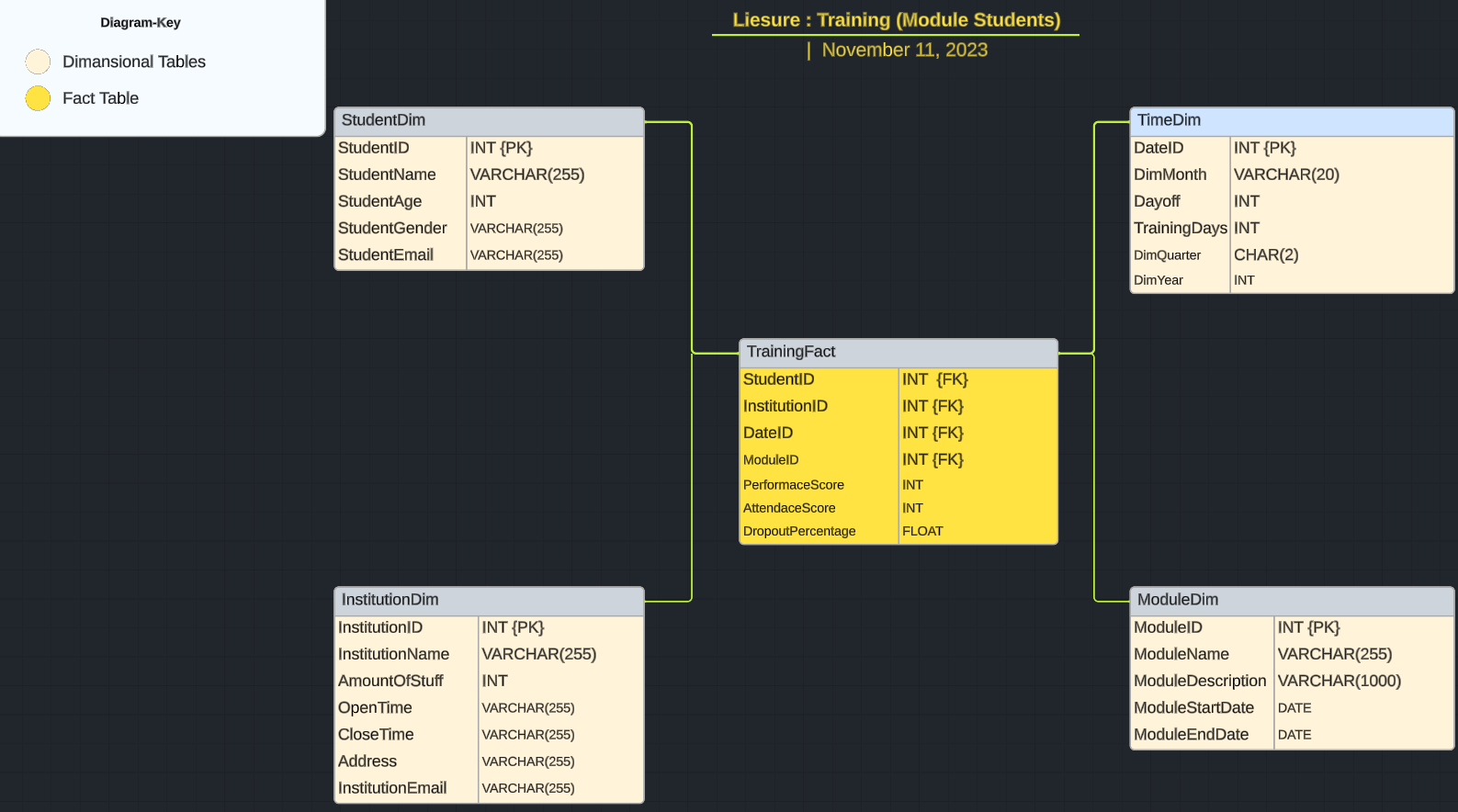
# **Dimensional Model**

We have chosen a star-like structure that contains five tables in total, four dimensional tables (StudentDim, TimeDim, ModuleDim, InstitutionDim) and one fact table (TrainingFact).

We have chosen the data types according to the business requirements.

