



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ»

ДИСЦИПЛИНА «БАЗЫ ДАННЫХ»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ
НА ТЕМУ:

База данных "Снабжение"

Студент ИУ6-42

Бурлаков А.С.

(Группа)

(Подпись, дата)

(И.О.Фамилия)

Руководитель курсового проекта

Скворцова М.А.

(Подпись, дата)

(И.О.Фамилия)

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э.
Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ6

А.В. Пролетарский

« ____ » _____ 2017 г.

З А Д А Н И Е
на выполнение курсовой работы

по дисциплине **Базы данных**

Студент группы ИУ6-42

Бурлаков Александр Сергеевич
(Фамилия, имя, отчество)

Тема курсовой работы

База Данных «Снабжение»

Направленность КР учебная

Источник тематики кафедра ИУ6

График выполнения КР: 25% к 3 нед., 50% к 7 нед., 75% к 10 нед., 100% к 14 нед.

Техническое задание Разработать базу данных “Снабжение”, состоящую минимум из 7 таблиц, имеющую сложные запросы и отчеты. Разработать удобный и понятный пользовательский интерфейс. В качестве СУБД для разработки использовать Oracle Database Express Edition 11g, Oracle Database SQL Developer. Для проектирования схемы данных использовать Oracle SQL Developer Data Modeler. Для проектирования пользовательского интерфейса необходимо использовать Oracle APEX версии 5.1.

Оформление курсовой работы:

Расчетно-пояснительная записка на не менее 25 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания «05» февраля 2018 г.

Руководитель курсовой работы

Студент

(Подпись, дата)

(Подпись, дата)

М.А. Скворцова
(И.О.Фамилия)

А.С. Бурлаков
(И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

РЕФЕРАТ

Расчетно-пояснительная записка 27 с., 9 рис., 10 табл., 4 источника

Объектом разработки является база данных для снабжения склада и пользовательский интерфейс для неё.

Цель работы — создание удобного и доступного интерфейса для базы данных снабжения склада.

Поставленная цель достигается за счет средств Oracle APEX версии 5.1 и Oracle SQL Developer Data Modeler. В качестве СУБД для разработки используется Oracle Database Express Edition 11g, Oracle Database SQL Developer. Разрабатываемая база данных содержит 13 таблиц, взаимодействующих между собой при помощи связей. Пользовательский интерфейс реализует взаимодействие сотрудников склада, поставщиков и заказчиков за счет формирования заказов, просмотра информации по предыдущим заказам и формированию накладных

Ключевые слова - Oracle APEX, база данных, СУБД, Oracle SQL Developer Data Modeler, таблицы, триггеры, записи, поля.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1 Анализ и проектирование базы данных.....	7
1.1 Исследование предметной области.....	7
1.2 Бизнес процессы пользователей информационной системы	7
1.3 Создание модели в Oracle SQL Developer Data Modeler.....	8
2 Формирование базы данных	12
2.1 Создание пользователя для базы данных	12
2.2 Перенос базы данных в Oracle Database SQL Developer.....	12
2.3 Добавление данных в таблицы	12
3 Создание интерфейса в среде Oracle APEX версии 5.1	14
3.1 Создание страницы авторизации.....	14
3.2 Создание страницы Администратора	15
3.3 Страница ввода.....	16
ЗАКЛЮЧЕНИЕ	18
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	19
Приложение А	20
Приложение Б.....	24
Приложение В	26

ВВЕДЕНИЕ

В данной работе производится создание базы данных для снабжения склада и пользовательского интерфейса для неё. Она предназначена для формирования заказов. Интерфейс предоставляет возможность работнику склада просматривать информацию о заказе, создавать заказы как для заказчиков, так и для дозакупки товаров, удалять и изменять заказы.

Разработка такой базы данных является актуальной, поскольку даёт возможность администрации склада контролировать клиентскую базу и грамотно распределять рабочие ресурсы. Данная база данных упрощает работу, выполняемую работниками склада.

Интерфейс разрабатывался в соответствии со следующими требованиями:

- наличие пользователя-администратора и пользователя-работника;
- простота и понятность обычному пользователю;
- возможность просмотра списка всех пользователей администратором;
- лаконичность интерфейса.

На данный момент существует множество аналогов такой базы данных. Можно найти множество различных интерфейсов для учета товаров на складе в интернете.

При разработке интерфейса для базы данных "Снабжение" учтены некоторые недостатки имеющихся аналогов, что позволило повысить универсальность разрабатываемого продукта.

1 Анализ и проектирование базы данных

1.1 Исследование предметной области

Предметной областью разрабатываемого программного продукта является формирование заказа для клиентов и формирование заказа для склада.

База данных «Снабжение» предназначена для упрощения обработки информации о клиентах, которые заказывают товары со склада.

Эта база данных предназначена для менеджеров (работников склада), которые вносят информацию в базу, регистрируя и работая с клиентами. Но также может использоваться людьми, имеющими непосредственное отношение к работе БД, такими как вышестоящее руководство.

База данных выполняет следующие функции:

1. Хранение информации о сотрудниках
2. Хранение информации о клиентах
3. Хранение информации о поставщиках
4. Хранение информации о заказах

Данные требования к разрабатываемой базе данных учтены при создании модели, а также доработаны в соответствии с возникающими проблемами при проектировании.

После рассмотрения предметной области было решено создать следующие страницы таблиц:

- а) «Сеть магазинов», в которой должна содержаться информация о торговых сетях («Пятёрочка»);
- б) «Магазин», в которой должна содержаться информация об адресе и названии магазина, а так же об принадлежности его сети магазинов;
- в) «Заказ», в которой должна находиться информация о складе, с которого заказывают товар, текущем статусе заказа;
- г) «Заказ склада», в которой должна находиться информация о складе, на который заказывают товар, текущем статусе этого заказа;
- д) «Продукт», в которой должен содержаться полный перечень названий и цен продуктов;

- е) «Продукт на складе», в которой должна находиться информация о количестве продукта, находящемся на определенном складе.
- ж) «Продукт заказа», в которой должна находиться информация о количестве продуктов определенного заказа;
- з) «Продукт заказа склада», в которой должна находиться информация о количестве продуктов определенного заказа склада;
- и) «Пользователь», в которой должна находиться информация о логине, пароле, роли и ФИО пользователя;
- к) «Склад», в которой должна содержаться информация о названии и балансе склада

Схема связи таблиц представлена на рисунке 1.

