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

/\* Connect to Database \*/

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

USE INFO430\_group7

GO

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

/\* Create Tables \*/

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

CREATE TABLE tblSTORE (

StoreID INT IDENTITY (1,1) PRIMARY KEY,

StoreName VARCHAR(100) NOT NULL

);

GO

CREATE TABLE tblAPP\_TYPE (

AppTypeID INT IDENTITY (1,1) PRIMARY KEY,

AppTypeName VARCHAR(50) NOT NULL

);

GO

CREATE TABLE tblPUBLISHER (

PublisherID INT IDENTITY (1,1) PRIMARY KEY,

PublisherName VARCHAR(100) NOT NULL

);

GO

CREATE TABLE tblREVIEW\_TYPE (

ReviewTypeID INT IDENTITY (1,1) PRIMARY KEY,

ReviewTypeName VARCHAR(20) NOT NULL

);

GO

CREATE TABLE tblRATING (

RatingID INT IDENTITY (1,1) PRIMARY KEY,

RatingName VARCHAR(20) NOT NULL,

RatingNum INT

);

GO

CREATE TABLE tblCUSTOMER (

CustID INT IDENTITY (1,1) PRIMARY KEY,

CustFname VARCHAR(50) NOT NULL,

CustLName VARCHAR(50) NOT NULL,

CustDOB DATE NOT NULL,

CustLoginID VARCHAR(50) NOT NULL

);

GO

CREATE TABLE tblSESSION (

SessionID INT IDENTITY (1,1) PRIMARY KEY,

SessionDate DATETIME NOT NULL,

SessionBegin TIME NOT NULL,

SessionEnd TIME NOT NULL,

CustID INT FOREIGN KEY REFERENCES tblCUSTOMER(CustID) NOT NULL,

)

GO

CREATE TABLE tblREVIEW (

ReviewID INT IDENTITY (1,1) PRIMARY KEY,

ReviewDate DATE,

ReviewDescr VARCHAR(300),

ReviewTypeID INT FOREIGN KEY REFERENCES tblREVIEW\_TYPE(ReviewTypeID) NOT NULL,

SessionID INT FOREIGN KEY REFERENCES tblSESSION(SessionID) NOT NULL,

RatingID INT FOREIGN KEY REFERENCES tblRATING(RatingID) NOT NULL,

)

GO

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

/\* Populate Lookup Tables \*/

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

-- Implement part of business rule for age range 13-120 years old

-- And dupe prevention for username with addition of (CustomerID - 100)

INSERT INTO tblCUSTOMER (CustFname, CustLName, CustDOB, CustLoginID)

SELECT TOP 1000000 CustomerFname, CustomerLname, DateOfBirth,

(CONCAT(CustomerFname, CustomerLname, REPLACE(STR(CustomerID - 100), ' ', '')))

FROM PEEPS.dbo.tblCUSTOMER

WHERE YEAR(DateOfBirth) BETWEEN 1902 AND 2009

GO

INSERT INTO tblREVIEW\_TYPE

VALUES ('positive'), ('negative')

GO

INSERT INTO tblRATING

VALUES ('Very Bad', 1), ('Okay', 2), ('Neutral', 3), ('Pretty Good', 4), ('The Best', 5)

GO

-- RESEEDING PURPOSES ONLY

/\*

USE INFO430\_group7

DBCC CHECKIDENT (tblRATING, reseed, 0)

\*/

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

/\* Nested Store Procedures \*/

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

CREATE OR ALTER PROCEDURE group7\_get\_CustID

@CFN VARCHAR(50),

@CLN VARCHAR(50),

@CDOB DATE,

@CLog VARCHAR(50),

@CID INT OUTPUT

AS

SET @CID = (SELECT CustID

FROM tblCUSTOMER

WHERE CustFname = @CFN

AND CustLname = @CLN

AND CustDOB = @CDOB

AND CustLoginID = @CLog)

GO

CREATE OR ALTER PROCEDURE group7\_get\_ReviewTypeID

@RTN VARCHAR(20),

@RTID INT OUTPUT

AS

SET @RTID = (SELECT ReviewTypeID

FROM tblREVIEW\_TYPE

WHERE ReviewTypeName = @RTN)

