

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение   
высшего образования

**«Дальневосточный федеральный университет»**

(ДВФУ)

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**

**Департамент программной инженерии и искусственного интеллекта**

ШЕВЕЛЕВ РОМАН ВАДИМОВИЧ

РАЗРАБОТКА БАЗЫ ДАННЫХ В ПРЕДМЕТНОЙ ОБЛАСТИ «АВТОЗАПРАВКИ»

**КУРСОВАЯ РАБОТА**

по дисциплине «Проектирование и разработка баз данных»

по образовательной программе подготовки бакалавров по направлению

09.03.04 - Программная инженерия

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |  |  | Студент гр. Б9121-09.03.04(5) | | |
|  |  |  | | Шевелев Р. В. |
|  | | | | | | |  |  | (подпись) | | (ФИО) |
|  | | | | | | | | | | | |
| Защищен с оценкой | | | | | | |  |  | Руководитель | | |
|  | | | | | | |  |  | Ассистент ДПИиИИ | | |
|  | | | |  |  | |  |  | учёная степень, должность |  | Иваненко Ю. С. |
| (подпись) | | | |  | (Фамилия И.О.) | |  |  | (подпись) |  | (ФИО) |
| « |  | » |  | | | 2024 г. |  |  |  | | |
|  | | | | | | | | | | | |

г. Владивосток

2024

**Оглавление**

[Введение 3](#_Toc156225666)

[1 Анализ предметной области 4](#_Toc156225667)

[2 Инфологическое проектирование 7](#_Toc156225668)

[3 Даталогическое проектирование 30](#_Toc156225669)

[4 Тестирование 58](#_Toc156225670)

[Заключение 69](#_Toc156225671)

[Список литературы 70](#_Toc156225672)

**Введение**

Сеть заправочных автостанций, предоставляющих услуги по реализации автомобильного топлива, сталкивается с необходимостью эффективного учета продаж и анализа спроса на различные виды топлива. Такая информация помогает предприятию оптимизировать свою деятельность, принимать решения и прогнозировать изменения в спросе. Поэтому требуется разработать базу данных, которая позволит автоматизировать процессы учета продаж и анализа спроса на автомобильное топливо.

В данной работе объектом исследования является конкретная фирма-поставщик автомобильного топлива со своей сетью заправочных автостанций.

Предметом исследования является возможность автоматизации процессов учета продаж автомобильного топлива, а также анализа спроса на различные виды топлива и динамики его изменения.

Цель работы: спроектировать и реализовать базу данных для учета продаж и анализа спроса на топливо в динамике и других связанных процессов.

Задачи:

1. Провести анализ предметной области;
2. Построить инфологическую модель;
3. Создать логическую модель в рамках конкретной системы управления базами данных (провести даталогическое проектирование);
4. Реализовать базу данных;
5. Провести комплексное тестирование реализованной базы данных;

Разработка базы данных для автозаправочных станций позволит автоматизировать процессы учета продаж и анализа спроса на топливо. Это поможет предприятию принимать эффективные управленческие решения и оказывать услуги, отвечающие потребностям клиентов.

**1 Анализ предметной области**

Автозаправочная станция осуществляет продажу топлива клиентам, ведет учет проданных объемов и имеющихся запасов, а также зарегистрированных клиентов. Автозаправка предоставляет услугу заправки автомобилей различных видов топливом.

Пример автозаправки:

|  |
| --- |
| Код автозаправки: 103  Название фирмы-поставщика: Лукойл  Адрес: г. Владивосток, ул. Уборевича, д. 28 |

Пример учета ежедневной продажи:

|  |
| --- |
| Карт-счёт клиента: 766092  Дата продажи: 29.12.2023 23:42  Код автозаправки: 1006  Код топлива: 110  Количество: 40,93 |

Каждый вид топлива на автозаправке представлен определенными характеристиками: наименование, указанное поставщиком, цена за единицу топлива, определяемая фирмой-поставщиком заправки, а также единица измерения самого топлива.

Пример топлива:

|  |
| --- |
| Код топлива: 110  Вид топлива: АИ-92  Единица измерения: ЛИТР  Цена (руб): 49,84 |

Автозаправка, получив информацию о новых видах топлива, вносит их в список доступных видов топлива на конкретной автозаправке. В начале каждого месяца работник заправки закупает топливо согласно списку от фирмы-поставщика по определенной цене.

Пример фирмы-поставщика:

|  |
| --- |
| Название фирмы: Лукойл  Адрес: г. Владивосток, ул. Красного Знамени, дом 37, к. 3, офис 228  Номер телефона: 79999999999 |

Клиент, желающий заправиться, может зарегистрироваться на автозаправке для удобства дальнейших покупок. Он предоставляет информацию о себе: номер карт-счета, ФИО, адрес и контактный телефон сотруднику заправки.

Пример клиента:

|  |
| --- |
| Номер карточного счёта: 768903  ФИО: Иванов Иван Иванович  Адрес: г. Владивосток, ул. Уборевича, д. 28, кв. 4  Номер телефона: 79949386702 |

Работник заправки подтверждает получение информации, оформляет ее и сохраняет в системе. Регистрация возможна только для лиц старше 18 лет, иначе работник заправки не может заверить данные клиента.

Изменения в информации о клиенте могут быть внесены по просьбе клиента. Вся информация о клиенте сохраняется даже после окончания его посещений заправки. Каждый клиент может иметь только один карточный счёт.

В данной предметной области можно выделить следующие задачи для каждого пользователя:

Таблица 1 – Функциональность пользователей

|  |  |
| --- | --- |
| **Пользователь** | **Задачи** |
| Клиент | Просмотр информации о своем карт-счете, Ф.И.О., адресе, телефоне и о совершённых покупках в сети автозаправочных станций. |

*Окончание таблицы 1*

|  |  |
| --- | --- |
| Фирма-поставщик | Добавление, просмотр, редактирование и удаление информации о видах топлива (код, вид единица измерения, цена);  Просмотр информации:  о своих автозаправках (код, название, адрес);  о продаже топлива клиентам (дата продажи, карт-счет клиента, код автозаправки, код топлива, количество) за выбранный период. |
| Администратор | Добавление, просмотр, редактирование и удаление информации:  о клиентах (карт-счет, Ф.И.О., адрес, телефон);  об автозаправке (код, название фирмы, адрес);  Добавление, просмотр, удаление информации о продаже топлива клиентам (дата продажи, карт-счет клиента, код автозаправки, код топлива, количество);  Просмотр информации о фирме-поставщике (название, юридический адрес, телефон). |

**2 Инфологическое проектирование**

Для решения задач предметной области необходимо построить модель, в которой можно выделить следующие объекты:

1. Администратор (автозаправочной станции) – человек, который контролирует работу автозаправки. Администратор только один. Добавляется в базу, удаляется из базы, редактируется разработчиком. Никто не может просматривать данные об администраторе. Имеет следующие атрибуты:
   1. Логин:
      1. Выдается администратору разработчиком базы данных;
      2. Строка длиной 4–12 символов, состоящая из символов английского алфавита (верхнего и нижнего регистра);
      3. Используется для идентификации администратора в системе;
      4. Может изменяться только разработчиком по запросу администратора;
   2. Пароль:
      1. Выдается разработчиком базы данных;
      2. Строка длиной не менее 6 символов, включая символы английского алфавита (верхнего и нижнего регистра) и цифры;
      3. Используется для подтверждения личности администратора;
      4. Может изменяться только разработчиком по запросу администратора;

Отношения между атрибутами:

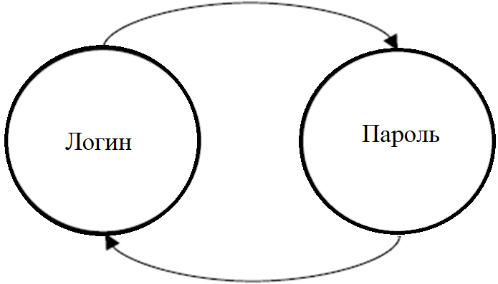


Рисунок 1. «Отношения между атрибутами пользователя “администратор” логин-пароль»

1. Фирма-поставщик - пользователь, отвечающий за информацию о своих автозаправках, видах топлива и продажах клиентам. Возможности: добавление, просмотр, редактирование и удаление информации (согласно таблице 1).
   1. Логин:
      1. Выдается разработчиком базы данных;
      2. Строка длиной 4–12 символов, состоящая из символов английского алфавита (верхнего и нижнего регистра);
      3. Используется для идентификации пользователя Фирма-поставщик в системе;
      4. Может изменяться только разработчиком по запросу фирмы;
   2. Пароль:
      1. Выдается разработчиком базы данных;
      2. Строка длиной не менее 6 символов, включая символы английского алфавита (верхнего и нижнего регистра) и цифры;
      3. Используется для подтверждения личности Фирма-поставщик;
      4. Может изменяться только разработчиком по запросу фирмы;

Отношения между атрибутами:

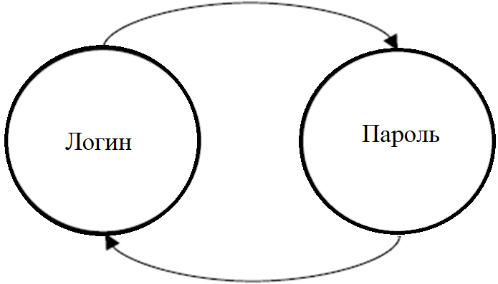


Рисунок 2. «Отношения между атрибутами пользователя “фирма-поставщик” логин-пароль»

1. Клиент - человек, имеющий возможность просматривать информацию о собственном карт-счете, Ф.И.О., адресе и телефоне. Покупает топливо на автозаправках. Добавляется, редактируется, просматривается администратором автозаправки. Удаляется администратором. Имеет следующие атрибуты:
   1. Карточный счёт: уникальный номер клиента.
      1. Целое положительное число;
      2. Генерируется при создании клиента;
      3. Не может быть пустым полем;
      4. Используется для идентификации пользователя (является уникальным идентификатором);
   2. ФИО:
      1. Представляет собой строку, состоящую из трех (двух – в случае отсутствия отчества получателя) слов, разделенных пробелом, на русском языке, первые буквы заглавные;
      2. Заполняется пользователем администратор автозаправки при создании;
      3. Поле обязательно для заполнения;
      4. Не является уникальным;
      5. Используется для обращения к клиенту;
   3. Адрес:
      1. Представляет собой строку, в которой через пробел и запятые указаны название города, улицы, номер;
      2. Город и улица – строки, состоящие из букв русского алфавита, начинающиеся с заглавной буквы;
      3. Номер дома – целое положительное число. Заполняется пользователем администратор заправки при регистрации клиента;
      4. Не является обязательным для заполнения полем;
      5. Не является уникальным;
   4. Телефон:
      1. Представляет собой 11 цифр, начинающихся с 7. Используется для связи с клиентом (отправка рассылок, предложений);
      2. По желанию клиента заполняется пользователем администратор при создании;
      3. Является уникальным полем, но необязательно к заполнению;

Отношения между атрибутами:

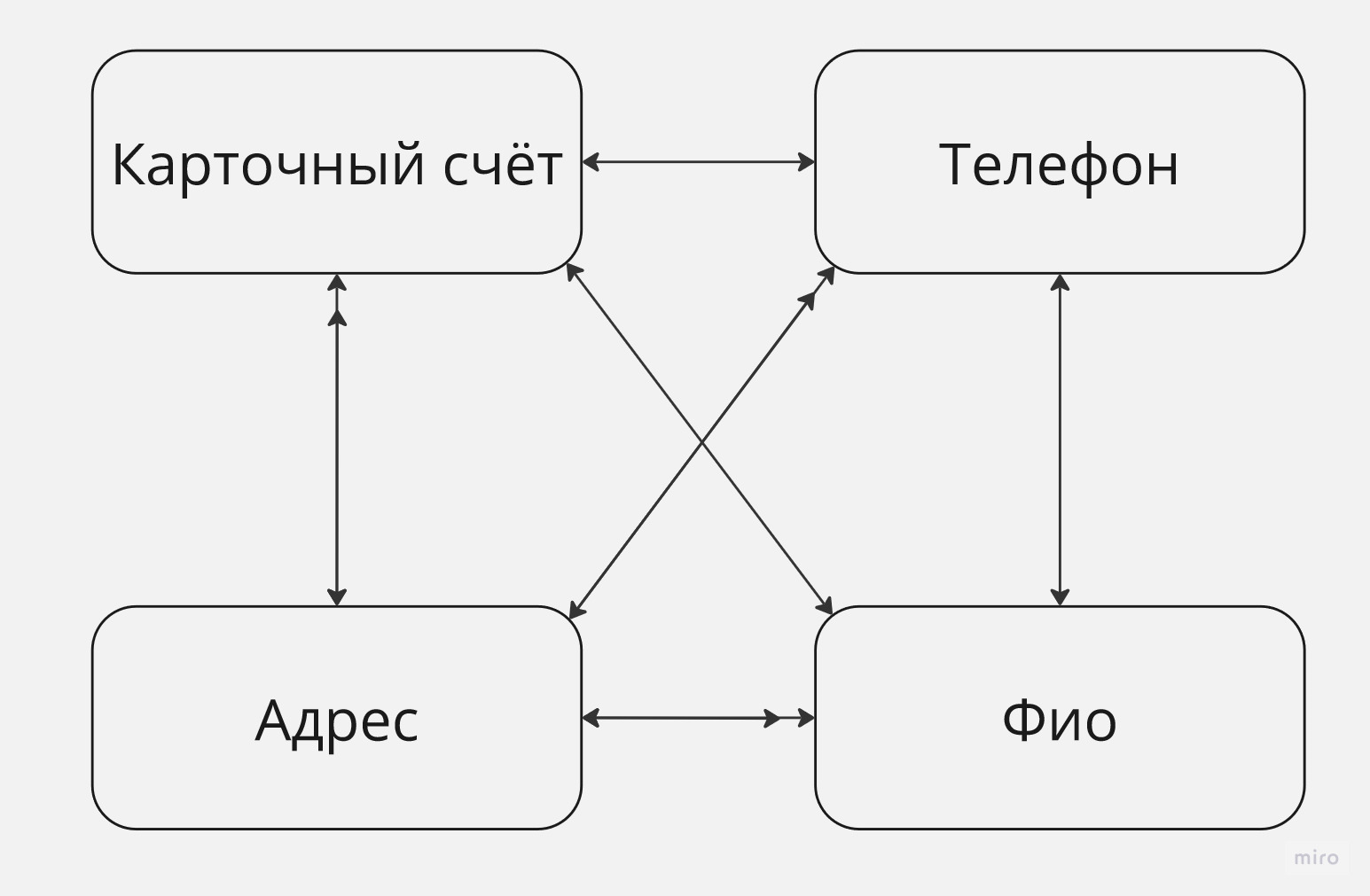


Рисунок 3. «Отношения между атрибутами сущности “Клиент”»

