

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра _____

«Допущен к защите»
Заведующий кафедрой ЖТ
Журалибаев З.К., д.ф.н., проф.
(Ф.И.О., ученая степень, звание)
Жу — «30» мая 2014г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Информационная система «Моделирование торгов рынка
электроэнергии»

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Курбский В.А. ВТ-10-4
(Фамилия и инициалы) группа

Научный руководитель Бимурзаев С.Б. д.ф.н.-м.н. профессор СМ
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:

Бреснева З.Я., с.н.с. преподаватель
(Фамилия и инициалы, ученая степень, звание)
Бреснева «14» 05 2014 г.
(подпись)

по безопасности жизнедеятельности:

Дрижадов И.Г., д.х.н. профессор
(Фамилия и инициалы, ученая степень, звание)
Дрижадов «19» 05 2014 г.
(подпись)

по применению вычислительной техники:

Бимурзаев С.Б. д.ф.н.-м.н. профессор СМ
(Фамилия и инициалы, ученая степень, звание)
«07» мая 2014 г.
(подпись)

(Фамилия и инициалы, ученая степень, звание)
« _____ » _____ 20 _____ г.
(подпись)

Нормоконтролер: Тусупов Д.Н. ДН
(Фамилия и инициалы, ученая степень, звание)
«28» мая 2014 г.
(подпись)

Рецензент: _____
(Фамилия и инициалы, ученая степень, звание)
« _____ » _____ 20 _____ г.
(подпись)

Алматы 2014 г.

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

Студент Курбский Вадим Алексеевич
(фамилия, имя, отчество)

Тема проекта Информационная система «Моделирование рынка

утверждена приказом ректора № 115 от «24» сентября 2013 г.

Срок сдачи законченной работы «7» июля 2014 г.

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

Рынок централизованных торгов электроэнергией
Методические указания по проведению централизованных торгов
электрической энергией на среднесрочный и долгосрочный периоды
Методические указания по проведению централизованных
spot-торгов электрической энергией в режиме «за день вперёд»

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

- 1) Анализ современного рынка торгов электроэнергией
- 2) Анализ предметной области
- 3) Проектирование БД
- 4) Создание приложения
- 5) Безопасность и надежность
- 6) Расчёт цены программной продукции

Перечень графического материала (с точным указанием обязательных чертежей)

- Рисунки для Бизнес плана
- Таблицы с описанием отношений
- Рисунки с оформлением приложений
- Рисунки для БЖД раздела
- Таблицы для Бизнес плана
- Рисунки для Бизнес плана

Рекомендуемая основная литература

- 1) «Методические указания по проведению централизованных торгов электрической энергией на средне-срочный и долгосрочный периоды»
- 2) «Методические указания по проведению централизованных спот-торгов электрической энергией в режиме на день вперед»
- 3) Т. Кейт «Oracle для профессионалов»
- 4) Д. Ошнов «База данных и Delphi»

Консультанты по проекту с указанием относящихся к ним разделов

| Раздел | Консультант | Сроки | Подпись |
|---------------|----------------|------------------|------------------|
| БЖД | Дмитриева Н.Г. | 11.04 - 19.05.14 | <i>[Подпись]</i> |
| Эр. Экономика | Брашнев В.Д. | 11.04 - 14.05.14 | <i>[Подпись]</i> |
| Контроль | Пучков Д.М. | 28.05.14г. | <i>[Подпись]</i> |
| | | | |
| | | | |
| | | | |
| | | | |

Г Р А Ф И К
подготовки дипломного проекта

| № п/п | Наименование разделов, перечень разрабатываемых вопросов | Сроки представления руководителю | Примечание |
|-------|--|----------------------------------|------------|
| 1 | Сбор необходимого материала | 01.04.14 | |
| 2 | Обследование объекта и его описание | 07.04.14 | |
| 3 | Построение предварительных схем (блоков) учебного процесса | 14.04.14 | |
| 4 | Разработка базы данных проекта | 21.04.14 | |
| 5 | Разработка алгоритмов и программ | 28.04.14 | |
| 6 | Создание приложения на Delphi | 01.05.14 | |
| 7 | Отладка и испытание программ | 07.05.14 | |
| 8 | Технико-экономическое обоснование | 14.05.14 | |
| 9 | Безопасность жизнедеятельности | 18.05.14 | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Дата выдачи задания « 03 » марта 2014 г.

Заведующий кафедрой _____ Куралбаев З. К.
(подпись) (Фамилия и инициалы)

Руководитель _____ Бимурзаев С. Б.
(подпись) (Фамилия и инициалы)

Задание принял к исполнению студент _____ Курбский В. А.
(подпись) (Фамилия и инициалы)

Аннотация

Электроэнергетика является важной отраслью Республики Казахстан. Казахстан обладает крупными запасами энергетических ресурсов.

Электрическая энергия является товаром, который приобретают участники оптового рынка у генерирующих компаний, а участники розничного рынка у энергосбытовых компаний.

Информационная система “Моделирование торгов рынка электроэнергии”, разработанная в данном дипломном проекте, поможет студентам более наглядно рассмотреть рынок торгов электроэнергией.

Андатпа

Электр энергетика Қазақстан Республикасының маңызды саласы болып табылады. Қазақстан қ энергетикалық ресурстардың ірі қорына ие мемлекеттердің бірі.

Электр энергия өндіруші компанияларда көтерме нарығына қатысушылар, сондай-ақ энергия үнемдеу компанияларында бөлшек нарығына қатысушылар қолданатын тауар болып табылады.

Осы диплом жабасында әзірленген «Электрэнергия нарығын саудалауды модельдеу» ақпараттық жүйесі студенттердің электрэнергия нарығы саудасын жетік меңгеруіне бағытталға.

Annotation

The power industry is an important sector of the Republic of Kazakhstan. Kazakhstan has large reserves of energy.

Electrical energy is a commodity that acquire the wholesale market from generation companies and retail market participants in energy supply companies.

Information System "Modeling trades electricity market" developed in this thesis project would help students more clearly see the market trading of electricity.

Содержание

| | |
|--|-----|
| Введение..... | 13 |
| 1 Концептуальное проектирование | 15 |
| 1.1 Современный рынок электроэнергии РК | 15 |
| 1.2 Анализ предметной области | 16 |
| 1.3 Инструментальные средства используемые при разработке | 17 |
| 1.4 Проектирование БД | 19 |
| 1.4.1 Разработка и построение подробной Er–диаграммы на основании бизнес правил. | 19 |
| 1.4.2 Определение атрибутов каждой сущности | 22 |
| 1.4.3 Составление реляционных отношений | 25 |
| 1.4.4 Расчет места для хранения БД..... | 28 |
| 2 Реализация приложения | 34 |
| 2.1 Физическое проектирование БД..... | 34 |
| 2.1.1 Создание табличного пространства..... | 34 |
| 2.1.2 Создание таблиц | 34 |
| 2.1.3 Создание реляционных отношений | 37 |
| 2.1.4 Создание последовательностей..... | 40 |
| 2.1.5 Заполнение созданной БД..... | 41 |
| 2.1.5 Представления базы данных..... | 47 |
| 2.1.6 Функции Процедуры | 48 |
| 2.1.7 Триггеры | 69 |
| 2.2 Описание интерфейса приложения. Работа с приложением | 70 |
| 3 Безопасность жизнедеятельности..... | 74 |
| 3.1 Анализ условий труда обслуживающего персонала при эксплуатации технического оборудования..... | 74 |
| 3.1.1 Магнитное поле. | 74 |
| 3.1.2 Шум..... | 75 |
| 3.1.3 В чем заключается вред работы ночью | 76 |
| 3.1.4 Технический персонал | 76 |
| 3.2 Организация рабочих мест..... | 77 |
| 3.3 Расчет системы искусственного освещения помещения | 82 |
| Выводы..... | 84 |
| 4 Бизнес план | 85 |
| 4.1 Описание работы и обоснование необходимости | 85 |
| 4.2 Расчет затрат на разработку информационных технологий..... | 86 |
| 4.3 Расчет цены программного продукта | 93 |
| Выводы..... | 95 |
| Заключение | 96 |
| Список литературы | 97 |
| Приложение А | 98 |
| Приложение Б..... | 100 |

| | |
|--------------------|-----|
| Приложение В | 103 |
| Приложение Г | 105 |
| Приложение Д | 106 |

Введение

Цель дипломного проекта: разработка информационной системы (ИС) “Моделирование торгов рынка электроэнергии”. Данная ИС должна эмулировать рынок торгов электроэнергии и будет использоваться в учебных целях. Для решения поставленной цели необходимо решить следующие задачи:

- изучить способы проведения торгов на рынке электроэнергии;
- исследовать принципы охраны труда, техники безопасности;
- рассчитать техникоэкономическое обоснование эффективности разработки ИС;
- разработать ИС “Моделирование торгов рынка электроэнергии”.

Оператором рынка централизованных торгов электроэнергии Республики Казахстан является АО “Казахстанский оператор рынка электрической энергии и мощности” (КОРЭМ. В соответствии с пп. 2 ст. 15 Закона РК «Об электроэнергетике») [1]. Оператор рынка должен осуществлять следующие функции:

- организация и проведение спот–торгов;
- организовать проведение централизованной торговли электрической энергией на среднесрочный (неделя, месяц) и долгосрочный (квартал, год) периоды;
- обеспечение равных условий доступа участникам оптового рынка электрической энергии на рынок централизованной торговли электрической энергией;
- определение соответствия субъектов оптового рынка электрической энергии требованиям, установленным правилами централизованной торговли электрической энергией;
- осуществление регистрации и учета заключенных сделок по купле–продаже электрической энергии на централизованных торгах электрической энергией;
- обеспечение в пределах своей компетенции субъектов оптового рынка электрической энергии информацией по индикативным ценам на электрическую энергию, сложившимся на централизованных торгах, и другой рыночной информацией.

Из-за физических свойств электроэнергия обладает особенностями, которые необходимо учитывать при организации рынка:

- процессы производства и потребления электроэнергии должны происходить одновременно, объема выработанной и потреблённой электроэнергии в каждый момент времени должны быть одинаковыми;
- электроэнергию невозможно запасать в количествах сопоставимых с масштабами энергосистемы;
- невозможно заранее точно оговорить объемы электроэнергии, которые будут произведены или потреблены;

– невозможно с физической точки зрения определить, кто произвёл электроэнергию, использованную тем или иным потребителем.

Рынки любой другой товарной продукции при кратковременном дисбалансе между производством и потреблением способны выстоять, потому как могут быть складские запасы или товары-заменители. Рынок электроэнергии же может нормально функционировать только при условии, что в каждый момент времени обеспечивается баланс производства и потребления.

На практике производители и потребители электроэнергии могут отклоняться от своих обязательств по производству и потреблению электроэнергии. Существуют краткосрочные (в пределах часа, суток и т. п.) и сезонные (в течение года) колебания нагрузки, так же располагаемая мощность электростанций с требуемым резервом может превышать величину годового максимума нагрузки. Эти факторы приводят к тому, что в течение года некоторый объем генерирующей мощности недогружен. Диспетчер, управляющий работой энергосистемы, не способен регулировать в режиме реального времени отпуск электроэнергии потребителям в соответствии с договорами на поставку, да и потребитель может отбирать электроэнергию с существенными отклонениями от договорных обязательств. Поэтому есть необходимость в наличии определённого количества маневренных электростанций для оперативного балансирования энергосистемы в условиях переменной нагрузки, способных быстро и в широких пределах менять величину выработки электроэнергии.

Невозможность создания запасов готовой продукции приводит к необходимости создания резервов генерирующих мощностей, пропускной способности электрических сетей и запасов топлива на электростанциях. Величина резервов нормируется, а затраты на поддержание резервов включаются в стоимость электроэнергии.

1 Концептуальное проектирование

1.1 Современный рынок электроэнергии РК

Концепция развития рыночных отношений в электроэнергетике Республики Казахстан прежде всего направлена на закрепление и развитие принципа разделения между участниками энергетического рынка следующих функций:

- производство электрической энергии;
- передача и распределение электрической энергии;
- поставка (сбыт) электрической энергии конечным потребителям.

Концепция предусматривает четкое разграничение двух уровней энергетической системы Казахстана: оптового и розничного рынка электроэнергии. Новая модель построения энергосистемы получила свое закрепление и на законодательном уровне. Так, были приняты Закон "Об электроэнергетике" от 9 июля 2004 года; Правила организации и функционирования оптового рынка электрической энергии Республики Казахстан; Правила организации и функционирования розничного рынка электрической энергии, а также доступа и предоставления услуг на данном рынке [1].

Во исполнение Закона Республики Казахстан «Об электроэнергетике» (далее – Закон) с 1 января 2008 года в Республике Казахстан начал функционировать балансирующий рынок электрической энергии [1].

В соответствии с п. 6 ст. 13 Закона, а также п. 7 гл. 1 «Правил функционирования балансирующего рынка электрической энергии», каждый субъект оптового рынка электрической энергии ЕЭС Казахстана обязан участвовать в балансирующем рынке электрической энергии.

Модель оптового рынка электрической энергии состоит из следующих сегментов [2]:

1 Децентрализованный рынок. Здесь участники оптового рынка (покупатели и продавцы электроэнергии) заключают между собой прямые двусторонние договоры купли–продажи.

2 Централизованный рынок представляет собой своего рода биржу, на которой участники продают и покупают электроэнергию. Основным предметом торгов на этом рынке являются договоры на поставку "за день вперед" (спот-рынок), а также среднесрочные и долгосрочные контракты на поставку энергии (форвардные контракты). На момент принятия Концепции объемы спот-торгов занимали только 1% от общего количества заключаемых контрактов. Все остальное – прямые двусторонние договоры купли-продажи.

3 Балансирующий рынок электроэнергии в режиме "реального времени" выполняет функции по физическому урегулированию возникающих дисбалансов между договорными и фактическими величинами перетоков электроэнергии. Системный оператор (KEGOC) устраняет возникающие

дисбалансы за счет использования резервных мощностей. Для этого государственными органами и КЕГОС будут определены конкретные электростанции, на которых размещаются резервы мощности. Участник рынка, допустивший превышение договорного объема потребления или снижение выработки электроэнергии, должен оплатить услуги системного оператора по урегулированию возникающих дисбалансов.

4 Рынок системных и вспомогательных услуг. Основным продавцом/покупателем на этом рынке выступает системный оператор – КЕГОС. В качестве продавца он оказывает всем участникам оптового рынка услуги, аналогичные услугам региональных электросетевых компаний на розничном рынке. К ним относится передача электрической энергии по сетям Национальной энергетической системы (220–500–1150 кВ); техническая диспетчеризация отпуска в сеть и потребления электрической энергии; регулирование мощности в процессе передачи электрической энергии и диспетчеризации. Все вышеуказанные услуги отнесены законодательством Казахстана к сфере естественной монополии. Тарифы на них предварительно согласовываются с антимонопольными органами и являются едиными для всех потребителей. В качестве покупателя КЕГОС приобретает у участников оптового рынка услуги по готовности необходимых объемов резервов электрической мощности (энергопроизводители), а также услуги по регулированию активной и реактивной мощности для обеспечения стандартов надежности и качества электроэнергии (РЭКи).

1.2 Анализ предметной области

Основной объект предметной области – участник рынка, у участника рынка есть учётные данные пользователя. Обычный участник рынка – покупатель или продавец, особый участник рынка – оператор. За участником рынка закрепляется предприятие, закрепляет предприятие оператор. Так же у каждого участника рынка есть счёт в банке.

Все предприятия связаны схемой линии электропередач. Предприятия бывают разных видов (ГЭС, ТЭЦ, РЭК, ЛЭП, покупатель...). Каждый вид предприятия выполняет одну из трёх ролей: поставщик электроэнергии (электростанция), потребитель электроэнергии (предприятие), посредник электроэнергии (распределительная компания, роль которую могут получить только игроки с особыми привилегиями).

Поставщик может подать заявку на продажу определённого количества электроэнергии по определённому тарифу, покупатель на покупку электроэнергии.

Заявкам независимо от их вида и статуса присваивается номер в порядке возрастания номеров подаваемых заявок без прерывания числовой последовательности [3].

Заявка должна содержать следующие данные:

1 идентификационный номер;

2 краткое наименование организации участника централизованных торгов, присвоенное оператором рынка централизованной торговли электрической энергией и указанное в регистрационной карте;

3 вид торгов;

4 вид заявки;

5 статус заявки;

6 период, в который предполагается осуществлять поставку/ потребление проданной/купленной на торгах электрической энергии;

7 почасовой объем электрической энергии, заявляемый на покупку/продажу;

8 почасовые цены покупки/продажи электрической энергии (без учета НДС).

Почасовой объем электрической энергии, заявляемый участником централизованных торгов на покупку/продажу, должен быть не менее 1 тыс.кВт*ч и кратен 1 тыс.кВт*ч.

Все поступившие заявки регистрируются в базе данных торговой системы в порядке очередности их поступления с фиксацией времени поступления заявки. Зарегистрированная в базе данных торговой системы заявка участника централизованных торгов проходит проверку на правомочность ее участия в централизованных торгах. Для допуска заявки к торгам она проверяется на своевременность поступления.

В краткосрочном рынке торги проводятся в форме двустороннего аукциона. В долгосрочном рынке участники могут покупать заявки.

В балансирующем рынке в определенное время заявки перестают добавляться, из заявок участников торгов на покупку и продажу по каждому часу суток составляются ранжированные графики спроса и предложения электрической энергии [4]. При этом:

1 заявки на покупку ранжируются в порядке убывания указанных в заявках цен, а при равных ценах – в порядке очередности их поступления в торговую систему. При отборе заявок для заключения сделок приоритет отдается заявкам с наивысшей ценой, а при равных ценах – заявкам, поступившим раньше;

2 заявки на продажу ранжируются в порядке возрастания указанных в заявках цен, а при равных ценах – в порядке очередности их поступления в торговую систему. При отборе заявок для заключения сделок приоритет отдается заявкам с наименьшей ценой, а при равных ценах – заявкам, поступившим раньше.

Обычных участников рынка представляют студенты, оператора (системный администратор) представляет преподаватель.

1.3 Инструментальные средства используемые при разработке

Для хранения данных и взаимодействия АРМов друг с другом выбрана СУБД Oracle Database 11g Express Edition (Oracle XE 11g) –

высокопроизводительным, масштабируемым, защищенным и надежным решением для кластерных сред или инфраструктур с одним сервером на платформах Windows, Linux и UNIX. Oracle XE 11g предлагает всеобъемлющие возможности для простого управления комплексными процессами передач, бизнес-аналитикой и контентом.

Oracle XE 11g защищает от поломок серверов, сбоев на сайтах, ошибок пользователей и уменьшает время вынужденного простоя. Редакция помогает соответствовать нормативно-правовым документам. Благодаря Oracle XE 11g администраторы могут управлять всем жизненным циклом информации в больших базах данных [5].

Oracle XE 11g предоставляет мощные, знакомые инструменты для профессионалов информационных технологий так же, как и для работников информационной сферы, уменьшая сложность создания, развёртывания, управления и использования данных предприятия и аналитических приложений на платформах от мобильных устройств до информационных систем предприятия. Благодаря исчерпывающему набору функций, взаимодействию с существующими системами и автоматизации типовых задач, Oracle XE 11g предоставляет полное решение в области хранения данных для предприятий всех масштабов.

Для разработки приложения выбрана интегрированная среда разработки Delphi XE2. Среда предназначена для быстрой (RAD) разработки прикладного ПО для операционных систем Windows, Mac OS X, а также IOS и Android. Благодаря уникальной совокупности простоты языка и генерации машинного кода, позволяет непосредственно, и, при желании, достаточно низкоуровнево взаимодействовать с операционной системой, а также с библиотеками, написанными на C/C++ [6]. Созданные программы не зависимы от стороннего ПО, как-то Microsoft .NET Framework, или Java Virtual Machine. Выделение и освобождение памяти контролируется в основном пользовательским кодом, что, с одной стороны, ужесточает требования к качеству кода, а с другой – делает возможным создание сложных приложений, с высокими требованиями к отзывчивости (работа в реальном времени). В кросс-компиляторах для мобильных платформ предусмотрен автоматический подсчет ссылок на объекты, облегчающий задачу управления их временем жизни.

Для получения данных с сервера базы данных использованы Oracle client и набор компонент Direct Oracle Access. Набор компонент прямого доступа Direct Oracle Access помогает организовать идеальную интеграцию инструментов разработки и базы данных [7]. Кроме того, что приложения смогут получить максимальный выигрыш от использования инструментов разработки и исполнения приложений.

Приложения пакетной обработки и обслуживания транзакций смогут использовать всю мощь платформы Oracle. Компоненты прямого доступа Direct Oracle Access, как и следует из названия, обеспечивают прямой доступ к интерфейсу Oracle Interface. Это гарантирует оптимальную производительность при использовании стандартных функций доступа к базе данных. Поддержка

функций Array Fetching, Array DML, блоков PL/SQL, таблиц PL/SQL, локального кэширования выражений Local Statement Caching и механизма прямой загрузки Direct Path Load Engine обеспечивает дополнительную оптимизацию производительности ваших приложений.

1.4 Проектирование БД

Учитывая особенности данной системы, рассмотренной в анализе предметной области, выделим таблицы и связи между ними, представленные на предварительной диаграмме (Рисунок 1.1).

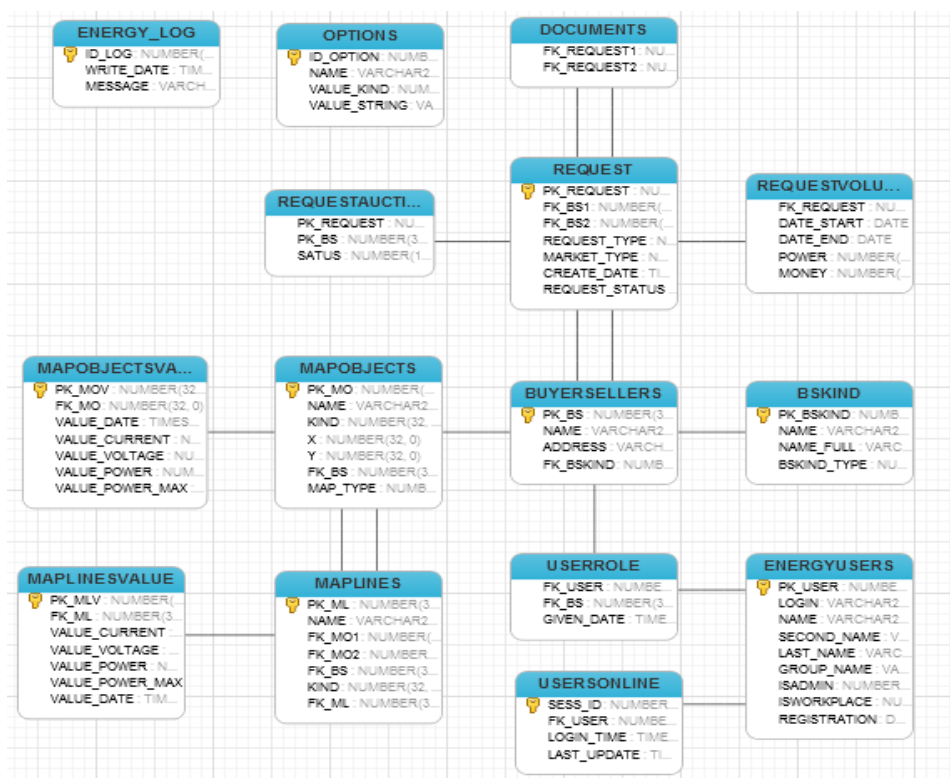


Рисунок 1.1 – Предварительная диаграмма

На основе этапа начальной разработки сформируем бизнес–правила. Затем определяем связи между сущностями, в основном опираясь на описание операций. Точнее связи сущности базируются на бизнес–правилах, построенных на основе подробного описания операций.

