ОБРАЗОВАТЕЛЬНЫЙ ПОРТАЛ «GEEK UNIVERSITY»

Факультет Python-разработки

КУРСОВОЙ ПРОЕКТ

по дисциплине: «Базы данных»

на тему: «Проектирование структуры базы данных информационной системы

для интернет-магазина «KW-shop»».

|  |  |
| --- | --- |
|  | Выполнил:  Студент курса Python-разработки  Сергей Солдатов |
|  | Проверил:  Преподаватель  Кирилл Иванов |

Москва, 2019

Проектирование структуры базы данных информационной системы

для интернет-магазина «KW-shop».

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Версия 2.0

Оглавление

[1. Требования к курсовому проекту 4](#_Toc27527150)

[2. Используемые инструменты 5](#_Toc27527151)

[3. Общие сведения о системе 6](#_Toc27527152)

[4. Описание сущностей 7](#_Toc27527153)

[4.1 Пользователь. 7](#_Toc27527154)

[4.2 Товар 7](#_Toc27527155)

[4.3 Прайс-лист 7](#_Toc27527156)

[4.4 Склад 7](#_Toc27527157)

[4.5 Корзина 7](#_Toc27527158)

[4.6 Заказ 7](#_Toc27527159)

[4.7 Отзывы 8](#_Toc27527160)

[4.8 Маркетинговая кампания 8](#_Toc27527161)

[5. Описание функционала 9](#_Toc27527162)

[5.1 Работа с номенклатурой. 9](#_Toc27527163)

[5.2 Работа с прайс-листами. 9](#_Toc27527164)

[5.3 Работа с пользователями. 9](#_Toc27527165)

[5.4 Работа с заказами. 9](#_Toc27527166)

[5.5 Работа со складом. 9](#_Toc27527167)

[5.6 Работа с маркетинговой кампанией. 9](#_Toc27527168)

[5.7 Работа со складом. 9](#_Toc27527169)

[5.8 Прочие функции. 9](#_Toc27527170)

[6. Скрипты характерных выборок. 10](#_Toc27527171)

[7. Представления. 11](#_Toc27527172)

[8. Структура таблиц и связей 12](#_Toc27527173)

[9. Листинг 13](#_Toc27527174)

[10. Заключение 14](#_Toc27527175)

# Требования к курсовому проекту

1. Составить общее текстовое описание БД и решаемых ею задач;
2. Минимальное количество таблиц - 10;
3. Скрипты создания структуры БД (с первичными ключами, индексами, внешними ключами);
4. Создать ERDiagram для БД;
5. Скрипты наполнения БД данными;
6. Скрипты характерных выборок (включающие группировки, JOIN'ы, вложенные таблицы);
7. Представления (минимум 2);
8. хранимые процедуры / триггеры;+

# Используемые инструменты

В процессе работы над курсовым проектом использовались следующие инструментальные средства:

* Средство виртуализации – Vmware Workstation 15 Player;
* ОС под СУБД –Linux Ubuntu 18.04.3 LTS Bionic (64-bit);
* СУБД (сервер) – mySQL (v. 5.7.28-0ubuntu0.18.04.4);
* Клиент БД – DBeaver v. 6.1.5.201908191349
* Средство генерации тестовых данных - <http://filldb.info/>

# Общие сведения о системе

Проектируемая база данных предназначена для перманентного хранения информации, необходимой для функционирования информационной системы Интернет-магазин «KW-shop», целью которого является онлайн-продажи вязанных товаров нескольких категорий (игрушки, вещи, аксессуары). Каждая категория товаров может содержать подкатегории, внутри которых группируются конечные товары (товарные позиции).

Пользователь базы данных (посетитель сайта) может посещать интернет-магазин и просматривать каталог товаров без необходимости регистрации. При попытке пользователя разместить первый заказ, ему будет предложено пройти процедуру авторизации и аутентификации. В случае отсутствия у пользователя учетной записи система предложит пройти регистрацию, в ходе которой необходимо заполнить следующую информацию: № мобильного телефона (обязательно), e-mail (обязательно), имя (обязательно), фамилия (по желанию), адрес проживания (по желанию), пол (обязательно). Данная информация будет храниться в профиле пользователя, который можно редактировать при посещении сайта после авторизации.

