

Final Data Warehouse Submission

< **IMT577_DW_APEKSHA_TEJWANI_STAGING_SNOWFLAKE_SHARE_1748492577147** ...

ACCOUNTADMIN Secure share Created: 3 minutes ago Azure - East US 2 (Virginia)

Shared With Add Consumers

NAME	ACCOUNT	TYPE	
'TARGETED ACCOUNTS'	—	Account	...

Description Add

A short summary of what this shared data includes and how it can be used.

Data Edit

IMT577_DW_APEKSHA_TEJWANI_STAGING 22 Tables, 15 Views

NAME	TYPE	SCHEMA
------	------	--------

--CREATING PASS TROUGH VIEWS

-- Dimension Table Views

CREATE OR REPLACE SECURE VIEW vw_Dim_Product AS

```
SELECT DimProductID, PRODUCTID, PRODUCTTYPEID, PRODUCTCATEGORYID, PRODUCTNAME,  
PRODUCTTYPE, PRODUCTCATEGORY, PRODUCT_RETAILPRICE, PRODUCT_WHOLESALEPRICE,  
PRODUCT_COST, PRODUCT_RETAILPROFIT, Product_WholesaleProfit,  
Product_ProfitMarginPercent
```

```
FROM Dim_Product;
```

```
SELECT * FROM vw_Dim_Product;
```

CREATE OR REPLACE SECURE VIEW vw_Dim_Channel AS

```
SELECT DimChannelID, ChannelID, ChannelCategoryID, ChannelName, ChannelCategory  
FROM Dim_Channel;
```

```
SELECT * FROM vw_Dim_Channel;
```

```
CREATE OR REPLACE SECURE VIEW vw_Dim_Location AS
```

```
SELECT DimLocationID, Address, City, State_Province, PostalCode, Country  
FROM Dim_Location;
```

```
SELECT * FROM vw_Dim_Location;
```

```
CREATE OR REPLACE SECURE VIEW vw_Dim_Store AS
```

```
SELECT DimStoreID, DimLocationID, StoreID, StoreNumber, StoreManager  
FROM Dim_Store;
```

```
SELECT * FROM vw_Dim_Store;
```

```
CREATE OR REPLACE SECURE VIEW vw_Dim_Reseller AS
```

```
SELECT DimResellerID, DimLocationID, ResellerID, ResellerName, Contact, PhoneNumber,  
EMAILADDRESS  
FROM Dim_Reseller;
```

```
SELECT * FROM vw_Dim_Reseller;
```

```
CREATE OR REPLACE SECURE VIEW vw_Dim_Customer AS
```

```
SELECT DimCustomerID, DimLocationID, CustomerID, CustomerFullName, CustomerFirstName,  
CustomerLastName, CustomerGender  
FROM Dim_Customer;
```

```
SELECT * FROM vw_Dim_Customer;
```

```
CREATE OR REPLACE SECURE VIEW vw_Dim_Date AS
```

```
SELECT DATE_PKEY, Date, DAY_NAME, DAY_NUM_IN_MONTH, MONTH_NAME,  
MONTH_NUM_IN_YEAR, FISCAL_YEAR
```

```
FROM Dim_Date;
```

```
SELECT * FROM vw_Dim_Date;
```

-- Fact Table Views

```
CREATE OR REPLACE SECURE VIEW vw_Fact_ProductSalesTarget AS
```

```
SELECT DimProductID, DimTargetDateID, ProductTargetSalesQuantity
```

```
FROM Fact_ProductSalesTarget;
```

```
SELECT * FROM vw_Fact_ProductSalesTarget;
```

```
CREATE OR REPLACE SECURE VIEW vw_Fact_SalesActual AS
```

```
SELECT DimProductID, DimStoreID, DimResellerID, DimCustomerID, DimChannelID,  
DimSaleDateID, DimLocationID, SalesHeaderID, SalesDetailID, SaleAmount, SaleQuantity,  
SaleUnitPrice, SaleExtendedCost, SaleTotalProfit
```

```
FROM Fact_SalesActual;
```

```
SELECT * FROM vw_Fact_SalesActual;
```

```
CREATE OR REPLACE SECURE VIEW vw_Fact_SRCSalesTarget AS
```

```
SELECT DimStoreID, DimChannelID, DimResellerID, DimTargetDateID, SalesTargetAmount
```

```
FROM Fact_SRCSalesTarget;
```

```
SELECT * FROM vw_Fact_SRCSalesTarget;
```

--CUSTOM VIEWS

--INSIGHT1:

--Give an overall assessment of stores number 10 and 21's sales.