1. "Фирма-поставщик" представляет компанию, занимающуюся поставкой топлива на автозаправки. В контексте базы данных по автозаправкам фирма-поставщик отвечает за поставку топлива на различные заправочные станции. Вот некоторые характеристики этой сущности:
   1. Название:
      1. Строка, состоящая из букв русского алфавита;
      2. Заполняется пользователем фирма-поставщик при создании;
      3. Поле является обязательным для заполнения;
   2. Юр. адрес:
      1. Представляет собой строку, в которой через пробел указаны название города, улицы, номер дома, офис (если есть);
      2. Город и улица – строки, состоящие из букв русского алфавита, начинающиеся с заглавной буквы;
      3. Номер дома и офиса (при наличии) – целые положительные числа;
      4. Заполняется разработчиком базы данных и не изменяется в течение работы с базой;
      5. Поле является обязательным для заполнения;
      6. Поле является уникальным;
   3. Телефон:
      1. Представляет собой 11 цифр, начинающихся с 7;
      2. Используется для связи с фирмой по телефону;
      3. Заполняется разработчиком базы данных и не изменяется;
      4. Поле является обязательным для заполнения;
      5. Поле является уникальным;

Отношения между атрибутами представлены на рисунке 4.

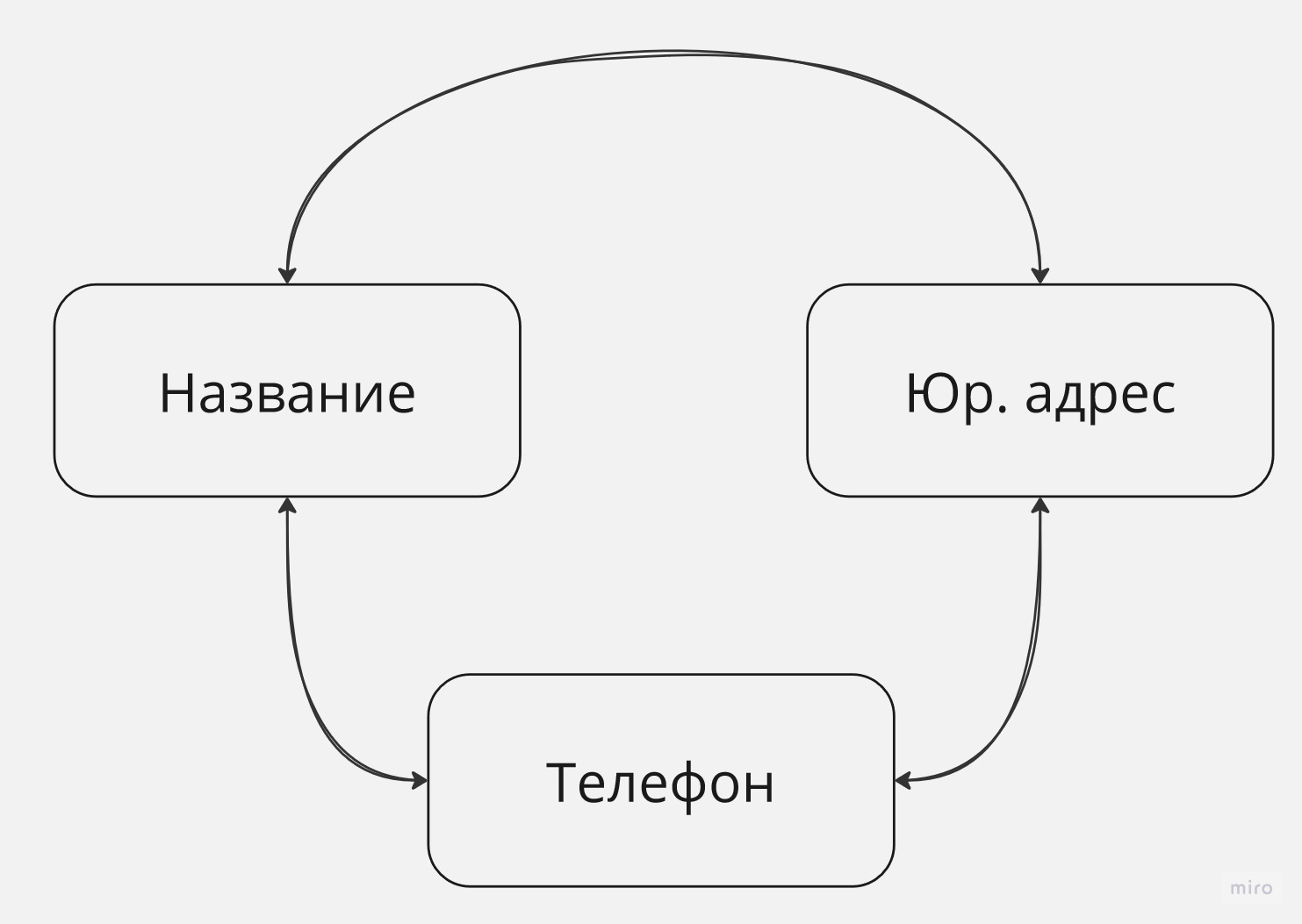


Рисунок 4. «Отношения между атрибутами сущности “Фирмы”»

1. Автозаправки - предоставляют услугу заправки автомобилей различных видов топливом, обеспечивая клиентов необходимым топливом для их транспортных средств.
   1. Код:
      1. Уникальный номер каждой заправки;
      2. Генерируется автоматически;
      3. Является уникальным полем;
      4. Обязательно для заполнения (не может быть пустым);
   2. Адрес:
      1. Представляет собой строку, в которой через пробел указаны название города, улицы, номер;
      2. Город и улица – строки, состоящие из букв русского алфавита, начинающиеся с заглавной буквы;
      3. Номер – целое положительное число;
      4. Поле является уникальным (не может быть создано 2 заправки на один адрес);
      5. Заполняется пользователем администратор при создании;
   3. Фирма:
      1. Связь с фирмой-поставщиком;
      2. Ссылка (название или номер) на фирму в базе данных;
      3. Поле уникальное;
      4. Обязательное для заполнения;
      5. Одна автозаправка принадлежит одной фирме, причём фирма имеет несколько автозаправок;

На рисунке 5 представлены отношения между атрибутами “Автозаправки”.

Изображение выглядит как текст, диаграмма, линия, Шрифт

Автоматически созданное описание

Рисунок 5. «Отношения между атрибутами сущности “Автозаправки”»

1. Виды топлива - представляют собой различные категории топлива, предназначенные для использования в транспортных средствах. Каждый вид топлива имеет уникальные характеристики и применяется в зависимости от типа двигателя автомобиля. Например, бензин с различными октановыми числами (76, 92, 95, 96), дизельное топливо или газ.
   1. Код:
      1. Уникальный идентификатор каждого вида топлива;
      2. Целое положительное число;
      3. Генерируется автоматически;
      4. Не может быть пустым;
      5. Является уникальным идентификатором;
   2. Вид:
      1. Описание вида топлива (92, 95, дизель и т. д.);
      2. Строка, представляющая вид топлива;
      3. Заполняется пользователем фирма-поставщик при создании;
      4. Является обязательным полем;
   3. Единица измерения:
      1. Строка, указывающая единицу измерения топлива;
      2. Не заполняется пользователем фирма-поставщик при создании, по умолчанию установлено поле ‘ЛИТР’;
      3. Обязательное поле;
   4. Цена:
      1. Десятичное положительное число;
      2. Заполняется пользователем фирма-поставщик при создании вида топлива;
      3. Является обязательным для заполнения;
      4. Не более трёх знаков в целой части, и двух в дробной.

Рисунок 6 отображает отношения между атрибутами сущности “Виды топлива”.

Изображение выглядит как текст, диаграмма, линия, зарисовка

Автоматически созданное описание

Рисунок 6. «Отношения между атрибутами сущности “Виды топлива”»

1. Продажи в контексте автозаправки представляют собой информацию о процессе реализации топлива клиентам. Это включает в себя следующие данные:
   1. **Дата:**
      1. Представляет собой значение в формате «дд.мм.гггг», где «дд» ­– две цифры, обозначающие день, «мм» - две цифры, обозначающие месяц, «гггг» –четыре цифры, обозначающие год, далее через пробел указаны чч:мм, где «чч» –две цифры, обозначающие часы продажи, «мм» – две цифры, обозначающие минуты продажи;
      2. Позволяет отслеживать и учитывать дату каждой сделки;
   2. Карточный счёт клиента:
      1. Уникальный номер клиента;
      2. Целое положительное число;
      3. Генерируется при создании клиента;
      4. Является уникальным идентификатором;
      5. Помогает отслеживать продажи конкретному клиенту;
   3. Код автозаправки:
      1. Уникальный номер каждой заправки;
      2. Генерируется автоматически;
      3. Является уникальным идентификатором;
      4. Код указывает место, где произошла продажа топлива;
   4. Код топлива:
      1. Уникальный идентификатор каждого вида топлива;
      2. Целое положительное число;
      3. Генерируется автоматически;
      4. Этот код указывает, какой вид топлива был продан клиенту;
   5. Количество:
      1. Объем проданного топлива в литрах;
      2. Десятичное положительное число;
      3. Заполняется пользователем администратор при создании продажи;
      4. Является обязательным полем;
      5. Значение помогает отслеживать и учитывать количество проданного топлива;



Рисунок 7. «Отношения между атрибутами сущности “Продажи”»

**Взаимосвязь объектов при решении задач ПО**

При решении пользователем «Клиент» задачи «Просмотр информации о карт-счёте» выделяются следующие взаимосвязи объектов предметной области:



Рисунок 8. «Отношения между объектами при решении задачи “Просмотр информации о клиенте”»

При решении пользователем «Клиент» задачи «Просмотр информации о совершённых покупках в сети автозаправочных станций» выделяются следующие взаимосвязи объектов предметной области (рисунок 9):

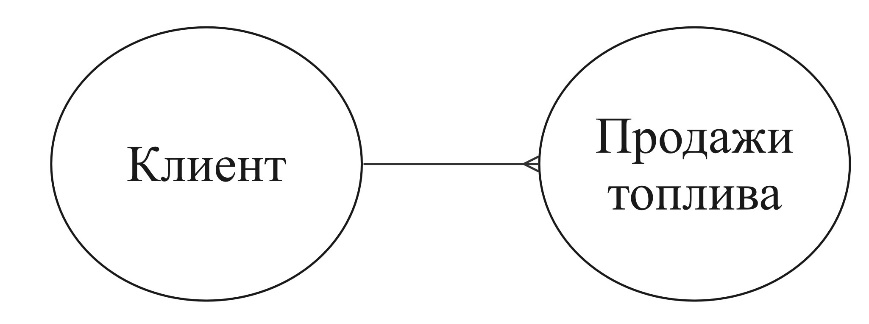


Рисунок 9. «Отношения между объектами при решении задачи “Просмотр информации о совершённых покупках в сети автозаправочных станций”»

При решении пользователем «Фирма-поставщик» задач «Добавление, просмотр, редактирование и удаление информации о видах топлива» выделяются следующие взаимосвязи объектов предметной области:

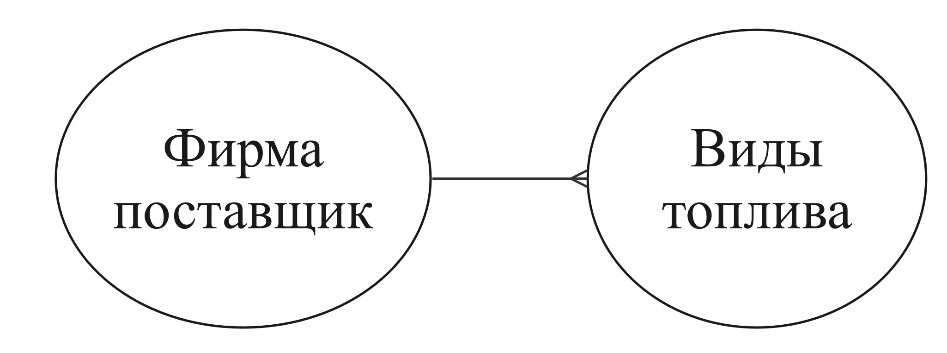


Рисунок 10. «Отношения между объектами при решении задачи “Добавление, просмотр, редактирование и удаление информации о видах топлива”»

При решении пользователем «Фирма-поставщик» задач «Просмотр информации о своих автозаправках» выделяются следующие взаимосвязи объектов предметной области:

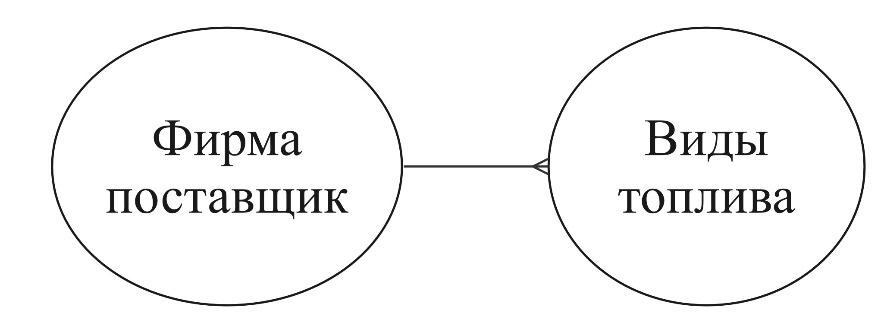


Рисунок 11. «Отношения между объектами при решении задачи “Просмотр информации о своих автозаправках”»

При решении пользователем «Фирма-поставщик» задач «Просмотр информации о продаже топлива клиентам за выбранный период» выделяются следующие взаимосвязи объектов предметной области:



Рисунок 12. «Отношения между объектами при решении задачи “Просмотр информации о продаже топлива клиентам за выбранный период”»

При решении пользователем «Администратор» задач «Добавление, просмотр, редактирование и удаление информации о клиентах» выделяются следующие взаимосвязи объектов предметной области:

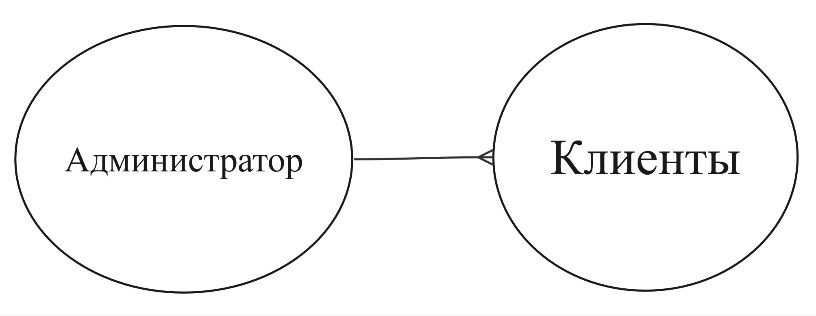


Рисунок 13. «Отношения между объектами при решении задачи “Добавление, просмотр, редактирование и удаление информации о клиентах”»

При решении пользователем «Администратор» задач «Добавление, просмотр, редактирование и удаление информации об автозаправках» выделяются следующие взаимосвязи объектов предметной области:

Изображение выглядит как круг, диаграмма

Автоматически созданное описание

Рисунок 14. «Отношения между объектами при решении задачи “Добавление, просмотр, редактирование и удаление информации об автозаправках”»

При решении пользователем «Администратор» задач «Добавление, просмотр, удаление информации о продаже топлива клиентам» выделяются следующие взаимосвязи объектов предметной области:

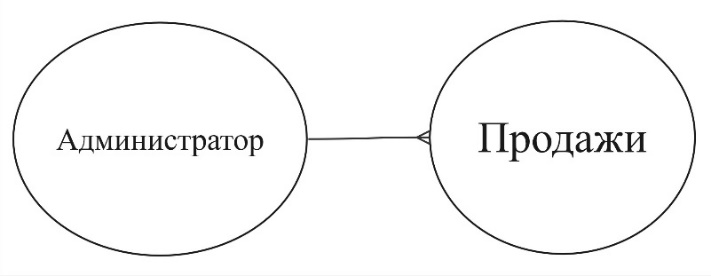


Рисунок 15. «Отношения между объектами при решении задачи “Добавление, просмотр, удаление информации о продаже топлива клиентам”»

При решении пользователем «Администратор» задач «Просмотр информации о фирме-поставщике» выделяются следующие взаимосвязи объектов предметной области:

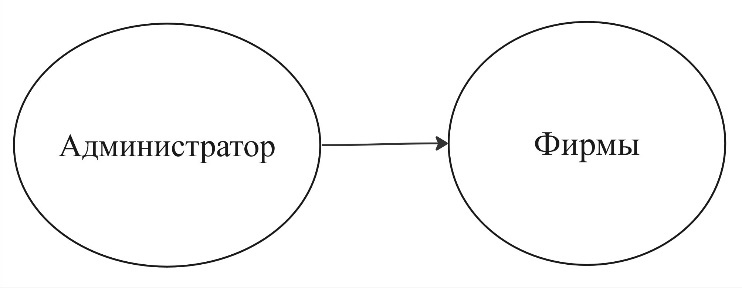


Рисунок 16. «Отношения между объектами при решении задачи “Просмотр информации о фирме-поставщике”»

**Концептуальная модель данных**

Для каждого из выделенных объектов предметной области строим реляционную модель данных. В итоговых таблицах ключевые поля в таблице будут выделены жирным шрифтом.

1. Информационному объекту “Администратор” ставим в соответствие таблицу:

|  |  |
| --- | --- |
| Логин | Пароль |
| admin | admin123 |

Выделим ключевое поле - «Логин».

Зависимости полей:



Рисунок 17. «Зависимости полей таблицы “Администратор”»

Поля не имеют дубликатов в различных записях, каждое поле содержит только одно значение. Следовательно, отношение находится в I нормальной форме.

Каждое неключевое поле неприводимо зависит от ключевого. Неключевое поле «Пароль» зависит только от ключевого поля «Логин». Следовательно, отношение находится во II нормальной форме.

Неключевые поля не зависят от других неключевых полей. Неключевое поле «Пароль» зависит только от ключевого поля «Логин». Следовательно, отношение находится в III нормальной форме.

При добавлении, удалении, редактировании записи разработчиком базы данных не возникает аномалий.

После нормализации получаем таблицу следующего вида:

|  |  |
| --- | --- |
| **Логин** | Пароль |
| admin | admin123 |

1. Информационному объекту “Фирма-поставщик” ставим в соответствие таблицу:

|  |  |
| --- | --- |
| Логин | Пароль |
| Лукойл | qwerty321! |

Выделим ключевое поле - «Логин».

Зависимости полей:

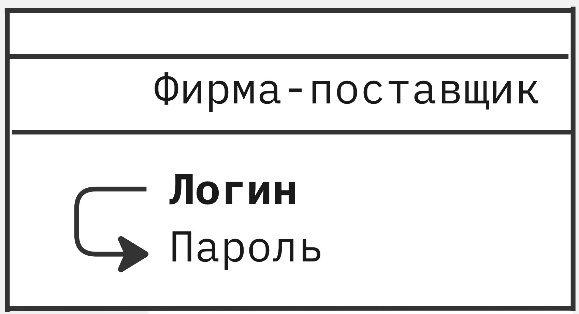


Рисунок 18. «Зависимости полей таблицы “Фирма-поставщик”»

Поля не имеют дубликатов в различных записях, каждое поле содержит только одно значение. Следовательно, отношение находится в I нормальной форме.

Каждое неключевое поле неприводимо зависит от ключевого. Неключевое поле «Пароль» зависит только от ключевого поля «Логин». Следовательно, отношение находится во II нормальной форме.

Неключевые поля не зависят от других неключевых полей. Неключевое поле «Пароль» зависит только от ключевого поля «Логин». Следовательно, отношение находится в III нормальной форме.

При решении задач “Добавление, просмотр, редактирование и удаление информации о видах топлива” и “Просмотр информации о своих автозаправках и о продаже топлива клиентам за выбранный период.” аномалий не возникает.

После нормализации получаем таблицу следующего вида:

|  |  |
| --- | --- |
| **Логин** | Пароль |
| Лукойл | qwerty321! |

1. Информационному объекту “Пользователь” ставим в соответствие таблицу:

|  |  |
| --- | --- |
| Логин | Пароль |
| 100003 | qwerty123 |

Выделим ключевое поле - «Логин».

Зависимости полей:

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рисунок 19. «Зависимости полей таблицы “Пользователь”»

Поля не имеют дубликатов в различных записях, каждое поле содержит только одно значение. Следовательно, отношение находится в I нормальной форме.

Каждое неключевое поле неприводимо зависит от ключевого. Неключевое поле «Пароль» зависит только от ключевого поля «Логин». Следовательно, отношение находится во II нормальной форме.

Неключевые поля не зависят от других неключевых полей. Неключевое поле «Пароль» зависит только от ключевого поля «Логин». Следовательно, отношение находится в III нормальной форме.

При решении задач “ Просмотр информации о своем карт-счете, Ф.И.О., адресе, телефоне и о совершённых покупках в сети автозаправочных станций” аномалий не возникает.

После нормализации получаем таблицу следующего вида:

|  |  |
| --- | --- |
| **Логин** | Пароль |
| 100003 | qwerty123 |

1. Поскольку после нормализации таблицы «Администратор», «Фирма-поставщик» и «Пользователь» оказались идентичны, чтобы устранить избыточность хранения данных, преобразуем таблицы «Администратор», «Фирма-поставщик» и «Пользователь» в одну таблицу «Клиент», имеющую дополнительное поле «Роль». Поле «Роль» будет принимать одно из значений «Администратор», «Фирма-поставщик» и «Пользователь» соответственно.

Зависимости полей:



Рисунок 20. «Зависимости полей таблицы “Клиент”»

При решении задач “Просмотр информации о своем карт-счете, Ф.И.О., адресе, телефоне и о совершённых покупках в сети автозаправочных станций.” аномалий не возникает.

Получаем таблицу следующего вида:

|  |  |  |
| --- | --- | --- |
| **Логин** | Пароль | Роль |
| 100003 | qwerty123 | Пользователь |

1. Информационному объекту “Пользователь” ставим в соответствие таблицу:

|  |  |  |  |
| --- | --- | --- | --- |
| Карточный счёт | ФИО | Адрес | Номер телефона |
| 100003 | Иванов Иван Иванович | г. Владивосток, ул. Уборевича, 26 | 79123456789 |

Выделим ключевое поле – “Карточный счёт”.

Зависимости полей:



Рисунок 21. «Зависимости полей таблицы “Пользователь”»

Поля не имеют дубликатов в различных записях, каждое поле содержит только одно значение. Следовательно, отношение находится в I нормальной форме.

Каждое неключевое поле неприводимо зависит от ключевого. Неключевые поля «ФИО, Адрес, Номер телефона» зависят только от ключевого поля «Карточный счёт». Следовательно, отношение находится во II нормальной форме.

Неключевые поля не зависят от других неключевых полей. Неключевые поля «ФИО, Адрес, Номер телефона» зависят только от ключевого поля «Карточный счёт». Следовательно, отношение находится в III нормальной форме.

При решении задач “Просмотр информации о своем карт-счете, Ф.И.О., адресе, телефоне и о совершённых покупках в сети автозаправочных станций” аномалий не возникает.

После нормализации получаем таблицу следующего вида:

|  |  |  |  |
| --- | --- | --- | --- |
| **Карточный счёт** | ФИО | Адрес | Номер телефона |
| 100003 | Иванов Иван Иванович | г. Владивосток, ул. Уборевича, 26 | 79123456789 |

1. Информационному объекту “Автозаправка” ставим в соответствие следующую таблицу:

|  |  |  |
| --- | --- | --- |
| Код автозаправки | Название фирмы | Адрес |
| 1006 | Лукойл | г. Владивосток, ул. Уборевича, 26 |

Выделим ключевое поле – “Код автозаправки”.

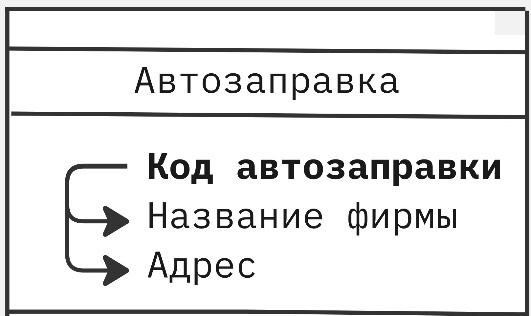


Рисунок 22. «Зависимости полей таблицы “Автозаправка”»

Поля не имеют дубликатов в различных записях, каждое поле содержит только одно значение. Следовательно, отношение находится в I нормальной форме.

Каждое неключевое поле неприводимо зависит от ключевого. Неключевые поля «Название фирмы, Адрес» зависят только от ключевого поля «Код автозаправки». Следовательно, отношение находится во II нормальной форме.

Неключевые поля не зависят от других неключевых полей. Неключевые поля «Название фирмы, Адрес» зависят только от ключевого поля «Код автозаправки». Следовательно, отношение находится в III нормальной форме.

При решении задач “Просмотр фирмой информации о своих автозаправках”, “Добавление, просмотр, редактирование и удаление информации об автозаправках администратором”, где задействован объект “автозаправка”, аномалий не возникает.

