Министерство науки и высшего образования РФ

федеральное государственное автономное  
образовательное учреждение высшего образования

«Омский государственный технический университет»

Факультет информационных технологий и компьютерных систем

Кафедра «Прикладная математика и фундаментальная информатика»

|  |
| --- |
| **Лабораторная работа №2** |
| по дисциплине **Системы управления базами данных**  **Тема: СОЗДАНИЕ НОВОЙ БАЗЫ ДАННЫХ В СРЕДЕ POSTGRESQL, СОЗДАНИЕ И СВЯЗЫВАНИЕ ТАБЛИЦ БАЗЫ ДАННЫХ В СРЕДЕ POSTGRESQL** |

|  |  |  |  |
| --- | --- | --- | --- |
| Студента | Кузиной Варвары Михайловны | | |
| Курс | 3 | Группа | МО-231 |
| Направление | 02.03.03 Математическое обеспечение и администрирование информационных систем | | |
| Руководитель | доц., к.н. | | |
|  | Моисеева Н.А. | | |
| Выполнил | 06.10 | | |
| Проверил |  | | |

Омск 2025

**Цель работы:**

Спроектировать и реализовать новую базу данных в PostgreSQL по индивидуальному варианту: создать БД, задать таблицы и типы полей, реализовать связи и ограничения целостности, получить схему БД и подготовить отчёт с результатами.

**Описание предметной области (кратко):**

Учёт заявок, чеков и накладных на товары. Сущности: сотрудники, поставщики, каталог товаров, заявки, состав заявок, чеки, состав чеков, накладные, состав накладных. Между сущностями реализованы связи 1:M.

**Описание целостности:**

• По сущностям (первичные ключи):

employees(employee\_id),

suppliers(supplier\_code),

product\_catalog(product\_code),

orders(order\_number),

receipts(receipt\_number),

invoices(invoice\_number),

order\_composition(order\_number, product\_code),

check\_composition(receipt\_number, product\_code),

invoice\_composition(invoice\_number, product\_code).

• Семантическая целостность:

— даты и моменты времени хранятся в типах date/timestamp; — наименования и адреса — text; — цены — numeric(10,2); — идентификаторы — integer; — для составных таблиц первичный ключ составной, исключающий дубликаты позиций.

• Ссылочная целостность:

Все связи реализованы внешними ключами с каскадными действиями ON DELETE/UPDATE CASCADE для поддержания согласованности данных при изменениях.

**Шаг 1. Создание базы данных (рисунок 5):**

-- выполнить в подключении к postgres  
DROP DATABASE IF EXISTS rgr\_store;  
  
CREATE DATABASE rgr\_store

WITH

OWNER = postgres

ENCODING = 'UTF8'

TEMPLATE = template0

LC\_COLLATE = 'Russian\_Russia.1251'

LC\_CTYPE = 'Russian\_Russia.1251';  
  
COMMENT ON DATABASE rgr\_store  
 IS 'RGR: база данных для оформления заказов/чеков/накладных';

