Data Bases Coursework Documentation

Technical University of Varna

Created by:

Georgi Sokolov

21621397

Task:

Design and implement a database on a topic chosen by the student, which contains a minimum of 6 tables, normalized to the third normal form - on a DBMS of choice (Oracle, MS SQL Server, MySQL, etc.), as well as a programming interface (implemented in C#, PHP, JAVA, etc.) - in the form of web / desktop / mobile - by choice.

The interface must implement the following capabilities:

- Adding/editing/deleting information in the tables
- Options for searches and reports (at least 5) with different criteria
- Printing the reports.

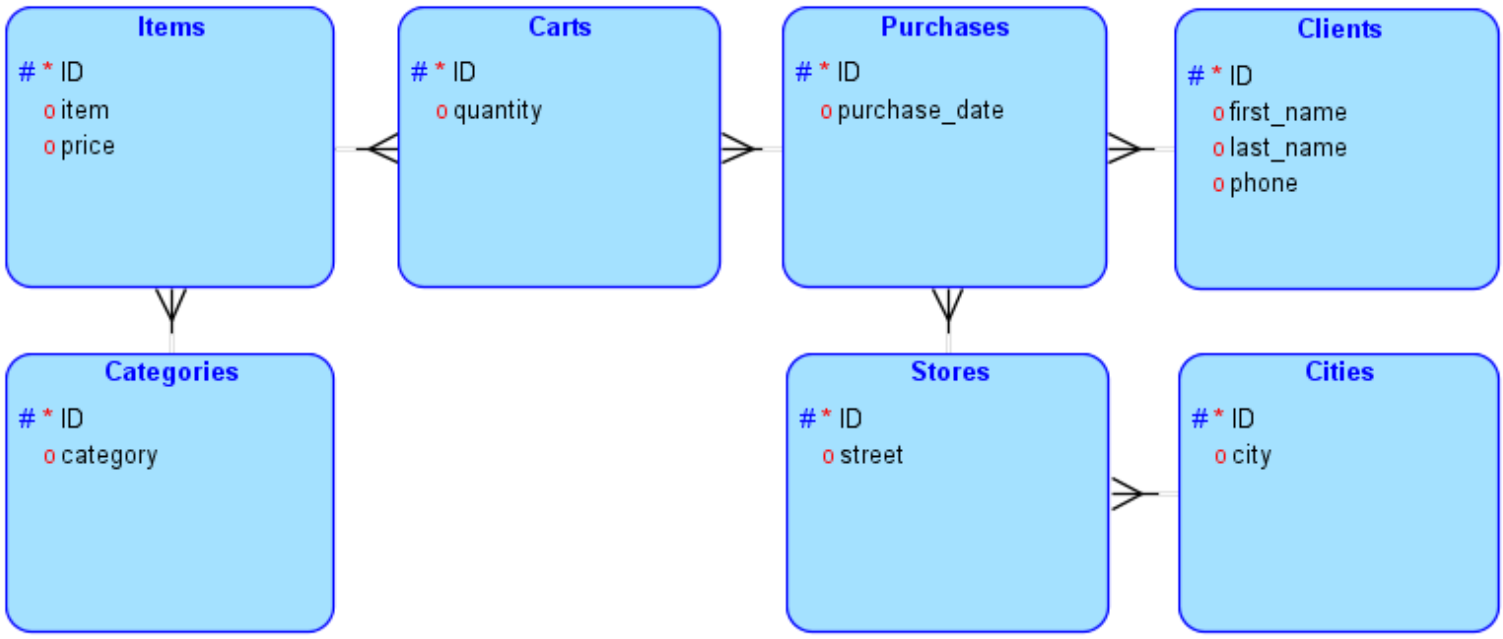
Chosen theme:

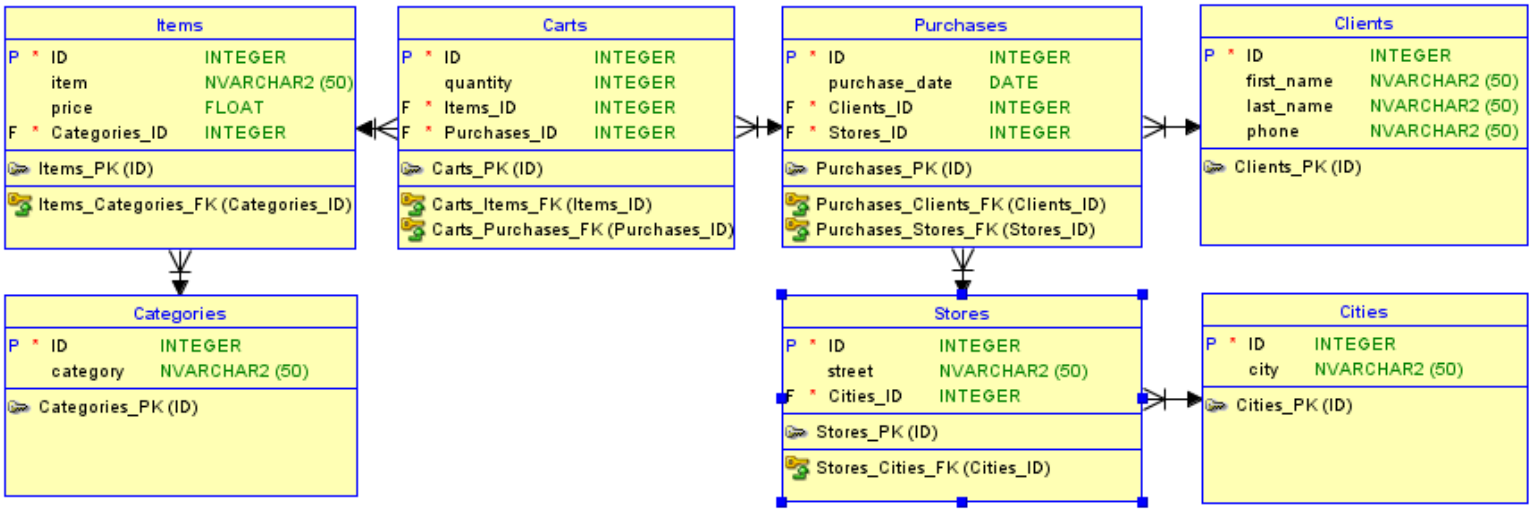
Database for a fictional store chain with 7 tables in 3rd normal form (Categories, Items, Carts, Purchases, Clients, Stores, Cities).

Chosen interface:

The UI is created in Visual Studio using Forms, C#. The database is also created using Visual Studio.