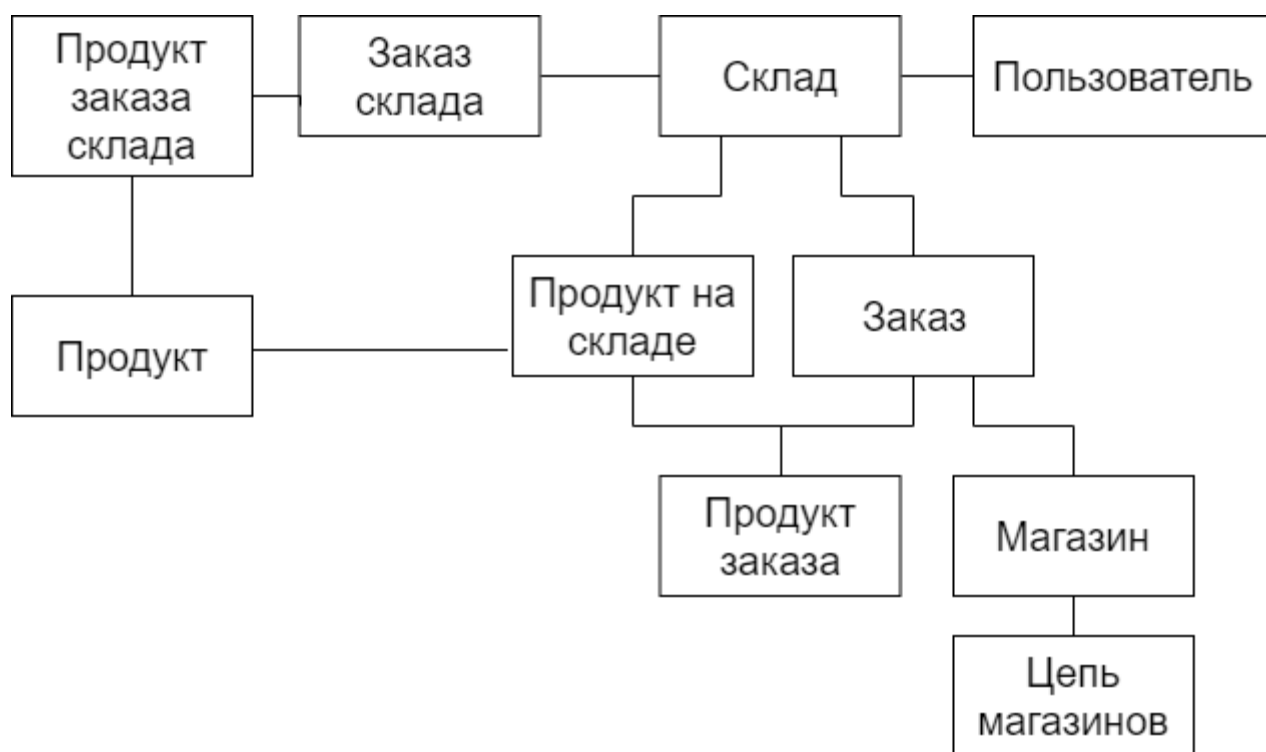


Рисунок 1 - Схема связи таблиц

1.2 Создание модели в Oracle SQL Developer Data Modeler

Для хранения баланса на складе, а так же учета сотрудников склада создана таблица «Склад». Она содержит информацию о балансе склада,

администраторе склада, полученным из таблицы «Пользователь» и его названии.

Так как необходимо учитывать работников склада создана таблица «Пользователь». Она содержит логин и пароль пользователя, с помощью которых он будет заходить в систему. Помимо этого таблица содержит данные о пользователе – его ФИО и текущий статус (работник склада «Р» или администратор «А»).

Основное назначение базы данных – формирование заказов, поэтому создана таблица «Заказ». В таблице содержится информация о магазине, информация о котором получена из таблицы «Магазин», складе, с которого заказ будет приниматься, текущий статус заказа (Закрыт «А», Открыт «У»). Помимо этого учитывается дата создания и закрытия заказа.

Для заказа так же необходимо учитывать продукты, которые в него входят. Для этого создана таблица «Продукт заказа», в которой содержится информация о продукте со склада, полученным из таблицы «Продукт склада», количестве заказанных продуктов, а так же номеру заказа, полученному из таблицы «Заказ», который необходим для реализации связи один-много.

Из предыдущей таблицы появляется необходимость создания следующей: таблицы «Продукт склада», которая содержит информацию о названии продукта, его количестве на складе, цене, а так же идентификатор склада, о продукте которого ведется описание. Идентификатор берется из таблицы «Склад».

Так как заказчиков у склада может быть много, решено создать таблицу «Цепь магазинов» которая объединяет некоторые магазины. В этой таблице хранится информация об названии цепи, например «Пятерочка».

Для учета заказчиков создана таблица «Магазин», в которой хранится информация о цепи магазинов, информация о которых берется из таблицы «Цепь магазинов», к которой магазин принадлежит, а так же его название и адрес.

Все следующие таблицы созданы для хранения информации о поставках на склад.

Таблица «Заказ склада» имеет информацию о складе, информация о котором берется из таблицы «Склад», на который завозят товары, текущем статусе заказа (Закрит «А» и Открыт «У»), дате создания и закрытия заказа.

Таблица «Продукт заказа склада» содержит информацию о заказываемом продукте, информация о котором берется из таблицы «Продукт», количестве этого продукта, а так же информация о заказе, которая берется из таблицы «Заказ склада»

Таблица «Продукт» содержит информацию о названии и цене продукта.

Модель разработанной базы данных "Снабжение" представлена на рисунке 5

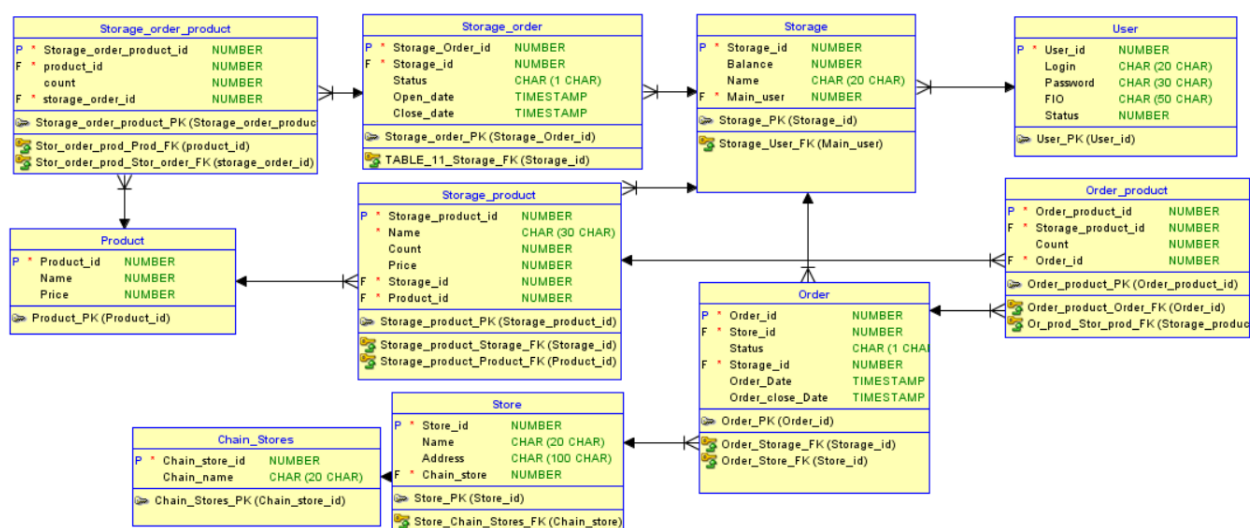


Рисунок 2 - Модель разработанной базы данных

Таблицей называется элемент, который представлен на рисунке 6.

1
2
3

Рисунок 3 - Таблица

В данной таблице присутствуют следующие обозначения:

- 1 Название таблицы;
- 2 Описание столбцов таблицы (название и тип данных);

3 Описание ключей, связывающих данную таблицу с другими таблицами.

Основными таблицами являются таблицы “Storage_order” и “Order”, так как основное назначение базы данных – создание заказов и дозакупка товаров.

Таблица “Product” содержит информацию о товарах. Эта таблица состоит из следующих полей:

- а) Product_id – первичный ключ;
- б) Name – название товара;
- в) Price – цена товара.

Таблица “Storage_product” содержит информацию о товарах склада. Эта таблица состоит из следующих полей:

- а) Storage_product_id – первичный ключ;
- б) Name – название товара (так как возможно его переименовали);
- в) Count – количество такого товара на складе;
- г) Price – цена товара;
- д) Storage_id – склад. Вторичный ключ, содержащий первичный ключ таблицы “Storage”;
- е) Product_id – товар. Вторичный ключ, содержащий первичный ключ таблицы “Product”.

Таблица “Storage_order_product” содержит информацию о товарах заказа товаров допоставки на склад. Эта таблица состоит из следующих полей:

- а) Storage_order_product_id – первичный ключ;
- б) product_id – продукт, вторичный ключ, содержащий первичный ключ таблицы “Product”;
- в) count – количество товара в заказе;
- г) storage_order_id – заказ допоставки, вторичный ключ, содержащий первичный ключ таблицы “Storage_order”.