--How are they performing compared to target? Will they meet their 2014 target?

--Should either store be closed? Why or why not?

-- 1. Store Performance & Target Comparison for 2014

```
CREATE OR REPLACE SECURE VIEW Q1 AS
```

```
WITH StoreSalesData AS (
```

```
    SELECT
```

```
        ds.StoreNumber,
```

```
        dd.FISCAL_YEAR,
```

```
        SUM(fsa.SaleAmount)/1000000 AS TotalSalesAmount
```

```
    FROM
```

```
        Fact_SalesActual fsa
```

```
        JOIN Dim_Store ds ON fsa.DimStoreID = ds.DimStoreID
```

```
        JOIN Dim_Date dd ON fsa.DimSaleDateID = dd.date_pkey
```

```
    WHERE
```

```
        ds.StoreNumber IN (10, 21)
```

```
        AND dd.fiscal_year = 2014
```

```
    GROUP BY
```

```
        ds.StoreNumber,
```

```

        dd.FISCAL_YEAR
    ),
    StoreTargetData AS (
        SELECT
            ds.StoreNumber,
            dd.FISCAL_YEAR,
            SUM(fst.SalesTargetAmount)/1000000 AS TotalTargetSales
        FROM
            Fact_SRCSalesTarget fst
            JOIN Dim_Store ds ON fst.DimStoreID = ds.DimStoreID
            JOIN Dim_Date dd ON fst.DimTargetDateID = dd.date_pkey
        WHERE
            ds.StoreNumber IN (10,21)
            AND dd.fiscal_year = 2014
        GROUP BY
            ds.StoreNumber,dd.FISCAL_YEAR
    )
    SELECT
        ssd.StoreNumber,
        ssd.fiscal_Year,
        ssd.TotalSalesAmount,
        std.TotalTargetSales,
        std.TotalTargetSales - ssd.TotalSalesAmount as deviation
    FROM
        StoreSalesData ssd
        JOIN StoreTargetData std ON ssd.StoreNumber = std.StoreNumber

```

ORDER BY

ssd.StoreNumber;

select * from Q1;

----2013 Bonus Pool Distribution (\$2,000,000)

CREATE OR REPLACE secure VIEW vw_StoreBonusDistribution_2013 AS

WITH ActualSales AS (

SELECT

ds.StoreNumber,

SUM(fsa.SaleAmount) / 1000000 AS ActualSales_Millions

FROM Fact_SalesActual fsa

JOIN Dim_Store ds ON fsa.DimStoreID = ds.DimStoreID

JOIN Dim_Date dd ON fsa.DimSaleDateID = dd.DATE_PKEY

WHERE dd.FISCAL_YEAR = 2013 AND ds.StoreNumber != -1

GROUP BY ds.StoreNumber

),

TargetSales AS (

SELECT

ds.StoreNumber,

SUM(t.SalesTargetAmount) / 1000000 AS TargetSales_Millions

FROM Fact_SRCSalesTarget t

JOIN Dim_Store ds ON t.DimStoreID = ds.DimStoreID

JOIN Dim_Date d ON t.DimTargetDateID = d.DATE_PKEY

```

WHERE d.FISCAL_YEAR = 2013 AND ds.StoreNumber != -1

GROUP BY ds.StoreNumber

),

Combined AS (

    SELECT

        a.StoreNumber,

        a.ActualSales_Millions,

        t.TargetSales_Millions,

        ROUND((a.ActualSales_Millions / NULLIF(t.TargetSales_Millions, 0)) * 100, 2) AS
PerformanceRatio

    FROM ActualSales a

    JOIN TargetSales t ON a.StoreNumber = t.StoreNumber

),

TotalRatio AS (

    SELECT SUM(PerformanceRatio) AS TotalScore FROM Combined

)

SELECT

    c.StoreNumber,

    ROUND(c.ActualSales_Millions, 2) AS ActualSales_Millions,

    ROUND(c.TargetSales_Millions, 2) AS TargetSales_Millions,

    ROUND(c.PerformanceRatio, 2) AS PerformanceRatio,

    ROUND((c.PerformanceRatio / tr.TotalScore) * 2000000, 2) AS RecommendedBonus

FROM Combined c

CROSS JOIN TotalRatio tr

ORDER BY RecommendedBonus DESC;

