

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

«Севастопольский государственный университет»

Кафедра «Информационные технологии и компьютерные системы»

Пояснительная записка  
по комплексной практической работе  
по дисциплине  
«Проектирование баз данных»  
Вариант 8

Выполнил:  
студент группы: ИТ/б-22-б-о  
Донец Н.О.  
Проверил:  
доцент кафедры ИТиКС  
Ченгарь О.В.

Севастополь  
2024 г.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	3
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	4
1.1. Описание входных и выходных данных .....	5
1.2. Описание функций предметной области .....	7
1.3. Анализ зависимости функций предметной области.....	7
1.4. Анализ документов предметной области .....	8
2. ИНФОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ .....	10
2.1. Определение пула сущностей предметной области .....	10
2.2. Концептуальная модель связей сущности .....	12
2.3. Концептуальная модель уровня свойств сущностей .....	14
2.4. Определение классов принадлежности сущностей предметной области	16
2.5. Переход к логической модели базы данных .....	20
3. ДАТАЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ .....	25
3.1. Описание логической модели данных .....	25
3.2. Уточнение логической модели базы данных .....	28
3.3. Описание функциональных зависимостей .....	28
3.4. Нормализация отношений .....	36
4. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ .....	40
4.1. Общая схема и описание физической модели базы данных .....	40
4.2. Проектирование физической модели базы данных в выбранной СУБД .	46
5. ТЕСТИРОВАНИЕ БАЗЫ ДАННЫХ .....	57
5.1. Запросы на добавление данных в таблицы базы данных .....	57
5.2. Тестирование базы данных на соблюдение ссылочной целостности ....	61
5.3. Тестирование базы данных на возможность выборки из одного источника данных .....	67
5.4. Тестирование базы данных на возможность выборки из нескольких источников данных .....	71
5.5 SQL-запросы на создание пользовательских представлений.....	75
5.6. SQL-запросы на создание индексов .....	78
ЗАКЛЮЧЕНИЕ .....	81

СПИСОК ИСТОЧНИКОВ.....	82
ПРИЛОЖЕНИЕ А .....	84

## **ВВЕДЕНИЕ**

Целью данной комплексной практической работы является разработка базы данных для ежемесячного учета оплаты услуг за теплоснабжение в теплосети района. База данных также должна содержать следующие данные: адрес (район, улица, номера дома и квартиры), план и площадь квартиры, месячные тарифы за потребленное тепло из расчета за 1 квадратный метр площади, общая плата за услуги, сведения об уплате с указанием дат и сумм, показатели оплаты услуг за отчетное полугодие по домам.

## **1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

Районная организация теплоснабжения производит ежемесячный учёт уплаты средств за потреблённое тепло. Для этого учитывается следующая информация:

1. Адрес квартиры, который представляет собой физический адрес квартиры потребителя. Он используется для идентификации расположения квартир. Адрес состоит из района, улицы, номера дома, а также номера квартиры в случае многоквартирного дома.
2. ФИО владельца отапливаемого помещения, которое позволяет идентифицировать личность, ответственную за оплату потребляемой электроэнергии в квартире.
3. План и площадь квартиры предоставляют возможность вычислить количество потреблённого тепла, чтобы далее определить сумму платы за предоставленные услуги.
4. Месячные тарифы за потреблённое тепло определяют стоимость тепла за 1 квадратный метр площади дома или квартиры и используются для подсчёта суммы к оплате на основе потребления.
5. Общая плата за услуги рассчитывается на основе площади квартиры, месячного тарифа за потреблённое тепло, определяющего стоимость тепла за 1 квадратный метр площади помещения, а также норматива отопления за учётный период.
6. Сведения об уплате с указанием дат и сумм используются для формирования истории платежей и для расчета суммы долга потребителя на определенный момент времени.
7. Показатели оплаты услуг за отчетное полугодие по домам, который позволяет агрегировать информацию об уплате услуг в разных квартирах и домах и помогает анализировать показатели оплаты услуг в разных домах, районах и улицах, также может использоваться

для создания отчётов об уплате потребителями предоставленных услуг.

8. Начисление на потребителя, которое будет показывать какое количество денежных единиц потребитель должен компании за указанный период. Оно будет вычисляться с помощью используемого тарифа и площади, которой владеет потребитель.

### **1.1. Описание входных и выходных данных**

Входными и выходными данными системы учета уплаты за потреблённое тепло являются следующие данные:

Входные данные:

1. Информация о местонахождении помещения:
  - адрес дома;
  - номер квартиры;
  - лицевой счёт.
2. Информация о помещении:
  - план;
  - площадь.
3. Информация о потребителях:
  - ФИО (для физ.лица) или наименование (для юр.лица);
  - паспортные данные (для физ.лица);
  - СНИЛС (для физ.лица);
  - ОГРН (для юр.лица);
  - дата рождения (для физ.лица);
  - номер телефона;
  - электронная почта.
4. Информация о тарифах

- стоимость одного квадратного метра по тарифу;
- дата вступления тарифа в силу;
- дата окончания тарифа;
- название тарифа.

5. Информация о начислениях:

- тариф, по которому сделано начисление;
- потребитель, которому начисление адресовано;
- период оплаты;
- сумма начисления.

6. Информация о платежах:

- тариф, по которому сделано начисление;
- дата оплаты;
- сумма оплаты;

Выходные данные:

1. Сумма, подлежащая уплате, показывающая, какую сумму должен уплатить потребитель за потребленное тепло.
2. Сведения об уплате, содержащие информацию о датах и суммах всех платежей, сделанных потребителем.
3. Текущая сумма долга, показывающая текущую задолженность потребителя перед организацией теплоснабжения.
4. Общие показатели оплаты услуг за отчетное полугодие по домам, представляющий собой агрегированную информацию об уплате предоставленных услуг в разных домах.

## **1.2. Описание функций предметной области**

В процессе учета ежемесячной уплаты средств за потребленное тепло в районной организации теплоснабжения можно выделить следующие ключевые функции:

1. Учет потребителей. Должен вестись учет информации о каждом потребителе, включая адрес, ФИО, номер телефона, почту, а также паспортные данные.
2. Учет потребленного тепла. С учётом площади, которой владеет потребитель, будет вычислен объём потреблённого тепла.
3. Расчет стоимости потреблённого тепла. С учетом тарифа за один квадратный метр и объёма потреблённого тепла должен происходить расчет суммы к оплате.
4. Учет платежей. Платежи от потребителей должны регистрироваться и привязываться к начислению.
5. Агрегация показателей оплаты услуг. Данные об оплате услуг должны быть агрегированы по домам для формирования общих показателей оплаты услуг в домах.
6. Формирование отчетов и счетов. Должны формироваться счета для потребителей и отчеты о потреблении и платежах.

## **1.3. Анализ зависимости функций предметной области**

В процессе учета ежемесячной уплаты средств за потребленное тепло в районной организации теплоснабжения можно выделить следующие зависимости функций:

1. Функция расчета стоимости потребленного тепла зависит от функции учета потребленного тепла, так как в процессе расчета стоимости потребленного тепла используется объем потреблённого тепла за определенный период.
2. Функция учета платежей зависит от функции расчета стоимости



потребленного тепла, так как сумма для оплаты напрямую зависит от рассчитанной стоимости потребленного тепла за определенный период и от функции учета потребителей, так как каждый потребитель имеет свои счета и площади.

3. Функция агрегации показателей оплаты услуг зависит от функции учета платежей, так как использует информацию о платежах за определенный период, а также от функции учета потребителей, так как учитывает адрес квартиры потребителя для агрегации потребления по домам.

4. Функция формирования отчетов и счетов зависит от функции учета потребителей, так как формирует индивидуальные счета для каждого потребителя, от функции учета платежей, так как использует информацию о ранее внесенных платежах и итоговых начислениях на определенный момент времени, от функции учета потребленного тепла, от функции расчета стоимости потребленного тепла, так как счета включают в себя стоимость потребленного тепла за определенные периоды времени, а также от функции агрегации показателей оплаты услуг, так как формируемые отчеты об оплате услуг включают в себя информацию о суммарной оплате услуг по домам в районе.

#### **1.4. Анализ документов предметной области**

В процессе учета ежемесячной уплаты средств за потребленное тепло в районной организации теплоснабжения можно выделить следующие документы:

- 1) Счета на оплату. Данные счета формируются организацией теплоснабжения и отправляются потребителям. Они содержат информацию о потреблении тепла, тарифах, а также общую

сумму, подлежащую уплате. Счета на оплату служат основным средством связи между организацией теплоснабжения и потребителями, предоставляя им информацию о текущей задолженности и сроках оплаты.

- 2) Квитанции об оплате. Квитанции представляют собой документы, подтверждающие факт уплаты за потребленное тепло. Они содержат информацию о дате платежа, сумме и реквизитах плательщика. Квитанции об оплате служат как подтверждение оплаты для потребителей, так и доказательство для организации теплоснабжения о получении платежей.
- 3) Договоры об оказании услуг по подаче тепла. Такие договоры заключаются между организацией теплоснабжения и потребителями. Они содержат условия использования и оплаты тепла, тарифы, права и обязанности сторон.

## 2. ИНФОЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

### 2.1. Определение пула сущностей предметной области

Для рассматриваемой предметной области были выделены следующие множества сущностей:

1. Квартира
2. Потребитель
3. Физическое лицо
4. Юридическое лицо
5. Начисления
6. Тариф
7. Оплата

Для выделенных множеств сущностей была построена табличная модель уровня сущностей, представленная в таблице 2.1.

Таблица 2.1 – Табличная модель уровня сущностей

Номер множества сущностей	Имя множества сущностей	Определение множества сущностей	Описание множества сущностей
E1	Квартира	Жилое помещение в доме, имеющее собственный номер, площадь и лицевой счёт.	Как только заключается договор о теплоснабжении дома или квартиры, формируются новые экземпляры данного множества сущностей.
E2	Потребитель	Лицо, владеющее или использующее квартиру и заключившее с организацией теплоснабжения договор о поставке тепла.	Как только с потребителем заключается договор о поставке тепла, формируется новый экземпляр данного множества сущностей.

E3	Физическое лицо	Потребитель, человек, заключивший с организацией теплоснабжения договор о поставке тепла.	Как только с потребителем - физическим лицом заключается договор о поставке тепла, формируется новый экземпляр данного множества сущностей.
E4	Юридическое лицо	Потребитель, организация, заключившая с организацией теплоснабжения договор о поставке тепла.	Как только с потребителем юридическим лицом заключается договор о поставке тепла, формируется новый экземпляр данного множества сущностей.
E5	Начисление	Сумма, которая должна быть выплачена потребителем компании теплоснабжения.	Начисление вычисляется на основе количественного показателя площади, которой владеет потребитель, а также на основе тарифа, который был определён для данного начисления
E6	Тариф	Установленный размер стоимости тепла за квадратный метр помещения.	Стоимость тепла устанавливает организация теплоснабжения. Как только принимается решение о изменении стоимости электроэнергии, формируется новый экземпляр данного множества сущностей.
E7	Оплата	Документально подтвержденная уплата средств за потреблённое тепло.	Как только потребитель совершает оплату за потреблённое тепло, формируется новый экземпляр данного множества сущностей.

## 2.2. Концептуальная модель связей сущности

Для определенных выше сущностей предметной области определены связи между сущностями в виде матрицы (таблица 2.2).

Таблица 2.2 – Матрица связей сущностей

	E1	E2	E3	E4	E5	E6	E7
E1		X					
E2	X		X	X	X		
E3		X					
E4		X					
E5		X				X	X
E6					X		
E7					X		

Все определенные выше связи между сущностями предметной области были описаны в виде таблицы (таблица 2.3).

Таблица 2.3 – Табличная модель связей сущностей

Номер связи	Номер 1-ой сущности	Номер 2-ой сущности	Название связи	Тип связи	Описание связи
R1	E1	E2	Владеет	M:M	Каждый потребитель может владеть одной или больше квартирами, или ни одной. Каждая квартира может принадлежать одному потребителю или нескольким.

R2	E2	E3	Является	1:1	Каждый потребитель может являться только одним физическим лицом, или не являться им вовсе. Каждое физическое лицо может быть только одним потребителем.
R3	E2	E4	Является	1:1	Каждый потребитель может являться только одним юридическим лицом, или не являться им вовсе. Каждое юридическое лицо может быть только одним потребителем.
R4	E2	E5	Получает	1:M	Каждый потребитель может получить неограниченное количество начислений, каждое начисление может иметь только одного потребителя
R5	E5	E6	Устанавливается	M:1	Каждое начисление может иметь только один тариф, каждый тариф может использоваться для неограниченного количества начислений
R6	E5	E7	Производится	1:1	Каждое начисление может быть оплачено только одним счётом, один счёт может оплачивать только одно начисление

На основе описанных сущностей и связей предметной области была построена концептуальная схема уровня сущностей. Полученная схема представлена на рисунке А.1 приложение А.

### 2.3. Концептуальная модель уровня свойств сущностей

Для связей и сущностей рассматриваемой предметной области были определены следующие свойства (таблица 2.4 – 2.10).

Таблица 2.4 – Свойства сущности «Квартира»:

Наименование свойства	Тип свойства
номер	ключевое
адрес	составное (район, улица, дом, квартира) необязательное
план	необязательное

Таблица 2.5 – Свойства сущности «Потребитель»:

Наименование свойства	Тип свойства
номер	ключевое
лицевой счёт	обязательное
наименование	необязательное
телефон	необязательное
электронная почта	необязательное
тип	необязательное
площадь	обязательное

Таблица 2.6 – Свойства сущности «Физическое лицо»:

Наименование свойства	Тип свойства
номер потребителя	ключевое
паспортные данные	составное (серия, номер, кем выдан и когда выдан)

СНИЛС	уникальное
ФИО	составное (фамилия, имя, отчество)
дата рождения	обязательное

Таблица 2.7 – Свойства сущности «Юридическое лицо»:

Наименование свойства	Тип свойства
номер потребителя	ключевое
наименование	необязательное
ОГРН	уникальное

Таблица 2.8 – Свойства сущности «Тариф»:

Наименование свойства	Тип свойства
номер	ключевое
наименование	необязательное
дата вступления в силу	необязательное
дата окончания	необязательное
стоимость 1 кв.м. тепла	обязательное

Таблица 2.9 – Свойства сущности «Оплата»:

Наименование свойства	Тип свойства
номер	ключевое
номер начисления	обязательное
дата оплаты	обязательное
сумма	обязательное

Таблица 2.10 – Свойства сущности «Начисление»:

Наименование свойства	Тип свойства
номер	ключевое
номер тарифа	обязательное
номер потребителя	обязательное
дата с	необязательное
дата по	необязательное



сумма	обязательное
-------	--------------

В концептуальной модели, помимо сущностей, свойства присущи и некоторым связям между сущностями (таблица 2.11).

Таблица 2.11 – Свойства связи между сущностями «Потребитель» и «Квартира»:

Наименование свойства	Тип свойства
номер документа	ключевое
тип документа	обязательное
дата вступления в собственность	обязательное
дата окончания права собственности	необязательное

На основе описанных свойств сущностей и связей предметной области была построена концептуальная схема уровня свойств. Полученная схема представлена на рисунке А.1 приложение А.

#### **2.4. Определение классов принадлежности сущностей предметной области**

Для всех сущностей предметной области, участвующих во всех связях, были описаны классы принадлежности на основе диаграмм ER-экземпляров и типов:

1. Связь «Потребитель – Квартира» М:М (рисунок 2.2).

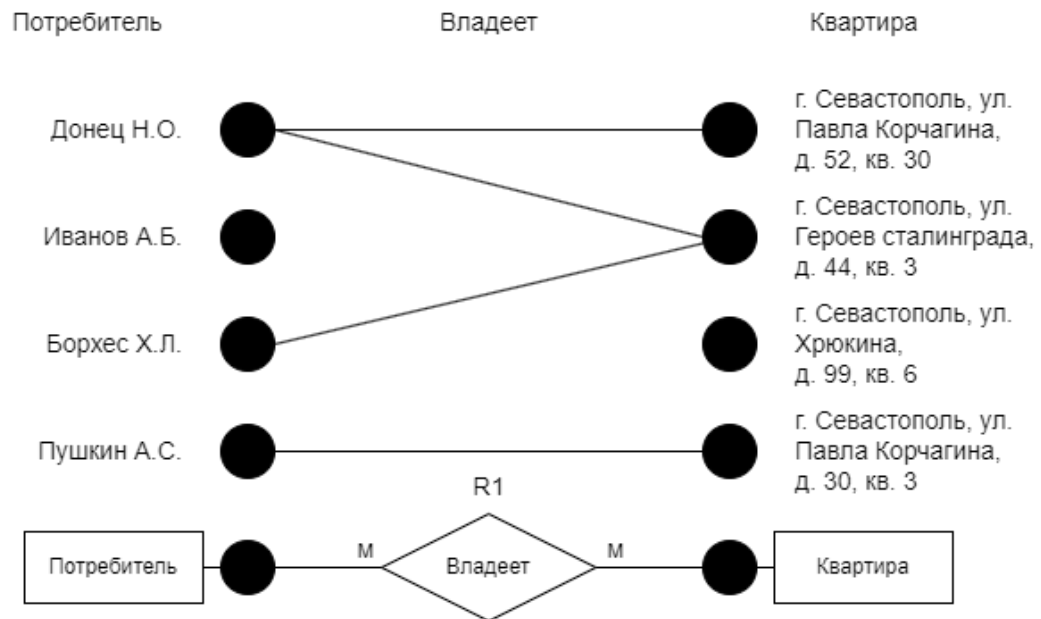


Рисунок 2.2 – Потребитель - Квартира

Сущность Потребитель в данной связи имеет необязательный класс принадлежности. Сущность Квартира в данной связи также имеет необязательный класс принадлежности.

