

# ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №2

Выполнение запросов SQL при помощи текстовых команд.

## КРАТКОЕ ОПИСАНИЕ КОМАНД SQL

Действия над базой данных SQL производятся при помощи специального языка запросов. Каждый запрос представляет собой простое текстовое сообщение, содержащее команду с набором параметров. Существует множество различных команд для работы со всеми существующими элементами базы данных. В этой лабораторной мы рассмотрим несколько операторов для работы с таблицами.

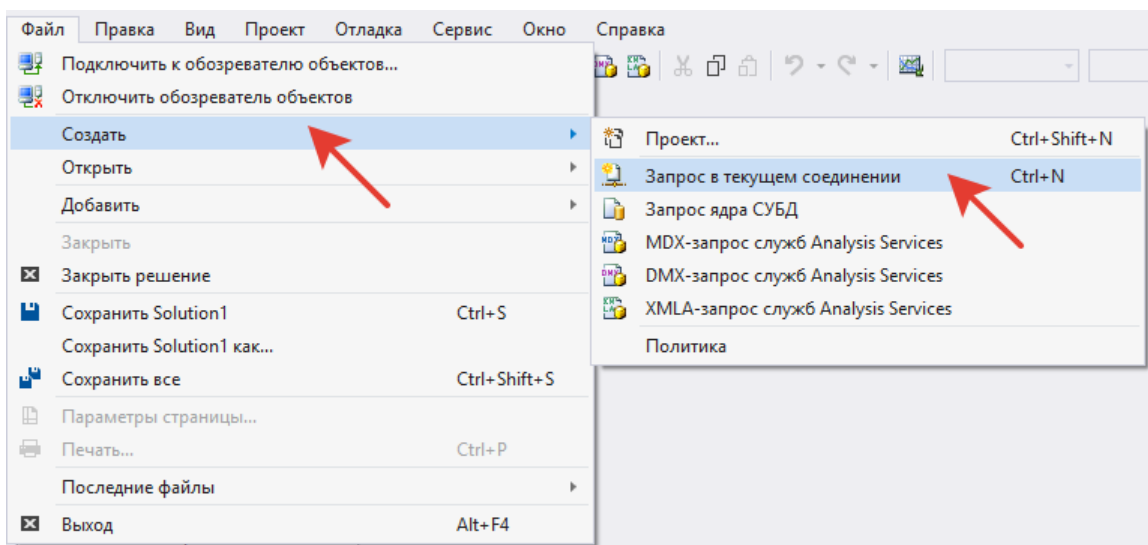
## ЗАПРОС НА ВЫБОРКУ ДАННЫХ (SELECT).

Имеет следующий синтаксис:

**SELECT** что\_выбрать **FROM** откуда\_выбрать;

Вместо "что\_выбрать" мы должны указать либо имя столбца, значения которого хотим увидеть, либо имена нескольких столбцов через запятую, либо символ звездочки (\*), означающий выбор всех столбцов таблицы. Вместо "откуда\_выбрать" следует указать имя таблицы.

Откройте учебную базу данных **aero** в среде **SQL Server Management Studio**. Выберите пункт меню **Файл** → **Создать** → **Запрос в текущем соединении** или нажмите **Ctrl+N**.



В открывшемся окне введите следующий запрос: **SELECT \* FROM Trip;**

В нижней части окна должен отобразиться результат выполнения запроса:

Результаты		Сообщения					
	trip_no	ID_comp	plane	town_from	town_to	time_out	time_in
1	1100	4	Boeing	Rostov	Paris	1900-01-01 14:30:00.000	1900-01-01 17:50:00.000
2	1101	4	Boeing	Paris	Rostov	1900-01-01 08:12:00.000	1900-01-01 11:45:00.000
3	1123	3	TU-154	Rostov	Vladivostok	1900-01-01 16:20:00.000	1900-01-01 03:40:00.000
4	1124	3	TU-154	Vladivostok	Rostov	1900-01-01 09:00:00.000	1900-01-01 19:50:00.000
5	1145	2	IL-86	Moscow	Rostov	1900-01-01 09:35:00.000	1900-01-01 11:23:00.000
6	1146	2	IL-86	Rostov	Moscow	1900-01-01 17:55:00.000	1900-01-01 20:01:00.000
7	1181	1	TU-134	Rostov	Moscow	1900-01-01 06:12:00.000	1900-01-01 08:01:00.000
8	1182	1	TU-134	Moscow	Rostov	1900-01-01 12:35:00.000	1900-01-01 14:30:00.000
9	1187	1	TU-134	Rostov	Moscow	1900-01-01 15:42:00.000	1900-01-01 17:39:00.000
10	1188	1	TU-134	Moscow	Rostov	1900-01-01 22:50:00.000	1900-01-01 00:48:00.000
11	1195	1	TU-154	Rostov	Moscow	1900-01-01 23:30:00.000	1900-01-01 01:11:00.000
12	1196	1	TU-154	Moscow	Rostov	1900-01-01 04:00:00.000	1900-01-01 05:45:00.000
13	7771	5	Boeing	London	Singapore	1900-01-01 01:00:00.000	1900-01-01 11:00:00.000
14	7772	5	Boeing	Singapore	London	1900-01-01 12:00:00.000	1900-01-01 02:00:00.000
15	7773	5	Boeing	London	Singapore	1900-01-01 03:00:00.000	1900-01-01 13:00:00.000
16	7774	5	Boeing	Singapore	London	1900-01-01 14:00:00.000	1900-01-01 06:00:00.000
17	7775	5	Boeing	London	Singapore	1900-01-01 09:00:00.000	1900-01-01 20:00:00.000
18	7776	5	Boeing	Singapore	London	1900-01-01 18:00:00.000	1900-01-01 08:00:00.000
19	7777	5	Boeing	London	Singapore	1900-01-01 18:00:00.000	1900-01-01 06:00:00.000
20	7778	5	Boeing	Singapore	London	1900-01-01 22:00:00.000	1900-01-01 12:00:00.000

Запрос успешно выполнен.

