Функции для работы с типами данных, агрегатные функции и UDF

- Создал таблицу и наполнил данными.

```
CREATE TABLE transactions
(
    `transaction_id` UInt32,
    `user_id` UInt32,
    `product_id` UInt32,
    `quantity` UInt8,
    `price` Float32,
    `transaction_date` Date
)
ENGINE = MergeTree
ORDER BY transaction_id

INSERT INTO transactions SELECT
    randUniform(1, 1000000.),
    randUniform(1, 1000000.),
    randUniform(1, 1000.),
    randUniform(1, 100.),
    randUniform(1, 10.),
    now()
FROM numbers(1000)
```

- Суммарный доход от всех транзакций

```
SELECT sum(price)
FROM transactions
```

```
1.  sum(price)
    5564.015149593353
```

- Средний доход от сделки

```
SELECT avg(price)
FROM transactions
```

```
1.  avg(price)
    5.564015149593353
```

- Количество проданной продукции

```
SELECT sum(quantity)
FROM transactions
```

```
1.  sum(quantity)
    51632
```

- Количество уникальных пользователей

```
SELECT uniq(user_id)
FROM transactions
```

```
1.  uniq(user_id)
    1000
```

- Преобразования колонок всё в одном

```
SELECT
    formatDateTimeInJodaSyntax(transaction_date, 'yyyy-MM-dd') AS date,
    toYear(transaction_date) AS year,
    toMonth(transaction_date) AS month,
    floor(price) AS price,
    toString(transaction_id) AS id
FROM transactions
LIMIT 10
```

	date	year	month	price	id
1.	2024-07-06	2024	7	2	759
2.	2024-07-06	2024	7	7	2865
3.	2024-07-06	2024	7	2	3497
4.	2024-07-06	2024	7	6	3676
5.	2024-07-06	2024	7	2	4581
6.	2024-07-06	2024	7	2	6907
7.	2024-07-06	2024	7	8	7206
8.	2024-07-06	2024	7	7	8367
9.	2024-07-06	2024	7	1	11346
10.	2024-07-06	2024	7	2	11420

- Создал 2 функции

  `CREATE FUNCTION calc_full_price AS (price, quantity) -> (price * quantity)`

  `CREATE FUNCTION is_pricy AS full_price -> if(full_price > 50, 'Pricy', 'Cheap')`

- И использовал их

```
SELECT
    calc_full_price(price, quantity) AS full_price,
    is_pricy(full_price) AS is_pricy
FROM transactions
LIMIT 10
```

	full_price	is_pricy
1.	104.95167446136475	Pricy
2.	95.4878921508789	Pricy
3.	63.85387849807739	Pricy
4.	19.178239345550537	Cheap
5.	17.414888381958008	Cheap
6.	228.05708742141724	Pricy
7.	94.83372020721436	Pricy
8.	345.1533203125	Pricy
9.	28.042781829833984	Cheap
10.	208.676203250885	Pricy