МИНИСТЕРСТВО НАУКИ и высшего образования РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кафедра систем сбора и обработки данных



**КУРСОВАЯ РАБОТА**

**по дисциплине: «**Базы данных**»**

**на тему:** **«**Телефонная компания**»**

Выполнил:Проверил:

Студент гр. АТ-14 К.т.н., доцент

АВТФ Щетинин Юрий Иванович

Черкасов Антон Юрьевич

«\_\_» марта 2024г.«\_\_» марта 2024 г.

Новосибирск  
2024

**Оглавление**

1. Задание………………………………………………………………. 3
2. Введение……………………………………………………………...4
3. Разработка БД………………………………………………………..4
   1. Модель «сущность-связь»………………………………………4
   2. Нормализация отношений………………………………………6
   3. Структура базы данных……………………………………….....9

3.4.1 Схема данных………………………………………………9

3.4.2 Таблицы……………………………………………………12

* 1. Запросы…………………………………………………………..17
  2. Отчёты……………………………………………………………20
  3. Формы и интерфейс пользователя……………………………...21

1. Заключение…………………………………………………………...28
2. Литература……………………………………………………………29

**1.Задание**

# ЗАДАНИЕ № 12

Разработать базу данных « ***Телефонная компания***».

**Сведения:**

* Идентификатор клиента, тип клиента (физическое / юридическое лицо), ФИО клиента (для физ. лица), название клиента (для юр. лица), номер телефона, адрес установки телефона, номер и дата договора, номер лицевого счета, дополнительные сведения,
* Ежемесячный тариф клиента, дата и сумма оплаты по месяцам года, номер квитанции оплаты,
* Юридический адрес клиента (для юр. лица), ФИО руководителя, банк клиента ( для юр. лица), номер счета в банке,
* Дата и время заявки о поломке, фамилия и код лица, принявшего заявку, ФИО исполнителя, дата и время восстановления связи.

Перечисленный выше набор сведений может быть дополнен.

**Запросы:**

* Данные клиента с номером…
* Номер лицевого счета клиента…
* Заявки на ремонт, поступившие на дату …
* Сведения об оплате за месяц… клиента...

**Отчет:**

Данные о ремонте телефонных линий и аппаратов на дату…

**2.Введение**

Базы данных – это наборы информации, которые хранятся упорядоченно в электронном виде. Они позволяют организовывать работу крупных предприятий, хранить сведения об объектах материального мира и упорядочивать хаотичные сведения. Наибольшее применение базы данных нашли в организации работы компаний. Крупным предприятиям необходимо хранить и обрабатывать большое количество информации: будь то сведения о товарах или работающих сотрудниках. С помощью баз данных работа с информацией значительно упрощается.

В этой курсовой работе будет описана разработка реляционной базы данных телефонной компании, которая будет хранить информацию о клиентах компании, их счетах, заявках и оплате.

Для работы с базами данных используются системы управления базами данных. На рынке существует множество разных СУБД, но в этой курсовой работе будет использоваться Microsoft Access. Данная СУБД проста в использовании, надёжна и обладает достаточным функционалом для поставленной задачи.

**3.Разработка БД**

**3.1 Модель «сущность-связь»**

Модель «сущность-связь» представляет собой модель данных, позволяющую описывать концептуальные схемы. В этой модели объект материального мира представляется как “сущность”, а его связь с другими объектами материального мира, то есть “сущностями”, описывается “связями”.

Диаграмма «сущность-связь» с учётом атрибутов каждой сущности и связей между ними приведена на рис. 1.

Ключевые атрибуты сущностей подчеркнуты.

**Сущность «Клиенты».** Атрибуты: идентификатор клиента, тип клиента, ФИО клиента, название компании, номер телефона, адрес установки, номер договора, дата договора, номер лицевого счета, дополнительные сведения.

**Сущность «Оплата».** Атрибуты: код клиента, месяц, номер квитанции, дата оплаты, сумма оплаты, ежемесячный тариф.

**Сущность «Банковские счета».** Атрибуты: код клиента, ФИО руководителя, номер счёта, банк, юридический адрес клиента.

**Сущность «Ремонтные работы».** Атрибуты: код заявки, код клиента, дата заявки, время заявки, дата закрытия заявки, время закрытия заявки, ФИО исполнителя.

Связи между сущностями заключаются в следующем:

Все сущности связаны с сущностью **«Клиенты»**, но не связаны между собой, так как они независимы и не используют атрибуты друг друга. Связь осуществляется через идентификатор/код клиента и имеет тип *один ко многим*, так как у одного клиента может быть: несколько заявок на ремонт, несколько банковских счетов и множество оплат.

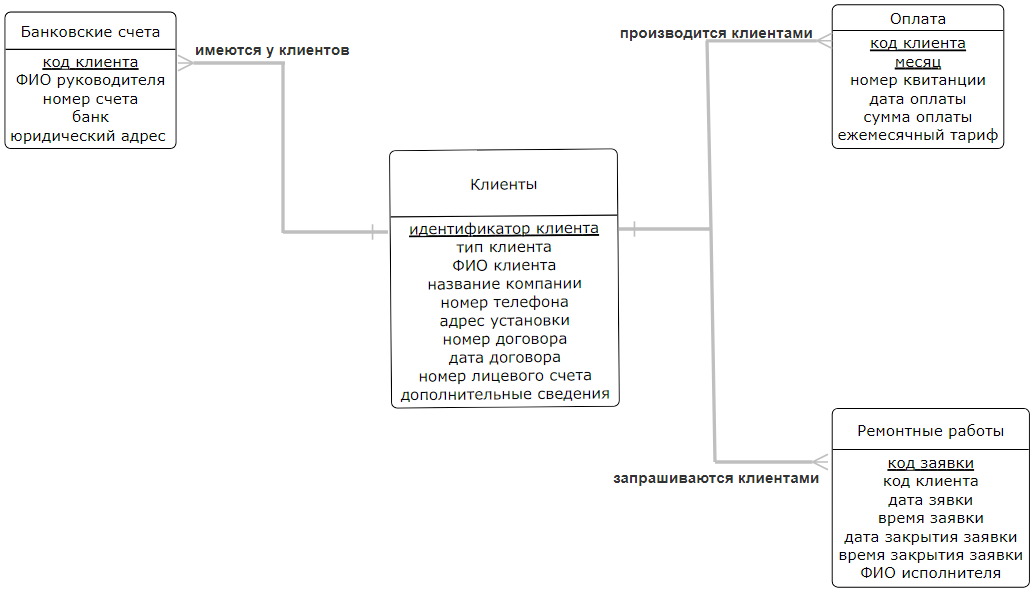


Рис. 1. Диаграмма «сущность-связь».

На диаграмме подчеркнутые атрибуты обозначают ключевые атрибуты сущностей, которые далее в базе данных будут использоваться в качестве первичных ключей, причем в таблице, отображающей оплату, ключ будет составным. Это сделано для обеспечения его уникальности.

Установленные наборы атрибутов для каждой сущности составляют фундамент таблиц будущей базы данных. Но эти атрибуты и сущности могут оказаться неподходящими для базы данных, в них может быть избыточность, что может привести к разного рода аномалиям и ошибкам в функционировании базы данных. Чтобы проверить наличие избыточности, и, при обнаружении, её минимизировать, используется нормализация.

**3.2 Нормализация отношений**

Нормализация – это процедура преобразования исходного отношения базы данных к виду требуемой нормальной формы отношений; процесс организации данных, включающий создание таблиц и установление отношений между ними в соответствии с правилами, которые обеспечивают защиту данных и делают базу данных более гибкой, устраняя избыточность и несогласованные зависимости.

Всего существует 6 нормальных форм отношений. Каждая последующая форма включает в себя условия предыдущей, то есть имея третью нормальную форму, по определению мы имеет и вторую, и первую.

Разберём отдельно каждую нормальную форму:

1. Первая нормальная форма (1НФ). Отношение находится в 1НФ, если все его атрибуты являются простыми (атомарными), и оно не имеет повторяющихся кортежей.
2. Вторая нормальная форма (2НФ). Отношение находится во 2НФ, если оно находится в 1НФ и каждый его неключевой атрибут зависит от полного ключа, а не от его частей. Другими словами, не должно быть функциональной зависимости неключевых атрибутов от подмножества ключа.
3. Третья нормальная форма (3НФ). Отношение находится в 3НФ, когда находится во 2НФ и в нём отсутствуют транзитивные зависимости не ключевых атрибутов от ключа. Если A → B и B → C, то говорят, что C транзитивно зависит от A через B.
4. Усиленная 3НФ, или нормальная форма Бойса-Кодда (НФБК). Отношение находится в НФБК, если оно находится в 3НФ и в нём отсутствуют зависимости атрибутов составного первичного ключа от не ключевых атрибутов. Другими словами, каждый детерминант отношения является потенциальным ключом.
5. Четвёртая нормальная форма (4НФ). Отношение находится в 4НФ, если оно находится в НФБК и все нетривиальные многозначные зависимости фактически являются функциональными зависимостями от ее потенциальных ключей.
6. Пятая нормальная форма (5НФ). Отношения находятся в 5НФ, если оно находится в 4НФ и отсутствуют сложные зависимые соединения между атрибутами. Другими словами, отношение в 5НФ – это такое отношение, в котором обратное соединение разделённых строк не приводит к появлению ложных новых строк. [1]

На практике 3НФ оказывается достаточно. Мы остановимся на 3НФ.

Проверим на соответствие условиям нормальных форм до третьей следующий получившийся из модели «сущность – связь» набор отношений (рис. 1):

В проектируемой базе данных некоторые составные атрибуты можно разложить на более простые составляющие, например атрибут «адрес» можно разложить на город, улицу, дом, квартиру или ФИО на фамилию, имя и отчество. Но в рамках создаваемой базы данных потребности в этом нет, поэтому будем считать атрибут «адрес» и «ФИО» атомарными. Тогда можно сделать вывод, что отношения находятся в 1НФ.

Теперь проверим отношения на 2НФ: все ключи, кроме 1, являются атомарными, а это значит, что у ключей нет подмножеств, а это значит, что нет и функциональных зависимостей не ключевых атрибутов от подмножеств ключа. Единственный неатомарный ключ у сущности «Оплата». Каждый неключевой атрибут зависит и от кода клиента и от месяца оплаты. Значит каждый . Делаем вывод, что отношения находятся в 2НФ.

Чтобы отношение было в 3НФ, необходимо отсутствие транзитивных зависимостей. В разрабатываемой базе данных таких зависимостей нет, а значит, все отношения находятся в 3НФ.

**3.3 Структура базы данных**

**3.3.1 Схема данных**

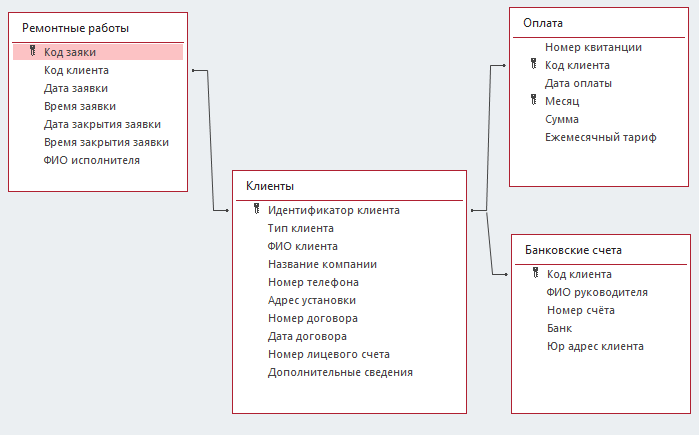
На рис. 2 представлена схема данных со всеми таблицами, их атрибутами и отношениями между ними.

Рис. 2. Схема данных.

**3.3.2 Таблицы**

Ниже приведена структура всех таблиц базы данных.

**Таблица «Клиенты»**

Хранит в себе информацию о клиентах телефонной компании.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Название поля* | *Тип данных* | *Индекс. поле* | *Обязат.*  *поле* | *Примечания* |
| Идентификатор клиента | Счетчик | Да | Да | Первичный ключ |
| Тип клиента | Короткий текст | Нет | Да | Подстановка из двух вариантов: физ.лицо и юр.лицо |
| ФИО клиента | Короткий текст | Нет | Да | Размер поля: 40 |
| Название компании | Короткий текст | Нет | Да | Размер поля: 40 |
| Номер телефона | Короткий текст | Нет | Да | Размер поля: 12 |
| Адрес установки | Короткий текст | Нет | Да | Размер поля: 60 |
| Номер договора | Короткий текст | Нет | Да | Размер поля: 10 |
| Дата договора | Дата и время | Нет | Да | Формат поля: краткий формат даты. |
| Номер лицевого счёта | Короткий текст | Нет | Да | Размер поля: 10 |
| Дополнительные сведения | Короткий текст | Нет | Нет | Размер поля: 255 |