После нормализации получаем таблицу следующего вида:

|  |  |  |
| --- | --- | --- |
| **Код автозаправки** | Название фирмы | Адрес |
| 1006 | Лукойл | г. Владивосток, ул. Уборевича, 26 |

Так как поле “Название фирмы” является ссылкой на объект “Фирма-поставщик” из п. 7, поэтому получим объединение двух таблиц:

“Автозаправка”

|  |  |  |
| --- | --- | --- |
| **Код автозаправки** | Название фирмы | Адрес |
| 1006 | Лукойл | г. Владивосток, ул. Уборевича, 26 |

“Фирма-поставщик”

|  |  |  |
| --- | --- | --- |
| Название фирмы | Юр. Адрес | Телефон |
| Лукойл | г. Керчь, ул. Пуло, 48 | 79991231231 |

После связывания этих таблиц получим зависимость:

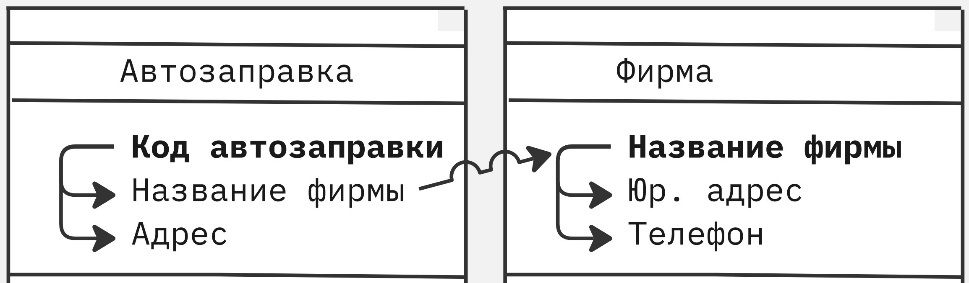


Рисунок 23. «Зависимости полей таблиц “Автозаправка” и “Фирма-поставщик”»

Ключевым полем для объекта “Фирма-поставщик” является “Название фирмы”.

|  |  |  |
| --- | --- | --- |
| **Название фирмы** | Юр. Адрес | Телефон |
| Лукойл | г. Керчь, ул. Пуло, 48 | 79991231231 |

1. Информационному объекту “Фирма-поставщик” ставим в соответствие следующую таблицу:

|  |  |  |
| --- | --- | --- |
| Название фирмы | Юр. Адрес | Телефон |
| Лукойл | г. Керчь, ул. Пуло, 48 | 79991231231 |

Выделим ключевое поле – “Название фирмы”.

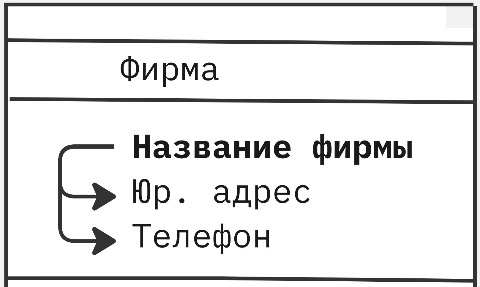


Рисунок 24. «Зависимости полей таблиц “Фирма-поставщик”»

Поля не имеют дубликатов в различных записях, каждое поле содержит только одно значение. Следовательно, отношение находится в I нормальной форме.

Каждое неключевое поле неприводимо зависит от ключевого. Неключевые поля «Юр. Адрес, номер телефона» зависят только от ключевого поля «Название фирмы». Следовательно, отношение находится во II нормальной форме.

Неключевые поля не зависят от других неключевых полей. Неключевые поля «Юр. Адрес, номер телефона» зависят только от ключевого поля «Название фирмы». Следовательно, отношение находится в III нормальной форме.

При решении задач “Просмотр фирмой информации о своих автозаправках”, “Добавление, просмотр, редактирование и удаление информации о фирмах разработчиком базы данных”, где задействован объект “фирма”, аномалий не возникает.

После нормализации получим таблицу:

|  |  |  |
| --- | --- | --- |
| **Название фирмы** | Юр. Адрес | Телефон |
| Лукойл | г. Керчь, ул. Пуло, 48 | 79991231231 |

1. Информационному объекту “Продажа топлива клиентам” ставим в соответствие следующую таблицу, где ID продажи будет являться ключевым полем:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID продажи** | Время продажи | Карточный счёт клиента | Код автозаправки | Код топлива | Объём |
| 500000 | г. Керчь, ул. Пуло, 48 | 100001 | 1000 | 100 | 50.00 |

Каждому неключевому полю кроме времени продажи и объёма поставим в соответствие соответствующую таблицу из пунктов 5, 6, 9. Тогда получим следующую связь между таблицами по неключевым полям (рисунок 25):

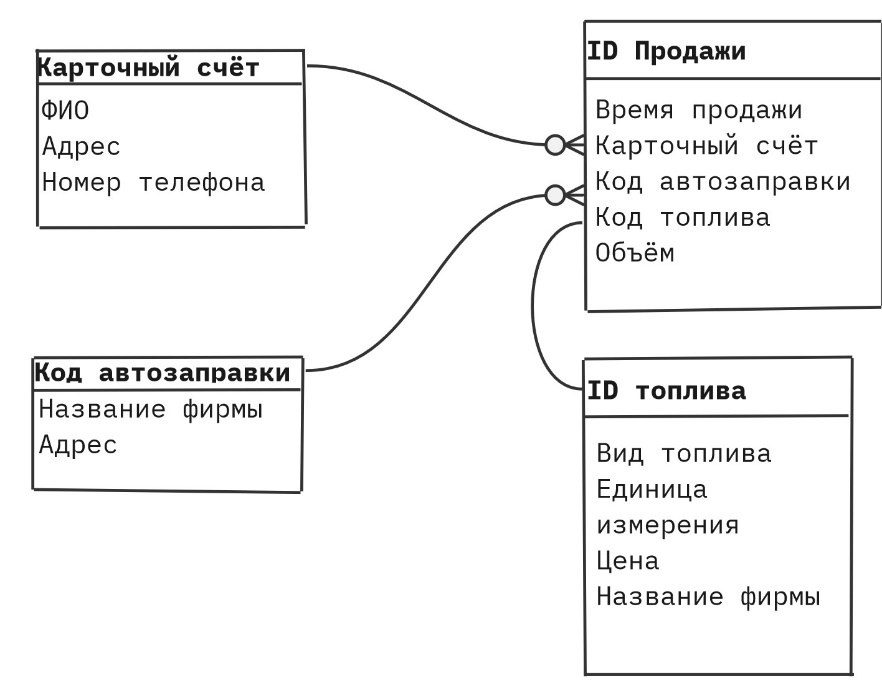


Рисунок 25. «Зависимости полей таблиц “Фирма-поставщик”, “Клиент”, “Продажи”, “Автозаправка”»

1. Информационному объекту “Топливо” ставим в соответствие следующую таблицу, где ID топлива будет являться ключевым полем:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID топлива** | Вид топлива | Единица измерения | Цена топлива | Фирма |
| 110 | АИ-92 | ЛИТР | 10,98 | Лукойл |

Подставим для неключевого поля “Фирма” ссылку на таблицу “Фирма” по ключевому полю название фирмы.

Получим следующее отношение (рисунок 26):

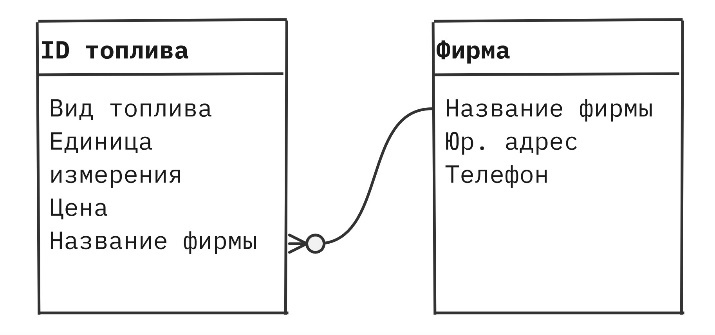


Рисунок 26. «Зависимости полей таблиц “Фирма-поставщик”, “Топливо”»

Исходя из описанных связей, можно построить следующую схему отношений между сущностями (Entity Relationship - диаграмма). Ознакомиться с ней вы можете на рисунке 27.

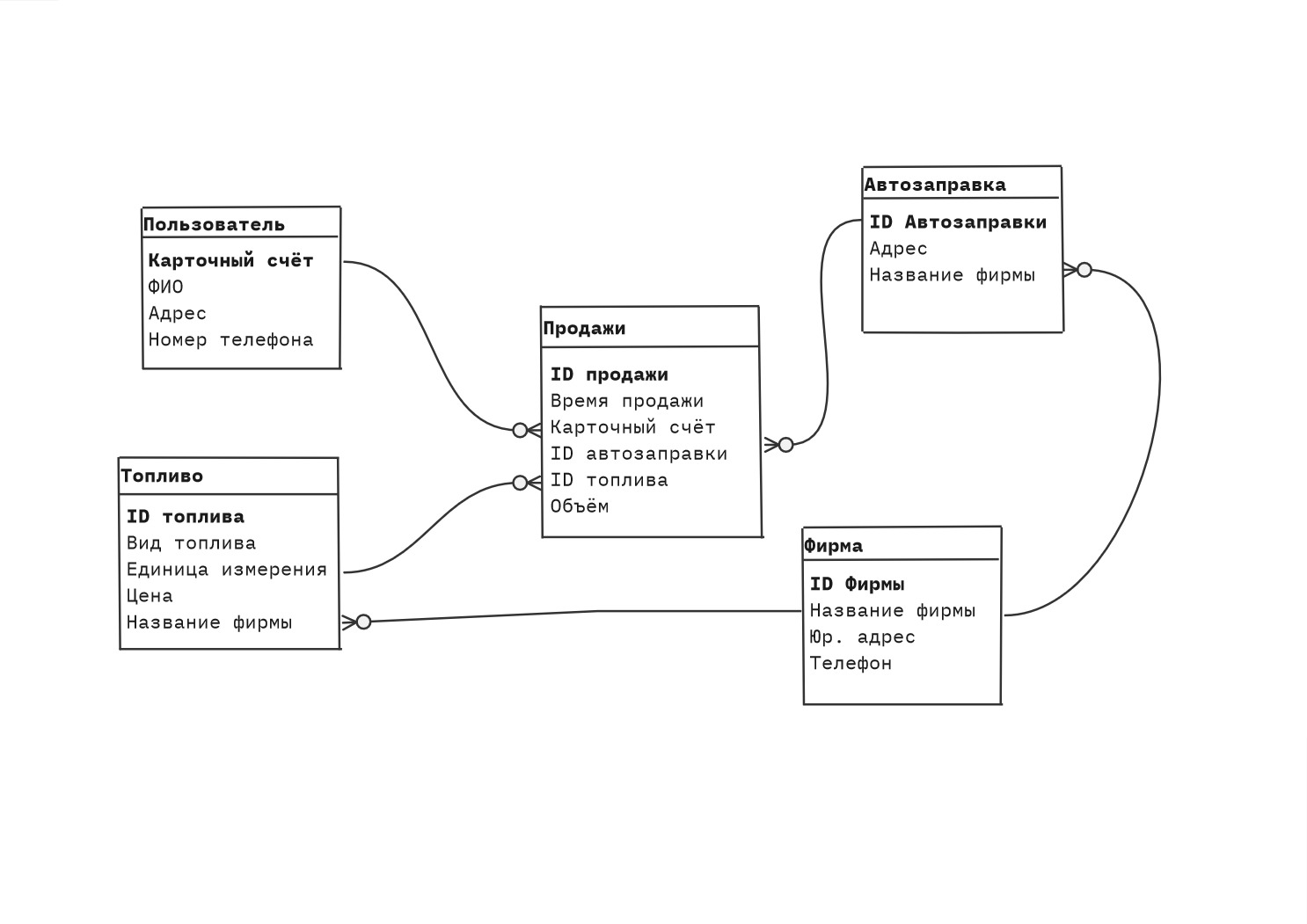


Рисунок 27. Схема отношений между нормализованными таблицами “Пользователь”, “Топливо”, “Продажи”, “Фирма”, “Автозаправка” базы данных.

**3 Даталогическое проектирование**

**Выбор СУБД**

Для реализации построенной модели данных претендентом является СУБД MySQL Workbench. MySQL Workbench — это инструмент разработки, управления и администрирования баз данных MySQL. Он был разработан и поддерживается компанией Oracle Corporation.

В отличие от Access, MySQL Workbench является СУБД типа реляционной модели и также поддерживает формирование запросов на языке SQL. Он предоставляет богатые возможности для создания и управления таблицами, включая запуск и оптимизацию запросов.

MySQL Workbench обладает интегрированной средой разработки, которая позволяет программировать интерфейс и создавать пользовательский интерфейс для удобной работы с данными. Она предоставляет мощные инструменты для автоматизации задач с помощью сценариев и макросов.

Ограничения для СУБД MySQL Workbench:

* Максимальный размер базы данных: ограничен только доступным местом на сервере MySQL.
* Максимальный размер таблицы: ограничен только доступным местом на сервере MySQL.
* Количество полей в таблице: ограничено до 4096.
* Количество открытых таблиц: ограничено до 4096.
* Количество одновременно работающих пользователей: ограничено до 4096.

Типы данных, поддерживаемые в MySQL Workbench:

* Текстовый: VARCHAR, TEXT, ENUM и другие.
* Числовой: INTEGER, FLOAT, DECIMAL и другие.
* Дата/время: DATE, TIME, DATETIME и другие.
* Бинарный: BLOB, BIT и другие.
* Другие типы данных, такие как JSON, XML.

Элементы базы данных:

1. Сервер: Элемент "Сервер" отображает доступные серверы MySQL, с которыми можно подключиться. Пользователь может добавлять, редактировать и удалять серверы из этого элемента.
2. Базы данных: Элемент "Базы данных" отображает список созданных баз данных на выбранном сервере. Здесь можно создавать новые базы данных, изменять их параметры и удалять.
3. Таблицы: Элемент "Таблицы" показывает структуру таблиц в выбранной базе данных. Здесь можно создавать новые таблицы, изменять поля таблиц, добавлять и удалять записи в таблицах.
4. Запросы: Элемент "Запросы" предоставляет возможность создания и выполнения SQL-запросов к базе данных. Здесь можно писать и отредактировать запросы, а затем выполнить их для получения результата.
5. Схема: Схема базы данных позволяет визуально представить структуру таблиц и связи между ними. Элемент "Схема" позволяет создавать и редактировать схему базы данных, добавлять таблицы, определять их поля и связи.
6. Модели данных: Элемент "Модели данных" используется для создания визуальной модели базы данных. Это графическое представление структуры таблиц и связей. Здесь можно создавать новые модели, редактировать их и сохранять как файлы проекта.
7. Резервные копии: Элемент "Резервные копии" позволяет выполнять резервное копирование и восстановление баз данных MySQL. Здесь можно создавать резервные копии данных, восстанавливать их и контролировать процесс восстановления.
8. Визуализация: MySQL Workbench предоставляет различные инструменты для визуализации данных и запросов. Здесь можно создавать и редактировать диаграммы данных, графики, отчеты и другие визуальные элементы.

