**Содержание**

Оглавление

[2.Анализ предметной области 2](#_Toc96730438)

[3.Постановка задачи 4](#_Toc96730439)

[4.Техническое задание 5](#_Toc96730440)

[5. Проектирование реляционной базы данных 6](#_Toc96730441)

[6.Логическое проектирование БД и нормализация полученных отношений 11](#_Toc96730442)

[7.Физическое проектирование БД 17](#_Toc96730443)

[8.Реализация проекта БД 17](#_Toc96730444)

[9.Заключение 22](#_Toc96730445)

[10.Список использованных источников 22](#_Toc96730446)

**3)Объектом исследования** является – база данных управляющей компании.

**4)Предмет исследования** – автоматизация базы данных управляющей компании.

**5)Цель работы -** разработка базы данных для автоматизации работы управляющей компании.

Для реализации целей надо выполнить следующие задачи:

1.Создать возможность получения данных по всем осуществляемым запросам

2. Обеспечить целостность данных в базе данных

3. выполнить концептуальное проектирование БД

4. выполнить инфологическое проектирование БД

5. выполнить логическое проектирование БД

# 2.Анализ предметной области

Управляющая компания функционирует в рамках хозяйственной и финансовой деятельности, подразумевающей осуществление различных хозяйственных работ и услуг, а также осуществление финансовых сборов и платежей. Большей части управляющих компаний (УК), присущи следующие наборы элементов:

1)Сведения о клиентах. Включает в себя: Ф.И.О клиента и его адрес проживания.

2)Сведения о работниках УК: Ф.И.О. и должность сотрудника

3)Различные виды услуг: Код услуги, ее название и разброс цен

4)Данные об оплате услуг УК. Данный элемент включает в себя такие поля, как: Дата отправления квитанции об оплате, Сумма квитанции, Даты оплаты, Наличие штрафных санкций.

Управляющая компания получает информацию о том, что ее клиенту необходимо предоставить определенную услугу, узнает его фамилию и имя, адрес проживания, по которому будет предоставлена услуга. После чего, УК, подбирает необходимую услугу по требованию клиента и объявляет возможную цену.

Затем, УК рассматривает наличие подходящего, для выполнения этой услуги, сотрудника.

После сбора информации, управляющая компания объединяет все вышеперечисленные критерии и создает новые записи, в которых указывается конкретная дата выполнения услуги и ее окончательная цена.

Также, в функции поля деятельности УК входит надзор за своевременной отправкой квитанции и слежение за ее оплатой клиентами. Вследствие чего компании необходимо создавать график оплаты квитанции, включающий в себя информацию об идентификационном номере оплаты квитанции, дате отправления квитанции, необходимой сумме оплаты, данные о фактической и плановой дате оплаты, сумму штрафных санкций в отношении клиента

Единая информационная система управляющей компании должна работать на платформе СУБД, а если быть точнее, то на MySQL или подобных платформах.

Автоматизированный метод ведения данных процессов позволяет наиболее точно, быстро и безошибочно собирать и производить разнообразные процедуры с данными. Таким образом, это позволит работнику быстрее и лучше выполнять работу, никак не отвлекаясь на перепроверку данных.

Идентификатор записи каждого обслуженного клиента, пользующегося услугами управляющей компании, является неповторимым, что позволяет легко и быстро найти данные о клиенте, содержащиеся в базе данных (Ф.И.О, адрес проживания, наличие долга по оплате и его сумме и так далее).

Главные проблемы, которые решает данная база данных:

-информативное предоставление сотрудников отделения материально-технологического обеспечения, чья работа напрямую сопряжена со базой данных

-помогает производить надзор за соблюдением договора о сроках оплаты кредита

Создаваемая нами сущность, должна соответствовать всем необходимым, для нормального функционирования нормам и стандартам предметной области (УК). Следовательно, должны выполняться требования национального стандарта РФ: ГОСТ Р 56038-2014 Услуги жилищно-коммунального хозяйства и управления многоквартирными домами. Услуги управления многоквартирными домами. Общие требования.

