Création de l'entrepôt de données Dw\_Northwind :

1- Créer la base de données :

CREATE DATABASE Dw\_Northwind

2- Création des tables :

USE Dw\_Northwind

/\*--- Création des tables de dimensions ----\*/

----------------------------

CREATE TABLE TD\_Cities(

City nvarchar(15) NOT NULL,

Country nvarchar(15),

PRIMARY KEY(City));

----------------------------

CREATE TABLE TD\_Discount(

Discount real NOT NULL,

PRIMARY KEY(Discount));

----------------------------

CREATE TABLE TD\_OrderMonth(

OrderMonth nvarchar(7) NOT NULL,

OrderYear int,

PRIMARY KEY(OrderMonth));

------------------------------

CREATE TABLE TD\_PriceLevel(

PriceLevel nvarchar(2) NOT NULL,

PRIMARY KEY(PriceLevel));

----------------------------

CREATE TABLE TD\_StockLevel(

StockLevel nvarchar(2) NOT NULL,

PRIMARY KEY(StockLevel));

------------------------------

CREATE TABLE TD\_SalesLevel(

SalesLevel nvarchar(6) NOT NULL,

PRIMARY KEY(SalesLevel));

-----------------------------

CREATE TABLE TD\_OrderLevel(

OrderLevel nvarchar(6) NOT NULL,

PRIMARY KEY(OrderLevel));

-----------------------------

CREATE TABLE TD\_Categories(

CategoryID int NOT NULL,

CategoryName nvarchar(15),

PRIMARY KEY(CategoryID));

----------------------------

CREATE TABLE TD\_Products(

ProductID int NOT NULL,

ProductName nvarchar(40)

CategoryID int NOT NULL,

PRIMARY KEY(ProductID)

CONSTRAINT FK\_ProductsCategories FOREIGN KEY(CategoryID) REFERENCES Categories(CategoryID))

----------------------------

CREATE TABLE TD\_Suppliers(

SupplierID int NOT NULL,

ContactName nvarchar(30),

City nvarchar(15) NOT NULL,

PRIMARY KEY(SupplierID),

CONSTRAINT FK\_City\_Suppliers FOREIGN KEY(City) REFERENCES TD\_Cities(City));

-----------------------------

CREATE TABLE TD\_Employees(

EmployeeID int NOT NULL,

LastName nvarchar(20),

FirstName nvarchar(10),

City nvarchar(15) NOT NULL,

PRIMARY KEY(EmployeeID),

CONSTRAINT FK\_City\_Employees FOREIGN KEY(City) REFERENCES TD\_Cities(City));

-----------------------------

CREATE TABLE TD\_Customers(

CustomerID nvarchar(5) NOT NULL,

ContactName nvarchar(30),

City nvarchar(15) NOT NULL,

PRIMARY KEY(CustomerID),

CONSTRAINT FK\_City\_Customers FOREIGN KEY(City) REFERENCES TD\_Cities(City));

3- Création des tables de faits :

USE Dw\_Northwind

-----------CREATION TABLES DE FAITS ---------------

CREATE TABLE TF\_Orders(

CustomerID nvarchar(5) NOT NULL,

EmployeeID int NOT NULL,

OrderMonth nvarchar(7) NOT NULL,

ShippedOrders int,

NotShippedOrders int,

CONSTRAINT PK\_TFOrders PRIMARY KEY(CustomerID, EmployeeID, OrderMonth),

CONSTRAINT FK\_OrdersCustomers FOREIGN KEY(CustomerID) REFERENCES TD\_Customers(CustomerID)

);

**-------------------------------**

CREATE TABLE TF\_Revenue(

OrderMonth nvarchar(7) NOT NULL,

CustomerID nvarchar(5) NOT NULL,

ProductID int NOT NULL,

CategoryID int NOT NULL,

revenueShipped money,

revenueNotShipped money,

PRIMARY KEY(OrderMonth, CustomerID, CategoryID),

CONSTRAINT FK\_RevenueCustomers FOREIGN KEY(CustomerID) REFERENCES TD\_Customers(CustomerID),

CONSTRAINT FK\_RevenueMonthOrder FOREIGN KEY(OrderMonth) REFERENCES TD\_OrderMonth(OrderMonth),

CONSTRAINT FK\_RevenueTDProducts FOREIGN KEY(ProductID) REFERENCES TD\_Products(ProductID),

CONSTRAINT FK\_RevenueCategories FOREIGN KEY(CategoryID) REFERENCES TD\_Categories(CategoryID));

