МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
ФГБОУ ВПО   
ИВАНОВСКИЙ ГОСУДАРСТВЕННЫЙ ХИМИКО - ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ (ИГХТУ)  
Институт управления финансов и информационных систем  
Кафедра экономики и финансов

**КУРСОВАЯ РАБОТА**   
  
по дисциплине «Базы данных»

на тему: «ПРОЕКТИРОВАНИЕ И СОЗДАНИЕ ПРОГРАММНОЙ СИСТЕМЫ  
 ДЛЯ ХЛЕБНОГО КОМБИНАТА»

Выполнил: студент 2 курса 47 группы  
Громов Антон  
Специальность: Бизнес-информатика  
Проверил: доцент, к.т.н.   
Константинов Евгений Сергеевич

Иваново 2014

Оглавление

[Задание 3](#_Toc406520864)

[Введение 4](#_Toc406520865)

[1. Техническое задание 5](#_Toc406520866)

[1.1. Термины, определения и сокращения 5](#_Toc406520867)

[1.2. Назначение системы 5](#_Toc406520868)

[1.3. Функции системы 5](#_Toc406520869)

[1.4. Структура базы данных 6](#_Toc406520870)

[2. Схема базы данных 9](#_Toc406520871)

[2.1. Выделение основных сущностей и проведение декомпозиции 9](#_Toc406520873)

[2.2. Описание функциональных зависимостей. 9](#_Toc406520874)

[3. Описание алгоритмов функционирования приложения 12](#_Toc406520875)

[3.1. Описание инструмента реализации 12](#_Toc406520877)

[3.2. Создание структуры базы данных 12](#_Toc406520878)

[3.3. Создание и реализация запросов 13](#_Toc406520879)

[4. Описание интерфейсов пользователя 26](#_Toc406520880)

[4.1. Формы 26](#_Toc406520882)

[4.2. Отчеты 38](#_Toc406520883)

[**Заключение** 42](#_Toc406520884)

[Список литературы 43](#_Toc406520885)

Задание

Пусть требуется создать программную систему, предназначенную для хлебного комбината. Такая система должна обеспечивать хранение сведений о производимых изделиях и их продажах, а также о закупах необходимых для производства продуктов. Сведения о производимых изделиях включает в себя: наименование изделия, вес, энергетическую ценность, объемом производства, дату производства, срок годности, краткую характеристику (примечания), состав, отпускную цену. Каждый ингредиент хранящийся на складе комбината характеризуется наименованием, количеством, датой поставки, сроком годности, энергетической ценностью, ценой за единицу.

Руководству комбината могут понадобиться следующие сведения:

Сколько на складе осталось продуктов в соответствии с расходами на производство?

Какова суммарная стоимость потраченных продуктов в изделии?

Какова суммарная энергетическая ценность с учетом всех составляющих продуктов?

Какова суммарная прибыль от реализации?

Необходимо предусмотреть возможность выдачи документа, представляющего отчет об остатках продуктов на складе, отчет о продажах по месяцам, отчет о закупках по годам.

Введение

Базы данных — это совокупность структур, предназначенных для хранения больших объемов информации и программных модулей, осуществляющих управление данными, их выборку, сортировку и другие подобные действия. Использование баз данных позволяет пользователю просматривать необходимую информацию, изменять ее, добавлять, копировать, удалять, сортировать, обеспечивать безопасность и т.д.

Целью данной курсовой работы является создание программной среды, предназначенной для работников хлебокомбината, которая должна обеспечивать хранение сведений о:

* производимых изделиях;
* продажах и закупах, необходимых для производства продуктов;
* количестве оставшихся продуктов на складе;
* стоимости потраченных продуктов;
* энергетической ценности производимых продуктов;
* суммарной прибыли предприятия.

Создание информационной поддержки деятельности предприятия существенно упростит труд работников, сделав работу более эффективной, т.к. вся необходимая информация будет храниться в одном месте.

1. Техническое задание

## Термины, определения и сокращения

В данной работе используются следующие термины:

Система – разрабатываемая система для хлебного завода;

Изделие – конечный продукт, производимый хлебным заводом;

Ингредиент – составной компонент изделий, производимых на хлебном заводе.

## Назначение системы

Разрабатываемая система требуется для автоматизации учета производства, закупок и реализации на хлебном комбинате.

## Функции системы

Система будет обеспечивать хранение сведений:

* о производимых изделиях;
* о продажах произведенной продукции;
* о закупках необходимых для производства продуктов.

Также, в соответствие с заданием заказчика, Система позволит получить следующие сведения:

* Количество продуктов на складе и информация о них;
* Суммарная стоимость ингредиентов для производства каждого изделия;
* Суммарную энергетическую ценность изделия, с учетом всех составляющих ингредиентов;
* Суммарную прибыль от реализации произведенной продукции.

Система будет иметь возможность предоставлять следующие отчеты:

* Отчет об остатках продуктов на складе;
* Отчет о продажах по месяцам;
* Отчет по закупкам по годам.

## Структура базы данных

Учитывая потребности заказчика, были выявлены основные сущности базы данных:

* Изделие - содержит информацию о каждом виде изделия, производимом на данном заводе.

Поля:

* Наименование
* Вес
* Энергетическая ценность
* Срок годности
* Примечания
* Цена
* Склад\_Изделий – хранит данные о произведенных, но еще не реализованных изделиях;

Поля:

* Изделие
* Дата производства
* Количество (количество данных изделий, произведенных в один день)
* Производство – содержит информацию о производстве изделий;

Поля:

* Изделие
* Объем производства
* Дата окончания
* Стоимость производства
* Статус операции
* Ингредиент - содержит информацию о каждом виде ингредиентов, используемых в производстве на данном заводе;

Поля:

* Название
* Срок годности
* Энергетическая ценность
* Цена
* Единицы измерения
* Склад\_Ингредиентов – данные о закупленных, но не использованных ингредиентах;

Поля:

* Ингредиент
* Количество
* Дата поставки
* Состав – сведения о составе каждого изделия, с указанием ингредиентов и их количества, необходимых для производства;

Поля:

* Изделие
* Ингредиент
* Количество
* Продажи – сведения о продажах произведенной продукции;

Поля:

* Покупатель
* Изделие
* Количество
* Дата
* Статус операции
* Закупки – сведения о закупках ингредиентов хлебным заводом;

Поля:

* Ингредиент
* Поставщик
* Количество
* Дата поставки
* Статус операции
* Покупатель – справочник, хранящий наименования компаний-покупателей продукции хлебного завода;

Поля:

* Наименование
* Поставщики – справочник, хранящий наименования поставщиков ингредиентов для производства на хлебном заводе;

Поля:

* Наименование
* Статус\_Операции – справочник, содержащий значения статуса операции («В исполнении», «Выполнено», «Отменено»).

Поля:

* Значение

В соответствие с этим была построена структурная схема для проектируемой базы данных Системы (Рис. 1)

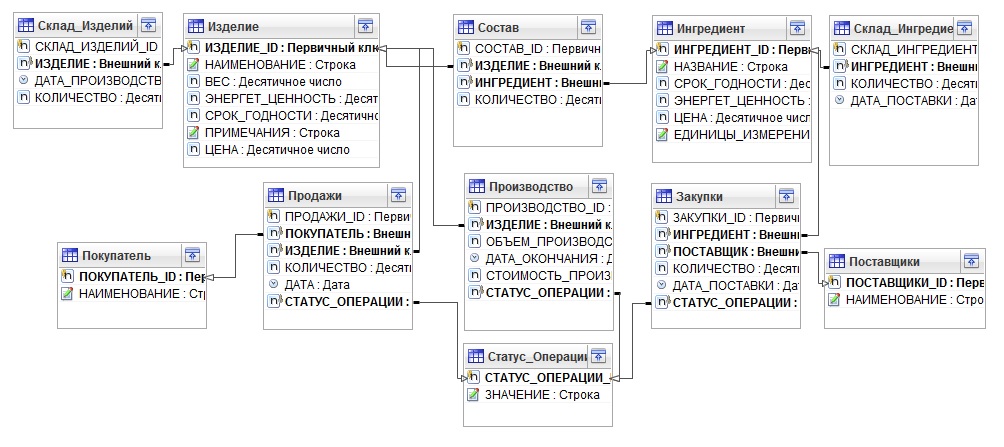


Рис. 1 - Структурная схема базы данных

1. Схема базы данных

## Выделение основных сущностей и проведение декомпозиции

Проанализировав выбранную предметную область курсового проекта в техническом задании были выявлены основные сущности базы данных.

