**МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ**

**ИМЕНИ И.Т. ТРУБИЛИНА»**

**Факультет прикладной информатики**

**Кафедра системного анализа и обработки информации**

**КУРСОВАЯ РАБОТА**

по дисциплине «Алгоритмизация и программирование»

на тему «Программирование приложения ведения БД «Полицейский участок»

Направление подготовки 09.03.03 «Прикладная информатика»

Направленность «Менеджмент проектов в области информационных технологий, создание и поддержка информационных систем»

Выполнила:

Хоружая Алина Максимовна

группа ПИ1901

Руководитель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Орлянская Н.П.

**Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Орлянская Н.П.

**Краснодар 2020**

**РЕФЕРАТ**

Пояснительная записка выполнена на 45 страницах, использовано 25 иллюстраций и 6 литературных источников.

Ключевые слова: СОЗДАНИЕ, ФОРМЫ, БАЗА ДАННЫХ, СВЯЗЬ, УДАЛЕНИЕ, CRUD, СУБД.

Целью работы является разработка приложения «Ведение БД Полицейский участок» в среде MS Visual Studio 2019 на языке программирования C#.

Объект исследования – базовые CRUD операции.

Предмет исследования – средства языков программирования для создания базы данных.

Разработанная программа позволяет вести базу данных для работы в полицейском участке.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc41323676)

[1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc41323677)

[1.1 Теоретический блок 5](#_Toc41323678)

[1.2 Выбор СУБД и средств для разработки 6](#_Toc41323679)

[2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ 8](#_Toc41323680)

[2.1 Проектирование базы данных 8](#_Toc41323681)

[2.2 Разработка базы данных 9](#_Toc41323682)

[3. РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ 10](#_Toc41323683)

[3.1 Описание интерфейса приложения 10](#_Toc41323684)

[3.1.1 Форма «Главная» 11](#_Toc41323685)

[3.1.2 Форма «Добавление нового сотрудника» 12](#_Toc41323686)

[3.1.3 Форма «Удаление сотрудника» 13](#_Toc41323687)

[3.1.4 Форма «Добавление задержанного АК» 13](#_Toc41323688)

[3.1.5 Форма «Добавление задержанного УК» 15](#_Toc41323689)

[3.1.6 Форма «Сотрудники» 16](#_Toc41323690)

[3.1.7 Форма «Задержанные на основе АК РФ» 17](#_Toc41323691)

[3.1.8 Форма «Задержанные на основе УК РФ» 18](#_Toc41323692)

[3.2 Описание основных процедур и функций 18](#_Toc41323693)

[3.2.1 Функции (методы) классов форм 19](#_Toc41323694)

[3.3 Работа с MySQL 23](#_Toc41323695)

[3.4 Результат 24](#_Toc41323696)

[4. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 25](#_Toc41323697)

[ЗАКЛЮЧЕНИЕ 31](#_Toc41323698)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 32](#_Toc41323699)

[ПРИЛОЖЕНИЕ 33](#_Toc41323700)

# **ВВЕДЕНИЕ**

Данные — поддающееся многократной интерпретации представление информации в формализованном виде, пригодном для передачи, связи, или обработки. В современном мире необходимо уметь разделять и группировать существующие данные. Какой бы формат они не имели, применяемые базовые операции останутся неизменными создание, чтение, редактирование или модификация, удаление.

Для комфортного использования базы данных, недостаточно просто «создать» ее. Необходимо разработать так называемый «орган», состоящий из алгоритмов и функций, который будет управлять и направлять данные. Визуализация, в свою очередь, поспособствует комфортному использованию.

Цель работы – разработка приложения «Ведение БД Полицейский участок» в среде Microsoft Visual Studio 2019 на языке программирования C#.

Задачи работы:

* Изучить предметную область, для которой создается продукт;
* Разработать структуру БД;
* Создать продукт, готовый к использованию в своей профессиональной среде.

# **ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**

На практике существует специальный отдел ЗИЦ (Зональный информационный отдел), который состоит из 12 отделов. Каждый отдел контролирует информацию вводимую в назначенную базу данных.

Данная курсовая работа направлена на автоматизацию работы Полицейского участка. Приложение позволяет быстро внести необходимую информацию, которая будет использоваться фундаментом для дальнейшей работы.

На вход мы получаем информацию о сотрудниках:

* Номер значка
* ФИО
* Звание

И информацию о задержанных:

* ФИО
* Дата рождения
* Гражданство
* Наличие ранних привлечений к ответственности
* Детали дела (номер статьи и дополнительные уточнения)

Этой информации достаточно на начальных этапах работы.

В таком виде деятельности необходима простота и эргономичность, которые будут не только соответствовать работе, но и поддерживать ее.

## **1.1 Теоретический блок**

База данных — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины.

Многие специалисты указывают на распространённую ошибку, состоящую в некорректном использовании термина «база данных» вместо термина «система управления базами данных», и указывают на необходимость различения этих понятий [1]. В паре слов коснемся существенных различий:

1. БД хранится и обрабатывается в вычислительной системе.
2. Данные в БД логически систематизированы.
3. БД включает метаданные.

Понятие CRUD введено в 1983г Дж. Мартином и является стандартной классификацией по манипуляции данными; аббревиатура, которая обозначает 4 основных функции при работе с БД: create (создание), read (чтение), update (модификация), delete (удаление). Без этих функций невозможна успешная работа и реализация.

СУБД - совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием БД.

## **1.2 Выбор СУБД и средств для разработки**

В качестве СУБД была выбрана MySQL 8.0, которая является одной из самых популярных СУБД на сегодняшний день является. Эта серверная система проста в освоении. К тому же, в интернете без каких-либо усилий можно найти руководства по освоению системы, плагины и расширения, которые помогут освоиться в этой среде.

Следует отметить следующие преимущества:

1. Простота в использовании.
2. Обширный функционал.
3. Безопасность.
4. Масштабируемость. Можно использовать для работы с малыми и большими объемами данных.
5. Скорость.

Приложение ведения БД разработано в Visual Studio 2019 (самая актуальная версия среды).

Microsoft Visual Studio имеет поддержку технологии Windows Forms, которая значительно упрощает типичные задачи приложения (чтение и запись в файловую систему). При использовании среды VS можно создавать WF клиентские приложения, которые будут отображать сведения, запрашивать данные от пользователя. Среда включает в себя редактор исходного кода с поддержкой технологии IntelliSense (автодополнение) и возможностью простейшего рефакторинга (процесс изменения внутренней структуры программы) кода. Встроенный отладчик на 2 уровнях: исходного кода и машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

Приложение написано на объектно-ориентированном языке программирования - C#, который был разработан в 1998-2001 годах, как язык разработки приложений для платформы Microsoft .NET Framework. В дальнейшем был стандартизирован. По синтаксису он близок к языку С++, который лежал в основе изучения курса АиООП в течение всего первого курса. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы. Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественное наследование интерфейсов).

# **ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА БАЗЫ ДАННЫХ**

## **2.1 Проектирование базы данных**

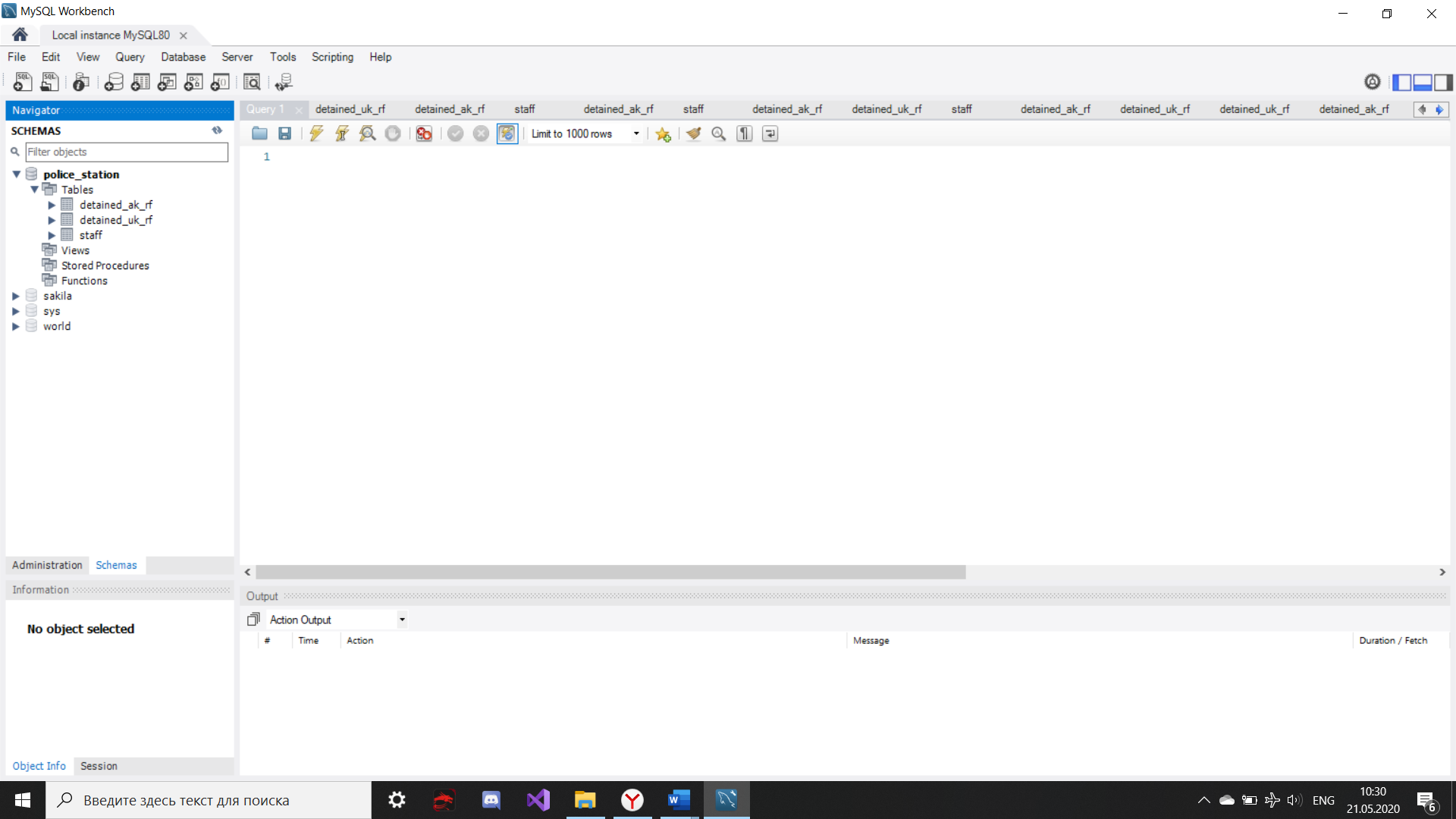
В системе MySQL сформирована база данных – police\_station, содержащая в себе 3 основные таблицы с данными: detained\_ak\_rf (задержанные на основе АК РФ), detained\_ak\_rf (задержанные на основе УК РФ), staff (сотрудники), (см. рис. 2.1).

Рисунок 2.1 - База данных и ее составляющие

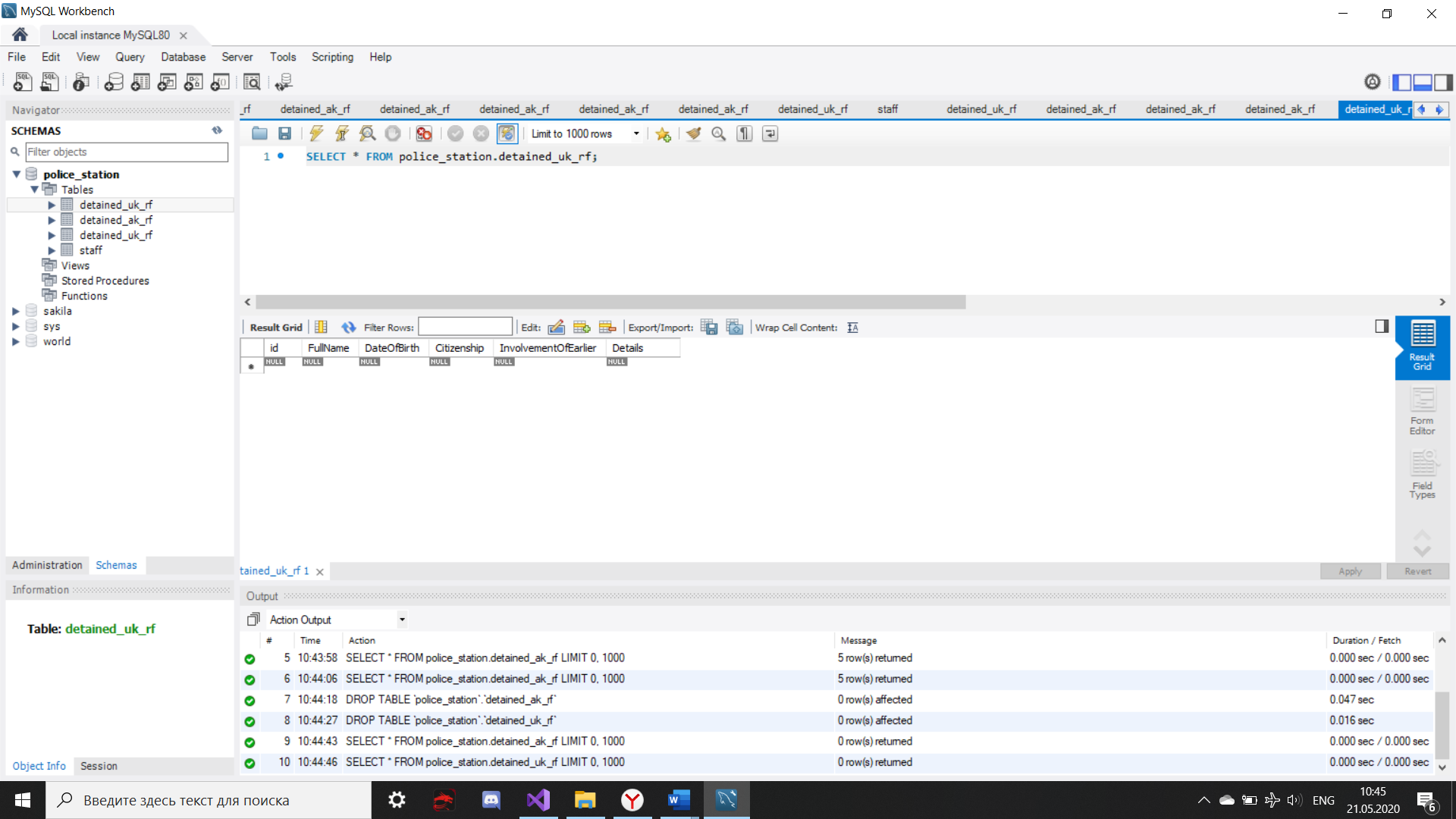
Пройдемся по содержимому таблиц и полям, которые необходимо заполнить. База с задержанными на основе АК (УК) РФ содержит 6 столбцов: id (порядковый номер, присваивается автоматически) – int AI PK, FullName (ФИО) - text, DateOfBirth (дата рождения) - data, Citizenship (гражданство) - text, InvolvementOfEarlier (привлечение ранее 1 = да, 0 = нет) - tinyint, Details (детали дела) - text. (см. рис. 2.2).

