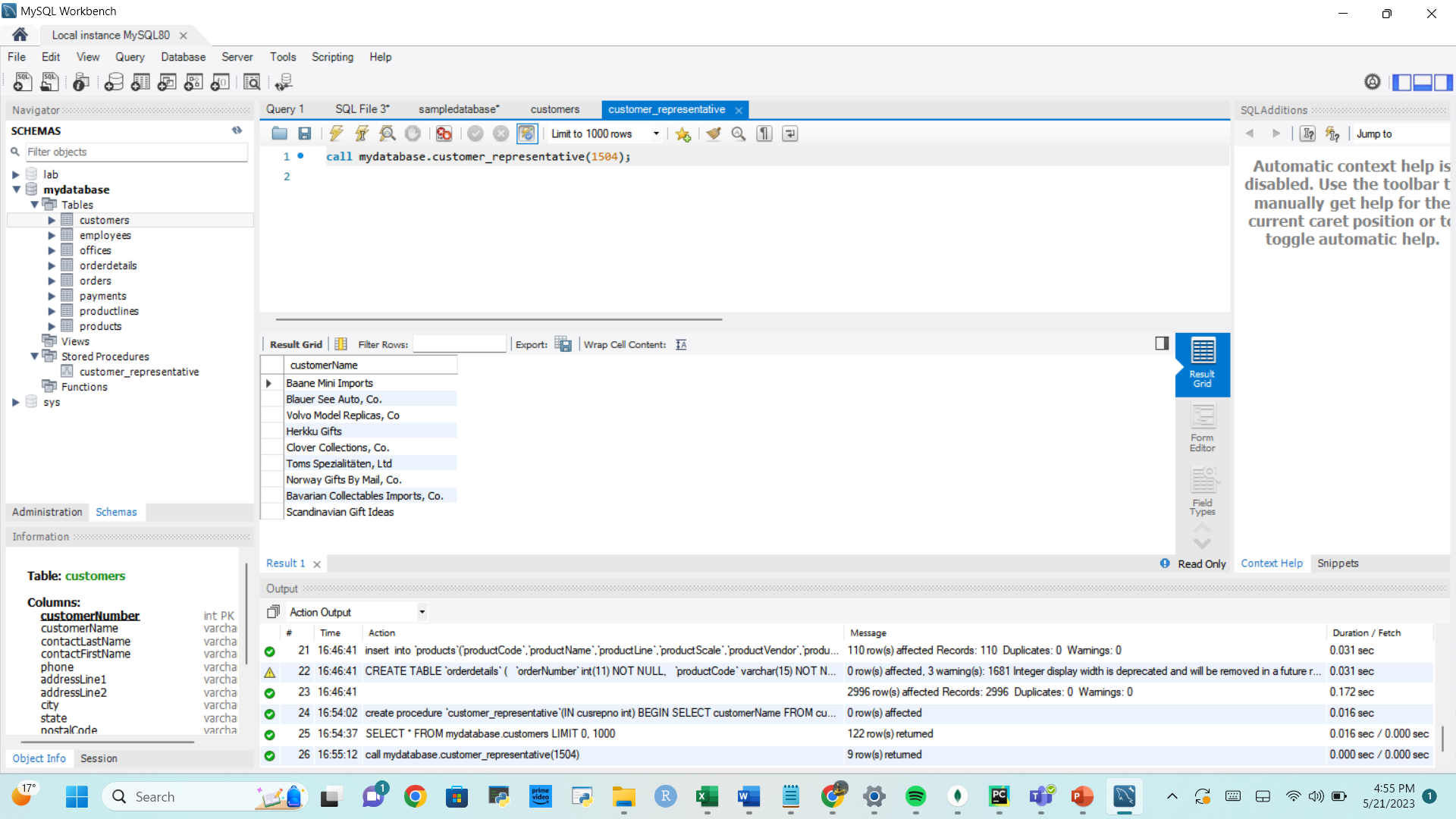
Task 1



Task 2

delimiter //

create procedure `product\_description`(IN pdt\_code varchar(15))

BEGIN

SELECT p.productCode, p.productName, p.productDescription, pl.textDescription as productLineDescription

