Министерство образования Московской области

**Государственное автономное профессиональное образовательное**

**учреждение Московской области «Егорьевский техникум»**

Допущена к защите

заместитель директора

по учебной работе

\_\_\_\_\_\_\_\_\_ С.Г. Большова

«\_\_\_» июня 2023г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**(ДИПЛОМНАЯ РАБОТА)**

**на тему: «Разработка информационной системы Свияжские холмы»**

**Студент** Буцалин Иван Романович

**Специальность** 09.02.07 Информационные системы и программирование

**Группа** Ип-93

**Руководитель** В.А. Тараканова

г. Егорьевск, 2023 г.

Оглавление

[**Введение 3**](#_Toc135761418)

[**Глава 1. Подбор инструментов для создания информационной системы 4**](#_Toc135761419)

[**1.1. Системы управления базами данных (СУБД) 5**](#_Toc135761420)

[**1.2. Visual studio (2022) 9**](#_Toc135761421)

[**1.3. Windows Presentation Foundation (WPF) 11**](#_Toc135761422)

[**Глава 2. Создание, наполнение и содержание базы данных 13**](#_Toc135761423)

[**2.1. Создание базы данных 14**](#_Toc135761424)

[**2.2. Импорт данных 18**](#_Toc135761425)

# **Введение**

Актуальность данного дипломного проекта обусловлена потребностью рынка в специалистах в сфере информационных систем и программирования, которые способны разработать, протестировать, развернуть и отладить информационные системы подобные этой.

Этот проект создан для демонстрации знаний по специальности информационные системы и программирование. Для разработки будут использованы СУБД (Система управления базами данных) – SQL server, Visual studio 2022(Visual Studio это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Помимо стандартного редактора и отладчика, которые есть в большинстве сред IDE, Visual Studio включает в себя компиляторы, средства автоматического завершения кода, графические конструкторы и многие другие функции для улучшения процесса разработки), WPF (**Windows Presentation Foundation** (**WPF**) — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем)

# **Глава 1. Подбор инструментов для создания информационной системы**

Чтобы хорошо выполнить работу, нужен соответствующий набор инструментов – таких, чтобы они вызывали доверие и были пригодны для решения тех задач, с которыми столкнетесь; к тому же нужно уметь ими пользоваться. Чтобы написать достойный код, нужны опытный мастер, хорошие инструменты и умение пользоваться этими инструментами.

Это важный вопрос. Благодаря умелому применению инструментов можно выделиться на общем фоне как действительно эффективно работающий программист. В отдельных случаях эти инструменты могут определить успех или провал проекта. Неумолимое наступление промышленного производства программ требует, чтобы не упускалось никаких средств, которые позволят писать лучший код и делать это быстрее и надежнее.

1.1. Системы управления базами данных (СУБД)

СУБД – одна из самых важных фрагментов информационной системы ведь в ней хранятся данные о пользователях от клиентов, администраторов и директоров до цен на любые товары или услуги.

К основным функциям СУБД относят:

Управление данными на внешней памяти, то есть на дисках

Управление данными в оперативной памяти с использованием дискового кеша

Запись изменений, резервное копирование и восстановление из резервных копий после сбоев

Состав СУБД

Обычно в состав современных СУБД включают следующее:

**ядро**, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию;

**процессор языка базы данных**, обеспечивающий [оптимизацию запросов](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D1%82%D0%B8%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81%D0%BE%D0%B2_%D0%A1%D0%A3%D0%91%D0%94) на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода;

