**Database & Management Systems Lab**

***CSL 220***

***Assignment***

**Number # 04**

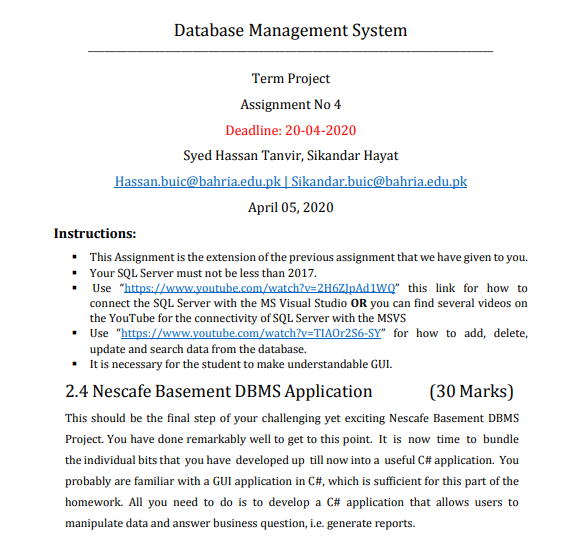


**MIAN AFZAAL ZAHOOR (094)**

**BS(CS)-4A**

**Department of Computer Sciences**

**BAHRIA UNIVERSITY, ISLAMABAD**



Contents

[1. Data Manipulation 4](#_heading=h.gjdgxs)

[I. Insert Data: 4](#_heading=h.30j0zll)

[II. Update Data: 4](#_heading=h.3znysh7)

[III. Delete Data: 5](#_heading=h.tyjcwt)

[IV. Display Data: 5](#_heading=h.1t3h5sf)

[Grid View Function for Displaying Records 6](#_heading=h.2s8eyo1)

[Commonly Used Buttons 7](#_heading=h.lnxbz9)

[2. Reports 7](#_heading=h.2jxsxqh)

[User Controls: 8](#_heading=h.1y810tw)

[Main Page 11](#_heading=h.qsh70q)

[Insert Button: 12](#_heading=h.49x2ik5)

[Update Button: 15](#_heading=h.1hmsyys)

[Delete Button: 18](#_heading=h.2u6wntf)

[Display Button: 20](#_heading=h.46r0co2)

[For each Business question (questions 1 to 10): 23](#_heading=h.2zbgiuw)

[Question 01: 24](#_heading=h.2dlolyb)

[Question 02: 25](#_heading=h.3cqmetx)

[Question 03: 26](#_heading=h.4bvk7pj)

[Question 04: 27](#_heading=h.1664s55)

[Question 05: 28](#_heading=h.25b2l0r)

[Question 06: 28](#_heading=h.34g0dwd)

[Question 07: 29](#_heading=h.43ky6rz)

[Question 08: 30](#_heading=h.xvir7l)

[Question 09: 31](#_heading=h.1x0gk37)

[Question 10: 32](#_heading=h.2w5ecyt)

**TABLE OF FIGURES**

[Figure 1 : PART I - INSERT DATA CODE 6](#_heading=h.1fob9te)

[Figure 2 : PART I - UPDATE DATA CODE 6](#_heading=h.2et92p0)

[Figure 3 : PART I - DELETE DATA CODE 7](#_heading=h.3dy6vkm)

[Figure 4 : PART I - DISPLAY DATA CODE 8](#_heading=h.4d34og8)

[Figure 5 : PART I - DISPLAY IN GRID FORM CODE (I) 8](#_heading=h.17dp8vu)

[Figure 6 : PART I - DISPLAY IN GRID FORM CODE (II) 8](#_heading=h.3rdcrjn)

[Figure 7 : PART I - EXAMPLE OF DISPLAY IN GRID FORM 9](#_heading=h.26in1rg)

[Figure 8 : PART I - COMMONLY USED EXIT TO MENU BUTTON 9](#_heading=h.35nkun2)

[Figure 9 : PART I - COMMONLY USED EXIT TO MENU BUTTON CODE 9](#_heading=h.1ksv4uv)

[Figure 10 : ACCESS DATA FROM DATABASE CODE 9](#_heading=h.44sinio)

[Figure 11 : PART II - EXAMPLE CODE 10](#_heading=h.z337ya)

[Figure 12 : PART II - WELCOME SCREEN 11](#_heading=h.3j2qqm3)

[Figure 13 : PART II - USER CONTROL (I) 12](#_heading=h.4i7ojhp)

[Figure 14 : PART II - USER CONTROL (II) 13](#_heading=h.2xcytpi)

[Figure 15 : PART II - USER CONTROL (III) 13](#_heading=h.1ci93xb)

[Figure 16 : PART II - CODE FOR HIDING IRRELEVENT USER CONTROLS (I) 14](#_heading=h.3whwml4)

[Figure 17 : PART II - CODE FOR HIDING IRRELEVENT USER CONTROLS (II) 14](#_heading=h.2bn6wsx)

[Figure 18 : PART II - CODE FOR RETURNING TO MAIN PAGE (I) 14](#_heading=h.3as4poj)

[Figure 19 : PART II - CODE FOR RETURNING TO MAIN PAGE (II) 15](#_heading=h.1pxezwc)

[Figure 20 : PART II - INSERT SCREEN 15](#_heading=h.2p2csry)

[Figure 21 : PART II - INSERT BUTTON CODE 16](#_heading=h.147n2zr)

[Figure 22 : PART II - INSERT INTO ADDRESS CODE 16](#_heading=h.3o7alnk)

[Figure 23 : PART II - INSERT SCREEN (I) 17](#_heading=h.23ckvvd)

[Figure 24 : PART II - INSERT SCREEN (II) 17](#_heading=h.ihv636)

[Figure 25 : PART II - INSERT SCREEN (III) 18](#_heading=h.32hioqz)

[Figure 26 : PART II - UPDATE SCREEN (I) 19](#_heading=h.41mghml)

[Figure 27 : PART II - UPDATE CODE 19](#_heading=h.2grqrue)

[Figure 28 : PART II - UPDATE SCREEN (II) 20](#_heading=h.vx1227)

[Figure 29 : PART II - UPDATE SCREEN (III) 20](#_heading=h.3fwokq0)

[Figure 30 : PART II - UPDATE SCREEN (IV) 21](#_heading=h.1v1yuxt)

[Figure 31 : PART II - UPDATE SCREEN (V) 22](#_heading=h.4f1mdlm)

[Figure 32 : PART II - DELETE SCREEN (I) 22](#_heading=h.19c6y18)

[Figure 33 : PART II - DELETE CODE 23](#_heading=h.3tbugp1)

[Figure 34 : PART II - DELETE SCREEN (II) 23](#_heading=h.28h4qwu)

[Figure 35 : PART II - DELETE SCREEN (III) 24](#_heading=h.nmf14n)

[Figure 36 : PART II - DELETE SCREEN (IV) 24](#_heading=h.37m2jsg)

[Figure 37 : PART II - DELETE SCREEN (V) 25](#_heading=h.1mrcu09)

[Figure 38 : PART II - DISPLAY SCREEN (I) 25](#_heading=h.2lwamvv)

[Figure 39 : PART II - DISPLAY CODE 26](#_heading=h.111kx3o)

[Figure 40 : PART II - DISPLAY SCREEN (II) 26](#_heading=h.3l18frh)

[Figure 41 : PART II - DISPLAY SCREEN (III) 27](#_heading=h.206ipza)

[Figure 42 : PART II - DISPLAY SCREEN (IV) 27](#_heading=h.4k668n3)

[Figure 43 : PART II - MENU FOR BUSINESS QUESTIONS 28](#_heading=h.1egqt2p)

[Figure 44 : PART II - CODE FOR MENU 28](#_heading=h.3ygebqi)

[Figure 45 : PART II - QUESTION 01 29](#_heading=h.sqyw64)

[Figure 46 : PART II - QUESTION 02 30](#_heading=h.1rvwp1q)

[Figure 47 : PART II - QUESTION 03 31](#_heading=h.2r0uhxc)

[Figure 48 : PART II - QUESTION 04 32](#_heading=h.3q5sasy)

[Figure 49 : PART II - QUESTION 05 33](#_heading=h.kgcv8k)

[Figure 50 : PART II - QUESTION 06 34](#_heading=h.1jlao46)

[Figure 51 : PART II - QUESTION 07 35](#_heading=h.2iq8gzs)

[Figure 52 : PART II - QUESTION 08 36](#_heading=h.3hv69ve)

[Figure 53 : PART II - QUESTION 09 37](#_heading=h.4h042r0)

[Figure 54 : PART II - QUESTION 10 38](#_heading=h.1baon6m)

**Your application should have the following functionality**

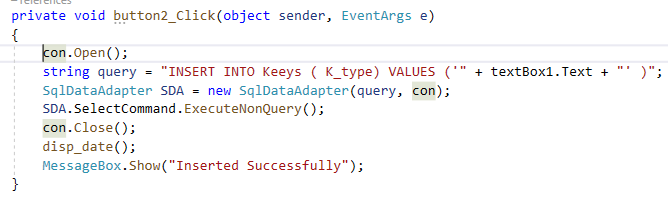
First of all, we will connect the SQL server with visual studio for the GUI of the application. And in below data manipulation the following functions are shown through key (Keey’s Table) as an example.