1.4.1 Разработка и построение подробной Er–диаграммы на основании бизнес правил.

Бизнес правило 1

Каждый пользователь может иметь разную роль в разный этап рынка. Роль за пользователем не прикрепляется навечно. На основе первого бизнес–правила мы получаем сегмент модели, представленный на рисунке 1.2.

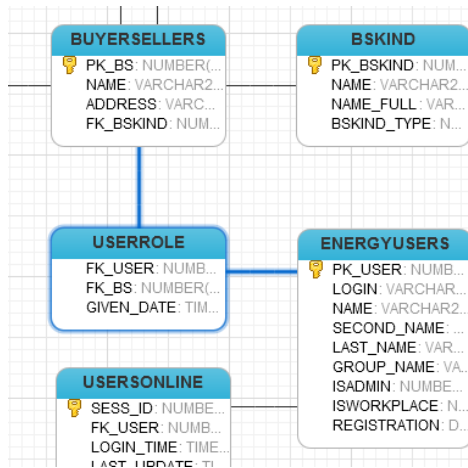


Рисунок 1.2 – Модель бизнес правила 1

Бизнес правило 2

У одного договора может быть несколько периодов поставки электроэнергии.

На основе второго бизнес–правила мы получаем сегмент модели, представленный на рисунке 1.3.

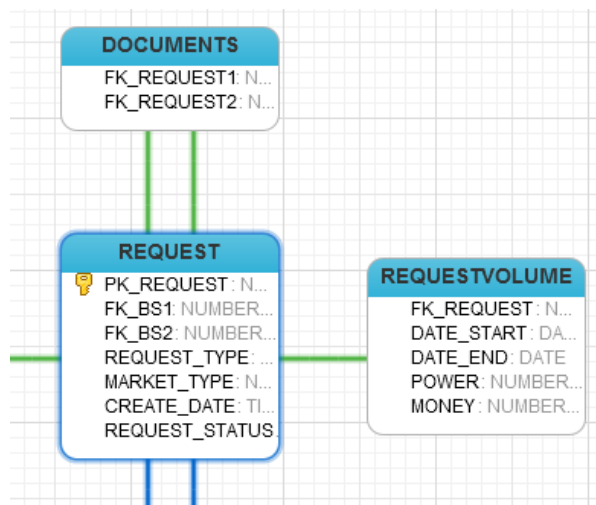


Рисунок 1.3 – Модель бизнес правила 2

Бизнес правило 3

У каждого предприятия есть тип, один тип может быть у нескольких предприятий.

На основе третьего бизнес–правила мы получаем сегмент VISIO–модели, представленный на рисунке 1.4.

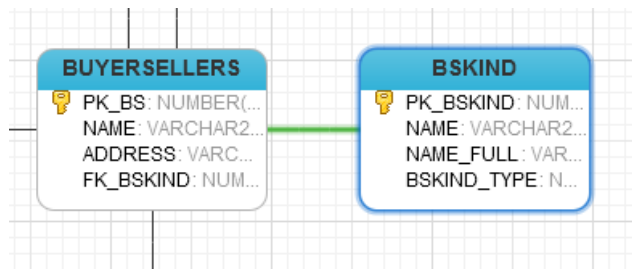


Рисунок 1.4 – Модель бизнес правила 3

Бизнес правило 4

У каждого объекта карты могут быть разные параметры в разное время.

На основе четвертого бизнес–правила мы получаем сегмент модели, представленный на рисунках 1.5, 1.6.

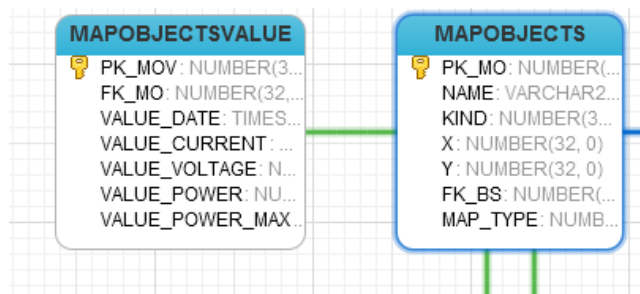


Рисунок 1.5 – Модель бизнес правила 4

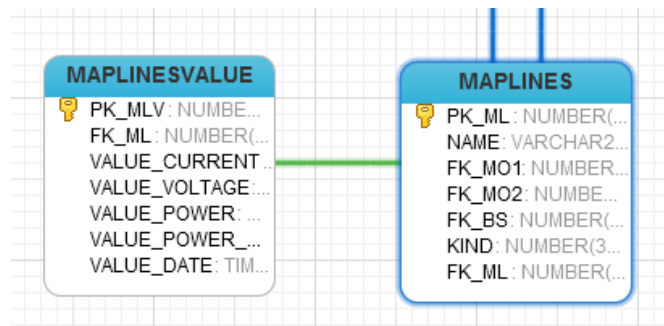


Рисунок 1.6 – Модель бизнес правила 4

Бизнес правило 5

В долгосрочном рынке на основе только 1 (о покупке или продаже) заявки может составляться договор.

На основе пятого бизнес–правила мы получаем сегмент модели, представленный на рисунке 1.7.

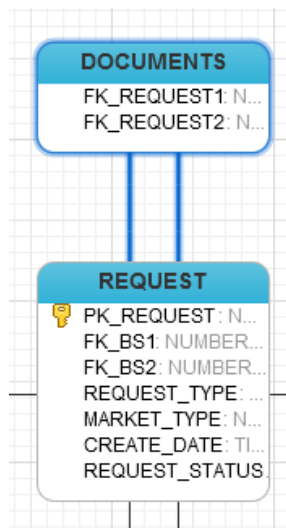


Рисунок 1.7 – Модель бизнес правила 5

Бизнес правило 6

Карта представляет из себя ориентированный граф, в сущности MapLines хранятся объект начала и объект конца.

На основе шестого бизнес–правила мы получаем сегмент модели, представленный на рисунке 1.8.

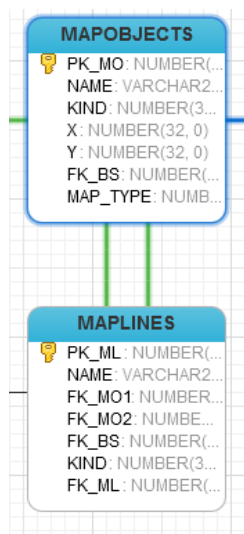
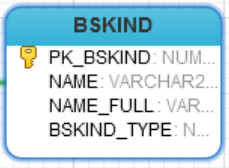

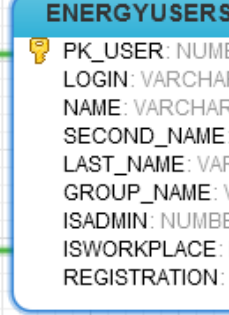
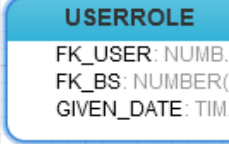
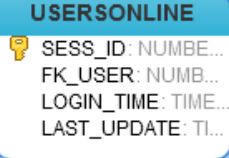
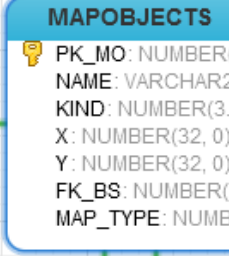
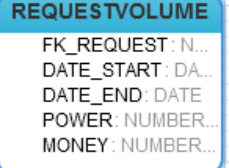


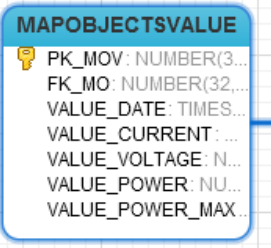
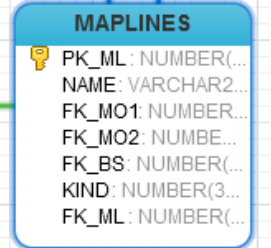
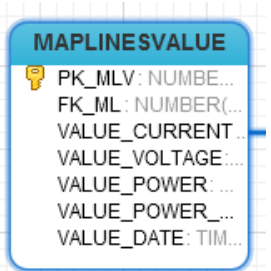

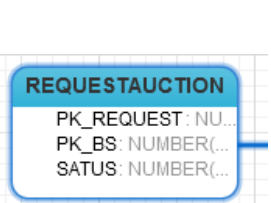
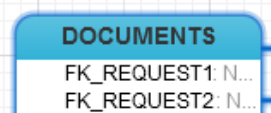
Рисунок 1.8 – Модель бизнес правила 6

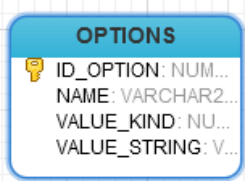
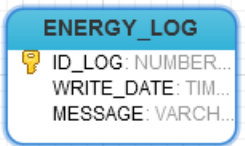
1.4.2 Определение атрибутов каждой сущности

В ниже приведенных таблице 1.1 описаны все атрибуты сущностей.

Т а б л и ц а 1.1 – Атрибуты каждой сущности

| Сегмент модели | Описание |
|--|--|
|  <p>BSKIND PK_BSKIND: NUM... NAME: VARCHAR2... NAME_FULL: VAR... BSKIND_TYPE: N...</p> | Таблица вид предприятия содержит следующие данные: pk_bskind – вид предприятия. name – название вида предприятия. name_full – полное название вида. bskind_type – тип предприятия. |
|  <p>BUYERSELLERS PK_BS: NUMBER(...) NAME: VARCHAR2... ADDRESS: VARC... FK_BSKIND: NUM...</p> | Таблица предприятия содержит следующие данные: pk_bs – типа предприятия. name – наименование предприятия. address – адрес предприятия. fk_bskind – тип предприятия. |
|  <p>ENERGYUSERS PK_USER: NUMB... LOGIN: VARCHAR... NAME: VARCHAR2... SECOND_NAME: ... LAST_NAME: VAR... GROUP_NAME: VA... ISADMIN: NUMBE... ISWORKPLACE: N... REGISTRATION: D...</p> | Таблица пользователи содержит следующие данные: pk_user – идентификатор пользователя login – имя пользователя, имя в БД name – имя second_name – отчество last_name – фамилия group_name – группа isadmin – статус isworkplace – является рабочим местом? registration – дата регистрации |
|  <p>USERROLE FK_USER: NUMB... FK_BS: NUMBER(...) GIVEN_DATE: TIM...</p> | Таблица роль пользователя содержит следующие данные: fk_user – идентификатор пользователя fk_bs – идентификатор предприятия given_date – дата получения роли |
|  <p>USERONLINE SESS_ID: NUMBE... FK_USER: NUMB... LOGIN_TIME: TIME... LAST_UPDATE: TI...</p> | Таблица пользователи в сети содержит следующие данные: sess_id – идентификатор сессии fk_user – идентификатор пользователя login_time – время входа last_update – последнее обновленик |
|  <p>MAPOBJECTS PK_MO: NUMBER(...) NAME: VARCHAR2... KIND: NUMBER(3...) X: NUMBER(32, 0) Y: NUMBER(32, 0) FK_BS: NUMBER(...) MAP_TYPE: NUMB...</p> | Таблица объекты карты содержит следующие данные: pk_mo – идентификатор объекта name – наименование объекта kind – вид объекта x – координата X y – координата Y fk_bs – идентификатор предприятия map_type – тип карты |
|  <p>REQUESTVOLUME FK_REQUEST: N... DATE_START: DA... DATE_END: DATE POWER: NUMBER... MONEY: NUMBER...</p> | Таблица объём заявки содержит следующие данные: fk_request – идентификатор заявки date_start – начало периода date_end – конец периода power – мощность money – стоимость |

| Сегмент модели | Описание |
|---|--|
|  <p>MAPOBJECTSVALUE</p> <ul style="list-style-type: none"> PK_MOV: NUMBER(3...) FK_MO: NUMBER(32...) VALUE_DATE: TIMES... VALUE_CURRENT: ... VALUE_VOLTAGE: N... VALUE_POWER: NU... VALUE_POWER_MAX... | <p>Таблица значения объекта карты содержит следующие данные:</p> <ul style="list-style-type: none"> pk_mov – идентификатор значения fk_mo – идентификатор объекта value_date – дата value_current – ток value_voltage – напряжение value_power – мощность value_power_max – предельная мощность |
|  <p>MAPLINES</p> <ul style="list-style-type: none"> PK_ML: NUMBER(...) NAME: VARCHAR2... FK_MO1: NUMBER... FK_MO2: NUMBE... FK_BS: NUMBER(...) KIND: NUMBER(3... FK_ML: NUMBER(...) | <p>Таблица линии карты содержит следующие данные:</p> <ul style="list-style-type: none"> pk_ml – идентификатор линии name – текст линии fk_mo1 – идентификатор объекта начала fk_mo2 – идентификатор объекта конца fk_bs – идентификатор предприятия kind – вид линии fk_ml – родительская линия |
|  <p>MAPLINESVALUE</p> <ul style="list-style-type: none"> PK_MLV: NUMBE... FK_ML: NUMBER(...) VALUE_CURRENT... VALUE_VOLTAGE... VALUE_POWER: ... VALUE_POWER_... VALUE_DATE: TIM... | <p>Таблица значения линии карты содержит следующие данные:</p> <ul style="list-style-type: none"> pk_mlv – идентификатор значения линии fk_ml – идентификатор линии value_current – ток value_voltage – напряжение value_power – мощность value_power_max – предельная мощность value_date – дата |
|  <p>REQUEST</p> <ul style="list-style-type: none"> PK_REQUEST: N... FK_BS1: NUMBER... FK_BS2: NUMBER... REQUEST_TYPE: ... MARKET_TYPE: N... CREATE_DATE: TI... REQUEST_STATUS | <p>Таблица заявки содержит следующие данные:</p> <ul style="list-style-type: none"> pk_request – идентификатор заявки продажи fk_bs1 – идентификатор предприятия подавшего заявку fk_bs2 – идентификатор предприятия купившего заявку request_type – тип заявки (покупка/продажа) market_type – тип рынка при котором была подана заявка create_date – дата подачи request_status – статус заявки |
|  <p>REQUESTAUCTION</p> <ul style="list-style-type: none"> PK_REQUEST: NU... PK_BS: NUMBER(...) SATUS: NUMBER(...) | <p>Таблица заявки аукциона содержит следующие данные:</p> <ul style="list-style-type: none"> pk_request – идентификатор заявки pk_bs – заявка продажи status – заявка покупки fk_seller – продавец fk_buyer – покупатель doc_date – дата doc_market – тип рынка |
|  <p>DOCUMENTS</p> <ul style="list-style-type: none"> FK_REQUEST1: N... FK_REQUEST2: N... | <p>Таблица контракты содержит следующие данные:</p> <ul style="list-style-type: none"> fk_request1 – идентификатор заявки 1 fk_request2 – идентификатор заявки 2 |

| Сегмент модели | Описание |
|---|--|
|  | Таблица настройки содержит следующие данные: id_option – идентификатор настройки name_option – наименование value_kind – тип value_string – строковое значение |
|  | Таблица лог содержит следующие данные: id_log – идентификатор сообщения write_date – дата записи message – сообщение |

1.4.3 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 есть).

Отношения приведены в таблицах 1.2–1.16. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Обязательное поле для краткости обозначено not null, необязательное – null. Типы данных обозначаются так: number – числовой, varchar2 – символьный, timestamp – дата [8].

Т а б л и ц а 1.2 – Схема отношения пользователи (EnergyUsers)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|----------------------------|--------------|------------|-------------------------|
| идентификатор пользователя | pk_user | number | Первичный ключ not null |
| имя пользователя | login | varchar2 | not null |
| Фамилия | name | varchar2 | null |
| имя | second_name | varchar2 | null |
| отчество | last_name | varchar2 | null |
| группа | Group | varchar2 | null |
| статус | Isadmin | number | default 0 |
| дата регистрации | Registration | date | default SYSDATE() |
| Признак рабочего места | isworkplace | number | default 0 |

Т а б л и ц а 1.3 – Схема отношения тип предприятия (BSKind)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|--------------------------------|-----------|------------|-------------------------|
| идентификатор типа предприятия | pk_bskind | number | Первичный ключ not null |
| название типа предприятия | Name | varchar2 | not null |
| полное название типа | Type | varchar2 | not null |
| Тип загрязнителя | name_full | number | default 0 |

Т а б л и ц а 1.4 – Схема отношения предприятия (BuyerSellers)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|----------------------------|-----------|------------|--|
| идентификатор предприятия. | pk_bs | number | Первичный ключ not null |
| наименование предприятия | name | varchar2 | not null |
| адрес предприятия | address | varchar2 | null |
| тип предприятия | fk_bskind | number | Внешний ключ (к типу предприятия) not null |

Т а б л и ц а 1.5 – Схема лог (EnergyLog)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|-------------------------|------------|------------|-------------------------|
| идентификатор настройки | id_log | number | Первичный ключ not null |
| наименование | write_date | varchar2 | not null |
| тип | Message | number | not null |

Т а б л и ц а 1.6 – Схема отношения линии карты (MapLines)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|------------------------------|----------|------------|--|
| идентификатор линии | pk_ml | number | Первичный ключ not null |
| текст линии | ml_name | varchar2 | null |
| идентификатор объекта начала | fk_mo1 | number | Внешний ключ (к объектам) not null |
| идентификатор объекта конца | fk_mo2 | number | Внешний ключ (к объектам) not null |
| идентификатор предприятия | fk_bs | number | Внешний ключ (к предприятиям) not null |
| тип линии | Kind | number | not null |
| родительская линия | fk_ml | number | not null |

Т а б л и ц а 1.7 – Схема отношения значения объекта карты (MapObjectsValue)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|------------------------|-----------------|------------|------------------------------------|
| идентификатор значения | pk_mov | number | Первичный ключ not null |
| идентификатор объекта | fk_mo | number | Внешний ключ (к объектам) not null |
| дата | value_date | timestamp | not null |
| ток | value_current | number | default 0 |
| напряжение | value_voltage | number | default 0 |
| мощность | value_power | number | default 0 |
| предельная мощность | value_power_max | number | default 0 |

Т а б л и ц а 1.8 – Схема отношения контракты (Documents)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|------------------------|-------------|------------|--|
| идентификатор заявки 1 | pk_request1 | number | Внешний ключ (к заявке продажи) not null |
| идентификатор заявки 2 | pk_request2 | number | Внешний ключ (к заявке продажи) not null |

Т а б л и ц а 1.9 – Схема отношения значения линий (MapLinesValue)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|------------------------------|-----------------|------------|----------------------------------|
| идентификатор значения линии | pk_mlv | number | Первичный ключ not null |
| идентификатор линии | fk_ml | number | Внешний ключ (к линиям) not null |
| ток | value_current | number | default 0 |
| напряжение | value_voltage | number | default 0 |
| мощность | value_power | number | default 0 |
| предельная мощность | value_power_max | number | default 0 |
| дата | value_date | timestamp | not null |

Т а б л и ц а 1.10 – Схема отношения объекты карты (MapObjects)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|---------------------------|----------|------------|--|
| идентификатор объекта | pk_mo | number | Первичный ключ not null |
| наименование объекта | Name | varchar2 | not null |
| тип объекта | Type | number | default 0 |
| координата X | X | number | default 0 |
| координата Y | Y | number | default 0 |
| идентификатор предприятия | fk_bs | number | Внешний ключ (к предприятиям) not null |
| тип карты | map_type | number | not null |

Т а б л и ц а 1.11 – Схема отношения заявки (Request)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|--|----------------|------------|--|
| идентификатор заявки | pk_request | number | Первичный ключ not null |
| идентификатор предприятия подавшего заявку | fk_bs1 | number | Внешний ключ (к предприятиям) not null |
| идентификатор предприятия взявшего заявку | fk_bs2 | number | Внешний ключ (к предприятиям) not null |
| Тип заявки | request_type | number | not null |
| Тип маркета | market_type | number | not null |
| дата | create_date | timestamp | not null |
| Статус заявки | Request_status | number | not null |

Т а б л и ц а 1.12 – Схема отношения настройки (Options)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|-------------------------|--------------|------------|-------------------------|
| идентификатор настройки | id_option | number | Первичный ключ not null |
| наименование | name_option | varchar2 | not null |
| тип | value_type | number | not null |
| числовое значение | value_number | number | null |
| строковое значение | value_string | varchar2 | null |

Т а б л и ц а 1.13 – Схема отношения роли пользователей (UserRole)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|----------------------------|----------|------------|---|
| идентификатор пользователя | fk_user | number | Внешний ключ (к пользователям) not null |
| идентификатор предприятия | fk_bs | number | not null |
| дата | ur_date | timestamp | default GETDATE() |

Т а б л и ц а 1.14 – Схема отношения пользователи в сети (UsersOnline)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|----------------------------|-------------|------------|---|
| идентификатор сессии | sess_id | number | Первичный ключ not null |
| идентификатор пользователя | fk_user | number | Внешний ключ (к пользователям) not null |
| дата входа | login_time | timestamp | null |
| последнее обновленик | last_update | timestamp | null |

Т а б л и ц а 1.15 – Схема отношения периоды заявки продажи (RequestVolume)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|------------------------|------------|------------|--|
| идентификатор заявки | fk_request | number | Внешний ключ (к заявке продажи) not null |
| Дата начала периода | date_start | timestamp | not null |
| Дата окончания периода | date_end | timestamp | not null |
| мощность | Power | number | not null |
| стоимость | Money | number | not null |

Т а б л и ц а 1.16 – Схема отношения заявки на аукционе (RequesAuction)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|----------------------|------------|------------|---------------------------------------|
| идентификатор заявки | pk_request | number | Внешний ключ (к заявке) not null |
| Предприятие | pk_bs | number | Внешний ключ (к предприятию) not null |
| статус | Status | number | not null |

1.4.4 Расчет места для хранения БД

На этом этапе, необходимо знать какой объем памяти будет занимать создаваемая база данных. Объем 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 . Объем памяти, занимаемый программными модулями пользователя, обычно невелик по сравнению с объемом самих данных, поэтому может не учитываться. В проекте рассчитывается предполагаемый максимальный объем памяти занимаемой БД. Размер данных каждой строки таблицы представлено в таблицах 1.17–1.31.

Т а б л и ц а 1.17 – Расчет физической памяти для таблицы пользователи (Users)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|----------------------------|--------------|------------|------------|
| идентификатор пользователя | pk_user | number | 8 |
| имя пользователя | login | varchar2 | 20 |
| фамилия | name | varchar2 | 20 |
| имя | second_name | varchar2 | 20 |
| отчество | last_name | varchar2 | 20 |
| группа | _group | varchar2 | 20 |
| статус | isadmin | number | 1 |
| дата регистрации | registration | date | 8 |
| Является ли рабочим местом | isworkplace | number | 1 |

Общая длина строки: 118 байт.

Число строк: ~ 75.

Общий объем требуемой памяти: ~ 8850 байт.