Создайте еще один запрос и введите команду: **SELECT name FROM Company**. Результат должен выглядеть следующим образом:

Результаты		Сообщения	
	name		
1	Don_avia		
2	Aeroflot		
3	Dale_avia		
4	air_France		
5	British_AW		
6	Победа		

## ЗАПРОС НА ДОБАВЛЕНИЕ ДАННЫХ В ТАБЛИЦУ

Для добавления данных используется оператор **INSERT**. Синтаксис можно использовать двух видов. Первый вариант используется для внесения данных во все поля таблицы:

**INSERT INTO** куда\_добавлять **VALUES** (список\_значений);

Второй вариант используется для внесения данных в отдельные поля таблицы:

**INSERT INTO** куда\_добавлять(список\_столбцов) **VALUES** (список\_значений);

Пример: добавим новую строку в таблицу **Company**. Для этого выполним следующий запрос:

**INSERT INTO Company VALUES (6, 'Virgin');**

**SELECT \* FROM Company;**

Результаты		Сообщения	
	ID_comp	name	
1	1	Don_avia	
2	2	Aeroflot	
3	3	Dale_avia	
4	4	air_France	
5	5	British_AW	
6	6	Virgin	

Первая команда из запроса добавляет новую авиакомпанию в список, а вторая команда выводит все содержимое таблицы в окно с результатами.

## ЗАПРОС НА УДАЛЕНИЕ ДАННЫХ ИЗ ТАБЛИЦЫ И УСЛОВИЕ WHERE

Команда для удаления элементов имеет следующий синтаксис:

**DELETEFROM**Откуда\_удалять**WHERE**условие\_удаления;

Рассмотрим пример. Выполним команду **DELETEFROMCompanyWHEREID\_comp>= 6** для базы данных **aero**. До выполнения команды таблица **Company** выглядела следующим образом:

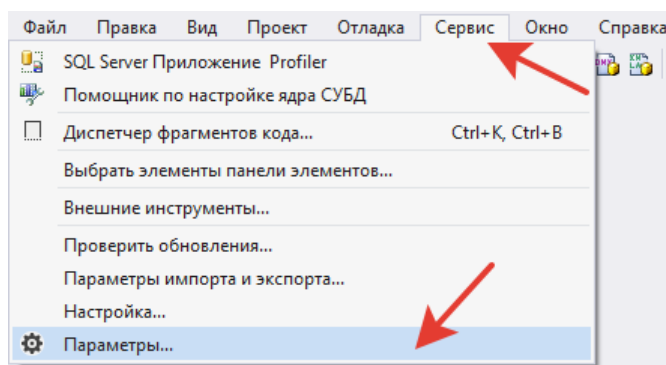
	ID_comp	name
1	1	Don_avia
2	2	Aeroflot
3	3	Dale_avia
4	4	air_France
5	5	British_AW
6	6	Virgin

После выполнения команды можно видеть, что исчезли все строки имевшие значение идентификатора **ID\_comp** больше 5.

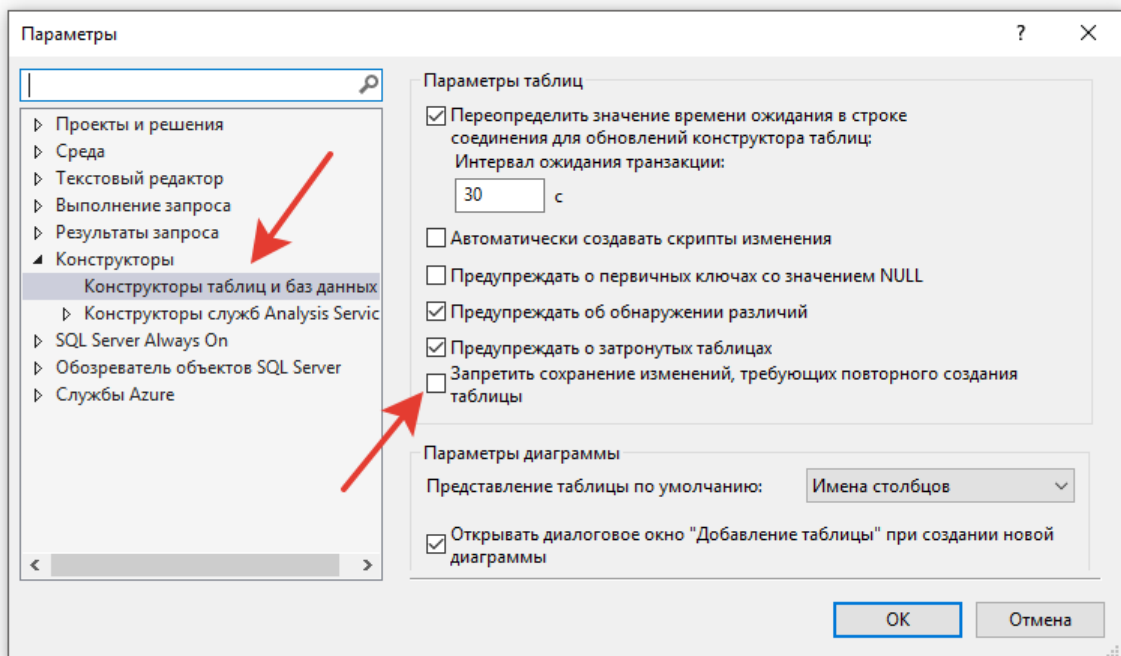
	ID_comp	name
1	1	Don_avia
2	2	Aeroflot
3	3	Dale_avia
4	4	air_France
5	5	British_AW

