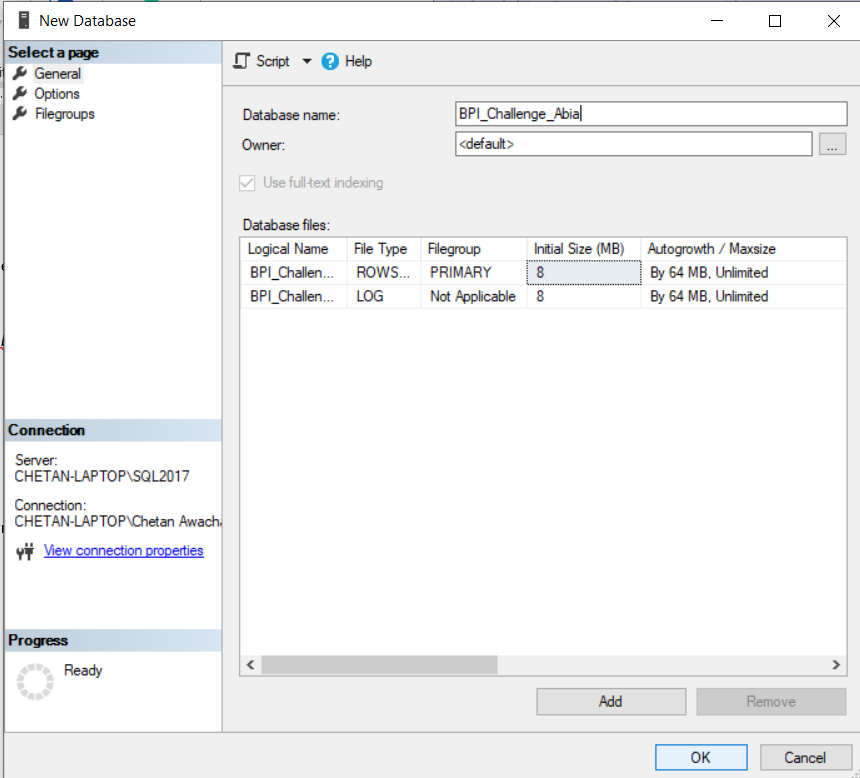
*In Object Explorer 🡪 (right click on New Database)* *🡪* Database Name: BPI\_Challenge\_Abia

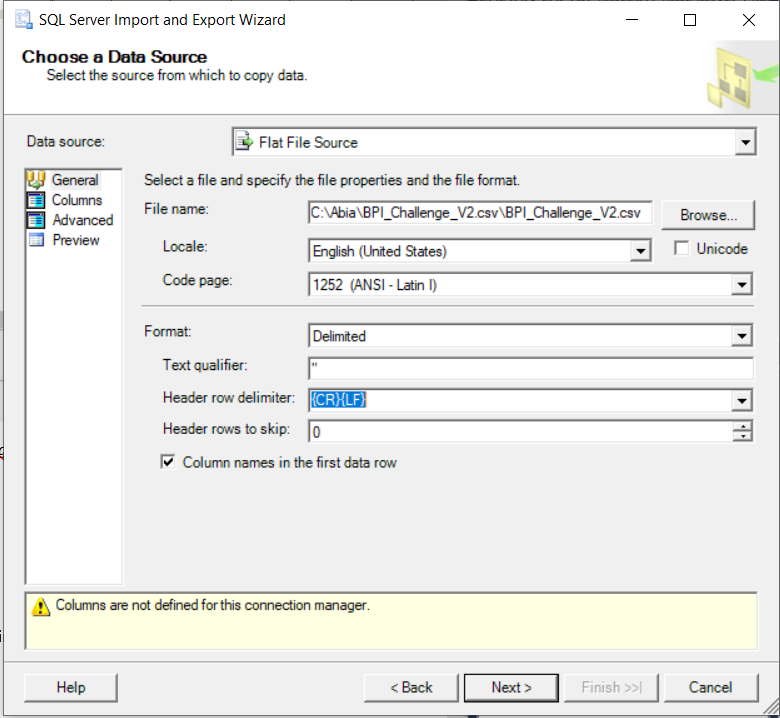


Hit OK

*In Object Explorer 🡪 (right click on Database BPI\_Challenge\_Abia) 🡪 Tasks 🡪 Import Data*

