**1.Fabric Sales Per Year Per Month**

SELECT T.fabric\_id, f.fabric\_name, T.total\_price, T.month, T.year

FROM

(

select pfm.fabric\_id,

DATE\_PART(month, transaction\_date) as month,

DATE\_PART(year, transaction\_date) as year,

SUM(t.price) as total\_price

from transactions t

join articles a on t.article\_id = a.article\_id

join product\_fabric\_map pfm on pfm.product\_code = a.product\_code

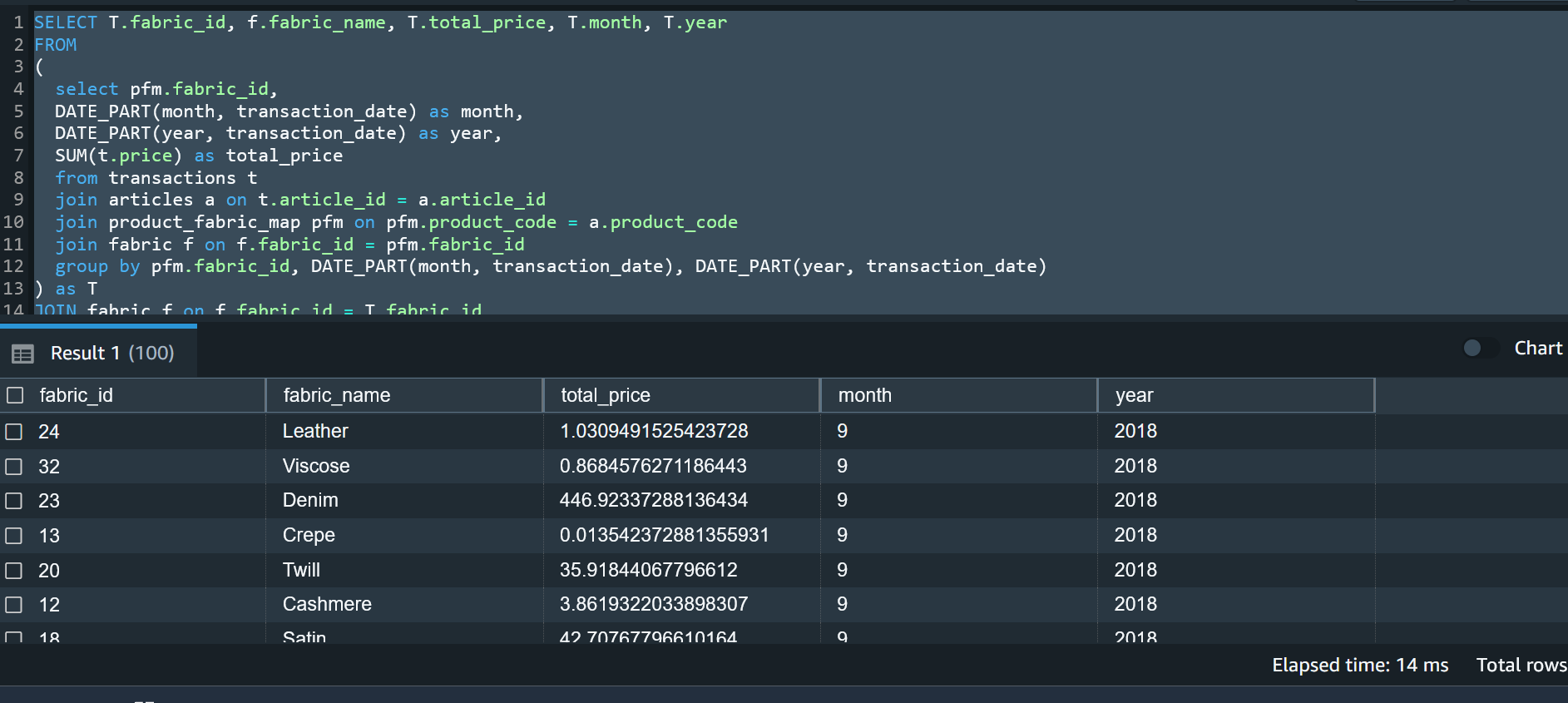
join fabric f on f.fabric\_id = pfm.fabric\_id

group by pfm.fabric\_id, DATE\_PART(month, transaction\_date), DATE\_PART(year, transaction\_date)

) as T

JOIN fabric f on f.fabric\_id = T.fabric\_id

order by T.year, T.month ASC;

****

**2.Overall most sold fabric with total sales**

SELECT T.fabric\_id, f.fabric\_name, T.total\_price

FROM

(

select pfm.fabric\_id, SUM(t.price) as total\_price

from transactions t

join articles a on t.article\_id = a.article\_id

join product\_fabric\_map pfm on pfm.product\_code = a.product\_code

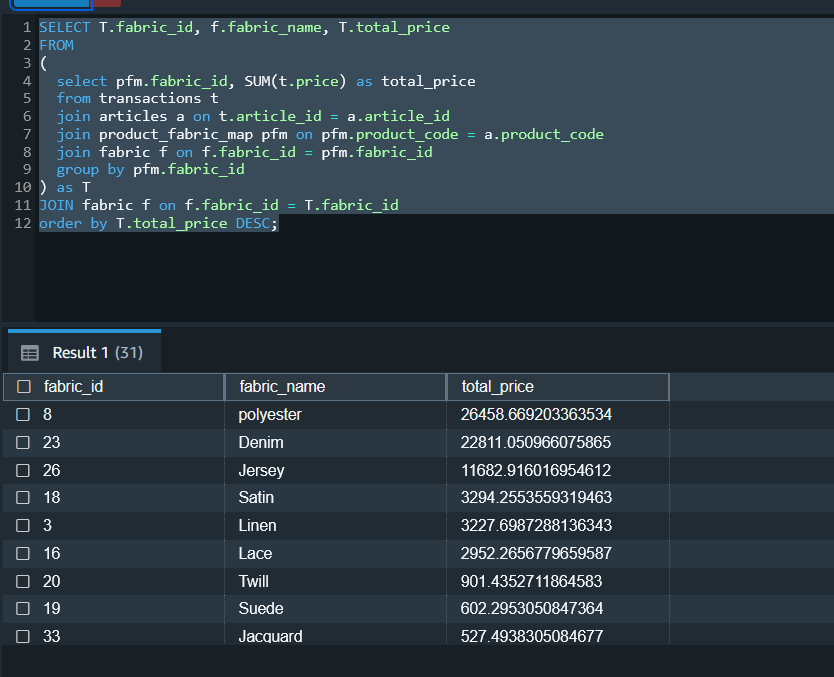
join fabric f on f.fabric\_id = pfm.fabric\_id

group by pfm.fabric\_id

) as T

JOIN fabric f on f.fabric\_id = T.fabric\_id

order by T.total\_price DESC;



**3.Most sold fabric with total sales Per Year Per Month**

SELECT S.fabric\_id, f.fabric\_name, S.month, S.year, S.total\_price

FROM (

select T.fabric\_id, T.month, T.year, T.total\_price,

MAX(T.total\_price) over (partition by T.month, T.year) as max\_sale

from

(

select pfm.fabric\_id,

DATE\_PART(month, transaction\_date) as month,

DATE\_PART(year, transaction\_date) as year,

SUM(t.price) as total\_price

from transactions t

join articles a on t.article\_id = a.article\_id

join product\_fabric\_map pfm on pfm.product\_code = a.product\_code

join fabric f on f.fabric\_id = pfm.fabric\_id

group by pfm.fabric\_id, DATE\_PART(month, transaction\_date), DATE\_PART(year, transaction\_date)

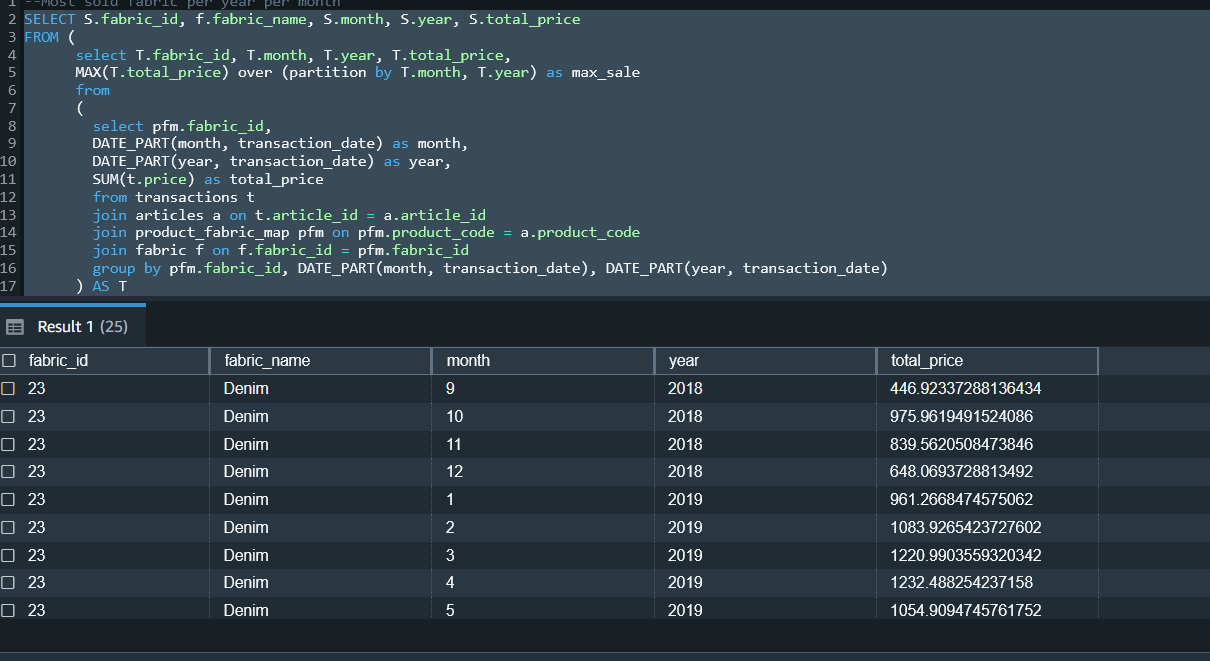
) AS T

) AS S

JOIN fabric f on f.fabric\_id = S.fabric\_id

WHERE S.total\_price = S.max\_sale

order by S.year, S.month;

