

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Кафедра Компьютерных технологий

«Допущен к защите»
Заведующий кафедрой
Куралбаев З.К. д.ф.-м.н. профессор


« 21 » мая 2014 г.

ДИПЛОМНЫЙ ПРОЕКТ

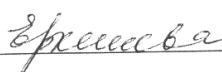
На тему: «Автоматизация биллинга электропотребления»

Специальность Вычислительная техника и программное обеспечение
Выполнила Шивазова М.К. группа ВТу-11-1

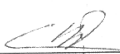
Научный руководитель Шайхин Б.М. к.ф.-м.н., доцент

 « 14 » мая 2014 г.

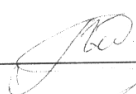
Консультанты:
по экономической части: Еркешева З.Д. ст. преподаватель

 « 22 » апреля 2014 г.


по безопасности жизнедеятельности: Бегимбетова А.С. ст. преподаватель

 « 4 » апреля 2014 г.

по применению вычислительной техники: Шайхин Б.М. к.ф.-м.н., доцент

 « 14 » мая 2014 г.

Нормоконтролер: Тусупов Д.М. ассистент

 « 12 » мая 2014 г.

Рецензент: Мукимбеков М. Ж. д.т.н., доцент

« _____ » _____ 20 _____ г.

Алматы 2014 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Информационных технологий
Специальность Вычислительная техника и программное обеспечение
Кафедра Компьютерных технологий

ЗАДАНИЕ

на выполнение дипломного проекта

Студентка Шивазова Майя Кемаровна

Тема проекта «Автоматизация биллинга электропотребления» утверждена приказом ректора № 115 от «24» сентября 2013 г.

Срок сдачи законченной работы «2» июня 2014 г.

Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта

- автоматизированная биллинговая система;
- техническое задание по проектированию баз данных;
- техническое задание по функциональному дизайну.

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:

- рассмотреть основные теоретические и практические вопросы, связанные с деятельностью компаний энергосбыта;
- рассмотреть принципы работы биллинговой системы;
- рассмотреть процесс построения концептуальных моделей и создания базы данных;
- провести анализ предметной области, спроектировать базу данных среде Microsoft SQL Server Management Studio 2008;
- рассчитать объем памяти базы данных;
- разработать хранимые процедуры, триггеры и функций на языке Transact-SQL;
- разработать формы для интерфейса в среде Microsoft Visual Studio 2010.


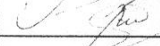
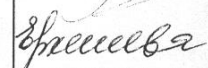

Перечень графического материала (с точным указанием обязательных чертежей)

<i>Теоретическая часть</i>
<i>Специальная часть</i>
<i>Технико-экономические обоснование проекта</i>
<i>Безопасность жизнедеятельности</i>

Рекомендуемая основная литература

- 1 Питер Роб, Карлос Коронел Системы баз данных: проектирование, реализ и управление, 5-е издание, – BHV Санкт-Петербург, 2004. – 1040 с.
- 2 Дрозд А.С. Организация биллинговой системы. – М.: Изд. Вильямс, 2000. – 400 с.
- 3 Ибраева Л.К., Сатимова Е.Г. Проектирование базы данных: Методические указания к выполнению лабораторных работ (для студентов специальностей). – Алматы: АИЭС, 2004. – 45 с.
- 4 Майкл Дж. Хернандес, Джон Л. Вьескас SQL-запросы для простых смертных. – М.: Изд. Лори 2000. – 520 с.
- 5 Плю Р., Стефенс Р., Райан К. Освой самостоятельно SQL за 24 часа. – М.: Изд. Вильямс, 2000. – 416 с.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Общая часть	Шайхин Б.М.	27.03.2014	
Специальная часть	Шайхин Б.М.	15.04.2014	
Технико – экономическое обоснование проекта	Еркешева З.Д.	22.04.2014	
Безопасность жизнедеятельности, охрана труда и промышленная экология	Бегимбетова А.С.	4.04.2014	


Г Р А Ф И К
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1	Оценки объема потребления электроэнергии	24.03.2014	
2	Погрешности методов оценок потребления	25.03.2014	
3	Реляционная модель и ANSI SQL	27.03.2014	
4	Анатомия реляционных баз данных	02.04.2014	
5	БД в биллинге	04.04.2014	
6	Клиент-серверная СУБД	09.04.2014	
7	Основные положения	11.04.2014	
8	Анализ предметной области	15.04.2014	
9	Бизнес правила	17.04.2014	
10	Составление реляционных отношений	18.04.2014	
11	Расчет размера базы данных	21.04.2014	
12	Меры по обеспечению безопасности	23.04.2014	
13	Логическое проектирование	25.04.2014	
14	Создание функций	28.04.2014	
15	Создание хранимых процедур	29.04.2014	
16	Создание триггеров	30.04.2014	
17	Работа с приложением	02.05.2014	

Дата выдачи задания «18» сентября 2013 г.

Заведующий кафедрой  Куралбаев З.К.

Руководитель  Шайхин Б.М.

задание принял к исполнению студент  Шивазова М.К.

АННОТАЦИЯ

Данный дипломный проект посвящён разработке программного обеспечения «Автоматизация биллинга электропотребления». В ходе реализации программного обеспечения были рассмотрены основные теоретические и практические вопросы, связанные с деятельностью компаний энергосбыта. Дипломный проект так же содержит описание принципов работы

биллинговой системы и процесса построения концептуальных моделей и создания базы данных.

База данных дипломного проекта реализована в среде SQL Server 2008, а интерфейс выполнен в среде Microsoft Visual Studio 2010 на языке C#. Автоматизация биллинга электропотребления была адаптирована под требования разрабатываемого процесса. Рассмотрены вопросы безопасности жизнедеятельности, условий труда, выполнен расчёт освещённости и кондиционирования рабочего помещения.

АНДАТПА

Бұл диплом жобасы «Электртұтыну биллингiсiн автоматтандыру» бағдарламалық қамту әзiрлемесiне арналды. Бағдарламалық қамтуларды жүзеге асыру барысында энергия өткiзушi компаниялардың қызметiмен байланысты негiзгi теориялық және практикалық мәселелер қарастырылды. Сондай-ақ диплом жобасында биллингтiк жүйенiң жұмыс iстеу процесi мен концептуалды модельдердi қалыптастыру және деректер базасын құру процестерiне сипаттама берiлдi.

Диплом жобасының деректер базасы SQL Server 2008 ортасында iске асырылды, ал интерфейс C# тiлiнде Microsoft Visual Studio 2010 ортасында орындалды. Электр тұтыну биллингiсiн автоматтандыру әзiрленген процестердiң талаптары негiзiнде бейiмделдi. Өмiр тiршiлiк қауiпсiздiгi, еңбек шарттары мәселелерi қарастырылды, жұмыс бөлмелерiнiң желдетiлуi мен жарықтандырылуының есебi жасалды.

ABSTRACT

The project for the degree of Bachelor is devoted to the software Engineering «Automation of power consumption billing». The basic theoretical and practical problems are discussed during the software realization, related to the activity of power marketing companies. The Bachelors' degree project contains the work principles of the billing system's description, the conceptual models of construction process and database creation.

The database of the Bachelors' degree project is realized in the environment of SQL Server 2008, and the interface is fulfilled in the environment of Microsoft Visual Studio 2010 in C# language. Automation of power consumption billing has been adapted to the developed process requirements. The problems of life activity and working conditions are considered, calculation of light exposure and air-conditioning of working premise is performed.

СОДЕРЖАНИЕ

ВЕДЕНИЕ	9
1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	11
1.1 Биллинг	11
1.2 Оценки объема потребления электроэнергии	12
1.2.1 Погрешности методов оценок потребления.....	14
1.2.2 Метод «по книжкам»	14
1.2.3 Метод «по счетам».....	15
1.2.4 Метод «по договорному»	16
1.2.5 Кредитование.....	17
1.2.6 Комбинированные методы.....	17
1.3 БД в биллинге.....	19
1.4 База данных.....	19
1.5 Общие сведения о языке SQL.....	21
1.5.1 Краткая история о языке SQL.....	21
1.5.2 Реляционная модель и ANSI SQL	22
1.5.3 Анатомия реляционных баз данных	22
1.6 Клиент-серверная СУБД.....	27
1.7 СУБД SQL SERVER 2008.....	28
2 СПЕЦИАЛЬНАЯ ЧАСТЬ.....	30
2.1 Основные положения	30
2.2 Анализ предметной области.....	31
2.3 Основная задача	31
2.4 Бизнес правила.....	33
2.5 Составление реляционных отношений	42
2.6 Расчет размера базы данных	46
2.7 Меры по обеспечению безопасности.....	51
2.8 Логическое проектирование	53
2.8.1 Задание места хранения БД	54
2.8.2 Создание таблиц.....	54
2.8.3 Создание функций	58
2.8.4 Создание триггеров.....	59
2.8.5 Создание процедур	64
2.9 Работа с приложением.....	67
3 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА.....	82
3.1 Описание работы и обоснование необходимости	82
3.2 Трудовые ресурсы, используемые в работе	82
3.3 Оборудование и программное обеспечение, используемое в работе	82
3.4 Расчет стоимости ПО	83
3.5 Сроки реализации проекта.....	84
3.5.1 Расчёт основной заработной платы производственного персонала .	85
3.5.2 Расчет затрат по социальному налогу.....	89

3.5.3 Расчет амортизационных отчислений.....	89
3.5.4 Расчет затрат на электроэнергию	90
3.5.6 Расчет стоимости по всем статьям затрат	92
3.6 ВЫВОД	94
4. БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ	95
4.1 АНАЛИЗ УСЛОВИЙ ТРУДА В ПРОИЗВОДСТВЕННОМ ПОМЕЩЕНИИ	95
4.1.1 Анализ освещенности рабочего места	96
4.1.2 Расчет искусственного освещения	98
4.1.3 Расчёт освещения методом удельной мощности.....	101
4.1.4 Расчет системы вентиляции	102
4.1.5 Оценка пожарной безопасности	106
4.2 ВЫВОДЫ	106
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	108
ПРИЛОЖЕНИЕ А	ОШИБКА! ЗАКЛАДКА НЕ ОПРЕДЕЛЕНА.

ВЕДЕНИЕ

На сегодняшний день такое понятие как биллинг все время на слуху, так как использование биллинга становится все более популярным с каждым днем.

Говоря биллинг, подразумевается целый комплекс процессов, направленных на сбор данных касающихся использования энергосбытовых услуг, их тарификации, а также выставление счетов абонентам, обработку платежей. Без биллинга просто не обойтись, ведь именно биллинг является одним из наиболее важных компонентов их деятельности.

В любой компании биллинговая система является жизненно необходимым элементом ИТ-инфраструктуры. От того насколько хорошо работает биллинг, насколько он соответствует потребностям оператора, и насколько он способен адаптироваться к изменяющимся условиям, зависит развитие бизнеса в условиях жесткой конкуренции и постоянно совершенствующихся технологий.

Использование биллинга существенно ускоряет и упрощает процесс составления счетов для клиентов - биллинг помогает наиболее эффективно организовать экономическую составляющую деятельности любой компании энергосбыта. Можно выделить основные функции биллинга, которые заключаются в проведении расчетных операций, организации информационного и финансового обслуживания. Кроме этого биллинг дает возможность контролировать доходы компании, организовывать эффективное управление прибылью, а также отслеживать возможные попытки мошенничества среди абонентов. Биллинг - это чрезвычайно удобно, выгодно и практично.

Традиционно при построении биллинга база данных (БД) используется в качестве хранилища исходных, а иногда и агрегированных данных.

При таком подходе главным требованием к используемой системе управления базами данных (СУБД) является минимальное время выполнения операций изменения, удаления и вставки строк в таблицы базы данных. Все остальные свойства и возможности СУБД, включая используемую модель организации данных, приобретают второстепенное значение.

Собственно сам биллинг представляет из себя некий программный модуль, написанный на самых различных языках программирования – от скриптовых до высокоуровневых.

Целью выполнения дипломного проекта ставится задача разработать программный модуль, который обеспечит учет потребления электроэнергии. Были поставлены следующие задачи:

- изучение и анализ информации, относящейся к теме дипломного проекта;

- проанализировать и спроектировать логическую и физическую модели пользовательского приложения;

– на основе результатов исследования обосновать необходимость разработки и разработать интерфейс биллинга.

В главе «Теоретическая часть» описывается понятие биллинговой системы электроэнергии, ее реализации. Так же, описывается система управления реляционными базами данных.

В главе «Специальная часть» идет описание предметной области, входной и выходной информации, функциональное и информационное моделирование, разработка структуры базы данных. Также включает в себя расчет размера базы данных, необходимый для функционирования системы, логическое проектирование и работа с приложением.

В «Экономической части» определяется трудоёмкость программного продукта, оценивается трудоёмкость отдельных видов работ, определяется цена научно-технического продукта и определяется экономический эффект от внедрения программного продукта.

Глава «Безопасность жизнедеятельности» включает в себя анализ состояния помещения. Нормализацию освещенности рабочего места производственного помещения. Оценку пожарной безопасности. Анализ системы вентиляции для комфортных условий труда в помещении, при которой каждый человек благодаря своей индивидуальной системе автоматической терморегуляции организма чувствовал бы себя комфортно. При обработке информации и их влияние на здоровье, и работоспособность оператора персонального компьютера, выбор и обоснование технического оснащения и организации рабочего места оператора персонального компьютера в целях оптимизации нервно-психических нагрузок, рекомендации по организации режима труда и отдыха оператора персонального компьютера.

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Биллинг

Термином «биллинг» (от англ. bill – «счёт») принято называть деятельность компании по учёту предоставляемых услуг, определению их стоимости, выставлению счетов и контролю последующих оплат.

Понятие *биллинг* уже давно широко применяется во многих сферах деятельности, которые каким-то образом связанные с тарифами, счетами и оплатами за товары, услуги. В некоторые источники, которые стремятся все русифицировать, слову «*биллинг*» противопоставляют словосочетание «*система тарификации*» или попросту – «*тарификатор*». Во всех видах связи биллинговые системы (биллинг) уже давно стали ключевыми понятиями. Разнообразные схемы тарификации услуг, механизмы оплат привели к необходимости создания автоматизированные системы учета услуг, безусловно, оплата за них. Также, в последнее время, биллинговые системы дополнились новейшими функциями – информационной поддержки принятия коммерческих решений.

В энергосбыте непременно существует сложная система тарификации услуг. И обязательно выставляются счета (*Bill-ы*) за предоставленные услуги. Также проводится анализ коммерческой деятельности, и на основании этого анализа принимаются решения о ее оптимизации с целью повышения эффективности.

Таким образом, *биллингом* в энергокомпаниях (ЭК) называют программно-алгоритмические комплексы, предназначенные для целей автоматизации процессов связанных с учетом, контролем за потреблением электроэнергии и оплат за это потребление. Следует включать методологию и организацию коммерческих процессов в понятие «биллинг», с применением средств автоматизации в деятельности (ЭК). Программное обеспечение в этом случае вторично и является лишь инструментом, отражающим данную методологию и организацию.

Коммерческая деятельность ЭК состоит в том, что отличительная особенность ее является услуга по доставке потребителю весьма специфического товара – электроэнергии. Специфичность этого товара определяется:

- поставка его только в момент потребления (есть вариант аккумуляирования электроэнергии, но ввиду его непрактичности он не рассматривается);
- фиксация объемов товарной массы только при помощи специальных приборов учета;
- большие возможности по безучетному потреблению (воровству) электроэнергии;

– объективные технические трудности по обеспечению необходимого качества товара (электроэнергии), что часто используется в качестве главного аргумента при нежелании оплачивать данный вид товара потребителем;

– порою значительные, не зависящие от организации обслуживания, потери электроэнергии в сетях передачи (доставки товара до потребителя).

Необходимо отметить наличие фактора субъективности в отношении потребителя, в особенности бытового, к электроэнергии, как к товару – потребитель очень часто воспринимает электроэнергию, как некий тип коммунальной услуги, оплату которого всегда можно оспорить. Мало того, популистские законодательные акты: списание коммунальных долгов (учитывая и электроэнергию), взаимозачеты, замораживание тарифов – создавали пренебрежительное отношение к электроэнергии, как к товару, равноправно фигурирующего на рынке товаров и услуг с другими видами товаров. И так как электроэнергия, на субъективном уровне, часто воспринимается неким видом социальной услуги, то законодательство, в разрешении конфликтов между ЭК и потребителем, чаще всего становится на сторону потребителя, что еще более снижает статус электроэнергии как товара.

Основой формирования юридических отношений «поставщик-потребитель» является «Договор компании поставщика электроэнергии и потребителя», типовая форма которого представлена в Правилах. Необходимые изменения, учитывающие методологические правила, используемые в каждой конкретной ЭК, могут фиксироваться в «Дополнении к Договору».

Вся коммерческая деятельность ЭК базируется на выше представленной методологической законодательной базе. В качестве инструмента функционирования которой часто используются автоматизированные средства по учету и контролю за потребленной электроэнергией.

Таким образом, биллинг должен обеспечить:

- наиболее реальную оценку потребленной электроэнергии (товара);
- получение и учет полноценных, своевременных оплат за отпущенную электроэнергию;
- формирование строгой финансовой и коммерческой отчетности.

1.2 Оценки объема потребления электроэнергии

Оценки потребления каждым потребителем в отдельности происходят путем снятия показаний данного потребления с приборов учета (счетчиков потребленной мощности). Данные оценки производятся в Квт/час (киловатт-часах).

Снятие показаний с приборов учета (ПУ) в настоящее время, в большинстве случаев происходит вручную в зависимости от принятой методологии: самим потребителем или ответственным лицом, представляющим интересы ЭК. Момент времени снятия показаний также зависит от исповедуемой ЭК методики.

Правила пользования электрической энергией (ППЭЭ) позволяют производить снятия показаний:

- представителями компании в период не превышающий полгода;
- бытовым пользователями ежемесячно, но с учетом времени наложения санкций этот период может составлять 60 дней (20 дней до предупреждения об оплате, плюс 20 дней после предупреждения об отключении);
- промышленным абонентам в сроки и числа, согласованные в Договоре;
- автоматическими счетчиками – в заданный период.

ППЭЭ разрешают три методики отношений между ЭК и потребителем:

- а) оплата по выставленным счетам (в обиходе «по счетам»);
- б) оплата по снятым показаниям самим потребителем с фиксацией данного потребления в абонентских книжках установленного образца (в обиходе - «по книжкам»);
- в) кредитование, предварительная оплата, абонентом предполагаемого потребления (в обиходе – «в кредит»).

Методы «по счетам» и «по книжке» можно считать прямыми методами – методами фиксирующие, а не рассчитывающие потребление. Прямыми эти методы можно считать без привязки к оцениваемому периоду потребления. В «чистой», не математической, а практической постановке вопроса оценивается не только потребление на момент времени, а потребление за установленный период. Периоды, за которые происходит учет потребления электроэнергии, обычно привязываются к отчетным бухгалтерским периодам или цикличности работы крупных промышленных потребителей.

При отсутствии ПУ и в некоторых случаях, например, связанных с учетом оценки потерь, используются косвенные или другими словами расчетные методы. Косвенные (расчетные) методы используют определенные математические правила и приемы с целью максимально приблизить реальное потребление за указанный период с показаниями, снятыми, в лучшем случае, во время действия оценочного периода.

Некоторые компании, используя возможности договора, выставляют счета с фиксированным, договорным значением потребления. Данная методика носит название «по договорному».

Интерес представляют комбинированные методы: «по счетам» плюс «по договорному», «договорное» плюс расчетное (пролонгированное, экстраполированное или расчетное значение с учетом физических закономерностей), а также и другие статистические методы (методы математической статистики), которые пока редко используются на практике.

Таким образом, чаще всего используют 6 методов оценок потребления:

- погрешность методов оценок потребления;
- метод «по книжкам»;
- метод «по счетам»;
- метод «по договорному»;
- метод кредитование;

– комбинированные методы.

1.2.1 Погрешности методов оценок потребления

Все методы оценок потребления, кроме прямого метода снятия показаний автоматическими электронными счетчиками в заданный момент времени, имеют методические погрешности.

В основе методической погрешности прямых методов лежит «бухгалтерское» правило: необходимость совпадения отчетного (бухгалтерского) и расчетного периодов. То есть, бухгалтерская отчетность должна отражать совпадение периодов бухгалтерской отчетности и периода расчета за потребленную энергию. В идеале: съем показаний, расчет за потребление и формирование бухгалтерской отчетности должны формироваться в один календарный день, и в единый момент времени – в 24.00 последнего дня месяца.

Современные технические средства (электронные счетчики с удаленным доступом, мощные компьютеры) позволяют реализовать требования данного правила, но требуют больших капитальных затрат на техническое перевооружение всего энергосбытового подразделения ЭК. Существует еще один метод, апробированный в мировой практике: отказ от данного правила или хотя бы уменьшения его строгости. Например, не привязывать строго к концу месяца отчетность перед финансовыми фискальными органами типа налоговых служб: деньги от расчетов потребителя за отпущенный товар, в виде налога на прибыль, сразу поступают на счет налоговых органов и вся отчетность, которая их интересует, формируется ими самими. А оценки эффективности работы компании вести по другим правилам и критериям, не слишком строго привязанным к определенным периодам.

1.2.2 Метод «по книжкам»

Данный метод предполагает определение потребления самим абонентом, по выставленному потреблению оплата им же.

Принципиальная погрешность: данный метод нарушает основной принцип товарно-денежных отношений при рыночной экономике – оплату за товар (услугу) по выставленной цене или договорному значению. Получается, что потребитель сам назначает цену за товар.

Погрешности метода обуславливаются:

а) субъективностью оценки *при съеме* показаний – потребитель сам определяет свое же потребление, естественно, возникает желание воспользоваться данным правом в свою пользу (уменьшить потребление, воровать);

б) субъективные погрешности *при расчете* различных льготных параметров оплачиваемого счета (расчет льготного потребления, расчет субсидий) – потребитель, в общем-то, не должен знать порою очень

замысловатых (как в случае расчета субсидий) методик расчетов льготных параметров своего потребления;

в) не совпадение расчетного и отчетного периода.

Субъективность оценок потребления данного метода предполагает снижение платежной дисциплины потребителей, привыкшего к различным социальным действиям правительства: списание долгов, определение неких форм взаимозачетов, не выгодных ЭК, которая должна рассчитываться с энергорынком «живыми» деньгами. С точки зрения психологического статуса отношений «поставщик-клиент», большой период в проведении контрольных обходов, создает прецедент возникновения ощущения безнаказанности или даже некоего прообраза чувства «хозяина положения». Что также усиливает желание производить неучтенное потребление электроэнергии - воровать.

Косвенно, данный метод также является причиной некоторых должностных преступлений работниками низшего звена ЭК, которые могут обвинить потребителя в недовыставлении потребления и приписывании ему факта «воровства». Например, пользуясь некоторой технической и юридической неграмотностью потребителя, некоторые контролеры занимались шантажом (запугиванием составлением актов о хищении) с целью вымогательства оплат за «покрывание воровства».

В итоге, как показывает практика, в лучшем случае данный метод обеспечивает 60-70% оплат за потребленную электроэнергию.

1.2.3 Метод «по счетам»

Данный метод предполагает периодическое (обычно ежемесячное) снятие показаний контролерами ЭК, с последующим выставлением счетов, доставляемых потребителям. Оплаченные счета учитываются и служат основой оценки эффективности менеджмента компании.

Погрешности метода обуславливаются:

а) не совпадением расчетного и отчетного периодов;

б) субъективными погрешностями, которые значительно меньше погрешностей метода «по книжкам», так как показания снимаются подготовленными контролерами – представителями ЭК.

Следует отметить, что погрешность не совпадения расчетного и отчетного периодов в данном методе может быть больше аналогичной погрешности метода «по книжкам». Это обуславливается тем, что, обычно, дисциплинированный потребитель (практика показывает, что количество оных составляет приблизительно 50-60% от общего числа потребителей) снимает показания в конце месяца (конец отчетного периода). А контролеры снимают показания на протяжении всего месяца (расчетного периода).

Чтобы уменьшить случайную (дисперсионную) часть погрешности менеджеры компании стараются организовать съем показаний у одного и того же потребителя в одно и то же число месяца. Таким образом, постоянная

составляющая погрешности, как бы переносится на следующий месяц, что не сильно влияет на показатели эффективности работы компании.

Данное замечание касается бытовых потребителей, так как в работе с юридическими потребителями даты съема показаний определены договорными обязательствами.

К погрешностям метода можно отнести и часть коммерческих потерь, обусловленных несвоевременными оплатами потребителями счетов по причине:

- не доставки счетов равно, как и сознательного уничтожения счета абонентом;
- не согласия потребителя с выставленными счетами;
- физической не возможности потребителя уплатить по счетам;
- нежеланием потребителя платить.

Деятельность по уменьшению доли данного вида потерь занимает собой значительную часть организационных усилий энергосбытового подразделения ЭК, работающего с бытовыми потребителями.

1.2.4 Метод «по договорному»

Хотя в ППЭЭ данный метод не указан прямо, но существует «лазейка» - установление в счете договорного потребления – потребления зафиксированного в договоре с потребителем, что не нарушает принципов товарно-денежных отношений между потребителями и поставщиками товара (услуги).

Привлекательность данного метода в том, что он как бы реализует некоторые функции метода «по счетам», но при этом значительно уменьшается количество персонала задействованного на съеме показаний. Но в данном методе заложен ряд методических погрешностей в оценке реального потребления. Мало того, этот метод противоречит основному правилу выставления счетов, зафиксированному в ППЭЭ: счет выставляется за фактическое потребление электроэнергии на время формирования счета. Этот факт может быть использован потребителем в качестве весомого аргумента при возникновении спорных вопросов с ЭК.

Основные погрешности метода:

- а) дисперсионная (случайная) составляющая – определяется несоответствием фактического потребления и договорного;
- б) неоднозначность оценок потерь;
- в) сложность в оценках эффективности работы компании.

Данный вид погрешностей настолько значителен, что компании, его практикующие, применяют ряд организационных мер по устранению данного вида погрешности:

- разрешена коррекция абонентом компании своего потребления, например, в специальном месте счета или по телефону, естественно с оплатой по скорректированному показанию;

– увеличение частоты обходов, что сводит на нет экономию трудовых ресурсов, которые сулит данный метод;

– увеличение норм контролеров, что увеличивает текучесть кадров.

Также применяются математические методы уменьшения случайной погрешности. Самым распространенным является метод пролонгации (экстраполяции). В данном случае нулевое приближение потребления рассчитывается по предыдущим значениям, в которых максимальный вес присваивается контрольным замерам.

Метод «по договорному» может явиться причиной увеличения коммерческих потерь, т.е. за счет возрастания хищений электроэнергии: уменьшается частота обходов - снижается степень риска по обнаружению внеучетного потребления.

1.2.5 Кредитование

Метод кредитования пока имеет место только в работе с промышленными потребителями. Но является перспективным в работе с бытовыми абонентами.

Кредит (предварительная оплата) часто возникает в бытовом секторе энергосбыта в методе работы «по книжкам» и может возникнуть в методе работы «по договорному» в случае, если договорное значение превышает фактическое потребление.

В случае кредитования может возникнуть ситуация, когда нужно будет выставить «отрицательные счета» - счет с отрицательным потреблением. В данном случае «отрицательное потребление» имеет не физическое, а математическое значение, а в счете абоненту может сообщаться цифра его кредита.

1.2.6 Комбинированные методы

Даже в случае принятия решения о выборе какого-то одного метода работы, частично могут применяться иные методы. Таким образом, биллинговые системы должны иметь опции, в настройках которых отражены различные методологии. Хотя бы потому, что это разрешено правилами.

Как показала практика, болезненным, в некотором смысле слова, является процесс перехода из одного метода на другой. Например, из метода «по книжкам» на метод «по счетам». Стереотип поведения потребителя привыкшего к определенной форме отношений «потребитель-поставщик» противится введению новшеств, особенно, когда при таком переходе возникают дополнительные платы, связанные с прошлыми долгами или с ошибками заполнения базы данных (конвертации базы данных). Чаще всего используются 3 комбинации методов:

- метод «по счетам» в комбинации с методом «по договорному»;
- метод «по книжкам» в комбинации с методом «по счетам» ;

– метод «по договорному» с определенными функциями метода «по книжкам».

1.2.6.1 Метод «по счетам» в комбинации с методом «по договорному»

При данном методе, в ситуации, когда контролеру не удалось снять показание с ПУ, показания выставляются по договорному значению.

Интерес представляет методика, применяемая в отдельных компаниях. В данной методике, даже в случае съема показания с ПУ контролером, значение потребления рассчитывается методом «пролонгации по договорному» от точки съема (можно «по среднему» и «по последнему»). Что позволяет уменьшить значение случайной составляющей погрешности. Также в этой компании принимаются организационные меры по уменьшению методической погрешности оценки потребления – разрешено потребителю в конце расчетного периода корректировать свои показания (по телефону, заявкой).

1.2.6.2 Метод «по книжкам» в комбинации с методом «по счетам»

Это самый сложный, с точки зрения оценок потребления и эффективности работы компании, комбинированный метод. Работают по нему, чаще всего, в переходной период – при переходе с метода «по книжкам» на метод «по счетам». Хотя, в некоторых случаях данный метод является вынужденным по причине:

- сложности доставки счетов и снятия показаний в малонаселенных районах;

- нежелание абонентов переходить на счета и ППЭЭ разрешают это.

- нежелательные, в сложившихся условиях, методологические последствия в отношениях «поставщик-потребитель» минимизируются организационными мероприятиями:

- назначение строгого графика объезда малонаселенных пунктов с целью съема показаний и вручение счетов;

- привлечение к работе по съему показаний и вручению счетов местных органов власти (сельсоветы);

- ведение с потребителем разъяснительной работы (промышленно развитых районах с привлечением прессы и телевидения);

- принятие жесткого решения на переход к единому методу, даже если это противоречит неким, не существенным, правам пользователя (часто эти права отстаиваются не с целью обеспечения оптимальности отношений, а с целью просто пользоваться правом для утверждения личной значимости).

Основные погрешности метода суммарно зависят от величин погрешностей метода «по книжкам» и метода «по счетам».

В общем, если рассматривать данный метод с точки зрения затрат на оптимизацию его функционирования, то можно сказать, что он является самым трудозатратным и не эффективным. Поэтому не следует его даже

рассматривать в качестве переходного метода работы сбытовых подразделений ЭК. Переход из одного метода на другой нужно хорошо спланировать и вводить поэтапно по различным населенным пунктам или РЭС.

1.2.6.3 Метод «по договорному» с определенными функциями метода «по книжкам»

Как было отмечено выше, метод «по договорному» один из наиболее не точных методов в части касаясь оценки фактического потребления электроэнергии, а следовательно и расчета различных параметров оценки деятельности ЭК. Строгая формализация договорного значения, в Договоре или его Приложении не позволяет гибко отслеживать изменения в потреблении.

Исходя из того, что потребителю желательно оплачивать фактическое потребление, компании работающие «по договорному» разрешают корректировать потребление потребителю самому, например, в специальном месте выставленного счета или по телефону (Интернету). Что свойственно методу «по книжкам». В данном случае, погрешности метода «по договорному» не суммируются с погрешностями метода «по книжкам» - они, предположительно, меньше. Случайная составляющая погрешности метода полностью зависит от оценочной погрешности метода «по договорному». Поэтому, та часть потребителей, которая корректирует показания потребления (это чаще всего добросовестные граждане, своей деятельностью уменьшает погрешность метода «по договорному»). В тоже время практикуется увеличение частоты контрольных обходов и придания контрольному обходу максимального приоритета в оценках потребления. [13]

1.3 БД в биллинге

Данные о потреблении электроэнергии и оплаты за него – это финансовый фундамент коммерческой деятельности ЭК. Вся эта информация хранится и используется биллинговой системой. И практически вся эта информация востребована всеми службами управления деятельностью ЭК. Отсюда следует, что следующим этапом совершенствования биллинговых систем является создание консолидированной базы данных, являющейся основой автоматизированной системы управления всей компанией.

1.4 База данных

База данных есть совокупность взаимосвязанных хранящихся вместе с отношениями между ними устойчивых (персистентных) данных при наличии такой минимальной избыточности, которая допускает их независимое использование оптимальным образом для одного или нескольких приложений.

При этом данные хранятся в таком виде, чтобы они были независимы от программ, использующих эти данные. Для добавления новых или для модификации существующих данных, а также для поиска данных в базе данных

применяется общий управляемый способ. Данные, хранящиеся в базе данных, должны удовлетворять заданным явно или неявно условиям целостности (правильности) и устойчивости (персистентности) данных.

Всем этим предполагается автоматизированная поддержка существования такой базы данных со стороны специальной системы управления базами данных (СУБД).

Архитектура БД. Существует архитектура БД, предложенная исследовательской группой ANSI/SPARC (Study Group on Data Management Systems) и называется она архитектурой ANSI/SPARC.

Уровни в архитектуре СУБД:

- внешний уровень (пользовательский);
- концептуальный уровень (логический);
- внутренний уровень (физический).

Эта архитектура позволяет обеспечить логическую (между уровнями 1 и 2) и физическую (между уровнями 2 и 3) независимость при работе с данными. Пример представлен на рисунке 1.1.

