/\*\* Initialize Data Structure \*\*/

USE Telephone\_Company

DROP TABLE Call

DROP TABLE Customer

DROP TABLE Peak\_Time

DROP TABLE Rate

DROP TABLE Sales\_Rep

DROP TABLE Calling\_Code

DROP TABLE Service

CREATE TABLE Service (service\_code VARCHAR(255) PRIMARY KEY)

CREATE TABLE Calling\_Code (calling\_code INT PRIMARY KEY, country\_name VARCHAR(255))

CREATE TABLE Sales\_Rep (salerep\_id INT PRIMARY KEY)

CREATE TABLE Rate (source\_code INT, dest\_code INT, service\_code VARCHAR(255),

effective\_date DATE, peak NUMERIC(3, 2), offpeak NUMERIC(3, 2),

CONSTRAINT PK\_Rate PRIMARY KEY(source\_code, dest\_code, service\_code, effective\_date),

FOREIGN KEY (source\_code) REFERENCES Calling\_Code(calling\_code),

FOREIGN KEY (dest\_code) REFERENCES Calling\_Code(calling\_code),

FOREIGN KEY (service\_code) REFERENCES Service(service\_code))

CREATE TABLE Peak\_Time (service\_code VARCHAR(255), source\_code INT, peak\_start TIME, offpeak\_start TIME,

CONSTRAINT PK\_Peak\_Time PRIMARY KEY(service\_code, source\_code),

FOREIGN KEY (service\_code) REFERENCES Service(service\_code),

FOREIGN KEY (source\_code) REFERENCES Calling\_Code(calling\_code))

CREATE TABLE Customer (telephone VARCHAR(20) PRIMARY KEY, name VARCHAR(255), address VARCHAR(255),

service\_code VARCHAR(255), salerep\_id INT, reps\_cmsrate INT,

FOREIGN KEY (service\_code) REFERENCES Service(service\_code),

FOREIGN KEY (salerep\_id) REFERENCES Sales\_Rep(salerep\_id))

CREATE TABLE Call (from\_tel VARCHAR(20), call\_date DATE, call\_time TIME,

from\_code INT, to\_code INT, to\_tel VARCHAR(20), duration INT,

CONSTRAINT PK\_Call PRIMARY KEY(from\_tel, call\_date, call\_time),

FOREIGN KEY(from\_tel) REFERENCES Customer(telephone),

FOREIGN KEY(from\_code) REFERENCES Calling\_Code(calling\_code),

FOREIGN KEY(to\_code) REFERENCES Calling\_Code(calling\_code))

/\*\* Initialize Storaged Procedure \*\*/

IF (OBJECT\_ID('MergeIntoCustomer')) IS NOT NULL DROP PROCEDURE MergeIntoCustomer

GO

IF (OBJECT\_ID('MergeIntoCustomerFromFile')) IS NOT NULL DROP PROCEDURE MergeIntoCustomerFromFile

GO

IF (OBJECT\_ID('MergeIntoRate')) IS NOT NULL DROP PROCEDURE MergeIntoRate

GO

IF (OBJECT\_ID('MergeIntoRateFromFile')) IS NOT NULL DROP PROCEDURE MergeIntoRateFromFile

GO

IF (OBJECT\_ID('MergeIntoCall')) IS NOT NULL DROP PROCEDURE MergeIntoCall

GO

IF (OBJECT\_ID('MergeIntoCallFromFile')) IS NOT NULL DROP PROCEDURE MergeIntoCallFromFile

GO

IF (OBJECT\_ID('CreateCustomerBill')) IS NOT NULL DROP PROCEDURE CreateCustomerBill

GO

IF (OBJECT\_ID('GenerateTrafficSummary')) IS NOT NULL DROP PROCEDURE GenerateTrafficSummary

GO

IF (OBJECT\_ID('GenerateSalesrepCommissions')) IS NOT NULL DROP PROCEDURE GenerateSalesrepCommissions

GO

IF (OBJECT\_ID('ExportCurrentRates')) IS NOT NULL DROP PROCEDURE ExportCurrentRates

GO

IF (OBJECT\_ID('UpdateCustomerWithSidAndCms')) IS NOT NULL DROP PROCEDURE UpdateCustomerWithSidAndCms

GO

/\*\* MergeIntoCustomer \*\*/

CREATE PROCEDURE MergeIntoCustomer @telephone VARCHAR(20), @name VARCHAR(255), @address VARCHAR(255),

@service\_code VARCHAR(255), @salerep\_id INT, @reps\_cmsrate INT

AS

DECLARE @count INT = 0

SELECT @count = count(1) FROM Customer WHERE telephone = @telephone

IF @count = 0

INSERT INTO Customer VALUES (@telephone, @name, @address, @service\_code, @salerep\_id, @reps\_cmsrate)

ELSE

UPDATE Customer SET name = @name, address = @address, service\_code = @service\_code

WHERE telephone = @telephone

GO

/\*\* Prepare Customer salerep\_id and reps\_cmsrate \*\*/

CREATE PROCEDURE UpdateCustomerWithSidAndCms

AS

/\*\* Update Customers' SalesReps and SalesReps' Commission Rate \*\*/

UPDATE Customer SET salerep\_id = 23, reps\_cmsrate = 8 WHERE name = 'Randy Jones'

UPDATE Customer SET salerep\_id = 12, reps\_cmsrate = 10 WHERE name = 'David Stephenson'

UPDATE Customer SET salerep\_id = 22, reps\_cmsrate = 7 WHERE name = 'Thomas Tripp'

UPDATE Customer SET salerep\_id = 15, reps\_cmsrate = 5 WHERE name = 'Mary Manning'

UPDATE Customer SET salerep\_id = 12, reps\_cmsrate = 8 WHERE name = 'Robert McCracken'

UPDATE Customer SET salerep\_id = 18, reps\_cmsrate = 10 WHERE name = 'Dale Malone'

UPDATE Customer SET salerep\_id = 23, reps\_cmsrate = 6 WHERE name = 'Harold Simpson'

UPDATE Customer SET salerep\_id = 12, reps\_cmsrate = 9 WHERE name = 'William Thurston'

UPDATE Customer SET salerep\_id = 18, reps\_cmsrate = 6 WHERE name = 'Mike Tucker'

UPDATE Customer SET salerep\_id = 24, reps\_cmsrate = 9 WHERE name = 'Conrad Nolte'

UPDATE Customer SET salerep\_id = 22, reps\_cmsrate = 10 WHERE name = 'Robert Olden'

UPDATE Customer SET salerep\_id = 14, reps\_cmsrate = 5 WHERE name = 'Theresa Palm'

UPDATE Customer SET salerep\_id = 15, reps\_cmsrate = 7 WHERE name = 'Alfred Kalmbach'

UPDATE Customer SET salerep\_id = 22, reps\_cmsrate = 10 WHERE name = 'Anna Stein'

UPDATE Customer SET salerep\_id = 23, reps\_cmsrate = 9 WHERE name = 'Emil Aston'

UPDATE Customer SET salerep\_id = 19, reps\_cmsrate = 10 WHERE name = 'Ewald Reynolds'

UPDATE Customer SET salerep\_id = 19, reps\_cmsrate = 5 WHERE name = 'Helga Stover'

UPDATE Customer SET salerep\_id = 17, reps\_cmsrate = 6 WHERE name = 'Rudolf Torpini'

UPDATE Customer SET salerep\_id = 17, reps\_cmsrate = 8 WHERE name = 'Alberto Russo'

UPDATE Customer SET salerep\_id = 19, reps\_cmsrate = 10 WHERE name = 'Adam Smith'

GO

/\*\* MergeIntoCustomerFromFile \*\*/

CREATE PROCEDURE MergeIntoCustomerFromFile @filePath VARCHAR(255), @sheetName VARCHAR(255)

AS

DECLARE @telephone VARCHAR(20), @name VARCHAR(255), @address VARCHAR(255),

@service\_code VARCHAR(255), @ssql VARCHAR(255)

CREATE TABLE #Temp\_Customer

(

telephone VARCHAR(20),

name VARCHAR(255),

address VARCHAR(255),

service VARCHAR(255)

)

SET @ssql = 'INSERT INTO #Temp\_Customer SELECT \* FROM OPENROWSET(''Microsoft.ACE.OLEDB.12.0'', ''Excel 12.0; Database=' + @filePath + ''', [' + @sheetName + '$])'

EXEC(@ssql)

DECLARE @count INT = 0

SELECT @count = count(1) FROM #Temp\_Customer

WHILE @count <> 0

BEGIN

SELECT TOP 1 @telephone = telephone, @name = name, @address = address, @service\_code = service FROM #Temp\_Customer

EXEC MergeIntoCustomer @telephone, @name, @address, @service\_code, NULL, NULL

DELETE FROM #Temp\_Customer WHERE telephone = @telephone

SET @count = @count - 1

END

EXEC UpdateCustomerWithSidAndCms

GO

/\*\* MergeIntoRate \*\*/

CREATE PROCEDURE MergeIntoRate @source\_name VARCHAR(255), @dest\_code INT, @service\_code VARCHAR(255),

@effective\_date DATE, @peak NUMERIC(3, 2), @offpeak NUMERIC(3, 2)

AS

DECLARE @count INT = 0

SELECT @count = count(1) FROM Rate r, Calling\_Code c WHERE r.source\_code = c.calling\_code AND c.country\_name = @source\_name AND dest\_code = @dest\_code AND service\_code = @service\_code AND effective\_date = @effective\_date

IF @count = 0

INSERT INTO Rate (source\_code, dest\_code, service\_code, effective\_date, peak, offpeak) SELECT c.calling\_code, @dest\_code, @service\_code, @effective\_date, @peak, @offpeak FROM Calling\_Code c WHERE c.country\_name = @source\_name

ELSE

UPDATE Rate SET peak = @peak, offpeak = @offpeak

WHERE EXISTS(SELECT 1 FROM Calling\_Code c WHERE source\_code = c.calling\_code AND c.country\_name = @source\_name AND dest\_code = @dest\_code AND service\_code = @service\_code AND effective\_date = @effective\_date)

GO

/\*\* MergeIntoRateFromFile \*\*/

CREATE PROCEDURE MergeIntoRateFromFile @filePath VARCHAR(255)

AS

DECLARE @source\_name VARCHAR(255), @dest\_code INT, @service\_code VARCHAR(255),

@effective\_date DATE, @peak NUMERIC(3, 2), @offpeak NUMERIC(3, 2), @linkedServerName SYSNAME = 'TempExcelSpreadsheet'

-- Remove existing linked server (if necessary)

IF EXISTS(SELECT NULL FROM sys.servers WHERE name = @linkedServerName) BEGIN

EXEC sp\_dropserver @server = @linkedServerName, @droplogins = 'droplogins'

END

-- Add the linked server

-- ACE 12.0 seems to work for both xsl and xslx, though some might prefer the older JET provider

EXEC sp\_addlinkedserver

@server = @linkedServerName,

@srvproduct = 'ACE 12.0',

@provider = 'Microsoft.ACE.OLEDB.12.0',

@datasrc = @filePath,

@provstr = 'Excel 12.0;HDR=Yes'

-- Grab the current user to use as a remote login

DECLARE @suser\_sname VARCHAR(256) = suser\_sname()

-- Add the current user as a login

EXEC sp\_addlinkedsrvlogin

@rmtsrvname = @linkedServerName,

@useself = 'false',

@locallogin = @suser\_sname,

@rmtuser = NULL,

@rmtpassword = NULL

-- Return the table/column info

DECLARE @tmpSheetName TABLE(TABLE\_CAT VARCHAR(20), TABLE\_SCHEM VARCHAR(20), TABLE\_NAME VARCHAR(255), TABLE\_TYPE VARCHAR(20), REMARKS VARCHAR(20))

INSERT INTO @tmpSheetName EXEC sp\_tables\_ex @linkedServerName

DECLARE @count INT = 0, @tbName VARCHAR(100)

SELECT @count = count(\*) FROM @tmpSheetName WHERE table\_name NOT LIKE '%FilterDatabase'

DECLARE @effdata VARCHAR(8) = SUBSTRING(@filePath, charindex('\_', @filePath) + 1, 8),

@sercode VARCHAR(20),

@countryname VARCHAR(50),

@ssql VARCHAR(500),

@countSheetData INT

CREATE TABLE #tmpRate(source\_name VARCHAR(255), dest\_code INT, service\_code VARCHAR(255),

effective\_date DATE, peak NUMERIC(3, 2), offpeak NUMERIC(3, 2))

WHILE @count <> 0

BEGIN

SELECT TOP 1 @tbName = TABLE\_NAME FROM @tmpSheetName WHERE table\_name NOT LIKE '%FilterDatabase'

SET @sercode = REPLACE(SUBSTRING(@tbName, 1, CHARINDEX('\_', @tbName) - 1),'''','')

SET @countryname = REPLACE(SUBSTRING(@tbName, CHARINDEX('\_', @tbName) + 1, LEN(@tbName) - CHARINDEX('\_', @tbName) - 1),'''','')

SET @ssql = 'INSERT INTO #tmpRate SELECT ''' + @countryname + ''', dest,''' + @sercode + ''',''' + @effdata + ''', peak, offpeak FROM OPENROWSET(''Microsoft.ACE.OLEDB.12.0'', ''Excel 12.0; Database=' + @filePath + ''', [' + @tbName + '])'

EXEC(@ssql)

SELECT @countSheetData = COUNT(1) FROM #tmpRate

WHILE @countSheetData <> 0

BEGIN

SELECT TOP 1 @source\_name = source\_name, @dest\_code = dest\_code, @service\_code = service\_code,

@effective\_date = effective\_date, @peak = peak, @offpeak = offpeak

FROM #tmpRate

EXEC MergeIntoRate @source\_name, @dest\_code, @service\_code, @effective\_date, @peak, @offpeak

DELETE FROM #tmpRate WHERE source\_name = @source\_name AND dest\_code = @dest\_code AND service\_code = @service\_code AND effective\_date = @effective\_date

SET @countSheetData = @countSheetData - 1

END

DELETE FROM @tmpSheetName WHERE TABLE\_NAME = @tbName

SET @count = @count - 1

END

-- Remove temp linked server

IF EXISTS(SELECT NULL FROM sys.servers WHERE name = @linkedServerName) BEGIN

EXEC sp\_dropserver @server = @linkedServerName, @droplogins = 'droplogins'

END

GO

/\*\* MergeIntoCall \*\*/

CREATE PROCEDURE MergeIntoCall @from\_tel VARCHAR(20), @call\_date DATE, @call\_time TIME,

@from\_code INT, @to\_code INT, @to\_tel VARCHAR(20), @duration INT

AS

DECLARE @count INT = 0

SELECT @count = count(1) FROM [Call] c WHERE c.from\_tel = @from\_tel AND c.call\_date = @call\_date AND c.call\_time = @call\_time

IF @count = 0

INSERT INTO [Call] (from\_tel, call\_date, call\_time, from\_code, to\_code, to\_tel, duration) VALUES (@from\_tel, @call\_date, @call\_time, @from\_code, @to\_code, @to\_tel, @duration)

ELSE

UPDATE [Call] SET from\_code = @from\_code, to\_code = @to\_code, to\_tel = @to\_tel, duration = @duration

WHERE from\_tel = @from\_tel AND call\_date = @call\_date AND call\_time = @call\_time

GO

/\*\* MergeIntoCustomerFromFile \*\*/

CREATE PROCEDURE MergeIntoCallFromFile @filePath VARCHAR(255), @sheetName VARCHAR(255)

AS

DECLARE @from\_tel VARCHAR(20), @call\_date DATE, @call\_time TIME,

@from\_code INT, @to\_code INT, @to\_tel VARCHAR(20), @duration INT, @ssql VARCHAR(500)

CREATE TABLE #Temp\_Call

(

from\_tel VARCHAR(20), call\_date DATE, call\_time TIME,

from\_code INT, to\_code INT, to\_tel VARCHAR(20), duration INT

)

SET @ssql = 'INSERT INTO #Temp\_Call SELECT from\_tel, call\_date, SUBSTRING(CAST(call\_time AS VARCHAR(10)), 1, LEN(CAST(call\_time AS VARCHAR(10))) - 2) + '':'' + SUBSTRING(CAST(call\_time AS VARCHAR(10)), LEN(CAST(call\_time AS VARCHAR(10))) - 1, 2) AS call\_time, from\_code, to\_code, to\_tel, duration FROM OPENROWSET(''Microsoft.ACE.OLEDB.12.0'', ''Excel 12.0; Database=' + @filePath + ''', [' + @sheetName + '$])'

EXEC(@ssql)

DECLARE @count INT = 0

SELECT @count = count(1) FROM #Temp\_Call

WHILE @count <> 0

BEGIN

SELECT TOP 1 @from\_tel = from\_tel, @call\_date = call\_date, @call\_time = call\_time,

@from\_code = from\_code, @to\_code = to\_code, @to\_tel = to\_tel, @duration = duration FROM #Temp\_Call

EXEC MergeIntoCall @from\_tel, @call\_date, @call\_time, @from\_code, @to\_code, @to\_tel, @duration

DELETE FROM #Temp\_Call WHERE from\_tel = @from\_tel AND call\_date = @call\_date AND call\_time = @call\_time

SET @count = @count - 1

END

GO

/\*\* Create Customer Bill \*\*/

CREATE PROCEDURE CreateCustomerBill @dateym VARCHAR(6), @filePath VARCHAR(255)

AS

DECLARE @firstDayThisMonth VARCHAR(8) = @dateym + '01'

SELECT \* INTO #TmpMaxValidData FROM(

SELECT MAX(r.effective\_date) maxEFDate, cu.name, cu.address, cu.service\_code, c.from\_tel, c.call\_date,

c.call\_time, c.duration, c.from\_code, c.to\_code, r.peak, r.offpeak

FROM Rate r, Call c, Customer cu

WHERE c.call\_date >= CAST(@firstDayThisMonth AS DATE) AND c.call\_date < DATEADD(month, 1, CAST(@firstDayThisMonth AS DATE)) AND

r.source\_code = c.from\_code AND r.dest\_code = c.to\_code AND c.from\_tel = cu.telephone AND r.service\_code = cu.service\_code

AND c.call\_date >= r.effective\_date

group by cu.name, cu.address, cu.service\_code, c.from\_tel, c.call\_date, c.call\_time, c.duration, c.from\_code, c.to\_code,

r.peak, r.offpeak) TEMP

SELECT \* INTO #DataWhichMaxEFDBeforeThatMonth FROM

(SELECT \* FROM #TmpMaxValidData WHERE maxEFDate <= CAST(@firstDayThisMonth AS DATE)) T

/\*SELECT \* INTO #DataWhichMaxEFDAfterThatMonth FROM

(SELECT \* FROM #TmpMaxValidData WHERE maxEFDate > CAST(@firstDayThisMonth AS DATE)) T\*/

--- Caculate maxEFDateBeforeThatMonth Bill

SELECT \* INTO #Bill FROM (SELECT d.name, d.address, d.from\_tel AS telephone, c.country\_name, d.duration, d.call\_time,

CASE WHEN d.call\_time >= p.peak\_start AND d.call\_time < p.offpeak\_start

THEN d.peak \* d.duration / 60

ELSE d.offpeak \* d.duration / 60

END AS cost

FROM #DataWhichMaxEFDBeforeThatMonth d, Calling\_Code c, Peak\_Time p

WHERE d.to\_code = c.calling\_code AND d.service\_code = p.service\_code AND d.from\_code = p.source\_code) T

DECLARE @ssql VARCHAR(500) = 'INSERT INTO OPENROWSET(''Microsoft.ACE.OLEDB.12.0'', ''Excel 12.0; Database=' + @filePath + ''', ''SELECT \* FROM [Sheet1$]'')'

+ 'SELECT b.\*, T.Amount\_Due FROM #Bill b,

(SELECT telephone, SUM(cost) AS Amount\_Due FROM #Bill

GROUP BY telephone) T

WHERE b.telephone = T.telephone'

EXEC(@ssql)

--- Caculate maxEFDateAfterThatMonth Bill

/\*SELECT \* INTO #AfterT FROM (SELECT d.name, d.address, d.from\_tel, c.country\_name, d.duration, d.call\_time

FROM #DataWhichMaxEFDAfterThatMonth d, Rate r, Calling\_Code c, Peak\_Time p

WHERE r.source\_code = d.from\_code AND r.dest\_code = d.to\_code AND r.service\_code = d.service\_code

AND d.to\_code = c.calling\_code AND d.service\_code = p.service\_code AND d.from\_code = p.source\_code

AND r.effective\_date < DATEADD(month, 1, CAST(@firstDayThisMonth AS DATE))) Tt

DECLARE @countD INT = 0, @countR INT = 0 \*/

GO

/\*\* Generate traffic summary \*\*/

CREATE PROCEDURE GenerateTrafficSummary @dateym VARCHAR(6), @filePath VARCHAR(255)

AS

DECLARE @firstDayThisMonth VARCHAR(8) = @dateym + '01'

SELECT \* INTO #TmpTrafficSummary FROM

(SELECT cu.service\_code AS service\_name, cc1.country\_name AS from\_country\_name, cc2.country\_name AS to\_country\_name, SUM(c.duration) total\_calling\_minutes

FROM [Call] c, Calling\_Code cc1, Calling\_Code cc2, Customer cu

WHERE c.from\_code = cc1.calling\_code AND c.to\_code = cc2.calling\_code

AND c.from\_tel = cu.telephone AND c.call\_date >= CAST(@firstDayThisMonth AS DATE)

AND c.call\_date < DATEADD(month, 1, CAST(@firstDayThisMonth AS DATE))

GROUP BY cu.service\_code, c.from\_code, cc1.country\_name, c.to\_code, cc2.country\_name) T

DECLARE @ssql VARCHAR(1000) = 'INSERT INTO OPENROWSET(''Microsoft.ACE.OLEDB.12.0'', ''Excel 12.0; Database=' + @filePath + ''', ''SELECT \* FROM [Sheet1$]'')'

+ 'SELECT \* FROM #TmpTrafficSummary'

EXEC(@ssql)

GO

/\*\* Generate Salesrep Commissions \*\*/

CREATE PROCEDURE GenerateSalesrepCommissions @dateym VARCHAR(6), @filePath VARCHAR(255)

AS

DECLARE @firstDayThisMonth VARCHAR(8) = @dateym + '01'

SELECT \* INTO #TmpMaxValidData FROM(

SELECT MAX(r.effective\_date) maxEFDate, cu.name, cu.address, cu.service\_code, c.from\_tel, c.call\_date,

c.call\_time, c.duration, c.from\_code, c.to\_code, r.peak, r.offpeak

FROM Rate r, Call c, Customer cu

WHERE c.call\_date >= CAST(@firstDayThisMonth AS DATE) AND c.call\_date < DATEADD(month, 1, CAST(@firstDayThisMonth AS DATE)) AND

r.source\_code = c.from\_code AND r.dest\_code = c.to\_code AND c.from\_tel = cu.telephone AND r.service\_code = cu.service\_code

AND c.call\_date >= r.effective\_date

group by cu.name, cu.address, cu.service\_code, c.from\_tel, c.call\_date, c.call\_time, c.duration, c.from\_code, c.to\_code,

r.peak, r.offpeak) TEMP

SELECT \* INTO #DataWhichMaxEFDBeforeThatMonth FROM

(SELECT \* FROM #TmpMaxValidData WHERE maxEFDate <= CAST(@firstDayThisMonth AS DATE)) T

--- Caculate maxEFDateBeforeThatMonth Bill

SELECT \* INTO #Bill FROM (SELECT d.name, d.address, d.from\_tel AS telephone, c.country\_name, d.duration, d.call\_time,

CASE WHEN d.call\_time >= p.peak\_start AND d.call\_time < p.offpeak\_start

THEN d.peak \* d.duration / 60

ELSE d.offpeak \* d.duration / 60

END AS cost

FROM #DataWhichMaxEFDBeforeThatMonth d, Calling\_Code c, Peak\_Time p

WHERE d.to\_code = c.calling\_code AND d.service\_code = p.service\_code AND d.from\_code = p.source\_code) T

DECLARE @ssql VARCHAR(500) = 'INSERT INTO OPENROWSET(''Microsoft.ACE.OLEDB.12.0'', ''Excel 12.0; Database=' + @filePath + ''', ''SELECT \* FROM [Sheet1$]'')'

+ 'SELECT c.salerep\_id, SUM(T.commission) total\_commission FROM Customer c,

(SELECT c.telephone, SUM(cost \* reps\_cmsrate / 100) AS commission FROM #Bill b, Customer c WHERE b.telephone = c.telephone

GROUP BY c.telephone, c.reps\_cmsrate) T

WHERE c.telephone = T.telephone GROUP BY c.salerep\_id'

EXEC(@ssql)

GO

/\*\* Generate Salesrep Commissions \*\*/

CREATE PROCEDURE ExportCurrentRates @serviceName VARCHAR(20), @sourceCountry VARCHAR(255), @filePath VARCHAR(255)

AS

SELECT \* INTO #ttRates FROM

(SELECT cc1.country\_name dest\_country\_name, r.peak, r.offpeak

FROM (SELECT MAX(r.effective\_date) effective\_date, r.source\_code, r.dest\_code, r.service\_code

FROM Rate r GROUP BY r.source\_code, r.dest\_code, r.service\_code) ra, Rate r, Calling\_Code cc, Calling\_Code cc1

WHERE ra.effective\_date = r.effective\_date AND ra.source\_code = r.source\_code AND ra.dest\_code = r.dest\_code AND ra.service\_code = r.service\_code

AND ra.source\_code = cc.calling\_code AND ra.dest\_code = cc1.calling\_code AND ra.service\_code = @serviceName AND cc.country\_name = @sourceCountry) T

DECLARE @ssql VARCHAR(500) = 'INSERT INTO OPENROWSET(''Microsoft.ACE.OLEDB.12.0'', ''Excel 12.0; Database=' + @filePath

+ ''', ''SELECT \* FROM [Sheet1$]'')'

+ 'SELECT \* FROM #ttRates'

EXEC(@ssql)

GO

/\*\* Initialize Data \*\*/

/\*\* Initialize Data of Calling\_Code \*\*/

INSERT INTO Calling\_Code VALUES (1, 'USA')

INSERT INTO Calling\_Code VALUES (7, 'Russia')

INSERT INTO Calling\_Code VALUES (20, 'Egypt')

INSERT INTO Calling\_Code VALUES (27, 'South Africa')

INSERT INTO Calling\_Code VALUES (30, 'Greece ')

INSERT INTO Calling\_Code VALUES (31, 'Netherlands')

INSERT INTO Calling\_Code VALUES (32, 'Belgium')

INSERT INTO Calling\_Code VALUES (33, 'France')

INSERT INTO Calling\_Code VALUES (34, 'Spain')

INSERT INTO Calling\_Code VALUES (36, 'Hungary ')

INSERT INTO Calling\_Code VALUES (39, 'Italy ')

INSERT INTO Calling\_Code VALUES (40, 'Romania')

INSERT INTO Calling\_Code VALUES (41, 'Switzerland')

INSERT INTO Calling\_Code VALUES (43, 'Austria')

INSERT INTO Calling\_Code VALUES (44, 'United Kingdom')

INSERT INTO Calling\_Code VALUES (45, 'Denmark')

INSERT INTO Calling\_Code VALUES (46, 'Sweden')

INSERT INTO Calling\_Code VALUES (47, 'Norway ')

INSERT INTO Calling\_Code VALUES (48, 'Poland')

INSERT INTO Calling\_Code VALUES (49, 'Germany')

INSERT INTO Calling\_Code VALUES (51, 'Peru')

INSERT INTO Calling\_Code VALUES (52, 'Mexico')

INSERT INTO Calling\_Code VALUES (53, 'Cuba')

INSERT INTO Calling\_Code VALUES (54, 'Argentina')

INSERT INTO Calling\_Code VALUES (55, 'Brazil ')

INSERT INTO Calling\_Code VALUES (56, 'Chile ')

INSERT INTO Calling\_Code VALUES (57, 'Colombia ')

INSERT INTO Calling\_Code VALUES (58, 'Venezuela')

INSERT INTO Calling\_Code VALUES (60, 'Malaysia')

INSERT INTO Calling\_Code VALUES (61, 'Australia ')

INSERT INTO Calling\_Code VALUES (62, 'Indonesia')

INSERT INTO Calling\_Code VALUES (63, 'Philippines')

INSERT INTO Calling\_Code VALUES (64, 'New Zealand')

INSERT INTO Calling\_Code VALUES (65, 'Singapore')

INSERT INTO Calling\_Code VALUES (66, 'Thailand')

INSERT INTO Calling\_Code VALUES (81, 'Japan ')

INSERT INTO Calling\_Code VALUES (82, 'Korea (South)')

INSERT INTO Calling\_Code VALUES (84, 'Vietnam')

INSERT INTO Calling\_Code VALUES (86, 'China (PRC)')

INSERT INTO Calling\_Code VALUES (90, 'Turkey')

INSERT INTO Calling\_Code VALUES (91, 'India')

INSERT INTO Calling\_Code VALUES (92, 'Pakistan')

INSERT INTO Calling\_Code VALUES (93, 'Afghanistan')

INSERT INTO Calling\_Code VALUES (94, 'Sri Lanka')

INSERT INTO Calling\_Code VALUES (95, 'Myanmar')

INSERT INTO Calling\_Code VALUES (98, 'Iran')

INSERT INTO Calling\_Code VALUES (212, 'Morocco')

INSERT INTO Calling\_Code VALUES (213, 'Algeria')

INSERT INTO Calling\_Code VALUES (216, 'Tunisia')

INSERT INTO Calling\_Code VALUES (218, 'Libya')

INSERT INTO Calling\_Code VALUES (220, 'Gambia')

INSERT INTO Calling\_Code VALUES (221, 'Senegal ')

INSERT INTO Calling\_Code VALUES (222, 'Mauritania')

INSERT INTO Calling\_Code VALUES (223, 'Mali Republic')

INSERT INTO Calling\_Code VALUES (224, 'Guinea')

INSERT INTO Calling\_Code VALUES (225, 'Côte d''Ivoire (Ivory Coast)')

INSERT INTO Calling\_Code VALUES (226, 'Burkina Faso ')

INSERT INTO Calling\_Code VALUES (227, 'Niger')

INSERT INTO Calling\_Code VALUES (228, 'Togolese Republic')

INSERT INTO Calling\_Code VALUES (229, 'Benin')

INSERT INTO Calling\_Code VALUES (230, 'Mauritius')

INSERT INTO Calling\_Code VALUES (231, 'Liberia')

INSERT INTO Calling\_Code VALUES (232, 'Sierra Leone')

INSERT INTO Calling\_Code VALUES (233, 'Ghana ')

INSERT INTO Calling\_Code VALUES (234, 'Nigeria')

INSERT INTO Calling\_Code VALUES (235, 'Chad ')

INSERT INTO Calling\_Code VALUES (236, 'Central African Republic')

INSERT INTO Calling\_Code VALUES (237, 'Cameroon')

INSERT INTO Calling\_Code VALUES (238, 'Cape Verde Islands')

INSERT INTO Calling\_Code VALUES (239, 'São Tomé and Principe')

INSERT INTO Calling\_Code VALUES (240, 'Equatorial Guinea')

INSERT INTO Calling\_Code VALUES (241, 'Gabonese Republic')

INSERT INTO Calling\_Code VALUES (242, 'Congo')

INSERT INTO Calling\_Code VALUES (243, 'Congo, Dem. Rep. of ')

INSERT INTO Calling\_Code VALUES (244, 'Angola ')

INSERT INTO Calling\_Code VALUES (245, 'Guinea-Bissau ')

INSERT INTO Calling\_Code VALUES (246, 'Diego Garcia')

INSERT INTO Calling\_Code VALUES (247, 'Ascension')

INSERT INTO Calling\_Code VALUES (248, 'Seychelles Republic')

INSERT INTO Calling\_Code VALUES (249, 'Sudan')

INSERT INTO Calling\_Code VALUES (250, 'Rwandese Republic')

INSERT INTO Calling\_Code VALUES (251, 'Ethiopia')

INSERT INTO Calling\_Code VALUES (252, 'Somali Democratic Republic')

INSERT INTO Calling\_Code VALUES (253, 'Djibouti')

INSERT INTO Calling\_Code VALUES (254, 'Kenya')

INSERT INTO Calling\_Code VALUES (255, 'Tanzania')

INSERT INTO Calling\_Code VALUES (256, 'Uganda')

INSERT INTO Calling\_Code VALUES (257, 'Burundi')

INSERT INTO Calling\_Code VALUES (258, 'Mozambique')

INSERT INTO Calling\_Code VALUES (260, 'Zambia ')

INSERT INTO Calling\_Code VALUES (261, 'Madagascar')

INSERT INTO Calling\_Code VALUES (262, 'Réunion Island')

INSERT INTO Calling\_Code VALUES (263, 'Zimbabwe ')

INSERT INTO Calling\_Code VALUES (264, 'Namibia')

INSERT INTO Calling\_Code VALUES (265, 'Malawi ')

INSERT INTO Calling\_Code VALUES (266, 'Lesotho')

INSERT INTO Calling\_Code VALUES (267, 'Botswana ')

INSERT INTO Calling\_Code VALUES (268, 'Swaziland')

INSERT INTO Calling\_Code VALUES (269, 'Comoros')

INSERT INTO Calling\_Code VALUES (290, 'St. Helena')

INSERT INTO Calling\_Code VALUES (291, 'Eritrea')

INSERT INTO Calling\_Code VALUES (297, 'Aruba')

INSERT INTO Calling\_Code VALUES (298, 'Faroe Islands')

INSERT INTO Calling\_Code VALUES (299, 'Greenland ')

INSERT INTO Calling\_Code VALUES (350, 'Gibraltar ')

INSERT INTO Calling\_Code VALUES (351, 'Portugal')

INSERT INTO Calling\_Code VALUES (352, 'Luxembourg')

INSERT INTO Calling\_Code VALUES (353, 'Ireland')

INSERT INTO Calling\_Code VALUES (354, 'Iceland')

INSERT INTO Calling\_Code VALUES (355, 'Albania')

INSERT INTO Calling\_Code VALUES (356, 'Malta')

INSERT INTO Calling\_Code VALUES (357, 'Cyprus')

INSERT INTO Calling\_Code VALUES (358, 'Finland')

INSERT INTO Calling\_Code VALUES (359, 'Bulgaria')

INSERT INTO Calling\_Code VALUES (370, 'Lithuania ')

INSERT INTO Calling\_Code VALUES (371, 'Latvia ')

INSERT INTO Calling\_Code VALUES (372, 'Estonia')

INSERT INTO Calling\_Code VALUES (373, 'Moldova ')

INSERT INTO Calling\_Code VALUES (374, 'Armenia ')

INSERT INTO Calling\_Code VALUES (375, 'Belarus')

INSERT INTO Calling\_Code VALUES (376, 'Andorra ')

INSERT INTO Calling\_Code VALUES (377, 'Monaco')

INSERT INTO Calling\_Code VALUES (378, 'San Marino')

INSERT INTO Calling\_Code VALUES (379, 'Vatican City')

INSERT INTO Calling\_Code VALUES (380, 'Ukraine')

INSERT INTO Calling\_Code VALUES (381, 'Serbia')

INSERT INTO Calling\_Code VALUES (382, 'Montenegro')

INSERT INTO Calling\_Code VALUES (385, 'Croatia')

INSERT INTO Calling\_Code VALUES (386, 'Slovenia ')

INSERT INTO Calling\_Code VALUES (387, 'Bosnia & Herzegovina ')

INSERT INTO Calling\_Code VALUES (389, 'Macedonia')

INSERT INTO Calling\_Code VALUES (420, 'Czech Republic')

INSERT INTO Calling\_Code VALUES (421, 'Slovak Republic')

INSERT INTO Calling\_Code VALUES (423, 'Liechtenstein')

INSERT INTO Calling\_Code VALUES (500, 'Falkland Islands')

INSERT INTO Calling\_Code VALUES (501, 'Belize')

INSERT INTO Calling\_Code VALUES (502, 'Guatemala ')

INSERT INTO Calling\_Code VALUES (503, 'El Salvador')

INSERT INTO Calling\_Code VALUES (504, 'Honduras')

INSERT INTO Calling\_Code VALUES (505, 'Nicaragua')

INSERT INTO Calling\_Code VALUES (506, 'Costa Rica')

INSERT INTO Calling\_Code VALUES (507, 'Panama')

INSERT INTO Calling\_Code VALUES (508, 'St. Pierre & Miquelon')

INSERT INTO Calling\_Code VALUES (509, 'Haiti ')

INSERT INTO Calling\_Code VALUES (590, 'Guadeloupe')

INSERT INTO Calling\_Code VALUES (591, 'Bolivia ')

INSERT INTO Calling\_Code VALUES (592, 'Guyana')

INSERT INTO Calling\_Code VALUES (593, 'Ecuador ')

INSERT INTO Calling\_Code VALUES (594, 'French Guiana')

INSERT INTO Calling\_Code VALUES (595, 'Paraguay')

INSERT INTO Calling\_Code VALUES (596, 'Martinique')

INSERT INTO Calling\_Code VALUES (597, 'Suriname ')

INSERT INTO Calling\_Code VALUES (598, 'Uruguay')

INSERT INTO Calling\_Code VALUES (599, 'Curaçao')

INSERT INTO Calling\_Code VALUES (670, 'East Timor')

INSERT INTO Calling\_Code VALUES (672, 'Antarctica')

INSERT INTO Calling\_Code VALUES (673, 'Brunei Darussalam')

INSERT INTO Calling\_Code VALUES (674, 'Nauru')

INSERT INTO Calling\_Code VALUES (675, 'Papua New Guinea')

INSERT INTO Calling\_Code VALUES (676, 'Tonga Islands')

INSERT INTO Calling\_Code VALUES (677, 'Solomon Islands')

INSERT INTO Calling\_Code VALUES (678, 'Vanuatu')

INSERT INTO Calling\_Code VALUES (679, 'Fiji Islands')

INSERT INTO Calling\_Code VALUES (680, 'Palau')

INSERT INTO Calling\_Code VALUES (681, 'Wallis and Futuna Islands')

INSERT INTO Calling\_Code VALUES (682, 'Cook Islands')

INSERT INTO Calling\_Code VALUES (683, 'Niue')

INSERT INTO Calling\_Code VALUES (685, 'Samoa')

INSERT INTO Calling\_Code VALUES (686, 'Kiribati ')

INSERT INTO Calling\_Code VALUES (687, 'New Caledonia')

INSERT INTO Calling\_Code VALUES (688, 'Tuvalu')

INSERT INTO Calling\_Code VALUES (689, 'French Polynesia')

INSERT INTO Calling\_Code VALUES (690, 'Tokelau')

INSERT INTO Calling\_Code VALUES (691, 'Micronesia')

INSERT INTO Calling\_Code VALUES (692, 'Marshall Islands')

INSERT INTO Calling\_Code VALUES (808, 'Wake Island')

INSERT INTO Calling\_Code VALUES (850, 'Korea (North)')

INSERT INTO Calling\_Code VALUES (852, 'Hong Kong')

INSERT INTO Calling\_Code VALUES (853, 'Macao')

INSERT INTO Calling\_Code VALUES (855, 'Cambodia')

INSERT INTO Calling\_Code VALUES (856, 'Laos')

INSERT INTO Calling\_Code VALUES (880, 'Bangladesh')

INSERT INTO Calling\_Code VALUES (886, 'Taiwan')

INSERT INTO Calling\_Code VALUES (960, 'Maldives')

INSERT INTO Calling\_Code VALUES (961, 'Lebanon')

INSERT INTO Calling\_Code VALUES (962, 'Jordan')

INSERT INTO Calling\_Code VALUES (963, 'Syria')

INSERT INTO Calling\_Code VALUES (964, 'Iraq')

INSERT INTO Calling\_Code VALUES (965, 'Kuwait ')

INSERT INTO Calling\_Code VALUES (966, 'Saudi Arabia')

INSERT INTO Calling\_Code VALUES (967, 'Yemen')

INSERT INTO Calling\_Code VALUES (968, 'Oman')

INSERT INTO Calling\_Code VALUES (971, 'United Arab Emirates')

INSERT INTO Calling\_Code VALUES (972, 'Israel ')

INSERT INTO Calling\_Code VALUES (973, 'Bahrain')

INSERT INTO Calling\_Code VALUES (974, 'Qatar')

INSERT INTO Calling\_Code VALUES (975, 'Bhutan')

INSERT INTO Calling\_Code VALUES (976, 'Mongolia ')

INSERT INTO Calling\_Code VALUES (977, 'Nepal ')

INSERT INTO Calling\_Code VALUES (992, 'Tajikistan')

INSERT INTO Calling\_Code VALUES (993, 'Turkmenistan ')

INSERT INTO Calling\_Code VALUES (994, 'Azerbaijan')

INSERT INTO Calling\_Code VALUES (995, 'Georgia')

INSERT INTO Calling\_Code VALUES (996, 'Kyrgyz Republic')

INSERT INTO Calling\_Code VALUES (998, 'Uzbekistan')

/\*\* Initialize Data of Calling\_Code \*\*/

INSERT INTO Service VALUES ('Spectra')

INSERT INTO Service VALUES ('Deluxe')

INSERT INTO Service VALUES ('VOIP')

INSERT INTO Service VALUES ('Budget')

INSERT INTO Service VALUES ('Premium')

INSERT INTO Service VALUES ('GACB')

/\*\* Initialize Peak\_Time Data \*\*/

INSERT INTO Peak\_Time VALUES ('Spectra', 1, '08:00', '17:00')

INSERT INTO Peak\_Time VALUES ('Deluxe', 1, '08:00', '17:00')

INSERT INTO Peak\_Time VALUES ('VOIP', 1, '08:00', '17:00')

INSERT INTO Peak\_Time VALUES ('Budget', 1, '08:00', '17:00')

INSERT INTO Peak\_Time VALUES ('Premium', 1, '08:00', '17:00')

INSERT INTO Peak\_Time VALUES ('Spectra', 49, '09:00', '19:00')

INSERT INTO Peak\_Time VALUES ('GACB', 49, '09:00', '19:00')

INSERT INTO Peak\_Time VALUES ('Spectra', 33, '09:00', '19:00')

INSERT INTO Peak\_Time VALUES ('GACB', 33, '09:00', '19:00')

INSERT INTO Peak\_Time VALUES ('Spectra', 45, '09:00', '19:00')

INSERT INTO Peak\_Time VALUES ('GACB', 45, '09:00', '19:00')

INSERT INTO Peak\_Time VALUES ('Spectra', 39, '09:00', '19:00')

INSERT INTO Peak\_Time VALUES ('GACB', 39, '09:00', '19:00')

INSERT INTO Peak\_Time VALUES ('Spectra', 44, '08:30', '18:00')

INSERT INTO Peak\_Time VALUES ('GACB', 44, '08:00', '18:00')

INSERT INTO Peak\_Time VALUES ('Spectra', 31, '08:30', '18:00')

INSERT INTO Peak\_Time VALUES ('GACB', 31, '08:00', '18:00')

INSERT INTO Peak\_Time VALUES ('GACB', 34, '09:00', '17:30')

INSERT INTO Peak\_Time VALUES ('GACB', 36, '09:00', '17:30')

INSERT INTO Peak\_Time VALUES ('GACB', 43, '09:00', '17:30')

/\*\* Initialize Peak\_Time Data \*\*/

INSERT INTO Sales\_Rep VALUES (23)

INSERT INTO Sales\_Rep VALUES (12)

INSERT INTO Sales\_Rep VALUES (22)

INSERT INTO Sales\_Rep VALUES (15)

INSERT INTO Sales\_Rep VALUES (18)

INSERT INTO Sales\_Rep VALUES (24)

INSERT INTO Sales\_Rep VALUES (14)

INSERT INTO Sales\_Rep VALUES (19)

INSERT INTO Sales\_Rep VALUES (17)

/\*\* END \*\*/