Challenge-7

Balanced Tree Clothing

By: Harshita Aswani

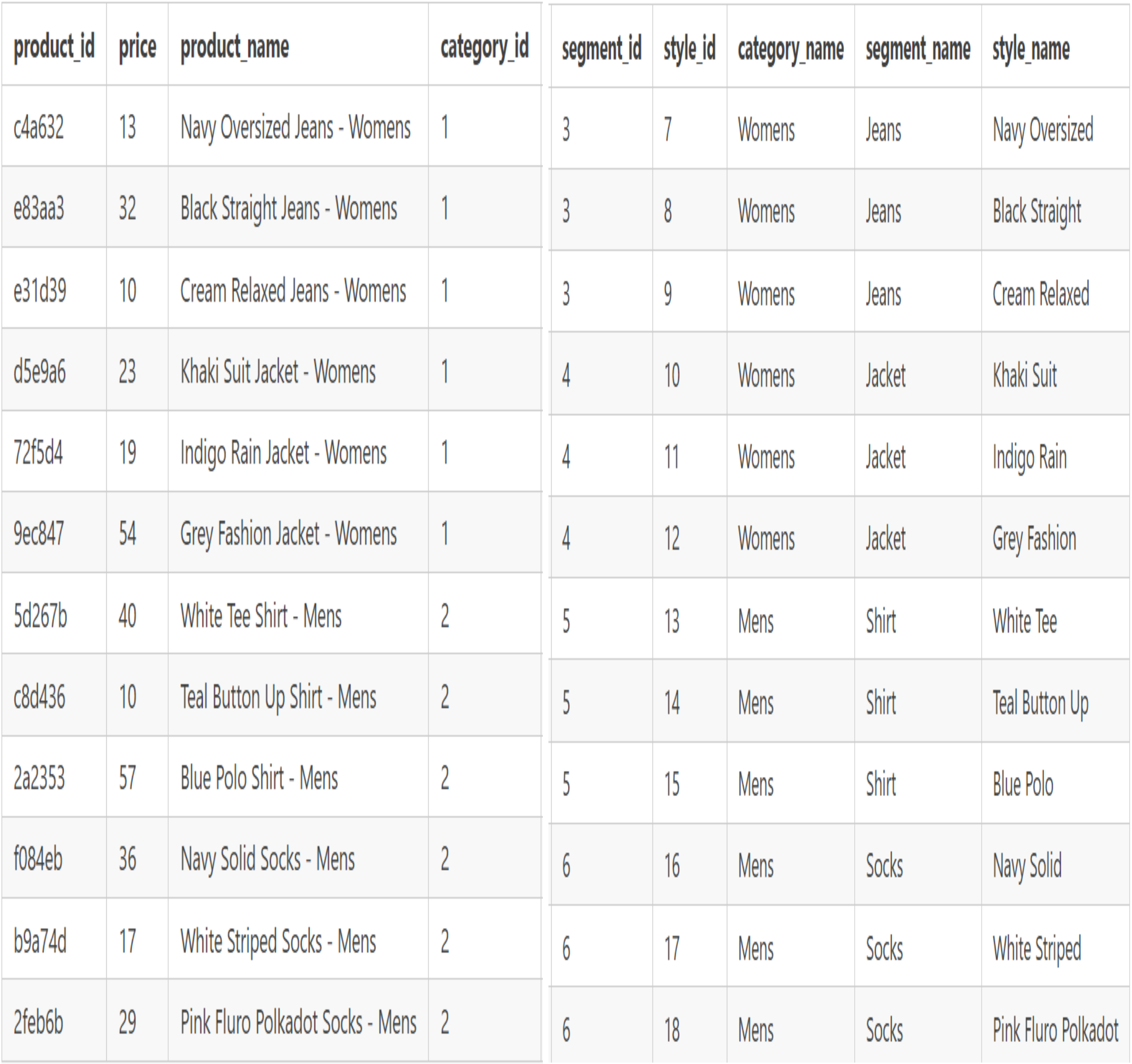
INTRODUCTION

Balanced Tree Clothing Company prides itself on providing an optimized range of clothing and lifestyle wear for the modern adventurer!

Danny, the CEO of this trendy fashion company has asked me to assist the team's merchandising teams in analysing their sales performance and generating a basic financial report to share with the wider business.

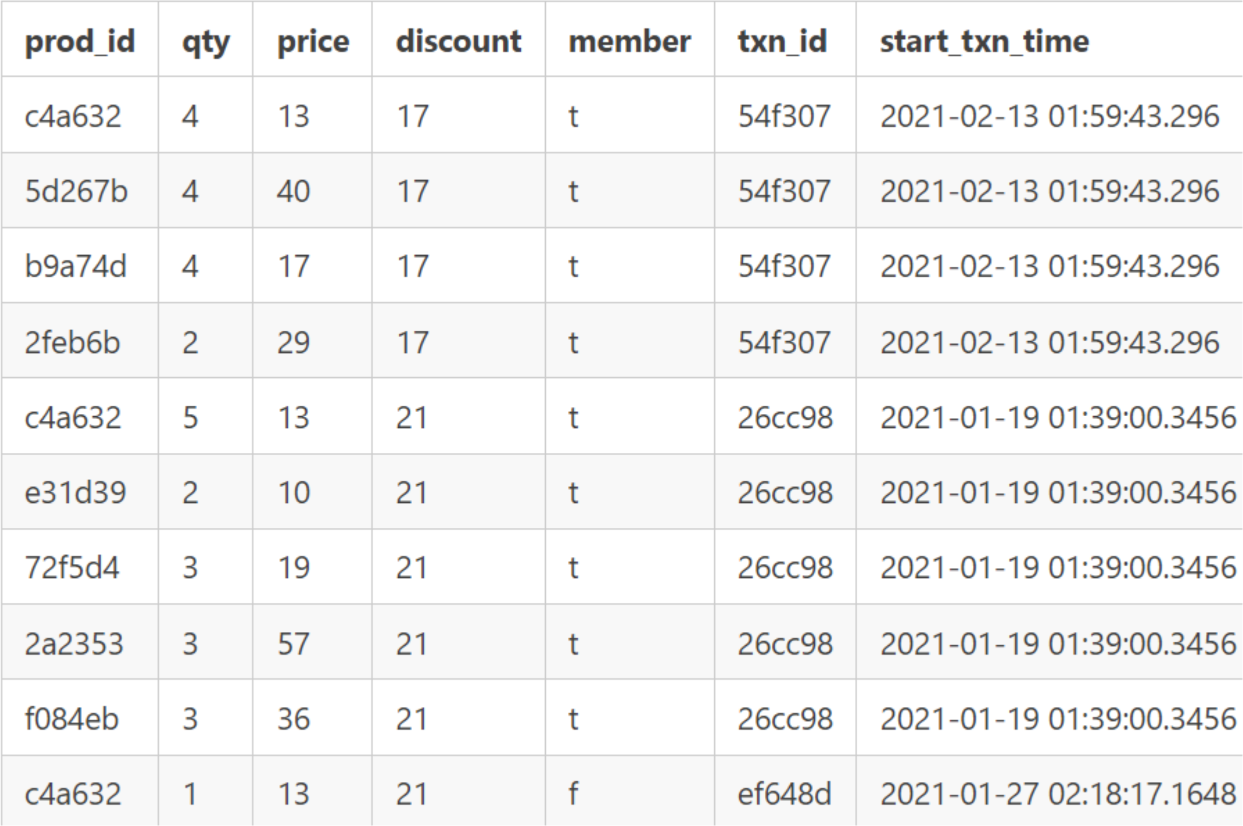
Tables

balanced_tree.product_details includes all information about the entire range that Balanced Clothing sells in their store.



product_id	price	product_name	category_id	segment_id	style_id	category_name	segment_name	style_name
c4a632	13	Navy Oversized Jeans - Womens	1	3	7	Womens	Jeans	Navy Oversized
e83aa3	32	Black Straight Jeans - Womens	1	3	8	Womens	Jeans	Black Straight
e31d39	10	Cream Relaxed Jeans - Womens	1	3	9	Womens	Jeans	Cream Relaxed
d5e9a6	23	Khaki Suit Jacket - Womens	1	4	10	Womens	Jacket	Khaki Suit
72f5d4	19	Indigo Rain Jacket - Womens	1	4	11	Womens	Jacket	Indigo Rain
9ec847	54	Grey Fashion Jacket - Womens	1	4	12	Womens	Jacket	Grey Fashion
5d267b	40	White Tee Shirt - Mens	2	5	13	Mens	Shirt	White Tee
c8d436	10	Teal Button Up Shirt - Mens	2	5	14	Mens	Shirt	Teal Button Up
2a2353	57	Blue Polo Shirt - Mens	2	5	15	Mens	Shirt	Blue Polo
f084eb	36	Navy Solid Socks - Mens	2	6	16	Mens	Socks	Navy Solid
b9a74d	17	White Striped Socks - Mens	2	6	17	Mens	Socks	White Striped
2feb6b	29	Pink Fluro Polkadot Socks - Mens	2	6	18	Mens	Socks	Pink Fluro Polkadot

balanced_tree.sales contains product level information for all the transactions made for Balanced Tree including quantity, price, percentage discount, member status, a transaction ID and also the transaction timestamp.



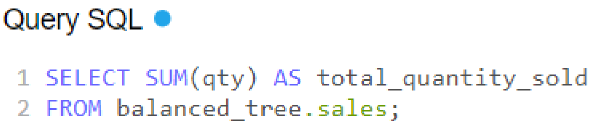
prod_id	qty	price	discount	member	txn_id	start_txn_time
c4a632	4	13	17	t	54f307	2021-02-13 01:59:43.296
5d267b	4	40	17	t	54f307	2021-02-13 01:59:43.296
b9a74d	4	17	17	t	54f307	2021-02-13 01:59:43.296
2feb6b	2	29	17	t	54f307	2021-02-13 01:59:43.296
c4a632	5	13	21	t	26cc98	2021-01-19 01:39:00.3456
e31d39	2	10	21	t	26cc98	2021-01-19 01:39:00.3456
72f5d4	3	19	21	t	26cc98	2021-01-19 01:39:00.3456
2a2353	3	57	21	t	26cc98	2021-01-19 01:39:00.3456
f084eb	3	36	21	t	26cc98	2021-01-19 01:39:00.3456
c4a632	1	13	21	f	ef648d	2021-01-27 02:18:17.1648

Case Study Questions

High Level Sales Analysis

I. What was the total quantity sold for all products?

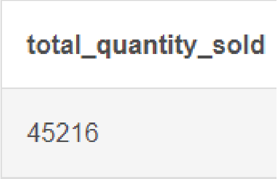
a.



Query SQL ●

```
SELECT SUM(qty) AS total_quantity_sold
FROM balanced_tree.sales;
```

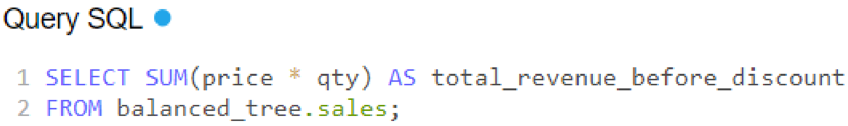
b.



total_quantity_sold
45216

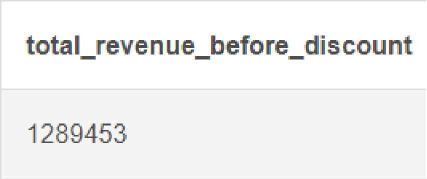
II. What is the total generated revenue for all products before discounts?

a.



Query SQL ●

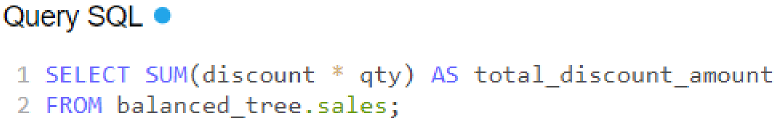
```
SELECT SUM(price * qty) AS total_revenue_before_discount
FROM balanced_tree.sales;
```



total_revenue_before_discount
1289453

b.

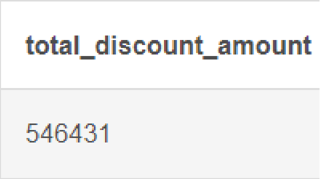
III. What was the total discount amount for all products?



Query SQL ●

```sql
SELECT SUM(discount * qty) AS total_discount_amount
FROM balanced_tree.sales;
```

a.

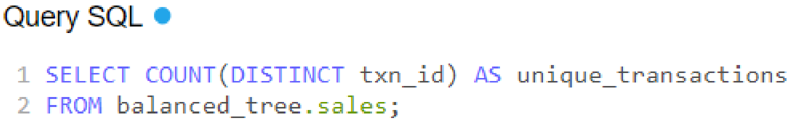


total_discount_amount
546431

b.

Transaction Analysis

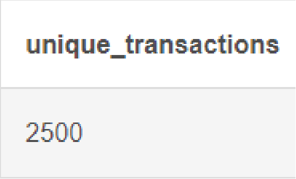
IV. How many unique transactions were there?



Query SQL ●

```sql
SELECT COUNT(DISTINCT txn_id) AS unique_transactions
FROM balanced_tree.sales;
```

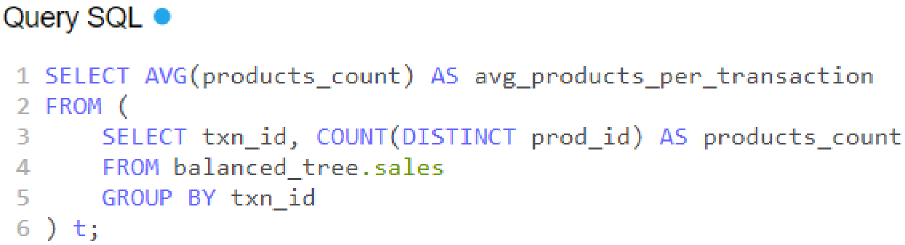
a.



unique_transactions
2500

b.

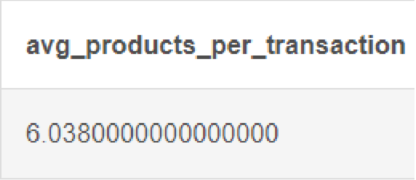
V. What is the average unique products purchased in each transaction?



Query SQL ●

```sql
SELECT AVG(products_count) AS avg_products_per_transaction
FROM (
    SELECT txn_id, COUNT(DISTINCT prod_id) AS products_count
    FROM balanced_tree.sales
    GROUP BY txn_id
) t;
```

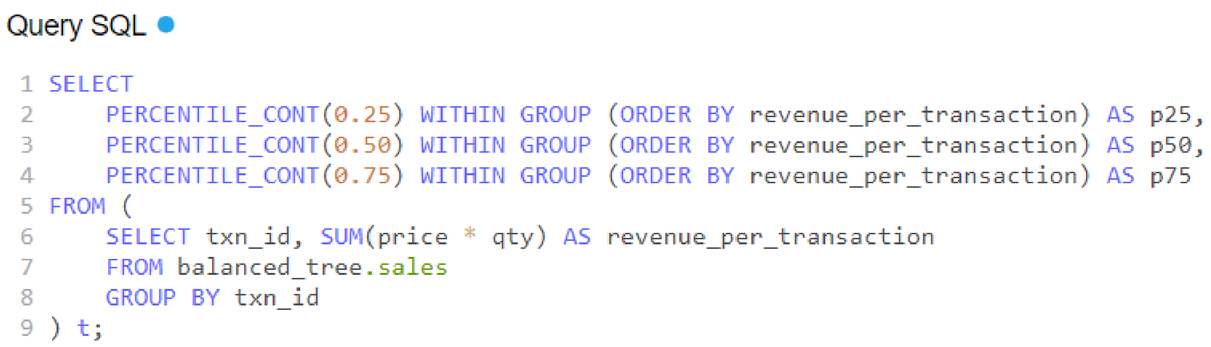
a.



avg_products_per_transaction
6.0380000000000000

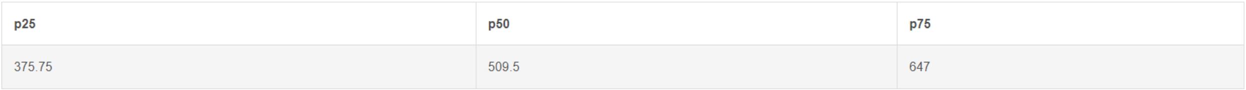
b.

VI. What are the 25th, 50th and 75th percentile values for the revenue per transaction?

a. 

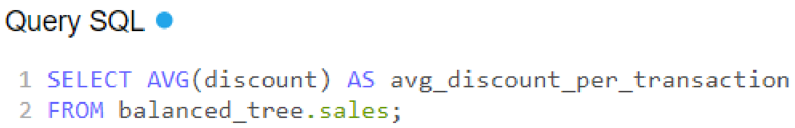
Query SQL ●

```sql
SELECT
    PERCENTILE_CONT(0.25) WITHIN GROUP (ORDER BY revenue_per_transaction) AS p25,
    PERCENTILE_CONT(0.50) WITHIN GROUP (ORDER BY revenue_per_transaction) AS p50,
    PERCENTILE_CONT(0.75) WITHIN GROUP (ORDER BY revenue_per_transaction) AS p75
FROM (
    SELECT txn_id, SUM(price * qty) AS revenue_per_transaction
    FROM balanced_tree.sales
    GROUP BY txn_id
) t;
```

b. 

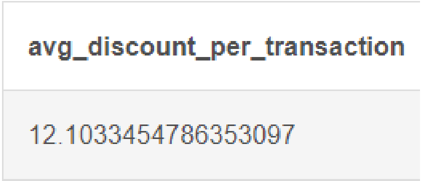
p25	p50	p75
375.75	509.5	647

VII. What is the average discount value per transaction?

a. 

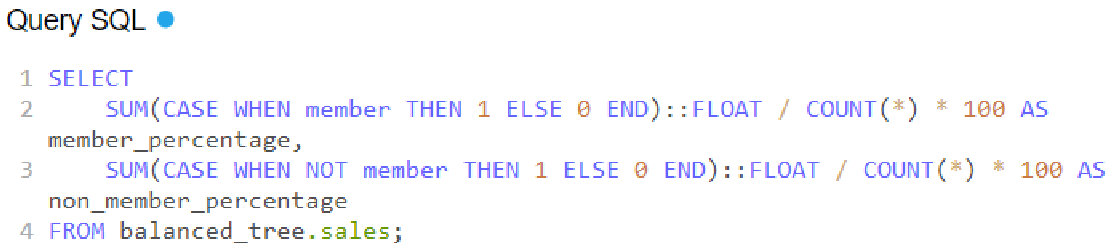
Query SQL ●

```sql
SELECT AVG(discount) AS avg_discount_per_transaction
FROM balanced_tree.sales;
```

b. 

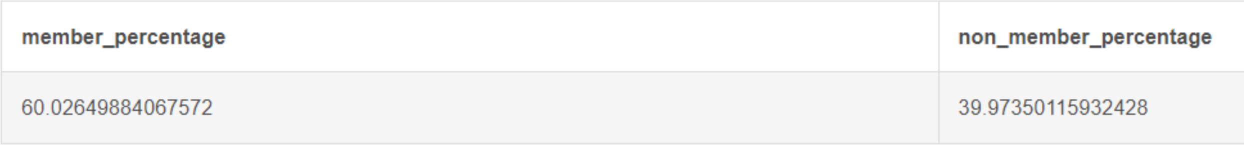
avg_discount_per_transaction
12.1033454786353097

VIII. What is the percentage split of all transactions for members vs non-members?

a. 

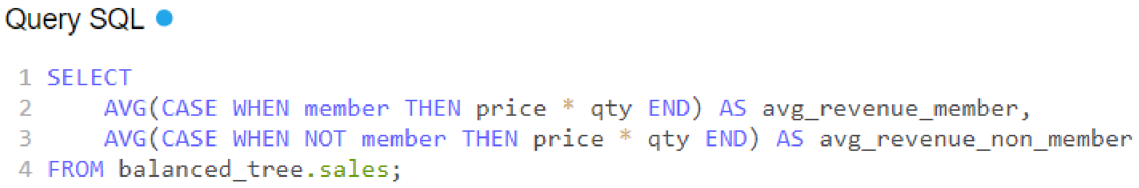
Query SQL ●

```sql
SELECT
    SUM(CASE WHEN member THEN 1 ELSE 0 END)::FLOAT / COUNT(*) * 100 AS
member_percentage,
    SUM(CASE WHEN NOT member THEN 1 ELSE 0 END)::FLOAT / COUNT(*) * 100 AS
non_member_percentage
FROM balanced_tree.sales;
```

b. 

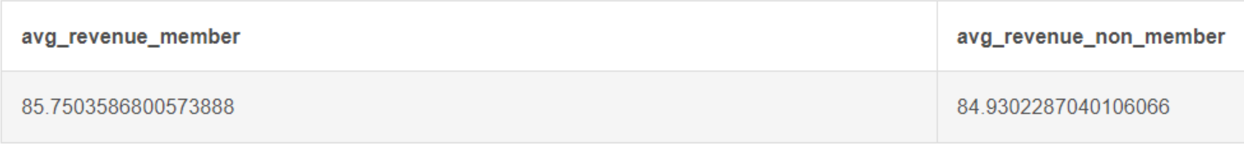
member_percentage	non_member_percentage
60.02649884067572	39.97350115932428

IX. What is the average revenue for member transactions and non-member transactions?

a. 

Query SQL ●

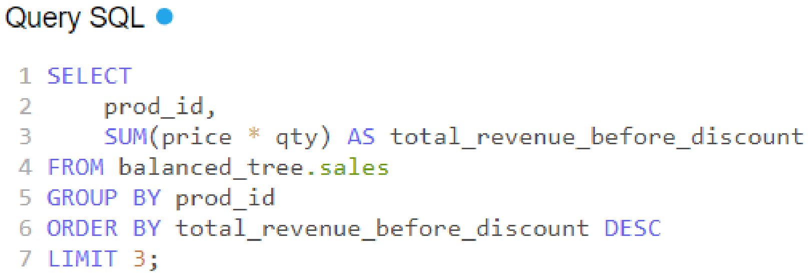
```sql
SELECT
    AVG(CASE WHEN member THEN price * qty END) AS avg_revenue_member,
    AVG(CASE WHEN NOT member THEN price * qty END) AS avg_revenue_non_member
FROM balanced_tree.sales;
```

b. 

avg_revenue_member	avg_revenue_non_member
85.7503586800573888	84.9302287040106066

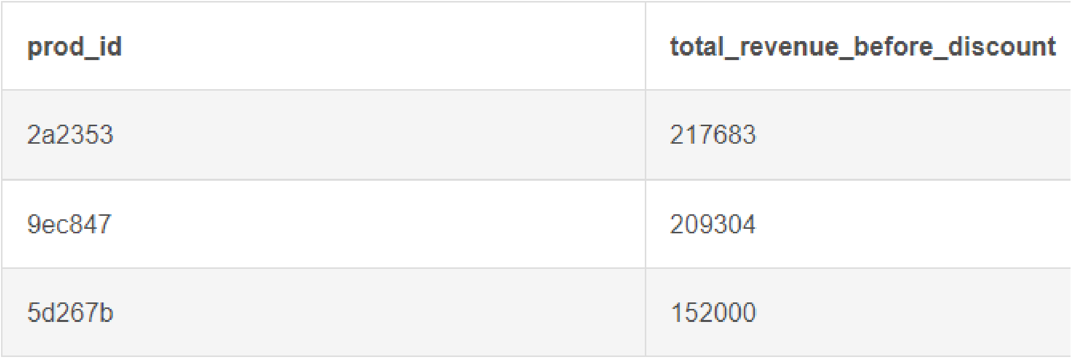
Product Analysis

X. What are the top 3 products by total revenue before discount?

a. 

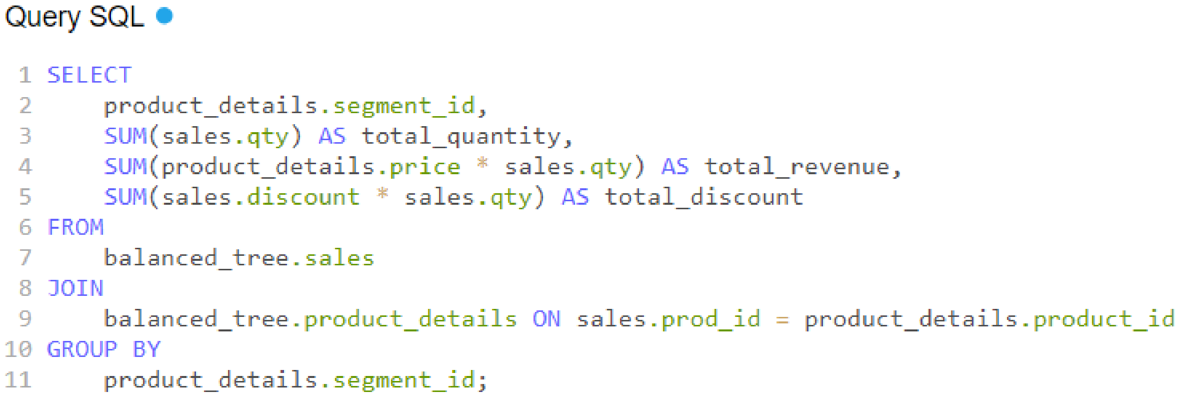
Query SQL ●

```sql
SELECT
    prod_id,
    SUM(price * qty) AS total_revenue_before_discount
FROM balanced_tree.sales
GROUP BY prod_id
ORDER BY total_revenue_before_discount DESC
LIMIT 3;
```

b. 

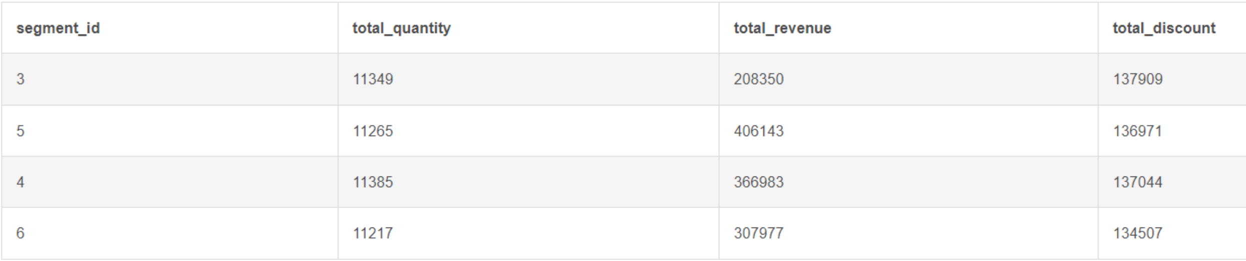
prod_id	total_revenue_before_discount
2a2353	217683
9ec847	209304
5d267b	152000

XI. What is the total quantity, revenue and discount for each segment?

a. 

Query SQL ●

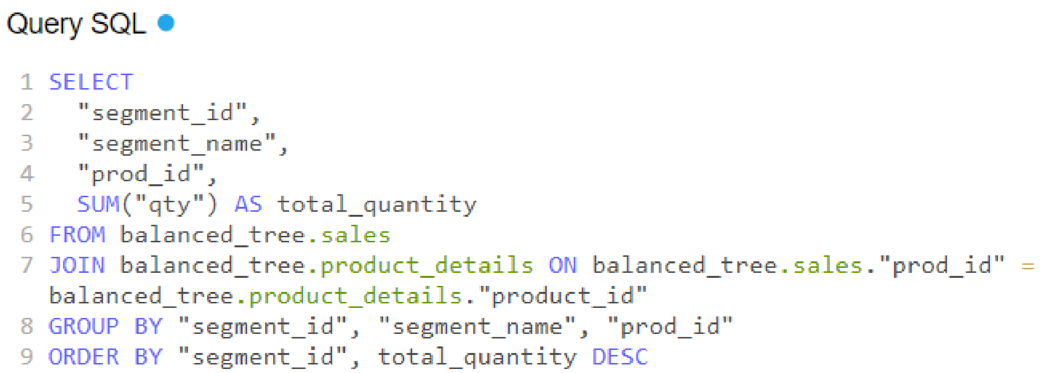
```sql
SELECT
    product_details.segment_id,
    SUM(sales.qty) AS total_quantity,
    SUM(product_details.price * sales.qty) AS total_revenue,
    SUM(sales.discount * sales.qty) AS total_discount
FROM
    balanced_tree.sales
JOIN
    balanced_tree.product_details ON sales.prod_id = product_details.product_id
GROUP BY
    product_details.segment_id;
```

b. 

segment_id	total_quantity	total_revenue	total_discount
3	11349	208350	137909
5	11265	406143	136971
4	11385	366983	137044
6	11217	307977	134507

XII. What is the top selling product for each segment?

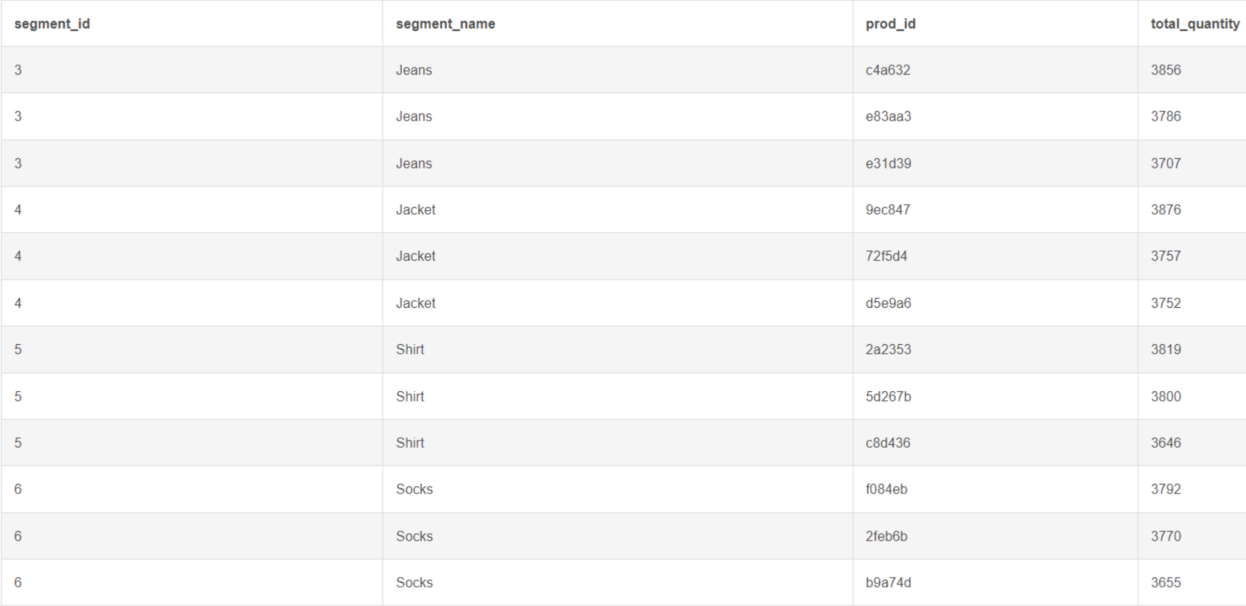
a.



Query SQL ●

```
SELECT
  "segment_id",
  "segment_name",
  "prod_id",
  SUM("qty") AS total_quantity
FROM balanced_tree.sales
JOIN balanced_tree.product_details ON balanced_tree.sales."prod_id" = balanced_tree.product_details."product_id"
GROUP BY "segment_id", "segment_name", "prod_id"
ORDER BY "segment_id", total_quantity DESC
```

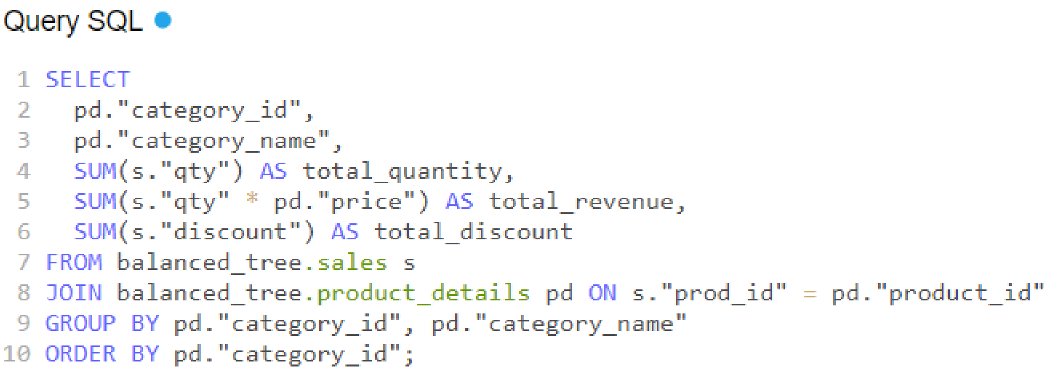
b.



segment_id	segment_name	prod_id	total_quantity
3	Jeans	c4a632	3856
3	Jeans	e83aa3	3786
3	Jeans	e31d39	3707
4	Jacket	9ec847	3876
4	Jacket	72f5d4	3757
4	Jacket	d5e9a6	3752
5	Shirt	2a2353	3819
5	Shirt	5d267b	3800
5	Shirt	c8d436	3646
6	Socks	f084eb	3792
6	Socks	2feb6b	3770
6	Socks	b9a74d	3655

XIII. What is the total quantity, revenue and discount for each category?

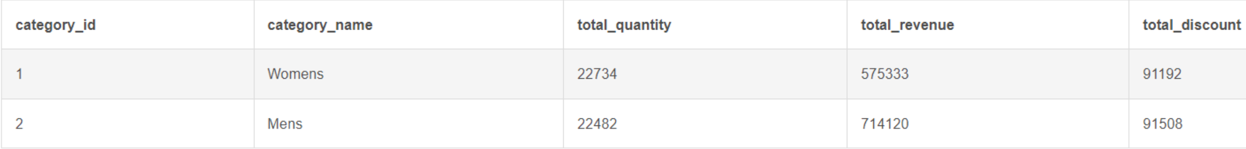
a.



Query SQL ●

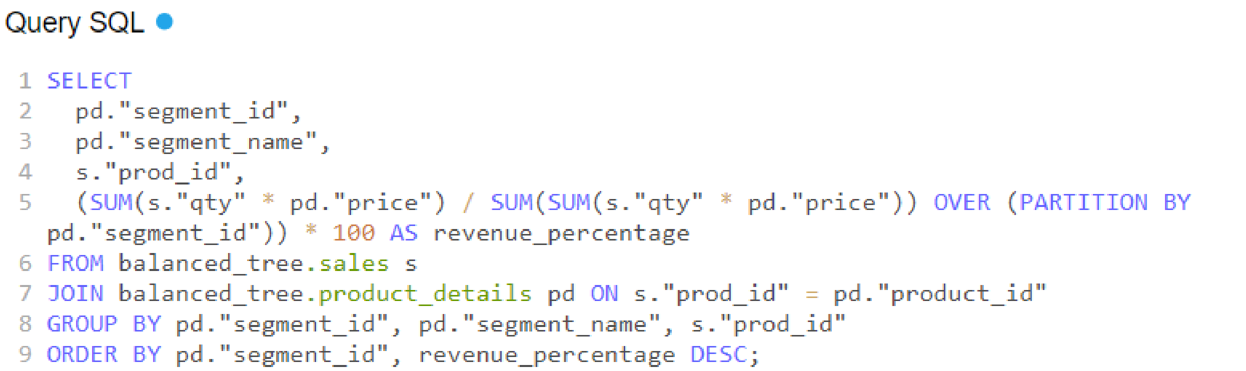
```
SELECT
  pd."category_id",
  pd."category_name",
  SUM(s."qty") AS total_quantity,
  SUM(s."qty" * pd."price") AS total_revenue,
  SUM(s."discount") AS total_discount
FROM balanced_tree.sales s
JOIN balanced_tree.product_details pd ON s."prod_id" = pd."product_id"
GROUP BY pd."category_id", pd."category_name"
ORDER BY pd."category_id";
```

b.

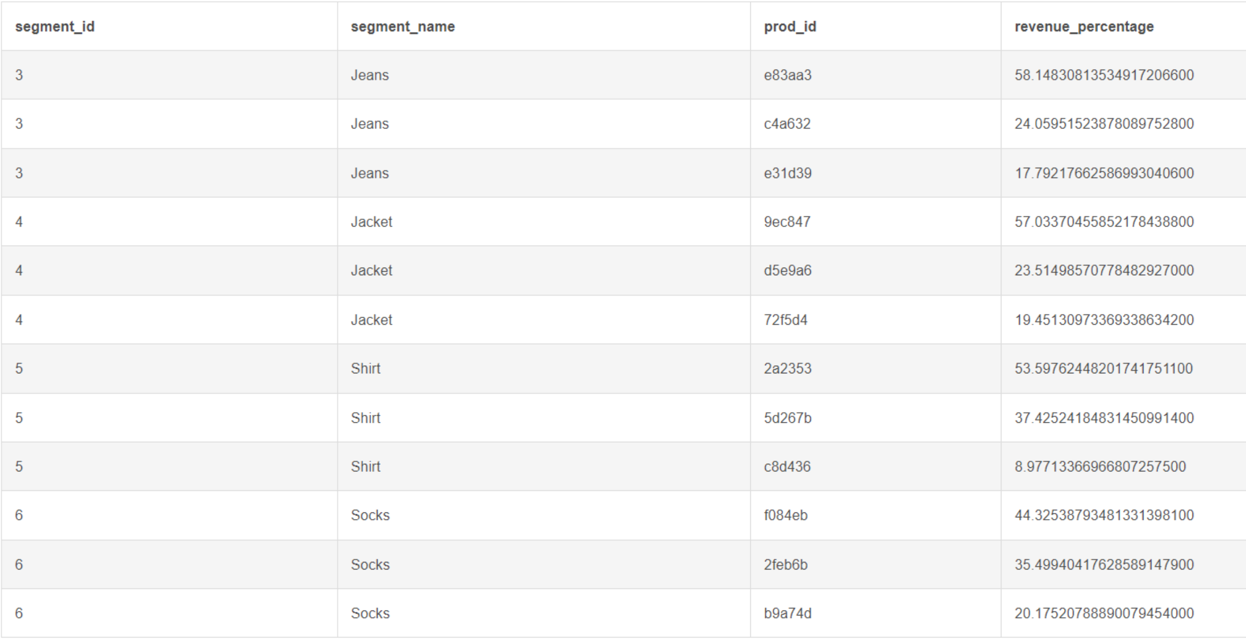


category_id	category_name	total_quantity	total_revenue	total_discount
1	Womens	22734	575333	91192
2	Mens	22482	714120	91508

XIV. What is the percentage split of revenue by product for each segment?

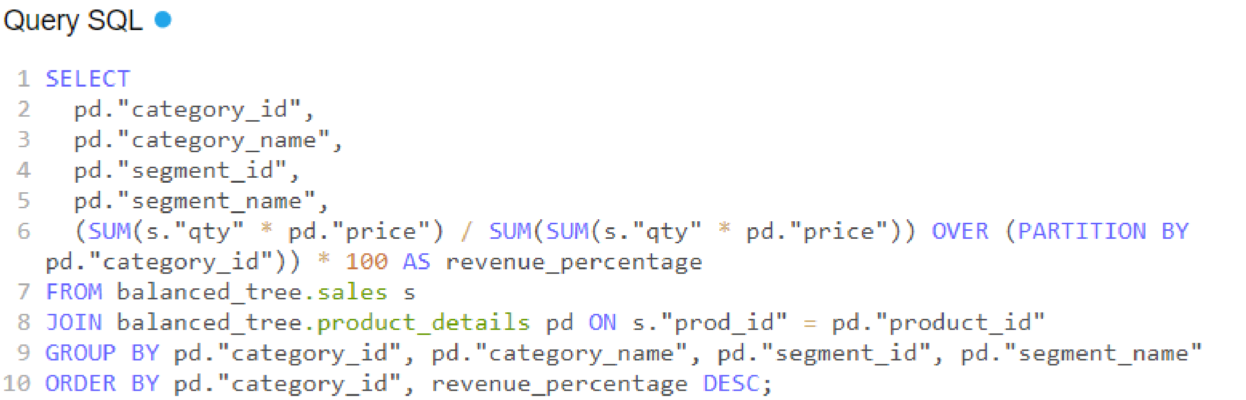
a. 
Query SQL ●

```
SELECT
  pd."segment_id",
  pd."segment_name",
  s."prod_id",
  (SUM(s."qty" * pd."price") / SUM(SUM(s."qty" * pd."price")) OVER (PARTITION BY pd."segment_id")) * 100 AS revenue_percentage
FROM balanced_tree.sales s
JOIN balanced_tree.product_details pd ON s."prod_id" = pd."product_id"
GROUP BY pd."segment_id", pd."segment_name", s."prod_id"
ORDER BY pd."segment_id", revenue_percentage DESC;
```

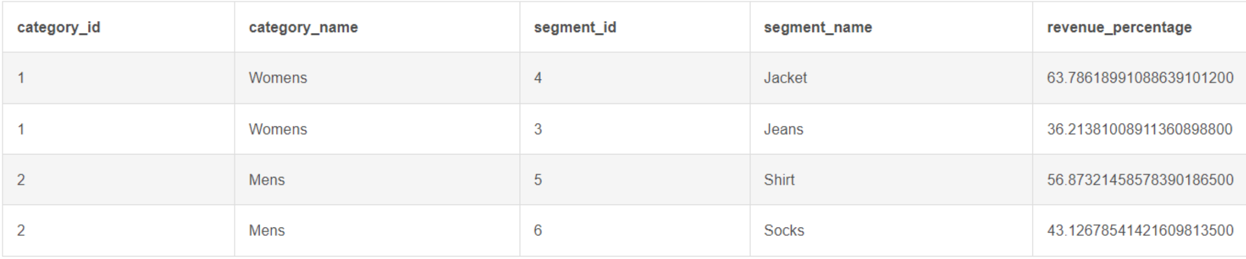
b. 

segment_id	segment_name	prod_id	revenue_percentage
3	Jeans	e83aa3	58.14830813534917206600
3	Jeans	c4a632	24.05951523878089752800
3	Jeans	e31d39	17.79217662586993040600
4	Jacket	9ec847	57.03370455852178438800
4	Jacket	d5e9a6	23.51498570778482927000
4	Jacket	72f5d4	19.45130973369338634200
5	Shirt	2a2353	53.59762448201741751100
5	Shirt	5d267b	37.42524184831450991400
5	Shirt	c8d436	8.97713366966807257500
6	Socks	f084eb	44.32538793481331398100
6	Socks	2feb6b	35.49940417628589147900
6	Socks	b9a74d	20.17520788890079454000

XV. What is the percentage split of revenue by segment for each category?

a. 
Query SQL ●

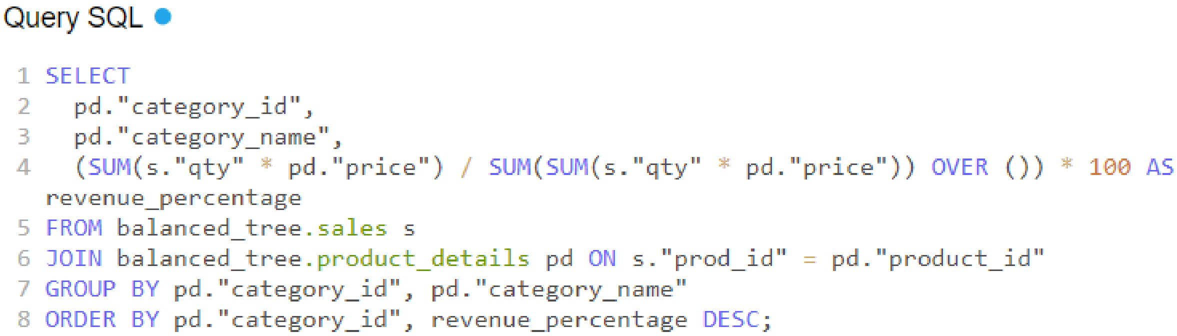
```
SELECT
  pd."category_id",
  pd."category_name",
  pd."segment_id",
  pd."segment_name",
  (SUM(s."qty" * pd."price") / SUM(SUM(s."qty" * pd."price")) OVER (PARTITION BY pd."category_id")) * 100 AS revenue_percentage
FROM balanced_tree.sales s
JOIN balanced_tree.product_details pd ON s."prod_id" = pd."product_id"
GROUP BY pd."category_id", pd."category_name", pd."segment_id", pd."segment_name"
ORDER BY pd."category_id", revenue_percentage DESC;
```

b. 

category_id	category_name	segment_id	segment_name	revenue_percentage
1	Womens	4	Jacket	63.78618991088639101200
1	Womens	3	Jeans	36.21381008911360898800
2	Mens	5	Shirt	56.87321458578390186500
2	Mens	6	Socks	43.12678541421609813500

XVI. What is the percentage split of total revenue by category?

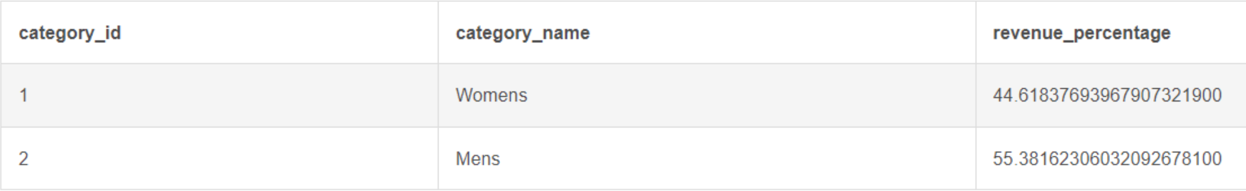
a.



Query SQL ●

```
SELECT
  pd."category_id",
  pd."category_name",
  (SUM(s."qty" * pd."price") / SUM(SUM(s."qty" * pd."price")) OVER ()) * 100 AS revenue_percentage
FROM balanced_tree.sales s
JOIN balanced_tree.product_details pd ON s."prod_id" = pd."product_id"
GROUP BY pd."category_id", pd."category_name"
ORDER BY pd."category_id", revenue_percentage DESC;
```

b.



category_id	category_name	revenue_percentage
1	Womens	44.61837693967907321900
2	Mens	55.38162306032092678100