西北工業大學

NORTHWESTERN POLYTECHNICAL UNIVERSITY

Lab report

Name : ABID ALI
Student_No. : 2019380141

Experiment 6

Experiment No:6

ODBC/JDBC

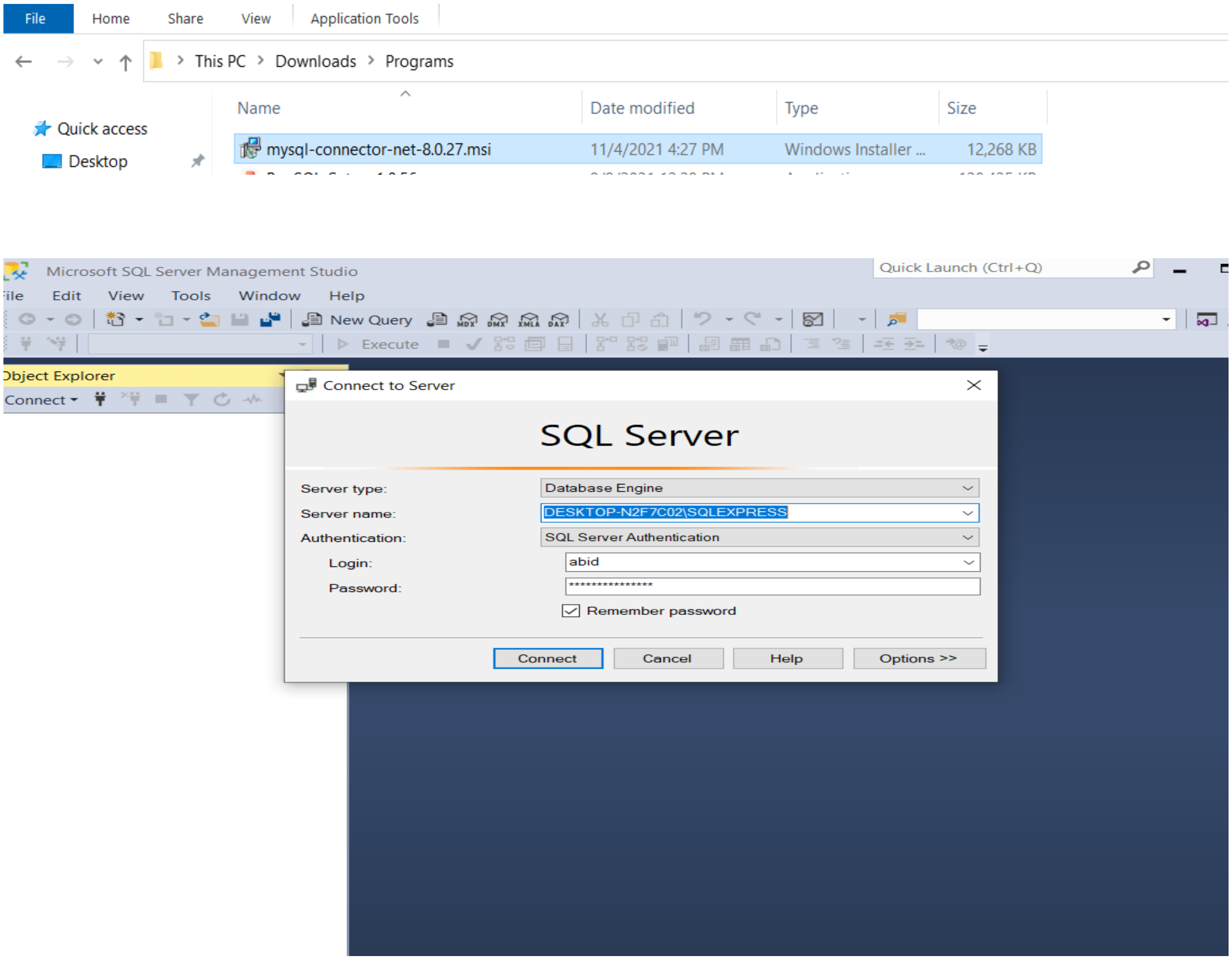
Goal:

To practice how to access the database from applications with ODBC/JDBC or other methods.

Content:

1. ODBC data source configuration and program debugging. (20 points)

(1) To configure an ODBC data source, the data source name is: student, which contains the S table (student information table).



This PC > Downloads > Programs

Name	Date modified	Type	Size
mysql-connector-net-8.0.27.msi	11/4/2021 4:27 PM	Windows Installer ...	12,268 KB

Server type: Database Engine

Server name: DESKTOP-N2F7C02\SQLEXPRESS

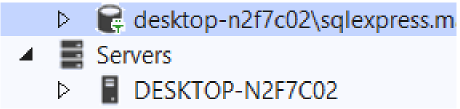
Authentication: SQL Server Authentication

Login: abid

Password: ***************

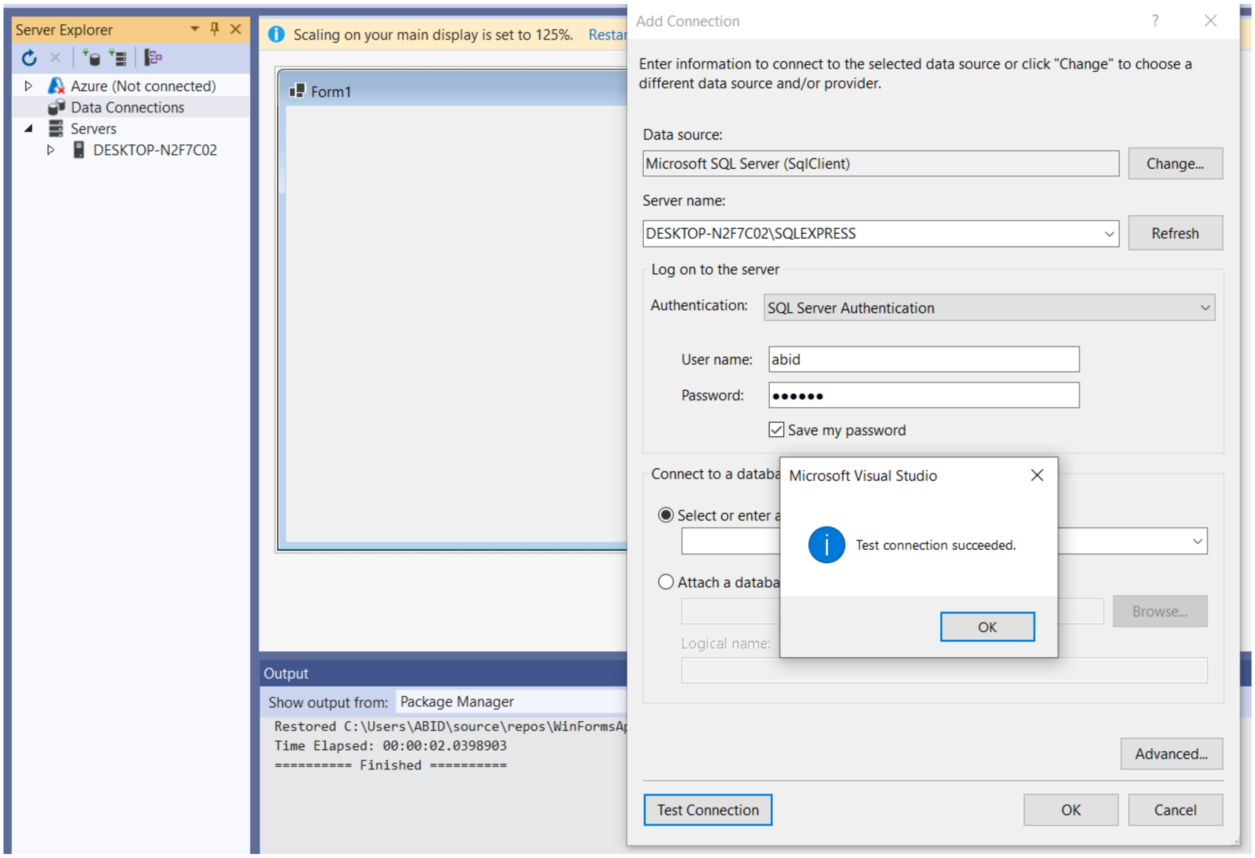
☑ Remember password

Fig: Creating SQL server



```
▷ desktop-n2f7c02\sqlexpress.m
◢ Servers
  ▷ DESKTOP-N2F7C02
```

Fig: Connected with server



Scaling on your main display is set to 125%. Restar

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source: Microsoft SQL Server (SqlClient)

Server name: DESKTOP-N2F7C02\SQLEXPRESS

Log on to the server

Authentication: SQL Server Authentication

User name: abid

Password: ••••••

☑ Save my password

Connect to a databa

● Select or enter a

○ Attach a databa

Logical name:

Microsoft Visual Studio

Test connection succeeded.

OK

Show output from: Package Manager

```
Restored C:\Users\ABID\source\repos\WinFormsAp
Time Elapsed: 00:00:02.0398903
========== Finished ==========
```

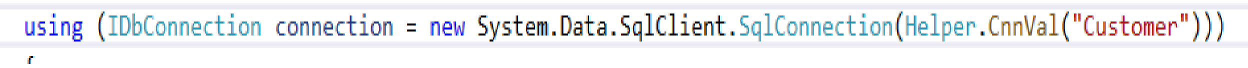
Fig: Testing the connection

(2) To understand ODBC programming, read and run the program example (MFC or CSharp code) given in the experiment, it is required to write your own understanding of the program, or draw a flow chart of the program, and give the screenshot of the program running results.

Solution:

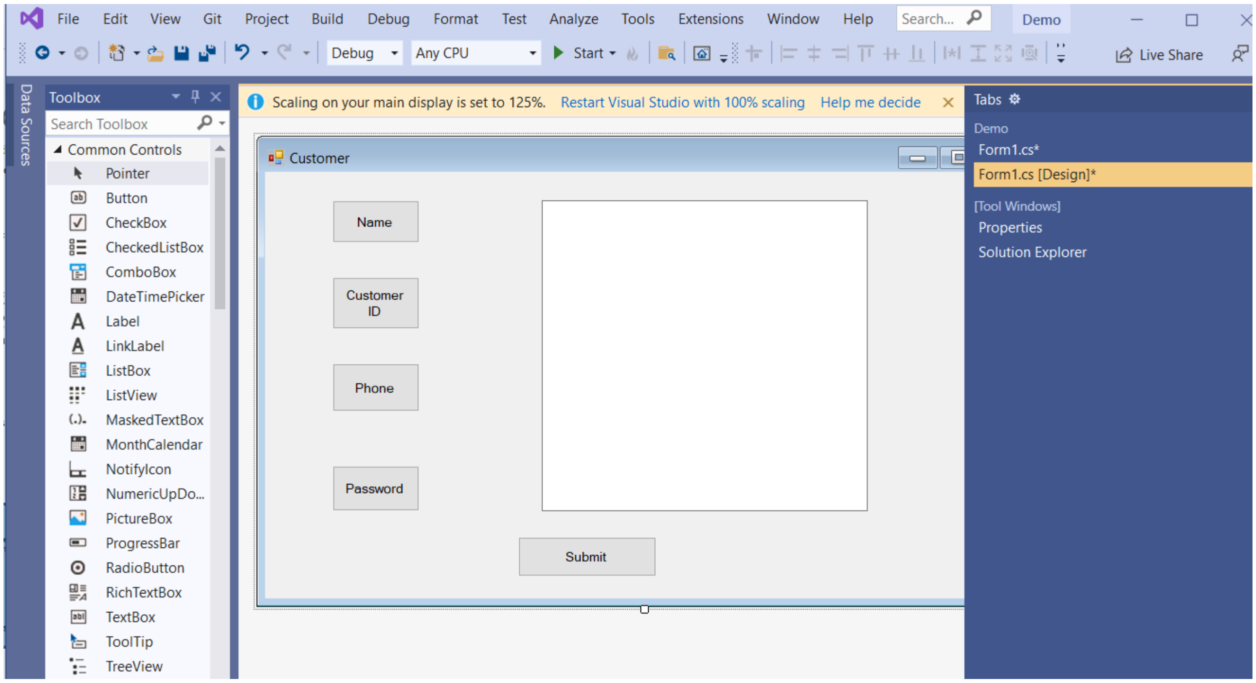
SqlConnection con = new SqlConnection@"Data Source=DESKTOP-N2F7C02\SQLEXPRESS;Persist Security Info=True; User ID=abid;Password=***********");

or



```
using (IDbConnection connection = new System.Data.SqlClient.SqlConnection(Helper.CnnVal("Customer")))
```

This is how to connect by using ODBC .



Scaling on your main display is set to 125%.

We can see that,we have made a GUI using C# using odbc here we can input the values it get inserted in mysql workbench.

Inserting Values in tables:

Code:

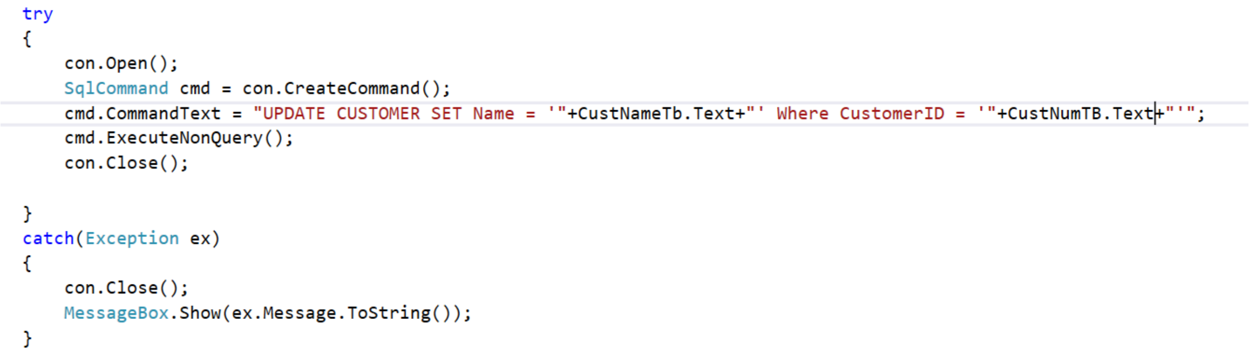
```
private void AddCustomerButton_Click(object sender, EventArgs e)
{
    try
    {
        con.Open();
        SqlCommand cmd = con.CreateCommand();
        cmd.CommandText = "INSERT INTO CUSTOMER VALUES(" + int.Parse(CustNumTB.Text) + ", '"
            + CustNameTb.Text + "', '" + PhoneTB.Text
            + "', '" + IDCardTb.Text + "', '" + PasswordTb.Text + "')";
        cmd.ExecuteNonQuery();
        con.Close();

    }
    catch(Exception ex)
    {
        con.Close();
        MessageBox.Show(ex.Message.ToString());
    }
}
```

Fig: Inserting values in table

```
{
      try
        {
            con.open( ) ;
            SqlCommand  cmd  = con.CreateCommand ( ) ;
            cmd.CommandText  = “ INSERT INTO CUSTOMER VALUES (“ + int.Parse(CustNumTB.
Text) + “  ”
                  + CustNameTb.Text + “  ‘  ,  ’  ” + PhoneTB.Text
                  +  “  ‘  ,  ’ ” + IDCardTb. Text +  “  ‘  ,  ’ ” + PasswordTb. Text +  “ ‘)” ;
         cmd. ExecuteNonQuery ( ) ;                                                    con.
         Close ( ) ;

        }
       catch( Exception  ex)
      {
            con.Close ( ) ;
            MessageBox. Show( ex.Message. ToString ( ) ) ;
      }
}
```

Updating values in the table:

Code:



```
try
{
    con.Open();
    SqlCommand cmd = con.CreateCommand();
    cmd.CommandText = "UPDATE CUSTOMER SET Name = '"+CustNameTb.Text+"' Where CustomerID = '"+CustNumTB.Text+"'";
    cmd.ExecuteNonQuery();
    con.Close();

}
catch(Exception ex)
{
    con.Close();
    MessageBox.Show(ex.Message.ToString());
}
```

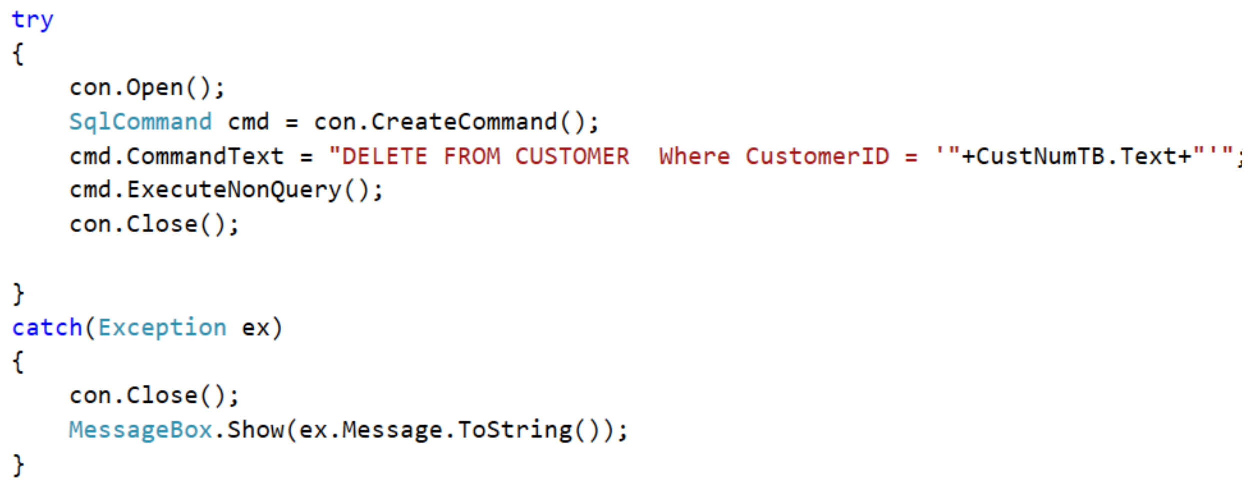
Fig: Updating values in table

```
try
 {
     Con. Open ( ) ;
     SqlCommand  cmd  = con. CreateCommand( ) ;
```

```
        cmd. CommandText = “ DELETE  FROM  CUSTOMER  Where Customer  ID = ’
” +  CustNumTB. Text + “ ‘  ” ;
        cmd. ExecuteNonQuery ( ) ;
        con. Close( ) ;


}
catch ( Exception  ex )
{
          con.Close ( ) ;
          MessageBox . Show(ex. Message . ToString (  ) ) ;
}
```

Deleting values in the table:

Code:



```
try
{
    con.Open();
    SqlCommand cmd = con.CreateCommand();
    cmd.CommandText = "DELETE FROM CUSTOMER  Where CustomerID = '"+CustNumTB.Text+"'";
    cmd.ExecuteNonQuery();
    con.Close();

}
catch(Exception ex)
{
    con.Close();
    MessageBox.Show(ex.Message.ToString());
}
```

Fig: Deleting values in table

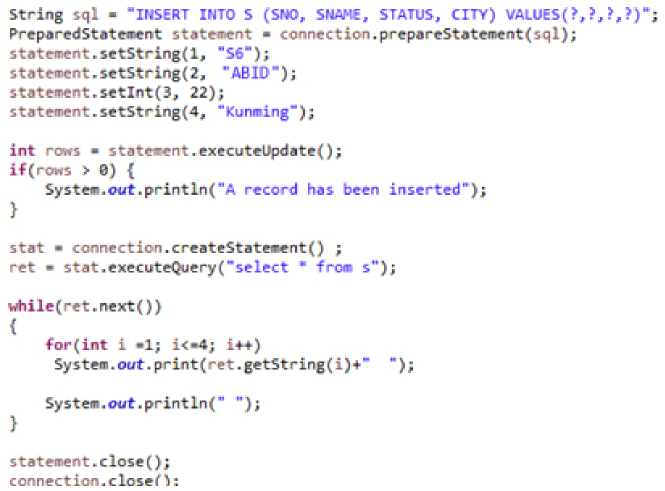
This is the general Structure of ODBC my using C sharp Language according to my understanding.

2. Refer to the above ODBC example program, use ODBC programming technology, write a simple program, including the database SPJ_ MNG connection, query, insert, modify and delete. (30 points)

Solution:

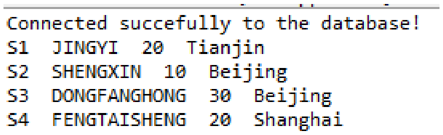
I am going to use to use JDBC for this part because I am familiar with Java language and IDE of

Eclipse. I had Eclipse downloaded already.I downloaded a connector to connect that connect with Eclipse is shown below:

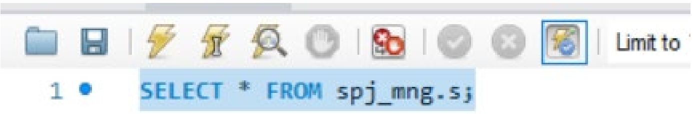


```java
String sql = "INSERT INTO S (SNO, SNAME, STATUS, CITY) VALUES(?,?,?,?)";
PreparedStatement statement = connection.prepareStatement(sql);
statement.setString(1, "S6");
statement.setString(2,  "ABID");
statement.setInt(3, 22);
statement.setString(4, "Kunming");

int rows = statement.executeUpdate();
if(rows > 0) {
    System.out.println("A record has been inserted");
}

stat = connection.createStatement() ;
ret = stat.executeQuery("select * from s");

while(ret.next())
{
    for(int i =1; i<=4; i++)
     System.out.print(ret.getString(i)+"  ");

    System.out.println(" ");
}

statement.close();
connection.close();
```

