**Вариант №1**

<https://www.youtube.com/shorts/9xzX4ACDsEI>

* Создание таблицы transaction и ее наполнение

**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**

**number as transaction\_id,**

**(intdiv(rand(),**

**429496) % 10000) + 1 as user\_id,**

**(intdiv(rand(),**

**429496) % 1000) + 1 as product\_id,**

**(intdiv(rand(),**

**429496) % 100) + 1 as quantity,**

**round(((rand() % 49500) / 100) + 5,**

**2) as price,**

**todate('2023-01-01') + (intdiv(rand(),**

**429496) % 365) as transaction\_date**

**from**

**numbers(100000)**

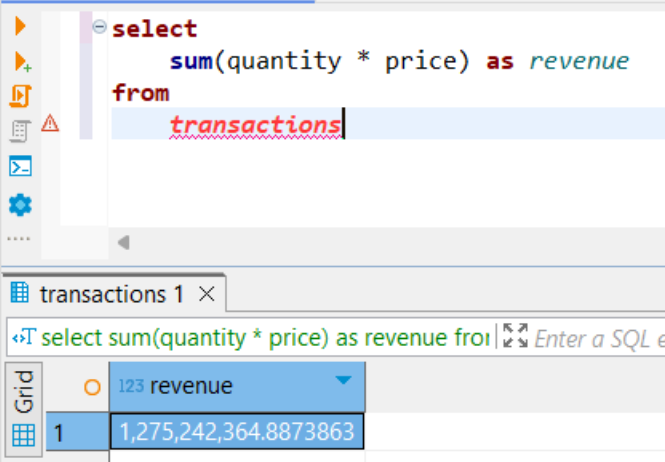
* Рассчитайте общий доход от всех операций

**select**

**sum(quantity \* price) as revenue**

**from**

**transactions**

******

* Найдите средний доход с одной сделки

**select round(avg(revenue), 2) as average\_revenue**

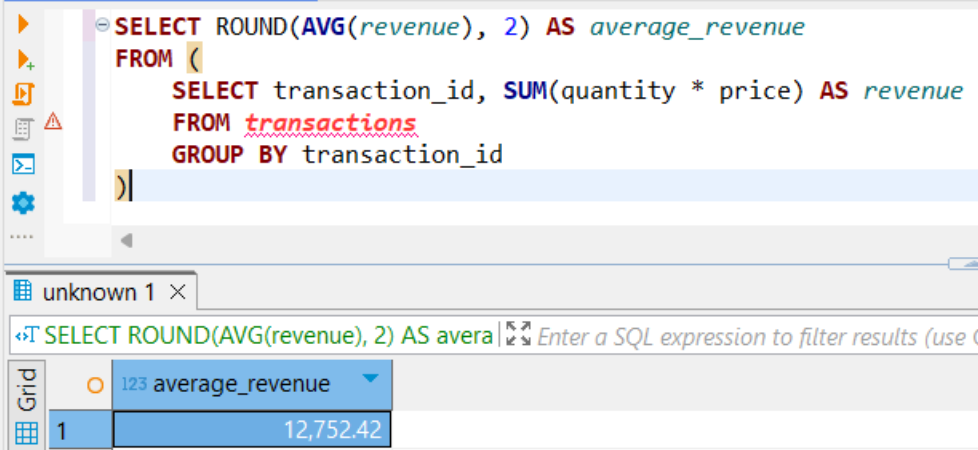
**from (**

**select transaction\_id, sum(quantity \* price) as revenue**

**from transactions**

**group by transaction\_id**

**)**



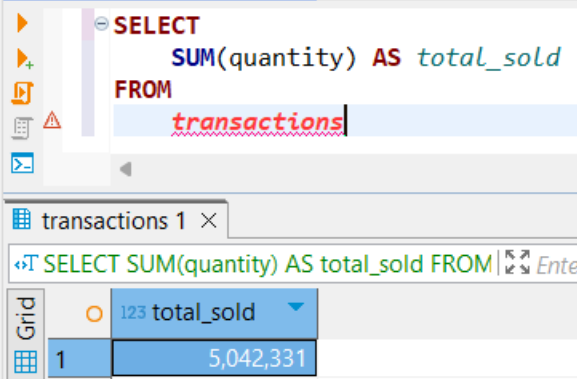
* Определите общее количество проданной продукции

