Министерство образования Республики Беларусь

Учреждение образования

«Могилевский государственный университет имени А.А. Кулешова»

разработка информационной системы

«фирма vi»

Курсовая работа

по дисциплине «Базы данных»

cтудентки 1 курса специальности 1-40-01-01

«Программное обеспечение информационных технологий»

заочной формы получения образования, группы 5242201

Полины Владимировны Лагута

Научный руководитель

старший преподаватель кафедры программного

обеспечения информационных технологий

Людмила Александровна Мороз

Могилев

2021

ОГЛАВЛЕНИЕ

[**ВВЕДЕНИЕ** 3](#_Toc73548176)

[**Глава 1 ПОСТАНОВКА ЗАДАЧИ** 4](#_Toc73548177)

[1.1 Техническое задание 4](#_Toc73548178)

[1.2 Анализ предметной области 4](#_Toc73548179)

[**ГЛАВА 2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ** 6](#_Toc73548180)

[2.1 Логическая структура базы данных 6](#_Toc73548181)

[2.2 Преобразование ER-диаграммы в схему базы данных 7](#_Toc73548182)

[2.3 Нормализация отношений 7](#_Toc73548183)

[2.4 Создание реляционных отношений 8](#_Toc73548184)

[**Глава 3 РЕАЛИЗАЦИЯ СИСТЕМЫ** 14](#_Toc73548185)

[3.1 Создание таблиц 14](#_Toc73548186)

[3.2 Создание запросов 19](#_Toc73548187)

[**ЗАКЛЮЧЕНИЕ** 23](#_Toc73548188)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 24](#_Toc73548189)

# **ВВЕДЕНИЕ**

В настоящее бурно развиваются информационные технологии и программные средства в сфере бизнеса и торговли. Это обусловливается тем эффектом, который можно получить при их правильном использовании. Современные информационные технологии позволяют организовать работу предприятия с максимальной эффективностью.

Уровень развития компьютерных технологий позволяет создавать программы, которые обладают неограниченными возможностями, при этом обеспечивают большую достоверность и позволяют перейти на качественно новый уровень проектирования.

Целью курсового проекта является разработка информационной системы «Фирма VI».

Ведение базы данных осуществляет сервер базы данных, что в свою очередь позволяет обеспечить независимость обработки данных в базе от программ пользователя. С точки зрения пользователя информационная система должна не только выполнять свои непосредственные функции, но и иметь достаточно простой в использовании интерфейс.

Таким образом, практическая значимость курсового проекта заключается в том, что данная информационная система позволит структурировать достаточно обширные объемы информации, и тем самым существенно облегчит работу и доступ к данным сотрудникам, участвующим в работе фирмы.

Задачи курсового проекта:

- изучить и проанализировать соответствующую литературу;

- провести анализ предметной области;

- разработать базу данных;

- составить запросы;

- протестировать выполнение запросов, выявить все допущенные ошибки и устранить их.

Объект данной работы – база данных торговой фирмы.

Предмет проектирования – система управления реляционными базами данных Microsoft SQL Server.

# **Глава 1 ПОСТАНОВКА ЗАДАЧИ**

## 1.1 Техническое задание

Перечень документов, на основании которых создаётся информационный ресурс:

* задание на курсовую работу;
* правила оформления курсовой работы;
* методическое пособие к курсовому проектированию;
* нормативно правовые акты (в рамках законодательства РБ).

Информационный ресурс передается в виде функционирующего комплекса программных средств в установленные плановые сроки. Вместе с программным продуктом предоставляется в печатной форме вся сопроводительная документация и требуемые сопутствующие материалы.

Систему предполагается создать для улучшения качества обслуживания клиентов торговой фирмы, позволит уменьшить вероятность утери информации.

Предмет проектирования – система управления реляционными базами данных Microsoft SQL Server.

При реализации курсового проекта будут использованы:

- язык структурированных запросов – SQL;

- Microsoft SQL Server 2018;

## 1.2 Анализ предметной области

Предметная область данного курсового проекта заключается в создании информационной системы «Фирма VI». Преимущества данной системы заключаются в упрощении просмотра и добавления данных. С помощью данной системы пользователям предоставляется доступ к данным, имеющимся в базе данных. В работе для реализации качественной информационной системы использованы разные подходы к разработке и моделированию информационной системы.

Основным назначением информационных ресурсов является оперативное обеспечение пользователя информацией путем реализации взаимодействия пользователя с системой, через запросы и ответы на них со стороны информационной системы. Информация представляется в информационном ресурсе в форме данных. Совокупность этих выделенных для информационных ресурсов данных, связей между ними и операций над ними образует информационную и функциональную модели предметной области, описывающие ее состояние с определенной точностью.

Основными достоинствами моделей исследуемых объектов или явлений, реализуемых с помощью ЭВМ, является гибкость и вариативность, возможность управления их поведением, реализация интерактивного режима работы.

Систему предполагается использовать на рабочих местах в фирмах для увеличения скорости обслуживания клиентов. Система позволит облегчить процесс поиска товаров, т.к. он будет вестись автоматизировано. При её применении станет легче подбирать товары, следить за историей заказов, проверять наличие товаров на складах.

