

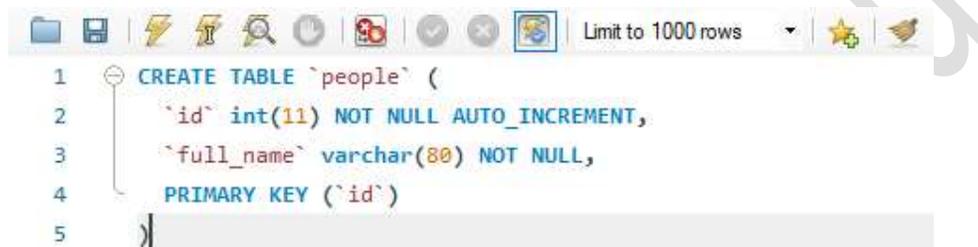
Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

Создание базы данных «Заказ тортов».



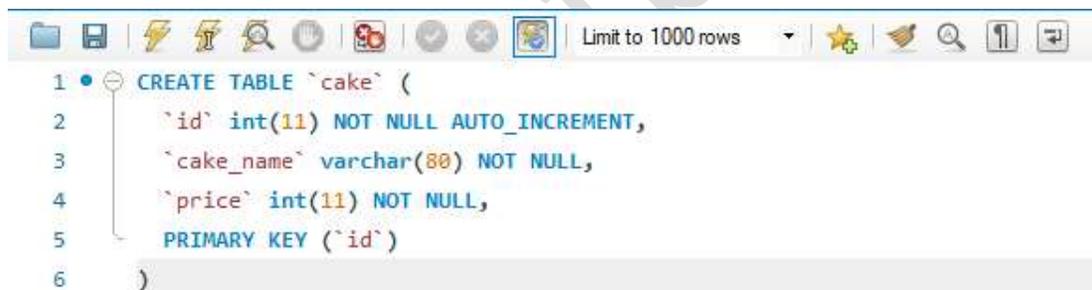
```
1 • CREATE SCHEMA `cakes`;
```

Рисунок 1 – Создание базы данных «Cakes»



```
1 CREATE TABLE `people` (  
2   `id` int(11) NOT NULL AUTO_INCREMENT,  
3   `full_name` varchar(80) NOT NULL,  
4   PRIMARY KEY (`id`)  
5 )
```

Рисунок 2 – Создание таблицы «People»



```
1 • CREATE TABLE `cake` (  
2   `id` int(11) NOT NULL AUTO_INCREMENT,  
3   `cake_name` varchar(80) NOT NULL,  
4   `price` int(11) NOT NULL,  
5   PRIMARY KEY (`id`)  
6 )
```

Рисунок 3 – Создание таблицы «Cake»



```
1 • CREATE TABLE `orders` (  
2   `id_order` int(11) NOT NULL AUTO_INCREMENT,  
3   `id_people` int(11) NOT NULL,  
4   `id_cake` int(11) NOT NULL,  
5   PRIMARY KEY (`id_order`),  
6   KEY `id_cake_idx` (`id_cake`),  
7   KEY `id_people_idx` (`id_people`),  
8   CONSTRAINT `id_cake` FOREIGN KEY (`id_cake`) REFERENCES `cake` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,  
9   CONSTRAINT `id_people` FOREIGN KEY (`id_people`) REFERENCES `people` (`id`) ON DELETE CASCADE ON UPDATE CASCADE  
10 )
```

Рисунок 4 – Создание таблицы «Orders» и внешних ключей для связи с таблицами «People» и «Cake»

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz



```
people
Limit to 1000 rows
1 INSERT INTO `cakes`.`people` (`full_name`) VALUES ('Порошенк Виктор Михайлович');
2 INSERT INTO `cakes`.`people` (`full_name`) VALUES ('Демидов Вадим Николаевич');
3 INSERT INTO `cakes`.`people` (`full_name`) VALUES ('Пономарева Екатерина Евгеньевна');
4 INSERT INTO `cakes`.`people` (`full_name`) VALUES ('Мамошина Ольга Владимировна');
5 INSERT INTO `cakes`.`people` (`full_name`) VALUES ('Маслова Анастасия Сергеевна');
```

