

پرسش ها

اگر بخواهیم گروه بندی انجام دهیم که خروجی آن گزارشی از میزان فروش سالیانه، ماه و روزانه در سالها و ماه ها و روز های خاص باشد باید از کدام دستور استفاده کرد؟

UP بنابر ترتیبی که مشخص کردیم گزارش میدهد. ROLE باید از دستور CUBE استفاده کنیم زیرا این دستور برای هر ماه گزارش تولید میکند که مورد قبول ما است در صورتی که

فرق تابع RANK() DENSE با تابع RANK() چیست؟

فرق این دو تابع در آن است که `RANK()` رتبه بندی کرده به طوری که همه اعضای یک گروه رتبه یکسانی دارند اما گروه بعدی به ازای تعداد ردیف گروه اول اضافه میشود. در `DENSE_RANK()` مانند `RANK()` رتبه بندی شده اما شماره گروه بعد به ترتیب میاید و به تعداد اعضای گروه پیشین وابسته نیست. برای نمونه تفاوت بین دو تابع را مشاهده میکنید:

V	ROW_NUMBER	RANK	DENSE_RANK
a	1	1	1
a	2	1	1
a	3	1	1
b	4	4	2
c	5	5	3
c	6	5	3
d	7	7	4
e	8	8	5

تمرین ها

سوال اول

```

dvdrental/postgres@PostgreSQL 13 ▾
Query Editor Query History
1 SELECT *
2 FROM CROSSTAB('
3     WITH rents AS(
4         SELECT *
5         FROM rental
6         INNER JOIN inventory USING (inventory_id)
7     ), films AS (      -- Connect films, film_category & category
8         SELECT *
9         FROM film
10        INNER JOIN film_category USING (film_id)
11    )
12    SELECT
13        store_id AS store,
14        category_id AS category,
15        COUNT(*)
16    FROM rents
17    INNER JOIN films USING (film_id)
18    GROUP BY
19        store_id, category
20    ORDER BY
21        store_id, category

```

	store integer	Action integer	Animation integer	Children integer	Classics integer	Comedy integer	Documentary integer	Drama integer	Family integer	Foreign integer	Games integer	Horror integer	Music integer	New integer	Sci-Fi integer
1	1	596	568	453	447	502	450	576	557	524	455	386	436	502	521
2	2	516	598	492	492	439	600	484	539	509	514	460	394	438	580

عکس و کد ضمیمه شده است. ابتدا با استفاده از جداول rental، inventory، film_category و film اطلاعات مربوط به category_id، store_id و توسط count() aggregation function تعداد کتاب های مربوطه محاسبه شد، سپس با استفاده از crosstab() خروجی transpose شده و مقادیر اصلی category به جای category id جایگزین شده اند.

سوال دوم

عکس از خروجی این دستور ضمیمه شده است.

سوال سوم

عکس از خروجی این دستور ضمیمه شده است. تفاوت این دو به این صورت است که دستور CUBE 124 سطر خروجی داده اما دستور ROLE UP فقط 34 سطر خروجی داده است. این به آن دلیل است که دستور CUBE گزارش را بر اساس تمامی جایگشت های ممکن برمیگرداند. برای مثال سال 2007، ماه 3، گزارش تمام 31 روز اما در ROLE UP صرفاً به ترتیب مشخص شده گزارش میدهد. (تفاوت تعداد سطر های خروجی به دلیل حالت های جایگشت های سال، ماه و روز است)

Query Editor					
<pre> 1 -- 2 2 SELECT 3 extract(year from payment_date) AS year, -- Year 4 extract(month from payment_date) AS month, -- Month 5 extract(day from payment_date) AS day, -- Day 6 SUM(amount) 7 FROM 8 payment 9 GROUP BY 10 ROLLUP(year, month, day) 11 ORDER BY 12 year, month, day </pre>					
Data Output	year	month	day	sum	
	double precision	double precision	double precision	numeric	
1	2007	2	14	116.73	
2	2007	2	15	1188.92	
3	2007	2	16	1154.18	
4	2007	2	17	1188.17	
5	2007	2	18	1275.98	
6	2007	2	19	1290.90	
7	2007	2	20	1219.09	
8	2007	2	21	917.87	
9	2007	2	[null]	8351.84	
10	2007	3	1	2808.24	
11	2007	3	2	2550.05	
12	2007	3	16	299.28	
13	2007	3	17	2442.16	

Query Editor					
<pre> 1 -- 3 2 SELECT 3 extract(year from payment_date) AS year, -- Year 4 extract(month from payment_date) AS month, -- Month 5 extract(day from payment_date) AS day, -- Day 6 SUM(amount) 7 FROM 8 payment 9 GROUP BY 10 CUBE(year, month, day) 11 ORDER BY 12 year, month, day </pre>					
Data Output	year	month	day	sum	
	double precision	double precision	double precision	numeric	
1	2007	2	14	116.73	
2	2007	2	15	1188.92	
3	2007	2	16	1154.18	
4	2007	2	17	1188.17	
5	2007	2	18	1275.98	
6	2007	2	19	1290.90	
7	2007	2	20	1219.09	
8	2007	2	21	917.87	
9	2007	2	[null]	8351.84	
10	2007	3	1	2808.24	
11	2007	3	2	2550.05	
12	2007	3	16	299.28	
13	2007	3	17	2442.16	

سوال چهارم

با استفاده از تابع ROW_NUMBER() و سورت کردن بر اساس length، کد و عکس خروجی ضمیمه شده است.