

Veri Tabanı Yönetimi ve Modellemesi

HAFTA 9

Dr. Fatmana Şentürk

Haftalık Ders Akışı

1. Veritabanı Kavramlarına Giriş
2. Veri Tabanı Türleri, İlişkisel Veri Tabanı Tasarımı
3. ER Diyagramları ve Normalizasyon
4. SQL Server Arayüzü, Veri Tabanı Nesneleri
5. T-SQL ve SQL Sorguları
6. İndeks ve View
7. Geçici Tablolar, Kontrol Yapıları
8. Ara Sınav
9. **Stored Procedure**
10. Fonksiyonlar
11. Tetikleyiciler
12. Transaction Kavramları ve Yedekleme
13. Kullanıcı Türleri ve Kullanıcı Yönetimi
14. No-SQL Veri Tabanları

Stored Procedure(SP)

- SP belirli bir işlevi, görevi yerine getirmek için yazılan SQL ifadeleri
- Sık kullanılan yapıların SP olarak tek bir defa yazılması ve kullanılmak istendiği yerde sadece çağırılması (Tekrar tekrar kullanılması)
- İlk çalıştığında derlenir, sonrasında sadece işletilir derleme ihtiyacına gerek kalmaz
- Server tarafında saklanabilir ve parametre alabilir
- Verinin saklanma biçiminin ve/veya tabloların tasarımının saklanması

Stored Procedure Türleri

- **Extended Stored Procedure:** Genellikle *.dll şeklinde prosedürlerdir.
- **CLR Stored Procedure:** SQL Server 2005 sonrasında CLR ortamında herhangi bir dili kullanarak kodlanan SP'lerdir.
- **System Stored Procedure:** Genellikle sp_ ön ekiyle başlarlar ve hepsi master veri tabanında tutulan SP'lerdir.
- **Kullanıcı Tanımlı SP:** Kullanıcıların tanımladığı SP'lerdir.

Stored Procedure

○ Genel Yapısı

```
CREATE PROCEDURE/PROC prosedur_adı
```

```
Parametre seçenekleri
```

```
AS
```

```
BEGIN
```

```
T-SQL ifadeleri
```

```
END
```

```
Çalıştırma Seçeneği:
```

```
EXEC prosedur_adı (parametreler)
```

Stored Procedure

```
CREATE PROC sp_getPersonelBilgileri AS BEGIN
SELECT tbl_Staff.name,surname,
gorevi=
CASE
WHEN staffRoleId IS NULL THEN '-'
ELSE tbl_StaffRole.name
END
FROM tbl_Staff LEFT JOIN tbl_StaffRole
ON tbl_Staff.staffRoleId=tbl_StaffRole.id
END
```

Çalıştırma: EXEC sp_getPersonelBilgileri

Parametre Alan SP

```
CREATE PROC sp_getGoreveGorePersonelListesi
@Gorev NVARCHAR(50)

AS BEGIN
SELECT tbl_Staff.name,surname,
gorevi=
CASE
WHEN staffRoleId IS NULL THEN '-'
ELSE tbl_StaffRole.name
END
FROM tbl_Staff LEFT JOIN tbl_StaffRole
ON tbl_Staff.staffRoleId=tbl_StaffRole.id
WHERE tbl_StaffRole.name LIKE '%' + @Gorev + '%'
END
```

Çalışma:

```
EXEC sp_getGoreveGorePersonelListesi 'Tem'
EXEC sp_getGoreveGorePersonelListesi
@Gorev = 'Temizlik'
```

Parametre Döndüren SP

```
CREATE PROCEDURE sp_ToplamPersonelSayisi  
    @sayi AS INT OUT  
AS BEGIN  
    SELECT @sayi=COUNT(*) FROM tbl_Staff  
END
```

Çalışma:

```
DECLARE @sonuc AS INT  
EXEC sp_ToplamPersonelSayisi @sayi=@sonuc OUT  
SELECT @sonuc
```


