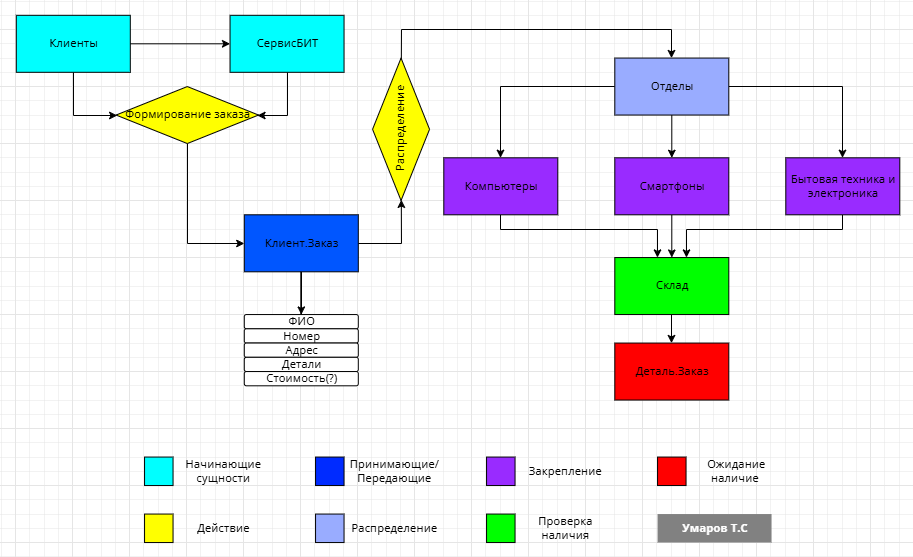
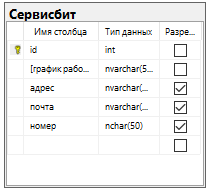
**Разработка ER-диаграммы / Логическая модель**

  
*Рис. 01 «ER-диаграмма»*

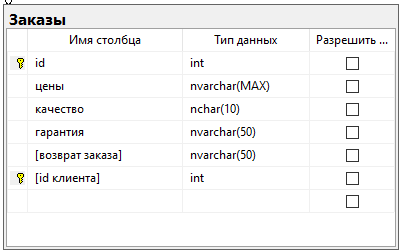
ER-диаграммы позволяют наглядно представить структуру базы данных, выявить возможные проблемы и неоднозначности в определении сущностей и связей, а также облегчить процесс коммуникации между разработчиками и пользователями системы. После создания и согласования ER-диаграммы, она может быть преобразована в логическую модель базы данных, которая будет служить основой для физического проектирования и реализации базы данных.

**Разработка таблиц для базы данных / Физическая модель**

  
Рис. 02 «Таблица СервисБит»

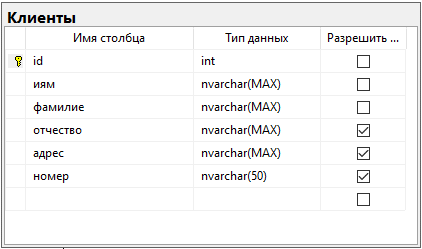
**Таблица «Сервисбит»** является ключевым справочником контактных данных и графика работы компании. Она обеспечивает согласованность и доступность важной информации для отделов продаж, поддержки и маркетинга.

С её помощью можно централизованно управлять рабочим временем, адресами, номерами телефонов и почтовыми ящиками, что критично для обеспечения бесперебойной работы с клиентами и партнерами.

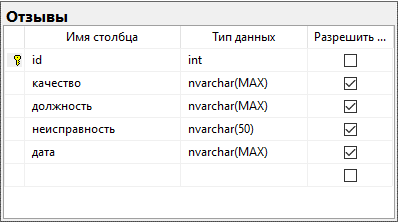
  
Рис. 02 «Таблица Заказы»

**Таблица «Заказы»** фиксирует финансовую и сервисную сторону сделок: цены, качество, гарантийные обязательства и возможность возврата. Связь с клиентами через «id клиента» позволяет проводить полный анализ продаж, определять наиболее прибыльные услуги и выстраивать эффективную ценовую политику, учитывая историю заказов и предпочтения клиентов.