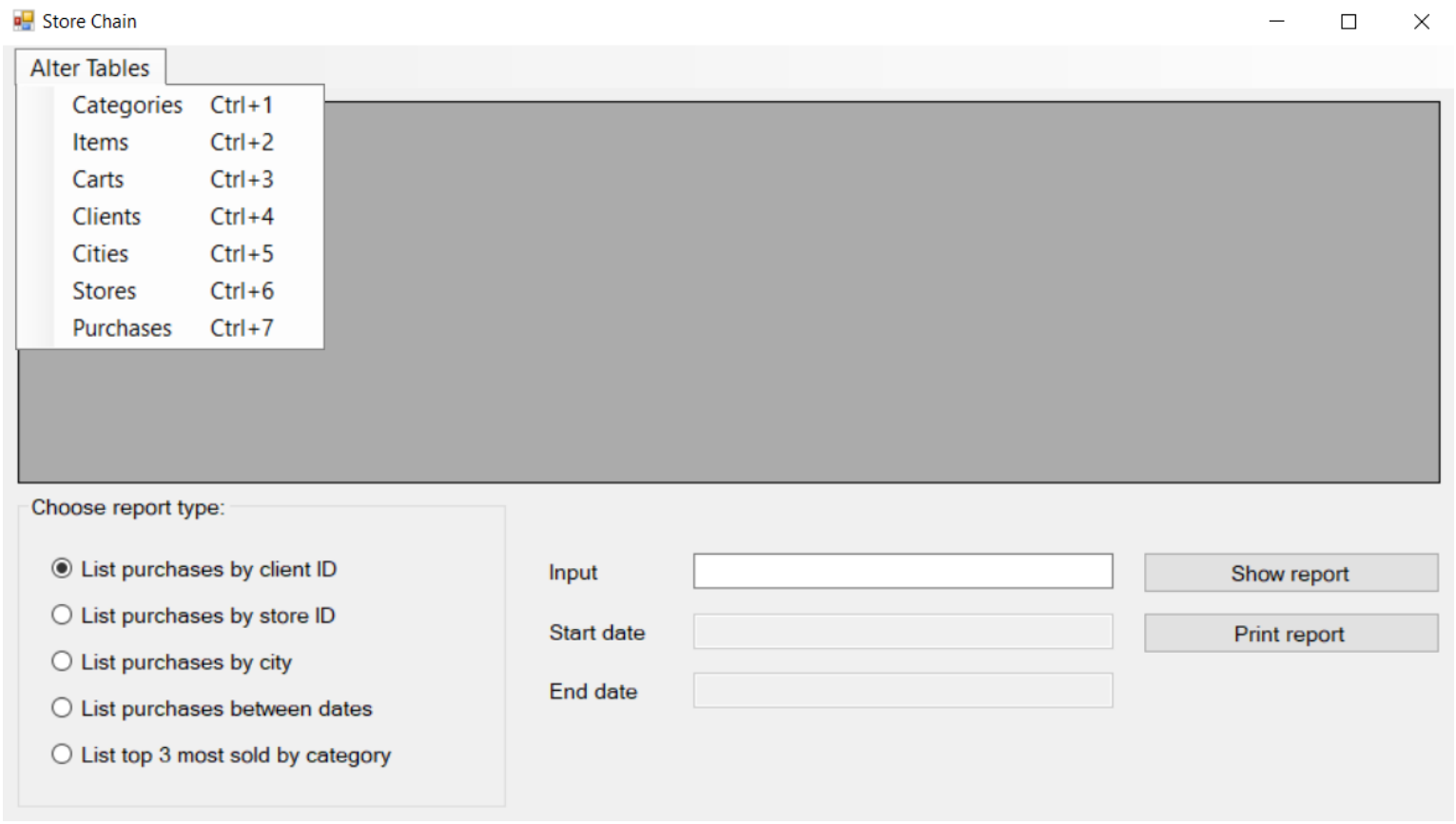
Relational and Logical models





User Manual

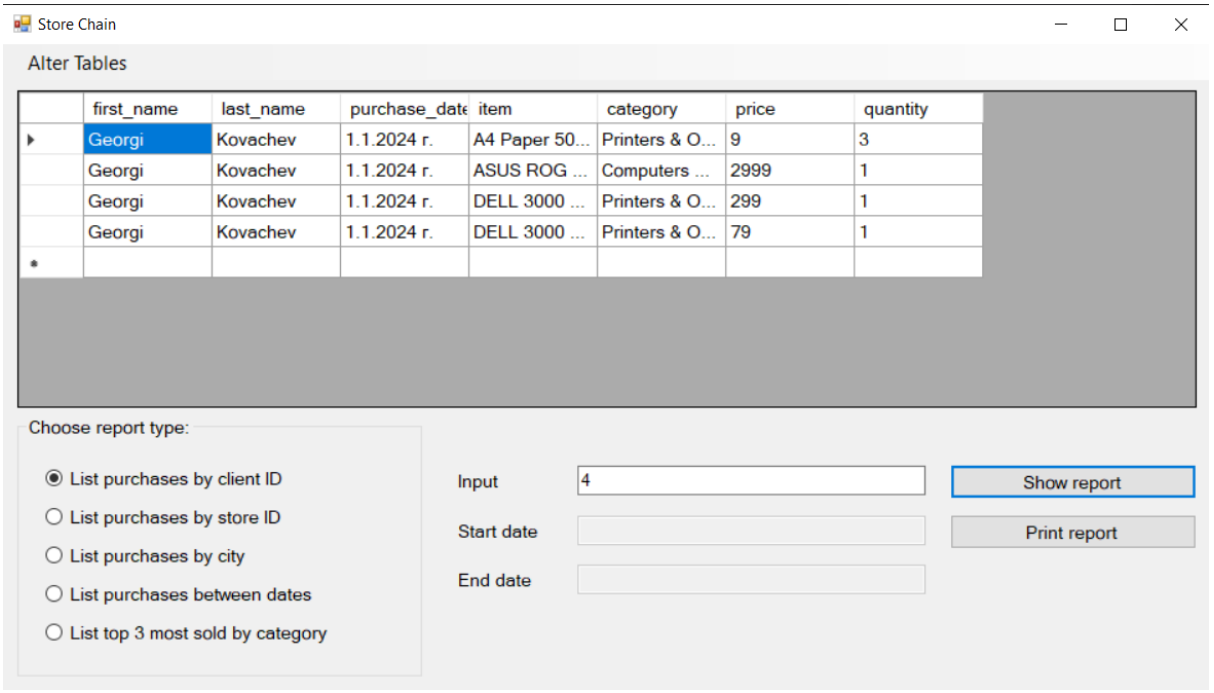
Main Form:

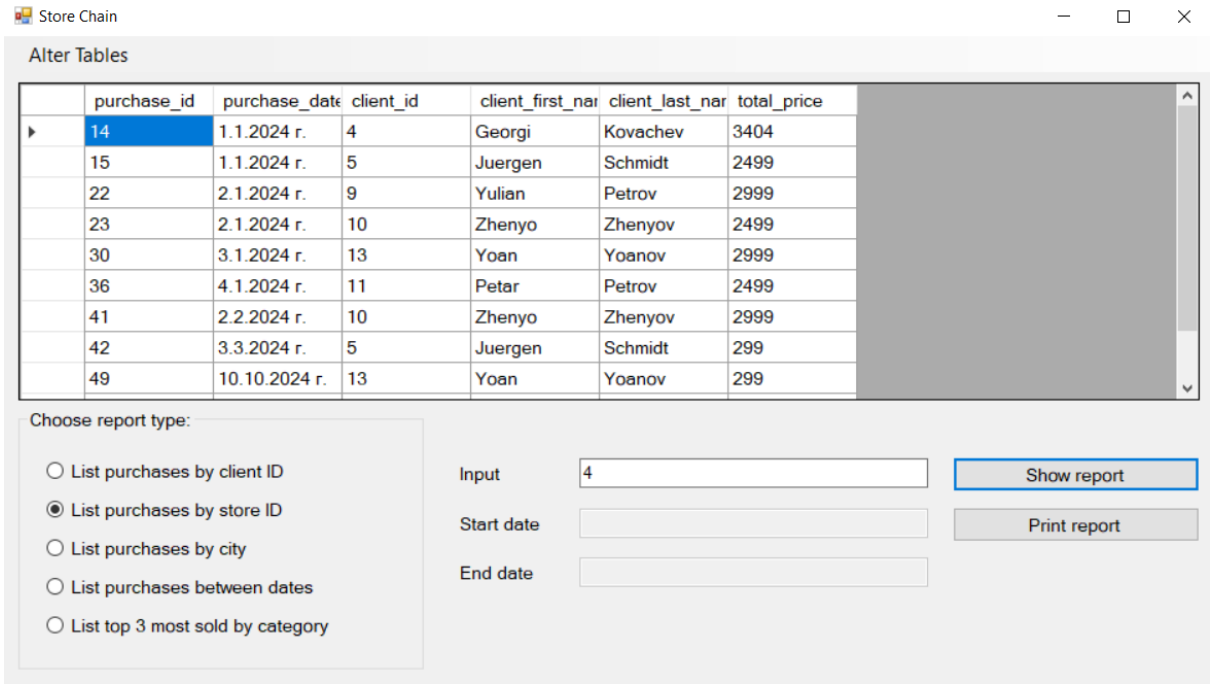


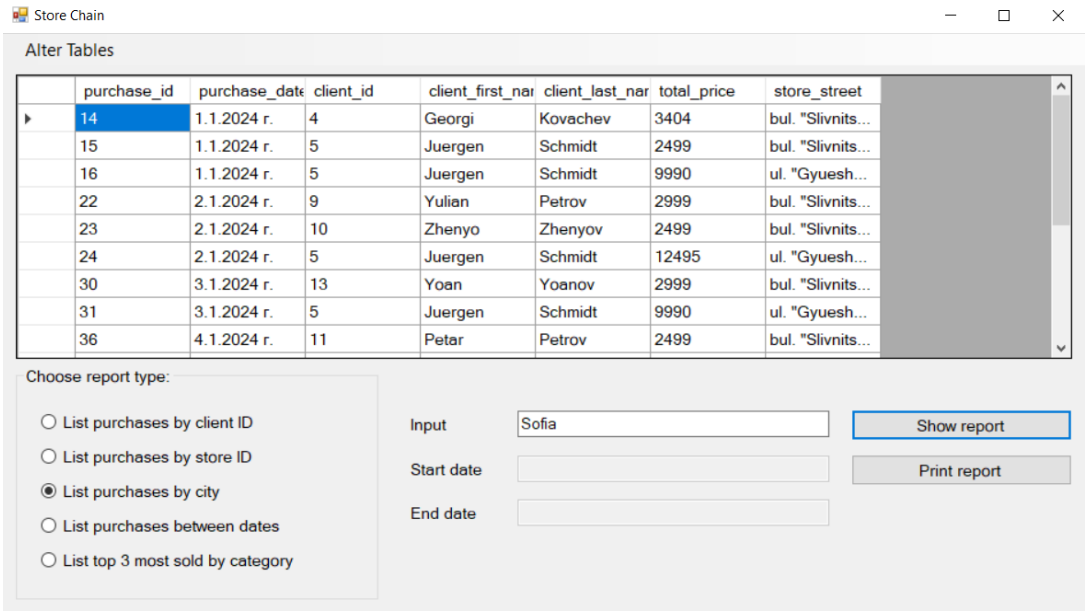
The forms for inserting into, updating, deleting from tables can be accessed using the