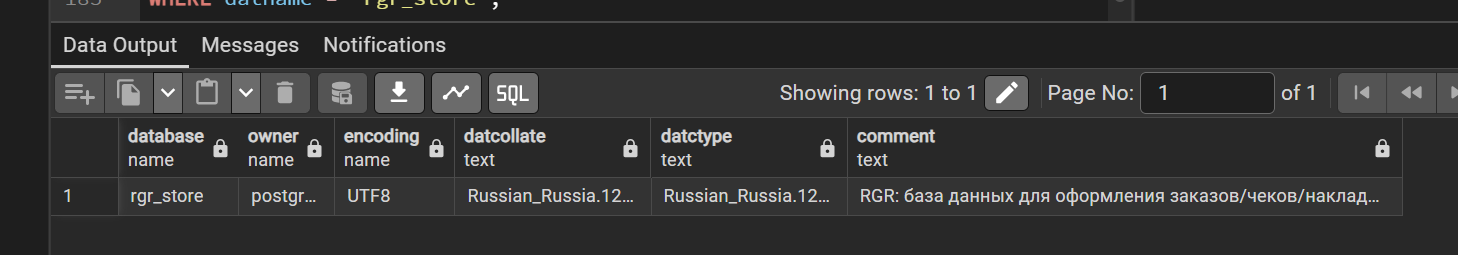


Рисунок 5 — Результат создания бд

**Шаг 2. Таблицы (PK/NOT NULL внутри CREATE TABLE) (рисунок 6):**

-- выполнять уже в базе rgr\_store  
  
-- При пересоздании  
DROP TABLE IF EXISTS invoice\_composition CASCADE;  
DROP TABLE IF EXISTS invoices CASCADE;  
DROP TABLE IF EXISTS check\_composition CASCADE;  
DROP TABLE IF EXISTS receipts CASCADE;  
DROP TABLE IF EXISTS order\_composition CASCADE;  
DROP TABLE IF EXISTS orders CASCADE;  
DROP TABLE IF EXISTS product\_catalog CASCADE;  
DROP TABLE IF EXISTS employees CASCADE;  
DROP TABLE IF EXISTS suppliers CASCADE;  
  
-- Сотрудники  
CREATE TABLE public.employees (  
 employee\_id integer PRIMARY KEY,  
 emp\_name text NOT NULL,  
 emp\_position text NOT NULL  
);  
  
-- Поставщики  
CREATE TABLE public.suppliers (  
 supplier\_code integer PRIMARY KEY,  
 supplier\_name text NOT NULL  
);  
  
-- Каталог товаров  
CREATE TABLE public.product\_catalog (  
 product\_code integer PRIMARY KEY,  
 product\_name text NOT NULL,  
 product\_category text NOT NULL,  
 product\_price numeric(10,2) NOT NULL  
);  
  
-- Заявки  
CREATE TABLE public.orders (  
 order\_number integer PRIMARY KEY,  
 order\_date date NOT NULL  
);  
  
-- Состав заявки  
CREATE TABLE public.order\_composition (  
 order\_number integer NOT NULL,  
 product\_code integer NOT NULL,  
 product\_quantity integer NOT NULL,  
 CONSTRAINT order\_composition\_pk PRIMARY KEY (order\_number, product\_code)  
);  
  
-- Чеки  
CREATE TABLE public.receipts (  
 receipt\_number integer PRIMARY KEY,  
 receipt\_date timestamp NOT NULL,  
 cash\_register\_number integer NOT NULL,  
 employee\_id integer NOT NULL  
);  
  
-- Состав чека  
CREATE TABLE public.check\_composition (  
 receipt\_number integer NOT NULL,  
 product\_code integer NOT NULL,  
 quantity\_in\_check integer NOT NULL,  
 CONSTRAINT check\_composition\_pk PRIMARY KEY (receipt\_number, product\_code)  
);  
  
-- Накладные  
CREATE TABLE public.invoices (  
 invoice\_number integer PRIMARY KEY,  
 invoice\_date date NOT NULL,  
 supplier\_code integer NOT NULL  
);  
  
-- Состав накладной  
CREATE TABLE public.invoice\_composition (  
 invoice\_number integer NOT NULL,  
 product\_code integer NOT NULL,  
 product\_quantity integer NOT NULL,  
 CONSTRAINT invoice\_composition\_pk PRIMARY KEY (invoice\_number, product\_code)  
);

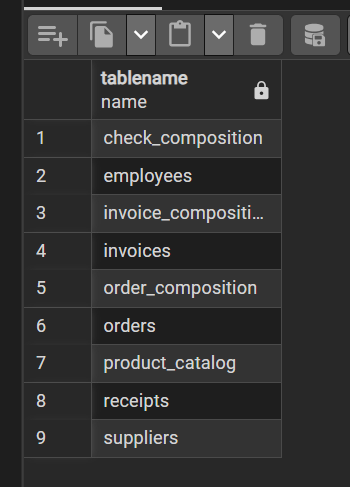


Рисунок 6 — Созданные таблицы

**Шаг 3. Связи и ограничения целостности (FK) (рисунок 7):**

-- order\_composition → orders / product\_catalog  
ALTER TABLE public.order\_composition  
 ADD CONSTRAINT oc\_fk\_order  
 FOREIGN KEY (order\_number)  
 REFERENCES public.orders (order\_number)  
 ON DELETE CASCADE ON UPDATE CASCADE;  
  
