

# ТЕХНОЛОГИЯ РАБОТЫ С БАЗАМИ ДАННЫХ ADO .NET

## Практическая работа «Загрузка локальной базы данных MySQL и работа с ней. Запрос Select»

### Описание приложения

Компания работает в сфере ИТ-услуг. Необходимо создать базу данных, которая должна содержать следующие объекты:

Должности – код, название, оклад

Сотрудники – код, имя, фамилия, пол, код должности

Проекты – код, название проекта

Управление проектами – код, код проекта, код сотрудника.

Приложение должно управлять базой данных (изменять, добавлять, удалять, группировать данные) и формировать два отчета:

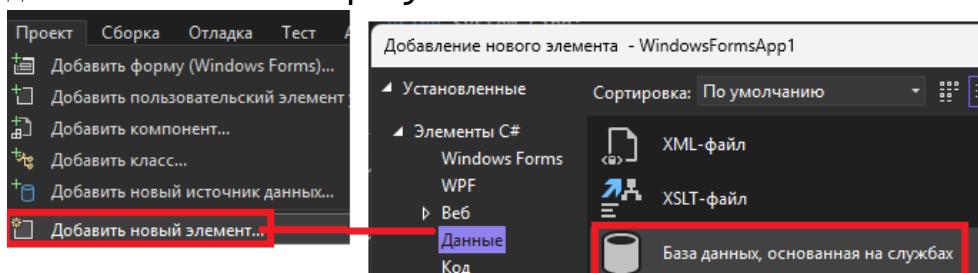
**Запрос 1** – найти количество сотрудников и сумму всех окладов

**Запрос 2** – найти количество сотрудников, участвующих в каждом проекте.

### Задание 1. Создание базы данных

**1.1** Создать папку с названием **Company**, а в ней **Приложение Windows Forms(.NET Framework)** с названием **BD**.

**1.2** Добавить в приложение новый элемент в виде пустой базы данных с названием **company.mdf**.



**1.3** Перейти к базе данным через обозреватель решений (дважды щелкнуть по названию базы данных). База данных должна содержать сведения о сотрудниках и проектах, над которыми они трудятся согласно прилагаемой схемы.

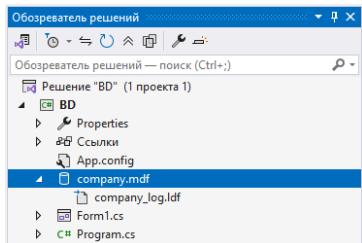
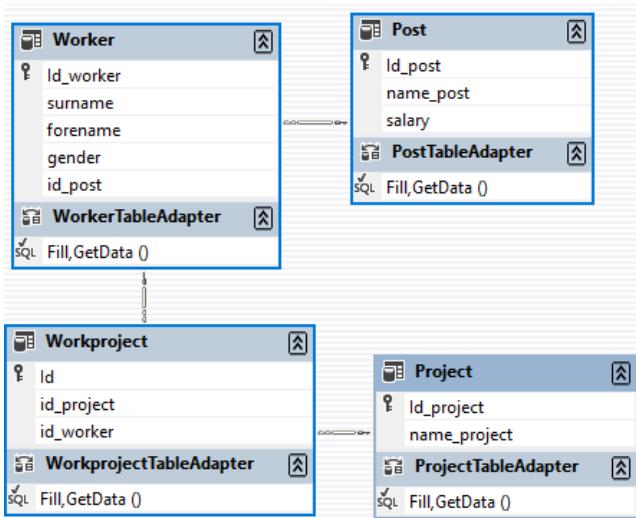


Схема базы данных **Company .mdf**

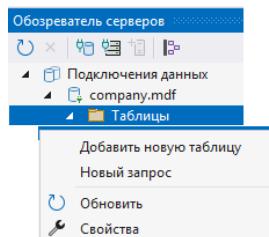


**1.4** Добавить в базу данных новую таблицу через правую кнопку мыши. Название таблицы **Post**, таблица содержит сведения о должностях компании:

Код должности **id\_post** (счетчик)

Название должности **name\_post**

Оклад - **salary**



Properties window for the 'id\_post' column:

- Файл скрипта: dbo.Table.sql\*
- Обновить (highlighted with a red box)
- Имя: id\_post
- Тип данных: int
- Разрешить значения NULL:
- Значение по умолч.:
- Классификация дан.: Набор столбцов: False
- Нет для репликации: False
- Описание: Параметры сортировки: True
- Первичный ключ: True
- Разрешить значения: False
- Разряженный: False
- Спецификация ид.: True (highlighted with a red box)
- Идентификатор: True (highlighted with a red box)
- Начальное значение: 1
- Шаг приращения: 1

Script tab (T-SQL):

```

CREATE TABLE [dbo].[Post]
(
    [id_post] INT NOT NULL PRIMARY KEY IDENTITY,
    [name_post] NCHAR(30) NULL,
    [salary] MONEY NULL
)
  
```

**Запустить скрипт через команду ОБНОВИТЬ, обновить базу.**

**1.5** Добавить в базу данных новую таблицу

Название таблицы – **Worker**, таблица хранит сведения о работниках

Код работника- **id\_worker**

Фамилия - **surname**

Имя - **forename**

Пол - **gender**

Код должности – **id\_post** – внешний ключ для связи с таблицей Post

Properties window for the 'Worker' table:

- Файл скрипта: dbo.Table\_1.sql\*
- Обновить (highlighted with a red box)
- Имя: Worker
- Тип данных: dbo.Table\_1.sql
- Разрешить значения NULL:
- По умолчанию:
- Ключи (1)
  - <Без имени> (Первичный ключ, Clustered: id\_worker)
- Проверочные ограничения (0)
- Индексы (0)
- Внешние ключи (1)
  - FK\_Worker\_ToPost
- Триггеры (0)

Script tab (T-SQL):

```

CREATE TABLE [dbo].[Worker]
(
    [id_worker] INT NOT NULL PRIMARY KEY IDENTITY,
    [surname] NCHAR(20) NULL,
    [forename] NCHAR(20) NULL,
    [gender] BIT NULL,
    [id_post] INT NULL,
    CONSTRAINT [FK_Worker_ToPost] FOREIGN KEY ([id_post]) REFERENCES [Post]([id_post])
)
  
```

**Запустить скрипт через команду ОБНОВИТЬ**

## 1.6 Добавить в базу данных новую таблицу **Project**

**id\_project** - код проекта

**name\_project** – название проекта

The screenshot shows the 'Object Explorer' on the left with 'Проектирование' selected. The 'Scripting' tab is active, showing the T-SQL script for creating the 'Project' table. The table has two columns: 'id\_project' (int, primary key, identity) and 'name\_project' (nchar(50), nullable).

Имя	Тип данных	Разрешить значения NULL	По умолчанию
id_project	int	<input type="checkbox"/>	
name_project	nchar(50)	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

```
CREATE TABLE [dbo].[Project]
(
    [id_project] INT NOT NULL PRIMARY KEY IDENTITY,
    [name_project] NCHAR(50) NULL
)
```

## 1.7 Добавить для управления проектами в базу данных новую таблицу **Workproject** с двумя внешними ключами.

**id** – код записи

**id\_project** - код проекта

**id\_worker** – код сотрудника

The screenshot shows the 'Object Explorer' on the left with 'Проектирование' selected. The 'Scripting' tab is active, showing the T-SQL script for creating the 'Workproject' table. The table has one primary key column 'id' (int, primary key, identity) and two foreign key columns 'id\_project' and 'id\_worker' (int, nullable). It includes two foreign key constraints: 'FK\_Workproject\_Project' linking to 'Project(id\_project)' and 'FK\_Workproject\_Worker' linking to 'Worker(id\_worker)'.