Так же, данная база данных должна отвечать следующему стандарту: ГОСТ 34.321-96 Информационные технологии (ИТ). Система стандартов по базам данных. Эталонная модель управления данными.

# 3.Постановка задачи

Предметная область, подлежащая дальнейшему нашему изучению – «управляющая компания». В сферу деятельности этой предметной области попадают учреждения и фирмы, занимающиеся оказанием услуг в области обслуживания домов. Цель функционирования предприятий этой предметной области – оказание услуг, таких как: уборка обслуживаемых территорий, надлежащее содержание и своевременный ремонт помещений нанимателей, надзор за своевременной оплатой услуг, получение прибыли, извлекаемой из оплаты этих услуг клиентами управляющей компании. Для оказания услуг необходимо наличие квалифицированных специалистов и помещений, удовлетворяющих нормам санитарных, противопожарных и других требований в соответствии с действующим законодательством.

**Целью курсовой работы является** создание информационно-программной системы для решения следующих задач:

1)автоматизации операций осуществления услуг клиентам.

2)централизованного хранения информации о сотрудниках с целью выбора подходящего сотрудника,

3)обеспечения хранения информации о клиентах, пользующимися услугами управляющей компании:

1.сбора информации об оплате квитанций,

2.расчет наличия у клиента долга,

3.возможность применения штрафных санкций, по отношению к клиентам.

1)провести обследование предметной области

2)предусмотреть возможность внесения изменений в данные и добавления новых записей

3)Наличие услуг до определенной суммы

4)Сколько стоит самая дорогая услуга?

5)Какая фамилия у сотрудника?

6)Поиск услуг, оказанным клиентам по определенным критериям

7)Осуществление поиска наличия пени

8)Поиск даты отправки и оплаты квитанции

9)Поиск самого «экономного» клиента

10)Поиск клиента с самым большим долгом

11)Поиск даты осуществления услуг с заданными требованиями

12)Поиск заказанных клиентами услуг

13)Поиск сумм по квитанциям, среди клиентов

# 4.Техническое задание

1. Введение

− Наименование: «База данных для управляющей компании».

− Созданная база данных будет использоваться в управляющих компаниях для хранения и обработки и выдачи информации. Работники управляющих компаний, используя запросы в базе данных, смогут находить и обрабатывать данные о своих клиентах.

2. Основание для разработки.

Учебный план по направлению 02.03.03 – «Математическое обеспечение и администрирование информационных систем» (бакалавриат).

3. Назначение разработки.

Программа предназначена для выполнения операций, помогающих управляющим компаниям, и содержит в себе следующие функции: хранение, поиск и сортировка данных.

4. Требования, предъявляемые к программе.

4.1) В создаваемой базе данных должны быть реализованы следующие данные:

-ФИО клиента

-Адрес проживания клиента

-ФИО сотрудника

-Должность сотрудника

-Название услуги

-Цены услуги

-Дата оказания услуги

-Дата отправление квитанции

-Сумма квитанции

-Фактическая и плановая оплаты квитанции

-Штрафные санкции

4.2) Требования к техническим средствам, используемым при работе программы.

Работа программы должна осуществляться на ПК, со следующими характеристиками:

Оперативная память:4Gb

ОС: Window 7,8.1,10

4.3) Требования к языкам программы и среде разработки программы.

База данных для управляющих компаний, должна осуществлять свою работу на следующих платформах СУБД: SQLyog, PHPMyAdmin и аналогичные.

5. Требования к информационным структурам на входе и выходе программы.

Входными данными информационной системы являются: числовые данные (INT, DOUBLE), данные о датах (тип Date), текстовые данные (Varchar). Они вводятся вручную, с клавиатуры.

Выходными данными информационной системы являются – числовые данные, даты, текстовые данные. Они выводятся на экран пользователя.

1)Обзор способов организации данных и обоснование выбора структуры данных для эффективного выполнения операций.

