

SQL NEDİR?

SQL, İngilizce adıyla “Structured Query Language” (Yapılandırılmış Sorgulama Dili) bir veri tabanı sorgulama dilidir. SQL ile veri tabanına yeni tablolar, kayıtlar ekleyip siler, mevcut veriler üzerinde düzenlemeler ve sorgular yapılabilir. SQL ile ORACLE, db2, Sybase, Informix, MS SQL Server, MS Access gibi veri tabanı yönetim sistemlerinde çalışılabilir. SQL, standart bir veri tabanı sorgu dilidir, bütün gelişmiş veri tabanı uygulamalarında kullanılır.

Transact-SQL (T-SQL)

SQL, düzeltilmesi veya değiştirilmesi istenen bilgileri açıkça belirtmeye izin veren ve yerine getirilebilecek başlıca işlemleri tanımlamamızı sağlayan bir komut takımudur. Bu komutların oluşturduğu yapıya *T-SQL* dili denir. T-SQL ile veri ve sorgulara erişebilir, güncelleyebilir ve ilişkisel veri tabanı sistemi yönetilebilir. T-SQL komutları kullanım amaçlarına göre üç genel kategoriye ayrılır.

Microsoft’un veri tabanı sorgulama dilidir. Transact-SQL, SQL Server ve istemci (client) arasında iletişimi sağlayan SQL sorgulama dilinin gelişmiş bir versiyonudur. Transact Structured Query Language kelimelerinin kısaltmasıdır. T-SQL kullanarak veri tabanına kayıt eklenebilir, silinebilir, güncellenebilir ya da sorgulama ve raporlama yapılabilir. T-SQL ile döngü veya mantıksal işlemler yapmak için bir derleyiciye gerek yoktur.

Bu komutlar, işlevlerine göre şu şekilde ayrılır;

- ** DDL (Data Definition Language):** Veri tanımlama dili
- ** DML (Data Manipulation Language) :** Veri işleme dili
- ** DCL (Data Control Language):** Veri kontrol dili

SQL Veri Tanımlama Dili

(Data Definition Language – DDL)

SQL Veri tanımlama dili verilerin tutulduğu nesneler olan tabloların yaratılmasını, silinmesini ve bazı temel özelliklerinin düzenlenmesini sağlar. En sık kullanılan bazı DDL komutları ve kullanım amaçları aşağıdaki gibidir:

CREATE TABLE:Yeni bir tablo yaratmak
ALTER TABLE:Tabloda değişiklik yapmak
DROP TABLE:Tabloyu silmek
CREATE INDEX:Tabloda dizin oluşturmak

SQL Veri İşleme Dili

(Data Manipulation Language – DML)

SQL veri işleme dili veri girmek, değiştirmek, silmek ve verileri almak için kullanılan DML komutlarının tümüdür. En sık kullanılan DML komutları ve kullanım amaçları aşağıdaki gibidir:

SELECT:Veri seçmek
DELETE:Veri silmek
UPDATE:Veri güncellemek
INSERT:Veri girmek

SQL Veri Kontrol Dili (Data Control Language – DCL)

SQL veri kontrol dili bir veri tabanı kullancısı veya rolü ile ilgili izinlerin düzenlenmesini sağlar. DCL komuları ve fonksiyonları şöyledir:

GRANT:Kullanıcıya yetki verir.

DENY:Kullanıcı, grup veya rolü herhangi bir eylem için eneller.

REVOKE:Daha önce atanmış olan yetki veya engeli kaldırır.

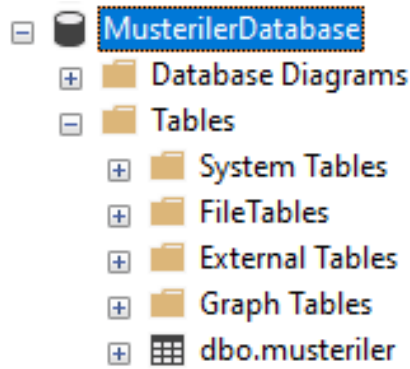
SQL Kullanımı Hakkında

SQL dili veritabanlarındaki tablolara bağlanıp veri eklemek, daha önceden eklenen verileri okumak ve veriler üzerinde işlem yapmak (güncelleme, silme) için kullanılan bir dildir. Sorgulama yapabilmek için öncelikli olarak veritabanımızda ez azından bir tablo olması gerekmektedir. Birden fazla tablo da olabilir. Ancak ister bir tablo olsun isterse birden çok tablo olsun neticede üzerinde işlem yapmak istediğimiz tabloyu belirtmemiz gerekmektedir. Hangi tablo üzerinde çalışacağımız **FROM** ifadesi ile belirtiriz. Bu ifade SQL sorgulama dilinin temelidir ve hemen hemen bütün SQL sorgularında kullanılmaktadır. Hemen hemen dedik çünkü bazı sorgu ifadelerinde kullanılmaz. Mesela yeni bir veritabanı veya tablo oluştururken FROM ifadesi kullanılmaz. FROM ifadesi sadece var olan verilerde sorgulama yaparken kullanılır.

Tek başına FROM ifadesi tabiki yetmez. FROM ile hangi tablo üzerinde işlem yapacağımız belirtiriz ancak ilgili tabloda ekleme işlemi mi, silme işlemi mi veya güncelleme işlemi mi yaptığımızı da belirtmemiz gerekmektedir. Bunun için belli başlı işlemler için hangi özel komutun kullanıldığını aşağıdaki tabloda görebilirsiniz.

Komut	İşlemi
Select	Veritabanından verileri okumak için kullanılır.
Update	Veritabanındaki var olan bir kaydın bilgilerini güncellemede kullanılır. Örneğin yaşadığı şehir alanı İstanbul olanları Ankara'ya çevirmek gibi.
Delete	Veritabanındaki bir kaydı silme için kullanılır.
Insert Into	Veritabanına yeni bir kayıt eklemek için kullanılır.
Create Database	Yeni bir veritabanı oluşturmak için kullanılır.
Alter Database	Mevcut veritabanı özelliklerinde değişiklik yapmak için kullanılır.
Create Table	Veritabanına yeni bir tablo eklemek için kullanılır.
Alter Table	Mevcut tablo özelliklerinde işlem yapmak için kullanılır. Bu komut ile mevcut kayıtlar üzerinde işlem yapılmaz.
Drop Table	Veritabanı içindeki bir tabloyu silmek için kullanılır. Bu işlem ile ilgili tablodaki kayıtlar da silinir.

Derste Yapılan Örnek Veritabanı Tasarımı ve Örnek Sorgular;



MD.MusterilerDatabase - dbo.musteriler SQLQuery6.sql - not conr

	Column Name	Data Type	Allow Nulls
	musteriNo	int	☐
	ad	nvarchar(50)	☒
	soyad	nvarchar(50)	☒
	cinsiyet	nvarchar(5)	☒
	gelir	money	☒
	meslek	nvarchar(50)	☒
	sehir	nvarchar(50)	☒
	yas	int	☒
			☐

MD.MusterilerDatabase - dbo.musteriler		MD.Katalog - dbo.Urunler*		MD.MusterilerDatabase - d				
	musteriNo	ad	soyad	cinsiyet	qelir	meslek	sehir	yas
	1	Batuhan	Ozer	Erkek	5000,0000	SQL ci	Rize	20
	2	Ali	Mali	Erkek	3500,0000	Yazılımcı	Ankara	25
	3	Akın	Deli	Erkek	4000,0000	Siberçi	İstanbul	30
	4	Bediha	Horoz	Kadın	6500,0000	Frontend	İstanbul	35
	5	İrem	Gül	Kadın	2500,0000	Sekreter	Burdur	40
	6	Gürhan	Güray	Erkek	3550,0000	Satın Alma	İzmit	42
▶*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
SELECT *FROM müşteriiler
```

```
SELECT meslek FROM musteriler
```

```
SELECT ad as ADI,soyad as SOYADI,meslek as MESLEĞİ,sehir
FROM musteriler
WHERE cinsiyet='Erkek'
```

```
SELECT ad,soyad,meslek,sehir
FROM musteriler
WHERE gelir>3000
```

```
SELECT ad,soyad,meslek,sehir
FROM musteriler
WHERE not Gelir<=2000
```

```
SELECT ad,soyad,meslek,sehir
FROM musteriler
WHERE sehir in ('İstanbul','İzmit','Ankara')
```