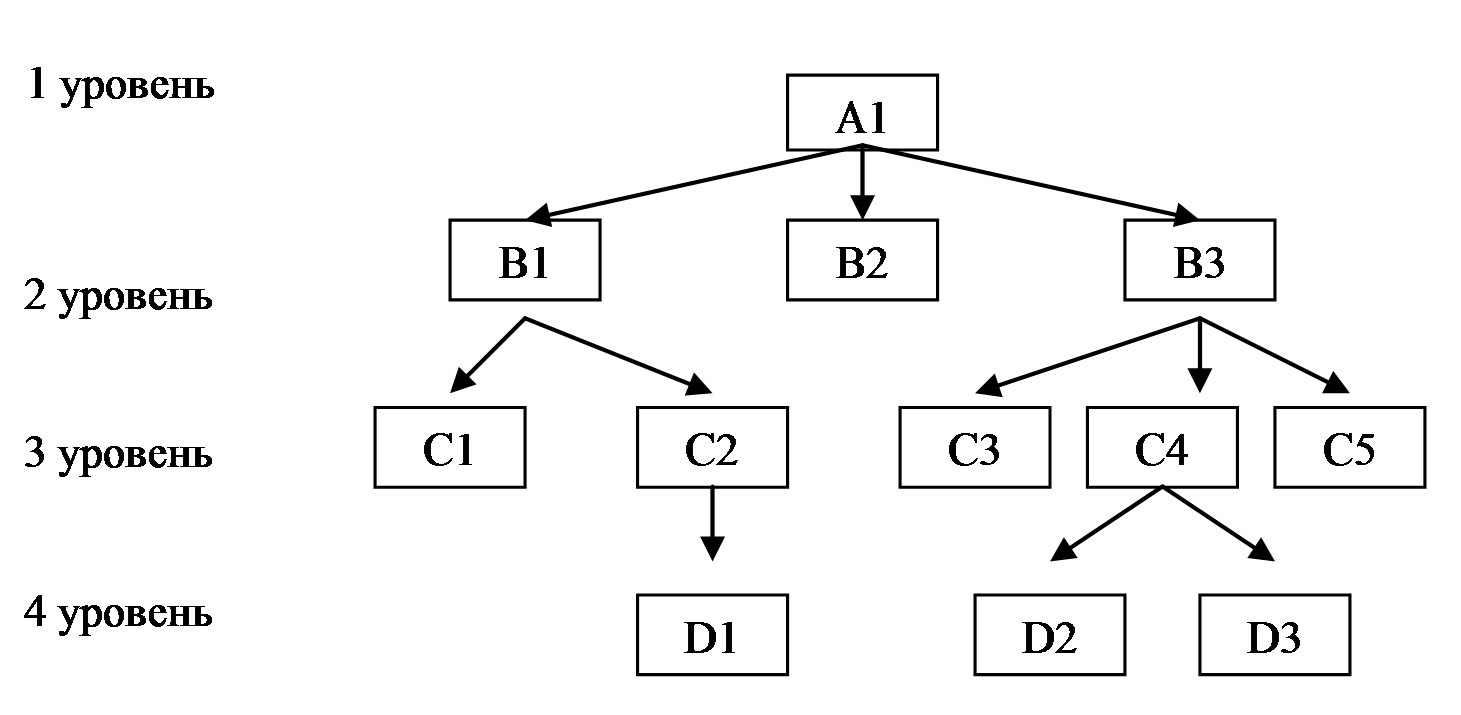
**подсистему поддержки времени исполнения**, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД;

**сервисные программы** (внешние [утилиты](https://ru.wikipedia.org/wiki/%D0%A3%D1%82%D0%B8%D0%BB%D0%B8%D1%82%D0%B0)), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

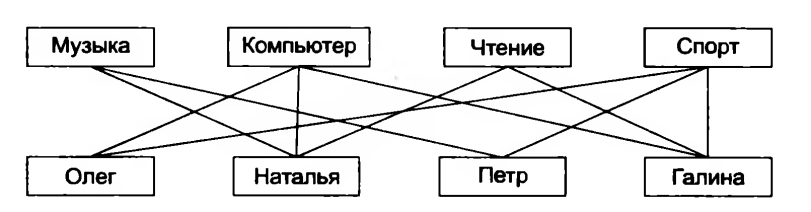
Классификация СУБД

По модели данных:

Иерархические



Сетевые



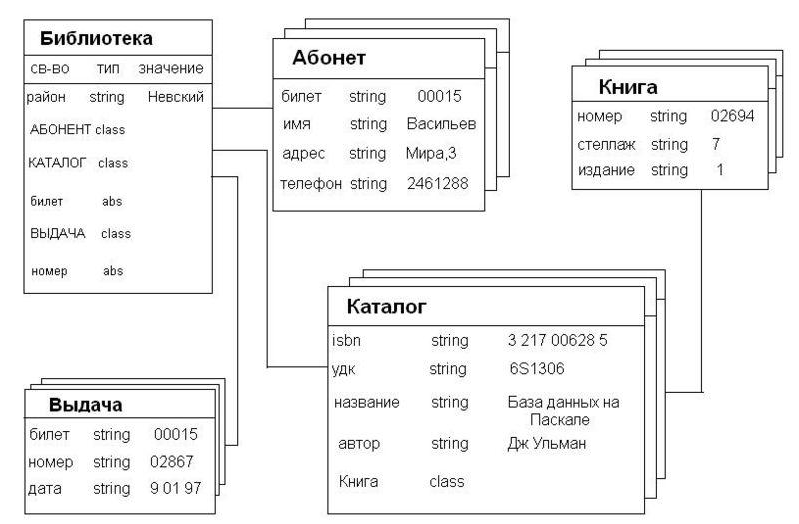
Реляционные



Объектно – ориентированные



Объектно – реляционные



По степени распределенности:

Локальные (на жестком диске)

Распределенные (облачные)

По способу доступа:

[Файл-серверные](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)

В файл-серверных СУБД файлы данных располагаются централизованно на [файл-сервере](https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80). СУБД располагается на каждом клиентском компьютере (рабочей станции). Доступ СУБД к данным осуществляется через [локальную сеть](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%BA%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%81%D0%B5%D1%82%D1%8C). Синхронизация чтений и обновлений осуществляется посредством файловых блокировок.

Преимуществом этой архитектуры является низкая нагрузка на процессор файлового сервера.

Недостатки: потенциально высокая загрузка локальной сети; затруднённость или невозможность [централизованного](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BD%D1%82%D1%80%D0%B0%D0%BB%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F) [управления](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BE%D1%80%D0%B8%D1%8F_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F); затруднённость или невозможность обеспечения таких важных характеристик, как высокая [надёжность](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B4%D1%91%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D1%8C), [высокая доступность](https://ru.wikipedia.org/wiki/%D0%92%D1%8B%D1%81%D0%BE%D0%BA%D0%B0%D1%8F_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%BD%D0%BE%D1%81%D1%82%D1%8C) и высокая [безопасность](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C).

[Клиент-серверные](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94)

Клиент-серверная СУБД располагается на сервере вместе с БД и осуществляет доступ к БД непосредственно, в монопольном режиме. Все клиентские запросы на обработку данных обрабатываются клиент-серверной СУБД централизованно.

Недостаток клиент-серверных СУБД состоит в повышенных требованиях к серверу.

Достоинства: потенциально более низкая загрузка локальной сети; удобство централизованного управления; удобство обеспечения таких важных характеристик, как высокая надёжность, высокая доступность и высокая безопасность.

[Встраиваемые](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D1%82%D1%80%D0%B0%D0%B8%D0%B2%D0%B0%D0%B5%D0%BC%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94)

Встраиваемая СУБД — СУБД, которая может поставляться как составная часть некоторого программного продукта, не требуя процедуры самостоятельной [установки](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%81%D1%82%D0%B0%D0%BB%D0%BB%D1%8F%D1%86%D0%B8%D1%8F_(%D0%9F%D0%9E)). Встраиваемая СУБД предназначена для локального хранения данных своего приложения и не рассчитана на коллективное использование в сети.

Физически встраиваемая СУБД чаще всего реализована в виде [подключаемой библиотеки](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)). Доступ к данным со стороны приложения может происходить через [SQL](https://ru.wikipedia.org/wiki/SQL) либо через специальные [программные интерфейсы](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D1%8B%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81).

По данным параметрам для разработки данного проекта лучше всего подойдет SQL Server, потому что придется работать с реляционной базой данных на жестком диске компьютера с возможностью переноса с помощью съемных носителей

1.2. Visual studio (2022)

Visual Studio — это мощное средство разработчика, которое можно использовать для единого завершения всего цикла разработки. Это комплексная интегрированная среда разработки (IDE), которую можно использовать для написания, редактирования, отладки и сборки кода, а затем для развертывания приложения. Помимо редактирования и отладки кода, Visual Studio включает компиляторы, средства завершения кода, систему управления версиями, расширения и многие другие функции для улучшения каждого этапа процесса разработки программного обеспечения.

Благодаря поддержке различных функций и языков в Visual Studio вы можете развиваться от написания первой программы "Hello World" до разработки и развертывания приложений. Например, можно создавать, отлаживать и тестировать приложения .NET и C++, изменять ASP.NET страницы в представлении веб-конструктора, разрабатывать кроссплатформенные мобильные и классические приложения с помощью .NET или создавать адаптивные веб-интерфейсы пользовательского интерфейса на C#.