# Data Manipulation

* “con.open()” is used to open the connection between visual studio and SQL Server.
* Below queries are executed afterwards (Each function will have a different query).
* “SDA.SelectCommand.ExecuteNonQuery();” is used to check the errors in the query or the error in the inserted data by the user on runtime.

## Insert Data:

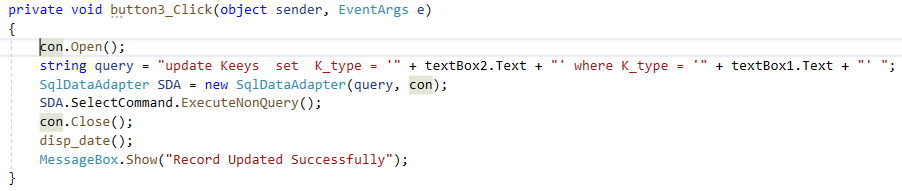




*Figure 1 : PART I - INSERT DATA CODE*

## Update Data:

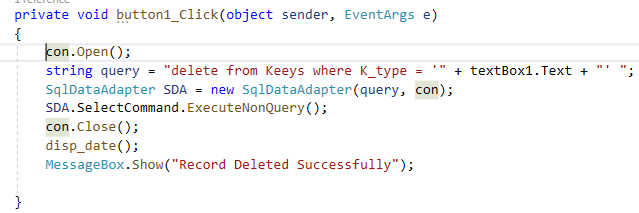




*Figure 2 : PART I - UPDATE DATA CODE*

## Delete Data:



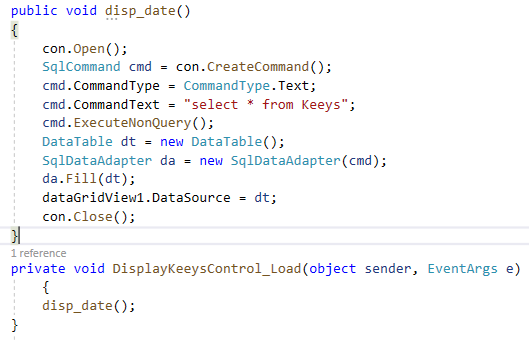


*Figure 3 : PART I - DELETE DATA CODE*

## Display Data:

This is the most commonly used button in the project. As the Display button is pressed the table will be appeared as it calls the grid view function whenever the user control is loaded.

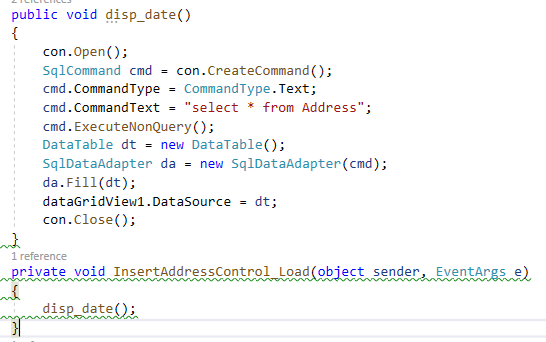




*Figure 4 : PART I - DISPLAY DATA CODE*

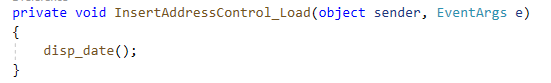
### Grid View Function for Displaying Records

The following function is used to display the table in grid view. By using this function, we can view the selected table in the database.



*Figure 5 : PART I - DISPLAY IN GRID FORM CODE (I)*

To show the grid view on screen below code is used.



*Figure 6 : PART I - DISPLAY IN GRID FORM CODE (II)*

**EXAMPLE**

Below is the example for the execution of above code for grid view display.



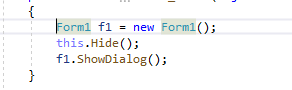
*Figure 7 : PART I - EXAMPLE OF DISPLAY IN GRID FORM*

# Commonly Used Buttons

This Button will open a new Form and hides the previously opened form.



*Figure 8 : PART I - COMMONLY USED EXIT TO MENU BUTTON*



*Figure 9 : PART I - COMMONLY USED EXIT TO MENU BUTTON CODE*

To connect and access the data from the database we have used the following code.

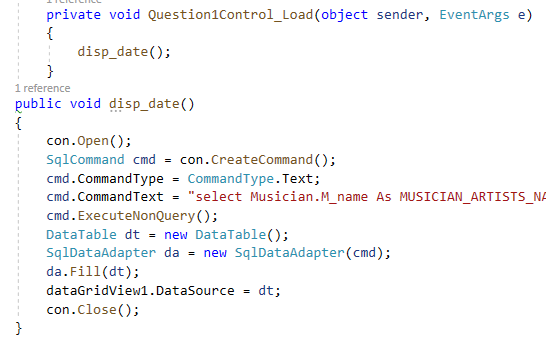


*Figure 10 : ACCESS DATA FROM DATABASE CODE*

This piece of code allows us to access the data in the databases and by using this we can easily add, modify or delete our data as we have given the path of our server and we have named the variable con for connection.

# Reports

Below are the reports for the following Actions performed.



*Figure 11 : PART II - EXAMPLE CODE*

We have created only one form in our assignment and this is our main page. On main page there are 5 buttons. And these buttons provide further control to other user control area. Many user controls are created in assignment each of the user control show its specific content which ever button is pressed the selected user control appears.



*Figure 12 : PART II - WELCOME SCREEN*

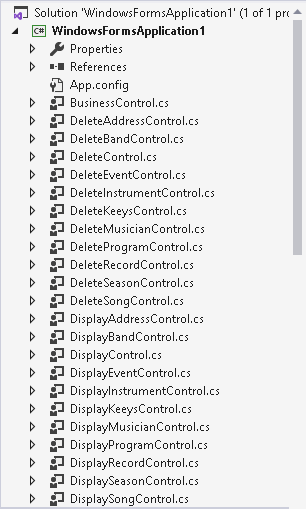
### User Controls:

These are all the user controls which are created for the assignment problems execution. There are 5 user controls for main page which are.

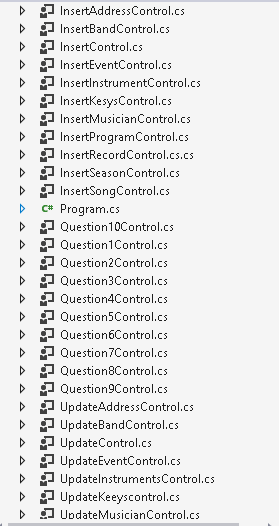
* + InsertControl
  + UpdateControl
  + DeleteControl
  + DisplayControl
  + BusinessControl

And there are 10 user controls in each of the above listed user control because we have 10 tables in our database and also there are 10 business questions:

* In **InsertControl** there are 10 user controls which are named as **Insert(Table Name)Control.**
* In **UpdateControl** there are 10 user controls which are named as **Update (Table Name)Control.**
* In **DeleteControl** there are 10 user controls which are named as **Delete(Table Name)Control.**
* In **DisplayControl** there are 10 user controls which are named as **Delete(Table Name)Control.**
* In **BusinessControl** there are 10 user controls which are named as **Question(number)Control.**



*Figure 13 : PART II - USER CONTROL (I)*

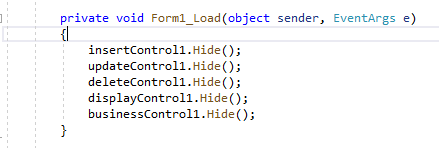


*Figure 14 : PART II - USER CONTROL (II)*



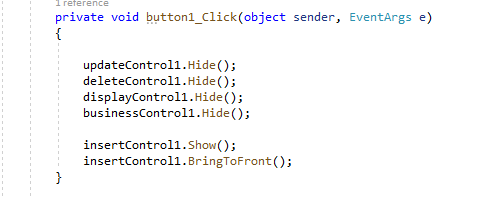
*Figure 15 : PART II - USER CONTROL (III)*

To show the selected user control and hide other user controls we have used following code.