-------------------------------

CREATE TABLE TF\_Customers(

OrderLevel nvarchar(6) NOT NULL,

SalesLevel nvarchar(6) NOT NULL,

City nvarchar(15) NOT NULL,

NumberCustomers int,

PRIMARY KEY(OrderLevel, SalesLevel, City),

CONSTRAINT FK\_CustomersOrderLevel FOREIGN KEY(OrderLevel) REFERENCES TD\_OrderLevel(OrderLevel),

CONSTRAINT FK\_CustomersSalesLevel FOREIGN KEY(SalesLevel) REFERENCES TD\_SalesLevel(SalesLevel),

CONSTRAINT FK\_CustomersCities FOREIGN KEY(City) REFERENCES TD\_Cities(City));

------------------------------

CREATE TABLE TF\_Products(

SupplierID int NOT NULL,

CategoryID int NOT NULL,

PriceLevel nvarchar(2) NOT NULL,

StockLevel nvarchar(2) NOT NULL,

NumberProducts int,

PRIMARY KEY(SupplierID, CategoryID, PriceLevel, StockLevel),

CONSTRAINT FK\_ProductsSupplies FOREIGN KEY(SupplierID) REFERENCES TD\_Suppliers(SupplierID),

CONSTRAINT FK\_ProductsCategories FOREIGN KEY(CategoryID) REFERENCES TD\_Categories(CategoryID),

CONSTRAINT FK\_ProductsPriceLevel FOREIGN KEY(PriceLevel) REFERENCES TD\_PriceLevel(PriceLevel),

CONSTRAINT FK\_ProductsStockLevel FOREIGN KEY(StockLevel) REFERENCES TD\_StockLevel(StockLevel));

-----------------------------

CREATE TABLE TF\_Discount(

OrderMonth nvarchar(7) NOT NULL,

CustomerID nvarchar(5) NOT NULL,

Discount real NOT NULL,

DiscountValue money,

PRIMARY KEY(OrderMonth, CustomerID, Discount),

CONSTRAINT FK\_DiscountCustomers FOREIGN KEY(CustomerID) REFERENCES TD\_Customers(CustomerID),

CONSTRAINT FK\_DiscountOrderMonth FOREIGN KEY(OrderMonth) REFERENCES TD\_OrderMonth(OrderMonth),

CONSTRAINT FK\_DiscountDiscount FOREIGN KEY(Discount) REFERENCES TD\_Discount(Discount));

Alimentation de l'entrepôt de données (Not Asked)

-----------Alimentation de l'entrepôt de données -------------

USE Northwind

---------------------------------------------------------------------------------------------------

/\*---------------- Cities, Employees, Customers, Suppliers, Categories, OrderMonth --------------------\*/

--------------------------------------------------------------------------------------------------

----------- TD\_Cities (95 row(s) affected) -------------

INSERT INTO Dw\_Northwind.dbo.TD\_Cities (City, Country)

SELECT City, Country FROM Customers WHERE City IS NOT NULL AND NOT EXISTS(Select City From Dw\_Northwind.dbo.TD\_Cities)

UNION

SELECT City, Country FROM Suppliers WHERE City IS NOT NULL AND NOT EXISTS(Select City From Dw\_Northwind.dbo.TD\_Cities)

UNION

SELECT City, Country FROM Employees WHERE City IS NOT NULL AND NOT EXISTS(Select City From Dw\_Northwind.dbo.TD\_Cities)

----------- TD\_Employees (9 row(s) affected)-------------

INSERT INTO Dw\_Northwind.dbo.TD\_Employees (EmployeeID, LastName, FirstName, City)

SELECT Employees.EmployeeID, Employees.LastName, Employees.FirstName, Employees.City

FROM Employees

WHERE NOT EXISTS(Select EmployeeID From Dw\_Northwind.dbo.TD\_Employees)

----------- TD\_Customers (91 row(s) affected)-------------

INSERT INTO Dw\_Northwind.dbo.TD\_Customers (CustomerID, ContactName, City)

SELECT CustomerID, ContactName, City

FROM Customers

WHERE NOT EXISTS(SELECT CustomerID FROM Dw\_Northwind.dbo.TD\_Customers)

----------- TD\_Suppliers (21 row(s) affected)-------------

INSERT INTO Dw\_Northwind.dbo.TD\_Suppliers (SupplierID, ContactName, City)

SELECT SupplierID, ContactName, City

FROM Suppliers

WHERE NOT EXISTS(SELECT SupplierID FROM Dw\_Northwind.dbo.TD\_Suppliers)

----------- TD\_Categories (8 row(s) affected)-------------

INSERT INTO Dw\_Northwind.dbo.TD\_Categories(CategoryID, CategoryName)

SELECT CategoryID, CategoryName

FROM Categories

WHERE NOT EXISTS(SELECT CategoryID FROM Dw\_Northwind.dbo.TD\_Categories)