Visual Studio предоставляет разработчикам полнофункциональную среду разработки для эффективной и совместной разработки высококачественного кода.

Установщик на основе рабочей нагрузки — установка только того, что вам нужно

Мощные средства и функции программирования — все, что необходимо для создания приложений в одном месте

Поддержка нескольких языков — код в C++, C#, JavaScript, TypeScript, Python и т. д.

Кроссплатформенная разработка — создание приложений для любой платформы

Интеграция системы управления версиями — совместная работа над кодом с коллегами по команде

Начало работы с Visual Studio

Разработка кода

Интегрированная среда разработки Visual Studio предоставляет множество функций, упрощающих написание кода и управление им. Например, вы можете быстро и точно использовать предложения по коду IntelliSense, быстро улучшить код с помощью лампочек, предлагающих действия, или развернуть или свернуть блоки кода с помощью структурирования. Организуйте и изучите код с помощью Обозреватель решений, который показывает код, упорядоченный по файлам, или представление классов, в которое отображается код, упорядоченный по классам.

Сборка приложения

Вы можете компилировать и создавать приложения, чтобы немедленно создавать сборки и тестировать их в отладчике. Для проектов C++ и C# можно запускать многопроцессорные сборки. Visual Studio также предоставляет несколько параметров, которые можно настроить при создании приложений. Вы можете создать настраиваемую конфигурацию сборки в дополнение к встроенным конфигурациям, скрыть определенные предупреждающие сообщения или увеличить выходные данные сборки.

Отладка кода

Встроенная отладка в Visual Studio позволяет легко выполнять отладку, профилирование и диагностику. Вы выполняете пошаговое выполнение кода и просматриваете значения, хранящиеся в переменных, задаете контрольные значения для переменных, чтобы увидеть, когда значения изменяются, проверяется путь выполнения кода и другие способы отладки кода во время его выполнения.

Для разработки данной информационной системы понадобиться мощное средство разработки, поддерживающее написание программы на языке C# с графическим интерфейсом на технологии WPF, поэтому выбор пал именно на Visual Studio, версия 22 года была выбрана как самая свежая и оптимизированная.

1.3. Windows Presentation Foundation (WPF)

Платформы пользовательского интерфейса, которая не зависит от разрешения и использует векторный механизм визуализации, способный использовать все преимущества современного графического оборудования. WPF предоставляет комплексный набор функций разработки приложений, которые включают в себя язык XAML, элементы управления, привязку к данным, макет, двумерную и трехмерную графику, анимацию, стили, шаблоны, документы, мультимедиа, текст и типографические функции. WPF является частью .NET, поэтому вы можете создавать приложения, включающие другие элементы .NET API.

Написание кода с WPF

WPF существует в виде подмножества типов .NET, которые по большей части находятся в пространстве имен System.Windows. Если ранее вы создавали приложения в .NET с помощью таких платформ, как ASP.NET и Windows Forms, то должны быть знакомы с фундаментальными принципами программирования WPF. Вы выполняете следующие действия.

Создание экземпляров классов

Настройка свойств

Вызов методов

Обработка событий

WPF включает в себя дополнительные конструкции программирования, которые расширяют возможности свойств и событий

Разметка и код

WPF позволяет разрабатывать приложения, используя как разметку, так и код программной части, что привычно для разработчиков на ASP.NET. Разметка XAML обычно используется для определения внешнего вида приложения, а управляемые языки программирования (код программной части) — для реализации его поведения. Такое разделение внешнего вида и поведения имеет ряд преимуществ.

Затраты на разработку и обслуживание снижаются, так как разметка, определяющая внешний вид, не связана тесно с кодом, обуславливающим поведение.

Повышается эффективность разработки, так как дизайнеры, занимающиеся внешним видом приложения, могут работать параллельно с разработчиками, реализующими поведение приложения.

Ниже перечислены встроенные элементы управления WPF.

Кнопки: Button и RepeatButton.

Вывод данных: DataGrid, ListView и TreeView.

Вывод и выбор дат: Calendar и DatePicker.

Диалоговые окна: OpenFileDialog, PrintDialog и SaveFileDialog.

Рукописный ввод: InkCanvas и InkPresenter.

Документы: DocumentViewer, FlowDocumentPageViewer, FlowDocumentReader, FlowDocumentScrollViewerи StickyNoteControl.

Ввод: TextBox, RichTextBox и PasswordBox.