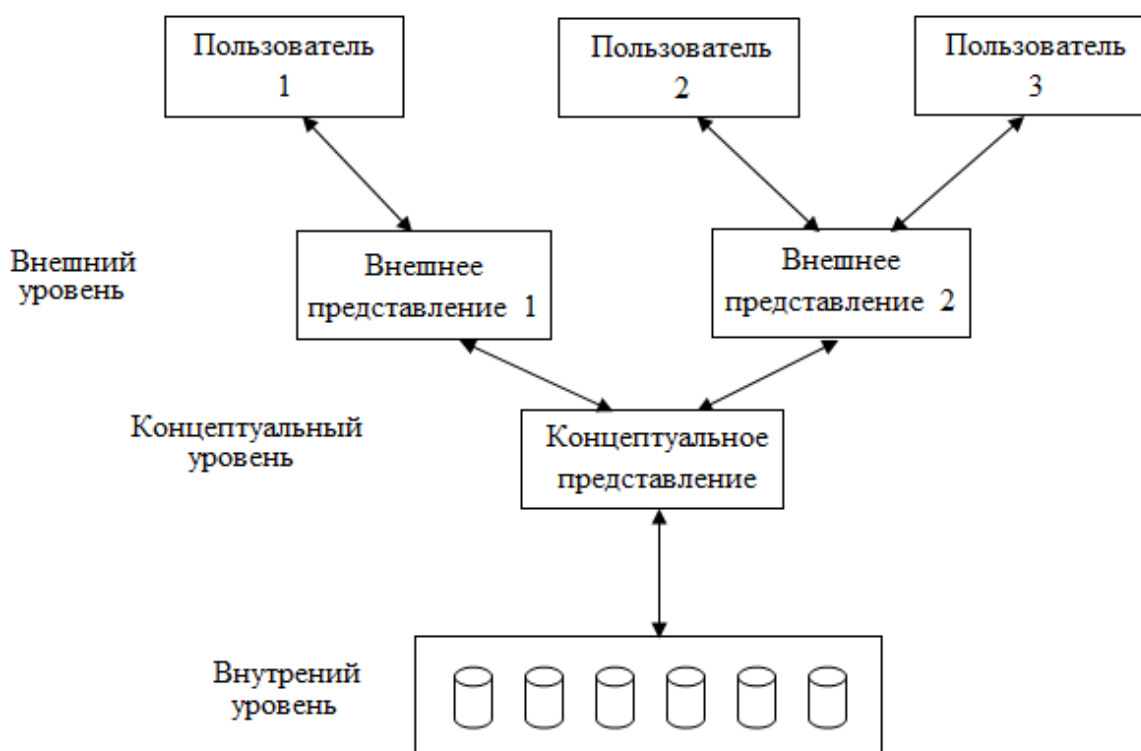


Рисунок 1.1 – Архитектура БД

Уровень внешних моделей — самый верхний уровень, где каждая модель имеет свое «видение» данных. Этот уровень определяет точку зрения на БД отдельных приложений. Каждое приложение видит и обрабатывает только те

данные, которые необходимы именно этому приложению. Например, система распределения работ использует сведения о квалификации сотрудника, но ее не интересуют сведения об окладе, домашнем адресе и телефоне сотрудника, и наоборот, именно эти сведения используются в подсистеме отдела кадров.

Концептуальный уровень — центральное управляющее звено, здесь база данных представлена в наиболее общем виде, который объединяет данные, используемые всеми приложениями, работающими с данной базой данных. Фактически концептуальный уровень отражает обобщенную модель предметной области (объектов реального мира), для которой создавалась база данных. Как любая модель, концептуальная модель отражает только существенные, с точки зрения обработки, особенности объектов реального мира.

Физический уровень — собственно данные, расположенные в файлах или в страничных структурах, расположенных на внешних носителях информации.

Эта архитектура позволяет обеспечить логическую (между уровнями 1 и 2) и физическую (между уровнями 2 и 3) независимость при работе с данными.

1.5 Общие сведения о языке SQL

SQL (Structured Query Language) – это структурированный язык запросов к реляционным базам данных (БД). SQL является непроцедурным языком программирования, применяемый для создания, модификации и управления данными в произвольной реляционной базе данных, управляемой соответствующей системой управления базами данных (СУБД). SQL основывается на исчислении кортежей.

1.5.1 Краткая история о языке SQL

В начале 70-х годов плодотворный труд исследователя из IBM доктора Кодда (E. F. Codd) привел к созданию продукта, связанного с реляционной моделью данных под названием SEQUEL (Structured English Query Language, структурированный английский язык для запросов), который в 1980 г. был переименован в SQL (Structured Query Language, структурированный язык запросов).

Компания IBM, а также другим производителям реляционных баз данных был нужен стандартизованный метод доступа к реляционной базе и манипулирования хранящимися в ней данными. Хотя компания IBM первая разработала теорию реляционных баз данных, первой на рынок с этой технологией вышла компания Oracle. Через какое-то время SQL завоевал на рынке достаточную популярность и привлек внимание Американского национального института стандартизации (American National Standards Institute, ANSI), который в 1986, 1989, 1992, 1999 и 2003 годах выпустил стандарты языка SQL.

Начиная с 1986 года несколько конкурирующих между собой языков позволяли программистам и разработчикам обращаться к реляционным данным и манипулировать ими. Программисты и администраторы теперь могли знать один язык, который с небольшим изменениями можно применять к разнообразным платформам баз данных, приложениям и прочим продуктам.

1.5.2 Реляционная модель и ANSI SQL

Система управления реляционными базами данных (Relation Database Management Systems) – это основа информационных систем по всему миру, и особенно в веб-приложениях и распределенных системах типа клиент-сервер. Они позволяют множеству пользователей быстро и одновременно обращаться к данным, создавать их, редактировать и манипулировать ими, не мешая другим пользователям и не оказывая влияния на их действия. Они также позволяют разработчикам создавать удобные приложения для доступа к ресурсам, а также предоставляют администраторам необходимые средства для обслуживания, обеспечения безопасности и оптимизации данных в организации.

Реляционная СУБД – это система, пользователи которой видят данные в форме набора таблиц, связанных друг с другом посредством общих значений. Данные хранятся в таблицах, которые состоят из строк и столбцов. Таблицы содержащие независимые данные, можно связать (или соотнести) друг с другом, если в каждой из них есть столбец уникальных идентификационных данных (называемых ключами), которые представляют те данные, которые являются общими для таблиц. Согласно реляционной модели Кодда данные были структурированными (в таблицы, состоящие из строк и столбцов), ими можно манипулировать с помощью таких операций, как выборка (selection), проекция (projection) и соединение (join), и они оставались непротиворечивыми как результат правил целостности, таких, как целостность ключей и ссылочных данных. Кодд также обозначил правила проектирования БД. Процесс применения этих правил называется *нормализацией*.

1.5.3 Анатомия реляционных баз данных

В соответствии с реляционной моделью данные в реляционной базе данных сохраняются в отношениях, или связях, которые воспринимаются пользователем как таблицы. Каждое отношение состоит из *кортежей* (записей) и *атрибутов* (полей).

Пример таблицы представлен на рис. 1.2.

Также реляционная база данных имеет некоторые другие характеристики, которые описываются ниже.

Address_ID	District	Street	House_Number	Flat_Number
101	Bostandyk	Jandosov	6	45
102	Auezov	Baizakov	15	16
103	Alatau	Abai	122	38

Поля

} Записи

Рисунок 1.2 – Пример таблицы

1.5.3.1 Отношение

Отношения (таблицы) являются основными структурами в базах данных. Каждая таблица описывает отдельный предмет. Логический порядок записей и полей в таблице совершенно не имеет значения. Каждая таблица содержит хотя бы одно поле (называемое первичным ключом), которое однозначно идентифицирует каждую из записей.

1.5.3.2 Атрибуты

Атрибут (поле) является атомарной структурой в базе данных, и оно представляет собой характеристику предмета таблицы, к которой оно относится. Поля реально используются для хранения данных. Данные из этих полей можно извлекать и представлять как информацию почти в любой воображаемой форме. Качество информации, получаемой из таблицы, прямо пропорционально времени, посвященному обеспечению структурной целостности и целостности данных в самих полях.

Каждое поле в базе данных, спроектированной надлежащим образом, содержит одно и только одно значение, а имя поля идентифицирует тип хранимого в нем значения. Это само по себе делает ввод данных в поле интуитивным.

1.5.3.3 Кортеж

Кортеж (запись) представляет собой уникальный экземпляр предмета таблицы. Она составляется из полного набора полей в таблице независимо от того, содержат ли эти поля какие-либо значения или нет. Вследствие способа определения таблицы каждая запись идентифицируется в базе данных уникальным значением в поле первичного ключа этой записи.

Записи являются ключевым фактором в понимании отношений между таблицами, так как запись одной таблице связана с записями в другой таблице.

1.5.3.4 Ключи

Ключи являются специальными полями, которые играют совершенно особую роль в таблице. Тип ключа определяется его предназначением. Хотя

таблица может содержать несколько типов ключей. Наиболее важными считаются *первичный ключ* и *внешний ключ*.

Первичный ключ – это атрибут (или группа полей), которое уникальным образом идентифицирует каждую запись в таблице. Когда первичный ключ состоит из двух или более полей, он называется составным первичным ключом. Первичный ключ является наиболее важным из всех по двум причинам: его *значение* идентифицирует конкретную запись из всей базы данных, а его *поле* идентифицирует указанную таблицу во всей базе данных. Первичные ключи обеспечивают целостность на уровне таблиц и помогают установить связи с другими таблицами. Каждая таблица в базе данных должна иметь первичный ключ.

При установлении связи двух таблиц друг с другом берут копию первичного ключа из первой таблицы и вставляют ее во вторую таблицу, где она становится внешним ключом.

Пример приведен на рисунке 1.3.

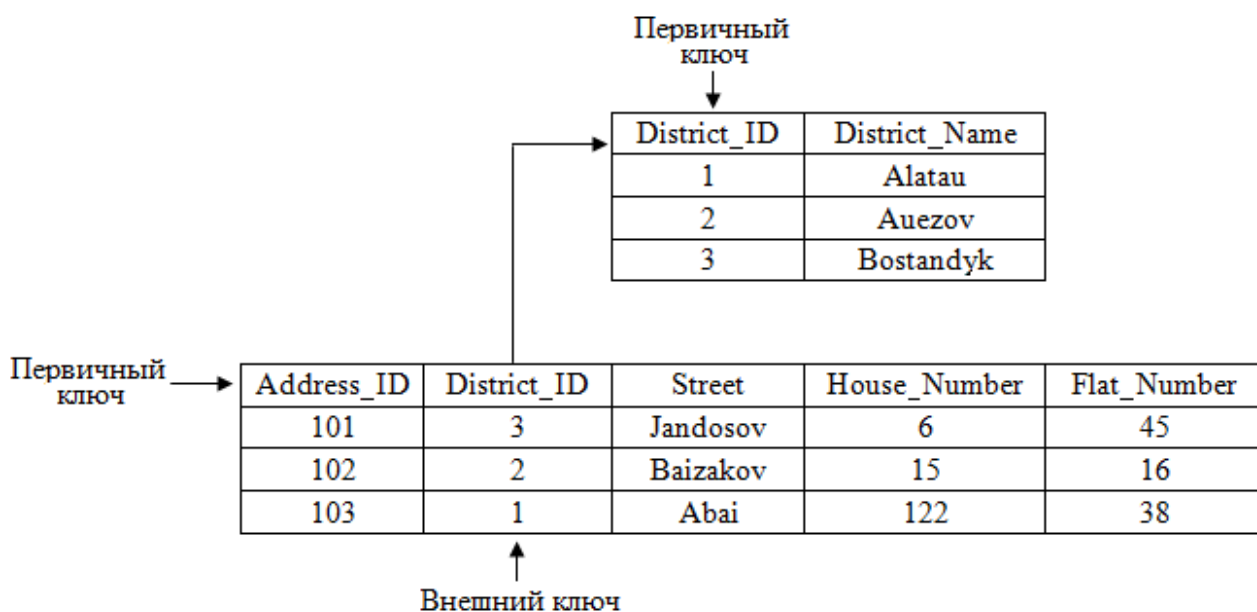


Рисунок 1.3 – Первичный и внешний ключи

Внешний ключ важен не только потому, что он помогает установить связь между двумя таблицами, но он также обеспечивает целостность на уровне связей (т.н. ссылочную целостность) базы данных. Это означает, что записи в обеих таблицах всегда будут правильно связаны.

1.5.3.5 Представления

Представление является виртуальной таблицей, которая составлена из полей одной или нескольких таблиц базы данных. Таблицы, из которых составляют представление, называются *базовыми*. В реляционной модели представление считается виртуальной таблицей, поскольку оно извлекает

данные из базовых таблиц, а не хранит все данные в себе. Единственной информацией о представлении, которая сохраняется в базе данных, является ее структура.

Представления позволяют рассматривать информацию в базе данных с различных точек зрения, обеспечивая тем самым большую гибкость в работе с данными.

Во многих системах управления реляционными базами данных представление обычно реализовано в виде хранимого запроса. В большинстве случаев запрос имеет все характеристики представления, а его единственное отличие состоит в том, что на него ссылаются по другому имени.

Пример типичного представления показан на рисунке 1.4.

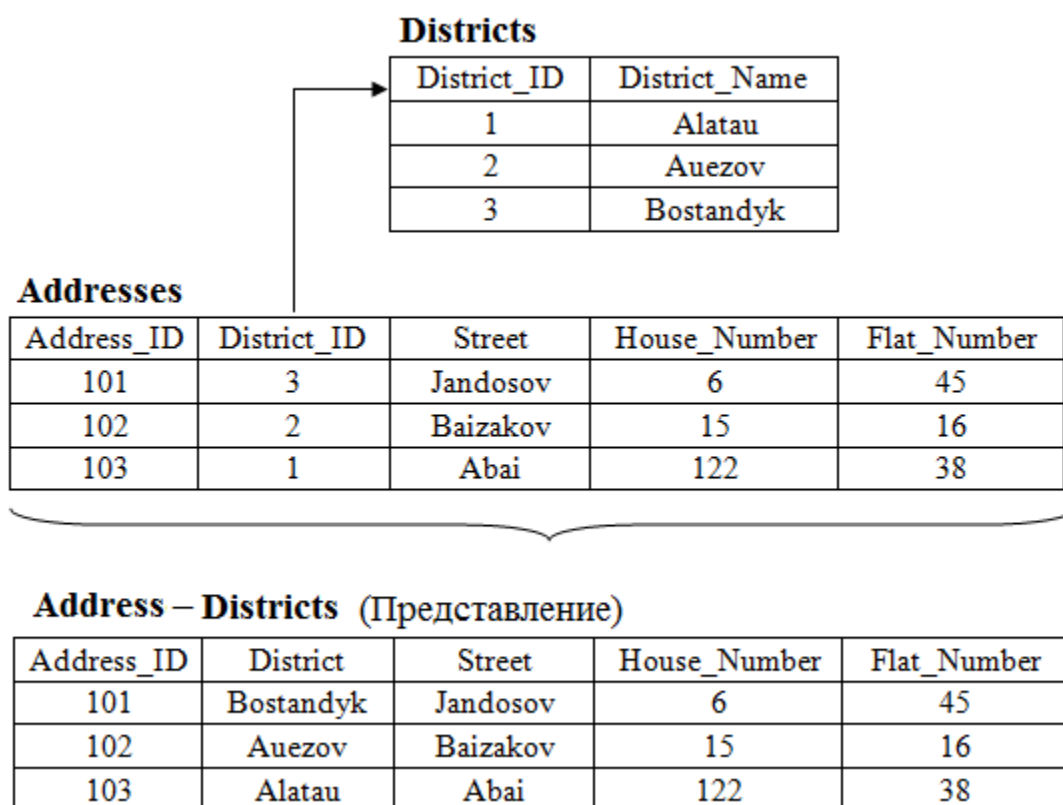


Рисунок 1.4 – Пример представления

1.5.3.6 Связи

Если записи указанной таблицы могут быть связаны некоторым образом с записями в другой таблице, то говорят, что между таблицами имеется связь. Способ установления связи зависит от типа связи.

Между двумя таблицами может существовать три типа связей:

– *Связь «один-к-одному»*. Пара таблиц связана отношением один-к-одному, если одна запись в первой таблице связана только с одной записью во второй таблице, а одна запись во второй таблице связана только с одной записью в первой таблице. В таком типе связи одна таблица называется первичной, а другая – вторичной. При связывании берется первичный ключ из

первой таблицы и вставляется во вторую таблицу, где он становится внешним ключом. Это особый тип связи, поскольку во многих случаях внешний ключ также является первичным ключом таблицы.

Пример представлен на рисунке 1.5.

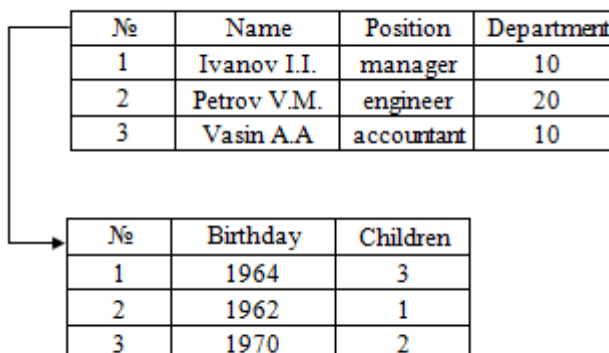


Рисунок 1.5 – Связь «один-к-одному».

– *Связь «один-ко-многим»*. Когда пара таблиц связана между собой по типу один-ко-многим, то одна запись в первой таблице может быть связана с несколькими записями во второй таблице, но одна запись во второй таблице может быть связана только с одной записью в первой таблице. При установлении такой связи берется первичный ключ в таблице со стороны «один» и вставляется в таблицу со стороны «многие», в которой он становится внешним ключом.

Пример представлен на рисунке 1.6.

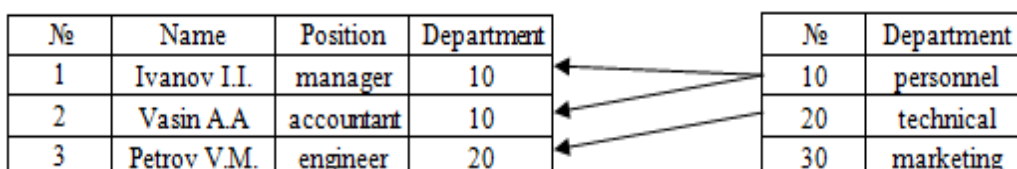


Рисунок 1.6 – Связь «один-ко-многим».

– *Связь «многие-ко-многим»*. Две таблицы связаны отношением многие-ко-многим, когда одна запись в первой таблице может быть связана с несколькими записями во второй таблице, а она запись во второй таблице может быть связана с несколькими записями в первой таблице. Для установления такого типа связи, необходимо создать так называемую связывающую таблицу, которая обеспечит легкий способ сопоставления записей из одной таблицы с записями другой и поможет гарантировать отсутствие проблем при добавлении, удалении или модификации любых связанных данных. Важно отметить, что при определении связывающей таблицы копия первичного ключа каждой таблицы используется для

образования структуры новой таблицы. Эти поля в действительности выполняют две различные роли: вместе они образуют составной первичный ключ связывающей таблицы, а по отдельности каждое из них является внешним ключом [4].

Пример связи «многие-ко-многим» представлен на рисунке 1.7.

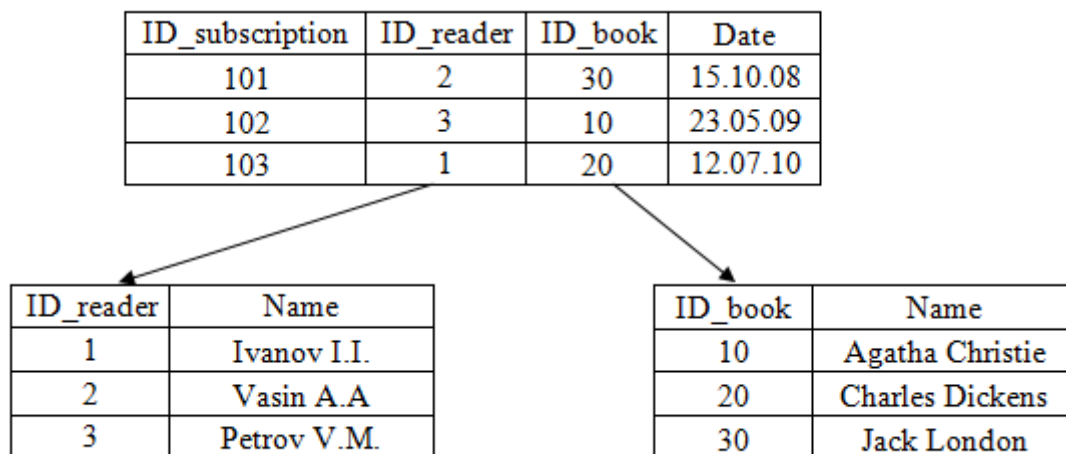


Рисунок 1.7 – Связь «многие-ко-многим».

1.6 Клиент-серверная СУБД

Клиент-серверная СУБД позволяет обмениваться клиенту и серверу минимально необходимыми объёмами информации. При этом основная вычислительная нагрузка ложится на сервер. Клиент может выполнять функции предварительной обработки перед передачей информации серверу, но в основном его функции заключаются в организации доступа пользователя к серверу.

В большинстве случаев клиент-серверная СУБД гораздо менее требовательна к пропускной способности компьютерной сети, чем файл-серверная СУБД, особенно при выполнении операции поиска в базе данных по заданным пользователем параметрам, т.к. для поиска нет необходимости получать на клиент весь массив данных: клиент передаёт параметры запроса серверу, а сервер производит поиск по полученному запросу в локальной базе данных. Результат выполнения запроса, который обычно на несколько порядков меньше по объёму, чем весь массив данных, возвращается клиенту, который обеспечивает отображение результата пользователю.

Сервер баз данных осуществляет целый комплекс действий по управлению данными. Основными его обязанностями являются:

- выполнение пользовательских запросов на выбор и модификацию данных и метаданных, получаемых от клиентских приложений, функционирующих на персональных компьютерах локальной сети;
- хранение и резервное копирование данных;

- поддержка ссылочной целостности данных согласно определенным в базе данных правилам;
- обеспечение авторизованного доступа к данным на основе проверки прав и привилегий пользователей;
- протоколирование операций и ведение журнала транзакций.

В качестве рабочего места пользователя может быть использован обычный персональный компьютер, что позволяет не отказываться от привычной рабочей среды. Иными словами, в простейшем случае клиент-серверная информационная система состоит из двух основных компонентов:

- сервера баз данных, управляющего данными и выполняющего запросы клиентских приложений;
- клиентских приложений, предоставляющих интерфейс пользователя и посылающих запросы к серверу.

Пример представлен на рисунке 1.8.

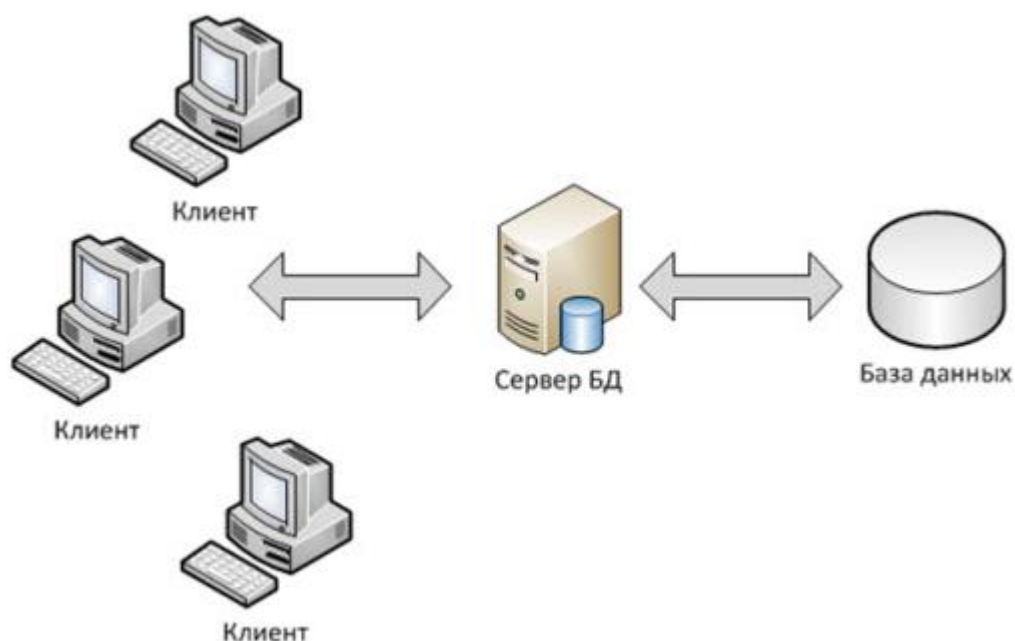


Рисунок 1.8 – Клиент-Сервер

1.7 СУБД SQL Server 2008

Microsoft SQL Server — система управления реляционными базами данных (СУРБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями.

Microsoft SQL Server 2008 - одна из наиболее мощных СУБД архитектуры клиент - сервер. Эта СУБД позволяет удовлетворять такие требования, предъявляемые к системам распределенной обработке данных, как

тиражирование данных, параллельная обработка, поддержка больших баз данных на относительно не дорогих аппаратных платформах при сохранении несмежного управления.

Важно отметить, что MS SQL Server не предназначен непосредственно для разработки пользовательских приложений, а выполняет функции управления базой данных. Для пользовательского приложения SQL Server является мощным источником генерации и управления нужными данными.

Microsoft SQL Server предназначен исключительно для поддержки систем, работающих в среде клиент-сервер. Он поддерживает широкий спектр средств разработки и максимально прост в интеграции с приложениями, работающими на ПК.

SQL Server 2008 позволяет разработчикам управляться с данными любых типов - от традиционных до географических. Это открывает дорогу к созданию приложений нового поколения с учетом информации о расположении и возможностью управления документами. SQL Server 2008 способен формировать отчеты и выполнять анализ любого объема и сложности, одновременно облегчая пользователям доступ к данным за счет более тесной интеграции с Microsoft Office.

Так же позволяет шифровать базу данных, файлы данных или файлы журналов, не внося в приложения никаких изменений. Благодаря этому становится возможен поиск в зашифрованных данных как по диапазону, так и с нечеткими критериями, а также поиск в защищенных данных, полученных от неавторизованных пользователей. Шифрование позволяет организациям соблюсти юридические требования и идти в ногу с общим стремлением к защищенности данных. SQL Server 2008 полностью соответствует этой растущей потребности, поддерживая HSM (Hardware Security Module, аппаратный модуль безопасности) и системы управления ключами сторонних производителей.

2 СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1 Основные положения

Задачей модели системы биллинговой базы данных является автоматизация учета электропотребления всех потребителей.

При работе с системой пользователь должен иметь возможность решать следующие задачи:

- Добавлять новых абонентов и изменять, удалять информацию о существующих;
- Проверять данные об абонентах.

Данная система имеет основные назначения:

- предназначение системы;
- требования пользователя.

Предназначение системы

В энергосбыте существует сложная система тарификации услуг энергетической компании (ЭК). И также выставляются счета (*Bill-ы*) за предоставленные услуги. И также проводится анализ коммерческой деятельности, и на основании этого анализа принимаются решения о ее оптимизации с целью повышения эффективности.

Таким образом, *биллингом* в энергокомпаниях (ЭК) называют программно-алгоритмические комплексы, предназначенные для целей автоматизации процессов связанных с учетом и контролем за потреблением электроэнергии и оплат за это потребление.

Требования пользователя

– система должна быть проста в использовании. Интерфейс приложения должен отвечать всем требованиям эргономичности, с тем, чтобы пользователь во время работы с клиентом не испытывал никаких неудобств;

– система должна обеспечивать безопасность. Это достигается с помощью дифференцированного подхода к пользователям системы и разграничения прав доступа в соответствии с принадлежностью к той или иной группе пользователей;

– система должна быть полностью интегрирована, что подразумевает исключение избыточности данных и избыточность обновлений. Система также должна гарантировать целостность БД:

– работа клиентского приложения никоим образом не должна зависеть от работы других программных приложений за исключением программного обеспечения баз данных.

2.2 Анализ предметной области

В качестве объекта исследования было выбрано автоматизация биллинга для компаний занимающихся электросбытом.

Биллинговая система по праву считается ядром автоматизации бизнеса любой компании, предоставляющей услуги за электропотребление.

Традиционно при построении биллинговых систем база данных (БД) используется в качестве хранилища исходных, а иногда и агрегированных данных. В некоторых случаях кроме основного хранилища используется ещё и дополнительная база данных для целей предбиллинга.

Собственно сама биллинговая система представляет из себя некий набор программных модулей, написанных на самых различных языках программирования – от скриптовых до высокоуровневых.

В итоге каждый из этих модулей работает с данными, хранящимися в базе. В зависимости от назначения модуля производятся операции ввода, удаления или модификации данных, и почти всегда ещё и чтения.

Так как разработка биллинговой системы это сложный и трудоемкий проект, на которое затрачивается большой срок и самое важное, что биллинговая система связана с оборудованием. Поэтому разработкой биллинговой системы для компаний энергосбыта занимаются компании, которые разрабатывают программное обеспечение и информационные системы.

В данном дипломном проекте по автоматизации биллинга системы электросбыта разрабатывается модуль биллинга электропотребления, в котором содержится необходимые данные, взятые из модулей «Договора», «Начисления» и «Оборотно-сальдовая ведомость».

2.3 Основная задача

Проектирование биллинговой базы данных биллинга необходимо для автоматизации: регистрации физических и юридических лиц, сбор информации об используемых тарифах, регистрации пользователей.

Спецификация данных, необходимых в данной системе

Адреса:

- В адресах содержится информация о районах, микро районах, улицах, номера домов и квартир.
- По каждому району соответствует определенная улица.

Физические лица:

- Каждое физическое лицо должно быть зарегистрировано в базе данных.
- По каждому физическому лицу должна быть обязательная информация: полное и сокращенное ФИО, дата рождения, телефонный номер, РНН, ИИН, Номер (паспорта/удостоверения) и не обязательное: E-mail, место работы, вид занятости.

- При регистрации физического лица необходима дополнительная информация удостоверяющая личность: вид документа (паспорта/удостоверения), дата выдачи документа и срок действия документа.
- Адрес проживания.

Юридические лица (организации):

- Каждая организация должна иметь информацию: сокращенное наименование организации, полное наименование на русском и казахском, дата регистрации, РНН, КБЕ (Код Бенефициара), ИИН/БИН.
- У каждой организации иметься владелец, которого регистрируют в физические лица.
- При регистрации организации необходимо иметь дополнительную информацию: сектор экономики организации, вид деятельности, головная организация.
- Также к регистрации организации указывается адрес для доставки электроэнергии, телефонный номер и электронная почта.

Пользователи:

- Для каждого пользователя иметься «логин», пароль, дата регистрации.
- У пользователя есть время для доступа в систему.
- При регистрации пользователя необходима информация о пользователе, которая находится в физических лицах.

Договора:

- Договор – это соглашение на поставку электроэнергии между физическими, юридическими лицами с компанией энергосбыта.
- У каждого договора имеется индивидуальный номер, который состоит из 6 цифр.
 - При регистрации договора обязательно указывается тариф, дата заключения договора и срок его действия.
 - В договоре также указывается потребитель, уполномоченное лицо, отраслевая группа и услуга.

Начисление:

- Начисление – это сумма предоставленная потребителю за потребленную электроэнергию.
- Начисление производится по номеру договора, в котором указан тариф.
 - Количество кВт энергии вводится вручную.
 - Если в начислении указано количество кВт, то сумма по тарифу и НДС высчитывается автоматически.

Оборотно-сальдовая ведомость:

Оборотно-сальдовая ведомость — это бухгалтерский документ, который содержит сумму начисления на начало и на конец периода и обороты по дебету и кредиту за данный период для каждого договора.

– Оборотно-сальдовая ведомость содержит информацию о потреблении электроэнергии за определенный период.

2.4 Бизнес правила

Каждый пользователь является физическим лицом, но не каждое физическое лицо является пользователем. Пример показан на рисунке 2.1.

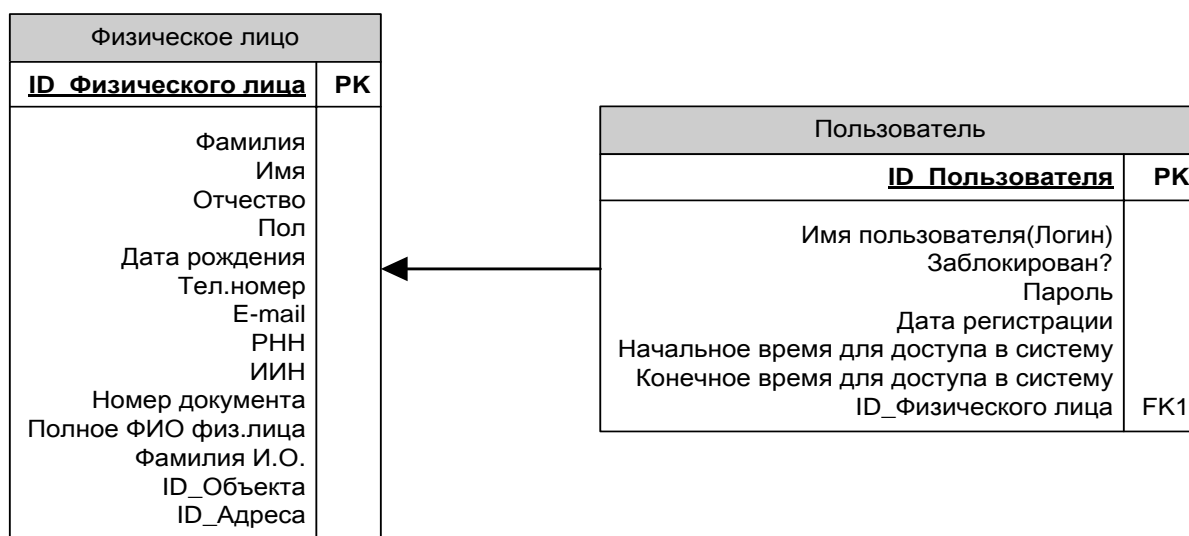


Рисунок 2.1 – Бизнес правило 1

У каждого физического лица имеются персональные данные по документам (Объекты). Пример представлен на рисунке 2.2.

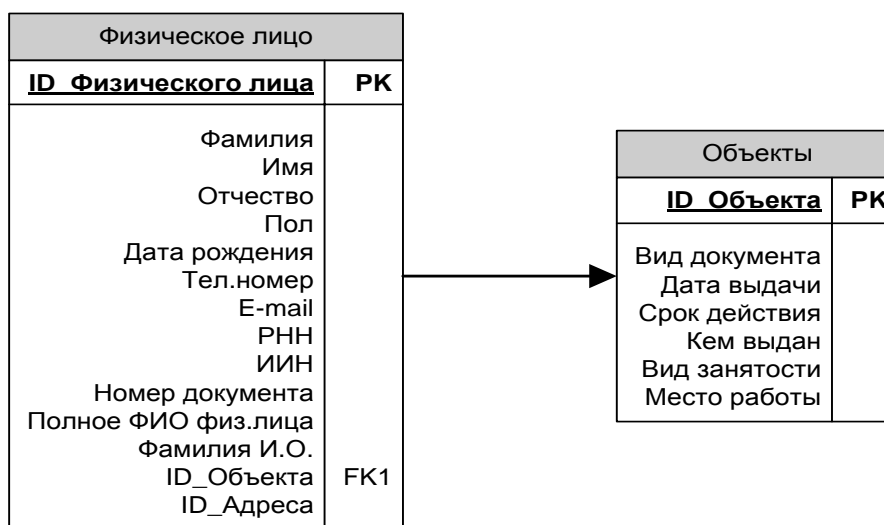


Рисунок 2.2 – Бизнес правило 2

Одно физическое лицо имеет один адрес место жительства, а на один адрес может быть много физических лиц. Пример показан на рисунке 2.3.