2)Разработка программы.

3)Разработка программной документации.

4)Оформление пояснительной записки.

5)Защита курсовой работы.

# 5. Проектирование реляционной базы данных

Концептуальное проектирование базы данных и инфологическое проектирование БД

**1)**Главной, основной сущностью, от которой мы будем отталкиваться, является «Квитанция-услуги-работники», потому что в управляющих компаниях, именно она определяет те услуги, которые оплачивает клиент.

В данной выше сущности, используются основные ключи для связи, такие как:

-IdОплатаКвитанций

-IdУслуги

-Квартира\_idКвартира

-IdРаботник\_idРаботник

Данная таблица составляет основу между Клиентам и услугами, предоставляемые от УК для технического обслуживания жильца и включения коммунальных услуг. Также данная содержит в себе:

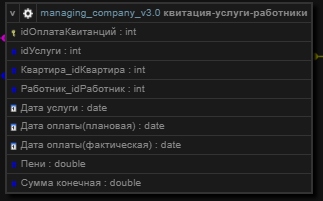
-Дата услуги

-Дата оплаты(плановая)

-Дата оплаты(фактическая)

-Пени

-Сумма конечная



Сущность "Квитанция-услуги-работники"

**2)**Следующей сущностью нашей базы данных является «клиенты», поскольку именно они заказывают и оплачивают услуги, предоставляющей управляющей компанией.

-IdКлиент- Ключевое поле

-Фамилия

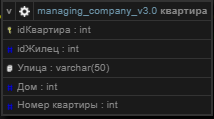
-Имя

Изображение выглядит как текст

Автоматически созданное описание

Сущность "Клиент"

**3)** Также, управляющим компаниям необходимо знать точный адрес оказания услуги клиенту, вследствие чего, мы добавили сущность «Квартира» имеет следующие атрибуты:



Сущность "Квартира"

**4)**Также, в базе данных представлена сущность «Услуги», обобщающая в себе услуги клиента и выводящая конечную обслуживающею функцию. Таблица хранит в себе следующее ключевое поле:

-IdКод услуги

Поскольку, клиент должен знать название услуг и их начальную и конечную стоимость, то сущность имеет необходимые для этого поля:

-Название услуги

-Цена

Изображение выглядит как текст

Автоматически созданное описание

Сущность "Услуги"

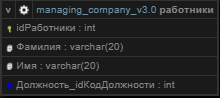
**5)**Сущность «Работники» является дополнительной информацией для возможности оказания услуги, поскольку она обеспечивает лучшее оказания услуги, путем подбора наиболее квалифицированного персонала под нее. «Работники» имеет следующие атрибуты:

-IdРаботник, являющееся ключевым полем

-Фамилия

-Имя

-Должность\_idКодДолжности(связующий атрибут)



Сущность "Работники"

**6)** Чтобы иметь представление о должности сотрудника управляющей компании, выделена отдельная сущность «Должность работника»:

-idДолжность работника

-Название должности

-Стаж работы

Изображение выглядит как текст

Автоматически созданное описание

Сущность "Должность работника"

**7)** Так как управляющая компания должна предоставлять жильцу квитанцию за оплату коммунальных услуг, в квартире должны быть установлены исправно рабочие приборы, фиксирующие использование воды, газа, света и предоставлять информацию об этом:

-idСчетчика – Ключевое поле

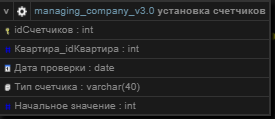
-Квартира\_idКвартира – вторичный ключ, показывающий к какой квартире относится счетчик

И данные, характеризующие тип и изначальную дату проверки прибора:

-Дата проверки

-Тип счетчика

-Начальное значение



Сущность "Установка счетчиков"

**8)** Последняя сущность в бд управляющей компании, которая выполняет опцию снятия показания данных с установленных счетчиков:

-Дата период (период, за который были сняты показатели с приборов)

Связующие ключи с другими таблицами:

-Установка счетчиков\_idСчетчиков

-Все услуги\_idУслуги

И данные, показывающие значения на счетчиках:

-Начальное значение

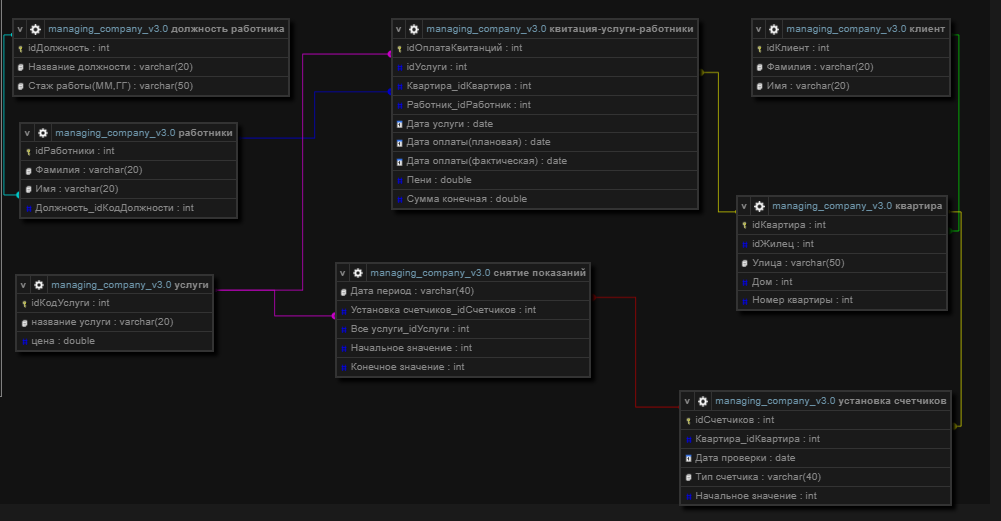
-Конечное значение

Изображение выглядит как текст

Автоматически созданное описание

Сущность "Снятие показаний"

Итоговая ER-модель выглядит следующим образом:



# 6.Логическое проектирование БД и нормализация полученных отношений

Первая нормальная форма (1 НФ) – Отношение находится в первой нормальной форме, если все его атрибуты имеют простые (атомарные) значения. Понятие атомарности определяется исходя из использования информационной системы.

Вторая нормальная форма (2 НФ) – Отношение находится во второй нормальной форме, если оно находится в 1 НФ и все его поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом. Если ключом является только один атрибут, то такое отношение уже находится во 2 НФ.

Третья нормальная форма (3 НФ) – Отношение находится в третьей нормальной форме, если оно находится во 2 НФ и ни одно из его не ключевых полей не зависит функционально от любого не ключевого поля.

**1)**Таблица «Квитанция-услуги-работники» имеет столбцы, необходимые для создания внешних связей, которые будут иметь тип INT

Строки «Дата услуги», «Дата оплаты(плановая)», «Дата оплаты(фактическая)» содержат в себе дату, следовательно, тип данных, заданный в поле, будет «Date».

Поле «Пенни» хранит в себе информацию о штрафных санкциях в адрес клиента по причине неуплаты или невыплаты в срок квитанции за оказание предоставленных услуг, соответственно имеет тип данных Double.

Поле «Сумма конечная» содержит в себе информацию о сумме оплаты услуги, поэтому для более надежного хранения, используем тип данных «Double».

Приведение к 3 НФ:

Квитанция-услуги-работники (IdОплатаКвитанций, IdУслуги, Квартира\_idКвартира, IdРаботник\_idРаботник, Дата услуги, Дата оплаты(плановая), Дата оплаты(фактическая), Пени, Сумма конечная)

1 НФ- все значения атомарные

2 НФ- ключ состоит только из одного атрибута idОплатаКвитанций

3 НФ- отсутствуют транзитивные зависимости

Изображение выглядит как текст, монитор, снимок экрана, экран

Автоматически созданное описание

**2)**Таблица «Клиент» содержит в себе информацию о нанимателях управляющей компании.

Поле idКлиент имеет тип данных INT, поскольку является ключевым полем, необходимым лишь для внешней связи.