## МОДИФИКАЦИЯ УЧЕБНОЙ БАЗЫ ДАННЫХ

Запустите SQL Server Management Studio (в дальнейшем SSMS) и подключитесь к серверу. Для продолжения выполнения лабораторной работы нам необходимо внести некоторые изменения в настройки сервера. В меню SSMS выберите пункт **Сервис → Параметры**.

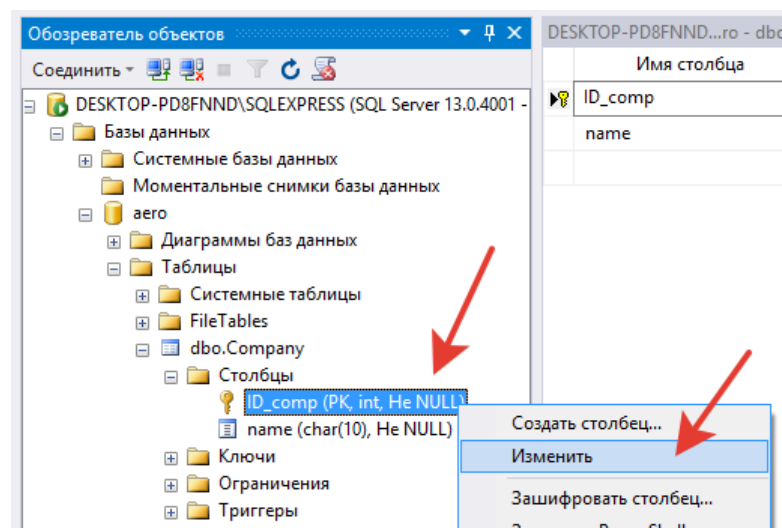


В открывшемся окне найдите пункт **Конструкторы** и откройте подпункт **Конструкторы таблиц и баз данных**. В правой части окна снимите галочку возле пункта **«Запретить сохранение изменений, требующих повторного создания таблицы»**. Это позволит нам редактировать типы и параметры столбцов в нашей базе данных.

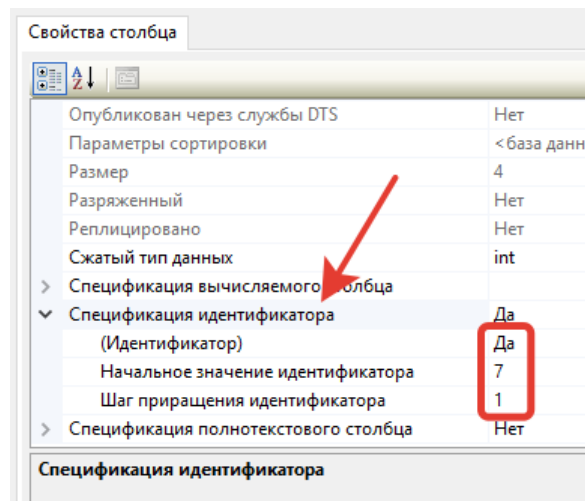


Нажмите кнопку **OK** и закройте окно.

Таблица **Company** в учебной базе данных содержит список авиакомпаний. У каждой авиакомпании есть имя **name** и идентификатор **ID\_comp**. Сейчас идентификатор является просто целым числом. Нам нужно преобразовать его в счетчик с автоматическим приращением. Для этого в **обозревателе объектов** найдите базу данных **aero**, откройте ее список таблиц, найдите таблицу **Company** и откройте для нее список столбцов. Нажмите на пункт **ID\_comp** правой кнопкой мыши и выберите пункт всплывающего меню **«Изменить»**.



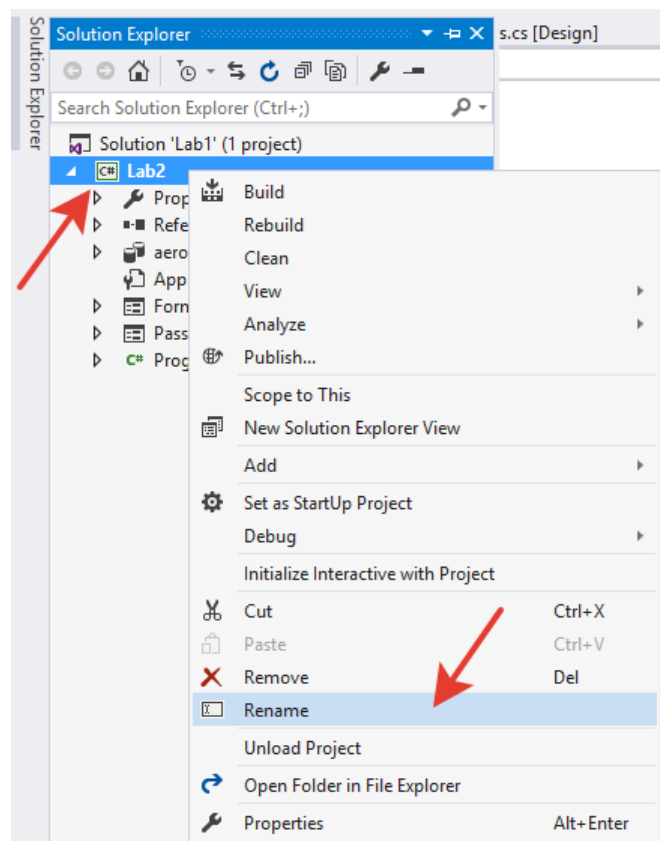
В нижней части экрана откроется окон свойств столбца. Найдите пункт **Спецификация идентификатора** и разверните его содержимое. Для подпункта **Идентификатор** поставьте значение **«Да»**. Для подпункта **Начальное значение идентификатора** выставьте число **7**, это будет начальное значение нашего счетчика. **Шаг приращения идентификатора** сделайте равным **1**.

