Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

“Поволжский государственный технологический университет”

Факультет информатики и вычислительной техники

Кафедра информационной безопасности

**РАЗРАБОТКА БАЗЫ ДАННЫХ И ПРИЛОЖЕНИЯ**

**«Аптека»**

пояснительная записка

к курсовому проекту по дисциплине

“Безопасность систем базы данных”

Вариант №5

Индексы и триггеры

Выполнили: студенты гр. БИ-32

Мамаев И.C., Хакимов Д.И.,

Михайлов С.В.

Проверил: доцент каф. ИБ

Сучков Д.С.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_

Йошкар-Ола

2019 г.

**Содержание**

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ 3

ВВЕДЕНИЕ 4

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5

1.1 Основные определения 5

1.2 Модели данных СУБД 7

1.3 Характеристика реляционной СУБД 8

2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 9

2.1 Анализ задания 9

2.2 Разработка состава и структуры таблиц 9

2.2.1 Функциональные зависимости данных 10

2.2.2 Поля и типы данных таблиц 10

2.3 Ключи и связи страниц 12

2.4 Создание различных пользователей 12

2.5 Создание триггеров 14

3. ПРОГРАММНАЯ ЧАСТЬ 18

3.1 Обоснование выбора языка программирования 18

3.2 Характеристика языка запросов SQL 18

4. ПРАВА ДОСТУПА К ДАННЫМ 18

ЗАКЛЮЧЕНИЕ 21

ЛИТЕРАТУРА 22

ПРИЛОЖЕНИЕ 23

SQL запрос для создания таблиц и формирование связей………………… .. 25

**ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ**

БД – база данных;

НФ – нормальная форма;

СУБД – система управления базами данных;

ФЗ – функциональные зависимости;

РМД – реляционная модель данных.

**ВВЕДЕНИЕ**

Сегодня информация рассматривается как один из основных ресурсов развития общества, а информационные системы и технологии как средство повышения производительности и эффективности работы людей, а потому применение баз данных и автоматизированных информационных систем - обязательная составляющая деловой деятельности современного человека.

Базы данных позволяют хранить информацию о большом числе объектов различных типов для многих областей деятельности человека. При этом имеется возможность быстрого и удобного поиска данных в базе, что выгодно отличает базы данных от файлов.

Актуальность темы курсовой работы в настоящее время очень высока. Использование грамотно разработанных БД и клиентского приложения позволяет своевременно получать достоверную информацию и также быстро вносить изменения, автоматизировать основные операции, вести учет расходов предприятия, минимизировать денежные, временные и иные затраты, тем самым обеспечить успешную деятельность предприятия.

Цель курсовой работы заключается в проектировании и создании БД «Аптека» и клиентского приложения для работы с ней.

Для достижения поставленной цели необходимо решить следующие задачи:

1) анализ предметной области;

2) разработка БД;

3) разработка пользовательского интерфейса;

4) разработка отчетов;

5) тестирование и отладка разработанного ПО.

**1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**1.1 Основные определения**

В современных системах данные – это отдельная категория, с которой могут работать не одно, а несколько приложений.

Под базой данных (БД) понимается совокупность данных, которая хранится строго определенным образом в соответствии со схемой данных, и с которой может работать любое количество приложений.

Из определения ясно, что строго определённая структура БД позволяет охарактеризовать множество объектов из конкретной предметной области – ограниченной области человеческой деятельности или реального мира, данные которой нуждаются в хранении или управлении.

При этом в каждой предметной области можно выделить информационные объекты, каждый из которых имеет набор характеристик.

В БД характеристики называют полями (атрибутами). Каждое поле определяется уникальным именем (идентификатором) и множеством допустимых значений (доменов).

Данные – это значения, принимаемые полями.

Структура, содержащая данные о конкретном информационном объекте, называется записью (кортежем).

А множество однотипных записей называется отношением (таблицей).

Информационные объекты предметной области находятся в определенных отношениях друг с другом. Эти отношения еще называют связями, которые делятся на обязательные и факультативные.

Важной характеристикой связи является кардинальность, показывающая, сколько сущностей информационного объекта участвует в конкретной связи. Выделяют три вида кардинальных связей:

1) Связь 1:1 (в любой момент времени одному экземпляру в сущности А соответствует только один экземпляр сущности В);

2) Связь 1:М (одному экземпляру в сущности А соответствует много экземпляров сущности В);

3) Связь М:М (экземпляры сущностей А и В имеют множественные связи).

Сами по себе базы данных не статичны, а подвержены постоянным изменениям, поэтому ей необходимо постоянно управлять. К тому же необходимы средства, предоставляющие возможности удобного и быстрого манипулирования данными из базы.

Система управления базой данных (СУБД) – совокупность языковых и программных средств, предоставляющих возможности по созданию и модификации базы данных. Таким образом, БД является лишь своеобразным хранилищем, работа с которым осуществляется через СУБД.

Программы, с помощью которых пользователи работают с БД, называются приложениями.В общем случае с одной БД могут работать множество различных приложений. СУБД с точки зрения приложения предоставляет универсальный программный интерфейс для взаимодействия с базой данных.

При рассмотрении приложений, работающих с одной БД, предполагается, что они могут работать параллельно и независимо друг от друга, и именно СУБД призвана обеспечить работу множества приложений с единой базой данных таким образом, чтобы каждое из них выполнялось корректно, но учитывало все изменения в базе данных, вносимые другими приложениями.

Поэтому к СУБД предъявляется ряд требований:

1) производительность;

2) минимальное дублирование данных;

3) поддержка целостности данных;

4) безопасность данных;

5) независимость данных;

6) синхронизация;

7) защита от отказов и восстановление данных;

8) наличие языка запросов для манипуляции с данными.

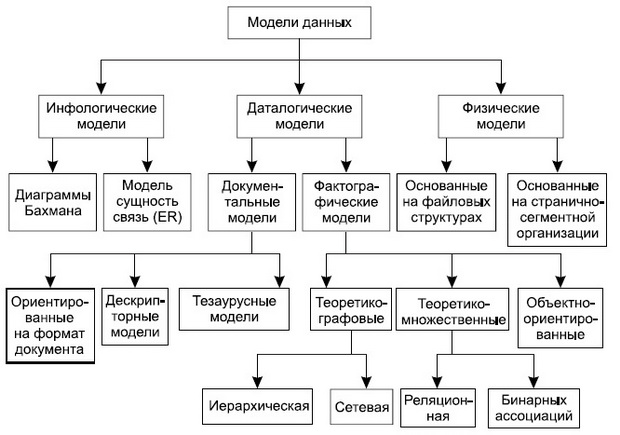
**1.2 Модели данных СУБД**

Хранимые в базе данные имеют определенную логическую структуру, то есть представлены некоторой моделью, поддерживаемой СУБД.