*Figure 16 : PART II - CODE FOR HIDING IRRELEVENT USER CONTROLS (I)*

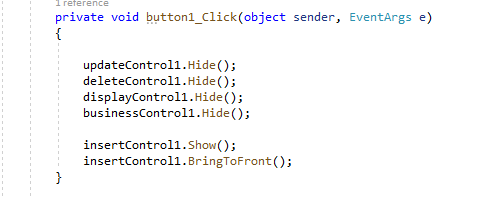
But when a specific button is pressed than to show its specific user control and hiding other user controls following code is used.



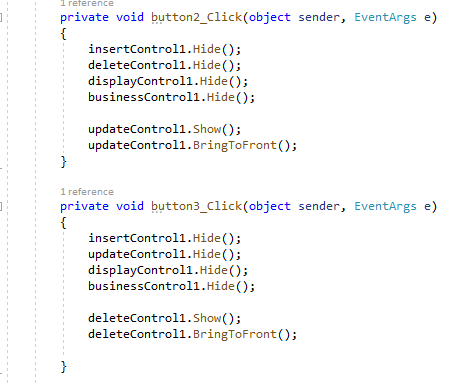
*Figure 17 : PART II - CODE FOR HIDING IRRELEVENT USER CONTROLS (II)*

## Main Page

Now below showing the information about the buttons on main page. Commonly all other buttons on main page work in the same way whenever the button is pressed its allotted user control appears on the screen.



*Figure 18 : PART II - CODE FOR RETURNING TO MAIN PAGE (I)*



*Figure 19 : PART II - CODE FOR RETURNING TO MAIN PAGE (II)*

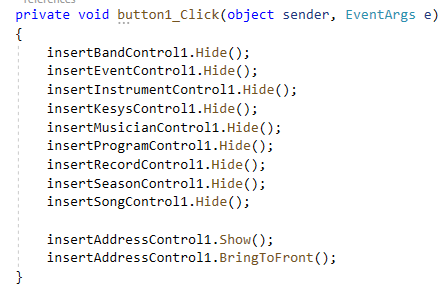
### Insert Button:



*Figure 20 : PART II - INSERT SCREEN*

All these buttons contain a user control on which the coding is done. From here the user can choose the desired table to which the user has to insert the data.

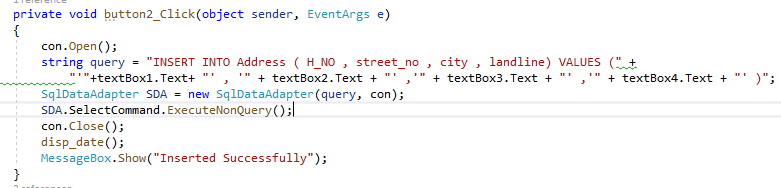
Following code is used to hide all other user controls and show a specific user control from insert menu.



*Figure 21 : PART II - INSERT BUTTON CODE*

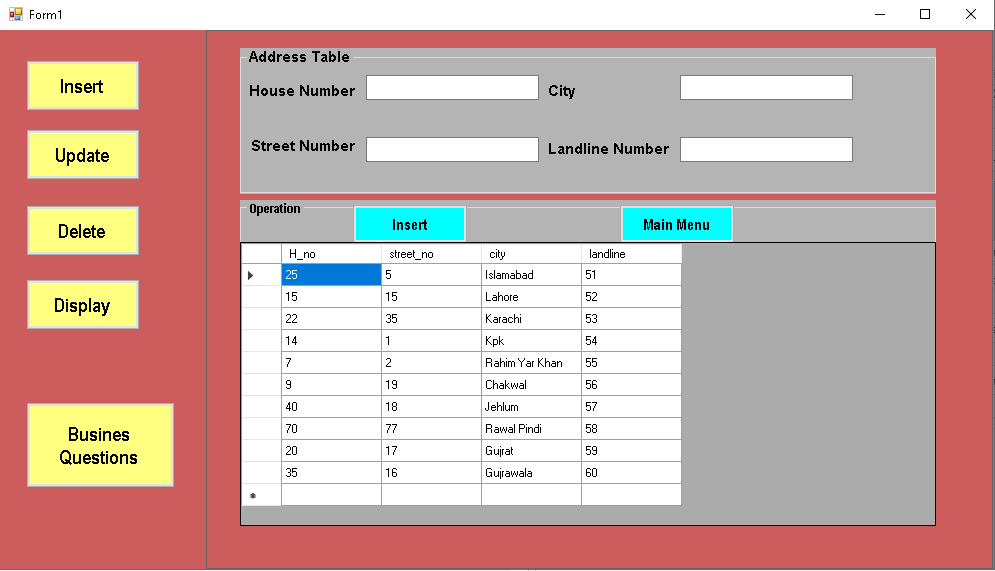
**Insert address:**

The code for the insert button is as follows.



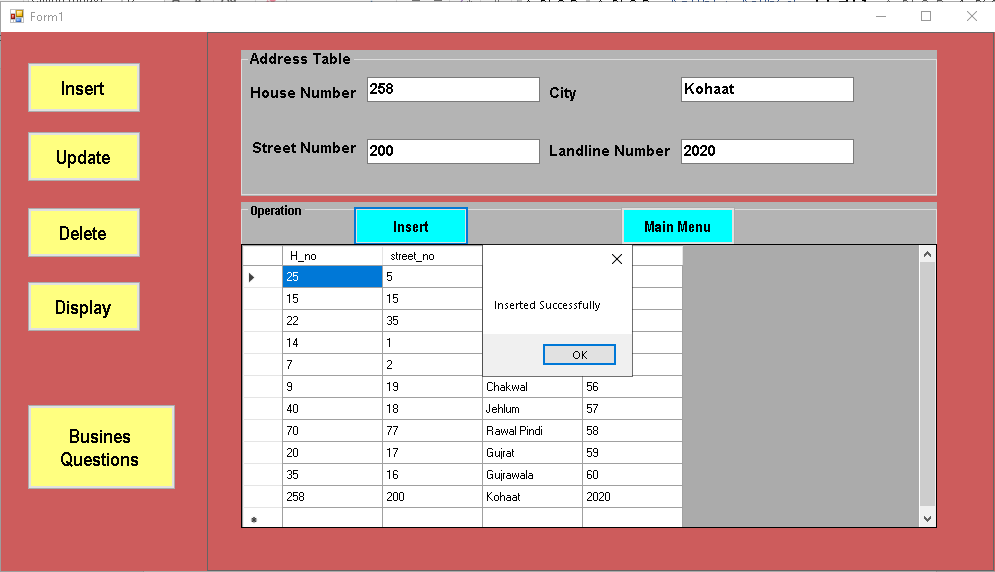
*Figure 22 : PART II - INSERT INTO ADDRESS CODE*

Main screen of insert shows inserted data as well as it displays the button to insert new data.



*Figure 23 : PART II - INSERT SCREEN (I)*

Whenever the data is inserted in the boxes and press the insert button than the data is inserted into the tables in the database and a dialog box appears which indicates that the data is inserted.

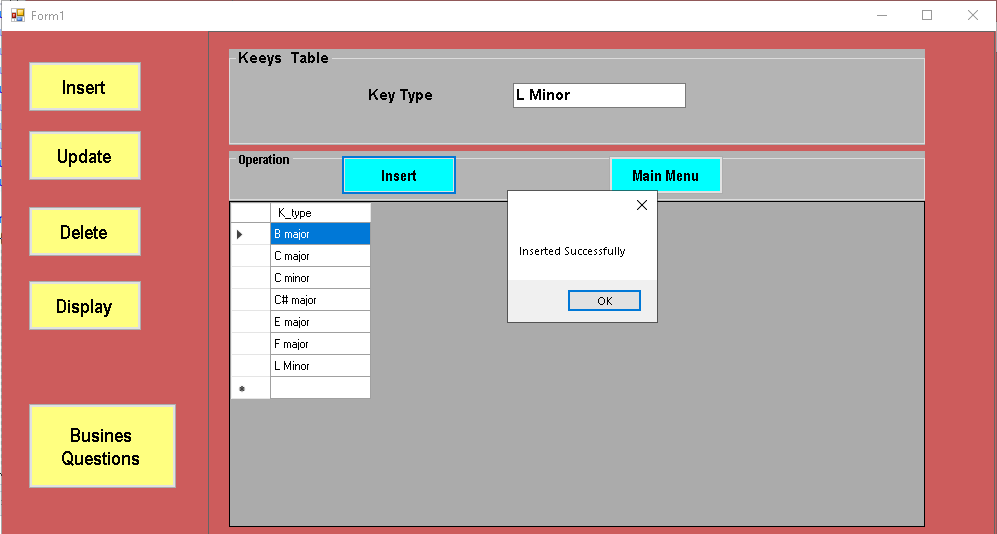


*Figure 24 : PART II - INSERT SCREEN (II)*

Following query is executed whenever and invalid data is entered so that invalid data cannot be entered in the database.