Рисунок 2.2 - Столбцы таблиц задержанных (АК\УК)

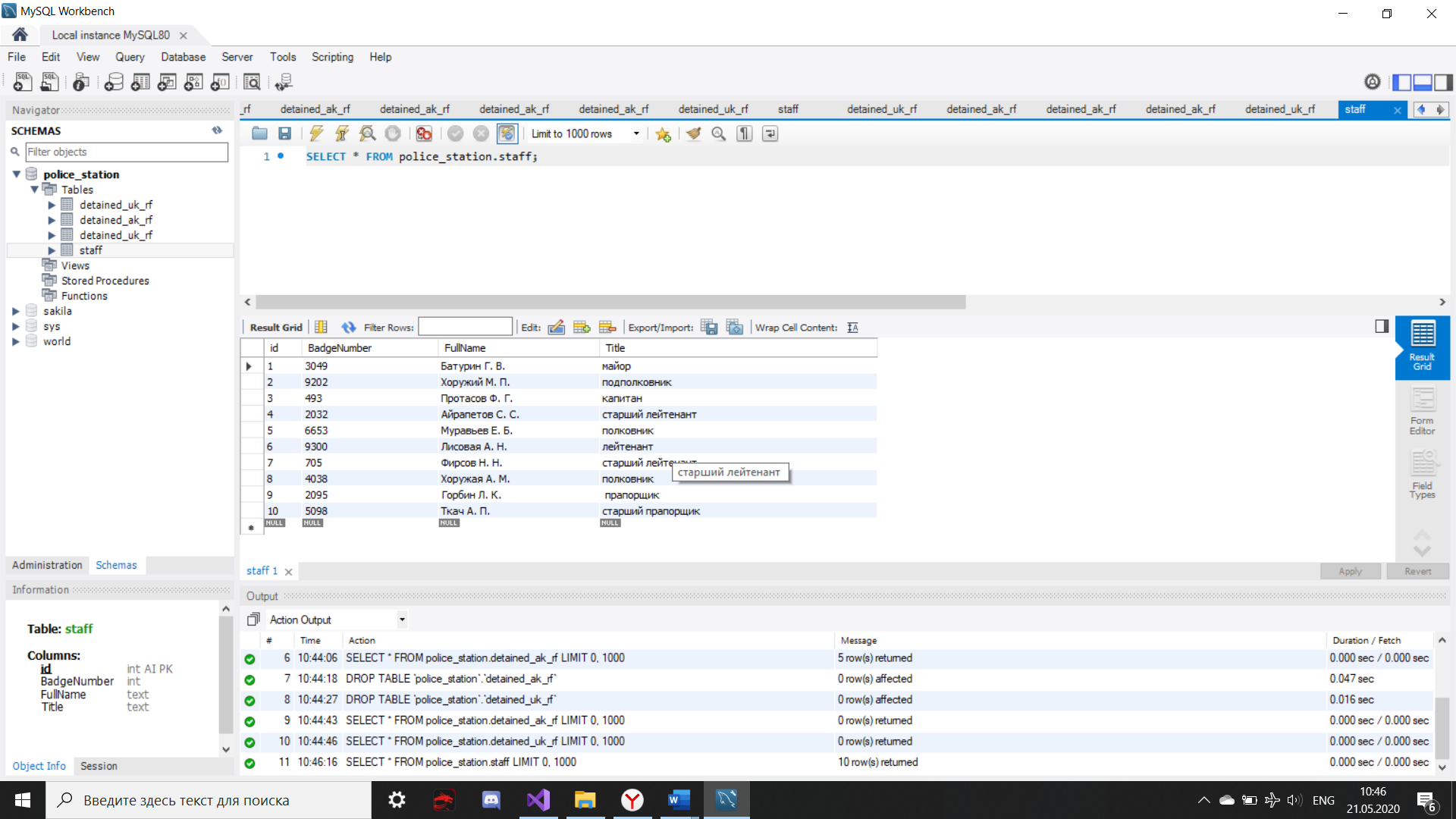
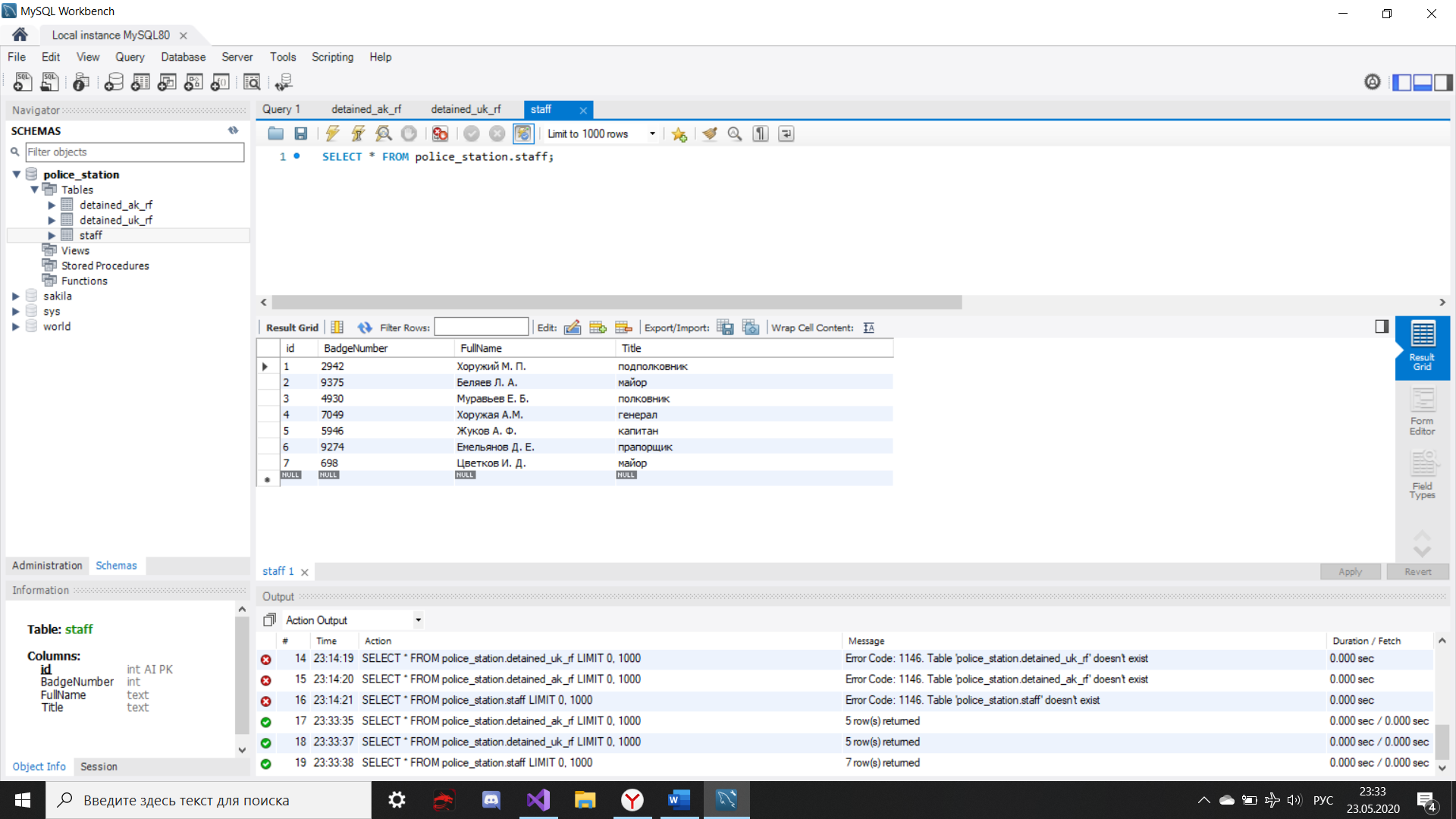
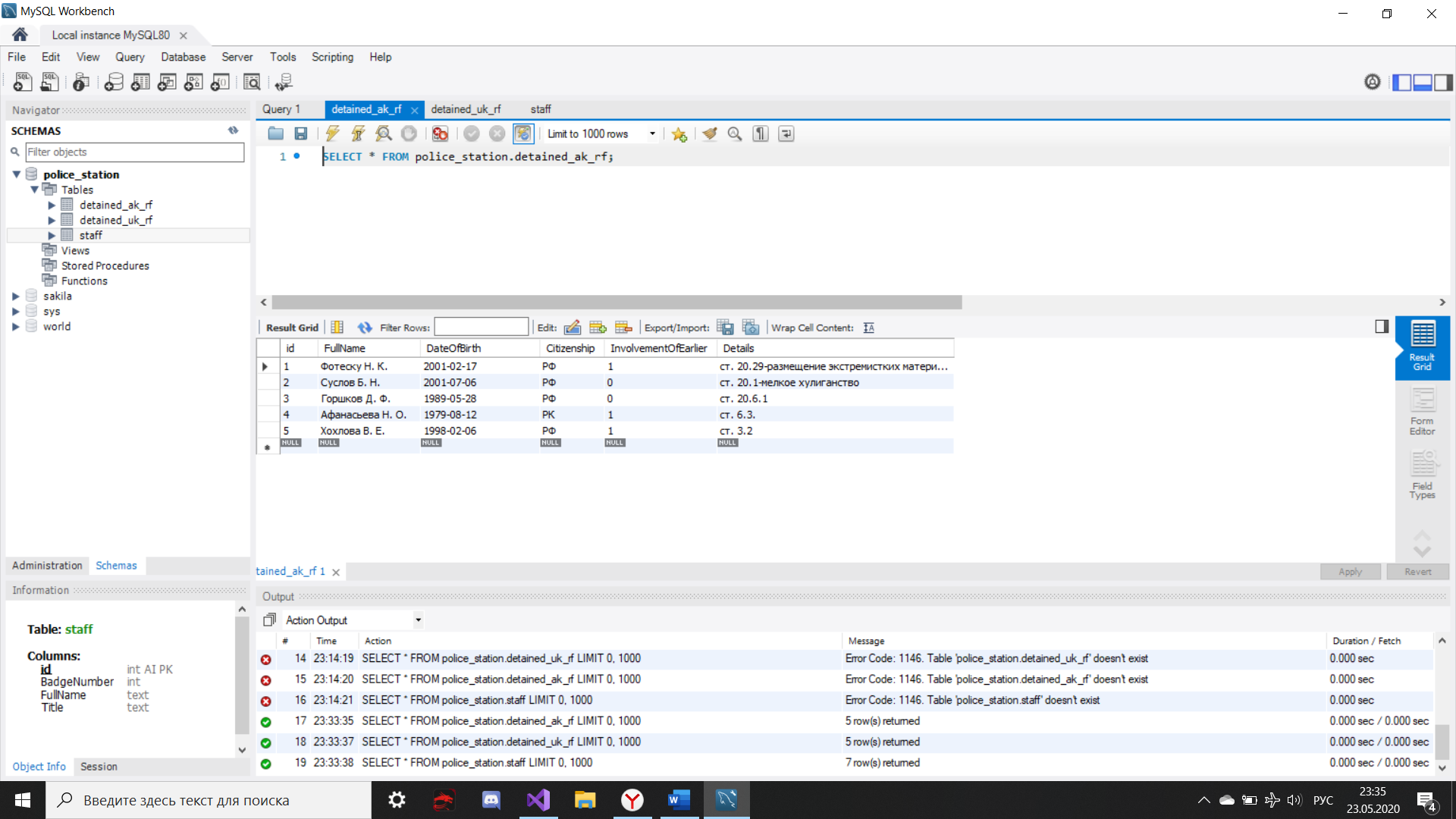
База с данными о сотрудниках содержит 4 стобца: id (порядковый номер, присваивается автоматически) – int AI PK, BadgeNumber (номер значка) - int, FullName (ФИО), Title (звание) – text. (см. рис. 2.3)

