

# Лабораторная работа 5

**Задача: сделать партицирование одной таблицы через наследование**

## Требования:

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

## Ход выполнения работы:

### Немного теории:



Партицирование применяется в тех случаях, когда одна из таблиц в базе данных сильно выросла, следовательно выросло время выполнения запросов. Таким образом, мы рассматриваем нашу мастер-таблицу как батон хлеба, а партиции - кусочки нашего батона.



Выделяют два вида партицирования: декларативное партицирование и партицирование через наследование. Декларативное партицирование имеет ряд минусов перед партицированием через наследование;

### Для начала создадим таблицу

```
CREATE TABLE sales (  
    id SERIAL PRIMARY KEY,  
    product_name text,  
    amount INT,  
    sale_date DATE  
);
```





Создали таблицу продаж, которая отображает id операции, название товара, дату продажи и количество проданных штук.

### Добавим данные в таблицу

```
INSERT INTO sales (product_name, amount, sale_date) VALUES  
( 'Бутылка воды', 3, '2022-03-15'),  
( 'Мангал', 1, '2022-03-15'),  
( 'Шашлык', 10, '2022-03-15'),  
( 'Яблоко', 5, '2022-03-15'),  
( 'Груша', 7, '2022-03-15'),  
( 'Яблоко', 20, '2023-03-15'),  
( 'Ананас', 4, '2023-03-15'),  
( 'Помидор', 17, '2023-03-15'),  
( 'Огурец', 11, '2023-03-15'),  
( 'Батончик Twix', 3, '2023-03-15'),
```

```
( 'Ягода', 10, '2024-03-15'),
( 'Груша', 17, '2024-03-15'),
( 'Помидор', 100, '2024-03-15'),
( 'Цветок', 101, '2024-03-15'),
( 'Бутылка воды', 2, '2024-03-15'),
( 'Лапша', 3, '2024-03-15');
```

Теперь таблица выглядит следующим образом:

	 id ▾	 product_name ▾	 amount ▾	 sale_date ▾
1	1	Бутылка воды	3	2022-03-15
2	2	Мангал	1	2022-03-15
3	3	Шашлык	10	2022-03-15
4	4	Яблоко	5	2022-03-15
5	5	Груша	7	2022-03-15
6	6	Яблоко	20	2023-03-15
7	7	Ананас	4	2023-03-15
8	8	Помидор	17	2023-03-15
9	9	Огурец	11	2023-03-15
10	10	Батончик Twix	3	2023-03-15
11	11	Ягода	10	2024-03-15
12	12	Груша	17	2024-03-15
13	13	Помидор	100	2024-03-15
14	14	Цветок	101	2024-03-15
15	15	Бутылка воды	2	2024-03-15
16	16	Лапша	3	2024-03-15

Будет партицировать эту таблицу по году, в который была совершена покупка

### Создали таблицы-партици

```
CREATE TABLE sales_2022 (
    CHECK (sale_date >= '2022-01-01'::DATE AND sale_date <= '2022-12-31'::DATE)
) INHERITS (sales);

CREATE TABLE sales_2023 (
    CHECK (sale_date >= '2023-01-01'::DATE AND sale_date <= '2023-12-31'::DATE)
) INHERITS (sales);

CREATE TABLE sales_2024 (
```

```
        CHECK (sale_date >= '2024-01-01'::DATE AND sale_date <= '2024-12-31'::DATE)
    ) INHERITS (sales);
```

## Создали функцию, обеспечивающую партиционирование

```
CREATE OR REPLACE FUNCTION sales_insert_function()
RETURNS TRIGGER AS $$
BEGIN
    IF (NEW.sale_date >= '2022-01-01'::DATE AND NEW.sale_date < '2023-01-01'::DATE) THEN
        INSERT INTO sales_2022 VALUES (NEW.*);
    ELSIF (NEW.sale_date >= '2023-01-01'::DATE AND NEW.sale_date < '2024-01-01'::DATE) THEN
        INSERT INTO sales_2023 VALUES (NEW.*);
    ELSIF (NEW.sale_date >= '2024-01-01'::DATE AND NEW.sale_date < '2025-01-01'::DATE) THEN
        INSERT INTO sales_2024 VALUES (NEW.*);
    ELSE
        RAISE EXCEPTION 'ERROR! DATE OUT OF RANGE!';
    END IF;
    RETURN NULL;
END;
$$ LANGUAGE plpgsql;
```

## Подключили функцию партиционирования к таблице

```
CREATE TRIGGER insert_sales_trigger
BEFORE INSERT ON sales
FOR EACH ROW EXECUTE FUNCTION sales_insert_function();
```

## Перенесли данные из мастер-таблицы в партиции

```
WITH moved_rows AS (
    DELETE FROM ONLY sales
    WHERE sale_date >= '2022-01-01' AND sale_date < '2023-01-01'
    RETURNING *
)
INSERT INTO sales_2022
SELECT * FROM moved_rows;

WITH moved_rows AS (
    DELETE FROM ONLY sales
    WHERE sale_date >= '2023-01-01' AND sale_date < '2024-01-01'
    RETURNING *
)
INSERT INTO sales_2023
SELECT * FROM moved_rows;
```

```

WITH moved_rows AS (
    DELETE FROM ONLY sales
    WHERE sale_date >= '2024-01-01' AND sale_date < '2025-01-01'
    RETURNING *
)
INSERT INTO sales_2024
SELECT * FROM moved_rows;

```

Таблицы после применения функция

sales\_2022:

	id	product_name	amount	sale_date
1	1	Бутылка воды	3	2022-03-15
2	2	Мангал	1	2022-03-15
3	3	Шашлык	10	2022-03-15
4	4	Яблоко	5	2022-03-15
5	5	Груша	7	2022-03-15

sales\_2023:

	id	product_name	amount	sale_date
1	6	Яблоко	20	2023-03-15
2	7	Ананас	4	2023-03-15
3	8	Помидор	17	2023-03-15
4	9	Огурец	11	2023-03-15
5	10	Батончик Twix	3	2023-03-15

sales\_2024:

	id	product_name	amount	sale_date
1	11	Ягода	10	2024-03-15
2	12	Груша	17	2024-03-15
3	13	Помидор	100	2024-03-15
4	14	Цветок	101	2024-03-15
5	15	Бутылка воды	2	2024-03-15
6	16	Лапша	3	2024-03-15









## Очищаем мастер-таблицу

```
TRUNCATE TABLE sales;
```

Вместе с ней очистились и созданные таблицы

## Добавление новых данных в таблицу sales ведет к автоматическому добавлению в соответствующую партицию

```
INSERT INTO sales (product_name, amount, sale_date) VALUES  
( 'банка воды', 3, '2022-03-15');
```

	 id 	 product_name 	 amount 	 sale_date 
1	17	Бутылка воды	3	2022-03-15
2	18	Бутылка воды	3	2022-03-15
3	19	Мангал	1	2022-03-15
4	20	Шашлык	10	2022-03-15
5	21	Яблоко	5	2022-03-15
6	22	Груша	7	2022-03-15
7	34	банка воды	3	2022-03-15