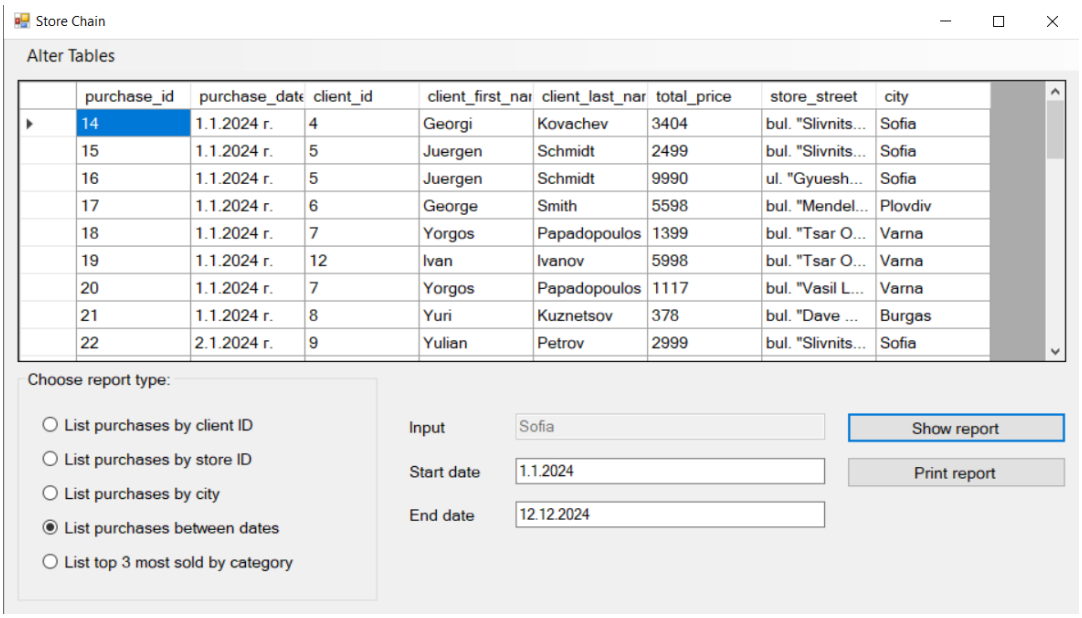
"Alter Tables" dropdown menu or by pressing: *Ctrl + (1 to 7)*.

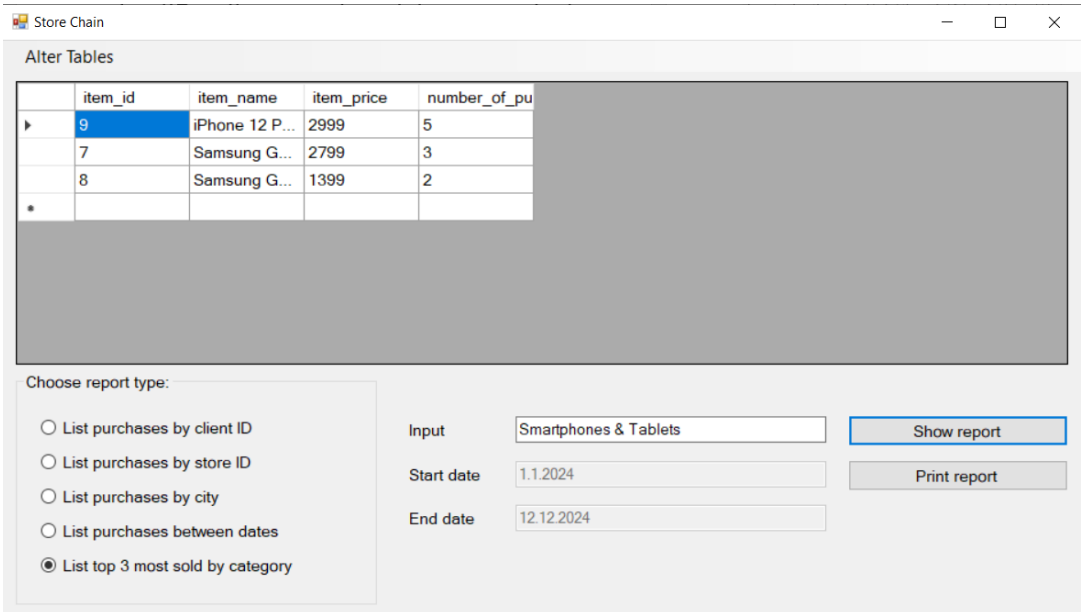
A report can be generated by selecting a radio button and entering the needed value (IDs, city name, two dates or product category) and pressing “*Show report*”.



