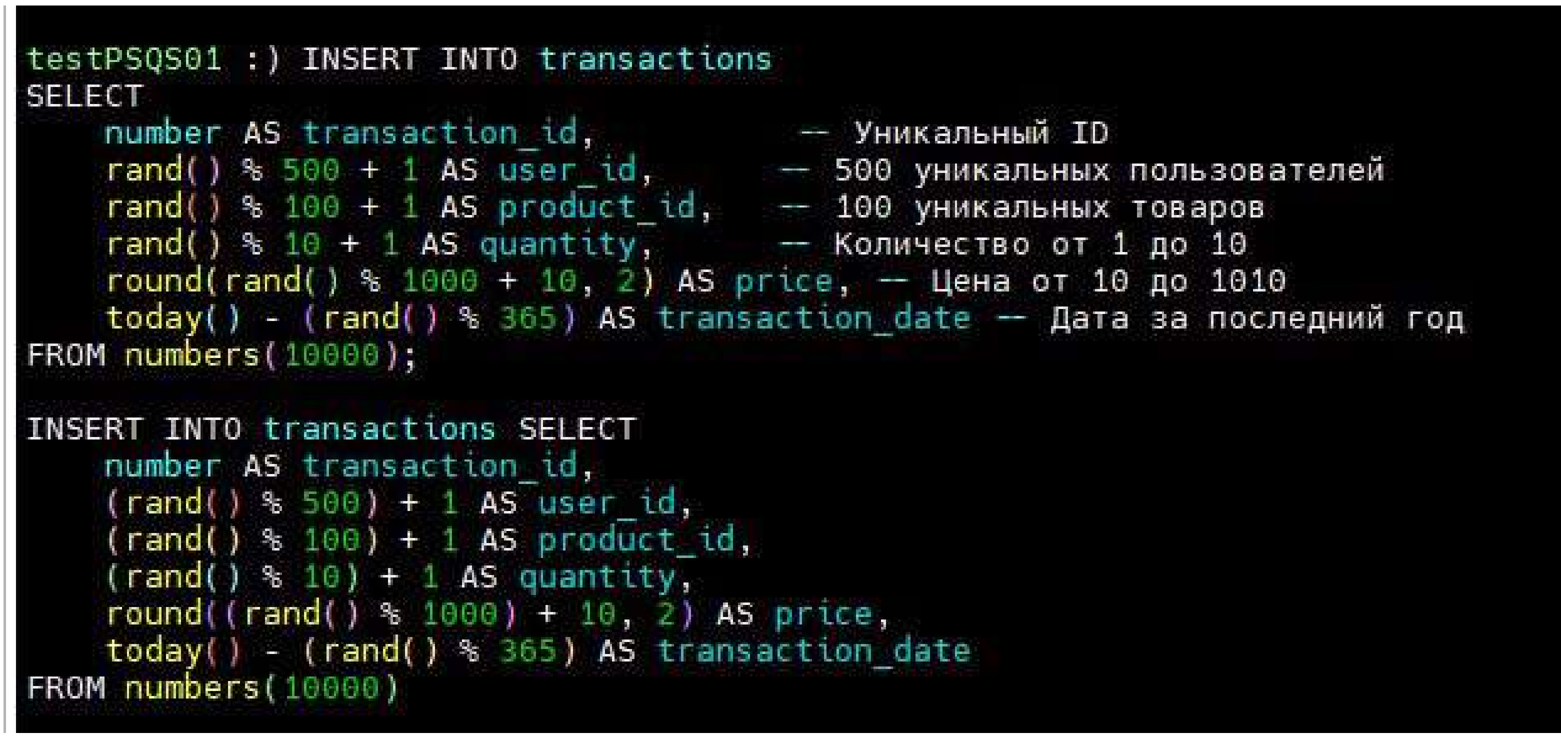
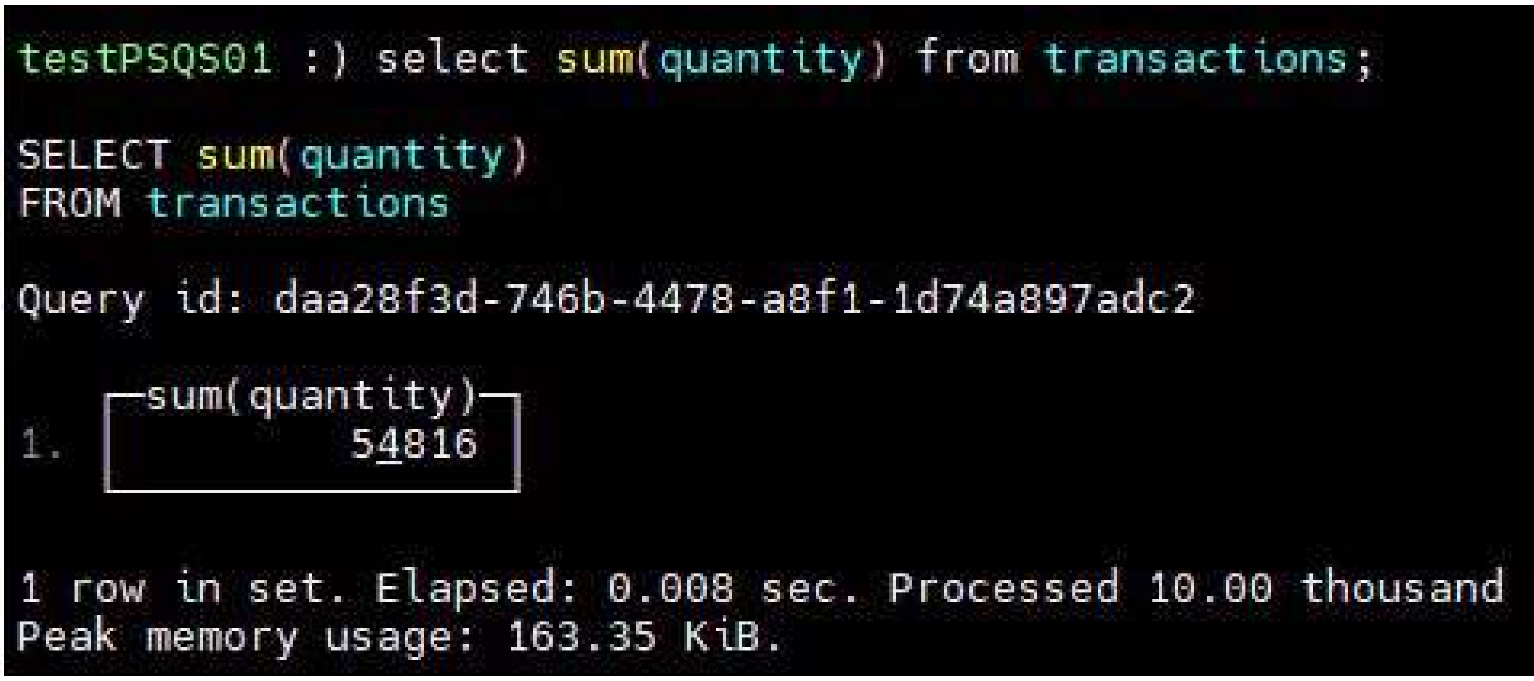
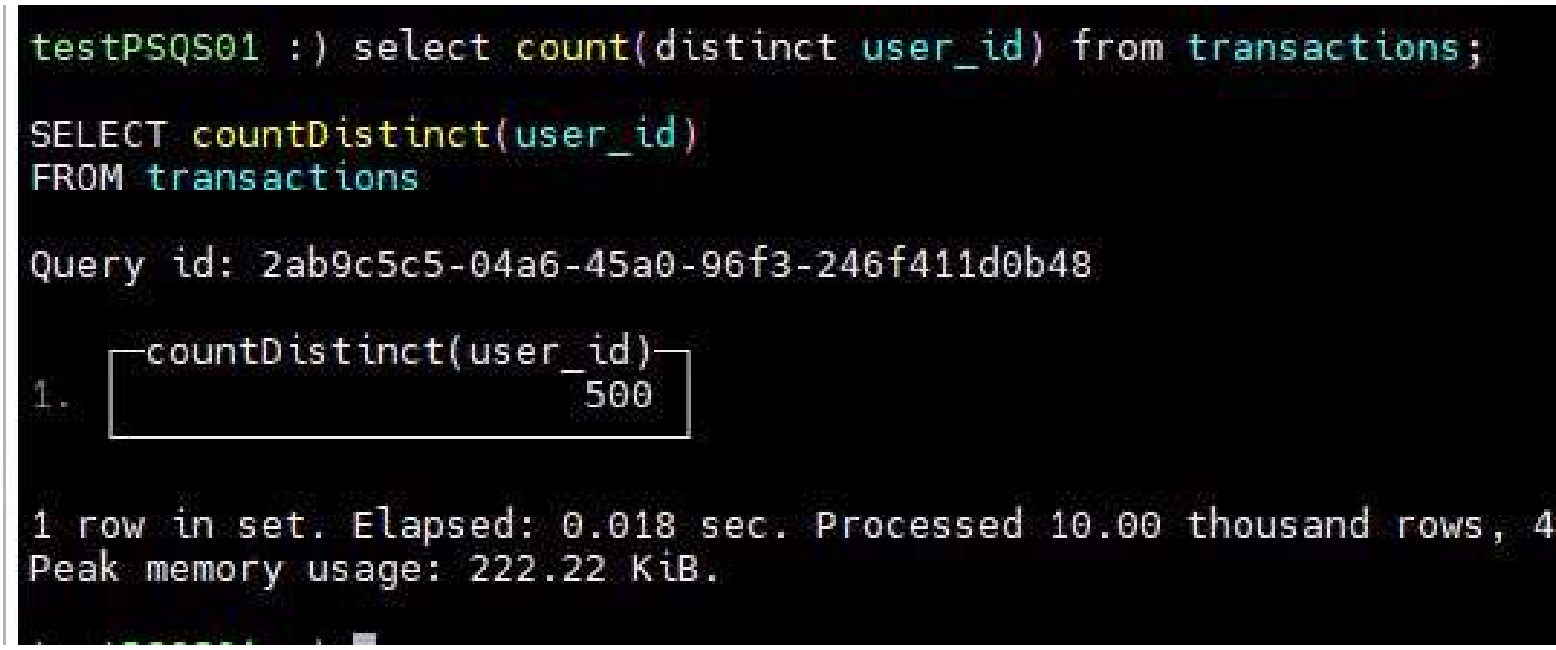


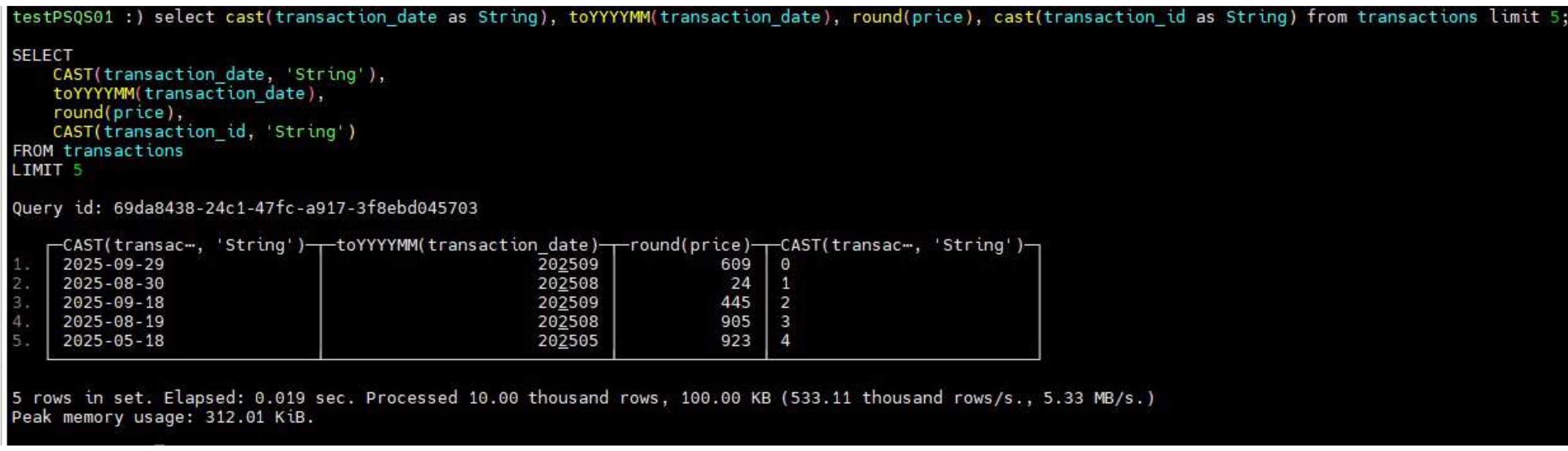
```
testPSQS01 :) INSERT INTO transactions
SELECT
    number AS transaction_id,           -- Уникальный ID
    rand() % 500 + 1 AS user_id,       -- 500 уникальных пользователей
