



VERİ TABANI VE YÖNETİM SİSTEMLERİ

Dr. Öğretim Üyesi
Alper Talha KARADENİZ



GENEL
DERS
İÇERİĞİ

1. SQL Veri Tipleri

2. SQL

3. DML

4. Gruplama Kümeleme Fonksiyonları

5. DDL

6. DCL

7. SQL Gruplandırma





Aşağıdaki linkleri kullanarak gerekli kurulumları yapınız.

SQL Server Management Studio

SQL Server'dan Azure SQL
Veritabanı'na kadar SQL
altyapısını birleşik bir ortamda
sorunsuz bir şekilde yönetin.



SQL Kurulumu

<https://www.microsoft.com/tr-tr/sql-server/sql-server-downloads>

Northwind Türkçe Veri Seti

<https://www.gencayyildiz.com/blog/northwind-turkce-versiyonu/>



SQL Veri Tipleri



Tablolar oluşturulurken alanların saklayacağı verilerin tiplerinin belirtilmesi gerekir. Girilecek veri, metin, tam sayı, ondalıklı sayı ya da tarih olabilir



1) Metinsel Veri Tipleri



Veritabanlarında ve programlama dillerinde metinsel veri tipleri, harf, rakam, sembol gibi karakterleri içeren verileri saklamak için kullanılır.

char: Sabit uzunlukta metin saklar. Tanımlanan karakter uzunluğu kadar yer ayırır.

Örnek: CHAR(10) → "Ali" yazarsan arkaya 7 boşluk eklenir.

varchar: Değişken uzunlukta metin saklar. Sadece girilen karakter kadar yer kaplar.

Örnek: VARCHAR(50) → "Ali" yazarsan sadece 3 karakterlik yer kaplar.

text: Çok büyük boyutlardaki metinleri (makaleler, açıklamalar vb.) saklamak için kullanılır.

nchar - nvarchar - ntext: Unicode karakterleri (çok dilli metinler: Türkçe, Çince, Arapça vs.) destekler. Verileri Unicode formatında sakladıkları için farklı dil karakterleri bozulmaz.

2) Sayısal Veri Tipleri



bit: Bir byte uzunluğunda tam sayı veri tipidir. Genellikle evet/hayır şeklinde mantıksal bilgileri tutmak için kullanılır.

int: 4 byte büyüklüğünde, -2 milyar / +2 milyar arasında değer tutabilen tam sayı veri tipidir.

bigint: 8 byte büyüklüğünde -2^{63} ve 2^{63} arasında değer tutabilen tam sayı veri tipidir

smallint: 2 byte büyüklüğünde -32.768 ve 32.768 arası değer alabilen tam sayı veri tipidir. ***tinyint***: 1 byte büyüklüğüne sahip, 0–255 arası tam sayı veriler için kullanılan tam sayı veri tipidir. ***float, real***: Çok büyük ve çok küçük kesirli sayılar için kullanılan veri tipidir. Boyutu 4 ile 8 byte arasındadır

3) Binary Veri Tipleri



binary: 1 ve 0 ları temsil eden ikilik taban veri tipidir. Sabit uzunluklu veri tiplerinde kullanılırlar. 8000 byte'a kadar depolama yapabilir.

varbinary: Binary'den farklı olarak girilen karakter kadar yer kaplar. Bu yüzden uzunlukların değişken olduğu durumlarda tercih edilir.

image: Resim dosyalarını saklamak için kullanılır. En fazla 2 GB'a kadar veri depolar. Bunun yerine varbinary (MAX) kullanılması tercih edilir

4) Parasal Veri Tipleri



money: 8 byte boyutunda, yaklaşık- 2^{64} ile 2^{64} arasında parasal değerleri tutmak için kullanılır. 4 basamağa kadar duyarlı ondalık tipli verileri saklar. *smallmoney*: 4 byte uzunluğunda yaklaşık-214.000 ile 214.000 arası parasal değerleri tutmak için kullanılır. Money tipinde olduğu gibi 4 basamağa kadar duyarlı ondalık tipli verileri saklarken kullanılır

5) Tarih-Zaman Veri Tipleri



date: Tarihleri YYYY-AA-GG (yıl-ay-gün) formatında saklar. 3 byte uzunluğunda veri tipidir.

smalldatetime: Tarih ve zaman verilerini yıl-ay-gün ve saat-dakika-saniye-salise şeklide saklar. 4 byte uzunluğunda veri tipidir.

datetime: YYYY-AA-GG şeklinde tarih ve zaman verilerini tutan 8 byte uzunluğunda veri tipidir. 1 Ocak 1753–31Aralık 9999 arası veriler için kullanılır.

time: Sadece saat verilerini saat-dakika-saniye-salise (7 basamaklı) şeklinde saklayan , boyutu kullanıcı tarafından değiştirilebilen 3–5 byte arası yer kaplayan veri tipidir.

datetimeoffset: Ülkelere göre değişen zaman farkını hesaplayıp tutarken kullanılır.