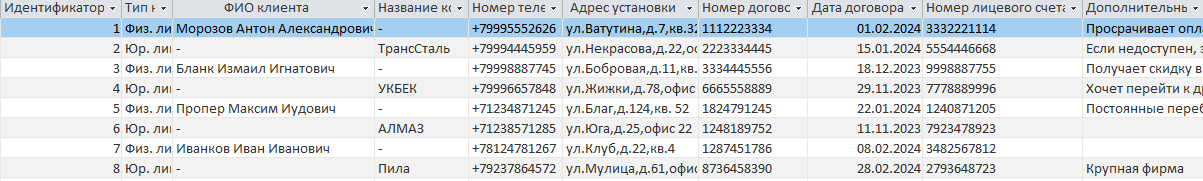
Табл. 1. Атрибуты таблицы «Клиенты».

Рис. 3. Таблица «Клиенты».

**Таблица «Оплата»**

Хранит в себе информацию об оплате ежемесячного тарифа.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Название поля* | *Тип данных* | *Индекс. поле* | *Обязат.*  *поле* | *Примечания* |
| Номер квитанции | Счетчик | Да | Да | Размер поля: длинное целое |
| Код клиента | Числовой | Нет | Да | Первичный ключ, внешний ключ |
| Дата оплаты | Дата и время | Нет | Да | Формат поля: краткий формат даты. |
| Месяц | Короткий текст | Нет | Да | Первичный ключ,  подстановка из 12 вариантов(месяцев) |
| Сумма | Денежный | Нет | Да |  |
| Ежемесячный тариф | Денежный | Нет | Да |  |

Табл. 2. Атрибуты таблицы «Оплата».

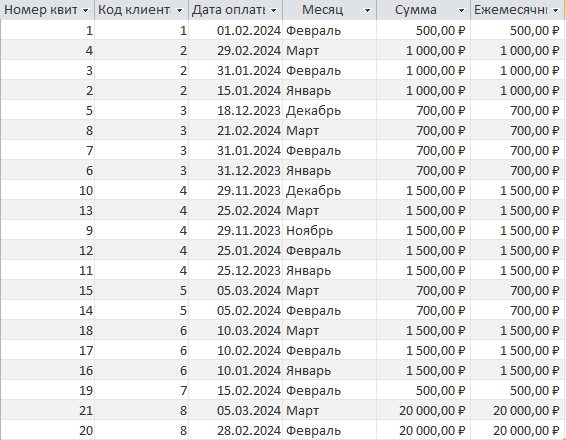


Рис. 4. Таблица «Клиенты».

**Таблица «Ремонтные работы»**

Хранит в себе информацию о заявках на ремонтные работы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Название поля* | *Тип данных* | *Индекс. поле* | *Обязат.*  *поле* | *Примечания* |
| Код заявки | Счетчик | Да | Да | Первичный ключ |
| Код клиента | Числовой | Нет | Да | Внешний ключ |
| Дата заявки | Дата и время | Нет | Да | Формат поля: краткий формат даты |
| Время заявки | Дата и время | Нет | Да | Формат поля: краткий формат времени |
| Дата закрытия заявки | Дата и время | Нет | Да | Формат поля: краткий формат даты |
| Время закрытия заявки | Дата и время | Нет | Да | Формат поля: краткий формат времени |
| ФИО исполнителя | Короткий текст | Нет | Да | Размер поля: 40 |

Табл. 3. Атрибуты таблицы «Ремонтные работы».

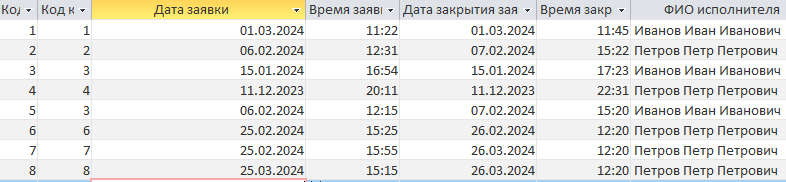


Рис. 5. Таблица «Ремонтные работы».