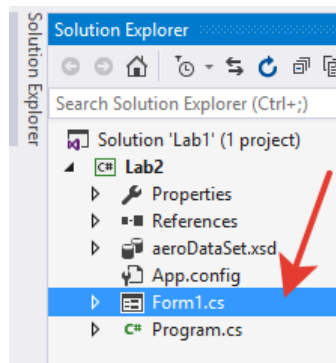


На этом работа с самой базой данных завершена.

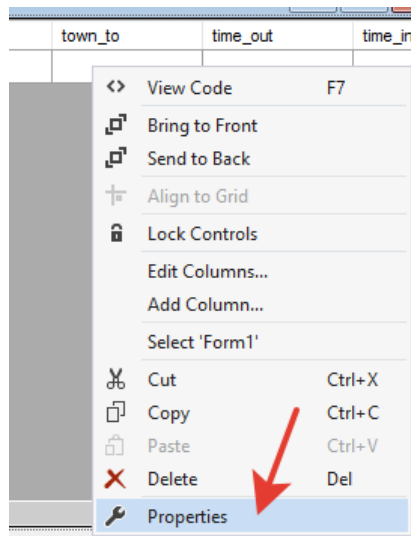
## ВЫПОЛНЕНИЕ ЗАПРОСОВ В БАЗУ ДАННЫХ

Запустите **Visual Studio** и откройте проект лабораторной работы №1. Для начала переименуем его в **Lab2**. Для этого в панели **Обозревателе решений (Solution explorer)** нажмите на узле проекта (он находится сразу под корневым узлом решения) правой кнопкой мыши и выберите пункт меню **Переименовать (Rename)**. В качестве альтернативы, можно выбрать узел проекта и нажать кнопку **F2**.

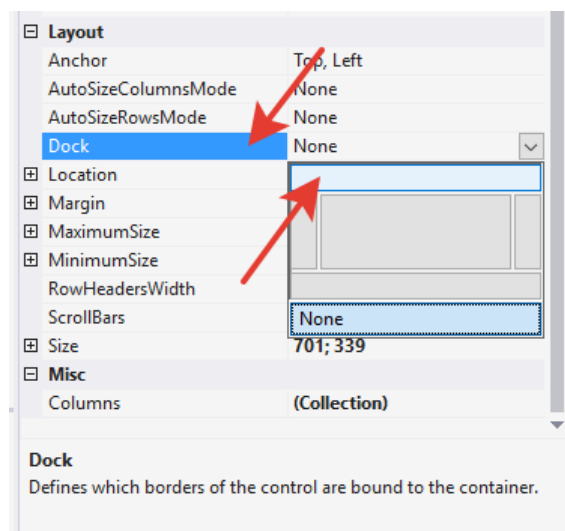




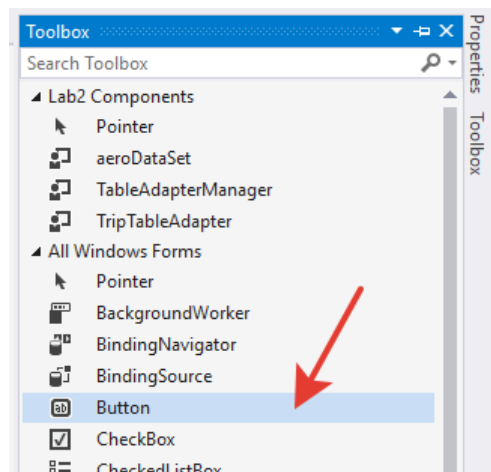
Слегка изменим дизайн нашей формы. Нажмите правой кнопкой на элементе управления **DataGridView** (таблица в центре формы) и выберите пункт **Свойства(Properties)** всплывающего меню. Также можно выбрать элемент и нажать кнопку **F4**.



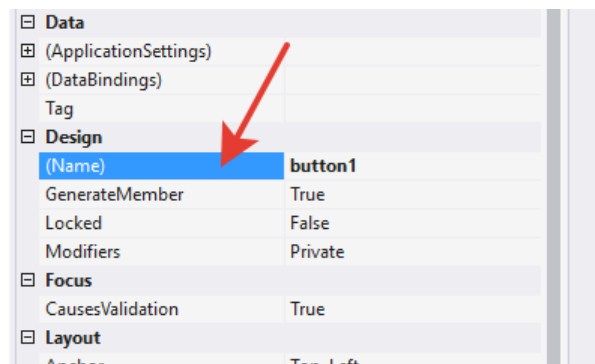
В открывшемся окне свойств элемента найдите пункт **Размещение (Dock)** и нажмите в поле справа от него. Прикрепите таблицу к верхней части окна, оставив свободное место внизу.