----------- TD\_Discount (11 row(s) affected)-------------

INSERT INTO Dw\_Northwind.dbo.TD\_Discount(Discount)

SELECT DISTINCT Discount

FROM [Order Details]

WHERE Discount IS NOT NULL AND NOT EXISTS(SELECT Discount FROM Dw\_Northwind.dbo.TD\_Discount)

----------- TD\_OrderMonth(23 row(s) affected)-------------

INSERT INTO Dw\_Northwind.dbo.TD\_OrderMonth(OrderMonth, OrderYear)

SELECT DISTINCT CAST(MONTH(OrderDate) as varchar(2)) + '/' + CAST(YEAR(OrderDate) as varchar(4)) AS OrderMonth, YEAR(OrderDate) AS OrderYear

FROM Orders

WHERE OrderDate IS NOT NULL AND NOT EXISTS(SELECT OrderMonth FROM Dw\_Northwind.dbo.TD\_OrderMonth)

----------- TD\_Products(77 row(s) affected)-------------

USE Northwind

INSERT INTO Dw\_Northwind.dbo.TD\_Products(ProductID, ProductName, CategoryID)

SELECT ProductID, ProductName, CategoryID

FROM Products

WHERE NOT EXISTS (SELECT \* FROM Dw\_Northwind.dbo.TD\_Products)

------------------------------------------------------------------------------------------------------

/\*--------------- PriceLevel, SalesLevel, OrderLevel, StockLevel ------------------\*/

-------------------------------------------------------------------------------------------------------

----------- TD\_PriceLevel (3 row(s) affected)-------------

IF NOT EXISTS (SELECT PriceLevel from dw\_northwind.dbo.TD\_PriceLevel)

INSERT INTO Dw\_Northwind.dbo.TD\_PriceLevel (PriceLevel)

VALUES

('P1'),

('P2'),

('P3')

----------- TD\_SalesLevel (3 row(s) affected)-------------

IF NOT EXISTS (SELECT SalesLevel from dw\_northwind.dbo.TD\_SalesLevel)

INSERT INTO Dw\_Northwind.dbo.TD\_SalesLevel(SalesLevel)

VALUES

('Low'),

('Medium'),

('High')

----------- TD\_OrderLevel (3 row(s) affected)-------------

IF NOT EXISTS (SELECT OrderLevel from dw\_northwind.dbo.TD\_OrderLevel)

INSERT INTO Dw\_Northwind.dbo.TD\_OrderLevel(OrderLevel)

VALUES

('Low'),

('Medium'),

('High')

----------- TD\_StockLevel (3 row(s) affected)-------------

IF NOT EXISTS (SELECT StockLevel from dw\_northwind.dbo.TD\_StockLevel)

INSERT INTO Dw\_Northwind.dbo.TD\_StockLevel(StockLevel)

VALUES

('S0'),

('S1'),

('S2')

------------------------------------------------------------------------------------------------------

/\*------------------ TF\_Orders ------------------\*/

-------------------------------------------------------------------------------------------------------

INSERT INTO Dw\_Northwind.dbo.TF\_Orders(CustomerID,EmployeeID,OrderMonth,ShippedOrders,NotShippedOrders)

SELECT CustomerID, EmployeeID,

CAST(MONTH(OrderDate) as nvarchar(2)) +'/'+ CAST(YEAR(OrderDate) as nvarchar(4)) AS OrderMonth,

SUM(CASE WHEN ShippedDate IS NULL THEN 0 ELSE 1 END) AS ShippedOrders,

SUM(CASE WHEN ShippedDate IS NULL THEN 1 ELSE 0 END) AS NotShippedOrders

FROM Orders

WHERE NOT EXISTS (SELECT \* FROM Dw\_Northwind.dbo.TF\_Orders)

GROUP BY

CustomerID, EmployeeID, CAST(MONTH(OrderDate) as nvarchar(2)) +'/'+ CAST(YEAR(OrderDate) as nvarchar(4))

------------------------------------------------------------------------------------------------------

/\*------------------ TF\_Products ------------------\*/

-------------------------------------------------------------------------------------------------------

INSERT INTO Dw\_Northwind.dbo.TF\_Products(SupplierID, CategoryID, PriceLevel, StockLevel, NumberProducts)

SELECT SupplierID, CategoryID,

CASE

WHEN UnitPrice BETWEEN 1 AND 50 THEN 'P1'

WHEN UnitPrice BETWEEN 51 AND 100 THEN 'P2'

WHEN UnitPrice>100 THEN 'P3'

END AS PriceLevel,

CASE

WHEN UnitsInStock = 0 THEN 'S0'

WHEN UnitsInStock BETWEEN 1 AND 30 THEN 'S1'

WHEN UnitsInStock >30 THEN 'S2'

