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

Отчет к курсовому проекту

по дисциплине «Безопасность систем баз данных»

**Разработка базы данных книжного магазина**

Выполнили: студенты группы БИ-31

Анисимов Д.В., Иванов С.В., Бровкин А.В.

Проверил: доцент кафедры

ИБ Сучков Д.С.

Йошкар-Ола

2019 г.

**СОДЕРЖАНИЕ**

**Введение**3

**1. Техническое задание4**

1.1 Требования к курсовой работе4

1.2 Требования к базе данных4

1.3 Требования к API (минимальное количество реализованных методов)4

**2. Порядок выполнения работы5**

2.1 Этапы разработки базы данных5-8

2.2 Этапы разработки API8

2.3Скриншоты программы8-10

**3. Приложения11**

3.1 ER-диаграмма11

3.2 Ссылка на github.com11

**4. Вывод11**

**Введение:** В курсовой работе рассматривается создание базы данных, предназначенной для работы книжного магазина. База данных позволяет клиентам просматривать ассортимент товаров, изменять свои данные. Также администратору взаимодействовать с базой пользователей и ассортиментом.

1. **Техническое задание**

***1.1 Требования к курсовой работе:***

* Получить структуру данных из файла, согласно варианту. Привести к 3й нормальной форме. Добавить недостающие таблицы.
* Составить ER-диаграмму
* Разработать API для базы данных на любом языке, выполняющемся на стороне сервера (php, ASP.NET, Java, python, node.js, etc)
* Взаимодействие должно осуществляться по клиент-серверной архитектуре, подключение с клиентской программы недопустимо
* Провести настройку пользователей базы данных для разграничения прав доступа, привести пример конфигурации
* Все документы и исходные коды для курсовой работы должны храниться под контролем системы контроля версий — git или mercurial (https://github.com/, https://bitbucket.org/)
* Во время сдачи курсового проекта необходимо предоставить отчет о проделанной работе в печатном виде (отчет)

***1.2 Требования к базе данных***

* Наличие не менее 7 таблиц, в том числе таблицы сессий и пользователей
* Структура таблицы должна содержать не менее 3-х полей, одно из которых ключевое
* Правомерное использование типов данных
* Обязательно использование триггеров и/или хранимых процедур
* Форма нормализации не менее 3NF
* Индексирование по полям поиска

**1.3 Требование к API (минимальное количество реализованных методов)**

* аутентификация пользователя (создание сессии);
* добавление/удаление/изменение данных в таблицах;
* выборка данных их ключевых таблиц по запросам;
* выборка данных из таблиц с объединением результатов.

1. **Порядок выполнения работы**
   1. Этапы разработки базы данных.

Разработана база данных, содержащая 7 таблиц. В том числе таблицы сессий (*session*) и пользователей ([*people\_table*](http://localhost:8888/phpMyAdmin/sql.php?server=1&db=db_shop&table=people_table&pos=0)). Для авторизации пользователя используется таблица – people\_table, содержащая информацию об аккаунте пользователя. Для хранения информации об ассортименте используются следующие таблицы:

*table\_products* – название, цена, обложка, жанр и описание товара(книг);

*table\_products\_info* – издательство, год выпуска товара;

Структуры, реализованных таблиц:

Таблица [*people\_table*](http://localhost:8888/phpMyAdmin/sql.php?server=1&db=db_shop&table=people_table&pos=0):

* Структура:

1. **CREATE** **TABLE** `people\_table` (
2. `user\_id` **varchar**(50) NOT NULL,
3. `username` **varchar**(255) NOT NULL,
4. `email` **varchar**(255) NOT NULL,
5. `**password**` **varchar**(255) NOT NULL,
6. `fio` **varchar**(255) NOT NULL,
7. `address` **varchar**(255) **DEFAULT** 'null',
8. `registration` **timestamp** NOT NULL **DEFAULT** CURRENT\_TIMESTAMP **ON** **UPDATE** CURRENT\_TIMESTAMP,
9. `role` **varchar**(5) NOT NULL
10. ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

* Индексы:

1. **ALTER** **TABLE** `people\_table`
2. **ADD** **PRIMARY** **KEY** (`user\_id`);
3. **COMMIT**;

Таблица *more\_people\_info*:

* Структура:

1. **CREATE** **TABLE** `more\_people\_info` (
2. `user\_id` **varchar**(50) NOT NULL,
3. `birthday\_date` **date** **DEFAULT** '1970-01-01',
4. `about` **varchar**(255) **DEFAULT** 'null',
5. `phone\_number` **varchar**(255) **DEFAULT** 'null',
6. `company` **varchar**(255) **DEFAULT** 'null'
7. ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

* Индексы:

1. **ALTER** **TABLE** `more\_people\_info`
2. **ADD** **PRIMARY** **KEY** (`user\_id`);
3. **COMMIT**;

Таблица *yet\_people\_info*:

* Структура:

1. **CREATE** **TABLE** `yet\_people\_info` (
2. `user\_id` **varchar**(50) NOT NULL,
3. `card\_number` **varchar**(255) **DEFAULT** 'null',
4. `amount` **float** **DEFAULT** '0'
5. ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

* Индексы:

1. **ALTER** **TABLE** `yet\_people\_info`
2. **ADD** **PRIMARY** **KEY** (`user\_id`);
3. **COMMIT**;

Таблица *session*:

* Структура:

1. **CREATE** **TABLE** `session` (
2. `id` **varchar**(50) NOT NULL,
3. `token` **varchar**(10) NOT NULL,
4. `date\_time` **timestamp** NULL **DEFAULT** CURRENT\_TIMESTAMP
5. ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

* Индексы:

1. **ALTER** **TABLE** `session`
2. **ADD** **PRIMARY** **KEY** (`id`);
3. **COMMIT**;

Таблица *table\_products*:

* Структура:

1. **CREATE** **TABLE** `table\_products` (
2. `product\_id` **int**(11) NOT NULL,
3. `product\_name` **varchar**(255) NOT NULL,
4. `price` **float** NOT NULL,
5. `image` **varchar**(255) NOT NULL,
6. `description` text NOT NULL,
7. `genre` **varchar**(255) NOT NULL
8. ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

* Индексы:

1. **ALTER** **TABLE** `table\_products`
2. **ADD** **PRIMARY** **KEY** (`product\_id`);

* Автоинкрементирование:

1. **ALTER** **TABLE** `table\_products`
2. **MODIFY** `product\_id` **int**(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=4;
3. **COMMIT**;

Таблица *table\_products\_info*:

* Структура:

1. **CREATE** **TABLE** `table\_products\_info` (
2. `product\_id` **int**(11) NOT NULL,
3. `edition` **varchar**(255) **DEFAULT** NULL,
4. `year` **int**(11) NOT NULL
5. ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

* Индексы:

1. **ALTER** **TABLE** `table\_products\_info`
2. **ADD** **PRIMARY** **KEY** (`product\_id`);

* Внешние ключи:

1. **ALTER** **TABLE** `table\_products\_info`
2. **ADD** **CONSTRAINT** `table\_products\_info\_ibfk\_1` **FOREIGN** **KEY** (`product\_id`) **REFERENCES** `table\_products` (`product\_id`) **ON** **DELETE** **CASCADE**;
3. **COMMIT**;

Таблица *warehouse*:

* Структура:

1. **CREATE** **TABLE** `warehouse` (
2. `warehouse\_id` **int**(11) NOT NULL,
3. `warehouse\_name` **varchar**(255) NOT NULL,
4. `product\_id` **int**(11) NOT NULL,
5. `count\_product` **int**(11) NOT NULL
6. ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;

* Индексы:

1. **ALTER** **TABLE** `warehouse`
2. **ADD** **PRIMARY** **KEY** (`warehouse\_id`),
3. **ADD** **KEY** `warehouse\_ibfk\_1` (`product\_id`);

* Внешние ключи:

1. **ALTER** **TABLE** `warehouse`
2. **ADD** **CONSTRAINT** `warehouse\_ibfk\_1` **FOREIGN** **KEY** (`product\_id`) **REFERENCES** `table\_products` (`product\_id`) **ON** **DELETE** **CASCADE**;
3. **COMMIT**;

* Автоинкрементирование:

1. **ALTER** **TABLE** `warehouse`
2. **MODIFY** `warehouse\_id` **int**(11) NOT NULL AUTO\_INCREMENT, AUTO\_INCREMENT=8;

Используемые триггеры:

* *delete\_user\_into\_more\_people\_info –* триггер, который удаляет информацию о пользователе из таблицы more\_people\_info.