SQL (Structured Query Language)

SQL bir programlama dili değildir. Yapısal sorgulama dilidir. Kullanıcılar SQL aracılığı ile veritabanlarına iletişim sağlarlar. İlişkisel veritabanı yönetim sistemlerinin hepsi bu dili kullandığı için bir standart haline gelmiştir. SQL komutları kullanılarak, veritabanına kayıt ekleme, kayıt silme, kayıt değiştirme, tablo oluşturma gibi birçok işlem gerçekleştirilir.





SQL (Structured Query Language)

Standart SQL ifadelerinde fonksiyon döngü ve karşılaştırma ifadeleri gibi programlamaya yönelik ifadeler yoktur. SQL sorgulama dilinin kullanıldığı bazı bilindik veritabanı işletim sistemleri: Oracle, Sybase, Microsoft SQL Server, Access vb. dir. SQL komutları ***DML***, ***DDL*** ve ***DCL*** olmak üzere üç kategoride incelenmektedir.





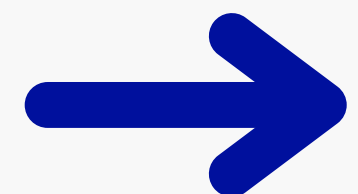
DML (Data Manipulation Language)



Veri işleme dilidir.

- Bu kategorideki SQL komutları veriler üzerinde işlemler yapmaya yönelik ifadeleri içermektedir.
- Tablolara ait kayıtlar için sorgulama, ekleme, silme ve değiştirme gibi işlemler yapılabilmektedir.

- SELECT → Sorgu için kullanılır
- INSERT → Yeni bir kayıt yapmak için kullanılır
- UPDATE → Kayıt üzerinde değişiklik yapmak için kullanılır
- DELETE → Kayıt silmek için kullanılır





SELECT

- SELECT ifadesi ile yazılan sorgular, veritabanından kayıt getirmemize yarar.
- Veritabanında en çok kullanılan ifadedir.
- Yazım kuralı:

SELECT sütun_adı

FROM tablo_adı

WHERE seçim_kriteri

NOT: Eğer yapacağımız sorguda herhangi bir koşul yoksa WHERE ifadesi yazılmaz. Bu durumda tüm kayıtlar seçilir.



SELECT

SELECT ad FROM personel → *Personel tablosundaki tüm adları sırası ile getirir.*

*SELECT*FROM personel* → veritabanında kayıtlı olan bütün kayıtları ekrana getirir.

*SELECT*FROM öğrenciler*

WHERE adres LIKE 'S'* → öğrenciler tablosunda adresi S harfi ile başlayan öğrencileri listeler.

NOT: Yıldız *S şeklinde olsaydı s harfi ile biten verileri listelerdi.



SELECT

ÖRNEK SORU: Maaşı 25000 TL'nin üzerinde olan personelin adı ve soyadını ekrana getiren SQL sorgusunu yazınız.

ÖRNEK SORU: Görevi müdür olan ve maaşı 60000 TL'nin üzerinde olan personeli ekrana getiren SQL sorgusunu yazınız.

ÖRNEK SORU: Personel tablosunda sonu “-han” ile biten isimleri listeleyen komut satırını yazınız.

SELECT



Cevap 1:

```
SELECT P_Adi, P_Soyadi  
FROM Personel  
WHERE Maas>25000
```

Cevap 2:

```
SELECT*FROM personel  
WHERE gorev = 'mudur' AND maas>60000
```

Cevap 3:

```
SELECT P_Adi FROM Personel  
WHERE P_Adi LIKE '*han'
```



Gruplama Kümeleme Fonksiyonları

- SUM(Topla) Fonksiyonu
- AVG(Ortalama) Fonksiyonu
- MAX(EnBüyük)Fonksiyonu
- MIN(EmKüçük) Fonksiyonu
- COUNT(Say) Fonksiyonu



SUM Fonksiyonu



Belirtilen sütundaki bütün değerlerin toplamını alır.

```
Select SUM <sütun_adı> FROM <tablo_adı>;
```

ÖRNEK SORU: Personel tablosunda, cinsiyeti erkek olanların toplam maaşını listeleyen SQL komut satırını yazınız

CEVAP:

```
Select SUM(maas) FROM personel  
WHERE cinsiyet='E'
```

AVG Fonksiyonu



Belirtilen sütundaki bütün değerlerin ortalamasını alır.

```
SELECT AVG <sütun_adı> FROM <tablo_adı>;
```

ÖRNEK SORU: Doğum yeri Samsun olan personellerin maaşlarının ortalamasını hesaplayan komut satırını yazınız.

