DWH MODELING

Case Study

mohamed alghaly

Sales Analysis for a Retail Company

Sales Analysis

Sales Business Process:

- Our objective is to analyze the company sales and find out ways to improve the business by increasing sales and improving net profit.

STEPS

1- Defining Business Processes
2- Defining Business KPIs (Data Warehouse Objectives)
3- Defining Granularity for the Analysis Scope
4- Defining Dimensions, Facts, and measurements
5- Data Warehouse Modeling (Defining the Schema)
6- Defining the Physical Model
7- Data Integration
8- Data Warehouse Indexing and Partitioning
9- Extracting insights to support the Management Decisions

STEP1

Defining Business Process

We will only focus on the Sales Business Process to support the Finance team.

SALES

The sales business process involves the end-to-end journey of converting customer interest into revenue. In our data warehouse modeling, we'll capture and optimize this process by modeling key entities such as customers, products, orders, and inventory. Through detailed schema design and relationships, we aim to represent customer interactions, order creation, and fulfillment seamlessly. Our focus is on ensuring accurate inventory management, precise pricing calculations, and comprehensive reporting for insights into sales performance. By modeling the sales business process, we strive to enhance efficiency, provide valuable analytics, and support informed decision-making within the retail company.

STEP2

Defining Business KPIs

In this step, we want to extract the KPIs that will help the finance team to improve the business.

KPI	Tables
Top customers (Customer segmentation)	SalesOrdrHeader/Customers
Top Sales reason	SalesOrderHeader/SalesReason
Effect of special offers on sale	SalesOrderHeader/SpecialOffer
Best-selling product category	SalesOrderHeader/Product
Best-selling store/city/state	SalesOrderHeader/Store
What payment method is more profitable	SalesOrderHeader/SpecialOffer
The effect of customer demographics on sales	SalesOrdrHeader/Customers
Effect of special offers on revenue	SalesOrderHeader/SpecialOffer/Purchase
Salesperson performance	SalesOrdrHeader/SalesPerson
The most effective shipment method	SalesOrdrHeader/ShpmentMethod
The effect of changing the price on sales	SalesOrdrHeader/Purchase
Most profitable product/category	SalesOrderHeader/Product

STEP3

Defining Granularity

In this step, we want to define the level of detail that we will use to calculate KPIs for each business process.

SALES Fact Table

- The most detailed grain is the combination of individual Customer, Product, Order Date, Shipping Date, Salesperson, Shipping Address, Billing Address, Promotion Code, Offer applied, Sales Reason, Order Due Date, Store, Currency, Payment Method, Shipment Method, Sales Territory, Shipment, and Order Status.
- In a single word we will accumulate from each item a customer orders.
- A Customer Orders a 25-item order of 6 different products at a specific moment will be represented in 6 rows in the sales Fact Table.

The next step is typically estimating the fact table size as follows:

- We have c Customers, p Products, s Shipment methods, m Payment Method, …
- So, we will have at most n * m * p * s * m * …. Rows, by estimating each record's size we can estimate the maximum data warehouse size.
- We still need a deep overview of the business to estimate an accurate value for the Sparsity (The ratio of non-existing rows) to calculate the estimated DWH size to complete the design.
- By estimating the size of each fact table, we could estimate the size of the DWH besides designing the optimal table partitioning model.

STEP4

Defining Dimensions, Facts, and Measurements

In this step, we want to define each fact table, which dimensions will be connected to it, and the measurements that will help the analysis.

1- Sales Fact
a. What?
   i. This is a fact table to keep track of each sales transaction, we will use it to help the finance team improve their operations.
b. Measurements:
   i. Number of Items(additive).
   ii. Item's Price(additive).
   iii. Sales Amount (additive)
   iv. Item's Cost (semi-additive as it is pointless to sum it up over some dimensions)
   v. Discount (non-additive as it is pointless, to sum it up over any dimension)
   vi. Profit (additive)
   vii. Total Profit (additive)
   viii. Tax Amount (non-additive)
   ix. Total Tax (semi-additive)
   x. Fright (semi-additive as it is pointless to sum it up over some dimensions like the Employee -salesperson-)
   xi. Total Discount (semi-additive)

Dimensions:

1- CUSTOMER

- Conformed Dimension
- A typical dimension holding data about each Customer.

