**Landing Layer**

drop table if exists landing.lnd\_Customers;

drop table if exists landing.[lnd\_Order Details];

drop table if exists landing.lnd\_categories;

drop table if exists landing.lnd\_Suppliers;

drop table if exists landing.lnd\_Employees;

drop table if exists landing.lnd\_Products;

drop table if exists landing.lnd\_Orders;

drop table if exists landing.lnd\_Shippers;

select \* into landing.lnd\_Customers from dbo.Customers;

alter table landing.lnd\_Customers add ModifiedDate datetime, SourceTable varchar(30);

update landing.lnd\_Customers set ModifiedDate = CURRENT\_TIMESTAMP,SourceTable='Customers';

select \* into landing.[lnd\_Order Details] from dbo.[Order Details];

alter table landing.[lnd\_Order Details] add ModifiedDate datetime,SourceTable varchar(30);

update landing.[lnd\_Order Details] set ModifiedDate = CURRENT\_TIMESTAMP,SourceTable='Order Details';

select \* into landing.lnd\_categories from dbo.Categories;

alter table landing.lnd\_categories add ModifiedDate datetime, SourceTable varchar(30);

update landing.lnd\_categories set ModifiedDate = CURRENT\_TIMESTAMP,SourceTable='categories';

select \* into landing.lnd\_Suppliers from dbo.Suppliers;

alter table landing.lnd\_Suppliers add ModifiedDate datetime, SourceTable varchar(50);

update landing.lnd\_Suppliers set ModifiedDate = CURRENT\_TIMESTAMP,SourceTable='Suppliers';

select \* into landing.lnd\_Employees from dbo.Employees;

alter table landing.lnd\_Employees add ModifiedDate datetime,SourceTable varchar(30);

update landing.lnd\_Employees set ModifiedDate = CURRENT\_TIMESTAMP,SourceTable='Employees';

select \* into landing.lnd\_Products from dbo.Products;

alter table landing.lnd\_Products add ModifiedDate datetime, SourceTable varchar(max);

update landing.lnd\_Products set ModifiedDate = CURRENT\_TIMESTAMP,SourceTable='Products';

select \* into landing.lnd\_Orders from dbo.Orders;

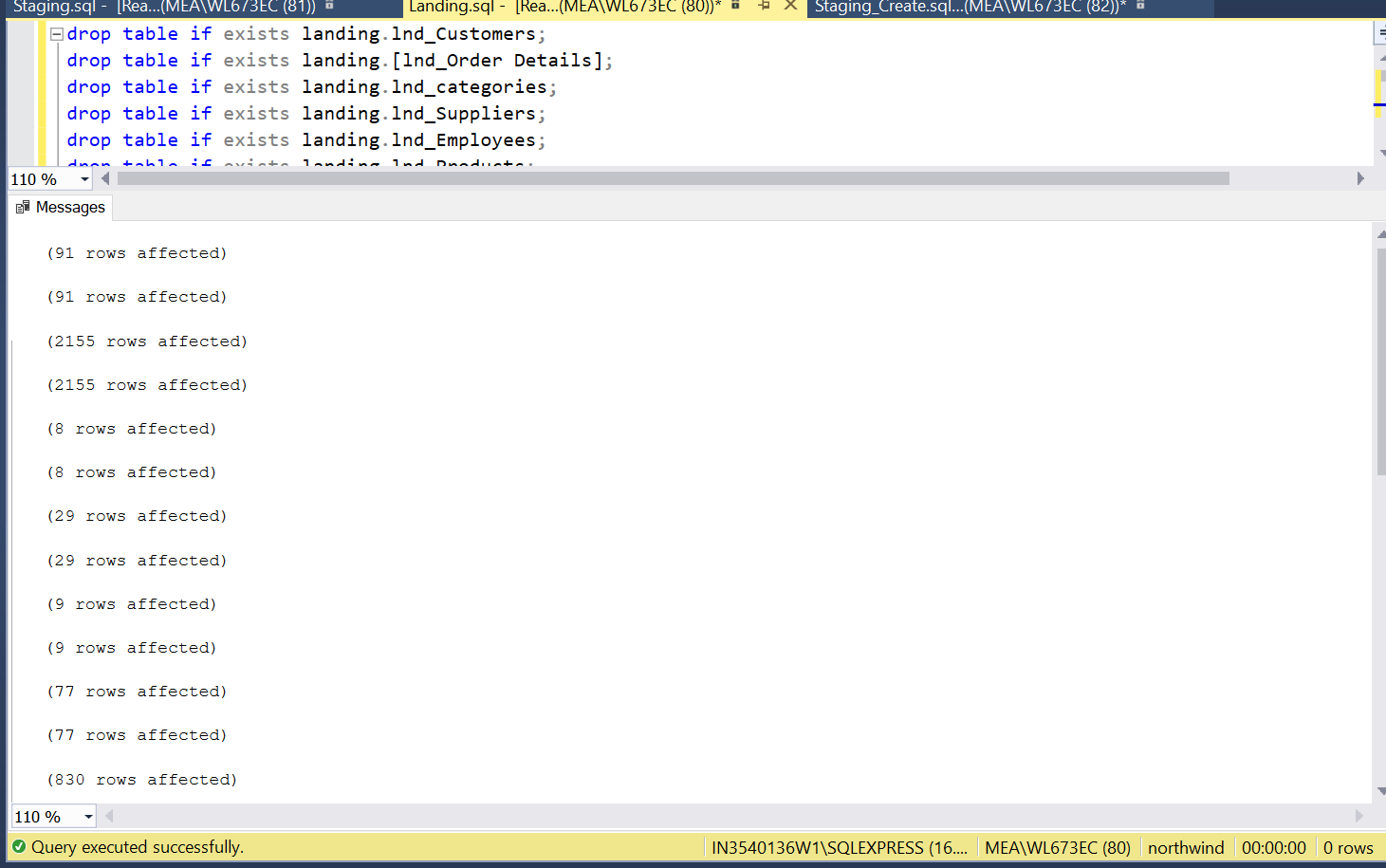
alter table landing.lnd\_Orders add ModifiedDate datetime, SourceTable varchar(50);

update landing.lnd\_Orders set ModifiedDate = CURRENT\_TIMESTAMP,SourceTable='Orders';

select \* into landing.lnd\_Shippers from dbo.Shippers;

alter table landing.lnd\_Shippers add ModifiedDate datetime,SourceTable varchar(30);

update landing.lnd\_Shippers set ModifiedDate = CURRENT\_TIMESTAMP,SourceTable='Shippers';