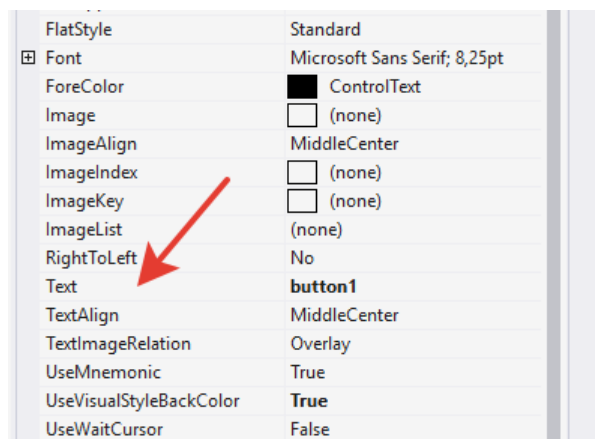
Добавим на форму кнопку. Для этого в **Панели элементов инструментов (Toolbox)** найдите элемент **Кнопка (Button)** и перетащите его на форму.



По аналогии с таблицей **DataGridView** откройте свойства кнопки. Найдите пункт **Имя (Name)** и замените **button1** на **company**. Этот пункт отвечает за название объекта класса «кнопка» в коде формы.



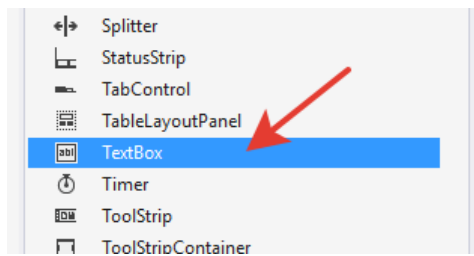
Найдите пункт **Текст (Text)** и пропишите там «**Авиакомпания**». Этот текст будет отображаться на кнопке в окне формы.



Аналогичным образом добавьте еще две кнопки:

1. Кнопка с именем **add** и текстом «**Добавить**».
2. Кнопка с именем **remove** и текстом «**Удалить изменения**».

В **Панели инструментов (Toolbox)** найдите элемент **Текстовое поле (TextBox)** и перетащите его на форму.



В свойствах элемента **TextBox** найдите пункт **Максимальный размер (MaxLength)** и установите его значение в 10 символов.

ContextMenuStrip	(none)
Enabled	True
HideSelection	True
ImeMode	NoControl
MaxLength	10
Multiline	False
PasswordChar	
ReadOnly	False

Измените пункт **Имя (Name)** с **textBox1** на **name**.

(ApplicationSettings)	
(DataBindings)	
Tag	
<b>Design</b>	
(Name)	name
GenerateMember	True
Locked	False
Modifiers	Private
<b>Focus</b>	
CausesValidation	True

После выполнения этих действий форма может выглядеть примерно так:

ID_comp	name
1	Don_avia
2	Aeroflot
3	Dale_avia
4	air_France
5	British_AW
*	

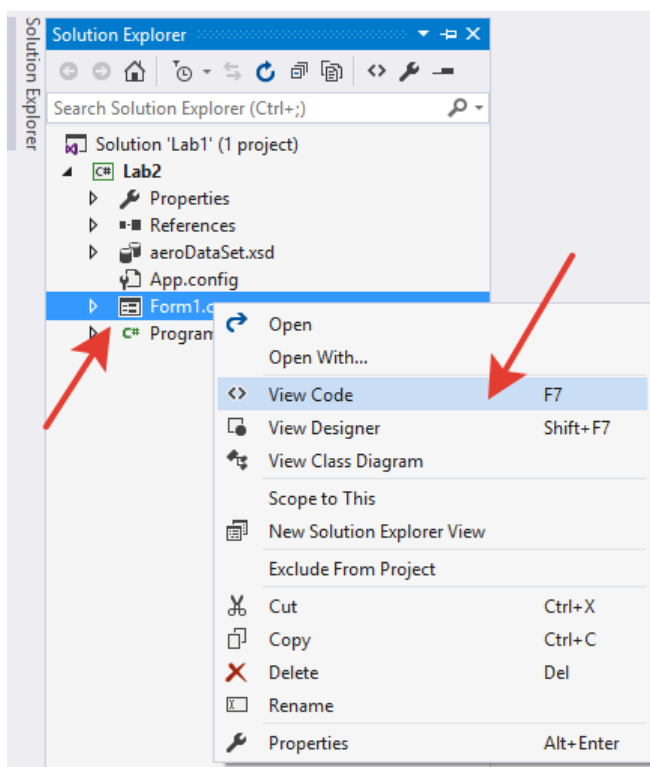
Авиакомпании

Дважды щелкните на кнопке с надписью «**Авиакомпании**». Откроется окно с кодом формы.

Переключитесь обратно на редактируемую форму и дважды щелкните на кнопке «**Добавить**». Повторите это действие для кнопки «**Удалить изменения**».



Если вы случайно закрыли закладку с формой, то повторно ее можно открыть при помощи панели **Обозреватель решений**. Найдите в нем вашу форму **Form1** и дважды щелкните на ней мышкой. Если вы хотите открыть окно с кодом формы, то щелкните на узле формы правой кнопкой мыши и выберите пункт **Показать код (Viewcode)**. Еще можно выбрать узел формы и нажать кнопку **F7**.



Каждый раз, когда мы дважды щелкаем по кнопке в код формы, добавляется новый обработчик события нажатия на кнопку. Теперь в коде формы должно появиться три новых метода: **company\_Click**, **add\_Click** и **remove\_Click**.

