



# VERİTABANI YÖNETİMİ

Dr. Öğr. üyesi  
Ender Şahinaslan

# BÖLÜM -7-

## SQL FONKSİYONLARININ KULLANIMI

The screenshot displays the Microsoft SQL Server Management Studio interface. The Object Explorer on the left shows the database structure for 'TRUVA (SQL Server 10.0.1600 - Truva\Fatih)'. The central area shows a query design view for the 'faculties' table, which is linked to the 'departments' and 'students' tables. The 'departments' table has columns 'departmentID' and 'facultyID'. The 'students' table has columns 'studentID', 'studentName', 'studentSurname', and 'studentNumber'. The 'faculties' table has columns 'facultyID' and 'faculty'. The query design view shows a join between 'faculties' and 'departments' on 'facultyID', and a join between 'departments' and 'students' on 'departmentID'. The query text in the bottom pane is:

```
SELECT TOP (200) students.studentNumber, students.studentName, students.studentSurname, departments.department, faculties.faculty
FROM
faculties INNER JOIN
departments ON faculties.facultyID = departments.facultyID INNER JOIN
students ON departments.departmentID = students.departmentID
```

The bottom pane shows the results of the query, displaying a table with columns 'studentNumber', 'studentName', 'studentSurname', 'department', and 'faculty'. The results are as follows:

studentNumber	studentName	studentSurname	department	faculty
87004007	Veli	Medeni	Bilgisayar Müh...	Mühendislik
80704003	Uğur	Tangül	Computer Engin...	Mühendislik
8074008	Semranur	Halo	Yazılım Mühendisliği	Mühendislik
80704001	Oğuz	Karademirci	Computer Engin...	Mühendislik

The Properties window on the right shows the properties for the 'faculties' table, including its name, TVO Type (Base Table), and the 'Query Designer' tab showing the column list and full name.

# GENEL BAKIŞ...

---

Fonksiyon kavramı,

SQL'de fonksiyonların yeri,

Tek satır fonksiyonlarının kullanımı:

- Karakter fonksiyonları,
- Sayısal fonksiyonlar,
- Tarih fonksiyonları

# 7.1. FONKSİYON NEDİR?

---

- **Fonksiyonlar** veya bir başka deyişle işlevler, SQL kullanımında güçlü özellikler katar.
- **SQL fonksiyonları**, aynen programlama dillerinde olduğu gibi, kullanıcıya bazı özel işlemleri ve hesaplamaları **otomatik olarak** sunar.
- **Kullanıcının**, fonksiyonun adını ve gerekli argümanları tanımlaması yeterlidir. Bu bilgilere dayalı olarak fonksiyon, bir sonuç döndürür (üretir).

## 7.2. SQL FONKSİYONLARI

---

- SQL fonksiyonları **iki ana grup** altında incelenmektedir:
  - **Tek satır** fonksiyonları,
  - **Çift satır** fonksiyonları.
- **Tek satır fonksiyonları**, tablonun bir satırı için uygulanır ve her satır için bir sonuç üretilir.
- **Çoklu satır fonksiyonları** ise, bir grup satıra uygulanır ve sonuç buna göre elde edilir.

## 7.2.1. TEK SATIR FONKSİYONLARI

---

- Tek satır fonksiyonları, tablonun her bir satırına uygulanan fonksiyonlardır.
- **Tek satır fonksiyonlarının** birçok **türü** bulunmaktadır.
  - **Karakter** fonksiyonları,
  - **Sayısal** fonksiyonlar,
  - **Tarih** fonksiyonları,
  - **Dönüştürme** fonksiyonları,
  - **Genel** fonksiyonlar.

## 7.2.1.1. KARAKTER FONKSİYONLARI

---

- Karakter fonksiyonları veri tablolarında, **veri türü** **char**, **varchar**, **nvarchar** olarak tanımlanan alanlar için kullanılmaktadır.
- Harflerden, harf, rakam ve simgelerden oluşan alanlar üzerinde işlem yapılmasını sağlamaktadır.
- Örneğin ad, soyad, adres gibi alanlar bu fonksiyonlarla işlem görebilmektedirler.



## 7.2.1.1. KARAKTER FONKSİYONLARI (DEVAM...)

---

- Karakter fonksiyonları olan;
  - **LEFT:** Metnin solundan n tane harf almak için,
  - **RIGHT:** Metnin sağından n tane harf almak için,
  - **UPPER:** Metni büyük harfe çevirmek için,
  - **LOWER:** Metni küçük harfe çevirmek için,
  - **LTRIM:** Metnin başındaki boşluk karakterleri **atmak** için,
  - **RTRIM:** Metnin sonundaki boşluk karakterleri **atmak** için,
  - **REPLACE:** Bir ifadeyi başka bir ifade ile değiştirmek için,
  - **LEN:** Bir alandaki karakterlerin uzunluklarını bulmak için,
  - **REVERSE:** Bir alandaki bilgileri tersen yazmak için, kullanılmaktadır.