GO

CREATE OR ALTER PROCEDURE group7\_get\_RatingID

@RatN VARCHAR(20),

@RatNum INT,

@RatID INT OUTPUT

AS

SET @RatID = (SELECT RatingID

FROM tblRATING

WHERE RatingName = @RatN

AND RatingNum = @RatNum)

GO

CREATE OR ALTER PROCEDURE group7\_get\_SessionID

@SeDate DATETIME,

@SeBg TIME,

@SeEnd TIME,

@CustFN VARCHAR(50),

@CustLN VARCHAR(50),

@CustDOB DATE,

@CustLogin VARCHAR(50),

@SeID INT OUTPUT

AS

DECLARE @CustID INT

EXEC group7\_get\_CustID

@CFN = @CustFN,

@CLN = @CustLN,

@CDOB = @CustDOB,

@CLog = @CustLogin,

@CID = @CustID OUTPUT

IF @CustID IS NULL

BEGIN

PRINT '@CustID is empty... check spelling';

THROW 54413, '@CustID cannot be NULL, process is terminated', 1;

END

SET @SeID = (SELECT SessionID

FROM tblSESSION

WHERE SessionDate = @SeDate

AND SessionBegin = @SeBg

AND SessionEnd = @SeEnd

AND CustID = @CustID)

GO

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

/\* Create a Stored Procedure to Populate tblSESSION, tblREVIEW, tblAPP\_CATEGORY \*/

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

/\* tblSESSION \*/

CREATE OR ALTER PROCEDURE group7\_add\_session

@SDate DATETIME,

@SBegin TIME,

@SEnd TIME,

@CustFName VARCHAR(50),

@CustLName VARCHAR(50),

@CustDOB DATE,

@CustLog VARCHAR(50),

@Scope\_Help INT OUTPUT

AS

DECLARE @CustomerID INT

-- Table variable to check whether a user has two different session at the same time

DECLARE @SessionDupes TABLE (PK INT PRIMARY KEY IDENTITY(1,1), BeginTime TIME, EndTime TIME)

EXEC group7\_get\_CustID

@CFN = @CustFName,

@CLN = @CustLName,

@CDOB = @CustDOB,

@CLog = @CustLog,

@CID = @CustomerID OUTPUT

IF @CustomerID IS NULL

BEGIN

PRINT '@CustomerID is empty... check spelling';

THROW 54413, '@CustomerID cannot be NULL, process is terminated', 1;

END

-- INSERTs into @SessionDupes

INSERT INTO @SessionDupes (BeginTime, EndTime)

(SELECT SessionBegin, SessionEnd FROM tblSESSION

WHERE CustID = @CustomerID AND SessionDate = @SDate)

DECLARE @Num INT = (SELECT COUNT(\*) FROM @SessionDupes)

DECLARE @Insert\_\_Valid INT = 0 -- Default TRUE / 1 --> FALSE

DECLARE @Init\_BegTime TIME, @Init\_EndTime TIME

DECLARE @Insert\_BegTime TIME, @Insert\_EndTime TIME

SET @Insert\_BegTime = @SBegin

SET @Insert\_EndTime = @SEnd

-- Check if there's an INSERT session with the same user on the same date

WHILE @Num > 0

BEGIN

SET @Init\_BegTime = (SELECT BeginTime FROM @SessionDupes WHERE PK = @Num)

SET @Init\_EndTime = (SELECT EndTime FROM @SessionDupes WHERE PK = @Num)

SET @Num = @Num - 1

IF @Insert\_BegTime >= @Init\_BegTime AND @Insert\_BegTime <= @Init\_EndTime

BEGIN

SET @Insert\_\_Valid = 1

PRINT 'BEGIN TIME INVALID; SESSION BOUNDARY CROSS';

END

IF @Insert\_BegTime < @Init\_BegTime AND @Insert\_EndTime > @Init\_BegTime

BEGIN

SET @Insert\_\_Valid = 1

PRINT 'END TIME INVALID; SESSION BOUNDARY CROSS';

END

END

-- INSERT only non-overlapping session times for same user

BEGIN TRAN T1

IF @Insert\_\_Valid = 0

BEGIN

INSERT INTO tblSESSION(SessionDate, SessionBegin, SessionEnd, CustID)

VALUES (@SDate, @SBegin, @SEnd, @CustomerID)

SET @Scope\_help = (SELECT SCOPE\_IDENTITY())

END

IF @@ERROR<> 0

BEGIN

PRINT 'INSERT to tblSession failed!!'