Рисунок 5 – Заполнение данными таблицы «People»



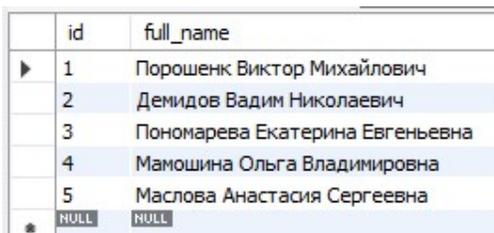
```
cake
Limit to 1000 rows
1 INSERT INTO `cakes`.`cake` (`cake_name`, `price`) VALUES ('Черепашка', '350');
2 INSERT INTO `cakes`.`cake` (`cake_name`, `price`) VALUES ('Наполеон', '500');
3 INSERT INTO `cakes`.`cake` (`cake_name`, `price`) VALUES ('Филибейке', '250');
4 INSERT INTO `cakes`.`cake` (`cake_name`, `price`) VALUES ('Медовый', '200');
5 INSERT INTO `cakes`.`cake` (`cake_name`, `price`) VALUES ('Сметанин', '400');
```

Рисунок 6 – Заполнение данными таблицы «Cake»



```
orders
Limit to 1000 rows
1 INSERT INTO `cakes`.`orders` (`id_people`, `id_cake`) VALUES ('1', '1');
2 INSERT INTO `cakes`.`orders` (`id_people`, `id_cake`) VALUES ('3', '2');
3 INSERT INTO `cakes`.`orders` (`id_people`, `id_cake`) VALUES ('4', '5');
4 INSERT INTO `cakes`.`orders` (`id_people`, `id_cake`) VALUES ('2', '3');
5 INSERT INTO `cakes`.`orders` (`id_people`, `id_cake`) VALUES ('5', '4');
```

Рисунок 7 – Заполнение данными таблицы «Orders»



	id	full_name
▶	1	Порошенк Виктор Михайлович
	2	Демидов Вадим Николаевич
	3	Пономарева Екатерина Евгеньевна
	4	Мамошина Ольга Владимировна
	5	Маслова Анастасия Сергеевна
*	NULL	NULL

Рисунок 8 – Заполненная таблица «People»

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

	id	cake_name	price
▶	1	Черепаша	350
	2	Наполеон	500
	3	Филибейке	250
	4	Медовый	200
	5	Сметанин	400
*	NULL	NULL	NULL

Рисунок 9 – Заполненная таблица «Cake»

	id_order	id_people	id_cake
▶	1	1	1
	2	3	2
	3	4	5
	4	2	3
	5	5	4
*	NULL	NULL	NULL

Рисунок 10 – Заполненная таблица «Orders»

