Создание баз данных с использованием PostgreSQL

В этом занятии вы узнаете:

- Что такое SQL
- Что такое postgre и pgadmin
- Как создавать базы данных в postgre и pgadmin
- Ограничение на столбцы в таблицах
- Как создавать таблицы в pgadmin
- Наполнение данными таблиц
- Редактирование данных в таблицах
- Удаление записей из таблиц и таблиц

Что такое SQL

Язык SQL — это непроцедурный язык, а декларативный. Он позволяет осуществлять работу с данными во всех реляционных СУБД. Несмотря на то, что в различных СУБД имеются отличия в наречиях SQL, перенос БД из одной СУБД в другую (миграция) обычно возможна, при этом запросы SQL сильно не модифицируются. Операторы (команды), написанные на этом языке, лишь указывают СУБД, какой результат должен быть получен, но не описывают процедуру получения этого результата. СУБД сама определяет способ выполнения команды пользователя.

SQL позволяет производить следующие операции над данными:

- Создание таблиц и задание ограничений на колонки
- Добавление данных в таблицу
- Удаление данных из таблиц и удаление самих таблиц
- Редактирование данных и таблиц
- Выборка данных из одной или нескольких таблиц.

Описание действий над данными, выполненное с использованием языка SQL, называется запрос.

Основные понятия БД

В реляционных базах данных пользователь воспринимает данные в виде таблиц. Поэтому термину «файл» соответствует термин «таблица», вместо термина «запись» используется термин «строка», а вместо термина «поле» — термин «столбец» (или «колонка»). Таким образом, таблицы состоят из строк и столбцов, на пересечении которых должны находиться «атомарные» значения, которые нельзя разбить на более мелкие элементы без потери смысла.

В формальной теории реляционных баз данных эти таблицы называют отношениями (relations) — поэтому и базы данных называются реляционными. Отношение — это математический термин. При определении свойств таких отношений используется теория множеств. В терминах данной теории строки таблицы будут называться кортежами (tuples), а колонки — атрибутами. Отношение имеет заголовок, который состоит из атрибутов, и тело, состоящее из кортежей. Количество атрибутов называется степенью отношения, а количество кортежей — кардинальным числом. Кроме теории множеств, одним из

оснований реляционной теории является такой раздел математической логики, как исчисление предикатов.

