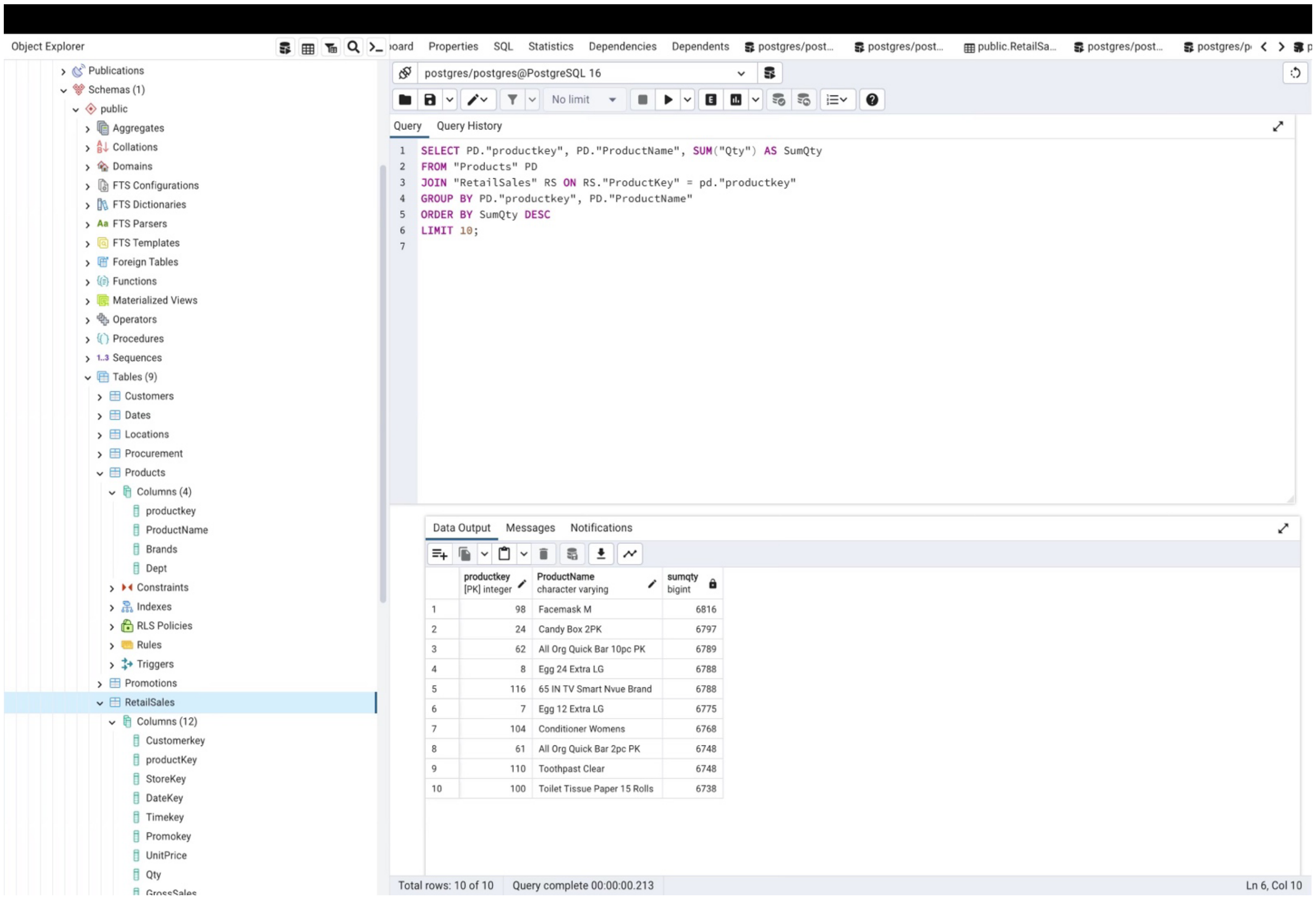
Sales Data Analysis

1. Write an SQL query to show the top 10 best-selling products by total Qty (i.e., quantity).

ANSWER:

SELECT PD."productkey", PD."ProductName", SUM("Qty") AS SumQty
FROM "Products" PD
JOIN "RetailSales" RS ON RS."ProductKey" = pd."productkey"
GROUP BY PD."productkey", PD."ProductName"
ORDER BY SumQty DESC
LIMIT 10;



```
SELECT PD."productkey", PD."ProductName", SUM("Qty") AS SumQty
FROM "Products" PD
JOIN "RetailSales" RS ON RS."ProductKey" = pd."productkey"
GROUP BY PD."productkey", PD."ProductName"
ORDER BY SumQty DESC
LIMIT 10;
```

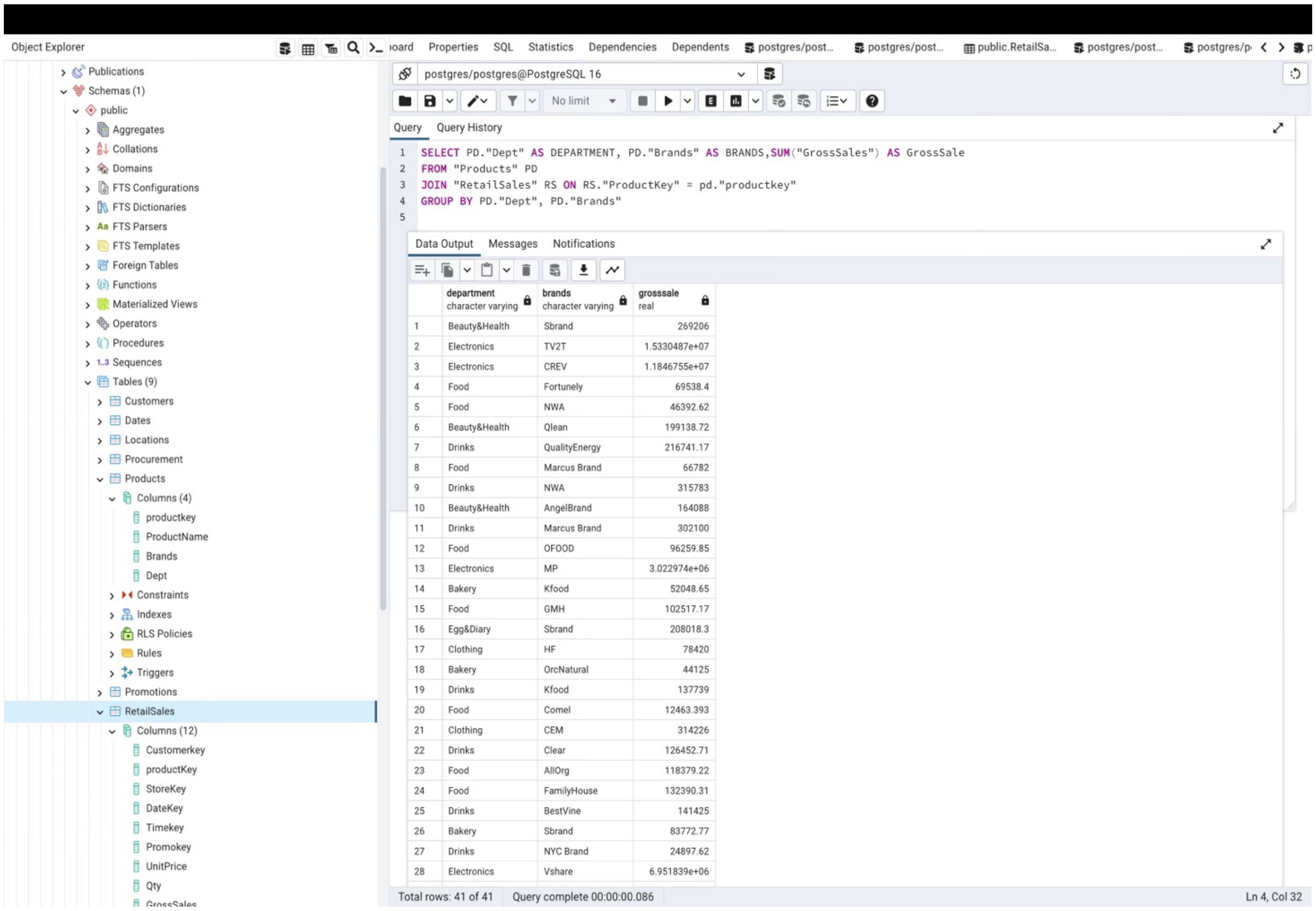
	productkey [PK] integer	ProductName character varying	sumqty bigint
1	98	Facemask M	6816
2	24	Candy Box 2PK	6797
3	62	All Org Quick Bar 10pc PK	6789
4	8	Egg 24 Extra LG	6788
5	116	65 IN TV Smart Nvue Brand	6788
6	7	Egg 12 Extra LG	6775
7	104	Conditioner Womens	6768
8	61	All Org Quick Bar 2pc PK	6748
9	110	Toothpast Clear	6748
10	100	Toilet Tissue Paper 15 Rolls	6738

Total rows: 10 of 10 Query complete 00:00:00.213 Ln 6, Col 10

2. Write an SQL query to show the total gross sales revenues each department makes by selling each brand.

ANSWER:

SELECT PD."Dept" AS DEPARTMENT, PD."Brands" AS BRANDS,SUM("GrossSales") AS GrossSale
FROM "Products" PD
JOIN "RetailSales" RS ON RS."ProductKey" = pd."productkey"
GROUP BY PD."Dept", PD."Brands"



```
SELECT PD."Dept" AS DEPARTMENT, PD."Brands" AS BRANDS,SUM("GrossSales") AS GrossSale
FROM "Products" PD
JOIN "RetailSales" RS ON RS."ProductKey" = pd."productkey"
GROUP BY PD."Dept", PD."Brands"
```

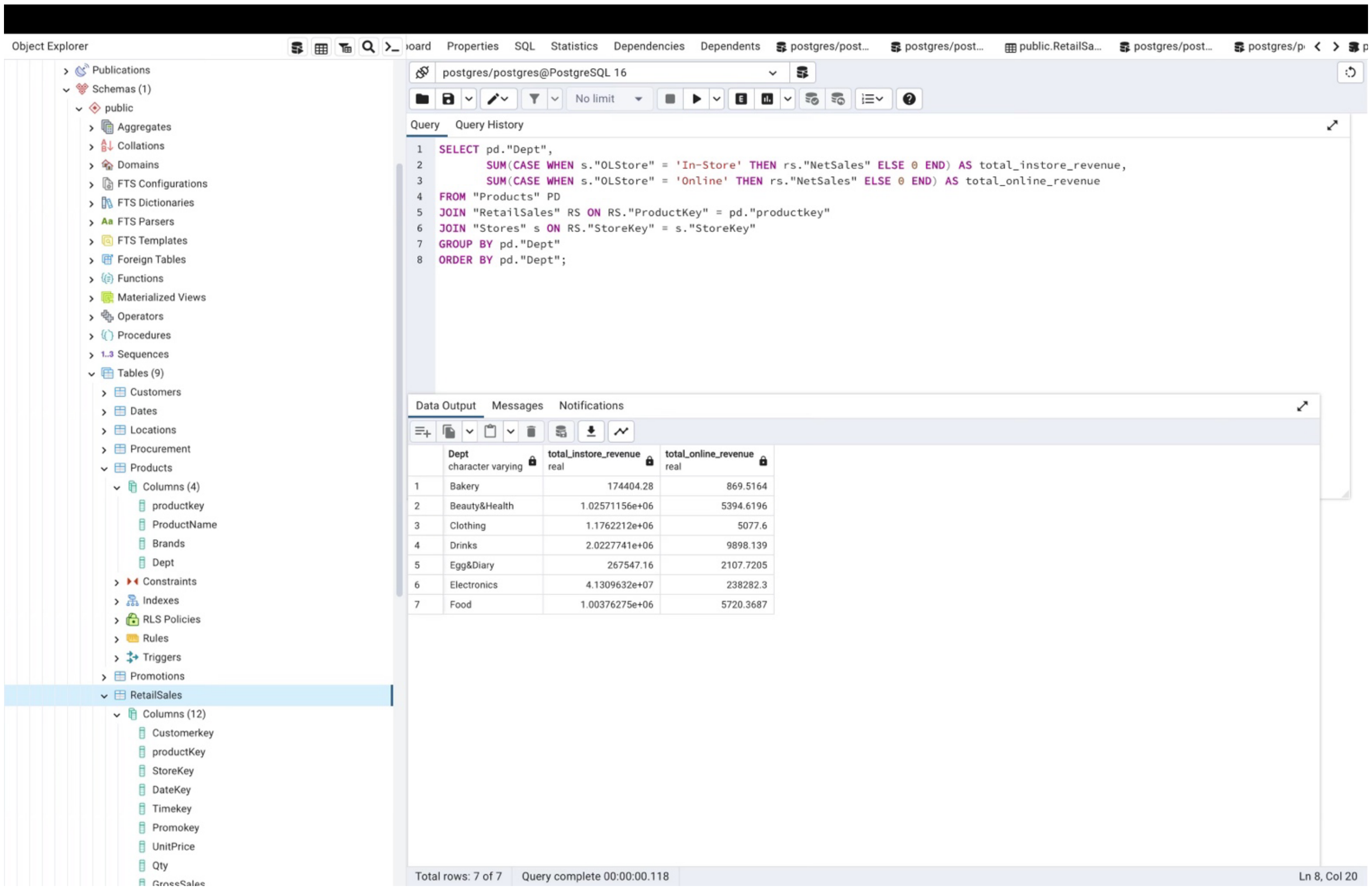
	department character varying	brands character varying	grosssale real
1	Beauty&Health	Sbrand	269206
2	Electronics	TV2T	1.5330487e+07
3	Electronics	CREV	1.1846755e+07
4	Food	Fortunely	69538.4
5	Food	NWA	46392.62
6	Beauty&Health	Qlean	199138.72
7	Drinks	QualityEnergy	216741.17
8	Food	Marcus Brand	66782
9	Drinks	NWA	315783
10	Beauty&Health	AngelBrand	164088
11	Drinks	Marcus Brand	302100
12	Food	OFOOD	96259.85
13	Electronics	MP	3.022974e+06
14	Bakery	Kfood	52048.65
15	Food	GMH	102517.17
16	Egg&Diary	Sbrand	208018.3
17	Clothing	HF	78420
18	Bakery	OrcNatural	44125
19	Drinks	Kfood	137739
20	Food	Comel	12463.393
21	Clothing	CEM	314226
22	Drinks	Clear	126452.71
23	Food	AllOrg	118379.22
24	Food	FamilyHouse	132390.31
25	Drinks	BestVine	141425
26	Bakery	Sbrand	83772.77
27	Drinks	NYC Brand	24897.62
28	Electronics	Vshare	6.951839e+06

Total rows: 41 of 41   Query complete 00:00:00.086   Ln 4, Col 32

3. Write an SQL query to show the total net sales revenues each department makes in-store and online respectively.

ANSWER:

```
SELECT pd."Dept",
       SUM(CASE WHEN s."OLStore" = 'In-Store' THEN rs."NetSales" ELSE 0 END) AS
total_instore_revenue,
       SUM(CASE WHEN s."OLStore" = 'Online' THEN rs."NetSales" ELSE 0 END) AS
total_online_revenue
FROM "Products" PD
JOIN "RetailSales" RS ON RS."ProductKey" = pd."productkey"
JOIN "Stores" s ON RS."StoreKey" = s."StoreKey"
GROUP BY pd."Dept"
ORDER BY pd."Dept";
```



```
SELECT pd."Dept",
       SUM(CASE WHEN s."OLStore" = 'In-Store' THEN rs."NetSales" ELSE 0 END) AS total_instore_revenue,
       SUM(CASE WHEN s."OLStore" = 'Online' THEN rs."NetSales" ELSE 0 END) AS total_online_revenue
FROM "Products" PD
JOIN "RetailSales" RS ON RS."ProductKey" = pd."productkey"
JOIN "Stores" s ON RS."StoreKey" = s."StoreKey"
GROUP BY pd."Dept"
ORDER BY pd."Dept";
```

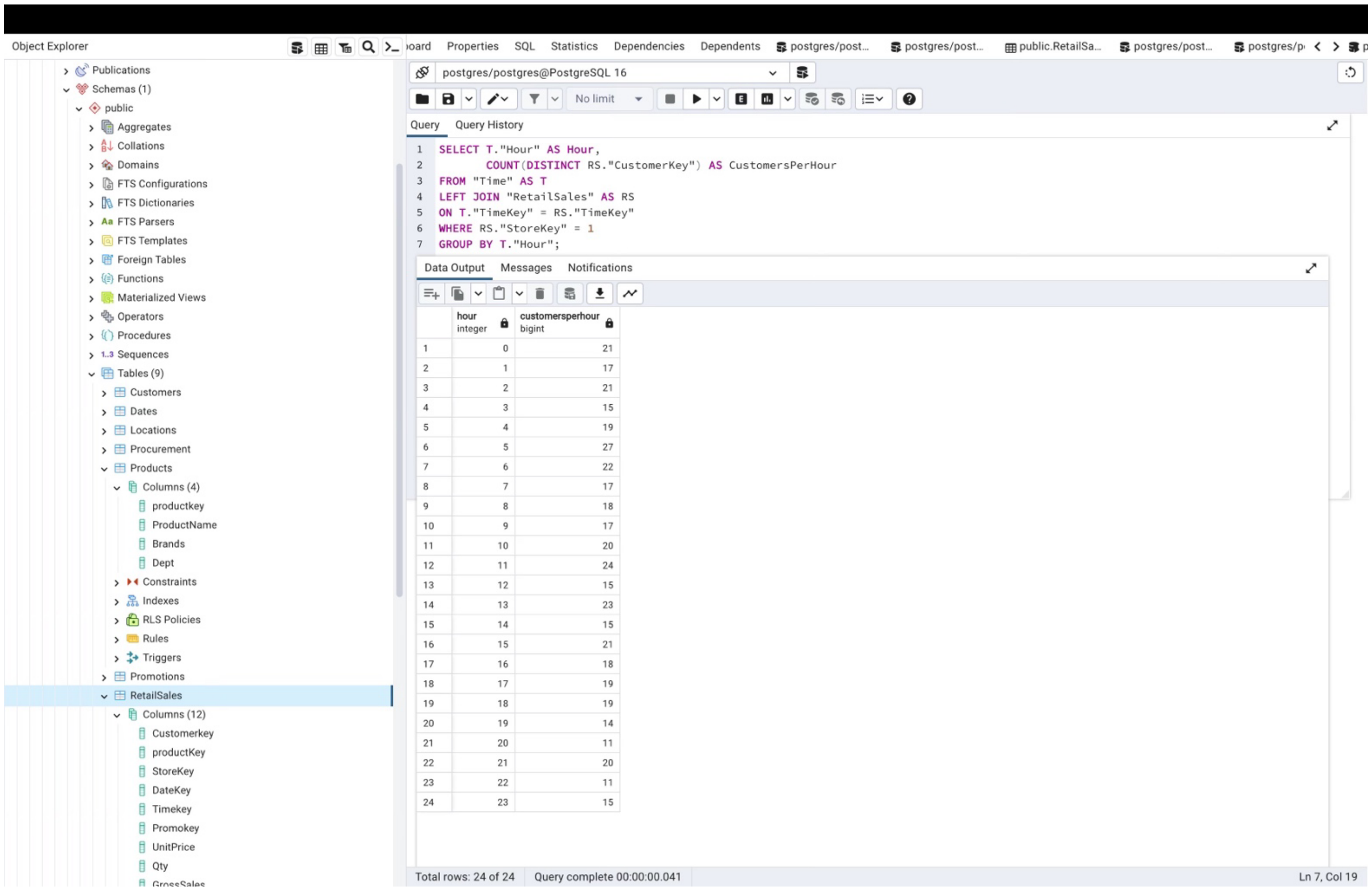
	Dept character varying	total_instore_revenue real	total_online_revenue real
1	Bakery	174404.28	869.5164
2	Beauty&Health	1.02571156e+06	5394.6196
3	Clothing	1.1762212e+06	5077.6
4	Drinks	2.0227741e+06	9898.139
5	Egg&Diary	267547.16	2107.7205
6	Electronics	4.1309632e+07	238282.3
7	Food	1.00376275e+06	5720.3687

Total rows: 7 of 7   Query complete 00:00:00.118   Ln 8, Col 20

4. Write an SQL query to show how many customers are served by all cash registers of Store #1 per hour? (Note: StoreKey should be 1 because only Store #1's cash registers are of analytical interest. The fact table records data from each scan. So, you should not use SUM(Qty) because the question asks the total number of customers per hour, not the total number of scanned products per hour. Hint: The degenerate dimension can help!)

ANSWER:

```
SELECT T."Hour" AS Hour,
      COUNT(DISTINCT RS."CustomerKey") AS CustomersPerHour
FROM "Time" AS T
LEFT JOIN "RetailSales" AS RS
ON T."TimeKey" = RS."TimeKey"
WHERE RS."StoreKey" = 1
GROUP BY T."Hour";
```



```
SELECT T."Hour" AS Hour,
       COUNT(DISTINCT RS."CustomerKey") AS CustomersPerHour
FROM "Time" AS T
LEFT JOIN "RetailSales" AS RS
ON T."TimeKey" = RS."TimeKey"
WHERE RS."StoreKey" = 1
GROUP BY T."Hour";
```

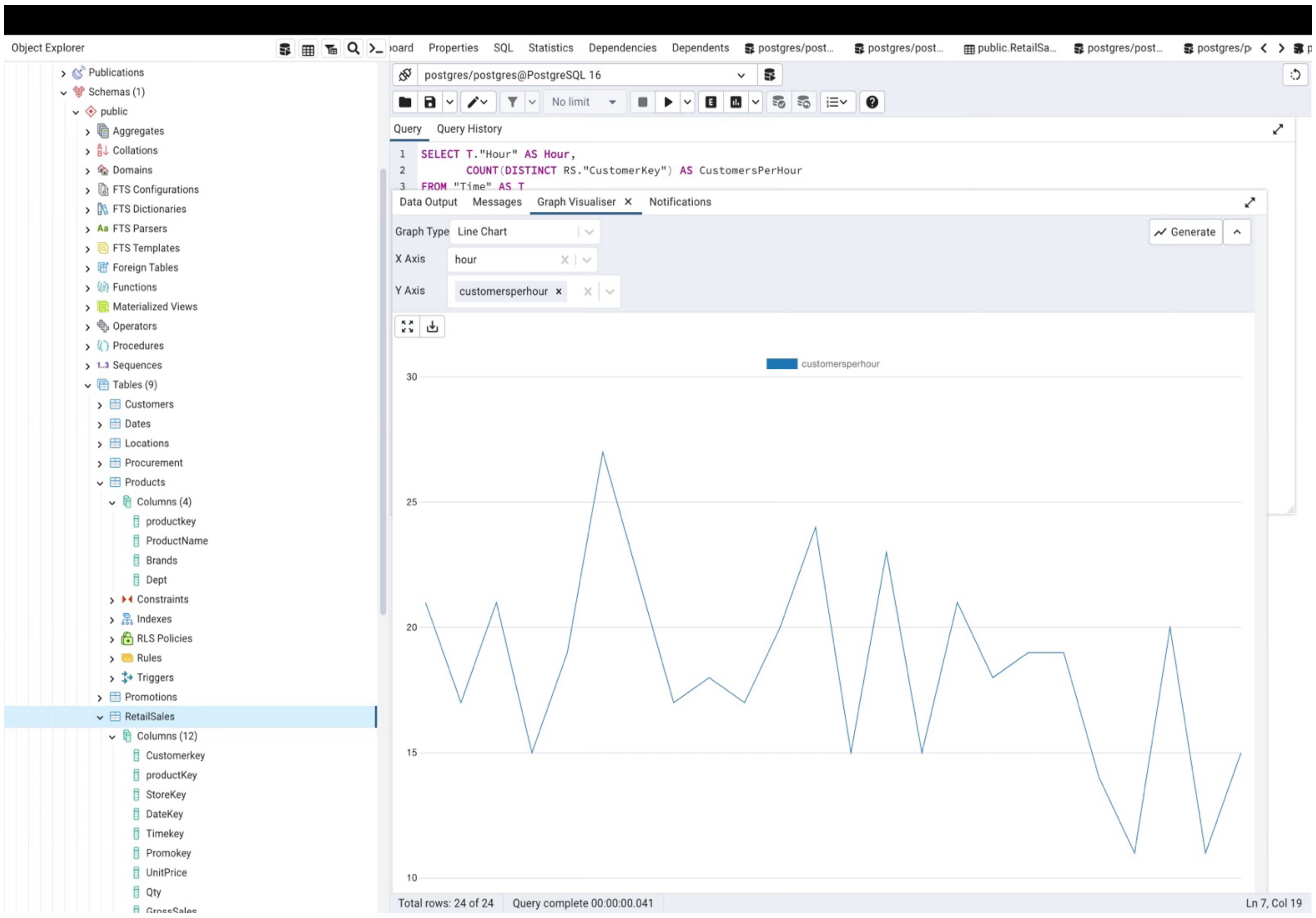
```
    hour      customersperhour
    integer   bigint
1         0        21
2         1        17
3         2        21
4         3        15
5         4        19
6         5        27
7         6        22
8         7        17
9         8        18
10        9        17
11       10        20
12       11        24
13       12        15
14       13        23
15       14        15
16       15        21
17       16        18
18       17        19
19       18        19
20       19        14
21       20        11
22       21        20
23       22        11
24       23        15
```

Total rows: 24 of 24   Query complete 00:00:00.041   Ln 7, Col 19

5. Use “Graph Visualiser” from PostgreSQL to visualize the results of question 4. Make a screenshot of the chart and save it into your submission.

ANSWER:

```
SELECT T."Hour" AS Hour,
       COUNT(DISTINCT RS."CustomerKey") AS CustomersPerHour
FROM "Time" AS T
LEFT JOIN "RetailSales" AS RS
ON T."TimeKey" = RS."TimeKey"
WHERE RS."StoreKey" = 1
GROUP BY T."Hour";
```



postgres/postgres@PostgreSQL 16

```
SELECT T."Hour" AS Hour,
       COUNT(DISTINCT RS."CustomerKey") AS CustomersPerHour
FROM "Time" AS T
```

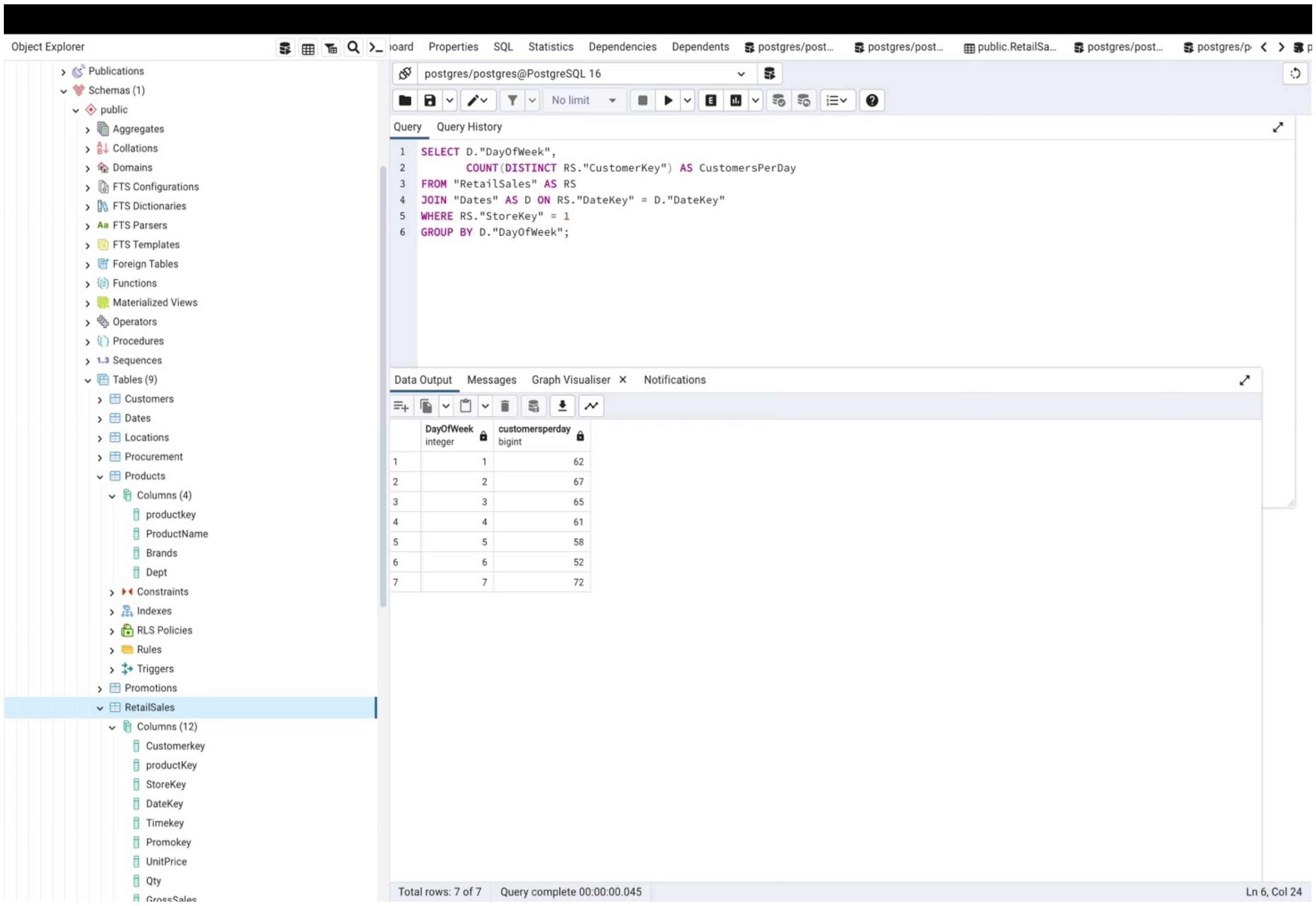
Graph Type: Line Chart
X Axis: hour
Y Axis: customersperhour

Total rows: 24 of 24   Query complete 00:00:00.041   Ln 7, Col 19

6. Write an SQL query to show how many customers are served by Store #1 in each day of a week. (Note: The SQL query should show the total number of customers served in Store #1 on Monday, Tuesday, … Sunday. The number of customers and week day information must be in the same table. You should not write a query for Monday, and then another query for Tuesday, and then another query for Wednesday, etc. Using just one query to show the number of customers served by Store #1 in each day of a week is required).

ANSWER:

```
SELECT D."DayOfWeek",
       COUNT(DISTINCT RS."CustomerKey") AS CustomersPerDay
FROM "RetailSales" AS RS
JOIN "Dates" AS D ON RS."DateKey" = D."DateKey"
WHERE RS."StoreKey" = 1
GROUP BY D."DayOfWeek";
```



```
SELECT D."DayOfWeek",
       COUNT(DISTINCT RS."CustomerKey") AS CustomersPerDay
FROM "RetailSales" AS RS
JOIN "Dates" AS D ON RS."DateKey" = D."DateKey"
WHERE RS."StoreKey" = 1
GROUP BY D."DayOfWeek";
```

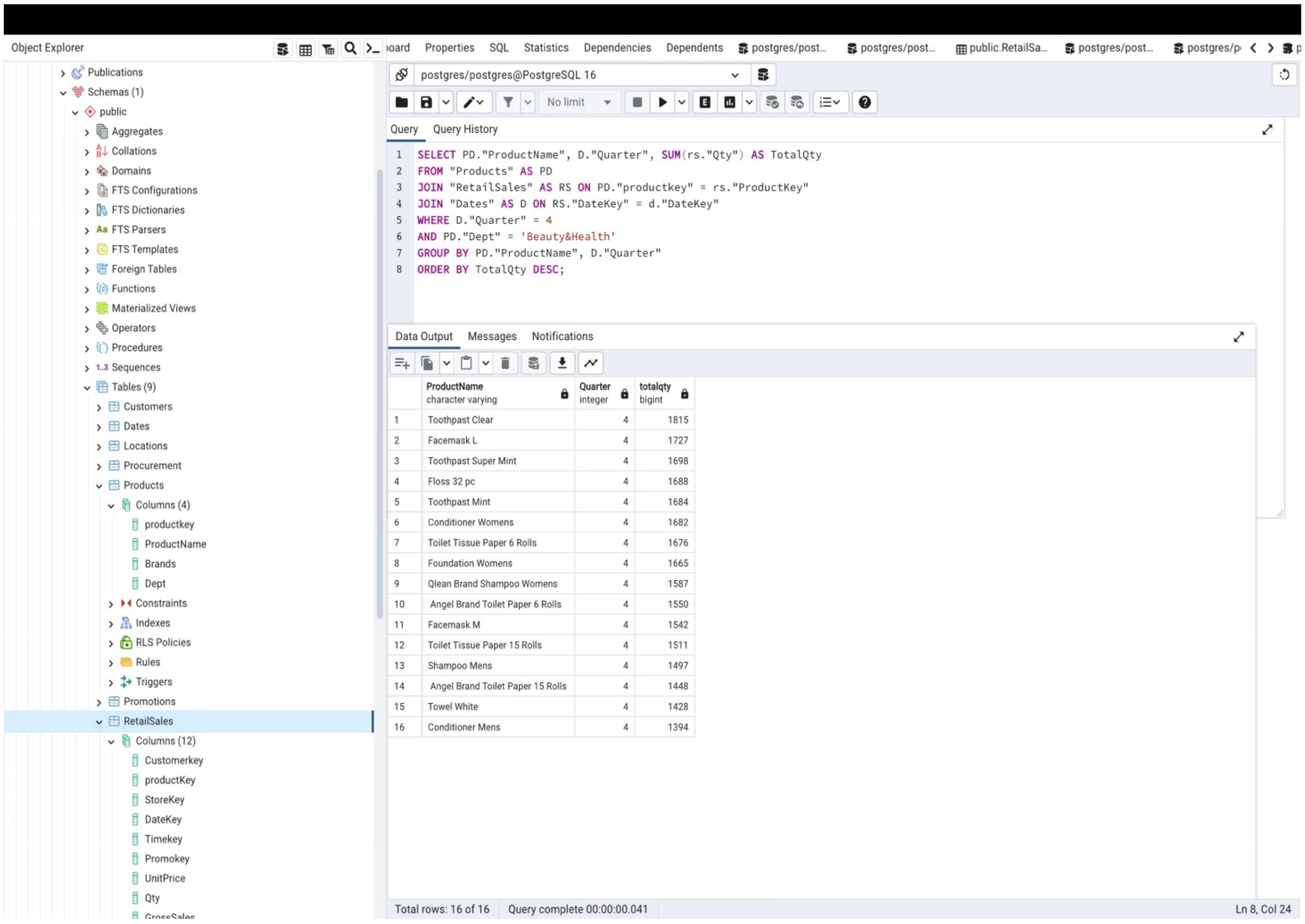
	DayOfWeek integer	customersperday bigint
1	1	62
2	2	67
3	3	65
4	4	61
5	5	58
6	6	52
7	7	72

Total rows: 7 of 7 Query complete 00:00:00.045 Ln 6, Col 24

7. Write an SQL query to show what are the best-selling products by total Qty in the fourth quarter in the “Beauty&Health” department.

ANSWER:

```
SELECT PD."ProductName", D."Quarter", SUM(rs."Qty") AS TotalQty
FROM "Products" AS PD
JOIN "RetailSales" AS RS ON PD."productkey" = rs."ProductKey"
JOIN "Dates" AS D ON RS."DateKey" = d."DateKey"
WHERE D."Quarter" = 4
AND PD."Dept" = 'Beauty&Health'
GROUP BY PD."ProductName", D."Quarter"
ORDER BY TotalQty DESC;
```



```
SELECT PD."ProductName", D."Quarter", SUM(rs."Qty") AS TotalQty
FROM "Products" AS PD
JOIN "RetailSales" AS RS ON PD."productkey" = rs."ProductKey"
JOIN "Dates" AS D ON RS."DateKey" = d."DateKey"
WHERE D."Quarter" = 4
AND PD."Dept" = 'Beauty&Health'
GROUP BY PD."ProductName", D."Quarter"
ORDER BY TotalQty DESC;
```

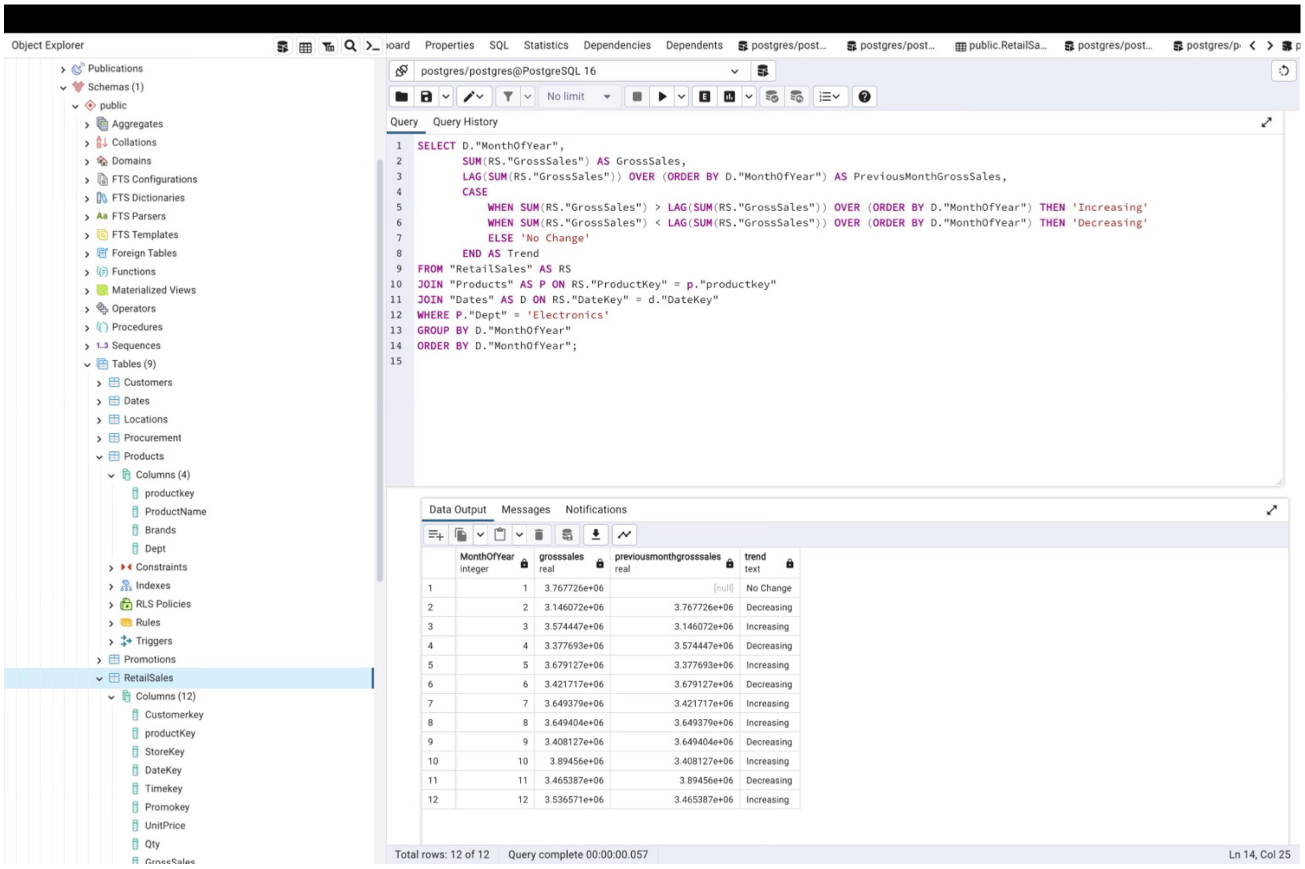
	ProductName character varying	Quarter integer	totalqty bigint
1	Toothpast Clear	4	1815
2	Facemask L	4	1727
3	Toothpast Super Mint	4	1698
4	Floss 32 pc	4	1688
5	Toothpast Mint	4	1684
6	Conditioner Womens	4	1682
7	Toilet Tissue Paper 6 Rolls	4	1676
8	Foundation Womens	4	1665
9	Qlean Brand Shampoo Womens	4	1587
10	Angel Brand Toilet Paper 6 Rolls	4	1550
11	Facemask M	4	1542
12	Toilet Tissue Paper 15 Rolls	4	1511
13	Shampoo Mens	4	1497
14	Angel Brand Toilet Paper 15 Rolls	4	1448
15	Towel White	4	1428
16	Conditioner Mens	4	1394

Total rows: 16 of 16 Query complete 00:00:00.041 Ln 8, Col 24

8. Write an SQL query to show whether there is a monthly increasing total GrossSales trend in the “Electronics” department.

ANSWER:

```
SELECT D."MonthOfYear",
      SUM(RS."GrossSales") AS GrossSales,
      LAG(SUM(RS."GrossSales")) OVER (ORDER BY D."MonthOfYear") AS
PreviousMonthGrossSales,
      CASE
          WHEN SUM(RS."GrossSales") > LAG(SUM(RS."GrossSales")) OVER (ORDER BY
D."MonthOfYear") THEN 'Increasing'
          WHEN SUM(RS."GrossSales") < LAG(SUM(RS."GrossSales")) OVER (ORDER BY
D."MonthOfYear") THEN 'Decreasing'
          ELSE 'No Change'
      END AS Trend
FROM "RetailSales" AS RS
JOIN "Products" AS P ON RS."ProductKey" = p."productkey"
JOIN "Dates" AS D ON RS."DateKey" = d."DateKey"
WHERE P."Dept" = 'Electronics'
GROUP BY D."MonthOfYear"
ORDER BY D."MonthOfYear";
```



postgres/postgres@PostgreSQL 16

```sql
SELECT D."MonthOfYear",
       SUM(RS."GrossSales") AS GrossSales,
       LAG(SUM(RS."GrossSales")) OVER (ORDER BY D."MonthOfYear") AS PreviousMonthGrossSales,
       CASE
           WHEN SUM(RS."GrossSales") > LAG(SUM(RS."GrossSales")) OVER (ORDER BY D."MonthOfYear") THEN 'Increasing'
           WHEN SUM(RS."GrossSales") < LAG(SUM(RS."GrossSales")) OVER (ORDER BY D."MonthOfYear") THEN 'Decreasing'
           ELSE 'No Change'
       END AS Trend
FROM "RetailSales" AS RS
JOIN "Products" AS P ON RS."ProductKey" = p."productkey"
JOIN "Dates" AS D ON RS."DateKey" = d."DateKey"
WHERE P."Dept" = 'Electronics'
GROUP BY D."MonthOfYear"
ORDER BY D."MonthOfYear";
```

	MonthOfYear integer	grosssales real	previousmonthgrosssales real	trend text
1	1	3.767726e+06	[null]	No Change
2	2	3.146072e+06	3.767726e+06	Decreasing
3	3	3.574447e+06	3.146072e+06	Increasing
4	4	3.377693e+06	3.574447e+06	Decreasing
5	5	3.679127e+06	3.377693e+06	Increasing
6	6	3.421717e+06	3.679127e+06	Decreasing
7	7	3.649379e+06	3.421717e+06	Increasing
8	8	3.649404e+06	3.649379e+06	Increasing
9	9	3.408127e+06	3.649404e+06	Decreasing
10	10	3.89456e+06	3.408127e+06	Increasing
11	11	3.465387e+06	3.89456e+06	Decreasing
12	12	3.536571e+06	3.465387e+06	Increasing

Total rows: 12 of 12 Query complete 00:00:00.057 Ln 14, Col 25