**SQL SORGU YAZMAK 1**

**SQL**🡪 Açılımı: Structured Query Language (Yapılandırılmış Sorgu Dili)

SQL dediğimiz zaman içerisinde sorgulamalar yapabildiğimiz koşullar koyabildiğimiz fonksiyonlar yazabildiğimiz bir mekanizma aklımıza gelir. Xml işaretleme dili iken Sql de bu anlattığımız mekanizmalara sahip C#, Java, C, Python gibi bir dildir.

Tepede Sql vardır. **Sql** dili toplamda iki ana bölüme ayrılır.

**DDL**   **DML**

**DML**🡪Data Manipulation Language: Datanın kendisini manipüle eden dilimiz. (veri ile uğraşır. örnek: oluşturduğumuz tabloya veri eklemek istersek)

* **SELECT :** Veritabanındaki nesneden alan çağırmayı ve gereken bilgiyi görebilmeyi sağlar.
* **UPDATE :** Nesnede bilgi güncellemeyi sağlar.
* **INSERT :** Nesneye alan eklemeyi sağlar.
* **DELETE :** Nesneden alan silmeyi sağlar.

**DDL**🡪Data Definition Language: Verinin saklandığı yapı ile uğraşır (tablo yapısı örnek: database oluşturmak, tablo oluşturmak, kolon oluşturmak)

* **CREATE :** Veritabanı üzerinde bir tablo yaratmak için kullanılır.
* **ALTER :** Tabloda değişiklik yapmak, yeni bir sütun eklemek, sütunun tipini veya uzunluğunu
* **DROP :** Tabloyu fiziksel olarak siler. (verilerle birlikte)
* **TRUNCATE :** Tablodaki verileri siler fakat tablo yapısı kalır.

**ÖNEMLİ**

Sorgu oluşturmak için New Query butonuna bastığımızda Sqlsrver ile ilgili aslında ne olur? Sadece bir sayfa mı açılır?

Çalıştırma1-) NewQuery 🡪Select @@SPID 🡪 Execute **🡪 Çıktı : 53**

Çalıştırma2-) NewQuery 🡪Select @@SPID 🡪 Execute 🡪 **Çıktı : 55**

Çalıştırma3-) NewQuery 🡪Select @@SPID 🡪 Execute 🡪 **Çıktı : 60**

Yani aslındaben her New Query’e bastığımda bir process başlatıyor ve database ile o anda bağlantıyı sağlıyorum ve bu bağlantıdaki bağlantı ID’mde çıktılarımdır diyor. (C# tarafında her Query bir **connection** demektir.)

**DML(Data Manipulation Language)**

Datanın kendisi ileuğraşan, datayı manipüle eden dildir.

**1-)Select:** Bizim sorgu göndermemize yarar. Bunu bir metod gibi düşünmeliyiz ve bunların Return Type (Dönüş tipleri)🡪List<Products> Liste döner. (Tablo)

**//TODO:Northwind ilişkilerini inceleyip anlat**

**Northwind**🡪 Microsoft’un geliştiriciler ve eğitimciler için hazırlamış olduğu database dir.

\*\*\*Tablo içinde kendi kendine gelen ilişki: aynı tablo içinde de bire çok ilişki kurabilirim. Tanımlamadığım veriye atama yapamam.

**SELECT**

**Kullanımı:**

**Sorgu çalıştırma :** yazılan sorguyu seçip “F5” yada “Execute” butonuna tıklamalıyız.

**Select** ifadesinden sonra **sütun adları** yazılır **From** **Tablo Adı** yazılır.

\*\*Mavi renkli kolon: içerisinde kullanılan bir Description (tablo ismi) var demektir.

[ ] parantezleri içerinde yazarsak tanımladığımız tabloya özgü olduğunu belirtmiş oluruz.

Alıştırmalar:

**1 - ) Categories** tablosunun “ **CategoryId, CategoryName,[Description],Picture** “ alanlarını getirme Sorgusu:

**Sorgu:** **Select CategoryId,CategoryName,[Description],Picture From Categories**

**2 - ) Categories** tablosunun “**CategoryName,[Description],Picture** “ alanlarını getirme Sorgusu:

**Sorgu:** **Select CategoryName,[Description],Picture From Categories**

**3-)** Herhangi bir tablodaki tüm kolonları çekmek istersek: “ **\*** ”Kullanılarak yapılır.

**Not:** “ \* ” In üzerine fare ile bir süre beklersek kolon isimlerini görebiliriz.

**Sorgu:** **Select \* From Categories**

**Önemli:**

Microsoft \* kullanımını desteklemez. Bunun çokça sebebi vardır. Biz veriyi bu şekilde çektiğimiz anda son kullanıcıya göstermek istemediğimiz kolonlarda gelmiş olur.(CategoryId vs.) Bir başka neden de \* bir join işleminde bir Subquery de view yazmamızı engeller.

