

```
USE sp21_3342_tun49199
```

```
GO
```

```
/*
```

```
Description: Here are the SQL scripts to create
the calendar database with a starter set of data
```

Changes Made:

```
JamesS   - 2021-04-05 - Original code.
JamesS   - 2021-04-09 - Add stored procedure TP_Account_Update_Password_SP
                        Add stored procedure TP_Account_Update_Password_SP
JamesS   - 2021-04-11 - Add TP_Calendar_Select_SP
                        TP_Calendar_Insert_SP
                        TP_Calendar_Update_SP
                        TP_Calendar_Delete_SP
JamesS   - 2021-04-13 - Add TP_Event_Select_SP
                        TP_Event_Insert_SP
                        TP_Event_Update_SP
                        TP_Event_Delete_SP
FrankP   - 2021-04-25 - Add stored procedures
                        TP_SecurityQuestion_SP
                        TP_GetAccountId_SP
JamesS   - 2021-05-31 - Add TP_Calendar.IsChecked & associated code updates
                        Add TP_Calendar_Update_IsChecked_SP stored procedure
JamesS   - 2021-06-24 - Update TP_Account table
                        add SecurityPIN column
                        add Active column
                        Add TP_Account_SecurityPIN_SP stored procedure
```

Testing Scripts:

```
select * from dbo.TP_Account
select * from dbo.TP_SecurityQuestion
select * from dbo.TP_Account a
        JOIN dbo.TP_SecurityQuestion s on s.AccountId = a.AccountId
```

```
select * from dbo.TP_SampleDocument
```

```
select * from dbo.TP_Calendar
select * from dbo.TP_Event order by begdatetime
```

```
dbo.TP_Calendar_Delete_SP
        @CreatedEmailAddress='jims@temple.edu',
        @Name='Holidays'
```

```
*/
```

```
/******
```

```
* Drop constraints to be recreated below
```

```
*****/
```

```
IF (SELECT object_id('TP_Account_PK')) is not null
ALTER TABLE dbo.TP_Account DROP CONSTRAINT TP_Account_PK;
```

```
IF (SELECT object_id('TP_CreatedEmailAddress_UN')) is not null
```

```
ALTER TABLE dbo.TP_Account DROP CONSTRAINT TP_CreatedEmailAddress_UN;

IF (SELECT object_id('TP_AccountDocument_PK')) is not null
ALTER TABLE dbo.TP_AccountDocument DROP CONSTRAINT TP_AccountDocument_PK;

IF (SELECT object_id('TP_SecurityQuestion_PK')) is not null
ALTER TABLE dbo.TP_SecurityQuestion DROP CONSTRAINT TP_SecurityQuestion_PK;

IF (SELECT object_id('TP_Event_PK')) is not null
ALTER TABLE dbo.TP_Event DROP CONSTRAINT TP_Event_PK;

IF (SELECT object_id('TP_Event_UN')) is not null
ALTER TABLE dbo.TP_Event DROP CONSTRAINT TP_Event_UN;

IF (SELECT object_id('TP_EventDocument_PK')) is not null
ALTER TABLE dbo.TP_EventDocument DROP CONSTRAINT TP_EventDocument_PK;

IF (SELECT object_id('TP_Calendar_PK')) is not null
ALTER TABLE dbo.TP_Calendar DROP CONSTRAINT TP_Calendar_PK;

IF (SELECT object_id('TP_Calendar_UN')) is not null
ALTER TABLE dbo.TP_Calendar DROP CONSTRAINT TP_Calendar_UN;

IF (SELECT object_id('TP_SampleDocument_PK')) is not null
ALTER TABLE dbo.TP_SampleDocument DROP CONSTRAINT TP_SampleDocument_PK;
GO
```

```
/*
* Drop tables to be recreated below with starter data
*/
IF (SELECT object_id('dbo.TP_AccountDocument')) is not null
DROP TABLE dbo.TP_AccountDocument;

IF (SELECT object_id('dbo.TP_Account')) is not null
DROP TABLE dbo.TP_Account;

IF (SELECT object_id('dbo.TP_SecurityQuestion')) is not null
DROP TABLE dbo.TP_SecurityQuestion;

IF (SELECT object_id('dbo.TP_Calendar')) is not null
DROP TABLE dbo.TP_Calendar;

IF (SELECT object_id('dbo.TP_Event')) is not null
DROP TABLE dbo.TP_Event;

IF (SELECT object_id('dbo.TP_EventDocument')) is not null
DROP TABLE dbo.TP_EventDocument;

IF (SELECT object_id('dbo.TP_SampleDocument')) is not null
DROP TABLE dbo.TP_SampleDocument;
GO
```

```

/*****
*   Create Sample Documents incl. DriverLicense, Biometric samples etc...
*****/
CREATE TABLE dbo.TP_SampleDocument (
    SampleDocumentId INT IDENTITY(1,1),
    DocumentType VARCHAR(254) NULL, -- 256 = 254 + 2 bytes for length
    DocumentUrl VARCHAR(254) NOT NULL,
    DateTimeStamp DATETIME DEFAULT GETDATE(),
    CONSTRAINT TP_SampleDocument_PK PRIMARY KEY CLUSTERED (SampleDocumentId)
);
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

/*****
*   Create AccountDocument table
*       DocumentType allowable values
*           BusPass
*           COVID19VaccRecord
*           DriverLicense
*****/
CREATE TABLE dbo.TP_AccountDocument (
    AccountDocumentId INT IDENTITY(1,1),
    AccountId INT,
    DocumentType VARCHAR(254) NOT NULL, -- 256 = 254 + 2 bytes for length
    DocumentUrl VARCHAR(254),
    DateTimeStamp DATETIME DEFAULT GETDATE(),
    CONSTRAINT TP_AccountDocument_PK PRIMARY KEY CLUSTERED (AccountDocumentId)
);
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

/*****
*   Create TP_Account table
*****/
CREATE TABLE dbo.TP_Account (
    AccountId INT IDENTITY(1,1),
    AccountRoleType VARCHAR(14) NOT NULL, --User or Administrator
    UserName VARCHAR(50),
    UserAddress VARCHAR(254),
    PhoneNumber VARCHAR(50),
    CreatedEmailAddress VARCHAR(254),
    ContactEmailAddress VARCHAR(254),
    Avatar INT,
    AccountPassword VARBINARY(MAX),
    Active VARCHAR(6), --YES/NO
    SecurityPIN INT,
    DateTimeStamp DATETIME DEFAULT GETDATE(),