FROM products p

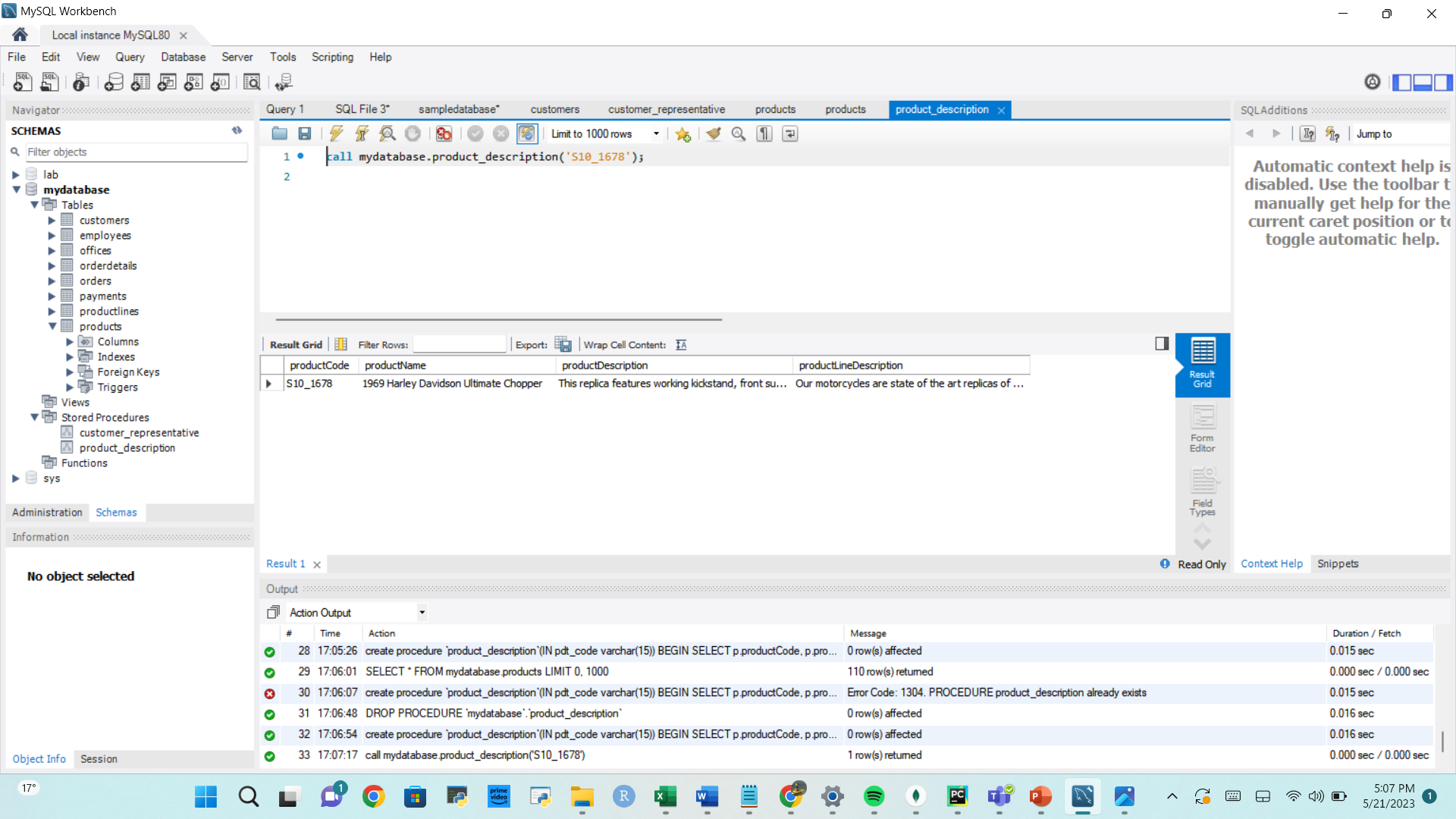
INNER JOIN productlines pl

ON p.productLine = pl.productLine

WHERE p.productCode = pdt\_code;

END//

delimiter;



Task 5

delimiter //

create procedure `highest\_three`()

