SUPPORT SPHERE

PROJECT REPORT

GROUP 12

KAVYA PATI BANDLA

LIKHITHA GUTTAPALLI

617-637-6694

857-204-7868

[patibandla.ka@northeastern.eduu](mailto:patibandla.ka@northeastern.eduu)

[guttapalli.l@northeastern.edu](mailto:guttapalli.l@northeastern.edu)

Percentage of effort contributed by Kavya: 50%

Percentage of effort contributed by Likhitha: 50%

Signature of Kavya: 

Signature of Likhitha:

Submission Date: 04/20/2024

**SUPPORT SPHERE**

In this era of global connectivity, the journey of immigration is overwhelming and requires a dedicated support system for guidance and encouragement. Support Sphere is one such platform that helps connect the Immigrants with people who are already settled in the destination country to seek advice and moral support. This platform by harnessing the strength of human relations envisions in creating a diverse community which goes beyond the simple information exchange, by sharing insights about the new cultural, individual experiences and the earnest desire of helping one another succeed in this unfamiliar environment.

**Theory for Support Sphere:**

Our initiative aims at assisting students in their plans to pursue their education from international institutions. Upon signing up on the website and providing personal details, users will be guided through tips, document requirements, and the entire process essential to complete this transition smoothly. Based on the processing stages of the Visa the Users are divided as Initial User, Intermediate User and Neophytes

If the user is in the Initial User stage information such as Education details, GRE Scores, IELTS Score, Colleges Applied, I20, DS160 and Visa status are collected. Based on the details provided an expert is connected to guide the user in the further visa processing.

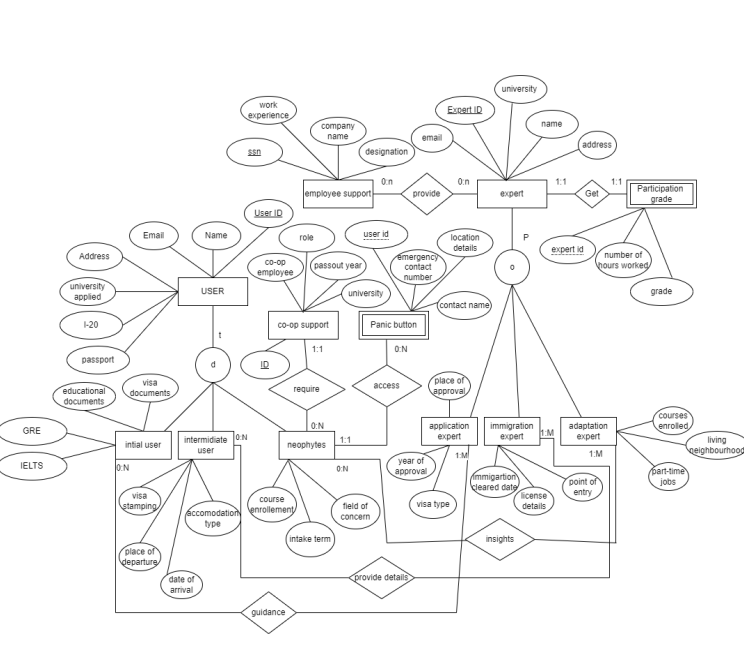
If the user is in the Intermediate stage details such as the College admitted, Approval letter, I20, Passport number, Visa page, Place of departure and Date of arrival are recorded. The user is given a chance to connect with the experts, who will guide them for a smooth transition and explain regarding the immigration process and will also help with co-passenger details travelling on the same day if available.

Neophytes are the students who have recently arrived in the country and are seeking guidance related to course registrations, accommodation and finding roommates, part time job opportunities, safety measurements and various means of commutes. Information such as personal details, course enrollment, term of studies, date of arrival and the field of concern are collected from them.

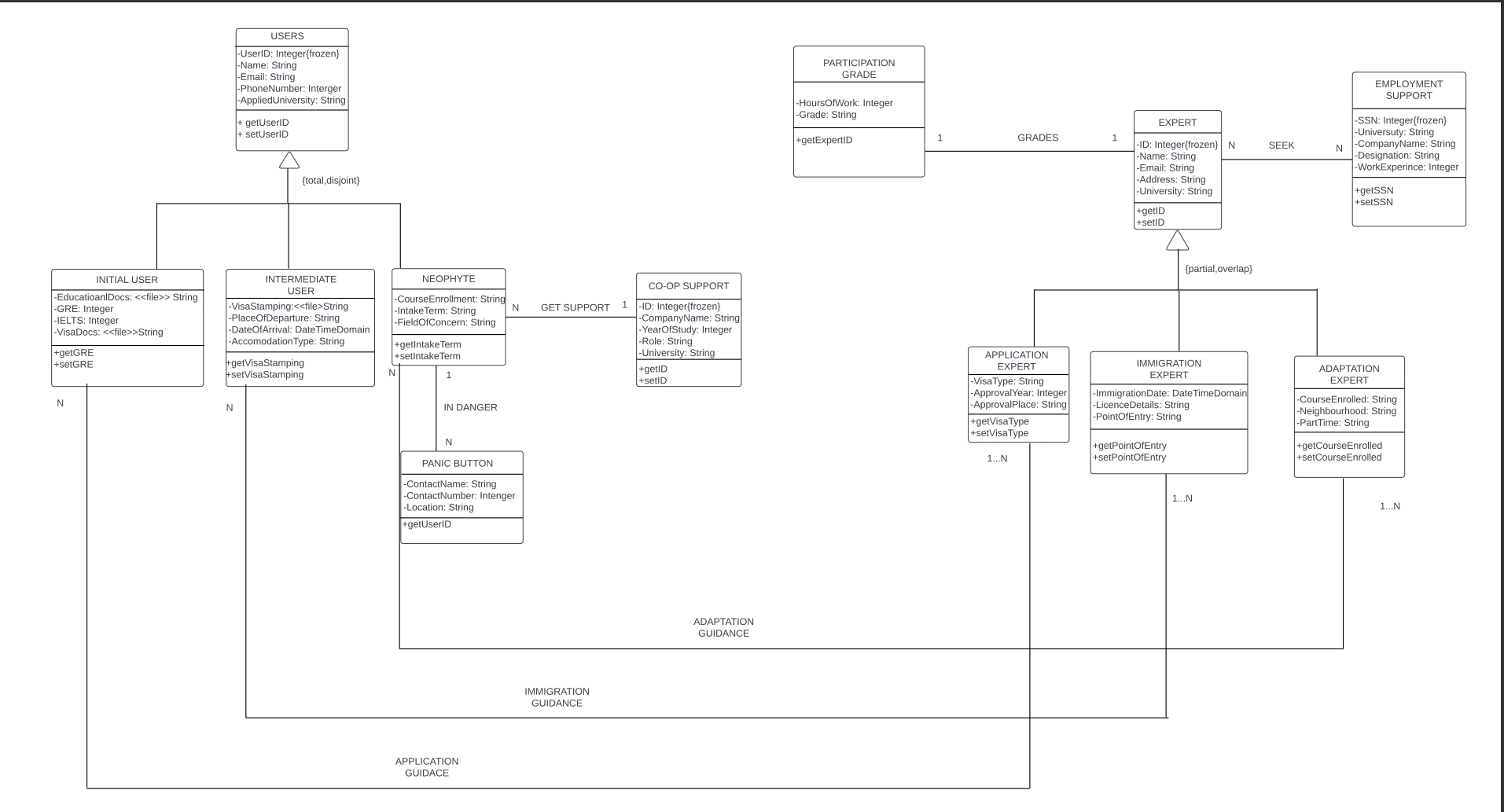
Experts are the already established and seasoned professionals who are ready to guide and help the Immigrants during their transformation period, an immigrant who have completed their transformation period through Support sphere are also considered to be experts and are given the privileges to help others if interested.

