**1. Create Database and Use Statement:**

USE InsigniaDB;

**2. Create Lineage Table:**

CREATE TABLE Lineage (

Lineage\_Id BIGINT IDENTITY(1,1) PRIMARY KEY,

Source\_System VARCHAR(100),

Load\_Start\_Datetime DATETIME,

Load\_End\_Datetime DATETIME,

Rows\_at\_Source INT,

Rows\_at\_Destination\_Fact INT,

Load\_Status BIT

);

**3. Create Date Dimension Table:**

CREATE TABLE DimDate (

DateKey INT PRIMARY KEY,

Date DATE,

Day\_Number INT,

Month\_Name VARCHAR(20),

Short\_Month CHAR(3),

Calendar\_Month\_Number INT,

Calendar\_Year INT,

Fiscal\_Month\_Number INT,

Fiscal\_Year INT,

Week\_Number INT

);

**4. Create Product Dimension Table**

CREATE TABLE DimProduct (

ProductKey INT IDENTITY(1,1) PRIMARY KEY,

ProductID INT,

ProductName VARCHAR(255),

ProductCategory VARCHAR(255),

ProductSubcategory VARCHAR(255),

Load\_Id BIGINT

);

**5. Create Customer Dimension Table (SCD Type 2):**

CREATE TABLE DimCustomer (

CustomerKey INT IDENTITY(1,1) PRIMARY KEY,

CustomerID INT,

CustomerName VARCHAR(255),

CustomerEmail VARCHAR(255),

CustomerPhone VARCHAR(50),

CustomerAddress VARCHAR(255),

StartDate DATETIME,

EndDate DATETIME,

IsCurrent BIT,

Load\_Id BIGINT

);

**6. Create Employee Dimension Table (SCD Type 2):**

CREATE TABLE DimEmployee (

EmployeeKey INT IDENTITY(1,1) PRIMARY KEY,

EmployeeID INT,

EmployeeName VARCHAR(255),

EmployeeRole VARCHAR(255),

EmployeeDepartment VARCHAR(255),

StartDate DATETIME,

EndDate DATETIME,

IsCurrent BIT,

Load\_Id BIGINT

);

**7. Create Geography Dimension Table (SCD Type 3):**

CREATE TABLE DimGeography (

GeographyKey INT IDENTITY(1,1) PRIMARY KEY,

GeographyID INT,

Country VARCHAR(255),

State VARCHAR(255),

City VARCHAR(255),

Population INT,

OldPopulation INT,

Load\_Id BIGINT

);

**8. Create Sales Fact Table:**

CREATE TABLE FactSales (

SalesKey INT IDENTITY(1,1) PRIMARY KEY,

DateKey INT,

ProductKey INT,

CustomerKey INT,

EmployeeKey INT,

GeographyKey INT,

SalesAmount DECIMAL(18,2),

QuantitySold INT,

Load\_Id BIGINT

);

**Loading Date Dimension**

DECLARE @StartDate DATE = '2000-01-01';

DECLARE @EndDate DATE = '2023-12-31';

WITH DateSequence AS (

SELECT @StartDate AS DateValue

UNION ALL

SELECT DATEADD(DAY, 1, DateValue)

FROM DateSequence

WHERE DATEADD(DAY, 1, DateValue) <= @EndDate

)

INSERT INTO DimDate (DateKey, Date, Day\_Number, Month\_Name, Short\_Month, Calendar\_Month\_Number, Calendar\_Year, Fiscal\_Month\_Number, Fiscal\_Year, Week\_Number)

SELECT

CONVERT(INT, FORMAT(DateValue, 'yyyyMMdd')) AS DateKey,

DateValue AS Date,

DATEPART(DAY, DateValue) AS Day\_Number,

DATENAME(MONTH, DateValue) AS Month\_Name,

LEFT(DATENAME(MONTH, DateValue), 3) AS Short\_Month,

DATEPART(MONTH, DateValue) AS Calendar\_Month\_Number,

DATEPART(YEAR, DateValue) AS Calendar\_Year,

CASE

WHEN DATEPART(MONTH, DateValue) >= 7 THEN DATEPART(MONTH, DateValue) - 6

ELSE DATEPART(MONTH, DateValue) + 6

END AS Fiscal\_Month\_Number,

CASE

WHEN DATEPART(MONTH, DateValue) >= 7 THEN DATEPART(YEAR, DateValue)

ELSE DATEPART(YEAR, DateValue) - 1

END AS Fiscal\_Year,

DATEPART(WEEK, DateValue) AS Week\_Number

FROM DateSequence

OPTION (MAXRECURSION 0);

**ETL Stored Procedure**

CREATE PROCEDURE LoadData

AS

BEGIN

-- Variables

DECLARE @Load\_Id BIGINT, @Rows\_at\_Source INT, @Rows\_at\_Destination\_Fact INT;

DECLARE @Load\_Start\_Datetime DATETIME, @Load\_End\_Datetime DATETIME;

SET @Load\_Start\_Datetime = GETDATE();

-- Get the row count of the source

SET @Rows\_at\_Source = (SELECT COUNT(\*) FROM Insignia\_staging);

-- Create a copy of the staging table

TRUNCATE TABLE Insignia\_staging\_copy;

INSERT INTO Insignia\_staging\_copy SELECT \* FROM Insignia\_incremental;

-- Insert data into Product Dimension

INSERT INTO DimProduct (ProductID, ProductName, ProductCategory, ProductSubcategory, Load\_Id)

SELECT DISTINCT ProductID, ProductName, ProductCategory, ProductSubcategory, @Load\_Id

FROM Insignia\_staging\_copy;

-- Insert data into Customer Dimension (SCD Type 2)