Форматы файлов для баз данных MySQL Workbench сохраняются в формате .sql, который содержит SQL-скрипт для создания и заполнения базы данных.

Таким образом, СУБД MySQL Workbench является хорошим вариантом для реализации созданной нами модели данных, предоставляя широкий набор инструментов для работы с базами данных.

**Конструирование таблиц базы данных**

Создание таблиц для реализации базы данных происходило через SQL-запросы для наработки практических навыков их написания.

1. Создание вида с логинами, паролями и ролями:

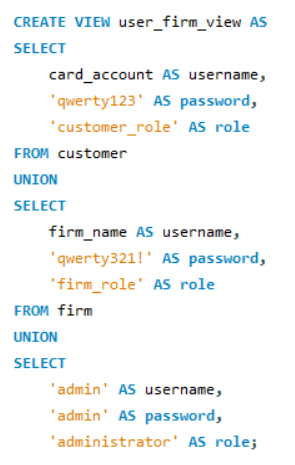


Рисунок 28. «Создание вида с логинами, паролями и ролями для существующих записей в соответствующих таблицах в СУБД MySql Workbench»

Поле логин для пользователей в данном случае берётся из номера карточного счёта, пароль по умолчанию “qwerty123”. При желании пароль можно изменить в ручном формате. При добавлении нового пользователя через интерфейс администратора аккаунт для клиента создаётся автоматически.

Поле логин для фирм-поставщиков соответствует названию фирмы. Пароль по умолчанию “qwerty321!”. При желании пароль можно изменить в ручном формате.

Данные для входа администратора: { “admin”, “admin” }. Меняется разработчиком базы данных.

1. Таблица “Фирмы”



Рисунок 29. «Создание таблицы “Фирмы” в СУБД MySql Workbench»

Поле “firm\_id” заполняется автоматически при создании фирмы разработчиком базы данных. Целочисленное значение.

Поля (firm\_name, firm\_address, firm\_phone) являются строковыми значениями с ограничениями в 100, 255, 11 символов соответственно. Все поля непустые и уникальные. Проверка на валидность данных в данном случае не делается, так как по умолчанию будем считать, что разработчик вводит корректные данные.

Пример корректных данных:

'Белнефтехим', 'Минск, ул. Притыцкого, 25', '70123456789' (firm\_id = 1)

'Славнефть', 'Гродно, ул. Ленина, 10', '70222113344' (firm\_id = 2)

'А-100', 'Брест, пр. Московский, 5', '70333221100' (firm\_id = 3)

1. Таблица “Пользователь”

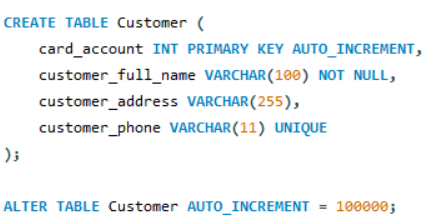


Рисунок 30. «Создание таблицы “Пользователь” в СУБД MySql Workbench»

Поле “card\_account” заполняется автоматически при создании пользователя администратором. Начальное значение – 100000. Поля (customer\_full\_name, customer\_address, customer\_phone) являются строковыми значениями с ограничениями в 100, 255, 11 символов соответственно. Поле ФИО непустое, обязательное к заполнению и не является уникальным. Состоит из трёх (в случае отсутствия отчества - двух) слов из букв русского алфавита, каждая первая буква нового слова – заглавная. Проверка на валидность данных в данном случае делается через регулярное выражение. Поле адрес и номер телефона также проверяются через регулярные выражения, но не являются обязательными к заполнению. Причём, если номер телефона заполнен для одного из пользователей, то невозможно будет добавить такой же номер телефона для нового пользователя и изменить номер телефона для существующего, так как он уникальный. Номер телефона начинается с 7 и содержит 11 цифр (включая 7).

Регулярные выражения для валидации данных:

ФИО: ^[А-ЯЁ][а-яё]+\s[А-ЯЁ][а-яё]+(\s[А-ЯЁ][а-яё]+)?$

Адрес: ^[А-ЯЁ][а-яё]+(?:,\s\*(ул\.|пр\.|просп\.|проспект)\s\*[А-ЯЁ][а-яё]+)?,\s\*\d+[А-ЯЁ]\*$

Номер телефона: ^7\d{10}$

Пример корректных данных:

'Иванов Иван Иванович', 'Минск, ул. Пушкина, 10', '79123456789’ (card\_number = 100000)

'Козлова Анна Сергеевна', 'Брест, пр. Гагарина, 15', '79345678901' (card\_number = 100001)

Обработка ошибок выполняется через интерфейс.

1. Таблица “Автозаправки”

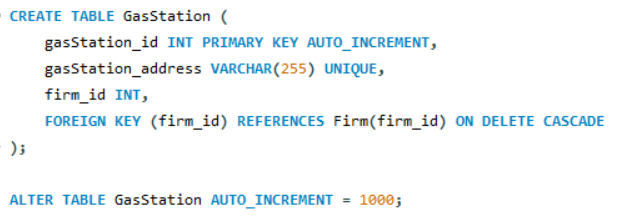


Рисунок 31. «Создание таблицы “Автозаправки” в СУБД MySql Workbench»

Поле “gasStation\_id” заполняется автоматически при создании автозаправки администратором. Начальное значение – 1000.

Поле gasStation\_address является строковым значением с ограничениям в 255 символов и является обязательным к заполнению. Невозможно создать две заправки по одному адресу.

Поле firm\_id является числовым значением и внешним ключом на таблицу “Фирмы”. При удалении фирмы, все связанные заправки пропадают. Обработка ошибок реализована через интерфейс.

Пример корректных данных:

'Минск, ул. Жукова, 25', 1 (gasStation\_id = 1000)

'Гродно, ул. Гагарина, 15', 2 (gasStation\_id = 1001)

'Брест, ул. Толстого, 20', 3 (gasStation\_id = 1002)

'Минск, ул. Строителей, 10', 1 (gasStation\_id = 1003)

'Гродно, ул. Пушкинская, 5', 2 (gasStation\_id = 1004)

1. Таблица “Топливо”

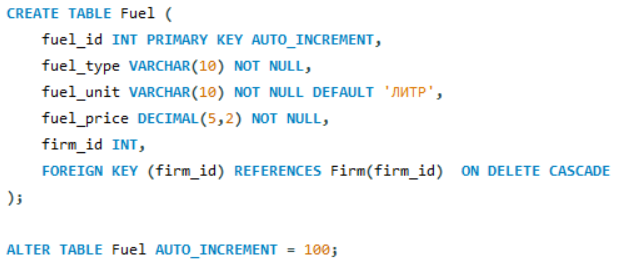


Рисунок 32. «Создание таблицы “Топливо” в СУБД MySql Workbench»

Поле “fuel\_id” заполняется автоматически при создании топлива фирмой-поставщиком. Начальное значение – 100. Поля (fuel\_type, fuel\_unit) являются строковыми значениями с ограничениями в 10 символов. По умолчанию для единицы измерения топлива установлено “ЛИТР”. Эта колонка недоступна для изменения. Поля fuel\_type выбираются из выпадающего списка и имеют значения {“АИ-76”, “АИ-92”, “АИ-95”, “АИ-96”, “Дизель”, “Газ”}. Поле fuel\_price является десятичным значением с тремя знаками в целой части и 2 знаками в дробной. При удалении фирмы, всё связанное с ним топливо пропадает. Обработка ошибок реализована через интерфейс.

Пример корректных данных:

'АИ-92', 2.40, 3 fuel\_id = 100, firm\_id = 3 (ID фирмы, создавшее топливо)

'АИ-95', 2.50, 3 fuel\_id = 101, firm\_id = 3 (ID фирмы, создавшее топливо)

'Газ', 1.00, 2 fuel\_id = 102, firm\_id = 2 (ID фирмы, создавшее топливо)

'АИ-92', 2.30, 2 fuel\_id = 103, firm\_id = 2 (ID фирмы, создавшее топливо)

'АИ-95', 2.43, 1 fuel\_id = 104, firm\_id = 1 (ID фирмы, создавшее топливо)

'Дизель', 4.00, 1 fuel\_id = 105, firm\_id = 1 (ID фирмы, создавшее топливо)

1. Таблица “Продажи”

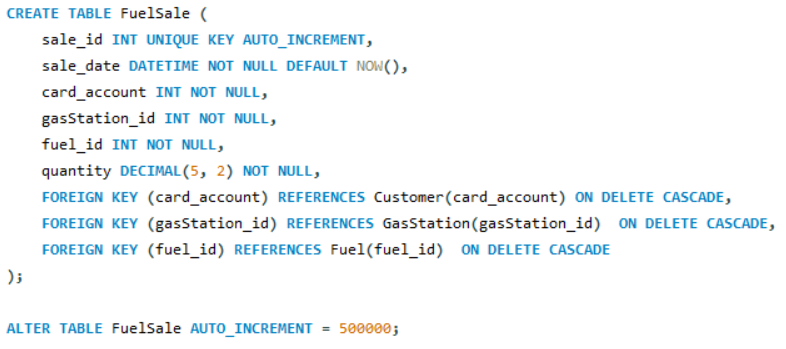


Рисунок 33. «Создание таблицы “Продажи” в СУБД MySql Workbench»

Поле “sale\_id” заполняется автоматически при создании продажи администратором. Начальное значение – 500000.

Поле sale\_date является значением типа ДатаВремя (ЧЧ.ММ.ГГГГ ЧЧ:ММ). Число, Месяц, Год, Часы, Минуты.

Card\_account – ID клиента, совершившего покупку. При удалении клиента связанные с ним записи удаляются. Если клиент не существует, то связать с ним продажу не получится.

gasStation\_id – ID автозаправки, на которой совершили покупку. Целочисленное значение. При удалении заправки все связанные с ней записи удаляются. Если заправки с таким кодом не существует, то связать с ней продажу не получится.

fuel\_id – ID купленного топлива. Если фирма, которой принадлежит заправка gasStation\_id не предоставляет вид топлива, то создать продажу не получится. Целочисленное значение. При удалении топлива все связанные с ней записи удаляются. Если топлива с таким кодом не существует, то связать с ним продажу не получится.

quantity является десятичным значением с тремя знаками в целой части и 2 знаками в дробной. Непустое значение, не может быть меньше нуля.

Обработка ошибок реализована через интерфейс.

Пример корректных данных:

100002, 1006, 108, 45.25 sale\_id = 500000

100003, 1006, 109, 60.75 sale\_id = 500001

100001, 1006, 110, 50.00 sale\_id = 500002

100002, 1006, 111, 45.25 sale\_id = 500003

В соответствии с построенными таблицами получаем следующую схему данных (рисунок 34):

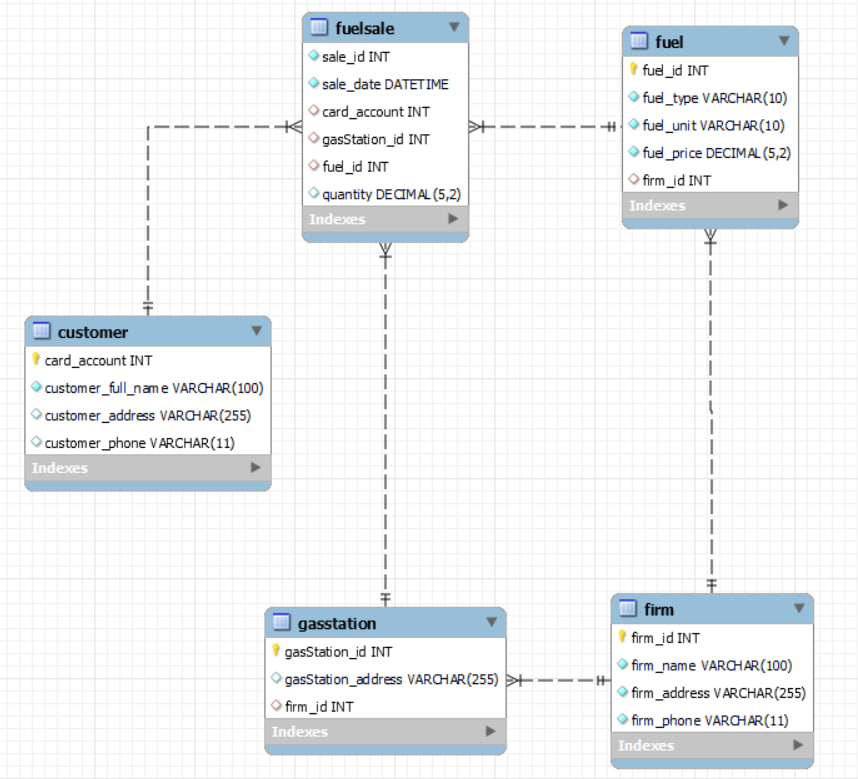


Рисунок 34. «Схема данных в СУБД MySql Workbench»

На схеме данных указаны ключевые поля и связи между атрибутами объектов.

Схема данных, автоматически построенная средствами СУБД MySql Workbench полностью соответствует схеме данных, построенной в процессе инфологического моделирования.

**Экранные формы**

Для работы пользователей с базой данных были созданы экранные формы с использованием библиотеки классов .NET Framework и языка программирования C#.

