Metric/KPI validation

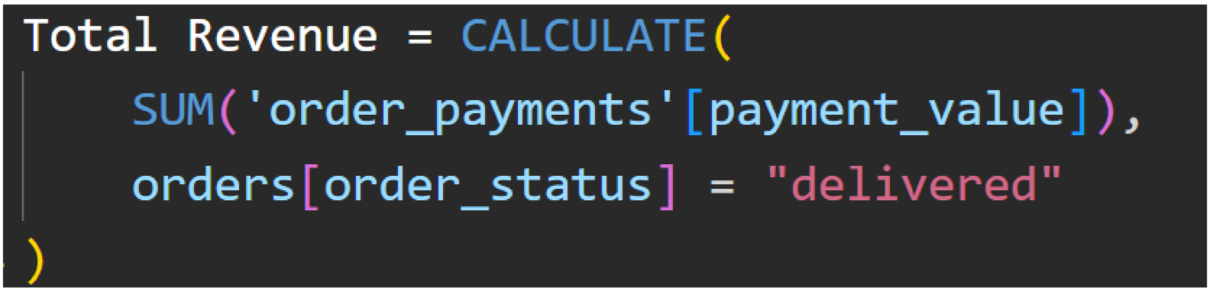
Metric validation, also known as reconciliation, is the process of checking if the numbers shown in reports and dashboards match the actual data stored in the system (such as in SQL databases). This step is important to ensure the insights and figures we share are accurate and trustworthy.

By comparing metrics from visuals with results from SQL queries, we can spot issues like incorrect formulas, wrong filters, or mismatched data joins. Validating metrics helps avoid confusion, builds confidence in the data, and ensures that business decisions are based on correct information.

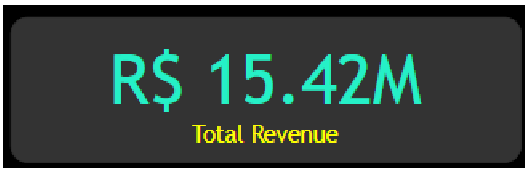
We have compared the outputs of various DAX measures and visuals with the outputs in SQL queries. We were able to successfully reconcile our measures and visuals in the SQL database. This enhances our confidence and trust in our reports and KPIS.

1. **What is the total revenue generated by Olist, and how has it changed over time?**

DAX measure:



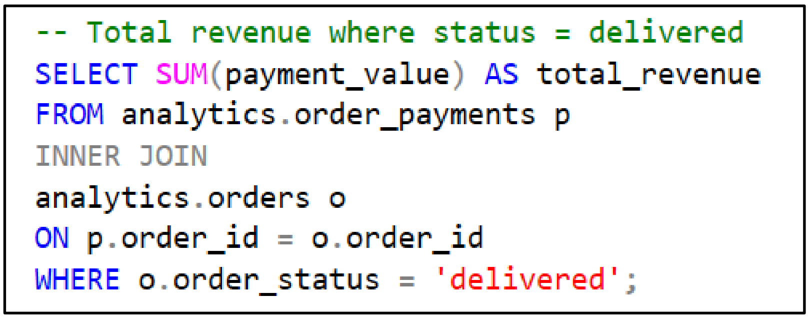
```
Total Revenue = CALCULATE(
    SUM('order_payments'[payment_value]),
    orders[order_status] = "delivered"
)
```



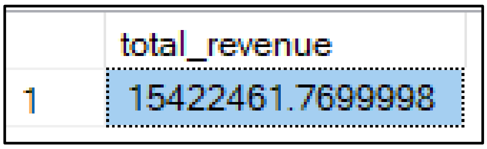
R$ 15.42M

Total Revenue

SQL query:



```
-- Total revenue where status = delivered
SELECT SUM(payment_value) AS total_revenue
FROM analytics.order_payments p
INNER JOIN
analytics.orders o
ON p.order_id = o.order_id
WHERE o.order_status = 'delivered';
```

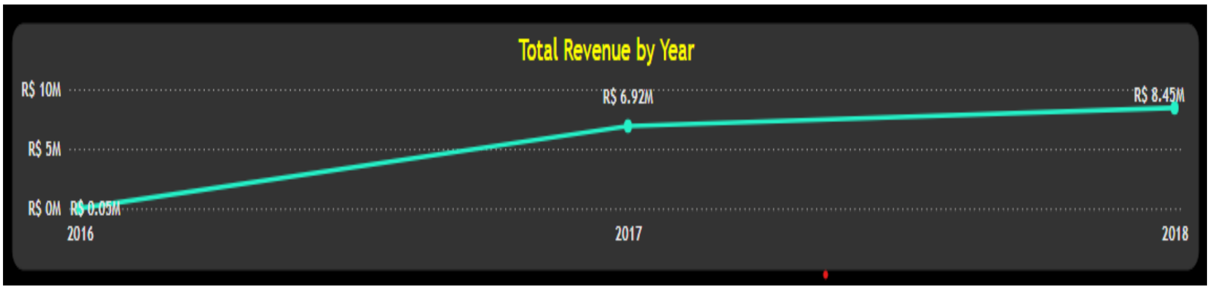


	total_revenue
1	15422461.7699998

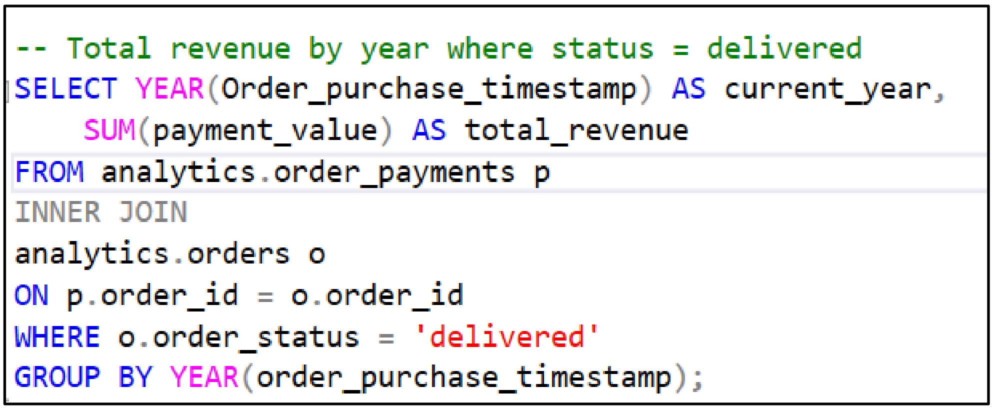
- **We got the same output from DAX measure and SQL query**

Total Revenue by Year:

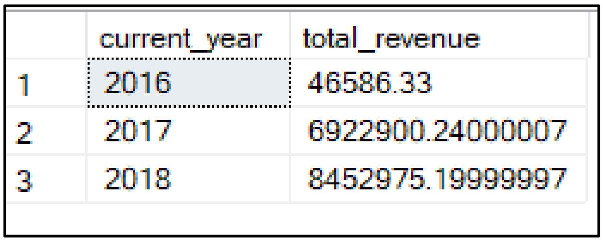
Power BI Visual:



SQL query:



```sql
-- Total revenue by year where status = delivered
SELECT YEAR(Order_purchase_timestamp) AS current_year,
    SUM(payment_value) AS total_revenue
FROM analytics.order_payments p
INNER JOIN
analytics.orders o
ON p.order_id = o.order_id
WHERE o.order_status = 'delivered'
GROUP BY YEAR(order_purchase_timestamp);
```

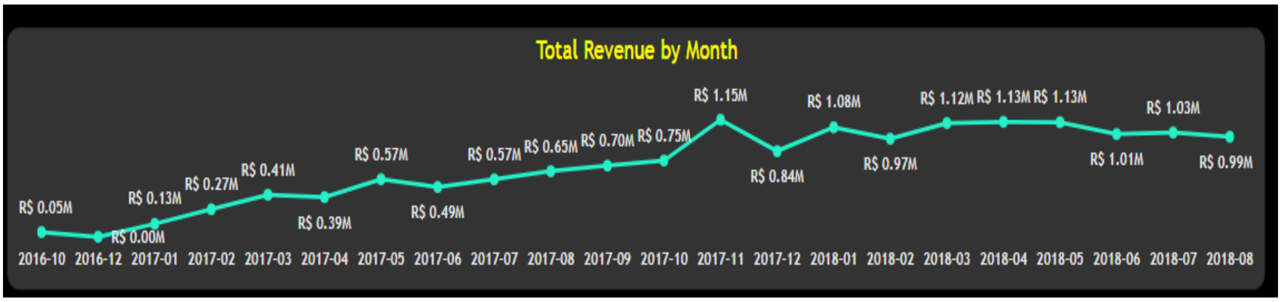


	current_year	total_revenue
1	2016	46586.33
2	2017	6922900.24000007
3	2018	8452975.19999997

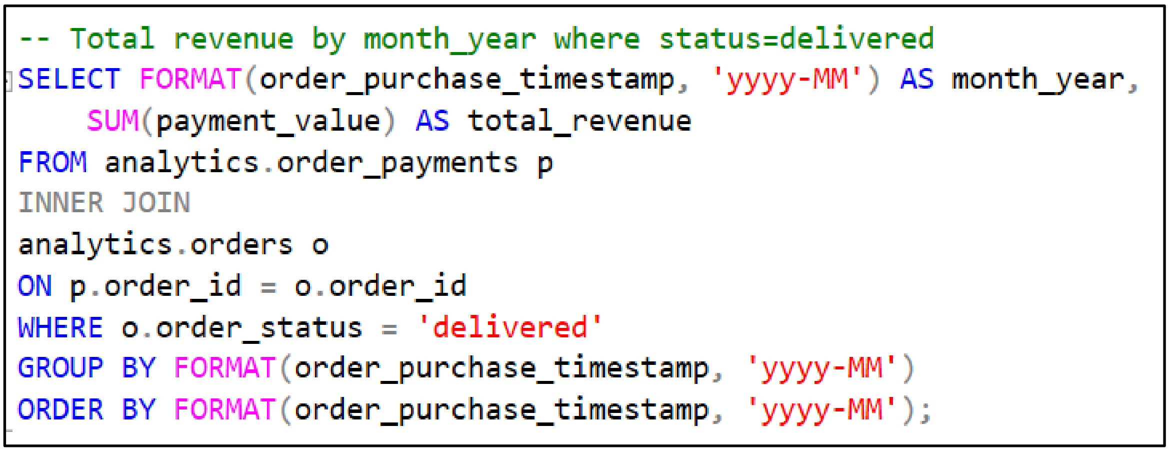
- **We got the same output from Power BI visual and SQL query**

Total Revenue by month:

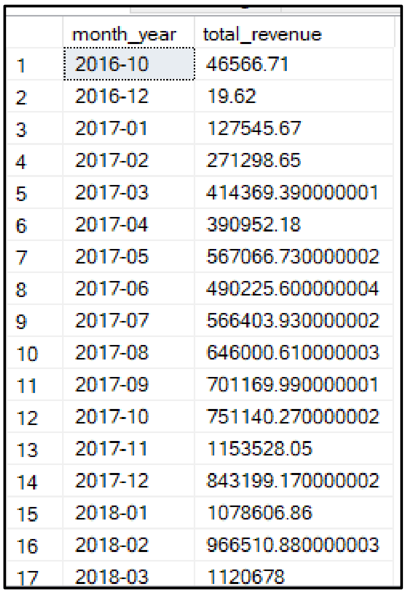
Power BI visual:



SQL query:



```sql
-- Total revenue by month_year where status=delivered
SELECT FORMAT(order_purchase_timestamp, 'yyyy-MM') AS month_year,
    SUM(payment_value) AS total_revenue
FROM analytics.order_payments p
INNER JOIN
analytics.orders o
ON p.order_id = o.order_id
WHERE o.order_status = 'delivered'
GROUP BY FORMAT(order_purchase_timestamp, 'yyyy-MM')
ORDER BY FORMAT(order_purchase_timestamp, 'yyyy-MM');
```

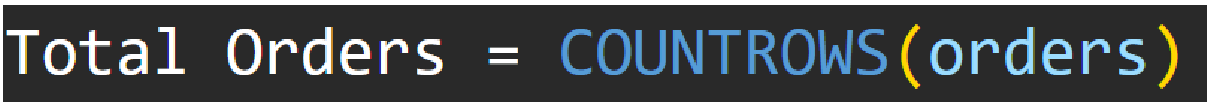


	month_year	total_revenue
1	2016-10	46566.71
2	2016-12	19.62
3	2017-01	127545.67
4	2017-02	271298.65
5	2017-03	414369.390000001
6	2017-04	390952.18
7	2017-05	567066.730000002
8	2017-06	490225.600000004
9	2017-07	566403.930000002
10	2017-08	646000.610000003
11	2017-09	701169.990000001
12	2017-10	751140.270000002
13	2017-11	1153528.05
14	2017-12	843199.170000002
15	2018-01	1078606.86
16	2018-02	966510.880000003
17	2018-03	1120678

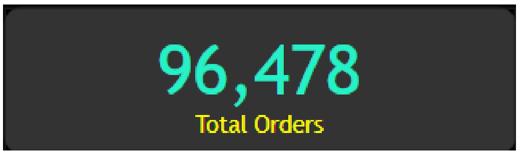
- **We get the same output from our Power BI visual and SQL query**

2. **How many orders were placed on Olist, and how does this vary by month or season?**

DAX Measure:



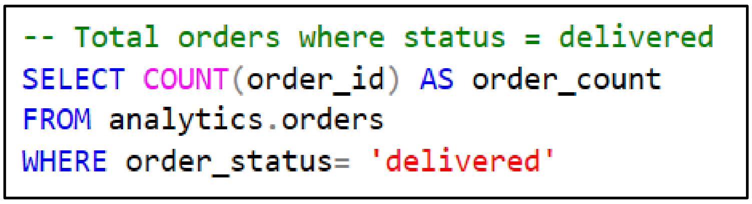
```
Total Orders = COUNTROWS(orders)
```



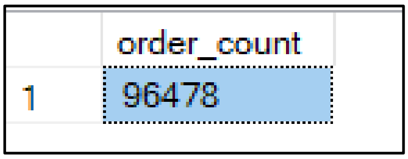
96,478

Total Orders

SQL query:



```
-- Total orders where status = delivered
SELECT COUNT(order_id) AS order_count
FROM analytics.orders
WHERE order_status= 'delivered'
```

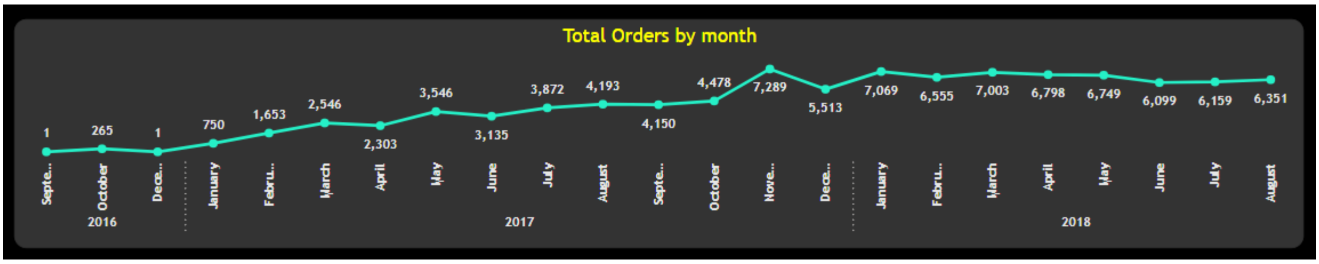


	order_count
1	96478

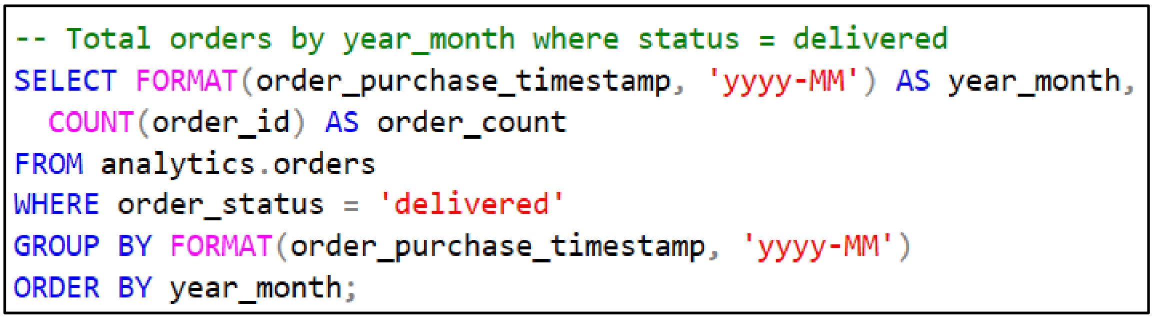
- **We got the same output from DAX measure and SQL query**

Total Orders by Month:

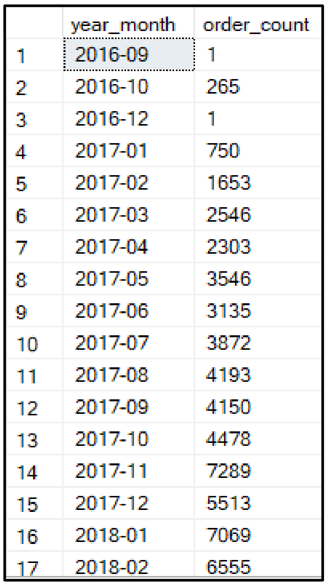
Power BI visual:



SQL query:



```sql
-- Total orders by year_month where status = delivered
SELECT FORMAT(order_purchase_timestamp, 'yyyy-MM') AS year_month,
  COUNT(order_id) AS order_count
FROM analytics.orders
WHERE order_status = 'delivered'
GROUP BY FORMAT(order_purchase_timestamp, 'yyyy-MM')
ORDER BY year_month;
```

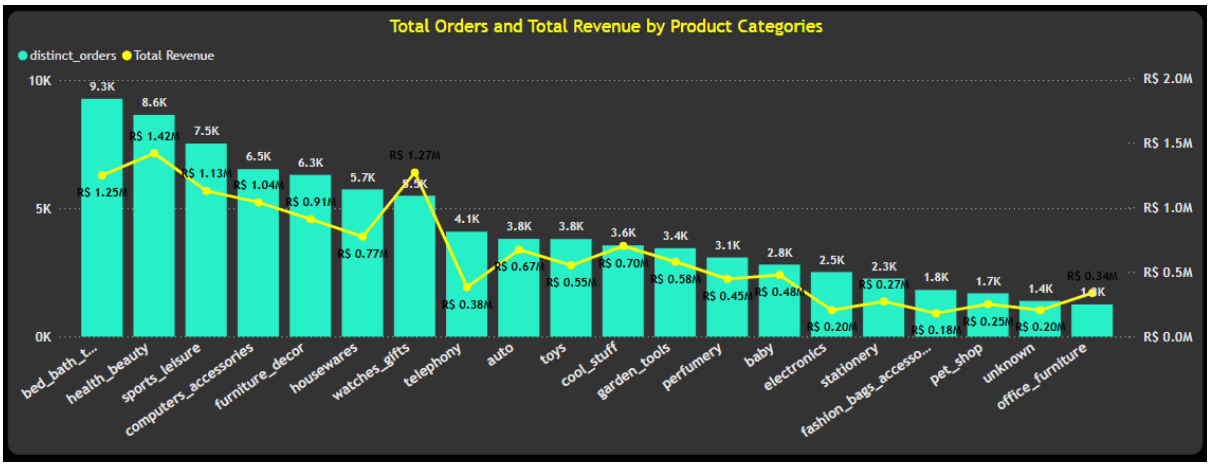


	year_month	order_count
1	2016-09	1
2	2016-10	265
3	2016-12	1
4	2017-01	750
5	2017-02	1653
6	2017-03	2546
7	2017-04	2303
8	2017-05	3546
9	2017-06	3135
10	2017-07	3872
11	2017-08	4193
12	2017-09	4150
13	2017-10	4478
14	2017-11	7289
15	2017-12	5513
16	2018-01	7069
17	2018-02	6555

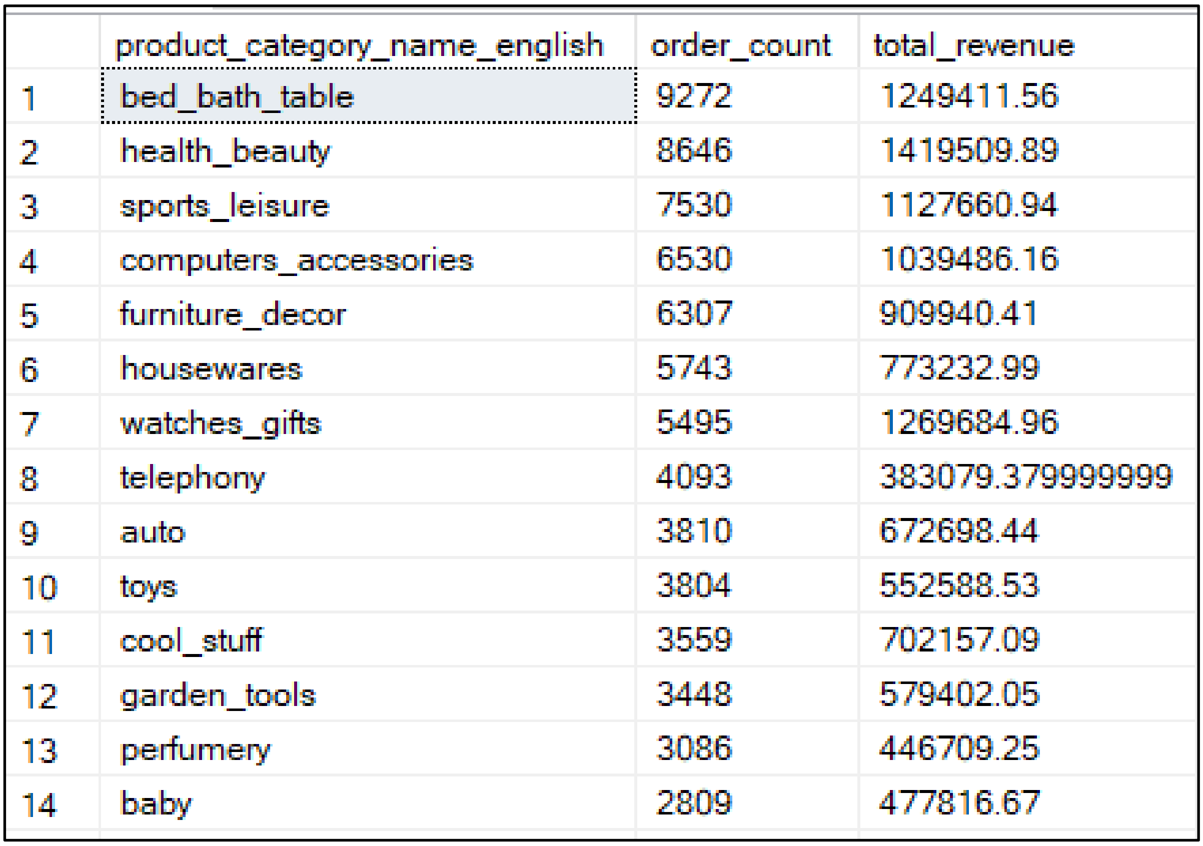
- **We got the same output from Power BI visual and SQL query**

3. What are the most popular product categories on Olist, and how do their sales volumes compare to each other?

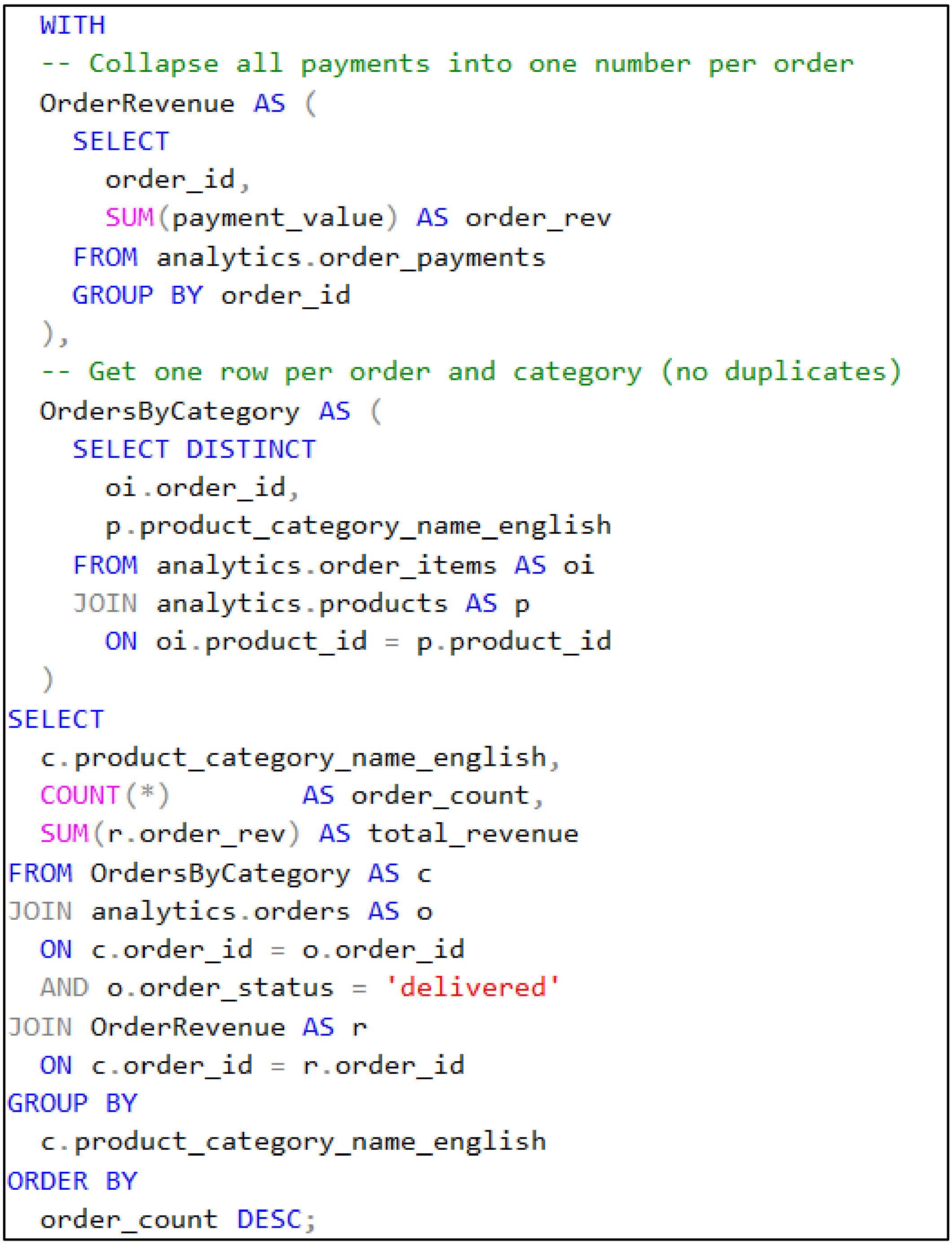
Power BI Visual:



SQL query:



	product_category_name_english	order_count	total_revenue
1	bed_bath_table	9272	1249411.56
2	health_beauty	8646	1419509.89
3	sports_leisure	7530	1127660.94
4	computers_accessories	6530	1039486.16
5	furniture_decor	6307	909940.41
6	housewares	5743	773232.99
7	watches_gifts	5495	1269684.96
8	telephony	4093	383079.379999999
9	auto	3810	672698.44
10	toys	3804	552588.53
11	cool_stuff	3559	702157.09
12	garden_tools	3448	579402.05
13	perfumery	3086	446709.25
14	baby	2809	477816.67

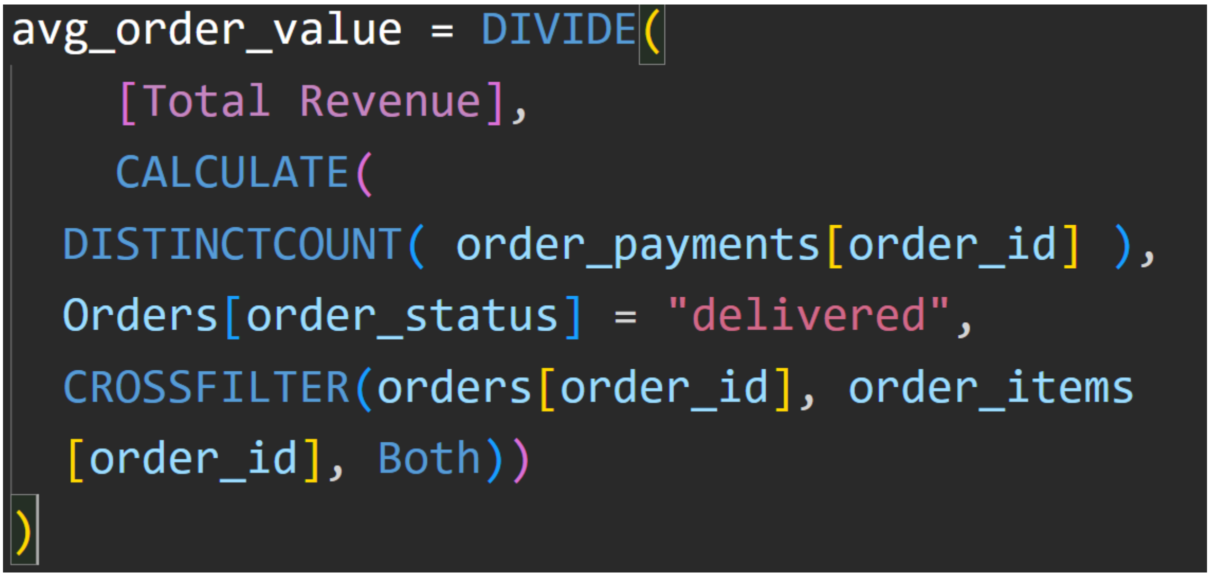


```sql
  WITH
  -- Collapse all payments into one number per order
  OrderRevenue AS (
    SELECT
      order_id,
      SUM(payment_value) AS order_rev
    FROM analytics.order_payments
    GROUP BY order_id
  ),
  -- Get one row per order and category (no duplicates)
  OrdersByCategory AS (
    SELECT DISTINCT
      oi.order_id,
      p.product_category_name_english
    FROM analytics.order_items AS oi
    JOIN analytics.products AS p
      ON oi.product_id = p.product_id
  )
SELECT
  c.product_category_name_english,
  COUNT(*)        AS order_count,
  SUM(r.order_rev) AS total_revenue
FROM OrdersByCategory AS c
JOIN analytics.orders AS o
  ON c.order_id = o.order_id
  AND o.order_status = 'delivered'
JOIN OrderRevenue AS r
  ON c.order_id = r.order_id
GROUP BY
  c.product_category_name_english
ORDER BY
  order_count DESC;
```

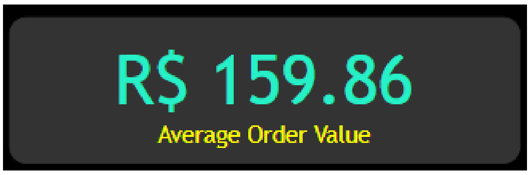
- **We got the same output from Power BI visual and SQL query**

4. **What is the average order value (AOV) on Olist, and how does this vary by product category or payment method?**

DAX measure:



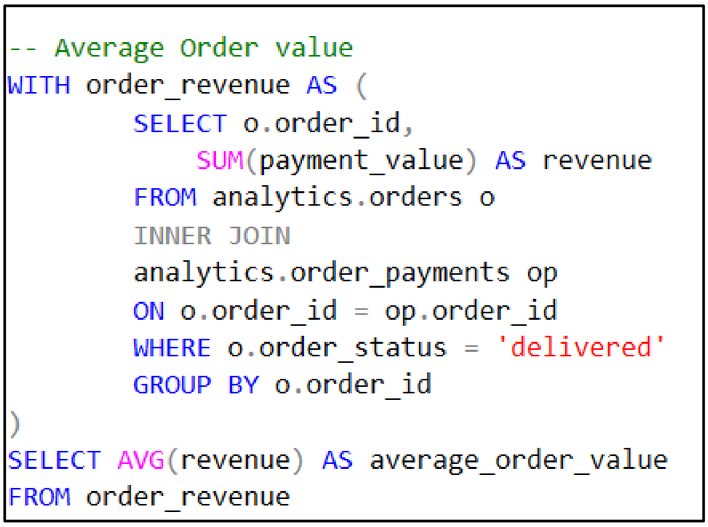
```
avg_order_value = DIVIDE(
     [Total Revenue],
     CALCULATE(
   DISTINCTCOUNT( order_payments[order_id] ),
   Orders[order_status] = "delivered",
   CROSSFILTER(orders[order_id], order_items
   [order_id], Both))
)
```



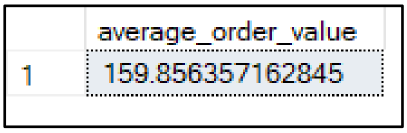
R$ 159.86

Average Order Value

SQL query:



```
-- Average Order value
WITH order_revenue AS (
        SELECT o.order_id,
            SUM(payment_value) AS revenue
        FROM analytics.orders o
        INNER JOIN
        analytics.order_payments op
        ON o.order_id = op.order_id
        WHERE o.order_status = 'delivered'
        GROUP BY o.order_id
)
SELECT AVG(revenue) AS average_order_value
FROM order_revenue
```

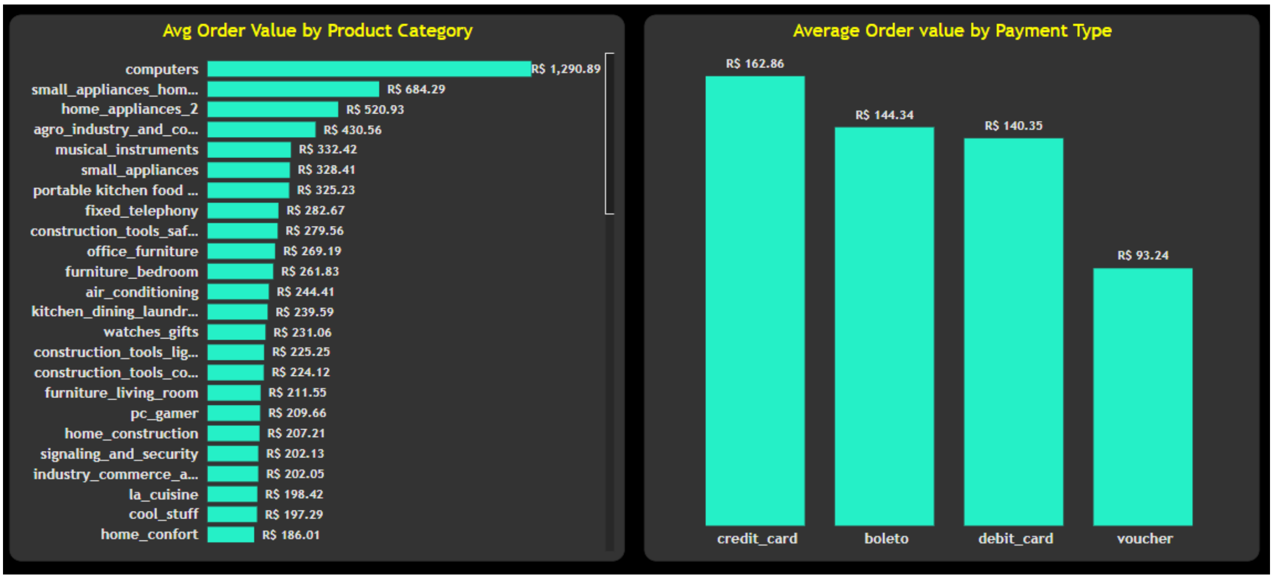


	average_order_value
1	159.856357162845

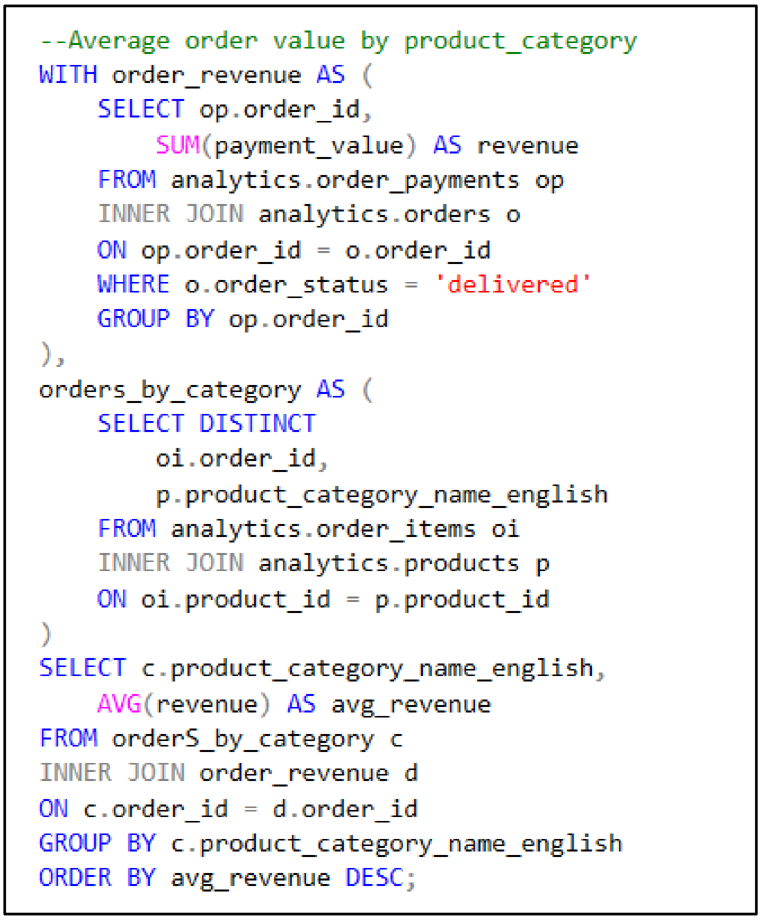
- **We got the same output from DAX measure and SQL query**

Average Order Value by Product Category and Payment Type

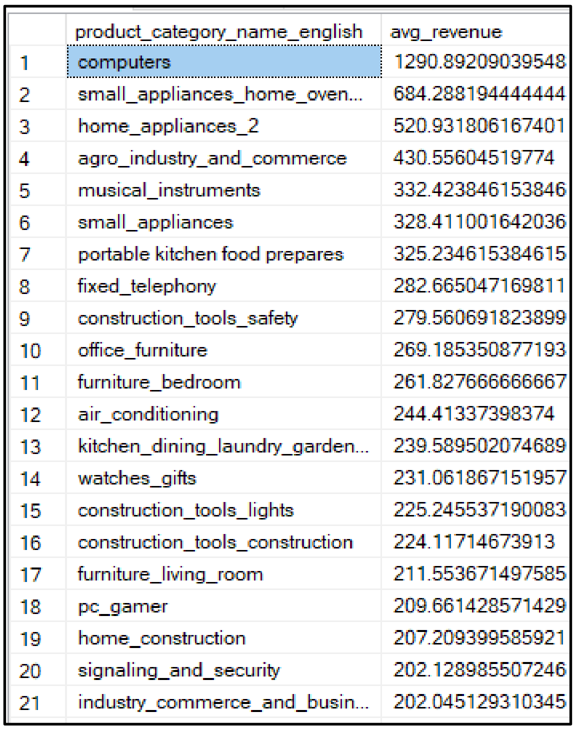
Power BI visual:



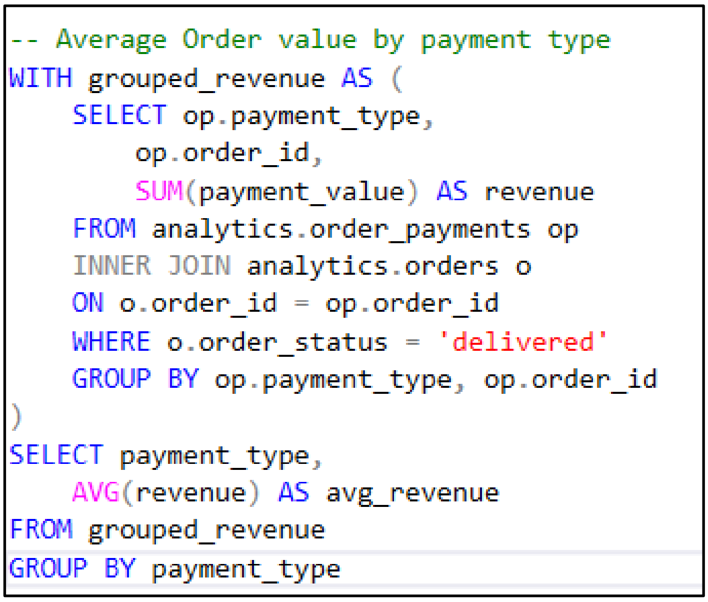
SQL query:



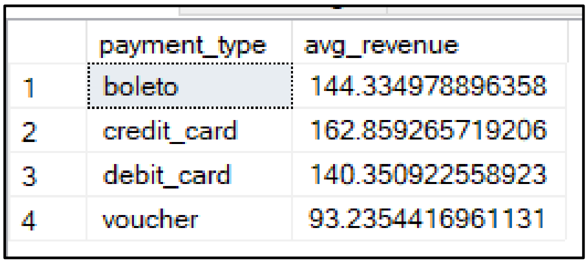
```
--Average order value by product_category
WITH order_revenue AS (
    SELECT op.order_id,
        SUM(payment_value) AS revenue
    FROM analytics.order_payments op
    INNER JOIN analytics.orders o
    ON op.order_id = o.order_id
    WHERE o.order_status = 'delivered'
    GROUP BY op.order_id
),
orders_by_category AS (
    SELECT DISTINCT
        oi.order_id,
        p.product_category_name_english
    FROM analytics.order_items oi
    INNER JOIN analytics.products p
    ON oi.product_id = p.product_id
)
SELECT c.product_category_name_english,
    AVG(revenue) AS avg_revenue
FROM orderS_by_category c
INNER JOIN order_revenue d
ON c.order_id = d.order_id
GROUP BY c.product_category_name_english
ORDER BY avg_revenue DESC;
```



	product_category_name_english	avg_revenue
1	computers	1290.89209039548
2	small_appliances_home_oven...	684.288194444444
3	home_appliances_2	520.931806167401
4	agro_industry_and_commerce	430.55604519774
5	musical_instruments	332.423846153846
6	small_appliances	328.411001642036
7	portable kitchen food prepares	325.234615384615
8	fixed_telephony	282.665047169811
9	construction_tools_safety	279.560691823899
10	office_furniture	269.185350877193
11	furniture_bedroom	261.827666666667
12	air_conditioning	244.41337398374
13	kitchen_dining_laundry_garden...	239.589502074689
14	watches_gifts	231.061867151957
15	construction_tools_lights	225.245537190083
16	construction_tools_construction	224.11714673913
17	furniture_living_room	211.553671497585
18	pc_gamer	209.661428571429
19	home_construction	207.209399585921
20	signaling_and_security	202.128985507246
21	industry_commerce_and_busin...	202.045129310345



```
-- Average Order value by payment type
WITH grouped_revenue AS (
    SELECT op.payment_type,
        op.order_id,
        SUM(payment_value) AS revenue
    FROM analytics.order_payments op
    INNER JOIN analytics.orders o
    ON o.order_id = op.order_id
    WHERE o.order_status = 'delivered'
    GROUP BY op.payment_type, op.order_id
)
SELECT payment_type,
    AVG(revenue) AS avg_revenue
FROM grouped_revenue
GROUP BY payment_type
```

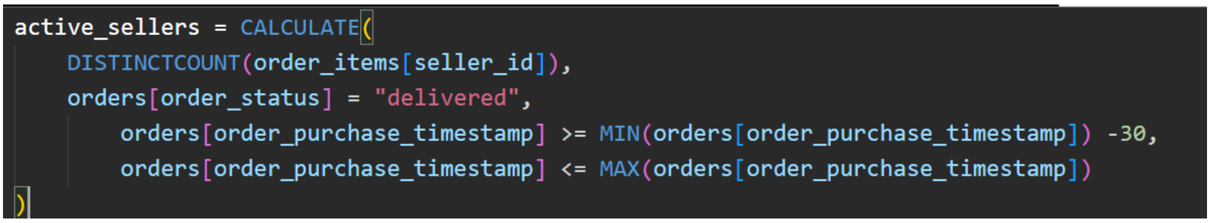


	payment_type	avg_revenue
1	boleto	144.334978896358
2	credit_card	162.859265719206
3	debit_card	140.350922558923
4	voucher	93.2354416961131

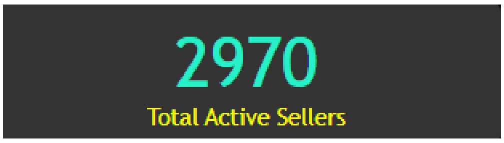
- **We get the same output from the Power BI visual and SQL query**

5. How many sellers are active on Olist?

DAX measure:



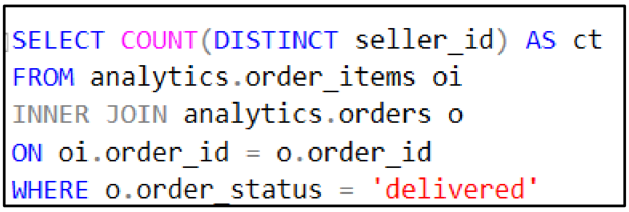
```
active_sellers = CALCULATE(
    DISTINCTCOUNT(order_items[seller_id]),
    orders[order_status] = "delivered",
        orders[order_purchase_timestamp] >= MIN(orders[order_purchase_timestamp]) -30,
        orders[order_purchase_timestamp] <= MAX(orders[order_purchase_timestamp])
)
```



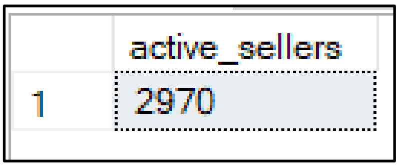
2970

Total Active Sellers

SQL query:



```
SELECT COUNT(DISTINCT seller_id) AS ct
FROM analytics.order_items oi
INNER JOIN analytics.orders o
ON oi.order_id = o.order_id
WHERE o.order_status = 'delivered'
```

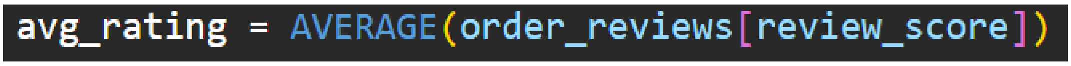


	active_sellers
1	2970

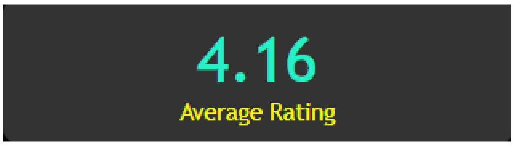
- We got the same output from DAX measure and SQL query

6. What is the average rating?

DAX measure:



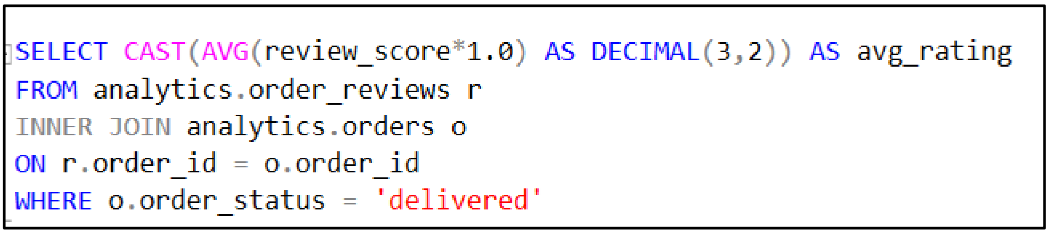
```
avg_rating = AVERAGE(order_reviews[review_score])
```



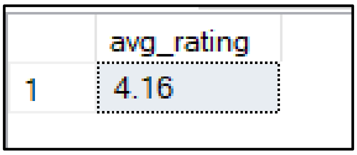
4.16

Average Rating

***SQL query*:**



```
SELECT CAST(AVG(review_score*1.0) AS DECIMAL(3,2)) AS avg_rating
FROM analytics.order_reviews r
INNER JOIN analytics.orders o
ON r.order_id = o.order_id
WHERE o.order_status = 'delivered'
```

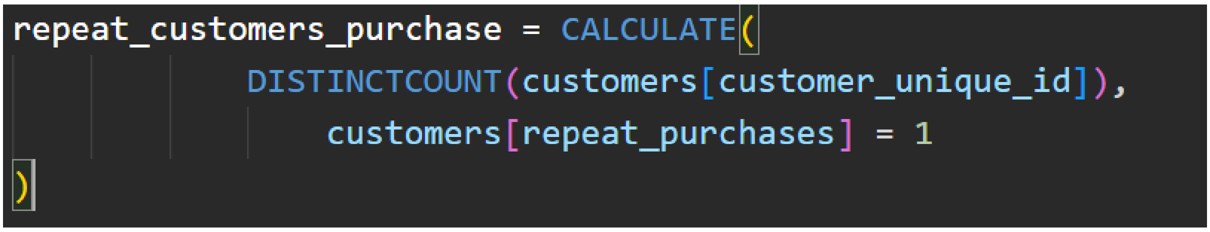


	avg_rating
1	4.16

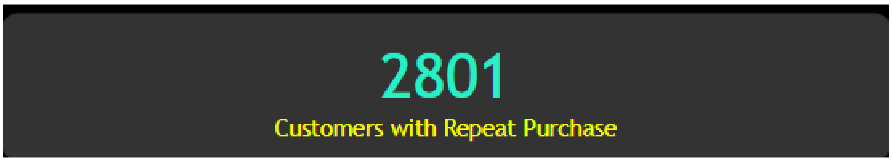
- **We got the same output from DAX measure and SQL query**

7. How many customers have made repeat purchases on Olist?

DAX measure:



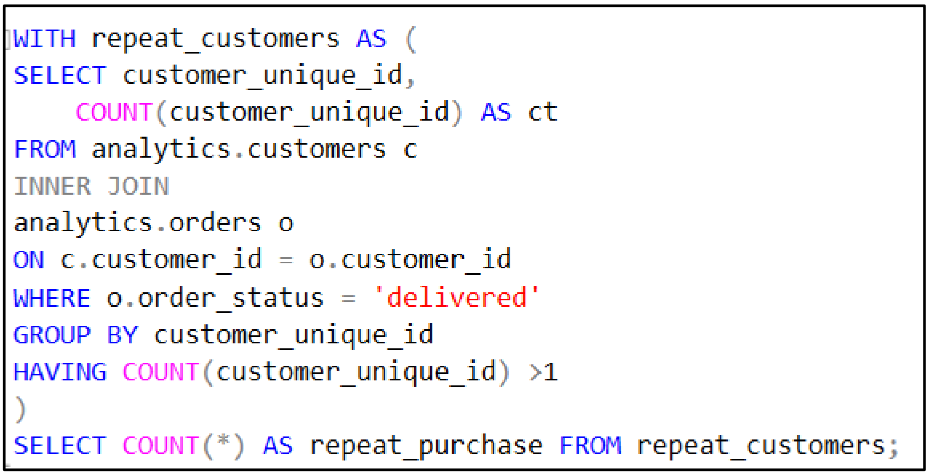
```
repeat_customers_purchase = CALCULATE(
                DISTINCTCOUNT(customers[customer_unique_id]),
                    customers[repeat_purchases] = 1
)
```



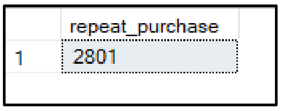
2801

Customers with Repeat Purchase

SQL query:



```
WITH repeat_customers AS (
SELECT customer_unique_id,
    COUNT(customer_unique_id) AS ct
FROM analytics.customers c
INNER JOIN
analytics.orders o
ON c.customer_id = o.customer_id
WHERE o.order_status = 'delivered'
GROUP BY customer_unique_id
HAVING COUNT(customer_unique_id) >1
)
SELECT COUNT(*) AS repeat_purchase FROM repeat_customers;
```

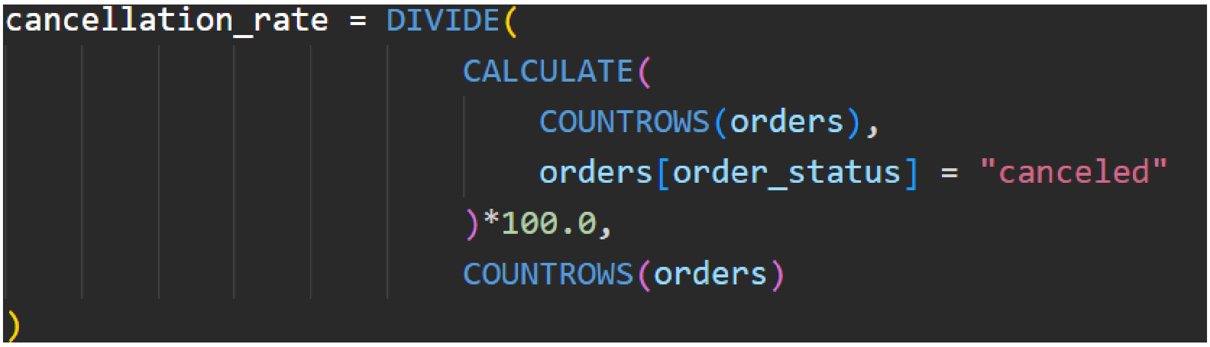


	repeat_purchase
1	2801

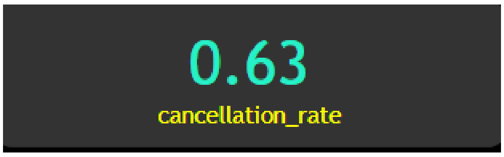
- **We got the same output from DAX measure and SQL query**

8. What is the average order cancellation rate on Olist?

DAX measure:



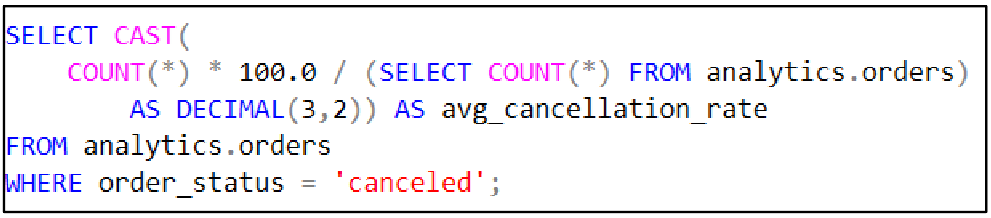
```
cancellation_rate = DIVIDE(
                        CALCULATE(
                            COUNTROWS(orders),
                            orders[order_status] = "canceled"
                        )*100.0,
                        COUNTROWS(orders)
)
```



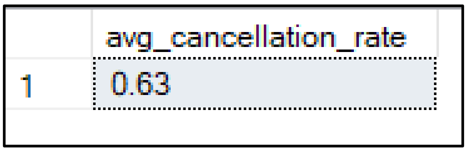
0.63

cancellation_rate

SQL query:



```
SELECT CAST(
    COUNT(*) * 100.0 / (SELECT COUNT(*) FROM analytics.orders)
        AS DECIMAL(3,2)) AS avg_cancellation_rate
FROM analytics.orders
WHERE order_status = 'canceled';
```

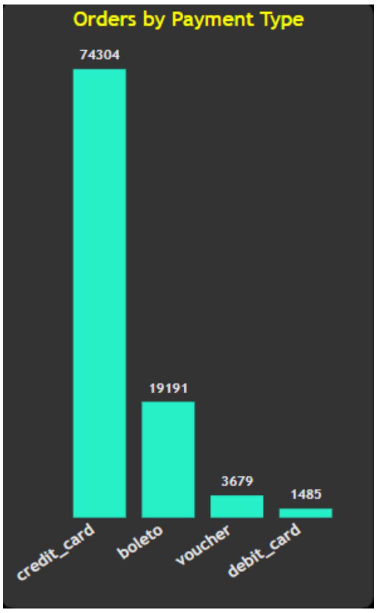


	avg_cancellation_rate
1	0.63

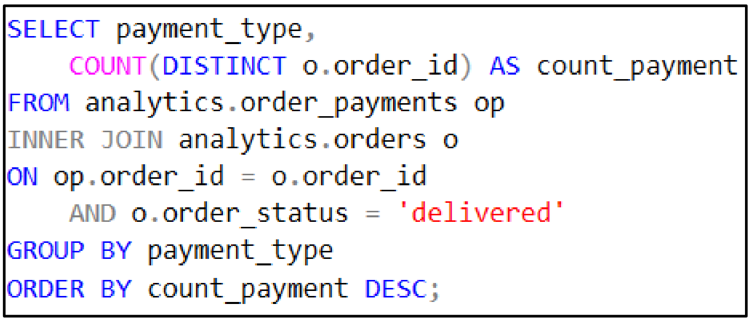
- We got the same output from DAX measure and SQL query

9. Which payment methods are most commonly used by Olist customers?

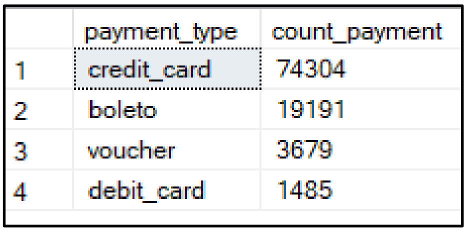
Power BI visual:



SQL query:



```
SELECT payment_type,
    COUNT(DISTINCT o.order_id) AS count_payment
FROM analytics.order_payments op
INNER JOIN analytics.orders o
ON op.order_id = o.order_id
    AND o.order_status = 'delivered'
GROUP BY payment_type
ORDER BY count_payment DESC;
```



	payment_type	count_payment
1	credit_card	74304
2	boleto	19191
3	voucher	3679
4	debit_card	1485

- **We got the same output from Power BI visual and SQL query**