Рисунок 2.3 - Столбцы таблицы с данными о сотрудниках

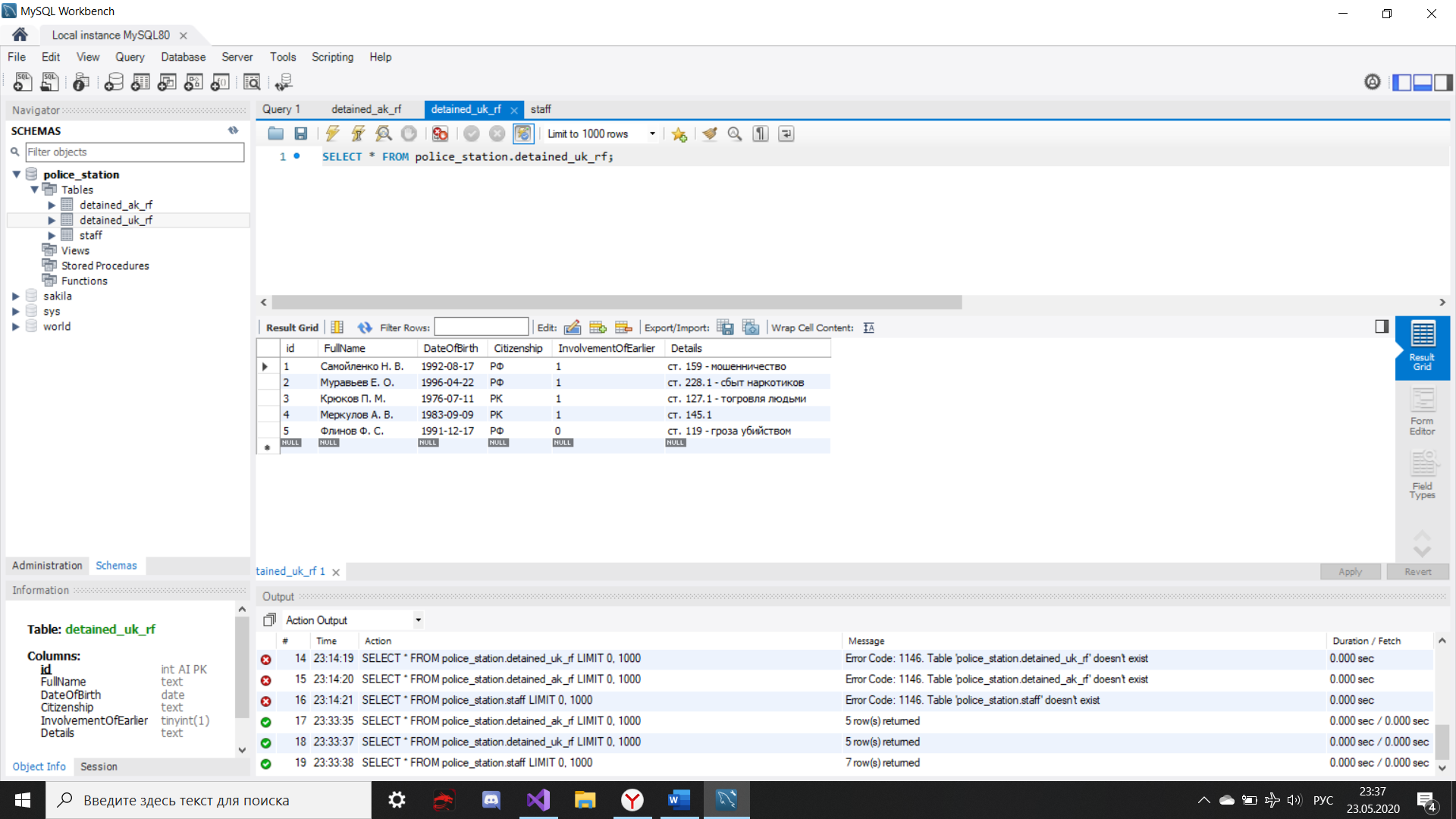
## **2.2 Разработка базы данных**

В работе были использованы следующие данные:

1. Таблица с данными о сотрудниках (см. рис. 2.4):

Рисунок 2.4 – сотрудники

1. Таблица с данными о задержанных на основе АК РФ (см. рис. 2.5):

Рисунок 2.5 – задержанные по АК РФ

1. Таблица с данными о задержанных на основе УК РФ (см. рис. 2.6):

Рисунок 2.6 – задержанные по УК РФ

# **РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ**

Приложение по управлению базой данных полицейского участка имеет следующий алгоритм решения:

1. Разработка графического макета приложения. Состоит из следующих элементов:

* Форма: сотрудники, задержанные на основе АК РФ, задержанные на основе УК РФ.
* Кнопка добавления сотрудника
* Кнопка удаления сотрудника
* Кнопка добавления задержанного на основе АК РФ
* Кнопка добавления задержанного на основе УК РФ
* Кнопка выхода
* (Кнопка отчета)

1. Создание функций для работы с БД: Загрузка информации из БД.
2. Программирование кнопок, работающих с таблицей( добавление, удаление).
3. Создание диалоговых окон для добавления (сотрудников\заключенных), удаления сотрудников а также выхода из приложения.
4. Создание полей для внесения соответствующих данных о сотрудниках и заключенных.

## **3.1 Описание интерфейса приложения**

Интерфейс приложения включает в себя 5 форм. Каждая из них способствует удобному а главное понятному использованию приложения. Подобранный минималистичный дизайн обеспечивает сконцентрированную работу.

### **3.1.1 Форма «Главная»**

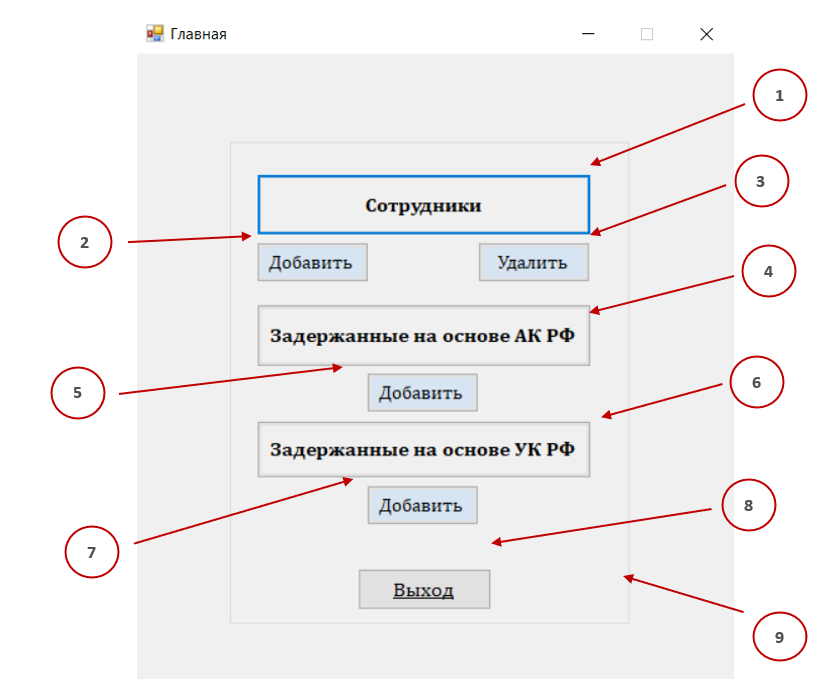
Имя компонента: Form1 (см. рис 3.1):

Рисунок 3.1. - Form1 и ее компоненты

1. Кнопка перехода к таблице с сотрудниками. Имя компонента – button\_staff. Класс компонента – Button.
2. Кнопка перехода к новой форме для добавления сотрудника в базу. Имя компонента – button\_add\_staff. Класс компонента – Button.
3. Кнопка перехода к новой форме для удаления сотрудника из базы. Имя компонента – button\_delete\_ staff. Класс компонента – Button.
4. Кнопка перехода к таблице с задержанными на основе АК РФ. Имя компонента – button\_AK. Класс компонента – Button.
5. Кнопка перехода к новой форме для добавления задержанного в базу. Имя компонента – button\_add\_AK. Класс компонента – Button.
6. Кнопка перехода к таблице с задержанными на основе УК РФ. Имя компонента – button\_UK. Класс компонента – Button.
7. Кнопка перехода к новой форме для добавления задержанного в базу. Имя компонента – button\_add\_UK. Класс компонента – Button.
8. Кнопка выхода из приложения с диалоговым окном. Имя компонента – button\_exit. Класс компонента – Button.
9. Группировка основных компонентов. Имя компонента – groupBox1. Класс компонента – GroupBox.

### **Форма «Добавление нового сотрудника»**

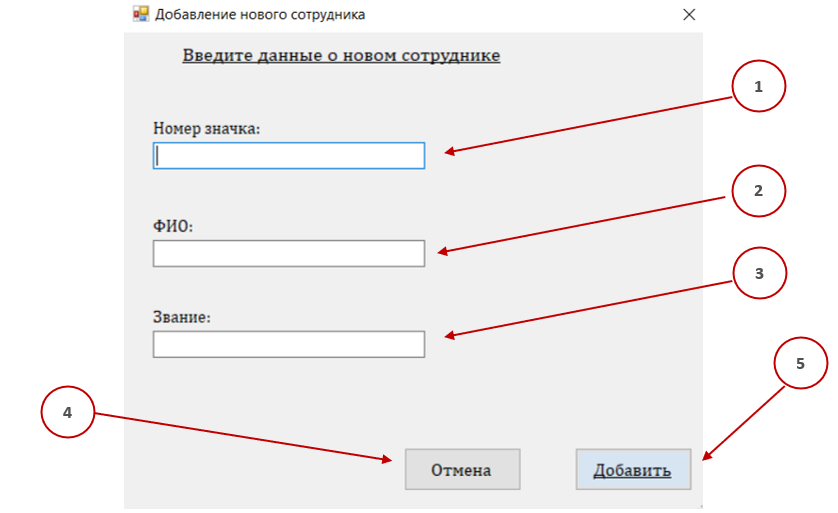
Имя компонента: AddStaff (см. рис. 3.2):

Рисунок 3.2 - AddStaff и ее компоненты

1. Поле ввода для номера значка сотрудника. Имя компонента - badge\_number\_box. Класс компонента – TextBox.
2. Поле ввода для фамилии, имени и отчества сотрудника. Имя компонента - fio\_box. Класс компонента – TextBox.
3. Поле ввода для звания сотрудника Имя компонента - title\_box. Класс компонента – TextBox.
4. Кнопка отмены и выхода из окна добавления сотрудника. Имя компонента – button\_Cancel. Класс компонента – Button.
5. Кнопка добавления сотрудника в базу. Имя компонента – button\_AddStaff. Класс компонента – Button.

### **Форма «Удаление сотрудника»**

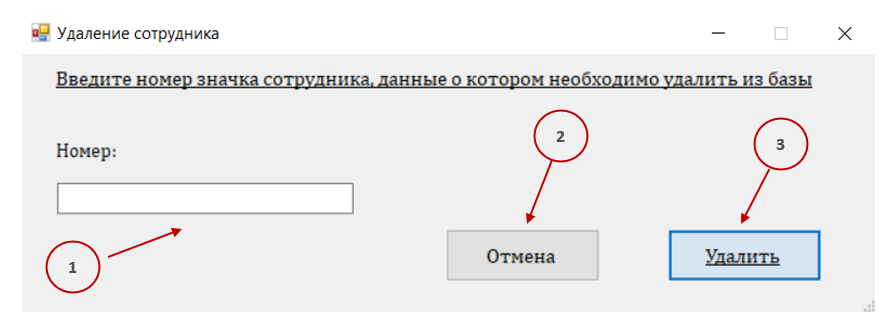
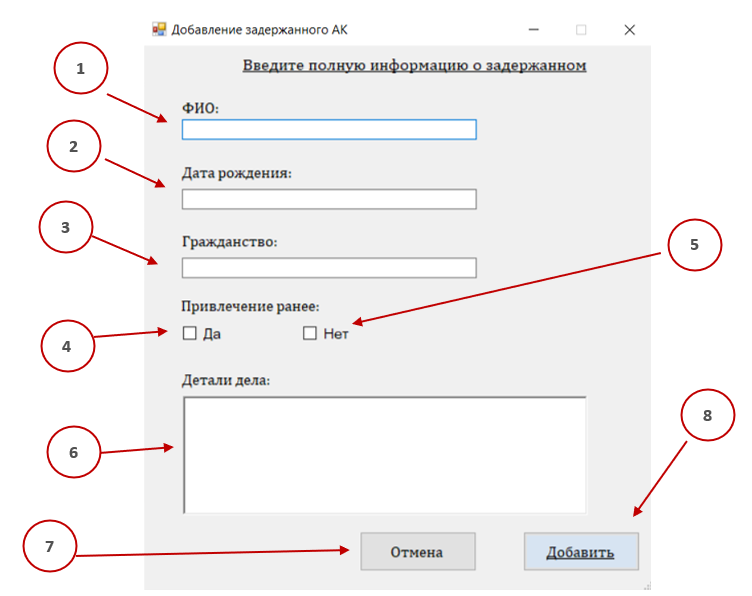
Имя компонента: DelteStaff (см. рис. 3.3):

Рисунок 3.3 - DelteStaff и ее компоненты

1. Поле ввода для номера значка сотрудника, по которому будет производиться удаление из базы. Имя компонента - number\_box. Класс компонента – TextBox.
2. Кнопка отмены и выхода из окна добавления сотрудника. Имя компонента – button\_Cancel. Класс компонента – Button.
3. Кнопка удаления сотрудника из базы. Имя компонента – button\_DeleteStaff. Класс компонента – Button.

### **Форма «Добавление задержанного АК»**

Имя компонента: AddAK (см. рис. 3.4)

Рисунок 3.4 - AddAK и ее компоненты

1. Поле ввода для фамилии, имени и отчества задержанного. Имя компонента – fio\_box. Класс компонента - TextBox.
2. Поле ввода для даты рождения задержанного. Имя компонента - date\_of\_brith\_box. Класс компонента - TextBox.
3. Поле ввода для гражданства задержанного. Имя компонента - zenship\_box. Класс компонента - TextBox.
4. Флаг на наличие имеющихся привлечений к ответственности. Имя компонента - checkBox\_Yes. Класс компонента – CheckBox.
5. Флаг на отсутствие имеющихся привлечений к ответственности. Имя компонента - checkBox\_No. Класс компонента – CheckBox.
6. Поле для ввода деталей дела (например, номер статьи и т.д.) Имя компонента - details\_case\_box. Класс компонента – RichTextBox.
7. Кнопка отмены и выхода из окна добавления сотрудника. Имя компонента – button\_Cancel. Класс компонента – Button.
8. Кнопка добавления задержанного в базу. Имя компонента - button\_Add. Класс компонента – Button.