Имя	Тип данных	Разрешить значения NULL	По умолчанию
id	int	<input type="checkbox"/>	
id_project	int	<input checked="" type="checkbox"/>	
id_worker	int	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Ключи (1)  
<без имени> (Первичный ключ, Clustered: id)  
Проверочные ограничения (0)  
Индексы (0)  
Внешние ключи (2)  
FK\_Workproject\_Project (id\_project)  
FK\_Workproject\_Worker (id\_worker)  
Триггеры (0)

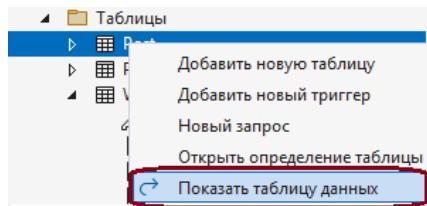
```
CREATE TABLE [dbo].[Workproject]
(
    [id] INT NOT NULL PRIMARY KEY IDENTITY,
    [id_project] INT NULL,
    [id_worker] INT NULL,
    CONSTRAINT [FK_Workproject_Project] FOREIGN KEY ([id_project]) REFERENCES [Project]([id_project]),
    CONSTRAINT [FK_Workproject_Worker] FOREIGN KEY ([id_worker]) REFERENCES [Worker]([id_worker])
)
```

## 1.8 Добавить в таблицы данные, согласно схеме данных

dbo.Post [Данные] Form1.cs [Конструктор]

Максимальное количество строк: 1000

	id_post	name_post	salary
1	менеджер	...	60000,0000
2	техник	...	50000,0000
3	программист	...	70000,0000
4	помощник	...	45000,0000



dbo.Worker [Данные] Form1.cs [Конструктор]

Максимальное количество строк: 1000

	id_worker	surname	forename	gender	id_post
1	Иванов	Сергей	True	1	
2	Петров	Андрей	True	2	
3	Зуев	Петр	True	2	
4	Агеева	Светлана	False	3	
5	Борисов	Алексей	True	3	
6	Волкова	Ольга	False	4	

dbo.Project [Данные] dbo.Workproject [Данные]

Максимальное количество строк: 1000

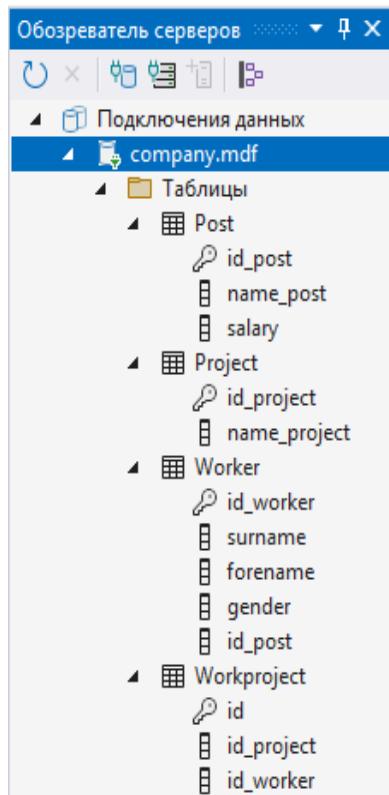
	id_project	name_project
1	Сайт для ООО "Темп"	...
2	Видеоролик для ИП Дроздова	...
3	Приложение "Такси"	...
4	Интернет-магазин	...

dbo.Workproject [Данные]

Максимальное количество строк: 1000

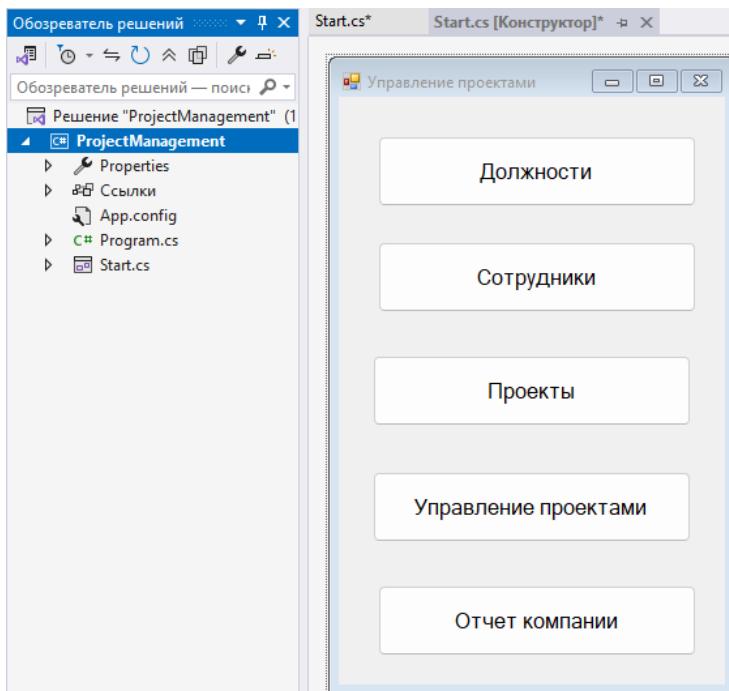
	id	id_project	id_worker
1	1	1	1
2	1	1	2
3	1	1	4
4	2	1	1
5	2	2	3
6	2	2	5
7	3	1	1
8	3	3	5
9	3	3	6
10	4	1	1
11	4	4	3
12	4	4	4
13	4	4	6

**1.9** Обновить базу данных и закрыть приложение. Отследить месторасположение базы данных **company.mdf**.



## Задание 2. Создание подключения к базе данных

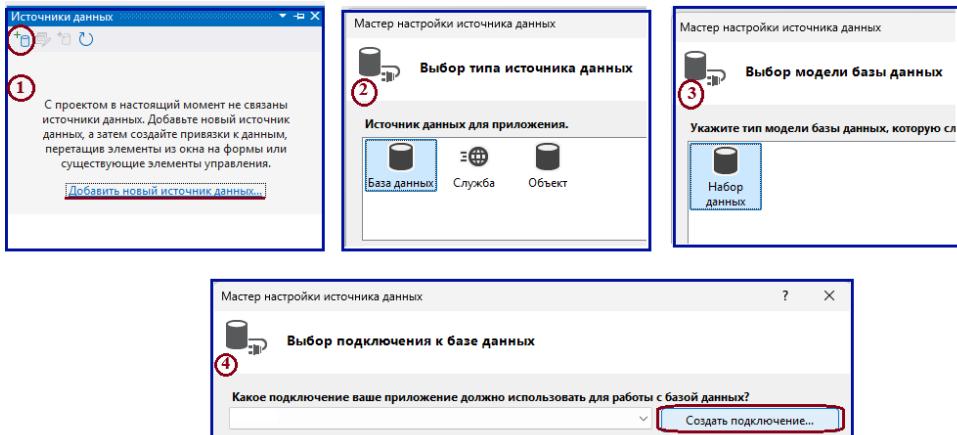
**2.1** Создать в папке **Company** новый проект под названием **ProjectManagement**. Переименовать первую форму проекта, дав название **Start**. Расположить элементы на форме.