Таблица “Storage_order” содержит информацию о заказах допоставки на склад. Эта таблица состоит из следующих полей:

- а) Storage_order_id – первичный ключ;

- б) Storage_id – склад, вторичный ключ, содержащий первичный ключ таблицы “Storage”;
- в) Status – текущий статус заказа. Хранит значение ‘А’, если заказ закрыт и значение ‘У’, если заказ активен;
- г) Open_date – дата открытия заказа;
- д) Close_date – дата закрытия заказа.

Таблица “Storage” содержит информацию о складах. Эта таблица состоит из следующих полей:

- а) Storage_id – первичный ключ;
- б) Balance – текущий баланс склада;
- в) Name – название склада;
- г) Main_user – пользователь-администратор, вторичный ключ, содержащий первичный ключ таблицы “User”.

Таблица “User” содержит информацию о пользователях программы. Эта таблица состоит из следующих полей:

- а) User_id – первичный ключ;
- б) Login – никнейм пользователя;
- в) Password – пароль для входа в систему;
- г) FIO – фамилия, имя, отчество пользователя;
- д) Status – роль, данная пользователю. Хранит значение ‘А’, если пользователь является администратором и значение ‘Р’, если пользователь не имеет таких прав.

Таблица “Chain_Stores” содержит информацию о торговых цепях. Эта таблица состоит из следующих полей:

- а) Chain_store_id – первичный ключ;
- б) Chain_name – название торговой цепи.

Таблица “Store” содержит информацию о магазинах. Эта таблица состоит из следующих полей:

- а) Store_id – первичный ключ;
- б) Name – название магазина;
- в) Address – адрес магазина;
- г) Chain_store – торговая цепь, в которую входит магазин, вторичный ключ, содержащий первичный ключ таблицы “Chain_Stores”.

Таблица “Order” содержит информацию о заказах склада. Эта таблица состоит из следующих полей:

- а) Order_id – первичный ключ;
- б) Store_id – магазин заказчик, вторичный ключ, содержащий первичный ключ таблицы “Store”;
- в) Status – текущий статус заказа. Хранит значение ‘A’, если заказ закрыт и ‘Y’, если заказ активен.
- г) Storage_id – склад, вторичный ключ, содержащий первичный ключ таблицы “Storage”.
- д) Order_Date – дата создания заказа;
- е) Order_close_date – дата закрытия заказа.

Таблица “Order_product” содержит информацию о товарах заказа склада. Эта таблица состоит из следующих полей:

- а) Order_product_id – первичный ключ;
- б) Storage_product_id – продукт склада, вторичный ключ, содержащий первичный ключ таблицы “Storage_product”;
- в) Count – количество заказанного товара;
- г) Order_id – заказ, вторичный ключ, содержащий первичный ключ таблицы “Order”.

1.3 Бизнес процессы пользователей информационной системы

В данной системе предусмотрены процессы создания заказа, а так же отметка о том, что заказ отдан. Следовательно, можно выделить два бизнес процесса:

- а) Создание заказа;
- б) Отметка об отдаче заказа;

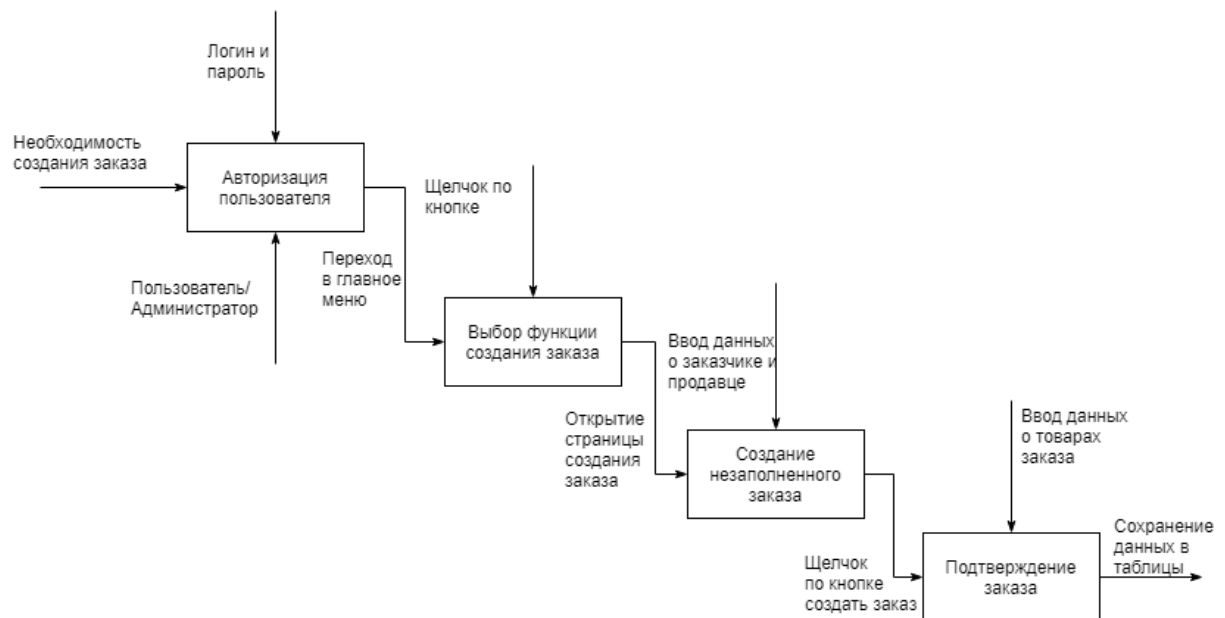


Рисунок 4 – Бизнес-процесс создания заказа

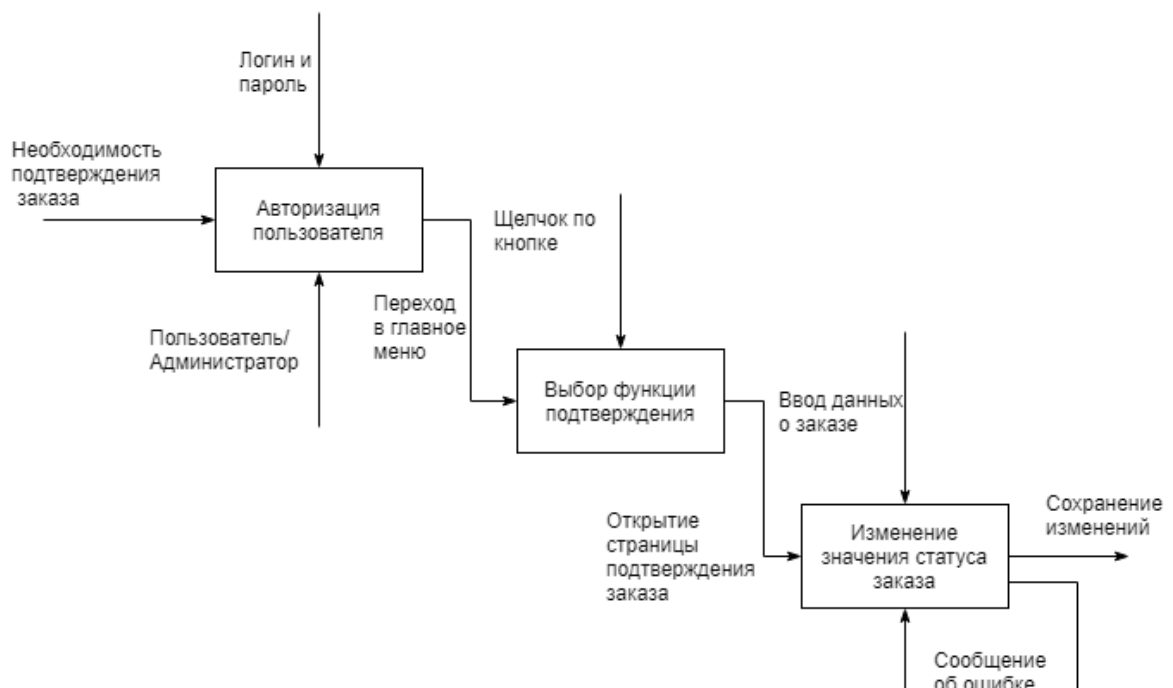


Рисунок 5 – Бизнес-процесс подтверждения заказа

2 Формирование базы данных

2.1 Создание пользователя для базы данных

Создание пользователя происходило с помощью графического интерфейса SQL Developer, т.к. данный способ создания позволяет видеть все права, которыми может обладать пользователь, и выбрать их из списка.