```

```
select * from vw_storebonusdistribution_2013;
```

--3. Assess product sales by day of the week at stores 10 and 21. What can we learn about sales trends?

```
CREATE OR REPLACE secure VIEW vw_StoreSalesByDayOfWeek AS
```

```
SELECT
```

```
    ds.StoreNumber,
```

```
    dd.DAY_NAME,
```

```
    SUM(fsa.SaleAmount) AS TotalSales
```

```
FROM Fact_SalesActual fsa
```

```
JOIN Dim_Store ds ON fsa.DimStoreID = ds.DimStoreID
```

```
JOIN Dim_Date dd ON fsa.DimSaleDateID = dd.date_pkey
```

```
WHERE ds.StoreNumber IN (10, 21)
```

```
GROUP BY dd.day_name, ds.storenumber
```

```
order by dd.day_name;
```

```
CREATE OR REPLACE secure VIEW vw_StoreSalesByDayPivot AS
```

```
SELECT
```

```
    dd.DAY_NAME AS Day,
```

```
    ROUND(SUM(CASE WHEN ds.StoreNumber = 10 THEN fsa.SaleAmount ELSE 0 END) /  
1000000, 2) || 'M' AS "Store 10",
```

```
    ROUND(SUM(CASE WHEN ds.StoreNumber = 21 THEN fsa.SaleAmount ELSE 0 END) /  
1000000, 2) || 'M' AS "Store 21"
```

```
FROM Fact_SalesActual fsa
```

```
JOIN Dim_Store ds ON fsa.DimStoreID = ds.DimStoreID
```

```
JOIN Dim_Date dd ON fsa.DimSaleDateID = dd.date_pkey
```



```

WHERE ds.StoreNumber IN (10, 21)

GROUP BY dd.DAY_NAME

ORDER BY

CASE

    WHEN dd.DAY_NAME = 'Monday' THEN 1

    WHEN dd.DAY_NAME = 'Tuesday' THEN 2

    WHEN dd.DAY_NAME = 'Wednesday' THEN 3

    WHEN dd.DAY_NAME = 'Thursday' THEN 4

    WHEN dd.DAY_NAME = 'Friday' THEN 5

    WHEN dd.DAY_NAME = 'Saturday' THEN 6

    WHEN dd.DAY_NAME = 'Sunday' THEN 7

END;

select * from vw_storesalesbydaypivot;

```

--4. Should any new stores be opened? Include all stores in your analysis if necessary. If so, where? Why or why not?

```

CREATE OR REPLACE SECURE VIEW vw_STORE_PERFORMANCE_ANALYSIS AS

SELECT

    s.StoreNumber,

    s.StoreManager,

    l.City,

    l.State_Province AS StateProvince,

    d.YEAR,

    -- Core Sales Metrics

    SUM(f.SaleAmount) AS ActualSales,

```

```

SUM(f.SaleQuantity) AS ActualQuantity,
COUNT(*) AS TransactionCount,
AVG(f.SaleAmount) AS AvgTransactionValue,

-- Target Sales and Achievement %
COALESCE(SUM(fst.SalesTargetAmount), 0) AS SalesTarget,
CASE
    WHEN SUM(fst.SalesTargetAmount) = 0 THEN NULL
    ELSE ROUND(SUM(f.SaleAmount) / SUM(fst.SalesTargetAmount) * 100, 2)
END AS AchievementPercent,

-- Monthly Sales Trend
SUM(f.SaleAmount) / COUNT(DISTINCT d.MONTH_NUM_IN_YEAR) AS AvgMonthlySales,

-- Store Ranking (by Sales Only)
RANK() OVER (PARTITION BY d.YEAR ORDER BY SUM(f.SaleAmount) DESC) AS
StoreRankBySales

FROM FACT_SALESACTUAL f
JOIN DIM_STORE s ON f.DIMSTOREID = s.DIMSTOREID
JOIN DIM_LOCATION l ON s.DIMLOCATIONID = l.DIMLOCATIONID
JOIN DIM_DATE d ON f.DIMSALEDATEID = d.DATE_PKEY
LEFT JOIN Fact_SRCSalesTarget fst ON fst.DimStoreID = s.DimStoreID

WHERE s.StoreID != -1 -- Exclude Unknown stores

```

GROUP BY

s.StoreNumber,

s.StoreManager,

l.City,

l.State_Province,

d.YEAR;

SELECT * FROM vw_STORE_PERFORMANCE_ANALYSIS;