



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт Информационных технологий (ИИТ)

Кафедра цифровой трансформации (ЦТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 6

по дисциплине

«Проектирование баз данных»

**Тема: «Основы SQL в
PostgreSQL»**

Выполнил студент группы ИНБО-10-23

Гаврилина А.П

Москва 2025

Цель: сформировать навык по созданию БД и заполнению таблиц с помощью Postgres.

Постановка задачи:

1. Установить PostgreSQL;
2. Установить Dbeaver;
3. Сгенерировать запросы CREATE TABLE по Практической работе № 5. Показать скриншотами;
4. Заполнить созданные таблицы с помощью запроса INSERT INTO;
5. Вывести на экран все таблицы SELECT * FROM Table1.

Выполнение практической работы

В рамках практической работы был установлен PostgreSQL и Dbeaver.

Для генерации запросов была открыта физическая модель данных в ChartDB (<https://chartdb.io/>) и экспортирована, в результате чего мы получили SQL скрипт (см. рис. 1).

Экспорт SQL ×

Экспортируйте схему диаграммы в SQL скрипт

```
1
2
3 CREATE TABLE order (
4   id_order bigint NOT NULL PRIMARY KEY,
5   id_employee bigint NOT NULL,
6   id_client bigint NOT NULL,
7   date date NOT NULL,
8   status varchar(500) NOT NULL,
9   sum decimal NOT NULL,
10  id_loyalty_card bigint,
11  id_coupon bigint
12 );
13
14
15 CREATE TABLE loyalty_level (
16   id_level bigint NOT NULL PRIMARY KEY,
17   level_name varchar(500) NOT NULL,
18   discount_amount decimal NOT NULL
19 );
20
```

Закрыть

Рисунок 1 – Экспорт SQL скрипта

Теперь создадим таблицы, вставив экспортированный скрипт в DBeaver (см. рис. 2-3).



















▼  gavriline_ap	
▼  Таблицы	
>  category	16K
>  client	8K
>  coupon	16K
>  document	8K
>  employee	8K
>  loyalty_card	8K
>  loyalty_level	16K
>  order_document	8K
>  order_product	8K
>  orders	16K
>  position	16K
>  product	16K
>  product_storage	8K
>  storage	8K
>  supplier	8K
>  transaction	8K