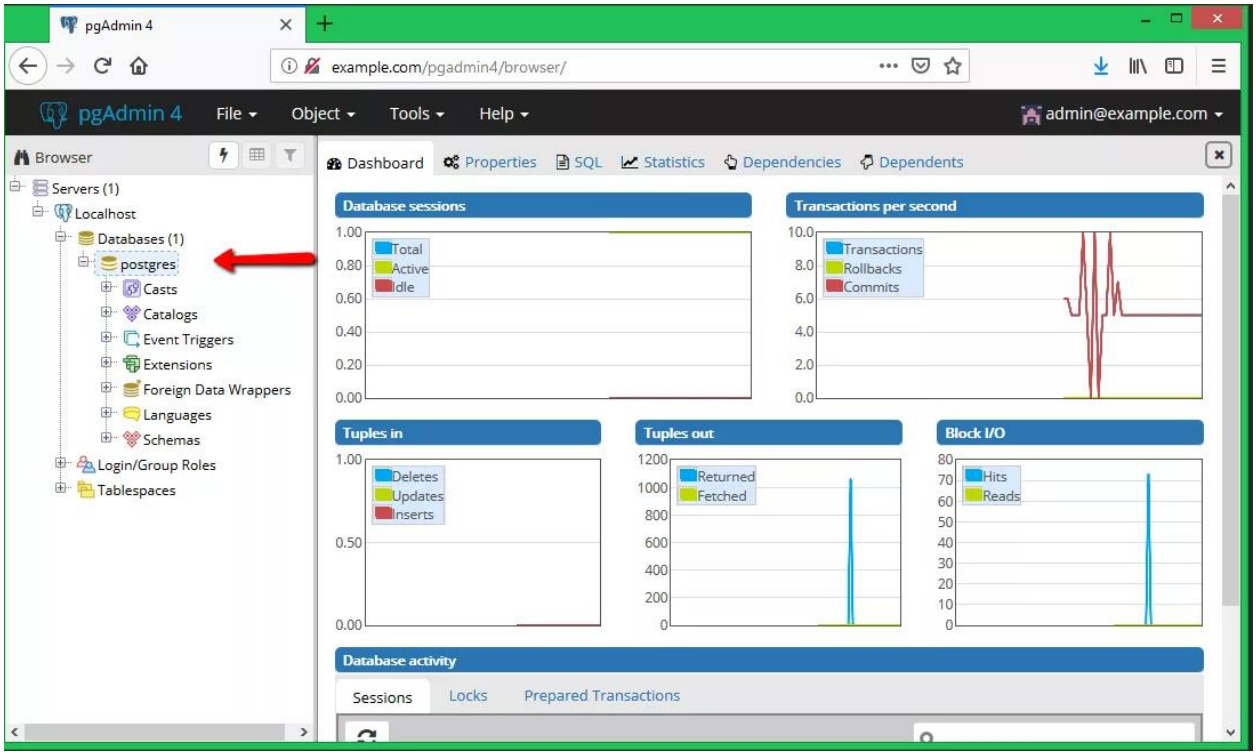
Что такое postgre и pgadmin

PostgreSQL - это мощная объектно-реляционная система управления базами данных с открытым исходным кодом с более чем 35-летним опытом активной разработки, которая заслужила прочную репутацию за надежность, функциональность и производительность.

Функции являются блоками кода, исполняемыми на сервере, а не на клиенте БД. Хотя они могут писаться на чистом SQL, реализация дополнительной логики, например, условных переходов и циклов, выходит за рамки SQL и требует использования некоторых языковых расширений. Функции могут писаться с использованием одного из следующих языков:

- Встроенный процедурный язык PL/pgSQL, во многом аналогичный языку PL/SQL, используемому в СУБД Oracle;
- Скриптовые языки — PL/Lua, PL/LOLCODE, PL/Perl, PL/PHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl, PL/Scheme, PL/v8 (Javascript);
- Классические языки — C, C++, Java (через модуль PL/Java);
- Статистический язык R (через модуль PL/R).

pgAdmin - это самая популярная и многофункциональная платформа администрирования и разработки с открытым исходным кодом для PostgreSQL, самой передовой базы данных с открытым исходным кодом.



pgAdmin предоставляет инструменты для администрирования баз данных, а также графический интерфейс для их использования.

Ограничение на столбцы в таблицах

Ограничения на столбцы в таблице – это правила, которые применяются к каждой колонке. Основные из них:

Первичный ключ (PRIMARY KEY). Это означает, что значения в данном столбце должны уникально идентифицировать каждую строку (значение в данном столбце уникальны для каждой отдельно взятой строки).

Уникальные значения (UNIQUE). Значения в колонках должны быть уникальны для каждой строки.

Запрет на пропуски (NOT NULL). В каждой строке данного столбца должно быть значение и не должно быть пропусков.

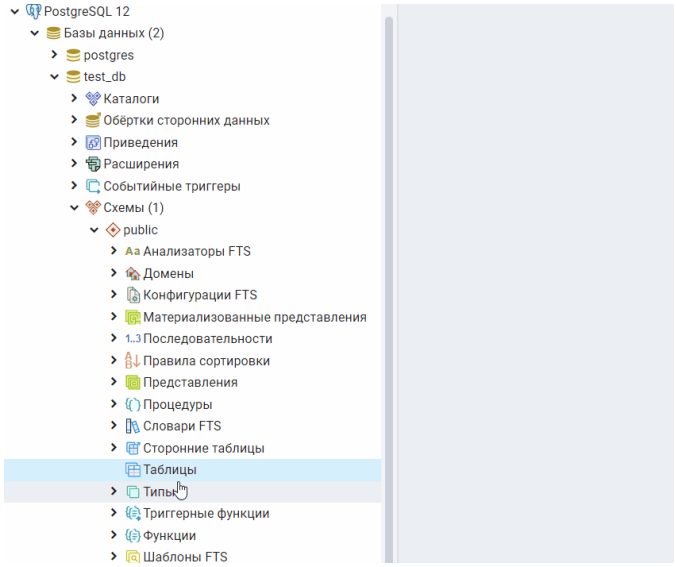
Значение по умолчанию (DEFAULT). Задается значение по умолчанию для строки колонки, если значение отсутствует.