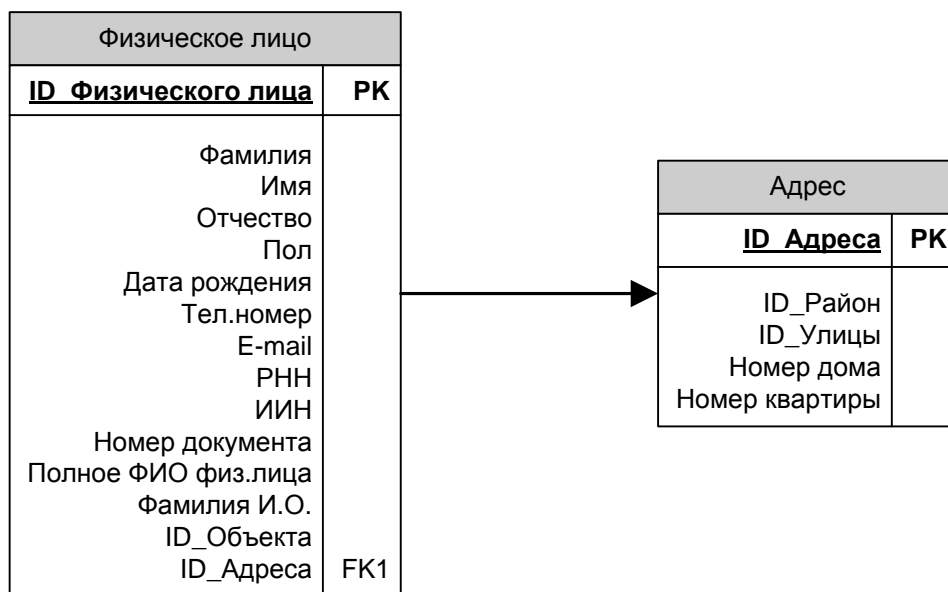


Рисунок 2.3 – Бизнес правило 3

Каждый адрес относится к определенному району, а каждый район может иметь множество адресов. Пример представлен на рисунке 2.4.

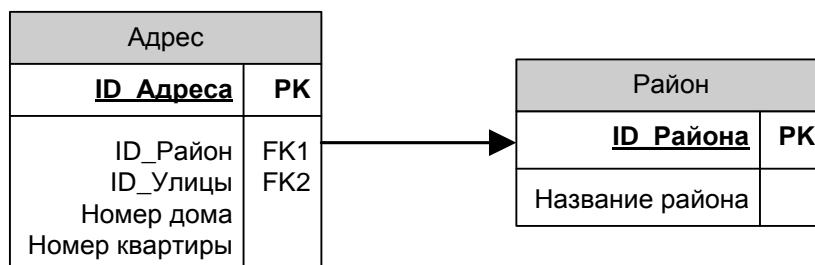


Рисунок 2.4 – Бизнес правило 4

Один адрес может относиться к одной улице, а одна улица может иметь несколько адресов. Пример показан на рисунке 2.5.

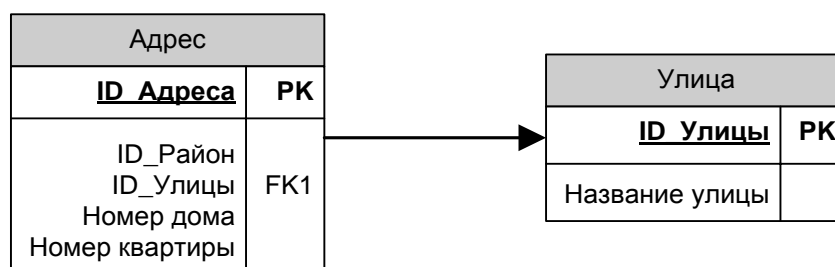


Рисунок 2.5 – Бизнес правило 5

На основе четвертого и пятого бизнес – правил получаем сегмент ER-модели, который представлен на рисунке 2.6.

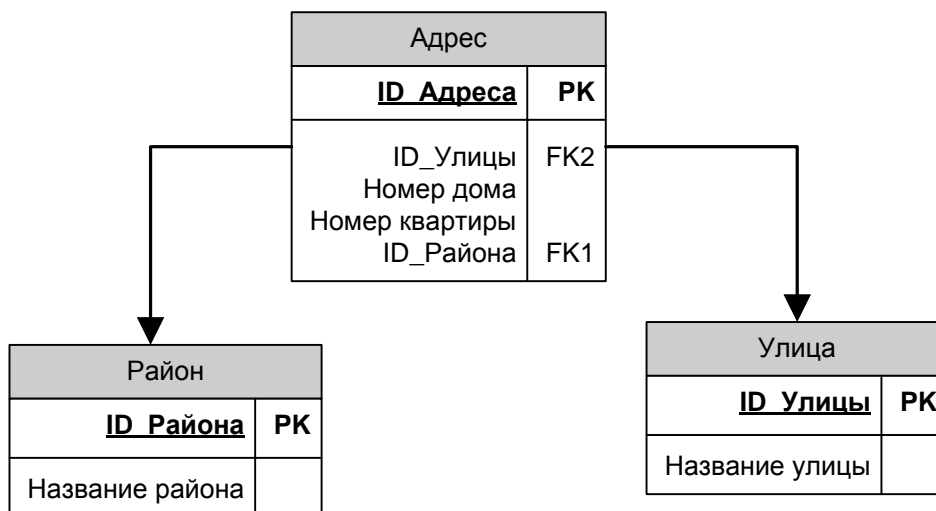


Рисунок 2.6 – Сегмент ER-модели 1

Каждая организация имеет владельца(физическое лицо), но не каждое физическое лицо имеет организацию. Пример показан на рисунке 2.7.

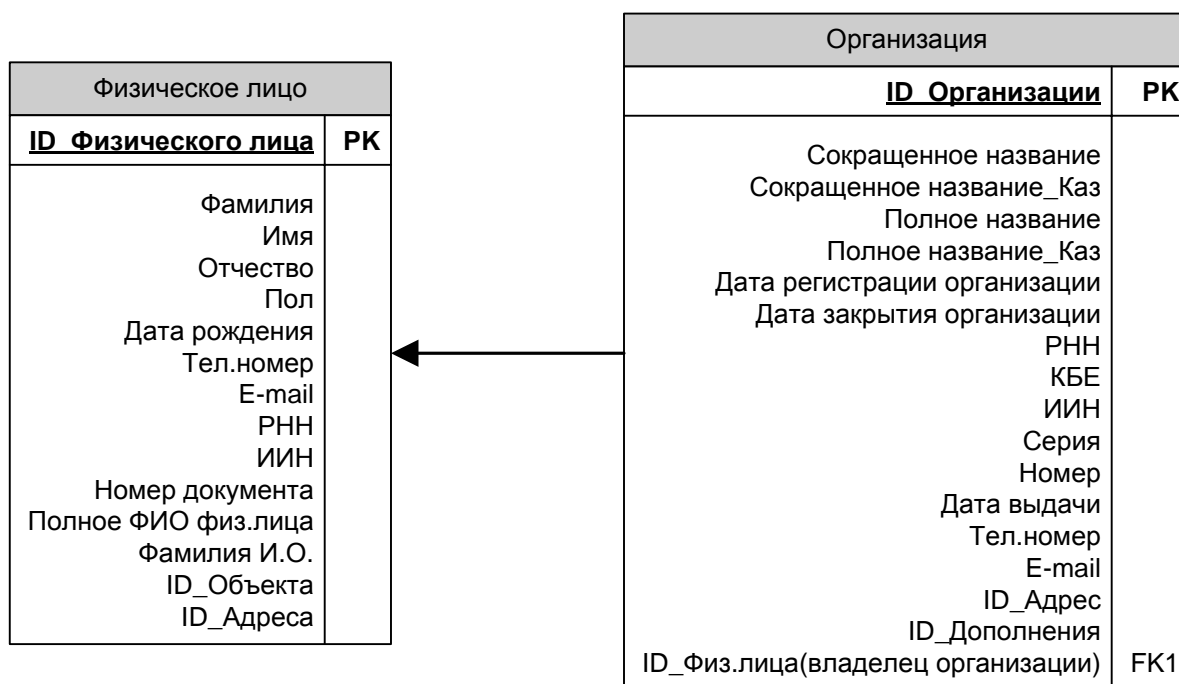


Рисунок 2.7 – Бизнес правило 6

У каждой организации имеются дополнительные сведения. Пример представлен на рисунке 2.8.

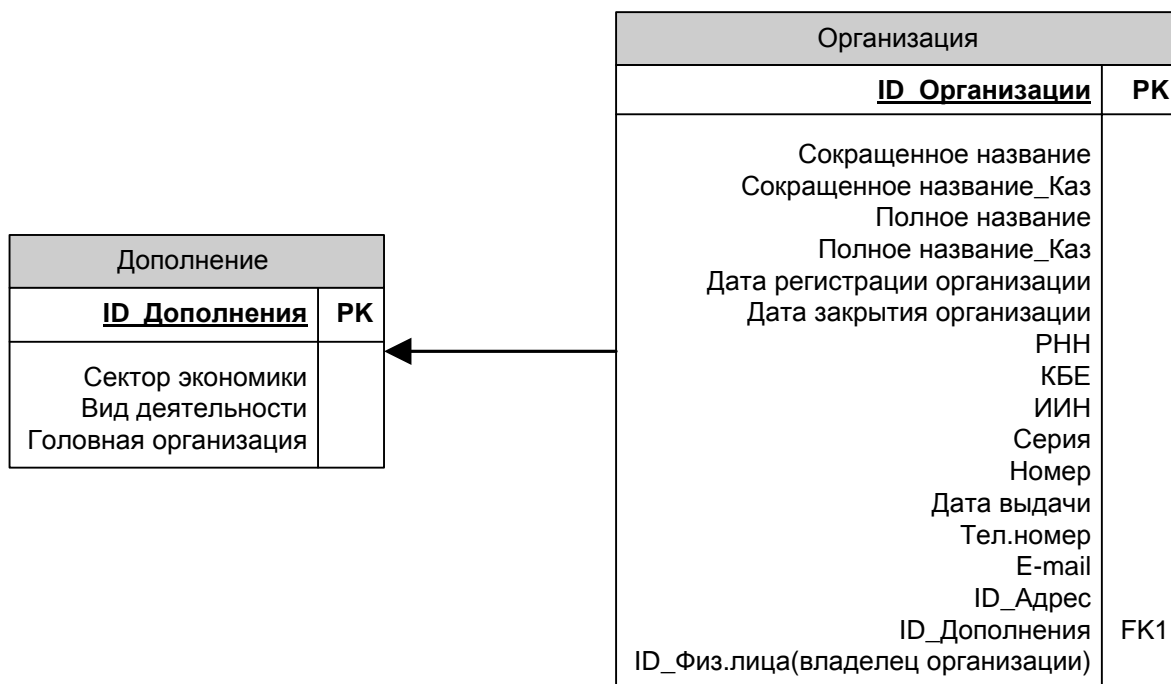


Рисунок 2.8 – Бизнес правило 7

Одна организация имеет один адрес, а на один адрес может быть много организаций. Пример показан на рисунке 2.9.

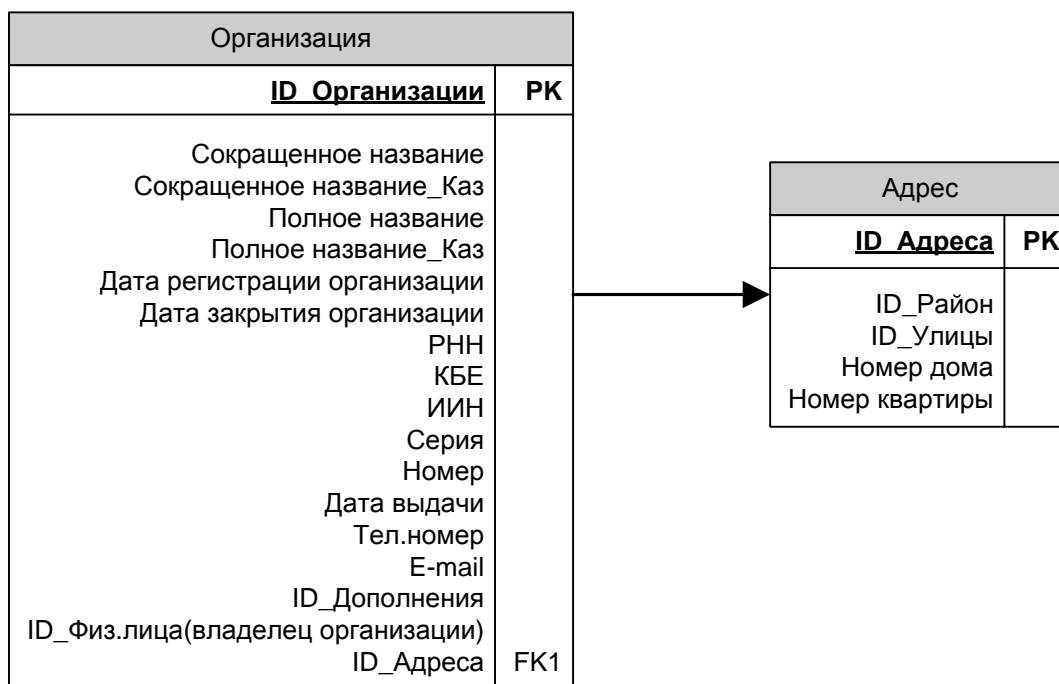


Рисунок 2.9 – Бизнес правило 8

Потребитель может принадлежать одному типу (Физическое лицо, Организация), а один тип имеет несколько потребителей. Пример представлен на рисунке 2.10.

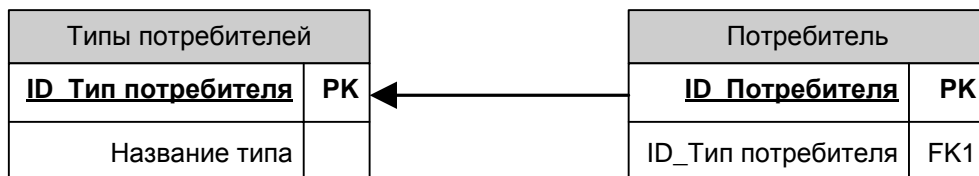


Рисунок 2.10 – Бизнес правило 9

Потребитель образует подтипы – организация и физическое лицо, которые имеют разные атрибуты, следовательно, не могут быть объединены в одну таблицу. Уникальность кода потребителя в таблицах Физическое лицо и Организация контролируется триггером. Пример приведен на рисунке 2.11.

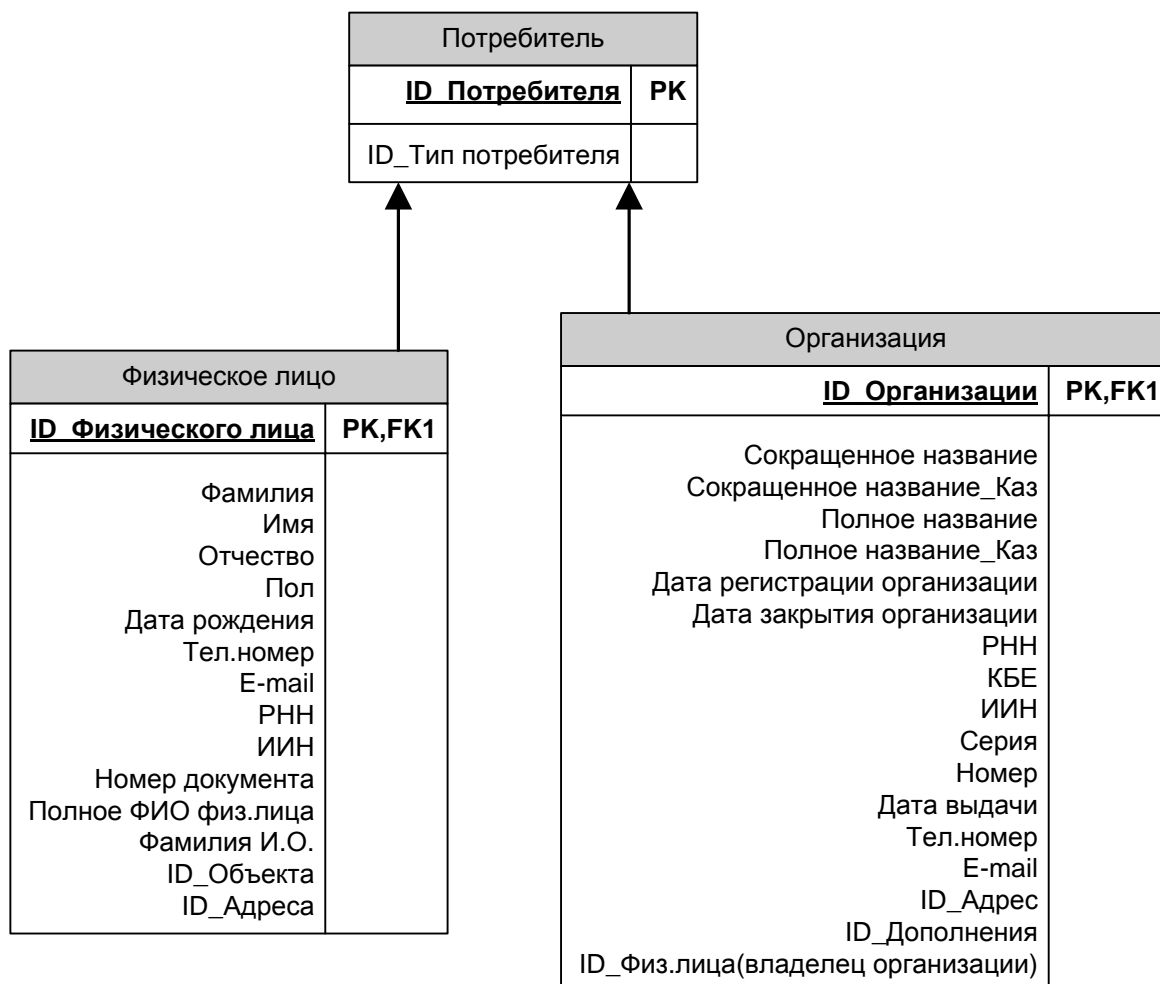


Рисунок 2.11 – Бизнес правило 10

На основе девятого, восьмого бизнес – правил получаем сегмент ER-модели, который представлен на рисунке 2.12.

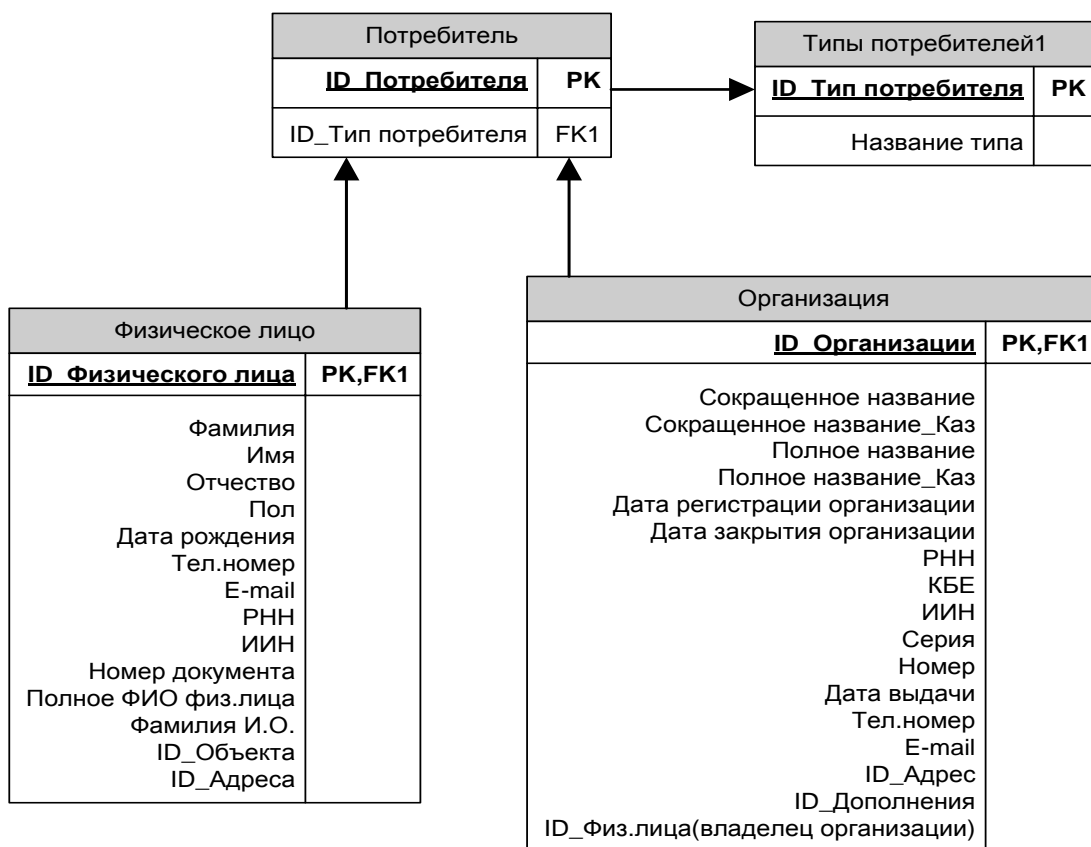


Рисунок 2.12 – Сегмент ER-модели 2

Один договор может заключаться с одним потребителем, а один потребитель может иметь много договоров. Пример показан на рисунке 2.13.

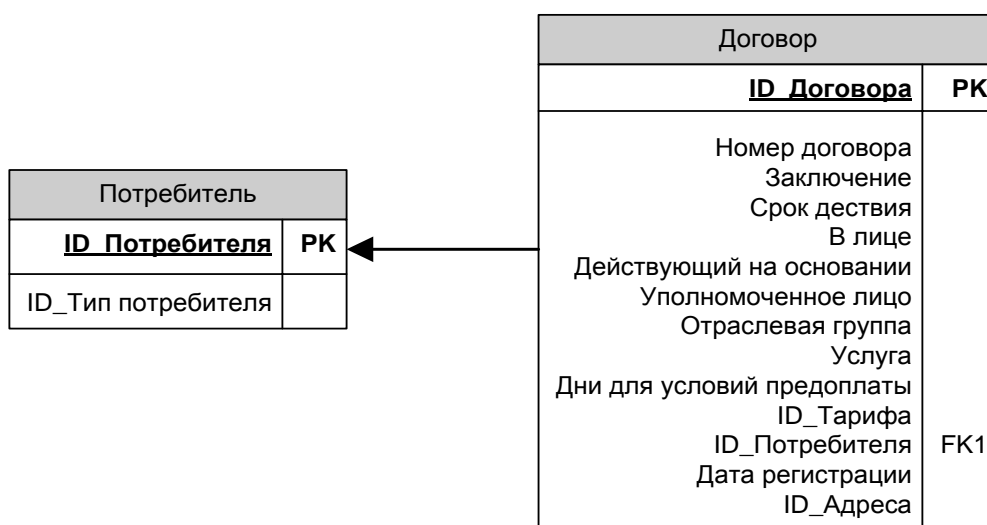


Рисунок 2.13 – Бизнес правило 11

В одном договоре указывается один тариф, а один тариф может быть во многих договорах. Пример представлен на рисунке 2.14.

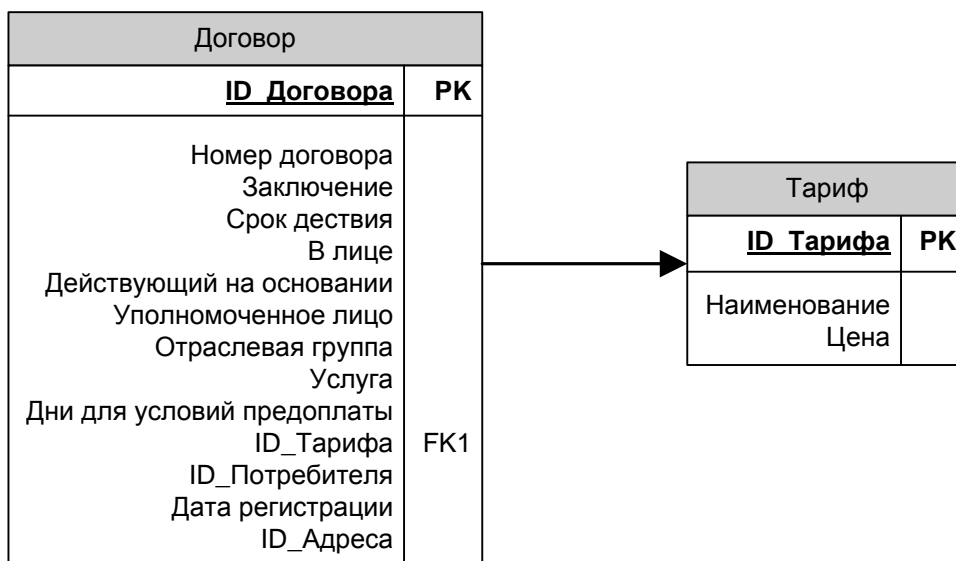


Рисунок 2.14 – Бизнес правило 12

В каждом договоре указывается один адрес доставки электроэнергии. Пример приведен на рисунке 2.15.

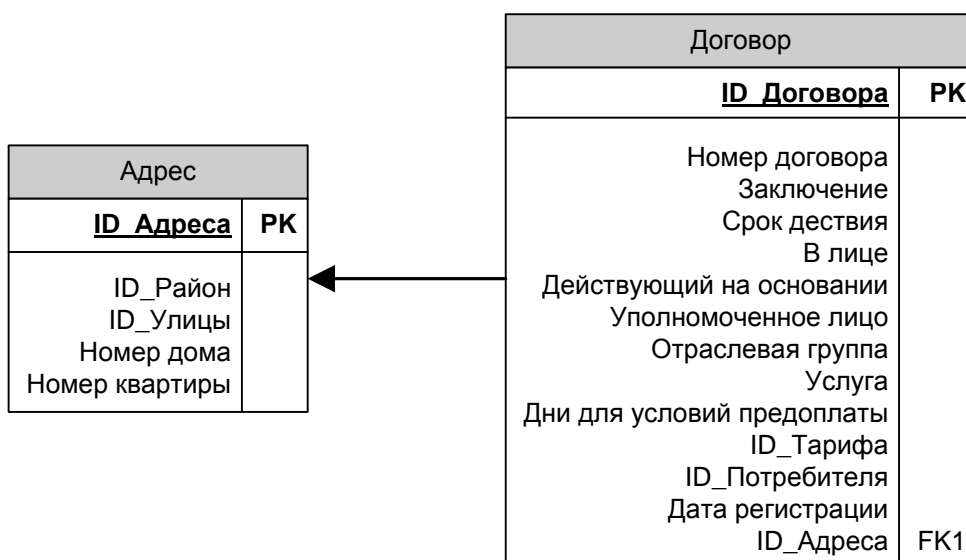


Рисунок 2.15 – Бизнес правило 13

На каждый договор проводится начисление. Пример представлен на рисунке 2.16.

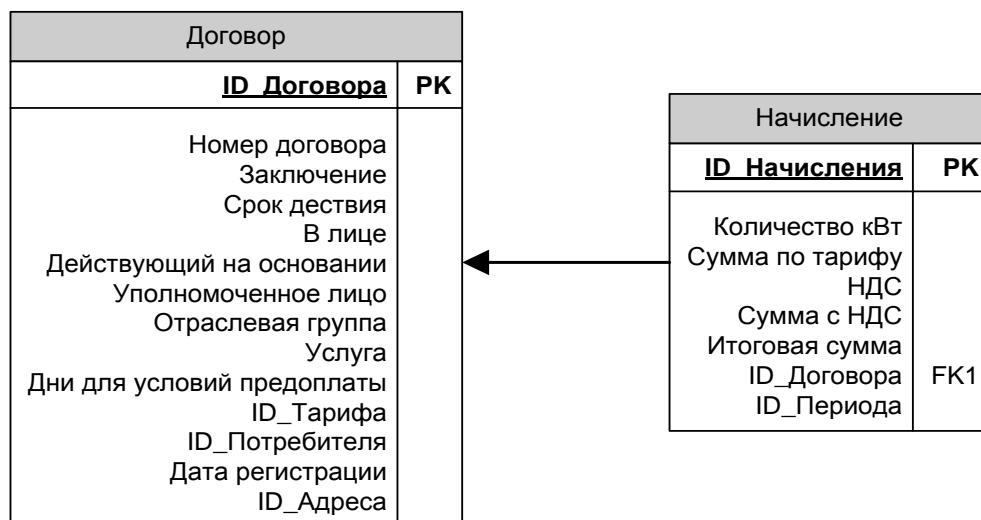


Рисунок 2.16 – Бизнес правило 14

За один период может быть несколько начислений, а одно начисление проводится на один период. Пример приведен на рисунке 2.17.

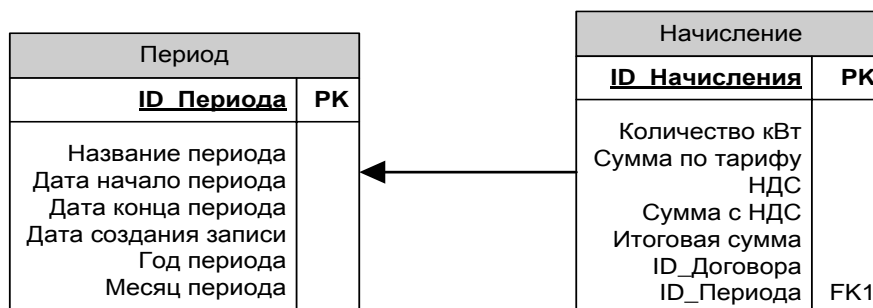


Рисунок 2.17 – Бизнес правило 15

На одно сальдо указывается одно начисление. Пример представлен на рисунке 2.18.

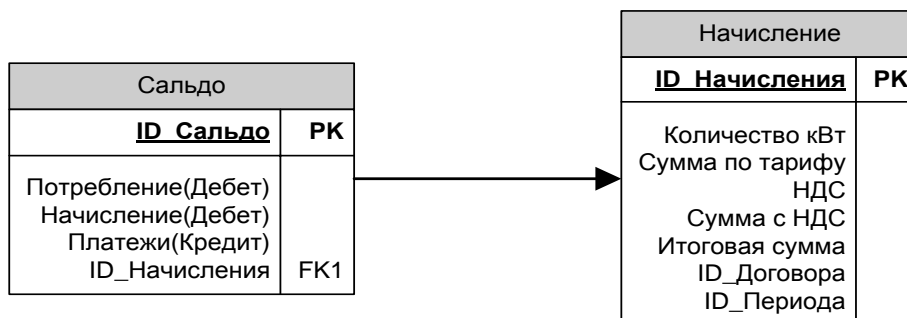


Рисунок 2.18 – Бизнес правило 16

Бизнес-правила достаточно часто порождают вопросы, как в данном случае, которые могут стать причиной появления новых сущностей, атрибутов и связей.

После всех преобразований, изменений и добавлений имеем окончательный вариант подробной ER-диаграммы модуля базы данных биллинг. Пример приведен на рисунке 2.19.

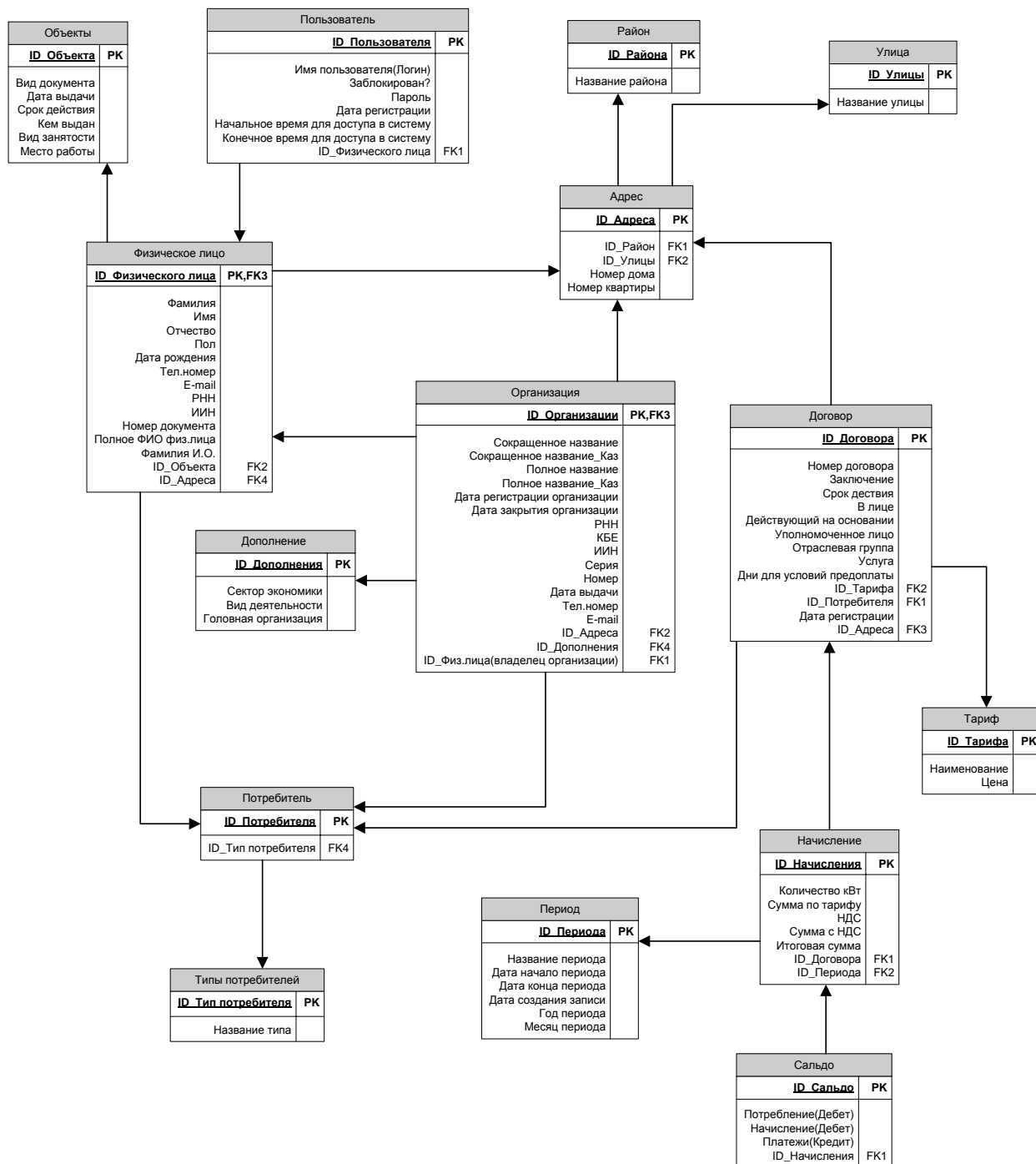


Рисунок 2.19 – Подробная ER-диаграмма

2.5 Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности и в него вносятся все атрибуты сущности. Для каждого отношения необходимо определить первичный ключ и внешние ключи (если они имеются).

