МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

"Санкт-Петербургский национальный исследовательский университет информационных технологий механики и оптики"

Мегафакультет: трансляционных информационных технологий

Факультет: информационных технологий и программирования

Лабораторная работа №5

По дисциплине: "Проектирование баз данных"

Тема: "Партицирование в PostgreSQL"

Выполнила студент группы №М3216: *Шевцов Роман Сергеевич*

Задача:

Сделать партицирование одной таблицы через наследование.

Требования:

- 1. Таблица должна быть разбита на не менее, чем 3 партиции.
- 2. В каждой партиции должно быть не менее 5 записей.

Порядок выполнения работы:

- 1. Выберите таблицу, которую можно разбить на партиции. Определите условия разбиения таблицы на партиции.
- 2. Создайте таблицы-партиции.
- 3. Создайте функцию, обеспечивающую партицирование.
- 4. Подключите функцию к мастер-таблице.
- 5. Перенесите данные из мастер-таблицы в партиции.
- 6. Очистите мастер-таблицу и добавьте в нее новые данные.

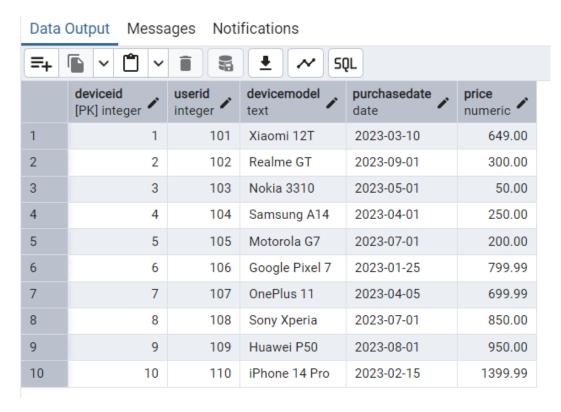
На защите лабораторной необходимо будет продемонстрировать мастер-таблицу, таблицы-партиции, функцию, обеспечивающую партицирование, результаты выполнения запросов для добавления новых записей и чтения таблиц.

Решение:

Создаем новую мастер-таблицу и вставляем в нее значения:

```
Query Query History
1 ➤ CREATE TABLE device (
      deviceid SERIAL PRIMARY KEY,
       userid INTEGER,
       devicemodel TEXT,
5
       purchasedate DATE,
6
        price NUMERIC
7 );
8
9 • INSERT INTO device (userid, devicemodel, purchasedate, price)
     (101, 'Xiaomi 12T', '2023-03-10', 649.00),
11
12
      (102, 'Realme GT', '2023-09-01', 300.00),
     (103, 'Nokia 3310', '2023-05-01', 50.00),
13
14 (104, 'Samsung A14', '2023-04-01', 250.00),
     (105, 'Motorola G7', '2023-07-01', 200.00),
15
     (106, 'Google Pixel 7', '2023-01-25', 799.99),
16
17
     (107, 'OnePlus 11', '2023-04-05', 699.99),
     (108, 'Sony Xperia', '2023-07-01', 850.00), (109, 'Huawei P50', '2023-08-01', 950.00),
18
19
      (110, 'iPhone 14 Pro', '2023-02-15', 1399.99);
20
21
22 SELECT * FROM device;
```

Проверяем вновь созданную таблицу:



Создаем партиции по диапазону цен:

```
Query Query History
1 		 CREATE TABLE device_low_price (
        CHECK (price < 700)
3
     ) INHERITS (device);
4
5 	✓ CREATE TABLE device_medium_price (
        CHECK (price >= 700 AND price < 1000)
6
     ) INHERITS (device);
7
8
CHECK (price >= 1000)
10
    ) INHERITS (device);
11
Data Output Messages Notifications
CREATE TABLE
Query returned successfully in 149 msec.
```

```
Query Query History
1 		 CREATE OR REPLACE FUNCTION device_insert_trigger()
2 RETURNS TRIGGER AS $$
3
   BEGIN
4
        IF (NEW.price < 700) THEN</pre>
5
             INSERT INTO device_low_price VALUES (NEW.*);
6 ~
       ELSIF (NEW.price >= 700 AND NEW.price < 1000) THEN
7
             INSERT INTO device_medium_price VALUES (NEW.*);
       ELSIF (NEW.price >= 1000) THEN
9
             INSERT INTO device_high_price VALUES (NEW.*);
10 🗸
       ELSE
             RAISE EXCEPTION 'Invalid price value: %', NEW.price;
11
       END IF;
12
13
       RETURN NULL;
14 END;
15  $$ LANGUAGE plpgsql;
Data Output Messages Notifications
CREATE FUNCTION
Query returned successfully in 68 msec.
```