All the relationships are from one-to-many dimensional tables to the fact table respectively.

All the other clarifications are mentioned on the tables visual representation down below.



# **DDL Script**

The scripts of creating tables were created using VSCode and were committed on GitHub then the scripts were gathered to be implemented on MySQL.

1. We have created database using this statement CREATE DATABASE Assig;  
   Assig is just the name of the database.
2. We use the database by running this statement USE Assig;.  
   Here is a screenshot.

A screenshot of a computer

Description automatically generated  
the green check means we don’t have any error creating and using the database.

1. We take the create tables codes in MySQL and then execute.

A screenshot of a computer

Description automatically generated  
Now the tables have been created in MySQL, but they are all empty

Here is the code of for the tables creation just for reference

A screen shot of a computer program

Description automatically generated

# **Data Loading**

## **DML Script**

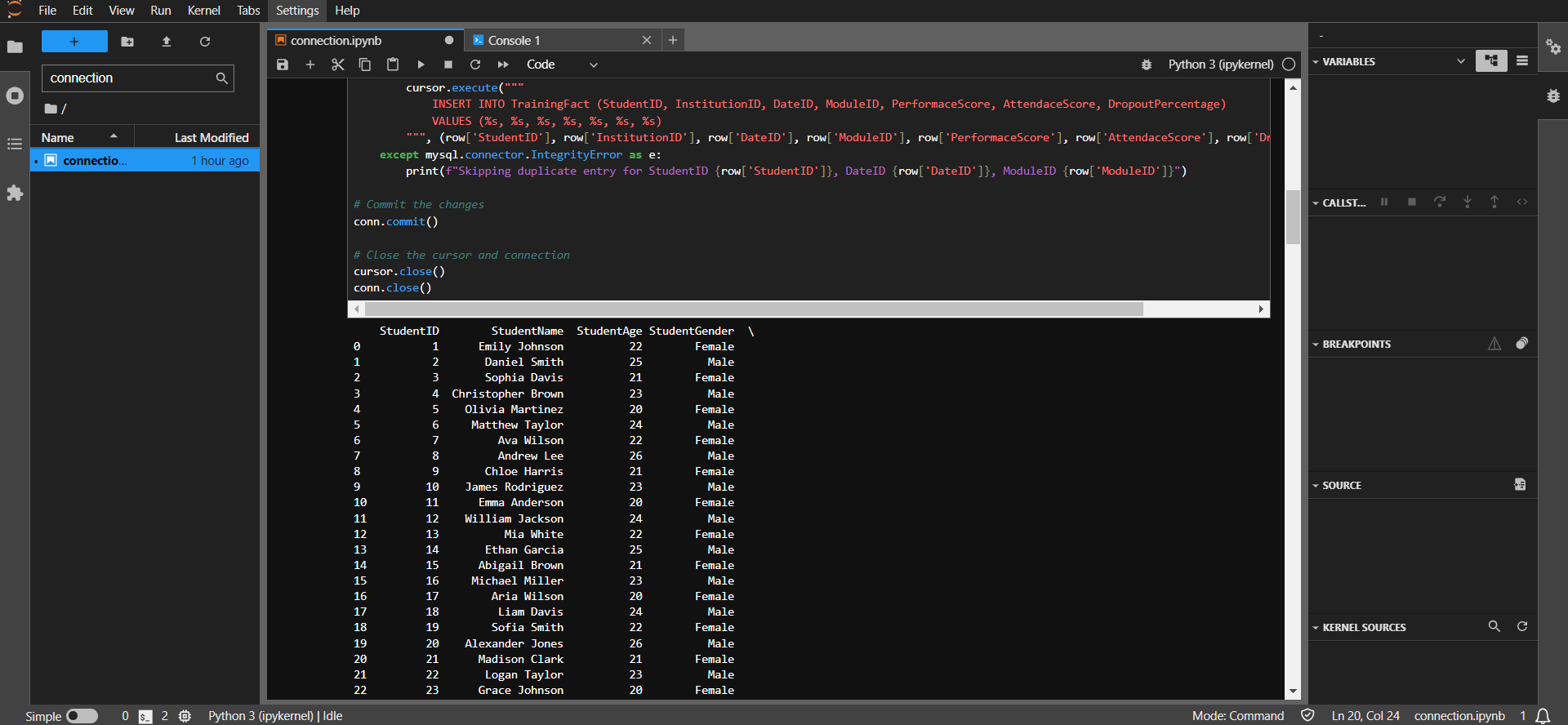
We have done this in two different ways.