Модель данных представляет собой комбинацию из 3 элементов:

1. коллекция типов объектов;
2. коллекция правил целостности данных;
3. коллекция операций, применяемых к объектам.

Классификация моделей данных представлена на рисунке 1.



**Рисунок 1. Классификация моделей данных**

Каждый класс моделей непосредственно связан с архитектурой СУБД, исключения составляют инфологические модели.

Инфологическая модель описывает структуру данных и процесса, предметной области в терминах, понятных пользователям информационной системы и программистам.

Физические модели определяют способы хранения данных в памяти ЭВМ.

Даталогические модели соответствуют концептуальному уровню архитектуры СУБД, они имеют наибольшее значение для разработчиков. В этом классе прежде всего выделяется прежде всего документальная модель, широко применяющаяся в системах обработки и хранения больших данных; в нее входят ориентированные на формат документы.

**1.3 Характеристика реляционной СУБД**

Реляционная модель данных (РМД) − логическая модель данных, строгая математическая теория, описывающая структурный аспект, аспект целостности и аспект обработки данных в реляционных базах данных.

РМД включает в себя 3 части:

1) Структурная (определяются понятия домена, атрибута, кортежа, отношений и утверждается единственная структура данных в РМД – отношение);

2) Манипуляционная (определяются способы работы с данными);

3) Целостная (фиксируются 2 базовых требования целостности: требование целостности сущности и требование целостности по ссылкам).

Достоинством РМД является простота представления данных в форме таблиц и развитый математический аппарат. К недостаткам можно отнести то, что моделирование связей, отличных от 1:1 и 1:М затруднено, так как при проектировании БД должны выполняться процедуры нормализации отношений, из-за этого растет количество отношений в БД. Другим недостатком является тот факт, что в РМД нет специальных механизмов навигации по данным, поэтому единственным способом является последовательный просмотр кортежей.

Основным методом проектирования РМД является нормализация – формальный метод анализа и преобразования отношений на основе первичного ключа и функциональных зависимостей между данными. Она включает в себя декомпозицию (разбиение) отношения на 2 или более отношений, при этом декомпозиция должна быть обратимой, то есть не должны появляться новые кортежи или исчезать существующие.

**2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

**2.1 Анализ задания**

Данное техническое задание предполагает построение инфологической модели предметной области «Аптека», выбор модели данных и СУБД, определение структуры БД: таблиц для хранения информации, связей между ними, ограничений на значения полей. В результате должна быть спроектирована БД, информация которой будет использоваться в дальнейшем пользователями, имеющими доступ к БД, для создания, изменения, удаления и выборки данных.

Разработка приложения предполагает наличие системы прав доступа к данным пользователей. Для этого необходимо сделать разграничение прав доступа с помощью авторизации с использованием пары логин/пароль для пользователей. В результате разрабатываемый проект должен осуществлять вход пользователя в систему после корректно введенных данных входа; выполнять операции по изменению строк таблиц БД (добавление, изменение, обновление); выполнять поиск по критериям, предусмотренными системой.

К проекту предъявляется ряд требований, а именно: понятный и удобный интерфейс, безошибочная работа, разработка средствами, указанными в задании.

**2.2 Разработка состава и структуры таблиц**

Структура таблиц разрабатывается путем анализирования предметной области, то есть определяется состав полей, их имена, последовательность размещения их в таблице, тип данных каждого поля, размер поля, первичные ключи, индексы и другие свойства полей. Таким образом, будет создано 7 таблиц, в которые войдут следующие поля:

1. positions (id\_position, name\_position, description\_position);
2. buyer (id\_buyer, fio\_buyer, phone\_number\_buyer);
3. drugstore (id\_drugstore, drugstore\_adress, phone\_number\_drugstore);
4. maintenance\_drugstore (id\_main, id\_maintenance\_drugstore, product, oreder\_id, availability\_date, worker);
5. orders (num\_order, description\_order, name\_buyer);
6. product (id\_product, name\_product, price\_product, product\_description);
7. workers (id\_worker, fio\_worker, position, phone\_number\_worker, worker\_adress)

**2.2.1 Функциональные зависимости данных**

Функциональная зависимость (ФЗ) описывает связь между ат­рибутами и является одним из основных понятий нормализации. Обозначается: **X→Y** (X функционально определяет Y). Левая часть выражения называется детерминантом ФЗ, правая – зависимой частью.

**2.2.3 Поля и типы данных таблиц**

В таблицах 1-5 перечислены поля и типы данных таблиц БД.