SDA.SelectCommand.ExecuteNonQuery();

**Insert Keeys:**



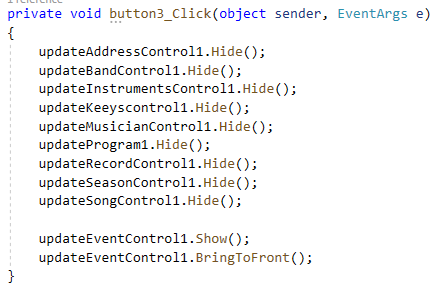
*Figure 25 : PART II - INSERT SCREEN (III)*

### Update Button:



*Figure 26 : PART II - UPDATE SCREEN (I)*

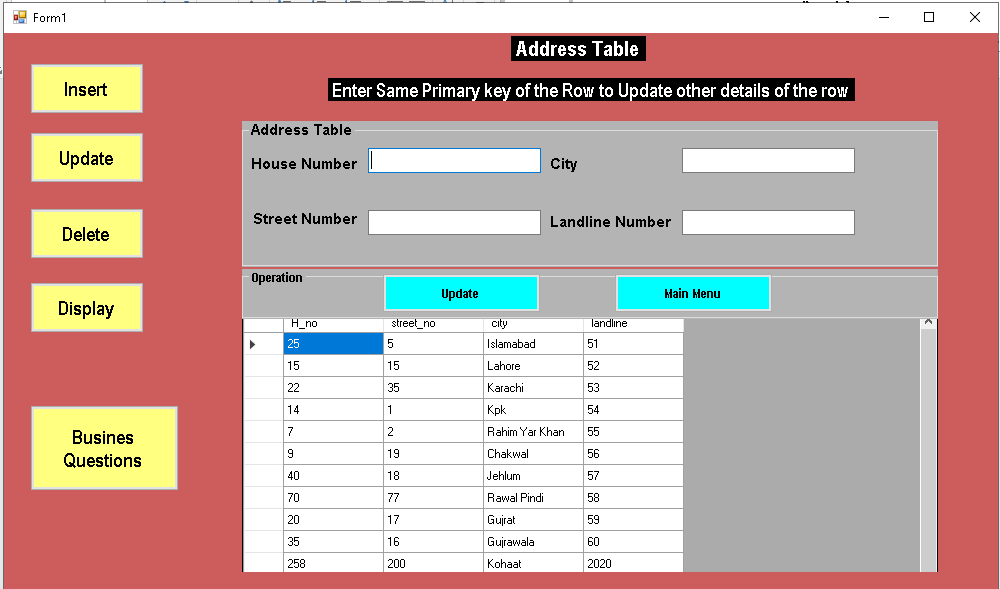
We can select any button and update its contents if we select insert button following screen appears and following code is used to hide all other user controls and show a specific user control from insert menu when it is clicked.



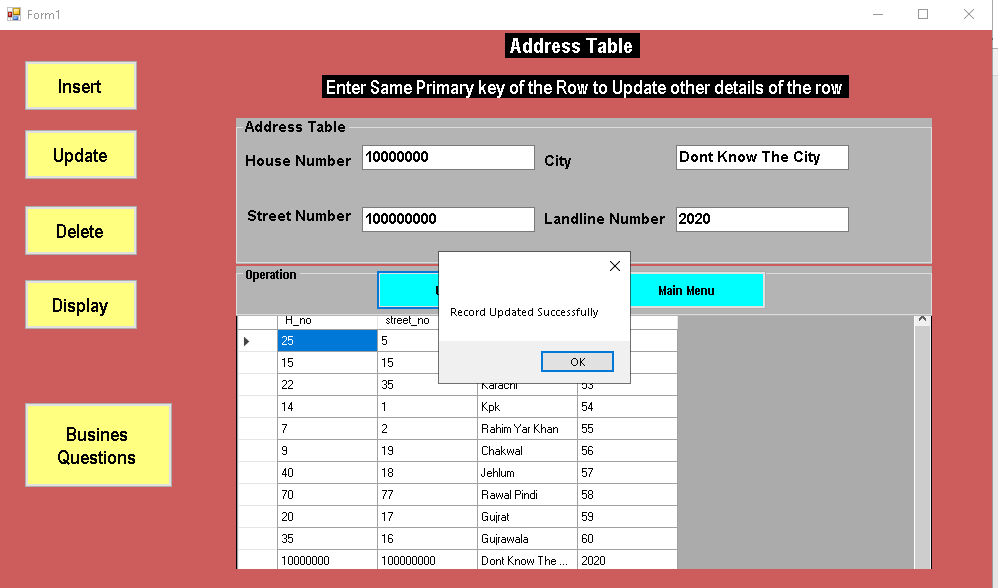
*Figure 27 : PART II - UPDATE CODE*

**Update Address:**

We will have to enter the same primary key from the table to update other details of the specific row we will select the last row to be updated.

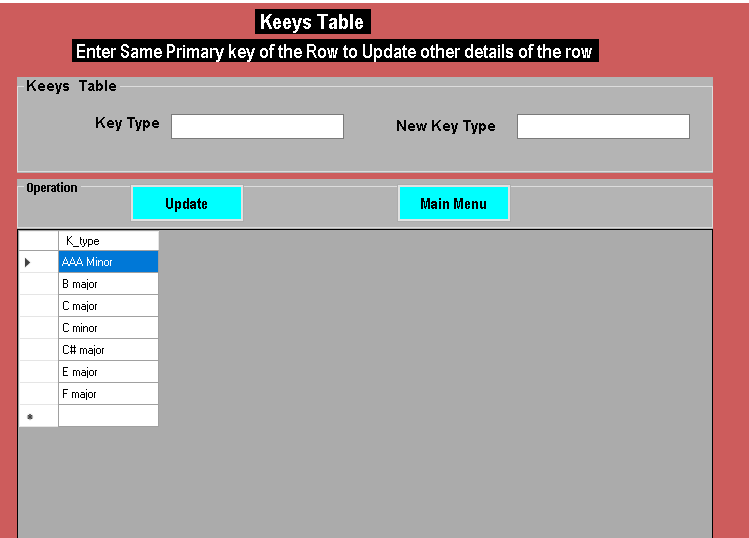


*Figure 28 : PART II - UPDATE SCREEN (II)*

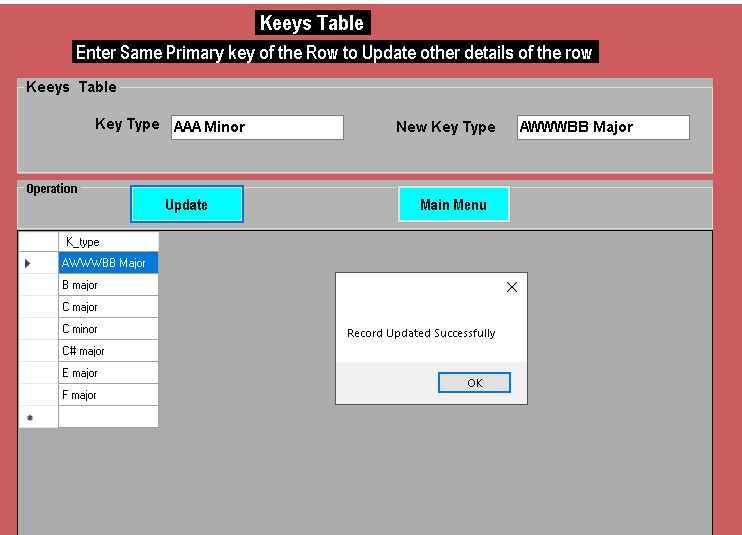


*Figure 29 : PART II - UPDATE SCREEN (III)*

**Update Keeys:**



*Figure 30 : PART II - UPDATE SCREEN (IV)*



*Figure 31 : PART II - UPDATE SCREEN (V)*

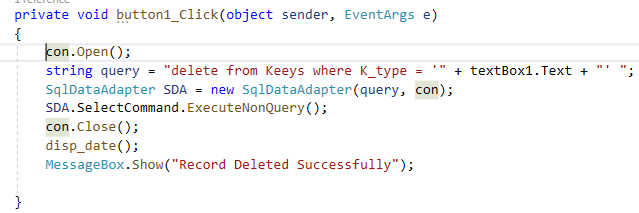
### Delete Button:

Now we can choose any button to delete data from the selected table.