```

```

        CONSTRAINT TP_Account_PK PRIMARY KEY CLUSTERED (AccountId),
        CONSTRAINT TP_CreatedEmailAddress_UN UNIQUE(CreatedEmailAddress)
    );
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

/*****
*   Create SecutityQuestion table
*
*   QuestionType allowable values
*       City   'In what town or city was your first full time job?'
*       Phone  'What were the last four digits of your childhood telephone number?'
*       School 'What primary school did you attend?'
*****/
CREATE TABLE dbo.TP_SecurityQuestion (
    SecurityQuestionId INT IDENTITY(1,1),
    AccountId INT NOT NULL,
    Question VARCHAR(254) NOT NULL,
    QuestionType VARCHAR(14) NOT NULL,
    Response VARCHAR(254),
    DateTimeStamp DATETIME DEFAULT GETDATE(),
    CONSTRAINT TP_SecurityQuestion_PK PRIMARY KEY CLUSTERED (SecurityQuestionId)
);
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

/*****
*   Create Appointment table
*****/
CREATE TABLE dbo.TP_Event (
    EventId INT IDENTITY(1,1),
    AccountId INT NOT NULL,
    CalendarId INT NOT NULL,
    Title VARCHAR(254) NOT NULL,
    BegDateTime DATETIME NOT NULL,
    EndDateTime DATETIME NOT NULL,
    AttachmentDocumentUrl VARCHAR(254),
    Location VARCHAR(254),
    Description VARCHAR(4094),
    Color VARCHAR(22),
    DateTimeStamp DATETIME DEFAULT GETDATE(),
    CONSTRAINT TP_Event_PK PRIMARY KEY CLUSTERED (EventId),
    CONSTRAINT TP_Event_UN UNIQUE(AccountId, CalendarId, BegDateTime)
);
GO
SET ANSI_NULLS ON
GO

```

```

SET QUOTED_IDENTIFIER ON
GO
/*****
*   Create EventDocument table
*****/
CREATE TABLE dbo.TP_EventDocument (
    EventDocumentId INT IDENTITY(1,1),
    AccountId INT,
    DocumentType VARCHAR(254) NOT NULL, -- 256 = 254 + 2 bytes for length
    DocumentUrl VARCHAR(254),
    DateTimeStamp DATETIME DEFAULT GETDATE(),
    CONSTRAINT TP_EventDocument_PK PRIMARY KEY CLUSTERED (EventDocumentId)
);
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
*   Create Calendar table
*       RequiredDoc
*           NULL/yes/no
*       RequiredDocType
*           If RequiredDoc = 'yes' see TP_AccountDocument.DocumentType
*****/
CREATE TABLE dbo.TP_Calendar (
    CalendarId INT IDENTITY(1,1),
    AccountId INT NOT NULL,
    Name VARCHAR(254) NOT NULL,
    --Future feature (require documentation to schedule an event
    --RequiredDoc VARCHAR(3) NULL,
    --RequiredDocType VARCHAR(254) NULL,
    Location VARCHAR(254),
    Description VARCHAR(4094),
    Color VARCHAR(22),
    IsChecked CHAR(1), -- y/n yes or no
    DateTimeStamp DATETIME DEFAULT GETDATE(),
    CONSTRAINT TP_Calendar_PK PRIMARY KEY CLUSTERED (CalendarId),
    CONSTRAINT TP_Calendar_UN UNIQUE(AccountId, Name)
);
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
*   Description: Procedure to LOGIN.
*
*   Parameters that must be passed in:
*       @CreatedEmailAddress
*       @AccountPassword
*
*****/

```

```

*****/
IF (SELECT object_id('dbo.TP_Account_Login_SP')) is not null
DROP PROCEDURE dbo.TP_Account_Login_SP;
GO
CREATE PROCEDURE dbo.TP_Account_Login_SP
    @CreatedEmailAddress VARCHAR(254)='',
    @AccountPassword VARBINARY(MAX)
AS
BEGIN
    SELECT
        Account.AccountId,
        Account.UserName,
        Account.UserAddress,
        Account.PhoneNumber,
        Account.CreatedEmailAddress,
        Account.ContactEmailAddress,
        Account.Avatar,
        Account.AccountPassword,
        Account.DateTimeStamp,
        Account.AccountRoleType
    FROM dbo.TP_Account Account
    WHERE CreatedEmailAddress = @CreatedEmailAddress
        AND AccountPassword = @AccountPassword;
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
*   Description: Procedure to answer security questions.
*   Return the number of correct responses.
*
*   Parameters that must be passed in:
*       @CreatedEmailAddress
*       @ResponseCity
*       @ResponsePhone
*       @ResponseSchool
*
*****/
IF (SELECT object_id('dbo.TP_Account_Security_Questions_SP')) is not null
DROP PROCEDURE dbo.TP_Account_Security_Questions_SP;
GO
CREATE PROCEDURE dbo.TP_Account_Security_Questions_SP
    @CreatedEmailAddress VARCHAR(254)='',
    @ResponseCity VARCHAR(254),
    @ResponsePhone VARCHAR(254),
    @ResponseSchool VARCHAR(254)
AS
BEGIN
    DECLARE @MatchCount INT
    SET @MatchCount = 0