Для каждого атрибута необходимо выбрать оптимальный размер, так как если предусмотреть заведомо избыточное число, то получится много пустых значений, а если выделить слишком мало места, то на резервирование будет накладываться дополнительное ограничение. И если в последствии нужно будет разрешить дополнительное резервирование, то необходимо будет менять структуру таблицы. Таким образом, для каждого поля необходимо выделить оптимальный размер.

Для каждого соотношения указаны атрибуты с их внутренним названием, типом и длиной.

Обязательное поле обозначено not null, необязательное null. Типы данных обозначаются: bigint, smallint, integer – числовой, char, varchar – символьный, date – дата, datetime – дата и время (имеет стандартную длину, зависящую от СУБД, поэтому она не указывается).

Т а б л и ц а 2.1 – Схема отношения «Районы» (Districts)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор района	District_ID	integer	PRIMARY KEY
Название района	District_Name	varchar(50)	NULL

Т а б л и ц а 2.2 – Схема отношения «Улицы» (Streets)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор улицы	Street_ID	integer	PRIMARY KEY
Название улицы	Street_Name	varchar(50)	NULL

Т а б л и ц а 2.3 – Схема отношения «Адреса» (Addresses)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор адреса	Address_ID	integer	PRIMARY KEY
Идентификатор района	District_ID	integer	Внешний ключ к (District_ID)
Идентификатор улицы	Street_ID	integer	Внешний ключ к (Street_ID)
Номер дома	House_Number	varchar(20)	NULL
Номер квартиры	Flat_Number	varchar(20)	NULL

Т а б л и ц а 2.4 – Схема отношения «Объекты» (Subjects)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор объекта	Subject_ID	integer	PRIMARY KEY

Окончание таблицы 2.4

Вид документа	DocumentType	varchar(50)	NULL
Дата выдачи	Date_Issued	varchar(20)	NULL
Срок действия	Validity	varchar(12)	NULL
Кем выдан	Issued_by	varchar(20)	NULL
Вид занятости	EmploymentType	varchar(200)	NULL
Место работы	Work_Organization	varchar(200)	NULL

Т а б л и ц а 2.5 – Схема отношения «Типы потребителей» (CustomerTypes)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор типа потребителя	CustomerType_ID	integer	PRIMARY KEY
Название потребителя	CustomerType_Name	varchar(20)	NULL

Т а б л и ц а 2.6 – Схема отношения «Потребители» (Customers)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор абонента	Customer_ID	integer	PRIMARY KEY
Идентификатор типа потребителя	CustomerType_ID	integer	Внешний ключ к (CustomerType_ID)

Т а б л и ц а 2.7 – Схема отношения «Физические лица» (Persons)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор физ.лица	Person_ID	integer	PRIMARY KEY
Фамилия	First_Name	varchar(50)	NULL
Имя	Last_Name	varchar(50)	NULL
Отчество	Patronymic_Name	varchar(50)	NULL
Пол	Sex	varchar(8)	NULL
Дата рождения	Birthday	date	NULL
Тел.номер	Phone_Num	varchar(20)	NULL
E-mail	E_Mail	varchar(20)	NULL
РНН	RNN	char(12)	NULL
ИИН	ID_Num	char(12)	NULL
Номер документа	NumberDoc	char(12)	NULL
Идентификатор объекта	Subject_ID	integer	Внешний ключ к (Subject_ID)
Идентификатор адреса	Address_ID	integer	Внешний ключ к (Address_ID)
Дата регистрации	Date_Registration	smalldatetime	NULL
Полное ФИО физ. лица	Full_Name	varchar(120)	NULL
Фамилия И.О.	Short_Name	varchar(50)	NULL

Т а б л и ц а 2.8 – Схема отношения «Пользователи» (Users)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор пользователя	User_ID	integer	PRIMARY KEY
Имя пользователя (Логин)	User_Name	varchar(50)	NOT NULL
Заблокирован?	IsLockedOut	bit	NOT NULL
Пароль	Password	varchar(15)	NOT NULL
Дата регистрации	Date_Create	smalldatetime	NOT NULL
Начальное время для доступа в систему	Start_Access_Time	time(7)	NOT NULL
Конечное время для доступа в систему	End_Access_Time	time(7)	NOT NULL
Идентификатор физ.лица	Person_ID	integer	Внешний ключ к (Person_ID)

Т а б л и ц а 2.9 – Схема отношения «Дополнение к организации» (Organizations)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор дополнения	Additional_ID	integer	PRIMARY KEY
Сектор экономики	Economic_sector	varchar(100)	NULL
Вид деятельности	Activities	varchar(100)	NULL
Головная организация	Parent_organization	varchar(100)	NULL

Т а б л и ц а 2.10 – Схема отношения «Организации» (Organizations)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор организации	ID_Organization	integer	PRIMARY KEY
Сокращенное название организации	Organization_ShortName	varchar(200)	NOT NULL
Полное название организации	Organization_Name	varchar(500)	NULL
РНН	RNN	char(12)	NULL
ИИН	IIN	char(12)	NULL
КБЕ	KBE	char(2)	NULL
Серия	Seriya	char(3)	NULL
Номер	Number	char(7)	NULL
Дата выдачи	Date_Issued	date	NULL
Телефонный номер	Tel_Num	char(20)	NULL
E-mail	E_mail	varchar(20)	NULL
Владелец организации	Owner_ID	integer	Внешний ключ к (Person_ID)
Идентификатор адреса	Address_ID	integer	Внешний ключ к (Address_ID)
Дата регистрации	Date_Registration	smalldatetime	NOT NULL

Окончание таблицы 2.10

Идентификатор дополнения	Additional_ID	integer	Внешний ключ к (Additional_ID)
--------------------------	---------------	---------	--------------------------------

Т а б л и ц а 2.11 – Схема отношения «Тарифы» (Tariffs)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор тарифа	Tariff_ID	integer	PRIMARY KEY
Наименование тарифа	Tariff_Name	varchar(20)	NOT NULL
Цена тарифа	Tariff_Price	char(15)	NOT NULL

Т а б л и ц а 2.12 – Схема отношения «Договора» (Agreements)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор договора	Agreement_ID	integer	PRIMARY KEY
Номер договора	Agreement_Num	char(6)	NULL
Заключение	Conclusion	date	NULL
Срок действия	Validity	date	NULL
В лице	Person	varchar(60)	NULL
Действующий на основании	Acting	varchar(10)	NULL
Уполномоченное лицо	Authorized	varchar(50)	NULL
Отраслевая группа	Industry_Group	varchar(25)	NULL
Услуга	Service_a	varchar(15)	NULL
День предоплаты	Prepayment_Day	tinyint	NULL
Идентификатор тарифа	Tariff_ID	integer	Внешний ключ к (Tariff_ID)
Идентификатор потребителя	Customer_ID	integer	Внешний ключ к (Customer_ID)
Идентификатор адреса	Address_ID	integer	Внешний ключ к (Address_ID)
Дата регистрации	Date_Registration	smalldatetime	NOT NULL

Т а б л и ц а 2.13 – Схема отношения «Периоды» (Periods)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор периода	Period_ID	smallint	PRIMARY KEY
Название периода	Period_Name	varchar(20)	NOT NULL
Дата начало периода	Date_Begin	date	NULL
Дата создания записи	Date_Create	date	NOT NULL
Год периода	Period_Year	smallint	NOT NULL
Месяц периода	Period_Month	tinyint	NOT NULL
Дата конца периода	Date_End	date	NULL

Т а б л и ц а 2.14 – Схема отношения «Начисления» (Accruals)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор начисления	Accrual_ID	integer	PRIMARY KEY
Дата начало периода	Date_Begin	date	NOT NULL
Дата конца периода	Date_End	date	NOT NULL
Количество кВт	Kilowatt	integer	NULL
Сумма по тарифу	Sum_With_Tariff	integer	NULL
НДС	VAT	integer	NULL
Сумма с НДС	Sum_With_VAT	integer	NULL
Итоговая сумма	Amount	integer	NULL
Идентификатор договора	Agreement_ID	integer	Внешний ключ к (Agreement_ID)
Идентификатор периода	Period_ID	smallint	Внешний ключ к (Period_ID)

Т а б л и ц а 2.15 – Схема отношения «Сальдо» (DealBalances)

Содержание поля	Имя поля	Тип, длина	Примечание
Идентификатор сальдо	DealBalance_ID	integer	PRIMARY KEY
Идентификатор периода	Period_ID	smallint	NULL
Потребление (Дебет)	Consumption_Value	integer	NULL
Начисление (Дебет)	Debit_Value	integer	NULL
Платежи (Кредит)	Credit_Value	integer	NULL
Идентификатор начисления	Accrual_ID	integer	Внешний ключ к (Accrual_ID)

2.6 Расчет размера базы данных

На данном этапе проектирования необходимо узнать, какой объем будет занимать база данных.

Объём внешней памяти, необходимый для функционирования системы, складывается из двух составляющих: память, занимаемая модулями СУБД (ядро, утилиты, вспомогательные программы), и память, отводимая под данные (M_d). Наиболее существенным обычно является M_d . Объём памяти M_d , требуемый для хранения данных, можно приблизительно оценить по формуле

$$M_c = 2 \sum_{i=1}^n L_i \cdot (N_i + N_{ai}) \quad (2.1)$$

Объём памяти, занимаемый программными модулями пользователя, обычно невелик по сравнению с объёмом самих данных, поэтому может не учитываться.

Т а б л и ц а 2.16 – Расчет физической памяти для таблицы «Районы» (Districts)

Содержание поля	Тип, длина	Длина
District_ID	integer	4
District_Name	varchar(50)	50

Расчет размера таблицы «Районы» (Districts)

$$L_i=4+50=54$$

$$54*7=378$$

Т а б л и ц а 2.17 – Расчет физической памяти для таблицы «Улицы» (Streets)

Содержание поля	Тип, длина	Длина
Street_ID	integer	4
Street_Name	varchar(50)	50

Расчет размера таблицы «Улицы» (Streets)

$$L_i=4+50=54$$

$$54*100=540$$

Т а б л и ц а 2.18 – Расчет физической памяти для таблицы «Адреса» (Addresses)

Содержание поля	Тип, длина	Длина
Address_ID	integer	4
District_ID	integer	4
Street_ID	integer	4
House_Number	varchar(20)	20
Flat_Number	varchar(20)	20

Расчет размера таблицы «Адреса» (Addresses)

$$L_i=4+4+4+20+20=52$$

$$52*100=520$$

Т а б л и ц а 2.19 – Расчет физической памяти для таблицы «Объекты» (Subjects)

Содержание поля	Тип, длина	Длина
Subject_ID	integer	4
DocumentType	varchar(50)	50
Date_Issued	varchar(20)	20
Validity	varchar(12)	12
Issued_by	varchar(20)	20
EmploymentType	varchar(200)	200
Work_Organization	varchar(200)	200

Расчет размера таблицы «Объекты» (Subjects)

$$L_i=4+50+20+12+20+200+200=506$$

$$506*50= 25300$$

Т а б л и ц а 2.20 – Расчет физической памяти для таблицы «Типы потребителей» (CustomerTypes)

Содержание поля	Тип, длина	Длина
CustomerType_ID	integer	4
CustomerType_Name	varchar(20)	20

Расчет размера таблицы «Типы потребителей» (CustomerTypes)

$$L_i=4+20=24$$

$$24*2=48$$

Т а б л и ц а 2.21 – Расчет физической памяти для таблицы «Потребители» (Customers)

Содержание поля	Тип, длина	Длина
Customer_ID	integer	4
CustomerType_ID	integer	4

Расчет размера таблицы «Потребители» (Customers)

$$L_i=4+4=8$$

$$8*100=800$$

Т а б л и ц а 2.22 – Расчет физической памяти для таблицы «Физические лица» (Persons)

Содержание поля	Тип, длина	Длина
Идентификатор физ.лица	integer	4
Фамилия	varchar(50)	50
Имя	varchar(50)	50
Отчество	varchar(50)	50
Пол	varchar(8)	8
Дата рождения	date	3
Тел.номер	varchar(20)	20
E-mail	varchar(20)	20
РНН	char(12)	12
ИИН	char(12)	12
Номер документа	char(12)	12
Идентификатор объекта	integer	4
Идентификатор адреса	integer	4
Дата регистрации	smalldatetime	4
Полное ФИО физ. лица	varchar(120)	120
Фамилия И.О.	varchar(50)	50

Расчет размера таблицы «Физические лица» (Persons)

$$L_i=4+50+50+50+8+3+20+20+12+12+12+4+4+4+120+50=423$$

$$423*50=21150$$

Т а б л и ц а 2.23 – Расчет физической памяти для таблицы «Пользователи» (Users)

Содержание поля	Тип, длина	Длина
User_ID	integer	4
User Name	varchar(50)	50
Password	varchar(15)	15
Date_Create	smalldatetime	4
Start_Access_Time	time(7)	7
End_Access_Time	time(7)	7
Person_ID	integer	4

Расчет размера таблицы «Пользователи» (Users)

$$L_i=4+50+15+4+7+7+4=91$$

$$91*10=910$$

Т а б л и ц а 2.24 – Расчет физической памяти для таблицы «Дополнение к организации» (Additional)

Содержание поля	Тип, длина	Примечание
Идентификатор дополнения	integer	4
Сектор экономики	varchar(100)	100
Вид деятельности	varchar(100)	100
Головная организация	varchar(100)	100

Расчет размера таблицы «Дополнение к организации» (Additional)

$$L_i=4+100+100+100=304$$

$$304*50=15200$$

Т а б л и ц а 2.25 – Расчет физической памяти для таблицы «Организации» (Organizations)

Содержание поля	Тип, длина	Длина
Идентификатор организации	integer	4
Сокращенное название организации	varchar(200)	200
Полное название организации	varchar(500)	500
РНН	char(12)	12
ИИН	char(12)	12
КБЕ	char(2)	2
Серия	char(3)	4
Номер	char(7)	7
Дата выдачи	date	3
Телефонный номер	char(20)	20
E-mail	varchar(20)	20
Владелец организации	integer	4
Дата регистрации	smalldatetime	4
Идентификатор адреса	integer	4

Окончание таблицы 2.25

Идентификатор дополнения	integer	4
--------------------------	---------	---

Расчет размера таблицы «Организации» (Organizations)

$$L_i=4+200+500+12+12+2+4+7+3+20+20+4+4+4+4=792$$

$$792*50=39600$$

Т а б л и ц а 2.26 – Расчет физической памяти для таблицы «Тарифы» (Tariffs)

Содержание поля	Тип, длина	Длина
Tariff_ID	integer	4
Tariff_Name	varchar(20)	20
Tariff_Price	char(7)	7

Расчет размера таблицы «Тарифы» (Tariffs)

$$L_i=4+20+7=31$$

$$31*3=93$$

Т а б л и ц а 2.27 – Расчет физической памяти для таблицы «Договора» (Agreements)

Содержание поля	Тип, длина	Длина
Идентификатор договора	integer	4
Номер договора	char(6)	6
Заключение	date	3
Срок действия	date	3
В лице	varchar(60)	60
Действующий на основании	varchar(10)	10
Уполномоченное лицо	varchar(50)	50
Отраслевая группа	varchar(25)	25
Услуга	varchar(15)	15
Идентификатор тарифа	integer	4
Идентификатор потребителя	integer	4
Идентификатор адреса	integer	4
Дата регистрации	smalldatetime	4

Расчет размера таблицы «Договора» (Agreements)

$$L_i=4+6+3+3+60+10+50+25+15+4+4+4+4=192$$

$$192*100=19200$$

Т а б л и ц а 2.28 – Расчет физической памяти для таблицы «Периоды» (Periods)

Содержание поля	Тип, длина	Длина
Period_ID	smallint	2
Period_Name	varchar(20)	20
Date_Begin	date	3
Date_Create	date	3
Period_Year	smallint	2

Окончание таблицы 2.28

Period_Month	tinyint	1
Date_End	date	3

Расчет размера таблицы «Периоды» (Periods)

$$L_i = 2 + 20 + 3 + 3 + 2 + 1 + 3 = 44$$

$$44 * 12 = 528$$

Т а б л и ц а 2.29 – Расчет физической памяти для таблицы «Начисления» (Accruals)

Содержание поля	Тип, длина	Длина
Accrual_ID	integer	4
Date_Begin	date	3
Date_End	date	3
Value_Kilowatt	integer	4
Sum_With_Tariff	integer	4
VAT	integer	4
Sum_With_VAT	integer	4
Amount	integer	4
Agreement_ID	integer	4
Period_ID	integer	4

Расчет размера таблицы «Начисления» (Accruals)

$$L_i = 4 + 3 + 3 + 4 + 4 + 4 + 4 + 4 + 4 = 38$$

$$38 * 100 = 3800$$

Т а б л и ц а 2.30 – Расчет физической памяти для таблицы «Сальдо» (DealBalances)

Содержание поля	Тип, длина	Длина
DealBalance_ID	integer	4
Consumption_Value	integer	4
Debit_Value	integer	4
Credit_Value	integer	4
Accrual_ID	integer	4

Расчет размера таблицы «Сальдо» (DealBalances)

$$L_i = 4 + 4 + 4 + 4 + 4 = 20$$

$$20 * 100 = 2000$$

Согласно формуле (2.1), объем памяти составляет:

$$M_c = 2 \cdot (378 + 540 + 520 + 25300 + 48 + 800 + 21150 + 910 + 15200 + 39600 + 93 + 20400 + 528 + 3800 + 2000) = 2 \cdot 131267 = 262534 \text{ байт} \approx 256,4 \text{ Кбайт.}$$

2.7 Меры по обеспечению безопасности

Создание пользователей, логинов и задание им паролей

Одной из важнейших составляющих проекта базы данных является разработка средств защиты БД. Защита данных имеет два аспекта: защита от сбоев и защита от несанкционированного доступа. Для защиты от сбоев разрабатывается стратегия резервного копирования. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа. Для данной базы данных создаются 6 учетные записи:

- администратор;
- пользователь «nastya»;
- пользователь «rus»;
- пользователь «bonmax»;
- пользователь «ali»;
- пользователь «sergey».

Администратору дается полный контроль над БД, права пользователям вносятся разрешение на изменение и обновление определенных таблиц, а также пользователи имеют право просматривать данные.

```
--Создание учетной записи
EXEC SP_ADDLOGIN [admin], adminadmin, DIP_BILLING
EXEC SP_ADDLOGIN nastya, useruser, DIP_BILLING
EXEC SP_ADDLOGIN rus, useruser, DIP_BILLING
EXEC SP_ADDLOGIN bonmax, useruser, DIP_BILLING
EXEC SP_ADDLOGIN sergey, useruser, DIP_BILLING
EXEC SP_ADDLOGIN ali, useruser, DIP_BILLING

--Создание роли
EXEC SP_ADDUSER 'admin';
EXEC SP_ADDUSER 'nastya';
EXEC SP_ADDUSER 'rus';
EXEC SP_ADDUSER 'bonmax';
EXEC SP_ADDUSER 'ali';
EXEC SP_ADDUSER 'sergey';

--Создание роли
EXEC SP_ADDROLE superuser;
EXEC SP_ADDROLE users;

--Роль пользователям
EXEC SP_ADDROLEMEMBER superuser, [admin]

--Роль пользователям
EXEC SP_ADDROLEMEMBER users, nastya;
EXEC SP_ADDROLEMEMBER users, rus;
EXEC SP_ADDROLEMEMBER users, bonmax;
EXEC SP_ADDROLEMEMBER users, sergey;
EXEC SP_ADDROLEMEMBER users, ali;

--Предоставление прав администратору
GRANT SELECT, UPDATE, INSERT, DELETE ON Accruals TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON Addresses TO [admin];
```

```

GRANT SELECT, UPDATE, INSERT, DELETE ON Agreements TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON DealBalances TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON Organizations TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON Persons TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON Users TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON Customers TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON Additionalals TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON CustomerTypes TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON Districts TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON Streets TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON Periods TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON Subjects TO [admin];
GRANT SELECT, UPDATE, INSERT, DELETE ON Tariffs TO [admin];

--Предоставление пользователям возможность просматривать данные
GRANT SELECT ON Accruals TO nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON Addresses TO nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON Agreements TO nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON DealBalances TO nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON Organizations TO nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON Persons TO nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON Users TO nastya,rus,bonmax,sergey,ali;

--Предоставление пользователям возможность просматривать данные
GRANT SELECT ON VW_Accruals TO [admin],nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON VW_Addresses TO [admin],nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON VW_DealBalances TO [admin],nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON VW_Organizations TO [admin],nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON VW_Persons TO [admin],nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON VW_Users TO [admin],nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON VW_Periods TO [admin],nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON VW_Subjects TO [admin],nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON VW_Agreements TO [admin],nastya,rus,bonmax,sergey,ali;
GRANT SELECT ON VW_Users TO [admin],nastya,rus,bonmax,sergey,ali;

--Предоставление пользователям возможность вставлять и вносить изменения
в данных
GRANT UPDATE, INSERT ON Accruals TO nastya,rus,bonmax,sergey,ali;
GRANT UPDATE, INSERT ON Addresses TO nastya,rus,bonmax,sergey,ali;
GRANT UPDATE, INSERT ON Agreements TO nastya,rus,bonmax,sergey,ali;
GRANT UPDATE, INSERT ON DealBalances TO nastya,rus,bonmax,sergey,ali;
GRANT UPDATE, INSERT ON Organizations TO nastya,rus,bonmax,sergey,ali;
GRANT UPDATE, INSERT ON Persons TO nastya,rus,bonmax,sergey,ali;
GRANT UPDATE, INSERT ON Districts TO nastya,rus,bonmax,sergey,ali;
GRANT UPDATE, INSERT ON Streets TO nastya,rus,bonmax,sergey,ali;

--Запрет пользователям на удаление
DENY Delete ON Addresses TO rus,ali;
DENY Delete ON Organizations TO rus,bonmax,ali;
DENY Delete ON Users TO nastya,rus,bonmax,sergey,ali;

```

2.8 Логическое проектирование

На этапе логического проектирования концептуальная модель базы данных трансформируется в формат, пригодный для конкретной СУБД (в

данном случае Microsoft SQL Server 2008 R2), которая будет использоваться при реализации системы. Задача логического проектирования состоит в создании реляционных табличных структур. Структуры таблиц и связей в проектируемой базе создаем с помощью Microsoft SQL Server 2008 R2.

2.8.1 Задание места хранения БД

```
CREATE DATABASE DIP_BILLING ON PRIMARY (NAME = DIP_BILLING, FILENAME =  
'D:\BILLING\DIP_BILLING.mdf', size=4,maxsize=30, filegrowth=1 mb)  
LOG ON(NAME = DIP_BILLING_log, FILENAME =  
'D:\BILLING\DIP_BILLING.ldf', size=4,maxsize=25, filegrowth=1 mb)
```

2.8.2 Создание таблиц

```
--Создание сущности "DISTRICTS"  
CREATE TABLE Districts (  
    District_ID    integer IDENTITY(1,1) NOT NULL PRIMARY KEY,  
    District_Name  varchar(50) null)  
  
--Создание сущности "STREETS"  
CREATE TABLE Streets (  
    Street_ID     integer IDENTITY(1,1) NOT NULL PRIMARY KEY,  
    Street_Name   varchar(50) NULL,  
    District_ID   integer NULL FOREIGN KEY REFERENCES  
Districts(District_ID))  
  
--Создание сущности "ADDRESSES"  
CREATE TABLE Addresses (  
    Address_ID    integer IDENTITY(1,1) NOT NULL PRIMARY KEY,  
    House_Number  varchar(20) NULL,  
    Flat_Number   varchar(20) NULL,  
    District_ID   integer NULL FOREIGN KEY REFERENCES  
Districts(District_ID),  
    Street_ID     integer NULL FOREIGN KEY REFERENCES Streets(Street_ID))  
  
--Создание сущности "SUBJECTS"  
CREATE TABLE Subjects (  
    Subject_ID integer IDENTITY(1,1) NOT NULL PRIMARY KEY,  
    DocumentType    varchar(50) NULL,  
    Date_Issued     varchar(20) NULL,  
    Validity        varchar(12) NULL,  
    Issued_by       varchar(40) NULL,  
    EmploymentType  varchar(100) NULL,  
    Work_Organization varchar(200) NULL)  
  
--Создание сущности "CUSTOMER TYPES"  
CREATE TABLE CustomerTypes(  
    CustomerType_ID integer NOT NULL PRIMARY KEY,  
    CustomerType_Name varchar(20) NULL)  
  
--Создание сущности "CUSTOMERS"  
CREATE TABLE Customers(  
    Customer_ID    integer NOT NULL PRIMARY KEY,
```

```

CustomerType_ID integer NOT NULL FOREIGN KEY REFERENCES
CustomerTypes(CustomerType_ID))

--Создание сущности "PERSONS"
CREATE TABLE Persons (
    Person_ID integer NOT NULL PRIMARY KEY,
    First_Name varchar(50) NULL,
    Last_Name varchar(50) NULL,
    Patronymic_Name varchar(50) NULL,
    Sex varchar(8) NULL,
    Birthday date NULL,
    Phone_Num varchar(20) NULL,
    E-Mail varchar(20) NULL,
    RNN char(12) NULL,
    ID_Num char(12) NULL,
    NumberDoc char(12) NULL,
    Subject_ID integer NULL FOREIGN KEY REFERENCES Subjects(Subject_ID),
    Address_ID integer NULL FOREIGN KEY REFERENCES Addresses(Address_ID),
    Date_Registration smalldatetime NOT NULL,
    Full_Name AS
    (((isnull([first_name], '')+space((1)))+(isnull([last_name], '')+space((1))
    ))+isnull([patronymic_name], '')) PERSISTED,
    Short_Name AS ((First_Name+case when [Last_Name] IS NOT NULL then ('
    '+left([first_name], (1))+'.' else '' end)+case when [patronymic_name] IS
    NOT NULL then (' '+left([Patronymic_Name], (1))+'.' else '' end)
    PERSISTED)
ALTER TABLE Persons WITH CHECK ADD CONSTRAINT CK_PERSONS CHECK
    (([Sex]='Женский' OR [Sex]='Мужской' OR [Sex] IS NULL)) GO
ALTER TABLE Persons ADD CONSTRAINT DF_Persons_Date_Registration DEFAULT
    (getdate()) FOR Date_Registration GO
ALTER TABLE [dbo].[Persons] WITH CHECK ADD CONSTRAINT
    [FK_Persons_Customers] FOREIGN KEY([Person_ID])REFERENCES
    [dbo].[Customers] ([Customer_ID]) GO

--Создание сущности "USERS"
CREATE TABLE Users (
    User_ID integer IDENTITY(1,1) NOT NULL PRIMARY KEY,
    User_Name varchar(15) NOT NULL,
    Passwordd varchar(15) NOT NULL,
    Date_Create smalldatetime NOT NULL,
    Start_Access_Time time(7) NOT NULL,
    End_Access_Time time(7) NOT NULL,
    Person_ID integer NULL FOREIGN KEY REFERENCES Persons (Person_ID))
ALTER TABLE Users ADD CONSTRAINT DF_Users_Passwordd DEFAULT ((1)) FOR
    Passwordd
ALTER TABLE Users ADD CONSTRAINT DF_Users_Date_Create DEFAULT (getdate())
    FOR Date_Create
ALTER TABLE Users ADD CONSTRAINT DF_Users_Start_Access_Time DEFAULT
    ('00:00') FOR Start_Access_Time
ALTER TABLE Users ADD CONSTRAINT DF_Users_End_Access_Time DEFAULT
    ('23:59') FOR End_Access_Time

--Создание сущности "ADDITIONALS"
CREATE TABLE Additional (
    Additional_ID integer IDENTITY(1,1) NOT NULL PRIMARY KEY,
    Economic_sector varchar(100) NULL,

```

```

Activities          varchar(100) NULL,
Parent_organization varchar(100) NULL)

```

--Создание сущности "ORGANIZATIONS"

```

CREATE TABLE Organizations (
    Organization_ID      integer NOT NULL PRIMARY KEY,
    Organization_ShortName varchar(200) NOT NULL,
    Organization_Name    varchar(500) NULL,
    RNN                  char(12) NULL,
    IIN                  char(12) NULL,
    KBE                  char(2) NULL,
    Seriya               char(3) NULL,
    Number               char(7) NULL,
    Date_Issued          date NULL,
    Tel_Num              char(20) NULL,
    E_mail               varchar(20) NULL,
    Owner_ID integer NULL FOREIGN KEY REFERENCES Persons(Person_ID),
    Date_Registration    smalldatetime NULL,
    Address_ID integer NULL FOREIGN KEY REFERENCES Addresses(Address_ID),
    Additional_ID integer NULL FOREIGN KEY REFERENCES
        Additional_Extras(Additional_ID))
ALTER TABLE Organizations ADD CONSTRAINT
DF_Organizations_Date_Registration DEFAULT (getdate()) FOR
Date_Registration
ALTER TABLE [dbo].[Organizations] WITH CHECK ADD CONSTRAINT
[FK_Organizations_Customers] FOREIGN KEY([Organization_ID])
REFERENCES [dbo].[Customers] ([Customer_ID])GO

```

--Создание сущности "TARIFFS"

```

CREATE TABLE Tariffs (
    Tariff_ID      integer IDENTITY(1,1) NOT NULL PRIMARY KEY,
    Tariff_Name    varchar(20) NOT NULL,
    Tariff_Price  char(7) NOT NULL)

```

--Создание сущности "AGREEMENTS"

```

CREATE TABLE Agreements (
    Agreement_ID      integer IDENTITY(401,1) NOT NULL PRIMARY KEY,
    Agreement_Num     char(6) NULL,
    Conclusion        date NULL,
    Validity          date NULL,
    Person            varchar(25) NULL,
    Acting            varchar(10) NULL,
    Authorized        varchar(50) NULL,
    Industry_Group    varchar(25) NULL,
    Service_a         varchar(15) NULL,
    Tariff_ID integer NULL FOREIGN KEY REFERENCES Tariffs(Tariff_ID),
    Customer_ID integer NULL FOREIGN KEY REFERENCES
Customers(Customer_ID),
    Address_ID integer NULL FOREIGN KEY REFERENCES Addresses(Address_ID),
    Date_Registration smalldatetime NOT NULL)
ALTER TABLE Agreements ADD CONSTRAINT DF_Agreements_Date_Registration
DEFAULT (getdate()) FOR Date_Registration

```

--Создание сущности "PERIODS"

```

CREATE TABLE Periods(
    Period_ID      integer IDENTITY(1,1) NOT NULL PRIMARY KEY,

```



```

        Period_Name      varchar(20) NOT NULL,
        Date_Begin       date          NOT NULL,
        Period_Year      smallint     NOT NULL,
        Period_Month     tinyint      NOT NULL,
        Period_Days_Count AS (datediff(day, dateadd(month, [Period_Month]-
(1), dateadd(year, [Period_Year]-
datepart(year, (0)), (0))), dateadd(month, [Period_Month], dateadd(year, [Period_
d_Year]-datepart(year, (0)), (0)))) PERSISTED,
        Date_End        AS
(dateadd(month, [Period_Month], dateadd(year, [Period_Year]-
datepart(year, (0)), (-1)))) PERSISTED)
ALTER TABLE [dbo].[Periods] ADD CONSTRAINT
[DF__Periods__Date_Cr__4F67C174] DEFAULT (getdate()) FOR [Date_Create]
GO
ALTER TABLE [dbo].[Periods] ADD CONSTRAINT [DF_Periods_Year] DEFAULT
(datepart(year, getdate())) FOR [Period_Year] GO
ALTER TABLE [dbo].[Periods] ADD CONSTRAINT [DF_Periods_Month] DEFAULT
(datepart(month, getdate())) FOR [Period_Month] GO

```

--Создание сущности "ACCRUALS"

```

CREATE TABLE Accruals (
    Accrual_ID      integer IDENTITY(201,1) NOT NULL PRIMARY KEY,
    Date_Begin      date          NOT NULL,
    Date_End        date          NOT NULL,
    Value_Kilowatt  integer       NULL,
    Sum_With_Tariff integer       NULL,
    VAT             integer       NULL,
    Sum_With_VAT AS ([Sum_With_Tariff]*0.12) PERSISTED,
    Amount          integer       NULL,
    Period_ID       integer       NULL FOREIGN KEY REFERENCES Periods(Period_ID),
    Agreement_ID    integer       NULL FOREIGN KEY REFERENCES
Agreements(Agreement_ID))