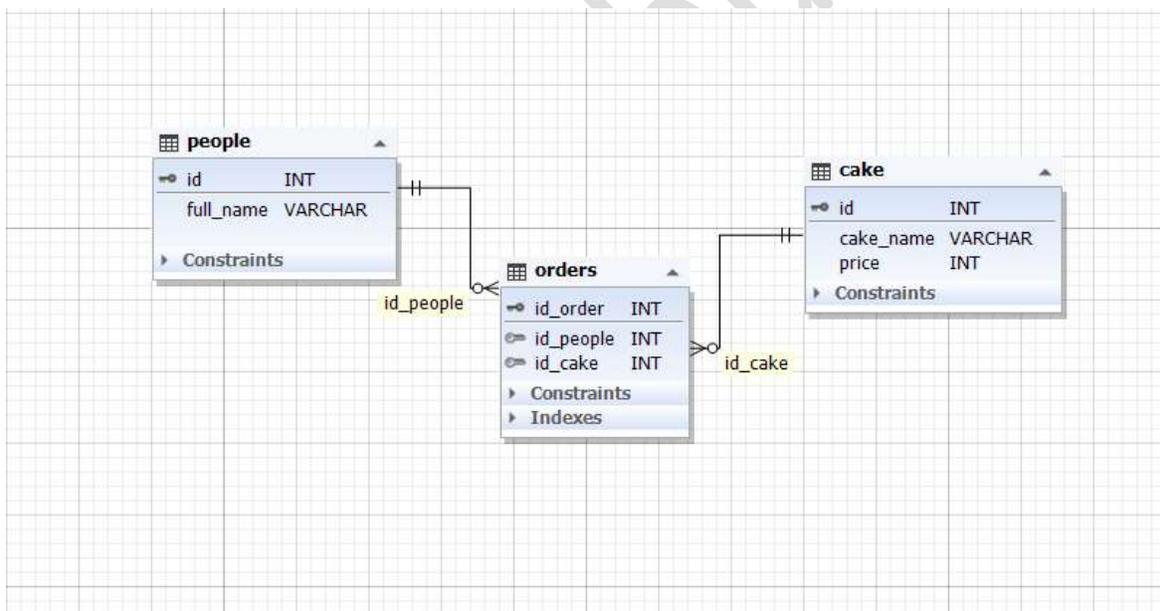


Рисунок 11 – Схема данных

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

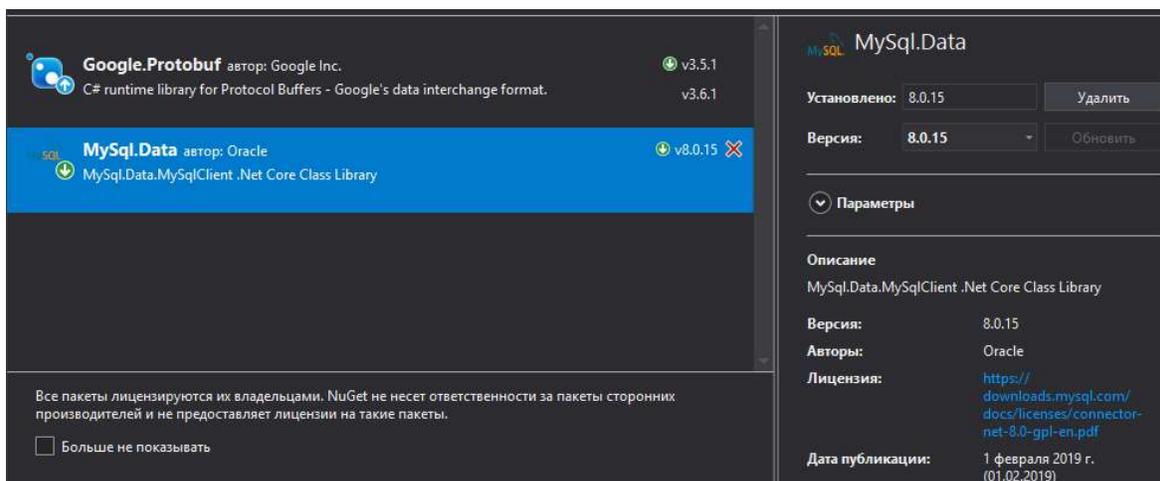


Рисунок 12 – Добавление ссылки на службу MySQL.Data для возможности подключения Visual Studio к базам MySQL