### **Форма «Добавление задержанного УК»**

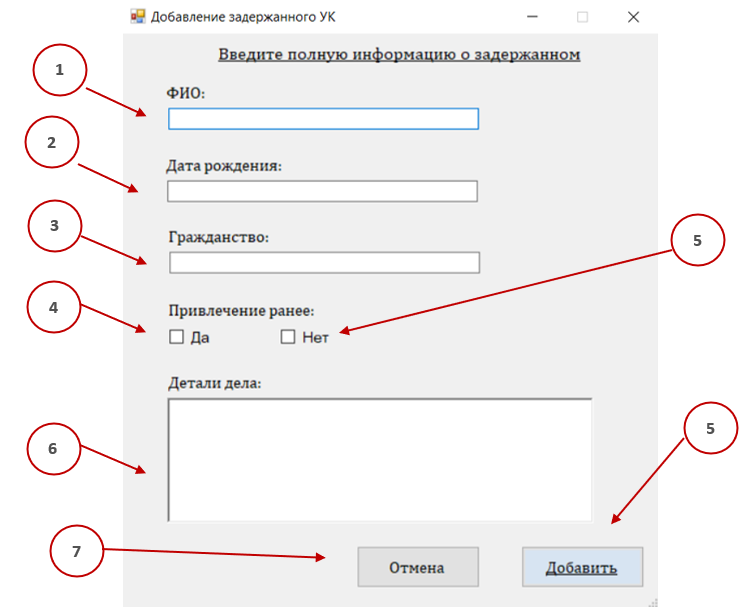
Имя компонента – AddUK (см. рис. 3.5)

Рисунок 3.5 – AddUK и ее компоненты

1. Поле ввода для фамилии, имени и отчества задержанного. Имя компонента – fio\_box. Класс компонента - TextBox.
2. Поле ввода для даты рождения задержанного. Имя компонента - date\_of\_brith\_box. Класс компонента - TextBox.
3. Поле ввода для гражданства задержанного. Имя компонента - zenship\_box. Класс компонента - TextBox.
4. Флаг на наличие имеющихся привлечений к ответственности. Имя компонента - checkBox\_Yes. Класс компонента – CheckBox.
5. Флаг на отсутствие имеющихся привлечений к ответственности. Имя компонента - checkBox\_No. Класс компонента – CheckBox.
6. Поле для ввода деталей дела (например, номер статьи и т.д.) Имя компонента - details\_case\_box. Класс компонента – RichTextBox.
7. Кнопка отмены и выхода из окна добавления сотрудника. Имя компонента – button\_Cancel. Класс компонента – Button.
8. Кнопка добавления задержанного в базу. Имя компонента - button\_Add. Класс компонента – Button.

### **Форма «Сотрудники»**

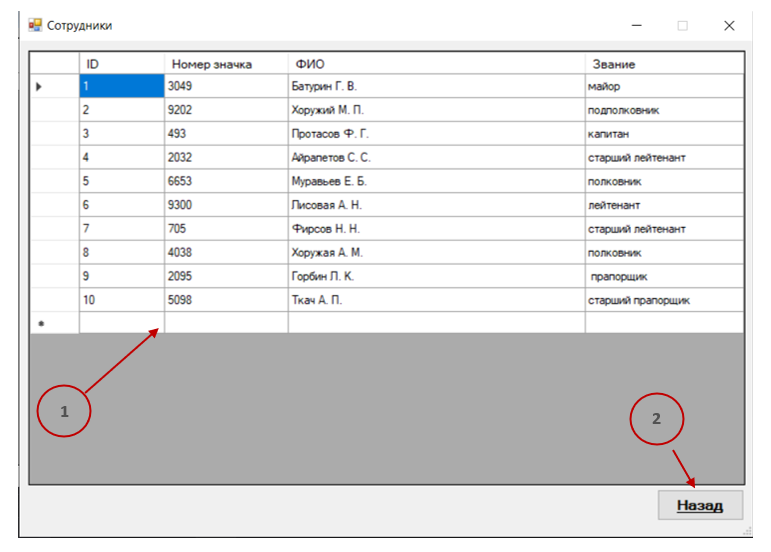
Имя компонента – Staff (см. рис. 3.6)

Рисунок 3.6 – Staff и ее компоненты

1. Таблица с данными о сотрудниках. Имя компонента - dataGridView\_Staff. Класс компонента – dataGridView
2. Кнопка отмены и выхода из окна с данными о сотрудниках. Имя компонента – button\_Cancel. Класс компонента – Button.

### **3.1.7 Форма «Задержанные на основе АК РФ»**

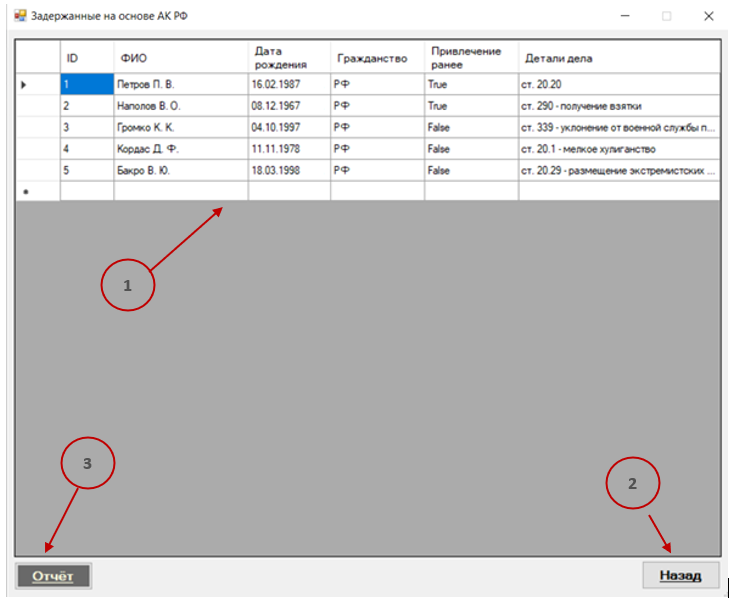
Имя компонента – Detained\_AK (см. рис. 3.7)

Рисунок 3.7 - Detained\_AK и ее компоненты

1. Таблица с данными задержанных на основе АК РФ. Имя компонента - dataGridView\_AK. Класс компонента – dataGridView
2. Кнопка отмены и выхода из окна с данными о задержанных. Имя компонента – button\_Cancel. Класс компонента – Button.
3. Кнопка экспорта данных в Excel. Имя компонента – button\_report. Класс компонента – Button.

### **Форма «Задержанные на основе УК РФ»**

Имя компонента – Detained\_UK (см. рис. 3.8)

Рисунок 3.8 - Detained\_UK и ее компоненты

1. Таблица с данными задержанных на основе УК РФ. Имя компонента - dataGridView\_UK. Класс компонента – dataGridView
2. Кнопка отмены и выхода из окна с данными о задержанных. Имя компонента – button\_Cancel. Класс компонента – Button.
3. Кнопка экспорта данных в Excel. Имя компонента – button\_report. Класс компонента – Button.

## **3.2 Описание основных процедур и функций**

Подключение к БД и установка соединения была вынесена в отдельный файл SQL\_connectors, по причине использования в каждой форме. Теперь достаточно прописать представленный блок кода, для осуществления подключения и дальнейшей успешной работы:

Функция подключения (в ином случае создания БД и всех ее составляющих): public void Get\_Connection\_First\_Time()

public partial class Detained\_AK : Form

{

SQL\_connector connector;

public Detained\_AK()

{

InitializeComponent();

connector = new SQL\_connector();

connector.Get\_Connection\_First\_Time();

}

### **3.2.1 Функции (методы) классов форм**

#### **3.2.1.1 Методы класса Form1. Форма «Главная»**

* Функция навигации: переход к форме с данными о сотрудниках: private void Button\_staff\_Click(object sender, EventArgs e)
* Функция навигации: переход к форме добавления нового сотрудника: private void Button\_add\_staff\_Click(object sender, EventArgs e)
* Функция навигации: переход к форме удаления сотрудника: private void Button\_delete\_staff\_Click(object sender, EventArgs e)
* Функция навигации: переход к форме с данными о задержанных на основе АК РФ: private void Button\_AK\_Click(object sender, EventArgs e)
* Функция навигации: переход к форме добавления задержанного по АК РФ: private void Button\_add\_AK\_Click(object sender, EventArgs e)
* Функция навигации: переход к форме с данными о задержанных на основе УК РФ: private void Button\_UK\_Click(object sender, EventArgs e)
* Функция навигации: переход к форме добавления задержанного по УК РФ: private void Button\_add\_UK\_Click(object sender, EventArgs e)
* Функция, завершающая работу приложения: private void Button\_exit\_Click(object sender, EventArgs e)

#### **3.2.1.2 Методы класса Staff. Форма «Сотрудники»**

* Функция, которая вызывает загрузку данных из БД в таблицу: private void LoadData().

Т. к. эта функция будет использована в 3 формах, опишем ее подробно.

Устанавливаем соединение с БД, указываем таблицу, из которой будем использовать данные. string query = "SELECT \* FROM staff ORDER BY id"

Создадим экземпляр класса, для чтения, который необходимо открыть и закрыть, после выполнения цикла. При помощи цикла while вызовем метод для чтения данных, который возвращает логическое значение True, пока есть данные. Для получения результата чтения используем объект data, а в квадратных скобках указываем номер элемента массива:

while (rdr.Read()){

data.Add(new string[4]);

data[data.Count - 1][0] = rdr[0].ToString();

* Функция, которая закрывает форму и возвращается к Главной: private void button\_Cancel\_Click(object sender, EventArgs e)

#### **3.2.1.3 Методы класса AddStaff. Форма «Добавление нового сотрудника»**

* Функция проверки заполненности всех полей формы и добавления нового сотрудника в базу: private void Button\_AddStaff\_Click(object sender, EventArgs e)

Т. к. эта функция будет использована в 3 формах, опишем ее подробно. Условие для проверки поля для ввода на его заполненность: if (string.IsNullOrWhiteSpace(badge\_number\_box.Text)).

Открываем соединение, подключаемся к таблице (если она отсутствует, создаем ее). Передаем каждый полученный параметр в колонку своего типа: cmd.Parameters.AddWithValue("@FullName", fio\_box.Text); После передачи всех данных, закрываем соединение. В диалоговом окне оповещаем о результате.

* Функция, которая закрывает форму и возвращается к Главной: private void Button\_Cancel\_Click(object sender, EventArgs e)

#### **3.2.1.4 Методы класса DeleteStaff. Форма «Удаление сотрудника»**

* Функция, проверки заполненности поля и удаления сотрудника: private async void Button\_delete\_Click(object sender, EventArgs e

Функция удаления используется только в этой форме, опишем ее. В первую очередь, осуществляется проверка на заполненность поля, по которому будет происходить удаление: if (string.IsNullOrEmpty(number\_box.Text).

В нашем случае, это номер значка. Далее, по известному нам пути, открываем соединение с БД, подключаемся к таблице, прописываем необходимую команду (удаление): cmd.CommandText = "DELETE FROM staff WHERE (BadgeNumber) = @BadgeNumber"; Указываем столбец, по которому осуществляется удаление. Передаем данные и закрываем соединение. Оповещаем пользователя о результате операции.

* Функция, которая закрывает форму и возвращается к Главной: private void button\_Cancel\_Click(object sender, EventArgs e)

#### **3.2.1.5 Методы класса Detained\_AK. Форма «Задержанные на основе АК РФ»**

* Функция, которая вызывает загрузку данных из БД в таблицу: private void LoadData()
* Функция, которая закрывает форму и возвращается к Главной: private void button\_Cancel\_Click(object sender, EventArgs e)
* Функция, экспортирующая данные из таблицы в Excel: private void button\_Export\_Excele\_Click(object sender, EventArgs e)

Эта функция будет использована в 2 формах. Опишем ее подробно. Для работы со средой MO Excel необходимо подключить дополнительный класс и пространство имен через директиву: using Microsoft.Office.Interop.Excel;

Создаем новое подключение, книгу, лист (указываем его название). При помощи первого цикла стягиваем названия столбцов.

for (int i = 1; i < dataGridView1.Columns.Count+1; i++

Вторым и третьим (вложенным) циклом считываем данные из нашего компонента dataGridView1.

for (int i = 0; i < dataGridView1.Rows.Count; i++) {

for (int j = 0; j < dataGridView1.Columns.Count; j++)

В диалоговом окне пользователь выбирает место сохранения, изменяет название файла (если дефолтное его не устроит).

#### **3.2.1.6 Методы класса AddAK. Форма «Добавление задержанного АК»**

* Функция проверки изменения состояния переключателя: private void CheckBox\_Yes\_CheckedChanged(object sender, EventArgs e)
* Функция проверки изменения состояния переключателя: private void CheckBox\_No\_CheckedChanged(object sender, EventArgs e)
* Функция проверки на заполненность строго 1 переключателя: private bool Check\_Current\_CheckBox()
* Функция корректности для ввода даты: private string Get\_Correct\_Date\_Birth()
* Функция проверки заполненности всех полей формы и добавления задержанного в базу: private void Button\_Add\_Click(object sender, EventArgs e)
* Функция, которая закрывает форму и возвращается к Главной: private void button\_Cancel\_Click(object sender, EventArgs e)

#### **3.2.1.7 Методы класса Detained\_UK. Форма «Задержанные на основе УК РФ)**

* Функция, которая вызывает загрузку данных из БД в таблицу: private void LoadData()
* Функция, которая закрывает форму и возвращается к Главной: private void button\_Cancel\_Click(object sender, EventArgs e)
* Функция, экспортирующая данные из таблицы в Excel: private void button\_Export\_Excele\_Click(object sender, EventArgs e)
* Функция, которая закрывает форму и возвращается к Главной: private void button\_Cancel\_Click(object sender, EventArgs e)

#### **3.2.1.8 Методы класса AddUK. Форма «Добавление задержанного УК»**

* Функция проверки изменения состояния переключателя: private void CheckBox\_Yes\_CheckedChanged(object sender, EventArgs e)
* Функция проверки изменения состояния переключателя: private void CheckBox\_No\_CheckedChanged(object sender, EventArgs e)
* Функция проверки на заполненность строго 1 переключателя: private bool Check\_Current\_CheckBox()
* Функция корректности для ввода даты: private string Get\_Correct\_Date\_Birth()
* Функция проверки заполненности всех полей формы и добавления задержанного в базу: private void Button\_Add\_Click(object sender, EventArgs e)
* Функция, которая закрывает форму и возвращается к Главной: private void button\_Cancel\_Click(object sender, EventArgs e)

## **3.3 Работа с MySQL**

Для начала необходимо подключить ссылку MySql.Data. Теперь для удобства использования классов, через директиву using подключаем пространство имен MySql.Data.MySqlClient; Далее строки, которые будут подключаться к среде и к БД. Они имеют адрес сервер, данные пользователя, название БД. (все используемые данные описаны перед этими строками)

String connString = "Server=" + host + ";" + ";port=" + port + ";User Id=" + username + ";password=" + password;

String connString\_with\_DB = "Server=" + host + ";" + ";database=" + database + ";port=" + port + ";User Id=" + username + ";password=" + password;

Создадим экземпляр класса, который будет устанавливать соединение с нашей БД: MySqlConnection conn = new MySqlConnection(connString);

В качестве аргумента передаем нашу строку подключения. Объект создан. Необходимо установить соединение, а по окончании работы, соответственно, закрыть его. Для этого вызываем методы: conn.Open(); //открытие подключения conn.Close(); //закрытие(разрыв) подключения

Используем команду на создание БД (если ранее она не была создана), далее команда на создание таблиц в ней (при том же условии). Console.WriteLine(ex.ToString()); - выводим запрошенные данные.