## 2. Связь «Тариф – Начисление» 1:M (рисунок 2.3).

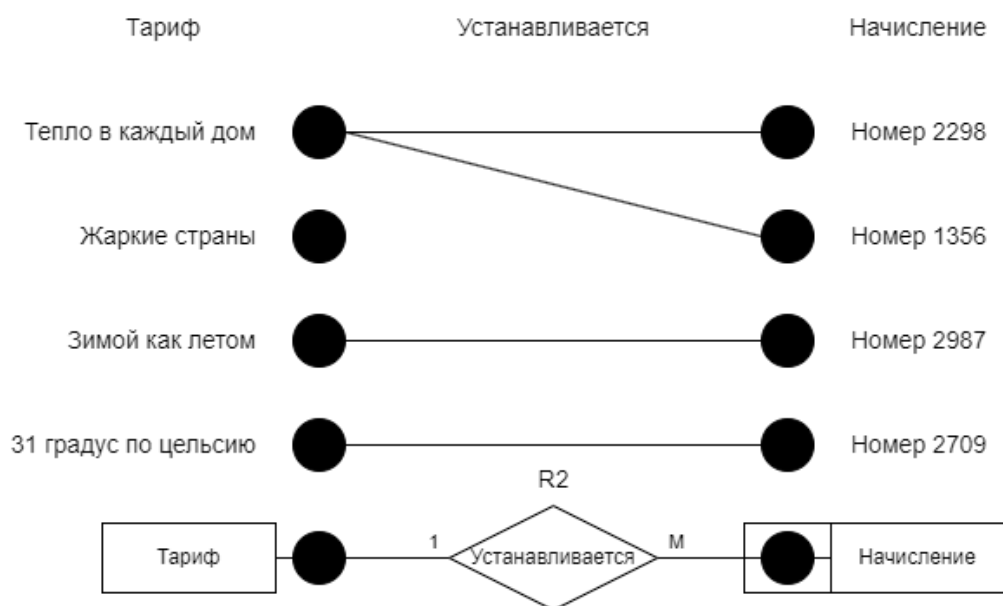


Рисунок 2.3 – Тариф - Начисление

Сущность Тариф в данной связи имеет необязательный класс принадлежности. Сущность Начисление в данной связи имеет обязательный класс принадлежности.

### 3. Связь «Потребитель – Начисление» 1:M (рисунок 2.4).

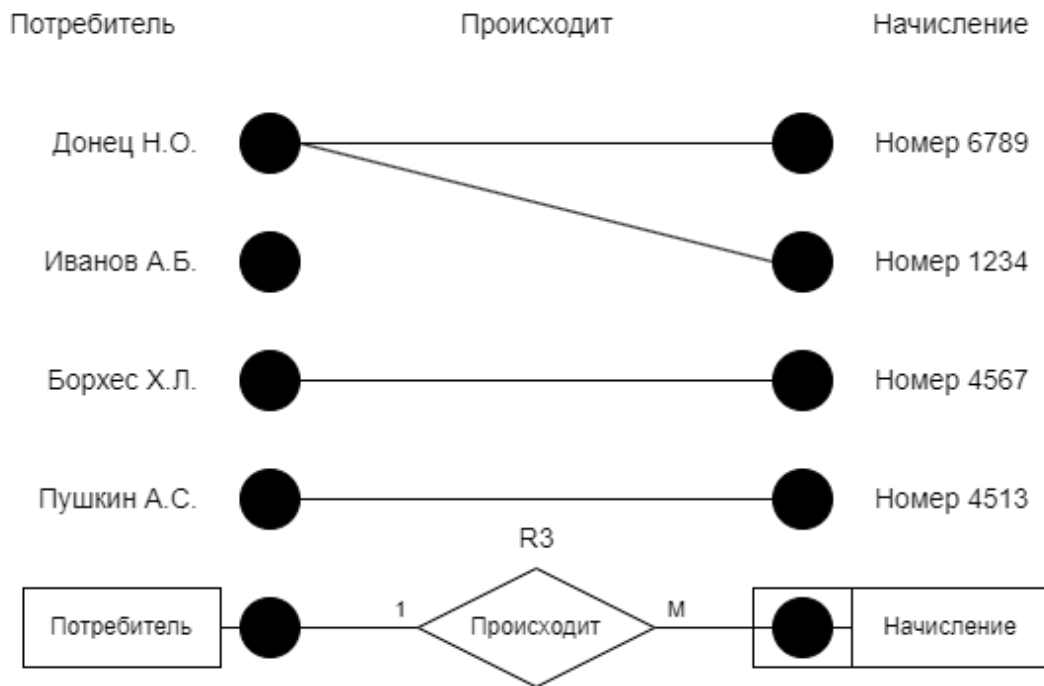


Рисунок 2.4 – Потребитель - Начисление

Сущность Потребитель в данной связи имеет необязательный класс принадлежности. Сущность Начисление в данной связи имеет обязательный класс принадлежности.

### 4. Связь «Потребитель – Физ. лицо» 1:1 (рисунок 2.5).

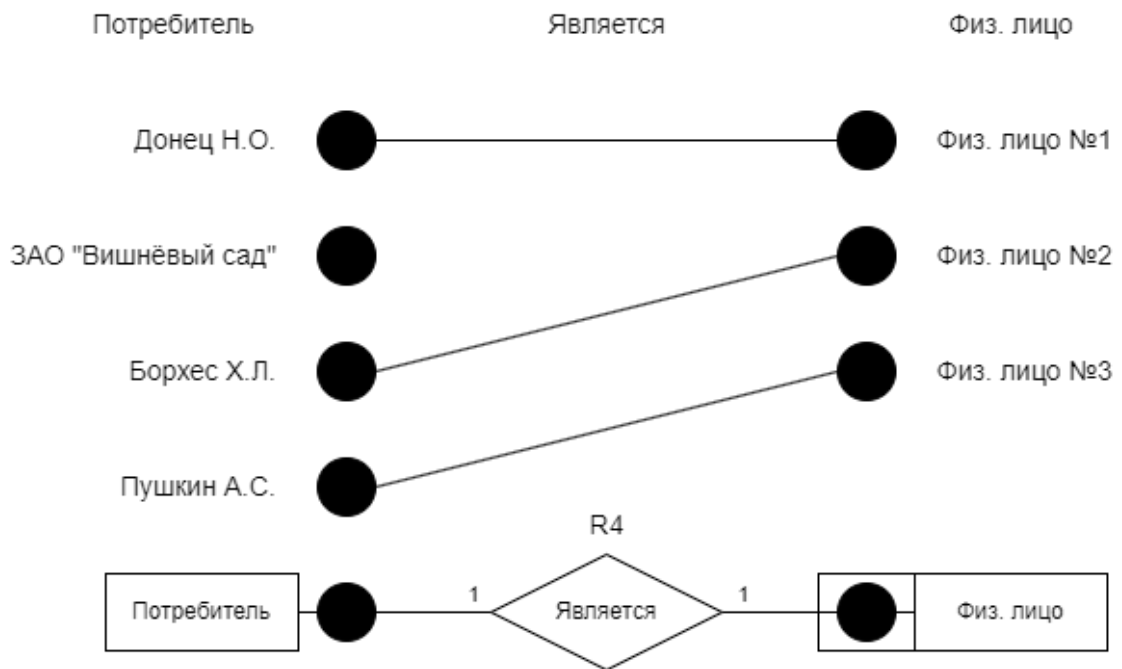


Рисунок 2.5 – Потребитель – Физ. лицо

Сущность Потребитель в данной связи имеет необязательный класс принадлежности. Сущность Физ. лицо в данной связи имеет обязательный класс принадлежности.

#### 5. Связь «Потребитель – Юр. лицо» 1:1 (рисунок 2.6).

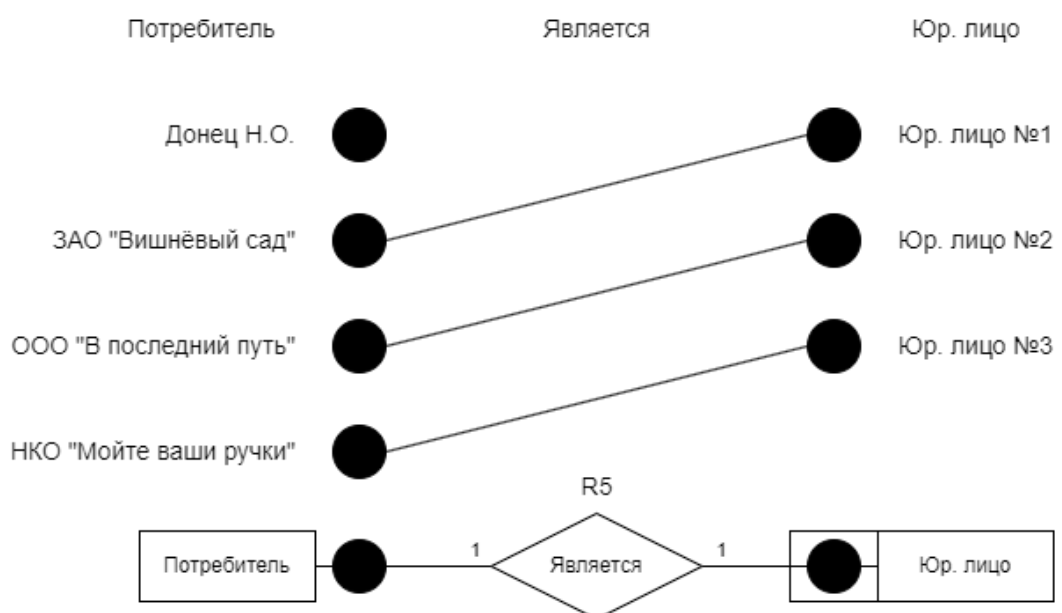


Рисунок 2.6 – Потребитель – Юр. лицо

Сущность Потребитель в данной связи имеет необязательный класс принадлежности. Сущность Юр. лицо в данной связи имеет обязательный класс принадлежности.

6. Связь «Начисление – Оплата» 1:1 (рисунок 2.7).

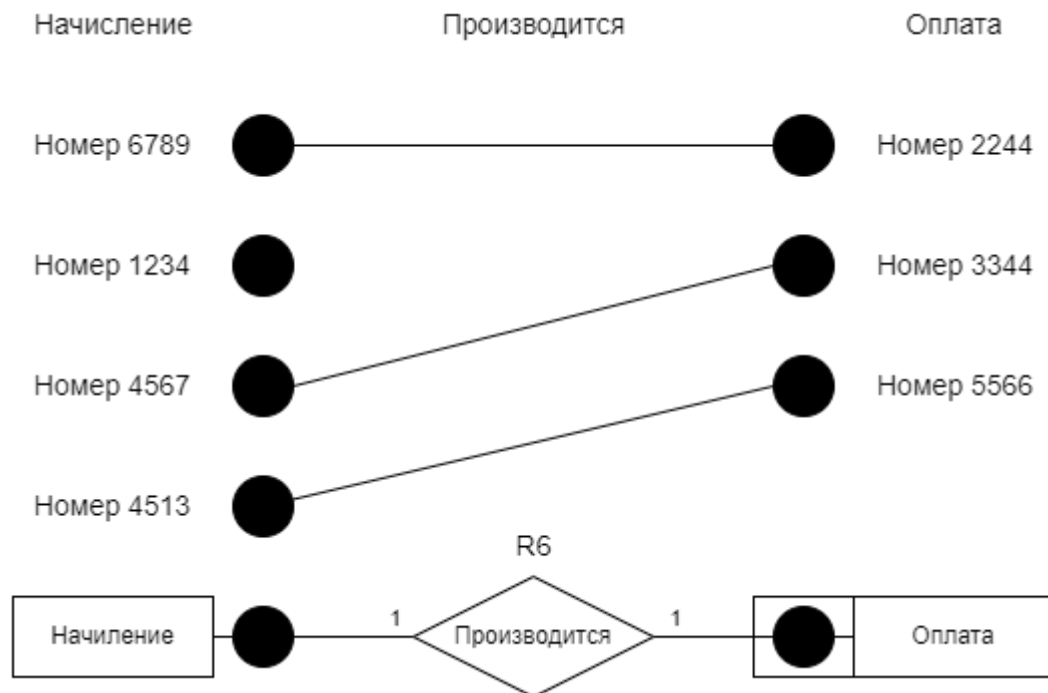


Рисунок 2.7 – Начисление - Оплата

Сущность Начисление в данной связи имеет необязательный класс принадлежности. Сущность Оплата в данной связи имеет обязательный класс принадлежности.

## 2.5. Переход к логической модели базы данных

Для рассмотренных связей обоснован переход от концептуальной к логической модели базы данных – реляционной модели.

### 1. Связь «Потребитель – Квартира» (рисунок 2.8).

Так как тип связи «Потребитель – Квартира» М:М и обе сущности имеют необязательный класс принадлежности, то логическая модель данной связи представляет собой 3 отношения, и одно из них связывает два других с помощью внешних ключей.

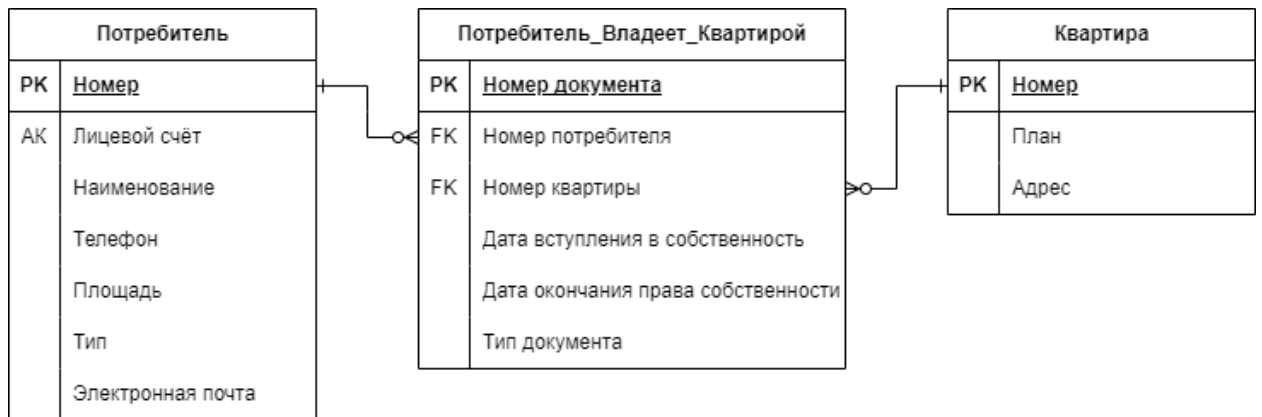


Рисунок 2.8 – Логическая структура для связи «Потребитель – Квартира»

### 2. Связь «Тариф – Начисление» (рисунок 2.9).

Так как тип связи «Тариф – Начисление» 1:М и М-связная сущность имеет обязательный класс принадлежности, то логическая модель данной связи представляет собой 2 отношения, и к отношению М-связной сущности добавляется в качестве атрибута ключ 1-связной сущности.

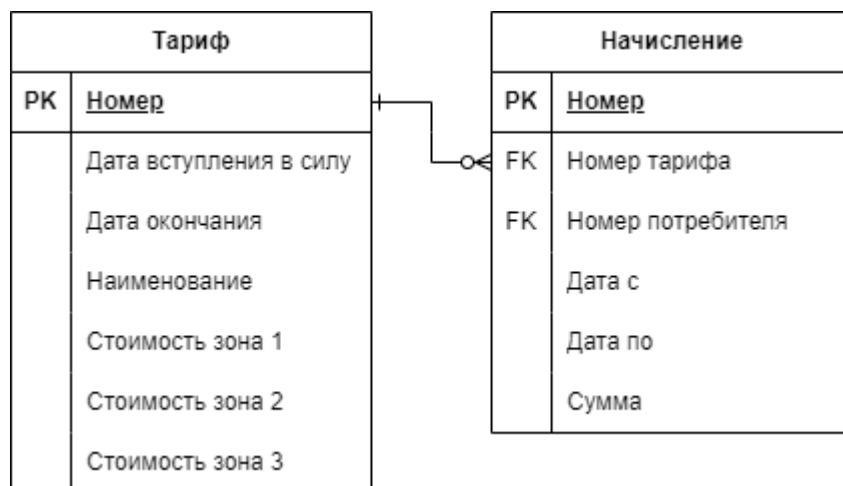


Рисунок 2.9 – Логическая структура для связи «Тариф – Начисление»

### 3. Связь «Потребитель – Начисление» (рисунок 2.10).

Так как тип связи «Потребитель – Начисление» 1:М и М-связная сущность имеет обязательный класс принадлежности, то логическая модель данной связи представляет собой 2 отношения, и к отношению М-связной сущности добавляется в качестве атрибута ключ 1-связной сущности.

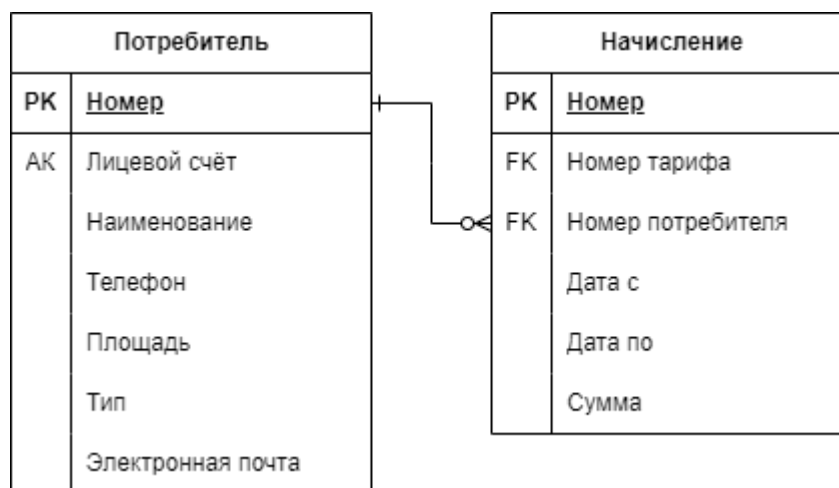


Рисунок 2.10 – Логическая структура для связи «Потребитель – Начисление»

### 4. Связи «Потребитель – Физ. лицо» и «Потребитель – Юр. лицо» (рисунок 2.11).

Так как тип связей «Потребитель – Физ. лицо» и «Потребитель – Юр. лицо» 1:1, и класс принадлежности сущности Потребитель необязательный, а Физ. лицо и Юр. лицо – обязательный, то логическая модель данных связей представляет собой 3 отношения, среди которых Физ. лицо и Юр. Лицо являются подтипами для отношения Потребитель.

