

Dream Super Shop SQL Analysis



Author: Shiva Prasad Akamgari



Email: prasadsiva81797@gmail.com



GitHub: github.com/Siva817



Portfolio: Siva817.github.io



Objective

This document showcases SQL-based analysis on the **Dream Super Shop** dataset. It demonstrates data analysis skills using SQL to extract insights, perform calculations, and summarize information from multiple related tables.



Topics Covered:

- 👤 Customer Insights
- 📦 Product & Sales Analysis
- 💰 Revenue Calculations
- 🏢 Branch Performance
- 📊 Advanced Analytics (Aggregation, Joins, Window Functions)



Datasets Used

Dataset Name	Description
Customer_Data	Customer details (ID, Name, Gender, Occupation, PremiumCustomer)
Product_Data	Product info (UnitCost, UnitPrice, SubCategoryKey)
Product_Categories_Data	Product category information
Product_Subcategories_Data	Subcategories linked to categories
Sales_Data	Sales transactions (Quantity, Date, Branch, Payment Method)
Calendar_Lookup	Date lookup for monthly, quarterly, yearly analysis



Instructions / Notes

- ✂️ SQL queries are written for **MySQL**, but logic applies to other relational databases.
- 📄 Each question includes the SQL query and a brief result description.
- 🔍 For large outputs, only relevant snippets are shown.

- 🎯 Goal: Showcase problem-solving, SQL proficiency, and data interpretation.
-

🌱 Beginner to Intermediate Questions

Q1. List all unique product categories available in the shop

```
-- CREATE OR REPLACE VIEW unique_product_categories AS
SELECT
    ROW_NUMBER() OVER (ORDER BY pc.categoryname, sc.subcategoryname)
    AS no,
    pc.categoryname,
    sc.subcategoryname
FROM product_categories_data pc
LEFT JOIN subcategories_data sc
    ON pc.categorykey = sc.categorykey
GROUP BY pc.categoryname, sc.subcategoryname;

SELECT * FROM unique_product_categories;
```

📌 Result :

no	categoryname	subcategoryname
1	Clothing	Children's Clothing
2	Clothing	Men's Clothing
3	Clothing	Women's Clothing
4	Electronics	Gadgets
5	Electronics	Mobile Accessories

Insights: There are **5 main categories** with multiple subcategories, showing diversity in the product catalog.

Q2. Find the total number of customers who are marked as PremiumCustomer

```
-- Select count(CustomerID) as total_premium_customers
From (
    Select CustomerID, CustomerName, PremiumCustomer
    From customer_data
    Where PremiumCustomer = 'Yes'
) as a;
```

✦ Result :

total_premium_customers
115

Insights: Approximately **115 customers** are premium, indicating the portion of high-value customers.

Q3. Get the top 5 most sold products based on quantity

-- -- Top 5 most sold products

```
SELECT
    p.ProductKey AS productID,
    p.ProductName AS productname,
    SUM(s.Quantity) AS total_quantity,
    pc.CategoryName,
    sc.SubCategoryName
FROM sales_data s
JOIN product_data p ON s.ProductID = p.ProductKey
LEFT JOIN subcategories_data sc ON p.SubCategoryKey =
sc.SubCategoryKey
LEFT JOIN product_categories_data pc ON sc.CategoryKey =
pc.CategoryKey
GROUP BY p.ProductKey, p.ProductName, pc.CategoryName,
sc.SubCategoryName
ORDER BY total_quantity DESC
LIMIT 5;
```

✦ Result

productID	productname	total_quantity	CategoryName	SubCategoryName
P059	Dinosaur Figure	569	Toys	Action Figures
P017	Rice	555	Food	Grains
P006	Potato	542	Food	Vegetables
P073	Headphones	538	Electronics	Mobile Accessories
P001	Milk	523	Food	Dairy

Insights: Food items dominate sales, but toys and electronics also have significant demand.

