ОБРАЗОВАТЕЛЬНЫЙ ПОРТАЛ «GEEK UNIVERSITY»

Факультет Python-разработки

КУРСОВАЯ РАБОТА

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 1.0

Оглавление

[1. Общие сведения 1](#_Toc24534565)

[2. Описание сущностей 2](#_Toc24534566)

[2.1 Пользователь. 2](#_Toc24534567)

[2.2 Товар 2](#_Toc24534568)

[2.3 Прайс-лист 3](#_Toc24534569)

[2.4 Склад 3](#_Toc24534570)

[2.5 Корзина 3](#_Toc24534571)

[2.6 Заказ 3](#_Toc24534572)

[2.7 Отзывы 3](#_Toc24534573)

[2.8 Маркетинговая кампания 4](#_Toc24534574)

[2.9 «Лайки» 4](#_Toc24534575)

[3. Описание функционала 4](#_Toc24534576)

[4. Структура таблиц и связей 4](#_Toc24534577)

[5. Листинг 4](#_Toc24534578)

# Общие сведения

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).

Любой пользователь 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 предполагается описать в разделе «Описание функционала» на более поздних стадиях проектирования.

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);
* Оценка товарных единиц (пользовательские лайки);
* 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);
* Оценка товарных единиц пользователями (лайки);
* 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 таблицами: users и user\_profiles

## 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.

## «Лайки»

Еще одним способом повышения привлекательности популярных товарных единиц в системе являются «лайки» (пользовательские отметки «нравится» на товарные позиции). Для реализации этой сущности в структуре базе данных предусмотрена таблица likes, связывающая пользователя с понравившимся товаром.

# Описание функционала

Описание функционала выходит за рамки настоящей работы и будет разработано на стадии проектирования бизнес-логики системы.

# Структура таблиц и связей

Базовая структура базы данных, описанная в работе состоит из 15 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).

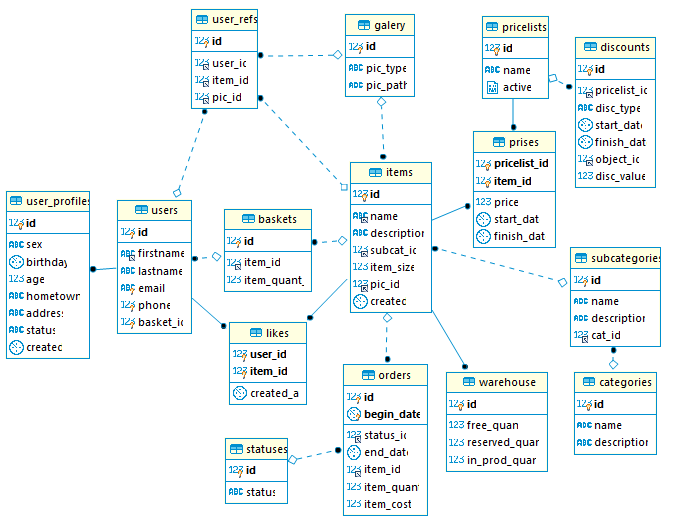


Рисунок 1. ER-диаграмма проектируемой базы данных

# Листинг

Листинг кода с комментариями, с помощью которого создается описанная выше база данных, представлен ниже.

/\*создание базы данных KW-shop - БД для харнения информации сайта онлайн

\* продаж вязаных товаров Knitted World \*/

**DROP** **DATABASE** **IF** **EXISTS** KW\_shop;

**CREATE** **DATABASE** **IF** **NOT** **EXISTS** KW\_shop;

**USE** KW\_shop;

/\*создание таблицы основной сущности "Пользователи"

структура данных должна обеспечивать авторизацию, аутентификацию, хранение информации

о пользователе для возможности исполнения заказов, а также накопления информации об

истории его активности для формирования в будущем маркетинговых кампаний

\*/

/\* Основная таблица сущности "Пользователи" - users - содержит поля, необходимые для

\* авторизации, аутентификации зарегистрированных пользователей

\*\*/

**DROP** **TABLE** **IF** **EXISTS** users;

**CREATE** **TABLE** users (

id SERIAL **PRIMARY** **KEY**, -- SERIAL = BIGINT UNSIGNED NOT NULL AUTO\_INCREMENT UNIQUE

firstname **VARCHAR**(50) **NOT** **NULL**, -- имя

lastname **VARCHAR**(50), -- фамилия

email **VARCHAR**(120) **UNIQUE** **NOT** **NULL**, -- электронная почта

phone **BIGINT** **UNIQUE** **NOT** **NULL**, -- номер мобильного телефона

basket\_id **BIGINT** **UNSIGNED** **UNIQUE**, -- номер корзины. внешний ключ бюдет добавлен после создания соответствующей таблицы

**INDEX** users\_phone\_idx(phone), -- индекс по полю моб. теленфон

**INDEX** users\_firstname\_lastname\_idx(firstname, lastname), -- индекс по полям: имя, фамилия

**INDEX** users\_email\_idx(email) -- индекс по полю электронной почты

);

/\*создание дополнительной таблицы Profile основной сущности "Пользователи".