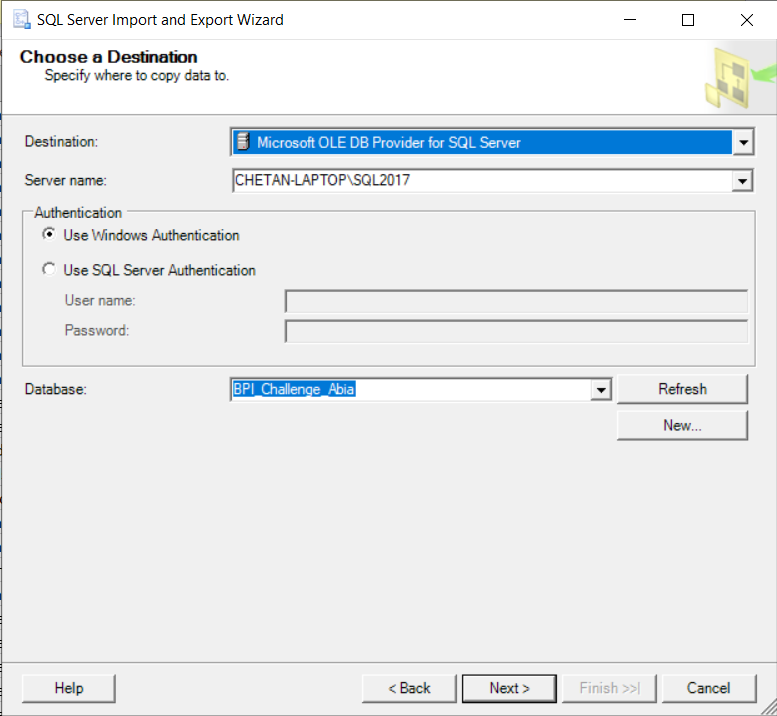
Select *Data Source: Flat File Source* and Select the Data file (“BPI\_Challenge\_V2.csv”) in *File name*

**Note:** Type **“** in Text Qualifier (If not done it would create incorrect columns as value in one of the columns is “3-way match, invoice before GR”)



Hit Next>

Select the “*Destination:”* and give Your Database a name



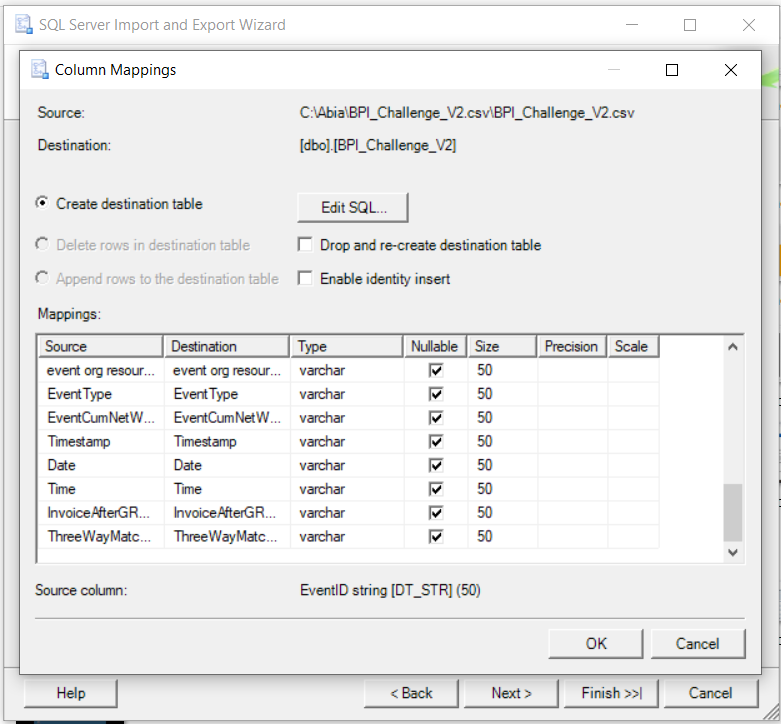
Hit Next>

In Source Tables and Views Select the Table that was created and hit *Edit Mappings…*

Here we observe that DataType of all the columns is varchar.