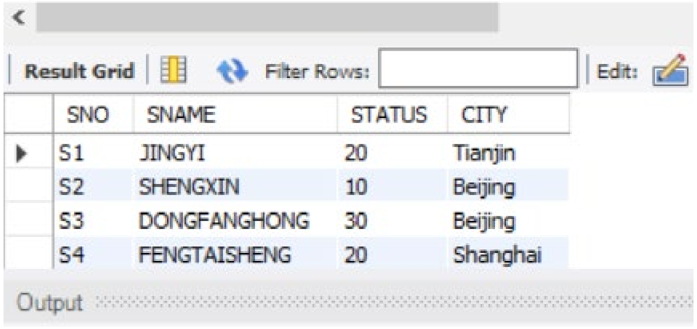
Fig: Connecting with JDBC and querying



```
Connected succefully to the database!
S1  JINGYI  20  Tianjin
S2  SHENGXIN  10  Beijing
S3  DONGFANGHONG  30  Beijing
S4  FENGTAISHENG  20  Shanghai
```

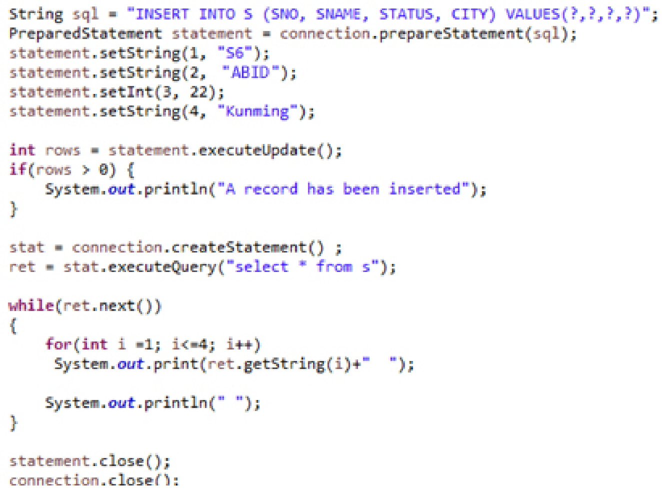


```sql
SELECT * FROM spj_mng.s;
```

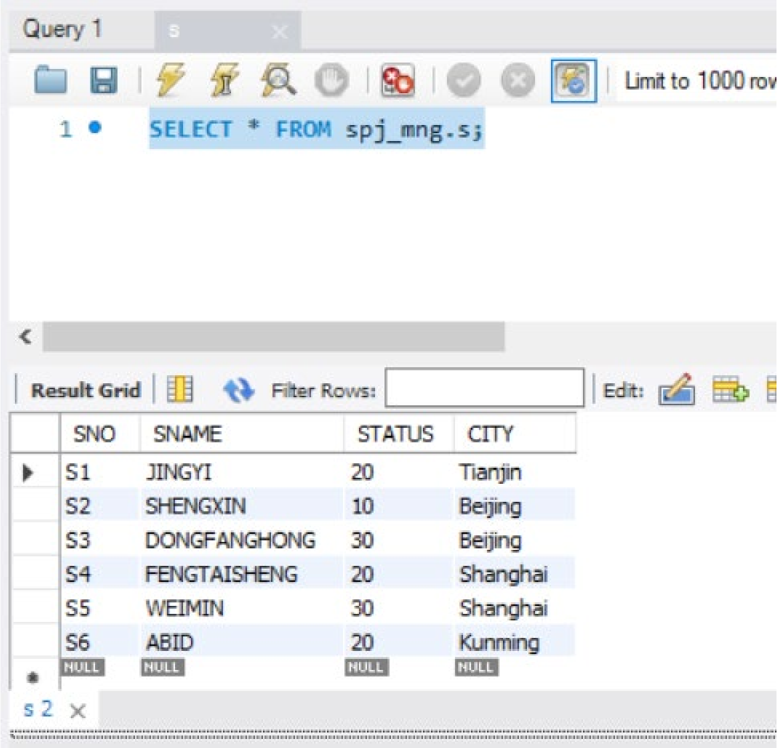


SNO	SNAME	STATUS	CITY
S1	JINGYI	20	Tianjin
S2	SHENGXIN	10	Beijing
S3	DONGFANGHONG	30	Beijing
S4	FENGTAISHENG	20	Shanghai

Inserting in table s



```
String sql = "INSERT INTO S (SNO, SNAME, STATUS, CITY) VALUES(?,?,?,?)";
PreparedStatement statement = connection.prepareStatement(sql);
statement.setString(1, "S6");
statement.setString(2,  "ABID");
statement.setInt(3, 22);
statement.setString(4, "Kunming");

int rows = statement.executeUpdate();
if(rows > 0) {
    System.out.println("A record has been inserted");
}

stat = connection.createStatement() ;
ret = stat.executeQuery("select * from s");

while(ret.next())
{
    for(int i =1; i<=4; i++)
     System.out.print(ret.getString(i)+"  ");

    System.out.println(" ");
}

statement.close();
connection.close();
```

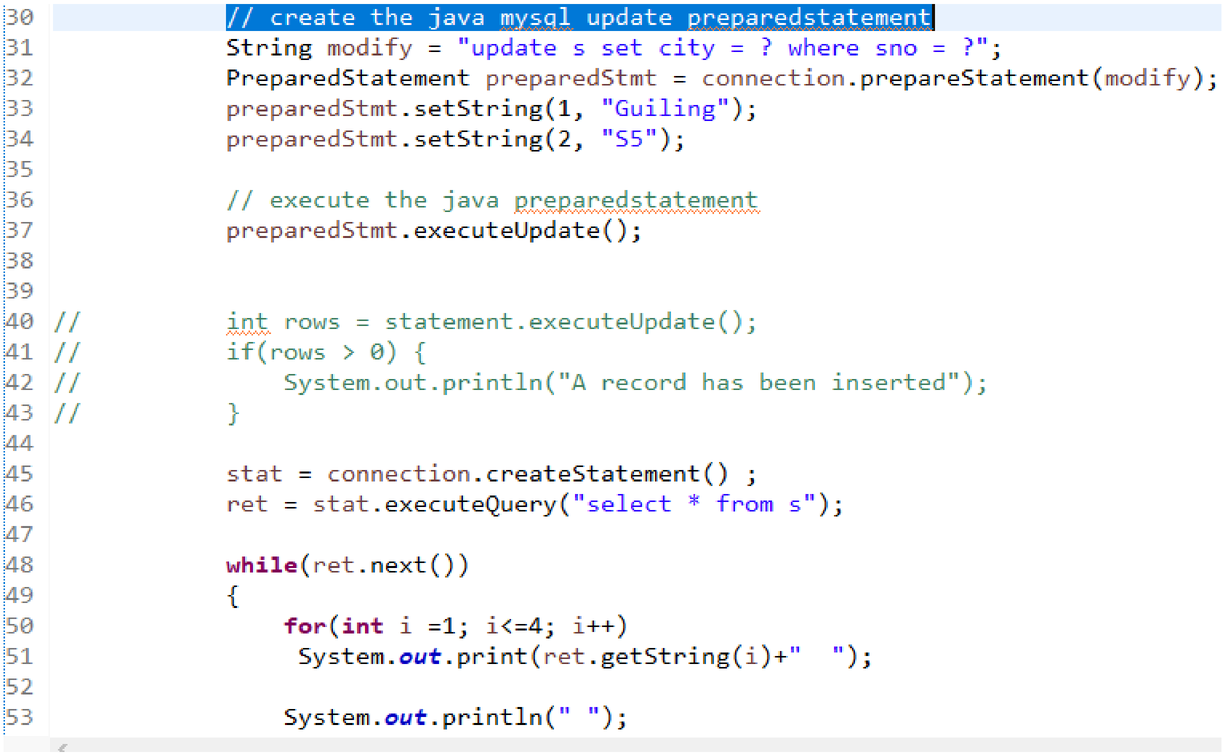


```
1  SELECT * FROM spj_mng.s;
```

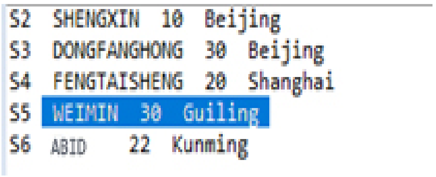
SNO	SNAME	STATUS	CITY
S1	JINGYI	20	Tianjin
S2	SHENGXIN	10	Beijing
S3	DONGFANGHONG	30	Beijing
S4	FENGTAISHENG	20	Shanghai
S5	WEIMIN	30	Shanghai
S6	ABID	20	Kunming
NULL	NULL	NULL	NULL

We can see that,we added my name in the table.That is SNO = S6 ,SNAME = ABID ,STATUS = 20 ,CITY =Kunming

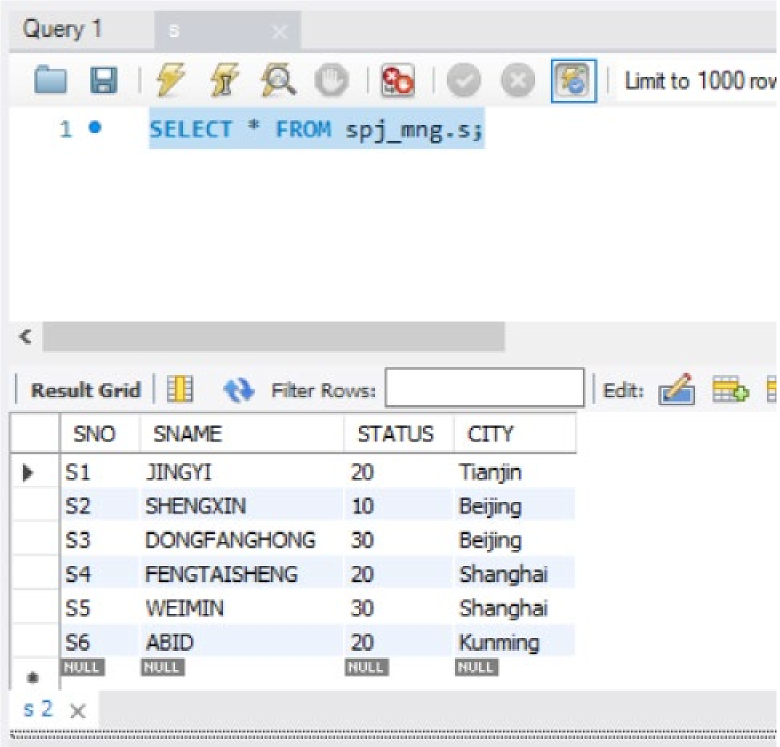
Modifying the table:



```
            // create the java mysql update preparedstatement
            String modify = "update s set city = ? where sno = ?";
            PreparedStatement preparedStmt = connection.prepareStatement(modify);
            preparedStmt.setString(1, "Guiling");
            preparedStmt.setString(2, "S5");

            // execute the java preparedstatement
            preparedStmt.executeUpdate();


//          int rows = statement.executeUpdate();
//          if(rows > 0) {
//              System.out.println("A record has been inserted");
//          }

            stat = connection.createStatement() ;
            ret = stat.executeQuery("select * from s");

            while(ret.next())
            {
                for(int i =1; i<=4; i++)
                 System.out.print(ret.getString(i)+"  ");

                System.out.println(" ");
```



```
S2  SHENGXIN  10  Beijing
S3  DONGFANGHONG  30  Beijing
S4  FENGTAISHENG  20  Shanghai
S5  WEIMIN  30  Guiling
S6  ABID    22  Kunming
```