Т а б л и ц а 1.18 – Расчет физической памяти для таблицы заявки (Request)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|------------------------------|----------------|------------|------------|
| идентификатор заявки продажи | pk_request | number | 8 |
| идентификатор предприятия | fk_bs1 | number | 8 |
| идентификатор покупателя | fk_bs2 | number | 8 |
| дата | Request_type | date | 8 |
| тип рынка | Market_type | number | 1 |
| Дата подачи заявки | Create_date | Number | 1 |
| Статус заявки | Request_Status | Number | 1 |

Общая длина строки: 35 байт.

Число строк: ~ 3200.

Общий объем требуемой памяти: ~ 112000 байт.

Т а б л и ц а 1.19 – Расчет физической памяти для таблицы объём заявки (RequestVolume)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|----------------------|------------|------------|------------|
| идентификатор заявки | fk_request | number | 8 |
| Дата начала | Date_start | timestamp | 8 |
| Дата окончания | Date_end | timestamp | 8 |
| мощность | Power | number | 8 |
| стоимость | Money | number | 8 |

Общая длина строки: 40 байт.

Число строк: ~ 15600.

Общий объем требуемой памяти: ~ 624000 байт.

Т а б л и ц а 1.20 – Расчет физической памяти для таблицы объекты карты (MapObjects)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|---------------------------|-------------|------------|------------|
| идентификатор объекта | pk_mo | number | 8 |
| наименование объекта | mo_name | varchar2 | 20 |
| тип объекта | mo_type | number | 8 |
| координата X | mo_x | number | 8 |
| координата Y | mo_y | number | 8 |
| идентификатор предприятия | fk_bs | number | 8 |
| тип карты | mo_map_type | number | 1 |

Общая длина строки: 61 байт.

Число строк: ~ 500.

Общий объем требуемой памяти: ~ 30500 байт.

Т а б л и ц а 1.21 – Расчет физической памяти для таблицы линии карты (MapLines)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|------------------------------|----------|------------|------------|
| идентификатор линии | pk_ml | number | 8 |
| текст линии | ml_name | varchar2 | 20 |
| идентификатор объекта начала | fk_mo1 | number | 8 |
| идентификатор объекта конца | fk_mo2 | number | 8 |
| идентификатор предприятия | fk_bs | number | 8 |
| тип линии | ml_type | number | 8 |
| родительская линия | fk_ml | number | 8 |

Общая длина строки: 68 байт.

Число строк: ~ 500.

Общий объем требуемой памяти: ~ 34000 байт.

Т а б л и ц а 1.22 – Расчет физической памяти для таблицы значения объекта карты (MapObjectsValue)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|------------------------|--------------|------------|------------|
| идентификатор значения | pk_mov | number | 8 |
| идентификатор объекта | fk_mo | number | 8 |
| дата | mo_date | timestamp | 8 |
| ток | mo_current | number | 8 |
| напряжение | mo_voltage | number | 8 |
| мощность | mo_power | number | 8 |
| предельная мощность | mo_power_max | number | 8 |

Общая длина строки: 56 байт.

Число строк: ~ 8000.

Общий объем требуемой памяти: ~ 448000 байт.

Т а б л и ц а 1.23 – Расчет физической памяти для таблицы значения линий (MapLinesValue)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|------------------------------|--------------|------------|------------|
| идентификатор значения линии | pk_mlv | number | 8 |
| идентификатор линии | fk_ml | number | 8 |
| ток | ml_current | number | 8 |
| напряжение | ml_voltage | number | 8 |
| мощность | ml_power | number | 8 |
| предельная мощность | ml_power_max | number | 8 |
| дата | ml_date | timestamp | 8 |

Общая длина строки: 56 байт.

Число строк: ~ 8000.

Общий объем требуемой памяти: ~ 448000 байт.

Т а б л и ц а 1.24 – Расчет физической памяти для таблицы Типы предприятий (BuyerSellersType)

| Содержание поля | Имя поля | Тип, длина | Длина (байт) |
|--------------------------------|-----------|------------|--------------|
| идентификатор типа предприятия | pk_bskind | number | 8 |
| название вида предприятия | Name | varchar2 | 20 |
| полное название вида | name_full | varchar2 | 100 |
| Тип предприятия | type | number | 8 |

Общая длина строки: 136 байт.

Число строк: ~ 15.

Общий объем требуемой памяти ~ 2040 байт.

Т а б л и ц а 1.25 – Расчет физической памяти для таблицы Предприятия (BuyerSellers)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|----------------------------|-----------|------------|------------|
| идентификатор предприятия. | pk_bt | number | 8 |
| наименование предприятия | name | Varchar2 | 100 |
| адрес предприятия | address | Varchar2 | 500 |
| тип предприятия | fk_bskind | Number | 8 |

Общая длина строки: 616 байт.

Число строк: ~ 50.

Общий объем требуемой памяти: ~ 30800 байт.

Т а б л и ц а 1.26 – Расчет физической памяти для таблицы роли пользователей (UserRole)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|----------------------------|----------|------------|------------|
| идентификатор пользователя | fk_user | number | 8 |
| идентификатор предприятия | fk_bs | number | 8 |
| дата | ur_date | timestamp | 8 |

Общая длина строки: 24 байт.
 Число строк: ~1200.
 Общий объем требуемой памяти: ~ 28800 байт.

Т а б л и ц а 1.27 – Расчет физической памяти для таблицы пользователи в сети (UsersOnline)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|----------------------------|----------------|------------|------------|
| идентификатор сессии | sess_id | number | 8 |
| идентификатор пользователя | fk_user | number | 8 |
| дата входа | usr_login_time | timestamp | 8 |
| последнее обновленик | last_update | timestamp | 8 |

Общая длина строки: 32 байт.
 Число строк: ~ 20.
 Общий объем требуемой памяти: ~ 640 байт.

Т а б л и ц а 1.28 – Расчет физической памяти для таблицы заявки аукциона (Request Auction)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|----------------------|------------|------------|------------|
| идентификатор заявки | pk_request | number | 8 |
| Участник рынка | fk_bs | number | 8 |
| Статус лота | Status | number | 1 |

Общая длина строки: 17 байт.
 Число строк: ~ 1200.
 Общий объем требуемой памяти: ~ 20400 байт.

Т а б л и ц а 1.29 – Расчет физической памяти для таблицы контракты (Documents)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|------------------------|-------------|------------|------------|
| идентификатор заявки 1 | fk_request1 | number | 8 |
| идентификатор заявки 2 | fk_request2 | number | 8 |

Общая длина строки: 16 байт.
 Число строк: ~ 3200.
 Общий объем требуемой памяти: ~ 51200 байт.

Т а б л и ц а 1.30 – Расчет физической памяти для таблицы настройки (Options)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|-------------------------|--------------|------------|------------|
| идентификатор настройки | id_option | number | 8 |
| наименование | name_option | varchar2 | 20 |
| тип | value_type | number | 32 |
| строковое значение | value_string | varchar2 | 1000 |

Общая длина строки: 1060 байт.
 Число строк: ~ 5.
 Общий объем требуемой памяти: ~ 5300 байт.

Т а б л и ц а 1.31 – Расчет физической памяти для таблицы лог (EnergyLog)

| Содержание поля | Имя поля | Тип, длина | Примечание |
|-------------------------|------------|------------|------------|
| идентификатор настройки | id_log | number | 8 |
| наименование | Write_date | varchar2 | 8 |
| тип | message | number | 4000 |

Общая длина строки: 4016 байт.
 Число строк: ~ 100000.
 Общий объем требуемой памяти: ~ 401600000 байт.

Таким образом, из полученных данным можно рассчитать приблизительный максимальный объем базы данных. Объем будет равен сумме объемов всех таблиц:

База данных = 403 444 530 (*байт*).

Представленные выше расчеты сделаны только для таблиц, т.к. данные в таблицах будут накапливаться, все процедуры и триггеры будут обрабатывать больше информации, то минимальный объем памяти будет составлять $403\ 444\ 530 * 2 = 806\ 889\ 060$ (*байт*). Для более эффективной работы БД необходимо будет выделить **5 ГБ (4 034 445 300 байт)**, данный размер памяти рассчитан на 10 лет использования БД.

Объём памяти, занимаемый программными модулями на стороне сервера занимает 3,5 ГБ. Объём памяти занимаемый программными модулями на стороне клиентов занимает 50 МБ. Один клиент это один компьютер в компьютерной аудитории, 9–10 мест. Требуемый объём оперативной памяти определяется на основании анализа интенсивности запросов и объёма результирующих данных. Oracle рекомендует выделить либо 40% от общего объёма оперативной памяти, либо 2 ГБ под базу данных.

2 Реализация приложения

2.1 Физическое проектирование БД

2.1.1 Создание табличного пространства

Для удобства все данные БД будут находиться в отдельном табличном пространстве.

```
create tablespace energy datafile
'd:\oracle\app\oracle\oradata\energy_01.dbf' size 100M autoextend
on;
```

2.1.2 Создание таблиц

```
-----
-- Структура таблицы "ENERGY"."BSKIND"
-----
CREATE TABLE "ENERGY"."BSKIND" (
  "PK_BSKIND" NUMBER(32) NOT NULL ,
  "NAME" VARCHAR2(20 BYTE) NULL ,
  "NAME_FULL" VARCHAR2(100 BYTE) NULL ,
  "BSKIND_TYPE" NUMBER(2) NULL
);

-----
-- Структура таблицы "ENERGY"."BUYERSELLERS"
-----
CREATE TABLE "ENERGY"."BUYERSELLERS" (
  "PK_BS" NUMBER(32) NOT NULL ,
  "NAME" VARCHAR2(100 BYTE) NULL ,
  "ADDRESS" VARCHAR2(500 BYTE) NULL ,
  "FK_BSKIND" NUMBER(32) NULL
);

-----
-- Структура таблицы "ENERGY"."DOCUMENTS"
-----
CREATE TABLE "ENERGY"."DOCUMENTS" (
  "FK_REQUEST1" NUMBER(32) NULL ,
  "FK_REQUEST2" NUMBER(32) NULL
);

-----
-- Структура таблицы "ENERGY"."ENERGY_LOG"
-----
CREATE TABLE "ENERGY"."ENERGY_LOG" (
  "ID_LOG" NUMBER(32) NOT NULL ,
  "WRITE_DATE" TIMESTAMP(6) NULL ,
```

```

"MESSAGE" VARCHAR2(4000 BYTE) NULL
);

-----
-- Структура таблицы "ENERGY"."ENERGYUSERS"
-----
CREATE TABLE "ENERGY"."ENERGYUSERS" (
"PK_USER" NUMBER(32) NOT NULL ,
"LOGIN" VARCHAR2(20 BYTE) NULL ,
"NAME" VARCHAR2(20 BYTE) NULL ,
"SECOND_NAME" VARCHAR2(20 BYTE) NULL ,
"LAST_NAME" VARCHAR2(20 BYTE) NULL ,
"GROUP_NAME" VARCHAR2(10 BYTE) NULL ,
"ISADMIN" NUMBER(1) DEFAULT 0 NULL ,
"ISWORKPLACE" NUMBER(1) DEFAULT 0 NULL ,
"REGISTRATION" DATE DEFAULT sysdate NULL
);

-----
-- Структура таблицы "ENERGY"."MAPLINES"
-----
CREATE TABLE "ENERGY"."MAPLINES" (
"PK_ML" NUMBER(32) NOT NULL ,
"NAME" VARCHAR2(50 BYTE) NULL ,
"FK_MO1" NUMBER(32) NULL ,
"FK_MO2" NUMBER(32) NULL ,
"FK_BS" NUMBER(32) NULL ,
"KIND" NUMBER(32) NULL ,
"FK_ML" NUMBER(32) NULL
);

-----
-- Структура таблицы "ENERGY"."MAPLINESVALUE"
-----
CREATE TABLE "ENERGY"."MAPLINESVALUE" (
"PK_MLV" NUMBER(32) NOT NULL ,
"FK_ML" NUMBER(32) NULL ,
"VALUE_CURRENT" NUMBER(32) DEFAULT 0 NULL ,
"VALUE_VOLTAGE" NUMBER(32) DEFAULT 0 NULL ,
"VALUE_POWER" NUMBER(32) DEFAULT 0 NULL ,
"VALUE_POWER_MAX" NUMBER(32) DEFAULT 0 NULL ,
"VALUE_DATE" TIMESTAMP(6) NULL
);

-----
-- Структура таблицы "ENERGY"."MAPOBJECTS"
-----
CREATE TABLE "ENERGY"."MAPOBJECTS" (
"PK_MO" NUMBER(32) NOT NULL ,
"NAME" VARCHAR2(50 BYTE) NULL ,
"KIND" NUMBER(32) DEFAULT 0 NULL ,

```

```

"X" NUMBER(32) DEFAULT 0 NULL ,
"Y" NUMBER(32) DEFAULT 0 NULL ,
"FK_BS" NUMBER(32) NULL ,
"MAP_TYPE" NUMBER(1) NULL
);

-----
-- Структура таблицы "ENERGY"."МАРОBJECTSVALUE"
-----
CREATE TABLE "ENERGY"."МАРОBJECTSVALUE" (
"PK_MOV" NUMBER(32) NOT NULL ,
"FK_MO" NUMBER(32) NULL ,
"VALUE_DATE" TIMESTAMP(6) NULL ,
"VALUE_CURRENT" NUMBER(32) DEFAULT 0 NULL ,
"VALUE_VOLTAGE" NUMBER(32) DEFAULT 0 NULL ,
"VALUE_POWER" NUMBER(32) DEFAULT 0 NULL ,
"VALUE_POWER_MAX" NUMBER(32) DEFAULT 0 NULL
);

-----
-- Структура таблицы "ENERGY"."OPTIONS"
-----
CREATE TABLE "ENERGY"."OPTIONS" (
"ID_OPTION" NUMBER(32) NOT NULL ,
"NAME" VARCHAR2(20 BYTE) NULL ,
"VALUE_KIND" NUMBER(32) NULL ,
"VALUE_STRING" VARCHAR2(1000 BYTE) NULL
);

-----
-- Структура таблицы "ENERGY"."REQUEST"
-----
CREATE TABLE "ENERGY"."REQUEST" (
"PK_REQUEST" NUMBER(32) NOT NULL ,
"FK_BS1" NUMBER(32) NULL ,
"FK_BS2" NUMBER(32) NULL ,
"REQUEST_TYPE" NUMBER(1) NULL ,
"MARKET_TYPE" NUMBER(1) NULL ,
"CREATE_DATE" TIMESTAMP(6) NULL ,
"REQUEST_STATUS" NUMBER(32) NULL
);

-----
-- Структура таблицы "ENERGY"."REQUESTAUCTION"
-----
CREATE TABLE "ENERGY"."REQUESTAUCTION" (
"PK_REQUEST" NUMBER(32) NULL ,
"PK_BS" NUMBER(32) NULL ,
"STATUS" NUMBER(1) NULL
);

```

```

-----
-- Структура таблицы "ENERGY"."REQUESTVOLUME"
-----
CREATE TABLE "ENERGY"."REQUESTVOLUME" (
"FK_REQUEST" NUMBER(32) NOT NULL ,
"DATE_START" DATE NULL ,
"DATE_END" DATE NULL ,
"POWER" NUMBER(32) NULL ,
"MONEY" NUMBER(32) NULL
);

```

```

-----
-- Структура таблицы "ENERGY"."USERROLE"
-----
CREATE TABLE "ENERGY"."USERROLE" (
"FK_USER" NUMBER(32) NULL ,
"FK_BS" NUMBER(32) NULL ,
"GIVEN_DATE" TIMESTAMP(6) DEFAULT SYSDATE NULL
);

```

```

-----
-- Структура таблицы "ENERGY"."USERSONLINE"
-----
CREATE TABLE "ENERGY"."USERSONLINE" (
"SESS_ID" NUMBER(32) NOT NULL ,
"FK_USER" NUMBER(32) NULL ,
"LOGIN_TIME" TIMESTAMP(6) NULL ,
"LAST_UPDATE" TIMESTAMP(6) NULL
);

```

2.1.3 Создание реляционных отношений

```

-----
-- Первичный ключ для таблицы "ENERGY"."BSKIND"
-----
ALTER TABLE "ENERGY"."BSKIND" ADD PRIMARY KEY ("PK_BSKIND");

```

```

-----
-- Первичный ключ для таблицы "ENERGY"."BUYERSELLERS"
-----
ALTER TABLE "ENERGY"."BUYERSELLERS" ADD PRIMARY KEY
("PK_BS");

```

```

-----
-- Первичный ключ для таблицы "ENERGY"."ENERGY_LOG"
-----
ALTER TABLE "ENERGY"."ENERGY_LOG" ADD PRIMARY KEY ("ID_LOG");

```

```

-----
-- Первичный ключ для таблицы "ENERGY"."ENERGYUSERS"
-----

```



```

ALTER TABLE "ENERGY"."ENERGYUSERS" ADD PRIMARY KEY
("PK_USER");

-----
-- Первичный ключ для таблицы "ENERGY"."MAPLINES"
-----
ALTER TABLE "ENERGY"."MAPLINES" ADD PRIMARY KEY ("PK_ML");

-----
-- Первичный ключ для таблицы "ENERGY"."MAPLINESVALUE"
-----
ALTER TABLE "ENERGY"."MAPLINESVALUE" ADD PRIMARY KEY
("PK_MLV");

-----
-- Первичный ключ для таблицы "ENERGY"."МАРОBJECTS"
-----
ALTER TABLE "ENERGY"."МАРОBJECTS" ADD PRIMARY KEY ("PK_MO");

-----
-- Первичный ключ для таблицы "ENERGY"."МАРОBJECTSVALUE"
-----
ALTER TABLE "ENERGY"."МАРОBJECTSVALUE" ADD PRIMARY KEY
("PK_MOV");

-----
-- Первичный ключ для таблицы "ENERGY"."OPTIONS"
-----
ALTER TABLE "ENERGY"."OPTIONS" ADD PRIMARY KEY ("ID_OPTION");

-----
-- Первичный ключ для таблицы "ENERGY"."REQUEST"
-----
ALTER TABLE "ENERGY"."REQUEST" ADD PRIMARY KEY
("PK_REQUEST");

-----
-- Checks structure for table "ENERGY"."REQUESTVOLUME"
-----
ALTER TABLE "ENERGY"."REQUESTVOLUME" ADD CHECK ("FK_REQUEST"
IS NOT NULL);

-----
-- Первичный ключ для таблицы "ENERGY"."USERSONLINE"
-----
ALTER TABLE "ENERGY"."USERSONLINE" ADD PRIMARY KEY
("SESS_ID");

-----
-- Вторичный ключ для таблицы "ENERGY"."BUYERSELLERS"
-----

```

```

ALTER TABLE "ENERGY"."BUYERSELLERS" ADD FOREIGN KEY
("FK_BSKIND") REFERENCES "ENERGY"."BSKIND" ("PK_BSKIND");

-- -----
-- Вторичный ключ для таблицы "ENERGY"."DOCUMENTS"
-- -----
ALTER TABLE "ENERGY"."DOCUMENTS" ADD FOREIGN KEY
("FK_REQUEST1") REFERENCES "ENERGY"."REQUEST" ("PK_REQUEST");
ALTER TABLE "ENERGY"."DOCUMENTS" ADD FOREIGN KEY
("FK_REQUEST2") REFERENCES "ENERGY"."REQUEST" ("PK_REQUEST");

-- -----
-- Вторичный ключ для таблицы "ENERGY"."MAPLINES"
-- -----
ALTER TABLE "ENERGY"."MAPLINES" ADD FOREIGN KEY ("FK_MO1")
REFERENCES "ENERGY"."MAPOBJECTS" ("PK_MO") ON DELETE CASCADE;
ALTER TABLE "ENERGY"."MAPLINES" ADD FOREIGN KEY ("FK_MO2")
REFERENCES "ENERGY"."MAPOBJECTS" ("PK_MO") ON DELETE CASCADE;

-- -----
-- Вторичный ключ для таблицы "ENERGY"."MAPLINESVALUE"
-- -----
ALTER TABLE "ENERGY"."MAPLINESVALUE" ADD FOREIGN KEY
("FK_ML") REFERENCES "ENERGY"."MAPLINES" ("PK_ML") ON DELETE
CASCADE;

-- -----
-- Вторичный ключ для таблицы "ENERGY"."MAPOBJECTS"
-- -----
ALTER TABLE "ENERGY"."MAPOBJECTS" ADD FOREIGN KEY ("FK_BS")
REFERENCES "ENERGY"."BUYERSELLERS" ("PK_BS") ON DELETE SET NULL;

-- -----
-- Вторичный ключ для таблицы "ENERGY"."MAPOBJECTSVALUE"
-- -----
ALTER TABLE "ENERGY"."MAPOBJECTSVALUE" ADD FOREIGN KEY
("FK_MO") REFERENCES "ENERGY"."MAPOBJECTS" ("PK_MO") ON DELETE
CASCADE;

-- -----
-- Вторичный ключ для таблицы "ENERGY"."REQUEST"
-- -----
ALTER TABLE "ENERGY"."REQUEST" ADD FOREIGN KEY ("FK_BS1")
REFERENCES "ENERGY"."BUYERSELLERS" ("PK_BS") ON DELETE SET NULL;
ALTER TABLE "ENERGY"."REQUEST" ADD FOREIGN KEY ("FK_BS2")
REFERENCES "ENERGY"."BUYERSELLERS" ("PK_BS") ON DELETE SET NULL;

-- -----
-- Вторичный ключ для таблицы "ENERGY"."REQUESTAUCTION"
-- -----
ALTER TABLE "ENERGY"."REQUESTAUCTION" ADD FOREIGN KEY
("PK_BS") REFERENCES "ENERGY"."BUYERSELLERS" ("PK_BS");

```

```

ALTER TABLE "ENERGY"."REQUESTAUCTION" ADD FOREIGN KEY
("PK_REQUEST") REFERENCES "ENERGY"."REQUEST" ("PK_REQUEST");

-----
-- Вторичный ключ для таблицы "ENERGY"."REQUESTVOLUME"
-----
ALTER TABLE "ENERGY"."REQUESTVOLUME" ADD FOREIGN KEY
("FK_REQUEST") REFERENCES "ENERGY"."REQUEST" ("PK_REQUEST");

-----
-- Вторичный ключ для таблицы "ENERGY"."USERROLE"
-----
ALTER TABLE "ENERGY"."USERROLE" ADD FOREIGN KEY ("FK_BS")
REFERENCES "ENERGY"."BUYERSELLERS" ("PK_BS") ON DELETE CASCADE;
ALTER TABLE "ENERGY"."USERROLE" ADD FOREIGN KEY ("FK_USER")
REFERENCES "ENERGY"."ENERGYUSERS" ("PK_USER") ON DELETE CASCADE;

-----
-- Вторичный ключ для таблицы "ENERGY"."USERONLINE"
-----
ALTER TABLE "ENERGY"."USERONLINE" ADD FOREIGN KEY
("FK_USER") REFERENCES "ENERGY"."ENERGYUSERS" ("PK_USER");

```

2.1.4 Создание последовательностей

Для создания уникальных первичных ключей были использованы последовательности.

```

create sequence s_BSKind minvalue 1 maxvalue 100 start with 1
increment by 1 cache 20;
create sequence s_BuyerSellers minvalue 1 maxvalue 999999
start with 1 increment by 1 cache 20;
create sequence s_EnergyUsers minvalue 1 maxvalue 999999
start with 1 increment by 1 cache 20;
create sequence s_UsersOnline minvalue 1 maxvalue
999999999999 start with 1 increment by 1 cache 20;
create sequence s_MapObjects minvalue 1 maxvalue 999999999
start with 1 increment by 1 cache 20;
create sequence s_MapLines minvalue 1 maxvalue 999999999
start with 1 increment by 1 cache 20;
create sequence s_MapObjectsValue minvalue 1 maxvalue
9999999999999999 start with 1 increment by 1 cache 20;
create sequence s_MapLinesValue minvalue 1 maxvalue
9999999999999999 start with 1 increment by 1 cache 20;
create sequence s_Request minvalue 1 maxvalue 999999999 start
with 1 increment by 1 cache 20;

```

2.1.5 Заполнение созданной БД

Созданная БД содержит все еще пустые табличные структуры. Для заполнения базы данных был использован SQL скрипт написанный вручную.

