

МИНОБРНАУКИ РОССИИ

ФГБОУ ВО

РТУ МИРЭА

**Колледж программирования и кибербезопасности**

**Отчет №1**

«Задание УП»

По УП.01.01 Учебная практика

Специальность 09.02.07 Информационные системы и программирование

Выполнил студент

Группы ПКС-35

Неструев О. Д.

Проверил преподаватель

Стоколос М. Д.

Москва

2024

**Цель:** разработать ER-диаграмму программного продукта, выбрать СУБД для базы данных, разместить базу данных в выбранной СУБД, выбрать язык программирования для разработки программного продукта, выбрать ИСР для разработки, Интегрировать базу данных в программный продукт.

# Разработка ER-диаграммы программного продукта.

Для разработки ER-диаграммы используется сайт «ERDPlus», который предоставляет нужный функционал. Пример ER-диаграммы представлен на рисунке 1.

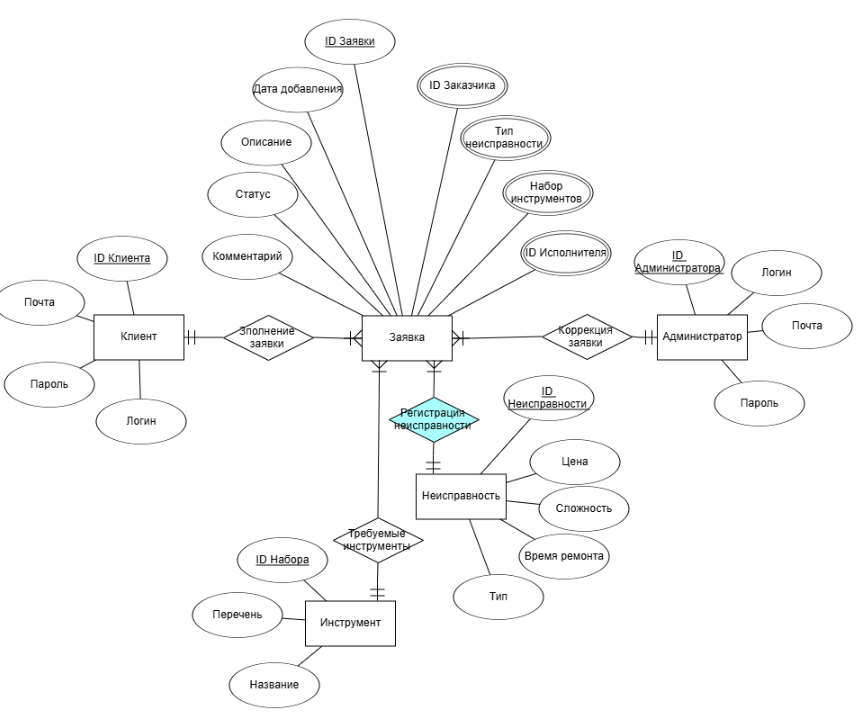


Рис. 1 пример ER-диаграммы

При разработке ER-диаграммы было выявлено пять сущностей, такие как:

* «Клиент»;
* «Администратор»;
* «Инструмент»;
* «Неисправность»;
* «Заявка».

Сущность «Клиент» хранит в себе следующие атрибуты:

* «ID Клиента» (первичный ключ);
* «Логин»;
* «Пароль»;
* «Почта».

Сущность «Администратор» содержит в себе следующие атрибуты:

* «ID Администратора» (первичный ключ);
* «Логин»;
* «Пароль»;
* «Почта».

Сущность «Инструмент» содержит в себе следующие атрибуты:

* «ID Набора» (первичный ключ);
* «Перечень»;
* «Название».

Сущность «Неисправность» содержит в себе следующие атрибуты:

* «ID Неисправности» (первичный ключ);
* «Цена»;
* «Сложность»;
* «Тип»;
* «Время работы».

Сущность «Заявка» содержит в себе следующие атрибуты:

* «ID Заявки» (первичный ключ);
* «Дата добавления»;
* «Описание»;
* «Статус»;
* «Комментарий»;
* «ID Заказчика» (вторичный ключ от сущности «Клиент»);
* «ID Исполнителя» (вторичный ключ от сущности «Администратор»);
* «Тип неисправности» (вторичный ключ от сущности «Неисправность»);
* «Набор инструментов» (вторичный ключ от сущности «Инструмент»);

Клиент может создать несколько заявок, однако заявка может принадлежать только одному клиенту. Это связь «Один ко многим» между сущностями «Клиент» и «Заявка».