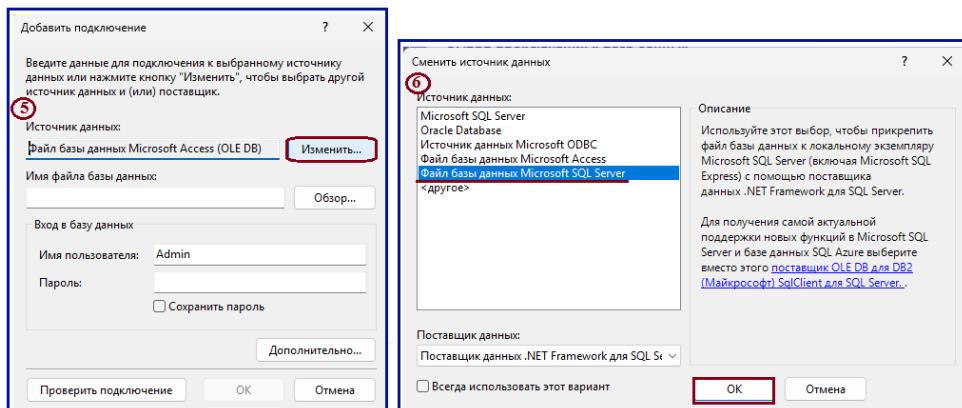
**2.2** Добавить в проект источник данных

**ВИД – ДРУГИЕ ОКНА – ИСТОЧНИК ДАННЫХ**

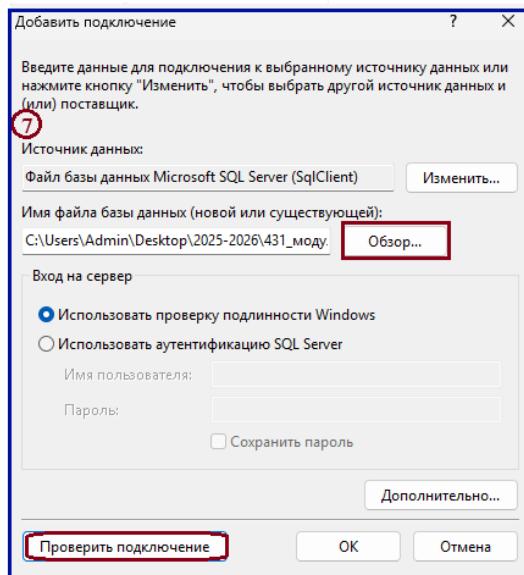
## 2.3 Создать новое подключение в качестве источника выбрать базу данных и набор данных



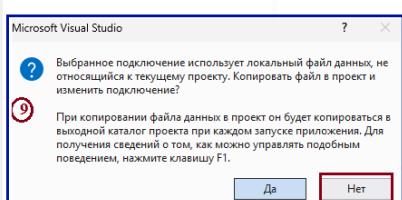
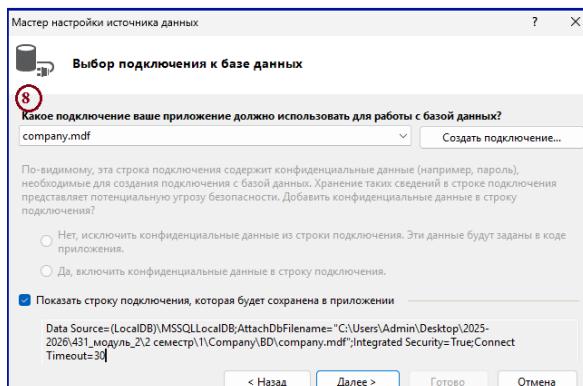
## 2.4 Изменить источник данных на файл базы данных MSQl



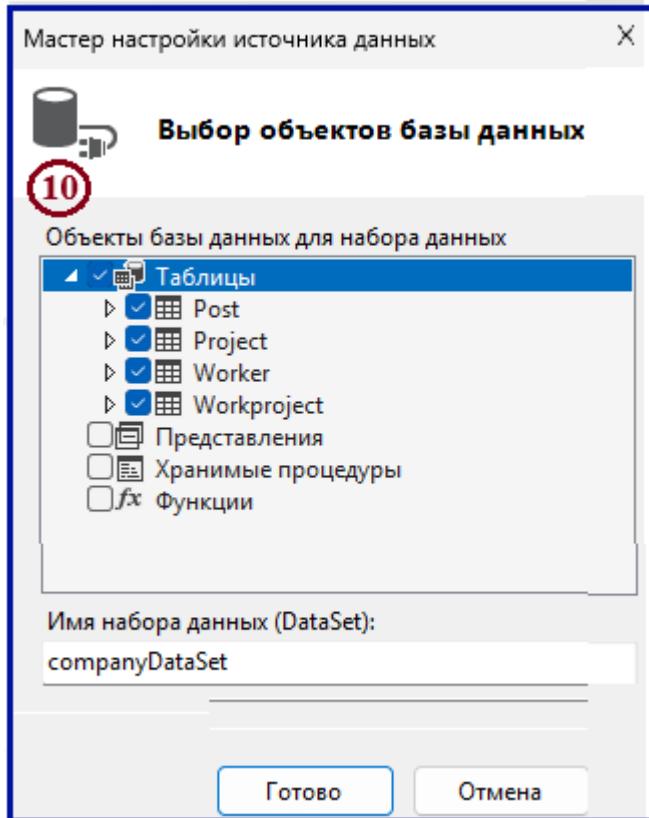
**2.5** Через обзор выбрать файл базы данных **company.mdf** и проверить подключение к файлу



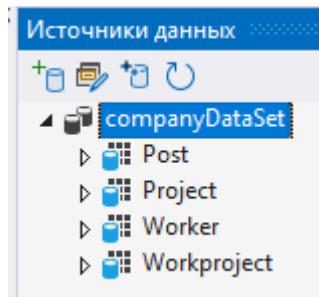
**2.6** Проверить строку подключения и дать отрицательный ответ на предложение копировать файл базы данных в выходной каталог



## 2.7 Выбрать объекты базы данных для подключения



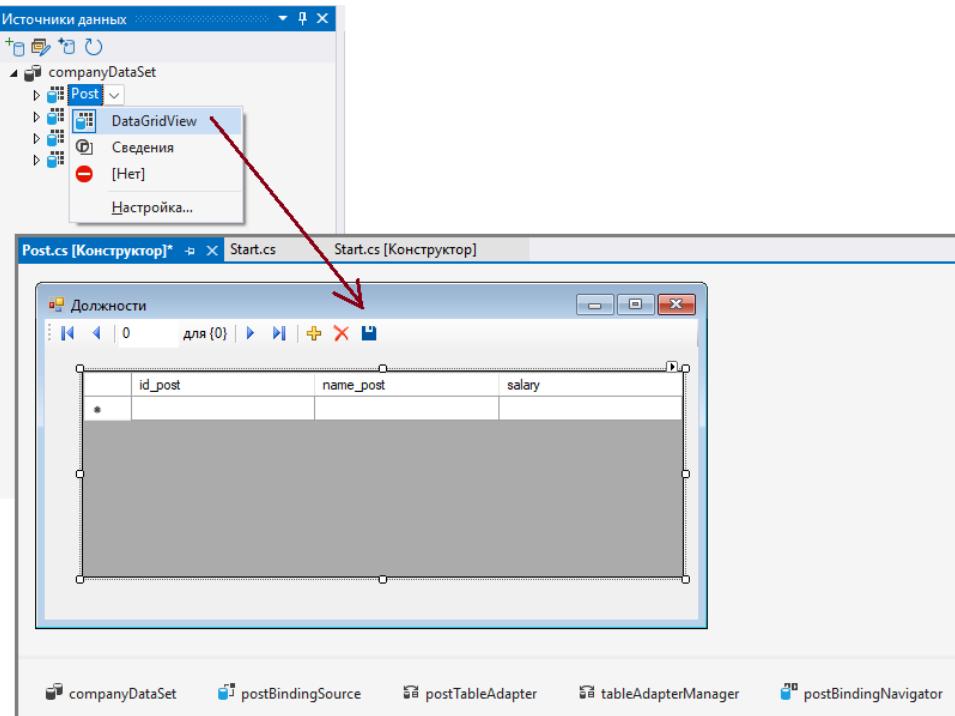
## 2.8 Открыть ИСТОЧНИК ДАННЫХ и убедиться, что база данных подсоединенна.



