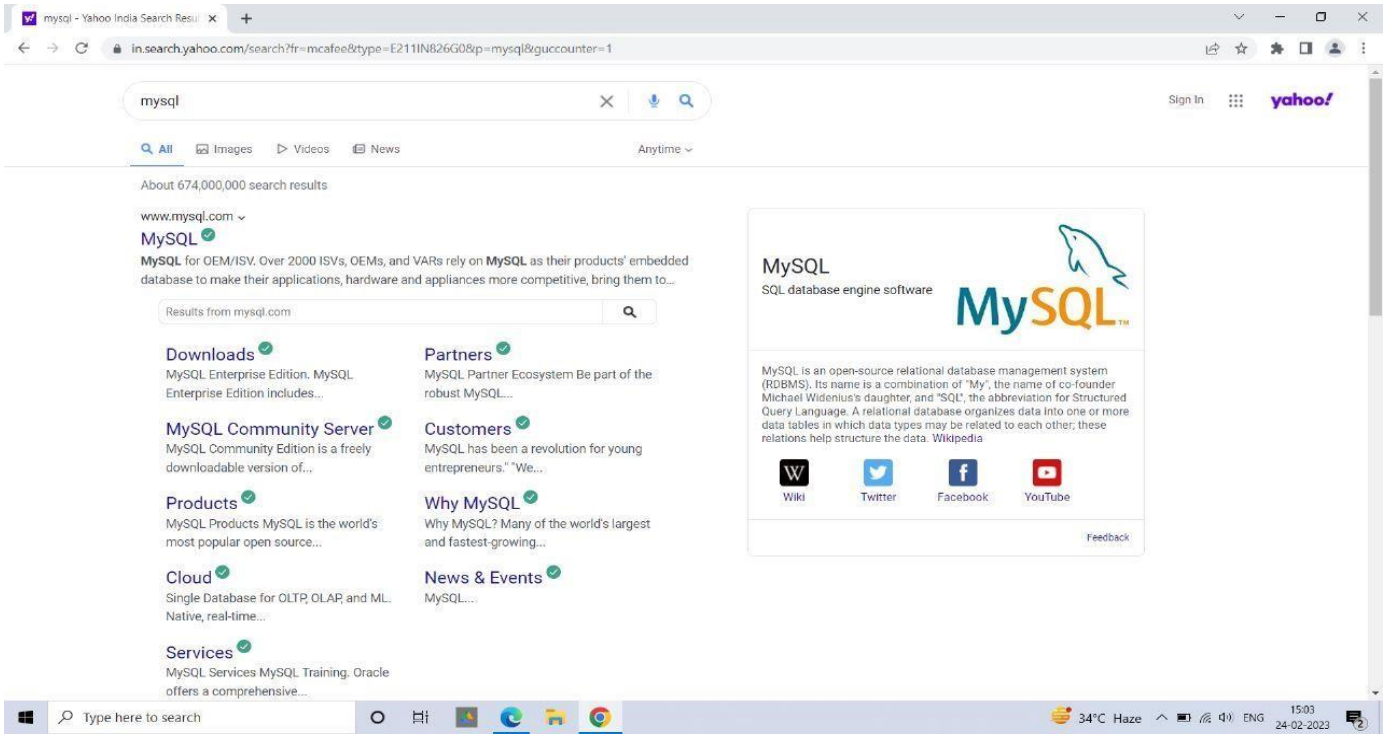
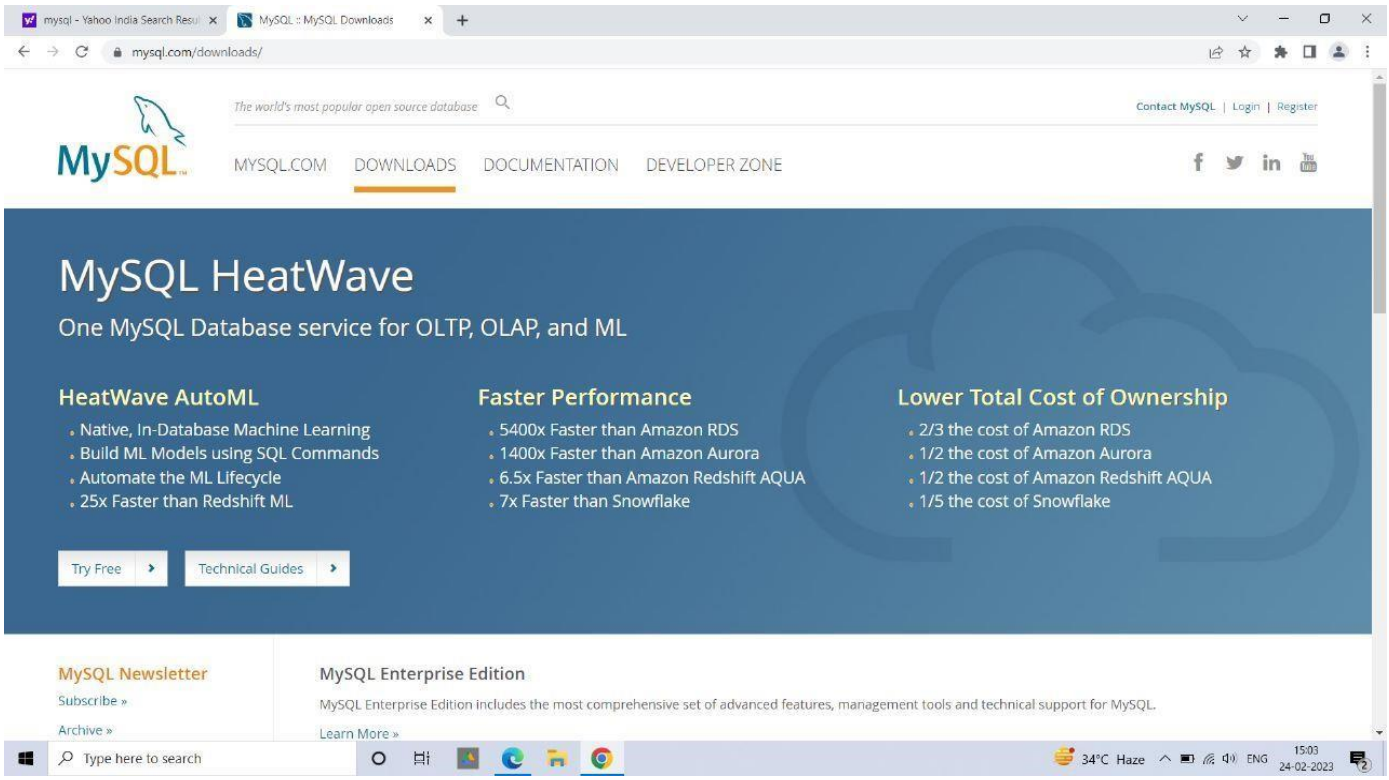
Group A: Study of Databases

2. Install and configure client and server of MySQL. (Show all commands and necessary steps for installation and configuration)

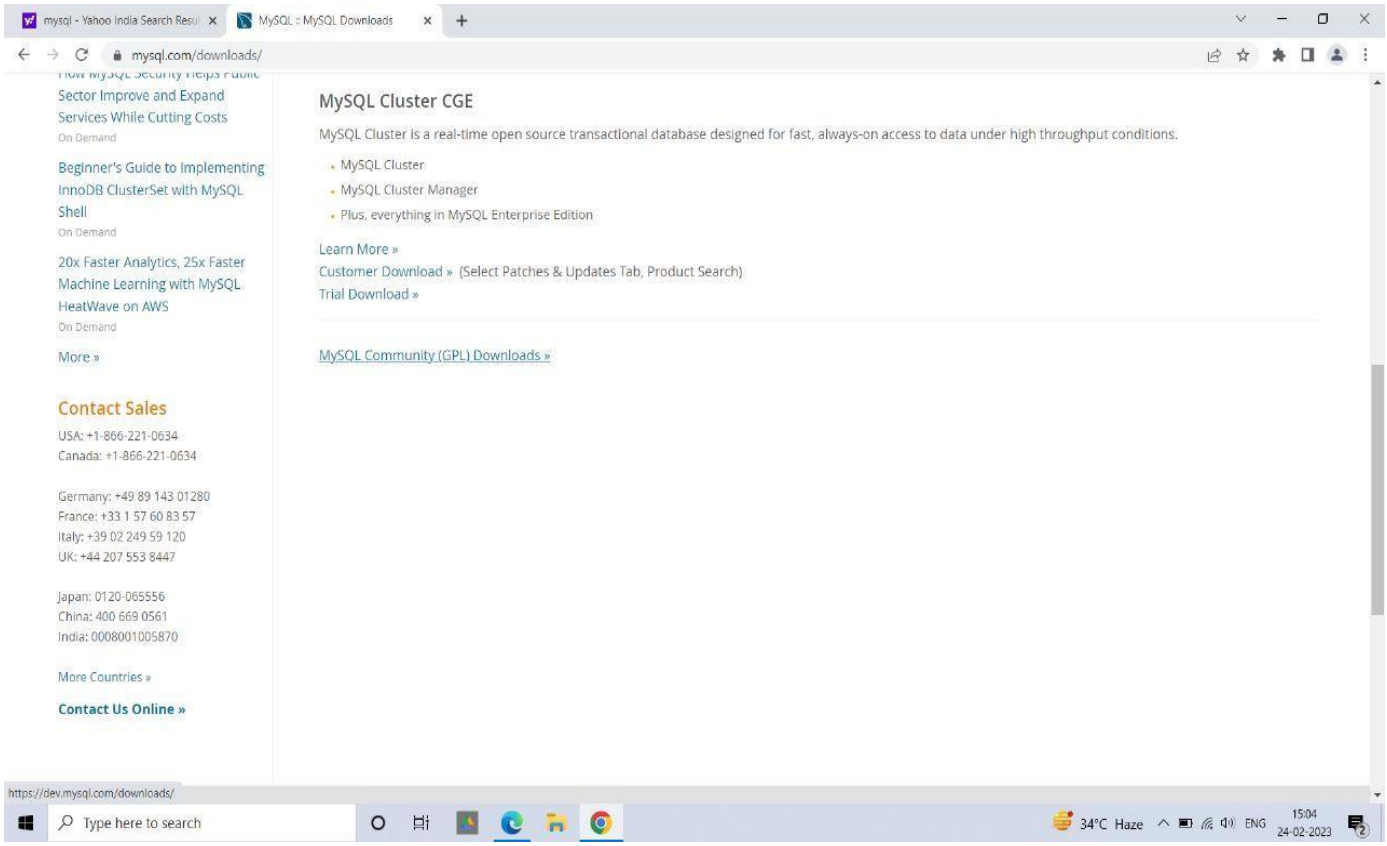
STEP 1:



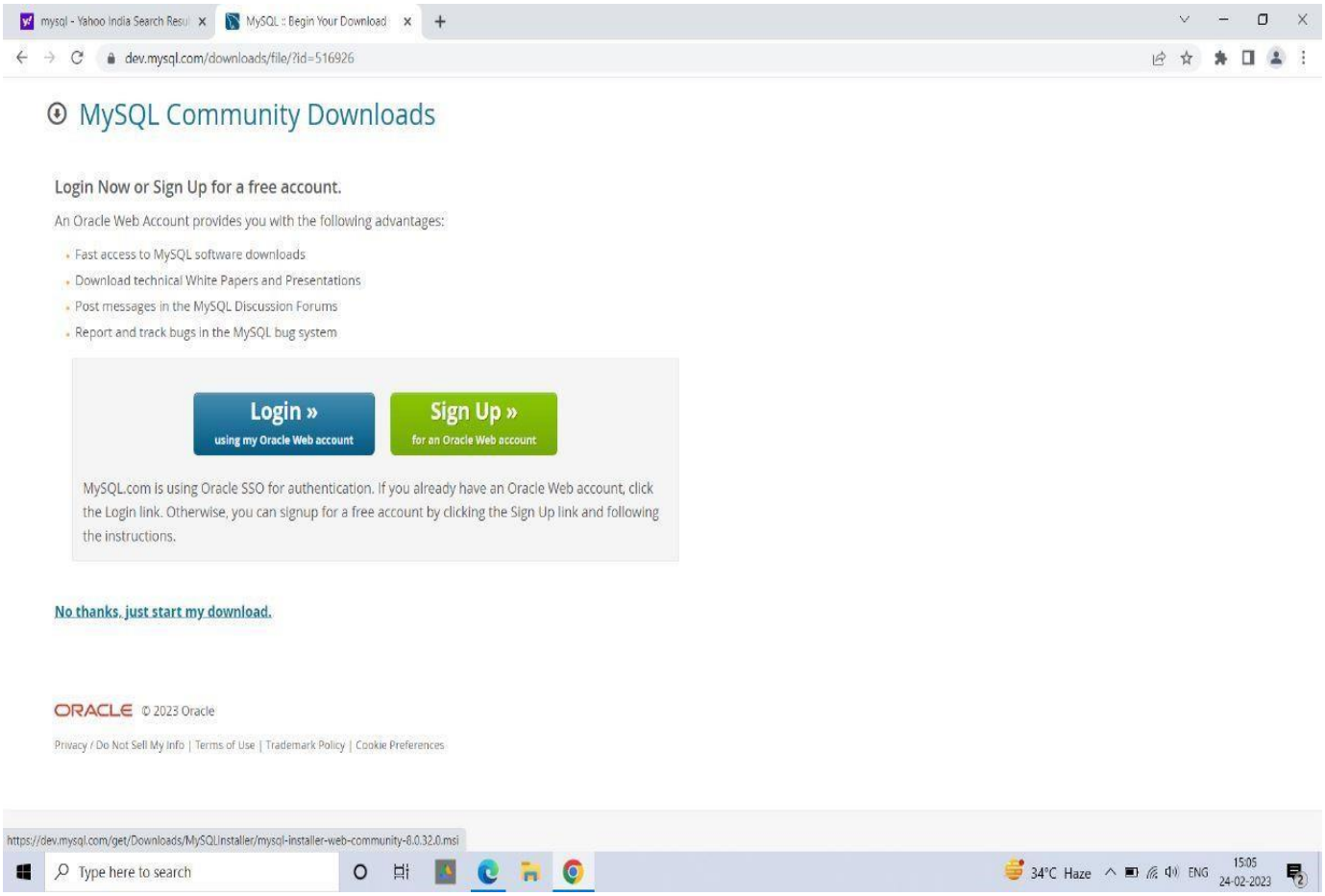
STEP 2:



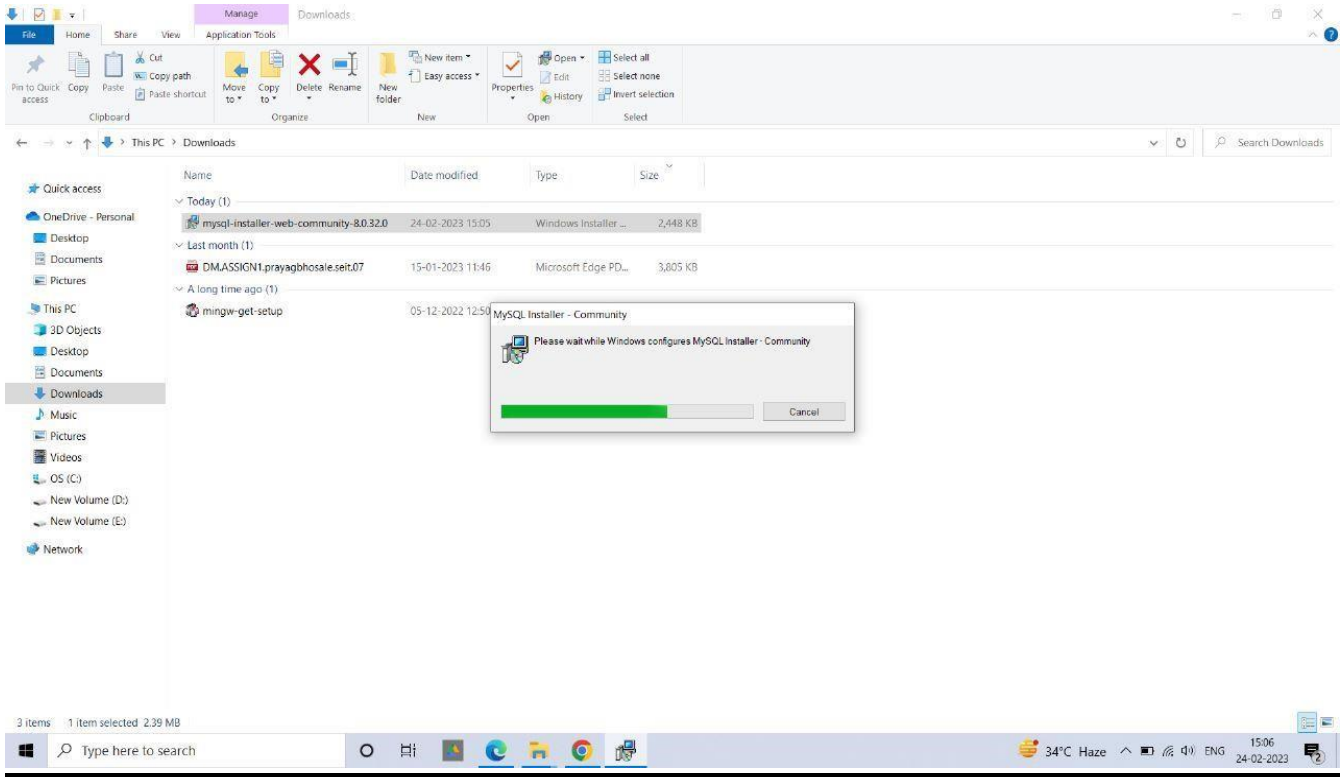
STEP 3:


mysql - Yahoo India Search Resul
MySQL :: MySQL Downloads
mysql.com/downloads/
Sector Improve and Expand Services While Cutting Costs
On Demand
Beginner's Guide to Implementing InnoDB ClusterSet with MySQL Shell
On Demand
20x Faster Analytics, 25x Faster Machine Learning with MySQL HeatWave on AWS
On Demand
More »
Contact Sales
USA: +1-866-221-0634
Canada: +1-866-221-0634
Germany: +49 89 143 01280
France: +33 1 57 60 83 57
Italy: +39 02 249 59 120
UK: +44 207 553 8447
Japan: 0120-065556
China: 400 669 0561
India: 0008001005870
More Countries »
Contact Us Online »
MySQL Cluster CGE
MySQL Cluster is a real-time open source transactional database designed for fast, always-on access to data under high throughput conditions.
MySQL Cluster
MySQL Cluster Manager
Plus, everything in MySQL Enterprise Edition
Learn More »
Customer Download » (Select Patches & Updates Tab, Product Search)
Trial Download »
MySQL Community (GPL) Downloads »
https://dev.mysql.com/downloads/
Type here to search
34°C Haze
ENG
15:04
24-02-2023

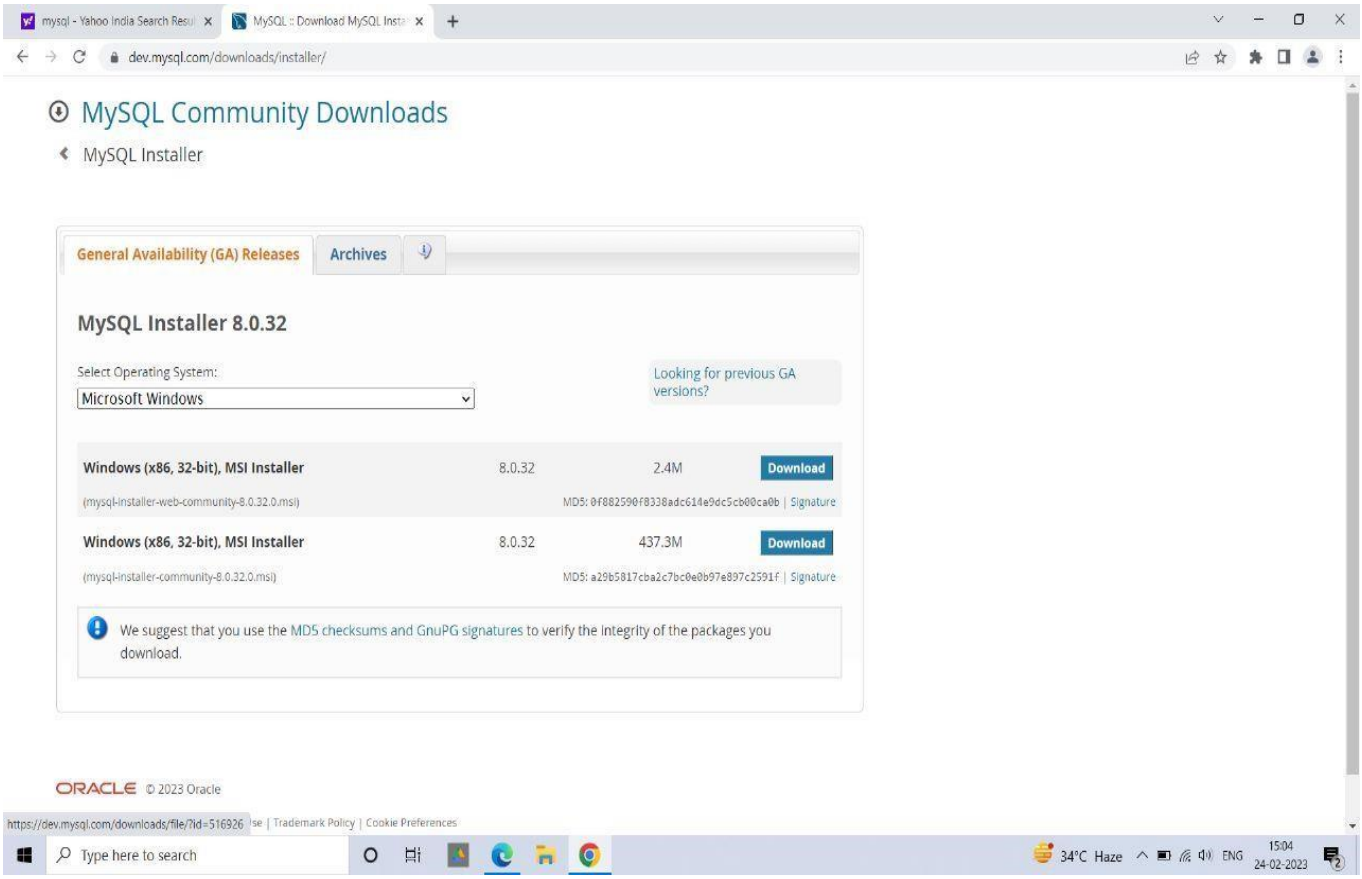
STEP 4:


mysql - Yahoo India Search Resul
MySQL :: Begin Your Download
dev.mysql.com/downloads/file/?id=516926
MySQL Community Downloads
Login Now or Sign Up for a free account.
An Oracle Web Account provides you with the following advantages:
Fast access to MySQL software downloads
Download technical White Papers and Presentations
Post messages in the MySQL Discussion Forums
Report and track bugs in the MySQL bug system
Login »
using my Oracle Web account
Sign Up »
for an Oracle Web account
MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.
No thanks, just start my download.
ORACLE © 2023 Oracle
Privacy / Do Not Sell My Info | Terms of Use | Trademark Policy | Cookie Preferences
https://dev.mysql.com/get/Downloads/MySQLInstaller/mysql-installer-web-community-8.0.32.0.msi
Type here to search
34°C Haze
ENG
15:05
24-02-2023

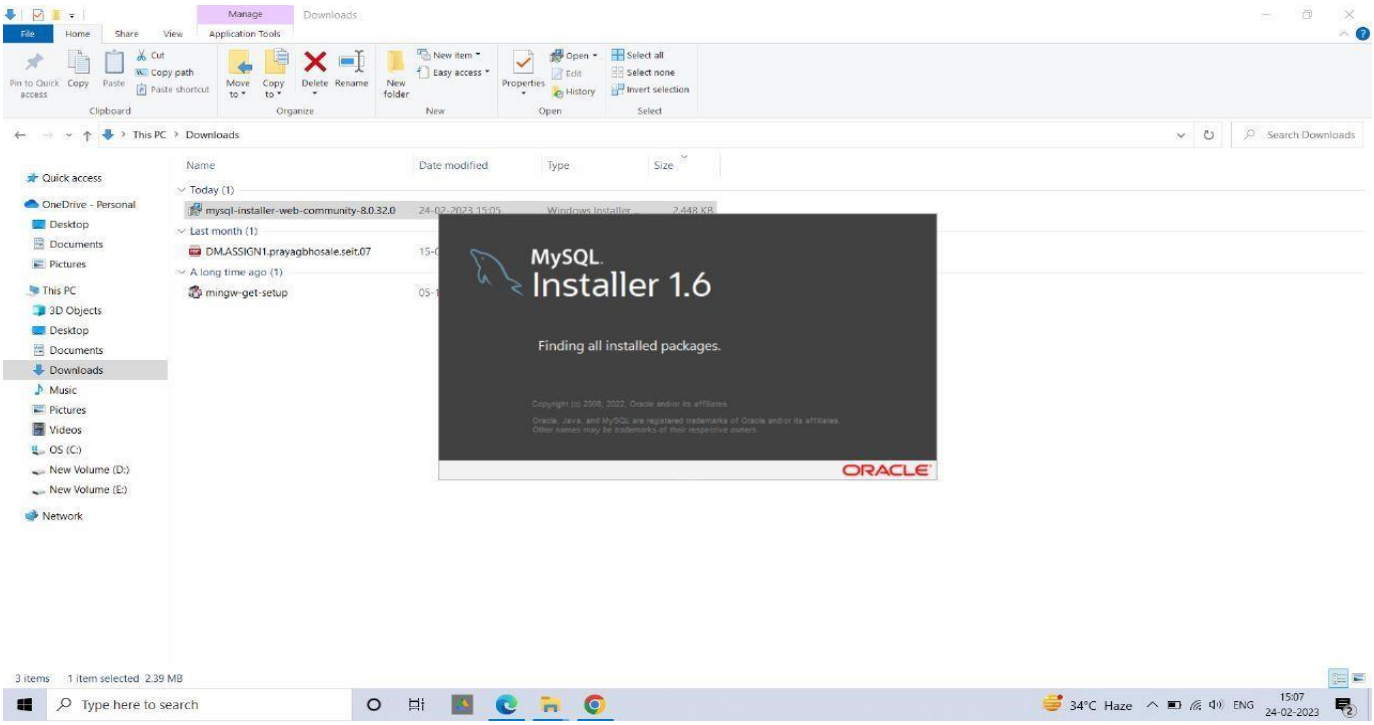
STEP 5:


Downloads
This PC › Downloads
Search Downloads
Name
Date modified
Type
Size
Today (1)
mysql-installer-web-community-8.0.32.0
24-02-2023 15:05
Windows Installer ...
2,448 KB
Last month (1)
DM.ASSIGN1.prayagbhosale.seit.07
15-01-2023 11:46
Microsoft Edge PD...
3,805 KB
A long time ago (1)
mingw-get-setup
05-12-2022 12:50
MySQL Installer - Community
Please wait while Windows configures MySQL Installer - Community
Cancel
3 items
1 item selected 2.39 MB
Type here to search
34°C Haze
ENG
15:06
24-02-2023

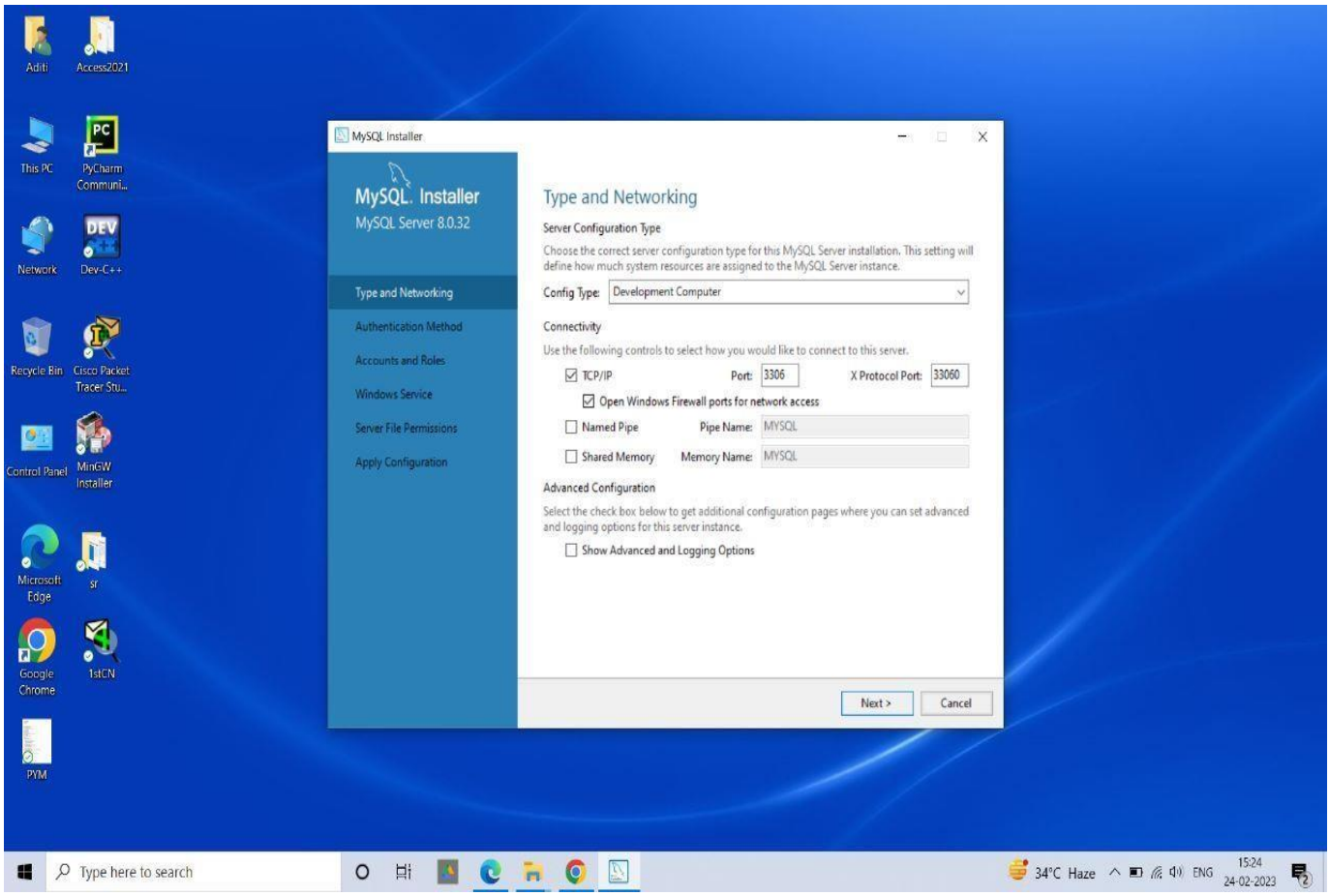
STEP 6:


dev.mysql.com/downloads/installer/
MySQL Community Downloads
MySQL Installer
General Availability (GA) Releases
Archives
MySQL Installer 8.0.32
Select Operating System:
Microsoft Windows
Looking for previous GA versions?
Windows (x86, 32-bit), MSI Installer
8.0.32
2.4M
Download
(mysql-installer-web-community-8.0.32.0.msi)
MD5: 0f882590f8338adc614e9dc5cb00ca0b | Signature
Windows (x86, 32-bit), MSI Installer
8.0.32
437.3M
Download
(mysql-installer-community-8.0.32.0.msi)
MD5: a29b5817cba2c7bc0e0b97e897c2591f | Signature
We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.
ORACLE © 2023 Oracle
Trademark Policy | Cookie Preferences
Type here to search
34°C Haze
ENG
15:04
24-02-2023

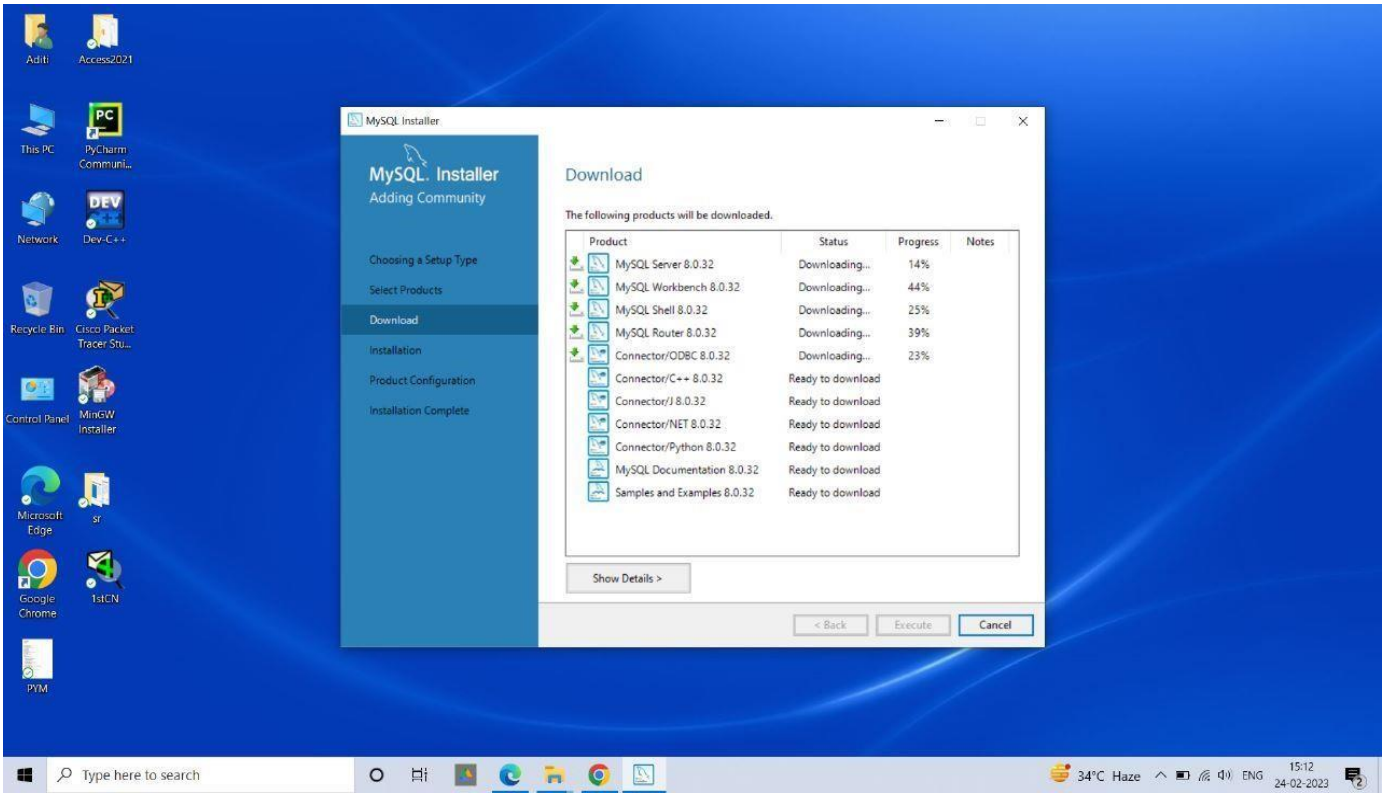
STEP 7:


MySQL.
Installer 1.6
Finding all installed packages.
ORACLE

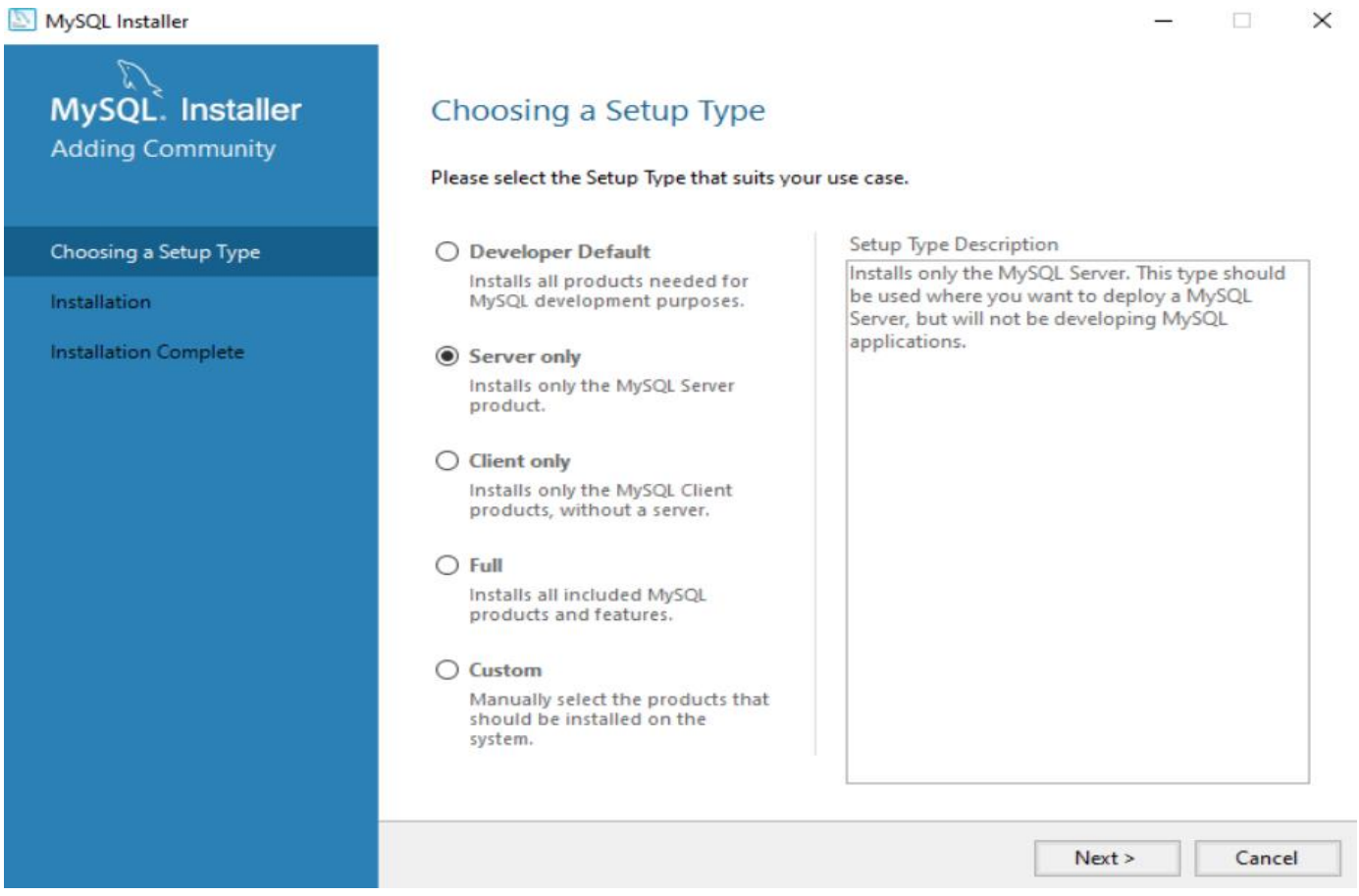
STEP 8:


MySQL Installer
MySQL. Installer
MySQL Server 8.0.32
Type and Networking
Authentication Method
Accounts and Roles
Windows Service
Server File Permissions
Apply Configuration
Type and Networking
Server Configuration Type
Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.
Config Type: Development Computer
Connectivity
Use the following controls to select how you would like to connect to this server.
TCP/IP
Port: 3306
X Protocol Port: 33060
Open Windows Firewall ports for network access
Named Pipe
Pipe Name: MYSQL
Shared Memory
Memory Name: MYSQL
Advanced Configuration
Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.
Show Advanced and Logging Options
Next >
Cancel

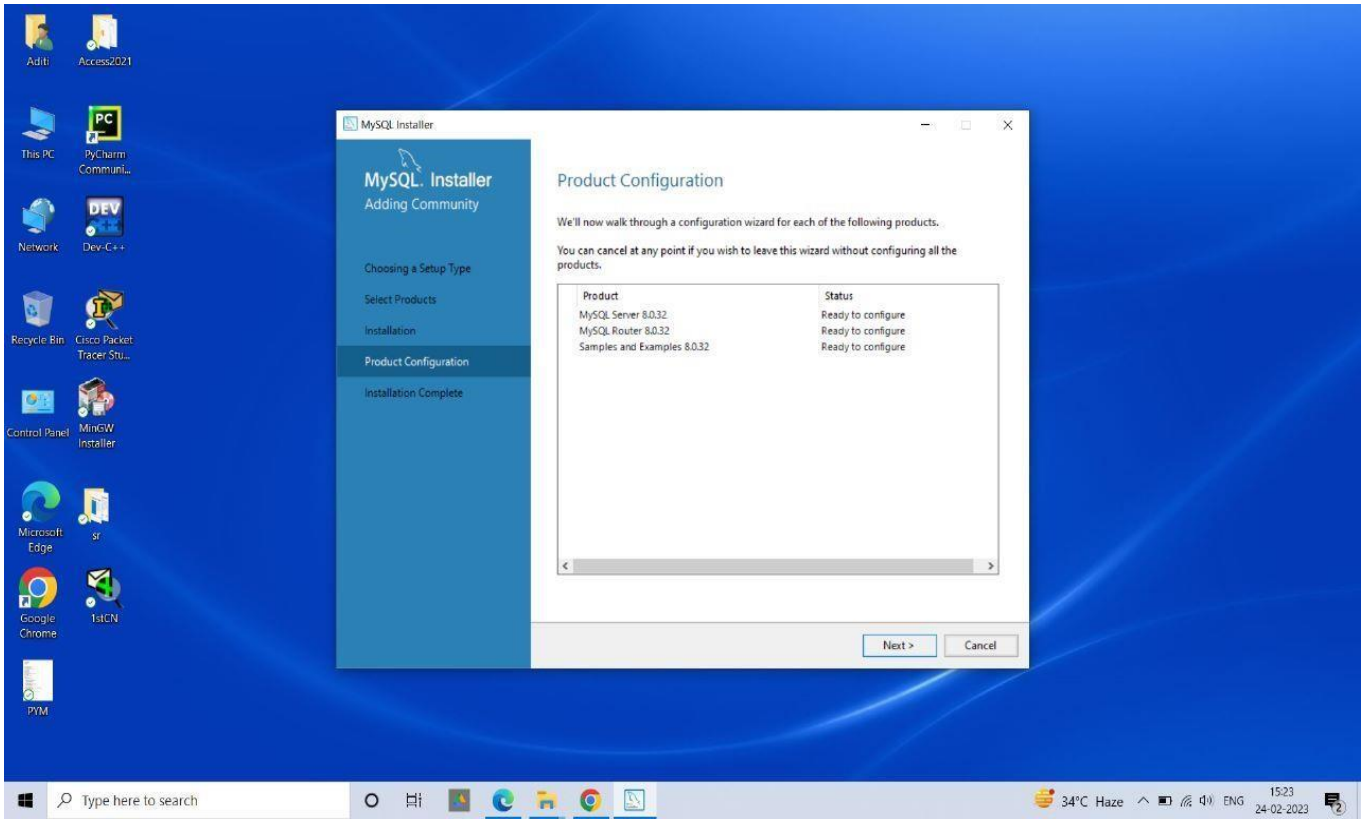
STEP 9:



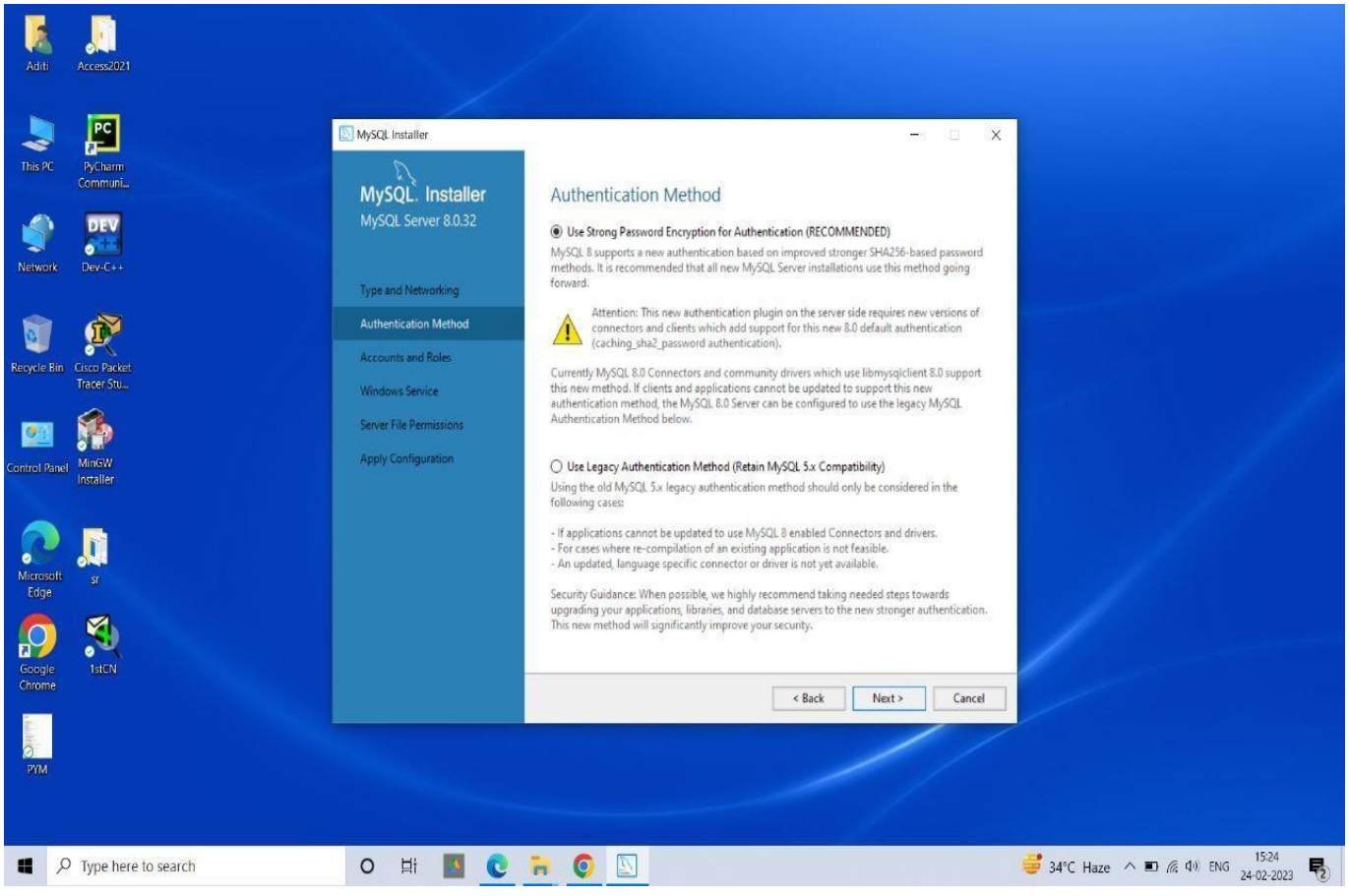
STEP 10:



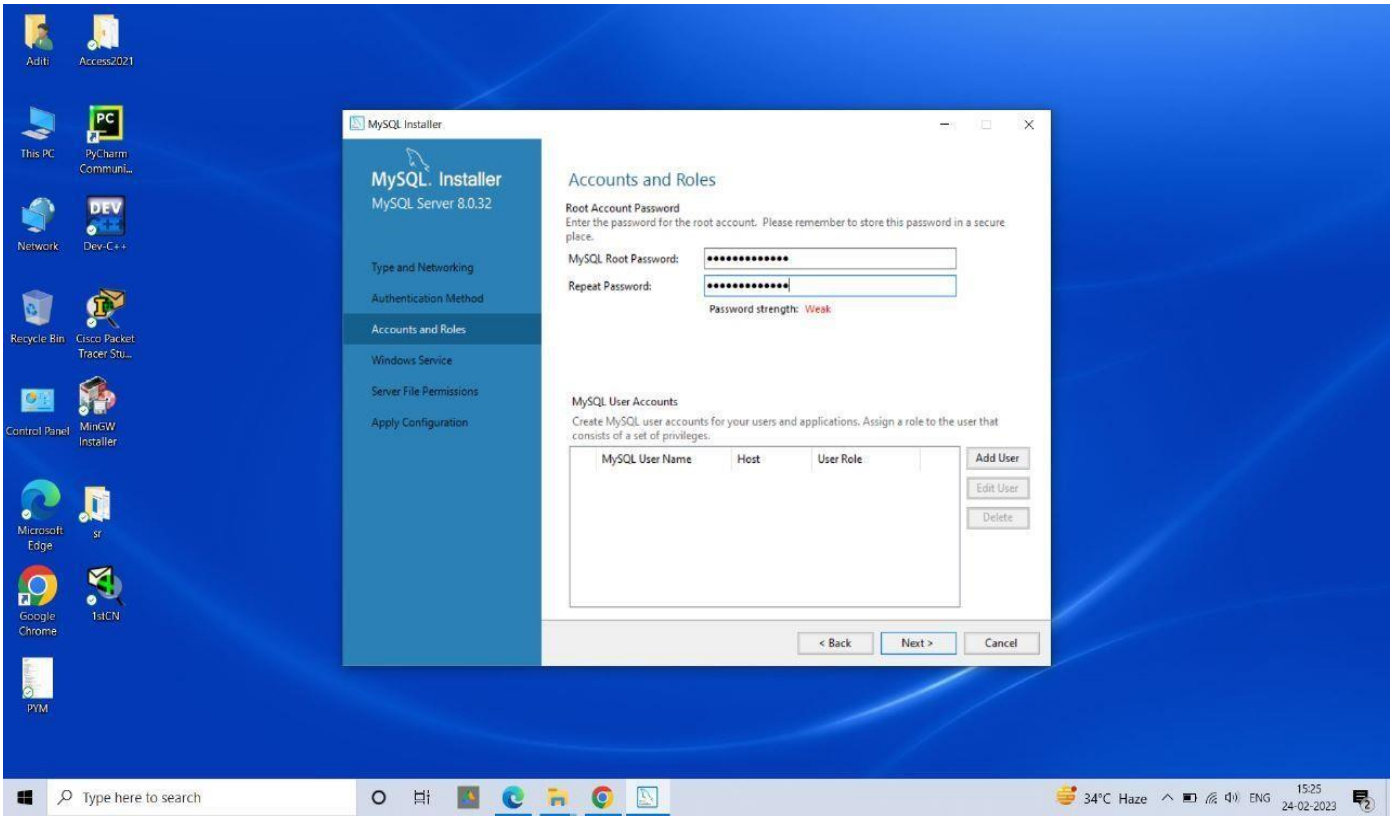
STEP 11:



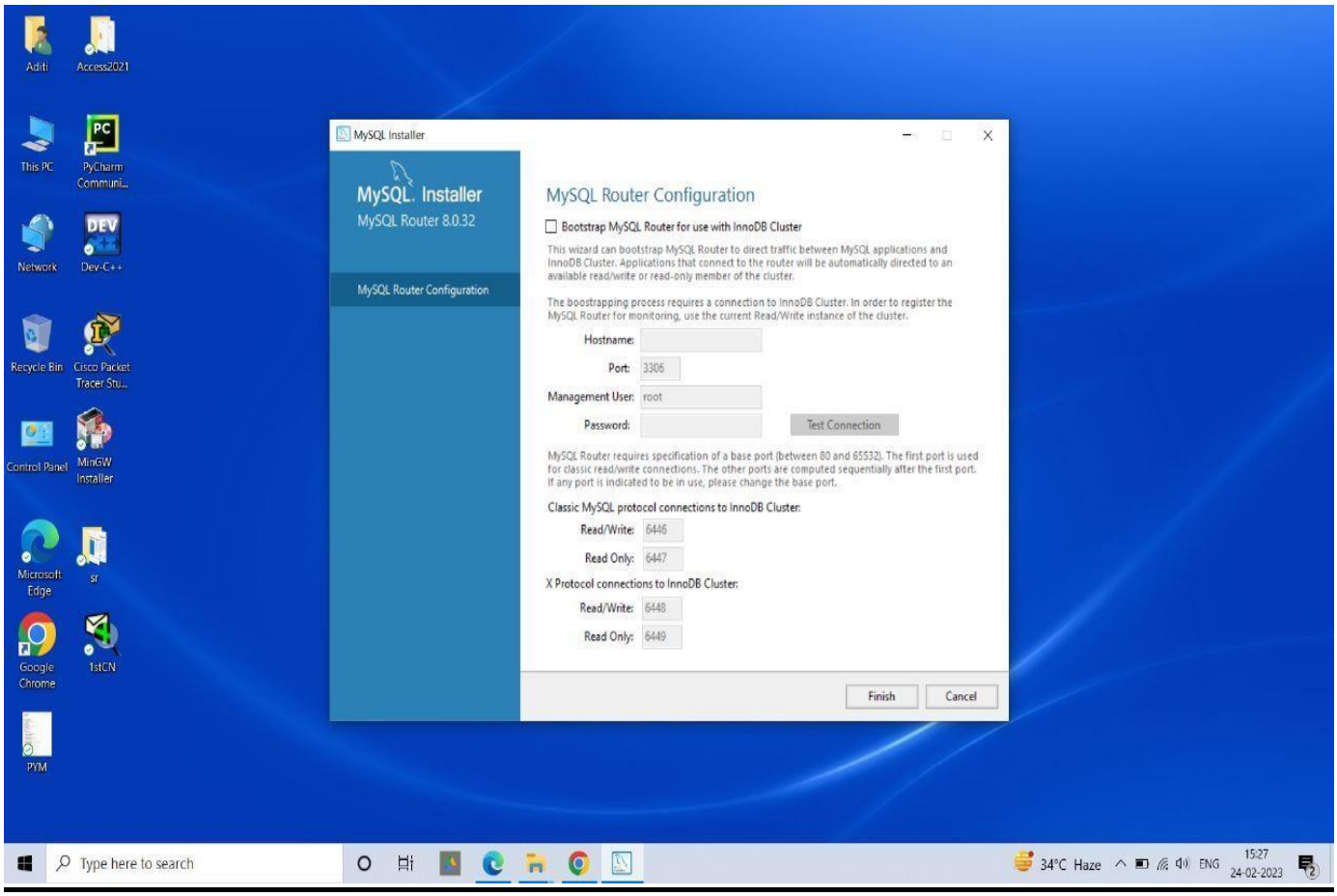
STEP 12:



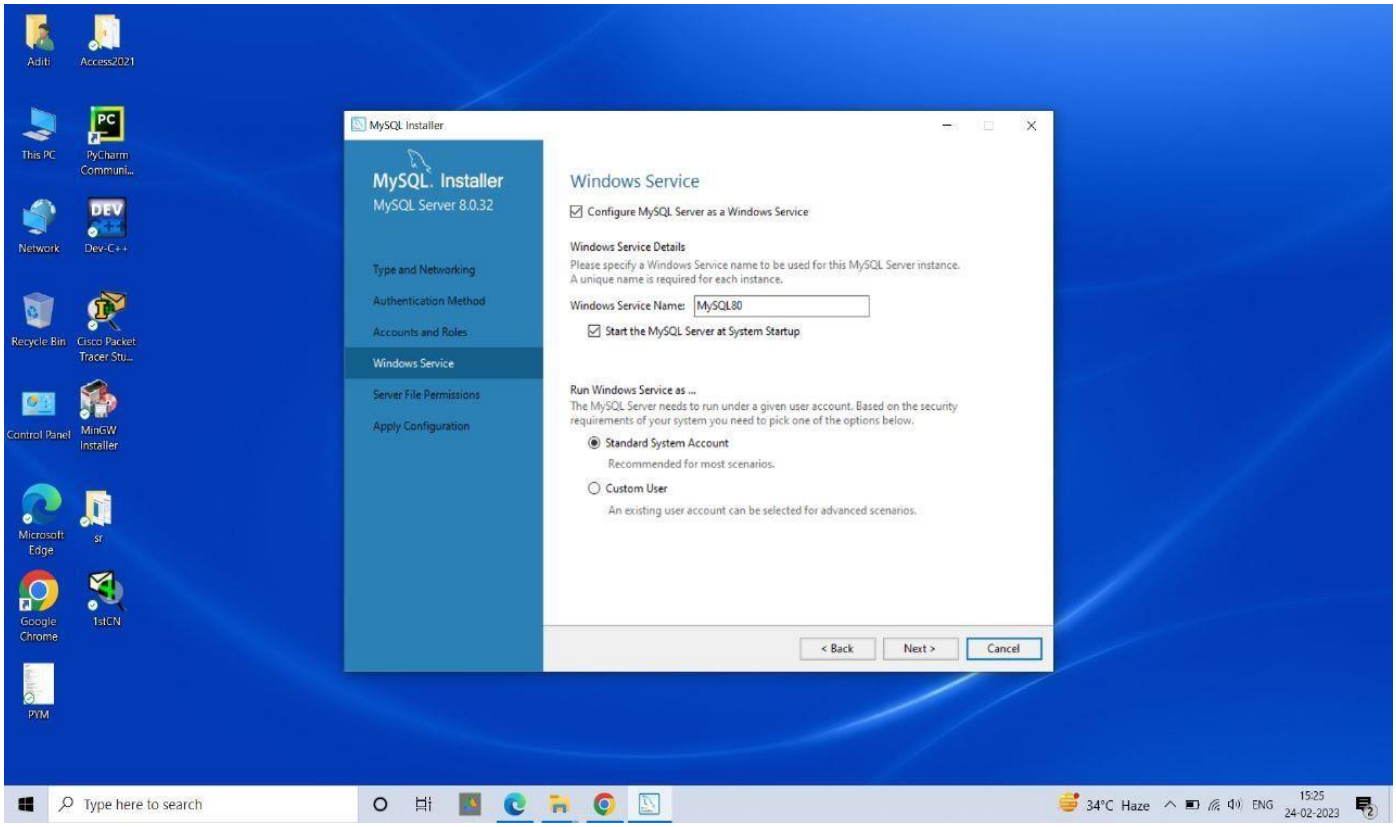
STEP 13:



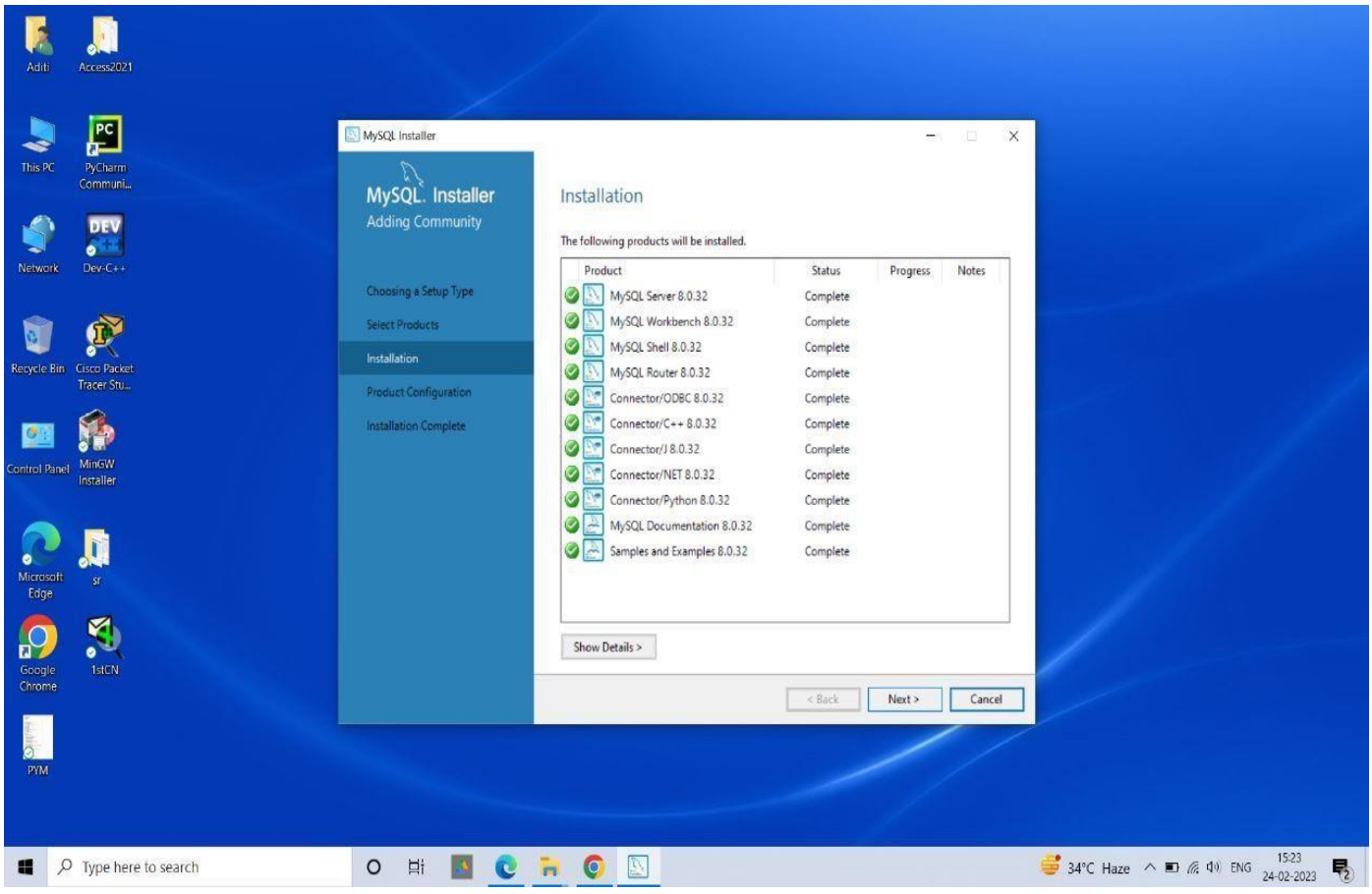
STEP 14:



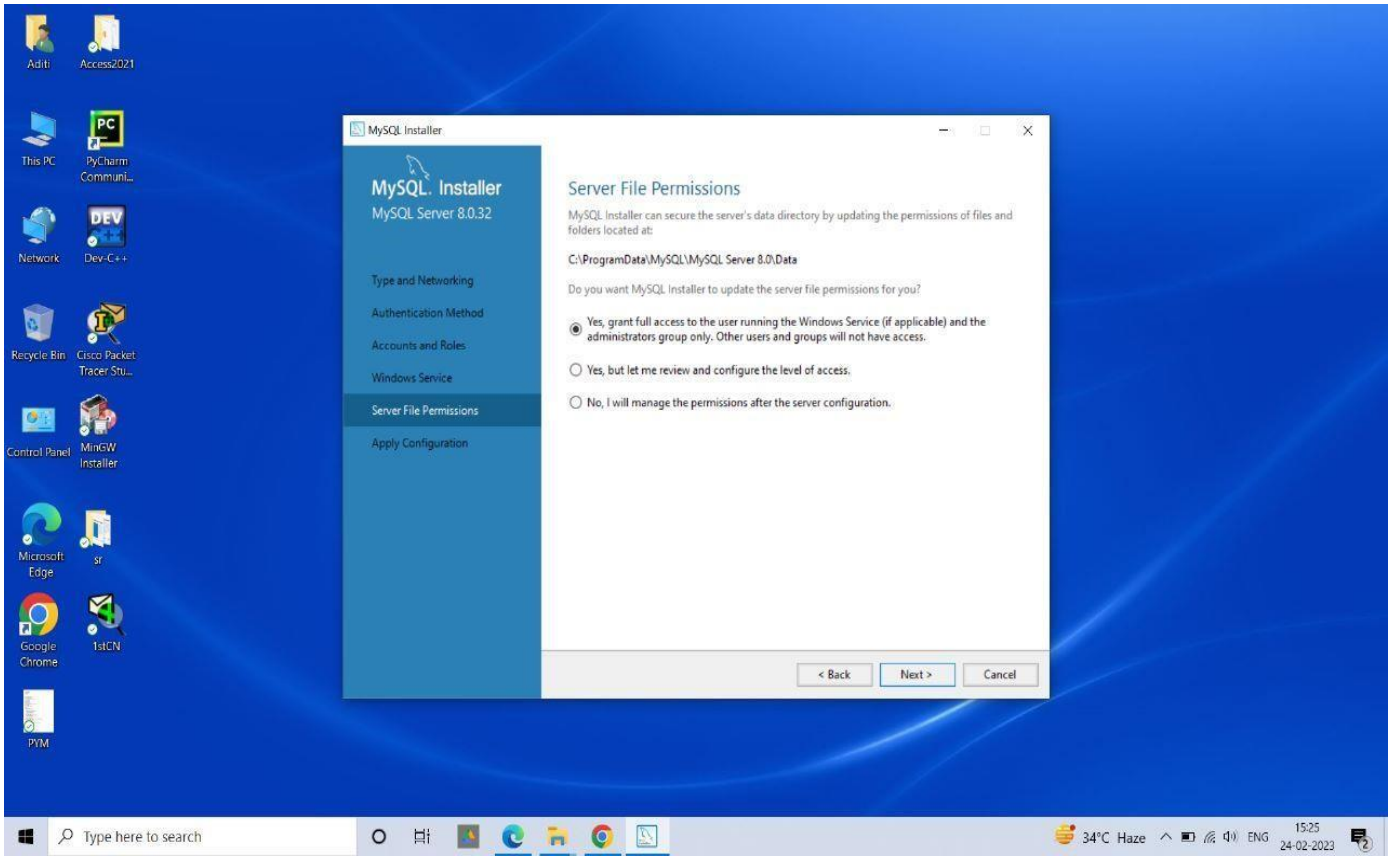
STEP 15:



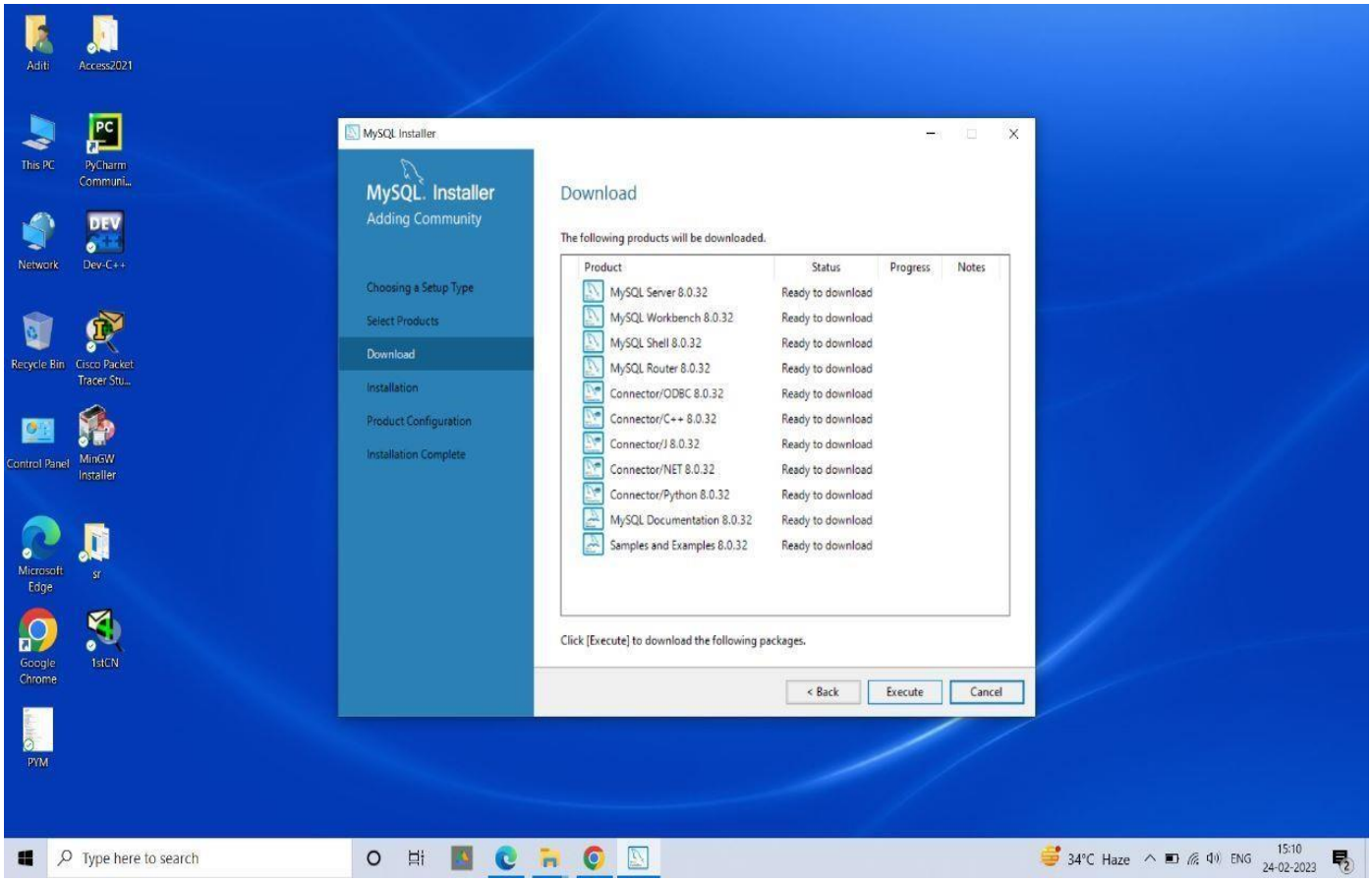
STEP 16:



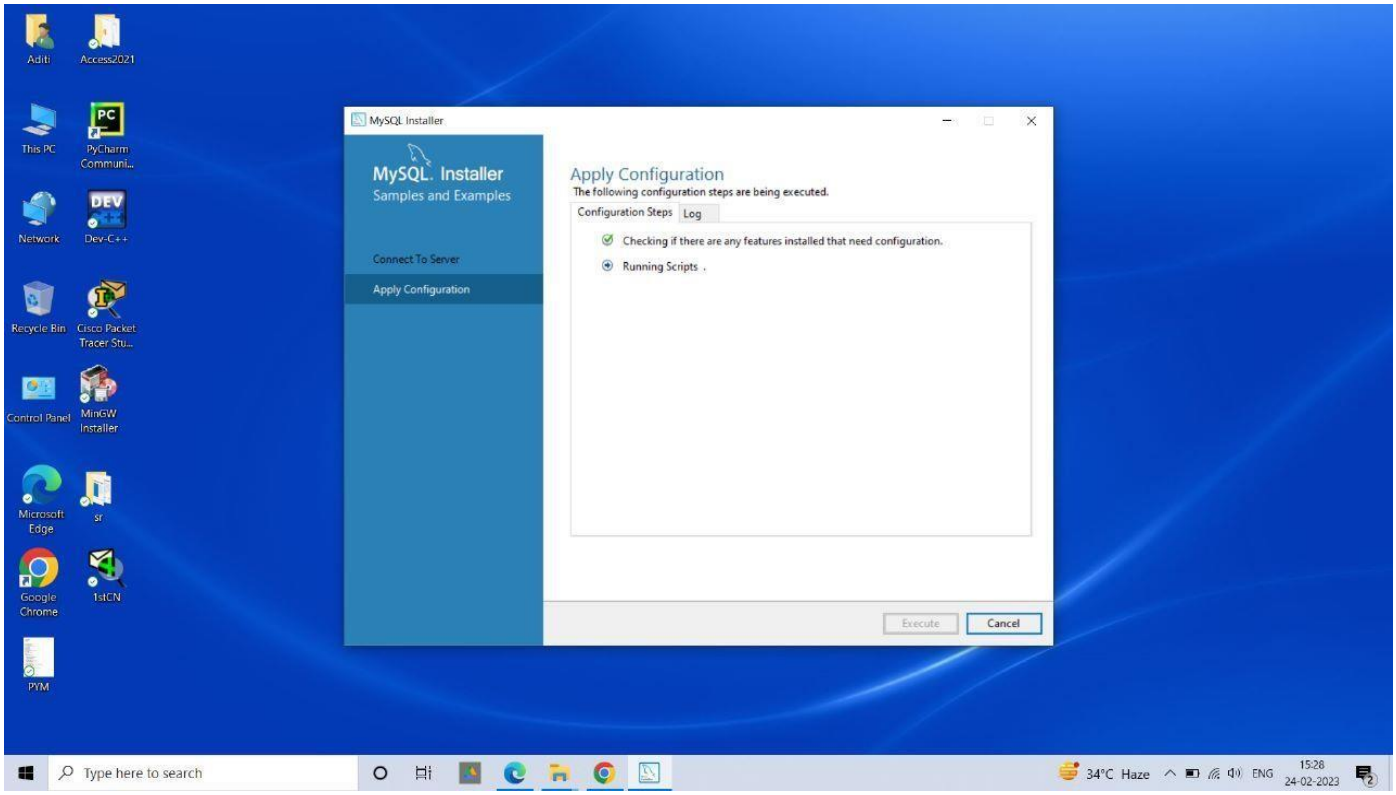
STEP 17:



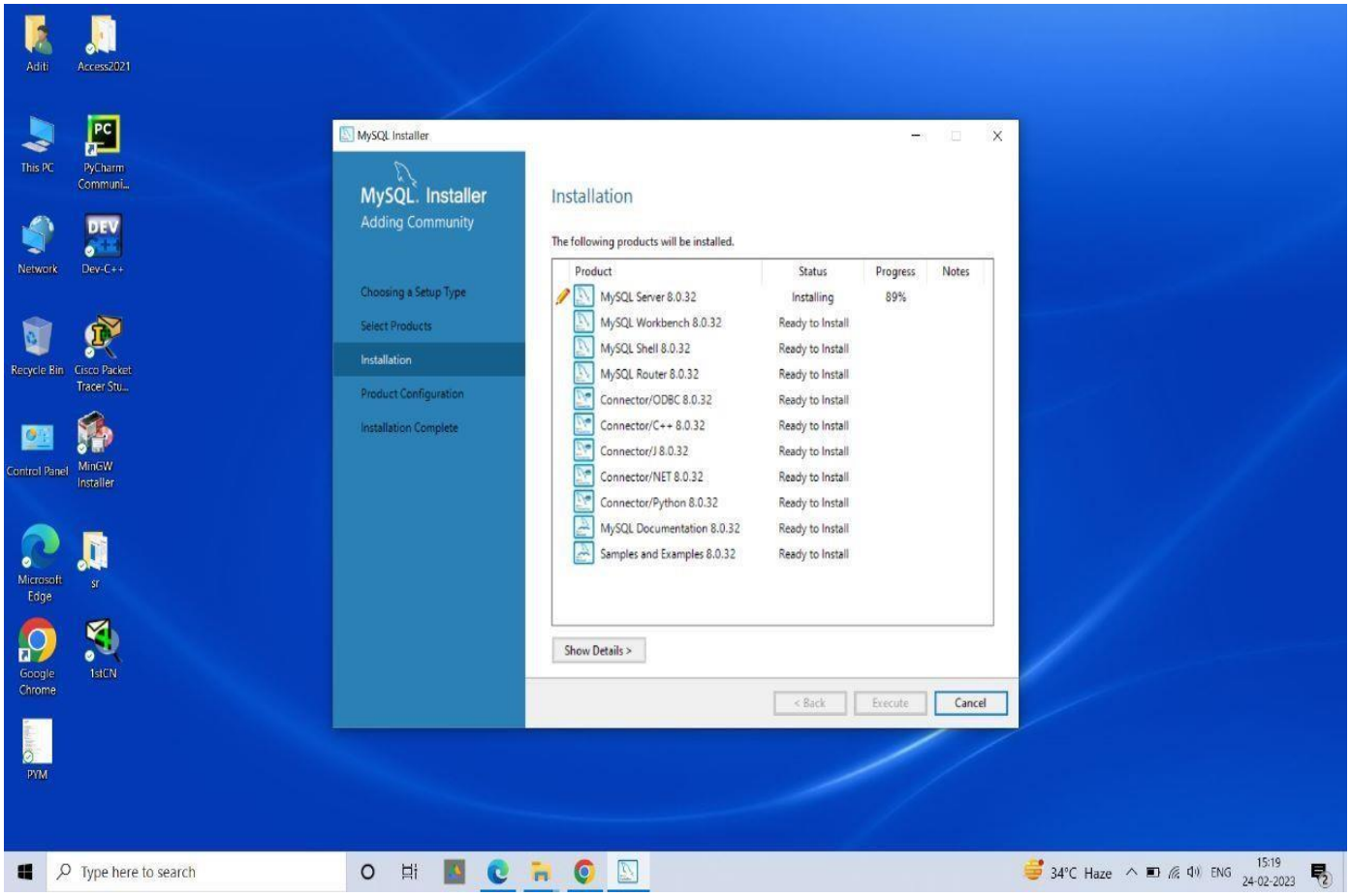
STEP 18:



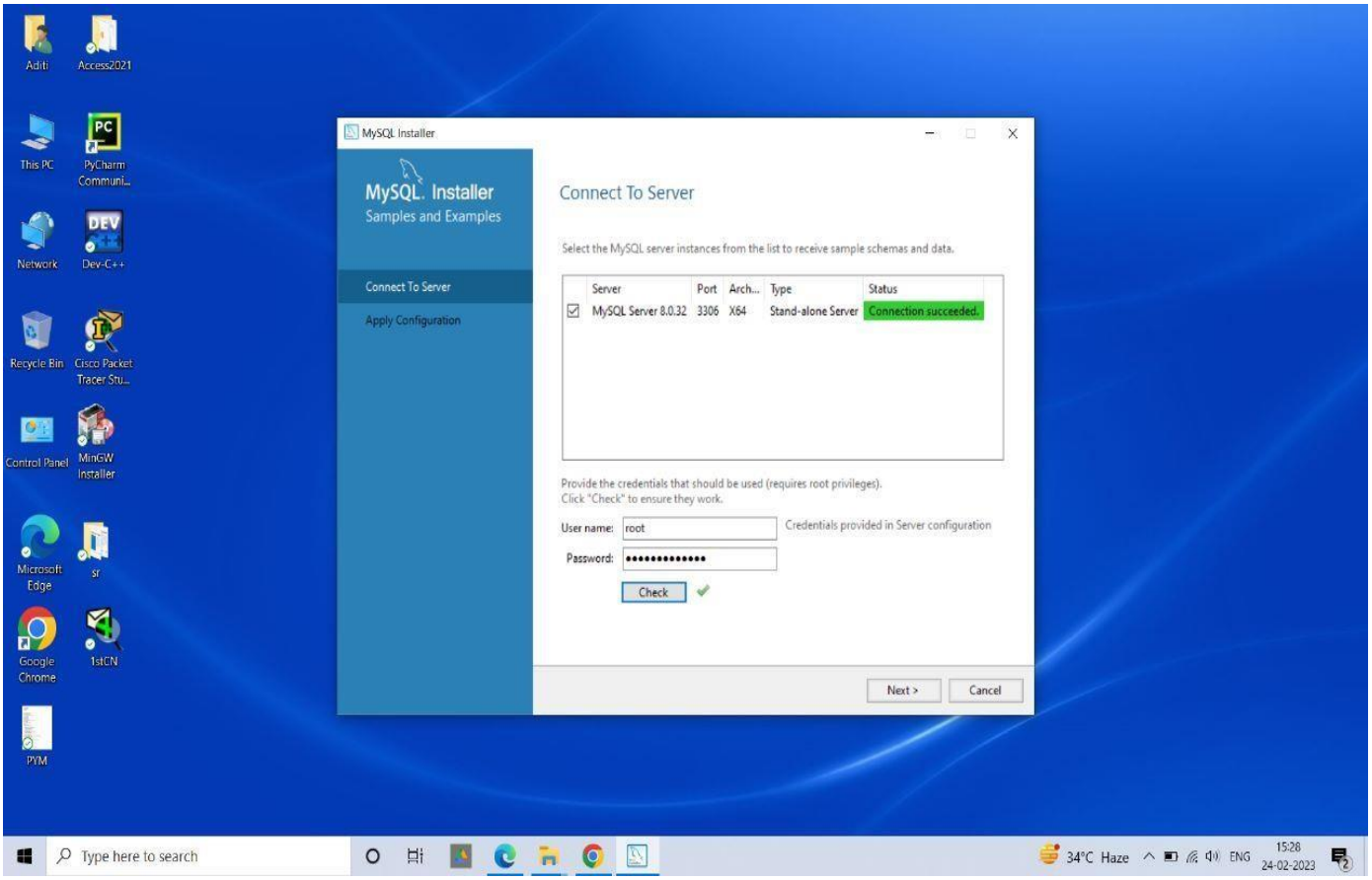
STEP 19:



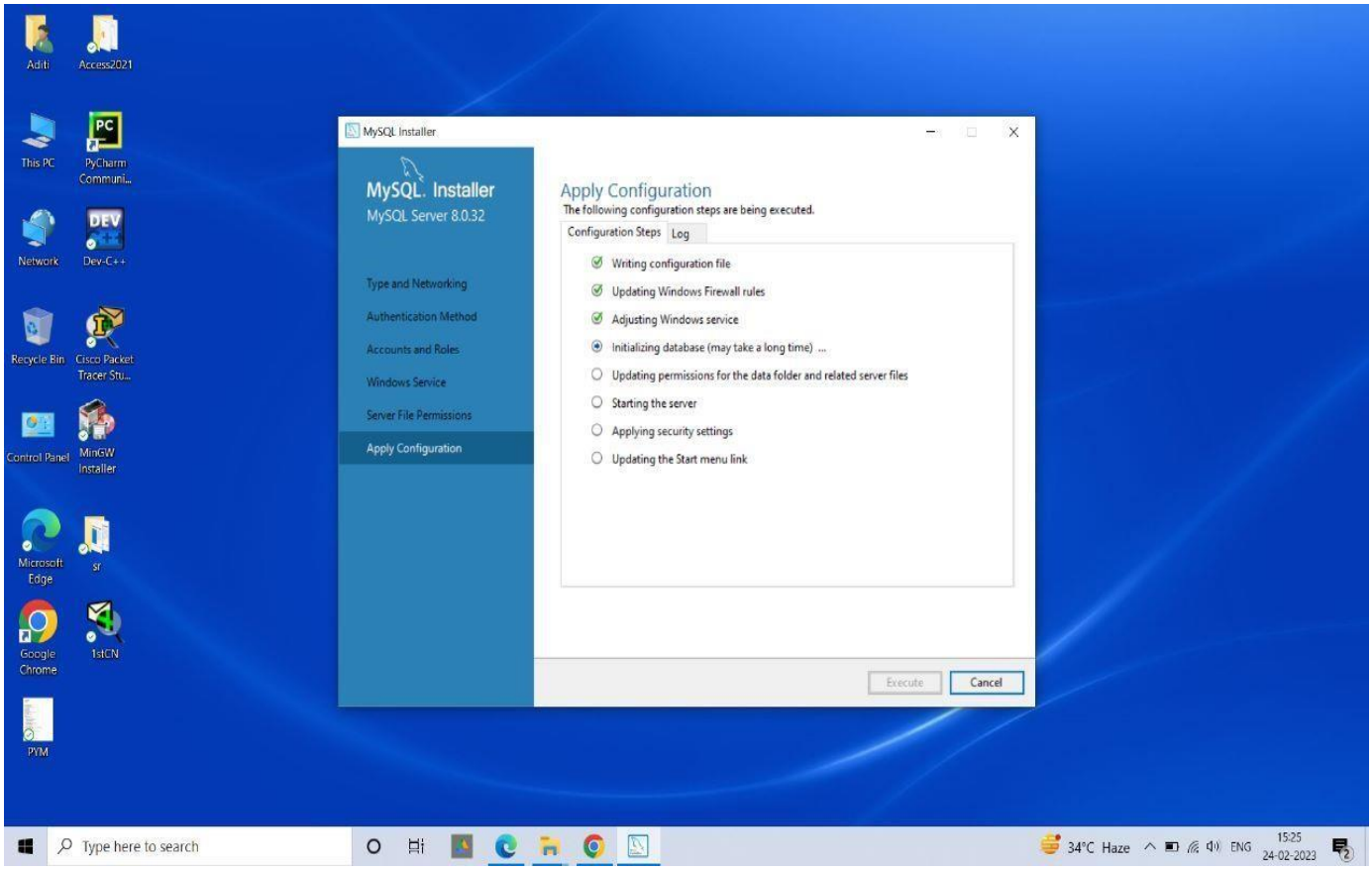
STEP 20:



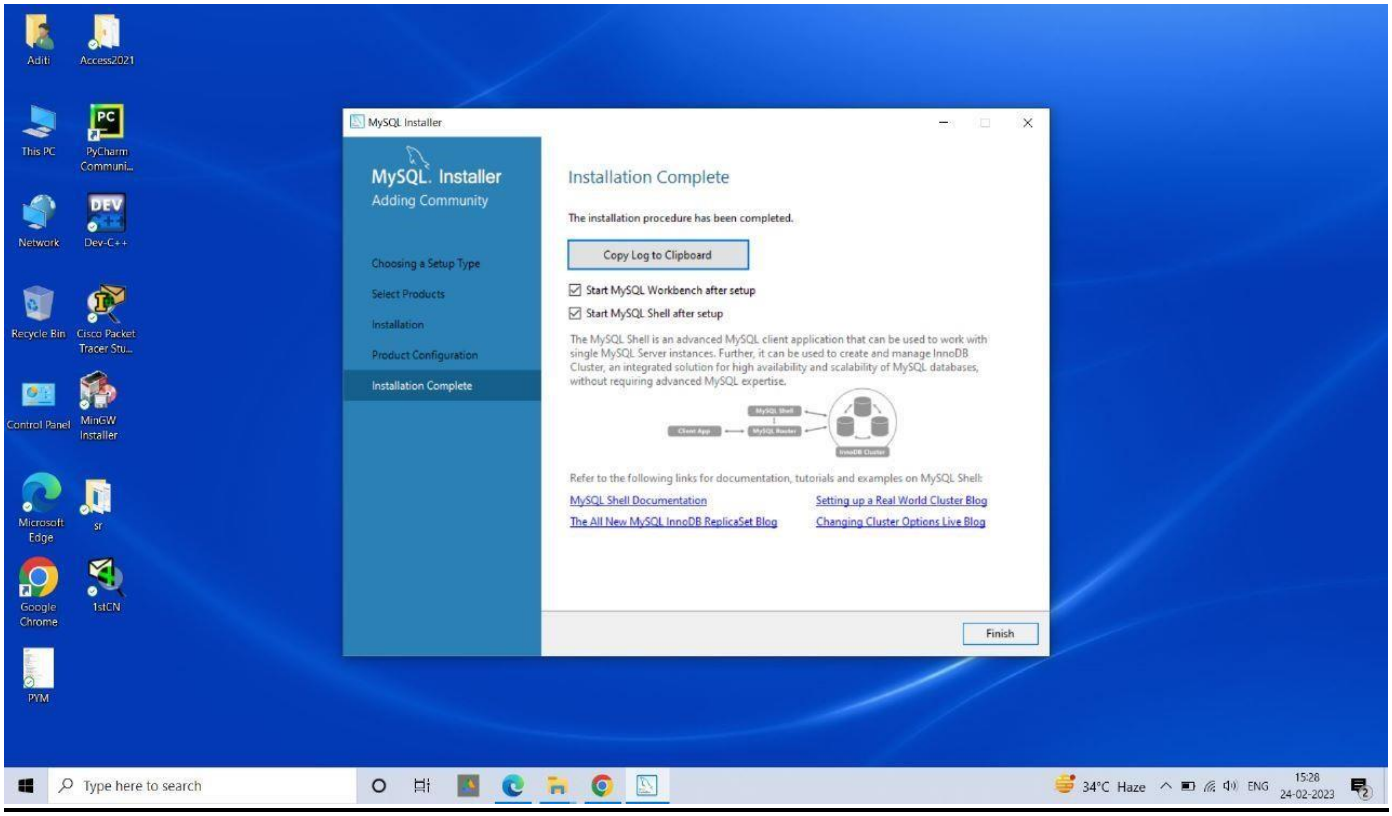
STEP 21:



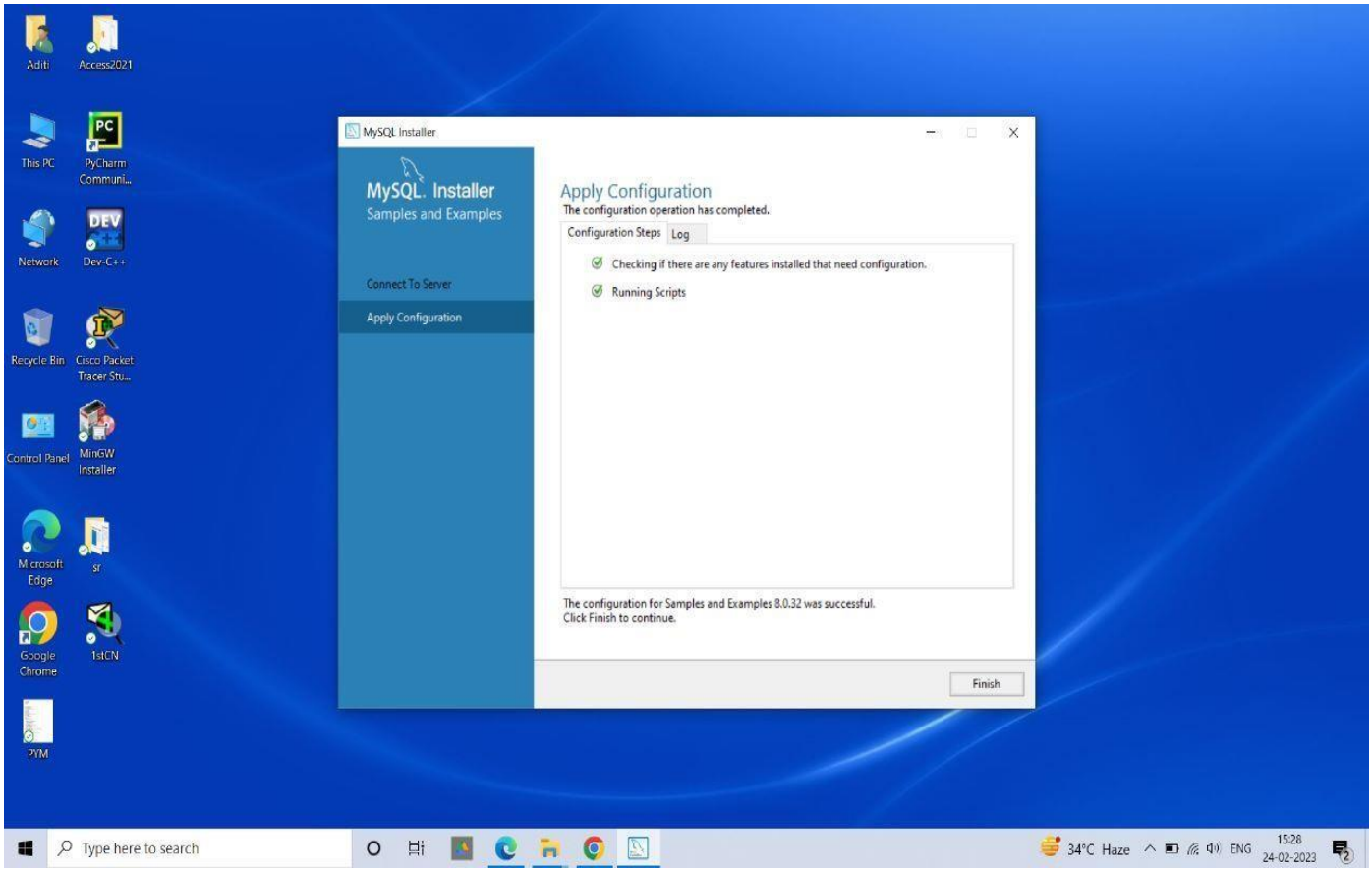
STEP 22:



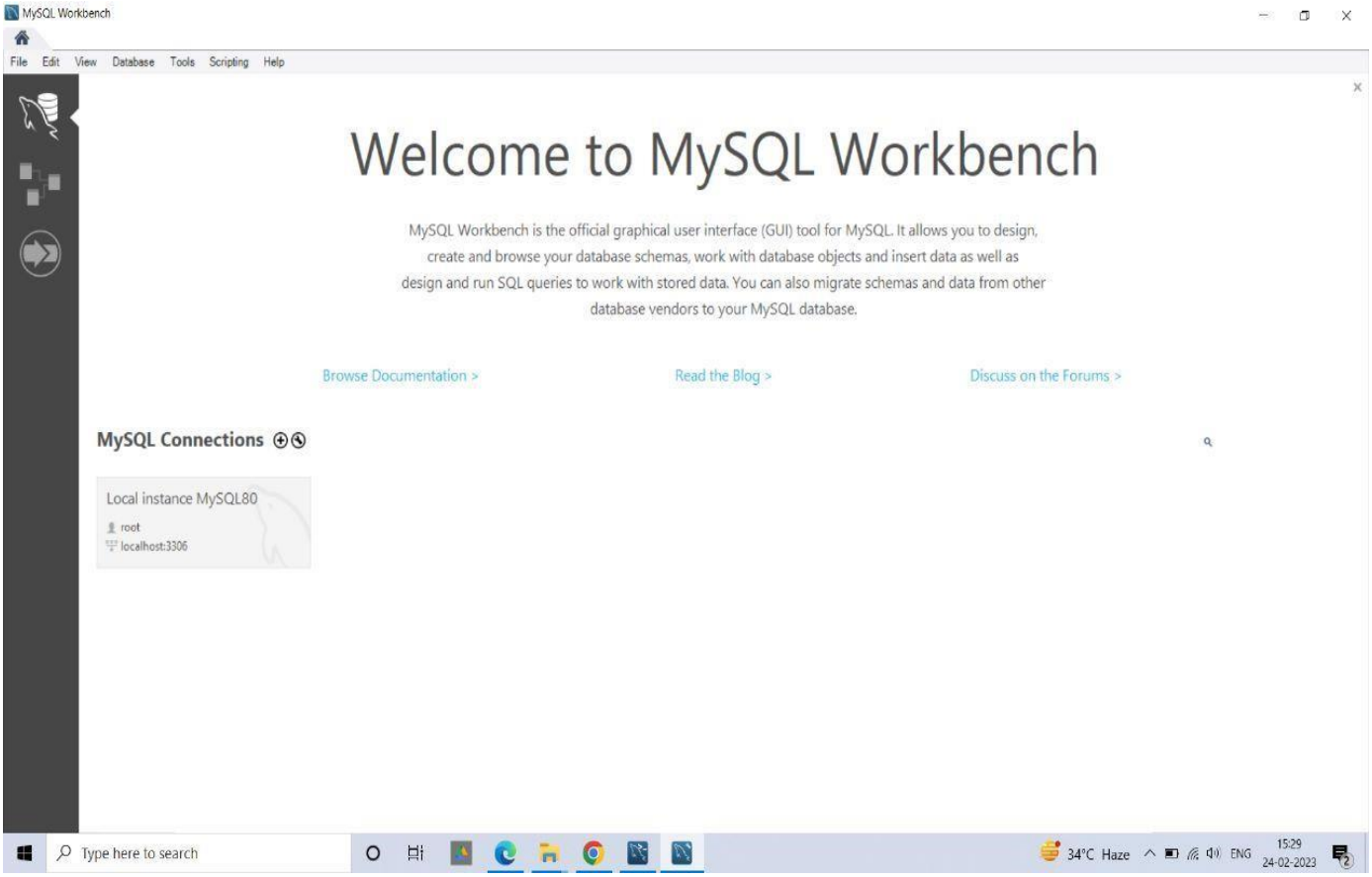
STEP 23:



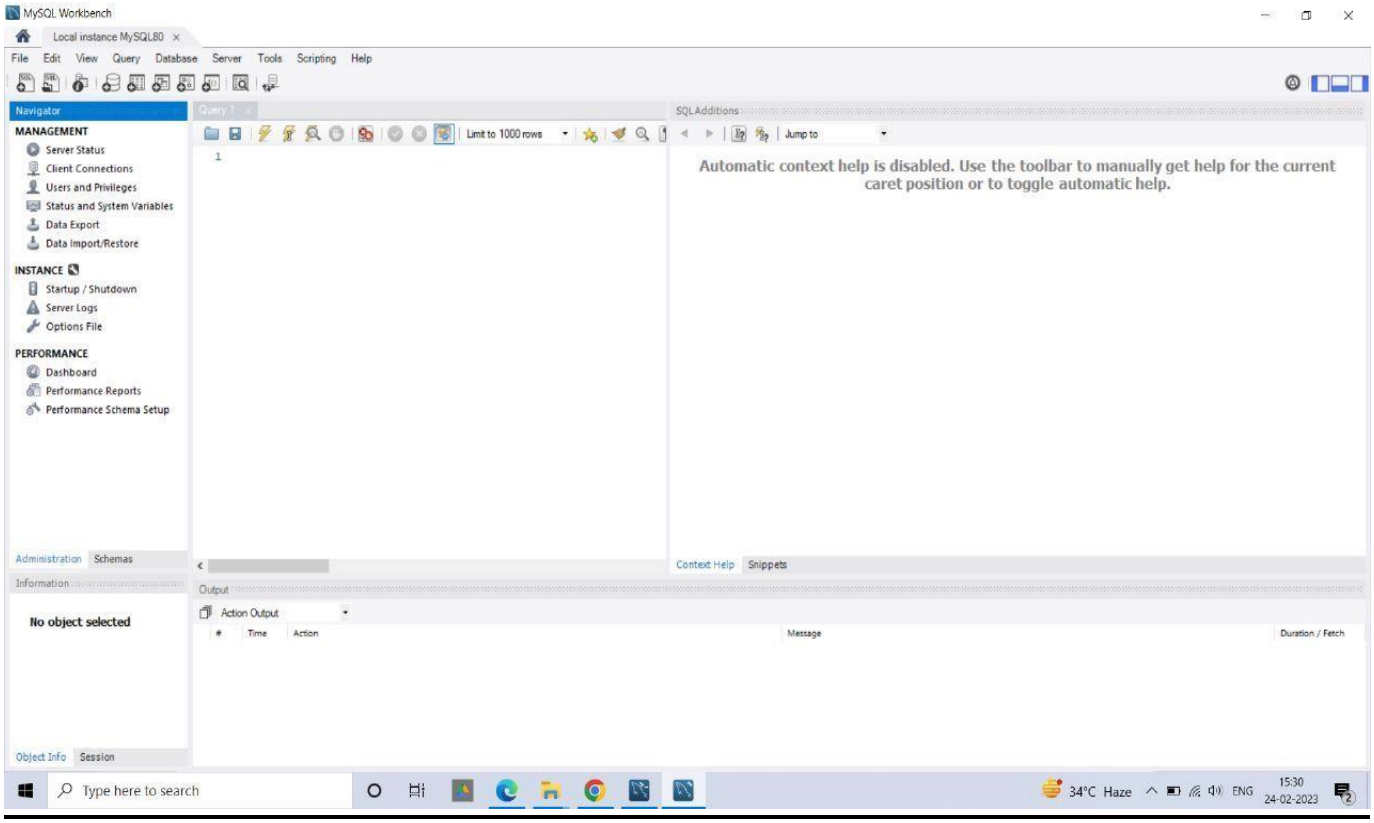
STEP 24:



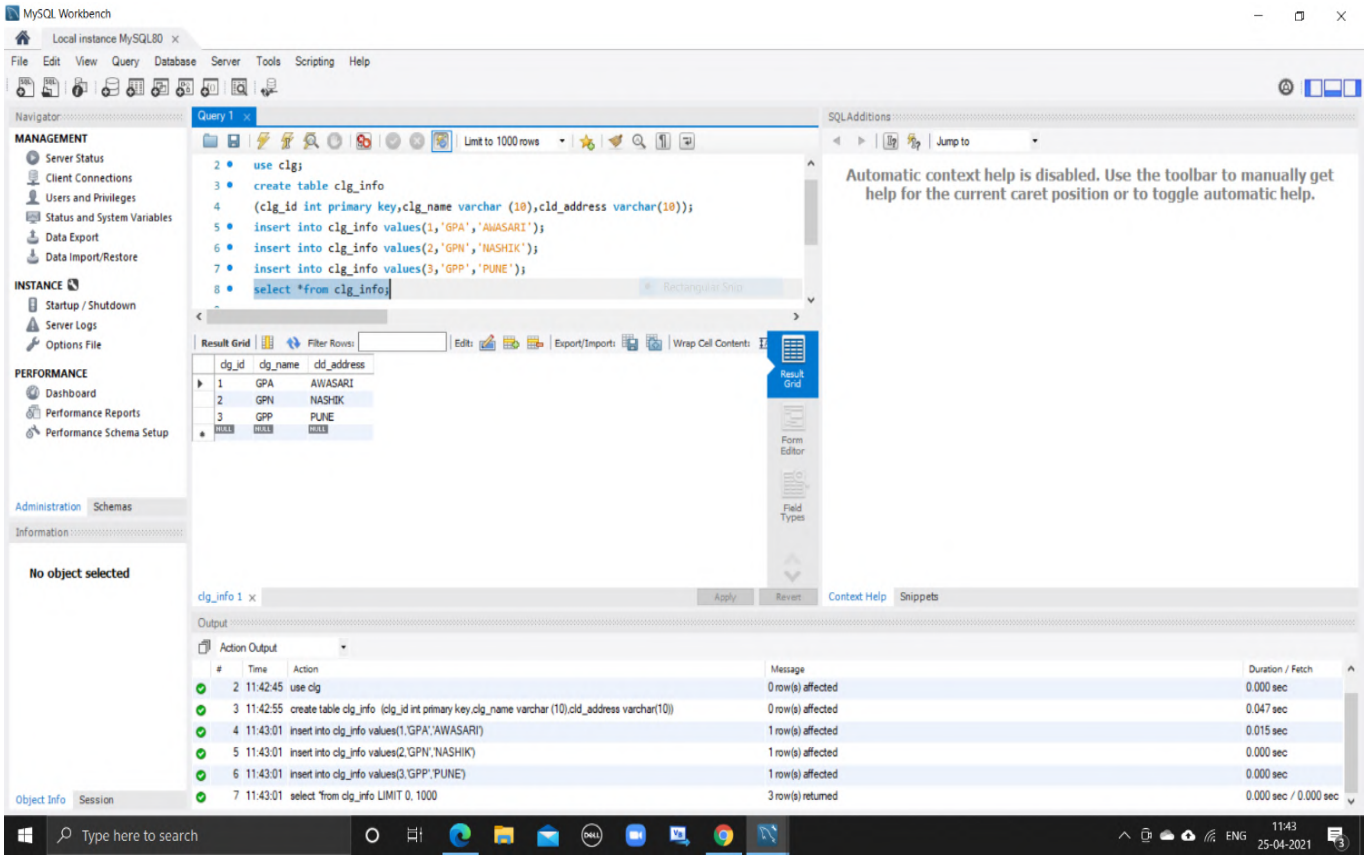
STEP 25:



STEP 26:



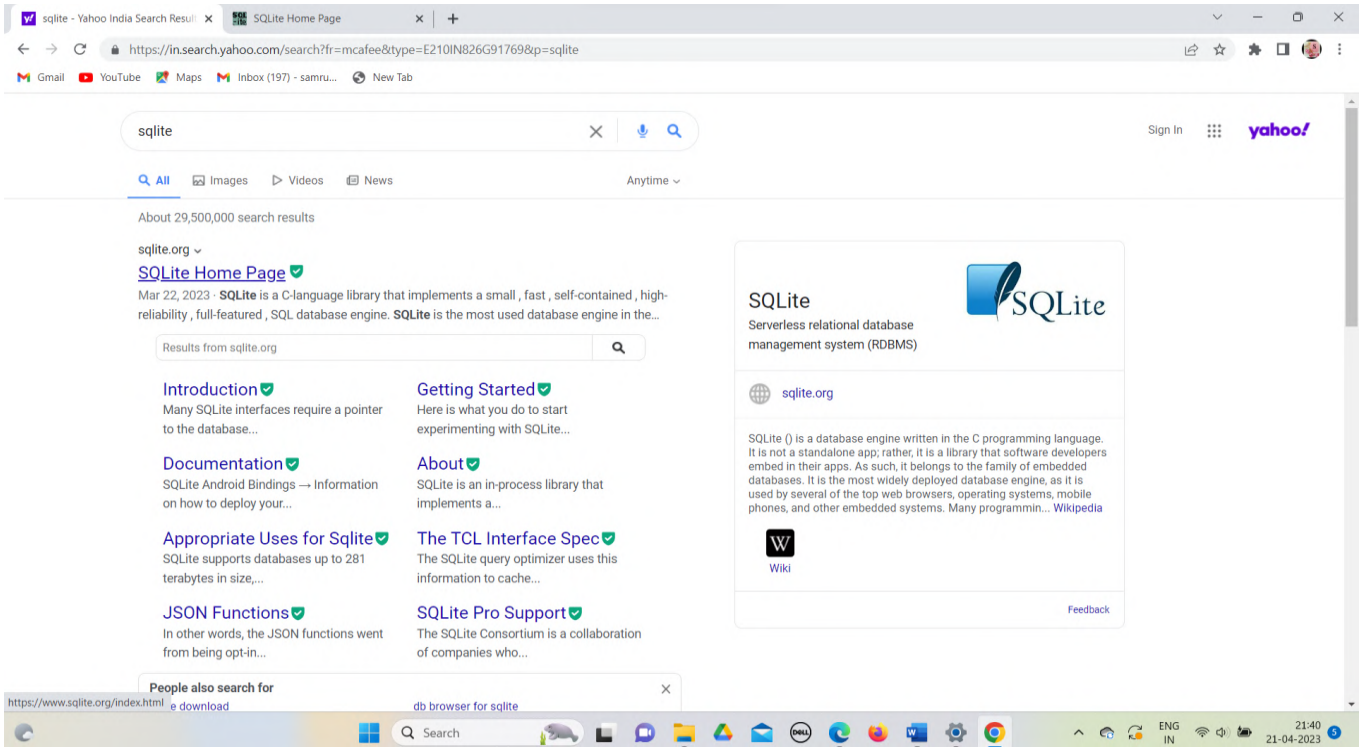
STEP 27:


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
MANAGEMENT
Server Status
Client Connections
Users and Privileges
Status and System Variables
Data Export
Data Import/Restore
INSTANCE
Startup / Shutdown
Server Logs
Options File
PERFORMANCE
Dashboard
Performance Reports
Performance Schema Setup
Administration Schemas
Information
No object selected
Object Info Session
Query 1
Limit to 1000 rows
use clg;
create table clg_info
(clg_id int primary key,clg_name varchar (10),cld_address varchar(10));
insert into clg_info values(1,'GPA','AWASARI');
insert into clg_info values(2,'GPN','NASHIK');
insert into clg_info values(3,'GPP','PUNE');
select *from clg_info;
Result Grid
clg_id clg_name cld_address
1 GPA AWASARI
2 GPN NASHIK
3 GPP PUNE
clg_info 1
Apply Revert
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output
Time Action Message Duration / Fetch
2 11:42:45 use clg 0 row(s) affected 0.000 sec
3 11:42:55 create table clg_info (clg_id int primary key,clg_name varchar (10),cld_address varchar(10)) 0 row(s) affected 0.047 sec
4 11:43:01 insert into clg_info values(1,'GPA','AWASARI') 1 row(s) affected 0.015 sec
5 11:43:01 insert into clg_info values(2,'GPN','NASHIK') 1 row(s) affected 0.000 sec
6 11:43:01 insert into clg_info values(3,'GPP','PUNE') 1 row(s) affected 0.000 sec
7 11:43:01 select *from clg_info LIMIT 0, 1000 3 row(s) returned 0.000 sec / 0.000 sec
Type here to search
ENG
11:43
25-04-2021

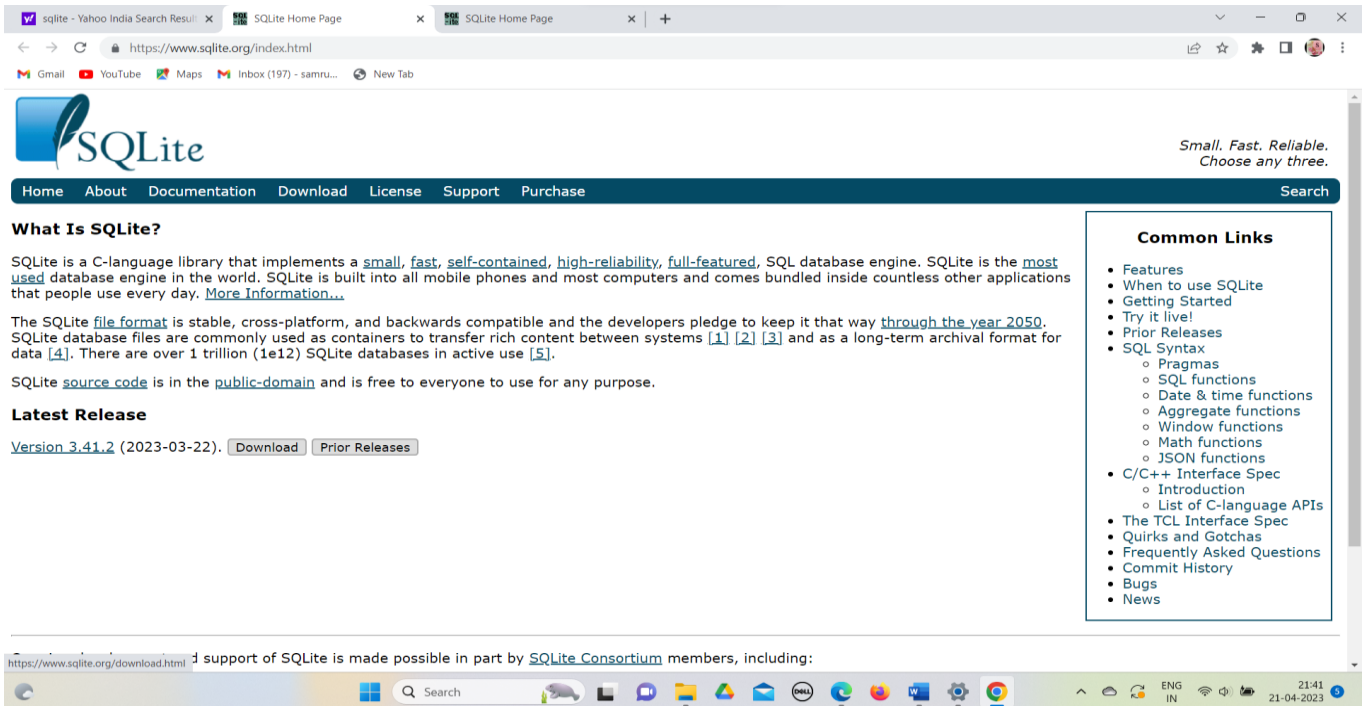
Group A: Study of Databases

3. Study of SQLite: What is SQLite? Uses of SQLite. Building and installing SQLite.

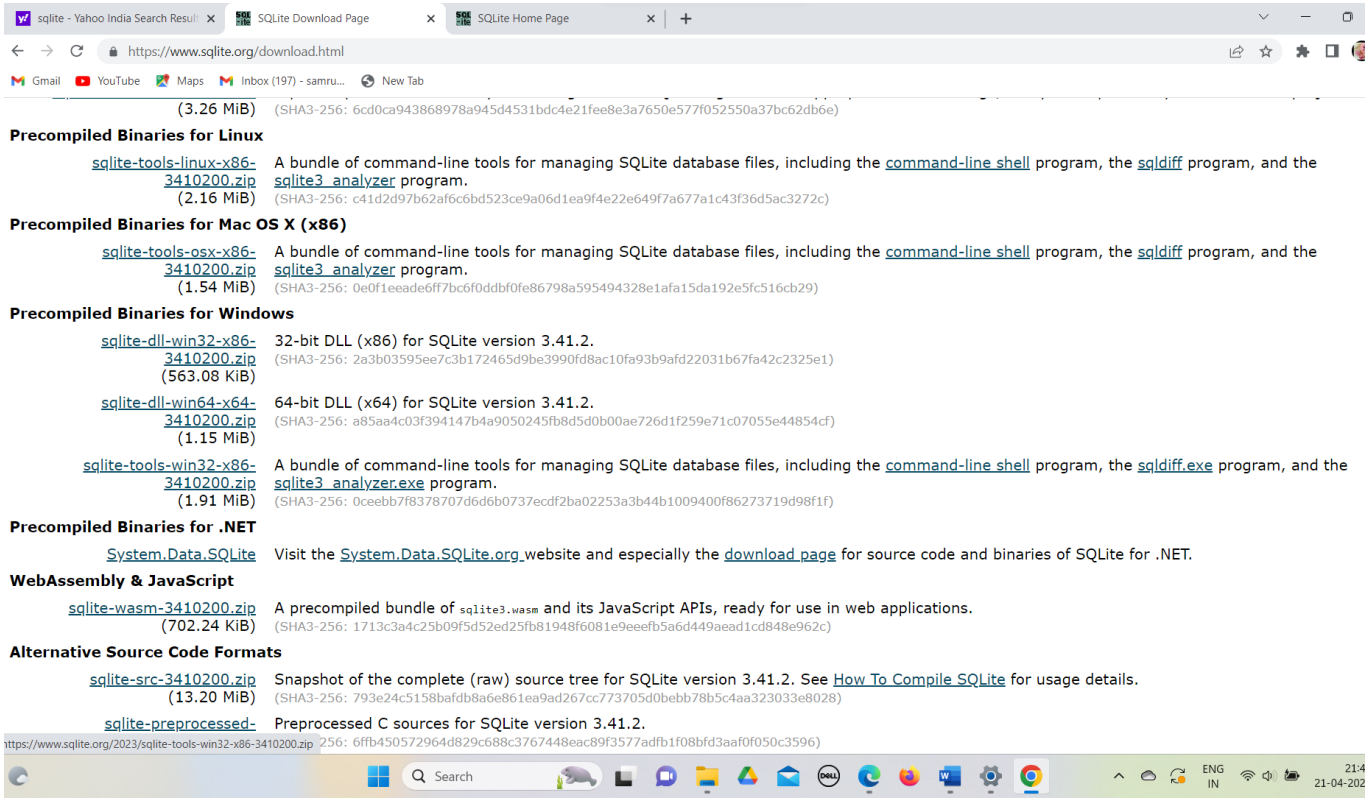
STEP 1:



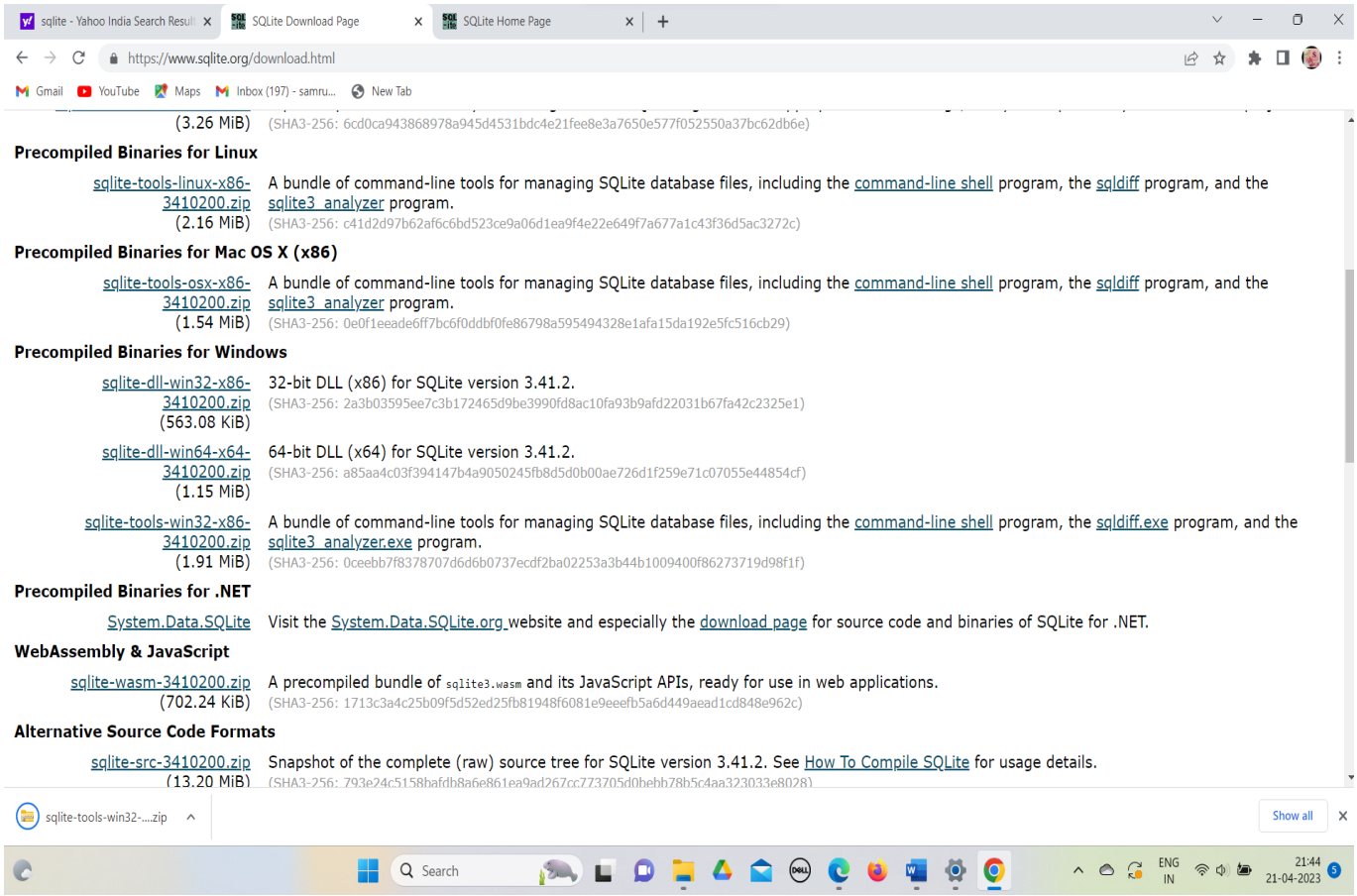
STEP 2:



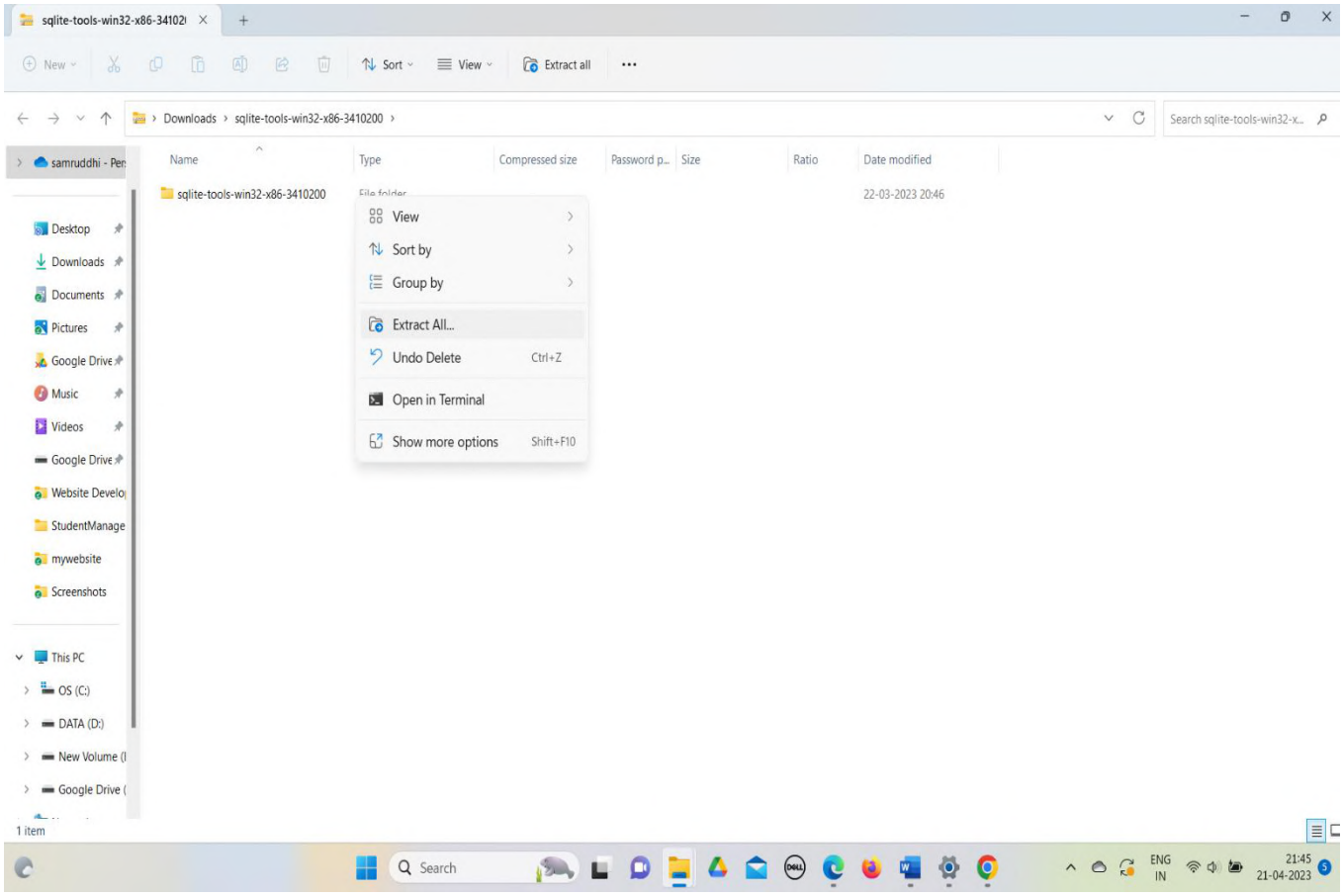
STEP 3:


sqlite - Yahoo India Search Result
SQLite Download Page
SQLite Home Page
https://www.sqlite.org/download.html
Gmail
YouTube
Maps
Inbox (197) - samru...
New Tab
(3.26 MiB) (SHA3-256: 6cd0ca943868978a945d4531bdc4e21fee8e3a7650e577f052550a37bc62db6e)
Precompiled Binaries for Linux
sqlite-tools-linux-x86-3410200.zip (2.16 MiB) A bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff program, and the sqlite3_analyzer program. (SHA3-256: c41d2d97b62af6c6bd523ce9a06d1ea9f4e22e649f7a677a1c43f36d5ac3272c)
Precompiled Binaries for Mac OS X (x86)
sqlite-tools-osx-x86-3410200.zip (1.54 MiB) A bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff program, and the sqlite3_analyzer program. (SHA3-256: 0e0f1eeade6ff7bc6f0ddbf0fe86798a595494328e1afa15da192e5fc516cb29)
Precompiled Binaries for Windows
sqlite-dll-win32-x86-3410200.zip (563.08 KiB) 32-bit DLL (x86) for SQLite version 3.41.2. (SHA3-256: 2a3b03595ee7c3b172465d9be3990fd8ac10fa93b9afd22031b67fa42c2325e1)
sqlite-dll-win64-x64-3410200.zip (1.15 MiB) 64-bit DLL (x64) for SQLite version 3.41.2. (SHA3-256: a85aa4c03f394147b4a9050245fb8d5d0b00ae726d1f259e71c07055e44854cf)
sqlite-tools-win32-x86-3410200.zip (1.91 MiB) A bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff.exe program, and the sqlite3_analyzer.exe program. (SHA3-256: 0ceebb7f8378707d6d6b0737ecdf2ba02253a3b44b1009400f86273719d98f1f)
Precompiled Binaries for .NET
System.Data.SQLite Visit the System.Data.SQLite.org website and especially the download page for source code and binaries of SQLite for .NET.
WebAssembly & JavaScript
sqlite-wasm-3410200.zip (702.24 KiB) A precompiled bundle of sqlite3.wasm and its JavaScript APIs, ready for use in web applications. (SHA3-256: 1713c3a4c25b09f5d52ed25fb81948f6081e9eeefb5a6d449aead1cd848e962c)
Alternative Source Code Formats
sqlite-src-3410200.zip (13.20 MiB) Snapshot of the complete (raw) source tree for SQLite version 3.41.2. See How To Compile SQLite for usage details. (SHA3-256: 793e24c5158bafdb8a6e861ea9ad267cc773705d0bebb78b5c4aa323033e8028)
sqlite-preprocessed- Preprocessed C sources for SQLite version 3.41.2.
https://www.sqlite.org/2023/sqlite-tools-win32-x86-3410200.zip
Search
ENG IN
21:4
21-04-202

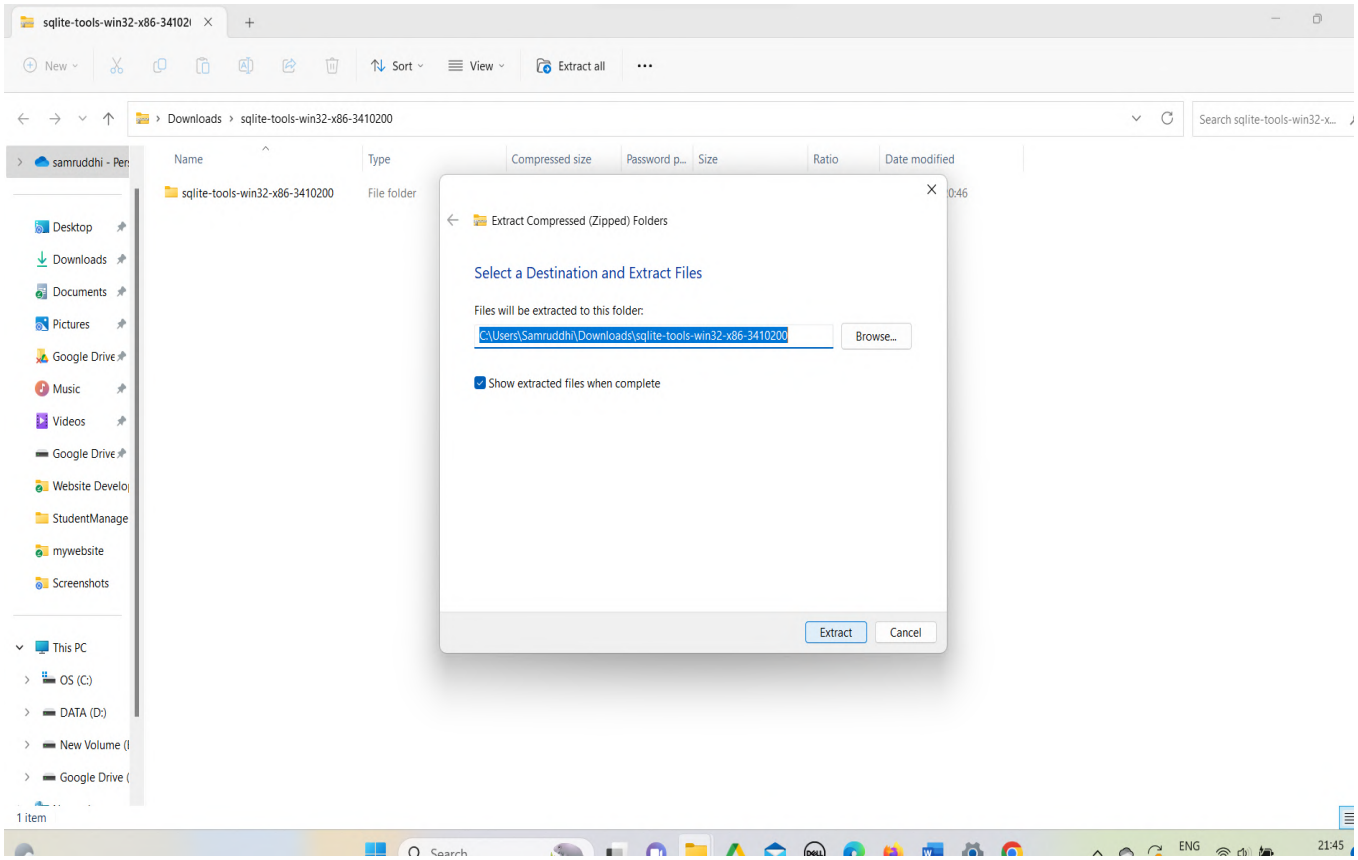
STEP 4:


sqlite - Yahoo India Search Result
SQLite Download Page
SQLite Home Page
https://www.sqlite.org/download.html
Gmail
YouTube
Maps
Inbox (197) - samru...
New Tab
(3.26 MiB) (SHA3-256: 6cd0ca943868978a945d4531bdc4e21fee8e3a7650e577f052550a37bc62db6e)
Precompiled Binaries for Linux
sqlite-tools-linux-x86-3410200.zip (2.16 MiB) A bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff program, and the sqlite3_analyzer program. (SHA3-256: c41d2d97b62af6c6bd523ce9a06d1ea9f4e22e649f7a677a1c43f36d5ac3272c)
Precompiled Binaries for Mac OS X (x86)
sqlite-tools-osx-x86-3410200.zip (1.54 MiB) A bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff program, and the sqlite3_analyzer program. (SHA3-256: 0e0f1eeade6ff7bc6f0ddbf0fe86798a595494328e1afa15da192e5fc516cb29)
Precompiled Binaries for Windows
sqlite-dll-win32-x86-3410200.zip (563.08 KiB) 32-bit DLL (x86) for SQLite version 3.41.2. (SHA3-256: 2a3b03595ee7c3b172465d9be3990fd8ac10fa93b9afd22031b67fa42c2325e1)
sqlite-dll-win64-x64-3410200.zip (1.15 MiB) 64-bit DLL (x64) for SQLite version 3.41.2. (SHA3-256: a85aa4c03f394147b4a9050245fb8d5d0b00ae726d1f259e71c07055e44854cf)
sqlite-tools-win32-x86-3410200.zip (1.91 MiB) A bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff.exe program, and the sqlite3_analyzer.exe program. (SHA3-256: 0ceebb7f8378707d6d6b0737ecdf2ba02253a3b44b1009400f86273719d98f1f)
Precompiled Binaries for .NET
System.Data.SQLite Visit the System.Data.SQLite.org website and especially the download page for source code and binaries of SQLite for .NET.
WebAssembly & JavaScript
sqlite-wasm-3410200.zip (702.24 KiB) A precompiled bundle of sqlite3.wasm and its JavaScript APIs, ready for use in web applications. (SHA3-256: 1713c3a4c25b09f5d52ed25fb81948f6081e9eeefb5a6d449aead1cd848e962c)
Alternative Source Code Formats
sqlite-src-3410200.zip (13.20 MiB) Snapshot of the complete (raw) source tree for SQLite version 3.41.2. See How To Compile SQLite for usage details. (SHA3-256: 793e24c5158bafdb8a6e861ea9ad267cc773705d0bebb78b5c4aa323033e8028)
sqlite-tools-win32-....zip
Show all
Search
ENG IN
21:44
21-04-2023

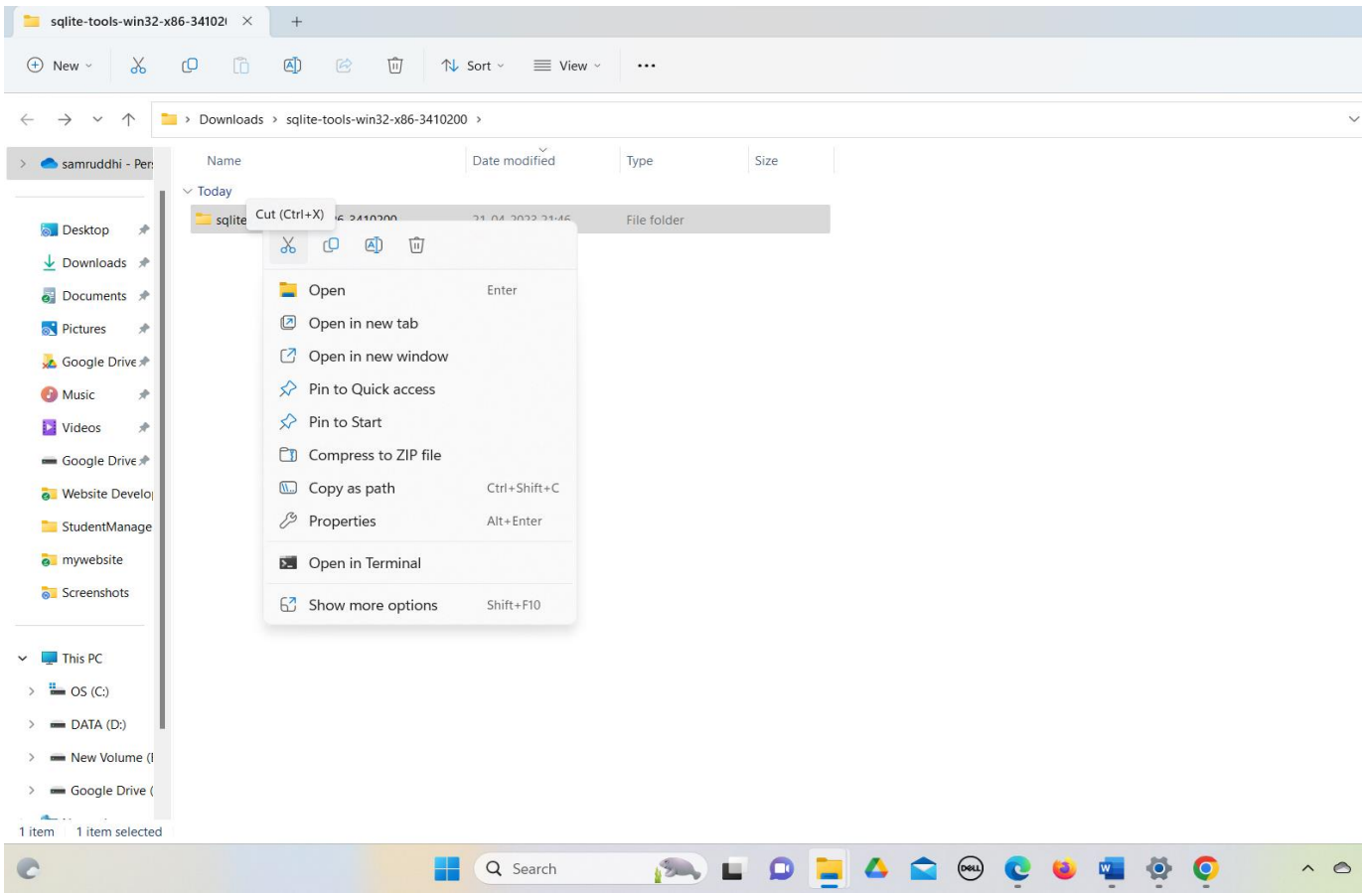
STEP 5:


sqlite-tools-win32-x86-34102
New
Sort
View
Extract all
Downloads › sqlite-tools-win32-x86-3410200 ›
Search sqlite-tools-win32-x...
Name
Type
Compressed size
Password p...
Size
Ratio
Date modified
sqlite-tools-win32-x86-3410200
22-03-2023 20:46
View
Sort by
Group by
Extract All...
Undo Delete
Ctrl+Z
Open in Terminal
Show more options
Shift+F10
Desktop
Downloads
Documents
Pictures
Google Drive
Music
Videos
Google Drive
Website Develo
StudentManage
mywebsite
Screenshots
This PC
OS (C:)
DATA (D:)
New Volume (I
Google Drive (
1 item
Search
ENG
IN
21:45
21-04-2023

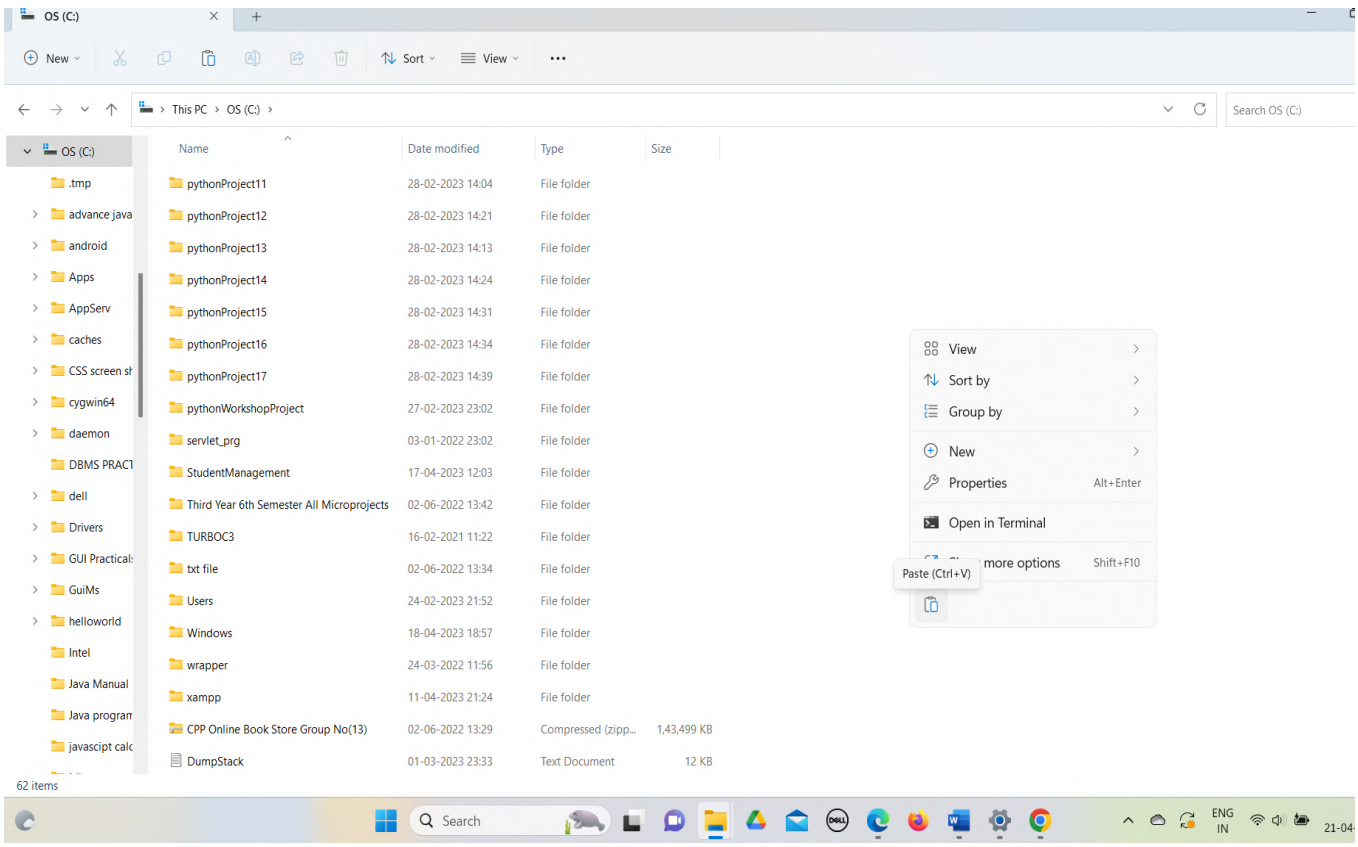
STEP 6:


sqlite-tools-win32-x86-34102
New
Sort
View
Extract all
Downloads › sqlite-tools-win32-x86-3410200
Search sqlite-tools-win32-x...
Name
Type
Compressed size
Password p...
Size
Ratio
Date modified
sqlite-tools-win32-x86-3410200
File folder
Extract Compressed (Zipped) Folders
Select a Destination and Extract Files
Files will be extracted to this folder:
C:\Users\Samruddhi\Downloads\sqlite-tools-win32-x86-3410200
Browse...
Show extracted files when complete
Extract
Cancel
Desktop
Downloads
Documents
Pictures
Google Drive
Music
Videos
Google Drive
Website Develo
StudentManage
mywebsite
Screenshots
This PC
OS (C:)
DATA (D:)
New Volume (I
Google Drive (
1 item
ENG
21:45

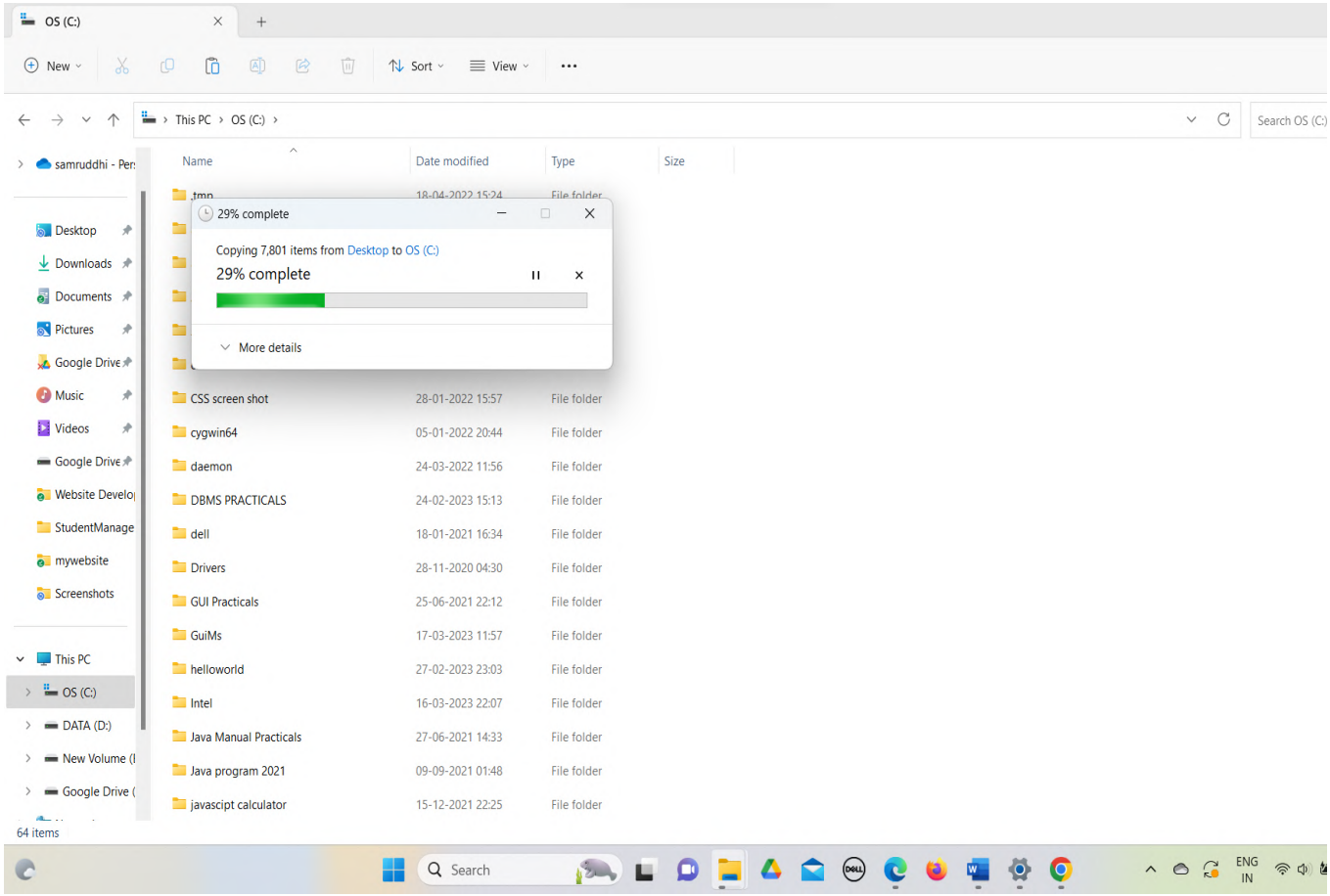
STEP 7:


sqlite-tools-win32-x86-341020
New
Sort
View
Downloads
sqlite-tools-win32-x86-3410200
Name
Date modified
Type
Size
Today
Cut (Ctrl+X)
File folder
Open
Enter
Open in new tab
Open in new window
Pin to Quick access
Pin to Start
Compress to ZIP file
Copy as path
Ctrl+Shift+C
Properties
Alt+Enter
Open in Terminal
Show more options
Shift+F10
Desktop
Downloads
Documents
Pictures
Google Drive
Music
Videos
Google Drive
Website Develo
StudentManage
mywebsite
Screenshots
This PC
OS (C:)
DATA (D:)
New Volume (I
Google Drive (
1 item
1 item selected
Search

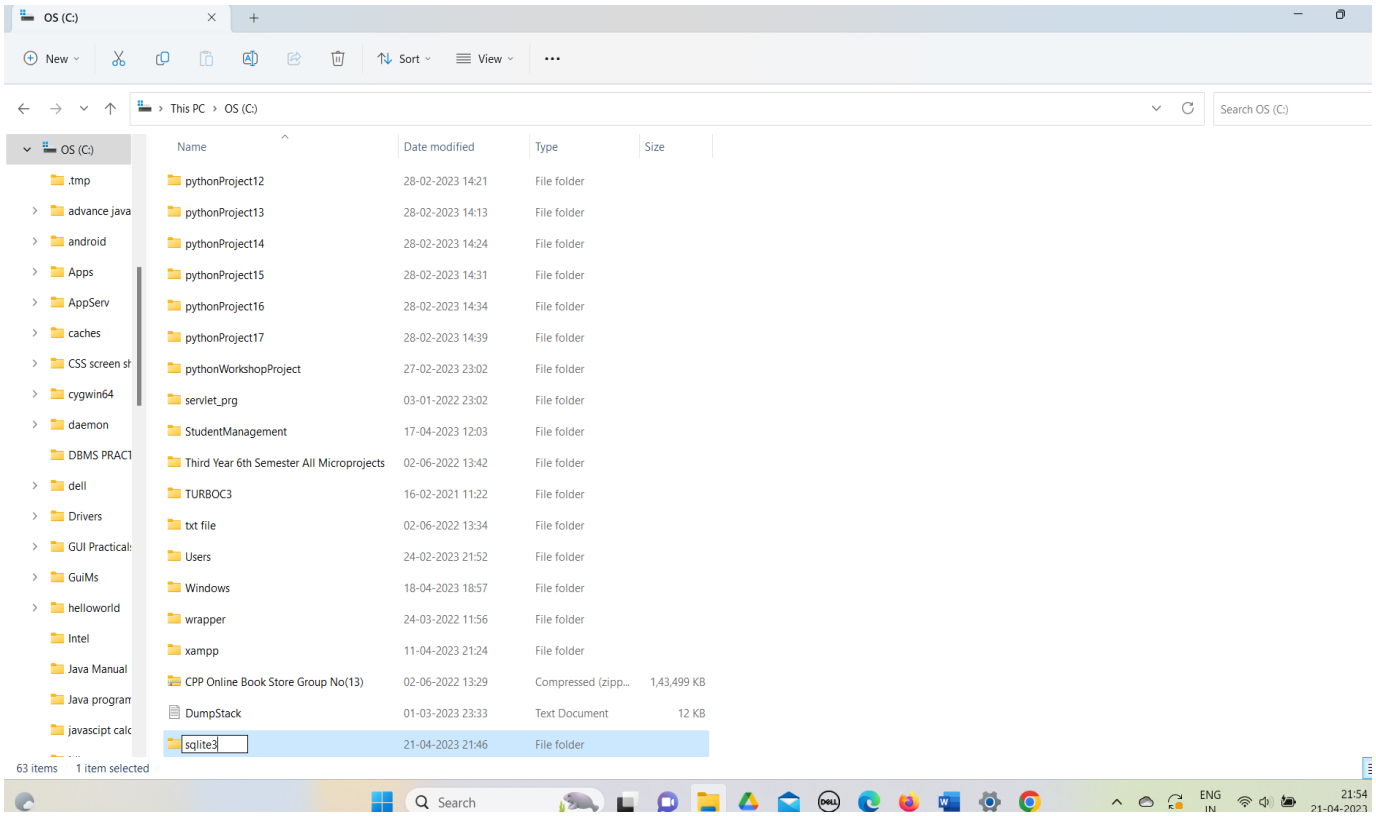
STEP 8:


OS (C:)
New
Sort
View
This PC
OS (C:)
Search OS (C:)
Name
Date modified
Type
Size
.tmp
advance java
android
Apps
AppServ
caches
CSS screen sh
cygwin64
daemon
DBMS PRACT
dell
Drivers
GUI Practicals
GuiMs
helloworld
Intel
Java Manual
Java program
javascipt calc
pythonProject11 28-02-2023 14:04 File folder
pythonProject12 28-02-2023 14:21 File folder
pythonProject13 28-02-2023 14:13 File folder
pythonProject14 28-02-2023 14:24 File folder
pythonProject15 28-02-2023 14:31 File folder
pythonProject16 28-02-2023 14:34 File folder
pythonProject17 28-02-2023 14:39 File folder
pythonWorkshopProject 27-02-2023 23:02 File folder
servlet_prg 03-01-2022 23:02 File folder
StudentManagement 17-04-2023 12:03 File folder
Third Year 6th Semester All Microprojects 02-06-2022 13:42 File folder
TURBOC3 16-02-2021 11:22 File folder
txt file 02-06-2022 13:34 File folder
Users 24-02-2023 21:52 File folder
Windows 18-04-2023 18:57 File folder
wrapper 24-03-2022 11:56 File folder
xampp 11-04-2023 21:24 File folder
CPP Online Book Store Group No(13) 02-06-2022 13:29 Compressed (zipp... 1,43,499 KB
DumpStack 01-03-2023 23:33 Text Document 12 KB
62 items
View
Sort by
Group by
New
Properties
Alt+Enter
Open in Terminal
more options
Shift+F10
Paste (Ctrl+V)
Search
ENG
IN

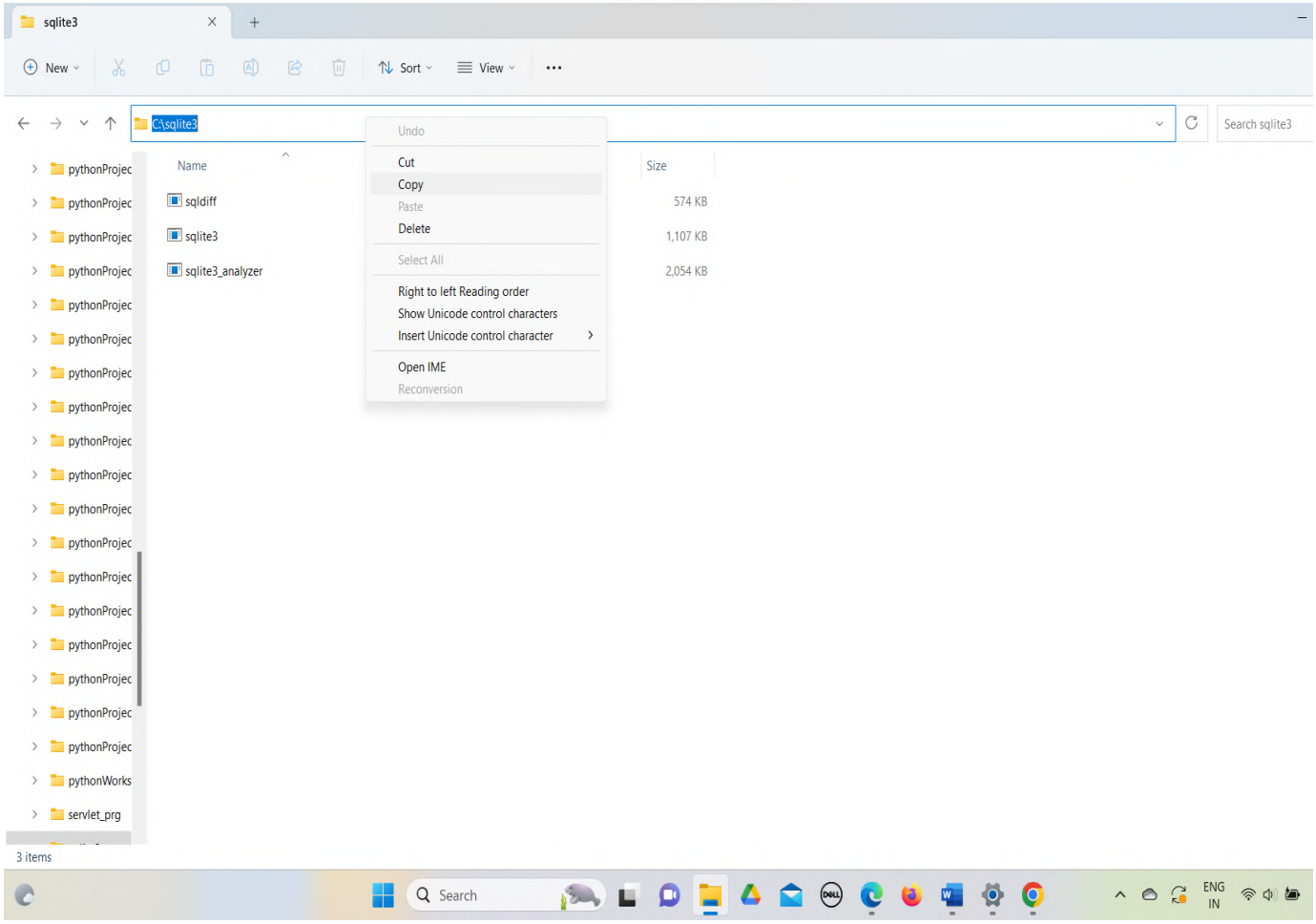
STEP 9:



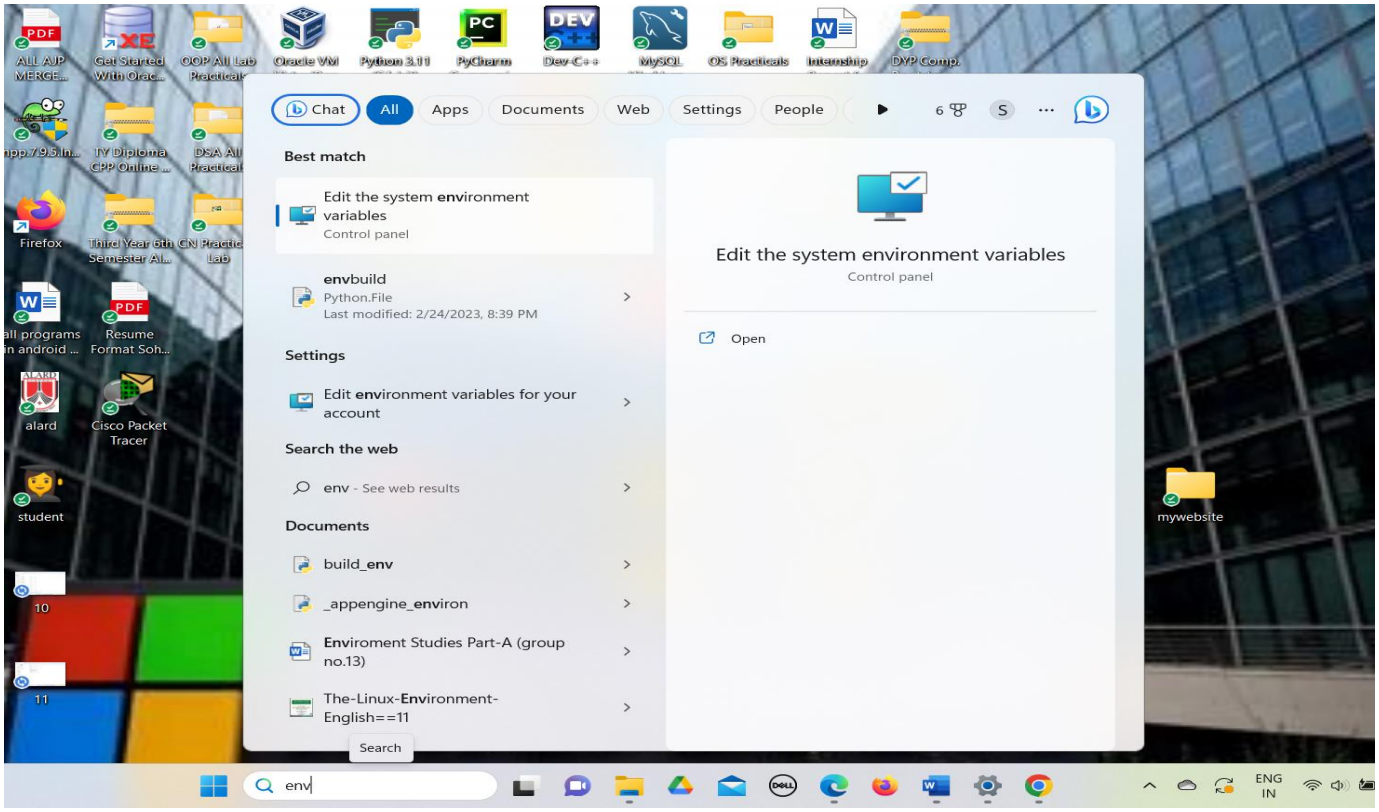
STEP 10:



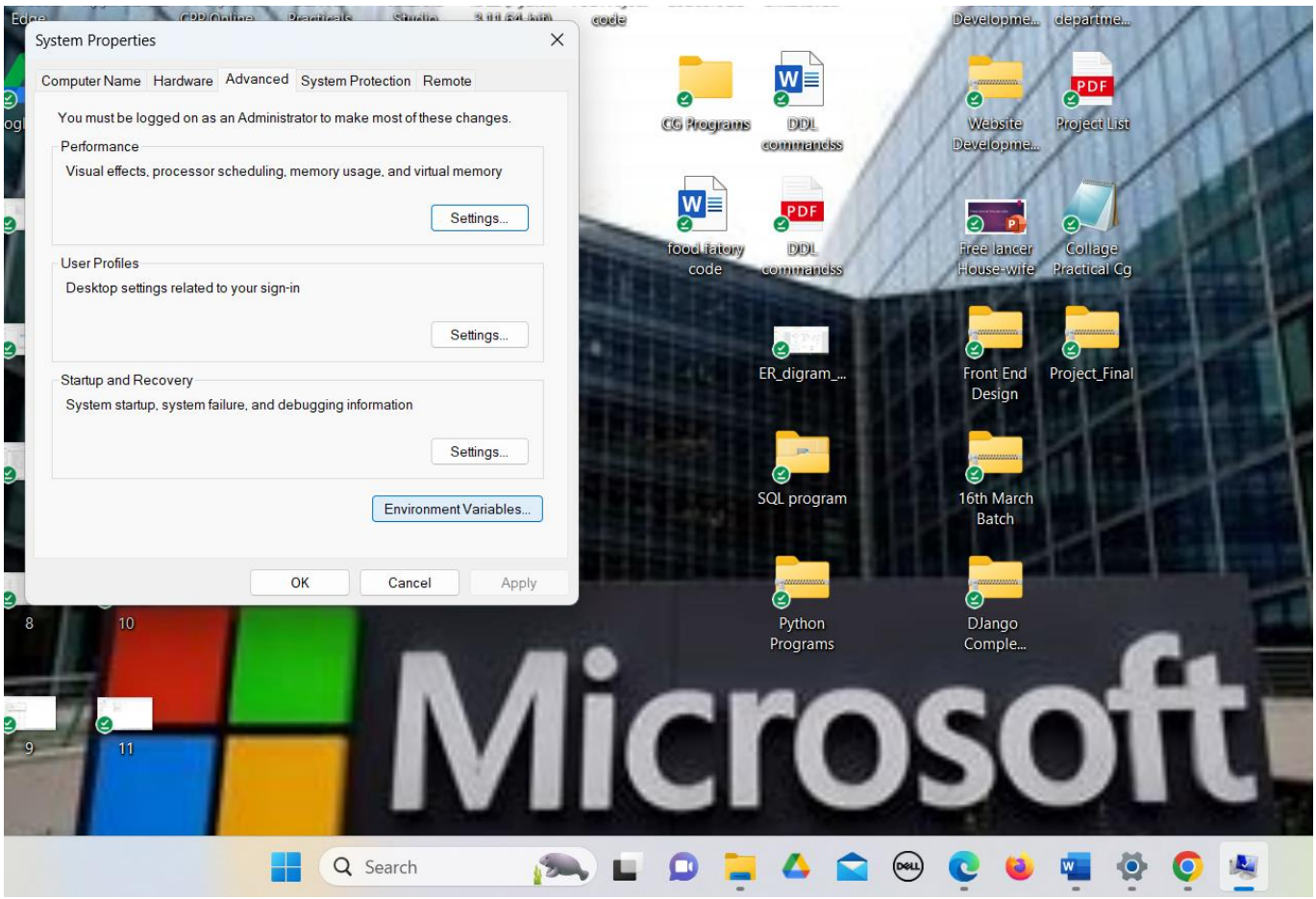
STEP 11:



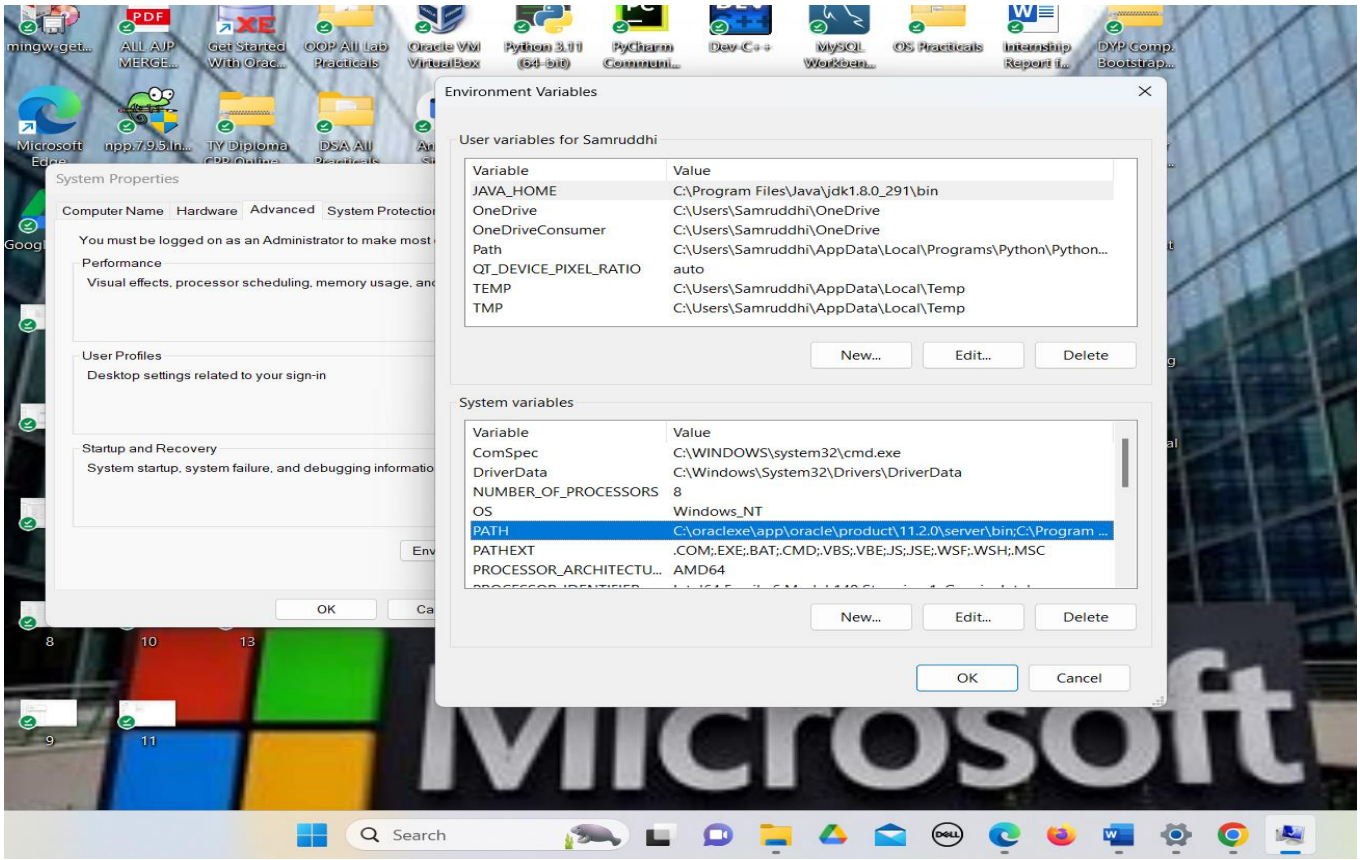
STEP 12:



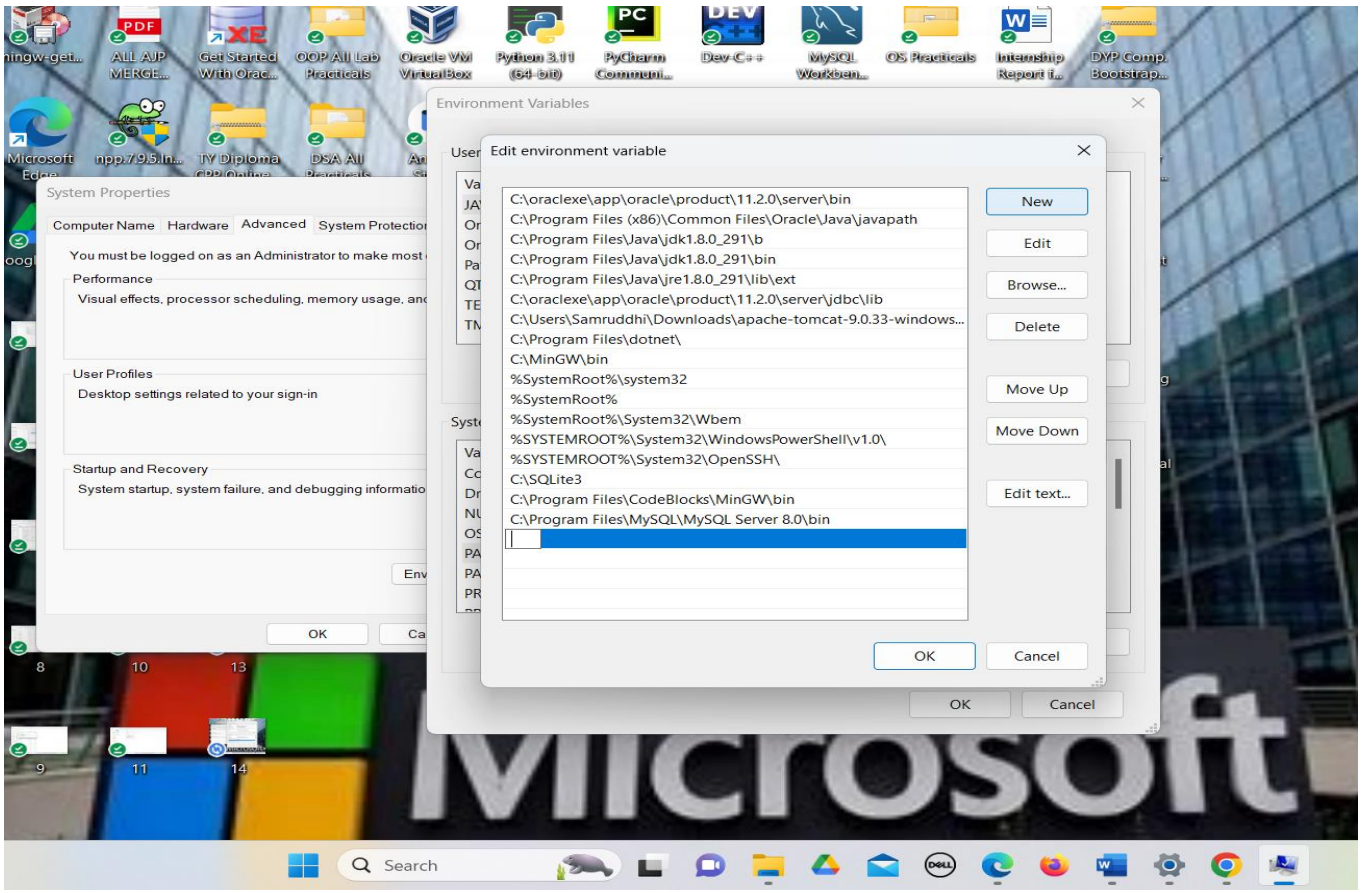
STEP 13:



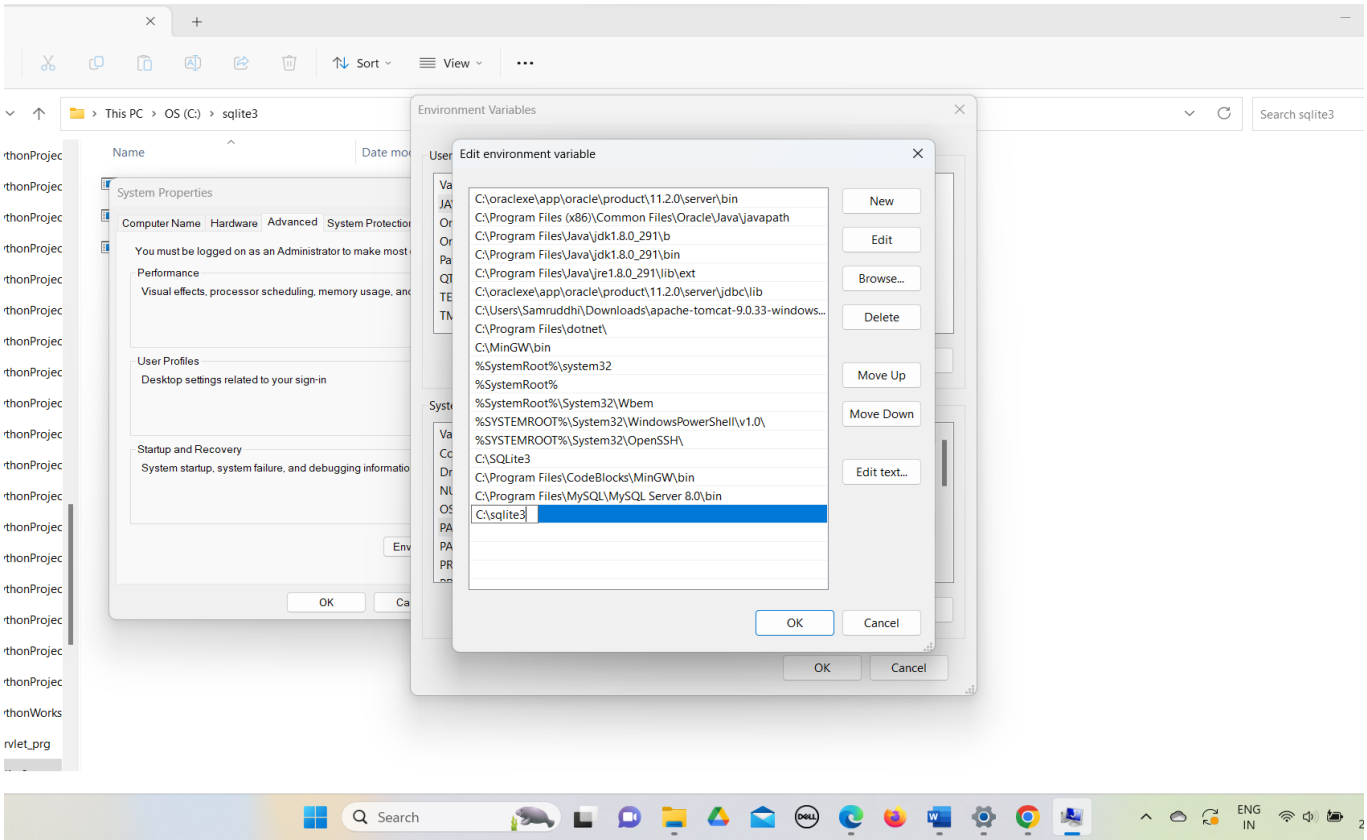
STEP 14:



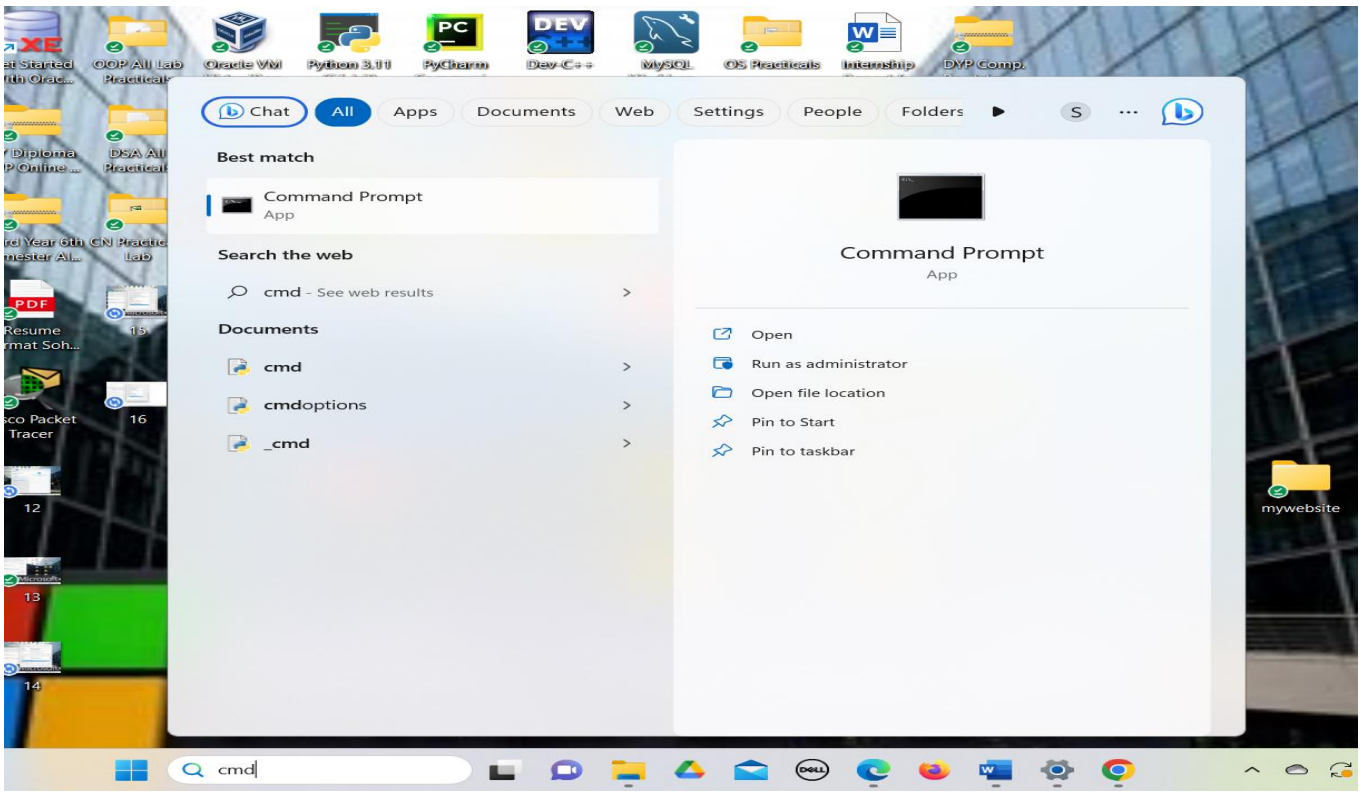
STEP 15:



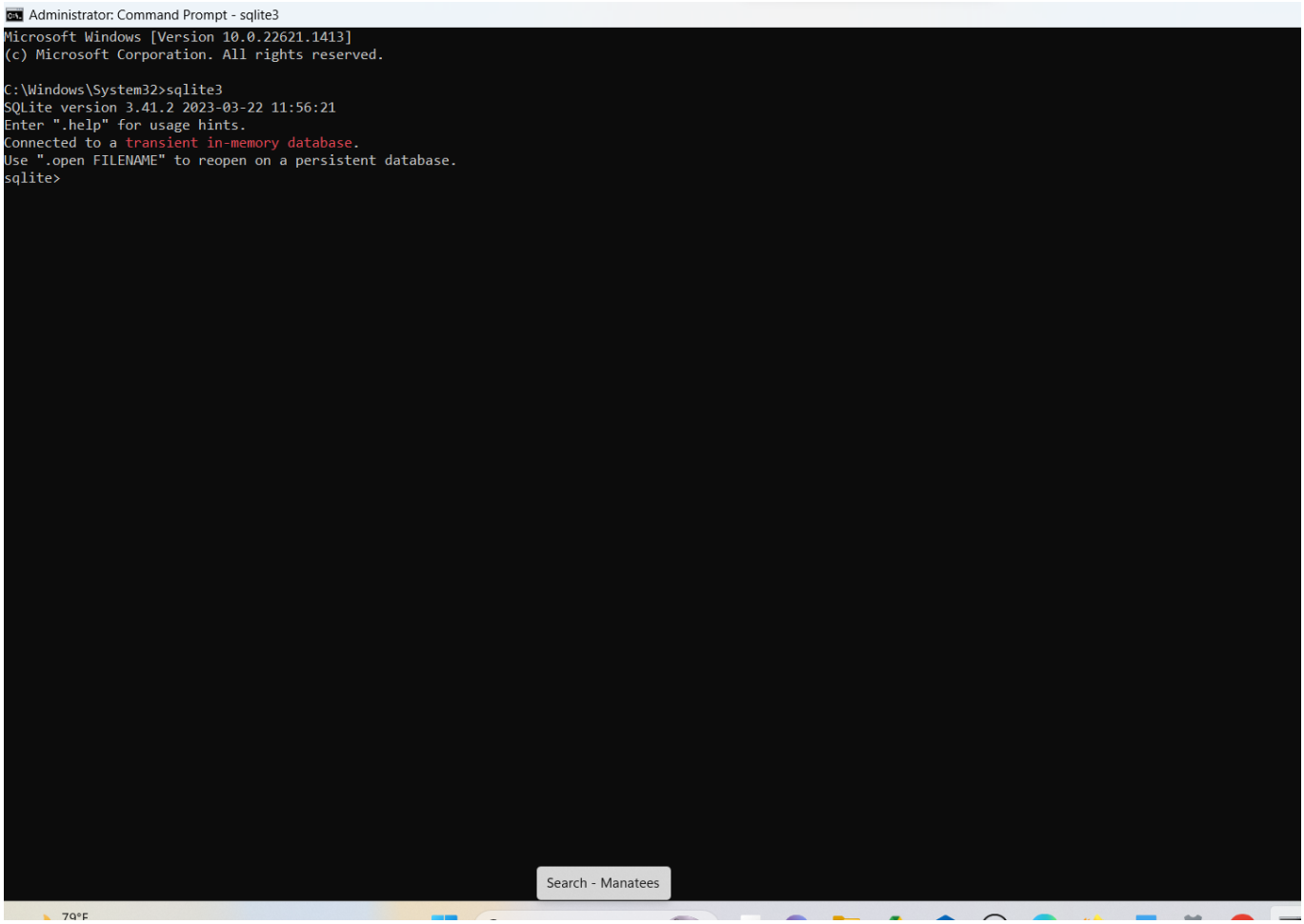
STEP 16:



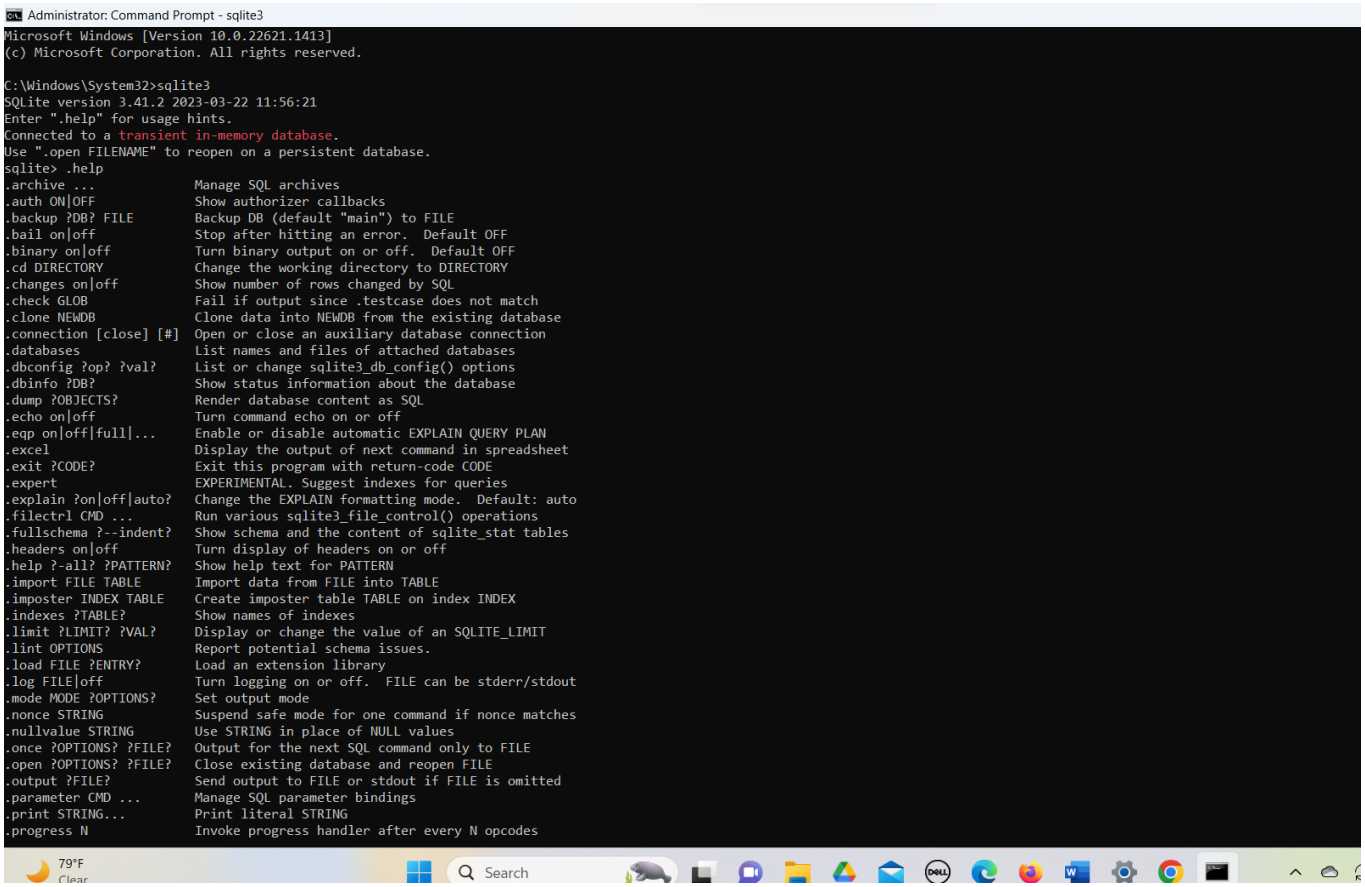
STEP 17:



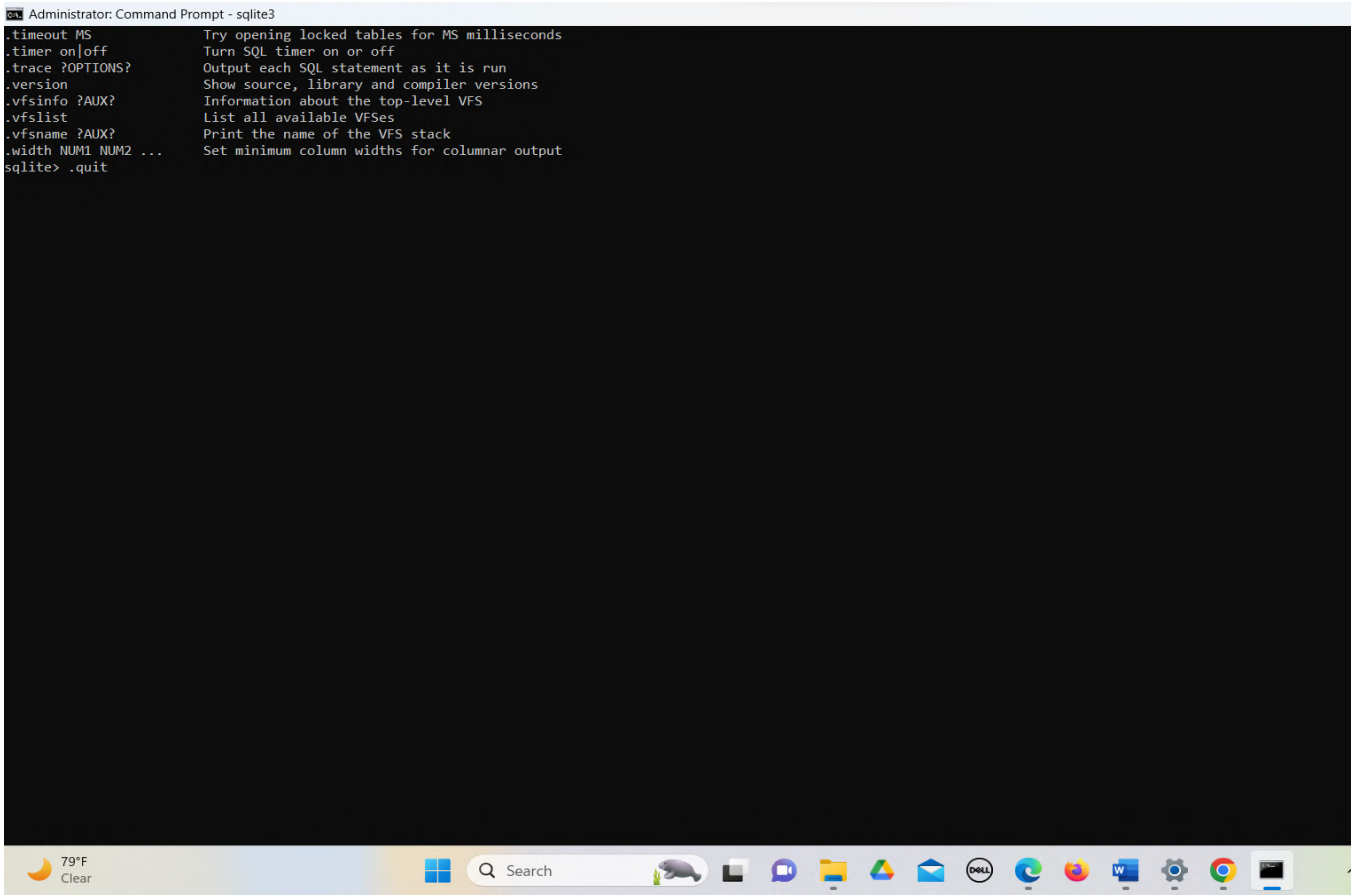
STEP 18:



STEP 19:

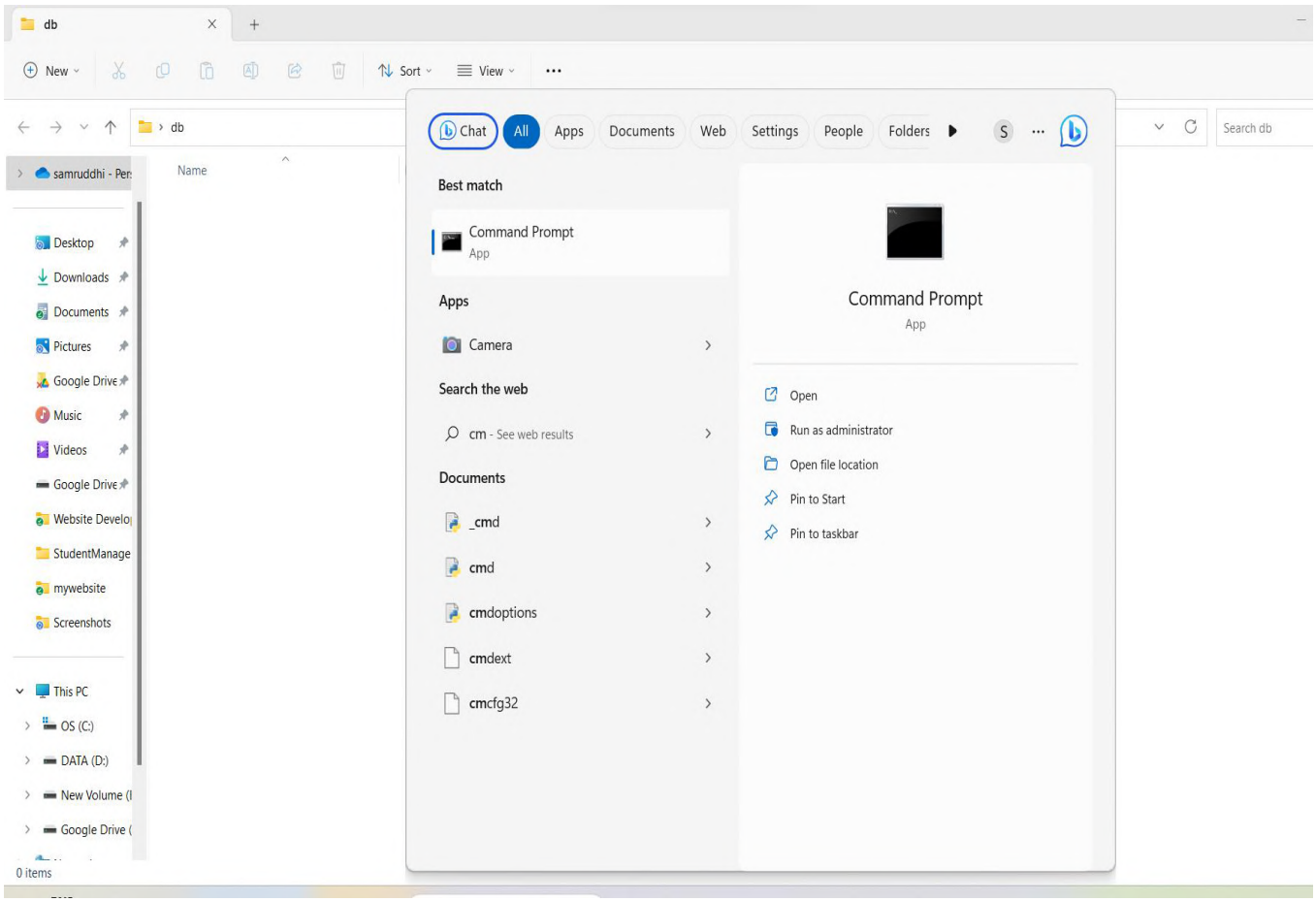


STEP 20:

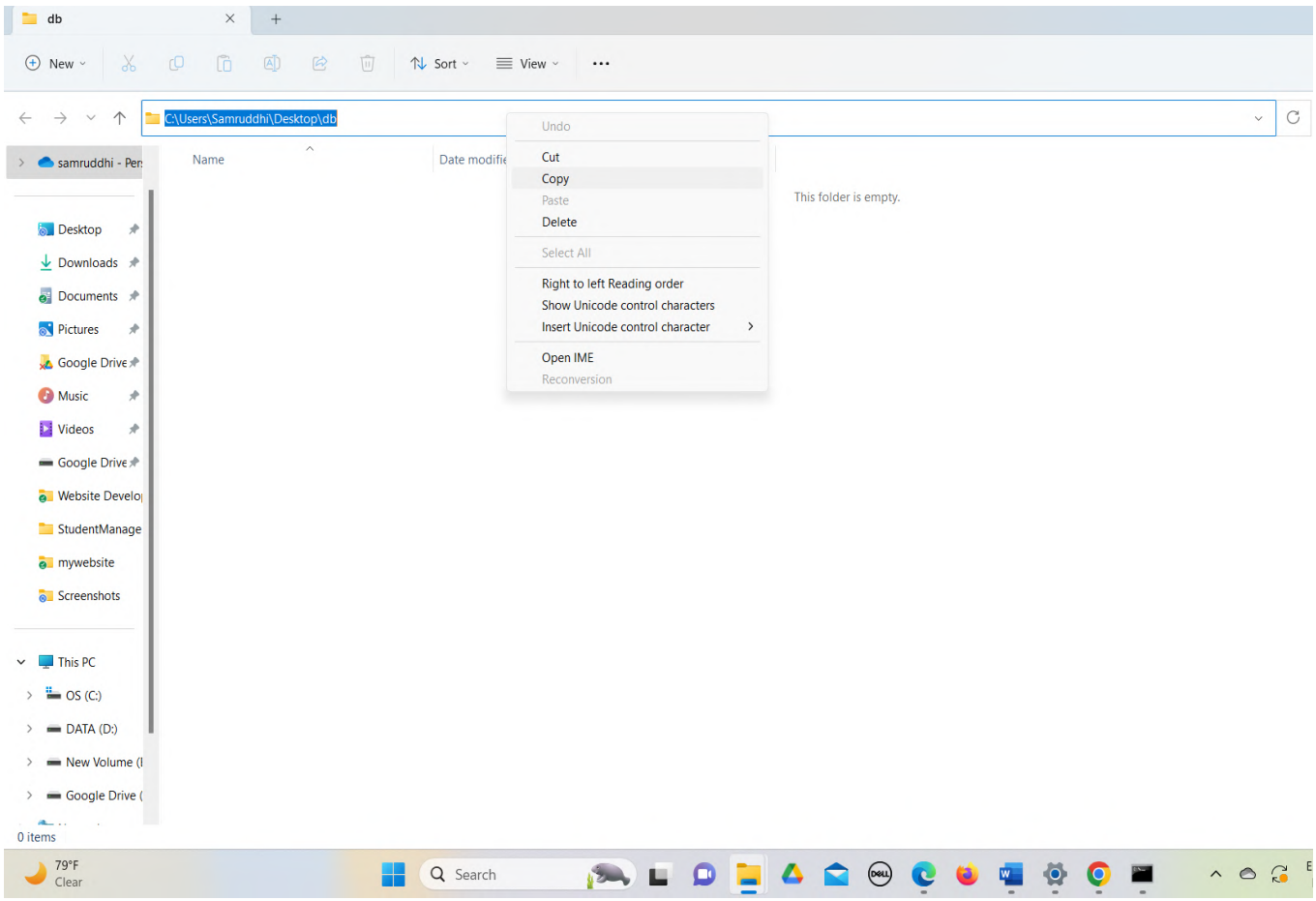


STEP 21:

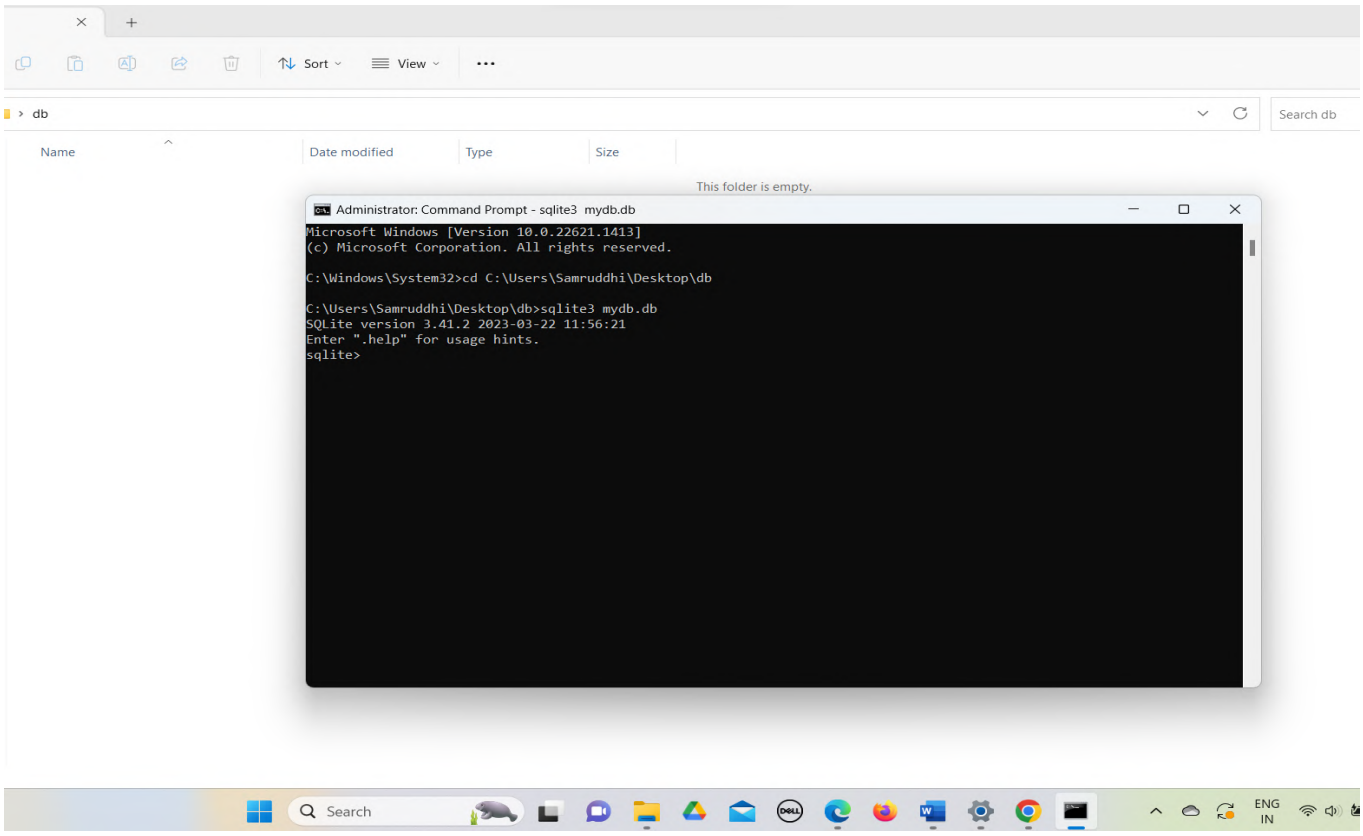
STEP 22:



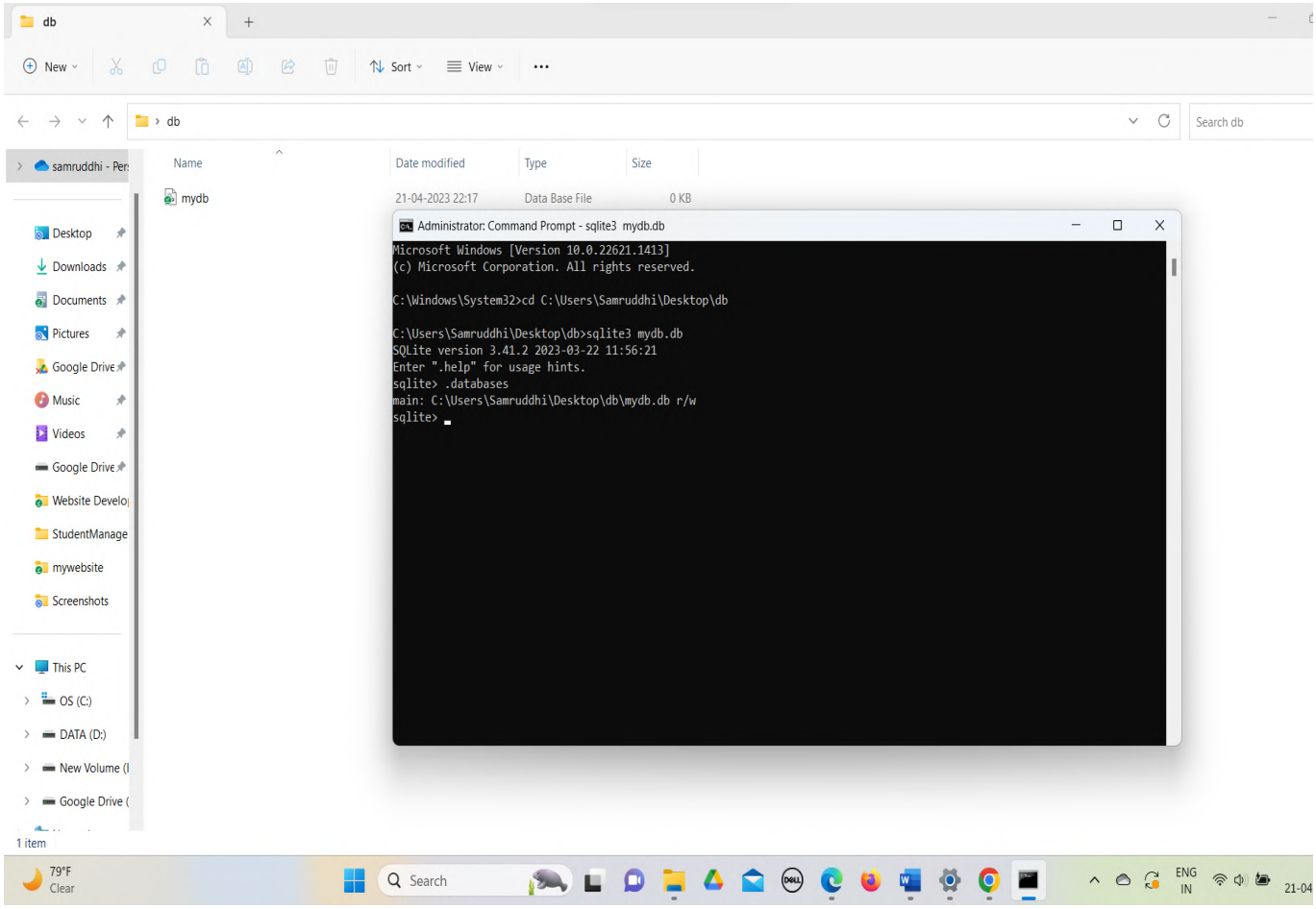
STEP 23:



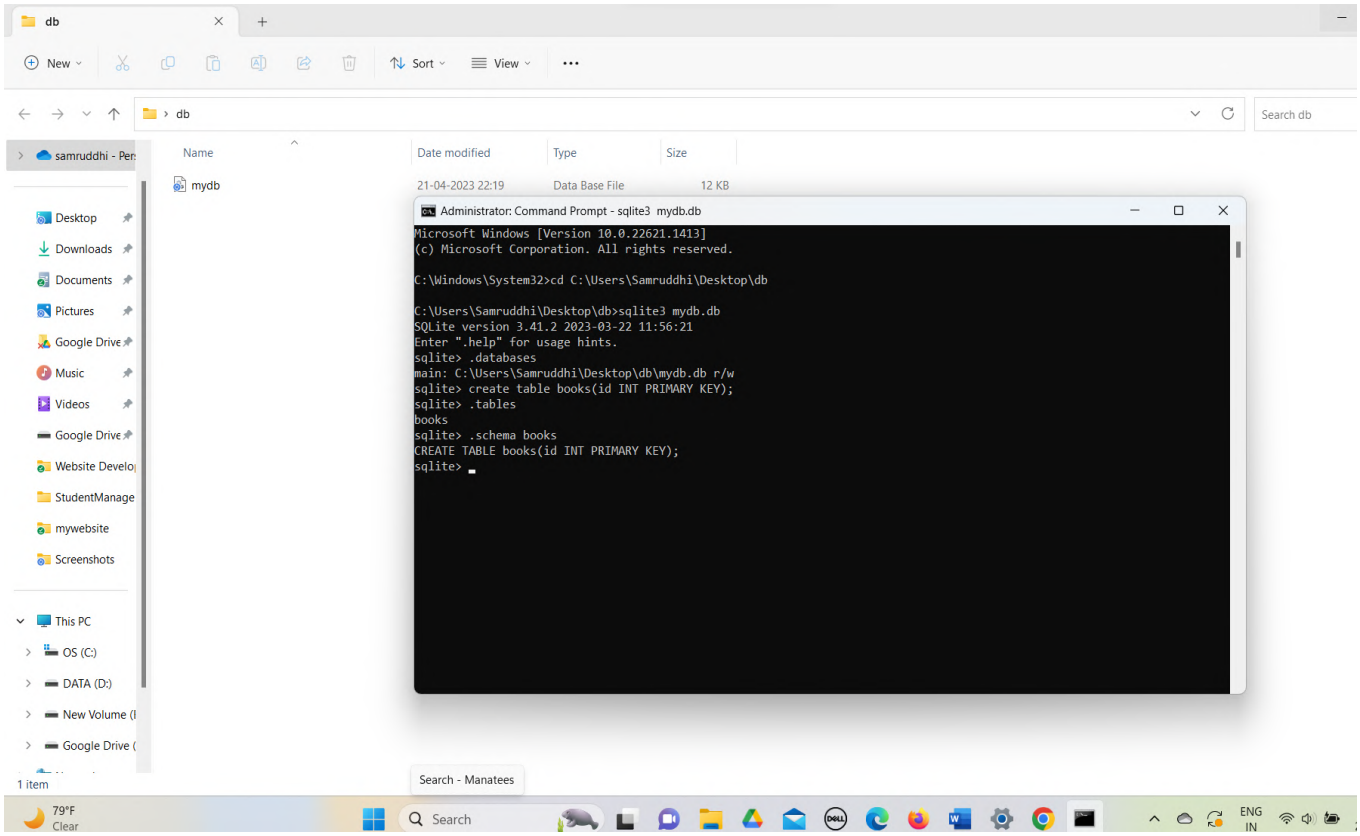
STEP 24:



STEP 25:

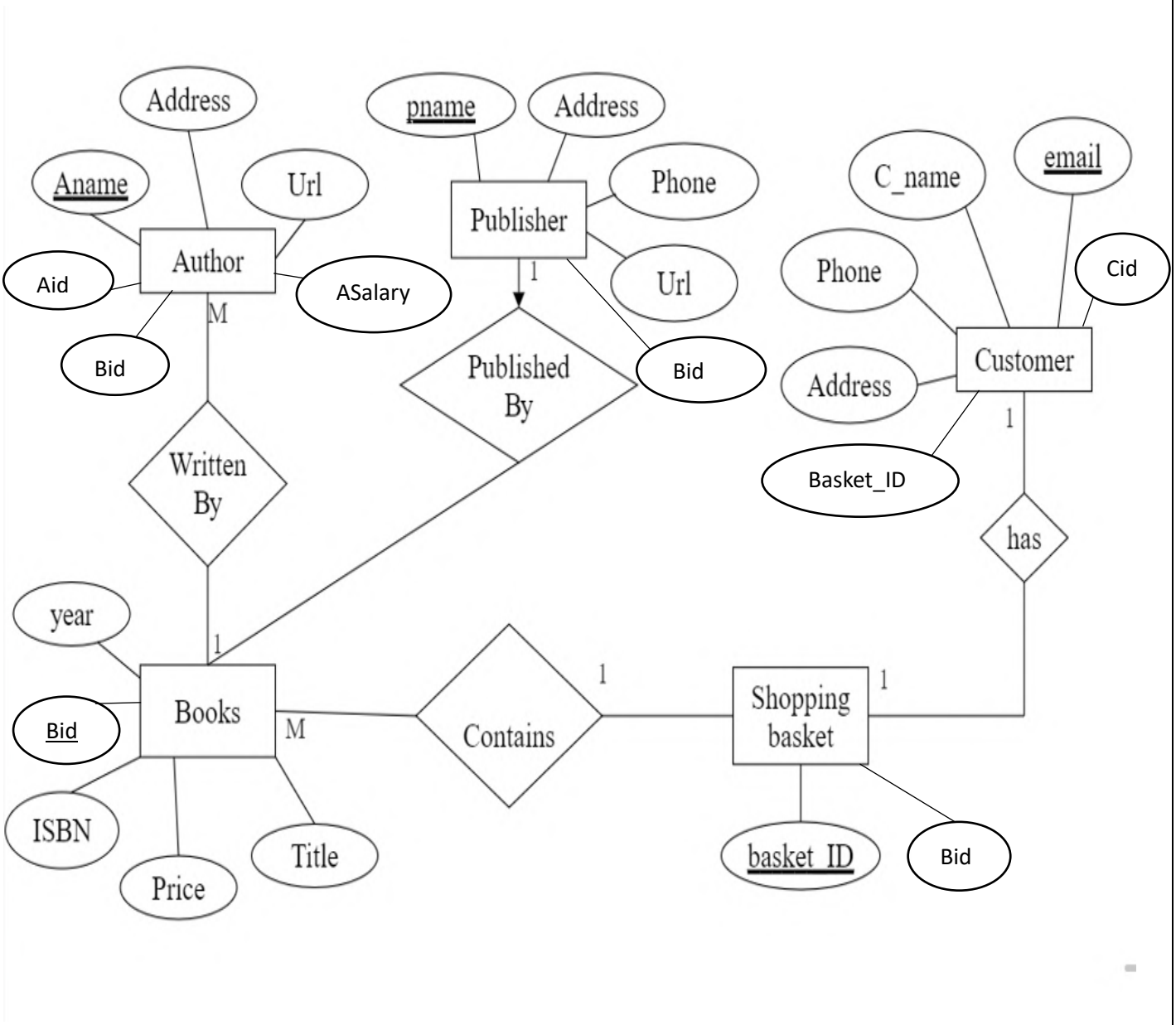


STEP 26:



Group B: MySQL

1.Design any database with at least 3 entities and relationships between them. Draw suitable ER/EER diagram for the system.



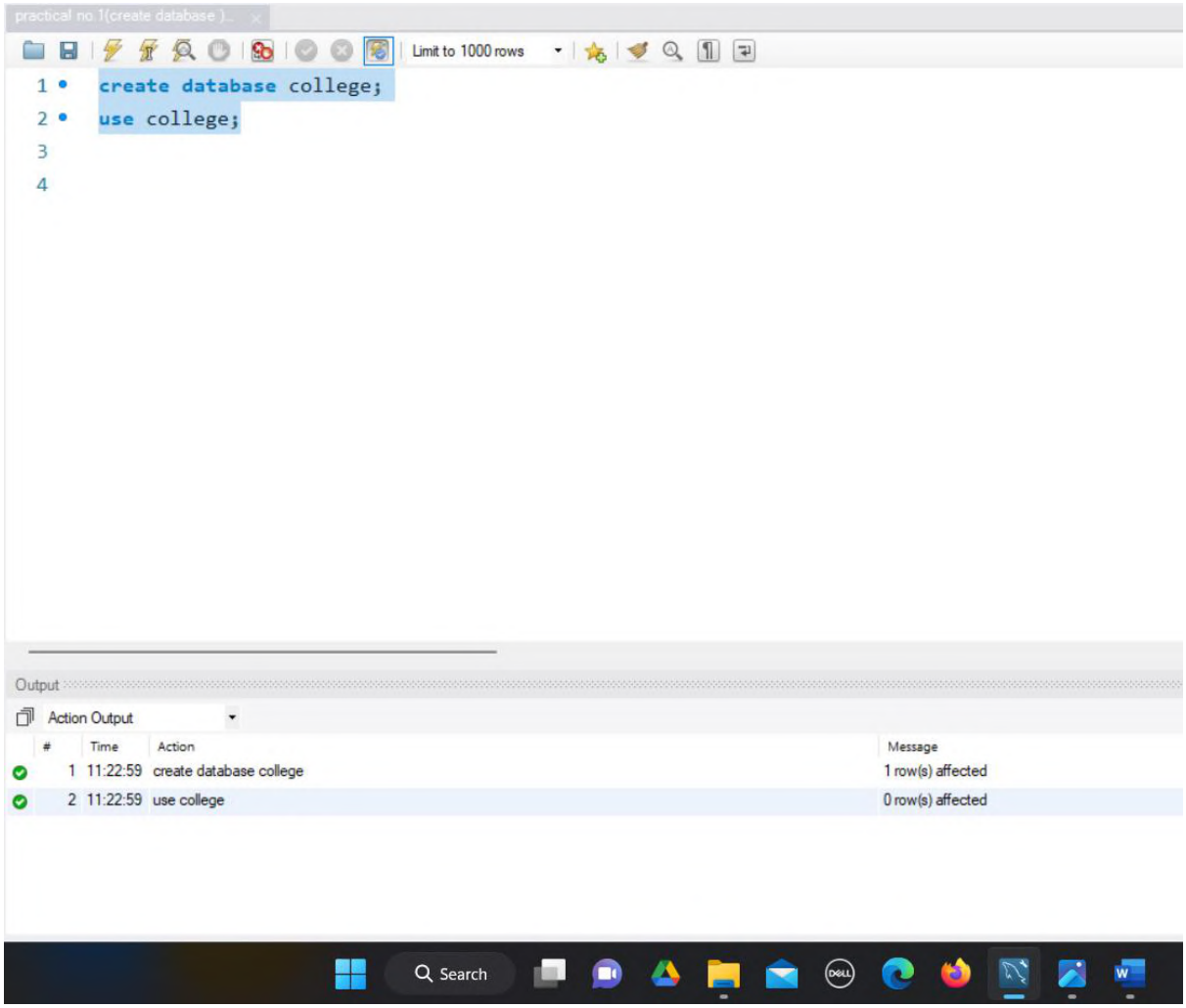
Group B: MySQL

2.Design and implement a database (for assignment no 1) using DDL statements and apply normalization on them.

Create Database:

create database college;

use college;



Create table:

create table college_info(clg_id int ,college_name varchar(10),address varchar(10),

Branches varchar(10));

insert into college_info values(1,'GPA','Awasari','IT');

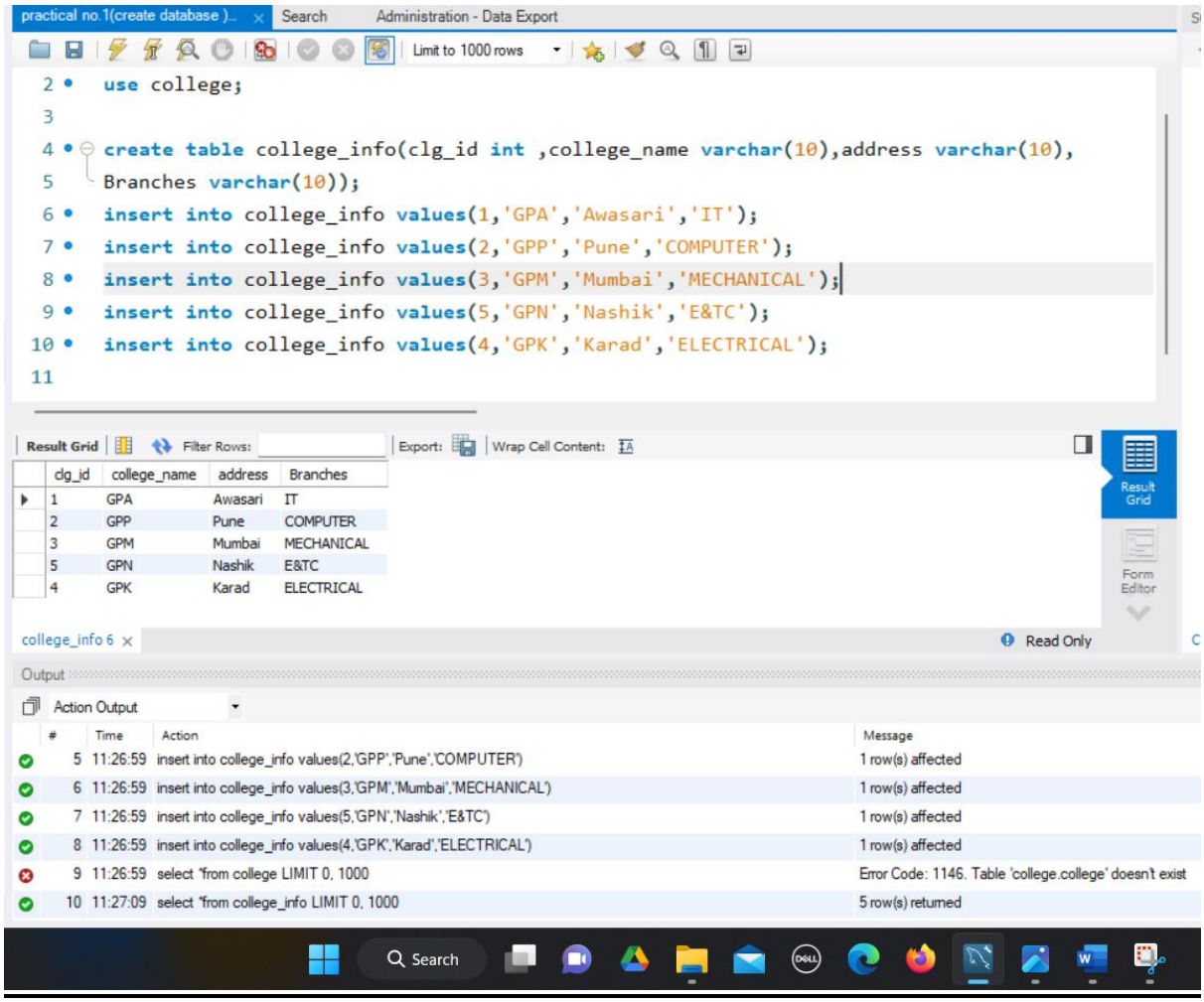
insert into college_info values(2,'GPP','Pune','COMPUTER');

insert into college_info values(3,'GPM','Mumbai','MECHANICAL');

insert into college_info values(5,'GPN','Nashik','E&TC');

insert into college_info values(4,'GPK','Karad','ELECTRICAL');

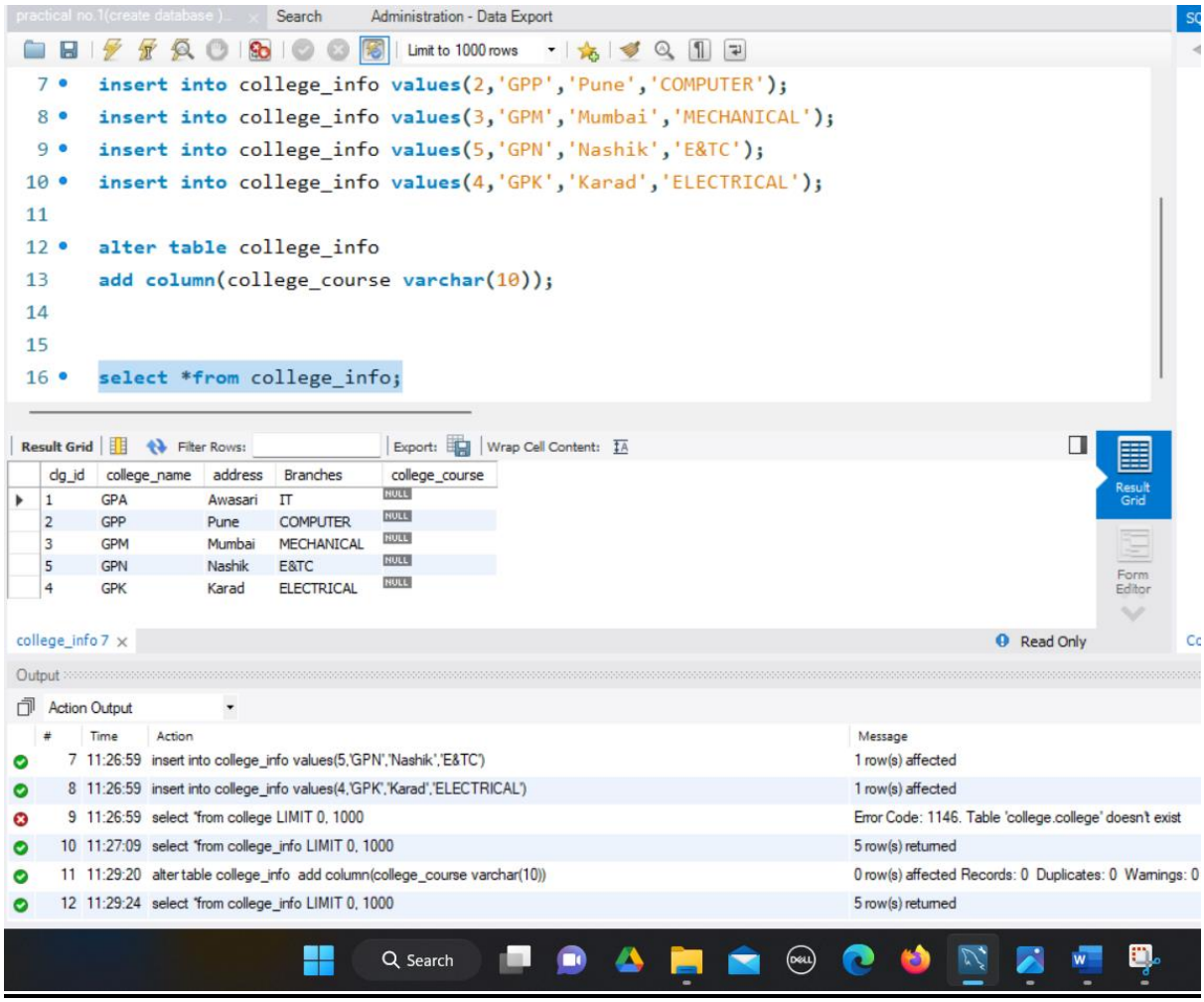
select *from college_info;



Alter Command:

alter table college_info

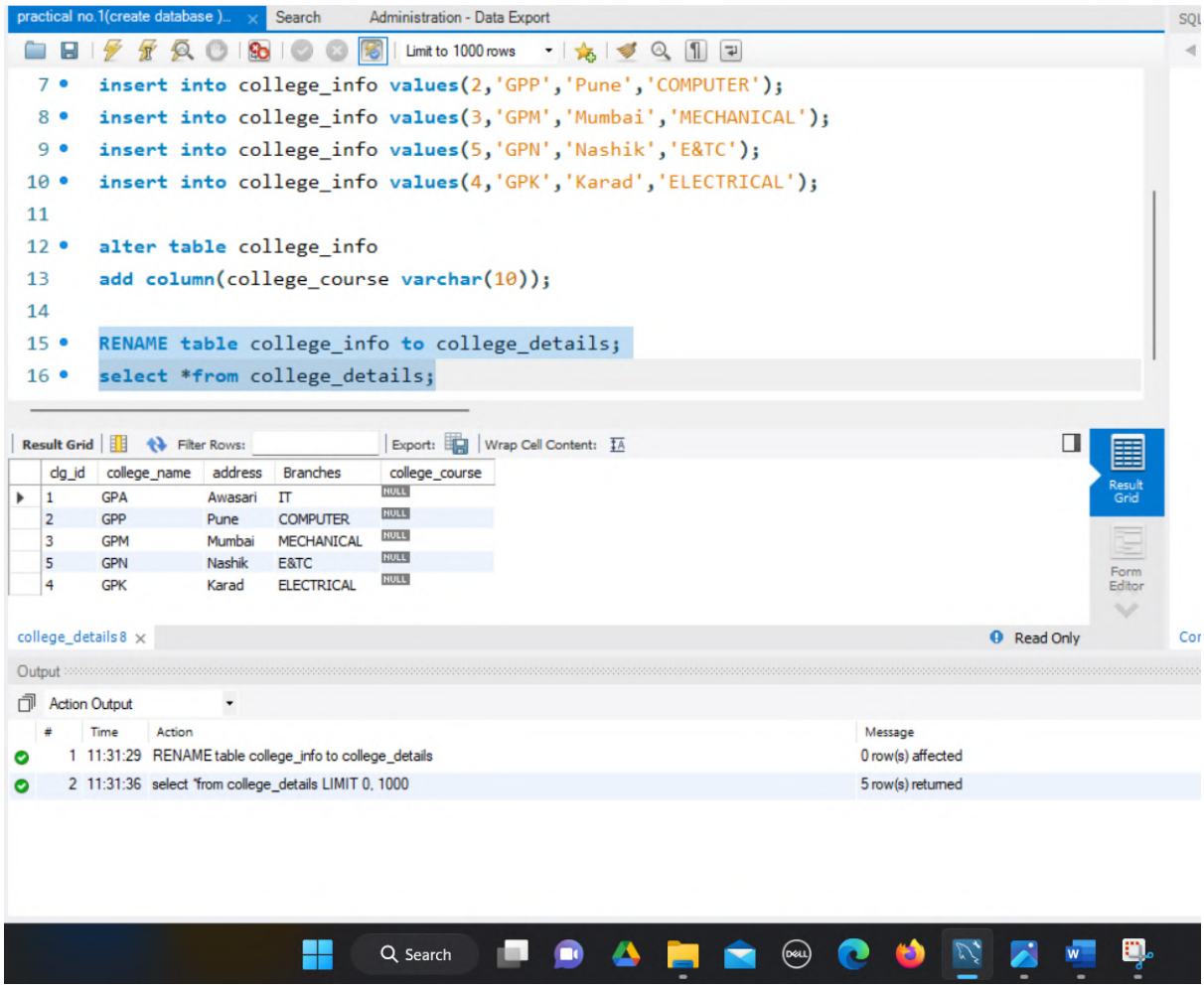
add column(college_course varchar(10));



Rename command:

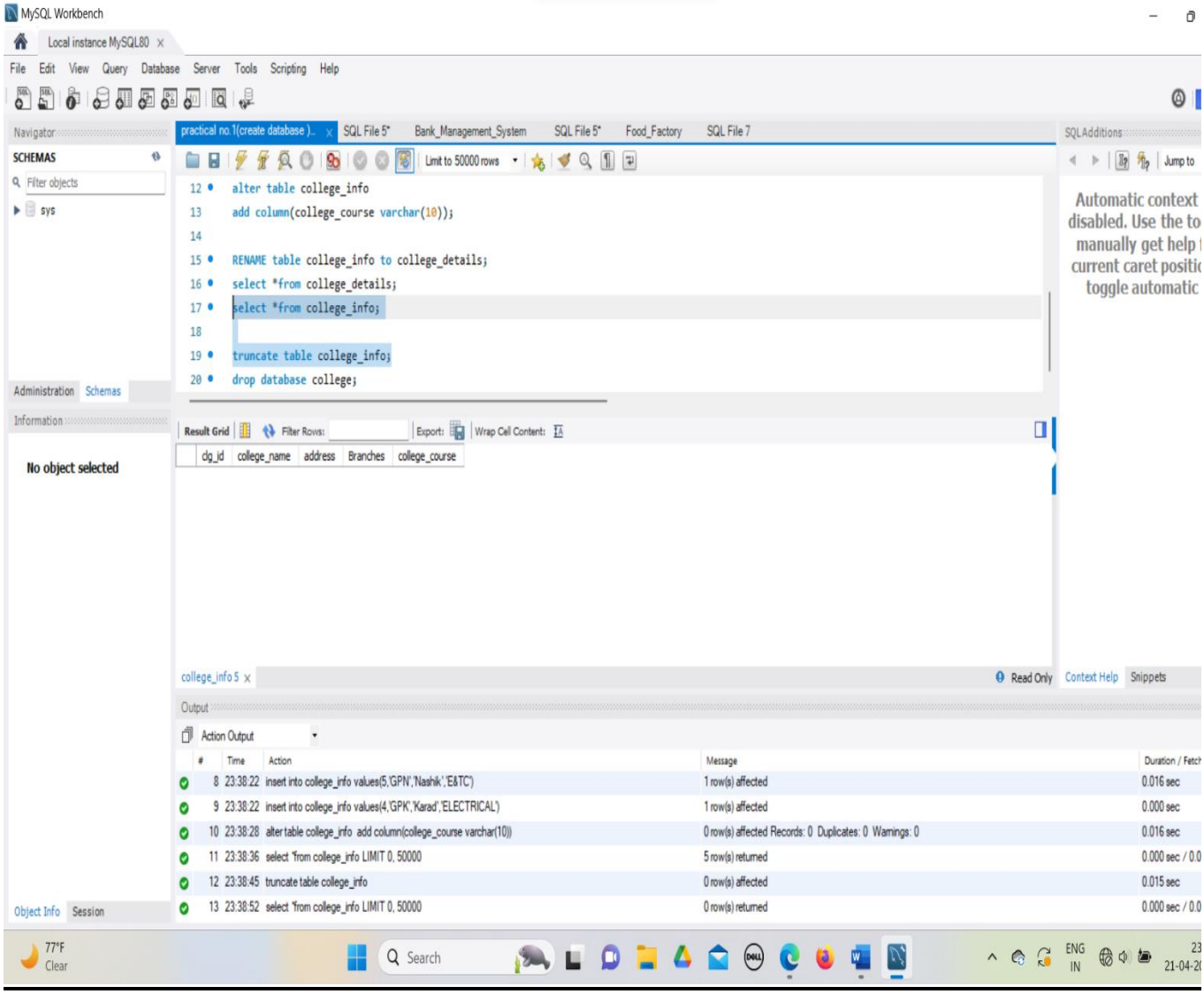
RENAME table college_info to college_details;

select *from college_details;



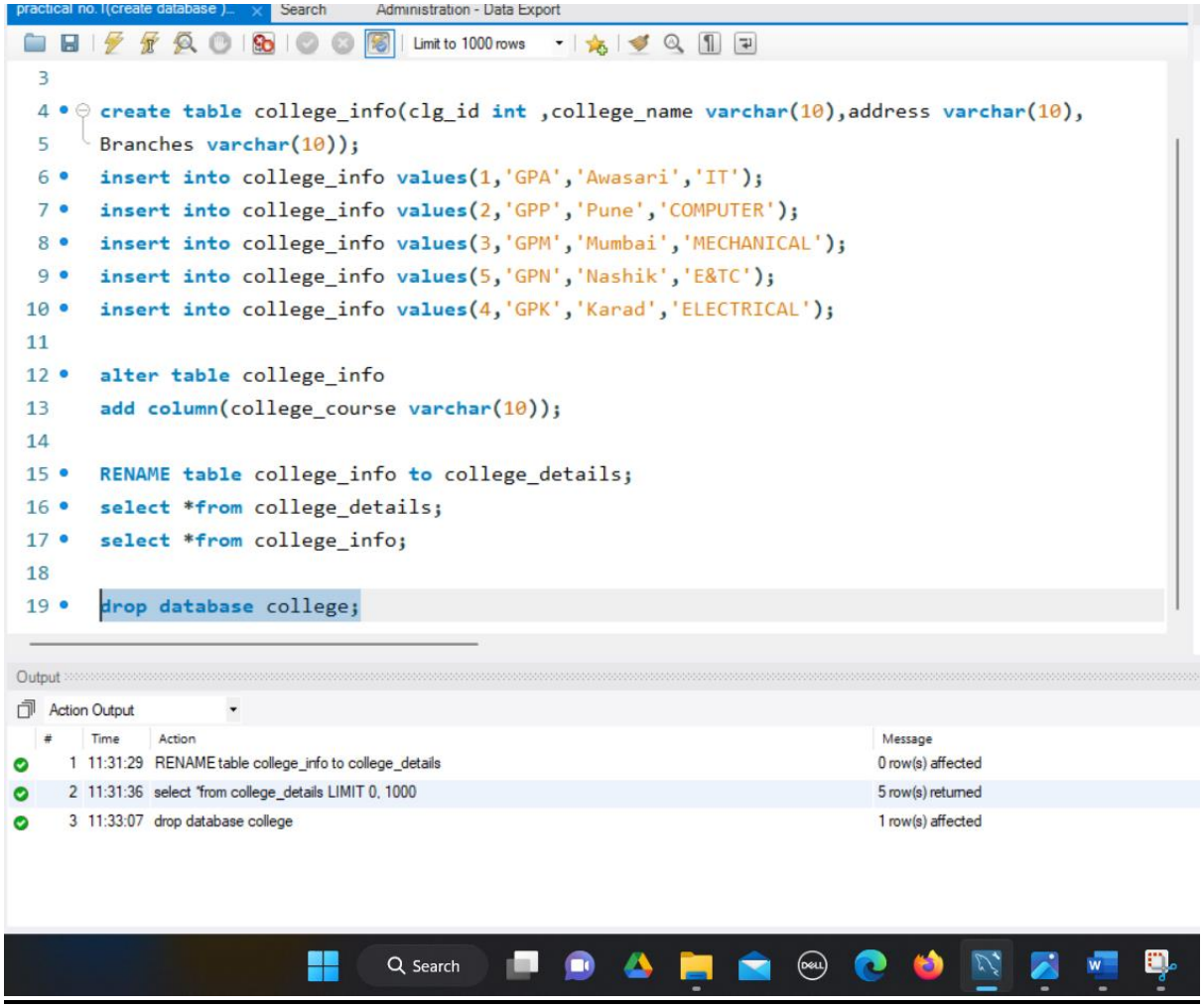
Truncate Command:

truncate table college_info;



Drop Command:

drop database college;



DDL QUERIES:

```
create database college;
use college;

create table college_info(clg_id int ,college_name varchar(10),address varchar(10),
Branches varchar(10));
insert into college_info values(1,'GPA','Awasari','IT');
insert into college_info values(2,'GPP','Pune','COMPUTER');
insert into college_info values(3,'GPM','Mumbai','MECHANICAL');
insert into college_info values(5,'GPN','Nashik','E&TC');
insert into college_info values(4,'GPK','Karad','ELECTRICAL');

alter table college_info
add column(college_course varchar(10));

RENAME table college_info to college_details;
select *from college_details;
select *from college_info;

truncate table college_info;
drop database college;
```

Group B: MySQL

3. Create Table with primary key and foreign key constraints.

a. Alter table with add n modify

b. Drop table

Creating tables with primary key and foreign key constraints.

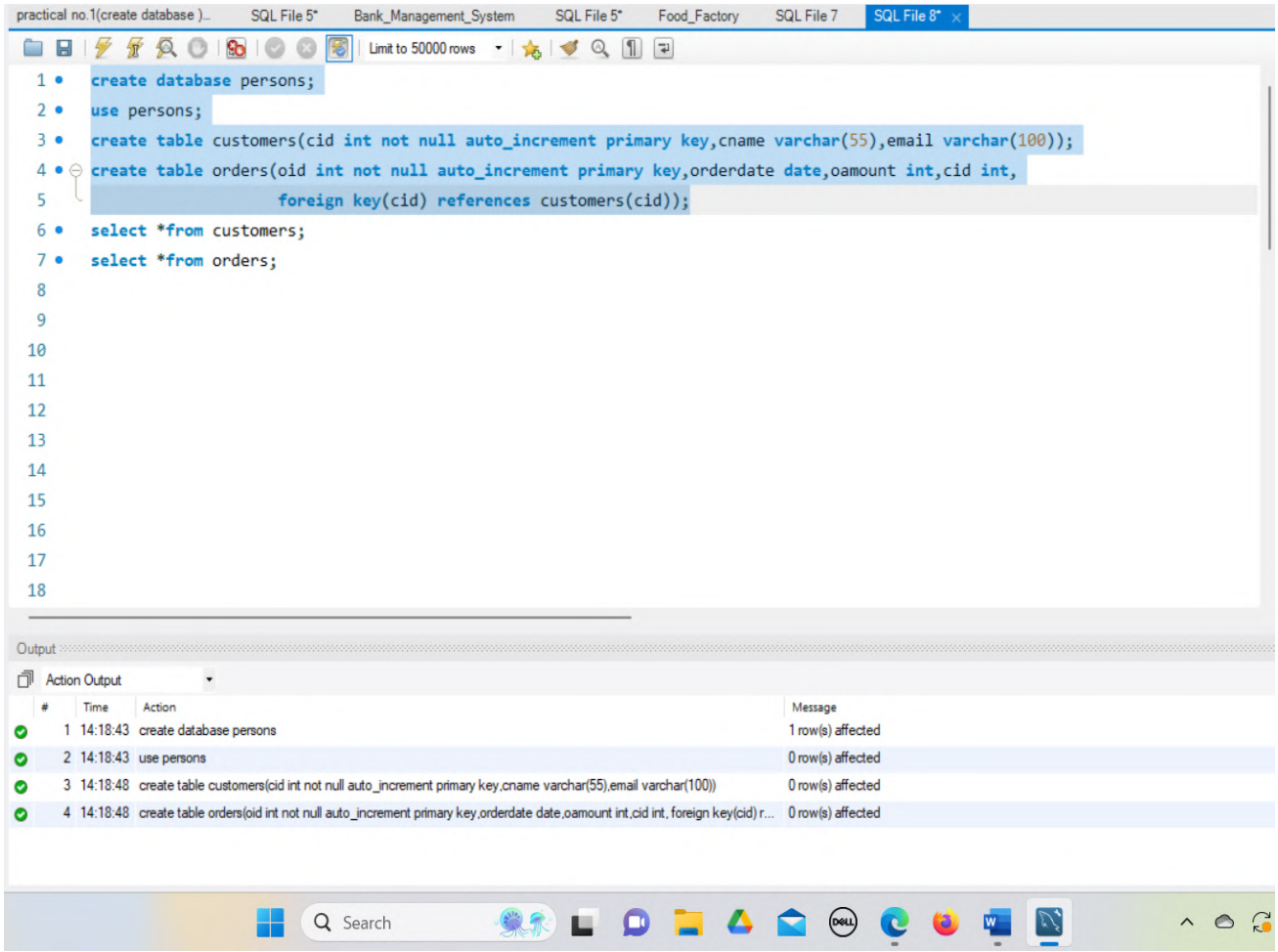
create database persons;

use persons;

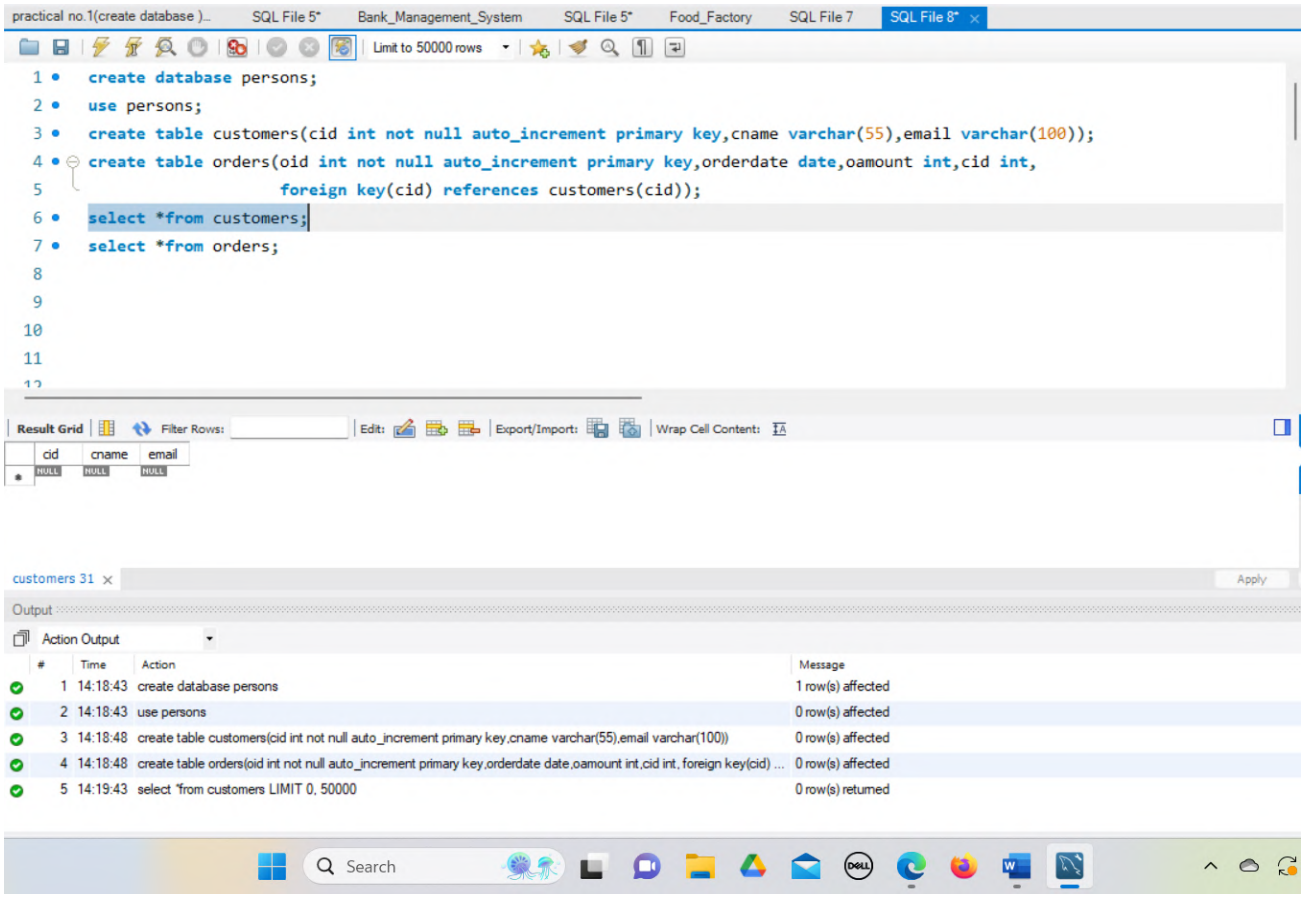
create table customers(cid int not null auto_increment primary key,cname varchar(55),email varchar(100));

create table orders(oid int not null auto_increment primary key,orderdate date,oamount int,cid int,

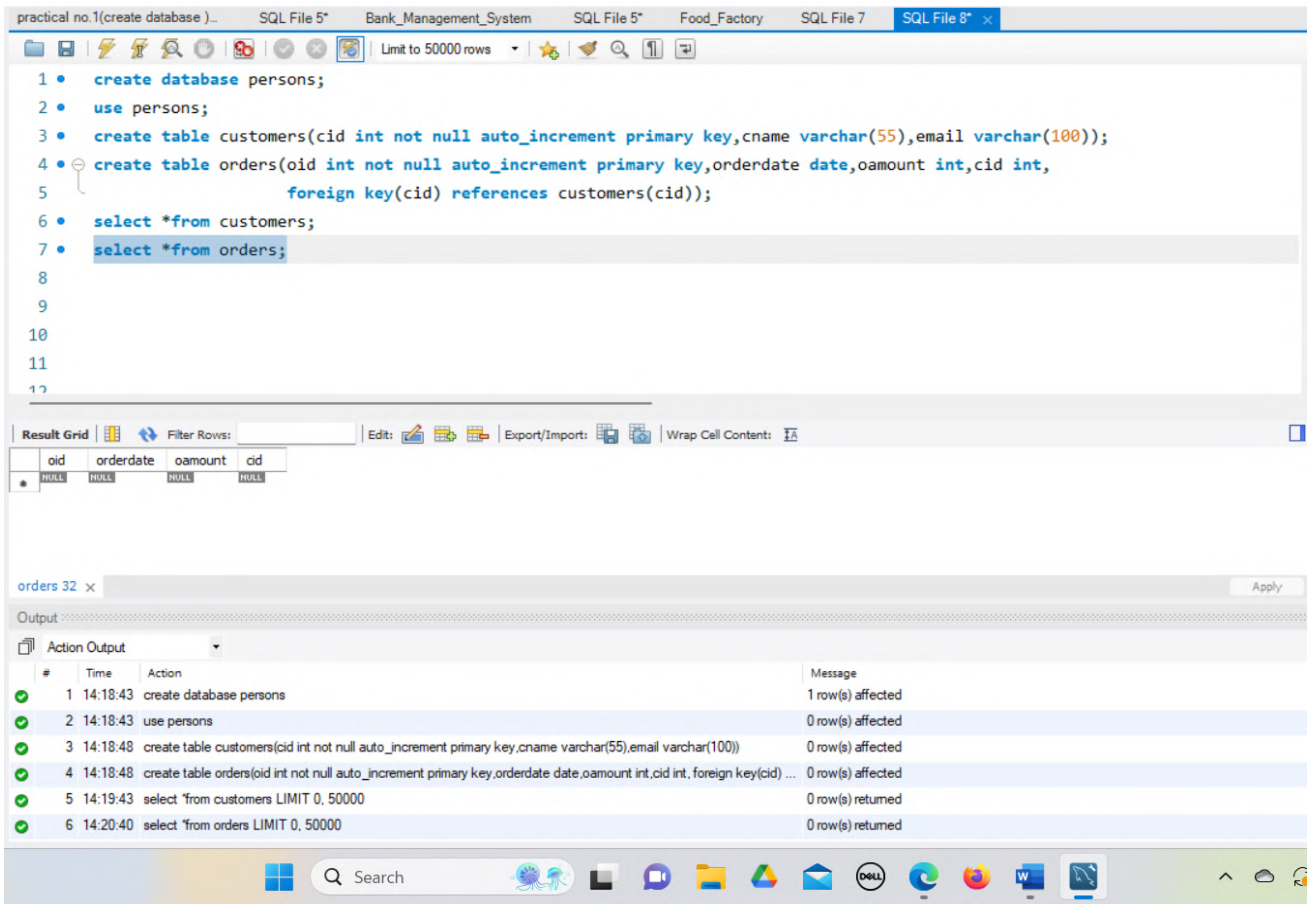
foreign key(cid) references customers(cid));



select *from customers;



select *from orders;



a. Alter table with add n modify

add constraints customer table:

create table customers(cid int not null auto_increment primary key,cname varchar(55),email varchar(100));

insert into customers values(1,'Samruddhi','samruddhi@gmail.com');

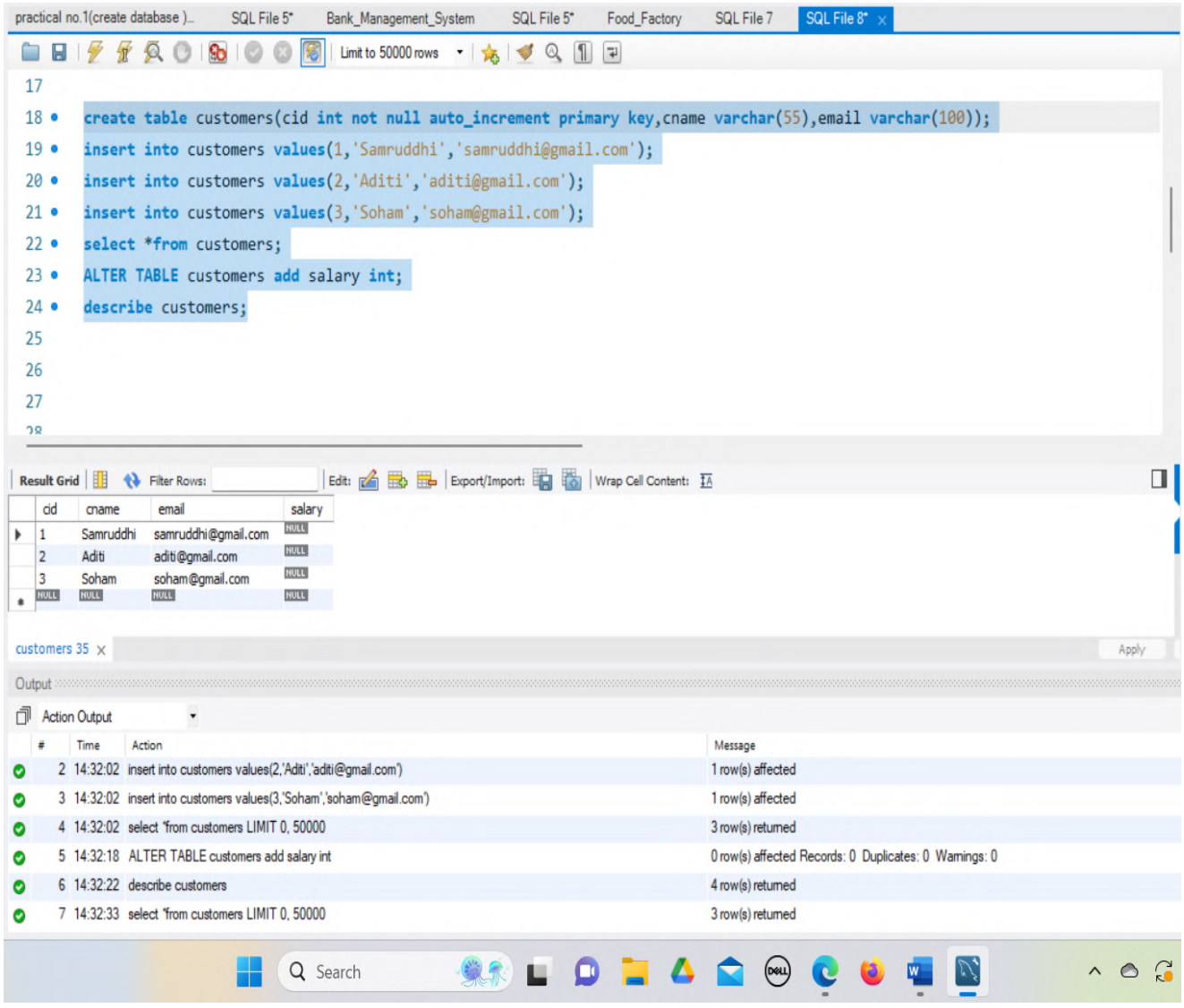
insert into customers values(2,'Aditi','aditi@gmail.com');

insert into customers values(3,'Soham','soham@gmail.com');

select *from customers;

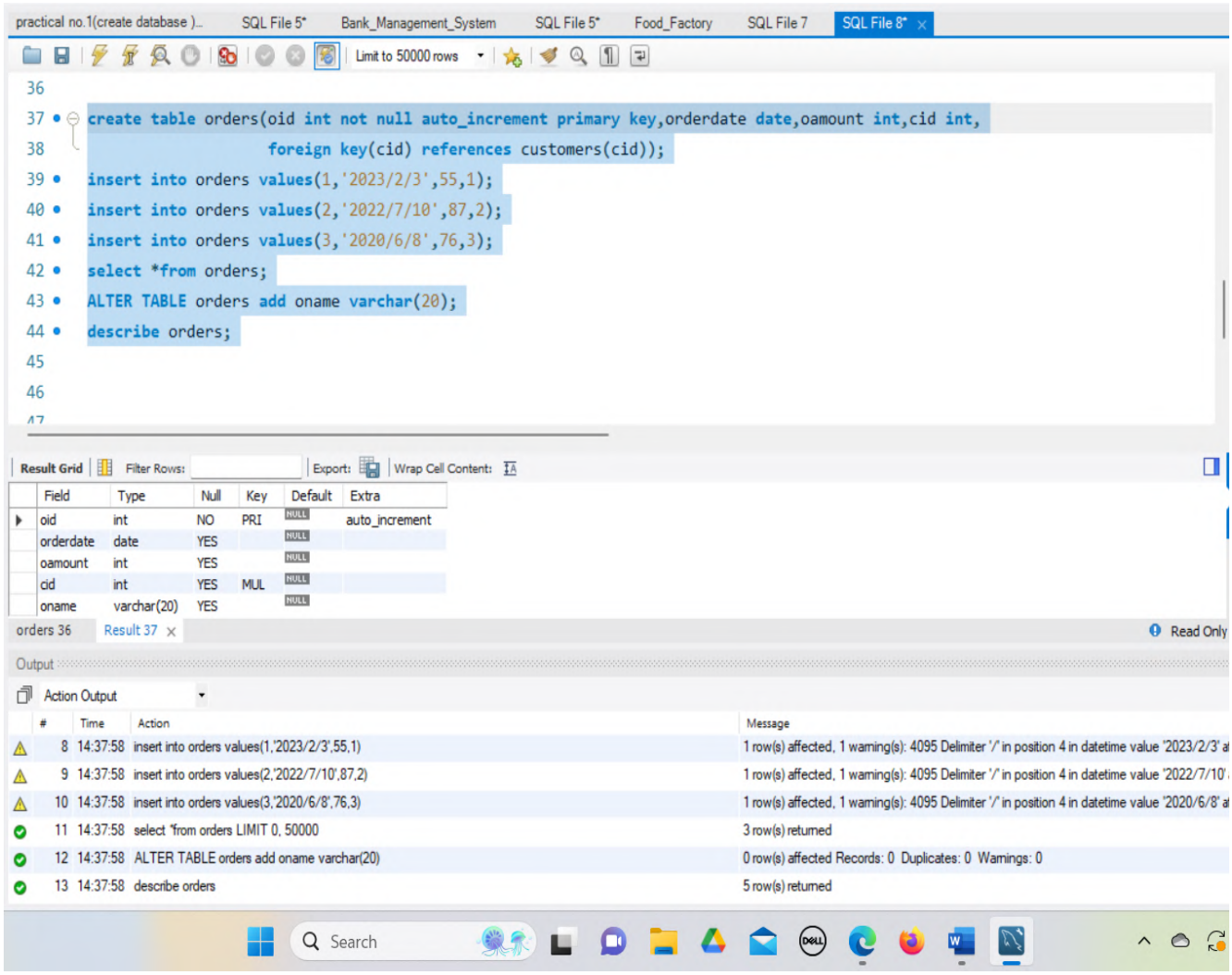
ALTER TABLE customers add salary int;

describe customers;



add constraints order table:

```
create table orders(oid int not null auto_increment primary key,orderdate date,oamount int,cid int,
                    foreign key(cid) references customers(cid));
insert into orders values(1,'2023/2/3',55,1);
insert into orders values(2,'2022/7/10',87,2);
insert into orders values(3,'2020/6/8',76,3);
select *from orders;
ALTER TABLE orders add oname varchar(20);
describe orders;
```



modify constraints customer table:

create table customers(cid int not null auto_increment primary key,cname varchar(55),email varchar(100));

insert into customers values(1,'Samruddhi','samruddhi@gmail.com');

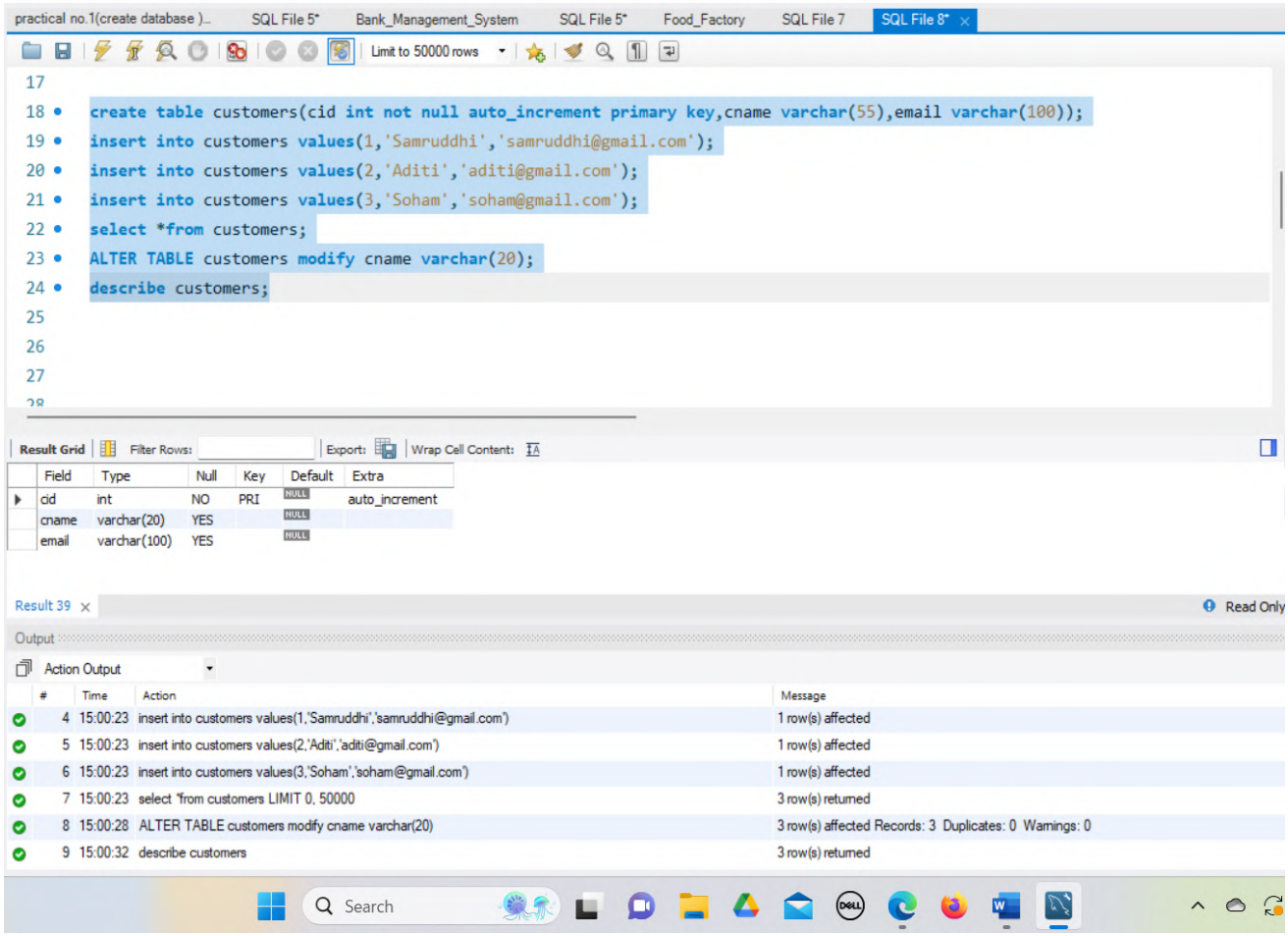
insert into customers values(2,'Aditi','aditi@gmail.com');

insert into customers values(3,'Soham','soham@gmail.com');

select *from customers;

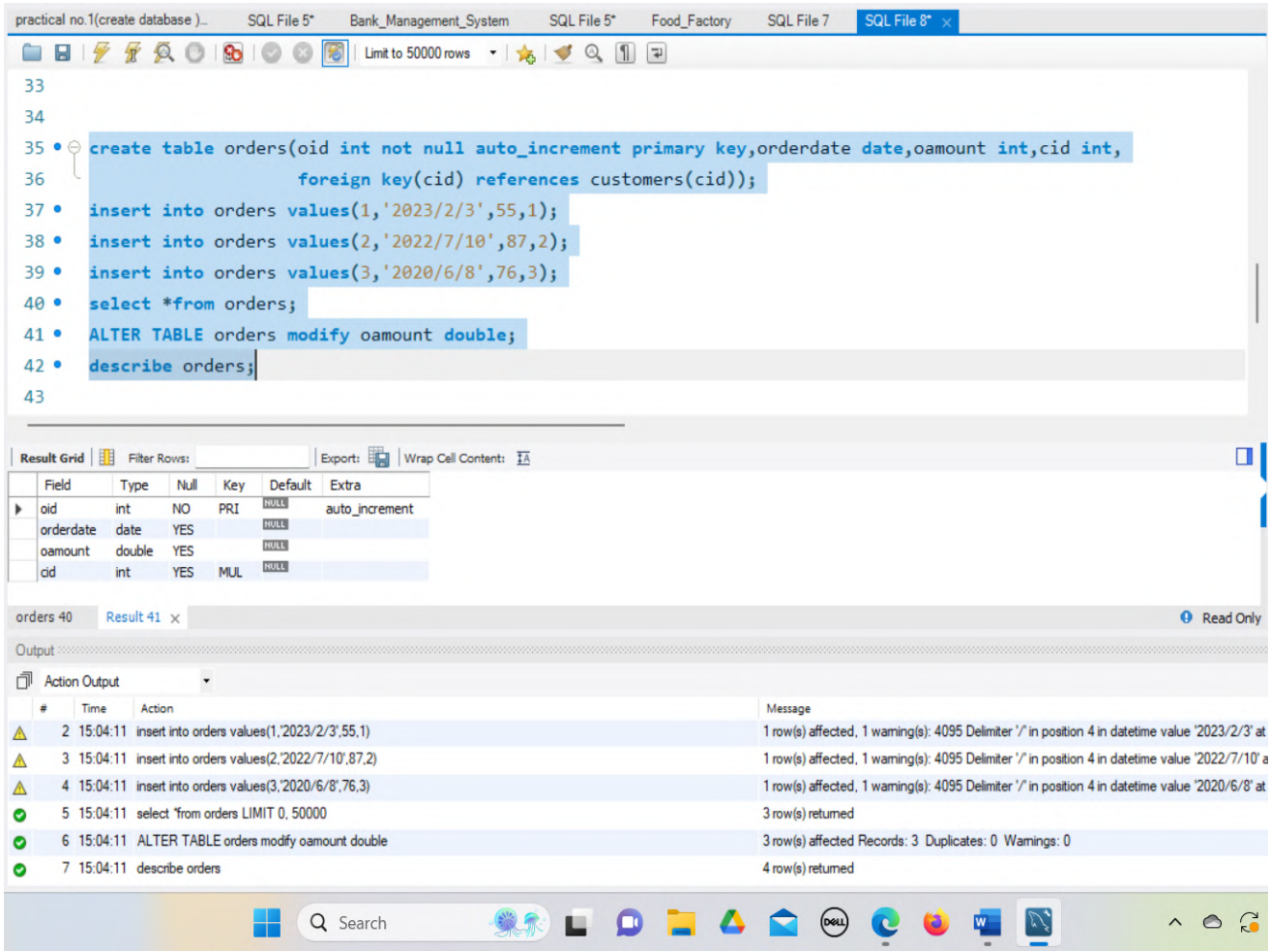
ALTER TABLE customers modify cname varchar(20);

describe customers;



modify constraints order table:

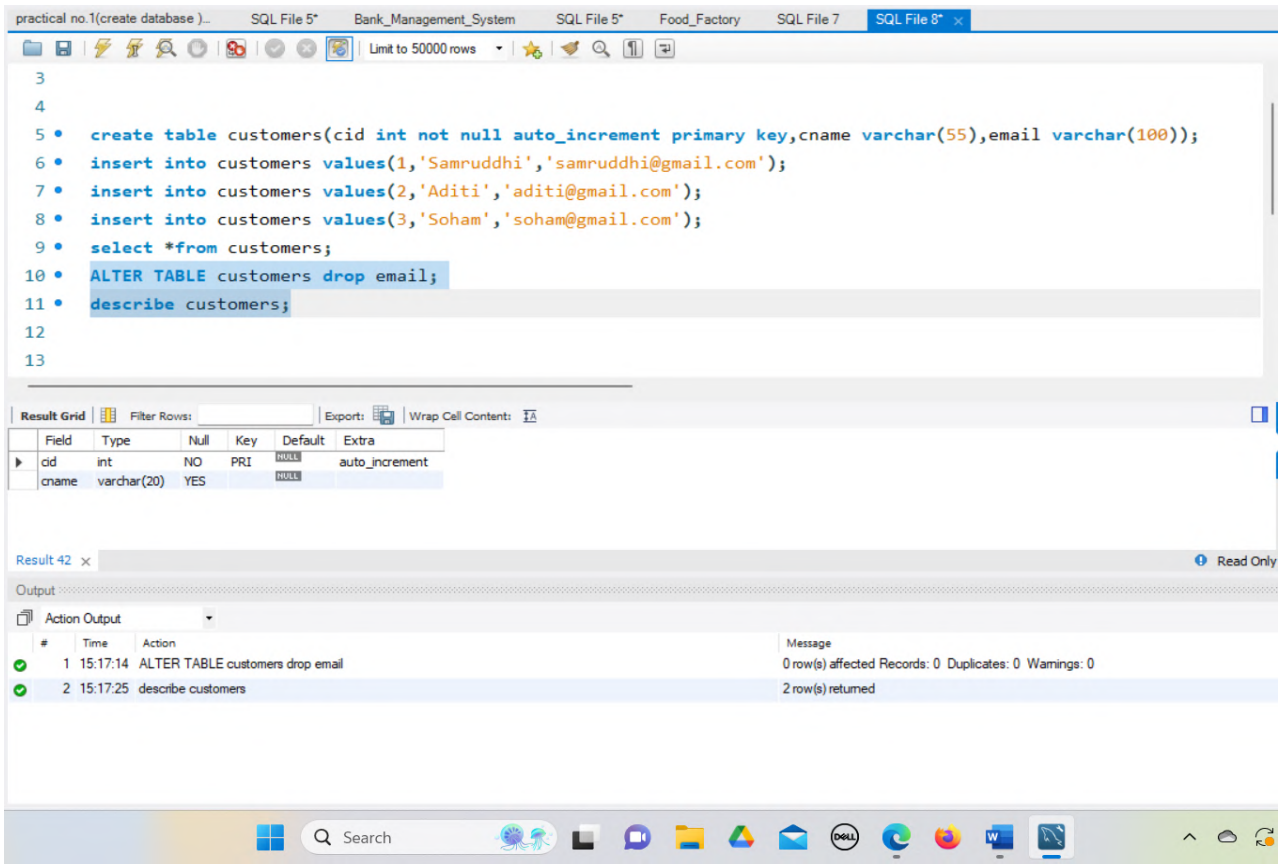
```
create table orders(oid int not null auto_increment primary key,orderdate date,oamount int,cid int,
                    foreign key(cid) references customers(cid));
insert into orders values(1,'2023/2/3',55,1);
insert into orders values(2,'2022/7/10',87,2);
insert into orders values(3,'2020/6/8',76,3);
select *from orders;
ALTER TABLE orders modify oamount double;
describe orders;
```



b. Drop table

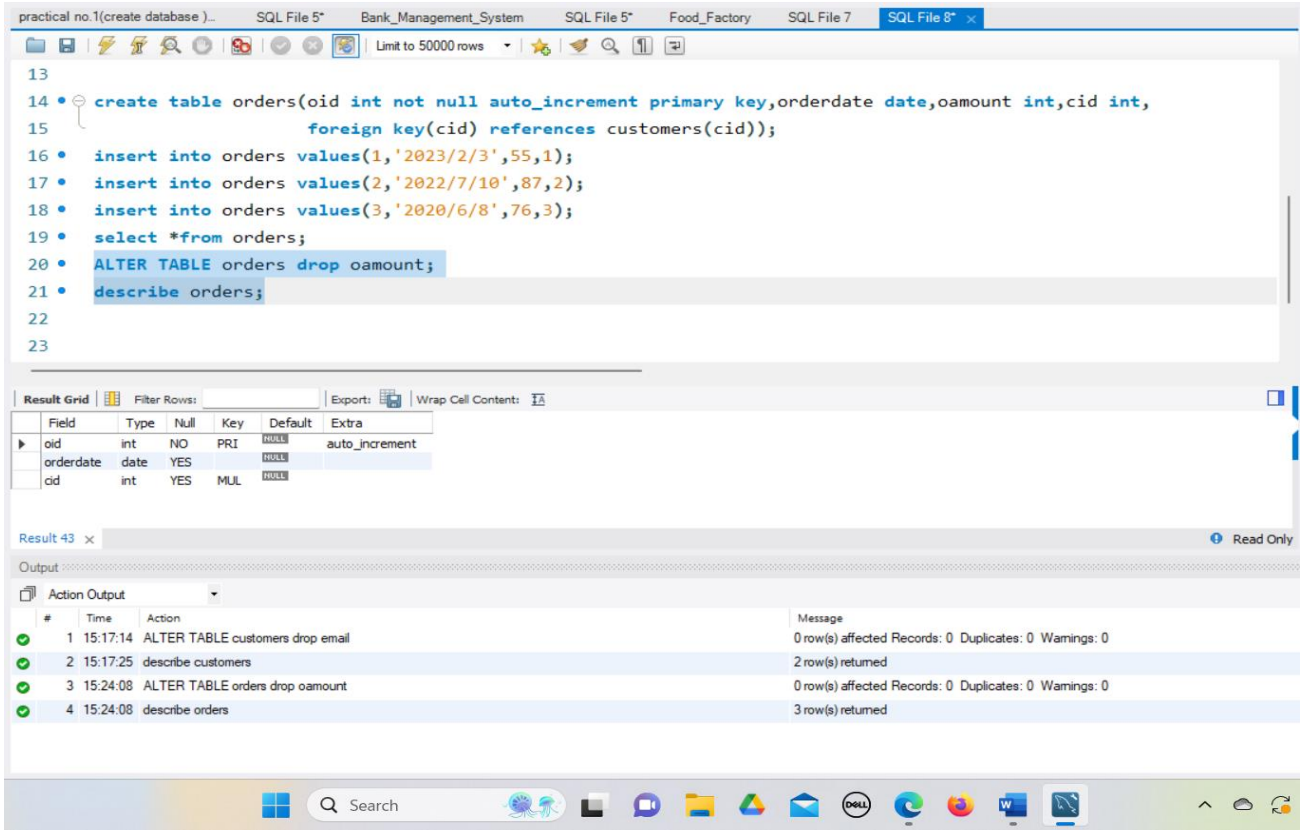
ALTER TABLE customers drop email;

describe customers;



ALTER TABLE orders drop oamount;

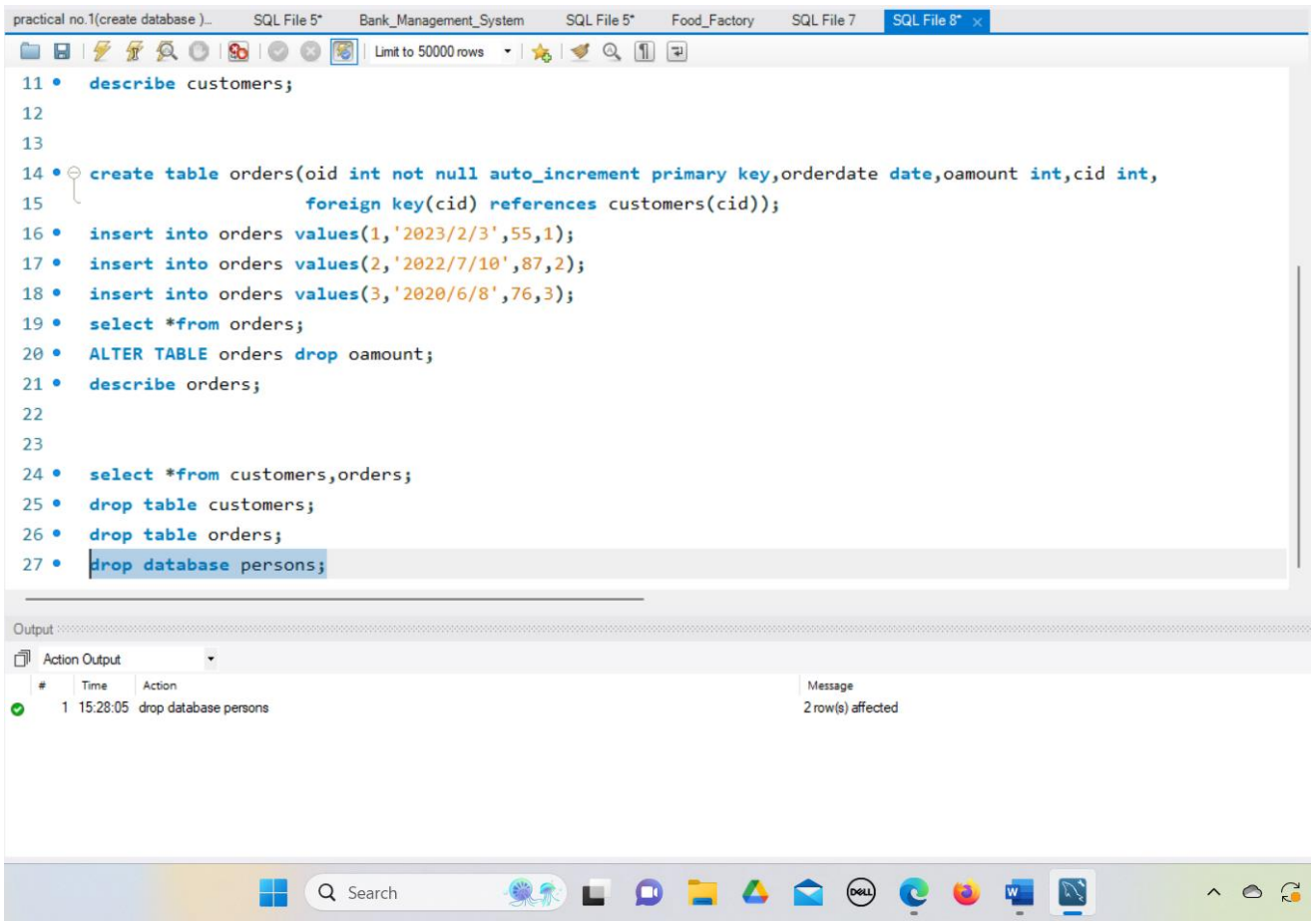
describe orders;



drop table customers;

drop table orders;

drop database persons;



Create Table with primary key and foreign key constraints.