Рисунок 2 – Созданные таблицы

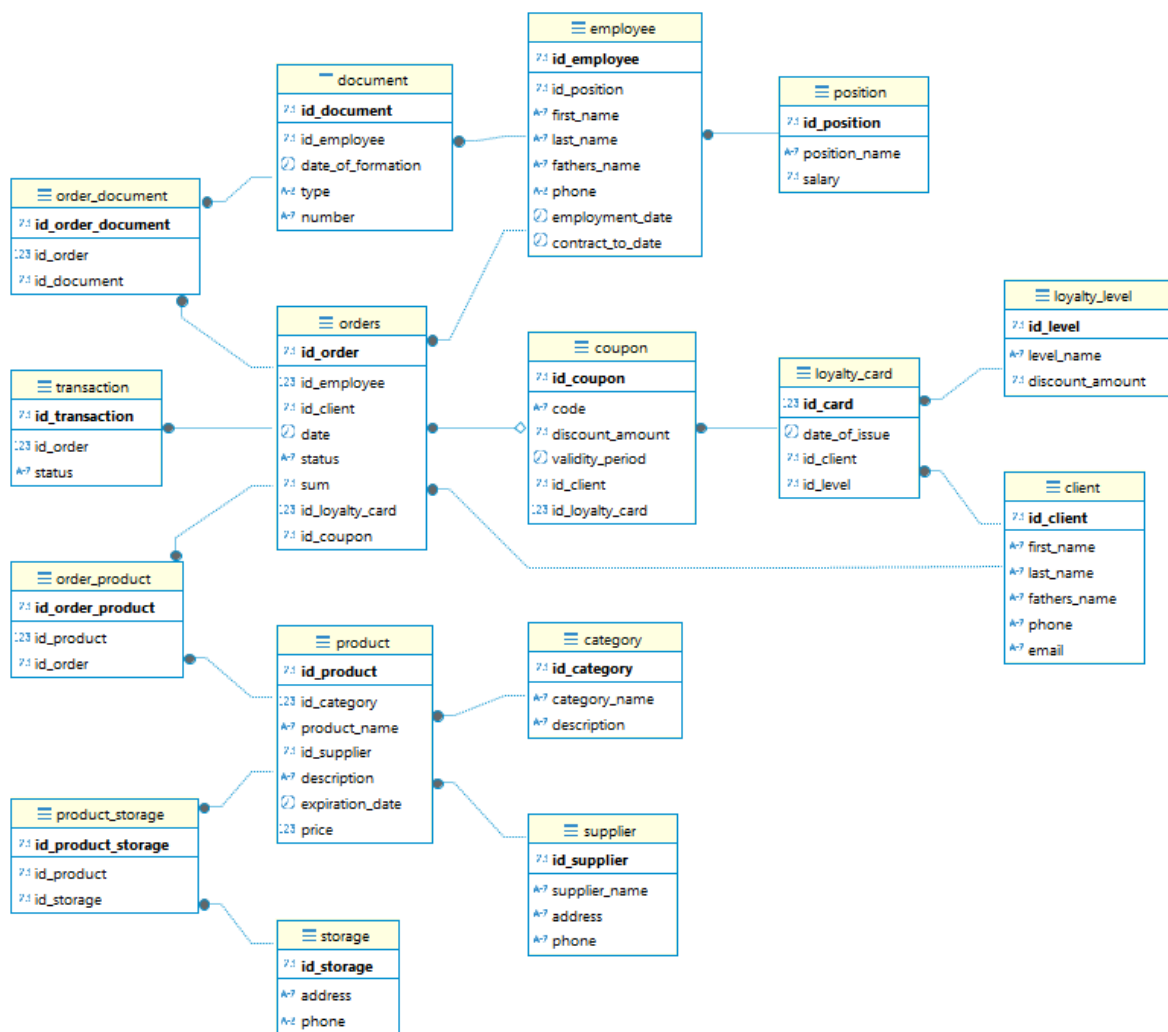


Рисунок 3 – Созданные таблицы

Заполним таблицы INSERT-скриптом (см. рис. 4-6).

```

*dbstud> Script
*dbstud> Script-1 x

-- INSERT INTO category (id_category, category_name, description) VALUES
(1, 'Кофе', 'Зерновой, молотый и растворимый кофе различных сортов'),
(2, 'Чай', 'Чёрный, зелёный, травяной и фруктовый чай'),
(3, 'Сиропы', 'Сиропы для кофе и напитков'),
(4, 'Аксессуары', 'Кофемолки, турки, чайники и другие аксессуары'),
(5, 'Десерты', 'Печенье, конфеты и другие сладости к чаю и кофе');

-- INSERT INTO supplier (id_supplier, supplier_name, address, phone) VALUES
(1, 'Coffee Masters', 'г. Москва, ул. Промышленная, 15', '+74951234567'),
(2, 'Tea Empire', 'г. Санкт-Петербург, ул. Чайная, 42', '+78129876543'),
(3, 'Sweet Delights', 'г. Казань, ул. Сладкая, 7', '+78432567890'),
(4, 'Brew Accessories', 'г. Екатеринбург, ул. Аксессуарная, 33', '+73435678901');

-- INSERT INTO product (id_product, id_category, product_name, id_supplier, description, expiration_date, price) VALUES
(1, 1, 'Кофе Arabica Premium', 1, '100% арабика, средняя обжарка', '2024-12-31', 1200.00),
(2, 1, 'Кофе Robusta Gold', 1, 'Смесь арабики и робусты', '2024-11-30', 950.00),
(3, 2, 'Чай Earl Grey', 2, 'Чёрный чай с бергамотом', '2025-06-30', 450.00),
(4, 2, 'Чай Зелёный жасминовый', 2, 'Зелёный чай с цветами жасмина', '2025-05-31', 500.00),
(5, 3, 'Сироп ванильный', 3, 'Ванильный сироп для кофе', '2024-10-31', 350.00),
(6, 3, 'Сироп карамельный', 3, 'Карамельный сироп для напитков', '2024-09-30', 350.00),
(7, 4, 'Френч-пресс 1л', 4, 'Стеклянный френч-пресс для кофе', '9999-12-31', 2500.00),
(8, 4, 'Керамическая турка', 4, 'Турка для приготовления кофе', '9999-12-31', 1200.00),
(9, 5, 'Печенье миндальное', 3, 'Хрустящее миндальное печенье', '2024-08-31', 280.00),
(10, 5, 'Шоколадные конфеты', 3, 'Ассорти шоколадных конфет', '2024-07-31', 650.00);

-- INSERT INTO loyalty_level (id_level, level_name, discount_amount) VALUES
(1, 'Стандарт', 0.00),
(2, 'Серебряный', 5.00),
(3, 'Золотой', 10.00),
(4, 'Платиновый', 15.00);

-- INSERT INTO storage (id_storage, address, phone) VALUES
(1, 'г. Москва, ул. Складская, 1', '+74957654321'),
(2, 'г. Санкт-Петербург, ул. Складская, 2', '+78122345678');