## **3.4 Результат**

После запуска приложения мы попадаем на Главную форму, на которой находятся основные кнопки работы. (см. рис. 3.1):

* Таблица с данными о сотрудниках
* Формы добавления и удаления сотрудников
* Таблица с данными о задержанных на основе АК РФ
* Форма добавления нового задержанного
* Таблица с данными о задержанных на основе УК РФ
* Форма добавления нового задержанного
* Кнопка выхода из приложения и завершения работы

Приложение защищает базу данных от некорректных данных, которые пытается ввести пользователь, и предотвращает наличие незаполненных полей в базе.

## **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

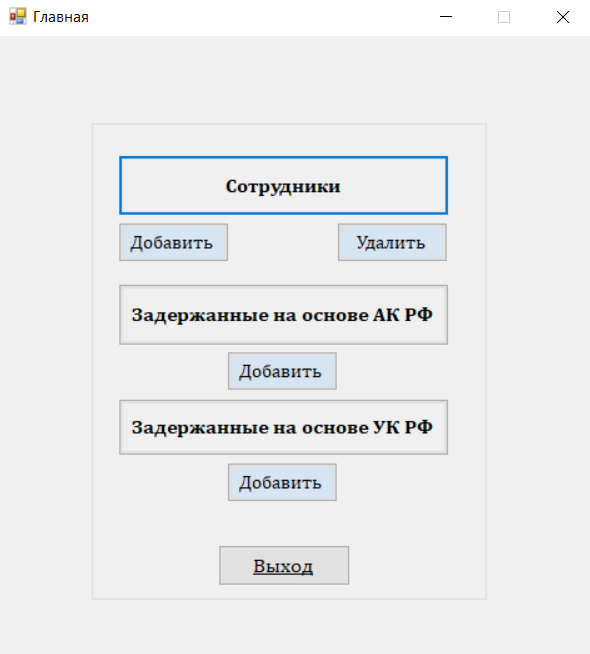
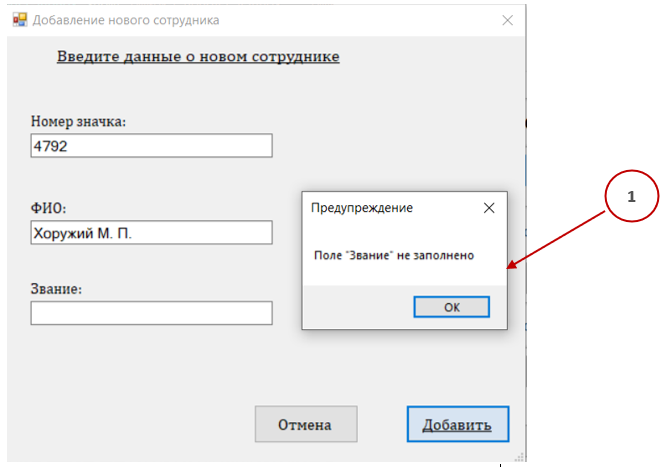
После запуска приложения открывается Главная форма. (см. рис. 4.1 )

Рисунок 4.1 - Главная форма

Формы, содержащие информацию в таблицах, имеют крупные (главные) кнопки, а для работы с базой имеют небольшие (второстепенные).

Итак. При первом запуске приложения автоматически создается БД, при условии, что ранее она отсутствовала. В последующих случаях будет автоматическое подключение.

При нажатии на кнопку «Добавить», относящуюся к данных о Сотрудниках, открывается такая форма. (см. рис. 4.2)

Рисунок 4.2 – Форма добавления сотрудника

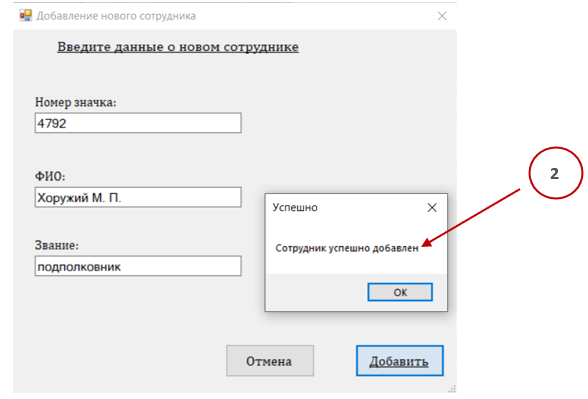
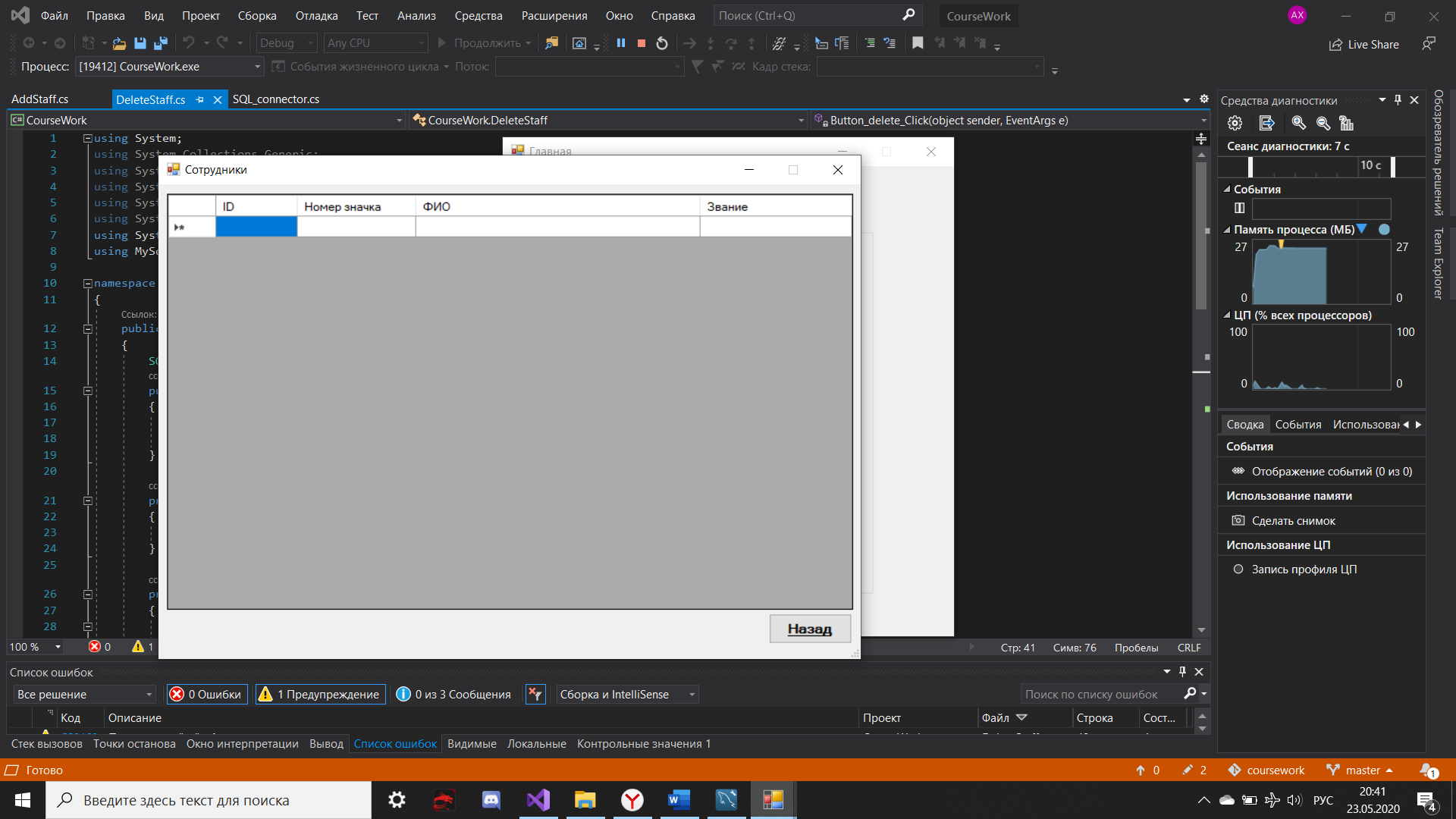
В поля ввода вам необходимо ввести данные о сотруднике. Для каждого поля есть проверка на его заполненность. В случае невнимательности или некорректного ввода, программа не позволит добавить такие данные. Диалоговое окно - область под номером 1, указывает на сообщение о предупреждении, в котором указано поле, которое вы по какой-либо причине не заполнили.

Рисунок 4.3 – форма успешного добавления

При корректном заполнении всех полей, сотрудник будет добавлен в базу. Об этом приложение также оповестит. (см. рис. 4.3, область 2)

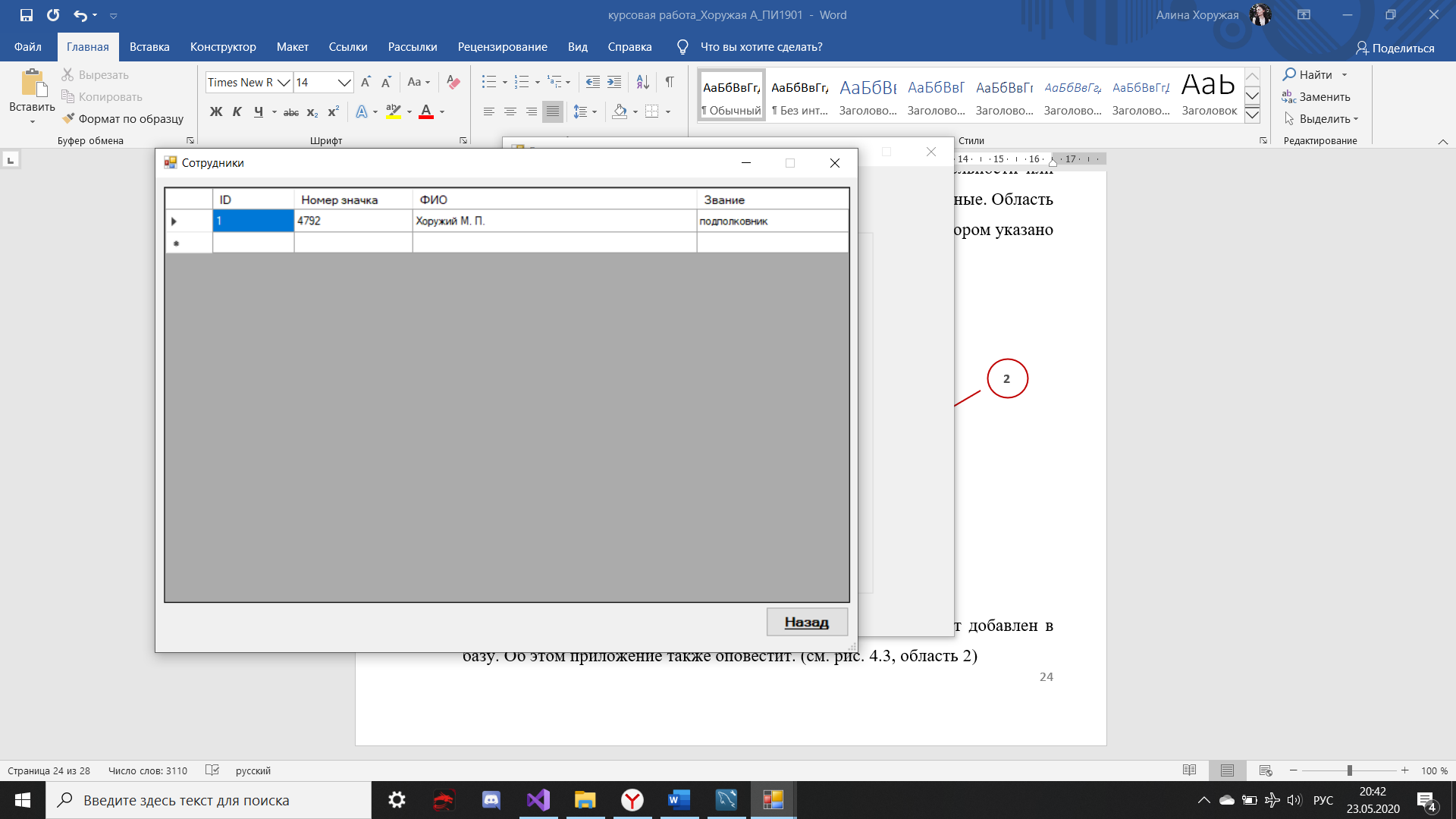
Рисунок 4.4 - Первоначальное состояние формы «Сотрудники»

Рисунок 4.5 – обновленная форма «Сотрудники»

