Углубление в SQL

Занятие 1.4

Алексей Кузьмин
Директор разработки, Data Scientist ДомКлик.ру

Алексей Кузьмин

Директор разработки, Data Scientist ДомКлик

О чём поговорим и что сделаем

1. Таблицы
2. Данные
3. Внешние ключи
4. Проверяющие ограничения

Таблицы

1

Создание таблиц

Для создания новой таблицы в PostgreSQL вы можете использовать команду CREATE TABLE:

```
CREATE TABLE table_name (
 column_name TYPE column_constraint,
 table_constraint table_constraint
);
```

- table_name — имя таблицы
- column_name — имя колонки
- TYPE — тип колонки
- column constraint — ограничение колонки
- table_constraint — ограничение таблицы

1

Ограничения колонок

- NOT NULL
- UNIQUE — каждое значение, кроме NULL, должно быть уникально
- PRIMARY KEY — комбинация NOT NULL + UNIQUE. На таблицу разрешён только один PRIMARY KEY

2

Ограничения таблиц

- UNIQUE (column_list) — уникальность на группу колонок
- PRIMARY KEY(column_list) — первичный ключ по группе колонок.

Пример создания таблицы с пользователями

```
CREATE TABLE account(
 user_id serial PRIMARY KEY,
 username VARCHAR (50) UNIQUE NOT NULL,
 password VARCHAR (50) NOT NULL,
 email VARCHAR (355) UNIQUE NOT NULL,
 created_on TIMESTAMP NOT NULL,
 last_login TIMESTAMP
);
```

Изменение таблиц

Используется команда ALTER TABLE:

```
ALTER TABLE table_name action;
```

С её помощью можно:

- добавить, удалить, переименовать или изменить тип у колонки
- установить значение по умолчанию
- переименовать таблицу

```
ALTER TABLE table_name ADD COLUMN new_column_name TYPE;

ALTER TABLE table_name DROP COLUMN column_name;

ALTER TABLE table_name RENAME COLUMN column_name TO new_column_name;

ALTER TABLE table_name ALTER COLUMN column_name [SET DEFAULT value | DROP
DEFAULT]

ALTER TABLE table_name ALTER COLUMN column_name [SET NOT NULL| DROP NOT NULL]

ALTER TABLE table_name ADD CONSTRAINT constraint_name constraint_definition

ALTER TABLE table_name RENAME TO new_table_name;
```

Удаление таблиц

Команда DROP TABLE:

```
DROP TABLE [IF EXISTS] table_name [CASCADE | RESTRICT];
```

RESTRICT/CASCADE — если какие-то таблицы ссылаются на эту, то режим RESTRICT не даст удалить таблицу. Режим CASCADE позволит каскадно удалить зависимости

Время практики

Практика 1

Создайте таблицу «Автор» с полями:

- Id
- full name
- псевдоним (может не быть)
- дата рождения
- дата создания (значение по умолчание)

Практика 1. Решение

```
CREATE TABLE author (
author_id serial PRIMARY KEY,
author_name varchar(50) NOT NULL,
nick_name varchar(50),
born_date date NOT NULL,
create_date timestamp DEFAULT now()
)
```

Данные

2

Вставка данных

Пример:

```
INSERT INTO table(column1, column2, …)
VALUES
 (value1, value2, …);
```

- table — название таблицы
- column1, ... — названия колонок
- value1, ... — значения для вставки

```
CREATE TABLE link (
 ID serial PRIMARY KEY,
 url VARCHAR (255) NOT NULL,
 name VARCHAR (255) NOT NULL,
 description VARCHAR (255),
 rel VARCHAR (50)
);
INSERT INTO link (url, name)
VALUES
 ('http://www.google.com','Google');
```

Можно вставлять несколько строк одновременно:

```
INSERT INTO table (column1, column2, …)
VALUES
 (value1, value2, …),
 (value1, value2, …) ,...;


INSERT INTO link (url, name)
VALUES
 ('http://www.yahoo.com','Yahoo'),
 ('http://www.bing.com','Bing');
```

Можно вставлять данные из других таблиц:

```
INSERT INTO table_1 (column1, column2, …)
SELECT value1, value2, … FROM table_2;


INSERT INTO customers (first_name, last_name)
SELECT first_name, last_name FROM users;
```

Модификация данных

Команда:

```
UPDATE table
SET column1 = value1,
    column2 = value2 ,...
WHERE
 condition;
```

Пример:

```
UPDATE link
SET description = 'no description'
WHERE
    description IS NULL;
```