BEGIN

SELECT productLine, AVG(quantityInStock) as averageStock

FROM products

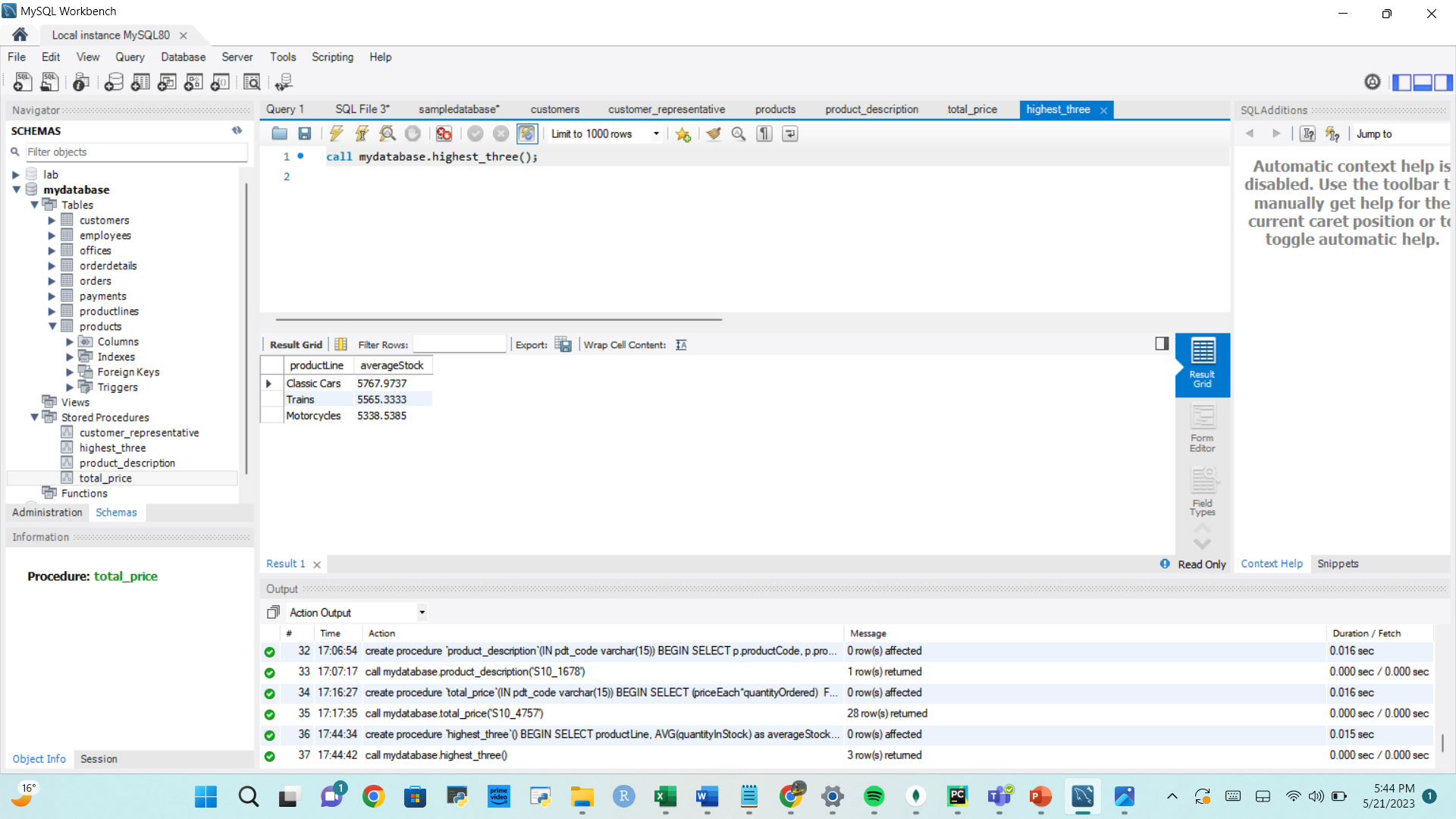
GROUP BY productLine

