----- Function

CREATE FUNCTION YesterdayDate()

RETURNS DateTime

AS

BEGIN

 RETURN (SELECT dateadd(d,datediff(d,0, getdate()),-1))

 -- Return the result of the function

END

GO

----- Function

CREATE FUNCTION whichContinent

(@Country nvarchar(15))

RETURNS varchar(30)

AS

BEGIN

declare @Return varchar(30)

select @return = case @Country

when 'Argentina' then 'South America'

when 'Belgium' then 'Europe'

when 'Brazil' then 'South America'

when 'Canada' then 'North America'

when 'Denmark' then 'Europe'

when 'Finland' then 'Europe'

when 'France' then 'Europe'

else 'Unknown'

end

return @return

end

-------- Function

IF OBJECT\_ID (N'dbo.ISOweek', N'FN') IS NOT NULL

 DROP FUNCTION dbo.ISOweek;

GO

CREATE FUNCTION dbo.ISOweek (@DATE datetime)

RETURNS int

WITH EXECUTE AS CALLER

AS

BEGIN

 DECLARE @ISOweek int;

 SET @ISOweek= DATEPART(wk,@DATE)+1

 -DATEPART(wk,CAST(DATEPART(yy,@DATE) as CHAR(4))+'0104');

--Special cases: Jan 1-3 may belong to the previous year

 IF (@ISOweek=0)

 SET @ISOweek=dbo.ISOweek(CAST(DATEPART(yy,@DATE)-1

 AS CHAR(4))+'12'+ CAST(24+DATEPART(DAY,@DATE) AS CHAR(2)))+1;

--Special case: Dec 29-31 may belong to the next year

 IF ((DATEPART(mm,@DATE)=12) AND

 ((DATEPART(dd,@DATE)-DATEPART(dw,@DATE))>= 28))

 SET @ISOweek=1;

 RETURN(@ISOweek);

END;

GO

------------ Stored Procedure

CREATE PROCEDURE spPersons

 AS SELECT FirstName, MiddleName, LastName

 FROM Person.Person

 ORDER BY LastName DESC

execute spPersons

SET DATEFIRST 1;

SELECT dbo.ISOweek(CONVERT(DATETIME,'11/12/2015',101)) AS 'ISO Week';

------- Triggers

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[Customers](

 [CustomerId] [int] IDENTITY(1,1) NOT NULL,

 [Name] [varchar](100) NOT NULL,

 [Country] [varchar](50) NOT NULL,

 CONSTRAINT [PK\_Customers] PRIMARY KEY CLUSTERED

(

 [CustomerId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

CREATE TABLE [dbo].[CustomerLogs](

 [LogId] [int] IDENTITY(1,1) NOT NULL,

 [CustomerId] [int] NOT NULL,

 [ACTION] [varchar](50) NOT NULL,

 CONSTRAINT [PK\_CustomerLogs] PRIMARY KEY CLUSTERED

(

 [LogId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

INSERT INTO Customers

SELECT 'John Hammond', 'United States'

UNION ALL

SELECT 'Mudassar Khan', 'India'

UNION ALL

SELECT 'Suzanne Mathews', 'France'

UNION ALL

SELECT 'Robert Schidner', 'Russia'

GO

CREATE TRIGGER [dbo].[Customer\_INSERT]

 ON [dbo].[Customers]

AFTER INSERT

AS

BEGIN

 SET NOCOUNT ON;

 DECLARE @CustomerId INT

 SELECT @CustomerId = INSERTED.CustomerId

 FROM INSERTED

 INSERT INTO CustomerLogs

 VALUES(@CustomerId, 'Inserted')

END

REATE TRIGGER [dbo].[Customer\_UPDATE]

 ON [dbo].[Customers]

AFTER UPDATE

AS

BEGIN

 SET NOCOUNT ON;

 DECLARE @CustomerId INT

 DECLARE @Action VARCHAR(50)

 SELECT @CustomerId = INSERTED.CustomerId

 FROM INSERTED

 IF UPDATE(Name)

 BEGIN

 SET @Action = 'Updated Name'

 END

 IF UPDATE(Country)

 BEGIN

 SET @Action = 'Updated Country'

 END

 INSERT INTO CustomerLogs

 VALUES(@CustomerId, @Action)

END