



PostgreSQL

Использование табличных выражений

Производные таблицы (Derived Tables)

- Производные таблицы — это именованные выражения запроса, созданные во внешнем операторе SELECT
 - Не хранится в базе данных — представляет собой виртуальную реляционную таблицу
 - Область видимости — запрос, в котором она определена

Производная таблица должна

- Иметь псевдоним
- Иметь уникальные имена для всех столбцов
- Не ссылаться несколько раз в одном запросе

Производная таблица может

- Использовать внутренние или внешние псевдонимы для столбцов
- Быть вложенной в другие производные таблицы



PostgreSQL

Использование производных таблиц SELECT

```
SELECT orderyear, COUNT(DISTINCT custid) AS cust_count
FROM (
    SELECT date_part('year',orderdate), custid
    FROM "Sales"."Orders") AS derived_year(orderyear, custid)
GROUP BY orderyear;
```

	123 orderyear	123 cust_count
1	2 006	67
2	2 007	86
3	2 008	81

```
SELECT orderyear, COUNT(DISTINCT custid) AS cust_count
FROM (
    SELECT date_part('year',orderdate) AS orderyear, custid
    FROM "Sales"."Orders" as o
    WHERE o.empid=9) AS derived_year
GROUP BY orderyear;
```

	123 orderyear	123 cust_count
1	2 006	5
2	2 007	16
3	2 008	16



Общие табличные выражения

- CTE (Common Table Expression)— это именованные табличные выражения, используемые для разбиения сложных запросов на простые части
 - Можно представить как определения временных таблиц, существующих **ТОЛЬКО ДЛЯ ОДНОГО ЗАПРОСА**
- CTE определяется в предложении WITH
 - Все возвращаемые столбцы должны иметь уникальные имена
 - CTE поддерживают несколько определений
 - На CTE допустимо ссылаться несколько раз **ТОЛЬКО В ОДНОМ** запросе

```
WITH CTE_year AS
(
  SELECT date_part('year',orderdate) AS orderyear, custid
  FROM "Sales"."Orders"
  where empid = 2
) ;
SELECT orderyear, COUNT(DISTINCT custid) AS cust_count
FROM CTE_year
GROUP BY orderyear;
```



	123 orderyear	123 cust_count
1	2 006	15
2	2 007	35
3	2 008	34

Рекурсивный запрос WITH

- Для определения рекурсивного запроса WITH используется указание **RECURSIVE**
- Рекурсивный запрос WITH включает:
 - не рекурсивную часть
 - оператор UNION (или UNION ALL)
 - рекурсивную часть
- Только в рекурсивной части можно обратиться к результату запроса

```
WITH RECURSIVE t(n)
```

```
AS (
```

```
  SELECT (1)
```

```
  UNION ALL
```

```
  SELECT n+1
```

```
  FROM t WHERE n < 10
```

```
)
```

```
SELECT n FROM t;
```

Временная
рабочая таблица

1

3

5

2

4

2

Временная
промежуточная
таблица



Использование производных таблиц VALUES

```
SELECT *  
FROM (VALUES (1, 'expensive', 250),  
            (2, 'middle', 150),  
            (3, 'cheap', 50)  
      ) as t1(id, preorder, desired_price);
```

```
with t1(id, preorder, desired_price)  
as  
(VALUES (1, 'expensive', 250),  
        (2, 'middle', 150),  
        (3, 'cheap', 50)  
 )  
SELECT *  
FROM t1;
```

	id	preorder	desired_price
1	1	expensive	250
2	2	middle	150
3	3	cheap	50



Подзапросы LATERAL

- При необходимости объединить записи исходной таблицы с записями, которые возвращает коррелированный табличный подзапрос или табличная функция необходимо использовать операторы **LATERAL**
 - **CROSS JOIN LATERAL** – аналог INNER JOIN
 - **LEFT JOIN LATERAL** – аналог LEFT JOIN
- Коррелированный табличный подзапрос должен быть указан справа от оператора LATERAL



CROSS JOIN LATERAL

```
select so.companyname,  
       so.country,  
       t.ord_qw  
from "Sales"."Customers" as so  
CROSS JOIN LATERAL  
(select count(*)  
 from "Sales"."Orders" as o  
 where o.custid = so.custid ) as t(ord_qw);
```

```
Select so.companyname,  
       so.country,  
       t.ord_qw  
from "Sales"."Customers" as so,  
LATERAL  
(select count(*)  
 from "Sales"."Orders" as o  
 where o.custid = so.custid ) as t(ord_qw);
```

	companyname	country	ord_qw
1	Customer EEALV	Canada	14
2	Customer JUW XK	Brazil	5
3	Customer KIDPX	Brazil	7
4	Customer YJCBX	Brazil	9
5	Customer YSIQX	USA	11
6	Customer FVXPQ	Venezuela	2
7	Customer IBVRG	Brazil	14



LEFT JOIN LATERAL

```
select so.custid ,so.companyname, so.country,  
       (select count(*) from "Sales"."Orders" where custid = so.custid ) as ord_qw  
from "Sales"."Customers" as so  
where (select count(*) from "Sales"."Orders" where custid = so.custid ) = 0;
```

```
select so.companyname, so.country, t.ord_qw  
from "Sales"."Customers" as so  
LEFT JOIN lateral  
(select count(*) from "Sales"."Orders" where custid = so.custid ) as t(ord_qw) ON true  
where t.ord_qw =0;
```



	123 custid	ABC companyname	ABC country	123 ord_qw
1	22	Customer DTDMMN	Spain	0
2	57	Customer WVAXS	France	0
3	73	Customer JMIKW	Denmark	0



LEFT JOIN LATERAL

```
select  so.custid,  
        so.companyname,  
        so.country,  
        t.orderid  
from    "Sales"."Customers" as so  
LEFT JOIN lateral  
        (select *  
         from "Sales"."Orders" as o  
         where custid = so.custid ) as t ON true  
where t.orderid is null;
```

select so.custid ,so.companyname, so.country, t.orderid from "Sales"."Customers" as so left join lateral (select * from "Sales"."Orders" as o where custid = so.custid) as t ON true where t.orderid is null;

	123 custid	ABC companyname	ABC country	123 orderid
1	57	Customer WVAXS	France	[NULL]
2	73	Customer JMIKW	Denmark	[NULL]
3	22	Customer DTDMMN	Spain	[NULL]