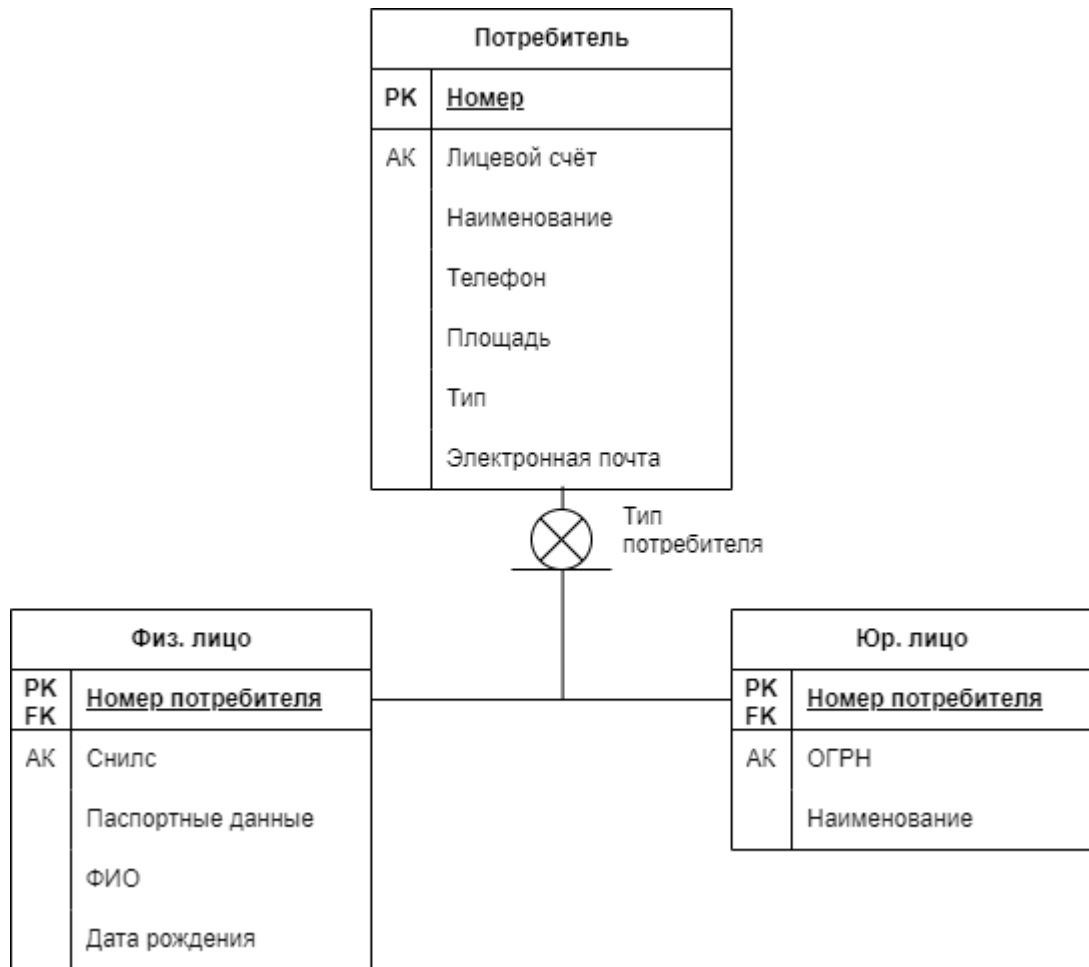


Рисунок 2.11 – Логическая структура для связей «Потребитель – Физ. лицо» и «Потребитель – Юр. лицо»

##### 5. Связь «Начисление – Оплата» (рисунок 2.12).

Так как тип связей «Начисление – Оплата» 1:1, и класс принадлежности сущности Начисление необязательный, а Оплата – обязательный, то логическая модель данных связей представляет собой 2 отношения, где к отношению, сущность которого имеет обязательный класс принадлежности, добавляется в качестве атрибута ключ сущности с необязательным классом принадлежности.



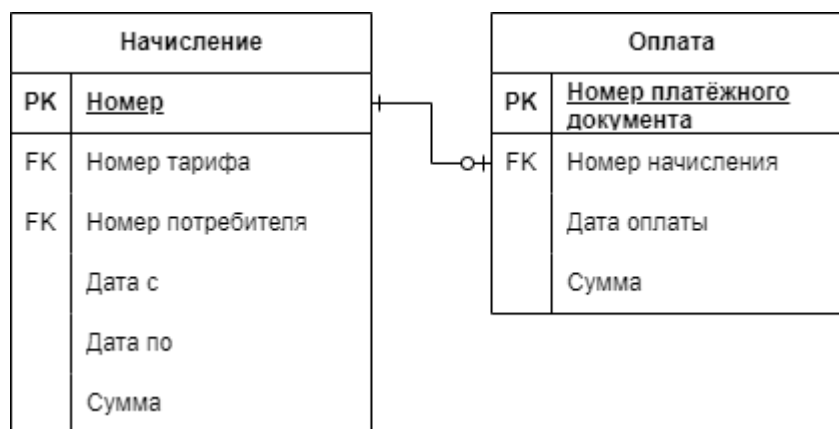


Рисунок 2.12 – Логическая структура для связи «Начисление – Оплата»

### 3. ДАТАЛОГИЧЕСКОЕ ПРОЕТИРОВАНИЕ БАЗЫ ДАННЫХ

#### 3.1. Описание логической модели данных

На основе определённых ранее переходов к логической модели для каждой сущности предметной области, была построена логическая модель базы данных (рисунок 3.1).

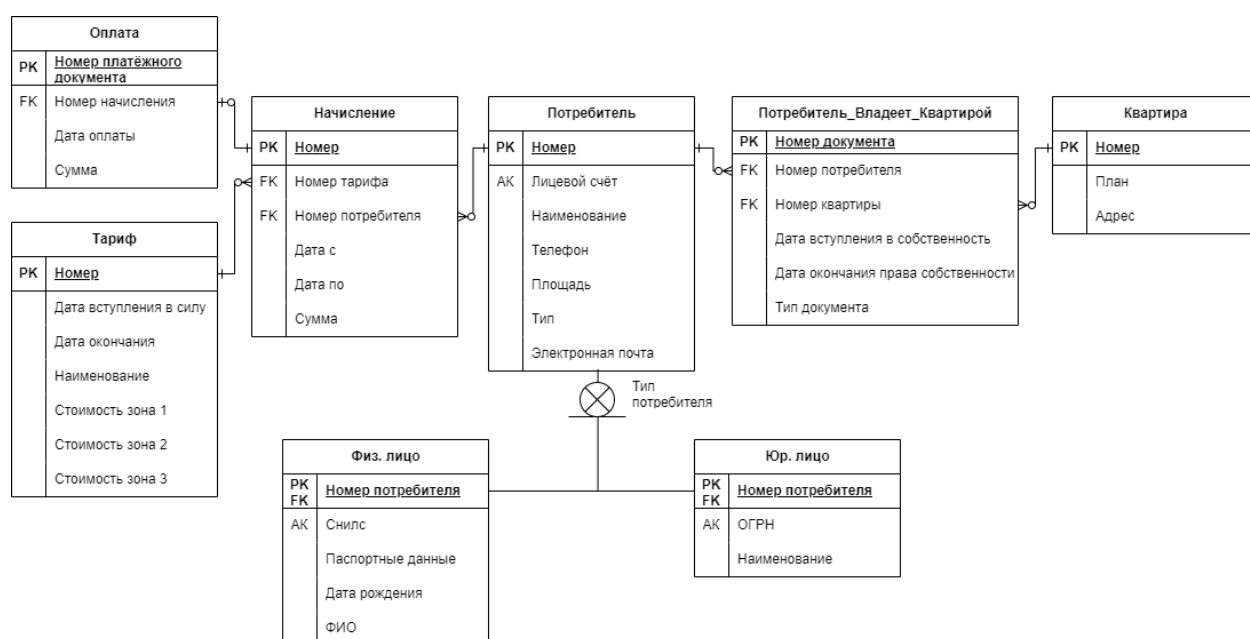


Рисунок 3.1 – Логическая модель базы данных

Для всех отношений описана логическая структура.

Отношение «Потребитель» содержит следующие ключевые атрибуты:

- 1) «Номер» — первичный ключ, однозначно идентифицирующий потребителя;
- 2) «Лицевой счёт» — альтернативный ключ (уникальное значение в пределах отношения).

Потребитель может быть как физическим лицом, так и юридическим. Тип потребителя определяют отношения «Физ. лицо» и «Юр. лицо».

Отношение «Физ. лицо» содержит следующие ключевые атрибуты:

- 1) «Номер потребителя» – внешний ключ к атрибуту «Номер» отношения «Потребитель» и одновременно первичный ключ, однозначно идентифицирующий физическое лицо;
- 2) «СНИЛС» – альтернативный ключ (уникальное значение в пределах отношения);
- 3) «Паспортные данные»: составной атрибут, который рационально заменить несколькими: «Серия и номер» – альтернативный ключ (уникальное значение в пределах отношения), «Кем выдан», «Когда выдан».

Также, для удобства, рационально атрибут «ФИО» заменить несколькими атрибутами: «Фамилия», «Имя», «Отчество».

Отношение «Юр. лицо» содержит следующие ключевые атрибуты:

- 1) «Номер потребителя» – внешний ключ к атрибуту «Номер» отношения «Потребитель» и одновременно первичный ключ, однозначно идентифицирующий юридическое лицо;
- 2) «ОГРН» – альтернативный ключ (уникальное значение в пределах отношения).

Отношение «Потребитель\_Владеет\_Квартирой» содержит следующие ключевые атрибуты:

- 1) «Номер документа» – первичный ключ, однозначно идентифицирующий каждый кортеж в отношении;
- 2) «Номер квартиры» – внешний ключ к атрибуту «Лицевой счет» отношения «Квартира»;
- 3) «Номер потребителя» – внешний ключ к атрибуту «Номер» отношения «Потребитель».

Отношение «Квартира» содержит ключевой атрибут «Номер» – первичный ключ, однозначно идентифицирующий каждую квартиру. Также, для удобства, рационально атрибут «Адрес» заменить несколькими атрибутами: «Район», «Улица», «Номер дома», «Квартира».

Отношение «Начисление» содержит следующие ключевые атрибуты:

- 1) «Номер» – первичный ключ, однозначно идентифицирующий каждый кортеж в отношении;
- 2) «Номер тарифа» – внешний ключ к атрибуту «Номер» отношения «Тариф»;
- 3) «Лицевой счёт потребителя» – внешний ключ к атрибуту «Лицевой счёт» отношения «Потребитель».

Отношение «Тариф» содержит ключевой атрибут «Номер» – первичный ключ, однозначно идентифицирующий каждый тариф.

Отношение «Оплата» содержит следующие ключевые атрибуты:

- 1) «Номер платёжного документа» – первичный ключ, однозначно идентифицирующий каждый кортеж в отношении;
- 2) «Номер начисления» – внешний ключ к атрибуту «Номер» отношения «Начисление»;

### 3.2. Уточнение логической модели базы данных

Расширив некоторые атрибуты отношений («Адрес», «ФИО», «Паспортные данные»), возможно несколько уточнить логическую модель базы данных предметной области (рисунок 3.2).

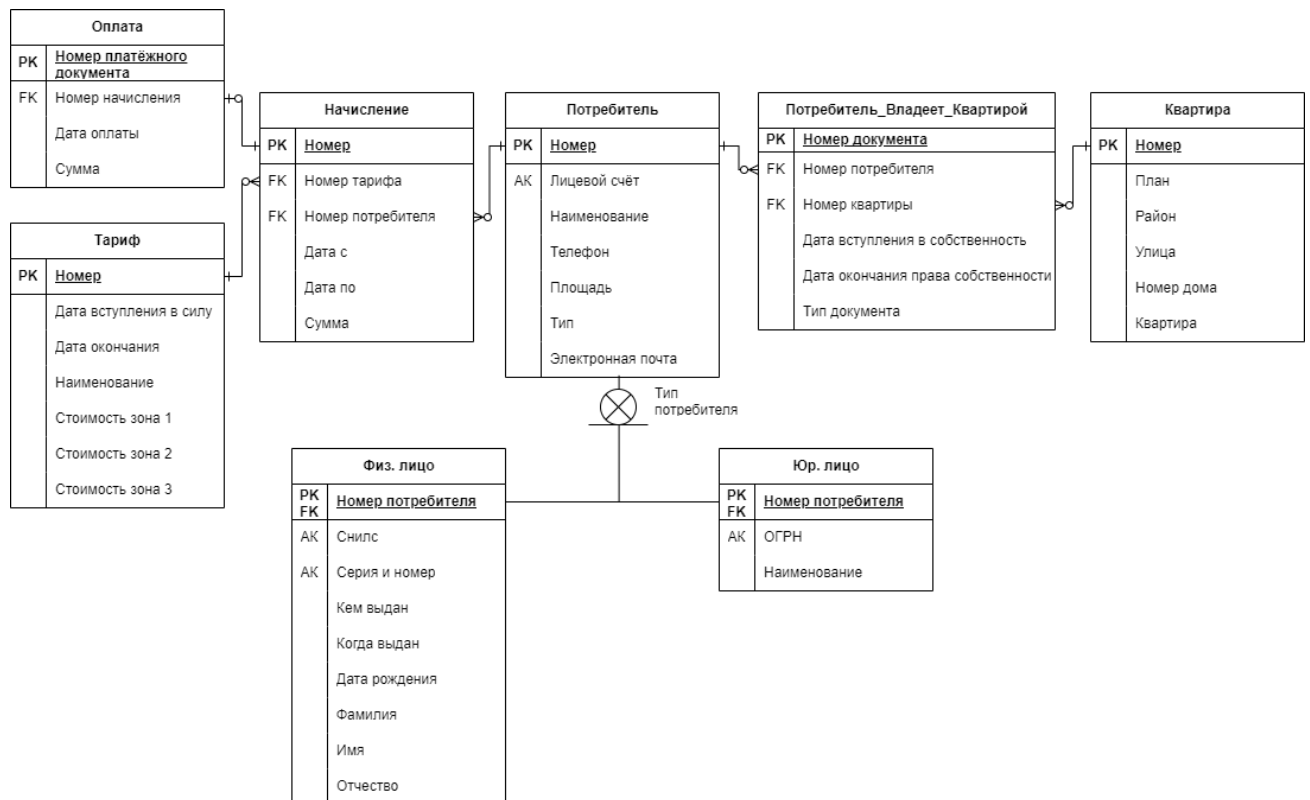


Рисунок 3.2 – Уточненная логическая модель базы данных

### 3.3. Описание функциональных зависимостей

На основе определённых ранее отношений базы данных, были определены их функциональные зависимости:

1. Отношение «Потребитель» (рисунок 3.3) имеет следующие функциональные зависимости (ФЗ):

Таблица 3.1 – ФЗ отношения «Потребитель»

ФЗ	Описание ФЗ
Номер → Лицевой счёт	Неприводимая ФЗ
Номер → Наименование	Неприводимая ФЗ
Номер → Телефон	Неприводимая ФЗ
Номер → Площадь	Неприводимая ФЗ
Номер → Тип	Неприводимая ФЗ
Номер → Электронная почта	Неприводимая ФЗ
Лицевой счёт → Номер	Неприводимая ФЗ
Лицевой счёт → Наименование	Неприводимая ФЗ
Лицевой счёт → Телефон	Неприводимая ФЗ
Лицевой счёт → Площадь	Неприводимая ФЗ
Лицевой счёт → Тип	Неприводимая ФЗ
Лицевой счёт → Электронная почта	Неприводимая ФЗ

Диаграмма функциональных зависимостей отношения «Потребитель» представлена на рисунке 3.3.

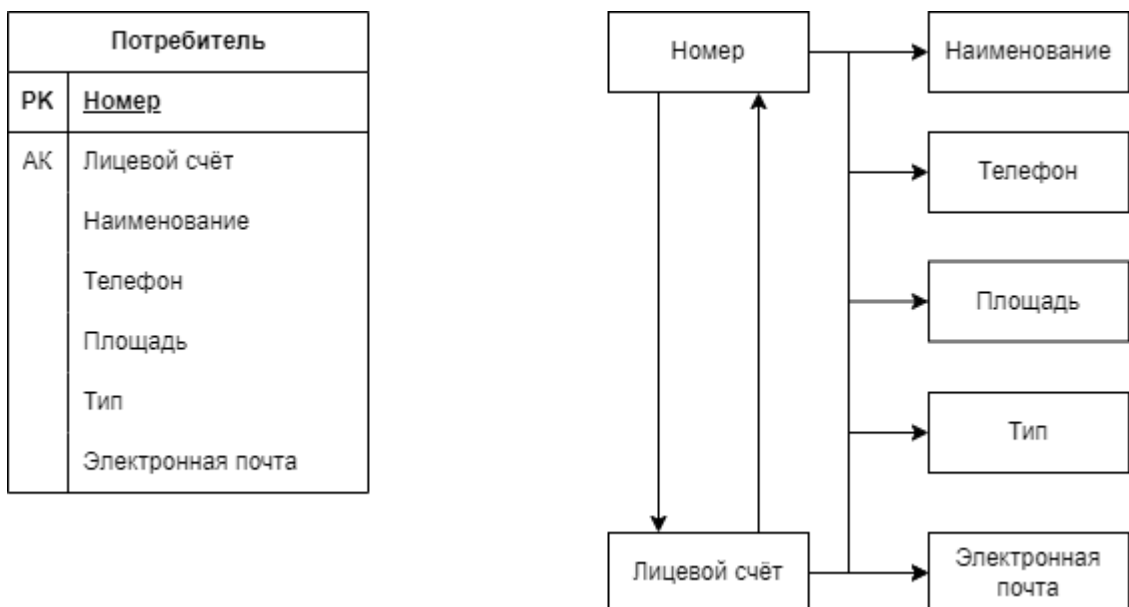


Рисунок 3.3 – ФЗ отношения «Потребитель»

2. Отношение «Физ. лицо» (рисунок 3.4) имеет следующие функциональные зависимости:

Таблица 3.2 – ФЗ отношения «Физ. лицо»

ФЗ	Описание ФЗ
Номер потребителя → Серия и номер	Неприводимая ФЗ
Номер потребителя → СНИЛС	Неприводимая ФЗ
Номер потребителя → Кем выдан	Неприводимая ФЗ
Номер потребителя → Когда выдан	Неприводимая ФЗ
Номер потребителя → Дата рождения	Неприводимая ФЗ
Номер потребителя → Фамилия	Неприводимая ФЗ
Номер потребителя → Имя	Неприводимая ФЗ
Номер потребителя → Отчество	Неприводимая ФЗ
Серия и номер → Номер потребителя	Неприводимая ФЗ
Серия и номер → СНИЛС	Неприводимая ФЗ
Серия и номер → Кем выдан	Неприводимая ФЗ
Серия и номер → Когда выдан	Неприводимая ФЗ
Серия и номер → Дата рождения	Неприводимая ФЗ
Серия и номер → Фамилия	Неприводимая ФЗ
Серия и номер → Имя	Неприводимая ФЗ
Серия и номер → Отчество	Неприводимая ФЗ
СНИЛС → Номер потребителя	Неприводимая ФЗ
СНИЛС → Серия и номер	Неприводимая ФЗ
СНИЛС → Кем выдан	Неприводимая ФЗ
СНИЛС → Когда выдан	Неприводимая ФЗ
СНИЛС → Дата рождения	Неприводимая ФЗ
СНИЛС → Фамилия	Неприводимая ФЗ
СНИЛС → Имя	Неприводимая ФЗ

СНИЛС → Отчество	Неприводимая ФЗ
------------------	-----------------

Диаграмма функциональных зависимостей отношения «Физ. лицо» представлена на рисунке 3.4.