A special SOS feature is provided for the Neophytes and the Experts which acts as the safety feature or panic button, when any of them feel unsafe and would want help they will automatically be connected to one another, and location details can also be shared.

**EER MODEL**



**UML MODEL**



**LIMITATIONS IN THE EER AND UML:**

* The Initial User, after the initial step's completion, can continue to seek support for the further process by turning into an intermediate user if interested.
* In the same way an Intermediate user can also be further moved on to a neophyte.
* A neophyte after their Co-op can also join the Co-op support experts.
* A neophyte after the completion of graduation can enroll as an expert to help others.
* An Expert after they secure a job, can also help by joining as an Employment support.
* The experts/ people in Employment Support should have a minimum of two years' Experience.

**RELATIONAL MODEL**

User (UserID, Name, Email, PhoneNo, AppliedUniversity, Address, I20, Passport)

Initial User (UserID, Name, Email, PhoneNo, AppliedUniversity, Address, I20, Passport, EducationDocs, GRE, IELTS, VisaDocs)

Intermediate User (UserID, Name, Email, PhoneNo, AppliedUniversity, Address, I20, Passport, VisaStamping, PlaceOfDeparture, DateOfArrival, AccomodationType)

Neophyte (UserID, Name, Email, PhoneNo, AppliedUniversity, Address, I20, Passport, CourseEnrollment, IntakeTerm, FieldOfConcern, *Co-op\_ID*)

*Co-op\_ID* – Foreign key, refers to Co-op\_ID from Co-op Support

specified as NOT NULL

Expert (ID, Name, Address, Email, University, *Grade*)

*Grade –* Foreign Key, refers to Grade from Participation Grade

specified as NOT NULL

Application Expert (ID, Name, Address, Email, University, VisaType, ApprovalYear, ApprovalPlace)

Immigration Expert (ID, Name, Address, Email, University, ImmigrationDate, LicenseDetails, PointOfEntry)

Adaptation Expert (ID, Name, Address, Email, University, CourseEnrolled, Neighbourhood. PartTime)

Panic Button (*UserID*, ContactName, ContactNumber, Location)

*UserID –* Foreign key, refers to UserID from Neophytes, specified as NOT NULL

Co-op Support (Co-op\_ID, CompanyName, Role, YearOfStudy, University)

Participation Grade (*ID*, HoursOfWork, Grade)

*ID –* Foreign key, refers to ID from Expert, specified as NOT NULL

Employment Support (SSN, University, CompanyName, Designation, WorkExperiance)

Application Guidance (*UserID, ID*)

*ID –* Foreign key, refers to ID from Application Expert, specified as NOT NULL

*UserID –* Foreign key, refers to UserID from Initial User, specified as NOT NULL

Immigration Guidance (*UserID, ID*)

*ID –* Foreign key, refers to ID from Immigration Expert, specified as NOT NULL

*UserID –* Foreign key, refers to UserID from Intermediate User

specified as NOT NULL

Adaptation Guidance (*UserID, ID*)

*ID –* Foreign key, refers to ID from Adaptation Expert, specified as NOT NULL

*UserID –* Foreign key, refers to UserID from Neophytes, specified as NOT NULL

Seeking Employment Support *(ID, SSN)*

*ID –* Foreign key, refers to ID from Expert, specified as NULL ALLOWED

*SSN –* Foreign key, refers to SSN from Employment Support

specified as NULL ALLOWED

**IMPLEMENTATION IN MYSQL**

**DDL COMMAND**

The table creation queries are written using DDL. One such query is shown below.

CREATE DATABASE IF NOT EXISTS supportsphere;

USE university;

Create table user

(UserId INT NOT NULL PRIMARY KEY,

Name VARCHAR (20) NOT NULL,

Email VARCHAR (30),

PhoneNo VARCHAR (20),

AppliedUniversity VARCHAR (20),

Address VARCHAR (40),

I20 VARCHAR (20),

Passport VARCHAR (20));

**DML COMMANDS:**

INSERT INTO initial\_user (UserID, Name, Email, PhoneNo, Address, AppliedUniversity, I20, Passport,EducationDocs,GRE,IELTS,VisaDoc) VALUES

(100, 'Declan Pollard', '[consequat@hotmail.com](mailto:consequat@hotmail.com)', '(223) 289-4739', '1038 Tempus Avenue', 'Georgia Tech', 'EA744I', 'KJ60738',"Pharma",323, 6.5, "AP54837"),

(110, 'Rana Wilkerson', '[leo.in.lobortis@google.net](mailto:leo.in.lobortis@google.net)', '(310) 937-8915', 'Ap #334-823 Rutrum, Ave', 'CMU', 'NI667H', 'CQ18748',"Pharma", 276,7.5, "VO57776"),

(120,"Melissa [Ramsey","lectus.justo@outlook.net","(864](mailto:Ramsey","lectus.justo@outlook.net","(864)) 169-4313","P.O. Box 364, 8499 Lorem, Avenue","UT Dallas","MM455Y","IL33716","Business", 292,7.5,"VW28372"),

(130,"Katell [Robinson","ac.risus@google.com","(461](mailto:Robinson","ac.risus@google.com","(461)) 471-4204","355-5153 Ut Avenue","Yale","QL188L","FX67513", "Business",325,6.0,"WK07138")

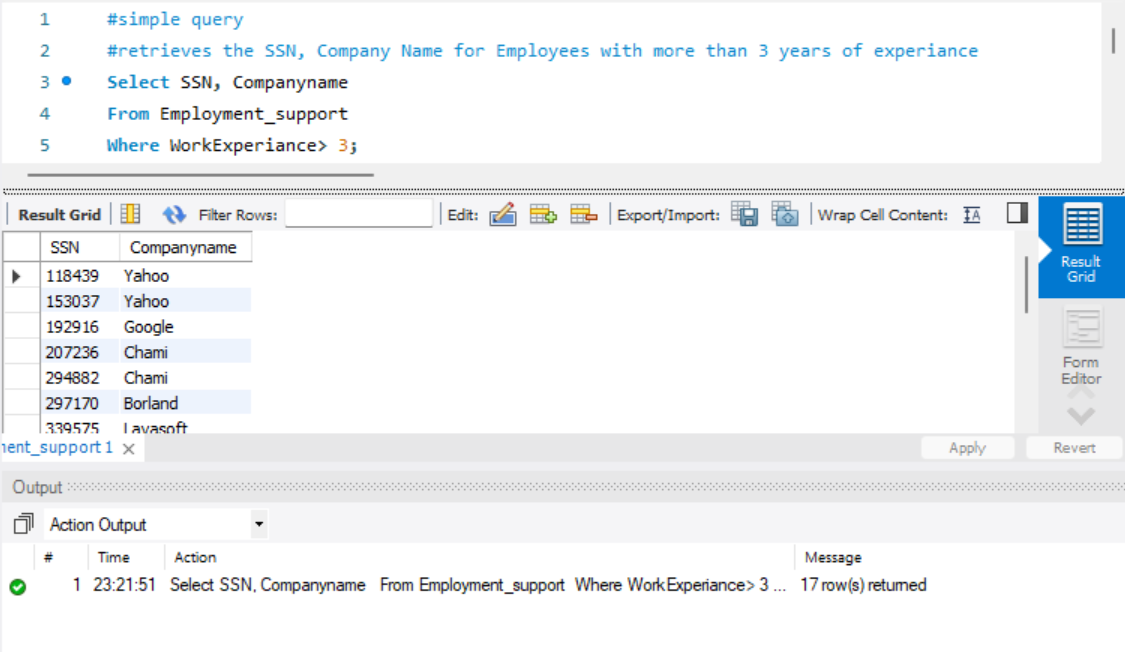
**DIFFERENT SQL QUERIES**

Simple query:

Select SSN, Companyname

From Employment\_support

Where WorkExperiance> 3

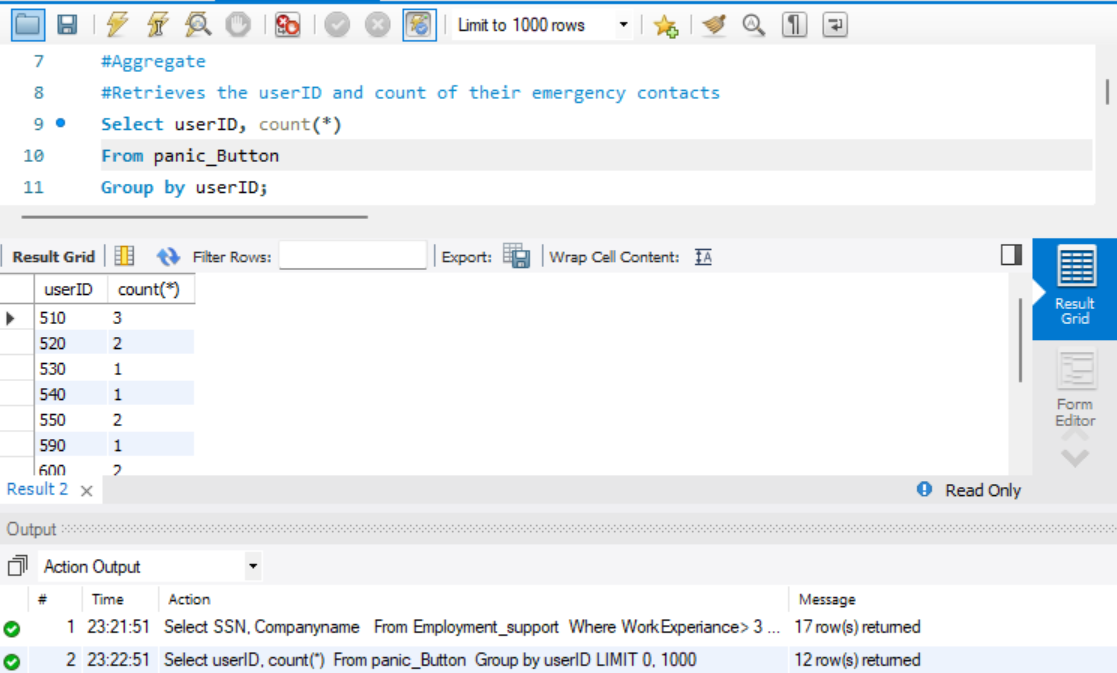


Aggregate query

Select userID, count(\*)