## Описание функциональных зависимостей.

Отношения между таблицами определяются отношением между группами объектов соответствующего типа. Организация связи между таблицами называется связыванием. Различают следующие разновидности связи:

* отношение «один к одному»;
* отношение «один ко многим»;
* отношение «многие к одному»;
* отношение «многие ко многим».

Особую роль при разработке базы данных играет нормализация, целью которой является устранение недостатков структуры базы данных, приводящих к вредной избыточности в данных. Нормальная форма — требование, предъявляемое к структуре таблиц в теории реляционных баз данных для устранения из базы избыточных функциональных зависимостей между атрибутами (полями таблиц). Нормальных форм существует несколько:

1. Первая нормальная форма (1NF)

В данной форме все строки должны быть различными и все элементы внутри ячеек не должны образовать списки; Методами приведения к 1NF являются: устранение повторяющихся групп в отдельных таблицах (одинаковые строки); создание отдельной таблицы для каждого набора связанных данных; идентификация каждого набора связанных данных с помощью первичного ключа.

1. Вторая нормальная форма (2NF)

Таблица должна находиться в первой нормальной форме. Любое поле, не входящее в состав первичного ключа таблицы, функционально полно зависит от первичного ключа. Методами приведения ко 2NF являются: создание отдельных таблиц для наборов значений, относящихся к нескольким записям; комбинирование этих таблиц с помощью внешнего ключа.

1. Третья нормальная форма (3NF)

Таблица должна находится во второй нормальной форме. Любой её не ключевой атрибут функционально зависит только от первичного ключа. Методом приведения является удаление полей, не зависящих от ключа.

1. Нормальная форма Бойса-Кодда (BCNF)

Эта форма почти то же самое, что и третья. С одним дополнительным условием: таблица находится в третьей нормальной форме и в ней должен быть только один потенциальный первичный ключ. Другими словами, в таблице должен быть только один первичный ключ и не должно быть других потенциальных. Методом приведения к BCNF служит вынесение в отдельную таблицу потенциальных первичных ключей.

1. Четвертая нормальная форма (4NF)

Форма включает в себя требования всех предыдущих форм. В этой форме дополнительное правило должно исключать многозначные зависимости. Другими словами, все строки таблицы должны быть независимыми друг от друга.

Из всего вышеперечисленного можно сделать вывод о том, что наши таблицы удовлетворяют условиям 4NF, т.к. каждый детерминант является первичным ключом.

Теперь определим набор отношений разрабатываемой базы данных:

* Сущности Изделие и Склад изделий – отношение «многие к одному»
* Сущности Изделие и Состав – отношение «один ко многим»
* Сущности Изделие и Продажи – отношение «один ко многим»
* Сущности Изделие и Производство – отношение «один ко многим»
* Сущности Ингредиент и Состав – отношение «один ко многим»
* Сущности Ингредиент и Склад\_ингредиентов – отношение «многие к одному»
* Сущности Ингредиент и Закупки – отношение «один ко многим»
* Сущности Покупатель и Продажи – отношение «один ко многим»
* Сущности Статус\_операции и Продажи – отношение «один ко многим»
* Сущности Поставщики и Закупки – отношение «один ко многим»
* Сущности Статус\_операции и Закупки – отношение «один ко многим»
* Сущности Статус\_операции и Производство – отношение «один ко многим»

1. Описание алгоритмов функционирования приложения

## Описание инструмента реализации

Инструментом реализации является программная среда Platypus Application Platform 3.0, предназначенная для построения корпоративных приложений любого уровня сложности.

* 1. Создание структуры базы данных

Для создания базы данных необходимо выполнить следующие действия:

1. Создаем проект;
2. В настройках соединения с базой необходимо «Построить JDBC Url» и установить автоматическое формирование JDBC Url, после чего задать имя пользователя и пароль для входа;
3. Устанавливаем соединение проекта с базой данных (необходимо запустить «H2», пройдя путь вкладка «Окно» - «Службы»);
4. Теперь создаем диаграмму структуры базы данных как представлено на рисунке (Рис. 2);

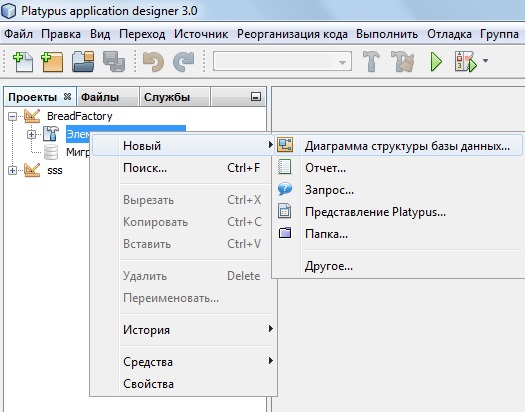


Рис. 2- Создание диаграммы структуры базы данных

1. Далее добавляем на диаграмму необходимые таблицы, при этом используя команду «создать таблицу» (Рис. 3)

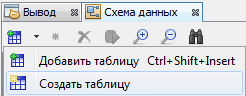


Рис. 3- Создание таблицы

Каждому полю таблицы присваиваем название и указываем тип строки.

Соединяем таблицы при помощи линий с указанием связей (от внешнего ключа к первичному).

* 1. Создание и реализация запросов

Для функционирования приложения необходимо работать с базой данных посредством запросов на языке SQL. С помощью них производится вывод данных на формах, а также редактирование и удаление информации из модели.

В системе были реализованы следующие запросы:

1. Все\_производство – запрашивает все записи из таблицы Производство (Рис. 4 и Рис. 5)

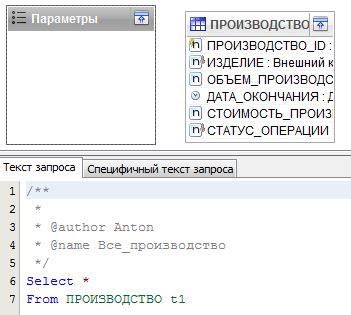


Рис. 4 - Запрос "Все\_производство"

C:\Users\Михаил\Desktop\скрины хлебокомб\запросы\34.JPG

Рис. 5- Результат запроса "Все\_производство"

1. Все\_закупки – запрашивает данные о всех закупках ингредиентов предприятием (Рис. 6 и Рис. 7)

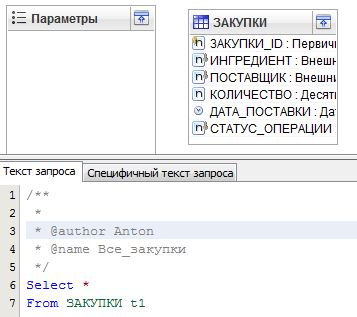


Рис. 6- Запрос "Все\_закупки"

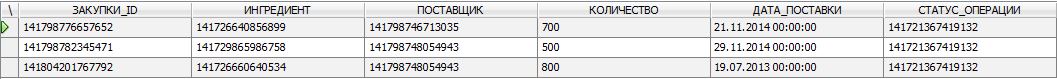


Рис. 7- Результат запроса "Все\_закупки"

1. Все\_продажи – запрашивает данные о всех продажах изделий, произведенных предприятием (Рис. 8 и Рис. 9)