CEVAP:

```
SELECT AVG(maas) AS ortalama_maas  
FROM personel WHERE dogum_yeri = 'Samsun';
```

MAX Fonksiyonu



Belirtilen sütundaki en büyük değeri verir.

Select MAX(sütun_adı) FROM tablo_adı;

ÖRNEK SORU: En yüksek maaşı alan personeli listeleyen fonksiyonu yazın

CEVAP:

```
SELECT *FROM personel  
WHERE maas = (SELECT MAX(maas) FROM personel);
```

MIN Fonksiyonu



Belirtilen sütundaki en küçük değeri verir

```
Select MIN(sütun_adı) FROM tablo_adı;
```

ÖRNEK SORU: Cinsiyeti erkek olan personeller arasında en düşük maaş nedir?

CEVAP:

```
SELECT MIN(maas) FROM personel where cinsiyet='e'
```



COUNT Fonksiyonu

- Tablo içinde, her hangi bir sayma işlemi gerçekleştirmek için kullanılır.
- COUNT fonksiyonu sorgu sonucu dönen kayıt sayısını verir.
- Eğer parametre olarak “*” girilirse tablodaki tüm kayıt sayısını verir.
- Parametre olarak bir sütun adı verilirse, o sütundaki içeriği NULL (boşluk) olmayan tüm kayıt sayısını verir.

```
Select COUNT(sütun_adı) FROM tablo_adı;
```




INSERT

- Yeni bir kayıt eklemek için kullanılır.
- Primary key alanı otomatik olarak doldurulduğu zaman o alana değer girilmez. Ancak otomatik olarak doldurulmuyorsa değer girilmek zorundadır.
- Yazım kuralı:

INSERT INTO yonetici(adi,soyadi)
VALUES('Alper','Karadeniz')

NOT: INSERT cümlesi yazarken sütun isimleri yazmak zorunda değiliz. Ancak bu durumda tabloda bulunan bütün sütunlara **sırası ile** veri girişi yapılmalıdır.



DELETE

Silme işlemi yapar.

DELETE FROM notlar WHERE ders_yılı = '2022-2026'

Notlar tablosundaki 2022-2026 yılına ait tüm kayıtları silen SQL kodu

DELETE FROM notlar

Notlar tablosundaki tüm kayıtları siler.



UPDATE

Güncelleme işlemi yapar.

UPDATE personel SET maas=maas*110/100;

Personel tablosundaki maaş sütununda bulunan tüm verileri %10 artırdı.

UPDATE personel SET maas=maas*110/100 WHERE adi like 'Ali';

Personel tablosunda adı Ali olan verilerin maaşını %10 artırdı.

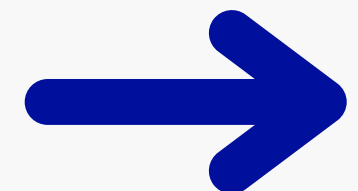
DDL (Data Definition Language)



Veri tanımlama dilidir.

- Bu kategorideki SQL komutları ile veritabanları, tablo, görünüm ve indekslerin oluşturulması, silinmesi ve değişikliklerin yapılması gibi işlemler yapılabilmektedir.
- Verinin ne olduğundan çok tutulan veri tipiyle ilgilenir.
- “Veritabanında hangi tablolar olacak, tablolarda hangi alanlar olacak, alanların türleri ne olacak?” İşte bu soruların cevaplarıyla DDL ilgilenir.

- CREATE → Nesne oluşturmak için kullanılır
- ALTER → Nesneler üzerinde değişiklik yapmak için kullanılır
- DROP → Nesneleri silmek için kullanılır





CREATE

Database, tablo, index, view, vb. veri tabanı objelerini yaratmada kullanılan komuttur. SQL komutları ile veri tabanında işlem yapılabilmesi için önce veri tabanı sonra da veri tabanında kullanılacak tablolar tanımlanmalıdır

CREATE DATABASE sirket → sirket isimli bir veritabanı oluşturur.

CREATE TABLE departman → departman adında bir tablo oluşturur.