структура данных обеспечивает хранение информации о пользователе для возможности исполнения

заказов, а также накопления информации об истории его активности для формирования в будущем

маркетинговых кампаний

\*\*/

**DROP** **TABLE** **IF** **EXISTS** user\_profiles;

**CREATE** **TABLE** user\_profiles (

id **BIGINT** **UNSIGNED** **UNIQUE** **primary** **key**,

sex **ENUM** ('m', 'f'), -- пол

birthday **DATE**, -- дата рождения

age **SMALLINT** **UNSIGNED**, -- возраст

hometown **VARCHAR**(100), -- город

address **VARCHAR**(255), -- адрес

status **ENUM** ('registred', 'active', 'deleted'),

created **DATETIME** **DEFAULT** **NOW**(), -- временая метка (создание записи)

/\* Описание статусов

\* 'registred' - зарегистрированный пользователь

\* 'active' - активный пользователь, совершивший одну или несколько покупок

\* 'deleted' - пользователь, по каким-то причинам помеченный как удаленный (доступ к системе заблокирован)

\*/

**FOREIGN** **KEY** (id) **REFERENCES** users(id)

**ON** **UPDATE** **CASCADE** -- каскадное обновление

**ON** **DELETE** **RESTRICT** -- удаление запрещено

);

/\* создание второй ключевой сущности "Товар" и связанной структурной сущности "Каталог"

\* Каталог - это двухуровневая иерархия, позволяющая классифицировать товар следующим

\* образом:

\* -->Категория| (таблица categories)

\* --> Подкатегория|(таблица subcategories)

\* --> Товар (таблица items)

\*/

/\*

\* создание таблицы верхнего уровня иерархии каталога - categories

\* примеры катгорий:

\* одежда, игрушки, аксесуары

\* \*/

**DROP** **TABLE** **IF** **EXISTS** categories;

**CREATE** **TABLE** categories (

id SERIAL **primary** **key**,

name **varchar**(25) **NOT** **NULL**,

description **varchar**(255)

);

/\*

\* создание таблицы нижнего уровня иерархии каталога - subcategories

\* примеры подкатегорий (для категории "Игрушки"):

\* куклы, животные, мультгерои

\* \*/

**DROP** **TABLE** **IF** **EXISTS** subcategories;

**CREATE** **TABLE** subcategories (

id SERIAL **primary** **key**,

name **varchar**(25) **NOT** **NULL**,

description **varchar**(255),

cat\_id **BIGINT** **UNSIGNED** **NOT** **NULL**,

-- индексы

**index** subcat\_cat\_idx (cat\_id),

-- внешние ключи

**FOREIGN** **KEY** (cat\_id) **REFERENCES** categories(id)

**ON** **UPDATE** **CASCADE**

**ON** **DELETE** **RESTRICT**

);

/\*

\* создание таблицы, содержащей структуру второй ключевой сущности - items.

\* Содержит все связанные характеристики товара: размер, ссылку на фото,

\* идентификатор подкатегории

\* \*/

**DROP** **TABLE** **IF** **EXISTS** items;

**CREATE** **TABLE** items (

id SERIAL **primary** **key**,

name **varchar**(25) **NOT** **NULL**,

description **varchar**(255),

subcat\_id **BIGINT** **UNSIGNED** **NOT** **NULL**,

item\_size **TINYINT** **unsigned**,

pic\_id **BIGINT** **UNSIGNED** **NOT** **NULL**, /\* - ссылка на сущность galary, которая будет определена ниже

после описания таблицы galary, необходимо дополнить таблицу items ограничением целостности

с внешним ключом по полю pic\_ID

\*/

created **DATETIME** **DEFAULT** **NOW**(),

-- индексы

**index** item\_subcat\_idx (subcat\_id),

**index** item\_name\_idx (name),

**index** item\_pic\_idx (pic\_id),

-- внешние ключи

**FOREIGN** **KEY** (subcat\_id) **REFERENCES** subcategories(id)

**ON** **UPDATE** **CASCADE**

**ON** **DELETE** **RESTRICT**

);

/\*

\* создание сущности "Прайс-лист", состоящей из двух таблиц:

\* pricelists и prices

\*

\*/

/\* создание таблицы pricelists, содержащей наименование прайслистов

\* изначально предполагается использовать 1 прайслист - базовый, однако

\* архитектурно закладывается возможность использования нескольких прайс-

\* листов

\*/

**DROP** **TABLE** **IF** **EXISTS** pricelists;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** pricelists (

id SERIAL **primary** **key**,

name **varchar**(100),

active **BINARY**

);

/\*создание таблицы prices, содержащей позиционные цены всех товаров для каждого

\* активного прайслиста

\*

\*/

**DROP** **TABLE** **IF** **EXISTS** prises;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** prises (

pricelist\_id **BIGINT** **UNSIGNED** **NOT** **NULL**,

item\_id **BIGINT** **UNSIGNED** **NOT** **NULL**,

price **INT** **UNSIGNED** **NOT** **NULL**,

start\_date **DATETIME** **NOT** **NULL** **DEFAULT** **NOW**(),

finish\_date **DATETIME** **NOT** **NULL**,

-- ключи

**PRIMARY** **KEY** (pricelist\_id, item\_id),

**FOREIGN** **KEY** (item\_id) **REFERENCES** items(id),

**FOREIGN** **KEY** (pricelist\_id) **REFERENCES** pricelists(id),

-- индексы

**INDEX** prices\_item\_idx (item\_id)

);

/\* создание сущности "Склад"

\*

\* free\_quant - количество товара, доступное к поставке;

\* reserved\_quant - количество товара, зарезервированное для заказа;

\* in\_prod\_quant - количество товара в производстве.

\* отношение с таблицей items: 1-1

\*/

**DROP** **TABLE** **IF** **EXISTS** warehouse;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** warehouse(

id **BIGINT** **UNSIGNED** **UNIQUE** **primary** **key**,

free\_quant **INTEGER** **DEFAULT** 0,

reserved\_quant **INTEGER** **DEFAULT** 0,

in\_prod\_quant **INTEGER** **DEFAULT** 0,

-- внешний ключ

**FOREIGN** **KEY** (id) **REFERENCES** items(id)

**ON** **UPDATE** **CASCADE**

**ON** **DELETE** **RESTRICT**

);

/\* создание сущности "Корзина"

\* для одного пользователя может существовать не более одной

\* корзины.

\* отношение с таблицей users: 1-1

\*/

**DROP** **TABLE** **IF** **EXISTS** baskets;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** baskets(

id SERIAL **primary** **key**, -- связан с users(basket\_id)

item\_id **BIGINT** **UNSIGNED** **NOT** **NULL**,

item\_quant\_ **INT** **UNSIGNED** **NOT** **NULL**,

-- внешний ключ

**FOREIGN** **KEY** (item\_id) **REFERENCES** items(id)

);

/\*

\* на данном этапе необходимо добавить ограничение целостности в таблицу users

\* по ключу basket\_id

\*/

**ALTER** **TABLE** users **ADD** **CONSTRAINT** **FOREIGN** **KEY** users\_basket (basket\_id) **REFERENCES** baskets(id);

/\* создание сущности "Заказ"

\* состоит из таблиц statuses и orders

\*/

/\* создание таблицы statuses

\* доступные статусы: размещен, оплачен, в производстве, готов к отправке,

\* отправлен, доставлен, исполнен, отменен

\*/

**DROP** **TABLE** **IF** **EXISTS** statuses;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** statuses(

id SERIAL **primary** **key**,

status **varchar** (20)

);

/\* создание таблицы orders (темпоральная)

\* данные из таблице НЕ УДАЛЯЮТСЯ, манипуляции производятся со полями

\* status, begin\_date и end\_date

\*

\* Перечень статусов во избежание хардкодирования в случае их добавления

\* (изменения, удаления в процессе проектирования) вынесен в отдельную таблицу

\* (описана выше).

\*/

**DROP** **TABLE** **IF** **EXISTS** orders;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** orders(

id **BIGINT** **UNSIGNED** **NOT** **NULL**,

status\_id **BIGINT** **UNSIGNED** **NOT** **NULL**,

begin\_date **DATETIME** **NOT** **NULL** **DEFAULT** **NOW**(),

end\_date **DATETIME** **NOT** **NULL**,

item\_id **BIGINT** **UNSIGNED** **NOT** **NULL**, -- INDEX

item\_quant **INT** **UNSIGNED** **NOT** **NULL**,

item\_cost **INT** **UNSIGNED** **NOT** **NULL**,

-- ключи

**PRIMARY** **KEY** (id, begin\_date),

**FOREIGN** **KEY** (item\_ID) **REFERENCES** items(id),

**FOREIGN** **KEY** (status\_id) **REFERENCES** statuses(id),

-- индексы

**INDEX** orders\_item\_idx (item\_id)

);

/\* создание сущности "Отзыв"

\* создаваемая таблица должна хранить user\_id, текст отзыва,

\* item\_id, по которому составлен отзыв, а также ссылку на фото.