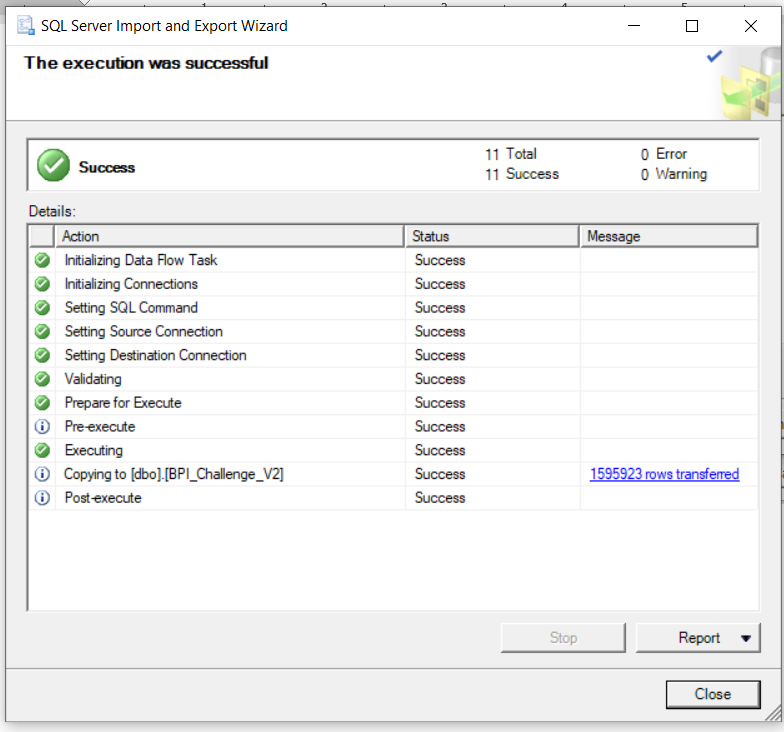
The DataType of the columns would be handled later using query.

(**Note:** If we try to change the DataType of the columns It throws an error while importing the data)



Hit Ok 🡪 Hit Next>

Select *Run Immediately 🡪* Hit Next> and Finish



We have successfully Imported the data

View the table In *Object Explorer 🡪 Database 🡪 BPI\_Challenge\_Abia 🡪 Tables 🡪 BPI\_Challenge\_V2*

Now, before moving to the Dimensions and fact table we will change the DataType of the columns and Handle the *Null* values of the Data.

In *Object Explorer 🡪 Database 🡪 right click on BPI\_Challenge\_Abia 🡪 New Query*

Execute the following query to change the Null values to ‘NONE’ –

UPDATE BPI\_Challenge\_V2

SET ItemArea = 'NONE'

WHERE ItemArea = '';

UPDATE BPI\_Challenge\_V2

SET ItemAreaDesc = 'NONE'

WHERE ItemAreaDesc = '';

UPDATE BPI\_Challenge\_V2

SET ItemClass = 'NONE'

WHERE ItemClass = '';

Execute the following query to change the DataType of the columns –

ALTER TABLE BPI\_Challenge\_V2

ALTER COLUMN InvoiceAfterGRValue bit NOT NULL;

ALTER TABLE BPI\_Challenge\_V2

ALTER COLUMN ThreeWayMatchValue bit NOT NULL;

ALTER TABLE BPI\_Challenge\_V2

ALTER COLUMN EventID bigint NOT NULL;

ALTER TABLE BPI\_Challenge\_V2

ALTER COLUMN PODocID bigint NOT NULL;

ALTER TABLE BPI\_Challenge\_V2

ALTER COLUMN EventCumNetWorth float NOT NULL;

ALTER TABLE BPI\_Challenge\_V2

ALTER COLUMN EventCumNetWorth int NOT NULL;

ALTER TABLE BPI\_Challenge\_V2

ALTER COLUMN ItemID Int NOT NULL;

ALTER TABLE BPI\_Challenge\_V2

ALTER COLUMN Date Date NOT NULL;

ALTER TABLE BPI\_Challenge\_V2

ALTER COLUMN Time time NOT NULL;

Using our main table BPI\_Challenge\_V2 we create the Dimensions Using the following Query-

**Note:** Alter Table query is used to introduce the Auto-Incremental Values in the respective table which will also act as a primary key for that table. In ThreeWayMatch and InvoiceAfterGR table we use the bit values directly as primary key.

Select distinct EventUser

into Dim\_EventUser

from BPI\_Challenge\_V2

WHERE EventUser IS NOT NULL;

Alter table Dim\_EventUser

ADD EventUser\_ID Int IDENTITY(1,1) PRIMARY KEY;

Select distinct Purchaser

into Dim\_Purchaser

from BPI\_Challenge\_V2

WHERE Purchaser IS NOT NULL;

Alter table Dim\_Purchaser

ADD Purchaser\_ID Int IDENTITY(1,1) PRIMARY KEY;

Select distinct ThreeWayMatch, ThreeWayMatchValue

into Dim\_ThreeWayMatch

from BPI\_Challenge\_V2

WHERE ThreeWayMatch IS NOT NULL AND ThreeWayMatchValue IS NOT NULL;

Alter table Dim\_ThreeWayMatch

ADD CONSTRAINT PK\_ThreeWayMatch PRIMARY KEY (ThreeWayMatchValue);

Select distinct ItemID, ItemType, ItemPOProcess

into Dim\_Item

from BPI\_Challenge\_V2

WHERE (ItemID IS NOT NULL) AND (ItemType IS NOT NULL) AND (ItemPOProcess IS NOT NULL);