\*\*Select Dönüş return🡪Lİst<Products>

**4-)** Categories tablosunda tüm kolonlardan ilk ikisini getiren sorgu: (**TOP**)

**Sorgu:** **Select top 2 \* from Categories**

**5-)** Sql de kodun bittiğini gösteren satır “ **;** ” dür.

**Select**

**top**

**2 \***

**from**

**Categories;**

**6-)** Bloklar arası geçiş yapmak istediğimizde (**GO**) 🡪 Bir sorgu bloğunun sona erdiğini ifade eder, kod yığını sona erdiğinde diğer kod bloğuna geçebilir. Aynı anda birden fazla sorgu, Prosedür vb. satırları aynı anda çalışacağı zaman kullanılmaktadır.

**go**

**Select**

**top**

**2 \***

**from**

**Categories;**

**7-) Distinct🡪** Bazen bir dorgu sonucunda tekrar eden veriler oluşabilir. Böyle durumların önüne geçmek için Distinc kullanılır.Tekrar etmeyen verileri getirir.

**Sorgu:** Çalışanların hangi şehirde yaşadığını listeleyelim(bir şehir bir daha tekrarlanmayacak):

**select Distinct City from Employees**

**8-)Where🡪** Yazdığımız sorgularda bir koşula göre veri seçmek, Görüntülemek istediğimizde kullanırız. Sorgularda Where ifadesinden sonra şart belirtilmesi gerekir.

**Kullanımı:**

**Select** ifadesinden sonra **sütun adları** yazılır **From** Tablo Adı yazılır **Where** ifadesinden sonra şart veya şartlar yazılır.

**KARŞILAŞTIRMA OPERATÖRLERİ:**

|  |  |
| --- | --- |
| **Operatör** | **Açıklama** |
| = | Eşitliğin sağındaki değer ile solundaki değer eşit ise **true** döner değilse **false** döner |
| != <> | Eşitliğin sağındaki değer ile solundaki değer eşit değil ise **true** döner eşitse **false** döner |
| > | Eşitliğin solundaki değer büyük ise sağındaki değerden **true** döner. Değilse **false** döner. |
| >= | Eşitliğin solundaki değer büyük veya eşit ise sağındaki değere **true** döner. |
| < | Eşitliğin solundaki değer küçük ise sağındaki değerden **true** döner. Değilse **false** döner. |
| <= | Eşitliğin solundaki değer küçük veya eşit ise sağındaki değere **true** döner. |

**Sorgu:** Productıd si 8 ola ürünü getir.

**select \* From Products where ProductID=8**

**Sorgu**: Productıd si 8 den büyük olan ürünleri getir.

**select \* From Products where ProductID>8**

**BETWEEN AND**

**Between** ifadesi ile iki değer arasındaki verileri sorgulayabilirsiniz. Sorguda where ifadesi belirtilen iki değer arasındaki verileri seçerken **Between-And** ile koşulu bildirebiliriz.

**Kullanımı:**

**Where** ifadesinden sonra koşulda kullanılacak sütün adı yazılır **Between** ifadesinden sonra **Değer1** And **Değer2** yazılır.

**Sorgu:** CategoryId si 1 ve 4 arasında olan ürünleri listeleyelim:

**Select \* from Products where CategoryID Between 1 And 4**

**NOT:Değer1** ve **Değer2** deki veriler seçime dahil edilir.

**AND**

**Where** ifadesi ile birlikte kullanılan **And**(ve) ifadesi, koşulların tümünün doğrulandığını kontrol etmek amaçlı kullanılır.

**Kullanımı:**

İfade1 and ifade2 and ifade3 and . . . ifaden ..

**Sorgu**: **Doğum tarihi** **1930** ile **1960** arasında olup da , **USA** de **çalışanları** listeleyen sorgu:

**Select FirstName,LastName ,BirthDate,Country from Employees where**

**Year(BirthDate) Between 1930 And 1960 AND Country='USA'**

**OR**

**Where** ifadesi ile birlikte kullanılan Or(Veya) ifadesi koşullardan herhangi birinin sağlanıp sağlanmadığını kontrol eder. Mesela 5 koşul olsa sadece birinden True dönse koşul sağlanmış olur.

