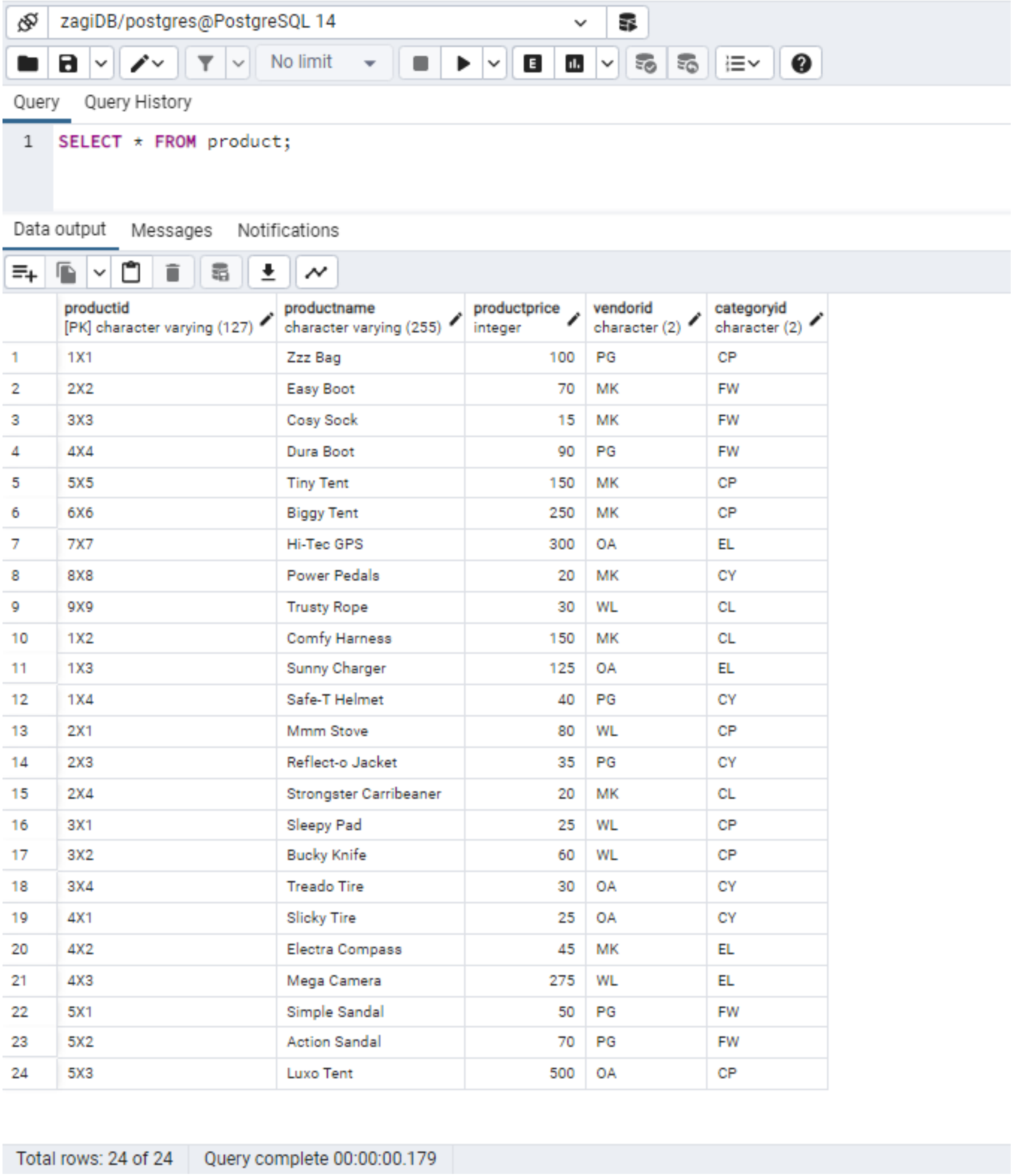
LAB4

SIMRAN- 100377444

QUERY1. Retrieve the entire contents of the Product table (all columns and all rows of the table).

ANS: SELECT * FROM product;



```
SELECT * FROM product;
```

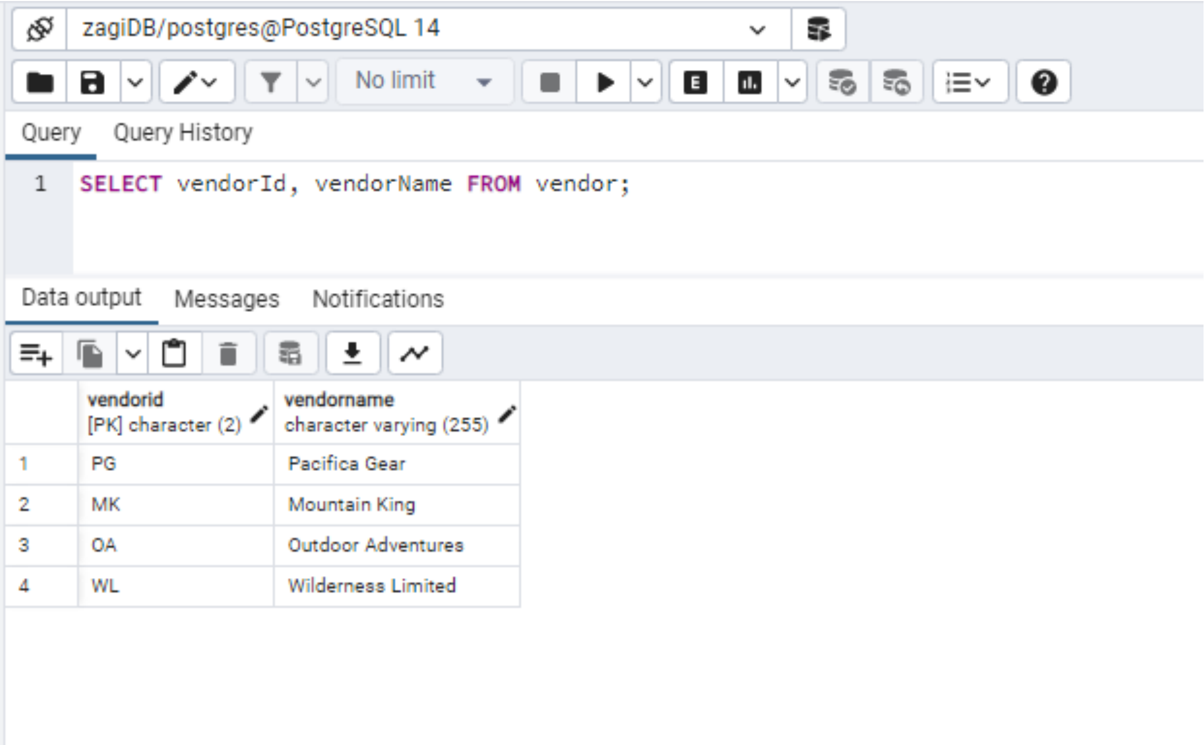
	productid [PK] character varying (127)	productname character varying (255)	productprice integer	vendorid character (2)	categoryid character (2)
1	1X1	Zzz Bag	100	PG	CP
2	2X2	Easy Boot	70	MK	FW
3	3X3	Cosy Sock	15	MK	FW
4	4X4	Dura Boot	90	PG	FW
5	5X5	Tiny Tent	150	MK	CP
6	6X6	Biggy Tent	250	MK	CP
7	7X7	Hi-Tec GPS	300	OA	EL
8	8X8	Power Pedals	20	MK	CY
9	9X9	Trusty Rope	30	WL	CL
10	1X2	Comfy Harness	150	MK	CL
11	1X3	Sunny Charger	125	OA	EL
12	1X4	Safe-T Helmet	40	PG	CY
13	2X1	Mmm Stove	80	WL	CP
14	2X3	Reflect-o Jacket	35	PG	CY
15	2X4	Strongster Carribeaner	20	MK	CL
16	3X1	Sleepy Pad	25	WL	CP
17	3X2	Bucky Knife	60	WL	CP
18	3X4	Treado Tire	30	OA	CY
19	4X1	Slicky Tire	25	OA	CY
20	4X2	Electra Compass	45	MK	EL
21	4X3	Mega Camera	275	WL	EL
22	5X1	Simple Sandal	50	PG	FW
23	5X2	Action Sandal	70	PG	FW
24	5X3	Luxo Tent	500	OA	CP

Total rows: 24 of 24   Query complete 00:00:00.179

Query2. Display the VendorID and VendorName for all vendors.

