09/30/2024

Data Analytics Project Using SQL | E-commerce Dataset

Introduction

kaggle

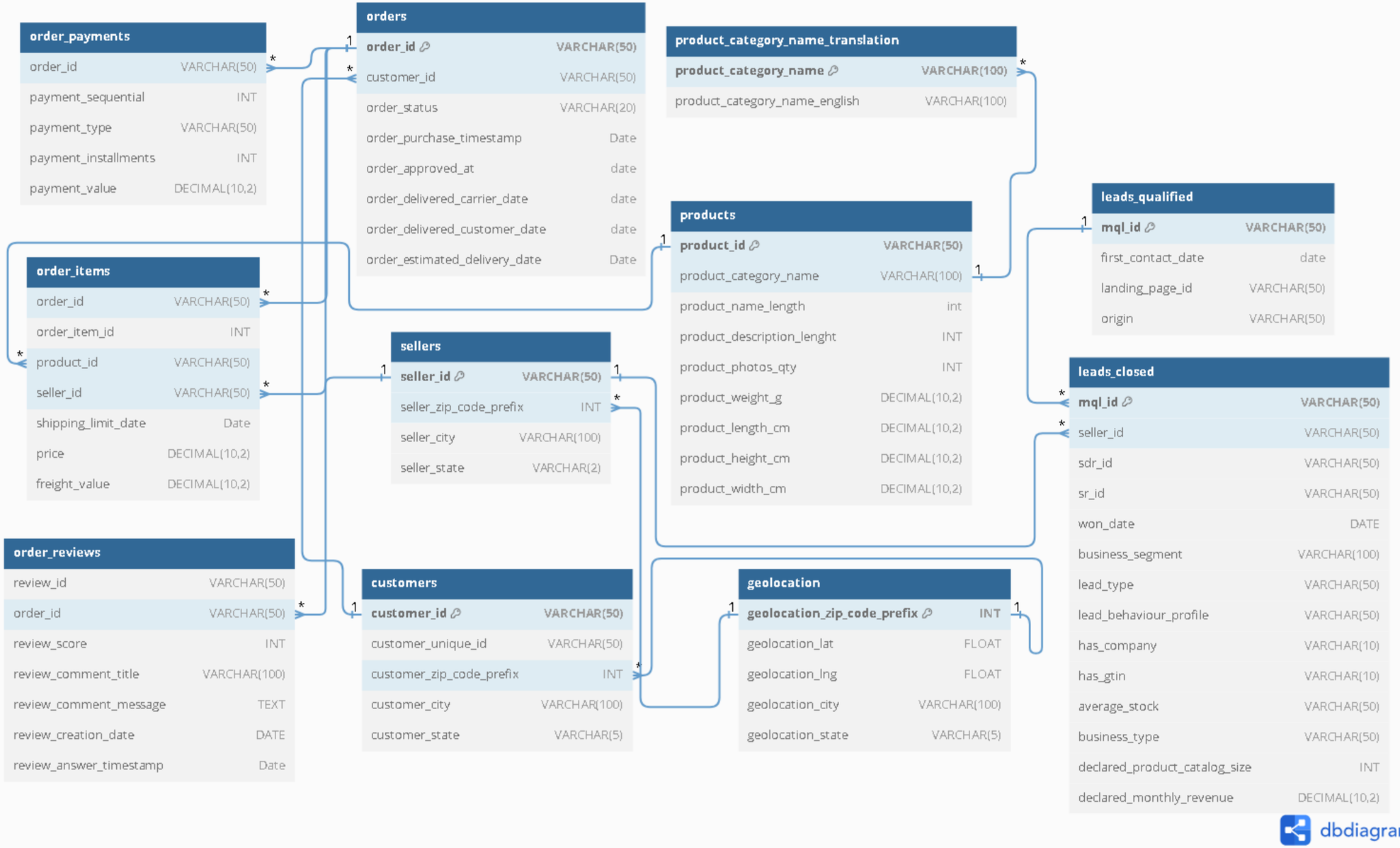
This project is a self-initiated endeavor that utilizes skills in SQL for data analysis. The data used is sourced from public datasets (e-commerce-dataset-by-olist-as-an-sqlite-database) on Kaggle. Thank you for viewing, and I welcome any suggestions for improvement

About The Dataset

The datasets provided is in two different zip files , one containing commercial transaction data, including information about customers, employees, product details, and order quantities and the other containing marketing leads and successful lead information of an e-commerce company operating in Brazil in different csv files. The entire dataset consists of 11 tables as follows:

1.Customers, 2.geolocation ,3.leads_closed,
4.leads_qualified,
5.order_items, 6.order_payments, 7.order_reviews,
8.orders,
9. product_category_name_translation, 10.products,
11.sellers

After briefly reviewing our dataset, I created an ERdiagram to facilitate the querying process with SQL which is shown in the next slide. I entered the data and built the subsequent model in SQL using the pgAdmin app, which is known for managing PostgreSQL databases.


order_payments
order_id VARCHAR(50)
payment_sequential INT
payment_type VARCHAR(50)
payment_installments INT
payment_value DECIMAL(10,2)
orders
order_id VARCHAR(50)
customer_id VARCHAR(50)
order_status VARCHAR(20)
order_purchase_timestamp Date
order_approved_at date
order_delivered_carrier_date date
order_delivered_customer_date date
order_estimated_delivery_date Date
product_category_name_translation
product_category_name VARCHAR(100)
product_category_name_english VARCHAR(100)
leads_qualified
mql_id VARCHAR(50)
first_contact_date date
landing_page_id VARCHAR(50)
origin VARCHAR(50)
products
product_id VARCHAR(50)
product_category_name VARCHAR(100)
product_name_length int
product_description_lenght INT
product_photos_qty INT
product_weight_g DECIMAL(10,2)
product_length_cm DECIMAL(10,2)
product_height_cm DECIMAL(10,2)
product_width_cm DECIMAL(10,2)
order_items
order_id VARCHAR(50)
order_item_id INT
product_id VARCHAR(50)
seller_id VARCHAR(50)
shipping_limit_date Date
price DECIMAL(10,2)
freight_value DECIMAL(10,2)
sellers
seller_id VARCHAR(50)
seller_zip_code_prefix INT
seller_city VARCHAR(100)
seller_state VARCHAR(2)
leads_closed
mql_id VARCHAR(50)
seller_id VARCHAR(50)
sdr_id VARCHAR(50)
sr_id VARCHAR(50)
won_date DATE
business_segment VARCHAR(100)
lead_type VARCHAR(50)
lead_behaviour_profile VARCHAR(50)
has_company VARCHAR(10)
has_gtin VARCHAR(10)
average_stock VARCHAR(50)
business_type VARCHAR(50)
declared_product_catalog_size INT
declared_monthly_revenue DECIMAL(10,2)
order_reviews
review_id VARCHAR(50)
order_id VARCHAR(50)
review_score INT
review_comment_title VARCHAR(100)
review_comment_message TEXT
review_creation_date DATE
review_answer_timestamp Date
customers
customer_id VARCHAR(50)
customer_unique_id VARCHAR(50)
customer_zip_code_prefix INT
customer_city VARCHAR(100)
customer_state VARCHAR(5)
geolocation
geolocation_zip_code_prefix INT
geolocation_lat FLOAT
geolocation_lng FLOAT
geolocation_city VARCHAR(100)
geolocation_state VARCHAR(5)
dbdiagram.io

Customer Metrics

Customer Lifetime Value (CLV):-- Sum of the total order values per customer (top 10)

This query identifies top 10 customers of Olist

```
SELECT customer_id, CONCAT('R$ ', SUM(payment_value)) AS
total_order_value
 FROM  orders  JOIN
order_payments
ON orders.order_id = order_payments.order_id
GROUP BY customer_id
Order By SUM(payment_value) desc LIMIT 10;
```

Result:

customer_id character varying (50)	total_order_value text
1617b1357756262bfa56ab541c47bc16	R$ 13664.08
ec5b2ba62e574342386871631fafd3fc	R$ 7274.88
c6e2731c5b391845f6800c97401a43a9	R$ 6929.31
f48d464a0baaea338cb25f816991ab1f	R$ 6922.21
3fd6777bbce08a352fddd04e4a7cc8f6	R$ 6726.66
05455dfa7cd02f13d132aa7a6a9729c6	R$ 6081.54
df55c14d1476a9a3467f131269c2477f	R$ 4950.34
e0a2412720e9ea4f26c1ac985f6a7358	R$ 4809.44
24bbf5fd2f2e1b359ee7de94defc4a15	R$ 4764.34
3d979689f636322c62418b6346b1c6d2	R$ 4681.78

Customer Metrics

Average Order Value (AOV)

This query calculates the average value of an order

```
SELECT
	CONCAT('R$ ', ROUND(SUM(payment_value) / COUNT(DISTINCT
orders.order_id), 2)) AS average_order_value
FROM  orders
JOIN order_payments
 ON  orders.order_id = order_payments.order_id;
```

Result:

average_order_value text
R$ 160.99

Customer Metrics

Customer Segmentation by Location

This query groups customers based on their location (customer_city), and shows the total order count per location.

<u>top 10 cities by number of orders-</u>

```
SELECT
  UPPER(customer_city) AS city,
  COUNT(orders.order_id) as city_order_count
FROM    customers
 JOIN orders
USING (customer_id)
GROUP BY customer_city
ORDER BY city_order_count DESC LIMIT 10;
```

Result:

	city text	city_order_count bigint
1	SAO PAULO	15540
2	RIO DE JANEIRO	6882
3	BELO HORIZONTE	2773
4	BRASILIA	2131
5	CURITIBA	1521
6	CAMPINAS	1444
7	PORTO ALEGRE	1379
8	SALVADOR	1245
9	GUARULHOS	1189
10	SAO BERNARDO DO CAMPO	938

Customer Metrics

Customer Segmentation by Location

This query groups customers based on their location (customer_state), and shows the total order count per location.

<u>top 10 states by number of orders-</u>

```
SELECT
 UPPER(customer_state) AS state,
 COUNT(orders.order_id) as city_order_count FROM
 customers   JOIN
orders
USING (customer_id)
GROUP BY customer_state
ORDER BY city_order_count DESC LIMIT 10;
```

Result:

	state text	states_order_count bigint
1	SP	41746
2	RJ	12852
3	MG	11635
4	RS	5466
5	PR	5045
6	SC	3637
7	BA	3380
8	DF	2140
9	ES	2033
10	GO	2020

Customer Insights

Top Customer (by spending)

Identified top customers on the platform and surprisingly they all have spent more than R$ 4000 reaching upto R$13000

Average Customer Spending

Average customer spending on th platform is R$ 161 which is pretty descent. Still Olist can make efforts increase this as much they can.

Customer segmentation regionally

Sao Paulo is the top city and region in customer spending followed by Rio de Generio and Bello Horizonte among cities and Rio de Generio and Minas Gerais among states.

Order Metrics

Total Orders

This query calculates total number of orders placed using Olist -

```
SELECT
 COUNT(order_id) AS total_orders
FROM     orders;
```

Result:

total_orders
bigint
99441

Order Metrics

Order Status Breakdown

This query returns the distribution of orders based on their status (e.g., delivered, pending, canceled)-

```
SELECT
 order_status, COUNT(order_id) AS order_count
FROM orders
GROUP BY order_status
 ORDER BY order_count DESC;
```

Result:

	order_status character varying (20)	order_count bigint
1	delivered	96478
2	shipped	1107
3	canceled	625
4	unavailable	609
5	invoiced	314
6	processing	301
7	created	5
8	approved	2

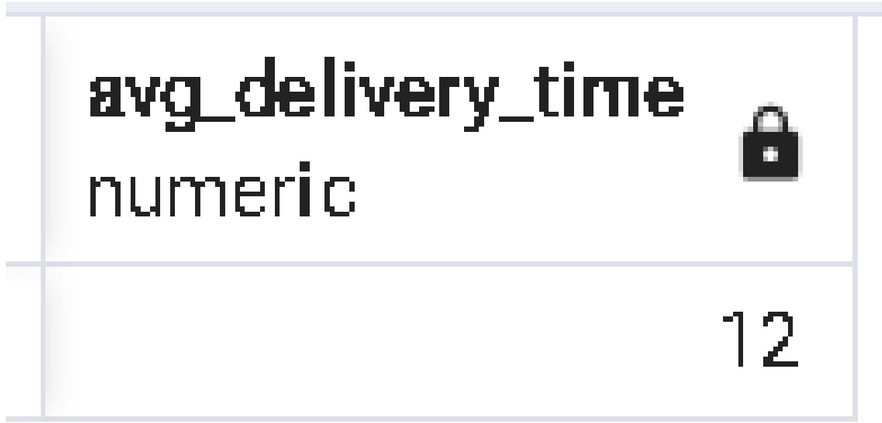
Order Metrics

Order Delivery Time

This query calculates the average time between the order placement and delivery to the customer (in days)

```
SELECT
Round(AVG(order_delivered_customer_date -
order_purchase_timestamp),0) AS avg_delivery_time
FROM orders
WHERE  order_delivered_customer_date IS NOT NULL;
```

Result:



Order Insights

Total Orders

Olist have generated nearly 100k orders in three years mostly from 2017-18 . among which more than 96k orders have been delivered successfully proving the delivery efficiency of the platform.

Average Delivery time

Average delivery time on the platform is around 12 days which is a bit longer than it's competiotors for sure. Olist can hire more delivery partners and make more delivery godowns to improve the delivery time.

Delivery Efficiency

Among nearly 100k orders , more than 96k orders have been delivered successfully proving the delivery efficiency of the platform.

Product Metrics

Top-Selling Products

This query returns the products with the highest number of sales, sorted in descending order.

```
SELECT
oi.product_id,
COUNT(oi.product_id) AS total_sales, pc.product_category_name_English
FROM order_items  oi
JOIN products  p   ON p.product_id = oi.product_id
JOIN product_category_name_translation  pc
ON pc.product_category_name = p.product_category_name
GROUP BY oi.product_id, pc.product_category_name_English
ORDER BY  total_sales DESC;
```

Result:

	product_id character varying (50)	total_sales bigint	product_category_name_english character varying (100)
1	aca2eb7d00ea1a7b8ebd4e68314663af	527	furniture_decor
2	99a4788cb24856965c36a24e339b6058	488	bed_bath_table
3	422879e10f46682990de24d770e7f83d	484	garden_tools
4	389d119b48cf3043d311335e499d9c6b	392	garden_tools
5	368c6c730842d78016ad823897a372db	388	garden_tools
6	53759a2ecddad2bb87a079a1f1519f73	373	garden_tools
7	d1c427060a0f73f6b889a5c7c61f2ac4	343	computers_accessories
8	53b36df67ebb7c41585e8d54d6772e08	323	watches_gifts
9	154e7e31ebfa092203795c972e5804a6	281	health_beauty
10	3dd2a17168ec895c781a9191c1e95ad7	274	computers_accessories
11	2b4609f8948be18874494203496bc318	260	health_beauty
12	7c1bd920dbdf22470b68bde975dd3ccf	231	health_beauty
13	a62e25e09e05e6faf31d90c6ec1aa3d1	226	watches_gifts
14	bb50f2e236e5eea0100680137654686c	195	health_beauty
15	e0d64dcfaa3b6db5c54ca298ae101d05	194	watches_gifts
16	42a2c92a0979a949ca4ea89ec5c7b934	183	housewares
17	e53e557d5a159f5aa2c5e995dfdf244b	183	computers_accessories
18	b532349fe46b38fbc7bb3914c1bdae07	169	furniture_decor
19	35afc973633aaeb6b877ff57b2793310	165	home_confort
20	a92930c327948861c015c919a0bcb4a8	160	watches_gifts
21	6cdd53843498f92890544667809f1595	156	health_beauty

***Note: Only a portion of the table is shown.**

Product Metrics

Average Product Rating

This query calculates the average review score for each product

```
SELECT
   oi.product_id,
   AVG(review_score) AS avg_rating,
   pc.product_category_name_english AS Category
FROM
   order_reviews rev
JOIN
   order_items oi ON rev.order_id = oi.order_id
JOIN
   products p ON p.product_id = oi.product_id
JOIN
   product_category_name_translation pc ON pc.product_category_name =
p.product_category_name
GROUP BY
   oi.product_id, Category
ORDER BY
   avg_rating DESC;
```

Result:

	product_id character varying (50)	avg_rating numeric	category character varying (100)
8367	a4f283ca7a807cf3ff056abf3ca9a621	5.0000000000000000	housewares
8368	a11b842166a17bc29bcdcb3cb45a50e3	5.0000000000000000	health_beauty
8369	ef9f731f8f63f85ea53e94df375baf54	5.0000000000000000	musical_instruments
8370	f7285f0b3043d966ede4c21886b49a38	5.0000000000000000	cool_stuff
8371	66f01f544ba693e1a1a7428caec79279	5.0000000000000000	furniture_decor
8372	930987da9df8e7d52d8072b5d7de5f6b	5.0000000000000000	garden_tools
8373	b7f127a1a10296074245d22db5f6d386	5.0000000000000000	baby
8374	1e771d5cef909641314a6b1544656a8d	5.0000000000000000	cool_stuff
8375	c72d57da8ba2412527531d97106eb8ef	5.0000000000000000	telephony
8376	c2c4115f38ec8f43e1052cf0735e289b	5.0000000000000000	furniture_decor
8377	543cfc1fc80d636c64df24e8545818b9	5.0000000000000000	dvds_blu_ray
8378	7766894470ea995b418764065e6bf9ba	5.0000000000000000	housewares
8379	3a3fb4bcc28c99c4b0dc6af206e2744d	5.0000000000000000	sports_leisure
8380	61e8dfe8d80294860f85e9f5cea7cceb	5.0000000000000000	auto
8381	d56781dd9e632e4460c10d95167e5840	5.0000000000000000	sports_leisure
8382	e31cf1512c0473f66814000fbc9ad337	5.0000000000000000	sports_leisure
8383	4e3338174342c9d868255b062718111e	5.0000000000000000	auto
8384	7e07a8d0f0ea4d119dae448cf1b595ec	5.0000000000000000	pet_shop
8385	ffe013e1b4603e3b0b02fbb159d5b400	5.0000000000000000	sports_leisure

***Note: Only a portion of the table is shown.**

Product Metrics

Product Return Rate

This query calculates the percentage of products that were returned or refunded.

```
WITH total_orders AS (
    SELECT product_id, COUNT(order_id) AS total_orders
    FROM order_items
    GROUP BY product_id
),
cancelled_orders AS (
    SELECT oi.product_id, COUNT(o.order_id) AS cancelled_orders
    FROM order_items oi
    JOIN orders o ON oi.order_id = o.order_id
    WHERE order_status = 'canceled' OR order_status = 'unavailable'
    GROUP BY oi.product_id
)
SELECT
    t.product_id,
    CONCAT(ROUND(r.cancelled_orders * 100.0 / t.total_orders, 2), '%') AS cancellation_rate
FROM
    total_orders t
LEFT JOIN
    cancelled_orders r ON t.product_id = r.product_id
WHERE
    (r.cancelled_orders * 100.0 / t.total_orders) > 0
ORDER BY
    (r.cancelled_orders * 100.0 / t.total_orders) DESC;
```

Result:

	product_id character varying (50)	cancellation_rate text
1	4ef0124968bdc099c9a992a37bcb4155	100.00%
2	db91d5a3b20ec0804d7608ab608bea95	100.00%
3	f8ccd11a8dd63145c9a4ad4424195c77	100.00%
4	e4fe462cce9f36e312b869418a86bc3a	100.00%
5	3af9d305a8389f5badc664382683a532	100.00%
6	3f6f946481fd39f4eda986012f6e0447	100.00%
7	41c1f03b4d5bac6d41d5f9d2a85389c6	100.00%
8	20dce72985857d1c8d09d506a8ec5187	100.00%
9	389c7d7f59a980be2afbe84d648cd80f	100.00%
10	6371d60c329c7474f4190b077378660e	100.00%
11	bf7727705a701b9c135cb00df5a2dc5d	100.00%
12	8510c0493319d7d46949a52df53e3d48	100.00%
13	9e86427a5a9119af3ad32f27ccd9df52	100.00%
14	63a033b030ff3da0b4b4eb3043c503d1	100.00%

***Note: Only a portion of the table is shown.**

Product Metrics

High Value Orders Product Category wise

This query calculates high value Orders product category wise that was ordered through Olist

```
SELECT
   pc.product_category_name_english AS product_category,
   CONCAT('R$ ', ROUND(SUM(op.payment_value), 2)) AS order_value
FROM
   orders o
JOIN
   customers c ON o.customer_id = c.customer_id
JOIN
   order_items oi ON o.order_id = oi.order_id
JOIN
   order_payments op ON o.order_id = op.order_id
JOIN
   products p ON oi.product_id = p.product_id
JOIN
   product_category_name_translation pc ON pc.product_category_name = p.product_category_name
GROUP BY
   pc.product_category_name
HAVING
   SUM(op.payment_value) > 1000
ORDER BY
   SUM(op.payment_value) DESC;
```

Result:

	product_category character varying (100)	order_value text
1	bed_bath_table	R$ 1712553.67
2	health_beauty	R$ 1657373.12
3	computers_accessories	R$ 1585330.45
4	furniture_decor	R$ 1430176.39
5	watches_gifts	R$ 1429216.68
6	sports_leisure	R$ 1392127.56
7	housewares	R$ 1094758.13
8	auto	R$ 852294.33
9	garden_tools	R$ 838280.75
10	cool_stuff	R$ 779698.00
11	office_furniture	R$ 646826.49
12	toys	R$ 619037.69
13	baby	R$ 539845.66
14	perfumery	R$ 506738.66

***Note: Only a portion of the table is shown.**

Product Metrics

Expensive Products Product Category wise

This query calculates expensive orders (>R$ 200) product category wise that was ordered through Olist

```
SELECT
    pc.product_category_name_english AS product_category,
    CONCAT('R$ ', ROUND(AVG(op.payment_value), 2)) AS total_order_value
FROM
    orders o
JOIN
    customers c ON o.customer_id = c.customer_id
JOIN
    order_items oi ON o.order_id = oi.order_id
JOIN
    order_payments op ON o.order_id = op.order_id
JOIN
    products p ON oi.product_id = p.product_id
JOIN
    product_category_name_translation pc ON pc.product_category_name = p.product_category_name
GROUP BY
    pc.product_category_name
HAVING
    AVG(op.payment_value) > 200
ORDER BY
    AVG(op.payment_value) DESC;
```

Result:

	product_category character varying (100)	total_order_value text
1	computers	R$ 1268.73
2	fixed_telephony	R$ 763.88
3	small_appliances_home_oven_and_cof...	R$ 656.79
4	agro_industry_and_commerce	R$ 471.15
5	home_appliances_2	R$ 464.79
6	office_furniture	R$ 363.79
7	signaling_and_security	R$ 340.74
8	construction_tools_safety	R$ 330.11
9	musical_instruments	R$ 324.62
10	small_appliances	R$ 321.80
11	air_conditioning	R$ 301.89
12	furniture_living_room	R$ 257.84
13	construction_tools_construction	R$ 253.92
14	furniture_bedroom	R$ 252.79

***Note: Only a portion of the table is shown.**

Products Insights

Total products metrics

Successfully identified top selling products , products with top average ratings and worst products by return rate.

High value orders category wise

Identified total orders by product categories which generated highest revenues for Olist. Bed_bath tables , health and beauty care, Computer accessories, decorative furnitures and watches are qamong the top 5.

Expensive products listed on Olist

Succefully identified top expensive products listed on Olist. Computers top the chart.

Seller Metrics

Order Delivery Performance by Seller

This query calculates top 10 worst and best Sellers by delivey time existing on Olist

```
SELECT
   seller_id,

ROUND(AVG(order_delivered_carrier_d
ate - order_approved_at), 0) AS
avg_delivery_time
FROM
   order_items oi
JOIN
   orders o ON oi.order_id = o.order_id
WHERE
   order_delivered_carrier_date IS NOT
NULL
GROUP BY
   seller_id
HAVING
   COUNT(o.order_id) > 10
ORDER BY
   avg_delivery_time DESC
LIMIT 10;
```

```
SELECT
   seller_id,

ROUND(AVG(order_delivered_carrier_d
ate - order_approved_at), 0) AS
avg_delivery_time
FROM
   order_items oi
JOIN
   orders o ON oi.order_id = o.order_id
WHERE
   order_delivered_carrier_date IS NOT
NULL
GROUP BY
   seller_id
HAVING
   COUNT(o.order_id) > 10
ORDER BY
   avg_delivery_time  ASC
LIMIT 10;
```

Result:

	seller_id character varying (50)	avg_delivery_time numeric
1	ecccfa2bb93b34a3bf033cc5d1dcdc69	26
2	66e0557ecc2b4dbea057e93f215f68d8	18
3	ed859002ad59dbf8cf3602696a6c3000	17
4	54965bbe3e4f07ae045b90b0b8541f52	16
5	5058e8c1e82653974541e83690655b4a	15
6	7fc87cc3e89b3d1d5cabdca32f8485aa	15
7	6fd52c528dcb38be2eea044946b811f8	14
8	8bd0e3abda539b9479c4b44a691be1ec	14
9	b1b3948701c5c72445495bd161b83a4c	14
10	817f85dbb65aa3e70831d90fe75cdf89	14

	seller_id character varying (50)	avg_delivery_time numeric
1	7ff588a03c2aeae4fbd23f9ae64b760d	0
2	334cab711dee080b079fa5779b584783	0
3	165b1235e9e9942cb5fae67103576fb0	0
4	e24d3429d294b2eb200b064ebb035879	0
5	c3e1abd72a42fe690fcd89cf5720fe29	0
6	6e1862e15f33d9994bc25922a85e1efc	0
7	5a413ade68e8f8d93071a7f52a64cb9e	0
8	b9a03475e6447e631b6799ec8274800f	0
9	ce69a8021d18961dd2a40269b7c2c293	1
10	5011f0d93373a4c5753adf58ca77af8d	1

***Note: Only a portion of the table is shown.**

Seller Metrics

Top Sellers by Revenue Generation

This query calculates top Sellers by revenue eneration existing on Olist

```
SELECT
   seller_id,
   CONCAT('R$ ', ROUND(SUM(payment_value), 2)) AS total_revenue
FROM
   order_items oi
JOIN
   order_payments op ON oi.order_id = op.order_id
GROUP BY
   seller_id
ORDER BY
   SUM(payment_value) DESC;
```

Result:

	seller_id character varying (50)	total_revenue text
1	7c67e1448b00f6e969d365cea6b010ab	R$ 507166.91
2	1025f0e2d44d7041d6cf58b6550e0bfa	R$ 308222.04
3	4a3ca9315b744ce9f8e9374361493884	R$ 301245.27
4	1f50f920176fa81dab994f9023523100	R$ 290253.42
5	53243585a1d6dc2643021fd1853d8905	R$ 284903.08
6	da8622b14eb17ae2831f4ac5b9dab84a	R$ 272219.32
7	4869f7a5dfa277a7dca6462dcf3b52b2	R$ 264166.12
8	955fee9216a65b617aa5c0531780ce60	R$ 236322.30
9	fa1c13f2614d7b5c4749cbc52fecda94	R$ 206513.23
10	7e93a43ef30c4f03f38b393420bc753a	R$ 185134.21
11	6560211a19b47992c3666cc44a7e94c0	R$ 179657.75
12	7a67c85e85bb2ce8582c35f2203ad736	R$ 169030.80
13	25c5c91f63607446a97b143d2d535d31	R$ 160534.74
14	a1043bafd471dff536d0c462352beb48	R$ 154356.91

***Note: Only a portion of the table is shown.**

Seller Insights

Top Sellers by delivery time

Successfully identified top sellers by avg delivery time and also the worst sellers by delivery time.

Top Sellers by revenue generation

Successfully identified top sellers by revenue generation for Olist. Top seller has generated more than R$500k in sales revenue.

Review Metrics

Average Review

This query calculates average review of the reviews given

```
SELECT
 AVG(review_score) AS avg_review_score
FROM  order_reviews;
```

Result:

avg_review_score
numeric
4.086420624042570 3

Review Metrics

Average Review Category wise

This query calculates average review given product category wise

```
SELECT    p.category_name_english AS Category,
AVG(review_score) AS avg_review_score
FROM     order_reviews o
JOIN order_items oi ON o.order_id=oi.order_id
JOIN ProductWithCategoryEnglish p    ON
p.product_id = oi.product_id
Group BY p.category_name_English
Order by AVG(review_score) DESC;
```

Result:

	category character varying (100)	avg_review_score numeric
1	cds_dvds_musicals	4.64
2	fashion_childrens_clothes	4.50
3	books_general_interest	4.45
4	costruction_tools_tools	4.44
5	flowers	4.42
6	books_imported	4.40
7	books_technical	4.37
8	food_drink	4.32
9	luggage_accessories	4.32
10	small_appliances_home_oven_and_cof...	4.30
11	fashion_sport	4.26
12	fashion_shoes	4.23
.....		
63	furniture_mattress_and_upholstery	3.82
64	fashio_female_clothing	3.78
65	party_supplies	3.77
66	fixed_telephony	3.68
67	fashion_male_clothing	3.64
68	home_comfort_2	3.63
69	office_furniture	3.49
70	diapers_and_hygiene	3.26
71	security_and_services	2.50

***Note: Only a portion of the table is shown.**

Review Metrics

Average Review for high value orders

This query calculates average review given for high value orders once for all the order and again for different categories product category wise

```
SELECT
   AVG(r.review_score) AS avg_review_score FROM  orders o
JOIN   order_items oi ON o.order_id = oi.order_id
JOIN  products p ON oi.product_id = p.product_id
JOIN product_category_name_translation pc ON p.product_category_name =
pc.product_category_name
JOIN order_reviews r ON o.order_id = r.order_id
WHERE
   oi.price > 200
ORDER BY
   avg_review_score DESC;

SELECT p.category_name_english AS Category,   ROUND(AVG(review_score), 2) AS
avg_review_score
FROM order_reviews o  JOIN  order_items oi ON o.order_id = oi.order_id
JOIN ProductWithCategoryEnglish p ON p.product_id = oi.product_id
GROUP BY  p.category_name_english
ORDER BY  AVG(review_score) DESC;
```

Result:

avg_review_score numeric
4.024292112654672 4

	category character varying (100)	avg_review_score numeric
1	cds_dvds_musicals	4.64
2	fashion_childrens_clothes	4.50
3	books_general_interest	4.45
4	costruction_tools_tools	4.44
5	flowers	4.42
6	books_imported	4.40
...	.....	...
66	fixed_telephony	3.68
67	fashion_male_clothing	3.64
68	home_comfort_2	3.63
69	office_furniture	3.49
70	diapers_and_hygiene	3.26
71	security_and_services	2.50

***Note: Only a portion of the table is shown.**

Review Insights

Average Review Score

Average reviews on the platform is slightly more than 4 which is good. Olist can make some effort to increase the average review on the platform.

Reviews for different categories

Fashion clothes, books, small home applainces, food items , shoes etc have high average review which has great market potential. On the down side , male clothing , mattress , office furnitures etc has significantly low average review. These are some great segments to generate revenue.

Reviews for High value orders

The platform has just above 4 as average review for high value orders which is alarming as big e commerce website try to keep atleast 4.2t o 4.5 for their high value orders average review. Particulary the segment of office furnitures is performing poorly considering it's potential and m,arket size. Olist has to make some serious effort in this area.

Marketing Metrics

Average days needed to convert a lead into a successful one & percentage of different kind of leads

This query calculates average days needed to convert a lead and subsequently the percentage share of total lead generation (irrespective of success) category wise

```
SELECT
   origin,
   ROUND(AVG(won_date - first_contact_date), 0) AS AVG_days_needed,
   CONCAT(ROUND((COUNT(mql_id) * 100.0 / SUM(COUNT(mql_id)) OVER
()), 2), '%') AS percentage_of_leads
FROM
   Marketing
GROUP BY
   origin;
```

Result:

	origin character varying (50)	avg_days_needed numeric	percentage_of_leads text
1	direct_traffic	31	6.65%
2	Unknown	49	1.66%
3	display	10	0.71%
4	email	52	1.78%
5	paid_search	57	23.16%
6	other_publicities	39	0.36%
7	referral	33	2.85%
8	social	61	8.91%
9	organic_search	50	32.19%
10	other	15	0.48%
11	unknown	41	21.26%

Marketing Metrics

Lead-to-Order-Delivery Conversion Time For different Sellers

This query calculates average days needed to convert a lead into a successful one and making the delivery seller wise wise

```
SELECT
   s.seller_id,
   ROUND(AVG(o.order_delivered_customer_date - m.first_contact_date), 0) AS
AVG_days_needed_lead_to_order
FROM
   Marketing m
JOIN
   sellers s ON m.seller_id = s.seller_id
JOIN
   order_items oi ON oi.seller_id = s.seller_id
JOIN
   orders o ON o.order_id = oi.order_id
GROUP BY
   s.seller_id
HAVING
   AVG(o.order_delivered_customer_date - m.first_contact_date) IS NOT NULL
IS NOT NULL ORDER BY AVG_days_needed_lead_to_order;
```

Result:

	seller_id [PK] character varying (50)	avg_days_needed_lead_to_order numeric
1	f1fdf2d13186575751aa25876536d85c	21
2	1f7dfad2cb384ea4d2d7e1ffbd78c407	22
3	7816cd9c5b1238e320545f5bf7eb80e8	26
4	447d377bdb757058acb569025ee18a93	28
5	880ce4951cf857ab1d9e0b75c1d856f4	29
6	d7827b2af99326a03b0ed9c7a24db0d3	29
7	6ebf4eceee4dd9847201c82e77ef8123	30
8	eb9267cccc90f1b49c8d2f9887c7dd97	32
9	a663d9c3797e90eac99ff60939416a56	33
10	1a8e2d9c38b84a9702ac7922924b0573	41
11	ce4755084bc097113867e6454f8f5e52	42
12	c8665a4dd081a2c436b1cd921079d0d9	43
13	bec568278124768c474ee90971ca94d1	44
14	ca7c6bd577e559472af1c699de9e764e	45
15	99cd94252748d2bdde08e17858233602	49
16	6f835fd4be26989b1b064399da346143	49
17	eb72802c83dc7547529c9546d1a9b8ef	55
18	dda37071807e404c5bb2a1590c66326f	55
19	4eeb99008a0f59d2c7759c59f9a346eb	55
20	117cfc326c6d50da67ca858ff5c0c852	57

***Note: Only a portion of the table is shown.**

Marketing Metrics

Lead-to-Order-Delivery Conversion Time For different Sellers

This query calculates the percentage share of different methods in generating 'successful' leads.

```
SELECT
   origin AS lead_generation_source,
   CONCAT(ROUND((COUNT(landing_page_id) * 100.0 /
SUM(COUNT(landing_page_id)) OVER ()), 2), '%') AS
percentage
FROM
   leads_qualified lc
GROUP BY
   origin;
```

Result:

	lead_generation_source character varying (50)	percentage text
1	direct_traffic	6.24%
2	Unknown	0.75%
3	display	1.48%
4	email	6.16%
5	paid_search	19.83%
6	other_publicities	0.81%
7	referral	3.55%
8	social	16.88%
9	organic_search	28.70%
10	other	1.88%
11	unknown	13.74%

***Note: Only a portion of the table is shown.**

Marketing Metrics

Comparison between Website and Social Media Marketing Effectiveness

This query compares between the values of percentage share of lead generating traffic in through the company website (organic) and social media (advertisement)

```
WITH total_leads AS (
   SELECT
      SUM(COUNT(landing_page_id)) OVER () AS total_leads
   FROM
      leads_qualified
)
SELECT
   CONCAT(ROUND(
      (COUNT(CASE WHEN origin IN ('direct_traffic', 'organic_search') THEN landing_page_id
END) * 100.0)
      / (SELECT total_leads FROM total_leads), 2), '%') AS lead_generation_from_website,
   CONCAT(ROUND(
      (COUNT(CASE WHEN origin IN ('social') THEN landing_page_id END) * 100.0)
      / (SELECT total_leads FROM total_leads), 2), '%') AS
lead_generation_through_social_media_marketing
FROM
   leads_qualified;
```

Result:

lead_generation_from_website text	lead_generation_through_social_media_marketing text
34.94%	16.88%

Marketing Insights

Lead Convertion to order

Lowest days needed for lead conversion is already as low as three weeks. Olist can try to achieve atleast two weeks as lowest.

Website vs Social Media Marketing

Website's organic traffic beats social media marketing lead generation which indicates that the brand name is getting stronger and established in the sector so that customers are trying to find us more

Effectiveness of Social media ads

Social media ads generates only 8.91% leads but it generates 16.88% of successful leads which indicates Olist's efficient social media campaigning. Olist now need to campaign a bit more aggresively

Regional Metrics

Sales by Region (Customer Location)

This query calculates total sales grouped by customer city and state.

```
SELECT
   customer_city,
   customer_state,
   CONCAT('R$ ', ROUND(SUM(payment_value), 2)) AS total_sales
FROM
   orders o
JOIN
   order_payments op ON o.order_id = op.order_id
JOIN
   customers c ON o.customer_id = c.customer_id
GROUP BY
   customer_city, customer_state
ORDER BY
   SUM(payment_value) DESC;
```

Result:

	customer_city character varying (100)	customer_state character varying (5)	total_sales text
1	sao paulo	SP	R$ 2203373.09
2	rio de janeiro	RJ	R$ 1161927.36
3	belo horizonte	MG	R$ 421765.12
4	brasilia	DF	R$ 354216.78
5	curitiba	PR	R$ 247392.48
6	porto alegre	RS	R$ 224731.42
7	salvador	BA	R$ 218071.50
8	campinas	SP	R$ 216248.43
9	guarulhos	SP	R$ 165121.99
10	niteroi	RJ	R$ 139996.99
11	goiania	GO	R$ 125494.60
12	sao bernardo do campo	SP	R$ 120434.84
13	fortaleza	CE	R$ 119863.47
14	santos	SP	R$ 112343.17
15	recife	PE	R$ 110745.23
16	florianopolis	SC	R$ 106512.16
17	santo andre	SP	R$ 105536.19
18	belem	PA	R$ 95842.26
19	osasco	SP	R$ 94358.72
20	jundiai	SP	R$ 93038.64
21	sao jose dos campos	SP	R$ 91637.75

***Note: Only a portion of the table is shown.**

Regional Metrics

Seller Concentration by Region

This query calculates the number of sellers in each city and state

```
SELECT
   seller_city,
   seller_state,
   COUNT(seller_id) AS seller_count
FROM
   sellers
WHERE seller_state <> 'NA'
GROUP BY
   seller_city, seller_state
ORDER BY
   seller_count DESC;
```

Result:

	seller_city character varying (100)	seller_state character varying (2)	seller_count bigint
1	sao paulo	SP	694
2	curitiba	PR	124
3	rio de janeiro	RJ	93
4	belo horizonte	MG	66
5	ribeirao preto	SP	52
6	guarulhos	SP	50
7	ibitinga	SP	49
8	santo andre	SP	45
9	campinas	SP	41
10	maringa	PR	40
11	sao jose do rio preto	SP	33
12	sorocaba	SP	32
13	osasco	SP	32
14	sao bernardo do campo	SP	32
15	brasilia	DF	28
16	porto alegre	RS	27
17	londrina	PR	25
18	goiania	GO	23
19	joinville	SC	22

***Note: Only a portion of the table is shown.**

Regional Metrics

Customer Concentration by Region

This query calculates the number of customers in each city and state

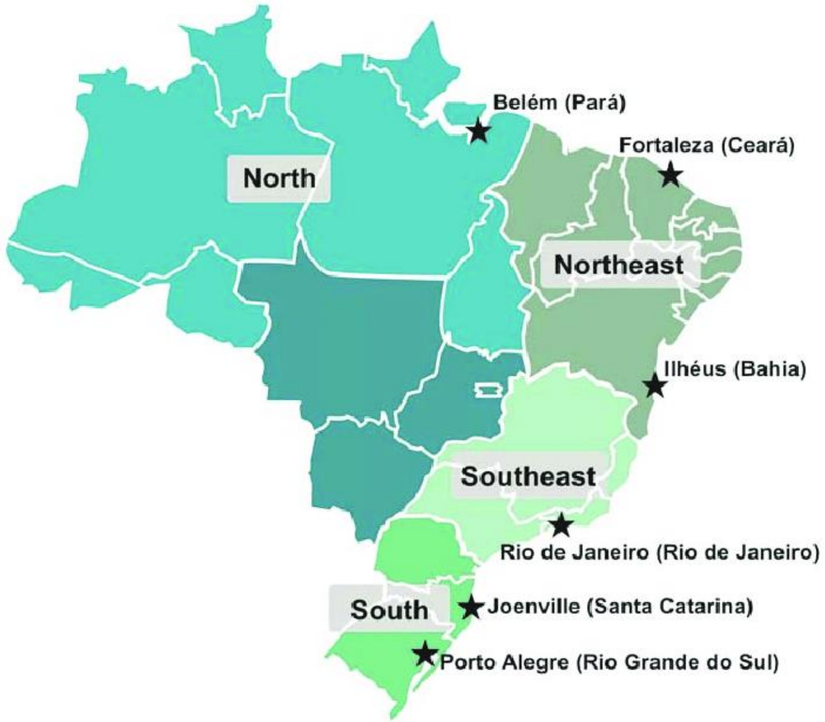
```
SELECT
customer_city,
customer_state,
COUNT( customer_id) AS  customer_count
FROM customers
WHERE  customer_state <> 'NA’
GROUP BY customer_city,  customer_state
ORDER BY customer_count DESC;
```

Result:

	customer_city character varying (100)	customer_state character varying (5)	customer_count bigint
1	sao paulo	SP	15540
2	rio de janeiro	RJ	6882
3	belo horizonte	MG	2773
4	brasilia	DF	2131
5	curitiba	PR	1521
6	campinas	SP	1444
7	porto alegre	RS	1379
8	salvador	BA	1245
9	guarulhos	SP	1189
10	sao bernardo do campo	SP	938
11	niteroi	RJ	849
12	santo andre	SP	796
13	osasco	SP	746
14	santos	SP	713
15	goiania	GO	692
16	sao jose dos campos	SP	691
17	fortaleza	CE	654
18	sorocaba	SP	633
19	recife	PE	613
20	florianopolis	SC	570

***Note: Only a portion of the table is shown.**

Regional Insights



Sales by Region

Successfully identified regions which has highest sales. Unsuprisingly , Sao Paulo , Rio and Bello Horizonte has the higest number of sales.

Customer Concentration

Successfully identified regions which has highest customer concentration. Unsuprisingly , Sao Paulo , Rio and Bello Horizonte has the higest number of customer concentration.

Seller Concentration

Successfully identified regions which has highest seller concentration. Sao Paulo , Curitiba and Rio has the higest number of customer concentration.

Payment Metrics

Payment Method Distribution

This query calculates percentage of orders paid via different payment methods (e.g., credit card, PayPal)

```
SELECT
    payment_type,
    COUNT(*) AS total_orders,
    CONCAT(ROUND((COUNT(*) * 100.0 / (SELECT COUNT(*) FROM
order_payments)), 2), '%') AS payment_method_percentage
FROM
    order_payments
GROUP BY
    payment_type
ORDER BY
    payment_method_percentage DESC;
```

Result:

	payment_type character varying (50)	total_orders bigint	payment_method_percentage text
1	credit_card	76795	73.92%
2	voucher	5775	5.56%
3	boleto	19784	19.04%
4	debit_card	1529	1.47%
5	not_defined	3	0.00%

Payment Metrics

Installment Plans Usage

This query calculates number of orders using installment payments

```
SELECT
    pc.product_category_name_english AS product_category,
    COUNT(oi.*) AS order_quantity,
    SUM(op.payment_installments) AS installment_orders,
    (SUM(op.payment_installments) / COUNT(oi.*)) AS avg_installments_per_order,
    CONCAT('R$ ', ROUND(AVG(op.payment_value / op.payment_installments), 2)) AS avg_installment_value
FROM
    order_payments op
JOIN
    order_items oi ON op.order_id = oi.order_id
JOIN
    products p ON oi.product_id = p.product_id
JOIN
    product_category_name_translation pc ON p.product_category_name = pc.product_category_name
WHERE
    op.payment_installments > 1
GROUP BY
    pc.product_category_name_english
ORDER BY
    avg_installments_per_order DESC;
```

Result:

	product_category character varying (100)	oder_quantity bigint	installment_orders bigint	avg_installments_per_orders bigint	avg_installment_value text
1	small_appliances_home_oven_and_cof...	51	401	7	R$ 109.29
2	computers	165	1267	7	R$ 200.30
3	home_appliances_2	146	978	6	R$ 110.74
4	agro_industry_and_commerce	98	590	6	R$ 109.83
5	office_furniture	998	6007	6	R$ 74.73
6	home_confort	278	1700	6	R$ 39.10
7	furniture_living_room	299	1897	6	R$ 50.84
8	watches_gifts	4012	20587	5	R$ 54.30
9	air_conditioning	141	745	5	R$ 80.52
10	bed_bath_table	7133	37319	5	R$ 37.04
11	cds_dvds_musicals	5	27	5	R$ 15.33
12	cine_photo	26	142	5	R$ 51.04
13	construction_tools_construction	520	3047	5	R$ 58.82
14	construction_tools_lights	148	752	5	R$ 67.89
15	construction_tools_safety	100	588	5	R$ 80.55
16	costruction_tools_garden	105	580	5	R$ 51.27

***Note: Only a portion of the table is shown.**

Payment Insights

Payment Method Distribution

Most of the payments , approx. 74% of the payments have been done using a credit card . Olist must collaborate with different credit card companies to give as many as offers and discounts they can give to increase the salses.

Availability of EMIs

Various home appliances and Computers has more number of average EMIs which are high in value too. Olist must focus to increase the availibilty and the ease of the process to increase sales in these areas.

Making the payment system efficient

Olist have done a tremendous job on creating an efficient online payment system as the pay on delivery method is almost neglible . Olist must keep it up such a good efficient system

Shipping Metrics

On-time Delivery Rate

This query calculates on-time Delivery Rate (i.e., the percentage of orders delivered before the estimated delivery date)

```
SELECT
   CONCAT(ROUND((COUNT(CASE WHEN
order_delivered_customer_date <=
order_estimated_delivery_date
      THEN 1 END) * 100.0 / COUNT(order_id)), 2), '%') AS
on_time_delivery_rate
FROM
   orders
WHERE
   order_delivered_customer_date IS NOT NULL
   AND order_estimated_delivery_date IS NOT NULL;
```

Result:

on_time_delivery_rate text
93.23%

Shipping Metrics

Average delivery cost region-wise

This query calculates average delivery cost region-wise

```
SELECT
   c.customer_city,
   c.customer_state,
   ROUND(AVG(oi.freight_value), 2) AS avg_freight_value
FROM
   orders o
JOIN
   customers c ON o.customer_id = c.customer_id
JOIN
   order_items oi ON o.order_id = oi.order_id
GROUP BY
   c.customer_city, c.customer_state
ORDER BY
   avg_freight_value DESC;
```

Result:

	customer_city character varying (100)	customer_state character varying (5)	avg_freight_value numeric
1	itupiranga	PA	203.38
2	amarante	PI	193.84
3	almino afonso	RN	170.11
4	canapi	AL	147.32
5	marilac	MG	142.49
6	sao martinho	RS	142.33
7	sanharo	PE	140.50
8	alhandra	PB	135.46
9	boa esperanca	RJ	127.52
10	pianco	PB	124.99
11	icatu	MA	114.90
12	graccho cardoso	SE	113.72
13	humildes	BA	110.82
14	engenheiro navarro	MG	109.55
15	sao vicente ferrer	MA	108.47
16	cedro	CE	106.21
17	araguana	MA	105.07
18	nova mamore	RO	104.13
19	soledade	PB	103.28
20	itaporanga d'ajuda	SE	103.25

***Note: Only a portion of the table is shown.**

Shipping Insights

On time Delivery

Olist completes 93.23% delivery on time which Indicates their determination for timely deliveries.

Delivery costs for different regions

Identified delivery costs for different regions. Itupiranga tops the chart with more than R$200 average delivery cost which surprisingly high. Olist can make delivery godowns in this place where delivery cost is unusually high.

Growth Metrics

Revenue, Net Profit , Average Order Profit year on year category wise

This query calculates Revenue, Net Profit , Average Order Profit year on year category wise

```
SELECT
    pc.product_category_name_english AS product_category,
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
    COUNT(op.order_id) AS number_of_orders,
    SUM(op.payment_value) AS total_revenue,
    SUM(op.payment_value - (oi.price + oi.freight_value)) AS net_profit,
    AVG(op.payment_value - (oi.price + oi.freight_value)) AS avg_order_profit
FROM
    products p
JOIN
    product_category_name_translation pc ON pc.product_category_name = p.product_category_name
JOIN
    order_items oi ON p.product_id = oi.product_id
JOIN
    order_payments op ON oi.order_id = op.order_id
JOIN
    orders o ON oi.order_id = o.order_id
GROUP BY
    pc.product_category_name_english, EXTRACT(YEAR FROM o.order_purchase_timestamp)
ORDER BY
    order_year DESC, net_profit DESC;
```

Result:

	product_category character varying (100)	order_year numeric	number_of_orders bigint	total_revenue numeric	net_profit numeric	avg_order_profit numeric
1	computers_accessories	2018	4834	878236.82	262460.77	54.2947393462970625
2	furniture_decor	2018	4289	744045.60	245893.97	57.3313056656563301
3	bed_bath_table	2018	6169	912947.74	233590.40	37.8651969525044578
4	housewares	2018	4247	699854.31	184615.43	43.4696091358606075
5	office_furniture	2018	903	353740.15	167884.38	185.9184717607973422
6	health_beauty	2018	6117	1033604.09	129193.52	21.1204054274971391
7	sports_leisure	2018	4663	746186.14	108700.64	23.3113103152476946
8	garden_tools	2018	1953	352735.09	84033.63	43.0279723502304147
9	construction_tools_construction	2018	815	207117.98	63279.05	77.6430061349693252
10	telephony	2018	2427	277984.71	51416.61	21.1852533992583436
11	watches_gifts	2018	3832	854349.04	50454.57	13.1666414405010438
12	drinks	2018	293	62570.09	40308.69	137.5723208191126280
13	signaling_and_security	2018	153	60361.75	37948.66	248.0304575163398693
14	auto	2018	2696	453031.45	31603.36	11.7223145400593472
15	computers	2018	77	101633.04	31554.68	409.8010389610389610
16	electronics	2018	1890	168245.28	31193.19	16.5043333333333333

***Note: Only a portion of the table is shown.**

Growth Metrics

Top 10 sellers of 2018

This query calculates top 10 sellers of 2018

```
WITH yearly_seller_stats AS ( SELECT   oi.seller_id,
      EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
SUM(op.payment_value) AS total_revenue
   FROM  order_items oi
   JOIN   orders o ON oi.order_id = o.order_id
   JOIN order_payments op ON oi.order_id = op.order_id
   GROUP BY   oi.seller_id, EXTRACT(YEAR FROM o.order_purchase_timestamp)
), seller_growth AS (
   SELECT  seller_id,   order_year,  total_revenue,
      LAG(total_revenue) OVER (PARTITION BY seller_id ORDER BY order_year) AS
prev_year_revenue,
      ROUND(((total_revenue - LAG(total_revenue) OVER (PARTITION BY seller_id ORDER BY
order_year)) /
         NULLIF(LAG(total_revenue) OVER (PARTITION BY seller_id ORDER BY order_year), 0))
* 100, 2) AS revenue_yoy_growth
   FROM   yearly_seller_stats
) SELECT  seller_id, order_year,    total_revenue,  revenue_yoy_growth FROM  seller_growth
WHERE  revenue_yoy_growth IS NOT NULL  AND order_year IN (2018)
ORDER BY total_revenue DESC  LIMIT 10;
```

Result:

	seller_id character varying (50)	order_year numeric	total_revenue numeric	revenue_yoy_growth numeric
1	7c67e1448b00f6e969d365cea6b010ab	2018	275095.67	18.54
2	1025f0e2d44d7041d6cf58b6550e0bfa	2018	228309.63	185.70
3	955fee9216a65b617aa5c0531780ce60	2018	203871.06	528.24
4	4869f7a5dfa277a7dca6462dcf3b52b2	2018	157684.65	48.09
5	da8622b14eb17ae2831f4ac5b9dab84a	2018	145417.85	14.68
6	1f50f920176fa81dab994f9023523100	2018	132205.74	-16.35
7	4a3ca9315b744ce9f8e9374361493884	2018	109764.38	-42.68
8	fa1c13f2614d7b5c4749cbc52fecda94	2018	101919.68	-2.56
9	6560211a19b47992c3666cc44a7e94c0	2018	97044.96	17.47
10	a1043bafd471dff536d0c462352beb48	2018	90385.14	41.29

Growth Metrics

Growth of Revenue, Net Profit , Average Order Profit year on year category wise

This query calculates growth of Revenue, Net Profit , Average Order Profit year on year category wise

Query is on the next tab

Result:

	product_category character varying (100)	order_year numeric	total_revenue numeric	revenue_yoy_growth_percentage numeric	net_profit_yoy_growth_percentage numeric	avg_order_profit_yoy_growth_percentage numeric
1	agro_industry_and_commerce	2017	65094.65	[null]	[null]	[null]
2	agro_industry_and_commerce	2018	53635.96	-17.60	-117.66	-105.89
3	air_conditioning	2016	4675.43	[null]	[null]	[null]
4	air_conditioning	2017	40746.85	771.51	215.10	-76.66
5	air_conditioning	2018	45748.38	12.27	93.12	66.06
6	art	2017	12060.75	[null]	[null]	[null]
7	art	2018	18932.18	56.97	-62.30	-89.70
8	arts_and_craftmanship	2017	218.63	[null]	[null]	[null]
9	arts_and_craftmanship	2018	2107.54	863.98	127.18	-79.35
10	audio	2016	183.03	[null]	[null]	[null]
11	audio	2017	21200.17	11482.89	[null]	[null]
12	audio	2018	38941.42	83.68	1723.70	1488.97
13	auto	2016	2716.70	[null]	[null]	[null]
14	auto	2017	396546.18	14496.61	23376.45	68.59
15	auto	2018	453031.45	14.24	-70.68	-81.83
16	baby	2016	2344.43	[null]	[null]	[null]
17	baby	2017	197887.52	8340.75	2326.30	-74.31
18	baby	2018	339613.71	71.62	61.33	14.66
19	bed_bath_table	2016	2291.71	[null]	[null]	[null]

***Note: Only a portion of the table is shown.**

```
WITH yearly_stats AS (
  SELECT
    pc.product_category_name_english AS product_category,
    EXTRACT(YEAR FROM o.order_purchase_timestamp) AS order_year,
    SUM(op.payment_value) AS total_revenue,
    SUM(op.payment_value - (oi.price + oi.freight_value)) AS net_profit,
    AVG(op.payment_value - (oi.price + oi.freight_value)) AS avg_order_profit
  FROM
    products p
  JOIN
    product_category_name_translation pc ON pc.product_category_name = p.product_category_name
  JOIN
    order_items oi ON p.product_id = oi.product_id
  JOIN
    order_payments op ON oi.order_id = op.order_id
  JOIN
    orders o ON oi.order_id = o.order_id
  GROUP BY
    pc.product_category_name_english, EXTRACT(YEAR FROM o.order_purchase_timestamp)
  ORDER BY
    order_year DESC, net_profit DESC
)
SELECT
  product_category,
  order_year,
  total_revenue,
  ROUND(((total_revenue - LAG(total_revenue) OVER (PARTITION BY product_category ORDER BY order_year)) /
    NULLIF(LAG(total_revenue) OVER (PARTITION BY product_category ORDER BY order_year), 0)) * 100, 2) AS revenue_yoy_growth_percentage,
  ROUND(((net_profit - LAG(net_profit) OVER (PARTITION BY product_category ORDER BY order_year)) /
    NULLIF(LAG(net_profit) OVER (PARTITION BY product_category ORDER BY order_year), 0)) * 100, 2) AS net_profit_yoy_growth_percentage,
  ROUND(((avg_order_profit - LAG(avg_order_profit) OVER (PARTITION BY product_category ORDER BY order_year)) /
    NULLIF(LAG(avg_order_profit) OVER (PARTITION BY product_category ORDER BY order_year), 0)) * 100, 2) AS avg_order_profit_yoy_growth_percentage
FROM
  yearly_stats
ORDER BY
  product_category, order_year;
```

Growth Insights

Revenue, Net profit

Identified top segments which generated most of the revenues and net profits. Computer accessories tops the chart while surprisingly despite pretty average reviews, revenue generation from office furnitures sits at no.5 indicating the the potential it this market has.

Top sellers in recent year

Identified top 10 sellers of 2018 with their yearly growth in sales. Some of them have shown growths from 100% ranging upto 500%.

YoY Growth in revenue, profits

Olist have recorded an huge surge in sales 2017 recording growth in 4 to 5 digits in some cases. However , growth in 2018 tanked to mere two digits or one digits in many categories and in some cases it went negative. For an ecommerce , it should grow exponentially in their initial years to establish the company n this sector. Olist have to make a lot of efforts to do that.

Recommendations to increase Business & Profit

Delivery	Marketing	Products	Threats
Increase seller network and establish more godowns to reduce average delivery time and delivery costs	Make the social media marketing more aggressive it tends to generates for successful lead generation	Focus on large markets which has huge potential yet Olist performs a bit poorly as per the average customer reviews suggests (ex- Office Furnitures)	Olist's growth tanked in 2018 in many segments. Ecommerce company must have growth exponentially to stay in the business and establish the brand. Olist's 2018 growth could be an alarming indication of their future

Thank You!

-From

Hafiz Adeel Arif

I’m excited to continue growing and taking on more challenges that blend analytics with business understanding! Feel free to add any suggestions or recommendations.