**Kullanımı:** ifade1 Or ifade2 Or ifade3 Or … ifaden …

**Sorgu:** Ürünlerin birim fiyatları 18,19 veya 25 olanların listelenmesi:

**Select P.ProductName ÜrünAdı, P.UnitsInStock BirimFiyat**

**From Products P**

**Where P.UnitsInStock=18 OR P.UnitsInStock=19 OR P.UnitsInStock=25**

**IN**

Birden fazla OR ifadesini yazmak yerine Kısaca IN ifadesinden faydalanırız.

**Kullanımı: IN(**ifade1,ifade2,ifade3,… ,ifadeN**)**

**Sorgu:** Ürünlerin birim fiyatları 18 veya 19,25,97,40 veya 30 olanları listeleyin:

**select \* from Products where UnitPrice IN(18,19,25,97,40,30)**

**Not**

Olmayanları getirmek için kullanılır. (!= <> gibi düşünebilirsiniz)

**Sorgu:** Ürünlerin birim fiyatları 18 veya 19,25,97,40 veya 30 olmayanları listeleyin:

**select \* from Products where UnitPrice NOT IN(18,19,25,97,40,30)**

**ORDER BY**

Sorgu sonucu oluşan veri grubunun sıralanmasını sağlar.

**Asc** parametresi ile🡪 **Artan** Sıralama yapar.(Default)

**Desc** parametresi ile🡪 **Azalan** Sıralama yapar.

**Sorgu:** Ürünlerin birim fiyatları 18 veya 19,25,97,40 veya 30 olmayanları azalan (büyükten küçüğe) olarak listeleyin:

**select \* from Products where UnitPrice NOT IN(18,19,25,97,40,30) order by CategoryID**

**IS NULL**

**Where** ile birlikte kullanıldığında **verinin boş olma durumunu** kontrol eder.

**Sorgu:** Employess de null olanları getiren sorgu:

**select \* from Employees where Region is null**

**Sorgu:** Müşteriler tablosunda bölge bilgisi girilen kayıtları listeleme sorgusu:

**select C.CompanyName,C.Region from Customers C where C.Region is not null**

**SPACE**

Boşluk karakterini ifade eder. Parametresine yazılan adet kadar boşluk karakteri oluşturur.

**Sorgu:** Personelin ad ve soyadını birleştirirken space fonksyonu kullanarak sorgu yazımı:

**Select E.FirstName+ SPACE(1)+E.LastName 'Ad-Soyad' from Employees E**

**NULL**

Hiçbir şey anlamına gelir**. “Tabloda bir sütün da hiçbir değer olmaması demektir**. “

Not: Tablodaki bir boşluk karakteri **Null** demek değildir.

**ISNULL**

**ISNULL** fonksiyonu iki parametre alır.

**Parametre 1**🡪hangi alanın null olması durumunun kontrol edileceğini sorar.

**Parametre 2**🡪alanın null olması durumunda aktarılacak değeri ifade eder.

**Sorgu:** Bölge verisi boş ise bölge yok yazan sorgu:

**select C.CompanyName,C.Region , ISNULL(C.Region,'BolgeYok')from Customers C**

**LIKE**

**String** değerler üzerinde arama yapmamızı sağlar. Bu operatör **benzerlik** bulma amacı ile kullanılır. Ancak özel karakterler yardımı ile kullanılması gerekmektedir.

**ÖZEL KARAKTERLER:**

|  |  |
| --- | --- |
| **Operatör** | **Açıklama** |
| \_ | Bir harf veya rakam karakterini ifade eder. |
| [A-Z] | A ile Z arasındaki harfleri ifade eder. |
| [0-9] | 0 ile 9 arası rakamları ifade eder. |
| [] | Herhangi bir harf veya rakam yerine gelebilecek harf veya rakamı ifade eder. |
| [^] | Herhangi bir harf veya rakam yerine gelemeyecek harf veya rakamı ifade eder. |
| % | Birden fazla harf veya rakamı ifade eder. |
| + | String ifadeleri birleştirir. Sayısal ifadeleri toplar. |

**Sorgu**: Ürün ismi ‘Chai’ ile başlayan tüm kayıtları getir.

🡪 **select \* from Products where ProductName = 'Chai'** **like Bu şekilde kull.da**

**🡪 select \* from Products where ProductName like 'Chai'** **eşittirden farkı**

**Yoktur.**

**Sorgu**: Ürün ismi ‘C ile başlayan tüm kayıtları getir.

🡪 **select \* from Products where ProductName like 'C'** 🡪 sorgu çalışmaz eşitttir

ile aynı işlemi yapar.

