**Source code:**

**SQL-Schema:**

DROP SCHEMA IF EXISTS cricket;

CREATE SCHEMA cricket;

USE cricket;

--

-- Table structure for all tables in cricket schema`

--

CREATE TABLE team (

team\_id NUMERIC,

team\_country VARCHAR (20) UNIQUE,

team\_rank NUMERIC,

team\_captain VARCHAR (30),

matches\_played NUMERIC,

matches\_win NUMERIC,

matches\_loss NUMERIC,

matches\_tie NUMERIC,

PRIMARY KEY (team\_id)

);

CREATE TABLE coach(

coach\_id NUMERIC,

coach\_name VARCHAR(30),

coach\_country VARCHAR (20),

coach\_dob date,

join\_date date,

team\_id NUMERIC ,

PRIMARY KEY (coach\_id),

CONSTRAINT `fk\_coach\_team\_id` FOREIGN KEY (team\_id) REFERENCES team (team\_id) on delete set null on update cascade

);

CREATE TABLE series(

series\_id NUMERIC,

series\_name VARCHAR(30),

man\_of\_series VARCHAR(30),

series\_winner VARCHAR(30),

start\_date date,

end\_date date,

PRIMARY KEY (series\_id)

);

CREATE TABLE stadium(

stadium\_id NUMERIC,

stadium\_name VARCHAR(20),

capacity NUMERIC,

city VARCHAR(15),

country VARCHAR(20),

PRIMARY KEY (stadium\_id)

);

CREATE TABLE umpire(

umpire\_id NUMERIC,

umpire\_name VARCHAR(30),

umpire\_country VARCHAR(20),

umpire\_dob date,

matches\_umpired NUMERIC,

PRIMARY KEY (umpire\_id)

);

CREATE TABLE matches(

match\_id NUMERIC,

toss\_win VARCHAR(20),

batting\_first VARCHAR(20),

man\_of\_match VARCHAR(20),

result VARCHAR(20),

match\_type VARCHAR(20),

match\_date date,

stadium\_id NUMERIC,

series\_id NUMERIC,

umpire\_id NUMERIC,

PRIMARY KEY (match\_id),

CONSTRAINT `fk\_matches\_umpire\_id` FOREIGN KEY (umpire\_id) REFERENCES umpire (umpire\_id) on delete set null on update cascade,

CONSTRAINT `fk\_matches\_stadium\_id` FOREIGN KEY (stadium\_id) REFERENCES stadium (stadium\_id) on delete set null on update cascade,

CONSTRAINT `fk\_matches\_series\_id` FOREIGN KEY (series\_id) REFERENCES series (series\_id) on delete cascade on update cascade

);

CREATE TABLE scorecard(

scorecard\_id NUMERIC,

run\_rate float,

total\_runs NUMERIC,

total\_wickets NUMERIC,

total\_overs float,

extras NUMERIC,

inning NUMERIC,

team\_id NUMERIC,

match\_id NUMERIC,

PRIMARY KEY (scorecard\_id),

CONSTRAINT `fk\_scorecard\_match\_id` FOREIGN KEY (match\_id) REFERENCES matches (match\_id) on delete cascade on update cascade,

CONSTRAINT `fk\_scorecard\_team\_id` FOREIGN KEY (team\_id) REFERENCES team (team\_id) on delete set null on update cascade

);

CREATE TABLE player(

player\_id NUMERIC,

player\_name VARCHAR(30),

player\_dob date,

player\_height NUMERIC,

debut\_date date,

team\_id NUMERIC,

PRIMARY KEY(player\_id),

CONSTRAINT `fk\_player\_team\_id` FOREIGN KEY (team\_id) REFERENCES team (team\_id) on delete cascade on update cascade

);

CREATE table squad(

player\_id NUMERIC,

team\_id NUMERIC,

CONSTRAINT `fk\_squad\_team\_id` FOREIGN KEY (team\_id) REFERENCES team (team\_id) on delete cascade,

CONSTRAINT `fk\_squad\_player\_id` FOREIGN KEY (player\_id) REFERENCES player (player\_id) on delete cascade

);

CREATE TABLE player\_statistics(

p\_statistics\_id NUMERIC,

player\_id NUMERIC,

runs NUMERIC,

sixers NUMERIC,

fours NUMERIC,

overs\_bowled numeric,

wkts\_taken NUMERIC,

maidens NUMERIC,

match\_id NUMERIC,

scorecard\_id NUMERIC,

PRIMARY KEY (p\_statistics\_id),

CONSTRAINT `fk\_player\_statistics\_scorecard\_id` FOREIGN KEY (scorecard\_id) REFERENCES scorecard (scorecard\_id) on delete set null on update cascade,