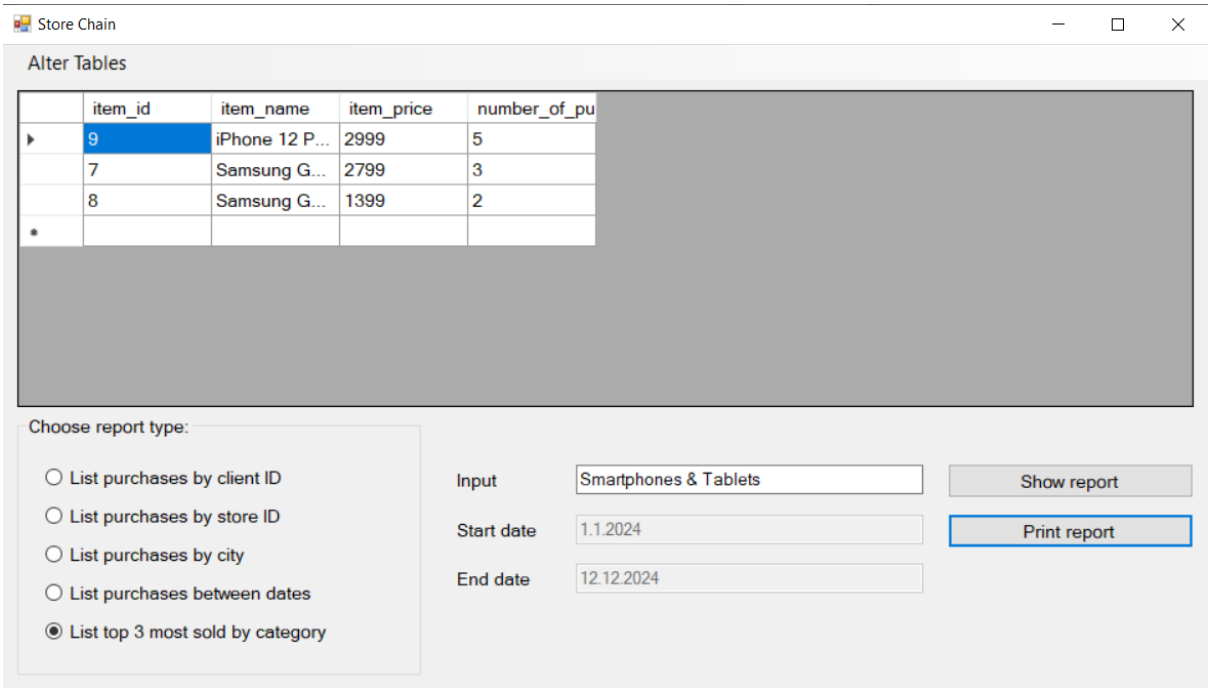


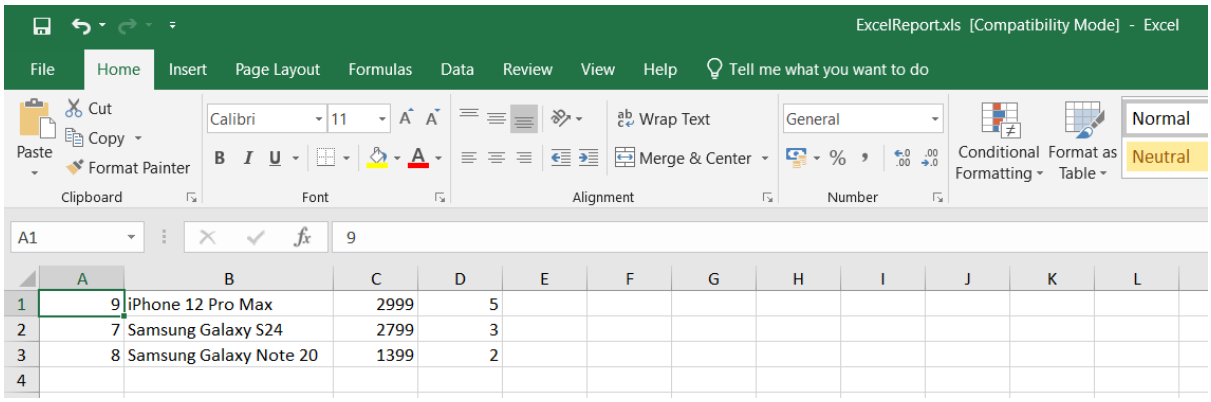

Store Chain
Alter Tables
purchase_id | purchase_date | client_id | client_first_nar | client_last_nar | total_price | store_street
14 | 1.1.2024 r. | 4 | Georgi | Kovachev | 3404 | bul. "Slivnits...
15 | 1.1.2024 r. | 5 | Juergen | Schmidt | 2499 | bul. "Slivnits...
16 | 1.1.2024 r. | 5 | Juergen | Schmidt | 9990 | ul. "Gyuesh...
22 | 2.1.2024 r. | 9 | Yulian | Petrov | 2999 | bul. "Slivnits...
23 | 2.1.2024 r. | 10 | Zhenyo | Zhenyov | 2499 | bul. "Slivnits...
24 | 2.1.2024 r. | 5 | Juergen | Schmidt | 12495 | ul. "Gyuesh...
30 | 3.1.2024 r. | 13 | Yoan | Yoanov | 2999 | bul. "Slivnits...
31 | 3.1.2024 r. | 5 | Juergen | Schmidt | 9990 | ul. "Gyuesh...
36 | 4.1.2024 r. | 11 | Petar | Petrov | 2499 | bul. "Slivnits...
Choose report type:
List purchases by client ID
List purchases by store ID
List purchases by city
List purchases between dates
List top 3 most sold by category
Input
Sofia
Start date
End date
Show report
Print report


Store Chain
Alter Tables
purchase_id | purchase_date | client_id | client_first_nar | client_last_nar | total_price | store_street | city
14 | 1.1.2024 r. | 4 | Georgi | Kovachev | 3404 | bul. "Slivnits... | Sofia
15 | 1.1.2024 r. | 5 | Juergen | Schmidt | 2499 | bul. "Slivnits... | Sofia
16 | 1.1.2024 r. | 5 | Juergen | Schmidt | 9990 | ul. "Gyuesh... | Sofia
17 | 1.1.2024 r. | 6 | George | Smith | 5598 | bul. "Mendel... | Plovdiv
18 | 1.1.2024 r. | 7 | Yorgos | Papadopoulos | 1399 | bul. "Tsar O... | Varna
19 | 1.1.2024 r. | 12 | Ivan | Ivanov | 5998 | bul. "Tsar O... | Varna
20 | 1.1.2024 r. | 7 | Yorgos | Papadopoulos | 1117 | bul. "Vasil L... | Varna
21 | 1.1.2024 r. | 8 | Yuri | Kuznetsov | 378 | bul. "Dave ... | Burgas
22 | 2.1.2024 r. | 9 | Yulian | Petrov | 2999 | bul. "Slivnits... | Sofia
Choose report type:
List purchases by client ID
List purchases by store ID
List purchases by city
List purchases between dates
List top 3 most sold by category
Input
Sofia
Start date
1.1.2024
End date
12.12.2024
Show report
Print report


Store Chain
Alter Tables
item_id | item_name | item_price | number_of_pu
9 | iPhone 12 P... | 2999 | 5
7 | Samsung G... | 2799 | 3
8 | Samsung G... | 1399 | 2
Choose report type:
List purchases by client ID
List purchases by store ID
List purchases by city
List purchases between dates
List top 3 most sold by category
Input
Smartphones & Tablets
Start date
1.1.2024
End date
12.12.2024
Show report
Print report

Pressing *"Print report"* creates an Excel spreadsheet ready for printing on the desktop, containing the data from the last generated report.





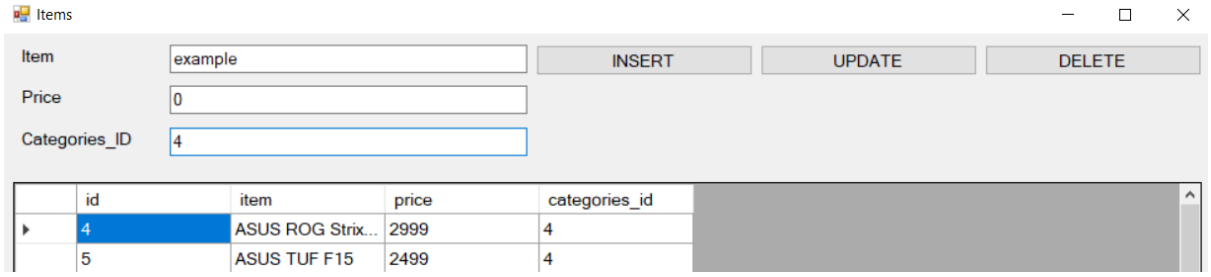
Inserting, Updating, Deleting

To alter a table an option in the “*Alter Tables*” menu has to be selected. IDs are incremented automatically

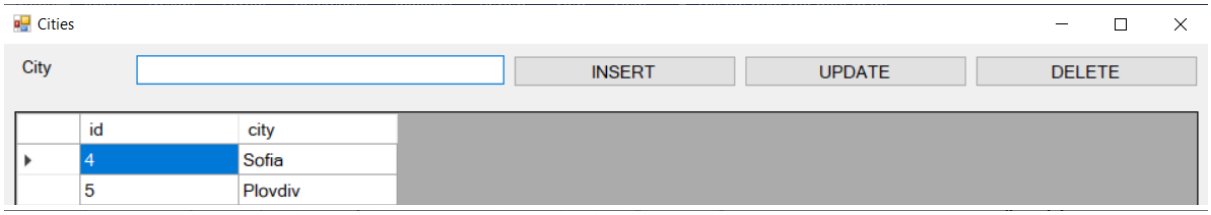
To insert into a table all values have to be entered in the textboxes, after which “INSERT” is pressed.

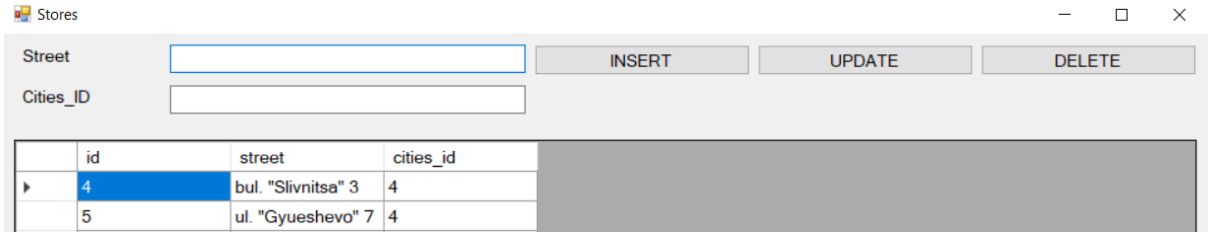
To update or delete a row needs to be selected by pressing on it and then using “UPDATE” or “DELETE”. The values of the row are automatically added in to the textboxes.