ÖRNEK SORU: personel adında bir tablo oluşturun. Sizden istenen bu tablonun içeriğinde PersonelID, P_Adi, P_Soyadi, Maas, P_DepartmanID, Cinsiyet sütunları olmasıdır.

CREATE



CEVAP:

```
CREATE TABLE Personel(  
    PersonelID int NOT NULL,  
    P_Adi varchar (30) NOT NULL,  
    P_Soyadi varchar (30) NOT NULL,  
    Maas decimal(10,2) NOT NULL DEFAULT (0),  
    P_DepartmanID int NOT NULL,  
    Cinsiyet char(1) NULL,  
    CONSTRAINT PK_Personel PRIMARY KEY CLUSTERED(PersonelID),  
    CONSTRAINT CK_Personel CHECK (Cinsiyet = 'K' or Cinsiyet = 'E')  
)
```



DROP

Silme işlemi yapar.

```
DROP DATABASE şirket
```

→ Adı belirtilen veritabanını silmek için kullanılır.

```
DROP TABLE Personel
```

→ Adı belirtilen tabloyu silmek için kullanılır.



ALTER

Daha önceden oluşturulmuş bir tablonun yapısını değiştirmek için ALTER TABLE komutu kullanılır.

```
ALTER TABLE Personel  
ADD baba_adi nvarchar(20)
```

→ Sonradan sütun eklemek için

```
ALTER TABLE Personel  
ADD PRIMARY KEY (P_DepartmanID)
```

→ Sonradan primary key eklemek için

```
ALTER TABLE Personel DROP Column baba_adi
```

Tabloda ismi belirtilen alanı silmek için

DCL (Data Control Language)



Veri kontrol dilidir.

- Bu kategorideki SQL komutları ile veritabanlarına yönelik olarak kullanıcılara erişim yetkilerinin verilmesi veya engellenmesi gibi işlemler yapılabilmektedir.

- GRANT → Bir kullanıcıya belirli bir işlem yapma izni verir
- DENY → Bir kullanıcıya özellikle bir yetkinin verilmemesini sağlar
- REVOKE → Daha önce verilmiş bir yetkiyi geri alma komutudur



CREATE

Database, tablo, index, view, vb. veri tabanı objelerini yaratmada kullanılan komuttur. SQL komutları ile veri tabanında işlem yapılabilmesi için önce veri tabanı sonra da veri tabanında kullanılacak tablolar tanımlanmalıdır.

```
CREATE DATABASE şirket
```

```
CREATE TABLE Personel(...)
```



USE



Önceden yapılmış veri tabanını aktif hale getirir

```
use sirket
```





ALTER TABLE

Daha önceden oluşturulmuş bir tablonun yapısını değiştirmek için ALTER TABLE komutu kullanılır.

```
ALTER TABLE Personel ALTER COLUMN Maas INT
```

NOT: INT veri tipi örnek olarak verilmiştir. İstedığınız veri tipini girerek güncelleme yapabilirsiniz.



DROP



Bir Tabloyu silmek için DROP deyimi kullanılır.

```
DROP TABLE Personel
```

Tablodan bir alan silmek için:

```
ALTER TABLE Personel  
DROP COLUMN baba_adi
```



TRUNCATE



TRUNCATE komutu tablo yapısını deęiřtirmeden, tablo içinde yer alan tüm verileri tek komutla silmenizi saęlar.

```
TRUNCATE TABLE Personel
```



SQL Gruplandırma



GROUP BY

SQL'de verileri gruplamak için kullanılır. Genellikle toplama fonksiyonlarıyla (COUNT, SUM, AVG, MIN, MAX) birlikte kullanılır. SELECT sütun_adı, FONKSIYON(sütun) FROM tablo GROUP BY sütun_adı;

ÖRNEK SORU: Verileri cinsiyete göre gruplayan ve her grup için ortalama maaşı gösteren fonksiyon nedir?

CEVAP:

```
SELECT cinsiyet, AVG(maas) AS ortalama_maas  
FROM personel  
GROUP BY cinsiyet;
```

SQL Gruplandırma



HAVING

HAVING, GROUP BY ile birlikte kullanılır ve gruplar üzerinde filtreleme yapar.

- WHERE → satırları filtreler
- HAVING → grupları filtreler

ÖRNEK SORU: Ortalama maaşı 25000'den fazla olan cinsiyet gruplarını listeleyen fonksiyonu yazınız

CEVAP:

```
SELECT cinsiyet, AVG(maas) AS ortalama_maas  
FROM personel GROUP BY cinsiyet  
HAVING AVG(maas) > 25000;
```