Функции аггрегации и UDF

Вариант 1

1. Создали таблицу и вставили в нее следующие значения:

2. **insert into** transactions **VALUES** (1,472,153,18,163.21,2023-02-17)

```
,(2,587,41,4,256.39,2023-12-10)
,(3,199,426,14,122.45,2023-06-24)
,(4,926,287,2,376.81,2024-11-09)
,(5,31,233,20,493.99,2025-01-20)
,(6,894,199,5,88.52,2023-08-31)
,(7,712,154,11,29.73,2023-04-14)
,(8,422,385,7,111.78,2025-07-15)
,(9,609,82,15,200.30,2024-05-03)
,(10,178,219,1,12.64,2023-03-22);
```

3. Затем вставили повторно теже данные, чтобы можно было легко проверять корректность аггрегатов

4. Итоговая таблица получилась:

```
SELECT *
FROM transactions

Query id: cce396af-d193-45f5-b511-e624277d410c

    ┌─transaction_id─┬─user_id─┬─product_id─┬─quantity─┬──price─┬─transaction_date─┐
 1. │              1 │     472 │        153 │       18 │ 163.21 │       1975-06-28 │
 2. │              2 │     587 │         41 │        4 │ 256.39 │       1975-06-25 │
 3. │              3 │     199 │        426 │       14 │ 122.45 │       1975-06-17 │
 4. │              4 │     926 │        287 │        2 │ 376.81 │       1975-06-28 │
 5. │              5 │      31 │        233 │       20 │ 493.99 │       1975-06-28 │
 6. │              6 │     894 │        199 │        5 │  88.52 │       1975-06-08 │
 7. │              7 │     712 │        154 │       11 │  29.73 │       1975-06-29 │
 8. │              8 │     422 │        385 │        7 │ 111.78 │       1975-06-27 │
 9. │              9 │     609 │         82 │       15 │  200.3 │       1975-07-10 │
10. │             10 │     178 │        219 │        1 │  12.64 │       1975-06-22 │
11. │              1 │     472 │        153 │       18 │ 163.21 │       1975-06-28 │
12. │              2 │     587 │         41 │        4 │ 256.39 │       1975-06-25 │
```

```
13. │                3 │    199 │         426 │         14 │ 122.45 │           1975-06-17 │
14. │                4 │    926 │         287 │          2 │ 376.81 │           1975-06-28 │
15. │                5 │     31 │         233 │         20 │ 493.99 │           1975-06-28 │
16. │                6 │    894 │         199 │          5 │  88.52 │           1975-06-08 │
17. │                7 │    712 │         154 │         11 │  29.73 │           1975-06-29 │
18. │                8 │    422 │         385 │          7 │ 111.78 │           1975-06-27 │
19. │                9 │    609 │          82 │         15 │  200.3 │           1975-07-10 │
20. │               10 │    178 │         219 │          1 │  12.64 │           1975-06-22 │
    └──────────────────┴────────┴─────────────┴────────────┴────────┴──────────────────────┘

20 rows in set. Elapsed: 0.003 sec.

5. Рассчитали
SELECT
    sum(price * quantity) AS total_income,
    avg(price * quantity) AS avg_income,
    sum(quantity) AS total_quantity,
    uniq(user_id) AS uniq_user_amount
FROM transactions

Query id: 8339bbfa-b400-41ab-8c09-2d2ff3a8c45e

   ┌───────total_income─┬─────────avg_income─┬─total_quantity─┬─uniq_user_amount─┐
1. │ 41760.579904556274 │ 2088.0289952278135 │            194 │               10 │
   └────────────────────┴────────────────────┴────────────────┴──────────────────┘

1 row in set. Elapsed: 0.003 sec.

SELECT
    toString(transaction_date) AS String,
    toTypeName(toString(transaction_date)),
    toYear(transaction_date) AS Year,
    round(price) AS round_price,
    price,
    toString(transaction_id) AS string_transaction_id,
    toTypeName(toString(transaction_id))
FROM transactions

Query id: 3c170f81-f11f-460d-9b88-57fa0538ac6f

    ┌─String─────┬─toTypeName(t⋯tion_date))─┬─Year─┬─round_price─┬──price─┬─string_transaction_id─┬─toTypeName(t⋯
action_id))─┐
 1. │ 1975-06-28 │ String                   │ 1975 │         163 │ 163.21 │ 1                     │ String
│
 2. │ 1975-06-25 │ String                   │ 1975 │         256 │ 256.39 │ 2                     │ String
│
 3. │ 1975-06-17 │ String                   │ 1975 │         122 │ 122.45 │ 3                     │ String
│
 4. │ 1975-06-28 │ String                   │ 1975 │         377 │ 376.81 │ 4                     │ String
│
 5. │ 1975-06-28 │ String                   │ 1975 │         494 │ 493.99 │ 5                     │ String
│
 6. │ 1975-06-08 │ String                   │ 1975 │          89 │  88.52 │ 6                     │ String
│
 7. │ 1975-06-29 │ String                   │ 1975 │          30 │  29.73 │ 7                     │ String
│
 8. │ 1975-06-27 │ String                   │ 1975 │         112 │ 111.78 │ 8                     │ String
│
 9. │ 1975-07-10 │ String                   │ 1975 │         200 │  200.3 │ 9                     │ String
│
10. │ 1975-06-22 │ String                   │ 1975 │          13 │  12.64 │ 10                    │ String
│
11. │ 1975-06-28 │ String                   │ 1975 │         163 │ 163.21 │ 1                     │ String
│
12. │ 1975-06-25 │ String                   │ 1975 │         256 │ 256.39 │ 2                     │ String
│
13. │ 1975-06-17 │ String                   │ 1975 │         122 │ 122.45 │ 3                     │ String
│
14. │ 1975-06-28 │ String                   │ 1975 │         377 │ 376.81 │ 4                     │ String
│
15. │ 1975-06-28 │ String                   │ 1975 │         494 │ 493.99 │ 5                     │ String
│
16. │ 1975-06-08 │ String                   │ 1975 │          89 │  88.52 │ 6                     │ String
│
17. │ 1975-06-29 │ String                   │ 1975 │          30 │  29.73 │ 7                     │ String
│
18. │ 1975-06-27 │ String                   │ 1975 │         112 │ 111.78 │ 8                     │ String
│
19. │ 1975-07-10 │ String                   │ 1975 │         200 │  200.3 │ 9                     │ String
│
20. │ 1975-06-22 │ String                   │ 1975 │          13 │  12.64 │ 10                    │ String
│
```

```
└──────────┴─────────────────────┴─────┴──────────┴──────┴──────────────────┴────────────────────
┘

20 rows in set. Elapsed: 0.005 sec.
```