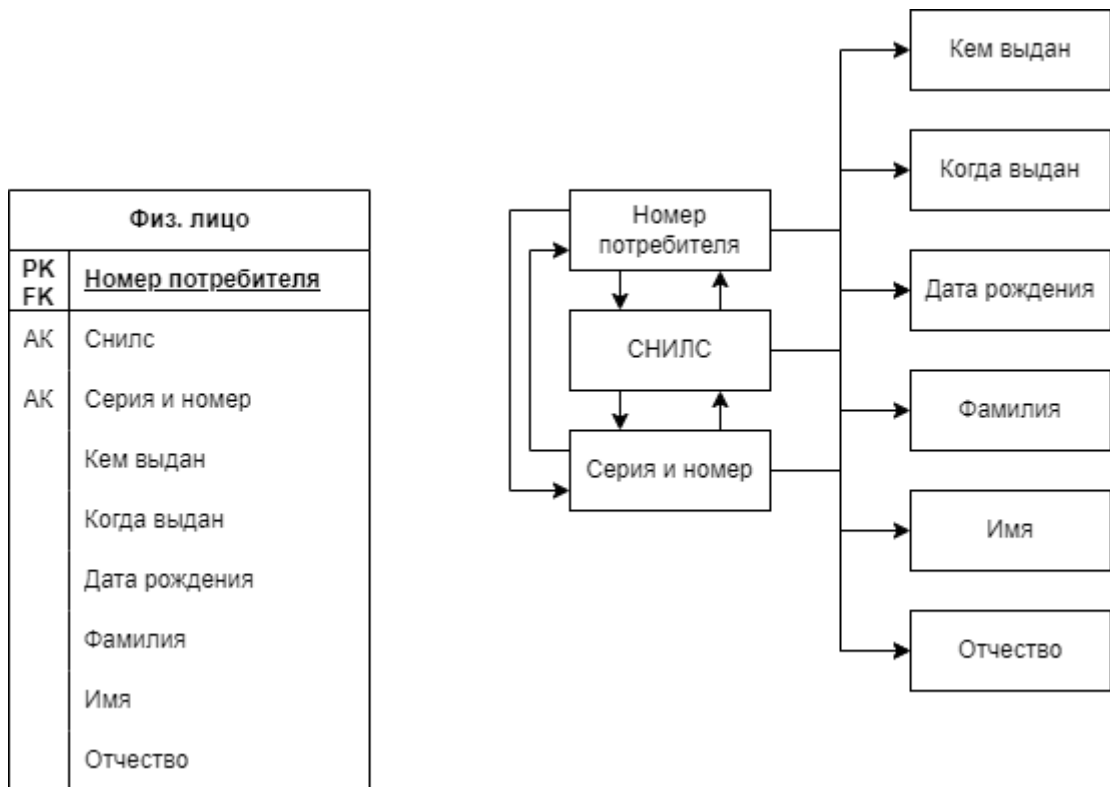


Рисунок 3.4 – ФЗ отношения «Физ. лицо»

3. Отношение «Юр. лицо» (рисунок 3.5) имеет следующие функциональные зависимости:

Таблица 3.3 – ФЗ отношения «Юр. лицо»

ФЗ	Описание ФЗ
Номер потребителя → ОГРН	Неприводимая ФЗ
Номер потребителя → Наименование	Неприводимая ФЗ
ОГРН → Номер потребителя	Неприводимая ФЗ
ОГРН → Наименование	Неприводимая ФЗ



Диаграмма функциональных зависимостей отношения «Юр. лицо» представлена на рисунке 3.5.

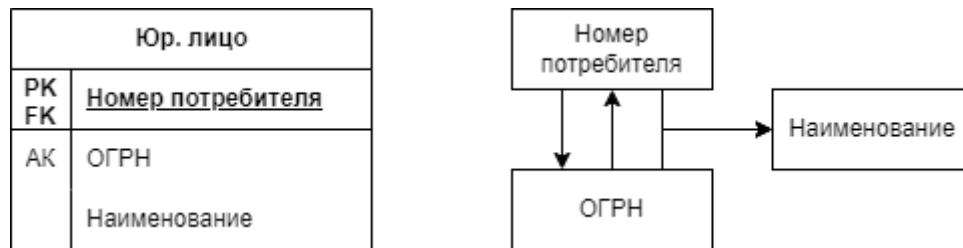


Рисунок 3.5 – ФЗ отношения «Юр. лицо»

4. Отношение «Начисление» (рисунок 3.6) имеет следующие функциональные зависимости:

Таблица 3.4 – ФЗ отношения «Начисление»

ФЗ	Описание ФЗ
Номер → Номер тарифа	Неприводимая ФЗ
Номер → Лицевой счёт потребителя	Неприводимая ФЗ
Номер → Дата с	Неприводимая ФЗ
Номер → Дата по	Неприводимая ФЗ
Номер → Сумма	Неприводимая ФЗ

Диаграмма функциональных зависимостей отношения «Начисление» представлена на рисунке 3.6.

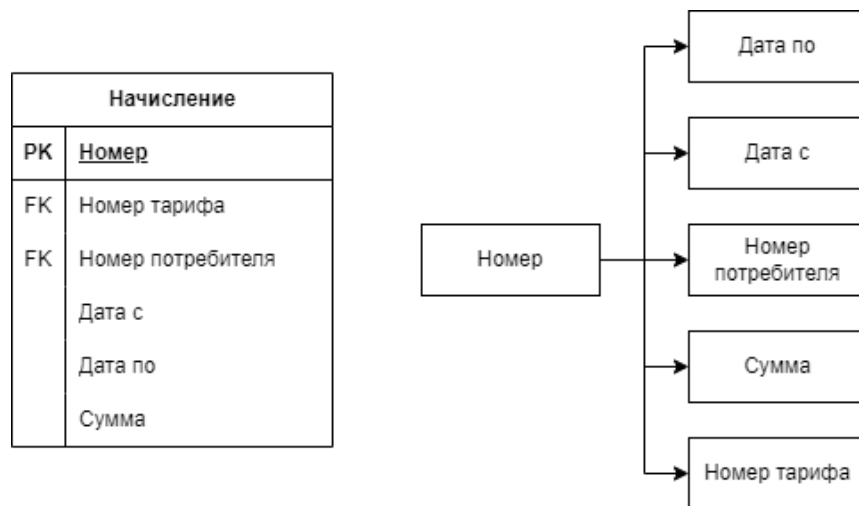


Рисунок 3.6 – ФЗ отношения «Начисление»

5. Отношение «Тариф» (рисунок 3.7) имеет следующие функциональные зависимости:

Таблица 3.5 – ФЗ отношения «Тариф»

ФЗ	Описание ФЗ
Номер → Дата вступления в силу	Неприводимая ФЗ
Номер → Дата окончания	Неприводимая ФЗ
Номер → Наименование	Неприводимая ФЗ
Номер → Стоимость	Неприводимая ФЗ

Диаграмма функциональных зависимостей отношения «Тариф» представлена на рисунке 3.7.

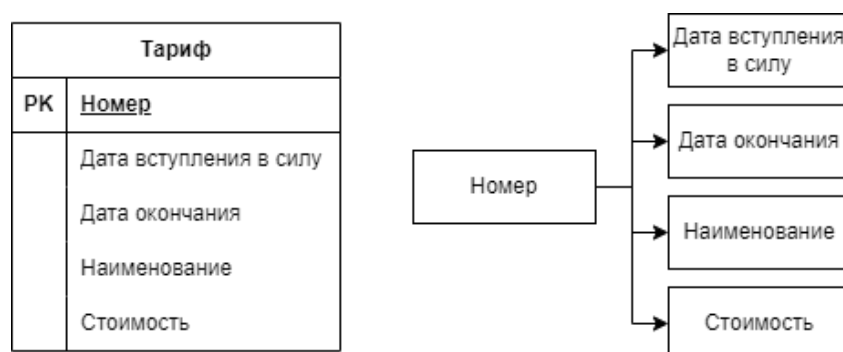


Рисунок 3.7 – ФЗ отношения «Тариф»

6. Отношение «Оплата» (рисунок 3.8) имеет следующие функциональные зависимости:

Таблица 3.6 – ФЗ отношения «Оплата»

ФЗ	Описание ФЗ
Номер платёжного документа → Номер начисления	Неприводимая ФЗ
Номер платёжного документа → Дата оплаты	Неприводимая ФЗ
Номер платёжного документа → Сумма	Неприводимая ФЗ

Диаграмма функциональных зависимостей отношения «Оплата» представлена на рисунке 3.8.



Рисунок 3.8 – ФЗ отношения «Оплата»

7. Отношение «Потребитель\_Владеет\_Квартирой» (рисунок 3.9) имеет следующие функциональные зависимости:

Таблица 3.7 – ФЗ отношения «Потребитель\_Владеет\_Квартирой»

ФЗ	Описание ФЗ
Номер документа → Номер потребителя	Неприводимая ФЗ

Номер документа → Дата вступления в собственность	Неприводимая ФЗ
Номер документа → Дата окончания собственности	Неприводимая ФЗ
Номер документа → Номер квартиры	Неприводимая ФЗ
Номер документа → Тип документа	Неприводимая ФЗ

Диаграмма функциональных зависимостей отношения «Потребитель\_Владеет\_Квартирой» представлена на рисунке 3.9.

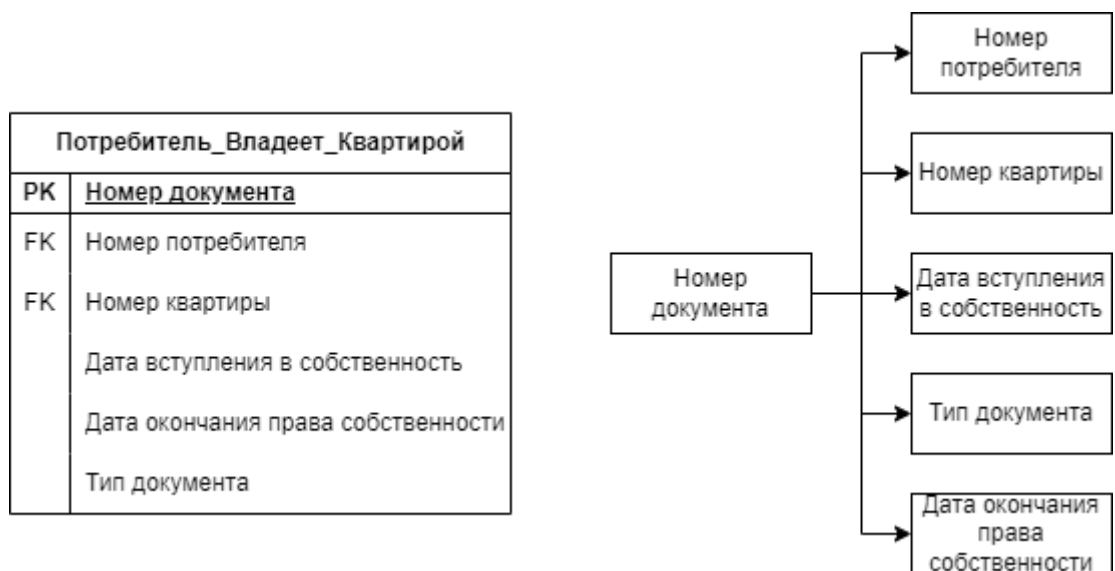


Рисунок 3.9 – ФЗ отношения «Потребитель\_Владеет\_Квартирой»

8. Отношение «Квартира» (рисунок 3.10) имеет следующие функциональные зависимости:

Таблица 3.8 – ФЗ отношения «Квартира»

ФЗ	Описание ФЗ
Номер → План	Неприводимая ФЗ
Номер → Район	Неприводимая ФЗ
Номер → Улица	Неприводимая ФЗ

Номер → Номер дома	Неприводимая ФЗ
Номер → Квартира	Неприводимая ФЗ

Диаграмма функциональных зависимостей отношения «Квартира» представлена на рисунке 3.10.

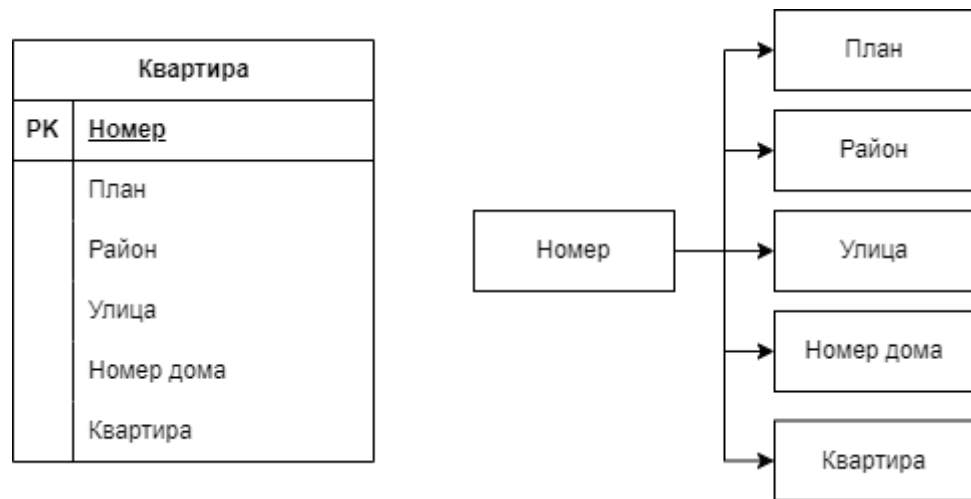


Рисунок 3.10 – ФЗ отношения «Квартира»

### 3.4. Нормализация отношений

1. Нормализация отношения «Потребитель», согласно построенной диаграммы ФЗ.

Данное отношение находится в первой нормальной форме (1НФ), так как каждая запись может содержать только одно значение для каждого из атрибутов.

Данное отношение также находится во 2НФ, так как оно находится в 1НФ, и все не ключевые атрибуты зависят от потенциальных ключей «Номер» и «Лицевой счёт».

Данное отношение также находится в НФБК, так как оно находится во 2НФ, и отсутствуют транзитивные зависимости неключевых атрибутов от потенциальных ключей «Номер» и «Лицевой счёт».

Отношение не требует нормализации, так как оно уже находится в НФБК.

2. Нормализация отношения «Физ. лицо», согласно построенной диаграммы ФЗ.

Данное отношение находится в 1НФ, так как каждая запись может содержать только одно значение для каждого из атрибутов.

Данное отношение также находится во 2НФ, так как оно находится в 1НФ, и все не ключевые атрибуты зависят от потенциальных ключей «Номер потребителя», «Серия и номер паспорта» и «СНИЛС».

Данное отношение также находится в НФБК, так как оно находится во 2НФ, и отсутствуют транзитивные зависимости неключевых атрибутов от потенциальных ключей «Номер потребителя», «Серия и номер» и «СНИЛС».

Отношение не требует нормализации, так как оно уже находится в НФБК.

3. Нормализация отношения «Юр. лицо», согласно построенной диаграммы ФЗ.

Данное отношение находится в 1НФ, так как каждая запись может содержать только одно значение для каждого из атрибутов.

Данное отношение также находится во 2НФ, так как оно находится в 1НФ, и все не ключевые атрибуты зависят от потенциальных ключей «Номер потребителя» и «ОГРН».

Данное отношение также находится в НФБК, так как оно находится во 2НФ, и отсутствуют транзитивные зависимости неключевых атрибутов от потенциальных ключей «Номер потребителя» и «ОГРН».

Отношение не требует нормализации, так как оно уже находится в НФБК.

4. Нормализация отношения «Начисление», согласно построенной диаграммы ФЗ.

Данное отношение находится в 1НФ, так как каждая запись может содержать только одно значение для каждого из атрибутов.

Данное отношение также находится во 2НФ, так как оно находится в 1НФ, и все не ключевые атрибуты зависят от первичного ключа «Номер».

Данное отношение также находится в 3НФ, так как оно находится во 2НФ, и отсутствуют транзитивные зависимости неключевых атрибутов от первичного ключа «Номер».

Отношение не требует нормализации, так как оно уже находится в 3НФ.

5. Нормализация отношения «Оплата», согласно построенной диаграммы ФЗ.

Данное отношение находится в 1НФ, так как каждая запись может содержать только одно значение для каждого из атрибутов.

Данное отношение также находится во 2НФ, так как оно находится в 1НФ, и все не ключевые атрибуты зависят от первичного ключа «Номер платёжного документа».

Данное отношение также находится в 3НФ, так как оно находится во 2НФ, и отсутствуют транзитивные зависимости неключевых атрибутов от первичного ключа «Номер платёжного документа».

Отношение не требует нормализации, так как оно уже находится в 3НФ.

6. Нормализация отношения «Тариф», согласно построенной диаграммы ФЗ.

Данное отношение находится в 1НФ, так как каждая запись может содержать только одно значение для каждого из атрибутов.

Данное отношение также находится во 2НФ, так как оно находится в 1НФ, и все не ключевые атрибуты зависят от первичного ключа «Номер».

Данное отношение также находится в 3НФ, так как оно находится во 2НФ, и отсутствуют транзитивные зависимости неключевых атрибутов от первичного ключа «Номер».

Отношение не требует нормализации, так как оно уже находится в 3НФ.

7. Нормализация отношения «Потребитель владеет квартирой», согласно построенной диаграммы ФЗ.

Данное отношение находится в 1НФ, так как каждая запись может содержать только одно значение для каждого из атрибутов.

Данное отношение также находится во 2НФ, так как оно находится в 1НФ, и все не ключевые атрибуты зависят от первичного ключа «Номер».

Данное отношение также находится в 3НФ, так как оно находится во 2НФ, и отсутствуют транзитивные зависимости неключевых атрибутов от первичного ключа «Номер».

