Создание таблиц в PostgreSQL с помощью pgAdmin 4

Разберем два способа создания таблиц, 1-й способ с помощью графического конструктора,

2-й способ с помощью языка SQL

Чтобы создать таблицу в БД, эта таблица должна в принципе существовать, таблице создаются не просто на сервере, они создаются в конкретной БД.

Иными словами БД в части PostgreSQL это некий контейнер относящийся к БД, а таблица, является тем объектом БД, в котором непосредственно хранятся сами БД, т.е. таблица является неким физическим представлением определенной сущности реального мира.

Кроме установки на сервере системной БД ничего нет. Поэтому сначала необходимо создать БД, т.е. тот контейнер в котором хранятся объекты БД.

1) Рассмотрим пример, сначала создадим БД, а затем в ней пару таблиц с несколькими ограничениями.

Databases🡪Create🡪Database, test\_db🡪OK , мы создали БД test\_db

2) C помощью языка SQL

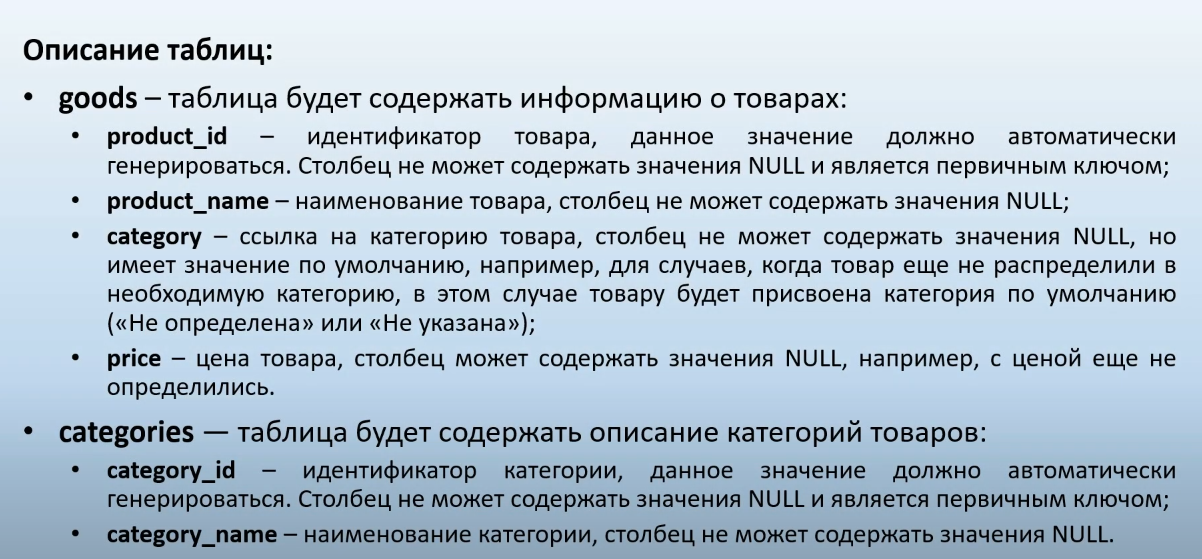
Переходим в Query Tool, запускаем редактор и вводим

CREATE DATABASE test\_db1; Databases🡪Refresh

В итоге будет создана такая же БД с настройками по умолчанию

Теперь мы можем создавать таблицы в этой БД

Представим, что нам нужно создать две таблицы, первая для хранения товаров, вторая для хранения категорий товаров, к которым относятся эти товары. Приведена структура таблиц.



1. Рассмотрим процесс создания таблиц с помощью графического конструктора pgadmin

Чтобы создать таблицу нужно зайти в обозревателе в БД, щелкнуть правой мышкой по контейнеру Schemas 🡪Tables🡪Create🡪Tables.

Создадим таблицу с категориями categories, чтобы потом в процессе создания таблиц с товарами у нас была возможность сразу определить ограничение внешнего ключа. Вводим в поле name categories. В случае необходимости Вы можете изменить владельца таблицы и схему в которой будет распологаться эта таблица. Схема – это пространство имен в БД, своего рода контейнер внутри БД, который позволяет логически разделять БД на схемы. После того как название таблицы задано, мы можем переходить к определению столбцов (columns). Нажав кнопку плюс (+) добавим нужные столбцы, поля.

Сначала создаем идентификатор категории, category\_id, у него должен быть целочисленный тип данных Integer, он не должен хранить значение NULL, он должен быть первичным ключом Primary key переключаем флаг, а также в нем должны автоматически генерироваться значения Edit row – ручка(карандаш). Переходим в ограничения (Constraints), переключаем в идентификатор (IDENTITY)

Далее также добавляем столбец (+) наименования категорий(category\_name), тип данных должен быть текстовый(character varying) например с длинной 100 - Length, не должен быть 0, и значение первичного ключа генерировать не нужно.