Принято решение наделить пользователя следующими правами доступа:

- возможность создания, изменения и удаления таблиц;
- возможность создания, изменения и удаления процедур;
- возможность создания, изменения и удаления триггеров;
- возможность создания представлений;
- возможность создания, изменения и удаления счетчиков;
- возможность создания сессий;
- возможность изменения пользователя.

2.2 Перенос базы данных в Oracle Database SQL Developer

Формирование базы данных началось с переноса схемы базы данных в Oracle Database SQL Developer. Для этого было необходимо получить схему базы данных в виде SQL кода и выполнить его в схеме пользователя, созданного предварительно в Oracle Database Express Edition 11g.

2.3 Добавление данных в таблицы

Было решено первыми заполнить таблицы, которые не зависят от данных в других таблицах. Сначала были заполнены таблицы «Product», «User », «Chain_stores», «store», «Storage».

Для добавления данных в таблицы было решено использовать SQL.[6] В схеме пользователя был написан и отлажен оператор INSERT, который создаёт записи в таблице. В листинге 1 представлено добавление записи в таблицу "STORAGE_PRODUCT" при помощи оператора INSERT:

Листинг 1 - Добавление записи в таблицу «*STORAGE_PRODUCT*»

```
INSERT INTO STORAGE_PRODUCT (NAME, PRICE, COUNT, STORAGE_ID)  
VALUES ('Продукт 1', 35, 10, 1);
```

Команда INSERT за один вызов заносит в таблицу лишь одну строку, поэтому, чтобы заполнить нашу таблицу нам понадобится вызвать команду столько раз, сколько строк мы планируем заполнить.

Для формирования идентификаторов у каждой таблицы необходимо создать последовательности. Для каждой таблицы были созданы последовательности, номер которых начинался с 0.

Листинг 2 – Добавление последовательности в таблицу order

```
CREATE SEQUENCE order_SEQ START WITH 0 INCREMENT BY 1 MINVALUE 0;
```

Для автоматического заполнения идентификаторов таблиц были созданы триггеры, использующие последовательности для заполнения номеров.

Листинг 3 – Добавление триггера для таблицы order

```
CREATE OR REPLACE TRIGGER order_TRG BEFORE  
INSERT ON "Order" FOR EACH ROW  
BEGIN  
:NEW.order_id := order_SEQ.NEXTVAL;  
END;
```

3 Создание интерфейса в среде Oracle APEX версии 5.1

3.1 Создание страницы авторизации

Было решено создать страницу авторизации, на которой пользователю предлагается ввести свои логин и пароль.

Для осуществления авторизации используется таблица “User”.

Для осуществления авторизации нового пользователя был создан пакет Pkg_Security.

Листинг 4 – код создания пакета Pkg_Security

```
create or replace Package Pkg_Security Is
--- Заголовок авторизации
Function Authenticate_User(p_Login Char
                        ,p_Password Char) Return Boolean;

-----
Procedure Process_Login(p_Login Char
                        ,p_Password Char
                        ,p_App_Id Number);

End Pkg_Security;
create or replace Package Body Pkg_Security Is
--- Авторизация
Function Authenticate_User(p_Login Char
                        ,p_Password Char) Return Boolean As
v_Password "User".Password%Type;
v_Status "User".Status%Type;
Begin
If p_Login Is Null Or p_Password Is Null Then

    -- Write to Session, Notification must enter a username and password
    Apex_Util.Set_Session_State('LOGIN_MESSAGE'
                                , 'Please enter Username and password. ');
    Return False;
End If;
-----
Begin
Select u.Password,
       u.Status
Into v_Password
    ,v_Status
From "User" u
Where u.Login = p_Login;
Exception
When No_Data_Found Then

    -- Write to Session, User not found.
    Apex_Util.Set_Session_State('LOGIN_MESSAGE'
```



```

        , 'User not found');
    Return False;
End;
If v_Password <> p_Password Then

    -- Write to Session, Password incorrect.
    Apex_Util.Set_Session_State('LOGIN_MESSAGE'
        , 'Password incorrect');

    Return False;
End If;
---
-- Write user information to Session.
--
Apex_Util.Set_Session_State('SESSION_USER_NAME'
    , p_Login);
Apex_Util.Set_Session_State('SESSION_USER_STATUS'
    , v_Status);
---
---
Return True;
End;

-----
Procedure Process_Login(p_Login char
    , p_Password char
    , p_App_Id Number) As
    v_Result Boolean := False;
Begin
    v_Result := Authenticate_User(p_Login
        , p_Password);
    If v_Result = True Then
        -- Redirect to Page 1 (Home Page).
        Wwv_Flow_Custom_Auth_Std.Post_Login(p_Login -- p_User_Name
            , p_Password -- p_Password
            , v('APP_SESSION') -- p_Session_Id
            , p_App_Id || ':1' -- p_Flow_page
            );
    Else
        -- Login Failure, redirect to page 101 (Login Page).
        Owa_Util.Redirect_Url('f?p=112:101:');
    End If;
End;

End Pkg_Security;

```

Спецификация описывает прототип функции `Authenticate_User`, результатом выполнения которого является логическое значение, зависящее от наличия пользователя в таблице «USER_LOGIN» (возвращает TRUE, если пользователь найден, FALSE - в противном случае), а так же передача в

текущую сессию. Аргумент значения логина текущего пользователя и его роль (Администратор или пользователь).

В теле функции осуществляется сравнение введенного имени пользователя в поле «Login» с полем «p_Login» и введенного пароля в поле «Password» с полем «p_Password».

Спецификация также описывает прототип функции Process_Login, которая производит переход на домашнюю страницу приложения и вызывает предыдущую функцию.

Для авторизации необходимо было создать страницу, в которой присутствуют поля для ввода логина и пароля и кнопка «Войти». Интерфейс страницы авторизации представлен на рисунке 7.

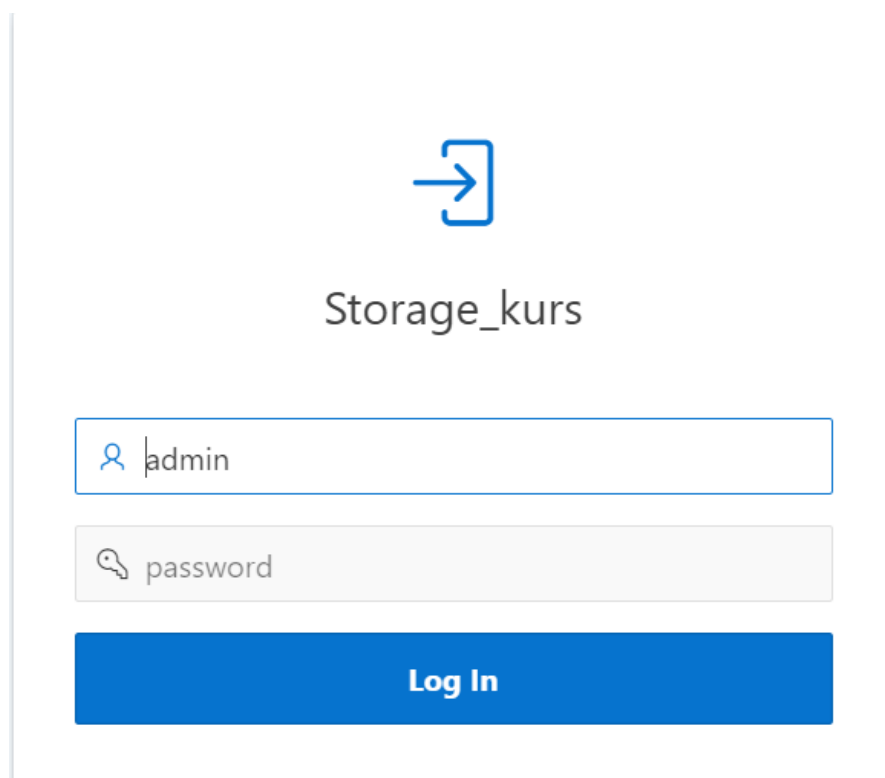
The image shows a login interface for an application named 'Storage_kurs'. At the top, there is a blue icon of a right-pointing arrow inside a square frame. Below the icon, the text 'Storage_kurs' is displayed. There are two input fields: the first one contains the text 'admin' and has a small blue user icon to its left; the second one contains the text 'password' and has a small grey key icon to its left. Below these fields is a solid blue button with the white text 'Log In'. The entire interface is enclosed in a light grey border.