**Staging Layer**

drop table if exists staging.stg\_Customers;

drop table if exists staging.[stg\_Order Details];

drop table if exists staging.stg\_categories;

drop table if exists staging.stg\_Suppliers;

drop table if exists staging.stg\_Employees;

drop table if exists staging.stg\_Products;

drop table if exists staging.stg\_Orders;

drop table if exists staging.stg\_Shippers;

select \* into staging.stg\_Suppliers from landing.lnd\_Suppliers where 1=2;

alter table staging.stg\_Suppliers drop column [Fax],[HomePage];

select \* into staging.stg\_Orders from landing.lnd\_Orders where 1=2;

select \* into staging.stg\_Customers from landing.lnd\_Customers where 1=2;

alter table staging.stg\_Customers drop column Fax;

select \* into staging.[stg\_Order Details] from landing.[lnd\_Order Details] where 1=2;

select \* into staging.stg\_categories from landing.lnd\_Categories where 1=2;

alter table staging.stg\_categories drop column Picture;

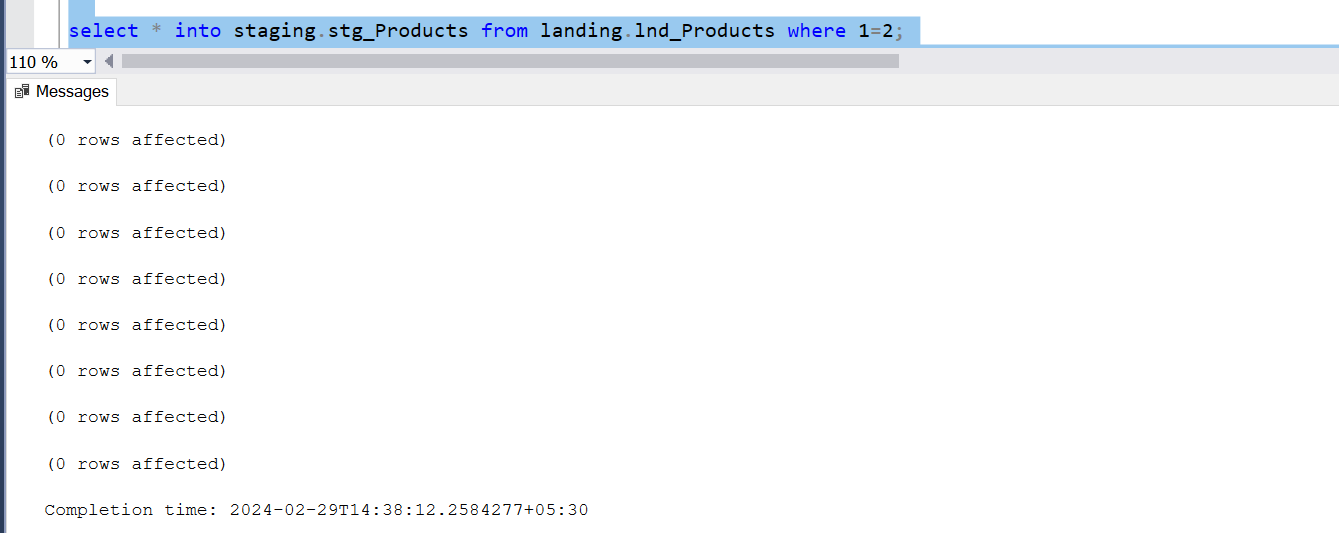
select \* into staging.stg\_Employees from landing.lnd\_Employees where 1=2;

alter table staging.stg\_Employees drop column [Title],[TitleOfCourtesy],[Address],[City],[PostalCode],[HomePhone],[Extension],[Photo],[Notes],[ReportsTo],[PhotoPath];

select \* into staging.stg\_Products from landing.lnd\_Products where 1=2;

alter table staging.stg\_Products drop column [SupplierID],[CategoryID],[QuantityPerUnit],[UnitsInStock],[UnitsOnOrder],[ReorderLevel];

select \* into staging.stg\_Shippers from landing.lnd\_Shippers where 1=2;



merge staging.stg\_Suppliers as tab1

using landing.lnd\_Suppliers as tab2 on tab1.SupplierID = tab2.SupplierID

when not matched

then insert(CompanyName,ContactName,ContactTitle,Address,City,Region,PostalCode,Country,Phone,ModifiedDate,SourceTable)

values(tab2.CompanyName, tab2.ContactName, tab2.ContactTitle, tab2.Address, tab2.City, tab2.Region, tab2.PostalCode,

tab2.Country, tab2.Phone, current\_timestamp, 'lnd\_Suppliers')

when not matched by source

then delete

when matched and

tab1.CompanyName != tab2.CompanyName or

tab1.ContactName != tab2.ContactName or

tab1.ContactTitle != tab2.ContactTitle or

tab1.Address != tab2.Address or

tab1.City != tab2.City or

tab1.Region != tab2.Region or

tab1.PostalCode != tab2.PostalCode or

tab1.Country != tab2.Country or

tab1.Phone != tab2.Phone

then update

set

tab1.CompanyName = tab2.CompanyName,

tab1.ContactName = tab2.ContactName,

tab1.ContactTitle = tab2.ContactTitle,

tab1.Address = tab2.Address,

tab1.City = tab2.City,

tab1.Region = tab2.Region,

tab1.PostalCode = tab2.PostalCode,

tab1.Country = tab2.Country,

tab1.Phone = tab2.Phone,

tab1.ModifiedDate = current\_timestamp;

set identity\_insert staging.stg\_Orders on;

merge staging.stg\_Orders as tab1

using landing.lnd\_Orders as tab2 on tab1.OrderID = tab2.OrderID

when not matched

then insert(OrderID,CustomerID,EmployeeID,OrderDate,RequiredDate,ShippedDate,ShipVia,Freight,ShipName,ShipAddress,ShipCity,ShipRegion,

ShipPostalCode,ShipCountry,ModifiedDate,SourceTable)

values(tab2.OrderID,tab2.CustomerID,tab2.EmployeeID,tab2.OrderDate,tab2.RequiredDate,tab2.ShippedDate,tab2.ShipVia,tab2.Freight,tab2.ShipName,

tab2.ShipAddress,tab2.ShipCity,tab2.ShipRegion,tab2.ShipPostalCode,tab2.ShipCountry,CURRENT\_TIMESTAMP,'lnd\_Orders')

when not matched by source

then delete

when matched and

tab1.CustomerID!=tab2.CustomerID or

tab1.EmployeeID!=tab2.EmployeeID or

tab1.OrderDate!=tab2.OrderDate or

tab1.RequiredDate!=tab2.RequiredDate or

tab1.ShippedDate!=tab2.ShippedDate or

tab1.ShipVia!=tab2.ShipVia or

tab1.Freight!=tab2.Freight or

tab1.ShipName!=tab2.ShipName or

tab1.ShipAddress!=tab2.ShipAddress or

tab1.ShipCity!=tab2.ShipCity or

tab1.ShipRegion!=tab2.ShipRegion or

tab1.ShipPostalCode!=tab2.ShipPostalCode or

tab1.ShipCountry!=tab2.ShipCountry

then update

set

tab1.CustomerID=tab2.CustomerID,

tab1.EmployeeID=tab2.EmployeeID,

tab1.OrderDate=tab2.OrderDate,

tab1.RequiredDate=tab2.RequiredDate,

tab1.ShippedDate=tab2.ShippedDate,

tab1.ShipVia=tab2.ShipVia,

tab1.Freight=tab2.Freight,

tab1.ShipName=tab2.ShipName,

tab1.ShipAddress=tab2.ShipAddress,

tab1.ShipCity=tab2.ShipCity,

tab1.ShipRegion=tab2.ShipRegion,

tab1.ShipPostalCode=tab2.ShipPostalCode,

tab1.ShipCountry=tab2.ShipCountry,

tab1.ModifiedDate = CURRENT\_TIMESTAMP;