### Задание 3. Управление таблицей должностей Post

3.1 Для управления таблицей должностей Post создать новую одноименную форму Post.

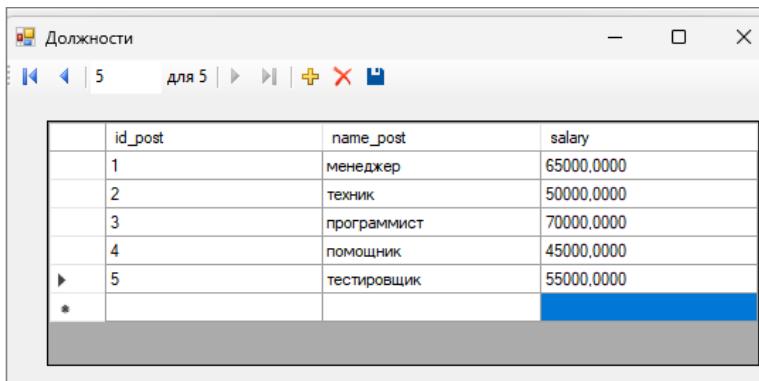
3.2 Из ИСТОЧНИКА ДАННЫХ перенести таблицу Post на одноименную форму.



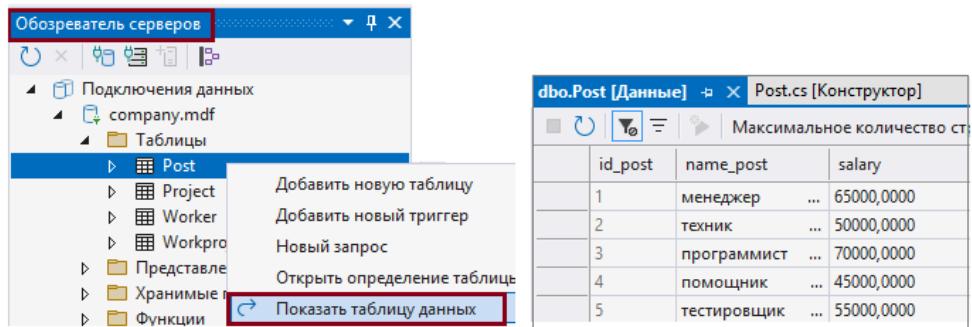
3.3 Обеспечить переход с главной формы на форму Post при нажатии на кнопку ДОЛЖНОСТИ.

```
private void button1_Click(object sender, EventArgs e)
{
    Post form = new Post();
    form.Show();
}
```

**3.4** Протестировать приложение с переходом на форму должностей. Добавить новую должность, изменить оклад у менеджера. Убедиться, что данные в таблице базы данных обновились.



	id_post	name_post	salary
	1	менеджер	65000,0000
	2	техник	50000,0000
	3	программист	70000,0000
	4	помощник	45000,0000
▶	5	тестировщик	55000,0000
*			



Обозреватель серверов

- Подключения данных
  - company.mdf
    - Таблицы
      - Post
      - Project
      - Worker
      - Workpro
    - Представле
    - Хранимые
    - Функции

Показать таблицу данных

id_post	name_post	salary
1	менеджер	65000,0000
2	техник	50000,0000
3	программист	70000,0000
4	помощник	45000,0000
5	тестировщик	55000,0000