****

**4. Creation of view**

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `getyourbestfit` AS

select distinct `pcm`.`article\_id` AS `article\_id`,`p`.`product\_code` AS `product\_code`,

`p`.`product\_type\_id` AS `product\_type\_id`,`p`.`prod\_name` AS `prod\_name`,

`p`.`detail\_desc` AS `detail\_desc`,`pt`.`type\_name`

AS `product\_type`,`c`.`bodytype` AS `body\_type`,`c`.`skin\_condition\_level`

AS `skin\_condition\_level`,`f`.`level` AS `fabric\_level`

from ((((((((`product` `p` join `product\_type` `pt` on((`pt`.`product\_type\_id` = `p`.`product\_type\_id`)))

join `product\_style\_map` `psm` on((`p`.`product\_code` = `psm`.`product\_code`)))

join `product\_style` `ps` on((`ps`.`product\_style\_id` = `psm`.`product\_style\_id`)))

join `product\_color\_map` `pcm` on((`pcm`.`product\_code` = `p`.`product\_code`)))

join `body\_style\_map` `bsm` on((`bsm`.`product\_style\_id` = `ps`.`product\_style\_id`)))

join `customer` `c` on((`c`.`body\_type\_id` = `bsm`.`body\_type\_id`)))

join `product\_fabric\_map` `pfm` on((`p`.`product\_code` = `pfm`.`product\_code`)))

join `fabric` `f` on((`f`.`fabric\_id` = `pfm`.`fabric\_id`)))

5. Find the rank of product based on the price and transaction date of the product.

SELECT distinct p.prod\_name as product, price, t.transaction\_date,

RANK() OVER(PARTITION BY p.prod\_name ORDER BY price DESC) Rank

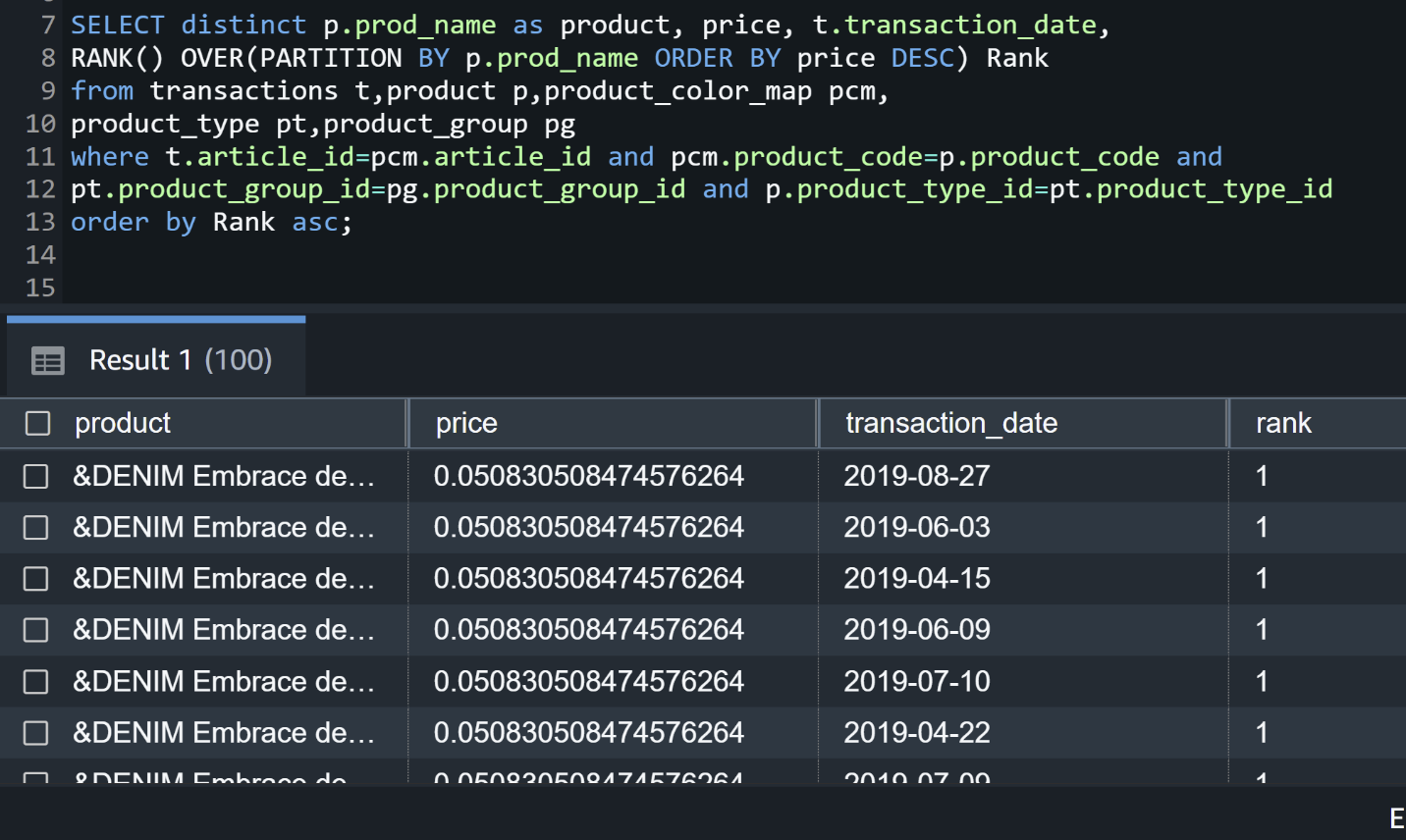
from transactions t,product p,product\_color\_map pcm,

product\_type pt,product\_group pg

where t.article\_id=pcm.article\_id and pcm.product\_code=p.product\_code and

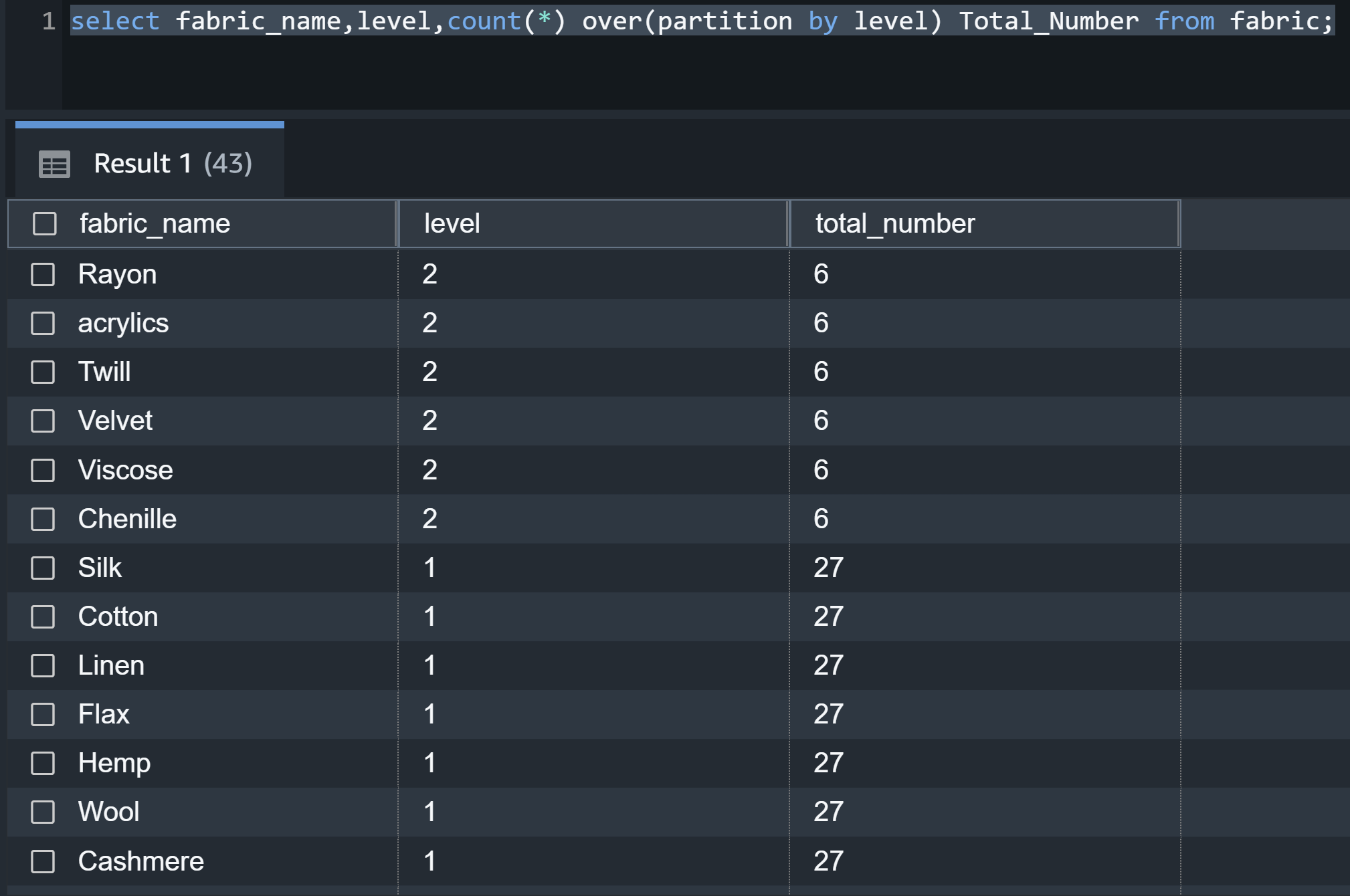
pt.product\_group\_id=pg.product\_group\_id and p.product\_type\_id=pt.product\_type\_id

order by Rank asc;