Таблица 1. positions

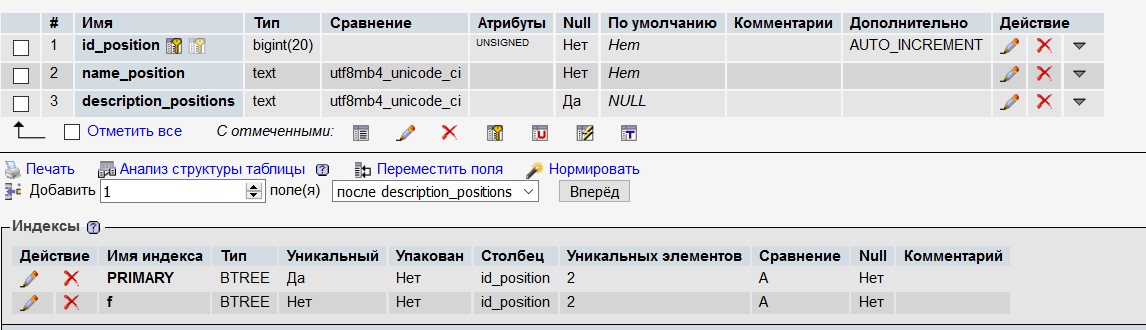


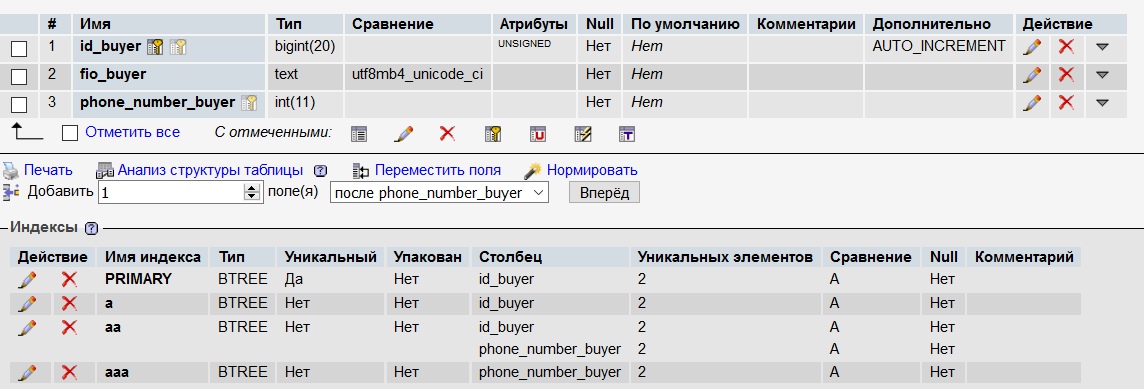
Таблица 2. buyer  


Таблица 3. drugstore

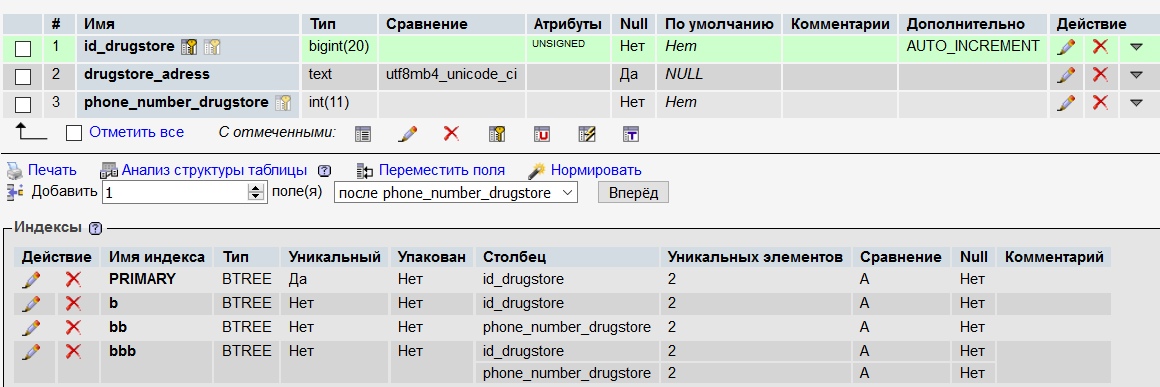


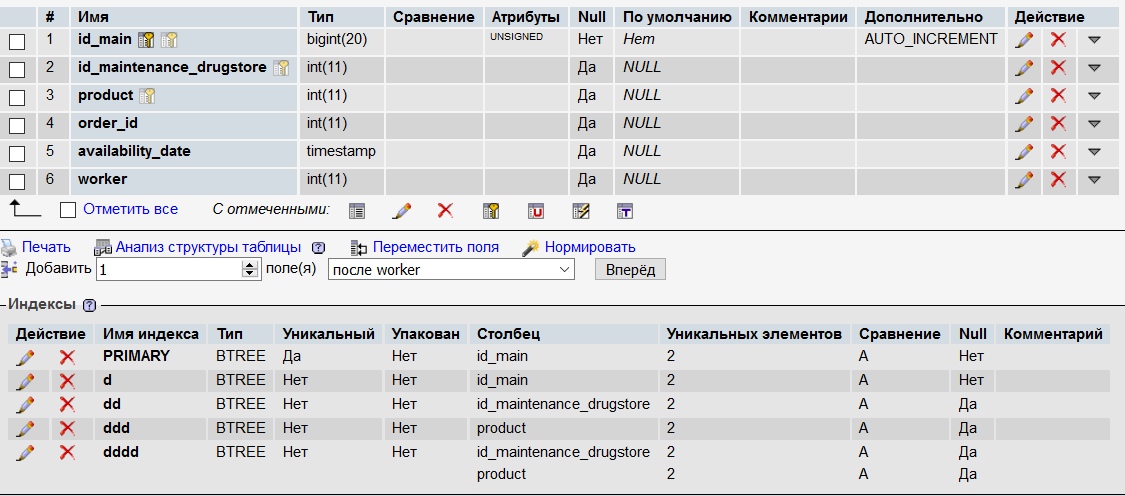
Таблица 4. maintenance\_drugstore

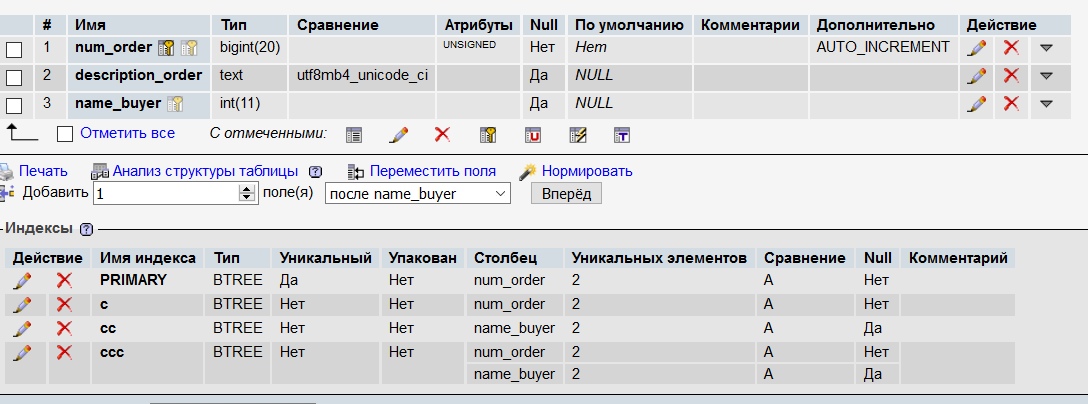
Таблица 5. orders

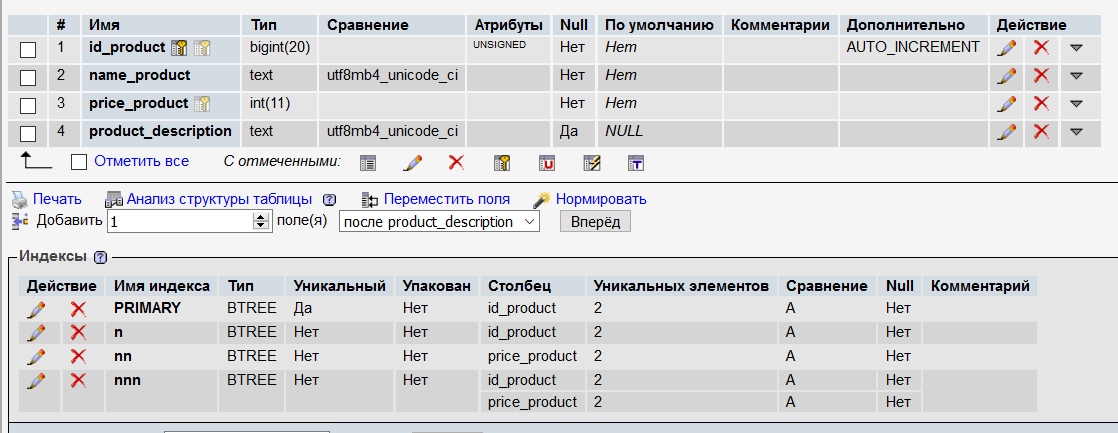
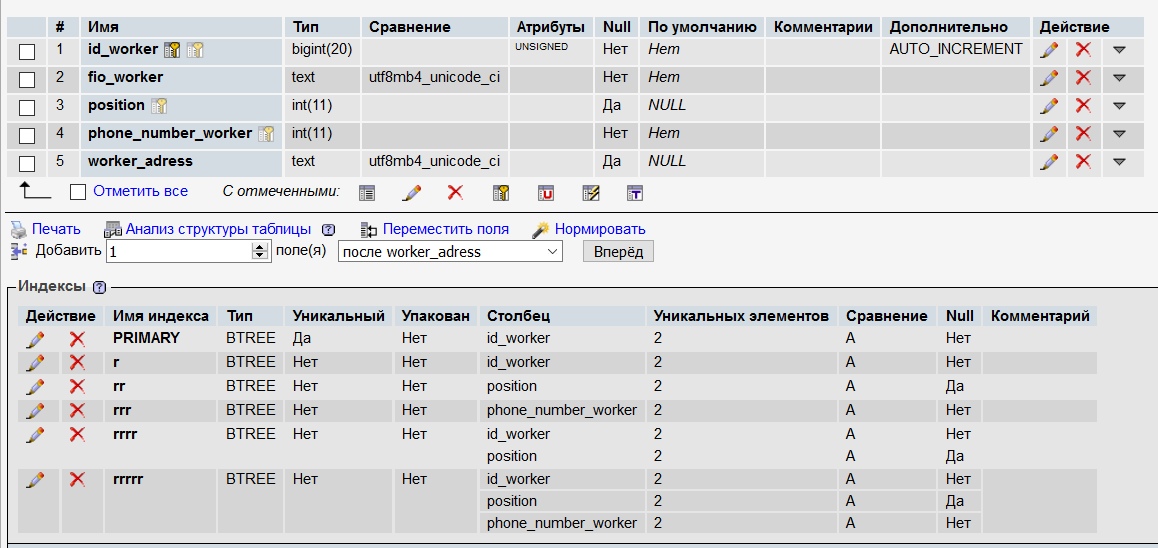
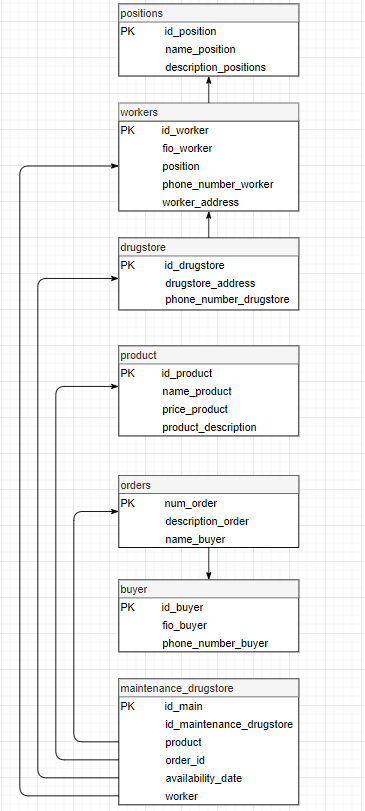
Таблица 6. product

Таблица 7. workers