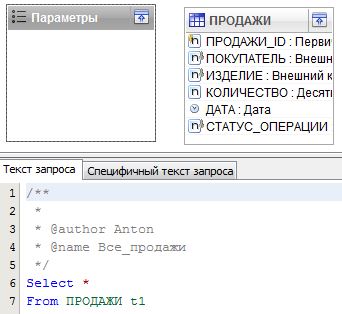


Рис. 8 - Запрос "Все\_продажи"

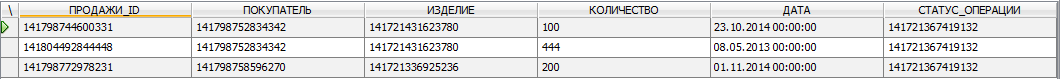


Рис. 9 - Результат запроса "Все\_продажи"

1. Весь\_состав – запрашивает все данные из таблицы Состав и позволяет работать с ними (Рис. 10 и Рис. 11)

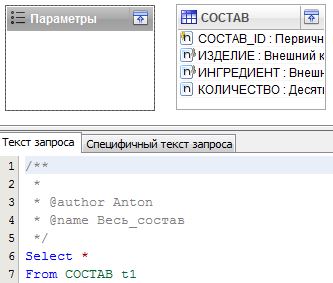


Рис. 10 - Запрос "Весь\_состав"

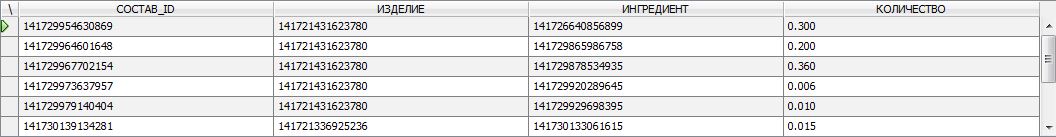


Рис. 11 - Результат запроса "Весь\_состав"

1. Весь\_склад\_ингредиентов – запрашивает всю информацию о складе ингредиентов из соответствующей таблицы (Рис. 12 и Рис. 13)

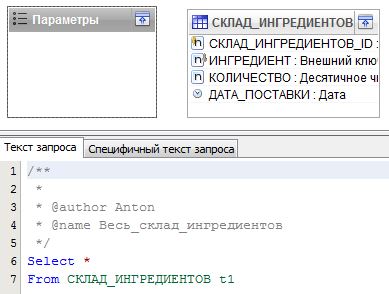


Рис. 12 - Запрос "Весь\_склад\_ингредиентов"

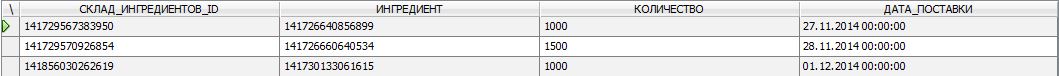


Рис. 13 - Результат запроса "Весь\_склад\_ингредиентов"

1. Весь\_склад\_изделий - запрашивает всю информацию о складе изделий из соответствующей таблицы (Рис. 14 и Рис. 15)

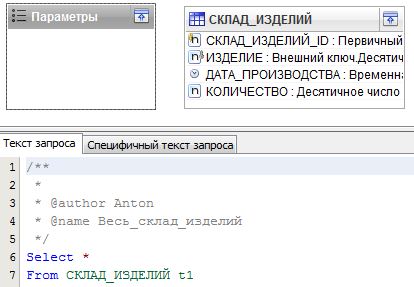


Рис. 14 - Запрос "Весь\_склад\_изделий"

C:\Users\Михаил\Desktop\скрины хлебокомб\10.JPG

Рис. 15 - Результат запроса "Весь\_склад\_изделий"

1. Каталог\_изделий – возвращает информацию обо всех изделиях, производимых хлебным комбинатом (Рис. 16 и Рис. 17)

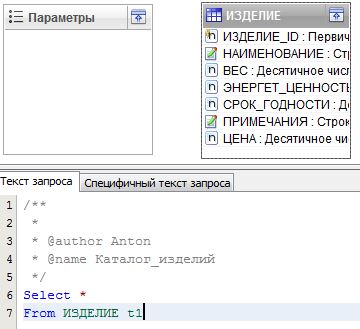


Рис. 16 - Запрос "Каталог\_изделий"

C:\Users\Михаил\Desktop\скрины хлебокомб\18.JPG

Рис. 17 - Результат запроса "Каталог\_изделий"

1. Каталог\_ингредиентов – возвращает информацию обо всех ингредиентах, закупаемых предприятием и используемых в производстве (Рис. 18 и Рис. 19)

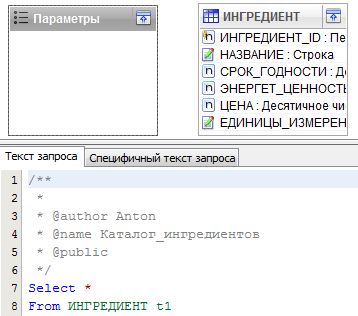


Рис. 18 - Запрос "Каталог\_ингредиентов"

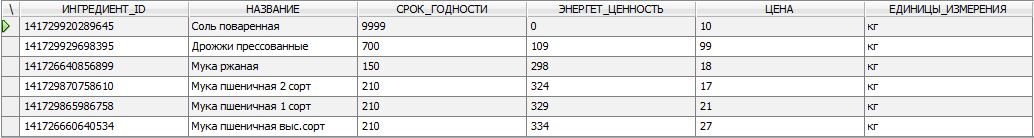


Рис. 19 - Результат запроса "Каталог\_ингредиентов"

1. Изделия\_склад – модифицированный запрос всех изделий на складе, который вместо идентификатора изделия возвращает его название и стоимость (Рис. 20 и Рис. 21)

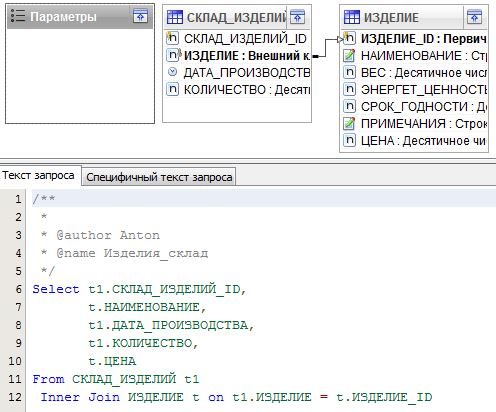


Рис. 20 - Запрос "Изделия\_склад"

C:\Users\Михаил\Desktop\скрины хлебокомб\12.JPG

Рис. 21 - Результат запроса "Изделия\_склад"

1. Ингредиенты\_склад – модифицированный запрос всех ингредиентов на складе, который вместо идентификатора ингредиента возвращает его наименование (Рис. 22 и Рис. 23)

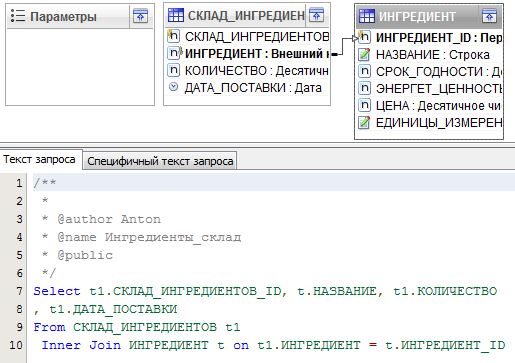


Рис. 22 - Запрос "Ингредиенты\_склад"

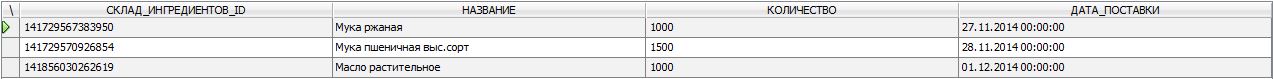


Рис. 23 - Результат запроса "Ингредиенты\_склад"

1. Состав\_изделия – запрос, который возвращает все записи из таблицы Состав, где фигурирует переданное в качестве входного параметра изделие (Рис. 24)