Поля «Фамилия», «Имя», обладают информацией, заданной в письменной форме, то есть имеет лишь различные символы (буквы, цифры и т.д.). Поскольку вышеперечисленные поля могут иметь достаточно длинные значения, задаем полям размер 20.

Приведение к 3 НФ:

Клиент (idКлиент, Фамилия, Имя)

1 НФ – все значения атомарные

2 НФ – ключ состоит только из одного атрибута idКлиент

3 НФ – отсутствуют транзитивные зависимости

Изображение выглядит как текст

Автоматически созданное описание

**3)**Таблица «Квартира» используется в качестве представления жилья для клиента, и соответствующих сведений о квартире.

Атрибут «idКвартира» мы используем в качестве идентификационного номера. Он будет содержать в себе числовые данные типа INT.

Вторичный ключ «idЖилец» служит для связи с таблицей клиента, поэтому имеет числовой тип данных INT.

Поле «Улица» содержит в себе информацию об улице, на которой расположено жилье. Название улицы имеет тип данных VARCHAR, хранящий в себе информацию текстового формата. Так как название улицы может содержать в себе большой объем текста ограничим его длиной 50.

Также поля «Дом» и «Номер квартиры» содержат в себе числовую информацию о местоположении квартиры, поэтому им зададим формат INT.

Приведение к 3 НФ:

Услуги (idКвартира, idЖилец, Улица, Дом, Номер квартиры)

1 НФ- все значения атомарные

2 НФ- ключ состоит только из одного атрибута idКвартира

3 НФ- отсутствуют транзитивные зависимости

Изображение выглядит как текст

Автоматически созданное описание

**4)**Следующая таблица, представленная нам, называется «Услуги».

Атрибут «idКодУслуги» мы используем в качестве идентификационного номера. Он будет содержать в себе числовые данные типа INT.

Атрибут «Цена» использует в своем составе лишь денежные единицы, представленные типом DOUBLE.

Поле «Название услуги» содержит в себе текстовый тип данных, ограниченный достаточной длиной 20.

Приведение к 3 НФ:

Услуги (idКодУслуги, название услуги, цена)

1 НФ – все значения атомарные

2 НФ – ключ состоит только из одного атрибута idКодУслуги

3 НФ – отсутствуют транзитивные зависимости

Изображение выглядит как стол

Автоматически созданное описание

**5)**Таблица «Работники» содержит информацию о сотрудниках управляющей компании.

Атрибут «idРаботники» представлена в числовом формате типа INT и необходима для внешней связи таблицы.

Поля «Фамилия», «Имя» используют тип данных VARCHAR, поскольку содержат в себе данные текстового формата и включают в себя различные символы. Они ограничены достаточным размером 20. А также связующий атрибут Должность\_idКодДолжности

Приведение к 3 НФ:

Работники (idРаботники, Фамилия, Имя, Должность\_idКодДолжности)

1 НФ – все значения атомарные

2 НФ – ключ состоит только из одного атрибута idРаботники

3 НФ – отсутствуют транзитивные зависимости

Изображение выглядит как текст

Автоматически созданное описание

**6)** Таблица «Должность работника» предоставляет нам информацию о специализации работника и его стаже.

Атрибут «idДолжность» представлена в числовом формате типа INT и необходима для внешней связи таблицы.

Поля «Название должности», «Стаж работы» используют тип данных VARCHAR, поскольку содержат в себе данные текстового формата и включают в себя различные символы. Они ограничены достаточным размером 20 и 50.

Приведение к 3 НФ:

Должность работника (idДолжность, Название должности, Стаж работы)

1 НФ – все значения атомарные

2 НФ – ключ состоит только из одного атрибута idДолжность

3 НФ – отсутствуют транзитивные зависимости

Изображение выглядит как текст

Автоматически созданное описание

**7)**Следующая таблица, представленная для рассмотрения, «Установка счетчика». В ней находятся данные о показателях установленных приборов, а также в какой квартире установлены счетчики.