**select**

**sum**(quantity) **as** total\_sold

**from**

**transactions**

******

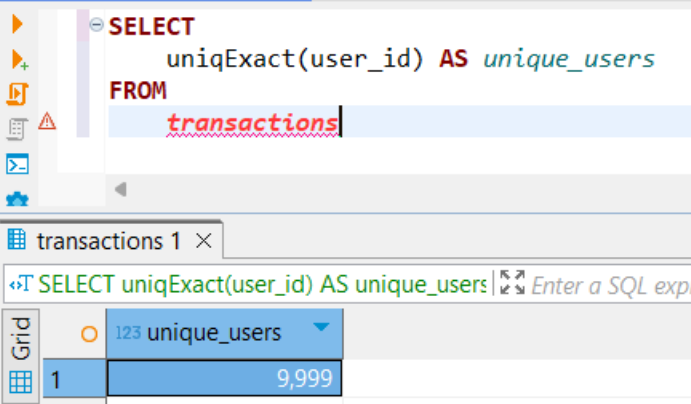
* Подсчитайте количество уникальных пользователей, совершивших покупку

**select**

uniqexact(user\_id) **as** unique\_users

**from**

**transactions**



* Преобразуйте `transaction\_date` в **строку** формата `YYYY-MM-DD`

**select**

transaction\_id,

user\_id,

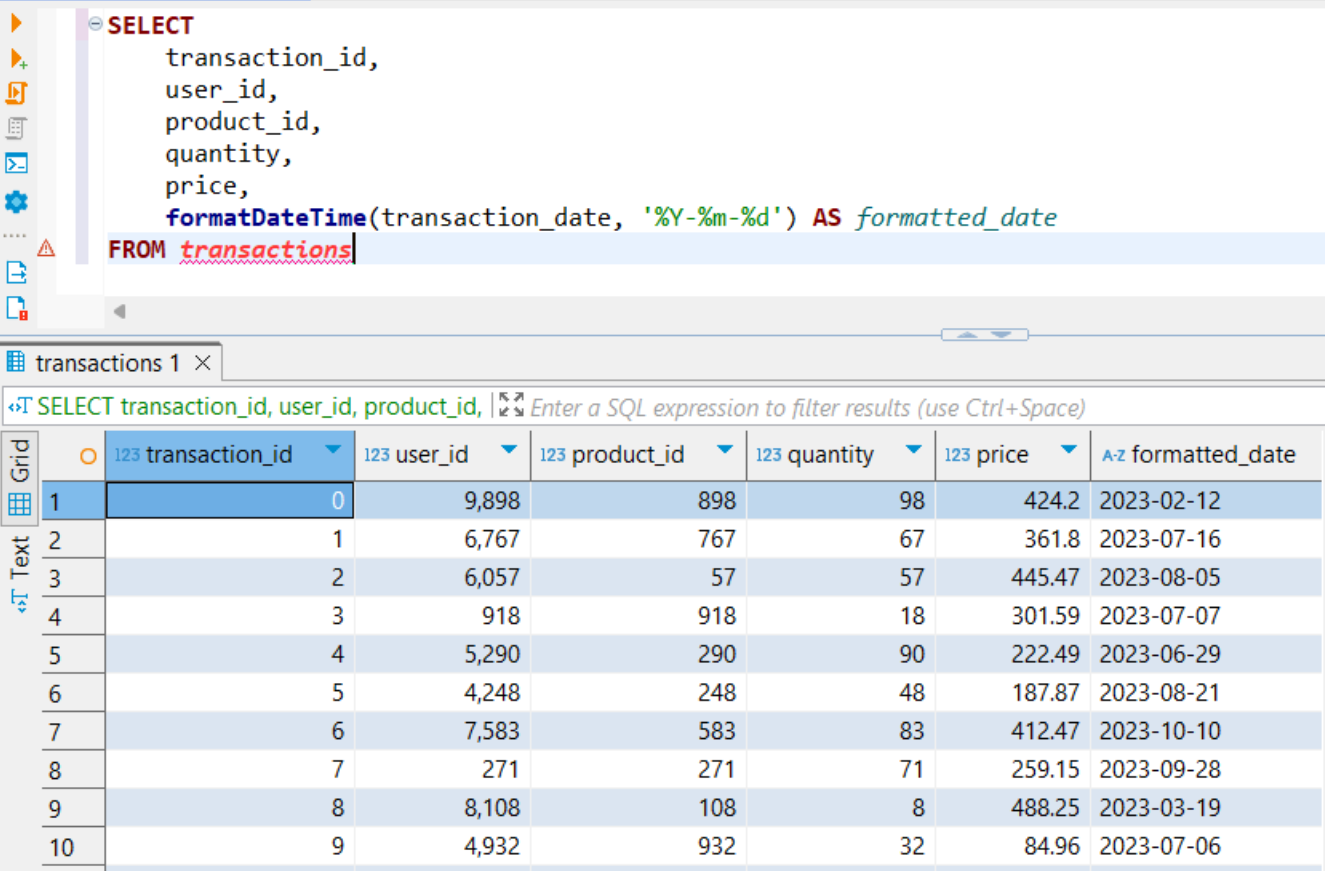