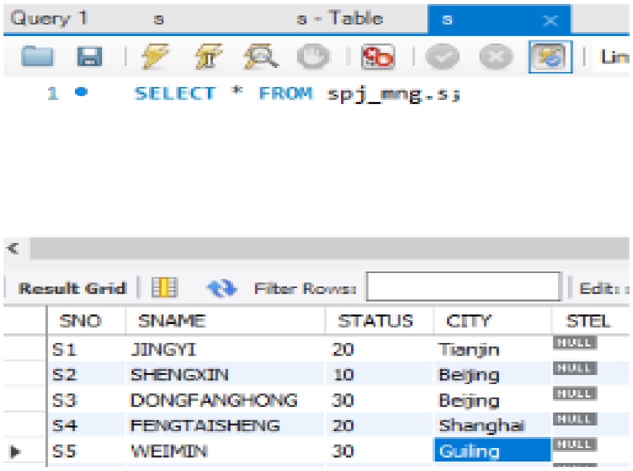


```
SELECT * FROM spj_mng.sj;
```

SNO	SNAME	STATUS	CITY
S1	JINGYI	20	Tianjin
S2	SHENGXIN	10	Beijing
S3	DONGFANGHONG	30	Beijing
S4	FENGTAISHENG	20	Shanghai
S5	WEIMIN	30	Shanghai
S6	ABID	20	Kunming
NULL	NULL	NULL	NULL

Fig: Before Modification

We modified "WEIMIN" where SNO =S5 from Shanghai To Guiling



```
SELECT * FROM spj_mng.sj
```

SNO	SNAME	STATUS	CITY	STEL
S1	JINGYI	20	Tianjin	NULL
S2	SHENGXIN	10	Beijing	NULL
S3	DONGFANGHONG	30	Beijing	NULL
S4	FENGTAISHENG	20	Shanghai	NULL
S5	WEIMIN	30	Guiling	NULL

Fig: After modifying the table

3. Comprehensive application experiment of bank finance. (50 points)

(1) According to the requirements, write SQL statements of different scenarios for the bank database.

Solution:

Creating A table for Bank card:

```
USE [BankDb]
GO
/****** Object: Table [dbo].[BankCard] Script Date: 11/06/2021 20:27:59 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[BankCard](
[Card Number] [int] NOT NULL,
[Customer Number] [int] NULL,
[Card type] [varchar](500) NULL,
[balance] [float] NULL,
CONSTRAINT [PK_BankCard] PRIMARY KEY CLUSTERED
(
[Card Number] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
```

Creating a Table for Customer:

```
USE [BankDb]
GO
/****** Object: Table [dbo].[Customer] Script Date: 11/06/2021 20:28:49 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Customer](
[Customer Number] [int] NOT NULL,
[Name] [varchar](500) NULL,
[Phone] [varchar](500) NULL,
[ID Card] [varchar](500) NULL,
[Password] [varchar](500) NULL,
CONSTRAINT [PK_Customer] PRIMARY KEY CLUSTERED
(
[Customer Number] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
```

Creating a Table for Finance Product:

```
USE [BankDb]
GO
/****** Object: Table [dbo].[Finance Product] Script Date: 11/06/2021 20:33:59 23:20:27 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Finance Product](
[Finance Number] [int] NOT NULL,
[Name] [varchar](500) NULL,
[Description] [varchar](MAX) NULL,
[Price] [float] NULL,
[Period] [varchar](500) NULL,
[Close start date] [date] NULL,
[Start start date] [date] NULL,
[status] [varchar](500) NULL,
CONSTRAINT [PK_Finance Product] PRIMARY KEY CLUSTERED
(
[Finance Number] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
```

(2) Referring to the programming examples of JDBC / ODBC / third-party library, use one of the ways to access the bank database from the application program, and execute the SQL statement of question (1).

The scenario description of specific banks is as follows:

Suppose that there are five basic entities for the business of Bank C: customers, bank cards, financial products, insurance and funds. For these entities, it is assumed that bank C has the following businesses:

A customer can apply for multiple bank cards.

A customer can purchase multiple financial products, and each type of financial product can be purchased by multiple customers.

A customer can purchase multiple funds, and the same type of fund can be purchased by multiple customers.

A customer can purchase multiple insurance, and the same type of insurance can be purchased by multiple customers

According to the business relationship of Bank C, the following ER diagram is obtained.

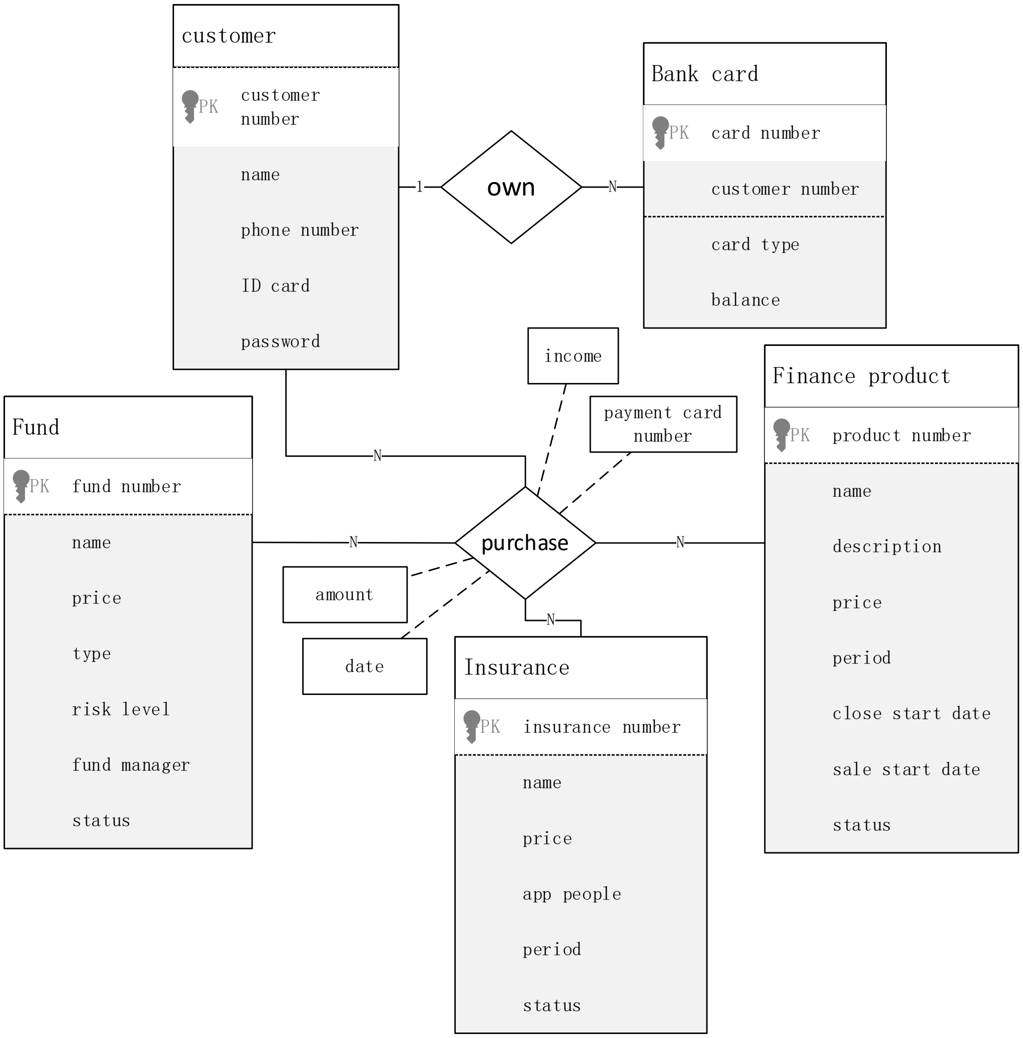


图 8-1 ER diagram of C Bbank System Finance

Customer Buys Funds:

- A complete description of the five entities in Bank C is as follows:

Customer: customer number, customer name, customer ID, customer phone number, customer login password

Bank card: bank card number, bank card type, customer number, balance

Financial products: product name, product number, product status, product price, time period, close start date, sale start date, product discription

Insurance: Insurance name, insurance number, insurance price, applicable population, insurance period, product status

Fund: fund name, fund number, fund type, fund price, risk level, fund manager, fund status

- The tables of all entities in the database finance are as follows:

Customer table: customer (C_ ID, C_ NAME, C_ ID_ CARD, C_ PHONE, C_ PASSWORD)

Card table: Bank_ card （B_ NUMBER, B_ TYPE, B_ C_ ID, B_ BALANCE）

Financial products table:finances_ product （P_ NAME，P_ ID，P_ DESCRIPTION，P_ PRICE，P_CLOSE_DATE, P_SALE_DATE, P_STATUS, P_ YEAR）

Insurance table:insurance(I_ NAME，I_ ID，I_ PRICE，I_ PERSON，I_ YEAR，I_ STATUS）

Fund table: fund (F_ NAME，F_ ID，F_ TYPE，F_ PRICE，RISK_ LEVEL，F_ MANAGER, F_STATUS）

- The purchase relationships between different entities in Bank C are described as follows:

Financial products purchase table: customer number, financial product number, purchase time,purchase quantity, cumulative income, payment bank card number

Insurance purchase table: customer number, insurance number, insurance time, purchase quantity, cumulative income, payment bank card number

Fund purchase table: customer number, fund number, time of fund purchase, purchase quantity, cumulative income, payment bank card number

- The corresponding table of purchase relationships in database finance are as follows:

Financial products purchase table: C_finances(c_ ID, P_ ID, P_ TIME, P_ AMOUNT, P_ INCOME, B_ C_ ID)

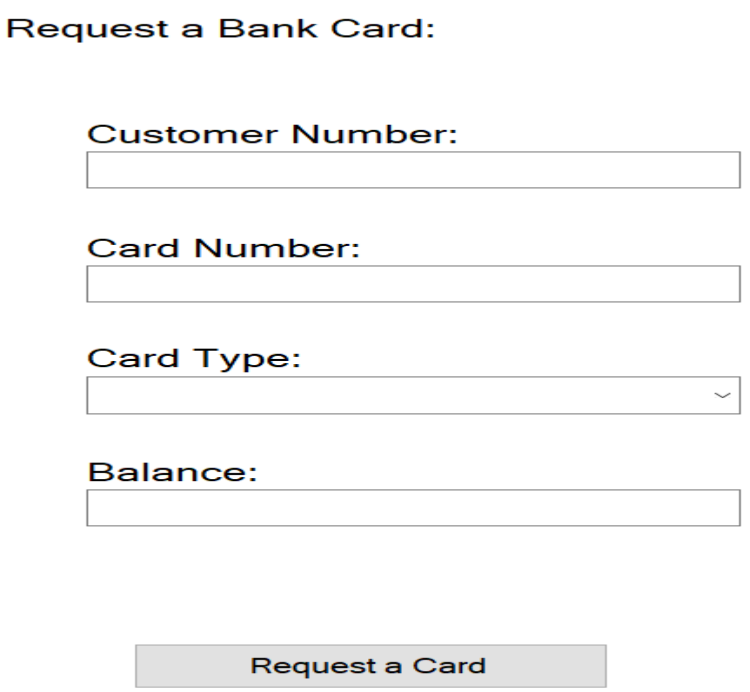
Insurance purchase table: C_ insurance (C_ ID, I_ ID, I_ TIME, I_ AMOUNT, I_ INCOME, B_ C_ ID)