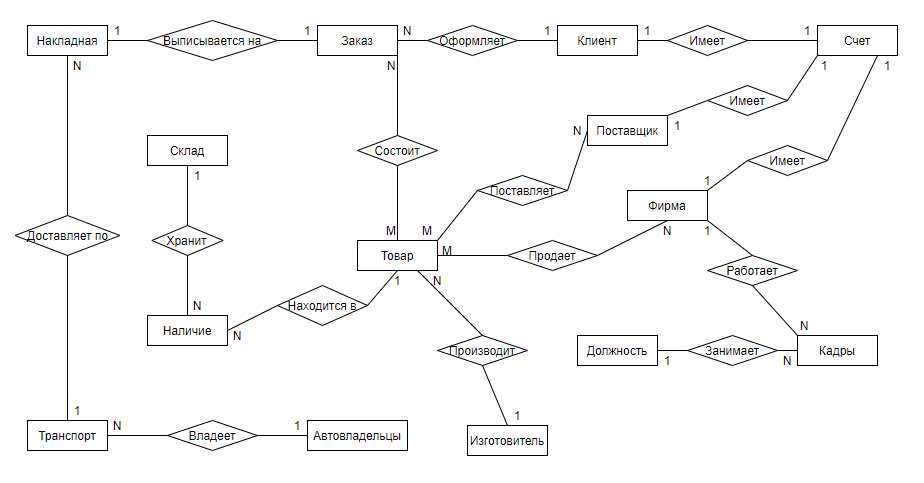
Систему предполагается создать для улучшения качества обслуживания клиентов, обслуживания большего числа клиентов, а также корректного хранения данных. Система позволит уменьшить вероятность утери информации.

# **ГЛАВА 2 ПРОЕКТИРОВАНИЕ СИСТЕМЫ**

## 2.1 Логическая структура базы данных

Исходя из анализа предметной области, набора требований и возможностей реляционной модели данных разработана ER-диаграмма данной информационной системы. Её структура представлена на рисунке 2.1.1. Она полностью удовлетворяет требованиям третьей нормальной формы.

Рисунок 2.1.1 ER-диаграмма базы данных «Фирма VI»



На диаграмме представлены следующие таблицы:

- «Накладная» – для фиксации информации накладных, выписанных на доставку заказа;

- «Заказ» – для хранения информации об оформленных заказах, состоящих из товаров;

- «Клиент» – запись информации о клиентах, которые пользуются услугами фирм и оформляют заказы;

- «Счет» – информация о счетах в банках, которые имеют клиенты, поставщики и фирмы;

- «Склад» - информация о складах;

- «Товар» хранит информацию о товарах;

- «Наличие» – информация о товарах, которые имеются в наличии;

- «Поставщик» хранит информацию о поставщиках товаров;

- «Кадры» - запись информации о работниках фирм;

- «Должности» хранит информацию о должностях фирм и соответствующих окладах;

- «Фирма» содержит информацию о фирмах-партнерах;

- «Изготовитель» - информация об изготовителях товаров;

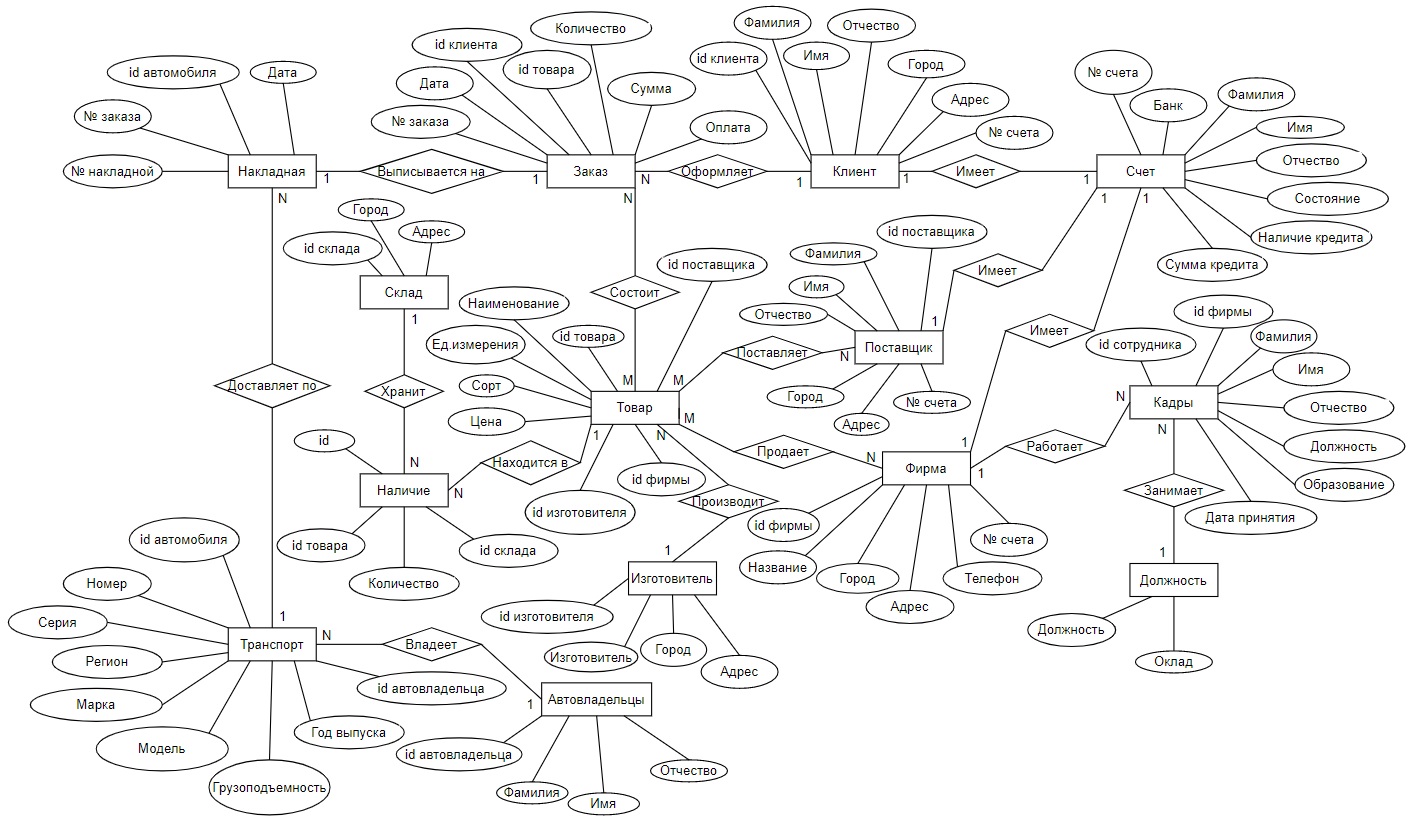
- «Транспорт» - перечень и описание транспорта, используемого в фирмах;