SP-INSERT

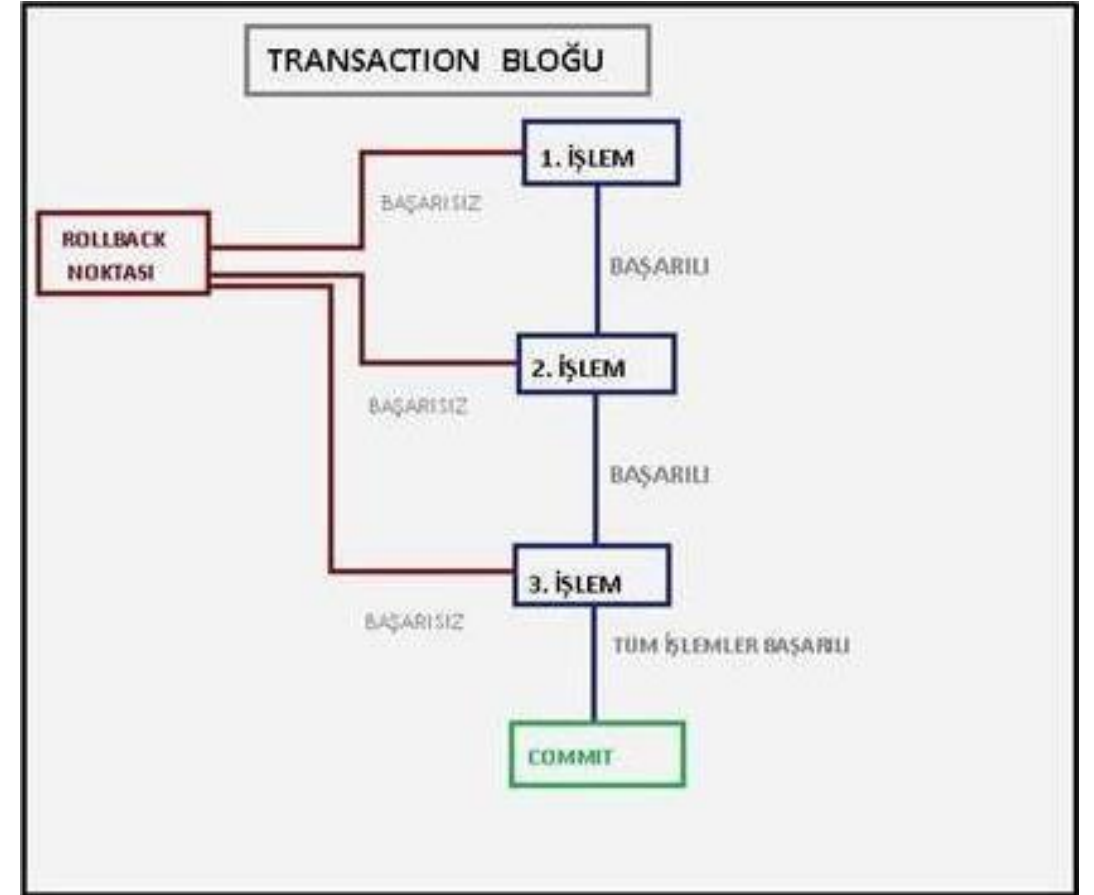
```
CREATE PROCEDURE sp_Add_Personel
@Ad AS NVARCHAR(50),@Soyad AS NVARCHAR(50), @PersonelRole AS INT,
@Adres AS NVARCHAR(50), @Telefon AS NVARCHAR(50),
@iseBaslamatarihi AS DATE, @dogumtarihi AS DATE, @maas AS INT
AS BEGIN
    INSERT INTO tbl_Staff VALUES (@Ad,@Soyad, @dogumtarihi, @iseBaslamatarihi,
    @PersonelRole, @Adres,@Telefon,@maas)
END
```

```
Çalışma: SET DATEFORMAT DMY;
DECLARE @Role AS INT
SELECT @Role=id FROM tbl_StaffRole Where name='veteriner'
```

```
EXEC sp_Add_Personel @Ad='Ayca', @soyad='Yaren', @PersonelRole=@Role, @Adres=NULL,
@Telefon='2342432434', @iseBaslamaTarihi='09.12.2024',@dogumtarihi='20.12.2000',
@maas=35000
```

Transaction

- Transaction en küçük işlem bloğu
- Transaction;
 - Ya bütün işlemleri gerçekleştirir ya da hiçbirini gerçekleştirmez.
- BEGIN ile başlar
- ROLLBACK : Tüm işlemleri geri alır
- COMMIT: Tüm işlemlerin başarılı şekilde yapıldığı düşünülür.
- ACID özelliği



SP-Transaction

```
CREATE PROCEDURE sp_Add_Personel_Controlled @Ad AS NVARCHAR(50),@Soyad AS NVARCHAR(50),
@PersonelRole AS INT, @Adres AS NVARCHAR(50), @Telefon AS NVARCHAR(50),
@iseBaslamatarihi AS DATE, @dogumtarihi AS DATE, @maas AS INT
AS BEGIN
    BEGIN TRANSACTION
        INSERT INTO tbl_Staff VALUES (@Ad,@Soyad, @dogumtarihi,
            @iseBaslamatarihi, @PersonelRole, @Adres,@Telefon,@maas)
        IF (@@error <> 0)
            BEGIN
                SELECT 'ROLEBACK'
                ROLLBACK TRANSACTION
            END
        COMMIT TRANSACTION
    END
END
```

SP-Transaction

```
CREATE PROCEDURE sp_Add_Personel_ControlledParameter @Ad AS NVARCHAR(50),@Soyad AS
NVARCHAR(50), @PersonelRole AS INT, @Adres AS NVARCHAR(50), @Telefon AS NVARCHAR(50),
@iseBaslamatarihi AS DATE, @dogumtarihi AS DATE, @maas AS INT,@sonuc AS INT OUT
AS BEGIN
    BEGIN TRANSACTION
        INSERT INTO tbl_Staff VALUES (@Ad,@Soyad, @dogumtarihi, @iseBaslamatarihi,
            @PersonelRole, @Adres,@Telefon,@maas)
    IF (@@error <> 0)
    BEGIN
        SELECT @sonuc=0
        ROLLBACK TRANSACTION
    END
    ELSE BEGIN
        SELECT @sonuc=1
        COMMIT TRANSACTION
    END
END
END
```

SP-Transaction

Çalışma:

```
SET DATEFORMAT DMY;  
DECLARE @Role AS INT  
SELECT @Role=id FROM tbl_StaffRole Where name='veteriner'  
DECLARE @IslemSonuc AS INT  
EXEC sp_Add_Personel_ControlledParameter  
@Ad='Ayca', @soyad='Yaren',  
@PersonelRole=@Role,  
@Adres=NULL, @Telefon=NULL,  
@iseBaslamaTarihi='09.12.2024',  
@dogumtarihi='20.12.2000',  
@maas=35000, @sonuc=@IslemSonuc OUT  
SELECT @IslemSonuc
```

ACID(Atomicity, Consistency, Isolation, Durability)

- **Atomicity (Bölünmezlik):** Bir transaction bloğu yarım kalmaz. Yarım kalan transaction bloğu veri tutarsızlığına neden olur. Ya tüm işlemler gerçekleştirilir, ya da transaction başlangıcına geri döner.
- **Consistency (Tutarlılık):** Transaction veri tutarlılığı sağlamalıdır. Yani bir transaction içerisinde güncelleme işlemi gerçekleştiyse ve ya kalan tüm işlemler de gerçekleşmeli ya da güncelle işlemi de geri alınmalıdır.
- **Isolation (İzolasyon):** Bir transaction tarafından gerçekleştirilen değişiklikler tamamlanmadan bir başka transaction tarafından görülememeli gerekir. Yani her transaction ayrı ayrı işlenmelidir.
- **Durability (Dayanıklılık):** Transaction'lar veri üzerinde karmaşık işlemler gerçekleştirirken verinin bütününe güvence altına almalıdır. Bu sebeple transaction hatalara karşı dayanıklı olmalıdır.