    rand() % 100 + 1 AS product_id,    -- 100 уникальных товаров
    rand() % 10 + 1 AS quantity,       -- Количество от 1 до 10
    round(rand() % 1000 + 10, 2) AS price, -- Цена от 10 до 1010
    today() - (rand() % 365) AS transaction_date -- Дата за последний год
FROM numbers(10000);

INSERT INTO transactions SELECT
    number AS transaction_id,
    (rand() % 500) + 1 AS user_id,
    (rand() % 100) + 1 AS product_id,
    (rand() % 10) + 1 AS quantity,
    round((rand() % 1000) + 10, 2) AS price,
    today() - (rand() % 365) AS transaction_date
FROM numbers(10000)
```



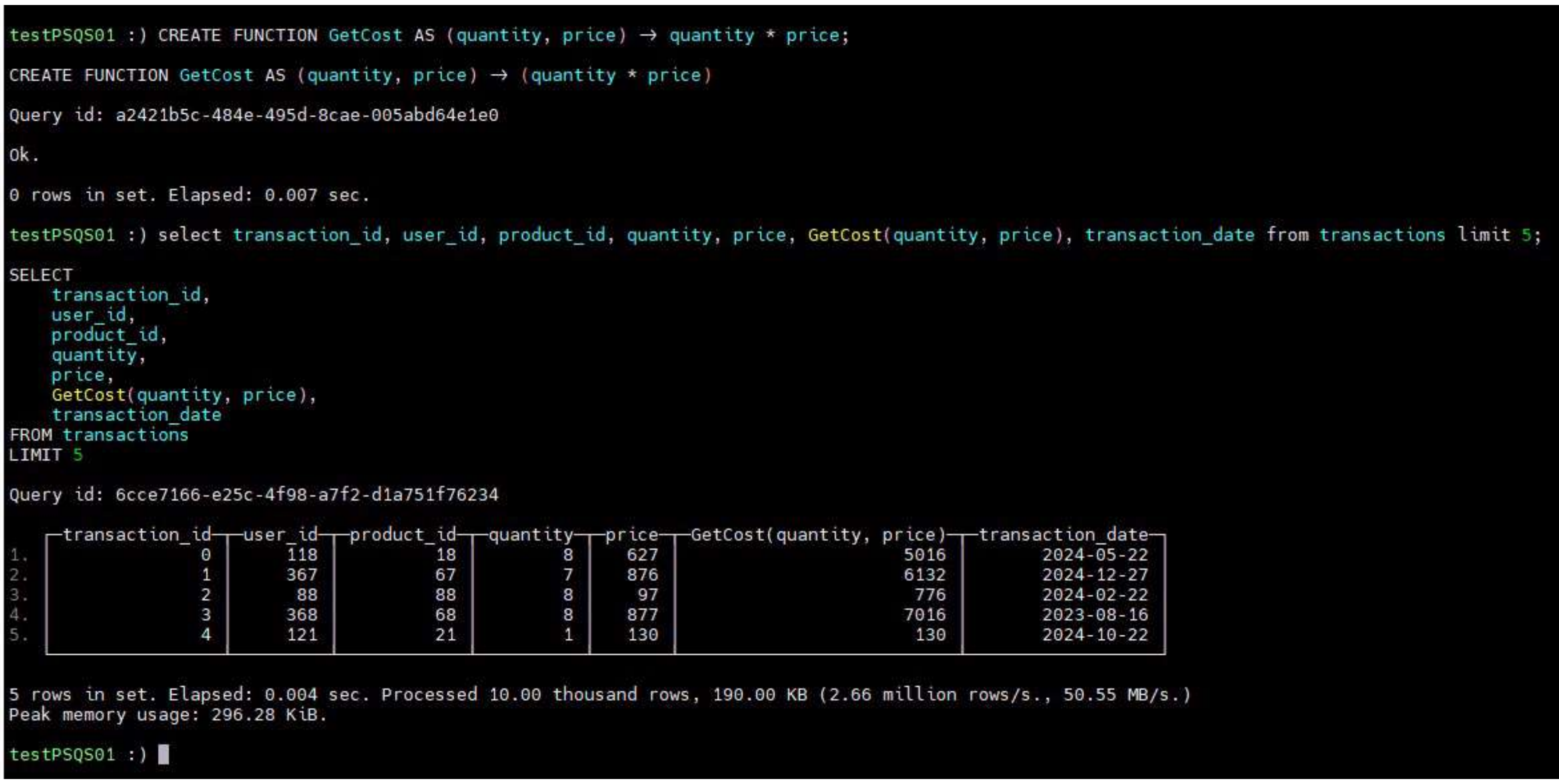
```
testPSQS01 :) select sum(quantity) from transactions;

SELECT sum(quantity)
FROM transactions

Query id: daa28f3d-746b-4478-a8f1-1d74a897adc2

   ┌─sum(quantity)─┐
1. │         54816 │
   └───────────────┘

1 row in set. Elapsed: 0.008 sec. Processed 10.00 thousand
Peak memory usage: 163.35 KiB.