2- PRODUCT

- Hierarchical Dimension
- A typical dimension holding data about each Product.

3- DATE

- Role-Playing Dimension
- The typical data warehouse DATE DIMENSION, which ensures that the DWH is time-variant.

4- EMPLOYEE

- Conformed Dimension
- A typical dimension holding data about each Employee.

5- LOCATION

- Role-Playing Dimension
- This dimension will be used to represent location details and will be used to represent both billing and shipping locations.

6- ORDER DETAIL

- Junk Dimension
- A Junk Dimension holding data about three pieces of information:
    a. Shipment Method, Payment Method, and Currency.

7- PROMOTION

- Conformed Dimension
- A typical Dimension holding data about the promotions offered by the marketing team.

8- STORE

- Conformed Dimension
- A typical dimension holding data about each Store.

9- SPECIAL OFFERS

- Conformed Dimension
- A typical Dimension holding data about the Special Offers offered by the marketing team.

10- SALES REASONS

- Conformed Dimension
- A typical Dimension holding data about the Sales Reason which involves categorical data (10 possible sales reasons)

11- ORDER STATUS

- Degenerated Dimension
- Fact table Attribute that holds the status of the order which can be canceled (0) or completed (1).

12- SHIPMENT NUMBER

- Degenerated Dimension
- Fact table Attribute that holds the shipment number of the order to keep track of the order delivery status.

13- Sales Territory

- Conformed Dimension
- A typical dimension holding geographical data that we will use to analyze our sales over different locations/areas/countries/etc.

STEP5

DWH Modeling

In this step, we will define the DWH Schema, and we will use a Star Schema.

NOTE

Just for simplicity we won't include the Dimensions attributes on the Diagram, and we will create the tables using the ER Diagram of the relational Database.

DimCustomer	FactSales fact table	DimeSalesReason
CustomerID	**SalesKey**	**ReasonKey**

DimStore		DimSpecialOffer
StoreID	CustomerID (FK)	**OfferID**
	StoreKey (FK)	

DimProduct	ProductID (FK)	DimPromotion
ProductID	EmployeeID (FK)	**PromotionKey**
	ReasonKey (FK)	
	OfferID (FK)	

DimSalesPerson	PromotionKey (FK)	DimSalesTerritory
EmployeeID	SalesTerrKey (FK)	**SalesTerrKey**
	OrderDetailID (FK)	

FactSales fact table:
- **SalesKey**
- CustomerID (FK)
- StoreKey (FK)
- ProductID (FK)
- EmployeeID (FK)
- ReasonKey (FK)
- OfferID (FK)
- PromotionKey (FK)
- SalesTerrKey (FK)
- OrderDetailID (FK)
- OrderDateKey (FK)
- DueDateKey (FK)
- ShipDateKey (FK)
- ShipToAddressKey (FK)
- ShipmentNumber
- BillToAddessKey (FK)
- DimOrderStatus
- Items
- Price
- Discount
- Fright
- TaxAmount
- Cost
- Profit
- SalesAmount
- DiscountAmount
- TotalTax
- TotalProfit

DimDate	DimShip_Pay_CurMethod
DateID	**OrderDetailID**

DimLocation
LocationID

STEP6,7

DWH Physical Model

In this step, we will create the Database tables on SQL Server.

Data Integration

In this step, we will populate the data into the tables.

BOTH Table Creation and Data Population Scripts are in a file named **DWH.SQL**.