Рисунок 6 - Страница авторизации

3.2. Создание страницы Администратора

Для администратора системы предусмотрена страница управления пользователями, которая включает в себя функцию создания, изменения и удаления пользователей и просмотра всех пользователей системы. Создание, удаление и изменение пользователей происходит при нажатии на соответствующие кнопки (см. Рисунок 7).

Storage_kurs Log Out

Домашняя страница

Эта программа для управления складами.

Изменение пользователей

Создать пользователя

ФИО

Логин

Пароль

Роль ☐ Администратор ☒ Пользователь

Создать

Изменить/удалить пользователя

Выберите пользователя

Логин

Пароль

ФИО

Удалить Изменить

Все пользователи

Номер ↑	Логин	Пароль	Роль	ФИО
1	admin	password	A	Иванов Иван Иванович
26	Crumple	Арех	A	Бурлаков Александр
31	Azazel	123456	P	Иванов Иван

1 - 3

Рисунок 7 - Страница управления пользователями

3.3. Страница входа

Страница входа в систему (Рисунок 8) состоит из формы для ввода имени учётной записи и пароля.

На этом этапе разница между пользователем и администратором отсутствует.

Также страница содержит кнопку входа “Log In”, запускающая проверку введённых данных на существование в таблице пользователей. Если пользователя, соответствующего введённым данным, не существует, то вход не будет произведён. Если проверка пройдена, то пользователь получает доступ к приложению в рамках своих прав.

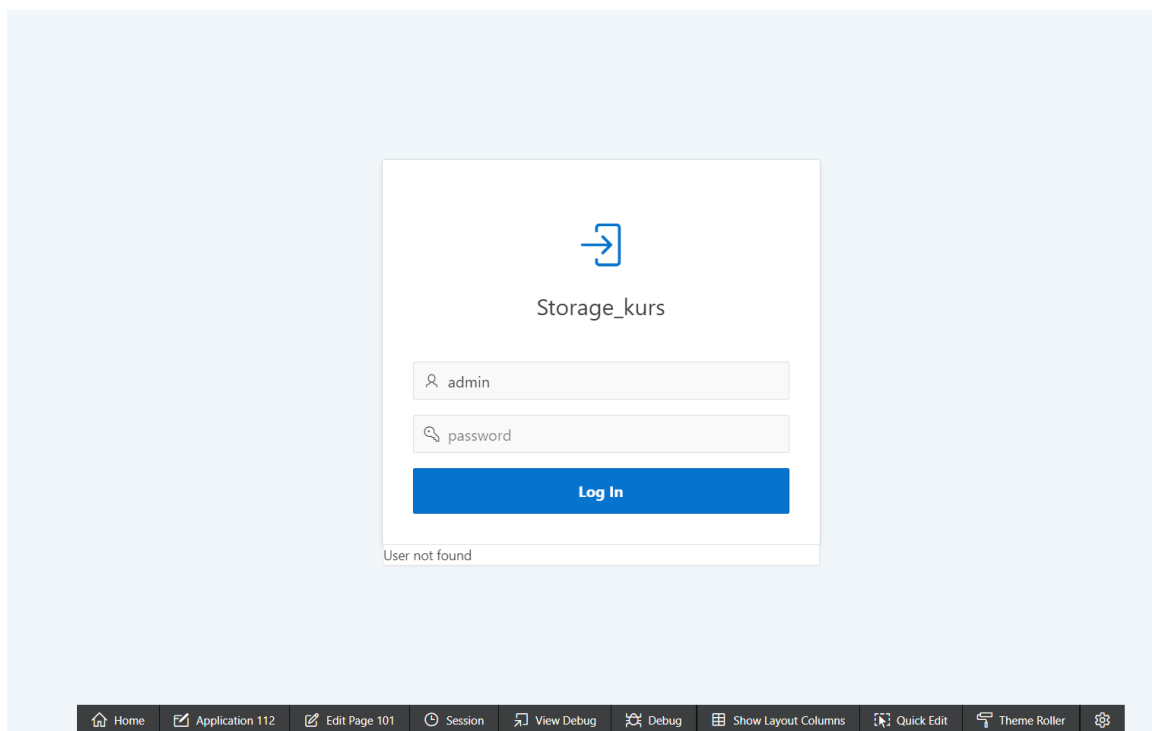


Рисунок 8 - Страница входа в систему

3.4 Страница создания заказа

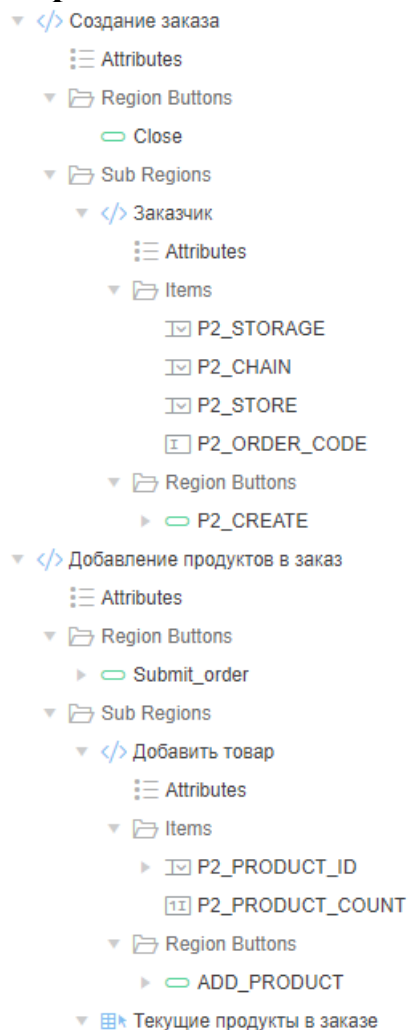


Рисунок 9 - поля страницы создания заказа

При создании страницы были прописаны SQL-коды для генерации значений Select List у полей P2_STORAGE, P2_CHAIN, P2_STORE, P2_PRODUCT_ID, которые представлены ниже:

SQL-код генерации значений у P2_STORAGE:

```
SELECT
  name, storage_id
FROM
  "STORAGE";
```

SQL-код генерации значений у P2_CHAIN:

```
SELECT
  chain_name, chain_store_id
FROM
  chain_stores;
```

SQL-код генерации значений у P2_STORE:

```
SELECT
  name||' '||address, store_id
FROM
  store
WHERE
  chain_store = :P2_CHAIN;
```

SQL-код генерации значений у P2_PRODUCT_ID:

```

select
    name, storage_product_id
from
    storage_product
where
    :P2_STORAGE = storage_id;