CONSTRAINT `fk\_player\_statistics\_player\_id` FOREIGN KEY (player\_id) REFERENCES player (player\_id) on delete cascade on update cascade

);

CREATE TABLE bowler(

player\_id NUMERIC unique,

bowler\_rank NUMERIC,

wickets NUMERIC,

best\_figures NUMERIC,

balls\_bowled NUMERIC,

fivers NUMERIC,

CONSTRAINT `fk\_bowler\_player\_id` FOREIGN KEY (player\_id) REFERENCES player (player\_id) on delete cascade on update cascade

);

CREATE TABLE batsman(

player\_id NUMERIC UNIQUE,

batting\_rank NUMERIC,

total\_runs NUMERIC,

balls\_faced NUMERIC,

style VARCHAR (10),

hundreds NUMERIC,

fifties NUMERIC,

total\_sixes NUMERIC,

total\_fours NUMERIC,

CONSTRAINT `fk\_batsman\_player\_id` FOREIGN KEY (player\_id) REFERENCES player (player\_id) on delete cascade on update cascade

);

**SQL-data:**

insert into team

values(101, 'pakistan', 5, 'sarfaraz ahmed', 500, 279, 219, 2),

(102, 'india', 2, 'virat kohli', 700, 400, 200, 0),

(103, 'australia', 4, 'aaron finch', 650, 350, 340, 10),

(104, 'england', 1, 'eoin morgan', 800, 480, 319, 1);

insert into coach

values(201, 'mickey arthur', 'south africa', '1968-05-17', '2016-6-27', 101),

(202, 'ravi shastri', 'india', '1962-05-27', '2017-8-17', 102),

(203, 'Justin Langer', 'australia', '1970-11-21', '2018-01-01', 103),

(204, 'Trevor Bayliss', 'australia', '1962-11-21', '2015-01-01', 104),

(205, 'Shane Warne', 'australia', '1969-08-13', '2014-11-20', 104);

insert into series

values(401, 'pak tour of aus 2018', 'babar azam', 'pakistan', '2018-03-14', '2018-03-30'),

(402, 'ind tour of eng 2k17-18', 'rohit sharma', 'india', '2017-12-20', '2018-01-10'),

(403, 'eng tour of aus 2017', 'eoin morgan', 'england', '2017-05-19', '2017-05-26');

insert into stadium

values(501, 'the oval', 20000, 'london', 'england'),

(502, 'eden gardens', 60000, 'kolkata', 'india'),

(503, 'melbourne', 90000, 'melbourne', 'australia'),

(504, 'Sydney Ground', 48000, 'sydney', 'australia');

insert into umpire

values(701, 'Aleem Dar', 'pakistan', '1968-06-19', 340),

(702, 'Billy Bowden', 'new zealand', '1974-12-06', 251),

(703, 'Marais Erasmus', 'south africa', '1964-02-27', 236),

(704, 'Nigel Llong', 'england', '1969-03-01', 244);

insert into matches

values(601, 'aus', 'pak', 'babar azam', 'pak won','one day', '2018-03-14', 503, 401,701),

(602, 'pak', 'pak', 'aaron finch', 'aus won','one day', '2018-03-23', 504, 401,701),

(603, 'aus', 'aus', 'babar azam', 'pak won','one day', '2018-03-30', 503, 401,701),

(604, 'eng', 'eng', 'rohit sharma', 'ind won','t20', '2017-12-20', 501, 402, 702),

(605, 'ind', 'eng', 'jaspirit bhumra', 'ind won','t20', '2018-01-10', 501, 402, 702),

(606, 'eng', 'aus', 'eoin morgan', 'eng won','t20', '2017-05-19', 504, 403, 704),

(607, 'aus', 'eng', 'stuart broad', 'eng won','t20', '2017-05-26', 503, 403, 704);

insert into scorecard

values(801, 6.23, 311, 6, 50, 11, 1, 101, 601),

(802, 5.5, 264, 11, 48, 14, 2, 103, 601),

(803, 5.8, 290, 8, 50, 12, 1, 101, 602),

(804, 5.9, 292, 4, 48, 8, 2, 103, 602),

(805, 5.88, 294, 7, 50, 15, 1, 103, 603),

(806, 6.43, 296, 4, 46, 9, 2, 101, 603);