```

```

SELECT @MatchCount = @MatchCount + 1
FROM dbo.TP_Account a
JOIN dbo.TP_SecurityQuestion s on s.AccountId = a.AccountId
WHERE a.CreatedEmailAddress = @CreatedEmailAddress
AND s.Response = @ResponseCity
AND s.QuestionType = 'City'

SELECT @MatchCount = @MatchCount + 1
FROM dbo.TP_Account a
JOIN dbo.TP_SecurityQuestion s on s.AccountId = a.AccountId
WHERE a.CreatedEmailAddress = @CreatedEmailAddress
AND s.Response = @ResponsePhone
AND s.QuestionType = 'Phone'

SELECT @MatchCount = @MatchCount + 1
FROM dbo.TP_Account a
JOIN dbo.TP_SecurityQuestion s on s.AccountId = a.AccountId
WHERE a.CreatedEmailAddress = @CreatedEmailAddress
AND s.Response = @ResponseSchool
AND s.QuestionType = 'School'

SELECT @MatchCount as NumOfCorrectResponses
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
*   Description: Procedure to UPDATE password in the TP_Account table.
*****/
IF (SELECT object_id('dbo.TP_Account_Update_Password_SP')) is not null
DROP PROCEDURE dbo.TP_Account_Update_Password_SP;
GO
CREATE PROCEDURE dbo.TP_Account_Update_Password_SP
    @CreatedEmailAddress VARCHAR(254),
    @AccountPassword VARBINARY(MAX)
AS
BEGIN TRANSACTION
BEGIN
    UPDATE dbo.TP_Account
    SET TP_Account.AccountPassword = @AccountPassword
    WHERE TP_Account.CreatedEmailAddress = @CreatedEmailAddress
    IF @@ERROR = -1 GOTO On_Account_Update_Password_Error

    COMMIT TRANSACTION
END
RETURN 0
On_Account_Update_Password_Error:
    ROLLBACK TRANSACTION
    RETURN -1
GO

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

/*****
 *   Description: Procedure to UPDATE SecurityPIN and Active
 *               columns in the TP_Account table.
 *****/
IF (SELECT object_id('dbo.TP_Account_SecurityPIN_SP')) is not null
DROP PROCEDURE dbo.TP_Account_SecurityPIN_SP;
GO
CREATE PROCEDURE dbo.TP_Account_SecurityPIN_SP
    @CreatedEmailAddress VARCHAR(254),
    @SecurityPIN INT
AS
BEGIN TRANSACTION
BEGIN
    UPDATE dbo.TP_Account
    SET TP_Account.Active = 'yes',
        TP_Account.SecurityPIN = null
    WHERE TP_Account.CreatedEmailAddress = @CreatedEmailAddress
    AND TP_Account.SecurityPIN = @SecurityPIN
    SELECT @@ROWCOUNT

    IF @@ROWCOUNT = 0 GOTO On_Account_SecurityPIN_Error

    COMMIT TRANSACTION
END
RETURN 0
On_Account_SecurityPIN_Error:
    ROLLBACK TRANSACTION
    RETURN -1
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

/*****
 *   Description: Procedure to INSERT a row into the TP_Account table.
 *   Note: An Account can be initially created without AccountDocuments
 *         which can be added later. They are optional depending on the
 *         calendar where the appointments is added.
 *****/
IF (SELECT object_id('dbo.TP_Account_Insert_SP')) is not null
DROP PROCEDURE dbo.TP_Account_Insert_SP;
GO
CREATE PROCEDURE dbo.TP_Account_Insert_SP
    @AccountRoleType VARCHAR(14) = '',
    @UserName VARCHAR(50) = '',
    @UserAddress VARCHAR(254) = ''

```



```

@PhoneNumber VARCHAR(50)='',
@CreatedEmailAddress VARCHAR(254)='',
@ContactEmailAddress VARCHAR(254)='',
@Avatar INT=0,
@AccountPassword VARBINARY=NULL,
@ResponseCity VARCHAR(254)='',
@ResponsePhone VARCHAR(254)='',
@ResponseSchool VARCHAR(254)='',
--@DocumentUrl VARCHAR(254)='',
--@DocumentType VARCHAR(254)='',
@Active VARCHAR(6)='', --YES/NO
@SecurityPIN int='',
@DateTimeStamp DATETIME=null
AS
BEGIN TRANSACTION
BEGIN

    IF (@DateTimeStamp IS NULL ) SET @DateTimeStamp = GETDATE();

    if @ResponseCity = '' or @ResponsePhone = '' or @ResponseSchool = '' GOTO ⚡
        On_Acct_Insert_Error

    DECLARE @AccountId INT;
    DECLARE @Question VARCHAR(254);

    INSERT INTO dbo.TP_Account (AccountRoleType, UserName, UserAddress,
        PhoneNumber, CreatedEmailAddress, ContactEmailAddress, Avatar,
        AccountPassword, DateTimeStamp, Active, SecurityPIN)
        VALUES
        (@AccountRoleType,@UserName,@UserAddress,@PhoneNumber,@CreatedEm
        ailAddress,@ContactEmailAddress,@Avatar,@AccountPassword,@DateTi
        meStamp,@Active,@SecurityPIN);

    SELECT @AccountId = @@IDENTITY;
    SELECT @Question = 'In what town or city was your first full time job?';
    INSERT INTO dbo.TP_SecurityQuestion(AccountId, QuestionType, Question,
        Response, DateTimeStamp)
        VALUES (@AccountId,'City', @Question,
        @ResponseCity,@DateTimeStamp);
    IF @@ERROR = -1 GOTO On_Acct_Insert_Error

    SELECT @Question = 'What were the last four digits of your childhood
        telephone number?';
    INSERT INTO dbo.TP_SecurityQuestion(AccountId, QuestionType, Question,
        Response, DateTimeStamp)
        VALUES (@AccountId,'Phone', @Question,
        @ResponsePhone,@DateTimeStamp);
    IF @@ERROR = -1 GOTO On_Acct_Insert_Error

    SELECT @Question = 'What primary school did you attend?';
    INSERT INTO dbo.TP_SecurityQuestion(AccountId, QuestionType, Question,
        Response, DateTimeStamp)