**5.List the name and total number of fabrics with their quality level.**

select fabric\_name,level,count(\*) over(partition by level) Total\_Number from fabric;

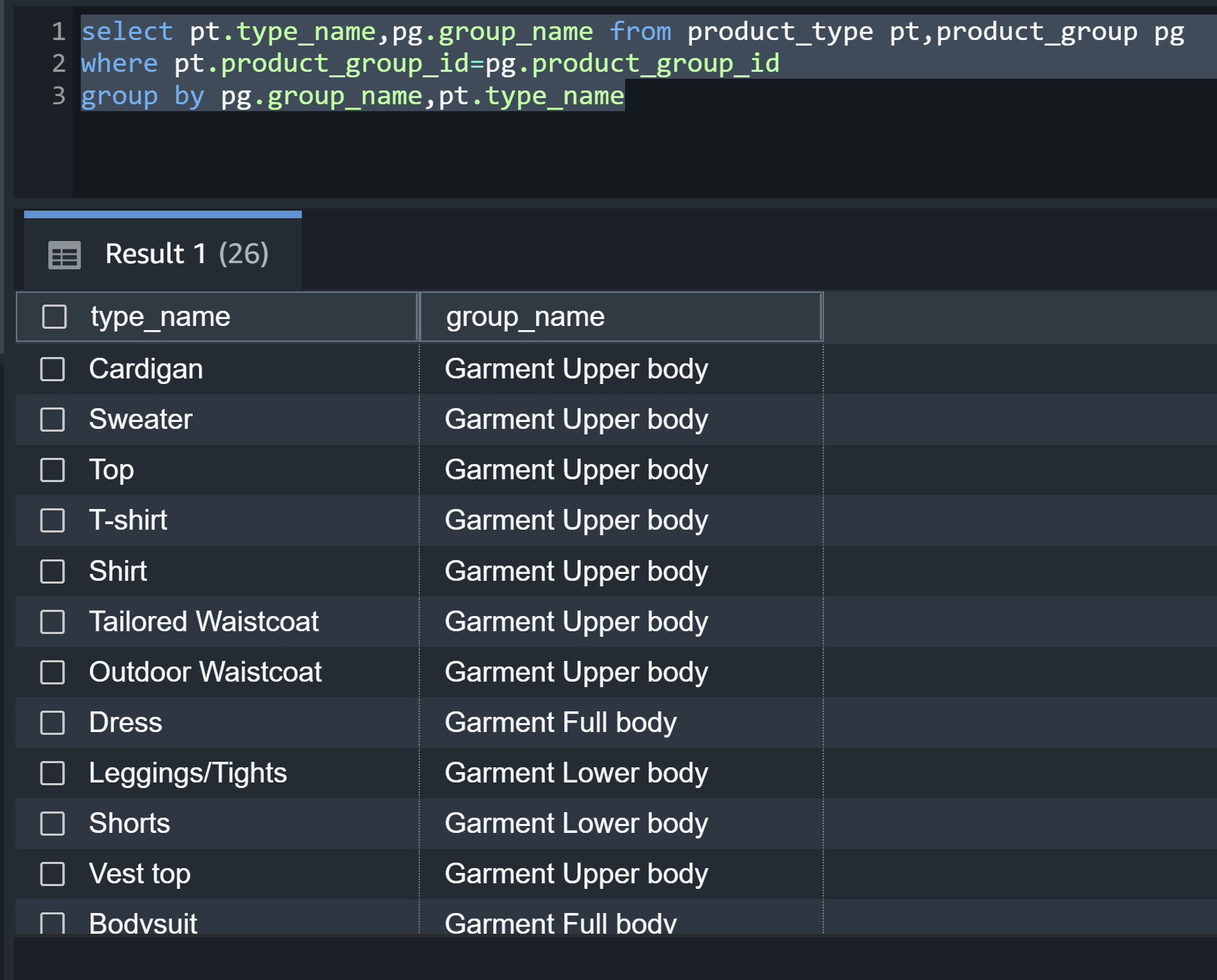


**6. List the product types per group**

select pt.type\_name,pg.group\_name from product\_type pt,product\_group pg

where pt.product\_group\_id=pg.product\_group\_id

group by pg.group\_name,pt.type\_name



**7.We need to find the total number of products with distinct fabrics with their quality level associated with them.**

select distinct fabric\_name, level as Qualitylevel,

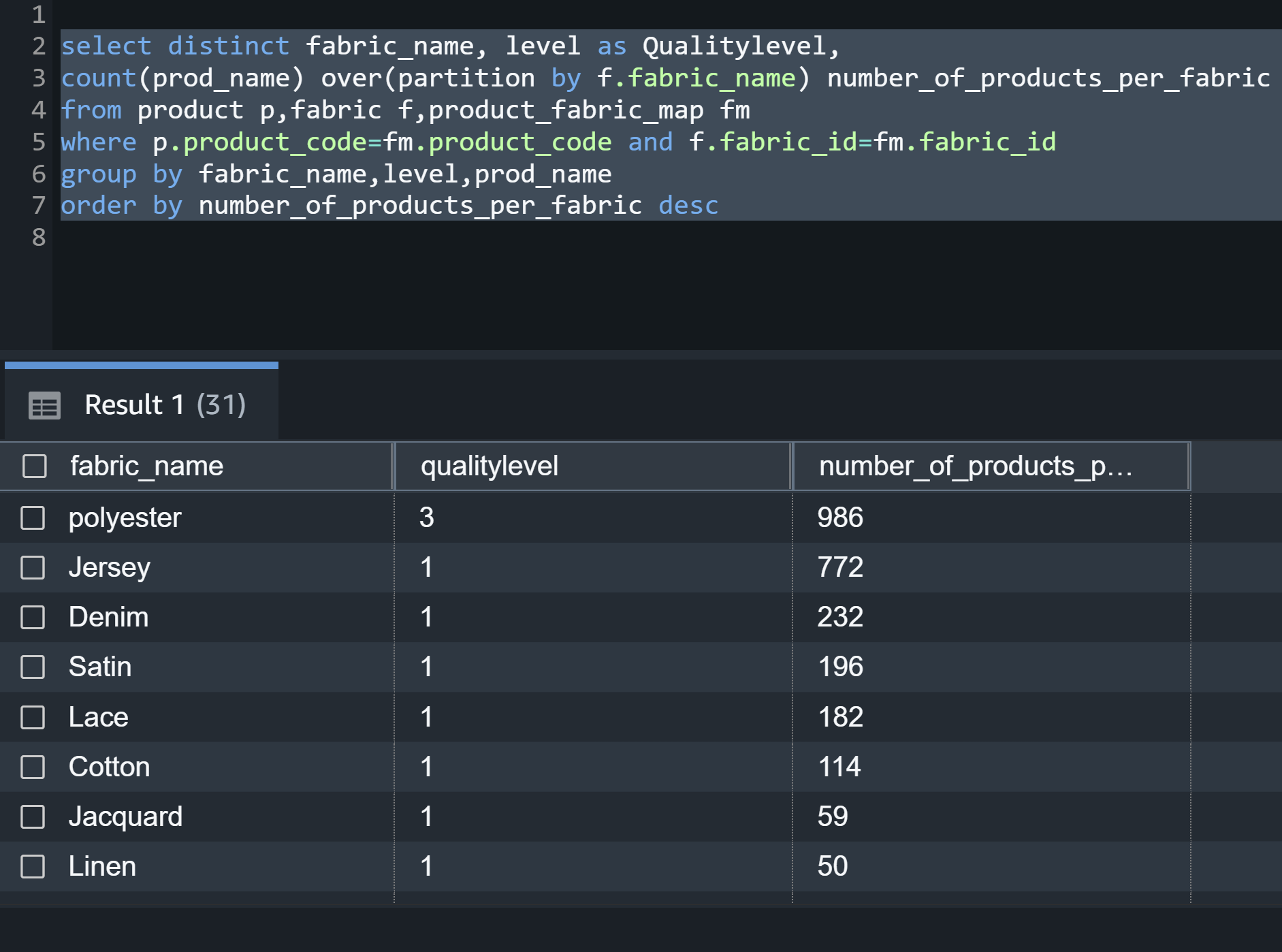
count(prod\_name) over(partition by f.fabric\_name) number\_of\_products\_per\_fabric