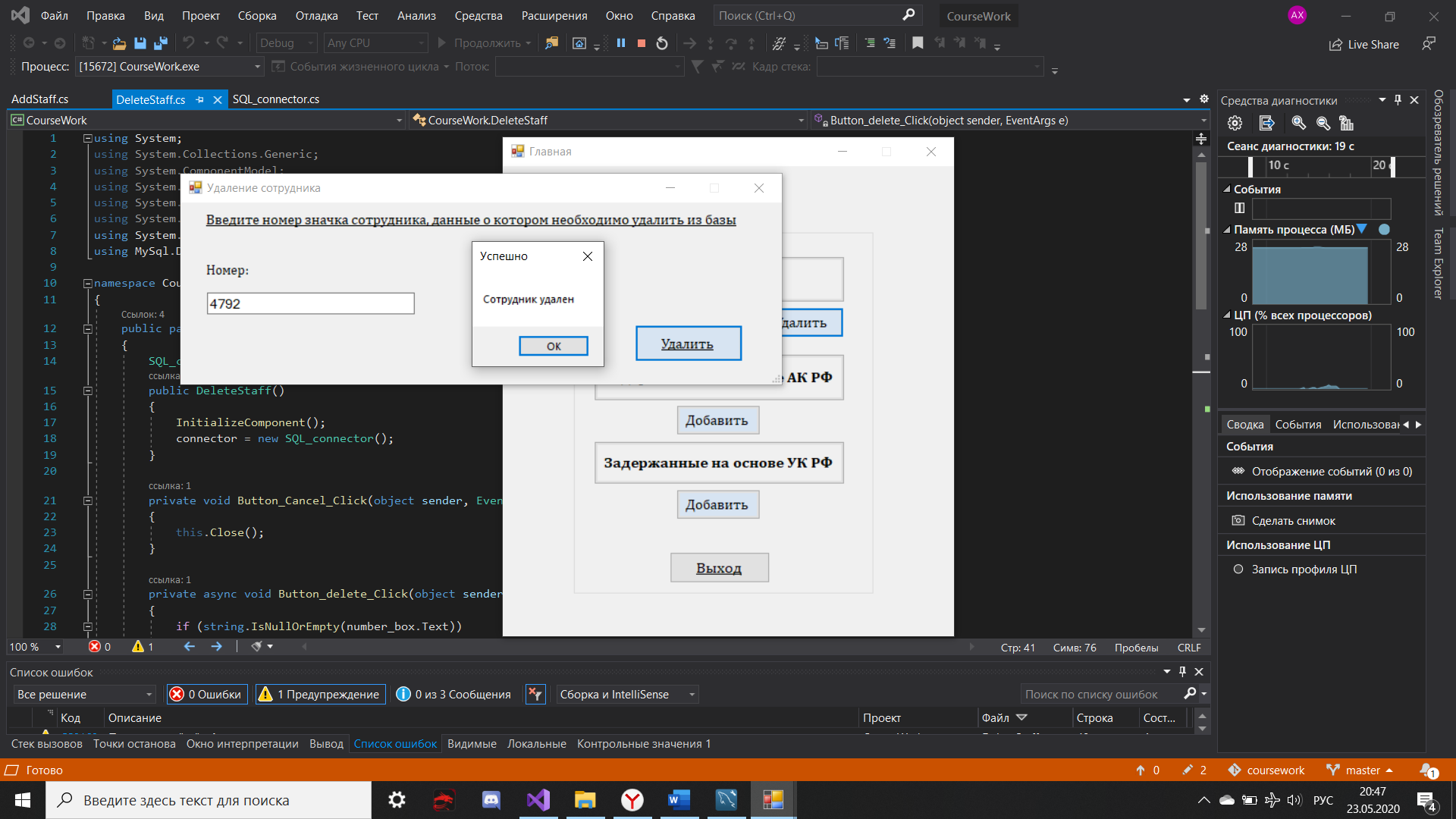
На рисунке 4.5 наблюдаем добавленные данные. Каждому сотруднику присвоен порядковый ID, номер значка (по которому будет осуществляться удаление сотрудника из базы), (см. рис. 4.6).

Рисунок 4.6 – удаление сотрудника

Форма удаления так же имеет проверку на заполненность поля ввода. В ином случае будет выведено диалоговое окно - область 2 (см. рис. 4.3)

В нашем случае, после удаления единственного сотрудника, база примет вид: (см. рис. 4.4).

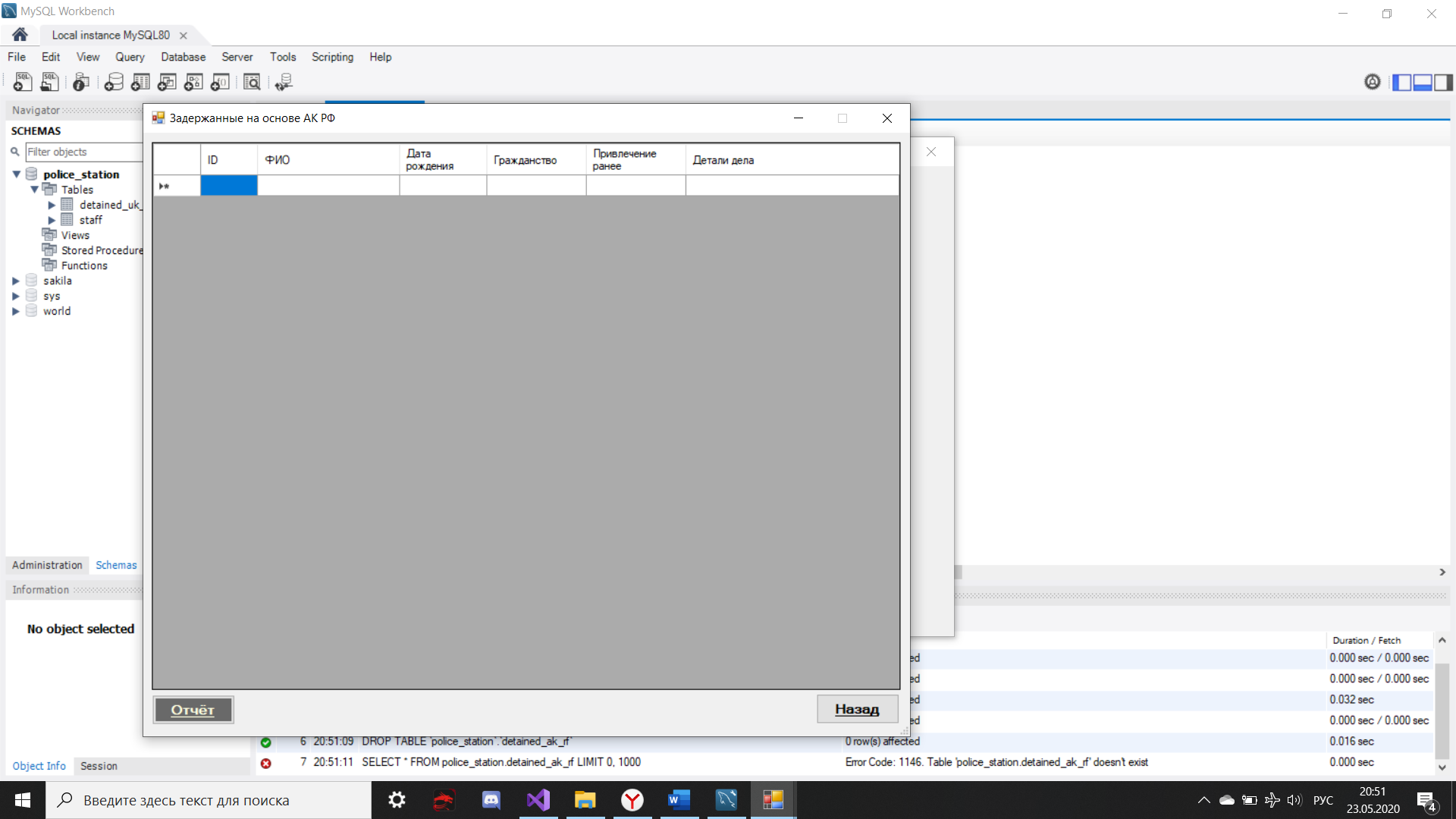
Базы с задержанными так же имеют «пустой» вид (см. рис. 4.7)

Рисунок 4.7 – форма с данными о задержанных

Информация о задержанных по АК и УК имеют одинаковые поля, т.к. в обоих случаях заполняются шаблонные данные: ФИО, дата рождения, гражданство, привлечение ранее, детали дела.

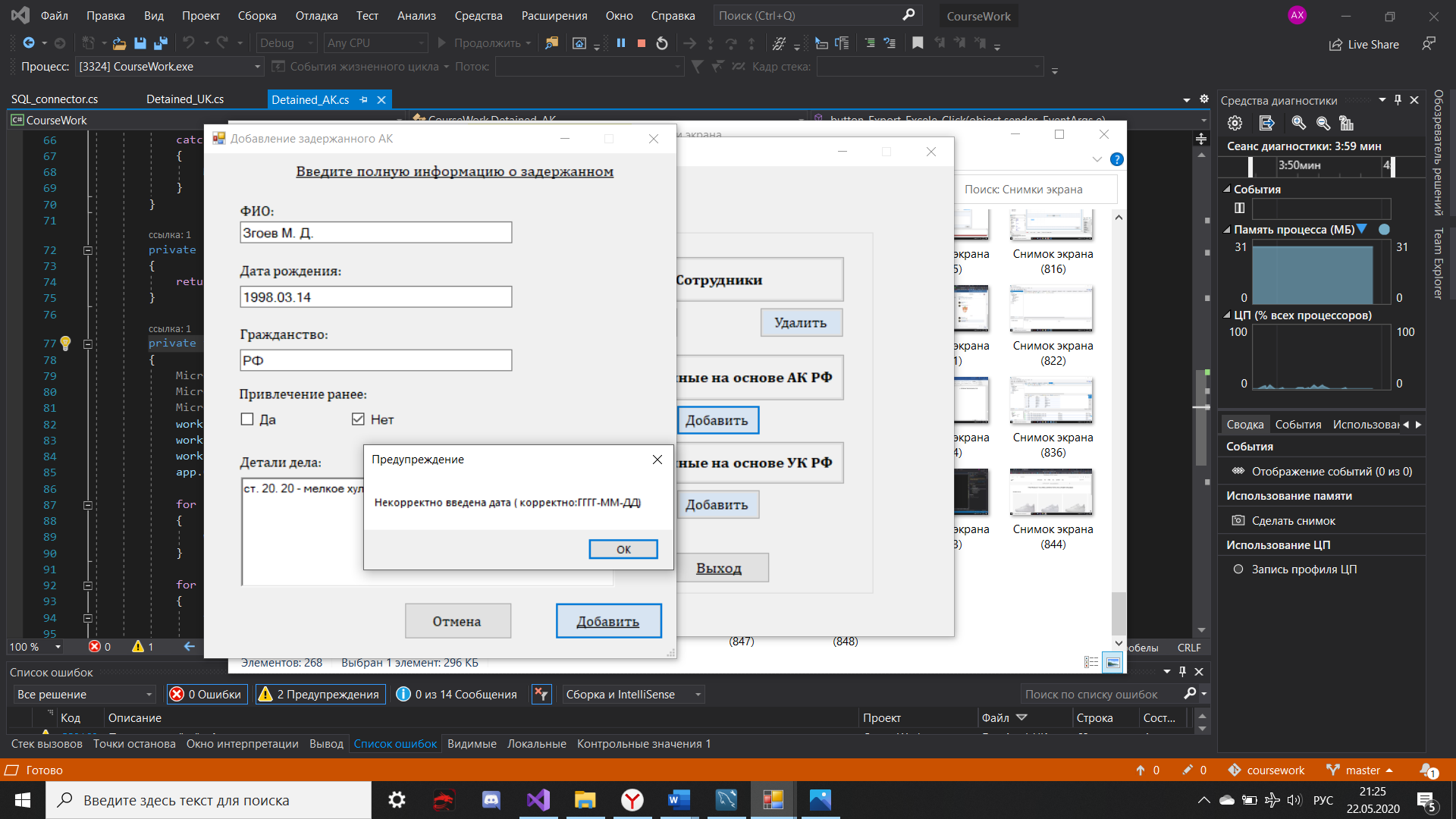
Формы добавления задержанных так же имеют защиту от некорректно введенных данных, например, на формат даты (см. рис. 4.8). Форма заполнения данных достаточно проста и требует лишь внимания. Наличие ранних привлечений к ответственности указывается в один клик по ячейке с указателем (Да\Нет). Для даты эталонным выбран единый формат записи: ГГГГ-ММ-ДД, с учетом всех возможных ошибочных вариантов. Область на рисунке 4.8 поможет корректно осуществить ввод.

Рисунок 4.8 – ввод даты

Форма ввода данных о задержанных на основе УК РФ имеет такой же вид и состав, как форма АК. А именно: проверку на заполненность всех полей и на корректный ввод даты. При успешном добавлении задержанного также будет оповещение подобное при добавлении сотрудника. (см. рис. 4.3 обл. 2)

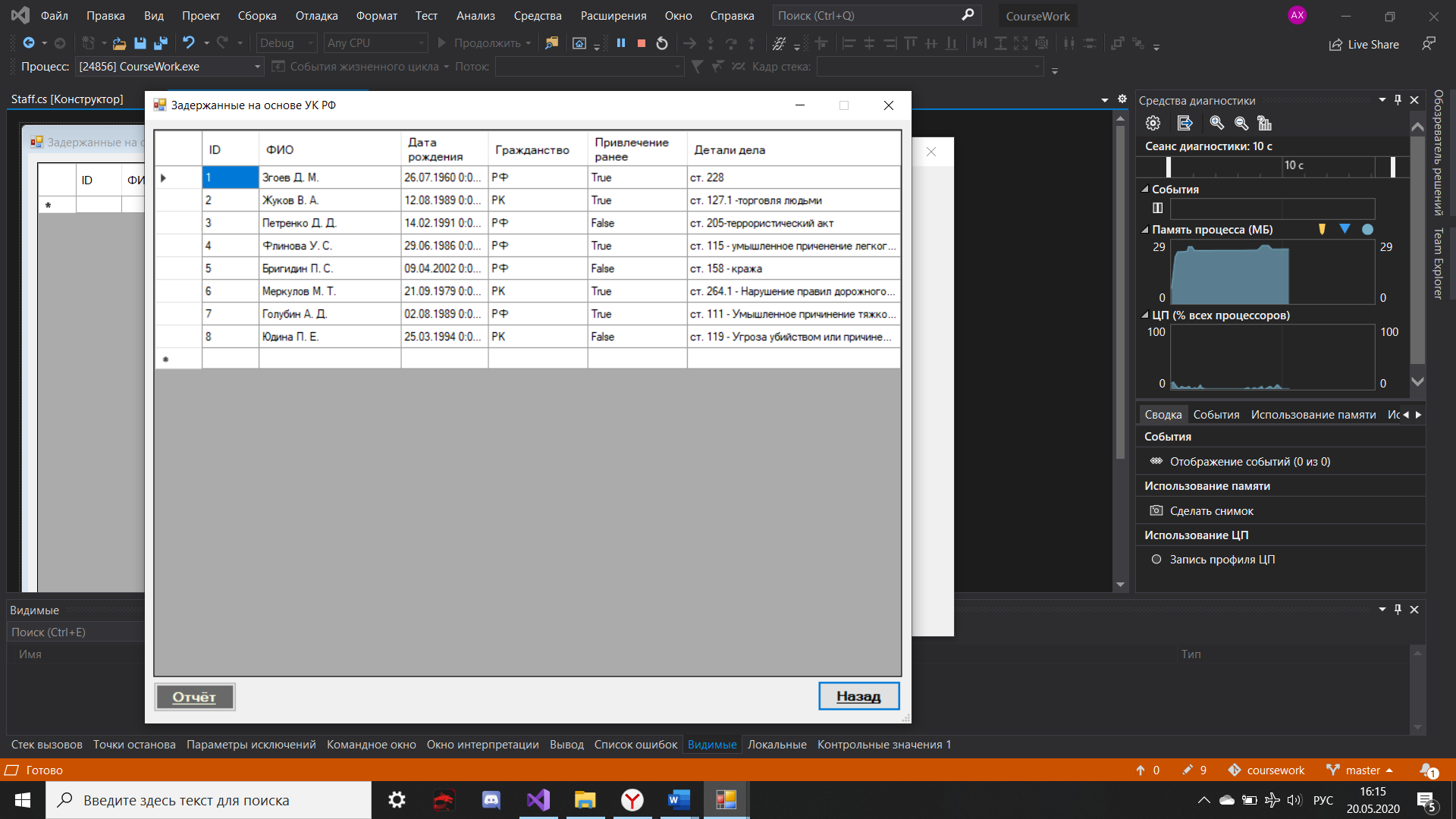
Теперь поговорим немного об экспорте данных о задержанных в Excel. Для экспорта имеется отдельная кнопка «Отчет» на формах с таблицами задержанных.