Отношение не требует нормализации, так как оно уже находится в 3НФ.

8. Нормализация отношения «Квартира», согласно построенной диаграммы ФЗ.

Данное отношение находится в 1НФ, так как каждая запись может содержать только одно значение для каждого из атрибутов.

Данное отношение также находится во 2НФ, так как оно находится в 1НФ, и все не ключевые атрибуты зависят от первичного ключа «Номер».

Данное отношение также находится в 3НФ, так как оно находится во 2НФ, и отсутствуют транзитивные зависимости неключевых атрибутов от первичного ключа «Номер».

Отношение не требует нормализации, так как оно уже находится в 3НФ.



## 4. ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

### 4.1. Общая схема и описание физической модели базы данных

Для всех отношений была описана логическая структура, которая содержит последовательность расположения атрибутов в отношении и их основные характеристики:

1. Отношение «Потребитель» имеет следующую структуру:

Таблица 4.1 – Структура отношения «Потребитель»

CONSUMERS					
Атрибут		Признак ключа	Формат атрибута		
Имя (обозначение)	Полное наименование		Тип	Длина	Ограничения
number	номер	PK	integer	8	NOT NULL
account	лицевой счёт	AK	char	10	NOT NULL
name	наименование	-	varchar	30	-
telephone	телефон	-	varchar	11	-
area	площадь	-	integer	8	NOT NULL
type	тип	-	enum	16	множество значений ENUM('Физ. лицо', 'Юр. лицо')
mail	электронная почта	-	varchar	30	-

2. Отношение «Физ. лицо» имеет следующую структуру:

Таблица 4.2 – Структура отношения «Физ. лицо»

NATURAL_PERSONS		
Атрибут		Формат атрибута

Имя (обозначение)	Полное наименован ие	Призна к ключа	Тип	Длин а	Ограничения
number	номер потребителя	PK, FK	integer	8	NOT NULL REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE
snils	СНИЛС	AK	varchar	11	NOT NULL
series_and_number	серия и номер паспорта	AK	varchar	10	NOT NULL
issued_by_whom	кем выдан	-	varchar	150	-
when_issued	когда выдан	-	date	-	-
birthday	дата рождения	-	date	-	-
lastname	фамилия	-	varchar	12	-
firstname	имя	-	varchar	12	-
patronymic	отчество	-	varchar	12	-

Для данного отношения были выбраны ограничения целостности ON UPDATE CASCADE и ON DELETE CASCADE, так как при изменении номера в отношении CONSUMERS необходимо также изменить соответствующее поле в данном отношении, а при удалении записи из отношения CONSUMERS необходимо удалить соответствующую запись данного отношения.

### 3. Отношение «Юр. лицо» имеет следующую структуру:

Таблица 4.3 – Структура отношения «Юр. лицо»

LEGAL_PERSONS		
Атрибут		Формат атрибута

Имя (обозначение)	Полное наименование	Признак ключа	Тип	Длина	Ограничения
number	номер потребителя	PK, FK	integer	8	NOT NULL REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE
ogrn	ОГРН	AK	varchar	14	NOT NULL
name	наименование	-	varchar	30	-

Для данного отношения были выбраны ограничения целостности ON UPDATE CASCADE и ON DELETE CASCADE, так как при изменении номера в отношении CONSUMERS необходимо также изменить соответствующее поле в данном отношении, а при удалении записи из отношения CONSUMERS необходимо удалить соответствующую запись данного отношения.

#### 4. Отношение «Квартира» имеет следующую структуру:

Таблица 4.4 – Структура отношения «Квартира»

FLATS					
Атрибут		Признак ключа	Формат атрибута		
Имя (обозначение)	Полное наименование		Тип	Длина	Ограничения
number	номер	PK	integer	8	NOT NULL
plan	план	-	longblob	-	-
district	район	-	varchar	30	-
street	улица	-	varchar	20	-
house_number	номер дома	-	integer	4	-
flat_number	номер квартиры	-	integer	4	-

5. Отношение «Потребитель\_Владеет\_Квартирой» имеет следующую структуру:

Таблица 4.5 – Структура отношения «Потребитель\_Владеет\_Квартирой»

CONSUMER_OWN_FLAT					
Атрибут		Признак ключа	Формат атрибута		
Имя (обозначение)	Полное наименование		Тип	Длина	Ограничения
number	номер документа	PK	integer	8	NOT NULL
consumer_number	номер потребителя	FK	integer	8	NOT NULL REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE
flat_number	номер квартиры	FK	integer	8	NOT NULL REFERENCES FLATS(number) ON DELETE CASCADE ON UPDATE CASCADE
own_date	дата вступления в собственность	-	date	-	-
end_own_date	дата окончания права собственности	-	date	-	-
document_type	тип документа	-	varchar	30	-

Для данного отношения были выбраны ограничения целостности ON UPDATE CASCADE и ON DELETE CASCADE, так как при изменении номера в отношении CONSUMERS или FLATS необходимо также изменить

соответствующее поле в данном отношении, а при удалении записи из отношения CONSUMERS или FLATS необходимо удалить соответствующую запись данного отношения.

6. Отношение «Тариф» имеет следующую структуру:

Таблица 4.6 – Структура отношения «Тариф»

RATES					
Атрибут		Признак ключа	Формат атрибута		
Имя (обозначение)	Полное наименование		Тип	Длина	Ограничения
number	номер	PK	integer	8	NOT NULL
start_date	дата ступления в силу	-	date	-	-
end_date	дата окончания	-	date	-	-
name	наименование	-	varchar	30	-
cost	стоимость	-	decimal	8	Точность 2

7. Отношение «Начисление» имеет следующую структуру:

Таблица 4.7 – Структура отношения «Начисление»

ACCURALS					
Атрибут		Признак ключа	Формат атрибута		
Имя (обозначение)	Полное наименование		Тип	Длина	Ограничения
number	номер	PK	integer	8	NOT NULL
rate_number	номер тарифа	FK	integer	8	NOT NULL REFERENCES RATES(number) ON DELETE CASCADE ON UPDATE CASCADE

consumer_number	номер потребителя	FK	integer	8	NOT NULL REFERENCES CONSUMERS(number) ON DELETE CASCADE ON UPDATE CASCADE
date_from	дата с	-	date	-	-
date_to	дата по	-	date	-	-
amount	сумма	-	decimal	10	NOT NULL, Точность 2

Для данного отношения были выбраны ограничения целостности ON UPDATE CASCADE и ON DELETE CASCADE, так как при изменении номера в отношении CONSUMERS или RATES необходимо также изменить соответствующее поле в данном отношении, а при удалении записи из отношения CONSUMERS или RATES необходимо удалить соответствующую запись данного отношения.

8. Отношение «Оплата» имеет следующую структуру:

Таблица 4.8 – Структура отношения «Оплата»

PAYMENTS					
Атрибут		Признак ключа	Формат атрибута		
Имя (обозначение)	Полное наименование		Тип	Длина	Ограничения
number	номер платёжного документа	PK	integer	8	NOT NULL
accural_number	номер начисления	FK	integer	8	NOT NULL REFERENCES ACCURALS(number) ON DELETE CASCADE ON UPDATE CASCADE

pay_date	дата оплаты	-	date	-	-
amount	сумма	-	decimal	10	NOT NULL Точность 2

Для данного отношения были выбраны ограничения целостности ON UPDATE CASCADE и ON DELETE CASCADE, так как при изменении номера в отношении ACCURALS необходимо также изменить соответствующее поле в данном отношении, а при удалении записи из отношения ACCURALS необходимо удалить соответствующую запись данного отношения.

## 4.2. Проектирование физической модели базы данных в выбранной СУБД

Были созданы запросы на создание отношений базы данных. Отношения базы данных представлены на рисунках 4.1 – 4.8.

### 1. SQL запрос на создание отношения «Потребитель»:

```
CREATE TABLE CONSUMERS (
    number integer(8) NOT NULL,
    account char(10) NOT NULL,
    name varchar(30),
    telephone varchar(11),
    area integer(8) NOT NULL,
    type ENUM('Физ. лицо', 'Юр. лицо'),
    mail varchar(30),
    PRIMARY KEY(number),
    UNIQUE(account)
);
```



	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1	<b>number</b> 	int			Нет	<i>Нет</i>
<input type="checkbox"/>	2	<b>account</b> 	char(10)	<i>utf8mb4_0900_ai_ci</i>		Нет	<i>Нет</i>
<input type="checkbox"/>	3	<b>name</b>	varchar(30)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/>	4	<b>telephone</b>	int			Да	<i>NULL</i>
<input type="checkbox"/>	5	<b>area</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/>	6	<b>type</b>	enum('Физ. лицо', 'Юр. лицо')	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/>	7	<b>mail</b>	varchar(30)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>

Рисунок 4.1 – Отношение «Потребитель»

## 2. SQL запрос на создание отношения «Физ. лицо»:

```

CREATE TABLE NATURAL_PERSONS (
    number integer(8) NOT NULL,
    snils varchar(11) NOT NULL,
    series_and_number varchar(10) NOT NULL,
    issued_by_whom varchar(150),
    when_issued date,
    birthday date,
    lastname varchar(12),
    firstname varchar(12),
    patronomic varchar(12),
    PRIMARY KEY(number),
    UNIQUE(snils),
    UNIQUE(series_and_number),
    FOREIGN KEY(number) REFERENCES CONSUMERS(number) ON DELETE CASCADE ON
    UPDATE CASCADE
);

```



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/> 1	<b>number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/> 2	<b>snils</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/> 3	<b>series_and_number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/> 4	<b>issued_by_whom</b>	varchar(150)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/> 5	<b>when_issued</b>	date			Да	<i>NULL</i>
<input type="checkbox"/> 6	<b>birthday</b>	date			Да	<i>NULL</i>
<input type="checkbox"/> 7	<b>lastname</b>	varchar(12)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/> 8	<b>firstname</b>	varchar(12)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/> 9	<b>patronymic</b>	varchar(12)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>

Рисунок 4.2 – Отношение «Физ. лицо»

### 3. SQL запрос на создание отношения «Юр. лицо»:

```

CREATE TABLE LEGAL_PERSONS (
    number integer(8) NOT NULL,
    ogrn varchar(14) NOT NULL,
    name varchar(30),
    PRIMARY KEY(number),
    UNIQUE(ogrn),
    FOREIGN KEY(number) REFERENCES CONSUMERS(number) ON DELETE CASCADE ON
    UPDATE CASCADE
);

```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/> 1	<b>number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/> 2	<b>ogrn</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/> 3	<b>name</b>	varchar(30)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>

Рисунок 4.3 – Отношение «Юр. лицо»

#### 4. SQL запрос на создание отношения «Квартира»:

```
CREATE TABLE FLATS (
    number integer(8) NOT NULL PRIMARY KEY,
    plan longblob,
    district varchar(30),
    street varchar(20),
    house_number integer(4),
    flat_number integer(4)
);
```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/> 1	<b>number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/> 2	<b>plan</b>	longblob			Да	<i>NULL</i>
<input type="checkbox"/> 3	<b>district</b>	varchar(30)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/> 4	<b>street</b>	varchar(20)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/> 5	<b>house_number</b>	int			Да	<i>NULL</i>
<input type="checkbox"/> 6	<b>flat_number</b>	int			Да	<i>NULL</i>

Рисунок 4.4 – Отношение «Квартира»

## 5. SQL запрос на создание отношения

### «Потребитель\_Владеет\_Квартирой»:

```
CREATE TABLE CONSUMER_OWN_FLAT (
    number integer(8) NOT NULL,
    consumer_number integer(8) NOT NULL,
    flat_number integer(8) NOT NULL,
    own_date date,
    end_own_date date,
    document_type varchar(30),
    PRIMARY KEY(number),
    FOREIGN KEY(consumer_number) REFERENCES CONSUMERS(number) ON DELETE
    CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(flat_number) REFERENCES FLATS(number) ON DELETE CASCADE ON
    UPDATE CASCADE
);
```




#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/> 1	<b>number</b> 	int			Нет	Нет
<input type="checkbox"/> 2	<b>consumer_number</b> 	int			Нет	Нет
<input type="checkbox"/> 3	<b>flat_number</b> 	int			Нет	Нет
<input type="checkbox"/> 4	<b>own_date</b>	date			Да	NULL
<input type="checkbox"/> 5	<b>end_own_date</b>	date			Да	NULL
<input type="checkbox"/> 6	<b>document_type</b>	varchar(30)	utf8mb4_0900_ai_ci		Да	NULL

Рисунок 4.5 – Отношение «Потребитель\_Владеет\_Квартирой»

## 6. SQL запрос на создание отношения «Тариф»:

```
CREATE TABLE RATES (
    number integer(8) NOT NULL PRIMARY KEY,
    start_date date,
    end_date date,
```

```

name varchar(30),
cost decimal(8,2)
);

```


#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/> 1	<b>number</b> 	int			Нет	<i>Нет</i>
<input type="checkbox"/> 2	<b>start_date</b>	date			Да	<i>NULL</i>
<input type="checkbox"/> 3	<b>end_date</b>	date			Да	<i>NULL</i>
<input type="checkbox"/> 4	<b>name</b>	varchar(30)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/> 5	<b>cost</b>	decimal(8,2)			Да	<i>NULL</i>

Рисунок 4.6 – Отношение «Тариф»

## 7. SQL запрос на создание отношения «Начисление»:

```

CREATE TABLE ACCURALS (
    number integer(8) NOT NULL,
    rate_number integer(8) NOT NULL,
    consumer_number integer(8) NOT NULL,
    date_from date,
    date_to date,
    amount decimal(10, 2) NOT NULL,
    PRIMARY KEY(number),
    FOREIGN KEY(rate_number) REFERENCES RATES(number) ON DELETE CASCADE ON
UPDATE CASCADE,
    FOREIGN KEY(consumer_number) REFERENCES CONSUMERS(number) ON DELETE
CASCADE ON UPDATE CASCADE
);

```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/> 1	<b>number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/> 2	<b>rate_number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/> 3	<b>consumer_number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/> 4	<b>date_from</b>	date			Да	<i>NULL</i>
<input type="checkbox"/> 5	<b>date_to</b>	date			Да	<i>NULL</i>
<input type="checkbox"/> 6	<b>amount</b>	decimal(10,2)			Нет	<i>Нет</i>

Рисунок 4.7 – Отношение «Начисление»

## 8. SQL запрос на создание отношения «Оплата»:

```
CREATE TABLE PAYMENTS (
    number integer(8) NOT NULL PRIMARY KEY,
    accrual_number integer(8) NOT NULL,
    pay_date date,
    amount decimal(10, 2) NOT NULL,
    FOREIGN KEY(accrual_number) REFERENCES ACCURALS(number) ON DELETE CASCADE
    ON UPDATE CASCADE
);
```

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/> 1	<b>number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/> 2	<b>accrual_number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/> 3	<b>pay_date</b>	date			Да	<i>NULL</i>
<input type="checkbox"/> 4	<b>amount</b>	decimal(10,2)			Нет	<i>Нет</i>

Рисунок 4.8 – Отношение «Начисление»

Также были созданы SQL-запросы на изменение структуры отношения, для чего была сделана временная копия таблицы RATES с названием RATES\_COPY. Было продемонстрировано умение работать с запросами

**ALTER TABLE:** добавление нового столбца после первого имеющегося, изменение типа данных в любом столбце, добавление ограничения целостности на любой столбец, удаление столбца.

### 1. Копирование таблицы с новым названием (рисунок 4.9):

```
CREATE TABLE RATES_COPY LIKE RATES;
```

Сервер: 127.0.0.1:3306 > База данных: HeatBD > Таблица: RATES\_COPY

СТРУКТУРА ТАБЛИЦЫ СВЯЗИ

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/> 1	<b>number</b>	int			Нет	Нет
<input type="checkbox"/> 2	<b>start_date</b>	date			Да	NULL
<input type="checkbox"/> 3	<b>end_date</b>	date			Да	NULL
<input type="checkbox"/> 4	<b>name</b>	varchar(30)	utf8mb4_0900_ai_ci		Да	NULL
<input type="checkbox"/> 5	<b>cost</b>	decimal(8,2)			Да	NULL

Рисунок 4.9 – Отношение «RATES\_COPY»

### 2. Добавление нового столбца после первого имеющегося (рисунок 4.10):

```
ALTER TABLE RATES_COPY
ADD new_column integer NOT NULL;
ALTER TABLE RATES_COPY
MODIFY new_column integer AFTER number;
```

	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1	<b>number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/>	2	<b>new_column</b>	int			Да	<i>NULL</i>
<input type="checkbox"/>	3	<b>start_date</b>	date			Да	<i>NULL</i>
<input type="checkbox"/>	4	<b>end_date</b>	date			Да	<i>NULL</i>
<input type="checkbox"/>	5	<b>name</b>	varchar(30)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/>	6	<b>cost</b>	decimal(8,2)			Да	<i>NULL</i>

Рисунок 4.10 – Добавлен столбец

### 3. Изменения типа данных в новом столбце (рисунок 4.11):

```
ALTER TABLE RATES_COPY
MODIFY new_column VARCHAR(12);
```

	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1	<b>number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/>	2	<b>new_column</b>	varchar(12)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/>	3	<b>start_date</b>	date			Да	<i>NULL</i>
<input type="checkbox"/>	4	<b>end_date</b>	date			Да	<i>NULL</i>
<input type="checkbox"/>	5	<b>name</b>	varchar(30)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/>	6	<b>cost</b>	decimal(8,2)			Да	<i>NULL</i>

Рисунок 4.11 – Изменён тип данных в столбце

### 4. Добавление ограничения на новый столбец (рисунок 4.12):

```
ALTER TABLE RATES_COPY
ADD CONSTRAINT UNIQUE(new_column),
ALTER new_column SET DEFAULT 'dojdik';
```

	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1	<b>number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/>	2	<b>new_column</b>	varchar(12)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>dojdik</i>
<input type="checkbox"/>	3	<b>start_date</b>	date			Да	<i>NULL</i>
<input type="checkbox"/>	4	<b>end_date</b>	date			Да	<i>NULL</i>
<input type="checkbox"/>	5	<b>name</b>	varchar(30)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>NULL</i>
<input type="checkbox"/>	6	<b>cost</b>	decimal(8,2)			Да	<i>NULL</i>

Рисунок 4.12 – Добавлено новое ограничение на столбец

## 5. Удаление столбца (рисунок 4.13):

```
ALTER TABLE RATES_COPY
```

```
DROP name;
```

	#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1	<b>number</b>	int			Нет	<i>Нет</i>
<input type="checkbox"/>	2	<b>new_column</b>	varchar(12)	<i>utf8mb4_0900_ai_ci</i>		Да	<i>dojdik</i>
<input type="checkbox"/>	3	<b>start_date</b>	date			Да	<i>NULL</i>
<input type="checkbox"/>	4	<b>end_date</b>	date			Да	<i>NULL</i>
<input type="checkbox"/>	5	<b>cost</b>	decimal(8,2)			Да	<i>NULL</i>

Рисунок 4.13 – Удалён столбец

Был создан SQL-запрос на удаление отношения RATES\_COPY:

```
DROP TABLE RATES_COPY;
```



Также был создана схема физической модели данных на основе выбранной СУБД (рисунок 4.14).