**Формы общего доступа**

1. Форма авторизации:

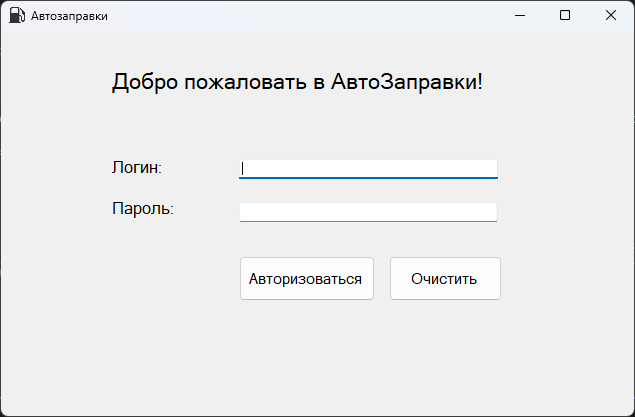


Рисунок 35. “Форма авторизации”

Форма выбора режима работы появляется при компиляции кода и подключении у СУБД. Предназначена для авторизации пользователя и продолжение работы в режиме «Администратора», «Фирмы-поставщика» или «Пользователя».

Компоненты формы:

* 1. Поле для ввода «Логин». Поле не имеет ограничений на ввод данных.
  2. Поле для ввода «Пароль». Поле не имеет ограничений на ввод данных.
  3. Кнопка «Очистить». Очищает поля «Логин» и «Пароль».
  4. Кнопка «Авторизоваться». Нажатие на кнопку приводит либо к закрытию данной формы и переходу к форме 2 (если введенные логин и пароль принадлежат пользователю с ролью «Администратор»), форме 3 (если введенный логин и пароль принадлежат пользователю с ролью «Фирма-поставщик») или форме 4 (если введённые логин и пароль принадлежат пользователю с ролью «Клиент»), либо появлению одного из следующих сообщений.
     1. “Введите логин. Введите пароль.”. Данное сообщение появляется, если поля «Логин» и «Пароль» пустые.

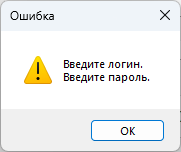


Рисунок 36. Сообщение “Введите логин. Введите пароль.” при авторизации.

* + 1. “Введите пароль.”. Данное сообщение появляется, если поле «Пароль» пустое.

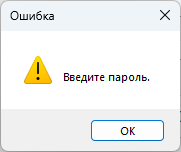


Рисунок 37. Сообщение “Введите пароль.” при авторизации.

* + 1. “Введите логин.”. Данное сообщение появляется, если поле «Логин» пустое.

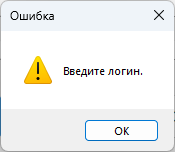


Рисунок 38. Сообщение “Введите логин.” при авторизации.

* + 1. “Неверный логин или пароль!” Данное сообщение появляется, если не удаётся найти данные в таблице с пользователями.

Изображение выглядит как текст, снимок экрана, Шрифт, Операционная система

Автоматически созданное описание

Рисунок 39. Сообщение “Неверный логин или пароль!” при авторизации.

* + 1. “{username}, добро пожаловать в АвтоЗаправки с правами {role}!” Данное сообщение появляется, если данные корректны. {username} – имя пользователя, {role} – права (Администратор, Фирма-поставщик, Клиент)

Изображение выглядит как текст, снимок экрана, Шрифт, Операционная система

Автоматически созданное описание

Рисунок 40. Сообщение “{username}, добро пожаловать в АвтоЗаправки с правами {role}!” при авторизации.

1. Форма администратора:

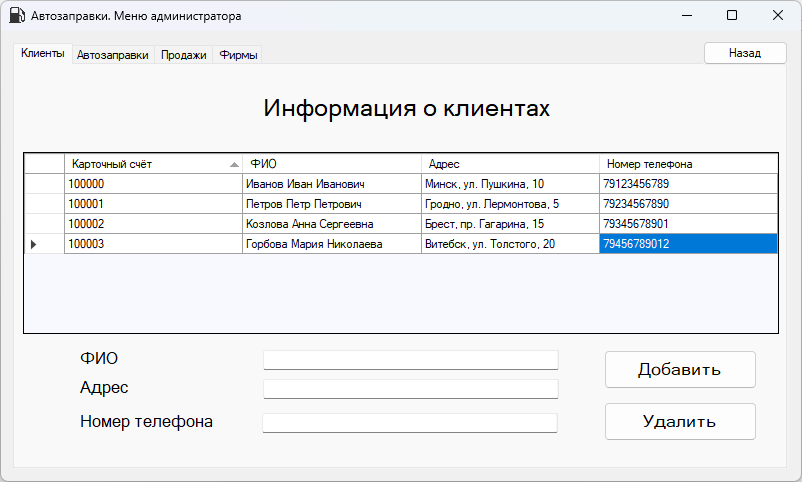


Рисунок 41. Форма администратора после авторизации.

Форма предназначена для просмотра, добавления, редактирования и удаления клиентов из базы данных.

Компоненты формы:

* 1. Кнопка «Назад». Нажатие на неё приводит к закрытию данной формы и открытию формы 1.
  2. Меню «Клиенты» (открыто по умолчанию, рисунок 41)
     1. Информация о клиентах. Состоит из четырёх полей «Карточный счёт», «ФИО», «Адрес» «Номер телефона». В строках отражаются соответствующие строки таблицы «Customer» базы данных.
     2. Поля ФИО, Адрес, Номер телефона доступные для заполнения.
     3. Кнопка «Добавить».
     4. Кнопка «Удалить».
     5. При двойном нажатии на ячейки «ФИО», «Адрес», «Номер телефона» появляется возможность редактирования.
     6. Поле карточный счёт недоступно для редактирования.
  3. Меню «Автозаправки»

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 42. Меню “Автозаправки”.

Информация об автозаправках. Состоит из трёх полей: “Код автозаправки”, “Адрес автозаправки”, “Название фирмы”.

* + 1. Поле код автозаправки недоступно для редактирования.
    2. В поле название фирмы можно вводить как текстовое название, так и ID существующей фирмы.
    3. Поле “Адрес”, необходимое для добавления автозаправки.
    4. Поле “Фирма-поставщик”, необходимое для добавления автозаправки.
    5. Для удаления автозаправки необходимо выбрать соответствующую строку и подтвердить удаление.
  1. Меню «Продажи»

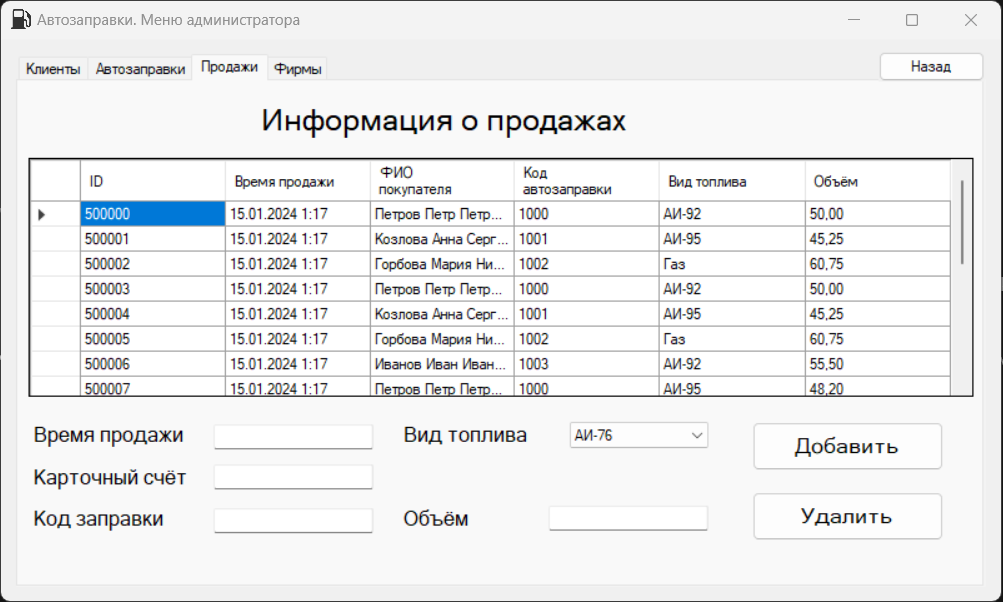


Рисунок 43. Меню “Продажи”.

Информация о продажах. Состоит из шести полей: “ID продажи”, “Время продажи”, “ФИО покупателя”, “Код автозаправки”, “Вид топлива”, “Объём”.

* + 1. Никакое поле недоступно для редактирования.
    2. Кнопка «Добавить» используется для добавления продажи.
    3. Кнопка «Удалить» используется для удаления продажи.
    4. Поле время продажи необязательно для заполнения. По умолчанию будет установлено текущее время. Остальные поля обязательны к заполнению и содержат в себе соответствующие проверки на данные.
  1. Меню «Фирмы»

Изображение выглядит как текст, снимок экрана, число, программное обеспечение

Автоматически созданное описание

Рисунок 44. Меню “Продажи”.

* + 1. Содержит в себе информацию о фирмах-поставщиках, которая была внесена разработчиком базы данных
    2. Поля доступны только для просмотра.

1. Форма Фирма-поставщик:

Изображение выглядит как текст, снимок экрана, программное обеспечение, число

Автоматически созданное описание

Рисунок 45. Форма пользователя “Фирма-поставщик”.

Компоненты формы:

* 1. Кнопка «Назад». Нажатие на неё приводит к закрытию данной формы и открытию формы 1.
  2. Меню «Виды топлива» (рисунок 44, открыто по умолчанию)
     1. Поле цена доступно для редактирования
     2. Кнопка «Добавить», используется для добавления нового вида топлива.
     3. Кнопка «Удалить», используется для удаления существующего вида топлива.
     4. Выпадающий список (АИ-76, АИ-92, АИ-95, АИ-96, Дизель, Газ) для добавления нового виды топлива.
     5. Поле цена для установки цены для нового вида топлива.
  3. Меню «Автозаправки»

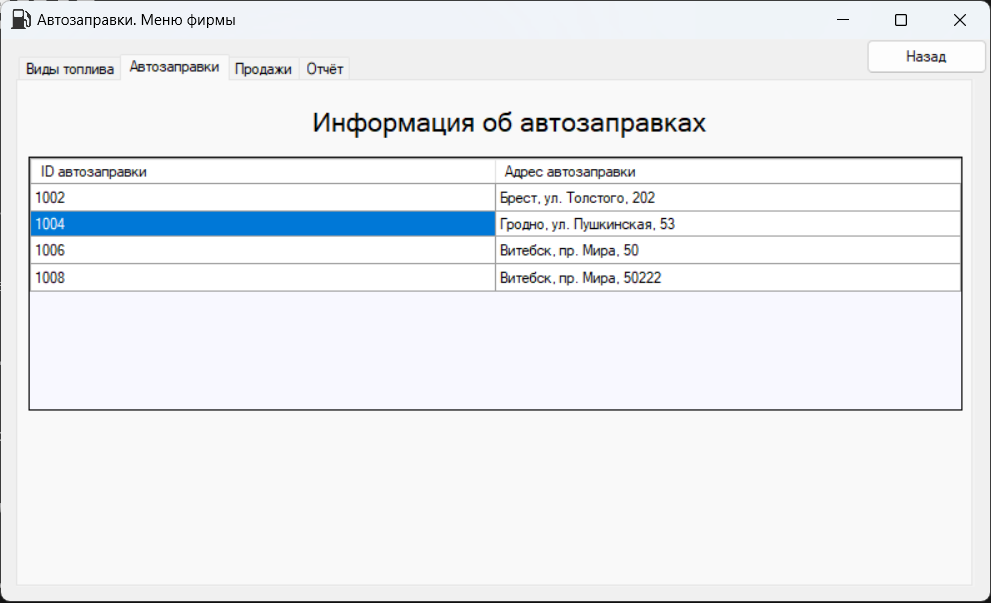


Рисунок 46. Меню «Автозаправки» пользователя “Фирма-поставщик”.

* + 1. Содержит информацию об автозаправках фирмы (ID, адрес).
    2. Поля доступны только для просмотра.
  1. Меню «Продажи»

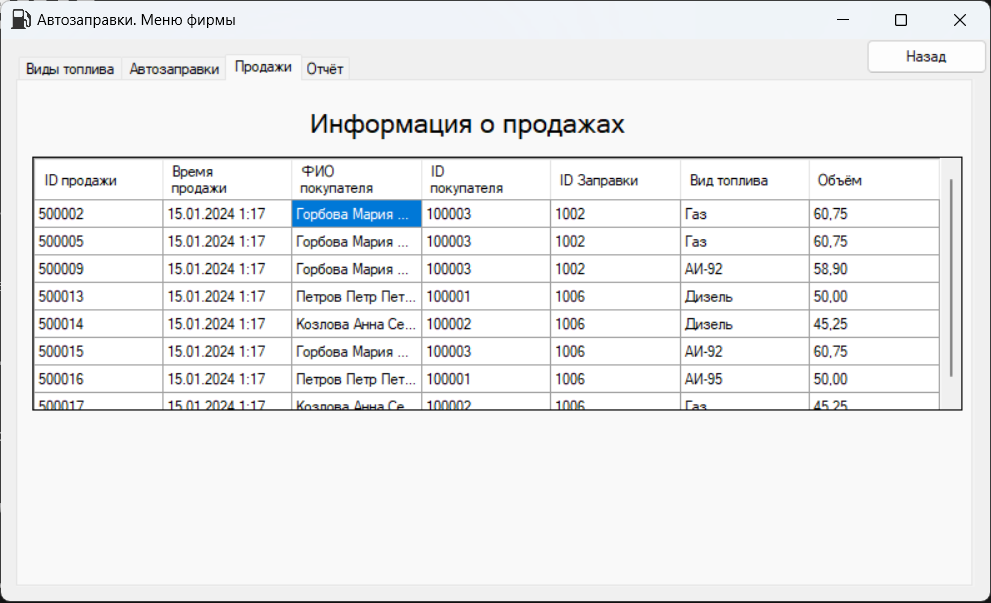


Рисунок 47. Меню «Продажи» пользователя “Фирма-поставщик”.

