

Week 2 Homework 3

Hermulia Hadie Putra

Window function dan contohnya

Perintah-perintah 'Window function' digunakan untuk melakukan perhitungan pada baris-baris tabel yang terkait dengan baris yang dijalankan pada query. Tipe perhitungan yang dilakukan pada perintah ini mirip dengan yang bisa dilakukan oleh perintah umum pada 'Aggregate Function'. Bedanya, perintah pada 'Window function' dapat melakukan agregasi tanpa harus mengurangi jumlah baris (tidak menyebabkan baris-baris dikelompokkan menjadi satu output karena masih mempertahankan identitasnya).

Ada 2 jenis dari window function, yaitu:

1. Aggregate window function

Sintaks yang digunakan sama seperti pada Aggregate Function biasa, namun diikuti oleh pernyataan klausa OVER. Contoh yang paling umum adalah penggunaan OVER dan PARTITION BY. OVER digunakan untuk mendapatkan nilai agregat (seperti SUM, AVG, COUNT, MIN, MAX) tanpa menggunakan GROUP BY. Dengan OVER kita tetap mendapatkan seluruh baris secara detail beserta nilai agregatnya.

Sebagai contoh pada tabel Orders berikut:

REGION	PRODUCT	QTY
east	desktop	332
east	notebook	350
east	software	125
center	tablet	325
center	smartphone	550
center	desktop	186
center	notebook	220
center	software	54
west	tablet	432
west	smartphone	621
west	desktop	125
west	notebook	188
west	software	75
west	camera	65

Dengan melakukan query:

```
SELECT *  
  , SUM(Qty) OVER() Sum_qty  
FROM Orders;
```

SUM(Qty) OVER() jika diterjemahkan kurang lebih berarti:

- SUM(Qty) : hitung jumlah Qty
- OVER : untuk semua baris
- () : secara keseluruhan

Akan menghasilkan tabel sebagai berikut:

REGION	PRODUCT	QTY	SUM_QTY
east	desktop	332	3648
east	notebook	350	3648
east	software	125	3648
west	tablet	432	3648
west	smartphone	621	3648
west	desktop	125	3648
west	notebook	188	3648
west	software	75	3648
west	camera	65	3648
center	tablet	325	3648
center	smartphone	550	3648
center	desktop	186	3648
center	notebook	220	3648
center	software	54	3648

Selain itu, dengan menggunakan PARTITION BY, kita juga dapat memisahkan hasil perhitungan berdasarkan kriteria pada kolom product sesuai keinginan. Dengan melakukan query:

```
SELECT *
, SUM(Qty) OVER(PARTITION BY Product) Sum_qty
FROM orders;
```

SUM(Qty) OVER(PARTITION BY Product) berarti
SUM(Qty) : hitung jumlah Qty
OVER : untuk semua baris
(PARTITION BY Product) : yang mempunyai produk yang sama

Akan menghasilkan tabel sebagai berikut:

REGION	PRODUCT	QTY	SUM_QTY
east	software	125	254
center	software	54	254
west	software	75	254
east	desktop	332	643
center	desktop	186	643
west	desktop	125	643
center	tablet	325	757
west	tablet	432	757
west	camera	65	65
east	notebook	350	758
center	notebook	220	758
west	notebook	188	758
center	smartphone	550	1171
west	smartphone	621	1171

Partisi yang digunakan adalah Product. Artinya, setiap produk akan diperlakukan secara terpisah dengan produk lainnya. Sehingga akan didapatkan total quantity untuk produk software, kemudian total untuk desktop dan seterusnya.

2. Built-in Window Function

Perintah-perintah yang digunakan pada built-in window function seperti: CUME_DIST(), DENSE_RANK(), FIRST_VALUE(), LAG(), LAST_VALUE(), LEAD(), NTH_VALUE(), NTILE(), PERCENT_RANK(), RANK(), ROW_NUMBER()

Contoh yang akan ditampilkan berikut adalah penggunaan RANK().

Pada tabel data_siswa berikut, ingin ditampilkan siswa yang memiliki nilai tertinggi dari setiap kelas.

ID	Name	Score	Kelas
1	Udin	100	A
2	Putu Anom	80	B
3	Gede Gede	90	A
4	Hola	70	B

Dengan menggunakan fungsi RANK(), kita bisa memberikan rank pada semua siswa. Query yang dilakukan adalah sebagai berikut:

```
SELECT name, score, kelas
FROM (SELECT name, score, kelas, RANK() OVER (PARTITION BY kelas ORDER BY score
DESC) as rank)
WHERE rank=1
```

Tabel yang dihasilkan adalah sebagai berikut:

ID	Name	Score	Kelas	Ranking
1	Udin	100	A	1
2	Putu Anom	80	B	1
3	Gede Gede	90	A	2
4	Hola	70	B	2

Menggunakan WHERE di dalam fungsi SELECT

WHERE Clause pada perintah SELECT digunakan untuk menyeleksi data atau record sesuai dengan kondisi yang diinginkan.

```
SELECT *  
FROM nama_table  
WHERE kondisi;
```

Operator-operator yang sering digunakan dalam WHERE Clause di SELECT Statement:

Operator	Keterangan
=	Sama dengan
!= atau <>	Tidak sama dengan
>	Lebih besar dari
>=	Lebih besar dari atau sama dengan
<	Kurang dari
<=	Kurang dari atau sama dengan
IS NULL	Apakah record sama dengan null
IS NOT NULL	Apakah record tidak sama dengan null
IN	Apakah record berada dalam
NOT IN	Apakah record berada diluar
LIKE	Pencarian record berdasarkan wildcard
NOT LIKE	Pencarian record tidak berdasarkan wildcard
BETWEEN	Pencarian record berada dalam range antara nilai minimum dan maksimum
AND	Logika AND (dan)
OR	Logika OR (atau)

Selain itu, fungsi WHERE juga dapat dikombinasikan dengan FILTER pada PostgreSQL dalam mencari Selective Aggregate. Hasil agregat yang dimunculkan hanyalah berupa baris-baris yang memenuhi batasan-batasan pada WHERE. Perintahnya adalah sebagai berikut:

```
SUM(<expression>) FILTER(WHERE <condition>)
```

Dengan pengecualian pada subquery dan Windows Function, <condition> dapat berisi ekspresi apa pun yang diizinkan dalam perintah WHERE yang regular.