■ Population Code for The Fact Table:

```sql
INSERT INTO  FactSales(SalesKey, CustomerID, BillToAddessKey, EmployeeID, SalesTerrKey,
ShipToAddressKey,
DimOrderStatus, ShipmentNumber, TotalTax, fright, SalesAmount, Items, Price, Discount,
DiscountAmount,
OfferID, ProductID, ReasonKey, Cost, Profit, TotalProfit)
SELECT
DWH.SalesOrderID as SalesKey,
DWH.CustomerID as CustomerID, DWH.BillToAddressID as BillToAddressKey,
DWH.SalesPersonID as EmployeeID, DWH.TerritoryID as SalesTerrKey,
DWH.ShipToAddressID as ShipToAddressKey, DWH.Status as DimOrderStatus,
DWH.SalesOrderNumber as ShipmentNumber,
DWH.TaxAmt as TotalTax, DWH.Freight as fright,
DWH.TotalDue as SalesAmount,
OD.OrderQty as Items, OD.UnitPrice as Price, OD.UnitPriceDiscount as Discount,
OD.UnitPriceDiscount * OD.UnitPrice * OD.OrderQty as DiscountAmount,
od.SpecialOfferID as OfferID, od.ProductID as ProductID,
SR.SalesReasonID as ReasonKey,
SP.StandardCost as Cost,
OD.UnitPrice - SP.StandardCost as Profit,
(OD.UnitPrice - SP.StandardCost) * OD.OrderQty - (OD.UnitPriceDiscount * OD.UnitPrice *
OD.OrderQty)  as TotalProfit
from [AdventureWorks2019].[Sales].[SalesOrderHeader] DWH
inner join [AdventureWorks2019].[Sales].SalesOrderDetail OD on od.SalesOrderID = dwh.SalesOrderID
left join
[AdventureWorks2019].[Sales].[SalesOrderHeaderSalesReason] SR on DWH.SalesOrderID =
SR.SalesOrderID
        left join [AdventureWorks2019].[Production].[Product] SP on SP.ProductID = OD.ProductID ;
```
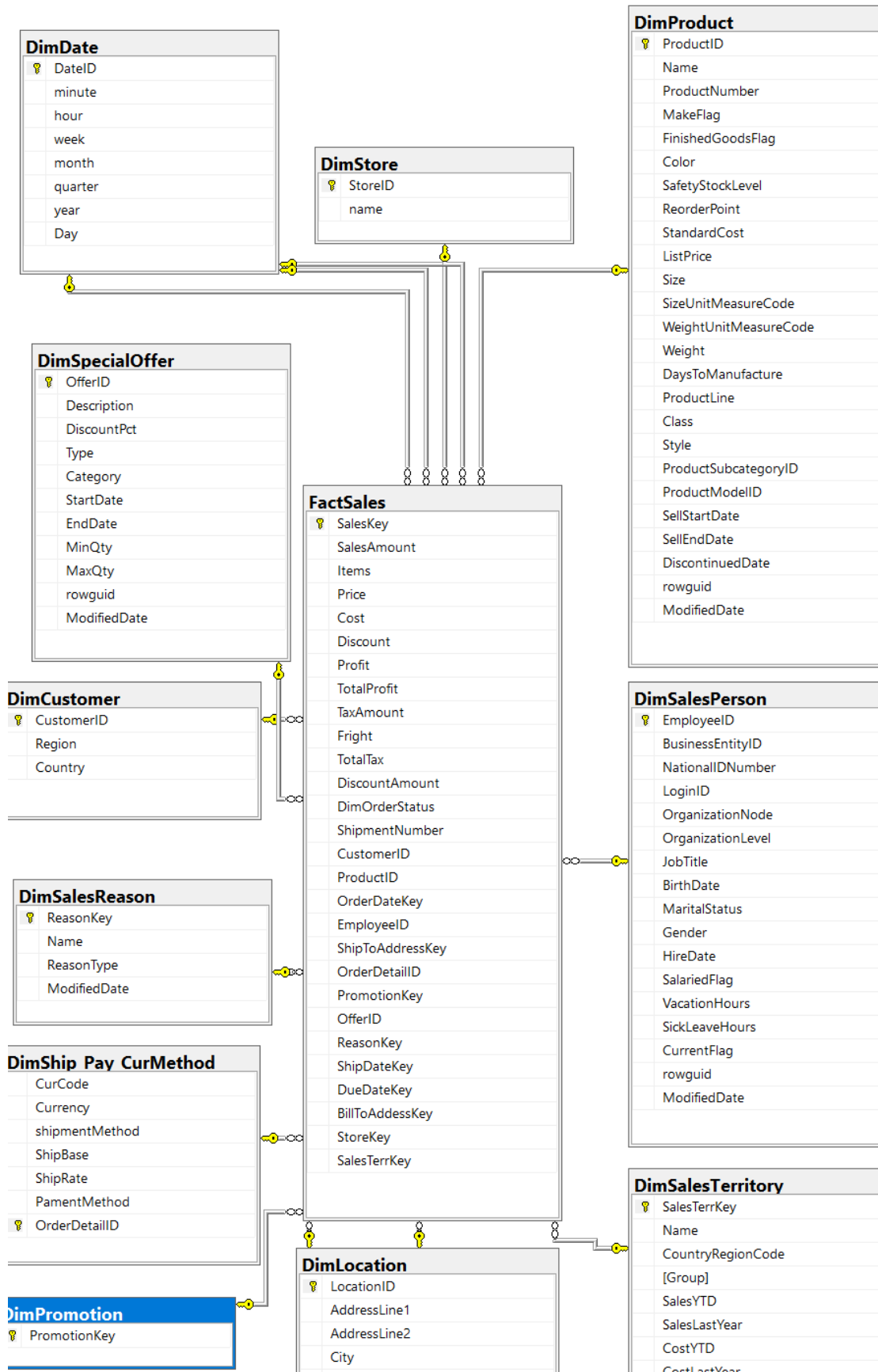
STEP8

