e-Commerce Sales Report

PROJECT PLATFORM : MYSQL

The eCommerce dataset project involves the analysis of an eCommerce platform's data stored in multiple tables. The primary goal is to understand customer behavior, product performance, and overall business metrics

Leju Monachan

Project Overview

The eCommerce dataset project involves the analysis of an eCommerce platform's data stored in multiple tables. The primary goal is to understand customer behavior, product performance, and overall business metrics. The dataset includes the following tables:

- Customers
- Products
- Orders
- Order Items
- Geolocation
- Payments

Q : Count the number of orders placed in 2017 ?

```
select * from `order_items`;
select count(order_id) from `order_items` where year(shipping_limit_date) = "2017"
```

Result Grid

	count(order_id)
▶	8568

Q :List all unique cities where customers are located ?

Result Grid | Filter Rows:

unique_customer_city
cristal
parambu
sumidouro
lagarto
putinga
caxambu

customers 3

Q : Find the total sales per category ?

```
select upper(product_category) as PRODUCT_CATEGORY, round(sum(payment_value),2)  as TOTAL_SALES from `products 1` join `order_items`
on `products 1`.product_id = order_items.product_id join payments
on payments.order_id = `order_items`.order_id
group by product_category order by TOTAL_SALES ;
```

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

PRODUCT_CATEGORY	TOTAL_SALES
ROOM FURNITURE	22.56
GENERAL INTEREST BOOKS	35.05
CASA CONSTRUCAO	44.63
FOODS	79.69
FURNITURE KITCHEN SERVICE AREA DINNER A...	92.99
TECHNICAL BOOKS	122.59

Result 9

Q : Calculate the number of orders per month in 2018 ?

```
select count(order_id) as Total , monthname(order_estimated_delivery_date) as Monthwise  from orders where year(order_estimated_delivery_date)
= "2018"
group by monthname(order_estimated_delivery_date);
```

Total	Monthwise
179	September
489	February
596	July
737	August
663	March
427	January

Result 15

Q : Count the number of customers from each state ?

```
select customer_state, count(customer_id) from customers
group by customer_state;
```

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

customer_state	count(customer_id)
SP	23512
SC	2019
MG	6627
PR	2853
RJ	7372
RS	3062

Result 1

Q : Find the average number of products per order, grouped by customer city ?

```sql
select * from orders;
select * from order_items;
with count_order as
(select orders.order_id,orders.customer_id ,count(order_items.order_id) as oc
from orders join order_items on
orders.order_id = order_items.order_id
group by orders.order_id, orders.customer_id )

select customers.customer_city, avg(count_order.oc) from customers join count_order
on customers.customer_id = count_order.customer_id
group by customers.customer_city order by avg(count_order.oc) desc;
```

Result Grid | Filter Rows: | Export:

customer_city	avg(count_order.oc)
poa	6.0000
campinas	3.0000
recife	2.0000
saloa	2.0000
campo grande	2.0000
cuiaba	2.0000

Result 14

Q : Calculate the percentage of total revenue contributed by each product category ?

```sql
select (sum(payment_value) / (select sum(payment_value) from payments))*100 REVENUE_PERCENTAGE , UPPER(product_category ) CATEGORY
from payments join order_items
on payments.order_id =order_items.order_id join products_1
on products_1.product_id = order_items.product_id group by product_category order by product_category ;
```

Result Grid | Filter Rows: | Export:

REVENUE_PERCENTAGE	CATEGORY
0.33105469723819786	AUTOMOTIVE
0.14665147933352868	BABIES
0.7272806814098743	BED TABLE BATH
0.02871268051175865	CASA CONSTRUCAO
0.13465301436502547	CLIMATIZATION
7.631313227165683	COMPUTER ACCESSORIES

Result 7

Q : Identify the correlation between product price and the number of times a product has been purchased ?

```
52
53 •  select product_category,avg(price) as PRICE_AVG ,count(order_items.product_id) AS COUNT_PRODUCT   from order_items join products_1
54    on order_items.product_id =  products_1.product_id
55    group by product_category;
56
57
58
```

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

product_category	PRICE_AVG	COUNT_PRODUCT
HEALTH BEAUTY	129.9879138872029	1649
sport leisure	111.57609257998523	1469
Cool Stuff	160.10352472089303	627
computer accessories	112.48074438202158	1424
Watches present	190.6934856007935	1007
housewares	94.83411910669928	1209

Result 16 ×

Result Grid

Form Editor

Read Only

Q : Calculate the total revenue generated by each seller, and rank them by revenue ?

```sql
SELECT * FROM e_commerce_platform.order_items;

SELECT *, RANK() OVER (ORDER BY payment_value DESC) AS Rank_list from
(select order_items.seller_id, sum(payment_value) as payment_value from order_items join payments
on order_items.order_id = payments.order_id group by order_items.seller_id  ) as ranktable;
```

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

seller_id	payment_value	Rank_list
25c5c91f63607446a97b143d2d535d31	6524.640000000001	1
70a12e78e608ac31179aea7f8422044b	4004.55	2
c3cfdc648177fdbbbb35635a37472c53	1066.41	3
d71d863e5ef30d94e440c11be17dcd8f	1017.63	4
723cd880edaacdb998898b67c8f9da30	724.34	5
7178f9f4dd81dcef02f62acdf8151e01	666.45	6

Result 23

Q : Calculate the cumulative sales per month for each year ?

```
select years ,months , payments , sum(payments) over(order by years,months ) as cumulative from

(select year(orders.order_purchase_timestamp) as years , monthname(orders.order_purchase_timestamp)as months, sum(payment_value) as payments
from orders join payments on orders.order_id = payments.order_id
group by years, months ) as a
```

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

years	months	payments	cumulative
NULL	NULL	75.49000000000001	75.49000000000001
2017	November	170.3	245.79000000000002
2018	August	292.26	538.05
2018	June	219.4	757.4499999999999

Q : Identify the top 3 customers who spent the most money in each year ?

```sql
select orders.customer_id, orders.order_id ,year(orders.order_purchase_timestamp) years, sum(payments.payment_value) Totalpayments,
dense_rank() over (partition by year(orders.order_purchase_timestamp)  order by sum(payments.payment_value)  )  Rankwise
from orders join payments
on orders.order_id = payments.order_id
group by orders.customer_id, orders.order_id ,year(orders.order_purchase_timestamp);
```

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

customer_id	order_id	years	Totalpayments	Rankwise
004d17fa2cb102a7c4817f040c884e59	4689b1816de42507a7d63a4617383c59	NULL	2	1
3	1be51feefcd481bee3118900e6777057	NULL	10.05	2
11	1be51feefcd481bee3118900e6777057	NULL	10.05	2
418af5f91d6a47660e62a208385edac5	7c95572fd6e7e0ab8b88b00fd8c70716	NULL	53.39	3
934a9204aacc7279a9db0896b83eb42e	1355bd6c7fa80ea43bdecab48ff8052c	2017	170.3	1
e6f5b234bb0d847f10eebd70130c5d49	ad133696906f6a78826daa0911b7daec	2018	219.4	1

Result 4

Q : Calculate the moving average of order values for each customer over their order history ?

```sql
select customer_id, order_purchase_timestamp ,avg(payment_value) over(partition by customer_id order by order_purchase_timestamp
rows between 2 preceding and current row) as moving_avg from
(select orders.customer_id,orders.order_purchase_timestamp ,payments.payment_value from
orders join payments on orders.order_id = payments.order_id) as a;
```

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

customer_id	order_purchase_timestamp	moving_avg
004d17fa2cb102a7c4817f040c884e59	sao paulo	2
11	uberlandia	10.05
3	passa tres	10.05
418af5f91d6a47660e62a208385edac5	7	53.39
934a9204aacc7279a9db0896b83eb42e	2017-11-27 01:29:11	85.15

Result 2

THANK YOU