* + 1. Содержит информацию о продажах конкретной фирмы. Содержит поля “ID продажи”, “Время продажи”, “ФИО покупателя”, “ID покупателя”, “ID заправки”, “Вид топлива”, “Объём”.
    2. Поля недоступны для изменения, возможен только просмотр.
  1. Меню «Отчёт»

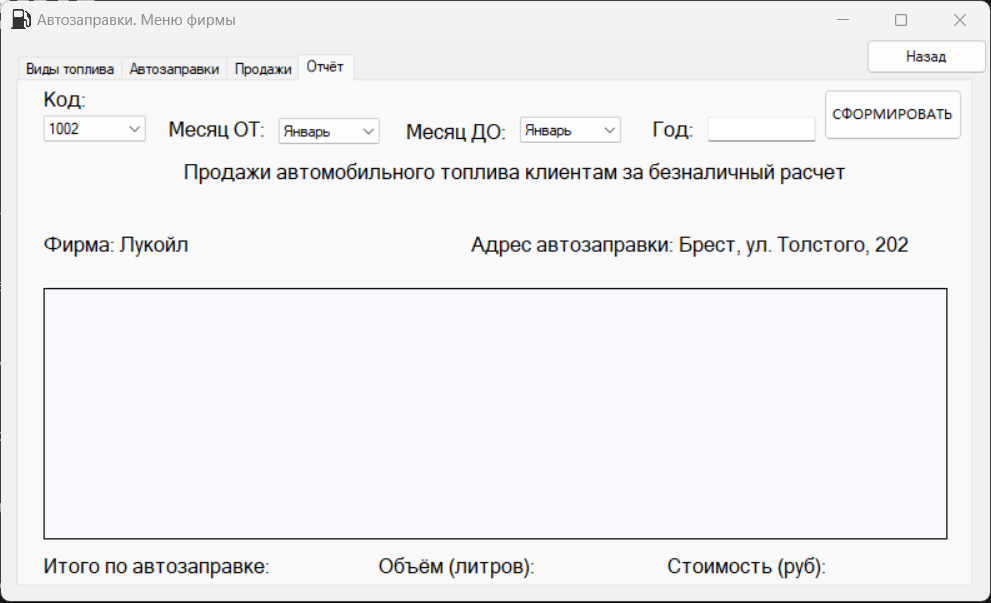


Рисунок 48. Меню «Отчёт» пользователя “Фирма-поставщик”.

* + 1. Меню служит для формирования отчёта.
    2. В поле код указывается код автозаправки, для которой необходимо сформировать отчёт.
    3. Заполняются поля “Месяц ОТ”, “Месяц ДО”, “Год”.
    4. После чего нажимается кнопка «Сформировать».

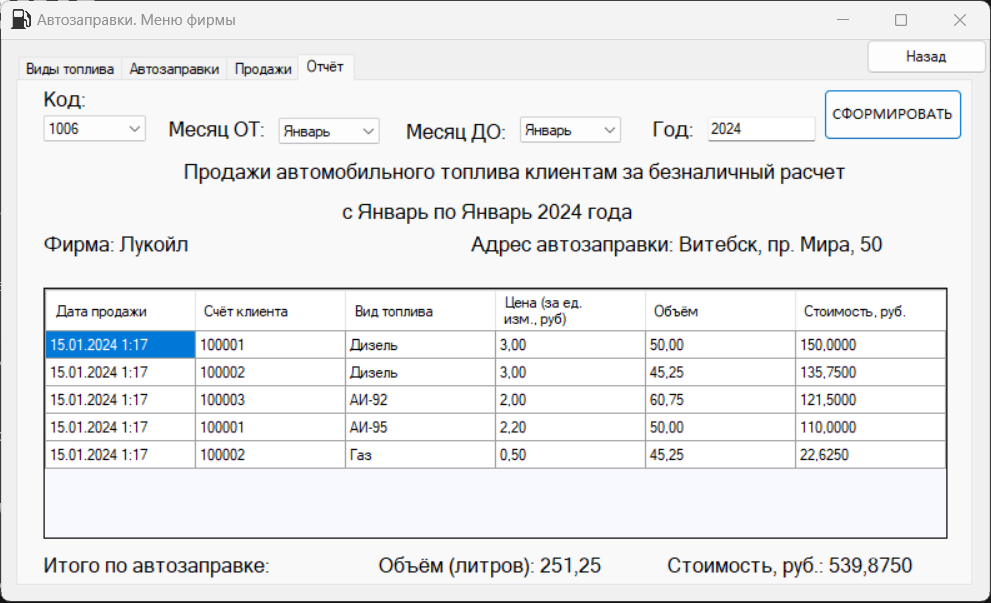


Рисунок 49. Пример сформированного отчёта пользователя “Фирма-поставщик”.

1. Форма «Пользователь»

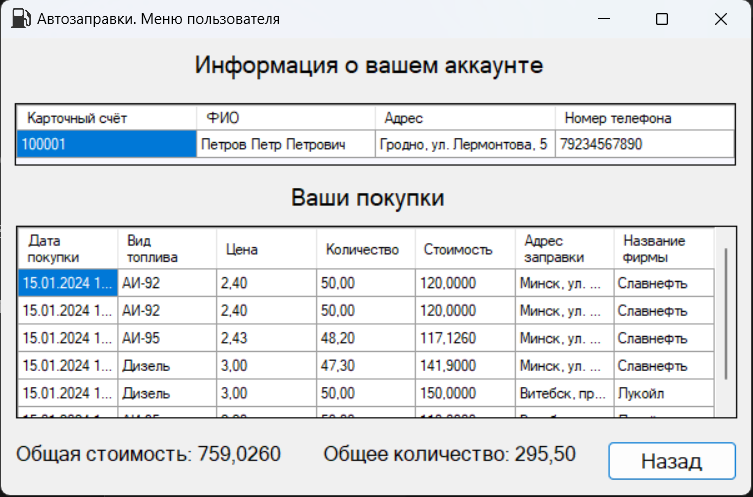


Рисунок 50. Форма пользователя “Пользователь”.

* 1. Содержит в себе две табличные части:
     1. Табличная часть “Информация о вашем аккаунте” состоит из полей “Карточный счёт”, “ФИО”, “Адрес”, “Номер телефона”. Поля доступны только для просмотра и представляют информацию об аккаунте конкретного пользователя.
     2. Табличная часть “Ваши покупки” состоит из семи полей: “Дата покупки”, “Вид топлива”, “Цена”, “Количество”, “Стоимость”, “Адрес заправки”, “Название фирмы”. Поля доступны только для просмотра и представляют информацию о покупках конкретного пользователя, причём если пользователь ещё не совершал покупок, то будет выведена соответствующая информация (рисунок 50)

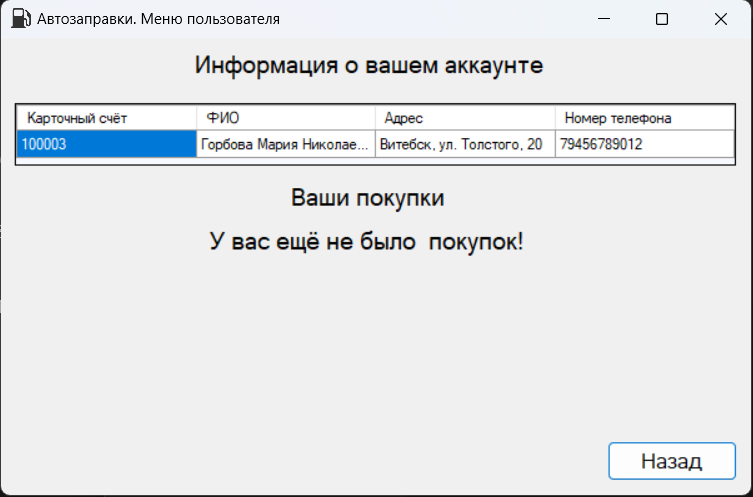


Рисунок 51. Форма пользователя “Пользователь”, который ещё не совершал покупок.

**Граф сценария диалога**

Исходя из вышеописанных форм можно составить граф сценария диалога, представленный на рисунке 52.

Изображение выглядит как зарисовка, текст, рисунок, диаграмма

Автоматически созданное описание

Рисунок 52. Граф сценария диалога

Каждой форме в приведённом выше сценарии сопоставлен номер

1. Форма «Авторизация»
2. Форма «Администратор»

* Кнопка «Назад»
  1. Меню «Клиенты»
  2. Меню «Автозаправки»
  3. Меню «Продажи»
  4. Меню «Фирмы»

1. Форма «Фирма-поставщик»

* Кнопка «Назад»
  1. Меню «Виды топлива»
  2. Меню «Автозаправки»
  3. Меню «Продажи»
  4. Меню «Отчёт»

1. Форма «Пользователь»

* Кнопка «Назад»
  1. Меню «Покупок нет»
  2. Меню «Есть покупки»

**Запросы**

Для эффективной работы с базой данных в среде СУБД MySQL Workbench были созданы следующие запросы.

1. **Запросы на получение всех записей объекта базы данных**
   1. **Для объектов, состоящих из одной таблицы**

Для получения всех записей из таблиц «Клиент», «Фирма», «Автозаправка», «Продажа топлива», «Топливо», используется следующий запрос:

Листинг 1 – Запрос на получение всех записей из таблицы

|  |
| --- |
| SELECT \* FROM <table\_name> |

* table\_name – название таблицы, принимает значения customer, firm, gasstation, fuelsale, fuel, для таблиц «Клиент», «Фирма», «Автозаправка», «Продажа топлива», «Топливо» соответственно.
  1. **Для объектов, состоящих из нескольких таблиц**

Так как объекты «Продажа топлива», «Автозаправки» включают в себя внешние ключи, то для получения записей из этих таблиц используются следующие запросы:

Для «Продажа топлива»:

Листинг 2 – Запрос на получение записей из таблиц “Пользователь” и “Автозаправки” для отображения продаж

|  |
| --- |
| SELECT  FuelSale.sale\_id as 'ID',  FuelSale.sale\_date as 'Время продажи',  Customer.customer\_full\_name as 'ФИО покупателя',  FuelSale.gasStation\_id as 'Код автозаправки',  Fuel.fuel\_type as 'Вид топлива',  FuelSale.quantity as 'Объём'  FROM  FuelSale  JOIN  Customer ON FuelSale.card\_account = Customer.card\_account  JOIN  GasStation ON FuelSale.gasStation\_id = GasStation.gasStation\_id  JOIN  Fuel ON FuelSale.fuel\_id = Fuel.fuel\_id  ORDER BY  FuelSale.sale\_id; |

Для «Автозаправки»

Листинг 3 – Запрос на получение записей из таблицы Автозаправки

|  |
| --- |
| SELECT  GasStation.gasStation\_id as 'Код автозаправки',  GasStation.gasStation\_address as 'Адрес автозаправки',  Firm.firm\_name as 'Название фирмы'  FROM  GasStation  JOIN  Firm ON GasStation.firm\_id = Firm.firm\_id  ORDER BY  GasStation.gasStation\_id; |

1. **Запросы на получение определенной записи**
   1. **Для объектов, состоящих из одной таблицы**

Для получения одной определённой записи из таблиц «Клиент», «Фирма», «Автозаправка», «Продажа топлива», «Топливо», используется следующий запрос:

Листинг 4 – Запрос на получение определённой записи из таблицы

|  |
| --- |
| SELECT \* FROM <table\_name>  WHERE <primary\_field> = <primary\_value> |

* table\_name – название таблицы, принимает значения customer, firm, gasstation, fuelsale, fuel, для таблиц «Клиент», «Фирма», «Автозаправка», «Продажа топлива», «Топливо» соответственно.
* primary\_field – название ключевого поля таблицы, принимает значения card\_account, firm\_id, gasstation\_id, fuelsale\_id, fuel\_id для таблиц «Клиент», «Фирма», «Автозаправка», «Продажа топлива», «Топливо» соответственно.
* primary\_value – ключевое значение, по которому происходит поиск.
  1. **Для объектов, состоящих из нескольких таблиц**

Листинг 5 – Запрос на получение объёма и стоимости для отображения данных клиенту

|  |
| --- |
| SELECT  SUM(fs.quantity) AS 'Общее количество',  SUM(f.fuel\_price \* fs.quantity) AS 'Общая стоимость'  FROM  FuelSale fs  JOIN  Fuel f ON fs.fuel\_id = f.fuel\_id  WHERE  fs.card\_account = @cardAccount; |

* + @cardAccount: Счет клиента. Фильтрует данные по полю card\_account в таблице FuelSale.

1. **Запросы для получения итогового отчёта**

Чтобы получить все продажи по определенной заправке за выбранный период, используется следующий запрос:

Листинг 6 – Запрос на формирование итогового отчёта по выбранной заправке определённой фирмы.

|  |
| --- |
| SELECT  fuelsale.sale\_date AS 'Дата продажи',  fuelsale.card\_account AS 'Счёт клиента',  fuel.fuel\_type AS 'Вид топлива',  fuel.fuel\_price AS 'Цена (за ед. изм., руб)',  fuelsale.quantity AS 'Объём',  (fuel.fuel\_price \* fuelsale.quantity) AS 'Стоимость, руб.'  FROM  fuelsale  JOIN  fuel ON fuelsale.fuel\_id = fuel.fuel\_id  WHERE  fuelsale.gasStation\_id = @gasStationId  AND MONTH(fuelsale.sale\_date) BETWEEN @monthFrom AND @monthTo  AND YEAR(fuelsale.sale\_date) = @year; |

* @gasStationId: Идентификатор автозаправки. Фильтрует данные по полю gasStation\_id в таблице fuelsale.
* @monthFrom: Начальный месяц периода для фильтрации по полю sale\_date.
* @monthTo: Конечный месяц периода для фильтрации по полю sale\_date.
* @year: Год для фильтрации по полю sale\_date.