DWH Performance

In this step, we will create the Database Indexes and partition the fact table to ensure optimum query performance.

We will Skip This Step for now!

ACTUAL DATABASE SCHEMA

DimDate
- DateID
- minute
- hour
- week
- month
- quarter
- year
- Day

DimStore
- StoreID
- name

DimProduct
- ProductID
- Name
- ProductNumber
- MakeFlag
- FinishedGoodsFlag
- Color
- SafetyStockLevel
- ReorderPoint
- StandardCost
- ListPrice
- Size
- SizeUnitMeasureCode
- WeightUnitMeasureCode
- Weight
- DaysToManufacture
- ProductLine
- Class
- Style
- ProductSubcategoryID
- ProductModelID
- SellStartDate
- SellEndDate
- DiscontinuedDate
- rowguid
- ModifiedDate

DimSpecialOffer
- OfferID
- Description
- DiscountPct
- Type
- Category
- StartDate
- EndDate
- MinQty
- MaxQty
- rowguid
- ModifiedDate

FactSales
- SalesKey
- SalesAmount
- Items
- Price
- Cost
- Discount
- Profit
- TotalProfit
- TaxAmount
- Fright
- TotalTax
- DiscountAmount
- DimOrderStatus
- ShipmentNumber
- CustomerID
- ProductID
- OrderDateKey
- EmployeeID
- ShipToAddressKey
- OrderDetailID
- PromotionKey
- OfferID
- ReasonKey
- ShipDateKey
- DueDateKey
- BillToAddessKey
- StoreKey
- SalesTerrKey

DimCustomer
- CustomerID
- Region
- Country

DimSalesReason
- ReasonKey
- Name
- ReasonType
- ModifiedDate

DimShip_Pay_CurMethod
- CurCode
- Currency
- shipmentMethod
- ShipBase
- ShipRate
- PamentMethod
- OrderDetailID

DimPromotion
- PromotionKey

DimLocation
- LocationID
- AddressLine1
- AddressLine2
- City

DimSalesPerson
- EmployeeID
- BusinessEntityID
- NationalIDNumber
- LoginID
- OrganizationNode
- OrganizationLevel
- JobTitle
- BirthDate
- MaritalStatus
- Gender
- HireDate
- SalariedFlag
- VacationHours
- SickLeaveHours
- CurrentFlag
- rowguid
- ModifiedDate

DimSalesTerritory
- SalesTerrKey
- Name
- CountryRegionCode
- [Group]
- SalesYTD
- SalesLastYear
- CostYTD
- CostLastYear

STEP9

Gaining Insights

In this step we will write sample SQL Queries to extract some insights from DWH.

We could extract hundreds of insights and I will only list a sample of those here.