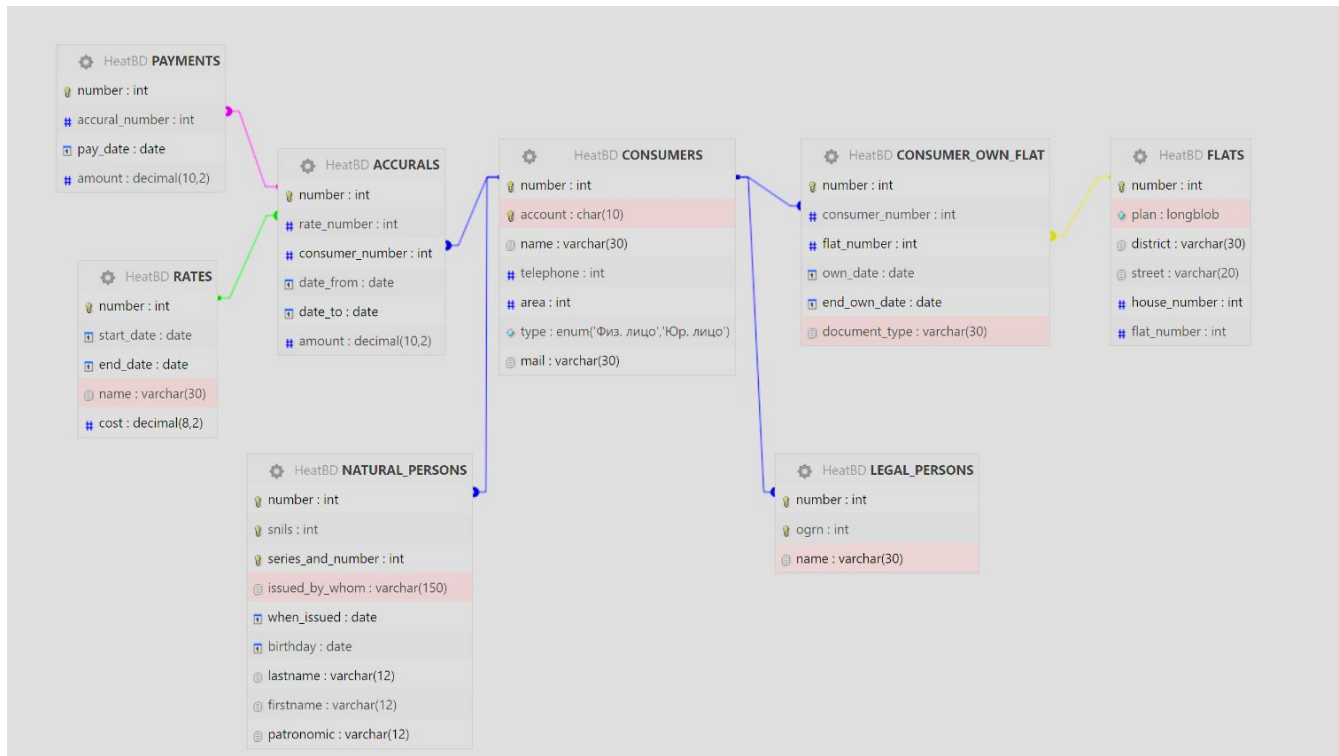


Рисунок 4.14 – Схема физической модели данных

## 5. ТЕСТИРОВАНИЕ БАЗЫ ДАННЫХ

### 5.1. Запросы на добавление данных в таблицы базы данных

Были созданы запросы на добавление данных во все таблицы. Результаты добавления данных приведены на рисунках 5.1 – 5.8.

#### SQL запрос на добавление данных в таблицу RATES:

```
INSERT INTO RATES VALUES (0, '2024-01-01', '2025-01-01', 'Землянка', 3080);
INSERT INTO RATES (start_date, end_date, number, cost) VALUES ('2024-02-01',
'2025-02-01', 1, 2490);
INSERT INTO RATES (number, start_date, end_date, name, cost) VALUES (2, '2024-
06-01', '2025-06-01', 'Тропики', 3666);
INSERT INTO RATES VALUES (3, '2024-03-01', '2025-01-01', 'Первый круг ада',
4010), (4, '2024-04-01', '2025-01-01', 'Второй круг ада', 4020), (5, '2024-05-
01', '2025-01-01', 'Третий круг ада', 4030);
```

#### SQL запрос на добавление данных в таблицу CONSUMERS:

```
INSERT INTO CONSUMERS (number, account, name, telephone, area, type, mail)
VALUES (0, '70ББ000584', 'Владимир', '79781195738', 40, 'Физ. лицо',
'misterincognito@mail.ru');
INSERT INTO CONSUMERS VALUES (1, '80ББ000584', 'Anonymous', '79788888888', 350,
'Физ. лицо', 'nevladimir@mail.ru');
INSERT INTO CONSUMERS (area, number, account) VALUES (56, 2, '90ББ000584');
INSERT INTO CONSUMERS VALUES (3, '77ББ000584', 'ООО Красавцы', '79787777777',
40, 'Юр. лицо', 'krasavci@mail.ru'), (4, '79ББ000584', 'ЗАО Свинец',
'79783333333', 40, 'Юр. лицо', 'svinec@mail.ru');
```

#### SQL запрос на добавление данных в таблицу FLATS:

```
INSERT INTO FLATS (number, plan, district, street, house_number, flat_number)
VALUES (0, LOAD_FILE('C:\Users\k_dod\Downloads\ггшл.jpeg'), 'Тагаринский',
'Пушкина', 4, 12);
INSERT INTO FLATS VALUES (1, LOAD_FILE('C:\Users\k_dod\Downloads\planirovka-
3komnatnoi-kvartiry-novostroiika.jpeg'), 'Ленинский', 'Колотушкина', 2, 15);
INSERT INTO FLATS (district, number) VALUES ('Киевский', 2);
INSERT INTO FLATS VALUES (3, LOAD_FILE('C:\Users\k_dod\Downloads\ггшл.jpeg'),
'Балаклавский', 'Киевская', 14, 15), (4,
LOAD_FILE('C:\Users\k_dod\Downloads\ггшл.jpeg'), 'Балаклавский', 'Киевская',
14, 16);
```

#### SQL запрос на добавление данных в таблицу CONSUMER\_OWN\_FLAT:

```
INSERT INTO CONSUMER_OWN_FLAT (number, consumer_number, flat_number, own_date,
end_own_date, document_type) VALUES (0, 0, 0, '2023-01-01', '2025-01-01',
'Договор аренды');
INSERT INTO CONSUMER_OWN_FLAT VALUES (1, 0, 1, '2023-01-01', '2025-01-01',
'Выписка из ЕГРН');
INSERT INTO CONSUMER_OWN_FLAT (consumer_number, flat_number, number) VALUES (1,
2, 2);
```

```
INSERT INTO CONSUMER_OWN_FLAT (number, consumer_number, flat_number) VALUES (3, 2, 1), (4, 3, 3), (5, 4, 4);
```

### SQL запрос на добавление данных в таблицу LEGAL\_PERSONS:

```
INSERT INTO LEGAL_PERSONS (number, ogrn, name) VALUES (3, 1027739642281, "ООО Красавцы");
INSERT INTO LEGAL_PERSONS (ogrn, number) VALUES (8327779642299, 4);
```

### SQL запрос на добавление данных в таблицу NATURAL\_PERSONS:

```
INSERT INTO NATURAL_PERSONS (number, snils, series_and_number, issued_by_whom, when_issued, birthday, lastname, firstname, patronomic) VALUES (0, '48095351208', '0913599987', 'УМВД России по г. Севастополю', '1985-03-04', '1971-03-04', 'Пейпиво', 'Виктор', 'Альбертович');
INSERT INTO NATURAL_PERSONS (snils, number, series_and_number) VALUES ('55544333208', 1, '0817666333');
```

Также для корректной работы был создан триггер для автоматического подсчёта суммы начисления.

### SQL запрос на создание триггера для таблицы ACCURALS:

```
CREATE TRIGGER `NEW_ACCURAL_BEFORE_INSERT`
BEFORE INSERT ON `ACCURALS`
FOR EACH ROW
set NEW.amount = (SELECT cost*(SELECT area FROM CONSUMERS WHERE CONSUMERS.number = NEW.consumer_number) FROM RATES WHERE RATES.number = NEW.rate_number)
```

### SQL запрос на добавление данных в таблицу ACCURALS:

```
INSERT INTO ACCURALS (number, rate_number, consumer_number, date_from, date_to) VALUES (0, 0, 0, '2024-01-01', '2024-02-01'), (1, 1, 1, '2024-01-01', '2024-02-01');
INSERT INTO ACCURALS (rate_number, number, consumer_number) VALUES (2, 2, 2), (3, 3, 3), (4, 4, 4);
```

### SQL запрос на добавление данных в таблицу PAYMENTS:

```
INSERT INTO PAYMENTS (number, accural_number, pay_date, amount) VALUES (0, 0, '2024-02-02', 123200);
INSERT INTO PAYMENTS VALUES (1, 1, '2024-02-02', 871500);
INSERT INTO PAYMENTS (accural_number, number, amount) VALUES (2, 2, 205296);
INSERT INTO PAYMENTS (number, accural_number, amount) VALUES (3, 3, 160400), (4, 4, 160800);
```



























		number	start_date	end_date	name	cost
   		0	2024-01-01	2025-01-01	Землянка	3080.00
   		1	2024-02-01	2025-02-01	NULL	2490.00
   		2	2024-06-01	2025-06-01	Тропики	3666.00
   		3	2024-03-01	2025-01-01	Первый круг ада	4010.00
   		4	2024-04-01	2025-01-01	Второй круг ада	4020.00
   		5	2024-05-01	2025-01-01	Третий круг ада	4030.00

Рисунок 5.1 – Таблица RATES

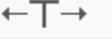





















		number	account	name	telephone	area	type	mail
   		0	70Б5000584	Владимир	79781195738	40	Физ. лицо	misterincognito@mail.ru
   		1	80Б5000584	Anonymous	79788888888	350	Физ. лицо	nevladimir@mail.ru
   		2	90Б5000584	NULL	NULL	56	NULL	NULL
   		3	77Б5000584	ООО Красавцы	79787777777	40	Юр. лицо	krasavci@mail.ru
   		4	79Б5000584	ЗАО Свинец	79783333333	40	Юр. лицо	svinec@mail.ru

Рисунок 5.2 – Таблица CONSUMERS























		number	plan	district	street	house_number	flat_number
   		0	[BLOB - 197.6 КиБ]	Гагаринский	Пушкина	4	12
   		1	[BLOB - 36.9 КиБ]	Ленинский	Колотушкина	2	15
   		2	NULL	Киевский	NULL	NULL	NULL
   		3	[BLOB - 197.6 КиБ]	Балаклавский	Киевская	14	15
   		4	[BLOB - 197.6 КиБ]	Балаклавский	Киевская	14	16

Рисунок 5.3 – Таблица FLATS



























		number	consumer_number	flat_number	own_date	end_own_date	document_type
   		0	0	0	2023-01-01	2025-01-01	Договор аренды
   		1	0	1	2023-01-01	2025-01-01	Выписка из ЕГРН
   		2	1	2	NULL	NULL	NULL
   		3	2	1	NULL	NULL	NULL
   		4	3	3	NULL	NULL	NULL
   		5	4	4	NULL	NULL	NULL

Рисунок 5.4 – Таблица CONSUMER\_OWN\_FLAT







		number	ogrn	name
<input type="checkbox"/>				3 1027739642281 ООО Красавцы
<input type="checkbox"/>				4 8327779642299 NULL

Рисунок 5.5 – Таблица LEGAL\_PERSON









		number	snils	series_and_number	issued_by_whom	when_issued	birthday	lastname	firstname	patronymic
<input type="checkbox"/>				0 48095351208 0913599987	УМВД России по г. Севастополю	1985-03-04	1971-03-04	Пейпиво	Виктор	Альбертович
<input type="checkbox"/>				1 55544333208 0817666333	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 5.6 – Таблица NATURAL\_PERSON







		number	rate_number	consumer_number	date_from	date_to	amount
<input type="checkbox"/>				0	0	0 2024-01-01 2024-02-01	123200.00
<input type="checkbox"/>				1	1	1 2024-01-01 2024-02-01	871500.00
<input type="checkbox"/>				2	2	2 NULL NULL	205296.00
<input type="checkbox"/>				3	3	3 NULL NULL	160400.00
<input type="checkbox"/>				4	4	4 NULL NULL	160800.00

Рисунок 5.7 – Таблица ACCURALS


















		number	accural_number	pay_date	amount
<input type="checkbox"/>				0	0 2024-02-02 123200.00
<input type="checkbox"/>				1	1 2024-02-02 871500.00
<input type="checkbox"/>				2	2 NULL 205296.00
<input type="checkbox"/>				3	3 NULL 160400.00
<input type="checkbox"/>				4	4 NULL 160800.00

Рисунок 5.8 – Таблица PAYMENTS

## 5.2. Тестирование базы данных на соблюдение ссылочной целостности

Были созданы запросы на изменение и удаление данных, которые могут привести к нарушениям ссылочной целостности. Также было сохранено состояние таблиц до и после отправки запросов.

1) Изменить номер записи о владении квартирой с 5 на 6 (рисунки 5.9-5.10):

```
UPDATE CONSUMER_OWN_FLAT SET number = 6 WHERE number = 5;
```

		number	consumer_number	flat_number	own_date	end_own_date	document_type
<input type="checkbox"/>		0	0	0	2023-01-01	2025-01-01	Договор аренды
<input type="checkbox"/>		1	0	1	2023-01-01	2025-01-01	Выписка из ЕГРН
<input type="checkbox"/>		2	1	2	NULL	NULL	NULL
<input type="checkbox"/>		3	2	1	NULL	NULL	NULL
<input type="checkbox"/>		4	3	3	NULL	NULL	NULL
<input type="checkbox"/>		5	4	4	NULL	NULL	NULL

Рисунок 5.9 – Таблица CONSUMER\_OWN\_FLAT до первого изменения

		number	consumer_number	flat_number	own_date	end_own_date	document_type
<input type="checkbox"/>		0	0	0	2023-01-01	2025-01-01	Договор аренды
<input type="checkbox"/>		1	0	1	2023-01-01	2025-01-01	Выписка из ЕГРН
<input type="checkbox"/>		2	1	2	NULL	NULL	NULL
<input type="checkbox"/>		3	2	1	NULL	NULL	NULL
<input type="checkbox"/>		4	3	3	NULL	NULL	NULL
<input type="checkbox"/>		6	4	4	NULL	NULL	NULL

Рисунок 5.10 – Таблица CONSUMER\_OWN\_FLAT после первого изменения

В данном запросе изменяется первичный ключ. Так как записи с таким ключом в этой таблицы ещё не существовало – такой запрос возможно выполнить.

## 2) Изменить у записи о владении квартирой с номером 6 номер квартиры на 2 (рисунок 5.11):

```
UPDATE CONSUMER_OWN_FLAT SET flat_number = 2 WHERE number = 6;
```

		number	consumer_number	flat_number	own_date	end_own_date	document_type
<input type="checkbox"/>		0	0	0	2023-01-01	2025-01-01	Договор аренды
<input type="checkbox"/>		1	0	1	2023-01-01	2025-01-01	Выписка из ЕГРН
<input type="checkbox"/>		2	1	2	NULL	NULL	NULL
<input type="checkbox"/>		3	2	1	NULL	NULL	NULL
<input type="checkbox"/>		4	3	3	NULL	NULL	NULL
<input type="checkbox"/>		6	4	2	NULL	NULL	NULL

Рисунок 5.11 – Таблица CONSUMER\_OWN\_FLAT после второго изменения

В данном запросе изменяется внешний ключ. Так как таблица, с которой происходит работа, является зависимой, и запись с таким ключом в основной таблице существует – такой запрос возможно выполнить.

## 3) Изменить номер нулевой квартиры на 5 (рисунки 5.12-5.14).

```
UPDATE FLATS SET number = 5 WHERE number = 0;
```

		number	plan	district	street	house_number	flat_number
<input type="checkbox"/>		0	[BLOB - 197.6 КиБ]	Гагаринский	Пушкина	4	12
<input type="checkbox"/>		1	[BLOB - 36.9 КиБ]	Ленинский	Колотушкина	2	15
<input type="checkbox"/>		2	NULL	Киевский	NULL	NULL	NULL
<input type="checkbox"/>		3	[BLOB - 197.6 КиБ]	Балаклавский	Киевская	14	15
<input type="checkbox"/>		4	[BLOB - 197.6 КиБ]	Балаклавский	Киевская	14	16

Рисунок 5.12 – Таблица FLATS до первого изменения

<div><div><div>←</div><div>T</div><div>→</div></div><div>▼</div></div>	number	plan	district	street	house_number	flat_number
<div><div><div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div></div></div></div> <div>1</div> <div>[BLOB - 36.9 КиБ]</div> <div>Ленинский</div> <div>Колотушкина</div> <div>2</div> <div>15</div>						
<div><div><div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div></div></div></div> <div>2</div> <div>NULL</div> <div>Киевский</div> <div>NULL</div> <div>NULL</div> <div>NULL</div>						
<div><div><div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div></div></div></div> <div>3</div> <div>[BLOB - 197.6 КиБ]</div> <div>Балаклавский</div> <div>Киевская</div> <div>14</div> <div>15</div>						
<div><div><div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div></div></div></div> <div>4</div> <div>[BLOB - 197.6 КиБ]</div> <div>Балаклавский</div> <div>Киевская</div> <div>14</div> <div>16</div>						
<div><div><div><div></div></div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div><div><div></div></div></div></div> <div>5</div> <div>[BLOB - 197.6 КиБ]</div> <div>Гагаринский</div> <div>Пушкина</div> <div>4</div> <div>12</div>						

Рисунок 5.13 – Таблица FLATS после первого изменения

<div><div><div>←T→</div><div>▼</div></div></div>				number	consumer_number	flat_number	own_date	end_own_date	document_type	
<div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div>					0	0	5	2023-01-01	2025-01-01	Договор аренды
<div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div>					1	0	1	2023-01-01	2025-01-01	Выписка из ЕГРН
<div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div>					2	1	2	NULL	NULL	NULL
<div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div>					3	2	1	NULL	NULL	NULL
<div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div>					4	3	3	NULL	NULL	NULL
<div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div>					6	4	2	NULL	NULL	NULL

Рисунок 5.14 – Таблица CONSUMER\_OWN\_FLAT после первого изменения FLATS

В данном запросе изменяется первичный ключ. Так как записи с таким ключом в этой таблицы ещё не существовало – такой запрос возможно выполнить. Так как для дополнительной таблицы установлено каскадное изменения данных – внешний ключ в соответствующей записи дополнительной таблицы автоматически изменится на новый.