merge staging.stg\_Customers as tab1

using landing.lnd\_Customers as tab2

on tab1.CustomerID=tab2.CustomerID

when not matched then

insert (CustomerID,CompanyName,ContactName,ContactTitle,Address,City,Region,PostalCode,Country,Phone,ModifiedDate,SourceTable)

values (tab2.CustomerID,tab2.CompanyName,tab2.ContactName,tab2.ContactTitle,tab2.Address,tab2.City,tab2.Region,tab2.PostalCode,tab2.Country,tab2.Phone,current\_timestamp,'lnd\_Customers')

when not matched by source

then delete

when matched and

(tab1.CustomerID != tab2.CustomerID or

tab1.CompanyName != tab2.CompanyName or

tab1.ContactName != tab2.ContactName or

tab1.ContactTitle != tab2.ContactTitle or

tab1.Address != tab2.Address or

tab1.City != tab2.City or

tab1.Region != tab2.Region or

tab1.PostalCode != tab2.PostalCode or

tab1.Country != tab2.Country or

tab1.Phone != tab2.Phone)

then update set tab1.CustomerID = tab2.CustomerID ,

tab1.CompanyName = tab2.CompanyName ,

tab1.ContactName = tab2.ContactName ,

tab1.ContactTitle = tab2.ContactTitle ,

tab1.Address = tab2.Address ,

tab1.City = tab2.City ,

tab1.Region = tab2.Region ,

tab1.PostalCode = tab2.PostalCode ,

tab1.Country = tab2.Country ,

tab1.Phone = tab2.Phone ,

tab1.ModifiedDate = current\_timestamp;

merge staging.[stg\_Order Details] as tab1

using landing.[lnd\_Order Details] as tab2

on tab1.orderid=tab2.orderid and tab1.productID=tab2.productID

when not matched then

insert (orderid,productid,unitprice,quantity,discount,ModifiedDate,SourceTable)

values (tab2.orderid,tab2.productid,tab2.unitprice,tab2.quantity,tab2.discount,current\_timestamp,'lnd\_Order Details')

when not matched by source

then delete

when matched and

tab1.orderid != tab2.orderid or

tab1.productid!= tab2.productid or

tab1.unitprice!=tab2.unitprice or

tab1.quantity!=tab2.quantity or

tab1.discount!=tab2.discount

then update set

tab1.orderid = tab2.orderid ,

tab1.productid = tab2.productid ,

tab1.unitprice=tab2.unitprice ,

tab1.quantity=tab2.quantity ,

tab1.discount=tab2.discount,

tab1.modifieddate=current\_timestamp;

set identity\_insert staging.stg\_Orders off;

set identity\_insert staging.stg\_categories on;

merge staging.stg\_categories as tab1

using landing.lnd\_Categories as tab2

on (tab1.CategoryID=tab2.CategoryID)

when not matched then

insert (CategoryID,CategoryName,Description,ModifiedDate,SourceTable)

values (tab2.CategoryID,tab2.CategoryName,tab2.Description,current\_timestamp,'lnd\_Categories')

when not matched by source

then delete

when matched and

(tab1.CategoryName != tab2.CategoryName or

convert(nvarchar(max),tab1.Description) <> convert(nvarchar(max),tab2.Description))

then update set

tab1.CategoryName = tab2.CategoryName ,

tab1.Description = tab2.Description ,

tab1.ModifiedDate=current\_timestamp;

set identity\_insert staging.stg\_Categories off;

set identity\_insert staging.stg\_employees on;

merge staging.stg\_Employees as tab1

using landing.lnd\_Employees as tab2

on tab1.EmployeeID=tab2.EmployeeID

when not matched then

insert (EmployeeID,LastName,Firstname,BirthDate,HireDate,Region,Country,ModifiedDate,SourceTable)

values (tab2.EmployeeID,tab2.LastName,tab2.Firstname,tab2.BirthDate,

tab2.HireDate,Region,tab2.Country,current\_timestamp,'lnd\_Employees')

when not matched by source

then delete

when matched and

tab1.LastName != tab2.LastName or

tab1.Firstname != tab2.Firstname or

tab1.BirthDate != tab2.BirthDate or

tab1.HireDate != tab2.HireDate or

tab1.Region != tab2.Region or

tab1.Country != tab2.Country

then update set

tab1.LastName = tab2.LastName ,

tab1.Firstname = tab2.Firstname ,

tab1.BirthDate = tab2.BirthDate ,

tab1.HireDate = tab2.HireDate ,

tab1.Region = tab2.Region ,

tab1.Country = tab2.Country ,

tab1.Modifieddate=current\_timestamp;

set identity\_insert staging.stg\_employees off;

set identity\_insert staging.stg\_Products on;

merge staging.stg\_Products as tab1

using landing.lnd\_Products as tab2

on tab1.ProductID=tab2.ProductID

when not matched then

insert (ProductID,ProductName,UnitPrice, Discontinued, ModifiedDate, SourceTable)

values (tab2.ProductID, tab2.ProductName, tab2.UnitPrice, tab2.Discontinued,current\_timestamp,'lnd\_Products')

when not matched by source

then delete

when matched and

tab1.ProductName != tab2.ProductName or

tab1.UnitPrice != tab2.UnitPrice or

tab1.Discontinued != tab2.Discontinued

then update set tab1.ProductName = tab2.ProductName ,

tab1.UnitPrice = tab2.UnitPrice ,

tab1.Discontinued = tab2.Discontinued,

tab1.ModifiedDate = current\_timestamp;

set identity\_insert staging.stg\_Products off;

set identity\_insert staging.stg\_Shippers on;

merge staging.stg\_Shippers as tab1

using landing.lnd\_Shippers as tab2

on tab1.shipperid=tab2.shipperid

when not matched then

insert (shipperid,companyname,phone,ModifiedDate,SourceTable)

values (tab2.shipperid,tab2.companyname,tab2.phone,current\_timestamp,'lnd\_Shippers')