Создаем и подключаем триггер к мастер-таблице:

```
Query Query History

1  CREATE TRIGGER device_insert_trigger

2  BEFORE INSERT ON device

3  FOR EACH ROW EXECUTE FUNCTION device_insert_trigger();

Data Output Messages Notifications

CREATE TRIGGER

Query returned successfully in 59 msec.
```

Переносим данные из мастер-таблицы в партиции и очищаем мастер-таблицу:

```
Query Query History

1  INSERT INTO device_low_price
2  SELECT * FROM device WHERE price < 700;

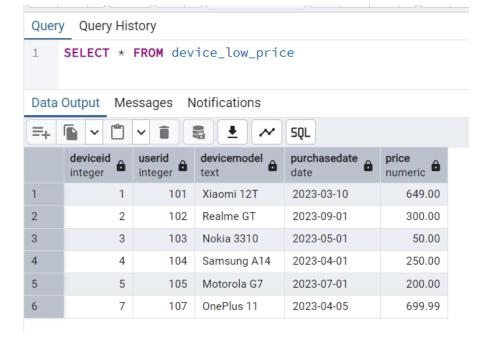
3  INSERT INTO device_medium_price
5  SELECT * FROM device WHERE price >= 700 AND price < 1000;

6  INSERT INTO device_high_price
8  SELECT * FROM device WHERE price >= 1000;

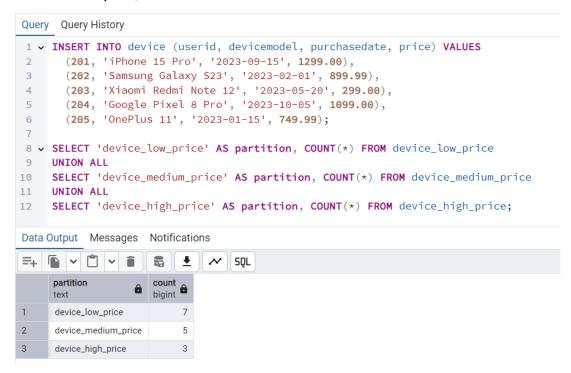
9  TRUNCATE ONLY device;

Data Output Messages Notifications

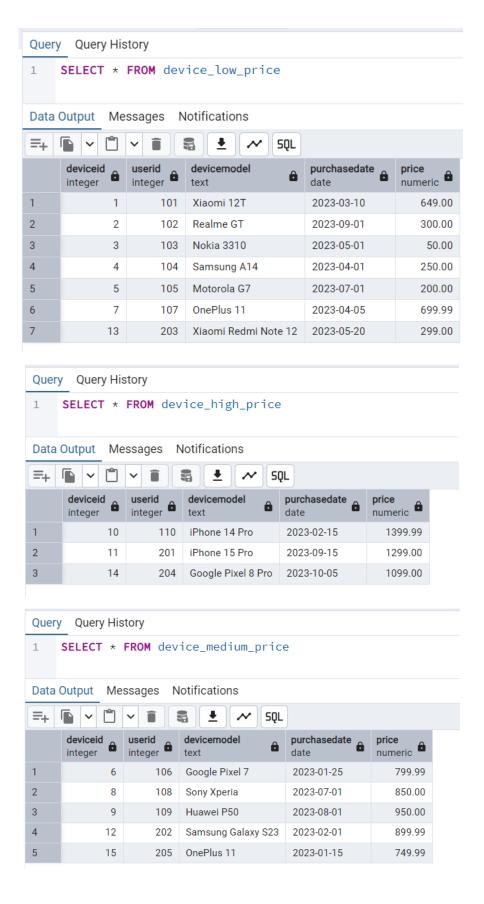
TRUNCATE TABLE
```



Вставим новые данные в мастер-таблицу и проверим что они автоматически добавляются в партиции:



Вновь проверим партицию device_low_price и остальные, убедимся что данные обновляются:



Поскольку в условии лабораторной просят не менее 5 записей в каждой партиции, добавим пару записей в device_high_price:

```
Query Query History

1 V INSERT INTO device (userid, devicemodel, purchasedate, price) VALUES

2 (206, 'Samsung Galaxy Z Fold5', '2023-08-10', 1799.00),

3 (207, 'iPhone 15 Pro Max', '2023-09-22', 1599.00),

4 (208, 'Google Pixel Fold', '2023-06-27', 1799.00),

5 (209, 'Huawei Mate X3', '2023-04-15', 1999.00);

Data Output Messages Notifications

INSERT 0 0

Query returned successfully in 174 msec.
```

Проверим количество записей в каждой партиции:

