

Fikret Cansel-19253078

Eyüp Duran -19253060

- Parametre alan ve hesaplama yapıp sonucunu parametre olarak döndüren SP

1) Bir kullanıcının istenilen aralıklardaki postlarını getiren Sp :

```
CREATE PROC [dbo].[Sp_UserPostsLastDay]
@day as int,
@username varchar(50)
AS BEGIN
Declare @userId int
Set @userId =(Select Top 1 Id from Users Where UserName=@username)
IF EXISTS(SELECT*FROM Users Where
Users.Id=@userId)
BEGIN
Select*from Posts Where
DATEDIFF(DAY,SendingDate,GETDATE())<=@day and
DATEDIFF(DAY,SendingDate,GETDATE())>0 and UserId=@userId Return
End
Select'Kullanıcı Bulunamadı'
End

EXEC Sp_UserPostsLastDay @day=10,@username='fcansel19'
```

2) Bir postun likelerini hesaplayıp döndüren Sp:

```
CREATE PROC [dbo].[Sp_UserPostLikes]
@postId as int
AS BEGIN
DECLARE @UserId NVARCHAR(50)-- database name
Declare @like int
Set @like = 0
DECLARE db_cursor CURSOR FOR
SELECT UserId
FROM PostLikes

OPEN db_cursor
FETCH NEXT FROM db_cursor INTO @UserId

WHILE @@FETCH_STATUS= 0
BEGIN
        Set @like=@like+1
        Select @like
        FETCH NEXT FROM db_cursor INTO @UserId
END
CLOSE db_cursor
DEALLOCATE db_cursor
End
```

- Birden fazla ilintili tabloya insert/update/delete(biri insert biri update olabilir yada türevleri) işlemlerini yapan SP (SP içerisinde transaction tanımlı olmalıdır)

1) Profil oluştururken kullanıcı kayıtlı değilse kullanıcıyı kaydeden, kullanıcı kayıtlıysa bu kullanıcı için bir profil oluşturan stored procedure

```
CREATE PROC [dbo].[Sp_CreateProfile]
@username AS NVARCHAR(50),
@email AS NVARCHAR(50),
@password AS NVARCHAR(50),
@gender AS NVARCHAR(50),
@cityId AS INT,
@pictureUrl AS NVARCHAR(50)
AS BEGIN
BEGIN TRANSACTION
IF NOT EXISTS(SELECT*FROM Users Where
Users.UserName=@username)
BEGIN
INSERT INTO Users(UserName,Email,Password)Values (@username,@email,@password)
End
ELSE
INSERT INTO Profiles(PictureUrl,Gender,UserId,CityId) Values (@pictureUrl,@gender,(Select
Id from Users where @username=UserName),@cityId)
End
IF (@@ERROR<>0)
BEGIN
ROLLBACK TRANSACTION
END
COMMIT TRANSACTION
```

1) Kullanıcıya yetki atayan bir stored procedure. Kullanıcıya verilmek istenen yetki OperationClaims tablosunda mevcut değilse yetkiyi bu tabloya kaydeden, kullanıcıya verilmek istenen yetki OperationClaims tablosunda mevcutsa bu kullanıcıya karşılık gelen yetkiyi UserOperationClaims tablosuna kaydeden bir stored procedure

```
CREATE PROC [dbo].[Sp_GiveClaimToUser]
@username AS NVARCHAR(50),
@operationClaimName AS NVARCHAR(50)
AS BEGIN
BEGIN TRANSACTION
IF NOT EXISTS(select * from OperationClaims where OperationClaims.Name =
@operationClaimName)
BEGIN
INSERT INTO OperationClaims(Name) Values (@operationClaimName)
End
ELSE
INSERT INTO UserOperationClaims(UserId,OperationClaimId)
Values ((Select Id from Users where @username=UserName),
(Select Id from OperationClaims where @operationClaimName=Name))
End
IF (@@ERROR<>0)
BEGIN
ROLLBACK TRANSACTION
END
COMMIT TRANSACTION
```