Заполнение таблицы Настройки

```
INSERT INTO
Options(id_option,name,value_string)VALUES(0,'market',0);
INSERT INTO
Options(id_option,name,value_string)VALUES(1,'status',0);
INSERT INTO
Options(id_option,name,value_string)VALUES(2,'torgend','');
INSERT INTO
Options(id_option,name,value_string)VALUES(3,'date_end','');
commit;
```

Заполнение таблицы Пользователи

```
INSERT INTO EnergyUsers(login, name, isadmin, isworkplace)
VALUES('energy', 'admin', 1, 0); -- админ
Commit;
```

Заполнение таблицы Типы предприятий

```
INSERT INTO BSKind(pk_bskind, name, name_full, bskind_kind)
VALUES(1,'РЭК','Региональная электросетевая компания',0);
INSERT INTO BSKind(pk_bskind,name, name_full, bskind_kind)
VALUES(2,'НЭС','',0);
INSERT INTO BSKind(pk_bskind,name, name_full, bskind_kind)
VALUES(3,'ГЭС','Гидроэлектростанция',1);
INSERT INTO BSKind(pk_bskind,ame, name_full, bskind_kind)
VALUES(4,'ТЭЦ','Теплоэлектростанция',1);
INSERT INTO BSKind(pk_bskind,name, name_full, bskind_kind)
VALUES(5,'ОП','Оптовый покупатель',2);
INSERT INTO BSKind(pk_bskind,name, name_full, bskind_kind)
VALUES(6,'РП','Рыночный покупатель',2);
INSERT INTO BSKind(pk_bskind,name, name_full, bskind_kind)
VALUES(7,'КЭС','Теплоэлектростанция',1);
commit;
```

Заполнение таблицы Предприятия

```
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES(1,'АО «Астана-РЭК»','',1);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES(2,'АО «Западно-Казахстанская РЭК»','',1);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES(3,'«Алма»','',2);
```

```

INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (4, 'ТОО «Караганды Жарык»', '', 2);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (5, 'Шульбинская ГЭС', 'река Иртыш, близ пос.
Шульбинск', 3);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (6, 'Бухтарминская ГЭС', 'на реке Иртыш, ниже устья
реки Бухтарма вблизи города Серебрянска', 3);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (7, 'Аркалык. ТЭЦ', 'г. Аркалык', 4);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (8, 'Астана-Энергия ТЭЦ1', 'г. Астана', 4);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (9, 'Петропавл. ТЭЦ', 'г. Петропавловск', 4);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (10, 'Ульбинский металлургический завод', '', 5);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (11, 'АУЭС', '', 6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (12, 'КЭС 1', '', 7);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (13, 'КЭС 2', '', 7);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (14, 'ПЭК1', '', 1);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (15, 'ПЭК2', '', 1);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (16, 'ПЭК3', '', 1);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (17, 'ПЭК4', '', 1);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (18, 'ПЭК5', '', 1);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (19, 'НЭС1', '', 2);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (20, 'КОП1', '', 5);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (21, 'КОП2', '', 5);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (22, 'МОП1', '', 5);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (23, 'МОП2', '', 5);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (24, 'МОП3', '', 5);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (25, 'КПП1', '', 6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (26, 'КПП2', '', 6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (27, 'КПП3', '', 6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (28, 'КПП4', '', 6);

```

```

INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (29,'КПП5',' ',6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (30,'КПП6',' ',6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (31,'КПП7',' ',6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (32,'МПП1',' ',6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (33,'МПП2',' ',6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (34,'МПП3',' ',6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (35,'МПП4',' ',6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (36,'МПП5',' ',6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (37,'МПП6',' ',6);
INSERT INTO BuyerSellers(pk_bs, name, address,
fk_bskind)VALUES (38,'МПП7',' ',6);
commit;

```

Заполнение таблицы Объекты карты

```

INSERT INTO MapObjects(fk_bs, name, kind, x, y,
map_type)VALUES (5,'Турбина 1', 1,25,25, 0);
INSERT INTO MapObjects(fk_bs, name, kind, x, y,
map_type)VALUES (5,'Турбина 2', 1,55,25, 0);
INSERT INTO MapObjects(fk_bs, name, kind, x, y,
map_type)VALUES (5,'Турбина 3', 1,75,25, 0);
INSERT INTO MapObjects(fk_bs, name, kind, x, y,
map_type)VALUES (5,'Магистраль', 3,50,50, 0);
INSERT INTO MapObjects(fk_bs, name, kind, x, y,
map_type)VALUES (5,' ', -1,25,50, 0);
INSERT INTO MapObjects(fk_bs, name, kind, x, y,
map_type)VALUES (5,' ', -1,75,50, 0);
INSERT INTO MapObjects(fk_bs, name, kind, x, y,
map_type)VALUES (5,'Шульбинская ГЭС', 100,50,50, 1);
commit;

```

Заполнение таблицы Линии Карты

```

INSERT INTO MapLines(fk_bs, name, fk_mo1, fk_mo2, kind,
fk_ml)VALUES (5,'Линия 1', 1,4,1,1);
INSERT INTO MapLines(fk_bs, name, fk_mo1, fk_mo2, kind,
fk_ml)VALUES (5,'Линия 2', 2,4,1,2);
INSERT INTO MapLines(fk_bs, name, fk_mo1, fk_mo2, kind,
fk_ml)VALUES (5,'Линия 3', 3,4,1,3);
INSERT INTO MapLines(fk_bs, name, fk_mo1, fk_mo2, kind,
fk_ml)VALUES (5,' ', 1,5,-1,1);

```

```

INSERT INTO MapLines(fk_bs, name, fk_mo1, fk_mo2, kind,
fk_ml)VALUES (5, '', 5, 4, -1, 1);
INSERT INTO MapLines(fk_bs, name, fk_mo1, fk_mo2, kind,
fk_ml)VALUES (5, '', 2, 4, -1, 2);
INSERT INTO MapLines(fk_bs, name, fk_mo1, fk_mo2, kind,
fk_ml)VALUES (5, '', 3, 6, -1, 3);
INSERT INTO MapLines(fk_bs, name, fk_mo1, fk_mo2, kind,
fk_ml)VALUES (5, '', 6, 4, -1, 3);
commit;

```

Заполнение таблицы Значения объектов карты

```

INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES (1, to_date('23.05.2013 09:00:00', 'DD.MM.YYYY
HH24:MI:SS'), 2, 3, 6, 6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES (1, to_date('23.05.2013 10:00:00', 'DD.MM.YYYY
HH24:MI:SS'), 2, 3, 6, 6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES (1, to_date('23.05.2013 11:00:00', 'DD.MM.YYYY
HH24:MI:SS'), 2, 3, 6, 6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES (1, to_date('23.05.2013 12:00:00', 'DD.MM.YYYY
HH24:MI:SS'), 2, 3, 6, 6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES (1, to_date('23.05.2013 13:00:00', 'DD.MM.YYYY
HH24:MI:SS'), 2, 3, 6, 6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES (1, to_date('23.05.2013 14:00:00', 'DD.MM.YYYY
HH24:MI:SS'), 2, 3, 6, 6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES (1, to_date('23.05.2013 15:00:00', 'DD.MM.YYYY
HH24:MI:SS'), 2, 3, 6, 6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES (1, to_date('23.05.2013 16:00:00', 'DD.MM.YYYY
HH24:MI:SS'), 2, 3, 6, 6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES (1, to_date('23.05.2013 17:00:00', 'DD.MM.YYYY
HH24:MI:SS'), 2, 3, 6, 6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES (1, to_date('23.05.2013 18:00:00', 'DD.MM.YYYY
HH24:MI:SS'), 2, 3, 6, 6);

```

```

INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(1,to_date('23.05.2013 19:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,3,6,6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(1,to_date('23.05.2013 20:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,3,6,6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(1,to_date('23.05.2013 21:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,3,6,6);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 09:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 10:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 11:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 12:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 13:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 14:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 15:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 16:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 17:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 18:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);

```



```

INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 19:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 20:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(2,to_date('23.05.2013 21:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 09:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 10:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 11:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 12:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 13:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 14:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 15:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 16:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 17:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 18:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);

```

```

INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 19:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 20:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
INSERT INTO MapObjectsValue(fk_mo, value_date, value_current,
value_voltage, value_power,
value_power_max)VALUES(3,to_date('23.05.2013 21:00:00','DD.MM.YYYY
HH24:MI:SS'), 2,2,4,4);
commit;

```

2.1.5 Представления базы данных

Представление `CurrentUserRole` показывает список пользователей с текущей ролью.

```

CREATE OR REPLACE FORCE VIEW "ENERGY"."CURRENTUSERROLE" AS
select bs.pk_bs, bs.name as bs_name, ur.fk_user,
ur.given_date
from userrole ur
inner join buyersellers bs on ur.fk_bs = bs.pk_bs
where ur.given_date in (
select max(ur2.given_date) as last_role
from userrole ur2
where ur2.fk_user = ur.fk_user
);

```

Представление `UserInfo` показывает текущую информацию о пользователе: номер сессии, учётные данные, данные роли.

```

CREATE OR REPLACE FORCE VIEW "ENERGY"."USERSINFO" AS
select uo.sess_id,
u.pk_user,
u.login,
u.name,
u.second_name,
u.last_name,
u.group_name,
urbs.pk_bs,
urbs.bs_name as bsname,
urbs.given_date,
uo.login_time
from energyusers u
left join usersonline uo on u.pk_user = uo.fk_user
left outer join currentuserrole urbs on u.pk_user =
urbs.fk_user;

```

Представление Roles полную информацию о предприятиях: данные предприятия и данные о его типе.

```
CREATE OR REPLACE FORCE VIEW "ENERGY"."BSROLES" AS
select bs.pk_bs,
       bs.name as bsname,
       bs.address as bsaddress,
       bsk.name as bskind_name,
       bsk.pk_bskind,
       bsk.bskind_type
from buyersellers bs
inner join bskind bsk on bs.fk_bskind = bsk.pk_bskind;
```

Представление Price краткую информацию о заявках покупки.

```
CREATE OR REPLACE FORCE VIEW "ENERGY"."PRICE" AS
select r.pk_request,
       r.request_type,
       r.create_date,
       r.market_type,
       b.name as bs_name,
       b.pk_bs,
       sum(v.power) as power,
       sum(v.money) as currency
from energy.request r
inner join energy.buyersellers b on r.fk_bs1 = b.pk_bs
inner join energy.requestvolume v on r.pk_request =
v.fk_request
group by r.pk_request, b.name, b.pk_bs, r.request_type,
r.create_date, r.market_type;
```

2.1.6 Функции и процедуры

Функции и процедуры были сгруппированы в 2 пакета: Energy_sys и Energy.

Пакет Energy_Sys.

Данный пакет хранит в себе функции и процедуры которыми могут пользоваться только операторы рынка. Эти функции позволяют редактировать пользователей, предприятия, роли; запускать, останавливать и менять рынок; запускать составление договоров и управлять договорами.

```
create or replace package energy_sys is

-- Author   : VADIMREX
-- Created  : 06.10.2013 14:41:39
-- Purpose  :
```

```

type RefCur is ref cursor;

function GetValByName (key varchar2) return varchar2;

procedure SetValByName(key varchar2, val varchar2);

procedure createcontracts;

function AddUser(p_login varchar2,
                p_pass varchar2,
                p_l_name varchar2,
                p_f_name varchar2,
                p_s_name varchar2,
                p_group_name varchar2,
                p_isworkplace integer default 0)
                return integer;

function UserEdit(p_pk_user number,
                 p_pass varchar2,
                 p_last_name varchar2,
                 p_first_name varchar2,
                 p_second_name varchar2,
                 p_group_name varchar2,
                 p_pk_bs number,
                 p_is_admin number) return number;

procedure UserDelete(p_pk_user in number);

procedure getUserInfo(p_pk_user number,
                    p_login out varchar2,
                    p_last_name out varchar2,
                    p_first_name out varchar2,
                    p_second_name out varchar2,
                    p_group_name out varchar2,
                    p_pk_bs out number,
                    p_bs_name out varchar2,
                    p_is_admin out number);

procedure AddRole(p_pk_usr number,
                 p_pk_bs number);

procedure SetBSInfo(p_pk_bs number,
                   p_bs_name varchar2,
                   p_bs_address varchar2,
                   p_pk_bskind number);

procedure SetBSKindInfo(p_pk_bs_kind number,
                       p_bsk_name varchar2,
                       p_bsk_full_name varchar2,
                       p_bsk_type number);

end energy_sys;

```

```

/
create or replace package body energy_sys is

type tnums is record
(
    val NUMBER(32)
);

type t_tnums is table of tnums;

-- получить значение переменной
function getvalbyname (key varchar2) return varchar2
is
res varchar2(1000);
begin
    select o.value_string
    into res
    from options o
    where o.name = key;
    return res;
end getvalbyname;

-- установить значение переменной
procedure setvalbyname(key varchar2, val varchar2)
is
begin
    update options o
    set o.value_string = val
    where o.name = key;
    commit;
end setvalbyname;

-- открыть рынок
procedure open_market(p_market_type number,
                    p_date_end varchar2)
is
begin
    null;
end open_market;

-- создать договоры
procedure createcontracts
is

    pk_sale number(32);
    fk_bs1 number(32);
    slp_power_value number(32);
    slp_currency number(32);
    sl_date timestamp;
    slp_date timestamp;
    pk_buying number(32);
    fk_bs2 number(32);
    byp_power_value number(32);

```

```

    byp_currency number(32);
    by_date timestamp;
    byp_date timestamp;
    market_type number(32);
    cur_time timestamp;

    tab t_tnums;
begin
    market_type := getvalbyname('market');
    cur_time := getvalbyname('date_end');
    if market_type = 0 then
        -- составление для балансирующего рынка
        -- список заявок продажи
        for cur1 in (select r.pk_request, r.fk_bs1,
r.create_date, v.power, v.money, v.date_start, v.date_end
                    from request r
                    inner join requestvolume v on r.pk_request =
v.fk_request
                    where r.request_type = 1
                          and r.market_type = 0
                          and to_date(to_char(v.date_start, 'YYYY-
MM-DD HH24')||':00:00', 'YYYY-MM-DD HH24:MI:SS') =
to_date(cur_time, 'YYYY-MM-DD HH24:MI:SS')
                          and r.request_status <> 1
                    order by (v.money / v.power), r.create_date
                    asc)
            loop
                -- список заявок покупок
                for cur2 in (select r.pk_request, r.fk_bs1,
r.create_date, v.power, v.money, v.date_start, v.date_end
                            from request r
                            inner join requestvolume v on r.pk_request =
v.fk_request
                            where r.request_type = 2
                                  and r.market_type = 0
                                  and to_date(to_char(v.date_start,
'YYYY-MM-DD HH24')||':00:00', 'YYYY-MM-DD HH24:MI:SS') =
to_date(cur_time, 'YYYY-MM-DD HH24:MI:SS')
                                  and r.request_status <> 1
                                  and r.pk_request <> cur1.pk_request
                            order by (v.money / v.power) desc,
r.create_date asc)
                    loop
                        -- переписываем значения мощностей в массивы, если
ещё не переписывали
                        if tab(cur1.pk_request).val is null then
                            tab(cur1.pk_request).val := cur1.power;
                        end if;
                        if tab(cur2.pk_request).val is null then
                            tab(cur2.pk_request).val := cur2.power;
                        end if;
                        -- сравнение заявок

```

```

        if tab(curl1.pk_request).val =
tab(cur2.pk_request).val then
    tab(curl1.pk_request).val := 0;
    tab(cur2.pk_request).val := 0;
    -- если объём энергии одинаков
    update request r set
        r.request_status = 1
    where r.pk_request = curl1.pk_request
        or r.pk_request = cur2.pk_request;
    insert into documents d
        (d.fk_request1, d.fk_request2)
    values
        (curl1.pk_request, cur2.pk_request);
    commit;
    exit;
    elsif tab(curl1.pk_request).val >
tab(cur2.pk_request).val then
    tab(curl1.pk_request).val :=
tab(curl1.pk_request).val -
tab(cur2.pk_request).val;
    tab(cur2.pk_request).val := 0;
    -- если объём покупки меньше продажи
    update request r set
        r.request_status = 2
    where r.pk_request = curl1.pk_request;
    update request r set
        r.request_status = 1
    where r.pk_request = cur2.pk_request;
    insert into documents d
        (d.fk_request1, d.fk_request2)
    values
        (curl1.pk_request, cur2.pk_request);
    commit;
    elsif tab(curl1.pk_request).val <
tab(cur2.pk_request).val then
    tab(cur2.pk_request).val :=
tab(cur2.pk_request).val -
tab(curl1.pk_request).val;
    tab(curl1.pk_request).val := 0;
    -- если объём пкупки больше продажи
    update request r set
        r.request_status = 1
    where r.pk_request = curl1.pk_request;
    update request r set
        r.request_status = 2
    where r.pk_request = cur2.pk_request;
    insert into documents d
        (d.fk_request1, d.fk_request2)
    values
        (curl1.pk_request, cur2.pk_request);

```

```

        commit;
        exit;
    end if;
end loop;
end loop;
end if;
end createcontracts;

-- добавить пользователя, необходимо право create user у
самого пользователя, не роли!
function AddUser(p_login varchar2, -- логин, оно же и
оракловое имя
                p_pass varchar2, -- пароль, если рабочее
место то не используется
                p_l_name varchar2, -- фамилия
                p_f_name varchar2, -- имч
                p_s_name varchar2, -- отчество
                p_group_name varchar2, -- название группы
                p_isworkplace integer default 0) -- 0 -
пользователь, 1 - рабочее место
return integer
--AUTHID CURRENT_USER AS
is
query varchar2(1000);
begin
    query := 'create user '||p_login||' identified ';
    if p_isworkplace = 1 then
        query := query || 'externally ';
    else
        query := query || 'by '||p_pass||' ';
    end if;
    query := query || 'default tablespace Energy ';
    query := query || 'temporary tablespace Temp ';
    query := query || 'quota 5000M on Energy ';
    query := query || 'account unlock';

EXECUTE IMMEDIATE query;

query := 'grant connect to '||p_login;

EXECUTE IMMEDIATE query;

query := 'grant EnergyUser to '||p_login;

EXECUTE IMMEDIATE query;

insert into EnergyUsers u
        (u.login,          name,          second_name, last_name,
group_name,  isadmin, isworkplace,      registration)
        values(UPPER(p_login), p_f_name, p_s_name,      p_l_name,
p_group_name, 0,          p_isworkplace, sysdate);

```



```

        dbms_alert.signal('Users', 'i');

        commit;
        return 1;
    exception
        when others then
            dbms_output.put_line(TO_CHAR(SQLCODE) || '-' ||
SQLERRM);
            energy.log_save('adduser:
-' || TO_CHAR(SQLCODE) || ': ' || SQLERRM);
            return SQLCODE;
    end AddUser;

    -- ДОБАВИТЬ ПОЛЬЗОВАТЕЛЮ РОЛЬ
    procedure AddRole(p_pk_usr number, -- id пользователя
        p_pk_bs number) -- id роли
    is
        cur_time timestamp;
    begin
        cur_time := sysdate;
        insert into userrole
            (fk_user, fk_bs, given_date)
        values
            (p_pk_usr, p_pk_bs, cur_time);
        commit;
    end AddRole;

    function UserEdit(p_pk_user number,
        p_pass varchar2,
        p_last_name varchar2,
        p_first_name varchar2,
        p_second_name varchar2,
        p_group_name varchar2,
        p_pk_bs number,
        p_is_admin number) return number
    is
        query varchar2(1000);
        l_login varchar2(100);
    begin
        if p_pass is not null then
            begin
                select l.login
                into l_login
                from energyusers l
                where l.pk_user = p_pk_user;
                query := 'alter user ' || l_login || ' identified by
' || p_pass;

                execute immediate query;
            exception
                when others then
                    return 1;
        end if;
    end UserEdit;

```

```

        end;
    end if;

    update energyusers e set
        e.name = p_first_name,
        e.second_name = p_second_name,
        e.last_name = p_last_name,
        e.group_name = p_group_name,
        e.isadmin = p_is_admin
    where e.pk_user = p_pk_user;

    dbms_alert.signal('Users', 'u'||p_pk_user);

    commit;

    if p_pk_bs is not null then
        insert into userrole u
            (u.fk_user, u.fk_bs, u.given_date)
        values
            (p_pk_user, p_pk_bs, sysdate);
        commit;
    end if;

    return 0;
exception
    when others then
        return 2;
end UserEdit;

procedure UserDelete(p_pk_user in number)
is
    l_login varchar2(255);
begin
    select e.login
    into l_login
    from energyusers e
    where e.pk_user = p_pk_user;
    if lower(l_login) NOT IN ('anonymous', 'apex_030200',
'apex_public_user', 'appqossys',
'ctxsys', 'dbsnmp', 'dip',
'exfsys', 'flows_files',
'java', 'mddata', 'mdsys',
'mgmt_view',
'olapsys', 'oracle_ocm',
'orddata',
'ordplugins', 'ordsys', 'outln',
'owbsys', 'owbsys_audit',
'public',
'scott', 'si_informtn_schema',
'spatial_csw_admin_usr',
'spatial_wfs_admin_usr',
'sys', 'system',

```

```

'xs$null',
'sh',
'sysman', 'wmsys', 'xdb',
'bi', 'hr', 'ix', 'oe', 'pm',
'energy') then
delete from energyusers where pk_user = p_pk_user;
execute immediate 'drop user '||l_login;

dbms_alert.signal('Users', 'd'||p_pk_user);

commit;
end if;
exception
when others then null;
end UserDelete;

procedure getUserInfo(p_pk_user in number,
                    p_login out varchar2,
                    p_last_name out varchar2,
                    p_first_name out varchar2,
                    p_second_name out varchar2,
                    p_group_name out varchar2,
                    p_pk_bs out number,
                    p_bs_name out varchar2,
                    p_is_admin out number)
is
begin
select e.login,
       e.last_name,
       e.name,
       e.second_name,
       e.group_name,
       nvl(e.isadmin, 0),
       nvl(c.pk_bs, 0),
       c.bs_name
into
       p_login,
       p_last_name,
       p_first_name,
       p_second_name,
       p_group_name,
       p_is_admin,
       p_pk_bs,
       p_bs_name
from energyusers e
left outer join currentuserrole c on e.pk_user = c.fk_user
where e.pk_user = p_pk_user;
end getUserInfo;

-- Обновить/Добавить данные о предприятии
procedure SetBSInfo(p_pk_bs number,
                  p_bs_name varchar2,