Более подробно функциональную часть проектируемой системы будет описана в разделе «Описание функционала».

Структура создаваемой базы данных должна поддерживать следующие укрупненные сущности:

* Пользователь (включая его профиль);
* Товарная единица (товар);
* Каталог товаров;
* Прайс-лист;
* Склад (перечень готовой продукции);
* Пользовательская корзина;
* Заказ;
* Отзывы;
* Маркетинговая кампания (акции, скидки, спец. предложения);
* Накладная;
* Проводки (оплаты);
* Фотогалерея.

Также структура разрабатываемой базы данных должна обеспечивать предоставление следующего функционала:

* Управление пользователями;
* Управление каталогом товаров;
* Управление товарными единицами;
* Управление заказом (формирование, жизненный цикл);
* Работа с корзиной товаров;
* Управление маркетинговыми кампаниями;
* Аналитика пользовательской активности (хранение истории заказов, посещений, просмотренных и отмеченных товаров);
* Управление фотоконтентом;
* Управление отзывами.

# Описание сущностей

С точки зрения архитектуры базы данных в системе присутствует 2 основные (или центральные) сущности: пользователь и товар. Остальные сущности и функционал системы предназначены для достижения основной цели проектируемой информационной системы – соединить две основные сущности через товарную корзину для последующего формирования и размещения заказа.

## Пользователь.

Структура данных Пользователя должна хранить всю информацию, запрашиваемую при регистрации. При этом часть информации обязательна для заполнения (поля: имя, пол, номер мобильного, e-mail), а часть – факультативна (фамилия, адрес, возраст, день рождения). Сущность представлена двумя таблей users. В ней также отражается статус пользователя (зарегистрирован, активен – совершил хотя бы одну покупку, подключен к системе, удален).

## Товар

Сущность «Товар» должна содержать исчерпывающую информацию о товаре – наименование, подкатегория, категория (опосредованно через сущность «Каталог», размер, ссылка на фото, описание. Некоторые параметры Товара могут храниться в других справочниках (например, информация о стоимости содержится в сущности «Прайс-лист»). Сущность «товар» (таблица items) состоит в тесной связи с сущностью «Каталог», представленной в базе данных двумя табличными объектами: subcategories и categories.

## Прайс-лист

Сущность «Прайс-лист» ассоциирует товарные позиции с их ценой. Предполагается, что для одной и той же позиции, в зависимости от различных факторов (география, возраст пользователя, прочее) возможно единовременное наличие нескольких действующих прайс-листов для одного и того же товара. По этой причине сущность представлена таблицами prices и pricelist, реализующими связь «один-ко-многим» товара с его ценой.

## Склад

Ввиду того, что на начальном этапе магазин предполагает продажу товаров на заказ, однако в будущем подразумевается продажа готовых позиций, на начальном этапе структура сущности «Склад» закладывается в минимальной наполненности, в которой будет хранится лишь количества товаров: заказанных, доступных к поставке, зарезервированных для заказа (до внесения оплаты).

## Корзина

Корзина товаров, в которую попадают выбранные пользователем товарные позиции, имеет короткое время жизни от добавления первого товара до оформления Заказа или сброса корзины. Однако, с целью сохранения выбранных товаров в Корзине после отключения от базы данных (разрыв соединения, выход из браузера и т.п.) без завершения операции над корзиной, под эту сущность создается отдельная таблица, в которой будут перманентно храниться товары, выбранные пользователем вплоть до одного из двух явных событий – оформление заказа или сброс Корзины. При этом для одного пользователя может существовать не более одной Корзины.

Сущность «Корзина» также является ключевой в структуре базы данных, так как является точкой соединения двух основных сущностей: «Пользователь» и «Товар».

## Заказ

После заполнения корзины в случае подтверждения покупки формируется с помощью соответствующей сущности формируется Заказ, в который должны попасть все значимые данные о покупке, включая товар, его количество, стоимость, дату размещения. Созданные заказы из системы не удаляются, а проходят из одного статуса в другой. Доступные статусы: размещен, оплачен, в производстве, готов к отправке, отправлен, доставлен, исполнен, отменен. Также для отслеживания истории заказа, сохраняются все статусы заказа под одним номером (id) c различными датами, по которым можно определить период актуальности соответствующей записи: start\_date и end\_date. Первичный ключ формируется посредством картежа: <id, start\_date>.