Alter table with add n modify

Drop table

```
create database persons;
use persons;
create table customers(cid int not null auto_increment primary key,cname varchar(55),email varchar(100));
create table orders(oid int not null auto_increment primary key,orderdate date,oamount int,cid int,
                    foreign key(cid) references customers(cid));

insert into customers values(1,'Samruddhi','samruddhi@gmail.com');
insert into customers values(2,'Aditi','aditi@gmail.com');
insert into customers values(3,'Soham','soham@gmail.com');
select *from customers;
ALTER TABLE customers add salary int;
ALTER TABLE customers modify cname varchar(20);
ALTER TABLE customers drop salary;
describe customers;

insert into orders values(1,'2023/2/3',55,1);
insert into orders values(2,'2022/7/10',87,2);
insert into orders values(3,'2020/6/8',76,3);
select *from orders;
ALTER TABLE orders add oname varchar(20);
ALTER TABLE orders modify oamount double;
ALTER TABLE orders drop oname;
describe orders;

select *from customers,orders;
drop table customers;
drop table orders;
drop database persons;
```

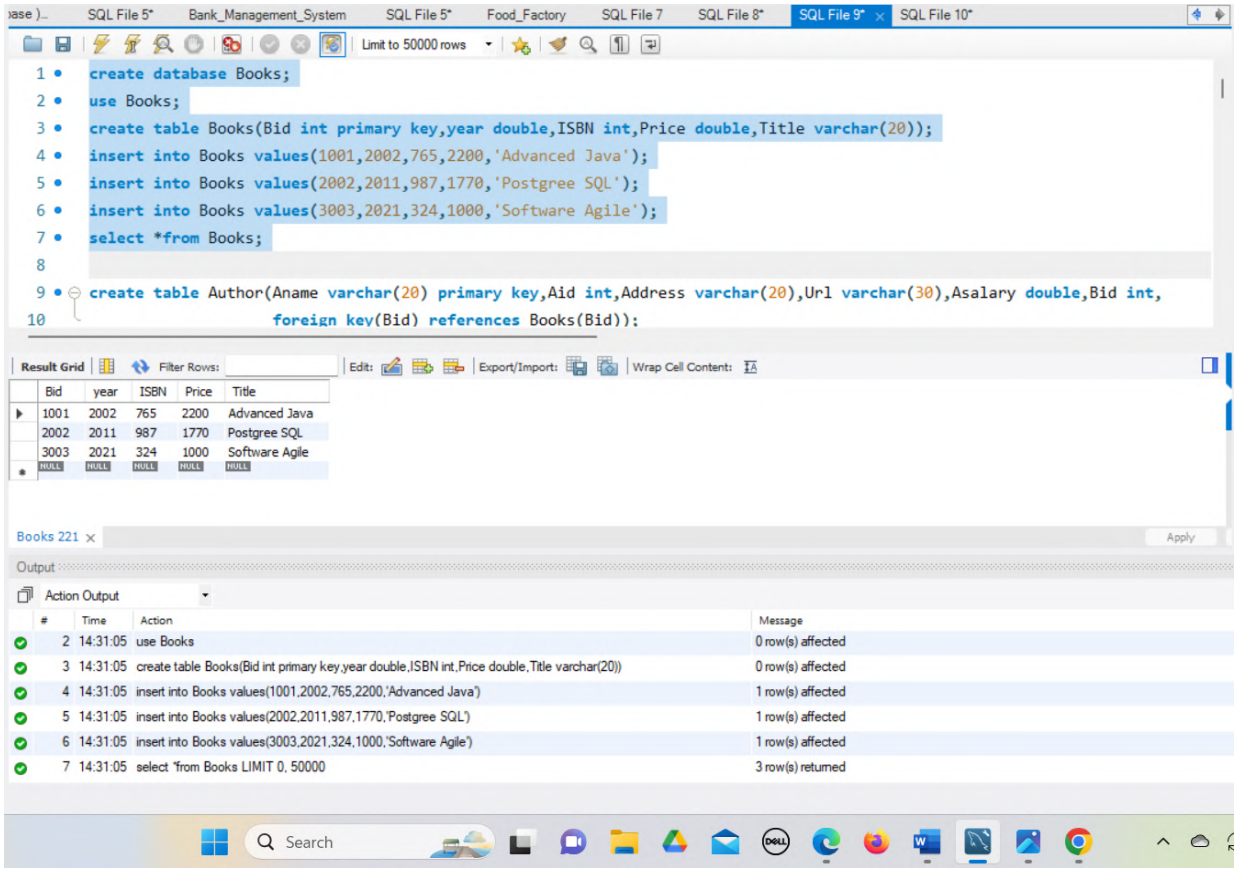
Group B: MySQL

4. Perform following SQL queries on the database created in assignment 1.

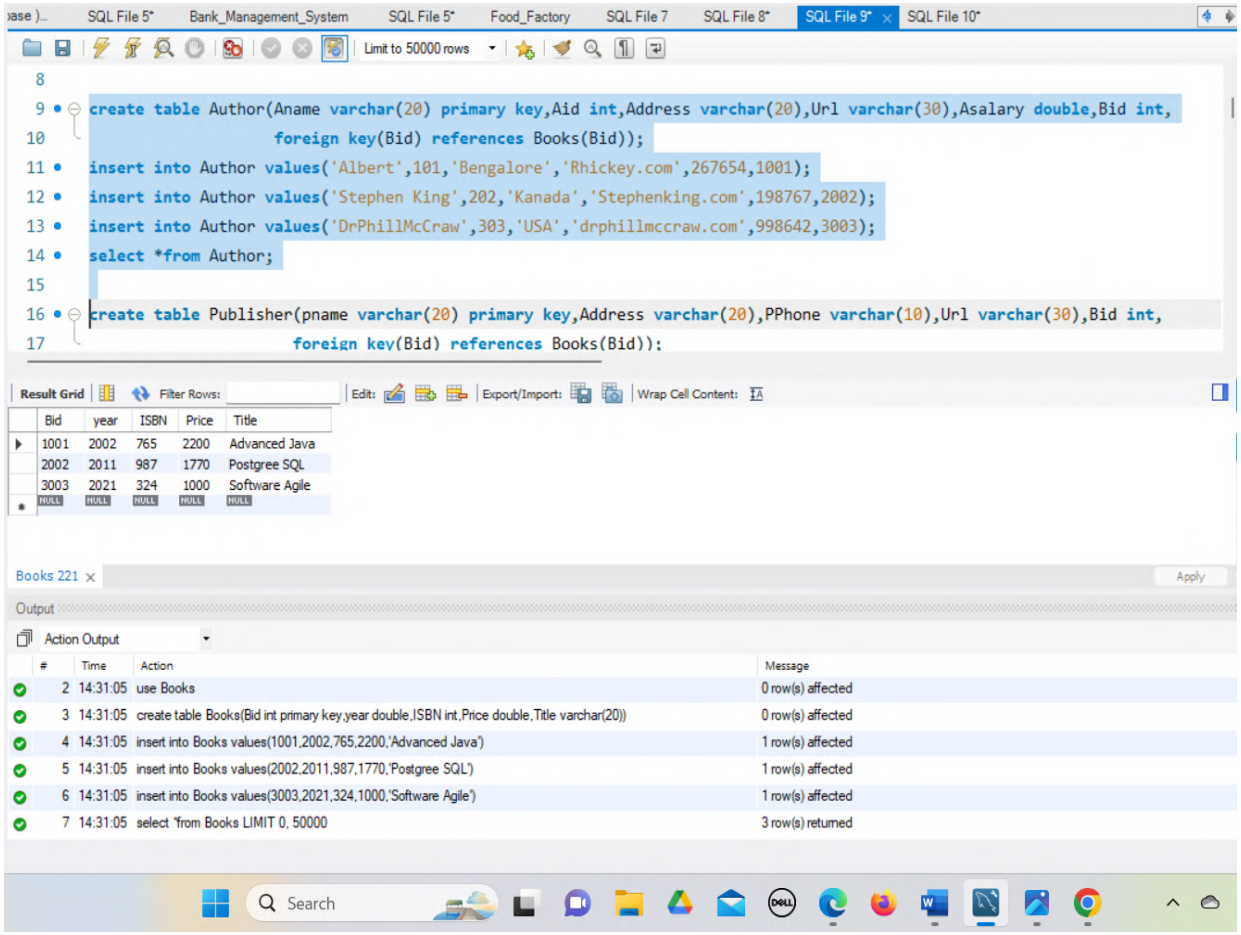
- **Implementation of relational operators in SQL**
- **Boolean operators and pattern matching**
- **Arithmetic operations and built in functions**
- **Group functions**
- **Processing Date and Time functions**
- **Complex queries and set operators**

```
create database Books;
use Books;
create table Books(Bid int primary key,year double,ISBN int,Price double,Title varchar(20));
insert into Books values(1001,2002,765,2200,'Advanced Java');
insert into Books values(2002,2011,987,1770,'Postgree SQL');
insert into Books values(3003,2021,324,1000,'Software Agile');
select *from Books;

create table Author(Aname varchar(20) primary key,Aid int,Address varchar(20),Url varchar(30),Asalary double,Bid int, foreign key(Bid) references Books(Bid));
insert into Author values('Albert',101,'Bengalore','Rhickey.com',267654,1001);
insert into Author values('Stephen King',202,'Kanada','Stephenking.com',198767,2002);
insert into Author values('DrPhillMcCraw',303,'USA','drphillmccraw.com',998642,3003);
select *from Author;

create table Publisher(pname varchar(20) primary key,Address varchar(20),PPhone varchar(10),Url varchar(30),Bid int,foreign key(Bid) references Books(Bid));
insert into Publisher values('John','Pune','9090987544','john.com',1001);
insert into Publisher values('Albert','mumbai','9865784321','albert.com',2002);
insert into Publisher values('hichey','kolkata','7689543221','hichey.com',3003);
select *from Publisher;

create table ShoppingBasket(basket_ID int primary key,Bid int,foreign key(Bid) references Books(Bid));
insert into ShoppingBasket values(298,1001);
insert into ShoppingBasket values(898,2002);
insert into ShoppingBasket values(760,3003);
select *from ShoppingBasket;

create table Customer(email varchar(30) primary key,Cid int,C_name varchar(20),CPhone varchar(10),Address varchar(30),CDOB date,basket_ID int,foreign key(basket_ID) references ShoppingBasket(basket_ID));
insert into Customer values('samruddhikangude@gmail.com',100,'samruddhi','7218380003','Pune','2017-06-06',298);
insert into Customer values('aditimagar@gmail.com',200,'aditi','8787878787','osmanabad','2015-05-04',898);
insert into Customer values('sohampawar@gmail.com',300,'soham','7765432234','Baramati','2019-09-12',760); select *from Customer; drop database Books;
```

Books Table:



Author Table:



Publisher Table:

```
14 • select *from Author;
15
16 • create table Publisher(pname varchar(20) primary key,Address varchar(20),PPhone varchar(10),Url varchar(30),Bid int,
17                         foreign key(Bid) references Books(Bid));
18 • insert into Publisher values('John','Pune','9090987544','john.com',1001);
19 • insert into Publisher values('Albert','mumbai','9865784321','albert.com',2002);
20 • insert into Publisher values('hichey','kolkata','7689543221','hichey.com',3003);
21 • select *from Publisher;
22
23 • create table ShoppingBasket(basket_ID int primary key,Bid int,
```

pname	Address	PPhone	Url	Bid
Albert	mumbai	9865784321	albert.com	2002
hichey	kolkata	7689543221	hichey.com	3003
John	Pune	9090987544	john.com	1001
NULL	NULL	NULL	NULL	NULL

Publisher 222

Output

Action Output

#	Time	Action	Message
7	14:31:05	select *from Books LIMIT 0, 50000	3 row(s) returned
8	14:34:12	create table Publisher(pname varchar(20) primary key,Address varchar(20),PPhone varchar(10),Url varchar(30),...	0 row(s) affected
9	14:34:12	insert into Publisher values('John','Pune','9090987544','john.com',1001)	1 row(s) affected
10	14:34:12	insert into Publisher values('Albert','mumbai','9865784321','albert.com',2002)	1 row(s) affected
11	14:34:12	insert into Publisher values('hichey','kolkata','7689543221','hichey.com',3003)	1 row(s) affected
12	14:34:12	select *from Publisher LIMIT 0, 50000	3 row(s) returned

ShoppingBasket Table:

```
21 • select *from Publisher;
22
23 • create table ShoppingBasket(basket_ID int primary key,Bid int,
24                         foreign key(Bid) references Books(Bid));
25 • insert into ShoppingBasket values(298,1001);
26 • insert into ShoppingBasket values(898,2002);
27 • insert into ShoppingBasket values(760,3003);
28 • select *from ShoppingBasket;
29
30 • create table Customer(email varchar(30) primary key,Cid int,C_name varchar(20),CPhone varchar(10),Address varchar(30),
```

basket_ID	Bid
298	1001
898	2002
760	3003
NULL	NULL

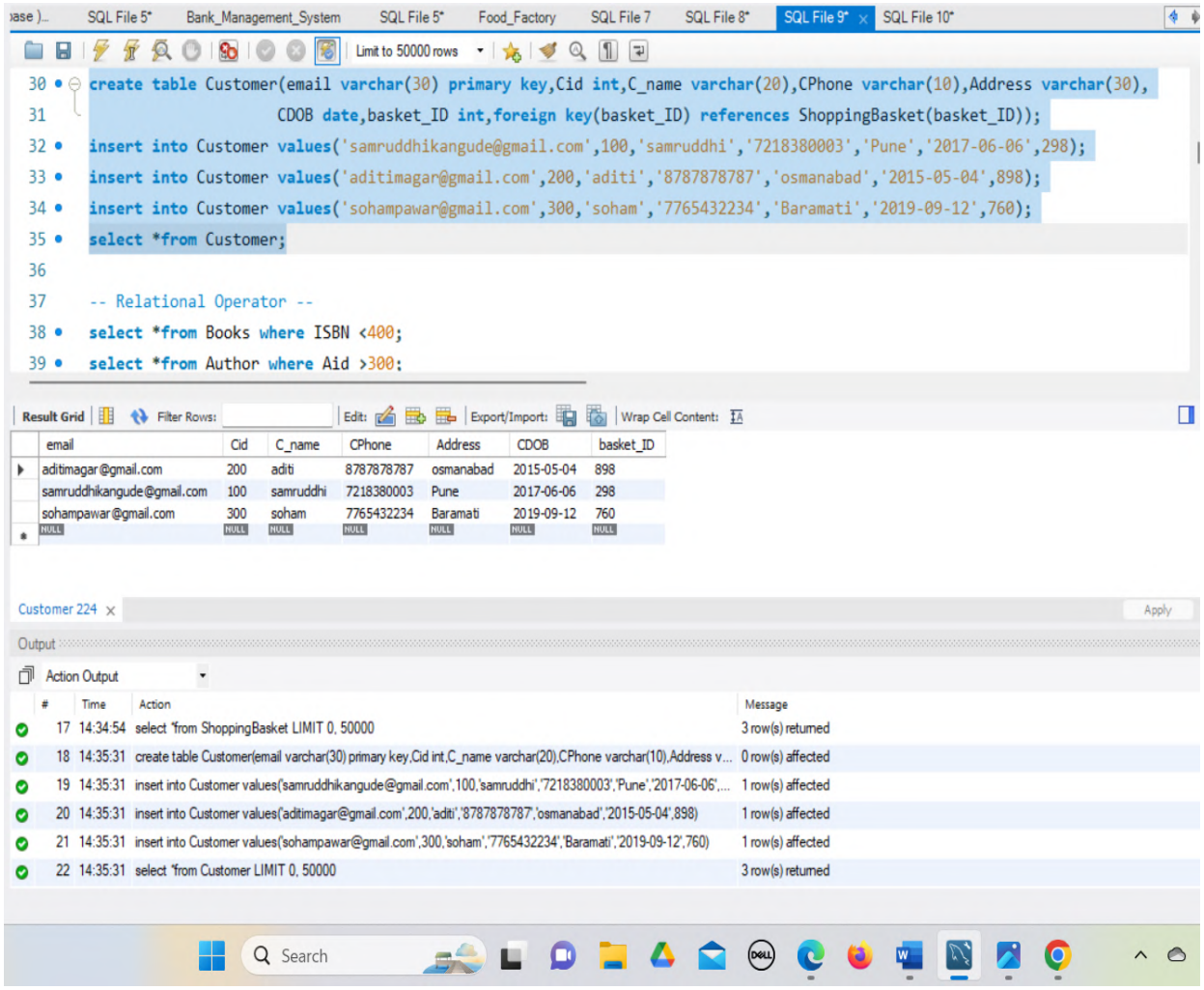
ShoppingBasket 223

Output

Action Output

#	Time	Action	Message
12	14:34:12	select *from Publisher LIMIT 0, 50000	3 row(s) returned
13	14:34:54	create table ShoppingBasket(basket_ID int primary key,Bid int, foreign key(Bid) references Books(...	0 row(s) affected
14	14:34:54	insert into ShoppingBasket values(298,1001)	1 row(s) affected
15	14:34:54	insert into ShoppingBasket values(898,2002)	1 row(s) affected
16	14:34:54	insert into ShoppingBasket values(760,3003)	1 row(s) affected
17	14:34:54	select *from ShoppingBasket LIMIT 0, 50000	3 row(s) returned

Customer Table:



```
create table Customer(email varchar(30) primary key,Cid int,C_name varchar(20),CPhone varchar(10),Address varchar(30),
                      CDOB date,basket_ID int,foreign key(basket_ID) references ShoppingBasket(basket_ID));
insert into Customer values('samruddhikangude@gmail.com',100,'samruddhi','7218380003','Pune','2017-06-06',298);
insert into Customer values('aditimagar@gmail.com',200,'aditi','8787878787','osmanabad','2015-05-04',898);
insert into Customer values('sohampawar@gmail.com',300,'soham','7765432234','Baramati','2019-09-12',760);
select *from Customer;

-- Relational Operator --
select *from Books where ISBN <400;
select *from Author where Aid >300;
```

Result Grid

email	Cid	C_name	CPhone	Address	CDOB	basket_ID
aditimagar@gmail.com	200	aditi	8787878787	osmanabad	2015-05-04	898
samruddhikangude@gmail.com	100	samruddhi	7218380003	Pune	2017-06-06	298
sohampawar@gmail.com	300	soham	7765432234	Baramati	2019-09-12	760
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Output

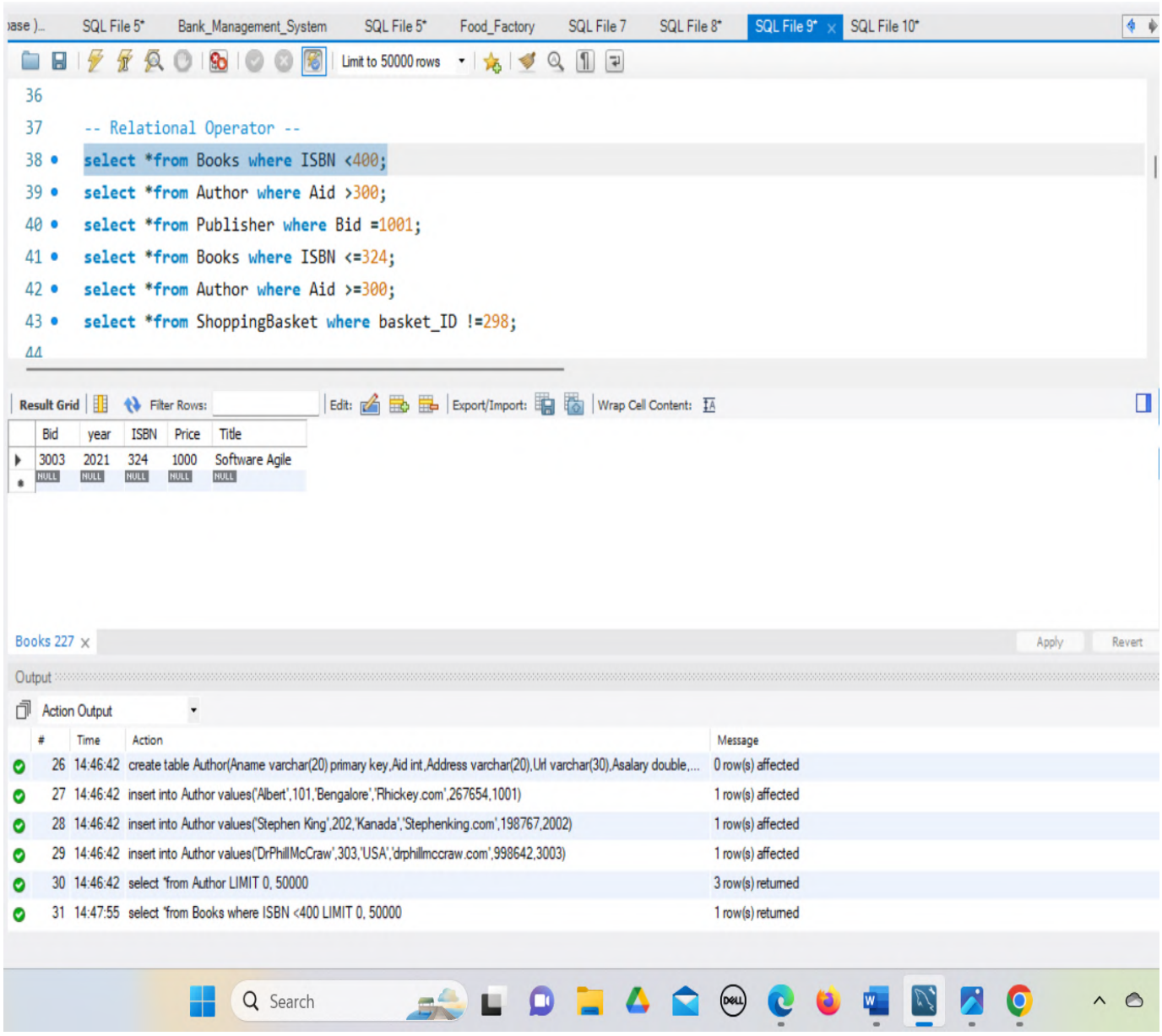
Action Output

#	Time	Action	Message
17	14:34:54	select *from ShoppingBasket LIMIT 0, 50000	3 row(s) returned
18	14:35:31	create table Customer(email varchar(30) primary key,Cid int,C_name varchar(20),CPhone varchar(10),Address v...	0 row(s) affected
19	14:35:31	insert into Customer values('samruddhikangude@gmail.com',100,'samruddhi','7218380003','Pune','2017-06-06',...	1 row(s) affected
20	14:35:31	insert into Customer values('aditimagar@gmail.com',200,'aditi','8787878787','osmanabad','2015-05-04',898)	1 row(s) affected
21	14:35:31	insert into Customer values('sohampawar@gmail.com',300,'soham','7765432234','Baramati','2019-09-12',760)	1 row(s) affected
22	14:35:31	select *from Customer LIMIT 0, 50000	3 row(s) returned

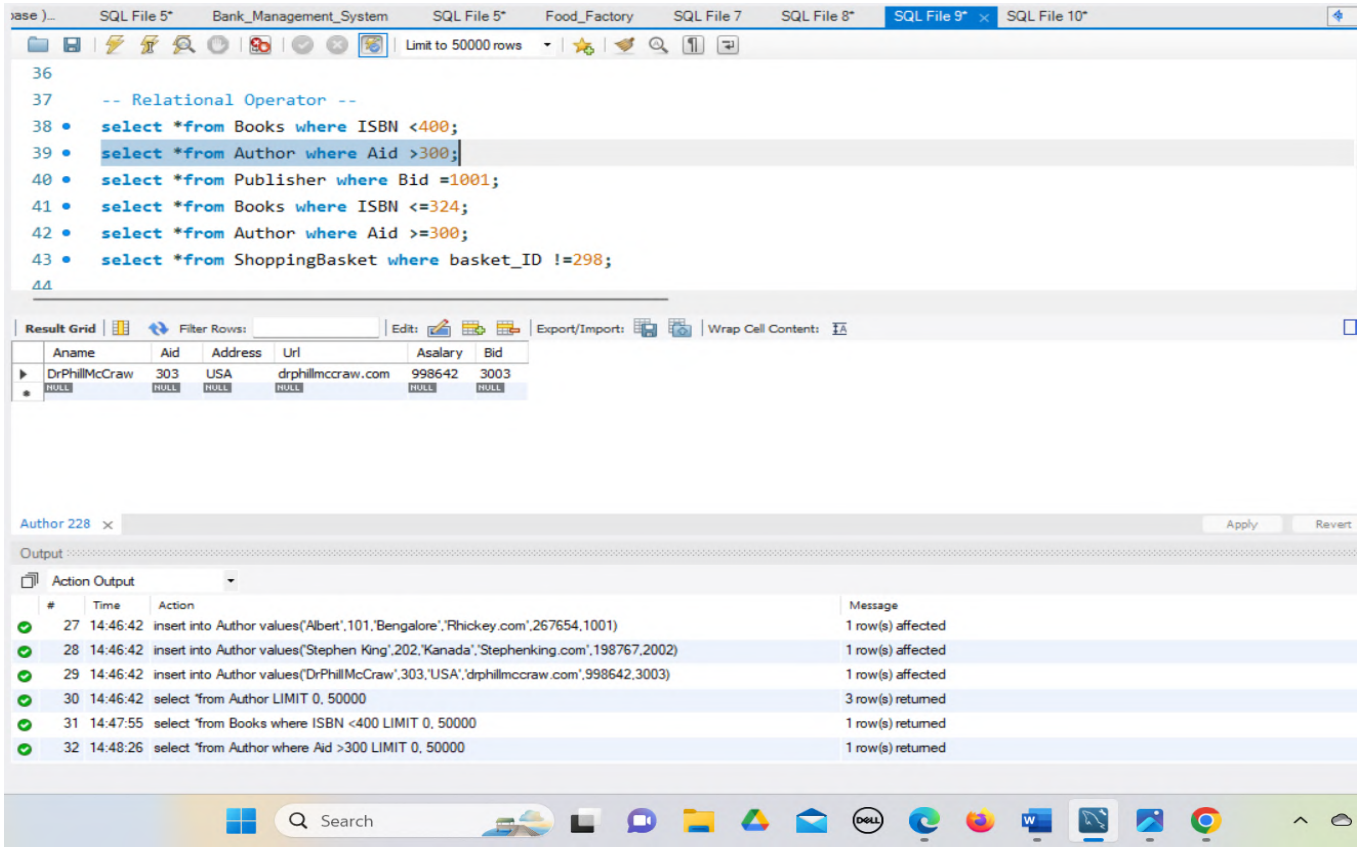
A: Implementation of relational operators in SQL:

-- Relational Operator --
select *from Books where ISBN <400;
select *from Author where Aid >300;
select *from Publisher where Bid =1001;
select *from Books where ISBN <=324;
select *from Author where Aid >=300;
select *from ShoppingBasket where basket_ID !=298;

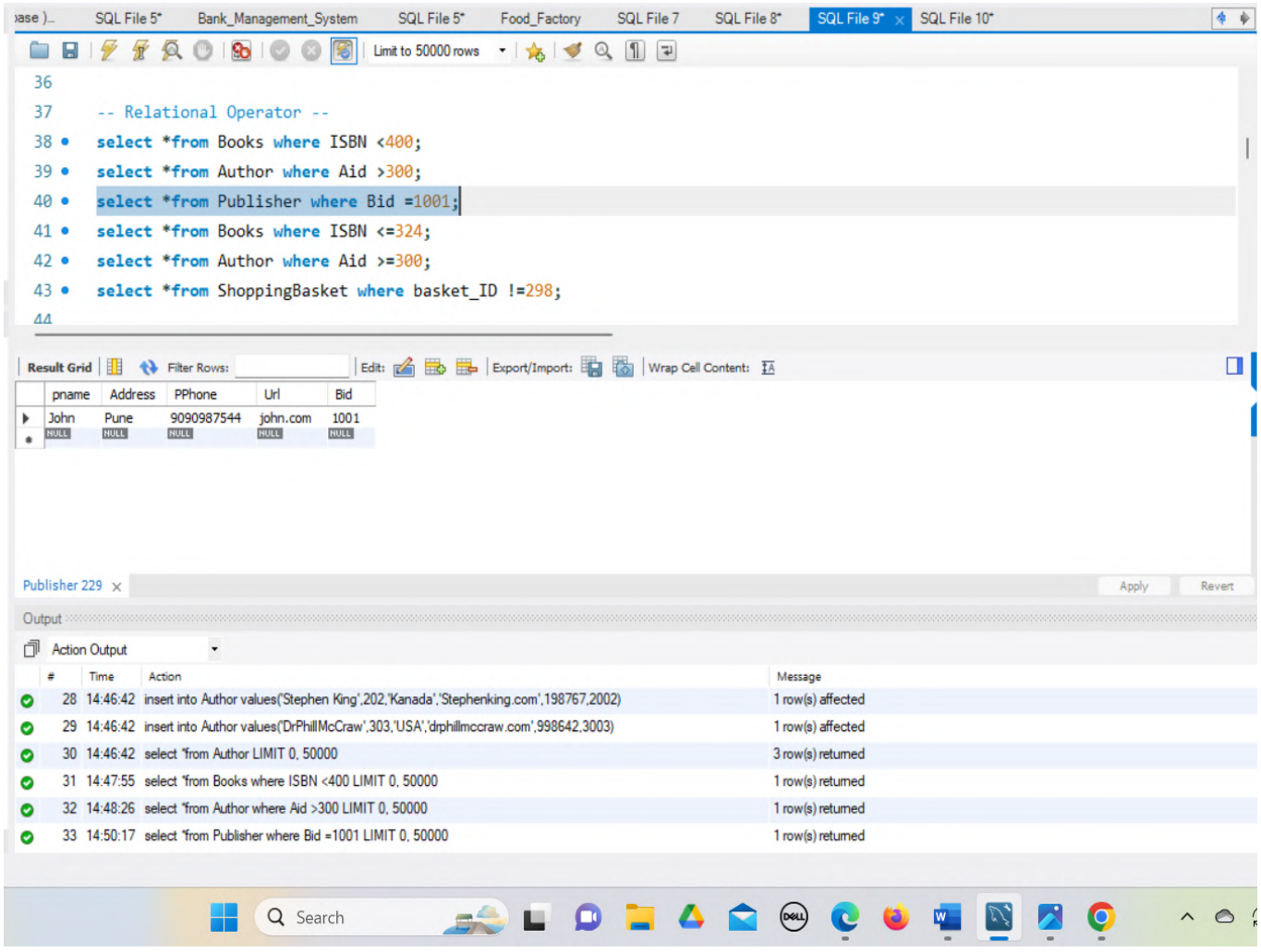
select *from Books where ISBN <400;



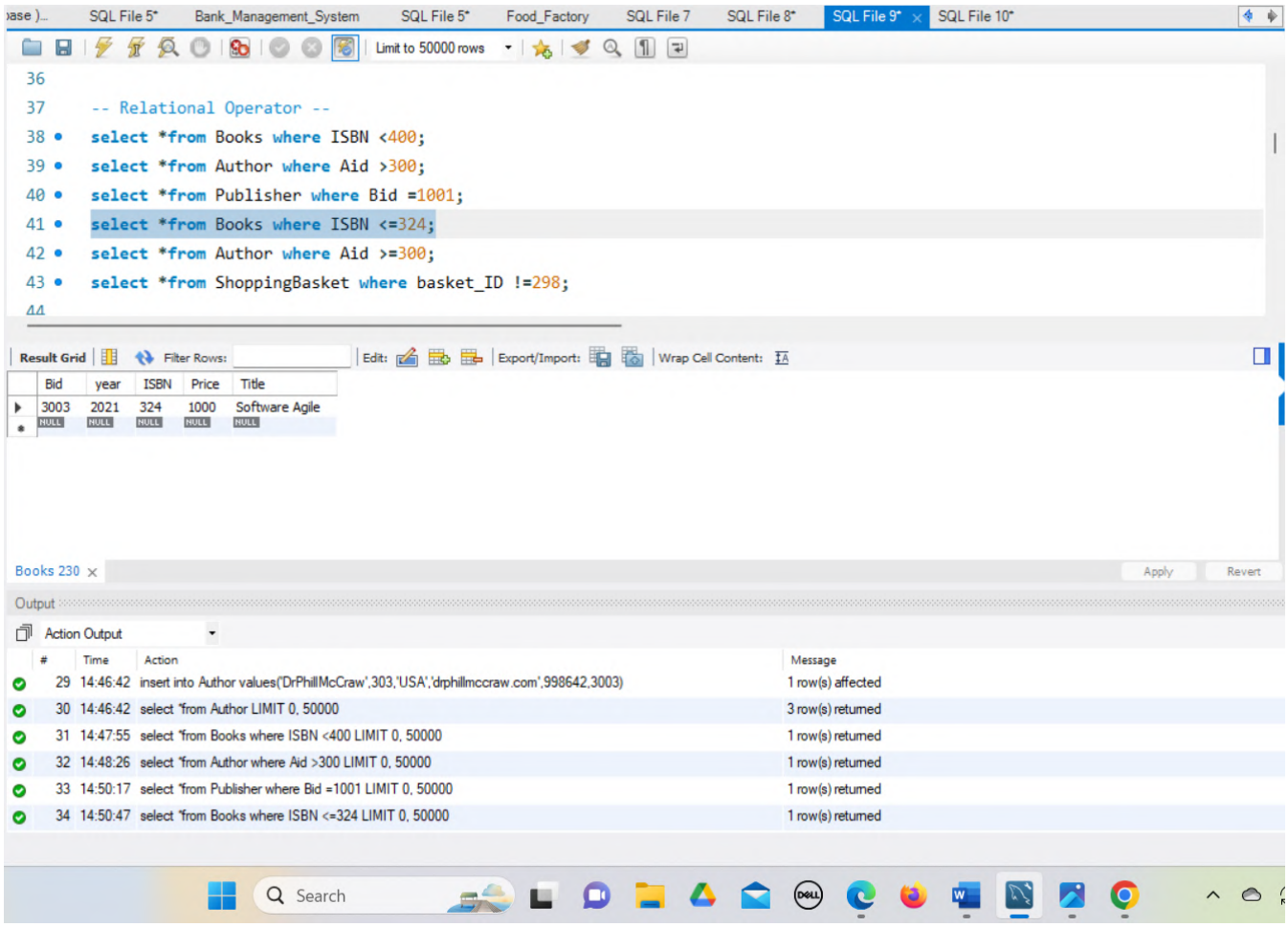
select *from Author where Aid >300;



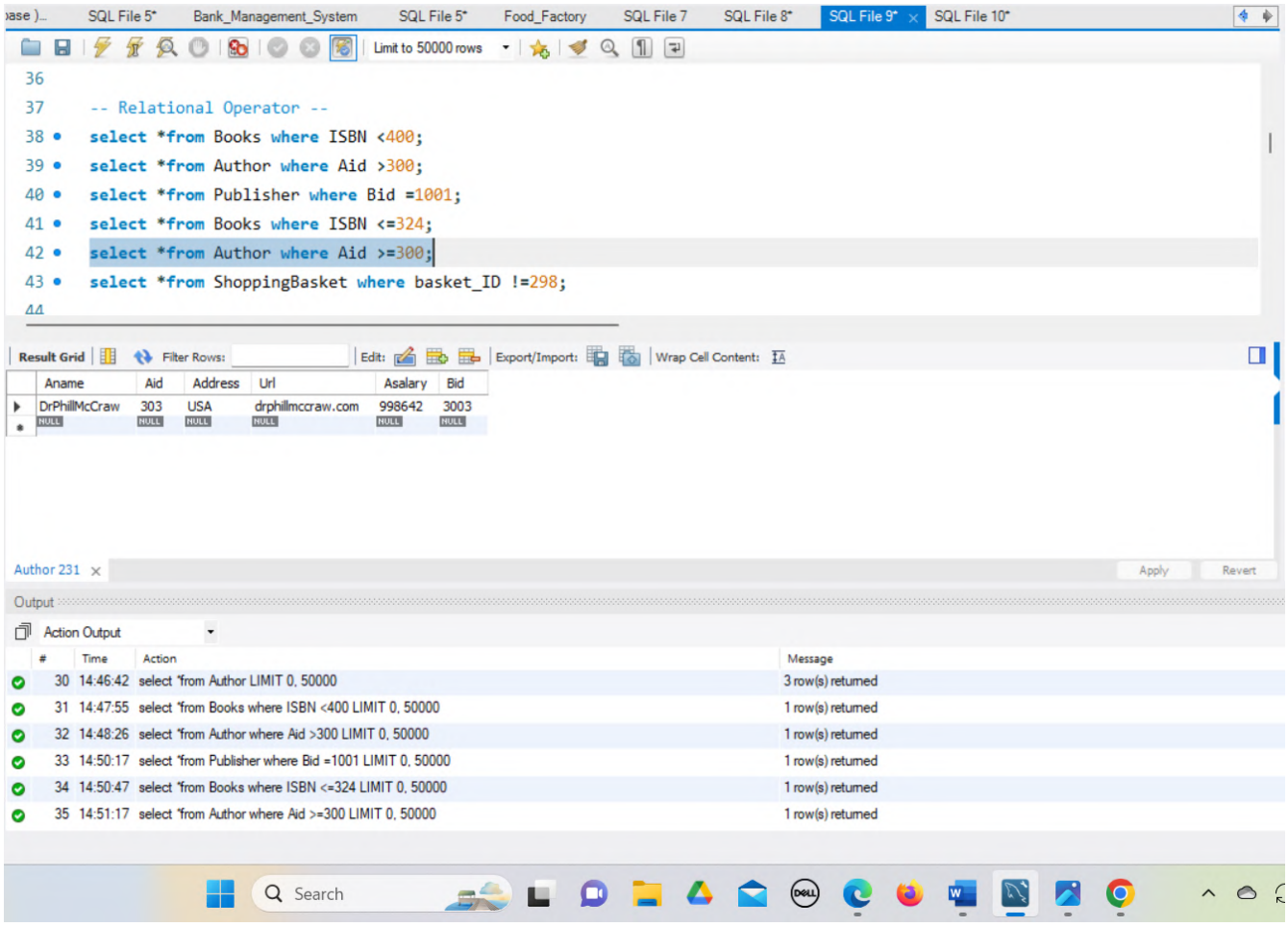
select *from Publisher where Bid =1001;



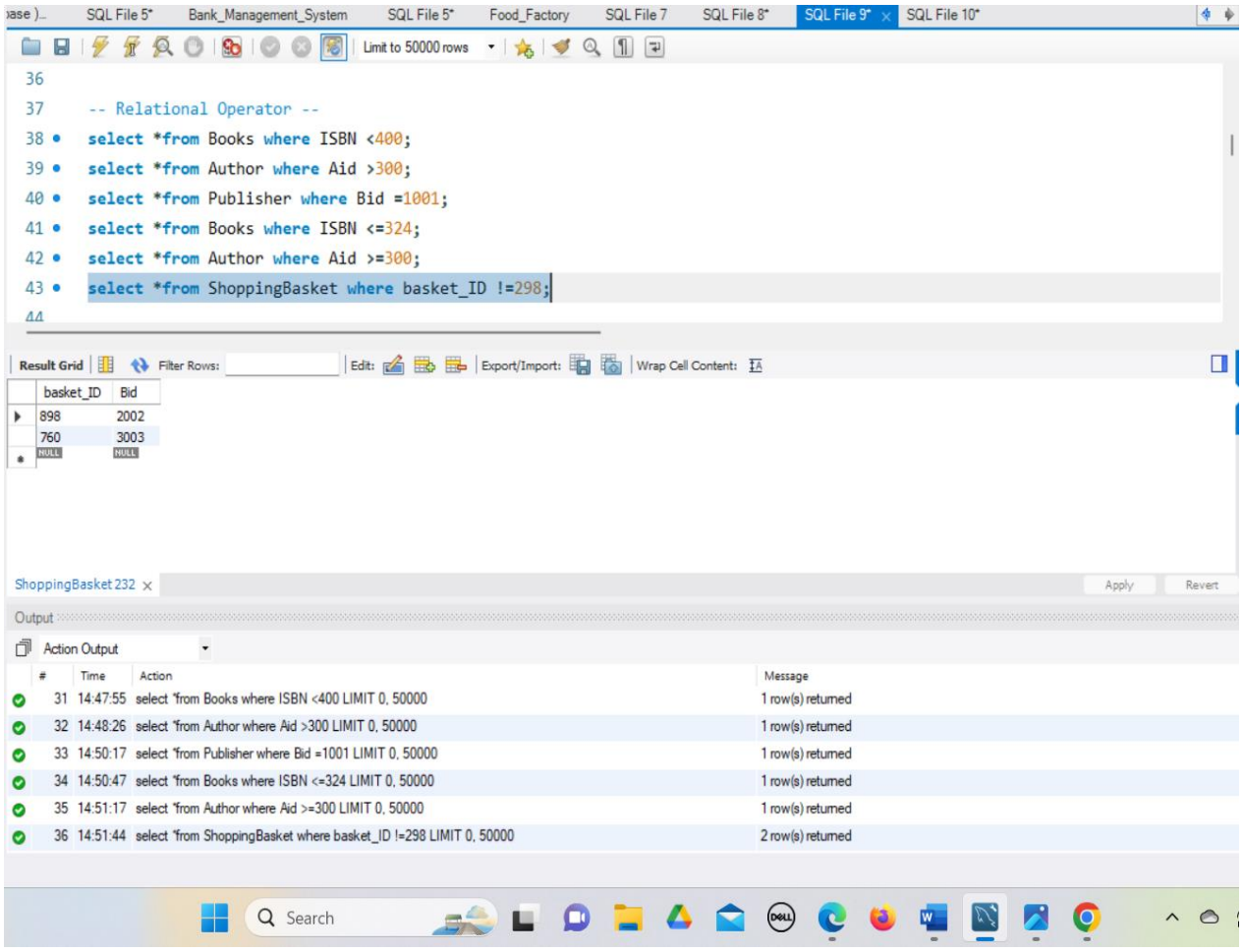
select *from Books where ISBN <=324;



select *from Author where Aid >=300;



select *from ShoppingBasket where basket_ID !=298;



B: Boolean operators and pattern matching:

-- Boolean operators --

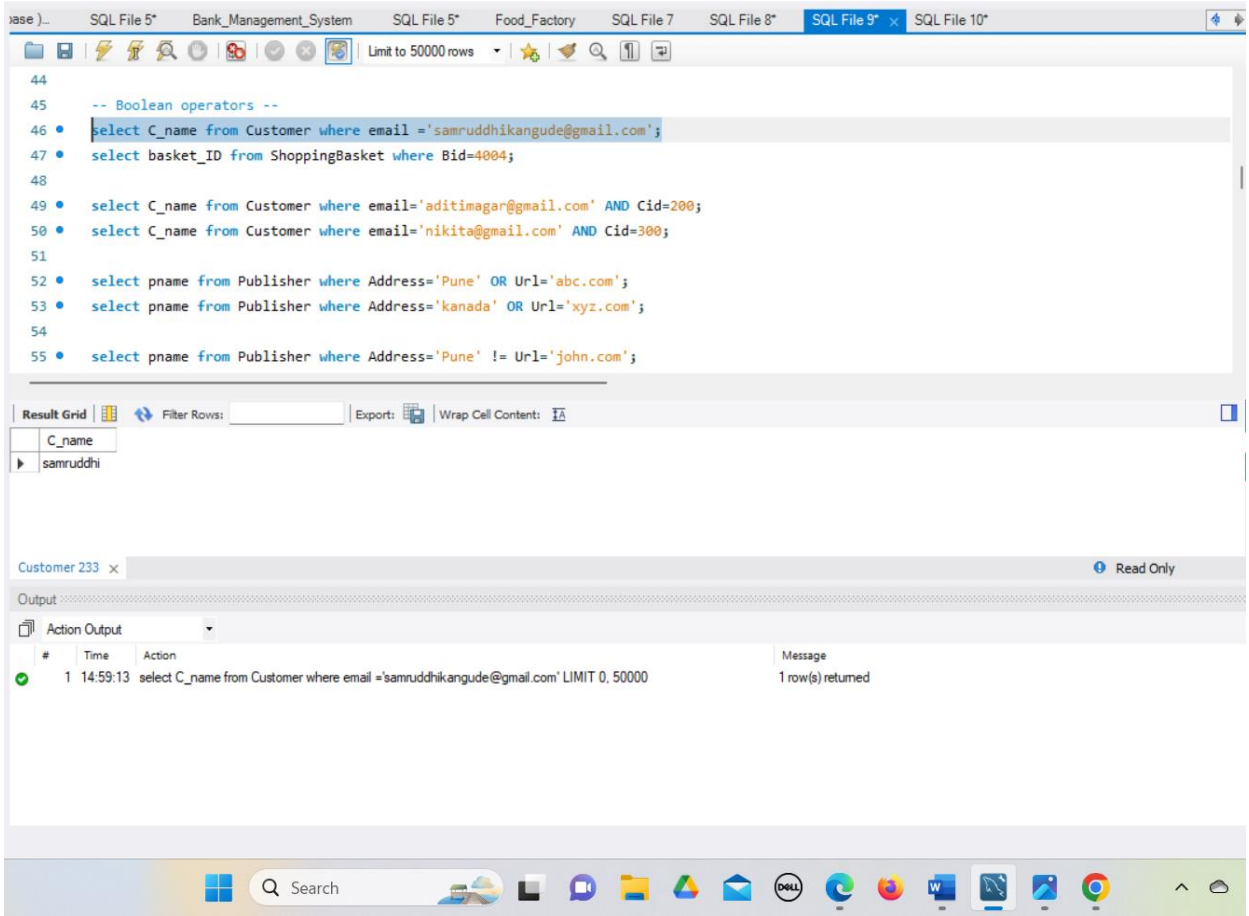
select C_name from Customer where email ='samruddhikangude@gmail.com';
select basket_ID from ShoppingBasket where Bid=4004;

select C_name from Customer where email='aditimagar@gmail.com' AND Cid=200;
select C_name from Customer where email='nikita@gmail.com' AND Cid=300;

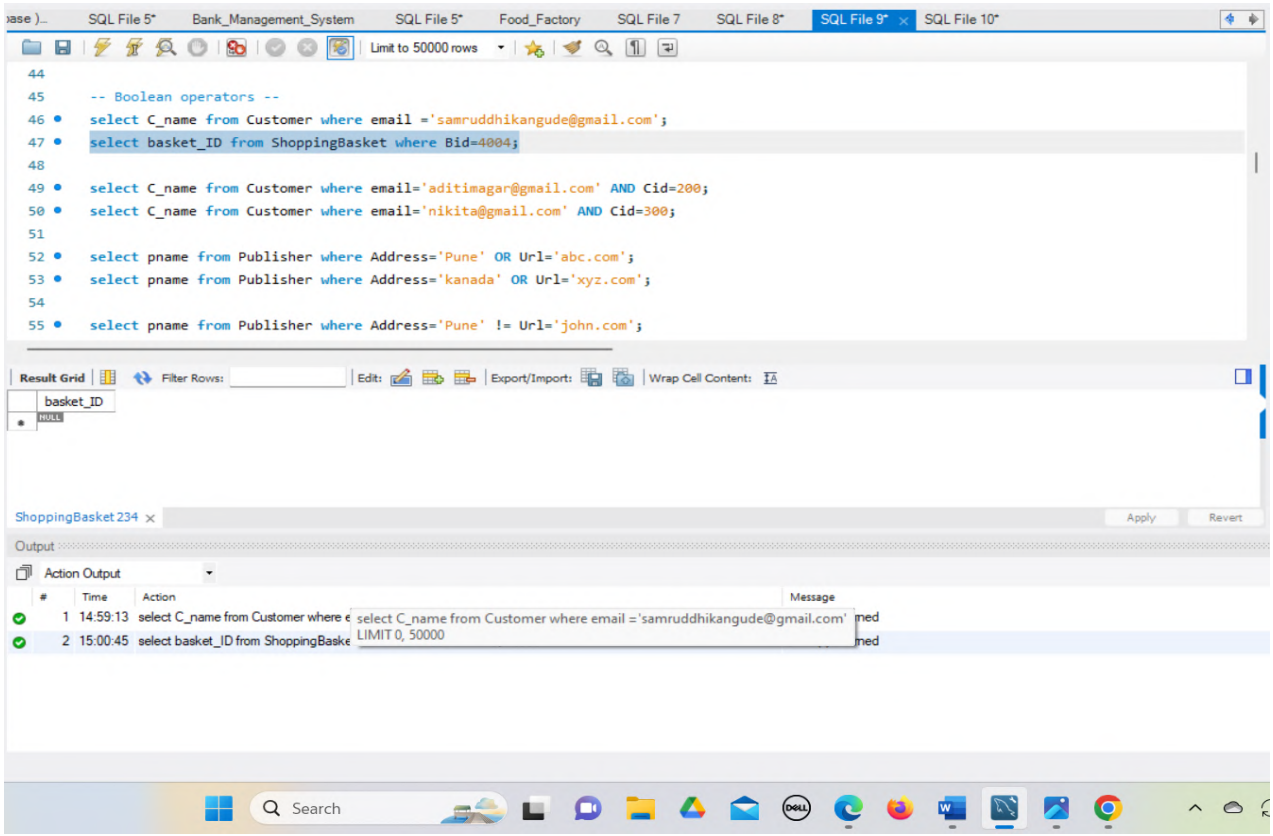
select pname from Publisher where Address='Pune' OR Url='abc.com';
select pname from Publisher where Address='kanada' OR Url='xyz.com';

select pname from Publisher where Address='Pune' != Url='john.com';

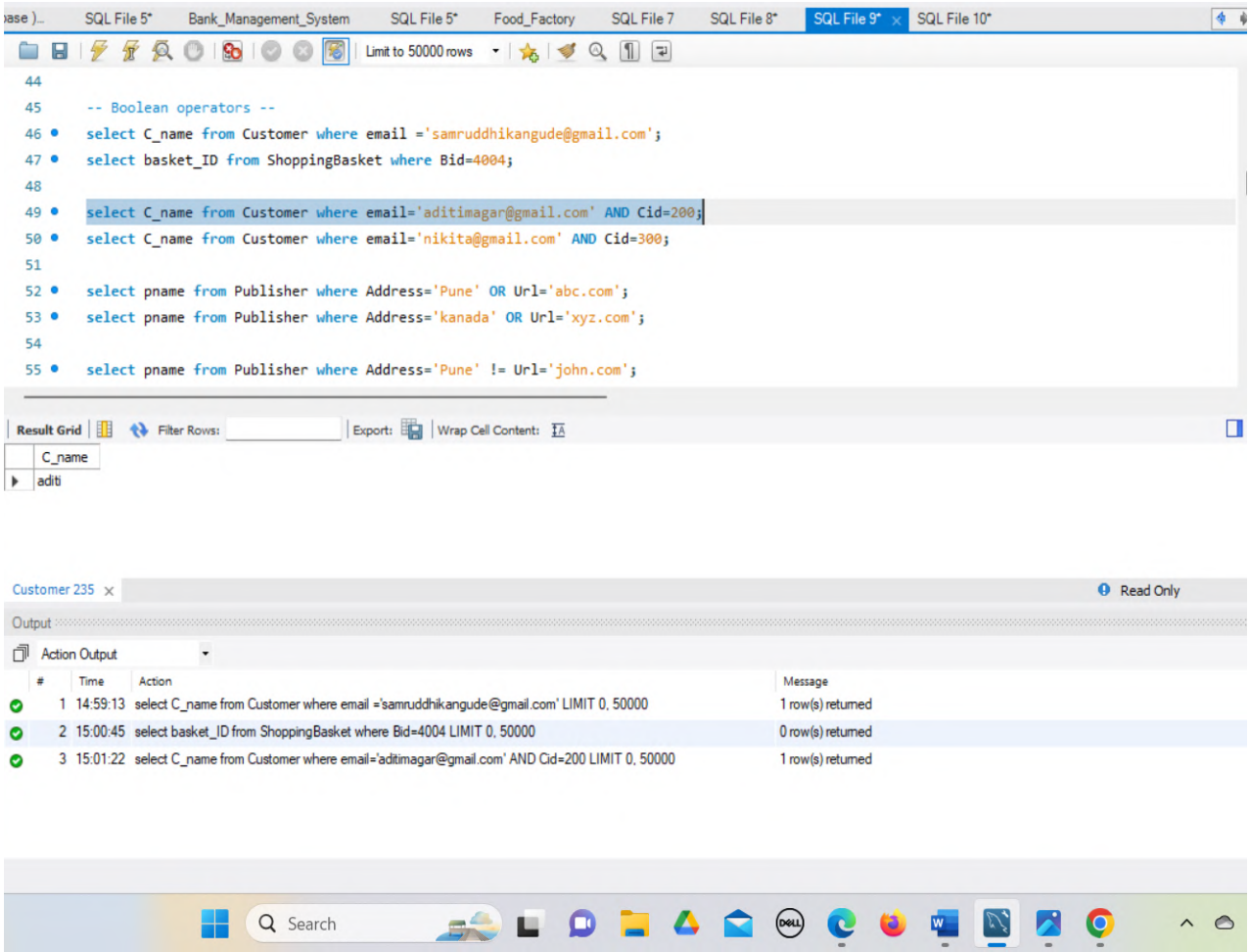
select C_name from Customer where email ='samruddhikangude@gmail.com';



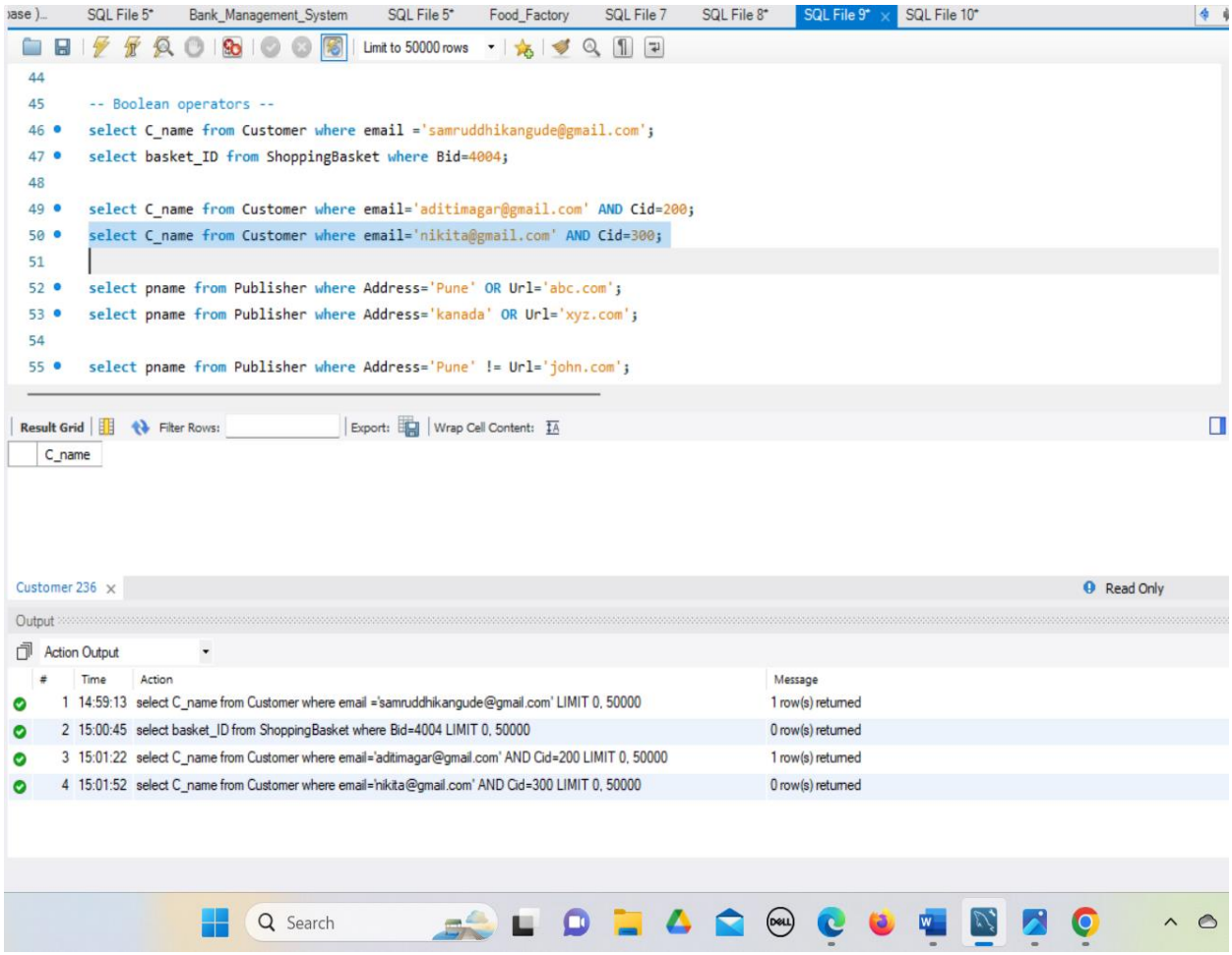
select basket_ID from ShoppingBasket where Bid=4004;



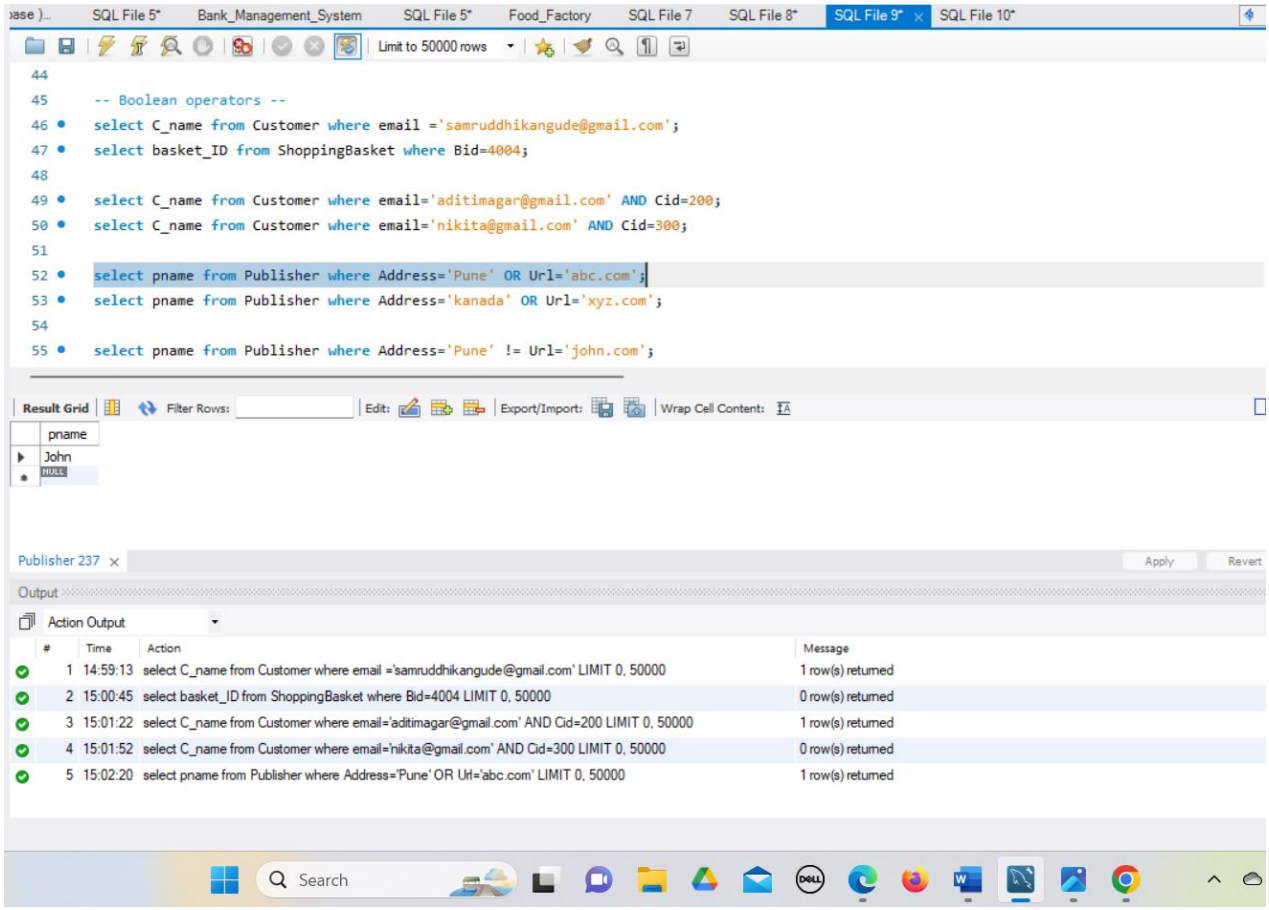
select C_name from Customer where email='aditimagar@gmail.com' AND Cid=200;



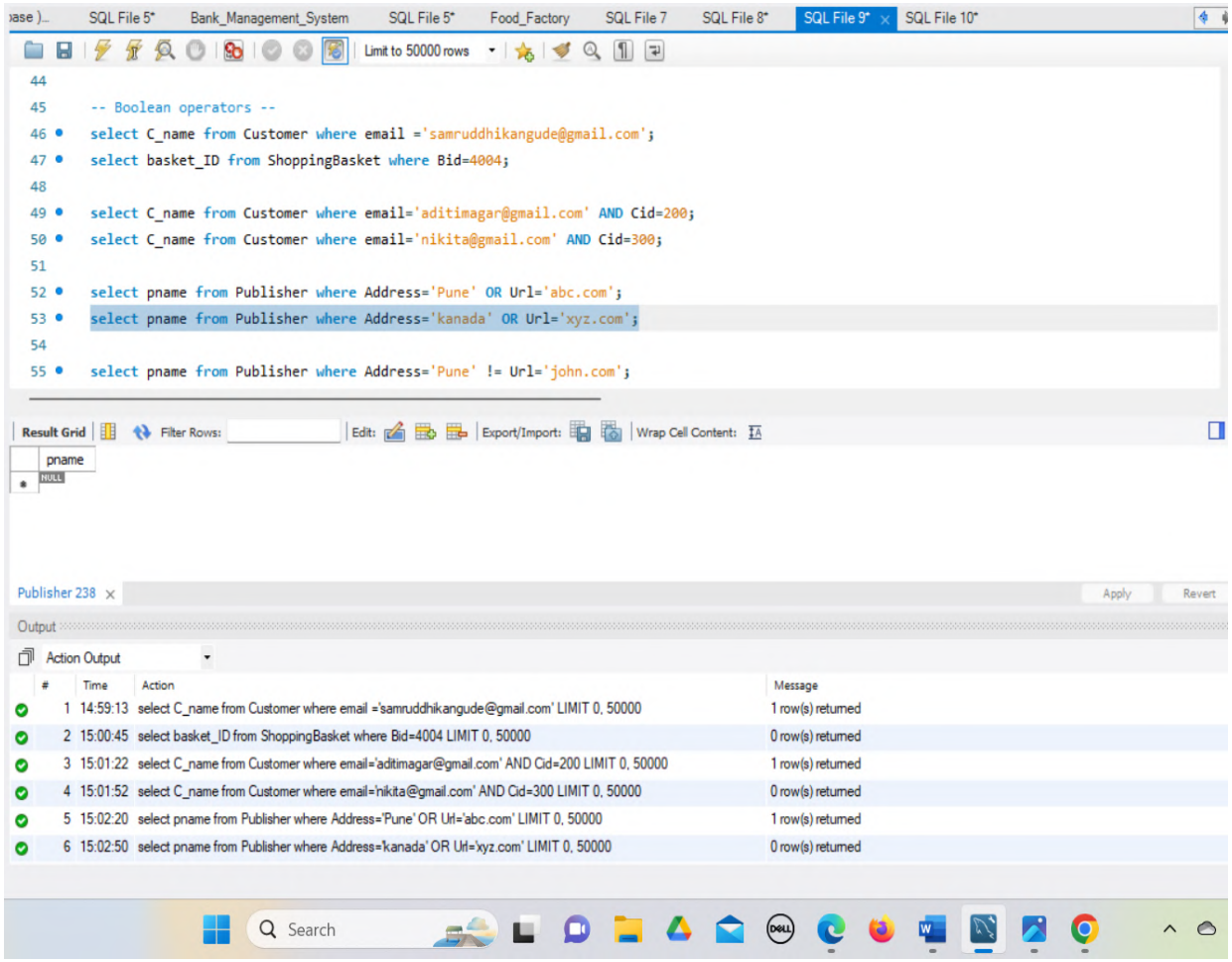
select C_name from Customer where email='nikita@gmail.com' AND Cid=300;



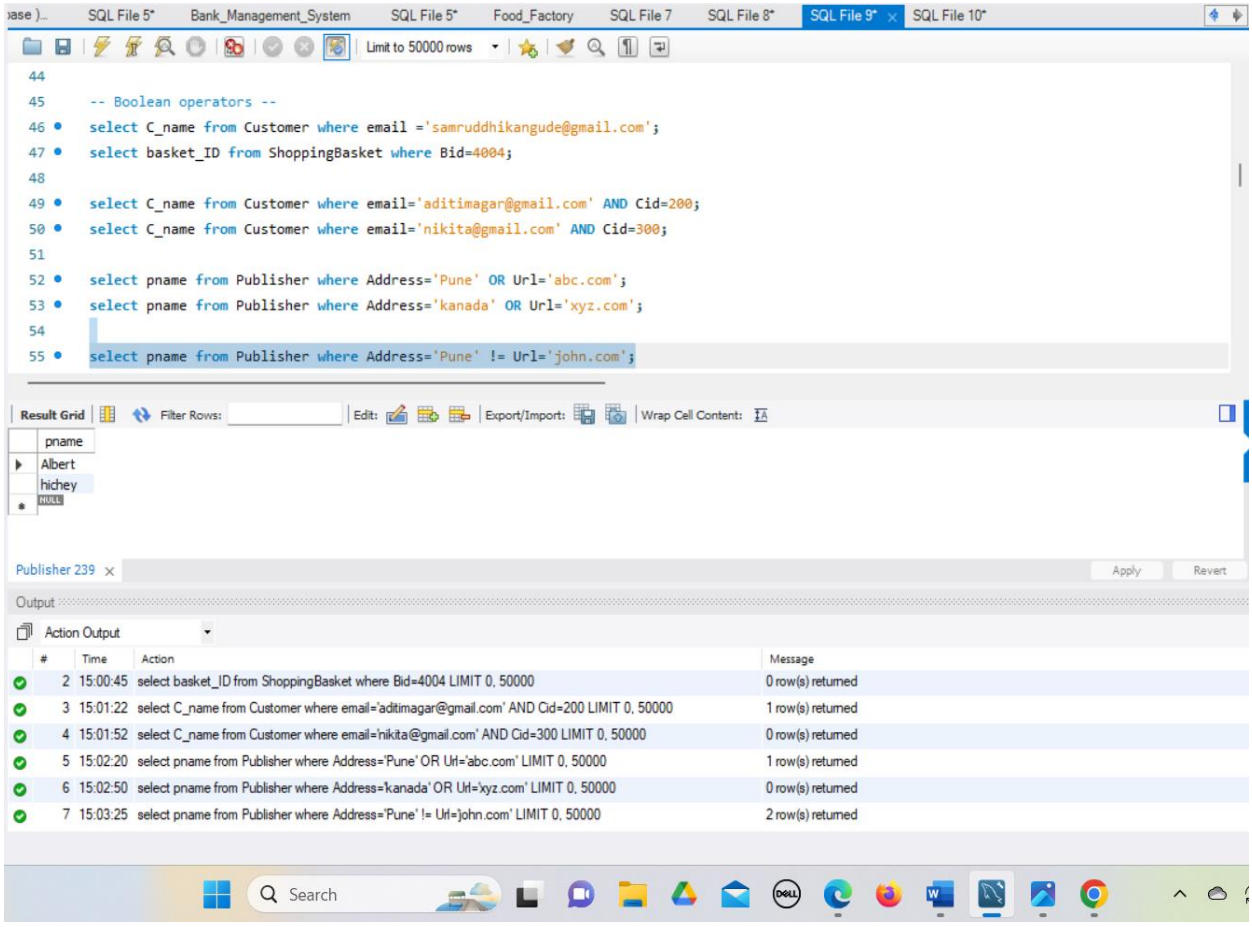
select pname from Publisher where Address='Pune' OR Url='abc.com';



select pname from Publisher where Address='kanada' OR Url='xyz.com';



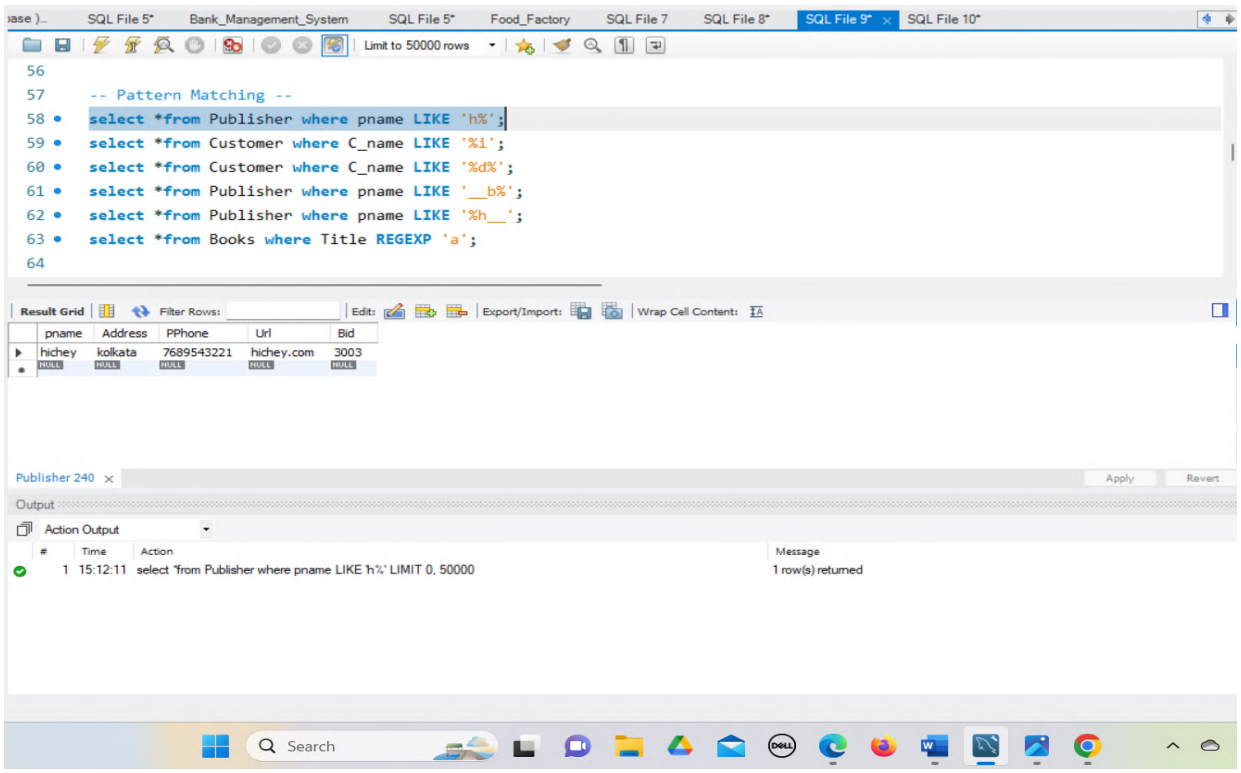
select pname from Publisher where Address='Pune' != Url='john.com';



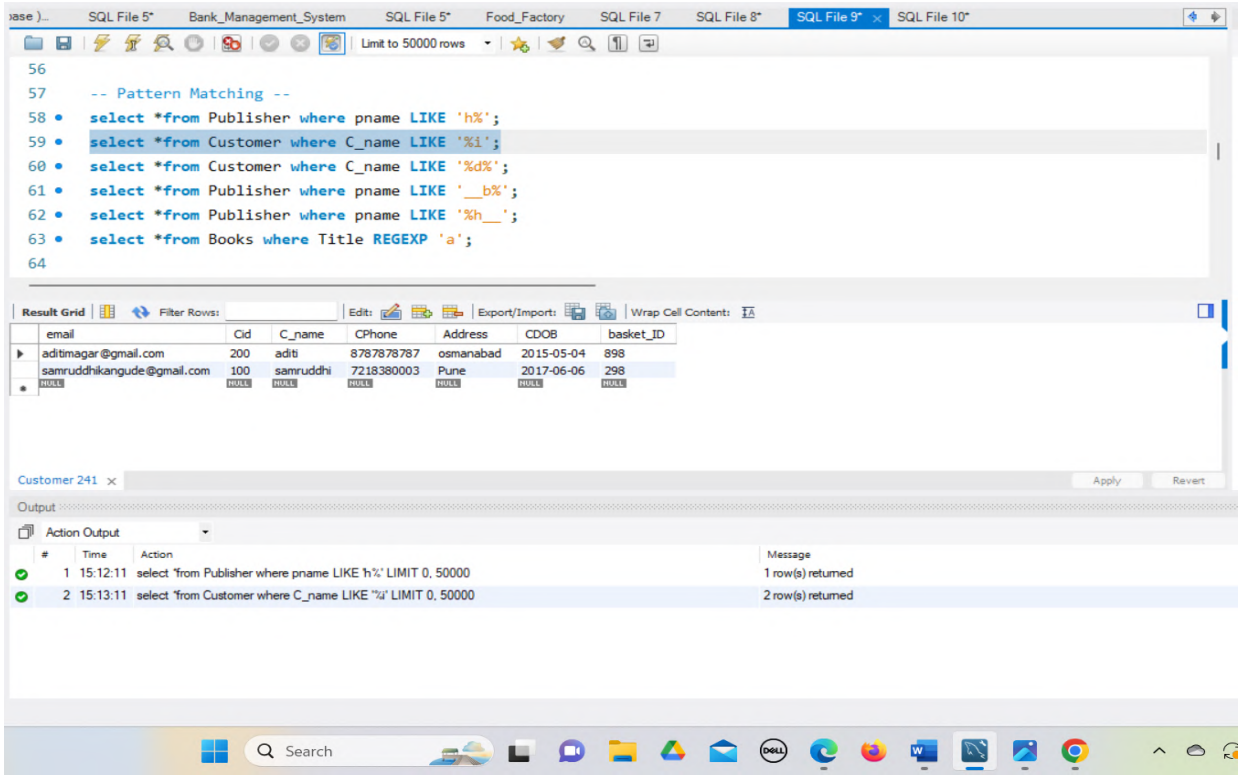
-- Pattern Matching --

```
select *from Publisher where pname LIKE 'h%';
select *from Customer where C_name LIKE '%i';
select *from Customer where C_name LIKE '%d%';
select *from Publisher where pname LIKE '__b%';
select *from Publisher where pname LIKE '%h__';
select *from Books where Title REGEXP 'a';
```

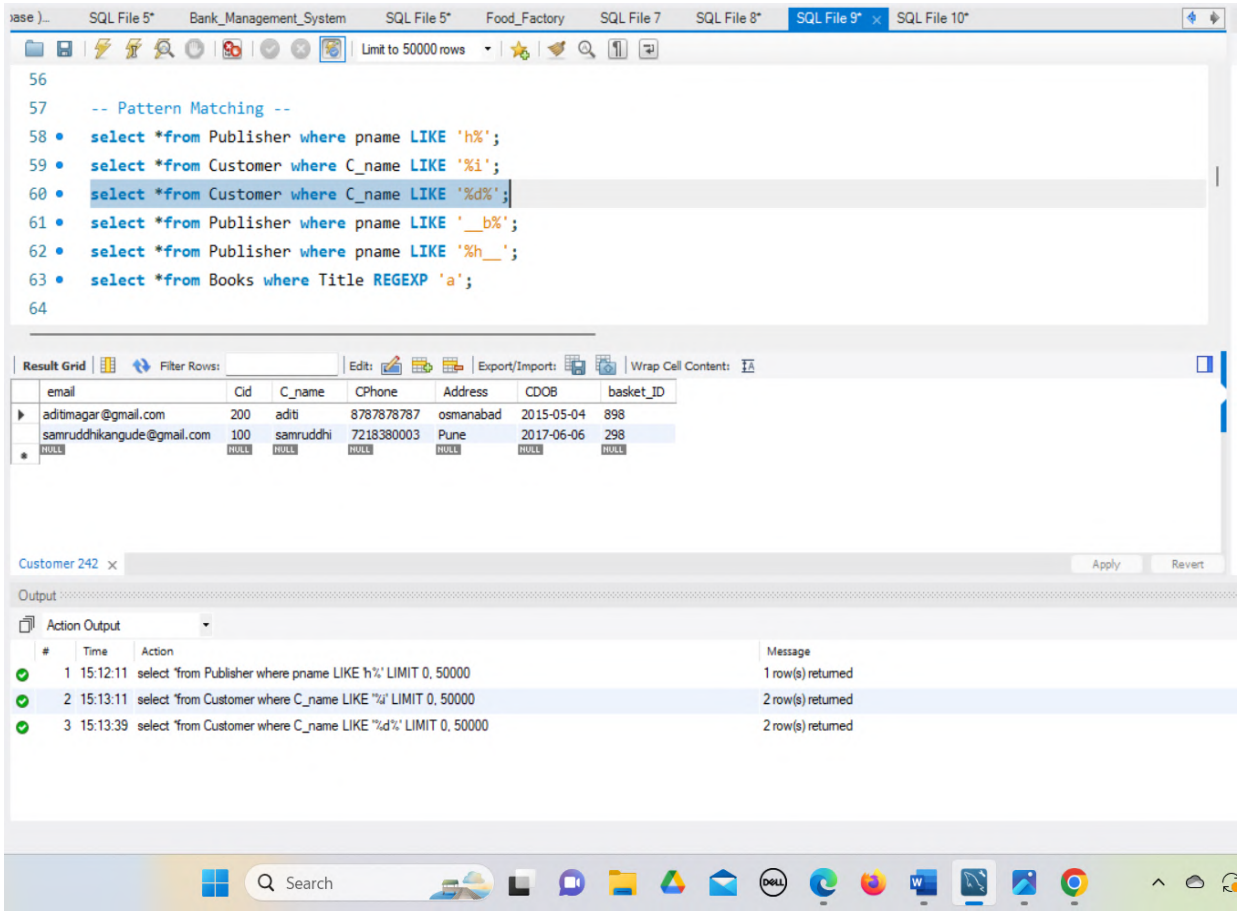
select *from Publisher where pname LIKE 'h%';



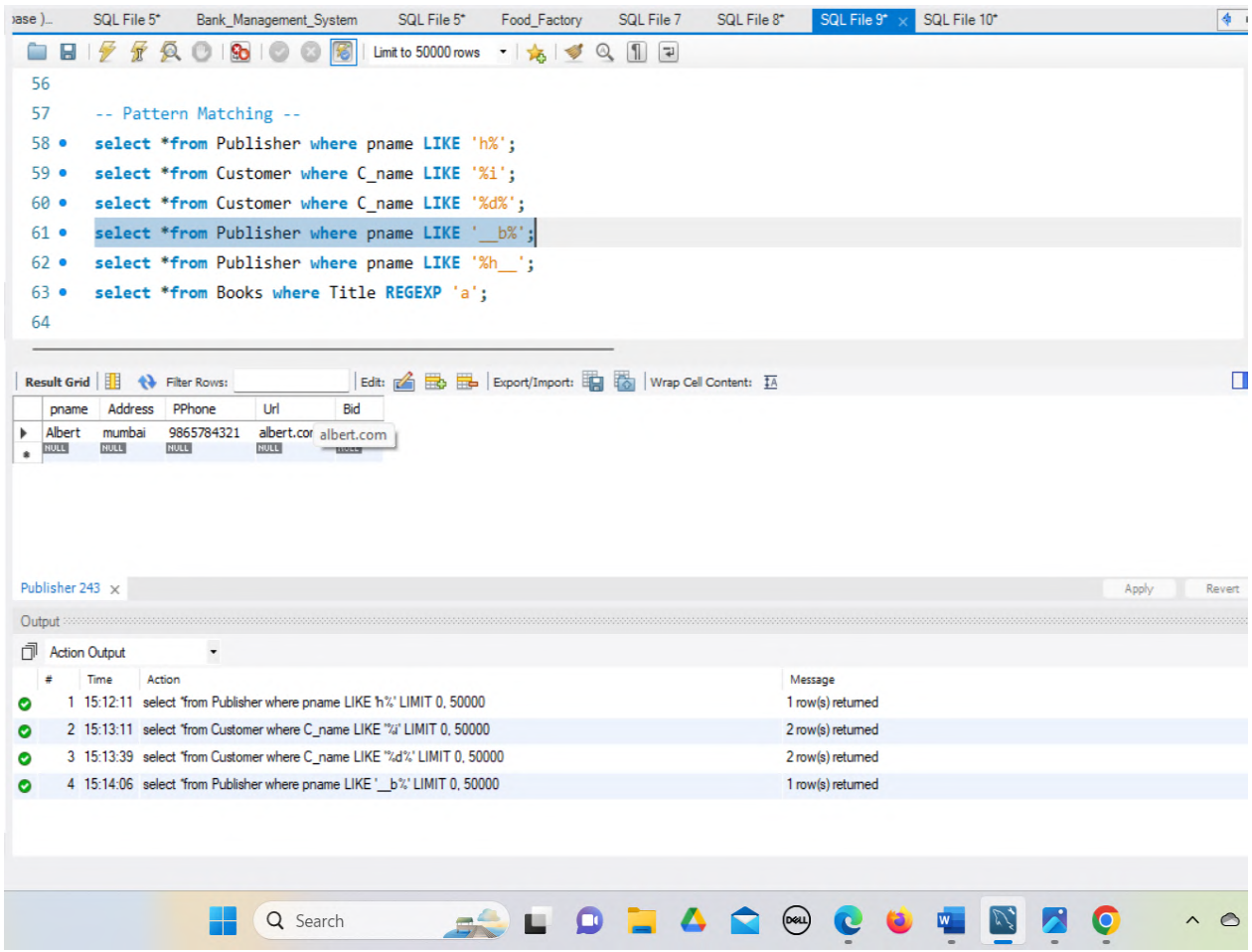
select *from Customer where C_name LIKE '%i';



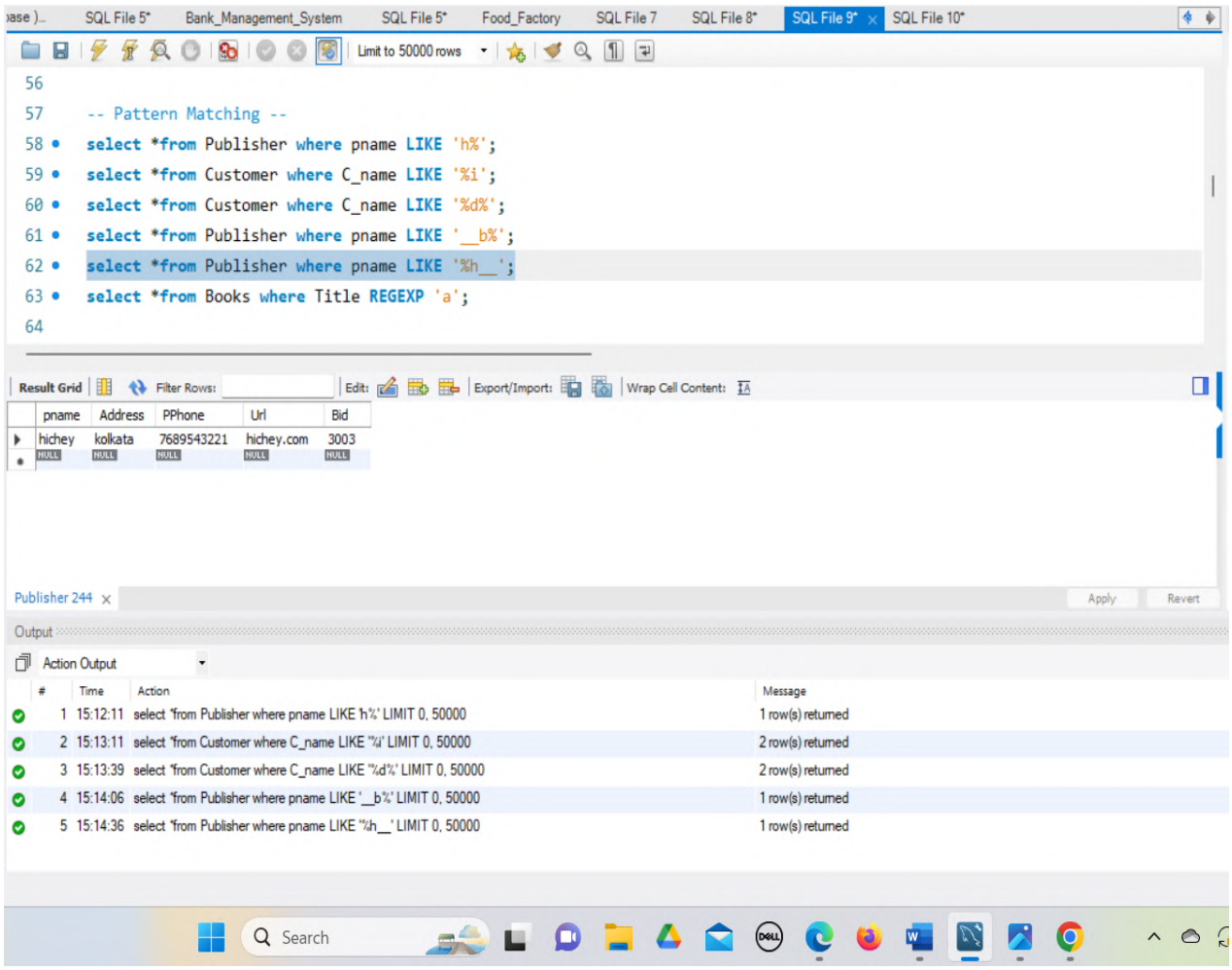
select *from Customer where C_name LIKE '%d%';