Данная сущность имеет в своем составе ключевое поле «idСчетчиков». Атрибут содержит в себе числовой тип данных INT.

Атрибуты «Квартира\_idКвартира» содержит в себе в себе связующий ключ с таблицей Квартиры. Вследствие чего он имеет тип данных INT, хранящий числовую информацию.

Также, в таблице присутствуют поле «Дата проверки», которая содержит в себе информацию о последней дате проверки счетчика. Для выполнения этого условия они имеют тип данных DATE.

Поле «Тип счетчика» показывает нам какой тип прибора установлен в квартире. Так как он содержит описание о приборе, то будет иметь текстовый формат VARCHAR, ограниченный значением 40

И атрибут «Начальное значение», указывающий базовый показатель счетчика при его установке. Хранит в себе числовую информацию, поэтому зададим ему тип данных INT

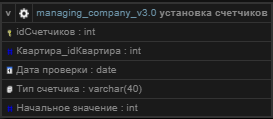
Приведение к 3 НФ:

Установка счетчика (idСчетчиков, Квартира\_idКвартира, Дата проверки, Тип счетчика, Начальное значение)

1 НФ – все значения атомарные

2 НФ – ключ состоит только из одного атрибута idСчетчиков

3 НФ – отсутствуют транзитивные зависимости



**8)**Таблица «Снятие показателей» содержит в себе информацию о перечне свойств счетчика, его начальное и последнее установленное значение.

Поле «Дата период» имеет тип данных VARCHAR, поскольку описывает время, когда снимались показания.

Поля «Установка счетчиков\_idСчетчиков», «Все услуги\_idУслуги», являются вторичными ключами для связи с другими таблицами, поэтому имеют тип данных INT.

Поля «Начальное значение», «Конечное значение», показывают нам значения счетчика в разные моменты. Они хранят числовую информацию, поэтому имеют тип данных INT.

Приведение к 3 НФ:

Снятие показателей (idКлиент, Фамилия, Имя)

1 НФ – все значения атомарные

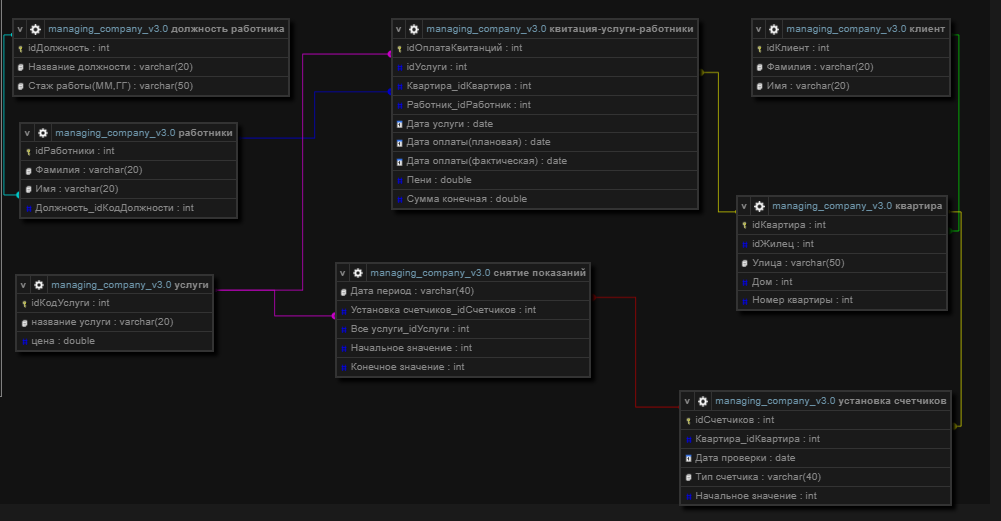
2 НФ – предоставлена главная информация, содержащая в себе период снятия показаний Дата(период)

3 НФ – отсутствуют транзитивные зависимости

Изображение выглядит как текст

Автоматически созданное описание

Как мы можем заметить, что все таблицы находятся в третьей нормальной форме, из чего следует, что созданная база данных, полностью соответствует третьей нормальной форме, показывая, что соблюдены все условия.