```
SELECT ad,soyad,meslek,sehir,gelir
FROM musteriler
WHERE gelir between 2000 and 5000
```

```
SELECT ad,soyad,meslek,sehir,gelir
FROM musteriler
WHERE ad Like '%A'
```

```
SELECT ad,soyad,meslek,sehir,gelir
FROM musteriler
WHERE soyad Like '___'
```

```
SELECT ad,soyad,meslek,sehir,gelir
FROM musteriler
WHERE soyad Like 's%' and gelir between 1000 and 3000 and cinsiyet='Erkek'
```

```
SELECT DISTINCT sehir FROM musteriler
```

```
SELECT DISTINCT yas,cinsiyet FROM musteriler
```

```
SELECT * FROM musteriler ORDER BY yas desc
```

```
SELECT * FROM musteriler ORDER BY gelir
```

```
SELECT * FROM musteriler ORDER BY yas asc,gelir desc
```

```
SELECT AVG(yas) FROM müşteriler
```

```
SELECT AVG(gelir) FROM musteriler
```

```
SELECT SUM(yas) FROM musteriler
```

```
SELECT MAX(yas) FROM müşteriler
```

```
SELECT MIN(yas) FROM musteriler
```

```
SELECT cinsiyet, COUNT(cinsiyet) as CinsiyetSayısı FROM musteriler
GROUP BY cinsiyet
```

```
SELECT sehir, COUNT(sehir)FROM musteriler
GROUP BY sehir
```

```
SELECT TOP 3 * FROM musteriler ORDER BY musterino desc
```

```
SELECT UPPER(ad),upper(soyad) FROM musteriler
```

```
SELECT LOWER(ad),upper(soyad) FROM musteriler
```

```
SELECT *from musteriler
WHERE gelir > (SELECT AVG(gelir) from musteriler where cinsiyet='Kadın')
```

```
INSERT INTO musteriler(ad,soyad,cinsiyet,gelir,meslek,sehir,yas)
VALUES ( 'Hasan', 'Yılmaz', 'Erkek', 5000, 'Tasarımcı', 'İstanbul', 30)
```

```
UPDATE musteriler SET yas=55 WHERE musterino=1
```

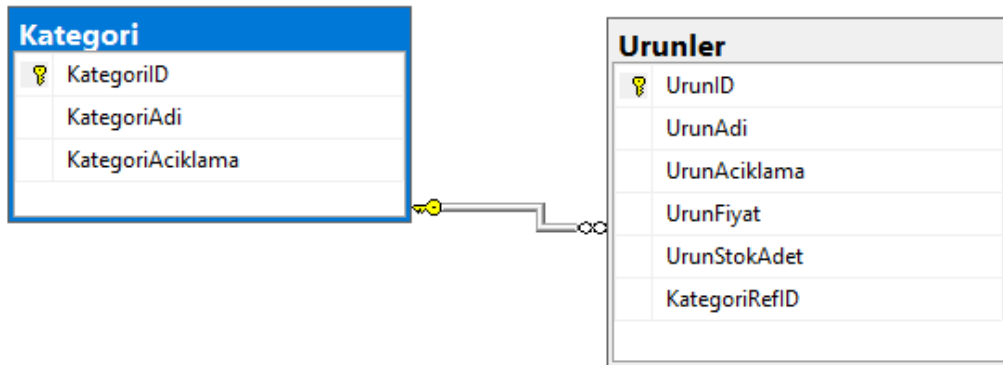
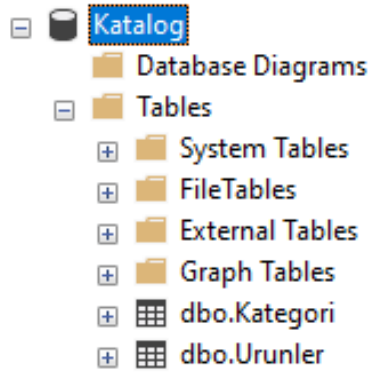
```
UPDATE musteriler SET sehir='Ankara' WHERE sehir='İstanbul'
```

