



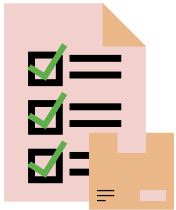
"Optimizing Inventory and Sales Strategies: A Comprehensive Analysis of Product Availability, Costs, and Sales Performance"



Overview of This Project

This project could focus on evaluating and optimizing inventory management, analyzing the relationship between product availability, costs, and sales performance to enhance overall business strategies.

About the Dataset



For completing this project, there are 3 datasets: Inventory, Product, and Sales. For inventory dataset, has 6 columns: Product ID, Store ID, Store Name, address, Neighborhood, and quantity available. The second one is a Product Table with 4 Columns in it: Product ID, Product Name, Supplier, And Product Cost. The last table is the Sales Table, with 6 columns: Sales ID, Store ID, Product ID, Date, Unit Price, and Quantity.

And the Primary Key for those three tables is Product ID.

Questions List



1. What is the total quantity available for each product in the inventory?
2. How many products are available in each store, and what product with the lowest quantity of Products per store?
3. What is the total quantity available for each neighborhood, and which neighborhood has the highest average product availability?
4. Which supplier provides products with the highest average cost, and how many products do they supply?
5. Which products have costs above the average cost?
6. Identify the top-selling products for each supplier based on sales quantity!
7. Calculate the total sales revenue for each product!
8. Determine the store with the highest total sales revenue by Product!
9. Determine the Top Product with the highest Revenue
10. Analyze yearly sales of product for a top Store!

--1.What is the total quantity available for each product in the inventory?

```
select inventory.ProductId AS Product_Id,
       product.ProductName AS product_name,
       SUM(inventory.QuantityAvailable) AS Total_Available
from `Sales.inventory` AS inventory
LEFT JOIN `Sales.product` AS product
ON inventory.ProductId = product.ProductId
GROUP BY inventory.ProductId, product.ProductName
order by inventory.ProductId;
```

Row	Product_Id	product_name	Total_Available
1	1	Chocolate Bar - Smarties	11
2	2	Pepper - Red Bell	1
3	3	Chickensplit Half	11
4	4	Zucchini - Green	1
5	5	Cod - Salted, Boneless	3
6	6	Flower - Dish Garden	9
7	7	Beef - Rouladin, Sliced	4
8	8	Cheese - Brie, Cups 125g	3

--2.How many products are available in each store, and what product with the lowest quantity per store?

```
with data1 AS (select DISTINCT StoreName, COUNT(ProductId) Total_Product
FROM `Sales.inventory`
GROUP BY StoreName),
data2 AS (select StoreName,ProductId, SUM(QuantityAvailable) Total
FROM `Sales.inventory` GROUP BY StoreName, ProductId),
data3 AS (Select StoreName, ProductId, Total,
              RANK() OVER(PARTITION BY StoreName ORDER BY Total ASC) AS Min_Quantity
FROM data2),
data5 AS (Select StoreName, CONCAT(ProductId,"_",Total) ProductId_TotalMinProduct
FROM data3 WHERE Min_Quantity = 1)
select data1.*, data5.ProductId_TotalMinProduct FROM
data1 LEFT JOIN data5 ON data1.StoreName = data5.StoreName;
```

Row	StoreName	Total_Product	ProductId_TotalMinProduct
1	Walmart	36	192_1
2	Walmart	36	652_1
3	Walmart	36	976_1
4	T.J. Maxx	38	97_1
5	T.J. Maxx	38	178_1
6	T.J. Maxx	38	278_1
7	T.J. Maxx	38	388_1
8	T.J. Maxx	38	716_1
9	T.J. Maxx	38	987_1
10	Dollar Tree	31	375_2

--3.What is the total quantity available for each neighborhood, and which neighborhood
--has the highest average product availability?

```
select DISTINCT neighborhood, Count(QuantityAvailable) AS Quantity_Available
FROM `Sales.inventory`
GROUP BY neighborhood
Order BY Quantity_Available DESC;
```

Row	neighborhood	Quantity_Available
1	Charles Village	57
2	Bridgeview/Greenlawn	46
3	Washington Village	44
4	Millhill	39
5	Seton Business Park	38
6	Ashburton	38
7	Sabina-Mattfeldt	36
8	Oliver	34
9	Central Park Heights	33
10	Frankford	32

--4.Which supplier provides products with the highest average cost,
--and how many products do they supply?

```
select Product.Supplier,
       ROUND(AVG(Product.ProductCost),2) AS Average_Cost,
       COUNT(inventory.ProductId) AS Total_Product
FROM `Sales.product` Product
LEFT JOIN `Sales.inventory` AS inventory
ON Product.ProductId = inventory.ProductId
GROUP BY Product.Supplier
Order by Average_Cost DESC;
```

Row	Supplier	Average_Cost	Total_Product
1	Burlington Coat Factory	2.38	31
2	Kmart	2.37	28
3	Ocean State Job Lot	2.31	27
4	Renys	2.28	30
5	Five Below	2.25	26
6	Target	2.21	22
7	Tuesday Morning	2.16	26
8	Gordmans	2.12	25
9	Shopko	2.08	39
10	Big Lots	2.06	21

--5. which products have costs above the average cost?

```
with product_abv AS (select DISTINCT ProductName,
                        AVG(ProductCost) Average_Cost
                        FROM `Sales.product`
                        GROUP BY ProductName)
select ProductName,Average_Cost
FROM product_abv
WHERE Average_Cost > (SELECT AVG(ProductCost) Average_Cost FROM `Sales.product`)
GROUP BY ProductName, Average_Cost
order by Average_Cost DESC;
```

Row	ProductName	Average_Cost
1	Broom - Corn	12.09
2	Sultanas	12.05
3	Flour - Rye	11.39
4	Pop Shoppe Cream Soda	9.62
5	Cookies - Englishbay Wht	8.84
6	Papadam	8.53
7	Chocolate - Liqueur Cups With ...	8.39

--6. Identify the top-selling products for each supplier based on sales quantity!

```
with SPQ AS (select product.Supplier AS Supplier,
                    product.ProductName AS product_name,
                    count(sales.Quantity) AS quantity,
                    FROM `Sales.product` AS product
                    LEFT JOIN `Sales.sales` AS sales
                    ON product.ProductId = sales.ProductId
                    GROUP BY Supplier, ProductName),
RANKING1 AS (select Supplier, product_name, quantity,
                    RANK() OVER(PARTITION BY Supplier ORDER BY quantity DESC) AS Ranking
                    FROM SPQ)
Select Supplier, product_name,quantity FROM RANKING1
WHERE Ranking = 1
ORDER BY RANKING1.Supplier ASC;
```

Row	Supplier	product_name	quantity
1	BJ's Wholesale Club	Muffin - Bran Ind Wrpd	402
2	Bargain Hunt	Sugar - Sweet N Low, Individual	228
3	Ben Franklin	Wine - Montecillo Rioja Crianza	407
4	Bi-Mart	Assorted Desserts	231
5	Big Lots	Soup - Knorr, French Onion	230
6	Burlington Coat Factory	Oil - Margarine	228
7	Costco	Beef - Bones, Marrow	228
8	Dd's Discounts	Versatainer Nc - 888	217

--7. Calculate the total sales revenue for each product!

```
with data1 AS (select DISTINCT product.ProductName,
    product.ProductCost, sales._UnitPrice_
    FROM `Sales.product` AS product
    LEFT JOIN `Sales.sales` AS sales
    ON product.ProductId = sales.ProductId
    ORDER BY ProductName),

    data2 AS (select product.ProductName, COUNT(sales.Quantity) Total_Quantity
    FROM `Sales.product` product
    LEFT JOIN `Sales.sales` sales ON sales.ProductId = product.ProductId
    GROUP BY product.ProductName)
select data1.ProductName,
ROUND(((data1._UnitPrice_ - data1.ProductCost)*data2.Total_Quantity),2) AS Revenue
FROM data1 LEFT JOIN data2 ON data1.ProductName = data2.ProductName;
```

Row	ProductName	Revenue
1	Pasta - Orzo, Dry	383.18
2	Cookies - Englishbay Wht	1359.15
3	Ecolab - Mikrokylene 4/4 L	161.92
4	Shark - Loin	226.24
5	Sultanas	1853.2
6	Foil - 4oz Custard Cup	247.04
7	Bread Base - Italian	27.16
8	Strawberries - California	819.0
9	Scallions - Live In Shell	587.82

--8. Determine the store with the highest total sales revenue by Product!

```
with data1 AS (select DISTINCT Sa.StoreId, Pr.ProductName,
    Pr.ProductId, (Sa._UnitPrice_ - Pr.ProductCost) AS Revenue, Sa.Quantity
    FROM `Sales.sales` AS Sa LEFT JOIN `Sales.product` AS Pr
    ON Sa.ProductId = Pr.ProductId),
    data2 AS (select StoreId, ProductId, SUM(Quantity) AS Tot_Quan
    FROM data1 GROUP BY StoreId, ProductId),
    data3 AS (select DISTINCT StoreId, ProductId, data1.Revenue
    FROM data1),
    data4 AS (Select data3.StoreId, data3.ProductId, (data3.Revenue*data2.Tot_Quan) Product_Rev
    FROM data3 LEFT JOIN data2
    ON (data3.StoreId = data2.StoreId AND data3.ProductId = data2.ProductId)
    GROUP BY data3.StoreId, data3.ProductId, Product_Rev), ---the key of query
    data5 AS (select StoreId, ProductId, Product_Rev, ROW_NUMBER() OVER(ORDER BY Product_Rev DESC) AS Num
    FROM data4)--for the Top Revenue by Product
Select * FROM data5 WHERE Num = 1;
```

Row	StoreId	ProductId	Product_Rev	Num
1	84879	117	7754.319999999...	1

```
--9.Determine the Top Product with the highest Revenue
with data1 AS (select DISTINCT Sa.StoreId, Pr.ProductName,
Pr.ProductId,(Sa._UnitPrice- Pr.ProductCost) AS Revenue, Sa.Quantity
FROM `Sales.sales` AS Sa LEFT JOIN `Sales.product` AS Pr
ON Sa.ProductId = Pr.ProductId),
> data2 AS (select StoreId, ProductId, SUM(Quantity) AS Tot_Quan...
> data3 AS (select DISTINCT StoreId, ProductId, data1.Revenue AS rev...
> data4 AS (select data3.StoreId, data3.ProductId, (data3.rev*data2.Tot_Quan) AS Rev_Product...
data5 AS (select *, ROW_NUMBER() OVER(ORDER BY Rev_Product DESC) Row_num from data4)
select * FROM data5 where Row_num = 1;
```

Row	StoreId	ProductId	Rev_Product	Row_num
1	84879	117	7754.319999999...	1

```
--10. Analyze yearly sales of product for a top Store!
--from our analysis before, we know that the top store is store with the Id (84879)
```

```
>with data1 AS (select ProductId, EXTRACT(YEAR FROM Date) AS Year, ...
select ProductId,
coalesce(Total_2017,0) AS Total_2017,
coalesce(Total_2018,0) AS Total_2018,
coalesce(Total_2019,0) AS Total_2019,
coalesce(Total_2020,0) AS Total_2020
FROM data7
ORDER BY ProductId;
```

Row	ProductId	Total_2017	Total_2018	Total_2019	Total_2020
1	1	87	0	31	124
2	2	161	53	196	200
3	3	172	37	0	296
4	4	16	17	215	69
5	6	304	74	128	68
6	7	107	140	109	30
7	8	178	0	116	0
8	10	45	57	204	208

Click [here](#) for the complete query and result

