**Loop:**

In order to be able to filter and present nicely over a period of time the analysis tools, we enlarge the "Usage Main" table by adding rows. We managed to do it automatically by writing a dedicated Loop in T-SQL language.

First, we used View to create a random number that is greater than 1 and an integer

Then, we created a function that we learned during the course that produces a random number.

Next, the Loop based on 'While' loop that gets an "N" variable of number of rows we would like to add. We decided as a team to "run" a loop twice to test it, in "Usage Main" table has 30,557 rows.

Each column of "Usage Main" table gets variable that gets a different parameters on each iteration. Some of the variables are determine randomly by a functions like dates and times, some are determine by sub query which compare to other table in order to keep the data reliability and some are calculated like duration.

In Loop, we added the ability to calculate the "Duration" column and make the data look real.

We also spread the data about five quarters so we can analyze the data interestingly in "Power BI" tool.

We also added an internal loop to avoid the "embarrassment" situation where a customer would call his number, the loop is type "Go to label".

Script attached.

Use TheVoice

go

--Create View Rand

create view [dbo].[vv\_getRANDValue]

as

select rand() as value

GO

--Create function for random number

Create function [dbo].[fn\_RandomNum](@Lower int, @Upper int)

returns int

as

Begin

DECLARE @Random INT;

if @Upper > @Lower

SELECT @Random = (@Upper - @Lower) \* (SELECT Value FROM vv\_getRANDValue) + @Lower

Else

SELECT @Random = (@Lower - @Upper) \* (SELECT Value FROM vv\_getRANDValue) + @Upper

return @Random

end

GO

--Check

select dbo.fn\_RandomNum (40,59)

-- T-SQL Exercise – Final Project

select\* from USAGE\_MAIN

select\* from [Call Type]

select\* from [OPFILEOPP]

select\* from [customer]

select\* from [Call Type]

--Let's see the participants

Select count(\*) from USAGE\_MAIN

Select Max(CALL\_NO) from USAGE\_MAIN

--USAGE\_MAIN Amt = 504

--Max USAGE\_MAIN = 504

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

Declare

@NumberOfNewRows int = 10557,

@I int = 1,

@CallNumber int,

@CallingNumber nvarchar(20),

@CalledNumber nvarchar(20),

@Duration int,

@CustomerID int,

@CallType nvarchar (20),

@DestinationNumber nvarchar(20),

@ProductType nvarchar (20),

@Cell int,

@CellOrigin int,

@RatedAmount int,

@AnswerTime datetime,

@DisconectTime datetime,

@FromDate datetime,

@ToDate datetime,

@Minutes int,

@Max int,

@Min int,

@CallDateTime date,

@N int

set @FromDate = '2013-10-10 19:32:44'

set @ToDate = '2014-12-31 23:59:59'

While @NumberOfNewRows >= @I

BEGIN

BEGIN

set @Minutes = DATEDIFF(MINUTE, @FromDate, @ToDate)

set @Max = (select MAX(DURATION) as Duration from USAGE\_MAIN)

set @Min = (select MIN(DURATION) as Duration from USAGE\_MAIN)

set @Duration = dbo.fn\_RandomNum(@Max,@Min)

END

BEGIN

set @Minutes = @Duration

set @AnswerTime =DATEADD(DAY,dbo.fn\_RandomNum(0,DATEDIFF(day,@FromDate,@ToDate)), @FromDate)

set @DisconectTime = DATEADD(MINUTE, @Duration, @AnswerTime)

set @CallDateTime = @AnswerTime

END

BEGIN

set @CallType = (Select top 1 [Call Type Code] from [Call Type] order by NEWID())

set @CallingNumber = (Select top 1 CUST\_NUMBER from customer order by NEWID()) set @CalledNumber = (Select top 1 CUST\_NUMBER from customer order by NEWID())

set @CustomerID = (select customer\_id from customer where CUST\_NUMBER = @CallingNumber)

set @RatedAmount = dbo.fn\_RandomNum(6,0)

set @DestinationNumber = @CalledNumber

set @N = 1

END

BEGIN

IF LEN(@CallingNumber) = 13

SET @Cell = 1

ELSE

SET @Cell = 0

END

BEGIN

IF LEN(@CalledNumber) = 13

SET @CellOrigin = 1

ELSE

SET @CellOrigin = 0

END

BEGIN

set @ProductType = (Select top 1 [Call Type Desc] from [Call Type] where @CallType = [Call Type Code])

Lable:

Insert into USAGE\_MAIN (ANSWER\_TIME,SEIZED\_TIME, DISCONNECT\_TIME, CALL\_DATETIME,CALLING\_NO,CALLED\_NO, DES\_NO, DURATION, CUST\_ID, CALL\_TYPE, PROD\_TYPE,RATED\_AMNT,RATED\_CURR\_CODE,CELL, CELL\_ORIGIN, HIGH\_LOW\_RATE, insert\_DATE,update\_date)

Values (@AnswerTime, @AnswerTime, @DisconectTime, @CallDateTime, @CallingNumber, @CalledNumber, @DestinationNumber, @Duration, @CustomerID, @CallType, @ProductType, @RatedAmount, 'SHEKEL', @Cell, @CellOrigin, 1, @AnswerTime, NULL)

While exists (Select \* From USAGE\_MAIN where CALLING\_NO = @CallingNumber and CALLED\_NO = @CalledNumber)

Begin

Set @CalledNumber = (Select top 1 CUST\_NUMBER from customer order by NEWID())

End

Set @N = @N + 1

If 1 >= @N GOTO Lable

Set @I = @I + 1

END

END