Рисунок 4.9 – кнопка экспорта данных в Excel

Экспорт имеет смысл и доступен только для баз с данными о задержанных. Давайте пошагово пройдем процедуру экспорта.

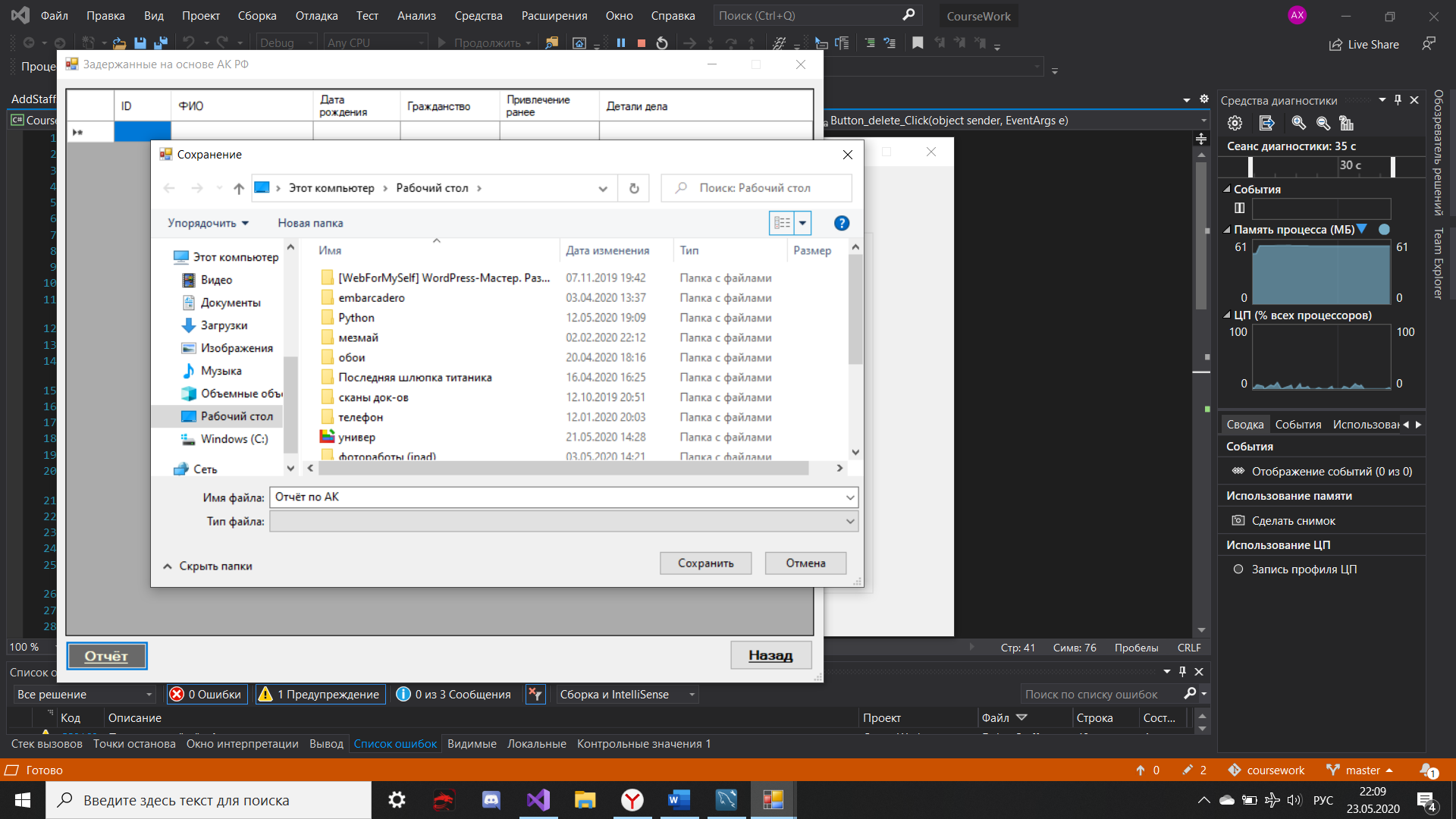
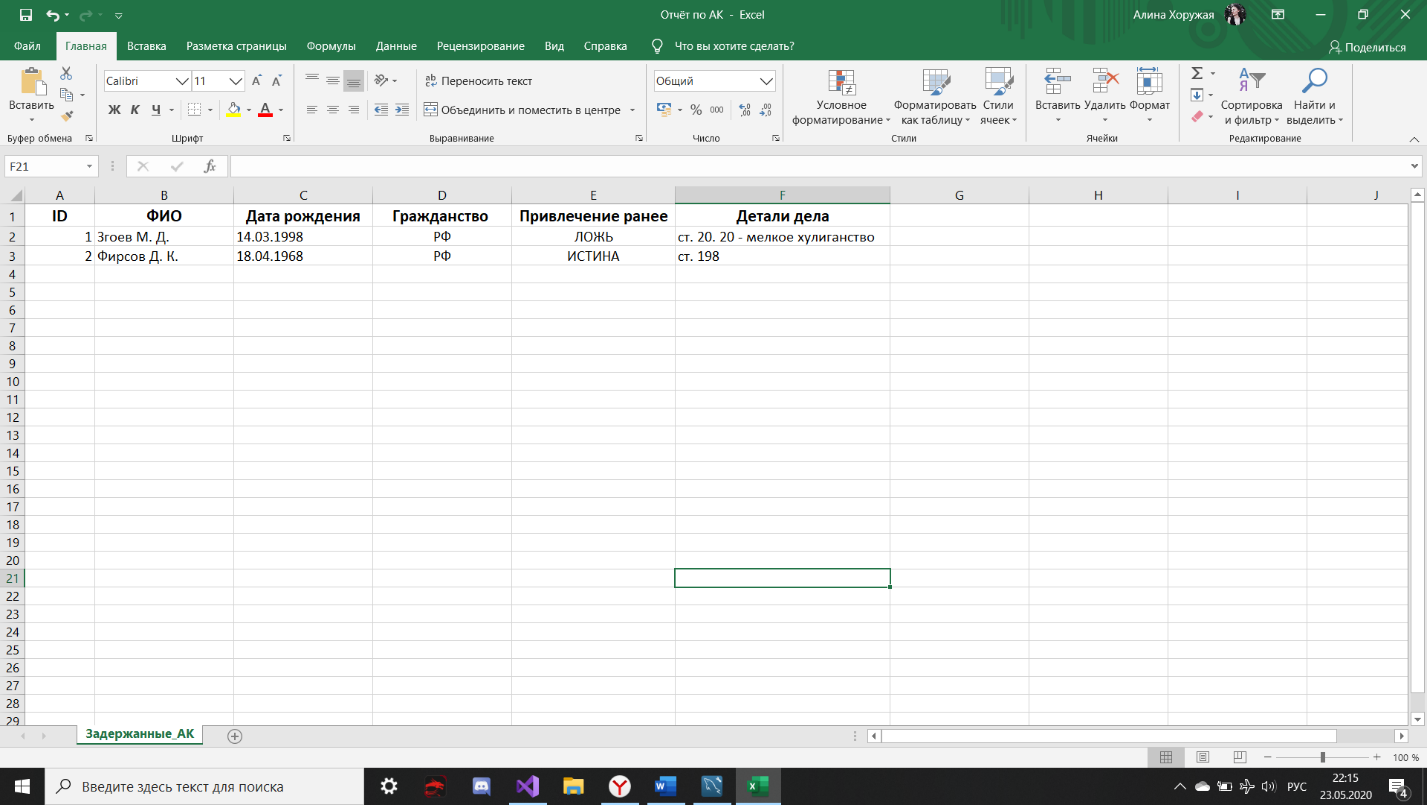
При нажатии на кнопку «Отчет» открывается диалоговое окно, которое предлагает выбрать место сохранения файла. Файл уже имеет дефолтное название в зависимости от кодекса (АК или УК) и расширение типа xlsx, (см. рис. 4.10)

Рисунок 4.10 – сохранение данных в Excel

На выходе мы получаем данные в таком формате (см. рис. 4.11):

Рисунок 4.11 – выходные данные

Завершить работу с БД и приложением вы можете в один клик по кнопке «Выход».

## **ЗАКЛЮЧЕНИЕ**

Перед выполнением работы ставилась задача, о создании приложения для управления БД. Посредством изучения предметной области, были выбраны дизайн и структура приложения. Зная, насколько важна точность в подобной сфере деятельности, приложение защищает БД от некорректных и неточных данных.

В процессе разработки были улучшены навыки в программировании на языке C#, а также более детально изучены графические компоненты Windows Forms в среде Microsoft Visual Studio. БД основана на свободной реляционной системе My SQL.

Результатом проделанной работы является готовый продукт - приложение для ведения БД «Полицейский участок», который может служить полноценной основой для проектирование наиболее масштабного и детального, а главное безопасного приложения.

## **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Объектно-ориентированное программирование: учеб. пособие /Е. А. Иванова, Н. В. Ефанова, Т. А. Крамаренко. – Краснодар : КубГАУ,2018 – 86 с.
2. Лойко В.И. Алгоритмы и структуры данных : учеб. пособие / В.И.Лойко, Н.В. Ефанова, Е.А. Иванова. – Краснодар : КубГАУ, 2018 – 206 с.
3. Программирование на C# : учеб. пособие : в 2 ч., Ч. 1/Дубровин В. В. – Тамбов: ТГТУ, 2017-81с.
4. Марченко А.Л. C# 2.0. Введение в программирование: Учебное пособие. - М.: Изд-во МГУ им. М.В. Ломоносова, 2005. - 258 с.
5. Изучаем PHP и MySQL — Линн Бейли, Майкл Моррисон: Эксмо, 2010г.
6. Настройка приложений баз данных — Б. А. Новиков, Г. Р. Домбровская: БХВ-Петербург, 2006г.

## **ПРИЛОЖЕНИЕ**

**Файл SQL\_connector.cs (Файл подключения к БД):**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using MySql.Data.MySqlClient;

namespace CourseWork

{

class SQL\_connector

{

static string host = "127.0.0.1"; //подключение к серверу

static int port = 3306;

static string database = "police\_station"; //название БД

static string username = "admin";

static string password = "password";

//авторизация

String connString = "Server=" + host + ";" + ";port=" + port + ";User Id=" + username + ";password=" + password;

String connString\_with\_DB = "Server=" + host + ";" + ";database=" + database + ";port=" + port + ";User Id=" + username + ";password=" + password;

public SQL\_connector()

{

}

public void Get\_Connection\_First\_Time()

{

MySqlConnection conn = new MySqlConnection(connString);

try

{

conn.Open(); //открытие подключения

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = conn;

cmd.CommandText = "CREATE DATABASE IF NOT EXISTS `police\_station`;"; //создание БД, если не создана

cmd.ExecuteNonQuery();

conn.Close(); //закрытие подключения

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

}

conn = new MySqlConnection(connString\_with\_DB);

try

{

conn.Open();

MySqlCommand cmd = new MySqlCommand();

cmd.Connection = conn;

//создание таблиц с указанием типа данных полей

cmd.CommandText = "CREATE TABLE IF NOT EXISTS `staff`(id INTEGER PRIMARY KEY AUTO\_INCREMENT, BadgeNumber INT, FullName TEXT, Title TEXT); ";

cmd.ExecuteNonQuery();

cmd.CommandText = "CREATE TABLE IF NOT EXISTS `Detained\_AK\_RF`(id INTEGER PRIMARY KEY AUTO\_INCREMENT, FullName TEXT, DateOfBirth DATE, Citizenship TEXT, InvolvementOfEarlier BOOLEAN, Details TEXT); ";

cmd.ExecuteNonQuery();

cmd.CommandText = "CREATE TABLE IF NOT EXISTS `Detained\_UK\_RF`(id INTEGER PRIMARY KEY AUTO\_INCREMENT, FullName TEXT, DateOfBirth DATE, Citizenship TEXT, InvolvementOfEarlier BOOLEAN, Details TEXT); ";

cmd.ExecuteNonQuery();

conn.Close();

}

catch (Exception ex)

{

Console.WriteLine(ex.ToString());

}

}

public MySqlConnection Get\_Connection\_For\_Operations()

{

return new MySqlConnection(connString\_with\_DB);

}

}

}

**Файл Form1.cs (главная форма приложения)**

using MySql.Data.MySqlClient;

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 CourseWork

{

public partial class Form1 : Form

{

SQL\_connector connector;

public Form1()

{

InitializeComponent();

connector = new SQL\_connector();

connector.Get\_Connection\_First\_Time();

}

private void Button\_staff\_Click(object sender, EventArgs e) //кнопка перехода к форме с сотрудниками

{

DialogResult dr = new DialogResult();

Staff frm6 = new Staff();

dr = frm6.ShowDialog();

}

private void Button\_exit\_Click(object sender, EventArgs e) //кнопка выхода из приложения

{

MessageBoxButtons msb = MessageBoxButtons.YesNo;