# 7.Физическое проектирование БД

Чтобы создаваемая нами база данных управляющей компании работала как можно эффективнее, требуется использовать минимально-допустимое количество памяти в ходе работы.

Чтобы добиться этого результата, необходимо четко и грамотно подбирать тип данных, длину строки. При проектировании базы данных нужно быть осторожным, ведь некоторые типы данных занимают память, даже если в соответствующие им строки ничего не записано.

Например, тип данных CHAR, именно он занимает память понапрасну. Вследствие чего мы делаем вывод, что гораздо целесообразнее, в полях, где используются текстовые типы данных, имеет смысл быть тип данных VARCHAR. Этот тип позволит нам экономить память, ведь он заполняет ее лишь по факту заполнения строки.

Также, стоит обдумать, какую длину необходимо задать строке. Ведь именно длина строки является ключевым способом экономии памяти. Например: зачем задавать длину строки больше, чем будет введено в нее информации.

Наравне с предыдущими этапами, можно сказать, что нам необходимо использовать минимальное количество таблиц, поскольку запросы с большим количеством таблиц обрабатывается дольше, нежели чем с малым количеством. При этом нельзя нарушать «логику» базы данных.

# 8.Реализация проекта БД

Логическое проектирование БД

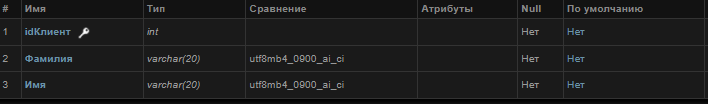
Чтобы приступить к созданию нашего проекта, необходимо запустить «PHPMyAdmin». Далее, мы создаем новую базу данных и начинаем процесс создания таблиц.

В поле «Имя» вводим название таблицы и задаем количество столбцов.



При нажатии на кнопку «Создать таблицу», перед нами откроется окно с полями, которые необходимо заполнить в соответствии с нашими требованиями.

После заполнения мы получаем следующий результат:



Аналогично проделанному выше, создадим еще 7 подобных таблицы и заполним их данными управляющей компании.

Конечный результат выглядит так:

Изображение выглядит как текст, электроника, калькулятор, картотека

Автоматически созданное описание

Создадим несколько запросов, чтобы убедиться в работоспособности нашей базы данных:

**1)**Клиенту необходима услуга, начальная стоимость которой, не превышает 1000 рублей

**SQL-запрос:**

Изображение выглядит как текст, устройство, датчик, счетчик

Автоматически созданное описание

**Результат:**

Изображение выглядит как текст

Автоматически созданное описание

**2)**Клиент хочет узнать название стоимость самой дорогой услуги

**SQL-запрос:**

Изображение выглядит как текст, темный, оранжевый

Автоматически созданное описание

**Результат:**

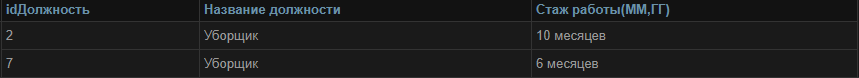


**3)**Клиенту необходимо найти фамилию уборщиков

**SQL-запрос:**

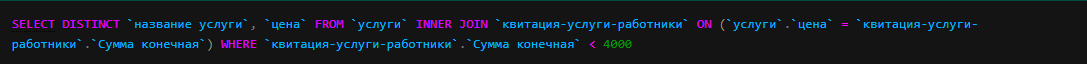


**Результат:**



**4)** Найдем оказанные клиентам услуги, по максимальному прайсу, при этом, их финальная стоимость была меньше 4000 рублей.