```

```

                p_bs_address varchar2,
                p_pk_bskind number)
is
pk number(32);
begin
    begin
        select bs.pk_bs
        into pk
        from buyersellers bs
        where bs.pk_bs = p_pk_bs;
    exception
        when no_data_found then
            pk := -1;
    end;
    if pk = -1 then
        insert into buyersellers
            (name, address, fk_bskind)
        values
            (p_bs_name, p_bs_address, p_pk_bskind);

        dbms_alert.signal('BuyerSellers', 'i');
    else
        update buyersellers bs set
            bs.name = p_bs_name,
            bs.address = p_bs_address,
            bs.fk_bskind = p_pk_bskind
        where
            bs.pk_bs = p_pk_bs;

        dbms_alert.signal('BuyerSellers', 'u'||p_pk_bs);
    end if;

    commit;
end SetBSInfo;

-- Обновить/Добавить данные о виде предприятия
procedure SetBSKindInfo(p_pk_bs_kind number,
                        p_bsk_name varchar2,
                        p_bsk_full_name varchar2,
                        p_bsk_type number)
is
pk number(32);
begin
    begin
        select bsk.pk_bskind
        into pk
        from bskind bsk
        where bsk.pk_bskind = p_pk_bs_kind;
    exception
        when no_data_found then
            pk := -1;
    end;
end;

```

```

if pk = -1 then
    insert into bskind
        (name, name_full, bskind_type)
    values
        (p_bsk_name, p_bsk_full_name, p_bsk_type);

    dbms_alert.signal('BSKind', 'i');
else
    update bskind bsk set
        bsk.name = p_bsk_name,
        bsk.name_full = p_bsk_full_name,
        bsk.bskind_type = p_bsk_type
    where
        bsk.pk_bskind = p_pk_bs_kind;

    dbms_alert.signal('BSKind', 'u' || p_pk_bs_kind);
end if;

commit;
end SetBSKindInfo;

-- удалить роль
procedure DeleteBS(p_pk_bs number)
is
begin
    delete from buyersellers where pk_bs = p_pk_bs;

    dbms_alert.signal('BuyerSellers', 'd' || p_pk_bs);

    commit;
end DeleteBS;

-- удалить вид роли
procedure DeleteBSKind(p_pk_bs_kind number)
is
begin
    delete from bskind where pk_bskind = p_pk_bs_kind;
    dbms_alert.signal('BSKind', 'd' || p_pk_bs_kind);
    commit;
end DeleteBSKind;

begin
    -- Initialization
    null;
end energy_sys;
/

```

Пакет Energy. Данный пакет реализует процедуры и функции которыми могут пользоваться все участники рынка. Эти функции позволяют работать с заявками.

```

create or replace package Energy is

  -- Author   : VADIMREX
  -- Created  : 04.10.2013 9:12:28
  -- Purpose  :

  -- Public type declarations

  type RefCur is ref cursor;

  function GetValByName (key varchar2) return varchar2;

  procedure log_save (p_message varchar2);

  procedure Loggin (p_login varchar2,
                   p_status out integer,
                   p_sess_id out integer);

  procedure SessionLive(p_sess_id number,
                       p_res out number,
                       p_role out number,
                       p_role_type out number,
                       p_status out number,
                       p_market out number);

  function GetBS(p_sess_id number)
             return number;

  function SendRequest(p_pk_bs1 number,
                     p_market_kind number,
                     p_pk_bs2 number default null,
                     p_price number default null,
                     p_power number default null) return
number;

  procedure RequesPeriodAdd(p_pk_request number,

                           p_date_start varchar2,

                           p_date_end varchar2,

                           p_price number,

                           p_power number);

  function GetWindowTable(p_sess_id number,
                         p_window_name varchar2,
                         params varchar2) return refcur;

  procedure GetBSInfo(p_pk_bs number,
                    p_bs_name out varchar2,
                    p_bs_address out varchar2,

```

```

        p_pk_bskind out number,
        p_bs_kind_name out varchar2);

procedure GetBSKindInfo(p_pk_bs_kind number,
                        p_bsk_name out varchar2,
                        p_bsk_full_name out varchar2,
                        p_bsk_type out number);

procedure shortbuy(p_pk_request number,
                  p_sessid number);

procedure longbuy(p_pk_request number,
                 p_sessid number);

procedure web_info;

end Energy;
/
create or replace package body Energy is

    -- получить значение переменной
function getvalbyname (key varchar2) return varchar2
is
res varchar2(1000);
begin
    select o.value_string
    into res
    from options o
    where o.name = key;
    return res;
end getvalbyname;

-- сохранение лога
procedure log_save (p_message varchar2)
is
begin
    insert into energy_log (message, write_date)
    values (p_message, SYSDATE);
    commit;
end log_save;

-- функция залогинивания, мало быть зарегистрированным в
базе оракла
procedure loggin (p_login varchar2, -- имя оракл
                 -- возвращаемые
                 p_status out integer, -- статус
                 p_sess_id out integer) -- id сессии
is
    l_is_adm number(1);
    l_is_workplace number(1);
    l_pk_usr number(32);
begin
    log_save('loggin:
    p_login=>'||p_login);

```

```

p_sess_id := 0;
p_status := 0;
begin
  select u.pk_user, u.isworkplace, u.isadmin
  into l_pk_usr, l_is_workplace, l_is_adm
  from EnergyUsers u
  where UPPER(u.login) = UPPER(p_login);

  delete from UsersOnline
  where fk_user = l_pk_usr;

  insert into UsersOnline u (u.fk_user, u.login_time)
values (l_pk_usr, sysdate)
  returning u.sess_id into p_sess_id;
  commit;

  p_status := 1;

  if l_is_adm = 1 then p_status := p_status + 2; end if;
  if l_is_workplace = 1 then p_status := p_status + 4; end
if;

  log_save('loggin:
p_status=>'||p_status||'
p_sess_id=>'||p_sess_id);
exception
  when NO_DATA_FOUND THEN
  p_sess_id := -1;
  p_status := -1;

  log_save('loggin:
-'||SQLCODE||' '||SQLERRM);
end;
end loggin;

-- функция общения с сервером, поддерживает сессию и
возвращает текущее состояние рынка
procedure sessionlive(p_sess_id number, -- id сессии
-- возвращаемые параметры
p_res out number, -- результат
p_role out number, -- роль
p_role_type out number, -- тип роли
p_status out number, -- статус рынка
p_market out number) -- вид рынка, 0 -
балансирующий, 1 - короткосрочный, 2 - долгосрочный
is
  l_cur_time timestamp;
  l_fk_user number(32);
  i number(32);
begin
  log_save('sessionlive:

```



```

    p_sess_id=>'||p_sess_id);
l_cur_time := sysdate;
p_res := 0;
p_role := 0;
p_role_type := 0;
if p_sess_id <> -1 then
    begin
        select u.fk_user
        into l_fk_user
        from usersonline u
        where u.sess_id = p_sess_id;
    exception
        when no_data_found then
            l_fk_user := -1;
    end;
    if l_fk_user > 0 then
        update usersonline u
        set u.last_update = l_cur_time
        where u.sess_id = p_sess_id;
        commit;

        p_res := 1;

        begin
            select ui.pk_bs, br.pk_bskind
            into p_role, p_role_type
            from usersinfo ui
            inner join bsroles br on ui.pk_bs = br.pk_bs
            where ui.sess_id = p_sess_id;
        exception
            when no_data_found then
                select u.isadmin
                into i
                from EnergyUsers u
                where u.pk_user = l_fk_user;
                if i = 0 then
                    p_res := 0;
                end if;
            end;
        end if;
    end if;
end if;

delete from usersonline -- TODO: ia?aianoe a job
where last_update < l_cur_time - INTERVAL '30' MINUTE;

p_status := energy_sys.getvalbyname('status');
p_market := energy_sys.getvalbyname('market');

log_save('sessionlive:
    p_sess_id=>'||p_sess_id);
end sessionlive;

```

```

-- отправка заявки
function SendRequest(p_pk_bs1 number,
                    p_market_kind number,
                    p_pk_bs2 number default null,
                    p_price number default null,
                    p_power number default null) return
number
is
    l_bs_kind number(1);
    l_market_kind number(1);
    l_market_status number(1);
    l_pk_request number(32);
    l_date_start timestamp;
    l_date_end timestamp;
    cur_time timestamp;
begin
    cur_time := to_date(energy_sys.GetValByName('date_end'),
'YYYY-MM-DD HH24:MI:SS');
    -- тип вида роли
    select r.bskind_type
    into l_bs_kind
    from BSRoles r
    where r.pk_bs = p_pk_bs1;
    -- информация о рынке
    l_market_kind := energy_sys.GetValByName('market');
    l_market_status := energy_sys.GetValByName('status');
    -- рынок не изменился и функционирует
    if p_market_kind = l_market_kind and l_market_status = 1
then
    -- добавить заявку
    insert into Request r
        (r.fk_bs1, r.fk_bs2, r.request_type,
r.market_type, r.create_date)
        values
            (p_pk_bs1, p_pk_bs2, l_bs_kind, l_market_kind,
cur_time)
    returning r.pk_request into l_pk_request;
    -- если балансирующий рынок сразу добавить и объём
    if l_market_kind = 0 then
        l_date_start := cur_time;
        l_date_end := to_date(to_char(cur_time, 'YYYY-MM-DD
')||(to_char(cur_time, 'HH24')+1)||':00:00','YYYY-MM-DD
HH24:MI:SS');
        insert into RequestVolume r
            (r.fk_request, r.power, r.money, r.date_start,
r.date_end)
            values
                (l_pk_request, p_power, p_price, l_date_start,
l_date_end);
    end if;

```

```

        -- сигнал что "a" добавил на "l_market_kind" рынок заявку
        "l_bs_kind" типа

dbms_alert.signal('Market','a' || l_market_kind || l_bs_kind || p_pk_bs1
|| ' ' || p_pk_bs2);

        commit;
        return l_pk_request;
    else
        return 0;
    end if;
end SendRequest;

-- добавка периода к заявке
procedure RequesPeriodAdd(p_pk_request number,
p_date_start varchar2,
p_date_end varchar2,
p_price number,
p_power number)
is
err EXCEPTION;
begin
    log_save('RequesPeriodAdd:
    p_pk_request=>' || p_pk_request || '
    p_date_start=>' || p_date_start || '
    p_date_end=>' || p_date_end || '
    p_price=>' || p_price || '
    p_power=>' || p_power);

        insert into RequestVolume r
            (r.fk_request, r.power, r.money,
r.date_start, r.date_end)
            values
                (p_pk_request, p_power, p_price,
to_date(p_date_start, 'YYYY-MM-DD HH24:MI:SS'),
to_date(p_date_end, 'YYYY-MM-DD HH24:MI:SS'));

        dbms_alert.signal('Market','p' || p_pk_request);

        commit;
exception
    when others then
        log_save('RequesPeriodAdd:
        -' || SQLCODE || ' ' || SQLERRM);
        RAISE_APPLICATION_ERROR(SQLCODE, SQLERRM);
end RequesPeriodAdd;

-- вытаскивает нужную таблицу, с правильным селектом

```

```

function GetWindowTable(p_sess_id number,
                       p_window_name varchar2,
                       params varchar2) return refcur
is
ret RefCur;
l_pk_user number(32); -- пк пользоваеля
l_pk_bs number(32); -- пк роли
l_bskind_type number(1); -- тип вида роли
begin
-- пк пользователя по сессии
begin
select u.fk_user
into l_pk_user
from usersonline u
where u.sess_id = p_sess_id;
exception
when no_data_found then
return null;
end;
-- информация о пользователе
select ui.pk_bs, bk.bskind_type
into l_pk_bs, l_bskind_type
from usersinfo ui
inner join buyersellers bs on ui.pk_bs = bs.pk_bs
inner join bskind bk on bs.fk_bskind = bk.pk_bskind
where ui.pk_user = l_pk_user;

ret := null;
case p_window_name
-- балансирующий рынок - выборка окна со своими заявками
when 'balance' then
if l_bskind_type in (1,2) then
open ret for
select *
from energy.price p
where p.request_type = l_bskind_type
and p.pk_bs = l_pk_bs
and p.market_type = 0;
end if;
-- короткосрочный рынок:
when 'shortleft' then -- левое окошко - выборка своих
заявок
if l_bskind_type = 1 then
open ret for
select *
from energy.price p
where p.request_type = 1
and p.pk_bs = l_pk_bs
and p.market_type = 1;
elsif l_bskind_type = 2 then
open ret for
select *

```

```

        from energy.price p
        where p.request_type = 2
              and p.pk_bs = l_pk_bs
              and p.market_type = 1;
    end if;
    when 'shortright' then -- правое окошко - выборка заявок
на которые можно поставить ставку
        if l_bskind_type = 1 then
            open ret for
                select *
                from energy.price p
                where p.request_type = 2
                      and p.market_type = 1;
        elsif l_bskind_type = 2 then
            open ret for
                select *
                from energy.price p
                where p.request_type = 1
                      and p.market_type = 1;
        end if;
    when 'shortcenter' then -- центральное окошко - выборка
заявок на которые была поставлена ставка
        if l_bskind_type = 1 then
            open ret for
                select *
                from energy.price p
                inner join energy.requestauction ra on
ra.pk_request = p.pk_request
                where ra.pk_bs = l_pk_bs
                      and p.market_type = 1;
        elsif l_bskind_type = 2 then
            open ret for
                select *
                from energy.price p
                inner join energy.requestauction ra on
ra.pk_request = p.pk_request
                where ra.pk_bs = l_pk_bs
                      and p.market_type = 1;
        end if;
    -- долгосрочный рынок:
    when 'longleft' then -- левое окошко - выборка своих
заявок
        if l_bskind_type = 1 then
            open ret for
                select *
                from energy.price p
                where p.request_type = 1
                      and p.pk_bs = l_pk_bs
                      and p.market_type = 2;
        elsif l_bskind_type = 2 then
            open ret for
                select *

```

```

        from energy.price p
        where p.request_type = 2
              and p.pk_bs = l_pk_bs
              and p.market_type = 2;
    end if;
    when 'longright' then -- правое окошко - выборка заявок
    которые можно удовлетворить
        if l_bskind_type = 1 then
            open ret for
            select *
            from energy.price p
            where p.request_type = 2
                  and p.market_type = 2;
        elsif l_bskind_type = 2 then
            open ret for
            select *
            from energy.price p
            where p.request_type = 1
                  and p.market_type = 2;
        end if;
    -- справочник видов ролей
    when 'roletypes' then
        open ret for
        select t.*,
               decode(bskind_type,
                      0, 'посредник',
                      1, 'производитель электроэнергии',
                      2, 'потребитель электроэнергии',
                      '') as bskind
        from energy.bskind t;
    -- справочник ролей, параметр влияет на фильтрование по
    типу
    when 'roles' then
        open ret for
        select bs.pk_bs, bs.name, bs.address, bk.name as
        bskind_name
        from energy.buyersellers bs, energy.bskind bk
        where bs.fk_bskind = bk.pk_bskind (+) and
              bk.bskind_type = decode(params, '',
        bk.bskind_type, params);
    else
        open ret for select * from dual;
    end case;
    return ret;
end GetWindowTable;
-- auoaeou ?ieu iieuciaaoaey ii nannee
function Getbs(p_sess_id number) -- id nannee
    return number -- id ?iee
is
l_pk_bs number(32);
begin
    select ui.pk_bs

```

```

    into l_pk_bs
        from usersinfo ui
        where ui.sess_id = p_sess_id;
    return l_pk_bs;
end Getbs;
-- Данные о предприятии
procedure GetBSInfo(p_pk_bs number,
                   p_bs_name out varchar2,
                   p_bs_address out varchar2,
                   p_pk_bskind out number,
                   p_bs_kind_name out varchar2)
is
begin
    SELECT bs.name, bs.address, bs.fk_bskind, bsk.name
    INTO p_bs_name, p_bs_address, p_pk_bskind, p_bs_kind_name
    FROM BuyerSellers bs, BSKind bsk
    WHERE bs.pk_bs = p_pk_bs and
          bs.fk_bskind = bsk.pk_bskind (+);
end GetBSInfo;
-- Данные о виде предприятия
procedure GetBSKindInfo(p_pk_bs_kind number,
                       p_bsk_name out varchar2,
                       p_bsk_full_name out varchar2,
                       p_bsk_type out number)
is
begin
    SELECT bsk.name, bsk.name_full, bsk.bskind_type
    INTO p_bsk_name, p_bsk_full_name, p_bsk_type
    FROM BSKind bsk
    WHERE bsk.pk_bskind= p_pk_bs_kind;
end GetBSKindInfo;

procedure shortbuy(p_pk_request number,
                  p_sessid number)
is
begin
    null;
end shortbuy;

procedure longbuy(p_pk_request number,
                  p_sessid number)
is
l_pk_bs number(32);
begin
    l_pk_bs := getbs(p_sessid);
    update request set
        fk_bs2 = l_pk_bs,
        request_status = 1
    where pk_request = p_pk_request;
    commit;
end longbuy;

```

```

-- web отображение
procedure web_info
is
begin
  http.htmlOpen;
  http.headOpen;
  http.title('Информация о рынке');
  http.headClose;
  http.bodyOpen;
  http.htitle('Информация о рынке');
  http.bodyClose;
  http.htmlClose;
end;
begin
null;
  -- Initialization
end Energy;
/

```

2.1.7 Триггеры

На каждой таблице с первичным ключом есть триггер на создание первичного ключа.

```

CREATE OR REPLACE TRIGGER "ENERGY"."T_REQUEST_BI" BEFORE
INSERT ON "ENERGY"."REQUEST" REFERENCING OLD AS "OLD" NEW AS "NEW"
FOR EACH ROW ENABLE
declare
  n NUMBER(32);
begin
  select s_Request.nextval into n from dual;
  :NEW.pk_Request := n;

end t_Request_bi;

```

Триггер USERSONLINE для создания сессии.

```

CREATE OR REPLACE TRIGGER "ENERGY"."T_USERSONLINE_BI" FOR
INSERT ON "ENERGY"."USERSONLINE" REFERENCING OLD AS "OLD" NEW AS
"NEW" ENABLE
compound trigger
  l_fk_user NUMBER(32);
  l_sessid NUMBER(32);

before each row is
begin
  select s_UsersOnline.nextval into l_sessid from dual;
  :NEW.sess_id := l_sessid;
  l_fk_user := :NEW.FK_USER;
end before each row;

```



```

after statement is
begin
  delete from UsersOnline
  where fk_user = l_fk_user
  and sess_id <> l_sessid;
end after statement;

end t_UsersOnline_bi;

```

2.2 Описание интерфейса приложения. Работа с приложением

Интерфейс данного приложения очень удобен для использования. С приложением могут работать три вида пользователей: администратор, участник рынка, гость. Для администратора предоставлены все права, участник рынка может подавать и отзывать заявки, гость может только просматривать заявки.

После запуска приложения открывается первая форма – форма входа. На данном этапе производится авторизация пользователя (Рисунок 2.1). При вводе неправильного пароля системы выдаст сообщение об ошибке (Рисунок 2.2). Если вход выполнен под оператором рынка, об этом сразу уведомляется окошком предупреждения (Рисунок 2.3).

Проверка пароля и логина происходит на сервере через посредством аутентификации на сервере Oracle. Так же логин проверяется вызовом процедуры login чтобы проверить есть ли у пользователя доступ к системе.

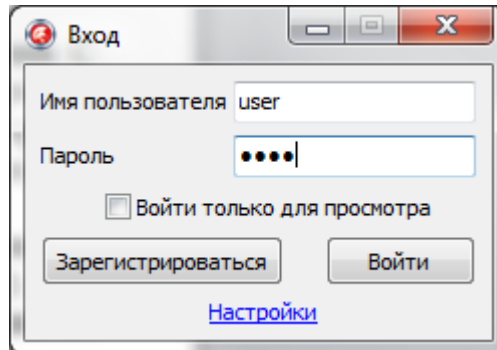


Рисунок 2.1 – Вводиться логин и пароль

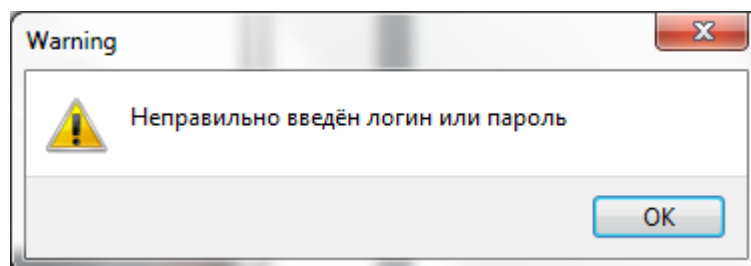


Рисунок 2.2 – Неправильно введён пароль

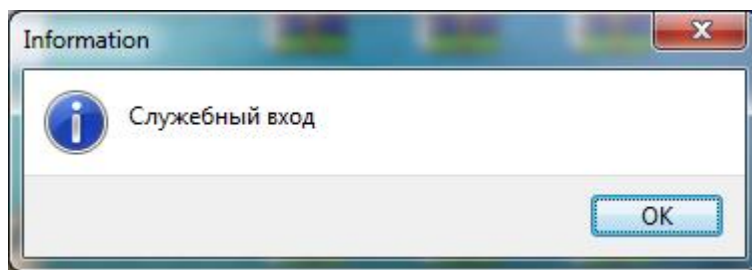


Рисунок 2.3 – Вход выполнен под оператором рынка

На главном окне отображается состояние рынка. Если рынок ещё не активен то пользователь увидит надпись Торги ещё не начались (Рисунок 2.4). Когда рынок активен или закончен, пользователь видит две таблицы: таблица заявок продажи и таблица заявок покупки (Рисунок 2.5).

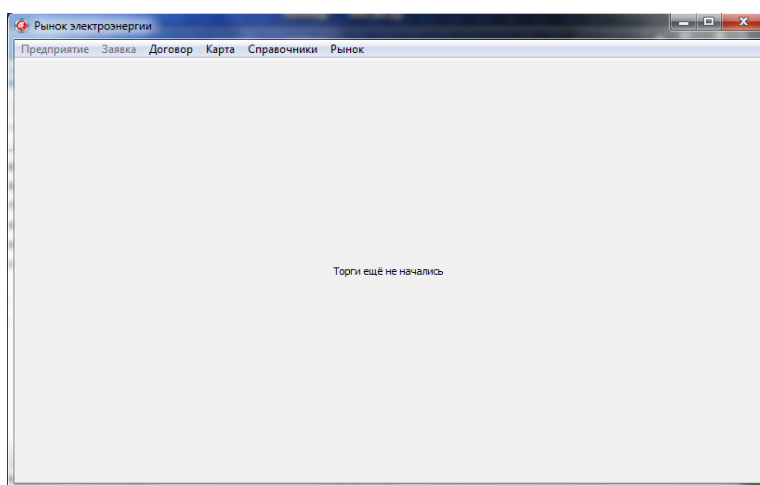


Рисунок 2.4 – Вход выполнен под обычным пользователем, торги не начались

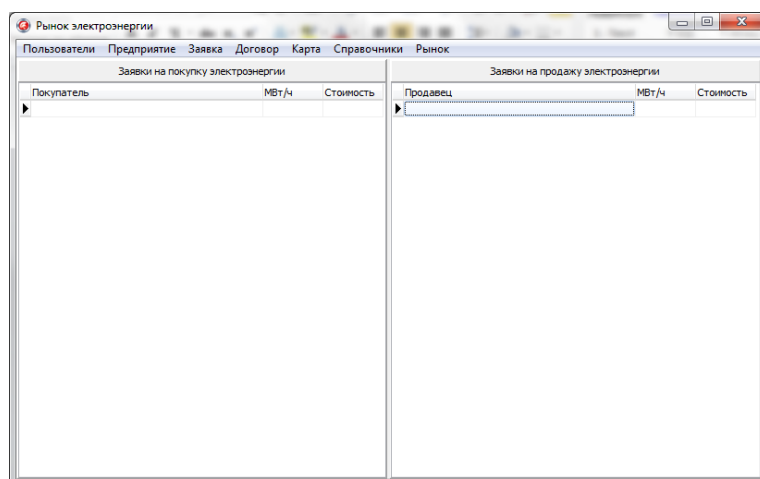


Рисунок 2.5 – Выполнен служебный вход под администратором, рынок активен

Администратор может управлять пользователями (Рисунок 2.6.), редактировать справочники (Рисунок 2.7.) и управлять рынком (Рисунок 2.8.).