**Таблица «Банковские счета»**

Хранит в себе информацию о банковских счетах юридических лиц.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Название поля* | *Тип данных* | *Индекс. поле* | *Обязат.*  *поле* | *Примечания* |
| Код клиента | Счетчик | Да | Да | Первичный ключ |
| ФИО руководителя | Короткий текст | Нет | Да | Размер поля: 50 |
| Номер счёта | Короткий текст | Нет | Да | Размер поля: 10 |
| Банк | Короткий текст | Нет | Да | Размер поля: 15 |
| Юр адрес клиента | Короткий текст | Нет | Да | Размер поля:60 |

Табл. 4. Атрибуты таблицы «Банковские счета».

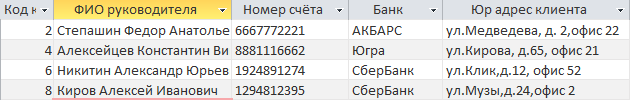


Рис. 6. Таблица «Банковские счета».

**3.4 Запросы**

Запрос – это обращение к БД для поиска или изменения в базе данных информации, соответствующей заданным критериям.

**Запрос «Заявки на дату»**

Запрос с параметрами. Выводит информацию о заявках. Параметр: дата.

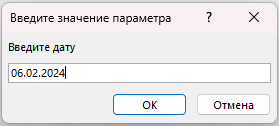
****

Рис. 7. Ввод параметра для запроса «Заявки на дату».



Рис. 8. Пример результата выполнения запроса «Заявки на дату».

**Запрос «Клиент по номеру лицевого счета»**

Запрос с параметрами. Выводит информацию о клиенте. Параметр: номер лицевого счета.

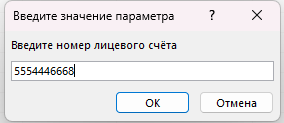
****

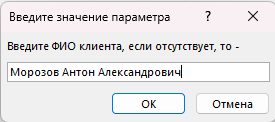
Рис. 9. Ввод параметра для запроса «Клиент по номеру лицевого счета».



Рис. 10. Пример результата выполнения запроса «Клиент по номеру лицевого счета».

**Запрос «Номер счёта по названию или фио»**

Запрос с параметрами. Выводит номер счёта клиента. Параметр: название или фио.

****

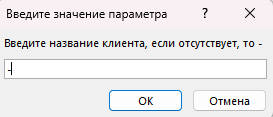
****

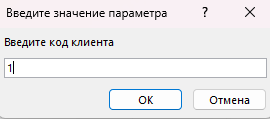
Рис. 11. Ввод параметра для запроса «Номер счета по названию или фио».



Рис. 12. Пример результата выполнения запроса «Номер счета по названию или фио».

**Запрос «Оплата за месяц от клиента»**

Запрос с параметрами. Выводит данные по оплате за месяц. Параметр: код и месяц.

****

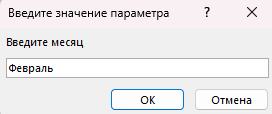
****

Рис. 13. Ввод параметра для запроса «Оплата за месяц от клиента».



Рис. 14. Пример результата выполнения запроса «Оплата за месяц от клиента».

**3.6 Отчёты**

Отчёт — это объект базы данных, который используется для вывода структурированной информации на экран, в печать или файл. Отчёты позволяют извлечь из таблиц или запросов базы данных необходимую информацию и представить ее в виде удобном для восприятия. В данной базе данных предусмотрен один отчет: “ Сведения об оплате по адресу за месяц текущего года.”

**Отчёт** «**Заявки на дату**»

Выводит информацию по заявкам на ремонтные работы за конкретный месяц. Основан на запросе «Заявки на дату».

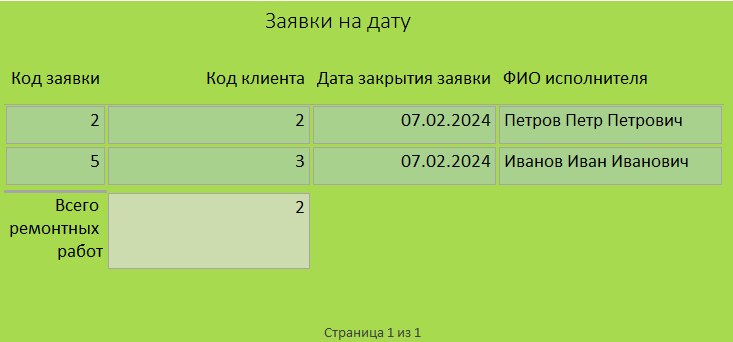


Рис. 15. Предварительный просмотр отчёта «Заявки на дату».