1. Is buy just running SQL statements to store data in the database, just simple as that and here is the code that was coded on VSCode for reference.

A screenshot of a computer program

Description automatically generated

1. The second way of doing this, grabbing the data that is in JSON file and to store the data in MySQL tables we ran a for loop on it to store the data according to each column.  
   We got help from [Stack overflow](https://stackoverflow.com/questions/71983339/how-do-i-take-my-json-data-and-put-it-into-a-mysql-table-with-python) to figure this one out.  
   Because a connection to MySQL was required here to finish the storing the data to MySQL tables by just running python script



## **Explaining the modification done to python code**

We were running each code individually and trying to connect each time to MySQL to get the for loop works on each table individually. Here is the code for one table before modifications.

A computer screen shot of a program code

Description automatically generated

We were trying to establish a connection between MySQL and Jupyterlab then running a for loop to store the data in TimeDim table that exists in MySQL.

The problem is we have more than one table that means each running code we need to make sure that the connection is established correctly.

We had so many errors and it was way too difficult to handle the errors in python, after some online researches, we knew that the errors were about the connection to MySQL.

We have changed the way of storing the data by modifying the five python codes (to store data in five tables) by merging them into one script, in this case we can handle the connection because we can connect once only and run the script to store all the data in five tables in MySQL by running five for loops in one python code and here is the code after modifications.

The reading from JSON files and connection

Noting that the JSON dataset were generated by [chatGPT](https://chat.openai.com/) because we were in a very limited time to find data that same structure as our database, so we used GPT-generator.

A computer screen shot of a program

Description automatically generated

The (for) loops to insert data to the tables in MySQL.

A screenshot of a computer program

Description automatically generated

To store data in five existing tables in MySQL we need to use five for loops, instead of establishing five times connections to MySQL and execute five python codes (one code for each table), we just put all the for loops in one python code and establish the connection only once.

# **Visual Studio Code**

## **Using VSCode to make python libraries useable in jupyterlab**

We have used the following libraries in VSCode terminal to make our python code possible to run on Jupyterlab

1. pip install matplotlib
2. pip install pandas

## **Using VSCode to make the connection possible between jupyterlab and MySQL**

We have used the following libraries in VSCode terminal to make the connection possible between Jupyterlab and MySQL

1. pip install sqlalchemy
2. pip install mysql-connector

# **Queries**

## **Student One: Eskandar Atrakchi**

According to the business requirements that been stated in the beginning of this document.

I have coded this using pandas library and by help of [stack overflow](https://stackoverflow.com/questions/71454857/make-a-bar-graph-visualization-using-python-with-data-from-sql-database).

A screenshot of a computer

Description automatically generated

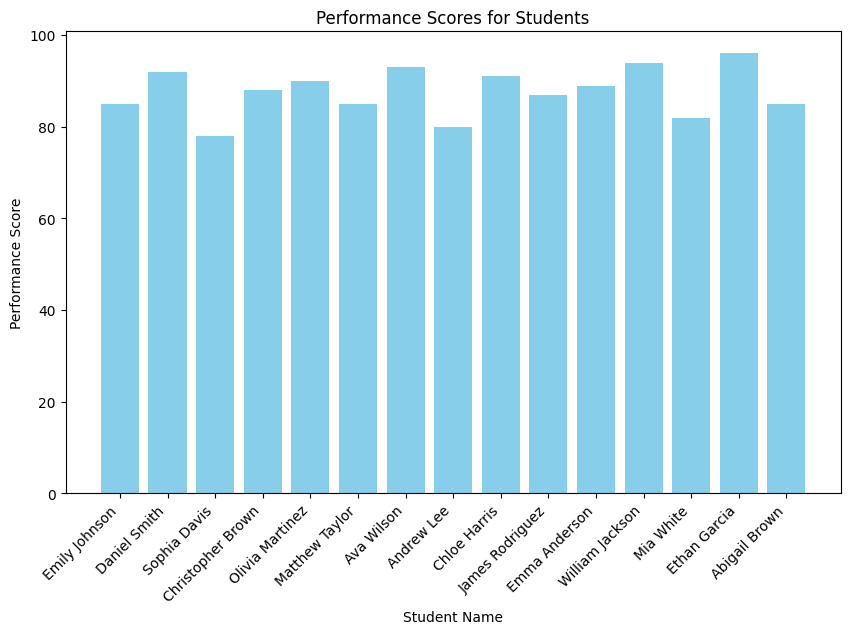
If a plot is implemented, the bar length will indicate the student performance.