1. **BEGIN**
2. **DELETE** **FROM** more\_people\_info **WHERE** user\_id = OLD.user\_id;
3. **END**

* *delete\_user\_yet\_people\_info –* триггер, который удаляет информацию о пользователе из таблицы yet\_people\_info

1. **BEGIN**
2. **DELETE** **FROM** yet\_people\_info **WHERE** user\_id = OLD.user\_id;
3. **END**

Проведена настройка пользователей базы данных для разграничения прав доступа:

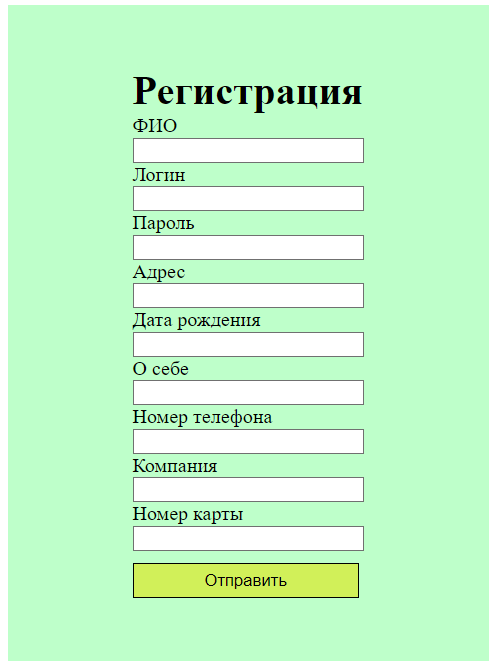


*Рис 1. Пользователи базы данных*

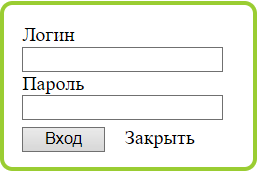
2.2. Этапы разработки API

Было разработано API для аутентификации и регистрации пользователей и взаимодействия с базой данных. API было разработано на языке PHP.