**Отчёт «Клиент по номеру лицевого счета»**

Выводит информацию по клиенту в зависимости от его номера лицевого счёта. Основан на запросе «Клиент по номеру лицевого счета».

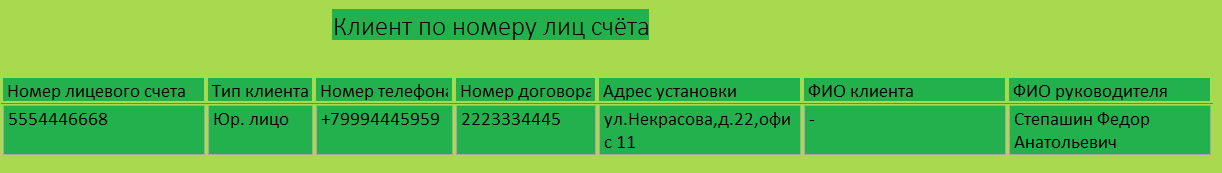


Рис. 16. Предварительный просмотр отчёта «Клиент по номеру лицевого счета».

**Отчёт «Номер счёта по названию или фио»**

Выводит номер счета клиента. Основан на запросе «Клиент по номеру лицевого счета».

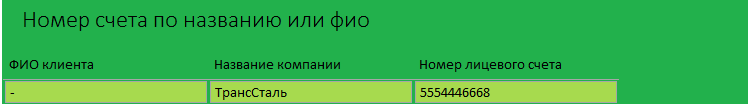


Рис. 17. Предварительный просмотр отчёта «Номер счета по названию или фио».

**Отчёт «Оплата за месяц от клиента»**

Выводит номер счета клиента. Основан на запросе «Клиент по номеру лицевого счета».

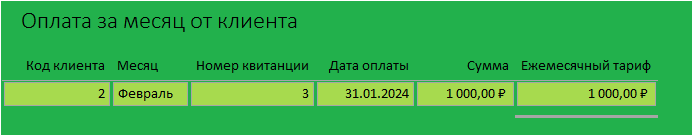


Рис. 18. Предварительный просмотр отчёта Рис. 10. Пример результата выполнения запроса «Оплата за месяц от клиента».

**3.7 Формы и интерфейс пользователя**

Форма – это окно с элементами управления, созданное для наглядного представления данных таблиц, ввода новых данных, редактирования и поиска записей в базе данных. На основе форм в базе данных реализуется пользовательский интерфейс. Таким образом, пользователю нет необходимости вдаваться в подробности, как работает эта база данных.

При открытии базы данных появляется заставка (рис. 14), которая закрывается через несколько секунд и перенаправляет пользователя к главной кнопочной форме.



Рис. 19. Форма-заставка.

**Главная страница кнопочной формы**

Открывающаяся главная кнопочная форма (рис. 20) имеет 6 кнопок: «Сведения о клиентах», «Все квитанции за год», «Заявки на ремонт», «Банковские счета юр.лиц». Эта форма является связующей для всех других оставшихся форм БД, а также запросов и отчётов. Далее, будет рассмотрена каждая из кнопок.



Рис. 20. Главная кнопочная форма.

**Кнопка “Основные сведения о клиентах”**

По нажатию на нее откроется всплывающее окно с формой. Данная форма представляет всю информацию о клиенте, которая есть у компании.



Рис. 21. “Основные сведения о клиентах”

**Кнопка “Все квитанции”**

По нажатию на нее откроется всплывающее окно с формой. Данная форма представляет всю информацию об ежемесячной оплате клиенте, произведенной клиентами.