**2.3 Ключи и связи таблиц**



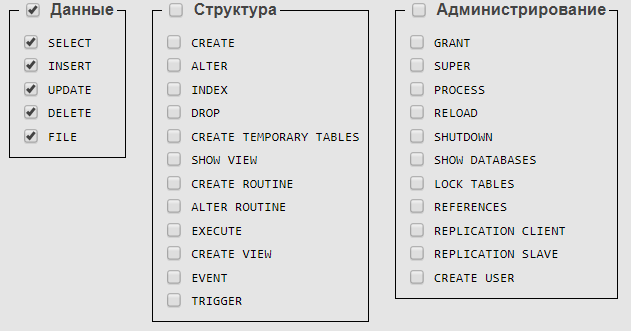
В рамках курсовой работы разработана реляционная база данных, приведенная к 3 нормальной форме.   
Переменная отношения R находится в 3NF тогда и только тогда, когда выполняются следующие условия:   
1) R находится во второй нормальной форме.

2) Ни один неключевой атрибут R не находится в транзитивной функциональной зависимости от потенциального ключа R.

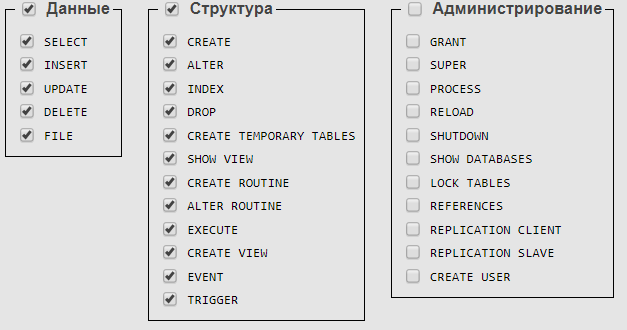
**2.4 Создание различных пользователей**

Существует 3 типа пользователей базы данных:

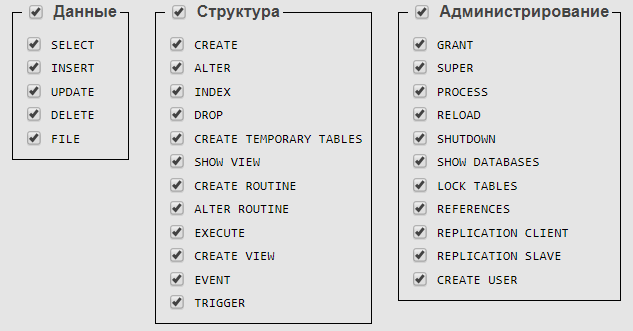
1) User – пользователь, который может работать только с данными БД



2) S\_user – пользователь, который может работать с данными и структурой БД



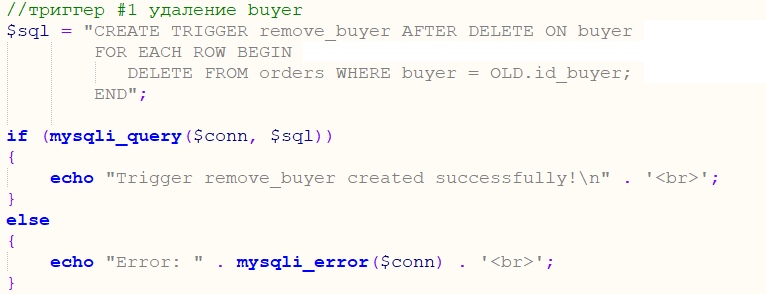
3) Admin – пользователь, которому предоставляются все возможности работы с БД



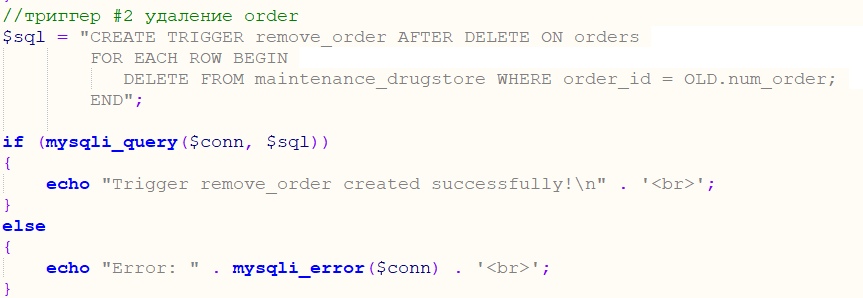
**2.5 Создание триггеров.**

Триггеры удаления:

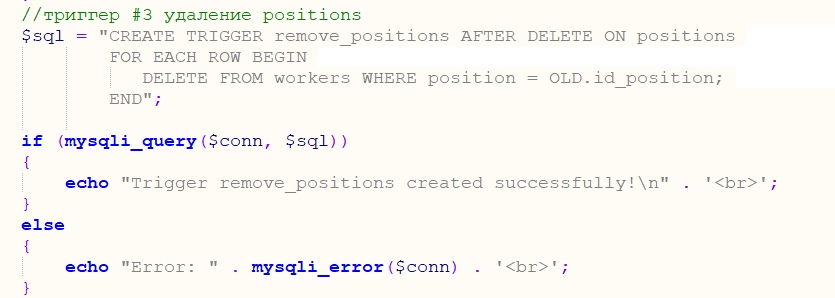
1) Удаление buyer:



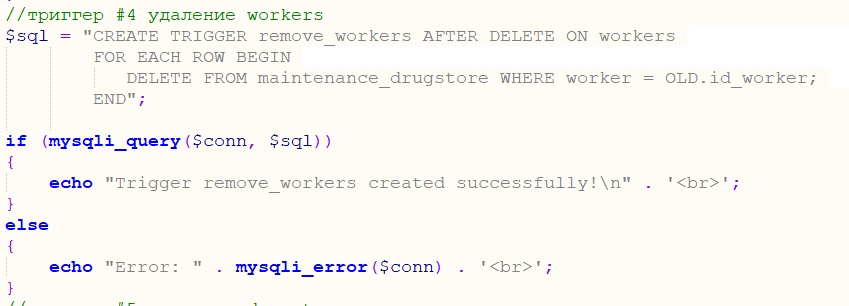
2) Удаление order:



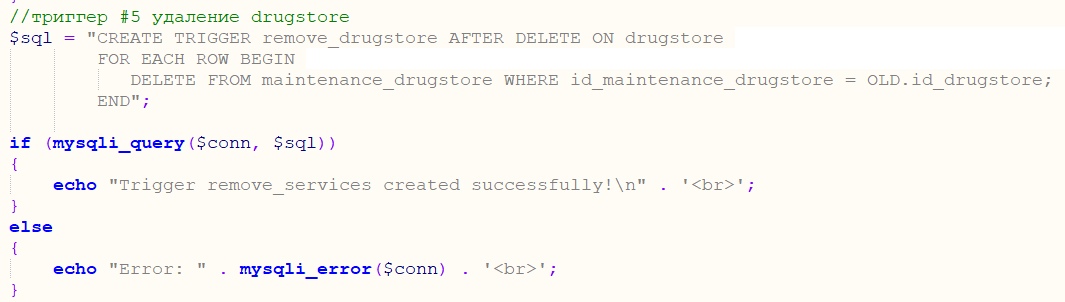
3) Удаление positions:



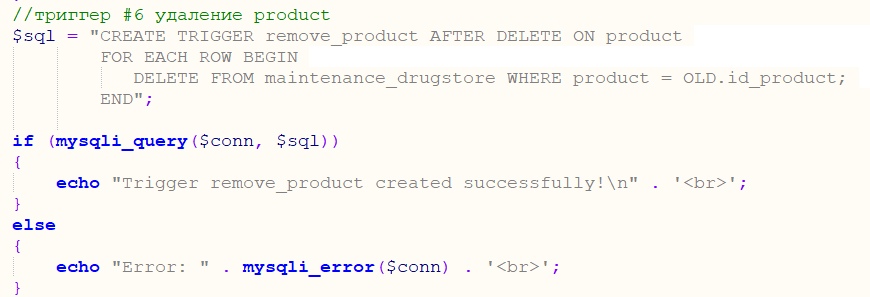
4) Удаление workers:



5) Удаление drugstore:

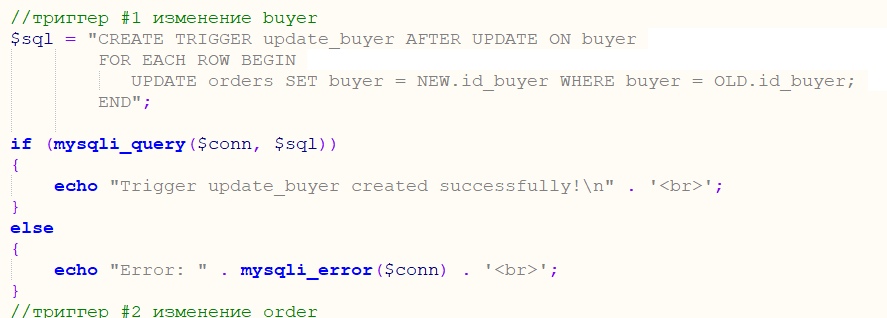


6) Удаление product:

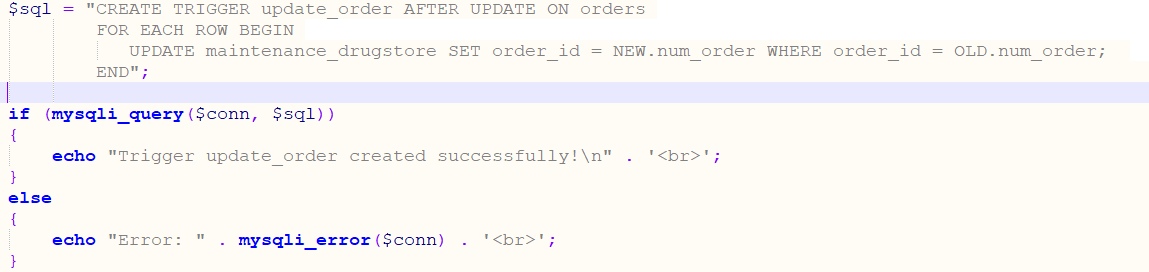


Триггеры изменения:

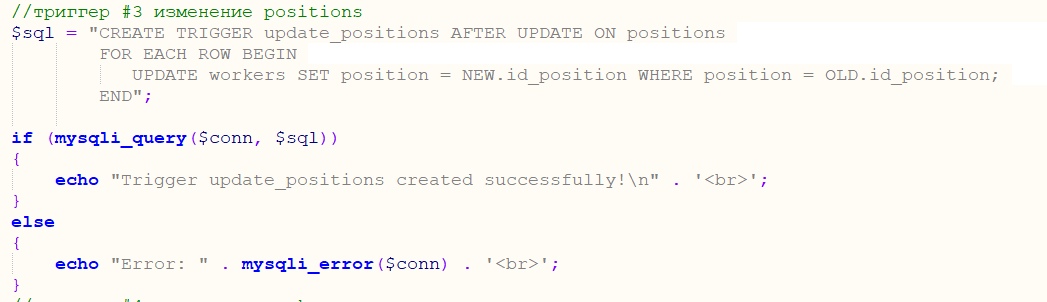
1) Изменение buyer:



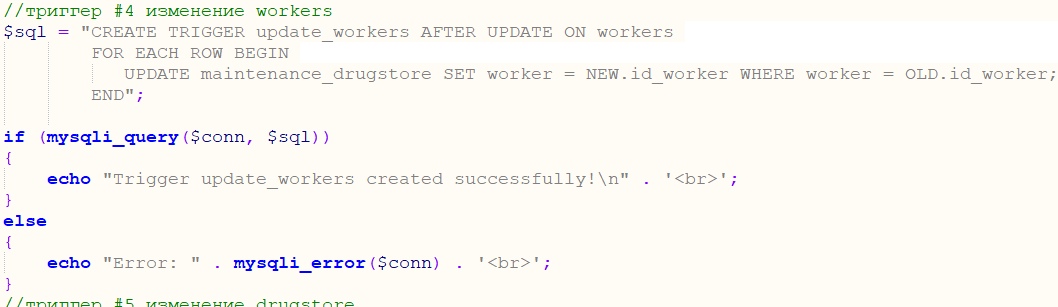
2) Изменение order:



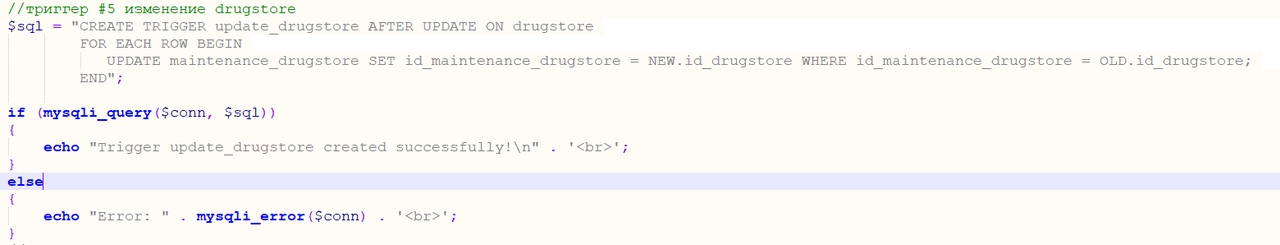
3) Изменение positions:



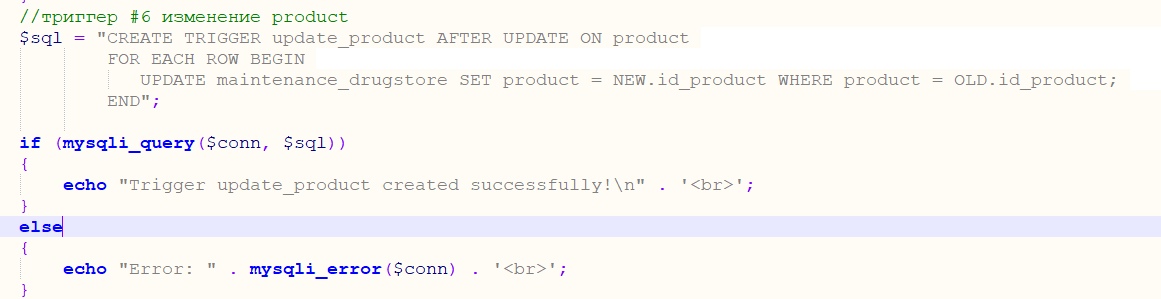
4) Изменение worker:



5) Изменение drugstore:

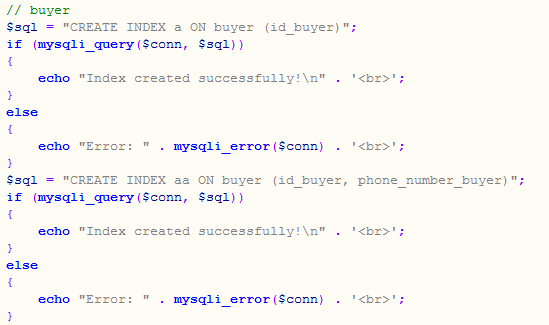


6) Изменение product:

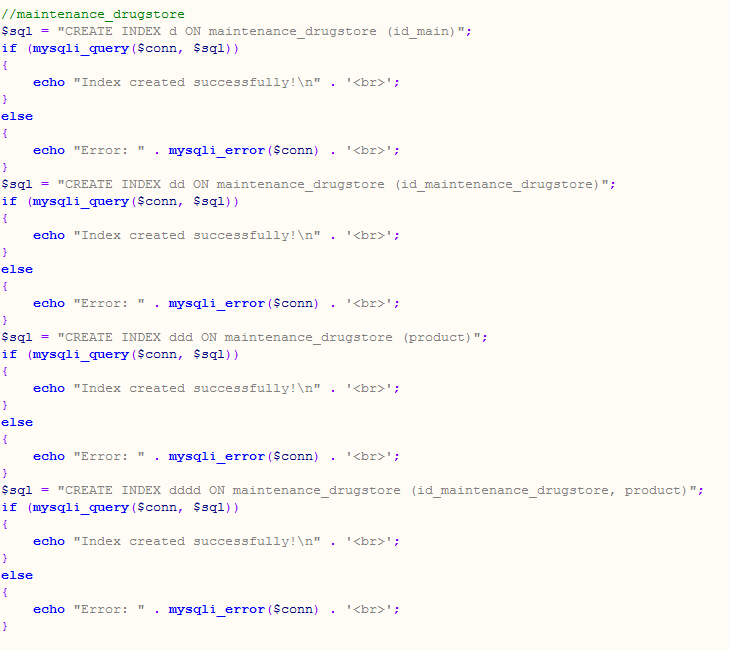


Индексы:

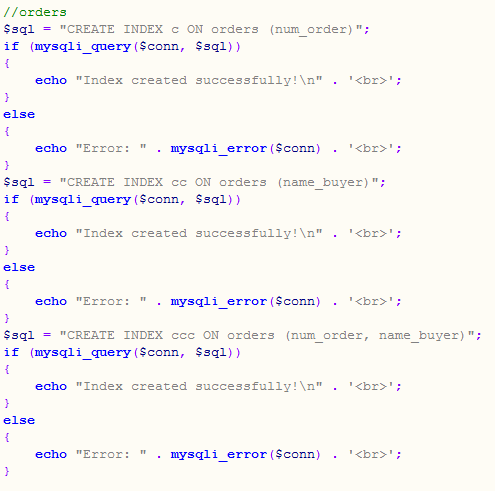
1)buyer



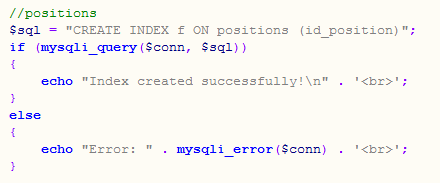
2) drugstore

3) maintenance\_drugstore

4) orders



5) positions



6) product



7) workers



**3. ПРОГРАММНАЯ ЧАСТЬ**

**3.1 Обоснование выбора языка программирования**

Для создания курсового проекта была выбрана интегральная среда разработки Open Server, потому что:

1) доступны широкие возможности языка php ;

2) имеются все необходимые средства и инструменты для разработки приложений любой сложности;

3) удобство в использовании, гибкость в настройках;

4) имеется возможность импортировать исходный код и проекты для интегральных сред разработки других операционных систем;

Языком программирования в Open Server был выбран php, потому что:

1) подлинная объектная ориентированность;

2) компонентно-ориентированное программирование;

3) безопасный (по сравнению с языками C и C++) код;

4) унифицированная система типизации;

**3.2 Характеристика языка запросов SQL**

Язык запросов SQL ориентирован на работу с множествами (отношениями), а не с отдельными записями. Как и в реляционной алгебре, операндами языка являются отношения (таблицы), результатами выполнения операции также являются отношения (таблицы). Язык SQL предназначен для выполнения операций над таблицами, причем как над таблицами в целом (создание, изменение структуры), так и над данными таблиц (выборка, изменение, добавление и удаление).

**4. ПРАВА ДОСТУПА К ДАННЫМ**

Защита данных от несанкционированного доступа является одной из приоритетных задач при разработке курсового проекта. Обеспечение информационной безопасности СУБД приобретает решающее значение при выборе конкретного средства обеспечения необходимого уровня безопасности организации в целом.

Для СУБД важны три основных аспекта информационной безопасности:

1) конфиденциальность;

2) целостность;

3) доступность.

Вся информация делится на общедоступные данные и конфиденциальные, доступ к которым разрешен только для отдельных групп лиц.

Общий принцип управления доступом к БД состоит в следующем: СУБД не должна разрешать пользователю выполнение какой-либо операции над данными, если он не получил на это права. Политика безопасности определяется администратором данных. В его обязанности входит:

1) назначение отдельным группам пользователей прав доступа (привилегий) к отдельным группам данных в соответствии с правилами ПО;

2) организация системы контроля доступа к данным;

3) тестирование вновь создаваемых средств защиты данных;

4) периодическое проведение проверок правильности работы системы защиты, исследование и предотвращение сбоев в её работе.

**Парольная идентификация** заключается в присвоении каждому пользователю двух параметров: имени (login) и пароля (password). При входе в систему она запрашивает у пользователя его имя, а для подтверждения того, что это имя ввёл его владелец, система запрашивает пароль. Имя выдаётся пользователю при регистрации, пароль пользователь устанавливает сам.

При создании пароля желательно придерживаться следующих правил:

* пароль должен содержать комбинацию букв и цифр или специальных знаков, пароль не может содержать пробелы;
* пароли должны часто меняться.

Управление доступом к данным осуществляется через СУБД, которая и обеспечивает защиту данных. Но такие данные вне СУБД становятся общедоступны. Если известен формат БД, можно осуществить к ней доступ с помощью другой программы (СУБД), и никакие ограничения при этом не помешают. Для таких случаев предусмотрено кодирование данных. Используются различные методы кодирования: перекомпоновка символов в кортеже, замена одних символов (групп символов) другими символами (группами символов) и т.д. Кодирование может быть применено не ко всему кортежу, а только к ключевым полям. Декодирование производится непосредственно в процессе обработки, что, естественно, увеличивает время доступа к данным. Поэтому к кодированию прибегают только в случае высоких требований к конфиденциальности данных.