product\_id,

quantity,

price,

**formatdatetime**(transaction\_date, '%y-%m-%d') **as** formatted\_date

**from** **transactions**



* Извлеките год и месяц из `transaction\_date`

**select**

**transaction\_id,**

**user\_id,**

**product\_id,**

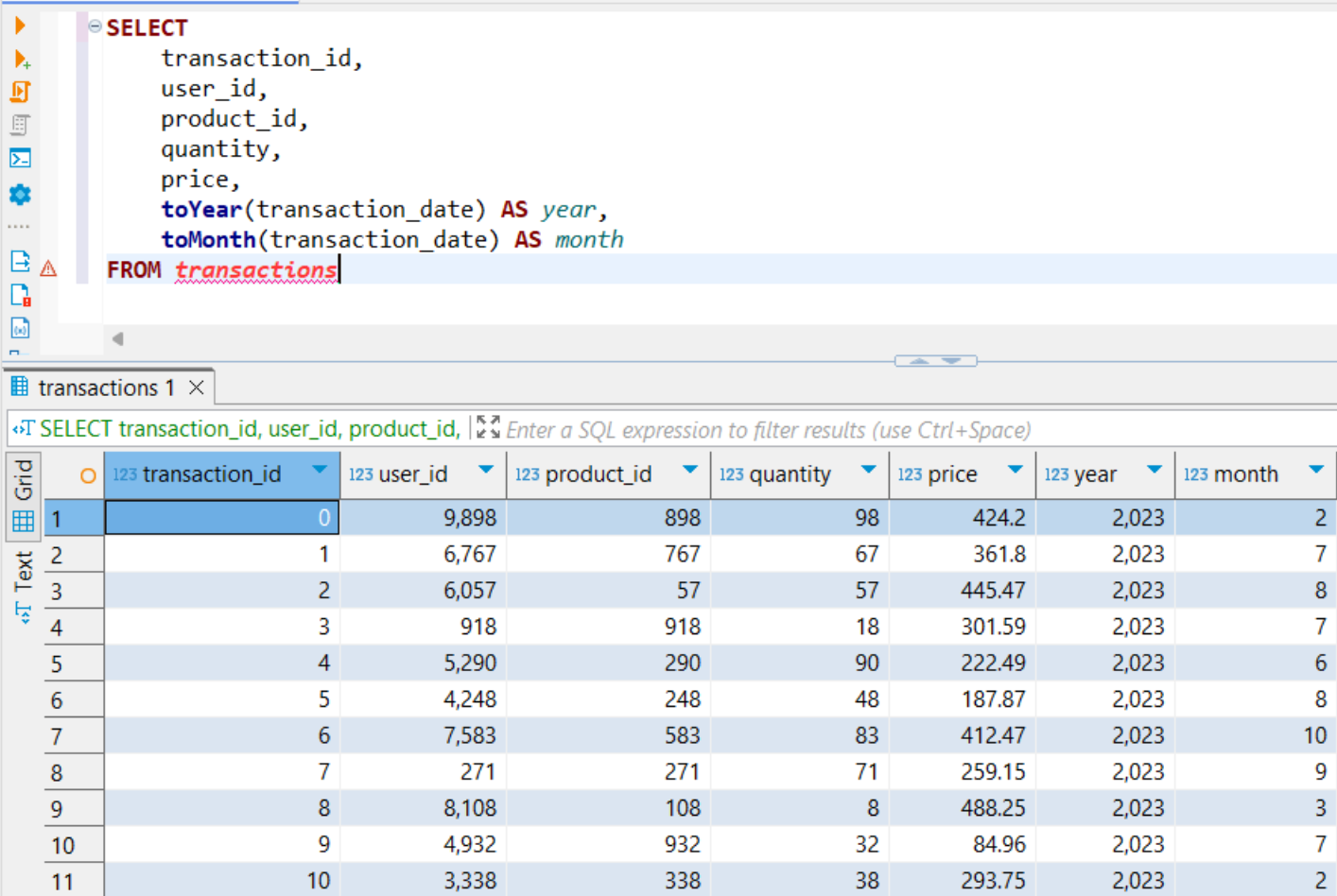
**quantity,**

**price,**

**toyear(transaction\_date) as year,**

**tomonth(transaction\_date) as month**

**from