```

SQL-код заполнения таблицы «Текущие продукты в заказе»:

```

select
    s.name as "Название товара",
    p.count as "Количество",
    s.price as "Цена за штуку"
from
    storage_product s,
    order_product p
where
    p.order_code = :P2_ORDER_CODE and
    p.storage_product_id = s.storage_product_id;

```

При нажатии на кнопку «Создать» создается неактивный заказ. SQL-код создания такого заказа:

```

IF (:P2_STORE IS NOT NULL) AND
    (:P2_STORAGE IS NOT NULL)
THEN
    INSERT INTO "Order"
        (store_id, status, storage_id, order_date)
    VALUES
        (:P2_STORE, 'N', :P2_STORAGE, CURRENT_DATE);
END IF;
:P2_ORDER_CODE := ORDER_SEQ.CURRVAL;

```

При нажатии на кнопку «Добавить продукт» создается новая запись в таблице «Order_product». SQL-код добавления товара:

```
add_order_product(:P2_PRODUCT_ID, :P2_PRODUCT_COUNT, :P2_ORDER_CODE);
```

При нажатии на кнопку «Определить цену» в поле P2_PRICE выводится общая цена заказа. SQL-код определения цены:

```

DECLARE
    C INT := 0;
BEGIN
    FOR PRODUCT IN (SELECT * FROM ORDER_PRODUCT WHERE ORDER_CODE =
:P2_ORDER_CODE) LOOP
        FOR STOR_PROD IN (SELECT * FROM STORAGE_PRODUCT
            WHERE PRODUCT.STORAGE_PRODUCT_ID = STORAGE_PRODUCT_ID) LOOP
            C := C + PRODUCT.COUNT * STOR_PROD.PRICE;
        END loop;
    END loop;
    :P2_PRICE := C;
end;

```

При нажатии на кнопку «Принять заказ» заказ становится активным и его можно увидеть во всех остальных страницах приложения. SQL-код активации заказа:

UPDATE "Order" SET status='Y' where order_id = :P2_ORDER_CODE;

Заказчик

Выберите склад: Склад им.Чехова

Сеть магазинов: Дикси

Магазин: Дикси Ковров, ул.Детярева д.56

Order code: 187

Создать заказ

На главную

Добавление продуктов в заказ

Добавить товар

Название продукта: Не выбрано

Добавить

Цена:

Определить цену

Q Go Actions

Q

Принять заказ

Рисунок 10 - страница создания заказа

ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта была создана система работы со складом.

Данная система основана на базе данных, реализованной при помощи программных средств СУБД MySQL, и имеет пользовательский интерфейс, созданный на основе APEX Oracle.

В процессе реализации проекта были выполнены:

- Исследование предметной области, ознакомление с существующими аналогами и разработка концепции системы;
- Построение схемы объектов базы данных при помощи Oracle Database
- Генерация SQL-кодов по созданной модели, исправление ошибок генерации и необходимая доработка в соответствии с особенностями СУБД
- Заполнение базы данных;
- Реализация пользовательского интерфейса с помощью APEX Oracle;
- Проверка функционирования всех форм интерфейса.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- 1 Райордан Р, Основы реляционных баз данных [Текст] / Ребекка Райордан: Основы реляционных баз данных - Microsoft Press Русская редакция, 2012, 390с. - ISBN 5-7502-0150-3
- 2 Грофф Джеймс Р., SQL Полное руководство [Текст] / Джеймс Р. Грофф, SQL Полное руководство - Издательский дом "Вильямс", 2015, 390с. - ISBN 978-5-8459-1654-9
- 3 Иванова Г.С., Ничушкина Т.Н. Оформление текстовых документов. Методические указания по оформлению расчетно-пояснительных записок дипломных и квалификационных работ. - М.: Издательство МГТУ им.Н.Э. Баумана, 2004. - 10 с.
- 4 Фейерштейн С., Прибыл Б. Oracle PL/SQL. Для профессионалов. 6-е изд. — СПб.: Питер, 2015. — 1024 с.

ПРИЛОЖЕНИЯ

Приложение А

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на 8 листах

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э.
Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)
Факультет «Информатика и системы управления»
Кафедра «Компьютерные системы и сети»

УТВЕРЖДАЮ
Зав. кафедрой ИУ6,
д.т.н., проф. _____ А.В. Пролетарский
" ____ " _____ 2018 г.

Прототип программного комплекса «СкладскийУчет»

Техническое задание на курсовую работу

По дисциплине «Базы данных»

Листов 8

Руководитель,
Ассистент _____ Скворцова Мария Александровна

Исполнитель,
Студент гр. ИУ6-42 _____ Бурлаков Александр Сергеевич

2018 г.

1 ВВЕДЕНИЕ

1.1 Наименование программного изделия

Полное наименование программной разработки: «Прототип программного комплекса автоматизации работы службы учета товаров на складе «СкладскийУчет»», в дальнейшем именуемый как ППК «СкладскийУчет».

1.2 Область применения

ППК «СкладскийУчет» предназначен для учета сотрудников склада, учета обслуживаемых магазинов и складов, учета товаров на складах, создания и формирования заказов и отгрузки товаров по накладным. ППК «СкладскийУчет» предназначен для использования сотрудниками склада, администрацией склада.

Актуальность разработки обоснована тем, что данная система является специализированным свободно распространяемым программным комплексом, конкурирующим в данной прикладной области с другими программными продуктами. В качестве аналога рассматривается программный продукт «1С: Торговля и склад».

2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Основанием для разработки программного комплекса является учебный план кафедры ИУ6 «Компьютерные системы и сети» факультета ИУ «Информатика и системы управления» МГТУ им. Н.Э.Баумана, утвержденный в установленном порядке.

3 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Основные назначения ППК «СкладскийУчет» заключается в:

3.1 Создание программного комплекса, необходимого для:

3.1.1 Учет сотрудников склада:

3.1.1.1 Администратор.

3.1.1.2 Складской работник.

3.1.2 Учета обслуживаемых складов.

3.2 Создание клиент-серверного web-приложения для работы всех сотрудников службы в ППК «СкладскийУчет» через браузер.

4 ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

4.1 Требования к функциональным характеристикам

4.1.1 Состав выполняемых функций

4.1.1.1 ППК «СкладскийУчет» обеспечивает выполнение администратором функций добавления, изменения и удаления информации о сотрудниках.

4.1.1.2 ППК «СкладскийУчет» обеспечивает выполнение менеджером по заказам функций создания, изменения и удаления заказов, функций создания, изменения и удаления заказа со склада поставщика.

4.1.1.3 ППК «СкладскийУчет» обеспечивает выполнение грузчиком функции отметки об отдаче товаров.

4.1.1.4 ППК «СкладскийУчет» обеспечивает выполнение пользователем функций определения объемов заказов по каждому из товаров на текущий месяц, определения количества отгруженных товаров каждому из магазинов, просмотра остатков товаров на складе, просмотра магазинов, которые заказывали имеющиеся на складе товары, просмотра заказов, просмотра заказов, сделанных определенным магазином.

4.1.2 Исходные данные

4.1.2.1 Описание функционального и эксплуатационного назначений (см. п.3).

4.1.2.2 ГОСТ 19.201-78 «Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению».

4.1.2.3 Аналог – программный продукт «1С: Торговля и склад».

4.1.2.4 Официальная документация к языкам программирования и фреймворкам (см п.4.5.1).

4.1.2.5 Структуры данных

а) Пользователь

- 1) Фамилия;
- 2) Имя;
- 3) Отчество;
- 4) Логин;

б) Магазин

- 1) Название;
- 2) Адрес.

с) Продукт

- 1) Название;
- 2) Количество;
- 3) Цена.

д) Склад

- 1) Баланс;
- 2) Название;
- 3) Администратор.

е) Заказ

- 1) Магазин;
- 2) Статус заказа;

3) Цена заказа;

4) Склад.

4.1.3 Результаты:

4.1.3.1 Прайс-лист всех товаров на складе.