```

--Создание сущности "DEAL BALANCES"

```

CREATE TABLE DealBalances(
    DealBalance_ID  integer IDENTITY(1,1) NOT NULL PRIMARY KEY,
    Consumption_Value integer       NULL,
    Debit_Value     integer       NULL,
    Credit_Value    integer       NULL,
    Accrual_ID      integer       NULL FOREIGN KEY REFERENCES
Accruals(Accrual_ID))
ALTER TABLE [dbo].[DealBalances] WITH CHECK ADD CONSTRAINT
[FK_DealBalances_Periods] FOREIGN KEY([Period_ID]) REFERENCES
[dbo].[Periods] ([Period_ID])GO
ALTER TABLE [dbo].[DealBalances] CHECK CONSTRAINT
[FK_DealBalances_Periods]GO
ALTER TABLE [dbo].[DealBalances] ADD CONSTRAINT
[DF_DealBalances_StartBalance_Value] DEFAULT ((0)) FOR
[StartBalance_Value]GO
ALTER TABLE [dbo].[DealBalances] ADD CONSTRAINT
[DF_DealBalances_Consumption_Value] DEFAULT ((0)) FOR
[Consumption_Value] GO
ALTER TABLE [dbo].[DealBalances] ADD CONSTRAINT
[DF_DealBalances_Debit_Value] DEFAULT ((0)) FOR [Debit_Value]GO
ALTER TABLE [dbo].[DealBalances] ADD CONSTRAINT
[DF_DealBalances_Credit_Value] DEFAULT ((0)) FOR [Credit_Value] GO

```

2.8.3 Создание функций

Функция представляет собой именованный блок кода, возвращающий по результатам своего выполнения некоторое значение. Функции могут участвовать в выражениях, вызываться в теле хранимых процедур, а также в теле других функций.

```
--Создание функции "Person_FullName"
CREATE FUNCTION fnGetPersonFullName
(@Person_ID int)
RETURNS nvarchar(150)
AS
BEGIN
DECLARE
    @retValue nvarchar(150)
SELECT
    @retValue = Full_Name
FROM dbo.Persons (nolock)
WHERE Person_ID = @Person_ID
RETURN @retValue
END
GO
```

```
--Создание функции "Person_ShortName"
CREATE FUNCTION fnGetPersonShortName
(@Person_ID int)
RETURNS nvarchar(100)
AS
BEGIN
DECLARE
    @retValue nvarchar(150)
SELECT
    @retValue = Short_Name
FROM dbo.Persons (nolock)
WHERE Person_ID = @Person_ID
RETURN @retValue
END
GO
```

```
--Создание функции "AddressName"
CREATE FUNCTION fnGetAddressName
(@Address_ID int)
RETURNS nvarchar(max)
AS
BEGIN
DECLARE
    @House_Number nvarchar(20),
    @District_Name nvarchar(100),
    @Street_Name nvarchar(100),
    @Flat_Number nvarchar(20),
    @Address_Name nvarchar(max)
SELECT
    @District_Name = d.District_Name,
```

```

        @Street_Name = s.Street_Name,
        @Flat_Number = a.Flat_Number
FROM dbo.Addresses a (nolock)
    join dbo.Districts d (nolock)
ON d.District_ID = a.District_ID
    join dbo.Streets s (nolock)
ON a.Street_ID = s.Street_ID
WHERE a.Address_ID = @Address_ID
SELECT
    @Address_Name = @District_Name
    + ' '
    + ' '
    + @Street_Name
    if(@House_Number is not null)
SET
    @Address_Name = @Address_Name + '-' + @House_Number
if(@Flat_Number is not null)
SET
    @Address_Name = @Address_Name + '-' + @Flat_Number
RETURN @Address_Name
END

```

2.8.4 Создание триггеров

Триггер — это особая разновидность хранимой процедуры, выполняемая автоматически при возникновении события на сервере базы данных.

```

--Триггер на повторение района
CREATE TRIGGER Repeat_District
ON Districts
FOR INSERT, UPDATE
AS
DECLARE @District_Name nvarchar(50)
DECLARE @COUNT INTEGER
SELECT @District_Name = District_Name FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Districts
    WHERE District_Name = @District_Name
    IF @COUNT>1
    BEGIN
        ROLLBACK TRAN
        RAISERROR ('Такой район уже существует в базе данных!',16,10)
    END

```

```

--Триггер на повторение улиц
CREATE TRIGGER Repeat_Street
ON Streets
FOR INSERT, UPDATE
AS
DECLARE @Street_Name nvarchar(50)
DECLARE @COUNT INTEGER
SELECT @Street_Name = Street_Name FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Streets
    WHERE Street_Name = @Street_Name
    IF @COUNT>1
    BEGIN

```

```

ROLLBACK TRAN
RAISERROR ('Такая улица уже существует в базе данных!',16,10)
END

--Триггер на повторения РНН
CREATE TRIGGER Repeat_RNN
ON Persons
FOR INSERT, UPDATE
AS
DECLARE @RNN varchar(12)
DECLARE @COUNT INTEGER
SELECT @RNN = RNN FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Persons
WHERE RNN = @RNN
IF @COUNT>1
BEGIN
ROLLBACK TRAN
RAISERROR ('Такой РНН записан! Введите другой РНН!',16,10)
END

--Триггер на повторения ИИН
CREATE TRIGGER Repeat_IIN
ON Persons
FOR INSERT, UPDATE
AS
DECLARE @ID_Num varchar(12)
DECLARE @COUNT INTEGER
SELECT @ID_Num = ID_Num FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Persons
WHERE ID_Num = @ID_Num
IF @COUNT>1
BEGIN
ROLLBACK TRAN
RAISERROR ('Такой ИИН уже Зарегистрирован!',16,10)
END

--Триггер на повторения Номера
CREATE TRIGGER Repeat_NumericDoc
ON Persons
FOR INSERT, UPDATE
AS
DECLARE @NumberDoc varchar(12)
DECLARE @COUNT INTEGER
SELECT @NumberDoc = NumberDoc FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Persons
WHERE NumberDoc = @NumberDoc
IF @COUNT>1
BEGIN
ROLLBACK TRAN
RAISERROR ('Такой Номер уже Зарегистрирован!',16,10)
END

--Триггер для логина
CREATE TRIGGER Repeat_User_Login
ON Users
FOR INSERT, UPDATE

```

```

AS
DECLARE @User_Name varchar(50)
DECLARE @COUNT INTEGER
SELECT @User_Name = User_Name FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Users
    WHERE User_Name = @User_Name
    IF @COUNT>1
    BEGIN
    ROLLBACK TRAN
    RAISERROR ('Такой логин уже существует в базе данных!',16,10)
    END

--Триггер для пароля
CREATE TRIGGER NEW_Passwordd
ON Users
FOR INSERT, UPDATE
AS
DECLARE @Passwordd varchar(15)
DECLARE @COUNT INTEGER
SELECT @Passwordd=Passwordd FROM INSERTED
SELECT @COUNT=COUNT(*) FROM Users
    WHERE Passwordd=@Passwordd
    IF @COUNT>1
    BEGIN
    ROLLBACK TRAN
    RAISERROR ('Введите другой пароль!',16,10)
    END

--Триггер для пользователя
CREATE TRIGGER Repeat_User_Person
ON Users
FOR INSERT, UPDATE
AS
DECLARE @Person_ID integer
DECLARE @COUNT INTEGER
SELECT @Person_ID = Person_ID FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Users
    WHERE Person_ID = @Person_ID
    IF @COUNT>1
    BEGIN
    ROLLBACK TRAN
    RAISERROR ('Данный пользователь уже имеет логин!',16,10)
    END

--Триггер на повторения организации
CREATE TRIGGER Repeat_Organization
ON Organizations
FOR INSERT, UPDATE
AS
DECLARE @Organization_ShortName nvarchar(200)
DECLARE @COUNT INTEGER
SELECT @Organization_ShortName = Organization_ShortName FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Organizations
    WHERE Organization_ShortName = @Organization_ShortName
    IF @COUNT>1
    BEGIN

```

```

ROLLBACK TRAN
RAISERROR ('Такая организация уже существует в базе данных!',16,10)
END

--Триггер на повторения РНН
CREATE TRIGGER Repeat_Organization_RNN
ON Organizations
FOR INSERT, UPDATE
AS
DECLARE @RNN varchar(12)
DECLARE @COUNT INTEGER
SELECT @RNN = RNN FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Organizations
WHERE RNN = @RNN
IF @COUNT>1
BEGIN
ROLLBACK TRAN
RAISERROR ('Такой РНН существует в базе данных!',16,10)
END

--Триггер на повторения ИИН
CREATE TRIGGER Repeat_Organization_IIN
ON Organizations
FOR INSERT, UPDATE
AS
DECLARE @IIN varchar(12)
DECLARE @COUNT INTEGER
SELECT @IIN = IIN FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Organizations
WHERE IIN = @IIN
IF @COUNT>1
BEGIN
ROLLBACK TRAN
RAISERROR ('Такой ИИН существует!',16,10)
END

--Триггер на повторения Номера
CREATE TRIGGER Repeat_Organization_Number
ON Organizations
FOR INSERT, UPDATE
AS
DECLARE @Number char(7)
DECLARE @COUNT INTEGER
SELECT @Number = Number FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Organizations
WHERE Number = @Number
IF @COUNT>1
BEGIN
ROLLBACK TRAN
RAISERROR ('Такой Номер уже существует в базе данных!',16,10)
END

--Триггер на повторения договора
CREATE TRIGGER Repeat_Agreement
ON Agreements
FOR INSERT, UPDATE

```

```

AS
DECLARE @Agreement_Num char(6)
DECLARE @COUNT INTEGER
SELECT @Agreement_Num = Agreement_Num FROM INSERTED
SELECT @COUNT = COUNT(*) FROM Agreements
WHERE Agreement_Num = @Agreement_Num
IF @COUNT>1
BEGIN
ROLLBACK TRAN
RAISERROR ('Такой номер договора уже Зарегистрирован!',16,10)
END

--Триггер "сумма за электроэнергию по тарифу"
CREATE TRIGGER TR_Sum_Tarrif
ON Accruals
AFTER
INSERT
AS
BEGIN
UPDATE Accruals
SET Sum_With_Tariff = Value_Kilowatt*12.95
FROM Accruals
END

--Триггер "итоговая сумма"
CREATE TRIGGER TR_Amount
ON Accruals
AFTER
INSERT
AS
BEGIN
UPDATE Accruals
SET Amount = Sum_With_Tariff+Is_With_VAT
FROM Accruals
END

--Триггер для ввода
CREATE TRIGGER Accruals_DealBalances
ON Accruals
AFTER
INSERT, UPDATE
AS
BEGIN
SET NOCOUNT ON;
INSERT INTO
    dbo.DealBalances
    (
        Consumption_Value,
        Debit_Value
    )
SELECT Value_Kilowatt,
        Amount
FROM inserted
END

```

2.8.5 Создание процедур

Хранимые процедуры – это набор операторов SQL, созданный для удобства использования в программах. Сохраненную процедуру использовать проще, чем каждый раз записывать весь набор входящих в нее операторов SQL. Сохраненные процедуры можно вкладывать одну в другую (уровень вложенности не может превышать 16).

```
--Процедура на удаление Адреса
CREATE PROCEDURE pAddresses_Delete
    @Address_ID int
AS
BEGIN
    begin try
        DELETE FROM Addresses
        WHERE Address_ID = @Address_ID
    end try
    begin catch
        return -1
    end catch
END

--Процедура для адреса
CREATE PROCEDURE pAddresses_GetByID
    @Address_ID int
AS
BEGIN
SELECT
    Address_ID
    ,House_Number
    ,Flat_Number
FROM dbo.vwAddresses (nolock)
WHERE Address_ID = @Address_ID
END

--Процедура на удаления организации
CREATE PROCEDURE dbo.pOrganizations_Delete
    @ Additional_ID int
AS
BEGIN
    begin transaction;
    begin try
        DELETE FROM dbo.Organizations
        WHERE Organization_ID = @ Additional_ID
        DELETE FROM dbo.Additionals
        WHERE Subject_ID = @ Additional_ID
    end try
    begin catch
        if @@trancount > 0
            rollback transaction;
    end catch
    DECLARE
        @ErrorMessage varchar(4000),
        @ErrorSeverity int,
        @ErrorState int
```



```

SELECT
    @ErrorMessage = error_message(),
    @ErrorSeverity = error_severity(),
    @ErrorState = error_state();
raiserror(Действие невозможно, выбранная организация уже задействована в
системе!', @ErrorSeverity, @ErrorState);
return -1
end catch;
if @@trancount > 0
commit transaction;
END
GO

```

--Процедура на добавление пользователя

```

CREATE PROCEDURE pUsers_Insert
    @User_Name          varchar(50)
    ,@Person_ID         int
    ,@Passwordd         varchar(15)
    ,@User_ID           int out
    ,@Start_Access_Time time(7)
    ,@End_Access_Time   time(7)
AS
BEGIN
INSERT INTO dbo.Users
    ( User_Name
    , Person_ID
    , Passwordd
    , Start_Access_Time
    , End_Access_Time )
VALUES
    ( @User_Name
    , @Person_ID
    , @Passwordd
    , @Start_Access_Time
    , @End_Access_Time )
SET @User_ID = scope_identity()
END
GO

```

--Процедура на удаление пользователя

```

CREATE PROCEDURE pUsers_Delete
    @User_ID int
as
begin
begin transaction;
begin try

DELETE FROM dbo.Users
WHERE User_ID = @User_ID
end try
begin catch
print 'Случилась ошибка откатимся...'
if @@trancount > 0
rollback transaction;
DECLARE
    @ErrorMessage varchar(4000),
    @ErrorSeverity int,

```

```

        @ErrorState      int
SELECT
        @ErrorMessage = error_message(),
        @ErrorSeverity = error_severity(),
        @ErrorState = error_state();
raiserror(@ErrorMessage, @ErrorSeverity, @ErrorState);
return -1
end catch;
if @@trancount > 0
commit transaction;
END
GO

```

--Процедура на обновление адреса

```

CREATE PROCEDURE pAddresses_Update
    @Address_ID      int
    , @House_Number  varchar(20)
    , @Flat_Number   varchar(20)
AS
BEGIN
UPDATE dbo.Addresses
SET
    House_Number = @House_Number
    , Flat_Number = @Flat_Number
WHERE Address_ID = @Address_ID
END
GO

```

--Процедура для периода

```

CREATE PROCEDURE pGetPeriods
    @Period_ID int = null
AS
if @Period_ID = 0
SELECT
    0 Period_ID,
    'Следующий период' Period_Name
ELSE
SELECT
    Period_ID
    , Period_Name
    , Date_Begin
    , Date_End
    , Period_Year
    , Period_Month
    , Period_Days_Count
    , Period_Start_Date
    , Period_Last_Date
FROM dbo.Periods (nolock)
WHERE Period_ID = @Period_ID or @Period_ID is null
order by
    Period_ID desc
GO

```

2.9 Работа с приложением

Приложение позволяет редактировать базу данных, добавлять новые записи, содержит удобный дружественный интерфейс и позволяет пользователю быстро и эффективно работать с базой данных.

При запуске приложения открывается первая форма входа. Форма представлена на рисунке 2.20.

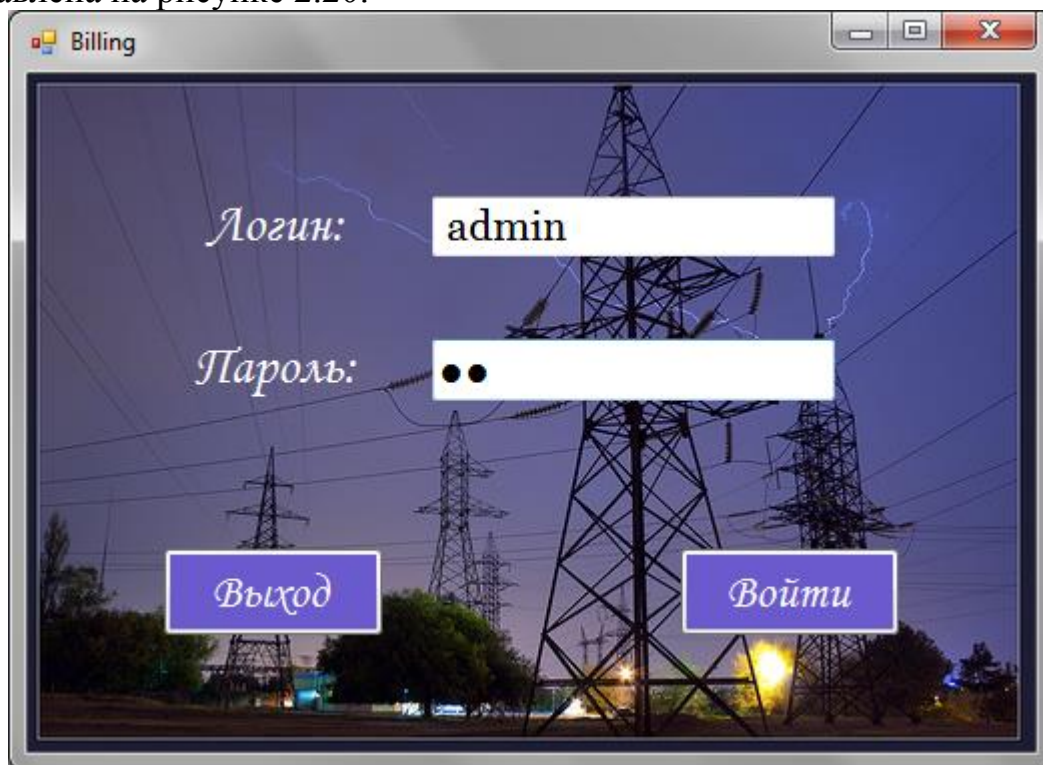


Рисунок 2.20 – Форма авторизации

Выплывающее окно сообщения при пустых строках или неверном вводе логина и пароля. Пример показан на рисунке 2.21.

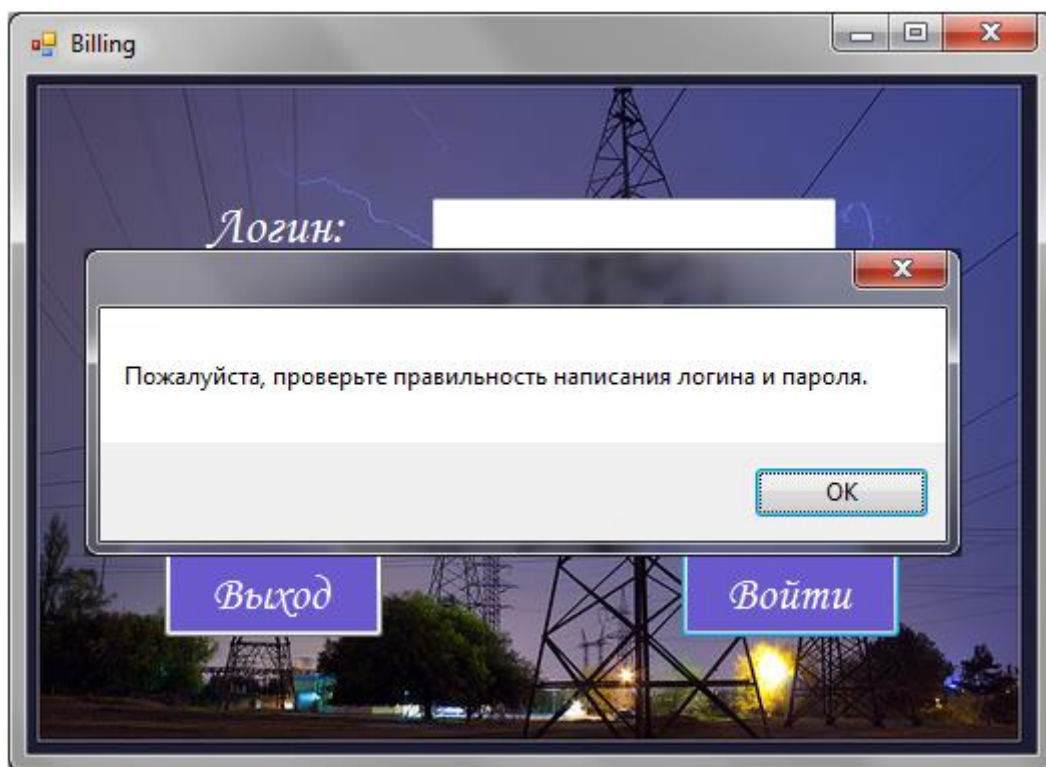


Рисунок 2.21 – Окно сообщения

После авторизации отрывается окно «Рабочий стол». Окно содержит в себе пункт меню, дату и время, поиск по договору и кнопку «Выйти». Форма приведена на рисунке 2.21.

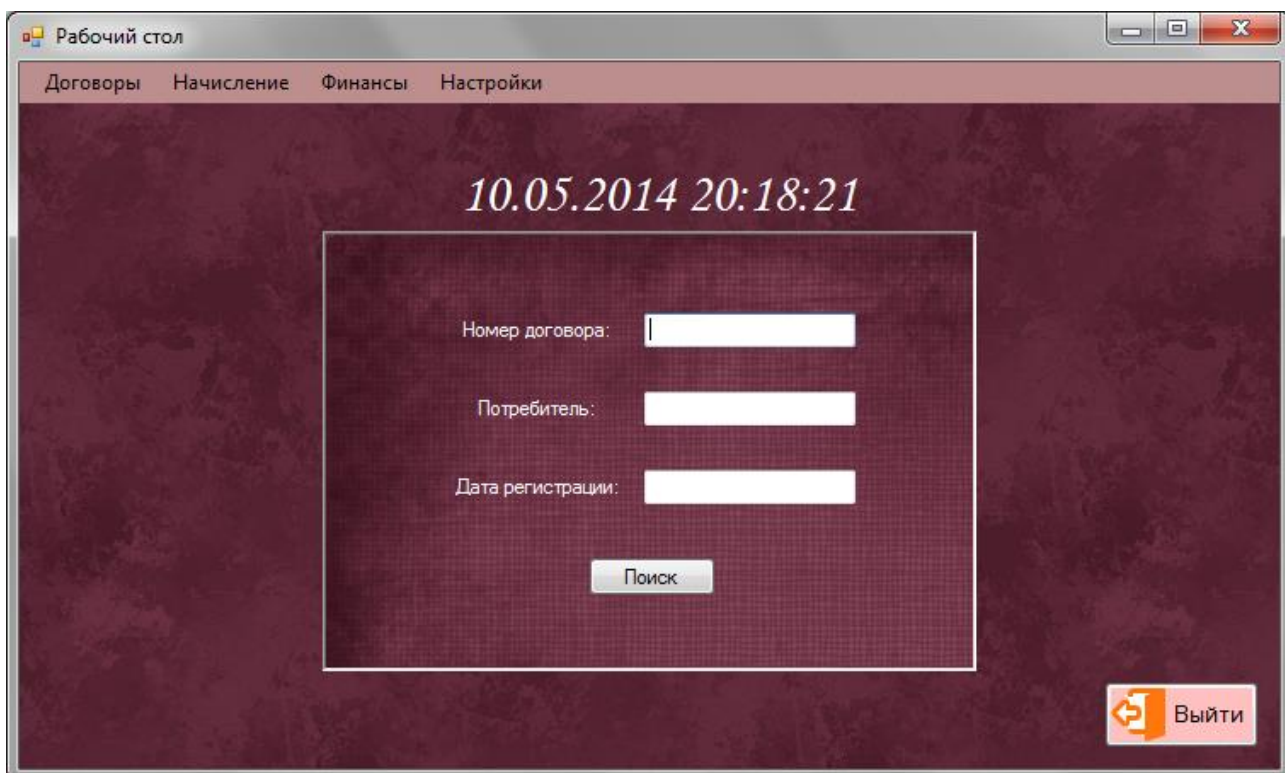


Рисунок 2.21 – Рабочий стол

Кнопка «Выйти» служит для выхода из программы. При нажатии кнопки отображается окно сообщения о подтверждения выхода из программы. Форма представлена на рисунке 2.22.

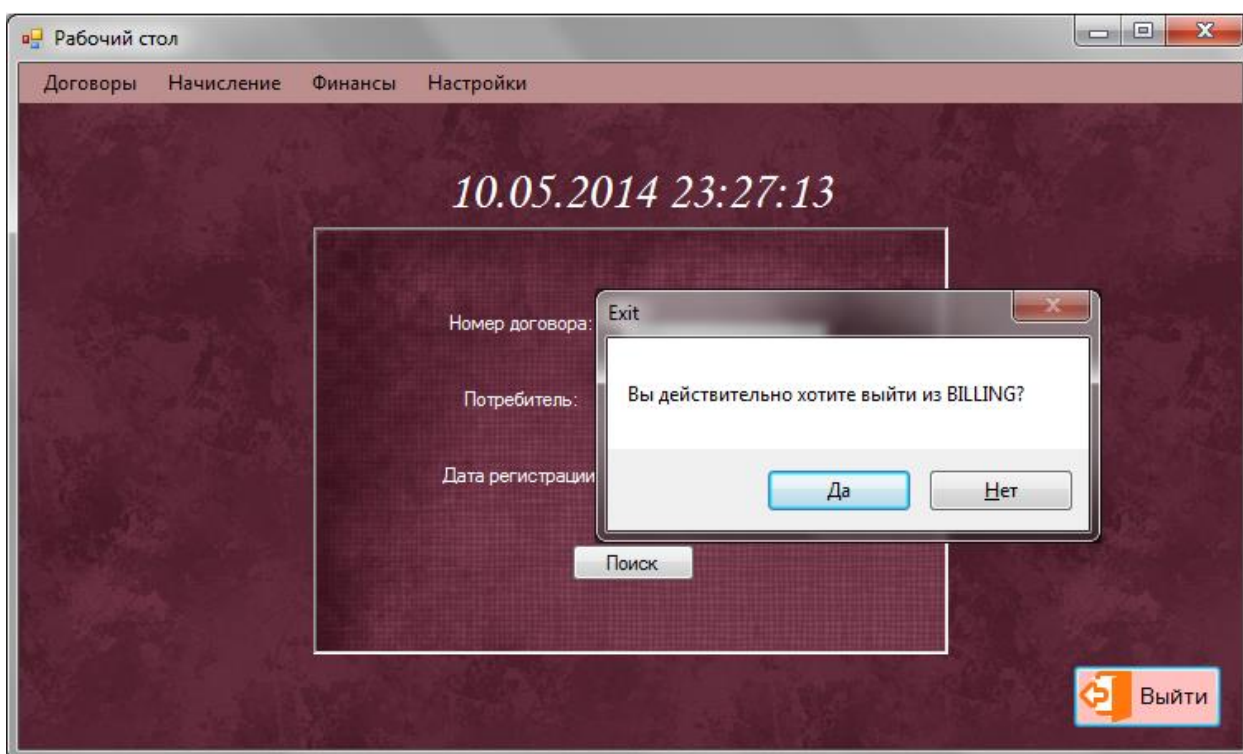


Рисунок 2.22 – Подтверждение выхода

При использовании пункта меню Договоры/Управление договорами, должно отобразиться форма «Договоры». В форме имеется строка поиска, в котором производится поиск по номеру договора, также имеется кнопка «Зарегистрировать договор», «Редактировать договор» и кнопка «Закреть». Данная форма представлена на рисунке 2.23.

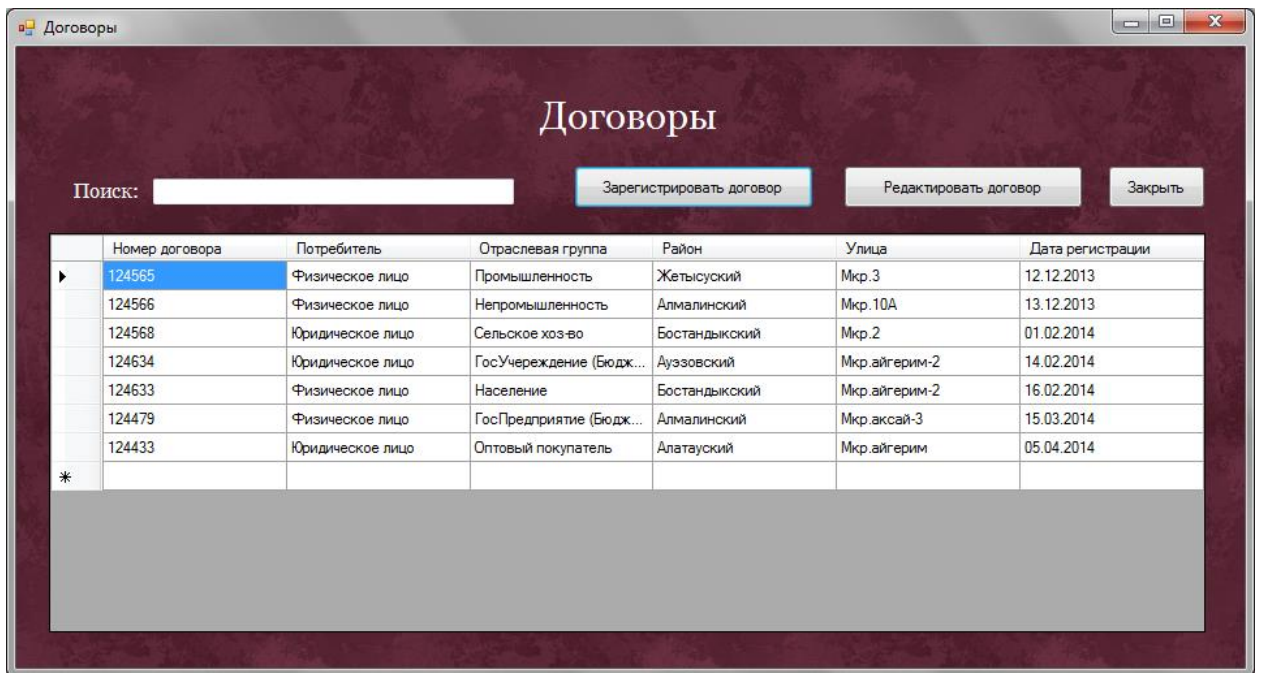


Рисунок 2.23 – Форма «Договоры»

Кнопка «Регистрация договора». При использовании кнопки отображается форма «Регистрация договора», которая содержит в себе строки для заполнения договора, кнопки «Записать договор», «Закреть» и кнопку «Домик», которая переносит на форму «Рабочий стол». Форма приведена на рисунке 2.24.

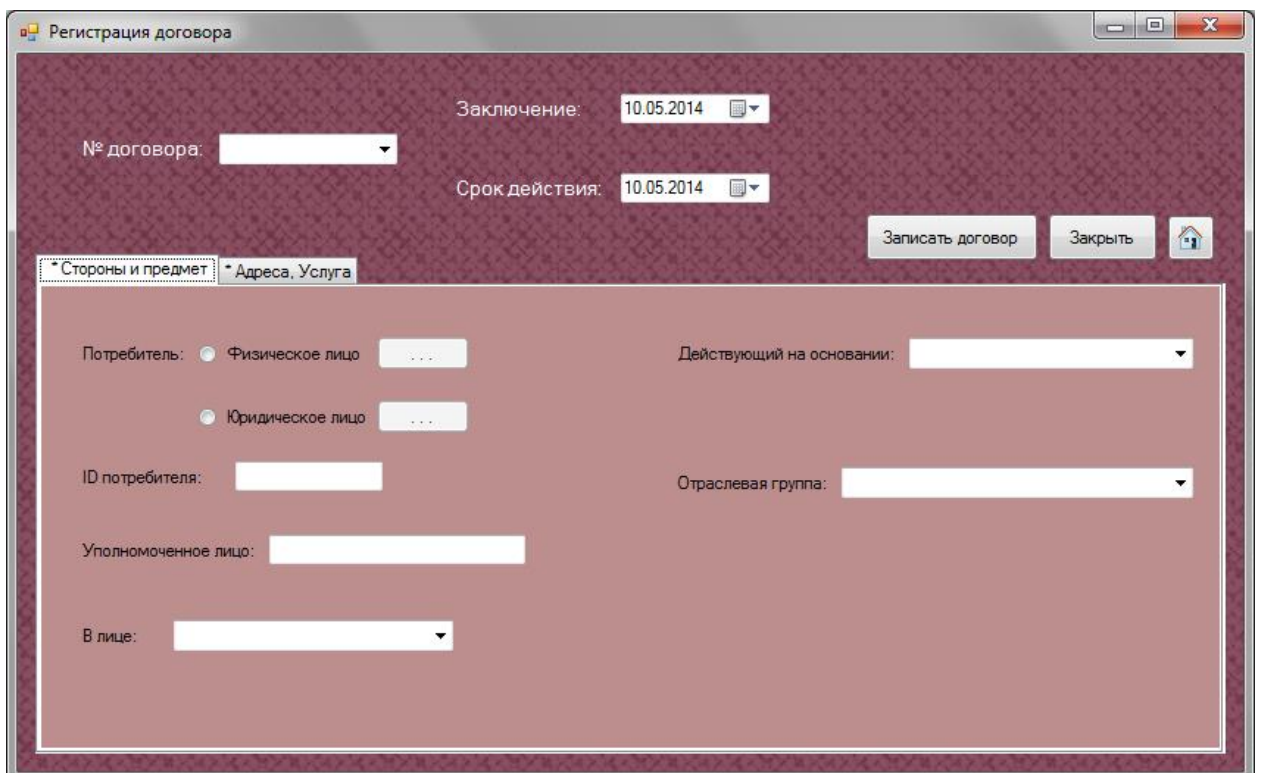


Рисунок 2.24 – Форма «Регистрация договора»

Вкладка «Адреса, Услуга» в форме «Регистрация договора». Вкладка представлена на рисунке 2.25.

The screenshot shows a software window titled "Регистрация договора" (Contract Registration). The window has a dark red background. At the top, there are two date fields: "Заключение:" (Conclusion) and "Срок действия:" (Term of validity), both set to "10.05.2014". To the left, there is a field for "№ договора:" (Contract No.). Below these, a tabbed interface shows two tabs: "Стороны и предмет" (Parties and Subject) and "Адреса, Услуга" (Addresses, Service), with the latter being active. In the active tab, there are three fields: "Адрес доставки:" (Delivery Address) with a dropdown menu and a button with three dots; "Услуга:" (Service) with a dropdown menu; and "Тариф:" (Rate) with a dropdown menu. In the top right corner of the window, there are three buttons: "Записать договор" (Record Contract), "Закрыть" (Close), and a home icon.