```

Рисунок 4 – Заполнение таблиц (часть 1)

```

-- INSERT INTO position (id_position, position_name, salary) VALUES
(1, 'Менеджер', 50000.00),
(2, 'Бариста', 40000.00),
(3, 'Кассир', 35000.00),
(4, 'Администратор', 55000.00);

-- INSERT INTO employee (id_employee, id_position, first_name, last_name, fathers_name, phone, employment_date, contract_to_date) VALUES
(1, 1, 'Анна', 'Иванова', 'Петровна', '+79161234567', '2023-01-15', '2025-01-15'),
(2, 2, 'Дмитрий', 'Смирнов', 'Алексеевич', '+79162345678', '2023-03-10', '2024-12-31'),
(3, 3, 'Елена', 'Кузнецова', 'Сергеевна', '+79163456789', '2023-05-20', '2024-12-31'),
(4, 4, 'Иван', 'Петров', 'Олегович', '+79164567890', '2022-11-01', '2025-11-01');

-- INSERT INTO client (id_client, first_name, last_name, fathers_name, phone, email) VALUES
(1, 'Ольга', 'Соколова', 'Михайловна', '+79165678901', 'olga.sokolova@mail.ru'),
(2, 'Алексей', 'Волков', 'Дмитриевич', '+79166789012', 'alex.volkov@gmail.com'),
(3, 'Мария', 'Лебедева', 'Андреевна', '+79167890123', 'maria.lebedeva@yandex.ru'),
(4, 'Сергей', 'Козлов', 'Игоревич', '+79168901234', 'sergey.kozlov@mail.ru');

-- INSERT INTO loyalty_card (id_card, date_of_issue, id_client, id_level) VALUES
(1, '2023-06-01', 1, 3),
(2, '2023-07-15', 2, 2),
(3, '2023-08-20', 3, 1),
(4, '2023-09-05', 4, 4);

-- INSERT INTO coupon (id_coupon, code, discount_amount, validity_period, id_client, id_loyalty_card) VALUES
(1, 'COFFEE10', 10.00, '2024-06-01', 1, 1),
(2, 'TEA5', 5.00, '2024-07-15', 2, 2),
(3, 'SWEET15', 15.00, '2024-09-05', 4, 4);

-- INSERT INTO product_storage (id_product, id_storage) VALUES
(1, 1), (1, 2), (2, 1), (3, 1), (3, 2), (4, 2),
(5, 1), (6, 1), (6, 2), (7, 1), (8, 2), (9, 1), (10, 2);

```

Рисунок 5 – Заполнение таблиц (часть 2)

```

INSERT INTO orders (id_order, id_employee, id_client, date, status, sum, id_loyalty_card, id_coupon) VALUES
(1, 3, 1, '2023-10-01', 'Выполнен', 2350.00, 1, NULL),
(2, 3, 2, '2023-10-02', 'Выполнен', 1800.00, 2, 2),
(3, 1, 4, '2023-10-03', 'В обработке', 4150.00, 4, 3),
(4, 2, 3, '2023-10-04', 'Отменен', 950.00, NULL, NULL);

INSERT INTO order_product (id_product, id_order) VALUES
(1, 1), (5, 1), (3, 2), (4, 2), (7, 3), (10, 3), (2, 4);

INSERT INTO transaction (id_order, status) VALUES
(1, 'Оплачено'),
(2, 'Оплачено'),
(3, 'Ожидает оплаты'),
(4, 'Отменено');

INSERT INTO document (id_document, id_employee, date_of_formation, type, number) VALUES
(1, 1, '2023-10-01', 'Накладная', 'НК-2023-1001'),
(2, 1, '2023-10-02', 'Накладная', 'НК-2023-1002'),
(3, 4, '2023-10-03', 'Счёт', 'СЧ-2023-1003'),
(4, 4, '2023-10-04', 'Акт возврата', 'АВ-2023-1004');

INSERT INTO order_document (id_order, id_document) VALUES
(1, 1), (2, 2), (3, 3), (4, 4);