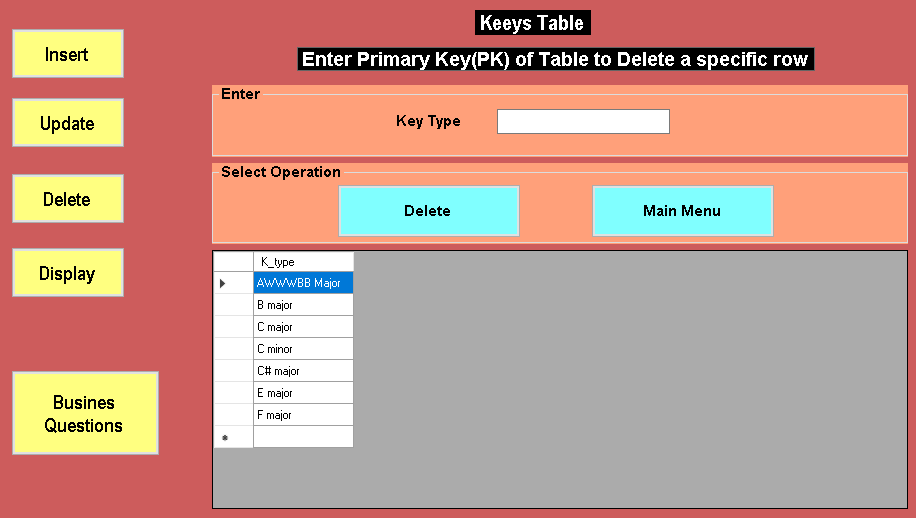
*Figure 32 : PART II - DELETE SCREEN (I)*

To Delete any row from The table we have to enter its primary key. And code for this delete button is:

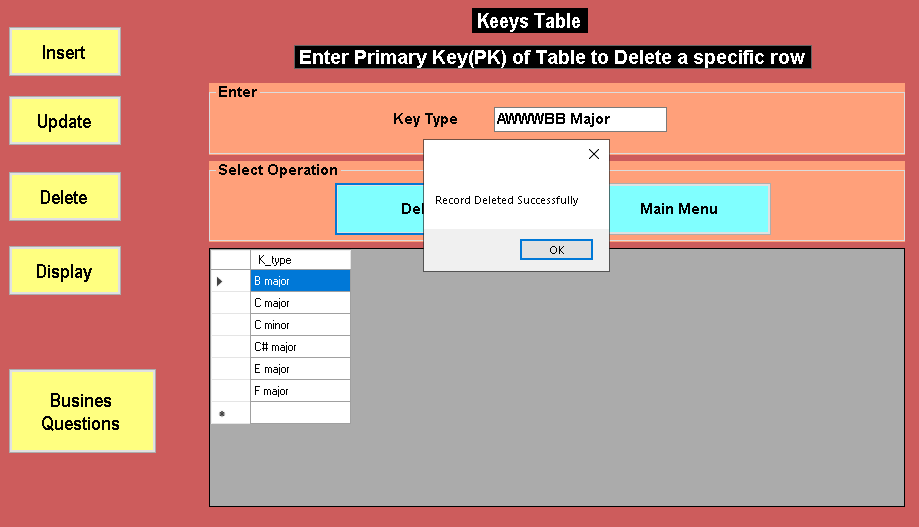


*Figure 33 : PART II - DELETE CODE*

**Delete from Keeys:**

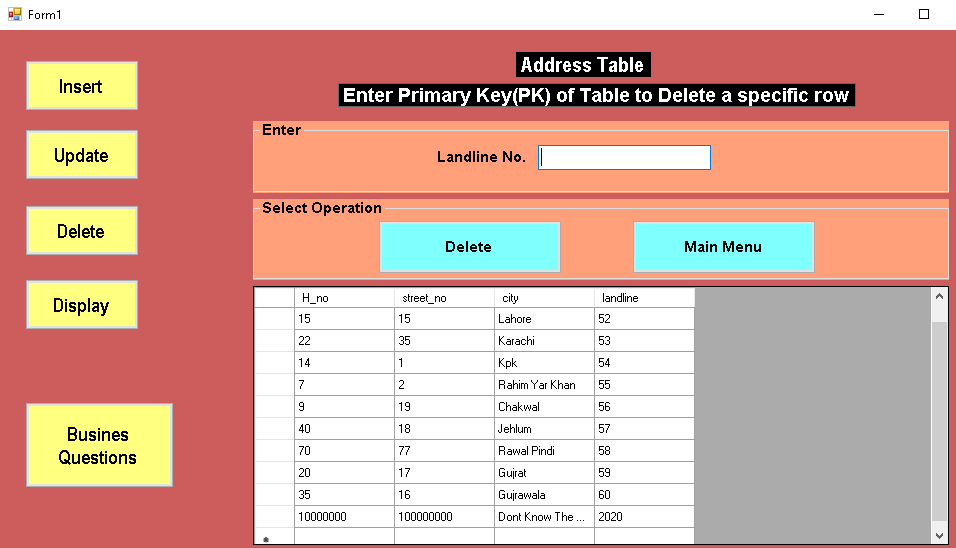


*Figure 34 : PART II - DELETE SCREEN (II)*

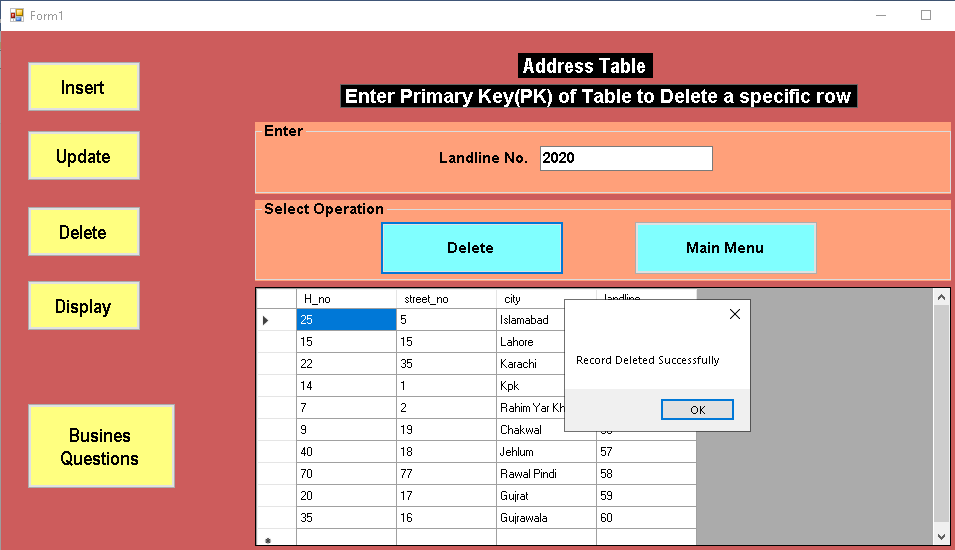


*Figure 35 : PART II - DELETE SCREEN (III)*

**Delete from Address:**



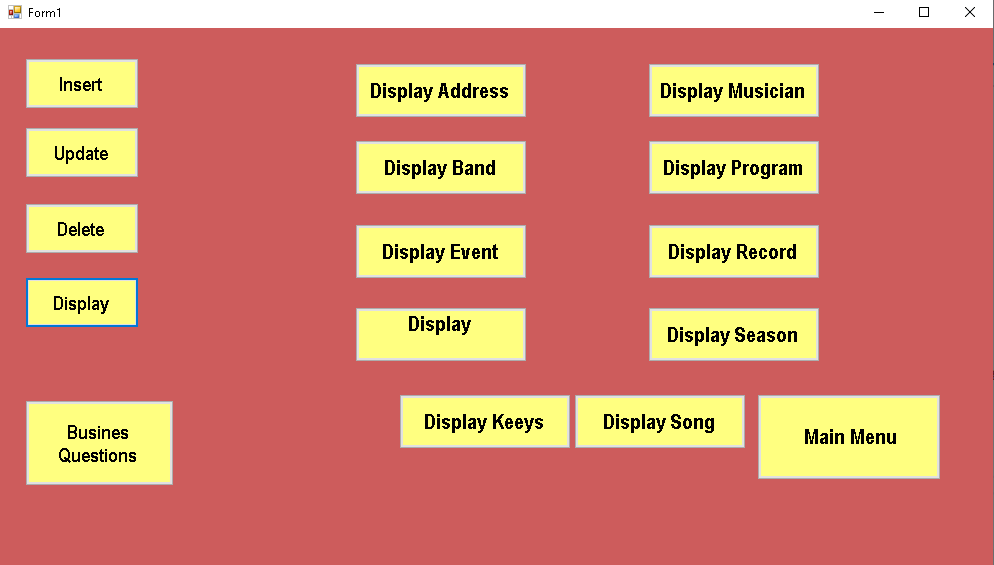
*Figure 36 : PART II - DELETE SCREEN (IV)*