```



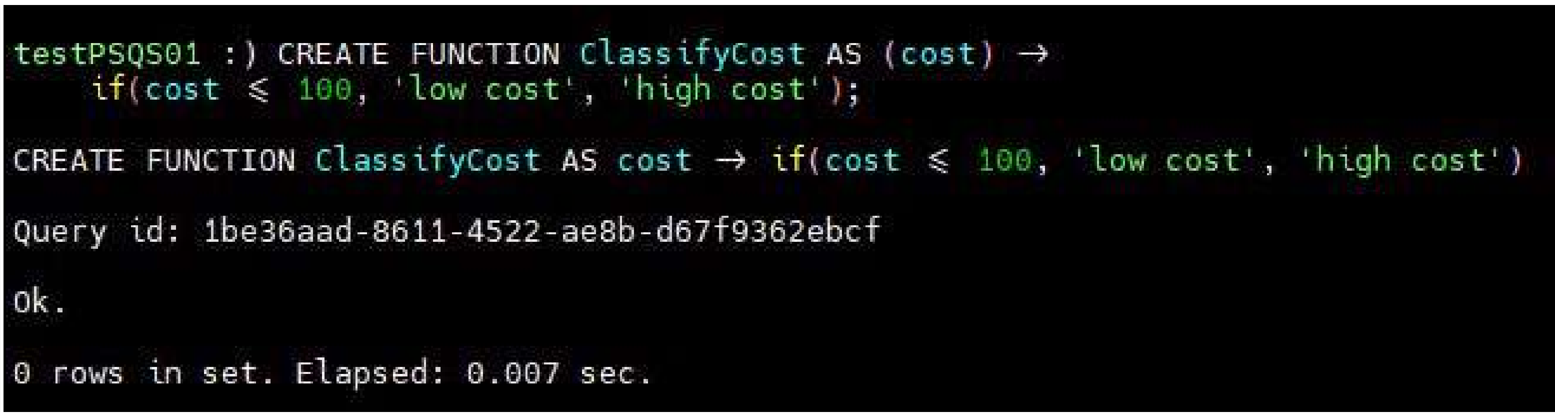
```
testPSQS01 :) select count(distinct user_id) from transactions;

SELECT countDistinct(user_id)
FROM transactions

Query id: 2ab9c5c5-04a6-45a0-96f3-246f411d0b48

   ┌─countDistinct(user_id)─┐
1. │                    500 │
   └────────────────────────┘

1 row in set. Elapsed: 0.018 sec. Processed 10.00 thousand rows, 4
Peak memory usage: 222.22 KiB.