## 7.2.1.1. KARAKTER FONKSİYONLARI (DEVAM...)

---

- Tüm karakter fonksiyonlarının kullanımlarına örnek aşağıda verilmiştir.
- **SELECT** **Adi**,  
    **LEN** (**Adi**) **AS** '**UZUNLUK**',  
    **LEFT** (**Adi**, 3) **AS** '**BAŞTAN**',  
    **RIGHT** (**Adi**, 3) **AS** '**SONDAN**',  
    **UPPER** (**Adi**) **AS** '**BÜYÜK**',  
    **LOWER** (**Adi**) **AS** '**KÜÇÜK**',  
    **LTRIM** (**Adi**) **AS** '**BAŞTAN BEYAZ**',  
    **REPLACE** (**Adi**, '**rda**', '**XXX**') **AS** '**DEĞİŞİKLİK**',  
    **REVERSE** (**Adi**) **AS** '**TERSTEN**'  
**FROM** **personel**

## 7.2.1.1. KARAKTER FONKSİYONLARI (DEVAM...)

- Bu sorgu çalıştırıldığında ortaya çıkan ekran çıktısı aşağıdaki gibidir:

	Adi	UZUNL...	BAŞTAN	SONDAN	BÜYÜK	KÜÇÜK	BAŞTAN BEYAZ	DEĞİŞİKLİK	TERSTEN
1	Erdal	5	Erd	dal	ERDAL	erdal	Erdal	E×××I	ladrE
2	Turgay	6	Tur	gay	TURGAY	turgay	Turgay	Turgay	yagruT
3	Birim	5	Bir	rim	BİRİM	birim	Birim	Birim	miniB
4	Fatih	5	Fat	tiH	FATİH	fatih	Fatih	Fatih	hitaF
5	Neşe	4	Neş	eşe	NEŞE	neşe	Neşe	Neşe	eşeN
6	Fatma	5	Fat	tma	FATMA	fatma	Fatma	Fatma	amtaF

## *7.2.1.2. SAYISAL FONKSİYONLAR*

---

- Sayısal fonksiyonlar dört işlemin yanı sıra **AVG, SUM, STDEV, VAR, MAX, MIN, COUNT, ABS, SQRT, RAND, POWER, CEILING, FLOOR, ROUND, EXP, LOG** ve trigonometrik fonksiyonlar olan **SIN, COS, TAN ve COT** fonksiyonlarını kapsamaktadır.
- Bu fonksiyonları örnekler üzerinde anlatarak devam edeceğiz. Bunun için bir personel tablosu oluşturup ve içerisine yeterli miktarda kayıt girelim.

## ÖRNEK VERİTABANI HAZIRLIK AŞAMASI

---

- Personel tablosunu oluşturmak ve kayıt eklemek için gerekli SQL ifadeleri aşağıda verilmektedir.

SQLQuery1.sql - TR...Truva\Fatih (54))\*

```
CREATE TABLE Personel (  
    personelNo int NOT NULL,  
    adi varchar (50),  
    soyadi varchar (50),  
    cinsiyet bit,  
    gsmNo varchar(50),  
    maasi int,  
    bolumID int)
```

TRUVA.satıcı - dbo.Personel SQLQuery1.sql - TR...Truva\Fatih (54))\*

```
INSERT INTO Personel (personelNo, adi, soyadi, cinsiyet, gsmNo, maasi, bolumID)  
VALUES (903, 'Cumhur', 'Yücalar', 1, 5333401903, 1800, 4)
```

## AVG ( ) FONKSİYONU

---

- Herhangi bir sütunun içerdiği sayısal değerlerin aritmetik ortalamasını hesaplamak amacıyla; **AVG ( )** fonksiyonu kullanılır. Bu fonksiyonun uygulandığı sütunun veri türü sayısal olmalıdır.
- Örnek-1: Personel maaşlarını göz önüne alalım. Maaşların ortalamasını bulmak için şöyle bir yol izlenir;
  - **SELECT AVG (maasi) AS 'Ortalama'**  
**FROM personel**