Обновление всех строк:

```
UPDATE link
SET rel = 'nofollow';
```

Все ряды, основываясь на другой колонке:

```
UPDATE link
SET description = name
```

Удаление данных

Команда:

```
DELETE FROM table
WHERE condition;
```

Пример:

```
DELETE FROM link
WHERE id = 1;
```

Время практики

Практика 2

- Вставьте данные по любым трём писателям в указанную таблицу
- Добавьте поле «место рождения» в таблицу
- Обновите данные, проставив корректное место рождения каждому писателю
- Нужно использовать только sql команды

Практика 2. Решение

```
INSERT INTO author (author_name, nick_name, born_date)
VALUES ('Жюль Габриэль Верн', null, '08.02.1828'),
('Михаил Юрьевич Лермонтов', 'Гр. Диарбекир', '03.10.1814'),
('Харуки Мураками', null, '12.01.1949')
```

```
ALTER TABLE author ADD COLUMN born_place varchar(50)
```

```
UPDATE author
SET born_place = 'Франция'
WHERE id = 1
```

```
UPDATE author
SET born_place = ‘Российская Империя'
WHERE id = 2
```

```
UPDATE author
SET born_place = ‘Япония'
WHERE id = 3
```

Внешние ключи

3

Внешний ключ — это поле, которое ссылается на строку в другой таблице.
Таблица, содержащая внешний ключ, обычно называется дочерней, а таблица, на которую указывают, — родительской.
Таблица может содержать несколько внешних ключей.

Таблица адресов доставки:

```
CREATE TABLE so_headers (
 id SERIAL PRIMARY KEY,
 customer_id INTEGER,
 ship_to VARCHAR (255)
);
```

Заказы:

```
CREATE TABLE so_items (
  item_id INTEGER NOT NULL,
  so_id INTEGER,
  product_id INTEGER,
  qty INTEGER,
  net_price NUMERIC,
  PRIMARY KEY (item_id,so_id)
);
```

Если мы хотим показать, что so_id ссылается на строку в so_headers
и требует, чтобы она была, то можно передать скрипт создания таблицы:

```
CREATE TABLE so_items (
  item_id INTEGER NOT NULL,
  so_id INTEGER REFERENCES so_headers(id),
  product_id INTEGER,
  qty INTEGER,
  net_price NUMERIC,
PRIMARY KEY (item_id,so_id)
);
```

Другой способ записи:

```
CREATE TABLE so_items (
 item_id INTEGER NOT NULL,
 so_id INTEGER,
 product_id INTEGER,
 qty INTEGER,
 net_price NUMERIC,
 PRIMARY KEY (item_id, so_id),
 FOREIGN KEY (so_id) REFERENCES so_headers (id)
);
```

PostgreSQL создает констраинт, который будет проверять наличие строки в so_headers при любом изменении таблицы so_items. Кроме того, из so_headers удалять можно будет только строки, на которые никто не ссылается из so_items

Проверяющие ограничения

4

Ограничение CHECK позволяет проверять допустимое множество значений для поля.

Пример:

```
CREATE TABLE employees (
 id SERIAL PRIMARY KEY,
 first_name VARCHAR (50),
 last_name VARCHAR (50),
 birth_date DATE CHECK (birth_date > '1900-01-01'),
 joined_date DATE CHECK (joined_date > birth_date),
 salary NUMERIC CHECK(salary > 0)
```

Время практики

Практика 3

- Создайте таблицу «Произведения» с полями: год, название, ссылка на автора. Установите foreign key contsraint
- Вставьте в таблицу пару значений
- Попробуйте удалить автора