Весь код формы должен выглядеть следующим образом:

```
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;

namespace Lab1 {
    public partial class Form1 : Form {
        public Form1() {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e) {
            this.tripTableAdapter.Fill(this.aeroDataSet.Trip);
        }

        private void company_Click(object sender, EventArgs e) {
        }

        private void add_Click(object sender, EventArgs e) {
        }

        private void remove_Click(object sender, EventArgs e) {
        }
    }
}
```

Как видите, форма представляет собой класс с набором полей и методов. Часть свойств формы скрыта от программиста и редактируется в режиме визуального конструирования.

Добавим в нашу форму код для выполнения действий над базой данных. Замените содержимое файла на следующее:

```
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 Lab1 {
    public partial class Form1 : Form {
        SqlConnection connection;
        SqlCommand command;
        SqlDataAdapter adapter;
        DataTable table;

        public Form1() {
            InitializeComponent();

            dataGridView1.AutoGenerateColumns = true;

            // УКАЖИТЕ СВОЙ АДРЕС СЕРВЕРА ИЗ СПЕДЫ ССМС!
            connection = new SqlConnection("Server=DESKTOP-PD8FNND\\SQLEXPRESS;Database=aero;Trusted_Connection=True;");

            command = new SqlCommand();
            command.Connection = connection;
            command.CommandType = CommandType.Text;

            adapter = new SqlDataAdapter(command);
            table = new DataTable();

            dataGridView1.DataSource = table;
        }

        private void Form1_Load(object sender, EventArgs e) {
            this.tripTableAdapter.Fill(this.aeroDataSet.Trip);
        }

        private void ShowTable(string text) {
            dataGridView1.Columns.Clear();
            dataGridView1.DataSource = null;

            command.CommandText = text;

            table.Clear();
            adapter.Fill(table);

            dataGridView1.DataSource = table;
        }

        private void company_Click(object sender, EventArgs e) {
            ShowTable("SELECT * FROM Company");
        }

        private void add_Click(object sender, EventArgs e) {
            connection.Open();
            command.CommandText = "INSERT INTO Company VALUES ('" + name.Text + "')";
            command.ExecuteReader();
            connection.Close();

            ShowTable("SELECT * FROM Company");
        }
    }
}
```

```
private void remove_Click(object sender, EventArgs e) {
    connection.Open();
    command.CommandText = "DELETE FROM Company WHERE ID_comp >= 6";
    command.ExecuteReader();
    connection.Close();

    ShowTable("SELECT * FROM Company");
}
}
```

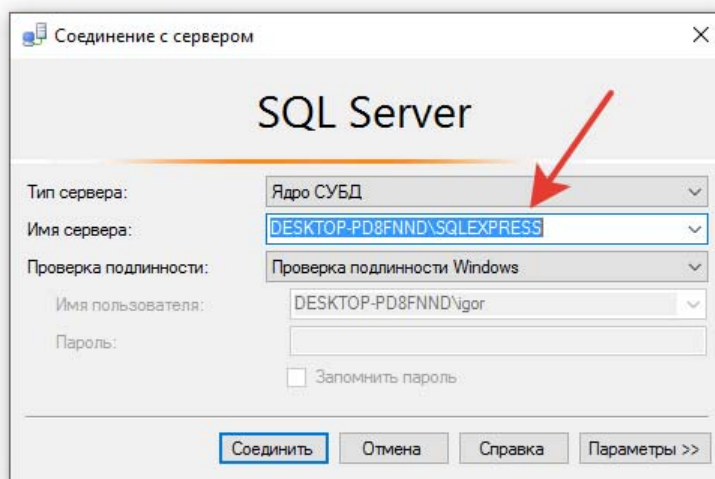
Рассмотрим этот код более подробно. В самом начале файла мы дополнительно подключаем пространство имен **System.Data.SqlClient**. Оно позволяет нам использовать классы и функции для работы с сервером MS SQL.

В классе формы появляются новые поля:

- Объект **connection** класса **SqlConnection** используется для установления соединения с базой данных.
- Объект **command** класса **SqlCommand** используется для выполнения текстовых запросов SQL.
- Объект **adapter** класса **SqlDataAdapter** занимается расшифровкой и преобразованием сообщений, полученных от базы данных.
- Объект **table** класса **DataTable** используется для хранения полученных данных в виде таблицы.

В конструкторе формы происходит создание и инициализация объектов. Следует обратить внимание на следующие моменты:

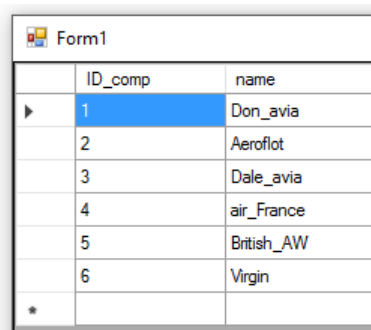
- При создании объекта **connection** нужно указать имя сервера SQL. Имя сервера отличается от компьютера к компьютеру, поэтому вам потребуется прописать его самостоятельно.
  - Откройте среду **SSMS**.
  - В главном меню выберите пункт **Файл → Подключить к обозревателю объектов**
  - Скопируйте имя сервера и вставьте в код создания объекта **connection**.



- Объект **adapter** использует объект **command**, а объект **command** использует объект **connection**. В языке C# схожим образом организована работа с файлами.

