МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

# ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Отчет по лабораторно работе №3

**«Работа с таблицами, ISP/JSTL, паттерн command, JRA»**

Студентка: Назарчук Дина гр.272301

МИНСК 2015

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc420971879)

[1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И КЛАССИФИКАЦИЯ СУЩНОСТЕЙ ДЛЯ РЕШАЕМОЙ ЗАДАЧИ 4](#_Toc420971880)

[2 ПОСТАНОВКА ЗАДАЧИ И ОБЗОР МЕТОДОВ ЕЕ РЕШЕНИЯ 7](#_Toc420971881)

[2.1 Задачи, решаемые системой «Расчёта затрат на предприятии» 7](#_Toc420971882)

[2.2 Требования к системе 7](#_Toc420971883)

[2.3 Спецификация вариантов использования системы «Расчет затрат на промышленном предприятии» 8](#_Toc420971884)

[3 МОДЕЛИ ПРЕДСТАВЛЕНИЯ СИСТЕМЫ И ИХ ОПИСАНИЕ 10](#_Toc420971885)

[3.1 Диаграмма классов системы «Расчёт затрат на промышленном предприятии» 10](#_Toc420971886)

[3.1.2 Диаграмма последовательности 13](#_Toc420971887)

[3.1.3 Диаграммы состояния основных объектов системы «Расчёт затрат на предприятии» 15](#_Toc420971888)

[3.2 Диаграмма компонентов системы «Расчёт затрат на предприятии» 16](#_Toc420971889)

[3.3. Диаграмма развертывания системы 17](#_Toc420971891)

[4 ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ И ЕЕ ОПИСАНИЕ 18](#_Toc420971892)

[5. РУКОВОДСТВО ПО РАЗВЕРТЫВАНИЮ СИСТЕМЫ 21](#_Toc420971893)

[5.1 Руководство по настройке сервера 21](#_Toc420971894)

[5.2 Руководство по настройке базы данных 21](#_Toc420971895)

[6 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ СИСТЕМЫ «УЧЕТ ПРОДАЖ КОМПЬЮТЕРНОЙ ТЕХНИКИ» И ОЦЕНКА ВЫПОЛНЕННЫХ ЗАДАЧ 25](#_Toc420971896)

[6.1 Тестовый пример работы приложения 25](#_Toc420971897)

[ЗАКЛЮЧЕНИЕ 32](#_Toc420971898)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 33](#_Toc420971899)

[ПРИЛОЖЕНИЕ А](#_Toc420971900)(обязательное) [Функциональная модель (IDEF0) 34](#_Toc420971902)

[ПРИЛОЖЕНИЕ Б (обязательное) Блок-схемы 38](#_Toc420971903)

[ПРИЛОЖЕНИЕ В](#_Toc420971904) (обязательное) [Листинг код 39](#_Toc420971906)

[ПРИЛОЖЕНИЕ Г (обязательное) Листинг скрипта 42](#_Toc420971907)

# 

# ВВЕДЕНИЕ

На этапе перехода к рыночным отношениям многие предприятия, утратив государственную поддержку, стали самостоятельными субъектами экономической жизни. Вопросы выбора вида деятельности, организации производственного процесса, выпуска и сбыта готовой продукции стали прерогативой самих предприятий. В условиях самофинансирования основной задачей для предприятий стала задача получения прибыли в размере, достаточном для воспроизводственного процесса. Основным условием увеличения прибыли предприятия является снижение издержек производства и сбыта продукции, в частности себестоимости выпускаемой продукции. Поэтому вопросы управления затратами являются весьма актуальными для предприятий.

Затраты **–** это денежная оценка стоимости материальных, трудовых, финансовых, природных, информационных и других видов ресурсов на производство и реализацию продукции за определенный период времени. Актуальность. Управление затратами является важнейшим направлением управленческой деятельности, поскольку именно соотношение между доходами и расходами предприятия определяет в конечном счете его финансовое состояние, прибыльность и перспективы развития.

Управление затратами осуществляется исходя из возможностей их контроля и регулирования. Система контроля и регулирования затрат обеспечивает полноту и правильность действий в будущем, направленных на снижение затрат и рост эффективности производства.

Целью курсового проекта является анализ и совершенствование системы управления затратами предприятия.

В рамках курсового проекта должны быть реализованы следующие задачи:

1) раскрыть понятие и классификацию затрат;

2) провести анализ системы управления затратами на предприятии;

3) применить паттерн проектирования;

4) обосновать оригинальные решения по использованию технических средств

5) описать алгоритмы реализации бизнес-логике системы;

Объектом исследования является предприятие по производству молочной и пирожной продукции.

Предметом курсовой работы является система управления затратами на предприятии

# 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И КЛАССИФИКАЦИЯ СУЩНОСТЕЙ ДЛЯ РЕШАЕМОЙ ЗАДАЧИ

Поэлементная классификация затрат необ­ходима для определения заданий по снижению себестоимости продукции. По элементам группируются затраты, однородные по экономическому содержанию (материальные, трудовые, затраты, имеющие характер комплексных денежных поступлений).  
 Успешность функционирования любого предприятия определяется уровнем ее конкурентоспособности. Наиболее конкурентоспособным является то предприятие, продукция которого имеет наилучшее соотношение «цена-качество».

Как известно, цена товара определяется затратами на его производство. От затрат зависит будущая прибыль предприятия. Чем выше прибыль, тем больше средств направляется на расширение, техническое перевооружение производства, разработку и внедрение новых видов продукции. Следовательно, важнейшим условием развития и расширения предприятий является умелое управление затратами на производство продукции.

Под управлением понимается целенаправленное воздействие субъекта управления на его объект для достижения определенных результатов. Следовательно, управление затратами – это воздействие на них с целью улучшения результатов деятельности предприятия, достижения высокого экономического результата. Оно распространяется на все элементы управления.

Основными принципами управления затратами являются:

* управление затратами на всех стадиях жизненного цикла продукции. Жизненный цикл продукции составляет процесс создания, разработки, производства, эксплуатации, обращения и утилизации продукта;
* системный подход к управлению затратами. Данный принцип предполагает изучение объекта управления и управляющей системы совместно и нераздельно. Системный подход означает необходимость использования системного анализа и синтеза в каждом управленческом решении. Данный подход находит выражение в том, что эффективность управления затратами оценивают по эффективности самого слабого звена системы;
* единство методов, практикуемых на разных уровнях управления затратами. Методическое единство предполагает единые требования к информационному обеспечению, планированию, учету, анализу затрат. Необходимо единство и соподчиненность используемых критериев эффективности;
* органическое сочетание снижения затрат с высоким качеством продукции. Конкурентоспособность предприятия во многом зависит от конкурентоспособности продукции, которая определяется соотношением цены и качества. Противоречие заключается в том, что повышение качества продукции сопровождается увеличением затрат, а следовательно, и увеличением цены. Оптимальный баланс между качеством и затратами на всех стадиях жизненного цикла продукта достигается грамотным управлением на основе исследований и экономических расчетов;
* недопущение излишних затрат;
* широкое внедрение эффективных методов снижения затрат;
* повышение заинтересованности всех подразделений предприятия в снижении затрат.

В соответствии с Основными положениями по составу затрат, включаемых в себестоимость продукции (работ, услуг) группи­ровка затрат содержит следующие элементы:

1. В элементе "Материальные затраты" отражается стоимость сырья, материалов, комплектующих изделий и полуфабрикатов, топлива и энергии всех видов, транспортных услуг сторонних организаций, работ и услуг производственного характера, износ малоценных и быстроизнашивающихся предметов.
2. В элемент "Затраты на оплату труда" включаются расходы на оплату труда основного производственного персонала предпри­ятия.
3. В элементе "Затраты на перевозку товара" включаются расходы на доставку товара от производителя заказчику.
4. К элементу "Прочие затраты" относятся все другие затраты, не вошедшие в ранее перечисленные элементы затрат: налоги, сборы.

В данном курсовом проекте были использованы следующие виды затрат:

* 1. Расчет затрат на сырьё:

,

где Ктр – коэффициент, учитывающий транспортно-заготовительные расходы;

n – номенклатура применяемых материалов;

Нp i – норма расхода сырья, материалов (основного и вспомогательного)   
i-го вида на единицу продукции (кг, м, л и пр.);

Цi – цена за единицу материала i-го вида (без учета НДС), ден. ед.

* 1. Расчет затрат на производство изделия:

*Pi=Ni\*Сi* ,

где Ni **– количество закупаемого сырья некоторого вида, ед.**

***Сi* – производственная себестоимость единицы продукции, ден.ед.**

* 1. Расчет затрат на реализацию:

,

где Hком **– норматив расходов на реализацию, %.**

**С*i* – производственная себестоимость единицы продукции, ден.ед.**

* 1. Расчет затрат на основную заработную плату производственных рабочих:

 ,

где Кnp – коэффициент премий за выполнение плановых показателей, согласно действующему законодательству (не более 1,3);

К – количество технологических операций, выполняемых при производстве изделия;

Тч i – часовая тарифная ставка, соответствующая разряду работ i-й операции, ден. ед./ч;

ti – норма времени на выполнение i-й операции, ч/шт.;

*Зд* – дополнительная заработная плата производственных рабочих.

* 1. Расчет отчислений в фонд социальной защиты населения, на обязательное страхование:

,

где Нсоц – установленный законодательством процент отчислений в фонд социальной защиты населения, на обязательное страхование.

# 2 ПОСТАНОВКА ЗАДАЧИ И ОБЗОР МЕТОДОВ ЕЕ РЕШЕНИЯ

# 2.1 Задачи, решаемые системой «Расчёта затрат на предприятии»

Для успешной реализации системы необходимо, в первую очередь, выделить основные задачи, которые будет решать система «Расчёта затрат», а также те задачи, которые необходимо выполнить для правильной работы системы.