```
UPDATE musteriler SET sehir='Ankara' WHERE musterino=1
UPDATE musteriler SET sehir='İstanbul' WHERE musterino=2
UPDATE musteriler SET sehir='İstanbul' WHERE musterino=3
UPDATE musteriler SET sehir='Rize' WHERE musterino=4
UPDATE musteriler SET sehir='Muş' WHERE musterino=5
UPDATE musteriler SET sehir='Hakkari' WHERE musterino=6
UPDATE musteriler SET sehir='Diyarbakır' WHERE musterino=7
```

```
DELETE FROM musteriler WHERE musterino=7
```

Derste Yapılan 2. Örnek Veritabanı Tasarımı ve Örnek Sorgular;

Katalog isminde bir veri tabanı oluşturuldu. Oluşturulan veritabanı içerisine, Kategori ve Urunler adında iki tablo oluşturuldu. Tablolar birbirlerine bire çok ilişki modeliyle bağlandı.



MD.Katalog - dbo.Kategori	MD.Katalog - Diagram_1*	MD
Column Name	Data Type	Allow Nulls
KategoriID	int	☐
KategoriAdi	nvarchar(30)	☐
KategoriAciklama	nvarchar(50)	☒

MD.Katalog - dbo.Urunler		MD.Katalog - dbo.Kategori	
	Column Name	Data Type	Allow Nulls
	UrunID	int	☐
	UrunAdi	nvarchar(50)	☐
	UrunAciklama	nvarchar(100)	☒
	UrunFiyat	money	☐
	UrunStokAdet	int	☐
	KategoriRefID	int	☐

MD.Katalog - dbo.Kategori*	MD.Katalog - dbo.Urunler	MD.Katalog - dbo.Kategori
KategoriID	KategoriAdi	KategoriAciklama
1	Elektronik	Elektronik eşyaların satıldığı stand
2	Beyaz Eşya	Buzdolabı, fırın, çamaşır makinası
3	Mobilya	Sehpa, masa, koltuk

MD.Katalog - dbo.Urunler*		MD.Katalog - dbo.Kategori*		MD.Katalog - dbo.Urunler		
	UrunID	UrunAdi	UrunAciklama	UrunFiyat	UrunStokAdet	KategoriRefID