Для отображения таблицы со списком авиакомпаний используется вспомогательный метод **ShowTable**. Внутри этой функции сначала очищается элемент управления **DataGridView**, затем передаваемым текстом инициализируется команда **command**, очищается табличное хранилище **table** и в конце выполняется сама команда посредством вызова метода **Fill** объекта **adapter**. Результаты выполнения команды **SQL** сохраняются в объекте **table**.

Метод **company\_Click** вызывается при нажатии на кнопку «Авиакомпании». При этом выполняется запрос на чтение содержимого всей таблицы из базы данных:

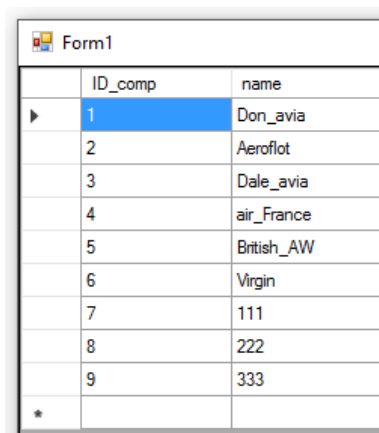


	ID_comp	name
▶	1	Don_avia
	2	Aeroflot
	3	Dale_avia
	4	air_France
	5	British_AW
	6	Virgin
*		

Метод **add\_Click** вызывается при нажатии на кнопку «Добавить». Сначала выполняется подключение к базе данных **connection.Open()**. Затем берется содержимое текстового поля **name** и встраивается в SQL запрос:

**"INSERT INTO Company VALUES ('" + name.Text + "');"**

После выполнения этого запроса в таблицу **Company** добавляется новая строка с введенным текстом.



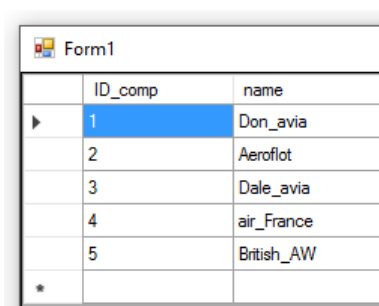
	ID_comp	name
▶	1	Don_avia
	2	Aeroflot
	3	Dale_avia
	4	air_France
	5	British_AW
	6	Virgin
	7	111
	8	222
	9	333
*		

Выполнение команды по вставке не вызывает обновления таблицы на форме. Нам необходимо произвести его вручную при помощи вызова нашей вспомогательной функции **ShowTable** с запросом на выборку всего содержимого таблицы.

Метод **remove\_Click** вызывается при нажатии на кнопку «Удалить изменения». Его структура аналогична структуре метода **add\_Click**, разница заключается только в выполняемой команде:

**DELETE FROM Company WHERE ID\_comp >= 6**

Данная команда SQL удалит все строки из таблицы **Company**, при условии, что идентификатор компании **ID\_comp** будет больше или равен шести:



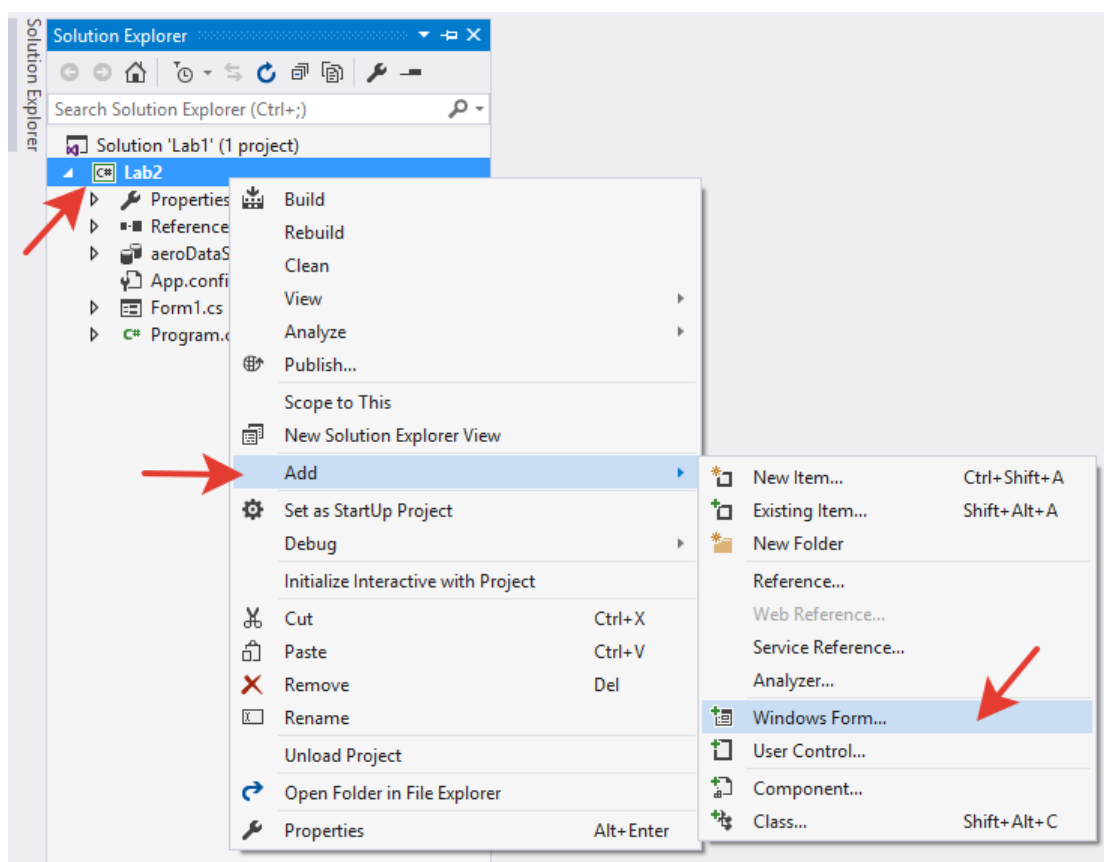
	ID_comp	name
▶	1	Don_avia
	2	Aeroflot
	3	Dale_avia
	4	air_France
	5	British_AW
*		