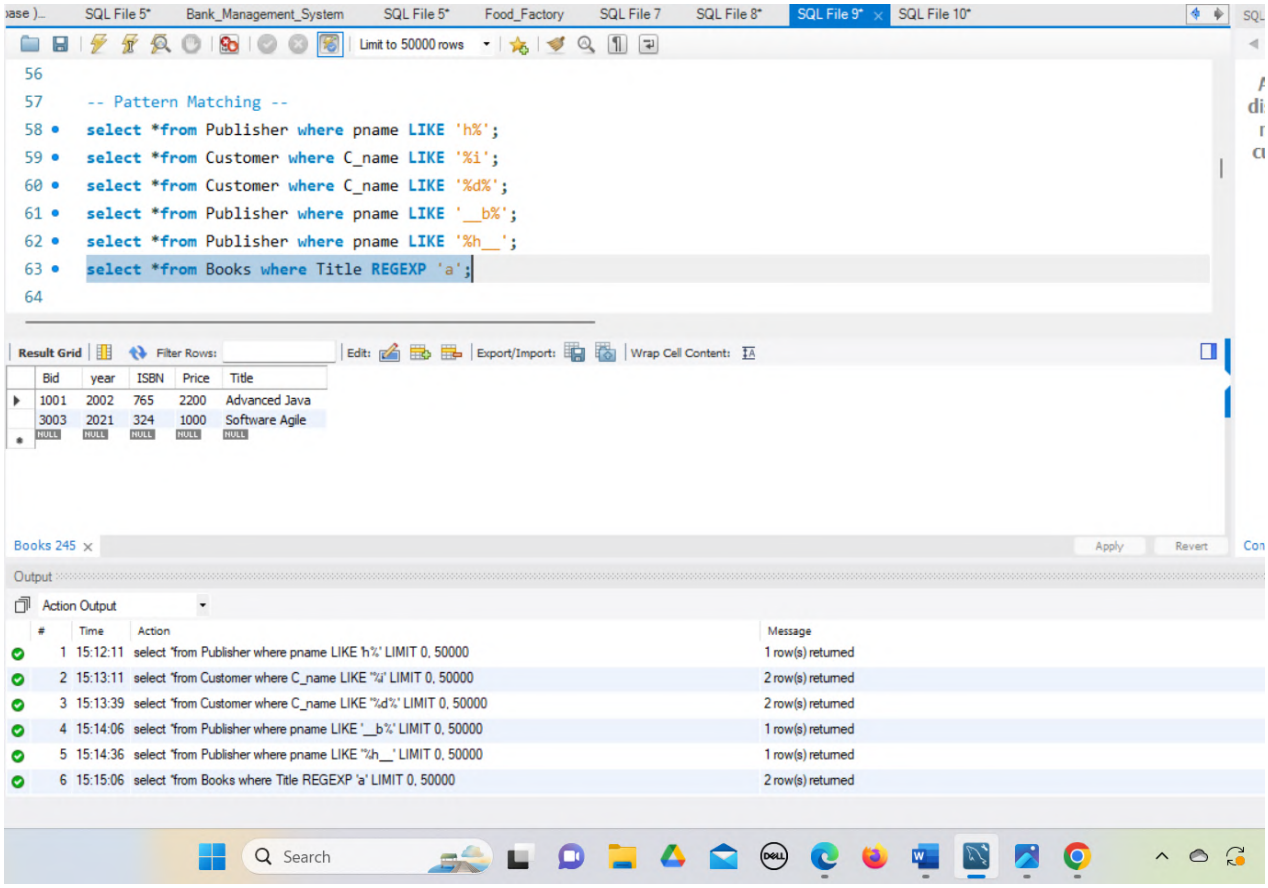
select *from Publisher where pname LIKE '__b%';



select *from Publisher where pname LIKE '%h__';



select *from Books where Title REGEXP 'a';

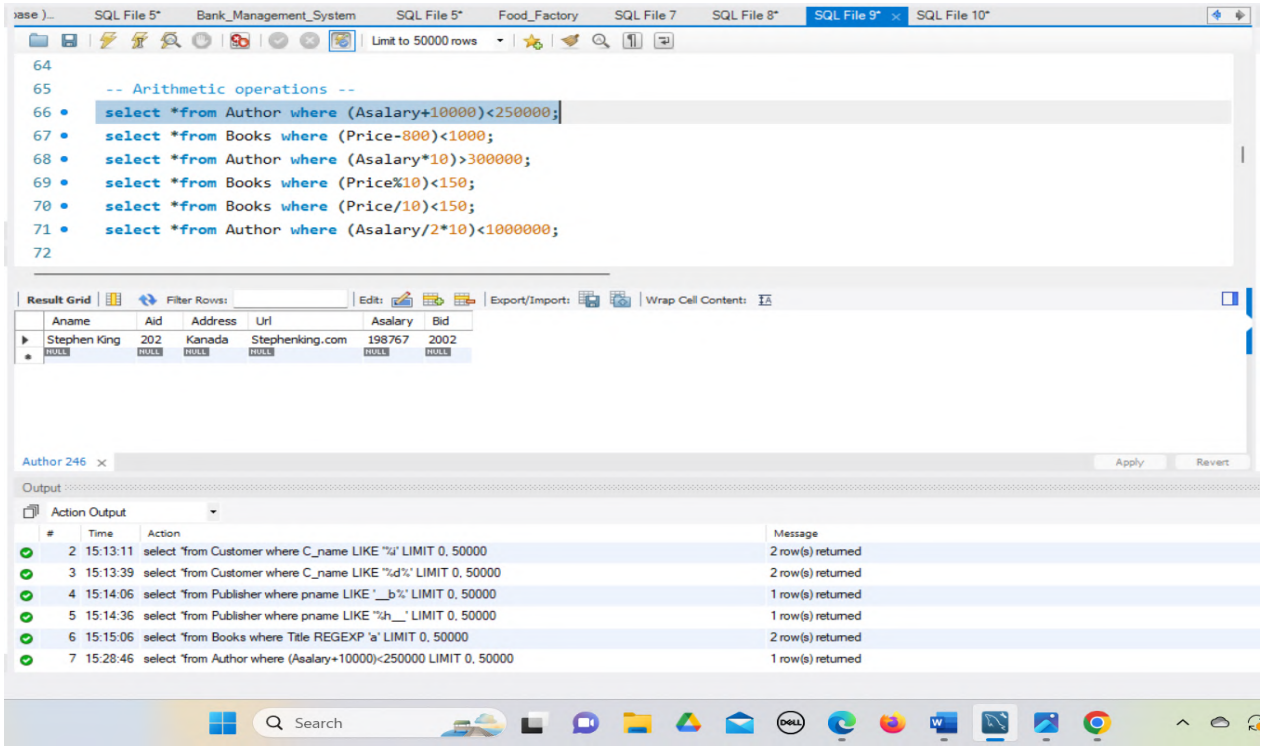


C: Arithmetic operations and built in functions:

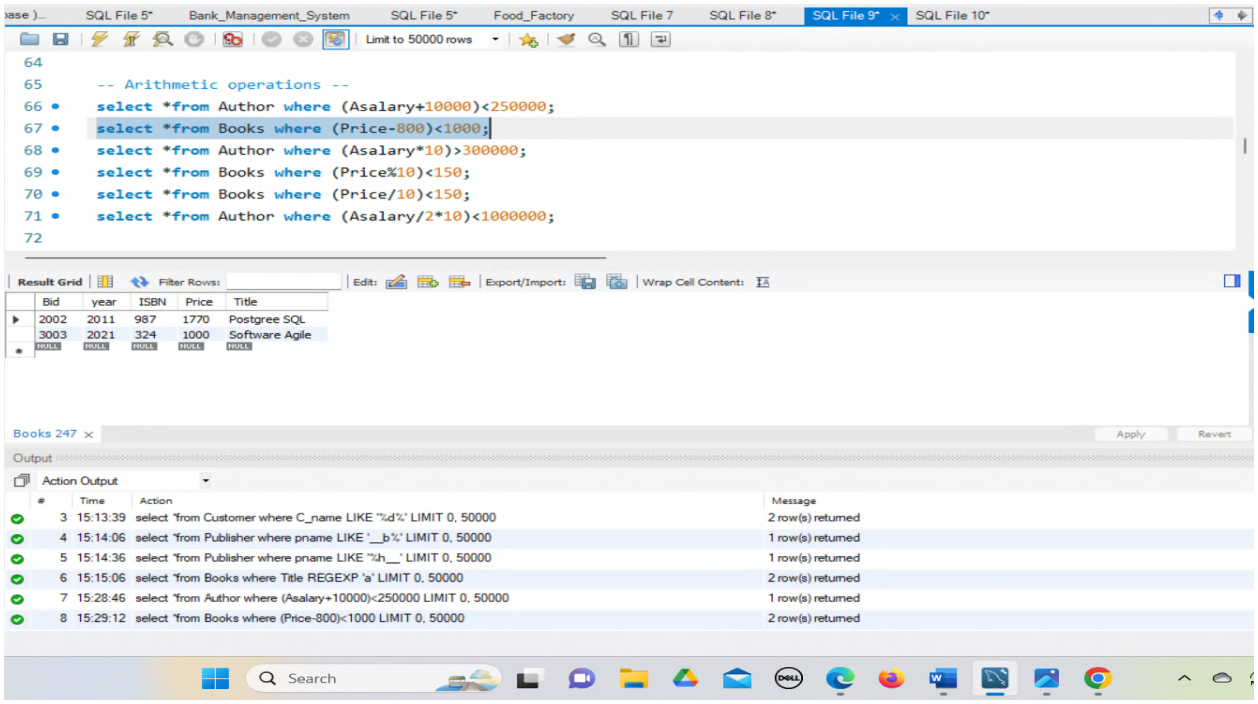
-- Arithmetic operations --

```
select *from Author where (Asalary+10000)<250000;
select *from Books where (Price-800)<1000;
select *from Author where (Asalary*10)>300000;
select *from Books where (Price%10)<150;
select *from Books where (Price/10)<150;
select *from Author where (Asalary/2*10)<1000000;
```

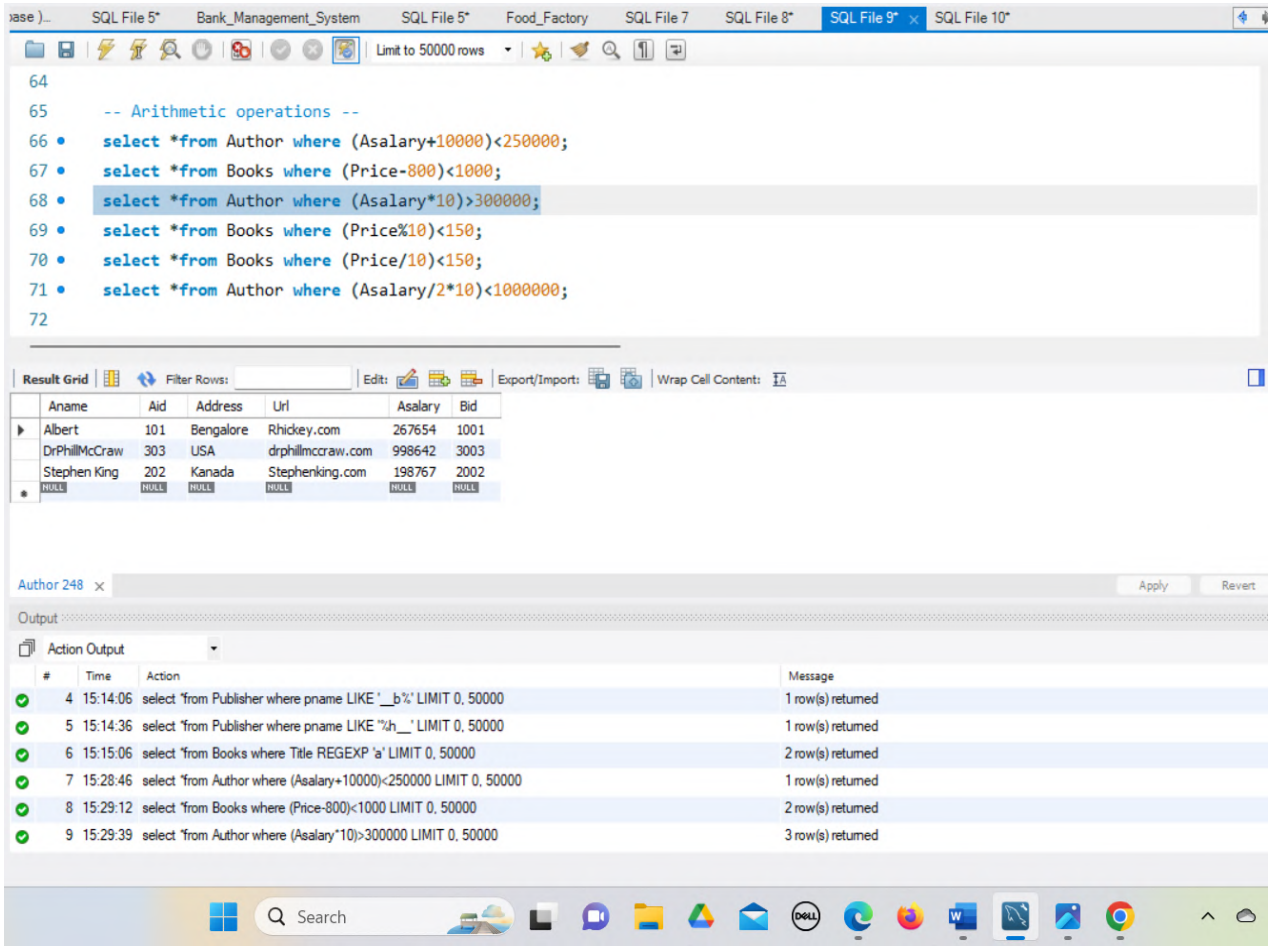
select *from Author where (Asalary+10000)<250000;



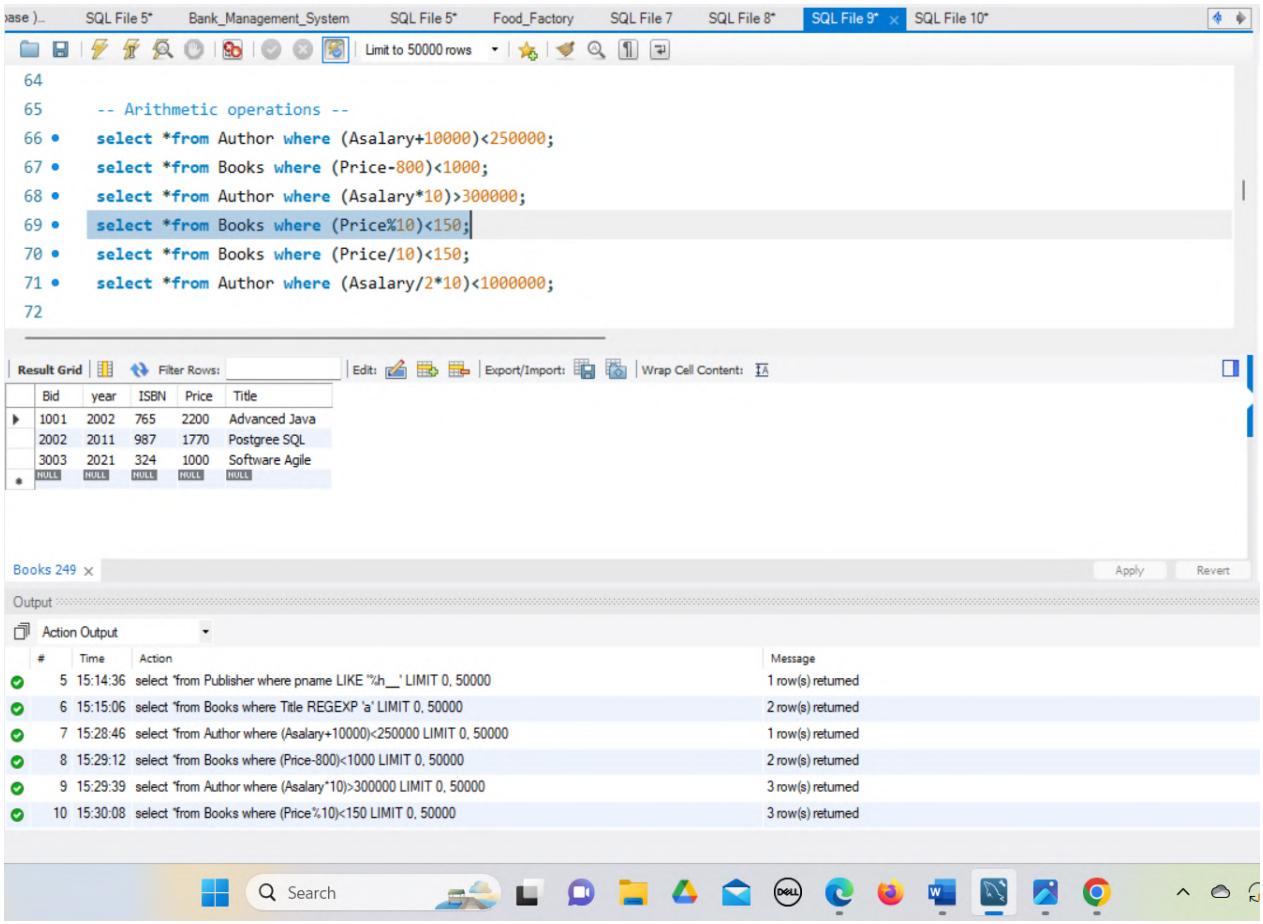
select *from Books where (Price-800)<1000;



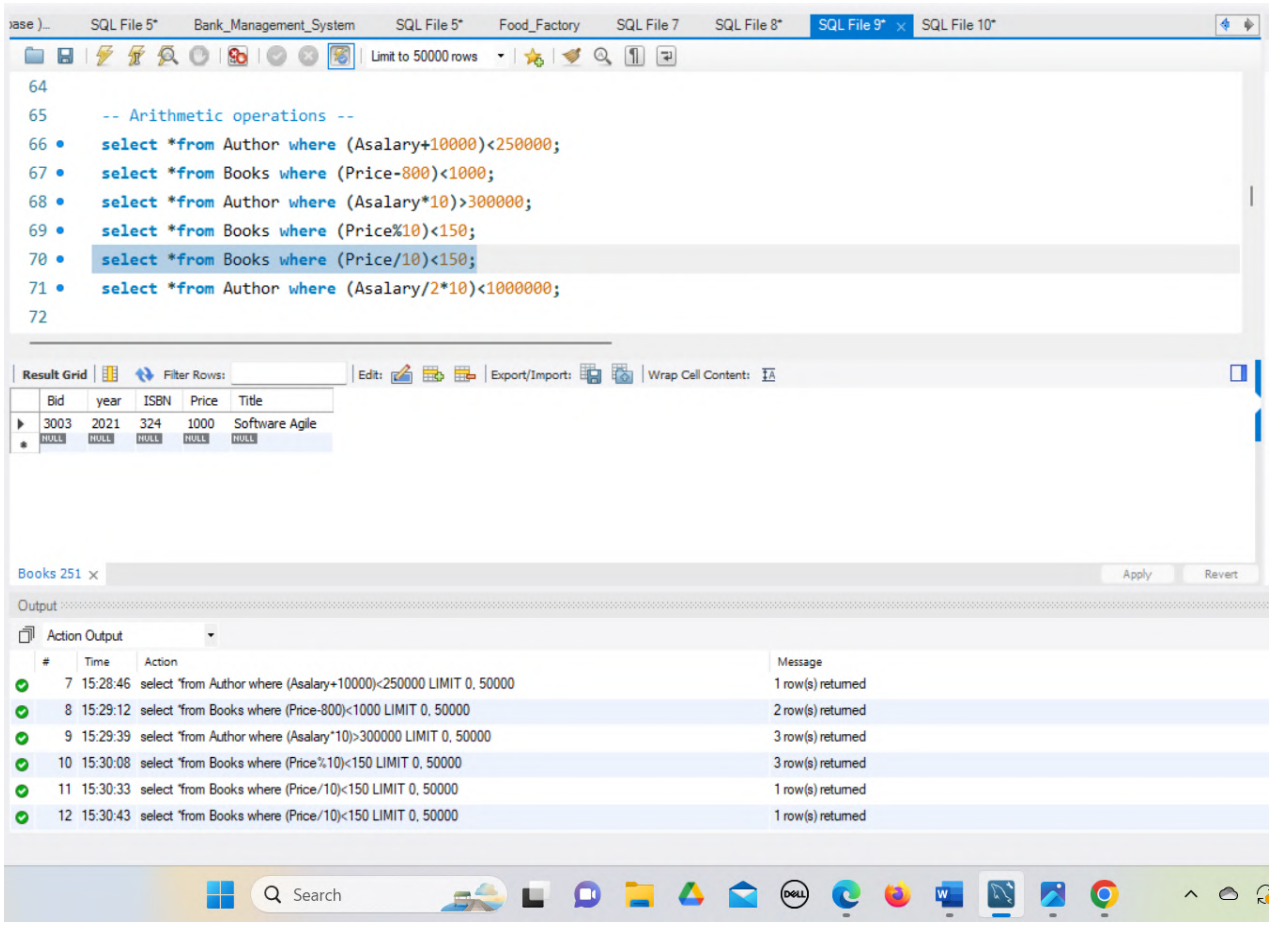
select *from Author where (Asalary*10)>300000;



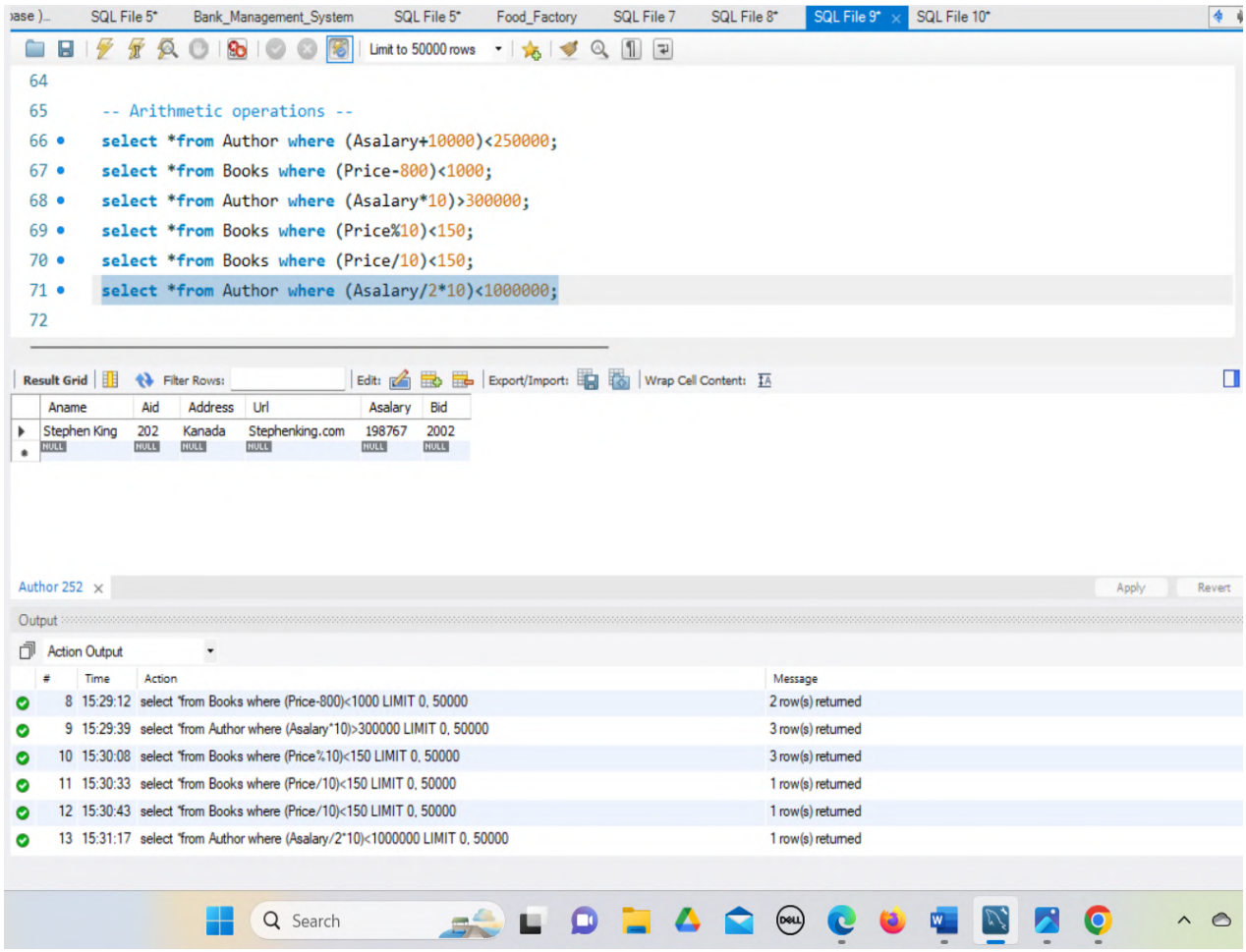
select *from Books where (Price%10)<150;



select *from Books where (Price/10)<150;



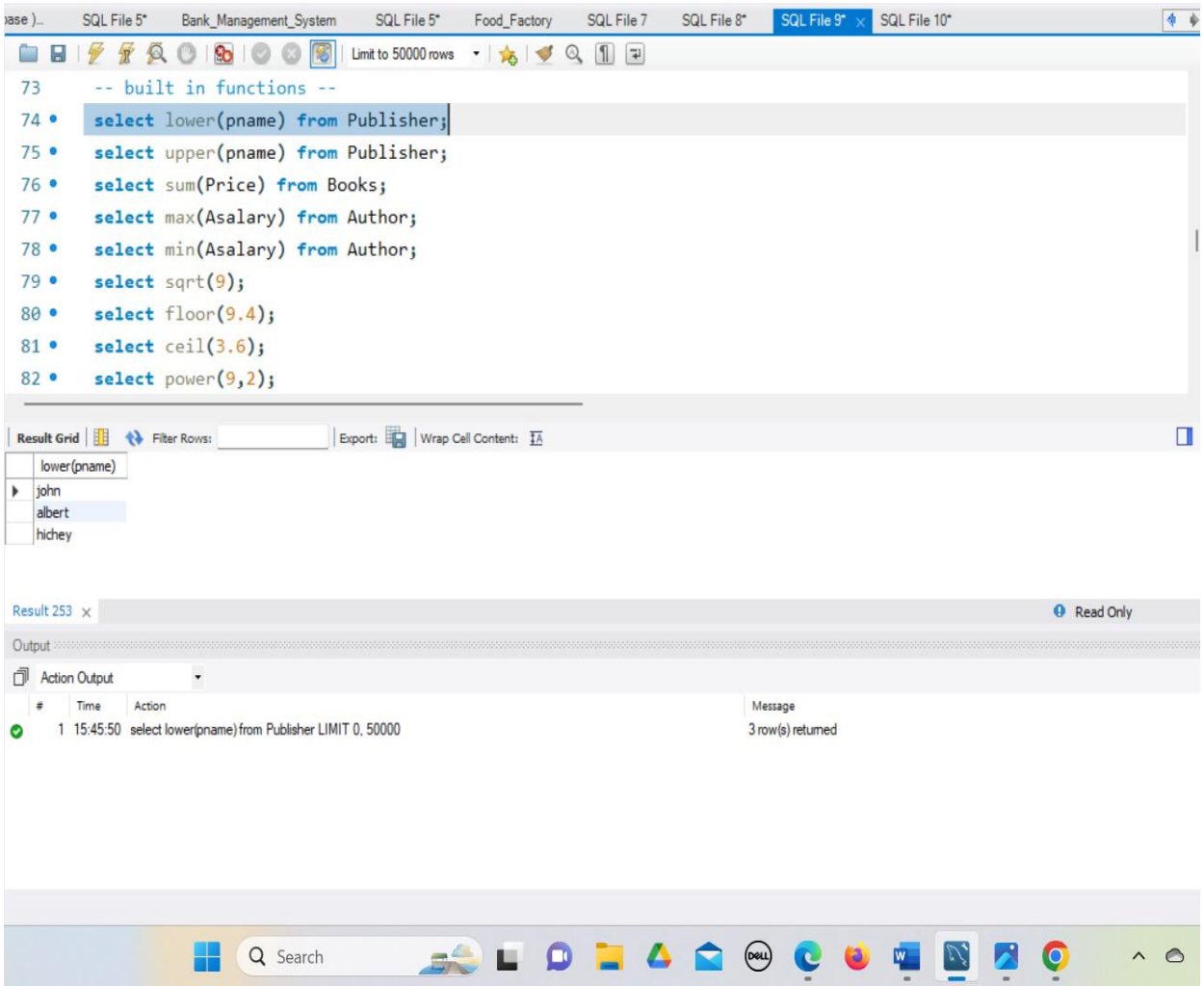
select *from Author where (Asalary/2*10)<1000000;



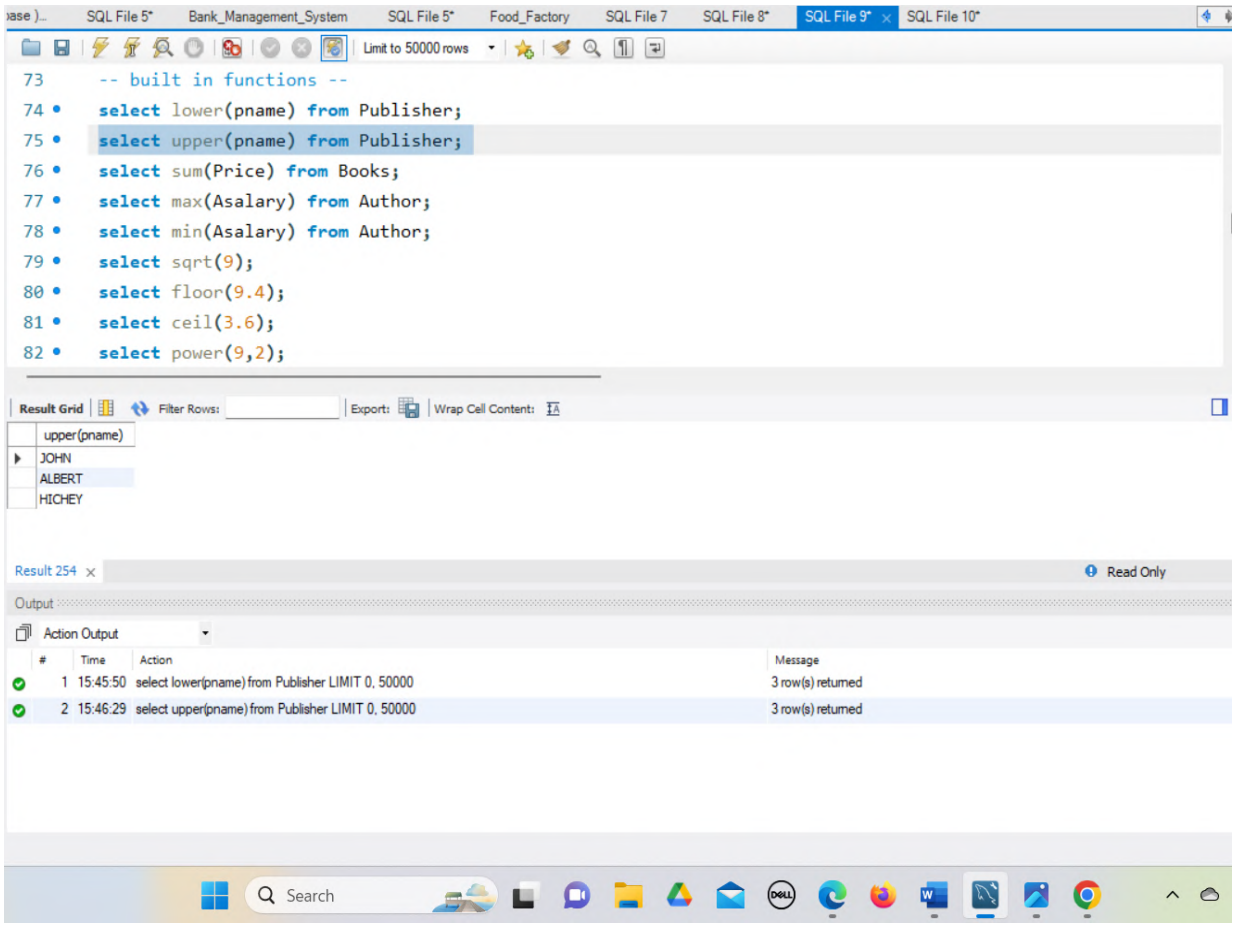
-- built in functions --

```
select lower(pname) from Publisher;
select upper(pname) from Publisher;
select sum(Price) from Books;
select max(Asalary) from Author;
select min(Asalary) from Author;
select sqrt(9);
select floor(9.4);
select ceil(3.6);
select power(9,2);
select strcmp('samruddhi','aditi');
select char_length('samruddhi');
select concat('Samruddhi',' kangude');
select instr('AIML DEPARTMENT','L');
select mid('AIML DEPARTMENT IN ALARD',5,11);
select reverse('SAMRUDDHI');
select space(100);
```

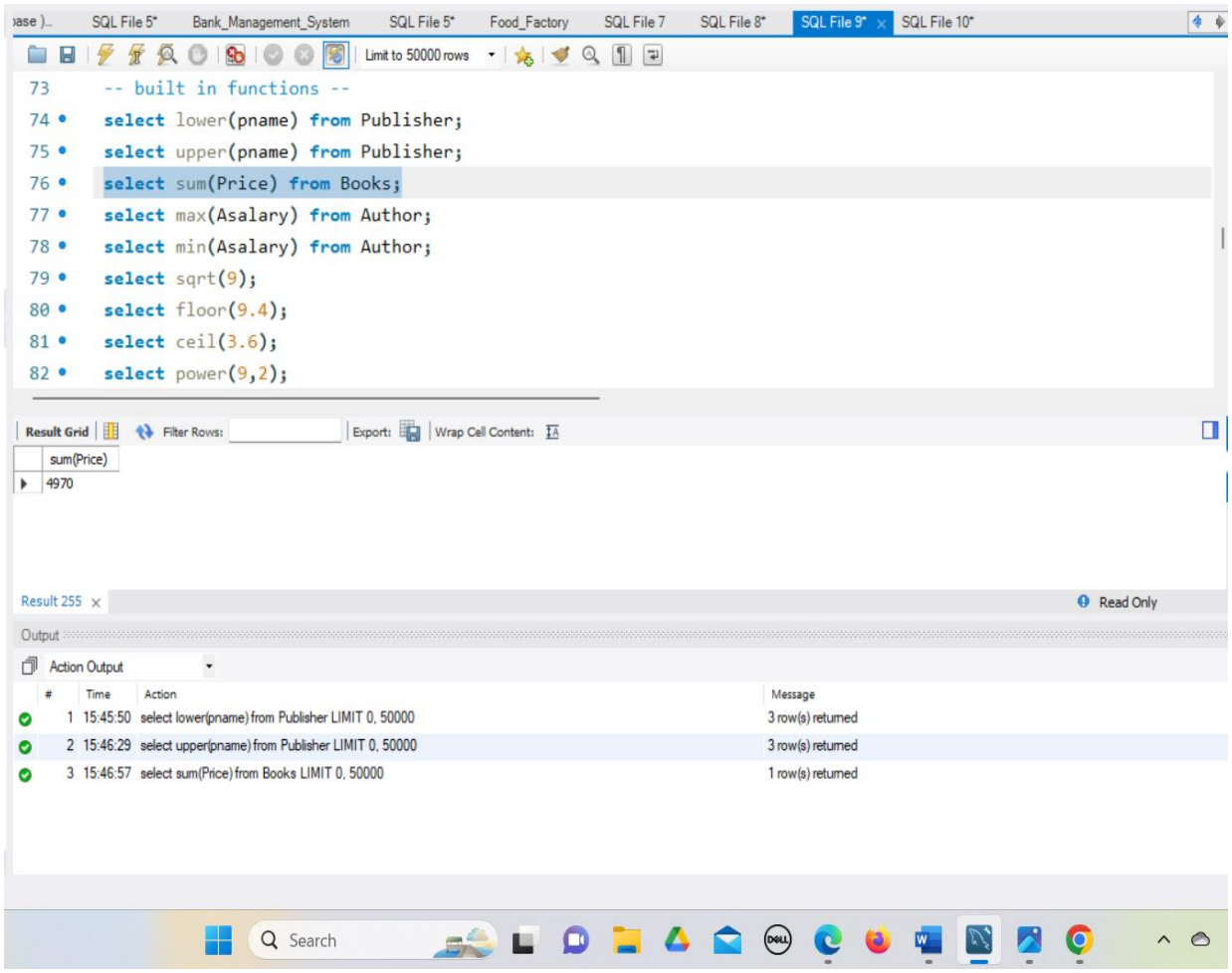
select lower(pname) from Publisher;



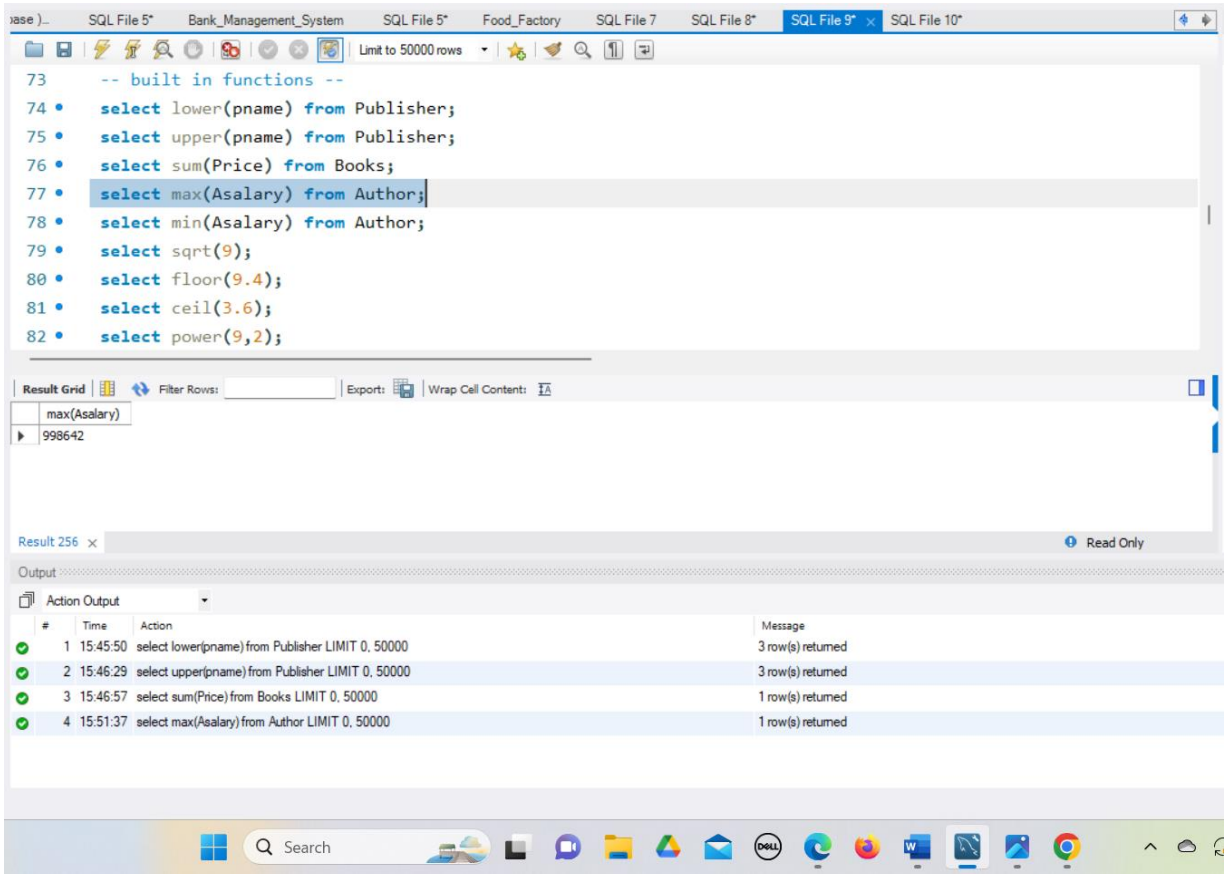
select upper(pname) from Publisher;



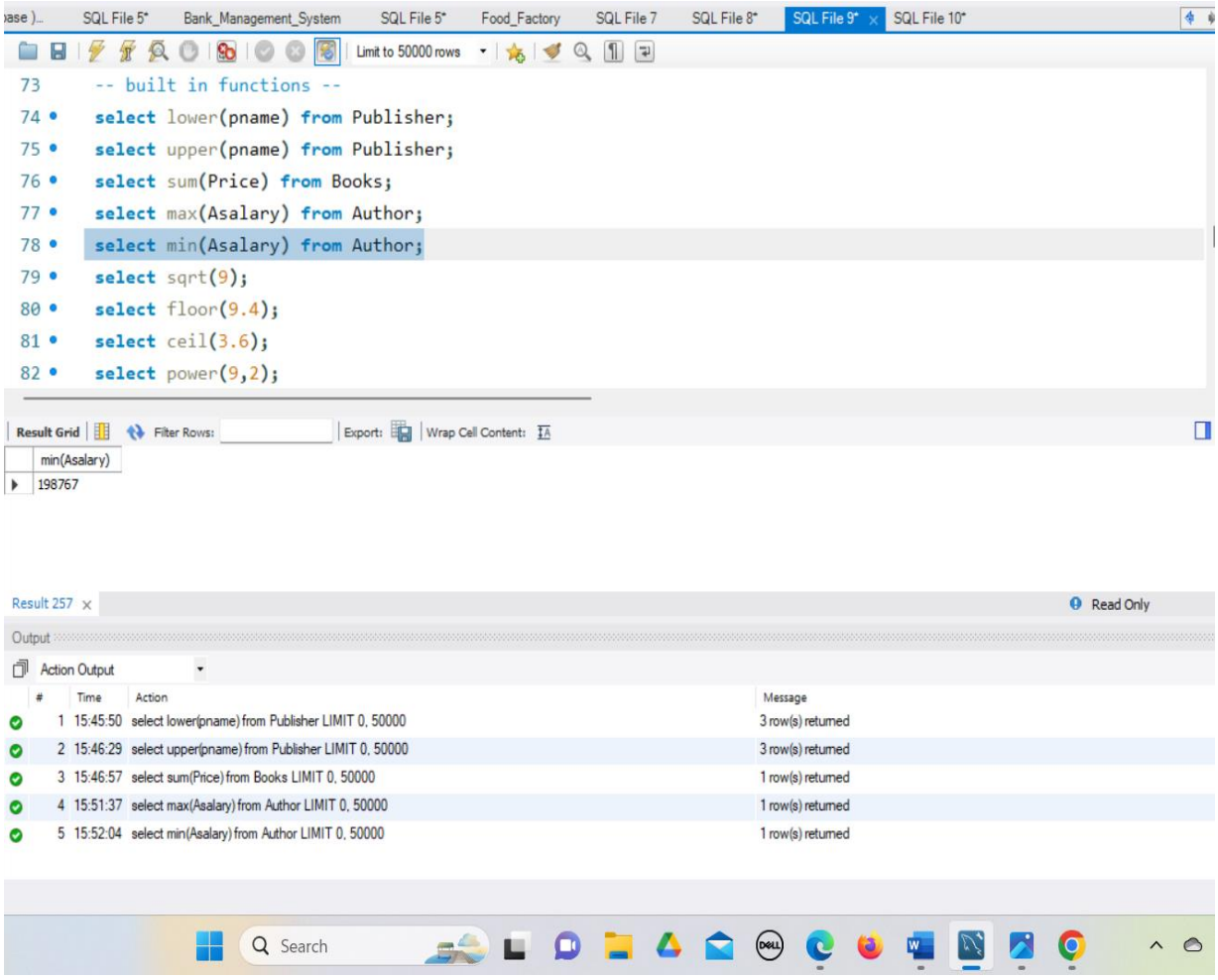
select sum(Price) from Books;



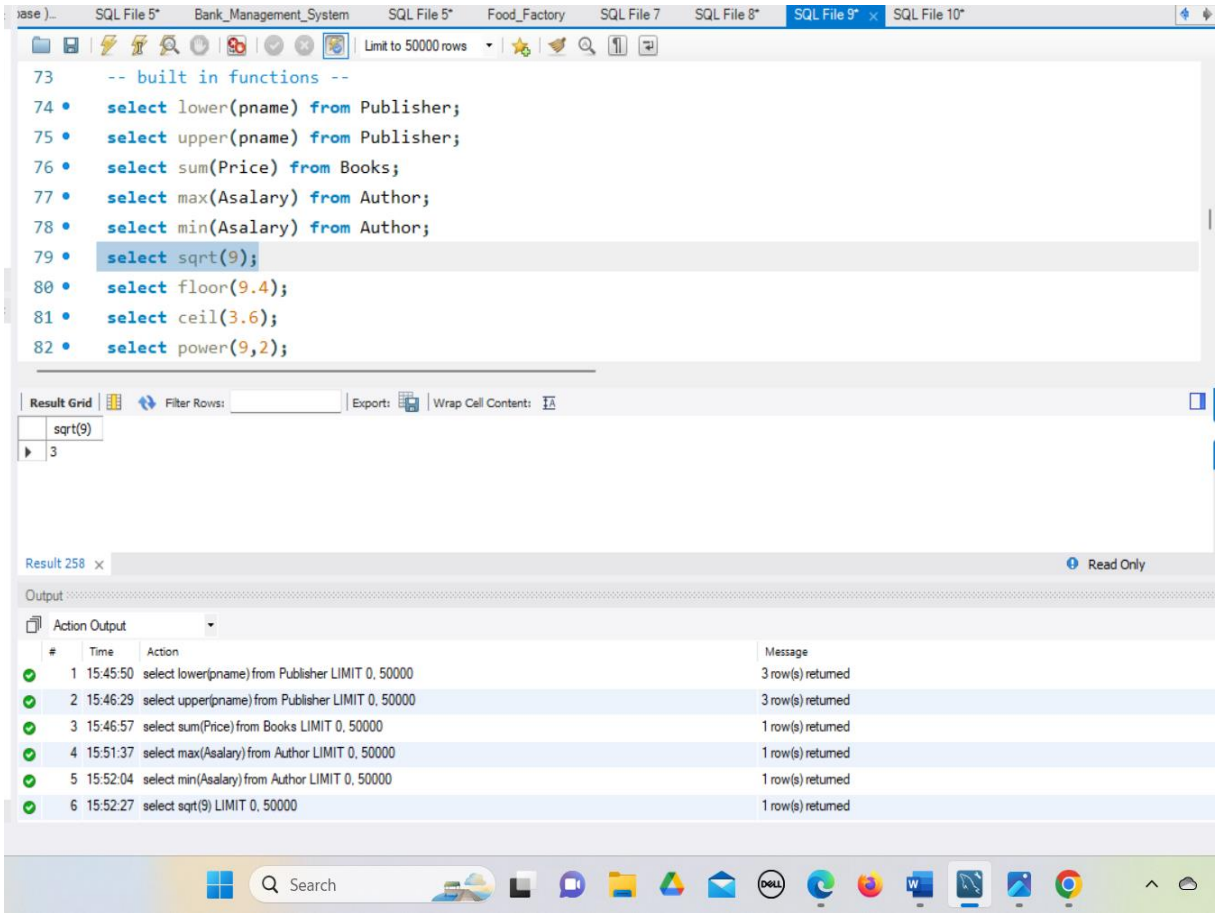
select max(Asalary) from Author;



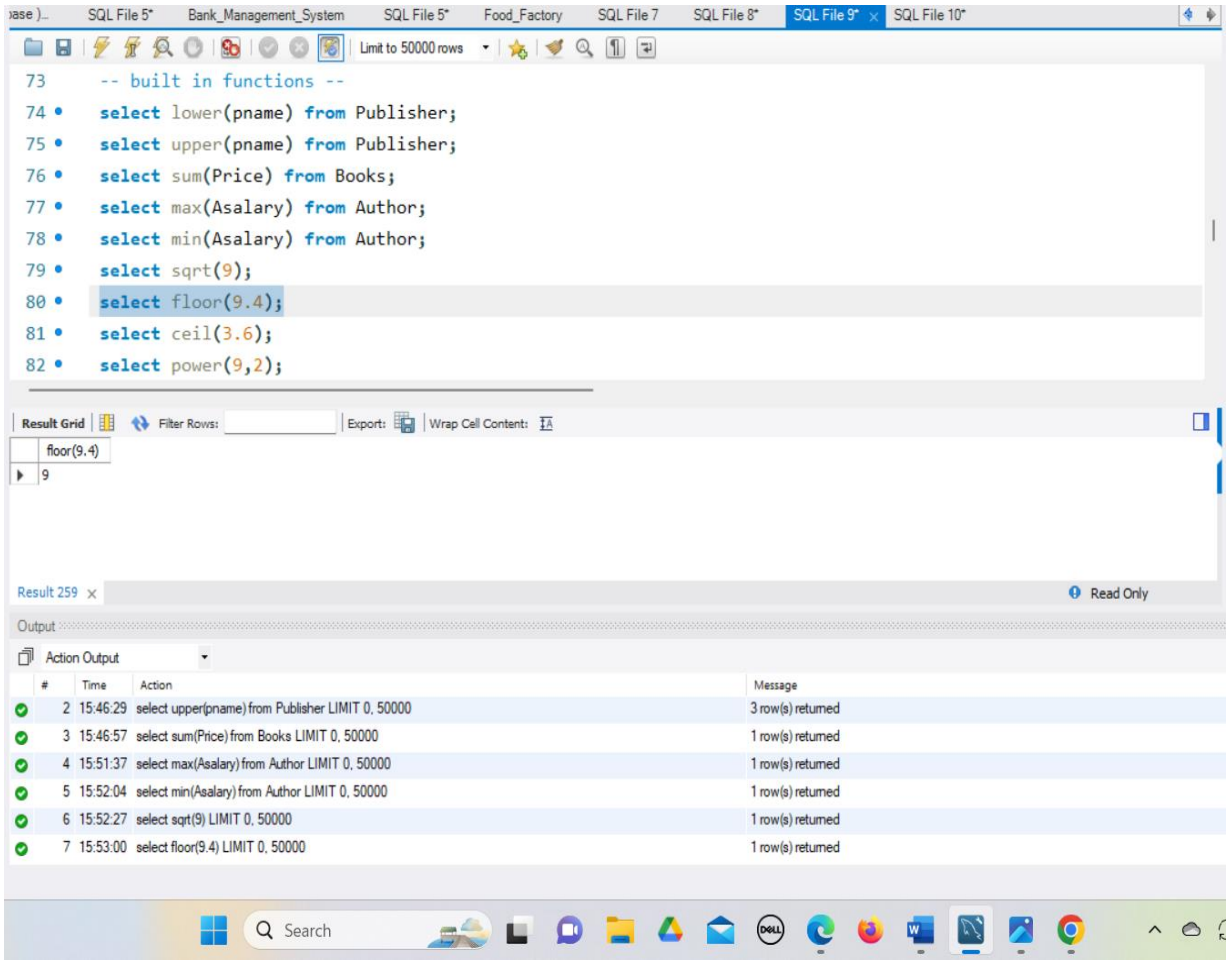
select min(Asalary) from Author;



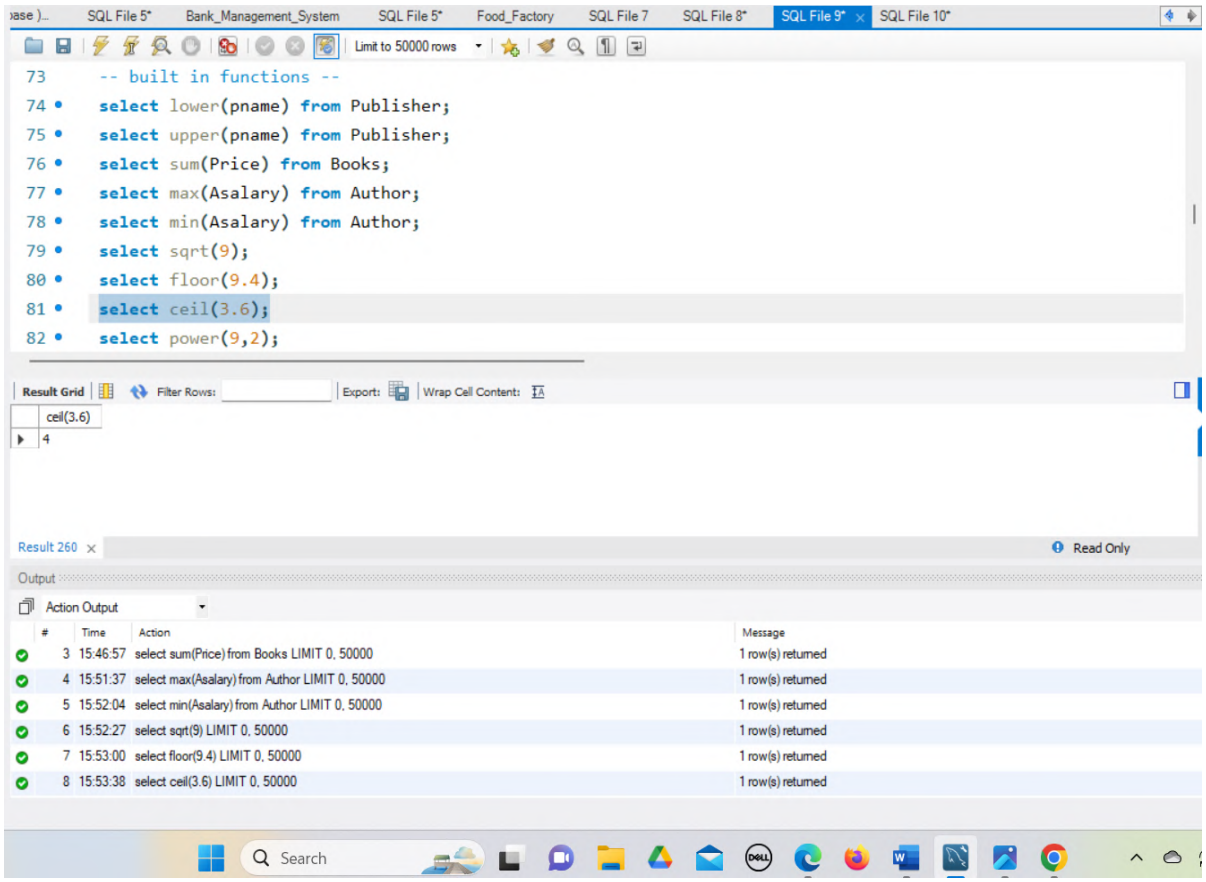
select sqrt(9);



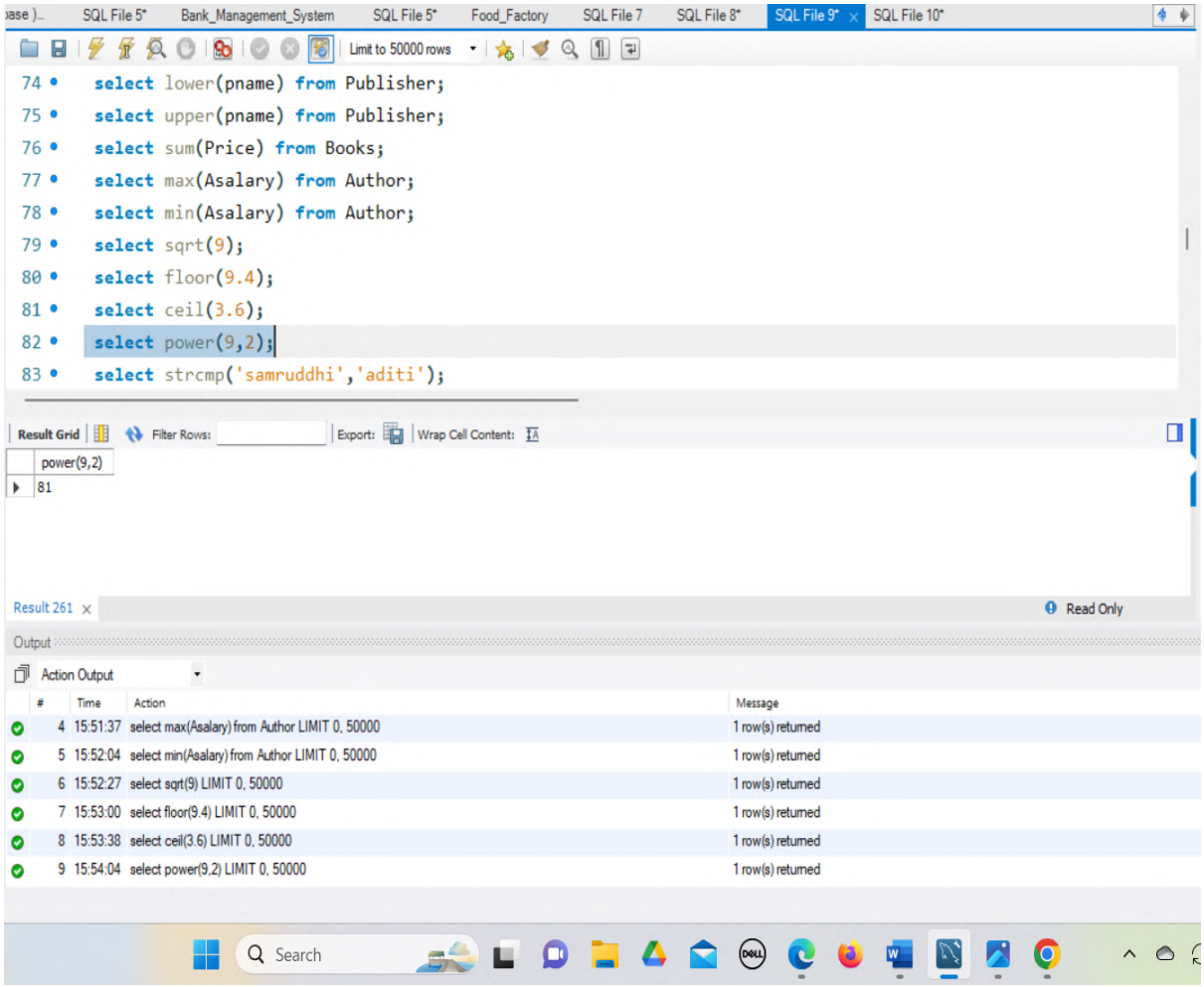
select floor(9.4);



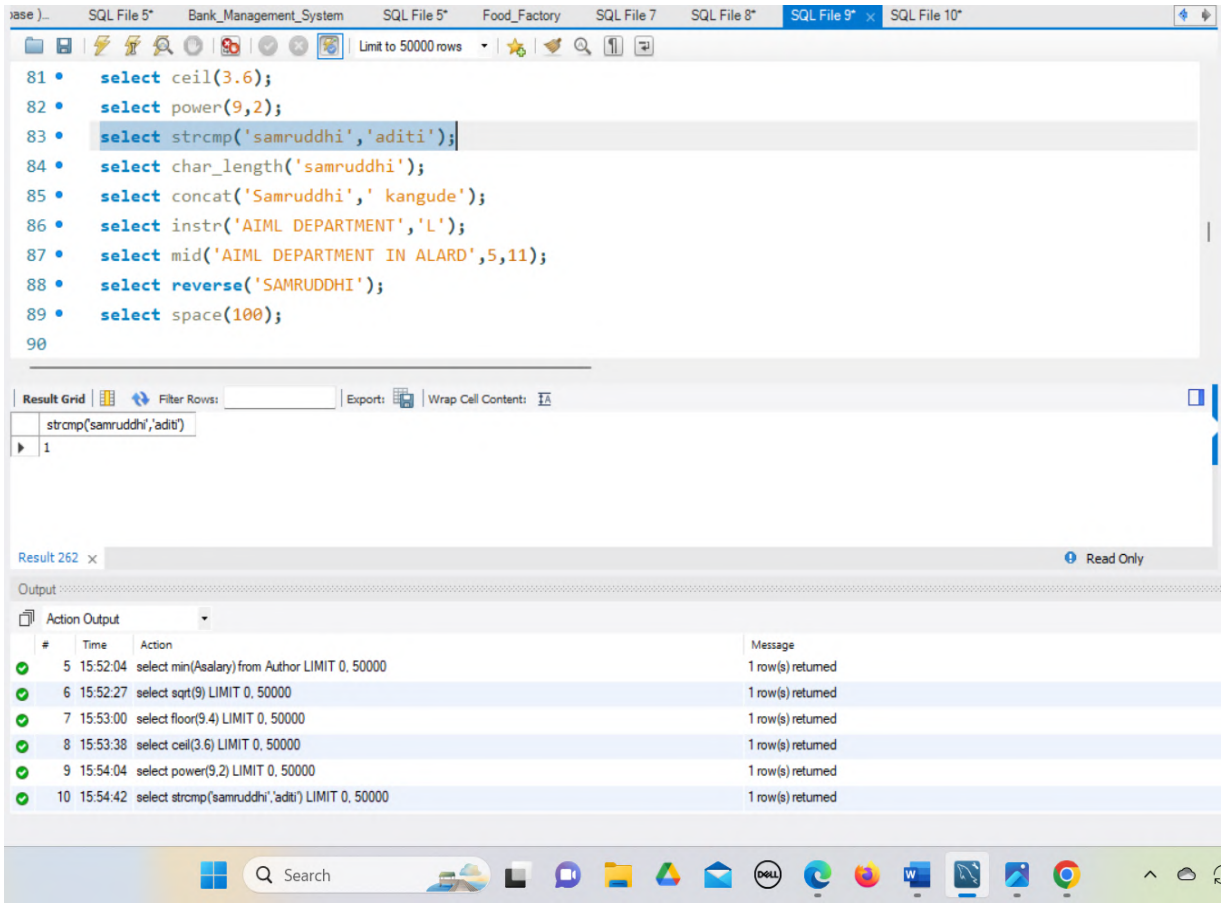
select ceil(3.6);



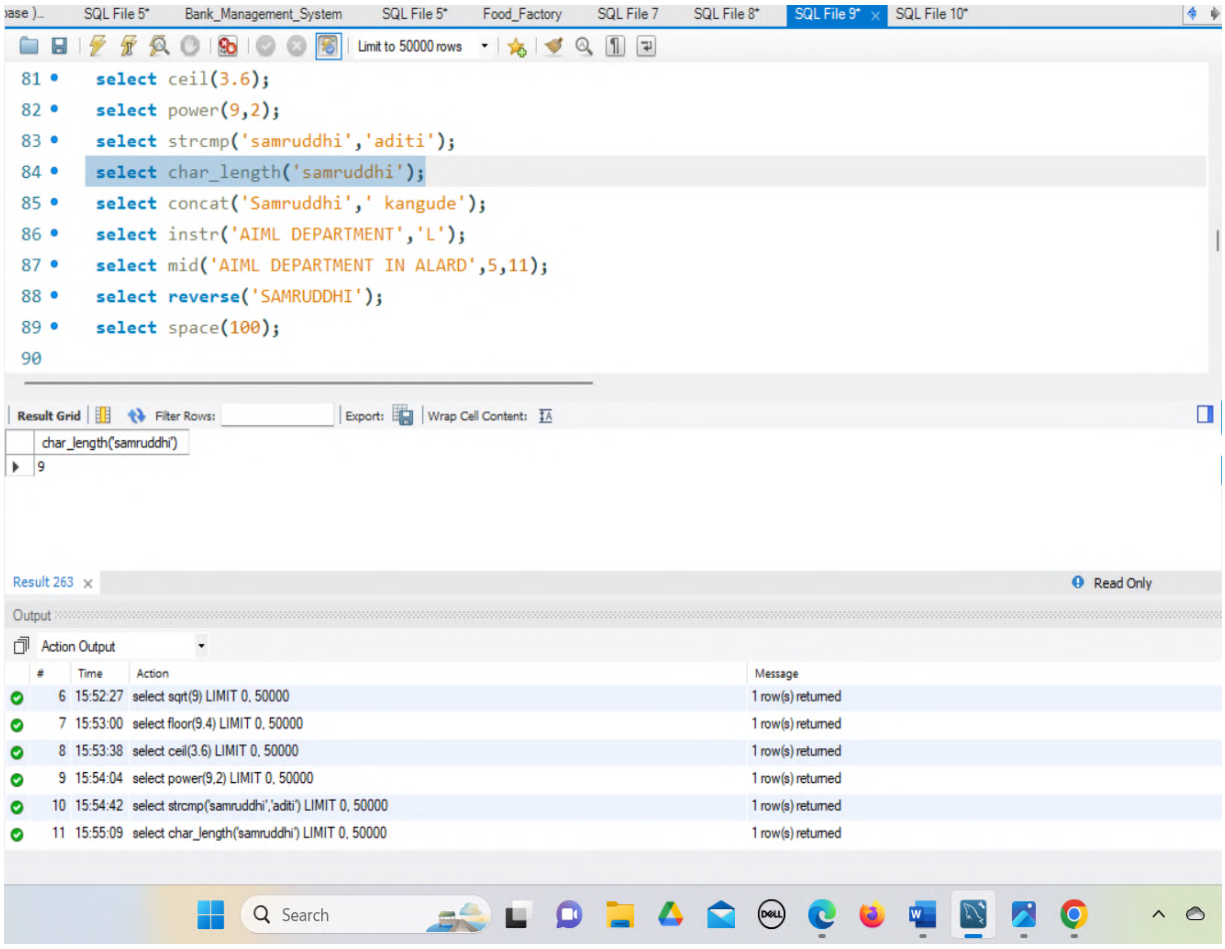
select power(9,2);



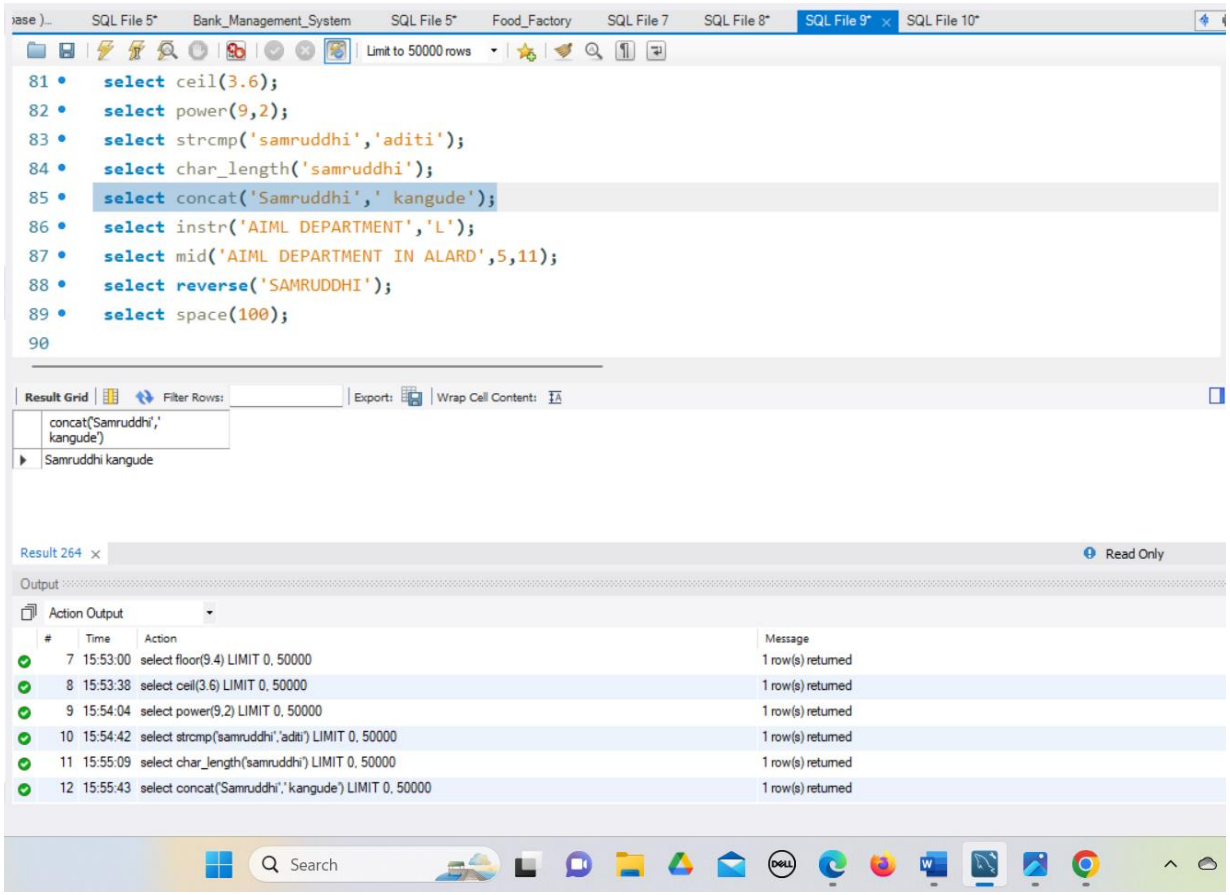
select strcmp('samruddhi','aditi');



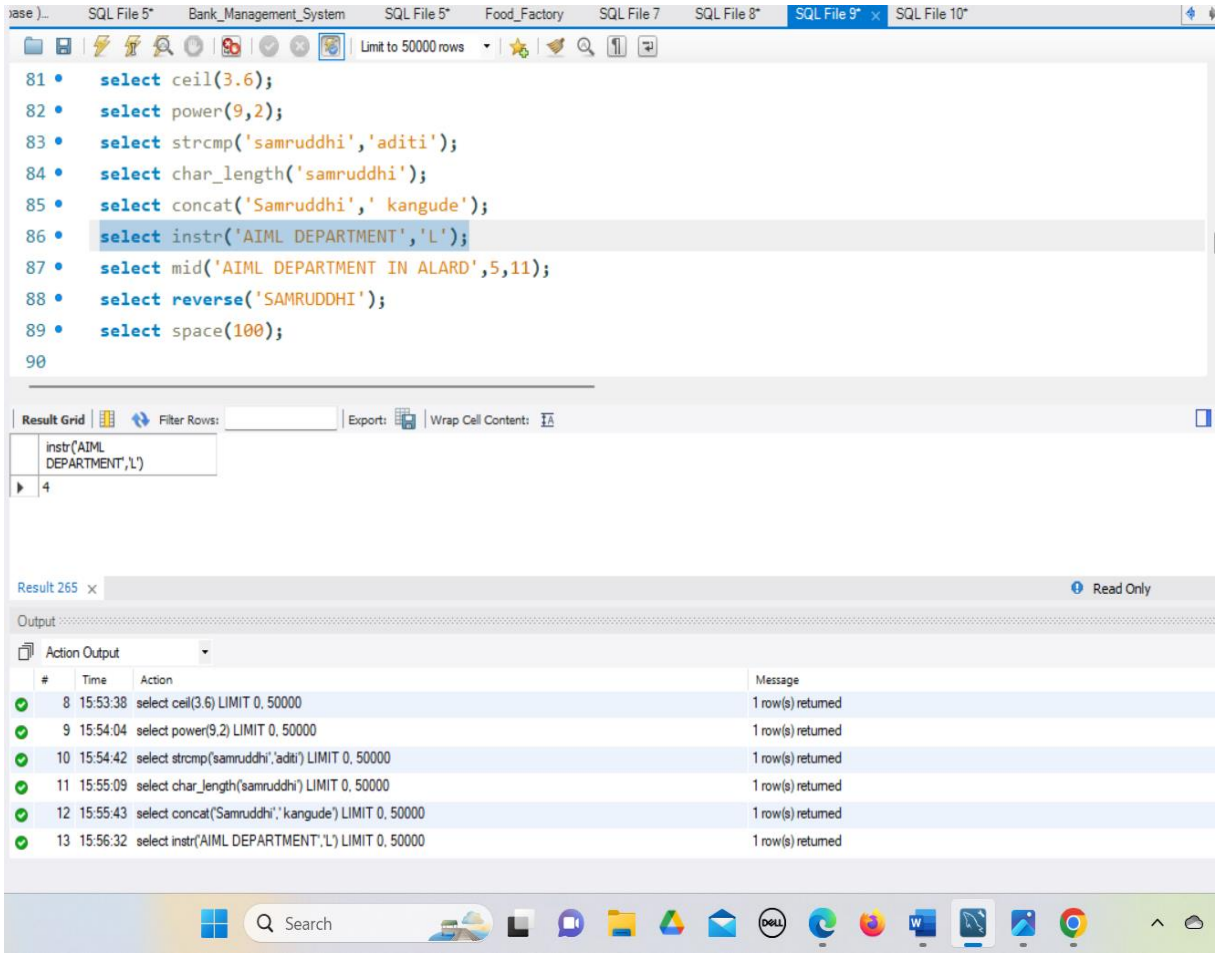
select char_length('samruddhi');



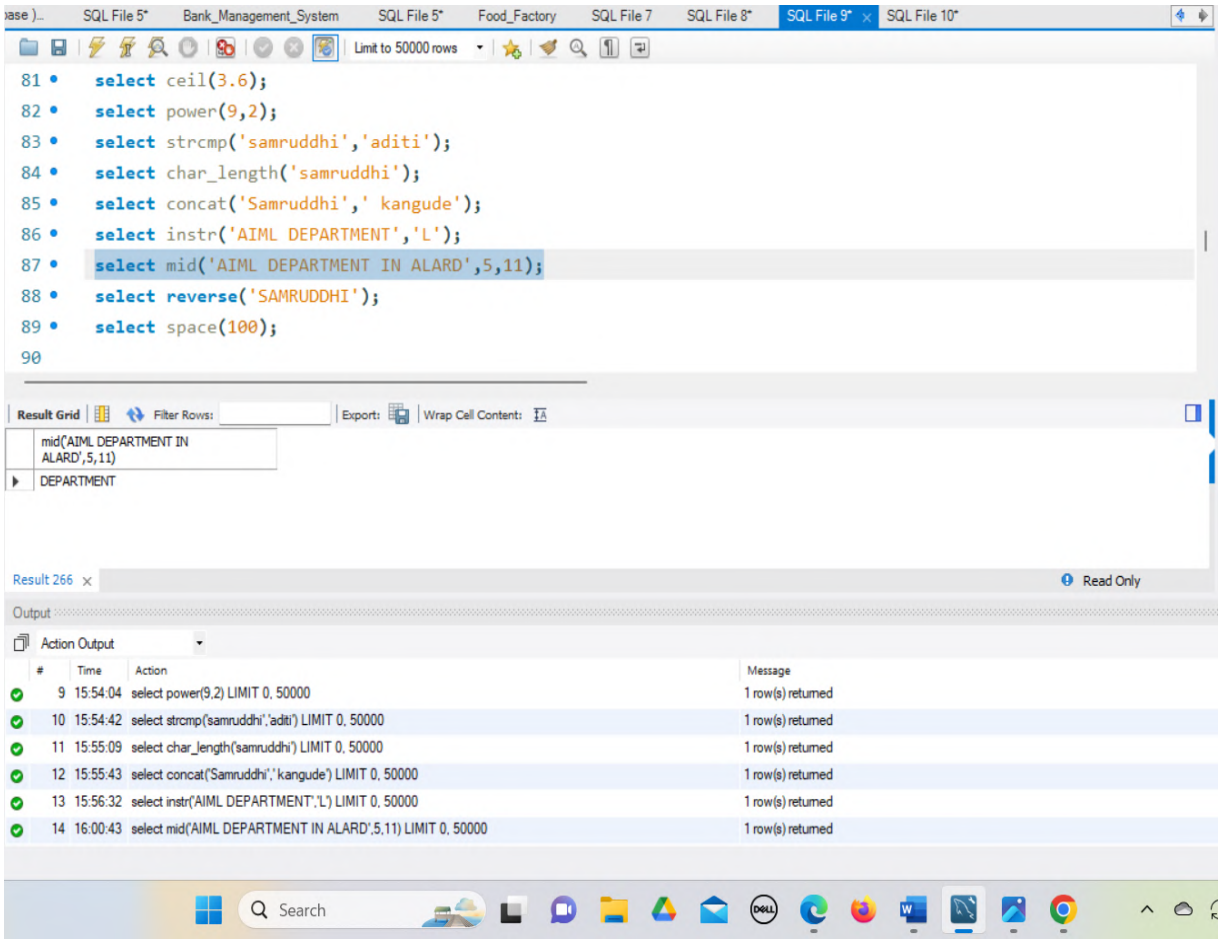
select concat('Samruddhi',' kangude');



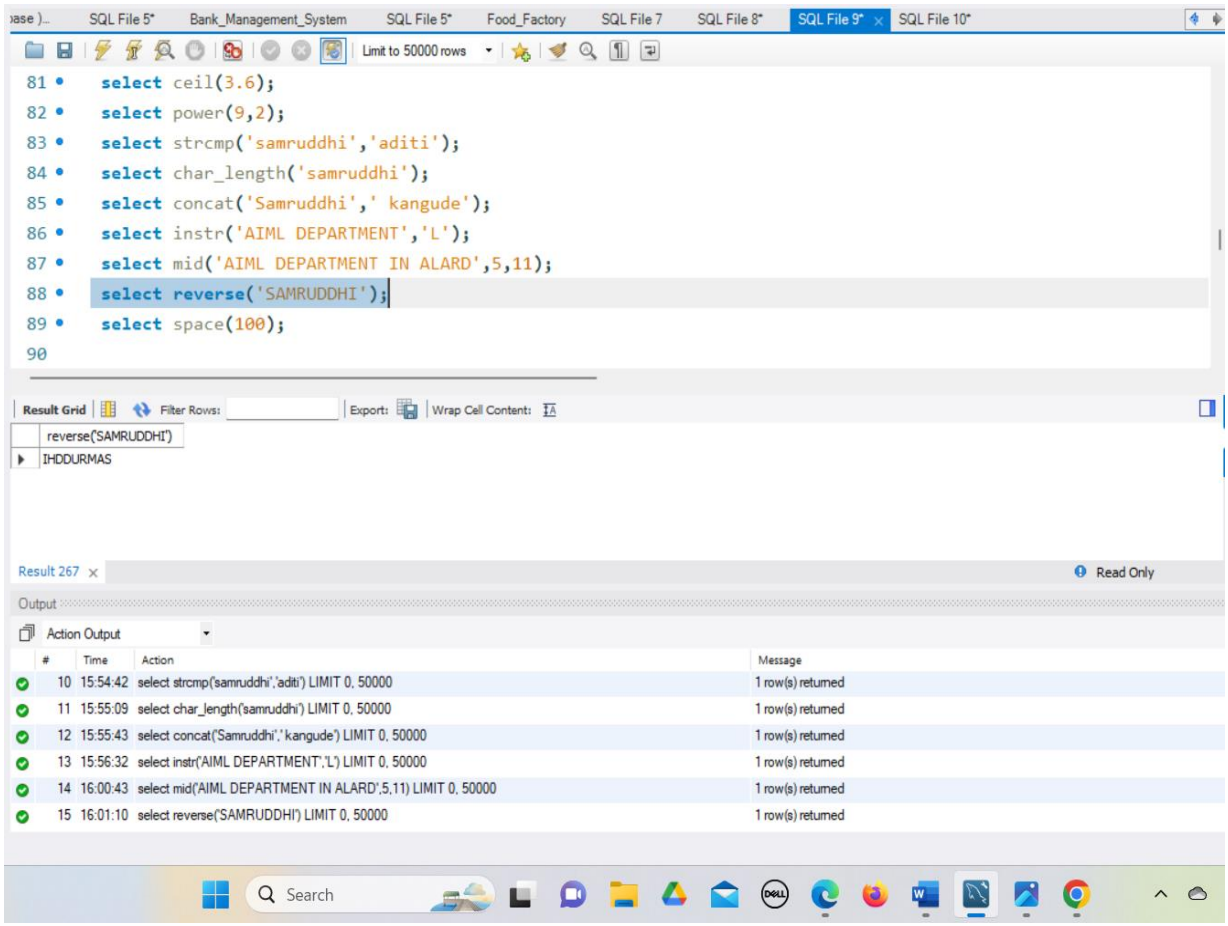
select instr('AIML DEPARTMENT','L');



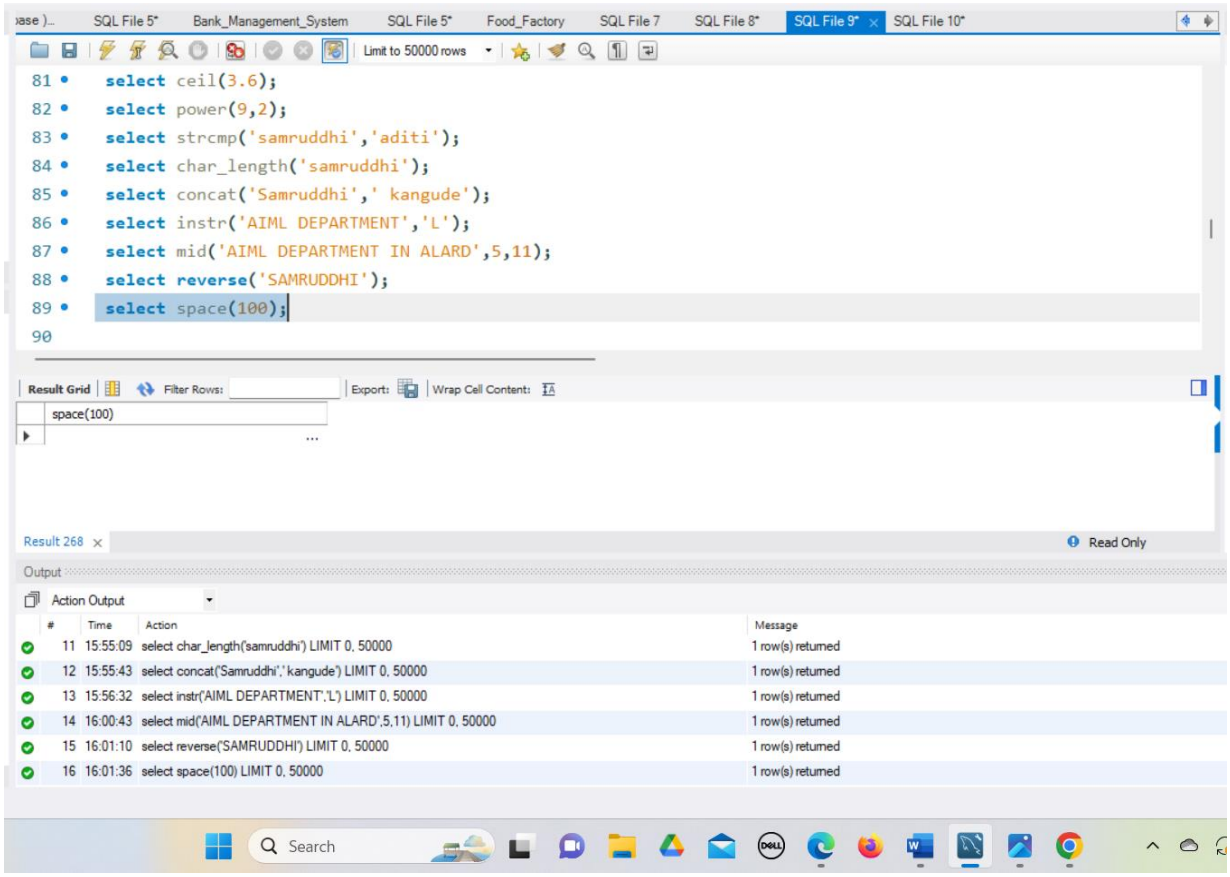
select mid('AIML DEPARTMENT IN ALARD',5,11);



select reverse('SAMRUDDHI');



select space(100);

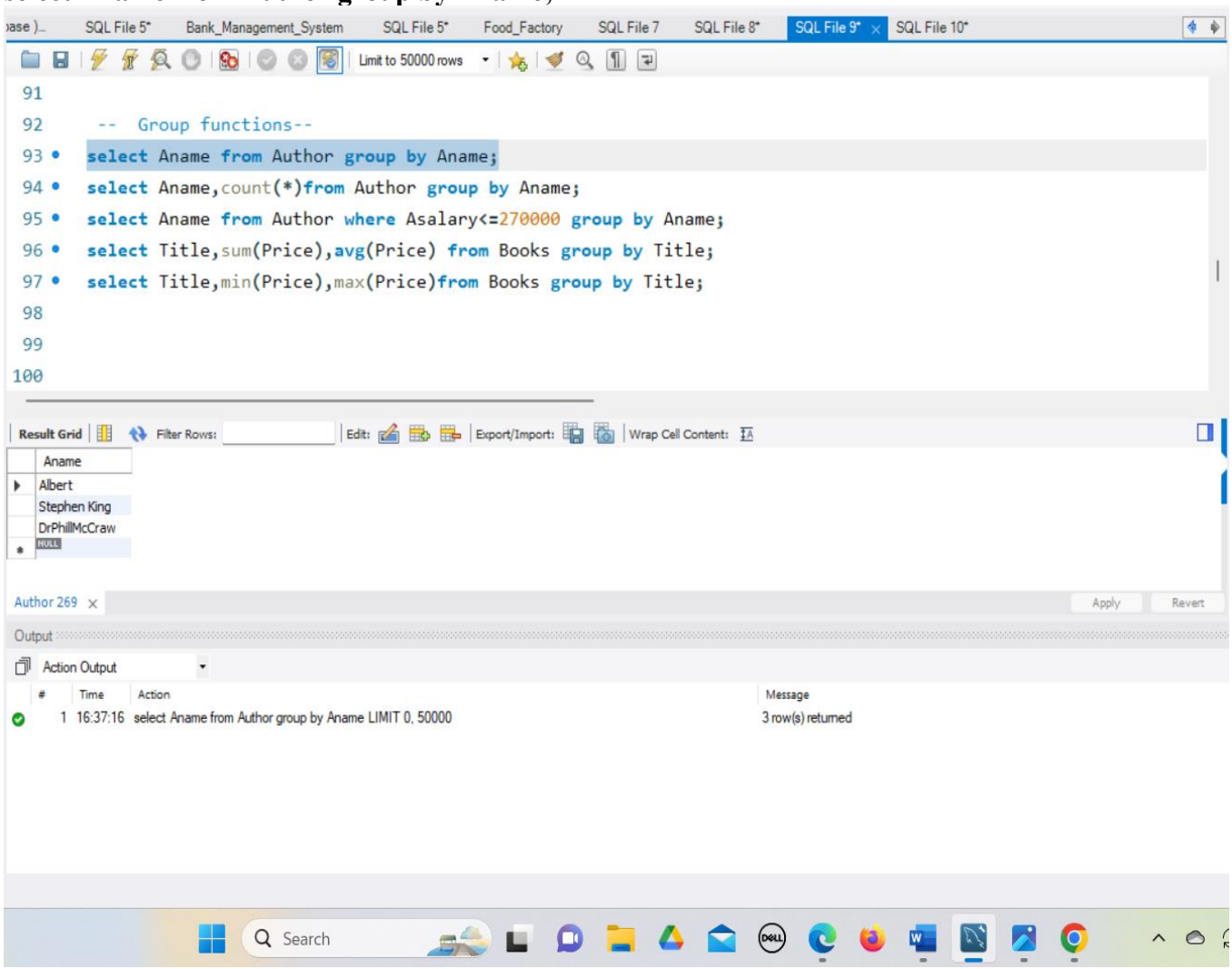


D: Group Functions:

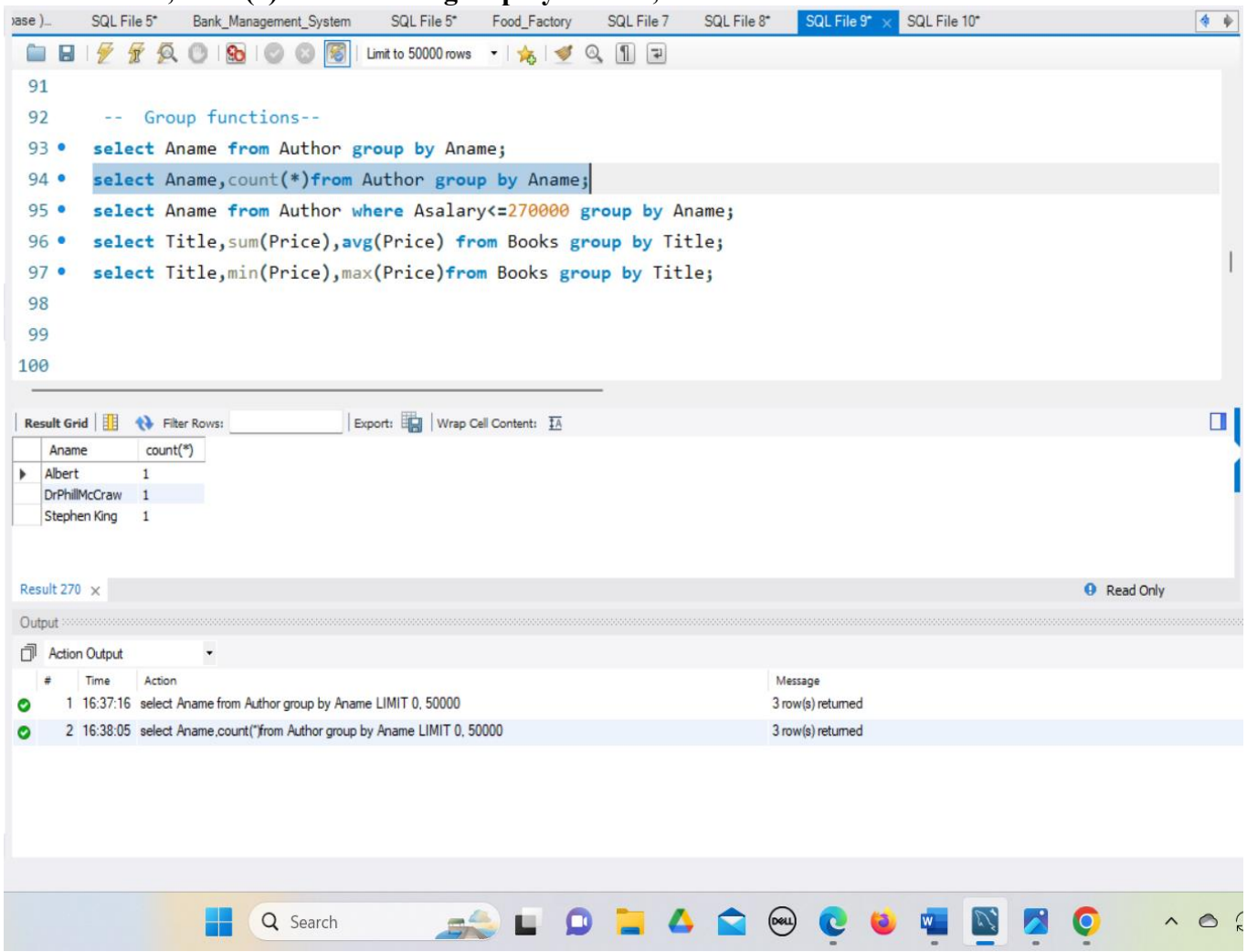
-- Group functions—

select Aname from Author group by Aname;
select Aname,count(*)from Author group by Aname;
select Aname from Author where Asalary<=270000 group by Aname;
select Title,sum(Price),avg(Price) from Books group by Title;
select Title,min(Price),max(Price)from Books group by Title;

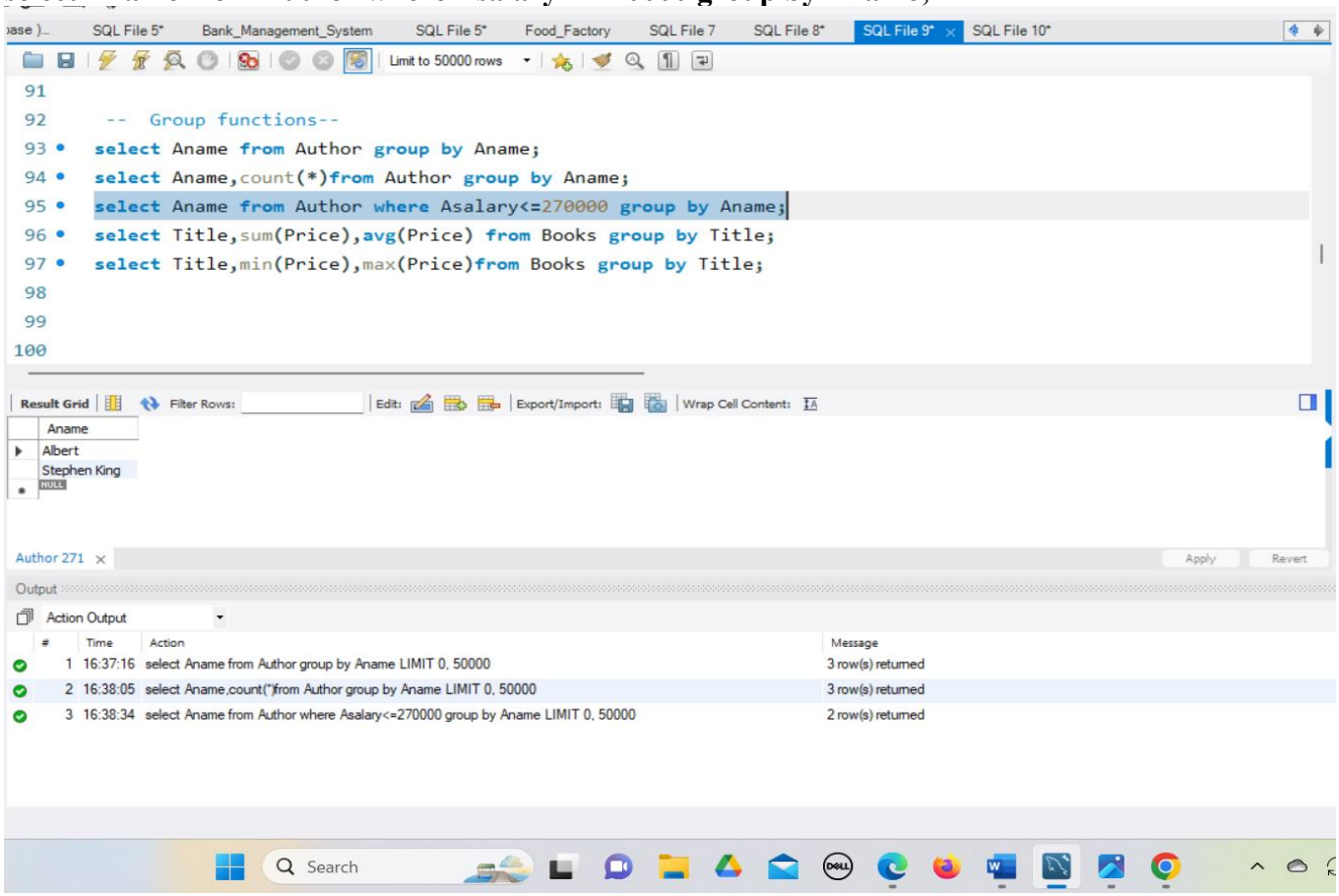
select Aname from Author group by Aname;



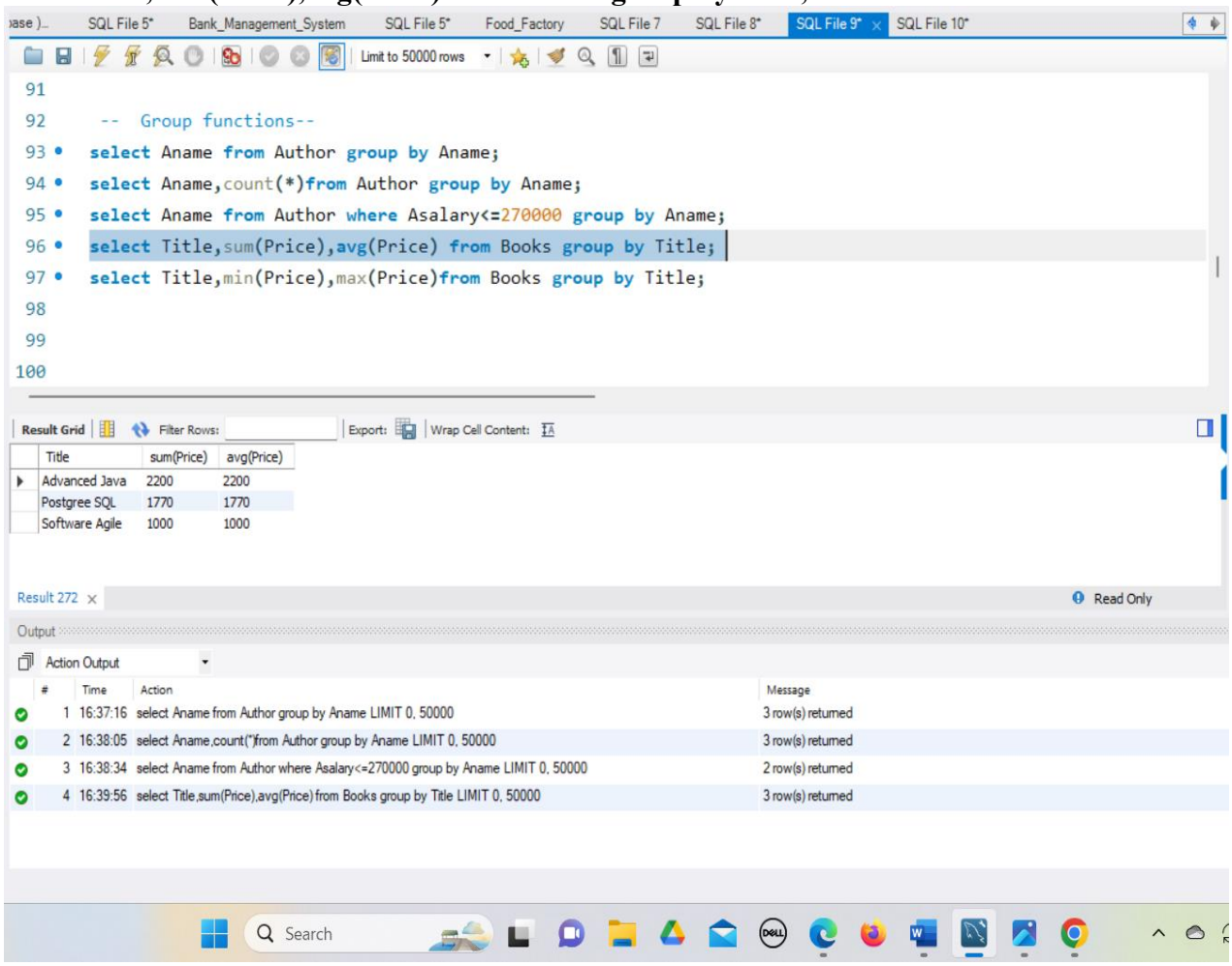
select Aname,count(*)from Author group by Aname;



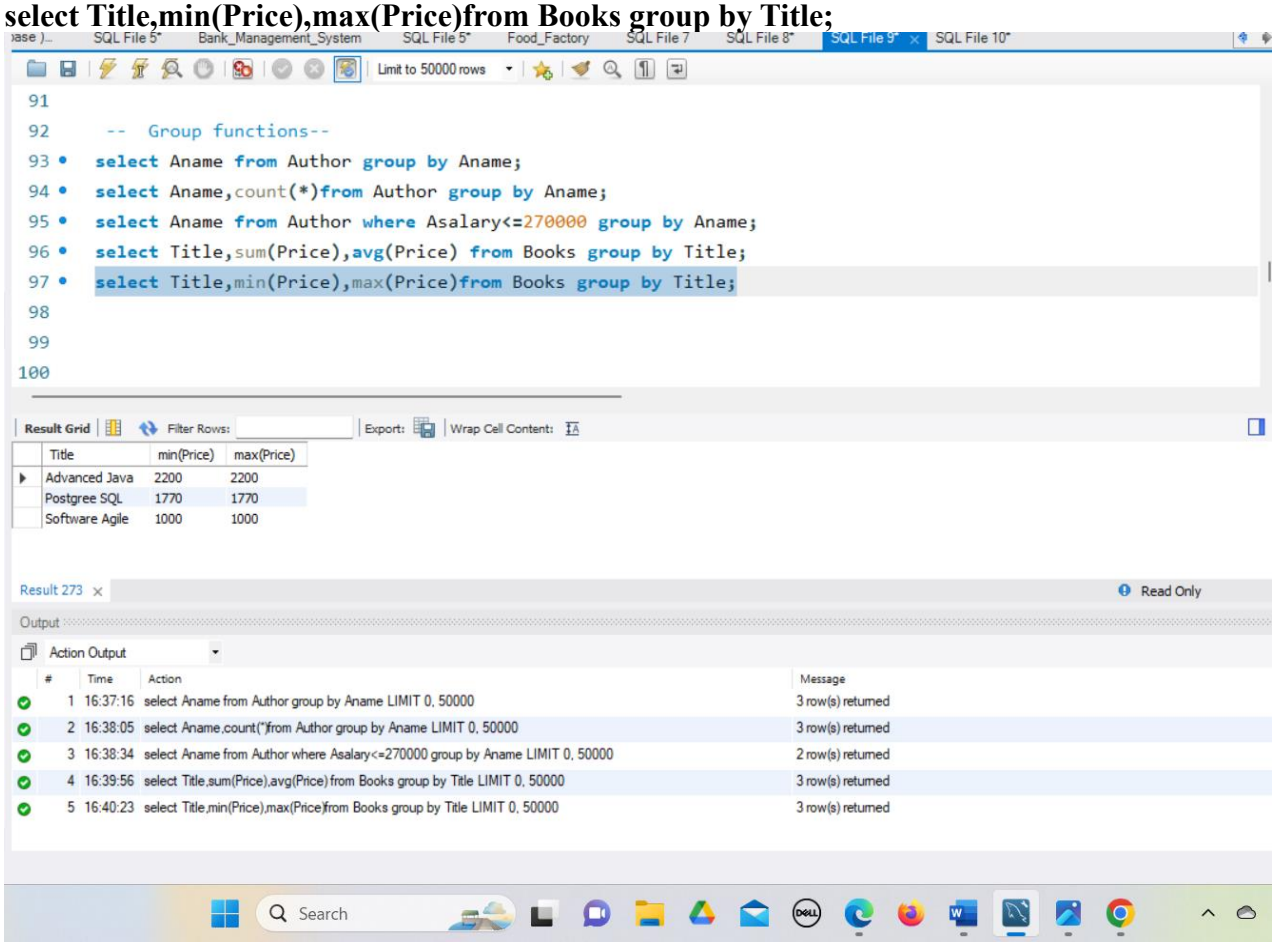
select Aname from Author where Asalary<=270000 group by Aname;



select Title,sum(Price),avg(Price) from Books group by Title;



select Title,min(Price),max(Price)from Books group by Title;

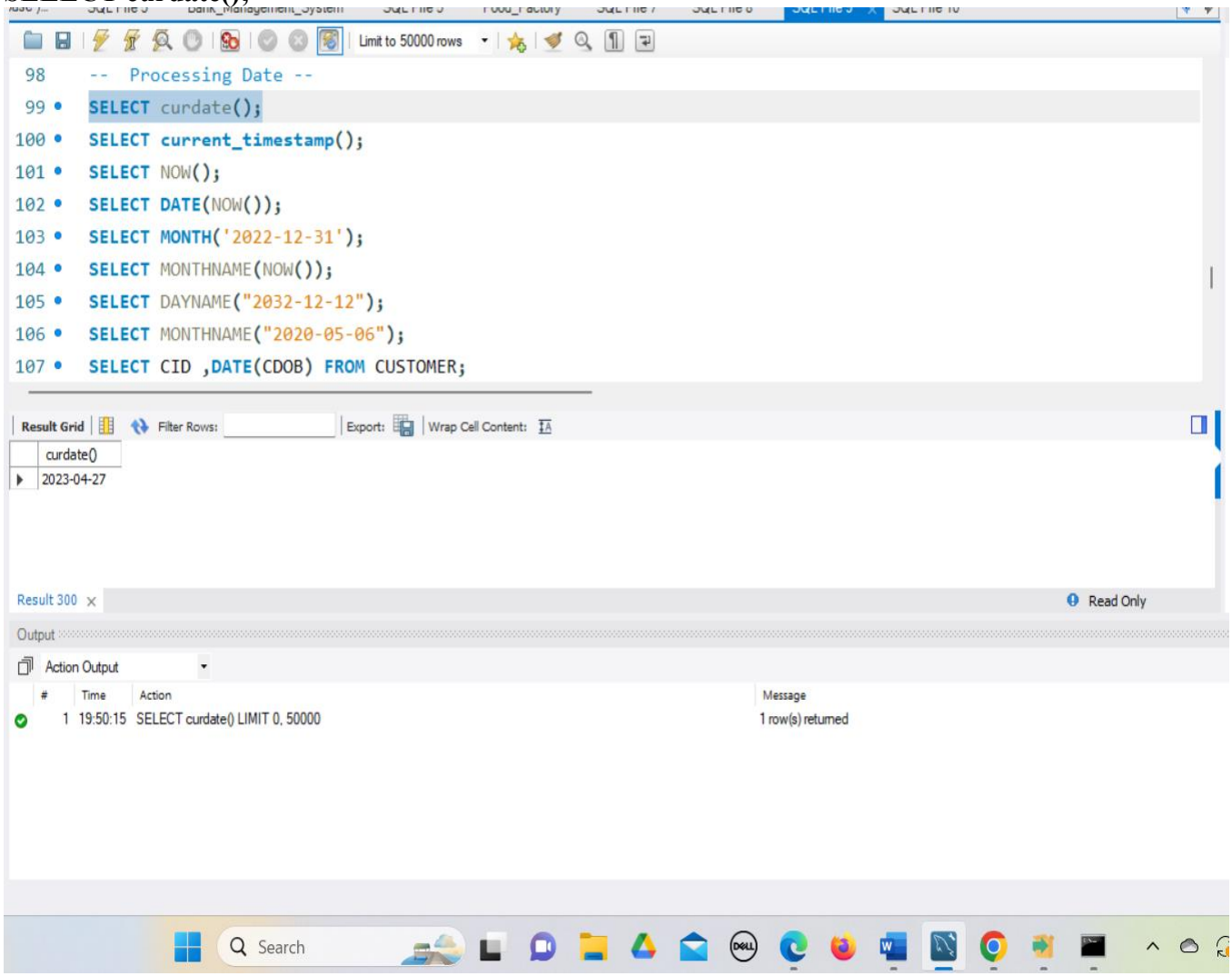


E: Processing Date and Time Functions:

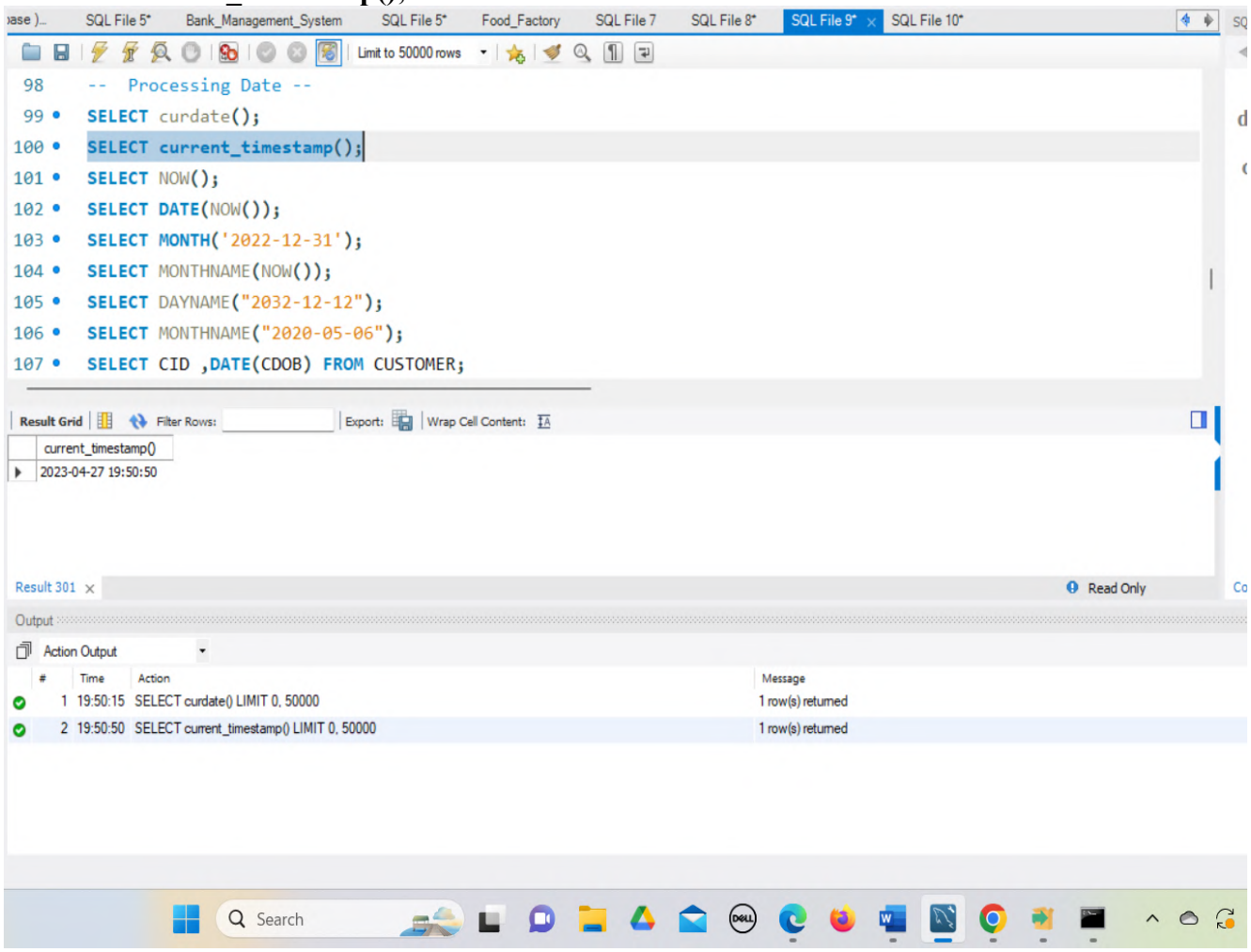
-- Processing Date --

```
SELECT curdate();
SELECT current_timestamp();
SELECT NOW();
SELECT DATE(NOW());
SELECT MONTH('2022-12-31');
SELECT MONTHNAME(NOW());
SELECT DAYNAME("2032-12-12");
SELECT MONTHNAME("2020-05-06");
SELECT CID ,DATE(CDOB) FROM CUSTOMER;
```

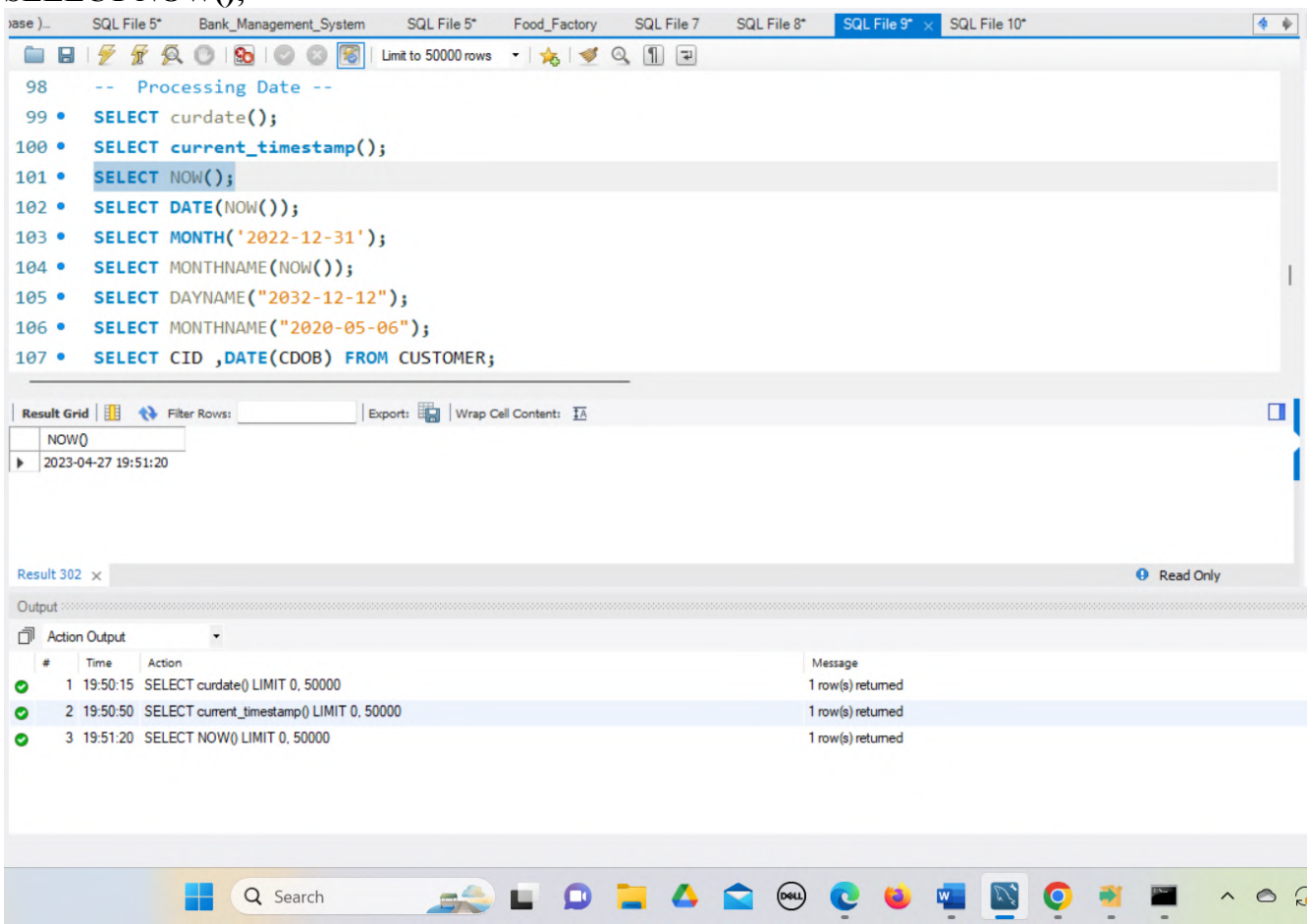
SELECT curdate();



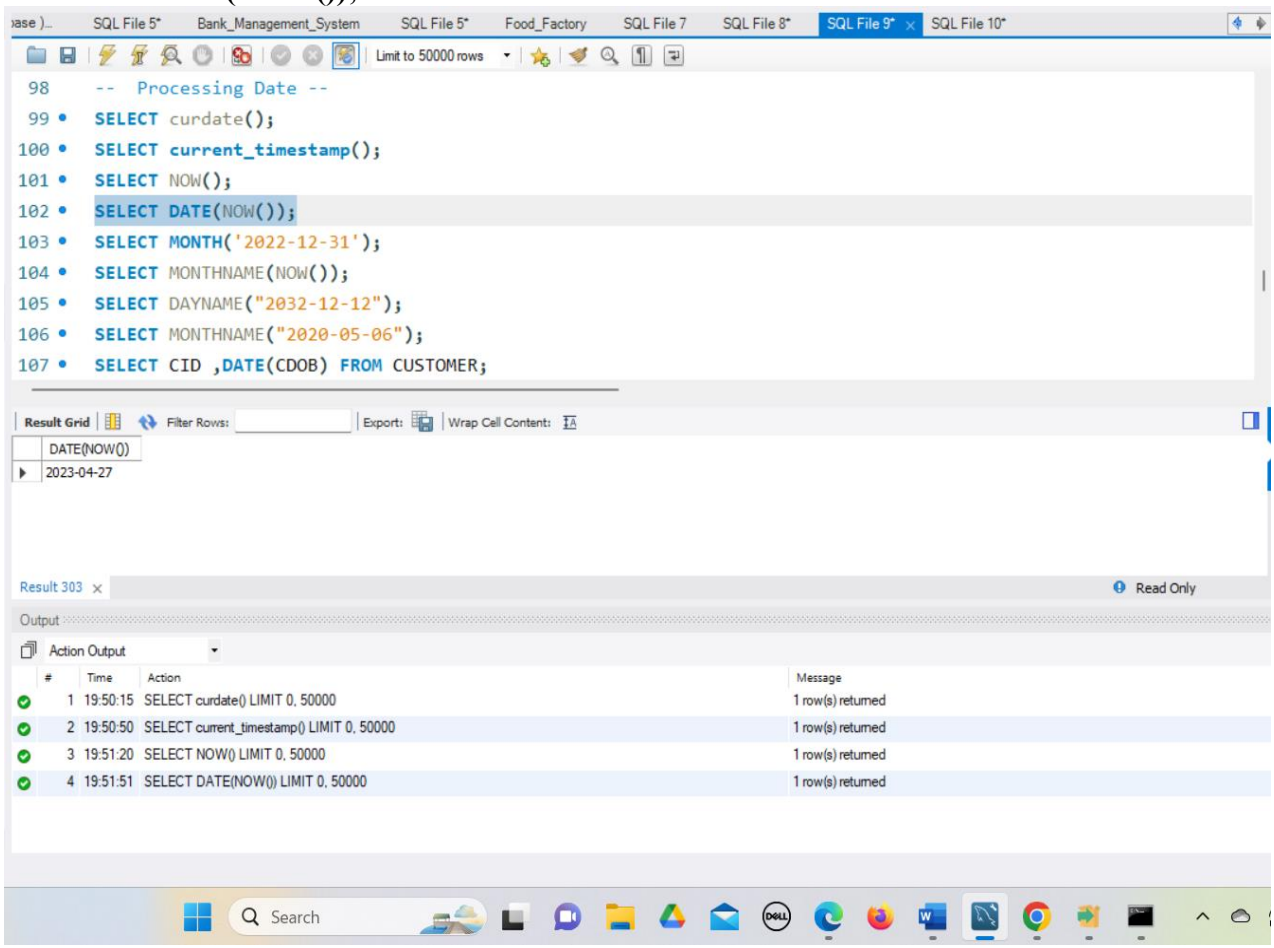
SELECT current_timestamp();



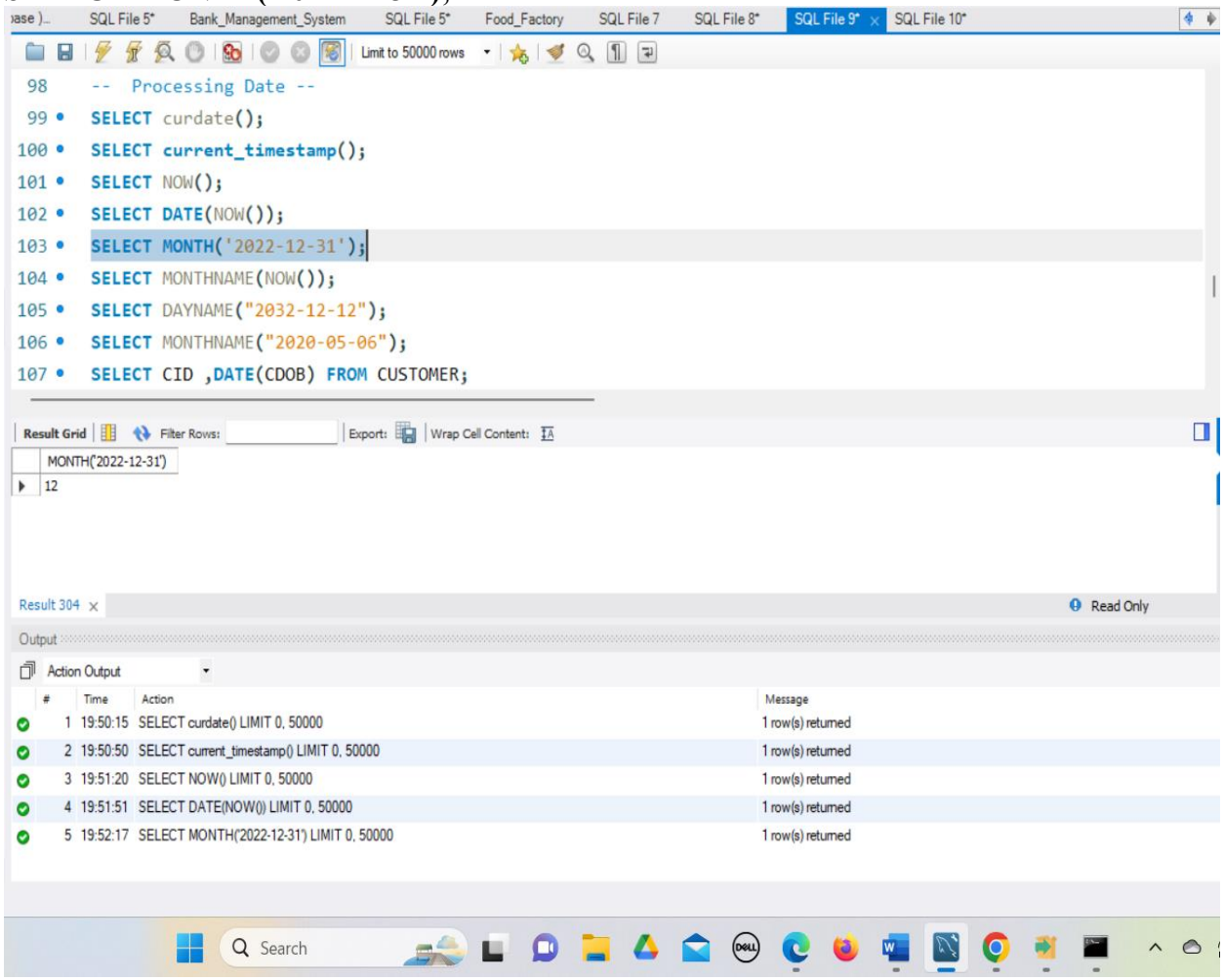
SELECT NOW();



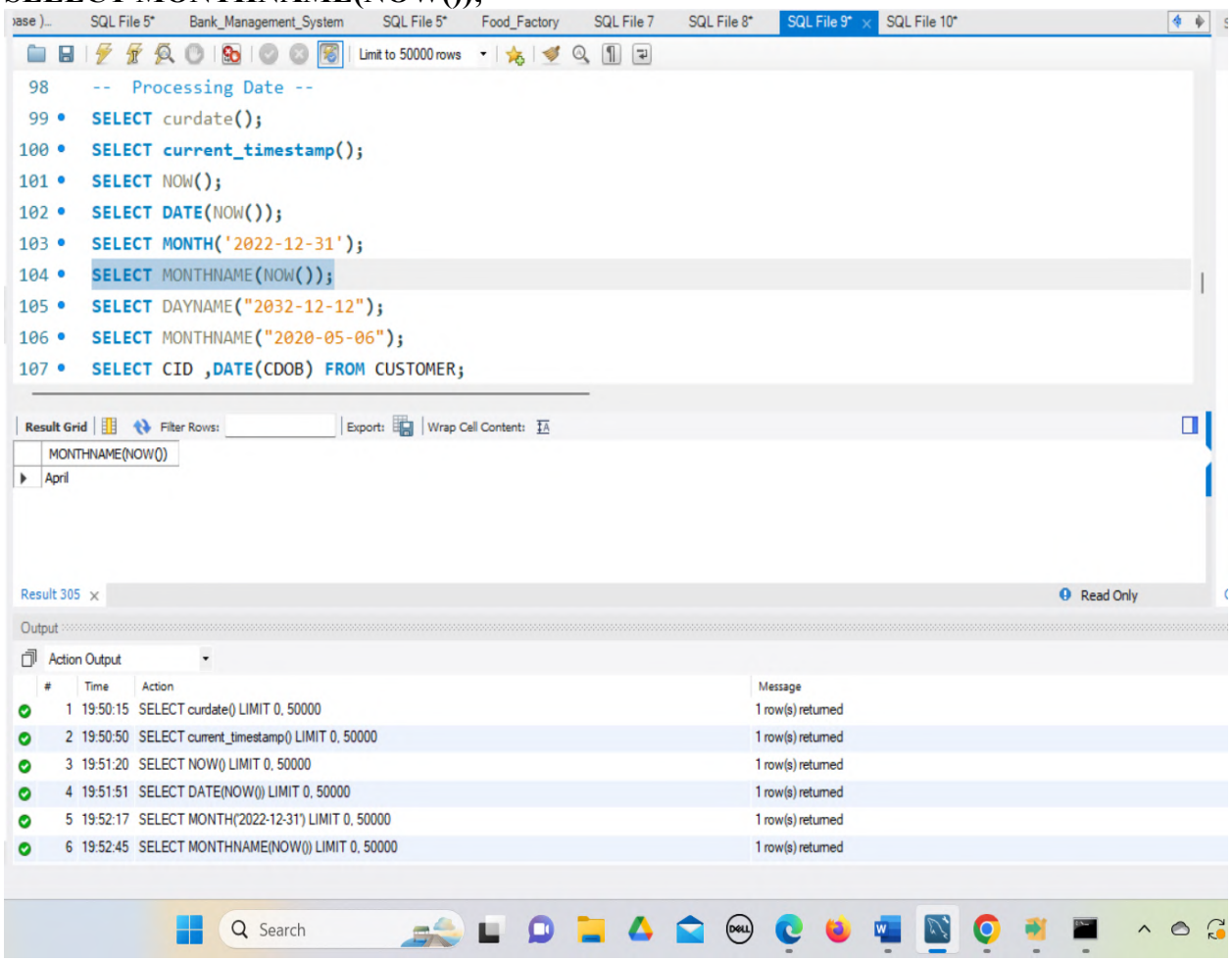
SELECT DATE(NOW());



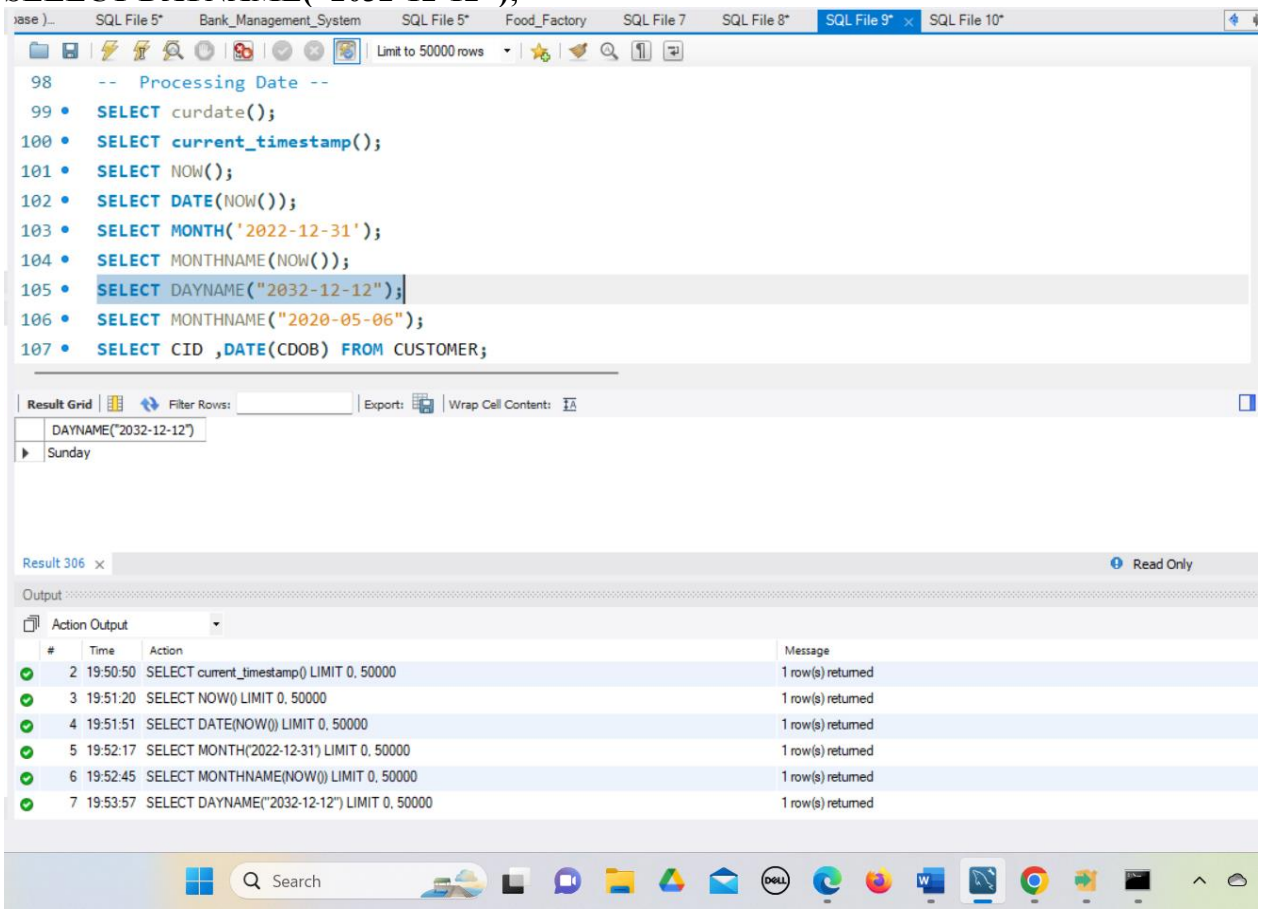
SELECT MONTH('2022-12-31');



SELECT MONTHNAME(NOW());



SELECT DAYNAME("2032-12-12");



SELECT MONTHNAME("2020-05-06");

```
98   -- Processing Date --
99   SELECT curdate();
100  SELECT current_timestamp();
101  SELECT NOW();
102  SELECT DATE(NOW());
103  SELECT MONTH('2022-12-31');
104  SELECT MONTHNAME(NOW());
105  SELECT DAYNAME("2032-12-12");
106  SELECT MONTHNAME("2020-05-06");
107  SELECT CID ,DATE(CDOB) FROM CUSTOMER;
```

MONTHNAME("2020-05-06")
May

#	Time	Action	Message
3	19:51:20	SELECT NOW() LIMIT 0, 50000	1 row(s) returned
4	19:51:51	SELECT DATE(NOW()) LIMIT 0, 50000	1 row(s) returned
5	19:52:17	SELECT MONTH('2022-12-31') LIMIT 0, 50000	1 row(s) returned
6	19:52:45	SELECT MONTHNAME(NOW()) LIMIT 0, 50000	1 row(s) returned
7	19:53:57	SELECT DAYNAME("2032-12-12") LIMIT 0, 50000	1 row(s) returned
8	19:54:24	SELECT MONTHNAME("2020-05-06") LIMIT 0, 50000	1 row(s) returned

SELECT CID ,DATE(CDOB) FROM CUSTOMER;

```
99   SELECT curdate();
100  SELECT current_timestamp();
101  SELECT NOW();
102  SELECT DATE(NOW());
103  SELECT MONTH('2022-12-31');
104  SELECT MONTHNAME(NOW());
105  SELECT DAYNAME("2032-12-12");
106  SELECT MONTHNAME("2020-05-06");
107  SELECT CID ,DATE(CDOB) FROM CUSTOMER;
108
```

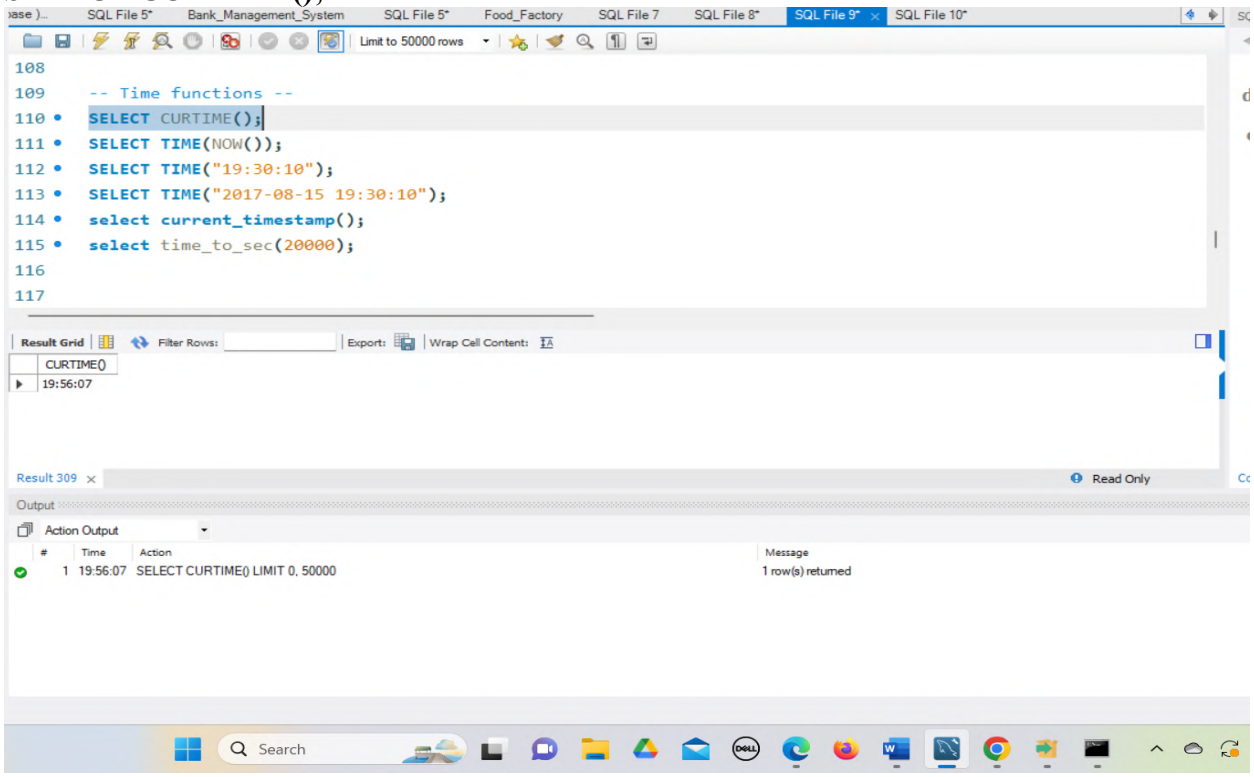
CID	DATE(CDOB)
200	2015-05-04
100	2017-06-06
300	2019-09-12

#	Time	Action	Message
4	19:51:51	SELECT DATE(NOW()) LIMIT 0, 50000	1 row(s) returned
5	19:52:17	SELECT MONTH('2022-12-31') LIMIT 0, 50000	1 row(s) returned
6	19:52:45	SELECT MONTHNAME(NOW()) LIMIT 0, 50000	1 row(s) returned
7	19:53:57	SELECT DAYNAME("2032-12-12") LIMIT 0, 50000	1 row(s) returned
8	19:54:24	SELECT MONTHNAME("2020-05-06") LIMIT 0, 50000	1 row(s) returned
9	19:55:09	SELECT CID ,DATE(CDOB) FROM CUSTOMER LIMIT 0, 50000	3 row(s) returned

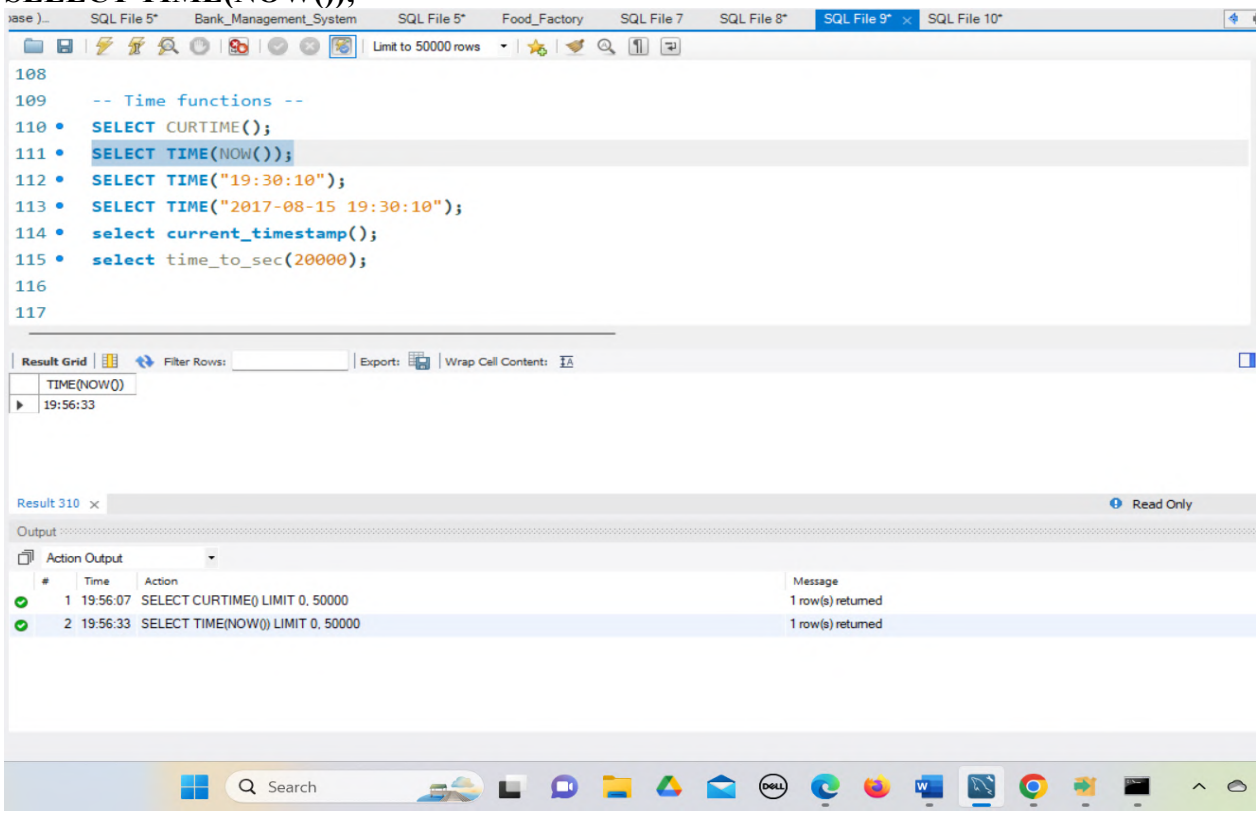
-- Time functions --

```
SELECT CURTIME();
SELECT TIME(NOW());
SELECT TIME("19:30:10");
SELECT TIME("2017-08-15 19:30:10");
select current_timestamp();
select time_to_sec(20000);
```

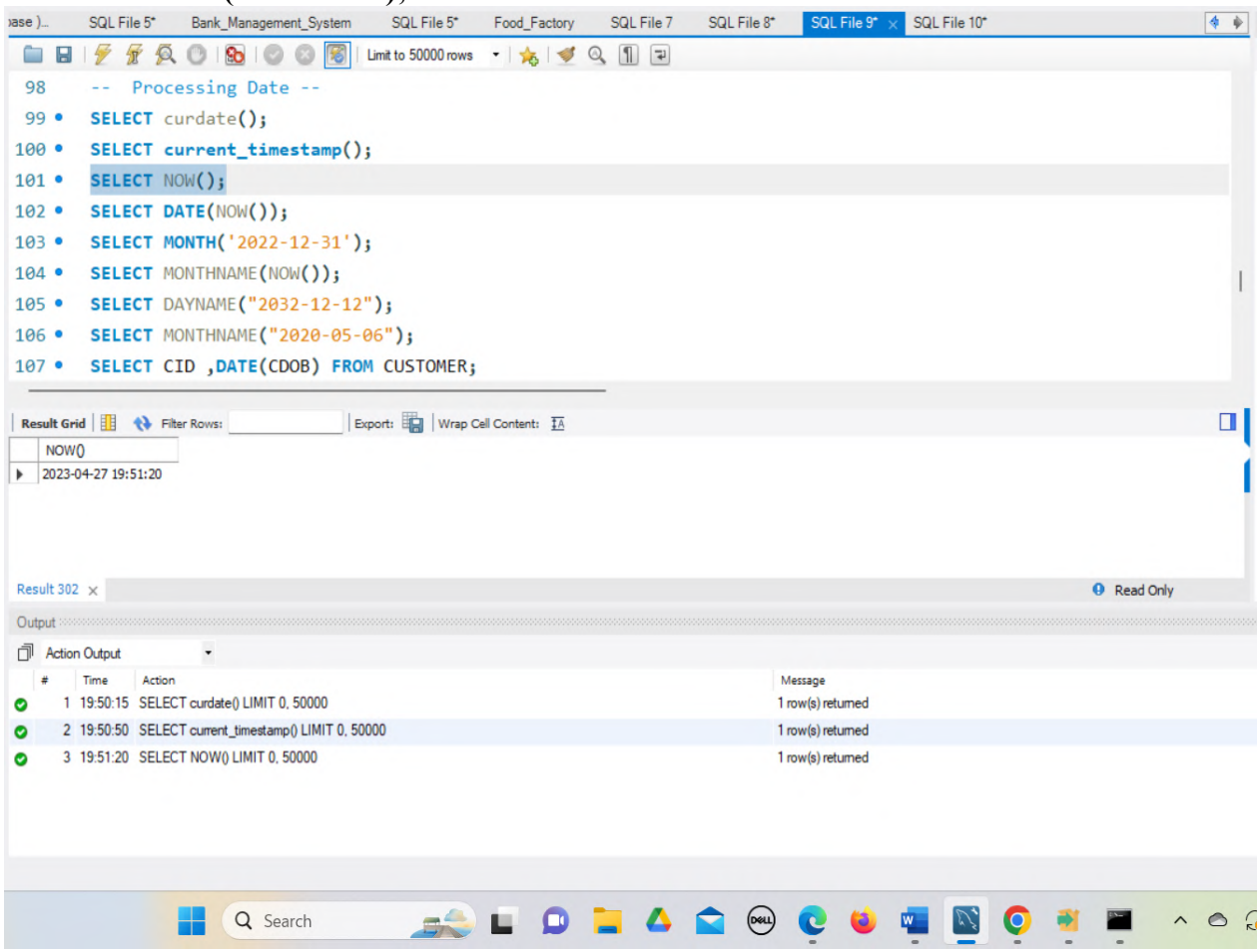
SELECT CURTIME();



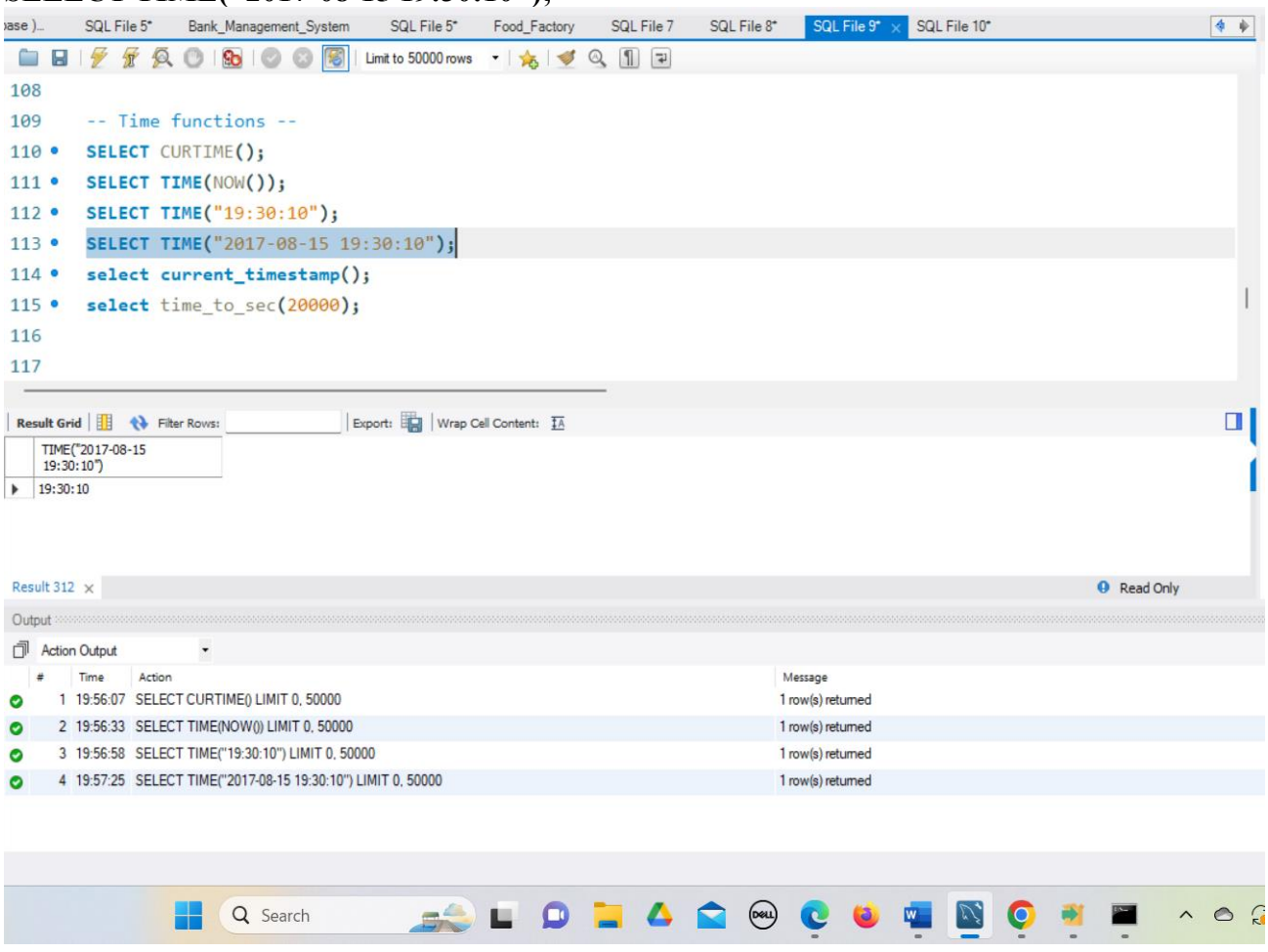
SELECT TIME(NOW());



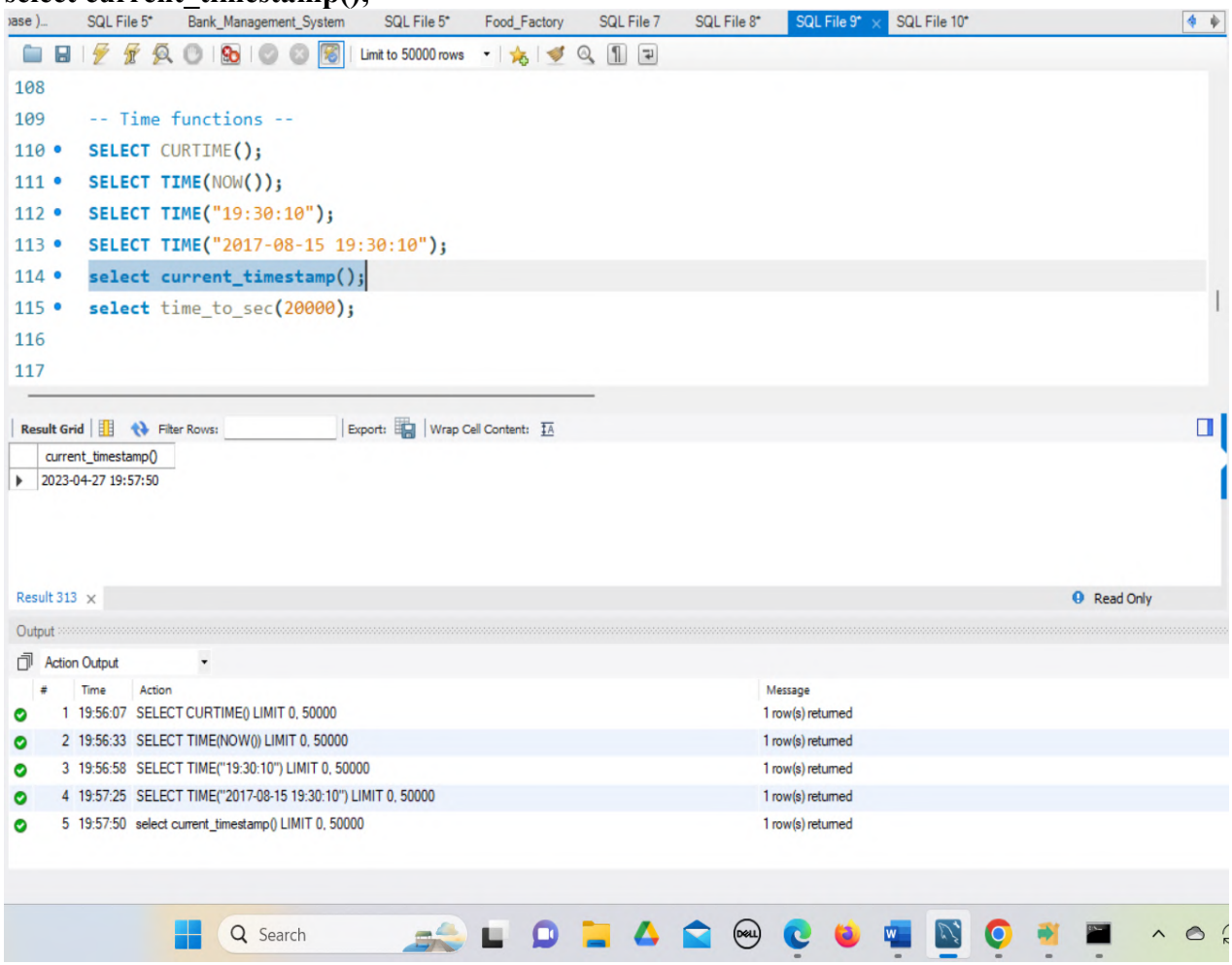
SELECT TIME("19:30:10");



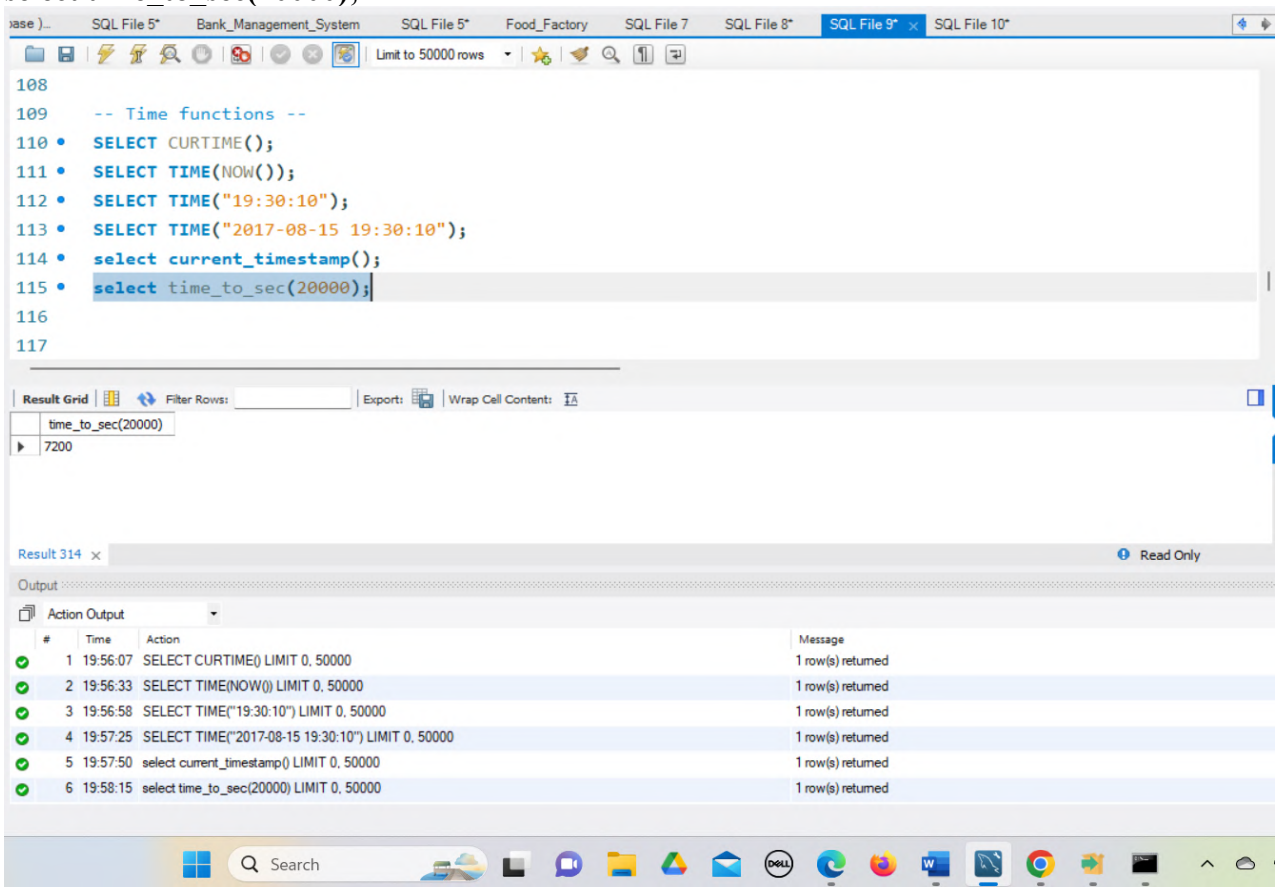
SELECT TIME("2017-08-15 19:30:10");



select current_timestamp();



select time_to_sec(20000);

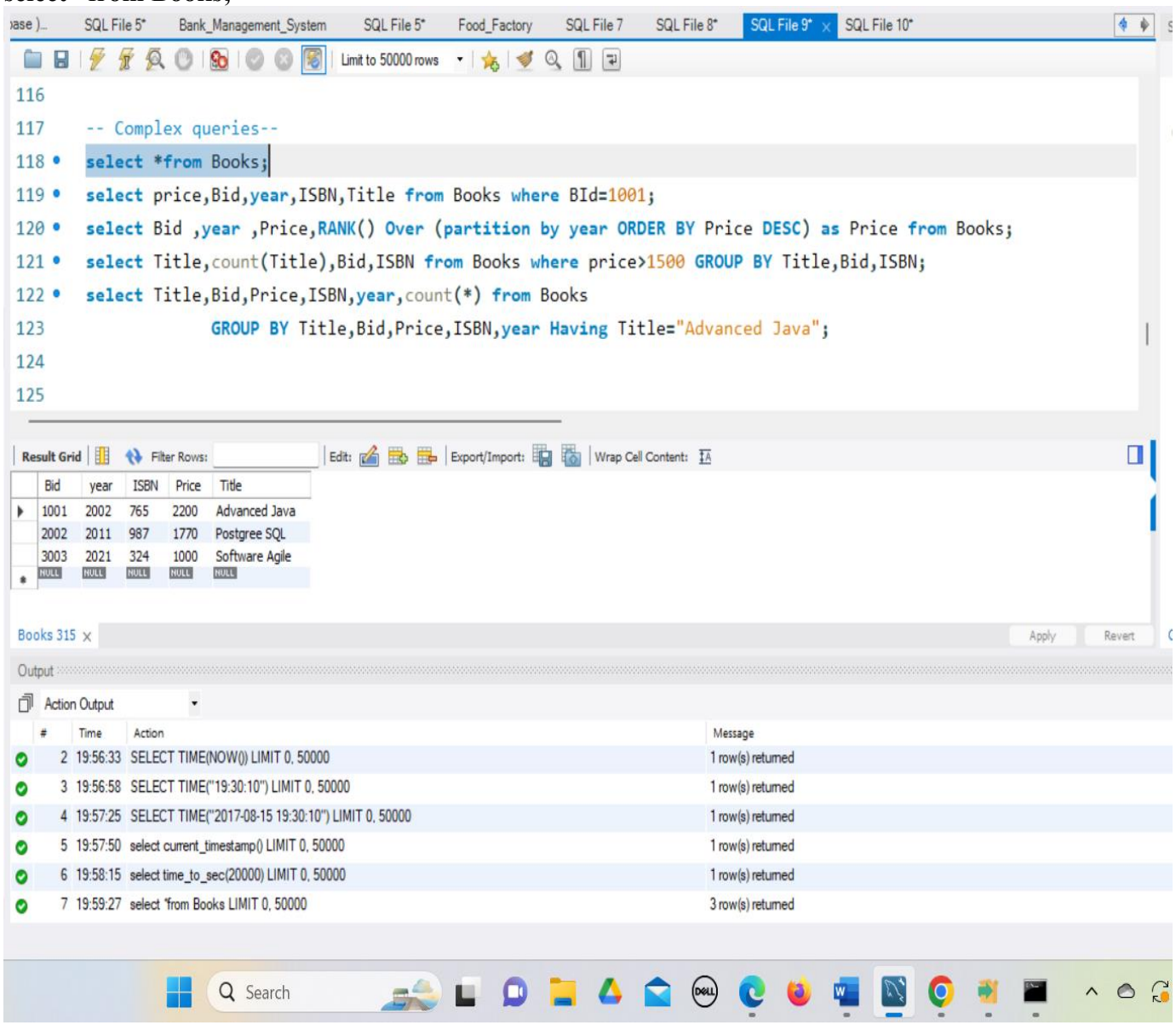


F: Complex Queries and Set Operators:

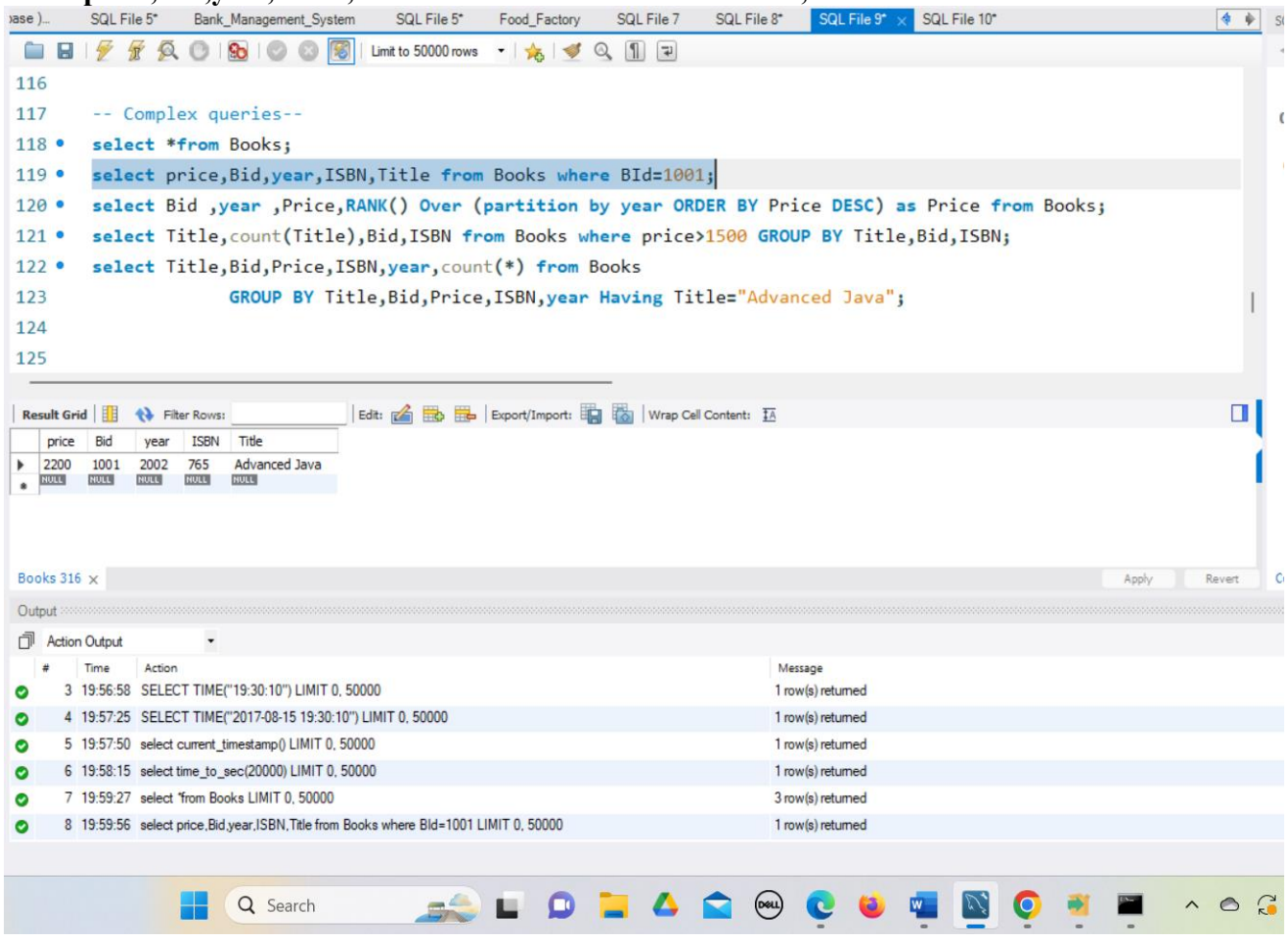
-- Complex queries--

select *from Books;
select price,Bid,year,ISBN,Title from Books where BId=1001;
select Bid ,year ,Price,RANK() Over (partition by year ORDER BY Price DESC) as Price from Books;
select Title,count(Title),Bid,ISBN from Books where price>1500 GROUP BY Title,Bid,ISBN;
select Title,Bid,Price,ISBN,year,count(*) from Books GROUP BY Title,Bid,Price,ISBN,year Having Title="Advanced Java";

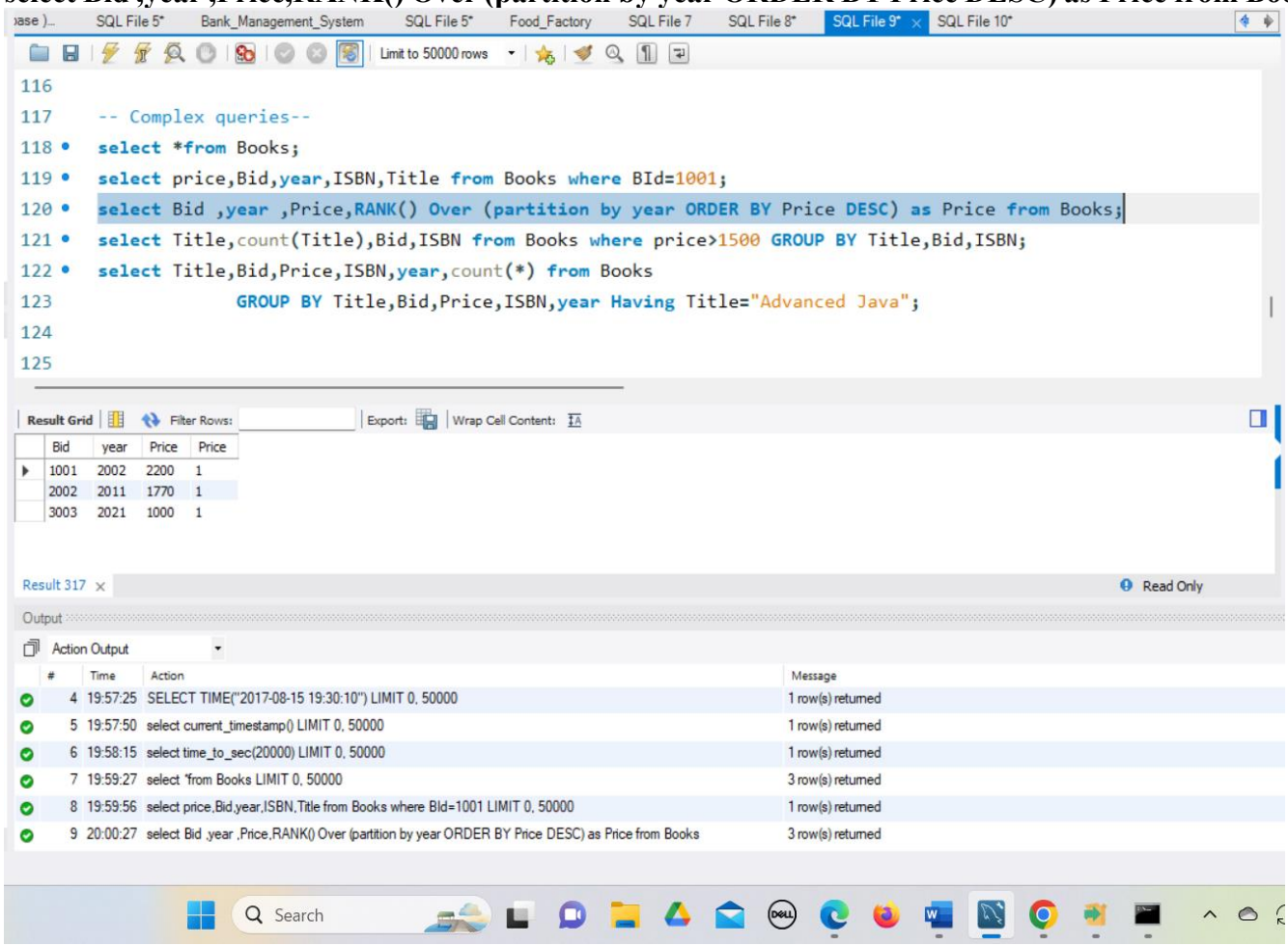
select *from Books;



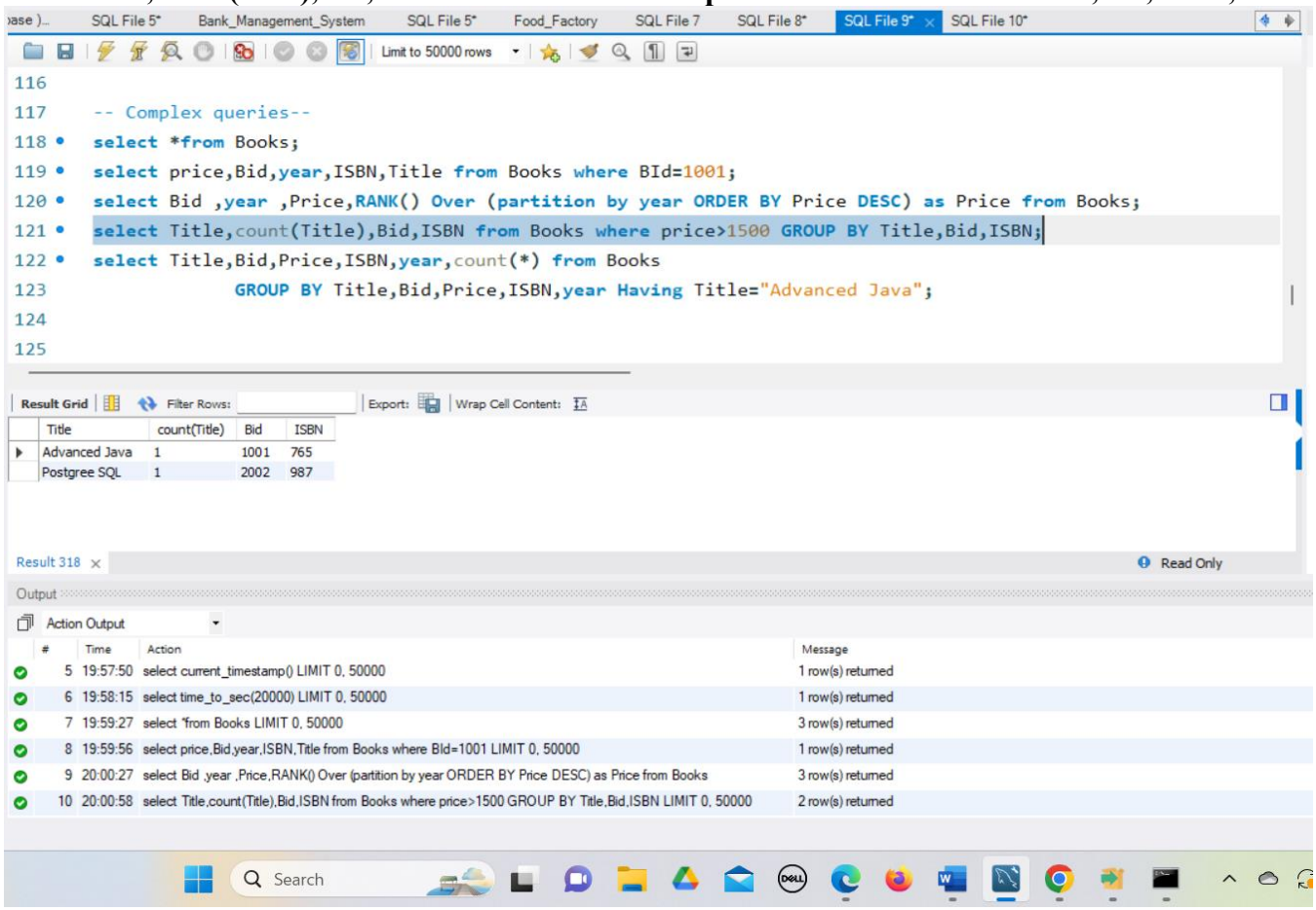
select price,Bid,year,ISBN,Title from Books where BId=1001;



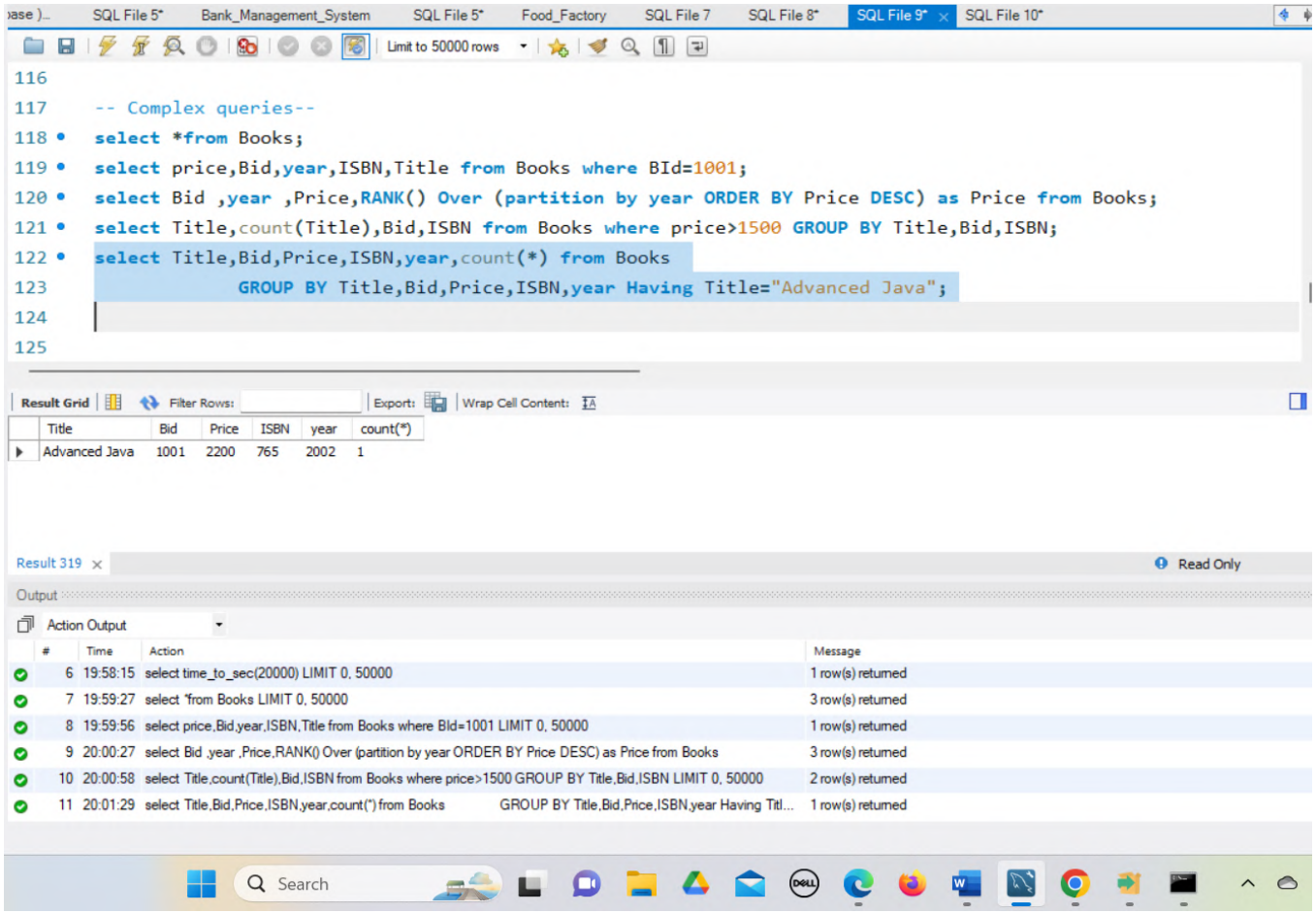
select Bid ,year ,Price,RANK() Over (partition by year ORDER BY Price DESC) as Price from Books;



select Title,count(Title),Bid,ISBN from Books where price>1500 GROUP BY Title,Bid,ISBN;



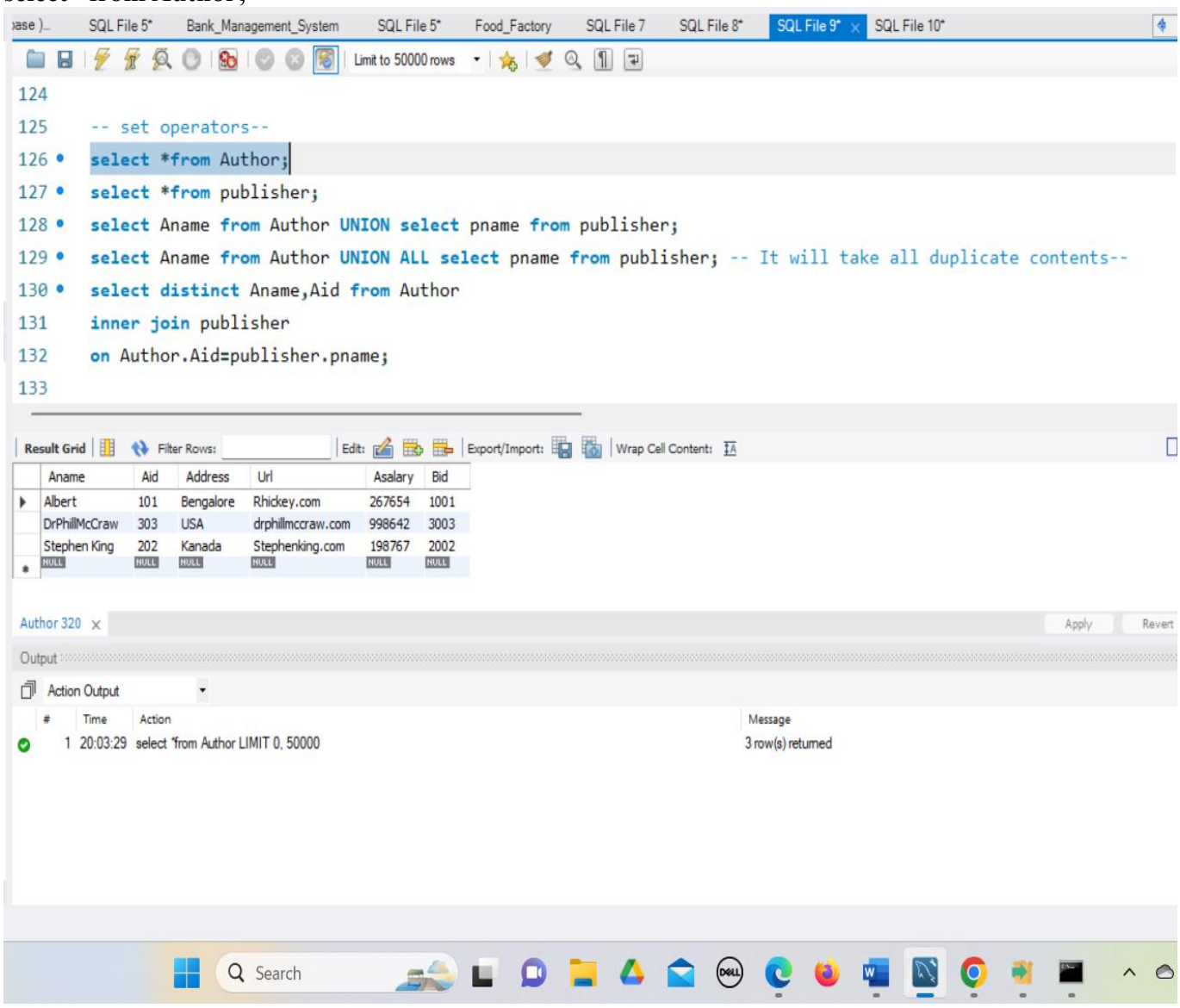
select Title,Bid,Price,ISBN,year,count(*) from Books GROUP BY Title,Bid,Price,ISBN,year Having Title="Advanced Java";