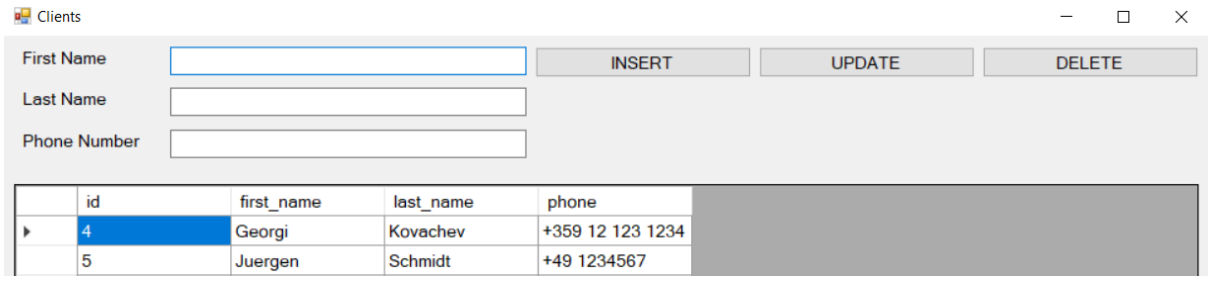
These operations are analogous for all tables.

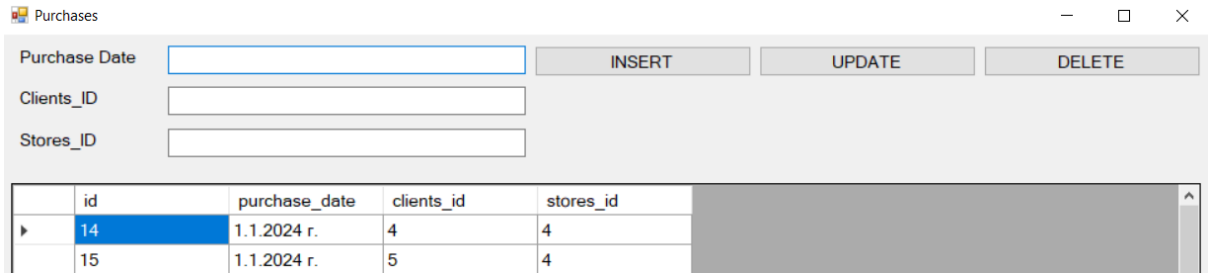

Items
Item
example
INSERT
UPDATE
DELETE
Price
0
Categories_ID
4
id
item
price
categories_id
4
ASUS ROG Strix...
2999
4
5
ASUS TUF F15
2499
4


Carts
Quantity
INSERT
UPDATE
DELETE
Items_ID
Purchases_ID
id
quantity
items_id
purchases_id
24
1
4
14
25
1
13
14


Cities
City
INSERT
UPDATE
DELETE
id
city
4
Sofia
5
Plovdiv


Stores
Street
INSERT
UPDATE
DELETE
Cities_ID
id
street
cities_id
4
bul. "Slivnitsa" 3
4
5
ul. "Gyueshevo" 7
4


Clients
First Name
INSERT
UPDATE
DELETE
Last Name
Phone Number
id
first_name
last_name
phone
4
Georgi
Kovachev
+359 12 123 1234
5
Juergen
Schmidt
+49 1234567


Purchases
Purchase Date
INSERT
UPDATE
DELETE
Clients_ID
Stores_ID
id
purchase_date
clients_id
stores_id
14
1.1.2024 г.
4
4
15
1.1.2024 г.
5
4

Code

Database:

```
CREATE TABLE carts (
   id          INTEGER NOT NULL PRIMARY KEY IDENTITY,
   quantity    INTEGER,
   items_id    INTEGER NOT NULL,
   purchases_id INTEGER NOT NULL
);
CREATE TABLE categories (
   id      INTEGER NOT NULL PRIMARY KEY IDENTITY,
   category NVARCHAR(50)
);
CREATE TABLE cities (
   id   INTEGER NOT NULL PRIMARY KEY IDENTITY,
   city NVARCHAR(50)
);
CREATE TABLE clients (
   id        INTEGER NOT NULL PRIMARY KEY IDENTITY,
   first_name NVARCHAR(50),
   last_name  NVARCHAR(50),
   phone     NVARCHAR(50)
);
CREATE TABLE items (
   id          INTEGER NOT NULL PRIMARY KEY IDENTITY,
   item         NVARCHAR(50),
   price        FLOAT,
   categories_id INTEGER NOT NULL
);
CREATE TABLE purchases (
   id          INTEGER NOT NULL PRIMARY KEY IDENTITY,
   purchase_date DATE,
   clients_id    INTEGER NOT NULL,
   stores_id     INT NOT NULL
);
CREATE TABLE stores (
   id       INTEGER NOT NULL PRIMARY KEY IDENTITY,
   street    NVARCHAR(50),
   cities_id INTEGER NOT NULL
);
ALTER TABLE carts
   ADD CONSTRAINT carts_items_fk FOREIGN KEY (
items_id )
      REFERENCES items ( id );
ALTER TABLE carts
   ADD CONSTRAINT carts_purchases_fk FOREIGN KEY (
purchases_id )
      REFERENCES purchases ( id );
ALTER TABLE items
   ADD CONSTRAINT items_categories_fk FOREIGN KEY (
categories_id )
      REFERENCES categories ( id );
ALTER TABLE purchases
   ADD CONSTRAINT purchases_clients_fk FOREIGN KEY
( clients_id )
      REFERENCES clients ( id );
ALTER TABLE purchases
   ADD CONSTRAINT purchases_stores_fk FOREIGN KEY (
stores_id )
      REFERENCES stores ( id );
ALTER TABLE stores
   ADD CONSTRAINT stores_cities_fk FOREIGN KEY (
cities_id )
      REFERENCES cities ( id );
```

F_Main.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;
using Excel = Microsoft.Office.Interop.Excel;

namespace DB_Georgi_Sokolov_21621397
{
    public partial class F_Main : Form
    {
        public F_Main()
        {
            InitializeComponent();
        }

        private void categoriesToolStripMenuItem_Click(object sender, EventArgs e)
        {
            F_Categories categories = new F_Categories();
            categories.Show();
        }

        private void itemsToolStripMenuItem_Click(object sender, EventArgs e)
        {
            F_Items items = new F_Items();
            items.Show();
        }

        private void cartsToolStripMenuItem_Click(object sender, EventArgs e)
        {
            F_Carts carts = new F_Carts();
            carts.Show();
        }

        private void clientsToolStripMenuItem_Click(object sender, EventArgs e)
        {
            F_Clients clients = new F_Clients();
            clients.Show();
        }

        private void citiesToolStripMenuItem_Click(object sender, EventArgs e)
        {
            F_Cities cities = new F_Cities();
            cities.Show();
        }

        private void storesToolStripMenuItem_Click(object sender, EventArgs e)
        {
            F_Stores stores = new F_Stores();
            stores.Show();
        }

        private void purchasesToolStripMenuItem_Click(object sender, EventArgs e)
        {
```

```
            F_Purchases purchases = new F_Purchases();
            purchases.Show();
        }

        private void setTextboxes(bool enabled)
        {
            textBox1.Enabled = !enabled;
            textBox2.Enabled = enabled;
            textBox3.Enabled = enabled;
        }

        string cs = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\gmsokolov\source\repos\DB_Geor
gi_Sokolov_21621397\DB_Store_Chain.mdf;Integrated Security=True";
        SqlConnection sqlconn;
        SqlCommand sqlcomm;
        string Query;
        DataTable dt;
        SqlDataAdapter sqladapter;
        int ID = 0;

        private void DisplayData(string query)
        {
            sqlconn = new SqlConnection(cs);
            sqlcomm = new SqlCommand(query, sqlconn);
            sqladapter = new SqlDataAdapter();
            dt = new DataTable();
            sqladapter.SelectCommand = sqlcomm;
            sqladapter.Fill(dt);
            dataGridView1.DataSource = dt;
        }

        private void ClearData()
        {
            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            ID = 0;
        }


        private void button1_Click(object sender, EventArgs e)
        {
            switch (groupBox1.Controls.OfType<RadioButton>().FirstOrDefault(r =>
r.Checked).Name)
            {
                case "radioButton1":
                    Query = "SELECT c.first_name, c.last_name, p.purchase_date,
i.item, cat.category, i.price, ca.quantity" +
                        " FROM clients c" +
                        " JOIN purchases p ON c.id = p.clients_id" +
                        " JOIN carts ca ON p.id = ca.purchases_id" +
                        " JOIN items i ON ca.items_id = i.id" +
                        " JOIN categories cat ON i.categories_id = cat.id" +
                        " WHERE c.id = " + textBox1.Text +
                        " ORDER BY p.purchase_date ASC, i.item ASC;";
                    DisplayData(Query);
                    break;
                case "radioButton2":
```