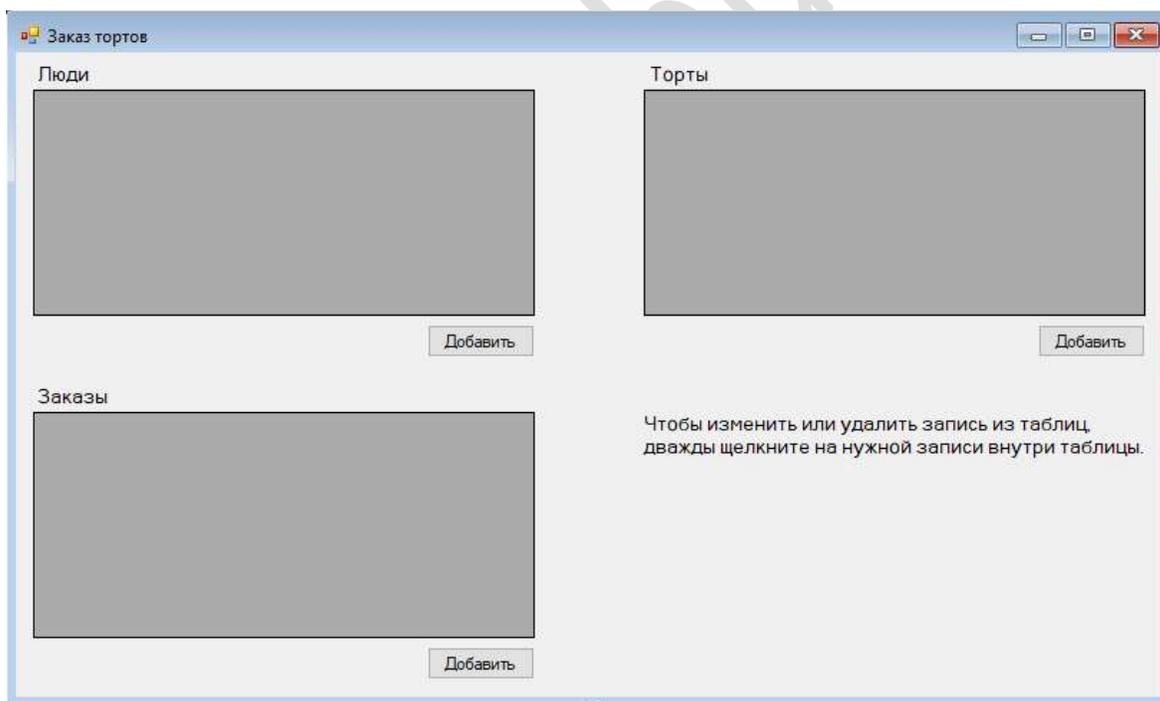


Рисунок 13 – Создание и проектирование интерфейса главной формы программы

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

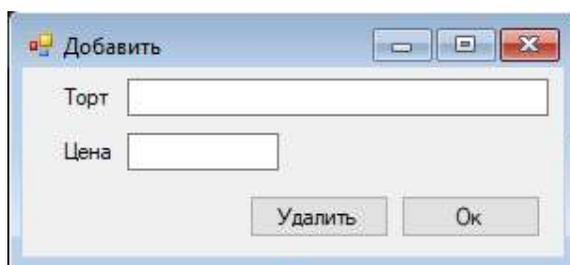


Рисунок 14 – Создание и проектирование интерфейса формы для добавления тортов

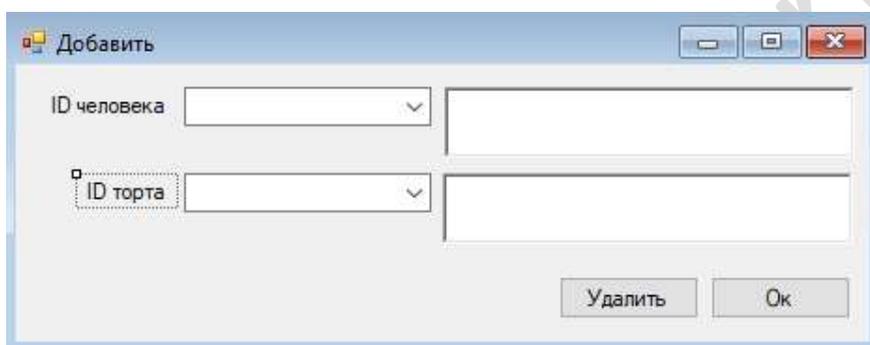


Рисунок 15 – Создание и проектирование интерфейса формы для добавления заказов

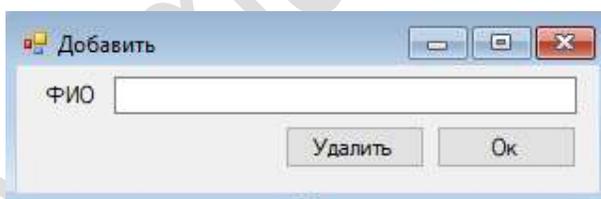


Рисунок 16 – Создание и проектирование интерфейса формы для добавления людей (покупателей, заказчиков)

```
10 using MySql.Data.MySqlClient;
```