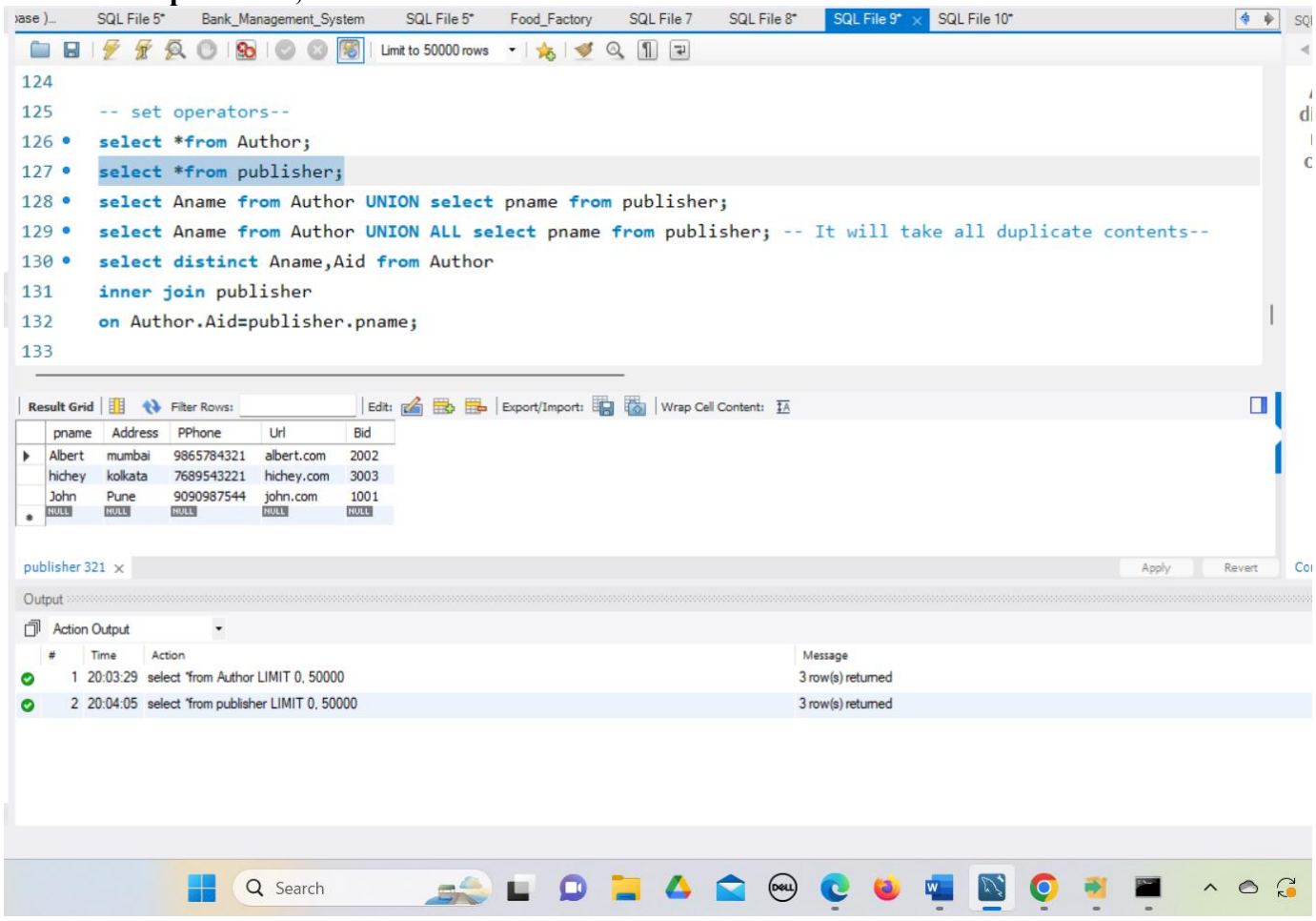
-- set operators--

select *from Author;
select *from publisher;
select Aname from Author UNION select pname from publisher;
select Aname from Author UNION ALL select pname from publisher; -- It will take all duplicate contents--

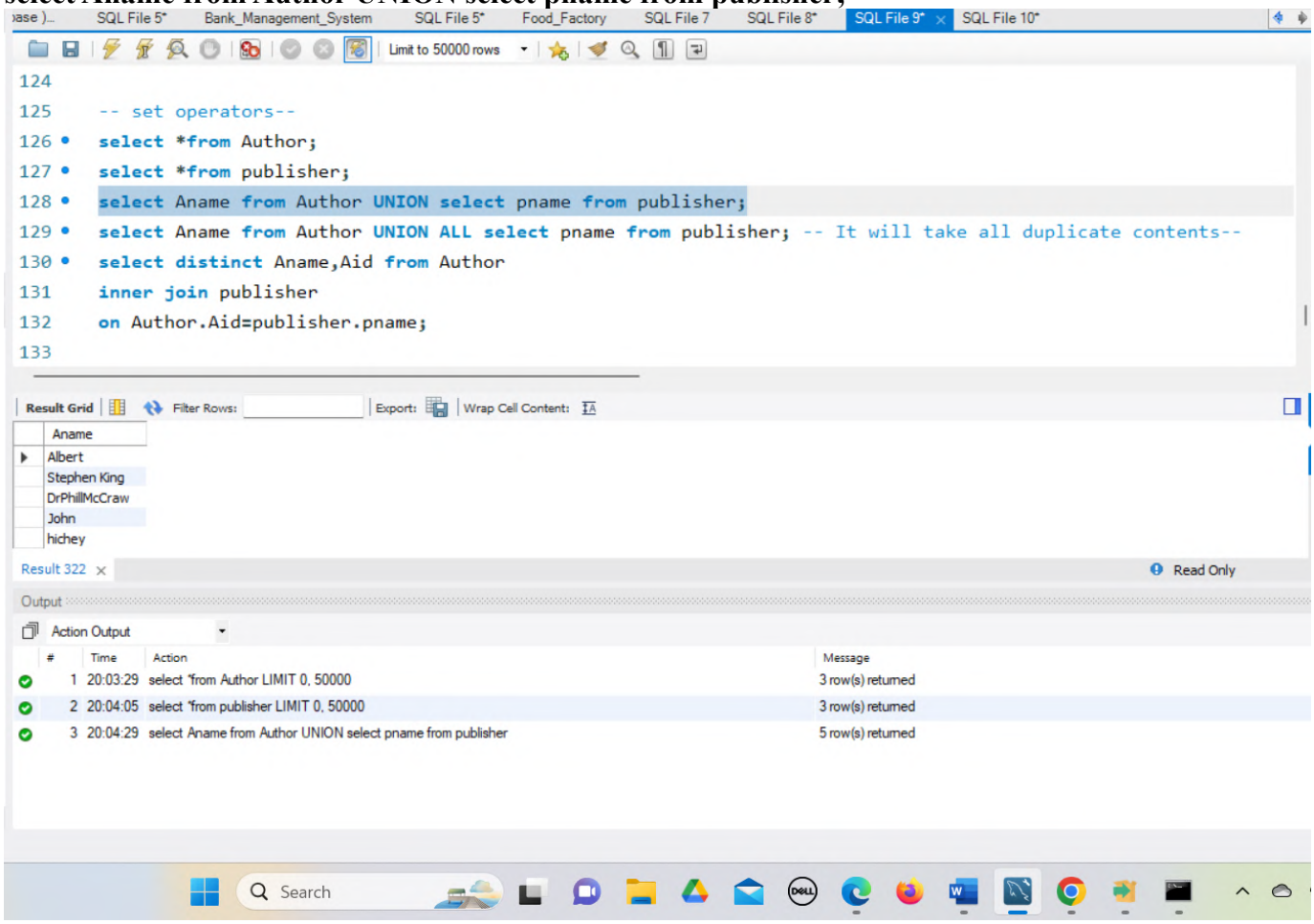
select *from Author;



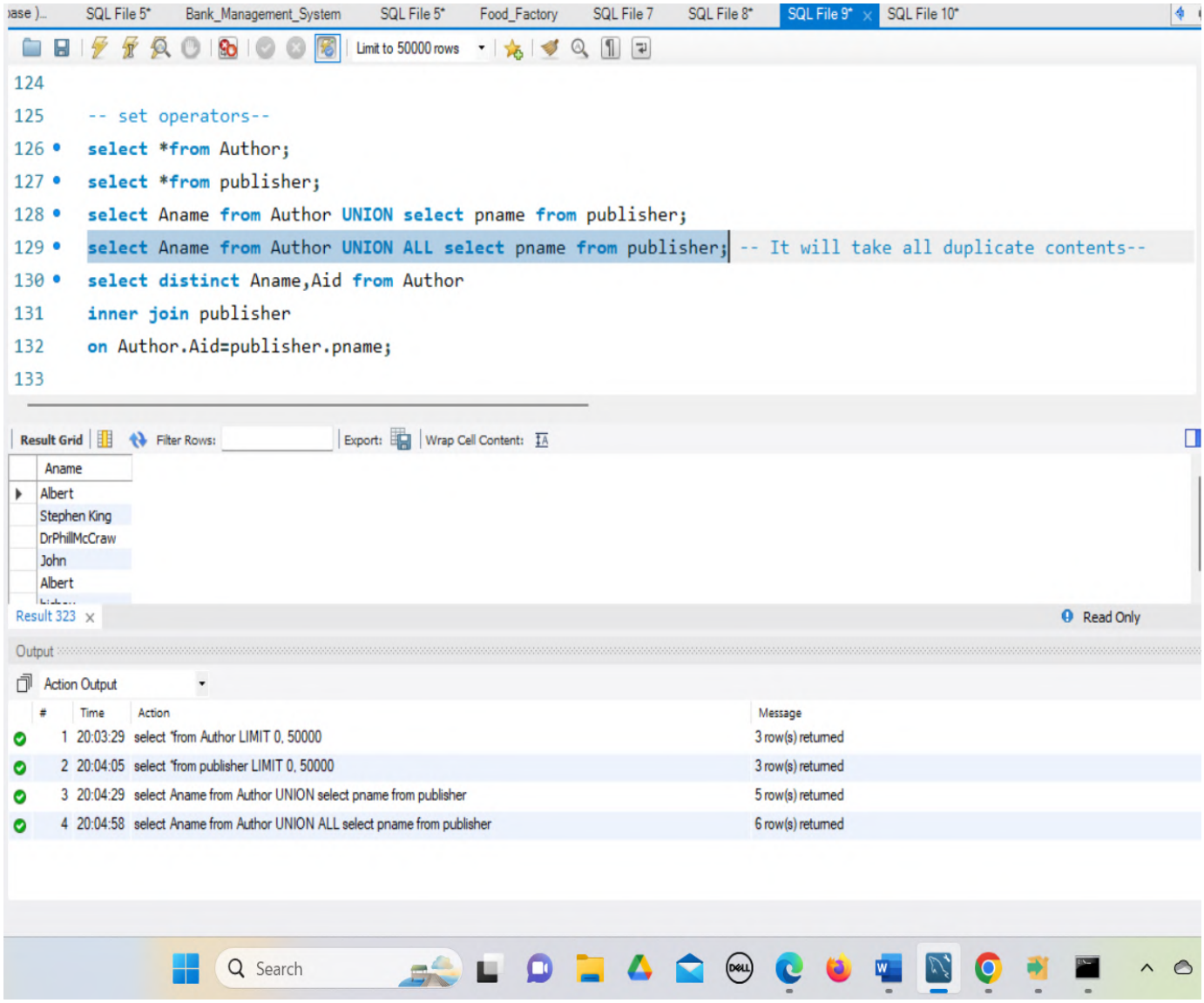
select *from publisher;



select Aname from Author UNION select pname from publisher;



select Aname from Author UNION ALL select pname from publisher; -- It will take all duplicate contents—

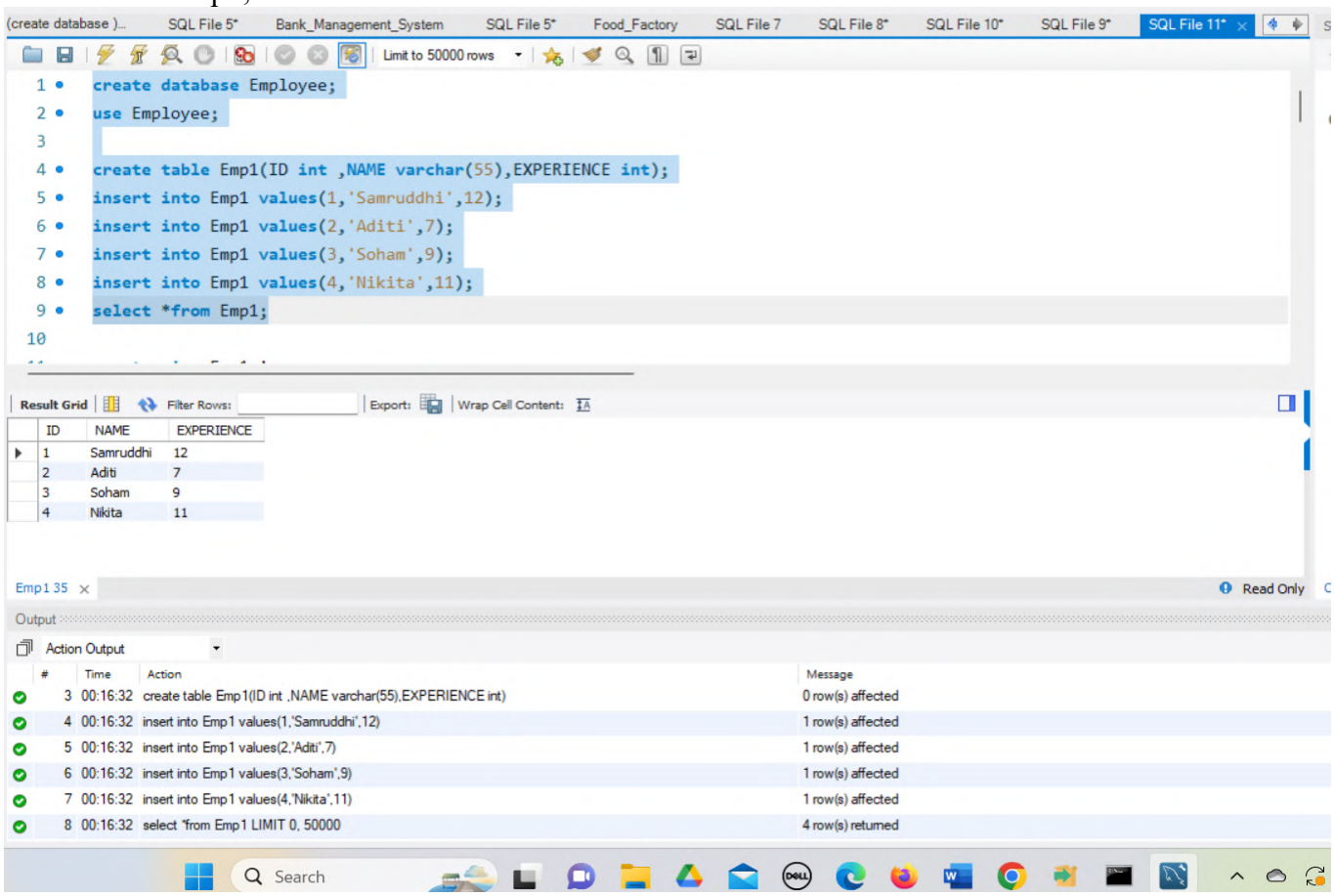


Group B: MySQL

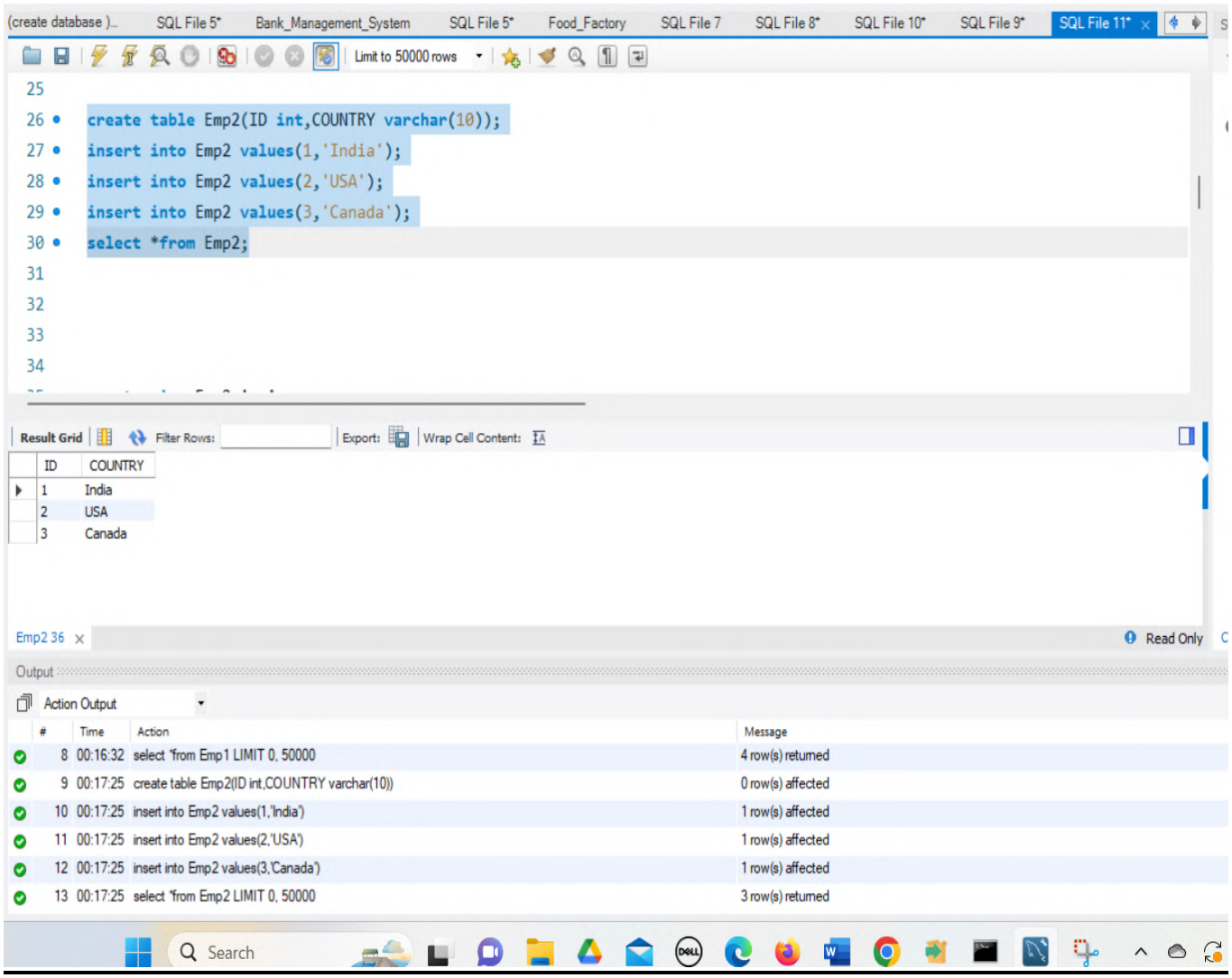
5. Execute DDL/DML statements which demonstrate the use of views. Update the base table using its corresponding view. Also consider restrictions on updatable views and perform view creation from multiple tables.

Creating a database and tables:

```
create database Employee;
use Employee;

create table Emp1(ID int ,NAME varchar(55),EXPERIENCE int);
insert into Emp1 values(1,'Samruddhi',12);
insert into Emp1 values(2,'Aditi',7);
insert into Emp1 values(3,'Soham',9);
insert into Emp1 values(4,'Nikita',11);
select *from Emp1;
```

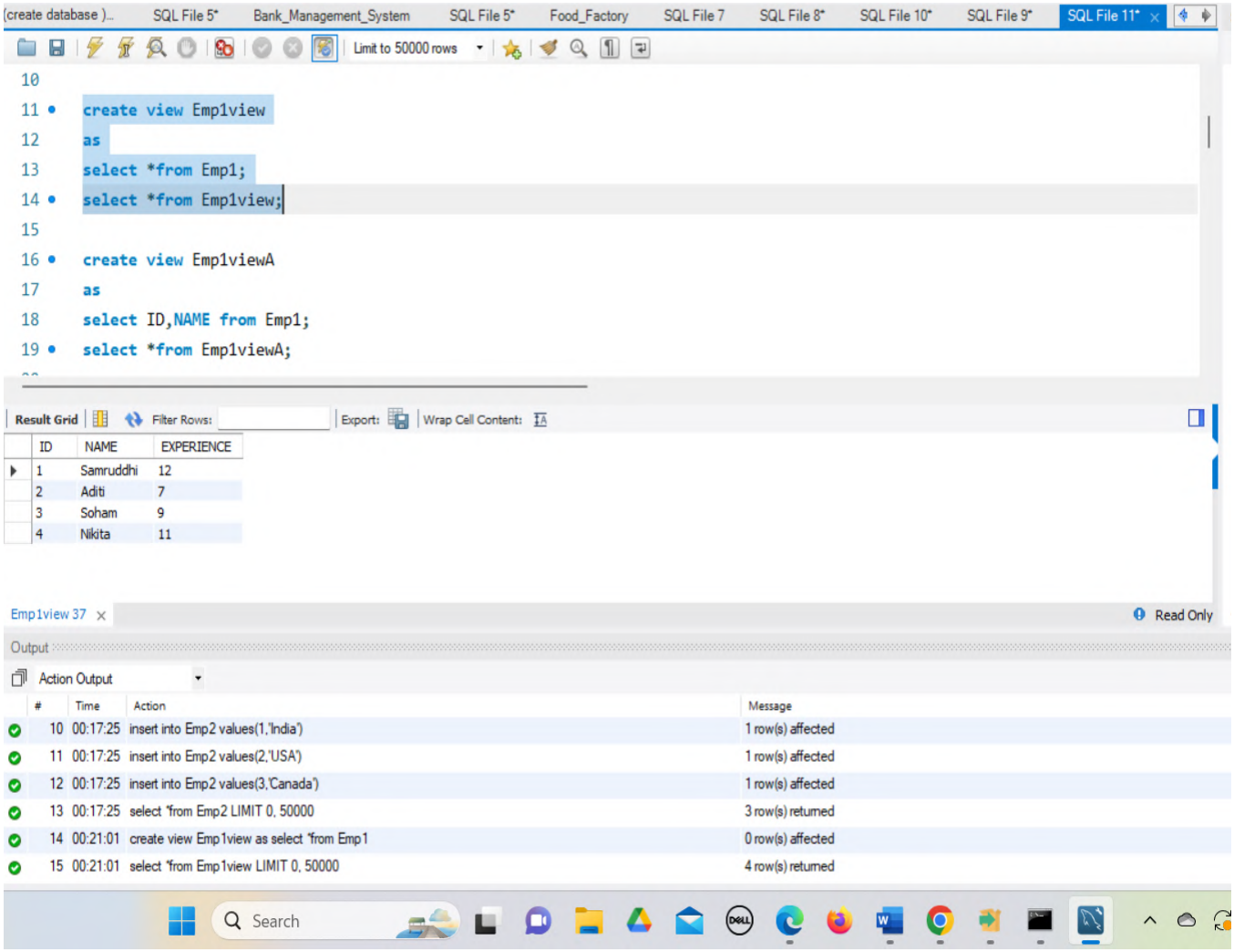


```
create table Emp2(ID int,COUNTRY varchar(10));
insert into Emp2 values(1,'India');
insert into Emp2 values(2,'USA');
insert into Emp2 values(3,'Canada');
select *from Emp2;
```

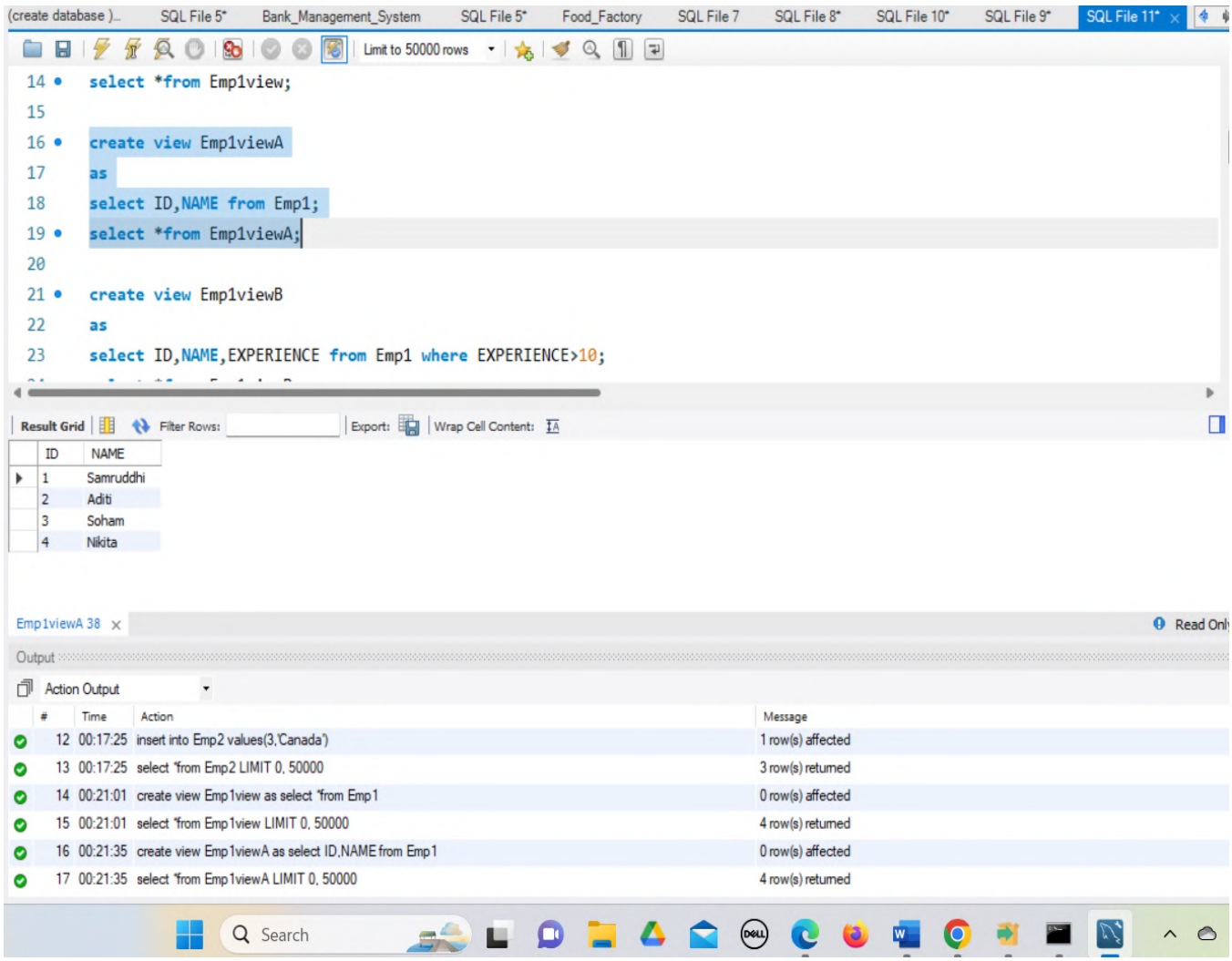


Creating a multiple views:

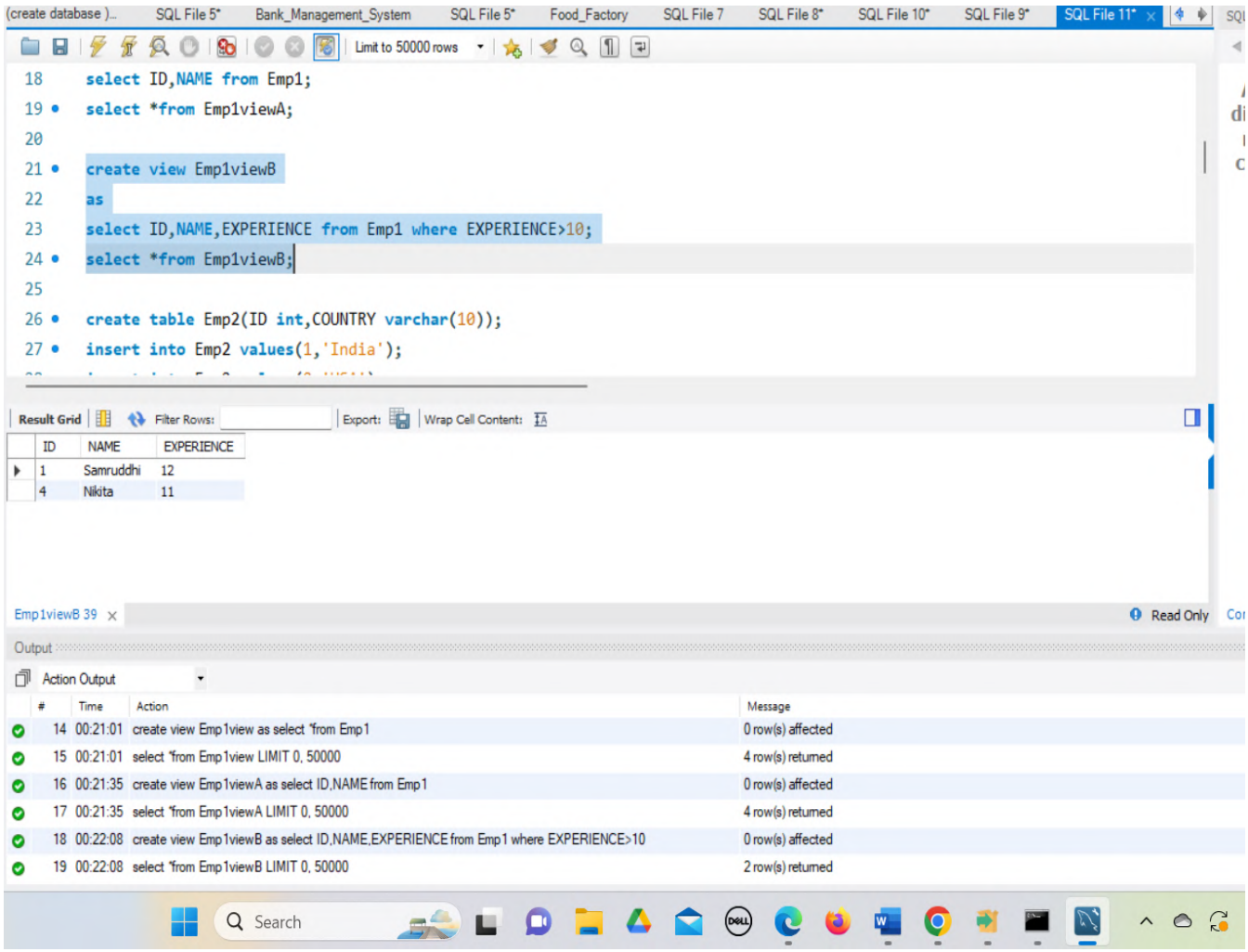
```
create view Emp1view
as
select *from Emp1;
select *from Emp1view;
```



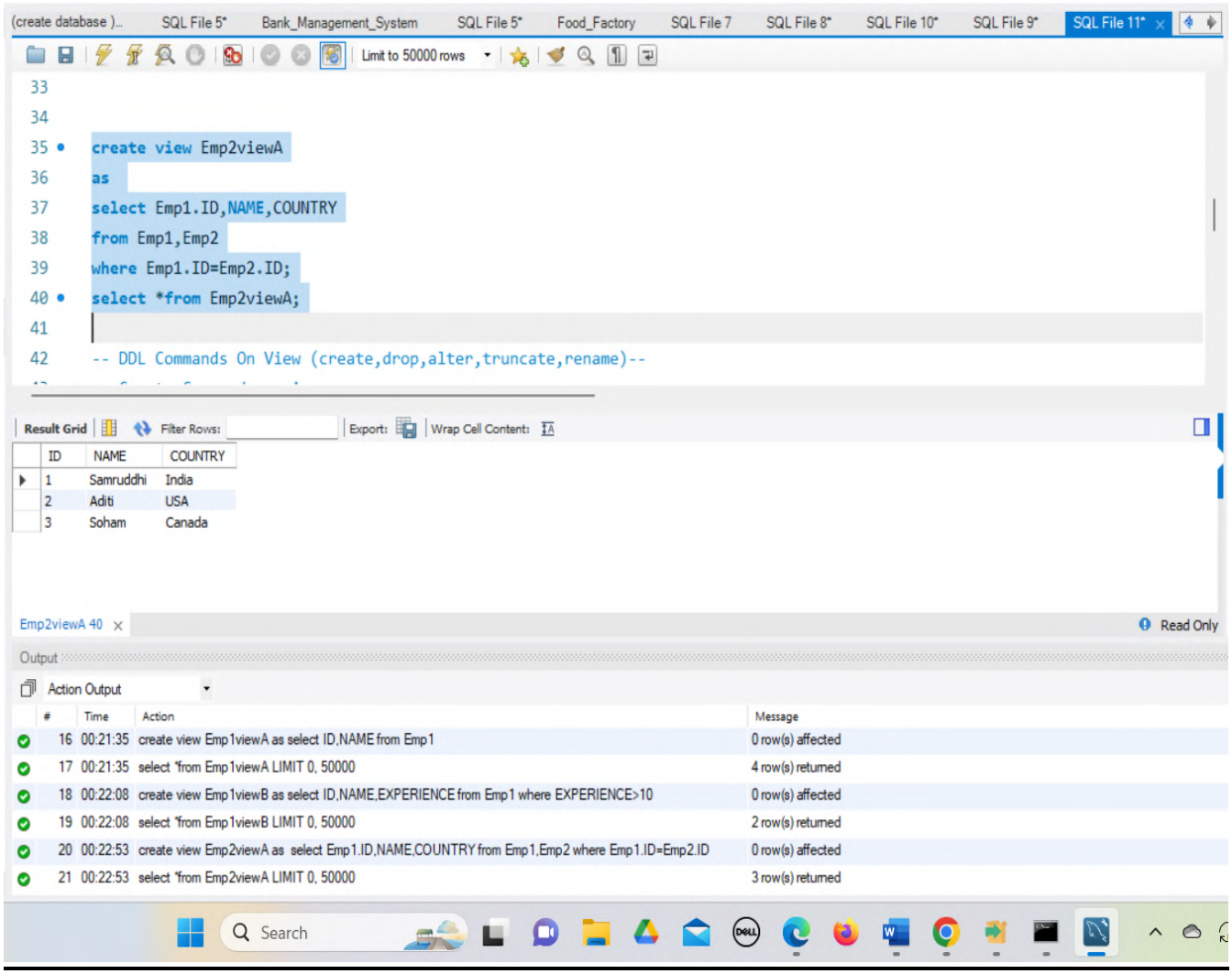
```
create view Emp1viewA
as
select ID,NAME from Emp1;
select *from Emp1viewA;
```



```
create view Emp1viewB
as
select ID,NAME,EXPERIENCE from Emp1 where EXPERIENCE>10;
select *from Emp1viewB;
```



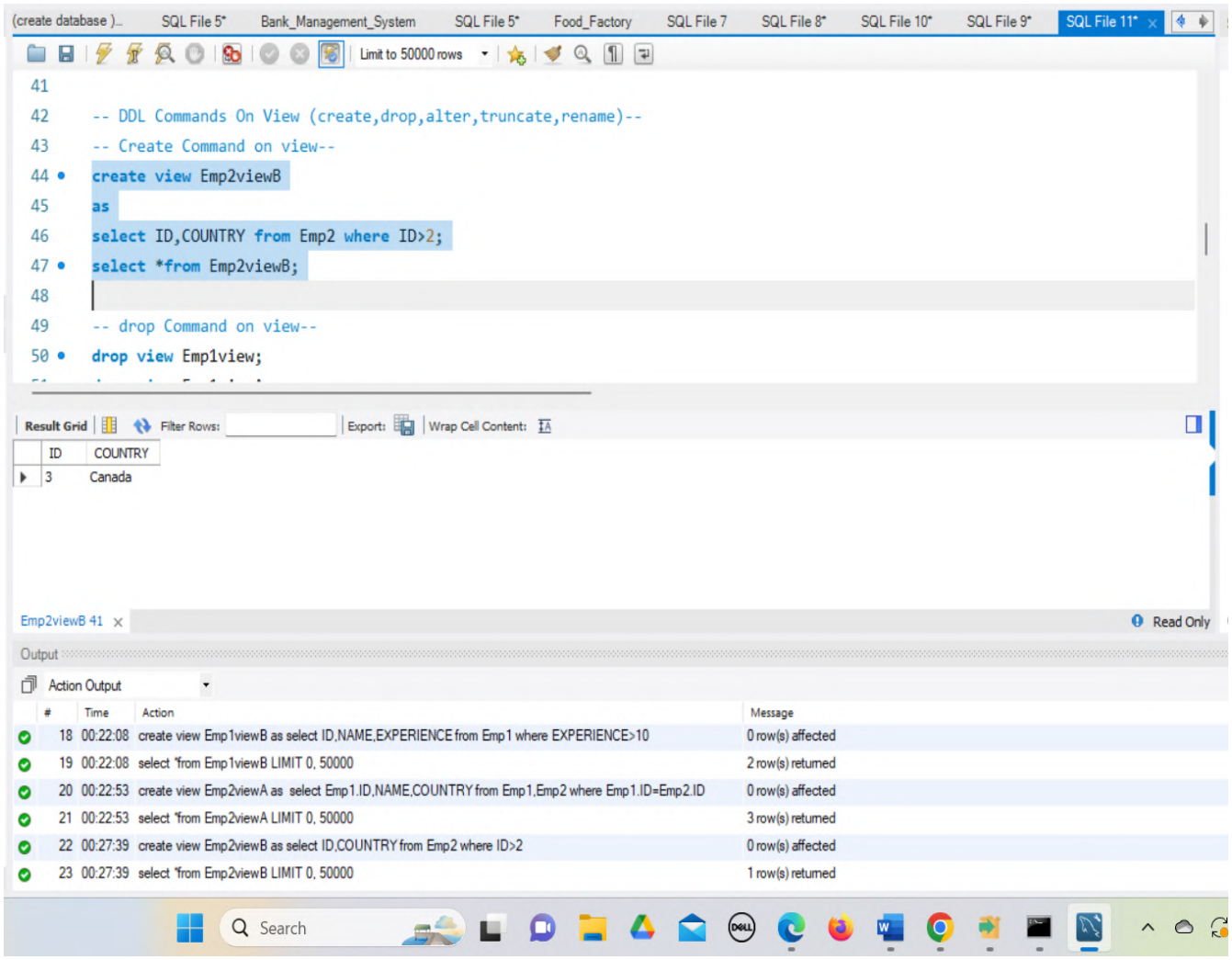
```
create view Emp2viewA
as
select Emp1.ID,NAME,COUNTRY
from Emp1,Emp2
where Emp1.ID=Emp2.ID;
select *from Emp2viewA;
```



DDL commands on view(create,drop,alter,truncate,rename):

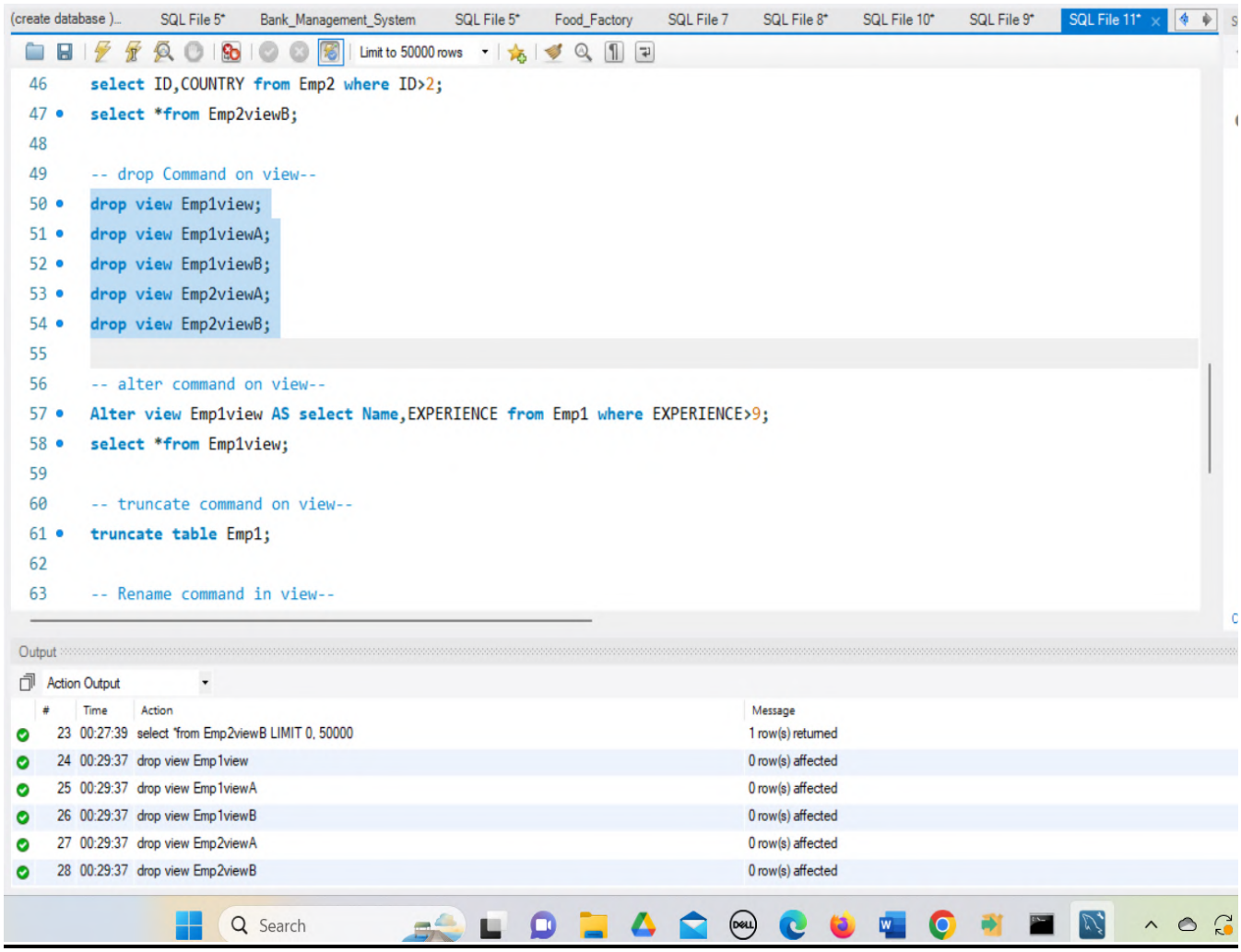
Create command:

create view Emp2viewB
as
select ID,COUNTRY from Emp2 where ID>2;
select *from Emp2viewB;



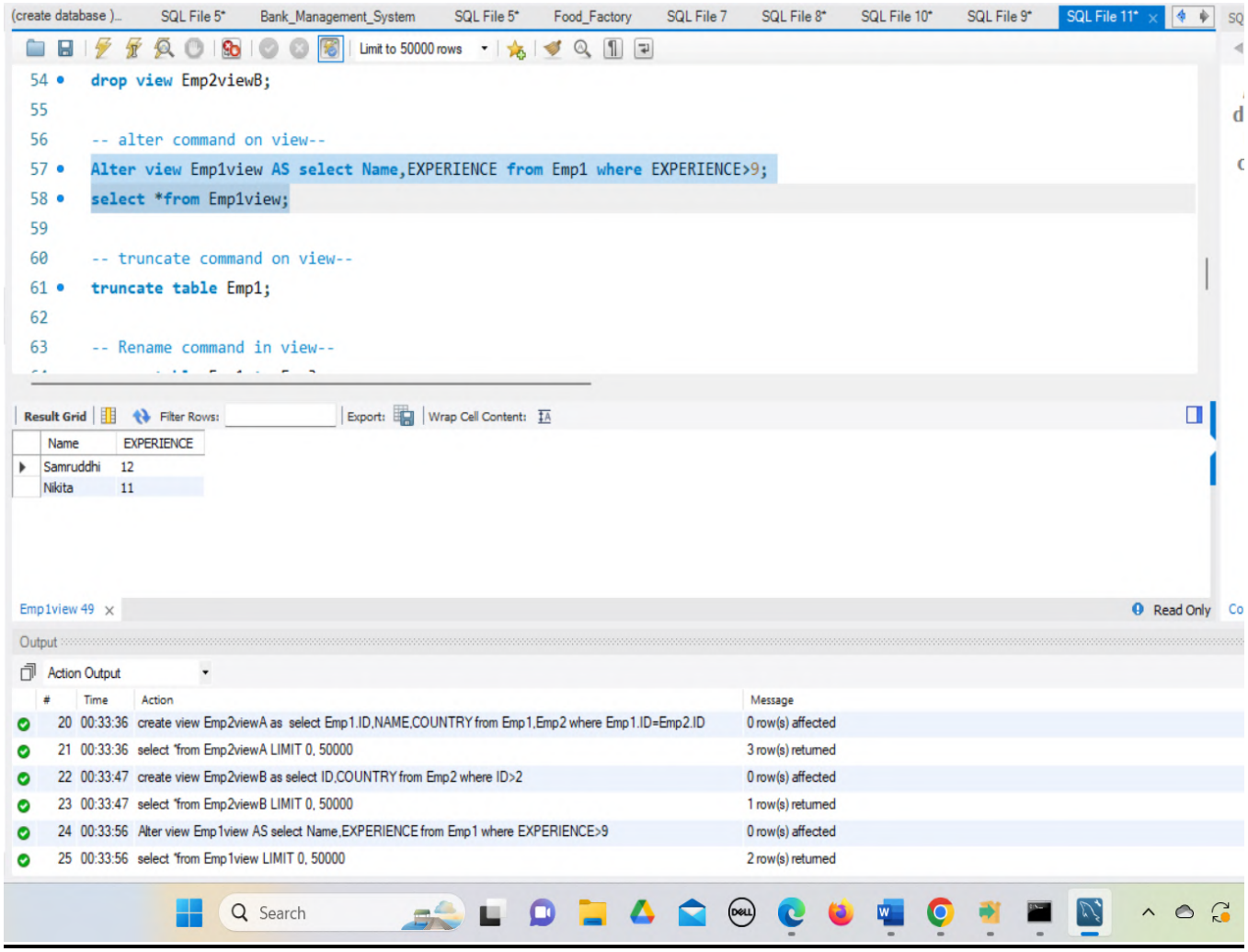
Drop command:

drop view Emp1view;
drop view Emp1viewA;
drop view Emp1viewB;
drop view Emp2viewA;
drop view Emp2viewB;



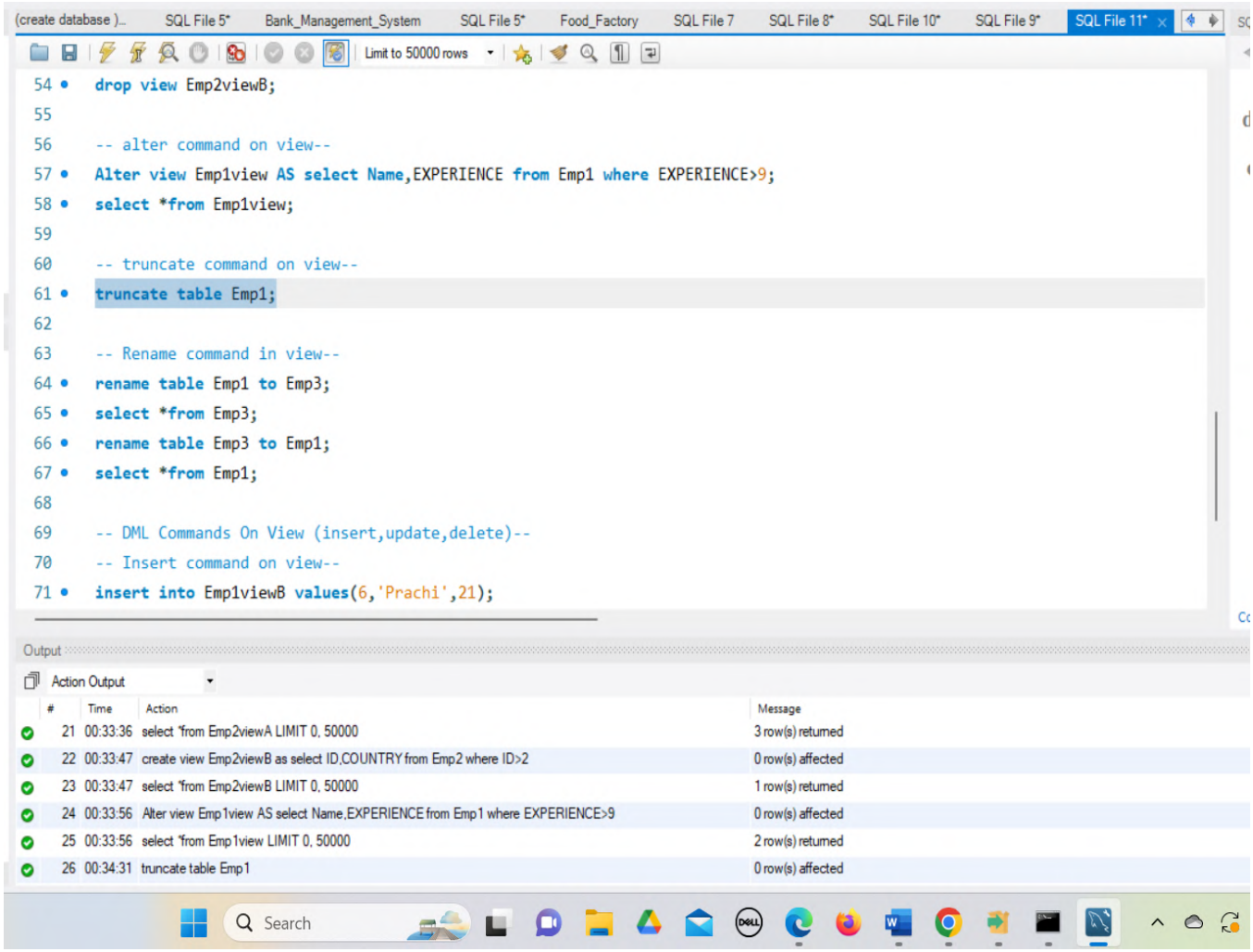
Alter command:

Alter view Emp1view AS select Name,EXPERIENCE from Emp1 where EXPERIENCE>9;
select *from Emp1view;



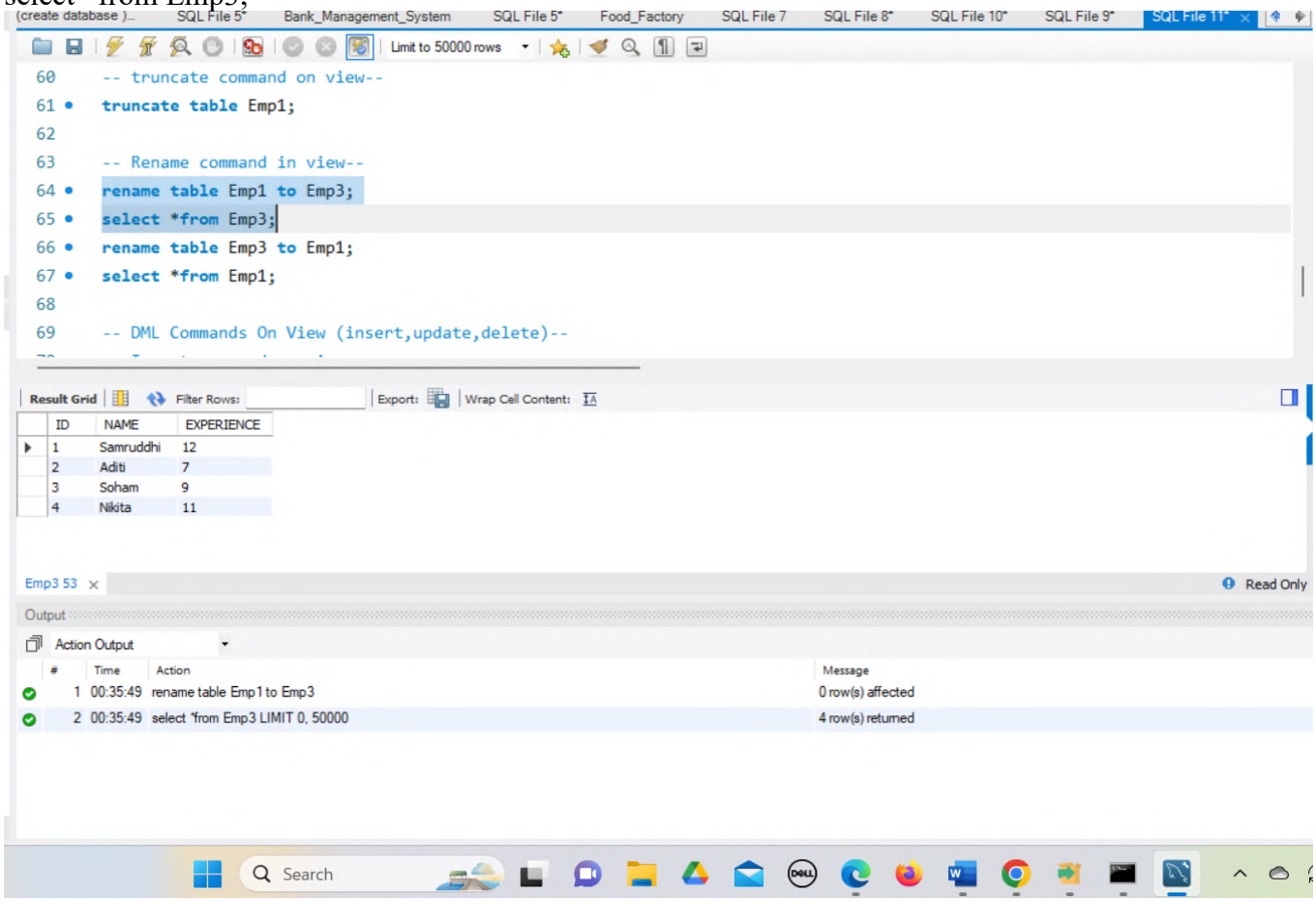
Truncate command:

truncate table Emp1;

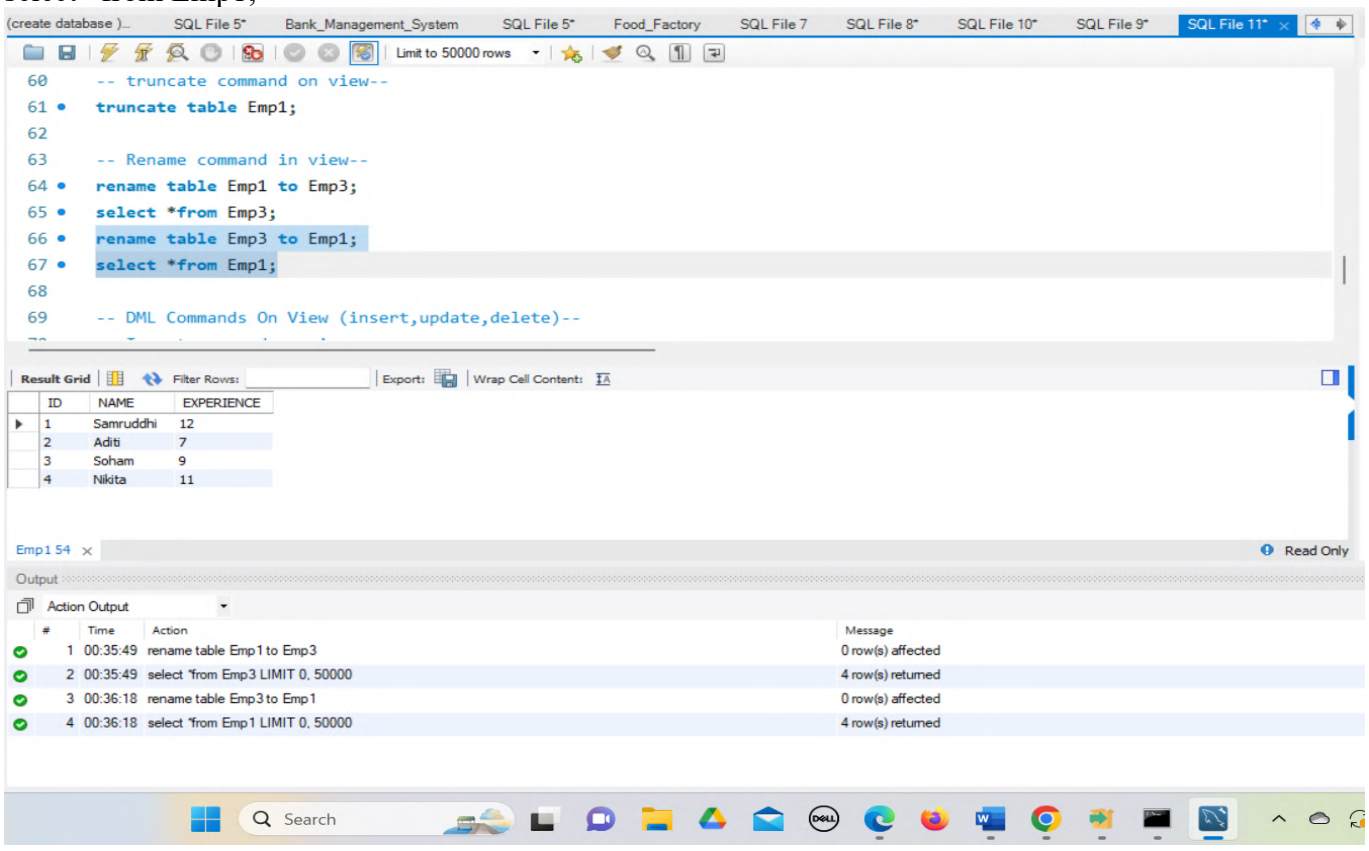


Rename command:

rename table Emp1 to Emp3;
select *from Emp3;



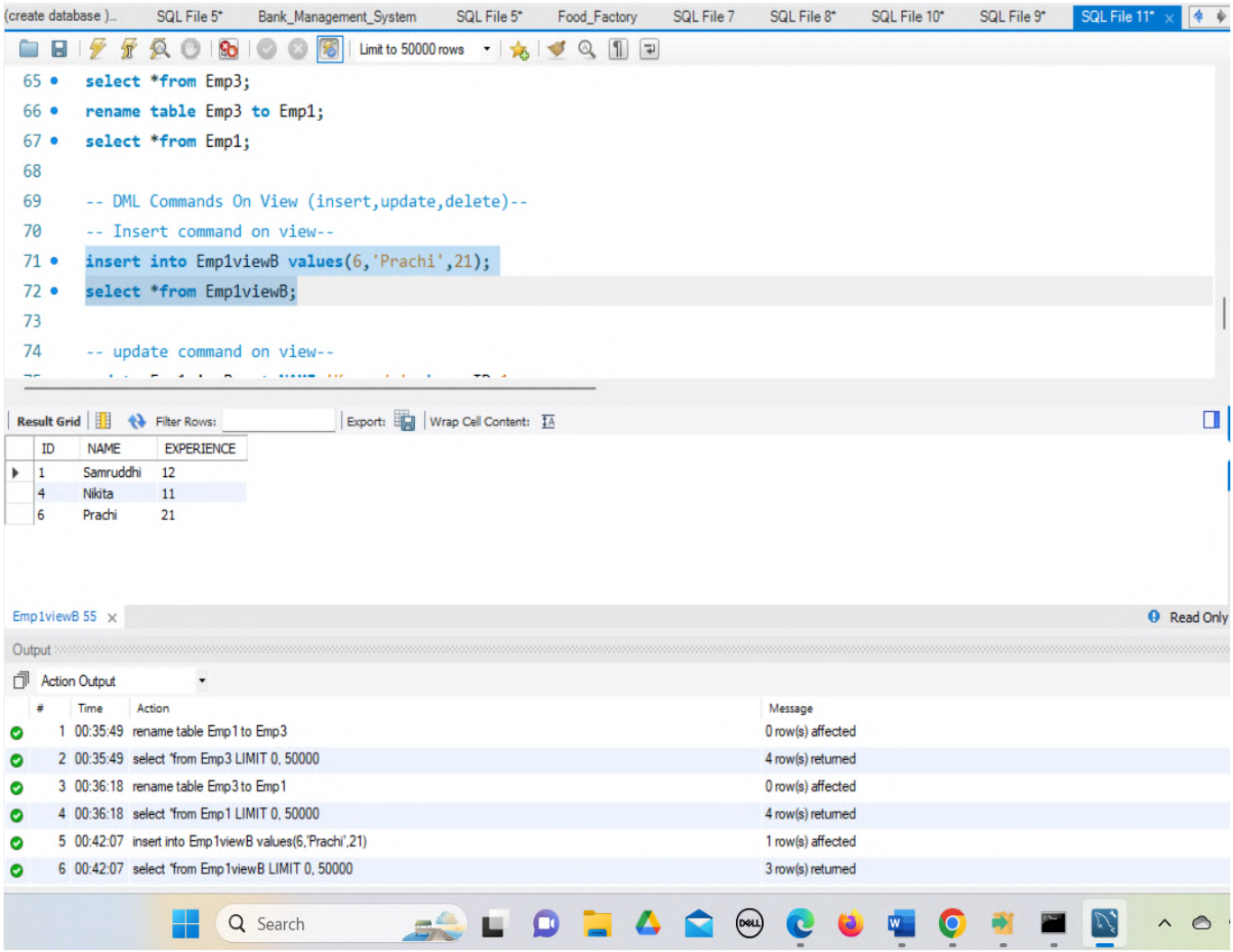
rename table Emp3 to Emp1;
select *from Emp1;



DML commands on view(insert,update,delete);

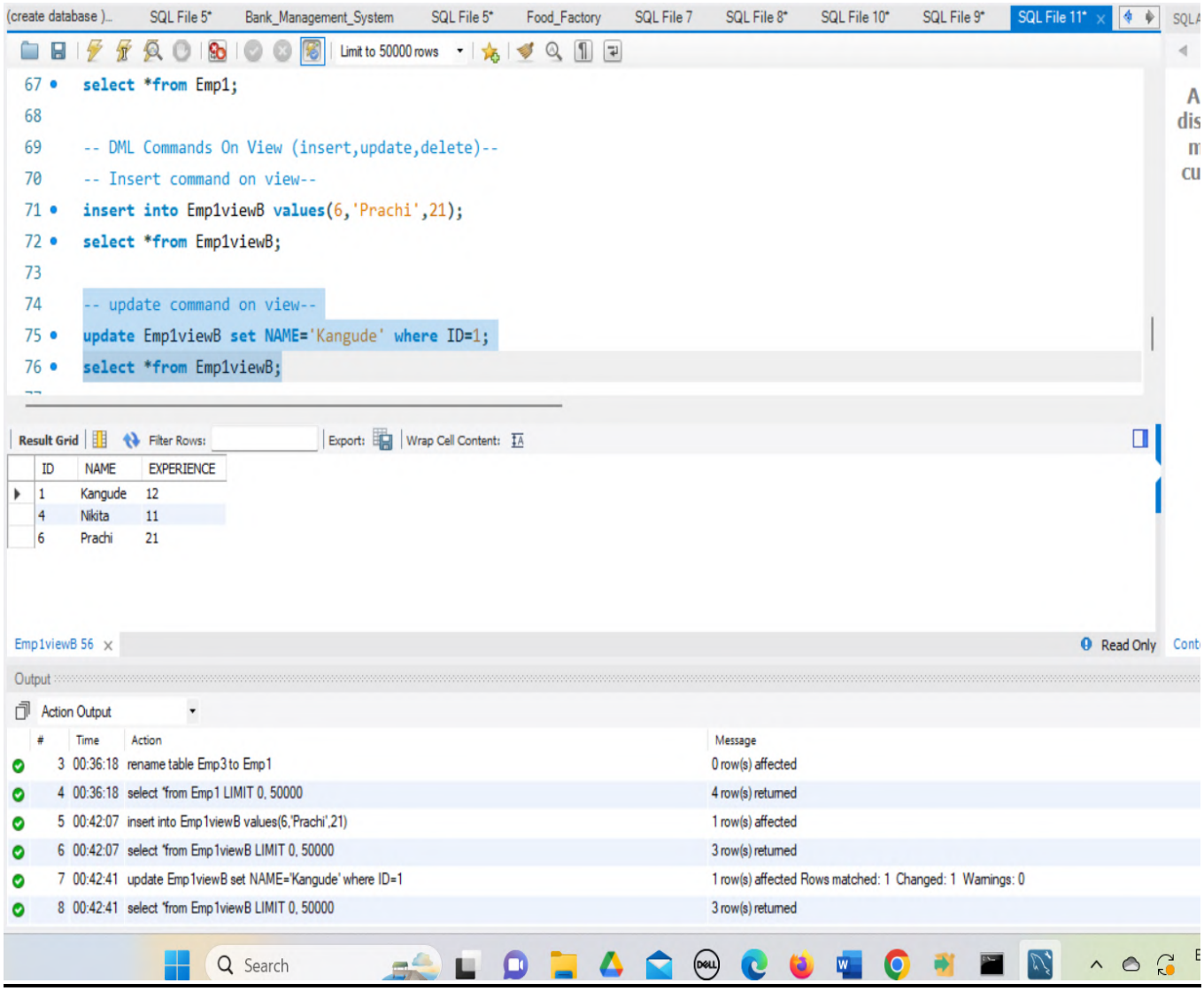
Insert command:

insert into Emp1viewB values(6,'Prachi',21);
select *from Emp1viewB;



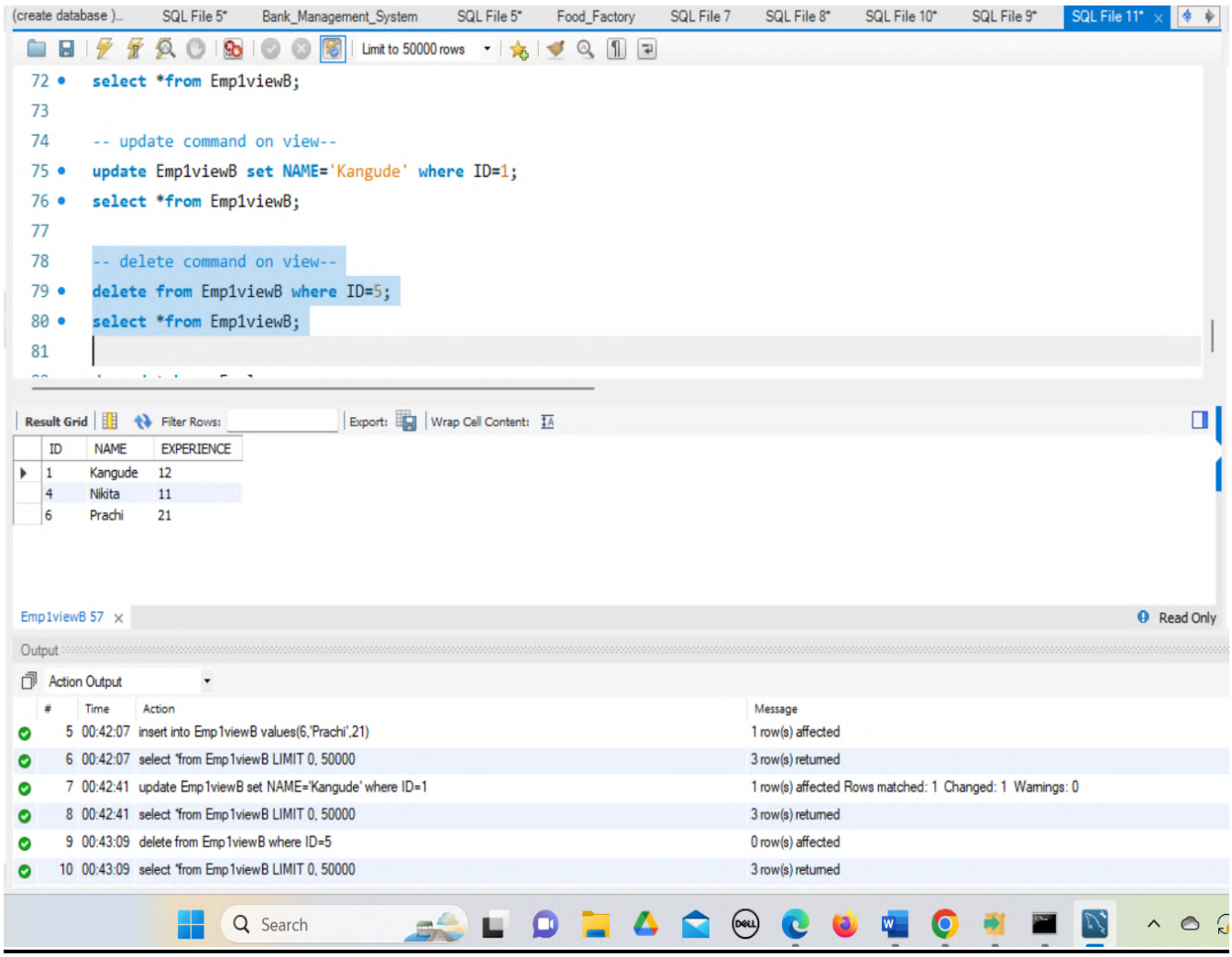
Update command:

update Emp1viewB set NAME='Kangude' where ID=1;
select *from Emp1viewB;



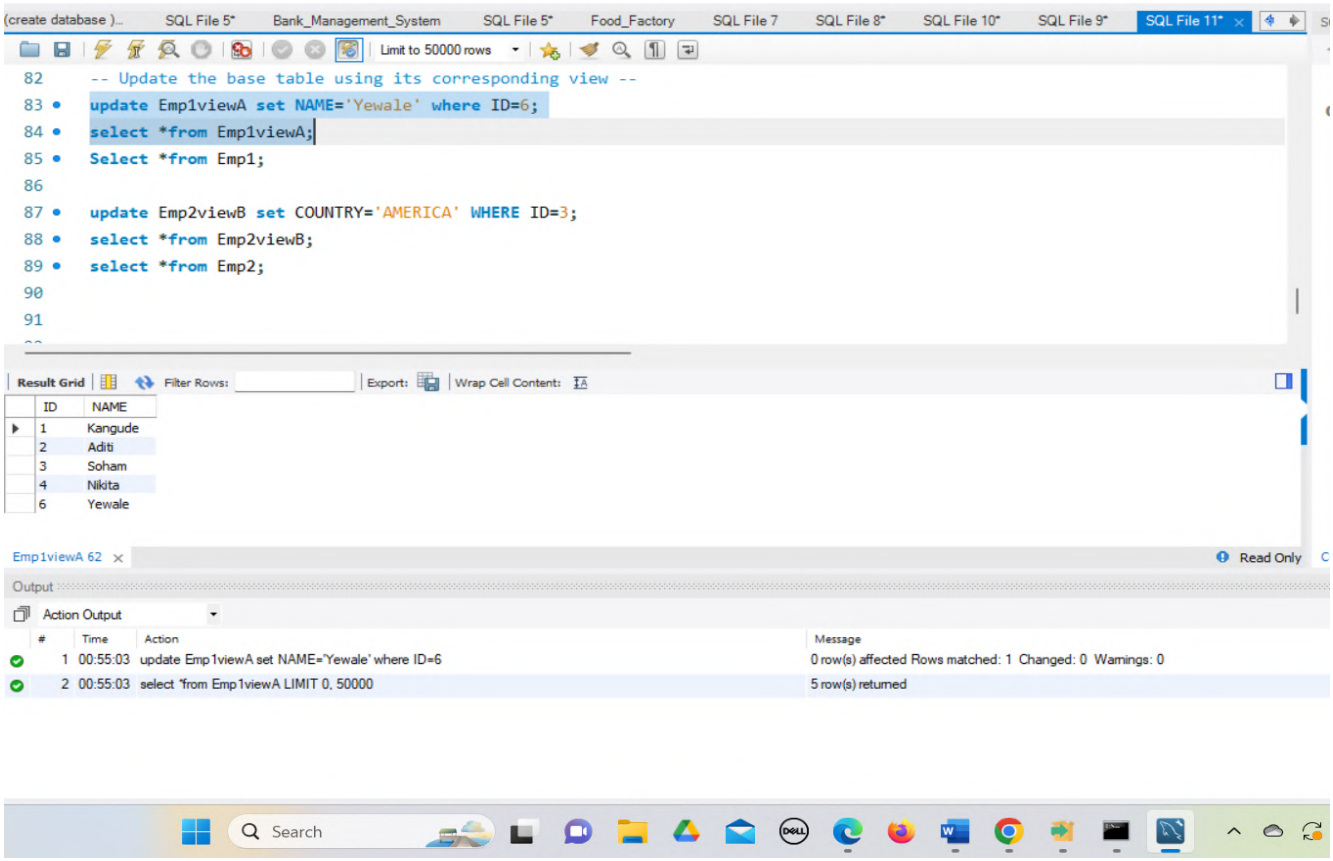
Delete command:

delete from Emp1viewB where ID=5;
select *from Emp1viewB;

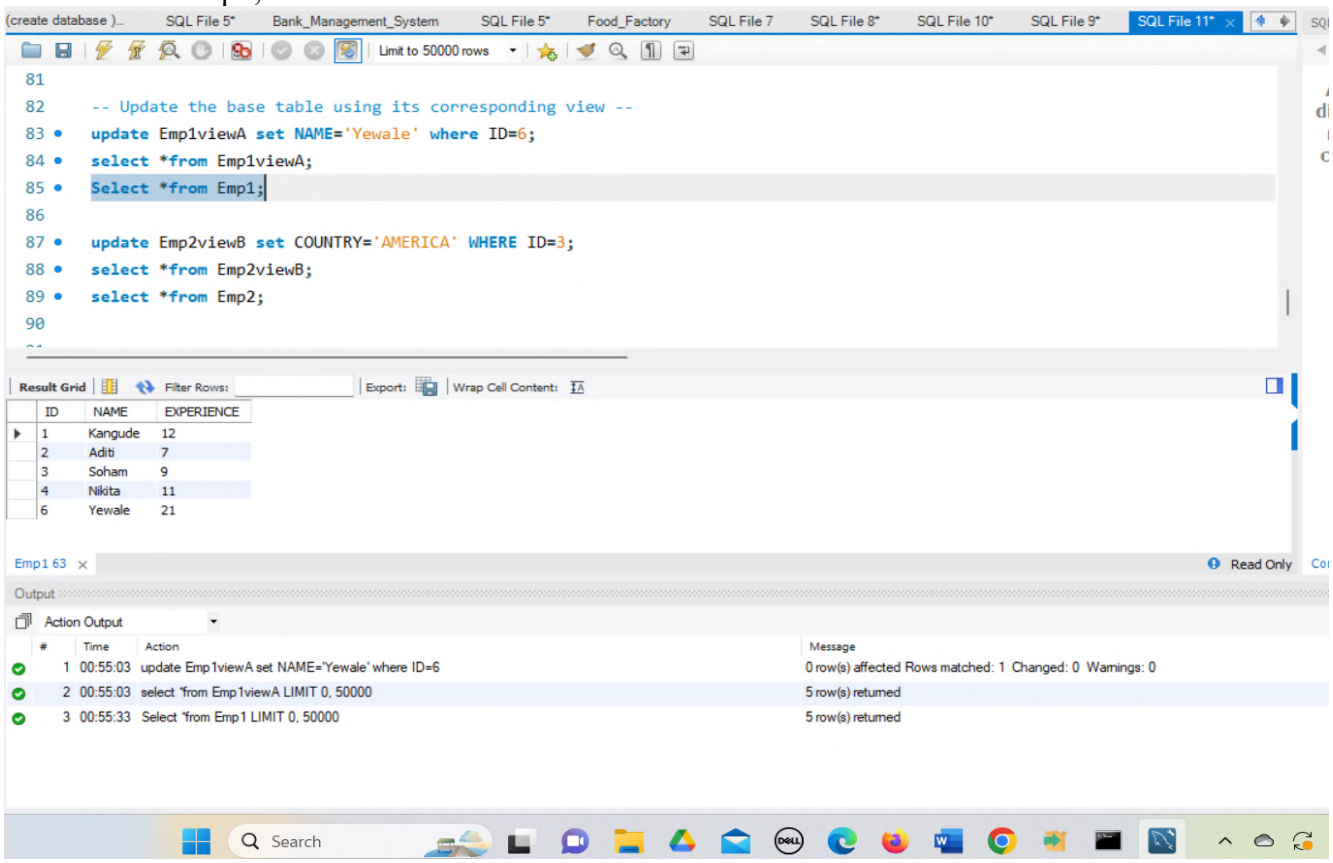


Update the base table using its corresponding view
restrictions on updatable views and perform view creation from multiple tables

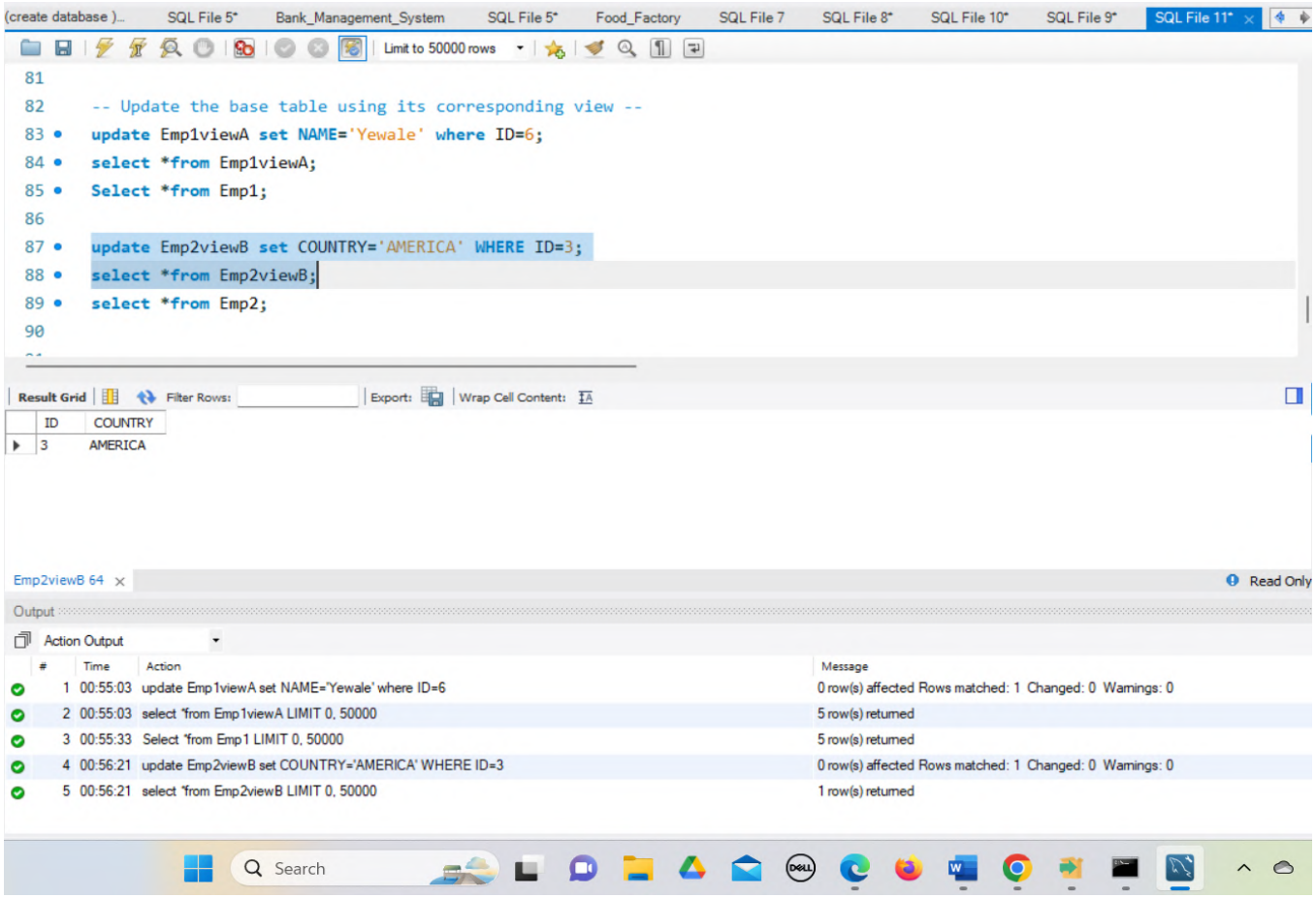
update Emp1viewA set NAME='Yewale' where ID=6;
select *from Emp1viewA;



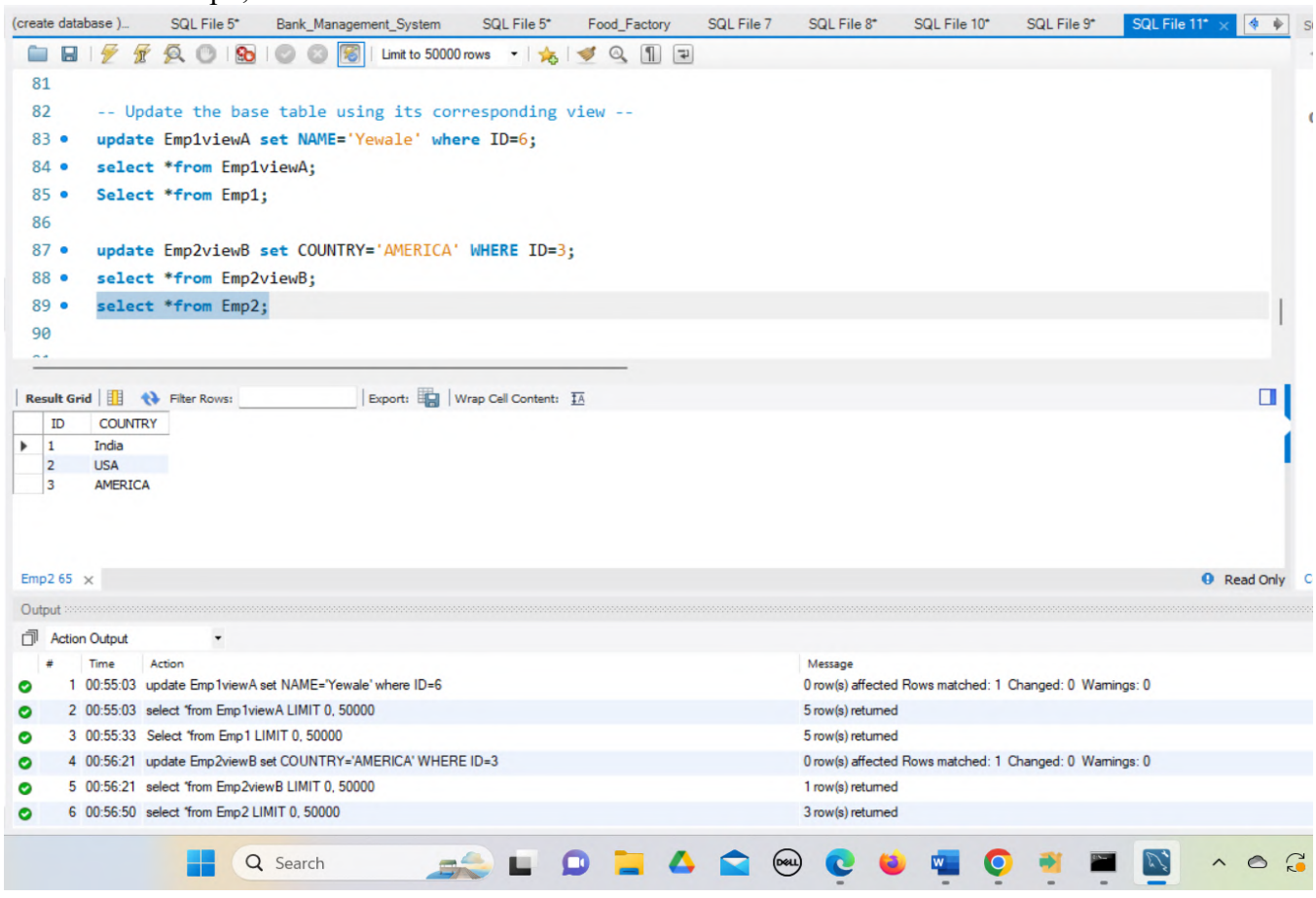
Select *from Emp1;



update Emp2viewB set COUNTRY='AMERICA' WHERE ID=3;
select *from Emp2viewB;



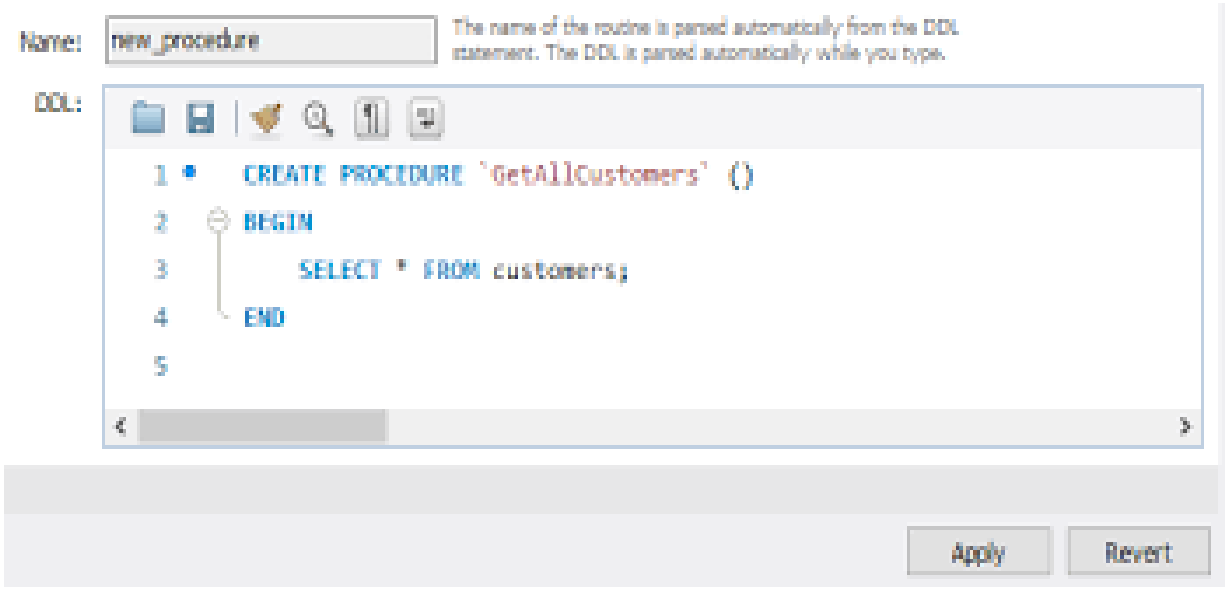
select *from Emp2;

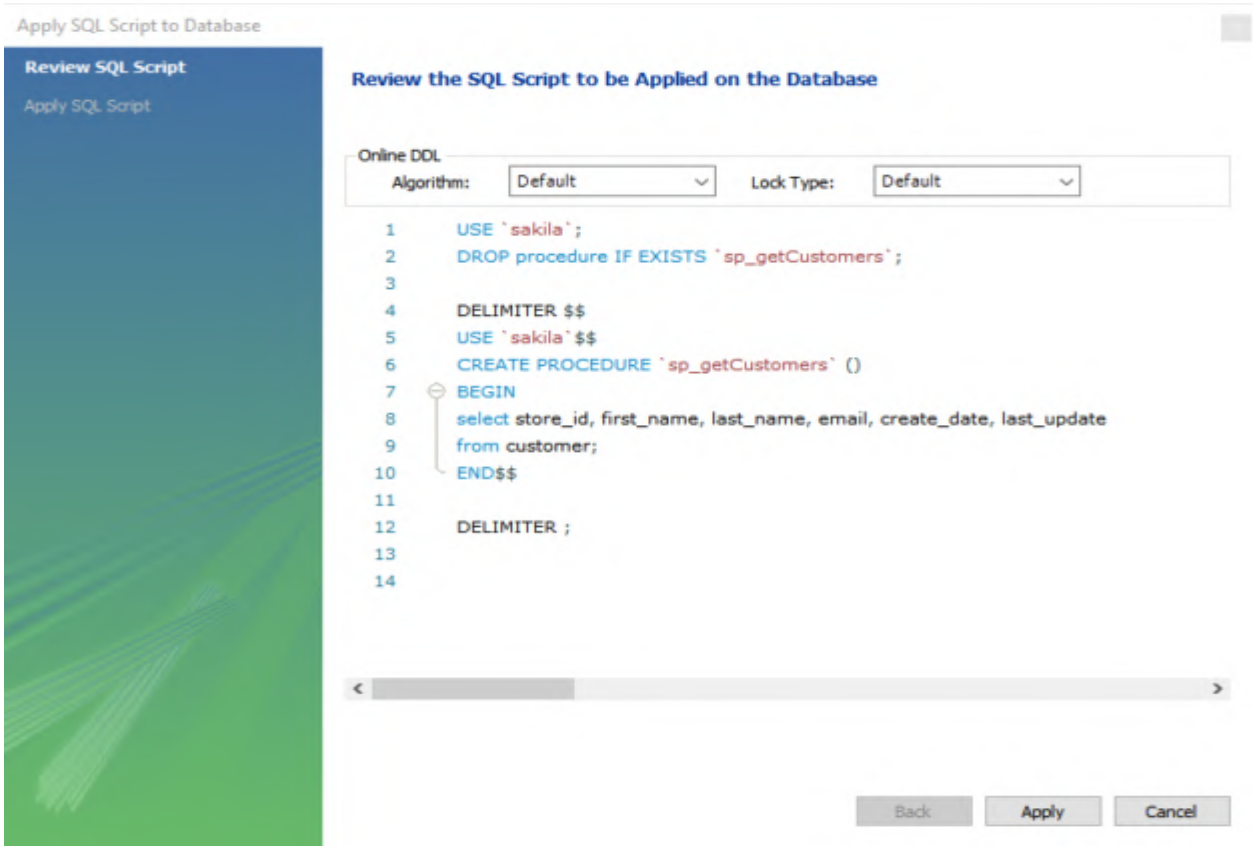


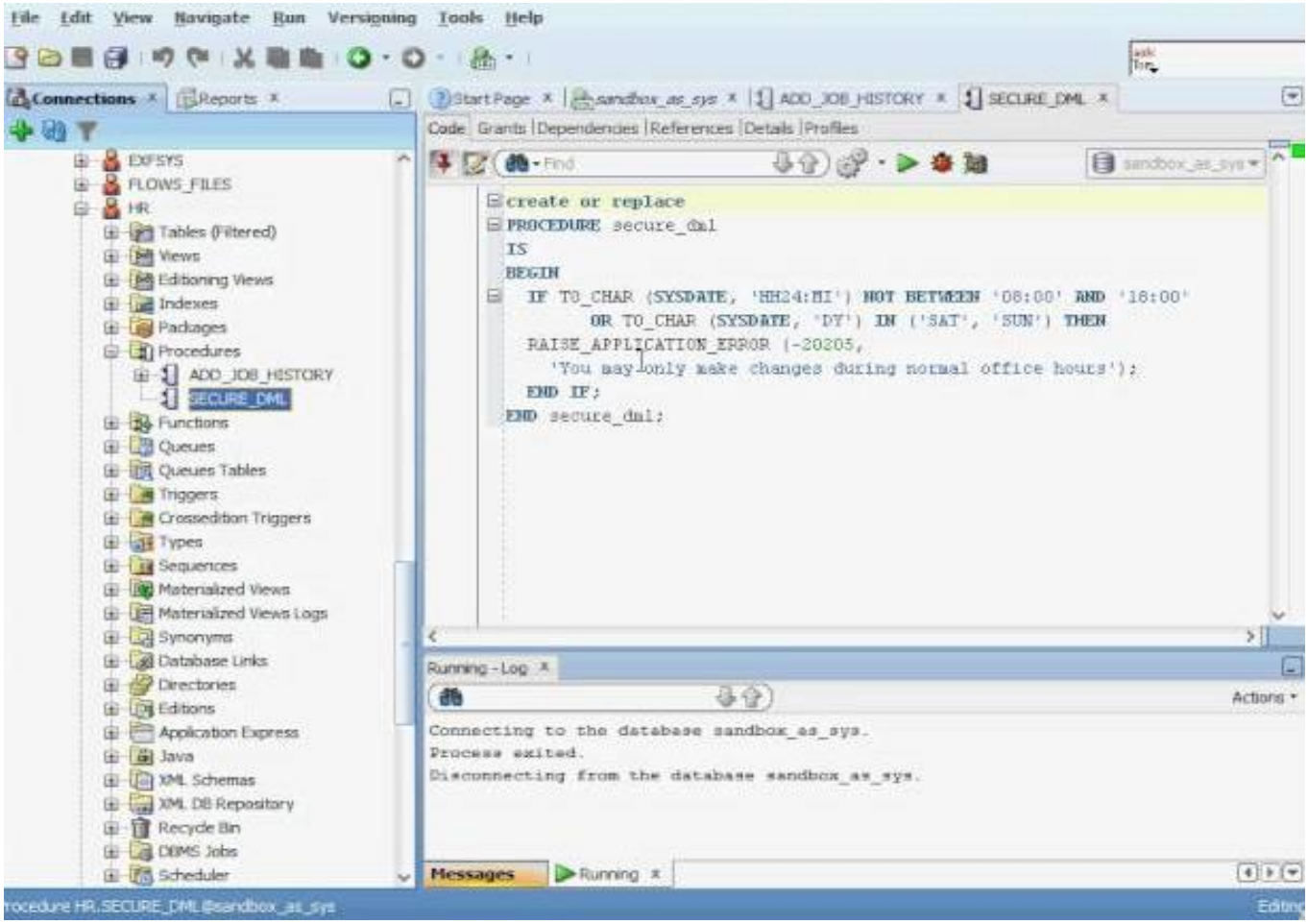
Group C: PL/SQL

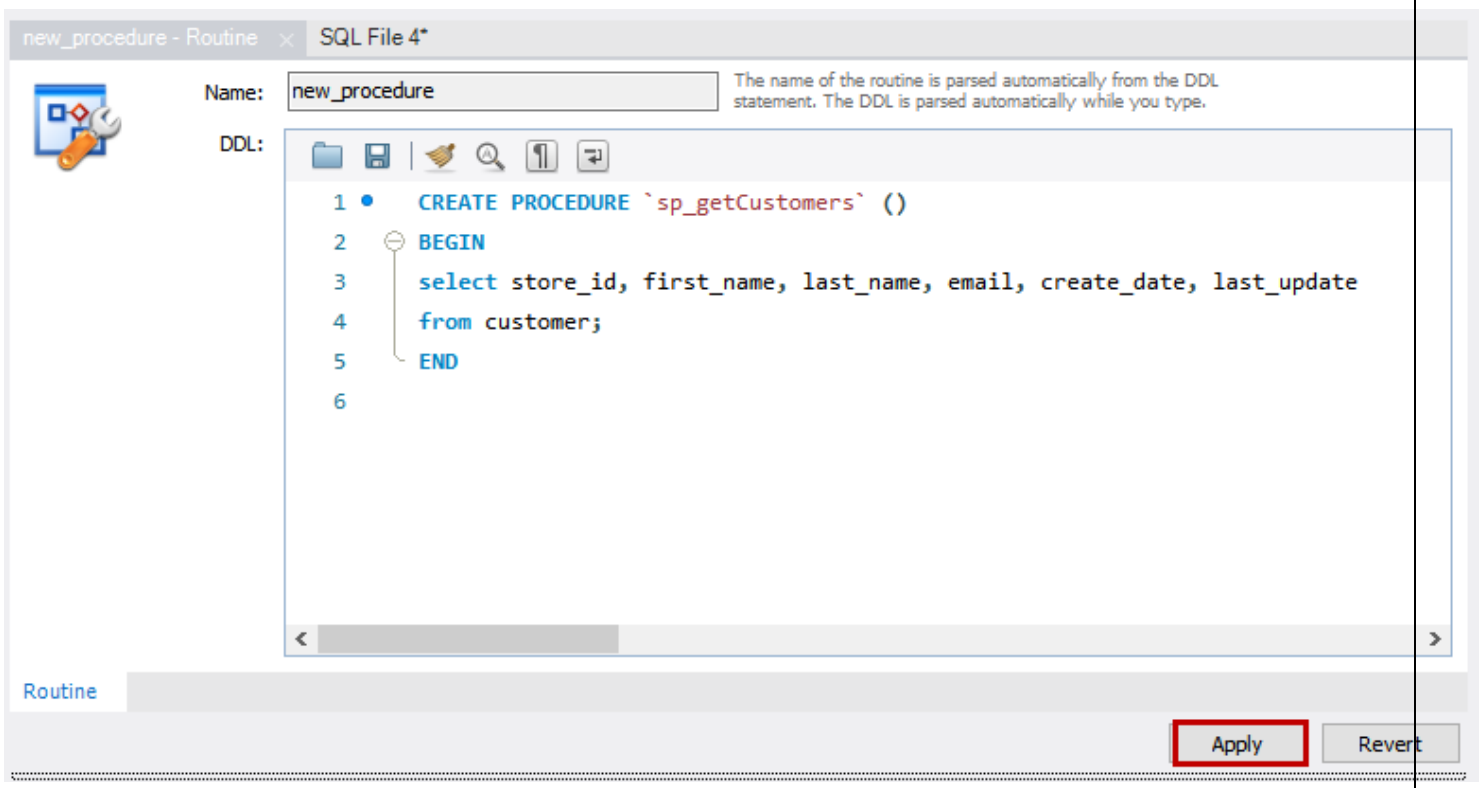
1. Write and execute PL/SQL stored procedure and function to perform a suitable task on the database. Demonstrate its use.