#### 4) Изменить номер квартиры с 5 на 6 (рисунки 5.15-5.16).

```
UPDATE FLATS SET number = 6 WHERE number = 5;
```






















   	number	plan	district	street	house_number	flat_number
<input type="checkbox"/>   	1	[BLOB - 36.9 КиБ]	Ленинский	Колотушкина	2	15
<input type="checkbox"/>   	2	NULL	Киевский	NULL	NULL	NULL
<input type="checkbox"/>   	3	[BLOB - 197.6 КиБ]	Балаклавский	Киевская	14	15
<input type="checkbox"/>   	4	[BLOB - 197.6 КиБ]	Балаклавский	Киевская	14	16
<input type="checkbox"/>   	6	[BLOB - 197.6 КиБ]	Гагаринский	Пушкина	4	12

Рисунок 5.15 – Таблица FLATS после второго изменения























   	number	consumer_number	flat_number	own_date	end_own_date	document_type
<input type="checkbox"/>   	0	0	6	2023-01-01	2025-01-01	Договор аренды
<input type="checkbox"/>   	1	0	1	2023-01-01	2025-01-01	Выписка из ЕГРН
<input type="checkbox"/>   	2	1	2	NULL	NULL	NULL
<input type="checkbox"/>   	3	2	1	NULL	NULL	NULL
<input type="checkbox"/>   	4	3	3	NULL	NULL	NULL
<input type="checkbox"/>   	6	4	2	NULL	NULL	NULL

Рисунок 5.16 – Таблица CONSUMER\_OWN\_FLAT после второго изменения  
FLATS

В данном запросе изменяется первичный ключ. Так как записи с таким ключом в этой таблицы ещё не существовало – такой запрос возможно выполнить. Так как для дополнительной таблицы установлено каскадное изменения данных – внешний ключ в соответствующей записи дополнительной таблицы автоматически изменится на новый.

5) Удалить квартиру с номером 6 (рисунки 5.17-5.18).

```
DELETE FROM FLATS WHERE number = 6;
```

<div><div><div>←</div><div>T</div><div>→</div></div><div><div></div><div></div></div></div>				number	plan	district	street	house_number	flat_number
<div><div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div></div></div>	1	[BLOB - 36.9 КиБ]	Ленинский	Колотушкина	2	15			
<div><div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div></div></div>	2	NULL	Киевский	NULL	NULL	NULL			
<div><div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div></div></div>	3	[BLOB - 197.6 КиБ]	Балаклавский	Киевская	14	15			
<div><div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div></div></div>	4	[BLOB - 197.6 КиБ]	Балаклавский	Киевская	14	16			

Рисунок 5.17 – Таблица FLATS после первого удаления

<div><div><div>←T→</div><div>▼</div></div></div>				number	consumer_number	flat_number	own_date	end_own_date	document_type	
<div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div>					1	0	1	2023-01-01	2025-01-01	Выписка из ЕГРН
<div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div>					2	1	2	NULL	NULL	NULL
<div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div>					3	2	1	NULL	NULL	NULL
<div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div>					4	3	3	NULL	NULL	NULL
<div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div><div><div><div></div></div></div></div>					6	4	2	NULL	NULL	NULL

Рисунок 5.18 – Таблица CONSUMER\_OWN\_FLAT после первого удаления из FLATS

В этом запросе удаляется запись из основной таблицы. Так как для дополнительной таблицы установлено каскадное удаления данных – соответствующая запись в дополнительной таблице автоматически удалится.

#### б) Удалить запись о третьей квартире (рисунки 5.19-5.20).

```
DELETE FROM FLATS WHERE number = 3;
```

<div><div><div>←T→</div><div>▼</div></div></div>				number	plan	district	street	house_number	flat_number
<div><div><div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div></div></div></div>	1	[BLOB - 36.9 КиБ]	Ленинский	Колотушкина	2	15			
<div><div><div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div></div></div></div>	2	NULL	Киевский	NULL	NULL	NULL			
<div><div><div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div><div><div></div></div></div><div><div><div></div></div></div></div></div>	4	[BLOB - 197.6 КиБ]	Балаклавский	Киевская	14	16			

Рисунок 5.19 – Таблица FLATS после второго удаления

<div><div><div><div>←</div><div>T</div><div>→</div></div><div>▼</div></div></div>				number	consumer_number	flat_number	own_date	end_own_date	document_type
<div><div><div><div><div></div></div><div><div></div><div></div><div></div></div></div><div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div></div></div></div>		1	0	1	2023-01-01	2025-01-01	Выписка из ЕГРН		
<div><div><div><div><div></div></div><div><div></div><div></div><div></div></div></div><div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div></div></div></div>		2	1	2	NULL	NULL	NULL		
<div><div><div><div><div></div></div><div><div></div><div></div><div></div></div></div><div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div></div></div></div>		3	2	1	NULL	NULL	NULL		
<div><div><div><div><div></div></div><div><div></div><div></div><div></div></div></div><div><div><div></div><div></div><div></div></div><div><div></div><div></div><div></div></div></div></div></div>		6	4	2	NULL	NULL	NULL		

Рисунок 5.20 – Таблица CONSUMER\_OWN\_FLAT после второго удаления из FLATS

В этом запросе удаляется запись из основной таблицы. Так как для дополнительной таблицы установлено каскадное удаления данных – соответствующая запись в дополнительной таблице автоматически удалится.

7) Удалить шестую запись о владении квартирой (рисунок 5.21).

```
DELETE FROM CONSUMER_OWN_FLAT WHERE number = 6;
```

<div><div>←T→</div><div>▼</div></div>				number	consumer_number	flat_number	own_date	end_own_date	document_type
<div><div><input type="checkbox"/></div><div></div><div></div><div></div></div>	1	0	1	2023-01-01	2025-01-01	Выписка из ЕГРН			
<div><div><input type="checkbox"/></div><div></div><div></div><div></div></div>	2	1	2	NULL	NULL	NULL			
<div><div><input type="checkbox"/></div><div></div><div></div><div></div></div>	3	2	1	NULL	NULL	NULL			

Рисунок 5.21 – Таблица CONSUMER\_OWN\_FLAT после первого удаления

В этом запросе удаляется запись из дополнительной таблицы. Так как таблица дополнительная – удаление данных из неё никак не затронет основную таблицу.

8) Удалить записи о владении квартирой потребителя с номером 2 (рисунок 5.22).

```
DELETE FROM CONSUMER_OWN_FLAT WHERE consumer_number = 2;
```

	number	consumer_number	flat_number	own_date	end_own_date	document_type
<input type="checkbox"/>	1	0	1	2023-01-01	2025-01-01	Выписка из ЕГРН
<input type="checkbox"/>	2	1	2	NULL	NULL	NULL

Рисунок 5.22 – Таблица CONSUMER\_OWN\_FLAT после второго удаления

В этом запросе удаляется запись из дополнительной таблицы. Так как таблица дополнительная – удаление данных из неё никак не затронет основную таблицу.

### 5.3. Тестирование базы данных на возможность выборки из одного источника данных

Были созданы запросы на выборку данных из таблицы CONSUMERS (рисунок 5.23). Результаты выполнения запросов представлены на рисунках 5.24-5.33.

	number	account	name	telephone	area	type	mail
<input type="checkbox"/>	0	70ББ000584	Владимир	79781195738	40	Физ. лицо	misterincognito@mail.ru
<input type="checkbox"/>	1	80ББ000584	Anonymous	79788888888	350	Физ. лицо	nevladimir@mail.ru
<input type="checkbox"/>	2	90ББ000584	NULL	NULL	56	NULL	NULL
<input type="checkbox"/>	3	77ББ000584	ООО Красавцы	79787777777	40	Юр. лицо	krasavci@mail.ru
<input type="checkbox"/>	4	79ББ000584	ЗАО Свинец	79783333333	40	Юр. лицо	svinec@mail.ru

Рисунок 5.23 – Таблица CONSUMERS

#### 1) Вывести имена со счетами всех потребителей:

```
SELECT name, account FROM CONSUMERS;
```

				name	account
<input type="checkbox"/>				Владимир	70ББ000584
<input type="checkbox"/>				Anonymous	80ББ000584
<input type="checkbox"/>				NULL	90ББ000584
<input type="checkbox"/>				ООО Красавцы	77ББ000584
<input type="checkbox"/>				ЗАО Свинец	79ББ000584

Рисунок 5.24 – Имена и счета потребителей

2) Вывести все имеющиеся размеры площади потребителей:

```
SELECT DISTINCT AREA FROM CONSUMERS;
```

				AREA
<input type="checkbox"/>				40
<input type="checkbox"/>				350
<input type="checkbox"/>				56

Рисунок 5.25 – Площадь

3) Вывести всё о потребителях, владеющих площадью больше 50 метров:

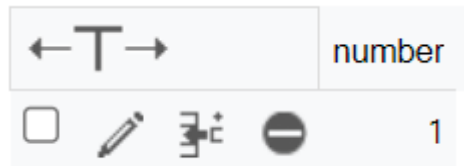
```
SELECT * FROM consumers WHERE area > 50;
```

				number	account	name	telephone	area	type	mail
<input type="checkbox"/>				1	80ББ000584	Anonymous	79788888888	350	Физ. лицо	nevladimir@mail.ru
<input type="checkbox"/>				2	90ББ000584	NULL	NULL	56	NULL	NULL

Рисунок 5.26 – Сравнение площадей

- 4) Вывести номера потребителей, владеющих площадью больше 50 метров и указавших почту:

```
SELECT number FROM CONSUMERS WHERE area > 50 and mail IS NOT NULL;
```

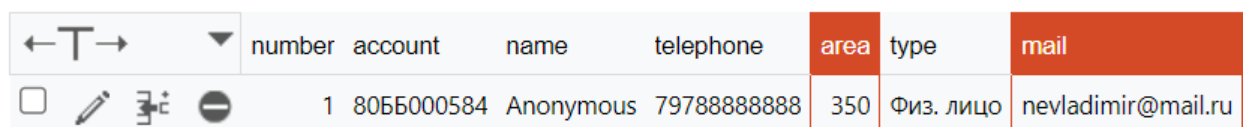


←T→	number
<input type="checkbox"/>	1

Рисунок 5.27 – Номера потребителей

- 5) Вывести всё о потребителях, владеющих площадью больше 50 метров и с почтовым доменом mail.ru:

```
SELECT * FROM CONSUMERS WHERE mail LIKE '%@mail.ru' AND area > 50;
```

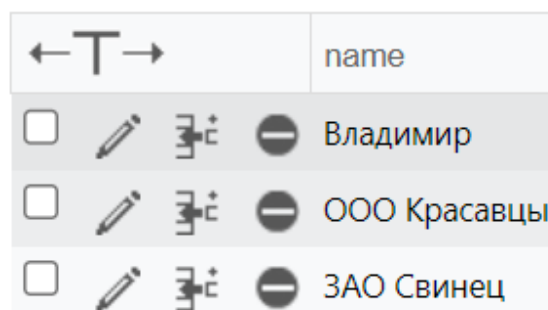


←T→	number	account	name	telephone	area	type	mail
<input type="checkbox"/>	1	80ББ000584	Anonymous	79788888888	350	Физ. лицо	nevladimir@mail.ru

Рисунок 5.28 – Потребители с mail.ru

- 6) Вывести имена потребителей, владеющих менее чем 50 метрами площади:

```
SELECT name FROM CONSUMERS WHERE NOT area > 50;
```



←T→	name
<input type="checkbox"/>	Владимир
<input type="checkbox"/>	ООО Красавцы
<input type="checkbox"/>	ЗАО Свинец

Рисунок 5.29 – Имена потребителей

- 7) Вывести какой суммарной площадью владеют типы потребителей:

```
SELECT type, SUM(area) FROM CONSUMERS GROUP BY type;
```

				type	SUM(area)
<input type="checkbox"/>				Физ. лицо	390
<input type="checkbox"/>				NULL	56
<input type="checkbox"/>				Юр. лицо	80

Рисунок 5.30 – Типы потребителей с площадью

- 8) Вывести какой суммарной площадью владеют типы потребителей с указанным типом:

```
SELECT type, SUM(area) FROM CONSUMERS GROUP BY type HAVING type IS NOT NULL;
```

				type	SUM(area)
<input type="checkbox"/>				Физ. лицо	390
<input type="checkbox"/>				Юр. лицо	80

Рисунок 5.31 – Типы потребителей без NULL с площадью

- 9) Подсчитать общее количество потребителей:

```
SELECT COUNT(*) AS 'Consumers amount' FROM CONSUMERS;
```

Consumers amount
5

Рисунок 5.32 – Общее количество потребителей

- 10) Вывести лицевой счёт с 20% площади каждого потребителя:

```
SELECT account, area*0.2 FROM CONSUMERS;
```


















		account	area*0.2
<input type="checkbox"/>   		70ББ000584	8.0
<input type="checkbox"/>   		80ББ000584	70.0
<input type="checkbox"/>   		90ББ000584	11.2
<input type="checkbox"/>   		77ББ000584	8.0
<input type="checkbox"/>   		79ББ000584	8.0

Рисунок 5.33 – 20% площади

#### 5.4. Тестирование базы данных на возможность выборки из нескольких источников данных

Были созданы запросы на выборку данных из базы данных. Структура таблиц, к которым происходили запросы представлены на рисунках 5.34 – 5.37. Результаты выполнения запросов представлены на рисунках 5.38 – 5.46.

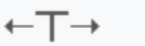
















		number	account	name	telephone	area	type	mail
<input type="checkbox"/>   		0	70ББ000584	Владимир	79781195738	40	Физ. лицо	misterincognito@mail.ru
<input type="checkbox"/>   		1	80ББ000584	Anonymous	79788888888	350	Физ. лицо	nevladimir@mail.ru
<input type="checkbox"/>   		2	90ББ000584	NULL	NULL	56	NULL	NULL
<input type="checkbox"/>   		3	77ББ000584	ООО Красавцы	79787777777	40	Юр. лицо	krasavci@mail.ru
<input type="checkbox"/>   		4	79ББ000584	ЗАО Свинец	79783333333	40	Юр. лицо	svinec@mail.ru

Рисунок 5.34 – Таблица CONSUMERS



←T→		number	rate_number	consumer_number	date_from	date_to	amount
<input type="checkbox"/>				0	0	0 2024-01-01 2024-02-01	123200.00
<input type="checkbox"/>				1	1	1 2024-01-01 2024-02-01	871500.00
<input type="checkbox"/>				2	2	2 NULL NULL	205296.00
<input type="checkbox"/>				3	3	3 NULL NULL	160400.00
<input type="checkbox"/>				4	4	4 NULL NULL	160800.00

Рисунок 5.35 – Таблица ACCURALS

←T→		number	start_date	end_date	name	cost
<input type="checkbox"/>				0	2024-01-01 2025-01-01	Землянка 3080.00
<input type="checkbox"/>				1	2024-02-01 2025-02-01	NULL 2490.00
<input type="checkbox"/>				2	2024-06-01 2025-06-01	Тропики 3666.00
<input type="checkbox"/>				3	2024-03-01 2025-01-01	Первый круг ада 4010.00
<input type="checkbox"/>				4	2024-04-01 2025-01-01	Второй круг ада 4020.00
<input type="checkbox"/>				5	2024-05-01 2025-01-01	Третий круг ада 4030.00

Рисунок 5.36 – Таблица RATES

←T→		number	accural_number	pay_date	amount
<input type="checkbox"/>				0	0 2024-02-02 123200.00
<input type="checkbox"/>				1	1 2024-02-02 871500.00
<input type="checkbox"/>				2	2 2024-02-02 205296.00
<input type="checkbox"/>				3	3 2024-02-02 160400.00
<input type="checkbox"/>				4	4 2024-02-02 160800.00

Рисунок 5.37 – Таблица PAYMENTS

1,2) Вывести имена и сумму начисления всех потребителей:

```
SELECT name, amount FROM CONSUMERS JOIN ACCURALS on CONSUMERS.number = consumer_number;
```

```
SELECT name, amount FROM CONSUMERS, ACCURALS WHERE CONSUMERS.number = consumer_number;
```

name	amount
Владимир	123200.00
Anonymous	871500.00
NULL	205296.00
ООО Красавцы	160400.00
ЗАО Свинец	160800.00

Рисунок 5.38 – Имена и суммы потребителей

3,4) Вывести имена и сумму начисления, а также название тарифа всех потребителей:

```
SELECT CONSUMERS.name, amount, RATES.name FROM (CONSUMERS JOIN ACCURALS on
CONSUMERS.number = ACCURALS.consumer_number) JOIN RATES on RATES.number =
ACCURALS.rate_number;
```

```
SELECT CONSUMERS.name, amount, RATES.name FROM CONSUMERS, ACCURALS, RATES
WHERE CONSUMERS.number = ACCURALS.consumer_number AND RATES.number =
ACCURALS.rate_number;
```

name	amount	name
Владимир	123200.00	Землянка
Anonymous	871500.00	NULL
NULL	205296.00	Тропики
ООО Красавцы	160400.00	Первый круг ада
ЗАО Свинец	160800.00	Второй круг ада

Рисунок 5.39 – Имена потребителей, суммы начисления и названия тарифов

5) Вывести всё о потребителях, владеющих площадью больше средней:

```
SELECT name, area FROM CONSUMERS _outer WHERE area >= (SELECT AVG(area) FROM
CONSUMERS _inner);
```