from product p,fabric f,product\_fabric\_map fm

where p.product\_code=fm.product\_code and f.fabric\_id=fm.fabric\_id

group by fabric\_name,level,prod\_name

order by number\_of\_products\_per\_fabric desc



**8. List name and number of product styles per body type.**

select style\_name,body\_type\_name,

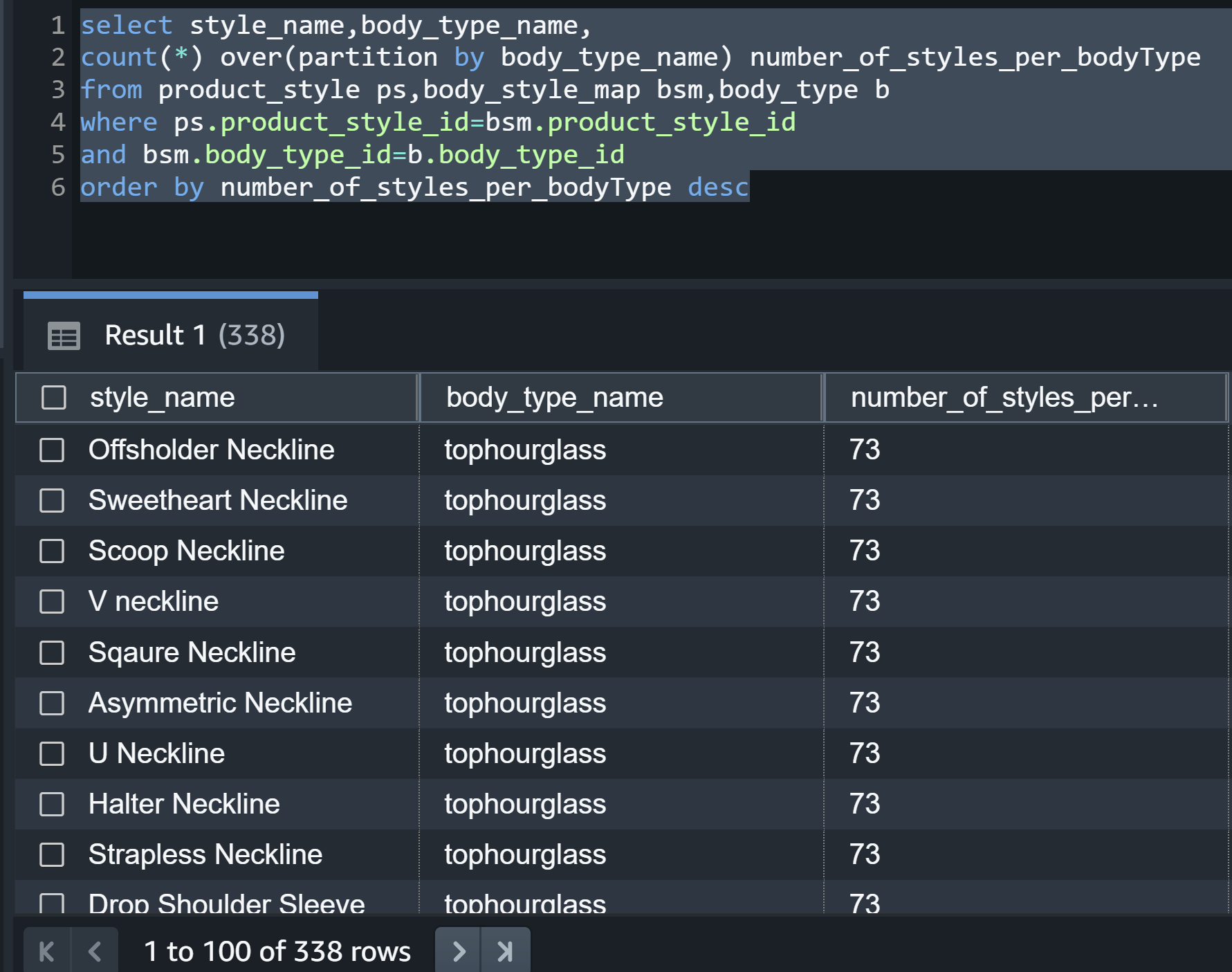
count(\*) over(partition by body\_type\_name) number\_of\_styles\_per\_bodyType