Рисунок 2.25 – Вкладка «Адреса, Услуга»

В строке «Адрес доставки» имеется кнопка, которая при нажатии отображает форму «Адрес». В форме можно выбрать район, улица/микрорайон, записать номер дома и квартиры. Форма представлена на рисунке 2.26.

Рисунок 2.26 – Форма «Адреса»

Также можно ввести новый адрес при использовании кнопки «Другое», которая отображает форму «Новый адрес». Форма показана на рисунке 2.27.

Рисунок 2.27 – Форма «Новый адрес»

Для отображения формы «Физические лица» используется пункт меню Договоры/Физические лица. В форме производится поиск по фамилии

физического лица, имеются кнопки: «Зарегистрировать», «Редактировать», «Заккрыть». Форма представлена на рисунке 2.28.

Физические лица

Поиск: Зарегистрировать Редактировать Заккрыть

ID	Сокращенное ФИО	Номер телефона	ИНН	Район	Улица	Номер дома	Номер квартиры
1001	Шивазова М. К.	8(771)413-57-88	94555977337	Бостандыкский	Мкр.2	15	57
1002	Рындина А. А.	8(702)454-454-2	47557933325	Медеуский	Мкр.11	5	14
1003	Федоров А. В.	8(705)44-22-388	19151797341	Алмалинский	Мкр.10А	15	63
1004	Минин С. П.	8(771)423-07-07	11124579777	Медеуский	Мкр.айгерим	45	64
1005	Мельник А. Д.	8(705)123-123-4	44497357711	Бостандыкский	Мкр.айгерим-2	88	36
1006	Кусаинов Р. Р.	8(707)166-45-86	99555973599	Алмалинский	Мкр.аксай-3	63	19
1007	Хайрулина А. С.	8(701)277-55-69	94559332745	Алатауский	Мкр.айгерим	14	42
1008	Алиев Н. Д.	8(701)731-56-17	94779971123	Жетысуский	Мкр.10	18	12
1009	Затеев Н. В.	8(701)224-76-76	94793777597	Турксибский	Мкр.6	55	78
1010	Бондарев М. Д.	8(777)721-37-38	47771217111	Жетысуский	Мкр.3	76	74
1011	Чичизаева А. А.	5-45-34	154546465423	Бостандыкский	Мкр.2	15	57

Рисунок 2.28 – Форма «Физические лица»

При использовании кнопки «Зарегистрировать» отображается форма «Регистрация физического лица». Вид формы представлен на рисунке 2.29.

Регистрация физического лица

ID: (1001-1999) * Фамилия: Пол:

* Имя: Дата рождения:

Отчество: Сохранить Заккрыть

* Документы * Адреса, Контакты

ИНН: РНН:

Документ удостоверяющий личность

Вид документа: Дата выдачи:

Серия: Срок действия:

Номер: Кем выдан:

Рисунок 2.29 – Форма «Регистрация физического лица»

Вкладка «Адреса, контакты» в форме «Регистрация физического лица». Вид вкладки показан на рисунке 2.30.

Регистрация физического лица

ID: (1001-1999) * Фамилия: Пол: Дата рождение: 10.05.2014

* Имя: Отчество: Сохранить Закрыть

Документы Адреса, Контакты

ID Адреса: ID: Телефон: E-mail: Вид занятости: Место работы:

Рисунок 2.30 – Вкладка «Адреса, контакты»

Нажав на кнопку «Редактировать» открывается форма «Редактирование физических лиц», в которой можно изменить, удалить и сохранить данные. Форма приведена на рисунке 2.31.

Редактирование физических лиц

1 of 12

Закрыть

ID	Фамилия	Имя	Отчество	Пол	Дата рождения	Телефон	E-Mail	РНН	ИИН
1001	Шивазова	Майя	Кемаровна	Женский	26.08.1992	8(771)413-57-88	mai92@mail.ru	145558966336	94555977337
1002	Рындина	Анастасия	Андреевна	Женский	28.11.1991	8(702)454-454-2	12@mail.ru	478556933325	47557933325
1003	Федоров	Андрей	Викторович	Мужской	07.08.1989	8(705)44-22-388	fd07@mail.ru	010507889634	19151797341
1004	Минин	Сергей	Павлович	Мужской	15.12.1986	8(771)423-07-07	15ser@mail.ru	000245689666	11124579777
1005	Мельник	Анна	Денисовна	Женский	14.02.1987	8(705)123-123-4	ann5@mail.ru	444888963567	44497357711
1006	Кусаинов	Руслан	Рустамович	Мужской	02.07.1982	8(707)166-45-86	rus1982@mail.ru	115558963599	99555973599
1007	Хайрулина	Альбина	Сергеевна	Женский	05.04.1984	8(701)277-55-69	al84@mail.ru	145593326485	94559332745
1008	Алиев	Нурсултан	Диасович	Мужской	12.11.1982	8(701)731-56-17	nur12@mail.ru	148779960023	94779971123
1009	Затеев	Николай	Владимирович	Мужской	30.05.1981	8(701)224-76-76	nik05@mail.ru	147936665896	9479377597
1010	Бондарев	Максим	Денисович	Мужской	24.09.1985	8(777)721-37-38	max85@mail.ru	477788020608	47771217111
1011	Ччизаева	Алина	Аликовна	Женский	05.04.1993	5-45-34	lll	133214612153	15454646542

Рисунок 2.31 – Форма «Редактирование физических лиц»

Для отображения данных в форме «Организации» используется пункт меню Договоры/Организация. В форме производится поиск по названию

организации, имеются кнопки: «Зарегистрировать», «Редактировать», «Заккрыть». Форма представлена на рисунке 2.32.

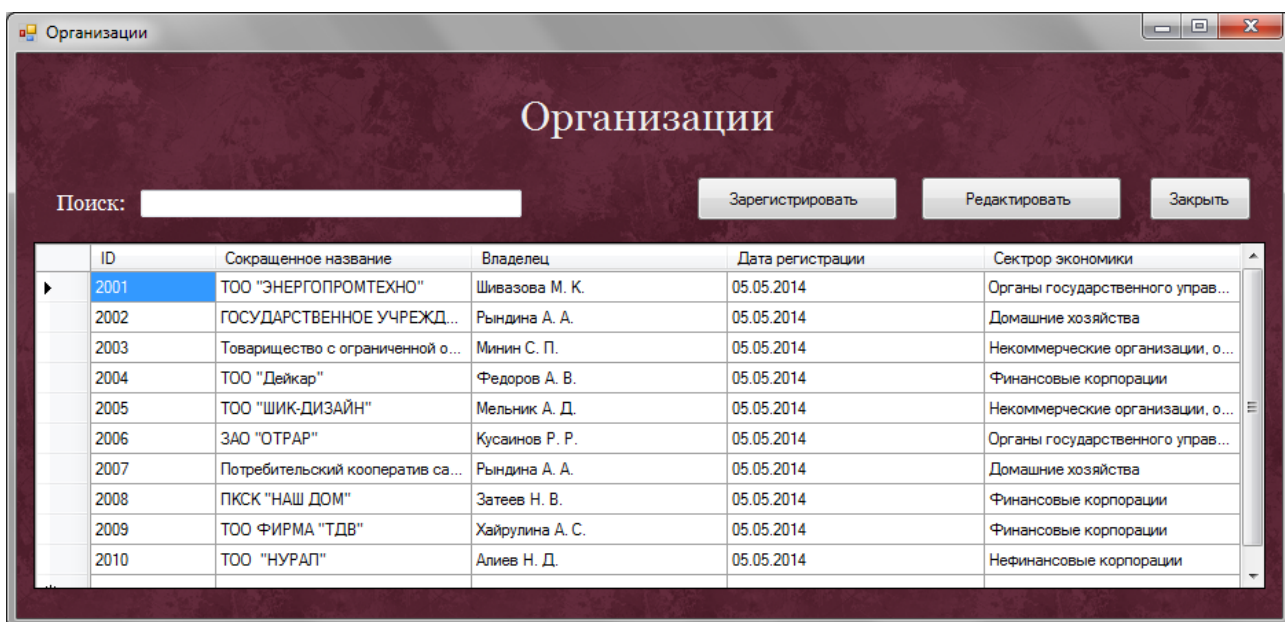


Рисунок 2.32 – Форма «Организации»

При нажатии кнопки «Зарегистрировать» отображается форма «Регистрация организации». Вид формы представлен на рисунке 2.33.

Регистрация организации

ID организации: 1011 (2001-2999)

* Сокращенное наименование: ИП Даурен

* Полное наименование: ИП Даурен

Сохранить Закрыть

* Реквизиты * Дополнительно * Адреса, Контакты

РНН: 321548898451

КБЕ: 12

Свидетельство о государственной регистрации

БИН/ИНН: 123156456432

Номер: 4546562 Дата выдачи свидетельства: 15.07.2006

Серия: 132

Рисунок 2.33 – Форма «Регистрация организации»

В вкладке «Дополнительно» записывается дополнительные сведения об организации. Вкладка приведена на рисунке 2.34.

The screenshot shows a web application window titled "Регистрация организации". At the top, there are three input fields: "ID организации:" with a dropdown menu and "(2001-2999)" below it; "* Сокращенное наименование:" with a text input field; and "* Полное наименование" with a text input field. Below these are three tabs: "* Реквизиты", "* Дополнительно" (which is active), and "* Адреса, Контакты". In the "Дополнительно" tab, there are three more input fields: "Сектор экономики:" with a dropdown menu; "Вид деятельности:" with a dropdown menu; and "Головная организация:" with a text input field. On the right side of the form, there are three buttons: "Сохранить", "Заккрыть", and a home icon.

Рисунок 2.34 – Форма «Регистрация организации»

Вкладка «Адреса, Контакты» в форме «Регистрация организации». Вкладка представлена на рисунке 2.35.

The screenshot shows the same "Регистрация организации" window, but with the "Адреса, Контакты" tab selected. The top section with "ID организации:", "* Сокращенное наименование:", and "* Полное наименование" remains the same. The tabs are now "* Реквизиты", "* Дополнительно", and "* Адреса, Контакты" (which is active). In the "Адреса, Контакты" tab, there is an "ID:" dropdown menu in the top right corner. Below it are three input fields: "ID Адреса:" with a dropdown menu and a blue button next to it; "Телефон:" with a text input field; and "E-mail:" with a text input field. The "Сохранить", "Заккрыть", and home icon buttons are still present on the right.

Рисунок 2.35 – Вкладка «Адреса, Контакты»

При нажатии кнопки «Редактировать» открывается форма «Редактирование организации», в которой можно изменить, удалить и сохранить данные. Форма приведена на рисунке 2.36.

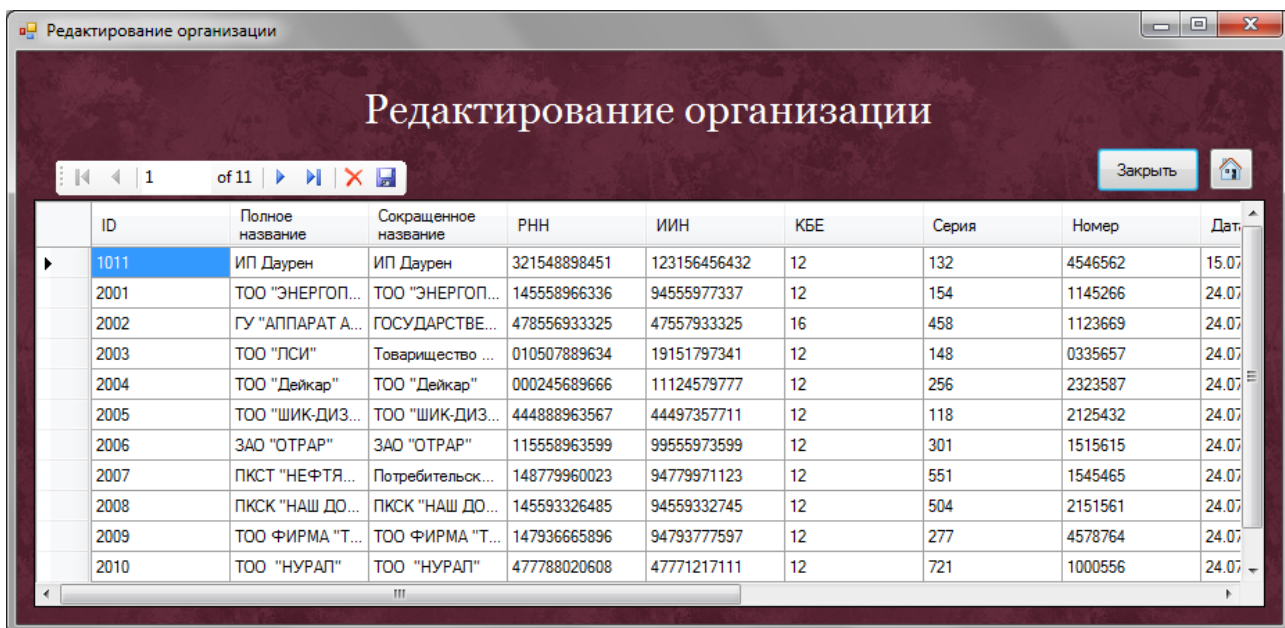


Рисунок 2.36 – Форма «Редактирование организации»

Для отображения данных в форме «Начисление» используется пункт меню Начисление/Начисление. В форме содержатся кнопки: «Провести начисление», «Закреть». Форма представлена на рисунке 2.37.

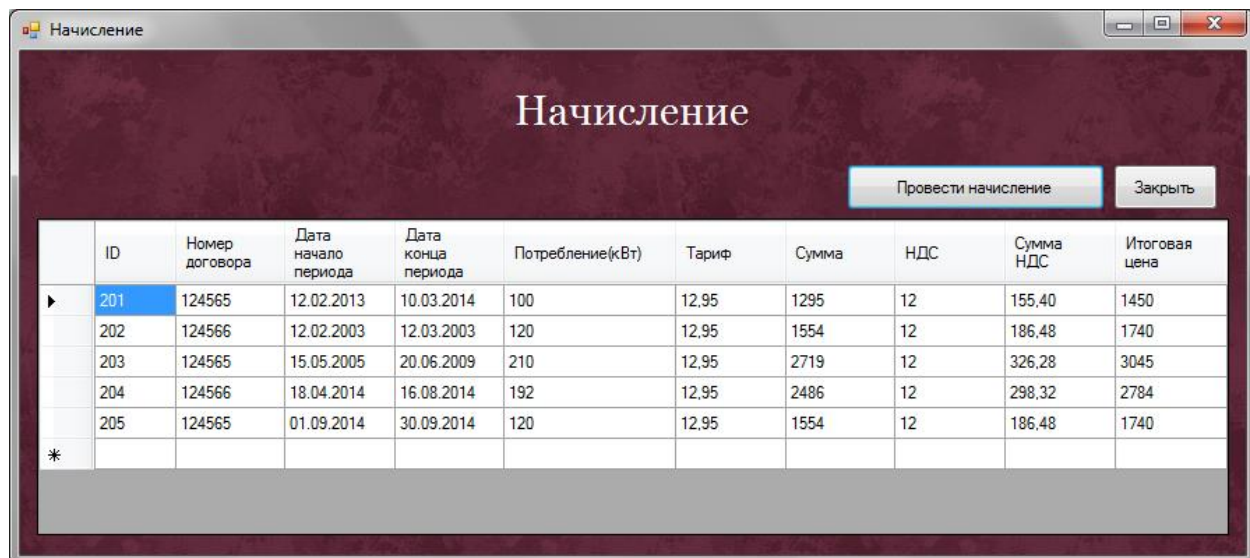


Рисунок 2.37 – Форма «Начисление»

При использовании кнопки «Провести начисление» отображается форма «Начисление электроэнергии». Вид формы представлен на рисунке 2.38.

Начисление электроэнергии

ID: 401

Период: Сентябрь 2014

Дата начало периода: 01.09.2014

Дата окончания периода: 30.09.2014

Потребление (кВт): 120

Тариф: 12,95 тенге

НДС: 12 %

Сохранить

Закрыть

Рисунок 2.38 – Форма «Провести начисление»

Для отображения формы «Оборотно-сальдовая ведомость» используется пункт меню Финансы/ Оборотно-сальдовая ведомость. Форма представлена на рисунке 2.39.

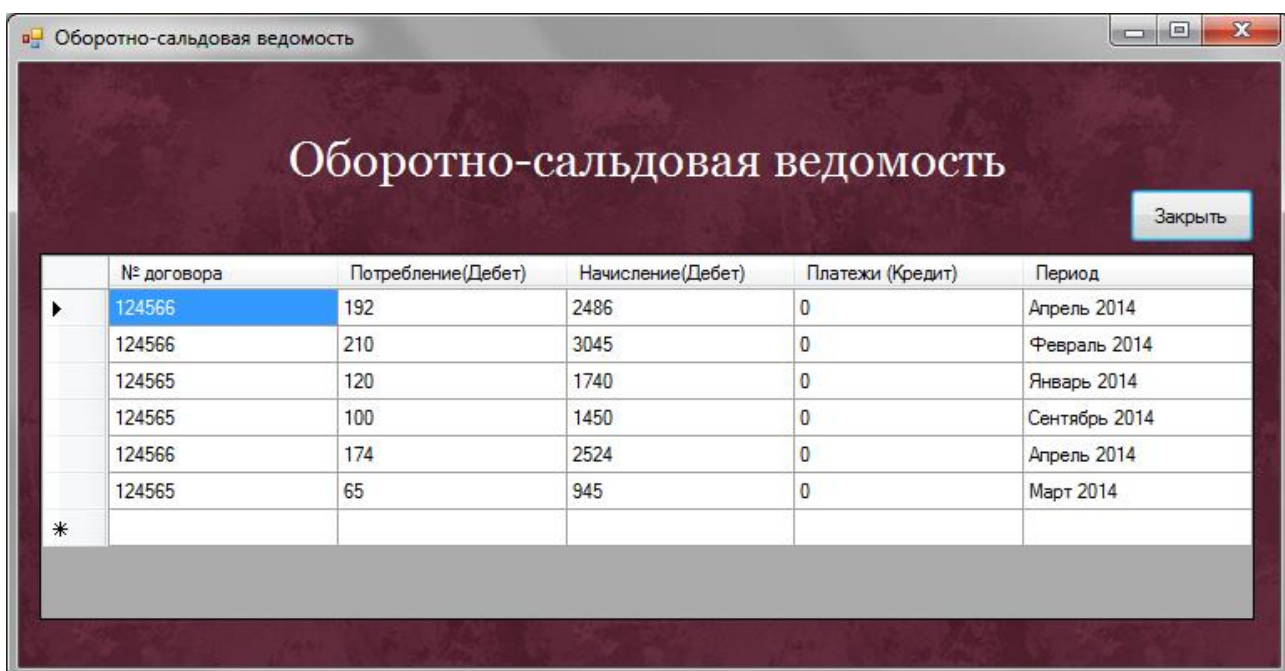


Рисунок 2.39 – Форма «Оборотно-сальдовая ведомость»

Для отображения формы «Управление учетными записями» используется пункт меню Настройки/ Учетные записи. В форме производится поиск по логину, имеются кнопки: «Добавить», «Редактировать», «Закреть». Форма представлена на рисунке 2.40.

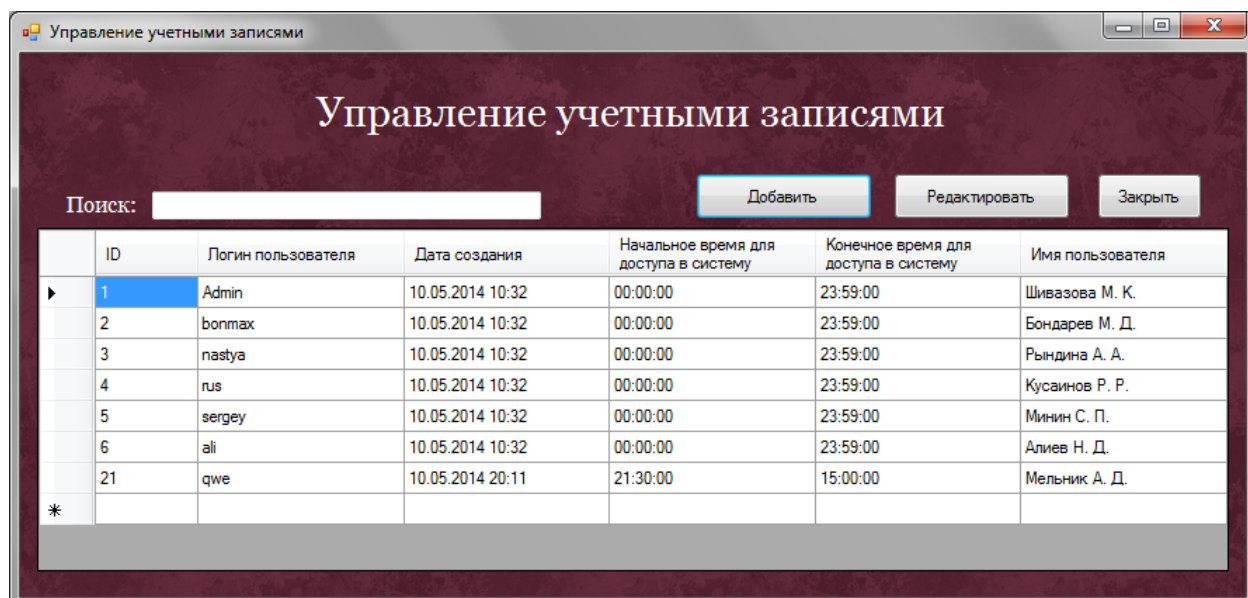


Рисунок 2.40 – Форма «Управление учетными записями»

При использовании кнопки «Добавить» отображается форма «Создание учетной записи». Форма представлена на рисунке 2.41.

Создание учетной записи

Логин пользователя: Mai

Пароль: ●●●●●

Выбор физ.лица: 1010 ...

Начальное время для доступа в систему: 11:30

Конечное время для доступа в систему: 13:30

Сохранить Отменить

Рисунок 2.41 – Форма «Создание учетной записи»

При нажатии кнопки «Редактировать» открывается форма «Редактирование учетной записи», в которой можно удалить и сохранить данные. Форма представлена на рисунке 2.42.

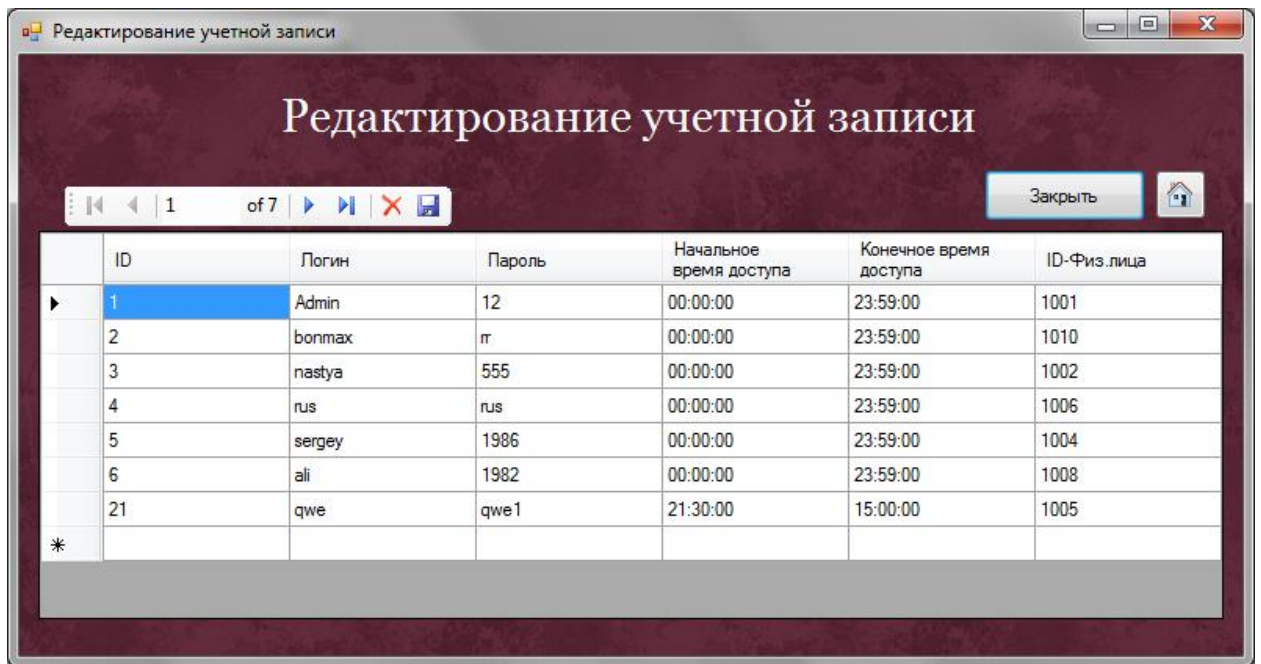


Рисунок 2.42 – Форма «Создание учетной записи»

3 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА

3.1 Описание работы и обоснование необходимости

В дипломном проекте рассмотрены вопросы создания автоматизации биллинга электропотребления. Для реализации биллинга использованы наиболее эффективные приемы и опыт создания БД.

Дипломный проект разрабатывается для жителей города Алматы.

В данном разделе дипломного проекта приводится экономический анализ расчетов разработки составляющей реализации данного биллинга, отражающей временные, трудовые и финансовые затраты на проект.

3.2 Трудовые ресурсы, используемые в работе

В данном дипломном проекте используется интеллектуальный труд. И для выполнения полного дипломного проекта задействована группа людей, состоящая из четырех человек:

- Руководитель проекта
- Программист-разработчик
- Консультант по экономической части
- Консультант по БЖД

Данные о работниках и их заработной платы представлены в таблице 3.1.

Т а б л и ц а 3.1 – Данные о работниках, задействованных в проекте и их заработная плата

Исполнитель	Количество человек	Зарботная плата, тенге
Руководитель	1	90 000
Программист-разработчик	1	210 000
Консультант по части «Экономика»	1	50 000
Консультант по части «БЖД»	1	50 000
Итого	4	400 000

3.3 Оборудование и программное обеспечение, используемое в работе

Необходимо учесть оборудование, которое использовано в разработке, так как оно входит долю затрат, затем определить долю амортизации от итоговой стоимости используемого ПО при разработке.

Для разработки используются следующее оборудование:

- ноутбук Samsung Intel(R) Core™ i5-3210M 2.50GHz 4Gb/HDD 500Gb;
- мышка Microsoft Explorer Touch/U5K-00015;

– USB-модем DIIJ.

Информация по количеству каждого вида оборудования и их стоимости представлены в таблице 3.2.

Т а б л и ц а 3.2 – Используемое оборудование

Наименование	Единица измерения	Количество	Цена за единицу в тенге	Сумма с НДС в тенге	Сумма без НДС в тенге
Ноутбук Samsung Intel(R) Core™ i5-3210M 2.50GHz 4Gb/HDD 500Gb	штук	1	150 000	150 000	132 000
Мышка Microsoft Explorer Touch/U5K-00015	штук	1	9 780	9 780	8 607
USB-модем DIIJ	штук	1	4 000	4 000	3 520
Windows 7 Professional	штук	1	37 200	37 200	32 736
Microsoft SQL Server 2008 R2	штук	1	0	0	0
Microsoft Visual Studio 2010	штук	1	0	0	0
Итого:				200 980	176 863

Все программное обеспечение, необходимое для реализации данного проекта, а именно: MS SQL Server 2008 R2, Microsoft Visual Studio 2010 имеют бесплатную форму распространения.

Общая стоимость расходов составляет 176 863 тенге.

3.4 Расчет стоимости ПО

Расчет осуществляется по калькуляционным статьям расходов. Прежде всего необходимо рассчитать себестоимость на протяжении всего жизненного цикла. Себестоимость – это все затраты на производство и реализацию продукции.

Себестоимость подсчитывается из следующих статей затрат:

- оплата труда;
- социальный налог;
- амортизационные отчисления;
- расходы на электроэнергию;
- накладные расходы.

Следовательно, себестоимость разработки проекта осуществляется по следующей формуле

$$C = \Phi OT + C_H + A + P_э + H \quad (3.1)$$

где C – себестоимость продукта;
 ΦOT – фонд оплаты труда;
 C_H – социальный налог;
 A – амортизационные отчисления;
 $P_э$ – расходы на электроэнергию;
 H – накладные расходы.

3.5 Сроки реализации проекта

Разработка биллинга включает следующие этапы:

- 1 этап: постановка задачи, выбор алгоритма;
- 2 этап: создание базы данных, создание пользователей и выдача им привилегий, создание процедур и триггеров, заполнение базы данных;
- 3 этап: создание интерфейса базы данных, оформление дизайна;
- 4 этап: проверка работоспособности ПО;
- 5 этап: оформление отчетов.

Этапы создания ПО «Автоматизация биллинга электропотребления» представлены в таблице 3.3.

Т а б л и ц а 3.3 – Этапы и сроки реализации работы

Перечень работ		Недели от начала работ									
		1	2	3	4	5	6	7	8	9	10
1 этап	Постановка задачи	■									
	Выбор алгоритма	■									
	Подбор и изучение литературы		■								
2 этап	Создание базы данных		■	■	■						
	Создание пользователей и выдача им привилегий				■						
	Создание процедур и триггеров					■					
	Заполнение базы данных					■					
3 этап	Создание интерфейса					■	■	■			
	Оформление дизайна							■			
4 этап	Тестирование ПО							■			
	Отладка недочетов								■		
5 этап	Подготовка раздела «Экономика»								■		
	Подготовка раздела «БЖД»									■	
	Проверка и сдача отчета									■	■

3.5.1 Расчёт основной заработной платы производственного персонала

Для расчета заработной платы необходимы следующие данные:

- численность задействованного персонала;
- среднемесячная заработная плата каждого работника;
- длительность разработки и каждого вида выполняемых работ;
- трудоемкость.

В процессе разработки данного ПО участвует 4 человека.

Месячная заработная плата работников:

- Руководитель – 90 000 тенге;
- Программист-разработчик – 210 000 тенге;
- Консультант по части «Экономика» – 50 000 тенге;
- Консультант по части «БЖД» – 50 000 тенге.

Расчет заработной платы производить по формуле

$$D = \frac{ЗПм}{Др \cdot Чр} \quad (3.2)$$

где $ЗПм$ – ежемесячный размер заработной платы;

$Др$ – количество рабочих дней в месяц (21 рабочий день);

$Чр$ – количество часов рабочего дня (при 8 часовом рабочем дне).

1 Заработная плата руководителя за один час составляет:

$$D = \frac{90000}{21 \cdot 8} = \frac{90000}{168} = 535,7 \text{ тенге/час};$$

2 Заработная плата программиста-разработчика за один час составляет:

$$D = \frac{210000}{21 \cdot 8} = \frac{210000}{168} = 1250 \text{ тенге/час};$$

3 Заработная плата консультанта по части «Экономика» за один час составляет:

$$D = \frac{50000}{21 \cdot 8} = \frac{50000}{168} = 298 \text{ тенге/час};$$

4 Заработная плата консультанта по части «БЖД» за один час составляет:

$$D = \frac{50000}{21 \cdot 8} = \frac{50000}{168} = 298 \text{ тенге/час.}$$

Длительность цикла в днях по каждому виду работы укрупнено можно определяется по формуле:

$$t_n = \frac{T}{q_n \cdot z \cdot K}, \quad (3.3)$$

где T – трудоёмкость этапа, нормо-час;

q_n – количество исполнителей по этапу;

z – продолжительность рабочего дня, $z=8$ часов;

K – коэффициент выполнения норм времени ($K=1,1$).

Полученная величина t_n округляется в большую сторону до целых дней.

$$t_1 = \frac{20}{1 \cdot 8 \cdot 1,1} = \frac{20}{8,8} \approx 3 \text{ дня; Руководитель: постановка задачи;}$$

$$t_2 = \frac{10}{1 \cdot 8 \cdot 1,1} = \frac{10}{8,8} \approx 2 \text{ дня.; Руководитель: выбор алгоритма;}$$

$$t_3 = \frac{17}{1 \cdot 8 \cdot 1,1} = \frac{17}{8,8} \approx 2 \text{ дня; Руководитель: подбор и изучение литературы;}$$

$$t_4 = \frac{95}{1 \cdot 8 \cdot 1,1} = \frac{95}{8,8} \approx 11 \text{ дней; Программист-разработчик: создание базы}$$

данных;

$$t_5 = \frac{12}{1 \cdot 8 \cdot 1,1} = \frac{12}{8,8} \approx 2 \text{ дня; Программист-разработчик: создание}$$

пользователей и выдача привилегий;

$$t_6 = \frac{24}{1 \cdot 8 \cdot 1,1} = \frac{24}{8,8} \approx 3 \text{ дня; Программист-разработчик: создание процедур и}$$

триггеров;

$$t_7 = \frac{8}{1 \cdot 8 \cdot 1,1} = \frac{8}{8,8} \approx 1 \text{ день; Программист-разработчик: заполнение базы}$$

данных;

$$t_8 = \frac{61}{1 \cdot 8 \cdot 1,1} = \frac{61}{8,8} \approx 7 \text{ дней; Программист-разработчик: создание}$$

интерфейса;

$$t_9 = \frac{17}{1 \cdot 8 \cdot 1,1} = \frac{17}{8,8} \approx 3 \text{ дня; Программист-разработчик: оформление дизайна;}$$

$$t_{10} = \frac{8}{1 \cdot 8 \cdot 1,1} = \frac{8}{8,8} \approx 1 \text{ день; Программист-разработчик: тестирование ПО;}$$

$$t_{11} = \frac{10}{1 \cdot 8 \cdot 1,1} = \frac{10}{8,8} \approx 2 \text{ дня; Программист-разработчик: отладка недочетов;}$$

$$t_{12} = \frac{17}{1 \cdot 8 \cdot 1,1} = \frac{17}{8,8} \approx 2 \text{ дня; Консультант по части «Экономика»: подготовка}$$

раздела экономика;

$$t_{13} = \frac{17}{1 \cdot 8 \cdot 1,1} = \frac{17}{8,8} \approx 2 \text{ дня; Программист-разработчик: подготовка раздела}$$

экономика;

$$t_{14} = \frac{17}{1 \cdot 8 \cdot 1,1} = \frac{17}{8,8} \approx 2 \text{ дня; Консультант по части «БЖД»: подготовка}$$

раздела безопасность жизнедеятельности;

$$t_{15} = \frac{17}{1 \cdot 8 \cdot 1,1} = \frac{17}{8,8} \approx 2 \text{ дня; Программист-разработчик: подготовка раздела}$$

безопасность жизнедеятельности;

$$t_{16} = \frac{17}{1 \cdot 8 \cdot 1,1} = \frac{17}{8,8} \approx 2 \text{ дня; Программист-разработчик: проверка и сдача}$$

отчета;

$$t_{17} = \frac{17}{1 \cdot 8 \cdot 1,1} = \frac{17}{8,8} \approx 2 \text{ дня; Руководитель: проверка и сдача отчета;}$$

Свободные результаты расчета затрат на основную заработную плату работников, задействованных в разработке автоматизации биллинга, предоставлены в таблице 3.4.