From panic\_Button

Group by userID

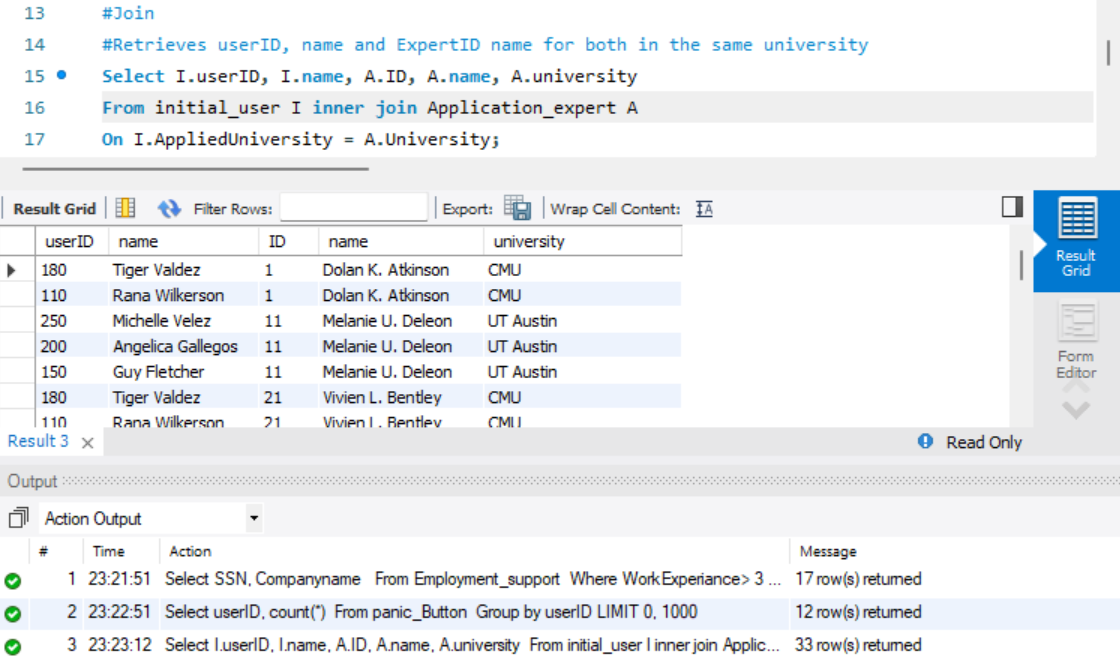


Join query

Select I.userID, I.name, A.ID, A.name, A.university

From intitial\_user I inner join Application\_expert A

On I.AppliedUniversity = A.University

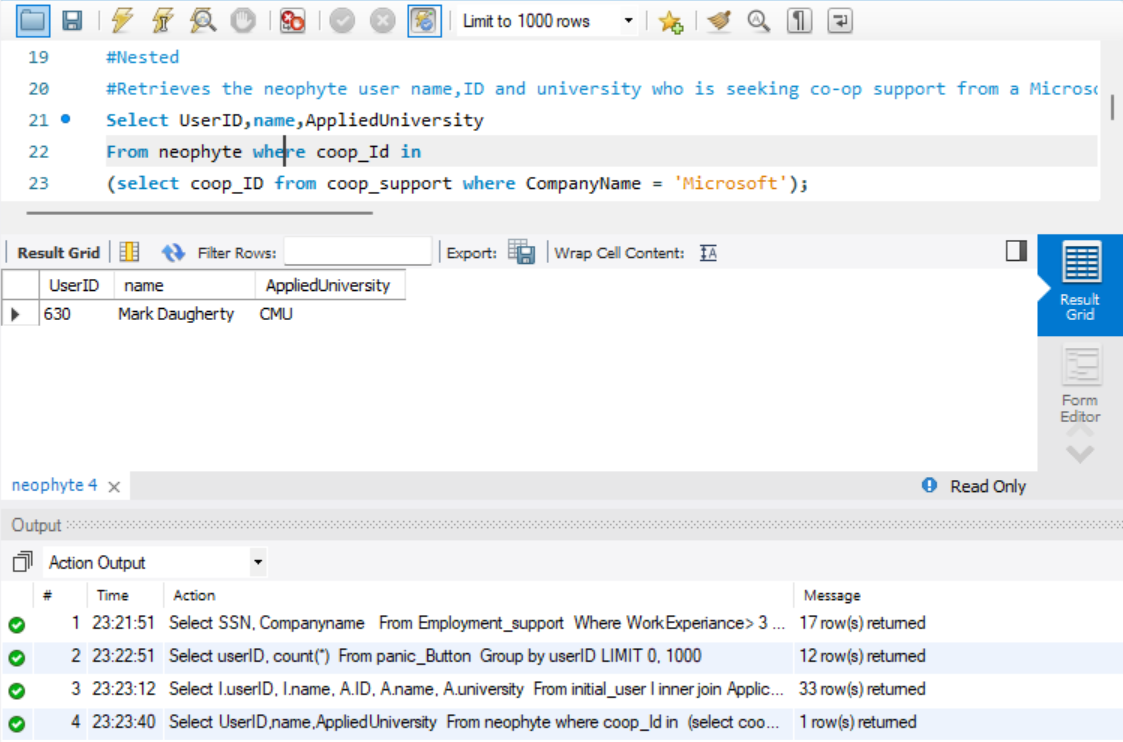


Nested Query

Select UserID,name AppliedUniversity

From neophytes where co-opId in

(select coopID from coop\_support where CompanyName = ‘Microsoft’)