ROLLBACK TRANSACTION T1

END

ELSE

COMMIT TRANSACTION T1

GO

/\* tblREVIEW \*/

CREATE OR ALTER PROCEDURE group7\_add\_review

@RDate DATE,

@RDescr VARCHAR(300),

@RTypeN VARCHAR(20),

@SDate DATETIME,

@SBegin TIME,

@SEnd TIME,

@CustFName VARCHAR(50),

@CustLName VARCHAR(50),

@CustoDOB DATE,

@CustoLogin VARCHAR(50),

@RatingNum INT,

@RatingN VARCHAR(20)

AS

DECLARE @SessionID INT, @RTypeID INT, @RatingID INT

-- Create/fill Temp table to populate tblReview reviewDescr

CREATE TABLE #reviewDescr\_forFill (

reviewDescr VARCHAR(200) NOT NULL,

descrNum INT NOT NULL

)

INSERT INTO #reviewDescr\_forFill

VALUES ('It was awful and I will never play again!', 1),

('Do not know what type of game I am playing.', 2),

('Only played 1 minute and I gave up.', 3),

('Do not waste your time on this game.', 4),

('Decent. OK way to spend your time.', 5),

('What a wonderful game!', 6),

('Very entertaining and great graphics.', 7),

('Game is great on all fronts.', 8),

('This game is a winner!', 9),

('Play the game all day and all night!', 10)

EXEC group7\_get\_SessionID

@SeDate = @SDate,

@SeBg = @SBegin,

@SeEnd = @SEnd,

@CustFN = @CustFName,

@CustLN = @CustLName,

@CustDOB = @CustoDOB,

@CustLogin = @CustoLogin,

@SeID = @SessionID OUTPUT

IF @SessionID IS NULL

BEGIN

PRINT '@SessionID is empty... check spelling';

THROW 54414, '@SessionID cannot be NULL, process is terminated', 1;

END

EXEC group7\_get\_ReviewTypeID

@RTN = @RTypeN,

@RTID = @RTypeID OUTPUT

IF @RTypeID IS NULL

BEGIN

PRINT '@RTypeID is empty... check spelling';

THROW 54415, '@RTypeID cannot be NULL, process is terminated', 1;

END

EXEC group7\_get\_RatingID

@RatN = @RatingN,

@RatNum = @RatingNum,

@RatID = @RatingID OUTPUT

IF @RatingID IS NULL

BEGIN

PRINT '@RatingID is empty... check spelling';

THROW 54416, '@RatingID cannot be NULL, process is terminated', 1;

END

BEGIN TRAN T1

BEGIN TRAN T2

IF @RatingNum = 1 OR @RatingNum = 2

BEGIN

SET @RDescr = (SELECT reviewDescr FROM #reviewDescr\_forFill WHERE descrNum = FLOOR(RAND()\*(4)+1))

END

IF @RatingNum = 3

BEGIN

SET @RDescr = (SELECT reviewDescr FROM #reviewDescr\_forFill WHERE descrNum = 5)

END

IF @RatingNum = 4 OR @RatingNum = 5

BEGIN

SET @RDescr = (SELECT reviewDescr FROM #reviewDescr\_forFill WHERE descrNum = FLOOR(RAND()\*(5)+6))

END

IF @@ERROR <> 0

BEGIN

ROLLBACK TRAN T1

END

IF @RDescr IS NULL

BEGIN

PRINT 'Review description is NULL';

THROW 54422, '@RDescr cannot be NULL, process is terminated', 1;

END

COMMIT TRAN T2

-- DROP TABLE #reviewDescr\_forFill

INSERT INTO tblREVIEW(ReviewDate, ReviewDescr, ReviewTypeID, SessionID, RatingID)

VALUES (@RDate, @RDescr, @RTypeID, @SessionID, @RatingID)

IF @@ERROR <> 0

BEGIN

PRINT 'INSERT FAIL FOR tblREVIEW'

ROLLBACK TRAN T1

END

ELSE

COMMIT TRAN T1

GO

CREATE OR ALTER PROCEDURE group7\_add\_app\_category

@AName VARCHAR(150),

@ADate DATE,

@CName VARCHAR(150)

