

Veri Tabanı Yönetimi ve Modellemesi

HAFTA 5

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. 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

CREATE

- Database
- Schema
- Table
- Index
- View
- UDF
- Store Procedure
- Trigger
- Users

Create

- create database Db_Name

- CREATE DATABASE [DbAnimalClinic] ON PRIMARY

(NAME = N' DbAnimalClinic', FILENAME = N'PATH\ DbAnimalClinic.mdf' , SIZE = 3072KB ,
MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB)

LOG ON

(NAME = N' DbAnimalClinic _log', FILENAME = N'PATH\ DbAnimalClinic.ldf' , SIZE = 1024KB ,
MAXSIZE = 2048GB , FILEGROWTH = 10%)

GO

Create

○ CREATE TABLE Table_Name
(columnName valueType,
columnName valueType,
columnName valueType NOT NULL,...
PRIMARY KEY (columnName))

Create

```
CREATE TABLE [dbo].[tbl_Staff](  
    [Id] [int] IDENTITY(1,1) NOT NULL,  
    [Name] [nvarchar](50) NULL,  
    [Surname] [nvarchar](50) NULL,  
    [BirthDate] date NULL,  
    [StartDate] date NULL,  
    StaffRoleId int NULL,  
    Salary float null,  
    CONSTRAINT [PK_tbl_Staff] PRIMARY KEY CLUSTERED  
    (  
        [Id] ASC))
```

GO

ALTER

- ALTER TABLE Table_Name ADD (COLUMN) column_Name dataType;
- ALTER TABLE Table_Name DROP COLUMN column_Name

```
ALTER TABLE [dbo].[tbl_Staff] WITH CHECK ADD CONSTRAINT  
[FK_tbl_Staff_tbl_StaffRole] FOREIGN KEY([StaffRoleId])  
REFERENCES [dbo].[tbl_StaffRole] ([Id])
```

DROP

- DROP DATABASE db_Name
- DROP table tbl_Name
- ALTER TABLE Table_Name DROP COLUMN column_Name
- IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].[FK_tbl_Staff_tbl_StaffRole]') AND parent_object_id = OBJECT_ID(N'[dbo].[tbl_Staff]'))
ALTER TABLE [dbo].[tbl_Staff] DROP CONSTRAINT [FK_tbl_Staff_tbl_StaffRole]
- IF EXISTS (SELECT * FROM sys.foreign_keys WHERE object_id = OBJECT_ID(N'[dbo].[FK_tbl_Staff_tbl_StaffRole]') AND parent_object_id = OBJECT_ID(N'[dbo].[tbl_StaffRole]'))
ALTER TABLE [dbo].[tbl_StaffRole] DROP CONSTRAINT [FK_tbl_Staff_tbl_StaffRole]
- IF EXISTS (SELECT * FROM sys.objects WHERE object_id = OBJECT_ID(N'[dbo].[tbl_StaffRole] ') AND type in (N'U'))
DROP TABLE [dbo].[tbl_StaffRole]

TRUNCATE

- Truncate table tbl_Name
- Truncate table tbl_Staff

SELECT

- Select * from tbl_Name
 - Staff tablosundaki verileri getiren sorgu:
 - Select * from tbl_Staff
- Select columnName,... From tbl_Name
 - Staff tablosundaki ad ve soyad bilgilerini getiren sorgu:
 - Select Name, Surname from tbl_Staff

Top

- Select Top 10 * from tbl_Name
 - Staff tablosundaki ilk 10 personeli getiren soru:
 - Select top 10 * from tbl_Staff

Where

- Select * from tbl_Name Where Statemen(s)
 - Staff tablosunda adresi olmayan personeli getiren sorgu:
 - Select * from tbl_Staff where Address is NULL
 - Staff tablosunda Id'si 2 olan personeli getiren sorgu:
 - select * from tbl_Staff where id=2
- 2020 yılından itibaren işe başlayan kişilerin bilgileri :
 - select * from tbl_Staff where startdate>='01.01.2020'

Koşul İfadeleri

○ =, >, <, >=, <=, <>, !=, LIKE

- Staff tablosunda Id'si 2 olan personeli getiren sorgu:
 - `select * from tbl_Staff where id=2`
- 2020 yılından itibaren işe başlayan kişilerin bilgileri :
 - `select * from tbl_Staff where startdate>='01.01.2020'`
- Staff tablosunda staf rolü 2 dışında olan kişiler:
 - `Select * from tbl_Staff where staffRoleId!=2`
 - `Select * from tbl_Staff where staffRoleId<>2`
- İsminde 'AK' geçen kişiler:
 - `select * from tbl_Staff where Name Like 'AK'`
 - `select * from tbl_Staff where Name Like '%AK'`
 - `select * from tbl_Staff where Name Like 'AK%'`
 - `select * from tbl_Staff where Name Like '%AK%'`