## AVG ( ) FONKSİYONU (DEVAM...)

---

- Örnek-2: Personel arasında, **bolumID**'si **5** olanların ortalama maaşı aşağıda gösterildiği biçimde hesaplanabilir;
  - **SELECT** **AVG (maasi)** **AS** 'Ortalama'  
**FROM** personel  
**WHERE** **bolumID = 5**

## SUM ( ) FONKSİYONU

---

- Sütunların içerdiği sayısal değerleri toplamak amacıyla **SUM ( )** fonksiyonundan yararlanılır.
- Örnek-1: Tüm personel maaşlarının toplamını hesaplayan **SELECT** deyimi aşağıda gösterildiği biçimde oluşturulabilir;
  - **SELECT SUM (maasi) AS 'Toplam'**  
**FROM personel**

## SUM ( ) FONKSİYONU (DEVAM...)

---

- Örnek-2: Sadece erkek personele ödenen aylık toplam maaş ne kadardır?
  - **SELECT SUM (maasi) AS 'EToplam Ödenen'**  
**FROM personel**  
**WHERE cinsiyet = 1**



# STDEV ( ) FONKSİYONU

**Standart sapma**, sayısal verilerin aritmetik ortalamalardan farklarının kareli ortalaması olarak bilinir. Bu kavram istatistiksel veri analizlerinde sıkça kullanılır. Söz konusu hesaplamayı yapan **SQL fonksiyonu** ise; **STDEV ( )** 'dir.

The diagram shows the formula for standard deviation: 
$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}}$$
 with the following labels in red: 'veri setindeki her bir veri' (each data point in the set) pointing to  $x_i$ , 'aritmetik ortalama' (arithmetic mean) pointing to  $\bar{x}$ , 'standart sapma' (standard deviation) pointing to  $S$ , and 'örneklem sayısı' (sample size) pointing to  $n-1$ .

- **Örnek-1:** Personel maaşlarının standart sapmasını bulmak istiyoruz. Bunun için **SELECT** deyimi aşağıda gösterildiği biçimde kullanılır;
  - **SELECT STDEV (maasi) AS 'Standart\_Sapma'**  
**FROM personel**

# VAR ( ) FONKSİYONU

- **Varyans**, sayısal değerler arasındaki değişkenliği ölçen bir kavramdır. İstatistiksel veri analizlerinde yaygın biçimde kullanılmaktadır. Standart sapmanın karesi olarak bilinmektedir. Bir dizi sayısal değerın varyansını hesaplamak için **VAR ( )** fonksiyonu kullanılır.
- Örnek-1: Personel maaş bilgilerinin ne ölçüde değişken olduğunu belirlemek istiyoruz;
  - **SELECT** **VAR (maasi)** **AS** 'Varyans'  
**FROM** personel

The diagram shows the formula for variance,  $s^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}$ . It includes the following labels in red:

- veri setindeki her bir veri** (each data point in the data set) pointing to  $x_i$
- aritmetik ortalama** (arithmetic mean) pointing to  $\bar{x}$
- varians** (variance) pointing to  $s^2$
- örneklem sayısı** (sample size) pointing to  $n-1$

## MAX ( ) VE MIN ( ) FONKSİYONLARI

---

- Tablonun içerdiği değerlerin en büyük ve en küçük olanlarını bulmak için **MIN ( )** ve **MAX ( )** fonksiyonları kullanılır. Bu fonksiyonlar herhangi bir veri türüne uygulanabilir.
- Örnek-1: Personel ücretlerini göz önüne alalım. En az ve en yüksek ücretin ne olduğunu öğrenmek istiyoruz. Amacımıza uygun **SELECT** deyimi aşağıda belirtildiği biçimde olabilir;
  - **SELECT** **MIN (maasi)** **AS** 'En Az',  
**MAX (maasi)** **AS** 'En Fazla'  
**FROM** personel

## MAX ( ) VE MIN ( ) FONKSİYONLARI (DEVAM...)

---

- Örnek-2: İşe en son giren personelin işe giriş tarihini belirlemek için **SELECT** deyimi şu şekilde düzenlenebilir;
  - **SELECT** **MAX (giris\_tarihi)**  
**FROM** personel
- Örnek-3: Personel isimlerini alfabetik olarak sıraladığımızda birinci sırada hangi ismin elde edileceğini şu şekilde öğrenebiliriz;
  - **SELECT** **MIN (Adi)**  
**FROM** personel

## MAX ( ) VE MIN ( ) FONKSİYONLARI (DEVAM...)

---

- Örnek-4: En düşük maaş alan personel kimdir?  
Bu sorgunun yapılabilmesi için bir alt sorguya daha ihtiyaç vardır. Bu durumda SQL ifadesi aşağıdaki gibi olacaktır.
  - **SELECT** adi, soyadi  
**FROM** personel  
**WHERE** maasi = (**SELECT** **MIN** (maasi)  
**FROM** personel)