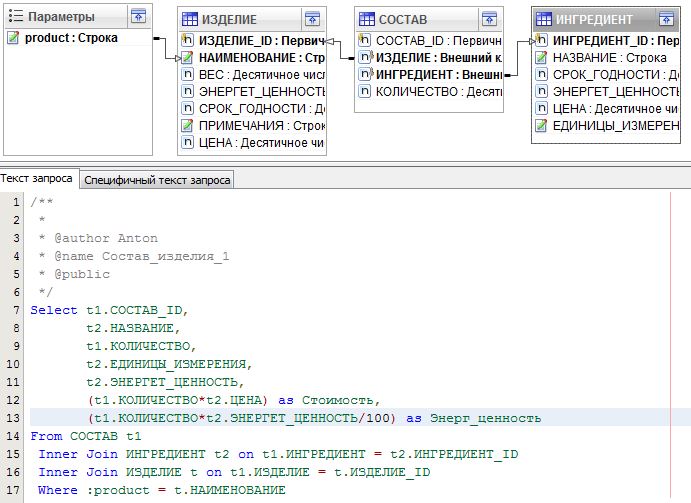


Рис. 24 - Запрос "Состав\_изделия"

При передаче параметра «Хлеб ржаной» получим следующий результат (Рис. 25)

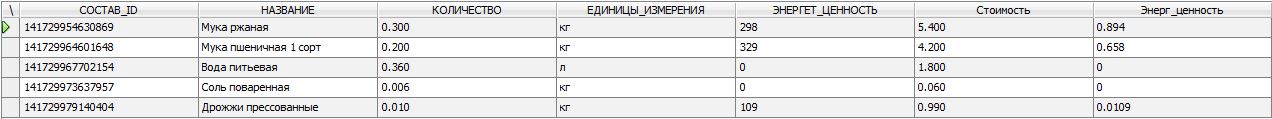


Рис. 25 - Результат запроса "Состав\_изделия" с параметром "Хлеб ржаной"

1. Архив\_продаж – возвращает все сведения о продажах предприятия, с указанием наименования заказчика, изделия и всех сопутствующих атрибутов (Рис. 26 и Рис. 27)

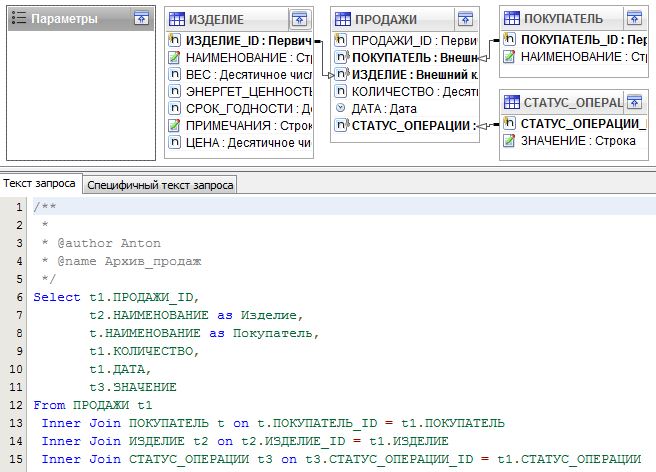


Рис. 26 - Запрос "Архив\_продаж"

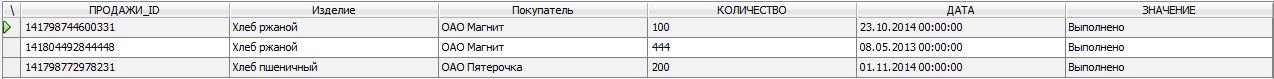


Рис. 27 - Результат запроса "Архив\_продаж"

1. Архив\_закупок – возвращает все сведения о закупках ингредиентов предприятием, включая наименование поставщика, ингредиента и всех сопутствующих атрибутов (Рис. 28 и Рис. 29)

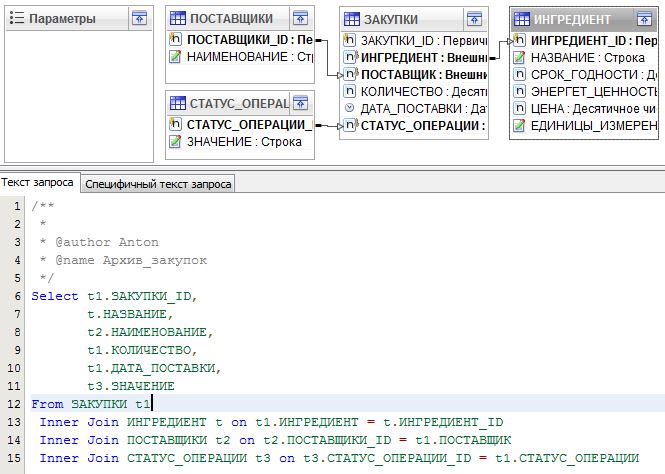


Рис. 28 - Запрос "Архив\_закупок"

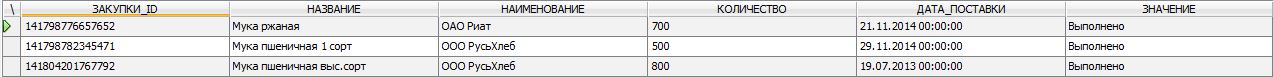


Рис. 29 - Результат запроса "Архив\_закупок"

1. Архив\_производства – возвращает сведения о текущем производстве, а также об уже произведенных изделиях и отмененных партиях (Рис. 30 и Рис. 31)

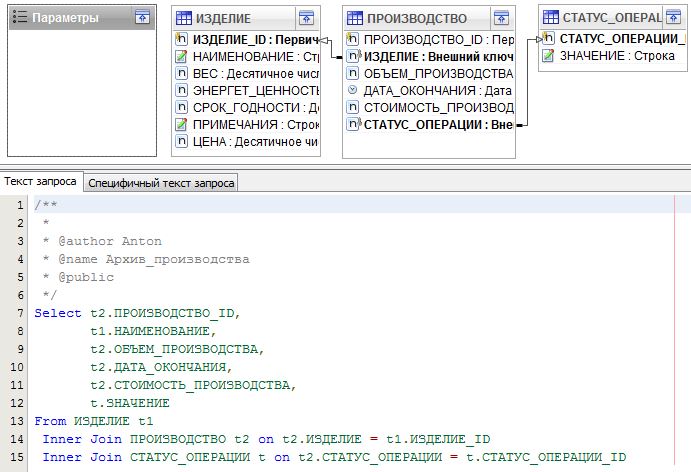


Рис. 30 - Запрос "Архив\_производства"

C:\Users\Михаил\Desktop\скрины хлебокомб\26.JPG

Рис. 31 - Результат запроса "Архив\_производства"

1. Названия\_изделий – запрос, возвращающий названия всех изделий. Используется при добавлении данных в модель, когда необходимо идентифицировать изделие (Рис. 32 и Рис. 33)

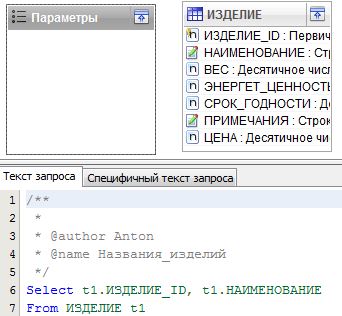


Рис. 32 - Запрос "Названия\_изделий"

C:\Users\Михаил\Desktop\скрины хлебокомб\28.JPG

Рис. 33 - Результат запроса "Названия\_изделий"

1. Названия\_ингредиентов - запрос, возвращающий названия всех ингредиентов. Используется при добавлении данных в модель, когда необходимо идентифицировать ингредиент (Рис. 34 и Рис. 35)

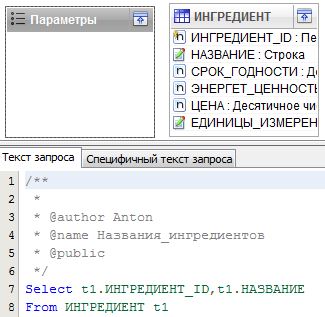


Рис. 34 - Запрос "Названия\_игредиентов"