Fund purchase table: C_ fund (C_ ID, F_ ID, F_ TIME, F_ AMOUNT, F_ INCOME , B_ C_ ID)

Based on the DDL file of the database and the prepared data, complete the following functions:

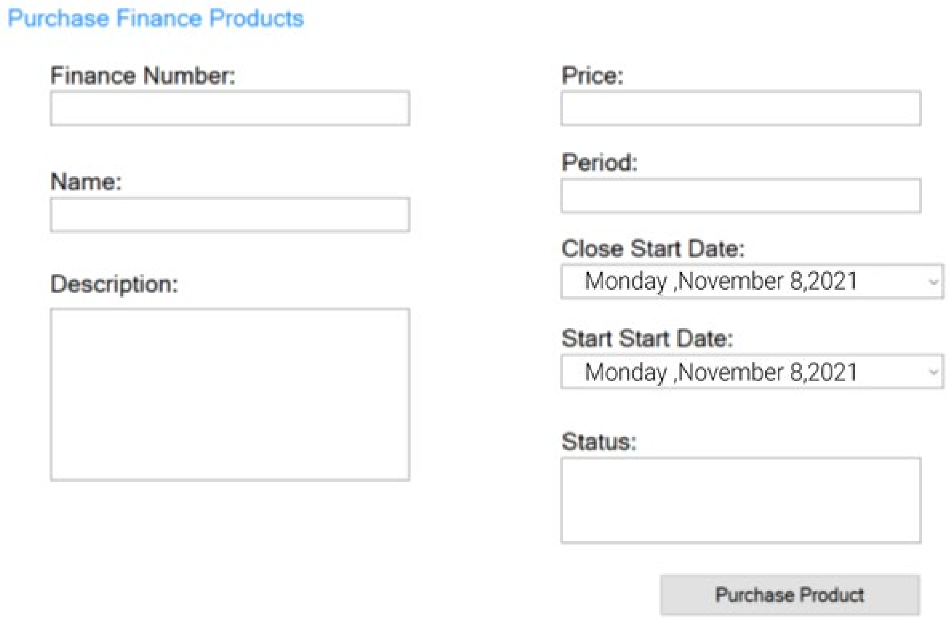
Solution:

The process how I solved the above questions are given below:



```
        Try
{
con.Open();
SqlCommand cmd = con.CreateCommand();
cmd.CommandText = "INSERT INTO BankCard VALUES('" + CardNumTb.Text
+ "', '" + CustomerNumTb.Text + "', '" + CardTypeComboBox.Text
+ "', '" + BalanceTb.Text + "')";
cmd.ExecuteNonQuery();
con.Close();
        }
●       catch
        { con.Close(); }
```

Customer Purchases Finance Products:

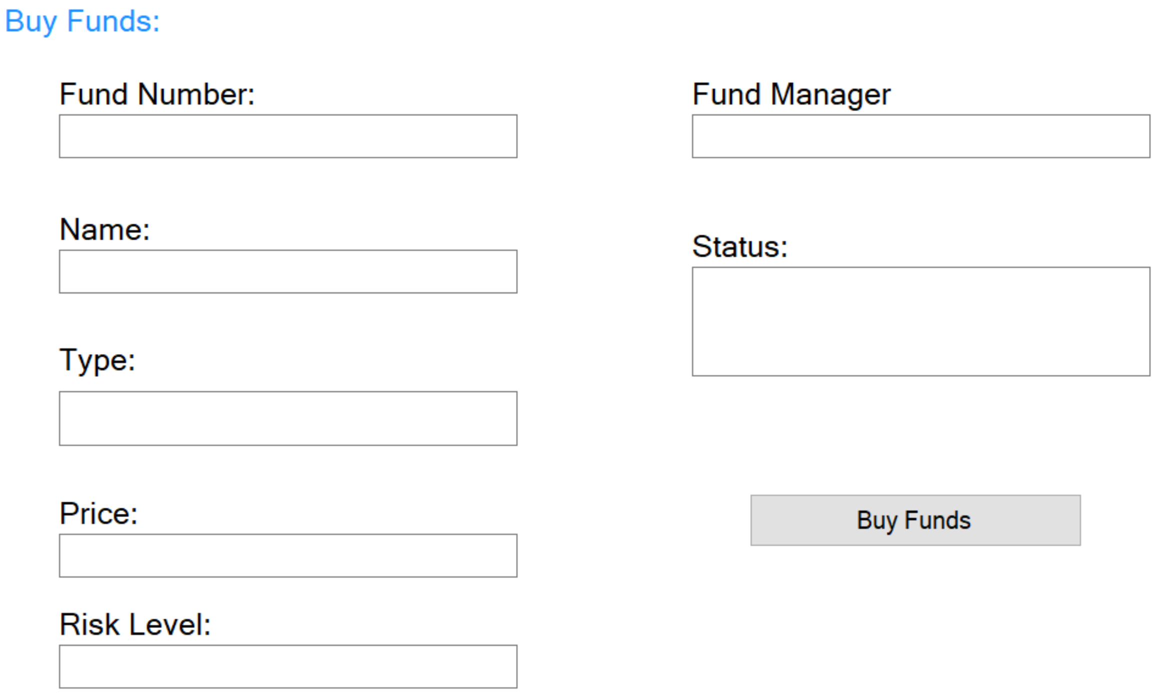


Close Start Date: Monday ,November 8,2021

Start Start Date: Monday ,November 8,2021

```
try
{
con.Open();
SqlCommand cmd = con.CreateCommand();
cmd.CommandText = "INSERT INTO [Finance Product] VALUES('" + FinanceNumTb.Text + "', '" +
NameTb.Text + "', '" + DescriptionTb.Text
+ "', '" + PriceTb.Text + "', '" + PeriodTb.Text
+ "', '" + CloseStartDateDtPicker.Value
+ "', '" + StartDateDtPicker.Value
+ "', '" + StatusTb.Text
+ "')";
cmd.ExecuteNonQuery();
con.Close();
}
catch
{ con.Close(); }
```

Buying funds:

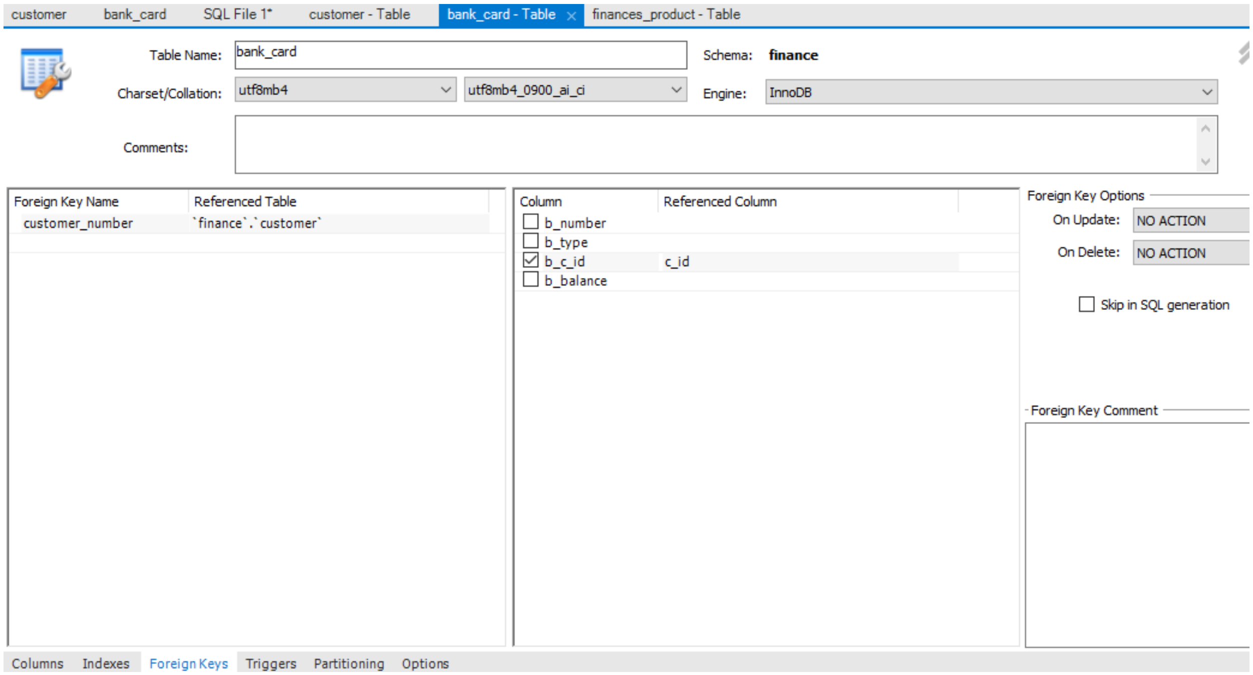


```
try
{
con.Open();
SqlCommand cmd = con.CreateCommand();
cmd.CommandText = "INSERT INTO [Fund] VALUES('" + FundNumberTb.Text + "', '" + NameTb.Text + "', '" + TypeTb.Text
+ "', '" + PriceTb.Text + "', '" + RiskLevelTb.Text
+ "', '" + FundManagerTb.Text
+ "', '" + StatusTb.Text
+ "')";
cmd.ExecuteNonQuery();
con.Close();
}
catch
        { con.Close(); }
```

(1) A new customer, whose ID number is "610103123456781234", registers at bank C, and applys for a new debit card. Please insert the record of the customer in the customer table and the card table.

(2) Add table constraints according to the business needs, and verify them after the constraints are added successfully.

① In the bank card table, financial product purchase table, insurance purchase table and fund purchase table, add the correct foreign key constraints: that is, the customer number is set as a foreign key, and references the customer number in the customer table, and the financial product number, insurance number and fund number refer to the corresponding table respectively.



customer | bank_card | SQL File 1* | customer - Table | bank_card - Table | finances_product - Table

Table Name: bank_card
Schema: finance
Charset/Collation: utf8mb4 | utf8mb4_0900_ai_ci
Engine: InnoDB
Comments:

Foreign Key Name	Referenced Table
customer_number	\`finance\`.\`customer\`

Column	Referenced Column
□ b_number	
□ b_type	
☑ b_c_id	c_id
□ b_balance	

Foreign Key Options
On Update: NO ACTION
On Delete: NO ACTION
□ Skip in SQL generation

Foreign Key Comment

② In the above basic tables, there are six attributes related to amount or price. In the real life, the amount or price will not be negative. Therefore, for these properties, add constraints whose value is greater than 0.