A screenshot of a computer

Description automatically generated

Each bar should contain the student’s name that is stored in the database so we will have reference for each bar representing a known student in the system.

This is the ending result:



This chart also compares students with each other, for example it’s clearly that Ethan Garcia is doing better performance than Abigail Brown.

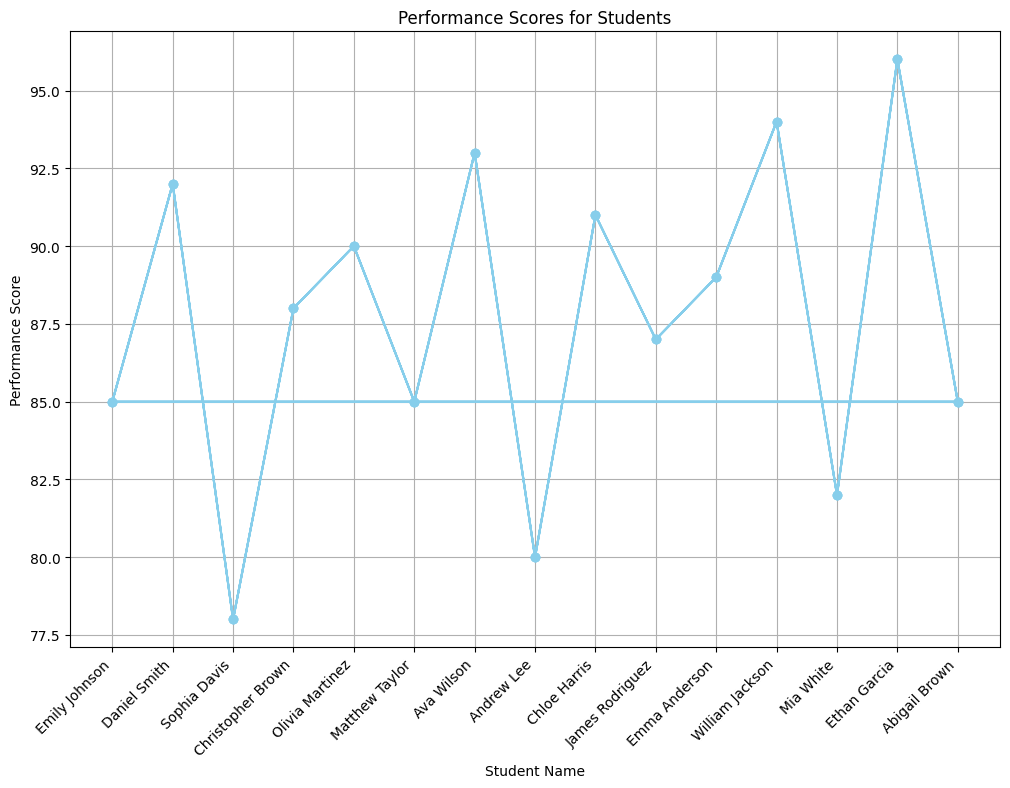
We can represent the data in a better way, there are a lot of libraries that we can use for our advantage, for example, we can calculate the average performance for all the students combine.

A screen shot of a computer

Description automatically generated

As we can see, it reads the performance of each student that is stored in the database and at the same time the average for the performance of all students all together, this is useful if we want to know if a certain student is doing good, bad, or average and of course the average line will determine that.

Here is the final results for better clarity.