```



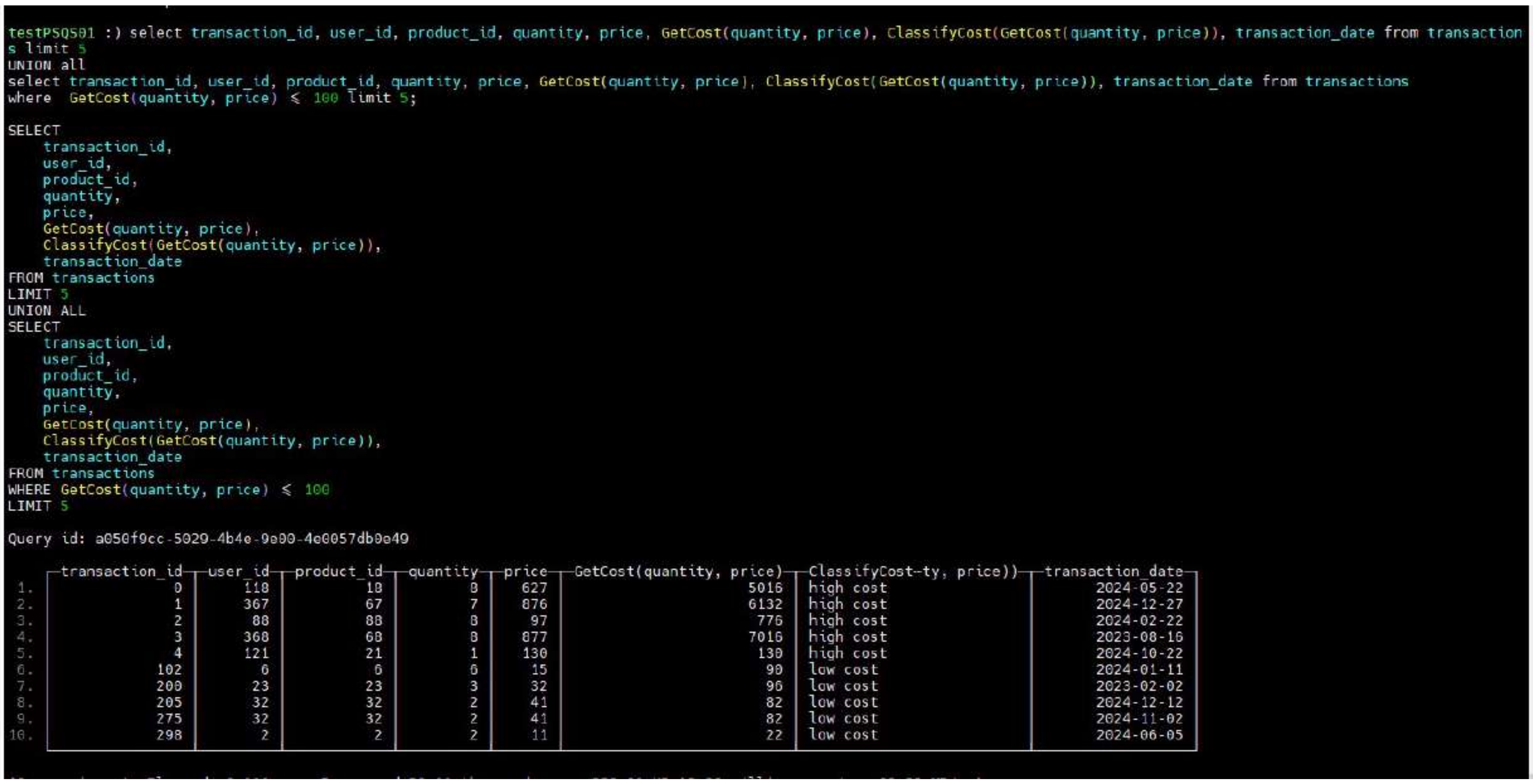
```
testPSQS01 :) select cast(transaction_date as String), toYYYYMM(transaction_date), round(price), cast(transaction_id as String) from transactions limit 5;

SELECT
    CAST(transaction_date, 'String'),
    toYYYYMM(transaction_date),
    round(price),
    CAST(transaction_id, 'String')
FROM transactions
LIMIT 5

Query id: 69da8438-24c1-47fc-a917-3f8ebd045703

   ┌─CAST(transac⋯, 'String')─┬─toYYYYMM(transaction_date)─┬─round(price)─┬─CAST(transac⋯, 'String')─┐
1. │ 2025-09-29               │                     202509 │          609 │ 0                        │
2. │ 2025-08-30               │                     202508 │           24 │ 1                        │
3. │ 2025-09-18               │                     202509 │          445 │ 2                        │
4. │ 2025-08-19               │                     202508 │          905 │ 3                        │
5. │ 2025-05-18               │                     202505 │          923 │ 4                        │
   └──────────────────────────┴────────────────────────────┴──────────────┴──────────────────────────┘

5 rows in set. Elapsed: 0.019 sec. Processed 10.00 thousand rows, 100.00 KB (533.11 thousand rows/s., 5.33 MB/s.)
Peak memory usage: 312.01 KiB.
```



```
testPSQS01 :) CREATE FUNCTION GetCost AS (quantity, price) -> quantity * price;

CREATE FUNCTION GetCost AS (quantity, price) -> (quantity * price)

Query id: a2421b5c-484e-495d-8cae-005abd64e1e0

Ok.

0 rows in set. Elapsed: 0.007 sec.