*Figure 37 : PART II - DELETE SCREEN (V)*

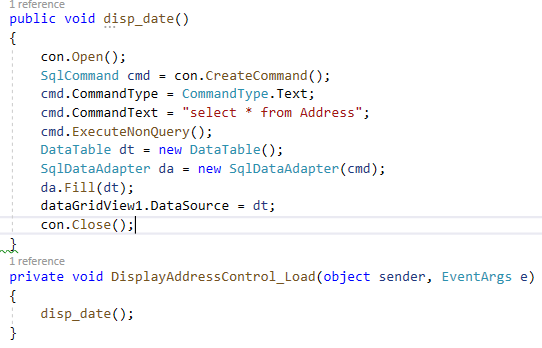
### Display Button:

Whenever Display option is selected from the main menu following screen appears from which we can select the table which we want to view.



*Figure 38 : PART II - DISPLAY SCREEN (I)*

As there is no button after clicking on display (specific) Table so the displaying function is called whenever the user control is loaded.



*Figure 39 : PART II - DISPLAY CODE*

**Display Address:**



*Figure 40 : PART II - DISPLAY SCREEN (II)*

**Display Records:**



*Figure 41 : PART II - DISPLAY SCREEN (III)*

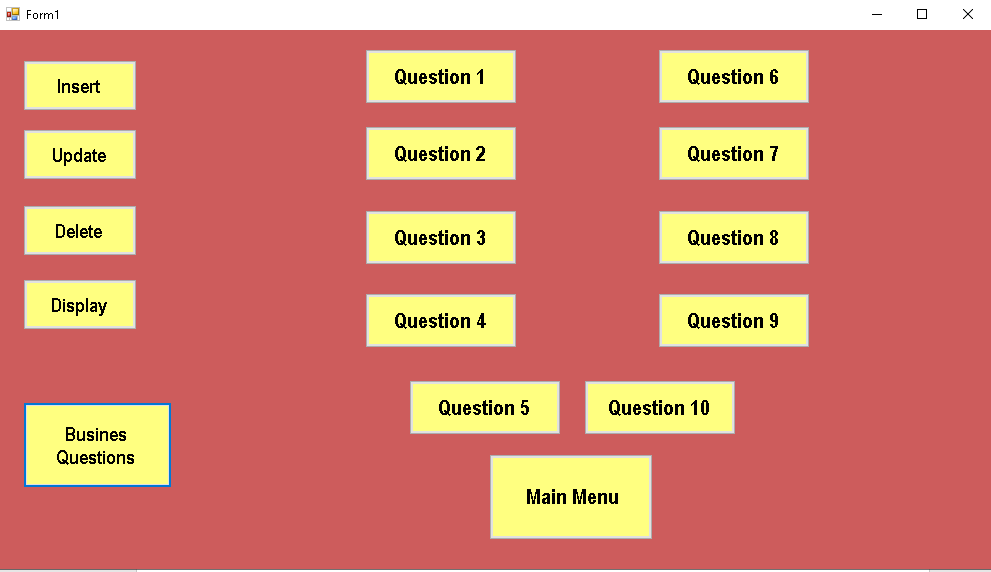
**Display Song:**



*Figure 42 : PART II - DISPLAY SCREEN (IV)*

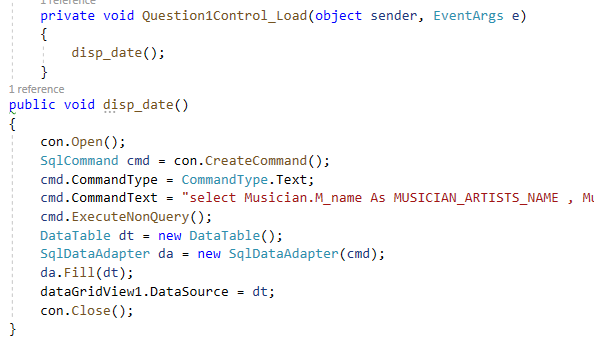
## For each Business question (questions 1 to 10):

If we select Business Question Button from the main page than the following screen appears on the screen.



*Figure 43 : PART II - MENU FOR BUSINESS QUESTIONS*

As clicking on question show the result when the page is loaded so its code is called when the question page is loaded. There is same code for all the questions and its grid view but the query is different for all the questions.



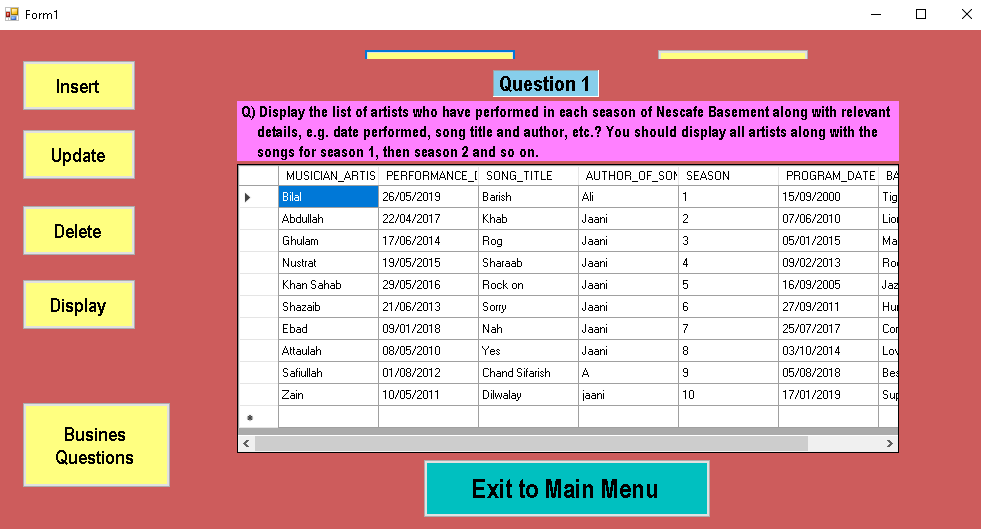
*Figure 44 : PART II - CODE FOR MENU*

### Question 01:

**QUERY**

select Musician.M\_name as MUSICIAN/ARTISTS\_NAME, Musician.pro\_date as PERFORMANCE\_DATE, Song.Title as SONG\_TITLE, Song.Author as AUTHOR\_OF\_SONG, Season.S\_id as SEASON, Program.copyright\_date as PROGRAM\_DATE, Band.Band\_name as BAND\_NAME from Musician inner join Song on Song.Musician\_ID = Musician.M\_id inner join SEASON on Season.S\_id = Musician.M\_id inner join Program on Program.season = Season.S\_id inner join Band on Band.Band\_id = Program.Pro\_id order by S\_id asc;

**SCREENSHOT**



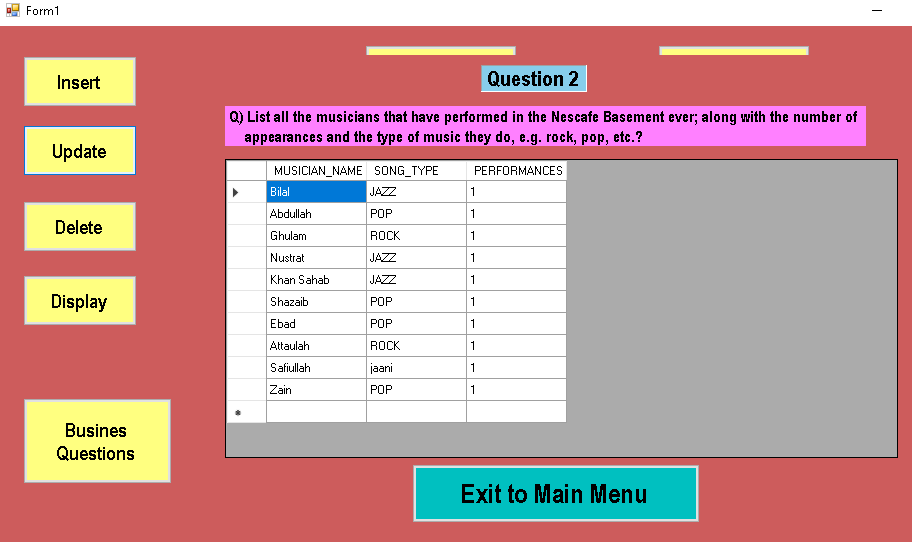
*Figure 45 : PART II - QUESTION 01*

### Question 02:

**QUERY**

select Musician.M\_name as MUSICIAN\_NAME, Song.S\_type as SONG\_TYPE, count(Musician.M\_id) over (partition by [M\_id]) as PERFORMANCES from Musician inner join Song on Song.Musician\_ID = Musician.M\_id;