\* отзывы допустимо оставлять на приобретенные товары (не более одного

\* на каждый товар). Такую проверку необходимо органиховать на фазе

\* программирования бизнес-логики приложения

\*/

**DROP** **TABLE** **IF** **EXISTS** user\_refs;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** user\_refs(

id SERIAL **primary** **key**,

user\_id **BIGINT** **UNSIGNED** **NOT** **NULL**,

item\_id **BIGINT** **UNSIGNED** **DEFAULT** **NULL**,

pic\_id **BIGINT** **UNSIGNED** **DEFAULT** **NULL**, /\* - ссылка на сущность galary, которая будет определена ниже

после описания таблицы galary, необходимо дополнить таблицу items ограничением целостности

с внешним ключом по полю pic\_ID

\*/

-- ключи

**FOREIGN** **KEY** (user\_id) **REFERENCES** users(id),

**FOREIGN** **KEY** (item\_id) **REFERENCES** items(id),

-- индексы

**INDEX** ref\_user\_idx(user\_id),

**INDEX** ref\_item\_idx(item\_id),

**INDEX** ref\_pic\_idx(pic\_id)

);

/\* создание сущности "Маркетинговая кампания". Таблица discounts

\* Для поля Discounts допустимы значения:

\* 'item' - скидка на конкретную товарную позицию

\* 'subcat' - скидка на подкатегорию товара

\* 'cat' - скидка на подкатегорию товара

\* 'user' - скидка, предоставляемая конкретному пользователю

\* 'all' - скидка "всем на все"

\*/

**DROP** **TABLE** **IF** **EXISTS** discounts;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** discounts (

id SERIAL **primary** **key**,

pricelist\_id **BIGINT** **UNSIGNED** **NOT** **NULL**, -- скидки предоставляются только на определенный прайслист

disc\_type **ENUM** ('item','subcat','cat','user','all'),

start\_date **DATETIME**, -- дата начала действия скидок

finish\_date **DATETIME**, -- дата окончания действия скидок

object\_id **BIGINT** **UNSIGNED**, /\* объект предоставления скидок (идентификатор

пользователя/категории/подкатегории/товара)

если поле NULL - значит disc\_type = 'all'

\*/

disc\_value **TINYINT** **DEFAULT** 0, /\* уровень скидок (%). Представлен целым числом. Необходимо учесть при расчете

окончательной цены на товар при формировании заказа

\*/

-- внешние ключи

**FOREIGN** **KEY** (pricelist\_id) **REFERENCES** pricelists(id),

-- индексы

**INDEX** discount\_obj\_idx(object\_id),

**INDEX** discount\_pricelist\_idx(pricelist\_id)

);

/\* создание сущности "Лайки" (таблица 'likes'

\* соответствие пользователя и отмеченного товара с фикасацией

\* даты отметки

\*/

**DROP** **TABLE** **IF** **EXISTS** likes;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** likes (

user\_id **BIGINT** **UNSIGNED** **NOT** **NULL**,

item\_id **BIGINT** **UNSIGNED** **NOT** **NULL**,

created\_at **DATETIME** **DEFAULT** **NOW**(),

-- ключи

**primary** **key** (item\_id, user\_id),

**FOREIGN** **key** (item\_id) **REFERENCES** items(id),

**FOREIGN** **key** (user\_id) **REFERENCES** users(id),

-- индексы

**INDEX** likes\_item\_idx (item\_id),

**INDEX** likes\_user\_idx (user\_id)

);

/\* создание сущности "Галерея"

\* хранит таблицу с идентификатором фотографий и ссылками на их хранение

\* содержит поле pic\_type со значениями:

\* portfolio - для хранения нескольких картинок товарной позиции

\* reference - для хранения пользовательских фотографий для отзывов

\*/

**DROP** **TABLE** **IF** **EXISTS** galery;

**CREATE** **TABLE** **IF** **NOT** **EXISTS** galery(

id SERIAL **primary** **key**,

pic\_type **ENUM** ('profile','reference'),

pic\_path **varchar**(255)

);

/\* добавление необходимых ключей, ссылающихся на таблицу фотогалереи

\*

\* таблица items

\*/

**ALTER** **TABLE** items **ADD** **CONSTRAINT** **foreign** **key** (pic\_id) **REFERENCES** galery(id);

/\*\*

\* таблица references

\*/

**ALTER** **TABLE** user\_refs **ADD** **CONSTRAINT** **foreign** **key** (pic\_id) **REFERENCES** galery(id);