←T→	name	area
<input type="checkbox"/>  	Anonymous	350

Рисунок 5.40 – Потребителя с площадью больше средней

6) Вывести всё о начислении, название тарифа которого “Землянка”:

```
SELECT * FROM ACCURALS WHERE rate_number = (SELECT number FROM RATES WHERE
name = 'Землянка');
```

	number	rate_number	consumer_number	date_from	date_to	amount
	0	0	0	2024-01-01	2024-02-01	123200.00

Рисунок 5.41 – Начисление

7) Вывести всё о начислениях, тарифы которых имеют названия:

```
SELECT * FROM ACCURALS WHERE rate_number IN (SELECT number FROM RATES WHERE name IS NOT NULL);
```

	number	rate_number	consumer_number	date_from	date_to	amount
	0	0	0	2024-01-01	2024-02-01	123200.00
	2	2	2	NULL	NULL	205296.00
	3	3	3	NULL	NULL	160400.00
	4	4	4	NULL	NULL	160800.00

Рисунок 5.42 – Начисления

8) Вывести всё о тарифах, которые использовались в начислениях с суммой больше средней:

```
SELECT * FROM RATES WHERE number = ANY(SELECT rate_number FROM ACCURALS WHERE amount >= (SELECT AVG(amount) FROM ACCURALS));
```

	number	start_date	end_date	name	cost
	1	2024-02-01	2025-02-01	NULL	2490.00

Рисунок 5.43 – Тарифы

9) Вывести количество потребителей по площадям:

```
SELECT area, COUNT(*) AS cons_amount FROM CONSUMERS WHERE 2 <= (SELECT COUNT(*) FROM CONSUMER_OWN_FLAT WHERE consumer_number = number) GROUP BY area;
```

area	cons_amount
40	2
350	1
56	1

Рисунок 5.44 – Площади с количеством потребителей

- 10) Вывести всё о начислениях, в который тариф был начислен более чем на 100 метров:

```
SELECT * FROM ACCURALS WHERE amount > (SELECT cost*100 FROM RATES WHERE RATES.number = rate_number);
```

number	rate_number	consumer_number	date_from	date_to	amount
1	1	1	2024-01-01	2024-02-01	871500.00

Рисунок 5.45 – Начисление более чем на сто метров

- 11) Показатели оплаты услуг за отчетное полугодие по домам:

```
SELECT street, district, house_number, SUM(PAYMENTS.amount)
FROM FLATS, ACCURALS, PAYMENTS, CONSUMER_OWN_FLAT, CONSUMERS
WHERE PAYMENTS.pay_date >= '2024-01-01' AND FLATS.number =
CONSUMER_OWN_FLAT.flat_number AND CONSUMER_OWN_FLAT.consumer_number =
CONSUMERS.number AND ACCURALS.consumer_number = CONSUMERS.number AND
PAYMENTS.accural_number = ACCURALS.number
GROUP BY street, district, house_number;
```

street	district	house_number	SUM(PAYMENTS.amount)
Колотушкина	Ленинский	2	283600.00
NULL	Киевский	NULL	871500.00
Киевская	Балаклавский	14	205296.00

Рисунок 5.46 – Показатели оплаты услуг

## 5.5 SQL-запросы на создание пользовательских представлений

Для базы данных были сформированы и выполнены следующие запросы:

1) SQL-запрос на создание представления для ввода данных в таблицу (с использованием ограничений на ввод) (рисунок 5.47):

```
CREATE VIEW EXPENSIVE_RATES AS
SELECT number, name, cost
FROM RATES WHERE cost > 4000
WITH CHECK OPTION;
```

2) SQL-запрос, который продемонстрирует успешный ввод данных через созданное представление (с учетом заданных ограничений на ввод) (рисунок 5.48):

```
INSERT INTO expensive_rates (number, name, cost) VALUES (7, 'Дорогой', 5000);
```

3) SQL-запрос, который продемонстрирует отрицательный результат при вводе данных через созданное представление (с учетом заданных ограничений на ввод) (рисунок 5.49):

```
INSERT INTO expensive_rates (number, name, cost) VALUES (7, 'Не Дорогой', 3500);
```

4) SQL-запрос на удаление созданного представления:

```
DROP VIEW expensive_rates;
```

5) SQL-запрос на создание представления для вывода неполных данных из двух или более таблиц одновременно на основе заданного условия (рисунок 5.50):

```
CREATE VIEW ACCURALS_AND_RATES AS
SELECT ACCURALS.number AS 'accural number', amount AS 'for all' , cost AS
'for 1 m', RATES.number AS 'rate number'
FROM ACCURALS JOIN RATES ON ACCURALS.rate_number = RATES.number;
```



Рисунок 5.47 – Представление expensive\_rates

	number	start_date	end_date	name	cost
<input type="checkbox"/>	0	2024-01-01	2025-01-01	Землянка	3080.00
<input type="checkbox"/>	1	2024-02-01	2025-02-01	NULL	2490.00
<input type="checkbox"/>	2	2024-06-01	2025-06-01	Тропики	3666.00
<input type="checkbox"/>	3	2024-03-01	2025-01-01	Первый круг ада	4010.00
<input type="checkbox"/>	4	2024-04-01	2025-01-01	Второй круг ада	4020.00
<input type="checkbox"/>	5	2024-05-01	2025-01-01	Третий круг ада	4030.00
<input type="checkbox"/>	7	NULL	NULL	Дорогой	5000.00

Рисунок 5.48 – Ввод данных в представление

## Ошибка

SQL запрос: [Копировать](#)

```
INSERT INTO expensive_rates (number, name, cost) VALUES (7, 'Не Дорогой', 3500);
```

Ответ MySQL:

#1369 - проверка CHECK OPTION для VIEW 'heatbd.expensive\_rates' провалилась

Рисунок 5.49 – Безуспешный ввод данных в представление

	accural number	for all	for 1 m	rate number
<input type="checkbox"/>	0	123200.00	3080.00	0
<input type="checkbox"/>	1	871500.00	2490.00	1
<input type="checkbox"/>	2	205296.00	3666.00	2
<input type="checkbox"/>	3	160400.00	4010.00	3
<input type="checkbox"/>	4	160800.00	4020.00	4

Рисунок 5.50 – Представление для вывода данных

## 5.6. SQL-запросы на создание индексов

Были разработаны и выполнены следующие запросы:

- 1) SQL-запрос на создание уникального индекса, выборку и удаление индекса (рисунок 5.51):

```
CREATE UNIQUE INDEX ind_name ON RATES(name);
```

```
SELECT * FROM RATES WHERE name = 'Землянка';
```

```
DROP INDEX ind_name ON RATES
```

					number	start_date	end_date	name	cost
<input type="checkbox"/>					0	2024-01-01	2025-01-01	Землянка	3080.00

Рисунок 5.51 – Выборка уникального индекса

- 2) SQL-запрос на создание индекса на поле, по которому сортируются данные, выборку и удаление индекса (рисунок 5.52):

```
CREATE INDEX name_cost ON RATES(name, cost);
```

```
SELECT * FROM RATES WHERE NAME IS NOT NULL ORDER BY cost;
```

```
DROP INDEX name_cost ON RATES;
```

					number	start_date	end_date	name	cost	▲ 1
<input type="checkbox"/>					0	2024-01-01	2025-01-01	Землянка	3080.00	
<input type="checkbox"/>					2	2024-06-01	2025-06-01	Тропики	3666.00	
<input type="checkbox"/>					3	2024-03-01	2025-01-01	Первый круг ада	4010.00	
<input type="checkbox"/>					4	2024-04-01	2025-01-01	Второй круг ада	4020.00	
<input type="checkbox"/>					5	2024-05-01	2025-01-01	Третий круг ада	4030.00	
<input type="checkbox"/>					7	NULL	NULL	Дорогой	5000.00	

Рисунок 5.52 – Выборка индекса для поля, по которому сортируются данные

- 3) SQL-запрос на создание индекса на поле, по которому происходит группировка во время агрегации, выборку и удаление индекса (рисунок 5.53);

```
CREATE INDEX sdate ON RATES(start_date);
```

```
SELECT start_date, AVG(cost) FROM RATES GROUP BY start_date;
```

```
DROP INDEX sdate ON RATES;
```

← T →				start_date	AVG(cost)
<input type="checkbox"/>				NULL	5000.000000
<input type="checkbox"/>				2024-01-01	3080.000000
<input type="checkbox"/>				2024-02-01	2490.000000
<input type="checkbox"/>				2024-03-01	4010.000000
<input type="checkbox"/>				2024-04-01	4020.000000
<input type="checkbox"/>				2024-05-01	4030.000000
<input type="checkbox"/>				2024-06-01	3666.000000

Рисунок 5.53 – Выборка индекса, по которому происходит группировка во время агрегации

- 4) SQL-запрос на создание составного индекса с учетом селективности колонок, выборку и удаление индекса (рисунок 5.54):

```
CREATE INDEX id_name ON RATES(number, name);
```

```
SELECT * FROM RATES WHERE number > 3 AND name IS NOT NULL;
```

```
DROP INDEX id_name ON RATES;
```



	number	start_date	end_date	name	cost
<input type="checkbox"/>	4	2024-04-01	2025-01-01	Второй круг ада	4020.00
<input type="checkbox"/>	5	2024-05-01	2025-01-01	Третий круг ада	4030.00
<input type="checkbox"/>	7	NULL	NULL	Дорогой	5000.00

Рисунок 5.54 – Выборка индекса с учетом селективности колонок

Для индексации были выбраны поля `number` и `name`, т.к. `number` – это первичный ключ, и он является уникальным, а `name` – название тарифа, которое имеет практически нулевую вероятность повторения.

#### 5) SQL-запрос на создание составного индекса, выборку и удаление индекса (рисунок 5.55):

```
CREATE INDEX id_name_c ON RATES(number, name, cost);
```

```
SELECT * FROM RATES WHERE number > 3 AND name IS NOT NULL ORDER BY cost;
```

```
DROP INDEX id_name_c ON RATES;
```

	number	start_date	end_date	name	cost ▲ 1
<input type="checkbox"/>	4	2024-04-01	2025-01-01	Второй круг ада	4020.00
<input type="checkbox"/>	5	2024-05-01	2025-01-01	Третий круг ада	4030.00
<input type="checkbox"/>	7	NULL	NULL	Дорогой	5000.00

Рисунок 5.55 – Выборка составного индекса

На первое место в индексе было поставлено поле `number`, т.к. оно будет первым в запросе и является первичным ключом, на последнее место было поставлено поле `cost`, т.к. по нему проводится сортировка.

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения комплексной работы была разработана база данных для ежемесячного учета оплаты услуг за теплоснабжение в теплосети района. База данных содержит следующие данные: адрес (район, улица, номера дома и квартиры), план и площадь квартиры, месячные тарифы за потребленное тепло из расчета за 1 квадратный метр площади, общая плата за услуги, сведения об уплате с указанием дат и сумм, показатели оплаты услуг за отчетное полугодие по домам.

Также в ходе выполнения данной комплексной работы был проведён анализ предметной области, было проведено инфологическое, даталогическое и физическое проектирование базы данных. Было проведено тестирование базы данных на: добавление данных в таблицы базы данных, соблюдение ссылочной целостности, возможность выборки из одного источника данных, возможность выборки из нескольких источников данных. Разработанная база данных успешно прошла все тесты.

Были написаны SQL-запросы на создание пользовательских представлений и на создание индексов.

## СПИСОК ИСТОЧНИКОВ

1. Советов, Б. Я. Базы данных : учебник для прикладного бакалавриата / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2019. — 420 с. — (Бакалавр. Прикладной курс). — ISBN 978-5-534-07217-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/431947> (дата обращения: 20.06.2022).
2. Голицына, О. Л. Базы данных : учеб. пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. — 4-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2019. — 400 с. — (Высшее образование: бакалавриат). - ISBN 978-5-00091-516-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1019244> (дата обращения: 20.06.2022). — Режим доступа: по подписке.
3. А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для проектирования информационных систем : учеб. пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 368 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-8199-0718-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1001370> (дата обращения: 20.06.2022). — Режим доступа: по подписке.
4. Нестеров, С. А. Базы данных : учебник и практикум для академического бакалавриата / С. А. Нестеров. — Москва : Издательство Юрайт, 2019. — 230 с. — (Бакалавр. Академический курс). — ISBN 978-5-534-00874-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/433369> (дата обращения: 20.06.2022).
5. Гордеев, С. И. Организация баз данных в 2 ч. Часть 1 : учебник для вузов / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 310 с. — (Университеты России). — ISBN 978-5-534-04469-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/437731> (дата обращения: 20.06.2022).
6. Гордеев, С. И. Организация баз данных в 2 ч. Часть 2 : учебник для вузов / С. И. Гордеев, В. Н. Волошина. — 2-е изд., испр. и доп. — Москва : Издательство Юрайт, 2019. — 513 с. — (Университеты России). — ISBN 978-5-534-04470-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/438946> (дата обращения: 20.06.2022).
7. Нестеров, С. А. Интеллектуальный анализ данных с использованием SQL Server : учебник для вузов / С. А. Нестеров. — 2-е изд., стер. —

- Санкт-Петербург : Лань, 2024. — 160 с. — ISBN 978-5-507-49483-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/393005> (дата обращения: 18.06.2024). — Режим доступа: для авториз. пользователей.
8. Фиайли, К. SQL / К. Фиайли. — Москва : ДМК Пресс, 2008. — 451 с. — ISBN 5-94074-233-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/1242> (дата обращения: 18.06.2024). — Режим доступа: для авториз. пользователей.
9. Бен-Ган, И. Оконные функции в T-SQL / И. Бен-Ган ; перевод с английского А. Ю. Гинько. — Москва : ДМК Пресс, 2022. — 344 с. — ISBN 978-5-93700-139-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/314852> (дата обращения: 18.06.2024). — Режим доступа: для авториз. пользователей.
10. Зудилова, Т. В. SQL и PL/SQL для разработчиков СУБД ORACLE : учебно-методическое пособие / Т. В. Зудилова, С. Е. Иванов, С. Э. Хоружников. — Санкт-Петербург : НИУ ИТМО, 2012. — 74 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/40721> (дата обращения: 18.06.2024). — Режим доступа: для авториз. пользователей.
11. Уорд, Б. Инновации SQL Server 2019 / Б. Уорд ; перевод с английского Н. Б. Желновой. — Москва : ДМК Пресс, 2020. — 408 с. — ISBN 978-5-97060-595-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/190722> (дата обращения: 18.06.2024). — Режим доступа: для авториз. пользователей.

## ПРИЛОЖЕНИЕ А

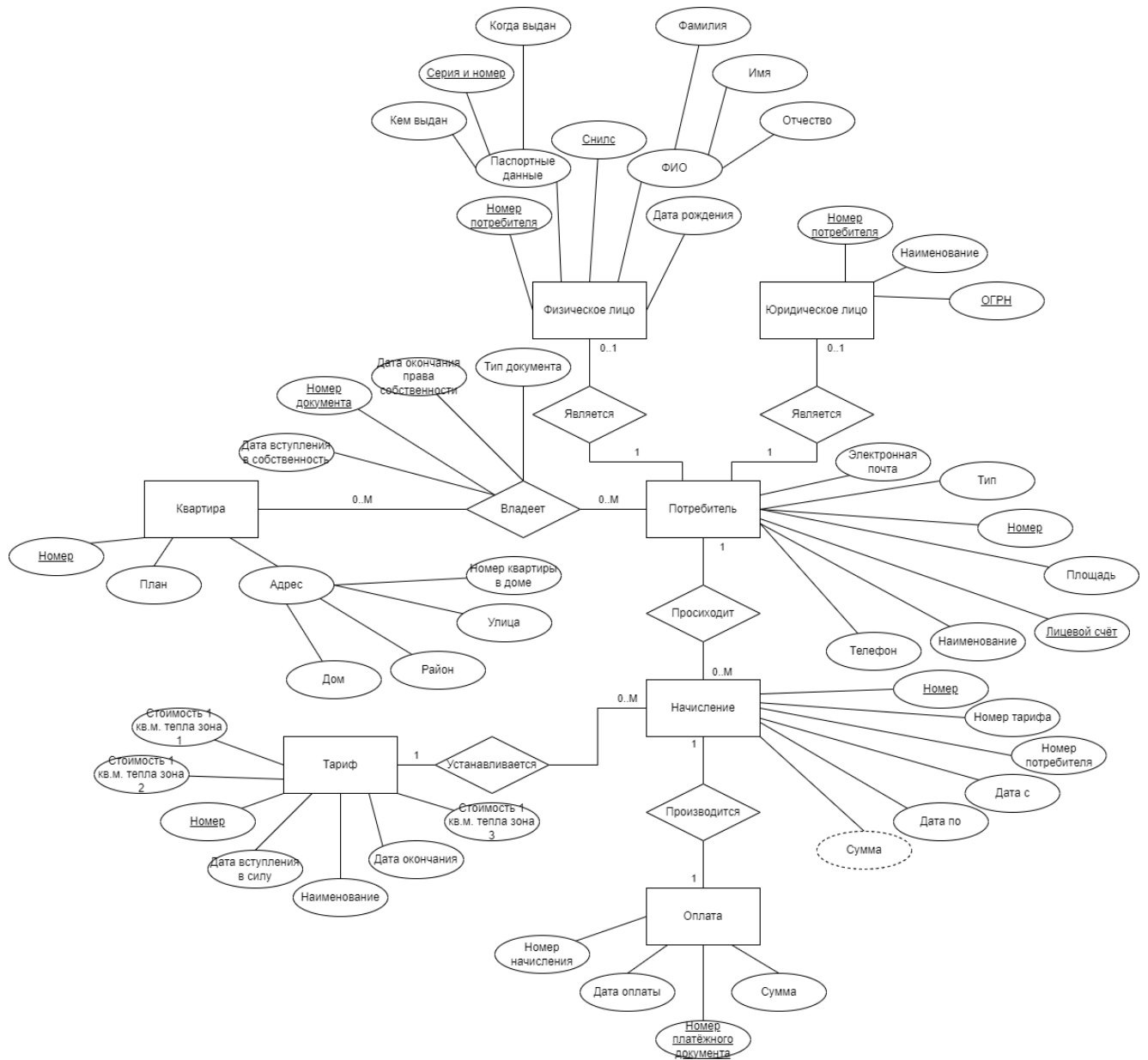


Рисунок А.1 – ER-диаграмма уровня свойств сущностей  
и связей предметной области