```

```

VALUES (@AccountId, 'School', @Question,
        @ResponseSchool, @DateTimeStamp);
IF @@ERROR = -1 GOTO On_Acct_Insert_Error

COMMIT TRANSACTION
END
RETURN 0
On_Acct_Insert_Error:
ROLLBACK TRANSACTION
RETURN -1

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
*   Description: Procedure to INSERT a row into the TP_Account table.
*   Note: An Account can be initially created without AccountDocuments
*   which can be added later. They are optional depending on the
*   type of appointments.
*****/
IF (SELECT object_id('dbo.TP_Account_Document_Insert_SP')) is not null
DROP PROCEDURE dbo.TP_Account_Document_Insert_SP;
GO
CREATE PROCEDURE dbo.TP_Account_Document_Insert_SP
    @CreatedEmailAddress VARCHAR(254) = '',
    @DocumentUrl VARCHAR(254) = '',
    @DocumentType VARCHAR(254) = '',
    @DateTimeStamp DATETIME
AS
BEGIN TRANSACTION
BEGIN
    IF (@DateTimeStamp IS NULL ) SET @DateTimeStamp = GETDATE();

    DECLARE @AccountId INT;

    SELECT @AccountId = AccountId
    FROM dbo.TP_Account
    WHERE TP_Account.CreatedEmailAddress = @CreatedEmailAddress
    IF @@ROWCOUNT != 1 GOTO On_AccountDocument_Insert_Error

    INSERT INTO dbo.TP_AccountDocument(AccountId, DocumentType, DocumentUrl,
        DateTimeStamp)
        VALUES
        (@AccountId, @DocumentType, @DocumentUrl, @DateTimeStamp);
    IF @@ERROR = -1 GOTO On_AccountDocument_Insert_Error

    COMMIT TRANSACTION
END
RETURN 0
On_AccountDocument_Insert_Error:
ROLLBACK TRANSACTION

```

```

        RETURN -1

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
 *   Description: Procedure to INSERT a row into the TP_Calendar table.
 *****/
IF (SELECT object_id('dbo.TP_Calendar_Insert_SP')) is not null
DROP PROCEDURE dbo.TP_Calendar_Insert_SP;
GO
CREATE PROCEDURE dbo.TP_Calendar_Insert_SP
    @CreatedEmailAddress VARCHAR(254)='',
    @Name VARCHAR(254)='',
    --Future feature (require documentation to schedule an event)
    -- @RequiredDoc VARCHAR(3)='',
    -- @RequiredDocType VARCHAR(254) NULL,
    @Location VARCHAR(254)='',
    @Description VARCHAR(4094)='',
    @Color VARCHAR(22),
    @IsChecked CHAR(1), --y/n yes/no
    @DateTimeStamp DateTime=NULL
AS
BEGIN TRANSACTION
BEGIN
    DECLARE @AccountId INT
    IF (@DateTimeStamp IS NULL ) SET @DateTimeStamp = GETDATE();

    SELECT @AccountId = TP_Account.AccountId FROM dbo.TP_Account
    WHERE TP_Account.CreatedEmailAddress = @CreatedEmailAddress
    IF @@ROWCOUNT = 0 GOTO On_Cal_Insert_Error

    INSERT INTO dbo.TP_Calendar (AccountId, Name, /*RequiredDoc,
        RequiredDocType,*/ Location, Description, Color, IsChecked,
        DateTimeStamp)
        VALUES (@AccountId,@Name,/
            *@RequiredDoc,@RequiredDocType,*/
            @Location,@Description,@Color,@IsChecked,@DateTimeStamp);
    IF @@ERROR = -1 GOTO On_Cal_Insert_Error

    COMMIT TRANSACTION
END
RETURN 0
On_Cal_Insert_Error:
    ROLLBACK TRANSACTION
    RETURN -1

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

/*****
*   Description: Procedure to DELETE a row from the TP_Calendar table.
*****/
IF (SELECT object_id('dbo.TP_Calendar_Delete_SP')) is not null
DROP PROCEDURE dbo.TP_Calendar_Delete_SP;
GO
CREATE PROCEDURE dbo.TP_Calendar_Delete_SP
    @CreatedEmailAddress VARCHAR(254)='',
    @Name VARCHAR(254)=''
AS
BEGIN TRANSACTION
BEGIN
    DECLARE @AccountId INT
    DECLARE @CalendarId INT

    SELECT @AccountId = TP_Account.AccountId FROM dbo.TP_Account
    WHERE TP_Account.CreatedEmailAddress = @CreatedEmailAddress
    IF @@ROWCOUNT = 0 GOTO On_Cal_Delete_Error

    SELECT @CalendarId = TP_Calendar.CalendarId
    FROM dbo.TP_Calendar
    WHERE AccountId = @AccountId
    AND Name = @Name
    IF @@ROWCOUNT = 0 GOTO On_Cal_Delete_Error

    --delete the events from the calendar
    DELETE FROM dbo.TP_Event
    WHERE TP_Event.CalendarId = @CalendarId
    IF @@ERROR = -1 GOTO On_Cal_Delete_Error

    --delete the calendar
    DELETE FROM dbo.TP_Calendar
    WHERE TP_Calendar.CalendarId = @CalendarId
    IF @@ERROR = -1 GOTO On_Cal_Delete_Error

    COMMIT TRANSACTION
END
RETURN 0
On_Cal_Delete_Error:
    ROLLBACK TRANSACTION
    RETURN -1
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
*   Description: Procedure to UPDATE a row from the TP_Calendar table.
*****/
IF (SELECT object_id('dbo.TP_Calendar_Update_SP')) is not null
DROP PROCEDURE dbo.TP_Calendar_Update_SP;
GO