Рисунок 17 – Подключение директивы `MySql.Data.MySqlClient` для возможности создания подключения к базам `MySQL`

```
16 public static string conString = "Server=127.0.0.1;Database=cakes;port=3306;User Id=root;password=root";  
17 MySqlConnection con = new MySqlConnection(conString);  
18 MySqlCommand cmd;
```

Рисунок 18 – Давление строки для подключения к базе данных, создание подключения `MySqlConnection` и объекта `MySqlCommand` для возможности выполнения запросов (`INSERT`, `UPDATE`, `DELETE` и прочее)

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

```
64 public void getPeople()
65 {
66     DataTable dt = new DataTable();
67     dataGridView1.Rows.Clear();
68     string sql = "SELECT * FROM cakes.people";
69     cmd = new MySqlCommand(sql, con);
70
71     try
72     {
73         con.Open();
74
75         adapter = new MySqlDataAdapter(cmd);
76
77         adapter.Fill(dt);
78
79         foreach (DataRow row in dt.Rows)
80         {
81             dataGridView1.Rows.Add(row[0].ToString(), row[1].ToString());
82         }
83
84         con.Close();
85
86         dt.Rows.Clear();
```

Рисунок 19 – Функция getPeople() выполняет запрос SELECT к таблице «people» из созданной базы «cakes» и заполняет данными элемент формы DataGridView (табличное представление)

```
96 public void getCakes()
97 {
98     DataTable dt = new DataTable();
99     dataGridView2.Rows.Clear();
100     string sql = "SELECT * FROM cakes.cake";
101     cmd = new MySqlCommand(sql, con);
102
103     try
104     {
105         con.Open();
106
107         adapter = new MySqlDataAdapter(cmd);
108
109         adapter.Fill(dt);
110
111         foreach (DataRow row in dt.Rows)
112         {
113             dataGridView2.Rows.Add(row[0].ToString(), row[1].ToString(), row[2].ToString());
114         }
115
116         con.Close();
117
118         dt.Rows.Clear();
```

Рисунок 20 – getCakes() работает аналогично getPeople(), но по отношению к таблице «cake» базы данных «cakes»

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

```
128 public void getOrders()
129 {
130     DataTable dt = new DataTable();
131     dataGridView3.Rows.Clear();
132     string sql = "SELECT * FROM cakes.orders";
133     cmd = new MySqlCommand(sql, con);
134
135     try
136     {
137         con.Open();
138
139         adapter = new MySqlDataAdapter(cmd);
140
141         adapter.Fill(dt);
142
143         foreach (DataRow row in dt.Rows)
144         {
145             dataGridView3.Rows.Add(row[0].ToString(), row[1].ToString(), row[2].ToString());
146         }
147
148         con.Close();
149
150         dt.Rows.Clear();

```

Рисунок 20 – getOrders() работает аналогично getPeople(), но по отношению к таблице «orders» базы данных «cakes»