Q4. Calculate the total revenue generated (UnitPrice * Quantity) by each branch

```
-- Select
    s.Branch,
    SUM(s.Quantity * p.UnitCost) as total_revenue_generated
From sales_data s
Left Join product_data p
    ON s.ProductID = p.ProductKey
group by Branch
order by total_revenue_generated desc
```

📌 Result

Branch	total_revenue_generated
Dhaka	4967035
Sylhet	2483737
Khulna	1614234
Rajshahi	1501125
Chittagong	1073101
Barisal	523074
Mymensingh	471491
Rangpur	199655

Insights: Dhaka is the **highest revenue-generating branch**, suggesting higher sales volume or higher-priced products sold there.

Q5. List customers who have never made a purchase

```
-- Select distinct c.CustomerID, c.customername  
  
From customer_data c  
  
Left JOIn sales_data s ON c.CustomerID = s.CustomerID  
  
Where SalesID is null  
  
order by CustomerID
```

🔴 Result

CustomerID	customername
------------	--------------

Insights: All customers in the database have made at least one purchase, showing good customer engagement.

Q6. Show the most popular payment method used by customers

```
-- Select PaymentMethod, count(PaymentMethod) as transactions_count  
  
From sales_data  
  
group by PaymentMethod  
  
order by transactions_count desc
```

🔴 Result ,

PaymentMethod	transactions_count
Cash	7293
Debit Card	3622
Credit Card	1250

Insights: Cash is the preferred payment method, highlighting potential opportunities for digital payment adoption.

Q7. Count the number of products under each subcategory and category

```
-- -- query for number of products per subcategory
```

```
Select  
  
    row_number() over (order by subc.SubCategoryName) as idx,
```

```

        subc.SubCategoryName,
        count(p.ProductKey) as number_of_products
From product_data p
Left Join subcategories_data subc
        On p.SubCategoryKey = subc.SubCategoryKey
group by subc.SubCategoryName;

```

📌 Result

idx	SubCategoryName	number_of_products
1	Action Figures	3
2	Baby Products	3
3	Board Games	3
4	Children's Clothing	3
5	Cleaning Supplies	3

-- -- query for number of products per category

```

Select
        row_number() over (order by cat.CategoryName) as idx,
        cat.CategoryName,
        count(p.ProductKey) as number_of_products
From product_data p
Left Join subcategories_data subc
        On p.SubCategoryKey = subc.SubCategoryKey
Left Join product_categories_data cat
        On subc.CategoryKey = cat.CategoryKey
group by cat.CategoryName;

```

📌 Result

idx	CategoryName	number_of_products
1	Clothing	9
2	Electronics	10
3	Food	26
4	Household Items	9
5	Personal Care	9

Insights: The product distribution is balanced, with **Food** being the largest category.

Q8. Display the average unit price of products in each category

```
-- Select

    row_number() over (order by avg_unit_price) as idx,
    CategoryName,
    round(avg_unit_price,1) as avg_unit_price
From (
    Select
        cat.CategoryName,
        avg(p.UnitPrice) as avg_unit_price
    From product_data p
    left Join subcategories_data subcat
        On p.SubCategoryKey = subcat.SubCategoryKey
    Left Join product_categories_data cat
        On subcat.CategoryKey = cat.CategoryKey
    group by cat.CategoryName
) as category_avg
```

✦ Result

idx	CategoryName	avg_unit_price
1	Toiletries	88.9
2	Food	90.5
3	Toys	276.0
4	Household Items	372.3
5	Personal Care	382.4

Insights: Electronics and personal care products are higher-priced, while toiletries and food items are affordable.

Q9. Find the top 3 branches with the highest sales revenue

```
-- Select
```

```

        s.Branch,
        SUM(s.Quantity * p.UnitCost) as total_revenue_generated
From sales_data s
Left Join product_data p
        ON s.ProductID = p.ProductKey
group by Branch
order by total_revenue_generated desc
limit 3

```

📌 Result

Branch	total_revenue_generated
Dhaka	4967035
Sylhet	2483737
Khulna	1614234

Insights: Dhaka remains the strongest branch, consistent with Q4 revenue analysis.