- «Автовладельцы» - информация о владельцах транспорта.

## 2.2 Преобразование ER-диаграммы в схему базы данных

База данных создается на основании схемы базы данных. Для преобразования ER-диаграммы в схему базы данных приведем уточненную ER-диаграмму, содержащую атрибуты сущностей (рисунок 2.2.1).

Рисунок 2.2.1 Уточненная ER-диаграмма фирмы



Преобразование ER–диаграммы в схему базы данных выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения базы данных.

## 2.3 Нормализация отношений

Нормализация баз данных заключается в приведении структуры хранения данных к нормальным формам.

Основным правилом первой нормальной формы является необходимость неделимости значения в каждом поле (столбце) строки – атомарность значений. Делим атрибуты Ф.И.О. на три отдельных атрибута: Фамилия, Имя и Отчество. Также атрибуты Адрес разделим на два атрибута: Город и Адрес.

Условиями второй нормальной формы являются: соответствие таблицы первой нормальной форме, таблицы должны иметь ключи и все не ключевые столбцы должны зависеть от полного ключа.

Требование третьей нормальной формы заключается в том, чтобы в таблицах отсутствовала транзитивная зависимость. Транзитивная зависимость – это когда не ключевые столбцы зависят от значений других не ключевых столбцов. Поле Оклад в атрибуте Кадры не зависит от первичного ключа Номер сотрудника, поэтому выносим Должность и оклад в отдельную таблицу.

## 2.4 Создание реляционных отношений

Реляционная модель данных — это способ рассмотрения данных, то есть предписание для способа представления данных (посредством таблиц) и для способа работы с таким представлением (посредством операторов). Она связана с тремя аспектами данных: структурой (объекты), целостностью и обработкой данных (операторы).

Цели создания реляционной модели данных:

* обеспечение более высокой степени независимости от данных;
* создание прочного фундамента для решения семантических вопросов и проблем непротиворечивости и избыточности данных;
* расширение языков управления данными за счёт включения операций над множествами.

Каждое реляционное отношение соответствует одной сущности и в него вносятся все атрибуты этой сущности. Для каждого отношения определяются первичный и внешние ключи. В том случае, если базовое отношение не имеет потенциальных ключей, вводится суррогатный первичный ключ, который не несет смысловой нагрузки и служит только для идентификации записей.

Для каждого отношения указываем атрибуты с их внутренним названием, типом и длиной. Типы данных обозначаются: N – числовой, C – символьный тип фиксированной длины, V – символьный тип переменной длины, M – денежный тип данных, D – дата, B – логический тип данных (принимает значение True или False).

Потенциальным составным ключом отношения Автовладельцы являются атрибуты Фамилия, Имя и Отчество, но он занимает много места и не удобен, поэтому введем суррогатный первичный ключ id\_автовладельца. Схема отношения Автовладельцы представлена в таблице 2.4.1.

Таблица 2.4.1 Схема отношения Автовладельцы

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Код автовладельца | id\_автовладельца | N | Суррогатный первичный ключ, уникальное значение |
| Фамилия автовладельца | Фамилия | V(20) | Обязательное поле |
| Имя автовладельца | Имя | V(20) | Обязательное поле |
| Отчество автовладельца | Отчество | V(30) | Обязательное поле |

Потенциальным ключом отношения Должности является атрибут Должность, вводим его в качестве первичного ключа. Схема отношения Должности представлена в таблице 2.4.2.