from product\_style ps,body\_style\_map bsm,body\_type b

where ps.product\_style\_id=bsm.product\_style\_id

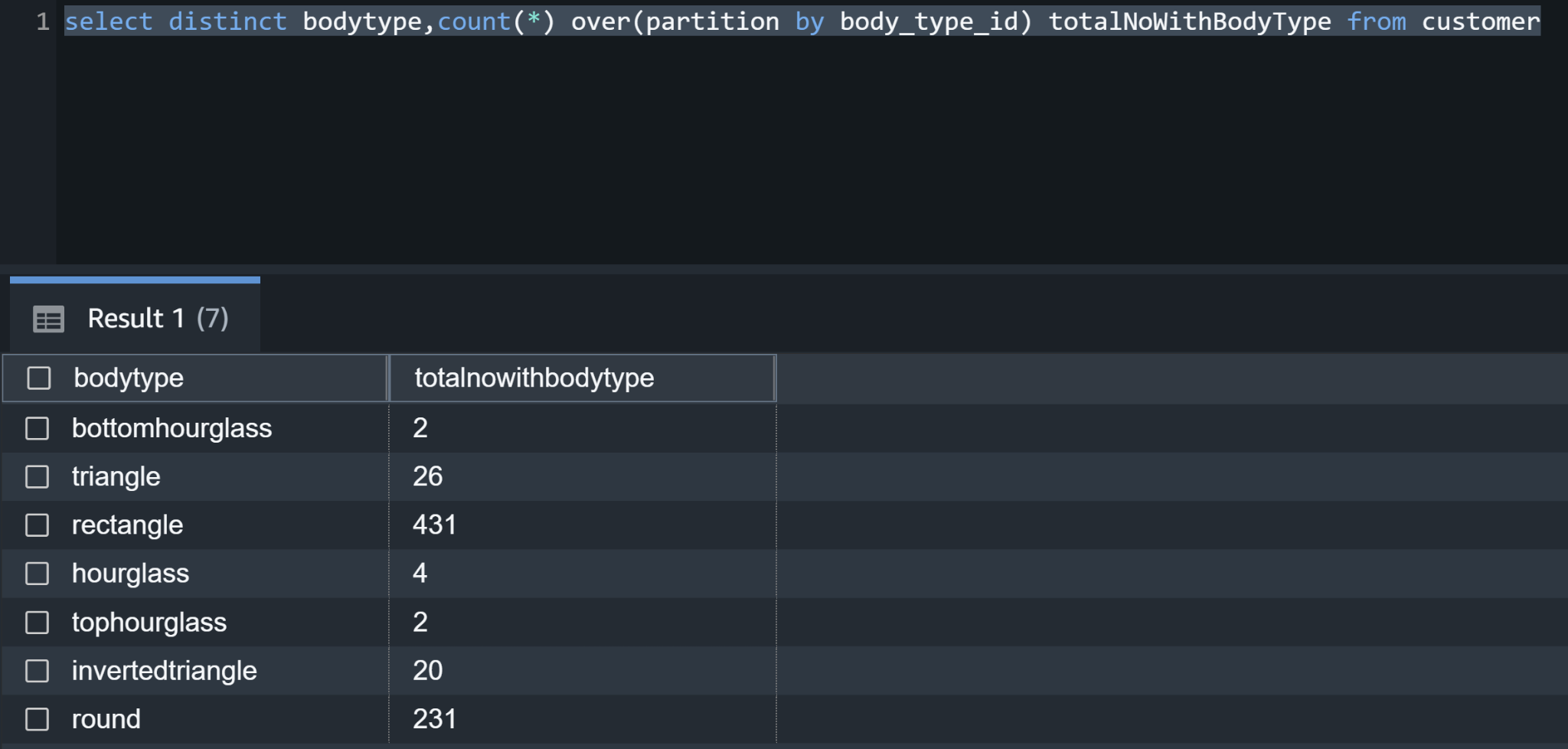
and bsm.body\_type\_id=b.body\_type\_id

order by number\_of\_styles\_per\_bodyType desc



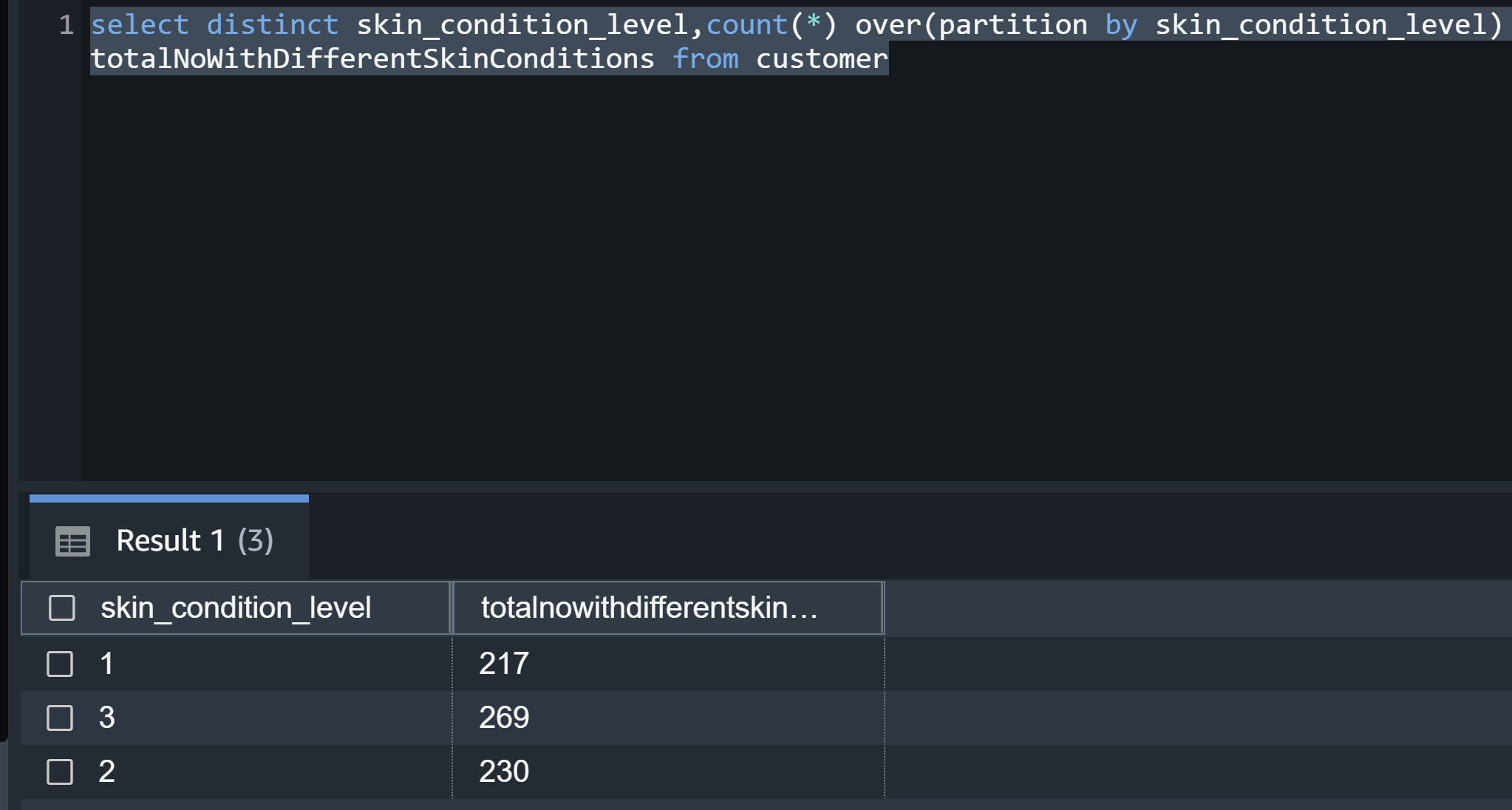
**9.Find total number of customers per bodytype**

select distinct bodytype,count(\*) over(partition by body\_type\_id) totalNoWithBodyType from customer



**10. People with different skin conditions**

select distinct skin\_condition\_level,count(\*) over(partition by skin\_condition\_level) totalNoWithDifferentSkinConditions from customer



**11.** **Average pricing per product group.**

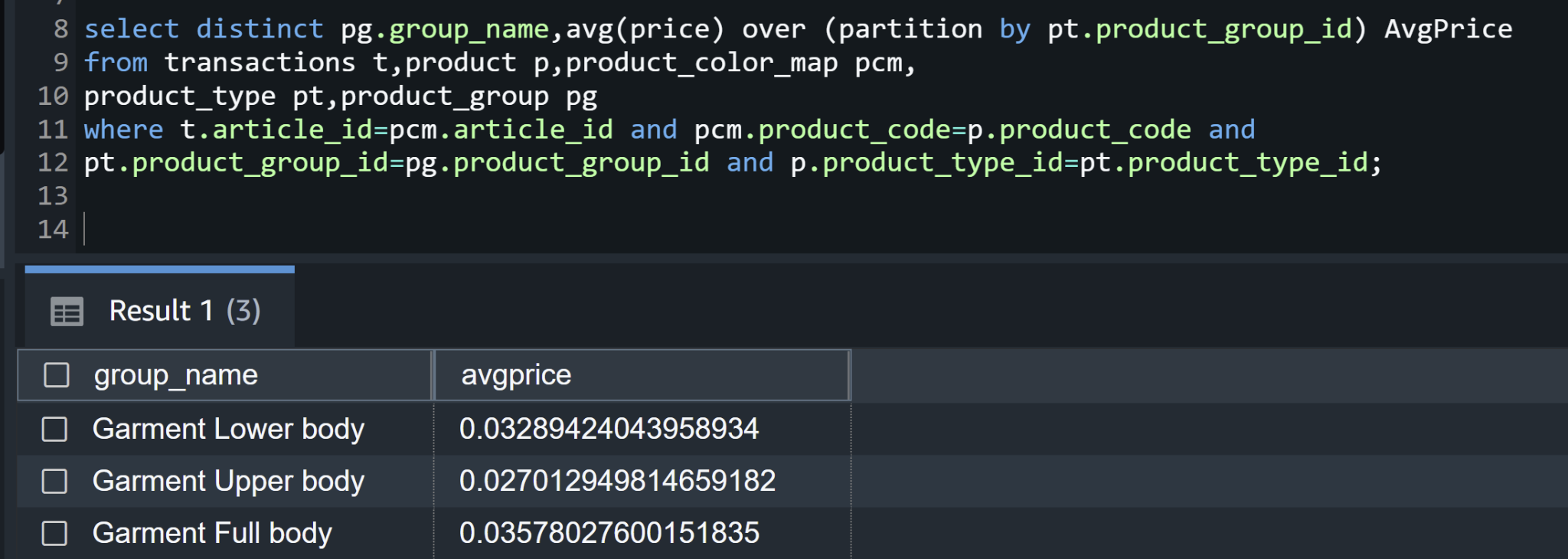
select distinct pg.group\_name,avg(price) over (partition by pt.product\_group\_id) AvgPrice

from transactions t,product p,product\_color\_map pcm,

product\_type pt,product\_group pg

where t.article\_id=pcm.article\_id and pcm.product\_code=p.product\_code and

pt.product\_group\_id=pg.product\_group\_id and p.product\_type\_id=pt.product\_type\_id;



**12.Number of transactions for every quarter of the year**

select extract(YEAR from transaction\_date) as year,

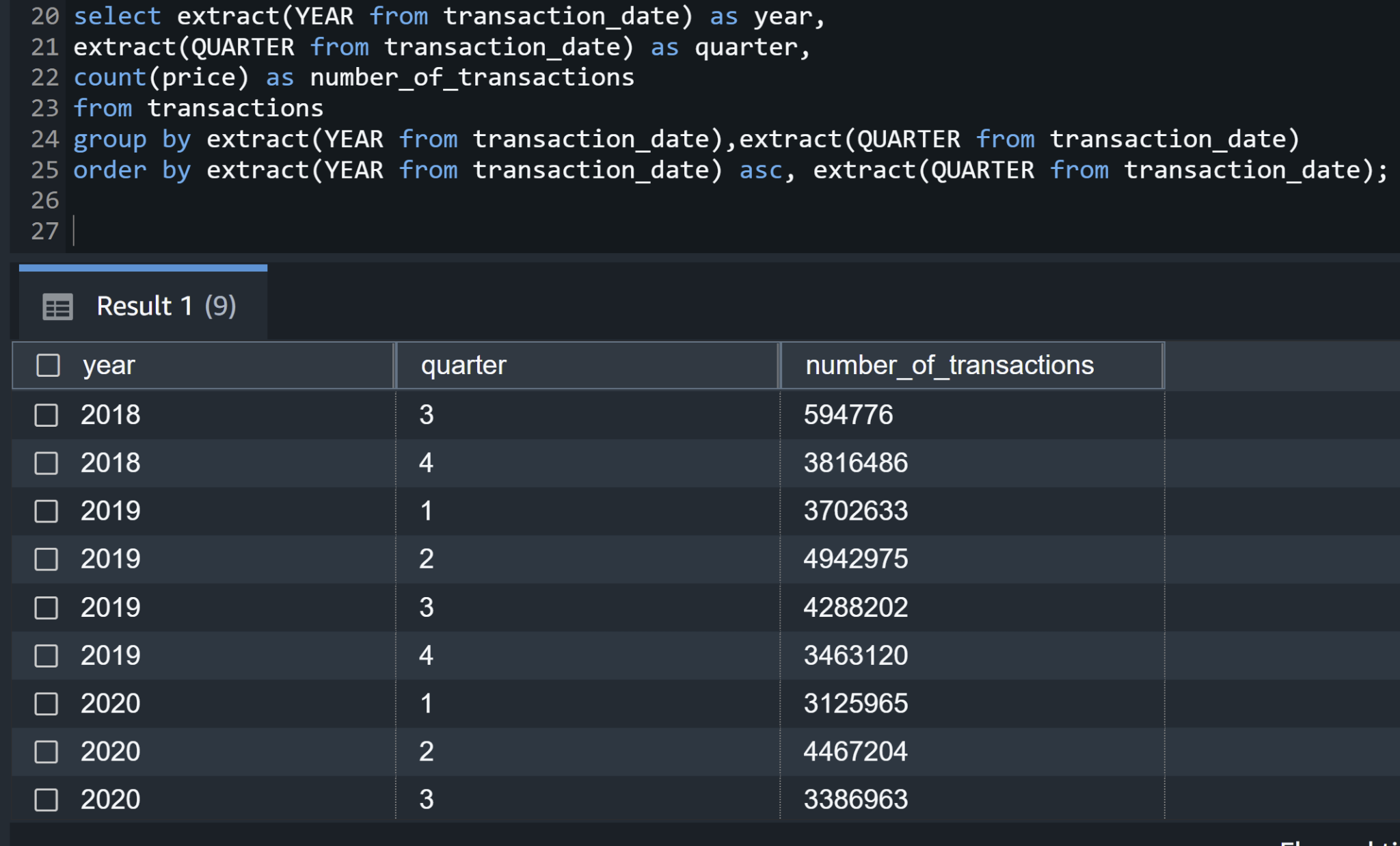
extract(QUARTER from transaction\_date) as quarter,

count(price) as number\_of\_transactions

from transactions

group by extract(YEAR from transaction\_date),extract(QUARTER from transaction\_date)

order by extract(YEAR from transaction\_date) asc, extract(QUARTER from transaction\_date);