ALTER TABLE public.order\_composition  
 ADD CONSTRAINT oc\_fk\_product  
 FOREIGN KEY (product\_code)  
 REFERENCES public.product\_catalog (product\_code)  
 ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- receipts → employees  
ALTER TABLE public.receipts  
 ADD CONSTRAINT receipts\_fk\_employee  
 FOREIGN KEY (employee\_id)  
 REFERENCES public.employees (employee\_id)  
 ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- check\_composition → receipts / product\_catalog  
ALTER TABLE public.check\_composition  
 ADD CONSTRAINT cc\_fk\_receipt  
 FOREIGN KEY (receipt\_number)  
 REFERENCES public.receipts (receipt\_number)  
 ON DELETE CASCADE ON UPDATE CASCADE;  
  
ALTER TABLE public.check\_composition  
 ADD CONSTRAINT cc\_fk\_product  
 FOREIGN KEY (product\_code)  
 REFERENCES public.product\_catalog (product\_code)  
 ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- invoices → suppliers  
ALTER TABLE public.invoices  
 ADD CONSTRAINT invoices\_fk\_supplier  
 FOREIGN KEY (supplier\_code)  
 REFERENCES public.suppliers (supplier\_code)  
 ON DELETE CASCADE ON UPDATE CASCADE;  
  
-- invoice\_composition → invoices / product\_catalog  
ALTER TABLE public.invoice\_composition  
 ADD CONSTRAINT ic\_fk\_invoice  
 FOREIGN KEY (invoice\_number)  
 REFERENCES public.invoices (invoice\_number)  
 ON DELETE CASCADE ON UPDATE CASCADE;  
  
ALTER TABLE public.invoice\_composition  
 ADD CONSTRAINT ic\_fk\_product  
 FOREIGN KEY (product\_code)  
 REFERENCES public.product\_catalog (product\_code)  
 ON DELETE CASCADE ON UPDATE CASCADE;

Проверка результата (вставить скриншоты вывода)

-- Таблицы в public  
SELECT tablename  
FROM pg\_catalog.pg\_tables  
WHERE schemaname='public'  
ORDER BY tablename;  
  
-- Все внешние ключи  
SELECT conname,  
 conrelid::regclass AS table\_name,  
 confrelid::regclass AS ref\_table  
FROM pg\_constraint  
WHERE contype='f'  
ORDER BY conname;

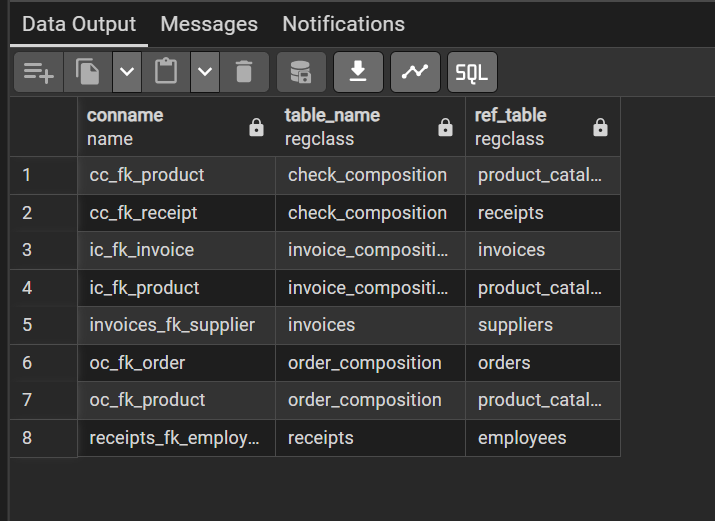


Рисунок 7 — Созданные связи

**Схема БД (ERD) (рисунок 8):**

Сформировать в pgAdmin: Tools → ERD Tool → ПКМ по rgr\_store → ERD for Database.