Таблица 2.4.2 Схема отношения Должности

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Должность работника | Должность | V(20) | Первичный ключ |
| Оклад | Фамилия | M | Обязательное поле |

Потенциальным ключом отношения Заказ является атрибут Номер заказа, он уникален, поэтому мы выбираем его в качестве первичного ключа. Схема отношения заказы представлена в таблице 2.4.3.

Таблица 2.4.3 Схема отношения Заказы

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Номер заказа | №\_заказа | N | Первичный ключ, уникальное значение |
| Дата заказа | Дата | D | Обязательное поле |
| Номер клиента | id\_клиента | N | Обязательное поле |
| Номер товара | id\_товара | N | Обязательное поле |
| Количество товаров | Количество | N | Обязательное поле |
| Итоговая сумма | Сумма | M | Обязательное поле |
| Подтверждение оплаты | Оплата | B | Обязательное поле |

Потенциальным ключом отношения Изготовитель является атрибут Наименование изготовителя, но он не удобен, поэтому мы вводим суррогатный первичный ключ Код изготовителя. Схема отношения Изготовитель представлена в таблице 2.4.5.

Таблица 2.4.5 Схема отношения Изготовитель

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Код изготовителя | id\_изготовителя | N | Суррогатный первичный ключ, уникальное значение |
| Наименование изготовителя | Изготовитель | V(20) | Обязательное поле |
| Город | Город | V(20) | Обязательное поле |
| Юридический адрес | Адрес | V(30) | Обязательное поле |

Потенциальным составным ключом отношения Кадры являются атрибуты Фамилия, Имя, Отчество и Дата принятия на работы, но он занимает много места и неудобен в использовании, поэтому вводим суррогатный первичный ключ Код сотрудника. Схема отношения Кадры представлена в таблице 2.4.6.

Таблица 2.4.6 Схема отношения Кадры

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Код сотрудника | id\_сотрудника | N | Суррогатный первичный ключ, уникальное значение |
| Код фирмы | id\_фирмы | N | Обязательное поле |
| Фамилия сотрудника | Фамилия | V(20) | Обязательное поле |
| Имя сотрудника | Имя | V(20) | Обязательное поле |
| Отчество сотрудника | Отчество | V(20) | Обязательное поле |
| Должность сотрудника | Должность | V(20) | Обязательное поле |
| Образование сотрудника | Образование | V(20) | Обязательное поле |
| Дата принятия на работу | Дата\_принятия | D | Обязательное поле |

Потенциальным ключом отношения Клиент является атрибут Код клиента, он уникален, поэтому мы выбираем его в качестве первичного ключа. Схема отношения Клиент представлена в таблице 2.4.7.

Таблица 2.4.7 Схема отношения Клиент

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Код сотрудника | id\_клиента | N | Первичный ключ, уникальное значение |
| Фамилия клиента | Фамилия | V(20) | Обязательное поле |
| Имя клиента | Имя | V(20) | Обязательное поле |
| Отчество клиента | Отчество | V(20) | Обязательное поле |
| Город проживания клиента | Город | V(20) | Необязательное поле |
| Адрес проживания клиента | Адрес | V(30) | Необязательное поле |
| Номер счета в банке | №\_счета | N | Необязательное поле |

Потенциальным ключом отношения Накладная является атрибут Номер накладной, каждая накладная имеет уникальный номер, поэтому выбираем его в качестве первичного ключа. Схема отношения Накладная представлена в таблице 2.4.8.

Таблица 2.4.8 Схема отношения Накладная

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Номер накладной | №\_накладной | N | Первичный ключ, уникальное значение |
| Номер заказа | №\_заказа | N | Обязательное поле |
| Код автомобиля | id\_автомобиля | N | Обязательное поле |
| Дата накладной | Дата | D | Обязательное поле |

Потенциального ключа в отношении Наличие нет, поэтому задаем суррогатный первичный ключ Идентификатор записи. Схема отношения Наличие представлена в таблице 2.4.9.

Таблица 2.4.9 Схема отношения Наличие

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Идентификатор | id | N | Первичный ключ, уникальное значение |
| Код товара | id\_товара | N | Обязательное поле |
| Количество | Количество | N | Обязательное поле |
| Код склада | id\_склада | N | Обязательное поле |

Потенциальным ключом отношения Поставщик является атрибут Номер поставщика, он уникален, поэтому мы выбираем его в качестве первичного ключа. Схема отношения Поставщик представлена в таблице 2.4.10.

Таблица 2.4.10 Схема отношения Поставщик

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Номер поставщика | id\_поставщика | N | Первичный ключ, уникальное значение |
| Фамилия поставщика | Фамилия | V(20) | Обязательное поле |
| Имя поставщика | Имя | V(20) | Обязательное поле |
| Отчество поставщика | Отчество | V(20) | Обязательное поле |
| Город | Город | V(20) | Необязательное поле |
| Адрес | Адрес | V(30) | Необязательное поле |
| Номер счета в банке | №\_счета | N | Обязательное поле |

Потенциальным ключом отношения Склад является атрибут Номер склада, он подходит, поэтому мы выбираем его в качестве первичного ключа. Атрибуты Наименование товара, Цена товара, Единицы измерения, количество могут быть получены из отношений Товар и Наличие. Схема отношения Склад представлена в таблице 2.4.11.