```

```

CREATE PROCEDURE dbo.TP_Calendar_Update_SP
    @CreatedEmailAddress VARCHAR(254)='',
    @Name VARCHAR(254)='',
    --Future feature (require documentation to schedule an event)
    --@RequiredDoc VARCHAR(3)='',
    --@RequiredDocType VARCHAR(254) NULL,
    @Location VARCHAR(254)='',
    @Description VARCHAR(4094)='',
    @Color VARCHAR(22)='',
    @IsChecked CHAR(1), -- y/n yes/no
    @DateTimeStamp DateTime=NULL
AS
BEGIN TRANSACTION
BEGIN
    DECLARE @AccountId INT
    IF (@DateTimeStamp IS NULL ) SET @DateTimeStamp = GETDATE();

    SELECT @AccountId = TP_Account.AccountId FROM dbo.TP_Account
    WHERE TP_Account.CreatedEmailAddress = @CreatedEmailAddress
    IF @@ROWCOUNT = 0 GOTO On_Cal_Update_Error

    UPDATE dbo.TP_Calendar
        SET Name = @Name,
            --Future feature (require documentation to schedule an event)
            --RequiredDoc = @RequiredDoc,
            --RequiredDocType = @RequiredDocType,
            Location = @Location,
            Description = @Description,
            Color = @Color,
            IsChecked = @IsChecked,
            DateTimeStamp = @DateTimeStamp
    WHERE AccountId = @AccountId
    AND Name = @Name
    IF @@ERROR = -1 GOTO On_Cal_Update_Error

    SELECT *
    FROM dbo.TP_Calendar
    JOIN dbo.TP_Event ON TP_Event.CalendarId = dbo.TP_Calendar.CalendarId
    WHERE TP_Calendar.AccountId = @AccountId
    AND TP_Calendar.Name = @Name

    IF (@@ROWCOUNT > 0)
    BEGIN
        UPDATE dbo.TP_Event
            SET TP_Event.Color = @Color,
                TP_Event.DateTimeStamp = @DateTimeStamp
        FROM dbo.TP_Calendar
        JOIN dbo.TP_Event ON TP_Event.CalendarId = dbo.TP_Calendar.CalendarId
        WHERE TP_Calendar.AccountId = @AccountId
        AND TP_Calendar.Name = @Name
        IF @@ERROR = -1 GOTO On_Cal_Update_Error
    END
END

```

```

        COMMIT TRANSACTION
    END
    RETURN 0
    On_Cal_Update_Error:
        ROLLBACK TRANSACTION
        RETURN -1
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
 *   Description: Procedure to get calendar(s)
 *****/
IF (SELECT object_id('dbo.TP_Calendar_Select_SP')) is not null
DROP PROCEDURE dbo.TP_Calendar_Select_SP;
GO
CREATE PROCEDURE dbo.TP_Calendar_Select_SP
    @CreatedEmailAddress VARCHAR(254) = null,
    @CalendarName VARCHAR(254) = null
    --Future feature (require documentation to schedule an event)
    --@RequiredDoc VARCHAR(254) = null,
    --@RequiredDocType VARCHAR(254) = null
AS
    DECLARE @AccountId INT

    SELECT @AccountId=AccountId
    FROM dbo.TP_Account
    WHERE CreatedEmailAddress = @CreatedEmailAddress

    select *
    FROM TP_Calendar cal
    WHERE (cal.AccountId = @AccountId OR @AccountId IS NULL)
    AND (cal.Name = @CalendarName OR @CalendarName IS NULL)
    --AND (cal.RequiredDoc = @RequiredDoc OR @RequiredDoc IS NULL)
    --AND (cal.RequiredDocType = @RequiredDocType OR @RequiredDocType IS NULL)

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
 *   Description: Procedure to UPDATE TP_Calendar.IsChecked
 *****/
IF (SELECT object_id('dbo.TP_Calendar_Update_IsChecked_SP')) is not null
DROP PROCEDURE dbo.TP_Calendar_Update_IsChecked_SP;
GO
CREATE PROCEDURE dbo.TP_Calendar_Update_IsChecked_SP
    @AccountId INT,
    @Name VARCHAR(254)=' ',

```

```

@IsChecked CHAR(1), --y/n yes/no
@DateTimeStamp DateTime=NULL
AS
BEGIN TRANSACTION
BEGIN
    IF (@DateTimeStamp IS NULL) SET @DateTimeStamp = GETDATE();

    UPDATE dbo.TP_Calendar
        SET IsChecked = @IsChecked,
            DateTimeStamp = @DateTimeStamp
    WHERE TP_Calendar.AccountId = @AccountId
    AND TP_Calendar.Name = @Name
    IF @@ROWCOUNT = 0 GOTO On_Calendar_Update_IsChecked_Error

    COMMIT TRANSACTION
END
RETURN 0
On_Calendar_Update_IsChecked_Error:
    ROLLBACK TRANSACTION
    RETURN -1

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
*   Description: Procedure to get calendar Event(s)
*****/
IF (SELECT object_id('dbo.TP_Event_Select_SP')) is not null DROP PROCEDURE
    dbo.TP_Event_Select_SP;
GO
CREATE PROCEDURE dbo.TP_Event_Select_SP
    @CreatedEmailAddress VARCHAR(254)=NULL,
    @CalendarName VARCHAR(254)=NULL,
    @BegDateTime DATETIME=NULL,
    @EndDateTime DATETIME=NULL
AS
    DECLARE @AccountId INT
    DECLARE @CalendarId INT

    SELECT @AccountId=AccountId
    FROM dbo.TP_Account
    WHERE CreatedEmailAddress = @CreatedEmailAddress

    SELECT @CalendarId=CalendarId
    FROM dbo.TP_Calendar
    WHERE TP_Calendar.AccountId = @AccountId
    AND TP_Calendar.Name = @CalendarName

    SELECT TP_Event.*
    FROM dbo.TP_Event
    WHERE TP_Event.AccountId = @AccountId