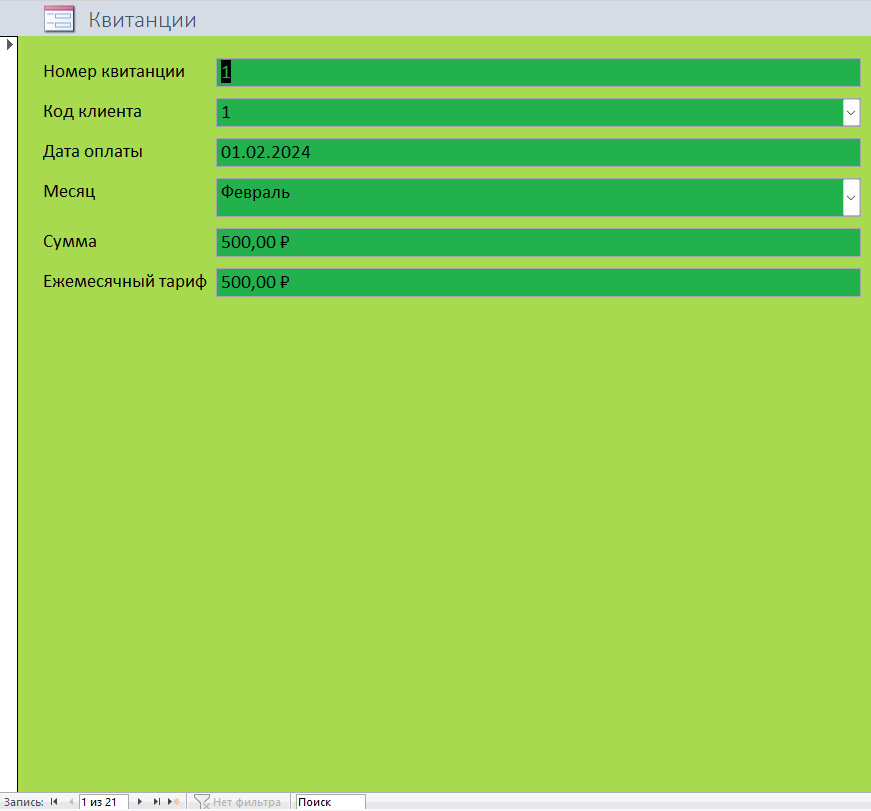


Рис. 22. “Все квитанции”

**Кнопка “Заявки на ремонт”**

По нажатию на нее откроется всплывающее окно с формой. Данная форма представляет всю информацию о заявках на ремонт от клиентов.



Рис. 23. “Заявки на ремонт”

**Кнопка “Банковские счета юр лиц”**

По нажатию на нее откроется всплывающее окно с формой. Данная форма представляет всю информацию о банковских счетах юридических лиц.