Таблица 2.4.11 Схема отношения Склад

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Номер склада | id\_склада | N | Первичный ключ, уникальное значение |
| Город | Город | V(20) | Обязательное поле |
| Адрес | Адрес | V(30) | Обязательное поле |

Потенциальным ключом отношения Счет является атрибут Номер счета, он уникален для каждого счета, поэтому мы выбираем его в качестве первичного ключа. Схема отношения Счет представлена в таблице 2.4.12.

Таблица 2.4.12 Схема отношения Счет

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Номер счета в банке | №\_счета | N | Первичный ключ, уникальное значение |
| Наименование банка | Банк | V(30) | Обязательное поле |
| Фамилия владельца счета | Фамилия | V(20) | Обязательное поле |
| Имя владельца счета | Имя | V(20) | Обязательное поле |
| Отчество владельца счета | Отчество | V(20) | Обязательное поле |
| Состояние счета | Состояние | V(10) | Обязательное поле |
| Наличие кредита | Наличие\_кредита | B | Обязательное поле |
| Сумма кредита | Сумма\_кредита | M | Необязательное поле |

Потенциальным составным ключом отношения Товар являются атрибуты Наименование товара, Изготовитель и Поставщик, но он не удобен, поэтому мы вводим суррогатный первичный ключ Код товара и выбираем его в качестве первичного ключа. Схема отношения Товар представлена в таблице 2.4.13.

Таблица 2.4.13 Схема отношения Товар

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Код товара | id\_товара | N | Суррогатный первичный ключ, уникальное значение |
| Наименование товара | Наименование | V(50) | Обязательное поле |
| Единицы измерения товара | Единицы\_измерения | V(10) | Необязательное поле |
| Сорт товара | Сорт | N | Необязательное поле |
| Цена товара | Цена | M | Обязательное поле |
| Код изготовителя товара | id\_изготовителя | N | Обязательное поле |
| Код фирмы | id\_фирмы | N | Обязательное поле |
| Код поставщика товара | id\_поставщика | N | Обязательное поле |

Потенциальным составным ключом отношения Транспорт являются атрибуты Номер, Серия, Регион, но он не удобен, поэтому мы вводим суррогатный первичный ключ Код автомобиля и выбираем его в качестве первичного ключа. Схема отношения Транспорт представлена в таблице 2.4.14.

Таблица 2.4.14 Схема отношения Транспорт

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Код автомобиля | id\_автомобиля | N | Суррогатный первичный ключ, уникальное значение |
| Регистрационный знак (номер) автомобиля | Номер | С(4) | Обязательное поле |
| Серия номера автомобиля | Серия | С(2) | Обязательное поле |
| Регион номера автомобиля | Регион | N | Обязательное поле |
| Марка автомобиля | Марка | V(20) | Необязательное поле |

Продолжение таблицы 2.4.14

|  |  |  |  |
| --- | --- | --- | --- |
| Модель автомобиля | Модель | V(20) | Необязательное поле |
| Грузоподъемность автомобиля | Грузоподъемность | N | Необязательное поле |
| Год выпуска автомобиля | Год\_выпуска | N | Необязательное поле |
| Код владельца автомобиля | id\_автовладельца | N | Обязательное поле |

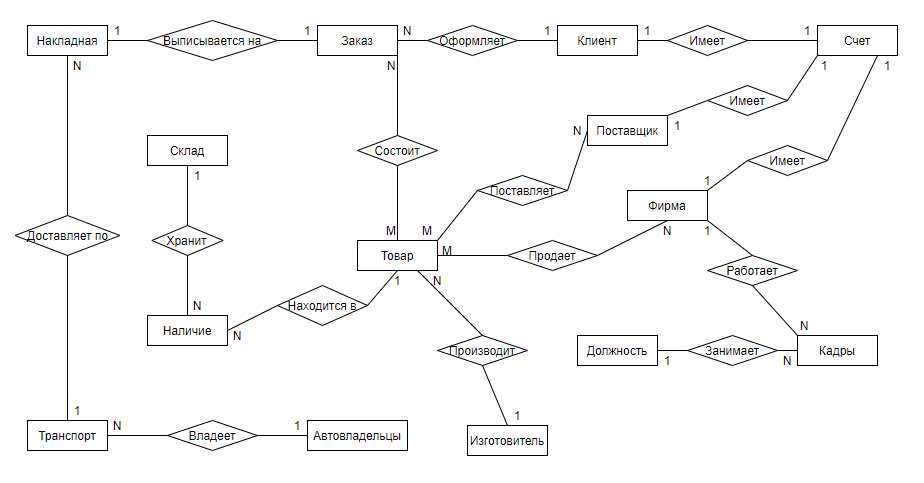
Потенциальным ключом отношения Фирма является атрибут Код фирмы, он уникален для каждой фирмы, поэтому мы выбираем его в качестве первичного ключа. Схема отношения Фирма представлена в таблице 2.4.15.