ANS: SELECT vendorId , vendorName

FROM vendor;



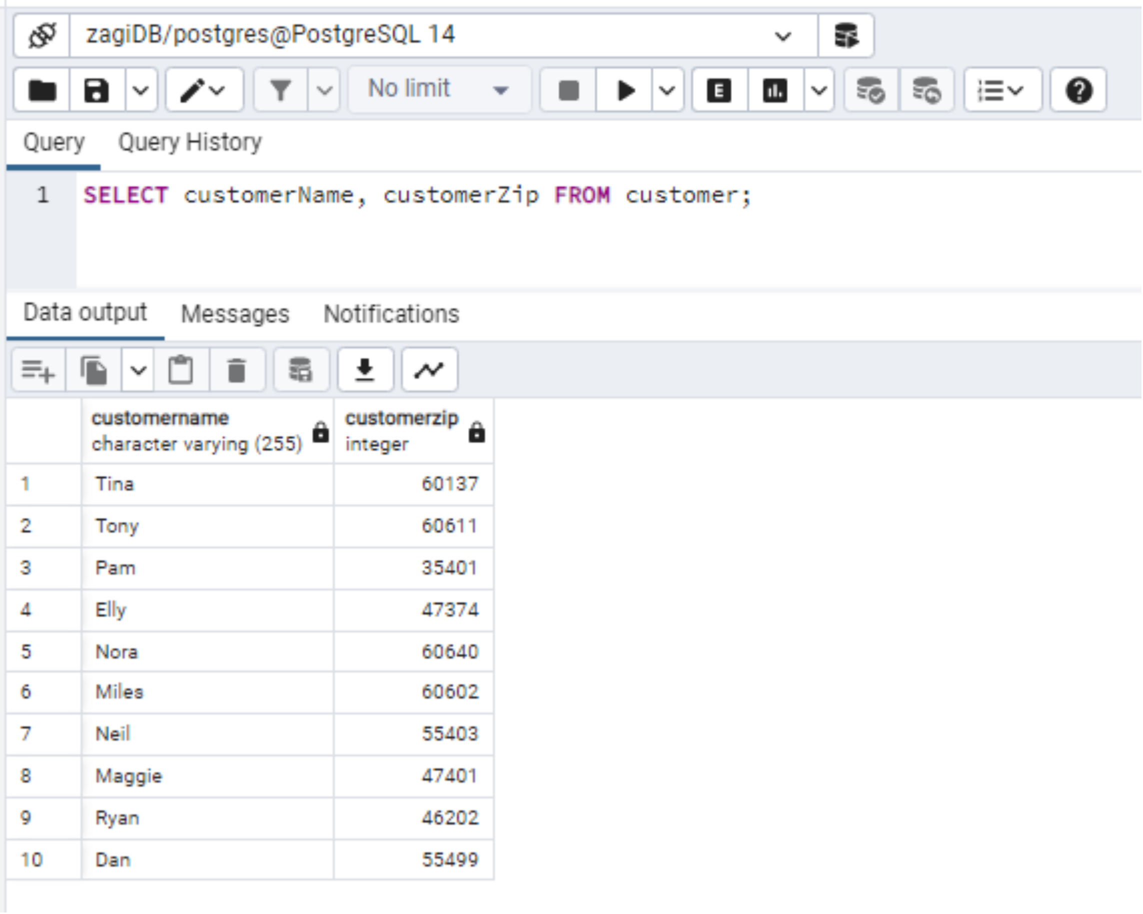
```
SELECT vendorId, vendorName FROM vendor;
```

	vendorid [PK] character (2)	vendorname character varying (255)
1	PG	Pacifica Gear
2	MK	Mountain King
3	OA	Outdoor Adventures
4	WL	Wilderness Limited

Query3. Display the CustomerName and CustomerZip for all customers.

ANS: SELECT CustomerName, CustomerZip

FROM customer;



```
SELECT customerName, customerZip FROM customer;
```

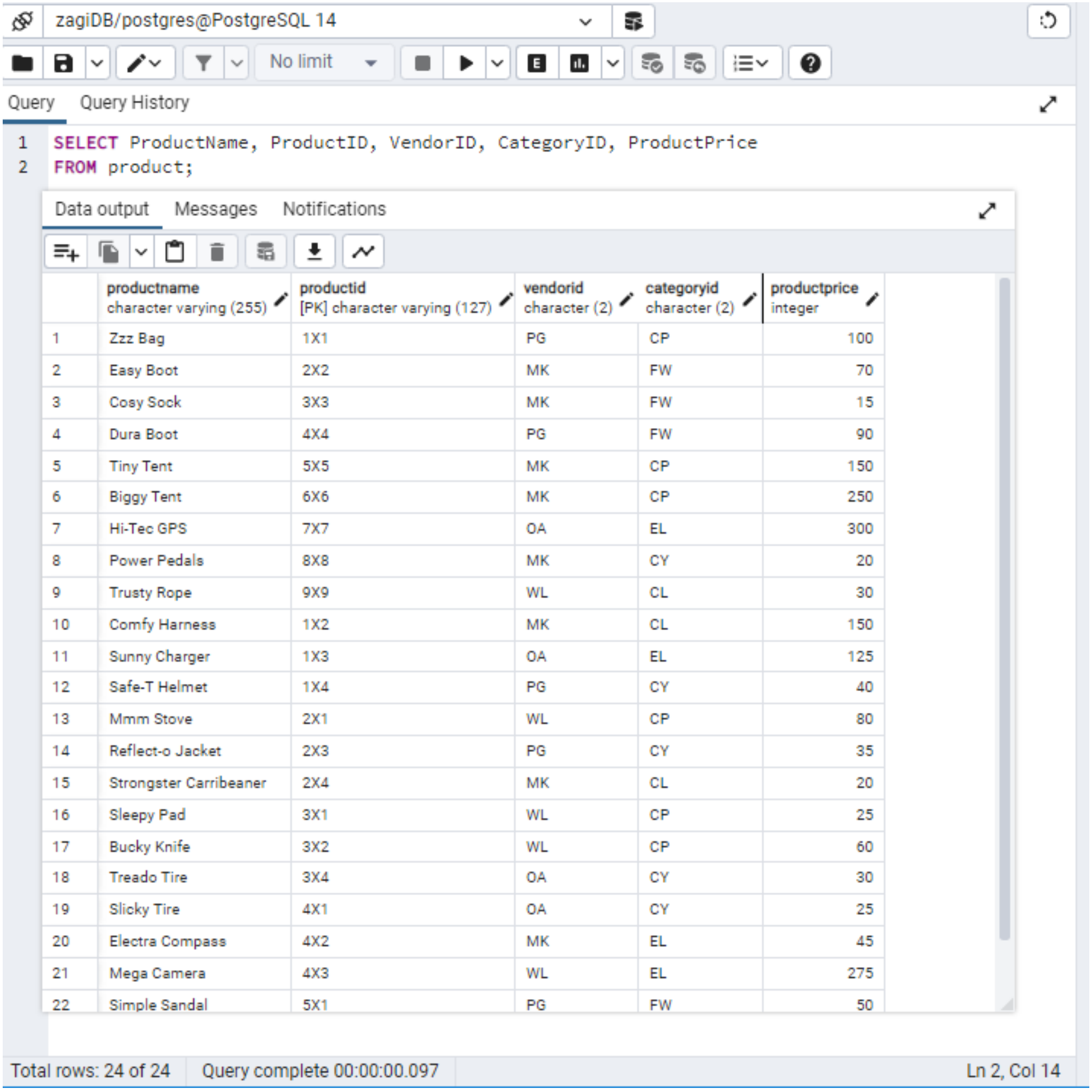
	customername character varying (255)	customerzip integer
1	Tina	60137
2	Tony	60611
3	Pam	35401
4	Elly	47374
5	Nora	60640
6	Miles	60602
7	Neil	55403
8	Maggie	47401
9	Ryan	46202
10	Dan	55499

QUERY4. Retrieve the entire contents of the table PRODUCT. The columns must be displayed in the following order: ProductName, ProductID, VendorID, CategoryID, ProductPrice.

ANS:

SELECT ProductName, ProductID, VendorID, CategoryID, ProductPrice

FROM product;



```
SELECT ProductName, ProductID, VendorID, CategoryID, ProductPrice
FROM product;
```