```
                    Query = "SELECT p.id AS purchase_id, p.purchase_date, cl.id AS
client_id, cl.first_name AS client_first_name, cl.last_name AS client_last_name,
SUM(i.price * ct.quantity) AS total_price" +
                        " FROM purchases p" +
                        " JOIN clients cl ON p.clients_id = cl.id" +
                        " JOIN carts ct ON p.id = ct.purchases_id" +
                        " JOIN items i ON ct.items_id = i.id" +
                        " WHERE p.stores_id = " + textBox1.Text +
                        " GROUP BY p.purchase_date, p.id, cl.id, cl.first_name,
cl.last_name" +
                        " ORDER BY p.purchase_date;";
                    DisplayData(Query);
                    break;
                case "radioButton3":
                    Query = "SELECT p.id AS purchase_id, p.purchase_date, s.street AS
store_street, cl.id AS client_id, cl.first_name AS client_first_name, cl.last_name AS
client_last_name, SUM(i.price * ct.quantity) AS total_price" +
                        " FROM purchases p" +
                        " JOIN clients cl ON p.clients_id = cl.id" +
                        " JOIN carts ct ON p.id = ct.purchases_id" +
                        " JOIN items i ON ct.items_id = i.id" +
                        " JOIN stores s ON p.stores_id = s.id" +
                        " JOIN cities c ON s.cities_id = c.id" +
                        " WHERE c.city = \'" + textBox1.Text + "\'" +
                        " GROUP BY p.id, p.purchase_date, s.street, cl.id,
cl.first_name, cl.last_name" +
                        " ORDER BY p.purchase_date;";
                    DisplayData(Query);
                    break;
                case "radioButton4":
                    Query = "SELECT p.id AS purchase_id, p.purchase_date, cl.id AS
client_id, cl.first_name AS client_first_name, cl.last_name AS client_last_name,
SUM(i.price * ct.quantity) AS total_price, s.street AS store_street, c.city" +
                        " FROM purchases p" +
                        " JOIN clients cl ON p.clients_id = cl.id" +
                        " JOIN carts ct ON p.id = ct.purchases_id" +
                        " JOIN items i ON ct.items_id = i.id" +
                        " JOIN stores s ON p.stores_id = s.id" +
                        " JOIN cities c ON s.cities_id = c.id" +
                        " WHERE p.purchase_date BETWEEN \'" + textBox2.Text + "\' AND
\'" + textBox3.Text + "\'" +
                        " GROUP BY p.id, p.purchase_date, cl.id, cl.first_name,
cl.last_name, s.street, c.city" +
                        " ORDER BY p.purchase_date;";
                    DisplayData(Query);
                    break;
                case "radioButton5":
                    Query = "SELECT i.id AS item_id, i.item AS item_name, i.price AS
item_price, COUNT(ct.items_id) AS number_of_purchases" +
                        " FROM items i" +
                        " JOIN carts ct ON i.id = ct.items_id" +
                        " JOIN categories c ON i.categories_id = c.id" +
                        " WHERE c.category = \'" + textBox1.Text + "\'" +
                        " GROUP BY i.id, i.item, i.price" +
                        " ORDER BY number_of_purchases DESC;";
                    DisplayData(Query);
                    break;
            }
        }

        private void radioButton1_CheckedChanged(object sender, EventArgs e)
        {
```

```
            setTextboxes(false);
        }

        private void radioButton2_CheckedChanged(object sender, EventArgs e)
        {
            setTextboxes(false);
        }

        private void radioButton3_CheckedChanged(object sender, EventArgs e)
        {
            setTextboxes(false);
        }

        private void radioButton4_CheckedChanged(object sender, EventArgs e)
        {
            setTextboxes(true);
        }

        private void radioButton5_CheckedChanged(object sender, EventArgs e)
        {
            setTextboxes(false);

        }

        private void F_Main_Load(object sender, EventArgs e)
        {
            setTextboxes(false);
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Excel.Application xlApp = new Excel.Application();
            if (xlApp == null)
            {
                MessageBox.Show("Excel is not properly installed!!");
                return;
            }
            Excel.Workbook xlWorkBook;
            Excel.Worksheet xlWorkSheet;
            object misValue = System.Reflection.Missing.Value;
            xlWorkBook = xlApp.Workbooks.Add(misValue);
            xlWorkSheet = (Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
            for(int i = 0; i <dataGridView1.RowCount; i++)
            {
                for(int j = 0; j < dataGridView1.ColumnCount; j++)
                {
                    xlWorkSheet.Cells[i+1, j+1] = dataGridView1.Rows[i].Cells[j].Value
?? "";
                }
            }
            xlWorkBook.SaveAs("C:\\Users\\gmsokolov\\Desktop\\ExcelReport.xls",
Excel.XlFileFormat.xlWorkbookNormal,
                misValue, misValue, misValue, misValue,
Excel.XlSaveAsAccessMode.xlExclusive,
                misValue, misValue, misValue, misValue, misValue);
            xlWorkBook.Close(true, misValue, misValue);
            xlApp.Quit();
            MessageBox.Show("Excel file created , you can find the file
C:\\Users\\gmsokolov\\Desktop\\ExcelReport.xls");
        }
    }
}
```

F_Carts.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace DB_Georgi_Sokolov_21621397
{
    public partial class F_Carts : Form
    {
        public F_Carts()
        {
            InitializeComponent();
        }

        private void F_Carts_Load(object sender, EventArgs e)
        {
            this.cartsTableAdapter.Fill(this.dS_Carts.carts);
        }

        string cs = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\gmsokolov\source\repos\DB_Geor
gi_Sokolov_21621397\DB_Store_Chain.mdf;Integrated Security=True";
        SqlConnection sqlconn;
        SqlCommand sqlcomm;
        string Query;
        DataTable dt;
        SqlDataAdapter sqladapter;
        int ID = 0;

        private void DisplayData()
        {
            sqlconn = new SqlConnection(cs);
            Query = "SELECT * FROM Carts";
            sqlcomm = new SqlCommand(Query, sqlconn);
            sqladapter = new SqlDataAdapter();
            dt = new DataTable();
            sqladapter.SelectCommand = sqlcomm;
            sqladapter.Fill(dt);
            dataGridView1.DataSource = dt;
        }

        private void ClearData()
        {
            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            ID = 0;
        }

        private void dataGridView1_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
        {
            ID = Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value);
```

```
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
            textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                if (textBox1.Text != "" || textBox2.Text != "" || textBox3.Text != "")
                {
                    sqlconn = new SqlConnection(cs);
                    sqlconn.Open();
                    Query = "INSERT INTO Carts (quantity,items_id,purchases_id) VALUES
(@quantity,@items_id,@purchases_id)";
                    sqlcomm = new SqlCommand(Query, sqlconn);
                    sqlcomm.Parameters.AddWithValue("@quantity",
int.Parse(textBox1.Text));
                    sqlcomm.Parameters.AddWithValue("@items_id",
int.Parse(textBox2.Text));
                    sqlcomm.Parameters.AddWithValue("@purchases_id",
int.Parse(textBox3.Text));
                    sqlcomm.ExecuteNonQuery();
                    sqlconn.Close();
                    MessageBox.Show("Inserted");
                    DisplayData();
                    ClearData();
                }
                else throw new Exception();
            }
            catch
            {
                MessageBox.Show("Insert failed");
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            try
            {
                if (textBox1.Text != "" || textBox2.Text != "" || textBox3.Text != "")
                {
                    sqlconn = new SqlConnection(cs);
                    sqlconn.Open();
                    Query = "UPDATE Carts SET
quantity=@quantity,items_id=@items_id,purchases_id=@purchases_id WHERE ID=@ID";
                    sqlcomm = new SqlCommand(Query, sqlconn);
                    sqlcomm.Parameters.AddWithValue("@ID", ID);
                    sqlcomm.Parameters.AddWithValue("@quantity",
int.Parse(textBox1.Text));
                    sqlcomm.Parameters.AddWithValue("@items_id",
int.Parse(textBox2.Text));
                    sqlcomm.Parameters.AddWithValue("@purchases_id",
int.Parse(textBox3.Text));
                    sqlcomm.ExecuteNonQuery();
                    sqlconn.Close();
                    MessageBox.Show("Updated");
                    DisplayData();
                    ClearData();
                }
                else throw new Exception();
            }
```

```
            catch
            {
                MessageBox.Show("Update failed");
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
            if (ID != 0)
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "DELETE Carts WHERE ID = @ID";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@ID", ID);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Delted");
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Delete failed");
        }
    }
}
```