```

```

        AND TP_Event.CalendarId = @CalendarId
        AND (TP_Event.BegDateTime BETWEEN @BegDateTime AND @EndDateTime)
        ORDER BY TP_Event.BegDateTime ASC

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
 *   Description: Procedure to get calendar Event(s) by EventId
 *****/
IF (SELECT object_id('dbo.TP_Event_Select_By_EventId_SP')) is not null DROP PROCEDURE
    dbo.TP_Event_Select_By_EventId_SP;
GO
CREATE PROCEDURE dbo.TP_Event_Select_By_EventId_SP
    @EventId INT
AS
    SELECT *
    FROM dbo.TP_Event
    WHERE TP_Event.EventId = @EventId

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
 *   Description: Procedure to INSERT a row into the TP_Event table.
 *   Notes:
 *****/
IF (SELECT object_id('dbo.TP_Event_Insert_SP')) is not null DROP PROCEDURE
    dbo.TP_Event_Insert_SP;
GO
CREATE PROCEDURE dbo.TP_Event_Insert_SP
    @CreatedEmailAddress VARCHAR(254)='',
    @CalendarName VARCHAR(254)='',
    @Title VARCHAR(254)='',
    @BegDateTime DATETIME='',
    @EndDateTime DATETIME='',
    @AttachmentDocumentUrl VARCHAR(254)='',
    @Location VARCHAR(254)='',
    @Description VARCHAR(4094),
    @DateTimeStamp DATETIME=null
AS
BEGIN TRANSACTION
    BEGIN
        DECLARE @AccountId INT
        DECLARE @CalendarId INT
        DECLARE @Color VARCHAR(22)
        IF (@DateTimeStamp IS NULL) SET @DateTimeStamp = GETDATE();

```



```

SELECT @AccountId = TP_Account.AccountId
FROM dbo.TP_Account
WHERE TP_Account.CreatedEmailAddress = @CreatedEmailAddress
IF @@ROWCOUNT = 0 GOTO On_Event_Insert_Error

--Insert color from the Calendar. All events in
--this calendar will have the same color.
SELECT @CalendarId = TP_Calendar.CalendarId,
       @Color = TP_Calendar.Color
FROM dbo.TP_Account
JOIN dbo.TP_Calendar ON TP_Calendar.AccountId = TP_Account.AccountId
WHERE TP_Account.AccountId = @AccountId
AND TP_Calendar.Name = @CalendarName
IF @@ROWCOUNT = 0 GOTO On_Event_Insert_Error

INSERT INTO dbo.TP_Event (AccountId, CalendarId, Title, BegDateTime,
                        EndDateTime, AttachmentDocumentUrl, Location, Description, Color,
                        DateTimeStamp)
                        VALUES
                        (@AccountId,@CalendarId,@Title,@BegDateTime,@EndDateTime,@Attach
                        mentDocumentUrl,@Location,@Description,@Color,@DateTimeStamp);
IF @@ERROR = -1 GOTO On_Event_Insert_Error

COMMIT TRANSACTION
END
RETURN 0
On_Event_Insert_Error:
ROLLBACK TRANSACTION
RETURN -1

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
*   Description: Procedure to UPDATE a row into the TP_Event table.
*   Notes:
*****/
IF (SELECT object_id('dbo.TP_Event_Update_SP')) is not null DROP PROCEDURE
dbo.TP_Event_Update_SP;
GO
CREATE PROCEDURE dbo.TP_Event_Update_SP
    @CreatedEmailAddress VARCHAR(254)='',
    @CalendarName VARCHAR(254)='',
    @Title VARCHAR(254)='',
    @BegDateTime DATETIME='',
    @EndDateTime DATETIME='',
    @AttachmentDocumentUrl VARCHAR(254)='',
    @Location VARCHAR(254)='',
    @Description VARCHAR(4094),
    @DateTimeStamp DATETIME=NULL
AS

```

```

BEGIN TRANSACTION
BEGIN
    DECLARE @AccountId INT,
            @CalendarId INT,
            @EventId INT,
            @Color VARCHAR(22)
    IF (@DateTimeStamp IS NULL ) SET @DateTimeStamp = GETDATE();

    SELECT
        @AccountId = TP_Account.AccountId,
        @CalendarId = TP_Calendar.CalendarId,
        @Color = TP_Calendar.Color
    FROM dbo.TP_Account
    JOIN dbo.TP_Calendar ON TP_Calendar.AccountId = TP_Account.AccountId
    WHERE TP_Account.CreatedEmailAddress = @CreatedEmailAddress
    AND TP_Calendar.Name = @CalendarName
    IF @@ROWCOUNT = 0 GOTO On_Event_Update_Error

    UPDATE dbo.TP_Event
    SET Title = @Title,
        BegDateTime = @BegDateTime,
        EndDateTime = @EndDateTime,
        AttachmentDocumentUrl = @AttachmentDocumentUrl,
        Location = @Location,
        Description = @Description,
        Color = @Color,
        DateTimeStamp = @DateTimeStamp
    WHERE
        TP_Event.AccountId = @AccountId
    AND TP_Event.CalendarId = @CalendarId
    AND TP_Event.BegDateTime = @BegDateTime
    IF @@ROWCOUNT = 0 GOTO On_Event_Update_Error

    COMMIT TRANSACTION
END
RETURN 0
On_Event_Update_Error:
    ROLLBACK TRANSACTION
    RETURN -1

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
*   Description: Procedure to DELETE a row into the TP_Event table.
*   Notes:
*****/
IF (SELECT object_id('dbo.TP_Event_Delete_SP')) is not null DROP PROCEDURE
    dbo.TP_Event_Delete_SP;
GO
CREATE PROCEDURE dbo.TP_Event_Delete_SP