ORDER BY averageStock DESC

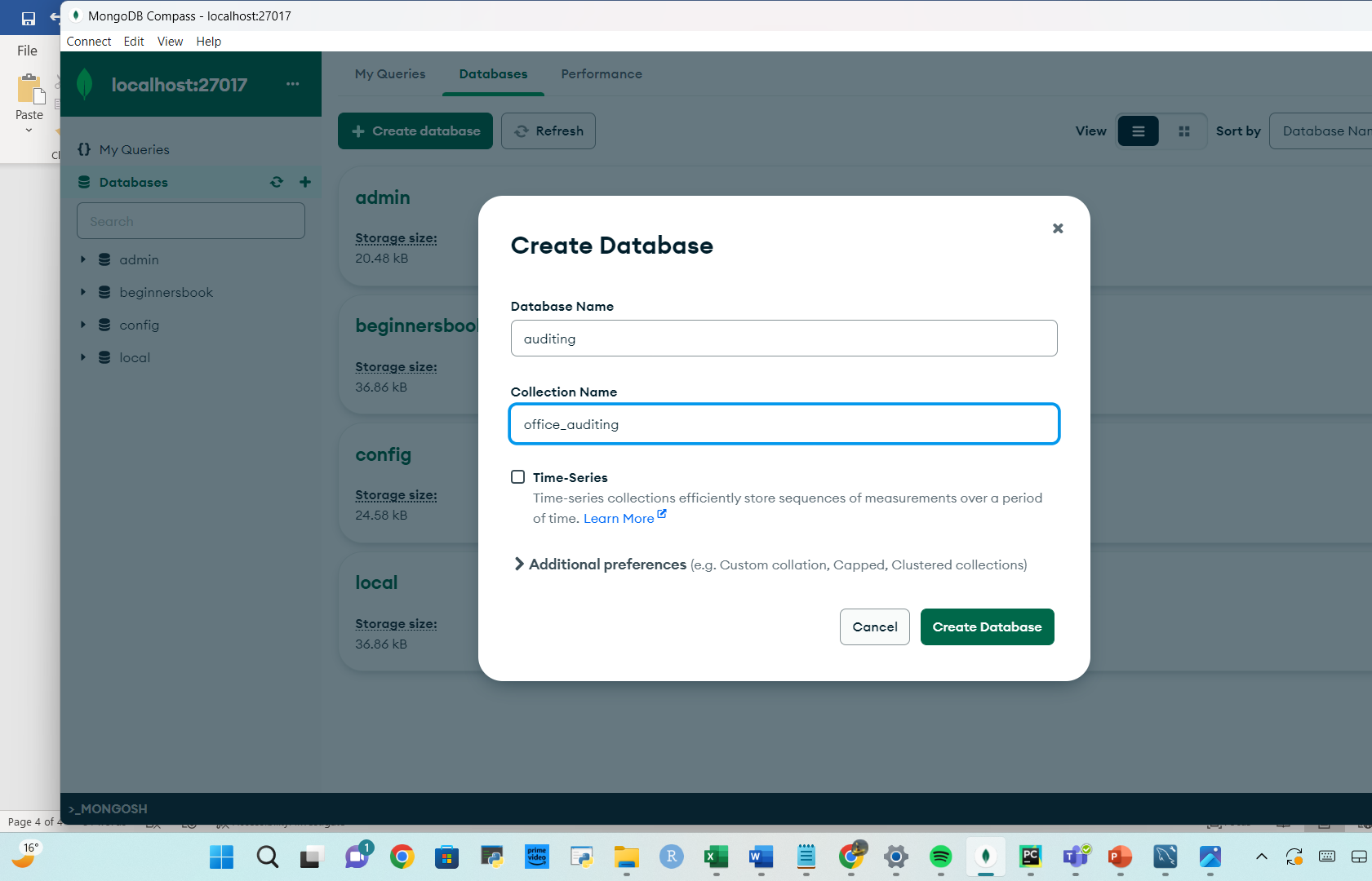
LIMIT 3;

END//

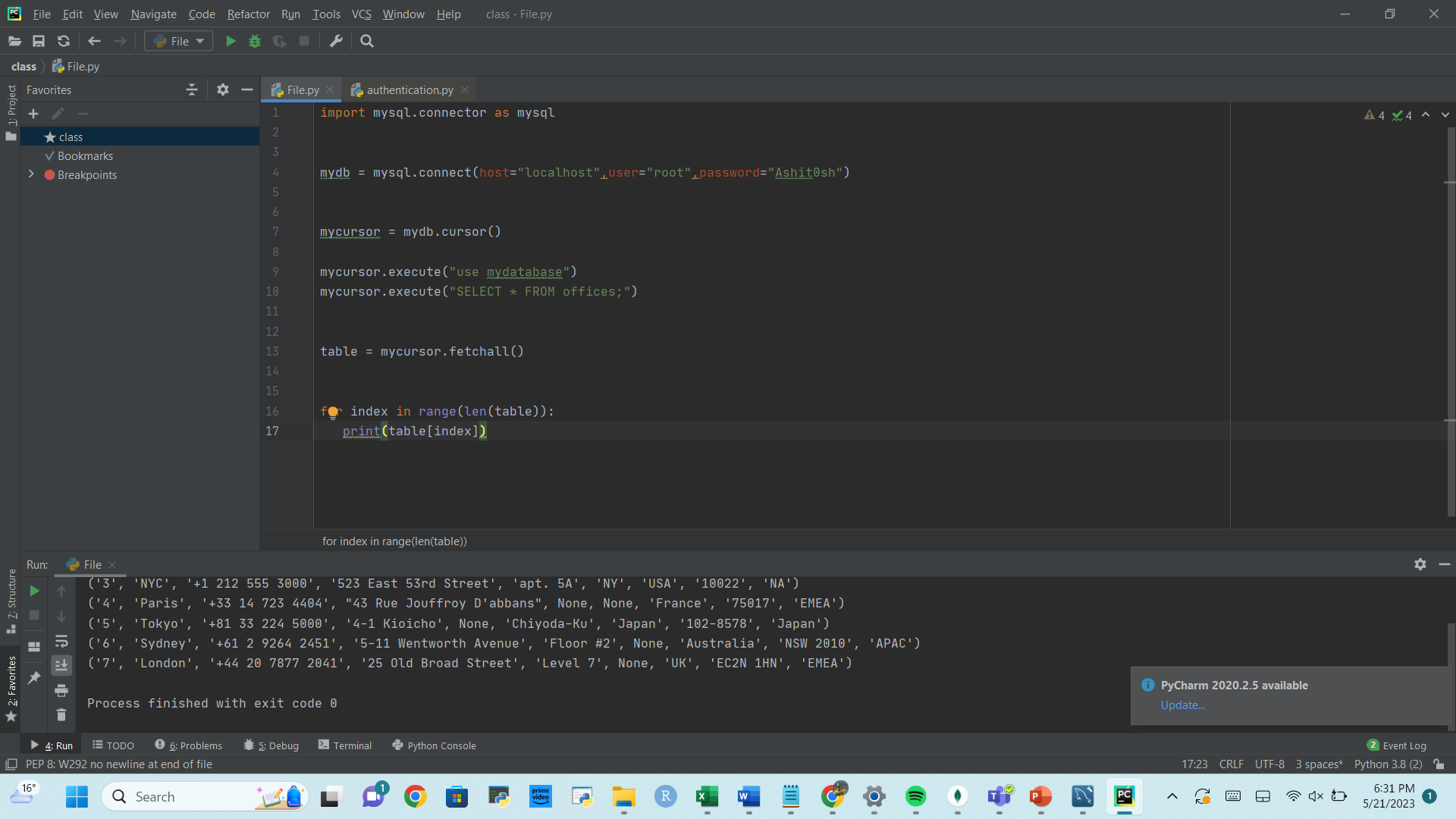
delimiter ;

