

TARGET BRAZIL

The story behind 100,000 orders...



INTRODUCTION

With the rapid growth of e-commerce, businesses rely on data to understand customer behavior, optimize sales strategies, and enhance operational efficiency. Target, a globally recognized retailer, operates in multiple markets, including Brazil, where online shopping has seen significant expansion.

This project explores Target's e-commerce sales data from 2016 to 2018, offering insights into various aspects of its online retail performance.



ABOUT THE DATASET

Target is a globally recognized brand and a leading retailer in the United States, known for its commitment to exceptional value, innovation, and a unique shopping experience.

This dataset focuses on Target's operations in Brazil, encompassing 100,000 orders placed between 2016 and 2018. It includes detailed insights into order status, pricing, payment methods, shipping performance, customer demographics, product attributes, and customer reviews. This data offers a valuable opportunity to analyze purchasing trends, operational efficiency, and customer satisfaction within the Brazilian market.



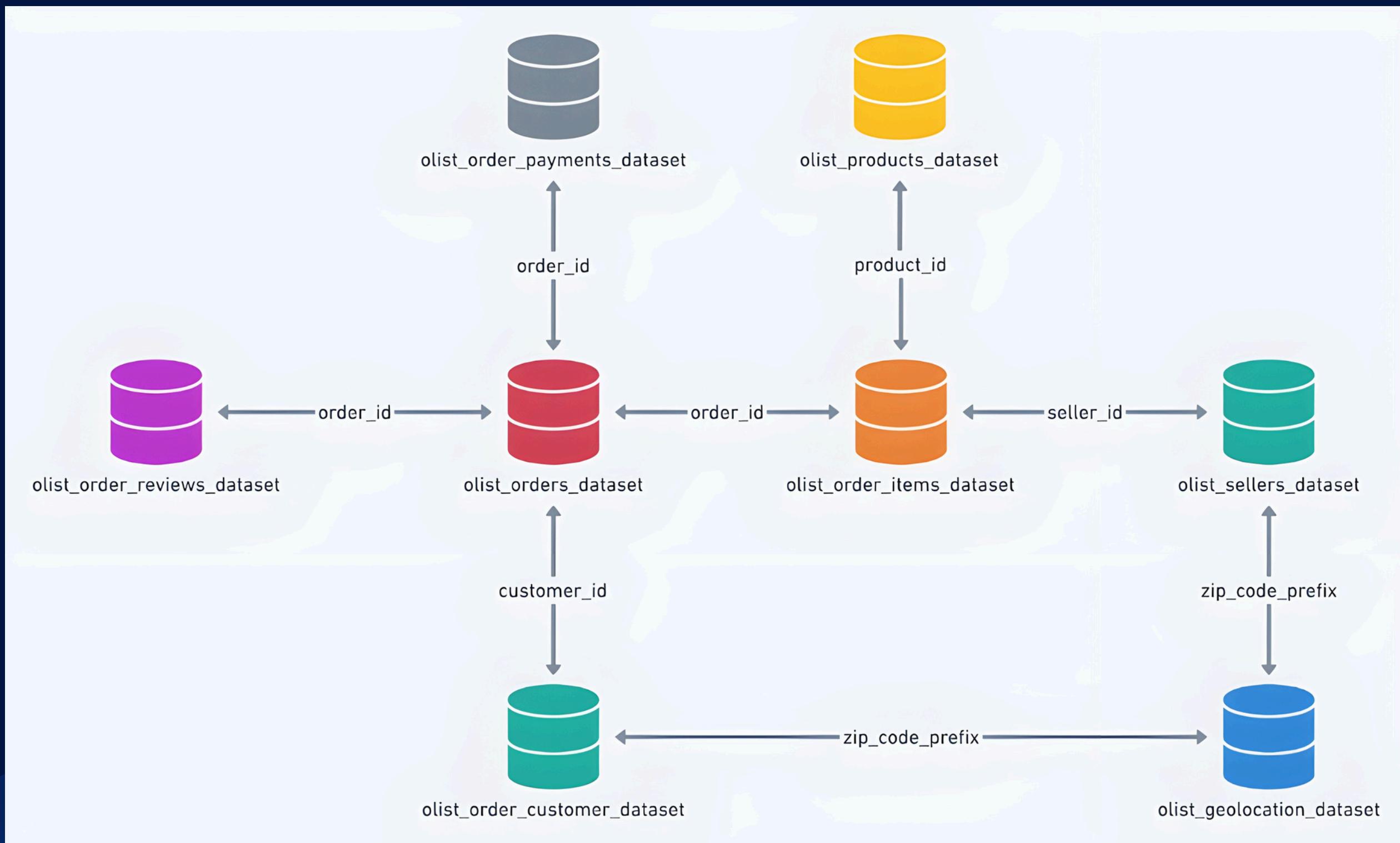
POTENTIAL USE CASES

Analyzing this dataset offers valuable insights into Target's Brazilian operations, revealing details about order processing, pricing strategies, payment and shipping efficiency, customer demographics, product characteristics, and customer satisfaction.

This comprehensive dataset is a valuable resource for understanding various business aspects and enhancing strategic decision-making.



DATASET SCHEMA



DATA IMPORT PROCESS

```
In [ ]: import pandas as pd
import mysql.connector
import os

# Database connection
conn = mysql.connector.connect(host='localhost', user='root', password='123456789', database='Target_sales')
cursor = conn.cursor()

# CSV files and corresponding table names
csv_files = ['customers', 'orders', 'geolocation', 'order_items', 'sellers', 'products', 'payments']
folder_path = 'D:/SQL Data Analysis'

# Function to determine SQL data type
def get_sql_type(dtype):
    return ('INT' if pd.api.types.is_integer_dtype(dtype) else
           'FLOAT' if pd.api.types.is_float_dtype(dtype) else
           'BOOLEAN' if pd.api.types.is_bool_dtype(dtype) else
           'DATETIME' if pd.api.types.is_datetime64_any_dtype(dtype) else 'TEXT')

# Import and insert data
for table in csv_files:
    df = pd.read_csv(os.path.join(folder_path, f"{table}.csv")).fillna(None)
    df.columns = [col.replace(' ', '_').replace('-', '_') for col in df.columns]

    # Create table
    columns = ', '.join([f'{col} {get_sql_type(df[col].dtype)}' for col in df.columns])
    cursor.execute(f'CREATE TABLE IF NOT EXISTS `{table}` ({columns})')

    # Insert data
    placeholders = ', '.join(['%s'] * len(df.columns))
    cursor.executemany(f"INSERT INTO `{table}` ({', '.join(df.columns)}) VALUES ({placeholders})", df.to_records(index=False))

conn.commit()
conn.close()
```

* Dataset Source : Kaggle

* Files Imported : CSV files

* Method Used : Python for importing data into SQL Server

DATASET TABLES OVERVIEW



CUSTOMERS TABLE

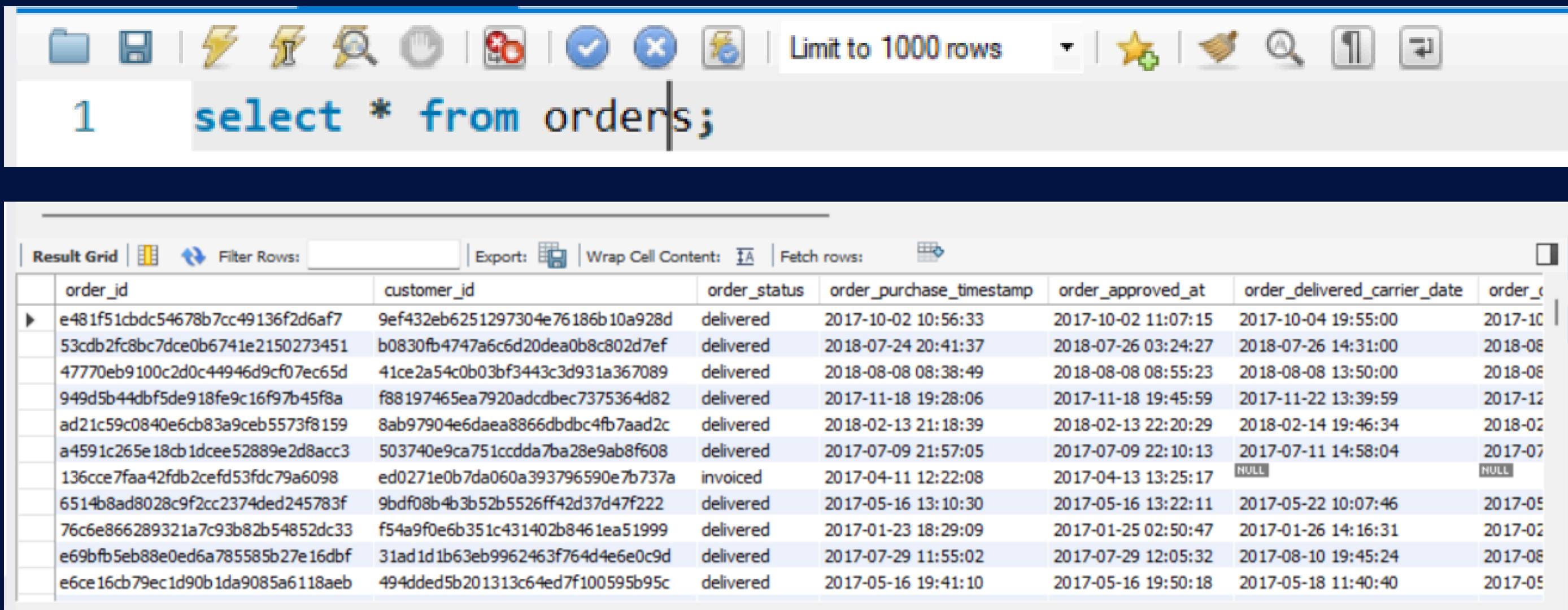
Target sales db* SQL File 1*

1 `select * from customers;`

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch rows:

	customer_id	customer_unique_id	customer_zip_code_prefix	customer_city	customer_state
▶	06b8999e2fba1a1fbc88172c00ba8bc7	861eff4711a542e4b93843c6dd7febb0	14409	franca	SP
	18955e83d337fd6b2def6b18a428ac77	290c77bc529b7ac935b93aa66c333dc3	9790	sao bernardo do campo	SP
	4e7b3e00288586ebd08712fdd0374a03	060e732b5b29e8181a18229c7b0b2b5e	1151	sao paulo	SP
	b2b6027bc5c5109e529d4dc6358b12c3	259dac757896d24d7702b9acbbff3f3c	8775	mogi das cruzes	SP
	4f2d8ab171c80ec8364f7c12e35b23ad	345ecd01c38d18a9036ed96c73b8d066	13056	campinas	SP
	879864dab9bc3047522c92c82e1212b8	4c93744516667ad3b8f1fb645a3116a4	89254	jaragua do sul	SC
	fd826e7cf63160e536e0908c76c3f441	addec96d2e059c80c30fe6871d30d177	4534	sao paulo	SP
	5e274e7a0c3809e14aba7ad5aae0d407	57b2a98a409812fe9618067b6b8ebe4f	35182	timoteo	MG
	5adf08e34b2e993982a47070956c5c65	1175e95fb47ddff9de6b2b06188f7e0d	81560	curitiba	PR
	4b7139f34592b3a31687243a302fa75b	9afe194fb833f79e300e37e580171f22	30575	belo horizonte	MG
	9fb35e4ed6f0a14a4977cd9aea4042bb	2a7745e1ed516b289ed9b29c7d0539a5	39400	montes claros	MG
	5aa9e4fdd4dfd20959cad2d772509598	2a46fb94aef5cbeeb850418118cee090	20231	rio de janeiro	RJ

ORDERS TABLE

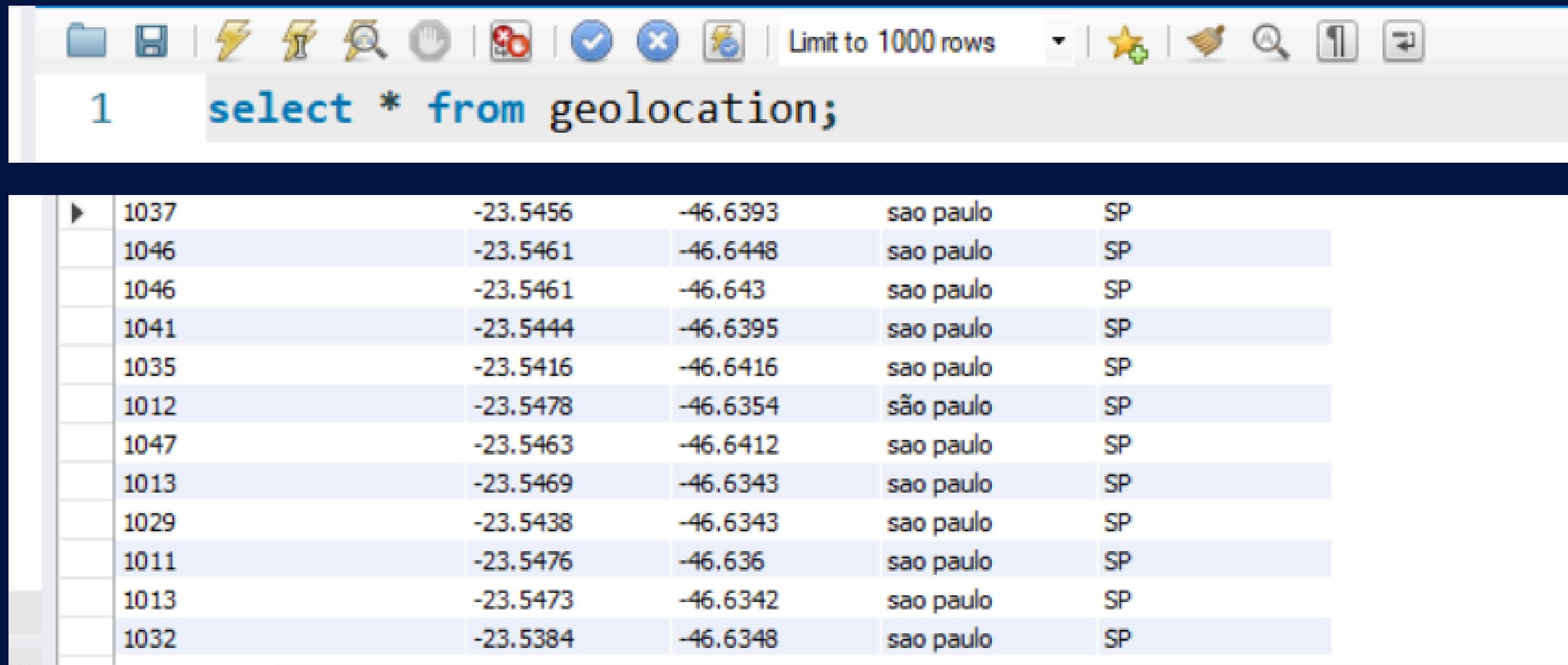


The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Print, Find, Copy, Paste, Undo, Redo), a search bar, and a "Limit to 1000 rows" dropdown.
- Query Editor:** Displays the SQL query: `1 select * from orders;`
- Result Grid:** Shows the data from the Orders table. The columns are: order_id, customer_id, order_status, order_purchase_timestamp, order_approved_at, order_delivered_carrier_date, order_delivered_customer_date, and order_canceled_at. The data includes various order IDs, customer IDs, statuses like 'delivered' or 'invoiced', purchase timestamps ranging from 2017-10-02 to 2018-07-24, and delivery dates.

order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_carrier_date	order_delivered_customer_date	order_canceled_at
e481f51cbdc54678b7cc49136f2d6af7	9ef432eb6251297304e76186b10a928d	delivered	2017-10-02 10:56:33	2017-10-02 11:07:15	2017-10-04 19:55:00	2017-10-04 19:55:00	2017-10-04 19:55:00
53cdb2fc8bc7dce0b6741e2150273451	b0830fb4747a6c6d20dea0b8c802d7ef	delivered	2018-07-24 20:41:37	2018-07-26 03:24:27	2018-07-26 14:31:00	2018-07-26 14:31:00	2018-08-01 00:00:00
47770eb9100c2d0c44946d9cf07ec65d	41ce2a54c0b03bf3443c3d931a367089	delivered	2018-08-08 08:38:49	2018-08-08 08:55:23	2018-08-08 13:50:00	2018-08-08 13:50:00	2018-08-08 13:50:00
949d5b44dbf5de918fe9c16f97b45f8a	f88197465ea7920adcdbec7375364d82	delivered	2017-11-18 19:28:06	2017-11-18 19:45:59	2017-11-22 13:39:59	2017-11-22 13:39:59	2017-12-01 00:00:00
ad21c59c0840e6cb83a9ceb5573f8159	8ab97904e6daea8866dbdbc4fb7aad2c	delivered	2018-02-13 21:18:39	2018-02-13 22:20:29	2018-02-14 19:46:34	2018-02-14 19:46:34	2018-02-14 19:46:34
a4591c265e18cb1dcee52889e2d8acc3	503740e9ca751ccdda7ba28e9ab8f608	delivered	2017-07-09 21:57:05	2017-07-09 22:10:13	2017-07-11 14:58:04	2017-07-11 14:58:04	2017-07-11 14:58:04
136cce7faa42fdb2cef53fdc79a6098	ed0271e0b7da060a393796590e7b737a	invoiced	2017-04-11 12:22:08	2017-04-13 13:25:17	NULL	NULL	NULL
6514b8ad8028c9f2cc2374ded245783f	9bdf08b4b3b52b5526ff42d37d47f222	delivered	2017-05-16 13:10:30	2017-05-16 13:22:11	2017-05-22 10:07:46	2017-05-22 10:07:46	2017-05-22 10:07:46
76c6e866289321a7c93b82b54852dc33	f54a9f0e6b351c431402b8461ea51999	delivered	2017-01-23 18:29:09	2017-01-25 02:50:47	2017-01-26 14:16:31	2017-01-26 14:16:31	2017-02-01 00:00:00
e69bfb5eb88e0ed6a785585b27e16dbf	31ad1d1b63eb9962463f764d4e6e0c9d	delivered	2017-07-29 11:55:02	2017-07-29 12:05:32	2017-08-10 19:45:24	2017-08-10 19:45:24	2017-08-10 19:45:24
e6ce16cb79ec1d90b1da9085a6118aeb	494dded5b201313c64ed7f100595b95c	delivered	2017-05-16 19:41:10	2017-05-16 19:50:18	2017-05-18 11:40:40	2017-05-18 11:40:40	2017-05-18 11:40:40

GEOLOCATION TABLE

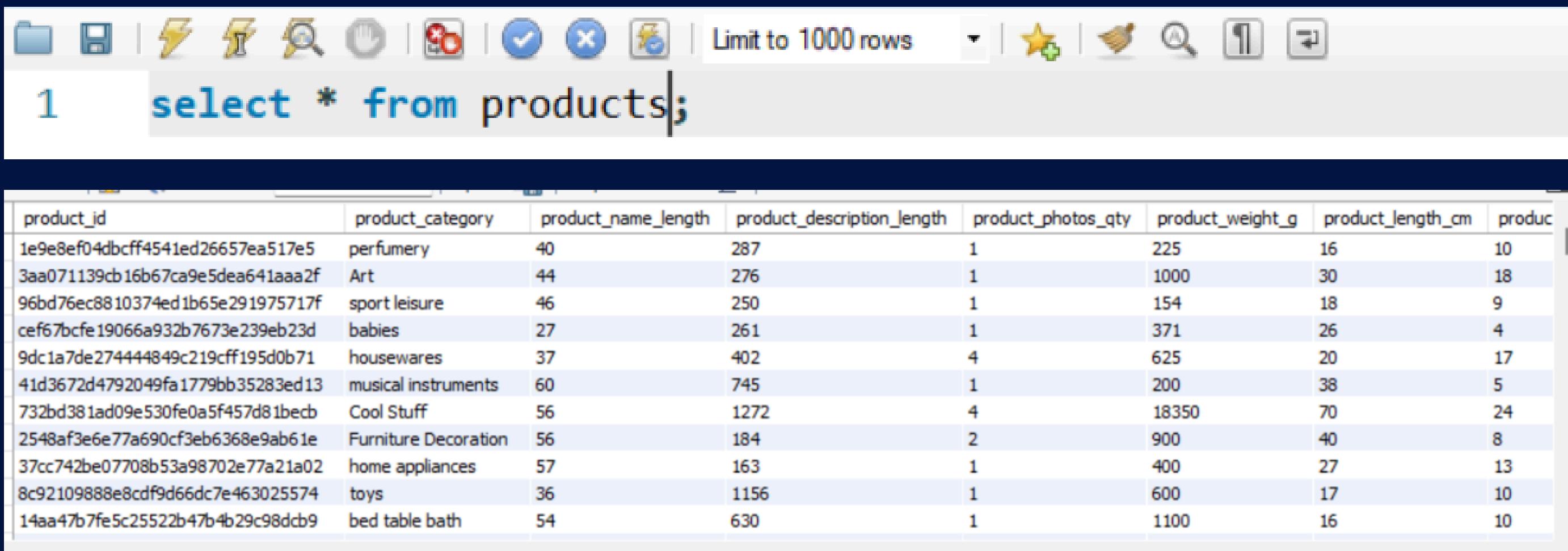


The screenshot shows a MySQL Workbench interface with the following details:

- Toolbar:** Includes icons for file operations (New, Open, Save, Import, Export), search, and various database management functions.
- Query Editor:** Displays the SQL query: `1 select * from geolocation;`
- Result Grid:** Shows 12 rows of data from the `geolocation` table. The columns are: ID, Latitude, Longitude, City, and State/Region. The data is as follows:

ID	Latitude	Longitude	City	State/Region
1037	-23.5456	-46.6393	sao paulo	SP
1046	-23.5461	-46.6448	sao paulo	SP
1046	-23.5461	-46.643	sao paulo	SP
1041	-23.5444	-46.6395	sao paulo	SP
1035	-23.5416	-46.6416	sao paulo	SP
1012	-23.5478	-46.6354	são paulo	SP
1047	-23.5463	-46.6412	sao paulo	SP
1013	-23.5469	-46.6343	sao paulo	SP
1029	-23.5438	-46.6343	sao paulo	SP
1011	-23.5476	-46.636	sao paulo	SP
1013	-23.5473	-46.6342	sao paulo	SP
1032	-23.5384	-46.6348	sao paulo	SP

PRODUCTS TABLE

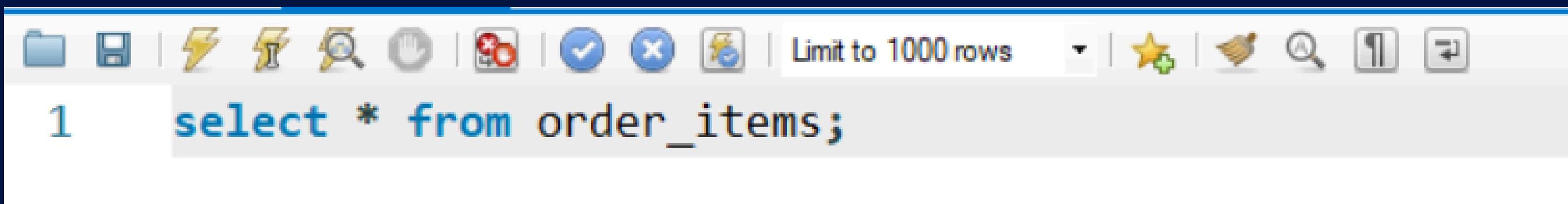


The screenshot shows the MySQL Workbench interface with the following details:

- Toolbar:** Includes icons for file operations, database selection, schema browser, table editor, search, and refresh.
- Query Editor:** Shows a single query: `1 select * from products;`. The number '1' indicates the current row in the results.
- Results Grid:** Displays the data from the 'products' table. The columns are: product_id, product_category, product_name_length, product_description_length, product_photos_qty, product_weight_g, product_length_cm, and product_width_mm. The data rows are as follows:

product_id	product_category	product_name_length	product_description_length	product_photos_qty	product_weight_g	product_length_cm	product_width_mm
1e9e8ef04dbcff4541ed26657ea517e5	perfumery	40	287	1	225	16	10
3aa071139cb16b67ca9e5dea641aaa2f	Art	44	276	1	1000	30	18
96bd76ec8810374ed1b65e291975717f	sport leisure	46	250	1	154	18	9
cef67bcfe19066a932b7673e239eb23d	babies	27	261	1	371	26	4
9dc1a7de274444849c219cff195d0b71	housewares	37	402	4	625	20	17
41d3672d4792049fa1779bb35283ed13	musical instruments	60	745	1	200	38	5
732bd381ad09e530fe0a5f457d81becb	Cool Stuff	56	1272	4	18350	70	24
2548af3e6e77a690cf3eb6368e9ab61e	Furniture Decoration	56	184	2	900	40	8
37cc742be07708b53a98702e77a21a02	home appliances	57	163	1	400	27	13
8c92109888e8cdf9d66dc7e463025574	toys	36	1156	1	600	17	10
14aa47b7fe5c25522b47b4b29c98dc9	bed table bath	54	630	1	1100	16	10

ORDER ITEMS TABLE



The screenshot shows a MySQL Workbench interface with a query editor and a results grid.

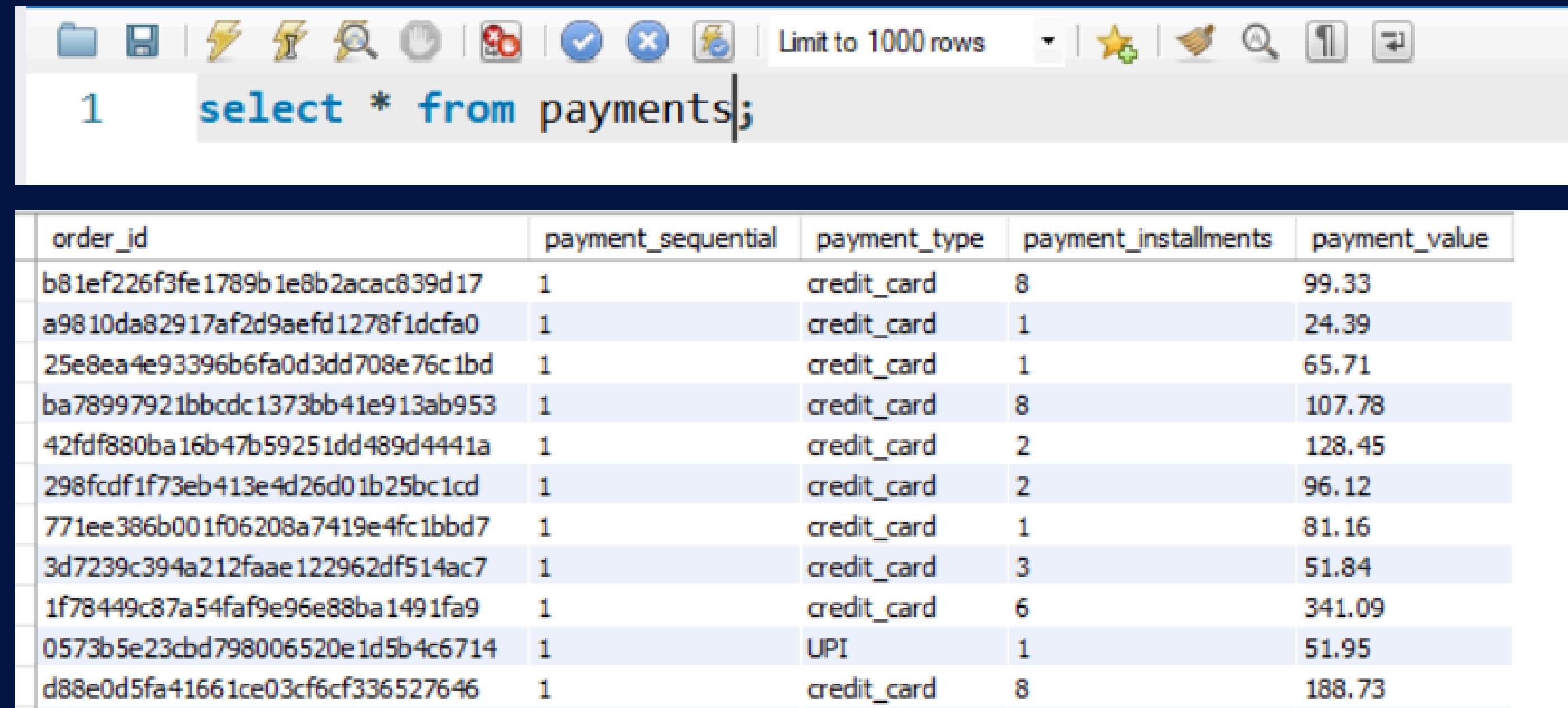
Query:

```
1 select * from order_items;
```

Results:

order_id	order_item_id	product_id	seller_id	shipping_limit_date	price	freight_value
00010242fe8c5a6d1ba2dd792cb16214	1	4244733e06e7ecb4970a6e2683c13e61	48436dade18ac8b2bce089ec2a041202	2017-09-19 09:45:35	58.9	13.29
00018f77f2f0320c557190d7a144bdd3	1	e5f2d52b802189ee658865ca93d83a8f	dd7ddc04e1b6c2c614352b383efe2d36	2017-05-03 11:05:13	239.9	19.93
000229ec398224ef6ca0657da4fc703e	1	c777355d18b72b67abbeef9df44fd0fd	5b51032eddd242adc84c38acab88f23d	2018-01-18 14:48:30	199	17.87
00024acbcdf0a6daa1e931b038114c75	1	7634da152a4610f1595efa32f14722fc	9d7a1d34a5052409006425275ba1c2b4	2018-08-15 10:10:18	12.99	12.79
00042b26cf59d7ce69dfabb4e55b4fd9	1	ac6c3623068f30de03045865e4e10089	df560393f3a51e74553ab94004ba5c87	2017-02-13 13:57:51	199.9	18.14
00048cc3ae777c65dbb7d2a0634bc1ea	1	ef92defde845ab8450f9d70c526ef70f	6426d21aca402a131fc0a5d0960a3c90	2017-05-23 03:55:27	21.9	12.69
00054e8431b9d7675808bcb819fb4a32	1	8d4f2bb7e93e6710a28f34fa83ee7d28	7040e82f899a04d1b434b795a43b4617	2017-12-14 12:10:31	19.9	11.85
000576fe39319847ccb9d288c5617fa6	1	557d850972a7d6f792fd18ae1400d9b6	5996cddab893a4652a15592fb58ab8db	2018-07-10 12:30:45	810	70.75
0005a1a1728c9d785b8e2b08b904576c	1	310ae3c140ff94b03219ad0adc3c778f	a416b6a846a11724393025641d4edd5e	2018-03-26 18:31:29	145.95	11.65
0005f50442cb953dcd1d21e1fb923495	1	4535b0e1091c278dfd193e5a1d63b39f	ba143b05f0110f0dc71ad71b4466ce92	2018-07-06 14:10:56	53.99	11.4
00061f2a7bc09da83e415a52dc8a4af1	1	d63c1011f49d98b976c352955b1c4bea	cc419e0650a3c5ba77189a1882b7556a	2018-03-29 22:28:09	59.99	8.88
00063b381e2406b52ad429470734ebd5	1	f177554ea93259a5b282f24e33f65ab6	8602a61d680a10a82cceeeda0d99ea3d	2018-07-31 17:30:39	45	12.98

PAYMENTS TABLE

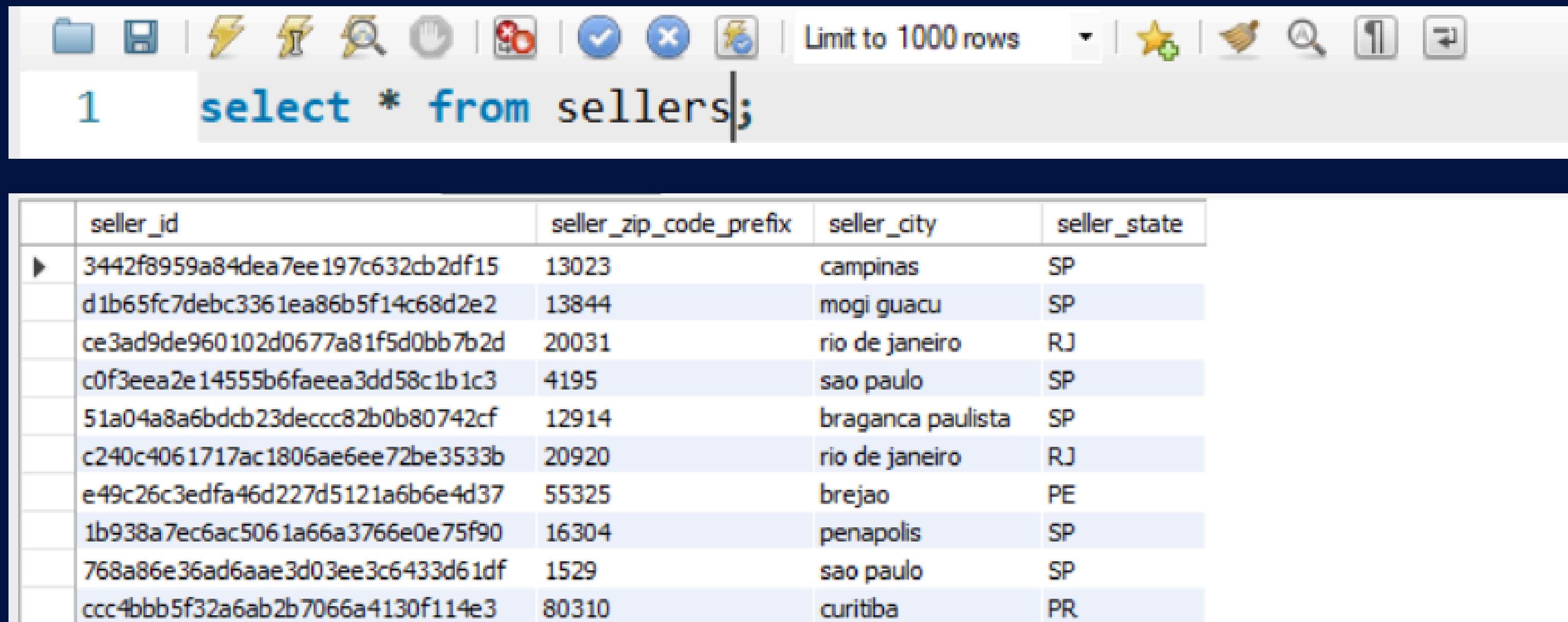


The screenshot shows a MySQL Workbench interface with the following details:

- Toolbar:** Includes icons for file operations, refresh, search, and other database management functions.
- Query Editor:** Displays the SQL query: `1 select * from payments;`
- Result Grid:** Shows the data from the payments table with the following columns and rows:

	order_id	payment_sequential	payment_type	payment_installments	payment_value
1	b81ef226f3fe1789b1e8b2acac839d17	1	credit_card	8	99.33
2	a9810da82917af2d9aefd1278f1dcfa0	1	credit_card	1	24.39
3	25e8ea4e93396b6fa0d3dd708e76c1bd	1	credit_card	1	65.71
4	ba78997921bbcdcc1373bb41e913ab953	1	credit_card	8	107.78
5	42fdf880ba16b47b59251dd489d4441a	1	credit_card	2	128.45
6	298fcdf1f73eb413e4d26d01b25bc1cd	1	credit_card	2	96.12
7	771ee386b001f06208a7419e4fc1bbd7	1	credit_card	1	81.16
8	3d7239c394a212faae122962df514ac7	1	credit_card	3	51.84
9	1f78449c87a54faf9e96e88ba1491fa9	1	credit_card	6	341.09
10	0573b5e23cbd798006520e1d5b4c6714	1	UPI	1	51.95
11	d88e0d5fa41661ce03cf6cf336527646	1	credit_card	8	188.73

SELLERS TABLE



The screenshot shows a MySQL Workbench interface with the following details:

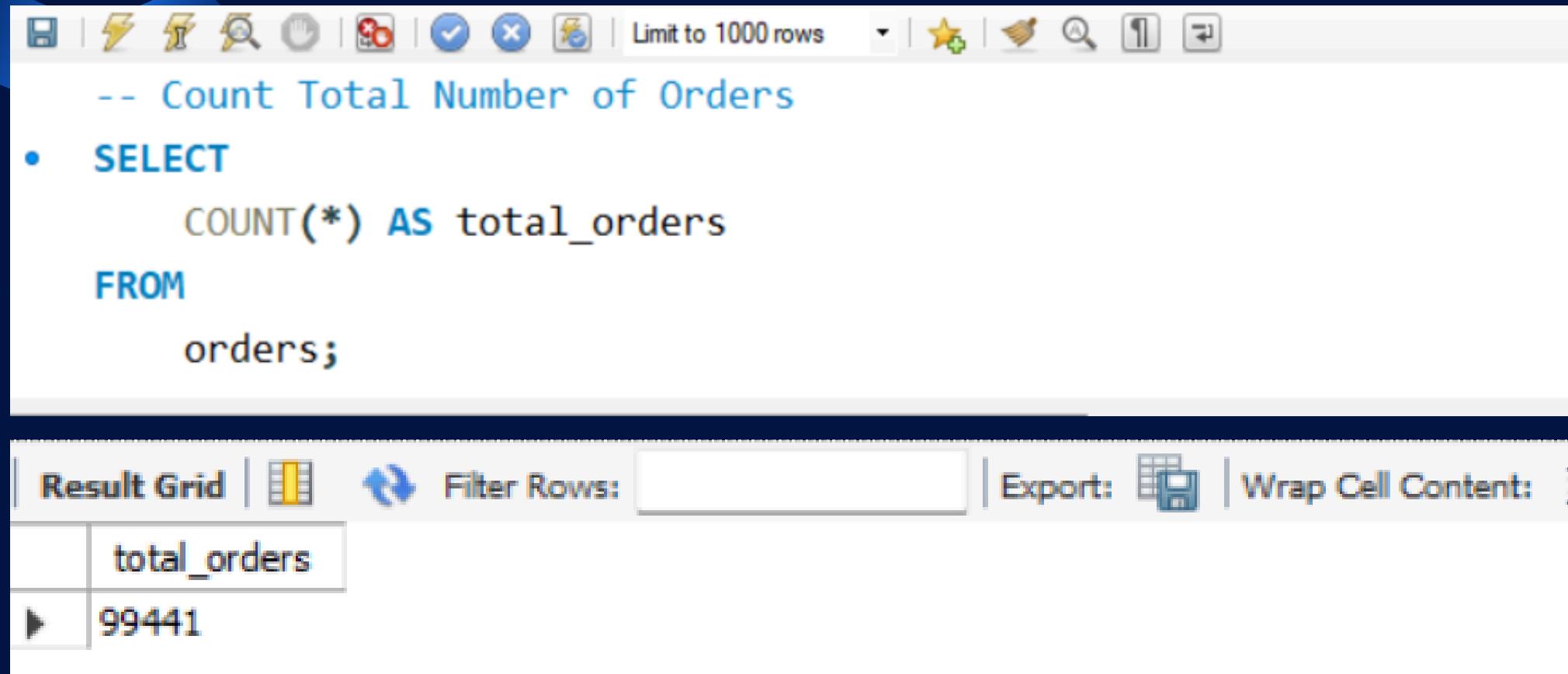
- Toolbar:** Includes icons for file operations (New, Open, Save, Import, Export, Paste), search (Find, Find Next, Replace, Find in Table), and other database management functions.
- Query Editor:** Displays the SQL query: `1 select * from sellers;`
- Results Grid:** Shows the data from the sellers table with the following columns: seller_id, seller_zip_code_prefix, seller_city, and seller_state. The data consists of ten rows, each with a unique seller_id and corresponding city and state information.

	seller_id	seller_zip_code_prefix	seller_city	seller_state
▶	3442f8959a84dea7ee197c632cb2df15	13023	campinas	SP
	d1b65fc7debc3361ea86b5f14c68d2e2	13844	mogi guacu	SP
	ce3ad9de960102d0677a81f5d0bb7b2d	20031	rio de janeiro	RJ
	c0f3eea2e14555b6faeee3dd58c1b1c3	4195	sao paulo	SP
	51a04a8a6bdcb23deccc82b0b80742cf	12914	braganca paulista	SP
	c240c4061717ac1806ae6ee72be3533b	20920	rio de janeiro	RJ
	e49c26c3edfa46d227d5121a6b6e4d37	55325	brejao	PE
	1b938a7ec6ac5061a66a3766e0e75f90	16304	penapolis	SP
	768a86e36ad6aae3d03ee3c6433d61df	1529	sao paulo	SP
	ccc4bbb5f32a6ab2b7066a4130f114e3	80310	curitiba	PR

SQl QUERIES & DATA INSIGHTS



COUNT TOTAL NUMBER OF ORDERS



-- Count Total Number of Orders

- **SELECT**

```
COUNT(*) AS total_orders
```

FROM

```
orders;
```

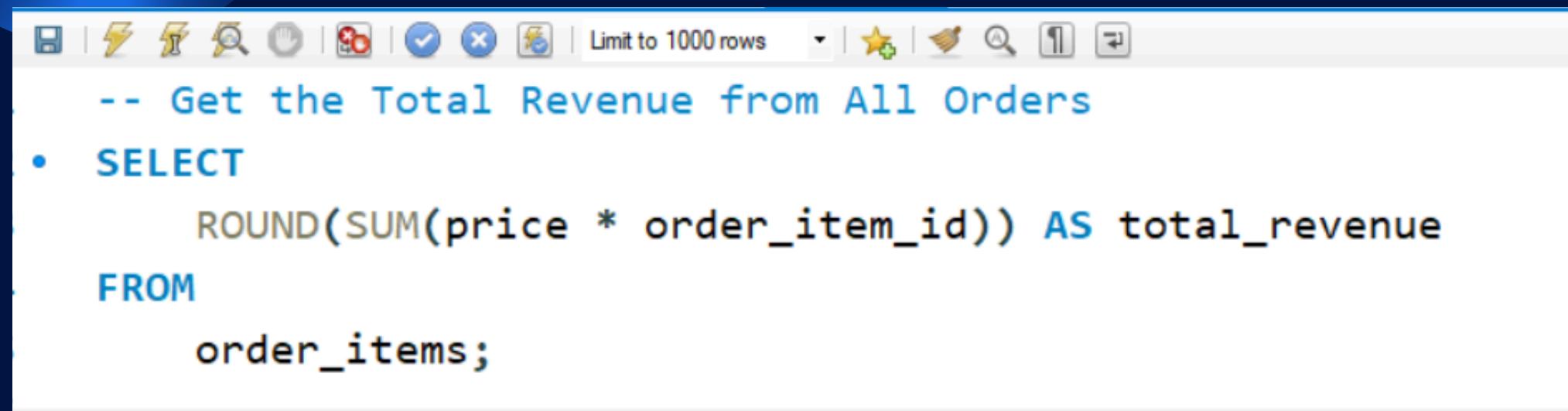
Result Grid | Filter Rows: Export: Wrap Cell Content:

	total_orders
▶	99441

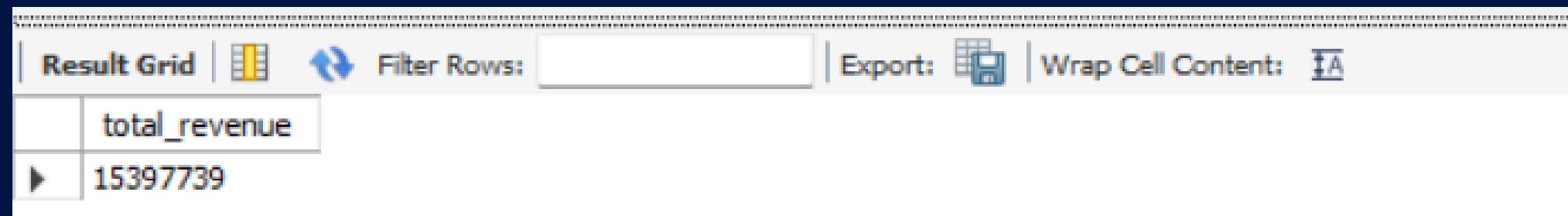
Key Insights:

- Total Orders: 99,441 orders were placed between 2016 and 2018.
- Indicates Market Demand: A high number of orders suggest strong customer engagement with Target's e-commerce platform in Brazil.
- Baseline for Further Analysis: This metric serves as a foundation to compare other factors, such as order trends over time, customer retention, and revenue generation.

GET THE TOTAL REVENUE FROM ALL ORDERS



```
-- Get the Total Revenue from All Orders
• SELECT
    ROUND(SUM(price * order_item_id)) AS total_revenue
FROM
    order_items;
```

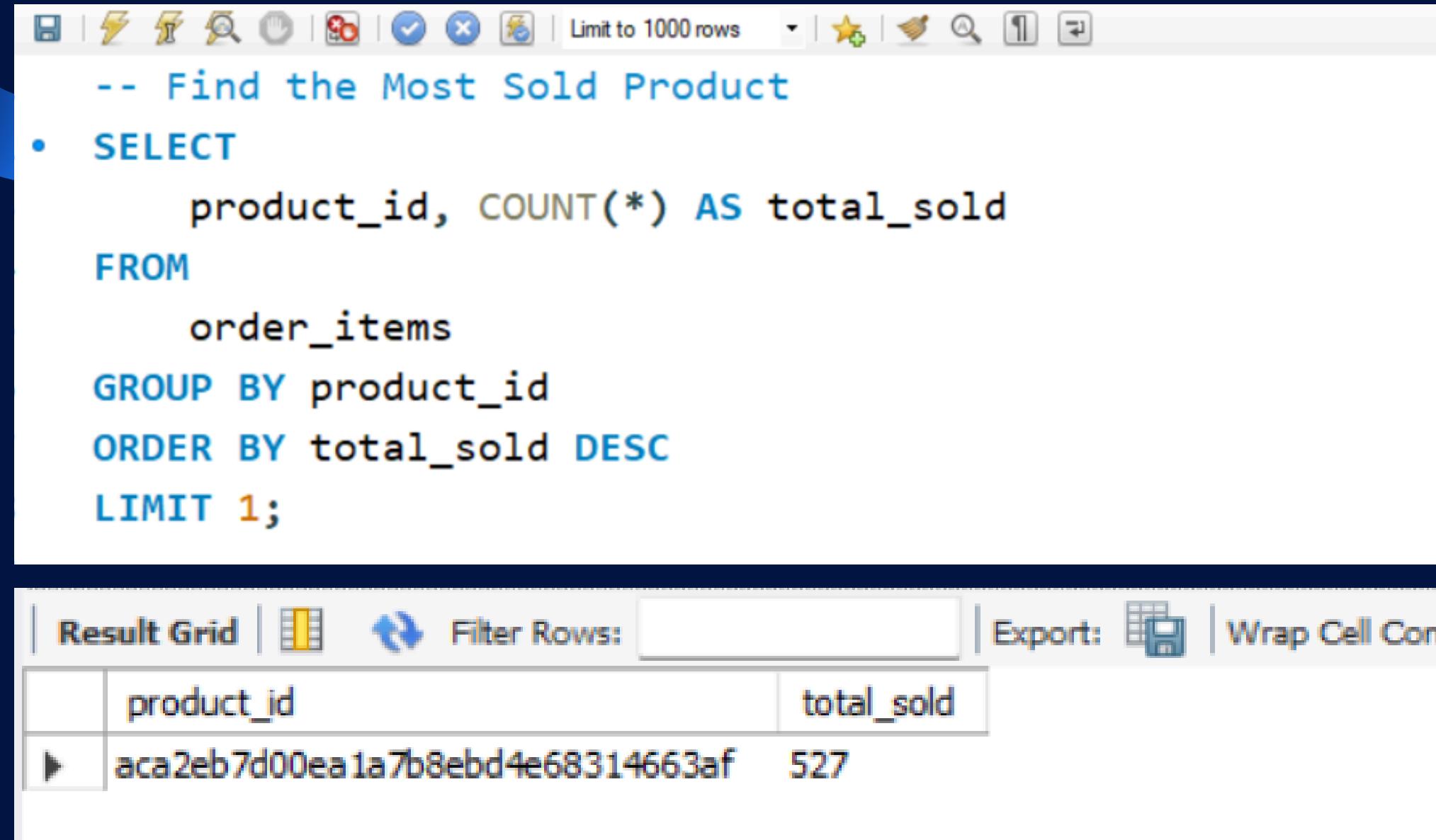


	total_revenue
▶	15397739

Key Insights:

- Total Revenue: The total revenue generated from all orders is 15,397,739
- Business Impact: This metric helps assess the financial performance of Target's e-commerce operations in Brazil and can be compared with marketing efforts, seasonal trends, and operational costs.

FIND THE MOST SOLD PRODUCT



-- Find the Most Sold Product

- **SELECT**

```
product_id, COUNT(*) AS total_sold
```
- **FROM**

```
order_items
```
- **GROUP BY** product_id
- **ORDER BY** total_sold **DESC**
- **LIMIT** 1;

product_id	total_sold
aca2eb7d00ea1a7b8ebd4e68314663af	527

Key Insights:

- Most Sold Product: The product with `product_id = 527` was sold the most.
- High Demand Indicator: This product had the highest number of purchases, making it a key revenue driver.

FIND TOTAL NUMBER OF ORDERS BY PAYMENT TYPE

- **SELECT**

```
payment_type, COUNT(*) AS total_orders
```

```
FROM
```

```
payments
```

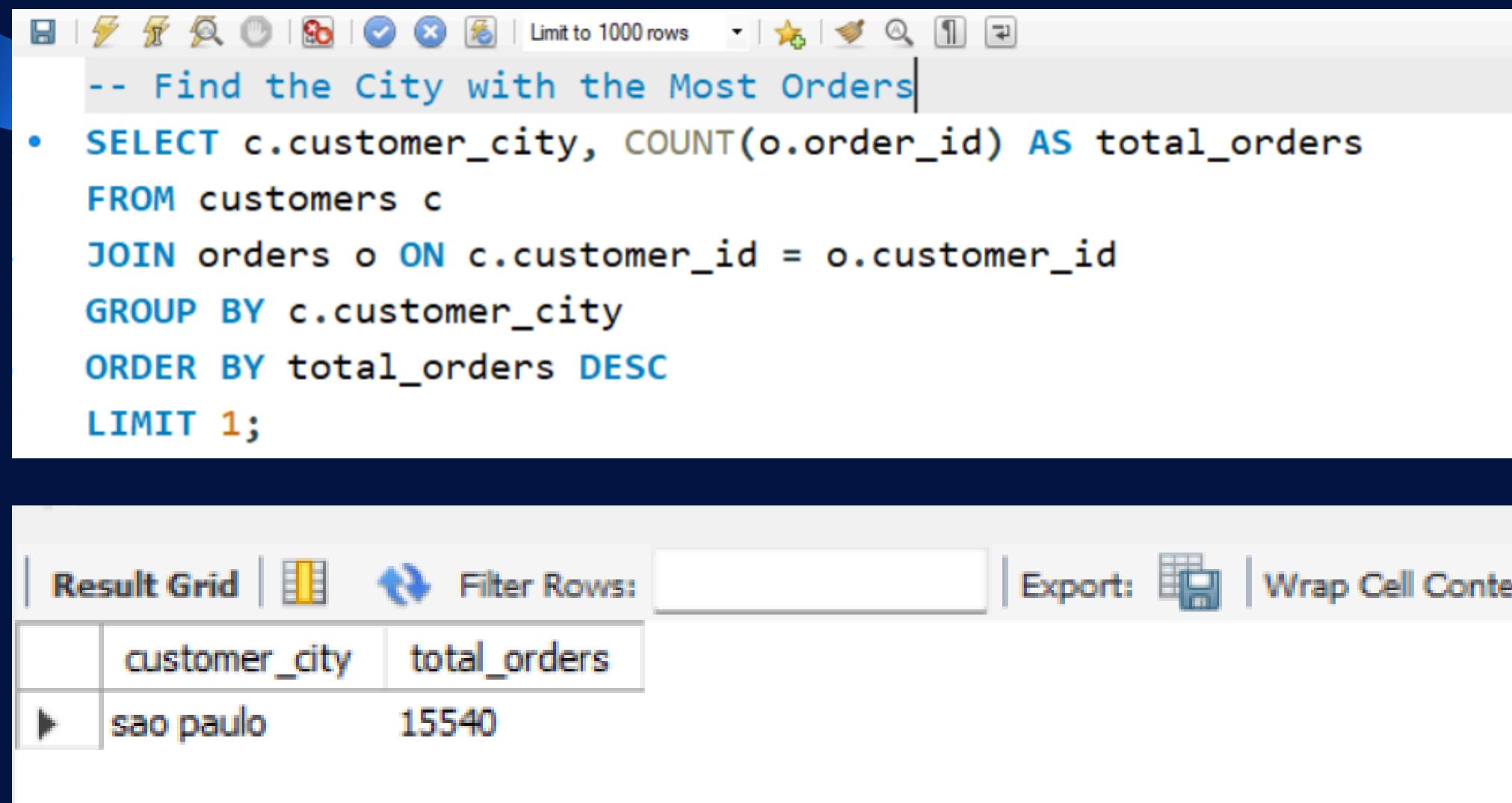
```
GROUP BY payment_type;
```

payment_type	total_orders
credit_card	76795
UPI	19784
voucher	5775
debit_card	1529

Key Insights:

- Most Preferred Payment Method: Credit Card is the dominant payment method, accounting for 76,795 orders ($\approx 77\%$ of total transactions).
- Digital Payments Adoption: UPI (19,784 orders) is the second most used method, showing a significant shift towards digital payments.
- Vouchers (5,775 orders) indicate that some customers prefer store credits or discounts.
- Debit Cards (1,529 orders) are the least used, possibly due to customer preference for credit-based transactions.

FIND THE CITY WITH MOST ORDERS



-- Find the City with the Most Orders

```
• SELECT c.customer_city, COUNT(o.order_id) AS total_orders
  FROM customers c
  JOIN orders o ON c.customer_id = o.customer_id
  GROUP BY c.customer_city
  ORDER BY total_orders DESC
  LIMIT 1;
```

	customer_city	total_orders
▶	sao paulo	15540

Key Insights:

- Top Ordering City: São Paulo has the highest number of orders, totaling 15,540.
- Major Customer Base: This indicates that São Paulo is a key market for Target's e-commerce business in Brazil.

GET THE MOST POPULAR PAYMENT TYPE IN EACH STATE

```
2 • SELECT
3     c.customer_state,
4     p.payment_type,
5     COUNT(*) AS total_transactions
6   FROM
7     payments p
8       JOIN
9     orders o ON p.order_id = o.order_id
10      JOIN
11    customers c ON o.customer_id = c.customer_id
12 GROUP BY c.customer_state , p.payment_type
13 ORDER BY c.customer state , total transactions DESC;
```

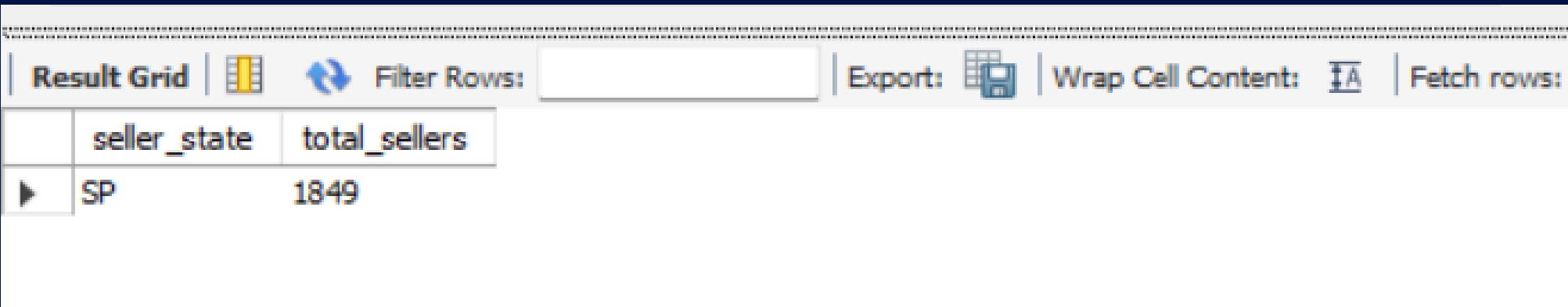
customer_state	payment_type	total_transactions
AC	credit_card	61
AC	UPI	16
AC	voucher	5
AC	debit_card	2
AL	credit_card	341
AL	UPI	68
AL	voucher	13
AL	debit_card	5
AM	credit_card	124
AM	UPI	21

Key Insights:

- State-wise Payment Preference: This query identifies the most frequently used payment method in each state.
- Dominance of Certain Payment Methods: Patterns may emerge, such as credit cards being the most preferred in urban areas while UPI or vouchers might be more common in rural regions.

FIND THE STATE WITH THE HIGHEST NUMBER OF SELLERS

```
-- Find the State with the Highest Number of Sellers
SELECT seller_state, COUNT(*) AS total_sellers
FROM sellers
GROUP BY seller_state
ORDER BY total_sellers DESC
LIMIT 1;
```



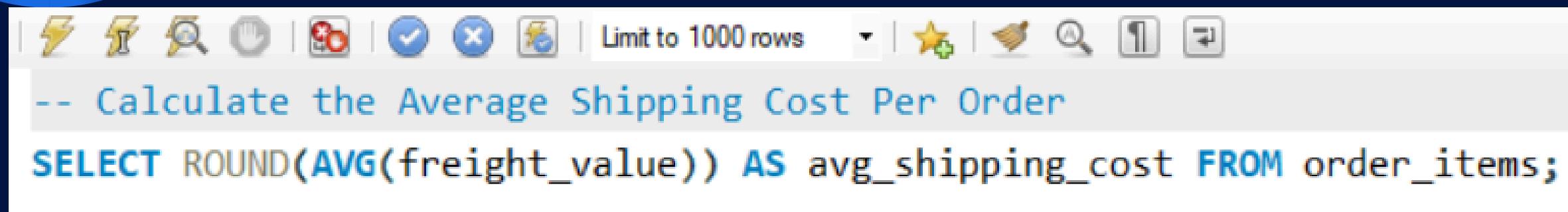
The screenshot shows a database query results grid. The grid has two columns: 'seller_state' and 'total_sellers'. There is one row of data: SP and 1849. The grid includes standard database navigation buttons like 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'.

	seller_state	total_sellers
▶	SP	1849

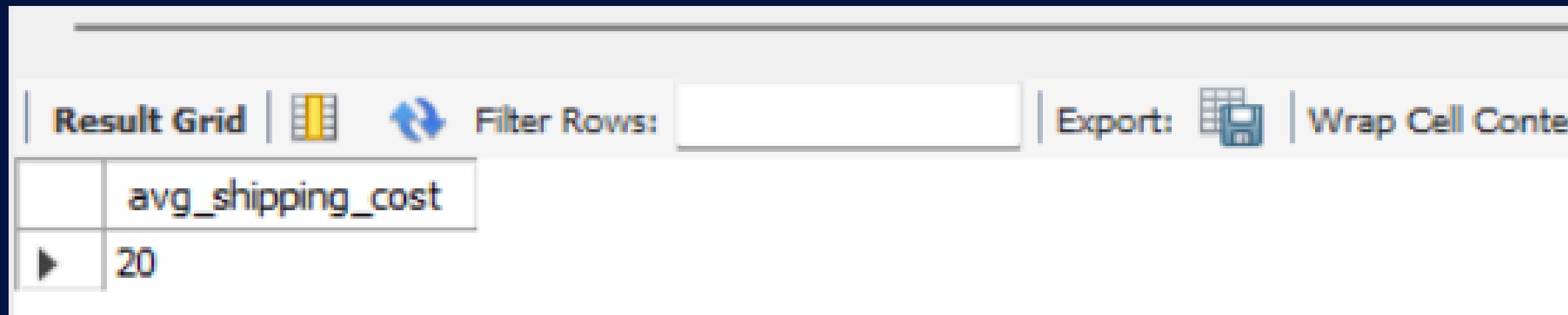
Key Insights:

- Top Seller State: São Paulo (SP) has the highest number of sellers, with 1,849 registered.
- E-Commerce Hub: This suggests that São Paulo is a major business hub for online sellers.

CALCULATE THE AVERAGE SHIPPING COST PER ORDER



```
-- Calculate the Average Shipping Cost Per Order
SELECT ROUND(AVG(freight_value)) AS avg_shipping_cost FROM order_items;
```



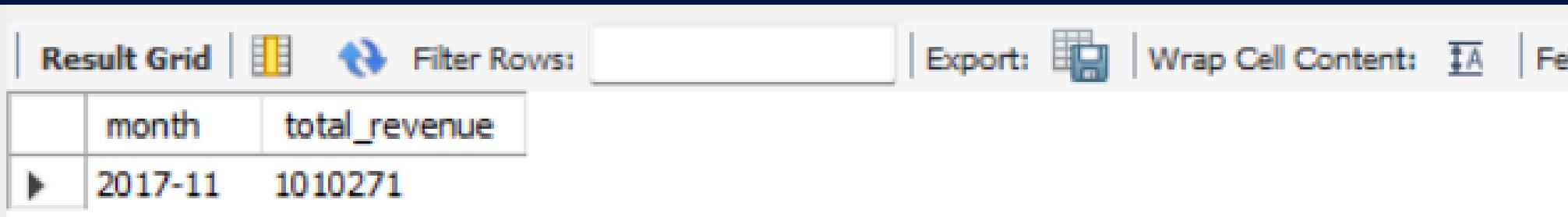
avg_shipping_cost
20

Key Insights:

- Average Shipping Cost: The typical shipping cost per order is 20.
- Operational Efficiency: This metric helps in understanding logistics costs and pricing strategies.

FIND THE MONTH WITH THE HIGHEST SALES REVENUE

```
-- Find the Month with the Highest Sales Revenue
• SELECT
    DATE_FORMAT(order_purchase_timestamp, '%Y-%m') AS month,
    ROUND(SUM(price)) AS total_revenue
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_id
GROUP BY month
ORDER BY total_revenue DESC
LIMIT 1;
```



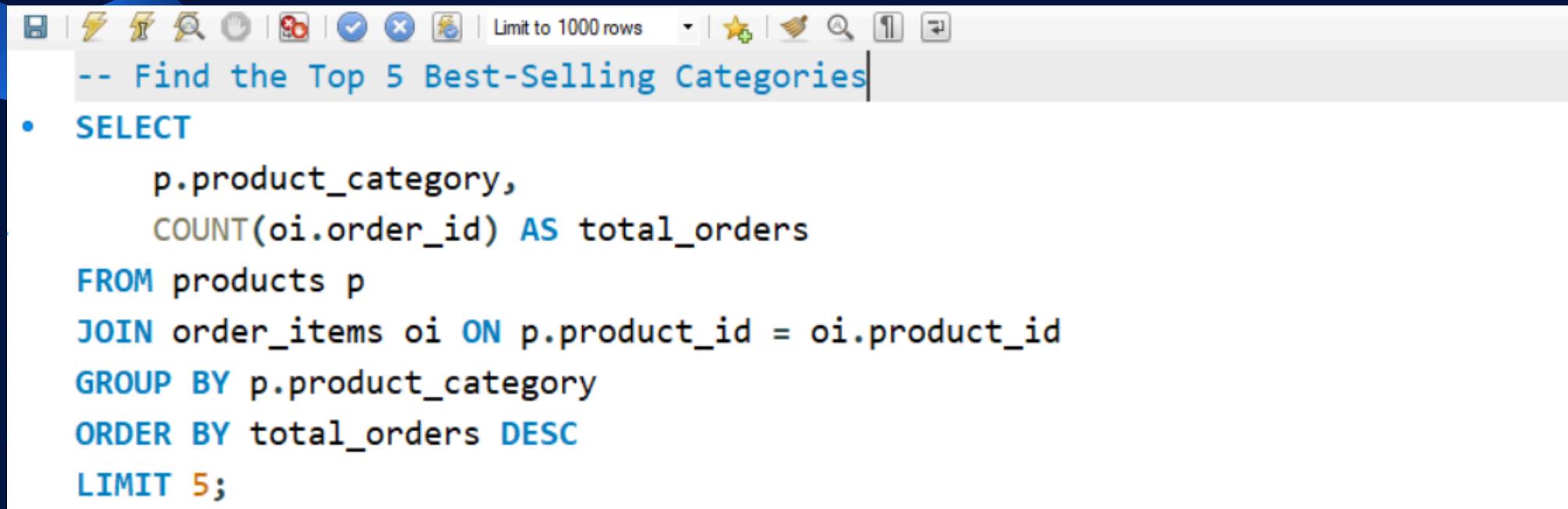
The screenshot shows a MySQL query results window. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows:', 'Export:', 'Wrap Cell Content:', and 'Fetch'. The result grid displays a single row of data:

	month	total_revenue
▶	2017-11	1010271

Key Insights:

- Highest Sales Month: November 2017 recorded the highest sales revenue of 1,010,271.
- Seasonal Sales Trends: This spike in revenue is likely due to major shopping events like Black Friday, which occurs in November.

FIND THE TOP 5 BEST SELLING CATEGORIES



```
-- Find the Top 5 Best-Selling Categories
• SELECT
    p.product_category,
    COUNT(oi.order_id) AS total_orders
  FROM products p
  JOIN order_items oi ON p.product_id = oi.product_id
  GROUP BY p.product_category
  ORDER BY total_orders DESC
  LIMIT 5;
```

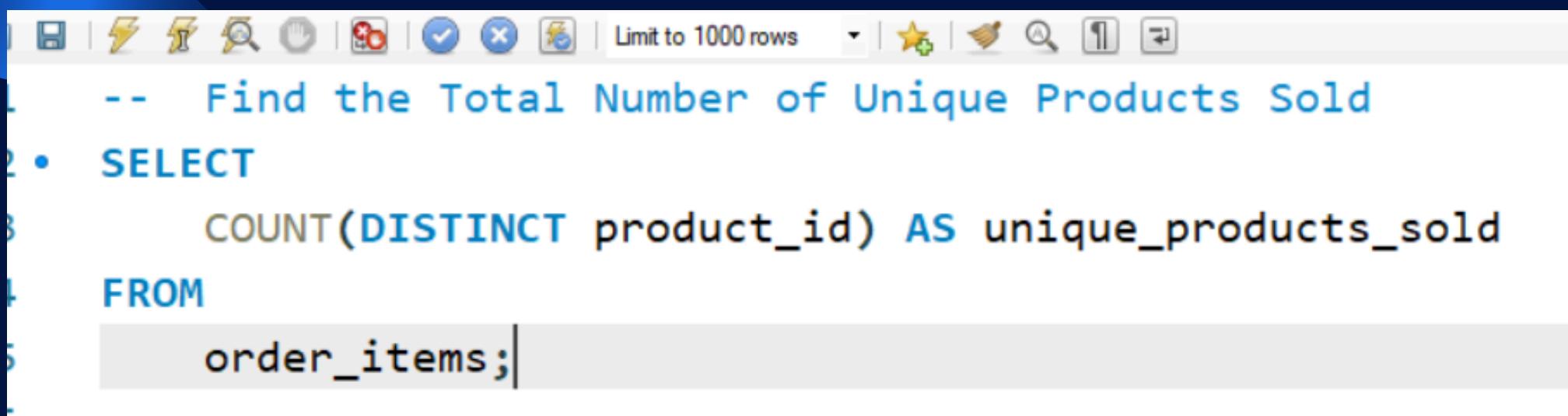


	product_category	total_orders
▶	bed table bath	11115
	HEALTH BEAUTY	9670
	sport leisure	8641
	Furniture Decoration	8334
	computer accessories	7827

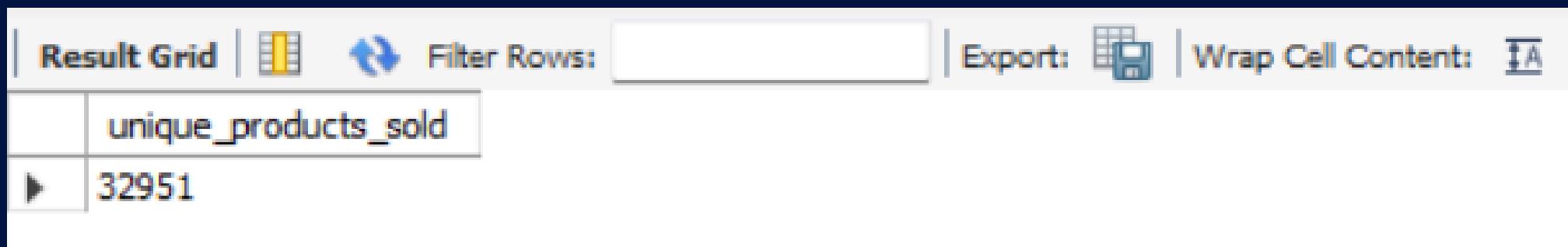
Key Insights:

- Diverse Customer Interests: The top categories span home essentials, personal care, fitness, home décor, and tech accessories, reflecting varied customer needs.
- Home & Lifestyle Dominate: The Bed, Table & Bath and Furniture & Decoration categories suggest high demand for home improvement products.

FIND THE TOTAL NUMBER OF UNIQUE PRODUCTS SOLD



```
-- Find the Total Number of Unique Products Sold
• SELECT
    COUNT(DISTINCT product_id) AS unique_products_sold
FROM
    order_items;
```



unique_products_sold
32951

Key Insights:

- Total Unique Products Sold: 32,951 different products were sold across all orders.
- Diverse Product Catalog: This indicates a wide variety of products being purchased, catering to different customer needs.
- Strong Marketplace Model: A high number of unique products suggests a broad selection from multiple sellers, enhancing customer choices.

CONCLUSION

This SQL-based analysis of Target's e-commerce sales in Brazil (2016-2018) provided valuable insights into customer behavior, sales performance, payment trends, and logistics efficiency. I identified top-selling products, seasonal revenue patterns, and regional market dominance, helping to understand key business trends.



FUTURE SCOPE

- ◆ Advanced Predictive Analytics: Implement machine learning models for demand forecasting.
- ◆ Customer Sentiment Analysis: Incorporate customer reviews to assess satisfaction and improve product offerings.
- ◆ Expansion Opportunities: Explore underperforming regions and categories for market growth.

THANK YOU