Основными задачами являются следующие:

1) Разработка базы данных для MySQLServer 5.5.

2) Система должна работать в рамках архитектуры web-приложение;

3) Система должна предоставлять удобный и минималистичный интерфейс для конечного пользователя;

4) В клиентской части программы со стороны администратора должны быть предусмотрены следующие возможности:

– Добавление, просмотр, изменение, удаление всех доступных данных;

– Просмотр рассчитанных затрат;

– Добавление и удаление новых пользователей;

5) В клиентской части программы со стороны пользователя должны быть предусмотрены следующий возможности:

– Просмотр информации о заказах, рабочих, изделиях, сырье;

– Просмотр рассчитанных затрат;

Программа дает возможность работнику улучшить производительность труда, посредством быстрого доступа ко всей интересующей информации. Со стороны клиента, данная система дает возможность посмотреть всю интересующую информацию о рассчитанных затратах.

# 2.2 Требования к системе

В связи с поставленными задачами, необходимо придерживаться следующих требований к системе:

1) В качестве языка для реализации выбран объектно-ориентированный язык Java.В данной работе будут реализованы: собственная иерархия классов, расширение базовых классов предоставляемых SDK, паттерн проектирования MVC, интерфейсы и абстрактные классы, обработка исключительных ситуаций.

2) СУБД – MySQL 5.5. Для разработки информационной модели системы был использован пакет моделирования MySQL Workbench. Доступ к данным в СУБД осуществляется через драйвер, предоставляемый производителем СУБД.

3) База данных приведена к 3-ей нормальной форме.

4) База данных должна генерироваться sql-скриптом.

5) Функционал серверной части не менее 5UseCase’ов (вариантов использования).

6) Паттерн реализации: «MVC».

8) Операционная система WindowsXPSP3.

## **2.3 Спецификация вариантов использования системы «Расчет затрат на промышленном предприятии»**

Диаграмма вариантов использования позволяет провести анализ использования системы, то есть провести описание ее основного предназначения.

Хорошо продуманная диаграмма вариантов использования является одни из самых важных видов моделей. Это объясняется тем, что четкая постановка, знание и организация задач особенно важны для их успешного выполнения.

Актерами, которые работают с данной системой являются администратор, который имеет полный доступ к системе и менеджер, который имеет все привилегии расчёта затрат на предприятии.

В данной работе будут реализованы следующие варианты использования:

1. Авторизоваться как администратор – авторизация на клиентской стороне программы в качестве администратора. Права администратора обеспечивают полный доступ к управлению системой.
2. Авторизоваться как менеджер – авторизация на клиентской стороне программы в качестве менеджера. Права менеджера обеспечивают неполный доступ к управлению системой.
3. Работать с данными – расширено с помощью вариантов использования «Добавить», «Удалить», «Просмотреть», «Изменить», «Сохранить».
4. Удалить администратора – удаление администратора, который более не должен иметь возможности доступа к системе с правами полного доступа.
5. Просмотреть данные любой таблицы – просмотр разрешен для всех актеров. Также есть возможность просмотра графика затрат.
6. Отсортировать – позволяет пользователю системы отсортировать доступную для просмотра информацию по различным критериям.
7. Расчёт всех видов затрат на предприятии – разрешено для любого пользователя , чтобы узнать информацию о затратах на предприятии.

Диаграмма вариантов использования данного курсового проекта представлена на рисунке 2.3.



Рисунок 2.3 – Диаграмма вариантов использования

# 

# 3 МОДЕЛИ ПРЕДСТАВЛЕНИЯ СИСТЕМЫ И ИХ ОПИСАНИЕ

## **3.1 Диаграмма классов системы «Расчёт затрат на промышленном предприятии»**

Диаграммы классов являются самым важным и наиболее часто используемым видом диаграмм . Диаграммы классов называют статической моделью системы, потому что они не отображают функционирование системы во времени. Диаграммы классов представляют элементы, из которых состоит система, и взаимосвязи между этими элементами. Диаграммы классов позволяют использовать для представления системы интерфейсы, пакеты, наследование и прочее.



Рисунок 3.1.1 –Диаграмма классов для пакета bsuir.controller



Рисунок 3.1.2 –Диаграмма классов для пакета bsuir.model.pojo



Рисунок 3.1.3 –Диаграмма классов для пакета bsuir.model.dao



Рисунок 3.1.4 –Диаграмма классов для пакета bsuir.model.dao.implement



Рисунок 3.1.5 –Диаграмма классов для пакета bsuir.model.service



Рисунок 3.1.6 –Диаграмма классов для пакета bsuir.model.service.impelement

# 3.1.2 Диаграмма последовательности

Диаграмма последовательности используется для того, чтобы показать, как несколько объектов поддерживают выполнение функций прецедента[1]. Диаграммы последовательности хорошо подходят для отображения того, как объекты взаимодействуют в прецеденте, но они малоэффективны для описания специфических алгоритмов системы.

Диаграммы последовательности также очень полезны в выявлении классов и методов. Классы легко можно определить как имена существительные экземпляров объектов, а методы могут быть сообщениями, которые в них вызываются. Сразу параметры методов могут быть не очевидны, но выявление классов и методов является хорошей отправной точкой для дальнейшего проектирования системы.

Диаграмма последовательности авторизации в программе и просмотра статистики представлена на рисунке 3.1.12

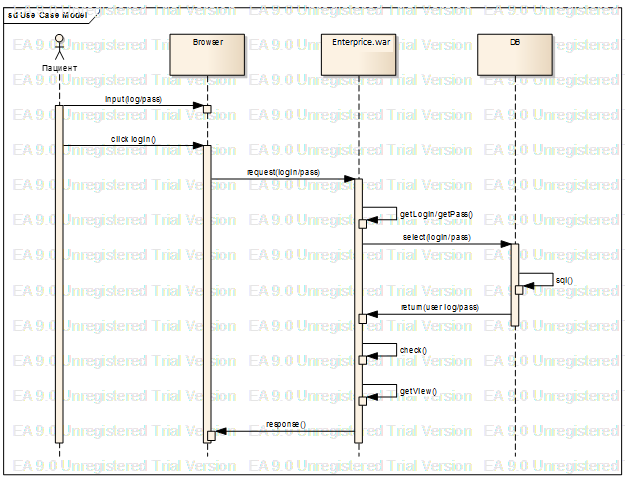


Рисунок 3.1.7 – Диаграмма последовательности авторизации

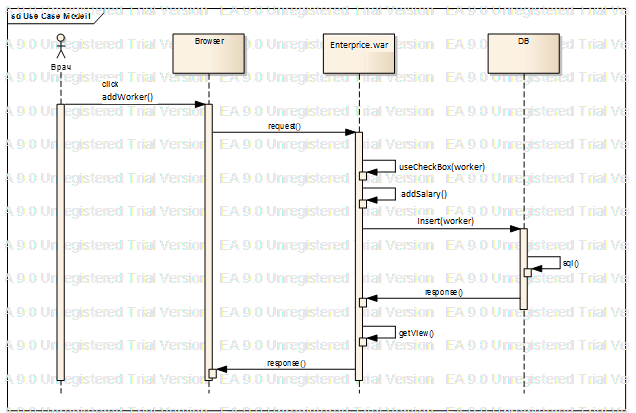


Рисунок 3.1.8 – Диаграмма последовательности добавления рабочего

## **3.1.3 Диаграммы состояния основных объектов системы «Расчёт затрат на предприятии»**

Диаграмма состояния является эффективной инструментом отображения состояния объекта при выполнении большинства прецедентов, а также хорошим способом представления протоколов, описывающих правильную последовательность сообщений, например, протоколов для доступа к базам данных или для обеспечения связи на основе ТСР.

Диаграмма состояния данного проекта описывает состояние таблицы с заказами , этапы, через которые этот объект проходит.

Диаграмма состояния для данного проекта представлена на рисунке 3.1.18

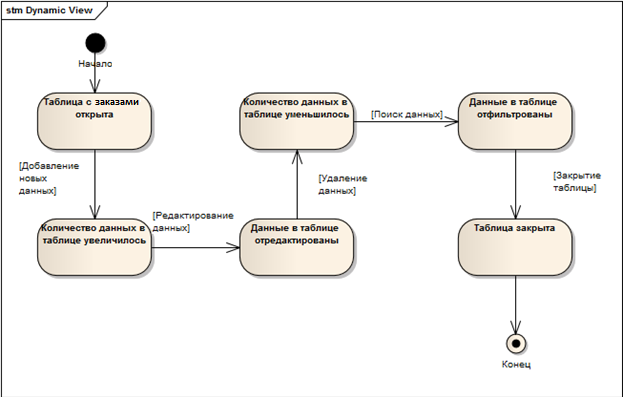


Рисунок 3.1.9 – Диаграмма состояния системы для каждой таблицы

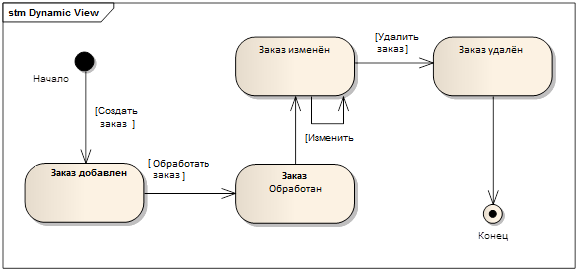


Рисунок 3.1.10 – Диаграмма состояния системы для таблицы с заказами

# 3.2 Диаграмма компонентов системы «Расчёт затрат на предприятии»

Диаграмма компонентов–статическая диаграмма, показывающая разбиение программной системы на компоненты, а также связи между ними. Чаще всего в качестве компонентов выступают различные библиотеки, исполняемые файлы, пакеты, интерфейсы, протоколы взаимодействия и т.д.