F_Categories.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace DB_Georgi_Sokolov_21621397
{
    public partial class F_Categories : Form
    {
        public F_Categories()
        {
            InitializeComponent();
        }

        private void Categories_Load(object sender, EventArgs e)
        {
            this.categoriesTableAdapter.Fill(this.database1DataSet1.categories);
        }
        string cs = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\gmsokolov\source\repos\DB_Geor
gi_Sokolov_21621397\DB_Store_Chain.mdf;Integrated Security=True";
        SqlConnection sqlconn;
        SqlCommand sqlcomm;
        string Query;
        DataTable dt;
        SqlDataAdapter sqladapter;
        int ID = 0;
```

```
        private void DisplayData()
        {
            sqlconn = new SqlConnection(cs);
            Query = "SELECT * FROM Categories";
            sqlcomm = new SqlCommand(Query, sqlconn);
            sqladapter = new SqlDataAdapter();
            dt = new DataTable();
            sqladapter.SelectCommand = sqlcomm;
            sqladapter.Fill(dt);
            dataGridView1.DataSource = dt;
        }

        private void ClearData()
        {
            textBox1.Text = "";
            ID = 0;
        }

        private void dataGridView1_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
        {
            ID = Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value);
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text != "") {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "INSERT INTO Categories (category) VALUES (@category)";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@category", textBox1.Text);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Inserted");
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Insert failed");
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (textBox1.Text!="")
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "UPDATE Categories SET category=@category WHERE ID=@ID";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@ID", ID);
                sqlcomm.Parameters.AddWithValue("@category", textBox1.Text);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Updated");
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Update failed");
        }
```

```
        private void button3_Click(object sender, EventArgs e)
        {
            if (ID != 0)
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "DELETE Categories WHERE ID = @ID";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@ID", ID);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Delted");
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Delete failed");
        }
    }
}
```

F_Cities.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace DB_Georgi_Sokolov_21621397
{
    public partial class F_Cities : Form
    {
        public F_Cities()
        {
            InitializeComponent();
        }

        private void Cities_Load(object sender, EventArgs e)
        {
            this.citiesTableAdapter.Fill(this.database1DataSet4.cities);
        }

        string cs = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\gmsokolov\source\repos\DB_Geor
gi_Sokolov_21621397\DB_Store_Chain.mdf;Integrated Security=True";
        SqlConnection sqlconn;
        SqlCommand sqlcomm;
        string Query;
        DataTable dt;
        SqlDataAdapter sqladapter;
        int ID = 0;

        private void DisplayData()
        {
            sqlconn = new SqlConnection(cs);
```

```
            Query = "SELECT * FROM Cities";
            sqlcomm = new SqlCommand(Query, sqlconn);
            sqladapter = new SqlDataAdapter();
            dt = new DataTable();
            sqladapter.SelectCommand = sqlcomm;
            sqladapter.Fill(dt);
            dataGridView1.DataSource = dt;
        }

        private void ClearData()
        {
            textBox1.Text = "";
            ID = 0;
        }

        private void dataGridView1_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
        {
            ID = Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value);
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text != "")
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "INSERT INTO Cities (city) VALUES (@city)";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@city", textBox1.Text);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Inserted");
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Insert failed");
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (textBox1.Text != "")
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "UPDATE Cities SET city=@city WHERE ID=@ID";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@ID", ID);
                sqlcomm.Parameters.AddWithValue("@city", textBox1.Text);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Updated");
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Update failed");
        }

        private void button3_Click(object sender, EventArgs e)
        {
            if (ID != 0)
```

```
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "DELETE Cities WHERE ID = @ID";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@ID", ID);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Delted");
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Delete failed");
        }
    }
}
```

F_Clients.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace DB_Georgi_Sokolov_21621397
{
    public partial class F_Clients : Form
    {
        public F_Clients()
        {
            InitializeComponent();
        }

        private void Clients_Load(object sender, EventArgs e)
        {
            this.clientsTableAdapter.Fill(this.database1DataSet3.clients);
        }

        string cs = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\gmsokolov\source\repos\DB_Geor
gi_Sokolov_21621397\DB_Store_Chain.mdf;Integrated Security=True";
        SqlConnection sqlconn;
        SqlCommand sqlcomm;
        string Query;
        DataTable dt;
        SqlDataAdapter sqladapter;
        int ID = 0;

        private void DisplayData()
        {
            sqlconn = new SqlConnection(cs);
            Query = "SELECT * FROM Clients";
            sqlcomm = new SqlCommand(Query, sqlconn);
```

```
            sqladapter = new SqlDataAdapter();
            dt = new DataTable();
            sqladapter.SelectCommand = sqlcomm;
            sqladapter.Fill(dt);
            dataGridView1.DataSource = dt;
        }

        private void ClearData()
        {
            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            ID = 0;
        }

        private void dataGridView1_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
        {
            ID = Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value);
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
            textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            if (textBox1.Text != "" || textBox2.Text != "" || textBox3.Text != "")
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "INSERT INTO Clients (first_name,last_name,phone) VALUES
(@first_name,@last_name,@phone)";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@first_name", textBox1.Text);
                sqlcomm.Parameters.AddWithValue("@last_name", textBox2.Text);
                sqlcomm.Parameters.AddWithValue("@phone", textBox3.Text);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Inserted");
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Insert failed");
        }

        private void button2_Click(object sender, EventArgs e)
        {
            if (textBox1.Text != "" || textBox2.Text != "" || textBox3.Text != "")
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "UPDATE Clients SET
first_name=@first_name,last_name=@last_name,phone=@phone WHERE ID=@ID";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@ID", ID);
                sqlcomm.Parameters.AddWithValue("@first_name", textBox1.Text);
                sqlcomm.Parameters.AddWithValue("@last_name", textBox2.Text);
                sqlcomm.Parameters.AddWithValue("@phone",textBox3.Text);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Updated");
                DisplayData();
```

```
                ClearData();
            }
            else MessageBox.Show("Update failed");
        }

        private void button3_Click(object sender, EventArgs e)
        {
            if (ID != 0)
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "DELETE Clients WHERE ID = @ID";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@ID", ID);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Delted");
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Delete failed");
        }
    }
}
```

F_Items.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace DB_Georgi_Sokolov_21621397
{
    public partial class F_Items : Form
    {
        public F_Items()
        {
            InitializeComponent();
        }

        private void Items_Load(object sender, EventArgs e)
        {
            this.itemsTableAdapter.Fill(this.database1DataSet2.items);
        }

        string cs = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\gmsokolov\source\repos\DB_Geor
gi_Sokolov_21621397\DB_Store_Chain.mdf;Integrated Security=True";
        SqlConnection sqlconn;
        SqlCommand sqlcomm;
        string Query;
        DataTable dt;
        SqlDataAdapter sqladapter;
        int ID = 0;
```

```
        private void DisplayData()
        {
            sqlconn = new SqlConnection(cs);
            Query = "SELECT * FROM Items";
            sqlcomm = new SqlCommand(Query, sqlconn);
            sqladapter = new SqlDataAdapter();
            dt = new DataTable();
            sqladapter.SelectCommand = sqlcomm;
            sqladapter.Fill(dt);
            dataGridView1.DataSource = dt;
        }

        private void ClearData()
        {
            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            ID = 0;
        }

        private void dataGridView1_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
        {
            ID = Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value);
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
            textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try {
                if (textBox1.Text != "" || textBox2.Text != "" || textBox3.Text != "")
                {
                    sqlconn = new SqlConnection(cs);
                    sqlconn.Open();
                    Query = "INSERT INTO Items (item,price,categories_id) VALUES
(@item,@price,@categories_id)";
                    sqlcomm = new SqlCommand(Query, sqlconn);
                    sqlcomm.Parameters.AddWithValue("@item", textBox1.Text);
                    sqlcomm.Parameters.AddWithValue("@price",
float.Parse(textBox2.Text));
                    sqlcomm.Parameters.AddWithValue("@categories_id",
int.Parse(textBox3.Text));
                    sqlcomm.ExecuteNonQuery();
                    sqlconn.Close();
                    MessageBox.Show("Inserted");
                    DisplayData();
                    ClearData();
                } else throw new Exception();
            } catch {
                MessageBox.Show("Insert failed");
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            try {
                if (textBox1.Text != "" || textBox2.Text != "" || textBox3.Text != "")
                {
                    sqlconn = new SqlConnection(cs);
```

```
                    sqlconn.Open();
                    Query = "UPDATE Items SET item=@item, price=@price,
categories_id=@categories_id WHERE ID=@ID";
                    sqlcomm = new SqlCommand(Query, sqlconn);
                    sqlcomm.Parameters.AddWithValue("@ID", ID);
                    sqlcomm.Parameters.AddWithValue("@item", textBox1.Text);
                    sqlcomm.Parameters.AddWithValue("@price",
float.Parse(textBox2.Text));
                    sqlcomm.Parameters.AddWithValue("@categories_id",
int.Parse(textBox3.Text));
                    sqlcomm.ExecuteNonQuery();
                    sqlconn.Close();
                    MessageBox.Show("Updated");
                    DisplayData();
                    ClearData();
                }
                else throw new Exception();
            } catch {
                MessageBox.Show("Update failed");
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
            if (ID != 0)
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "DELETE Items WHERE ID = @ID";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@ID", ID);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Delted");
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Delete failed");
        }
    }
}
```