Администратор может работать с одной заявкой, однако заявка не может принадлежать нескольким администраторам и исполнителям. Это связь «Один ко многим» между сущностями «Администратор» и «Заявка».

Набор инструментов может использоваться в нескольких заявках, однако одна заявка не может ссылаться на несколько наборов. Это связь «Один ко многим» между сущностями «Инструмент» и «Заявка».

Одна неисправность может присутствовать в нескольких заявках, однако одна заявка хранит в себе одну неисправность. Это связь «Один ко многим» между сущностями «Неисправность» и «Заявка».

# Выбор СУБД

В качестве СУБД был выбран MS SQL. При выборе оценивались инструменты СУБД и совместная работа с языками программированиями.

# Создание базы данных

Для создания базы данных использовался язык T-SQL, который позволяет создавать базы данных, таблицы и заполнять их, а также изменять и удалять записи.

Создание базы данных «RepairService» представлено в следующем Y-SQL запросе:

-- Создание базы данных

create database RepairService;

Создание таблицы «Клиент» представлено в следующем T-SQL запросе:

-- Создание таблицы Клиент

create table Клиент(

[ID Клиента] int primary key not null,

Логин nchar(30) not null,

Пароль nchar(10) not null,

Почта nchar(50) not null,

)

В таблице определяется первичный ключ primary key, который позволяет различать записи. Тип данных nchar() позволяет хранить текстовую информацию.

Создание таблицы «Администратор» представлено в следующем T-SQL запросе:

-- Создание таблицы Администратор

create table Администратор(

[ID Администратора] int primary key not null,

Логин nchar(30) not null,

Пароль nchar(10) not null,

Почта nchar(50) not null,

)

Таблица создана по аналогии с таблицей «Клиент».

Создание таблицы «Неисправность» представлена в следующем T-SQL запросе:

-- Хранит в себе типы неисправностей

create table Неисправность(

[ID Неисправности] int primary key not null,

Цена int not null,

Сложность int not null, -- Оценка сложности /10

Тип nchar(50) not null,

[Время работы] time not null

)

Тип данных int позволяет хранить в себе целочисленные данные. В атрибуте «Сложность» хранится сложность ремонта от одного до десяти баллов.

Создание таблицы «Инструмент» представлено в следующем T-SQL зпросе:

-- Хранит в себе набор инструментов

create table Инструмент(

[ID Набора] int primary key not null,

Название nchar(20) not null,

Перечень nchar(200) not null

)

Атрибут «Перечень» хранит в себе до двухсот символов, что позволяет вписать большой перечень инструментов.

Создание таблицы «Заявка» представлено в следующем T-SQL запросе:

create table Заявка(

[ID Заявки] int primary key not null,

[Дата добавления] date not null,

Описание nchar(300),

Статус nchar(20) not null,

Комментарий nchar(100),

[ID Заказчика] int not null,

-- Вторичный ключ на ID клиента

FOREIGN KEY ([ID Заказчика]) REFERENCES Клиент ([ID Клиента]),

[ID Исполнителя] int not null,

-- Вторичный ключ на ID Администратора

FOREIGN KEY ([ID Исполнителя]) REFERENCES Администратор ([ID Администратора]),

[Тип неисправности] int not null,

-- Вторичный ключ на ID Неисправности

FOREIGN KEY ([Тип Неисправности]) REFERENCES Неисправность ([ID Неисправности]),

[Набор инструментов] int not null,

-- Вторичный ключ на ID Неисправности

FOREIGN KEY ([Набор инструментов]) REFERENCES Инструмент ([ID Набора]),

)

Команда foreign key создает вторичный ключ, из соседней таблицы. Атрибутам «Описание» и «Комментарий» предоставлена возможность оставаться пустыми, в случае, если заказчик не будет вводить текст.

Пример диаграммы базы данных представлен на рисунке 2.

