

POSTGRESQL ROLL UP

Dashboard Properties SQL Statistics Dependencies Dependents [dvdrental/postgres@](#)

No limit

dvdrental/postgres@PostgreSQL 13 ▾

Query Editor Query History Explain Notifications

```
1  --create the sales table
2  DROP TABLE IF EXISTS sales;
3  CREATE TABLE sales (
4      brand VARCHAR NOT NULL,
5      segment VARCHAR NOT NULL,
6      quantity INT NOT NULL,
7      PRIMARY KEY (brand, segment)
8  );
9
10 INSERT INTO sales (brand, segment, quantity)
11 VALUES
12     ('ABC', 'Premium', 100),
13     ('ABC', 'Basic', 200),
14     ('XYZ', 'Premium', 100),
15     ('XYZ', 'Basic', 300);
16
```

Messages

INSERT 0 4

Query returned successfully in 608 msec.


dvdrental/postgres@PostgreSQL 13 ▾

Query Editor Query History Explain Notifications Messages

```
1  --This query uses the ROLLUP clause to find the number of
2  --products sold by brand (subtotal) and by all brands and segments (total).
3
4  SELECT
5      brand,
6      segment,
7      SUM (quantity)
8  FROM
9      sales
10 GROUP BY
11     ROLLUP (brand, segment)
12 ORDER BY
13     brand,
14     segment;
```

Data Output

	brand [PK] character varying	segment [PK] character varying	sum bigint
1	ABC	Basic	200
2	ABC	Premium	100
3	ABC	[null]	300
4	XYZ	Basic	300
5	XYZ	Premium	100
6	XYZ	[null]	400
7	[null]	[null]	700


dvdrental/postgres@PostgreSQL 13

[Query Editor](#)
[Query History](#)
[Explain](#)
[Notifications](#)
[Messages](#)


```

1  --This query change the order of brand and segment, the result will be different
2
3  SELECT
4      segment,
5      brand,
6      SUM (quantity)
7  FROM
8      sales
9  GROUP BY
10     ROLLUP (segment, brand)
11 ORDER BY
12     segment,
13     brand;
14

```

Data Output

	segment [PK] character varying	brand [PK] character varying	sum bigint
1	Basic	ABC	200
2	Basic	XYZ	300
3	Basic	[null]	500
4	Premium	ABC	100
5	Premium	XYZ	100
6	Premium	[null]	200
7	[null]	[null]	700


dvdrental/postgres@PostgreSQL 13

[Query Editor](#)
[Query History](#)
[Explain](#)
[Notifications](#)
[Messages](#)

```

1  --This query performs a partial roll-up
2
3  SELECT
4      segment,
5      brand,
6      SUM (quantity)
7  FROM
8      sales
9  GROUP BY
10     segment,
11     ROLLUP (brand)
12 ORDER BY
13     segment,
14     brand;
15

```

Data Output

	segment [PK] character varying	brand [PK] character varying	sum bigint
1	Basic	ABC	200
2	Basic	XYZ	300
3	Basic	[null]	500
4	Premium	ABC	100
5	Premium	XYZ	100
6	Premium	[null]	200

```

1  --This statement finds the number of rental per day, month, and year by using the ROLLUP
2  SELECT
3      EXTRACT (YEAR FROM rental_date) y,
4      EXTRACT (MONTH FROM rental_date) m,
5      EXTRACT (DAY FROM rental_date) d,
6      COUNT (rental_id)
7  FROM
8      rental
9  GROUP BY
10     ROLLUP (
11         EXTRACT (YEAR FROM rental_date),
12         EXTRACT (MONTH FROM rental_date),
13         EXTRACT (DAY FROM rental_date)
14     );
15

```

Data Output

	<div> <div>y</div> <div>double precision</div> <div></div> </div>	<div> <div>m</div> <div>double precision</div> <div></div> </div>	<div> <div>d</div> <div>double precision</div> <div></div> </div>	<div> <div>count</div> <div>bigint</div> <div></div> </div>	
1	2005	5	24	8	
2	2005	5	25	137	
3	2005	5	26	174	
4	2005	5	27	166	
5	2005	5	28	196	
6	2005	5	29	154	
7	2005	5	30	158	
8	2005	5	31	163	