F_Purchases.ch:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace DB_Georgi_Sokolov_21621397
{
    public partial class F_Purchases : Form
    {
        public F_Purchases()
        {
```

```
            InitializeComponent();
        }

        private void F_Purchases_Load(object sender, EventArgs e)
        {
            this.purchasesTableAdapter.Fill(this.dS_Purchases.purchases);
        }

        string cs = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\gmsokolov\source\repos\DB_Geor
gi_Sokolov_21621397\DB_Store_Chain.mdf;Integrated Security=True";
        SqlConnection sqlconn;
        SqlCommand sqlcomm;
        string Query;
        DataTable dt;
        SqlDataAdapter sqladapter;
        int ID = 0;

        private void DisplayData()
        {
            sqlconn = new SqlConnection(cs);
            Query = "SELECT * FROM Purchases";
            sqlcomm = new SqlCommand(Query, sqlconn);
            sqladapter = new SqlDataAdapter();
            dt = new DataTable();
            sqladapter.SelectCommand = sqlcomm;
            sqladapter.Fill(dt);
            dataGridView1.DataSource = dt;
        }

        private void ClearData()
        {
            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            ID = 0;
        }

        private void dataGridView1_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
        {
            ID = Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value);
            string[] date =
dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString().Split(' ');
            textBox1.Text = date[0];
            textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
            textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                if (textBox1.Text != "" || textBox2.Text != "" || textBox3.Text != "")
                {
                    sqlconn = new SqlConnection(cs);
                    sqlconn.Open();
                    Query = "INSERT INTO Purchases
(purchase_date,clients_id,stores_id) VALUES (@purchase_date,@clients_id,@stores_id)";
                    sqlcomm = new SqlCommand(Query, sqlconn);
                    sqlcomm.Parameters.AddWithValue("@purchase_date", textBox1.Text);
```

```
                    sqlcomm.Parameters.AddWithValue("@clients_id",
int.Parse(textBox2.Text));
                    sqlcomm.Parameters.AddWithValue("@stores_id",
int.Parse(textBox3.Text));
                    sqlcomm.ExecuteNonQuery();
                    sqlconn.Close();
                    MessageBox.Show("Inserted");
                    DisplayData();
                    ClearData();
                }
                else throw new Exception();
            }
            catch
            {
                MessageBox.Show("Insert failed");
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
            try
            {
                if (textBox1.Text != "" || textBox2.Text != "" || textBox3.Text != "")
                {
                    sqlconn = new SqlConnection(cs);
                    sqlconn.Open();
                    Query = "UPDATE Purchases SET
purchase_date=@purchase_date,clients_id=@clients_id,stores_id=@stores_id WHERE
ID=@ID";
                    sqlcomm = new SqlCommand(Query, sqlconn);
                    sqlcomm.Parameters.AddWithValue("@ID", ID);
                    sqlcomm.Parameters.AddWithValue("@purchase_date",textBox1.Text);
                    sqlcomm.Parameters.AddWithValue("@clients_id",
int.Parse(textBox2.Text));
                    sqlcomm.Parameters.AddWithValue("@stores_id",
int.Parse(textBox3.Text));
                    sqlcomm.ExecuteNonQuery();
                    sqlconn.Close();
                    MessageBox.Show("Updated");
                    DisplayData();
                    ClearData();
                }
                else throw new Exception();
            }
            catch
            {
                MessageBox.Show("Update failed");
            }
        }

        private void button3_Click(object sender, EventArgs e)
        {
            if (ID != 0)
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "DELETE Purchases WHERE ID = @ID";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@ID", ID);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Delted");
```

```
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Delete failed");
        }
    }
}
```

F_Stores.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace DB_Georgi_Sokolov_21621397
{
    public partial class F_Stores : Form
    {
        public F_Stores()
        {
            InitializeComponent();
        }

        private void F_Stores_Load(object sender, EventArgs e)
        {
            this.storesTableAdapter.Fill(this.dB_Store_ChainDataSet.stores);
        }

        string cs = @"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\gmsokolov\source\repos\DB_Geor
gi_Sokolov_21621397\DB_Store_Chain.mdf;Integrated Security=True";
        SqlConnection sqlconn;
        SqlCommand sqlcomm;
        string Query;
        DataTable dt;
        SqlDataAdapter sqladapter;
        int ID = 0;

        private void DisplayData()
        {
            sqlconn = new SqlConnection(cs);
            Query = "SELECT * FROM Stores";
            sqlcomm = new SqlCommand(Query, sqlconn);
            sqladapter = new SqlDataAdapter();
            dt = new DataTable();
            sqladapter.SelectCommand = sqlcomm;
            sqladapter.Fill(dt);
            dataGridView1.DataSource = dt;
        }

        private void ClearData()
        {
            textBox1.Text = "";
```

```
            textBox2.Text = "";
            ID = 0;
        }

        private void dataGridView1_RowHeaderMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
        {
            ID = Convert.ToInt32(dataGridView1.Rows[e.RowIndex].Cells[0].Value);
            textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
            textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                if (textBox1.Text != "" || textBox2.Text != "")
                {
                    sqlconn = new SqlConnection(cs);
                    sqlconn.Open();
                    Query = "INSERT INTO Stores (street,cities_id) VALUES
(@street,@cities_id)";
                    sqlcomm = new SqlCommand(Query, sqlconn);
                    sqlcomm.Parameters.AddWithValue("@street", textBox1.Text);
                    sqlcomm.Parameters.AddWithValue("@cities_id",
int.Parse(textBox2.Text));
                    sqlcomm.ExecuteNonQuery();
                    sqlconn.Close();
                    MessageBox.Show("Inserted");
                    DisplayData();
                    ClearData();
                }
                else throw new Exception();
            }
            catch
            {
                MessageBox.Show("Insert failed");
            }
        }

        private void button2_Click(object sender, EventArgs e)
        {
                try
                {
                    if (textBox1.Text != "" || textBox2.Text != "")
                    {
                        sqlconn = new SqlConnection(cs);
                        sqlconn.Open();
                        Query = "UPDATE Stores SET street=@street,
cities_id=@cities_id WHERE ID=@ID";
                        sqlcomm = new SqlCommand(Query, sqlconn);
                        sqlcomm.Parameters.AddWithValue("@ID", ID);
                        sqlcomm.Parameters.AddWithValue("@street", textBox1.Text);
                        sqlcomm.Parameters.AddWithValue("@cities_id",
int.Parse(textBox2.Text));
                        sqlcomm.ExecuteNonQuery();
                        sqlconn.Close();
                        MessageBox.Show("Updated");
                        DisplayData();
                        ClearData();
                    }
                    else throw new Exception();
```

```
                }
                catch
                {
                    MessageBox.Show("Update failed");
                }
            }

        private void button3_Click(object sender, EventArgs e)
        {
            if (ID != 0)
            {
                sqlconn = new SqlConnection(cs);
                sqlconn.Open();
                Query = "DELETE Stores WHERE ID = @ID";
                sqlcomm = new SqlCommand(Query, sqlconn);
                sqlcomm.Parameters.AddWithValue("@ID", ID);
                sqlcomm.ExecuteNonQuery();
                sqlconn.Close();
                MessageBox.Show("Delted");
                DisplayData();
                ClearData();
            }
            else MessageBox.Show("Delete failed");
        }
    }
}
```