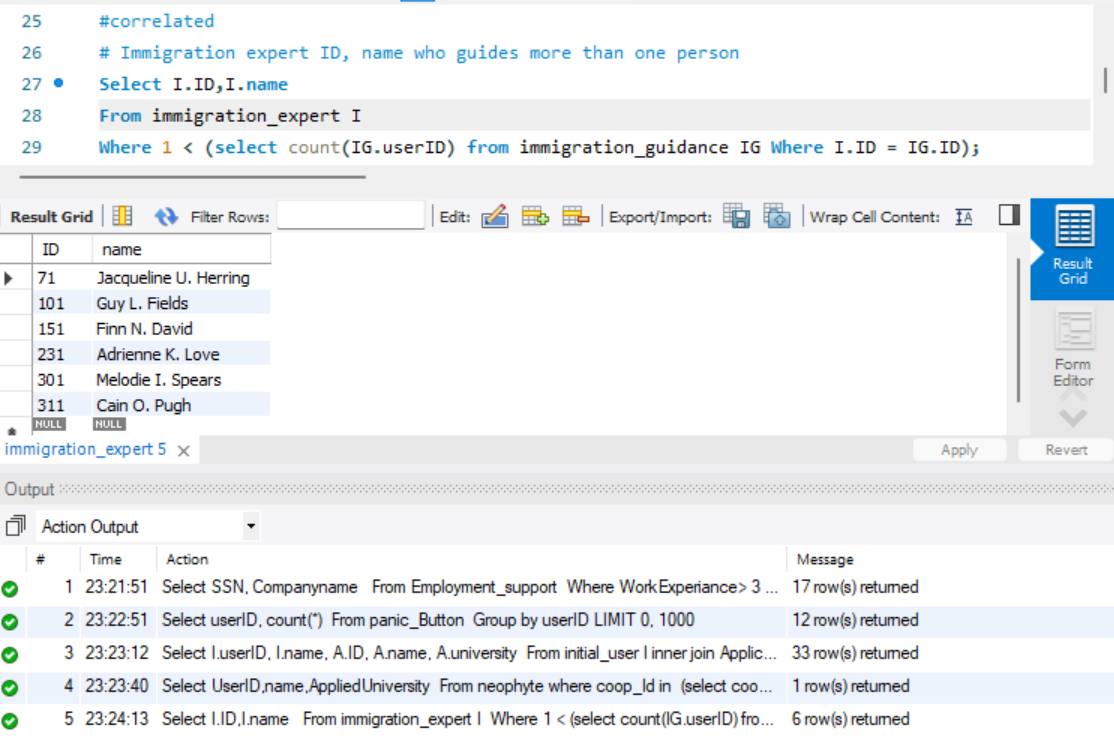
Correlated Query

Select I.ID,I.name

From immigration\_expert I

Where 1 < (select count (IG.userID) from immigration\_guidance IG

Where I.ID = IG.ID)



ALL/ANY/EXISTS

Select name

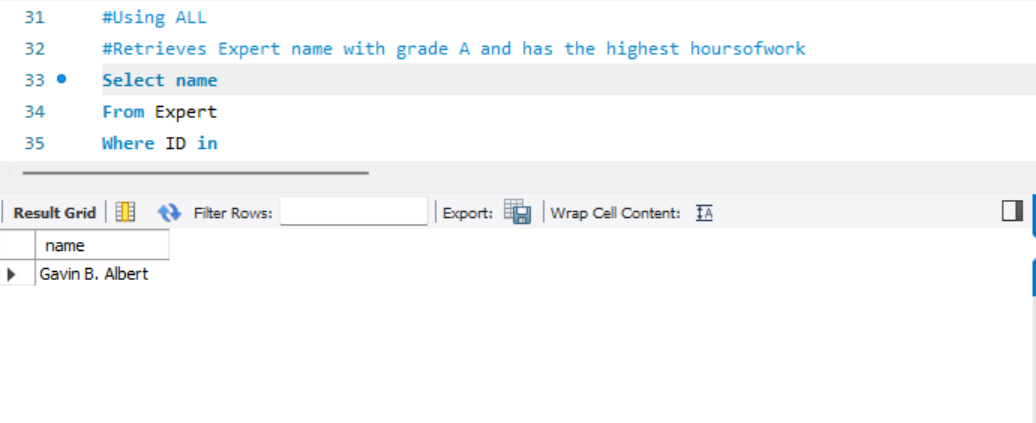
From Expert

Where ID in

(select ID from participation\_grade

where Grade = ‘A’ and hoursofwork > = ALL

(select hoursofwork from participation-grade where grade =’A’))



SET OPERATIONS

Select UserID, Name

From neophytes

Where university = ‘Georgia Tech’

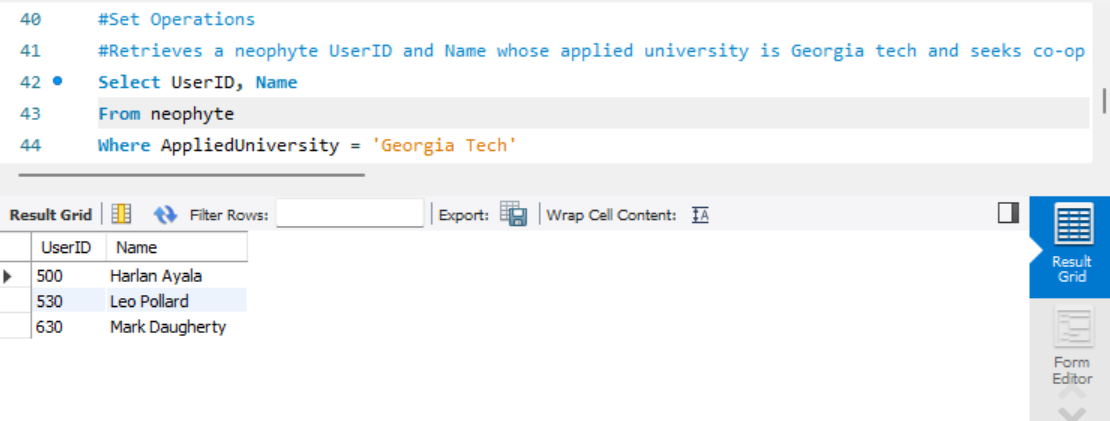
UNOIN

Select N1.UserID,N1.Name

FROM Neophytes N1, Coopsupport C

Where N1.coopID = C.coopID

And C.companyName = ‘Microsoft’