insert into player

values(901, 'Imam ul haq', '1995-05-07', 5, '2017-03-21', 101),

(911, 'Fakhar zaman', '1997-04-12', 6, '2018-01-02', 101),

(902, 'babar azam', '1996-12-21', 6, '2016-02-29', 101),

(903, 'azhar ali', '1994-07-08', 5, '2015-10-27', 101),

(904, 'shoib malik', '1993-11-17', 6, '2013-06-22', 101),

(905, 'sarfaraz ahmed', '1991-01-14', 5, '2012-09-01', 101),

(906, 'haris sohail', '1996-12-12', 5, '2016-09-13', 101),

(907, 'shadab khan', '1997-08-10', 5, '2018-02-26', 101),

(908, 'yasir shah', '1990-04-19', 5, '2013-11-11', 101),

(909, 'Mohammad amir', '1992-12-07', 6, '2011-11-21', 101),

(910, 'shaheen afridi', '1993-01-16', 6, '2017-09-13', 101),

(912, 'aaron finch', '1991-04-19', 5, '2014-12-21', 103),

(913, 'marcus haris', '1992-03-27', 6, '2011-03-11', 103),

(914, 'usman khawaja', '1994-12-27', 6, '2012-11-12', 103),

(915, 'shaun marsh', '1991-03-24', 5, '2010-12-27', 103),

(916, 'peter handscomb', '1995-05-22', 5, '2017-04-05', 103),

(917, 'tim pane', '1995-07-17', 6, '2011-10-26', 103),

(918, 'mitchell starc', '1994-10-30', 6, '2013-03-21', 103),

(919, 'nathon lyon', '1996-01-21', 5, '2014-07-27', 103),

(920, 'josh hazlewood', '1991-02-03', 6, '2015-02-19', 103),

(921, 'pat cummins', '1990-05-05', 6, '2013-03-21', 103),

(922, 'peter siddle', '1989-12-17', 5, '2014-02-01', 103);