MySQL Stored procedure:

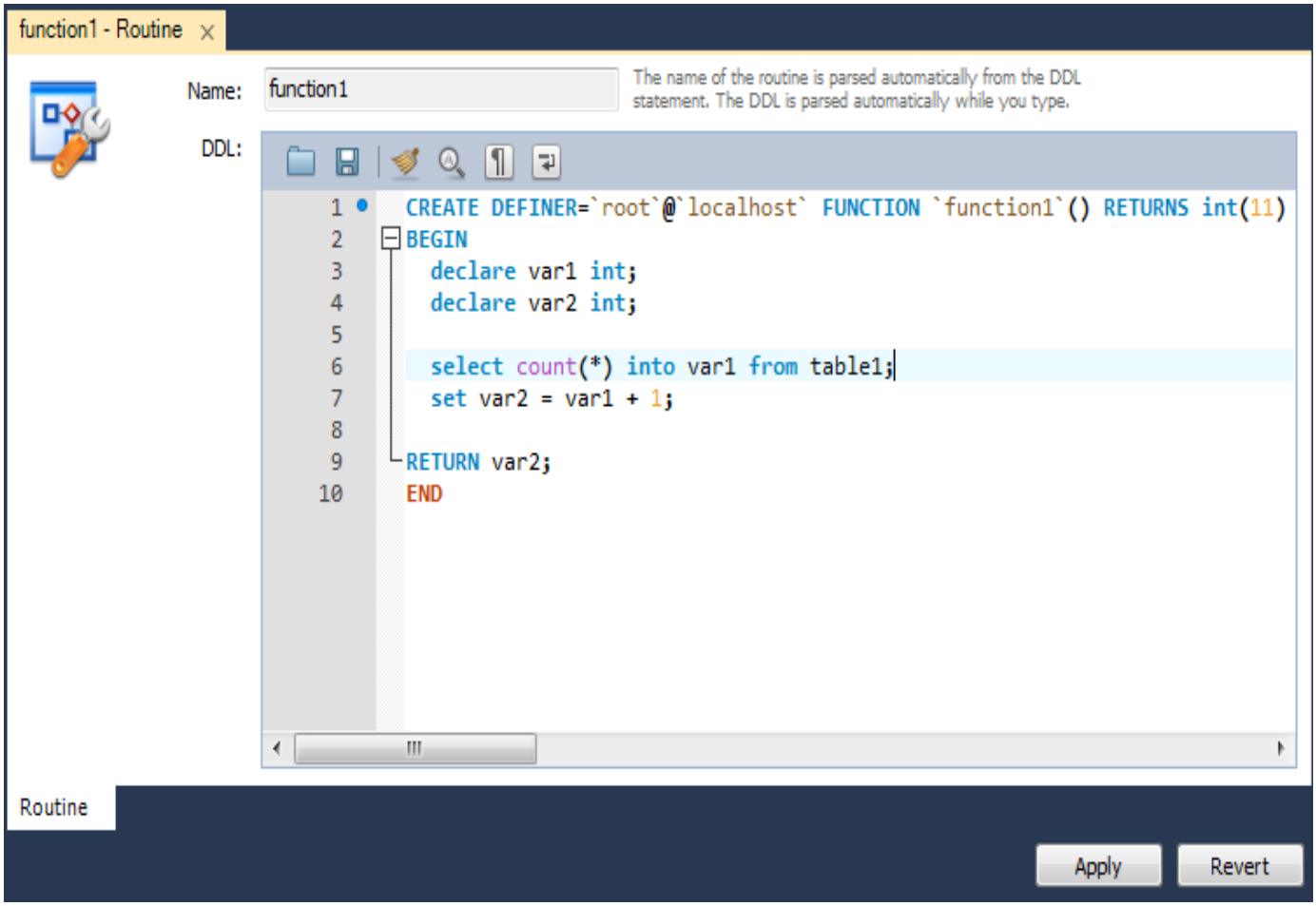


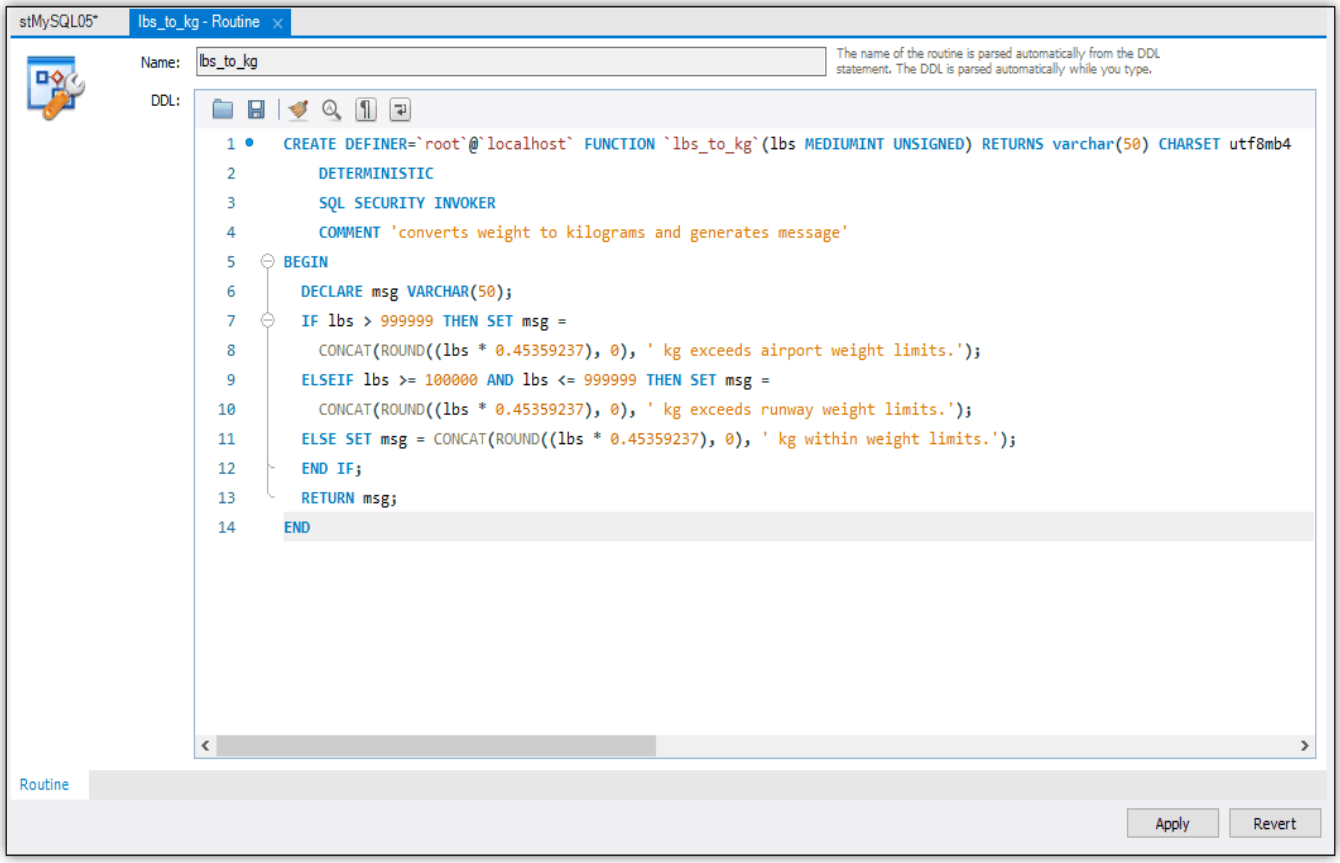



File Edit View Navigate Run Versioning Tools Help
Connections
Reports
Start Page
sandbox_as_sys
ADD_JOB_HISTORY
SECURE_DML
Code Grants Dependencies References Details Profiles
create or replace
PROCEDURE secure_dml
IS
BEGIN
IF TO_CHAR (SYSDATE, 'HH24:MI') NOT BETWEEN '08:00' AND '18:00'
OR TO_CHAR (SYSDATE, 'DY') IN ('SAT', 'SUN') THEN
RAISE_APPLICATION_ERROR (-20205,
'You may only make changes during normal office hours');
END IF;
END secure_dml;
EXFSYS
FLOWS_FILES
HR
Tables (Filtered)
Views
Editioning Views
Indexes
Packages
Procedures
ADD_JOB_HISTORY
SECURE_DML
Functions
Queues
Queues Tables
Triggers
Crossedition Triggers
Types
Sequences
Materialized Views
Materialized Views Logs
Synonyms
Database Links
Directories
Editions
Application Express
Java
XML Schemas
XML DB Repository
Recycle Bin
DBMS Jobs
Scheduler
Running - Log
Actions
Connecting to the database sandbox_as_sys.
Process exited.
Disconnecting from the database sandbox_as_sys.
Messages
Running
procedure HR.SECURE_DML@sandbox_as_sys
Editing


new_procedure - Routine
SQL File 4*
Name:
new_procedure
The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.
DDL:
1 CREATE PROCEDURE `sp_getCustomers` ()
2 BEGIN
3 select store_id, first_name, last_name, email, create_date, last_update
4 from customer;
5 END
6
Routine
Apply
Revert

PL/SQL Function:





Worksheet | Query Builder

```
1  CREATE OR REPLACE FUNCTION get_total_sales(
2      in_year PLS_INTEGER
3  )
4  RETURN NUMBER
5  IS
6      l_total_sales NUMBER := 0;
7  BEGIN
8      -- get total sales
9      SELECT SUM(unit_price * quantity)
10     INTO l_total_sales
11     FROM order_items
12     INNER JOIN orders USING (order_id)
13     WHERE status = 'Shipped'
14     GROUP BY EXTRACT(YEAR FROM order_date)
15     HAVING EXTRACT(YEAR FROM order_date) = in_year;
16
17     -- return the total sales
18     RETURN l_total_sales;
19 END;
```

Group C: PL/SQL

2. Write and execute suitable database triggers. Consider row level and statement level triggers.

```
CREATE TRIGGER login_trigger
   BEFORE INSERT ON customers
   FOR EACH ROW
INSERT INTO audit_log
SET audit_date = NOW(),
    transaction_type = 'LOGIN',
    account_id = NEW.customer_id;
```

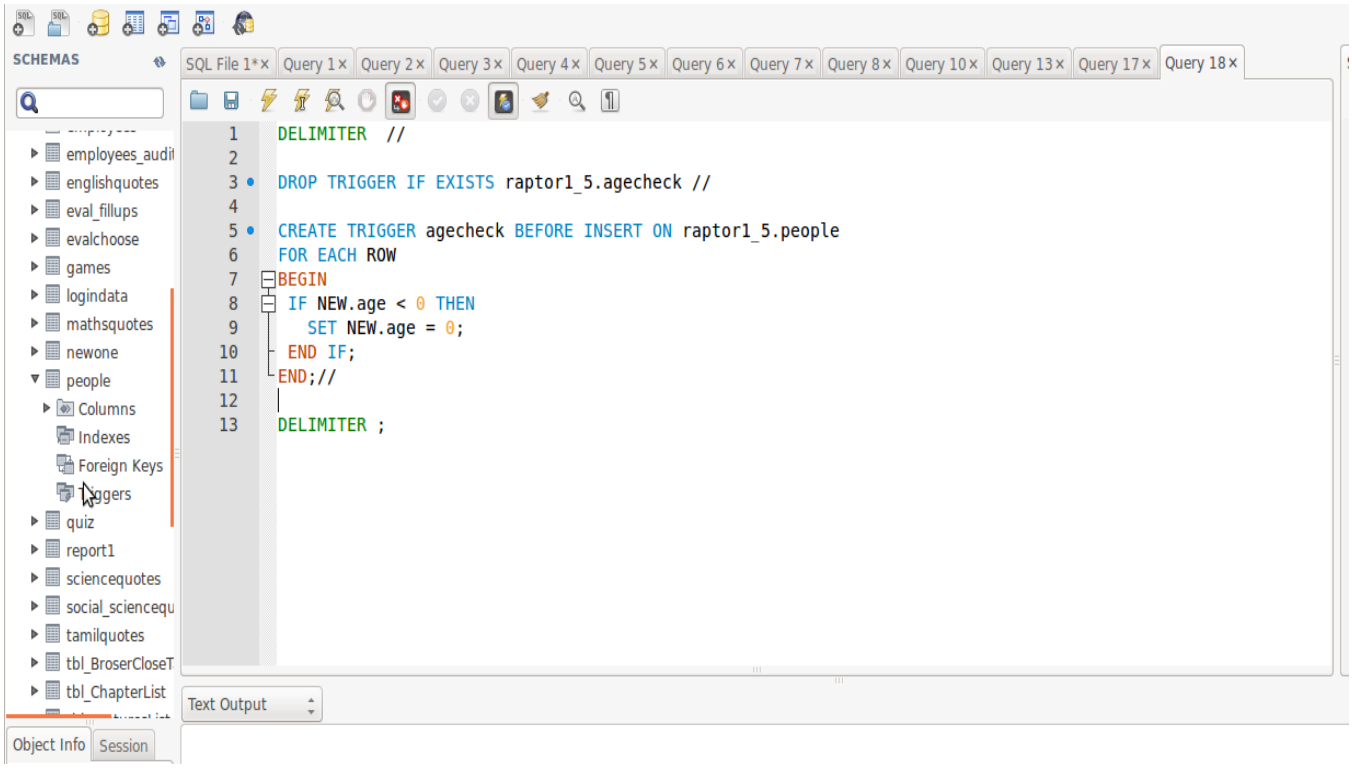
Output

Action Output

#	Time	Action	Message
1	08:16:38	CREATE TRIGGER login_trigger BEFORE INSERT ON custom...	0 row(s) affected

```
USE `ga_abtest_logging`;
DELIMITER $$
CREATE TRIGGER avoid_empty
    BEFORE INSERT ON daily_analytics
        FOR EACH ROW
        BEGIN
            IF profileID = ''
                THEN SET profileID = NULL
            END IF;
        END$$
```

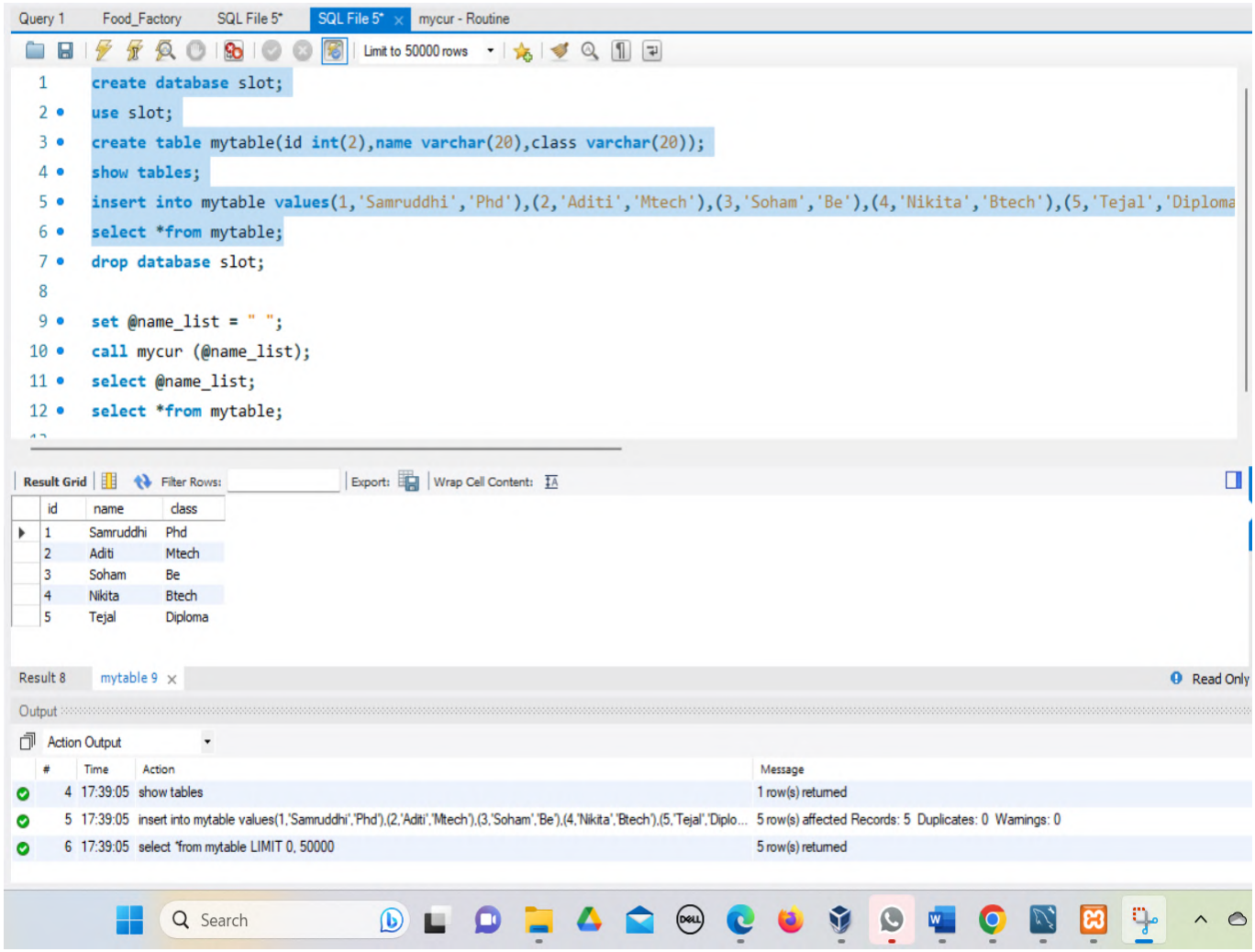
```sql
CREATE TRIGGER book_update_audit_trigger
AFTER UPDATE ON book FOR EACH ROW
BEGIN
    INSERT INTO book_audit_log (
        book_id,
        old_row_data,
        new_row_data,
        dml_type,
        dml_timestamp,
        dml_created_by
    )
    VALUES(
        NEW.id,
        JSON_OBJECT(
            "title", OLD.title,
            "author", OLD.author,
            "price_in_cents", OLD.price_in_cents,
            "publisher", OLD.publisher
        ),
        JSON_OBJECT(
            "title", NEW.title,
            "author", NEW.author,
            "price_in_cents", NEW.price_in_cents,
            "publisher", NEW.publisher
        ),
        'UPDATE',
        CURRENT_TIMESTAMP,
        @logged_user
    );
END
```



Group C: PL/SQL

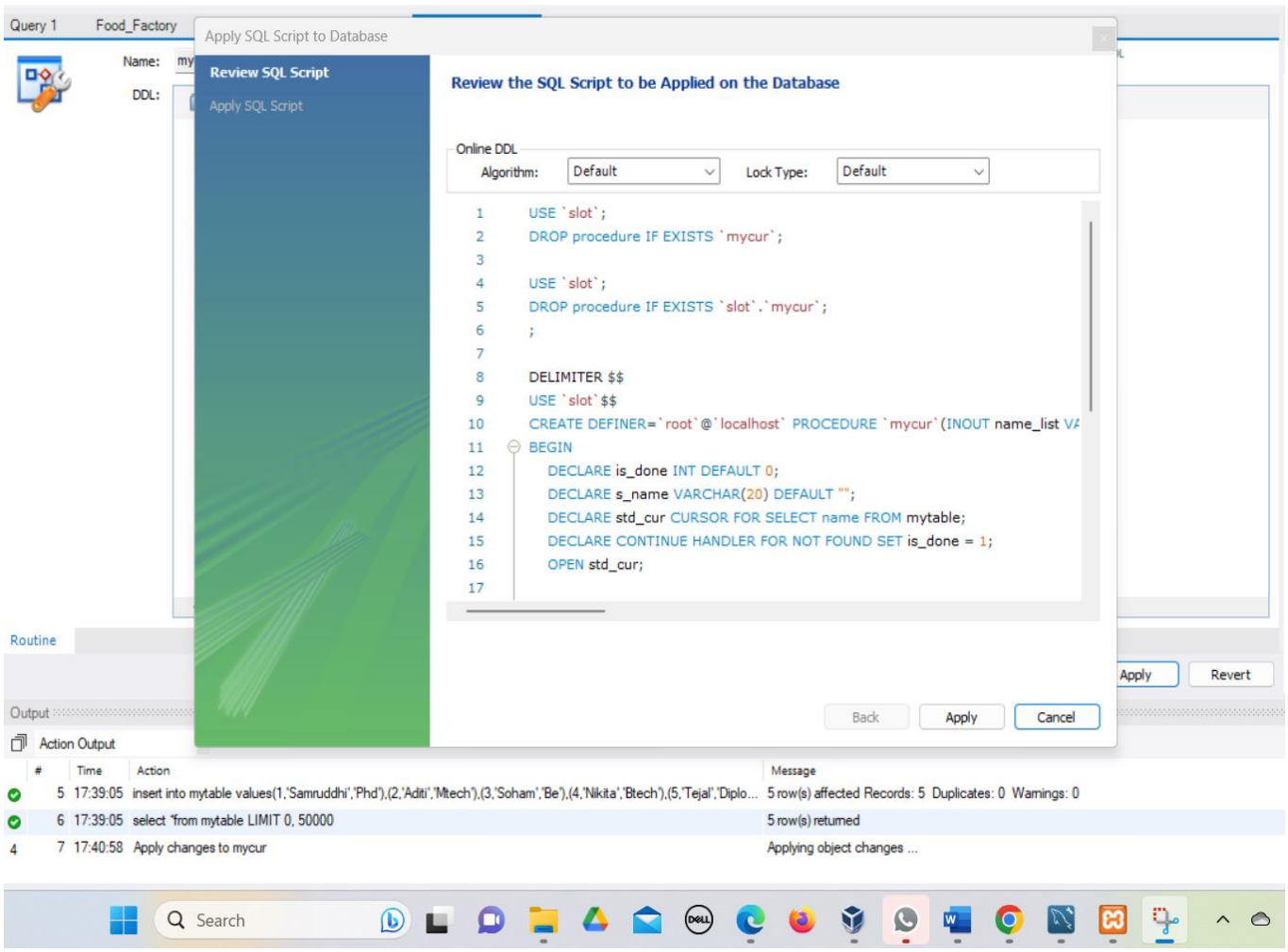
3. Write a PL/SQL block to implement all types of cursors.

```
create database slot;
use slot;
create table mytable(id int(2),name varchar(20),class varchar(20));
insert into mytable
values(1,'Samruddhi','Phd'),(2,'Aditi','Mtech'),(3,'Soham','Be'),(4,'Nikita','Btech'),(5,'Tejal','Diploma');
select *from mytable;
show tables;
drop database slot;

set @name_list = " ";
call mycur (@name_list);
select @name_list;
select *from mytable;
```



CREATING STORED PROCEDURE IN CURSOR:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `mycur`(INOUT name_list VARCHAR(100))
BEGIN
   DECLARE is_done INT DEFAULT 0;
   DECLARE s_name VARCHAR(20) DEFAULT "";
   DECLARE std_cur CURSOR FOR SELECT name FROM mytable;
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET is_done = 1;
   OPEN std_cur;

   getdata: LOOP
      FETCH std_cur INTO s_name;
      IF is_done = 1 THEN
         LEAVE getdata;
      END IF;
      SET name_list = CONCAT(s_name, " ", name_list);
   END LOOP;

   CLOSE std_cur;
END
```



Name: mycur

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```
1  CREATE DEFINER=`root`@`localhost` PROCEDURE `mycur`(INOUT name_list VARCHAR(100))
2  BEGIN
3      DECLARE is_done INT DEFAULT 0;
4      DECLARE s_name VARCHAR(20) DEFAULT "";
5      DECLARE std_cur CURSOR FOR SELECT name FROM mytable;
6      DECLARE CONTINUE HANDLER FOR NOT FOUND SET is_done = 1;
7      OPEN std_cur;
8
9      getdata: LOOP
10         FETCH std_cur INTO s_name;
11         IF is_done = 1 THEN
12             LEAVE getdata;
13         END IF;
14         SET name_list = CONCAT(s_name, " ", name_list);
15     END LOOP;
16
17     CLOSE std_cur;
18 END
```

Routine

Apply Revert

Output

Action Output

#	Time	Action	Message
5	17:39:05	insert into mytable values(1,'Samruddhi','Phd'),(2,'Aditi','Mtech'),(3,'Soham','Be'),(4,'Nikita','Btech'),(5,'Tejal','Diplo...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
6	17:39:05	select *from mytable LIMIT 0, 50000	5 row(s) returned
7	17:40:58	Apply changes to mycur	Changes applied

set @name_list = " ";
call mycur (@name_list);
select @name_list;

Query 1 | Food_Factory | SQL File 5* | SQL File 5* | mycur - Routine

Limit to 50000 rows

```
1  create database slot;
2  use slot;
3  create table mytable(id int(2),name varchar(20),class varchar(20));
4  show tables;
5  insert into mytable values(1,'Samruddhi','Phd'),(2,'Aditi','Mtech'),(3,'Soham','Be'),(4,'Nikita','Btech'),(5,'Tejal','Diploma
6  select *from mytable;
7  drop database slot;
8
9  set @name_list = " ";
10 call mycur (@name_list);
11 select @name_list;
12 select *from mytable;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

@name_list
Tejal Nikita Soham Aditi Samruddhi

Result 10

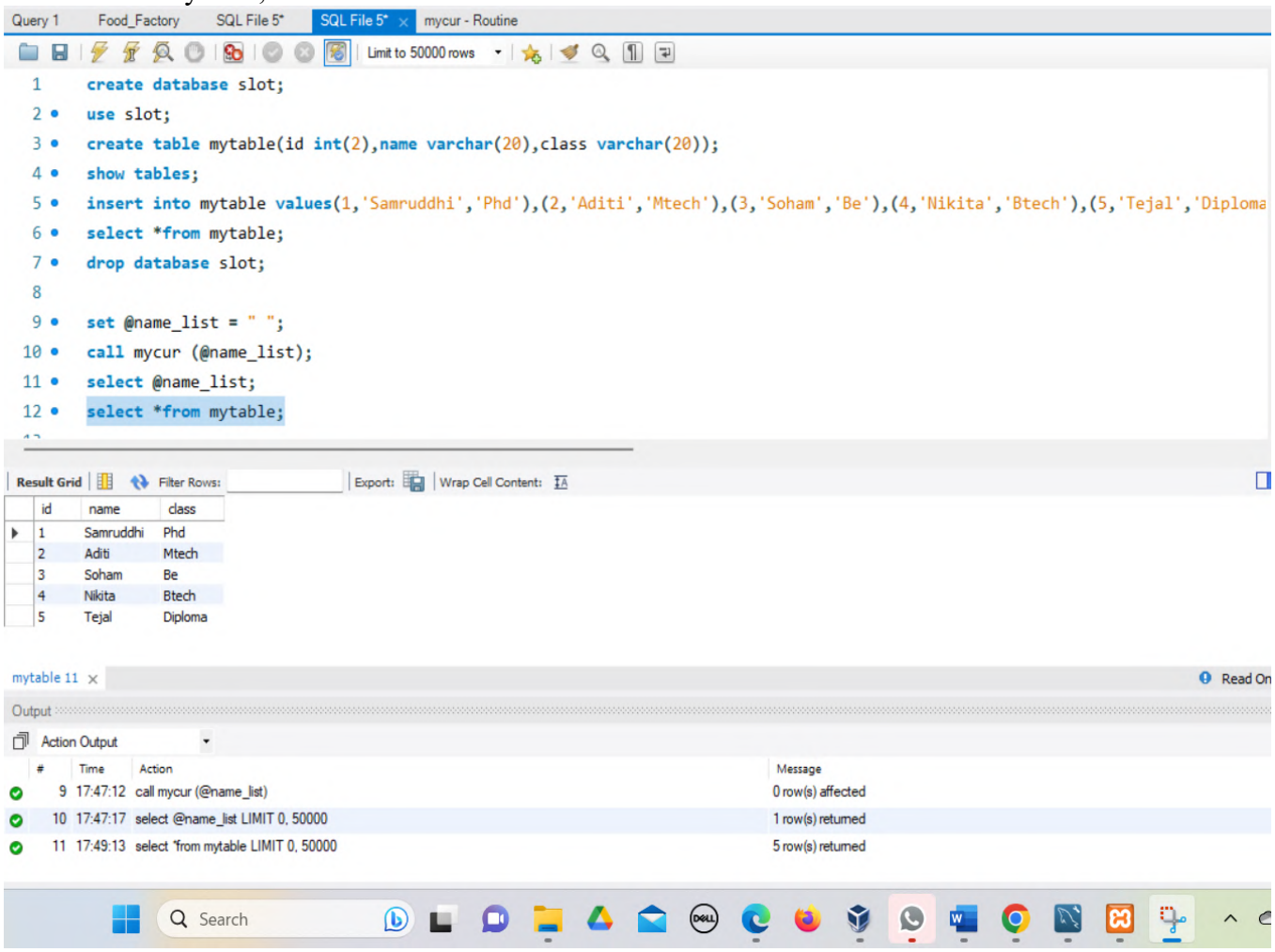
Read Onl

Output

Action Output

#	Time	Action	Message
8	17:47:07	set @name_list = " "	0 row(s) affected
9	17:47:12	call mycur (@name_list)	0 row(s) affected
10	17:47:17	select @name_list LIMIT 0, 50000	1 row(s) returned

select *from mytable;



Group D: Relational Database Design

1. Design and case study of any organization (back end only).

Food Factory Project(back-end)

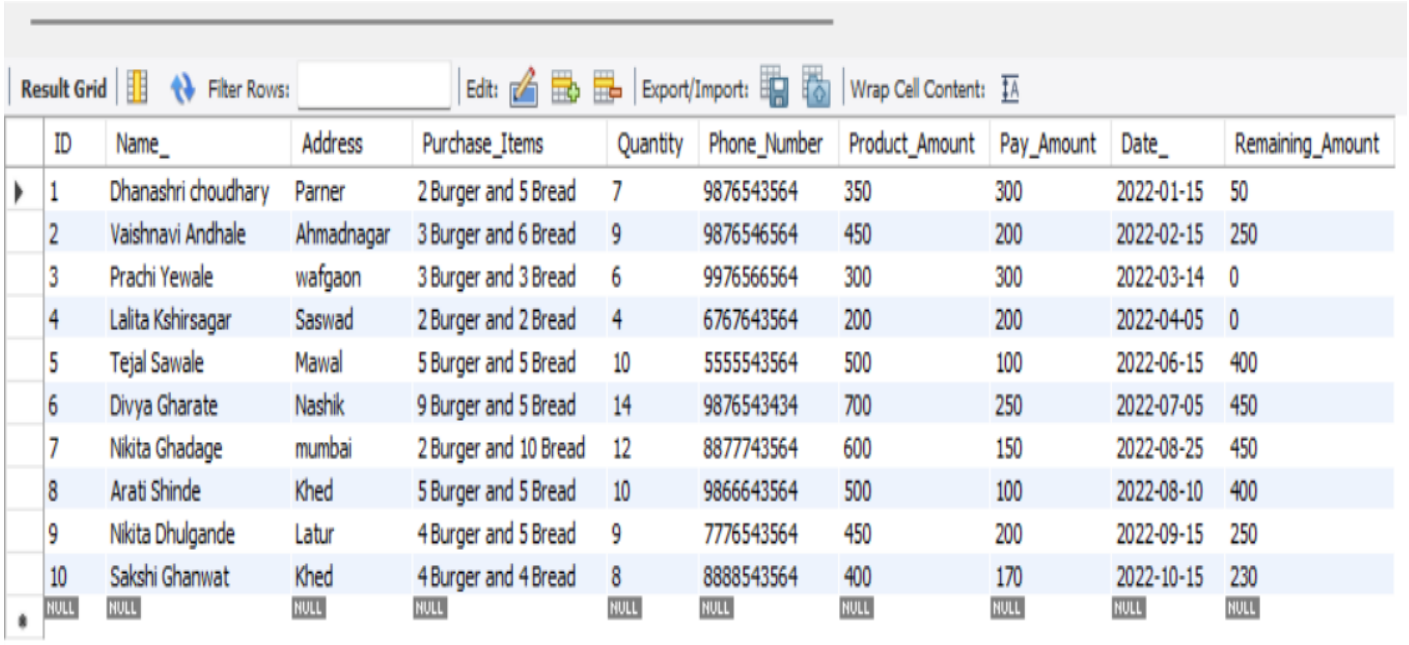
```
create database Food_Factory;
use Food_Factory;

-- EMPLOYEE TABLE
use Food_Factory;
create table Employee(
ID int not null primary key auto_increment primary key,Name_ varchar (50) not null,Address varchar (55) not null,Designation double not null,Salary double not null,DOJ date not null,Absents int not null,Received_Salary double not null);
show tables;
insert into Employee(Name_,Address,Designation,Salary,DOJ,Absents,Received_Salary)
values('Samruddhi Kangude','Kothrud',1,30500,'2022-10-04',5,((Employee.Designation*Employee.Salary)/(30))*(30-Employee.Absents)),
('Aditi Magar','Osmanabad',2,40500,'2022-02-04',1,((Employee.Designation*Employee.Salary)/(30))*(30-Employee.Absents)),
('Soham Pawar','Baramati',3,50500,'2022-02-02',2,((Employee.Designation*Employee.Salary)/(30))*(30-Employee.Absents)),
('Vaishnavi Gaikwad','Nashik',5,60500,'2022-10-07',3,((Employee.Designation*Employee.Salary)/(30))*(30-Employee.Absents)),
('Manthan Vaidya','Chakan',4,80500,'2022-06-10',4,((Employee.Designation*Employee.Salary)/(30))*(30-Employee.Absents)),
('Vikrant Kothimbire','Pune',10,70500,'2022-10-06',6,((Employee.Designation*Employee.Salary)/(30))*(30-Employee.Absents)),
('Vishal Narwade','Karve Nagar',8,90500,'2022-11-12',8,((Employee.Designation*Employee.Salary)/(30))*(30-Employee.Absents)),
('Siddharth choudhary','Mumbai',9,37500,'2022-12-11',7,((Employee.Designation*Employee.Salary)/(30))*(30-Employee.Absents)),
('Vaishnavi Panchal','Latur',6,77500,'2022-11-10',10,((Employee.Designation*Employee.Salary)/(30))*(30-Employee.Absents)),
('Nikita Nagwade','Parner',7,67500,'2022-10-11',9,((Employee.Designation*Employee.Salary)/(30))*(30-Employee.Absents));
select *from Employee;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ID	Name_	Address	Designation	Salary	DOJ	Absents	Received_Salary
1	Samruddhi Kangude	Kothrud	1	30500	2022-10-04	5	25416.666666666664
2	Aditi Magar	Osmanabad	2	40500	2022-02-04	1	78300
3	Soham Pawar	Baramati	3	50500	2022-02-02	2	141400
4	Vaishnavi Gaikwad	Nashik	5	60500	2022-10-07	3	272250
5	Manthan Vaidya	Chakan	4	80500	2022-06-10	4	279066.6666666667
6	Vikrant Kothimbire	Pune	10	70500	2022-10-06	6	564000
7	Vishal Narwade	Karve Nagar	8	90500	2022-11-12	8	530933.3333333333
8	Siddharth choudhary	Mumbai	9	37500	2022-12-11	7	258750
9	Vaishnavi Panchal	Latur	6	77500	2022-11-10	10	310000
10	Nikita Nagwade	Parner	7	67500	2022-10-11	9	330750
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- CUSTOMER TABLE

```
use Food_Factory;
create table Customer(
ID int not null auto_increment,Name_ varchar(50) not null,Address varchar(50) not null,Purchase_Items varchar(50),
Quantity int not null,Phone_Number varchar(50) not null,Product_Amount double not null,Pay_Amount double not
null,Date_ date,Remaining_Amount int,PRIMARY KEY (ID));
show tables;
insert                                                                                                    into
Customer(Name_,Address,Purchase_Items,Quantity,Phone_Number,Product_Amount,Pay_Amount,Date_,Remainin
g_Amount)
values('Dhanashri choudhary','Parner','2 Burger and 5 Bread',7,'9876543564',
Customer.Quantity*50,300,'2022-1-15',Customer.Product_Amount-Customer.Pay_Amount),
('Vaishnavi Andhale','Ahmadnagar','3 Burger and 6 Bread',9,'9876546564',
Customer.Quantity*50,200,'2022-2-15',Customer.Product_Amount-Customer.Pay_Amount),
('Prachi Yewale','wafgaon','3 Burger and 3 Bread',6,'9976566564',
Customer.Quantity*50,300,'2022-3-14',Customer.Product_Amount-Customer.Pay_Amount),
('Lalita Kshirsagar','Saswad','2 Burger and 2 Bread',4,'6767643564',
Customer.Quantity*50,200,'2022-4-5',Customer.Product_Amount-Customer.Pay_Amount),
('Tejal Sawale','Mawal','5 Burger and 5 Bread',10,'5555543564',
Customer.Quantity*50,100,'2022-6-15',Customer.Product_Amount-Customer.Pay_Amount),
('Divya Gharate','Nashik','9 Burger and 5 Bread',14,'9876543434',
Customer.Quantity*50,250,'2022-7-5',Customer.Product_Amount-Customer.Pay_Amount),
('Nikita Ghadage','mumbai','2 Burger and 10 Bread',12,'8877743564',
Customer.Quantity*50,150,'2022-8-25',Customer.Product_Amount-Customer.Pay_Amount),
('Arati Shinde','Khed','5 Burger and 5 Bread',10,'9866643564',
Customer.Quantity*50,100,'2022-8-10',Customer.Product_Amount-Customer.Pay_Amount),
('Nikita Dhulgande','Latur','4 Burger and 5 Bread',9,'7776543564',
Customer.Quantity*50,200,'2022-9-15',Customer.Product_Amount-Customer.Pay_Amount),
('Sakshi Ghanwat','Khed','4 Burger and 4 Bread',8,'8888543564',
Customer.Quantity*50,170,'2022-10-15',Customer.Product_Amount-Customer.Pay_Amount);
select *from Customer;
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ID	Name_	Address	Purchase_Items	Quantity	Phone_Number	Product_Amount	Pay_Amount	Date_	Remaining_Amount
1	Dhanashri choudhary	Parner	2 Burger and 5 Bread	7	9876543564	350	300	2022-01-15	50
2	Vaishnavi Andhale	Ahmadnagar	3 Burger and 6 Bread	9	9876546564	450	200	2022-02-15	250
3	Prachi Yewale	wafgaon	3 Burger and 3 Bread	6	9976566564	300	300	2022-03-14	0
4	Lalita Kshirsagar	Saswad	2 Burger and 2 Bread	4	6767643564	200	200	2022-04-05	0
5	Tejal Sawale	Mawal	5 Burger and 5 Bread	10	5555543564	500	100	2022-06-15	400
6	Divya Gharate	Nashik	9 Burger and 5 Bread	14	9876543434	700	250	2022-07-05	450
7	Nikita Ghadage	mumbai	2 Burger and 10 Bread	12	8877743564	600	150	2022-08-25	450
8	Arati Shinde	Khed	5 Burger and 5 Bread	10	9866643564	500	100	2022-08-10	400
9	Nikita Dhulgande	Latur	4 Burger and 5 Bread	9	7776543564	450	200	2022-09-15	250
10	Sakshi Ghanwat	Khed	4 Burger and 4 Bread	8	8888543564	400	170	2022-10-15	230
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- PURCHASE TABLE

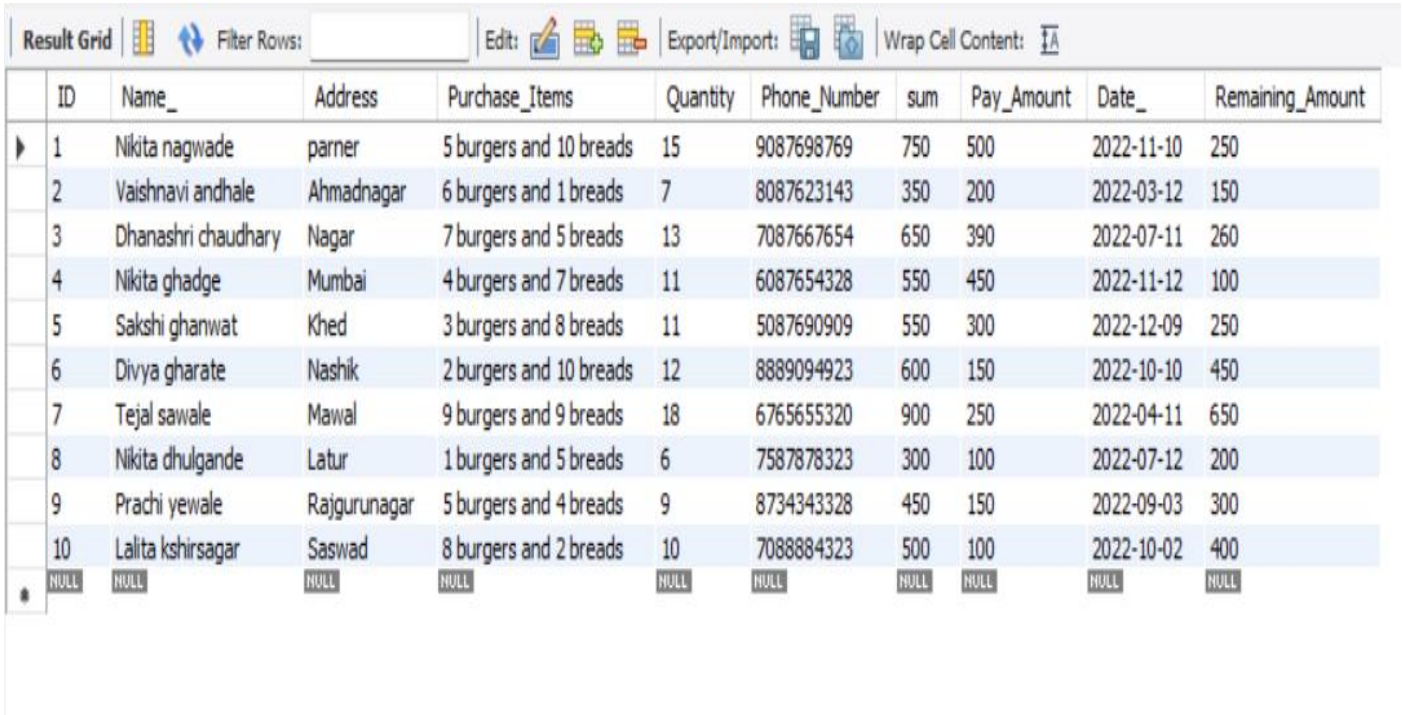
```
Use food_factory;
create table purchase (
Serial_No int not null auto_increment,Salad double,Chocolate double not null,Bread double not null,PawBhaji double not null,MilkShake double not null,Pizza double not null,Sandwitch double not null,Cake double,Noodles double,Pasta double,Dosa double,Ice_Cream double,Burger double,Veg_Biryani double,sum double,PRIMARY KEY(Serial_No));
show tables;
insert into purchase(Salad,Chocolate,Bread,PawBhaji,MilkShake,Pizza,Sandwitch,Cake,Noodles,Pasta,Dosa,Ice_Cream,Burger,Veg_Biryani,sum)
values(2,5,26,7,6,2,2,2,2,2,2,2,2,2,purchase.Salad+purchase.Chocolate+purchase.Bread+purchase.PawBhaji+purchase.MilkShake+purchase.Pizza+purchase.Sandwitch+purchase.Cake+purchase.Noodles+purchase.Pasta+purchase.Dosa+purchase.Ice_Cream+purchase.Burger+purchase.Veg_Biryani),
(3,5,25,7,7,2,4,5,6,2,7,2,9,8,purchase.Salad+purchase.Chocolate+purchase.Bread+purchase.PawBhaji+purchase.MilkShake+purchase.Pizza+purchase.Sandwitch+purchase.Cake+purchase.Noodles+purchase.Pasta+purchase.Dosa+purchase.Ice_Cream+purchase.Burger+purchase.Veg_Biryani),
(4,5,16,4,6,3,4,4,6,8,9,2,7,2,purchase.Salad+purchase.Chocolate+purchase.Bread+purchase.PawBhaji+purchase.MilkShake+purchase.Pizza+purchase.Sandwitch+purchase.Cake+purchase.Noodles+purchase.Pasta+purchase.Dosa+purchase.Ice_Cream+purchase.Burger+purchase.Veg_Biryani),
(5,5,26,7,6,3,5,2,3,2,2,7,4,2,purchase.Salad+purchase.Chocolate+purchase.Bread+purchase.PawBhaji+purchase.MilkShake+purchase.Pizza+purchase.Sandwitch+purchase.Cake+purchase.Noodles+purchase.Pasta+purchase.Dosa+purchase.Ice_Cream+purchase.Burger+purchase.Veg_Biryani),
(3,5,7,5,6,2,2,2,2,7,4,7,2,3,purchase.Salad+purchase.Chocolate+purchase.Bread+purchase.PawBhaji+purchase.MilkShake+purchase.Pizza+purchase.Sandwitch+purchase.Cake+purchase.Noodles+purchase.Pasta+purchase.Dosa+purchase.Ice_Cream+purchase.Burger+purchase.Veg_Biryani),
(4,3,14,7,6,25,2,7,2,2,5,2,8,2,purchase.Salad+purchase.Chocolate+purchase.Bread+purchase.PawBhaji+purchase.MilkShake+purchase.Pizza+purchase.Sandwitch+purchase.Cake+purchase.Noodles+purchase.Pasta+purchase.Dosa+purchase.Ice_Cream+purchase.Burger+purchase.Veg_Biryani),
(2,5,6,7,6,6,7,5,2,9,2,2,7,2,purchase.Salad+purchase.Chocolate+purchase.Bread+purchase.PawBhaji+purchase.MilkShake+purchase.Pizza+purchase.Sandwitch+purchase.Cake+purchase.Noodles+purchase.Pasta+purchase.Dosa+purchase.Ice_Cream+purchase.Burger+purchase.Veg_Biryani),
(8,5,3,7,6,7,4,2,3,5,7,2,2,2,purchase.Salad+purchase.Chocolate+purchase.Bread+purchase.PawBhaji+purchase.MilkShake+purchase.Pizza+purchase.Sandwitch+purchase.Cake+purchase.Noodles+purchase.Pasta+purchase.Dosa+purchase.Ice_Cream+purchase.Burger+purchase.Veg_Biryani),
(9,5,6,3,6,2,3,2,8,2,9,5,2,2,purchase.Salad+purchase.Chocolate+purchase.Bread+purchase.PawBhaji+purchase.MilkShake+purchase.Pizza+purchase.Sandwitch+purchase.Cake+purchase.Noodles+purchase.Pasta+purchase.Dosa+purchase.Ice_Cream+purchase.Burger+purchase.Veg_Biryani),
(6,7,26,7,3,7,2,7,2,9,9,9,5,2,purchase.Salad+purchase.Chocolate+purchase.Bread+purchase.PawBhaji+purchase.MilkShake+purchase.Pizza+purchase.Sandwitch+purchase.Cake+purchase.Noodles+purchase.Pasta+purchase.Dosa+purchase.Ice_Cream+purchase.Burger+purchase.Veg_Biryani);
select *from purchase;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

Serial_No	Salad	Chocolate	Bread	PawBhaji	MilkShake	Pizza	Sandwitch	Cake	Noodles	Pasta	Dosa	Ice_Cream	Burger	Veg_Biryani	sum
1	2	5	26	7	6	2	2	2	2	2	2	2	2	2	64
2	3	5	25	7	7	2	4	5	6	2	7	2	9	8	92
3	4	5	16	4	6	3	4	4	6	8	9	2	7	2	80
4	5	5	26	7	6	3	5	2	3	2	2	7	4	2	79
5	3	5	7	5	6	2	2	2	2	7	4	7	2	3	57
6	4	3	14	7	6	25	2	7	2	2	5	2	8	2	89
7	2	5	6	7	6	6	7	5	2	9	2	2	7	2	68
8	8	5	3	7	6	7	4	2	3	5	7	2	2	2	63
9	9	5	6	3	6	2	3	2	8	2	9	5	2	2	64
10	6	7	26	7	3	7	2	7	2	9	9	9	5	2	101
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- SALE_MAN TABLE

```
use food_factory;
create table sale_man(ID int not null auto_increment,Name_ varchar (50)not null,Address varchar(50) not null,Purchase_Items varchar(50),Quantity int not null,Phone_Number varchar(50) not null,sum double not null,Pay_Amount double not null,Date_ date,Remaining_Amount int,PRIMARY KEY (ID));
show tables;
insert into sale_man(Name_,Address,Purchase_Items,Quantity,Phone_Number,sum,Pay_Amount,Date_,Remaining_Amount)
values('Nikita nagwade','parner','5 burgers and 10 breads',15,'9087698769',sale_man.Quantity*50,500,'2022-11-10',sale_man.sum-sale_man.Pay_Amount),
('Vaishnavi andhale','Ahmadnagar','6 burgers and 1 breads',7,'8087623143',sale_man.Quantity*50,200,'2022-03-12',sale_man.sum-sale_man.Pay_Amount),
('Dhanashri chaudhary','Nagar','7 burgers and 5 breads',13,'7087667654',sale_man.Quantity*50,390,'2022-07-11',sale_man.sum-sale_man.Pay_Amount),
('Nikita ghadge','Mumbai','4 burgers and 7 breads',11,'6087654328',sale_man.Quantity*50,450,'2022-11-12',sale_man.sum-sale_man.Pay_Amount),
('Sakshi ghanwat','Khed','3 burgers and 8 breads',11,'5087690909',sale_man.Quantity*50,300,'2022-12-09',sale_man.sum-sale_man.Pay_Amount),
('Divya gharate','Nashik','2 burgers and 10 breads',12,'8889094923',sale_man.Quantity*50,150,'2022-10-10',sale_man.sum-sale_man.Pay_Amount),
('Tejal sawale','Mawal','9 burgers and 9 breads',18,'6765655320',sale_man.Quantity*50,250,'2022-04-11',sale_man.sum-sale_man.Pay_Amount),
('Nikita dhulgande','Latur','1 burgers and 5 breads',6,'7587878323',sale_man.Quantity*50,100,'2022-07-12',sale_man.sum-sale_man.Pay_Amount),
('Prachi yewale','Rajgurunagar','5 burgers and 4 breads',9,'8734343328',sale_man.Quantity*50,150,'2022-09-03',sale_man.sum-sale_man.Pay_Amount),
('Lalita kshirsagar','Saswad','8 burgers and 2 breads',10,'7088884323',sale_man.Quantity*50,100,'2022-10-02',sale_man.sum-sale_man.Pay_Amount);
select *from sale_man;
```



Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

ID	Name_	Address	Purchase_Items	Quantity	Phone_Number	sum	Pay_Amount	Date_	Remaining_Amount
1	Nikita nagwade	parner	5 burgers and 10 breads	15	9087698769	750	500	2022-11-10	250
2	Vaishnavi andhale	Ahmadnagar	6 burgers and 1 breads	7	8087623143	350	200	2022-03-12	150
3	Dhanashri chaudhary	Nagar	7 burgers and 5 breads	13	7087667654	650	390	2022-07-11	260
4	Nikita ghadge	Mumbai	4 burgers and 7 breads	11	6087654328	550	450	2022-11-12	100
5	Sakshi ghanwat	Khed	3 burgers and 8 breads	11	5087690909	550	300	2022-12-09	250
6	Divya gharate	Nashik	2 burgers and 10 breads	12	8889094923	600	150	2022-10-10	450
7	Tejal sawale	Mawal	9 burgers and 9 breads	18	6765655320	900	250	2022-04-11	650
8	Nikita dhulgande	Latur	1 burgers and 5 breads	6	7587878323	300	100	2022-07-12	200
9	Prachi yewale	Rajgurunagar	5 burgers and 4 breads	9	8734343328	450	150	2022-09-03	300
10	Lalita kshirsagar	Saswad	8 burgers and 2 breads	10	7088884323	500	100	2022-10-02	400
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

-- SALARIES TABLE

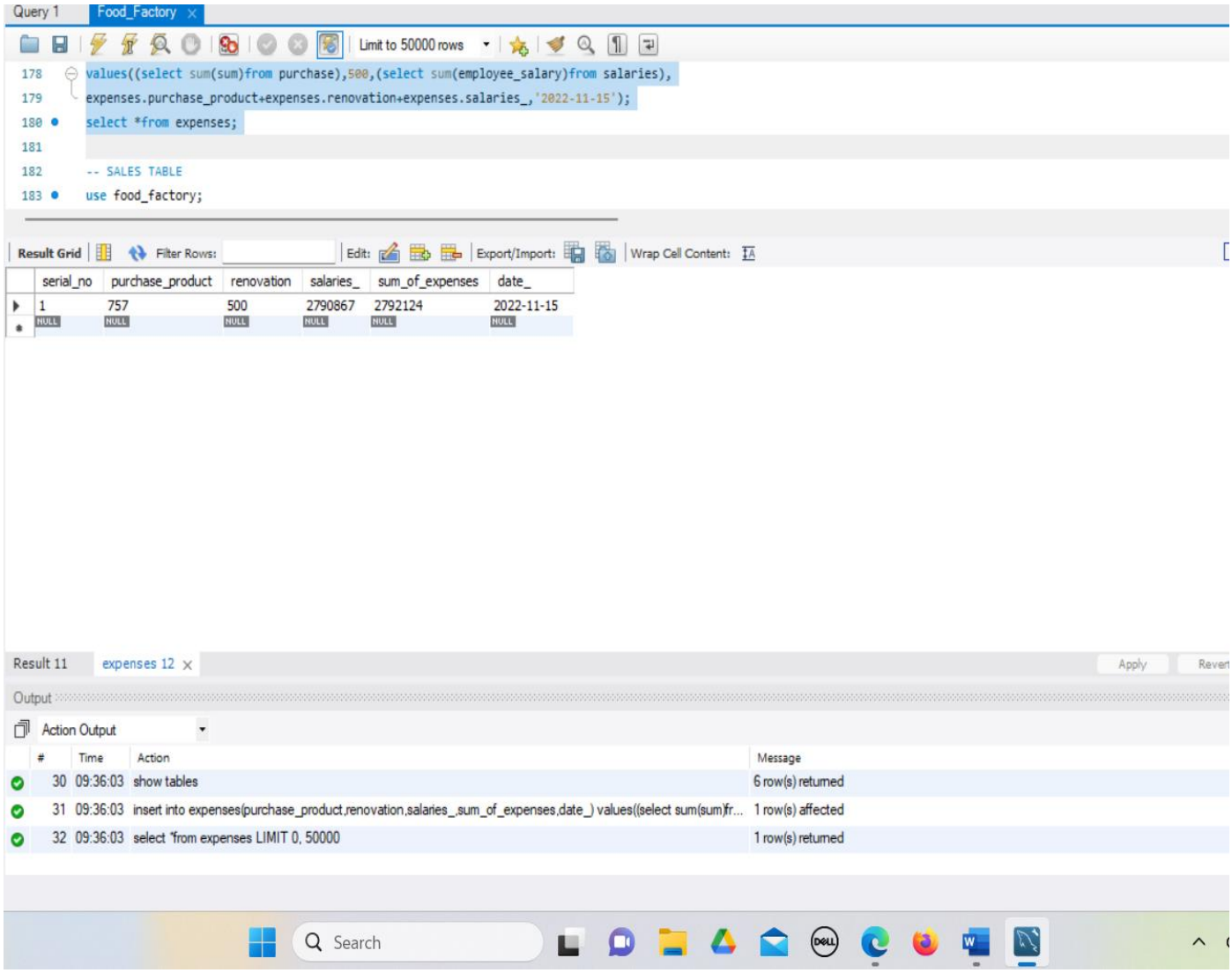
```
use food_factory;
create table salaries(serial_no int auto_increment primary key,employee_id int,employee_name varchar(40),employee_salary int,foreign key(employee_id) references employee (id));
show tables;
insert into salaries(employee_id,employee_name,employee_salary)
values(1,(select name_ from employee where id=1),
(select received_salary from employee where id=1)),
(2,(select name_ from employee where id=2),
(select received_salary from employee where id=2)),
(3,(select name_ from employee where id=3),
(select received_salary from employee where id=3)),
(4,(select name_ from employee where id=4),
(select received_salary from employee where id=4)),
(5,(select name_ from employee where id=5),
(select received_salary from employee where id=5)),
(6,(select name_ from employee where id=6),
(select received_salary from employee where id=6)),
(7,(select name_ from employee where id=7),
(select received_salary from employee where id=7)),
(8,(select name_ from employee where id=8),
(select received_salary from employee where id=8)),
(9,(select name_ from employee where id=9),
(select received_salary from employee where id=9)),
(10,(select name_ from employee where id=10),
(select received_salary from employee where id=10));
select *from salaries;
```

Result Grid | Filter Rows: | Edit:

serial_no	employee_id	employee_name	employee_salary
1	1	Samruddhi Kangude	25417
2	2	Aditi Magar	78300
3	3	Soham Pawar	141400
4	4	Vaishnavi Gaikwad	272250
5	5	Manthan Vaidya	279067
6	6	Vikrant Kothimbire	564000
7	7	Vishal Narwade	530933
8	8	Siddharth choudhary	258750
9	9	Vaishnavi Panchal	310000
10	10	Nikita Nagwade	330750
NULL	NULL	NULL	NULL

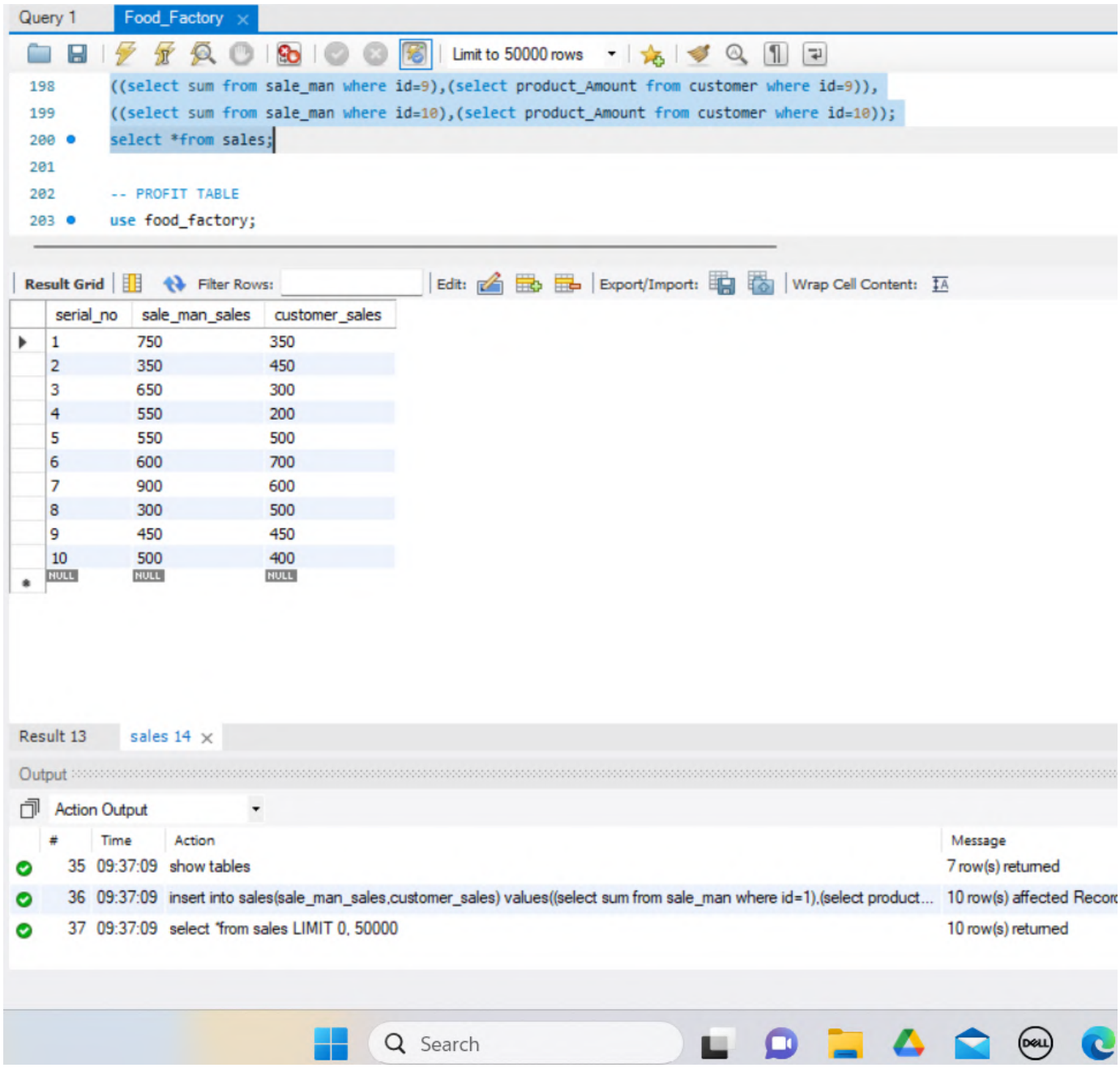
-- EXPENSES TABLE

```
use food_factory;
create table expenses(
serial_no int primary key auto_increment,
purchase_product double,
renovation double,
salaries_ double,
sum_of_expenses double,
date_ date
);
show tables;
insert into expenses(purchase_product,renovation,salaries_,sum_of_expenses,date_)
values((select sum(sum)from purchase),500,(select sum(employee_salary)from salaries),
expenses.purchase_product+expenses.renovation+expenses.salaries_,'2022-11-15');
select *from expenses;
```



-- SALES TABLE

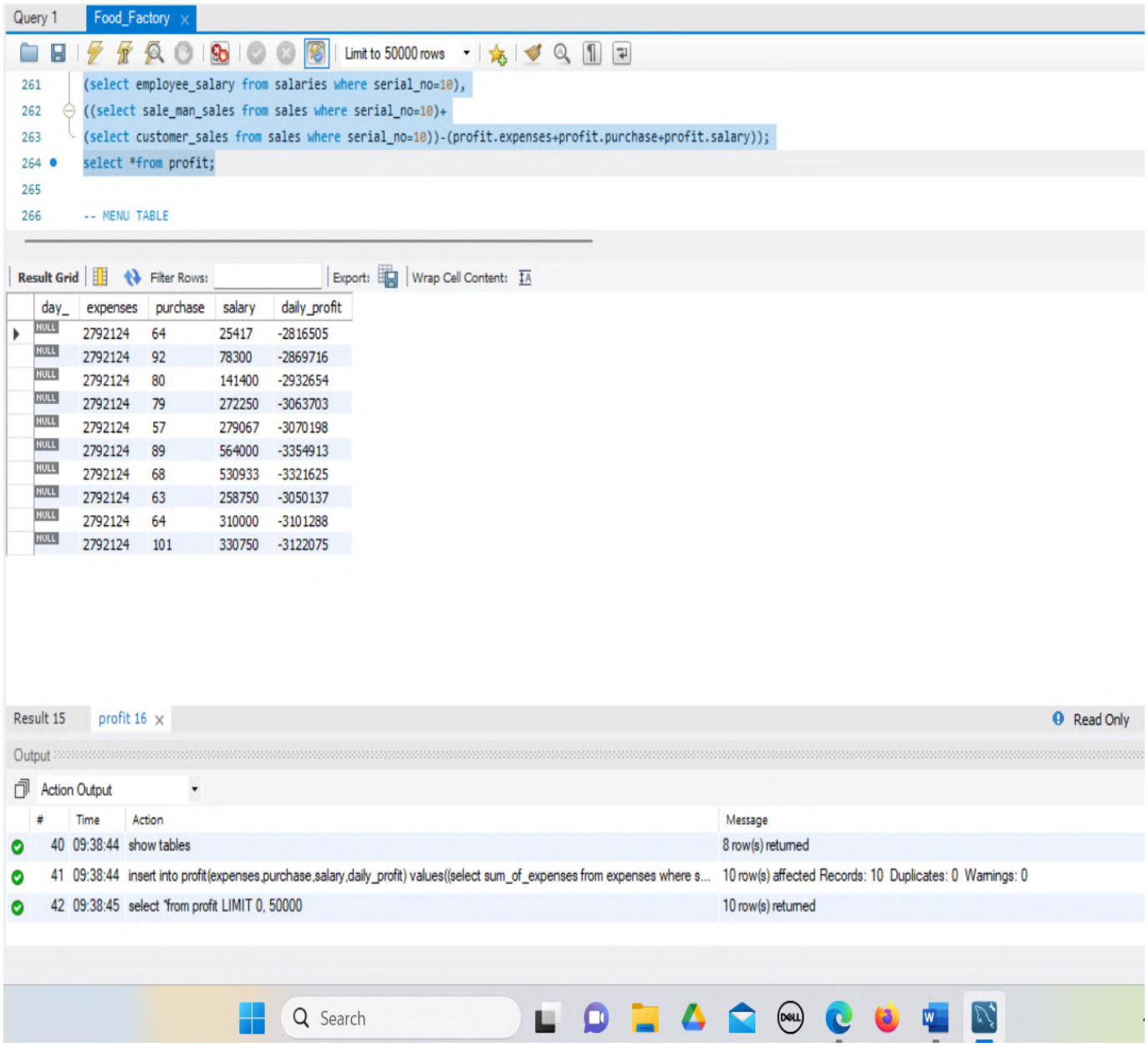
```
use food_factory;
create table sales(
serial_no int auto_increment primary key,
sale_man_sales int,
customer_sales int);
show tables;
insert into sales(sale_man_sales,customer_sales)
values((select sum from sale_man where id=1),(select product_Amount from customer where id=1)),
((select sum from sale_man where id=2),(select product_Amount from customer where id=2)),
((select sum from sale_man where id=3),(select product_Amount from customer where id=3)),
((select sum from sale_man where id=4),(select product_Amount from customer where id=4)),
((select sum from sale_man where id=5),(select product_Amount from customer where id=5)),
((select sum from sale_man where id=6),(select product_Amount from customer where id=6)),
((select sum from sale_man where id=7),(select product_Amount from customer where id=7)),
((select sum from sale_man where id=8),(select product_Amount from customer where id=8)),
((select sum from sale_man where id=9),(select product_Amount from customer where id=9)),
((select sum from sale_man where id=10),(select product_Amount from customer where id=10));
select *from sales;
```



-- PROFIT TABLE

```
use food_factory;
create table profit(
day_ int,
expenses double,
purchase double,
salary double,
daily_profit int,
foreign key(day_) references expenses (serial_no)
);
show tables;
insert into profit(expenses,purchase,salary,daily_profit)
values((select sum_of_expenses from expenses where serial_no=1),
(select sum from purchase where serial_no=1),
(select employee_salary from salaries where serial_no=1),
((select sale_man_sales from sales where serial_no=1)+
(select customer_sales from sales where serial_no=1))-(profit.expenses+profit.purchase+profit.salary)),
((select sum_of_expenses from expenses where serial_no=1),
(select sum from purchase where serial_no=2),
(select employee_salary from salaries where serial_no=2),
((select sale_man_sales from sales where serial_no=2)+
(select customer_sales from sales where serial_no=2))-(profit.expenses+profit.purchase+profit.salary)),
((select sum_of_expenses from expenses where serial_no=1),
(select sum from purchase where serial_no=3),
(select employee_salary from salaries where serial_no=3),
((select sale_man_sales from sales where serial_no=3)+
(select customer_sales from sales where serial_no=3))-(profit.expenses+profit.purchase+profit.salary)),
((select sum_of_expenses from expenses where serial_no=1),
(select sum from purchase where serial_no=4),
(select employee_salary from salaries where serial_no=4),
((select sale_man_sales from sales where serial_no=4)+
(select customer_sales from sales where serial_no=4))-(profit.expenses+profit.purchase+profit.salary)),
((select sum_of_expenses from expenses where serial_no=1),
(select sum from purchase where serial_no=5),
(select employee_salary from salaries where serial_no=5),
((select sale_man_sales from sales where serial_no=5)+
(select customer_sales from sales where serial_no=5))-(profit.expenses+profit.purchase+profit.salary)),
((select sum_of_expenses from expenses where serial_no=1),
(select sum from purchase where serial_no=6),
(select employee_salary from salaries where serial_no=6),
((select sale_man_sales from sales where serial_no=6)+
(select customer_sales from sales where serial_no=6))-(profit.expenses+profit.purchase+profit.salary)),
((select sum_of_expenses from expenses where serial_no=1),
(select sum from purchase where serial_no=7),
(select employee_salary from salaries where serial_no=7),
((select sale_man_sales from sales where serial_no=7)+
(select customer_sales from sales where serial_no=7))-(profit.expenses+profit.purchase+profit.salary)),
((select sum_of_expenses from expenses where serial_no=1),
(select sum from purchase where serial_no=8),
(select employee_salary from salaries where serial_no=8),
((select sale_man_sales from sales where serial_no=8)+
(select customer_sales from sales where serial_no=8))-(profit.expenses+profit.purchase+profit.salary)),
((select sum_of_expenses from expenses where serial_no=1),
(select sum from purchase where serial_no=9),
(select employee_salary from salaries where serial_no=9),
((select sale_man_sales from sales where serial_no=9)+
(select customer_sales from sales where serial_no=9))-(profit.expenses+profit.purchase+profit.salary)),
((select sum_of_expenses from expenses where serial_no=1),
(select sum from purchase where serial_no=10),
```

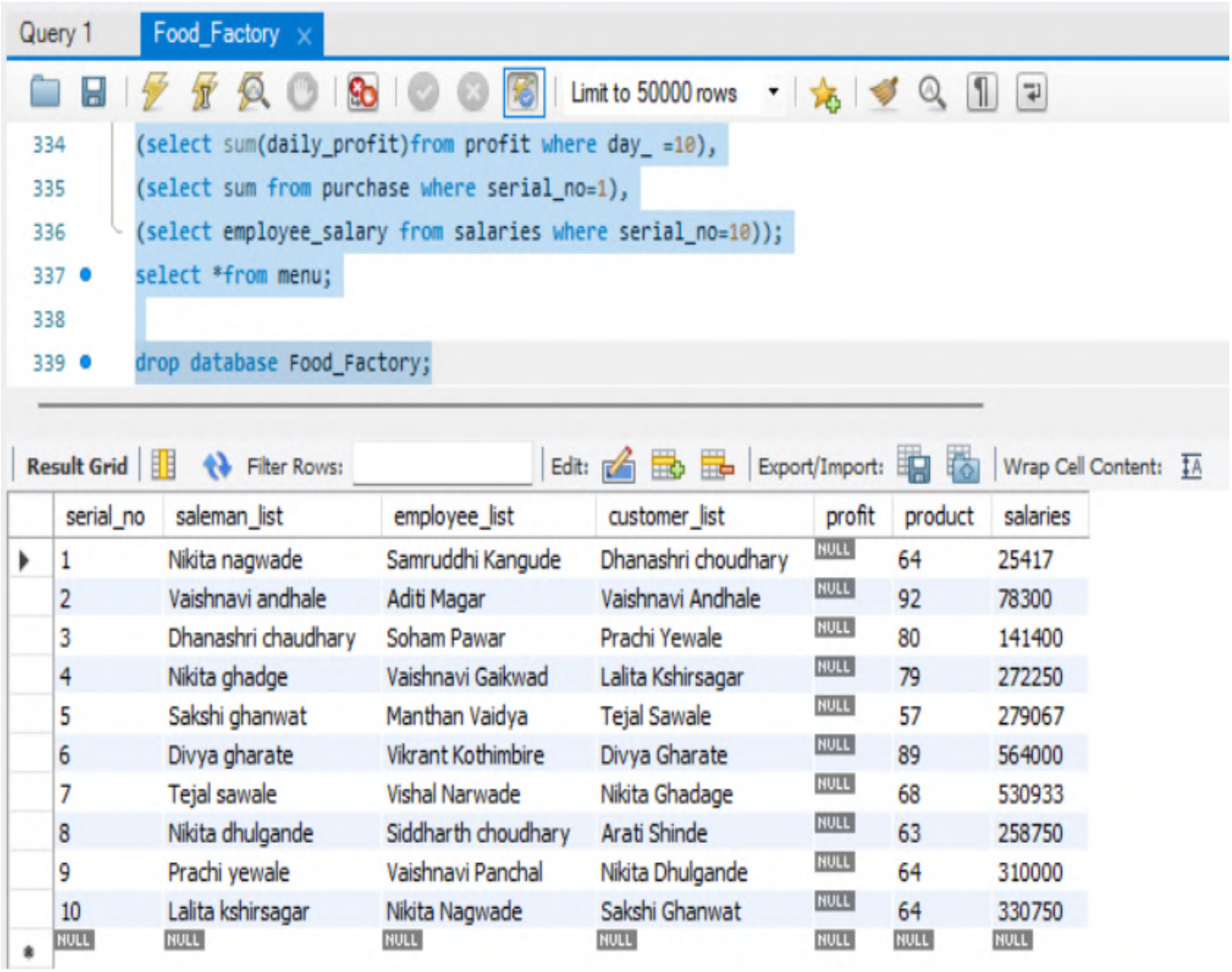
(select employee_salary from salaries where serial_no=10),
((select sale_man_sales from sales where serial_no=10)+
(select customer_sales from sales where serial_no=10))-(profit.expenses+profit.purchase+profit.salary));
select *from profit;



-- MENU TABLE

```
use food_factory;
create table menu(
serial_no int auto_increment primary key,
saleman_list varchar(50),
employee_list varchar(50),
customer_list varchar(50),
profit double,
product double not null,
salaries double not null);
insert into menu(saleman_list,employee_list,customer_list,profit,product,salaries)
values((select name_ from sale_man where id=1),
(select name_ from employee where id=1),
(select name_ from customer where id=1),
(select sum(daily_profit)from profit where day_ =1),
(select sum from purchase where serial_no=1),
(select employee_salary from salaries where serial_no=1)),
((select name_ from sale_man where id=2),
(select name_ from employee where id=2),
(select name_ from customer where id=2),
(select sum(daily_profit)from profit where day_ =1),
(select sum from purchase where serial_no=2),
(select employee_salary from salaries where serial_no=2)),
((select name_ from sale_man where id=3),
(select name_ from employee where id=3),
(select name_ from customer where id=3),
(select sum(daily_profit)from profit where day_ =1),
(select sum from purchase where serial_no=3),
(select employee_salary from salaries where serial_no=3)),
((select name_ from sale_man where id=4),
(select name_ from employee where id=4),
(select name_ from customer where id=4),
(select sum(daily_profit)from profit where day_ =1),
(select sum from purchase where serial_no=4),
(select employee_salary from salaries where serial_no=4)),
((select name_ from sale_man where id=5),
(select name_ from employee where id=5),
(select name_ from customer where id=5),
(select sum(daily_profit)from profit where day_ =1),
(select sum from purchase where serial_no=5),
(select employee_salary from salaries where serial_no=5)),
((select name_ from sale_man where id=6),
(select name_ from employee where id=6),
(select name_ from customer where id=6),
(select sum(daily_profit)from profit where day_ =1),
(select sum from purchase where serial_no=6),
(select employee_salary from salaries where serial_no=6)),
((select name_ from sale_man where id=7),
(select name_ from employee where id=7),
(select name_ from customer where id=7),
(select sum(daily_profit)from profit where day_ =1),
(select sum from purchase where serial_no=7),
(select employee_salary from salaries where serial_no=7)),
((select name_ from sale_man where id=8),
(select name_ from employee where id=8),
(select name_ from customer where id=8),
(select sum(daily_profit)from profit where day_ =1),
(select sum from purchase where serial_no=8),
(select employee_salary from salaries where serial_no=8)),
```

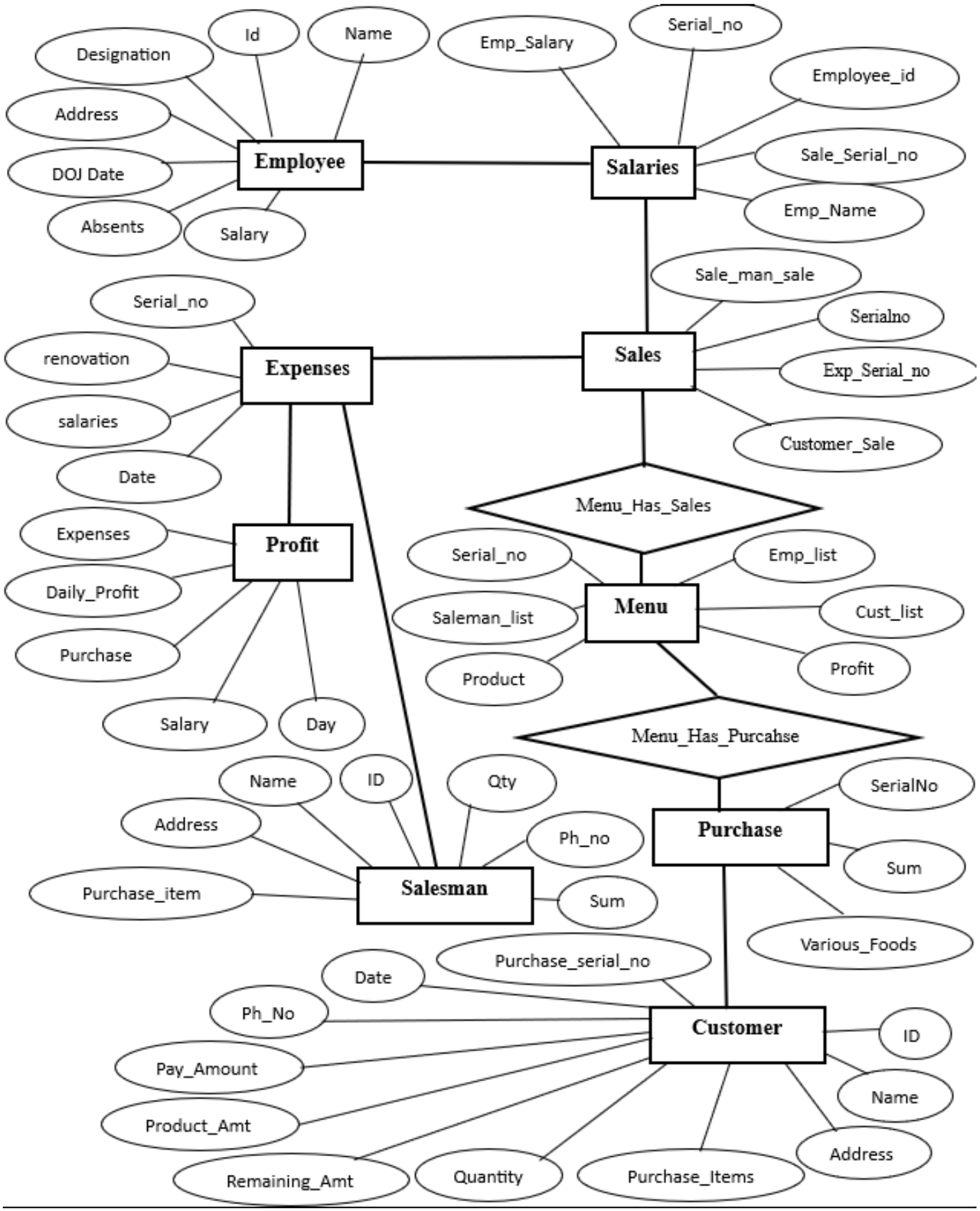
```
((select name_ from sale_man where id=9),
(select name_ from employee where id=9),
(select name_ from customer where id=9),
(select sum(daily_profit)from profit where day_ =1),
(select sum from purchase where serial_no=9),
(select employee_salary from salaries where serial_no=9)),
((select name_ from sale_man where id=10),
(select name_ from employee where id=10),
(select name_ from customer where id=10),
(select sum(daily_profit)from profit where day_ =10),
(select sum from purchase where serial_no=1),
(select employee_salary from salaries where serial_no=10));
select *from menu;

drop database Food_Factory;
```



serial_no	saleman_list	employee_list	customer_list	profit	product	salaries
1	Nikita nagwade	Samruddhi Kangude	Dhanashri choudhary	NULL	64	25417
2	Vaishnavi andhale	Aditi Magar	Vaishnavi Andhale	NULL	92	78300
3	Dhanashri chaudhary	Soham Pawar	Prachi Yewale	NULL	80	141400
4	Nikita ghadge	Vaishnavi Gaikwad	Lalita Kshirsagar	NULL	79	272250
5	Sakshi ghanwat	Manthan Vaidya	Tejal Sawale	NULL	57	279067
6	Divya gharate	Vikrant Kothimbire	Divya Gharate	NULL	89	564000
7	Tejal sawale	Vishal Narwade	Nikita Ghadage	NULL	68	530933
8	Nikita dhulgande	Siddharth choudhary	Arati Shinde	NULL	63	258750
9	Prachi yewale	Vaishnavi Panchal	Nikita Dhulgande	NULL	64	310000
10	Lalita kshirsagar	Nikita Nagwade	Sakshi Ghanwat	NULL	64	330750
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Group D: Relational Database Design

Project ER Diagram and Database Design
For ER diagram and Database design following guidelines can be used:
Draw an ER diagram of your project.



2. Reduce this ER diagram into the tables and complete database design.
3. Subsequently, list all the functional dependencies on each table that you expect will hold.
4. Check that the database schema is in 3NF/BCNF. If it is not, apply normalization. Use non loss decomposition and bring the database schema in 3NF/BCNF.
Give the ER diagram and the data dictionary as part of the requirement specifications file which you created for the project proposal.