Листинг 7 – Запрос на получение итоговых значени для полей объём и общая сумма

|  |
| --- |
| SELECT  SUM(fuelsale.quantity) AS 'Объём',  SUM(fuel.fuel\_price \* fuelsale.quantity) AS 'Стоимость, руб.'  FROM  fuelsale  JOIN  fuel ON fuelsale.fuel\_id = fuel.fuel\_id  WHERE  fuelsale.gasStation\_id = @gasStationId  AND MONTH(fuelsale.sale\_date) BETWEEN @startMonth AND @endMonth  AND YEAR(fuelsale.sale\_date) = @year; |

* @gasStationId: Код автозаправки. Фильтрует данные по полю gasStation\_id в таблице fuelsale.
* @startMonth: Начальный месяц для фильтрации по дате продажи.
* @endMonth: Конечный месяц для фильтрации по дате продажи.
* @year: Год для фильтрации по дате продажи.

1. **Запросы на добавление записи**

Для добавления новой записи в таблицы «Клиент», «Фирма», «Автозаправка», «Продажа топлива», «Топливо» используется следующий запрос:

Листинг 8 – Запрос на добавление записи в таблицу

|  |
| --- |
| INSERT INTO <table\_name>  VALUES (<values>) |

* table\_name – название таблицы, принимает значения customer, firm, gasstation, fuelsale, fuel, для таблиц «Клиент», «Фирма», «Автозаправка», «Продажа топлива», «Топливо» соответственно.
* values – значения добавляемой записи, перечисленные через запятую.

1. **Запросы на удаление записей**

Для удаления записей из таблиц «Клиент», «Фирма», «Автозаправка», «Продажа топлива», «Топливо» используется следующий запрос:

Листинг 9 – Запрос на удаление записи из таблицы

|  |
| --- |
| DELETE FROM <table\_name>  WHERE <primary\_field> = <primary\_value> |

* table\_name – название таблицы, принимает значения customer, firm, gasstation, fuelsale, fuel, для таблиц «Клиент», «Фирма», «Автозаправка», «Продажа топлива», «Топливо» соответственно.
* primary\_field – ключевое поле таблицы.
* primary\_value – значение ключевого поля записи.

1. **Запросы на изменение записей**

Для изменения записей в таблицах «Клиент», «Фирма», «Автозаправка», «Продажа топлива», «Топливо» используется следующий запрос:

Листинг 10 – Запрос на изменение записи в таблице

|  |
| --- |
| UPDATE <table\_name>  SET <field> = <field\_value>, …  WHERE <primary\_field> = <primary\_value> |

* table\_name – название таблицы, принимает значения customer, firm, gasstation, fuelsale, fuel, для таблиц «Клиент», «Фирма», «Автозаправка», «Продажа топлива», «Топливо» соответственно.
* field – названия полей таблицы.
* field\_value – изменённые значения полей записи.
* primary\_field – ключевое поле таблицы.
* primary\_value – значение ключевого поля записи.

**4 Тестирование**

4.1 Тестирование формы «Авторизация»

Таблица 4.1.1 – Тестирование формы «Авторизация»

|  |  |  |
| --- | --- | --- |
| **Тестовая ситуация** | **Входные данные** | **Выходные данные** |
| Попытка авторизации при пустых полях «логин» и «пароль» | – |  |
| Попытка авторизации при пустом поле «логин» | –;123 |  |
| Попытка авторизации при пустом поле «пароль» | 123;– |  |
| Попытка авторизации при неверных полях «логин» и «пароль» | asdasd;asdasd |  |
| Авторизация в качестве администратора | admin;admin |  |
| Авторизация в качестве фирмы-поставщика | Лукойл;qwerty321! |  |
| Авторизация в качестве пользователя | 100003; qwerty123 |  |

*Окончание таблицы 4.1.1*

|  |  |  |
| --- | --- | --- |
| Попытка авторизации в качестве администратора с неверным паролем | admin;123 |  |
| Попытка авторизации в качестве фирмы-поставщика с неверным паролем | А-100;123 |  |
| Попытка авторизации в качестве пользователя с неверным паролем | 100000;123 |  |
| Очистка полей | admin;321 | –;– |
| Попытка закрытия СУБД | Нажатие на кнопку «закрыть» | Завершение работы СУБД |

4.2 Тестирование формы «Администратор»

Таблица 4.2.1 – Тестирование меню «Клиенты» формы «Администратор»

|  |  |  |
| --- | --- | --- |
| **Тестовая ситуация** | **Входные данные** | **Выходные данные** |
| Добавление клиента с корректным ФИО | Горбова Мария Николаевна | Пользователь добавлен |
| Обязательное поле ФИО незаполнено | – |  |
| ФИО заполнено корректно, некорретно заполнен адрес | Горбова Мария Николаевна;2132131 |  |
| Добавление повторяющегося номера телефона | 79123456789 |  |

*Продолжение таблицы 4.2.1*

|  |  |  |
| --- | --- | --- |
| Добавление некорректного номера телефона | фыв |  |
| Добавление клиента с заполнением ФИО и адреса | Горбова Мария Николаевна;  Гродно, ул. Лермонтова, 5 |  |
| Добавление клиента с заполнением всех полей | Горбова Мария Николаевна;  Гродно, ул. Лермонтова, 5;  79123456789 | Пользователь добавлен |
| Изменение ФИО на некорректное значение | 123 |  |
| Изменение адреса на некорректное значение | 123 |  |
| Изменение номера телефона на некорректное значение | 123 |  |
| Изменение номера телефона на существующий в базе | 79123456789 |  |
| Изменение ФИО с корректными данными без отчества | Иванов Иван |  |
| Изменение ФИО с корректными данными | Иванов Иван Иванович |  |

*Окончание таблицы 4.2.1*

|  |  |  |
| --- | --- | --- |
| Изменение адреса с корректными данными | Гродно, ул. Лермонтова, 5 |  |
| Удаление пользователя без покупок | 100004 | Пользователь удалён |
| Попытка удаления пользователя без выбора пользователя | – |  |
| Удаление пользователя со связанными покупками | 100001 | Пользователь удалён |

Таблица 4.2.2 – Тестирование меню «Автозаправки» формы «Администратор»

|  |  |  |
| --- | --- | --- |
| **Тестовая ситуация** | **Входные данные** | **Выходные данные** |
| Добавление заправки с пустыми полями Адрес и Фирма-поставщик | –;– |  |
| Поле Адрес заполнено некорректными данными | фвыв; – |  |

*Продолжение таблицы 4.2.2*

|  |  |  |
| --- | --- | --- |
| Поле Адрес заполнено корректными данными, поле фирма незаполнено | Гродно, ул. Пушкинская, 531; – |  |
| Попытка добавления заправки на существующий адрес | Гродно, ул. Пушкинская, 531; |  |
| Добавление заправки с корректными данными (фирма указана через firm\_id) | Гродно, ул. Пушкинская, 531; 1 |  |
| Добавление заправки с корректными данными (фирма указана через название) | Гродно, ул. Пушкинская, 531; Лукойл |  |
| Добавление заправки с некорректным номером фирмы (указано через ID) | Гродно, ул. Пушкинская, 531; 6 |  |
| Добавление заправки с некорректным номером фирмы (указано через название) | Гродно, ул. Пушкинская, 531; РОСНЕФТЬ |  |
| Попытка изменения адреса заправки на существующий | Гродно, ул. Пушкинская, 531 |  |
| Попытка изменения названия фирмы на несуществующую (через ID) | 6 |  |

*Окончание таблицы 4.2.2*

|  |  |  |
| --- | --- | --- |
| Попытка изменения названия фирмы на несуществующую (через название) | РОСНЕФТЬ |  |
| Изменение адреса автозаправки с корректными данными | Минск, ул. Строителей, 10 |  |
| Изменение фирмы автозаправки с корректными данными (через название) | Славнефть |  |
| Изменение фирмы автозаправки с корректными данными (через ID) | 2 |  |
| Удаление автозаправки, с которой не связаны продажи топлива | 1012 |  |
| Удаление автозаправки, с которой связаны продажи топлива | 1001 |  |
| Автозаправка для удаления не выбрана | – |  |

Таблица 4.2.3 – Тестирование меню «Продажи» формы «Администратор»

|  |  |  |
| --- | --- | --- |
| **Тестовая ситуация** | **Входные данные** | **Выходные данные** |
| Добавление продажи без указания кода заправки | – |  |
| Добавление продажи без указания карточного счёта | – |  |
| Добавление продажи с указанием некорректного карточного счёта | фывввфыв |  |
| Добавление продажи с указанием некорретного времени | 32 234 324 |  |
| Добавление продажи с несуществующим карточным счётом | 1 |  |
| Добавление продажи с несуществующим номером заправки | – |  |
| Попытка добавления продажи с видом топлива, которое не предоставляет фирма | АИ-76; 1006 |  |
| Добавление корректных данных без указания времени продажи | 100000; 1006; АИ-95;1 |  |

*Окончание таблицы 4.2.3*

|  |  |  |
| --- | --- | --- |
| Значение объём не заполнено | – |  |
| Значение объём равно или меньше 0 | 0 |  |
| Добавление корректных данных с указанием времени продажи | 23.10.2004 19:39; 100000; 1006;АИ-95;1 |  |
| Удаление продажи топлива | 500014 |  |
| Попытка удаления продажи топлива без выбора продажи | – |  |

4.3 Тестирование формы «Фирма-поставщик»

Таблица 4.3.1 – Тестирование меню «Виды топлива» формы «Фирма»

|  |  |  |
| --- | --- | --- |
| **Тестовая ситуация** | **Входные данные** | **Выходные данные** |
| Добавление топлива с пустым полем цена | АИ-76;– |  |

*Продолжение таблицы 4.3.1*

|  |  |  |
| --- | --- | --- |
| Добавление дубликата топлива | АИ-76;1 |  |
| Добавление числа с 4 знаками в целой части в цену топлива | 1000 |  |
| Удаление топлива, связанного с продажами | АИ-95 |  |
| Удаление топлива, не связанного с продажами | Дизель |  |
| Добавление корректного значения топлива | АИ-96;100,23 |  |
| Попытка изменения цены не некорректное значение | 10000 |  |

*Окончание таблицы 4.3.1*

|  |  |  |
| --- | --- | --- |
| Попытка изменения цены не некорректное значение | 0,903025 |  |

Таблица 4.3.2 – Тестирование меню «Отчёт» формы «Фирма»

|  |  |  |
| --- | --- | --- |
| **Тестовая ситуация** | **Входные данные** | **Выходные данные** |
| Попытка сформировать отчёт без указания года | – |  |
| Попытка сформировать отчёт указав неверный год | фыввф |  |
| Формирование отчёта с пустыми продажами | Год: 2003 |  |
| Формирование отчёта с марта по январь | Месяц ОТ: март Месяц ДО: январь |  |
| Формирование непустого отчёта | Январь 2024 |  |

*Окончание таблицы 4.3.2*

|  |  |  |
| --- | --- | --- |
| Формирование отчёта после изменения цены на один из видов топлива | Январь 2024  Газ: 1->100 |  |

**Заключение**

В ходе курсовой работы были решены следующие задачи:

1. Проанализирована и описана предметная область. Были выделены термины, их смысл, основные задачи предметной области.
2. Построена инфологическая модель, которая является абстракцией данных, отражающей их взаимосвязи и структуру в предметной области.
3. Создана логическая модель в рамках конкретной системы управления базами данных (MySql Workbench), проведено даталогическое проектирование, определены таблицы, поля, связи между ними.
4. База данных была успешно реализована в MySql Workbench с учетом всех определенных требований и структур данных.
5. Проведено комплексное тестирование реализованной базы данных, включая проверку на соответствие функциональным требованиям и корректность хранения данных. Всего было рассмотрено 73 тестовые ситуации.

В результате успешного выполнения данного проекта можно утверждать, что спроектированная и реализованная база данных в MySql Workbench в сочетании с интерфейсом на C# и WinForms обеспечивает эффективное управление учетом продаж и анализом спроса на топливо.

Разработанное решение отвечает поставленным задачам и требованиям, предоставляя пользователям интуитивно понятный интерфейс и надежное хранение данных.

База данных была спроектирована с расчётом на расширение и дополнение, следовательно, в дальнейшем можно автоматизировать некоторые другие процессы данной предметной области.

**Список литературы**

1. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение с использованием MySQL / Т. Коннолли, К. Бегг. — М.: Вильямс, 2006. — 800 с.
2. Лафоре, Р. MySQL. Руководство по изучению языка / Р. Лафоре. — М.: Питер, 2005. — 528 с.
3. Харрингтон, Д. MySQL. Справочник программиста / Д. Харрингтон. — СПб.: БХВ-Петербург, 2007. — 416 с.
4. Трейнор, Ш. C# 6.0 и платформа .NET 4.6 для профессионалов / Ш. Трейнор. — М.: Вильямс, 2016. — 800 с.
5. Троелсен, А. C# 7.0 и платформа .NET Core: новые возможности / А. Троелсен. — М.: ДМК Пресс, 2018. — 896 с.
6. Шилдт, Г. C# 8.0. Полное руководство / Г. Шилдт. — М.: ДМК Пресс, 2019. — 640 с.
7. Макдональд, М. Программирование на C# в примерах и задачах / М. Макдональд. — М.: Питер, 2011. — 816 с.
8. Герберт, Ш. Программирование на C# 5.0 / Ш. Герберт. — СПб.: Питер, 2013. — 864 с.
9. Томпсон, Д. WinForms в .NET 5 для профессионалов / Д. Томпсон. — М.: Питер, 2021. — 656 с.
10. Горбачев, А.А. Создание Windows-приложений на платформе Microsoft .NET: Учебное пособие / А.А. Горбачев. — М.: ДМК Пресс, 2015. — 400 с.