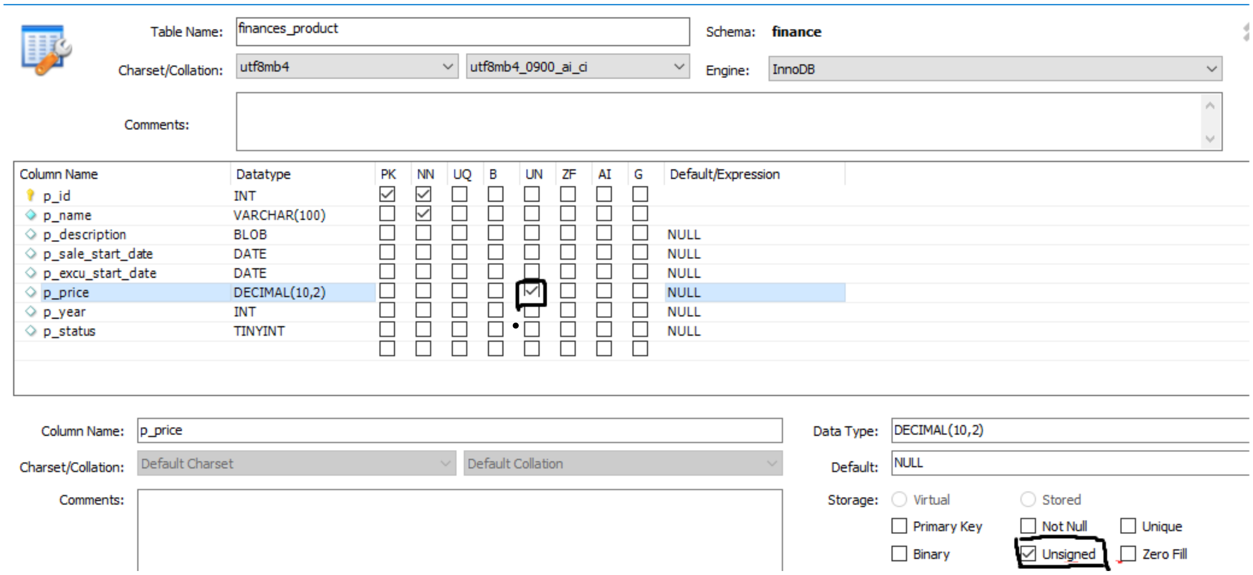


Table Name: finances_product
Schema: finance
Charset/Collation: utf8mb4 | utf8mb4_0900_ai_ci
Engine: InnoDB
Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
p_id	INT	☑	☑	□	□	□	□	□	□	
p_name	VARCHAR(100)	□	☑	□	□	□	□	□	□	
p_description	BLOB	□	□	□	□	□	□	□	□	NULL
p_sale_start_date	DATE	□	□	□	□	□	□	□	□	NULL
p_excu_start_date	DATE	□	□	□	□	□	□	□	□	NULL
p_price	DECIMAL(10,2)	□	□	□	□	☑	□	□	□	NULL
p_year	INT	□	□	□	□	□	□	□	□	NULL
p_status	TINYINT	□	□	□	□	□	□	□	□	NULL
		□	□	□	□	□	□	□	□	

Column Name: p_price
Charset/Collation: Default Charset | Default Collation
Comments:
Data Type: DECIMAL(10,2)
Default: NULL
Storage: ○ Virtual ○ Stored
□ Primary Key □ Not Null □ Unique
□ Binary ☑ Unsigned □ Zero Fill

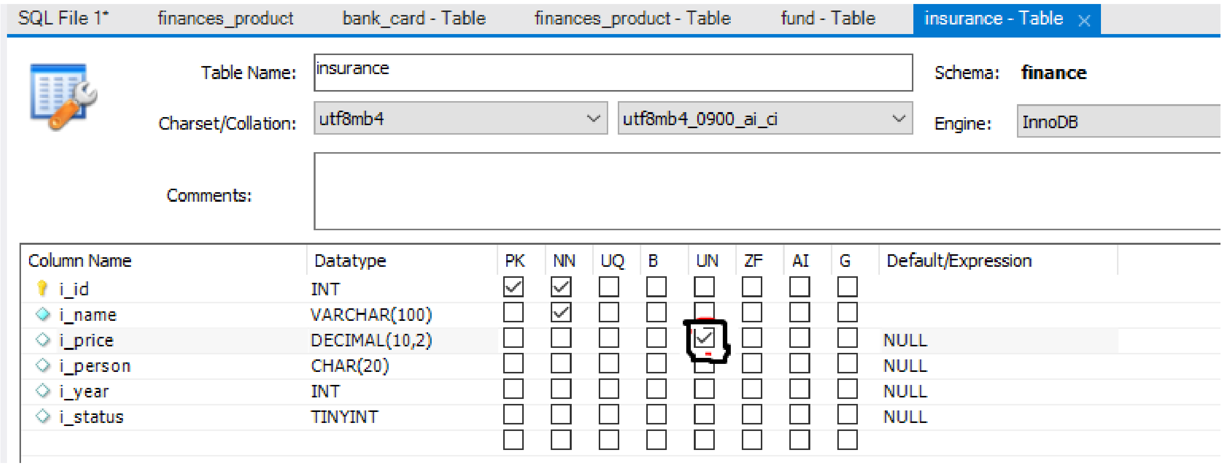


Table Name: insurance

Schema: **finance**

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci

Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
i_id	INT	☑	☑	□	□	□	□	□	□	
i_name	VARCHAR(100)	□	☑	□	□	□	□	□	□	
i_price	DECIMAL(10,2)	□	□	□	□	☑	□	□	□	NULL
i_person	CHAR(20)	□	□	□	□	□	□	□	□	NULL
i_year	INT	□	□	□	□	□	□	□	□	NULL
i_status	TINYINT	□	□	□	□	□	□	□	□	NULL
		□	□	□	□	□	□	□	□	

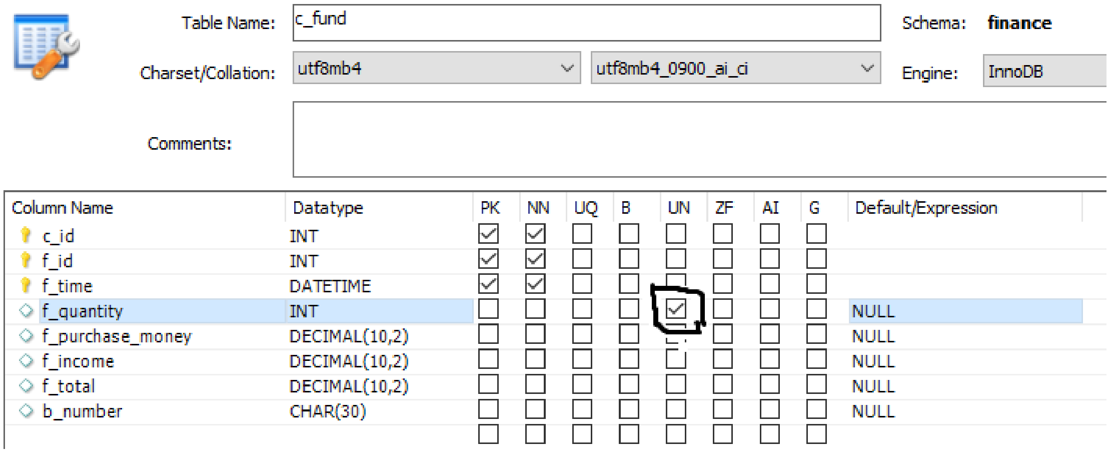


Table Name: c_fund

Schema: **finance**

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci

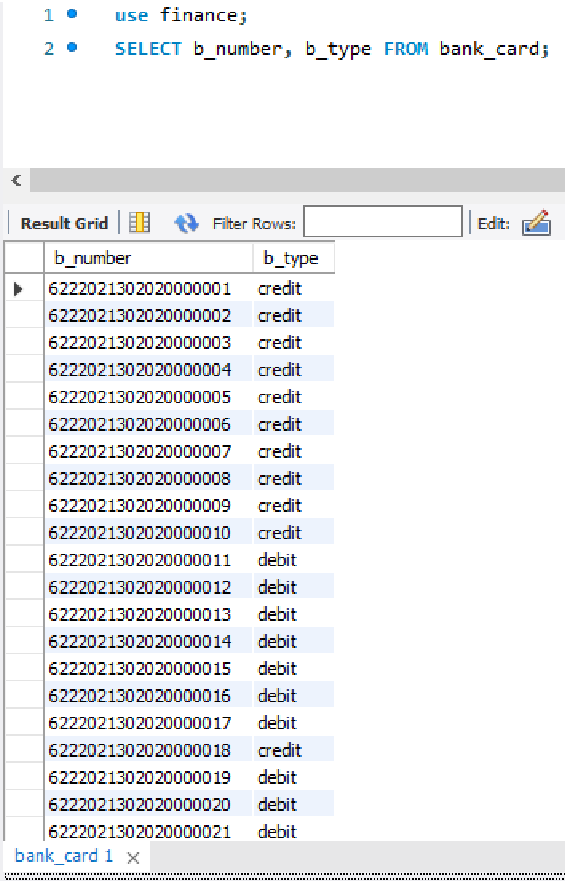
Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
c_id	INT	☑	☑	□	□	□	□	□	□	
f_id	INT	☑	☑	□	□	□	□	□	□	
f_time	DATETIME	☑	☑	□	□	□	□	□	□	
f_quantity	INT	□	□	□	□	☑	□	□	□	NULL
f_purchase_money	DECIMAL(10,2)	□	□	□	□	□	□	□	□	NULL
f_income	DECIMAL(10,2)	□	□	□	□	□	□	□	□	NULL
f_total	DECIMAL(10,2)	□	□	□	□	□	□	□	□	NULL
b_number	CHAR(30)	□	□	□	□	□	□	□	□	NULL
		□	□	□	□	□	□	□	□	

(3) Simulate the following business to write SQL statements for query:

① Query the card number and type information of all bank cards of bank C.

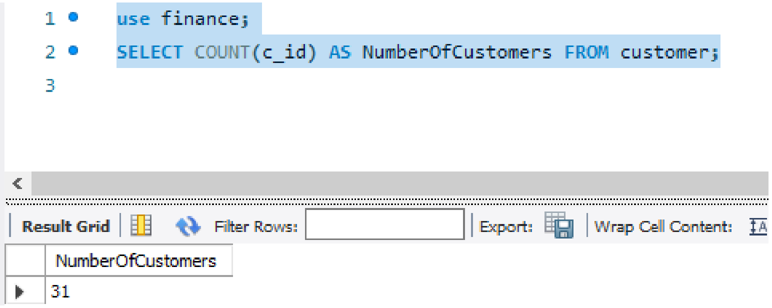


```
use finance;
SELECT b_number, b_type FROM bank_card;
```

b_number	b_type
6222021302020000001	credit
6222021302020000002	credit
6222021302020000003	credit
6222021302020000004	credit
6222021302020000005	credit
6222021302020000006	credit
6222021302020000007	credit
6222021302020000008	credit
6222021302020000009	credit
6222021302020000010	credit
6222021302020000011	debit
6222021302020000012	debit
6222021302020000013	debit
6222021302020000014	debit
6222021302020000015	debit
6222021302020000016	debit
6222021302020000017	debit
6222021302020000018	credit
6222021302020000019	debit
6222021302020000020	debit
6222021302020000021	debit

We can see different important cards in the table.