**SQL-запрос:**

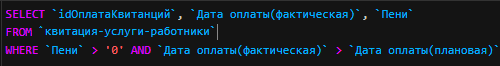


**Результат:**



**5)**Управляющая компания хочет найти записи о датах оплаты квитанции и начисления пени

**SQL-запрос:**



**Результат:**

Изображение выглядит как текст

Автоматически созданное описание

**6)** Посчитаем общий итог трат жильцов за коммунальные услуги



**Результат:**



**7)**Необходимо найти запись клиента, имеющего наименьшую сумму счета в квитанции

**SQL-запрос:**



**Результат:**



**8)**Управляющая компания хочет найти запись своего наибольшего должника

**SQL-запрос:**



**Результат:**



**9)**Необходимо найти запись даты услуг, осуществленных не позднее чем 2020-12-15

**SQL-запрос:**



**Результат:** Изображение выглядит как текст, электроника, снимок экрана, компьютер

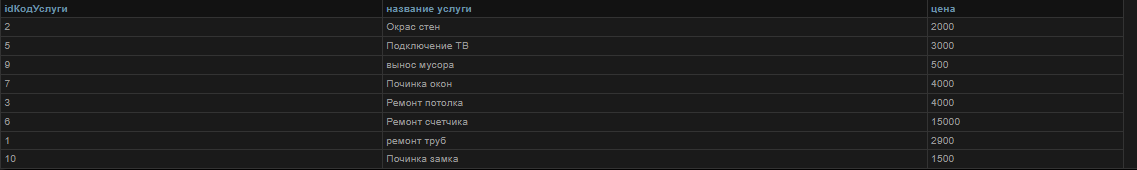
Автоматически созданное описание

**10)**Осуществим поиск записей услуг, которые были проданы клиентам по максимально-доустимой цене

**SQL-запрос:**



**Результат:**

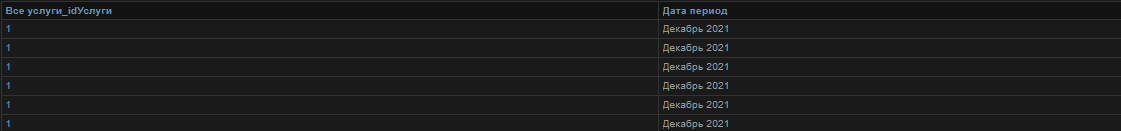


**11)**Найдем все записи по идентификационному номеру услуги, которую клиенты заказывали

**SQL-запрос:**



**Результат:**

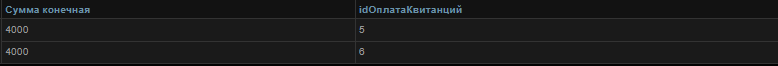


**12)**УК хочет найти сумму к оплате квитанций клиентов от 4000 до 8000 рублей

**SQL-запрос:**



**Результат:**



# 9.Заключение

Подводя итоги, могу с уверенностью сказать, что было проведено исследование заданной мне предметной области, выполнено концептуальное, инфологическое и логическое проектирования базы данных.

Созданная база данных, исправно работает и выполняет все возложенные на нее задачи. Например: запросы, возможность удаления, редактирования и добавления записей в базе данных.

Грамотно создаваемые запросы, позволяют нам легко найти: цену услуги, дат оказания услуги, ФИО клиентов, их адрес жительства, информацию о работниках и их работе по заказам клиента и так далее.

Созданная нами база данных, позволяет автоматизировать работу с данными для управляющей компании, что повысит эффективность работы последней в несколько раз.

В ходе проделанной работы были созданы:

-ER-модель

-12 запросов

-8 таблиц

# 10.Список использованных источников

1. Документация по MySQL – http://www.mysql.ru/docs/

2.  Сергей Моисеенко. SQL: Задачи и решения. Интерактивный учебник. - http://www.sql-tutorial.ru/ru/content.html

3.  Уроки SQL – https://site-do.ru/db/db.php#2