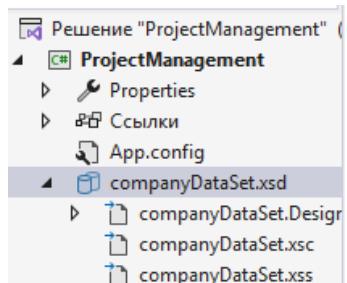
**Замечание.** При нажатии на кнопки удаления и сохранения происходит конфликт, т.к. есть записи в таблице Worker, которые ссылаются на код должности. Можно обработать исключение

```
private void postBindingNavigatorSaveItem_Click(object sender, EventArgs e)
{
    try
    {
        this.Validate();
        this.postBindingSource.EndEdit();
        this.tableAdapterManager.UpdateAll(this.companyDataSet);
    }
    catch (Exception) // Обрабатываем возникшее исключение
    {
        MessageBox.Show("Нельзя удалить связанные данные", "Ошибка",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

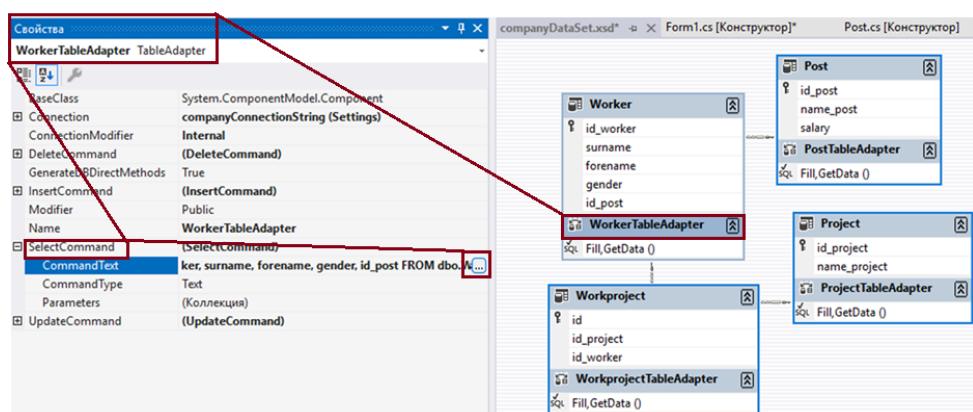
## Задание 4. Управление таблицей сотрудников Worker.

4.1 Для управления таблицей сотрудников **Worker** создать новую форму **Worker**.

4.2 Для того, чтобы форма отображала не только код должности, но и название должности, необходимо отредактировать команду Select для адаптера таблицы. В обозревателе решений найти и открыть файл **companyDataSet.xsd**



4.3 Вызвать свойства **WorkerTableAdapter** таблицы **Worker**. Выберите в свойствах поле **SelectCommand** и активизируйте редактор текста команды



4.4 Через конструктор запросов добавить таблицу **Post** (правая кнопка мыши) и выбрать поле **name\_post** для запроса.

Конструктор запросов

Worker

Post

Столбец	Псевдо...	Таблица	Вывод	Тип сортиро...	Порядок сор...	Фильтр	Или.
<input checked="" type="checkbox"/> id_worker		Worker	<input checked="" type="checkbox"/>				
<input type="checkbox"/> surname		Worker	<input checked="" type="checkbox"/>				
<input type="checkbox"/> forename		Worker	<input checked="" type="checkbox"/>				
<input type="checkbox"/> gender		Worker	<input checked="" type="checkbox"/>				
<input type="checkbox"/> id_post		Worker	<input checked="" type="checkbox"/>				
<input type="checkbox"/> name_post		Post	<input checked="" type="checkbox"/>				

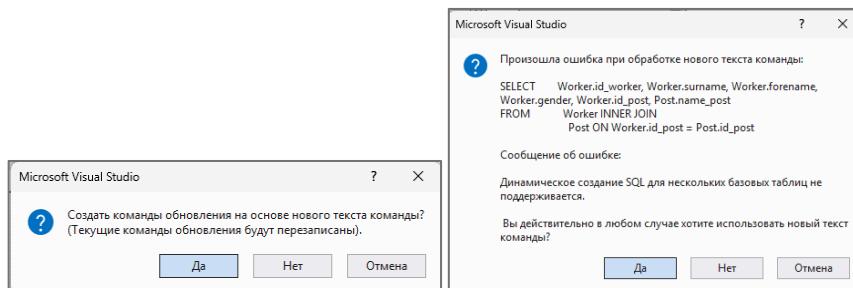
```
SELECT Worker.id_worker, Worker.surname, Worker.forename, Worker.gender, Worker.id_post, Post.name_post
FROM Worker INNER JOIN
Post ON Worker.id_post = Post.id_post
```

Выполнить запрос

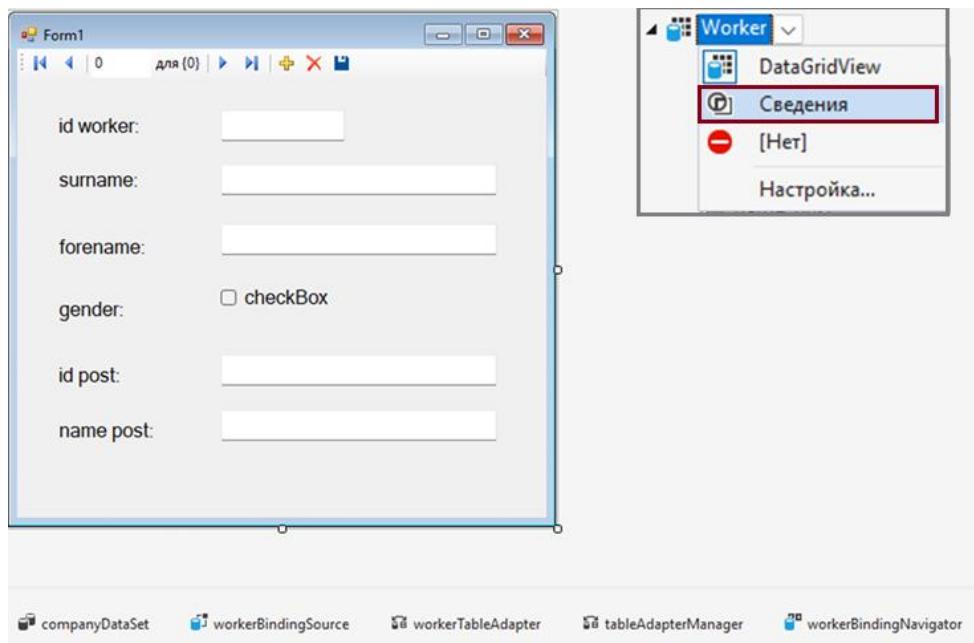
OK

Отмена

**4.5** При закрытии конструктора будет предложено обновить команду и использовать новый текст команды. Закрыть окно со схемой данных.



**4.6** Из **ИСТОЧНИКА ДАННЫХ** перенести таблицу **Worker** в форме сведений на одноименную форму.



**Установите для id\_worker и name\_post свойство Unabled=False**

**4.7** Обеспечить переход с главной формы на форму **Worker** при нажатии на кнопку **СОТРУДНИКИ**.

```
private void button2_Click(object sender, EventArgs e)
{
    Worker form = new Worker();
    form.Show();
}
```

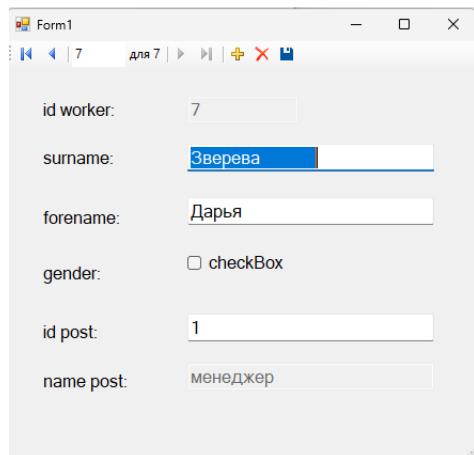
**4.8** Протестировать приложение с переходом на форму сотрудников, добавить одного сотрудника.

**Замечание.** При добавлении новой записи или изменения поля **name\_post** обновляется только после закрытия формы.

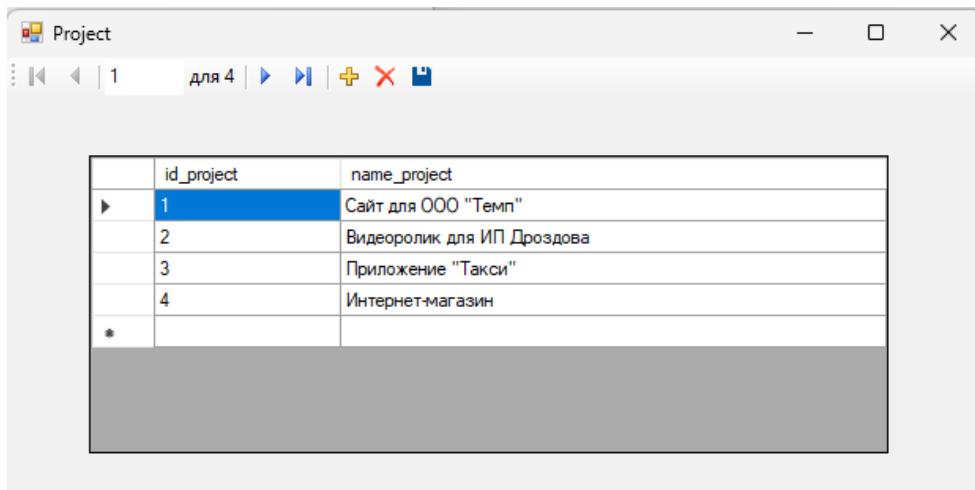
Для мгновенного обновления необходимо вызвать процедуру нажатия на кнопку сохранения двойным нажатием мыши и добавить команду на заливку данных в таблицу.