🡪 **select \* from Products where ProductName like 'C%'**🡪 başharfi c olsun / sonra ne

gelirse gelsin % anlamını verir.

**Sorgu**: Ürün isiminde ikinci harfi h olan ürünleri listeleme sorgusu:

**select P.ProductName from Products P where ProductName like '\_h%'**

**Sorgu:** Ürün isminde ilk harfi a ve ya c olan ürünleri listeleyen sorgu:

**select P.ProductName from Products P where ProductName like '[A,C]%'**

**Sorgu:** Ürün isminde ilk harfi a ve ya c ikinci harfi h olan ürünleri listeleyen sorgu:

**select P.ProductName from Products P where ProductName like '[A,C]H%'**

**Sorgu:** Ürün isminde ilk harfi a ve ya c ikinci harfi h olan üçüncü harfi ne olursa olsun dördüncü harfi f olan ürünleri listeleyen sorgu:

**select P.ProductName from Products P where ProductName like '[A,C]H\_F%'**

**Sorgu:** Ürün isminde ilk harfi a ve ya c olan son harfi n veye g olan ürünleri listeleyen sorgu:

**select P.ProductName from Products P where ProductName like '[A,C]%[n,g]'**

**Sorgu:** Ürün isminde ‘o’ geçen ürünleri listeleyen sorgu:

**select \* from Products where ProductName like '%o%'**

**Sorgu:** Ürün isminde ‘of’ geçen ürünleri listeleyen sorgu:

**select \* from Products where ProductName like '%of%'**

**Sorgu:** Shippers tablosunun CompanyName🡪Firma / Phone🡪telefon kolonlarınının isimleri olarak değiştiren sorgu:

Kullanım 1:

**select S.CompanyName FirmaAdı,S.Phone Telefon from Shippers S**

kullanım 2:

**select S.CompanyName 'Firma Adı',S.Phone 'Telefon' from Shippers S**

kullanım 3:

**select S.CompanyName as FirmaAdı,S.Phone as Telefon from Shippers S**

**Not**: **Like** ı büyük verileri aramada neden kullanmayız? Çünkü like ile gönderdiğimiz sorguların hiçbiri indexleme ile çalışmaz.

**AGGREGATE FONKSİYONLAR**

Kümeleme fonksiyonlarıdır. Birden fazla satırda işlem yapma yeteneğine sahiptirler. **NOT:** Tek başlarına kullanıldıklarında bir satır bir sütunluk veri döndürürler. **(Tek değer dödürür-yanlarına ikinci bir kolon kullanamazsın : bunun için group by kullanmalısın)**

\*\*SADECE SAYISAL VERİLER İLE ÇALIŞIRLAR.

\*\*AGGREGATE fonksiyonlar kullanarak oluşan sonuçları içeren veriyi filtreleme yaparken WHERE kullanılmalıdır.

\*\* AGGREGATE fonksiyonlar kullanarak oluşan sonuçları sonuçları içeren veriyi filtreleme yaparken WHERE ifadesi yerine Having ifadesi de kullanılabilir.

\*\*Sütunlara verilen takma (alias) isimler HAVİNG ile kullanılamazlar.

\*\*Eğer sorguya kümeleme yapıcak sütundan başka bir sütun daha eklenirse ,eklenen bu sütuna göre sorguyu gruplamak gerekir.

\*\*Gruplama işlemine eklenen her sütunun dahil edilmesi gerekir. Group by ifadesinde aggregate fonksiyon kullanılan sütundan başka, her sütun eklenmelidir.

|  |  |
| --- | --- |
| **Fonksiyon** | **Açıklama** |
| Max | Kendisine sunulan değerlerden en büyük olan değeri bulur. |
| Min | Kendisine sunulan değerlerden en küçük olan değeri bulur. |
| Count | Sorgu sonucunda dönen satır sayısını döndürür. **Not**: NULL olanları getirmez |
| Sum | Parametresinde sunulan sütuna ait toplamı döndürür.(o kolonda ne varsa toplayıp getirir.) |
| Avg | Ortalama işlevidir. Parametresinde sunulan sayısal değerlerin ortalamasını bulur. |

**Örnek sorgular:**

1. ürünler tablosunda fiyatı en yüksek değeri getiren sorgu:

**select Max(P.UnitPrice) from Products P**

1. ürünler tablosunda fiyatı en düşük değeri getiren sorgu:

**select Min(P.UnitPrice) from Products P**

1. çalışanlardan yaşça en büyük ve en küçük olanlarını bulan sorgu:

**select MAX(DATEDIFF(YEAR,E.BirthDate,GETDATE())) 'Büyük Yaş',**

**Min(DATEDIFF(YEAR,E.BirthDate,GETDATE())) 'Küçük Yaş' from Employees E**

1. ürünler tablosunda productname in olduğu satır sayısını bulan sorgu:

(kolon ismi yazınca null olanlar hariç getirir.)

