CREATE TABLE tbl\_Login

(

Id INT PRIMARY KEY IDENTITY(1000,1),

Name VARCHAR(50),

Email VARCHAR(50),

Gender CHAR(1) CHECK (Gender = 'M' OR Gender = 'F'),

UserName VARCHAR(50),

Password VARCHAR(100)

)

INSERT INTO tbl\_Login VALUES('Sumanth Gunda','sgnw8@mail.umkc.edu','M','sumanthgundaa','sum'),('Goutham Donthu','gd6d9@mail.umkc.edu','M','gouthamdonthu','gou'),('Rajesh Jonnalagadda','rj9rb@mail.umkc.edu','M','rajeshjonnalagadda','raj'),('Ravi Teja Yakkala','rybp3@mail.umkc.edu','M','ravitejayakkala','rav')

SELECT \* FROM tbl\_Login

DROP TABLE tbl\_Login

CREATE PROCEDURE usp\_Login(@UserName VARCHAR(50),@Password VARCHAR(50))

AS

BEGIN

BEGIN TRY

IF EXISTS(SELECT UserName FROM tbl\_Login WHERE UserName = @UserName)

BEGIN

IF EXISTS(SELECT UserName,Password FROM tbl\_Login WHERE UserName = @UserName AND Password = @Password)

BEGIN

RETURN 1

END

ELSE

BEGIN

RETURN 2

END

END

ELSE

BEGIN

RETURN 3

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE TABLE tbl\_Elements

(

ElementId INT PRIMARY KEY IDENTITY(1,1),

ElementName VARCHAR(50),

ElementDisplayName VARCHAR(50),

Tool VARCHAR(50),

TimeZone CHAR(2),

Active INT,

LastUpdatedGMT DATETIME,

UpdatedBy VARCHAR(50)

)

DROP TABLE tbl\_Elements

SELECT \* FROM tbl\_Elements

INSERT INTO tbl\_Elements VALUES('ATLNGAUS-MME-01','ATL-MME-01','FirstAlertVendor1','ET',1,'9/4/2014 20:48:12','sumanthgundaa'),

('ATLNGAUS-MME-02','ATL-MME-02','FirstAlertVendor1','ET',1,'9/4/2014 20:48:12','sumanthgundaa'),

('ATLNGAUS-MME-03','ATL-MME-03','FirstAlertVendor1','ET',1,'9/4/2014 20:48:12','sumanthgundaa'),

('KSCYMOEC-MME-01','KSC-MME-01','FirstAlertVendor1','CT',1,'9/4/2014 20:48:12','sumanthgundaa'),

('ATLNGAUS-MME-02','ATL-MME-02','FirstAlertVendor2','ET',1,'9/4/2014 20:48:12','sumanthgundaa'),

('KSCYMOEC-MME-01','KSC-MME-01','FirstAlertVendor2','CT',1,'9/4/2014 20:48:12','sumanthgundaa')

CREATE PROCEDURE usp\_AddElement(@ElementName VARCHAR(50),@ElementDisplayName VARCHAR(50),@Tool VARCHAR(50),@TimeZone CHAR(2),@UpdatedBy VARCHAR(50))

AS

BEGIN

DECLARE @Active INT

SET @Active = 1

DECLARE @LastUpdatedGMT DATETIME

SET @LastUpdatedGMT = GETDATE()

BEGIN TRY

IF EXISTS(SELECT ElementName FROM tbl\_Elements WHERE ElementName = @ElementName AND ElementDisplayName = @ElementDisplayName AND Tool = @Tool)

BEGIN

RETURN 1

END

ELSE

BEGIN

INSERT INTO tbl\_Elements VALUES(@ElementName,@ElementDisplayName,@Tool,@TimeZone,@Active,@LastUpdatedGMT,@UpdatedBy)

RETURN 2

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE PROCEDURE usp\_DisableElement(@ElementId INT)

AS

BEGIN

BEGIN TRY

DECLARE @Active INT

SET @Active = 0

IF(@Active = (SELECT Active FROM tbl\_Elements WHERE ElementId = @ElementId))

BEGIN

RETURN 1

END

ELSE

BEGIN

UPDATE tbl\_Elements SET Active = 0 WHERE ElementId = @ElementId

RETURN 2

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE PROCEDURE usp\_UpdateElement(@ElementId INT,@ElementName VARCHAR(50),@ElementDisplayName VARCHAR(50),@TimeZone CHAR(2),@UpdatedBy VARCHAR(50))

AS

BEGIN

DECLARE @LastUpdatedGMT DATETIME

SET @LastUpdatedGMT = GETDATE()