② Query the number of customers owned by bank C.



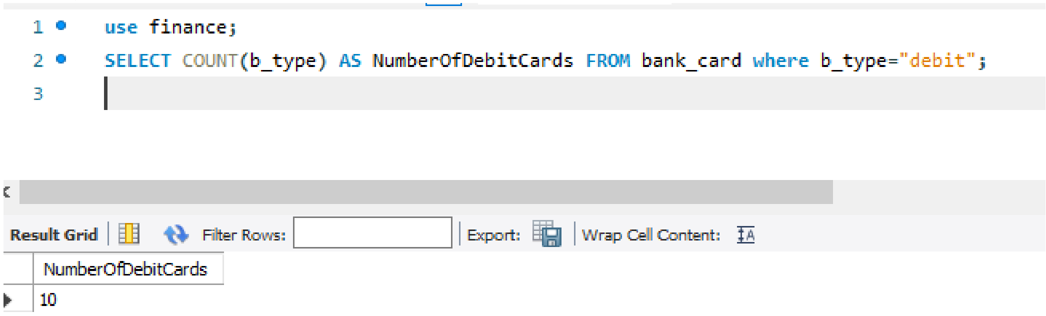
```
use finance;
SELECT COUNT(c_id) AS NumberOfCustomers FROM customer;

```

NumberOfCustomers
31

Bank has 31 customers

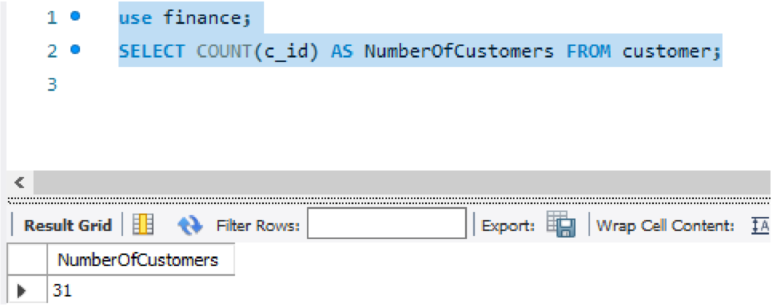
③ query the customer number, name and ID number information of the bank cards owners.

④ Statistics the amount of debit cards and credit cards respectively in all the bank cards.



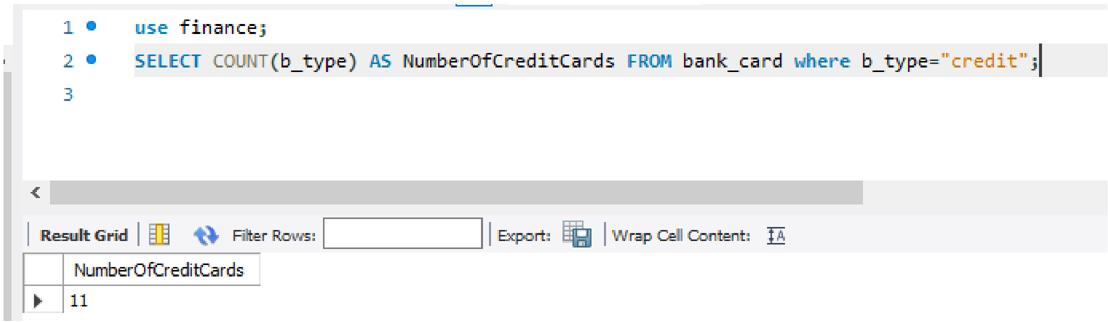
```
use finance;
SELECT COUNT(b_type) AS NumberOfDebitCards FROM bank_card where b_type="debit";
```

```
NumberOfDebitCards
10
```



```
use finance;
SELECT COUNT(c_id) AS NumberOfCustomers FROM customer;
```

```
NumberOfCustomers
31
```

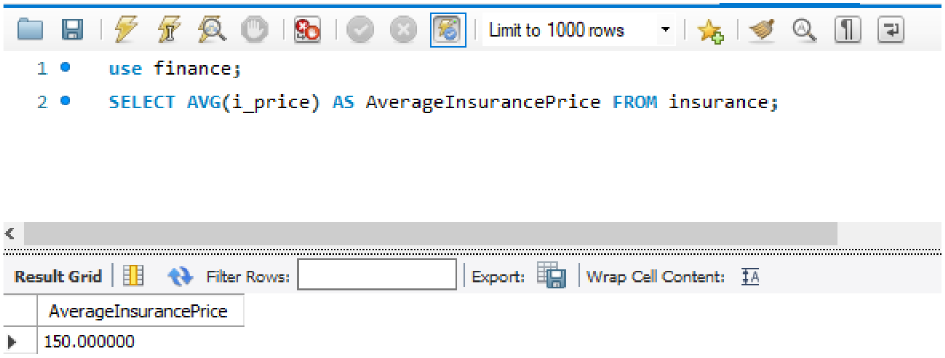


```
use finance;
SELECT COUNT(b_type) AS NumberOfCreditCards FROM bank_card where b_type="credit";
```

```
NumberOfCreditCards
11
```

Amount of debit cards and credit cards respectively in all the bank cards are shown above.

⑤ Query the average insurance price in the insurance table.

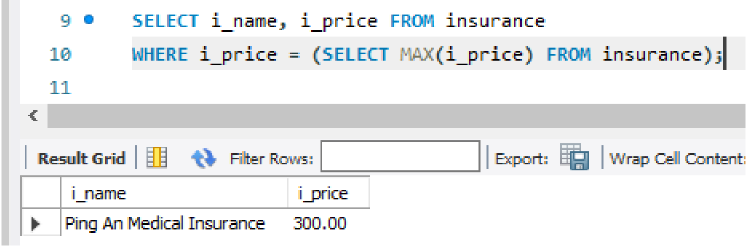


```sql
use finance;
SELECT AVG(i_price) AS AverageInsurancePrice FROM insurance;
```

AverageInsurancePrice
150.000000

We have created the average of the insurance by using average function.

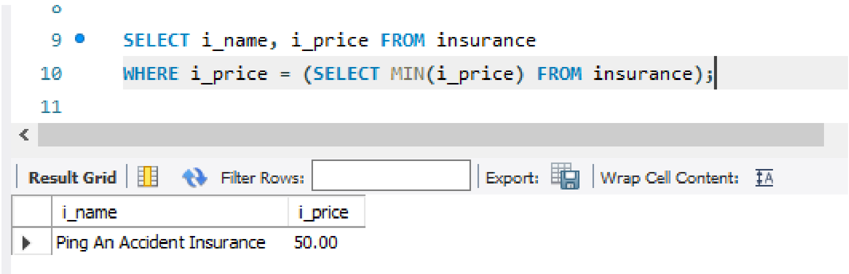
⑥ Query the insurance type and price corresponding to the maximum and minimum of insurance price in the insurance table.



```sql
SELECT i_name, i_price FROM insurance
WHERE i_price = (SELECT MAX(i_price) FROM insurance);
```

i_name	i_price
Ping An Medical Insurance	300.00

Fig: Maximum Insurance price

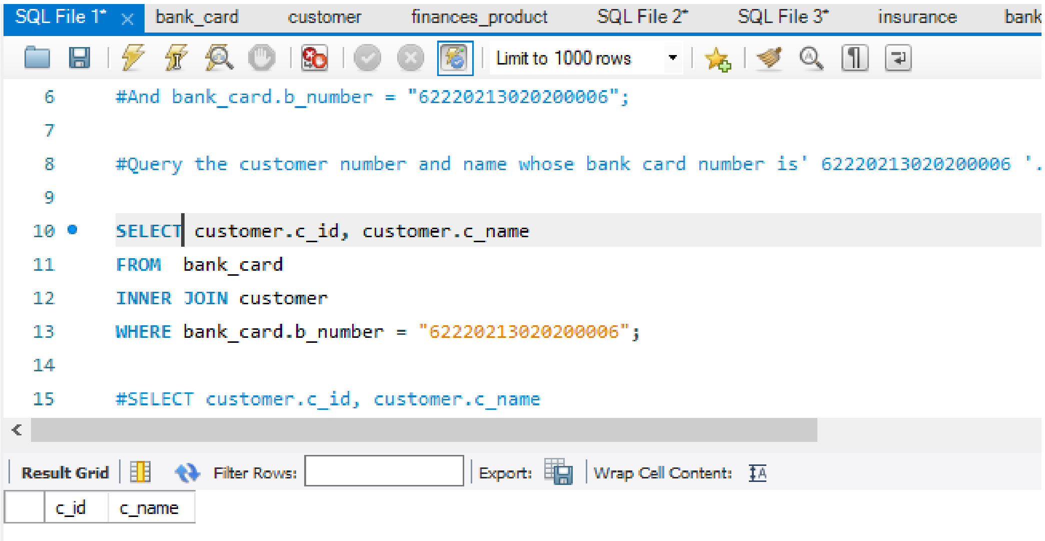


```sql
SELECT i_name, i_price FROM insurance
WHERE i_price = (SELECT MIN(i_price) FROM insurance);
```

i_name	i_price
Ping An Accident Insurance	50.00

Fig: Minimum Insurance price

⑦ Query the customer number and name whose bank card number is' 6222021302020006 '.

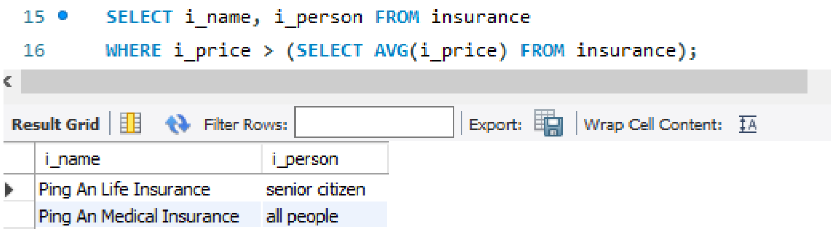


```
#And bank_card.b_number = "6222021302020006";

#Query the customer number and name whose bank card number is' 6222021302020006 '.

SELECT customer.c_id, customer.c_name
FROM  bank_card
INNER JOIN customer
WHERE bank_card.b_number = "6222021302020006";

#SELECT customer.c_id, customer.c_name
```

c_id	c_name

We searched the entry.Couldn't find the entry

⑧ Query the insurance name and applicable population whose insurance price is greater than the average value in the insurance product.

