## **CASE STUDY 2**

# **TEXTURE TALES**



## **INTRODUCTION:**

Texture Tales Clothing Company prides themselves on providing an optimized range of clothing and lifestyle wear for the modern adventurer!

I, the CEO of this trendy fashion company is asking you to assist the team's merchandising teams analyze their sales performance and generate a basic financial report to share with the wider business.

# create database tales use tales

```
-- Table: product_details

create table product_details (
    product_id varchar(20) primary key,
    price int not null,
    product_name varchar(100) not null,
    category_id int not null,
    segment_id int not null,
    style_id int not null,
    category_name varchar(50) not null,
    segment_name varchar(50) not null,
    style_name varchar(50) not null,
```

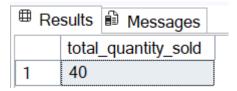
```
-- Table: product_prices
create table product_prices (
        id int primary key,
        product_id varchar(20) foreign key references product_details(product_id),
        price int not null
  );
  -- Table: product_hierarchy
create table product_hierarchy (
        id int primary key,
        product_id varchar(20) foreign key references product_details(product_id),
        price int not null,
        start_date date not null,
        end_date date not null
);
  -- Table: sales
r create table sales (
        prod_id varchar(20) foreign key references product_details(product_id),
        qty int not null,
        price int not null,
        discount int not null,
        member varchar(20) not null,
        txn_id varchar(30) primary key,
        start_txn_time varchar(30) not null
);
 insert into product_details (product_id, price, product_name, category_id, segment_id, style_id, category_name, segment_name, style_name)
 ('P001', 1200, 'T-Shirt', 1, 1, 1, 'Tops', 'Men', 'Casual'),
 ('P002', 1500, 'Jacket', 1, 2, 2, 'Tops', 'Women', 'Winter'),
 ('P003', 900, 'Shorts', 2, 1, 3, 'Bottoms', 'Men', 'Activewear'),
 ('P004', 1800, 'Hoodie', 1, 2, 2, 'Tops', 'Women', 'Winter'),
 ('P005', 500, 'Cap', 3, 1, 4, 'Accessories', 'Men', 'Caps'),
 ('P006', 2000, 'Jeans', 2, 2, 5, 'Bottoms', 'Women', 'Denim'),
 ('P007', 750, 'Scarf', 3, 2, 4, 'Accessories', 'Women', 'Scarves'),
 ('P008', 1700, 'Track Pants', 2, 1, 3, 'Bottoms', 'Men', 'Activewear'),
 ('P009', 2200, 'Sweater', 1, 2, 2, 'Tops', 'Women', 'Winter'),
 ('P010', 850, 'Tank Top', 1, 1, 1, 'Tops', 'Men', 'Casual'),
 ('P011', 1299, 'Rain Jacket', 1, 1, 6, 'Tops', 'Men', 'Jackets'),
 ('P012', 1599, 'Windbreaker', 1, 2, 6, 'Tops', 'Women', 'Jackets'),
 ('P013', 999, 'Cargo Shorts', 2, 1, 3, 'Bottoms', 'Men', 'Shorts'),
 ('P014', 1899, 'Puffer Jacket', 1, 2, 6, 'Tops', 'Women', 'Jackets'),
 ('P015', 599, 'Bandana', 3, 1, 7, 'Accessories', 'Men', 'Bandana'),
 ('P016', 1199, 'Fleece Pants', 2, 2, 5, 'Bottoms', 'Women', 'Pants'),
 ('P017', 1399, 'Travel Duffel', 3, 1, 8, 'Accessories', 'Men', 'Bags'),
 ('P018', 799, 'Sports Bra', 1, 2, 9, 'Tops', 'Women', 'Innerwear'),
 ('P019', 899, 'Beanie', 3, 2, 4, 'Accessories', 'Women', 'Caps'),
 ('P020', 1099, 'Thermal Top', 1, 1, 10, 'Tops', 'Men', 'Thermal');
```

```
-- INSERT INTO product_prices (20 rows)
insert into product_prices (id, product_id, price)
values
(1, 'P001', 1200),
(2, 'P002', 1500),
(3, 'P003', 900),
(4, 'P004', 1800),
(5, 'P005', 500),
(6, 'P006', 2000),
(7, 'P007', 750),
(8, 'P008', 1700),
(9, 'P009', 2200),
(10, 'P010', 850),
(11, 'P011', 1299),
(12, 'P012', 1599),
(13, 'P013', 999),
(14, 'P014', 1899),
(15, 'P015', 599),
(16, 'P016', 1199),
(17, 'P017', 1399),
(18, 'P018', 799),
(19, 'P019', 899),
(20, 'P020', 1099);
-- INSERT INTO product_hierarchy (20 rows)
insert into product_hierarchy (id, product_id, price, start_date, end_date)
values
(1, 'P001', 1200, '2023-01-01', '2023-06-30'),
(2, 'P002', 1500, '2023-01-01', '2023-06-30'),
(3, 'P003', 900, '2023-01-01', '2023-06-30'),
(4, 'P004', 1800, '2023-01-01', '2023-06-30'),
(5, 'P005', 500, '2023-01-01', '2023-06-30'),
(5, 'P005', 500, '2023-01-01', '2023-06-30'),
(6, 'P006', 2000, '2023-07-01', '2023-12-31'),
(7, 'P007', 750, '2023-07-01', '2023-12-31'),
(8, 'P008', 1700, '2023-07-01', '2023-12-31'),
(9, 'P009', 2200, '2023-07-01', '2023-12-31'),
(10, 'P010', 850, '2023-07-01', '2023-12-31'),
(11, 'P011', 1299, '2023-01-01', '2023-06-30')
(11, 'P011', 1299, '2023-01-01', '2023-06-30'), (12, 'P012', 1599, '2023-01-01', '2023-06-30'), (13, 'P013', 999, '2023-01-01', '2023-06-30'),
(14, 'P014', 1899, '2023-01-01', '2023-06-30'), (15, 'P015', 599, '2023-01-01', '2023-06-30'),
(16, 'P016', 1199, '2023-07-01', '2023-12-31'), (17, 'P017', 1399, '2023-07-01', '2023-12-31'),
(18, 'P018', 799, '2023-07-01', '2023-12-31'), (19, 'P019', 899, '2023-07-01', '2023-12-31'),
(20, 'P020', 1099, '2023-07-01', '2023-12-31');
```

```
-- INSERT INTO sales (20 rows)
insert into sales (prod_id, qty, price, discount, member, txn_id, start_txn_time)
  ('P001', 2, 1200, 10, 'Y', 'TXN001', '2023-01-15 09:00'),
  ('P002', 1, 1500, 0, 'N', 'TXN002', '2023-01-18 13:00'), ('P003', 3, 900, 5, 'Y', 'TXN003', '2023-02-10 16:45'),
  ('P004', 2, 1800, 20, 'Y', 'TXN004', '2023-02-20 11:30'),
  ('P005', 1, 500, 0, 'N', 'TXN005', '2023-03-05 10:15'),
  ('P006', 2, 2000, 15, 'Y', 'TXN006', '2023-03-10 14:50'),
  ('P007', 1, 750, 10, 'N', 'TXN007', '2023-04-12 12:00'),
  ('P008', 2, 1700, 5, 'Y', 'TXN008', '2023-04-25 15:40'), ('P009', 1, 2200, 10, 'Y', 'TXN009', '2023-05-08 09:20'),
  ('P010', 3, 850, 0, 'N', 'TXN010', '2023-05-18 17:30'), ('P011', 1, 1299, 10, 'Y', 'TXN016', '2023-01-20 10:30'), ('P012', 2, 1599, 15, 'Y', 'TXN017', '2023-02-18 09:45'),
  ('P013', 1, 999, 0, 'N', 'TXN018', '2023-02-25 17:15'),
  ('P014', 1, 1899, 20, 'Y', 'TXN019', '2023-03-02 11:30'),
  ('P015', 2, 599, 5, 'N', 'TXN020', '2023-03-12 19:45'),
  ('P016', 1, 1199, 10, 'Y', 'TXN021', '2023-04-10 08:10'), ('P017', 3, 1399, 10, 'Y', 'TXN022', '2023-04-12 15:20'),
  ('P018', 2, 799, 0, 'N', 'TXN023', '2023-05-01 14:30'), ('P019', 1, 899, 5, 'Y', 'TXN024', '2023-05-05 13:40'),
  ('P020', 2, 1099, 15, 'Y', 'TXN025', '2023-05-20 10:05'),
  ('P013', 1, 999, 0, 'N', 'TXN026', '2023-06-15 11:20'),
  ('P015', 1, 599, 0, 'Y', 'TXN027', '2023-06-25 12:40'),
  ('P016', 2, 1199, 10, 'N', 'TXN028', '2023-07-03 17:00'), ('P018', 1, 799, 0, 'N', 'TXN029', '2023-07-07 18:20'),
  ('P020', 1, 1099, 5, 'Y', 'TXN030', '2023-07-11 20:30');
```

#### -- Q1: Total quantity sold for all products

select sum(qty) as total\_quantity\_sold from sales;



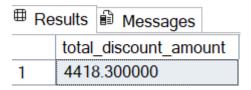
#### -- Q2: Total revenue before discounts

select sum(qty \* price) as total\_revenue\_before\_discount from sales;

```
⊞ Resultsi Messagestotal_revenue_before_discount1 48178
```

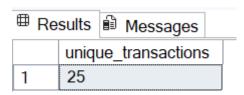
#### -- Q3: Total discount amount for all products

select sum((qty \* price) \* discount / 100.0) as total\_discount\_amount from sales;



#### -- Q4: Number of unique transactions

select count(distinct txn\_id) as unique\_transactions from sales;



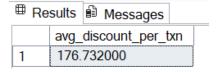
#### -- Q5: Average unique products purchased per transaction

select avg(product\_count) as avg\_unique\_products\_per\_txn from ( select txn\_id, count(distinct prod\_id) as product\_count from sales group by txn\_id ) as txn\_products;



#### -- Q6: Average discount value per transaction

select avg(discount\_value) as avg\_discount\_per\_txn from ( select txn\_id, sum((qty \* price) \* discount / 100.0) as discount\_value from sales group by txn\_id ) as txn\_discounts;



### -- Q7: Average revenue for member and non-member transactions

select member, avg(qty \* price) as avg\_revenue

from sales

group by member;

member		avg_revenue		
1	N	1329		
2	Y	2325		

#### -- Q8: Top 3 products by total revenue before discount

select top 3 prod\_id,

sum(qty \* price) as total\_revenue

from sales

group by prod\_id

order by total\_revenue desc;

■ Results				
	prod_id	total_revenue		
1	P017	4197		
2	P006	4000		
3	P004	3600		

#### -- Q9: Total quantity, revenue, and discount for each segment

select pd.segment\_name, sum(s.qty) as total\_qty, sum(s.qty \* s.price) as total\_revenue, sum((s.qty \* s.price) \* s.discount / 100.0) as total\_discount from sales s join product\_details pd on s.prod\_id = pd.product\_id group by pd.segment\_name;

<sup>⊞</sup> Re	sults 🖺 Message	es			
	segment_name	total_qty	total_revenue	total_discount	
1	Men	23	24138	1539.150000	
2	Women	17	24040	2879.150000	

#### -- Q10: Top selling product for each segment (by quantity)

with segment\_sales as (select pd.segment\_name, s.prod\_id, sum(s.qty) as total\_qty from sales s join product\_details pd on s.prod\_id = pd.product\_id group by pd.segment\_name, s.prod\_id),ranked\_segment as (select \*, rank() over (partition by segment\_name order by total\_qty desc) as rnk from segment\_sales) select segment\_name, prod\_id, total\_qty from ranked\_segment where rnk = 1;

■ Results  ■ Messages					
	segment_name	prod_id	total_qty		
1	Men	P003	3		
2	Men	P010	3		
3	Men	P015	3		
4	Men	P017	3		
5	Men	P020	3		
6	Women	P018	3		
7	Women	P016	3		

#### -- Q11: Total quantity, revenue, and discount for each category

select pd.category\_name, sum(s.qty) as total\_qty, sum(s.qty \* s.price) as total\_revenue, sum((s.qty \* s.price) \* s.discount / 100.0) as total\_discount from sales s join product\_details pd on s.prod\_id = pd.product\_id group by pd.category\_name;

⊞ Re	sults	Messages			
	category_name		total_qty	total_revenue	total_discount
1	Accessories		9	8143	599.550000
2	Bottoms		12	15695	1264.700000
3	Tops		19	24340	2554.050000

#### -- Q12: Top selling product for each category (by quantity)

with category\_sales as (select pd.category\_name, s.prod\_id, sum(s.qty) as total\_qty from sales s join product\_details pd on s.prod\_id = pd.product\_id group by pd.category\_name, s.prod\_id), ranked\_category as (select \*, rank() over (partition by category\_name order by total\_qty desc) as rnk from category\_sales) select category\_name, prod\_id, total\_qty from ranked\_category where rnk = 1;

⊞ Re	esults 🖺 Message	es	
	category_name	prod_id	total_qty
1	Accessories	P015	3
2	Accessories	P017	3
3	Bottoms	P003	3
4	Bottoms	P016	3
5	Tops	P010	3
6	Tops	P018	3
7	Tops	P020	3