После ввода данных сохраняем все изменения (save) и создается таблица.

Также создаем таблицу с товарами(goods), и определяем для нее соответствующие столбцы.

Столбец идентификатор товара (product\_id) определяем как первичный ключ, не NULLи включаем у него удентификацию с автоматическим генерированием.

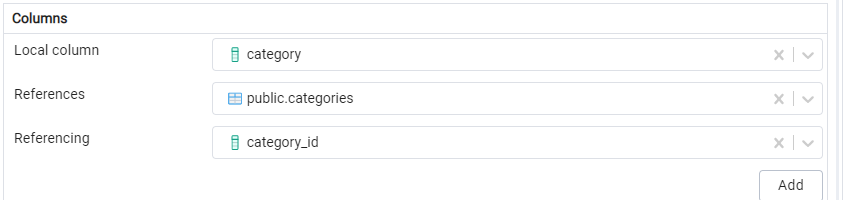
Затем заполняем product\_name, category, price (datatype – numeric, Length 18 , Scale -2).

Нужно добавить значение по умолчанию для поля category, например для случаев, когда товары еще не распределили в необходимую категорию, в этом случае товару будет присвоена категория по умолчанию, например не определена или не указана. А также давайте определим ограничение внешнего ключа, чтоб у нас отсутствовала возможность внести товар с несуществующей категорией. Чтоб у столбца(поля) задать значение по умолчанию необходимо открыть детализированное свойство столбца category (карандаш-ручка) перейти на вкладку ограничения (Constraints) и в поле Default указать значение, которое будет присваиваться по умолчанию, например 1., т.е. это есть идентификатор категории которая будет назначаться по умолчанию.

Осталось определить ограничение внешнего ключа, иными словами, чтобы столбец(поле) category таблицы goods ссылался на столбец(поле) категории (category\_id) таблицы categories.

Таким образом мы определим связь Relation – между этими таблицами. Чтобы это сделать переходим на вкладку ограничения(Constraints) основной формы создания таблиц, затем переходим на вкладку внешний ключ(Foreign Key), и спомощью кнопки (+) добавляем новый внешний ключ.

Вводим название ограничения (fk\_category\_goods), затем переходим в (ручку-карандаш), далее на вкладке столбцы(columns) задаем связь между таблицами



Локальный столбец

Ссылается на

Зависимый столбец

Затем нажимаем (+) или Add для добавления связи.

После этого сохраняем таблицу (Save). Созданная таблица отображается в обозревателе.

2. Рассмотрим процесс создания таблиц на языке SQL

--Удаление таблиц

-- DROP Table IF EXISTS goods, categories;

-- Создание таблицы categories

CREATE TABLE categories (

category\_id INT NOT NULL GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

category\_name VARCHAR(100) NOT NULL

);

-- Создание таблицы goods

CREATE TABLE goods (

product\_id INT NOT NULL GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

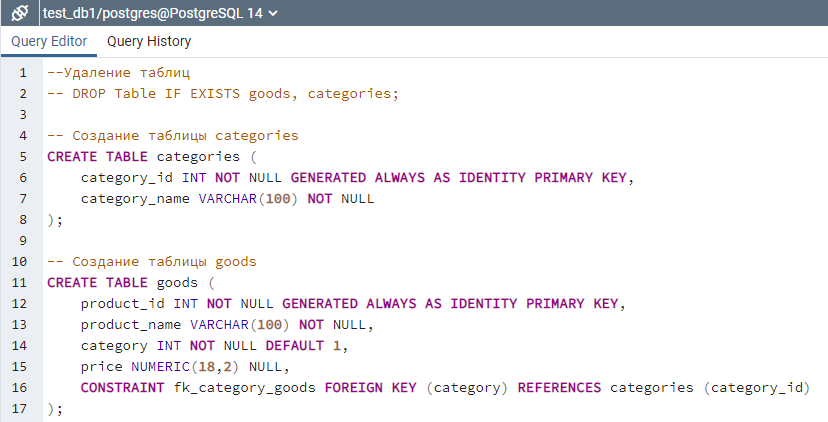
product\_name VARCHAR(100) NOT NULL,

category INT NOT NULL DEFAULT 1,

price NUMERIC(18,2) NULL,

CONSTRAINT fk\_category\_goods FOREIGN KEY (category) REFERENCES categories (category\_id)

);



В пределах одной схемы(БД) не может быть таблиц с одинаковыми названиями.

Пока наши таблицы не содержат данных, добавим данные в таблицы