Таблица 2.4.15 Схема отношения Фирма

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечание |
| Код фирмы | id\_фирмы | N | Первичный ключ, уникальное значение |
| Наименование фирмы | Название | V(20) | Обязательное поле |
| Город | Город | V(20) | Необязательное поле |
| Адрес | Адрес | V(30) | Необязательное поле |
| Номер телефона фирмы | Телефон | С(13) | Обязательное поле |
| Номер счета в банке | №\_счета | N | Необязательное поле |

Окончательная схема базы данных представлена на рисунке 2.4.1.

Рисунок 2.4.1 Схема базы данных «Фирма VI»



# **Глава 3 РЕАЛИЗАЦИЯ СИСТЕМЫ**

## 3.1 Создание таблиц

\*Отношение Автовладельцы

CREATE TABLE [dbo].[Автовладельцы](

[id\_автовладельца] [int] IDENTITY(1,1) NOT NULL,

[Фамилия] [varchar](20) NOT NULL,

[Имя] [varchar](20) NOT NULL,

[Отчество] [varchar](20) NOT NULL,

CONSTRAINT [PK\_Автовладельцы] PRIMARY KEY CLUSTERED

( [id\_автовладельца] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Должности

CREATE TABLE [dbo].[Должности](

[Должность] [varchar](20) NOT NULL,

[Оклад] [money] NOT NULL,

CONSTRAINT [PK\_Должности] PRIMARY KEY CLUSTERED

( [Должность] ASC

)WITH (PAD\_INDEX=OFF, STATISTICS\_NORECOMPUTE=OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Заказ