**13.** **List Per day transaction running amount per product type for year 2020.**

select distinct(t.transaction\_date) , pt.type\_name,

sum(t.price) over (PARTITION BY p.prod\_name ORDER BY t.transaction\_date ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) as transaction\_runnin\_total

from transactions t,product p,product\_color\_map pcm,

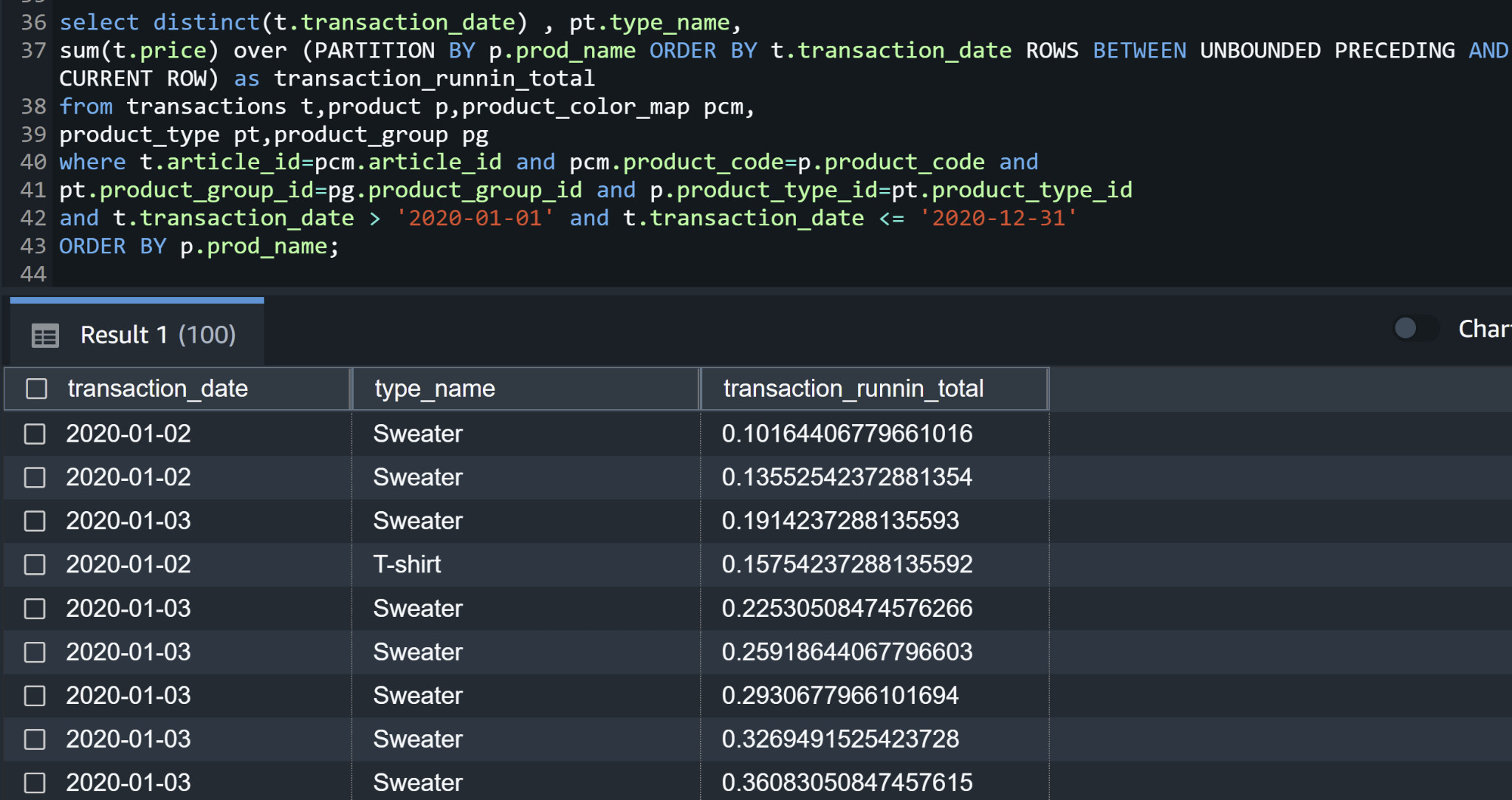
product\_type pt,product\_group pg

where t.article\_id=pcm.article\_id and pcm.product\_code=p.product\_code and

pt.product\_group\_id=pg.product\_group\_id and p.product\_type\_id=pt.product\_type\_id

and t.transaction\_date > '2020-01-01' and t.transaction\_date <= '2020-12-31'

ORDER BY p.prod\_name;



**14. List all the quarters of the years with minimum sales**

select extract(YEAR from transaction\_date) as year,

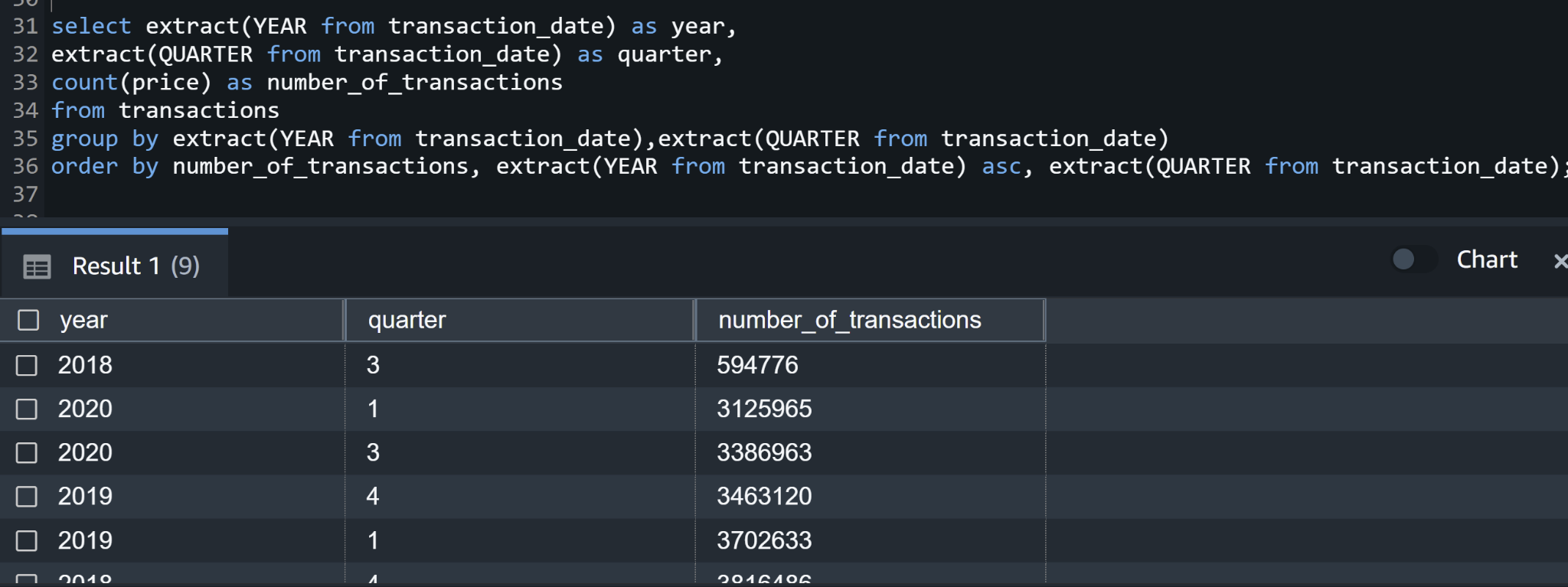
extract(QUARTER from transaction\_date) as quarter,

count(price) as number\_of\_transactions

from transactions

group by extract(YEAR from transaction\_date),extract(QUARTER from transaction\_date)

order by number\_of\_transactions, extract(YEAR from transaction\_date) asc, extract(QUARTER from transaction\_date);



Since 2018, we do not have much data, therefore we can neglect that value.

The second most minimum value of sales happened in the first quarter of 2020.