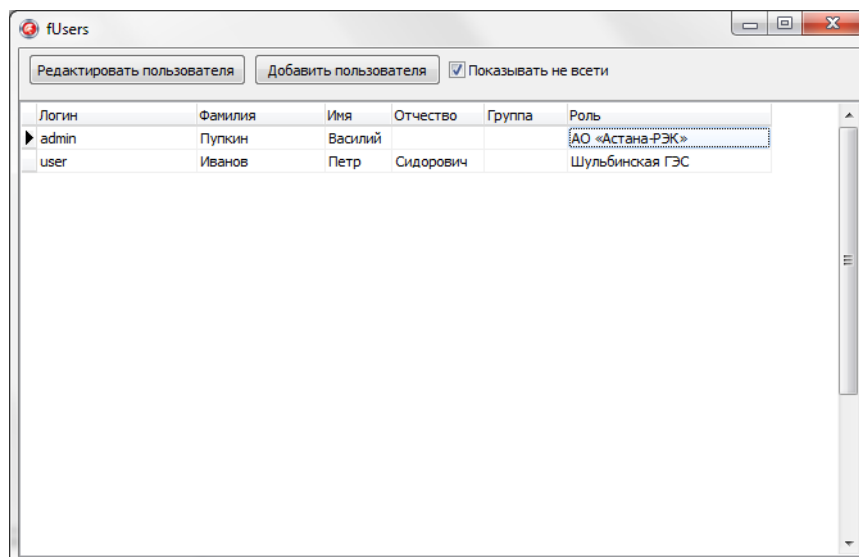


Рисунок 2.6 – Управление пользователями

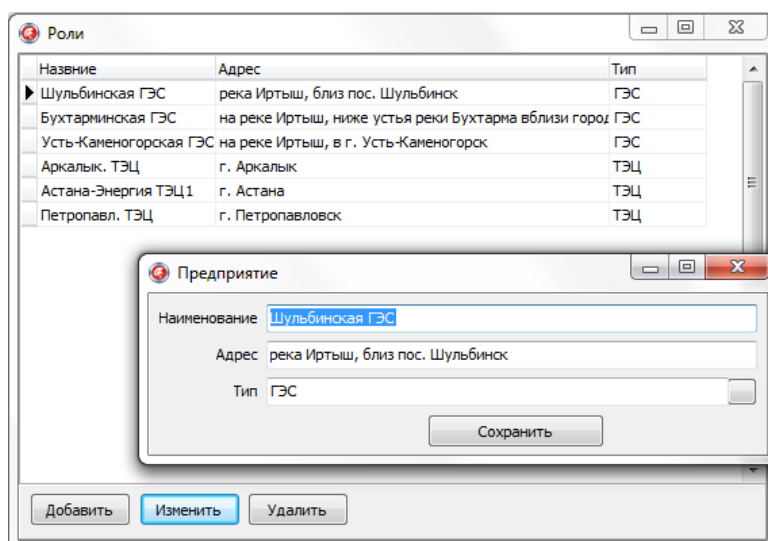


Рисунок 2.7 – Редактирование справочника

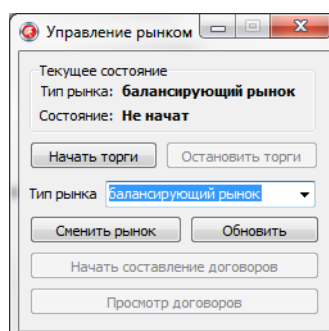


Рисунок 2.8 – Управление рынком

Заявки вносятся на форме заявка (Рисунок 2.9). Заявки могут подавать только те пользователи, которым прикрепили предприятие. Для краткосрочного

или долгосрочного рынка есть возможность просмотра заявки в виде графика (Рисунок 2.10).

Заявка

Дата начала: 16.05.2013

Количество месяцев: 7

Посмотреть в виде графика

Подать заявку

| | МВт/ч | тг/ч | МВт | тг |
|------------|-------|------|-----|----|
| 16.05.2013 | | | | |
| 16.06.2013 | | | | |
| 16.07.2013 | | | | |
| 16.08.2013 | | | | |
| 16.09.2013 | | | | |
| 16.10.2013 | | | | |
| 16.11.2013 | | | | |
| | 7 | 7 | 7 | 7 |

Рисунок 2.9 – Подача заявки

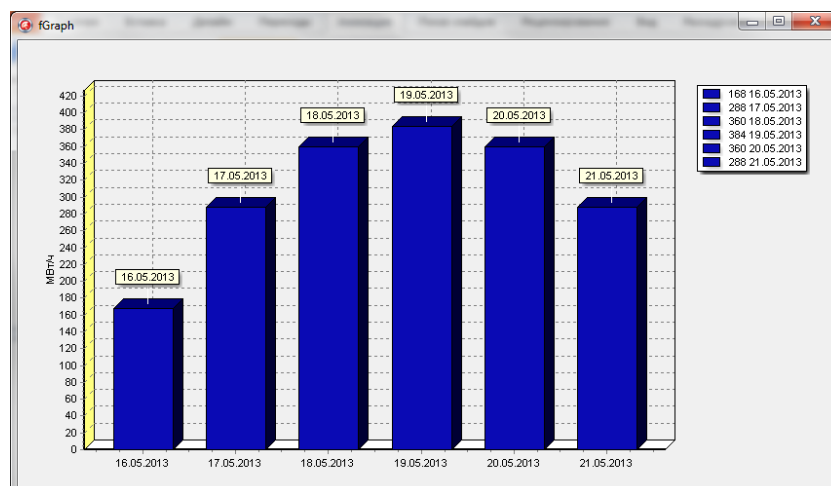


Рисунок 2.10 – Просмотр заявки в режиме графика

3 Безопасность жизнедеятельности

3.1 Анализ условий труда обслуживающего персонала при эксплуатации технического оборудования

Главной целью данного проекта является информационная система “Моделирование торгов рынка электроэнергии”.

В список опасных и вредных факторов при работе за компьютером входят:

- 1 повышенная напряженность электрического поля;
- 2 токсические вещества;
- 3 повышенный уровень шума на рабочем месте;
- 4 пониженная контрастность;
- 5 повышенная напряженность магнитного поля;
- 6 недостаточная освещенность рабочей зоны;
- 7 повышенный уровень статистического электричества.

Касательно здоровья сотрудников, можно выделить несколько факторов риска, которым сопровождается влияние компьютера на организм человека:

- 1 проблемы, обусловленные наличием электромагнитного излучения;
- 2 проблемы зрения;
- 3 проблемы, связанные с мышцами и суставами;
- 4 стресс, депрессия и другие нервные расстройства, которые обуславливаются влиянием компьютера на психику человека.
- 5 малоподвижный образ жизни;
- 6 переработка (более 9 часов в сутки);
- 7 стрессы;
- 8 работа в ночное время суток, и как следствие нарушение выработки гормона мелатонина.

3.1.1 Магнитное поле

Компьютер при работе создает вокруг себя электромагнитное поле, которое обладает способностью биологического специфического и теплового воздействия на организм человека. За счет влияния электромагнитного поля на клетки и ткани человека происходят нарушения условно–рефлекторной деятельности, снижение активности мозга. Все это проявляется в головной боли, утомляемости, ухудшении самочувствия, гипотонии.

За счет теплового воздействия электромагнитного поля повышается температура тела, идет нагрев тканей и органов. Больше всего подвержены тепловому облучению такие органы как печень, поджелудочная железа, мочевого пузырь, желудок. Все это может вызвать язвы кровотечения и перфорации [9].

3.1.2 Шум

На рабочем месте сотрудников источниками шума, как правило, являются разговаривающие люди, внешний шум и отчасти – компьютер, принтер. Они издают довольно незначительный шум, поэтому в помещении достаточно использовать звукопоглощение.

Из строительно–акустических методов защиты от шума (СНиП–II–12–77) выбран метод для помещения, представленного на рисунке 3.1 – план рабочего помещения с звукопоглощающими конструкцией и экранами.

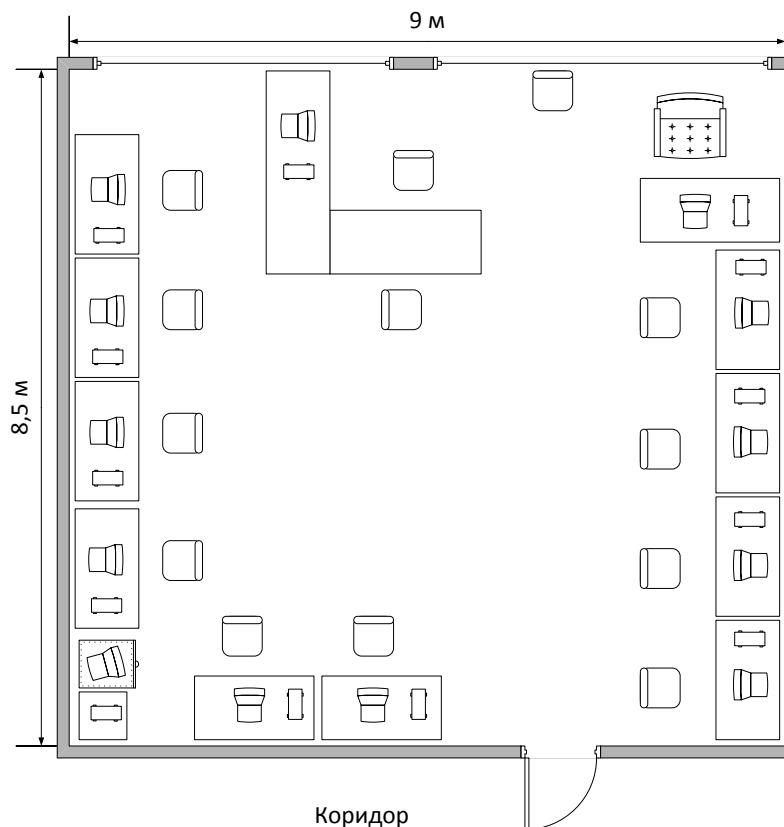


Рисунок 3.1 – План рабочего помещения

Для выбранного помещения выбрано звукопоглощающие облицовка состоящая из матов, из супертонкого стекловолокна с оболочкой из стеклоткани, которую нужно разместить на потолке и верхних частях стен. Максимальное звукопоглощение будет достигнуто при облицовке не менее 60 % общей площади ограждающих поверхностей помещения.

Электростатическое поле, вредные вещества в воздухе

При работе компьютер образует вокруг себя электростатическое поле, которое деионизирует окружающую среду, а при нагревании платы и корпус монитора испускают в воздух вредные вещества. Всё это делает воздух очень сухим, слабо ионизированным, со специфическим запахом и в общем "тяжёлым" для дыхания. Естественно, такой воздух не может быть полезен для

организма и может привести к заболеваниям аллергического характера, болезням органов дыхания и другим расстройствам [10].

3.1.3 В чем заключается вред работы ночью

Мелатонин – основной гормон эпифиза, регулятор суточного ритма. Вырабатывается ночью, и его максимальная концентрация достигается к 5 часам утра. Этот гормон регулирует функции клеточного обновления, нейтрализует разрушительные последствия окислительных процессов, которые являются основной причиной старения и увядания кожи, участвует в защите организма от неблагоприятных воздействий. Мелатонин очень важен для организма, и, поскольку выработка этого гормона происходит преимущественно ночью, то любой сбой сна приводит к снижению его выработки.

Известно, что в организме человека существуют так называемые биологические часы. Причем одни получают информацию о смене дня и ночи напрямую через зрительный канал, а другие – наш внутренний “счетчик времени”.

В нормальном режиме часы работают синхронно. Одни при наступлении темноты запускают “режим сна”, другие синхронно этому запускают соответствующие нейробиологические и гормональные процессы.

Достаточно просто просиживать большое количество времени перед ярким монитором, чтобы гарантированно испытать ослабленный эффект десинхронии. Работая в ночное время мы стимулируем эпифиз ярким светом монитора и вводим в заблуждение наши внутренние часы, которые “рассчитывают” на ночь. Как следствие такой работы – блокировка секреции мелатонина.

3.1.4 Технический персонал

Технический персонал состоит из одного сотрудника: системный администратор. В рабочем помещении кроме технического персонала могут находиться преподаватель и студенты, вместе до 12 человек.

Системный администратор работает каждый день, по 8 часов с учетом перерыва на обед, который составляет 1 час и он может выбрать время обеда с 12:00 – 15:00. С понедельника по пятницу. Выходной день воскресенье [11].

Работа сотрудников непосредственно связана с компьютером, а соответственно с вредным дополнительным воздействием целой группы факторов, что существенно снижает производительность их труда.

К таким факторам можно отнести:

- 1 неправильная освещенность;
- 2 нарушение микроклимата;
- 3 наличие напряжения.

Согласно ГОСТ 12.1.005–88 ССБТ «Оптимальные и допустимые нормы микроклимата, в зависимости от категории работ», работа людей в помещении относится к работе лёгкой тяжести(1а), так как управление оборудованием осуществляется дистанционно с помощью компьютеров [12].

С целью создания нормальных условий для работников предприятий Связи установлены нормы производственного микроклимата. В помещениях при работе с ПК должны соблюдаться следующие климатические условия [10]:

- Холодный период года:
 - оптимальная температура 24 С°, допустимая температура 26 С°;
 - относительная влажность 45 %, допустимая влажность 60%;
 - скорость движение воздуха относительная и допустимая 0,05 м/с.
- Тёплый период года:
 - оптимальная температура 23 С°, допустимая температура 25 С°;
 - относительная влажность 50 %, допустимая влажность 55%;
 - скорость движение воздуха относительная и допустимая 0,1 м/с.

Помещение имеет размеры: длина (L) = 9 метров, ширина (В) = 8,5 метров, высота (Н) = 3 метра. Помещение находится в здании на 4–м этаже, рассчитано на 12 рабочих мест.

План помещения выбранного для размещения оборудования и технического персонала изображен на рисунке 3.1.

Рабочее место состоит из следующих компонентов:

- тринадцати столов;
- пятнадцати стульев;
- один сейф;
- 13 компьютеров, один из которых сервер.

3.2 Организация рабочих мест

Рабочие места с персональными компьютерами по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, желательно слева.

Схемы размещения рабочих мест с персональными компьютерами должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м.

Рабочий стол может быть любой конструкции, отвечающей современным требованиям эргономики и позволяющей удобно разместить на рабочей поверхности оборудование с учетом его количества, размеров и характера выполняемой работы. Целесообразно применение столов, имеющих отдельную от основной столешницы специальную рабочую поверхность для размещения

клавиатуры. Используются рабочие столы с регулируемой и нерегулируемой высотой рабочей поверхности. При отсутствии регулировки высота стола должна быть в пределах от 680 до 800 мм.

Глубина рабочей поверхности стола должна составлять 800 мм (допускаемая не менее 600 мм), ширина – соответственно 1 600 мм и 1 200 мм. Рабочая поверхность стола не должна иметь острых углов и краев, иметь матовую или полуматовую фактуру.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм [13].

Моторное поле – пространство рабочего места, в котором могут осуществляться двигательные действия человека.

Максимальная зона досягаемости рук – это часть моторного поля рабочего места, ограниченного дугами, описываемыми максимально вытянутыми руками при движении их в плечевом суставе.

Оптимальная зона – часть моторного поля рабочего места, ограниченного дугами, описываемыми предплечьями при движении в локтевых суставах с опорой в точке локтя и с относительно неподвижным плечом (Рисунок 3.2).

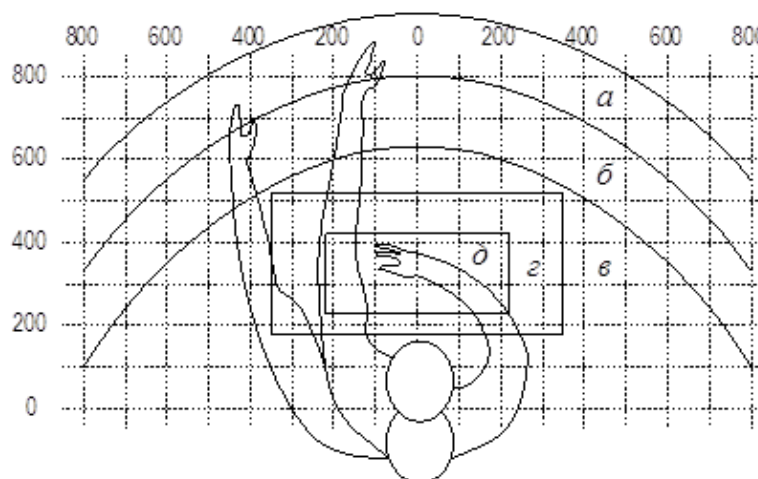


Рисунок 3.2 – Зоны досягаемости рук в горизонтальной плоскости

На рисунке 3.2 показаны Зоны досягаемости рук:

- а – зона максимальной досягаемости;
- б – зона досягаемости пальцев при вытянутой руке;
- в – зона легкой досягаемости ладони;
- г – оптимальное пространство для грубой ручной работы;
- д – оптимальное пространство для тонкой ручной работы.

Оптимальное размещение предметов труда и документации в зонах досягаемости (Рисунок 3.3):

- дисплей размещается в зоне а (в центре);
- системный блок размещается в предусмотренной нише стола;
- клавиатура – в зоне г/д;

- мышь – в зоне в справа;
- сканер в зоне а/б (слева);
- принтер находится в зоне а (справа);
- документация: необходимая при работе – в зоне легкой досягаемости ладони – в, а в выдвижных ящиках стола – литература, неиспользуемая постоянно.

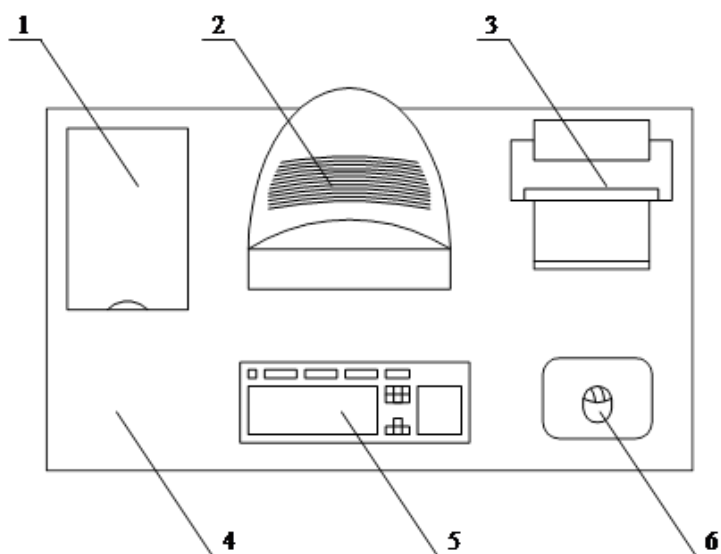


Рисунок 3.3 – Размещение основных и периферийных составляющих ПК

На рисунке 3.3 показан пример размещения основных и периферийных составляющих ПК на рабочем столе программиста:

- 1 сканер;
- 2 монитор;
- 3 принтер;
- 4 поверхность рабочего стола;
- 5 клавиатура;
- 6 манипулятор типа «мышь».

Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 градусов вниз от горизонтали).

Клавиатура должна располагаться на поверхности стола на расстоянии 100–300 мм от края, обращенного к пользователю.

Для удобства считывания информации с документов применяются подвижные подставки (пюпитры), размеры которых по длине и ширине соответствуют размерам устанавливаемых на них документов. Пюпитр размещается в одной плоскости и на одной высоте с экраном.

Для обеспечения физиологически рациональной рабочей позы, создания условий для ее изменения в течение рабочего дня применяются подъемно–

поворотные рабочие стулья с сиденьем и спинкой, регулируемые по высоте и углам наклона, а также расстоянию спинки от переднего края сидения.

Конструкция стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;
- регулировку высоты поверхности сиденья в пределах 400–550 мм и углом наклона вперед до 15 градусов и назад до 5 градусов.;
- высоту опорной поверхности спинки 300 ± 20 мм, ширину – не менее 380 мм и радиус кривизны горизонтальной плоскости 400 мм;
- угол наклона спинки в вертикальной плоскости в пределах 0 ± 30 градусов;
- регулировку расстояния спинки от переднего края сидения в пределах 260–400 мм;
- стационарные или съемные подлокотники длиной не менее 250 мм и шириной 50–70 мм;
- регулировку подлокотников по высоте над сиденьем в пределах 230 ± 30 мм и внутреннего расстояния между подлокотниками в пределах 350–500 мм;
- поверхность сиденья, спинки и подлокотников должна быть полумягкой, с нескользящим неэлектризующимся, воздухопроницаемым покрытием, легко очищаемым от загрязнения.

Рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 град. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

Режим труда и отдыха предусматривает соблюдение определенной длительности непрерывной работы на ПК и перерывов, регламентированных с учетом продолжительности рабочей смены, видов и категории трудовой деятельности.

Виды трудовой деятельности на ПК разделяются на 3 группы: группа А – работа по считыванию информации с экрана с предварительным запросом; группа Б – работа по вводу информации; группа В – творческая работа в режиме диалога с ПК .

Если в течение рабочей смены пользователь выполняет разные виды работ, то его деятельность относят к той группе работ, на выполнение которой тратится не менее 50% времени рабочей смены.

Категории тяжести и напряженности работы на ПК определяются уровнем нагрузки за рабочую смену: для группы А – по суммарному числу считываемых знаков; для группы Б – по суммарному числу считываемых или вводимых знаков; для группы В – по суммарному времени непосредственной работы на ПК. В таблице приведены категории тяжести и напряженности работ в зависимости от уровня нагрузки за рабочую смену.

Количество и длительность регламентированных перерывов, их распределение в течение рабочей смены устанавливается в зависимости от категории работ на ПК и продолжительности рабочей смены.

При 8-часовой рабочей смене и работе на ПК регламентированные перерывы следует устанавливать:

- для первой категории работ через 2 часа от начала смены и через 2 часа после обеденного перерыва продолжительностью 15 минут каждый;

- для второй категории работ – через 2 часа от начала рабочей смены и через 1,5–2,0 часа после обеденного перерыва продолжительностью 15 минут каждый или продолжительностью 10 минут через каждый час работы;

- для третьей категории работ – через 1,5– 2,0 часа от начала рабочей смены и через 1,5–2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

При 12-часовой рабочей смене регламентированные перерывы должны устанавливаться в первые 8 часов работы аналогично перерывам при 8-часовой рабочей смене, а в течение последних 4 часов работы, независимо от категории и вида работ, каждый час продолжительностью 15 минут.

Продолжительность непрерывной работы на ПК без регламентированного перерыва не должна превышать 2 часа.

При работе на ПК в ночную смену продолжительность регламентированных перерывов увеличивается на 60 минут независимо от категории и вида трудовой деятельности.

Эффективными являются нерегламентированные перерывы (микропаузы) длительностью 1–3 минуты.

Регламентированные перерывы и микропаузы целесообразно использовать для выполнения комплекса упражнений и гимнастики для глаз, пальцев рук, а также массажа. Комплексы упражнений целесообразно менять через 2–3 недели.

Пользователям ПК, выполняющим работу с высоким уровнем напряженности, показана психологическая разгрузка во время регламентированных перерывов и в конце рабочего дня в специально оборудованных помещениях (комнатах психологической разгрузки).

Медико-профилактические и оздоровительные мероприятия. Все профессиональные пользователи ПК должны проходить обязательные предварительные медицинские осмотры при поступлении на работу, периодические медицинские осмотры с обязательным участием терапевта, невропатолога и окулиста, а также проведением общего анализа крови и ЭКГ.

Не допускаются к работе на ПК женщины со времени установления беременности и в период кормления грудью.

Близорукость, дальнозоркость и другие нарушения рефракции должны быть полностью скорректированы очками. Для работы должны использоваться очки, подобранные с учетом рабочего расстояния от глаз до экрана дисплея. При более серьезных нарушениях состояния зрения вопрос о возможности работы на ПК решается врачом-офтальмологом.

Интенсивно работающим целесообразно использовать такие новейшие средства профилактики зрения, как очки ЛПО–тренер и офтальмологические тренажеры ДАК и «Снайпер–ультра».

Досуг рекомендуется использовать для пассивного и активного отдыха (занятия на тренажерах, плавание, езда на велосипеде, бег, игра в теннис, футбол, лыжи, аэробика, прогулки по парку, лесу, экскурсии, прослушивание музыки и т.п.). Дважды в год (весной и поздней осенью) рекомендуется проводить курс витаминотерапии в течение месяца. Следует отказаться от курения. Категорически должно быть запрещено курение на рабочих местах и в помещениях с ПК.

3.3 Расчет системы искусственного освещения помещения

Помещение зала имеет естественное освещение через 2 больших боковых окон, и искусственное освещение, которое позволяет вести работы в темное время суток и днем в местах, где показатель КЕО не соответствует нормативам.

Поэтому рассчитаем общее освещение помещения длиной $A = 9$ м., шириной $B = 8,5$ м., высотой $H = 3$ м. Так как ориентация окон в предложенном плане помещения (Рисунок 3.1) южная то стены будут окрашены в белый цвет, пол будет оранжево–красный. Так как работа будет связана с компьютерами то коэффициент отражения будет составлять:

- 1 для потолка: 62%;
- 2 для стен: 43%;
- 3 для пола: 28%;
- 4 Для других поверхностей и рабочей мебели: 36% [14].

Разряд зрительной работы – III высокой точности. Нормируемая освещенность – 400 лк. [15]. Для помещения используем люминесцентную лампу ЛБ (белого цвета), мощностью 40 Вт., световым потоком 3300 лм., диаметром 40 мм. и длиной со штырьками 1210,6 мм. [15].

Высота светильника $h_c = 3 - r$, где r – высота лампочки

$$h_c = 3 - 2,5 = 0,5 \text{ м}$$

Высота рабочей поверхности $h_p = 1,2$ м.

Определим необходимое расстояние между светильниками [14]

$$L = \lambda \cdot h \tag{3.1}$$

где $\lambda = 1,2 \div 1,4$ [15].

Высота светильника над освещаемой поверхностью

$$h = H - h_p - h_c \tag{3.2}$$

$$h = 3 - 1,2 - 0,5 = 1,3 \text{ м}$$

По этим данным находим, что необходимое расстояние между светильниками равно

$$L = 1,2 \cdot 1,3 = 1,57 \text{ м}$$

Определим индекс помещения I [15]

$$I = \frac{A \cdot B}{h \cdot (A + B)} \quad (3.3)$$

$$I = \frac{9 \cdot 8,5}{2,5 \cdot (9 + 8,5)} = 1,74$$

Определим коэффициент использования η [15]

$$\eta = 0,54$$

В качестве светильника возьмем ЛСП06 рассчитанный на две лампы мощностью 40 Вт, диаметром 40 мм и длиной со штырьками 1210,6 мм. Длина светильника 1234 мм, ширина 276 мм. Световой поток лампы ЛБ 40 Фл составляет 3300 лм., световой поток, излучаемый светильником $\Phi_{св}$ равен

$$\Phi_{св} = \Phi_{л} \cdot 2 \quad (3.4)$$

$$\Phi_{св} = 3300 \cdot 2 = 6600 \text{ лм}$$

Определим число светильников

$$N = \frac{E \cdot K_3 \cdot S \cdot Z}{n \cdot \Phi_{л} \cdot \eta} \quad (3.5)$$

где S – площадь помещения, $S=76,5 \text{ м}^2$;

K_3 – коэффициент запаса, $K_3=1,5$ [15];

E – заданная минимальная освещенность, $E=400 \text{ лк}$. [15];

Z – коэффициент неравномерности освещения, $Z=1,2$ [15];

n – количество ламп в светильнике, $n=2$;

$\Phi_{л}$ – световой поток выбранной лампы, $\Phi_{л}=3300 \text{ лм}$.;

η – коэффициент использования, $\eta=0,54$ [15].

$$N = \frac{400 \cdot 1,5 \cdot 76,5 \cdot 1,2}{2 \cdot 3300 \cdot 0,54} = 15,45 \approx 15 \text{ светильников}$$

Расположение светильников показано на рисунке 3.4.

Итого, для создания нормированной освещенности нам понадобится 15 ламп в 15-ти светильниках располагающихся в пять рядов, в каждом светильнике по одной лампе.

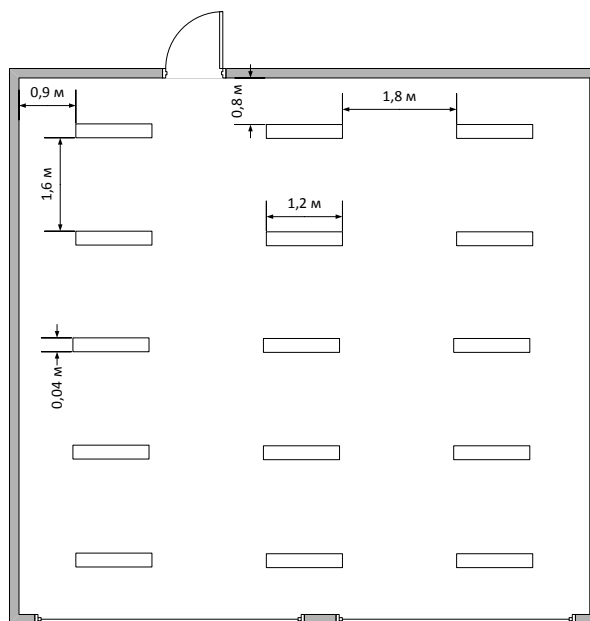


Рисунок 3.4 – Расположение светильников в помещении

Выводы

В данном разделе был произведён анализ условий труда в рабочем помещении для двух сотрудников. Уровень условий труда признан допустимым, и данные, полученные из расчетов полностью удовлетворяют требованиям стандартов безопасности жизнедеятельности.

Так как в помещении есть два окна, которые в дневное время суток обеспечивают достаточным количеством света для работы, необходимо было рассчитать освещенность в основном для позднего времени суток. Поэтому для создания нормированной освещенности вечером(ночью) понадобится 15 ламп мощностью 40 Вт., световым потоком 3300 лм., диаметром 40 мм. и длиной со штырьками 1210,6 мм. в 15-ти светильниках, располагающихся в пять рядов, в каждом светильнике по одной лампе.

4 Бизнес план

4.1 Описание работы и обоснование необходимости

Тема дипломной работы – информационная система “Моделирование торгов рынка электроэнергии”.

Цель проекта – разработка для учебных целей информационной системы “Моделирование торгов рынка электроэнергии”. Разработка включает в себя организацию торгов электроэнергии, автоматизированные рабочие места для участников рынка.

В данной работе будут решены такие задачи как:

- 1 Будет рассчитана смета затрат и цена программного обеспечения.
- 2 Определены объем и трудоемкость разработки программного обеспечения.
- 3 Рассчитаны затраты на разработку информационной технологии.
- 4 Будет рассчитана цена программного продукта.
- 5 Будет произведен расчет эксплуатационных затрат при использовании информационных технологий.

В рыночных условиях программное обеспечение выступает преимущественно в виде продукции научно–технических организаций, представляющей собой функционально завершенные и имеющие товарный вид программные средства, поставляемые заказчиком и продаваемые покупателям по рыночным ценам. Все завершенные разработки ПО являются научно–технической продукцией.

Стоимостная оценка ПО выражается и включает следующие статьи расходов [16]:

- заработную плату исполнителей, основную (Z_o) и дополнительную (Z_d);
- отчисления на социальный налог ($Z_{сз}$);
- материалы и комплектующие (M);
- спецоборудование (P_c);
- машинное время (P_m);
- расходы на научные командировки ($P_{нк}$);
- прочие расходы (P_3);
- накладные расходы (P_n).

В данном разделе приводится рассмотрение экономической составляющей реализации данного дипломного проекта, отражающей временные, трудовые и финансовые затраты на разработку проекта [16].

4.2 Расчет затрат на разработку информационных технологий

Под информационными технологиями понимаются экономические информационные системы (ЭИС), программные продукты (ПП), информационные базы данных и т.д.

Расчет полных затрат на разработку проектного решения в виде информационных технологий ($C_{\text{пi}}$) осуществляется по формуле [17]

$$C_{\text{пi}} = Z_{\text{фот}} + Z_{\text{сзи}} + M_i + P_{\text{ми}} + P_{\text{нки}} + П_{\text{зи}} + P_{\text{ни}} \quad (4.1)$$

где $Z_{\text{фот}}$ – общий фонд оплаты труда разработчиков, тенге;

$Z_{\text{сзи}}$ – отчисления по социальному налогу, тенге;

M_i – затраты на материалы, тенге;

$P_{\text{ми}}$ – затраты, связанные с эксплуатацией техники, тенге;

$P_{\text{нки}}$ – затраты на научные командировки, тенге;

$П_{\text{зи}}$ – прочие затраты, тенге;

$P_{\text{ни}}$ – накладные расходы, тенге.

Размер фонда оплаты труда разработчиков ($Z_{\text{фот}}$) рассчитывается по формуле

$$Z_{\text{фот}} = Z_{\text{oi}} + Z_{\text{ди}} \quad (4.2)$$

где Z_{oi} – основная заработная плата, тенге;

$Z_{\text{ди}}$ – дополнительная заработная плата, тенге.

Затраты на оплату труда зависят от объема и трудоемкости разработки программного обеспечения.

Общий объем (V_0) программного продукта определяется исходя из количества и объема функции, реализуемых программой

$$V_0 = \sum_{j=1}^n V_j \quad (4.3)$$

где V_j – объем отдельной функции ПО;

n – общее число функций.

Исходной информацией необходимой для определения общего объема программного продукта являются «Укрупнённые нормы времени на разработку ПО в зависимости от уточнённого объема ПО и группы сложности» (см приложение В) и составляет

$$V_0 = 16\,440$$

Общая трудоемкость небольших проектов рассчитывается по формуле

$$T_o = T_n * K_c * K_m * K_n \quad (4.4)$$

где K_c – коэффициент, учитывающий сложность ПО;

K_T – поправочный коэффициент, учитывающий степень использования при разработке стандартных модулей;

K_n – коэффициент, учитывающий степень новизны ПО;

T_n – нормативная трудоемкость.

Коэффициент сложности определяется на основе данных представленных в таблице 4.1 и составляет $K_c=0,12$, т.к. в ПП присутствуют 2 характеристики – работа в реальном времени и управление удаленными объектами.

Т а б л и ц а 4.1 – Дополнительные коэффициенты сложности ПО

| Характеристика ПО | Значения K_c |
|--|----------------|
| 1. Функционирование ПО в расширенной операционной среде (связь с другими ПО) | 0,08 |
| 2. Интерактивный доступ | 0,06 |
| 3. Обеспечение хранения, ведения и поиска данных в сложных структурах | 0,07 |
| 4. Наличие у ПО одновременно нескольких характеристик по табл.Г4.1, приложение Г | |
| 4.1 2 характеристики | 0,12 |
| 4.2 3 характеристики | 0,18 |
| 4.3 Свыше 3-х характеристик | 0,26 |

Поправочный коэффициент, учитывающий степень использования при разработке ПО стандартных модулей (K_m), определяется удельным весом этих модулей в общем объеме проектируемого продукта (от 40% до 60%) и составляет 0,7.

Поправочный коэффициент, учитывающий новизну разрабатываемого ПО (K_n) определяется на основе данных представленных в таблице 4.2 и составляет 1,0.

Т а б л и ц а 4.2 – Поправочные коэффициенты, учитывающие новизну ПО(K_n)

| Категория новизны | Степень новизны | Использование | | Значение K_n |
|-------------------|---|---------------------|------------------|----------------|
| | | На основе нового ПК | В среде новой ОС | |
| А | Принципиально новые ПО, не имеющие доступных аналогов | + | + | 1,75 |
| | | – | + | 1,6 |
| | | + | – | 1,2 |
| | | – | – | 1,0 |
| Б | ПО, являющиеся развитием определенного параметрического ряда ПО | + | + | 1,0 |
| | | – | – | 0,9 |
| | | + | – | 0,8 |

| Категория новизны | Степень новизны | Использование | | Значение K_n |
|-------------------|---|---------------------|------------------|----------------|
| | | На основе нового ПК | В среде новой ОС | |
| В | ПО, являющиеся развитием определенного параметрического ряда ПО, разработанных для ранее освоенных типов конфигурации ПК и ОС | – | – | 0,7 |

Базой для определения нормативной трудоемкости являются, укрупненные нормы времени на разработку ПО в зависимости от уточненного объема ПО и группы сложности (см. Приложение В).

Нормативная трудоемкость ПО (T_n) определяется на основе принятого в расчет объема ПП и категории сложности, которая уточняется с учетом сложности и новизны проекта и степени использования стандартных модулей при разработке.

Учитывая данные из Приложения В: для 1–ой категории сложности ПО

$$T_n = 487$$

Следовательно T_o будет равно

$$T_o = 487 * 0,12 * 0,7 * 1,0 = 40,908$$

Численность исполнителей и срок разработки ПО. На основе трудоемкости определяются плановое число разработчиков ($Ч_p$) и плановые сроки, необходимые для реализации проекта в целом (T_p). При этом могут решаться следующие задачи:

- расчет числа исполнителей при заданных сроках разработки проекта;
- определение сроков разработки проекта при заданной численности исполнителей.

Численность исполнителей проекта ($Ч_p$) рассчитывается по формуле

$$Ч = T_o / (T_p \times \Phi_{эф}) \quad (4.5)$$

где $\Phi_{эф}$ – эффективный фонд времени работы одного работника в течение года (дн.);

T_o – общая трудоемкость разработки проекта (чел./дн.);

T_p – срок разработки проекта (лет).

Срок разработки проекта (T_p) определяется по формуле

$$T_p = T_o / (C_p \times \Phi_{\text{эф}}) \quad (4.6)$$

где C_p – плановое число разработчиков.

Эффективный фонд времени работы одного работника ($\Phi_{\text{эф}}$) рассчитывается по формуле

$$\Phi_{\text{эф}} = D_r - D_n - D_v - D_o \quad (4.7)$$

где D_r – количество дней в году;

D_n – количество праздничных дней в году;

D_v – количество выходных дней в году;

D_o – количество дней отпуска.

В соответствии производственным календарем на 2014 год

$$D_r = 365,$$

$$D_n = 14,$$

$$D_v = 103,$$

$$D_o = 10$$

то по формуле (4.7) получим

$$\Phi_{\text{эф}} = 365 - 14 - 106 - 10 = 235 \text{ дней}$$

Плановое число разработчиков $C_p = 1$, следовательно по формуле (4.6)

$$T_p = 40,908 / (1 * 235) = 0,17 \text{ лет}$$

Таким образом, согласно произведенным расчетам и в соответствие с формулой (4.4)

$$Ч = 40,908 / (0,17 * 235) = 1 \text{ чел}$$

Основная заработная плата исполнителей на конкретное ПО рассчитывается по формуле

$$Z_{oi} = \sum_{i=1}^n T_{чи} \times T_{ч} \times \Phi_{п} \times K \quad (4.8)$$

где n – количество исполнителей, занятых разработкой конкретного ПО;

$T_{чи}$ – часовая тарифная ставка i -го исполнителя (тыс.тенге);

Φ_{Π} – плановый фонд рабочего времени i -го исполнителя (дней), 21 раб день в месяц;

$T_{\text{ч}}$ – количество часов работы в день (час), 8 часов;

K – коэффициент премирования, составляет 1,38.

По данным о специфике и сложности выполняемых функций составляется штатное расписание группы специалистов–исполнителей, участвующих в разработке ПО, с определением образования, специальности, квалификации и должности (Таблица 4.3).

Т а б л и ц а 4.3 – Сведения по работникам, задействованным в проекте

| Специалист – Исполнитель | Количество, человек | Заработная плата в месяц, тенге |
|--------------------------|---------------------|---------------------------------|
| Программист | 1 | 185 000 |
| Итого | | 185 000 |

Часовая тарифная ставка рассчитывается путем деления месячной тарифной ставки на установленную при 40–часовой недельной норме рабочего времени расчетную среднемесячную норму рабочего времени в часах (Φ_p)

$$T_{\text{ч}} = \frac{T_{\text{М}}}{\Phi_p} \quad (4.9)$$

где $T_{\text{ч}}$ – часовая тарифная ставка (тыс.тенге);

$T_{\text{М}}$ – месячная тарифная ставка (тыс.тенге).

Таким образом

$$T_{\text{ч}} = \frac{185\,000}{168} = 1101,2$$

По формуле (4.8) можно рассчитать основную заработную плата исполнителей

$$З_{oi} = 1101,2 * 8 * 40 * 1,38 = 486\,290 \text{ тенге}$$

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

$$З_{di} = З_{oi} * N_d / 100 \quad (4.10)$$

где N_d – коэффициент дополнительной заработной платы разработчиков 23%

$$З_{di} = 486\,290 * 0,23 = 111\,847 \text{ тенге}$$

Таким образом, затраты на оплату труда составят

$$З_{\text{ФОТ}} = 486\,290 + 111\,847 = 598\,137 \text{ тенге}$$

Социальный налог составляет 11% (ст. 358 п. 1 НК РК) от дохода работника, и рассчитывается по формуле

$$З_{\text{сзи}} = (\text{ФОТ} - \text{ПО}) \times 11\% \quad (4.11)$$

где ПО – пенсионные отчисления, которые составляют 10% от ФОТ и социальным налогом не облагаются

$$\text{ПО} = \text{ФОТ} \times 10\% \quad (4.12)$$

Таким образом

$$\text{ПО} = 598\,137 \times 0,1 = 59\,814 \text{ тенге,}$$

$$З_{\text{сзи}} = (598\,137 - 59\,814) \times 0,11 = 59\,216 \text{ тенге}$$

Затрат на материалы определяются по формуле

$$M_i = (З_{\text{осн.}} \times N_{\text{мз}}) / 100\% \quad (4.13)$$

где $N_{\text{мз}}$ – норма расхода материалов от основной заработной платы (3–5%).

$$M_i = 486\,290 \times 0,04 = 19\,452 \text{ тенге}$$

Расходы по статье «Машинное время» ($P_{\text{ми}}$) включают оплату машинного времени, необходимого для разработки и отладки ПО, которое определяется по нормативам (в машино–часах) на 100 строк исходного кода ($N_{\text{мв}}$) машинного времени в зависимости от характера решаемых задач и типа ПК

$$P_{\text{ми}} = C_{\text{ми}} \times (V_{\text{oi}}/100) \times N_{\text{мв}} \quad (4.14)$$

где $C_{\text{ми}}$ – цена одного машино–часа (тыс.тенге);

V_{oi} – общий объем ПО (строк исходного кода);

$N_{\text{мв}}$ – норматив расхода машинного времени на отладку 100 строк исходного кода (машино–часов).

Учитывая данные из Приложения Д: наименование подсистемы АС и ДОС – 1 и средний расход машинного времени, ч/100 сток кода составляет 12.

Таким образом

$$P_{Mi} = 1101,2 * (16\ 440 / 100) * 12 = 2\ 172\ 447 \text{ тенге}$$

Расходы по статье «Научные командировки» ($P_{нki}$) на конкретное ПО определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате

$$P_{нki} = Z_{oi} \times H_{рнк}/100 \quad (4.15)$$

где $H_{рнк}$ – норматив расходов на командировки в целом по организации в (%), в дипломной работе нужно брать 30%.

Подставляем все данные в формулу (4.15) получаем

$$P_{нki} = 486\ 290 * 0,3 = 145\ 887 \text{ тенге}$$

Расходы по статье «Прочие затраты» (P_{zi}) на конкретное ПО включают затраты на приобретение и подготовку специальной научно–технической информации и специальной литературы. Определяются по нормативу, разрабатываемому в целом по организации, в процентах к основной заработной плате

$$P_{zi} = Z_{oi} \times H_{пз}/100 \quad (4.16)$$

где $H_{пз}$ – норматив прочих затрат в целом по организации в (%), в дипломной работе нужно брать 20% .

Подставляем все данные в формулу (16) получаем

$$P_{zi} = 486\ 290 * 0,2 = 97\ 258 \text{ тенге}$$

Затраты по статье «Накладные расходы» (P_{ni}), связанные с необходимостью содержания аппарата управления, вспомогательных хозяйств и опытных(экспериментальных) производств, а также с расходами на общехозяйственные нужды (P_{ni}), относятся на конкретное ПО по нормативу ($H_{рн}$) в процентном отношении к основной заработной плате исполнителей. Норматив устанавливается в целом по организации

$$P_{ni} = Z_{oi} \times H_{рн}/100\% \quad (4.17)$$

где P_{ni} – накладные расходы на конкретную ПО (тыс.тенге);

$H_{рн}$ – норматив накладных расходов в целом по организации в (%), в дипломной работе нужно брать 70% .

Подставляем все данные в формулу (4.17) получаем

$$P_{ni} = 486\ 290 * 0,7 = 340\ 403 \text{ тенге}$$

Таким образом, общая сумма затрат на разработку проектного решения составят

$$C_{\text{пi}} = 598\,137 + 59\,216 + 19\,452 + 2\,172\,447 + 145\,887 + 97\,258 + 340\,403,$$

$$C_{\text{пi}} = 3\,432\,800 \text{ тенге}$$

Результаты выполненных расчетов представлены в виде таблицы 4.4 и на рисунке 4.1.