Внешний ключ (FOREIGN KEY). Внешние ключи являются средством поддержания так называемой ссылочной целостности (referential integrity) между связанными таблицами.

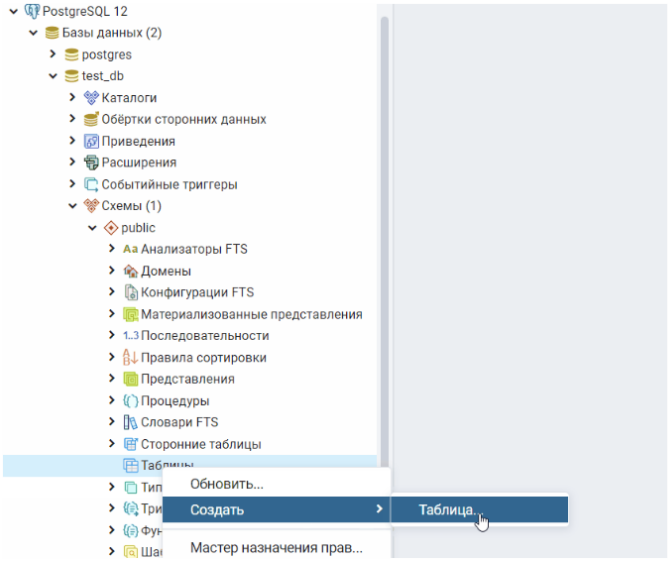
Может быть только один первичный ключ (PRIMARY KEY), но при этом колонок с уникальными значениями (UNIQUE) может быть несколько.

Как создавать базы данных в postgre и pgadmin

В pgadmin на панели слева расположен обозреватель решений. Для создания таблиц находим пункт «Таблицы», нажать левой кнопкой мыши.



Из появившегося меню выбираем создать – таблица.



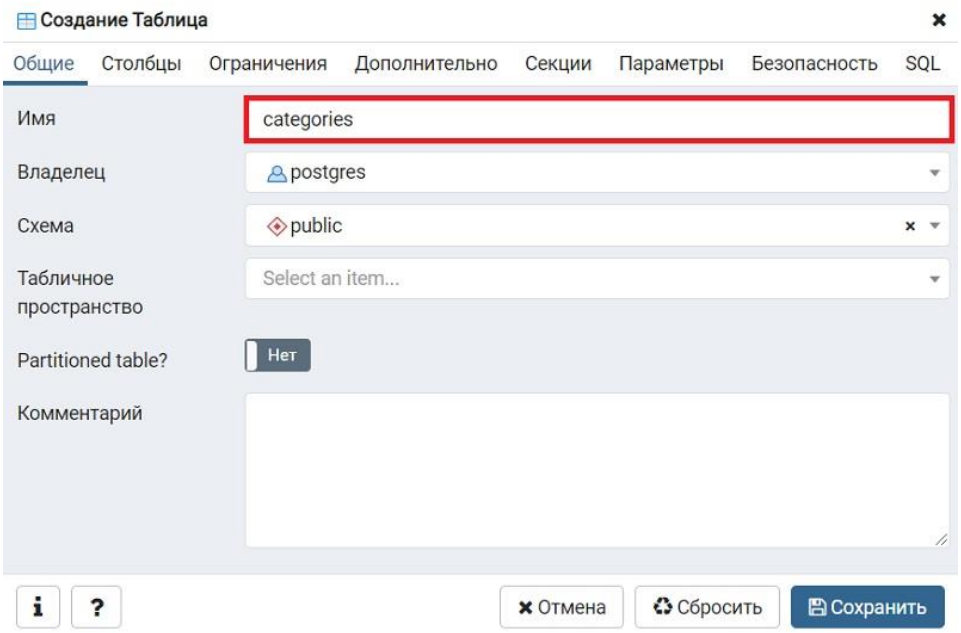
Затем запустится графический конструктор таблиц.

Давайте представим, что нам нужно создать две таблицы, первая – для хранения товаров, и вторая – для хранения категорий, к которым относятся эти товары.

Структура таблиц будет следующая:

- goods – таблица будет содержать информацию о товарах:
  - product_id – идентификатор товара, данное значение должно автоматически генерироваться. Столбец не может содержать значения NULL и является первичным ключом;
  - product_name – наименование товара, столбец не может содержать значения NULL;
  - category – ссылка на категорию товара, столбец не может содержать значения NULL, но имеет значение по умолчанию, например, для случаев, когда товар еще не распределили в необходимую категорию, в этом случае товару будет присвоена категория по умолчанию («Не определена» или «Не указана»);
  - price – цена товара, столбец может содержать значения NULL, например, с ценой еще не определились.
- categories — таблица будет содержать описание категорий товаров:
  - category_id – идентификатор категории, данное значение должно автоматически генерироваться. Столбец не может содержать значения NULL и является первичным ключом;
  - category_name – наименование категории, столбец не может содержать значения NULL.

Затем запустится графический конструктор таблиц, где первым делом нам необходимо ввести название таблицы, сначала давайте создадим таблицу с категориями, чтобы потом в процессе создания таблицы с товарами у нас была возможность сразу определить ограничение внешнего ключа...

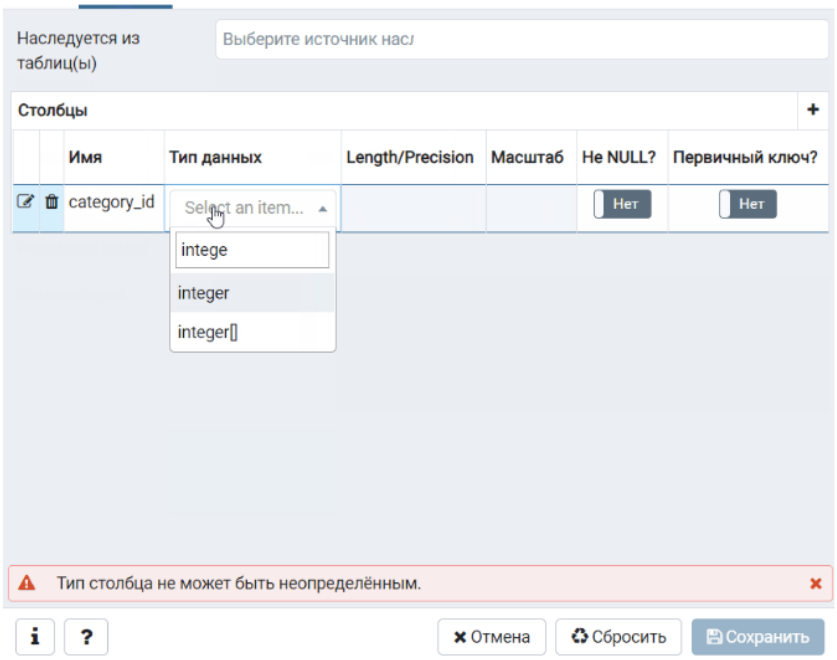


После того как название таблицы задано, мы можем переходить к определению столбцов. Для этого необходимо перейти на вкладку «Столбцы» и, используя кнопку плюс «+», добавить нужные столбцы.

Сначала создаем идентификатор категории, по условиям нашей задачи он должен отвечать определенным требованиям, поэтому мы должны задать следующие свойства у столбца:

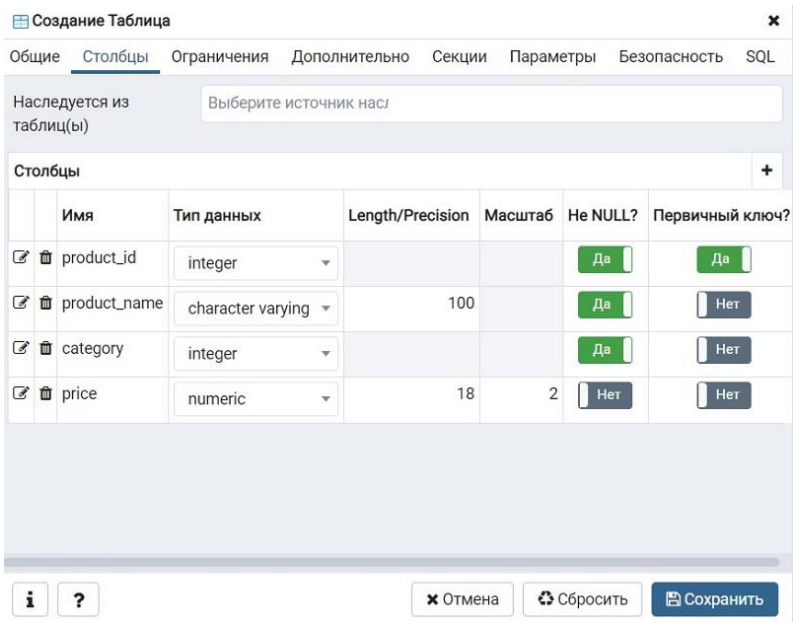
- Включить параметр «Не NULL», чтобы параметр не мог хранить значения NULL;
- Включить параметр «Первичный ключ», чтобы столбец выполнял роль первичного ключа;
- Включить идентификацию, чтобы в столбце автоматически генерировались значения.

В качестве типа данных выберем целочисленный тип integer.



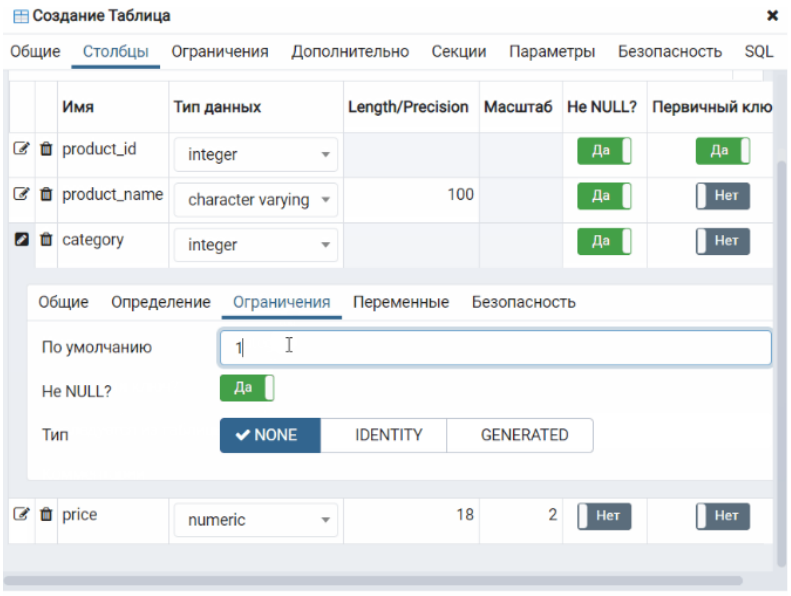
Далее точно так же добавляем столбец для хранения наименования категории. При этом тип данных у нас уже должен быть текстовый, например, character varying (VARCHAR) с длинной 100. Он уже не должен быть первичным ключом и значения генерировать здесь не нужно.

Затем точно также создаем таблицу с товарами и определяем соответствующие для нее столбцы. Столбец идентификатора товара определяем, как первичный ключ, и включаем у него идентификацию с автоматическим генерированием значений.



После того как столбцы определены, нам необходимо добавить значение по умолчанию для столбца category, а также определить ограничение внешнего ключа.

Чтобы у столбца задать значение по умолчанию, необходимо открыть детализированные свойства столбца, перейти там на вкладку «Ограничения» и в поле «По умолчанию» указать значение, которое будет присваиваться по умолчанию, например, 1.

