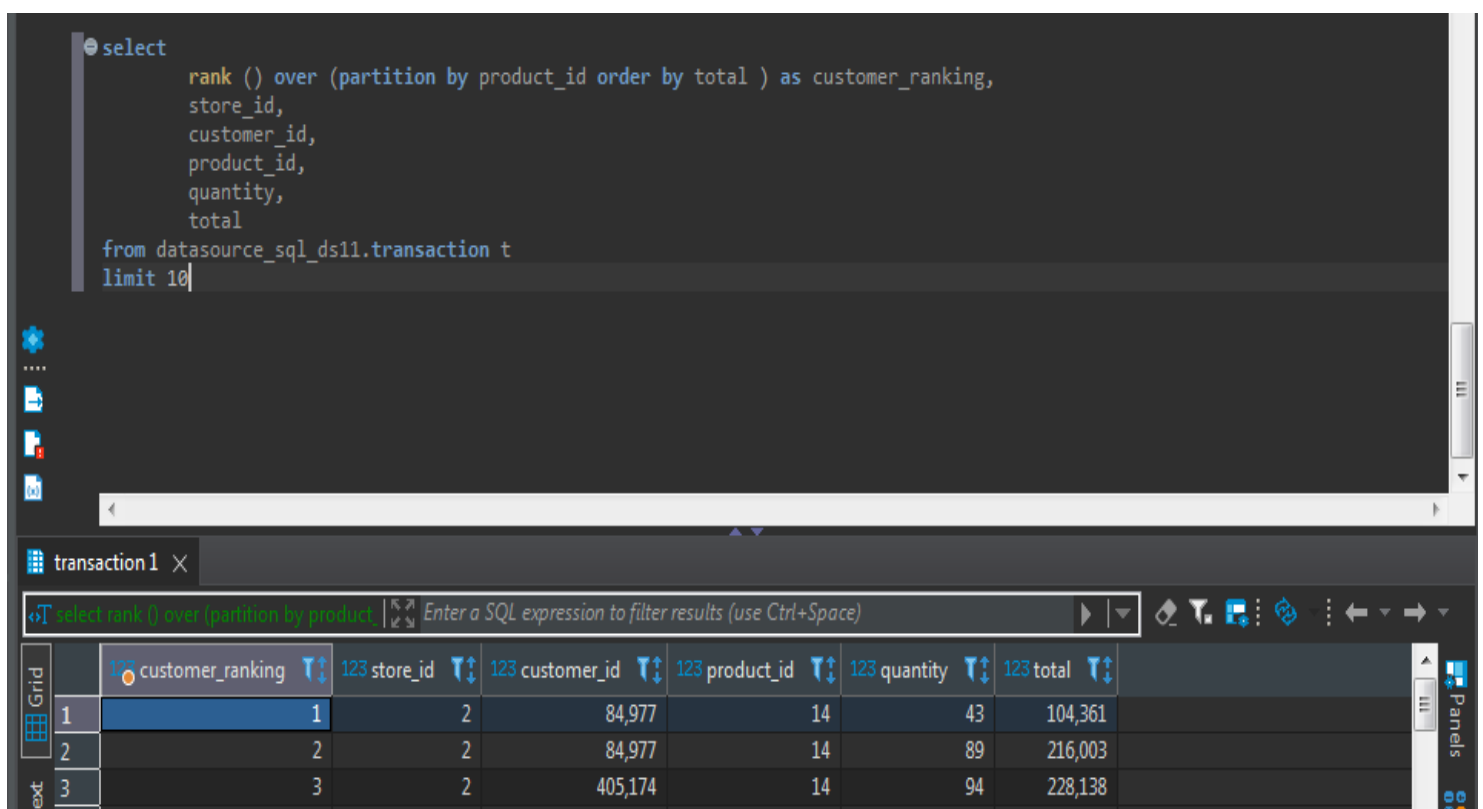


1. Apa Itu Windows Function?

Sebuah fungsi dimana kita dapat melakukan perhitungan dalam beberapa baris yang berbeda dan hasilnya akan ditampilkan dalam suatu baris yang baru.

Windows Function mirip dengan Agregate Function, tetapi ada satu perbedaan penting. Saat kita menggunakan Agregate Function dengan klausa GROUP BY, kita “kehilangan” baris individual. Kami tidak dapat mencampur atribut dari baris individual dengan hasil Agregate Function; fungsi dilakukan pada baris sebagai seluruh grup. Ini tidak terjadi ketika kita menggunakan fungsi jendela SQL: kita dapat menghasilkan kumpulan hasil dengan beberapa atribut dari baris individu bersama-sama dengan hasil dari Windows Function. Ini bagus untuk diingat oleh pengembang SQL baru. Jadi mari kita periksa contoh Windows Function jendela SQL sederhana dalam tindakan.

contoh windows function



The screenshot shows a SQL IDE interface. The top panel displays a SQL query using the `RANK()` window function. The query selects `rank()` over a partition of `product_id` ordered by `total`, along with `store_id`, `customer_id`, `product_id`, `quantity`, and `total`. It filters results from `datasource_sql_ds11.transaction t` and limits the output to 10 rows.

```
select
  rank () over (partition by product_id order by total ) as customer_ranking,
  store_id,
  customer_id,
  product_id,
  quantity,
  total
from datasource_sql_ds11.transaction t
limit 10
```

The bottom panel shows the results of the query in a grid view. The grid has columns for `customer_ranking`, `store_id`, `customer_id`, `product_id`, `quantity`, and `total`. The first three rows are visible, showing the ranking of products within each store.

	customer_ranking	store_id	customer_id	product_id	quantity	total
1	1	2	84,977	14	43	104,361
2	2	2	84,977	14	89	216,003
3	3	2	405,174	14	94	228,138

2. Menggunakan Where dalam fungsi Select

Menggunakan Where dalam fungsi select dapat dilakukan dengan melakukan Sub Query

Contoh :

```
with
  filtered_transactions as (
    select *
    from datasource_sql_ds11.transaction
    where store_id = 2
  ),
  pool as (
    select id from datasource_sql_ds11.customer
    where email <> 'Gmail' and city = 'Jakarta'
  )

select
  c.id,
  sum(t.quantity)
from datasource_sql_ds11.customer c
left join filtered_transactions t on c.id = t.customer_id
left join (
  select id, type from datasource_sql_ds11.store where id = 2
) s on t.store_id = s.id
left join datasource_sql_ds11.product p on t.product_id = p.id
where c.id in (select id from pool)
group by 1
order by 2 desc
limit 10
```