BEGIN TRY

IF EXISTS(SELECT ElementName FROM tbl\_Elements WHERE ElementName = @ElementName AND ElementDisplayName = @ElementDisplayName AND ElementId = @ElementId)

BEGIN

RETURN 1

END

ELSE

BEGIN

UPDATE tbl\_Elements SET ElementName = @ElementName WHERE ElementId = @ElementId

UPDATE tbl\_Elements SET ElementDisplayName = @ElementDisplayName WHERE ElementId = @ElementId

UPDATE tbl\_Elements SET TimeZone = @TimeZone WHERE ElementId = @ElementId

UPDATE tbl\_Elements SET LastUpdatedGMT = @LastUpdatedGMT WHERE ElementId = @ElementId

UPDATE tbl\_Elements SET UpdatedBy = @UpdatedBy WHERE ElementId = @ElementId

RETURN 2

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE TABLE tbl\_Schedule

(

ScheduleId INT PRIMARY KEY IDENTITY(1,1),

ScheduleName VARCHAR(50),

StartHour INT,

EndHour INT,

Active INT,

LastUpdatedGMT DATETIME,

UpdatedBy VARCHAR(50)

)

SELECT \* FROM tbl\_Schedule

DROP TABLE tbl\_Schedule

ALTER PROCEDURE usp\_AddSchedule(@ScheduleName VARCHAR(50),@StartHour INT,@EndHour INT,@UpdatedBy VARCHAR(50))

AS

BEGIN

DECLARE @LastUpdatedGMT DATETIME

SET @LastUpdatedGMT = GETDATE()

DECLARE @Count INT

DECLARE @SHour INT

DECLARE @EHour INT

DECLARE @Active INT

BEGIN TRY

SET @Active = 1

SET @Count = (SELECT Count(\*) FROM tbl\_Schedule)

IF(@Count = 0)

BEGIN

INSERT INTO tbl\_Schedule VALUES(@ScheduleName,@StartHour,@EndHour,@Active,@LastUpdatedGMT,@UpdatedBy)

RETURN 1

END

ELSE

BEGIN

IF(@Count = 1)

BEGIN

SET @SHour = (SELECT StartHour FROM tbl\_Schedule)

SET @EHour = (SELECT EndHour FROM tbl\_Schedule)

IF(@SHour < @Ehour)

BEGIN

IF(@StartHour > @SHour AND @EndHour < @EHour AND @EHour > @StartHour)

BEGIN

RETURN -1

END

ELSE

BEGIN

INSERT INTO tbl\_Schedule VALUES(@ScheduleName,@StartHour,@EndHour,@Active,@LastUpdatedGMT,@UpdatedBy)

RETURN 1

END

END

ELSE IF(@Shour > @Ehour)

BEGIN

IF(@StartHour > @SHour AND @EndHour > @EHour)

BEGIN

RETURN -1

END

ELSE IF(@StartHour > @SHour AND @EndHour < @EHour)

BEGIN

RETURN -1

END

ELSE

BEGIN

INSERT INTO tbl\_Schedule VALUES(@ScheduleName,@StartHour,@EndHour,@Active,@LastUpdatedGMT,@UpdatedBy)

RETURN 1

END

END

END

ELSE IF(@Count > 1)

BEGIN

INSERT INTO tbl\_Schedule VALUES(@ScheduleName,@StartHour,@EndHour,@Active,@LastUpdatedGMT,@UpdatedBy)

RETURN 1

END

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE PROCEDURE usp\_DisableSchedule(@ScheduleId INT)

AS

BEGIN

BEGIN TRY

DECLARE @Active INT

SET @Active = 0

IF(@Active = (SELECT Active FROM tbl\_Schedule WHERE ScheduleId = @ScheduleId))

BEGIN

RETURN 1

END

ELSE

BEGIN

UPDATE tbl\_Schedule SET Active = 0 WHERE ScheduleId = @ScheduleId

RETURN 2

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE PROCEDURE usp\_UpdateSchedule(@ScheduleId INT,@ScheduleName VARCHAR(50),@StartHour INT,@EndHour INT,@UpdatedBy VARCHAR(50))

AS

BEGIN

DECLARE @LastUpdatedGMT DATETIME

SET @LastUpdatedGMT = GETDATE()

BEGIN TRY

IF EXISTS(SELECT ScheduleName FROM tbl\_Schedule WHERE ScheduleName = @ScheduleName AND StartHour = @StartHour AND EndHour = @EndHour AND ScheduleId = @ScheduleId)

BEGIN

RETURN 1

END

ELSE

BEGIN

UPDATE tbl\_Schedule SET ScheduleName = @ScheduleName WHERE ScheduleId = @ScheduleId