```

Рисунок 6 – Заполнение таблиц (часть 3)

Затем были созданы SELECT-скрипты для вывода таблиц (см. рис. 7-11).

```

SELECT
  c.id_client AS "ID клиента",
  c.last_name AS "Фамилия",
  c.first_name AS "Имя",
  c.phone AS "Телефон",
  ll.level_name AS "Уровень лояльности",
  ll.discount_amount AS "Скидка (%)",
  lc.date_of_issue AS "Дата выдачи карты"
FROM
  client c
LEFT JOIN
  loyalty_card lc ON c.id_client = lc.id_client
LEFT JOIN
  loyalty_level ll ON lc.id_level = ll.id_level
ORDER BY
  c.last_name, c.first_name;

```

client(+) 1 ×

Введите SQL выражение чтобы отфильтровать результаты

ID клиента	Фамилия	Имя	Телефон	Уровень лояльности	Скидка (%)	Дата выдачи карты
2	Волков	Алексей	+79166789012	Серебряный	5	2023-07-15
4	Козлов	Сергей	+79168901234	Платиновый	15	2023-09-05
3	Лебедева	Мария	+79167890123	Стандарт	0	2023-08-20
1	Соколова	Ольга	+79165678901	Золотой	10	2023-06-01

Рисунок 7 – Просмотр таблицы клиентов с их картами лояльности

SELECT

```

p.id_product AS "ID товара",
p.product_name AS "Название товара",
c.category_name AS "Категория",
s.supplier_name AS "Поставщик",
p.price AS "Цена",
p.expiration_date AS "Срок годности",
p.description AS "Описание"
FROM
    product p
JOIN
    category c ON p.id_category = c.id_category
JOIN
    supplier s ON p.id_supplier = s.id_supplier
ORDER BY
    c.category_name, p.product_name;

```

product(+) 1 ×

SELECT p.id_product AS "ID товара", p.product_name AS "Название товара", c.category_name AS "Категория", s.supplier_name AS "Поставщик", p.price AS "Цена", p.expiration_date AS "Срок годности", p.description AS "Описание"

	ID товара	Название товара	Категория	Поставщик	Цена	Срок годности	Описание
1	8	Керамическая турка	Аксессуары	Brew Accessories	1 200	9999-12-31	Турка для приготовления кофе
2	7	Френч-пресс 1л	Аксессуары	Brew Accessories	2 500	9999-12-31	Стеклянный френч-пресс для кофе
3	9	Печенье миндальное	Десерты	Sweet Delights	280	2024-08-31	Хрустящее миндальное печенье
4	10	Шоколадные конфеты	Десерты	Sweet Delights	650	2024-07-31	Ассорти шоколадных конфет
5	1	Кофе Arabica Premium	Кофе	Coffee Masters	1 200	2024-12-31	100% арабика, средняя обжарка
6	2	Кофе Robusta Gold	Кофе	Coffee Masters	950	2024-11-30	Смесь арабики и робусты
7	5	Сироп ванильный	Сиропы	Sweet Delights	350	2024-10-31	Ванильный сироп для кофе
8	6	Сироп карамельный	Сиропы	Sweet Delights	350	2024-09-30	Карамельный сироп для напитков
9	3	Чай Earl Grey	Чай	Tea Empire	450	2025-06-30	Чёрный чай с бергамотом
10	4	Чай Зелёный жасминовый	Чай	Tea Empire	500	2025-05-31	Зелёный чай с цветами жасмина

Рисунок 8 – Просмотр таблицы товаров с их категориями и поставщиками

SELECT

```

o.id_order AS "Номер заказа",
o.date AS "Дата заказа",
CONCAT(c.last_name, ' ', c.first_name) AS "Клиент",
CONCAT(e.last_name, ' ', e.first_name) AS "Сотрудник",
o.sum AS "Сумма заказа",
o.status AS "Статус",
ll.level_name AS "Уровень лояльности",
cp.discount_amount AS "Скидка по купону"
FROM
    orders o
JOIN
    client c ON o.id_client = c.id_client
JOIN
    employee e ON o.id_employee = e.id_employee
LEFT JOIN
    loyalty_card lc ON o.id_loyalty_card = lc.id_card
LEFT JOIN
    loyalty_level ll ON lc.id_level = ll.id_level
LEFT JOIN
    coupon cp ON o.id_coupon = cp.id_coupon
ORDER BY
    o.date DESC;