Alter table Dim\_Item

ADD ItemNo Int IDENTITY(1,1) PRIMARY KEY;

Select distinct EventType

into Dim\_EventType

from BPI\_Challenge\_V2

WHERE EventType IS NOT NULL;

Alter table Dim\_EventType

ADD EventType\_ID Int IDENTITY(1,1) PRIMARY KEY;

Select distinct InvoiceAfterGR, InvoiceAfterGRValue

into Dim\_InvoiceAfterGR

from BPI\_Challenge\_V2

WHERE (InvoiceAfterGRValue IS NOT NULL) AND (InvoiceAfterGRValue IS NOT NULL);

Alter table Dim\_InvoiceAfterGR

ADD CONSTRAINT PK\_InvoiceAfterGR PRIMARY KEY (InvoiceAfterGRValue);

Select distinct Vendor, VendorReceivedPO

into Dim\_Vendor

from BPI\_Challenge\_V2

WHERE (Vendor IS NOT NULL) AND (VendorReceivedPO IS NOT NULL);

Alter table Dim\_Vendor

ADD Vendor\_ID Int IDENTITY(1,1) PRIMARY KEY;

Select distinct POType, PODocID

into Dim\_PurchaseOrder

from BPI\_Challenge\_V2

WHERE (PODocID IS NOT NULL) AND (POType IS NOT NULL);

Alter table Dim\_PurchaseOrder

ADD PO\_ID Int IDENTITY(1,1) PRIMARY KEY;

Select distinct ItemArea, ItemAreaDesc, ItemClass

into Dim\_Spend

from BPI\_Challenge\_V2

WHERE ItemArea IS NOT NULL AND ItemAreaDesc IS NOT NULL AND ItemClass IS NOT NULL;

Alter table Dim\_Spend

ADD Spend\_ID Int IDENTITY(1,1) PRIMARY KEY;

**Query for Date Dimension –**

CREATE TABLE [dbo].[Dim\_Date](

[DateKey] [int] NULL,

[Date] [date] NULL,

[Year No] [int] NULL,

[Quarter Year No] [varchar](100) NULL,

[Quarter Year Desc] [varchar](100) NULL,

[Month Year No] [int] NULL,

[Month Year Desc] [varchar](100) NULL,

[Week Year No] [varchar](100) NULL,

[Week Year Desc] [varchar](100) NULL,

[Day of Month] [int] NULL,

[Day Name Abbr] [varchar](100) NULL,

[Day Name] [varchar](100) NULL,

[Week Details] [varchar](100) NULL,

[Day Name Order] [int] NULL

)

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

create Procedure [dbo].[DimdateSP]

@StartDate Date , ------------------------MM-DD-YYYY

@EndDate Date ----------------------------MM-DD-YYYY

--Example: Data from 01-jan-2014 to 31-dec-2014

--@StartDate='01-01-2014' and @EndDate='12-31-2014'

As

DECLARE @CurrDate DateTime

SET @CurrDate= @StartDate

DECLARE @FinalYear Datetime

SET @FinalYear = @EndDate

DECLARE @DateKey int

DECLARE @years VARCHAR(100)

DECLARE @QuarterNo VARCHAR(100)

DECLARE @QuarterDesc VARCHAR(100)

DECLARE @MonthNo INT

DECLARE @MonthDesc VARCHAR(100)

DECLARE @WeekNo VARCHAR(100)

DECLARE @WeekDesc VARCHAR(100)

DECLARE @FirstdayofWeek VARCHAR(100)

DECLARE @LastdayofWeek VARCHAR(100)

DECLARE @MonthDay INT

DECLARE @WeekDay VARCHAR(100)

DECLARE @WeekDetails VARCHAR(100)

DECLARE @weekfirstdate Date DECLARE @weekLastdate Date

declare @dim table(

[DateKey] [int] NULL,

[Date] [datetime] NULL,

[Year No] [int] NULL,

[Quarter Year No] [varchar](100) NULL,

[Quarter Year Desc] [varchar](100) NULL,

[Month Year No] [int] NULL,

[Month Year Desc] [varchar](100) NULL,

[Week Year No] [varchar](100) NULL,

[Week Year Desc] [varchar](100) NULL,

[Day of Month] [int] NULL,

[Day Name] [varchar](100) NULL,

[Week Details] [varchar](100) NULL,

[Day Name Order] [int] NULL,

[Week First Date] Date null,

[Week Last Date] Date null

)

SET @DateKey=CONVERT(varchar, @CurrDate, 112)

set @years =DATEPART(YYYY,@CurrDate)

SET @QuarterNo=DATEPART(Q,@CurrDate)

SET @MonthNo=CONVERT(varchar(6), @CurrDate, 112)