END AS StockLevel,

COUNT(DISTINCT ProductID) AS NumberProducts

FROM Products

WHERE NOT EXISTS (SELECT SupplierID, CategoryID, PriceLevel FROM Dw\_Northwind.dbo.TF\_Products)

GROUP BY SupplierID, CategoryID,

CASE

WHEN UnitPrice BETWEEN 1 AND 50 THEN 'P1'

WHEN UnitPrice BETWEEN 51 AND 100 THEN 'P2'

WHEN UnitPrice>100 THEN 'P3'

END,

CASE

WHEN UnitsInStock = 0 THEN 'S0'

WHEN UnitsInStock BETWEEN 1 AND 30 THEN 'S1'

WHEN UnitsInStock >30 THEN 'S2'

END

------------------------------------------------------------------------------------------------------

/\*------------------ TF\_Customers ------------------\*/

-------------------------------------------------------------------------------------------------------

insert into dw\_northwind.dbo.TF\_Customers(City,OrderLevel,SalesLevel,NumberCustomers)

SELECT City, OrderLevel, SalesLevel, COUNT (\*) AS NumberCustomers

FROM

(SELECT City,

CASE

WHEN COUNT(O.OrderID) between 1 and 10 THEN 'Low'

WHEN COUNT(O.OrderID) between 11 and 20 THEN 'Medium'

WHEN COUNT(O.OrderID) > 20 THEN 'High'

END AS OrderLevel,

CASE

WHEN SUM((OD.quantity\*OD.UnitPrice) - (OD.quantity\*OD.UnitPrice\*discount)) between 100 and 10000 THEN 'Low'

WHEN SUM((OD.quantity\*OD.UnitPrice) - (OD.quantity\*OD.UnitPrice\*discount)) between 10001 and 80000 THEN 'Medium'

WHEN SUM((OD.quantity\*OD.UnitPrice) - (OD.quantity\*OD.UnitPrice\*discount)) > 80000 THEN 'High'

END AS SalesLevel

FROM Customers C

JOIN Orders O ON O.CustomerID=C.CustomerID

JOIN [Order Details] OD ON O.OrderID=OD.OrderID

GROUP BY O.CustomerID, city) as T

WHERE NOT EXISTS (SELECT \* FROM dw\_northwind.dbo.TF\_Customers)

GROUP BY City, OrderLevel, SalesLevel

------------------------------------------------------------------------------------------------------

/\*------------------ TF\_Revenue ------------------\*/

-------------------------------------------------------------------------------------------------------

USE Northwind

INSERT INTO Dw\_Northwind.dbo.TF\_Revenue(OrderMonth,CustomerID, ProductID, CategoryID, revenueShipped, revenueNotShipped)

SELECT CAST(MONTH(OrderDate) as varchar(2)) +'/'+ CAST(YEAR(OrderDate) as varchar(4)) as OrderMonth,O.CustomerID, OD.ProductID, P.CategoryID,

SUM(CASE WHEN ShippedDate IS NOT NULL THEN (1-Discount) \* (OD.UnitPrice \* Quantity) ELSE 0 END) AS revenueShipped,

SUM(CASE WHEN ShippedDate IS NULL THEN (1-Discount) \* (OD.UnitPrice \* Quantity) ELSE 0 END) AS revenueNotShipped

FROM Orders O JOIN [Order Details] OD ON O.OrderID = OD.OrderID

JOIN Products P ON OD.ProductID = P.ProductID

WHERE NOT EXISTS( SELECT \* FROM Dw\_Northwind.dbo.TF\_Revenue)

GROUP BY CAST(MONTH(OrderDate) as varchar(2)) +'/'+ CAST(YEAR(OrderDate) as varchar(4)),O.CustomerID, OD.ProductID, P.CategoryID

------------------------------------------------------------------------------------------------------

/\*------------------ TF\_Discount ------------------\*/

-------------------------------------------------------------------------------------------------------

USE Northwind

INSERT INTO Dw\_Northwind.dbo.TF\_Discount(OrderMonth, CustomerID, Discount, DiscountValue)

SELECT CAST(MONTH(OrderDate) as varchar(2)) +'/'+ CAST(YEAR(OrderDate) as varchar(4)) as OrderMonth,O.CustomerID, OD.Discount,

SUM(UnitPrice \* Quantity \* Discount) AS DiscountValue

FROM Orders O JOIN [Order Details] OD ON O.OrderID = OD.OrderID

WHERE NOT EXISTS( SELECT \* FROM Dw\_Northwind.dbo.TF\_Discount)

GROUP BY CAST(MONTH(OrderDate) as varchar(2)) +'/'+ CAST(YEAR(OrderDate) as varchar(4)),O.CustomerID, OD.Discount