Build a Data Pipeline for Processing and Storing Sales Data

THE SITUATION

Created a data pipeline to process and store sales data from a CSV file. The pipeline will transform and load the data with Python into a PostgreSQL and MySQL database for further analysis using Power BI.
Tools required: Python (Pandas), PostgreSQL, MySQL, DBdiagram.io and Power BI

THE STEPS

Used DBdiagram.io:

- To design a dimensional model for MavenMarket Database.

Used MySQL and PostgreSQL to:

- Create a database title MavenMarket.
- Define the database schema under the database (MavenMarket):
  Product,
  region,
  store,
  return product.

Used Python to:

- Use pandas library to load the sales data from the CSV file into pandas dataframe.
- Use pandas to clean and transform the data as needed. For instance, drop some columns that are not needed in product and store table.
- Use the psycopg2 and mysql library to connect to the PostgreSQL and MySQL database.
- Use pandas, psycopg2, and mysql to load the cleaned and transformed data into the database.
- Runed some test queries on the data in the database to ensure the data pipeline is working correctly.

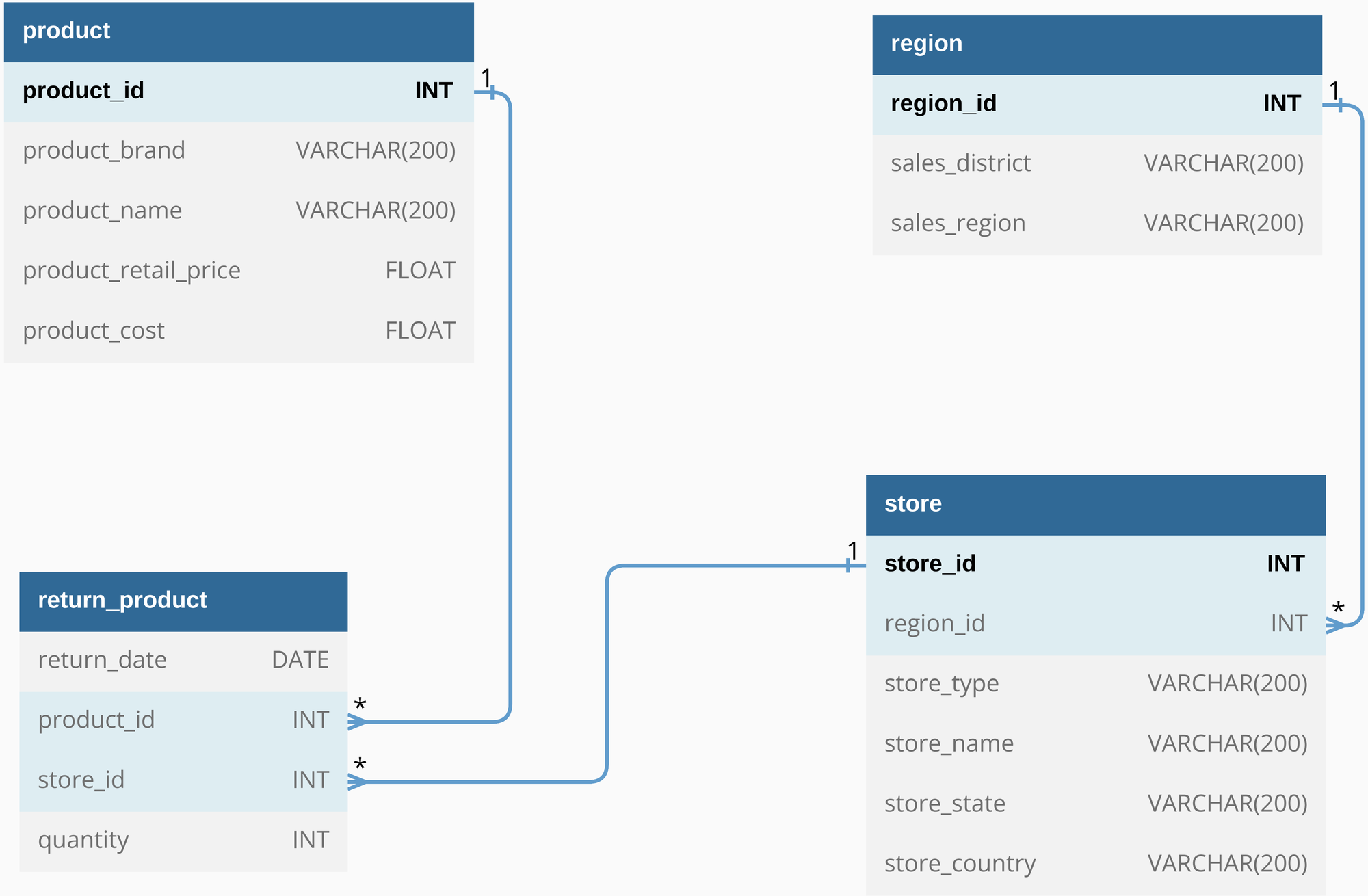
Used PowerBI to:

- To connect the raw data
- Build a relational data model
- Create new calculated columns and DAX measures
- Design an interactive report to analyze and visualize the data

THE SUMMARY

After performing data wrangling, data modelling and data visualization. Here, are the following insights from the dataset.

- Portland reached 1,000 sales in December to close out the year
- High Top product returns doubled in Mexico (4 to 8), at a return rate of 1.2%
- Plato products drove the strongest overall profit margin (63.55%) in 1998.


product
product_id INT
product_brand VARCHAR(200)
product_name VARCHAR(200)
product_retail_price FLOAT
product_cost FLOAT
region
region_id INT
sales_district VARCHAR(200)
sales_region VARCHAR(200)
store
store_id INT
region_id INT
store_type VARCHAR(200)
store_name VARCHAR(200)
store_state VARCHAR(200)
store_country VARCHAR(200)
return_product
return_date DATE
product_id INT
store_id INT
quantity INT
1
*
1
*
1
*
dbdiagram.io

SQL SCRIPTS FOR MAVENMARKET

```
CREATE TABLE product (
  product_id INT PRIMARY KEY,
  product_brand VARCHAR(200),
  product_name VARCHAR(200),
  product_retail_price FLOAT,
  product_cost FLOAT
);
CREATE TABLE region (
  region_id INT PRIMARY KEY,
  sales_district VARCHAR(200),
  sales_region VARCHAR(200)
);
CREATE TABLE return_product (
  return_date DATE,
  product_id INT,
  store_id INT,
  quantity INT,
  FOREIGN KEY (product_id)
    REFERENCES product (product_id),
  FOREIGN KEY (store_id)
    REFERENCES store (store_id)
);
CREATE TABLE store (
  store_id INT PRIMARY KEY,
  region_id INT,
  store_type VARCHAR(200),
  store_name VARCHAR(200),
  store_state VARCHAR(200),
  store_country VARCHAR(200),
  FOREIGN KEY (region_id)
    REFERENCES region (region_id) );
```

Read Dataset

In [1]:
```
import pandas as pd
```

In [2]:
```
# Read Products data
MavenMarket_Products = pd.read_csv(r'C:\Users\Lenovo\Desktop\Personal Documents\DataSet\Maven+Market+CSV+Files\MavenMarket_Products.csv',delimiter = ",")
MavenMarket_Products.head(5)
```

Out[2]:

	product_id	product_brand	product_name	product_sku	product_retail_price	product_cost	product_weight	recyclable	low_fat
0	1	Washington	Washington Berry Juice	90748583674	2.85	0.94	8.39	NaN	NaN
1	2	Washington	Washington Mango Drink	96516502499	0.74	0.26	7.42	NaN	1.0
2	3	Washington	Washington Strawberry Drink	58427771925	0.83	0.40	13.10	1.0	1.0
3	4	Washington	Washington Cream Soda	64412155747	3.64	1.64	10.60	1.0	NaN
4	5	Washington	Washington Diet Soda	85561191439	2.19	0.77	6.66	1.0	NaN

In [3]:
```
# Read Regions data
Regions = pd.read_csv(r'C:\Users\Lenovo\Desktop\Personal Documents\DataSet\Maven+Market+CSV+Files\MavenMarket_Regions.csv',delimiter = ",")
Regions.head(5)
```

Out[3]:

	region_id	sales_district	sales_region
0	1	San Francisco	Central West
1	2	Mexico City	Mexico Central
2	3	Los Angeles	South West
3	4	Guadalajara	Mexico West
4	5	Vancouver	Canada West

In [4]:
```
# Read MavenMarket_Returns_1997-1998 data
Returns = pd.read_csv(r'C:\Users\Lenovo\Desktop\Personal Documents\DataSet\Maven+Market+CSV+Files\MavenMarket_Returns_1997-1998.csv',delimiter = ",", parse_dates = ["return_date"])
Returns.head(5)
```

Out[4]:

	return_date	product_id	store_id	quantity
0	1997-01-01	250	6	1
1	1997-01-01	628	6	1
2	1997-01-01	869	6	1
3	1997-01-02	469	11	1
4	1997-01-02	532	23	2

In [5]:
```
# Read MavenMarket_Stores_1997-1998 data
MavenMarket_Stores = pd.read_csv(r'C:\Users\Lenovo\Desktop\Personal Documents\DataSet\Maven+Market+CSV+Files\MavenMarket_Stores.csv',delimiter = ",")
MavenMarket_Stores.head(5)
```

Out[5]:

	store_id	region_id	store_type	store_name	store_street_address	store_city	store_state	store_country	store_phone	first_opened_date	last_remodel_date	total_sqft	grocery_sqft
0	1	28	Supermarket	Store 1	2853 Bailey Rd	Acapulco	Guerrero	Mexico	262-555-5124	1/9/1982	12/5/1990	23593	17475
1	2	78	Small Grocery	Store 2	5203 Catanzaro Way	Bellingham	WA	USA	605-555-8203	4/2/1970	6/4/1973	28206	22271
2	3	76	Supermarket	Store 3	1501 Ramsey Circle	Bremerton	WA	USA	509-555-1596	6/14/1959	11/19/1967	39696	24390
3	4	27	Gourmet Supermarket	Store 4	433 St George Dr	Camacho	Zacatecas	Mexico	304-555-1474	9/27/1994	12/1/1995	23759	16844
4	5	4	Small Grocery	Store 5	1250 Coggins Drive	Guadalajara	Jalisco	Mexico	801-555-4324	9/18/1978	6/29/1991	24597	15012

Data Wranging

In [6]:
```
#drop some columns
Product = MavenMarket_Products.drop(["product_sku", "product_weight", "recyclable", "low_fat"], axis = 1)
Product.head(5)
```

Out[6]:

	product_id	product_brand	product_name	product_retail_price	product_cost
0	1	Washington	Washington Berry Juice	2.85	0.94
1	2	Washington	Washington Mango Drink	0.74	0.26
2	3	Washington	Washington Strawberry Drink	0.83	0.40
3	4	Washington	Washington Cream Soda	3.64	1.64
4	5	Washington	Washington Diet Soda	2.19	0.77

In [7]:
```
Store = MavenMarket_Stores.drop(["store_phone","first_opened_date","last_remodel_date","total_sqft","grocery_sqft", "store_street_address","store_city"], axis = 1)
Store.head(5)
```

Out[7]:

	store_id	region_id	store_type	store_name	store_state	store_country
0	1	28	Supermarket	Store 1	Guerrero	Mexico
1	2	78	Small Grocery	Store 2	WA	USA
2	3	76	Supermarket	Store 3	WA	USA
3	4	27	Gourmet Supermarket	Store 4	Zacatecas	Mexico
4	5	4	Small Grocery	Store 5	Jalisco	Mexico

Load Data in MySQL

In [8]:
```
pip install sqlalchemy
```

```
Requirement already satisfied: sqlalchemy in c:\users\lenovo\anaconda3\lib\site-packages (1.4.32)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\lenovo\anaconda3\lib\site-packages (from sqlalchemy) (1.1.1)
Note: you may need to restart the kernel to use updated packages.
```

In [9]:
```
import sqlalchemy
```

In [10]:
```
pip install mysql
```

```
Requirement already satisfied: mysql in c:\users\lenovo\anaconda3\lib\site-packages (0.0.3)
Requirement already satisfied: mysqlclient in c:\users\lenovo\anaconda3\lib\site-packages (from mysql) (2.1.1)
Note: you may need to restart the kernel to use updated packages.
```

In [11]:
```
engine1 = sqlalchemy.create_engine("mysql://sammy:password@localhost/mavenmarket")
```

In [20]:
```
Product.to_sql(name = 'product', con = engine1, index = False, if_exists = "replace")
```

Out[20]:
```
1560
```

In [19]:
```
Regions.to_sql(name = 'region', con = engine1, index = False, if_exists = 'replace')
```

Out[19]:
```
109
```

In [18]:
```
Store.to_sql(name = 'store', con = engine1, index = False, if_exists = 'replace')
```

Out[18]:
```
24
```

In [17]:
```
Returns.to_sql(name = 'return_product', con = engine1, index = False, if_exists = 'replace')
```

Out[17]:
```
7087
```

Load Data in PostgreSQL

In [21]:
```
engine = sqlalchemy.create_engine("postgresql+psycopg2://Uche:diamond@localhost/MavenMarket")
```

In [26]:
```
Product.to_sql(name = 'product', con = engine, index = False, if_exists = 'replace')
```

Out[26]:
```
560
```

In [25]:
```
Regions.to_sql(name = 'region', con = engine, index = False, if_exists = 'replace')
```

Out[25]:
```
109
```

In [24]:
```
Store.to_sql(name = 'store', con = engine, index = False, if_exists = 'replace')
```

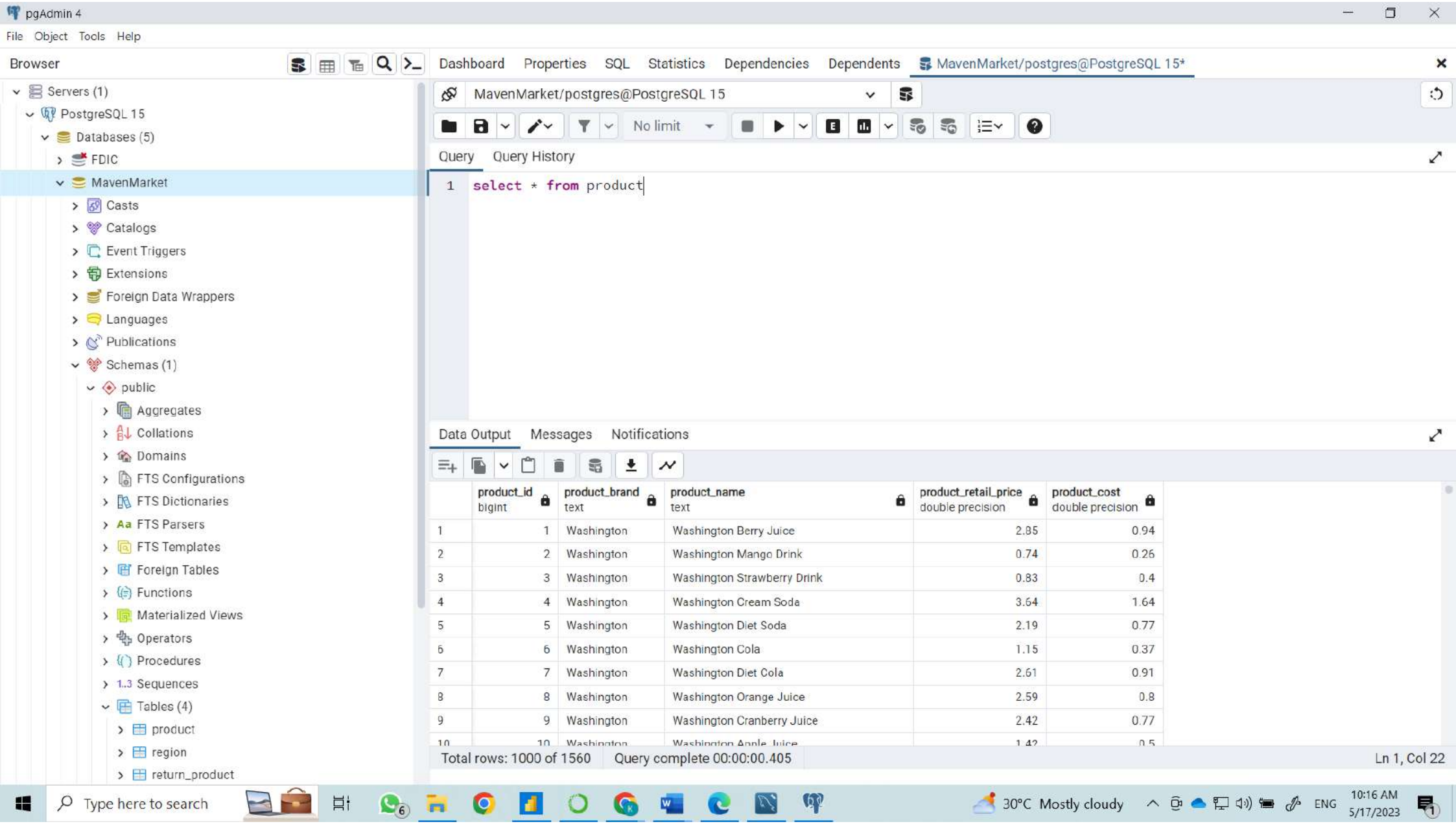
Out[24]:
```
24
```

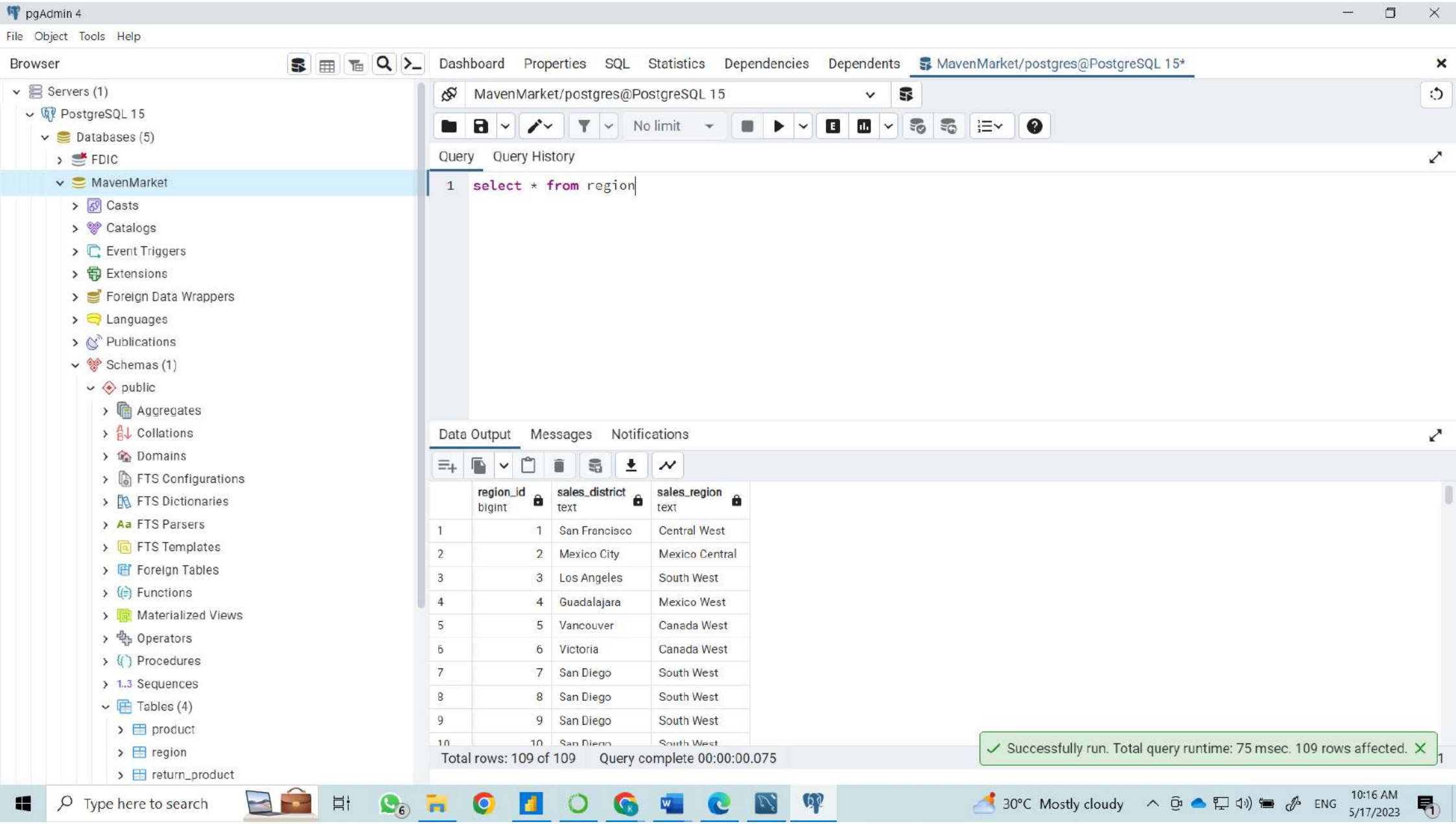
In [23]:
```
Returns.to_sql(name = 'return_product', con = engine, index = False, if_exists = 'replace')
```

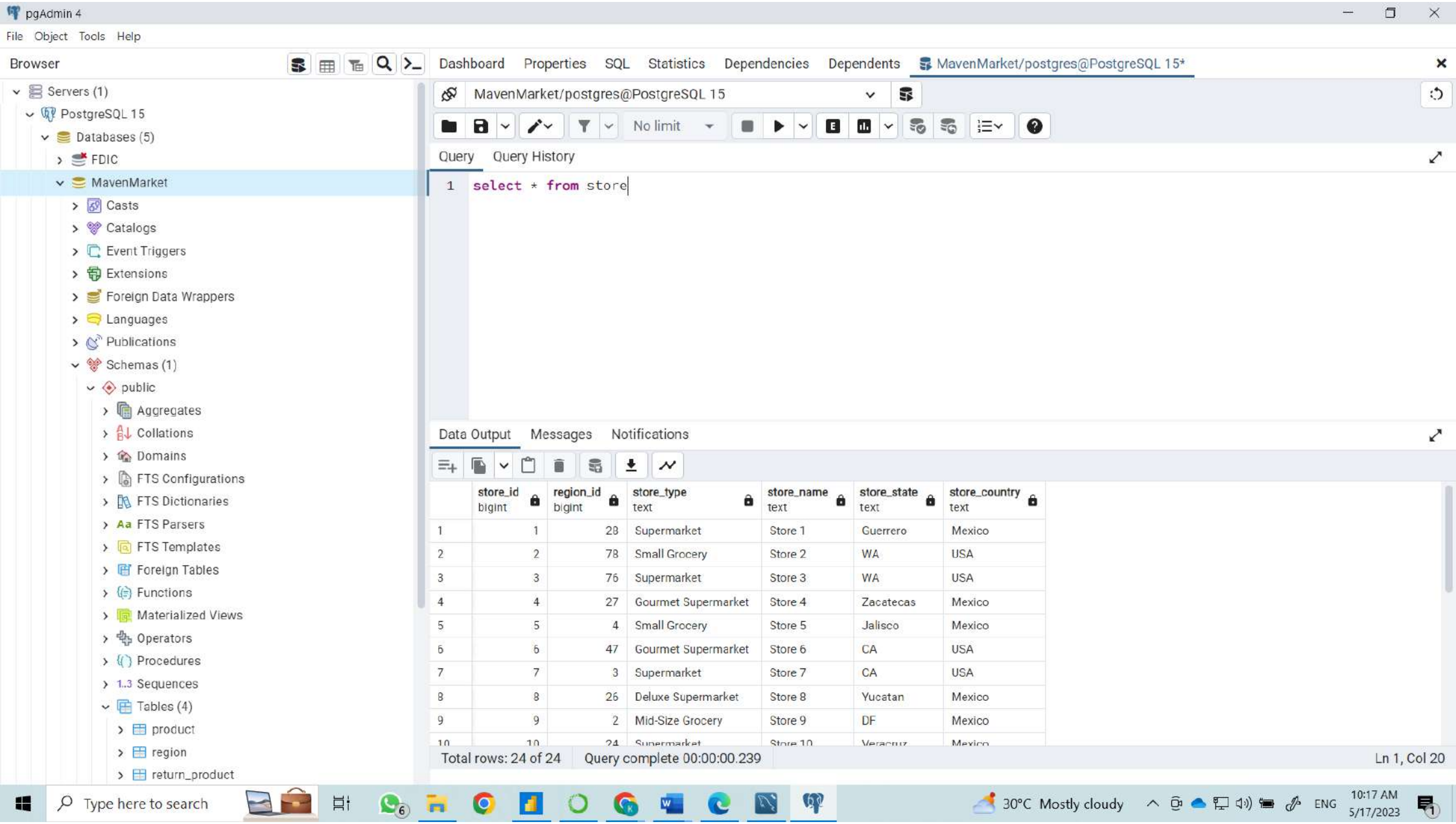
Out[23]:
```
87
```

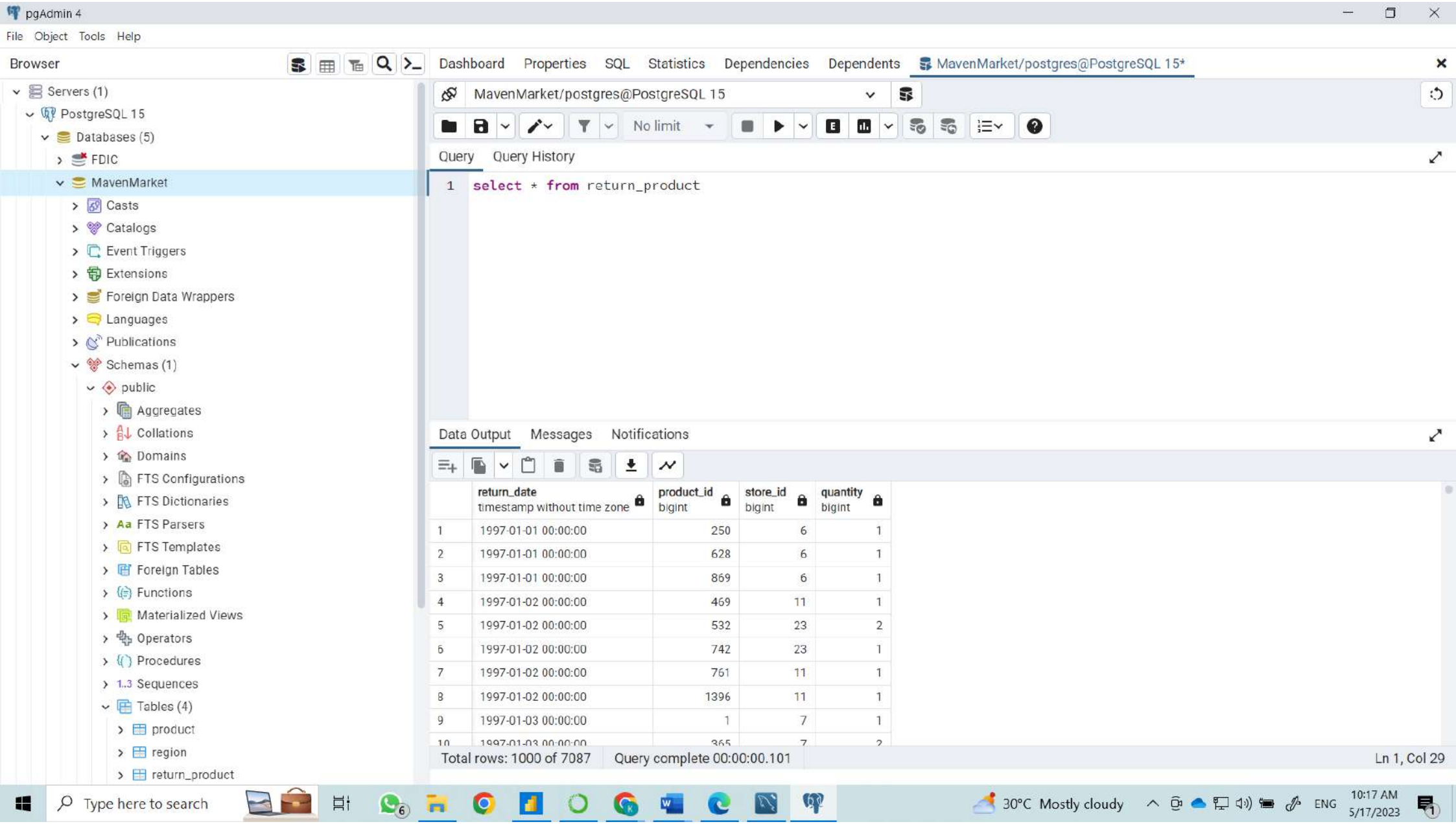
In [ ]:
```

```


pgAdmin 4
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents MavenMarket/postgres@PostgreSQL 15*
Servers (1)
PostgreSQL 15
Databases (5)
FDIC
MavenMarket
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (4)
product
region
return_product
MavenMarket/postgres@PostgreSQL 15
No limit
Query Query History
1 select * from product
Data Output Messages Notifications
product_id bigint
product_brand text
product_name text
product_retail_price double precision
product_cost double precision
1 1 Washington Washington Berry Juice 2.85 0.94
2 2 Washington Washington Mango Drink 0.74 0.26
3 3 Washington Washington Strawberry Drink 0.83 0.4
4 4 Washington Washington Cream Soda 3.64 1.64
5 5 Washington Washington Diet Soda 2.19 0.77
6 6 Washington Washington Cola 1.15 0.37
7 7 Washington Washington Diet Cola 2.61 0.91
8 8 Washington Washington Orange Juice 2.59 0.8
9 9 Washington Washington Cranberry Juice 2.42 0.77
Total rows: 1000 of 1560
Query complete 00:00:00.405
Ln 1, Col 22
Type here to search
30°C Mostly cloudy
ENG
10:16 AM
5/17/2023


pgAdmin 4
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents MavenMarket/postgres@PostgreSQL 15*
Servers (1)
PostgreSQL 15
Databases (5)
FDIC
MavenMarket
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (4)
product
region
return_product
MavenMarket/postgres@PostgreSQL 15
No limit
Query Query History
1 select * from region
Data Output Messages Notifications
region_id bigint
sales_district text
sales_region text
1 1 San Francisco Central West
2 2 Mexico City Mexico Central
3 3 Los Angeles South West
4 4 Guadalajara Mexico West
5 5 Vancouver Canada West
6 6 Victoria Canada West
7 7 San Diego South West
8 8 San Diego South West
9 9 San Diego South West
Total rows: 109 of 109
Query complete 00:00:00.075
Successfully run. Total query runtime: 75 msec. 109 rows affected.
Type here to search
30°C Mostly cloudy
ENG
10:16 AM
5/17/2023


pgAdmin 4
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents MavenMarket/postgres@PostgreSQL 15*
Servers (1)
PostgreSQL 15
Databases (5)
FDIC
MavenMarket
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (4)
product
region
return_product
MavenMarket/postgres@PostgreSQL 15
No limit
Query Query History
1 select * from store
Data Output Messages Notifications
store_id bigint
region_id bigint
store_type text
store_name text
store_state text
store_country text
1 1 28 Supermarket Store 1 Guerrero Mexico
2 2 78 Small Grocery Store 2 WA USA
3 3 76 Supermarket Store 3 WA USA
4 4 27 Gourmet Supermarket Store 4 Zacatecas Mexico
5 5 4 Small Grocery Store 5 Jalisco Mexico
6 6 47 Gourmet Supermarket Store 6 CA USA
7 7 3 Supermarket Store 7 CA USA
8 8 26 Deluxe Supermarket Store 8 Yucatan Mexico
9 9 2 Mid-Size Grocery Store 9 DF Mexico
Total rows: 24 of 24
Query complete 00:00:00.239
Ln 1, Col 20
Type here to search
30°C Mostly cloudy
ENG
10:17 AM
5/17/2023


pgAdmin 4
File Object Tools Help
Browser
Dashboard Properties SQL Statistics Dependencies Dependents MavenMarket/postgres@PostgreSQL 15*
Servers (1)
PostgreSQL 15
Databases (5)
FDIC
MavenMarket
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (4)
product
region
return_product
MavenMarket/postgres@PostgreSQL 15
No limit
Query Query History
1 select * from return_product
Data Output Messages Notifications
return_date timestamp without time zone
product_id bigint
store_id bigint
quantity bigint
1 1997-01-01 00:00:00 250 6 1
2 1997-01-01 00:00:00 628 6 1
3 1997-01-01 00:00:00 869 6 1
4 1997-01-02 00:00:00 469 11 1
5 1997-01-02 00:00:00 532 23 2
6 1997-01-02 00:00:00 742 23 1
7 1997-01-02 00:00:00 761 11 1
8 1997-01-02 00:00:00 1396 11 1
9 1997-01-03 00:00:00 1 7 1
Total rows: 1000 of 7087
Query complete 00:00:00.101
Ln 1, Col 29
Type here to search
30°C Mostly cloudy
ENG
10:17 AM
5/17/2023

MAVEN MARKET

Product Brand	Total Transactions	Total Profit	Profit Margin	Return Rate
Hermanos	5,342	$21,753	58.64%	0.95%
Ebony	5,238	$20,354	59.81%	0.96%
Tell Tale	5,112	$19,982	58.05%	0.99%
Tri-State	5,099	$19,980	58.91%	1.10%
High Top	4,940	$19,810	60.42%	1.01%
Nationeel	4,408	$18,617	60.44%	1.18%
Best Choice	4,218	$18,355	60.64%	0.81%
Horatio	4,195	$17,737	58.42%	1.26%
Fort West	4,108	$15,834	59.80%	0.97%
Fast	4,097	$16,469	61.03%	1.07%
Sunset	3,953	$14,018	60.45%	1.03%
Carrington	3,891	$14,883	59.52%	0.78%
Red Wing	3,870	$15,870	59.36%	1.06%
Big Time	3,816	$15,560	60.20%	1.05%
Cormorant	3,744	$15,749	61.60%	0.87%
Imagine	3,634	$15,102	61.40%	1.06%
Super	3,618	$13,868	60.59%	0.96%
Denny	3,584	$16,015	58.02%	0.99%
High Quality	3,577	$16,139	59.98%	1.13%
Golden	3,550	$13,256	58.72%	0.88%
BBB Best	3,514	$12,991	62.12%	0.80%
PigTail	3,467	$11,617	60.68%	1.04%
Plato	3,352	$12,748	63.55%	1.06%
Landslide	3,270	$10,647	58.65%	0.98%
CDR	3,078	$12,062	58.98%	1.11%
Better	2,823	$9,179	61.15%	1.07%
Carlson	2,564	$10,534	61.20%	0.97%
Pleasant	2,564	$10,187	60.18%	0.92%
Just Right	2,558	$9,283	59.54%	0.83%
Bravo	2,484	$11,027	59.15%	0.82%
Total	**113,668**	**$449,627**	**59.94%**	**1.00%**

Current Month Transactions

18,325

Goal: 17,339 (+5.69%)

Current Month Profit

$71,682

Goal: 67.87K (+5.61%)

Current Month Returns

496

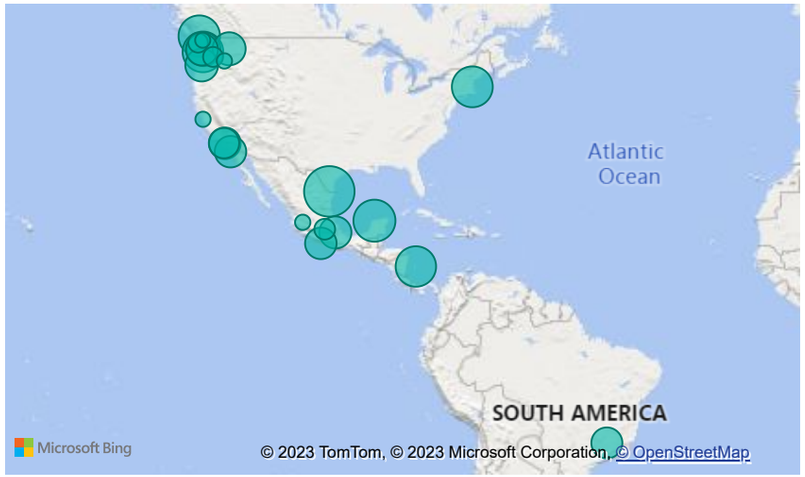
Goal: 482 (-2.9%)

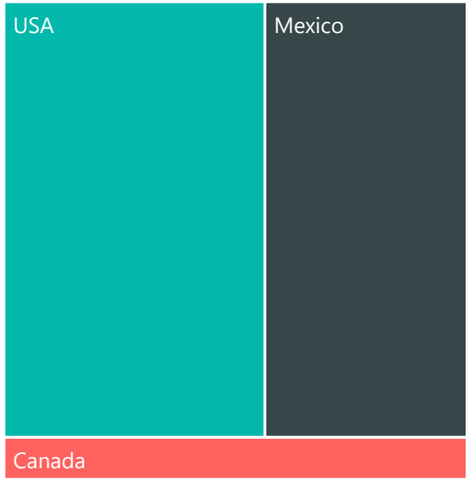
Select all

Canada

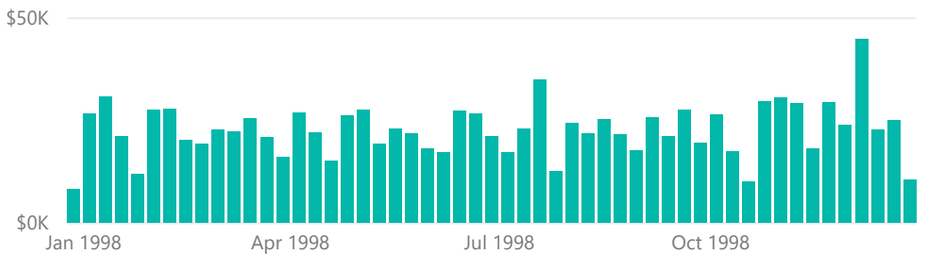
Mexico

USA





Weekly Revenue Trending



Revenue vs. Target