```
private void workerBindingNavigatorSaveItem_Click_1(object
sender, EventArgs e)
{
    this.Validate();
    this.workerBindingSource.EndEdit();
    this.tableAdapterManager.UpdateAll(this.companyDataSet);
    this.workerTableAdapter.Fill(this.companyDataSet.Worker);
}
```

**4.9** Обработать самостоятельно исключение при нажатии на кнопку **СОХРАНИТЬ**, как это было предложено для таблицы **Post**



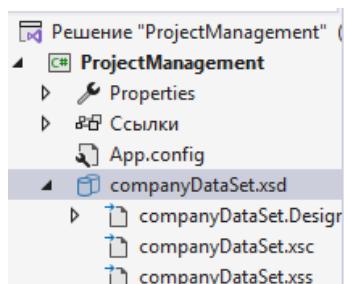
**Задание 5.** Самостоятельно обеспечить управление таблицей проекты Project.



**Задание 6.** Обеспечить управление таблицей Workproject

**6.1** Для управления таблицей Workproject создать новую форму Workproject .

**6.2** Для того, чтобы форма отображала не только коды, но и название проекта, участников проекта и их должности, необходимо отредактировать команду Select для адаптера таблицы Workproject . В обозревателе решений найти и открыть файл companyDataSet .xsd



**6.3** Вызвать свойства **WorkerTableAdapter** таблицы **Workproject**. Выберите в свойствах поле **SelectCommand** и активизируйте редактор текста команды.

**6.4** Через конструктор запросов добавить таблицы **Post**, **Worker**, **Project** (правая кнопка мыши), выбрать поля **name\_project**, **surname**, **forename**, **name\_post** для запроса. Сгруппировать данные по коду проекта.

Конструктор запросов

Столбец	Псевдо...	Таблица	Вывод	Тип сортиро...	Порядок сор...	Группировка	Фильтр
id		Workproje...	<input checked="" type="checkbox"/>			Group By	
id_project		Workproje...	<input checked="" type="checkbox"/>			Group By	
name_project		Project	<input checked="" type="checkbox"/>			Group By	
id_worker		Workproje...	<input checked="" type="checkbox"/>			Group By	
surname		Worker	<input checked="" type="checkbox"/>			Group By	
forename		Worker	<input checked="" type="checkbox"/>			Group By	
			<input type="checkbox"/>				
			<input type="checkbox"/>				

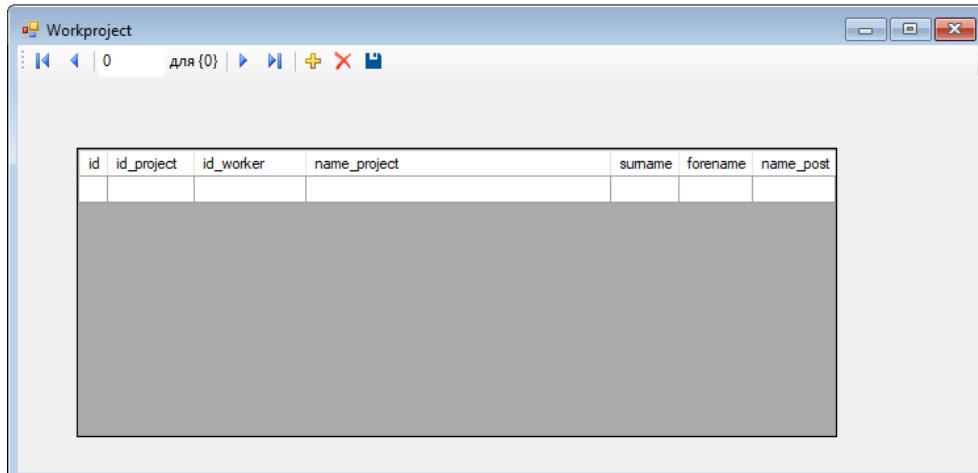
```
SELECT Workproject.id, Workproject.id_project, Project.name_project, Workproject.id_worker, Worker.surname, Worker.fc
FROM Workproject INNER JOIN
Project ON Workproject.id_project = Project.id_project INNER JOIN
Worker ON Workproject.id_worker = Worker.id_worker
```

id	id_project	name_project	id_worker	surname	forename
1	1	Сайт для ООО ...	1	Иванов	Сергей
2	1	Сайт для ООО ...	2	Петров	Андрей

1    для 13 | ► | ▶ | ⟲ | ⟳ | ⟲ | ⟳ | Ячейка доступна только для чтения.

Выполнить запрос    OK    Отмена

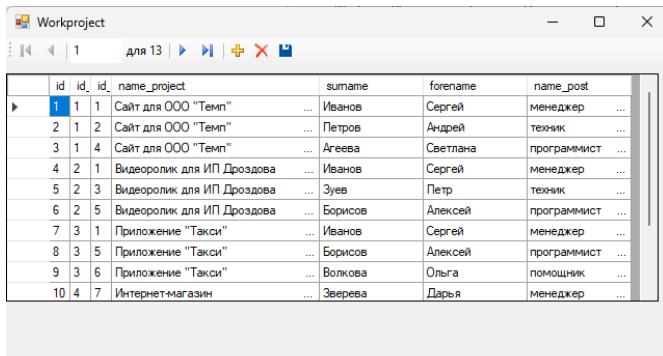
**6.5** Добавьте из источника данных таблицу **Workproject**, сделайте недоступными для редактирования поля из внешних таблиц.



**6.6** Обеспечить переход с главной формы на форму **Workproject** при нажатии на кнопку **УПРАВЛЕНИЕ ПРОЕКТАМИ**.