Макет: Border, BulletDecorator, Canvas, DockPanel, Expander, Grid, GridView, GridSplitter, GroupBox, Panel, ResizeGrip, Separator, ScrollBar, ScrollViewer, StackPanel, Thumb, Viewbox, VirtualizingStackPanel, Window и WrapPanel.

Мультимедиа: Image, MediaElement и SoundPlayerAction.

Меню: ContextMenu, Menu и ToolBar.

Навигация: Frame, Hyperlink, Page, NavigationWindow и TabControl.

Выбор: CheckBox, ComboBox, ListBox, RadioButton и Slider.

Информирование пользователя: AccessText, Label, Popup, ProgressBar, StatusBar, TextBlock и ToolTip.

Вышеперечисленных возможностей уже достаточно для создания информационной системы WPF позволит сделать приложение приятным глазу и понятным даже не очень опытному пользователю благодаря своему большому инструментарию

# **Глава 2. Создание, наполнение и содержание базы данных**

В SQL Server понадобиться создать базу данных содержащую таблицы с данными о работниках, клиентах, услугах и заказах, потребуется так же создать дополнительную таблицу для создания связей между таблицами без нарушения 3ей нормальной формы.

Таблица работники будет содержать информацию:

ID работника, должность, имя, фамилию, отчество, логин, пароль, дату последнего входа, тип последнего входа (успешно/не успешно).

Таблица клиенты будет содержать информацию:

Имя, фамилия, отчество, ID клиента, серию паспорта, номер паспорта, дату рождения, адрес, е-mail, пароль.

Таблица услуги будет содержать информацию:

ID услуги, название, код, цену

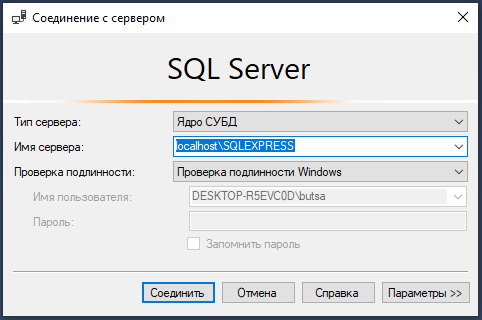
Таблица заказы будет содержать информацию:

ID заказа, код заказа, дата создания заказа, время создания заказа, ID клиента,

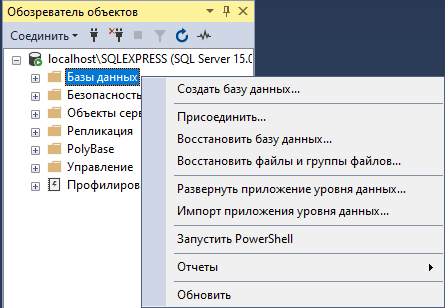
Статус, дата закрытия, время аренды.

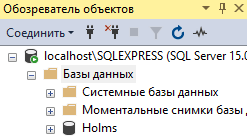
## **2.1. Создание базы данных**

Для создания базы данных потребуется открыть и подключиться к локальному серверу SQL



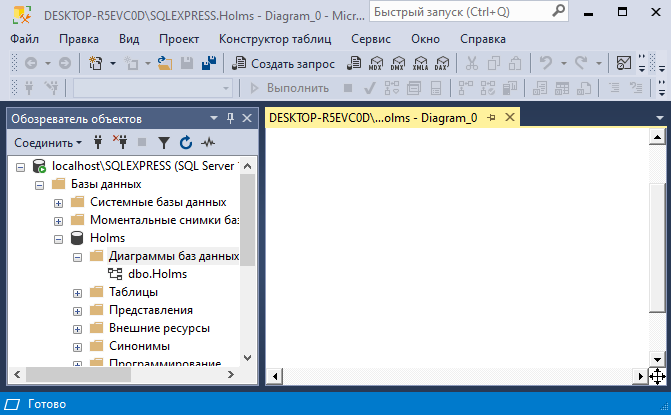
В обозревателе объектов найти папку «Базы данных», нажать на нее правой клавишей мыши и выбрать пункт «Создать базу данных»



После создания базы данных она отобразиться в обозревателе объектов 

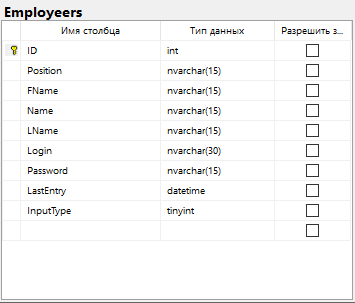
Теперь можно приступить к созданию диаграммы, таблиц и связей.