**SCREENSHOT**



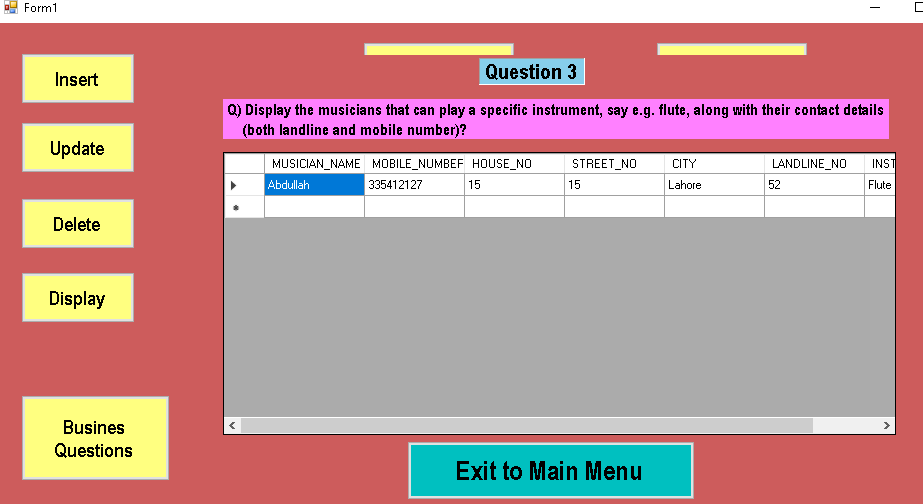
*Figure 46 : PART II - QUESTION 02*

### Question 03:

**QUERY**

select Musician.M\_name as MUSICIAN\_NAME, Musician.Mob\_no as MOBILE\_NUMBER, Address.H\_no as HOUSE\_NO, Address.street\_no as STREET\_NO, Address.city as CITY, Address.landline as LANDLINE\_NO, Instrument.I\_type as INSTRUMENT from Musician inner join Address on Address.landline = Musician.Landline inner join Instrument on Instrument.I\_no = Musician.M\_id where Instrument.I\_type = 'Flute';

**SCREENSHOT**



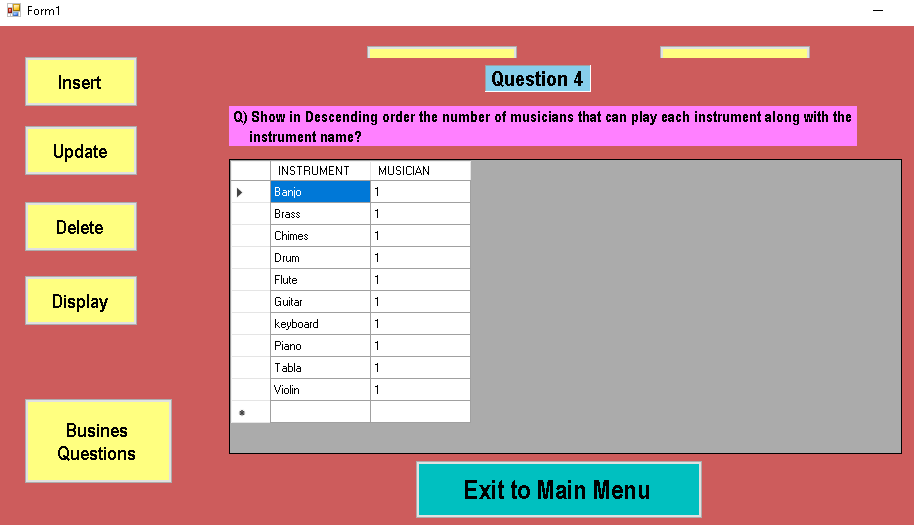
*Figure 47 : PART II - QUESTION 03*

### Question 04:

**QUERY**

select Instrument.I\_type as INSTRUMENT, count(\*) as MUSICIAN from Musician inner join Instrument on instrument.I\_no = Musician.M\_id group by I\_type order by count(I\_type) desc;

**SCREENSHOT**



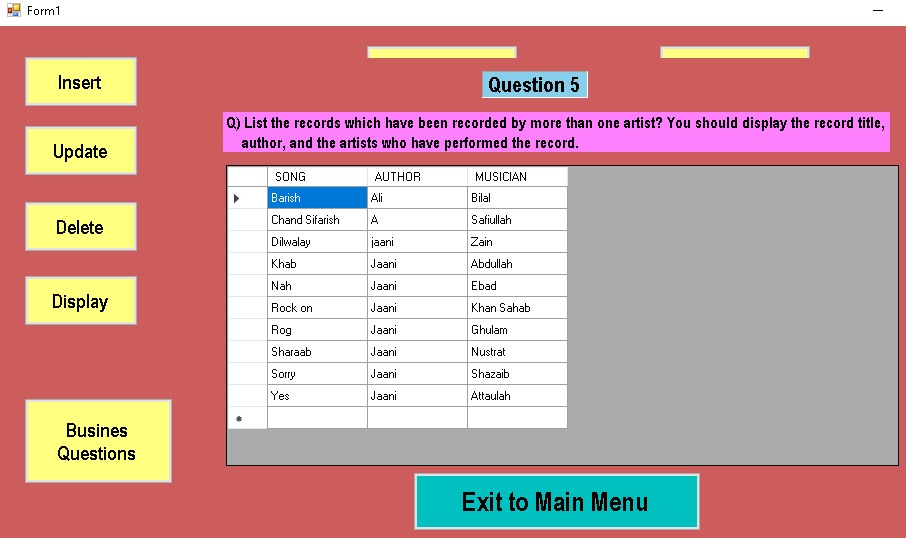
*Figure 48 : PART II - QUESTION 04*

### Question 05:

**QUERY**

select Song.Title as SONG, Song.Author as AUTHOR, Musician.M\_name as MUSICIAN from Musician inner join Song on Song.Musician\_ID = Musician.M\_id where Title in (select Title from Song group by Title having count (\*) > 1;

**SCREENSHOT**



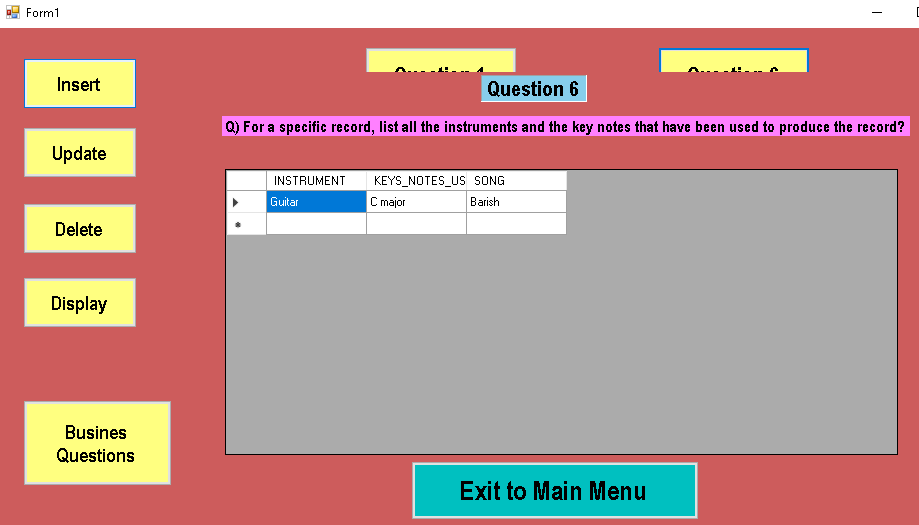
*Figure 49 : PART II - QUESTION 05*

### Question 06:

**QUERY**

select Instrument.I\_type as INSTRUMENT, Song.keys as KEYS\_NOTES\_USED, Song.Title as SONG from Instrument inner join Song on Song.Musician\_ID = Instrument.I\_no where Song.Title = 'Barish';