AND, OR, NOT

- SELECT columnName, ... FROM tbl_name WHERE condition1 AND condition2 AND condition3 ...;
 - İşe Başalama tarihi 01.01.2019 dan büyük eşit ve Staf rolü 3 olan Kişiler:
 - select * from tbl_Staff where startdate >='01.01.2019' and staffRoleId=3
- SELECT columnName, ... FROM tbl_name WHERE condition1 OR condition2
 - İşe başlama tarihi 01.01.2019 dan büyük eşit veya staf rolü 3 olan kişiler :
 - select * from tbl_Staff where startdate >='01.01.2019' or staffRoleId=3
- SELECT columnName, ... FROM tbl_name WHERE NOT Condition
 - Staf tablosunda adresi olan kişiler:
 - Select * FROM tbl_Staff where Address is NOT NULL

DISTINCT

- Select distinct columnName from tbl_Name
 - Staf rolü olan kişilerin tekrarlamayacak şekilde StafRolü Id leri:
 - Select distinct StaffRoleId FROM tbl_Staff

ORDER BY

- SELECT columnName, ... FROM tbl_name ORDER BY column1, column2, ... ASC|DESC
 - İşe Başalama tarihi 01.01.2019 dan büyük eşit Kişilerin ismine göre sıralı listelenmesi:
 - select * from tbl_Staff where startdate >='01.01.2019' order by Name
 - İsme göre artan sıralı gösterimi
 - select * from tbl_Staff where startdate >='01.01.2019' order by Nameasc
 - İsme göre azalan sıralı gösterimi
 - select * from tbl_Staff where startdate >='01.01.2019' order by Namedesc

GROUP BY

- `SELECT columnName,... FROM tbl_name WHERE condition
GROUP BY column_name(s) ORDER BY column_name(s);`
- Kişiler arasında hangi Staff türünden kaçar adet kişi bulunmaktadır?
 - `Select StaffRoleId, COUNT(Id) AS KisiSayisi from tbl_Staff group by StaffRoleId order by StaffRoleId`
 - `Select StaffRoleId, KisiSayisi =COUNT(Id) from tbl_Staff group by StaffRoleId order by StaffRoleId`

T-Sql ifadeleri

- ABS: Mutlak değer
 - `select ABS(-3)`
- CEILING/FLOOR/ROUND: Yuvarlama işlemi
 - `select CEILING(1.34)`
 - `select FLOOR(1.34)`
 - `select ROUND(1.34,1)`
- RAND: [0,1] arasında rastgele değer üretir
 - `select RAND()`
- SQRT: Karekök alma
 - `select SQRT(4)`

T-Sql ifadeleri

- Count

- select COUNT(*) from tbl_Staff

- Max

- select MAX(Id) from tbl_Staff

- Min

- select min(Id) from tbl_Staff

T-Sql ifadeleri

- CHAR

- select CHAR(65)

- CHARINDEX

- select CHARINDEX('l','ali veli',1)

- LEFT

- Select LEFT('ali veli',3)

- RIGHT

- select RIGHT('ali veli',2)

- LEN

- select LEN('ali veli')

- UPPER

- select UPPER('ali veli')

- LOWER

- select LOWER('ALİ VELİ')

- LTRIM

- select LTRIM(' ali veli')

- RTRIM

- select RTRIM('ali veli ')

- REPLACE

- select REPLACE('ali veli','veli','yılmaz')

- REVERSE

- select REVERSE('ali')

- SUBSTRING

- Select SUBSTRING('ali veli',3,2)

T-Sql ifadeleri

○ GETDATE()

- select GETDATE()
- select CONVERT(varchar,GETDATE(),105)

○ DATEPART

- select DATEPART(wk,getdate())/ select DATEPART(dy,getdate()) / select DATEPART(m,getdate())

○ DAY/MONTH/YEAR

- select DAY(getdate()) /select DAY(Convert(datetime,'18.12.2018',104))
- select MONTH(GETDATE())
- select YEAR(GETDATE())

○ DATEADD

- select DATEADD(day,3,GETDATE())
- select DATEADD(MONTH,3,GETDATE())
- select DATEADD(YEAR,3,GETDATE())

T-Sql ifadeleri

○ DATEDIFF

- `select DATEDIFF(day,Convert(datetime,'16.10.2019',104),convert(datetime,'1.11.2019',104))`
- `select DATEDIFF(MONTH,Convert(datetime,'16.10.2019',104),convert(datetime,'1.11.2019',104))`
- `select DATEDIFF(year,Convert(datetime,'16.10.2019',104),convert(datetime,'1.11.2019',104))`

○ DATENAME

- `SELECT DATENAME(M, '2017/08/25') AS DatePartString;`

IN - NOT IN

- IN

- Ali, Büşra ve Ceren adındaki kişilerin bilgileri:

- `select * from tbl_Staff where Name in ('Ali','Büşra','Ceren')`

- NOT IN

- Ali, Büşra ve Ceren adındaki kişilerin haricindeki kişilerin bilgileri:

- `select * from tbl_Staff where Name NOT in ('Ali','Büşra','Ceren')`

Çoklu Sorgular

○ IN

- `SELECT * FROM tbl_StaffRole Where Id IN (SELECT StaffRoleId FROM tbl_Staff)`

○ EXISTS

- Personellerin mevcut olan görev türleri:
- `select * from tbl_StaffRole where EXISTS(select StaffRoleId from tbl_Staff where StaffRoleId = tbl_StaffRole.Id)`

○ NOT EXISTS

- İşe alımlar arasında tanımlı görev türlerinden henüz hiç personel alınmamış olan türler:
- `select * from tbl_StaffRole where NOT EXISTS(select StaffRoleId from tbl_Staff where StaffRoleId = tbl_StaffRole.Id)`

Çoklu Sorgular

- AS
 - Toplam Kişi Sayısı
 - `Select COUNT(Id) AS ToplamKisiSayisi from tbl_Staff`

Çoklu Sorgular

- Klasik Join
- Inner Join
- Outer Join
- Cross Join

Klasik Join

- `SELECT * FROM tbl_StaffRole Where Id IN
(select StaffRoleId from tbl_Staff)`

INNER JOIN

- Select * from tbl_Name1 [INNER] JOIN tbl_Name2
ON tbl_Name1 .ColumnName= tbl_Name2.ColumnName

- Select * from tbl_StaffRole **INNER JOIN** tbl_Staff **ON** tbl_StaffRole.Id = tbl_StaffRole .
StaffRoleId

OUTER JOIN

○ LEFT

- Select * from tbl_StaffRole **LEFT JOIN** tbl_Staff **ON** tbl_StaffRole.Id = tbl_StaffRole . StaffRoleId

○ RIGHT

- Select * from tbl_StaffRole **RIGHT JOIN** tbl_Staff **ON** tbl_StaffRole.Id = tbl_StaffRole . StaffRoleId

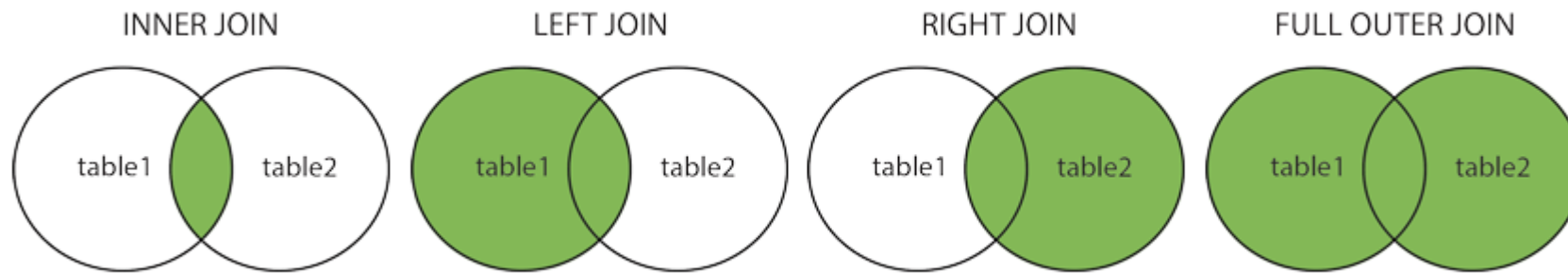
○ FULL

- Select * from tbl_StaffRole **FULL JOIN** tbl_Staff **ON** tbl_StaffRole.Id = tbl_StaffRole . StaffRoleId

CROSS JOIN

- `SELECT * from tbl_Name1 CROSS JOIN tbl_Name2`
- `Select * from tbl_StaffRole CROSS JOIN tbl_Staff`

JOIN



UNION

○ SELECT * FROM tbl_Name1

UNION

SELECT * FROM tbl_Name2

○ SELECT * FROM tbl_StaffRole

UNION

SELECT * FROM tbl_VaccineType

INSERT

- Insert into tbl_name (columnName,...) values (valuesofData,...)
 - insert into tbl_Staff(Name, SurName) VALUES ('Ali', 'AK')
- Insert into tbl_name (columnName,...) values (valuesofData,...)
 - insert into tbl_Staff VALUES ('Büşra', 'Yıldız', '2000.02.16', '2018.01.01', 1, 20000, NULL, NULL)
- Select columnName,... INTO new_Tbl from tbl_name
 - select * into tbl_tmpStaff from tbl_Staff

UPDATE

- Update tbl_name

Set columnName=newValue

Where statement

- UPDATE tbl_Staff SET

Address='Denizli' where Id=7

DELETE

- DELETE FROM tblName WHERE statement(s)
 - Delete from tbl_Staff where Id=7