Между собой компоненты связываются благодаря зависимостям, когда соединяются интерфейсы двух различных компонентов. Отношение зависимости показывает, что один компонент предоставляет другому компоненту интерфейс для взаимодействия.

Также существует связь делегирования – она показывает связь внешнего компонента с внутренней реализацией этого поведения внутренними компонентами

Диаграмма компонентов представлена на рисунке 3.3.1

# 

Рисунок 3.3.1– Диаграмма компонентов системы

# 3.3. Диаграмма развертывания системы

Диаграмма развертывания показывает, как будет выглядеть система, когда она будет введена в эксплуатацию. Данный вид диаграмм показывает логические элементы системы, их физическое размещение, то, как эти элементы взаимодействуют, а также их количество[1].

Диаграмма развертывания показывает, как приведенные ниже элементы системы связаны друг с другом, какие протоколы они используют, какие операционные системы и физические устройства, включая компьютеры и другие вычислительные средства, имеются в информационной системе и в каком порядке необходимо проводить развертывание системы.

Диаграммы развертывания не так сложно создавать, они в основном не содержат большого количества элементов и применяются только для приложений средней и высокой сложности. Данные диаграммы эффективны для визуализации среды развертывания системы со множеством составных элементов.

Диаграмма развертывания представлена на рисунке 3.4.1

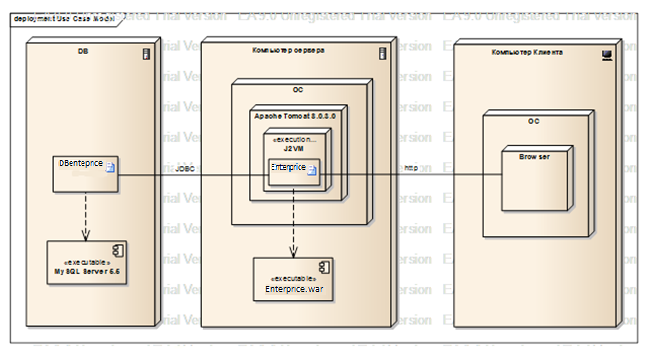


Рисунок 3.4.1 – Диаграмма компонентов системы

# 4 ИНФОРМАЦИОННАЯ МОДЕЛЬ СИСТЕМЫ И ЕЕ ОПИСАНИЕ

Для построения и описания информационной модели было использовано техническое средство MySqlWorkbench.

Основными компонентами для представления информационной модели являются сущности, связи и атрибуты. Каждая сущность отражает множество уникальных объектов. Атрибут характеризует некое свойство сущности. Связи показывают зависимости между различными сущностями. В конкретной БД сущности соответствует таблица, атрибуту – поле таблицы, а связям – внешние ключи. Под построением информационной модели имеется в виду выделение необходимых сущностей, атрибутов и связей.

При разработке информационной модели разрабатываются два разных уровня – логический и физический. На рисунке 4.1 представлен логический уровень. Логический уровень , который показывает модель так, как она выглядит в реальной жизни. На рисунке 4.2 представлен физический уровень. Физический уровень зависит от конкретной СУБД, а значит является зависимым от реализации СУБД.

Рассмотрим подробнее каждую сущность данной модели, а также связи между ними.

Сущность «Предприятие» используется для хранения учетных данных предприятий , которые делают заказ. Ключевым полем является id . Неключевое поле: Название – содержит название предприятия , которое будет делать заказ.

Сущность «Заказ» используется для хранения информации о заказах . Данная сущность связана с сущностью «Предприятие» связью один ко многим, так как одно предприятие-заказчик может делать несколько заказов.. Ключевым полем является id. Неключевые поля: Описание, Срок выполнения, Стоимость доставки, Масса, Предприятие\_id , которое является внешним ключом.

Сущность «Рабочий» используется для хранения информации рабочих , которые работают на предприятии. Ключевым полем является id. Данная сущность связана связью один ко многим с сущностью «Заказ», т.к. один заказ может быть выполнен несколькими рабочими. Неключевые поля: Фамилия , Имя , Отчество , Возраст , Стаж , Отработано часов , Часовая ставка, Размер премии , Логин , Пароль, Должность(внешний ключ), Заказ\_id(внешний ключ).

Сущность «Должность» используется для хранения информации о должностях. Данная сущность связана с сущностью «Рабочий» связью многие ко многим. Ключевым полем является id. Неключевые поля: Должность.

Сущность «Изделие» используется для хранения информации об изделиях, которые создает наше предприятие. Данная сущность связана с сущностью «Заказ» один ко многим. Т.к. в одном заказе может содержаться сразу несколько изделий.

Ключевым полем является id. Неключевые поля: Наименование, Количество, Артикул, Описание , Масса , Заказ\_id ( внешний ключ).

Сущность «Сырье» используется для хранения информации о сырье, из которого изготавливается изделие. Данная сущность связана с сущностью «Изделие» связью один ко многим, т.к. одно изделие может содержать много видов сырья. Ключевое поле id. Неключевые поля: Наименование, Артикул, Количество , Стоимость, Заметки, Изделие\_id( внешний ключ).

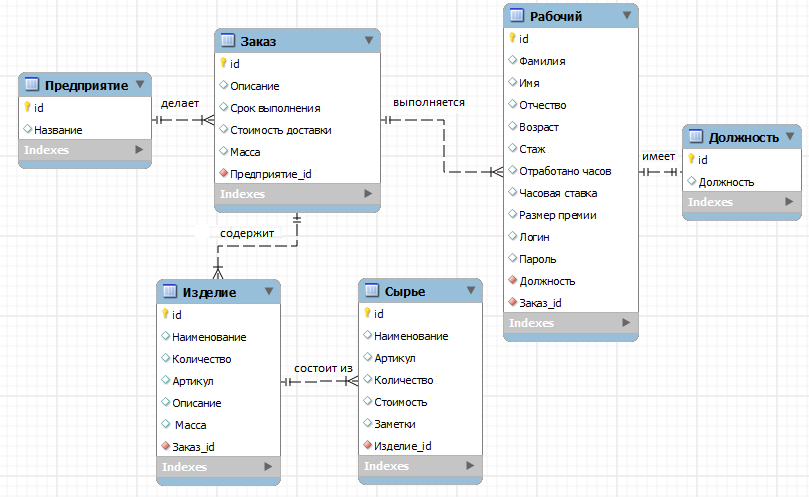


Рисунок 4.1 – Логический уровень информационной модели

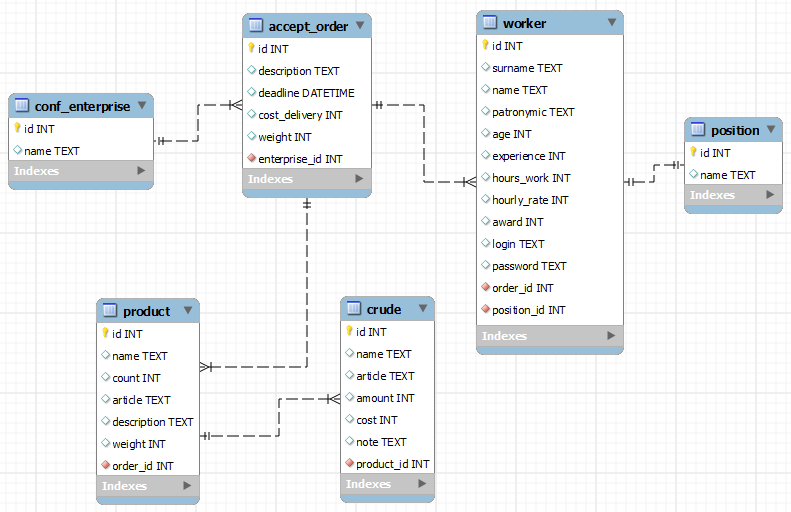


Рисунок 4.2 – Физический уровень информационной модели

5. РУКОВОДСТВО ПО РАЗВЕРТЫВАНИЮ СИСТЕМЫ

5.1 Руководство по настройке сервера

Для запуска сервера необходимым требованием является наличие JREверсии не ранее, чем 1.7, а также MySQL Server 5.5, Navicat Premium, Tomcat 8.0.3.0.

В качестве БД для сервера выступает MySQL 5.5. Доступ к БД осуществляется через файл настроек.

5.2 Руководство по настройке базы данных

Необходимо установить программу Navicat Premium. Послее установки открываем ее и выполняем следующие действия:

1) Создать подключение к MySQL. Показано на рисунках 5.1­– 5.2.

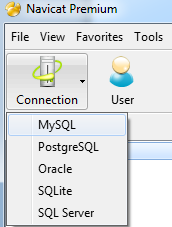


Рисунок 5.1 – Создание подключения к MySQL

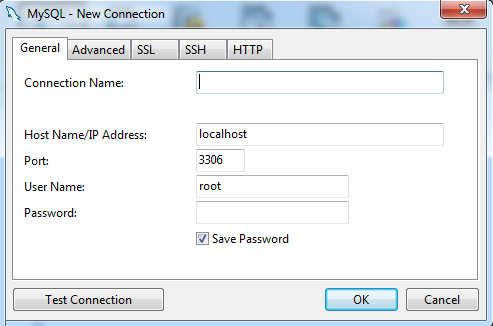


Рисунок 5.2 – Заполнение выбранного подключения

2) Открыть созданное подключение и создать новую базу данных. Показано на рисунках 5.3­– 5.4.