Q10. Get the total number of male and female customers in each marital status group

```

-- Select  gender, MaritalStatus, count(MaritalStatus) as count
From customer_data
group by gender, MaritalStatus
order by gender, MaritalStatus

```

📌 Result

gender	MaritalStatus	count
Female	Married	45
Female	Single	56
Male	Married	49
Male	Single	50

Insights: Customer base is fairly evenly split between **male/female** and **married/single** groups, suggesting a diverse audience.

Advanced Questions

Q11. Find the customers who bought more than 50 different products

-- Select

```
    CustomerID,
    count( distinct ProductID) as different_products
From sales_data
group by CustomerID
Having different_products > 50
```

 Result

CustomerID	different_products
C009	51
C030	51
C038	52
C062	51
C122	51
C195	51

Insights: These are highly engaged customers, potentially **loyal shoppers or bulk buyers**.

Q12. Show the top 5 customers who spent the most money across all branches

-- SELECT

```
    CustomerID,
    Branch,
    SUM(s.Quantity * p.UnitPrice) AS total_spent
FROM sales_data s
LEFT JOIN product_data p ON s.ProductID = p.ProductKey
GROUP BY CustomerID, branch
ORDER BY total_spent DESC
LIMIT 5;
```

 Result

CustomerID	Branch	total_spent
C096	Khulna	150790
C020	Khulna	146351
C062	Rajshahi	130838
C190	Sylhet	123707
C200	Mymensingh	122718

Insights: High spenders can be targeted for **premium promotions or loyalty programs**.

Q13. List the products that were never sold

```
-- Select *
From product_data p
Left Join sales_data s
ON p.ProductKey = s.ProductID
Where s.ProductID is null
```

 Result

--

Insights: Every product was sold at least once, indicating **good product turnover**.

Q14. Calculate month-wise sales revenue using the Calendar lookup table

```
-- Select
    date_format(s.date, '%y-%m') as yearmonth,
    sum(s.quantity * p.UnitPrice) as total_revenue
From sales_data s
Left Join product_data p
On s.ProductID = p.ProductKey
group by yearmonth
```

 Result

yearmonth	total_revenue
24-05	1976228

24-06 1912777

24-07 2657532

24-08 2603925

24-09 2293899

Insights: July shows peak sales, possibly due to promotions or seasonal demand.

Q15. Identify the most profitable product (based on profit = UnitPrice - UnitCost)

```
-- Select

    RANK() over (order by ((UnitPrice - UnitCost) * 100/ UnitCost)
desc) as 'rank',

    ProductName,

    UnitCost,

    UnitPrice,

    UnitPrice - UnitCost as profit,

    concat(round(((UnitPrice - UnitCost) * 100/ UnitCost), 2),
'%') as profit_percentage
From product_data

limit 5
```

🔴 Result

rank	ProductName	UnitCost	UnitPrice	profit	percentage
1	Hair Conditioner	117	384	267	228.21%
2	Stuffed Rabbit	112	312	200	178.57%
3	Charger	151	397	246	162.91%
4	Onion	11	26	15	136.36%
5	Robot Toy	136	304	168	123.53%

Insights: Electronics and personal care items provide **higher profit margins**, while food items have lower margins.