when not matched by source

then delete

when matched and

tab1.shipperid !=tab2.shipperid or

tab1.companyname!=tab2.companyname or

tab1.phone!=tab2.phone

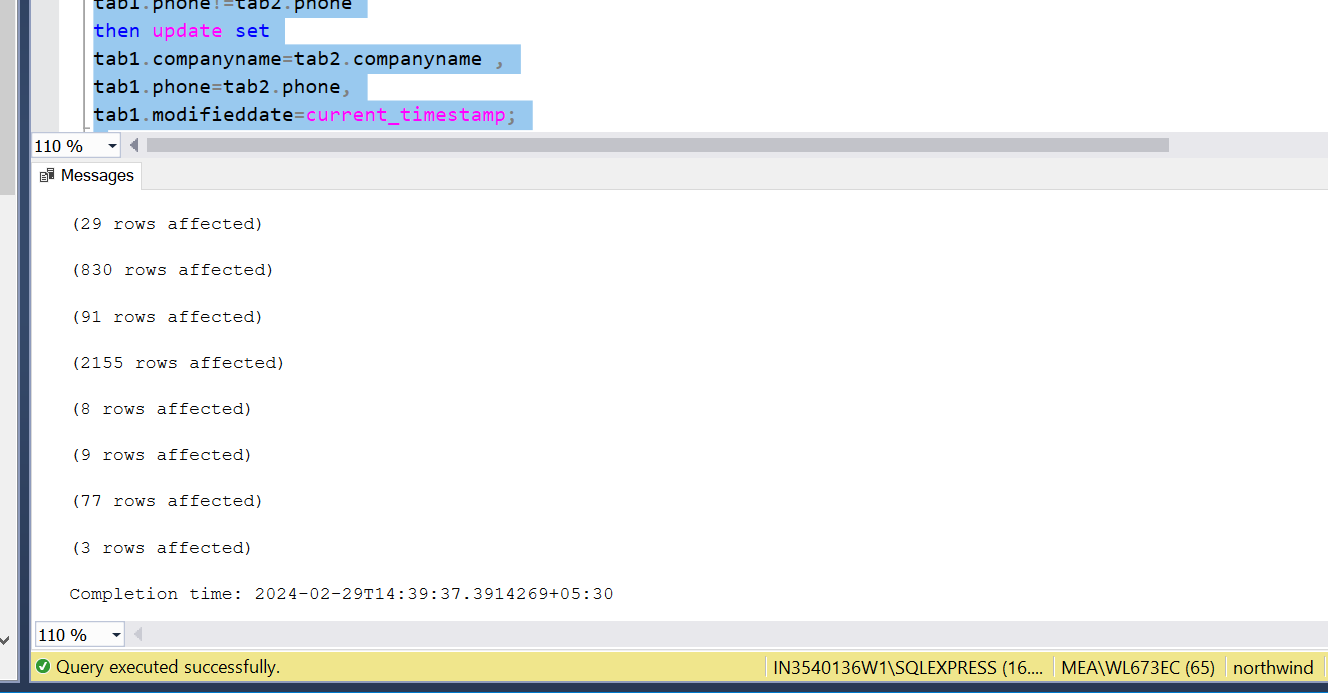
then update set

tab1.companyname=tab2.companyname ,

tab1.phone=tab2.phone,

tab1.modifieddate=current\_timestamp;

set identity\_insert staging.stg\_Shippers off;



**Dimension Tables**

drop table if exists DW.Customers\_Dim

DROP TABLE IF EXISTS DW.Employees\_Dim

DROP TABLE IF EXISTS DW.Suppliers\_Dim

drop table if exists DW.Categories\_Dim

drop table if exists DW.Products\_Dim

CREATE TABLE DW.Customers\_Dim(

CustomerKey int identity(1,1) primary key,

[CustomerID] [varchar](10) NOT NULL,

[CompanyName] [nvarchar](50) NOT NULL,

[ContactName] [nvarchar](50) NULL,

[ContactTitle] [nvarchar](50) NULL,

[Address] [nvarchar](60) NULL,

[City] [nvarchar](20) NULL,

[Region] [nvarchar](20) NULL,

[PostalCode] [nvarchar](20) NULL,

[Country] [nvarchar](20) NULL,

[Phone] [nvarchar](20) NULL,

[StartDate] [datetime] NULL,

[EndDate] [datetime] NULL);

CREATE TABLE DW.Employees\_Dim (

"EmployeeKey" "int" identity(1,1) primary key,

"EmployeeID" "int" NOT NULL,

"LastName" nvarchar (max) NOT NULL,

"FirstName" nvarchar (max) NOT NULL,

"BirthDate" "datetime" NULL,

"HireDate" "datetime" NULL,

"Region" nvarchar (max) NULL,

"Country" nvarchar (max) NULL,

"StartDate" "datetime" NULL,

"EndDate" "datetime" NULL

);

CREATE TABLE DW.Suppliers\_Dim(

SupplierKey int IDENTITY(1,1) primary key,

"SupplierID" "int" NOT NULL ,

"CompanyName" nvarchar (max) NOT NULL ,

"ContactName" nvarchar (max) NULL ,

"ContactTitle" nvarchar (max) NULL ,

"Address" nvarchar (max) NULL ,

"City" nvarchar (max) NULL ,

"Region" nvarchar (max) NULL ,

"PostalCode" nvarchar (max) NULL ,

"Country" nvarchar (max) NULL ,

"Phone" nvarchar (max) NULL,

"StartDate" [datetime] NULL,

"EndDate" [datetime] NULL

);

CREATE TABLE DW.Categories\_Dim(

CategoriesKey int identity(1,1),

"CategoryID" "int" NOT NULL ,

"CategoryName" nvarchar (15) NOT NULL ,

"Description" nvarchar (max) NULL,

StartDate datetime NULL,

EndDate datetime NULL);