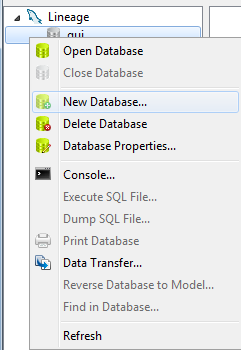


Рисунок 5.3 – Создание новой базы данных

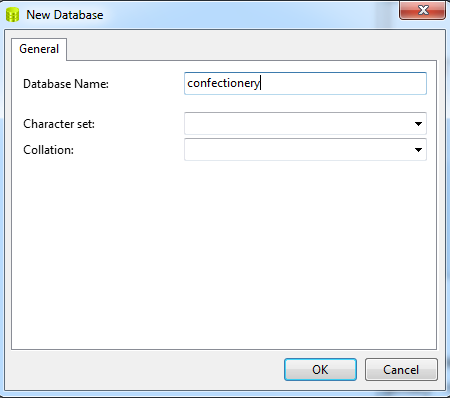


Рисунок 5.4 – Заполнение окна новой базы данных

3) Открыть созданную базу данных и в контекстном меню выбрать «ExecuteSQLFile…», выбрать путь к файлу и нажать кнопку «Start». После успешного завершения нажать кнопку «Close».Показано на рисунках 4.5­– 4.7

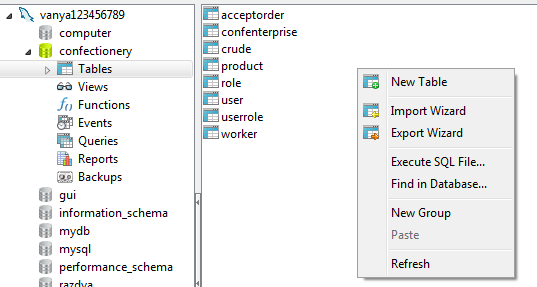
.

Рисунок 5.5 – Импортирование SQLфайла

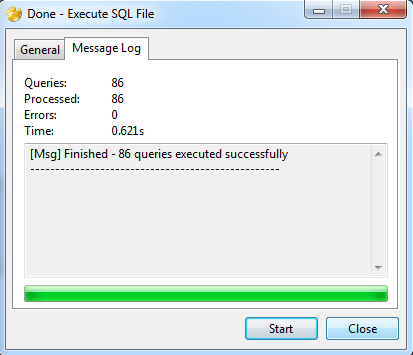


Рисунок 5.6 – Завершение импорта SQLфайла

6 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ СИСТЕМЫ «УЧЕТ ПРОДАЖ КОМПЬЮТЕРНОЙ ТЕХНИКИ» И ОЦЕНКА ВЫПОЛНЕННЫХ ЗАДАЧ

6.1 Тестовый пример работы приложения

Пользователь работает с клиентской стороной программы с помощью Browser. Для начала, фигуранту необходимо перейти по URL адресу. После выполнения этих действий, программа предложит пользователю авторизоваться (войти в систему). Для этого необходимо ввести свой логин и пароль. После входа в систему пользователь имеет возможность работать с данными в зависимости от своих прав доступа. Пользователь, вошедший с правами «Администратор» может просматривать, удалять, изменять и добавлять любые данные в программе. Пользователь, вошедший с правами «Пользователь» может лишь просматривать часть данных. Для обеспечения безопасности все пароли шифруются, обрабатывается неправильный ввод при авторизации и другие исключительные ситуации.

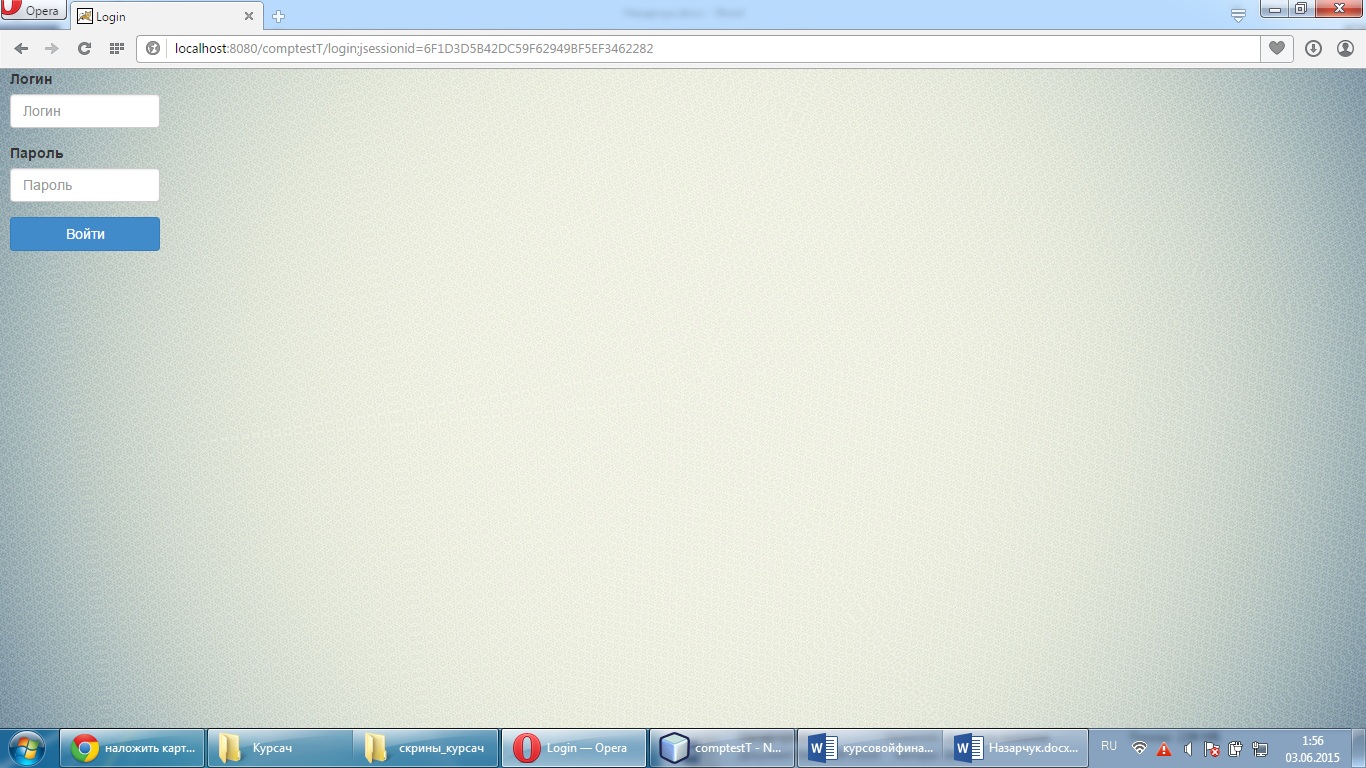


Рисунок 6.1 – Страница «Логина»