# COUNT ( ) FONKSİYONU

---

- Bir tablodaki kayıtların sayılması amacıyla **COUNT ( )** fonksiyonu kullanılır. Bu fonksiyon iki farklı biçimde kullanılabilir;
  - **COUNT (\*)**
  - **COUNT (sütun)**
- **COUNT (\*)** fonksiyonu, **NULL** değerleri de içeren tüm kayıtların sayılmasına neden olur. **WHERE** ile birlikte kullanılırsa, bu koşula uygun tüm kayıtların sayısının öğrenilmesini sağlar.
- İkinci biçim tercih edilirse, yani **COUNT (sütun)** biçiminde kullanılırsa, söz konusu sütunda **NULL** değerler içermeyen tüm kayıtların sayılmasına neden olur.

## COUNT ( ) FONKSİYONU (DEVAM...)

---

- Örnek-1: 30 numaralı bölümde görev yapan kaç tane personel olduğunu bulmak istiyoruz. **SELECT** deyimi şu şekilde düzenlenir;
  - **SELECT COUNT (\*) AS 'Personel Sayısı'**  
**FROM** personel  
**WHERE** bolumID = 30

## ABS ( ) FONKSİYONU

---

- Parametre olarak verilen **sayısal veri türündeki değerin mutlak değerinin** alınmasını sağlamaktadır. Seçilen nitelikteki bütün değerlerin negatif yada pozitif olduğuna bakmadan pozitif olarak gösterilmesini sağlar.

■ **SELECT ABS (-15) AS 'Argüman-1'**

**ABS (15) AS 'Argüman-2'**

■ Ekran Çıktısı:

Argüman-1	Argüman-2
15	15



## SQRT ( ) FONKSİYONU

---

- Parametre olarak verilen sayısal veri türündeki değerin **karekökünü hesaplar.**

■ **SELECT** maasi, **SQRT (maasi)** **AS** 'Karekökü'  
**FROM** personel

■ Ekran Çıktısı:

maasi	Karekökü
1 200	34,641
1 500	38,729

## RAND ( ) FONKSİYONU

---

- Rastgele sayı üretimini sağlayan bir fonksiyondur. **0 ile 1 arasında rasgele** değer üretir.

- **SELECT RAND ( ) AS 'Rastgele Sayı'**

- Ekran Çıktısı:

Rastgele Sayı
0,2324169653486

## RAND ( ) FONKSİYONU (DEVAM...)

---

- 0 ile 100 arasında rastgele sayı üretilmek istenirse aşağıdaki SQL ifadesi kullanılabilir.

■ **SELECT RAND ( ) \* 100**

**AS '0-100 Arası Sayı'**

■ Ekran Çıktısı:

0-100 Arası Sayı
81,6071018002395

## POWER (X, Y) FONKSİYONU

---

- $X^Y$  şeklindeki sayıların hesaplanmasında kullanılmaktadır. Yani  $3^3 = 3 \times 3 \times 3 = 27$  gibi 3'ün 3' üncü dereceden kuvvetini hesaplamaktadır.

- **SELECT POWER (3, 3) AS 'Kuvveti'**

- Ekran Çıktısı:

Kuvveti
27

# CEILING ( ) FONKSİYONU

---

- Parametre olarak verilen sayısal değeri bir üst sayıya yuvarlar. MS Access’ te kullanılmayan bir fonksiyondur.

- **SELECT CEILING (5.2) AS 'Pozitif Yuvarlama'**

- Ekran Çıktısı:

Pozitif Yuvarlama
6

- **SELECT CEILING (-7.9) AS 'Negatif Yuvarlama'**

- Ekran Çıktısı:

Negatif Yuvarlama
-7

## FLOOR ( ) FONKSİYONU

---

- Parametre olarak verilen sayısal değeri bir alt sayıya yuvarlar. CEILING fonksiyonunun tam tersi yuvarlama işlemini gerçekleştirir.

- **SELECT FLOOR (5.2) AS 'Pozitif Yuvarlama'**

- Ekran Çıktısı:

Pozitif Yuvarlama
5

- **SELECT FLOOR (-7.9) AS 'Negatif Yuvarlama'**

- Ekran Çıktısı:

Negatif Yuvarlama
-8

# ROUND ( ) FONKSİYONU

---

- Normal yuvarlama işlemini gerçekleştirir. 5.6, 5.8 gibi değerler bir üstü olan 6 ya, 5.4, 5.2 gibi değerlerde bir altı olan 5'e yuvarlanmaktadır.