**SCREENSHOT**



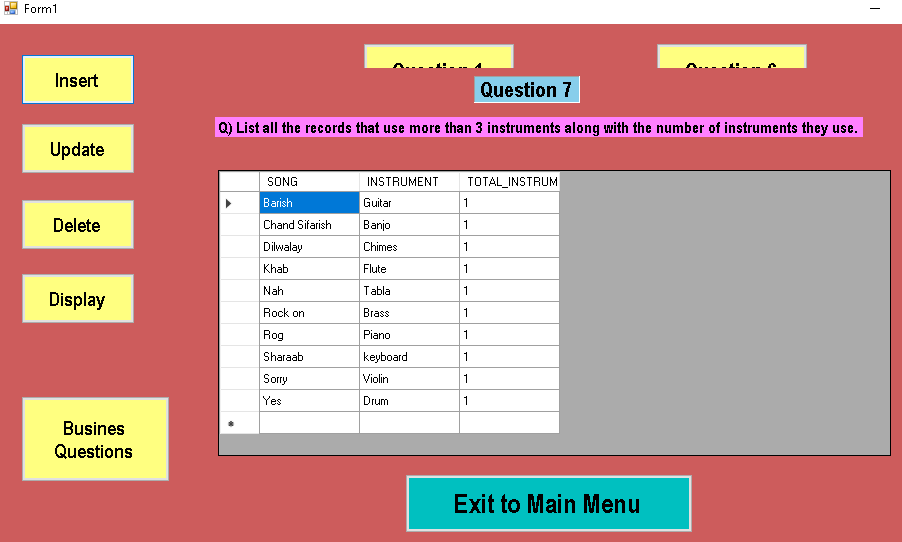
*Figure 50 : PART II - QUESTION 06*

### Question 07:

**QUERY**

select Song.Title as SONG, Instrument.I\_type as INSTRUMENT, ( len(Instrument.I\_type) - len(replace(Instrument.I\_type,',',' ')) + 1) as TOTAL\_INSTRUMENTS from Song inner join Instrument on Instrument.I\_no = Song.Musician\_ID where ( len(Instrument.I\_type) - len(replace(Instrument.I\_type,',',' ')) + 1) > 3;

**SCREENSHOT**



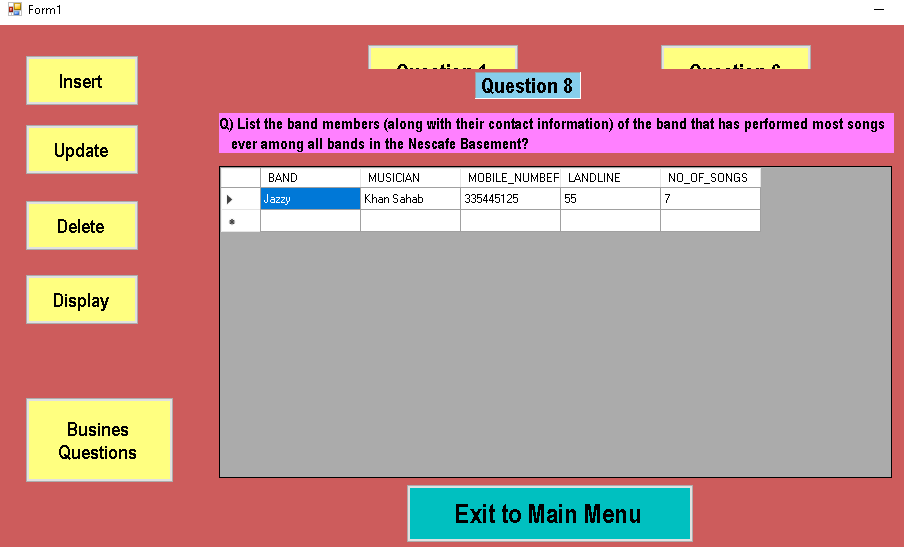
*Figure 51 : PART II - QUESTION 07*

### Question 08:

**QUERY**

select Band.Band\_name as BAND, Musician.M\_name as MUSICIAN, Musician.Mob\_no as MOBILE\_NUMBER, Musician.Landline as LANDLINE, Program.no\_of\_musc as NO\_OF\_SONGS from Musician inner join Band on Band.Band\_id = Musician.M\_id inner join Program on Program.Pro\_id = Musician.M\_id where Program.no\_of\_musc in (select max(Program.no\_of\_musc) from Program);

**SCREENSHOT**



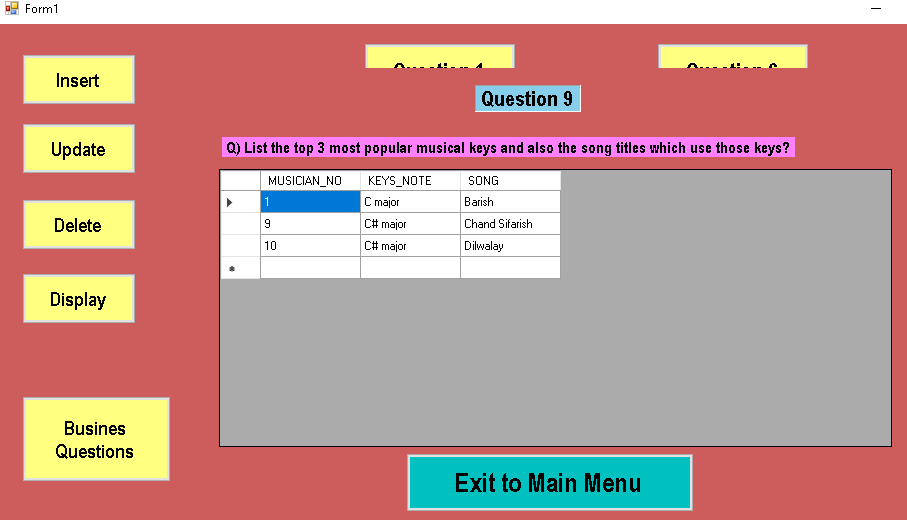
*Figure 52 : PART II - QUESTION 08*

### Question 09:

**QUERY**

select TOP 3 Musician.M\_id as MUSICIAN\_NO, Song.keys as KEYS\_NOTE, Song.Title as SONG from Song inner join Musician on Musician.M\_id = Song.Musician\_ID;

**SCREENSHOT**



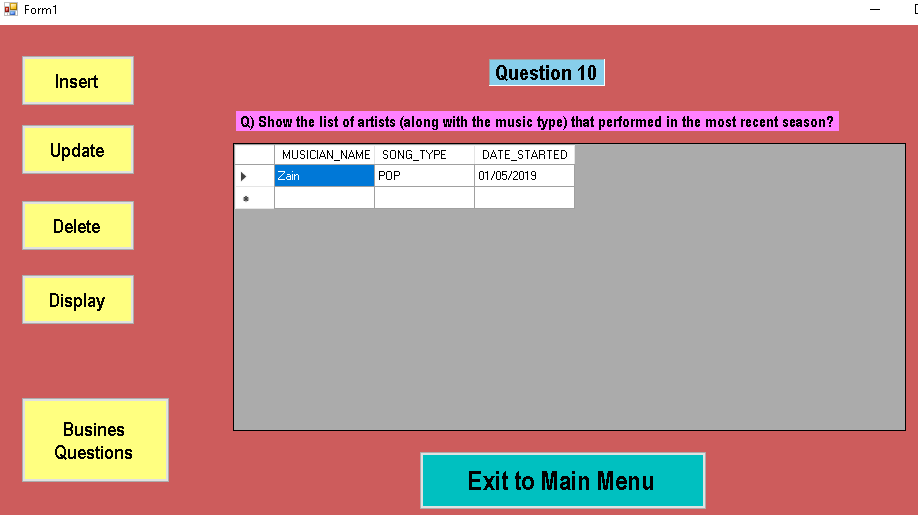
*Figure 53 : PART II - QUESTION 09*

### Question 10:

**QUERY**

select Musician.M\_name as MUSICIAN\_NAME, Song.S\_type as SONG\_TYPE, Season.S\_date as DATE\_STARTED from Musician inner join Song on Song.Musician\_ID = Musician.M\_id inner join Season on Season.S\_id = Musician.M\_id where Season.S\_date = (select max(S\_date) from Season) and Season.S\_id in (select max(S\_id) from Season);

**SCREENSHOT**



*Figure 54 : PART II - QUESTION 10*

**Link of ZIP file that include code and MSVS files of program.**

<https://drive.google.com/open?id=1D0hvnQQEA4SzoPbBUbjuOY--f1Sojele>