

Veri Tabanı Yönetimi ve Modellemesi

HAFTA 7

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. **Stored Procedure ve Fonksiyonlar**
8. Ara Sınav
9. Tetikleyiciler
10. Transaction Kavramları ve Yedekleme
11. Kullanıcı Türleri ve Kullanıcı Yönetimi
12. No-SQL Veri Tabanları
13. No-SQL Veri Tabanları
14. Proje Sunumu
15. Proje Sunumları

ROW_NUMBER() ifadesi

```
SELECT
ROW_NUMBER() OVER(ORDER BY FT.description ASC)R_Id,
Name=Ft.description,FS.stock,NeededStock=SUM(quantity)
FROM tbl_animal_Food_Schedule AS AFS
    inner join tbl_Food_Stock AS FS ON FS.food_id=AFS.food_id
    inner join tbl_FoodType AS FT ON FT.id=AFS.food_id
GROUP BY Ft.description,FS.stock ---HAVING FS.stock<SUM(quantity)
```

Değişken Tanımları

- Declare @parameterName ParameterTYpe
- SET @parameterName=Value
- DECLARE @param CHAR(10)
SET@param=GETDATE()
SELECT @param
- DECLARE @deger INT
SELECT @deger =1
SELECT @deger

CASE-WHEN Yapısı

○ DECLARE @deger INT

SET @deger=1

SELECT

 CASE @deger

 WHEN 1 THEN 'bir'

 WHEN 2 THEN 'iki'

 END

CASE-WHEN Yapısı

```
SELECT
    CASE
        WHEN address IS NULL THEN 'Yok'
        ELSE address
    END
FROM tbl_staff
```

CASE-WHEN Yapısı

```
SELECT Name=Ft.description,FS.stock,NeededStock=SUM(quantity),  
       CASE  
           WHEN FS.stock<SUM(quantity) THEN 'Alınması gerekli'  
           ELSE 'Stokta var'  
       END  
FROM tbl_animal_Food_Schedule AS AFS  
     inner join tbl_Food_Stock AS FS ON FS.food_id=AFS.food_id  
     inner join tbl_FoodType AS FT ON FT.id=AFS.food_id  
GROUP BY Ft.description,FS.stock
```

CASE-WHEN

```
SELECT id,Name,animalType_Id,  
CASE  
    WHEN birthDate is NULL THEN arrivalDate  
    WHEN arrivalDate is NULL THEN birthDate  
END  
FROM tbl_animal
```


IF –ELSE Yapısı

```
DECLARE @id int
```

```
SET @id=1
```

```
DECLARE @param AS VARCHAR(50)
```

```
    IF (@id=1)
```

```
        SET @param='BİR'
```

```
    ELSE
```

```
        SET @param='İKİ'
```

```
SELECT @param
```

IF –ELSE IF Yapısı

```
DECLARE @id int
```

```
SET @id=3
```

```
DECLARE @param AS VARCHAR(50)
```

```
    IF (@id=1)
```

```
        SET @param='BİR'
```

```
    ELSE IF(@id=2)
```

```
        SET @param='İKİ'
```

```
    ELSE
```

```
        SET @param='DEFAULT'
```

```
SELECT @param
```

IF –ELSE IF Yapısı

```
DECLARE @Tarih DATETIME
SET @Tarih='2020-02-01'---CONVERT
--SET
@Tarih=CONVERT(DATETIME,'01.12.2019',104)

    DECLARE @Ad VARCHAR (7)
    DECLARE @AyNo TINYINT
    SET @AyNo = MONTH(@Tarih)

    IF @AyNo = 1 SET @Ad = 'Ocak'
    ELSE IF @AyNo = 2 SET @Ad = 'Şubat'
    ELSE IF @AyNo = 3 SET @Ad = 'Mart'
```

```
    ELSE IF @AyNo = 4 SET @Ad = 'Nisan'
    ELSE IF @AyNo = 5 SET @Ad = 'Mayıs'
    ELSE IF @AyNo = 6 SET @Ad = 'Haziran'
    ELSE IF @AyNo = 7 SET @Ad = 'Temmuz'
    ELSE IF @AyNo = 8 SET @Ad = 'Ağustos'
    ELSE IF @AyNo = 9 SET @Ad = 'Eylül'
    ELSE IF @AyNo = 10 SET @Ad = 'Ekim'
    ELSE IF @AyNo = 11 SET @Ad = 'Kasım'
    ELSE IF @AyNo = 12 SET @Ad = 'Aralık'

SELECT @Ad
```

WHILE Yapısı

```
DECLARE @x int
```

```
SET @x=1
```

```
WHILE (@x<3) BEGIN
```

```
    SET @x=@x+1
```

```
END
```

```
SELECT @x
```

WHILE Yapısı

```
DECLARE @x int
```

```
SET @x=1
```

```
WHILE (@x<3) BEGIN
```

```
    SET @x=@x+1
```

```
    IF (@x=3)
```

```
        SELECT 'EXIT'
```

```
END
```

```
SELECT @x
```

TRY-CATCH Yapısı

```
BEGIN TRY
```

```
    PRINT 3/0;
```

```
END TRY
```

```
BEGIN CATCH
```

```
    PRINT 'Catch bloğunun içi';
```

```
    PRINT ERROR_NUMBER();
```

```
    PRINT ERROR_MESSAGE();
```

```
    PRINT ERROR_NUMBER();
```

```
END CATCH
```

```
PRINT 'Catch bloğu bittikten sonra'
```

```
PRINT ERROR_NUMBER()
```

CURSOR ifadesi

```
DECLARE @name AS nvarchar(50)

DECLARE @stock AS FLOAT

DECLARE @needed AS FLOAT

DECLARE @index AS INT

SET @index=0

DECLARE @sonuc AS VARCHAR(50)

DECLARE stockControl Scroll CURSOR FOR

    SELECT Name=Ft.description,FS.stock,NeededStock=SUM(quantity)

        FROM tbl_animal_Food_Schedule AS AFS

        INNER JOIN tbl_Food_Stock AS FS ON

FS.food_id=AFS.food_id

        INNER JOIN tbl_FoodType AS FT ON FT.id=AFS.food_id

        GROUP BY Ft.description,FS.stock

OPEN stockControl

FETCH stockControl INTO @name,@stock,@needed

WHILE (@@Fetch_Status<>-1)

Begin

    if (@stock<@needed)

        SELECT @sonuc=@name+'-Alınmalı'

    Fetch stockControl into @name,@stock,@needed

    SET @index=@index+1

End

Close stockControl

DeAllocate stockControl

Select @sonuc as sonuc

SELECT CAST(@index AS NVARCHAR) +' ' +'SATIR gezildi'
```

Sınav İçeriği

- ER Diyagramları
- Normalizasyon
- Veri Tabanı Nesneleri
- T-SQL Sorguları
 - Insert
 - Update
 - Delete
 - Select
 - Çoklu Select
 - İç-içe select
 - Join ifadeleri
- Constraints
- İndeks
- View