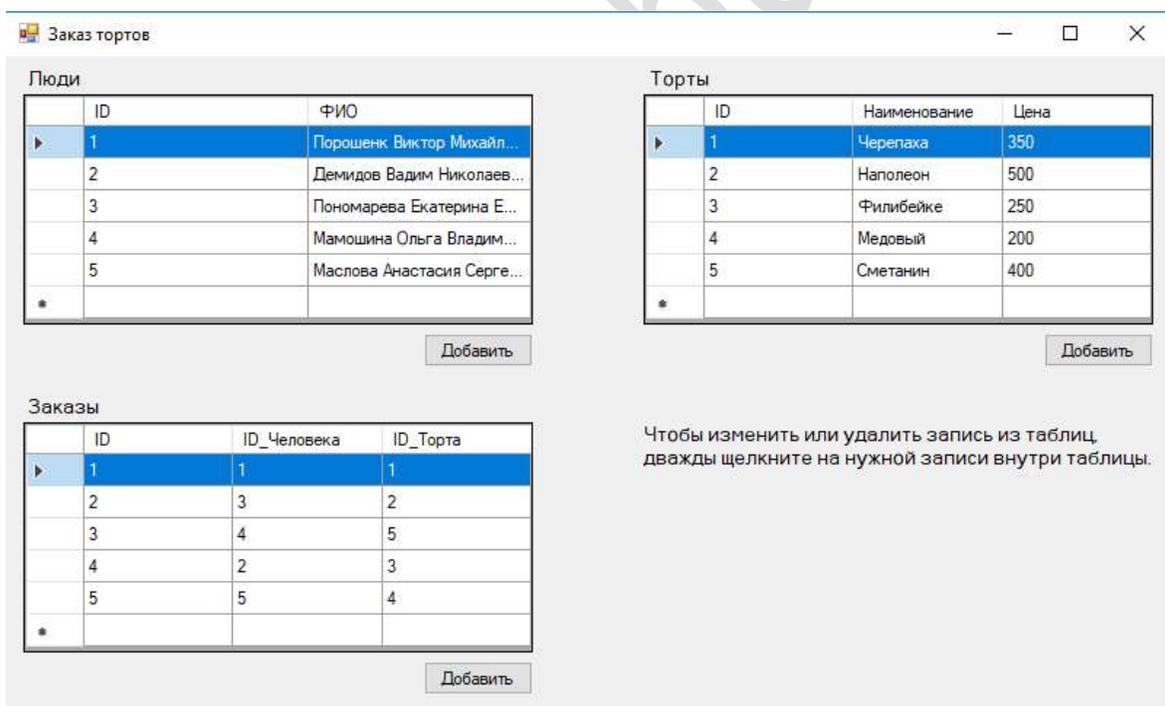


Рисунок 21 – Результат выполнения запросов SELECT к базе «cakes»

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

```
46 private void button1_Click(object sender, EventArgs e)
47 {
48     if (f1.editCake == 0)
49     {
50         con.Open();
51         string query = "INSERT INTO `cakes`.`cake` (`cake_name`, `price`) VALUES ('" + textBox1.Text + "', '" + textBox2.Text + "')";
52         //строка запроса на добавление
53
54         try
55         {
56             MySqlCommand cmd = new MySqlCommand(query, con);
57             cmd.ExecuteNonQuery();
58             f1.getCakes();
59             MessageBox.Show("Запись добавлена");
60         }
61         catch (Exception ex)
62         {
63             MessageBox.Show(ex.Message);
64         }
65         con.Close();
66         this.Close();
67     }
}
```

Рисунок 22 – Реализация запроса INSERT к базе для добавления записи в таблицу «cake» после нажатия кнопки «Добавить»

```
69 {
70     con.Open();
71     string query = "UPDATE `cakes`.`cake` SET `cake_name` = '" + textBox1.Text + "', `price` = '" + textBox2.Text + "' WHERE (`id` = '" + id + "')";
72     //строка запроса на изменение
73
74     try
75     {
76         MySqlCommand cmd = new MySqlCommand(query, con);
77         cmd.ExecuteNonQuery();
78         f1.getCakes();
79         MessageBox.Show("Запись изменена");
80     }
81     catch (Exception ex)
82     {
83         MessageBox.Show(ex.Message);
84     }
85     con.Close();
86     this.Close();
87 }
88 }
```

Рисунок 23 – Реализация запроса UPDATE к базе для изменения записи в таблице «cake» после нажатия кнопки «Изменить»

```
91 private void button2_Click(object sender, EventArgs e)
92 {
93     con.Open();
94     string query = "DELETE FROM `cakes`.`cake` WHERE (`id` = '" + id + "')";
95     //строка запроса на изменение
96
97     try
98     {
99         MySqlCommand cmd = new MySqlCommand(query, con);
100         cmd.ExecuteNonQuery();
101         f1.getCakes();
102         MessageBox.Show("Запись удалена");
103     }
104     catch (Exception ex)
105     {
106         MessageBox.Show(ex.Message);
107     }
108     con.Close();
109     this.Close();
110 }
111 }
```