## ДОПОЛНИТЕЛЬНОЕ ЗАДАНИЕ

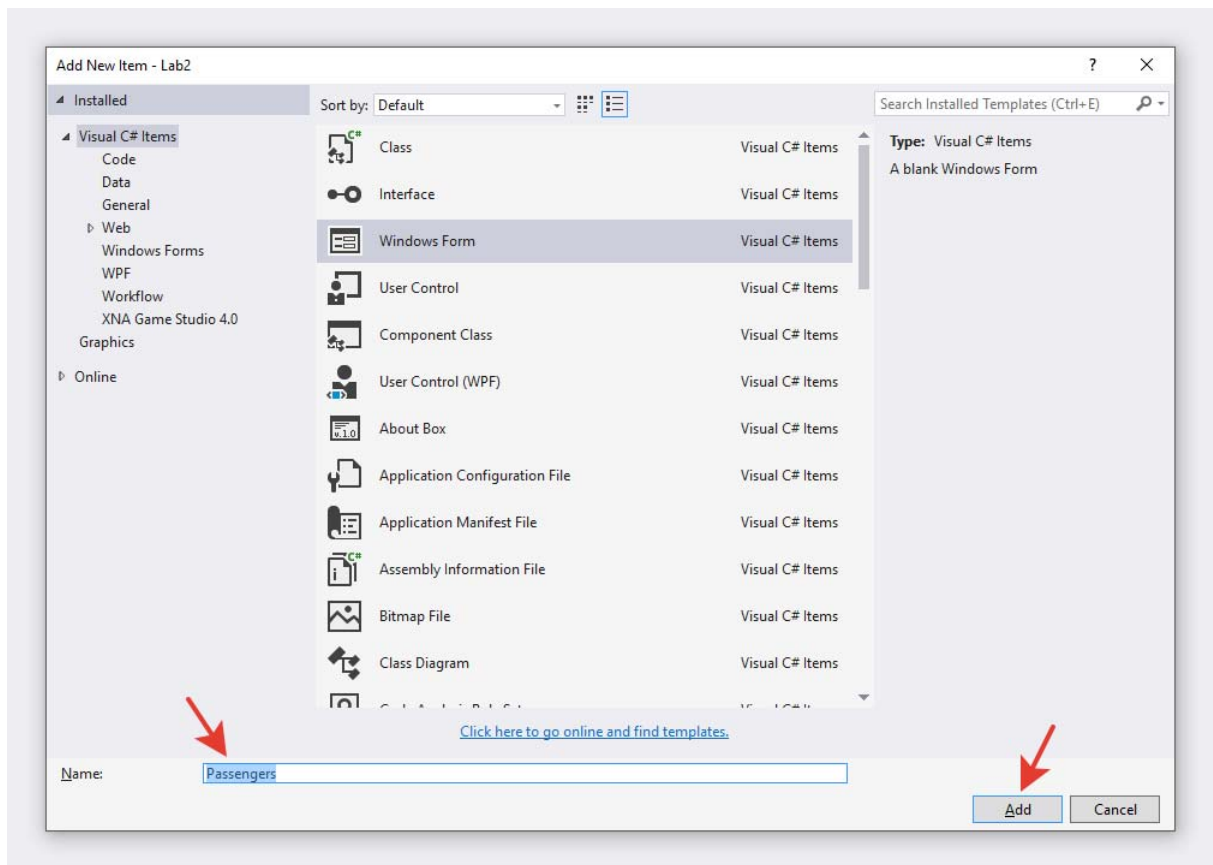
По аналогии с параграфом «**Модификация учебной базы данных**» добавьте полю **ID\_psg** таблицы **Passenger** возможность автоматического приращения.

Для подпункта **Начальное значение идентификатора** выставьте число 7, это будет начальное значение нашего счетчика. **Шаг приращения идентификатора** сделайте равным 1.

Добавьте в проект C# новую форму. Для этого щелкните правой кнопкой мыши на ветке проекта в **обозревателе решений**. Выберите пункты **Добавить**→**Форма Windows** всплывающего меню.



В открывшемся диалоговом окне введите название новой формы – **Passengers**(Пассажиры) и нажмите кнопку **Добавить**.



При помощи панели Обозревателя решений откройте файл **Program.cs**. Найдите строку

**Application.Run(new Form1());**

и замените ее на:

**Application.Run(new Passenger());**  
**Application.Run(new Passengers());**

Теперь при запуске программы вместо старой формы **Form1** будет открываться новая форма **Passenger**.

Задание:

- По аналогии с лабораторной работой №1 разместите на форме элемент управления **DataGridView**.
- Реализуйте на форме элементы управления для работы с таблицей **Passenger**:
  - Добавление новой строки.
  - Вывод списка пассажиров **без столбца с идентификатором пассажира** (только имена).
  - Удаление изменений.
- Сохранение проекта:
  - В меню «Файл» выберите пункт «Сохранить Form1.cs как...».
  - Выберите в качестве места сохранения свою флэш карту.
  - Нажмите кнопку «Сохранить».