CREATE TABLE [dbo].[Заказ](

[№\_заказа] [int] IDENTITY(1,1) NOT NULL,

[Дата] [date] NOT NULL,

[id\_клиента] [int] NOT NULL,

[id\_товара] [int] NOT NULL,

[Количество] [int] NOT NULL,

[Сумма] [money] NOT NULL,

[Оплата] [bit] NOT NULL,

CONSTRAINT [PK\_Заказ] PRIMARY KEY CLUSTERED

( [№\_заказа] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Изготовитель

CREATE TABLE [dbo].[Изготовитель](

[id\_изготовителя] [int] IDENTITY(1,1) NOT NULL,

[Изготовитель] [varchar](20) NOT NULL,

[Город] [varchar](20) NOT NULL,

[Адрес] [varchar](30) NOT NULL,

CONSTRAINT [PK\_Изготовитель] PRIMARY KEY CLUSTERED

( [id\_изготовителя] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Кадры

CREATE TABLE [dbo].[Кадры](

[id\_сотрудника] [int] IDENTITY(1,1) NOT NULL,

[id\_фирмы] [int] NOT NULL,

[Фамилия] [varchar](20) NOT NULL,

[Имя] [varchar](20) NOT NULL,

[Отчество] [varchar](20) NOT NULL,

[Должность] [varchar](20) NOT NULL,

[Образование] [varchar](20) NOT NULL,

[Дата\_принятия] [date] NOT NULL,

CONSTRAINT [PK\_Кадры\_1] PRIMARY KEY CLUSTERED

( [id\_сотрудника] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Клиент

CREATE TABLE [dbo].[Клиент](

[id\_клиента] [int] IDENTITY(1,1) NOT NULL,

[Фамилия] [varchar](20) NOT NULL,

[Имя] [varchar](20) NOT NULL,

[Отчество] [varchar](20) NOT NULL,

[Город] [varchar](20) NULL,

[Адрес] [varchar](30) NULL,

[№\_счета] [int] NULL,

CONSTRAINT [PK\_Клиент\_1] PRIMARY KEY CLUSTERED

( [id\_клиента] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Накладная

CREATE TABLE [dbo].[Накладная](

[№\_накладной] [int] NOT NULL,

[№\_заказа] [int] NOT NULL,

[id\_автомобиля] [int] NOT NULL,

[Дата] [date] NOT NULL,

CONSTRAINT [PK\_Накладная] PRIMARY KEY CLUSTERED

(

[№\_накладной] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Наличие

CREATE TABLE [dbo].[Наличие](

[id] [int] IDENTITY(1,1) NOT NULL,

[id\_товара] [int] NOT NULL,

[Количество] [int] NOT NULL,

[id\_склада] [int] NOT NULL,

CONSTRAINT [PK\_Наличие] PRIMARY KEY CLUSTERED

( [id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Поставщик

CREATE TABLE [dbo].[Поставщик](

[id\_поставщика] [int] IDENTITY(1,1) NOT NULL,

[Фамилия] [varchar](20) NOT NULL,

[Имя] [varchar](20) NOT NULL,

[Отчество] [varchar](20) NOT NULL,

[Город] [varchar](20) NULL,

[Адрес] [varchar](30) NULL,

[№\_счета] [int] NOT NULL,

CONSTRAINT [PK\_Поставщик] PRIMARY KEY CLUSTERED

( [id\_поставщика] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Склад

CREATE TABLE [dbo].[Склад](

[id\_склада] [int] IDENTITY(1,1) NOT NULL,

[Город] [varchar](20) NOT NULL,

[Адрес] [varchar](30) NOT NULL,

CONSTRAINT [PK\_Склад] PRIMARY KEY CLUSTERED

( [id\_склада] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Счет

CREATE TABLE [dbo].[Счет](

[№\_счета] [int] NOT NULL,

[Банк] [varchar](30) NOT NULL,

[Фамилия] [varchar](20) NOT NULL,

[Имя] [varchar](20) NOT NULL,

[Отчество] [varchar](20) NOT NULL,

[Состояние] [varchar](10) NOT NULL,

[Наличие\_кредита] [bit] NOT NULL,

[Сумма\_кредита] [money] NULL,

CONSTRAINT [PK\_Счет] PRIMARY KEY CLUSTERED

( [№\_счета] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Товар

CREATE TABLE [dbo].[Товар](

[id\_товара] [int] NOT NULL,

[Наименование\_товара] [varchar](50) NOT NULL,

[Единицы\_измерения] [varchar](10) NULL,

[Сорт] [int] NULL,

[Цена] [money] NOT NULL,

[id\_изготовителя] [int] NOT NULL,

[id\_фирмы] [int] NOT NULL,

[id\_поставщика] [int] NOT NULL,

CONSTRAINT [PK\_Товар] PRIMARY KEY CLUSTERED

(

[id\_товара] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Транспорт

CREATE TABLE [dbo].[Транспорт](

[id\_автомобиля] [int] IDENTITY(1,1) NOT NULL,

[Номер] [varchar](4) NOT NULL,

[Серия] [varchar](2) NOT NULL,

[Регион] [int] NOT NULL,

[Марка] [varchar](20) NULL,

[Модель] [varchar](20) NULL,

[Грузоподъемность\_кг] [int] NULL,

[Год\_выпуска] [int] NULL,

[id\_автовладельца] [int] NOT NULL,

CONSTRAINT [PK\_Транспорт\_1] PRIMARY KEY CLUSTERED

(

[id\_автомобиля] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

\*Отношение Фирма

CREATE TABLE [dbo].[Фирма](

[id\_фирмы] [int] IDENTITY(1,1) NOT NULL,

[Название] [varchar](20) NOT NULL,

[Город] [varchar](20) NULL,

[Адрес] [varchar](30) NULL,

[Телефон] [varchar](13) NOT NULL,

[№\_счета] [int] NULL,

CONSTRAINT [PK\_Фирма\_1] PRIMARY KEY CLUSTERED

(

[id\_фирмы] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

## 3.2 Создание запросов

***Задание 1***. Перечислить номера накладных, выписанных на данный товар.

Текст запроса:

SELECT

Накладная.[№\_накладной],

Товар.Наименование\_товара

INTO Zapros\_A

FROM Накладная

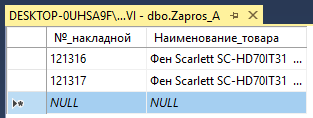
JOIN Заказ ON Накладная.[№\_заказа]=Заказ.[№\_заказа]

JOIN Товар ON Заказ.id\_товара = Товар.id\_товара

WHERE Товар.Наименование\_товара = 'Фен Scarlett SC-HD70IT31'

Для примера выполнения запроса взяли существующее наименование товара Фен Scarlett SC-HD70IT31 из таблицы Товар. Результат выполнения запроса сохраняется в таблицу Zapros\_A. Пример представлен на рисунке 3.2.1.

Рисунок 3.2.1 Результат запроса 1



***Задание 2.*** Кто из клиентов заказал данный товар?

Текст запроса:

SELECT

Клиент.Фамилия,

Клиент.Имя,

Клиент.Отчество,

Товар.Наименование\_товара

INTO Zapros\_B

FROM Заказ

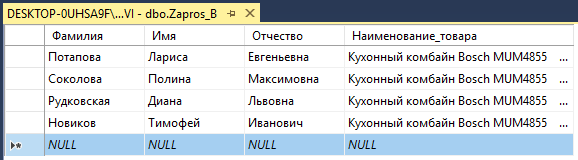
JOIN Клиент ON Клиент.id\_клиента=Заказ.id\_клиента

JOIN Товар ON Заказ.id\_товара = Товар.id\_товара

WHERE Товар.Наименование\_товара = 'Кухонный комбайн Bosch MUM4855'

Для примера выполнения запроса взяли существующее наименование товара Кухонный комбайн Bosch MUM4855 из таблицы Товар. Результат выполнения запроса сохраняется в отдельную таблицу Zapros\_B. Пример представлен на рисунке 3.2.2.

Рисунок 3.2.2 Результат запроса 2



***Задание 3***. Перечислить номера заказов и ФИО клиентов, которые заказали данный товар, в данном количестве.

Текст запроса:

SELECT

Заказ.[№\_заказа],

Товар.Наименование\_товара,

Заказ.Количество,

Клиент.Фамилия,

Клиент.Имя,

Клиент.Отчество

INTO Zapros\_C

FROM Заказ

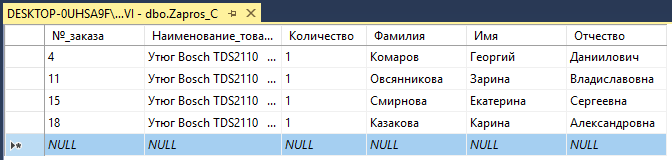
JOIN Клиент ON Клиент.id\_клиента=Заказ.id\_клиента

JOIN Товар ON Заказ.id\_товара = Товар.id\_товара

WHERE Товар.Наименование\_товара = 'Утюг Bosch TDS2110' and Заказ.Количество='1'

Для примера выполнения запроса взяли существующие значения: наименование товара Утюг Bosch TDS2110 из таблицы Товар и количество 1 из таблицы Заказ. Результат выполнения запроса сохраняется в отдельную таблицу Zapros\_C. Пример представлен на рисунке 3.2.3.