**select Count(ProductName) from Products**

**5-** ürünler tablosundaki toplam sayıyı getiren sorgu(satır sayılarının count’ unu getirir.)

**Select COUNT(\*) from Products**

**6-** Order Details adet\*Birim fiyatının toplanması

**select Sum(OD.UnitPrice \*OD.Quantity) 'Sipariş Toplamı' from [Order Details] OD**

**Açıklama**: Birim fiyat ile adet değerini çarp ve çıkan sonuçların tümünü topla

**7-** elimdeki tüm ürünlerin fiyatının ortalamasını bulan sorgu:

**select Avg(UnitPrice) from Products P**

**8-** çalışanların yaş ortalamasını bulan sorgu:

**select AVG(DATEDIFF(year,BirthDate,GETDATE())) from Employees**

**GROUP BY**

Aggregate fonksiyonların yanına bir kolon daha yazacaksanız bunu group by ile yapmanız gerekir. Gruplandırma işlemi yapar.Gruplandırma yaparken gruplandıracağın şeyi de bana göstermen gereklidir.

Sorgu: **select CategoryID ,COUNT(CategoryID) from Products group By CategoryID**

Karşısında ne olsunNeye göre grupla :CategoryId ye göre

**Örnek** : categotId NoColumsName

1 12 🡪 **categoryId si 1 olan 12 tane kolon varmış** !!!

**Sorgu**: çalışanlar tablosunda kaç tane mr,ms , mrs,dr var?

**SELECT E.TitleOfCourtesy , COUNT(EmployeeID) FROM Employees E GROUP BY TitleOfCourtesy**

**SUB QUERY(iç içe yazılan sorgular)**

Bazı durumlarda bir sorgunun, diğer bir sorgu sonucuna göre çalışması gerekebilir. Bu durumda SubQuery kullanılır. **(Select ile from arasına yazılan SubQuery) Önemliiii**

**\*\*\*ÖRNEK SORGULAR:**

**Sorgu 1:** ürünler tablosunda örneğin 'chai' adlı ürünün tedarikçi adı ve kategori adını listeleyen sorgu:

**select P.ProductName AS 'ÜRÜN ADI',**

**(select S.CompanyName from Suppliers S where S.SupplierID=P.SupplierID) AS 'TEDARİKÇİ',**

**(SELECT C.CategoryName FROM Categories C WHERE C.CategoryID=P.CategoryID)AS 'KATAGORİ',**

**P.QuantityPerUnit AS 'BİRİM ADET',**

**P.UnitPrice AS 'FİYAT'**

**from Products P**

**Sorgu 2:** kategori adı Beverages olan ürünlerin satış detay listesini gösteren sorgu:

**select \* from [Order Details]**

**where ProductID in(select ProductId from Products**

**where CategoryID in(select CategoryID from Categories where CategoryName='Beverages'))**

**Sorgu 3:** Nancy nin hangi kategoriden kaç tane ürün sattığını listeleyen sorgu:

**select (SELECT CategoryName FROM Categories where Categories.CategoryID=Products.CategoryID ) as 'KATAGORİ',COUNT(CategoryID)from Products**

**where ProductID in(select ProductID from [Order Details]**

**where OrderId in (select OrderId from Orders where EmployeeID=**

**(select EmployeeID from Employees where FirstName='Nancy')))**

**group by CategoryID**

**Sorgu 4:** Nancy nin sattığı ürünleri listeleyen sorgu:

**select \* from Products**

**where ProductID in(select ProductID from [Order Details]**

**where OrderId in (select OrderId from Orders**

**where EmployeeID=(select EmployeeID from Employees where FirstName='Nancy')))**

**Sorgu 4:** Alfreds Futterkiste aldığı ürünleri listeleyen sorgu:

**select \* from Products**

**where ProductID in(select ProductID from [Order Details]**

**WHERE OrderID in (select OrderID from Orders**

**where CustomerID=(select CustomerID from Customers where CompanyName='Alfreds Futterkiste')))**

**Dönüş tipleri:**

**Select \* From Categories 🡪List<Products>🡪Tablo döner**

**Select top 1 \* from Categories 🡪 Product (tek bir product döner)**