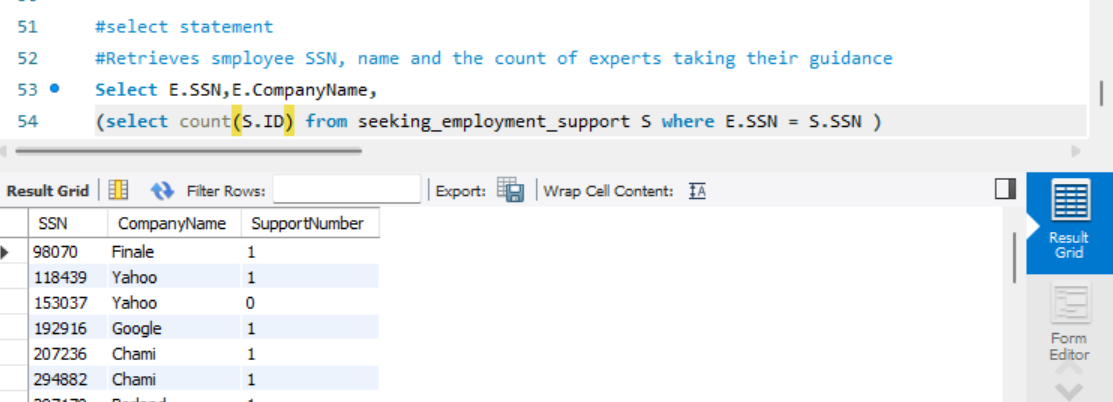
Subqueries in SELECT and FROM

Select E.SSN,E.name,

(select count(S.ID) from seeking\_employment\_support S where E.SSN = S.SSN )

as SupportNumber

From expert E



**IMPLEMENTATION IN PYTHON**

Python code to connect to the MySQL database.

import mysql.connector

from mysql.connector import Error

try:

connection = mysql.connector.connect(host='localhost',

database='supportsphere',

user='root',

password='Kavya@158')

if connection.is\_connected():

db\_Info = connection.get\_server\_info()

print("Connected to MySQL Server version ", db\_Info)

cursor = connection.cursor()

cursor.execute("select database();")

record = cursor.fetchone()

print("Your connected to database: ", record)

#QUERY 1

sql\_select\_Query = "select name from expert where University = 'CMU'"

cursor = connection.cursor()

cursor.execute(sql\_select\_Query)

records = cursor.fetchall()

print("Experts from CMU:\n")

for row in records:

print('Expert =',row[0],"\n")

#QUERY2

sql\_select\_Query = "Select SSN, Companyname From Employment\_support Where WorkExperiance> 3;"

cursor = connection.cursor()

cursor.execute(sql\_select\_Query)

records = cursor.fetchall()

print("Employees with more than 3 years experiance:\n")

for row in records:

print('Employees =',row[0],"\n")

#QUERY 3

sql\_select\_Query = "Select I.ID,I.name From immigration\_expert I Where 1 < (select count(IG.userID) from immigration\_guidance IG Where I.ID = IG.ID);"

cursor = connection.cursor()

cursor.execute(sql\_select\_Query)

records = cursor.fetchall()

print("Immigration experts who guide more than one user:\n")

for row in records:

print('Experts =',row[0],"\n")

#EXCEPT BLOCK

except Error as e:

print("Error while connecting to MySQL", e)

finally:

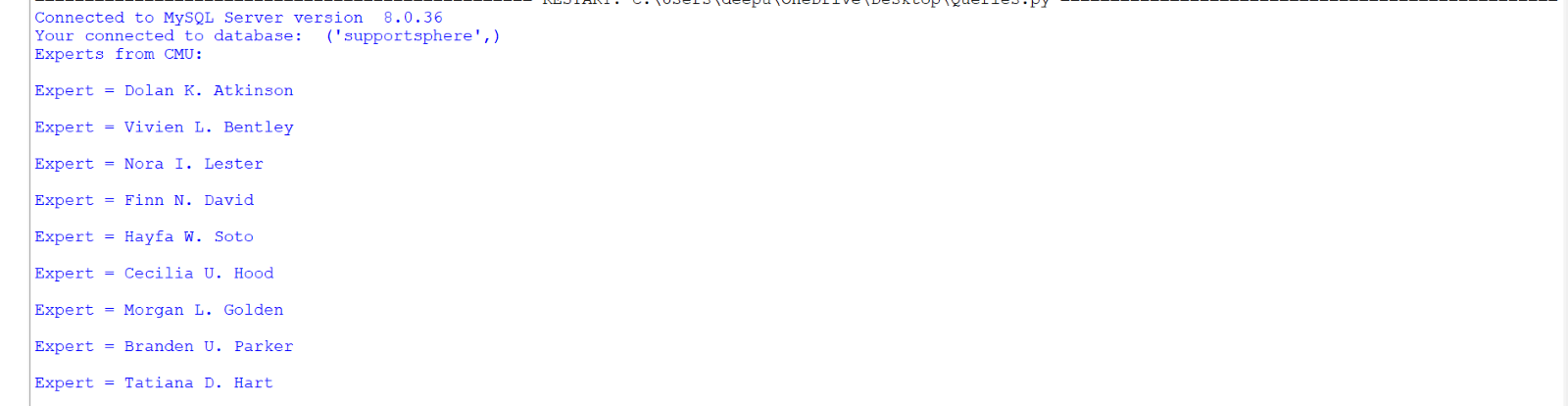
if (connection.is\_connected()):

cursor.close()

connection.close()

print("MySQL connection is closed")

**OUTPUT:**







**PLOTTING GRAPHS USING PYTHON**

import matplotlib.pyplot as plt

import mysql.connector

from mysql.connector import Error

try:

connection = mysql.connector.connect(host='localhost',

database='supportsphere',

user='root',

password='Kavya@158')

if connection.is\_connected():

db\_Info = connection.get\_server\_info()

print("Connected to MySQL Server version ", db\_Info)

cursor = connection.cursor()

cursor.execute("select database();")

record = cursor.fetchone()

print("Your connected to database: ", record)

# Fetch data from the database

sql\_select\_Query = "Select E.SSN,(select count(S.ID) from seeking\_employment\_support S where E.SSN = S.SSN ) as SupportNumber From employment\_support E;"

cursor = connection.cursor()

cursor.execute(sql\_select\_Query)

records = cursor.fetchall()

SSN = [row[0] for row in records]

Count = [row[1] for row in records]

# Plot the data

plt.plot(SSN, Count)

plt.xlabel('SSN')

plt.ylabel('Count')

plt.title('Employees over experts seeking support')

plt.xticks(rotation=45)

plt.tight\_layout()

# Show the plot

plt.show()

# Close the connection

except Error as e:

print("Error while connecting to MySQL", e)

finally:

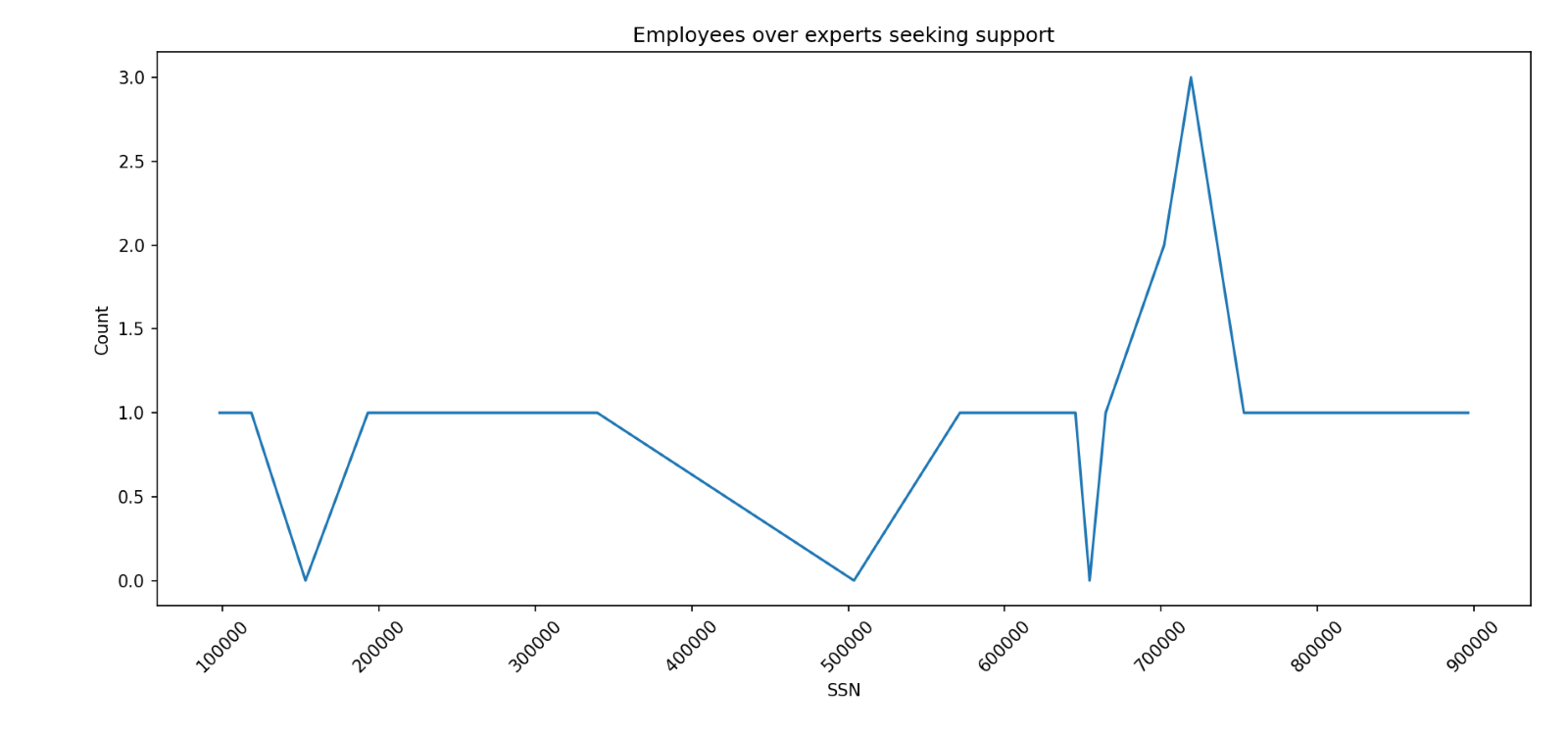
if (connection.is\_connected()):

cursor.close()

connection.close()

print("MySQL connection is closed")

**OUTPUT:**



**IMPLEMENTATION IN NOSQL**

The code to connect to MongoDB and export MySQL into the MongoDB:

import pymysql

import pymongo

import mysql.connector

# Connect to MySQL

connection = mysql.connector.connect(host='localhost',database='supportsphere',user='root',password='Kavya@158')

mysql\_cursor = connection.cursor()

# Connect to MongoDB

client = pymongo.MongoClient("mongodb://localhost:27017")

mongo\_db = client["SupportSphere"]

mongo\_collection = mongo\_db["User"]

# Query data from MySQL and insert into MongoDB

mysql\_cursor.execute("SELECT UserID,Name,AppliedUniversity FROM user")

rows = mysql\_cursor.fetchall()

