CREATE DATABASE Bikes;

USE Bikes;

--Bridge table linked to Fact table and Product table

CREATE TABLE DimDepartment(

DepartmentID VARCHAR(255) NOT NULL,

DepartmentName VARCHAR(255),

constraint DimDepartment\_PK primary key (DepartmentID));

BULK

INSERT DimDepartment

FROM 'C:\Users\zj6362\Desktop\Project2\Department.txt'

WITH

(

FIRSTROW = 2,

FIELDTERMINATOR = ',',

ROWTERMINATOR = '\n'

)

GO

Select \* from DimDepartment;

CREATE TABLE DimRegion(

RegionID VARCHAR(255) NOT NULL,

RegionName VARCHAR(255),

constraint DimRegion\_PK primary key (RegionID));

BULK

INSERT DimRegion

FROM 'C:\Users\zj6362\Desktop\Project2\Region.txt'

WITH

(

FIRSTROW = 2,

FIELDTERMINATOR = ',',

ROWTERMINATOR = '\n'

)

GO

select \* from DimRegion;

CREATE TABLE DimEmployee(

EmployeeKey int identity(1,1) not null,

EmployeeID VARCHAR(255) NOT NULL,

EmployeeFirstName VARCHAR(255) NOT NULL,

EmployeeLastName VARCHAR(255) NOT NULL,

DepartmentID VARCHAR(255) NOT NULL,

HistoricalDepartmentID VARCHAR(255),

EmployeeAddress VARCHAR(255),

Gender VARCHAR(255),

EmployeeBirthDate VARCHAR(255),

Salary money,

RegionID VARCHAR(255) NOT NULL,

RowEffectiveDate datetime,

RowExpirationDate datetime,

CurrentRowIndicator varchar(255) default 'Current',

CONSTRAINT DimEmploee\_PK PRIMARY KEY (EmployeeKey),

constraint DimEmployee\_FK1 foreign key (DepartmentID) references DimDepartment(DepartmentID),

constraint DimEmployee\_FK2 foreign key (RegionID) references DimRegion(RegionID));

select \* from DimEmployee

order by EmployeeID;

--Mini dimension as an outrigger

create table CurrentMiniDimCustomer

(CurrentDemographicsKey int identity(1,1) not null,

CurrentAgeFrom int,

CurrentAgeTo int,

CurrentYearsOfExperienceFrom int,

CurrentYearsOfExperienceTo int,

constraint CurrentMiniDimCustomer\_PK primary key (CurrentDemographicsKey));

insert into CurrentMiniDimCustomer

(CurrentAgeFrom, CurrentAgeTo, CurrentYearsOfExperienceFrom, CurrentYearsOfExperienceTo)

values

(20, 25, 0, 5),

(26, 30, 0, 5),

(31, 35, 0, 5),

(20, 25, 6, 20),

(26, 30, 6, 20),

(31, 35, 6, 20);

select \* from CurrentMiniDimCustomer;

--delete from CurrentMiniDimCustomer;

--DBCC CHECKIDENT ('CurrentMiniDimCustomer', RESEED, 0);

--CustFirstName - type 0, CustLastName - type 1, RepeatPurchaser - type 2

CREATE TABLE DimCustomer(

CustomerKey int identity(1,1) not null,

CustomerID VARCHAR(255) NOT NULL,

CustomerFirstName VARCHAR(255) NOT NULL,

CustomerLastName VARCHAR(255) NOT NULL,

CustomerLocation VARCHAR(255),

CustomerAge int,

YearsOfExperience int,

RowEffectiveDate datetime,

RowExpirationDate datetime,

CurrentRowIndicator varchar(255) default 'Current',

CurrentDemographicsKey int,

CONSTRAINT DimCustomer\_PK PRIMARY KEY (CustomerKey),

constraint DimCustomer\_FK foreign key (CurrentDemographicsKey) references CurrentMiniDimCustomer(CurrentDemographicsKey));

select \* from DimCustomer

order by CustomerID;