UPDATE tbl\_Schedule SET StartHour = @StartHour WHERE ScheduleId = @ScheduleId

UPDATE tbl\_Schedule SET EndHour = @EndHour WHERE ScheduleId = @ScheduleId

UPDATE tbl\_Schedule SET LastUpdatedGMT = @LastUpdatedGMT WHERE ScheduleId = @ScheduleId

UPDATE tbl\_Schedule SET UpdatedBy = @UpdatedBy WHERE ScheduleId = @ScheduleId

RETURN 2

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE TABLE tbl\_KPI

(

KPIId INT,

ScheduleId INT CONSTRAINT fk5 FOREIGN KEY REFERENCES tbl\_Schedule(ScheduleId),

DarkGreenValue INT,

GreenValue INT,

YellowValue INT,

OrangeValue INT,

RedValue INT,

KPIName VARCHAR(50),

OrderName CHAR(3),

Tool VARCHAR(50),

Active INT,

LastUpdatedGMT DATETIME,

UpdatedBy VARCHAR(50),

CONSTRAINT ck\_pkkey PRIMARY KEY(KPIId,ScheduleId)

)

DROP TABLE tbl\_KPI

SELECT \* FROM tbl\_KPI

CREATE PROCEDURE usp\_AddKPI(@KPIId INT,@ScheduleId INT,@DarkGreenValue INT,@GreenValue INT,@YellowValue INT,@OrangeValue INT,@RedValue INT,@KpiName VARCHAR(50),@OrderName CHAR(3),@Tool VARCHAR(50),@UpdatedBy VARCHAR(50))

AS

BEGIN

DECLARE @Active INT

SET @Active = 1

DECLARE @LastUpdatedGMT DATETIME

SET @LastUpdatedGMT = GETDATE()

BEGIN TRY

IF EXISTS(SELECT \* FROM tbl\_KPI WHERE KPIId = @KPIId AND ScheduleId = @ScheduleId)

BEGIN

RETURN 1

END

ELSE

BEGIN

INSERT INTO tbl\_KPI VALUES(@KPIId,@ScheduleId,@DarkGreenValue,@GreenValue,@YellowValue,@OrangeValue,@RedValue,@KpiName,@OrderName,@Tool,@Active,@LastUpdatedGMT,@UpdatedBy)

RETURN 2

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE PROCEDURE usp\_DisableKPI(@KPIId INT,@ScheduleId INT)

AS

BEGIN

BEGIN TRY

DECLARE @Active INT

SET @Active = 0

IF(@Active = (SELECT Active FROM tbl\_KPI WHERE KPIId = @KPIId AND ScheduleId = @ScheduleId))

BEGIN

RETURN 1

END

ELSE

BEGIN

UPDATE tbl\_KPI SET Active = 0 WHERE KPIId = @KPIId AND ScheduleId = @ScheduleId

RETURN 2

END

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END

CREATE PROCEDURE usp\_UpdateKPI(@KPIId INT,@DarkGreenValue INT,@GreenValue INT,@YellowValue INT,@OrangeValue INT,@RedValue INT,@KpiName VARCHAR(50),@OrderName CHAR(3),@Tool VARCHAR(50),@UpdatedBy VARCHAR(50))

AS

BEGIN

DECLARE @LastUpdatedGMT DATETIME

SET @LastUpdatedGMT = GETDATE()

BEGIN TRY

UPDATE tbl\_KPI SET DarkGreenValue = @DarkGreenValue WHERE KPIId = @KPIId

UPDATE tbl\_KPI SET GreenValue = @GreenValue WHERE KPIId = @KPIId

UPDATE tbl\_KPI SET YellowValue = @YellowValue WHERE KPIId = @KPIId

UPDATE tbl\_KPI SET OrangeValue = @OrangeValue WHERE KPIId = @KPIId

UPDATE tbl\_KPI SET RedValue = @RedValue WHERE KPIId = @KPIId

UPDATE tbl\_KPI SET KPIName = @KpiName WHERE KPIId = @KPIId

UPDATE tbl\_KPI SET OrderName = @OrderName WHERE KPIId = @KPIId

UPDATE tbl\_KPI SET Tool = @Tool WHERE KPIId = @KPIId

UPDATE tbl\_KPI SET LastUpdatedGMT = @LastUpdatedGMT WHERE KPIId = @KPIId

UPDATE tbl\_KPI SET UpdatedBy = @UpdatedBy WHERE KPIId = @KPIId

RETURN 2

END TRY

BEGIN CATCH

RETURN -99

END CATCH

END