# Insert data into MongoDB

for row in rows:

doc = {"UserID": row[0], "Name": row[1],"AppliedUniversity":row[2]}

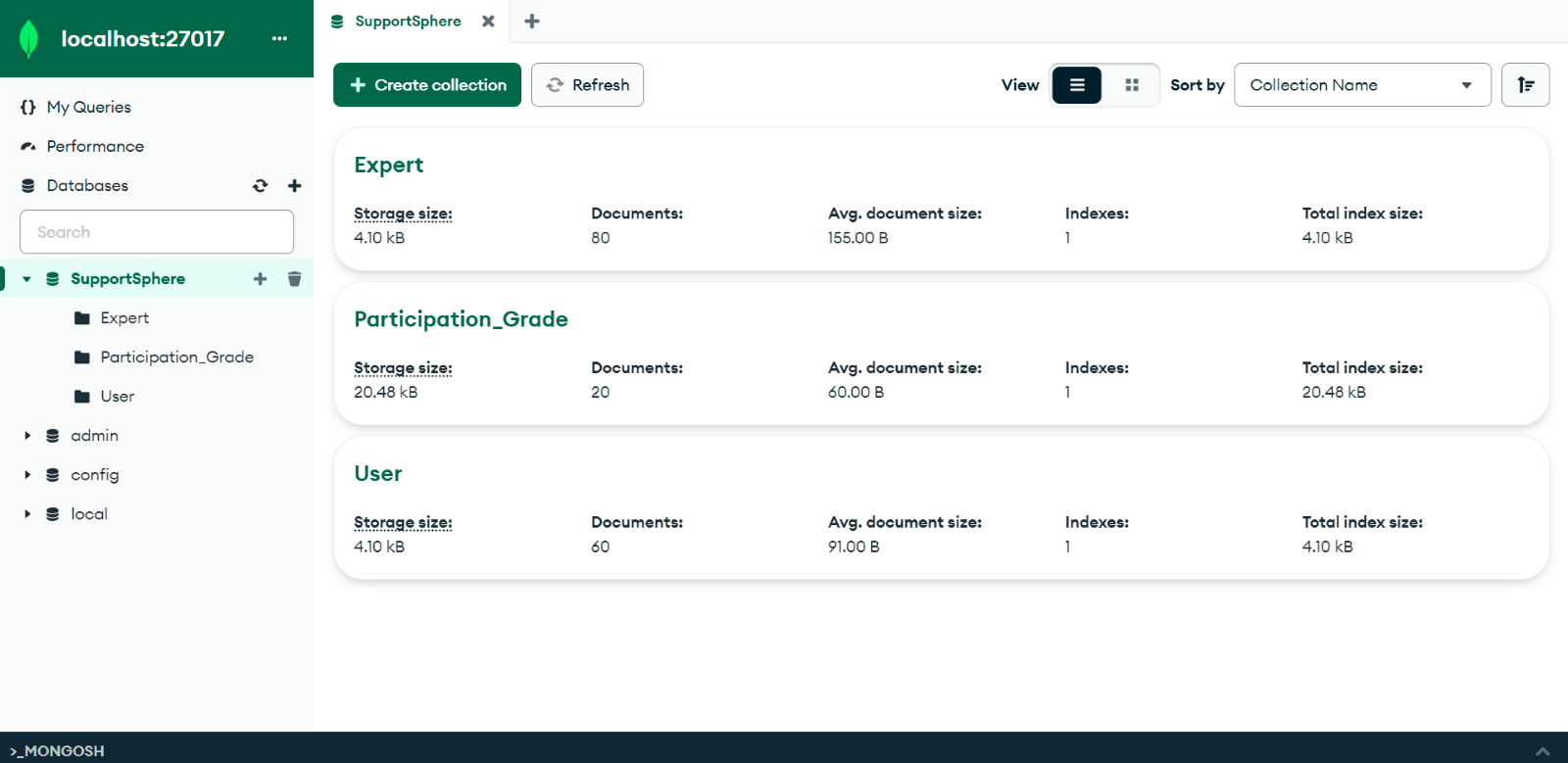
# Create a document from MySQL row

mongo\_collection.insert\_one(doc)

# Close connections

connection.close()

client.close()



**MONGODB QUERIES**

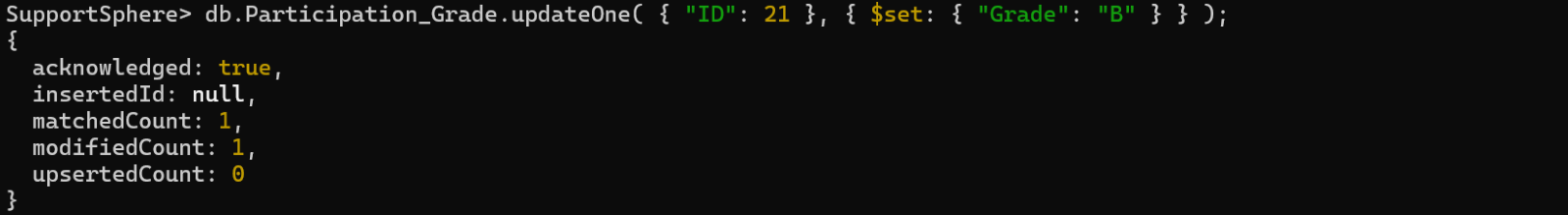
db.Participation\_grade.update(

{"ID":21},

{$set:

{"Grade":"B"}}

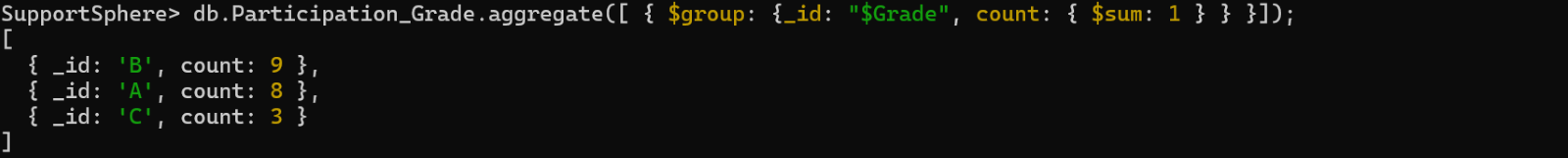
);



db.Participation\_Grade.aggregate(

[ { $group:

{\_id: "$Grade", count: { $sum: 1 } } }]);



var mapFunction = function() {

emit(this.AppliedUniversity, 1);

};

var reduceFunction = function(key, values) {

return Array.sum(values);

};

db.User.mapReduce(

mapFunction,

reduceFunction,

{

out: { inline: 1 }

}

);