--

-- match:1, inning:1

--

insert into player\_statistics

values(1001, 901, 50, 1, 5, 0, 0, 0, 601, 801),

(1002, 911, 48, 3, 3, 0, 0, 0, 601, 801),

(1003, 902, 105, 3, 10, 0, 0, 0, 601, 801),

(1004, 903, 0, 0, 0, 0, 0, 0, 601, 801),

(1005, 904, 50, 1, 5, 0, 0, 0, 601, 801),

(1006, 905, 05, 1, 0, 0, 0, 0, 601, 801),

(1007, 906, 34, 3, 2, 8, 1, 1, 601, 801),

(1008, 907, 19, 1, 2, 10, 2, 2, 601, 801),

(1009, 908, 0, 0, 0, 10, 3, 3, 601, 801),

(1010, 909, 0, 0, 0, 10, 3, 2, 601, 801),

(1011, 910, 0, 0, 0, 10, 1, 1, 601, 801),

-- match:1, inning 2

(1012, 912, 44, 1, 5, 0, 0, 0, 601, 802),

(1013, 913, 40, 2, 4, 0, 0, 0, 601, 802),

(1014, 914, 50, 0, 7, 0, 0, 0, 601, 802),

(1015, 915, 60, 2, 5, 0, 0, 0, 601, 802),

(1016, 916, 40, 1, 4, 0, 0, 0, 601, 802),

(1017, 917, 0, 0, 0, 0, 0, 0, 601, 802),

(1018, 918, 12, 1, 0, 10, 0, 0, 601, 802),

(1019, 919, 3, 0, 0, 10, 2, 1, 601, 802),

(1020, 920, 5, 0, 1, 10, 2, 2, 601, 802),

(1021, 921, 0, 0, 0, 10, 1, 1, 601, 802),

(1022, 922, 10, 1, 0, 10, 1, 0, 601, 802),

-- match 2, inning 1

(1023, 901, 13, 0, 1, 0, 0, 0, 602, 803),

(1024, 911, 30, 0, 4, 0, 0, 0, 602, 803),

(1025, 902, 90, 4, 9, 0, 0, 0, 602, 803),

(1026, 903, 10, 0, 0, 0, 0, 0, 602, 803),

(1027, 904, 35, 1, 2, 0, 0, 0, 602, 803),

(1028, 905, 62, 2, 4, 0, 0, 0, 602, 803),

(1029, 906, 34, 1, 3, 8, 0, 0, 602, 803),

(1030, 907, 5, 0, 0, 10, 0, 0, 602, 803),

(1031, 908, 15, 0, 1, 10, 2, 3, 602, 803),

(1032, 909, 0, 0, 0, 10, 2, 0, 602, 803),

(1033, 910, 0, 0, 0, 10, 0, 0, 602, 803),

-- match 2, inning 2

(1034, 912, 108, 7, 5, 0, 0, 0, 602, 804),

(1035, 913, 40, 2, 3, 0, 0, 0, 602, 804),

(1036, 914, 10, 0, 0, 0, 0, 0, 602, 804),

(1037, 915, 60, 2, 5, 0, 0, 0, 602, 804),

(1038, 916, 45, 1, 4, 0, 0, 0, 602, 804),

(1039, 917, 29, 0, 3, 0, 0, 0, 602, 804),

(1040, 918, 0, 1, 0, 10, 0, 1, 602, 804),

(1041, 919, 0, 0, 0, 10, 1, 2, 602, 804),

(1042, 920, 0, 0, 1, 10, 2, 2, 602, 804),

(1043, 921, 0, 0, 0, 10, 1, 1, 602, 804),

(1044, 922, 0, 1, 0, 10, 1, 2, 602, 804),

-- match 3, inning 1

(1045, 912, 75, 3, 4, 0, 0, 0, 603, 805),

(1046, 913, 5, 0, 0, 0, 0, 0, 603, 805),

(1047, 914, 30, 1, 0, 0, 0, 0, 603, 805),

(1048, 915, 60, 0, 8, 0, 0, 0, 603, 805),

(1049, 916, 48, 1, 5, 0, 0, 0, 603, 805),

(1050, 917, 30, 2, 1, 0, 0, 0, 603, 805),

(1051, 918, 6, 1, 2, 6, 0, 0, 603, 805),

(1052, 919, 20, 0, 0, 10, 0, 1, 603, 805),

(1053, 920, 10, 0, 1, 10, 1, 0, 603, 805),

(1054, 921, 5, 0, 0, 10, 1, 0, 603, 805),

(1055, 922, 0, 0, 0, 10, 2, 1, 603, 805),

-- match 3, inning 2

(1056, 901, 60, 0, 5, 0, 0, 0, 603, 806),

(1057, 911, 40, 3, 2, 0, 0, 0, 603, 806),

(1058, 902, 130, 5, 13, 0, 0, 0, 603, 806),

(1059, 903, 10, 0, 0, 0, 0, 0, 603, 806),

(1060, 904, 36, 1, 2, 0, 0, 0, 603, 806),

(1061, 905, 13, 0, 1, 0, 0, 0, 603, 806),

(1062, 906, 0, 0, 0, 10, 0, 0, 603, 806),

(1063, 907, 0, 0, 0, 10, 0, 0, 603, 806),

(1064, 908, 0, 0, 0, 10, 1, 0, 603, 806),

(1065, 909, 0, 0, 0, 10, 2, 0, 603, 806),

(1066, 910, 0, 0, 0, 10, 4, 3, 603, 806);

insert into bowler

values(906, 20, 30, 3-30, 1000, 10),

(907, 15, 120, 5-28, 20000, 11),

(908, 03, 202, 6-40, 34000, 12),

(909, 06, 90, 5-34, 7000, 13),

(910, 30, 48, 3-50, 5000, 14),

(918, 01, 400, 6-24, 10000, 15),

(919, 13, 80, 3-40, 8000,8),

(920, 10, 100, 3-45, 5000, 10),

(921, 08, 40, 5-28, 32000,12),

(922, 20, 90, 3-50, 9000,5);

-- for pakistan players

insert into batsman

values(901, 20, 1000, 5000, 'left hand', 2, 10, 10, 50),

(911, 15, 3000, 9000, 'left hand', 5, 18, 40, 60),

(902, 3, 3500, 20000, 'right hand', 15, 30, 50, 100),

(903, 16, 4000, 25000, 'right hand', 8, 15, 30, 80),

(904, 09, 5000, 15000, 'right hand', 7, 16, 20, 55),

(905, 12, 4500, 25000, 'right hand', 6, 20, 30, 60),

(906, 08, 8000, 30000, 'left hand', 10, 13, 25, 70),

(907, 20, 4000, 8000, 'right hand', 2, 2, 8, 20),

(908, 39, 2000, 5000, 'right hand', 0, 3, 5, 9),

(909, 39, 1000, 4000, 'right hand', 0, 0, 0, 7),

(910, 42, 1500, 3000, 'left hand', 0, 0, 1, 5),

-- for australia players

(912, 06, 8000, 7000, 'left hand', 15, 50, 20, 60),

(913, 14, 9000, 6000, 'left hand', 14, 40, 30, 50),

(914, 19, 7500, 6200, 'right hand', 13, 30, 25, 55),

(915, 21, 6000, 5000, 'right hand', 12, 20, 40, 40),

(916, 07, 5000, 6000, 'right hand', 6, 10, 16, 30),

(917, 18, 4000, 7000, 'right hand', 2, 10, 23, 30),

(918, 35, 2000, 3000, 'left hand', 1, 10, 12, 20),

(919, 38, 100, 2000, 'right hand', 0, 2, 2, 15),

(920, 48, 500, 2500, 'right hand', 0, 3, 0, 10),

(921, 40, 700, 1000, 'left hand', 0, 6, 1, 8),

(922, 45, 300, 500, 'right hand', 0, 3, 0, 5);