transactions**



* Округлите `price` до ближайшего целого числа

**select**

**transaction\_id,**

**user\_id,**

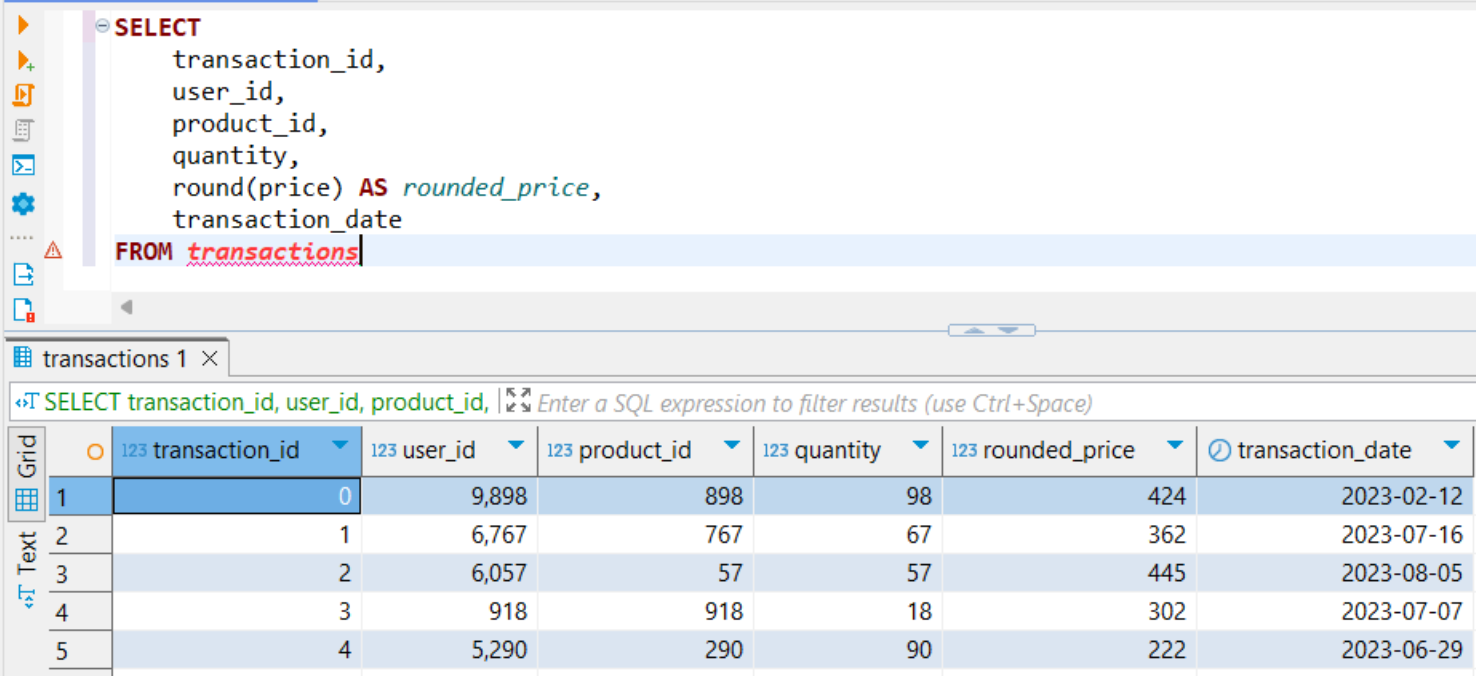
**product\_id,**

**quantity,**

**round(price) as rounded\_price,**

**transaction\_date**

**from transactions**



* Преобразуйте `transaction\_id` в строку

**select**

**tostring(transaction\_id) as transaction\_id\_str,**

**user\_id,**

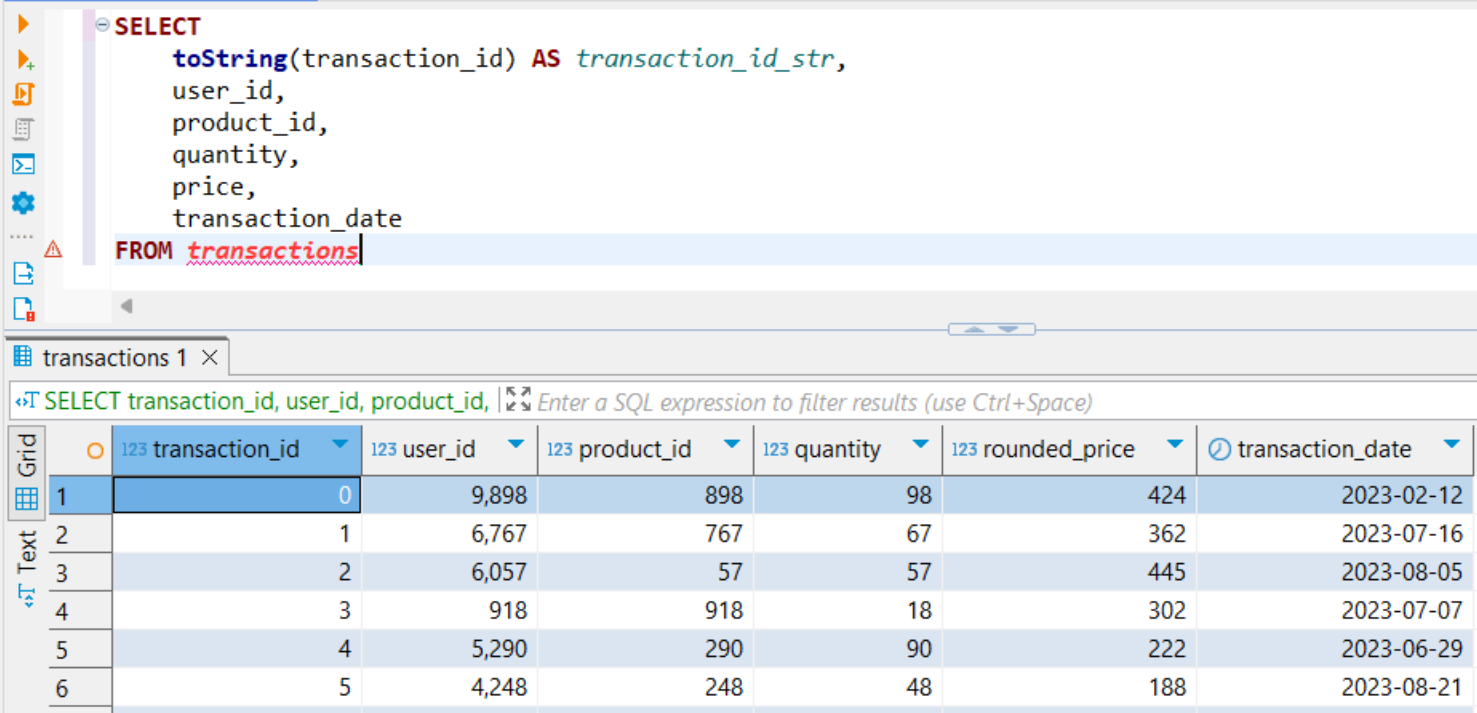
