**SQL Scripts**

/\*Database to store the raw data tables\*/

create database ThinkES

/\*Reading superstore\_south\_sales JSON file with the below script \*/

declare @json\_table nvarchar(max)

select @json\_table = BulkColumn

from openrowset(bulk 'E:\Visual Studio 2019\Downloads\Data Analyst Project\group\_sales\_data\superstore\_south\_sales.json', single\_clob) as NewTable

insert into superstore\_south\_sales select [SampleDataSouth].[Order], [SampleDataSouth].[Date], [SampleDataSouth].[Date Shipped],

[SampleDataSouth].[Shipping Method], [SampleDataSouth].[Customer Number], [SampleDataSouth].[Segment], [SampleDataSouth].[Postal Code 1],

[SampleDataSouth].[Product ID], [SampleDataSouth].[Sales], [SampleDataSouth].[Quantity], [SampleDataSouth].[Discount], [SampleDataSouth].[Profit]

from openjson(@json\_table)

with(

[sample\_data\_south] nvarchar(max) as json

) as [sample\_data]

cross apply openjson([sample\_data].[sample\_data\_south])

with(

[Order] nvarchar(50),

[Date] nvarchar(50),

[Date Shipped] nvarchar(50),

[Shipping Method] nvarchar(50),

[Customer Number] nvarchar(100),

[Segment] nvarchar(50),

[Postal Code 1] int,

[Product ID] nvarchar(100),

[Sales] nvarchar(50),

[Quantity] int,

[Discount] nvarchar(50),

[Profit] nvarchar(100)

)as [SampleDataSouth]

/\*drop table superstore\_south\_sales;\*/

/\*Retrieving all the records of the four table imported using the flat file importer after data cleaning and modification with Python\*/

select \* from [superstore\_south\_sales];

select \* from superstore\_east\_sales;

select \* from superstore\_central\_sales;

select \* from superstore\_west\_sales;

/\*Removing null values from superstore\_west\_sales table \*/

delete from [superstore\_west\_sales] where [postal\_code] is null;

select \* from Companies;

select \* from Postal\_codes;

select \* from Product\_types;

select \* from Products;

/\*Dropping the additional column 'column1' of the Python dataframe\*/

alter table [superstore\_south\_sales] drop column column1

alter table [superstore\_west\_sales] drop column column1

alter table [superstore\_east\_sales] drop column column1

alter table [superstore\_central\_sales] drop column column1

/\*ETL data insertion query for Products\_Dimension table in other data warehouse database named as ThinkESDW\*/

/\*#######################################################################################################################################################################\*/

insert into ThinkESDW.dbo.Products\_Dimension(Order\_id, Product\_id, Product\_name, Product\_type, Product\_category) select distinct a.[Order], a.[Product\_ID], b.product\_name, c.product\_type, c.product\_category from [superstore\_south\_sales] a, [Products] b, [Product\_types] c

where a.[Product\_ID] = b.product\_id and b.category = c.product\_type order by [Order];

insert into ThinkESDW.dbo.Products\_Dimension(Order\_id, Product\_id, Product\_name, Product\_type, Product\_category) select distinct a.[order\_number], a.[product\_id], b.product\_name, c.product\_type, c.product\_category from [superstore\_east\_sales] a, [Products] b, [Product\_types] c

where a.[product\_id] = b.product\_id and b.category = c.product\_type order by [order\_number];

insert into ThinkESDW.dbo.Products\_Dimension(Order\_id, Product\_id, Product\_name, Product\_type, Product\_category) select distinct a.[Order\_ID], a.[Product\_ID], b.product\_name, c.product\_type, c.product\_category from [superstore\_central\_sales] a, [Products] b, [Product\_types] c

where a.[Product\_ID] = b.product\_id and b.category = c.product\_type order by [Order\_ID];

insert into ThinkESDW.dbo.Products\_Dimension(Order\_id, Product\_id, Product\_name, Product\_type, Product\_category) select distinct a.[order\_id], a.[product\_id], b.product\_name, c.product\_type, c.product\_category from [superstore\_west\_sales] a, [Products] b, [Product\_types] c

where a.[product\_id] = b.product\_id and b.category = c.product\_type order by [order\_id];

/\*#######################################################################################################################################################################\*/

/\*ETL data insertion query for Customer\_Dimension table in other data warehouse database named as ThinkESDW\*/