**Front End C#**

**Form 1(main)**

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 MetroFramework.Forms;

using MySql.Data.MySqlClient;

namespace FrontEnd

{

public partial class Form1 : MetroForm

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

Form8 form8 = new Form8();

form8.Show();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button2\_Click(object sender, EventArgs e)

{

Form2 form2 = new Form2();

form2.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

Form3 form3 = new Form3();

form3.Show();

}

private void button4\_Click(object sender, EventArgs e)

{

Form4 form4 = new Form4();

form4.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

Form5 form5 = new Form5();

form5.Show();

}

private void button6\_Click(object sender, EventArgs e)

{

Form6 form6 = new Form6();

form6.Show();

}

private void button7\_Click(object sender, EventArgs e)

{

Form7 form7 = new Form7();

form7.Show();

}

private void button8\_Click(object sender, EventArgs e)

{

Form9 form9 = new Form9();

form9.Show();

}

}

}

**Form 2**

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 MetroFramework.Forms;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace FrontEnd

{

public partial class Form2 : MetroForm

{

public Form2()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

runQuery();

}

private void runQuery()

{

string input = textBox1.Text;

if (input == "")

{

MessageBox.Show("Insert an SQL Query!");

return;

}

string MySQLConnectionString = "datasource=127.0.0.1;port=3306;username=root;password=batmanv3;database=cricket";

string query = "select \* from series where series\_id = " + input + ";";

MySqlConnection DatabaseConnection = new MySqlConnection(MySQLConnectionString);

MySqlCommand CommandDatabase = new MySqlCommand(query, DatabaseConnection);

CommandDatabase.CommandTimeout = 60;

try

{

DatabaseConnection.Open();

MySqlDataReader myReader = CommandDatabase.ExecuteReader();

if (myReader.HasRows)

{

MessageBox.Show("The Results generated by the query are given in the console!");

while (myReader.Read())

{

Console.WriteLine("Series ID | Name\t\t\t\t|Man Of Series\t| Series Winner | Start Date \t\t|\t End Date");

Console.WriteLine(myReader.GetString(0) + "\t\t | " + myReader.GetString(1) + " | " + myReader.GetString(2) + "\t| " + myReader.GetString(3) + "\t\t\t| " + myReader.GetString(4) + "\t | " + myReader.GetString(5));

}

}

else

{

MessageBox.Show("Query has been processed!");

}

}

catch (Exception e)

{

MessageBox.Show("Query Failed!" + e.Message);

}

}

}

}

**From 3**

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 MetroFramework.Forms;

using MySql.Data.MySqlClient;

namespace FrontEnd

{

public partial class Form3 : MetroForm

{

public Form3()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

runQuery();

}

private void runQuery()

{

string input = textBox1.Text;

if (input == "")

{

MessageBox.Show("Insert an SQL Query!");

return;

}

string MySQLConnectionString = "datasource=127.0.0.1;port=3306;username=root;password=batmanv3;database=cricket";

string query = "select \* from team where team\_id = " + input + ";";

MySqlConnection DatabaseConnection = new MySqlConnection(MySQLConnectionString);

MySqlCommand CommandDatabase = new MySqlCommand(query, DatabaseConnection);

CommandDatabase.CommandTimeout = 60;

try

{

DatabaseConnection.Open();

MySqlDataReader myReader = CommandDatabase.ExecuteReader();

if (myReader.HasRows)

{

MessageBox.Show("The Results generated by the query are given in the console!");

while (myReader.Read())

{

Console.WriteLine("Team ID\t" + " | " + "Team Country\t" + " | " + "Rank\t" + " | " + "Captain\t\t" + " | " + "Matches Played\t" + " | " + "Matches Won\t" + " | " + "Matches Lost\t "+ " | " + "Matches Tied");

Console.WriteLine(myReader.GetString(0) + "\t\t | " + myReader.GetString(1) + "\t\t | " + myReader.GetString(2) + "\t | " + myReader.GetString(3) + "| " + myReader.GetString(4) + "\t\t\t\t | " + myReader.GetString(5) + " \t\t\t| " + myReader.GetString(6) + "\t\t\t | " + myReader.GetString(7));