INSERT INTO DimCustomer (CustomerID, CustomerName, CustomerEmail, CustomerPhone, CustomerAddress, StartDate, EndDate, IsCurrent, Load\_Id)

SELECT CustomerID, CustomerName, CustomerEmail, CustomerPhone, CustomerAddress, @Load\_Start\_Datetime, '9999-12-31', 1, @Load\_Id

FROM Insignia\_staging\_copy s

LEFT JOIN DimCustomer c ON s.CustomerID = c.CustomerID

WHERE c.CustomerID IS NULL;

-- Update existing customers

UPDATE c

SET EndDate = @Load\_Start\_Datetime, IsCurrent = 0

FROM DimCustomer c

JOIN Insignia\_staging\_copy s ON c.CustomerID = s.CustomerID

WHERE c.IsCurrent = 1 AND (

c.CustomerName <> s.CustomerName OR

c.CustomerEmail <> s.CustomerEmail OR

c.CustomerPhone <> s.CustomerPhone OR

c.CustomerAddress <> s.CustomerAddress

);

-- Insert updated customers

INSERT INTO DimCustomer (CustomerID, CustomerName, CustomerEmail, CustomerPhone, CustomerAddress, StartDate, EndDate, IsCurrent, Load\_Id)

SELECT CustomerID, CustomerName, CustomerEmail, CustomerPhone, CustomerAddress, @Load\_Start\_Datetime, '9999-12-31', 1, @Load\_Id

FROM Insignia\_staging\_copy s

LEFT JOIN DimCustomer c ON s.CustomerID = c.CustomerID

WHERE c.IsCurrent = 0;

-- Insert data into Employee Dimension (SCD Type 2)

INSERT INTO DimEmployee (EmployeeID, EmployeeName, EmployeeRole, EmployeeDepartment, StartDate, EndDate, IsCurrent, Load\_Id)

SELECT EmployeeID, EmployeeName, EmployeeRole, EmployeeDepartment, @Load\_Start\_Datetime, '9999-12-31', 1, @Load\_Id

FROM Insignia\_staging\_copy s

LEFT JOIN DimEmployee e ON s.EmployeeID = e.EmployeeID

WHERE e.EmployeeID IS NULL;

-- Update existing employees

UPDATE e

SET EndDate = @Load\_Start\_Datetime, IsCurrent = 0

FROM DimEmployee e

JOIN Insignia\_staging\_copy s ON e.EmployeeID = s.EmployeeID

WHERE e.IsCurrent = 1 AND (

e.EmployeeName <> s.EmployeeName OR

e.EmployeeRole <> s.EmployeeRole OR

e.EmployeeDepartment <> s.EmployeeDepartment

);

-- Insert updated employees

INSERT INTO DimEmployee (EmployeeID, EmployeeName, EmployeeRole, EmployeeDepartment, StartDate, EndDate, IsCurrent, Load\_Id)

SELECT EmployeeID, EmployeeName, EmployeeRole, EmployeeDepartment, @Load\_Start\_Datetime, '9999-12-31', 1, @Load\_Id

FROM Insignia\_staging\_copy s

LEFT JOIN DimEmployee e ON s.EmployeeID = e.EmployeeID

WHERE e.IsCurrent = 0;

-- Insert data into Geography Dimension (SCD Type 3)

INSERT INTO DimGeography (GeographyID, Country, State, City, Population, OldPopulation, Load\_Id)

SELECT GeographyID, Country, State, City, Population, NULL, @Load\_Id

FROM Insignia\_staging\_copy s

LEFT JOIN DimGeography g ON s.GeographyID = g.GeographyID

WHERE g.GeographyID IS NULL;

-- Update existing geography data

UPDATE g

SET OldPopulation = g.Population, Population = s.Population

FROM DimGeography g

JOIN Insignia\_staging\_copy s ON g.GeographyID = s.GeographyID

WHERE g.Population <> s.Population;

-- Insert data into FactSales

INSERT INTO FactSales (DateKey, ProductKey, CustomerKey, EmployeeKey, GeographyKey, SalesAmount, QuantitySold, Load\_Id)

SELECT

d.DateKey,

p.ProductKey,

c.CustomerKey,

e.EmployeeKey,

g.GeographyKey,

s.SalesAmount,

s.QuantitySold,

@Load\_Id

FROM Insignia\_staging\_copy s

JOIN DimDate d ON s.SaleDate = d.Date

JOIN DimProduct p ON s.ProductID = p.ProductID

JOIN DimCustomer c ON s.CustomerID = c.CustomerID AND c.IsCurrent = 1

JOIN DimEmployee e ON s.EmployeeID = e.EmployeeID AND e.IsCurrent = 1

JOIN DimGeography g ON s.GeographyID = g.GeographyID;

-- Update lineage information

SET @Rows\_at\_Destination\_Fact = (SELECT COUNT(\*) FROM FactSales WHERE Load\_Id = @Load\_Id);

SET @Load\_End\_Datetime = GETDATE();

INSERT INTO Lineage (Source\_System, Load\_Start\_Datetime, Load\_End\_Datetime, Rows\_at\_Source, Rows\_at\_Destination\_Fact, Load\_Status)

VALUES ('Insignia', @Load\_Start\_Datetime, @Load\_End\_Datetime, @Rows\_at\_Source, @Rows\_at\_Destination\_Fact, 1);

END;

**Reconciliation Module**

CREATE PROCEDURE ReconcileData

AS

BEGIN

SELECT

Lineage\_Id,

Source\_System,

Load\_Start\_Datetime,

Load\_End\_Datetime,

Rows\_at\_Source,

Rows\_at\_Destination\_Fact,

Load\_Status

FROM Lineage;

END;