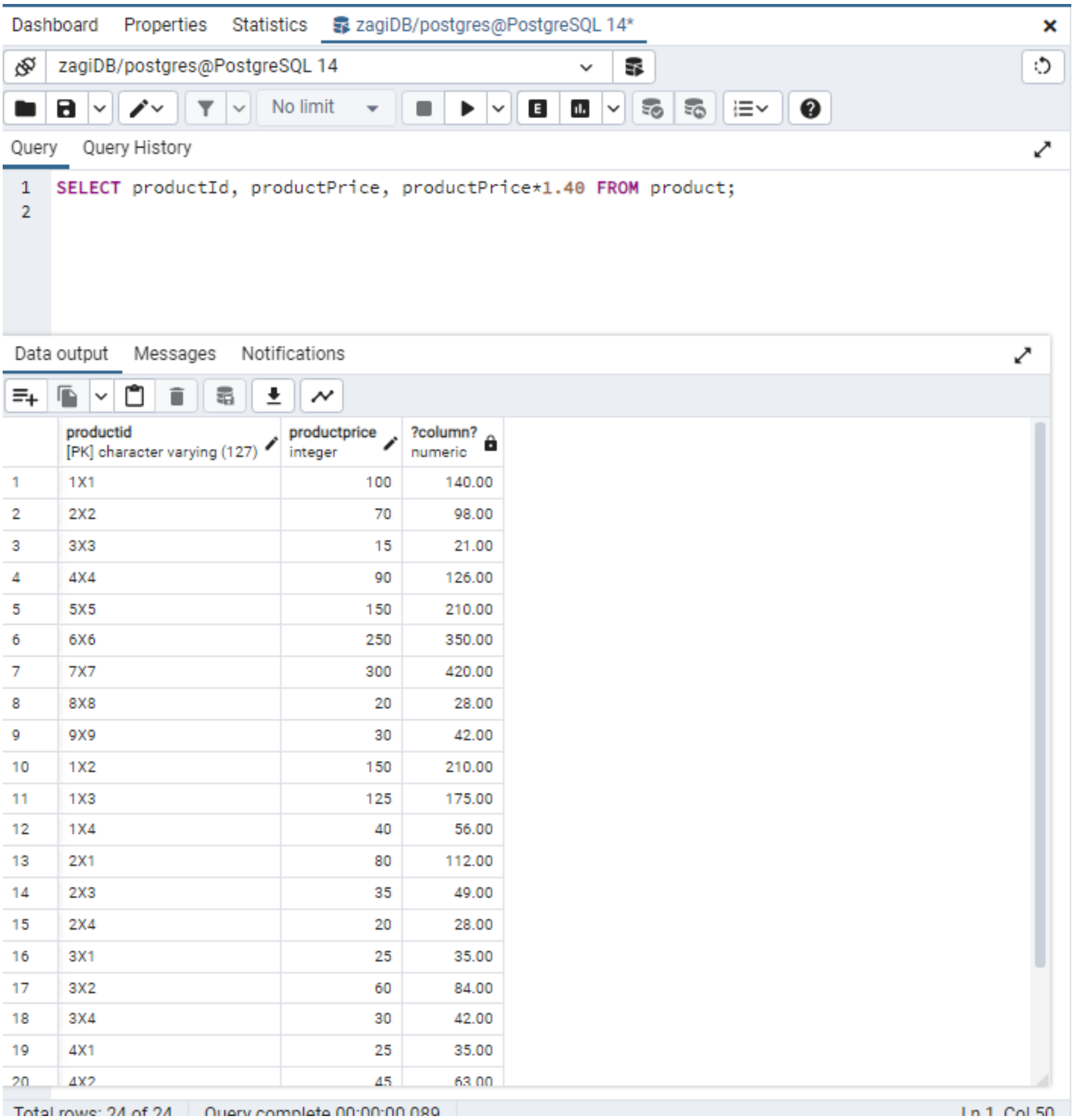
	productname character varying (255)	productid [PK] character varying (127)	vendorid character (2)	categoryid character (2)	productprice integer
1	Zzz Bag	1X1	PG	CP	100
2	Easy Boot	2X2	MK	FW	70
3	Cosy Sock	3X3	MK	FW	15
4	Dura Boot	4X4	PG	FW	90
5	Tiny Tent	5X5	MK	CP	150
6	Biggy Tent	6X6	MK	CP	250
7	Hi-Tec GPS	7X7	OA	EL	300
8	Power Pedals	8X8	MK	CY	20
9	Trusty Rope	9X9	WL	CL	30
10	Comfy Harness	1X2	MK	CL	150
11	Sunny Charger	1X3	OA	EL	125
12	Safe-T Helmet	1X4	PG	CY	40
13	Mmm Stove	2X1	WL	CP	80
14	Reflect-o Jacket	2X3	PG	CY	35
15	Strongster Carribeaner	2X4	MK	CL	20
16	Sleepy Pad	3X1	WL	CP	25
17	Bucky Knife	3X2	WL	CP	60
18	Treado Tire	3X4	OA	CY	30
19	Slicky Tire	4X1	OA	CY	25
20	Electra Compass	4X2	MK	EL	45
21	Mega Camera	4X3	WL	EL	275
22	Simple Sandal	5X1	PG	FW	50

Total rows: 24 of 24   Query complete 00:00:00.097   Ln 2, Col 14

QUERY5. For the table PRODUCT, display 3 columns ProductID, ProductPrice, and a column showing ProductPrice increased by 40%

ANS: SELECT productID, productPrice, productPrice*1.40

FROM product;



```
SELECT productId, productPrice, productPrice*1.40 FROM product;
```

	productid [PK] character varying (127)	productprice integer	?column? numeric
1	1X1	100	140.00
2	2X2	70	98.00
3	3X3	15	21.00
4	4X4	90	126.00
5	5X5	150	210.00
6	6X6	250	350.00
7	7X7	300	420.00
8	8X8	20	28.00
9	9X9	30	42.00
10	1X2	150	210.00
11	1X3	125	175.00
12	1X4	40	56.00
13	2X1	80	112.00
14	2X3	35	49.00
15	2X4	20	28.00
16	3X1	25	35.00
17	3X2	60	84.00
18	3X4	30	42.00
19	4X1	25	35.00
20	4X2	45	63.00

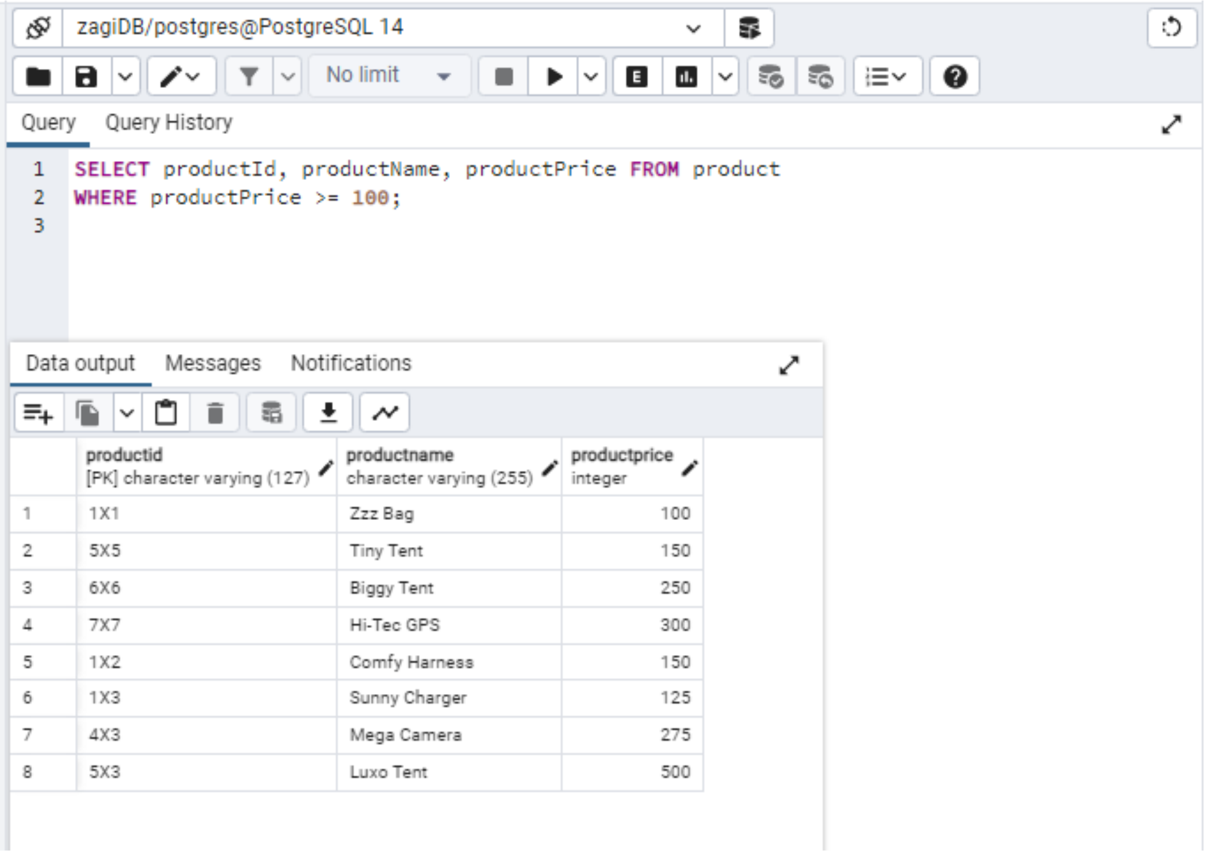
Total rows: 24 of 24 Query complete 00:00:00.089 Ln 1, Col 50

QUERY6. Display the ProductID, ProductName, and ProductPrice for products with a ProductPrice of $100 or higher.

ANS: SELECT productID, productName, productPrice

FROM product

WHERE productPrice >= 100;



```sql
SELECT productId, productName, productPrice FROM product
WHERE productPrice >= 100;
```

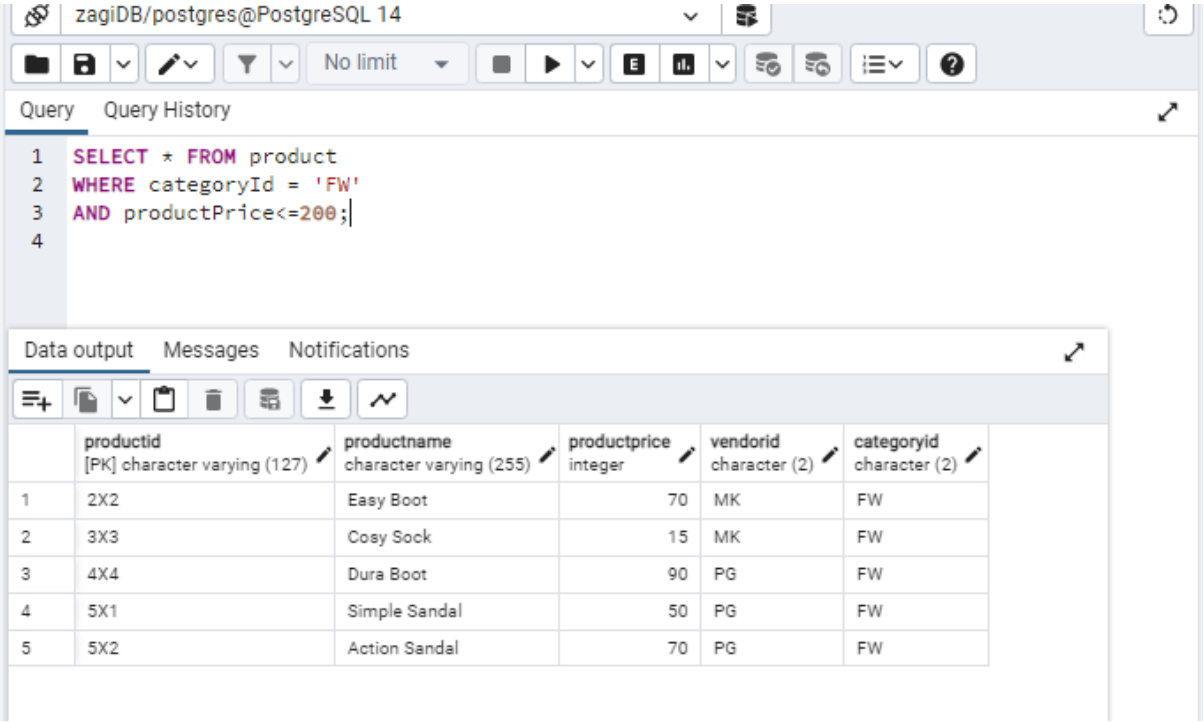
	productid [PK] character varying (127)	productname character varying (255)	productprice integer
1	1X1	Zzz Bag	100
2	5X5	Tiny Tent	150
3	6X6	Biggy Tent	250
4	7X7	Hi-Tec GPS	300
5	1X2	Comfy Harness	150
6	1X3	Sunny Charger	125
7	4X3	Mega Camera	275
8	5X3	Luxo Tent	500