```

```

@CreatedEmailAddress VARCHAR(254)='',
@CalendarName VARCHAR(254)='',
@BegDateTime DATETIME=''

AS
BEGIN TRANSACTION
BEGIN
    DECLARE @AccountId INT,
            @CalendarId INT,
            @EventId INT

    SELECT
        @AccountId = TP_Account.AccountId,
        @CalendarId = TP_Calendar.CalendarId
    FROM dbo.TP_Account
    JOIN dbo.TP_Calendar ON TP_Calendar.AccountId = TP_Account.AccountId
    WHERE TP_Account.CreatedEmailAddress = @CreatedEmailAddress
    AND TP_Calendar.Name = @CalendarName
    IF @@ROWCOUNT = 0 GOTO On_Event_Delete_Error

    DELETE FROM dbo.TP_Event
    WHERE
        TP_Event.AccountId = @AccountId
    AND TP_Event.CalendarId = @CalendarId
    AND TP_Event.BegDateTime = @BegDateTime
    IF @@ROWCOUNT = 0 GOTO On_Event_Delete_Error

    COMMIT TRANSACTION
END
RETURN 0
On_Event_Delete_Error:
    ROLLBACK TRANSACTION
    RETURN -1

GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
/*****
*   FrankP: Procedure to INSERT into the SecurityQuestions table
*****/
IF (SELECT object_id('dbo.TP_SecurityQuestion_SP')) is not null DROP PROCEDURE
    dbo.TP_SecurityQuestion_SP;
GO
CREATE PROCEDURE [dbo].[TP_SecurityQuestion_SP]
    @theAccountId int,
    @theQuestion VARCHAR(50),
    @theQuestionType VARCHAR(50),
    @theResponse VARCHAR(50)
AS
    INSERT INTO TP_SecurityQuestion
    ([AccountId], [Question], [QuestionType], [Response])
    Values

```

```

        (@theAccountId, @theQuestion, @theQuestionType, @theResponse)

    RETURN 0

GO

/*****
 *   FrankP: Procedure to SELECT the AccountID matching theEmail parm
 *****/
IF (SELECT object_id('dbo.TP_GetAccountId_SP')) is not null DROP PROCEDURE
    dbo.TP_GetAccountId_SP;
GO
CREATE PROCEDURE [dbo].[TP_GetAccountId_SP]
    @theEmail VARCHAR (50)
AS
    SELECT AccountId, UserName, AccountRoleType, CreatedEmailAddress
    FROM TP_Account
    WHERE CreatedEmailAddress = @theEmail
RETURN 0
GO

/*****
 *   Insert starter Account data
 *****/
exec dbo.TP_Account_Insert_SP
    @AccountRoleType='User',
    @UserName='JimS',
    @UserAddress='101 Main, Philadelphia, PA 01234',
    @PhoneNumber='+11234567890',
    @CreatedEmailAddress='jims@temple.edu',
    @ContactEmailAddress='tun49199@temple.edu',
    @Avatar=3,
    @AccountPassword=0x2673BA5EA47ADBACDC45E9D9B2EF6B2B, --lower case 'p'
    @ResponseCity= 'city',
    @ResponsePhone= '1234',
    @ResponseSchool= 'school',
    --@DocumentUrl='',
    --@DocumentType='',
    @Active='no',
    @SecurityPIN='1234',
    @DateTimeStamp=NULL

exec dbo.TP_Account_Insert_SP
    @AccountRoleType='Administrator',
    @UserName='FrankP',
    @UserAddress='202 Main, Philadelphia, PA 01234',
    @PhoneNumber='+10987654321',
    @CreatedEmailAddress='frankp@temple.edu',
    @ContactEmailAddress='frankp@hotmail.com',
    @Avatar=4,
    @AccountPassword=0x2673BA5EA47ADBACDC45E9D9B2EF6B2B, --lower case 'p'
    @ResponseCity= 'city',
    @ResponsePhone= '1234',
    @ResponseSchool= 'school',
    --@DocumentUrl='',

```

```

--@DocumentType='',
@Active='no',
@SecurityPIN='1234',
@DateTimeStamp=NULL

GO

/*****
* Insert sample documents
*****/
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES (NULL, '/
Images/AppleWatch-EKG-John-Doe.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES (NULL, '/
Images/AppleWatch-Sleeping-BPM-John-Doe.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES (NULL, '/
Images/Garmin-PulseOx-John-Doe.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES ('BusPass',
'/Images/BusPass-Eric-Cartman.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES ('BusPass',
'/Images/BusPass-Kenny-McCormick.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES ('BusPass',
'/Images/BusPass-Kyle-Broflofski.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES ('BusPass',
'/Images/BusPass-Ralph-Kramden.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES
('COVID19VaccRecord', '/Images/COVID19VaccRecord-Blank.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES
('DriverLicense', '/Images/DriverLicense-Curly-Howard.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES
('DriverLicense', '/Images/DriverLicense-Larry-Fine.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES
('DriverLicense', '/Images/DriverLicense-Moe-Howard.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES
('DriverLicense', '/Images/DriverLicense-Shemp-Howard.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES
('VoterRegistration', '/Images/VoterRegistration-John-Democrat.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES
('VoterRegistration', '/Images/VoterRegistration-John-Independent.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES
('VoterRegistration', '/Images/VoterRegistration-John-Republican.jpg');
INSERT INTO dbo.TP_SampleDocument (DocumentType, DocumentUrl) VALUES
('VoterRegistration', '/Images/VoterRegistration-John-Uncolored.jpg')
GO

/*****
* Insert starter AccountDocument data
*****/
dbo.TP_Account_Document_Insert_SP
@CreatedEmailAddress='jims@temple.edu',
@DocumentUrl='/Images/BusPass-Eric-Cartman.jpg',
@DocumentType='BusPass',
@DateTimeStamp=NULL

GO