SET @MonthDesc=cast((DATENAME(month,@CurrDate))as varchar(3))+'-'+RIGHT(DATENAME(YEAR, @CurrDate),4)

SET @WeekNo=DatePart(week, @CurrDate)

SET @FirstdayofWeek=DATEPART(D,DATEADD(wk, DATEDIFF(wk, 7, @CurrDate), 6))

SET @LastdayofWeek=DATEPART(D,DATEADD(wk, DATEDIFF(wk, 6, @CurrDate), 6 + 6))

SET @WeekDesc = cast((DATENAME(month,@CurrDate))as varchar(3))+'-'+RIGHT(DATENAME(YEAR, @CurrDate),4)

SET @MonthDay =DATEPART(d,@CurrDate)

SET @WeekDay = cast(datename(DW,DATEPART(DW,@CurrDate-2))as varchar(3))

SET @weekfirstdate=convert(date,dateadd(week, datediff(week, 0, @CurrDate), -1))

SET @weekLastdate=convert(date,DATEADD(wk, DATEDIFF(wk, 6, @CurrDate), 6 + 6))

WHILE YEAR(@CurrDate)<= YEAR(@FinalYear)

BEGIN

IF((@WeekDay='Sat') or(@WeekDay='Sun'))

SET @WeekDetails='Week Ends'

ELSE

SET @WeekDetails='Week Days'

If DATEPART(Q,@CurrDate) = 1

begin

set @QuarterDesc='Jan-Mar'+' '+RIGHT(DATENAME(YEAR, @CurrDate),4)

end

If DATEPART(Q,@CurrDate) = 2

begin

set @QuarterDesc='Apr-Jun'+' '+RIGHT(DATENAME(YEAR, @CurrDate),4)

end

If DATEPART(Q,@CurrDate) = 3

begin

set @QuarterDesc='Jul-Sep' +' '+RIGHT(DATENAME(YEAR, @CurrDate),4)

end

If DATEPART(Q,@CurrDate) = 4

begin

set @QuarterDesc='Oct-Dec'+' '+RIGHT(DATENAME(YEAR, @CurrDate),4)

end

IF MONTH(@weekfirstdate)>MONTH(@weekLastdate) and Year(@weekfirstdate)=Year(@CurrDate)

BEGIN

if day(@weekfirstdate)>day(@weekLastdate)

begin

set @LastdayofWeek=datepart(d,CONVERT(date,DATEADD(dd,-(DAY(DATEADD(mm,1,@weekfirstdate))),DATEADD(mm,1,@weekfirstdate)),101))

end

else

set @FirstdayofWeek=datepart(d,CONVERT(date,DATEADD(dd,-(DAY(@weekLastdate)-1),@weekLastdate),101))

END

else if MONTH(@weekfirstdate)>MONTH(@weekLastdate) and Year(@weekfirstdate)<Year(@CurrDate)

BEGIN

if day(@weekfirstdate)>day(@weekLastdate)

begin

set @FirstdayofWeek=datepart(d,CONVERT(date,DATEADD(dd,-(DAY(@weekLastdate)-1),@weekLastdate),101))

end

END

else if Year(@weekfirstdate)=Year(@weekLastdate) and Month(@weekfirstdate)<Month(@weekLastdate) and Month(@CurrDate)=month(@weekLastdate)

begin

set @FirstdayofWeek=datepart(d,CONVERT(date,DATEADD(dd,-(DAY(@weekLastdate)-1),@weekLastdate),101))

end

else if Year(@weekfirstdate)=Year(@weekLastdate) and Month(@weekfirstdate)<Month(@weekLastdate) and Month(@CurrDate)<month(@weekLastdate)

begin

set @LastdayofWeek=datepart(d,CONVERT(date,DATEADD(dd,-(DAY(DATEADD(mm,1,@weekfirstdate))),DATEADD(mm,1,@weekfirstdate)),101))

end

IF (@CurrDate>@FinalYear)

BEGIN

BREAK

END

INSERT into @dim

Values(@DateKey,@CurrDate,@years,@years+''+@QuarterNo,@QuarterDesc,@MonthNo,@MonthDesc,@years+''+@WeekNo,

'WK ('+@FirstdayofWeek+'-'+@LastdayofWeek+')/'+@WeekDesc,@MonthDay,@WeekDay,@WeekDetails,'',@weekfirstdate,@weekLastdate)

SET @CurrDate = DATEADD(d,1,@CurrDate)

SET @DateKey=CONVERT(varchar, @CurrDate, 112)

set @years =DATEPART(YYYY,@CurrDate)

SET @QuarterNo=DATEPART(Q,@CurrDate)

SET @MonthNo=CONVERT(varchar(6), @CurrDate, 112)