	2	Asus Laptop	Çok iyi	5000,0000	36	1
	3	Tablet IOS	BERBAT	3000,0000	20	1
	4	Arçelik Buzdolabı	Dayanıklı	5500,0000	50	2
	5	Samsung Fırın	Sağlam	3600,0000	22	2
	6	Köşe Koltuk	Köşeyi dönenle...	2600,0000	10	3
	7	Sehpa	iyi sehpa	560,0000	25	3

SELECT *FROM Urunler

SELECT *FROM Kategori

SELECT UrunAdi,UrunFiyat FROM Urunler WHERE KategoriRefID=1

/*ÜRÜN FİYATI ORTALAMA ÜRÜN FİYATINDAN YÜKSEK OLAN ÜRÜNLER*/

SELECT *FROM Urunler
WHERE UrunFiyat>(SELECT AVG(UrunFiyat)FROM Urunler)

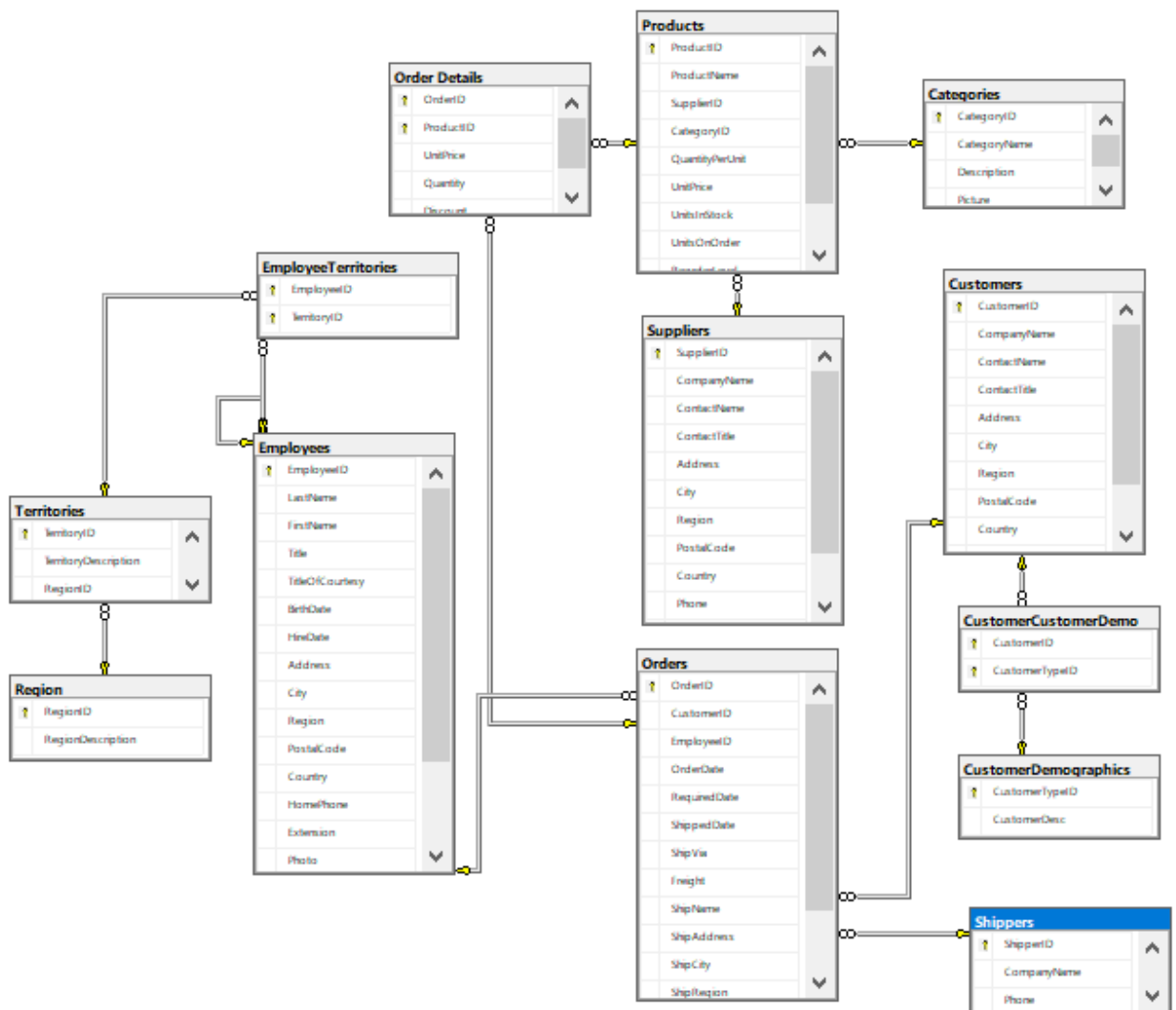
/*ÜRÜN FİYATI ORTALAMA ÜRÜN FİYATINDAN YÜKSEK OLAN ELEKTRONİK ÜRÜNLER*/

SELECT *FROM Urunler
WHERE UrunFiyat>(SELECT AVG(UrunFiyat)FROM Urunler) AND KategoriRefID=1

SELECT UrunAdi, UrunFiyat, UrunStokAdet,(UrunFiyat*UrunStokAdet) AS toplamTutar FROM Urunler

SELECT sum(UrunFiyat*UrunStokAdet) FROM Urunler

Derste Yapılan 3. Örnek Hazır Northwind veritabanı ve Örnek Sorgular;



```
SELECT COUNT(*) FROM [Order Details]
```

```
SELECT * FROM [Order Details]
```

```
SELECT UnitPrice,Quantity, (UnitPrice*Quantity) AS ToplamFİYAT FROM [Order Details]
```

```
SELECT sum(UnitPrice*Quantity) AS TOTALGELİR FROM [Order Details]
```

```
SELECT avg(UnitPrice*Quantity) AS TOTALGELİR FROM [Order Details]
```

```
SELECT ProductName,UnitPrice
FROM Products
WHERE UnitPrice>10 ORDER BY UnitPrice ASC
```

```
SELECT ShipCity, ShipCountry, OrderDate
FROM Orders
WHERE YEAR(OrderDate) = 1997
ORDER BY ShipCity
```

```
SELECT Count(*)
FROM Orders
WHERE YEAR(OrderDate) = 1997
```