CREATE TABLE DW.Products\_Dim(

"ProductKey" int identity(1,1) primary key,

"ProductID" "int" NOT NULL ,

"ProductName" nvarchar (40) NOT NULL ,

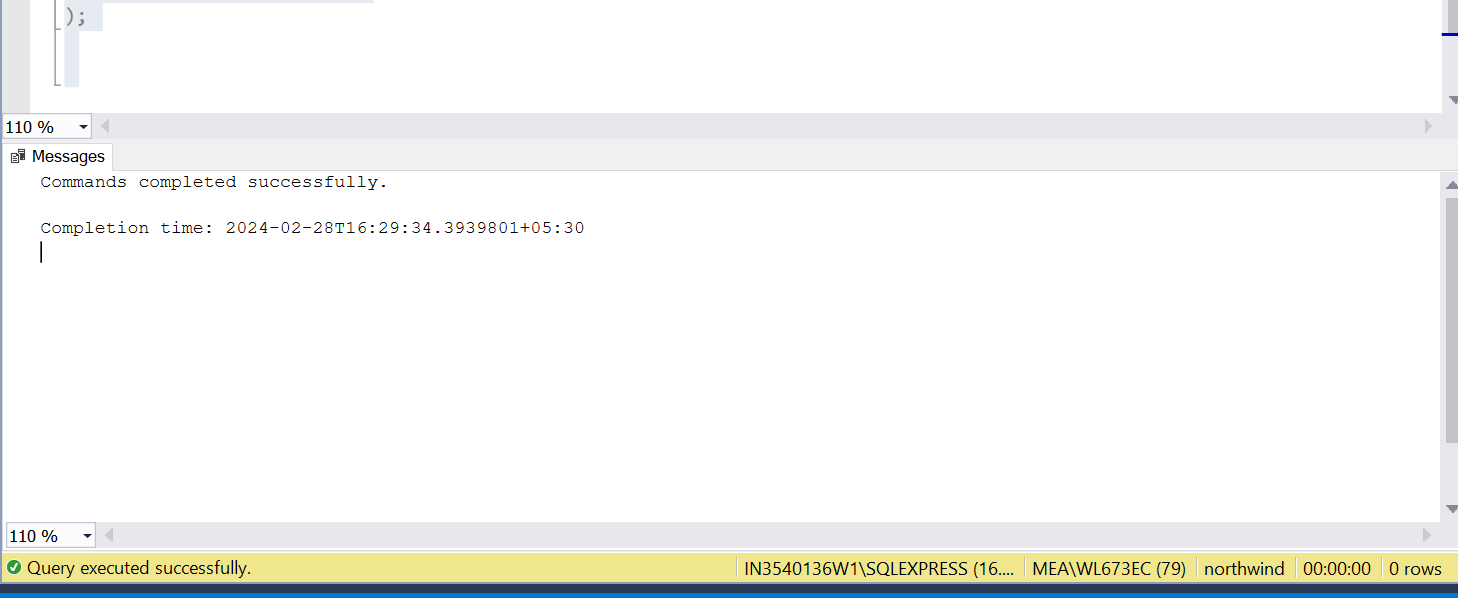
"UnitPrice" nvarchar (max) NULL ,

"Discontinued" "int" NOT NULL,

"StartDate" [datetime] NULL,

"EndDate" [datetime] NULL

);



insert into DW.Customers\_Dim(CustomerID,CompanyName,ContactName,ContactTitle,Address,City,Region,PostalCode,Country,Phone,StartDate,EndDate)

select CustomerID,CompanyName,ContactName,ContactTitle,Address,City,Region,PostalCode,Country,Phone,CURRENT\_TIMESTAMP,NULL from (

merge DW.Customers\_Dim as tab1

using staging.stg\_customers as tab2

on tab1.CustomerID=tab2.CustomerID

when not matched then

insert (CustomerID,CompanyName,ContactName,ContactTitle,Address,City,Region,PostalCode,Country,Phone,StartDate)

values (tab2.CustomerID,tab2.CompanyName,tab2.ContactName,tab2.ContactTitle,tab2.Address,tab2.City,tab2.Region,tab2.PostalCode,

tab2.Country,tab2.Phone,Current\_Timestamp)

when matched and

tab1.Enddate is NULL and

(tab1.CustomerID != tab2.CustomerID or

tab1.CompanyName != tab2.CompanyName or

tab1.ContactName != tab2.ContactName or

tab1.ContactTitle != tab2.ContactTitle or

tab1.Address != tab2.Address or

tab1.City != tab2.City or

tab1.Region != tab2.Region or

tab1.PostalCode != tab2.PostalCode or

tab1.Country != tab2.Country or

tab1.Phone != tab2.Phone )

then

update set tab1.EndDate=Current\_Timestamp

output tab2.CustomerID,tab2.CompanyName,tab2.ContactName,tab2.ContactTitle,tab2.Address,tab2.City,tab2.Region,tab2.PostalCode,

tab2.Country,tab2.Phone,

$action as MergeAction) as MRG

where MRG.MergeAction = 'UPDATE';

insert into DW.Categories\_Dim (CategoryID,CategoryName,Description,StartDate,EndDate)

select CategoryID,CategoryName,Description,CURRENT\_TIMESTAMP,NULL from (

merge DW.Categories\_Dim as tab1

using staging.stg\_Categories as tab2

on tab1.CategoryID=tab2.CategoryID

when not matched then

insert (CategoryID,CategoryName,Description,StartDate)

values (tab2.CategoryID,tab2.CategoryName,tab2.Description,current\_timestamp)

when matched and

tab1.Enddate is NULL and

(tab1.CategoryName != tab2.CategoryName or

convert(nvarchar(max),tab1.Description) <>convert(nvarchar(max),tab2.Description) )

then update set tab1.EndDate=Current\_Timestamp

output tab2.CategoryID,tab2.CategoryName,tab2.Description,

$action as MergeAction) as MRG

where MRG.MergeAction = 'UPDATE';

insert into DW.Suppliers\_Dim(SupplierID, CompanyName,ContactName,ContactTitle, Address, City, Region, PostalCode,

Country,Phone,StartDate, EndDate)

select SupplierID, CompanyName, ContactName, ContactTitle, Address, City, Region,PostalCode, Country,

Phone, CURRENT\_TIMESTAMP, NULL from(

merge into DW.Suppliers\_Dim as tab1

using staging.stg\_Suppliers as tab2 on tab1.SupplierID=tab2.SupplierID

when not matched then

insert(SupplierID,CompanyName,ContactName,ContactTitle,Address,City,Region,PostalCode,Country,Phone,StartDate)

values(tab2.SupplierID,tab2.CompanyName,tab2.ContactName,tab2.ContactTitle, tab2.Address, tab2.City,tab2.Region,

tab2.PostalCode,tab2.Country,tab2.Phone,CURRENT\_TIMESTAMP)

when matched and

tab1.Enddate is NULL and

(tab1.SupplierID != tab2.SupplierID or

tab1.CompanyName != tab2.CompanyName or

tab1.ContactName != tab2.ContactName or

tab1.ContactTitle != tab2.ContactTitle or

tab1.Address != tab2.Address or

tab1.City != tab2.City or

tab1.Region != tab2.Region or

tab1.PostalCode != tab2.PostalCode or

tab1.Country != tab2.Country or

tab1.Phone != tab2.Phone)

then

update set EndDate = CURRENT\_TIMESTAMP

output tab2.SupplierID,tab2.CompanyName,tab2.ContactName,tab2.ContactTitle,tab2.Address,tab2.City,tab2.Region,tab2.PostalCode,

tab2.Country,tab2.Phone,

$action as MergeAction) as MRG

where MRG.MergeAction = 'UPDATE';

insert into DW.Employees\_Dim(EmployeeID,LastName,FirstName,BirthDate,HireDate,Region,Country,StartDate,EndDate)

select EmployeeID,LastName,FirstName,BirthDate,HireDate,Region,Country, CURRENT\_TIMESTAMP, NULL from(

merge into DW.Employees\_Dim as tab1

using staging.stg\_Employees as tab2 on tab1.EmployeeID=tab2.EmployeeID

when not matched then

insert(EmployeeID,LastName,FirstName,BirthDate,HireDate,Region,Country,StartDate)

values(tab2.EmployeeID,tab2.LastName,tab2.FirstName,tab2.BirthDate, tab2.HireDate, tab2.Region,tab2.Country,