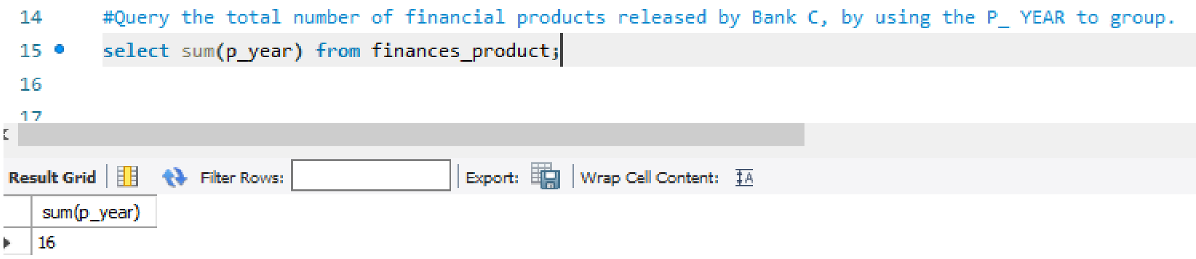


```
SELECT i_name, i_person FROM insurance
WHERE i_price > (SELECT AVG(i_price) FROM insurance);
```

i_name	i_person
Ping An Life Insurance	senior citizen
Ping An Medical Insurance	all people

The insurance name and applicable population whose insurance price is greater than the average value in the insurance product given above.

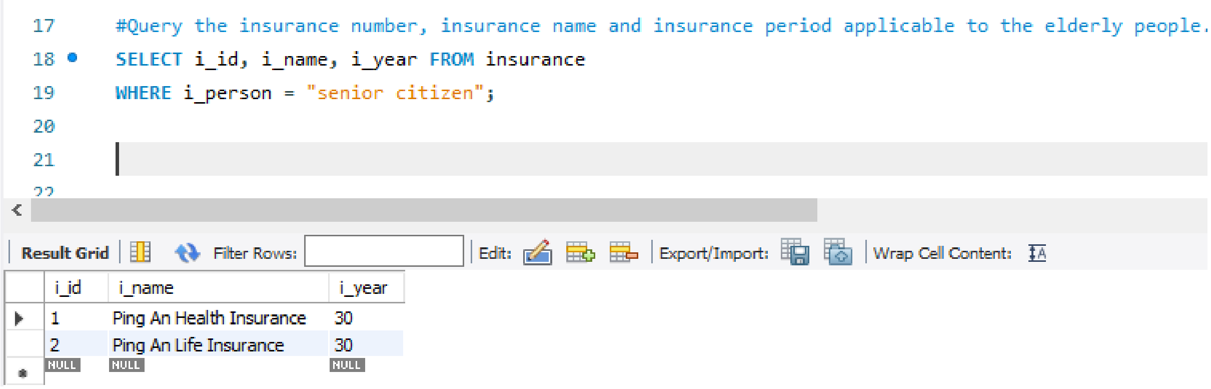
⑨ Query the total number of financial products released by Bank C, by using the P_ YEAR to group.



```
#Query the total number of financial products released by Bank C, by using the P_ YEAR to group.
select sum(p_year) from finances_product;
```

sum(p_year)
16

⑩ Query the insurance number, insurance name and insurance period applicable to the elderly people.

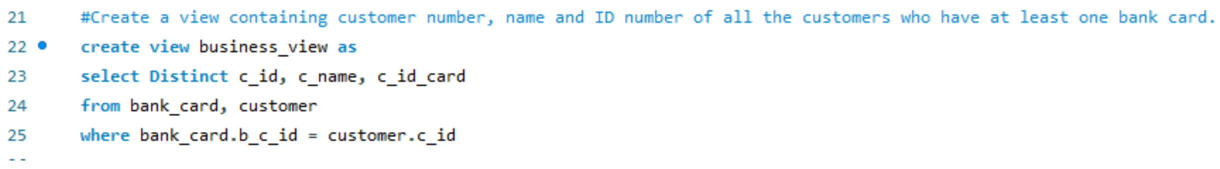


```
#Query the insurance number, insurance name and insurance period applicable to the elderly people.
SELECT i_id, i_name, i_year FROM insurance
WHERE i_person = "senior citizen";


```

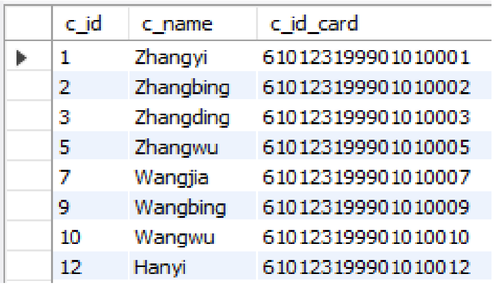
i_id	i_name	i_year
1	Ping An Health Insurance	30
2	Ping An Life Insurance	30
NULL	NULL	NULL

(4) Create a business view based on the following queries:

① Create a view containing customer number, name and ID number of all the customers who have at least one bank card.

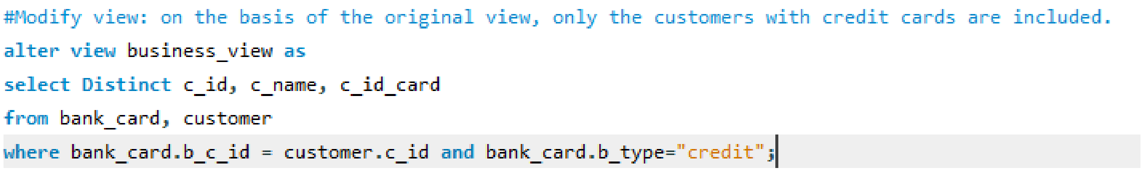


```
#Create a view containing customer number, name and ID number of all the customers who have at least one bank card.
create view business_view as
select Distinct c_id, c_name, c_id_card
from bank_card, customer
where bank_card.b_c_id = customer.c_id
```

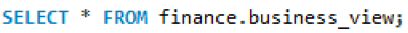


c_id	c_name	c_id_card
1	Zhangyi	610123199901010001
2	Zhangbing	610123199901010002
3	Zhangding	610123199901010003
5	Zhangwu	610123199901010005
7	Wangjia	610123199901010007
9	Wangbing	610123199901010009
10	Wangwu	610123199901010010
12	Hanyi	610123199901010012

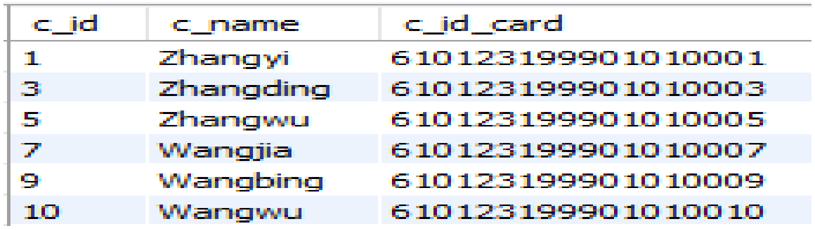
② Modify view: on the basis of the original view, only the customers with credit cards are included.



```
#Modify view: on the basis of the original view, only the customers with credit cards are included.
alter view business_view as
select Distinct c_id, c_name, c_id_card
from bank_card, customer
where bank_card.b_c_id = customer.c_id and bank_card.b_type="credit";
```



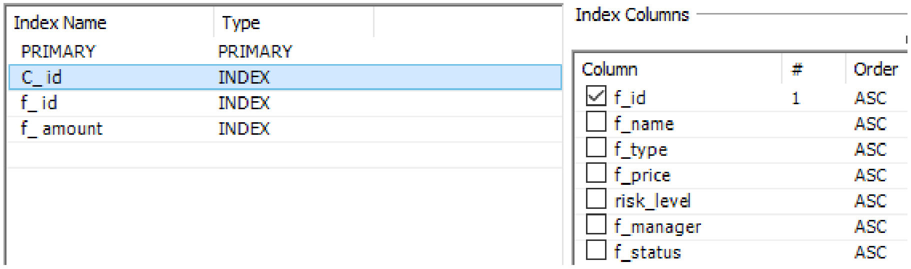
```
SELECT * FROM finance.business_view;
```



c_id	c_name	c_id_card
1	Zhangyi	610123199901010001
3	Zhangding	610123199901010003
5	Zhangwu	610123199901010005
7	Wangjia	610123199901010007
9	Wangbing	610123199901010009
10	Wangwu	610123199901010010

(5) Simulate the business changes, people's demand for fund query has increased significantly. Create a composite index on the fund purchase table:

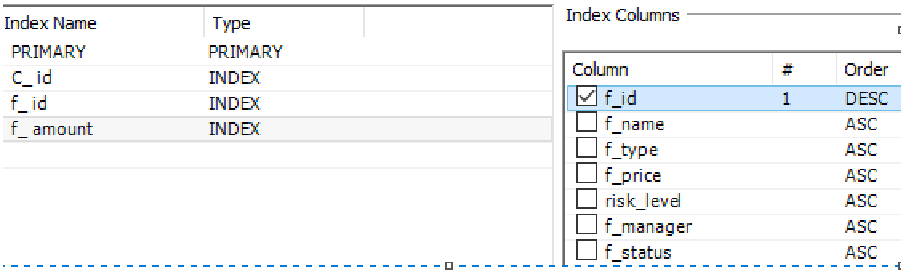
C_ id ASC, f_ id ASC, f_ amount DESC



Index Name	Type
PRIMARY	PRIMARY
C_ id	INDEX
f_ id	INDEX
f_ amount	INDEX

Index Columns

Column	#	Order
☑ f_id	1	ASC
☐ f_name		ASC
☐ f_type		ASC
☐ f_price		ASC
☐ risk_level		ASC
☐ f_manager		ASC
☐ f_status		ASC



Index Name	Type
PRIMARY	PRIMARY
C_ id	INDEX
f_ id	INDEX
f_ amount	INDEX

Index Columns

Column	#	Order
☑ f_id	1	DESC
☐ f_name		ASC
☐ f_type		ASC
☐ f_price		ASC
☐ risk_level		ASC
☐ f_manager		ASC
☐ f_status		ASC

Problems:

I have never used visual studio and JAVA to make buttons .In this project ,I felt overwhelmed because I have to do so many things in short amount of time. I had to learn C# language which I felt little bit unfamiliar and familiar because it looks like C++ and JAVA but there are it's own set of rules.The main problems were setting sqlserver database in Visual Studio and IDE related problem.Problems with the settings.

Solutions:

To solve these problems which I faced during doing this practical, I took help from internet especially YouTube,StackOverflow and W3school to get information about these errors for the solution. I also asked the teacher to help me understand them. And provided instructions helped to solve some of my errors during the experiment.

Summary:

In this project I had to learn lot of things.I learnt about the establishment of the connection sqlserver with database and visual studio application along with how to make buttons and connect those buttons with mysql.I learned the basic use of backend and frontend.

Attachments:

1) DB4_2019380141_ABID ALI.pdf

References:

1) https://www.w3schools.com/

2) https://stackoverflow.com/

3) https://youtube.com/

4) https://www.youtube.com/watch?v=-EPMOaV7h_Q

5) https://www.youtube.com/watch?v=GhQdlIFylQ8

6) https://www.youtube.com/watch?v=_JxC6EUxbDo

7) https://www.youtube.com/watch?v=deRSq-Fb2BM