```

```

/*****
* Insert sample Calendar
*****/

dbo.TP_Calendar_Insert_SP
    @CreatedEmailAddress='jims@temple.edu',
    @Name='Election',
    --@RequiredDoc='Yes',
    --@RequiredDocType='BusPass',
    @Location='Ballot Box',
    @Description='Election Day',
    @Color='Chartreuse',
    @IsChecked='y',
    @DateTimeStamp=NULL
GO

dbo.TP_Calendar_Insert_SP
    @CreatedEmailAddress='jims@temple.edu',
    @Name='Urgent Care',
    --@RequiredDoc='No',
    --@RequiredDocType='',
    @Location='123 NearYou Street, Hometown USA',
    @Description='Better Care. Faster Care.',
    @Color='Aqua',
    @IsChecked='y',
    @DateTimeStamp=NULL
GO

dbo.TP_Calendar_Insert_SP
    @CreatedEmailAddress='jims@temple.edu',
    @Name='Office',
    --@RequiredDoc='Yes',
    --@RequiredDocType='BusPass',
    @Location='Zoom Meeting ID 111-222-1234',
    @Description='WebApp Solutions',
    @Color='Orange',
    @IsChecked='y',
    @DateTimeStamp=NULL
GO

dbo.TP_Calendar_Insert_SP
    @CreatedEmailAddress='jims@temple.edu',
    @Name='Birthdays',
    --@RequiredDoc='No',
    --@RequiredDocType='',
    @Location='Home',
    @Description='Its your Birthday',
    @Color='SkyBlue',
    @IsChecked='y',
    @DateTimeStamp=NULL
GO

dbo.TP_Calendar_Insert_SP
    @CreatedEmailAddress='jims@temple.edu',
    @Name='Holidays',
    --@RequiredDoc='No',
    --@RequiredDocType='',

```

```

        @Location='Home',
        @Description='No work today!',
        @Color='Cyan',
        @IsChecked='y',
        @DateTimeStamp=NULL
GO
dbo.TP_Calendar_Insert_SP
    @CreatedEmailAddress='frankp@temple.edu',
    @Name='Election',
    --@RequiredDoc='Yes',
    --@RequiredDocType='BusPass',
    @Location='Ballot Box',
    @Description='Election Day',
    @Color='Chartreuse',
    @IsChecked='y',
    @DateTimeStamp=NULL
GO
dbo.TP_Calendar_Insert_SP
    @CreatedEmailAddress='frankp@temple.edu',
    @Name='Office',
    --@RequiredDoc='Yes',
    --@RequiredDocType='BusPass',
    @Location='Zoom Meeting ID 111-222-1234',
    @Description='WebApp Solutions',
    @Color='Orange',
    @IsChecked='y',
    @DateTimeStamp=NULL
GO
dbo.TP_Calendar_Insert_SP
    @CreatedEmailAddress='frankp@temple.edu',
    @Name='Birthdays',
    --@RequiredDoc='No',
    --@RequiredDocType='',
    @Location='38.435092, -78.8697548',
    @Description='Its your Birthday',
    @Color='SkyBlue',
    @IsChecked='y',
    @DateTimeStamp=NULL
GO
dbo.TP_Calendar_Insert_SP
    @CreatedEmailAddress='frankp@temple.edu',
    @Name='Holidays',
    --@RequiredDoc='No',
    --@RequiredDocType='',
    @Location='Home',
    @Description='No work today!',
    @Color='Cyan',
    @IsChecked='y',
    @DateTimeStamp=NULL
GO
/*****
*   Insert sample CalendarEvent

```

*****/

```
dbo.TP_Event_Insert_SP
    @CreatedEmailAddress='jims@temple.edu',
    @CalendarName='Election',
    @Title='Governer Race',
    @BegDateTime='2021-04-26 15:30:00',
    @EndDateTime='2021-04-26 16:50:00',
    @AttachmentDocumentUrl='',
    @Location='Ballot box',
    @Description='Time to choose.',
    @DateTimeStamp=NULL
GO

dbo.TP_Event_Insert_SP
    @CreatedEmailAddress='jims@temple.edu',
    @CalendarName='Office',
    @Title='Standup',
    @BegDateTime='2021-04-26 15:30:00',
    @EndDateTime='2021-04-26 16:50:00',
    @AttachmentDocumentUrl='',
    @Location='Zoom Meeting ID 111-222-1234',
    @Description='Standup Meeting',
    @DateTimeStamp=NULL
GO

dbo.TP_Event_Insert_SP
    @CreatedEmailAddress='jims@temple.edu',
    @CalendarName='Holidays',
    @Title='Christmas',
    @BegDateTime='2021-12-25 00:00:00',
    @EndDateTime='2021-12-25 23:59:59',
    @AttachmentDocumentUrl='',
    @Location='Home',
    @Description='Ho Ho Ho!',
    @DateTimeStamp=NULL
GO

dbo.TP_Event_Insert_SP
    @CreatedEmailAddress='jims@temple.edu',
    @CalendarName='Holidays',
    @Title='4th Of July',
    @BegDateTime='2021-07-04 00:00:00',
    @EndDateTime='2021-07-04 23:59:59',
    @AttachmentDocumentUrl='',
    @Location='Home',
    @Description='Happy Birthday America',
    @DateTimeStamp=NULL
GO

dbo.TP_Event_Insert_SP
    @CreatedEmailAddress='jims@temple.edu',
    @CalendarName='Holidays',
    @Title='Memorial Day',
    @BegDateTime='2021-05-31 00:00:00',
    @EndDateTime='2021-05-31 23:59:59',
    @AttachmentDocumentUrl='',
```



```
@Location='USA',
@Description='Honor The Fallen Soldiers',
@DateTimeStamp=NULL
GO
dbo.TP_Event_Insert_SP
@CreatedEmailAddress='frankp@temple.edu',
@CalendarName='Office',
@Title='Standup',
@BegDateTime='2021-04-26 09:30:00',
@EndDateTime='2021-04-26 10:50:00',
@AttachmentDocumentUrl='',
@Location='Zoom Meeting ID 111-222-1234',
@Description='Standup Meeting',
@DateTimeStamp=NULL
GO
```