CURRENT\_TIMESTAMP)

when matched and

tab1.Enddate is NULL and

(tab1.EmployeeID != tab2.EmployeeID or

tab1.LastName != tab2.LastName or

tab1.FirstName != tab2.FirstName or

tab1.BirthDate != tab2.BirthDate or

tab1.HireDate != tab2.HireDate or

tab1.Region != tab2.Region or

tab1.Country != tab2.Country)

then

update set EndDate = CURRENT\_TIMESTAMP

output tab2.EmployeeID,tab2.LastName,tab2.FirstName,tab2.BirthDate,tab2.HireDate,tab2.Region,tab2.Country,

$action as MergeAction) as MRG

where MRG.MergeAction = 'UPDATE';

insert into DW.Products\_Dim(ProductID,ProductName,UnitPrice,Discontinued,StartDate,EndDate)

select ProductID,ProductName,UnitPrice,Discontinued,CURRENT\_TIMESTAMP,NULL from (

merge DW.Products\_Dim as tab1

using staging.stg\_Products as tab2

on tab1.ProductID=tab2.ProductID

when not matched then

insert (ProductID,ProductName,UnitPrice,Discontinued,StartDate)

values (tab2.ProductID,tab2.ProductName,tab2.UnitPrice,tab2.Discontinued,Current\_Timestamp)

when matched and

tab1.Enddate is NULL and

(tab1.ProductID != tab2.ProductID or

tab1.ProductName != tab2.ProductName or

tab1.UnitPrice != tab2.UnitPrice or

tab1.Discontinued != tab2.Discontinued )

then

update set tab1.EndDate=Current\_Timestamp

output tab2.ProductID,tab2.ProductName,tab2.UnitPrice,tab2.Discontinued,

$action as MergeAction) as MRG

where MRG.MergeAction = 'UPDATE';

DROP TABLE IF EXISTS DW.Calendar\_Dim;

CREATE TABLE DW.Calendar\_Dim(

CalendarKey INT NOT NULL IDENTITY(1,1),

FullDate datetime,

DAYOFWEEK char(15),

DayType char(20),

DAYOFMONTH int,

MONTH char(10),

QUARTER char(2),

YEAR int,

PRIMARY KEY (CalendarKey));

INSERT INTO DW.Calendar\_Dim (FullDate, DAYOFWEEK, DayType, DAYOFMONTH, MONTH, QUARTER, YEAR)

SELECT DISTINCT(CONVERT(varchar, O.OrderDate, 23)) date, DATENAME(DW,O.OrderDate),

Choose(DATEPART(DW,O.OrderDate),'Weekend','Weekday','Weekday','Weekday', 'Weekday','Weekday','Weekend'),

DATEPART(DAY, O.OrderDate), MONTH(O.OrderDate), CONCAT('Q', DATEPART(q,O.OrderDate)), YEAR(O.OrderDate)

FROM Staging.stg\_Orders O;

drop table if exists DW.CustomerEmployee\_Fact;

Create table DW.CustomerEmployee\_Fact(

CustomerKey INT,

EmployeeKey INT,

CalendarKey INT,

OrderID INT,

Sales money);

insert into DW.CustomerEmployee\_Fact(CustomerKey, EmployeeKey, CalendarKey, OrderID,Sales)

select

ISNULL(C.CustomerKey,0),

ISNULL(E.EmployeeKey,0),

ISNULL(CD.CalendarKey, 0),

ISNULL(O.OrderId, 0),

s.ord

from staging.stg\_Orders O left join DW.Customers\_Dim C on O.CustomerID=C.CustomerID

left join DW.Employees\_Dim E on O.EmployeeID=E.EmployeeID

left join DW.Calendar\_Dim CD on O.OrderDate=CD.FullDate

left join (select OrderID,sum(UnitPrice\*Quantity) as ord from staging.[stg\_Order Details] group by OrderID)as s on s.OrderID=O.orderID;

drop table if exists DW.ProductInStock\_fact;

create table DW.ProductInStock\_Fact(

CalendarKey int,

ProductKey int,

CategoriesKey int,

SupplierKey int,

UnitinStock int,

UnitsOnOrder int,

ReorderLevel int,

TotalQuantity int,

OrderID int);

insert into DW.ProductInStock\_Fact(CalendarKey,ProductKey,CategoriesKey,SupplierKey,UnitinStock,UnitsOnOrder,

ReorderLevel,TotalQuantity,OrderID)

select

ISNULL(CD.CalendarKey, 0),

ISNULL(PD.ProductKey,0),

ISNULL(C.CategoriesKey, 0),

ISNULL(S.SupplierKey, 0),

ISNULL(P.UnitsInStock, 0),

ISNULL(P.UnitsOnOrder, 0),

ISNULL(P.ReorderLevel, 0),

ISNULL(P.UnitsInStock+P.UnitsOnOrder,0),

ISNULL(O.OrderId, 0)

from

staging.stg\_Orders O left join staging.[stg\_Order Details] OD on O.OrderID=OD.OrderID

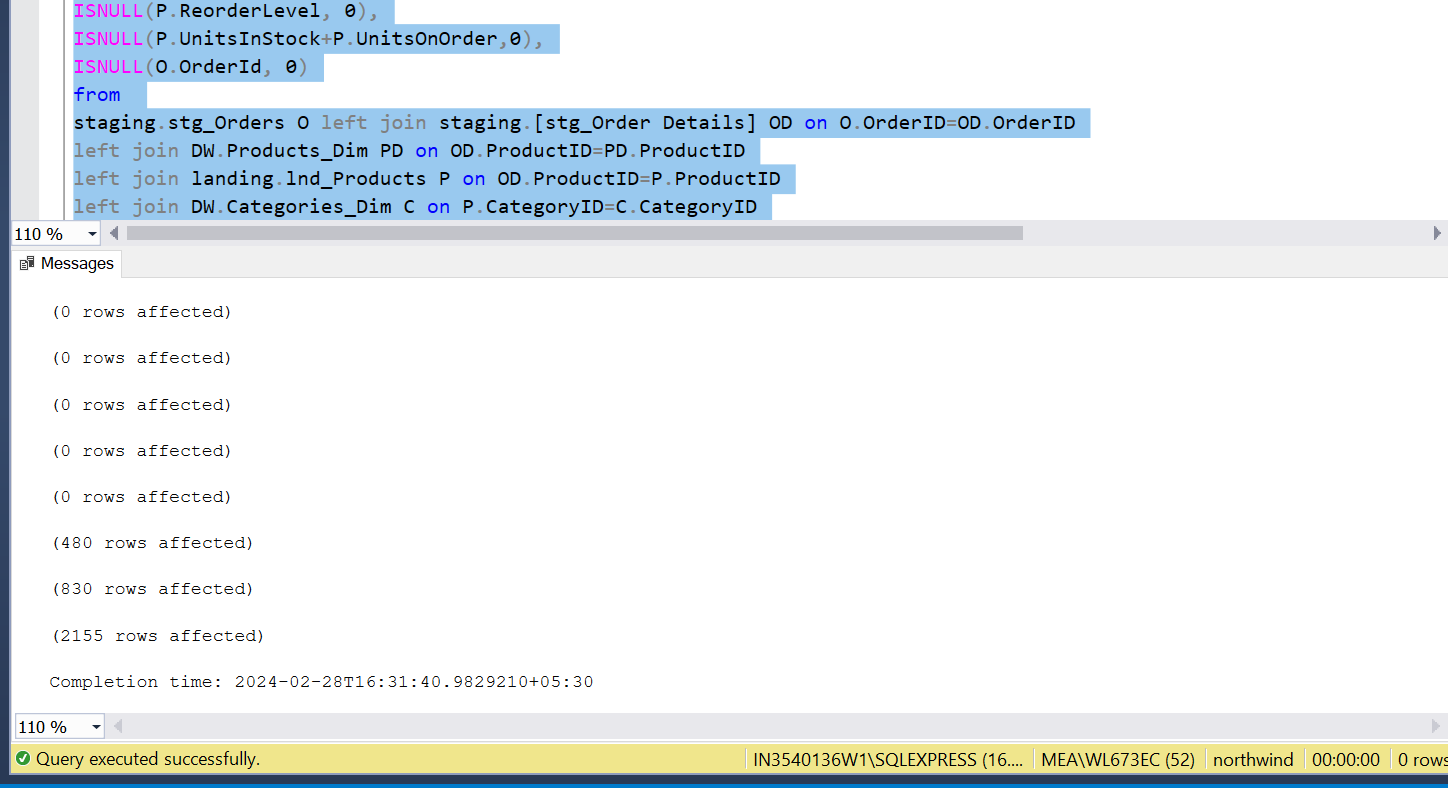
left join DW.Products\_Dim PD on OD.ProductID=PD.ProductID