- **SELECT ROUND (5.2, 0) AS 'Yuvarlama'**

- Ekran Çıktısı:

Yuvarlama
5.0

- **SELECT ROUND (5.6, 0) AS 'Yuvarlama'**

- Ekran Çıktısı:

Yuvarlama
6.0

# TRİGONOMETRİK FONKSİYONLAR

---

- **SIN, COS, TAN, COT:** Trigonometrik fonksiyonlardır. Belirtilen sayısal veri türü içeren alanda sinüs, kosinüs, tanjant ve kotanjant trigonometrik değerlerinin hesaplanmasını sağlamaktadır. Örneğin bir üçgenin bir açısına ait sinüs, kosinüs, tanjant ve kotanjant hesaplamalarına ait örnek uygulaması şöyledir.

- **SELECT** aci, **SIN (aci), COS (aci), TAN (aci), COT (aci)**  
**FROM** Ucgen



# TRİGONOMETRİK FONKSİYONLAR (DEVAM...)

$$a^x = b \Leftrightarrow x = \log_a b$$

esittir  
üzeri

- **EXP, LOG:** Exp üssel bir fonksiyon olmasını yanı sıra veritabanındaki bir değerin kuvvetinin alınmasını sağlar.

**e değeri 2.71** dir. Log fonksiyonu ise belirtilen değerin logaritmasının hesaplanmasını sağlar.



▪ **SELECT vize,**

**EXP (vize) AS 'Üssü',**

**LOG (vize) AS 'Logaritması'**

**FROM Ogrenciler**

### *7.2.1.3. TARİH - ZAMAN FONKSİYONLARI*

---

- **Tarih – zaman fonksiyonları**, bir tabloda tarih ve zaman sorgulaması yapıldığında kullanılmaktadır.
- Örneğin bir sipariş tablosunda **belli bir tarih aralığında yapılan siparişler listelenmek** istenildiğinde kullanılması gereken fonksiyonlardır. Bunun yanında bu fonksiyonlar, siparişlerin verilme tarihi, gönderilme tarihi gibi işlemlerin yapılmasında kullanılmaktadır.
- Bir sonraki slaytta **SQL’de kullanılan tarih ve zaman fonksiyonları** detaylı olarak anlatılacaktır.

### 7.2.1.3. *TARİH-ZAMAN FONKSİYONLARI* (DEVAM...)

---

- Parametre olarak verilen bir alandaki tarih bilgisinin;
  - **DAY:** Gün bilgisini,
  - **MONTH:** Ay bilgisini,
  - **YEAR:** Yıl bilgisini,
  - **HOURL:** Saat bilgisini,
  - **MINUTE:** Dakika bilgisini,
  - **SECOND:** Saniye bilgisini,
  - **MILLISECOND:** Milisaniye bilgisini,
  - **WEEK:** Yılın kaçını haftası olduğu bilgisini,
  - **WEEKDAY:** Haftanın kaçın gün olduğu bilgisini verir.

### 7.2.1.3. TARİH-ZAMAN FONKSİYONLARI (DEVAM...)

---

- Tüm tarih – zaman fonksiyonları ile ilgili genel bir örnek aşağıda verilmiştir.

```
■ SELECT GETDATE () AS 'Şimdi',  
DATENAME (YEAR, GETDATE ()) AS 'Yıl',  
DATENAME (MONTH, GETDATE ()) AS 'Ay',  
DATENAME (DAY, GETDATE ()) AS 'Gün',  
DATENAME (WEEK, GETDATE ()) AS 'Hafta',  
DATENAME (WEEKDAY, GETDATE ()) AS 'Haftanın Günü',  
DATENAME (HOUR, GETDATE ()) AS 'Saat',  
DATENAME (MINUTE, GETDATE ()) AS 'Dakika',  
DATENAME (SECOND, GETDATE ()) AS 'Saniye',  
DATENAME (MILLISECOND, GETDATE ()) AS 'Milisaniye'
```

### 7.2.1.3. TARİH-ZAMAN FONKSİYONLARI (DEVAM...)

---

- SQL sorgusu çalıştırıldığında ortaya çıkan sonuçlar aşağıda verilmiştir.

	Şimdi	Yıl	Ay	Gün	Hafta	Haftanın Günü	Saat	Dakika	Saniye	Milisaniye
1	2024-04-14 22:11:43.617	2024	April	14	16	Sunday	22	11	43	636

*TEŞEKKÜRLER...*

*SORULARINIZ ?*

---