Рис. 35 Результат запроса "Названия\_игредиентов"

1. Стоимость\_изделий\_на\_складе – возвращает количество и стоимость всех изделий, находящихся в данный момент на складе (Рис. 36 и Рис. 37)

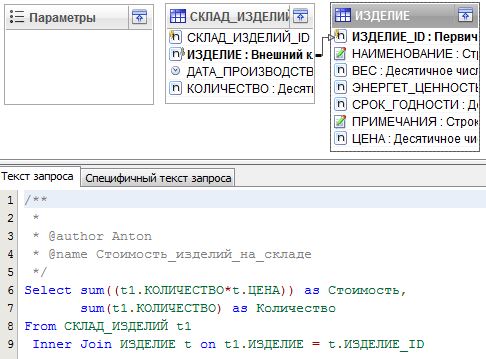


Рис. 36 - Запрос "Стоимость\_изделий\_на\_складе"

C:\Users\Михаил\Desktop\скрины хлебокомб\32.JPG

Рис. 37 - Результат запроса "Стоимость\_изделий\_на\_складе"

1. Описание интерфейсов пользователя

## Формы

Для взаимодействия программы с пользователем необходим пользовательский интерфейс. Человек, работающий с системой, не должен работать с низкоуровневыми запросами к базе данных. Всю информацию он может получить и редактировать путем открытия необходимого окна и нажатия соответствующей кнопки.

Рассмотрим формы, использующиеся в системе:

1. Форма **«Главная страница»**

Это стартовое окно, которое открывается при запуске программы. В нем пользователь может выбрать интересующий отдел и перейти в него, для дальнейшей работы (Рис. 38).

На форме представлены кнопки:

* Производство – переход в раздел производства фирмы
* Продажи – переход к архиву продаж
* Закупки – переход к архиву закупок
* Отчеты – переход к выбору отчета, который необходимо получить

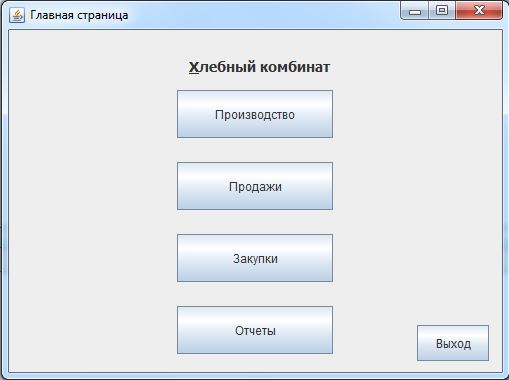


Рис. 38 - Форма "Главная страница"

Для открытия очередной формы необходимо прописать для кнопки обработчик нажатия.

Пример обработчика нажатия кнопки для открытия окна «Производство»:

function ProductionBtnActionPerformed (evt) {

close (new Производство().show());

}

После закрытия главного окна выполнится callback-функция, отображающая новое окно «Производство».

1. Форма **«Производство»**

[Главная – Производство]

В данном окне отображается и редактируется информация о производстве на предприятии, его изделиях и ингредиентах.

На форме представлены 3 вкладки:

* Производство
* Изделия
* Ингредиенты

На вкладке *Производство* (Рис. 39) представлена сетка данных, в которой отображаются все записи таблицы *Производство*. Можно добавлять и удалять записи при помощи соответствующих кнопок. При добавлении в поле *Изделие* сущности *Производство* вставляется текущее значение идентификатора изделия в левом нижнем углу формы. Аналогично происходит со статусом.

При удалении записи отображается диалог *Удалить «Название изделия»?* При положительном ответе пользователя текущая запись, у которой установлен курсор, будет удаляется.

Редактирование записей производится в сетке данных. Чтобы сохранить изменения, внесенные пользователем, необходимо нажать кнопку «Сохранить».

Для возвращения к главной странице программы необходимо нажать кнопку «На главную».

Добавление данных:

function addBtnActionPerformed(evt) {

все\_производство.insert();

все\_производство.ИЗДЕЛИЕ = названия\_изделий.ИЗДЕЛИЕ\_ID;

все\_производство.СТАТУС\_ОПЕРАЦИИ = СТАТУС\_ОПЕРАЦИИ.СТАТУС\_ОПЕРАЦИИ\_ID;

model.save();

архив\_производства.requery();

}

Удаление данных:

function deleteBtnActionPerformed(evt) {

if (confirm("Удалить " + архив\_производства.НАИМЕНОВАНИЕ + "?")) {

архив\_производства.deleteRow();

alert("Удалено!", model.save());

}

}

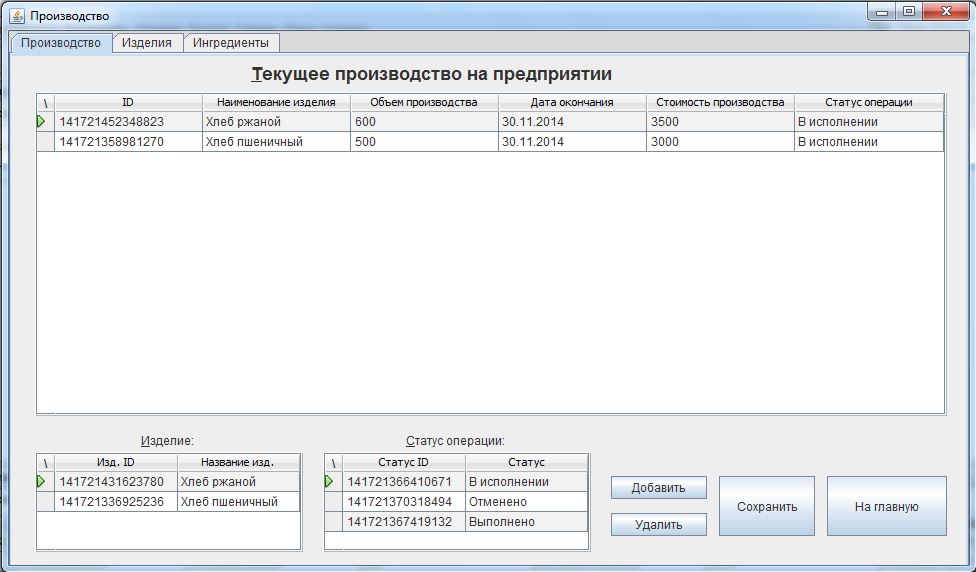


Рис. 39 - Форма "Производство". Вкладка Производство

На вкладке *Изделия* (Рис. 40) отображается каталог всей продукции предприятия. Аналогично предыдущей вкладке производится добавление, удаление и сохранение данных. Для перехода к складу изделий необходимо нажать на кнопку «Склад изделий». Также можно посмотреть состав выбранного изделия, кликнув по кнопке «Состав».

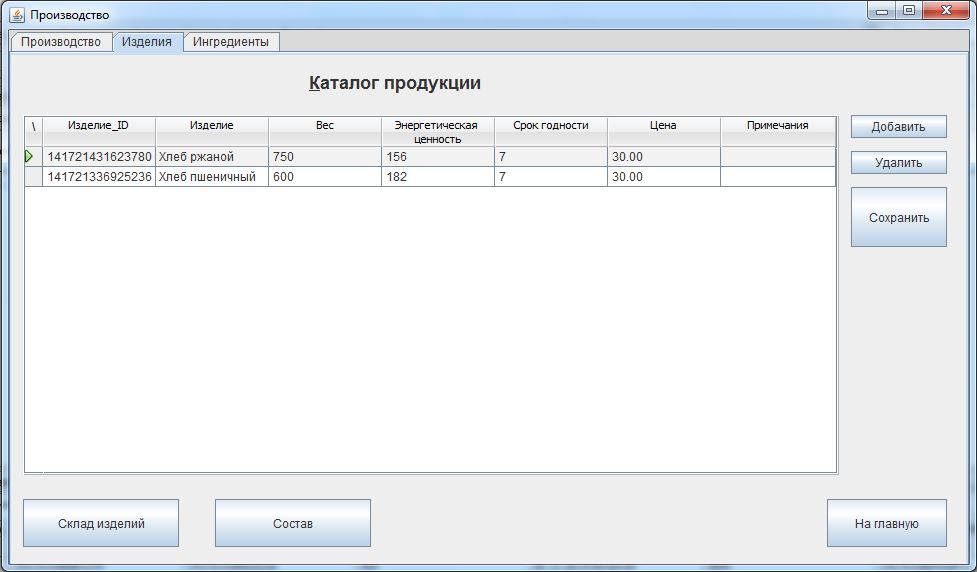


Рис. 40 - Форма "Производство". Вкладка Изделия

На вкладке *Ингредиенты* (Рис. 41)отображается каталог всех ингредиентов, используемых в производстве на предприятии. Можно просмотреть состояние склада ингредиентов, нажав на кнопку «Склад ингредиентов». Функции остальных кнопок аналогичны описанным выше.

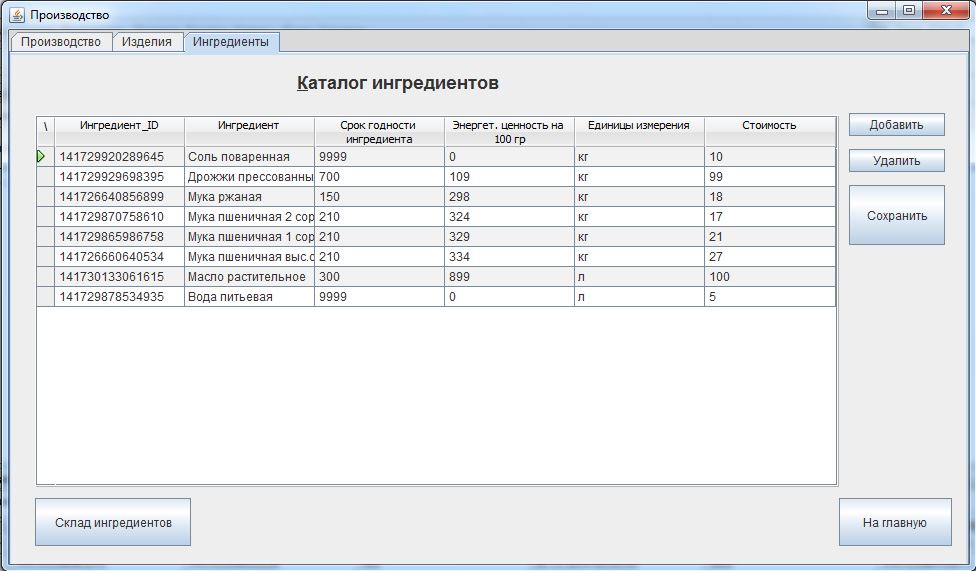


Рис. 41 - Форма "Производство". Вкладка Ингредиенты

1. Форма **«Склад\_изделий»**

[Главная – Производство (вкладка Изделия) – Склад\_изделий]

В данном окне отображается текущее состояние склада изделий. В верхней части формы приводится общее количество изделий на складе и их суммарная стоимость. Ниже, в сетке данных, находится информация обо всех изделиях, находящихся в данный момент на складе (Рис. 42).

Аналогично предыдущим формам реализовано добавление, удаление и сохранение данных. После каждого из этих действий происходит пересчет общего количества и суммы, и обновление данных значений.

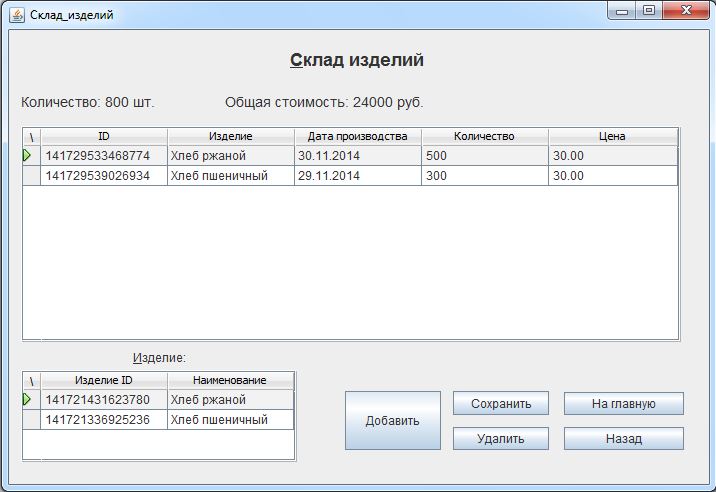


Рис. 42 - Форма "Склад\_изделий"

1. Форма **«Склад\_ингредиентов»**

[Главная – Производство (вкладка Ингредиенты) – Склад\_ингредиентов]

В данном окне отображается текущее состояние склада ингредиентов, используемых при производстве на предприятии (Рис. 43).

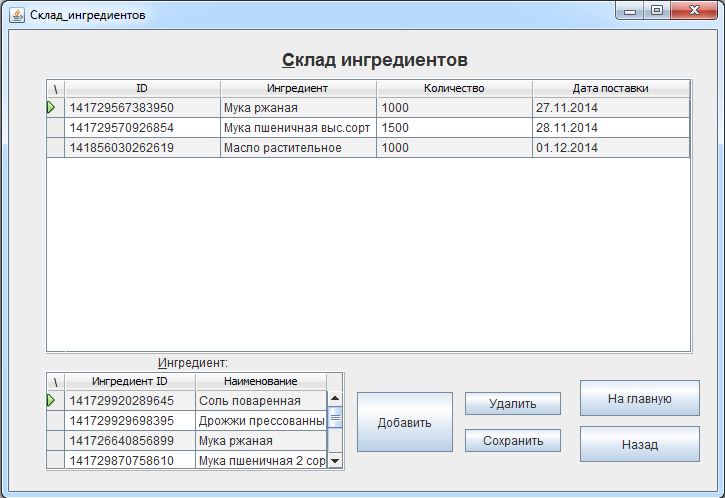


Рис. 43 - Форма "Склад\_инредиентов"

1. Форма **«Состав\_изделия»**

[Главная – Производство (вкладка Изделия) - Состав\_изделия]

В данном окне отображается состав выбранного изделия. При переходе в данное окно необходимо передать в качестве параметра название изделия. Автоматически выполниться запрос «Название\_изделия» и в сетку данных будут выведены соответствующие строки. Для добавления ингредиента в состав не обходимо нажать кнопку *Добавить*. Откроется окно Добавить\_в\_состав, в котором необходимо выбрать ингредиент и ввести его количество.

В нижней части формы рассчитывается общая стоимость и суммарная энергетическая ценность изделия. При сохранении и удалении происходит пересчет значений (Рис. 44).

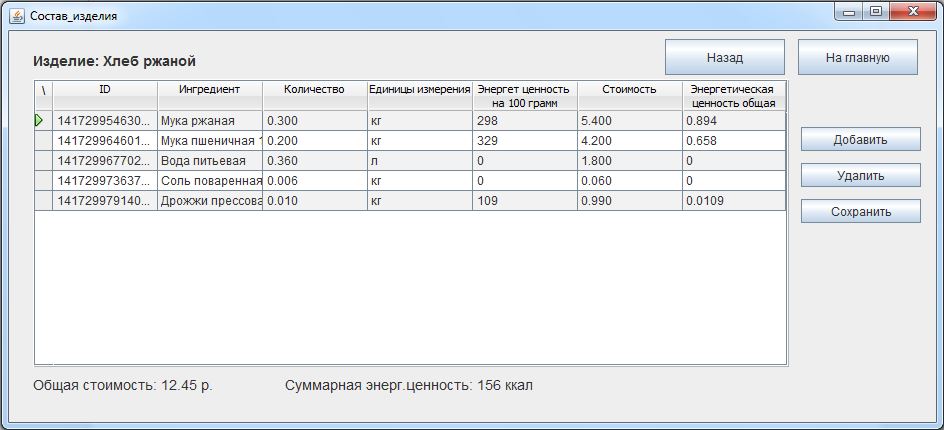


Рис. 44 - Форма "Состав\_изделия"

1. Форма **«Добавить\_в\_состав»**

[Главная – Производство (вкладка Изделия) - Состав\_изделия – Добавить\_в\_состав]

Данное окно является вспомогательным и служит для добавления ингредиента в состав изделия. В текстовом поле вводится количество добавляемого ингредиента, а в сетке данных выбирается сам ингредиент. При нажатии кнопки *Добавить* выполняется добавление ингредиента к данному изделию и открывается окно *Состав\_изделия* с измененным составом (Рис. 46).