```
private void button4_Click(object sender, EventArgs e)
{
    Workproject form = new Workproject();
    form.Show();
}
```

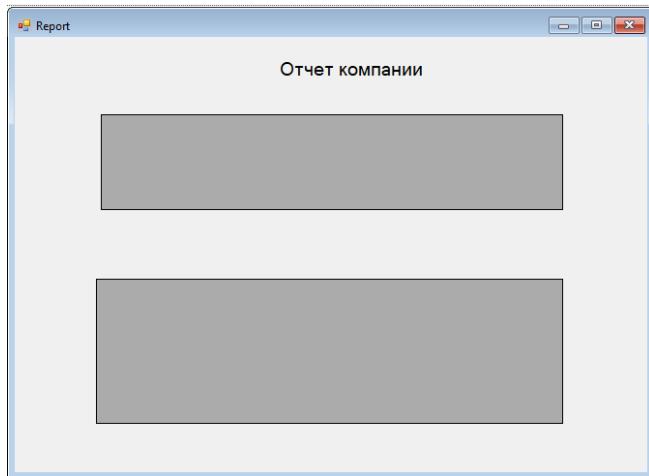
**6.7** Протестировать приложение с переходом на форму управления проектами.



**Задание 7.** Создание отчета компании, который должен содержать сколько человек в компании, чему равен фонд заработной платы, сколько человек трудятся над каждым проектом.

**7.1** Создать новую форму под названием **Report**.

**7.2** Разместить на форме заголовок и два элемента **dataGridView**



**7.3** При загрузки формы создадим соединение с базой данных и выполним два запроса:

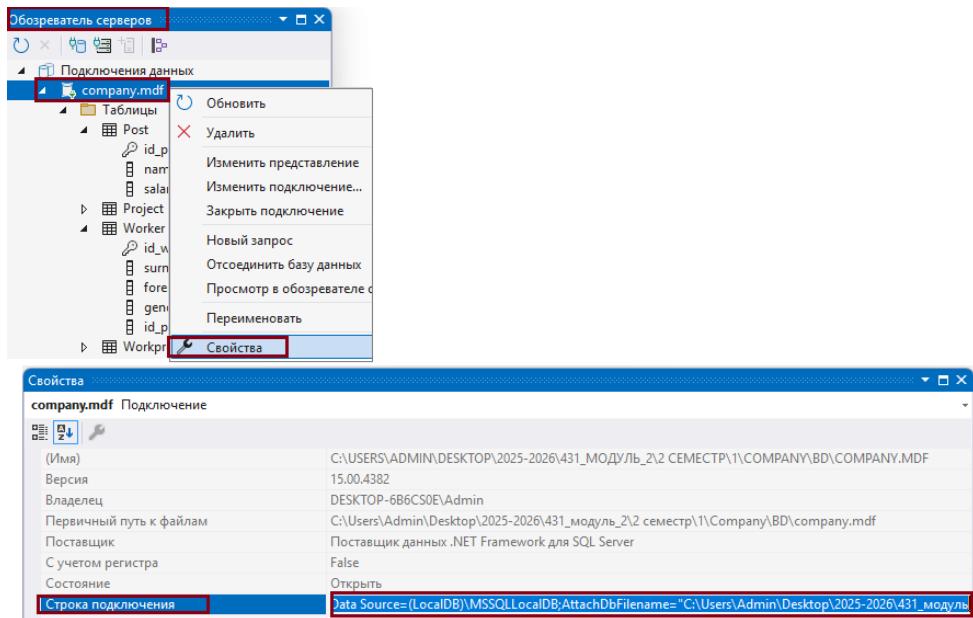
**Запрос 1** – используя таблицы **Post** и **Worker**, найти количество сотрудников и сумму всех окладов

**Запрос 2** – используя таблицы **Project**, **Worker** и **Workproject**, сгруппировать данные по коду проекта и найти количество сотрудников, участвующих в проекте.

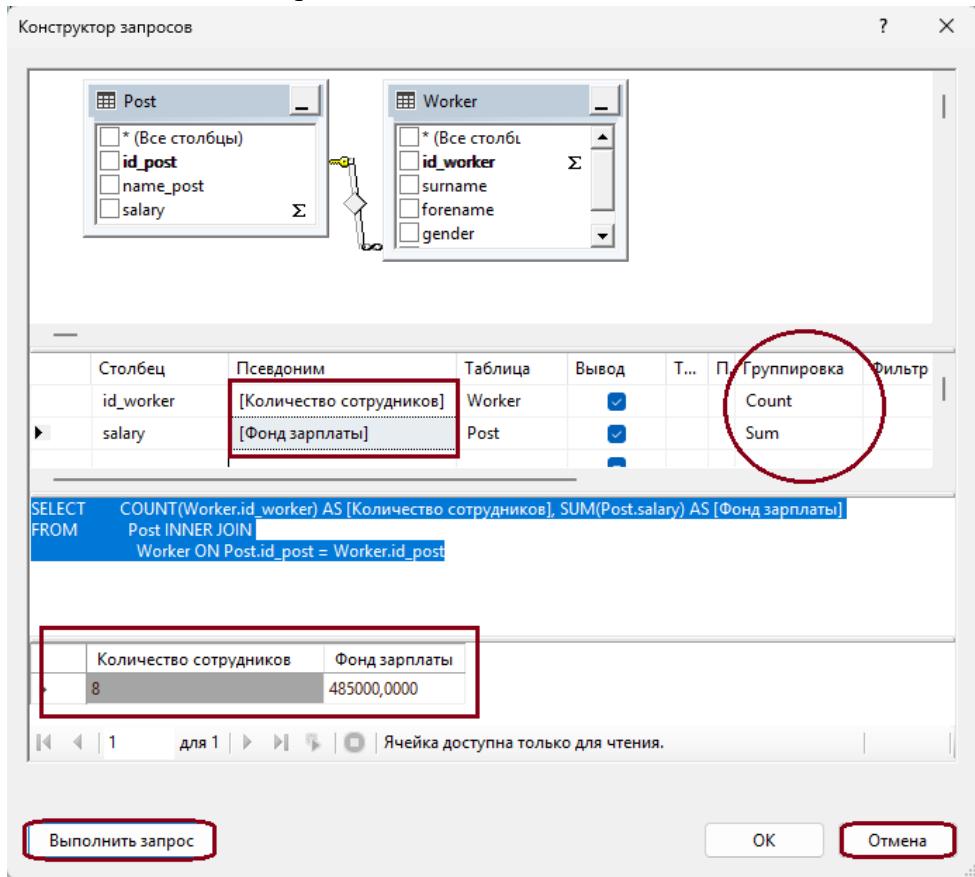
```
//процедура загрузки формы
private void Report_Load(object sender, EventArgs e)
{
String str_con = ""; // строка для соединения
string sqlExpression_1 = ""; //строка для первого запроса
string sqlExpression_2 = ""; //строка для второго запроса
//используем данное соединение и открываем его
    using (SqlConnection conn = new SqlConnection(str_con))
    {
conn.Open(); //открыть соединение
//создаем команду для первого запроса и выполняем ее
    SqlCommand command1 = new SqlCommand(sqlExpression_1, conn);
// возвращаем количество строк запроса
    command1.ExecuteNonQuery();
// Создаем объект DataAdapter
    SqlDataAdapter adapter1 = new SqlDataAdapter(sqlExpression_1, conn);
// Создаем объект Dataset
    DataSet ds1 = new DataSet();
// Заполняем Dataset
    adapter1.Fill(ds1);
// Отображаем данные
    dataGridView1.DataSource = ds1.Tables[0];
//проделываем тоже самое для второго запроса
    SqlCommand command2= new SqlCommand(sqlExpression_2, conn);
    command2.ExecuteNonQuery();
// Создаем объект DataAdapter
    SqlDataAdapter adapter2 = new SqlDataAdapter(sqlExpression_2, conn);
// Создаем объект Dataset
    DataSet ds2 = new DataSet();
// Заполняем Dataset
    adapter2.Fill(ds2);
// Отображаем данные
    dataGridView2.DataSource = ds2.Tables[0];
//закрываем соединение
    conn.Close();
    }
}
```

7.6 В обозревателе серверов перейти к свойствам базы данных company.mdf и скопировать целиком строку для подключения. Подставьте эту строку в код

```
String str_con = "Data Source= (LocalDB)\MSSQLLocalDB; AttachDbFilename..";
```



**7.7** Для того, чтобы автоматически сформировать строку для первого запроса (узнать количество сотрудников и фонд их зарплаты), откройте команду Select адаптера таблицы Post. Через конструктор добавьте таблицу Worker. Сделайте группировку. Запустите запрос в конструкторе, скопируйте текст запроса, а затем сделайте отмену, чтобы не изменить в адаптере таблицы базы данных.



Столбец	Псевдоним	Таблица	Выход	Группировка	Фильтр
id_worker	[Количество сотрудников]	Worker	<input checked="" type="checkbox"/>	Count	
salary	[Фонд зарплаты]	Post	<input checked="" type="checkbox"/>	Sum	

```
SELECT COUNT(Worker.id_worker) AS [Количество сотрудников], SUM(Post.salary) AS [Фонд зарплаты]
FROM Post INNER JOIN Worker
ON Post.id_post = Worker.id_post
```

Количество сотрудников	Фонд зарплаты
8	485000,0000

Выполнить запрос

OK

Отмена

Вставьте строку запроса в код