Таблицы в настоящей базе данных, обладающие описанной характеристикой, далее будут называться темпоральными.

Для удобства управления статусами Заказа (добавление, изменение) в процессе эволюции разрабатываемой структуры базы данных, в рамках сущности выделена отдельная таблица для хранения статусов.

## Отзывы

В целях продвижения самых популярных товаров зарегистрированные пользователи могут оставлять комментарии к заказанным ими товарам с возможностью прикрепления фотографии. Структура, отражающая сущность «Отзывы» должна поля с текстом отзыва и ссылкой на фотографию товара, а также автора отзыва.

## Маркетинговая кампания

В целях гибкого регулирования цен на товары структурой базы данных предусмотрена дополнительная таблица discounts, реализующая сущность «Маркетинговая кампания». В данную таблицу предполагается вносить записи, отражающие уровень скидок, действующий определенный период времени и распространяющиеся на тот или иной товар, подкатегорию, категорию или пользователя.

Скидки применимы только по отношению к определенному прайслисту (внешний ключ на pricelists.id). Логика формирования цены – берется цена товара из прайс-листа, осуществляется поиск в таблице discounts в действующем диапазоне дат на текущего пользователя, товар, подкатегорию, категорию, и, в случае наличия одной или нескольких записей со скидками, применятся максимальная. Скидки не суммируются.

# Описание функционала

Данный раздел включает описание созданных сущностей СУБД MySQL, реализующих функционал системы, таких как хранимые процедуры, хранимые функции, триггеры.

Бизнес-функции системы реализованы в виде хранимых процедур, функций и триггеров, изменяющих данные таблиц. Предполагается, что напрямую из приложения SQL-код, модифицирующий содержание таблиц вызываться не будет.

Исходные коды всех описанных в данном разделе структур приведены в файле kw\_stored.sql.

## Работа с номенклатурой.

* sp\_add\_cat – хранимая процедура, добавить категорию товара;
* sp\_edit\_cat – хранимая процедура, изменить категорию товара;
* sp\_del\_cat – хранимая процедура, удалить категорию товара;
* sp\_add\_subcat – хранимая процедура, добавить категорию товара;
* sp\_edit\_subcat – хранимая процедура, изменить категорию товара;
* sp\_del\_subcat – хранимая процедура, удалить категорию товара;
* sp\_add\_item – хранимая процедура, добавить товарную позицию;
* sp\_edit\_item – хранимая процедура, изменить товарную позицию;
* sp\_del\_item – хранимая процедура, удалить товарную позицию;

## Работа с прайс-листами.

Предстоит разработать позднее

## Работа с пользователями.

Предстоит разработать позднее

## Работа с заказами.

* sp\_add\_order – хранимая процедура, добавить заказ;
* sp\_change\_order\_status – хранимая процедура, изменяющая статус заказа;
* sp\_edit\_order - хранимая процедура, предстоит разработать позднее;
* sp\_del\_order - хранимая процедура, предстоит разработать позднее;
* sf\_calc\_order\_sum – хранимая функция, расчет суммы заказа. Используется триггерами, инициированными при изменении товарных позиций.
* sp\_add\_order\_item – хранимая процедура, предстоит разработать позднее;
* sp\_edit\_order\_item – хранимая процедура, предстоит разработать позднее;
* sp\_del\_order\_item - хранимая процедура, предстоит разработать позднее;

## Работа со складом.

Предстоит разработать позднее

## Работа с маркетинговой кампанией.

Предстоит разработать позднее

## Работа со складом.

Предстоит разработать позднее

## Прочие функции.

Предстоит разработать позднее

# Скрипты характерных выборок.

В данном разделе описаны cкрипты характерных выборок, предназначенные для формирования отчетных данных.

Характерными выборками в магазине могут быть такие запросы:

* Вернуть список всех товаров заданной подкатегории;
* Вернуть список всех категорий заданной подкатегории;
* Вернуть список цен всех товаров (прайс-лист);
* Вернуть список всех размещенных, но не исполненных заказов с указанием товарных позиций;
* Вернуть список топовых пользователей (топ 10) с максимальным количеством покупок и совокупной суммой покупок

# Представления.

В данном разделе описаны используемые для частых запросов представления, предназначенные для ускорения работы системы.