Örnek: Default Parametrelili SP

```
CREATE PROCEDURE [dbo].[sp_Add_Personel_DefaultParametre]
@Ad AS NVARCHAR(50),@Soyad AS NVARCHAR(50), @PersonelRole AS INT,
@Adres AS NVARCHAR(50), @Telefon AS NVARCHAR(50),
@iseBaslamatarihi AS DATE=NULL, @dogumtarihi AS DATE, @maas AS INT
AS BEGIN
    IF @iseBaslamatarihi IS NULL
        SELECT @iseBaslamatarihi=GETDATE()
    BEGIN TRANSACTION
    INSERT INTO tbl_Staff VALUES (@Ad,@Soyad, @dogumtarihi,
        @iseBaslamatarihi, @PersonelRole, @Adres,@Telefon,@maas)
    IF (@@error <> 0)
    BEGIN
        ROLLBACK TRANSACTION
    END
    ELSE BEGIN
        COMMIT TRANSACTION
    END
END
END
```

Çalışma:

```
SET DATEFORMAT DMY;  
DECLARE @Role AS INT  
SELECT @Role=id FROM tbl_StaffRole Where name='veteriner'  
EXEC sp_Add_Personel_DefaultParametre  
@Ad='Ayca', @soyad='Yaren', @PersonelRole=@Role, @Adres=NULL, @Telefon='34535',  
@dogumtarihi='20.12.2000', @maas=35000
```

```
SET DATEFORMAT DMY;  
DECLARE @Role AS INT  
SELECT @Role=id FROM tbl_StaffRole Where name='veteriner'  
EXEC sp_Add_Personel_DefaultParametre  
@Ad='Ayca', @soyad='Yaren', @PersonelRole=@Role, @Adres=NULL, @Telefon='345345',  
@iseBaslamaTarihi=DEFAULT, @dogumtarihi='20.12.2000', @maas=35000
```


Çalışma:

```
SET DATEFORMAT DMY;  
DECLARE @Role AS INT  
SELECT @Role=id FROM tbl_StaffRole Where name='veteriner'  
EXEC sp_Add_Personel_DefaultParametre  
@Ad='Ayca', @soyad='Yaren', @PersonelRole=@Role, @Adres=NULL, @Telefon='345345',  
@iseBaslamaTarihi='20.12.2024', @dogumtarihi='20.12.2000', @maas=35000
```

Örnek: Eklenen personelin Id'sini döndüren SP

```
CREATE PROCEDURE [dbo].[sp_Add_Personel_ControlledParameterV2] @Ad AS NVARCHAR(50),@Soyad
AS NVARCHAR(50), @PersonelRole AS INT, @Adres AS NVARCHAR(50), @Telefon AS NVARCHAR(50),
@iseBaslamatarihi AS DATE, @dogumtarihi AS DATE, @maas AS INT,@sonuc AS INT OUT
AS BEGIN
    BEGIN TRANSACTION
        INSERT INTO tbl_Staff VALUES (@Ad,@Soyad, @dogumtarihi, @iseBaslamatarihi,
            @PersonelRole, @Adres,@Telefon,@maas)
    IF (@@error <> 0)
    BEGIN
        SELECT @sonuc=0
        ROLLBACK TRANSACTION
    END
    ELSE BEGIN
        SELECT @sonuc=@@IDENTITY
        COMMIT TRANSACTION
    END
END
END
```

```
SET DATEFORMAT DMY;
DECLARE @Role AS INT
SELECT @Role=id FROM tbl_StaffRole Where name='veteriner'
DECLARE @IslemSonuc AS INT
EXEC sp_Add_Personel_ControlledParameterV2
@Ad='Ayca', @soyad='Yaren',
@PersonelRole=@Role,
@Adres=NULL, @Telefon='345345',
@iseBaslamaTarihi='09.12.2024',
@dogumtarihi='20.12.2000',
@maas=35000, @sonuc=@IslemSonuc OUT
SELECT @IslemSonuc
```