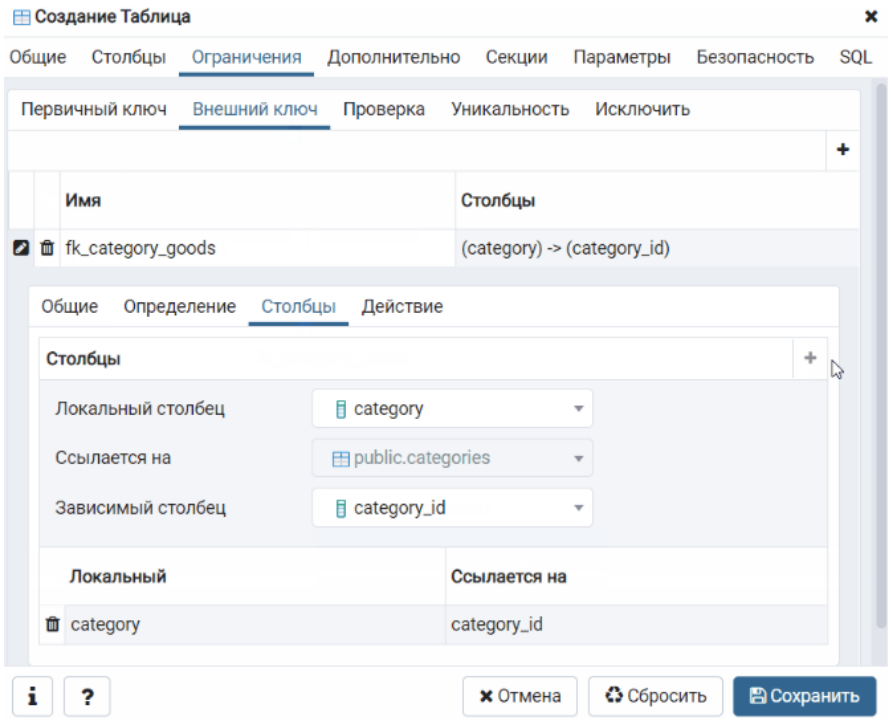


Осталось определить ограничение внешнего ключа, иными словами, чтобы столбец category таблицы goods ссылался на столбец category_id таблицы categories, таким образом, мы определим связь между этими таблицами.

Чтобы это сделать, переходим на вкладку «Ограничения» (основной формы создания таблиц), затем переходим на вкладку «Внешний ключ» и с помощью кнопки плюс «+» добавляем новый внешний ключ.

Вводим название ограничения, и в детализированных свойствах на вкладке «Столбцы» задаем связь между таблицами, т.е. указываем столбцы и нажимаем на плюс «+» для добавления связи.

После этого все требования, указанные в нашей задаче, будут выполнены и мы можем нажать кнопку «Сохранить» для создания таблицы.



Задание

Таблицу aircrafts следующего вида:

Описание атрибута	Имя атрибута	Тип данных	Тип PostgreSQL	Ограничения
Код самолета, IATA	aircraft_code	Символьный	char(3)	NOT NULL
Модель самолета	model	Символьный	text	NOT NULL
Максимальная дальность полета, км	range	Числовой	integer	NOT NULL range > 0

Как создавать таблицы в pgadmin (SQL)

Теперь давайте рассмотрим процесс создания таблиц в PostgreSQL на языке SQL.
Таблицы в SQL создаются с помощью инструкции CREATE TABLE.

Ее полный синтаксис представлен в документации на PostgreSQL, а упрощенный синтаксис таков:

CREATE TABLE имя-таблицы

(

имя-поля тип-данных [ограничения-целостности],

имя-поля тип-данных [ограничения-целостности], ...

имя-поля тип-данных [ограничения-целостности], [ограничение-целостности],

[первичный-ключ],

[внешний-ключ]