Q16. Find the product category that generates the highest profit

```
-- Select
```

```

        rank() over (order by (sum((p.UnitPrice - p.UnitCost) *
s.Quantity)) DESC) as 'profit_rank',

        rank() over (order by (((sum(p.Unitprice * s.Quantity) -
sum(p.UnitCost * s.Quantity))/sum(p.UnitCost * s.Quantity)) * 100)
desc) as profit_perc_rank,

        cat.CategoryName,

        sum(p.UnitCost * s.Quantity) as total_unitcost,

        sum(p.Unitprice * s.Quantity) as total_unitprice,

        sum((p.UnitPrice - p.UnitCost) * s.Quantity) as total_profit,

        ((sum(p.Unitprice * s.Quantity) - sum(p.UnitCost *
s.Quantity))/sum(p.UnitCost * s.Quantity)) * 100 as
profit_percentage

From sales_data s

Left Join product_data p on s.ProductID = p.ProductKey

Left Join subcategories_data subcat on p.SubCategoryKey =
subcat.SubCategoryKey

Left Join product_categories_data cat on subcat.CategoryKey =
cat.CategoryKey

group by cat.CategoryName

order by profit_percentage desc

```

🔴 Result , , , , , ,

profit_rank	profit_perc_rank	CategoryName	total_unitcost	total_unitprice	total_profit	profit_percentage
3	1	Toys	711047	1138410	427363	60.1%
5	2	Food	664761	1009119	344358	51.8%
4	3	Household Items	1086676	1504194	417518	38.4%
7	4	Toiletries	273943	356463	82520	30.1%
1	5	Electronics	4995122	6392217	1397095	27.9%
6	6	Personal Care	1274508	1590846	316338	24.8%
2	7	Clothing	3827395	4408710	581315	15.18%

Insights: Toys and Food categories are most profitable relative to cost; Electronics generate high total profit but lower margin %.

Q17. Determine the number of sales per occupation group

-- Select

```

rank() over (order by (count (SalesID)) desc) as 'rank',

```

```

Occupation, count(SalesID) as number_of_sales
From sales_data s
Left Join customer_data cx ON s.CustomerID = cx.CustomerID
Group by Occupation

```

✦ Result

rank	Occupation	number_of_sales
1	Student	4880
2	Teacher	1534
3	Engineer	1234
4	Businessman	1203
5	Farmer	995
6	Housewife	929

Insights: Students are the primary buyers, indicating opportunities for student-focused marketing.

Q18. Calculate the average spending per customer per transaction

```

-- Select cx.CustomerName, round(avg(p.UnitPrice * s.Quantity),2) as
avg_amount
From sales_data s
Left Join product_data p on s.ProductID = p.ProductKey
Left Join customer_data cx on s.CustomerID = cx.CustomerID
group by cx.CustomerID, cx.CustomerName
order by avg_amount desc

```

✦ Result

CustomerName	avg_amount
Ashraful Hossain	2613.41
Zakia Mahmud	2432.10
Ashraful Nabi	2071.51
Nazmun Nahar	1963.94

Shakib Chowhan	1960.72
Anwar Kabir	1930.85

Insights: High average spending is concentrated among few customers, suggesting a **Pareto distribution of revenue**.

Q19. Find repeat customers who made purchases in more than one branch

```
-- Select CustomerID, count(distinct branch) as branch_count
From sales_data
group by CustomerID
having branch_count>1
```

📌 Result

CustomerID	branch_count

Insights: Customers tend to shop in a **single branch**, indicating branch loyalty or geographic constraints.

Q20. Use a window function to rank products by total sales within each subcategory

```
-- Select
    rank() over (order by (sum(s.Quantity)) desc) as 'rank',
    scat.SubCategoryName,
    sum(s.Quantity) as total_quantity
From sales_data s
Left Join product_data p on s.ProductID = p.ProductKey
Left Join subcategories_data scat on p.SubCategoryKey =
scat.SubCategoryKey
group by scat.SubCategoryName, scat.SubCategoryKey
order by total_quantity desc
```

📌 Result

rank	SubCategoryName	total_quantity

1	Vegetables	1923
2	Grains	1921
3	Dairy	1919
4	Meat and Fish	1868
5	Fruits	1775
6	Mobile Accessories	1507

Insights: Food subcategories dominate sales volume, highlighting the **importance of stocking staple items**.