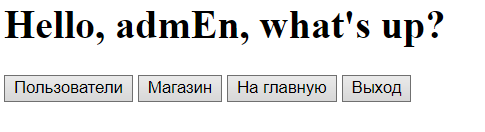
2.3. Скриншоты



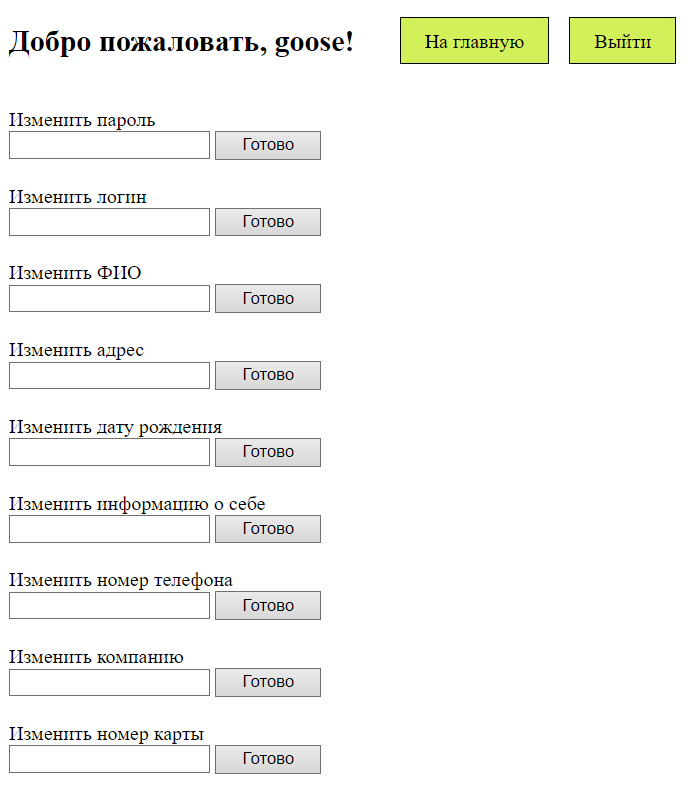
*Рис 2. Регистрация*

****

*Рис3 . Авторизаиця*

****

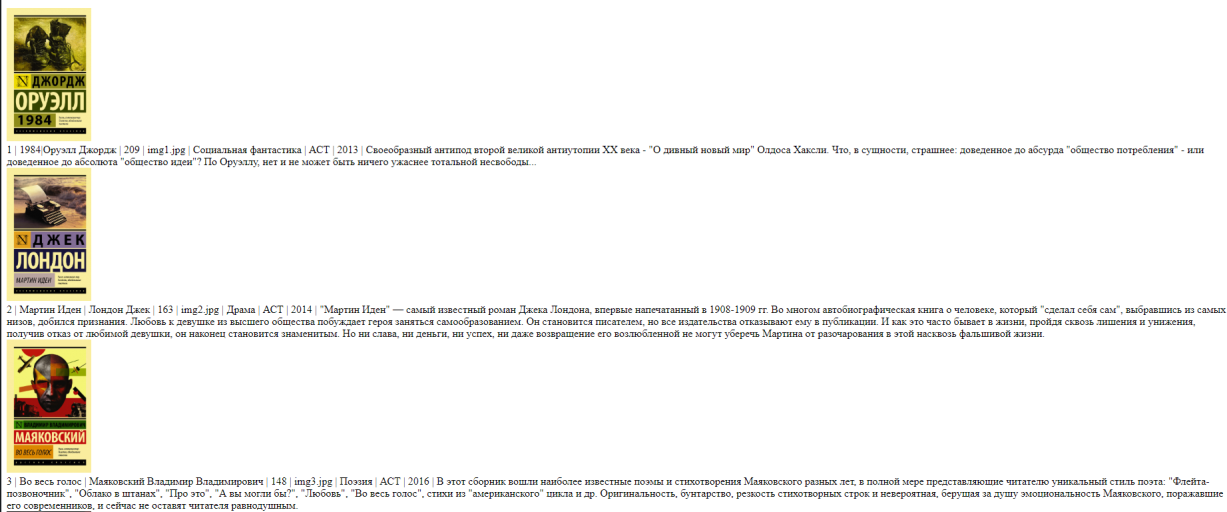
*Рис 4. Главное меню(администратор)*

**

*Рис 5. Главное меню(пользователь)*

**

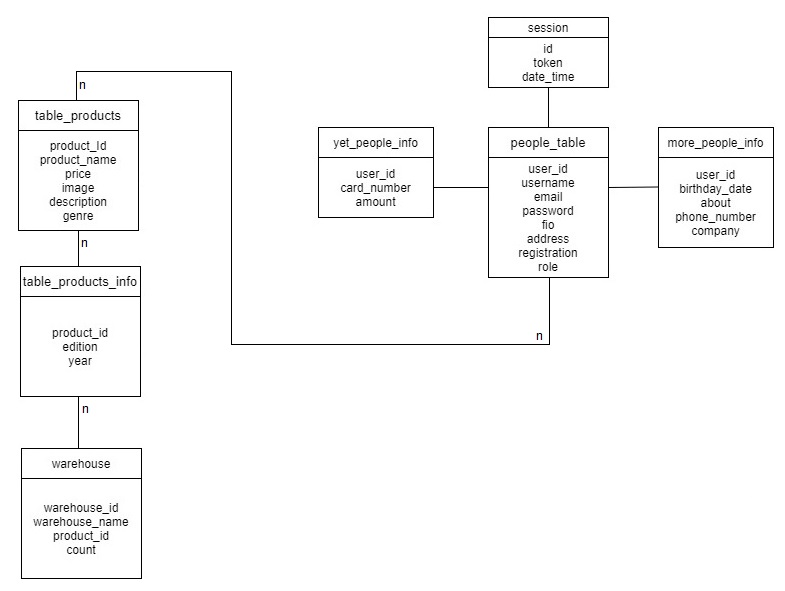
*Рис 6. Таблица пользователей, доступная администратору*

**

*Рис 7. Таблица товаров, доступная администратору*

**Приложения**

1. ER-диаграмма



База данных находится в первой нормальной форме, т.к. в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов. Т.к. база данных находится в 1НФ, то она также находится во второй нормальной форме, потому что каждый неключевой атрибут неприводимо (функционально полно) зависит от её потенциального ключа. Наличие 2НФ и отсутствие зависимости неключевых атрибутов от ключевых доказывает, что база данных находится в третьей нормальной форме.

1. Исходные коды и документы:

https://github.com/anisimovDanil/database\_system\_security

**Вывод:** При выполнении курсового проекта были изучены и освоены методы работы с базами данных, способы управления базами данных. В ходе работы была разработана база данных, а также разные интерфейсы, для работы книжного магазина.