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

Рисунок 24 – Реализация запроса DELETE к базе для удаления записи в таблице «cake» после нажатия кнопки «Удалить»

```
26 private void button1_Click(object sender, EventArgs e)
27 {
28     if (f1.editPeople == 0)
29     {
30         con.Open();
31         string query = "INSERT INTO `cakes`.`people` (`full_name`) VALUES ('" + textBox1.Text + "')";
32         //строка запроса на добавление
33
34         try
35         {
36             MySqlCommand cmd = new MySqlCommand(query, con);
37             cmd.ExecuteNonQuery();
38             f1.getPeople();
39             MessageBox.Show("Запись добавлена");
40         }
41         catch (Exception ex)
42         {
43             MessageBox.Show(ex.Message);
44         }
45         con.Close();
46         this.Close();
47     }
}
```

Рисунок 25 – Реализация запроса INSERT к базе для добавления записи в таблицу «people» после нажатия кнопки «Добавить»

```
49 {
50     con.Open();
51     string query = "UPDATE `cakes`.`people` SET `full_name` = '" + textBox1.Text + "' WHERE (`id` = '" + id + "')";
52     //строка запроса на изменение
53
54     try
55     {
56         MySqlCommand cmd = new MySqlCommand(query, con);
57         cmd.ExecuteNonQuery();
58         f1.getPeople();
59         MessageBox.Show("Запись изменена");
60     }
61     catch (Exception ex)
62     {
63         MessageBox.Show(ex.Message);
64     }
65     con.Close();
66     this.Close();
67 }
68 }
```

Рисунок 26 – Реализация запроса UPDATE к базе для изменения записи в таблице «people» после нажатия кнопки «Изменить»

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

```
90     private void button2_Click(object sender, EventArgs e)
91     {
92         con.Open();
93         string query = "DELETE FROM `cakes`.`people` WHERE (`id` = '" + id + "')";
94         //стока запроса на изменение
95
96         try
97         {
98             MySqlCommand cmd = new MySqlCommand(query, con);
99             cmd.ExecuteNonQuery();
100            f1.getPeople();
101            MessageBox.Show("Запись удалена");
102        }
103        catch (Exception ex)
104        {
105
106            MessageBox.Show(ex.Message);
107        }
108        con.Close();
109        this.Close();
110    }
```

Рисунок 27 – Реализация запроса DELETE к базе для удаления записи в таблице «people» после нажатия кнопки «Удалить»

```
117     private void button1_Click(object sender, EventArgs e)
118     {
119         if (f1.editOrder == 0)
120         {
121             con.Open();
122             string query = "INSERT INTO `cakes`.`orders` (`id_people`, `id_cake`) VALUES ('" + comboBox1.Text + "', '" + comboBox2.Text + "')";
123             //стока запроса на добавление
124             try
125             {
126                 MySqlCommand cmd = new MySqlCommand(query, con);
127                 cmd.ExecuteNonQuery();
128                 f1.getOrders();
129                 MessageBox.Show("Запись добавлена");
130             }
131             catch (Exception ex)
132             {
133                 MessageBox.Show(ex.Message);
134             }
135             con.Close();
136             this.Close();
137         }
138     }
```

Рисунок 28 – Реализация запроса INSERT к базе для добавления записи в таблицу «orders» после нажатия кнопки «Добавить»

```
139     {
140         con.Open();
141         string query = "UPDATE `cakes`.`orders` SET `id_people` = '" + comboBox1.Text + "', `id_cake` = '" + comboBox2.Text + "' WHERE (`id_order` = '" + id + "')";
142         //стока запроса на изменение
143
144         try
145         {
146             MySqlCommand cmd = new MySqlCommand(query, con);
147             cmd.ExecuteNonQuery();
148             f1.getOrders();
149             MessageBox.Show("Запись изменена");
150         }
151         catch (Exception ex)
152         {
153
154             MessageBox.Show(ex.Message);
155         }
156         con.Close();
157         this.Close();
158     }
```