```
string sqlExpression_1 = "SELECT ... "
```

**Замечание.** Уберите из запроса знаки перехода на новую строку и определите заголовки столбцов

**7.8** Аналогично, используя команду Select адаптера таблицы **Workproject**, получить строку для второго запроса, т.е. узнать, сколько человек участвуют в каждом проекте.

Конструктор запросов

Столбец Псевдоним Таблица Вывод Т... П... Группировка Фильтр

Столбец	Псевдоним	Таблица	Вывод	Т...	П...	Группировка	Фильтр
<input checked="" type="checkbox"/> id_project	[Название проекта]	Workprojec...	<input type="checkbox"/>			<input checked="" type="checkbox"/> Group By	
<input checked="" type="checkbox"/> name_project		Project	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/> Group By	
<input checked="" type="checkbox"/> id_worker	[Количество сотрудников]	Worker	<input checked="" type="checkbox"/>			<input checked="" type="checkbox"/> Count	

```
SELECT Project.name_project AS [Название проекта], COUNT(Worker.id_worker) AS [Количество сотрудников]
FROM Workproject INNER JOIN
Project ON Workproject.id_project = Project.id_project INNER JOIN
Worker ON Workproject.id_worker = Worker.id_worker
GROUP BY Workproject.id_project, Project.name_project
```

	Название проекта	Количество сотрудников
▶	Сайт для ООО "Темп"	3
	Видеоролик для ИП Дрозд...	3
	Приложение "Такси"	3
	Интернет-магазин	5
	Приложение "Агентство н...	4

◀ ▶ 1      для 5 | ▶ | ⏪ | ⏩ | ⏴ | ⏵ | Ячейка доступна только для чтения.

Выполнить запрос      OK      Отмена

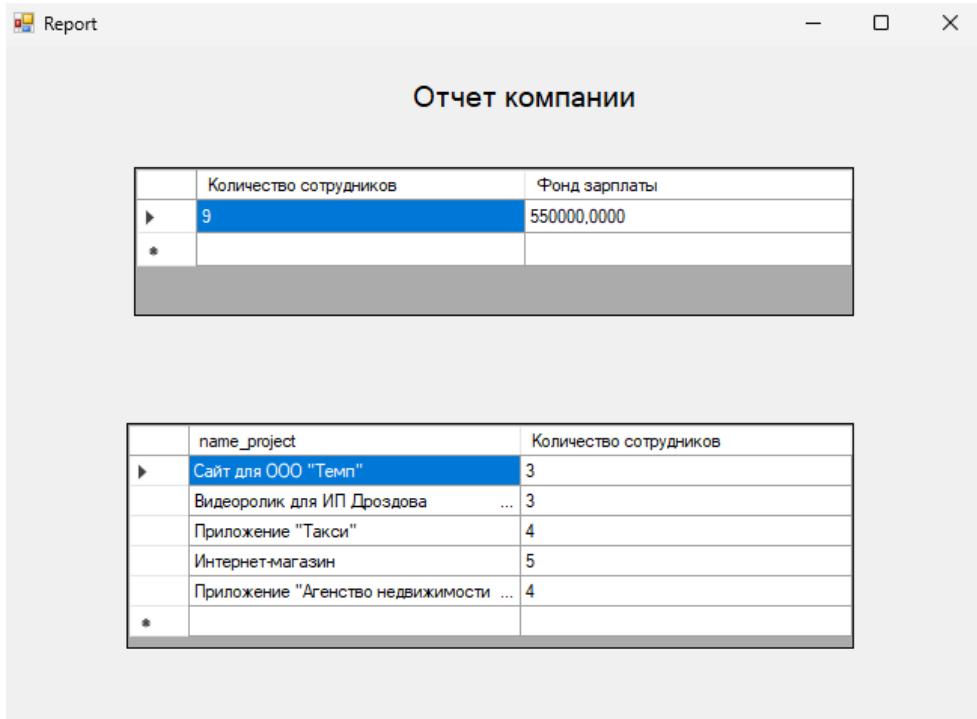
Вставьте строку запроса в код

```
string sqlExpression_2 = "SELECT ... "
```

**Замечание.** Уберите из запроса знаки перехода на новую строку и определите заголовки столбцов

**7.9** Обеспечить переход на форму отчета при нажатии на кнопку ОТЧЕТ КОМПАНИИ.

**7.10** Протестировать получение отчета.



Report

Отчет компании

	Количество сотрудников	Фонд зарплаты
▶	9	550000,0000
*		

	name_project	Количество сотрудников
▶	Сайт для ООО "Темп"	3
	Видеоролик для ИП Дроздова	...
	Приложение "Такси"	4
	Интернет-магазин	5
*	Приложение "Агентство недвижимости" ...	4

**Задание 8.** Протестировать приложение целиком

**Задание 9 (выполняется самостоятельно).** Привести данные к третьей нормальной форме. Создать приложение для бронирования номеров и проживания в гостинице. Приложение должно вести учет всех данных и выдавать отчет, который содержит следующую информацию:

*Сколько номеров имеют холодильник, сколько имеют телевизор, сколько имеют кондиционер и сколько Wi-Fi*

*На каждый день показать количество занятых, забронированных и свободных номеров.*

*Чему равен доход гостиницы.*

*Сколько неоплаченных номеров и на какую сумму.*

Заказчик предоставил следующую документацию для создания базы данных.

### **Список гостиничных номеров**

№1 - одноместный номер, есть удобства (холодильник, телевизор), сутки проживания – 3500 руб.

№2 - одноместный номер, есть удобства (холодильник, телевизор), сутки проживания – 3500 руб.

№3 – двухместный номер, есть удобства (кондиционер, холодильник, телевизор), сутки проживания – 5000 руб.

№4 - двухместный номер, есть удобства (холодильник, телевизор), сутки проживания – 5000 руб.

№5 – номер люкс, есть удобства (кондиционер, холодильник, телевизор, Wi-Fi), сутки проживания – 7000 руб.

### **Журнал регистрации фиксирует информацию на каждый день для каждого номера.**

01.01.2026 – номер №1, клиент - Иванов, адрес проживания Москва, мужчина, проживание, оплачен

01.01.2026 – номер №2, клиент – Сидоров, адрес проживания Псков, мужчина, проживание, оплачен

01.01.2026 – номер №3, Клиент – Петрова, адрес проживания Москва, женщина, оплачен

01.01.2026 – номер №3, Клиент – Петров, адрес проживания Москва, мужчина, оплачен

01.01.2026 – номер №4, Клиент – Гусева, адрес проживания Новгород, женщина, оплачен

01.01.2026 – номер №5, свободен

02.01.2026 – номер №1, клиент - Иванов, адрес проживания Москва, мужчина, проживание, оплачен

02.01.2026 – номер №2, свободен

02.01.2026 – номер №3, Клиент – Петрова, адрес проживания Москва, женщина, оплачен

02.01.2026 – номер №3, Клиент – Петров, адрес проживания Москва, мужчина, оплачен

02.01.2026 – номер №4, свободен

02.01.2026 – номер №5, свободен

03.01.2026 - номер №1, клиент - Иванов, адрес проживания Москва, мужчина, проживание, оплачен

03.01.2026 – номер №2, клиент – Ухов, адрес проживания Псков, мужчина, проживание, не оплачен

03.01.2026 – номер №3, свободен

03.01.2026 - номер №4, забронирован

04.01.2026 - номер №4, забронирован

05.01.2026 - номер №4, забронирован

03.01.2026 – номер №5, клиент – Николаева, адрес проживания Новгород, женщина, проживание, оплачен