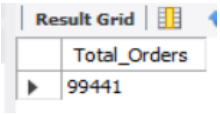
BASIC SQL QUESTIONS (Foundations) Answers

1. How many total orders are there in the dataset?

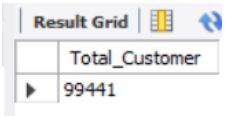
select count(order_id) as Total_Orders from orders;



Total_Orders
99441

2. How many unique customers placed orders?

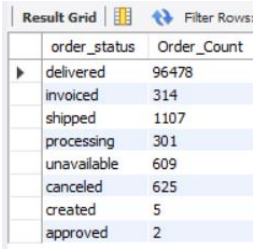
select count(distinct customer_id) as Total_Customer from customers;



Total_Customer
99441

3. What are the different order statuses, and how many orders fall under each status?

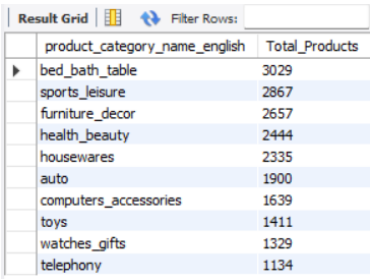
select order_status, count(*) as Order_Count from orders group by order_status;



order_status	Order_Count
delivered	96478
invoiced	314
shipped	1107
processing	301
unavailable	609
canceled	625
created	5
approved	2

4. List the top 10 product categories by number of products.

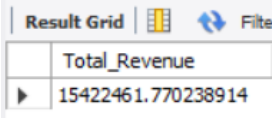
select product_category_name_english, count(*) as Total_Products from products group by product_category_name_english order by Total_Products desc limit 10;



product_category_name_english	Total_Products
bed_bath_table	3029
sports_leisure	2867
furniture_decor	2657
health_beauty	2444
housewares	2335
auto	1900
computers_accessories	1639
toys	1411
watches_gifts	1329
telephony	1134

5. What is the total revenue generated from delivered orders?

select sum(op.payment_value) as Total_Revenue from order_payments op join orders o on op.order_id=o.order_id where o.order_status = "delivered";

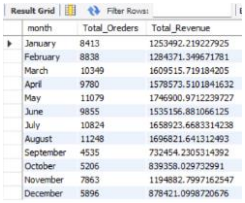


Total_Revenue
15422461.770238914

INTERMEDIATE SQL QUESTIONS (Analytics Thinking)

1. What is the monthly trend of orders and revenue?

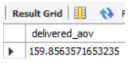
select date_format(order_purchase_timestamp, '%M') as month, count(*) as Total_Oreders, sum(op.payment_value) as Total_Revenue from order_payments op join orders o on op.order_id=o.order_id group by month order by field(month, 'January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September','October', 'November', 'December');



month	Total_Oreders	Total_Revenue
January	8413	1253492.219227925
February	8838	1284371.349671781
March	10349	1609515.719184205
April	9780	1578573.5101841632
May	11079	1746900.9712239727
June	9855	1535156.881066125
July	10824	1658923.6683314238
August	11248	1696821.641312493
September	4535	732454.2305314392
October	5206	839358.029732991
November	7863	1194882.7997162547
December	5896	878421.0998720676

2. What is the average order value (AOV)?

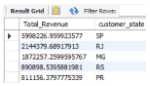
select sum(op.payment_value)/count(distinct o.order_id) as delivered_aov from orders o join order_payments op on o.order_id = op.order_id where o.order_status = 'delivered';



delivered_aov
159.8563571653235

3. Which states generate the highest revenue?

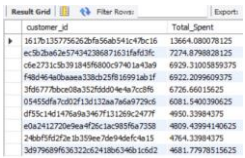
select sum(op.payment_value) as Total_Revenue, customer_state from order_payments op join orders o on op.order_id=o.order_id join customers c on c.customer_id = o.customer_id group by customer_state order by Total_Revenue desc limit 5;



Total_Revenue	customer_state
5998226.959923577	SP
2144379.68917913	RJ
1872257.2599599767	MG
890898.5395881981	RS
811156.3797775339	PR

4. Who are the top 10 customers by total spend?

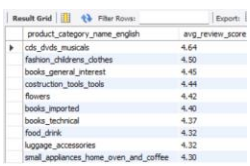
select o.customer_id , sum(op.payment_value) as Total_Spent from orders o join order_payments op on o.order_id=op.order_id group by o.customer_id order by Total_Spent desc limit 10;



customer_id	Total_Spent
1617b1357756262bfa56ab541c47bc16	13664.080078125
ec5b2ba62e574342386871631fafd3fc	7274.8798828125
c6e2731c5b391845f6800c97401a43a9	6929.31005859375
f48d464a0baaea338cb25f816991ab1f	6922.2099609375
3fd6777bbce08a352fddd04e4a7cc8f6	6726.66015625
01455dfa7cd02f13d132aa7a6a9729c6	6081.5400390625
df55c14d1476a9a3467f131269c2477f	4950.33984375
e0a2412720e9ea4f26c1ac985f6a7358	4809.43994140625
24bbf5fd2f2e1b359ee7de94defc4a15	4764.33984375
3d979689f636322c62418b6346b1c6d2	4681.77978515625

5. What are the top 10 by product category average review scores?

select p.product_category_name_english, round(avg(ore.review_score),2) as avg_review_score from products p join order_items oi on p.product_id=oi.product_id join order_reviews ore on ore.order_id=oi.order_id group by p.product_category_name_english order by avg_review_score desc limit 10;



product_category_name_english	avg_review_score
cds_dvds_musicals	4.64
fashion_childrens_clothes	4.50
books_general_interest	4.45
costruction_tools_tools	4.44
flowers	4.42
books_imported	4.40
books_technical	4.37
food_drink	4.32
luggage_accessories	4.32
small_appliances_home_oven_and_coffee	4.30

ADVANCED SQL QUESTIONS (Business & Strategy)

1. What is the month-over-month (MoM) revenue growth rate?

```
with monthly_revenue as (
select
date_format(o.order_purchase_timestamp,'%m') as month,
sum(op.payment_value) as revenue
from orders o
join order_payments op on o.order_id = op.order_id
group by month
)
select
month,
revenue,
lag(revenue) over (order by month) as prev_month_revenue,
round(((revenue-lag(revenue) over (order by month))/lag(revenue) over (order by month)) * 100, 2) as mom_growth_rate
from monthly_revenue
order by month;
```

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

month	revenue	prev_month_revenue	mom_growth_rate
01	1253492.219227925	NULL	NULL
02	1284371.349671781	1253492.219227925	2.46
03	1609515.719184205	1284371.349671781	25.32
04	1578573.5101841632	1609515.719184205	-1.92
05	1746900.9712239727	1578573.5101841632	10.66
06	1535156.881066125	1746900.9712239727	-12.12
07	1658923.6683314238	1535156.881066125	8.06
08	1696821.641312493	1658923.6683314238	2.28
09	732454.2305314392	1696821.641312493	-56.83
10	839358.029732991	732454.2305314392	14.6
11	1194882.7997162547	839358.029732991	42.36
12	878421.0998720676	1194882.7997162547	-26.48

2. Does faster delivery correlate with higher review scores?

```
select
ore.review_score, avg(datediff(o.order_delivered_customer_date, o.order_purchase_timestamp)) as avg_delivery_days
from
orders o
join
order_reviews ore on o.order_id=ore.order_id
where
o.order_status = 'delivered'
and o.order_delivered_customer_date is not null
group by ore.review_score
order by ore.review_score desc;
```

Result Grid | Filter Rows:

review_score	avg_delivery_days
5	10.6254
4	12.2531
3	14.2024
2	16.6059
1	21.2519

3. Which customers have the highest lifetime value (CLV)?

```
SELECT
  o.customer_id,
  SUM(oi.price) AS customer_lifetime_value
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_id
WHERE o.order_status = 'delivered'
GROUP BY o.customer_id
ORDER BY customer_lifetime_value DESC limit 10;
```

Result Grid | Filter Rows: | Export:

customer_id	customer_lifetime_value
1617b1357756262bfa56ab541c47bc16	13440
ec5b2ba62e574342386871631fafd3fc	7160
c6e2731c5b391845f6800c97401a43a9	6735
f48d464a0baaea338cb25f816991ab1f	6729
3fd6777bbce08a352fddd04e4a7cc8f6	6499
05455dfa7cd02f13d132aa7a6a9729c6	5934.599853515625
df55c14d1476a9a3467f131269c2477f	4799
24bbf5fd2f2e1b359ee7de94defc4a15	4690
3d979689f636322c62418b6346b1c6d2	4590
cc803a2c412833101651d3f90ca7de24	4400

4. What percentage of orders are delivered late?

```
SELECT
  ROUND(
    COUNT(CASE
      WHEN order_delivered_customer_date > order_estimated_delivery_date
      THEN 1 END) * 100.0 / COUNT(*), 2
  ) AS late_delivery_percentage
FROM orders
WHERE order_status = 'delivered';
```

Result Grid | Filter R

late_delivery_percentage
8.11

5. Which product categories account for 80% of total revenue?

```
WITH category_revenue AS (
  SELECT
    p.product_category_name_english,
    SUM(oi.price) AS revenue
  FROM order_items oi
  JOIN products p ON oi.product_id = p.product_id
  JOIN orders o ON oi.order_id = o.order_id
  WHERE o.order_status = 'delivered'
  GROUP BY p.product_category_name_english
)
SELECT
  product_category_name_english,
  revenue,
  ROUND(revenue * 100.0 / SUM(revenue) OVER (), 2) AS revenue_pct
FROM category_revenue
ORDER BY revenue DESC limit 5;
```

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

product_category_name_english	revenue	revenue_pct
health_beauty	1233131.7208693027	9.45
watches_gifts	1166176.9777069092	8.94
bed_bath_table	1023434.7600488663	7.85
sports_leisure	954852.5489358902	7.32
computers_accessories	888724.6074113846	6.81

6. What is the top 10 Sellers' Revenue Concentration (in %)?

```
WITH seller_revenue AS (
   SELECT
      seller_id,
      SUM(price) AS revenue
   FROM order_items
   GROUP BY seller_id
),
ranked AS (
   SELECT
      seller_id,
      revenue,
      RANK() OVER (ORDER BY revenue DESC) AS rnk
   FROM seller_revenue
)
SELECT
   ROUND(SUM(revenue) * 100.0 /
       (SELECT SUM(revenue) FROM seller_revenue), 2)
       AS top_10_seller_revenue_pct
FROM ranked
WHERE rnk <= 10;
```