6. Создали SQL Defined UDF

```
CREATE FUNCTION
transaction_income AS (price, amount) -> price*amount;

CREATE FUNCTION transaction_income AS (price, amount) -> (price * amount)

Query id: 737f5c9e-8845-4fef-ad3f-90a547fb7b00

Ok.

0 rows in set. Elapsed: 0.029 sec.

clickhouse-node.ru-central1.internal :) select transaction_income(price, quantity), price,
quantity, transaction_id from transactions

SELECT
    transaction_income(price, quantity),
    price,
    quantity,
    transaction_id
FROM transactions

Query id: 54d9e4ae-1053-43eb-8637-90f5ec974842

    ┌─transaction_⋯, quantity)─┬──price─┬─quantity─┬─transaction_id─┐
 1. │       2937.7801208496094 │ 163.21 │       18 │              1 │
 2. │        1025.56005859375  │ 256.39 │        4 │              2 │
 3. │       1714.2999572753906 │ 122.45 │       14 │              3 │
 4. │        753.6199951171875 │ 376.81 │        2 │              4 │
 5. │         9879.7998046875  │ 493.99 │       20 │              5 │
 6. │       442.59998321533203 │  88.52 │        5 │              6 │
 7. │        327.0299949645996 │  29.73 │       11 │              7 │
 8. │        782.4599914550781 │ 111.78 │        7 │              8 │
 9. │        3004.500045776367 │  200.3 │       15 │              9 │
10. │       12.640000343322754 │  12.64 │        1 │             10 │
11. │       2937.7801208496094 │ 163.21 │       18 │              1 │
12. │        1025.56005859375  │ 256.39 │        4 │              2 │
13. │       1714.2999572753906 │ 122.45 │       14 │              3 │
14. │        753.6199951171875 │ 376.81 │        2 │              4 │
15. │         9879.7998046875  │ 493.99 │       20 │              5 │
16. │       442.59998321533203 │  88.52 │        5 │              6 │
17. │        327.0299949645996 │  29.73 │       11 │              7 │
18. │        782.4599914550781 │ 111.78 │        7 │              8 │
19. │        3004.500045776367 │  200.3 │       15 │              9 │
20. │       12.640000343322754 │  12.64 │        1 │             10 │
    └──────────────────────────┴────────┴──────────┴────────────────┘
```

