

Убираем крякозябры:
шрифт Lucida console
C:\Users\Viktor>chcp 1251

\\ Подключаемся к БД
C:\Users\Viktor>psql -U postgres
C:\Users\Viktor>psql
C:\Users\Viktor>psql -U postgres -d postgres

\\ Создать пользователя user1
postgres=# CREATE USER user1 WITH PASSWORD '1111';
CREATE ROLE

\\ Создать базу данных db1 с владельцем user1
postgres=# CREATE DATABASE db1 WITH OWNER user1;
CREATE DATABASE

\\ выход
postgres=# \q

\\ Зайти в базу данных db1 от имени пользователя user1
C:\Users\Viktor>psql -U user1 -d db1
Пароль пользователя user1:
psql (16.0)
Введите "help", чтобы получить справку.
db1=>

\\ Задать владельца существующей БД:
postgres=# CREATE DATABASE test;
CREATE DATABASE
postgres=# ALTER DATABASE test OWNER TO user1;
ALTER DATABASE

\\ список всех баз данных:
db1=> \l

\\ Удалить базу:
postgres=# DROP DATABASE test;
DROP DATABASE

\\ DDL

\\ Создать таблицу Categories с полями Id, Name
db1=> CREATE TABLE IF NOT EXISTS Categories (Id SERIAL PRIMARY KEY, Name VARCHAR(40) NOT NULL UNIQUE);
CREATE TABLE
db1=> \dt \\ список всех таблиц

\\ Создать таблицу Products с полями Id, Name, Price, Description, и внешним ключом Categories(Id)
CREATE TABLE IF NOT EXISTS Products (
 Id SERIAL PRIMARY KEY,
 Name VARCHAR(40) NOT NULL,
 Price INTEGER CHECK(Price > 0),
 Description TEXT,
 CategoryId INTEGER REFERENCES Categories(Id) NOT NULL
);

\\ Создать таблицу Products с полями Id, Rating, Autor, Text, и внешним ключом Products(Id)
CREATE TABLE Reviews (
 Id SERIAL PRIMARY KEY,
 Rating INTEGER CHECK(Rating > 0 AND Rating <= 5),
 Autor VARCHAR(50),
 Text TEXT,
 ProductId INTEGER NOT NULL REFERENCES Products(Id)
);

\\ Список всех таблиц
db1=> \dt
Список отношений
Схема | Имя | Тип | Владелец
-----+-----+-----+-----
public | categories | таблица | user1
public | products | таблица | user1
public | reviews | таблица | user1
(3 строки)

\\ Посмотреть поля таблицы и типы
db1=> \d products;

\\ Запросы из лекции

```
CREATE TABLE IF NOT EXISTS Categories (
    Id SERIAL PRIMARY KEY,
    Name VARCHAR(50) not null
);
```

```
CREATE TABLE IF NOT EXISTS Products (
    Id SERIAL PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Category_Id INTEGER REFERENCES Categories(Id) NOT NULL,
    Description TEXT, Photo BYTEA
);
```

```
CREATE TABLE IF NOT EXISTS Reviews (
    Id SERIAL PRIMARY KEY,
    Autor VARCHAR(50),
    text TEXT,
    data TIMESTAMP,
    Rating INTEGER CHECK(Rating > 0 AND Rating <= 5) NOT NULL,
    ProductId INTEGER REFERENCES Products(Id) NOT NULL
);
```

\\ Добавление записей в таблицу

\\ Сначала посмотрим поля таблицы и их типы

Db1=> \d categories

Таблица "public.categories"				
Столбец	Тип	Правило сортировки	Допустимость NULL	По умолчанию
id	integer		not null	nextval('categories_id_seq'::regclass)
name	character varying(50)		not null	

Индексы:

"categories_pkey" PRIMARY KEY, btree (id)

Ссылки извне:

TABLE "products" CONSTRAINT "products_category_id_fkey" FOREIGN KEY (category_id) REFERENCES categories(id)

\\ Добавим пару категорий (Id добавляется автоматически)

```
db2=> INSERT INTO categories(name) VALUES('Phones');
INSERT 0 1
db2=> INSERT INTO categories(name) VALUES('Accessories');
INSERT 0 1
db2=> SELECT * FROM categories;
 id | name
----+-----
  1 | Phones
  2 | Accessories
(2 строки)
```

\\ тоже для таблицы products, где допускается оставляем поля пустыми

db2=> \d products

Таблица "public.products"				
Столбец	Тип	Правило сортировки	Допустимость NULL	По умолчанию
id	integer		not null	nextval('products_id_seq'::regclass)
name	character varying(100)		not null	
category_id	integer		not null	
description	text			
photo	bytea			
price	numeric(10,2)		not null	

Индексы:

"products_pkey" PRIMARY KEY, btree (id)

Ограничения внешнего ключа:

"products_category_id_fkey" FOREIGN KEY (category_id) REFERENCES categories(id)

Ссылки извне:

TABLE "reviews" CONSTRAINT "reviews_productid_fkey" FOREIGN KEY (productid) REFERENCES products(id)

```
db2=> INSERT INTO Products(name, category_id, price) VALUES('Iphone 15', 1, 999.90);
INSERT 0 1
db2=> INSERT INTO Products(name, category_id, price) VALUES('Iphone 14', 1, 799.99);
INSERT 0 1
db2=> INSERT INTO Products(name, category_id, price) VALUES('Iphone 13', 1, 699.50);
INSERT 0 1
db2=> INSERT INTO Products(name, category_id, price) VALUES('Cover Iphone 13', 2, 9.29);
INSERT 0 1
db2=> SELECT * FROM products;
 id | name          | category_id | description | photo | price
----+-----+-----+-----+-----+-----
  1 | Iphone 15     |          1 |             |       | 999.90
  2 | Iphone 14     |          1 |             |       | 799.99
  3 | Iphone 13     |          1 |             |       | 699.50
```

4 | Cover Iphone 13 | 2 | 9.29
(4 строки)

```
INSERT INTO reviews(author, text, data, rating, productid)
VALUES('Viktor', 'Неплохой телефон!', now(), 4, 2);
INSERT INTO reviews(author, text, data, rating, productid)
VALUES('Азамат', 'Супер аппарат!', now(), 5, 3);
```