4.1.3.2 Полная информация о всех заказах.

4.1.3.3 Информация о наличии товаров на складе.

4.2 Требования к надежности

4.2.1 Предусмотреть контроль вводимой информации.

4.2.2 Предусмотреть защиту от некорректных действий пользователя.

4.2.3 Обеспечить целостность информации в базе данных встроенными в фреймворк средствами (см. п.4.1.2.4).

4.3 Условия эксплуатации

4.3.1 Условия эксплуатации в соответствии с СанПиН 2.2.2/2.4.1340-03.

4.3.2 Системный администратор обеспечивает запуск серверной части программного комплекса после перезагрузки машины, на которой она располагается.

4.3.3 Выполнение функций системного администратора для обслуживания серверной части программного комплекса предусматривается штатным расписанием.

4.4 Требования к составу и параметрам технических средств

4.4.1 Серверная часть

Минимальная конфигурация технических средств для функционирования серверной части программного комплекса:

4.4.1.1 Тип процессора: Intel Core 2 Duo

4.4.1.2 Объем ОЗУ: 2048 MB.

4.4.1.3 Дискковое пространство: 2048 МВ.

4.4.1.4 Наличие адаптера подключения к целевой сети развертывания системы со скоростью передачи данных не менее 20Мбит/с (сетевая карта, модем и т.п.).

4.4.2 Клиентская часть

Минимальная конфигурация технических средств для функционирования клиентской части программного комплекса:

4.4.2.1 Наличие адаптера подключения к целевой сети развертывания системы со скоростью передачи данных не менее 20Мбит/с (сетевая карта, модем и т.п.).

4.4.2.2 Минимальный объем ОЗУ и тип процессора зависит от минимальных технических требований браузеров из п.п.

4.5.4.2.

4.5 Требования к информационной и программной совместимости

4.5.1 Требования к языкам программирования

Разработка серверной части программного комплекса осуществляется с использованием Oracle SQL, используя фреймворк Apex 5.0 и выше.

4.5.2 Требования к программным средствам, используемым программой

4.5.2.1 Серверная часть программного комплекса

Для работы серверной части программного комплекса необходима операционная система Windows 7 и более поздние версии.

4.5.2.2 Клиентская часть программного комплекса

Для работы клиентской части программного комплекса необходим один из следующих браузеров:

- а) Google Chrome v62.0.3202.94 и выше;
- б) Яндекс.Браузер v18.3.1.1232 и выше.

4.6 Требования к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются.

4.7 Требования к транспортированию и хранению

Требования к транспортировке и хранению не предъявляются.

5 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1 Разрабатываемые программные модули должны быть самодокументированы, т.е. тексты программ должны содержать все необходимые комментарии.

5.2 В состав сопровождающей документации должны входить:

5.2.1 Расчетно-пояснительная записка на 15-25 листах формата А4 (без приложений 5.2.2, 5.2.3 и 5.2.4).

5.2.2 Техническое задание (Приложение А).

5.2.3 Графическая часть (формата А4)(Приложение Б).

5.2.3.1 Схема структур данных.

5.2.3.2 Диаграмма вариантов использования.

5.2.3.3 Схема структурная программного комплекса.

5.2.3.4 Граф состояний интерфейса.

5.2.3.5 Схемы структурные меню

5.2.3.6 Формы интерфейса

5.2.4 Руководство пользователя (Приложение В).

5.2.5 Руководство системного администратора (Приложение Г).

6 ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Технико-экономические показатели не определяются

7 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

№	Название этапа	Срок, недели	Отчетность
1	Разработка технического задания	1-2	Техническое задание. п.5.2.2.
2	Анализ требований и уточнение спецификаций(эскизный проект).	3-4	Структурные схемы меню и формы интерфейса п.5.2.3.5-5.2.3.6
3	Проектирование структуры программного обеспечения, проектирование компонентов	5-9	Схема структуры системы и спецификация компонентов. Проектная документация п.5.2.3.1-5.2.3.4
4	Реализация компонентов.	10-14	Рабочая версия программного комплекса. Листинги программных модулей. п.5.2.6.
5	Разработка программной документации	14-15	Расчетно-пояснительная записка. Руководство пользователя. п.5.2.1, п.5.2.4-5.2.5
6	Подготовка доклада и предзащита	15 50 %	Доклад
7	Защита проекта	15-16 100 %	

8 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

8.1 Порядок контроля

Контроль выполнения осуществляется руководителем еженедельно.

8.2 Порядок защиты

Защита осуществляется перед комиссией преподавателей кафедры.

8.3 Срок защиты

Срок защиты: 16 неделя.

9 ПРИМЕЧАНИЕ

В процессе выполнения работы возможно уточнение отдельных требований технического задания по взаимному согласованию руководителя и исполнителя.

ПРИЛОЖЕНИЕ Б

Руководство пользователя

При входе в приложение вы попадете на страницу авторизации (см. рисунок Б.1).

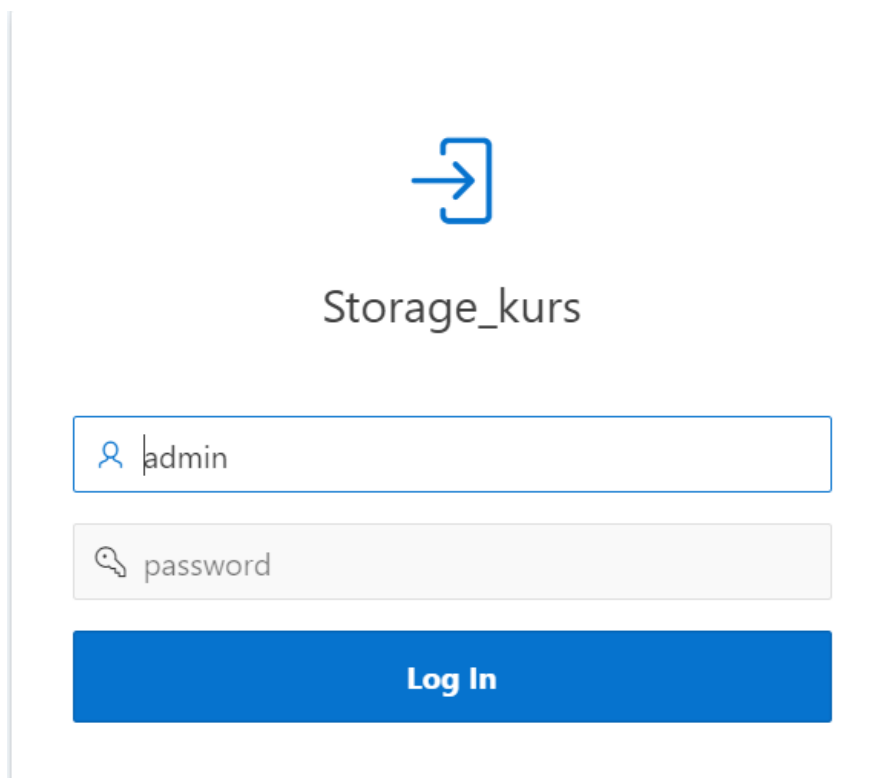
The image shows a login interface for an application named 'Storage_kurs'. At the top center is a blue icon of a square with an arrow pointing right. Below the icon is the text 'Storage_kurs'. There are two input fields: the first one contains the text 'admin' and has a small blue person icon on the left; the second one contains the text 'password' and has a small grey key icon on the left. Below these fields is a large blue button with the text 'Log In' in white.

Рисунок Б.1 – страница авторизации

После этого откроется начальная страница программы. Для перемещения между страницами используйте меню слева. После выбора нужной вам страницы и щелчку по кнопке вы сможете получить доступ к следующему:

- а) Создание заказа;
- б) Изменение и удаление заказа;
- в) Подтверждение заказа;
- г) Создание и подтверждение заказа дозакупки;
- д) Просмотр информации по заказам.