1- Top customers (Customer segmentation)

```sql
WITH
    sales (customerid, sales)
    AS
        (  SELECT customerid, SUM (SalesAmount)
             FROM FactSales
          GROUP BY customerid)
SELECT customerid, sales, NTILE (3) OVER (ORDER BY sales DESC)
  FROM sales;
```

2- Effect of special offers on sale

```sql
SELECT Description, SUM (SalesAmount) sales
    FROM DimSpecialOffer o INNER JOIN
factsales s ON s.OfferID = o.OfferID
GROUP BY Description
  ORDER BY sales DESC;
```

	Description	sales
1	No Discount	2743497559.28059
2	Volume Discount 11 to 14	92785082.9788011
3	Volume Discount 15 to 24	40820134.1373
4	Touring-3000 Promotion	25724698.3948
5	Touring-1000 Promotion	11075941.886
6	Mountain-500 Silver Clearance Sale	7497361.0031
7	Volume Discount 25 to 40	5540046.7703
8	Sport Helmet Discount-2002	5142021.4538
9	Sport Helmet Discount-2003	4405695.0115
10	Road-650 Overstock	3424403.0756
11	Mountain-100 Clearance Sale	2959401.8322
12	Volume Discount 41 to 60	167479.5391

3- Best-selling product category

```sql
SELECT top 10 name, SUM (salesamount) sales
    FROM DimProduct P INNER JOIN
FactSales s ON s.ProductID = p.ProductID
GROUP BY name
ORDER BY sales DESC;
```

▦ Results ▤▯ Messages

	name	sales
1	AWC Logo Cap	50452164.9513002
2	Long-Sleeve Logo Jersey, L	47770103.7515999
3	Sport-100 Helmet, Blue	46452611.3843
4	Sport-100 Helmet, Black	45162760.3351999
5	Sport-100 Helmet, Red	43483450.0667
6	Long-Sleeve Logo Jersey, M	39403059.4420999
7	Long-Sleeve Logo Jersey, XL	34755079.782
8	Half-Finger Gloves, M	30529153.4439001
9	LL Road Frame - Black, 52	27242670.5007
10	Road-250 Black, 44	26091358.0471999

4- Best-selling store/city/state

```sql
SELECT CountryRegionCode, SUM (salesamount)
sales
    FROM DimSalesTerritory T
        INNER JOIN factsales s ON
s.SalesTerrKey = t.SalesTerrKey
GROUP BY CountryRegionCode
ORDER BY sales DESC;
```

	CountryRegionCode	sales
1	US	1857141201.25516
2	CA	528180676.991385
3	FR	199428397.546105
4	GB	189090593.470699
5	DE	94308357.8587999
6	AU	74890598.2407936

```sql
SELECT NAME AS REGION, SUM (salesamount)
sales
    FROM DimSalesTerritory T
        INNER JOIN factsales s ON
s.SalesTerrKey = t.SalesTerrKey
GROUP BY NAME
ORDER BY sales DESC;
```

	REGION	sales
1	Southwest	700478635.960674
2	Canada	528180676.991385
3	Northwest	413353522.623092
4	Central	263102103.074001
5	Northeast	253771294.444598
6	Southeast	226435645.152799
7	France	199428397.546105
8	United Kingdom	189090593.470699
9	Germany	94308357.8587999
10	Australia	74890598.2407936

5- Effect of special offers (Discount) on revenue

```sql
SELECT DiscountPct, ROUND (SUM (TotalProfit),
0) AS profit
    FROM DimSpecialOffer O INNER JOIN FactSales S
ON s.OfferID = o.OfferID
GROUP BY DiscountPct;
```

Results Messages

	DiscountPct	profit
1	0.00	13459338
2	0.02	1011967
3	0.05	-115521
4	0.10	-20432
5	0.15	-268187
6	0.20	-524450
7	0.30	-97973
8	0.35	-617514
9	0.40	-91840

6- Employees Performance

```sql
SELECT JobTitle, ROUND (SUM (SalesAmount), 0)
sales
    FROM DimSalesPerson E INNER JOIN FactSales S ON S.EmployeeID =
E.EmployeeID
GROUP BY JobTitle
ORDER BY sales DESC;
```

Results Messages

	JobTitle	sales
1	Sales Representative	2780918430
2	North American Sales Manager	45434066
3	European Sales Manager	27128772
4	Pacific Sales Manager	8042820

Those were just a sample of hundreds other insights/KPIs that we can extract from the DWH.