SET @MonthDesc=cast((DATENAME(month,@CurrDate))as varchar(3))+'-'+RIGHT(DATENAME(YEAR, @CurrDate),4)

SET @WeekNo=DatePart(week, @CurrDate)

SET @FirstdayofWeek=DATEPART(D,DATEADD(wk, DATEDIFF(wk, 7, @CurrDate), 6))

SET @LastdayofWeek=DATEPART(D,DATEADD(wk, DATEDIFF(wk, 6, @CurrDate), 6 + 6))

SET @WeekDesc = cast((DATENAME(month,@CurrDate))as varchar(3))+'-'+RIGHT(DATENAME(YEAR, @CurrDate),4)

SET @MonthDay =DATEPART(d,@CurrDate)

SET @WeekDay = cast(datename(DW,DATEPART(DW,@CurrDate-2))as varchar(3))

SET @weekfirstdate=convert(date,dateadd(week, datediff(week, 0, @CurrDate), -1))

SET @weekLastdate=convert(date,DATEADD(wk, DATEDIFF(wk, 6, @CurrDate), 6 + 6))

IF MONTH(@weekfirstdate)>MONTH(@weekLastdate) and Year(@weekfirstdate)=Year(@CurrDate)

BEGIN

if day(@weekfirstdate)>day(@weekLastdate)

begin

set @LastdayofWeek=datepart(d,CONVERT(date,DATEADD(dd,-(DAY(DATEADD(mm,1,@weekfirstdate))),DATEADD(mm,1,@weekfirstdate)),101))

end

else

set @FirstdayofWeek=datepart(d,CONVERT(date,DATEADD(dd,-(DAY(@weekLastdate)-1),@weekLastdate),101))

END

else if MONTH(@weekfirstdate)>MONTH(@weekLastdate) and Year(@weekfirstdate)<Year(@CurrDate)

BEGIN

if day(@weekfirstdate)>day(@weekLastdate)

begin

set @FirstdayofWeek=datepart(d,CONVERT(date,DATEADD(dd,-(DAY(@weekLastdate)-1),@weekLastdate),101))

end

END

else if Year(@weekfirstdate)=Year(@weekLastdate) and Month(@weekfirstdate)<Month(@weekLastdate) and Month(@CurrDate)=month(@weekLastdate)

begin

set @FirstdayofWeek=datepart(d,CONVERT(date,DATEADD(dd,-(DAY(@weekLastdate)-1),@weekLastdate),101))

end

else if Year(@weekfirstdate)=Year(@weekLastdate) and Month(@weekfirstdate)<Month(@weekLastdate) and Month(@CurrDate)<month(@weekLastdate)

begin

set @LastdayofWeek=datepart(d,CONVERT(date,DATEADD(dd,-(DAY(DATEADD(mm,1,@weekfirstdate))),DATEADD(mm,1,@weekfirstdate)),101))

end

UPDATE @dim SET [Day Name Order]=1 WHERE [Day Name]='Mon'

UPDATE @dim SET [Day Name Order]=2 WHERE [Day Name]='Tue'

UPDATE @dim SET [Day Name Order]=3 WHERE [Day Name]='Wed'

UPDATE @dim SET [Day Name Order]=4 WHERE [Day Name]='Thu'

UPDATE @dim SET [Day Name Order]=5 WHERE [Day Name]='Fri'

UPDATE @dim SET [Day Name Order]=6 WHERE [Day Name]='Sat'

UPDATE @dim SET [Day Name Order]=7 WHERE [Day Name]='Sun'

END

INSERT INTO Dim\_Date

([DateKey],[Date],[Year No],[Quarter Year No],[Quarter Year Desc],[Month Year No],[Month Year Desc],[Week Year No]