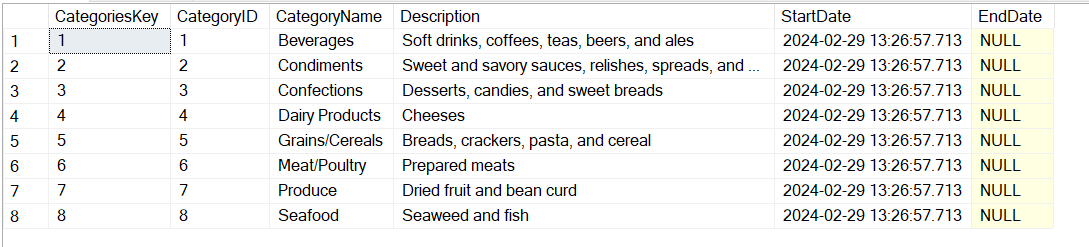
left join landing.lnd\_Products P on OD.ProductID=P.ProductID

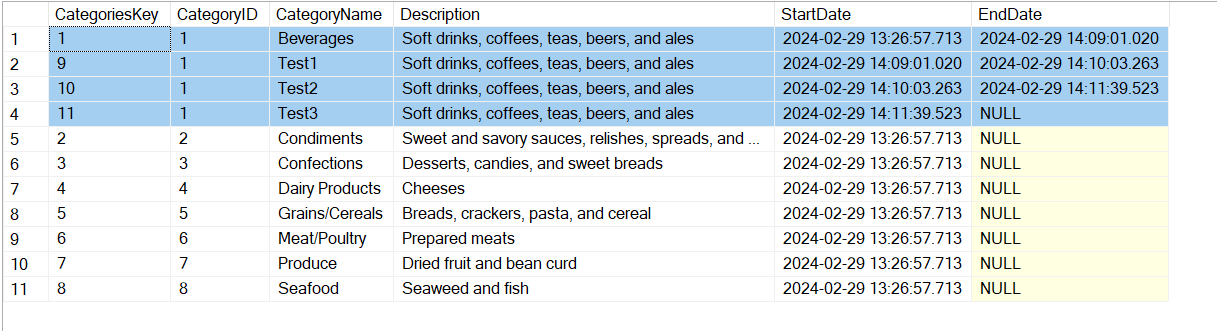
left join DW.Categories\_Dim C on P.CategoryID=C.CategoryID

left join DW.Suppliers\_Dim S on P.SupplierID=S.SupplierID

left join DW.Calendar\_Dim CD on O.OrderDate = CD.FullDate;







**PYTHON**

def function1():

    with open('landing.py', 'r') as file:

        exec(file.read())

def function2():

    with open('stg create.py', 'r') as file1:

        exec(file1.read())

    with open('dim create.py', 'r') as file2:

        exec(file2.read())

def function3():

    with open('staging.py', 'r') as file:

        exec(file.read())

def function4():

    with open('dim.py', 'r') as file:

        exec(file.read())

def function5():

    with open('landing.py', 'r') as file:

        exec(file.read())

    with open('staging.py', 'r') as file:

        exec(file.read())

    with open('dim.py', 'r') as file:

        exec(file.read())

def default():

    print('Invalid Input')

switcher = {

    1: function1,

    2: function2,

    3: function3,

    4: function4,

    5: function5

}

def switch(case):

    switcher.get(case, default)()

n=int(input(('\n Enter the operation:\n1.Landing\n2.Create stg and dim tables\n3.Staging\n4.Dimension\n5.OLTP to DW\n')))

switch(n)

**Landing:**

import pyodbc

conn = pyodbc.connect('DRIVER={SQL Server};'

                      'SERVER=IN3540136W1\SQLEXPRESS;'

                      'DATABASE=nw;'

                      'Trusted\_Connection=yes')

cursor = conn.cursor()

with open('C:/Users/LD184YA/OneDrive - EY/Documents/SQL Server Management Studio/Landing.sql', 'r') as f:

    sql\_script = f.read()

try:

    cursor.execute(sql\_script)

except Exception as e:

    print(f"Error executing command: {sql\_script}", e)

print('Landing Successfully')

conn.commit()

conn.close()

import pyodbc

import pandas as pd

conn = pyodbc.connect('DRIVER={SQL Server};'

                      'SERVER=XWCS50Q04\SQLEXPRESS;'

                      'DATABASE=northwind;'

                      'Trusted\_Connection=yes')

cursor = conn.cursor()

sqlfile = 'SELECT \* FROM landing.lnd\_categories'

df = pd.read\_sql\_query(sqlfile, conn)

print(df)

conn.close()

**Staging:**

import pyodbc

conn = pyodbc.connect('DRIVER={SQL Server};'

                      'SERVER=XWCS50Q04\SQLEXPRESS;'

                      'DATABASE=nw;'

                      'Trusted\_Connection=yes')

cursor = conn.cursor()

with open('C:/Users/LD184YA/OneDrive - EY/Documents/SQL Server Management Studio/Staging\_Create.sql', 'r') as f:

    sqlstg = f.read()

try:

    cursor.execute(sqlstg)

except Exception as e:

    print("Error executing SQL command:", sqlstg)

    print("Error message:", e)

print("Staging tables created Succesfully")

conn.commit()

conn.close()

import pyodbc

conn = pyodbc.connect('DRIVER={SQL Server};'