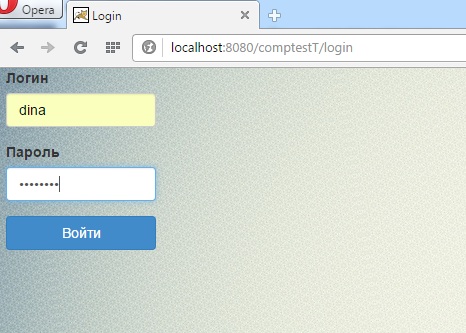


Рисунок 6.2 – Выпадающее меню входа в систему

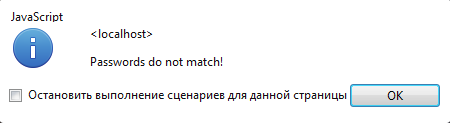


Рисунок 6.3 – Проверка на правильность ввода пароля

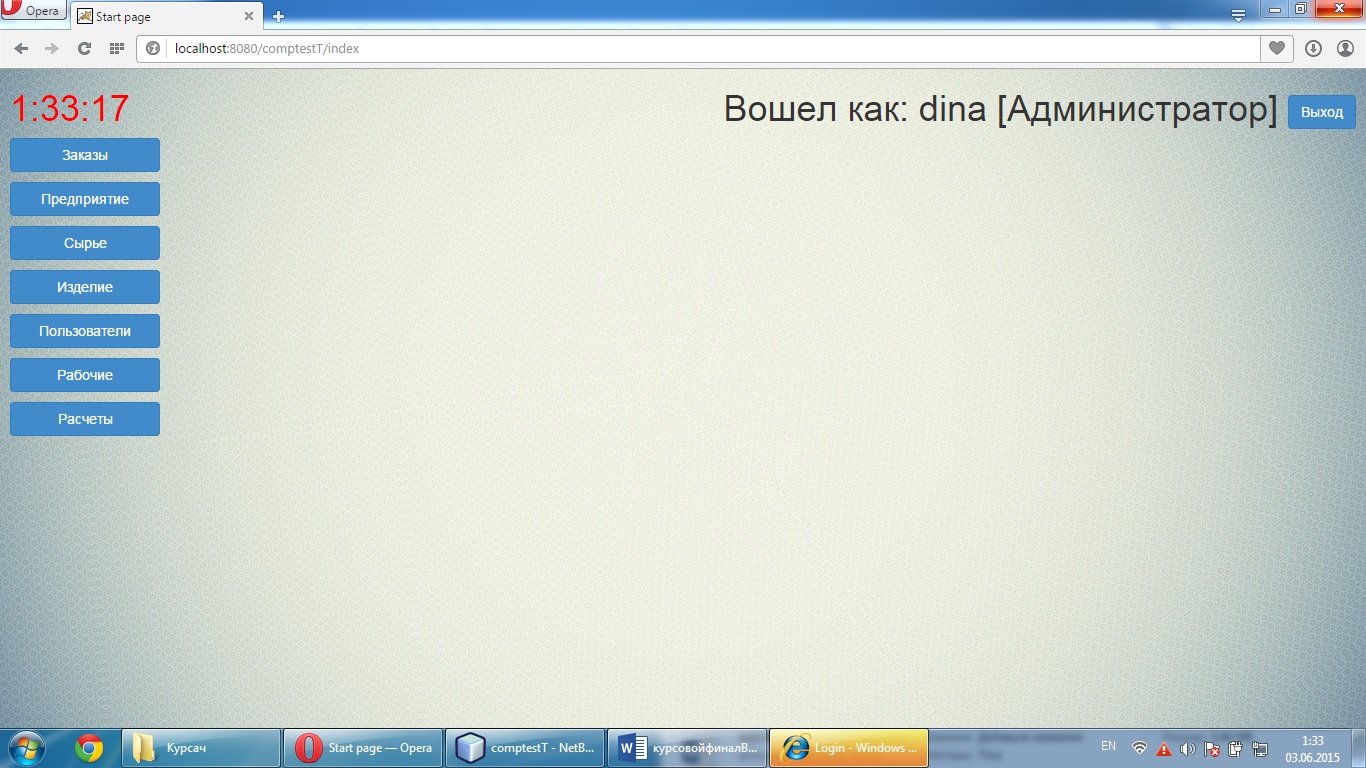


Рисунок 6.4 – Индекс страница после успешной авторизации

После авторизации доступны (в зависимости от роли) возможности для пользователей. Так же показано, с какого логина и с какой ролью был произведен вход в систему.