```

orders(+) 1 ×

SELECT o.id_order AS "Номер заказа", o.date AS "Дата заказа", c.last_name AS "Клиент", e.first_name AS "Сотрудник", o.sum AS "Сумма заказа", o.status AS "Статус", ll.level_name AS "Уровень лояльности", cp.discount_amount AS "Скидка по купону"

	Номер заказа	Дата заказа	Клиент	Сотрудник	Сумма заказа	Статус	Уровень лояльности	Скидка по купону
1	4	2023-10-04	Лебедева Мария	Смирнов Дмитрий	950	Отменен	[NULL]	[NULL]
2	3	2023-10-03	Козлов Сергей	Иванова Анна	4 150	В обработке	Платиновый	15
3	2	2023-10-02	Волков Алексей	Кузнецова Елена	1 800	Выполнен	Серебряный	5
4	1	2023-10-01	Соколова Ольга	Кузнецова Елена	2 350	Выполнен	Золотой	[NULL]

Рисунок 9 – Просмотр таблицы заказов с клиентами, сотрудниками и суммами

```

SELECT
  o.id_order AS "Номер заказа",
  o.date AS "Дата заказа",
  p.product_name AS "Товар",
  c.category_name AS "Категория",
  p.price AS "Цена за единицу",
  COUNT(op.id_product) AS "Количество",
  (p.price * COUNT(op.id_product)) AS "Общая стоимость"
FROM
  orders o
JOIN
  order_product op ON o.id_order = op.id_order
JOIN
  product p ON op.id_product = p.id_product
JOIN
  category c ON p.id_category = c.id_category
GROUP BY
  o.id_order, o.date, p.product_name, c.category_name, p.price
ORDER BY
  o.id_order;

```

	123 Номер заказа	Дата заказа	А2 Товар	А2 Категория	123 Цена за единицу	123 Количество	123 Общая стоимость
1	1	2023-10-01	Сироп ванильный	Сиропоы	350	1	350
2	1	2023-10-01	Кофе Arabica Premium	Кофе	1 200	1	1 200
3	2	2023-10-02	Чай Earl Grey	Чай	450	1	450
4	2	2023-10-02	Чай Зелёный жасминовый	Чай	500	1	500
5	3	2023-10-03	Френч-пресс 1л	Аксессуары	2 500	1	2 500
6	3	2023-10-03	Шоколадные конфеты	Десерты	650	1	650
7	4	2023-10-04	Кофе Robusta Gold	Кофе	950	1	950

Рисунок 10 - Просмотр состава заказов (товары в заказах)

```

SELECT
  e.id_employee AS "ID сотрудника",
  CONCAT(e.last_name, ' ', e.first_name) AS "ФИО",
  p.position_name AS "Должность",
  p.salary AS "Зарплата",
  e.employment_date AS "Дата приема",
  e.contract_to_date AS "Дата окончания контракта",
  e.phone AS "Телефон"
FROM
  employee e
JOIN
  position p ON e.id_position = p.id_position
ORDER BY
  e.last_name, e.first_name;

```

	123 ID сотрудника	А2 ФИО	А2 Должность	123 Зарплата	Дата приема	Дата окончания контракта	А2 Телефон
1	1	Иванова Анна	Менеджер	50 000	2023-01-15	2025-01-15	+79161234567
2	3	Кузнецова Елена	Кассир	35 000	2023-05-20	2024-12-31	+79163456789
3	4	Петров Иван	Администратор	55 000	2022-11-01	2025-11-01	+79164567890
4	2	Смирнов Дмитрий	Бариста	40 000	2023-03-10	2024-12-31	+79162345678

Рисунок 11 - Просмотр таблицы сотрудников с их должностями

Вывод

В результате выполнения практической работы были успешно освоены ключевые этапы работы с реляционной базой данных в PostgreSQL. На практике отработаны следующие навыки: установка и настройка PostgreSQL и DBeaver для работы с базами данных, создание структуры БД с помощью SQL-запросов «CREATE TABLE», наполнение таблиц данными с использованием «INSERT INTO», проверка корректности структуры и заполнения БД через «SELECT»-запросы.