Т а б л и ц а 4.4 – Себестоимость разработки Информационной системы модели торгов рынка электроэнергии .

| Затраты на разработку | Условное обозначение | Значение, тенге | В процентах от общей суммы |
|-----------------------|----------------------|-----------------|----------------------------|
| Фонд оплаты труда | $Z_{\text{ФОТ}}$ | 598 137 | 17,4 |
| Социальный налог | $Z_{\text{сзи}}$ | 59 216 | 1,7 |
| Материалы | M_i | 19 452 | 0,6 |
| Машинное время | $P_{\text{ми}}$ | 2 172 447 | 63,2 |
| Научные командировки | $P_{\text{нки}}$ | 145 887 | 4,2 |
| Прочие затраты | $P_{\text{зи}}$ | 97 258 | 2,8 |
| Накладные расходы | $P_{\text{ни}}$ | 340 403 | 9,9 |
| Итого: | | 3 432 800 | 100 |

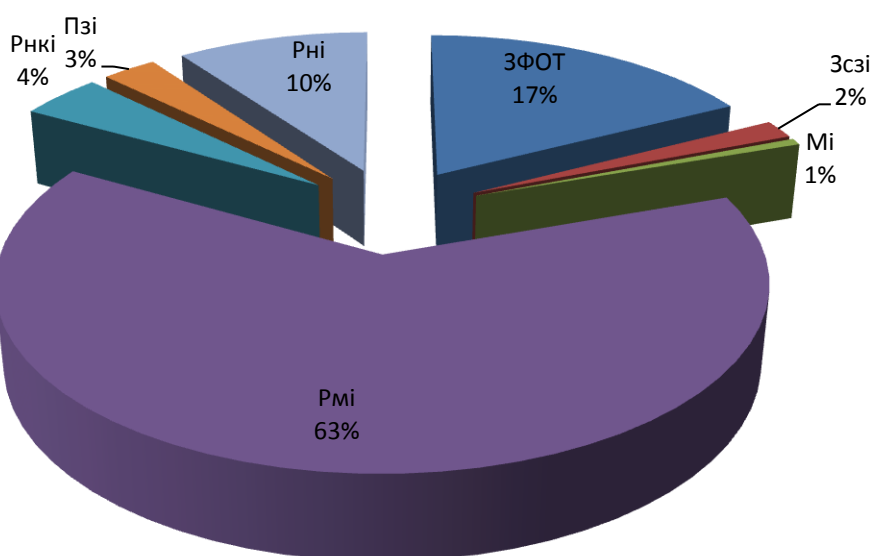


Рисунок 4.1 – Структура затрат на разработку ПП

4.3 Расчет цены программного продукта

Расчет цены ПП в организационно–экономической части дипломной работы предлагается производить следующим образом [18]:

– если ПП разработан одной организацией по заказу другой и не предназначен для тиражирования, то затраты на разработку ПП считаются его себестоимостью, и при формировании цены применяется затратный метод;

– если ПП предназначен для тиражирования, то конечная цена определяется путем экспертных оценок на основании ценностного подхода с учетом текущих цен конкурентов (если существуют аналогичные ПП).

Расчет цены ПП, который разработан одной организацией по заказу другой и не предназначен для тиражирования, осуществляется по формуле

$$C_{ПП} = Z_{РПП} + П_n + НДС \quad (4.18)$$

где $C_{ПП}$ – цена программного продукта, тенге;

$Z_{РПП}$ – затраты на разработку проектного решения, в данном случае программного продукта, тенге;

$П_n$ – планируемая прибыль, тенге;

$НДС$ – налог на добавленную стоимость, тенге.

Планируемая прибыль рассчитывается по формуле

$$П_n = Z_{РПП} + R_{НПП} \quad (4.19)$$

где $R_{НПП}$ – нормативная рентабельность ПП, определяемая организацией.

$НДС$, начисленный на ПП, определяется следующим образом

$$НДС = (Z_{РПП} + П_n) * k_{НДС} \quad (4.20)$$

где $k_{НДС}$ – ставка налога на добавленную стоимость.

Подставляем все значения в формулы (4.18) – (4.20) и получаем:

По формуле (4.19) учитывая, что $Z_{РПП} = C_{пi}$, $R_{НПП}$ – это процент рентабельности по отношению к себестоимости составляет 20%

$$R_{НПП} = C_{пi} * 0,20 = 3\,432\,800 * 0,20 = 686\,560 \text{ тенге,}$$

$$П_n = 3\,432\,800 + 686\,560 = 4\,119\,360 \text{ тенге}$$

Подставив данные в формуле (4.20) получаем

$$НДС = (3\,432\,800 + 4\,119\,360) * 0,12 = 988\,646 \text{ тенге}$$

Подставив данные в формуле (4.18) получаем

$$C_{ПП} = 3\,432\,800 + 4\,119\,360 + 988\,646 = 8\,540\,806 \text{ тенге}$$

Выводы

Стоимость разработки Информационной системы торгов рынка электроэнергии составила 3 432 800 тенге.

Наибольшую долю в общей сумме затрат на разработку ПП составляют затраты на машинное время 2 172 447 тенге, которые являются определяющими при расчете затрат на оплату труда.

Таким образом, общая сумма затрат на оплату труда составила 2 658 737 тенге или 77% от общей суммы понесенных затрат.

Цена реализации ПП составила 9 227 366 тенге, в долларах \$49 599 (курс доллара к тенге на май 2014 года, составляет \$1 = 182,04 тенге).

Заключение

Цель выполнения дипломного проекта – разработка информационной системы модели торгов рынка электроэнергии, выполнена. Внедрение данного проекта позволит в значительной степени повысить качество и производительность выполнения лабораторных работ.

В ходе написания были решены следующие поставленные задачи:

- произведены теоретические исследования по современному рынку электроэнергии;
- выявлены способы разработки ИС;
- исследованы мероприятия по охране труда, технике безопасности;
- проанализирована экономическая целесообразность выполнения продукта;
- разработана ис модели торгов рынка электроэнергии.

В ходе исследования был изучен современный рынок торгов электроэнергией и методы проведения торгов.

Был произведён анализ условий труда в рабочем помещении для двух сотрудников. Уровень условий труда признан допустимым, и данные, полученные из расчетов полностью удовлетворяют требованиям стандартов безопасности жизнедеятельности.

Была рассчитана цена реализации ПП, которая составила составила 9 227 366 тенге, в долларах \$49 599 (курс доллара к тенге на май 2014 года, составляет \$1 = 182,04 тенге). Наибольшую долю в разработке составила сумма затрат на оплату труда

Разработанная ИС способна работать в 3х режимах: балансирующий рынок, краткосрочный рынок, долгосрочный рынок. Участники рынка в реальном времени могут подавать заявки и участвовать в торгах.

Разработанная ИС выполнена качественно и находится на стадии внедрения и в скором времени будет использоваться в учебном заведении. В процессе эксплуатации ИС может быть доработана с учётом требований заказчика.

Список литературы

- 1 Закон «Об электроэнергетике»
- 2 Сайт <http://www.investkz.com/journals/43/113.html>
- 3 Методические указания по проведению централизованных торгов электрической энергией на среднесрочный (неделя, месяц) и долгосрочный (квартал, год) периоды» АО «КОРЭМ».–Астана, 2013.
- 4 Методические указания по проведению централизованных спот–торгов электрической энергией в режиме «за день вперед» АО «КОРЭМ».–Астана, 2013.
- 5 Сайт <http://www.oracle.com/us/products/database/overview/index.html>
- 6 Сайт <http://www.embarcadero.com/products/delphi>
- 7 Сайт <http://www.allroundautomations.com/doa.html?gclid=CLa4-u-4yb4CFckBwwodcR0ArQ>
- 8 Сайт http://docs.oracle.com/cd/B28359_01/server.111/b28318/datatype.htm
- 9 Дубовцев В.А. Безопасность жизнедеятельности: Учебное пособие для дипломников. – Киров: изд. КирПИ, 1992.
- 10 Безопасность жизнедеятельности. /Под ред. Н.А. Белова. – М.: Знание, 2000. – 364с.
- 11 Мотузко Ф.Я. Охрана труда. – М.: Высшая школа, 1989. – 336с.
- 12 Согласно ГОСТ 12.1.005–88. ССБТ. Оптимальные и допустимые нормы микроклимата, в зависимости от категории работ.
- 13 Зинченко В.П. Основы эргономики. – М.: МГУ, 1979. – 179с.
- 14 Самгин Э.Б. Освещение рабочих мест. – М.: МИРЭА, 1989. – 186с.
- 15 Справочная книга для проектирования электрического освещения. / Под ред. Г.Б. Кнорринга. – Л.: Энергия, 1976.
- 16 Еркешева З.Д, Боканова Г.Ш. – Методические указания к выполнению экономической части дипломных работ для студентов специальности 5В070400 – Вычислительная техника и программное обеспечение. – Алматы: АУЭС, 2014.
- 17 Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат. – М.: Издательский дом «Вильямс», 2004. – 1136 с.
- 18 Липаев В. В. Техничко–экономическое обоснование проектов сложных программных систем. – М.: СИНТЕГ, 2004. – 284 с.

Приложение А

Классификация типов программного обеспечения

Т а б л и ц а А.1 – Классификация типов программного обеспечения

| Код типа | Наименование типа ПО | Состав и содержание типа ПО |
|----------|---|---|
| 1.0 | ПО общего назначения | 1.1 ПО СУБД. 1.2 ПО систем ведения линейных файлов. 1.3 ПО ведения баз данных и линейных файлов. 1.4 ПО информационно–поисковых и информационно–справочных систем. 1.5 ПО ввода информации. 1.6 ПО мониторов телеобработки и сетей ПЭВМ. 1.7 ПО окружения СУБД, расширяющие возможности существующих СУБД. 1.8 ПО, расширяющие возможности обработки. |
| 2.0 | ПО технологии автоматизации программирования и проектирования АСУ | 2.1 ПО автоматизации проектирования для автоматизации проектирования различных АСУ. 2.2 ПО технологии программирования. 2.3 ПО автоматизации программирования (для автоматизации процессов обработки и вывода информации). 2.4 ПО, расширяющие существующие языки программирования для повышения их компактности и простоты пользования. 2.5 ПО общего назначения, функционально–ориентированные. 2.6 ПО автоматического программирования. Реализуют различные классы экономико–математических методов и являются системами общего назначения, которые применяются в различных АСУ, для научно–технических расчетов и исследований. |
| 3.0 | ПО методоориентированных расчетов | 3.1 ПО оптимизационных расчетов (обеспечивают решение различного класса задач оптимального планирования и управление производством). 3.2 ПО статистического анализа и прогнозирования (для прогнозирования ТЭП, спроса и т.д.). 3.3 ПО сетевого планирования и управления. 3.4 ПО общей математики. 3.5 ПО имитационного моделирования. |

Продолжение приложения А

Окончание таблицы А.1

| Код типа | Наименование типа ПО | Состав и содержание типа ПО |
|----------|---|--|
| 4.0 | ПО организации вычислительного процесса | <p>Автоматизация процесса ведения наборов данных, при обеспечении их надежного и систематизированного хранения.</p> <p>Повышение производительности ПЭВМ и пользователей ПО.</p> <p>Формирование и выдача отчетов о работе ПЭВМ.</p> <p>Оперативный контроль системы и ресурсов.</p> <p>Для автоматизации обработки экономических данных, при этом выделяются ПО, несущие функциональную нагрузку в АСУ. ПО данного типа выполнены в основном автономно.</p> |
| 5.0 | ПО функционального назначения | <p>5.1 ПО системы ПОП и СУП.</p> <p>5.2 ПО оперативного управления основным производством.</p> <p>5.3 ПО управления технической подготовкой производства.</p> <p>5.4 ПО бухгалтерского учета и управления финансами.</p> <p>5.5 ПО управления кадрами.</p> <p>5.6 ПО, не вошедшие ни в один из перечисленных типов ПО.</p> |

Приложение Б

Каталог функций программного обеспечения

Т а б л и ц а Б.1 – Каталог функций программного обеспечения

| № п/п | Наименование (содержание) функций | Объем функций (строк исходного кода) с использованием среды разработки приложений | | |
|--|---|---|-----------------------------|---------------------------|
| | | Delphi (Borland) | C++ Builder (Borland) | Visual C++ (Microsoft) |
| | | 1. Ввод, анализ входной информации, генерация кодов и процессор входного языка | | |
| 101 | Организация ввода информации | 100 | 110 | 150 |
| 102 | Контроль, предварительная обработка и ввод информации | 550 | | |
| 103 | Анализ входного языка (синтаксический и семантический) | 630 | 660 | 980 |
| 104 | Преобразование операторов входного языка и команды другого языка | 1050 | 1050 | 980 |
| 105 | Обработка входного заказа и формирование таблиц | 750 | 900 | 1340 |
| 106 | Преобразование входного языка в машинные команды (транслятор, препроцессор, макрогенератор) | 1300 | 4300 | 5100 |
| 107 | Синтаксический и семантический анализ входного языка и генерация кодов команд | 8700 | 5400 | 5400 |
| 108 | Процессор языка | 3000 | 2300 | 2300 |
| 109 | Организация ввода/вывода информации в интерактивном режиме | 220 | 220 | 320 |
| 110 | Организация ввода/вывода информации с сети терминалов | 3680 | 3340 | 3200 |
| 110 | Управление вводом/выводом | 2700 | 2900 | 2400 |
| 2. Формирование, введение и обслуживание баз данных | | | | |
| 201 | Генерация структуры базы данных | 3450 | 3950 | 4300 |
| 202 | Генерация структуры базы данных | 1540 | 1610 | 2060 |
| 203 | Формирование баз данных | 1700 | 1750 | 2180 |
| 204 | Обработка наборов и записей базы данных | 2050 | 2350 | 2670 |
| 205 | Обслуживание базы данных в пакетном режиме | 1030 | 1100 | 1260 |

Продолжение приложения Б

Продолжение таблицы Б.1

| № п/п | Наименование (содержание) функций | Объем функций (строк исходного кода) С использованием среды разработки приложений | | |
|--|---|--|--------------------------|--------------------------------------|
| | | Delphi (Borland) | C++ Builder (Borland) | Visual C++ (Micros oft) |
| 206 | Обслуживание базы данных в интер-активном режиме | 3800 | 4400 | 6950 |
| 208 | Организация поиска и поиск в базе данных | 5230 | 5460 | 5480 |
| 209 | Реорганизация базы данных | 130 | 190 | 220 |
| 210 | Загрузки базы данных | 3150 | 2950 | 2780 |
| 3. Формирование и обработка файлов | | | | |
| 301 | Формирование последовательного файла | 340 | 360 | 290 |
| 302 | Сортировка файла | 340 | 360 | 290 |
| 303 | Автоматическая сортировка файлов | 1040 | 1150 | 930 |
| 304 | Обслуживание файлов | 520 | 540 | 420 |
| 305 | Обработка файлов | 750 | 800 | 720 |
| 306 | Обработка файлов в диалоговом режиме | 2400 | 2600 | 3050 |
| 307 | Совместная обработка группы файлов | 4900 | 5300 | 6180 |
| 308 | Управление файлами | 5130 | 5380 | 5750 |
| 309 | Формирование файла | 1100 | 1080 | 1020 |
| 4. Генерация программ и ПО, а также настройка ПО | | | | |
| 401 | Генерация рабочих программ | 3680 | 4120 | 3360 |
| 402 | Генерация программ по описанию пользователей | 10870 | 12330 | 9880 |
| 403 | Формирование служебных таблиц | 570 | 620 | 1070 |
| 404 | Система генерации ПО | 5120 | 5340 | 4980 |
| 405 | Система генерации ПО | 250 | 300 | 370 |
| 5. Управление ПО, компонентами ПО и внешними устройствами | | | | |
| 501 | Монитор ПО (управление работой компонентов) | 350 | 360 | 740 |
| 502 | Монитор системы (управление работой комплекса ПО) | 3750 | 3880 | 7740 |
| 503 | Управление внешними устройствами и объектами | 6850 | 7340 | 5900 |
| 504 | Обработка прерываний | 890 | 730 | 540 |
| 505 | Управление внешней памятью | 250 | 210 | 200 |

Продолжение приложения Б

Окончание таблицы Б.1

| № п/п | Наименование (содержание) функций | Объем функций (строк исходного кода) С использованием среды разработки приложений | | |
|--|--|--|-----------------------------|----------------------------------|
| | | Delphi (Borland) | C++ Builder (Borland) | Visual C++ (Micros oft) |
| 506 | Обработка ошибочных и сбойных ситуаций | 430 | 410 | 410 |
| 507 | Обеспечение интерфейса между компонентами | 730 | 750 | 970 |
| 6. Отладка прикладных программ, обмен информацией между магнитным диском и магнитной лентой, вспомогательные программные функции | | | | |
| 601 | Отладка прикладных программ в интерактивном режиме | 4500 | 4700 | 4300 |
| 602 | Обмен информацией между магнитным диском и магнитной лентой | * | * | * |
| 603 | Копирование наборов данных на магнитной ленте и восстановление | * | * | * |
| 604 | Справка и обучение | 680 | 680 | 720 |
| 605 | Вспомогательные и сервисные программы | 460 | 490 | 580 |
| 7. Расчетные задачи, формирование и вывод на внешние носители (АЦПУ) документов сложной формы и файлов | | | | |
| 701 | Матстатистика и прогнозирование | 8370 | 9570 | 9320 |
| 702 | Расчетные задачи (расчет режимов обр-ки) | 12600 | 15300 | 14800 |
| 703 | Расчет показателей | 410 | 500 | 460 |
| 704 | Процессор отчетов | 1070 | 1230 | 3200 |
| 705 | Формирование и вывод на внеш.носители | 2650 | 2850 | 3500 |
| 706 | Предварит.обработка и печать файлов | 540 | 560 | 470 |
| 707 | Графический вывод результатов | 300 | | 480 |
| 708 | Интерактивный редактор текста | 3800 | 3910 | 4540 |
| 709 | Измерение состояния ресурсов в интерактивной системе | 650 | 440 | 480 |

Приложение В

Укрупненные нормы времени на разработку ПО в зависимости от уточненного объема ПО и группы сложности ПО

Т а б л и ц а В.1 – Укрупненные нормы времени на разработку ПО (T_n) в зависимости от уточненного объема ПО (V_v) и группы сложности ПО (чел./дн.)

| Объем ПО (строки исходного кода, LOC) | Категории сложности ПО | | | Категории сложности ПО |
|--|------------------------|-----|-----|------------------------------|
| | 1-я | 2-я | 3-я | |
| 200 | – | – | 21 | 1 |
| 300 | – | – | 23 | 2 |
| 400 | – | – | 25 | 3 |
| 500 | – | – | 27 | 4 |
| 600 | – | 33 | 28 | 5 |
| 700 | – | 36 | 30 | 6 |
| 800 | – | 38 | 32 | 7 |
| 900 | – | 40 | 34 | 8 |
| 1000 | 51 | 43 | 36 | 9 |
| 1200 | 54 | 45 | 38 | 10 |
| 1400 | 57 | 48 | 40 | 11 |
| 1600 | 60 | 50 | 42 | 12 |
| 1800 | 64 | 54 | 45 | 13 |
| 2000 | 68 | 57 | 48 | 14 |
| 2200 | 73 | 61 | 51 | 15 |
| 2400 | 76 | 64 | 54 | 16 |
| 2600 | 81 | 68 | 57 | 17 |
| 2800 | 86 | 72 | 60 | 18 |
| 3000 | 91 | 76 | 64 | 19 |
| 3200 | 97 | 81 | 68 | 20 |
| 3400 | 103 | 86 | 72 | 21 |
| 3600 | 110 | 92 | 77 | 22 |
| 3800 | 117 | 98 | 82 | 23 |
| 4000 | 124 | 104 | 87 | 24 |
| 4200 | 133 | 111 | 93 | 25 |
| 4400 | 141 | 118 | 99 | 26 |
| 4600 | 151 | 126 | 105 | 27 |
| 4800 | 160 | 134 | 112 | 28 |
| 5000 | 170 | 142 | 119 | 29 |
| 5500 | 182 | 152 | 127 | 30 |
| 6000 | 194 | 162 | 135 | 31 |
| 6500 | 206 | 172 | 144 | 32 |
| 7000 | 220 | 184 | 154 | 33 |
| 7500 | 235 | 196 | 164 | 34 |
| 8000 | 252 | 210 | 175 | 35 |
| 8500 | 268 | 224 | 187 | 36 |
| 9000 | 288 | 240 | 200 | 37 |

Продолжение приложения В

Окончание таблицы В.1

| Объем ПО (строки исходного кода, LOC) | Категории сложности ПО | | | Категории сложности ПО |
|--|------------------------|------|------|------------------------------|
| | 1-я | 2-я | 3-я | |
| 9500 | 307 | 256 | 214 | 38 |
| 10000 | 327 | 273 | 228 | 39 |
| 11000 | 349 | 291 | 243 | 40 |
| 12000 | 374 | 312 | 260 | 41 |
| 13000 | 399 | 333 | 278 | 42 |
| 14000 | 427 | 356 | 297 | 43 |
| 15000 | 456 | 380 | 317 | 44 |
| 16000 | 487 | 406 | 339 | 45 |
| 18000 | 520 | 434 | 362 | 46 |
| 20000 | 556 | 464 | 387 | 47 |
| 22000 | 595 | 496 | 414 | 48 |
| 24000 | 636 | 530 | 442 | 49 |
| 26000 | 679 | 566 | 472 | 50 |
| 28000 | 727 | 606 | 505 | 51 |
| 30000 | 775 | 646 | 540 | 52 |
| 32000 | 830 | 692 | 577 | 53 |
| 34000 | 888 | 740 | 617 | 54 |
| 36000 | 950 | 792 | 660 | 55 |
| 38000 | 1016 | 847 | 706 | 56 |
| 40000 | 1087 | 906 | 755 | 57 |
| 42000 | 1161 | 968 | 807 | 58 |
| 44000 | 1242 | 1035 | 863 | 59 |
| 46000 | 1328 | 1107 | 923 | 60 |
| 48000 | 1420 | 1184 | 987 | 61 |
| 50000 | 1620 | 1267 | 1056 | 62 |

Приложение Г

Характеристики категорий сложности ПО

Т а б л и ц а Г.1 – Характеристики категорий сложности ПО

| Категории сложности | Характеристики ПО |
|---------------------|---|
| 1 | ПО, обладающие одной или несколькими из следующих характеристик: 1) Наличие сложного интеллектуального языкового интерфейса с пользователем. 2) Режим работы в реальном времени. 3) Обеспечение телекоммуникационной обработки данных и управление удаленными объектами. 4) Машинная графика. 5) Многомашинные комплексы. 6) Обеспечение существенного распараллеливания вычислений |
| 2 | ПО, обладающие одной или несколькими из следующих характеристик: 1) Оптимизационные расчеты. 2) Обеспечение настройки ПО на изменения структур входных и выходных данных. 3) Настройка ПО на нестандартную конфигурацию технических средств. 4) Обеспечение переносимости ПО. 5) Реализация особо сложных инженерных и научных расчетов. |
| 3 | ПО, не обладающие перечисленными выше характеристиками. |

Приложение Д

Оценка значений среднего машинного времени на отладку 100 строк
исходного кода без применения ПО

Т а б л и ц а Д.1 – Оценка значений среднего машинного времени на
отладку 100 строк исходного кода без применения ПО

| Наименование подсистемы АС и СОД | Средний расход машинного времени, ч/100 строк кода |
|--|---|
| 1. Общесистемные задачи: ведение линейных файлов, информационно–поисковые системы и информационно–справочные системы, сбор информации, ввод информации, расширение возможностей средств обработки данных, организация вычислительного процесса | 12 |
| 2. Задачи расчетного характера | 15 |
| 3. Оперативное управление производством, расчеты по ценообразованию | 7 |
| 4. Техническая подготовка производства, транспортное, ремонтное, энергетическое и инструментальное обслуживание производства | 8 |
| 5. Бухгалтерский учет, финансовые расчеты, учет пенсий и пособий, учет страховых операций, качество продукции | 13 |
| 6. Управление кадрами | 13 |
| 7. Техничко–экономическое планирование | 13 |
| 8. Материально–техническое снабжение, реализация и сбыт готовой продукции | 13 |