testPSQS01 :) select transaction_id, user_id, product_id, quantity, price, GetCost(quantity, price), transaction_date from transactions limit 5;

SELECT
    transaction_id,
    user_id,
    product_id,
    quantity,
    price,
    GetCost(quantity, price),
    transaction_date
FROM transactions
LIMIT 5

Query id: 6cce7166-e25c-4f98-a7f2-d1a751f76234

   ┌─transaction_id─┬─user_id─┬─product_id─┬─quantity─┬─price─┬─GetCost(quantity, price)─┬─transaction_date─┐
1. │              0 │     118 │         18 │        8 │   627 │                     5016 │       2024-05-22 │
2. │              1 │     367 │         67 │        7 │   876 │                     6132 │       2024-12-27 │
3. │              2 │      88 │         88 │        8 │    97 │                      776 │       2024-02-22 │
4. │              3 │     368 │         68 │        8 │   877 │                     7016 │       2023-08-16 │
5. │              4 │     121 │         21 │        1 │   130 │                      130 │       2024-10-22 │
   └────────────────┴─────────┴────────────┴──────────┴───────┴──────────────────────────┴──────────────────┘

5 rows in set. Elapsed: 0.004 sec. Processed 10.00 thousand rows, 190.00 KB (2.66 million rows/s., 50.55 MB/s.)
Peak memory usage: 296.28 KiB.

testPSQS01 :) 
```



```
testPSQS01 :) CREATE FUNCTION ClassifyCost AS (cost) →
    if(cost ⩽ 100, 'low cost', 'high cost');

CREATE FUNCTION ClassifyCost AS cost → if(cost ⩽ 100, 'low cost', 'high cost')

Query id: 1be36aad-8611-4522-ae8b-d67f9362ebcf

Ok.

0 rows in set. Elapsed: 0.007 sec.
```



```
testPSQS01 :) select transaction_id, user_id, product_id, quantity, price, GetCost(quantity, price), ClassifyCost(GetCost(quantity, price)), transaction_date from transaction
s limit 5
UNION all
select transaction_id, user_id, product_id, quantity, price, GetCost(quantity, price), ClassifyCost(GetCost(quantity, price)), transaction_date from transactions
where  GetCost(quantity, price) ⩽ 100 limit 5;

SELECT
    transaction_id,
    user_id,
    product_id,
    quantity,
    price,
    GetCost(quantity, price),
    ClassifyCost(GetCost(quantity, price)),
    transaction_date
FROM transactions
LIMIT 5
UNION ALL
SELECT
    transaction_id,
    user_id,
    product_id,
    quantity,
    price,
    GetCost(quantity, price),
    ClassifyCost(GetCost(quantity, price)),
    transaction_date
FROM transactions
WHERE GetCost(quantity, price) ⩽ 100
LIMIT 5

Query id: a050f9cc-5029-4b4e-9e00-4e0057db0e49

    ┌─transaction_id─┬─user_id─┬─product_id─┬─quantity─┬─price─┬─GetCost(quantity, price)─┬─ClassifyCost─ty, price))─┬─transaction_date─┐
 1. │              0 │     118 │         18 │        8 │   627 │                     5016 │ high cost                │       2024-05-22 │
 2. │              1 │     367 │         67 │        7 │   876 │                     6132 │ high cost                │       2024-12-27 │
 3. │              2 │      88 │         88 │        8 │    97 │                      776 │ high cost                │       2024-02-22 │
 4. │              3 │     368 │         68 │        8 │   877 │                     7016 │ high cost                │       2023-08-16 │
 5. │              4 │     121 │         21 │        1 │   130 │                      130 │ high cost                │       2024-10-22 │
 6. │            102 │       6 │          6 │        6 │    15 │                       90 │ low cost                 │       2024-01-11 │
 7. │            200 │      23 │         23 │        3 │    32 │                       96 │ low cost                 │       2023-02-02 │
 8. │            205 │      32 │         32 │        2 │    41 │                       82 │ low cost                 │       2024-12-12 │
 9. │            275 │      32 │         32 │        2 │    41 │                       82 │ low cost                 │       2024-11-02 │
10. │            298 │       2 │          2 │        2 │    11 │                       22 │ low cost                 │       2024-06-05 │
    └────────────────┴─────────┴────────────┴──────────┴───────┴──────────────────────────┴──────────────────────────┴──────────────────┘
```