Входной параметр данного окна: название изделия. При загрузке страницы происходит получение идентификатора данного товара:

function formWindowOpened(evt) {

goodsID = ИЗДЕЛИЕ.find(ИЗДЕЛИЕ.md.НАИМЕНОВАНИЕ, goodsName)[0].ИЗДЕЛИЕ\_ID;

label1.text += goodsName;

}

Запрос *Названия\_ингредиентов* используется для отображения в сетке данных названий для выбора пользователем.

Для добавления данных используется запрос *Весь\_состав*.

Модель данных изображена на Рис. 45.

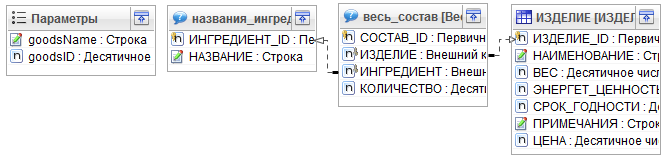


Рис. 45- Модель данных формы "Добавить\_в\_состав"

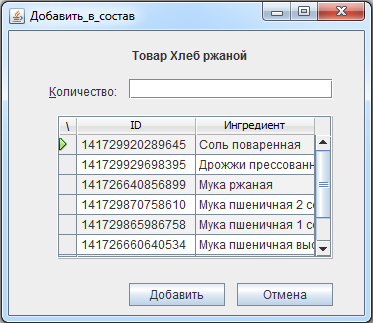


Рис. 46 - Форма "Добавить\_в\_состав"

1. Форма **«Окно\_продажи»**

[Главная – Окно\_продажи]

В данном окне отображается архив продаж предприятия с указанием заказчика, изделия, количества продукции, даты продажи и текущего статуса операции (Рис. 47).

Для добавления нового контракта в список продаж необходимо нажать кнопку *Добавить,* после чего появится вспомогательное окно, в котором можно быстро и удобно выбрать необходимые данные.

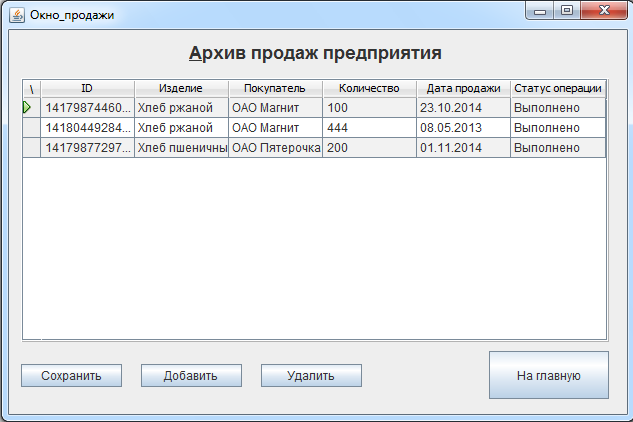


Рис. 47- Форма "Окно\_продажи"

1. Форма **«Добавить\_продажу»**

[Главная – Окно\_продажи – Добавить\_продажи]

Данная форма является вспомогательной и служит для добавления контрактов в архив продаж. Изделие, покупатель и статус операции выбираются из списков. Количество необходимо ввести в текстовое поле с соответствующей меткой. Дата продажи выбирается с помощью компонента modelDate связанным с параметром формы date (Рис. 48 и Рис. 49).

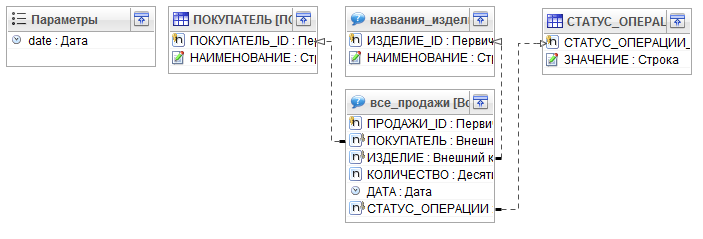


Рис. 48 - Модель данных формы "Добавить\_продажу"

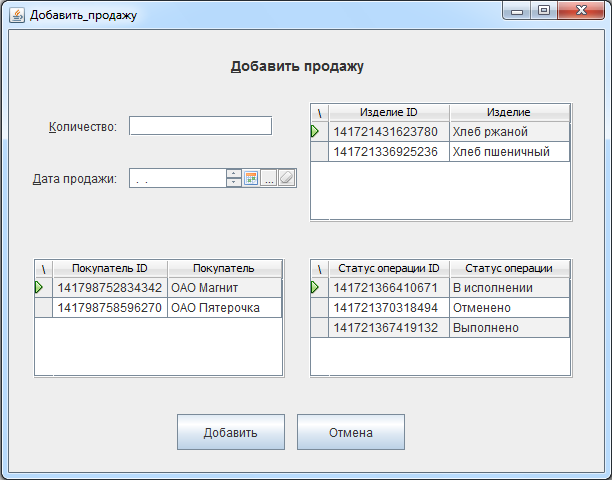


Рис. 49- Форма "Добавить\_продажу"

1. Форма «Окно\_закупки»

[Главная – Окно\_закупки]

В данном окне отображается архив всех закупок ингредиентов хлебным комбинатом, с указанием названия ингредиента, поставщика, количества, даты поставки и статуса операции (Рис. 50).

При добавлении, аналогично окну продаж, необходимо заполнить вспомогательную форму.

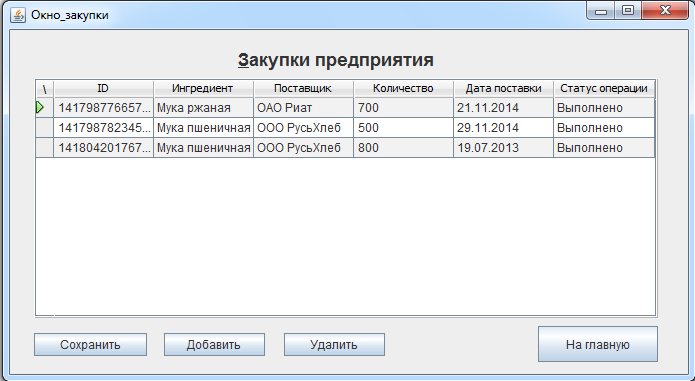


Рис. 50 - Форма "Окно\_закупки"

1. Форма **«Добавить\_закупку»**

[Главная – Окно\_закупки – Добавить\_закупку]

Вспомогательная форма для добавления контракта по закупке ингредиентов предприятием.

Модель данных формы представлена на Рис. 51.

В данном окне присутствуют 3 справочника: ингредиенты, поставщики и статус операции. Из них необходимо выбрать нужное значение. В текстовом поле вводится количество, а в modelDate – дата поставки (Рис. 52).