--ProdQOH and ProdNextShipDate are type 3 SCD

CREATE TABLE DimProduct (

ProductKey int identity(1,1) not null,

ProductID VARCHAR(255) NOT NULL,

ProductName VARCHAR(255) NOT NULL,

Cost money,

WholeSalePrice money,

PreviousWholeSalePrice money,

MSRP money,

EffectivePriceDate datetime,

PriceChangePercentage float,

CONSTRAINT DimProduct\_PK PRIMARY KEY (ProductKey));

select \* from DimProduct

order by ProductID;

--delete from DimProduct;

--DBCC CHECKIDENT ('DimProduct', RESEED, 0);

--EmpFirstName - type 0, EmpLastName - type 1, CurrentSupEmpNo - type 6

--delete from DimEmployee;

--DBCC CHECKIDENT ('DimEmployee', RESEED, 0);

--Mini dimension linked to the Fact table

create table MiniDimCustomer

(DemographicsKey int identity(1,1) not null,

AgeFrom int,

AgeTo int,

YearsOfExperienceFrom int,

YearsOfExperienceeTo int,

constraint MiniDimCustomer\_PK primary key (DemographicsKey));

insert into MiniDimCustomer

(AgeFrom, AgeTo, YearsOfExperienceFrom, YearsOfExperienceeTo)

values

(20, 25, 0, 5),

(26, 30, 0, 5),

(31, 35, 0, 5),

(20, 25, 6, 20),

(26, 30, 6, 20),

(31, 35, 6, 20);

select \* from MiniDimCustomer;

--Fact table references to CustomerKey, EmployeeKey, and DemographicsKey

CREATE TABLE SalesOrder(

OrderID VARCHAR(255) NOT NULL,

PODate datetime,

CustomerPO VARCHAR(255) NOT NULL,

Quantity int,

UnitPrice float,

CustomerKey int NOT NULL,

EmployeeKey int NOT NULL,

ProductKey int NOT NULL,

DemographicsKey int NOT NULL,

constraint SalesOrder\_PK primary key (OrderID),

constraint SalesOrder\_FK1 foreign key (CustomerKey) references DimCustomer(CustomerKey),

constraint SalesOrder\_FK2 foreign key (EmployeeKey) references DimEmployee(EmployeeKey),

constraint SalesOrder\_FK3 foreign key (ProductKey) references DimProduct(ProductKey),

constraint SalesOrder\_FK4 foreign key (DemographicsKey) references MiniDimCustomer(DemographicsKey));

select \* from SalesOrder;

--Script to add values to CurrentDemographicsKey.

if (Row.CopyofCustomerAge <= 25 && Row.CopyofCustomerAge >= 20)

{

switch (Row.CopyofYearsOfExperience)

{

case int n when (n >= 0 && n <= 5):

Row.CurrentDemographicsKey = 1;

break;

case int n when (n >= 6 && n <= 20):

Row.CurrentDemographicsKey = 4;

break;

}

}

if (Row.CopyofCustomerAge <= 30 && Row.CopyofCustomerAge >= 26)

{

switch (Row.CopyofYearsOfExperience)

{

case int n when (n >= 0 && n <= 5):

Row.CurrentDemographicsKey = 1;

break;

case int n when (n >= 6 && n <= 20):

Row.CurrentDemographicsKey = 5;

break;

}

}

if (Row.CopyofCustomerAge <= 35 && Row.CopyofCustomerAge >= 31)

{

switch (Row.CopyofYearsOfExperience)

{

case int n when (n >= 0 && n <= 5):

Row.CurrentDemographicsKey = 1;

break;

case int n when (n >= 6 && n <= 20):

Row.CurrentDemographicsKey = 6;

break;

}

}

select \* from [dbo].[DimCustomer]

where [CurrentRowIndicator] = 'Current';

select \* from [dbo].[DimEmployee]

where [CurrentRowIndicator] = 'Current';

update [dbo].[DimEmployee]

set[DepartmentID]=?

where [EmployeeID]=?;

update [dbo].[DimProduct]

set [PreviousWholeSalePrice] = [WholeSalePrice],

[WholeSalePrice] =?

where [ProductID]=?;

update [dbo].[DimProduct]

set [PriceChangePercentage] = ([WholeSalePrice] - [PreviousWholeSalePrice])/[PreviousWholeSalePrice]\*100,

[EffectivePriceDate] = GETDATE()

where [PreviousWholeSalePrice] is not null;