We can see that Ethan Garcia is doing better performance than Abigail Brown at the same time we can see that Abigail Brown is doing average performance and Ethan Garcia is doing better than average performance.

# **Optimizations**

## **Student One: Eskandar Atrakchi**

To enhance the performance of my query search into my database tables I have created indexes and here are the codes of the two indexes.

A screenshot of a computer

Description automatically generated

The reason why I have chosen those two tables to preform indexes on them it’s because I’m trying to get the performance (which exist in TrainingFact table) of each student (which exist in StudentDim table)

So, my indexes are in TrainingFact table and StudentDim table. Just telling the database to create a primary index named idx\_StudentDim\_StudentID on StudentDim, and for the second line saying create a secondary index named idx\_TrainingFact\_StudentID to speed up searches for information based on the StudentID in the TrainingFact table.

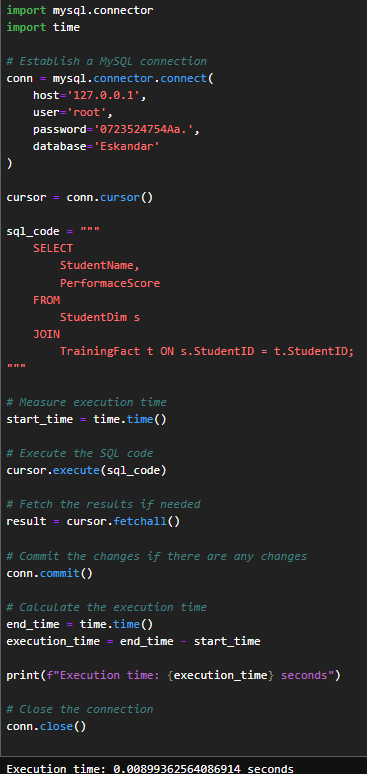
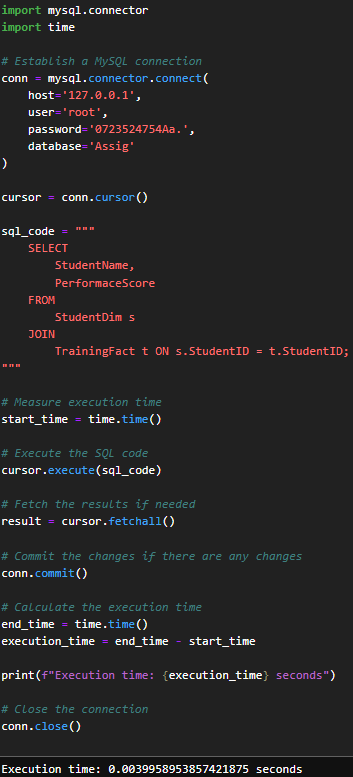
The results after implementation of indexes in MySQL

A screenshot of a computer

Description automatically generated

The green checks marks means there are no errors.

The effect is positive because if I create another database, and I name it Eskandar and I already have database called Assig knowing that I have done indexing to my Assig database as it shows in the screenshot above.

The same query for both of them.

The running time is different.

The name of the database is different.

Indexing helped to run query on my database in faster time, I have run python code multiple times just to make sure that this number is pure and was not affected by windows update or whatsoever.

# **GitHub**

I have committed everything to GitHub and this graph shows the work has been done to this project.

A screenshot of a computer

Description automatically generated

The project, from its beginning to the end is committed to GitHub on this username EskandarAtrakchi

Or simply click on this link to access the files on GitHub [Here](https://github.com/EskandarAtrakchi/NCI-YEAR-2-FIRST/tree/main/Advanced%20Databases/AssigOne)

# **Conclusion**

Before start coding it’s important to define the requirements and visualize the structure of the database.

The coding of SQL was done though VSCode in connection with MySQL, inserting the data was done in running five for loops using one python code to manage the connection between Jupyterlab and MySQL.

After running the query python on Jupyterlab I can see the results as a bar-chart that shows details of what has been stored in the system, the running time of the query can be unfair, the performance of the querying can be enhanced by indexing the foreign key in the fact table and the same key shows as primary in one of the dimensional tables would make a big difference especially if we were running big queries with big data.

Finally, this document provides a valuable resource of understanding the project structure and implementation details.