**ЗАКЛЮЧЕНИЕ**

В рамках курсового проекта было создано приложение «Аптека» в связке с БД, развернутой на MS SQL 2008. В процессе разработки была использована система управления базами данных Open Server, язык программирования php. Разработанный программный продукт соответствует всем функциональным и эксплуатационным требованиям технического задания.

Проведен анализ предметной области. Выделены следующие объекты, необходимые для описания «Аптекой»: Должность, Лекарства, Сотрудники, Покупатели, Заказы, Аптеки, Обслуживающая аптека. Для каждого объекта выделены и занесены в БД его характеристики.

БД для предметной области «Аптека» разработана в среде Open Server. В нее занесены все объекты предметной области.

Также изучены особенности среды программирования и языка программирования php. Сделаны выводы о том, что данный язык и среда программирования полностью подходят для достижения поставленной цели. Для разработки клиентского приложения выбран и использован шаблон Windows Forms.

В результате реализованы следующие функции для работы пользователей: выбор таблиц; просмотр таблиц; изменение, добавление данных. Организовано подключение разработанного клиентского приложения к БД.

Проведено тестирование и отладка программного обеспечения. В проект добавлена защита от несанкционированного доступа, а также от некорректного ввода данных.

На последних циклах тестирования ошибок не обнаружено.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1) Галочкин, В.И. Базы данных Учебное пособие / В.И. Галочкин. – Йошкар-Ола: МарГТУ, 2009. – 200 с.

2) Дейт, К. Введение в системы баз данных / К. Дейт. – 8-е изд. – М.: Вильямс, 2006. – 1072 с.

3)  Форта, Б. Освой самостоятельно SQL. 10 минут на урок / Б. Форта – М.: Вильямс, 2006. – 288 с.

4)  Хомоненко А.Д. Работа с базами данных в Delphi / А.Д. Хомоненко, В.Э.Гофман – 7-е изд. – СПб.: БХВ-Перербург, 2005. – 640 с.

5)  Фаронов В.В. Программирование баз данных в Delphi 7 / В. В. Фаронов. – СПб: Питер, 2003. – 458 с.

6) Петкович Д. Microsoft SQL Server 2008. Руководство для начинающих. - Спб.: БХВ-Петербург, 2009 - 884 с.

7) Зрюмов Е.А., Зрюмова А.Г. Базы данных для инженеров. - Барнаул: АлтГТУ, 2010. - 131 с.

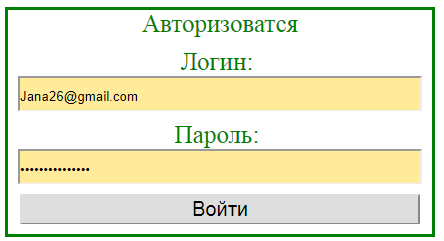
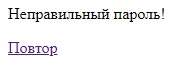
Ссылки на Internet-источники:

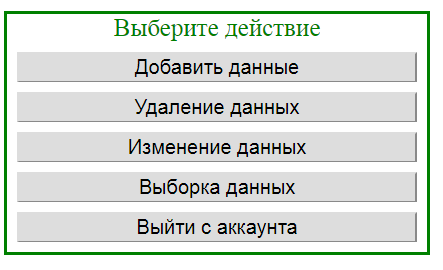
8)  http://citforum.ru/database/.

9) http://www.ict.edu.ru/.

10)  <http://archae-dev.com/>.

**ПРИЛОЖЕНИЕ**

1. При каждом запуске приложения необходимо авторизоваться. Ввести логин и пароль в специальные поля, представленные на форме.  
   
2. После нажатия на кнопку «Войти» производится проверка существует ли пользователь с ведённой связкой логин-пароль. Если не удаётся найти, то выводится соответствующее сообщение об ошибке.  
   
3. Если же удалось найти зарегистрированного пользователя в таблице в БД и этому пользователю отведена роль «Правообладатель», тогда в этом случае открывается форма, в которой реализован весь необходимый функционал для пользователя.

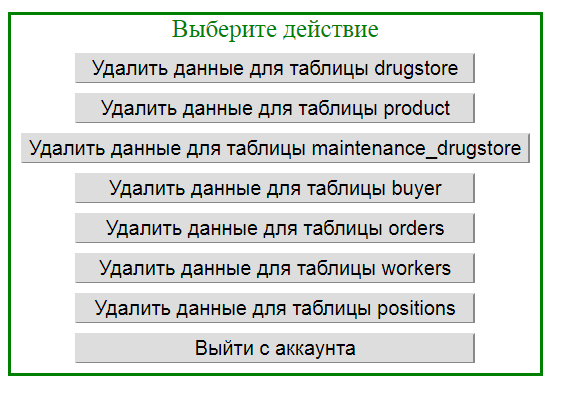


1. Далее «Правообладатель» может:

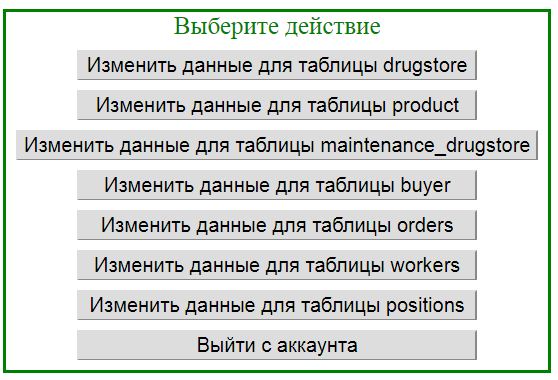
А) Добавить данные:



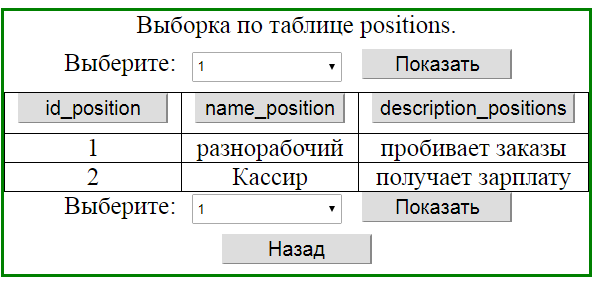
Б) Удалить данные:



В) Изменить данные:



Г) Произвести выборку данных из ключевых таблиц по запросам:



Данная возможность дает сделать выбоку из всех существующих таблиц, а также сортировка по столбцам.

**SQL запрос для создания таблиц и формирование связей**