/\*#######################################################################################################################################################################\*/

insert into ThinkESDW.dbo.Customer\_Dimension(Customer\_id, Customer\_section, Order\_id) select distinct Customer\_Number, Segment, [Order] from superstore\_south\_sales order by [Order]

insert into ThinkESDW.dbo.Customer\_Dimension(Customer\_id, Customer\_section, Order\_id) select distinct customer\_key, customer\_section, order\_number from superstore\_east\_sales order by [order\_number]

insert into ThinkESDW.dbo.Customer\_Dimension(Customer\_id, Customer\_section, Order\_id) select distinct Customer\_ID, Segment, [Order\_ID] from superstore\_central\_sales order by [Order\_ID]

insert into ThinkESDW.dbo.Customer\_Dimension(Customer\_id, Customer\_section, Order\_id) select distinct customer\_id, segment, order\_id from superstore\_west\_sales order by [order\_id]

/\*#######################################################################################################################################################################\*/

/\*ETL data insertion query for Order\_Dimension table in other data warehouse database named as ThinkESDW\*/

/\*#######################################################################################################################################################################\*/

insert into ThinkESDW.dbo.Order\_Dimension(Order\_id, Order\_date, Order\_sales, Order\_quantity, Order\_discount, Order\_profit)

select distinct [Order], [Date], Sales, Quantity, Discount, Profit from [superstore\_south\_sales] order by [Order]

insert into ThinkESDW.dbo.Order\_Dimension(Order\_id, Order\_date, Order\_sales, Order\_quantity, Order\_discount, Order\_profit)

select distinct [order\_number], [ord\_date], sales, items, reduction, total from [superstore\_east\_sales] order by [order\_number]

insert into ThinkESDW.dbo.Order\_Dimension(Order\_id, Order\_date, Order\_sales, Order\_quantity, Order\_discount, Order\_profit)

select distinct [Order\_ID], [Order\_Date], Sales, Quantity, Discount, Profit from [superstore\_central\_sales] order by [Order\_ID]

insert into ThinkESDW.dbo.Order\_Dimension(Order\_id, Order\_date, Order\_sales, Order\_quantity, Order\_discount, Order\_profit)

select distinct [order\_id], [order\_date], sales, quantity, discount, profit from [superstore\_west\_sales] order by [order\_id]

/\*#######################################################################################################################################################################\*/

/\*ETL data insertion query for Shipping\_Dimension table in other data warehouse database named as ThinkESDW\*/

/\*#######################################################################################################################################################################\*/

insert into ThinkESDW.dbo.Shipping\_Dimension(Shipping\_date, Order\_id, Shipping\_type, Postal\_code, Region, [State])

select distinct a.[Date\_Shipped], a.[Order], a.[Shipping\_Method], a.[Postal\_Code\_1], b.region, b.[state] from [superstore\_south\_sales] a, [Postal\_codes] b

where a.[Postal\_Code\_1] = b.postal\_code order by [Order];

insert into ThinkESDW.dbo.Shipping\_Dimension(Shipping\_date, Order\_id, Shipping\_type, Postal\_code, Region, [State])

select distinct a.[ship\_date], a.[order\_number], a.[ship\_type], a.[post\_code], b.region, b.[state] from [superstore\_east\_sales] a, [Postal\_codes] b

where a.[post\_code] = b.postal\_code order by [order\_number];

insert into ThinkESDW.dbo.Shipping\_Dimension(Shipping\_date, Order\_id, Shipping\_type, Postal\_code, Region, [State])

select distinct a.[Ship\_Date], a.[Order\_ID], a.[Ship\_Mode], a.[Postal\_Code], b.region, b.[state] from [superstore\_central\_sales] a, [Postal\_codes] b

where a.[Postal\_Code] = b.postal\_code order by [order\_id];

insert into ThinkESDW.dbo.Shipping\_Dimension(Shipping\_date, Order\_id, Shipping\_type, Postal\_code, Region, [State])

select distinct a.[ship\_date], a.[order\_id], a.[ship\_mode], a.[postal\_code], b.region, b.[state] from [superstore\_west\_sales] a, [Postal\_codes] b

where a.[postal\_code] = b.postal\_code order by [Order\_ID];

/\*#######################################################################################################################################################################\*/

/\*Database to store the cleaned and processed data as a datawarehouse\*/

create database ThinkESDW

use ThinkESDW;

/\*Dimension table creation based on the shipping data\*/

create table Shipping\_Dimension(

Shipping\_Key int not null identity(1,1) primary key,

Order\_id nvarchar(50),

Shipping\_date date,

Shipping\_type nvarchar(50),

Postal\_code int,

Region nvarchar(50),

[State] nvarchar(50)

);

/\*drop table Shipping\_Dimension;\*/

/\*Dimension table creation based on the products data\*/

create table Products\_Dimension(

Product\_Key int not null identity(1,1) primary key,

Order\_id nvarchar(50),

Product\_id nvarchar(50),

Product\_name nvarchar(100),

Product\_type nvarchar(50),

Product\_category nvarchar(50)

)