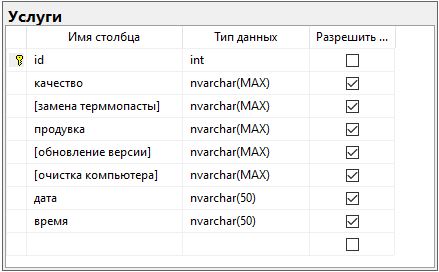
СДЕЛАНО УМАРОВ Т.С (П-41)

  
Рис. 03 «Таблица Клиенты»

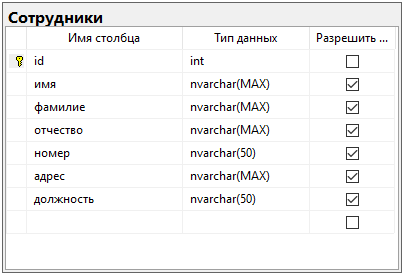
**Таблица «Клиенты»** является ключевым инструментом для управления взаимоотношениями. Она хранит идентификационные и контактные данные (имя, фамилия, адрес, номер телефона), что позволяет отслеживать историю взаимодействий, персонализировать обслуживание и анализировать клиентскую базу для повышения уровня удовлетворенности.

  
Рис. 04 «Таблица Отзывы»

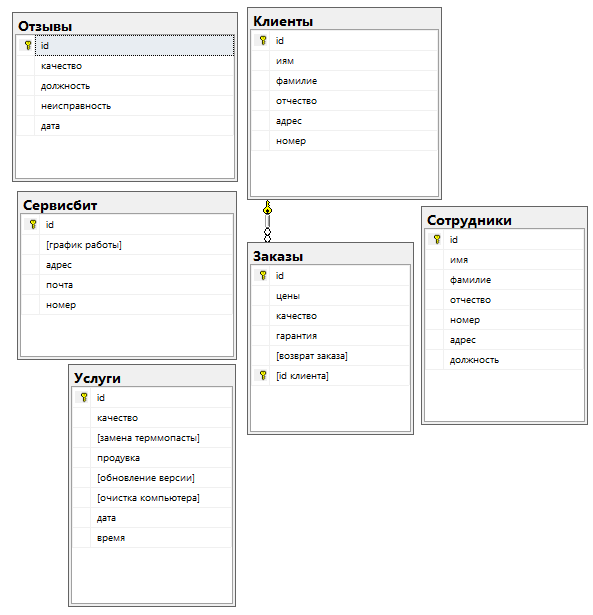
**Таблица «Отзывы»** предназначена для сбора и анализа обратной связи от клиентов. Она хранит оценки качества обслуживания, информацию о неисправностях и должности сотрудников, что позволяет выявлять слабые места в работе, оценивать эффективность персонала и улучшать общий уровень клиентского сервиса.

  
Рис. 05 «Таблица Услуги»

**Таблица «Услуги»** является ядром системы учета предоставляемых сервисов. Она позволяет фиксировать перечень работ (замена термопасты, продувка, обновление версий), их качество, дату и время оказания. Это обеспечивает контроль над процессом обслуживания, помогает планировать ресурсы и анализировать востребованность отдельных услуг для оптимизации сервисного предложения.