                      'SERVER=XWCS50Q04\SQLEXPRESS;'

                      'DATABASE=nw;'

                      'Trusted\_Connection=yes')

cursor = conn.cursor()

with open('C:/Users/LD184YA/OneDrive - EY/Documents/SQL Server Management Studio/Staging.sql', 'r') as f:

    sqlfile = f.read()

try:

    cursor.execute(sqlfile)

except Exception as e:

    print("Error executing SQL command:", sqlfile)

    print("Error message:", e)

print("Staging Merged Succesfully")

conn.commit()

conn.close()

**Dimension:**

import pyodbc

conn = pyodbc.connect('DRIVER={SQL Server};'

                      'SERVER=XWCS50Q04\SQLEXPRESS;'

                      'DATABASE=nw;'

                      'Trusted\_Connection=yes')

cursor = conn.cursor()

with open('C:/Users/LD184YA/OneDrive - EY/Documents/SQL Server Management Studio/Dim\_Create.sql', 'r') as l:

    sqldim = l.read()

try:

    cursor.execute(sqldim)

except Exception as e:

    print("Error executing SQL command:", sqldim)

    print("Error message:", e)

print("Dimension tables created Succesfully")

conn.commit()

conn.close()

import pyodbc

conn = pyodbc.connect('DRIVER={SQL Server};'

                      'SERVER=XWCS50Q04\SQLEXPRESS;'

                      'DATABASE=nw;'

                      'Trusted\_Connection=yes')

cursor = conn.cursor()

with open('C:/Users/LD184YA/OneDrive - EY/Documents/SQL Server Management Studio/Dim.sql', 'r') as f:

    sqlfile = f.read()

try:

    cursor.execute(sqlfile)

except Exception as e:

    print("Error executing SQL command:", sqlfile)

    print("Error message:", e)

print("Dimensions Merged Succesfully")

conn.commit()

conn.close()

**Test Operations:**

import pyodbc

import pandas as pd

def function1():

    conn = pyodbc.connect('DRIVER={SQL Server};'

                      'SERVER=XWCS50Q04\SQLEXPRESS;'

                      'DATABASE=northwind;'

                      'Trusted\_Connection=yes')

    cursor = conn.cursor()

    sqlfile = 'select \* from DW.Products\_Dim where Discontinued=1 and EndDate is NULL;'

    df = pd.read\_sql\_query(sqlfile, conn)

    print(df)

    conn.close()

def function2():

    conn = pyodbc.connect('DRIVER={SQL Server};'

                      'SERVER=XWCS50Q04\SQLEXPRESS;'

                      'DATABASE=northwind;'

                      'Trusted\_Connection=yes')

    cursor = conn.cursor()

    sqlfile = 'SELECT Distinct P.ProductKey,P.ProductID,P.ProductName,P.UnitPrice,PF.UnitsOnOrder,C.CategoryID from DW.Products\_Dim P inner join DW.ProductInStock\_fact PF on P.ProductKey=PF.ProductKey inner join DW.Categories\_Dim as C on C.CategoriesKey=PF.CategoriesKey inner join (select C.CategoryID,Max(UnitsOnOrder) as BestSellingProduct from DW.Categories\_Dim as C inner join DW.ProductInStock\_fact as PF on C.CategoriesKey=PF.CategoriesKey Group BY C.CategoryID)as s on s.CategoryID=C.CategoryID where s.BestSellingProduct=PF.UnitsOnOrder and P.EndDate is NULL order by C.CategoryID;'

    df = pd.read\_sql\_query(sqlfile, conn)

    print(df)

    sqlfile1 = 'SELECT Distinct P.ProductKey,P.ProductID,P.ProductName,P.UnitPrice,PF.UnitsOnOrder,C.CategoryID from DW.Products\_Dim P inner join DW.ProductInStock\_fact PF on P.ProductKey=PF.ProductKey inner join DW.Categories\_Dim as C on C.CategoriesKey=PF.CategoriesKey inner join (select C.CategoryID,MIN(UnitsOnOrder) as BestSellingProduct from DW.Categories\_Dim as C inner join DW.ProductInStock\_fact as PF on C.CategoriesKey=PF.CategoriesKey Group BY C.CategoryID)as s on s.CategoryID=C.CategoryID where s.BestSellingProduct=PF.UnitsOnOrder and P.EndDate is NULL order by C.CategoryID;'

    df1 = pd.read\_sql\_query(sqlfile1, conn)

    print(df1)

    conn.close()

def function3():

    conn = pyodbc.connect('DRIVER={SQL Server};'

                      'SERVER=XWCS50Q04\SQLEXPRESS;'

                      'DATABASE=northwind;'

                      'Trusted\_Connection=yes')

    cursor = conn.cursor()

    sqlfile = 'select distinct C1.CustomerKey, MAX(t.sales) over(partition by C1.CustomerKey) MaxCustomerSales,MIN (t.sales) over(partition by C1.CustomerKey) MinCustomerSales,AVG(t.sales) over(partition by C1.CustomerKey) AvgCustomerSales from DW.Customers\_Dim as C1 inner join (select CF.OrderID,CF.CustomerKey,Sum(Cf.Sales) as Sales from DW.Customers\_Dim C inner join Dw.CustomerEmployee\_Fact CF on C.CustomerKey=Cf.CustomerKey group by CF.OrderID,CF.CustomerKey) as t on C1.CustomerKey=t.CustomerKey;'

    df = pd.read\_sql\_query(sqlfile, conn)

    print(df)

    conn.close()

def function4():

    conn = pyodbc.connect('DRIVER={SQL Server};'

                      'SERVER=XWCS50Q04\SQLEXPRESS;'

                      'DATABASE=northwind;'

                      'Trusted\_Connection=yes')

    cursor = conn.cursor()

    sqlfile = 'select \* from Dw.Employees\_Dim where EmployeeID=(select E.EmployeeID from DW.Employees\_Dim E inner join DW.CustomerEmployee\_fact CF on E.EmployeeKey=CF.EmployeeKey group by EmployeeID having sum(CF.sales) in (select max(s.su) from (Select E.EmployeeID,Sum(CF.sales)as su from DW.Employees\_Dim E inner join DW.CustomerEmployee\_fact CF on E.EmployeeKey=CF.EmployeeKey group by E.EmployeeID) s)) ;'

    df = pd.read\_sql\_query(sqlfile, conn)

    print(df)

    conn.close()

def default():

    print('Invalid Input')

switcher = {

    1: function1,

    2: function2,

    3: function3,

    4: function4

}

def switch(case):

    switcher.get(case, default)()

n=int(input(('\n Enter the operation:\n1.Discontinued\_product\_report\n2.Best and least selling product in each category\n3.Min max avg customer billing\n4.Best Salesperson\n')))

switch(n)