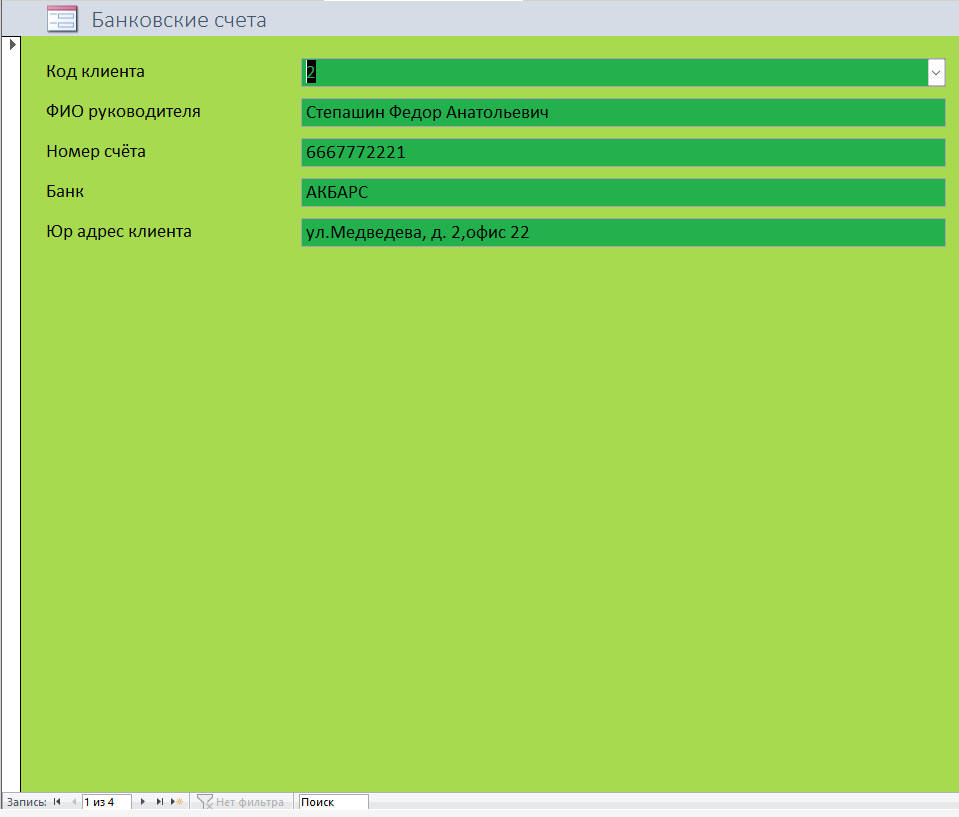


Рис. 24. “Банковские счета юр лиц”

**Кнопка “Запросы”**

По нажатию на нее откроется всплывающее окно с формой. Данная форма позволяет выполнить запросы разработанные ранее, которые будут выведены в виде одноименных отчетов.

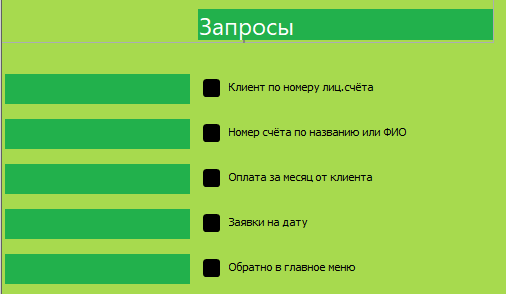


Рис. 25. “Банковские счета юр лиц”

**Кнопка “Выход”**

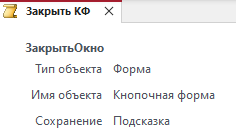
По нажатию на нее, вся кнопочная форма будет закрыта.

Рис. 26. Макрос для закрытия кнопочной формы.

**4. Заключение**

1. При выполнении данного задания была спроектирована база данных «Телефонная компания» при помощи СУБД Microsoft Office Access. В ходе разработки была получена диаграмма «сущность-связь». Первичный вид отношений был приведён к нормальным формам до 3НФ. Приведён состав каждого отношения, рассмотрены особенности всех полей таблиц. Таблицы заполнены данными. На основе полученных отношений были реализованы запросы, макросы, формы и отчёты. Создана главная кнопочная форма, связывающая все объекты базы данных воедино.

2. Созданная база данных соответствует техническому заданию. Все данные, требуемые в задании, хранятся в базе данных. Все необходимые запросы, отчёты и формы были созданы.

**5. Литература и другие источники информации**

1. Щетинин Ю. И. Базы данных: Конспект лекций. 2023г. Новосибирск, ул. Немировича-Данченко, 134, ssodserver/study/ Базы данных\_2023/Лекции

2. Вейскас Д.Э., Эффективная работа с Microsoft Access, Санкт – Петербург, «Питер Пресс», 2014 — 856 с.

3. Основные сведения об отчетах в Access, <https://support.microsoft.com/ru-ru/office/основные-сведения-об-отчетах-в-access-e0869f59-7536-4d19-8e05-7158dcd3681c>, 08.12.2023.

4. Добавление списка или списка со списком, <https://support.microsoft.com/ru-ru/office/добавление-списка-или-списка-со-списком-70abf4a9-0439-4885-9099-b9fa83517603>, 09.12.2021.

5. Создание макроса пользовательского интерфейса, <https://support.microsoft.com/ru-ru/office/создание-макроса-пользовательского-интерфейса-12590d3b-b326-4207-bfe5-19234f53f08b>, 10.10.2021.

6. Общие сведения о программировании в Access, <https://support.microsoft.com/ru-ru/office/общие-сведения-о-программировании-в-access-92eb616b-3204-4121-9277-70649e33be4f>, 05.12.2023

7. Создание формы, содержащей подчиненную форму, <https://support.microsoft.com/ru-ru/office/создание-формы-содержащей-подчиненную-форму-форма-с-отношением-один-ко-многим-ddf3822f-8aba-49cb-831a-1e74d6f5f06b>, 11.12.2022

8. Создание формы в Access, https://support.microsoft.com/ru-ru/office/создание-формы-в-access-5d550a3d-92e1-4f38-9772-7e7e21e80c6b, 28.12.2022