}

}

else

{

MessageBox.Show("Query has been processed!");

}

}

catch (Exception e)

{

MessageBox.Show("Query Failed!" + e.Message);

}

}

}

}

**Form 4**

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 MetroFramework.Forms;

using MySql.Data.MySqlClient;

namespace FrontEnd

{

public partial class Form4 : MetroForm

{

public Form4()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

runQuery();

}

private void runQuery()

{

string input = textBox1.Text;

if (input == "")

{

MessageBox.Show("Insert an SQL Query!");

return;

}

string MySQLConnectionString = "datasource=127.0.0.1;port=3306;username=root;password=batmanv3;database=cricket";

string query = "select \* from stadium where stadium\_id = " + input + ";";

MySqlConnection DatabaseConnection = new MySqlConnection(MySQLConnectionString);

MySqlCommand CommandDatabase = new MySqlCommand(query, DatabaseConnection);

CommandDatabase.CommandTimeout = 60;

try

{

DatabaseConnection.Open();

MySqlDataReader myReader = CommandDatabase.ExecuteReader();

if (myReader.HasRows)

{

MessageBox.Show("The Results generated by the query are given in the console!");

while (myReader.Read())

{

Console.WriteLine("Stadium ID\t" + " | " + "Stadium Name\t" + " | " + "Capacity\t" + " | " + " City " + " | " + "Country\t");

Console.WriteLine(myReader.GetString(0) + "\t\t\t | " + myReader.GetString(1) + "\t\t | " + myReader.GetString(2) + "\t | " + myReader.GetString(3) + "\t\t| " + myReader.GetString(4));

}

}

else

{

MessageBox.Show("Query has been processed!");

}

}

catch (Exception e)

{

MessageBox.Show("Query Failed!" + e.Message);

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

}

}

**Form 5**

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 MetroFramework.Forms;

using MySql.Data.MySqlClient;

namespace FrontEnd

{

public partial class Form6 : MetroForm

{

public Form6()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

runQuery();

}

private void runQuery()

{

string input = textBox1.Text;

if (input == "")

{

MessageBox.Show("Insert an SQL Query!");

return;

}

string MySQLConnectionString = "datasource=127.0.0.1;port=3306;username=root;password=batmanv3;database=cricket";

string query = "select \* from player where team\_id = " + input + ";";

MySqlConnection DatabaseConnection = new MySqlConnection(MySQLConnectionString);

MySqlCommand CommandDatabase = new MySqlCommand(query, DatabaseConnection);

CommandDatabase.CommandTimeout = 60;

try

{

DatabaseConnection.Open();

MySqlDataReader myReader = CommandDatabase.ExecuteReader();

if (myReader.HasRows)

{

MessageBox.Show("The Results generated by the query are given in the console!");

while (myReader.Read())

{

Console.WriteLine("Player ID\t|\t\t Name" + "\t\t |\t\t " + "Date of Birth\t" + "\t | " + "Height" + " |\t\t " + "Debut Date" + "\t\t | " + "Team ID");

Console.WriteLine(myReader.GetString(0) + "\t\t\t| " + myReader.GetString(1) + "\t\t | " + myReader.GetString(2) + "\t | " + myReader.GetString(3) + "\t |\t" + myReader.GetString(4) + " | " + myReader.GetString(5));

}

}

else

{

MessageBox.Show("Query has been processed!");

}

}

catch (Exception e)

{

MessageBox.Show("Query Failed!" + e.Message);

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

}

}

**Form 6**

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 MetroFramework.Forms;

using MySql.Data.MySqlClient;

namespace FrontEnd

{

public partial class Form6 : MetroForm

{

public Form6()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

runQuery();

}

private void runQuery()

{

string input = textBox1.Text;

if (input == "")

{

MessageBox.Show("Insert an SQL Query!");

return;

}

string MySQLConnectionString = "datasource=127.0.0.1;port=3306;username=root;password=batmanv3;database=cricket";

string query = "select \* from player where team\_id = " + input + ";";

MySqlConnection DatabaseConnection = new MySqlConnection(MySQLConnectionString);

MySqlCommand CommandDatabase = new MySqlCommand(query, DatabaseConnection);

CommandDatabase.CommandTimeout = 60;

try

{

DatabaseConnection.Open();

MySqlDataReader myReader = CommandDatabase.ExecuteReader();

if (myReader.HasRows)

{

MessageBox.Show("The Results generated by the query are given in the console!");

while (myReader.Read())