Рисунок 3.2.3 Результат запроса 3



***Задание 4***. Кто из клиентов (ФИО) получал данный товар и по каким номерам накладных?

Текст запроса:

SELECT

Клиент.Фамилия,

Клиент.Имя,

Клиент.Отчество,

Товар.Наименование\_товара,

Накладная.[№\_накладной]

INTO Zapros\_D

FROM Заказ

JOIN Клиент ON Клиент.id\_клиента=Заказ.id\_клиента

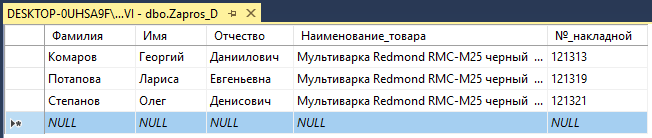
JOIN Товар ON Заказ.id\_товара = Товар.id\_товара

JOIN Накладная ON Заказ.[№\_заказа] = Накладная.[№\_заказа]

WHERE Товар.Наименование\_товара = 'Мультиварка Redmond RMC-M25 черный'

Для примера выполнения запроса взяли существующее значение Наименование товара Мультиварка Redmond RMC-M25. Результат выполнения запроса сохраняется в отдельную таблицу Zapros\_D. Пример представлен на рисунке 3.2.4.

Рисунок 3.2.4 Результат запроса 4



***Задание 5.*** Перечислить номера складов, на которых имеется данный товар, данного изготовителя.

Запрос:

SELECT

Товар.Наименование\_товара, Наличие.Количество, Наличие.id\_склада, Изготовитель.Изготовитель

INTO Zapros\_E

FROM Товар

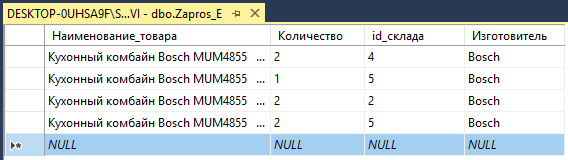
JOIN Наличие ON Товар.id\_товара = Наличие.id\_товара

JOIN Изготовитель ON Товар.id\_изготовителя = Изготовитель.id\_изготовителя

WHERE Товар.Наименование\_товара = 'Кухонный комбайн Bosch MUM4855' and Изготовитель.Изготовитель = 'Bosch'

Для примера выполнения запроса взяли существующие значения: наименование товара 'Кухонный комбайн Bosch MUM4855 из таблицы Товар и изготовитель Bosch из таблицы Изготовители. Результат выполнения запроса сохраняется в отдельную таблицу Zapros\_E. Пример представлен на рисунке 3.2.5.

Рисунок 3.2.5 Результат запроса 5



# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы была достигнута цель создания базы данных в заданной предметной области и выполнены следующие задачи:

* провести анализ предметной области;
* составить рабочий план создания проекта;
* разработать, отладить и протестировать рабочий проект;
* разработать сопроводительную документацию к рабочему проекту.

Создана база данных, позволяющая пользователю:

* просматривать данные;
* редактировать данные;
* просматривать результат запросов.

В процессе разработки была использована нормализованная реляционная модель. Проведение нормализации позволило спроектировать базу данных, в которой нет избыточных (ненужных) данных и противоречий. Была обеспечена целостность данных, которая способствовала адекватности отражаемых сведений.

При выполнении курсовой работы были решены задачи, поставленные в начале работы. При этом были учтены все требования, выдвинутые в начале выполнения данной работы.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Дейт, К. Введение в системы баз данных / К. Дейт. – 8-е изд. – Москва:Санкт-Петербург : Киев : Вильямс, 2017. – 1328 с.
2. Куликов С.С. Реляционные базы данных в примерах : практическое пособие для программистов и тестировщи- ков / С. С. Куликов. — Минск: Четыре четверти, 2020. — 424 с.
3. Коннолли Т., Бегг К. Базы данных: проектирование, реализация, сопровождение. Теория и практика. – 3-е изд.: Пер. с англ.: Уч. пос. – М.: Изд. дом "Вильямс", 2003. – 1440 с
4. Грофф, Дж. Р.SQL : Полное руководство / Дж. Р. Грофф, П. Н. Вайнберг ; под ред. В. Р. Гинзбург. – 2-е изд., перераб. и доп. – Киев : BHV: Ирина, 2001. – 816 с
5. MySQL 8 для больших данных / пер. с англ. А.В. Логунова / Чаллавала Ш., Лакхатария Дж., Мехта Ч., Патель К. - Москва: ДМК Пресс, 2018. – 226 с

Текст программы: