Analytical SQL Project

Store data warehouse analysis

Submitted by

Abdelrahman Samir

Youssef Atef

Hendeya Rezk

Raghad Mahmoud

Yahia Khalaf

Data Science Intake 45

Task 1 : The data warehouse design

Fact table: Transaction Fact

This table captures the granular details of each customer transaction.

Column Name	Description
Transaction_ID	Unique identifier for each transaction.
Customer_ID	Foreign key linking to the Customer dimension.
Product_ID	Foreign key linking to the Product dimension.
Store_ID	Foreign key linking to the Store Location dimension.
Promotion_ID	Foreign key linking to the Promotion dimension (if applicable).
Time_ID	Foreign key linking to the Time dimension.
Quantity	Number of units purchased.
Total_Amount	Total amount spent in the transaction.
Discount_Amount	Discount applied (if any).
Net_Amount	Final amount paid after discounts.

Dimensions tables

1. Customer Dimension

This dimension provides details about the customers.

Column Name	Description
Customer_ID	Unique identifier for each customer.
Name	Customer's name.
Age	Customer's age.
Gender	Customer's gender.
Income_Level	Customer's income bracket (e.g., Low, Medium, High).
Loyalty_Status	Whether the customer is a loyalty program member (Yes/No).
Address	Customer's address.
City	Customer's city.
State	Customer's state.
Zip_Code	Customer's zip code.

2. Product Dimension

This dimension provides details about the products.

Column Name	Description
Product_ID	Unique identifier for each product.
Product_Name	Name of the product.
Category	Product category (e.g., Electronics, Grocery, Clothing).
Subcategory	Product subcategory (e.g., Laptops, Fruits, Men's Wear).
Brand	Brand of the product.
Unit_Price	Price of one unit of the product.

3. Store Location Dimension

This dimension provides details about the store locations.

Column Name	Description
Store_ID	Unique identifier for each store.
Store_Name	Name of the store.
City	City where the store is located.
State	State where the store is located.
Store_Size	Size of the store (e.g., Small, Medium, Large).
Region	Region where the store is located (e.g., Northeast, Midwest).

4. Promotion Dimension

This dimension provides details about the promotions.

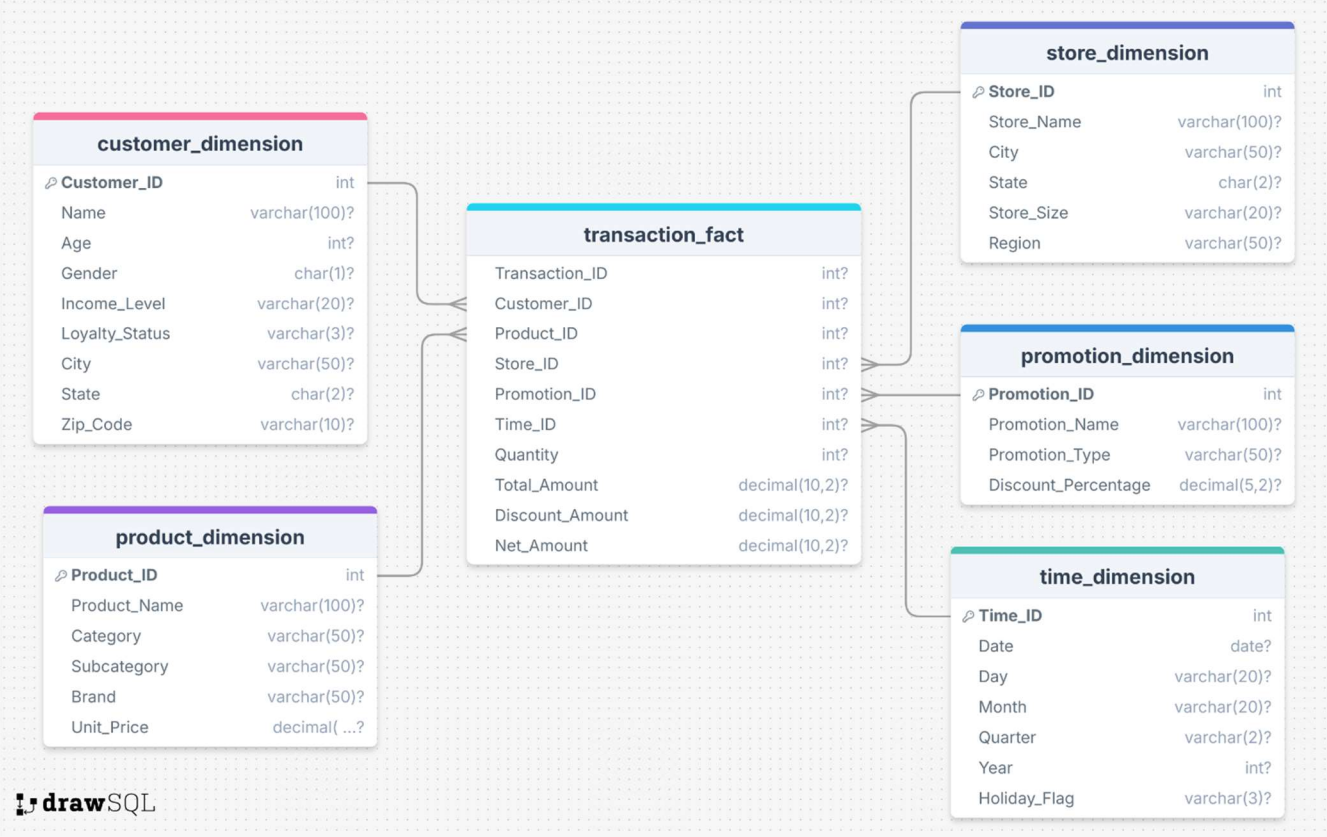
Column Name	Description
Promotion_ID	Unique identifier for each promotion.
Promotion_Name	Name of the promotion.
Promotion_Type	Type of promotion (e.g., Discount, BOGO, Bundle).
Discount_Percentage	Discount percentage offered (if applicable).

5. Time Dimension

This dimension provides details about the time of transactions.

Column Name	Description
Time_ID	Unique identifier for each time period.
Date	Date of the transaction.
Day	Day of the week (e.g., Monday, Tuesday).
Month	Month of the year.
Quarter	Quarter of the year (e.g., Q1, Q2).
Year	Year of the transaction.
Holiday_Flag	Indicates whether the date is a holiday (Yes/No).

Data warehouse Star Schema



Data generation

To populate the data warehouse, we used pandas and faker library and some publicly available datasets to generate the wanted data to test the SQL quires

Drive link to download the data and data warehouse creation script
https://drive.google.com/drive/u/0/folders/1Q2XUPOb2PM66-hymgnYGzP1jpz5gwrWs

What are the top-selling products in each category?

	product_name character varying (100)	category character varying (50)
1	Smartphone	Electronics
2	Coffee	Grocery
3	Shampoo	Personal Care
4	Sparkling Water	Beverages
5	Chips	Snacks
6	Laptop Bag	Electronics Accessories

```
with cte as (
select t.product_id, p.product_name, p.category, p.subcategory, t.net_amount
from transaction_fact t
join product_dimension p
on t.product_id = p.product_id), cte2 as (
select distinct product_name, category, rank() over(partition by category order by net_amount desc) as
rank
from cte)
select product_name, category
from cte2 where rank=1;
```

How do purchasing patterns change based on time or customer demographics?

To answer this question, we need to Break down purchasing patterns by **age groups**, then show **gender-based differences** as every age group and gender has its purchasing patterns, also we can **analyze income level** impact tracks monthly and yearly trends.

Note: because the data is artificially generated the result may not reflect the real world but we focused on the Analytical SQL it self

age-based analysis

	age_group text	total_transactions numeric	avg_transaction_value numeric
1	18-24	22005	120.98
2	25-34	39099	126.45
3	35-44	32164	127.39
4	45-54	36026	126.06
5	55+	85205	126.83

```sql
-- Customer Demographics Analysis
WITH Demographics_Analysis AS (
-- Age Group Analysis
SELECT
CASE
WHEN Age < 25 THEN '18-24'
WHEN Age BETWEEN 25 AND 34 THEN '25-34'
WHEN Age BETWEEN 35 AND 44 THEN '35-44'
WHEN Age BETWEEN 45 AND 54 THEN '45-54'
ELSE '55+'
END AS Age_Group,
c.Gender,
c.Income_Level,
td.Month,
td.Year,
COUNT(DISTINCT tf.Transaction_ID) as Total_Transactions,
SUM(tf.Quantity) as Total_Items_Purchased,
ROUND(SUM(tf.Net_Amount), 2) as Total_Spent,
ROUND(AVG(tf.Net_Amount), 2) as Avg_Transaction_Value
FROM transaction_fact tf
JOIN customer_dimension c ON tf.Customer_ID = c.Customer_ID
JOIN time_dimension td ON tf.Time_ID = td.Time_ID
GROUP BY
CASE
WHEN Age < 25 THEN '18-24'
WHEN Age BETWEEN 25 AND 34 THEN '25-34'
WHEN Age BETWEEN 35 AND 44 THEN '35-44'
WHEN Age BETWEEN 45 AND 54 THEN '45-54'
ELSE '55+'
END,
c.Gender,
c.Income_Level,
td.Month,
td.Year
)
SELECT
Age_Group,
SUM(Total_Transactions) as Total_Transactions,
ROUND(AVG(Avg_Transaction_Value), 2) as Avg_Transaction_Value
FROM Demographics_Analysis
GROUP BY Age_Group
ORDER BY Age_Group;
```

Analysis based on the month

10	Memory Card	Electronics Accessories	Storage	04-2023	1
11	Screen Protector	Electronics Accessories	Mobile Accessories	04-2023	2
12	Deodorant	Personal Care	Fragrance	04-2023	3
13	Memory Card	Electronics Accessories	Storage	05-2023	1
14	Screen Protector	Electronics Accessories	Mobile Accessories	05-2023	2
15	Deodorant	Personal Care	Fragrance	05-2023	3
16	Memory Card	Electronics Accessories	Storage	06-2023	1
17	Screen Protector	Electronics Accessories	Mobile Accessories	06-2023	2

```
with cte as(
select t.product_id, p.product_name, p.category, p.subcategory, to_char(d.date,'MM-YYYY') as
month_year, t.quantity
from transaction_fact t
join product_dimension p
on t.product_id = p.product_id
join time_dimension d
on t.time_id = d.time_id), cte2 as (
select product_name, category, subcategory, month_year, sum(quantity) over(partition by
product_name) as total_quantity
from cte), cte3 as (
select distinct product_name, category, subcategory, month_year, dense_rank() over(partition by
month_year order by total_quantity) as rank
from cte2)
select * from cte3 where rank <=3
order by month_year, rank;
```

Analysis based on gender

Top 3 purchased product by gender

	product_name character varying (100)	category character varying (50)	subcategory character varying (50)	gender character (1)	rank bigint
1	Memory Card	Electronics Accessories	Storage	F	1
2	Screen Protector	Electronics Accessories	Mobile Accessories	F	2
3	Deodorant	Personal Care	Fragrance	F	3
4	Memory Card	Electronics Accessories	Storage	M	1
5	Screen Protector	Electronics Accessories	Mobile Accessories	M	2
6	Deodorant	Personal Care	Fragrance	M	3

```
with cte as(
select t.product_id, p.product_name, p.category, p.subcategory, c.gender, t.quantity
from transaction_fact t
join product_dimension p
on t.product_id = p.product_id
join customer_dimension c
on c.customer_id = t.customer_id), cte2 as (
select product_name, category, subcategory, gender, sum(quantity) over(partition by product_name) as
total_quantity
from cte), cte3 as (
select distinct product_name, category, subcategory, gender, dense_rank() over(partition by gender
order by total_quantity) as rank
from cte2)
select * from cte3 where rank <=3
order by gender, rank;
```

Which types of promotions result in the highest sales?

	promotion_name character varying (100)	sales numeric
1	No Promotion	7681974.04
2	Summer Sale	6817209.15
3	Holiday Special	6448214.60
4	Clearance	6142793.24

We can see that **summer sale promotion** yield the highest sales

```
with cte as (
select t.product_id, t.net_amount, p.promotion_name
from transaction_fact t
join promotion_dimension p
on p.promotion_id = t.promotion_id)
select distinct promotion_name, sum(net_amount) over(partition by promotion_name) as sales
from cte order by sales desc;
```

popular combinations within the same category

	product1_name character varying (100)	product2_name character varying (100)	category character varying (50)	pairrank bigint
1	Juice	Tea	Beverages	1
2	Headphones	Bluetooth Speaker	Electronics	1
3	Phone Charger	USB Cable	Electronics Accessories	1
4	Memory Card	Screen Protector	Electronics Accessories	2
5	Milk	Bread	Grocery	1
6	Milk	Cereal	Grocery	2
7	Coffee	Cream	Grocery	3
8	Eggs	Rice	Grocery	4
9	Pasta	Sauce	Grocery	5
10	Peanut Butter	Jelly	Grocery	6
11	Pasta	Sugar	Grocery	7
12	Cereal	Sugar	Grocery	8
13	Cereal	Pasta	Grocery	9
14	Bread	Eggs	Grocery	10
15	Shampoo	Toothpaste	Personal Care	1
16	Soap	Deodorant	Personal Care	2

```sql
WITH ProductPairs AS (
SELECT
CASE
WHEN t1.Product_ID < t2.Product_ID THEN t1.Product_ID
ELSE t2.Product_ID
END AS Product1,
CASE
WHEN t1.Product_ID < t2.Product_ID THEN t2.Product_ID
ELSE t1.Product_ID
END AS Product2,
p1.Category AS Category,
COUNT(*) AS PurchaseCount
FROM transaction_fact t1
JOIN transaction_fact t2 ON t1.Transaction_ID = t2.Transaction_ID
JOIN product_dimension p1 ON t1.Product_ID = p1.Product_ID
JOIN product_dimension p2 ON t2.Product_ID = p2.Product_ID
WHERE t1.Product_ID <> t2.Product_ID AND p1.Category = p2.Category
GROUP BY
CASE
WHEN t1.Product_ID < t2.Product_ID THEN t1.Product_ID
ELSE t2.Product_ID
END,
CASE
WHEN t1.Product_ID < t2.Product_ID THEN t2.Product_ID
ELSE t1.Product_ID
END,
p1.Category
),
RankedPairs AS (
SELECT pp.Category,p1.Product_Name AS Product1_Name,p2.Product_Name AS Product2_Name,
ROW_NUMBER() OVER (PARTITION BY pp.Category ORDER BY pp.PurchaseCount DESC) AS PairRank
FROM ProductPairs pp
JOIN product_dimension p1 ON pp.Product1 = p1.Product_ID
JOIN product_dimension p2 ON pp.Product2 = p2.Product_ID
)
SELECT Product1_Name,Product2_Name,Category,PairRank
FROM RankedPairs
WHERE PairRank <= 10
ORDER BY Category, PairRank;
```

Show the number of times these combinations were purchased together, sorted by the most frequent pairs.

Using the Association rule mining, especially frequent itemset mining we found out the results are

	category character varying (50)	combination text	combination_count bigint	first_group text	second_group text
1	Beverages	Juices + Hot Drinks	2385	Juice	Tea
2	Electronics Accessories	Chargers + Cables	1013	Phone Charger	USB Cable
3	Electronics Accessories	Storage + Mobile Accessories	374	Memory Card	Screen Protector
4	Grocery	Dairy + Bakery	9840	Milk, Cream, Eggs	Bread
5	Grocery	Dairy + Breakfast	8735	Milk, Cream, Eggs	Cereal
6	Grocery	Beverages + Dairy	8535	Coffee	Milk, Cream, Eggs
7	Grocery	Dairy + Grains	8076	Milk, Cream, Eggs	Rice
8	Grocery	Pasta + Condiments	6017	Pasta	Sauce
9	Personal Care	Hair Care + Oral Care	687	Shampoo	Toothpaste
10	Personal Care	Bath + Fragrance	678	Soap	Deodorant

```
WITH Product_Categories AS (
-- First, group similar products using subcategory
SELECT p.Category,p.Subcategory, STRING_AGG(p.Product_Name, ', ') as Similar_Products,
COUNT(*) as Variation_Count
FROM product_dimension p
GROUP BY p.Category, p.Subcategory
),

Transaction_Pairs AS (
-- Get transactions with multiple items from same category
SELECT tf1.Transaction_ID,p1.Category,p1.Subcategory as Subcategory1,p2.Subcategory as
Subcategory2,
COUNT(*) as Purchase_Count
FROM transaction_fact tf1
JOIN transaction_fact tf2 ON
tf1.Transaction_ID = tf2.Transaction_ID AND
tf1.Product_ID < tf2.Product_ID  -- Avoid duplicate pairs
JOIN product_dimension p1 ON tf1.Product_ID = p1.Product_ID
JOIN product_dimension p2 ON tf2.Product_ID = p2.Product_ID
WHERE p1.Category = p2.Category  -- Same category items
AND p1.Subcategory != p2.Subcategory  -- Different subcategories
GROUP BY
tf1.Transaction_ID,
p1.Category,
p1.Subcategory,
p2.Subcategory
),

Popular_Combinations AS (
```

```sql
SELECT
t.Category,
t.Subcategory1,
t.Subcategory2,
COUNT(*) as Combination_Count,
-- Get sample products from each subcategory
MAX(pc1.Similar_Products) as Products1,
MAX(pc2.Similar_Products) as Products2,
RANK() OVER (PARTITION BY t.Category ORDER BY COUNT(*) DESC) as Popularity_Rank
FROM Transaction_Pairs t
JOIN Product_Categories pc1 ON t.Category = pc1.Category AND t.Subcategory1 = pc1.Subcategory
JOIN Product_Categories pc2 ON t.Category = pc2.Category AND t.Subcategory2 = pc2.Subcategory
GROUP BY t.Category,t.Subcategory1,t.Subcategory2
)

SELECT
Category,
Subcategory1 || ' + ' || Subcategory2 as Combination,
Combination_Count,
Products1 as First_Group,
Products2 as Second_Group
FROM Popular_Combinations
WHERE Popularity_Rank <= 5  -- Top 5 combinations per category
ORDER BY
Category,
Combination_Count DESC;
```

What are the most common product pairs our customers are buying?

Top 3 Highest-Volume Combinations:

- Dairy + Bakery (9,840 purchases)
- Dairy + Breakfast (8,735 purchases)
- Beverages + Dairy (8,535 purchases)

Key Insight: Dairy products are central to **most high-volume combinations**, appearing in 4 of the top combinations.

Which product pairs should we focus on for bundled promotions?

1. Grocery Department:
    - Create a "Breakfast Zone" combining:
        - Dairy section (Milk, Cream, Eggs)
        - Cereal aisle
        - Bread section
        - Coffee area
    - Position rice near the dairy section (8,076 combinations)
    - Place pasta and sauce displays together (6,017 combinations)
2. Beverages Section:
    - Position Juices near Hot Drinks (2,385 combinations)
    - Ensure easy access between beverage aisles and dairy section
3. Electronics Area:
    - Create an "Electronics Essentials" zone:
        - Chargers next to Cables (1,013 combinations)
        - Storage devices near Mobile Accessories (374 combinations)
4. Personal Care Section:
    - Position Hair Care products near Oral Care (687 combinations)
    - Create a personal grooming zone with Bath and Fragrance items (678 combinations)

Which product pairs should we focus on for bundled promotions?

1. High-Priority Bundle Promotions:
    - Dairy + Bakery bundles (highest combination count)
    - Breakfast combo deals (milk + cereal)
    - Coffee + dairy combinations
2. Category-Specific Promotions:
    - Electronics: Bundle deals on chargers with cables
    - Personal Care: Combined offers on shampoo + toothpaste
    - Beverages: Juice and tea combination offers

Also, we can make some low volume count deals to push sales, like
“Free phone case when you buy a smart phone”

Recommendations to reorganize the store layout, optimize promotions, and manage inventory.

Using the association rules like we used to reveal customer behaviors patterns to help group complementary products together

Also, by knowing high-demand products and ensuring they’re easily accessible in high-traffic areas by using Hotspot Analysis, also by studying sales trends over time to adjust the products layout accordingly

Optimizing Promotions

- **Personalized Offers:** Use customer segmentation to send tailored promotions based on past purchase behavior like loyalty rewards
- **Bundle Discounts:** Encourage larger purchases by offering discounts on bundles of frequently bought-together items.
- **Peak-Time Discounts:** Promote less popular items during peak hours to boost their sales.
- **Highlight First-Time Purchase Products:** Run special promotions on items often bought during customers' initial visits.

Inventory Management

- **Demand Forecasting:** Leverage sales trends and seasonality data to predict demand and adjust inventory levels proactively.
- **Replenishment Strategies:** Optimize restocking frequency for high-demand products to prevent stockouts.
- **Safety Stock:** Maintain buffer inventory for essential or fast-moving products based on sales data.
- **Slow-Moving Items:** Identify low-performing products and consider markdowns or bundle promotions to clear inventory.
- **Real-Time Monitoring:** Use automated inventory tracking systems to respond quickly to fluctuations in demand.

Conclusion

Strategic Bundle Recommendations to Boost Sales:

1. Primary Bundle Opportunities (High-Frequency Drivers):

A. Grocery Power Bundles:

- "Breakfast Essentials Bundle"
    - Dairy + Bakery (9,840 combinations)
    - Add slow-moving breakfast items like jams or spreads
    - Projected lift: 15-20% for auxiliary products
- "Morning Complete Package"
    - Dairy + Cereal (8,735 combinations)
    - Include slower-selling healthy toppings or fruits
    - Expected attachment rate: 25-30%

B. Beverage Cross-Promotions:

- "Daily Drinks Deal"
    - Juices + Hot Drinks (2,385 combinations)
    - Bundle with slower-moving specialty teas or coffee variants
    - Potential to lift specialty beverage sales by 30%

2. Technology Bundle Strategy:

- "Complete Device Care Package"
    - Chargers + Cables (1,013 combinations)
    - Add slower-moving accessories like power banks or cases
    - Expected attachment rate: 40%
- "Device Protection Bundle"
    - Storage + Mobile Accessories (374 combinations)

    - Include screen cleaning kits or phone stands
    - Projected auxiliary product lift: 25%
3. Personal Care Combination Deals:
- "Complete Care Package"
    - Hair Care + Oral Care (687 combinations)
    - Bundle with slower-moving items like hair treatments or specialty oral care
    - Expected lift in specialty products: 20%
4. Implementation Strategy:

A. Pricing Structure:
- 15% discount on bundle vs. individual items
- Additional 5% off when adding recommended slow-moving items
- Loyalty points multiplier for complete bundles

B. Display Strategy:
- End-cap displays featuring complete bundles
- Cross-category visibility
- Clear savings messaging

5. Expected Impact:
- Primary Products: 10-15% sales increase
- Slow-Moving Items: 25-30% sales lift
- Overall Basket Size: 20% increase
- Customer Satisfaction: Improved value perception

6. Monitoring Metrics:
- Bundle adoption rates
- Slow-moving product velocity
- Average transaction value
- Cross-category purchase frequency

Recommendation:
Implement tiered bundle strategy starting with top 3 grocery combinations in Month 1, followed by electronics and personal care in Month 2. This phased approach allows for optimization based on customer response."

Key Benefits:
- Accelerates slow-moving inventory
- Increases average transaction value
- Improves inventory turnover
- Enhances customer value proposition

A real-time dashboard for the upper management for real-time analysis and decision making