QUERY7. Retrieve the ProductID, ProductName, VendorID, CategoryID, and ProductPrice of products in the FW category whose price is equal to or below $200 (Hint: two conditions in WHERE clause)

ANS: SELECT * FROM product

WHERE categoryId = ‘FW’

AND productPrice <=200;



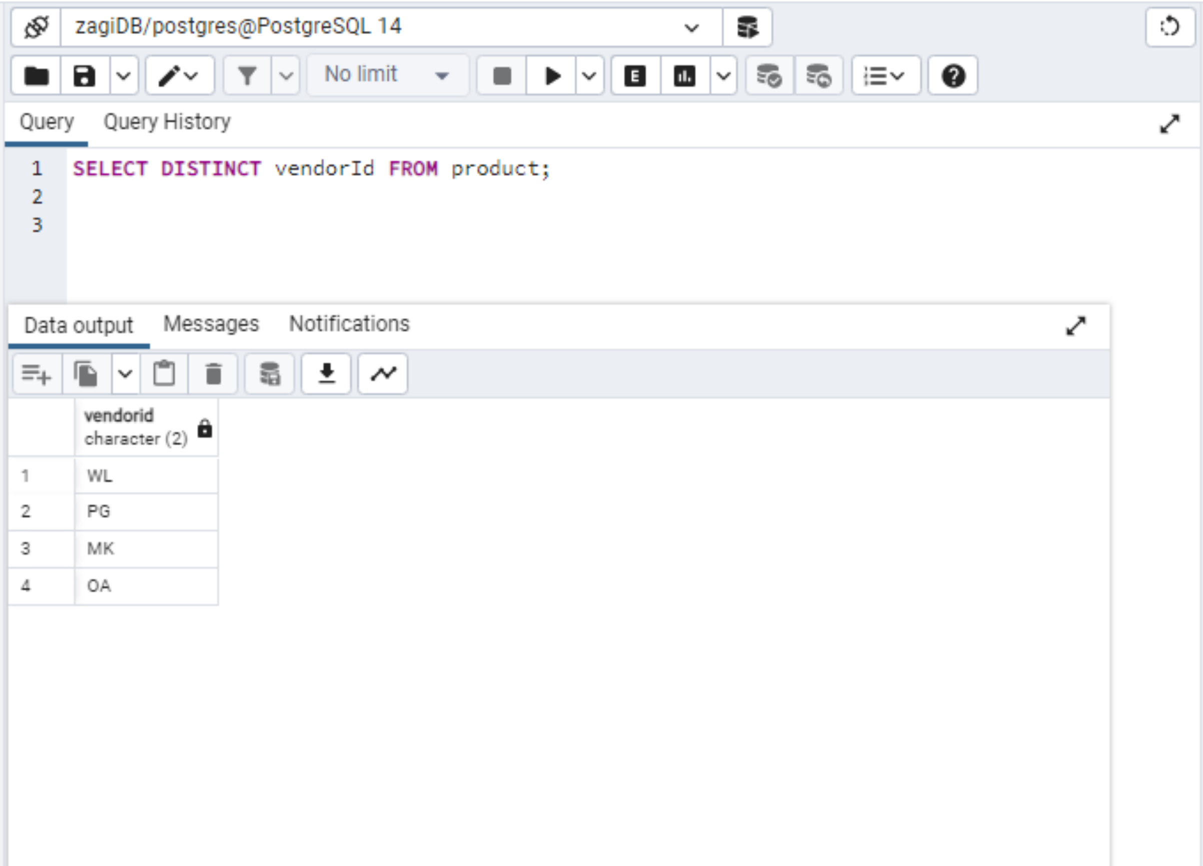
```
SELECT * FROM product
WHERE categoryId = 'FW'
AND productPrice<=200;
```

	productid [PK] character varying (127)	productname character varying (255)	productprice integer	vendorid character (2)	categoryid character (2)
1	2X2	Easy Boot	70	MK	FW
2	3X3	Cosy Sock	15	MK	FW
3	4X4	Dura Boot	90	PG	FW
4	5X1	Simple Sandal	50	PG	FW
5	5X2	Action Sandal	70	PG	FW

QUERY8. Display the VendorID of all vendors that we have a product from them. In the result, we must not see duplicate vendorIDs.

ANS: SELECT DISTINCT vendorId

FROM Product;



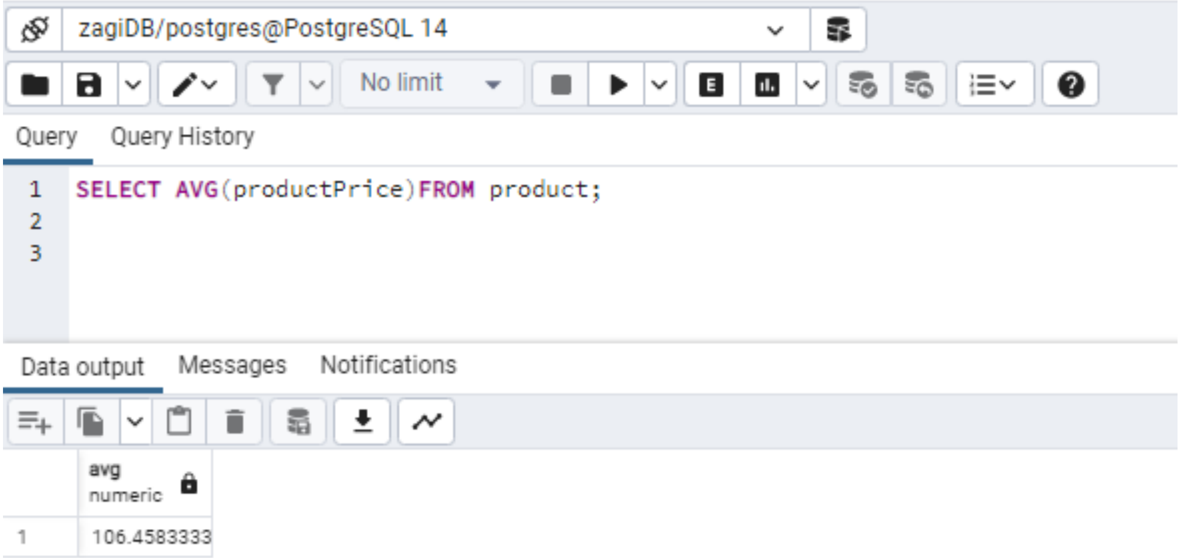
```
SELECT DISTINCT vendorId FROM product;
```

	vendorid character (2)
1	WL
2	PG
3	MK
4	OA

QUERY 9. Retrieve the average price of all products. (Hint: Use AVG)

ANS: SELECT AVG(productPrice)

FROM product;


zagiDB/postgres@PostgreSQL 14

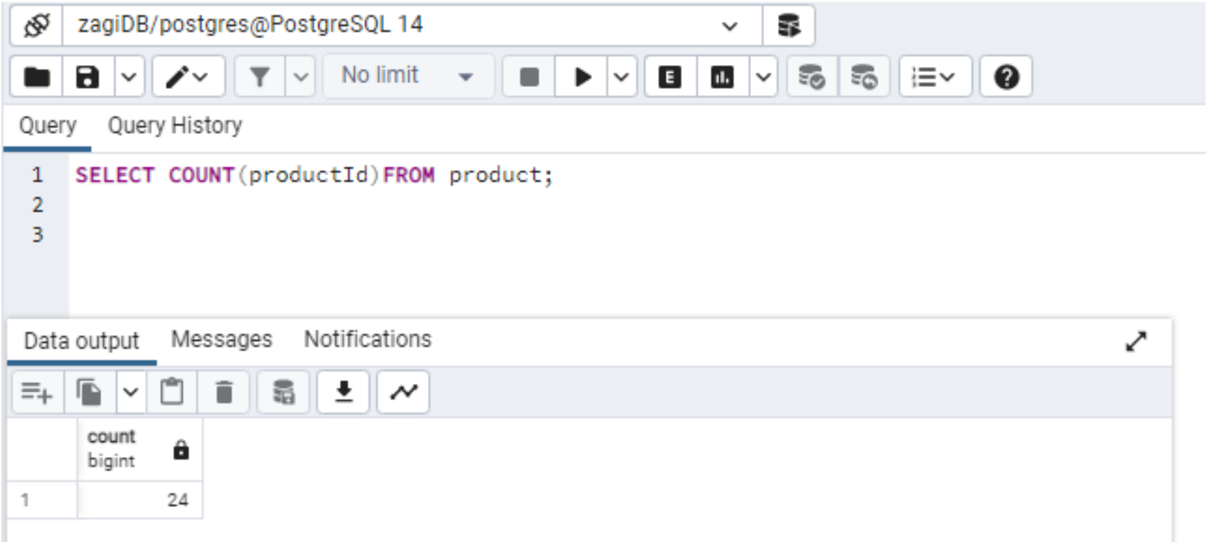
```
SELECT AVG(productPrice)FROM product;
```

```
avg
numeric
1  106.4583333
```