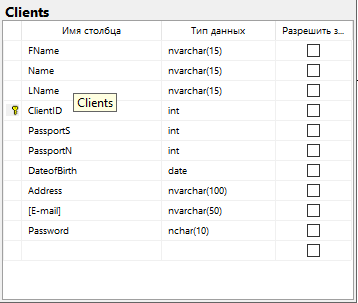
В папке «диаграммы баз данных» нужно создать новую пустую диаграмму

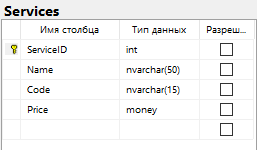


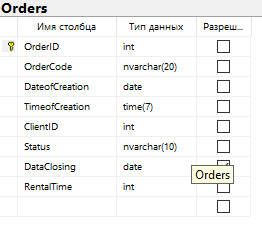
В диаграмме создать 4 таблицы Employeers, Clients, Services, Order.

Наполнить их соответствующими столбцами и указать нужные типы данных



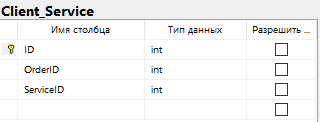






Теперь понадобиться создать еще одну таблицу: Client\_Services

Эта таблица будет соединять таблицу клиентов с таблицей услуг так что бы не нарушать 3ю нормальную форму



Соединяю таблицы таким образом с помощью ключей:

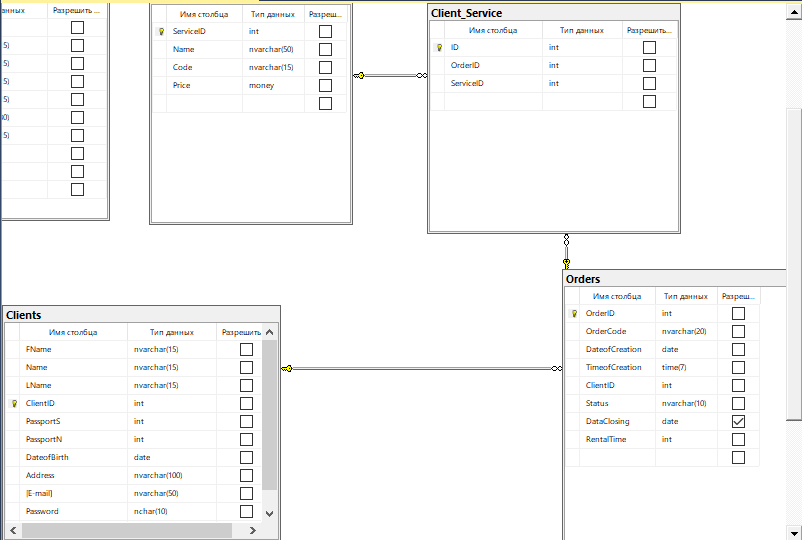
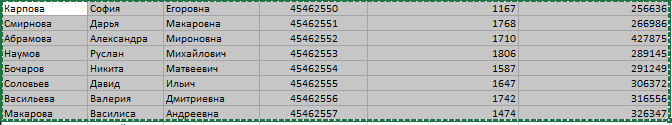


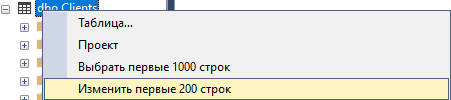
Диаграмма и таблицы базы данных готовы осталось импортировать данные.

## **2.2. Импорт данных**

Импорт данных будет происходить из таблиц Excel следующим образом:

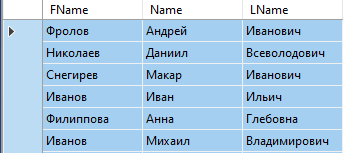
Открываю нужную таблицу, выделяю в ней данные и сочетанием ctrl+с копирую их



Открываю нужную таблицу в SQL 

И нажав на верхний левый угол таблицы выделяю все строки и столбцы

Нажимаю сочетание клавиш ctrl+v для того что бы вставить данные



По аналогии поступаю со всеми таблицами и получаю в итоге готовую базу данных в которой есть: диаграмма, таблицы, связи между таблицами, и данные импортированны в таблицы.