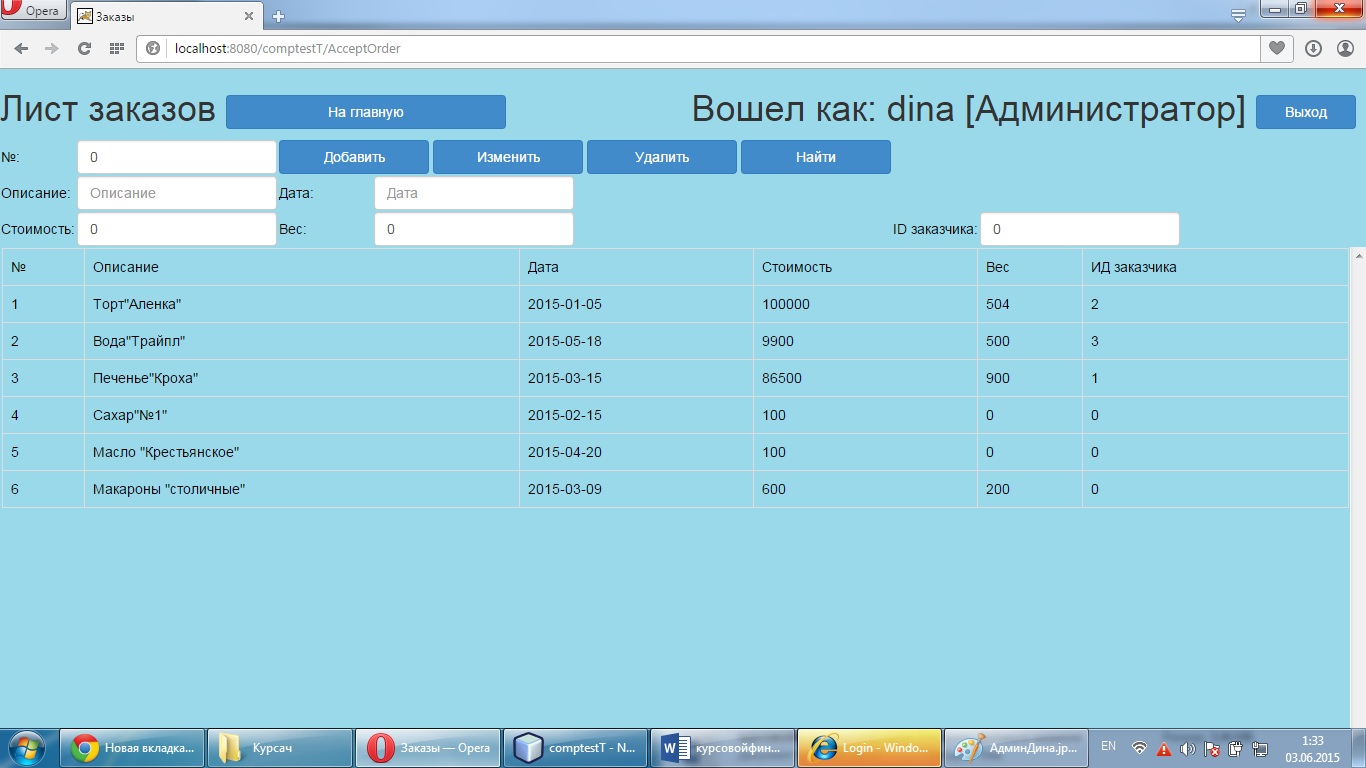


Рисунок 6.5 – Страница с базой заказов



Рисунок 6.6 – Страница с базой заказчиков

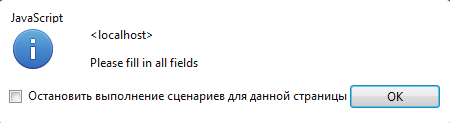


Рисунок 6.7 – Проверка на заполнение всех полей

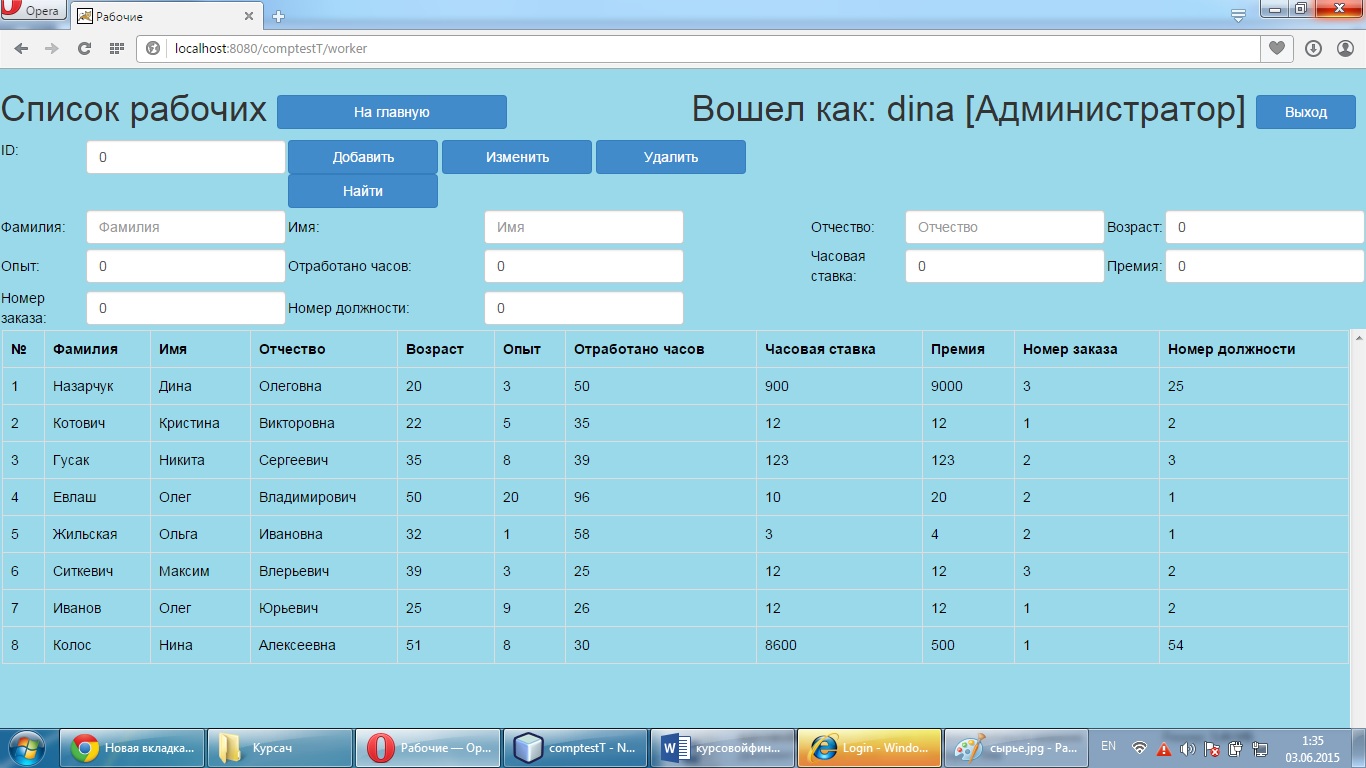


Рисунок 6.8 – Страница с базой рабочих

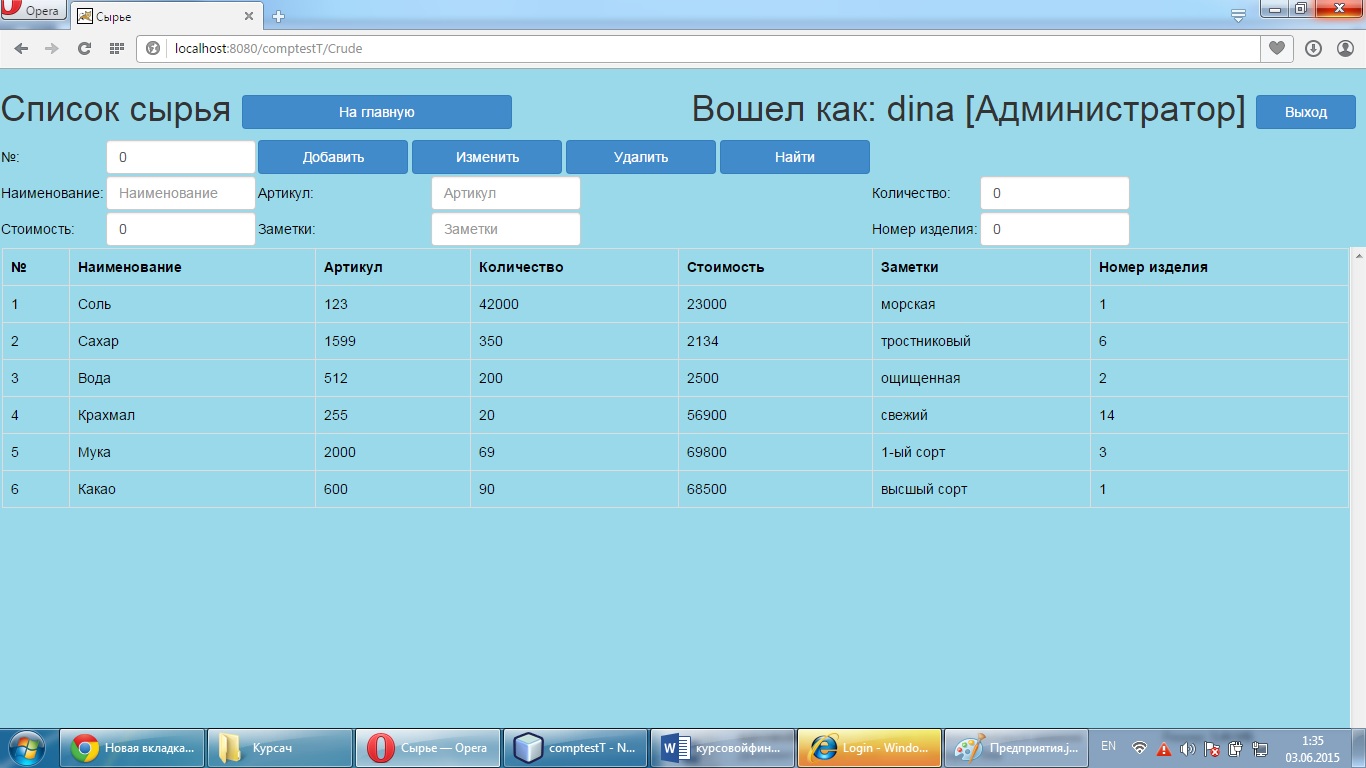


Рисунок 6.9 – Страница со списком сырья

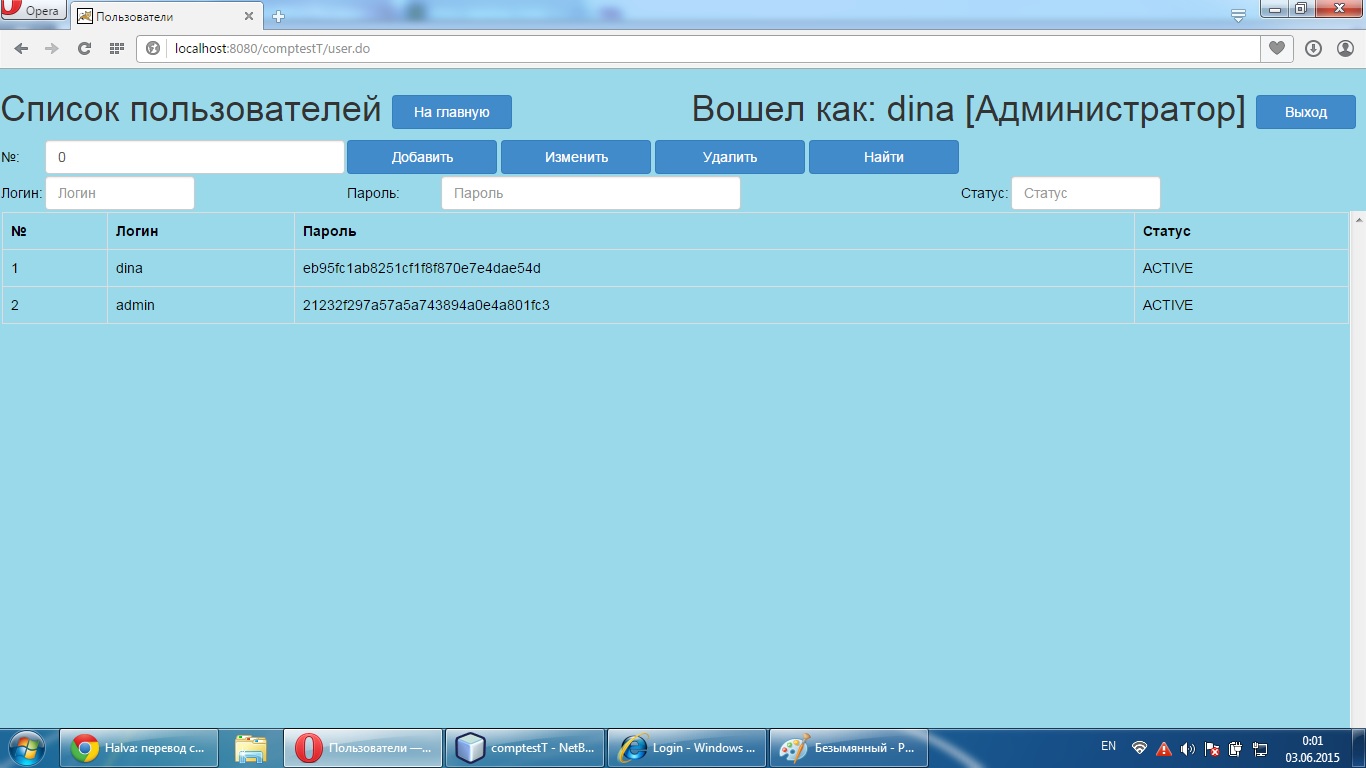


Рисунок 6.10 – Страница со списком пользователей

Пароли хранятся в зашифрованном виде с использованием MD5.

Остальные страницы имеют похожий интерфейс. На всех страницах: для администратора доступны все функции (добавление, удаление, редактирование, поиск). Так же имеется кнопки возврата на главную страницу и клавиша «Выход» для выхода из системы.

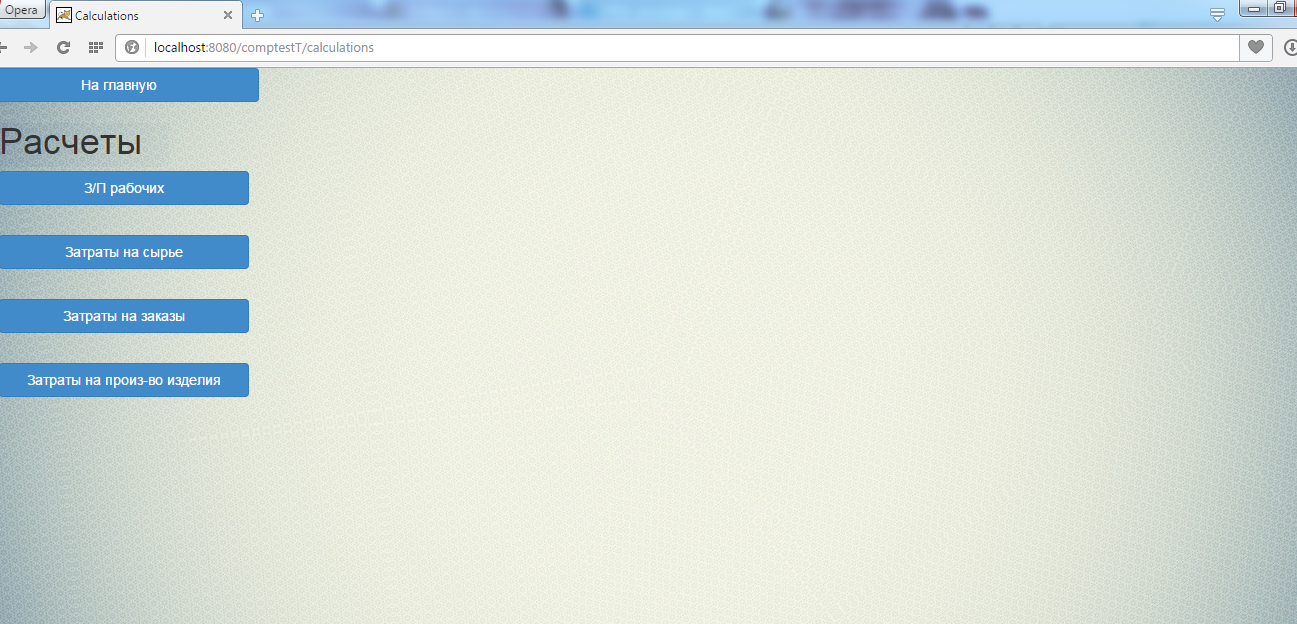


Рисунок 6.11 – Страница расчёта затрат.