/\*drop table Products\_Dimension\*/

/\*Dimension table creation based on the customer data\*/

create table Customer\_Dimension(

Customer\_Key int not null identity(1,1) primary key,

Customer\_id nvarchar(50),

Order\_id nvarchar(50),

Customer\_section nvarchar(50)

);

/\*drop table Customer\_Dimension\*/

/\*Dimension table creation based on the Order data\*/

create table Order\_Dimension(

Order\_Key int not null identity(1,1) primary key,

Order\_id nvarchar(50),

Order\_date date,

Order\_sales float,

Order\_quantity int,

Order\_discount float,

Order\_profit float

);

/\*drop table Order\_Dimension\*/

/\*Centralised Fact table combining all the dimensions giving new facts/insights. Linked with dimension tables with primary-foreign key relationship and represents a star schema\*/

create table Sales\_Fact(

Sales\_Key int not null identity(1,1) primary key,

Order\_Key int FOREIGN KEY REFERENCES Order\_Dimension(Order\_Key),

Customer\_Key int FOREIGN KEY REFERENCES Customer\_Dimension(Customer\_Key),

Product\_Key int FOREIGN KEY REFERENCES Products\_Dimension(Product\_Key),

Shipping\_Key int FOREIGN KEY REFERENCES Shipping\_Dimension(Shipping\_Key),

Product\_price float,

Product\_discount\_price float,

Total\_discount\_price float,

Product\_profit float,

Total\_sales float

);

select \* from Sales\_Fact;

select \* from Order\_Dimension;

select \* from Shipping\_Dimension;

select \* from Products\_Dimension;

select \* from Customer\_Dimension;

/\*drop table Sales\_Fact;\*/

/\*ETL query to load the final fact table\*/

insert into Sales\_Fact(Order\_Key, Shipping\_Key, Product\_Key, Customer\_Key, Product\_price, Product\_discount\_price, Total\_discount\_price, Product\_profit, Total\_sales)

select a.Order\_Key, b.Shipping\_Key, c.Product\_Key, d.Customer\_Key, round((a.Order\_sales/a.Order\_quantity), 2) as [Product\_price],

round(((a.Order\_sales/a.Order\_quantity) \* a.Order\_discount), 2) as [Product\_discount\_price], round((a.Order\_sales \* a.Order\_discount), 2) as [Total\_discount\_price],

round((a.Order\_profit/a.Order\_quantity), 2) as [Product\_profit], round((a.Order\_sales + a.Order\_profit), 2) as [Total\_sales]

from Order\_Dimension a, Customer\_Dimension d, Products\_Dimension c, Shipping\_Dimension b

where a.[Order\_id] = b.[Order\_id] and a.[Order\_id] = c.[Order\_id] and a.[Order\_id] = d.[Order\_id];

**Unit Test Scripts**

/\*Stored procedure used for Unit Testing on Sales\_Fact table\*/

create proc facttest

(

@Order\_Key int = NULL ,

@Customer\_Key int = NULL,

@Product\_Key int = NULL ,

@Shipping\_Key int = NULL,

@Product\_price float = NULL,

@Product\_discount\_price float = NULL,

@Total\_discount\_price float = NULL,

@Product\_profit float = NULL,

@Total\_sales float = NULL

)

as

INSERT INTO Sales\_Fact

VALUES

(

@Order\_Key,

@Customer\_Key,

@Product\_Key,

@Shipping\_Key,

@Product\_price,

@Product\_discount\_price,

@Total\_discount\_price,

@Product\_profit,

@Total\_sales

)

go

select \* from Sales\_Fact;

-- database unit test for dbo.facttest

DECLARE @RC AS INT, @Order\_Key AS INT, @Customer\_Key AS INT, @Product\_Key AS INT, @Shipping\_Key AS INT, @Product\_price AS FLOAT (53), @Product\_discount\_price AS FLOAT (53), @Total\_discount\_price AS FLOAT (53), @Product\_profit AS FLOAT (53), @Total\_sales AS FLOAT (53);

SELECT @RC = 0,

@Order\_Key = 1,

@Customer\_Key = 1,

@Product\_Key = 1,

@Shipping\_Key = 1,

@Product\_price = 233.32,

@Product\_discount\_price = 104.99,

@Total\_discount\_price = 419.97,

@Product\_profit = -114.54,

@Total\_sales = 475.12;

EXECUTE @RC = [dbo].[facttest] @Order\_Key, @Customer\_Key, @Product\_Key, @Shipping\_Key, @Product\_price, @Product\_discount\_price, @Total\_discount\_price, @Product\_profit, @Total\_sales;

SELECT @RC AS RC;

--pre-test—script

delete from Sales\_Fact where Product\_Key = 1 and Product\_price = 233.32 and

Product\_discount\_price = 104.99 and

Total\_discount\_price = 419.97 and

Product\_profit = -114.54 and

Total\_sales = 475.12;