Для возвращения на страницу авторизации щелкните по кнопке «Log out» (см. рисунок Б.2).



Рисунок Б.2 – кнопка возвращения на страницу авторизации

Для пользователя-администратора

Вход в приложение производится так же, как и у обычного пользователя. После входа в приложение откроется дополненная начальная страница, которая содержит в себе формы добавления, изменения и удаления пользователей. Помимо этого можно просматривать всех текущих пользователей (см. рисунок Б.3).

Изменение пользователей

Создать пользователя

ФИО

Логин

Пароль

Роль

Администратор

Пользователь

Создать

Изменить/удалить пользователя

Выберите пользователя

Не выбрано

Логин

Пароль

ФИО

Удалить

Изменить

Рисунок Б.3 – начальная страница для администратора

Все пользователи

Номер ↑≡	Логин	Пароль	Роль	Фιο
1	admin	password	A	Иванов Иван Иванович
26	Crumple	Apex	A	Бурлаков Александр
31	Azazel	123456	P	Иванов Иван

1 - 3

Рисунок Б.3 – продолжение

ПРИЛОЖЕНИЕ В

SQL-код генерации таблиц

```
-- Generated by Oracle SQL Developer Data Modeler 17.2.0.188.1059
-- at:    2018-05-19 16:25:34 MSK
-- site:   Oracle Database 11g
-- type:   Oracle Database 11g
```

```
CREATE TABLE chain_stores (
  chain_store_id NUMBER NOT NULL,
  chain_name     CHAR(20 CHAR)
);
```

```
ALTER TABLE chain_stores ADD CONSTRAINT chain_stores_pk PRIMARY KEY ( chain_store_id );
```

```
CREATE TABLE "Order" (
  order_id      NUMBER NOT NULL,
  store_id      NUMBER NOT NULL,
  status        CHAR(1 CHAR),
  storage_id    NUMBER NOT NULL,
  order_date    TIMESTAMP,
  order_close_date  TIMESTAMP
);
```

```
ALTER TABLE "Order" ADD CONSTRAINT order_pk PRIMARY KEY ( order_id );
```

```
CREATE TABLE order_product (
  order_product_id NUMBER NOT NULL,
  storage_product_id NUMBER NOT NULL,
  count            NUMBER,
  order_id         NUMBER NOT NULL
);
```

```
ALTER TABLE order_product ADD CONSTRAINT order_product_pk PRIMARY KEY ( order_product_id );
```

```
CREATE TABLE product (
  product_id NUMBER NOT NULL,
  name       NUMBER,
  price      NUMBER
);
```

```
ALTER TABLE product ADD CONSTRAINT product_pk PRIMARY KEY ( product_id );
```

```
CREATE TABLE storage (
  storage_id NUMBER NOT NULL,
  balance    NUMBER,
  name       CHAR(20 CHAR),
  main_user  NUMBER NOT NULL
);
```

```
ALTER TABLE storage ADD CONSTRAINT storage_pk PRIMARY KEY ( storage_id );
```

```
CREATE TABLE storage_order (
  storage_order_id NUMBER NOT NULL,
  storage_id       NUMBER NOT NULL,
  status           CHAR(1 CHAR),
  open_date        TIMESTAMP,
  close_date       TIMESTAMP
);
```

```
ALTER TABLE storage_order ADD CONSTRAINT storage_order_pk PRIMARY KEY ( storage_order_id );
```

```
CREATE TABLE storage_order_product (
  storage_order_product_id NUMBER NOT NULL,
  product_id              NUMBER NOT NULL,
  count                   NUMBER,
  storage_order_id        NUMBER NOT NULL
);
```

```
ALTER TABLE storage_order_product ADD CONSTRAINT storage_order_product_pk PRIMARY KEY ( storage_order_product_id );
```

```

CREATE TABLE storage_product (
    storage_product_id NUMBER NOT NULL,
    name CHAR(30 CHAR) NOT NULL,
    count NUMBER,
    price NUMBER,
    storage_id NUMBER NOT NULL
);

ALTER TABLE storage_product ADD CONSTRAINT storage_product_pk PRIMARY KEY ( storage_product_id );

CREATE TABLE store (
    store_id NUMBER NOT NULL,
    name CHAR(20 CHAR),
    address CHAR(100 CHAR),
    chain_store NUMBER NOT NULL
);

ALTER TABLE store ADD CONSTRAINT store_pk PRIMARY KEY ( store_id );

CREATE TABLE "User" (
    user_id NUMBER NOT NULL,
    login CHAR(20 CHAR),
    password CHAR(30 CHAR),
    fio CHAR(50 CHAR),
    status NUMBER
);

ALTER TABLE "User" ADD CONSTRAINT user_pk PRIMARY KEY ( user_id );

ALTER TABLE order_product
    ADD CONSTRAINT or_prod_stor_prod_fk FOREIGN KEY ( storage_product_id )
        REFERENCES storage_product ( storage_product_id );

ALTER TABLE order_product
    ADD CONSTRAINT order_product_order_fk FOREIGN KEY ( order_id )
        REFERENCES "Order" ( order_id );

ALTER TABLE "Order"
    ADD CONSTRAINT order_storage_fk FOREIGN KEY ( storage_id )
        REFERENCES storage ( storage_id );

ALTER TABLE "Order"
    ADD CONSTRAINT order_store_fk FOREIGN KEY ( store_id )
        REFERENCES store ( store_id );

ALTER TABLE storage_order_product
    ADD CONSTRAINT stor_order_prod_prod_fk FOREIGN KEY ( product_id )
        REFERENCES product ( product_id );

ALTER TABLE storage_order_product
    ADD CONSTRAINT stor_order_prod_stor_order_fk FOREIGN KEY ( storage_order_id )
        REFERENCES storage_order ( storage_order_id );

ALTER TABLE storage_product
    ADD CONSTRAINT storage_product_storage_fk FOREIGN KEY ( storage_id )
        REFERENCES storage ( storage_id );

ALTER TABLE storage
    ADD CONSTRAINT storage_user_fk FOREIGN KEY ( main_user )
        REFERENCES "User" ( user_id );

ALTER TABLE store
    ADD CONSTRAINT store_chain_stores_fk FOREIGN KEY ( chain_store )
        REFERENCES chain_stores ( chain_store_id );

ALTER TABLE storage_order
    ADD CONSTRAINT table_11_storage_fk FOREIGN KEY ( storage_id )
        REFERENCES storage ( storage_id );

```

-- Oracle SQL Developer Data Modeler Summary Report:

```

--
-- CREATE TABLE          10
-- CREATE INDEX           0

```



```

-- ALTER TABLE          20
-- CREATE VIEW            0
-- ALTER VIEW             0
-- CREATE PACKAGE         0
-- CREATE PACKAGE BODY    0
-- CREATE PROCEDURE       0
-- CREATE FUNCTION        0
-- CREATE TRIGGER         0
-- ALTER TRIGGER          0
-- CREATE COLLECTION TYPE  0
-- CREATE STRUCTURED TYPE  0
-- CREATE STRUCTURED TYPE BODY 0
-- CREATE CLUSTER         0
-- CREATE CONTEXT         0
-- CREATE DATABASE        0
-- CREATE DIMENSION       0
-- CREATE DIRECTORY       0
-- CREATE DISK GROUP      0
-- CREATE ROLE            0
-- CREATE ROLLBACK SEGMENT 0
-- CREATE SEQUENCE        0
-- CREATE MATERIALIZED VIEW 0
-- CREATE SYNONYM         0
-- CREATE TABLESPACE     0
-- CREATE USER            0
--
-- DROP TABLESPACE      0
-- DROP DATABASE         0
--
-- REDACTION POLICY      0
--
-- ORDS DROP SCHEMA      0
-- ORDS ENABLE SCHEMA    0
-- ORDS ENABLE OBJECT    0
--
-- ERRORS                0
-- WARNINGS              0

```