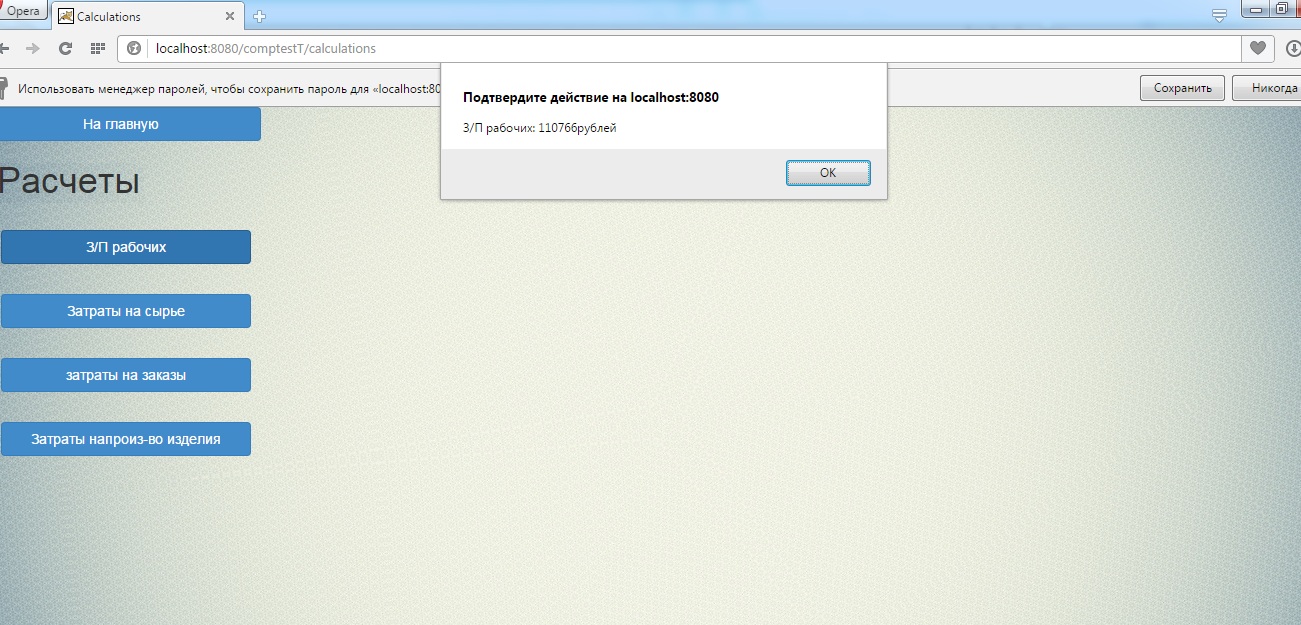


Рисунок 6.12 – Результат расчёта

# ЗАКЛЮЧЕНИЕ

Результатом выполнения данной работы является программа, которая представляет собой систему расчёта затрат на промышленном предприятии.

Оценивая проделанную работу, можно сделать вывод, что все основные поставленные цели были выполнены, а именно:

– Программа предоставляет возможность легко и просто просматривать и управлять затратами на предприятии;

– На сервере предусмотрено параллельное выполнение запросов;

– Безопасность обеспечивается за счет разграничения прав доступа, системы авторизации и шифрования паролей;

– Для удобства пользователя программа располагает удобным интерфейсом;

Данный продукт может быть использован , любым работникам предприятия ,

чтобы оценить его положение на рынке.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1]Киммел, П. UML Универсальный язык проектирования / П. Киммел. – М.: НТ Пресс, 2008.

[2]Основы и лингвистическое обеспечение баз данных: учебно-методическое пособие. Орешко И.Г. [и др.]. – Минск: БГУИР, 2007.

[3][Гамма](http://oz.by/people/more905851.html) Э.Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, [Р. Хелм](http://oz.by/people/more908690.html). – СПб.: Питер, 2013.

[4] Герберт Ш. Java. Полное руководство / Ш. Герберт. – М.: Вильямс,2012.

[5] Хортон А., Java 2. В 2 т. / А.Хортон – СПб.: Лори, 2013.

[6]JUnit [Электронный ресурс] – программно-ориентированный фреймворк для тестирования программ на языке Java. – Электронные данные. – Режим доступа: <http://www.junit.org>

[7]JFreeChart[Электронный ресурс] – построение графиков для JavaSwing. – Электронные данные. – Режим доступа: <http://www.jfree.org/jfreechart/>

[8] Wikipedia[Электронный ресурс] – электронная энциклопедия. – Электронные данные. – Режим доступа: <http://www.wikipedia.org>

# ПРИЛОЖЕНИЕ А

# (ОБЯЗАТЕЛЬНОЕ)

# Функциональная модель (IDEF0)

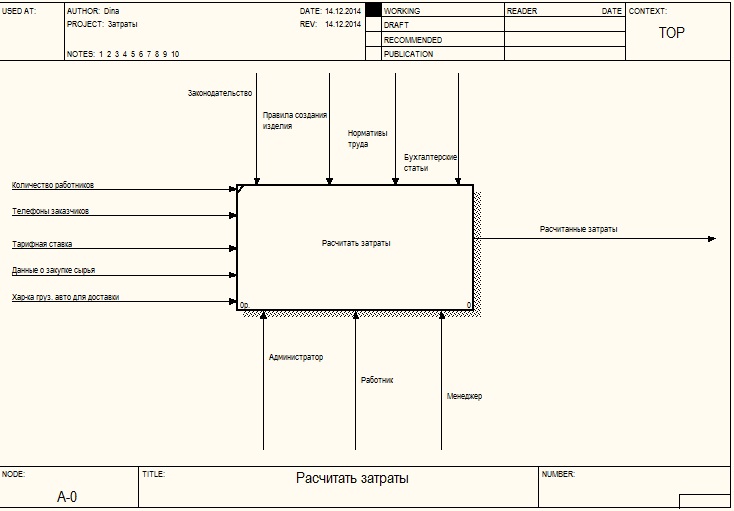


Рисунок А.1 – Главный процесс

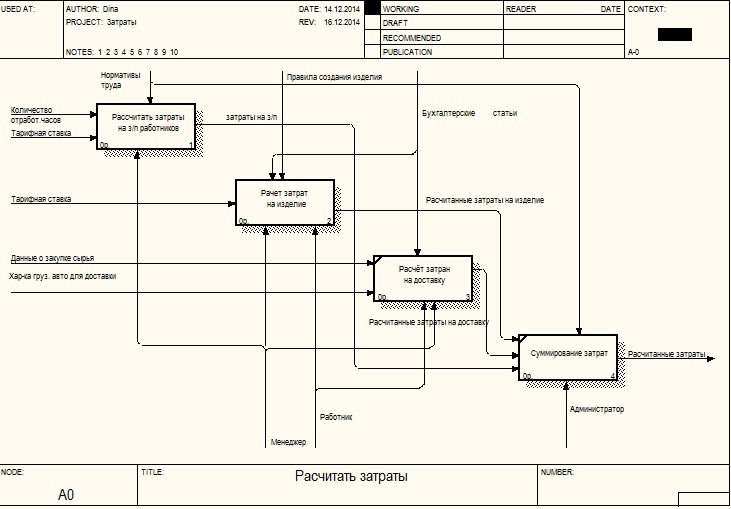


Рисунок А.2 – Декомпозиция главного процесса

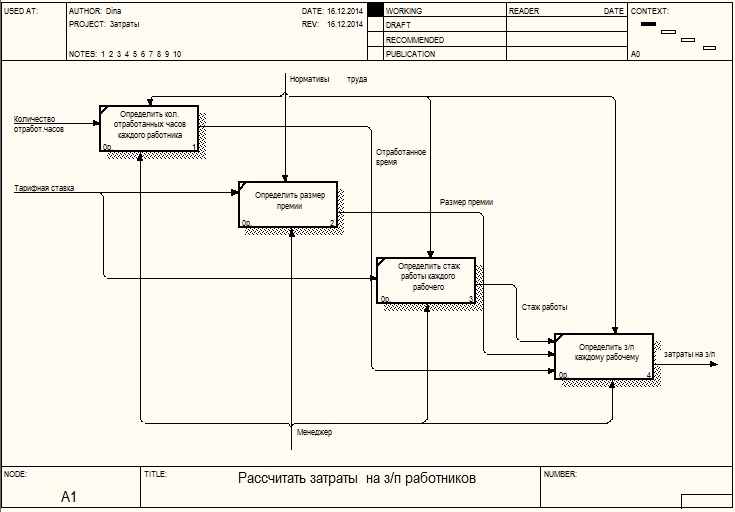


Рисунок А.3 – Декомпозиция блока «Расчёт затрат на з/п работникам»

**ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А**

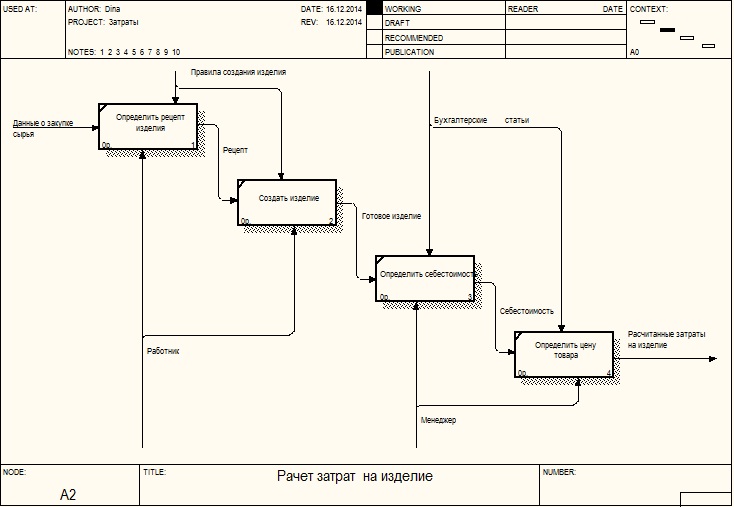


Рисунок А.4 – Декомпозиция блока «Расчёт затрат на изделие»

**ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А**

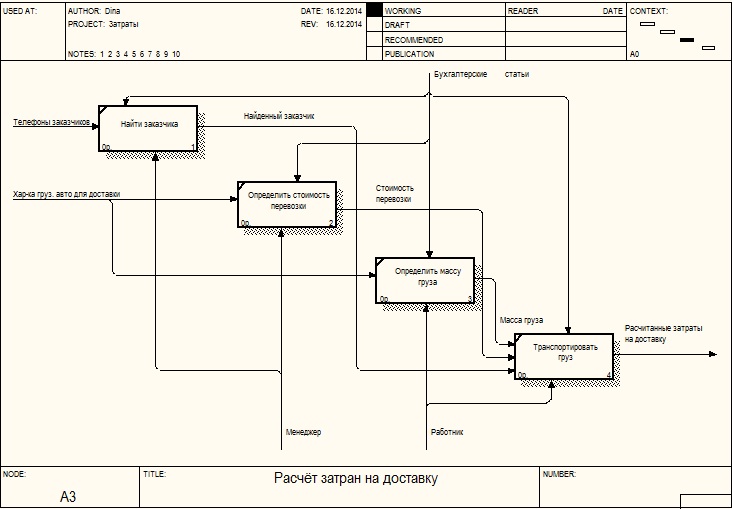


Рисунок А.5 – Декомпозиция блока «Расчёт на доставку»

# ПРИЛОЖЕНИЕ Б Блок-схемы

(ОБЯЗАТЕЛЬНОЕ)

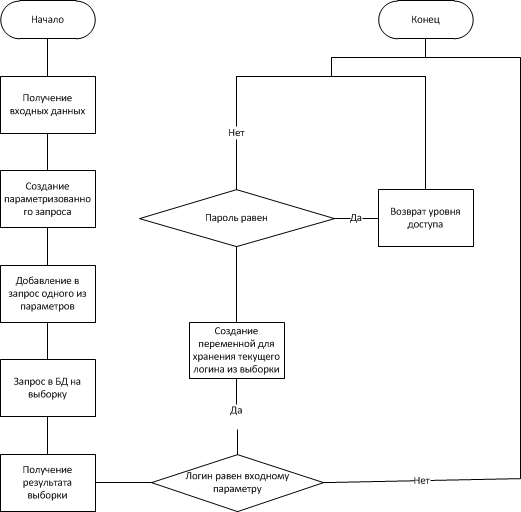


Рисунок Б.1 – Схема алгоритма авторизации

# ПРИЛОЖЕНИЕ В

# (ОБЯЗАТЕЛЬНОЕ)

# Листинг код

package bsuir.model.pojo;

import java.util.Date;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Temporal;

@Entity

public class AcceptOrder implements java.io.Serializable {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private int id;

private String description;

@Temporal(javax.persistence.TemporalType.DATE)

private Date deadline;

private int costDelivery;

private int weight;

private int enterpriseId;

public AcceptOrder() {

}

public AcceptOrder(int id, String description, Date deadline, int costDelivery, int weight, int enterpriseId) {

super();

this.id = id;

this.description = description;

this.deadline = deadline;

this.costDelivery = costDelivery;

this.weight = weight;

this.enterpriseId = enterpriseId;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public Date getDeadline() {

return deadline;

}

public void setDeadline(Date deadline) {

this.deadline = deadline;

}

public int getCostDelivery() {

return costDelivery;

}

public void setCostDelivery(int costDelivery) {

this.costDelivery = costDelivery;

}

public int getWeight() {

return weight;

}

public void setWeight(int weight) {

this.weight = weight;

}

public int getEnterpriseId() {

return enterpriseId;

}

public void setEnterpriseId(int enterpriseId) {

this.enterpriseId = enterpriseId;

}

}

**Index.css**

html {

height: 100%;

}

body {

background-image: url(../pic/w.jpg);

background-size: 100%;

background-attachment: fixed;

width: 100%;

height: 100%;

overflow: hidden;

}

form {

height: 25px;

margin-left: 10px;

}

p {

color: red;

margin-left: 10px !important;

}

tr {

color: black;

//text-align: left;

}

b {

color: #fff000;

text-align: left;

}

h1 {

font-size: 24pt;

font-family: Georgia, Times, serif;

color: black;

font-weight: normal;

}

# ПРИЛОЖЕНИЕ Г

(ОБЯЗАТЕЛЬНОЕ)

# Листинг скрипта

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

DROP SCHEMA IF EXISTS `confectionery` ;

CREATE SCHEMA IF NOT EXISTS `confectionery` DEFAULT CHARACTER SET utf8 COLLATE utf8\_general\_ci ;

USE `confectionery` ;

-- -----------------------------------------------------

-- Table `confectionery`.`conf\_enterprise`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `confectionery`.`conf\_enterprise` ;

CREATE TABLE IF NOT EXISTS `confectionery`.`conf\_enterprise` (

`id` INT NOT NULL,

`name` TEXT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `confectionery`.`accept\_order`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `confectionery`.`accept\_order` ;

CREATE TABLE IF NOT EXISTS `confectionery`.`accept\_order` (

`id` INT NOT NULL AUTO\_INCREMENT,

`description` TEXT NULL,

`deadline` DATETIME NULL,

`cost\_delivery` INT NULL,

`weight` INT NULL,

`enterprise\_id` INT NOT NULL,

PRIMARY KEY (`id`),

CONSTRAINT `fk\_Order\_Еnterprise1`

FOREIGN KEY (`enterprise\_id`)

REFERENCES `confectionery`.`conf\_enterprise` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `confectionery`.`position`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `confectionery`.`position` ;

CREATE TABLE IF NOT EXISTS `confectionery`.`position` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` TEXT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `confectionery`.`worker`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `confectionery`.`worker` ;

CREATE TABLE IF NOT EXISTS `confectionery`.`worker` (

`id` INT NOT NULL AUTO\_INCREMENT,

`surname` TEXT NULL,

`name` TEXT NULL,

`patronymic` TEXT NULL,

`age` INT NULL,

`experience` INT NULL,

`hours\_work` INT NULL,

`hourly\_rate` INT NULL,

`award` INT NULL,

`login` TEXT NULL,

`password` TEXT NULL,

`order\_id` INT NOT NULL,

`position\_id` INT NOT NULL,

PRIMARY KEY (`id`),

CONSTRAINT `fk\_Worker\_Order1`

FOREIGN KEY (`order\_id`)

REFERENCES `confectionery`.`accept\_order` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Worker\_Position1`

FOREIGN KEY (`position\_id`)

REFERENCES `confectionery`.`position` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `confectionery`.`product`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `confectionery`.`product` ;

CREATE TABLE IF NOT EXISTS `confectionery`.`product` (

`id` INT NOT NULL,

`name` TEXT NULL,

`count` INT NULL,

`article` TEXT NULL,

`description` TEXT NULL,

`weight` INT NULL,

`order\_id` INT NOT NULL,

PRIMARY KEY (`id`),

CONSTRAINT `fk\_Product\_Order`

FOREIGN KEY (`order\_id`)

REFERENCES `confectionery`.`accept\_order` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `confectionery`.`crude`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `confectionery`.`crude` ;

CREATE TABLE IF NOT EXISTS `confectionery`.`crude` (

`id` INT NOT NULL,

`name` TEXT NULL,

`article` TEXT NULL,

`amount` INT NULL,

`cost` INT NULL,

`note` TEXT NULL,

`product\_id` INT NOT NULL,

PRIMARY KEY (`id`),

CONSTRAINT `fk\_Crude\_Product1`

FOREIGN KEY (`product\_id`)

REFERENCES `confectionery`.`product` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

INSERT INTO `accept\_order` VALUES ('1', 'Торт', '2015-03-25 00:00:00', '123', '451', '1');

INSERT INTO `accept\_order` VALUES ('2', 'Шоколад', '2015-04-19 00:00:00', '4536', '4356', '1');

INSERT INTO `conf\_enterprise` VALUES ('1', 'Савушкин продукт');

INSERT INTO `crude` VALUES ('1', 'Молоко', '123', '4325423', '2345', 'Свежее', '1');

INSERT INTO `crude` VALUES ('2', 'Сахар', '2345', '1234', '2134', 'тростниковый', '1');

INSERT INTO `crude` VALUES ('4', 'Соль', '51234efsd', '2343', '2343', 'Морская', '2');

INSERT INTO `position` VALUES ('1', 'Администратор');

INSERT INTO `position` VALUES ('2', 'Менеджер');

INSERT INTO `product` VALUES ('1', 'Пирожное', '34', '124515', 'Нежное', '23', '1');

INSERT INTO `product` VALUES ('2', 'мороженое', '23', '4235', 'Лед', '54', '1');

INSERT INTO `product` VALUES ('3', 'Халва', '1234234', '12341235', '2134234', 'Азовская', '2');

INSERT INTO `worker` VALUES ('1', 'Иванова', 'Мария', 'Владимировна', '45', '23', '8', '423', '1576', '1234', '5eac43aceba42c8757b54003a58277b5', '0', '1');

INSERT INTO `worker` VALUES ('2', 'Петров', 'Перец', 'Олегович', '12', '12', '12', '12', '12', '12345', '5eac43aceba42c8757b54003a58277b5', '0', '2');

INSERT INTO `worker` VALUES ('3', 'Сидоров', '', 'мен', '123', '31', '4123', '123', '123', '123456', '5eac43aceba42c8757b54003a58277b5', '0', '3');

INSERT INTO `worker` VALUES ('4', 'Назарчук', 'Дина', 'Олеговна', '19', '6', '6', '10', '20', '3452346', '1245', '2', '1');

INSERT INTO `worker` VALUES ('5', 'Котович', 'Кристина', 'Викторовна', '7', '5', '4', '3', '4', 'qwert', '5eac43aceba42c8757b54003a58277b5', '2', '1');

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;