AS

DECLARE @AppID INT, @CategoryID INT

EXEC group7\_get\_App

@A\_Name = @AName,

@A\_RD = @ADate,

@A\_ID = @AppID OUTPUT

IF @AppID IS NULL

BEGIN

PRINT '@AppID is empty... check spelling';

THROW 54417, '@AppID cannot be NULL, process is terminated', 1;

END

EXEC group7\_get\_CatID

@CT\_Name = @CName,

@CT\_ID = @CategoryID OUTPUT

IF @CategoryID IS NULL

BEGIN

PRINT '@CategoryID is empty... check spelling';

THROW 54418, '@CategoryID cannot be NULL, process is terminated', 1;

END

BEGIN TRAN T1

INSERT INTO tblAPP\_CATEGORY(AppID, CatID)

VALUES (@AppID, @CategoryID)

IF @@ERROR <> 0

BEGIN

PRINT 'INSERT FAIL FOR tblREVIEW'

ROLLBACK TRAN T1

END

ELSE

COMMIT TRAN T1

GO

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

/\* Create Synthetic Transaction \*/

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

/\* INSERT tblSESSION, tblAPP\_PLATORM \*/

CREATE OR ALTER PROCEDURE wrapper\_group7\_add\_sess\_app\_plat

@Run INT

AS

DECLARE @SessDate DATETIME, @SessBegin TIME, @SessEnd TIME, @CFName VARCHAR(50), @CLName VARCHAR(50), @CDOB2 DATE, @CLogin VARCHAR(50),

@AppName VARCHAR(150), @AppDate DATE, @PlatName VARCHAR(100), @Sess\_ID INT

DECLARE @CustRowCount INT = (SELECT COUNT(\*) FROM tblCUSTOMER)

DECLARE @AppRowCount INT = (SELECT COUNT(\*) FROM tblAPP)

DECLARE @PlatRowCount INT = (SELECT COUNT(\*) FROM tblPLATFORM)

DECLARE @CustPK INT, @AppPK INT, @PlatPK INT

DECLARE @RAND INT

WHILE @Run > 0

BEGIN

SET @CustPK = (SELECT RAND() \* @CustRowCount + 1)

SET @AppPK = (SELECT RAND() \* @AppRowCount + 1)

SET @PlatPK = (SELECT RAND() \* @PlatRowCount + 1)

SET @CFName = (SELECT CustFname FROM tblCUSTOMER WHERE CustID = @CustPK)

SET @CLName = (SELECT CustLname FROM tblCUSTOMER WHERE CustID = @CustPK)

SET @CDOB2 = (SELECT CustDOB FROM tblCUSTOMER WHERE CustID = @CustPK)

SET @CLogin = (SELECT CustLoginID FROM tblCUSTOMER WHERE CustID = @CustPK)

SET @SessDate = (SELECT GETDATE() - (SELECT RAND() \* 1000 + 1))

SET @SessBegin = (SELECT CONVERT(time, @SessDate))

SET @SessEnd = (SELECT DateAdd(MINUTE, (SELECT RAND() \* 2000 + 1), @SessBegin)) -- Look into DateAdd

SET @AppName = (SELECT AppName FROM tblAPP WHERE AppID = @AppPK)

SET @AppDate = (SELECT ReleaseDate FROM tblAPP WHERE AppID = @AppPK)

SET @PlatName = (SELECT PlatName FROM tblPLATFORM WHERE PlatID = @PlatPK)

BEGIN TRAN T1

EXEC group7\_add\_session

@SDate = @SessDate,

@SBegin = @SessBegin,

@SEnd = @SessEnd,

@CustFName = @CFName,

@CustLName = @CLName,

@CustDOB = @CDOB2,

@CustLog = @CLogin,

@Scope\_Help = @Sess\_ID OUTPUT

IF @@ERROR <> 0

BEGIN

ROLLBACK TRAN T1

END

IF @Sess\_ID IS NULL

BEGIN

PRINT '@Sess\_ID is empty... check spelling';

THROW 54777, '@Sess\_ID cannot be NULL, process is terminated', 3

END

EXEC group7\_insert\_app\_platform

@App\_N = @AppName,

@App\_RD = @AppDate,

@Plat\_Name = @PlatName,

@S\_Identity = @Sess\_ID

IF @@ERROR <> 0

BEGIN

PRINT 'INSERT FAIL FOR tblSESSION and tblApp\_Platform'