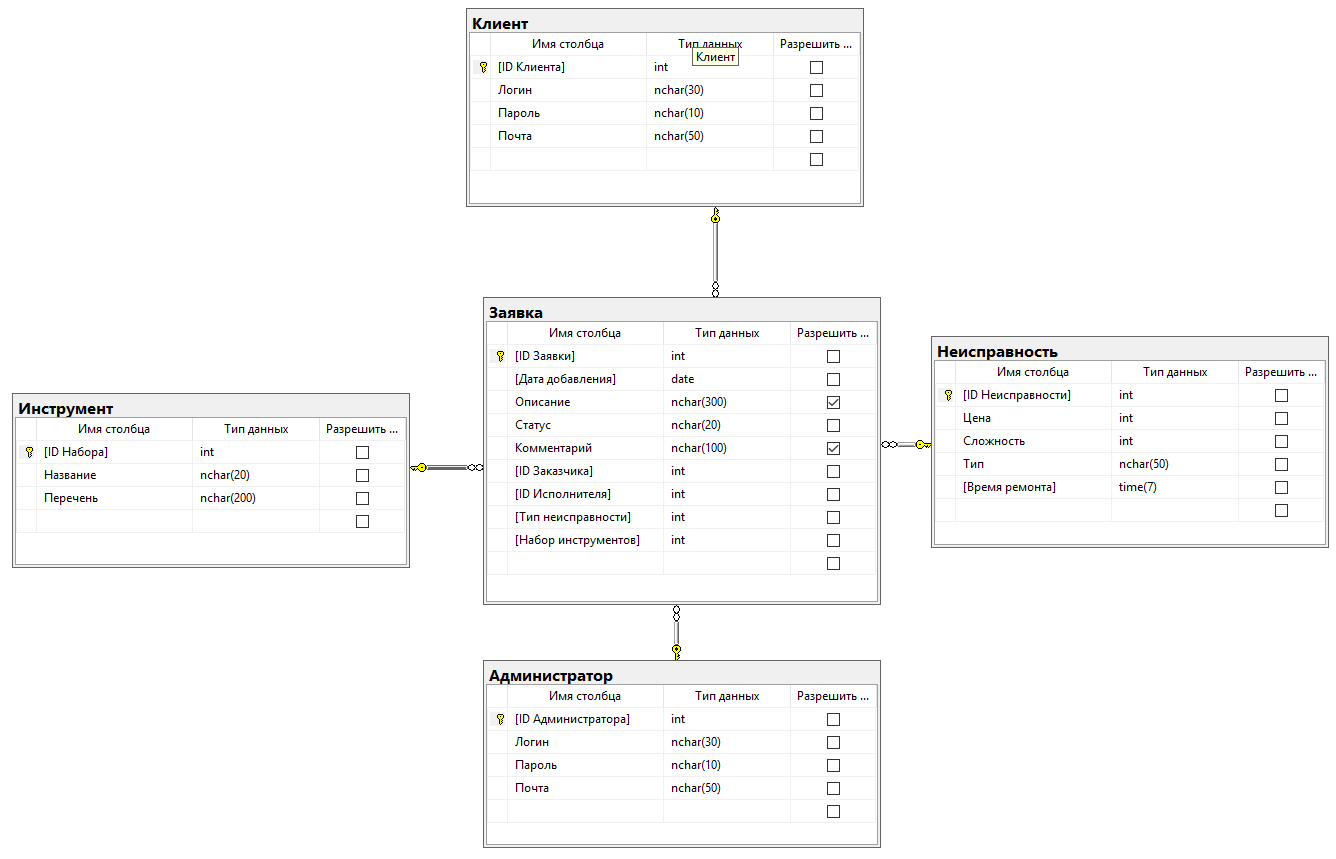


Рис. 2 диаграмма базы данных

# Выбор языка программирования

В качестве языка программирования был выбран C#. Язык позволяет работать с фреймворком WPF, который предназначен для разработки программного обеспечения под персональные компьютеры и ноутбуки. Язык программирования позволяет работать в связке с СУБД MS SQL, в которой располагается база данных.

# Выбор ИСР

В качестве интегрированной среды разработки была выбрана Visual Studio 2022. ИСР располагает инструментарием для работы с фреймворком WPF, а также позволяет произвести безопасное подключение с СУБД MS SQL.

# Интегрирование базы данных в программный код

База данных располагается на локальном SQL Server, который находится по адресу «.\\MSSQLSERVER2022». Подключение происходит с использованием строки подключения, представленной в следующем C# коде:

private const string url = "data source=.\\MSSQLSERVER2022;" +

"Database=RepairService;" +

"User Id=sa;" +

"Password=123;" +

"TrustServerCertificate=True;";

Переменная «url» является константой, чтобы избежать изменений, а также private, для предотвращения несанкционированного доступа из других классов. В строке «Database=RepairService;» указывается имя базы данных, к которой идет подключение. В строке «User Id=sa;» указывается имя администратора базы данных, с чьего аккаунта производится подключение. В строке «Password=123;» указан пароль от аккаунта администратора. Строка «TrustServerCertificate=True;» хранит в себе подтверждение того, что пользователь доверяет подключение к серверу.