**product\_id,**

**quantity,**

**price,**

**transaction\_date**

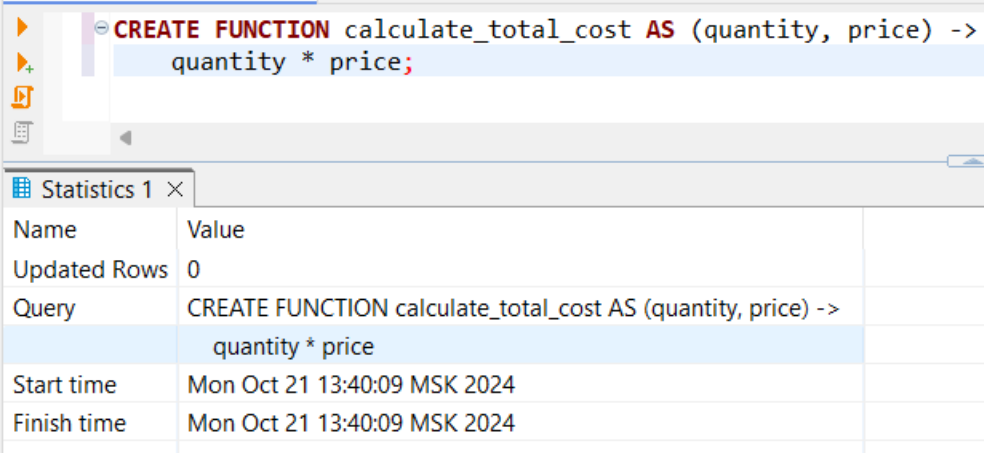
**from transactions**



* Создайте простую UDF для расчета общей стоимости транзакции

**create function calculate\_total\_cost as (quantity, price) ->**

**quantity \* price;**

****

* Используйте созданную UDF для расчета общей цены для каждой транзакции