{

Console.WriteLine("Player ID\t|\t\t Name" + "\t\t |\t\t " + "Date of Birth\t" + "\t | " + "Height" + " |\t\t " + "Debut Date" + "\t\t | " + "Team ID");

Console.WriteLine(myReader.GetString(0) + "\t\t\t| " + myReader.GetString(1) + "\t\t | " + myReader.GetString(2) + "\t | " + myReader.GetString(3) + "\t |\t" + myReader.GetString(4) + " | " + myReader.GetString(5));

}

}

else

{

MessageBox.Show("Query has been processed!");

}

}

catch (Exception e)

{

MessageBox.Show("Query Failed!" + e.Message);

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

}

}

**Form 7**

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 MetroFramework.Forms;

using MySql.Data.MySqlClient;

namespace FrontEnd

{

public partial class Form7 : MetroForm

{

public Form7()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void runQuery()

{

string input = textBox1.Text;

if (input == "")

{

MessageBox.Show("Insert an SQL Query!");

return;

}

string MySQLConnectionString = "datasource=127.0.0.1;port=3306;username=root;password=batmanv3;database=cricket";

string query = "select \* from umpire where umpire\_id = " + input + ";";

MySqlConnection DatabaseConnection = new MySqlConnection(MySQLConnectionString);

MySqlCommand CommandDatabase = new MySqlCommand(query, DatabaseConnection);

CommandDatabase.CommandTimeout = 60;

try

{

DatabaseConnection.Open();

MySqlDataReader myReader = CommandDatabase.ExecuteReader();

if (myReader.HasRows)

{

MessageBox.Show("The Results generated by the query are given in the console!");

while (myReader.Read())

{

Console.WriteLine("Umpire ID\t|\t Umpire Name\t" + " | \t" + "Country\t" + " |\t " + "Date of Birth" + "\t\t\t\t | " + "Matches Umpired");

Console.WriteLine(myReader.GetString(0) + "\t\t\t|\t " + myReader.GetString(1) + "\t\t | " + myReader.GetString(2) + "\t | " + myReader.GetString(3) + "\t\t |\t" + myReader.GetString(4));

}

}

else

{

MessageBox.Show("Query has been processed!");

}

}

catch (Exception e)

{

MessageBox.Show("Query Failed!" + e.Message);

}

}

private void button2\_Click(object sender, EventArgs e)

{

runQuery();

}

}

}

**Form 8**

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 MetroFramework.Forms;

using MySql.Data.MySqlClient;

namespace FrontEnd

{

public partial class Form8 : MetroForm

{

public Form8()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

runQuery();

}

private void runQuery()

{

string query;

string column = textBox1.Text;

int noofcolumn;

Int32.TryParse(textBox2.Text, out noofcolumn);

Console.WriteLine(noofcolumn);

string table1 = textBox3.Text;

string table2 = textBox4.Text;

if (column == "")

{

MessageBox.Show("Insert column name!");

return;

}

else if(column == "all")

{

column = "\*";

}

if (table2 != "")

{

table2 = "natural join " + table2;

query = "select " + column + " from " + table1 + table2 + ";";

}

else

query = "select " + column + " from " + table1 + ";";

if(table2 == "" && column == "\*")

{

if (table1 == "player")

noofcolumn = 6;

else if(table1 == "batsman")

noofcolumn = 9;

else if (table1 == "bowler")

noofcolumn = 6;

else if (table1 == "coach")

noofcolumn = 6;

else if (table1 == "matches")

noofcolumn = 10;

else if (table1 == "player\_statistics")

noofcolumn = 10;

else if (table1 == "scorecard")

noofcolumn = 9;

else if (table1 == "series")

noofcolumn = 6;

else if (table1 == "squad")

noofcolumn = 2;

else if (table1 == "stadium")

noofcolumn = 5;

else if (table1 == "team")

noofcolumn = 8;

else if (table1 == "umpire")

noofcolumn = 5;

}

string MySQLConnectionString = "datasource=127.0.0.1;port=3306;username=root;password=batmanv3;database=cricket";

MySqlConnection DatabaseConnection = new MySqlConnection(MySQLConnectionString);

MySqlCommand CommandDatabase = new MySqlCommand(query, DatabaseConnection);

CommandDatabase.CommandTimeout = 60;

try

{

DatabaseConnection.Open();

MySqlDataReader myReader = CommandDatabase.ExecuteReader();

if (myReader.HasRows)

{

MessageBox.Show("The Results generated by the query are given in the console!");

while (myReader.Read())

{

switch (noofcolumn)

{

case 1:

Console.WriteLine(myReader.GetString(0));

break;

case 2:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1));

break;

case 3:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2));

break;

case 4:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3));

break;

case 5:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4));

break;

case 6:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5));

break;

case 7:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6));

break;

case 8:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7));

break;

case 9:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8));

break;

case 10:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8) + " | " + myReader.GetString(9));

break;

case 11:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8) + " | " + myReader.GetString(9) + " | " + myReader.GetString(10));

break;

case 12:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8) + " | " + myReader.GetString(9) + " | " + myReader.GetString(10) + " | " + myReader.GetString(11));

break;

case 13:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8) + " | " + myReader.GetString(9) + " | " + myReader.GetString(10) + " | " + myReader.GetString(11) + " | " + myReader.GetString(12));

break;

case 14:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8) + " | " + myReader.GetString(9) + " | " + myReader.GetString(10) + " | " + myReader.GetString(11) + " | " + myReader.GetString(12) + " | " + myReader.GetString(13));

break;

case 15:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8) + " | " + myReader.GetString(9) + " | " + myReader.GetString(10) + " | " + myReader.GetString(11) + " | " + myReader.GetString(12) + " | " + myReader.GetString(13) + " | " + myReader.GetString(14));

break;

case 16:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8) + " | " + myReader.GetString(9) + " | " + myReader.GetString(10) + " | " + myReader.GetString(11) + " | " + myReader.GetString(12) + " | " + myReader.GetString(13) + " | " + myReader.GetString(14) + " | " + myReader.GetString(15));

break;

case 17:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8) + " | " + myReader.GetString(9) + " | " + myReader.GetString(10) + " | " + myReader.GetString(11) + " | " + myReader.GetString(12) + " | " + myReader.GetString(13) + " | " + myReader.GetString(14) + " | " + myReader.GetString(15) + " | " + myReader.GetString(16));

break;

case 18:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8) + " | " + myReader.GetString(9) + " | " + myReader.GetString(10) + " | " + myReader.GetString(11) + " | " + myReader.GetString(12) + " | " + myReader.GetString(13) + " | " + myReader.GetString(14) + " | " + myReader.GetString(15) + " | " + myReader.GetString(16) + " | " + myReader.GetString(17));

break;

case 19:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8) + " | " + myReader.GetString(9) + " | " + myReader.GetString(10) + " | " + myReader.GetString(11) + " | " + myReader.GetString(12) + " | " + myReader.GetString(13) + " | " + myReader.GetString(14) + " | " + myReader.GetString(15) + " | " + myReader.GetString(16) + " | " + myReader.GetString(17) + " | " + myReader.GetString(18));

break;

case 20:

Console.WriteLine(myReader.GetString(0) + " | " + myReader.GetString(1) + " | " + myReader.GetString(2) + " | " + myReader.GetString(3) + " | " + myReader.GetString(4) + " | " + myReader.GetString(5) + " | " + myReader.GetString(6) + " | " + myReader.GetString(7) + " | " + myReader.GetString(8) + " | " + myReader.GetString(9) + " | " + myReader.GetString(10) + " | " + myReader.GetString(11) + " | " + myReader.GetString(12) + " | " + myReader.GetString(13) + " | " + myReader.GetString(14) + " | " + myReader.GetString(15) + " | " + myReader.GetString(16) + " | " + myReader.GetString(17) + " | " + myReader.GetString(18) + " | " + myReader.GetString(19));

break;

default:

break;

}

}

}

else

{

MessageBox.Show("Query has been processed!");

}

}

catch (Exception e)

{

MessageBox.Show("Query Failed!" + e.Message);

}

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

}

}

**Form 9**

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 MetroFramework.Forms;

using MySql.Data.MySqlClient;

namespace FrontEnd

{

public partial class Form9 : MetroForm

{

public Form9()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

runQuery();

}

private void runQuery()

{

string query = textBox1.Text;

if (query == "")

{

MessageBox.Show("Insert an SQL Query!");

return;

}

string MySQLConnectionString = "datasource=127.0.0.1;port=3306;username=root;password=batmanv3;database=cricket";

MySqlConnection DatabaseConnection = new MySqlConnection(MySQLConnectionString);

MySqlCommand CommandDatabase = new MySqlCommand(query, DatabaseConnection);

CommandDatabase.CommandTimeout = 60;

try

{

DatabaseConnection.Open();

MySqlDataReader myReader = CommandDatabase.ExecuteReader();

if (myReader.HasRows)

{

MessageBox.Show("The Results generated by the query are given in the console!");

}

else

{

MessageBox.Show("Query has been processed!");

}

}

catch (Exception e)

{

MessageBox.Show("Query Failed!" + e.Message);

}

}

}

}