QUERY10. Show how many products are there for sale. (Hint: COUNT)

ANS: SELECT COUNT(productId)

FROM product;


zagiDB/postgres@PostgreSQL 14

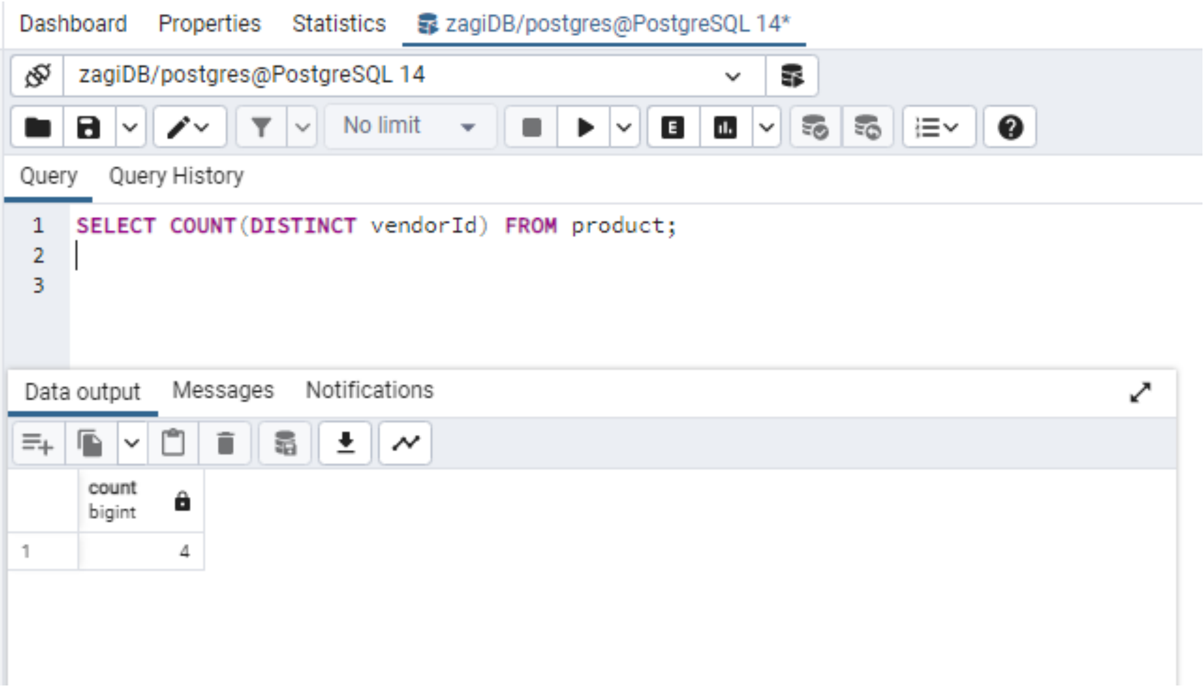
```
SELECT COUNT(productId)FROM product;
```

```
count
bigint
1      24
```

QUERY 11. Count how many distinct vendors are there in the product table. (Hint: the answer is 4)

ANS: SELECT COUNT (DISTINCT vendorId)

FROM product;



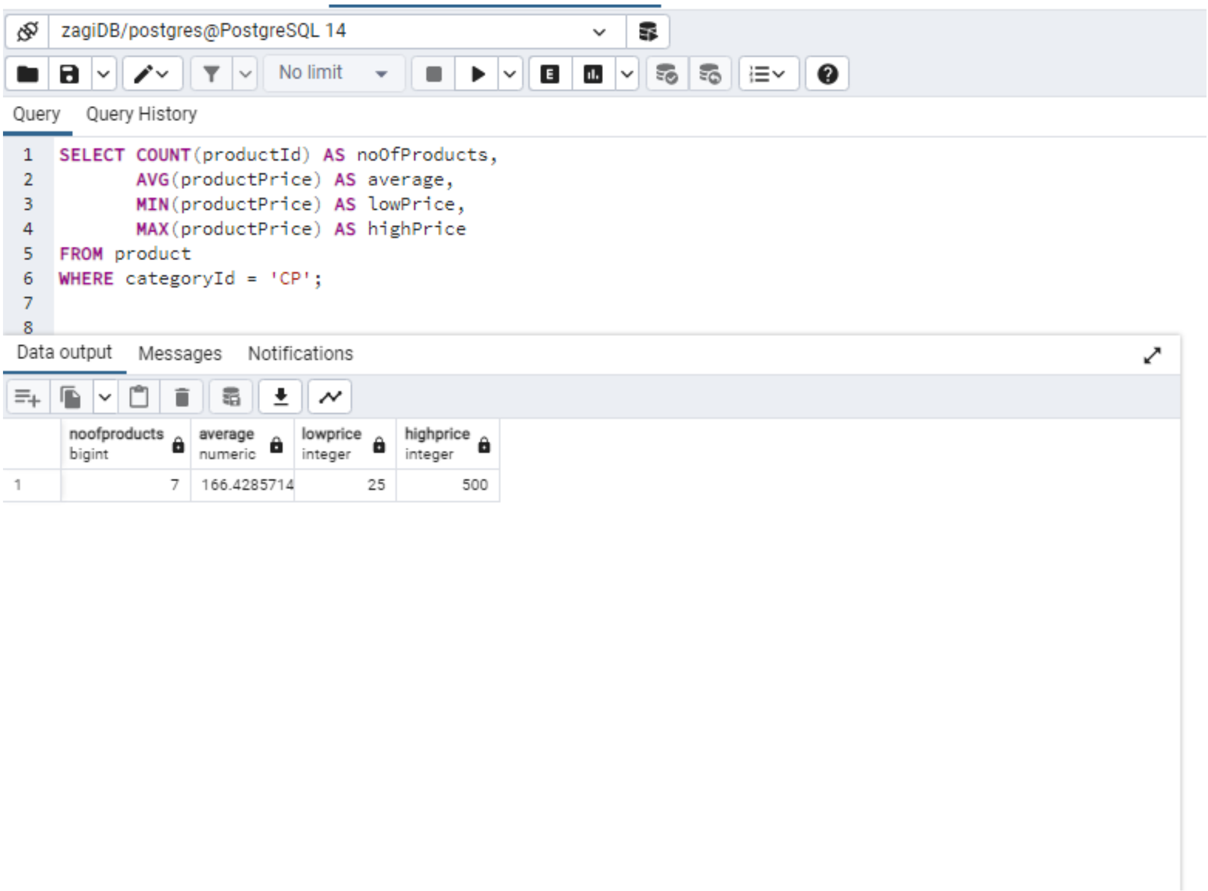
```
SELECT COUNT(DISTINCT vendorId) FROM product;
```

	count bigint
1	4

QUERY12. Retrieve the number of products, average product price, lowest product price, and highest product price in the CP product category.

ANS:

```
SELECT COUNT(productId) AS noOfProducts,
              AVG (productPrice) AS average,
             MIN(productPrice ) AS lowPrice,
             MAX(productPrice) AS highPrice
FROM product
WHERE category = ’CP’;
```



```
SELECT COUNT(productId) AS noOfProducts,
       AVG(productPrice) AS average,
       MIN(productPrice) AS lowPrice,
       MAX(productPrice) AS highPrice
FROM product
WHERE categoryId = 'CP';
```

	noofproducts bigint	average numeric	lowprice integer	highprice integer
1	7	166.4285714	25	500

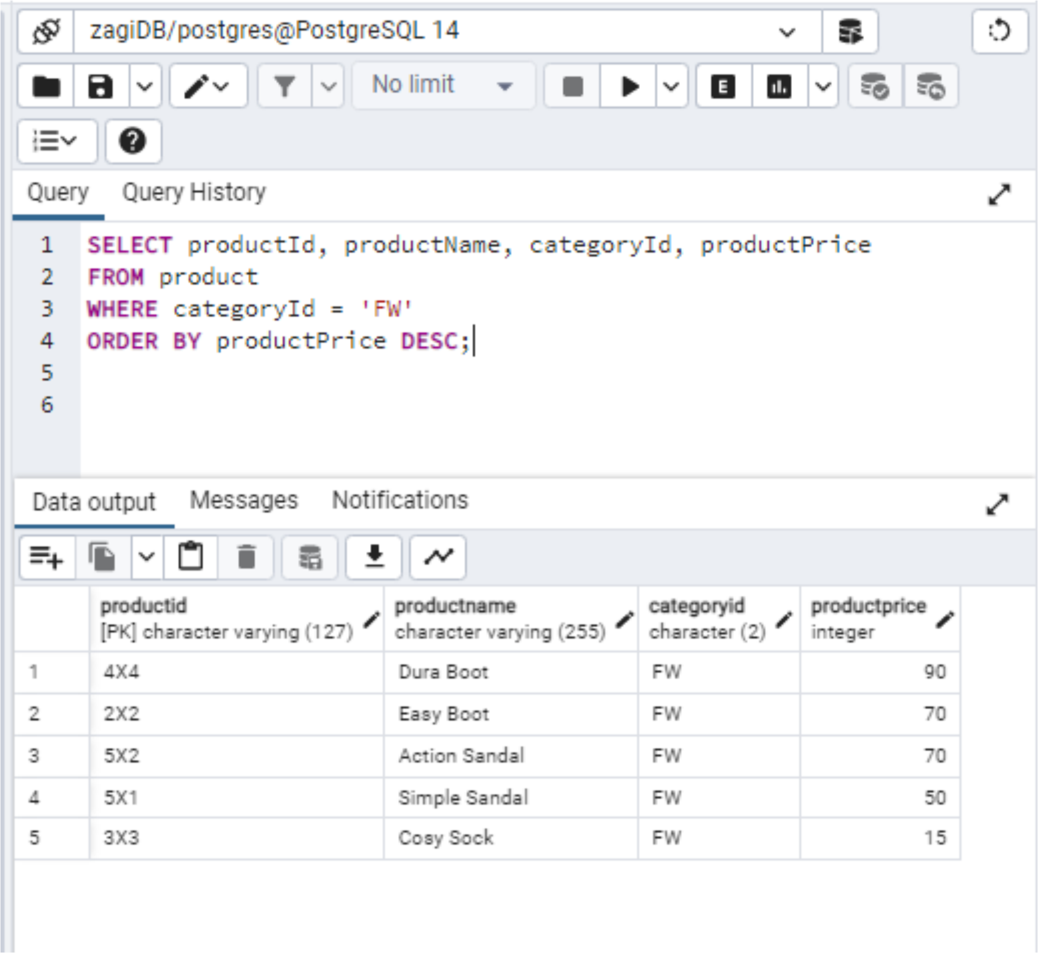
QUERY13. Retrieve the product ID, product name, category ID, and product price for each product in the FW product category, sorted by product price in descending order.11

ANS: SELECT productID, productName, categoryId, productPrice

FROM product

WHERE category = ‘FW’

ORDER BY productPrice DESC;



```sql
SELECT productId, productName, categoryId, productPrice
FROM product
WHERE categoryId = 'FW'
ORDER BY productPrice DESC;
```

	productid [PK] character varying (127)	productname character varying (255)	categoryid character (2)	productprice integer
1	4X4	Dura Boot	FW	90
2	2X2	Easy Boot	FW	70
3	5X2	Action Sandal	FW	70
4	5X1	Simple Sandal	FW	50
5	3X3	Cosy Sock	FW	15

QUERY14. For each product, retrieve the ProductID, and the total number of product items sold within all transactions. (Hint: transactions can be found in SoldVia table. NoOfItems attribute holds how many items sold in one transaction, but we need to find the total sold of a product in all transactions. See the table below for your reference).

We need the heading of the columns be exactly ‘PRODUCTID’ and ‘Total Sold’ as you can see below.

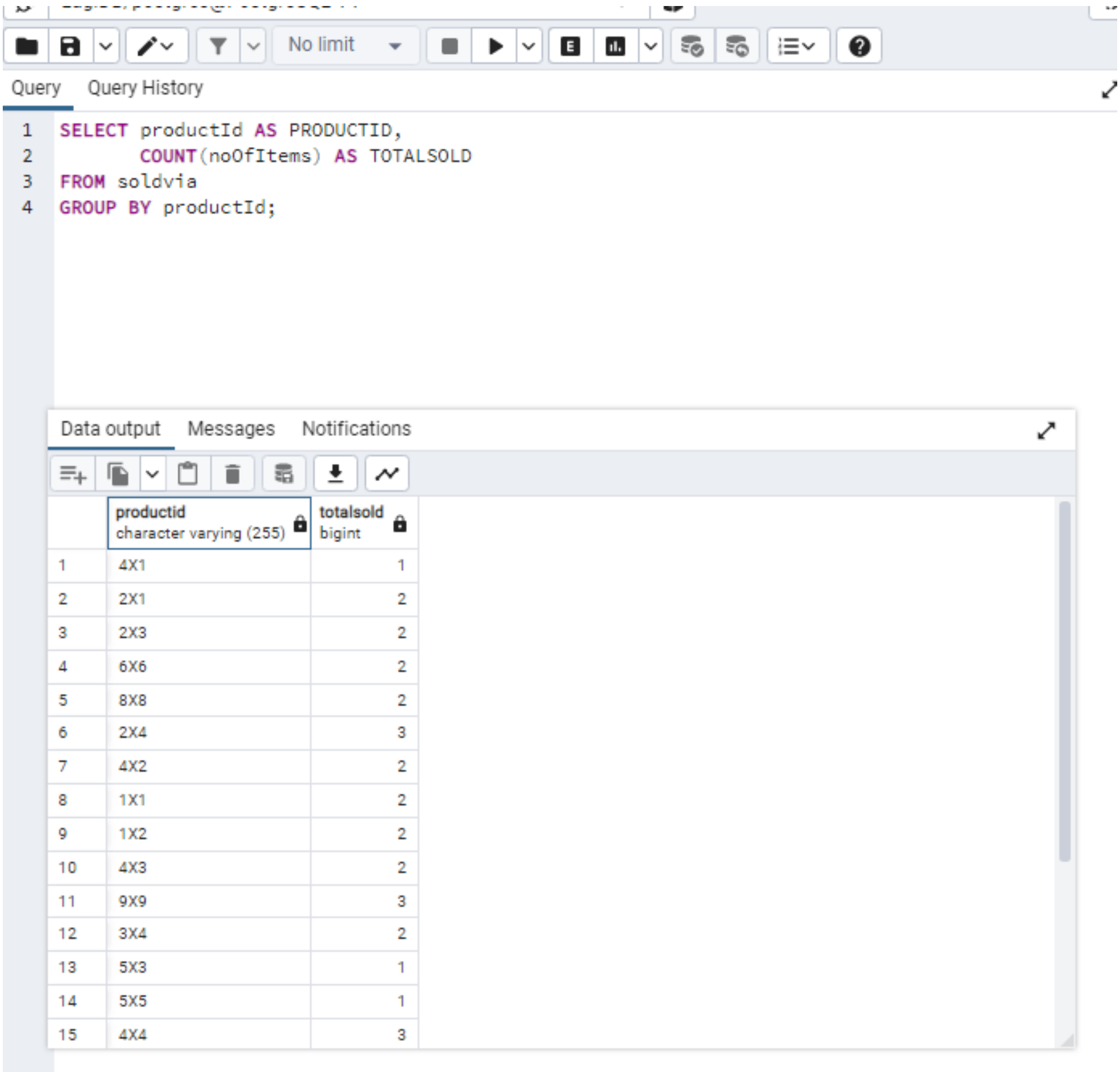
ANS:

SELECT productId AS PRODUCTID,

COUNT(noOfItems) AS TOTALSOLD

FROM soldvia

GROUP BY productId;



```
SELECT productId AS PRODUCTID,
       COUNT(noOfItems) AS TOTALSOLD
FROM soldvia
GROUP BY productId;
```

	productid character varying (255)	totalsold bigint
1	4X1	1
2	2X1	2
3	2X3	2
4	6X6	2
5	8X8	2
6	2X4	3
7	4X2	2
8	1X1	2
9	1X2	2
10	4X3	2
11	9X9	3
12	3X4	2
13	5X3	1
14	5X5	1
15	4X4	3

Query 15. For each vendor, retrieve the VendorID, number of products supplied by the vendor, and average price of the products supplied by the vendor.

ANS:

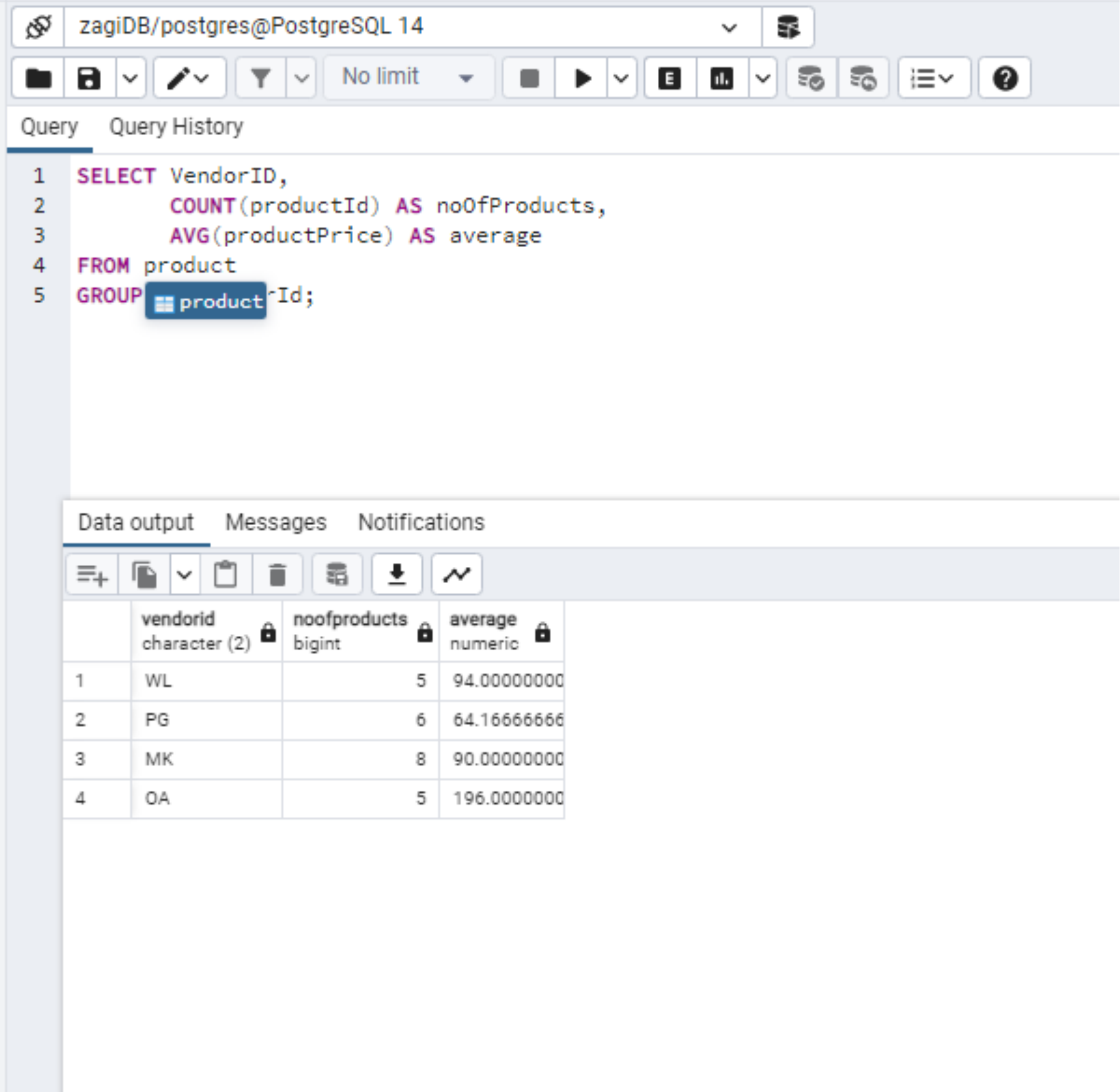
SELECT VendorID,

COUNT(productId) AS noOfProducts,

AVG(productPrice) AS average

FROM product

GROUP BY vendorId;



```
SELECT VendorID,
       COUNT(productId) AS noOfProducts,
       AVG(productPrice) AS average
FROM product
GROUP product rId;
```

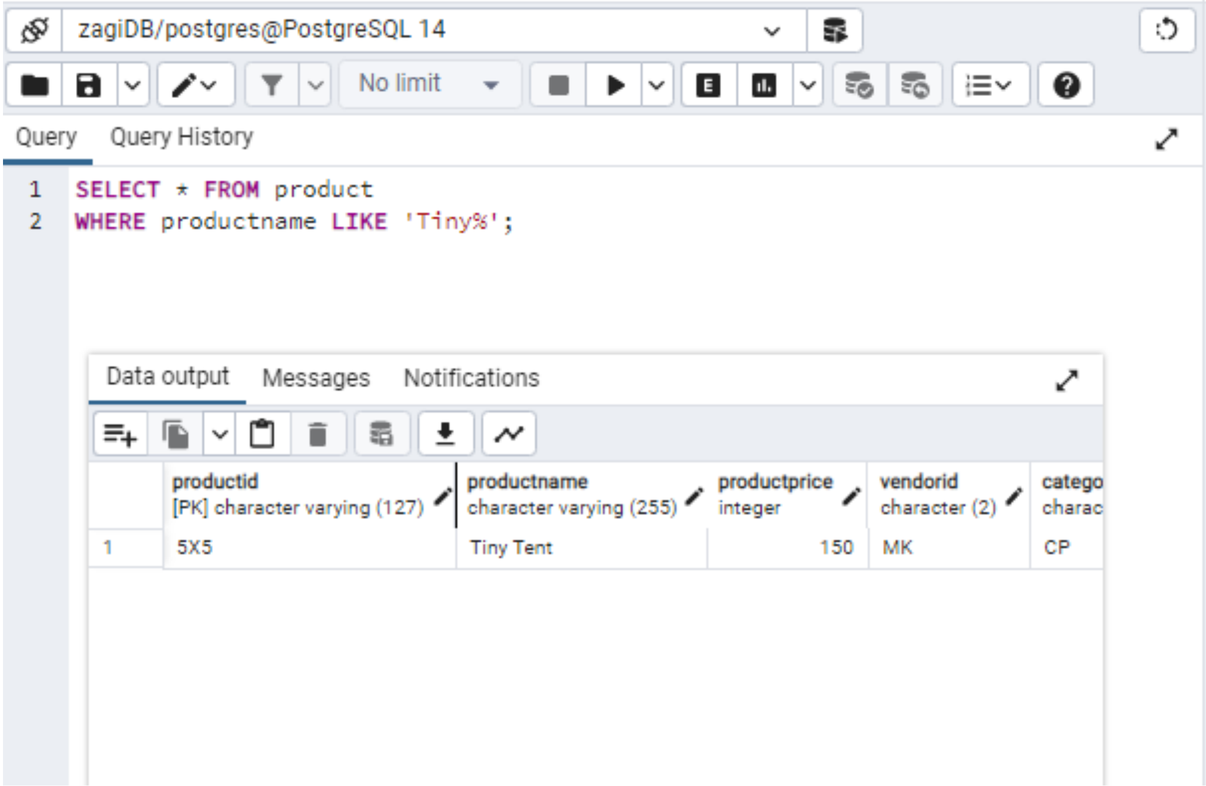
	vendorid character (2)	noofproducts bigint	average numeric
1	WL	5	94.00000000
2	PG	6	64.16666666
3	MK	8	90.00000000
4	OA	5	196.0000000

Query 16. Retrieve all attributes of products whose name starts with “Tiny”, for example, ‘Tiny Tent’

ANS:

SELECT * FROM product

WHERE productname LIKE 'Tiny%';



```
SELECT * FROM product
WHERE productname LIKE 'Tiny%';
```

	productid [PK] character varying (127)	productname character varying (255)	productprice integer	vendorid character (2)	catego charac
1	5X5	Tiny Tent	150	MK	CP

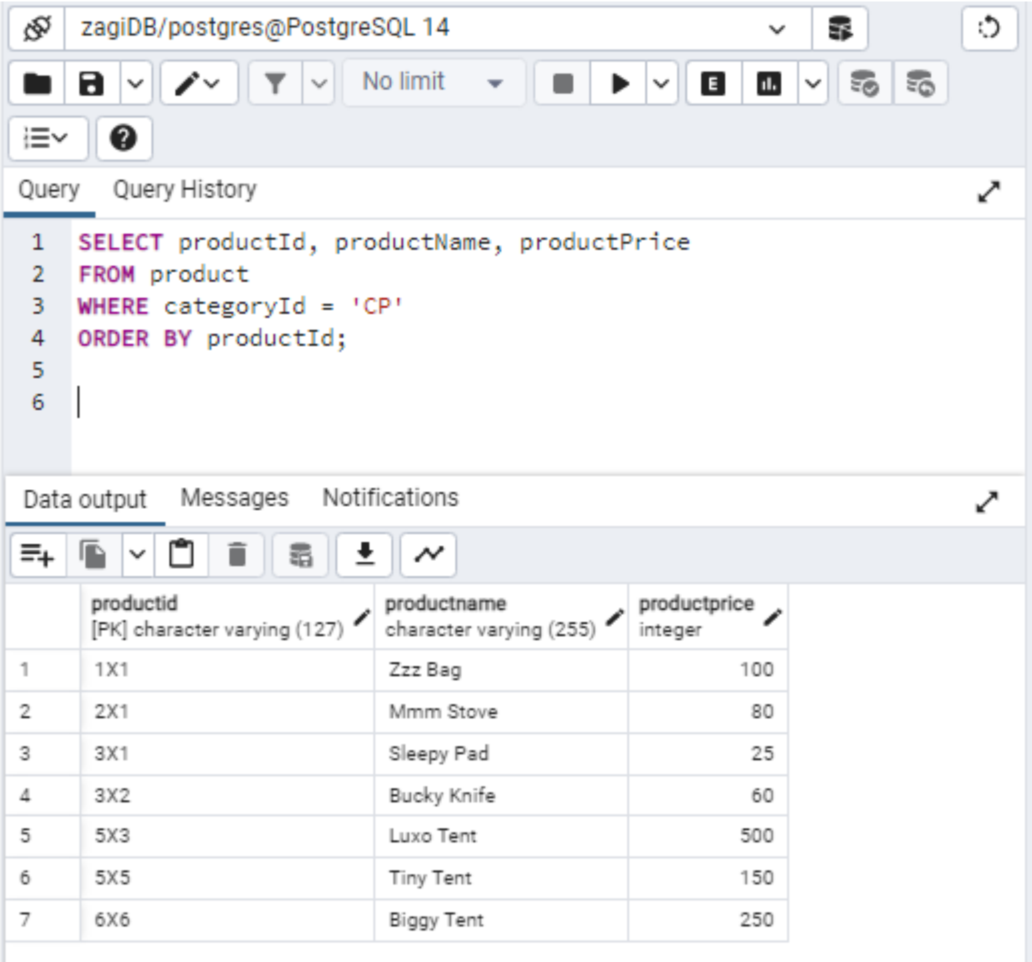
Query 17. Display the ProductID, ProductName, and ProductPrice for productsin the category whose CategoryID value is‘CP’. Sort the results by ProductID.

ANS: SELECT productID, productName, productPrice

FROM product

WHERE categoryId =’CP’

ORDER BY productId;



```
SELECT productId, productName, productPrice
FROM product
WHERE categoryId = 'CP'
ORDER BY productId;

```

	productid [PK] character varying (127)	productname character varying (255)	productprice integer
1	1X1	Zzz Bag	100
2	2X1	Mmm Stove	80
3	3X1	Sleepy Pad	25
4	3X2	Bucky Knife	60
5	5X3	Luxo Tent	500
6	5X5	Tiny Tent	150
7	6X6	Biggy Tent	250

QUERY18. Display the transaction id (TID) and the total number of items sold in that transaction (of all products) that the total number of items (of all products) sold in that transaction is greater than five. In other words, we want to get the sample result as follows.

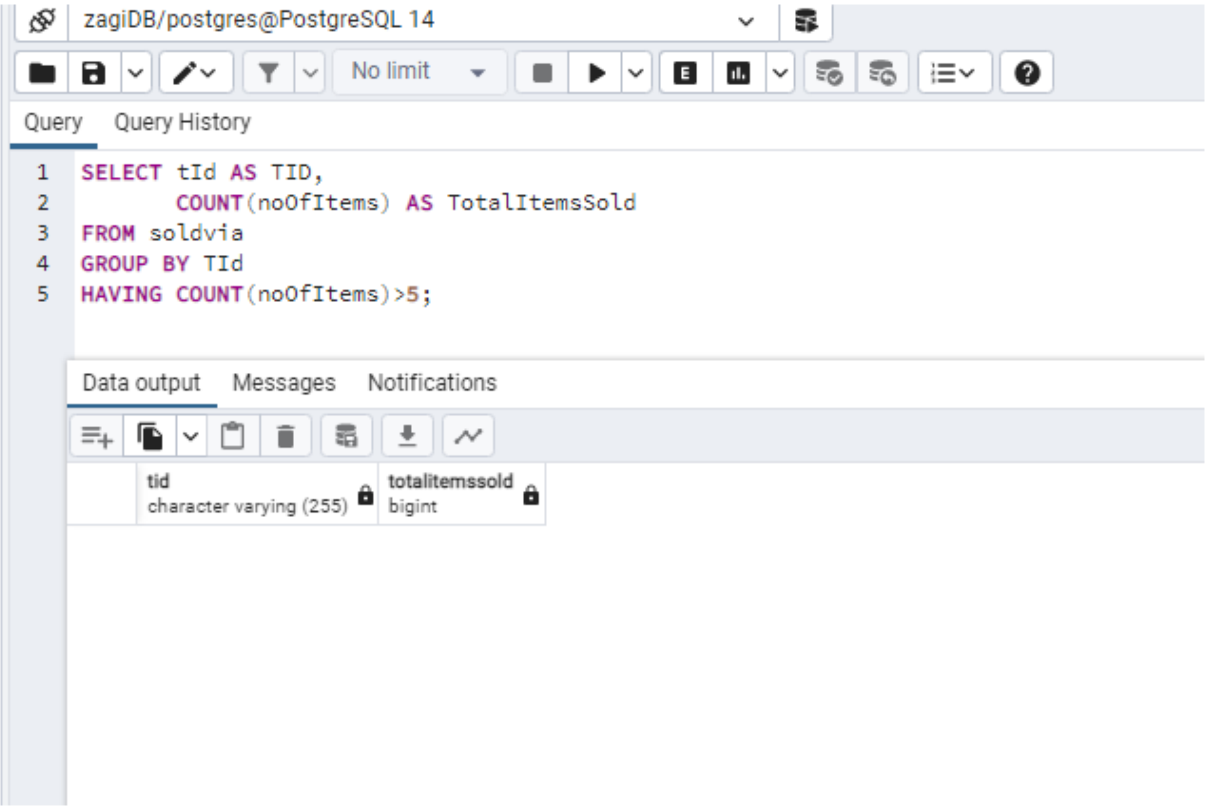
We need the heading of the columns be exactly ‘TID’ and ‘Total Items Sold’ as you can see below.

ANS:

```
SELECT tId AS TID,

        COUNT(noOfItems) AS TotalItemsSold

FROM soldvia

GROUP BY TId
```

HAVING COUNT(noOfItems)>5;

# There is no transaction in which the number of items sold are more than 5.



```
SELECT tId AS TID,
       COUNT(noOfItems) AS TotalItemsSold
FROM soldvia
GROUP BY TId
HAVING COUNT(noOfItems)>5;
```

tid character varying (255)	totalitemssold bigint

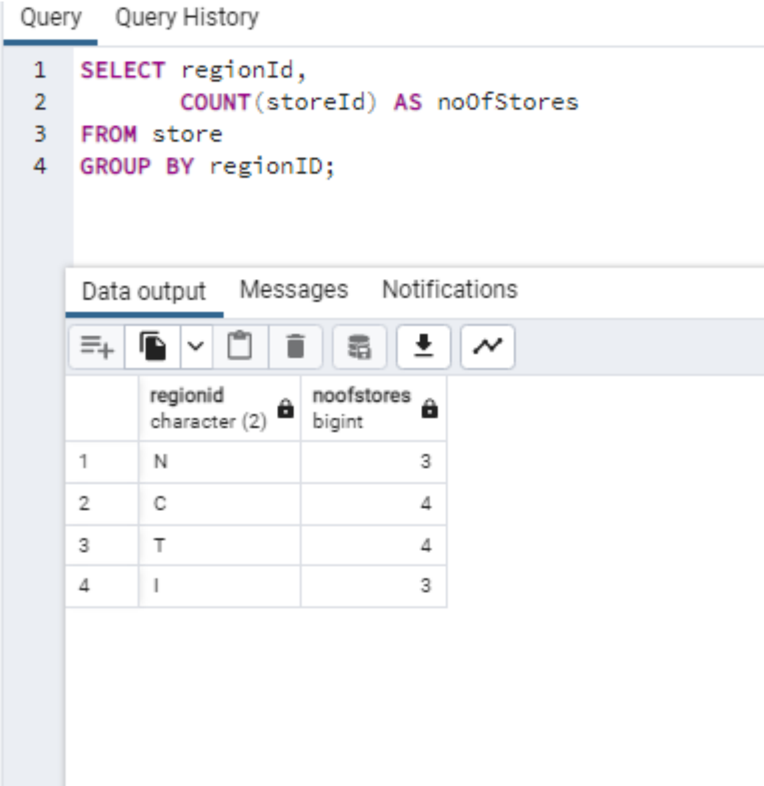
Query 19. Display all RegionIDs and number of stores in their region.

ANS: SELECT regionId,

COUNT(storeId) AS noOfStores

FROM store

GROUP BY regionID;



```
SELECT regionId,
       COUNT(storeId) AS noOfStores
FROM store
GROUP BY regionID;
```

	regionid character (2)	noofstores bigint
1	N	3
2	C	4
3	T	4
4	I	3

Query 20. Display RegionID and number of Stores in regions that number of stores in their is 4 or more.

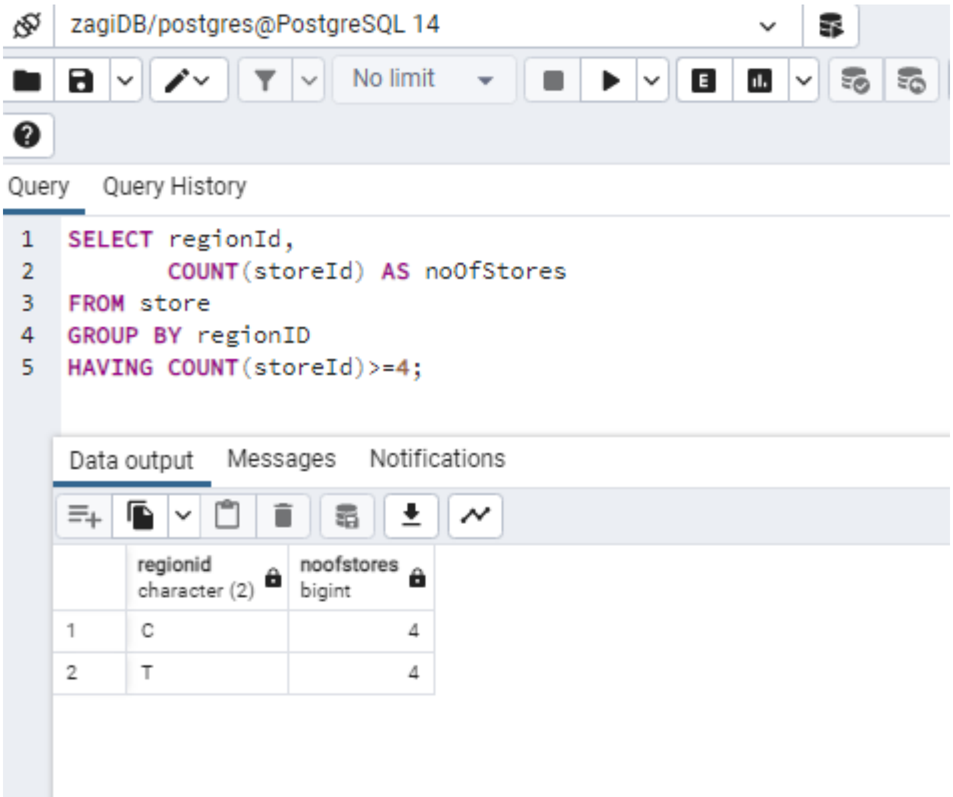
ANS: SELECT regionId,

COUNT(storeId) AS noOfStores

FROM store

GROUP BY regionID

HAVING COUNT(storeId)>=4;



```
SELECT regionId,
       COUNT(storeId) AS noOfStores
FROM store
GROUP BY regionID
HAVING COUNT(storeId)>=4;
```

	regionid character (2)	noofstores bigint
1	C	4
2	T	4