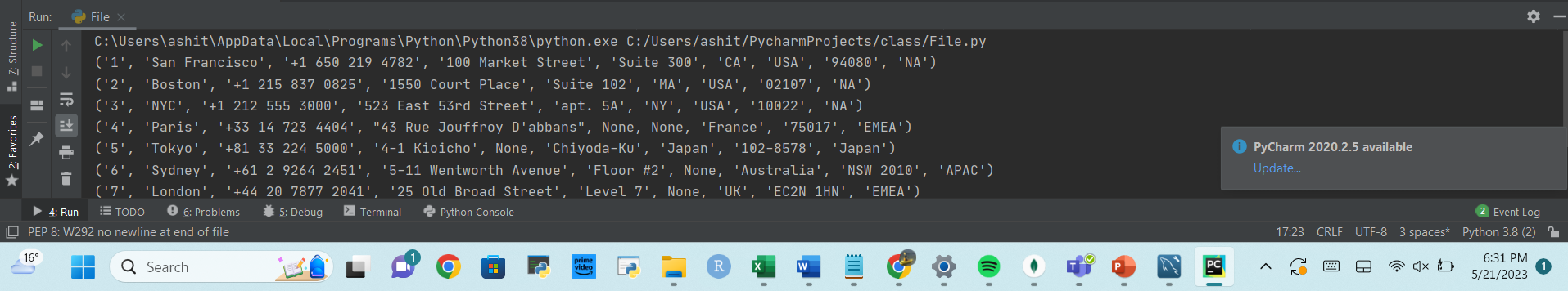


Task 6

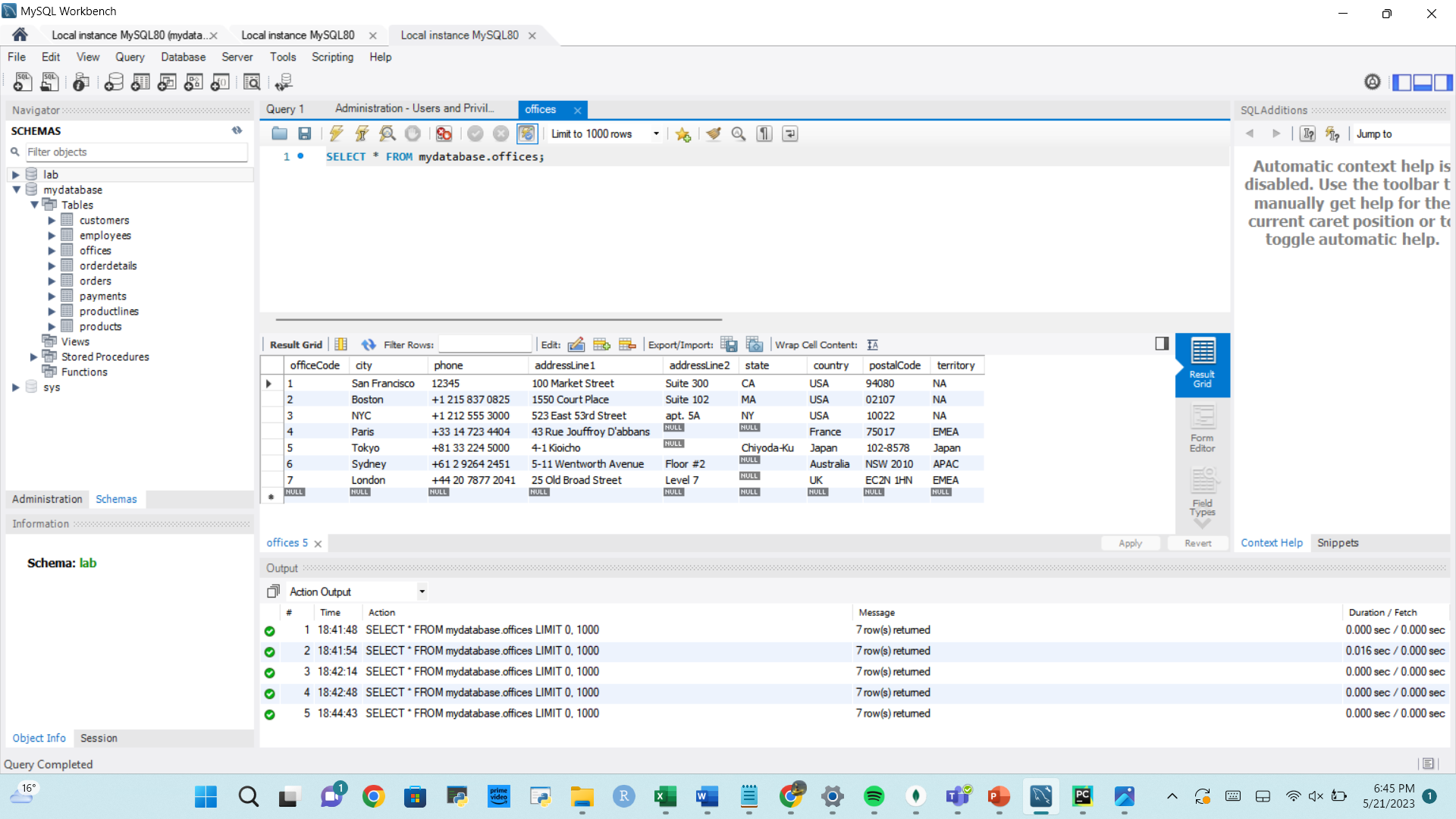


Task 7

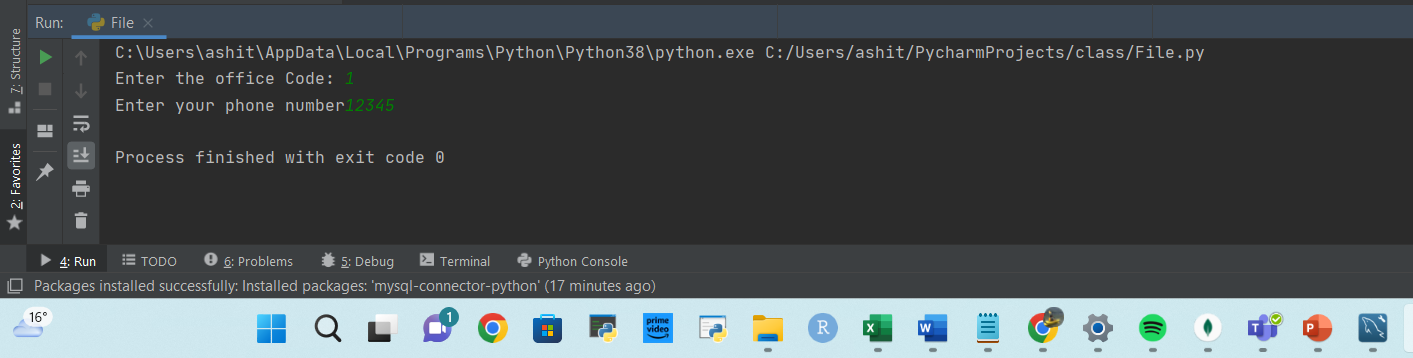




Task 8



import mysql.connector as mysql  
  
  
mydb = mysql.connect(host="localhost",  
 user="root",  
 password="Ashit0sh")  
  
  
mycursor = mydb.cursor()  
  
  
  
  
  
officeCode = ""  
  
  
while not officeCode in ["1", "2", "3", "4", "5", "6", "7"]:  
 officeCode = input("Enter the office Code: ")  
  
  
phonenumber = ""  
  
  
while not phonenumber.isnumeric():  
 phonenumber = input("Enter your phone number")  
  
mycursor.execute("use mydatabase")  
  
mycursor.execute("UPDATE offices SET phone = %s WHERE officeCode = %s", (phonenumber,officeCode))  
  
  
mydb.commit()