ROLLBACK TRAN T1

END

ELSE

COMMIT TRAN T1

SET @Run = @Run - 1

END

GO

-- Execute wrapper

EXEC wrapper\_group7\_add\_sess\_app\_plat 500

GO

/\* INSERT tblREVIEW Wrapper \*/

CREATE OR ALTER PROCEDURE wrapper\_group7\_add\_review

@Run INT

AS

DECLARE @RevDate DATE, @RevDescr VARCHAR(300), @RevTypeName VARCHAR(20), @SessionDate DATETIME, @SessionBegin TIME, @SessionEnd TIME, @CustFirstName VARCHAR(50),

@CustLastName VARCHAR(50), @CustDOB2 DATE, @CustoLog VARCHAR(50), @RatingNumber INT, @RatingName VARCHAR(20)

DECLARE @SessRowCount INT = (SELECT COUNT(\*) FROM tblSESSION)

DECLARE @RatingRowCount INT = (SELECT COUNT(\*) FROM tblRATING)

-- Establish range of sessionDate and GETDATE for Review date

DECLARE @Range INT

DECLARE @SessPK INT, @RatPK INT

DECLARE @RAND INT

WHILE @Run > 0

BEGIN

SET @SessPK = (SELECT RAND() \* @SessRowCount + 1)

SET @RatPK = (SELECT RAND() \* @RatingRowCount + 1)

SET @RatingNumber = (SELECT RatingNum FROM tblRATING WHERE RatingID = @RatPK)

IF @RatingNumber >= 3

BEGIN

SET @RevTypeName = (SELECT ReviewTypeName FROM tblREVIEW\_TYPE WHERE ReviewTypeID = 1)

END

ELSE

SET @RevTypeName = (SELECT ReviewTypeName FROM tblREVIEW\_TYPE WHERE ReviewTypeID = 2)

SET @RatingName = (SELECT RatingName FROM tblRATING WHERE RatingID = @RatPK)

SET @SessionDate = (SELECT SessionDate FROM tblSESSION WHERE SessionID = @SessPK)

IF @SessionDate IS NULL

BEGIN

PRINT '@SessionDate NULL';

THROW 54444, 'Wrapper will terminate becasue @SessionDate NULL', 1;

END

SET @SessionBegin = (SELECT SessionBegin FROM tblSESSION WHERE SessionID = @SessPK)

SET @SessionEnd = (SELECT SessionEnd FROM tblSESSION WHERE SessionID = @SessPK)

SET @CustFirstName = (SELECT CustFname

FROM tblCUSTOMER C

JOIN tblSESSION S ON C.CustID = S.CustID

WHERE SessionID = @SessPK)

IF @CustFirstName IS NULL

BEGIN

PRINT '@CustFirstName NULL';

THROW 54448, 'Wrapper will terminate becasue @CustFirstName NULL', 1;

END

SET @CustLastName = (SELECT CustLname

FROM tblCUSTOMER C

JOIN tblSESSION S ON C.CustID = S.CustID

WHERE SessionID = @SessPK)

SET @CustDOB2 = (SELECT CustDOB

FROM tblCUSTOMER C

JOIN tblSESSION S ON C.CustID = S.CustID

WHERE SessionID = @SessPK)

SET @CustoLog = (SELECT CustLoginID

FROM tblCUSTOMER C

JOIN tblSESSION S ON C.CustID = S.CustID

WHERE SessionID = @SessPK)

SET @Range = (SELECT DATEDIFF(DAY, @SessionDate, GETDATE()))

SET @RevDate = (SELECT DATEADD(DAY, (SELECT RAND() \* @RANGE + 1), @SessionDate))

EXEC group7\_add\_review

@RDate = @RevDate,

@RDescr = @RevDescr,

@RTypeN = @RevTypeName,

@SDate = @SessionDate,

@SBegin = @SessionBegin,

@SEnd = @SessionEnd,

@CustFName = @CustFirstName,

@CustLName = @CustLastName,

@CustoDOB = @CustDOB2,

@CustoLogin = @CustoLog,

@RatingNum = @RatingNumber,

@RatingN = @RatingName

SET @Run = @Run - 1

END

-- Execute wrapper

EXEC wrapper\_group7\_add\_review 10000

GO