**select**

**transaction\_id,**

**user\_id,**

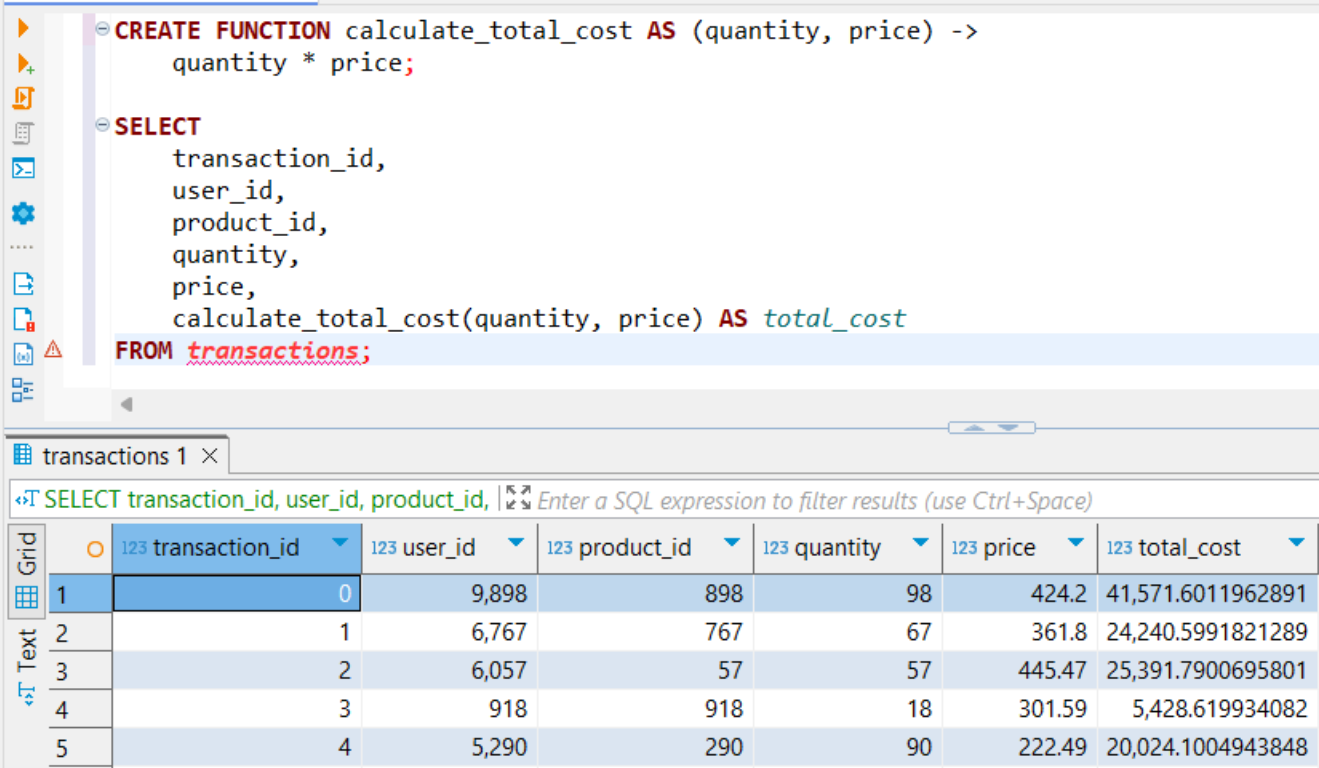
**product\_id,**

**quantity,**

**price,**

**calculate\_total\_cost(quantity, price) as total\_cost**

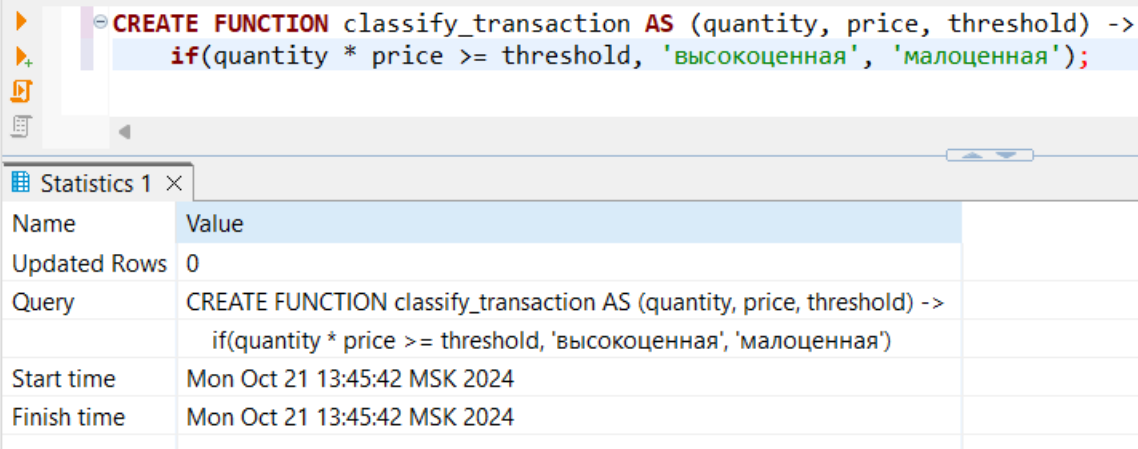
**from transactions;**



* Создайте UDF для классификации транзакций на «высокоценные» и «малоценные» на основе порогового значения (например, 100)

**create function classify\_transaction as (quantity, price, threshold) ->**

**if(quantity \* price >= threshold, 'высокоценная', 'малоценная');**



* Примените UDF для категоризации каждой транзакции

**select**

**transaction\_id,**

**user\_id,**

**product\_id,**

**quantity,**

**price,**

**calculate\_total\_cost(quantity, price) as total\_cost,**

**classify\_transaction(quantity, price, 100) as transaction\_class**

**from transactions**

**order by transaction\_id;**