Task 9

use mydatabase;

create table office\_updates(

officeCode INT,

prevCity VARCHAR(255),

newCity VARCHAR(255),

updateTime timestamp

);

delimiter //

create trigger office\_update\_before

before update

on offices

for each row

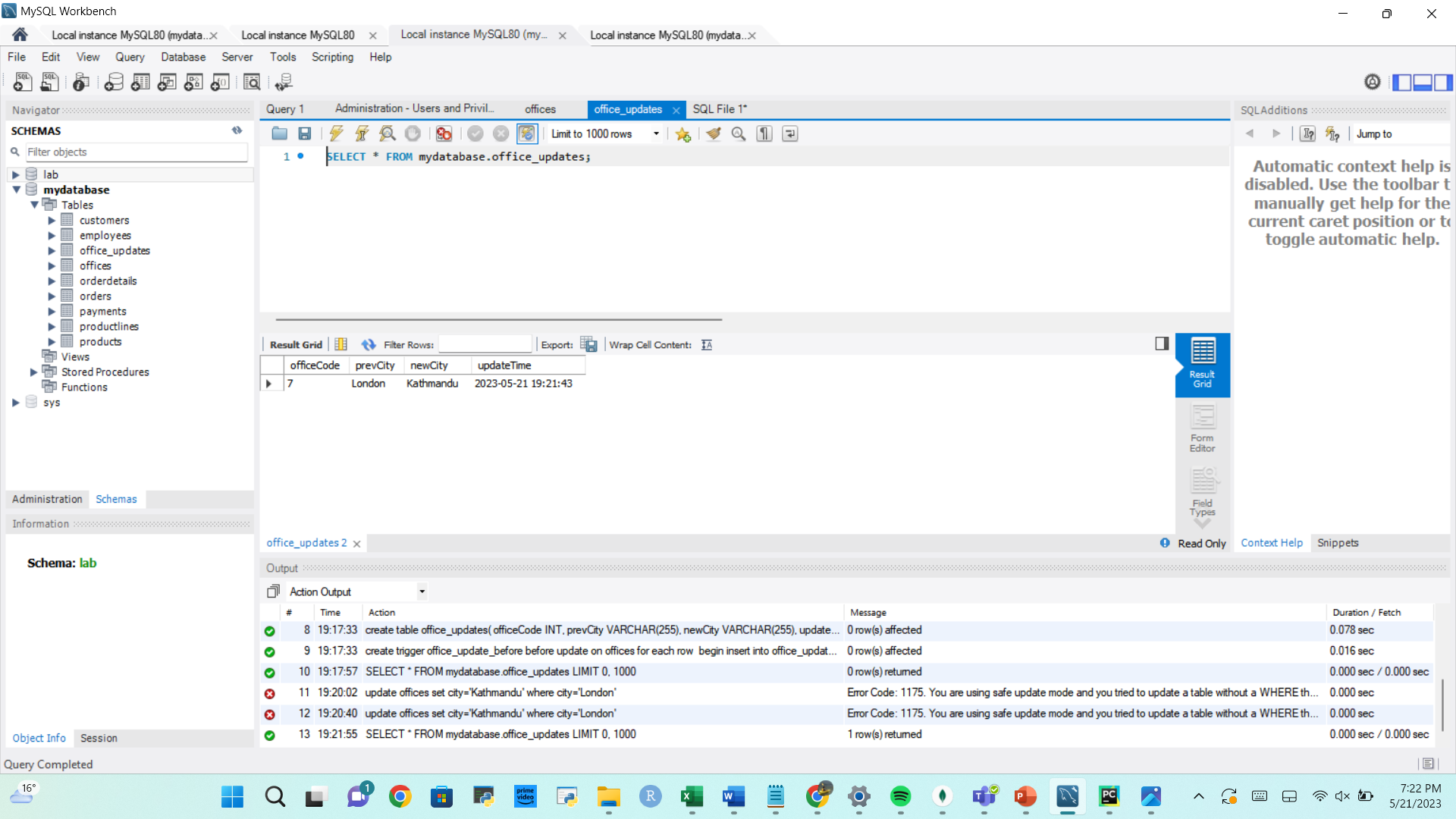
begin

insert into office\_updates(officeCode,prevCity,newCity,updateTime)

values (new.officeCode, old.city, new.city, now());

end //

delimiter ;