  
Рис. 06 «Таблица Сотрудники»

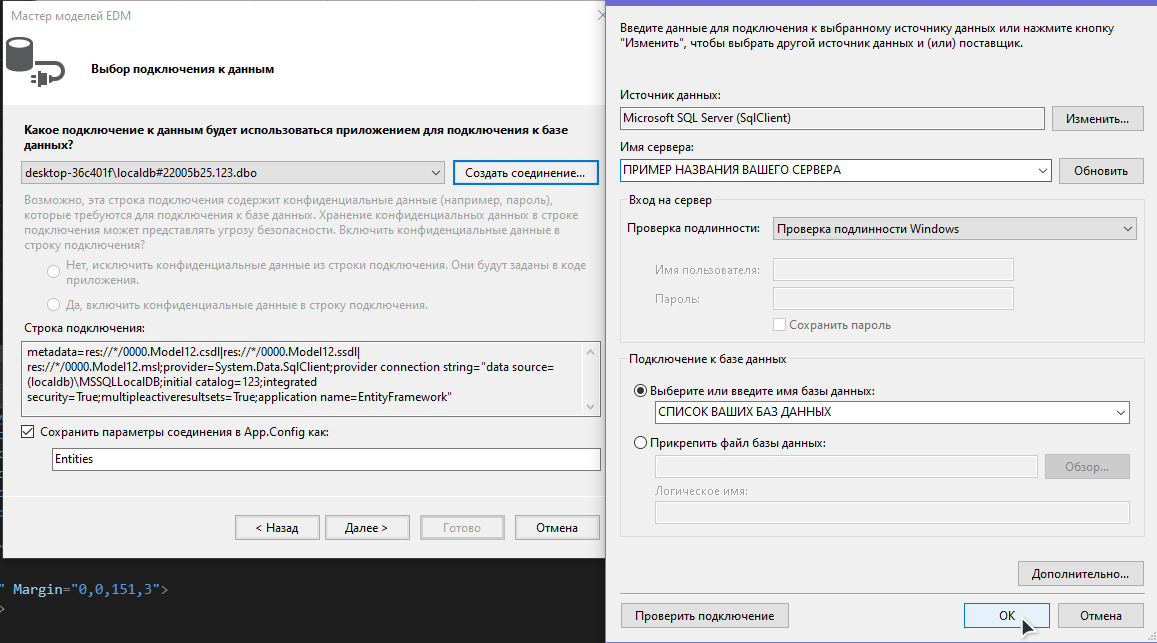
**Таблица «Сотрудники»** служит для централизованного учета персонала компании. Она содержит личные данные, контакты и должности сотрудников, что необходимо для координации работы, планирования задач и обеспечения контроля над исполнением обязанностей. Интеграция с другими таблицами позволяет анализировать эффективность работы каждого специалиста.

  
Рис. 07 «Общий вид БД»

**Интеграция базы данных в приложение**

Интеграция базы данных в приложение Visual Studio C# с использованием Модели ADO.NET EDM (Entity Data Model) происходит в несколько этапов. Вот пошаговое руководство:

1. Создание проекта: Откройте Visual Studio и создайте новый проект C# (например, Windows Forms или WPF).
2. Добавление элемента данных: В обозревателе решений (Solution Explorer) щелкните правой кнопкой мыши на проекте и выберите "Добавить" > "Новый элемент". В диалоговом окне "Добавить новый элемент" выберите "Модель ADO.NET EDM" и задайте имя для файла модели (например, MyDatabaseModel.edmx).
3. Настройка подключения к базе данных: В мастере "Модель сущности ADO.NET" выберите "Создать из базы данных" и нажмите "Далее". В следующем окне выберите источник данных (например, Microsoft SQL Server) и укажите строку подключения к базе данных. Если необходимо, введите учетные данные для подключения к серверу базы данных. Нажмите "Далее":



*Рис.9 «Выбор базы данных или ее ручной ввод»*

1. Выбор объектов базы данных: В мастере выберите объекты базы данных (таблицы, представления, хранимые процедуры), которые вы хотите интегрировать в ваше приложение. Нажмите "Завершить" для создания модели.
2. Использование модели в коде: Теперь вы можете использовать созданную модель в коде вашего приложения. Для этого создайте экземпляр контекста базы данных и выполняйте операции с данными(вписывать данный код нужно в **[yourDBname].Context.cs**:

private static TimurEntities \_context;

public static TimurEntities GetContext()

{

if (\_context == null)

\_context = new TimurEntities ();

return \_context;

}