**15.** **List the first quarter of 2020 with daily sum price and daily price difference**.

select t.transaction\_date , sum(t.price) as daily\_sum\_price,

(sum(t.price) - lag(sum(t.price)) over ( order by t.transaction\_date))

as daily\_difference

from transactions t,product p,product\_color\_map pcm,

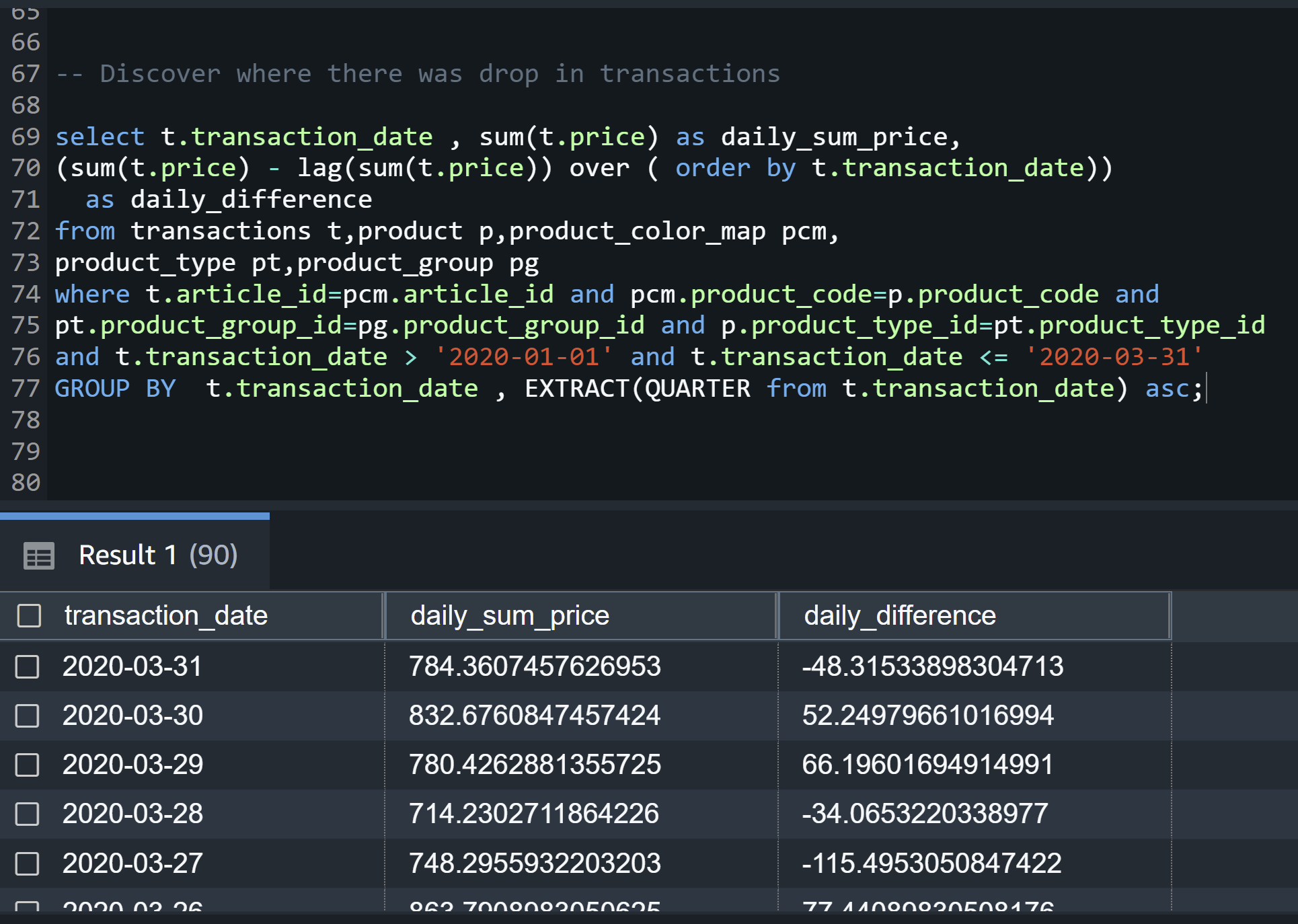
product\_type pt,product\_group pg

where t.article\_id=pcm.article\_id and pcm.product\_code=p.product\_code and

pt.product\_group\_id=pg.product\_group\_id and p.product\_type\_id=pt.product\_type\_id

and t.transaction\_date > '2020-01-01' and t.transaction\_date <= '2020-03-31'

GROUP BY t.transaction\_date , EXTRACT(QUARTER from t.transaction\_date) asc;



**16.****Customer sales for the year 2019 and 2020. Sales for 2020 has declined for all customers, most probably due to the pandemic.**

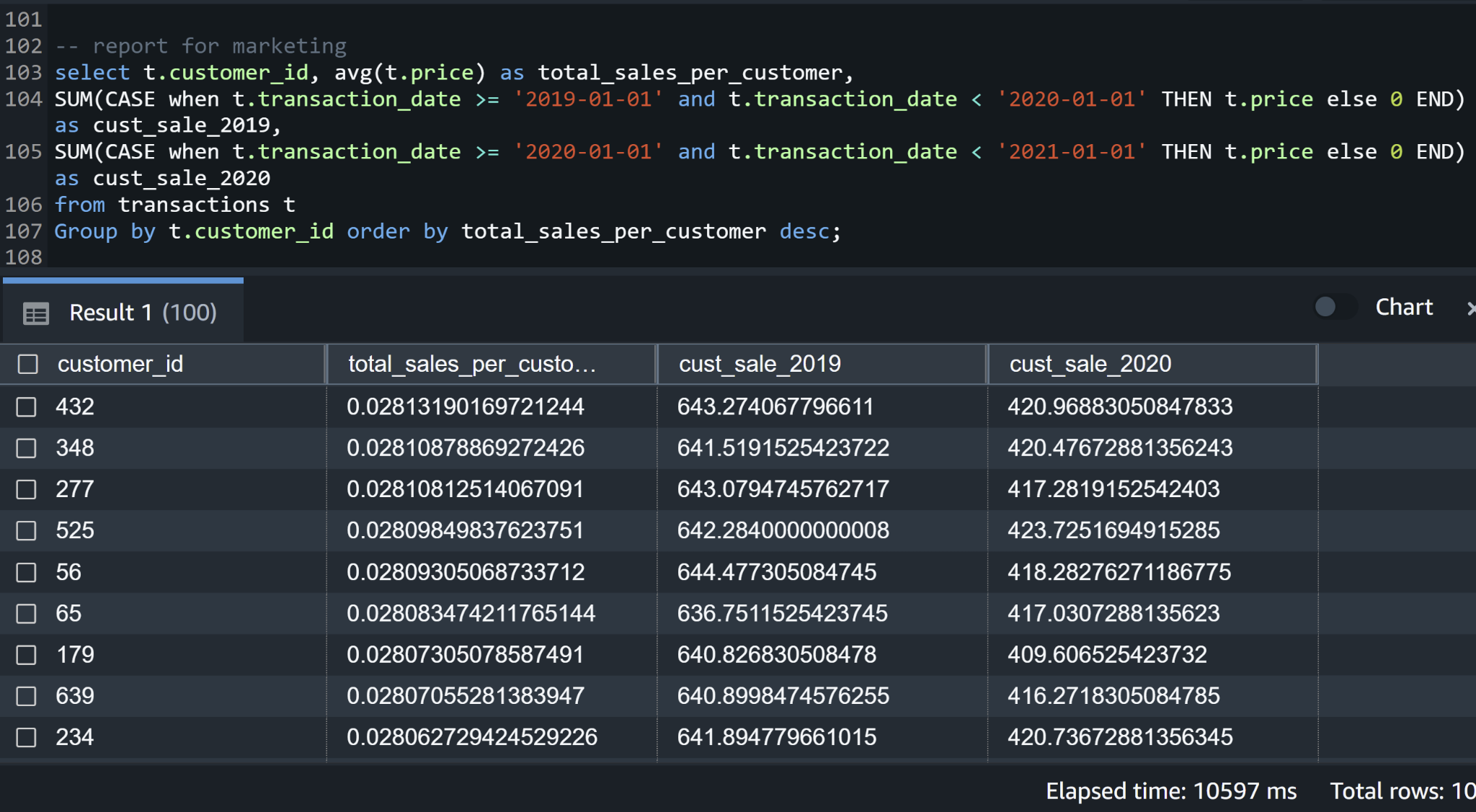
select t.customer\_id, avg(t.price) as total\_sales\_per\_customer,

SUM(CASE when t.transaction\_date >= '2019-01-01' and t.transaction\_date < '2020-01-01' THEN t.price else 0 END) as cust\_sale\_2019,

SUM(CASE when t.transaction\_date >= '2020-01-01' and t.transaction\_date < '2021-01-01' THEN t.price else 0 END) as cust\_sale\_2020

from transactions t

Group by t.customer\_id order by total\_sales\_per\_customer desc;



**17.****Cumulative distribution of customer transitions**

select EXTRACT(MONTH from t.transaction\_date) as Month, t.customer\_id, t.price ,

cume\_dist() OVER(

PARTITION BY EXTRACT(MONTH from t.transaction\_date)