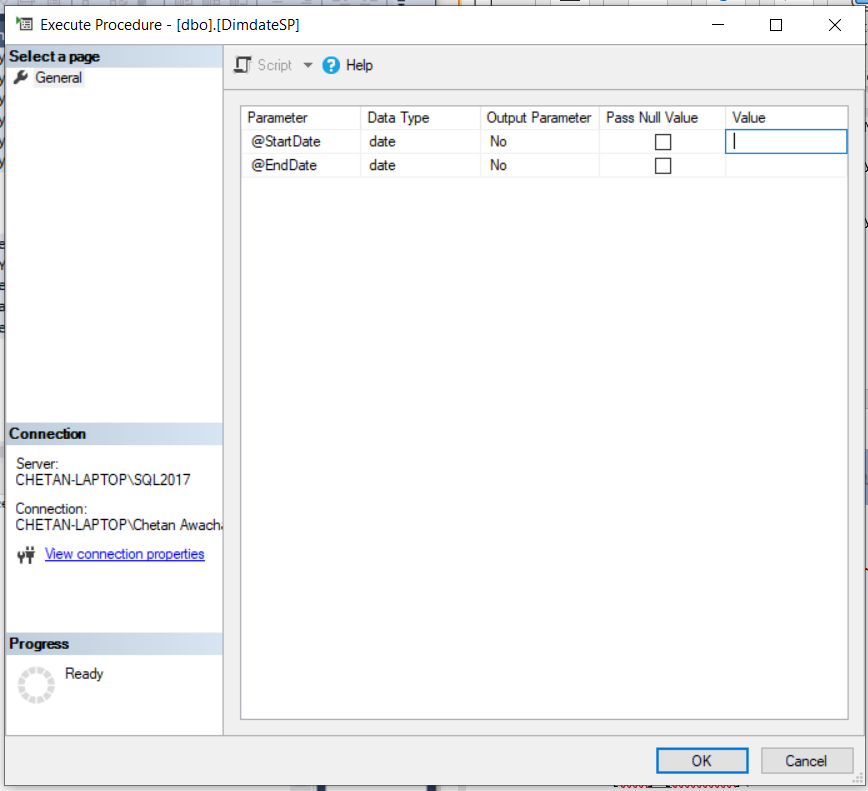
,[Week Year Desc],[Day of Month],[Day Name],[Week Details],[Day Name Order])

select [DateKey],[Date],[Year No],[Quarter Year No],[Quarter Year Desc],[Month Year No],[Month Year Desc],[Week Year No]

,[Week Year Desc],[Day of Month],[Day Name],[Week Details],[Day Name Order] from @dim

Execute the Procedure

In *Object Explorer 🡪 Database 🡪 BPI\_Challenge\_Abia🡪 Programmability🡪 right click on dbo.DimdateSP🡪 Execute Stored Procedure*

**

Enter the required @StartDate and @EndDate in the value column

**(NOTE:** Prefer entering values for just 5 years at a time if you need more that 5 years records in the Date Dimension table)

HIT OK

**Query for Time Dimension –**

-- Create a new table

CREATE TABLE [dbo].[DimTime](

[TimeId] [int] IDENTITY(1,1) NOT NULL Primary Key,

[Time] [time](0) NULL,

[Hour] [int] NULL,

[Minute] [int] NULL,

[MilitaryHour] int NOT null,

[MilitaryMinute] int NOT null,

[AMPM] [varchar](2) NOT NULL,

[DayPartEN] [varchar](10) NULL,

[DayPartNL] [varchar](10) NULL,

[HourFromTo12] [varchar](17) NULL,

[HourFromTo24] [varchar](13) NULL,

[Notation12] [varchar](10) NULL,

[Notation24] [varchar](10) NULL

);

GO

-- Create a time and a counter variable for the loop

DECLARE @Time as time;

SET @Time = '0:00';

DECLARE @counter as int;

SET @counter = 0;

-- Two variables to store the day part for two languages

DECLARE @daypartEN as varchar(20);

set @daypartEN = '';

DECLARE @daypartNL as varchar(20);

SET @daypartNL = '';

-- Loop 1440 times (24hours \* 60minutes)

WHILE @counter < 1440

BEGIN

-- Determine datepart

SELECT @daypartEN = CASE

WHEN (@Time >= '0:00' and @Time < '6:00') THEN 'Night'

WHEN (@Time >= '6:00' and @Time < '12:00') THEN 'Morning'

WHEN (@Time >= '12:00' and @Time < '18:00') THEN 'Afternoon'

ELSE 'Evening'

END

, @daypartNL = CASE

WHEN (@Time >= '0:00' and @Time < '6:00') THEN 'Nacht'

WHEN (@Time >= '6:00' and @Time < '12:00') THEN 'Ochtend'

WHEN (@Time >= '12:00' and @Time < '18:00') THEN 'Middag'

ELSE 'Avond'

END;

INSERT INTO DimTime ([Time]

, [Hour]

, [Minute]

, [MilitaryHour]

, [MilitaryMinute]

, [AMPM]

, [DayPartEN]

, [DayPartNL]

, [HourFromTo12]

, [HourFromTo24]

, [Notation12]

, [Notation24])

VALUES (@Time

, DATEPART(Hour, @Time) + 1

, DATEPART(Minute, @Time) + 1

, DATEPART(Hour, @Time)

, DATEPART(Minute, @Time)

, CASE WHEN (DATEPART(Hour, @Time) < 12) THEN 'AM' ELSE 'PM' END

, @daypartEN

, @daypartNL

, CONVERT(varchar(10), DATEADD(Minute, -DATEPART(Minute,@Time), @Time),100) + ' - ' + CONVERT(varchar(10), DATEADD(Hour, 1, DATEADD(Minute, -DATEPART(Minute,@Time), @Time)),100)