String message = "Вы действительно хотите выйти?"; //диалоговое окно

String caption = "Выход";

if (MessageBox.Show(message, caption, msb) == DialogResult.Yes)

this.Close(); //закрытие приложения

}

private void Button\_AK\_Click(object sender, EventArgs e) //кнопка перехода к форме с задержанными по АК РФ

{

DialogResult dr = new DialogResult();

Detained\_AK frm7 = new Detained\_AK();

dr = frm7.ShowDialog();

}

private void Button\_UK\_Click(object sender, EventArgs e) //кнопка перехода к форме с задержанными по УК РФ

{

DialogResult dr = new DialogResult();

Detained\_UK frm8 = new Detained\_UK();

dr = frm8.ShowDialog();

}

private void Button\_add\_staff\_Click(object sender, EventArgs e) //кнопка перехода к форме добавления сотрудника

{

DialogResult dr = new DialogResult();

AddStaff frm2 = new AddStaff();

dr = frm2.ShowDialog();

}

private void Button\_delete\_staff\_Click(object sender, EventArgs e) //кнопка перехода к форме удаления сотрудника

{

DialogResult dr = new DialogResult();

DeleteStaff frm5 = new DeleteStaff();

dr = frm5.ShowDialog();

}

private void Button\_add\_AK\_Click(object sender, EventArgs e) //кнопка перехода к форме добавления нового задержанного на основе АК РФ

{

DialogResult dr = new DialogResult();

AddAK frm4 = new AddAK();

dr = frm4.ShowDialog();

}

private void Button\_add\_UK\_Click(object sender, EventArgs e) //кнопка перехода к форме добавления нового задержанного на основе АК РФ

{

DialogResult dr = new DialogResult();

AddUK frm3 = new AddUK();

dr = frm3.ShowDialog();

}

}

}

**Файл Staff.cs**

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 MySql.Data.MySqlClient;

namespace CourseWork

{

public partial class Staff : Form

{

SQL\_connector connector;

public Staff()

{

InitializeComponent();

connector = new SQL\_connector();

connector.Get\_Connection\_First\_Time();

LoadData();

}

private void button\_Cancel\_Click(object sender, EventArgs e) //кнопка закрытия формы

{

this.Close();

}

private void LoadData()

{

MySqlConnection conn = connector.Get\_Connection\_For\_Operations(); //установление соединения

try

{

conn.Open();

string query = "SELECT \* FROM staff ORDER BY id"; //указываем таблицу, из которой выводятся данные

MySqlCommand cmd = new MySqlCommand(query);

cmd.Connection = conn;

MySqlDataReader rdr = cmd.ExecuteReader();

List<string[]> data = new List<string[]>();

while (rdr.Read())

{

data.Add(new string[4]);

data[data.Count - 1][0] = rdr[0].ToString(); //прочитываем 0 столбец

data[data.Count - 1][1] = rdr[1].ToString(); //прочитываем 1 столбец

data[data.Count - 1][2] = rdr[2].ToString(); //прочитываем 2 столбец

data[data.Count - 1][3] = rdr[3].ToString(); //прочитываем 3 столбец

}

rdr.Close();

foreach (string[] s in data)

dataGridView1.Rows.Add(s);

conn.Close();

}

catch (Exception ex)

{

MessageBox.Show("Ошибка вывода данных", "Ошибка"); //диалоговое окно при ошибке подключения и вывода данных

}

}

}

}

**Файл AddStaff.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace CourseWork

{

public partial class AddStaff : Form

{

SQL\_connector connector;

public AddStaff()

{

InitializeComponent();

connector = new SQL\_connector();

}

private void Button\_Cancel\_Click(object sender, EventArgs e)

{

this.Close(); //закрытие окна

}

private void Button\_AddStaff\_Click(object sender, EventArgs e)

{

if (string.IsNullOrWhiteSpace(badge\_number\_box.Text)) //проверка на заполненность поля

{

MessageBox.Show("Поле \"Номер значка\" не заполнено", "Предупреждение"); //предупреждение о незаполненном поле

return;

}

if (string.IsNullOrWhiteSpace(fio\_box.Text))

{

MessageBox.Show("Поле \"ФИО\" не заполнено", "Предупреждение"); //предупреждение о незаполненном поле

return;

}

if (string.IsNullOrWhiteSpace(title\_box.Text))

{

MessageBox.Show("Поле \"Звание\" не заполнено", "Предупреждение"); //предупреждение о незаполненном поле

return;

}

MySqlConnection conn = connector.Get\_Connection\_For\_Operations();

try

{

conn.Open(); //открываем соединение

MySqlCommand cmd = new MySqlCommand(); //подключаемся к таблице

cmd.Connection = conn;

cmd.CommandText = "INSERT INTO staff(BadgeNumber, FullName, Title) VALUES(@BadgeNumber, @FullName, @Title)"; //если таблица отсутствует, создает

cmd.Parameters.AddWithValue("@BadgeNumber", int.Parse(badge\_number\_box.Text));

cmd.Parameters.AddWithValue("@FullName", fio\_box.Text);

cmd.Parameters.AddWithValue("@Title", title\_box.Text);

cmd.ExecuteNonQuery();

conn.Close(); //передаем данные и закрываем соединение

}

catch (Exception ex)

{

MessageBox.Show("Ошибка добавления нового сотрудника", " Ошибка "); //сообщение о результате

}

MessageBox.Show("Сотрудник успешно добавлен", "Успешно");

this.Close();

}

}

}

**Файл DeleteStaff.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace CourseWork

{

public partial class DeleteStaff : Form

{

SQL\_connector connector;

public DeleteStaff()

{

InitializeComponent();

connector = new SQL\_connector();

}

private void Button\_Cancel\_Click(object sender, EventArgs e)

{

this.Close(); //закрытие формы

}

private async void Button\_delete\_Click(object sender, EventArgs e)

{

if (string.IsNullOrEmpty(number\_box.Text)) //проверка на заполненность поля

{

MessageBox.Show("Поле \"Номер значка\" не заполнено", "Предупреждение"); //предупреждение о незаполненном поле

return;

}

MySqlConnection conn = connector.Get\_Connection\_For\_Operations();

try

{

conn.Open(); //открываем соединение

MySqlCommand cmd = new MySqlCommand(); //подключаемся к таблице

cmd.Connection = conn;

cmd.CommandText = "DELETE FROM staff WHERE (BadgeNumber) = @BadgeNumber";

cmd.Parameters.AddWithValue("@BadgeNumber", int.Parse(number\_box.Text));

await cmd.ExecuteNonQueryAsync();

cmd.ExecuteNonQuery();

conn.Close(); //передаем данные и закрываем соединение

}

catch (Exception ex)

{

MessageBox.Show("Ошибка удаления сотрудника", " Ошибка "); //сообщение о результате

}

MessageBox.Show("Сотрудник удален", "Успешно");

this.Close();

}

}

}

**Файл Detained\_AK.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

using Microsoft.Office.Interop.Excel;

namespace CourseWork

{

public partial class Detained\_AK : Form

{

SQL\_connector connector;

List<string[]> data;

public Detained\_AK()

{

InitializeComponent();

connector = new SQL\_connector();

LoadData();

}

private void button\_Cancel\_Click(object sender, EventArgs e)

{

this.Close();

}

private void LoadData()

{

MySqlConnection conn = connector.Get\_Connection\_For\_Operations();

try

{

conn.Open(); //открываем соединение

string query = "SELECT \* FROM detained\_ak\_rf ORDER BY id"; //указываем таблицу из которой выносим данные

MySqlCommand cmd = new MySqlCommand(query);

cmd.Connection = conn;

MySqlDataReader rdr = cmd.ExecuteReader();

data = new List<string[]>();

while (rdr.Read())

{

data.Add(new string[6]); //считывание данных по столбцам

data[data.Count - 1][0] = rdr[0].ToString(); //ID

data[data.Count - 1][1] = rdr[1].ToString(); //FullName

data[data.Count - 1][2] = changeDataOfBirth(rdr[2].ToString()); //DateOfBirth

data[data.Count - 1][3] = rdr[3].ToString(); //Citizenship

data[data.Count - 1][4] = rdr[4].ToString(); //Involment

data[data.Count - 1][5] = rdr[5].ToString(); //Details

}

rdr.Close(); //заканчиваем считывание

foreach (string[] s in data)

dataGridView1.Rows.Add(s);

conn.Close(); //закрываем соединение

}

catch (Exception ex)

{

MessageBox.Show("Ошибка вывода данных", "Ошибка");

}

}

private string changeDataOfBirth(string value)

{

return value.Split(' ')[0]; //разбивает строку на подстроки

}

private void button\_Export\_Excele\_Click(object sender, EventArgs e) //кнопка экспорта

{

Microsoft.Office.Interop.Excel.\_Application app = new Microsoft.Office.Interop.Excel.Application();

Microsoft.Office.Interop.Excel.\_Workbook workbook = app.Workbooks.Add(Type.Missing);

Microsoft.Office.Interop.Excel.\_Worksheet worksheet = null;

worksheet = workbook.Sheets["Лист1"]; //создание листа

worksheet = workbook.ActiveSheet;

worksheet.Name = "Задержанные\_АК"; //название создаваемого листа в книге

app.Columns.ColumnWidth = 20; //ширина столбцов

for (int i = 1; i < dataGridView1.Columns.Count+1; i++)

{

worksheet.Cells[1, i] = dataGridView1.Columns[i - 1].HeaderText;

}

for (int i = 0; i < dataGridView1.Rows.Count; i++) //цикл считывания данных

{

for (int j = 0; j < dataGridView1.Columns.Count; j++)

{

worksheet.Cells[i + 2, j + 1] = dataGridView1.Rows[i].Cells[j].Value;

}

}

var saveFileDialoge = new SaveFileDialog(); //диалоговое окно

saveFileDialoge.FileName = "Отчёт по АК"; //дефолтное название сохраняемого файла

saveFileDialoge.DefaultExt = ".xlsx"; //расширение файла

if (saveFileDialoge.ShowDialog() == DialogResult.OK)

{

workbook.SaveAs(saveFileDialoge.FileName, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Microsoft.Office.Interop.Excel.XlSaveAsAccessMode.xlExclusive, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing);

}

app.Quit();

}

}

}

**Файл Detained\_UK.cs (абсолютно аналогичен файлу Detained\_AK, т.к. они имеют одинаковую структуру форм).**

**Файл AddAK.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Text;

using System.Text.RegularExpressions;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

using MySqlX.XDevAPI.Common;

namespace CourseWork

{

public partial class AddAK : Form

{

SQL\_connector connector;

public AddAK()

{

InitializeComponent();

connector = new SQL\_connector();

}

private void CheckBox\_Yes\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox\_Yes.Checked) checkBox\_No.Checked = false; //единственный выбор переключателя

}

private void CheckBox\_No\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox\_No.Checked) checkBox\_Yes.Checked = false; //единственный выбор переключателя

}

private void Button\_Cancel\_Click(object sender, EventArgs e)

{

this.Close(); //закрытие окна

}

private bool Check\_Current\_CheckBox()

{

bool result = false;

if (checkBox\_Yes.Checked) //проверка на заполненность чекбокса

{

result = true;

}

return result;

}

private string Get\_Correct\_Date\_Birth()

{

string result = "";

if (Regex.IsMatch(date\_of\_brith\_box.Text, @"[0-9]{4}\-[0-9]{2}\-[0-9]{2}", RegexOptions.IgnoreCase)) //условие для ввода даты

{

result = date\_of\_brith\_box.Text;

}

return result;

}

private void Button\_Add\_Click(object sender, EventArgs e) //проверка на заполненность всех полей

{

if (string.IsNullOrWhiteSpace(fio\_box.Text))

{

MessageBox.Show("Поле \"ФИО\" не заполнено", "Предупреждение"); //предупреждение о незаполненном поле

return;

}

if (string.IsNullOrWhiteSpace(date\_of\_brith\_box.Text))

{

MessageBox.Show("Поле\"Дата рождения\" не заполнено", "Предупреждение"); //предупреждение о незаполненном поле

return;

}

if (Get\_Correct\_Date\_Birth() == "")

{

MessageBox.Show("Некорректно введена дата ( корректно:ГГГГ-ММ-ДД) ", "Предупреждение"); //предупреждение о некорректно введенем формате даты

return;

}

if (string.IsNullOrWhiteSpace(zenship\_box.Text))

{

MessageBox.Show("Поле\"Гражданство\" не заполнено", "Предупреждение"); //предупреждение о незаполненном поле

return;

}

if (string.IsNullOrWhiteSpace(details\_case\_box.Text))

{

MessageBox.Show("Поле\"Детали дела\" не заполнено", "Предупреждение"); //предупреждение о незаполненном поле

return;

}

MySqlConnection conn = connector.Get\_Connection\_For\_Operations(); //открываем соединение

try

{

conn.Open();

MySqlCommand cmd = new MySqlCommand(); //подключаемся к таблице

cmd.Connection = conn;

cmd.CommandText = "INSERT INTO Detained\_AK\_RF(FullName, DateOfBirth, Citizenship, InvolvementOfEarlier, Details) VALUES (@FullName, @DateOfBirth, @Citizenship, @InvolvementOfEarlier, @Details)";

cmd.Parameters.AddWithValue("@FullName", fio\_box.Text); //передаем параметр ФИО

cmd.Parameters.AddWithValue("@DateOfBirth", date\_of\_brith\_box.Text); //передаем параметр дата рождения

cmd.Parameters.AddWithValue("@Citizenship", zenship\_box.Text); //передаем параметр гражданство

if (Check\_Current\_CheckBox())

{

cmd.Parameters.AddWithValue("@InvolvementOfEarlier", 1); //вывод 1 при нажатии на "ДА"

}

else {

cmd.Parameters.AddWithValue("@InvolvementOfEarlier",0); //вывод 0 при нажатии на "НЕТ"

}

cmd.Parameters.AddWithValue("@Details", details\_case\_box.Text);

cmd.ExecuteNonQuery();

conn.Close(); //передаем данные и закрываем соединение

MessageBox.Show("Задержанный добавлен", "Успешно");

this.Close();

}

catch (Exception ex)

{

MessageBox.Show("Ошибка добавления нового задержанного", " Ошибка "); //сообщение о результате

}

}

}

}

**Файл AddUK.cs (абсолютно аналогичен файлу AddAK, т.к. они имеют одинаковую структуру форм).**