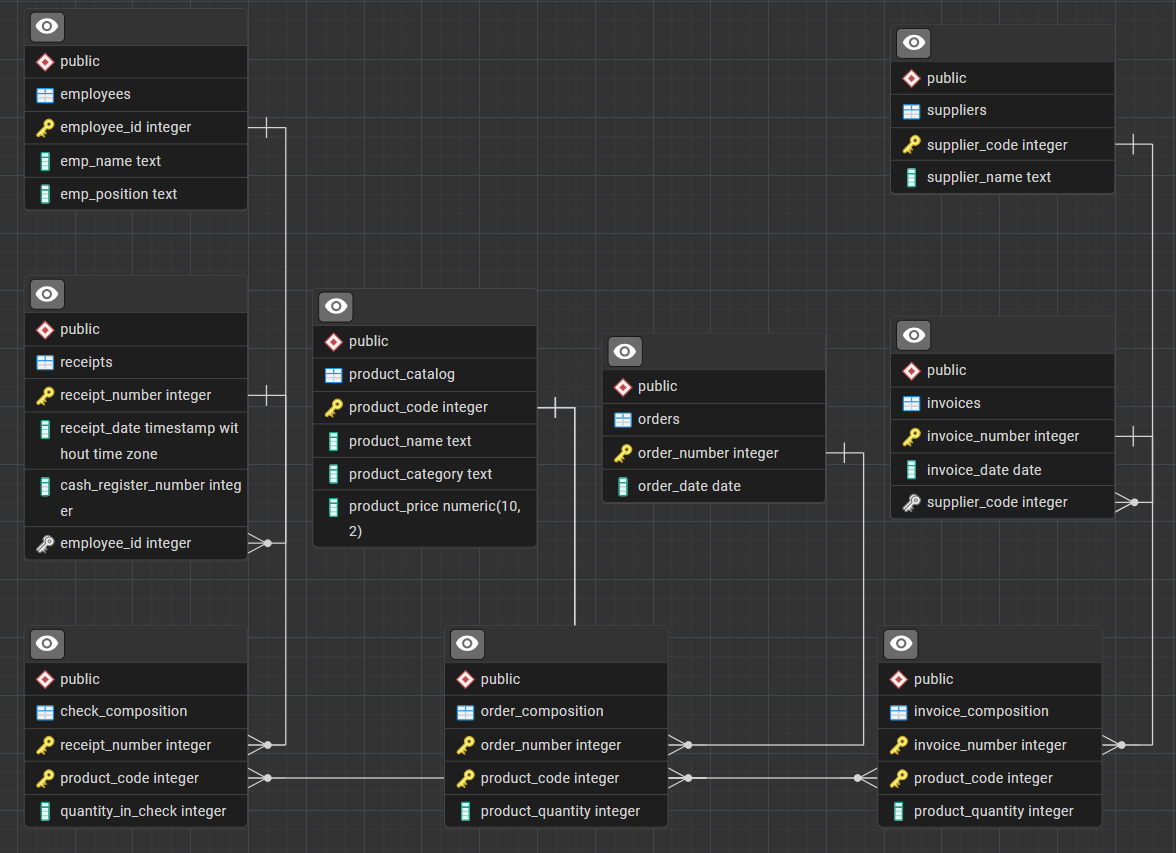


Рисунок 8 — Схема БД

**Заключение:**

Создана новая БД rgr\_store. Реализованы таблицы и связи предметной области, заданы первичные ключи, внешние ключи с каскадными действиями, обеспечены семантическая и ссылочная целостность. Результаты подтверждены выполнимыми скриптами и проверочными выборками.