Т а б л и ц а 3.4 – Свободные результаты расчета затрат на заработную плату.

№	Наименование содержания работы	Исполнитель	Трудоемкость		Длительность цикла, дни	Заработная плата за час работы, тенге	Сумма заработной платы, тенге
			Нормы часы	% от общей трудоемкости			
1	Постановка задачи	Руководитель	20	5,2	3	535,7	10714
2	Выбор алгоритма	Руководитель	10	2,6	2	535,7	5357
3	Подбор и изучение литературы	Руководитель	17	4,43	2	535,7	9106,9
4	Создание базы данных	Программист-разработчик	95	24,7	11	1250	118750

Окончание таблицы 3.4

5	Создание пользователей и выдача им привилегий	Программист-разработчик	12	3,13	2	1250	15000
6	Создание процедур и триггеров	Программист-разработчик	24	6,3	3	1250	30000
7	Заполнение базы данных	Программист-разработчик	8	2,08	1	1250	10000
8	Создание интерфейса	Программист-разработчик	61	15,9	7	1250	76250
9	Оформление дизайна	Программист-разработчик	17	4,43	2	1250	21250
10	Тестирование ПО	Программист-разработчик	8	2,08	1	1250	10000
11	Отладка недочетов	Программист-разработчик	10	2,6	2	1250	12500
12	Подготовка раздела экономика	Консультант по части «Экономика»	17	4,43	2	298	5066
13	Подготовка раздела экономика	Программист-разработчик	17	4,43	2	1250	21250
14	Подготовка раздела безопасность жизнедеятельности	Консультант по части «БЖД»	17	4,43	2	298	5066
15	Подготовка раздела безопасность жизнедеятельности	Программист-разработчик	17	4,43	2	1250	21250
16	Проверка и сдача отчета	Программист-разработчик	17	4,43	2	1250	21250
17	Проверка и сдача отчета	Руководитель	17	4,43	2	535,7	9106,9
Итого:			384	100	48	-	401917

Дополнительная заработная плата составляет 10% от основной заработной платы и рассчитывается по формуле

$$Z_{\text{доп}} = ЗП \cdot 10\% \quad (3.4)$$

Согласно формуле (3.4), дополнительная заработная плата составляет:

$$Z_{\text{доп}} = 401917 \cdot 0,1 = 40191,7 \text{ тенге}$$

Таким образом, суммарные фонд оплаты труда работников составляет:

$$\Phi OT = 401917 + 40191,7 = 442108,7 \text{ тенге}$$

3.5.2 Расчет затрат по социальному налогу

Согласно разделу 11 «Социальный налог» статья 358 «Налогового кодекса» НК РК для юридических лиц - резидентов, для юридических лиц - нерезидентов, осуществляющих деятельность через постоянное учреждение, филиалы и представительства, уплачивают социальный налог в размере 11% от начисленных доходов.

Социальный налог рассчитывается по формуле

$$C_H = (\Phi OT - \text{ПО}) \cdot 11\% \quad (3.5)$$

где ΦOT – фонд оплаты труда;
 ПО – пенсионные отчисления.

Пенсионные отчисления вычисляются по формуле

$$\text{ПО} = \Phi OT \cdot 10\% \quad (3.6)$$

$$\text{ПО} = 442108,7 \cdot 10\% = 44210,87 \text{ тенге}$$

Таким образом, отчисления на социальный налог составляет:

$$C_H = (442108,7 - 44210,87) \cdot 11\% = 397897,83 \cdot 11\% = 43768,8 \text{ тенге}$$

3.5.3 Расчет амортизационных отчислений

Амортизационные отчисления рассчитываются по формуле

$$A_i = \frac{H_A \cdot C_{\text{ПЕР}} \cdot N}{100 \cdot 12 \cdot n} \quad (3.7)$$

где H_A – норма амортизации;
 $C_{\text{ПЕР}}$ – первоначальная стоимость оборудования;
 N – количество дней на выполнение работ;
 n – количество дней в рабочем месяце.

Норма амортизации H_A на компьютерную технику составляет 40% от стоимости всего оборудования.

Таким образом, амортизационные отчисления по используемому оборудованию, в соответствии с формулой 3.6 составляет:

– на ноутбук:

$$A_1 = \frac{40 \cdot 132000 \cdot 48}{100 \cdot 12 \cdot 21} = \frac{253440000}{25200} = 10057,14 \text{ тенге}$$

– на мышь:

$$A_2 = \frac{40 \cdot 8607 \cdot 48}{100 \cdot 12 \cdot 21} = \frac{16525440}{25200} = 655,8 \text{ тенге}$$

– на USB-модем:

$$A_3 = \frac{40 \cdot 3520 \cdot 48}{100 \cdot 12 \cdot 21} = \frac{6758400}{25200} = 268,19 \text{ тенге}$$

Свободные результаты расчета амортизационных отчислений предоставлены в таблице 3.5

Т а б л и ц а 3.5 – Сведения по затратам на амортизацию.

Наименование оборудования	Количество	Цена за единицу, тенге	Норма амортизации, %	Сумма амортизации, тенге
Ноутбук Samsung Intel(R) Core™ i5-3210M	1	132 000	40	10057,14
Мышка Microsoft Explorer Touch/U5K-00015	1	8 607	40	655,8
USB-модем DIIJ	1	3 520	40	268,19
Итого				10 981,13

3.5.4 Расчет затрат на электроэнергию

В процессе производства используется электрооборудование, необходимо рассчитать затраты на электроэнергию. Затраты на электроэнергию рассчитывается по формуле (3.8) для производственных нужд включают в себя расходы электроэнергии на оборудование и дополнительные нужды.

$$\mathcal{E} = \mathcal{E}_{\text{эл.+эн.+обор}} + \mathcal{E}_{\text{доп. нуж.}} \quad (3.8)$$

где $Z_{эл.+эн.+обор}$ – затраты на электроэнергию оборудования;

$Z_{доп. нуж.}$ – затраты электроэнергии на дополнительные нужды.

Расходы на электроэнергию на оборудование рассчитываются по формуле

$$Z_{эл.+эн.+обор} = W \cdot T \cdot S \cdot K_{исп}, \quad (3.9)$$

где W – потребляемая мощность, Вт;

T – время работы;

S – тариф (1 кВт=21,4тг/кВт в час без НДС);

$K_{исп}$ – коэффициент использования ($K_{исп}=0,9$).

$$Z_{эл.+эн.+обор1} = 0,8 \cdot 384 \cdot 21,4 \cdot 0,9 = 5916,7 \text{ тенге}$$

$$Z_{эл.+эн.+обор2} = 0,6 \cdot 40 \cdot 21,4 \cdot 0,9 = 462,24 \text{ тенге}$$

Суммарные затраты на электроэнергию основного оборудования составляют:

$$Z_{эл.+эн.+обор} = 5916,7 + 462,24 = 6378,9 \text{ тенге}$$

Затраты на дополнительные нужды берутся по укрупненному показателю в размере 5% от затрат на оборудование и составляют:

$$Z_{эл.+эн.+обор} = 6378,9 \cdot 0,05 = 318,9 \text{ тенге}$$

Таким образом суммарные затраты на электроэнергию составляют:

$$C_3 = 6378,9 + 318,9 = 6697,9 \text{ тенге}$$

Сводные результаты расчета затрат на электроэнергию представлены в таблице 3.6.

Т а б л и ц а 3.6 – Затраты на электроэнергию

Наименование приборов	W, Вт	Число рабочих дней	$K_{исп}$	Время работы прибора, час	$\sum W$, кВт	Сумма за электроэнергию, тенге
Ноутбук	0,8	48	0,9	384	307,2	6378,9
Принтер	0,6	5	0,9	40	24	318,9
Итого $\sum W$					324,8	6697,9

3.5.5 Расчет затрат на накладные расходы

Накладные расходы на разработку программного обеспечения составляют от 25% от общей суммы затрат и рассчитываются по формуле

$$H = \Phi OT \cdot 0,25 \quad (3.9)$$

Тогда, согласно формуле 3.9, накладные расходы будут равны:

$$H = 442108,7 \cdot 0,25 = 110527,2 \text{ тенге}$$

3.5.6 Расчет стоимости по всем статьям затрат

В соответствии с произведенными расчетами по статьям затрат себестоимости проекта, согласно формуле 3.1, будет равна:

$$C = 442108,7 + 43768,8 + 10981,13 + 6697,9 + 110527,2 = 614083,7 \text{ тенге}$$

Сводные результаты расчета себестоимости разработки ПО представлены в таблице 3.7 и на рисунке 3.1.

Т а б л и ц а 3.7 – Стоимость разработки ПО «Автоматизации биллинга электропотребления»

Наименование статей затрат	Сумма, тенге	Структура затрат, %
ФОТ	442108,7	56
Отчисления на социальные нужды	43768,8	5,6
Амортизация	10981,13	1,4
Затраты на электроэнергию	6697,9	0,85
Используемое оборудование	176863	22,4
Накладные расходы	110527,2	14
Итого	790946,73	100

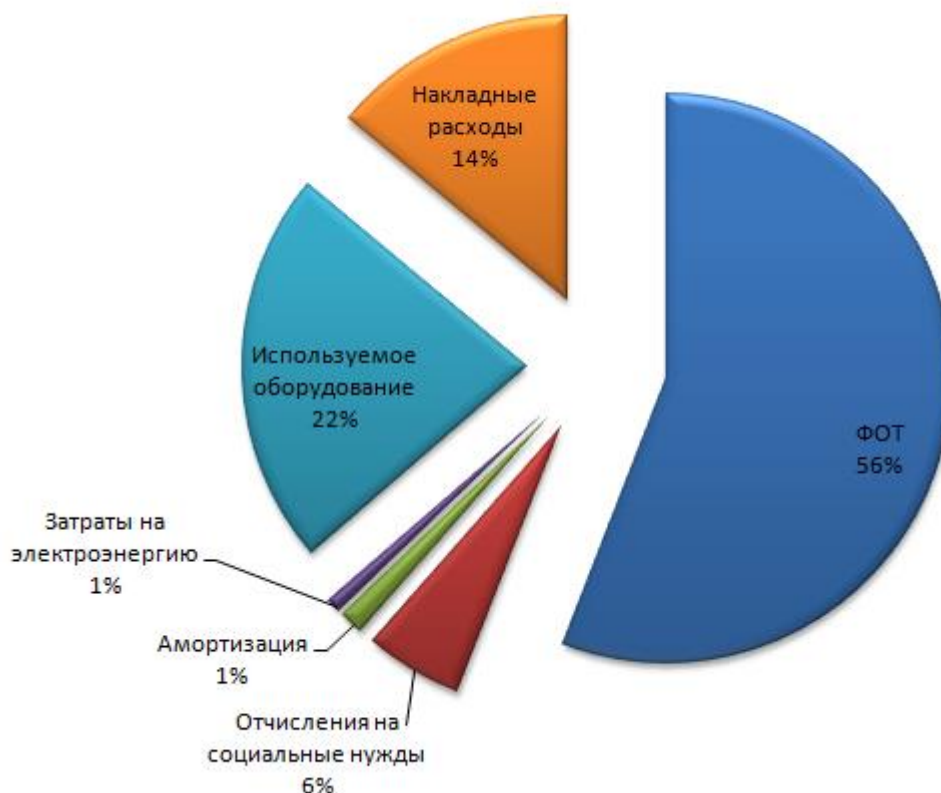


Рисунок 3.1 – Структура себестоимости разработки ПО «Автоматизации биллинга электропотребления»

Цена реализации программного продукта складывается из себестоимости и чистого дохода, и вычисляется по формуле

$$Ц = C + П \quad (3.10)$$

где C – себестоимость продукта;
 $П$ – чистый доход.

Первоначальная цена рассчитывается через рентабельность работы. Учитывая, что желаемый уровень рентабельности составляет 30%, применим следующую формулу

$$Ц_{п} = C \cdot \left(1 + \frac{P}{100}\right) \quad (3.11)$$

где P – рентабельность (30%).

Согласно формуле 3.10 первоначальная цена равна:

$$C_{II} = 790946,7 \cdot \left(1 + \frac{30}{100}\right) = 1028231 \text{ тенге}$$

Цена реализации готовой продукции рассчитывается по формуле

$$C_p = C_{II} + НДС \quad (3.12)$$

Поскольку на сегодняшний день размер НДС в РК составляет 12%, следовательно

$$НДС = \frac{12}{100} \cdot C_{II} \quad (3.13)$$

$$НДС = \frac{12}{100} \cdot 1028231 = 123387,7 \text{ тенге}$$

Тогда согласно формуле (3.12) цена реализации составит:

$$C_p = 1028231 + 123387,7 = 1151618 \text{ тенге}$$

3.6 Вывод

Себестоимость программного продукта «Автоматизация биллинга электропотребления» составила 790946,73 тенге. Наибольшую долю в себестоимости ПО составляют затраты на оплату труда 442108,7 тенге (56%) и затраты на используемое оборудование 176863 тенге (22%), а цена реализации продукта составляет 1151618 тенге.

4. БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

В данном дипломном проекте разрабатываю программное обеспечение «Автоматизация биллинга электропотребления» для города Алматы.

Работа с программным обеспечением производится из головного офиса. С целью обеспечения всех норм и правил безопасности жизнедеятельности, в проекте рассмотрено помещение главного офиса компании, где и будет находиться обслуживающий персонал. Работа и обслуживание будет осуществляться с помощью шести специалистов..

4.1 Анализ условий труда в производственном помещении

Головной офис компании будет находиться в двенадцатиэтажном здании на втором этаже.

На рисунке 4.1 показан план помещения для обслуживающего персонала компании. Помещение имеет следующие размеры: длина $L = 10$ м, ширина $B = 7$ м и высота $H = 4$ м.

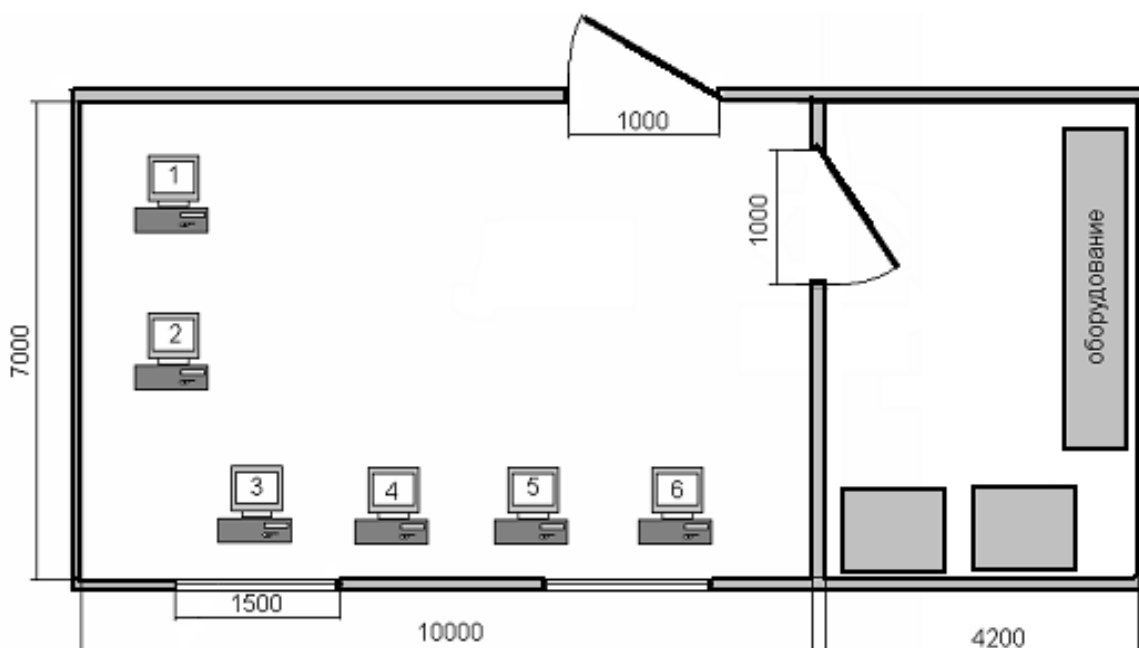


Рисунок 4.1 – План помещения

Комната с оборудованием ранее проектировалась как серверная, и все нормы освещения и кондиционирования соблюдены. И так как основное время обслуживающий персонал будет проводить за своими рабочими местами, позднее более подробно остановлюсь на соблюдении норм именно в этой комнате.

В здании офиса и серверной уровень опасных и вредных факторов не превышает установленных нормативов на рабочих местах и каждое рабочее место оператора максимально приспособлено для характера выполняемых

работ. Помещение светлое, сухое и чистое, полностью соответствует санитарно-гигиеническим нормам.

Расстояние между столами не менее 1,2 м. А минимальные размеры рабочего места оператора составляют: площадь – 6 м². В зале находятся 6 рабочих мест для обслуживающего персонала. Далее определим размеры площади на каждое рабочее место и получим, что на каждое рабочее место приходится около 10 м², то есть соответствует нормам (10 м² > 6 м²).

В комнате располагается два световых проема размером 1,5м×2м (Д×В), при которых освещение соответствует нормам. Анализируя освещенность рабочего места, необходимо правильно организовать искусственное освещение в производственном помещении.

Микроклимат производственного помещения оказывает значительное влияние на работника. В зависимости от энергозатрат организма ГОСТ 12.1.005-76 ССБТ "Воздух рабочей зоны, общие санитарно-гигиенические требования" предусматривает 3 категории работ.

В данном помещении работа относится к категории Ia – легкая физическая работа, где энергозатраты организма (расход энергии при выполнении работы) – 173 – 233 Дж/с, 150 – 300 ккал/ч – производится сидя и не требует физического напряжения. Оптимальные и допустимые параметры микроклимата для этой категории работ в теплый и холодный период года приведены в таблице 4.1.

Т а б л и ц а 4.1 – Параметры микроклимата для категории работ Ia

Нормы	Оптимальные			Допустимые		
	Период работы	Температура, °С	Относит влажность, %	Скорость движения воздуха, м/с, не более	Температура воздуха, °С	Относит. влажность, %, не более
Холодный	22-24	30-60	0,1	21-25	80	0,1
Теплый	23-25	40-60	0,1	22-28	75	0,1-0,2

4.1.1 Анализ освещенности рабочего места

При помощи глаз человек получает около 90% всей информации. Полученная информация во многом зависит от освещения. При неудовлетворительном освещении зрительный аппарат человека напрягается, что ведет к ухудшению зрения и состояния организма в целом. Одновременно с этим человек теряет ориентацию среди оборудования, что повышает опасность его травмирования.

Освещение РМ должно быть сходно по спектральному составу с солнечным светом как наиболее гигиеничным; достаточным и соответствовать СНиП II-4-79; равномерным и устойчивым; без резких теней и блеклости в поле зрения; соответствующей цветности и не являться источником дополнительных вредных и опасных факторов.

В зависимости от источника света освещение может быть естественным, искусственным и совмещенным. По функциональному назначению освещение подразделяется на рабочее, аварийное, эвакуационное и дежурное. Рабочее освещение использует естественный и искусственный свет, а другие виды освещения – только искусственный свет.

Искусственное освещение используется в темное время суток и в помещениях, где нет естественного освещения. Источниками искусственного освещения являются лампы накаливания (ЛН) и газоразрядные лампы (ГРЛ). Выбор искусственных источников света производят по СНиП II - 4-79 в зависимости от характера зрительных работ по цветоразличению.

Уровень освещенности на рабочем столе в зоне расположения документов должен быть в границах 300 - 500 лк. Общее освещение - неяркий рассеянный свет от люминесцентных светильников дневного света на потолке. Яркость светильников общего освещения в зоне углов излучения от 50° до 90° , условно вертикали в продольной и поперечной плоскостях должна составлять не больше 200 кд/м^2 , а защитный угол светильников должен быть не больше 40° , согласно ГОСТ 2.04.05-91. Также необходимо ограничивать отраженный блеск посредством правильного выбора типов светильников и размещением рабочих мест относительно источников естественного и искусственного освещения. При такой яркости отблесков на экранах видеомониторов не должна превышать 40 кд/м^2

К освещению рабочих помещений предъявляются исключительные требования. Естественное освещение осуществляется через окна, и нормирование производится с помощью коэффициента естественной освещенности (КЕО,%)

$$KEO = \frac{E_B}{E_H} \cdot 100, \quad (4.1)$$

где E_B – освещенность точки внутри помещения, лк.

E_H – наружная освещенность, лк.

КЕО определяется по СНиП II-4-79 .

В помещении, где проектируется программное обеспечение, располагается два оконных проема и освещение отвечает нормам. Высота рабочей поверхности над уровнем пола – 0,8 м. Окна начинаются с высоты – 1,2 м. В здании используется боковое одностороннее освещение, световые

проемы расположены на одной стороне стены. Количество окон равно двум, длина каждой равна 1,5 м, а высота 2 м.

Так как работа в помещении предполагается до позднего вечера, то необходимо произвести расчет искусственного освещения.

4.1.2 Расчет искусственного освещения

Проектирование искусственного освещения заключается в решении следующих задач: выбор системы освещения, типа источника света, расположение светильников, выполнение светотехнического расчета и определение мощности осветительной установки.

Расчет искусственного общего освещения выполняется по методу коэффициента использования светового потока.

Этот метод предназначен для расчета общего равномерного освещения горизонтальных поверхностей при отсутствии крупных затененных предметов.

Размеры объектов различения находятся в пределах 1-5 мм, разряд зрительной зоны работы определен пятой малой точности, поэтому будет экономична система общего освещения, при которой светильники будут расположены в верхней зоне, и обеспечивать равномерную освещенность рабочего помещения площадью 70 м², высотой 4 м.

На основании этих требований проведем расчет системы общего освещения рабочего места оператора.

Расположение светильников в помещении определяется: h – высотой подвеса над рабочей поверхностью, z – расстояние между рядами люминесцентных ламп, l – расстоянием от крайних рядов светильников до стены.

$$h = H - (h_c + h_p), \quad (4.2)$$

где h_c – расстояние от светильника до перекрытия;

h_p – высота рабочей поверхности над полом.

Расчетная высота подвеса – рабочая поверхность находится на высоте 0,8 м от пола, высота свеса ламп – 0,5 м., следовательно,

$$h = 4 - (0,8 + 0,5) = 2,7 \text{ м}$$

Расстояние между рядами светильников:

$$z = \lambda \cdot h = 1,2 \cdot 2,7 = 3,24 \text{ м} \quad (4.3)$$

где $\lambda = 1,2$ – коэффициент неравномерности;

h – высота подвеса.

Принимаем 2 ряда светильников с расстоянием от стен 1,2 м., между рядами по 3,6 м.

Нормируемая минимальная освещенность определяется по формуле:

$$E_{\min} = \frac{F_{\text{л}} \cdot N \cdot \eta}{S \cdot K \cdot z} \quad (4.4)$$

где $F_{\text{л}}$ – световой поток одной лампы;

N – число ламп в помещении;

η – коэффициент использования светового потока, т.е. доля светового потока всех ламп, падающая на освещаемую поверхность;

z – коэффициент неравномерности освещения;

$S = A \cdot B$ – площадь поля освещаемого помещения;

K – коэффициент запаса, учитывающий снижение освещенности в процессе эксплуатации системы освещения (загрязнение светильников, старение ламп).

Коэффициент использования светового потока представляет собой отношение светового потока, достигающего освещаемой поверхности, к полному световому потоку в помещении. Зависит от коэффициентов отражения стен $\rho_{\text{с}}$ и потолка $\rho_{\text{п}}$, показателя (индекса) помещения, который вычисляется по формуле:

$$i = \frac{A \cdot B}{H_{\text{р}}(A + B)}, \quad (4.5)$$

где $H_{\text{р}}$ – высота подвеса светильников над рабочей поверхностью (условно рабочей поверхностью считается горизонтальная поверхность на высоте 0,8 м. от пола). Люминесцентные светильники рекомендуется устанавливать на высоте 2,5 или 4 м.

Так как нормируется минимальная освещенность рабочей поверхности, то при расчетах вводится коэффициент неравномерности освещения Z .

Для люминесцентных ламп $Z = 1,1$.

Задавшись числом ламп, из формулы (4.4) имеем

$$F_{\text{л}} = \frac{E_{\min} \cdot S \cdot K \cdot Z}{N \cdot \eta} \quad (4.6)$$

Для этой категории работ при общем освещении наименьшая освещенность $E_{\min} = 300$ лк.

Коэффициент пульсации освещенности не более 15 %.

Коэффициент запаса $K=1,5$.

Коэффициент неравномерности освещения $z = 1,1$.

Пусть помещение, где установлено оборудование, имеет следующие размеры: длина $A = 10$ м, ширина $B = 7$ м, высота $H = 4$ м.

Подвесной потолок оборудован светильниками с люминесцентными лампами ЛБ-40.

Коэффициенты отражения светового потока от стен и потолка соответственно равны: $\rho_{ст} = 50\%$, $\rho_{пт} = 70\%$.

Определим необходимое число светильников при общей системе освещения.

Для помещения с ЭВМ уровень рабочей поверхности над полом равен 0,8 м. При этом $H_p = 3,2$ (высота подвеса над рабочей поверхностью).

Площадь помещения:

$$S = A \cdot B = 10 \cdot 7 = 70 \text{ м}^2$$

Световой поток, создаваемый одной лампой $F_{л}$

$$F_{л} = \frac{300 \cdot 70 \cdot 1,5 \cdot 1,1}{N \cdot \eta} = 3120 \text{ лм.}$$

Определим сначала показатель (индекс) помещения

$$i = \frac{(A \cdot B)}{(H_p (A + B))} = \frac{10 \cdot 7}{(3,2 \cdot (10 + 7))} = 0,3 \quad (4.7)$$

Теперь для $i=1$, коэффициентов отражения потолка $\rho_{пт} = 0,7$ и стен $\rho_{ст} = 0,5$, находим коэффициент использования светового потока $\eta = 50\%$.

Необходимое число светильников определяется по формуле:

$$N = \frac{(E_{\min} \cdot S \cdot K \cdot Z)}{(F_{л} \cdot N \cdot \eta)} = \frac{(300 \cdot 70 \cdot 1,5 \cdot 1,1)}{(3120 \cdot 2 \cdot 0,5)} = 8,1 = 8 \text{ шт.} \quad (4.8)$$

Число ламп в светильнике равно двум. Общее количество ламп равно:

$$N = 2 \cdot 8 = 16 \text{ шт.}$$

Разделив N на число рядов, можно определить число светильников устанавливаемых в каждом ряду.

Пусть светильники устанавливаются в два ряда. По восемь светильников в каждом ряду.

Длина светильника ЛД-40 = 1,213 м, длина одного ряда $8 \cdot 1,213 = 9,6$ м. Размещаем в один ряд 8 светильников с расстоянием от стены 0,2 м.

4.1.3 Расчёт освещения методом удельной мощности

Сущность расчета освещения по методу удельной мощности заключается в том, что в зависимости от типа светильника и места его установки, высоты подвеса над рабочей поверхностью, освещенности на горизонтальной поверхности и площади помещения определяется значение удельной мощности.

Удельной мощностью называется отношение установленной мощности ламп к величине освещаемой площади ($\text{Вт}/\text{м}^2$), значения ее для различных ламп приведены в таблицах.

Большие значения удельной мощности принимаются для помещений с меньшей площадью освещения.

Затем, задавшись числом светильников, обеспечивающих равномерности освещенности, определяют мощность общей лампы по формуле:

$$\rho = \frac{\omega \cdot S}{N}, \quad (4.9)$$

где ω - удельная мощность;
 S – площадь помещения;
 N – число светильников.

Этот метод является частным случаем метода коэффициента использования.

Принимаем к установке светильники ЛД и располагаем их параллельно в два ряда по восемь в каждом.

Удельная мощность светильника $\omega = 12 \text{ Вт}/\text{м}^2$. Мощность одного светильника:

$$\rho = \frac{12 \cdot 70}{16} = 52,5 \text{ Вт.}$$

Следовательно, нужно установить шестнадцать светильников мощностью 60 Вт. На рисунке 4.2 представлена схема расположения светильников.

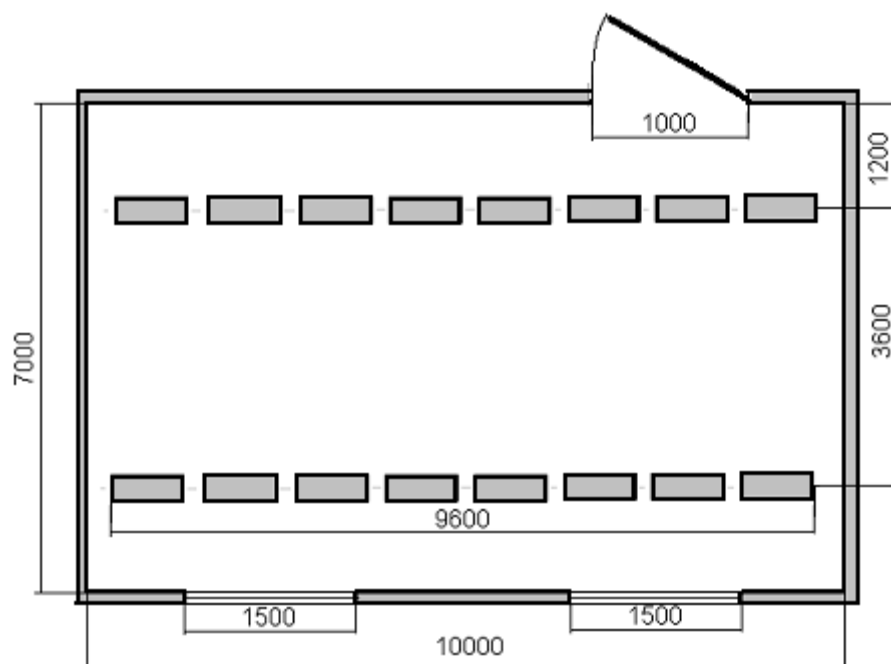


Рисунок 4.2 – Расположение светильников

В ходе данной работы при проектировании осветительной установки определены следующие основные вопросы:

- а) выбрали схему освещения, тип источника света, тип светильников;
- б) произвели размещение светильников;
- в) рассчитали освещенность в нужных точках;
- г) уточнили размещение и число светильников;
- д) определили единичную мощность светильников и ламп.

4.1.4 Расчет системы вентиляции

В основу расчета всех систем вентиляции лежат приближенные методы, учитывающие с помощью коэффициентов различные факторы, влияющие на производительность вентиляции.

Чем больше коэффициентов входит в расчетные формулы, тем больше факторов они учитывают, и точнее дают результаты.

Стоимость систем вентиляции и кондиционирования воздуха в зданиях сопоставима по величине с затратами, которые являются следствием плохого качества воздуха. Задача кондиционирования воздуха состоит в поддержании таких параметров воздушной среды, при которых каждый человек благодаря своей индивидуальной системе автоматической терморегуляции организма чувствовал бы себя комфортно, т.е. не замечал влияния этой среды.

Перспективными с точки зрения создания нормальных микроклиматических условий в рабочей зоне является использование кондиционирующих установок.

Определим необходимое количество кондиционеров для формирования комфортных условий труда в производственном помещении. Расчет произведем по методике для теплого времени года. В помещении за счет тепловыделений производственного оборудования могут иметь место значительные избытки тепла (разность между тепловыделениями в помещении и теплоотдачей через стены, окна, двери и т.д.), удаление которых, прежде всего, должна обеспечить система вентиляции.

В помещениях со значительными тепловыделениями количество приточного воздуха, м³/ч, необходимого для поглощения избытка тепла, рассчитывается по формуле:

$$L = \frac{\Sigma Q_{изб}}{C(t_{уд} - t_{пр}) \cdot \gamma_{пр}}, \text{ м}^3/\text{ч} \quad (4.10)$$

где $Q_{изб}$ – избыточное выделение явной теплоты, Дж/ч;

C – теплоемкость воздуха, Дж/(кг·град);

$t_{уд}$ – температура удаляемого из помещения воздуха, 27°С;

$t_{пр}$ – температура приточного воздуха, 20°С;

$\gamma_{пр}$ – плотность приточного воздуха, 1,2 кг/м³.

Избыточное тепло определяется по формуле

$$Q_{изб} = Q_1 + Q_2 + Q_3 + Q_4, \quad (4.11)$$

где Q_1 – тепловыделение от аппаратуры;

Q_2 – тепловыделение от источников освещения;

Q_3 – тепловыделение от людей;

Q_4 – теплопоступление от солнечной радиации сквозь окна.