--post-test-script

select \* from Sales\_Fact where Product\_Key = 1 and Product\_price = 233.32 and

Product\_discount\_price = 104.99 and

Total\_discount\_price = 419.97 and

Product\_profit = -114.54 and

Total\_sales = 475.12;

**Python Scripts**

import pandas as pd

dfc = pd.read\_csv('E:/Visual Studio 2019/Downloads/Data Analyst Project/group\_sales\_data/sales\_data\_csv/superstore\_central\_sales.csv')

dfc.head()

dfc['Order Date'] = pd.to\_datetime(dfc['Order Date'], errors='coerce')

dfc['Ship Date'] = pd.to\_datetime(dfc['Ship Date'], errors='coerce')

dfc['Sales'] = pd.to\_numeric(dfc['Sales'], errors='coerce')

dfc['Quantity'] = pd.to\_numeric(dfc['Quantity'], errors='coerce')

dfc['Discount'] = pd.to\_numeric(dfc['Discount'], errors='coerce')

dfc['Profit'] = pd.to\_numeric(dfc['Profit'], errors='coerce')

dfc['Postal Code'] = pd.to\_numeric(dfc['Postal Code'], errors='coerce')

dfc.dtypes

dfc=dfc.dropna()

dfc.to\_csv('E:/Visual Studio 2019/Downloads/Data Analyst Project/group\_sales\_data/superstore\_central\_sales1.csv')

dfw = pd.read\_csv('E:/Visual Studio 2019/Downloads/Data Analyst Project/group\_sales\_data/sales\_data\_csv/superstore\_west\_sales.csv')

dfw.head()

dfw['order\_date'] = pd.to\_datetime(dfw['order\_date'], errors='coerce')

dfw['ship\_date'] = pd.to\_datetime(dfw['ship\_date'], errors='coerce')

dfw['sales'] = pd.to\_numeric(dfw['sales'], errors='coerce')

dfw['quantity'] = pd.to\_numeric(dfw['quantity'], errors='coerce')

dfw['discount'] = pd.to\_numeric(dfw['discount'], errors='coerce')

dfw['profit'] = pd.to\_numeric(dfw['profit'], errors='coerce')

dfw['postal\_code'] = pd.to\_numeric(dfw['postal\_code'], errors='coerce')

dfw.dtypes

dfw=dfw.dropna()

dfw.to\_csv('E:/Visual Studio 2019/Downloads/Data Analyst Project/group\_sales\_data/superstore\_west\_sales1.csv')

dfs = pd.read\_csv('E:/Visual Studio 2019/Downloads/Data Analyst Project/group\_sales\_data/superstore\_south\_sales.csv')

dfs.head()

dfs['Date'] = pd.to\_datetime(dfs['Date'], errors='coerce')

dfs['Date Shipped'] = pd.to\_datetime(dfs['Date Shipped'], errors='coerce')

dfs['Sales'] = pd.to\_numeric(dfs['Sales'], errors='coerce')

dfs['Quantity'] = pd.to\_numeric(dfs['Quantity'], errors='coerce')

dfs['Discount'] = pd.to\_numeric(dfs['Discount'], errors='coerce')

dfs['Profit'] = pd.to\_numeric(dfs['Profit'], errors='coerce')

dfs['Postal Code 1'] = pd.to\_numeric(dfs['Postal Code 1'], errors='coerce')

dfs.dtypes

dfs=dfs.dropna()

dfs.head()

dfs.to\_csv('E:/Visual Studio 2019/Downloads/Data Analyst Project/group\_sales\_data/superstore\_south\_sales1.csv')

dfe = pd.read\_csv('E:/Visual Studio 2019/Downloads/Data Analyst Project/group\_sales\_data/superstore\_east\_sales.csv')

dfe.head()

dfe['ord\_date'] = pd.to\_datetime(dfe['ord\_date'], errors='coerce')

dfe['ship\_date'] = pd.to\_datetime(dfe['ship\_date'], errors='coerce')

dfe['sales'] = pd.to\_numeric(dfe['sales'], errors='coerce')

dfe['#\_items'] = pd.to\_numeric(dfe['#\_items'], errors='coerce')

dfe['reduction'] = pd.to\_numeric(dfe['reduction'], errors='coerce')

dfe['total'] = pd.to\_numeric(dfe['total'], errors='coerce')

dfe['post\_code'] = pd.to\_numeric(dfe['post\_code'], errors='coerce')

dfe.dtypes

dfe=dfe.dropna()

dfe.to\_csv('E:/Visual Studio 2019/Downloads/Data Analyst Project/group\_sales\_data/superstore\_east\_sales1.csv')