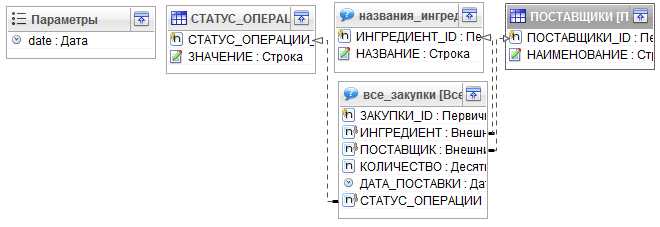


Рис. 51 - Модель данных формы "Добавить\_закупку"

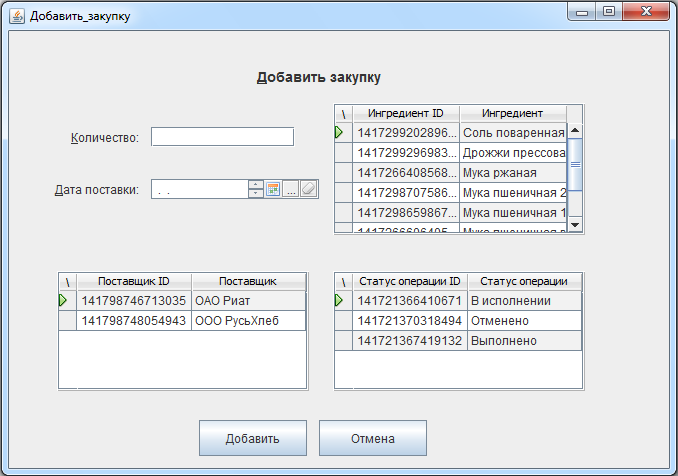


Рис. 52 - Форма "Добавить\_закупку"

1. Форма **«Окно\_отчеты»**

[Главная – Окно\_отчеты]

Данная форма предназначена для получения необходимого отчета в формате xls (Рис. 53).

Представлены 3 вида отчетов:

* Отчет о товарах на складе
* Отчет продаж по месяцам
* Отчет закупок по годам

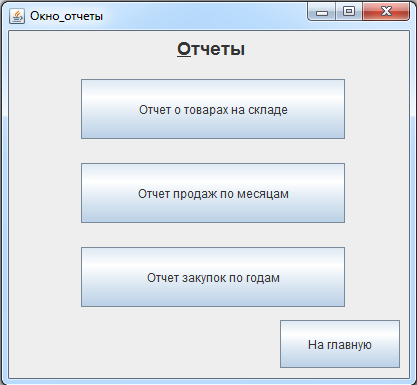


Рис. 53 - Форма "Отчеты"

## Отчеты

Для получения структурированной информации в формате электронной таблицы используются отчеты. Они позволяют вести бумажную документацию деятельности фирмы, группируя и сортируя результаты.

1. **Отчет о товарах на складе**

Отчет содержит в себе информацию о товарах, находящихся в данный момент на складе, такую как наименование изделия, количество, дата производства партии и цена за штуку. Также подсчитываются суммарное количество и суммарная стоимость всех изделий на складе (Рис. 54).



Рис. 54 - Пример отчета о товарах на складе

Привязка к данным модели и объектам JavaScript происходит с помощью тэгов, синтаксис которых соответствует синтаксису тэгов библиотеки jXLS(Рис. 55).

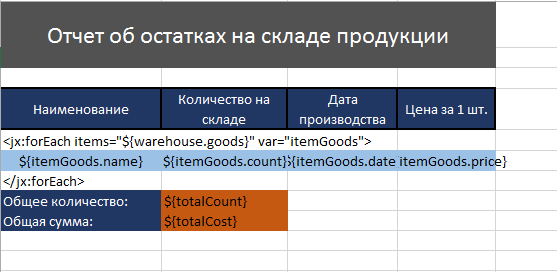


Рис. 55- Шаблон отчета об остатках на складе продукции

Перед формированием отчета создаются объекты с данными, выводимыми в помеченные тэгами места.

var warehouse = {};

warehouse.goods = [];

for (i = 0; i < изделия\_склад.length; i++)

{

var goods = {};

goods.count=изделия\_склад[i].КОЛИЧЕСТВО;

goods.name=изделия\_склад[i].НАИМЕНОВАНИЕ;

goods.date=изделия\_склад[i].ДАТА\_ПРОИЗВОДСТВА;

goods.price=изделия\_склад[i].ЦЕНА;

warehouse.goods.push(goods);

}

var totalCost = стоимость\_изделий\_на\_складе.Стоимость;

var totalCount = стоимость\_изделий\_на\_складе.Количество;

1. **Отчет о продажах по месяцам (текущего года)**

В данном отчете представлена информация о продажах предприятия в текущем году, сгруппированная и отсортированная по месяцам (Рис. 56).



Рис. 56 - Пример отчета о продажах по месяцам

Номер месяца получается при помощи функции getMonth(). По данным номерам происходит сортировка результатов.

Для вывода названия месяца используется следующая функция:

function getMonthName(monthNum) {

var months = ['январь','февраль','март','апрель','май','июнь','июль','август','сентябрь','октябрь','ноябрь','декабрь'] ;

return months[monthNum];

}

MonthNum – номер месяца, полученный функцией getMonth().

Шаблон данного отчета представлен на Рис. 57.

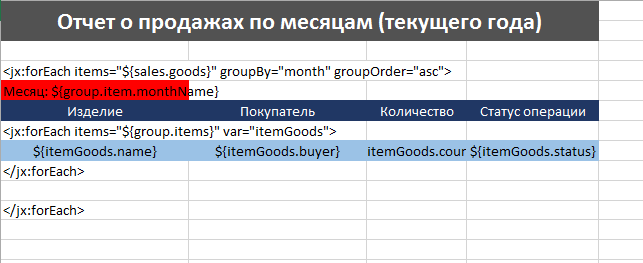


Рис. 57 - Шаблон отчета о продажах по месяцам

Данные для заполнения формируются аналогичным образом.

1. **Отчет закупок по годам**

В данном отчете содержатся все закупки ингредиентов предприятием, сгруппированные и отсортированные по годам (Рис. 58).



Рис. 58 - Пример отчета о закупках по годам

В шаблоне отчета помимо группировки по году происходит и сортировка по возрастанию (Рис. 59).

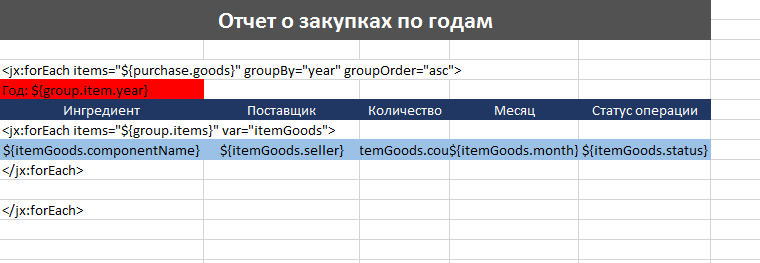


Рис. 59 - Шаблон отчета о закупках по годам

# **Заключение**

В ходе выполнения курсовой работы была достигнута цель – спроектирована и разработана программная система для хлебного комбината». Созданная база данных готова к использованию.

В результате выполнения данной работы:

* Была спроектирована база данных для данной системы;
* Был создан проект программной системы;
* Была разработана и реализована система с понятым интерфейсом и удовлетворяющая функциональным требованиям заказчика.

Созданное приложение позволяет собрать все данные в едином пространстве и делает их доступным для сотрудников в удобной форме. Также данный продукт дает возможность вести структурированный учет всей информации и предоставляет оперативный доступ к ней. Функционал системы позволяет автоматизировать стандартные процессы документирования. Результатом ввода в эксплуатацию данной системы будет оптимизация работы предприятия.

Отчет о проделанной работе выполнен в соответствии с ГОСТами и требованиями, предъявляемыми к технической документации.

# Список литературы

1. *Platypus Platform.Компания Альтернатива-Софт.* (б.д.). Получено 3 Декабря 2014 г., из Сайт компании Альтернатива-Софт: http://altsoft.biz/platypus/
2. Аблов, И. В. (26 Август 2012 г.). *Образование.Сайт myshared.ru.* Получено 3 Декабрь 2014 г., из сайт myshared.ru: http://www.myshared.ru/slide/56479/