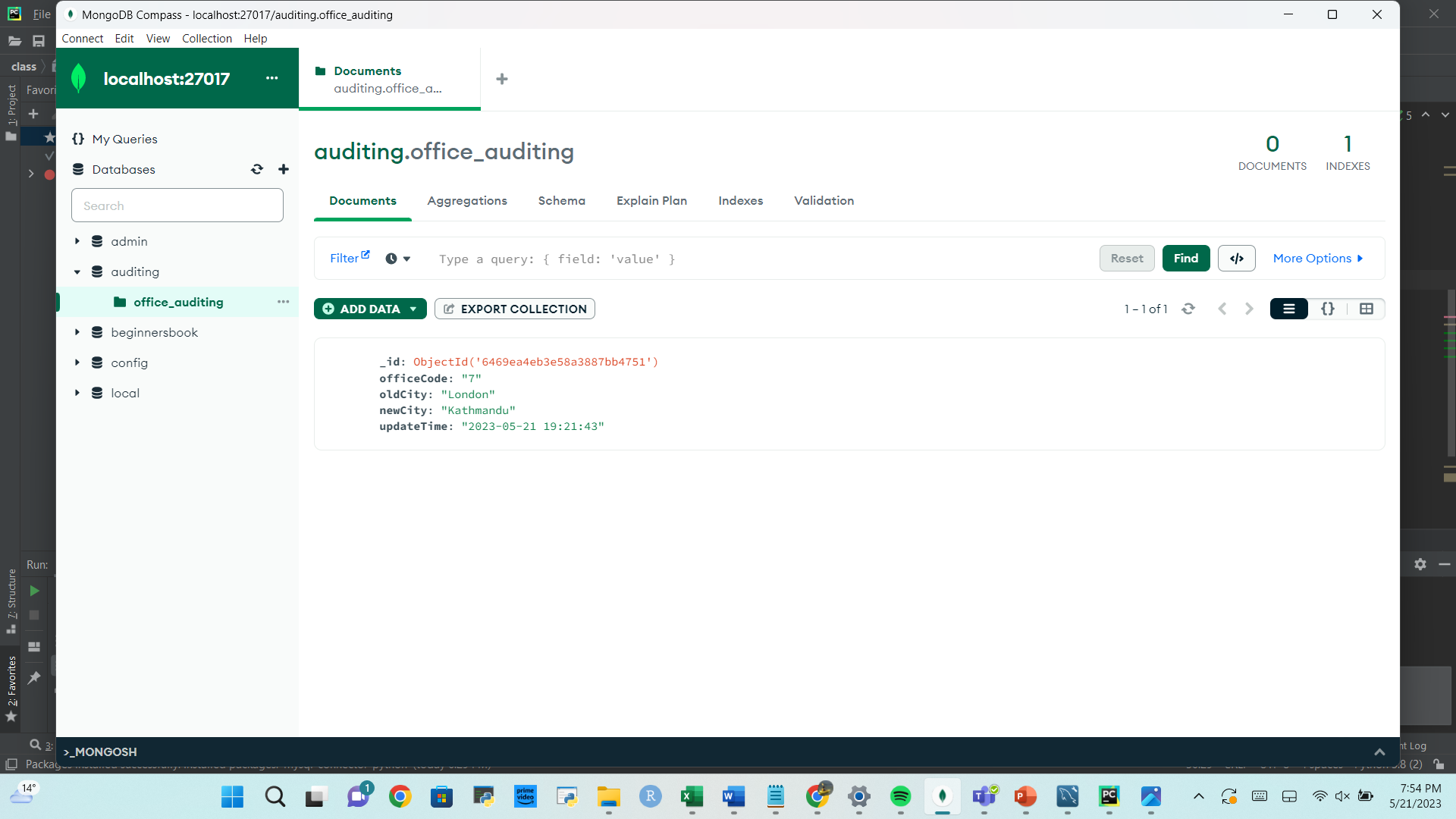


update offices

set city='Kathmandu' where city='London'

Task 10

import mysql.connector as mysql  
import pymongo  
  
  
mydb = mysql.connect(host="localhost",  
 user="root",  
 password="Ashit0sh")  
  
  
mycursor = mydb.cursor()  
  
  
mycursor.execute('USE mydatabase')  
  
  
mycursor.execute("SELECT \* FROM office\_updates")  
  
  
updates = mycursor.fetchall()  
  
  
mongo\_updates = []  
  
  
for row in updates:  
 mongo\_updates.append(  
 {"officeCode":f"{row[0]}",  
 "oldCity": f"{row[1]}",  
 "newCity": f"{row[2]}",  
 "updateTime": f"{row[3]}"}  
 )  
  
  
myclient = pymongo.MongoClient("mongodb://localhost:27017")  
  
  
auditing = myclient['auditing']  
  
  
office\_auditing = auditing['office\_auditing']  
  
  
for doc in mongo\_updates:  
 office\_auditing.insert\_one(doc)



Task 3

delimiter //

create function total\_price(product\_code varchar(15))

returns float

deterministic

begin

declare total float;

select SUM(quantityOrdered \* priceEach) into total

from orderdetails

where productCode = product\_code;

return total;

end //

delimiter ;