**Представление 1.** Таблица заказов в актуальном состоянии. Так как таблица orders, описанная в разделе 3, является темпоральной, т.е. все изменения каждого заказа сохраняются в рамках периода времени, ограниченного значениями столбцов begin\_date и end\_date, то для быстрого просмотра актуального состояния заказов целесообразно создать представление v\_orders, которое фильтрует таблицу orders по полю end\_date. Если оно NULL, значит запись по данному order\_id последняя и актуальная.

**Представление 2.** Представление most\_ordered\_view, содержащее «хиты продаж» - топ 5 позиций каталога, которые заказывались наибольшее количество раз, причем статус заказа должен быть не ниже 'оплачен’.

# Структура таблиц и связей

Базовая структура базы данных, описанная в работе состоит из 14 таблиц, использованы ограничения целостности на внешних ключах, описаны индексы полей таблиц с ожидаемо большим количеством записей, к которым предполагается частое обращение.

Структура таблиц и отражение связей проектируемой базы данных и ограничений целостности представлены на ER-диаграмме (рис.1).

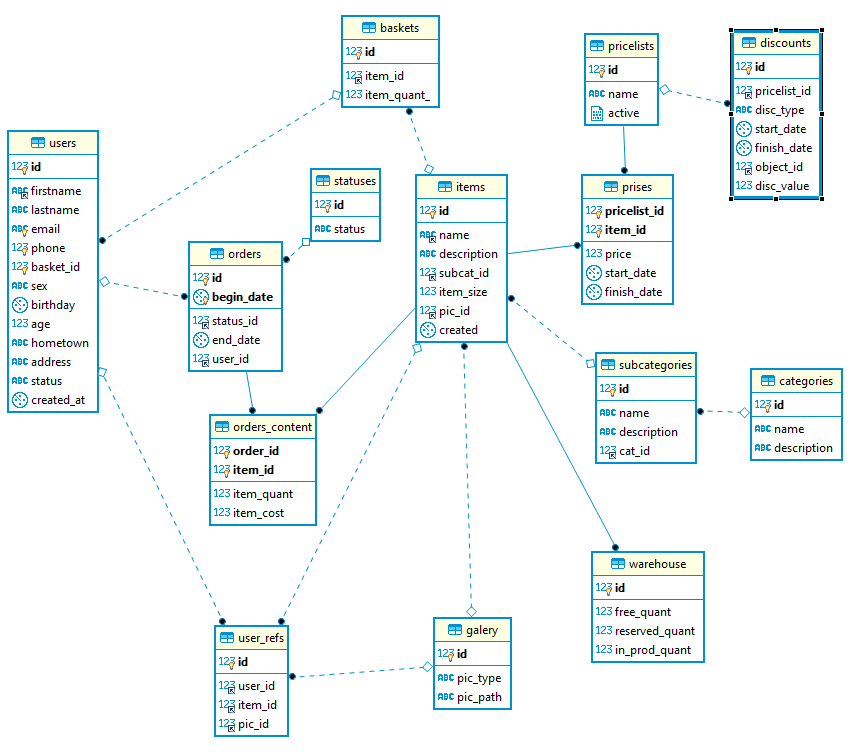


Рисунок 1. ER-диаграмма проектируемой базы данных

# Листинг

Листинг кода с комментариями, с помощью которого создается описанная выше база данных, представлен в следующих файлах:

* Листинг кода создания БД - KW\_shop.sql;
* Листинг кода представлений и характерных выборок - KW\_shop.sql (выборки возвращают значения после запуска скрипта наполнения БД данными);
* Листинг кода наполнения БД данными – kw\_fill\_in\_output.sql;
* Листинг кода хранимых процедур, функций и триггров – kw\_stored.sql

# Заключение

Ввиду того, что создаваемая в рамках курсового проекта БД является прототипом реальной системы для создания сайта, а также с учетом ограниченного времени, объем разработки соответствует требованиям курсового проекта и не является на момент сдачи работы законченным проектом.

Предполагается его постоянная доработка в рамках дальнейшей учебы на курсе Python-разработки с целью выйти на законченный проект и запуска его в промышленную эксплуатацию.

По этой причине существенная часть функционала пока не доработана, а разработанные функции представлены в упрощенном варианте, требующем итерационных доработок в процессе дальнейшей разработки.