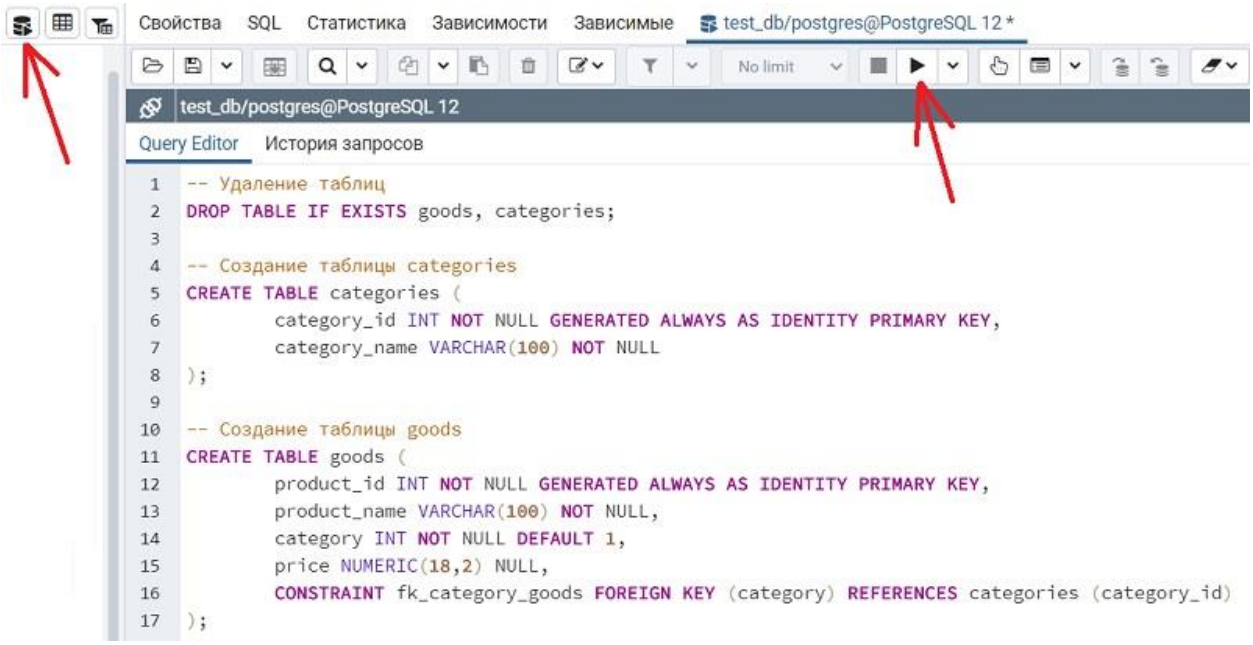
);

В квадратных скобках показаны необязательные элементы команды. После команды нужно поставить символ «;».

Чтобы выполнить инструкции, открываем редактор запросов (Запросник), вводим инструкции и нажимаем «Execute».

```
-- 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,
    FOREIGN KEY (category)
    REFERENCES categories (category_id)
     ON DELETE CASCADE
);
```



Задание

Удалите созданную ранее таблицу и создайте ее же с помощью SQL.

Создайте таблицу seats:

Описание атрибута	Имя атрибута	Тип данных	Тип PostgreSQL	Ограничения
Код самолета, IATA	aircraft_code	Символьный	char(3)	NOT NULL
Номер места	seat_no	Символьный	varchar(4)	NOT NULL
Класс обслуживания	fare_conditions	Символьный	varchar(10)	NOT NULL Значения из списка: Economy, Comfort, Business

Проверки вхождения значения в список значений используется CHECK и оператор IN

Наполнение данными таблиц

Теперь вы можете приступить к вводу данных в таблицу «Самолеты». Для выполнения этой операции служит команда INSERT. Ее упрощенный формат таков

INSERT INTO имя-таблицы [(имя-атрибута, имя-атрибута, ...)]

VALUES (значение-атрибута, значение-атрибута, ...);

Добавим значения в таблицу категорий:

INSERT INTO categories (category_name)

VALUES ('Булочки'),

('Пироженные');

Заполним данными таблицу товары:

INSERT INTO goods (product_name, category, price)

VALUES ('с маком', 1, 57),

('Катошка', 2, 87);

Задание

Заполните таблицы следующими данными:

Aircrafts:

```
 aircraft_code |        model         | range
---------------+----------------------+-------
 SU9           | Sukhoi SuperJet-100 |  3000
 773           | Boeing 777-300      | 11100
 763           | Boeing 767-300      |  7900
 733           | Boeing 737-300      |  4200
 320           | Airbus A320-200     |  5700
 321           | Airbus A321-200     |  5600
 319           | Airbus A319-100     |  6700
 CN1           | Cessna 208 Caravan  |  1200
 CR2           | Bombardier CRJ-200  |  2700
(9 строк)
```

Seats:

```
( 'SU9', '1A', 'Business' ),
( 'SU9', '1B', 'Business' ),
( 'SU9', '10A', 'Economy' ),
( 'SU9', '10B', 'Economy' ),
( 'SU9', '10F', 'Economy' ),
( 'SU9', '20F', 'Economy' );
```

Редактирование данных в таблицах

Теперь мы ознакомимся с командой UPDATE, предназначенной для обновления данных в таблицах. Ее упрощенный синтаксис такой:

UPDATE имя-таблицы

SET имя-атрибута1 = значение-атрибута1,

имя-атрибута2 = значение-атрибута2, ...

WHERE условие;

Предположим, что поменялись цены на пирожные картошка и теперь они стоят 90. Для внесения данных изменений напишем следующую инструкцию:

UPDATE goods

SET price = 90 WHERE product_name = 'Катошка';

Задание

Давайте предположим, что российские инженеры немного улучшили летные характеристики самолета Sukhoi SuperJet, и теперь дальность его полета стала на 500 км больше. Внесите изменения.

Удаление записей из таблиц и таблиц

Итак, мы добрались до операции удаления строк из таблиц. Для этого используется команда DELETE,

DELETE FROM имя-таблицы

WHERE условие;

Например, удалим пирожные из категорий.

DELETE FROM categories

WHERE category_name = 'Пироженные'

Заключение

На этом занятии вы узнали:

Что такое PostgreSQL и pgadmin

Основные команды для работы с таблицами в БД языка SQL

Самостоятельная работа

Этап 1. Создание базы данных

Создать новую базу данных ТорговыйДом(ваша фамилия).

В базе данных ТорговыйДом(ваша фамилия) выполнить следующе:

1. Создать таблицу Производители с полями:

− КодФирмы – ключевое поле;тип: счетчик;

− НазваниеФирмы – тип: короткий текст, длина 100 символов;

− Телефон – тип: короткий текст, длина 20 символов;

− Адрес – тип: короткий текст, длина 255 символов;

2. . Создать таблицу Товары с полями:

− КодТовара – ключевое поле, тип: короткий текст 10 символов;

− НаименованиеТовара – тип: короткий текст, длина 150 символов;

− Производитель – поле, связанное с полем КодФирмы таблицы Производители (для создания списка вместо типа выбрать Мастер подстановок);

− РозничнаяЦена – тип: денежный; формат: денежный; − ОптоваяЦена

– тип: денежный; формат: денежный;

− ОписаниеТовара -тип: длинный текст.

3. Создать таблицу Продавцы с полями:

− КодСотрудника – ключевое поле; тип: короткий текст 5 символов;

− Фамилия – тип: короткий текст 100 символов;

− Имя – тип: короткий текст 40 символов;

− Отчество – тип: короткий текст 50 символов;

− ДатаРождения – тип: Дата/время; формат поля

- средний формат даты; значение по умолчанию – не задавать;

− МестоРождения – тип: короткий текст; задать значение по умолчанию.

4. Создать таблицу Клиенты с полями (длину полей определить самостоятельно):

− КодКлиента - ключевое поле, счетчик;

− Фамилия – тип: короткий текст;

− Адрес– тип: короткий текст;

− РабочийТелефон– тип: короткий текст

ДомашнийТелефон– тип: короткий текст

− МобильныйТелефон– тип: короткий текст

− ЭлектроннаяПочта – тип: короткий текст (задать формат и маску).

5. Создать таблицу Заказы с полями:

− НомерЗаказа – ключевое поле, счетчик;

− Товар – список, связанный с соответствующим полем таблицы Товары

− Количество

– тип: числовой; Не может быть отрицательным.

− Клиент –поле, связано с полем КодКлиента таблицы Клиенты;

− Продавец – поле связано с полем КодСотрудника таблицы Продавцы