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

Рисунок 29 – Реализация запроса UPDATE к базе для изменения записи в таблице «orders» после нажатия кнопки «Изменить»

```
161 private void button2_Click(object sender, EventArgs e)
162 {
163     con.Open();
164     string query = "DELETE FROM `cakes`.`orders` WHERE (`id_order` = '" + id + "')";
165     //стока запроса на изменение
166
167     try
168     {
169         MySqlCommand cmd = new MySqlCommand(query, con);
170         cmd.ExecuteNonQuery();
171         f1.getOrders();
172         MessageBox.Show("Запись удалена");
173     }
174     catch (Exception ex)
175     {
176
177         MessageBox.Show(ex.Message);
178     }
179     con.Close();
180     this.Close();
181 }
```

Рисунок 30 – Реализация запроса DELETE к базе для удаления записи в таблице «orders» после нажатия кнопки «Удалить»

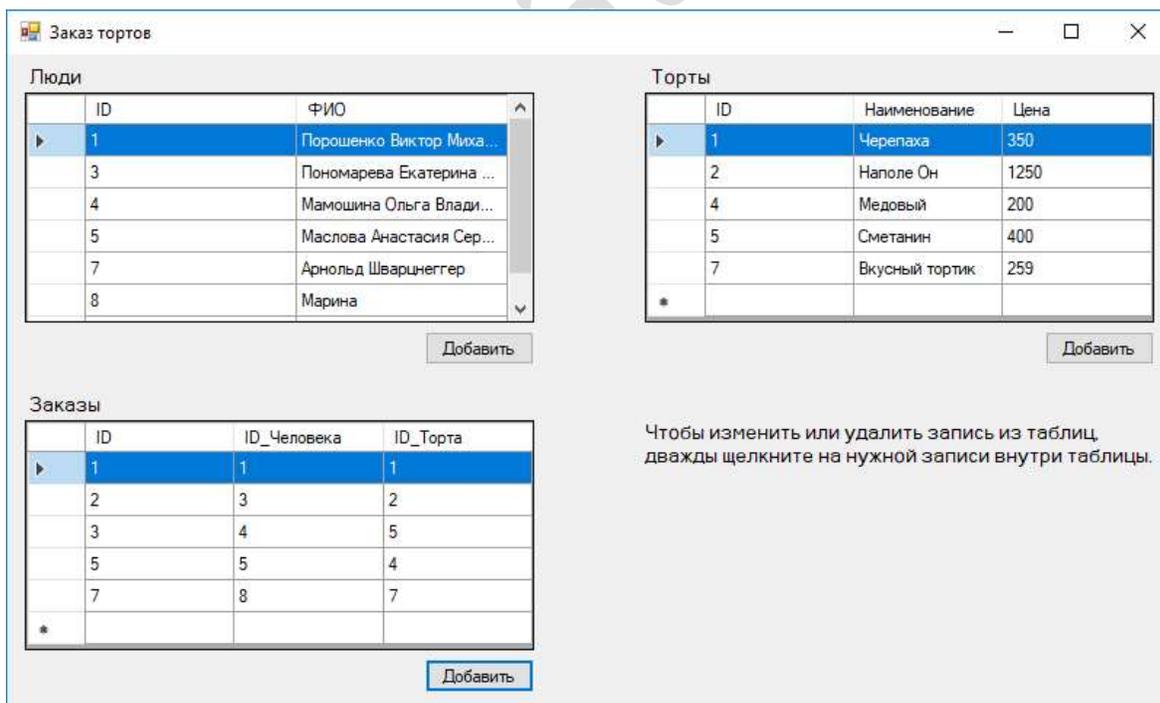


Рисунок 31 – База данных после всех вышеописанных манипуляций

Контрольная работа выполнена в www.MatBuro.ru

©МатБюро – Консультации по математике, программированию, экономике, праву, естественным наукам

Поможем вам с написанием программ: www.matburo.ru/sub_subject.php?p=pz

www.matburo.ru