Тепловыделение от аппаратуры определяется по формуле

$$Q_1 = \eta_{НОМ} \cdot \eta_s \cdot \eta_0 \cdot \eta_4 \cdot P_{НОМ} \quad (4.12)$$

где $\eta_{НОМ}$ – коэффициент использования номинальной мощности, 0,7 – 0,9;

η_s – коэффициент загрузки, т.е. отношение средней потребляемой мощности к максимально необходимой, равной 0,5 – 0,8;

η_0 – коэффициент одновременной работы аппаратуры, 0,5 – 1;

η_4 – коэффициент, характеризующий долю механической энергии, превратившийся в тепло ($\eta_4 = 0,65 - 1,0$);

$P_{НОМ}$ – номинальная мощность всей аппаратуры.

$$Q_1 = 0,8 \cdot 0,6 \cdot 0,6 \cdot 0,75 \cdot 3500 = 756 \text{ кДж/ч.}$$

Тепловыделение от источников искусственного освещения рассчитывают, предполагая, что практически вся затрачиваемая энергия, в конечном счете, преобразуется в тепло, по формуле:

$$Q_2 = \varphi \cdot N_{осв} \quad (4.13)$$

где φ – коэффициент, учитывающий количество энергии переходящей в тепло, $\varphi = 0,8$;

$N_{осв}$ – мощность осветительной установки помещения (16 ламп по 60 Вт каждая).

$$Q_2 = 0,8 \cdot (16 \cdot 60) = 768 \text{ Дж/ч.}$$

Тепловыделение от людей определяют по формуле

$$Q_3 = n \cdot q_n \quad (4.14)$$

где n – число работающих;

q_n – количество тепла, выделяемого одним человеком, Вт.

$$Q_3 = 6 \cdot 120 = 720 \text{ Дж/ч.}$$

Количество тепла, поступающего в помещение через окна от солнечной радиации, определяют по формулам

$$Q_4 = F_{ост} \cdot q \cdot m \cdot k, \quad (4.15)$$

где $F_{ост}$ – площадь окна, м²;

m – число окон;

k – поправочный множитель, для металлического переплета $k = 1,25$;

q – теплоступление через 1 м² окна, $q = 224 \text{ Вт/ м}^2$ (окна выходят на юг, юго-восток, юго-запад).

$$Q_4 = 3 \cdot 224 \cdot 2 \cdot 1,25 = 1680 \text{ Дж/ч.}$$

Определяем избыточное явное выделяемое тепло

$$Q_{изб} = Q_1 + Q_2 + Q_3 + Q_4 = 756 + 768 + 720 + 1680 = 3924 \text{ Дж/ч.}$$

Количество воздуха, которое необходимо ввести в помещение для поглощения избытков тепла

$$L = \frac{3924 \text{ Дж/ч}}{1 \cdot 1 \text{ кг/м}^3 \cdot (27^\circ \text{C} - 20^\circ \text{C})} = 489 \text{ м}^3/\text{ч.}$$

По полученным расчетам необходимо выбрать кондиционер, обеспечивающий комфортные условия, т. е создающий и автоматически поддерживающий температуру, влажность, чистоту и скорость движения воздуха и отвечающий оптимальным санитарно – гигиеническим требованиям. В данном случае мы выбираем настенный кондиционер (сплит-система) SAMSUNG SH24TA6D.

Система ВIOnizer Чистый воздух - одна из основных составляющих нашего здоровья. Городской воздух, загрязненный пылью и выхлопными газами автомобилей, оказывает разрушающее воздействие на организм человека. Система очистки воздуха, используемая в кондиционерах Samsung, удаляет микрочастицы пыли и неприятные запахи с помощью фильтров и специального генератора анионов.

Антибактериальный биофильтр, обработанный уникальным антибактериальным составом, улавливает мельчайшие частицы пыли, содержащиеся в воздухе.

Т а б л и ц а 4.2 – Технические характеристики кондиционера

Название параметра	Значение
Мощность охлаждения:	6.8, кВт
Питание:	220-50-1, В-Гц-Фаз
Потребляемая мощность холод / тепло:	2100, Вт
Габариты внутр. блока (ШхВхГ):	880х638х310 мм
Площадь помещения	до 70, кв.м
Цена, в тенге	123 000

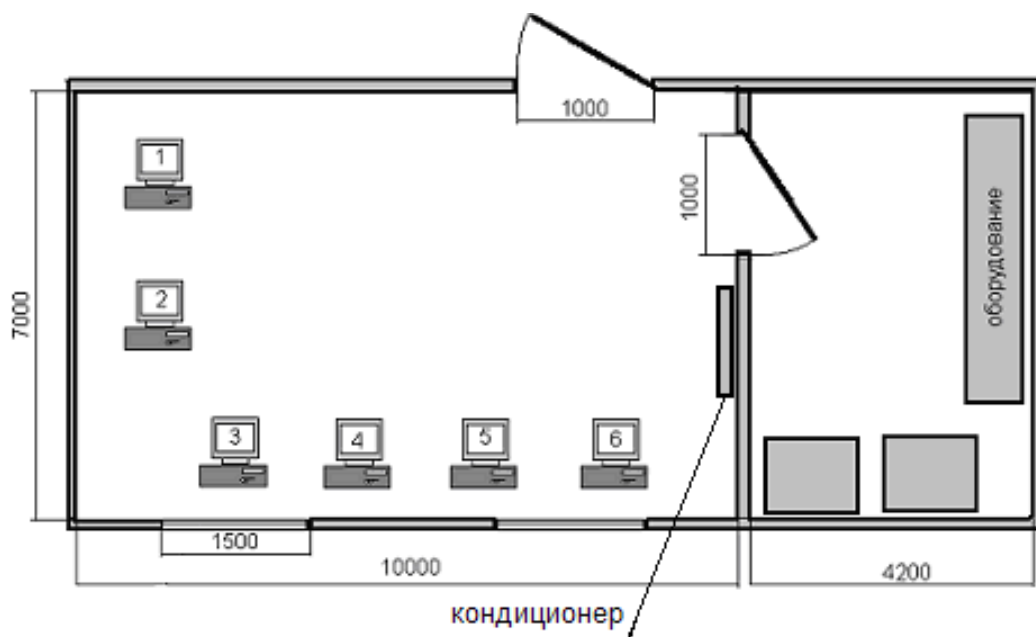


Рисунок 4.5 – Расположение кондиционера

4.1.5 Оценка пожарной безопасности

Огнетушители, стационарные и переносные средства пожаротушения будут периодически проверяться, испытываться и при необходимости перезаряжаться (огнетушители). Весь пожарный инвентарь, противопожарное оборудование и первичные средства пожаротушения будут содержаться в исправном состоянии, находиться на видном месте и к ним в любое время суток может быть обеспечен беспрепятственный доступ.

4.2 Выводы

В данной части дипломного проекта были рассчитаны такие разделы дисциплины «Безопасность жизнедеятельности», как «Производственное освещение», «Промышленная вентиляция, расчет воздухообмена в производственных помещениях». По результатам расчетов был выбран соответствующий всем требованиям кондиционер SAMSUNG SH24TA6D. А также был произведен анализ условий труда и производственного помещения в соответствии с требуемыми нормами по ПУЭ и СНиП.

Заключение

Программное обеспечение «Автоматизация биллинга электропотребления» является законченным программным продуктом, в дальнейшем возможна его доработка.

Программа имеет удобный пользовательский интерфейс, она предназначена для использования в операционной системе Windows XP/VISTA/7. Пользование программой не составит труда для тех, кто знаком с данной оболочкой.

Для дальнейшей доработки этой программы необходимо сначала в течение некоторого времени применять её для оценки эффективности, а потом добавлять или исправлять все недочеты, которые пропустил или не посчитал нужным реализовывать разработчик. При выполнении дипломного проекта усовершенствованы знания в области объектно-ориентированного программирования, разработке баз данных, приобретены навыки использования научно-технической и экономической информации.

В данном дипломном проекте разработано программное средство, автоматизирующее работу по учету электроэнергии.

Для выполнения данной задачи были проведены исследование существующих программ, определена их функциональность, выявлены недостатки, которые не позволяют использовать их для решения задачи Республике Казахстан.

Выполнена постановка задач на проектирование программного обеспечения, обоснование его архитектурных построений.

Разработка базы данных велась в среде Microsoft SQL Server 2008 и в Microsoft Office Visio 2007 (для получения схемы данных).

Произведена разработка реляционной базы данных с нормализацией таблиц.

Данное программное обеспечение может быть доведено до программного продукта после более детального тестирования у заказчика и разработки программной документации.

В случае разработки последующих версий программы можно добавить функциональность, новые формы и усовершенствовать существующую базу данных.

Список используемой литературы

- 1 Питер Роб, Карлос Коронел Системы баз данных: проектирование, реализация и управление, 5-е издание, – ВНУ Санкт-Петербург, 2004. -1040 с.
- 2 Дрозд А.С. Организация биллинговой системы. – 2006.
- 3 Ибраева Л.К., Сатимова Е.Г. «ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ», методические указания к выполнению лабораторных работ (для студентов всех специальностей). – Алматы: АИЭС, 2004 – 45 с.
- 4 Майкл Дж. Хернандес, Джон Л. Вьескас SQL-запросы для простых смертных.-Москва, 2000.
- 5 Безопасность жизнедеятельности. Методические указания к выполнению раздела «Расчет производственного освещения» в выпускных работах для всех специальностей. Бакалавриат. – А.: АИЭС, 2009. – 26 с.
- 6 СНиП РК 2.04-05-2002. Естественное и искусственное освещение. Министерство индустрии и торговли Республики Казахстан. – Астана, 2006. – 35 с.
- 7 СНиП РК 4.02-05-2001. Отопление, вентиляция и кондиционирование. Министерство индустрии и торговли Республики Казахстан. – Астана, 2001. – 37 с.
- 8 Инструкция по оформлению диссертации и автореферата. Изд. третье, дополненное и исправленное. - Введ. 2004-10-15. – Алматы: 2004. – 22 с.
- 9 Титов В.И. Экономика предприятия. – М.: Эксмо, 2008. – 354 с.
- 10 Новицкий Н.И. Основы менеджмента: Организация и планирование производства. – М.: Финансы и статистика, 2008. – 208 с.
- 11 Экономика и статистика предприятия. / Под ред. Ильенкова С.Д., Сиротиной Т.П. – Москва, 2009. – 72 с.
- 12 Базылов К.Б., Алибаева С.А., Бабич А.А. Методические указания по выполнению экономического раздела выпускной работы бакалавров для студентов всех форм обучения специальности 050719 – Радиотехника, электроника и телекоммуникации. – А.: АИЭС, 2008. –19 с.
- 13 <http://uchebilka.ru/buhgalteriya/6275/index.html?page=2>

Приложение А

Заполнение модуля биллинга базы данных

```
--Заполнение сущности "DISTRICTS"
INSERT INTO Districts VALUES ('Алатауский')
INSERT INTO Districts VALUES ('Алмалинский')
INSERT INTO Districts VALUES ('Ауэзовский')
INSERT INTO Districts VALUES ('Бостандыкский')
INSERT INTO Districts VALUES ('Жетысуский')
INSERT INTO Districts VALUES ('Медеуский')
INSERT INTO Districts VALUES ('Турксибский')

--Заполнение сущности "STREETS"
INSERT INTO Streets
VALUES ('Мкр.1'), ('Мкр.10'), ('Мкр.10А'), ('Мкр.11'),
('Мкр.12'), ('Мкр.2'), ('Мкр.3'), ('Мкр.4')
INSERT INTO Streets
VALUES ('Мкр.5'), ('Мкр.6'), ('Мкр.7'), ('Мкр.8'),
('Мкр.9'), ('Мкр.айгерим'), ('Мкр.айгерим-1')
INSERT INTO Streets VALUES ('Мкр.айгерим-
2'), ('Мкр.айнабулак'),
('Мкр.айнабулак 2'), ('Мкр.айнабулак3'),
('Мкр.айнабулак 4')
INSERT INTO Streets VALUES ('Мкр.акбулак'), ('Мкр.аксай
1А'),
('Мкр.аксай-1'), ('Мкр.аксай-2'), ('Мкр.аксай-
3'), ('Мкр.аксай-3А')
INSERT INTO Streets VALUES ('Мкр.аксай-
3Б'), ('Мкр.аксай4'),
('Мкр.алмагуль'), ('Мкр.алтай-1'), ('Мкр.алтай-2')
INSERT INTO Streets VALUES ('Мкр.алтын-
Бесик'), ('Мкр.астана'),
('Мкр.атырау-1'), ('Мкр.атырау-2'), ('Мкр.атырау-3')
INSERT INTO Streets VALUES ('Мкр.достык'), ('Мкр.думан-
1'),
('Мкр.жетысу-1'), ('Мкр.жетысу-2'), ('Мкр.жетысу-
3'), ('Мкр.жетысу-4')
INSERT INTO Streets VALUES ('Мкр.заря
Востока'), ('Мкр.казахфильм'),
('Мкр.кок-Тюбе'), ('Мкр.кок-Тюбе 2'), ('Мкр.кок-Тюбе-
2')
INSERT INTO Streets VALUES ('Мкр.коктем-
1'), ('Мкр.коктем-2'),
```

```

('Мкр.коктем-3'), ('Мкр.мамыр 1'), ('Мкр.мамыр
2'), ('Мкр.мамыр 3')
INSERT INTO Streets VALUES('Мкр.мамыр 4'), ('Мкр.мамыр
5'),
('Мкр.мамыр 7'), ('Мкр.орбита-1'), ('Мкр.орбита-
2'), ('Мкр.орбита-3')
INSERT INTO Streets VALUES('Мкр.орбита-
4'), ('Мкр.сайран'),
('Мкр.самал-1'), ('Мкр.самал-2'), ('Мкр.самал-
3'), ('Мкр.тастак-1')
INSERT INTO Streets VALUES('Мкр.тастак-2'), ('Мкр.тау
Самал'),
('Мкр.таугуль'), ('Мкр.таугуль-1'), ('Мкр.таугуль-
1, пос. мамыр')
INSERT INTO Streets VALUES('Мкр.таугуль-
2'), ('Мкр.таугуль-3'),
('Мкр.таугуль-Мжк'), ('Мкр.шанырак 1'), ('Мкр.шанырак
2')
--Улицы
INSERT INTO Streets VALUES('Ул.Молдагалиева
Ж. '), ('Ул.Тажигаевой П. '),
('Ул.Жубановой Г. '), ('Ул.Егизбаева К. ')
INSERT INTO Streets VALUES('Ул.Каратаева
М. '), ('Ул.Райымбека'),
('Ул.Жамакаева Б. '), ('Ул.Калдаякова Ш. ')
INSERT INTO Streets VALUES('Ул.Казыбаева
К. '), ('Ул.Сокпакбаева Б. '),
('Ул.Бокейханова А. '), ('Ул.Хожамьярова М. ')
INSERT INTO Streets VALUES('Ул.Тургутта
Озала'), ('Ул.Дуллати М. '),
('Ул.Жолдасбекова У. '), ('Ул.Жансугурова И. ')
INSERT INTO Streets VALUES('Ул.Бухар
жырау'), ('Ул.Карасай батыра'),
('Ул.Жибек Жолы'), ('Ул.Сулейменова Р. ')
INSERT INTO Streets VALUES('Ул.Наурызбай
батыра'), ('Ул.Улугбека'),
('Ул.Кудерина Ж. '), ('Ул.Кабанбай батыра')
INSERT INTO Streets VALUES('Ул.Кунаева
Д. '), ('Ул.Богенбай батыра'),
('Ул.Байкадамова Б. '), ('Ул.Каипова')
INSERT INTO Streets VALUES('Ул.Аблай Хана'), ('Ул.Толе
би'),
('Ул.Байтурсынулы А. '), ('Ул.Валиханова Ч. ')

```

Продолжение приложения А

```

INSERT INTO Streets VALUES ('Ул.Бейсебаева
М. '), ('Ул.Достык' ),
('Ул.Биянху' ), ('Ул.Гете' ), ('Ул.Утеген Батыра' )
INSERT INTO Streets
VALUES ('Ул.Желтоксан' ), ('Ул.Рыскулбекова К. '),
('Ул.Айтеке би' ), ('Ул.Макатаева' ), ('Ул.Алтынсарина
Ы. ')
INSERT INTO Streets
VALUES ('Ул.Брусиловского' ), ('Ул.Казыбек би' ),
('Ул.Тлендиева Н. '), ('Ул.Жумабаева М. '), ('Ул.Шашкина
З. ')
INSERT INTO Streets VALUES ('Ул.Токтабаева
К. '), ('Ул.Торайгырова С. '),
('Ул.Манаса' ), ('Ул.Таньшбаева М. ')

```

--Заполнение сущности "Addresses"

```

INSERT INTO Addresses VALUES (15,57,4,6) , (45,64,6,14)
INSERT INTO Addresses VALUES (5,14,6,4) , (76,74,5,7)
INSERT INTO Addresses VALUES (18,12,5,2) , (15,63,2,3)
INSERT INTO Addresses VALUES
(55,78,7,10) , (14,42,1,14)
INSERT INTO Addresses VALUES
(88,36,4,16) , (63,19,2,25)
INSERT INTO Addresses VALUES (96,87,3,25) , (9,31,3,16)
INSERT INTO Addresses VALUES
(13,69,2,14) , (18,16,4,20)
INSERT INTO Addresses VALUES
(27,17,1,18) , (90,75,7,12)
INSERT INTO Addresses VALUES (34,5,1,6) , (81,4,7,1)

```

--Заполнение "Subjects"

```

INSERT INTO Subjects VALUES
('Удостоверение' , '12.12.2008' , '12.01.2017' ,
'Министерство Юстиции РК' , 'Пенсионер' , 'Школа №27' )
INSERT INTO Subjects VALUES
('Паспорт' , '14.07.2001' , '12.09.2010' ,
'ГУВД Алматы' , 'Госслужащий' , 'Акимат г.Алматы' )
INSERT INTO Subjects VALUES
('Удостоверение' , '27.02.2007' , '05.04.2016' ,
'Паспортный стол' , 'Временно безработный' , NULL)
INSERT INTO Subjects VALUES
('Паспорт' , '06.08.2007' , '09.07.2015' ,
'г.Алматы' , 'Менеджер' , 'Турфирма "Айгуль"')
INSERT INTO Subjects VALUES
('Удостоверение' , '19.04.2005' , '17.16.2014' ,

```

```
'БТИ г.Алматы', 'Предприниматель', 'Фото-ателье')
INSERT INTO Subjects VALUES
('Паспорт', '07.09.2004', '09.06.2014', 'МВД
ПК', 'Военнослужащий', 'ВЧ 2091')
```

```
--Заполнение "CustomerTypes"
INSERT INTO CustomerTypes VALUES(1, 'Физические лица')
INSERT INTO CustomerTypes VALUES(2, 'Юридические
лица')
```

```
--Заполнение "CustomerTypes"
INSERT INTO Customers
VALUES(1001,1), (1002,1), (1003,1), (1004,1), (1005,1),
(1006,1), (1007,1), (1008,1), (1009,1), (1010,1)
INSERT INTO Customers
VALUES(1011,1), (1012,1), (1013,1), (1014,1), (1015,1),
(1016,1), (1017,1), (1018,1), (1019,1), (1020,1)
INSERT INTO Customers
VALUES(1021,1), (1022,1), (1023,1), (1024,1), (1025,1),
(1026,1), (1027,1), (1028,1), (1029,1), (1030,1)
INSERT INTO Customers
VALUES(1031,1), (1032,1), (1033,1), (1034,1), (1035,1),
(1036,1), (1037,1), (1038,1), (1039,1), (1040,1)
INSERT INTO Customers
VALUES(1041,1), (1042,1), (1043,1), (1044,1), (1045,1),
(1046,1), (1047,1), (1048,1), (1049,1), (1050,1)
INSERT INTO Customers
VALUES(1051,1), (1052,1), (1053,1), (1054,1), (1055,1),
(1056,1), (1057,1), (1058,1), (1059,1), (1060,1)
INSERT INTO Customers
VALUES(1061,1), (1062,1), (1063,1), (1064,1), (1065,1),
(1066,1), (1067,1), (1068,1), (1069,1), (1070,1)
INSERT INTO Customers
VALUES(1071,1), (1072,1), (1073,1), (1074,1), (1075,1),
(1076,1), (1077,1), (1078,1), (1079,1), (1080,1)
```

Продолжение приложения А

```
INSERT INTO Customers
VALUES(1081,1), (1082,1), (1083,1), (1084,1), (1085,1),
(1086,1), (1087,1), (1088,1), (1089,1), (1090,1)
INSERT INTO Customers
VALUES(1091,1), (1092,1), (1093,1), (1094,1), (1095,1),
(1096,1), (1097,1), (1098,1), (1099,1), (1100,1)
INSERT INTO Customers
VALUES(2001,2), (2002,2), (2003,2), (2004,2), (2005,2),
```



```

(2006,2) , (2007,2) , (2008,2) , (2009,2) , (2010,2)
INSERT INTO Customers
VALUES (2011,2) , (2012,2) , (2013,2) , (2014,2) , (2015,2)
, (2016,2) , (2017,2) , (2018,2) , (2019,2) , (2020,2)
INSERT INTO Customers
VALUES (2021,2) , (2022,2) , (2023,2) , (2024,2) , (2025,2) ,
(2026,2) , (2027,2) , (2028,2) , (2029,2) , (2030,2)
INSERT INTO Customers
VALUES (2031,2) , (2032,2) , (2033,2) , (2034,2) , (2035,2) ,
(2036,2) , (2037,2) , (2038,2) , (2039,2) , (2040,2)
INSERT INTO Customers
VALUES (2041,2) , (2042,2) , (2043,2) , (2044,2) , (2045,2) ,
(2046,2) , (2047,2) , (2048,2) , (2049,2) , (2050,2)
INSERT INTO Customers
VALUES (2051,2) , (2052,2) , (2053,2) , (2054,2) , (2055,2) ,
(2056,2) , (2057,2) , (2058,2) , (2059,2) , (2060,2)
INSERT INTO Customers
VALUES (2061,2) , (2062,2) , (2063,2) , (2064,2) , (2065,2) ,
(2066,2) , (2067,2) , (2068,2) , (2069,2) , (2070,2)
INSERT INTO Customers
VALUES (2071,2) , (2072,2) , (2073,2) , (2074,2) , (2075,2) ,
(2076,2) , (2077,2) , (2078,2) , (2079,2) , (2080,2)
INSERT INTO Customers
VALUES (2081,2) , (2082,2) , (2083,2) , (2084,2) , (2085,2) ,
(2086,2) , (2087,2) , (2088,2) , (2089,2) , (2090,2)
INSERT INTO Customers
VALUES (2091,2) , (2092,2) , (2093,2) , (2094,2) , (2095,2) ,
(2096,2) , (2097,2) , (2098,2) , (2099,2) , (2200,2)

```

--Заполнение сущности "Persons"----

```

INSERT INTO Persons
VALUES (1001, 'Шивазова', 'Майя', 'Кемаровна', 'Женский',
'26-08-1992', '8(771)413-57-
88', 'mai92@mail.ru', '145558966336',
'94555977337', '775869633', 3, 1, '05.05.2014')
INSERT INTO Persons
VALUES (1002, 'Рындина', 'Анастасия', 'Андреевна',
'Женский', '28-11-1991', '8(702)454-454-
2', '12@mail.ru', '478556933325',
'47557933325', '010958869', 4, 3, '05.05.2014')
INSERT INTO Persons
VALUES (1003, 'Федоров', 'Андрей', 'Викторович',
'Мужской', '07-08-1989', '8(705)44-22-
388', 'fd07@mail.ru', '010507889634',
'19151797341', '232388896', 6, 6, '05.05.2014')

```

```

INSERT INTO Persons
VALUES (1004, 'Минин', 'Сергей', 'Павлович',
'Mужской', '15-12-1986', '8 (771) 423-07-
07', '15ser@mail.ru', '000245689666',
'11124579777', '447896355', 5, 2, '05.05.2014')
INSERT INTO Persons
VALUES (1005, 'Мельник', 'Анна', 'Денисовна', 'Женский',
'14-02-1987', '8 (705) 123-123-
4', 'ann5@mail.ru', '444888963567',
'44497357711', '114799666', 4, 9, '05.05.2014')
INSERT INTO Persons
VALUES (1006, 'Кусаинов', 'Руслан', 'Рустамович',
'Mужской', '02-07-1982', '8 (707) 166-45-
86', 'rus1982@mail.ru',
'115558963599', '99555973599', '144563336', 2, 10, '05.05.
2014')
INSERT INTO Persons
VALUES (1007, 'Хайрулина', 'Альбина', 'Сергеевна',
'Женский', '05-04-1984', '8 (701) 277-55-
69', 'al84@mail.ru', '145593326485',
'94559332745', '010305896', 3, 8, '05.05.2014')
INSERT INTO Persons
VALUES (1008, 'Алиев', 'Нурсултан', 'Диасович',
'Mужской', '12-11-1982', '8 (701) 731-56-
17', 'nur12@mail.ru', '148779960023',
'94779971123', '040478951', 1, 5, '05.05.2014')
INSERT INTO Persons
VALUES (1009, 'Затеев', 'Николай', 'Владимирович',
'Mужской', '30-05-1981', '8 (701) 224-76-
76', 'nik05@mail.ru', '147936665896',
'94793777597', '040495180', 3, 7, '05.05.2014')
INSERT INTO Persons
VALUES (1010, 'Бондарев', 'Максим', 'Денисович',
'Mужской', '24-09-1985', '8 (777) 721-37-
38', 'max85@mail.ru', '477788020608',

```

Продолжение приложения А

```

'47771217111', '040445951', 1, 4, '05.05.2014')
--Заполнение сущности "Users"
INSERT INTO Users VALUES
('Admin', 12, GETDATE(), '00:00', '23:59', 1001)
INSERT INTO Users VALUES
('bonmax', 'rr', GETDATE(), '00:00', '23:59', 1010)

```

```

INSERT INTO Users VALUES
('nastya',555,GETDATE(),'00:00','23:59',1002)
INSERT INTO Users VALUES
('rus','rus',GETDATE(),'00:00','23:59',1006)
INSERT INTO Users VALUES
('sergey',1986,GETDATE(),'00:00','23:59',1004)
INSERT INTO Users VALUES
('ali',1982,GETDATE(),'00:00','23:59',1008)

```

--Заполнение сущности "Additionalс"

```

INSERT INTO Additionalс VALUES('Нефинансовые
корпорации','СЕЛЬСКОЕ, ЛЕСНОЕ И РЫБНОЕ ХОЗЯЙСТВО','АО
"НУР"')
INSERT INTO Additionalс VALUES('Финансовые
корпорации','СТРОИТЕЛЬСТВО',
'ТОО "Регион"')
INSERT INTO Additionalс VALUES('Органы
государственного управления',
'ТРАНСПОРТ И СКЛАДИРОВАНИЕ','ГУ')
INSERT INTO Additionalс VALUES('Некоммерческие
организации, обслуживающие домашние хозяйства','СЕЛЬСКОЕ,
ЛЕСНОЕ И РЫБНОЕ ХОЗЯЙСТВО','ИП Федоров')
INSERT INTO Additionalс VALUES('Домашние
хозяйства','СЕЛЬСКОЕ, ЛЕСНОЕ И РЫБНОЕ ХОЗЯЙСТВО','ИП
Оксана')

```

--Заполнение сущности "Organization"

```

INSERT INTO Organizations VALUES(2001,'ТОО
"ЭНЕРГОПРОМТЕХНО",'ТОО
"ЭНЕРГОПРОМТЕХНО",'145558966336','94555977337',
'12','154','1145266',
'24-07-2013','8(771)413-57-
88','maiу92@mail.ru',1001,'05.05.2014',1,3)
INSERT INTO Organizations VALUES(2002,'ГУ "АППАРАТ
АКИМА АКТОГАЙСКОГО РАЙОНА",'ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
"АППАРАТ
АКИМА
АКТОГАЙСКОГО
РАЙОНА",'478556933325','47557933325','16','458','1123669
',
'24-07-2013','8(702)454-454-
2','12@mail.ru',1002,'05.05.2014',10,5)
INSERT INTO Organizations VALUES(2003,'ТОО
"ЛСИ",'Товарищество с ограниченной ответственностью
"ЛСИ",'010507889634','19151797341','12',
'148','0335657','24-07-2013','8(705)44-22-
388','fd07@mail.ru',1004,

```

```

'05.05.2014',8,4)
INSERT INTO Organizations VALUES(2004,'ТОО
"Дейкар"', 'ТОО "Дейкар"',
'000245689666', '11124579777', '12', '256', '2323587', '24
-07-2013',
'8(771)423-07-
07', '15ser@mail.ru', 1003, '05.05.2014', 5, 2)
INSERT INTO Organizations VALUES(2005,'ТОО "ШИК-
ДИЗАЙН"',
'ТОО "ШИК-
ДИЗАЙН"', '444888963567', '44497357711', '12', '118', '2125432
',
'24-07-2013', '8(705)123-123-
4', 'ann5@mail.ru', 1005, '05.05.2014', 2, 4)
INSERT INTO Organizations VALUES(2006,'ЗАО
"ОТПАР"', 'ЗАО
"ОТПАР"', '115558963599', '99555973599', '12', '301', '1515615
', '24-07-2013',
'8(707)166-45-
86', 'rus1982@mail.ru', 1006, '05.05.2014', 11, 3)
INSERT INTO Organizations VALUES(2007,'ПКСТ
"НЕФТЯНИК" ОФИС',
'Потребительский кооператив садоводческого
товарищества "Нефтяник"',
'148779960023', '94779971123', '12', '551', '1545465', '24
-07-2013',
'8(777)721-37-
38', 'max85@mail.ru', 1002, '05.05.2014', 3, 5)
INSERT INTO Organizations VALUES(2008,'ПКСК "НАШ
ДОМ"', 'ПКСК "НАШ ДОМ"',
'145593326485
', '94559332745', '12', '504', '2151561', '24-07-2013',
'8(701)277-55-69', 'al84@mail.ru', 1009, '05.05.2014', 4, 2)

INSERT INTO Organizations VALUES(2009,'ТОО ФИРМА
"ТДВ"', 'ТОО ФИРМА ТДВ"',
'147936665896', '94793777597', '12', '277', '4578764', '24
-07-2013', '8(701)731-
56-17', 'nur12@mail.ru', 1007, '05.05.2014', 8, 2)
INSERT INTO Organizations VALUES(2010,'ТОО
"НУРАЛ"', 'ТОО "НУРАЛ"',

```

Продолжение приложения А

'477788020608', '47771217111', '12', '721', '1000556', '24
-07-2013', '8(701)224-
76-76', 'nik05@mail.ru', 1008, '05.05.2014', 6, 1)

--Заполнение сущности "Tariffs"-----

INSERT INTO Tariffs VALUES ('Стандартный', '12,95')

--Заполнение сущности "Agreements"

INSERT INTO Agreements
VALUES ('124565', '12.12.2013', '12.12.2014',
'Ведущий инженер', 'Устава', 'Иванов
С.Г.', 'Промышленность',
'Энергоснабжение', 1, 1001, 2, '12.12.2013')

INSERT INTO Agreements
VALUES ('124566', '13.12.2013', '13.12.2014',
'Директор', 'Устава', 'Даулетбаков
Г.К.', 'Непромышленность',
'Энергоснабжение', 1, 1003, 6, '13.12.2013')

INSERT INTO Agreements
VALUES ('124568', '01.02.2014', '01.02.2015',
'Индивидуальный предприниматель', 'Устава', 'Киселева
К.В.',
'Сельское хоз-
во', 'Энергоснабжение', 1, 2004, 4, '01.02.2014')

INSERT INTO Agreements
VALUES ('124634', '14.02.2014', '14.02.2016',
'Заместитель директора', 'Устава', 'Селиванова Л.Ф.',
'ГосУчереждение (Бюджет)', 'Энергоснабжение', 1, 2005, 12,
'14.02.2014')

INSERT INTO Agreements
VALUES ('124633', '16.02.2014', '16.02.2015', NULL,
'Устава', 'Максутова
А.А.', 'Население', 'Энергоснабжение', 1, 1008, 9,
'16.02.2014')

INSERT INTO Agreements
VALUES ('124479', '15.03.2014', '15.03.2017',
'Начальник', 'Устава', 'Зайтов С.Д.', 'ГосПредприятие
(Бюджет)',
'Энергоснабжение', 1, 1002, 10, '15.03.2014')

INSERT INTO Agreements
VALUES ('124433', '05.04.2014', '05.04.2015',
'Частный предприниматель', 'Устава', 'Валов
Р.В.', 'Оптовый покупатель',
'Энергоснабжение', 1, 2007, 7, '05.04.2014')

```

--Заполнение сущности "Periods"
INSERT INTO Periods VALUES('Январь 2014', '2014-01-01', '2014-04-20', 2014,1)
INSERT INTO Periods VALUES('Февраль 2014', '2014-02-01', '2014-04-20', 2014,2)
INSERT INTO Periods VALUES('Март 2014', '2014-01-26', '2014-04-20', 2014,3)
INSERT INTO Periods VALUES('Апрель 2014', '2014-01-26', '2014-04-20', 2014,4)
INSERT INTO Periods VALUES('Май 2014', '2014-01-26', '2014-04-20', 2014,5)
INSERT INTO Periods VALUES('Июнь 2014', '2014-01-26', '2014-04-20', 2014,6)
INSERT INTO Periods VALUES('Июль 2014', '2014-01-26', '2014-04-20', 2014,7)
INSERT INTO Periods VALUES('Август 2014', '2014-01-26', '2014-04-20', 2014,8)
INSERT INTO Periods VALUES('Сентябрь 2014', '2014-01-26', '2014-04-20', 2014,9)
INSERT INTO Periods VALUES('Октябрь 2014', '2014-01-26', '2014-04-20', 2014,10)
INSERT INTO Periods VALUES('Ноябрь 2014', '2014-01-26', '2014-04-20', 2014,11)
INSERT INTO Periods VALUES('Декабрь 2014', '2014-01-26', '2014-04-20', 2014,12)

```

```

--Заполнение сущности "Accruals"
INSERT INTO Accruals
VALUES('12.02.2013', '10.03.2013', '100', 1295, 12, 155.40, 1450, 402, 1)

```

Продолжение приложения А

```

INSERT INTO Accruals
VALUES('12.02.2013', '10.03.2013', '150', 1942, 12, 233.04, 2175, 404, 1)

```

```

INSERT INTO Accruals
VALUES ('15.02.2013', '13.03.2013', '50', 647, 12,
77.64, 724, 405, 2)
INSERT INTO Accruals
VALUES ('17.02.2013', '14.03.2013', '90', 1165, 12,
139.80, 1304, 407, 2)
INSERT INTO Accruals
VALUES ('18.02.2013', '15.03.2013', '75', 971, 12,
166.52, 1087, 406, 3)
INSERT INTO Accruals
VALUES ('19.02.2013', '16.03.2013', '120', 1554, 12,
186.48, 1740, 401, 3)

```

--Заполнение сущности "DealBalances"

```

INSERT INTO DealBalances VALUES (100, 1450, 0, 201)
INSERT INTO DealBalances VALUES (150, 2175, 0, 203)
INSERT INTO DealBalances VALUES (50, 724, 0, 206)
INSERT INTO DealBalances VALUES (90, 1304, 0, 204)
INSERT INTO DealBalances VALUES (75, 1087, 0, 205)

```