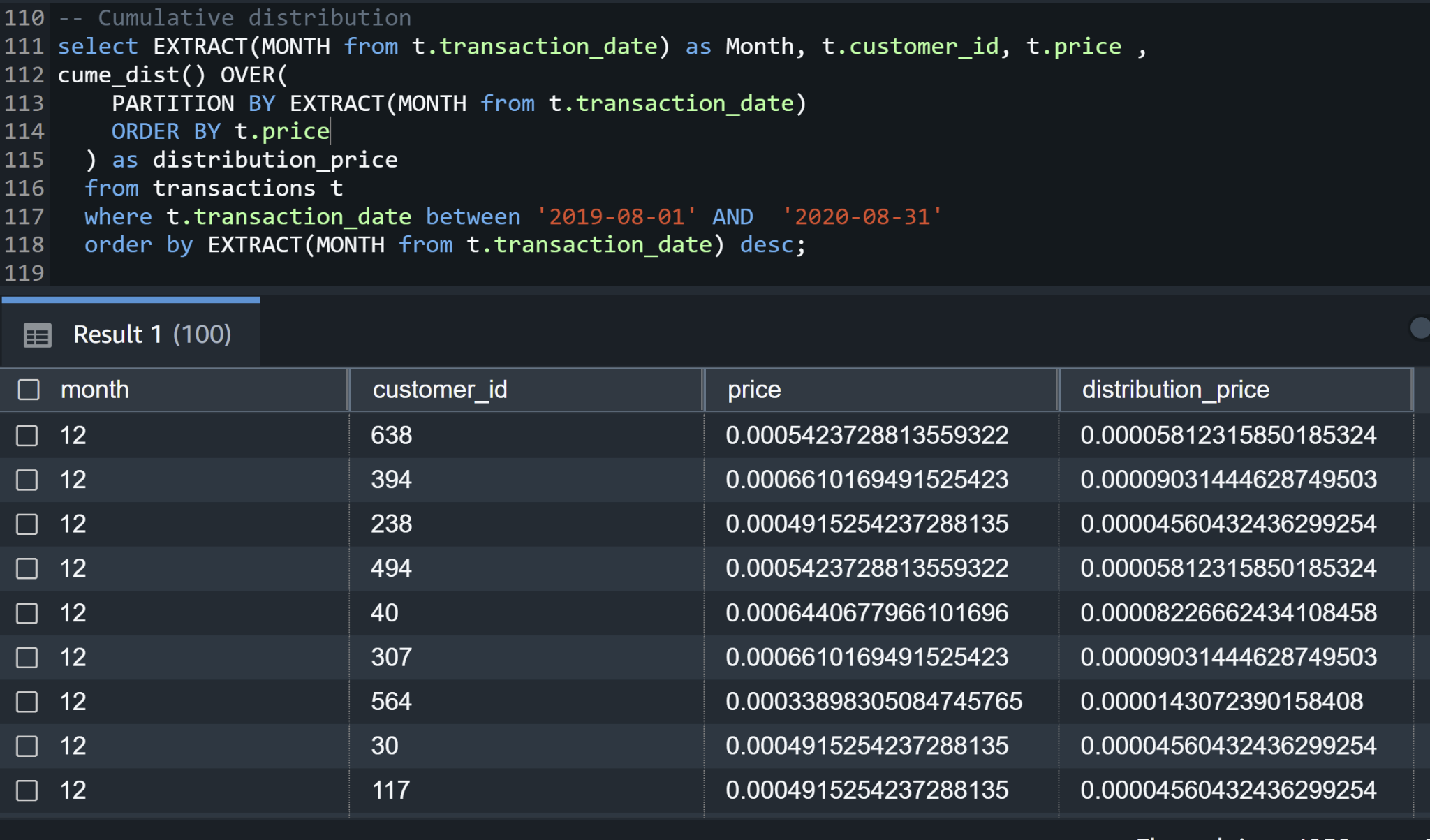
ORDER BY t.price

) as distribution\_price

from transactions t

where t.transaction\_date between '2019-01-01' AND '2020-08-31'

order by EXTRACT(MONTH from t.transaction\_date) desc;



**18.Inventory itemization based on product type, style name, fabric name, color name, count**

SELECT pt.type\_name, ps.style\_name, f.fabric\_name, pc.color\_name, T.count

FROM (

SELECT p.product\_code, p.product\_type\_id, pcm.product\_color\_id, pfm.fabric\_id, psm.product\_style\_id, count(\*) as count

FROM product p

JOIN product\_color\_map pcm on p.product\_code = pcm.product\_code

JOIN product\_fabric\_map pfm on p.product\_code = pfm.product\_code

JOIN product\_style\_map psm on p.product\_code = psm.product\_code

GROUP BY p.product\_code, p.product\_type\_id, pcm.product\_color\_id, pfm.fabric\_id, psm.product\_style\_id

) as T

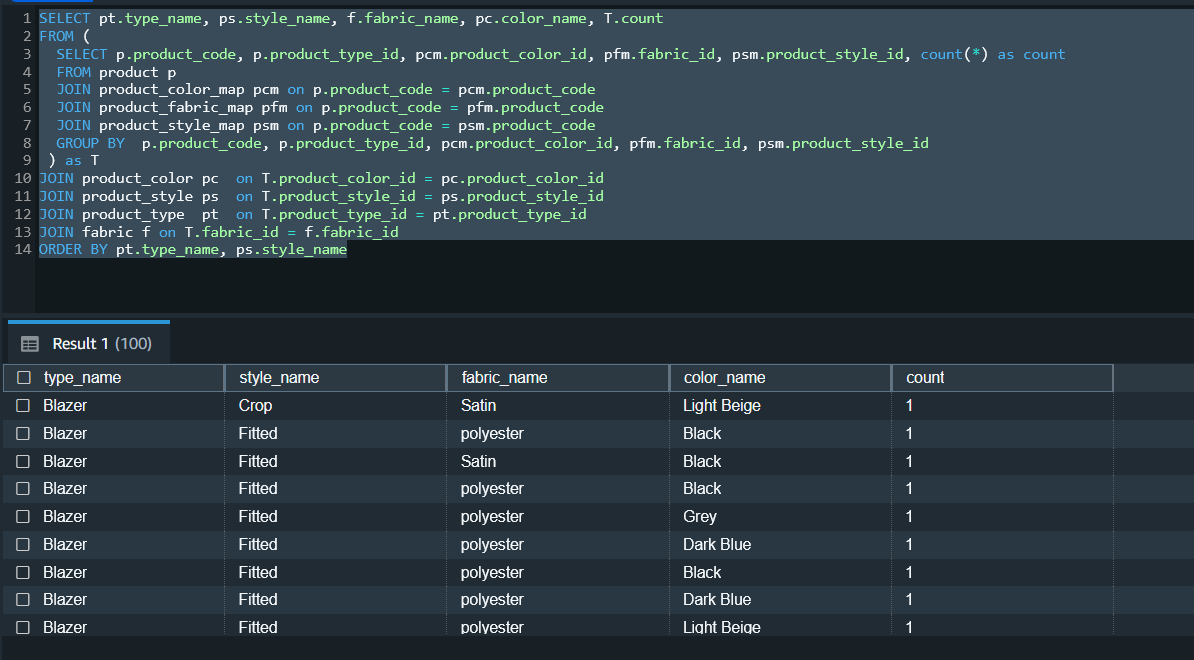
JOIN product\_color pc on T.product\_color\_id = pc.product\_color\_id

JOIN product\_style ps on T.product\_style\_id = ps.product\_style\_id

JOIN product\_type pt on T.product\_type\_id = pt.product\_type\_id

JOIN fabric f on T.fabric\_id = f.fabric\_id

ORDER BY pt.type\_name, ps.style\_name



**19. Top 20 Items in inventory based on Product Type, Product Style, Fabric Name and Color**

SELECT U.\* FROM

(

SELECT pt.type\_name, ps.style\_name, f.fabric\_name, pc.color\_name, T.count

FROM (

SELECT p.product\_code, p.product\_type\_id, pcm.product\_color\_id, pfm.fabric\_id, psm.product\_style\_id, count(\*) as count

FROM product p

JOIN product\_color\_map pcm on p.product\_code = pcm.product\_code

JOIN product\_fabric\_map pfm on p.product\_code = pfm.product\_code

JOIN product\_style\_map psm on p.product\_code = psm.product\_code

GROUP BY p.product\_code, p.product\_type\_id, pcm.product\_color\_id, pfm.fabric\_id, psm.product\_style\_id

) as T

JOIN product\_color pc on T.product\_color\_id = pc.product\_color\_id

JOIN product\_style ps on T.product\_style\_id = ps.product\_style\_id

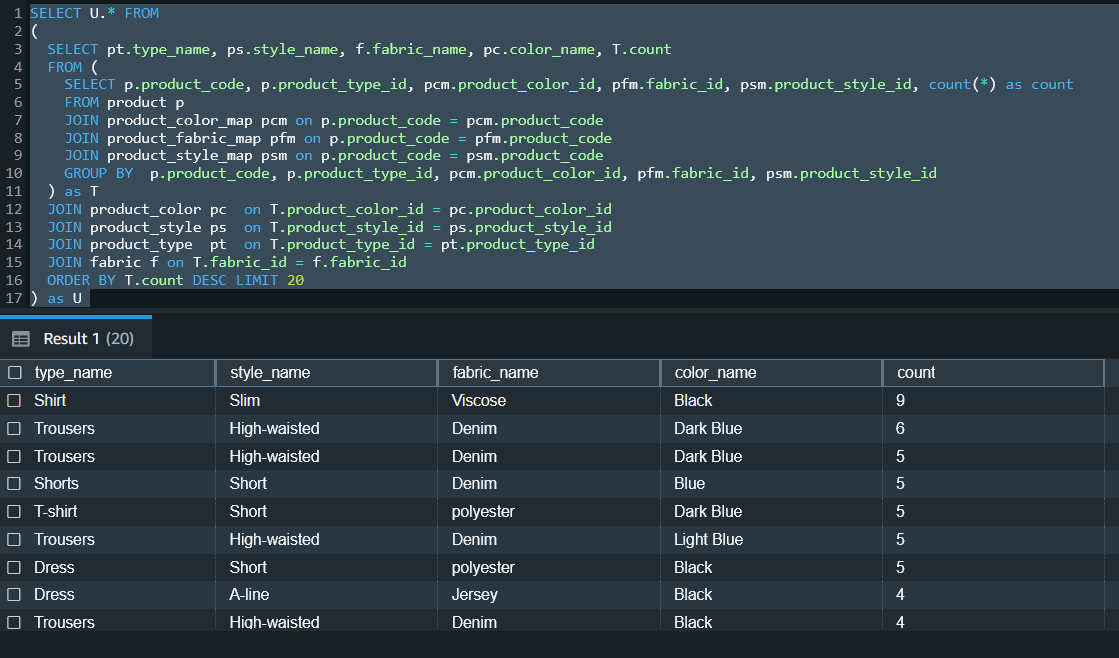
JOIN product\_type pt on T.product\_type\_id = pt.product\_type\_id

JOIN fabric f on T.fabric\_id = f.fabric\_id

ORDER BY T.count DESC LIMIT 20

) as U

ORDER BY U.type\_name, U.style\_name;

****