7. Функция категоризации с возможностью задания порога

```
CREATE FUNCTION
Transaction_category AS (price, amount, level) -> CAST(If((price*amount)>level, 'High', 'Low'),
'Enum(\'High\',\'Low\')')

CREATE FUNCTION Transaction_category AS (price, amount, level) -> CAST(If((price * amount) > level,
'High', 'Low'), 'Enum(\'High\',\'Low\')')

Query id: ef64ed03-c3bf-4d59-92ed-b9af9875048d

Ok.

0 rows in set. Elapsed: 0.022 sec.
```

```
clickhouse-node.ru-central1.internal :) select transaction_income(price,
quantity),Transaction_category(price, quantity, 1000), price, quantity, transaction_id from
transactions

SELECT
    transaction_income(price, quantity),
    Transaction_category(price, quantity, 1000),
    price,
    quantity,
    transaction_id
FROM transactions

Query id: 6e799afa-9a90-4ef0-b9d2-2d34f4b7b5b4

    ┌─transaction_⋯, quantity)─┬─Transaction_⋯tity, 1000)─┬──price─┬─quantity─┬─transaction_id─┐
 1. │       2937.7801208496094 │ High                     │ 163.21 │       18 │              1 │
 2. │       2937.7801208496094 │ High                     │ 163.21 │       18 │              1 │
 3. │         1025.56005859375 │ High                     │ 256.39 │        4 │              2 │
 4. │         1025.56005859375 │ High                     │ 256.39 │        4 │              2 │
 5. │       1714.2999572753906 │ High                     │ 122.45 │       14 │              3 │
 6. │       1714.2999572753906 │ High                     │ 122.45 │       14 │              3 │
 7. │        753.6199951171875 │ Low                      │ 376.81 │        2 │              4 │
 8. │        753.6199951171875 │ Low                      │ 376.81 │        2 │              4 │
 9. │         9879.7998046875 │ High                     │ 493.99 │       20 │              5 │
10. │         9879.7998046875 │ High                     │ 493.99 │       20 │              5 │
11. │       442.59998321533203 │ Low                      │  88.52 │        5 │              6 │
12. │       442.59998321533203 │ Low                      │  88.52 │        5 │              6 │
13. │        327.0299949645996 │ Low                      │  29.73 │       11 │              7 │
14. │        327.0299949645996 │ Low                      │  29.73 │       11 │              7 │
15. │        782.4599914550781 │ Low                      │ 111.78 │        7 │              8 │
16. │        782.4599914550781 │ Low                      │ 111.78 │        7 │              8 │
17. │        3004.500045776367 │ High                     │  200.3 │       15 │              9 │
18. │        3004.500045776367 │ High                     │  200.3 │       15 │              9 │
19. │       12.640000343322754 │ Low                      │  12.64 │        1 │             10 │
20. │       12.640000343322754 │ Low                      │  12.64 │        1 │             10 │
    └──────────────────────────┴──────────────────────────┴────────┴──────────┴────────────────┘

20 rows in set. Elapsed: 0.004 sec.
```