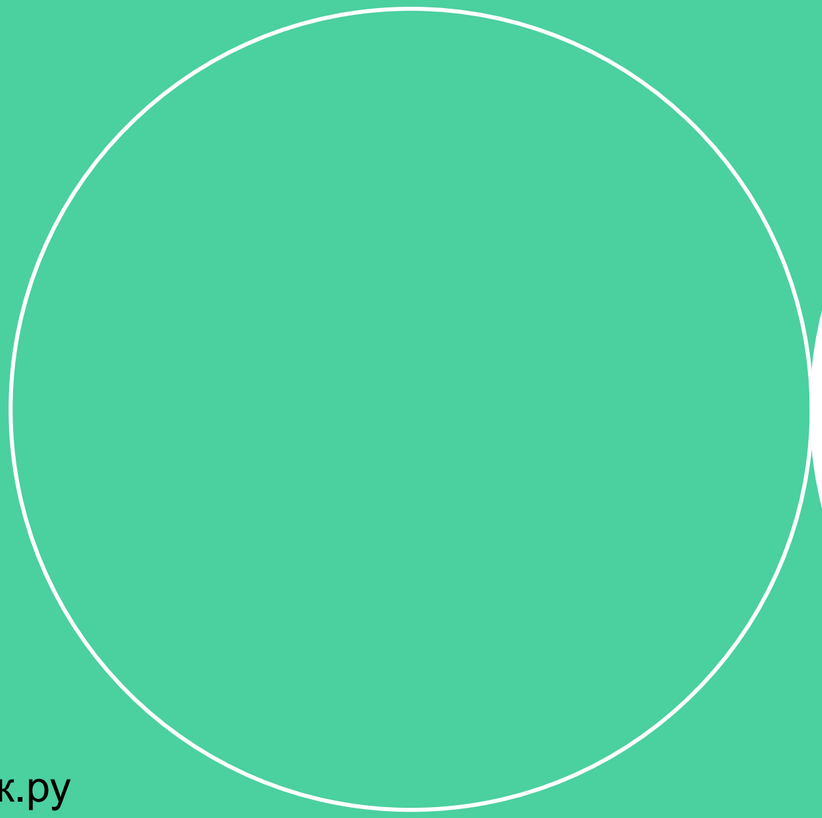
Практика 3. Решение

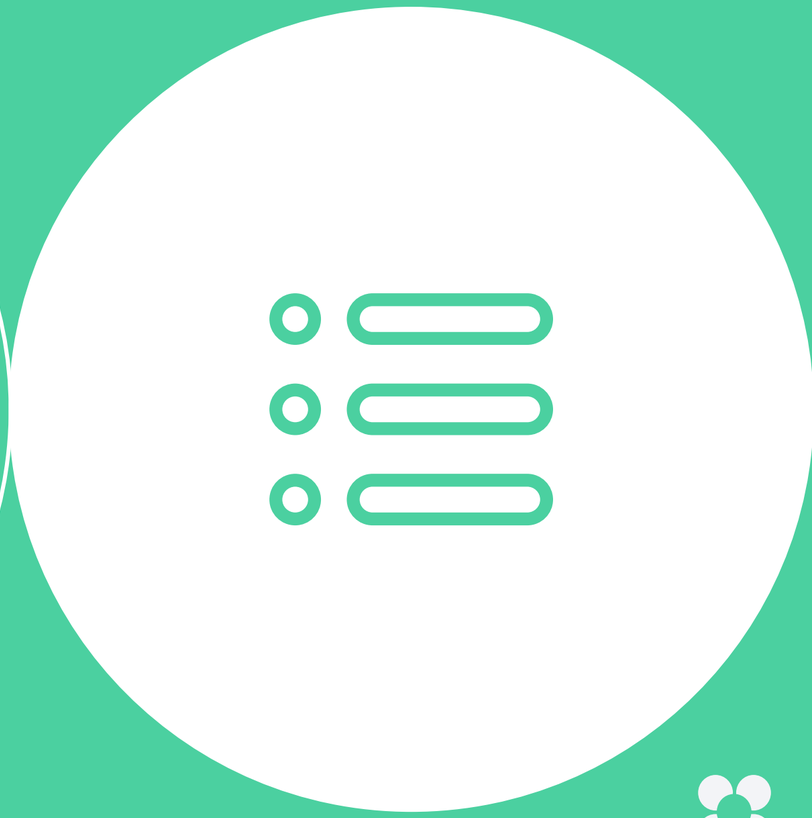
```
CREATE TABLE books (
book_id serial PRIMARY KEY,
book_name varchar(150) NOT NULL,
book_year int2 NOT NULL CHECK (book_year >= 0 AND book_year <= 2100),
author_id int2 REFERENCES author(author_id)
create_date timestamp DEFAULT now(),
)
```

```
INSERT INTO books (book_name, book_year, author_id )
VALUES ('Двадцать тысяч льё под водой', 1916, 1),
('Бородино', 1837, 2),
('Норвежский лес', 1980, 3)
```

```
DELETE FROM author
WHERE id = 1
```

Итоги занятия





Алексей Кузьмин
Директор разработки, Data Scientist ДомКлик.ру

Мы сегодня изучили:

1. Создание и модификацию таблиц
2. Вставку и модификацию данных
3. Constraints

Спасибо за внимание!

Алексей Кузьмин
Директор разработки, Data Scientist ДомКлик.ру