, CAST(DATEADD(Minute, -DATEPART(Minute,@Time), @Time) as varchar(5)) + ' - ' + CAST(DATEADD(Hour, 1, DATEADD(Minute, -DATEPART(Minute,@Time), @Time)) as varchar(5))

, CONVERT(varchar(10), @Time,100)

, CAST(@Time as varchar(5))

);

-- Raise time with one minute

SET @Time = DATEADD(minute, 1, @Time);

-- Raise counter by one

set @counter = @counter + 1;

END

We are done creating all the dimension tables. Now using all the dimension tables and our main table we will create Fact Table using LEFT JOIN.

**Query for Fact Table –**

Select Distinct BPI\_Challenge\_V2.EventID ,

Dim\_PurchaseOrder.PO\_ID,

Dim\_EventType.EventType\_ID,

Dim\_Item.ItemNo ,

Dim\_Vendor.Vendor\_ID ,

Dim\_EventUser.EventUser\_ID ,

Dim\_Purchaser.Purchaser\_ID ,

Dim\_InvoiceAfterGR.InvoiceAfterGRValue,

Dim\_ThreeWayMatch.ThreeWayMatchValue ,

Dim\_Spend.Spend\_ID ,

BPI\_Challenge\_V2.Date ,

DimTime.TimeId ,

BPI\_Challenge\_V2.EventCumNetWorth

INTO Fact\_Event --1595923

FROM BPI\_Challenge\_V2

LEFT JOIN Dim\_EventType ON BPI\_Challenge\_V2.EventType = Dim\_EventType.EventType

LEFT JOIN Dim\_EventUser ON BPI\_Challenge\_V2.EventUser = Dim\_EventUser.EventUser

LEFT JOIN Dim\_InvoiceAfterGR ON BPI\_Challenge\_V2.InvoiceAfterGR = Dim\_InvoiceAfterGR.InvoiceAfterGR AND BPI\_Challenge\_V2.InvoiceAfterGRValue = Dim\_InvoiceAfterGR.InvoiceAfterGRValue

LEFT JOIN Dim\_Item ON BPI\_Challenge\_V2.ItemID = Dim\_Item.ItemID AND BPI\_Challenge\_V2.ItemType = Dim\_Item.ItemType AND BPI\_Challenge\_V2.ItemPOProcess = Dim\_Item.ItemPOProcess

LEFT JOIN Dim\_PurchaseOrder ON BPI\_Challenge\_V2.POType = Dim\_PurchaseOrder.POType AND BPI\_Challenge\_V2.PODocID = Dim\_PurchaseOrder.PODocID

LEFT JOIN Dim\_Purchaser ON BPI\_Challenge\_V2.Purchaser = Dim\_Purchaser.Purchaser

LEFT JOIN Dim\_Spend ON BPI\_Challenge\_V2.ItemArea = Dim\_Spend.ItemArea AND BPI\_Challenge\_V2.ItemAreaDesc = Dim\_Spend.ItemAreaDesc AND BPI\_Challenge\_V2.ItemClass = Dim\_Spend.ItemClass

LEFT JOIN Dim\_ThreeWayMatch ON BPI\_Challenge\_V2.ThreeWayMatch = Dim\_ThreeWayMatch.ThreeWayMatch AND BPI\_Challenge\_V2.ThreeWayMatchValue = Dim\_ThreeWayMatch.ThreeWayMatchValue

LEFT JOIN Dim\_Vendor ON BPI\_Challenge\_V2.Vendor = Dim\_Vendor.Vendor AND BPI\_Challenge\_V2.VendorReceivedPO = Dim\_Vendor.VendorReceivedPO

LEFT JOIN Dim\_Date ON BPI\_Challenge\_V2.Date = Dim\_Date.Date

LEFT JOIN DimTime ON BPI\_Challenge\_V2.Time = DimTime.Time;

Alter table Fact\_Event

ADD EventNo Int IDENTITY(1,1) PRIMARY KEY;

**Queries For 4 PO Process Tables for Process Mining –**

select \*

into Consignmentprocess

from BPI\_Challenge\_V2

where ItemPOProcess = 'Consignment';

select \*

into ThreewayAfterGRprocess

from BPI\_Challenge\_V2

where ItemPOProcess = '3-way match, invoice after GR';

select \*

into ThreewayBeforeGRprocess

from BPI\_Challenge\_V2

where ItemPOProcess = '3-way match, invoice before GR';

select \*

into Twowayprocess

from BPI\_Challenge\_V2

where ItemPOProcess = '2-way match';

\*\*Thank You\*\*