

Laboratuvar 1

SQL SELECT CÜMLELERİ

Oracle birçok kullanışlı ve güçlü özellikleri olan bir veritabanıdır. Bu özelliklerinin birçoğu SQL ile ilgilidir. VTYS'lerinin çoğunluğunda veriler ile çalışmak için SQL kullanılmaktadır.

SQL, Oracle veritabanından verileri çekmek ya da veriler üzerinde değişiklik yapmak için kullanıla basit bir dildir. Diğer programlama dilleri gibi döngüleri vs. yoktur.

SQL SELECT cümlecikleri ise veritabanından verileri **sorgulamak** (*query*), çekmek için kullanılmaktadır.

SQL SELECT Cümleleri

Bir veritabanının verileri depolamak için kullandığı temel yapı taşı tablolardır. Bir tablo, sütunlar ve özelliklerinden meydana gelmektedir. Veriler bu tablolarda satır şeklinde tutulmaktadır.

SQL, cümlecikleri, belirli kuralları olan basit bir dildir. Burada bahsedilen basitlik kavramını yanlış anlamamak gerekir. SQL çok kolay öğrenilebilmekle birlikte tüm veritabanı işlemleri için yeterli ve güçlü bir dildir.

SQL SELECT Cümlelerinin Yetenekleri

SQL SELECT cümlecikleri veritabanından veriyi çekmek için kullanılır. **SELECT** cümlecığı kullanarak aşağıdaki işlemleri yapabiliriz:

- **Projeksiyon:** SQL'in bu yeteneği ile bir tablodaki istediğiniz sütunları seçip sadece o sütunlardaki verileri alabilirsiniz.
- **Seçme:** SQL'in bu yeteneğini kullanarak tablolardan istediğini veri satırlarını alabilirsiniz. Satırları seçmek için çeşitli kriterler kullanabilirsiniz.
- **Birleştirme:** SQL'in bu yeteneğini kullanarak farklı tablolardaki verileri birleştirerek alabilirsiniz.

Basit SQL Sorguları

Veritabanından verilerin **SELECT** cümleleri ile alınması işlemine **sorgulama** denir.

Temel SQL SELECT Cümlesi

SELECT cümlecığı SQL yazılırken en çok kullanılan ifadedir. Basit bir **SELECT** cümlecığı aşağıdaki bölümlerden oluşur:

- **SELECT** Veri çekilecek sütunları belirler.
- **FROM** Verilerin yer aldığı tabloyu belirler.

```
SELECT * | { [DISTINCT] sütun | deyim ya da işlem [adı],... }  
FROM tablo;
```

SELECT cümlesi yazarken eğer tabloya ait tüm sütunları seçecekseniz sütun isimlerini tek tek yazmak yerine ya da sütun isimlerini bilmiyorsanız "*" karakterini kullanabilirsiniz. "*" karakteri tüm sütunlar anlamına gelir.

Sütun Takma Adları (ALIAS)

Sütun takma adları tablolarda yer alan sütun isimlerini daha anlaşılır şekilde görüntülemek için kullanılır.

Bir sütuna takma ad vermek için sütun isminden sonra <boşluk> **AS** <boşluk> <takma ad> ya da <boşluk> <takma ad> yazılır. Sütun takma adları sorgu çalıştırıldığı zaman otomatikman büyük harfe çevrilir. Eğer takma adların yazıldığı şekliyle kalması isteniyorsa " " işareti arasına yazılmalıdır.

Örnek:

```
SELECT ogrenci_adi AS ADI  
FROM ogrenci;
```

Veya

```
SELECT ogrenci_adi ADI  
FROM ogrenci;
```

Tekilliyi Saęlamak (DISTINCT)

Bazı zamanlar sütünlarda yer alan veriler tekrar edebilir. Örneęin bir bölümünde çalışanların hepsinin bölüm bilgisi aynı olacaktır. Bu durumlarda tekilliyi saęlamak için DISTINCT (veya UNIQUE) anahtar kelimeleri kullanılır.

Örnek:

```
SELECT distinct bolumNo  
FROM ogrenci;
```

Veya

```
SELECT unique bolumNo  
FROM ogrenci;
```

Çekilen Kayıtları Sınırlandırma (WHERE)

Veritabanından kayıtları sorgularken her zaman bütün kayıtlara ihtiyacımız olmaz. Bu durumda sadece bizim için gerekli olan kayıtları çekmek için sorgulamada sınırlama yapmamız gerekmektedir.

Çekilen kayıtları sınırlandırmak için SELECT cümleciklerinde WHERE sözcüğünü kullanırız. Bu durumda WHERE sözcüğünde belirtilen koşula uyan kayıtlar gelir.

WHERE koşulu yazarken dikkat edilmesi gereken nokta şudur. WHERE koşulunda sütün isimleri ya da ifadeler kullanabilirsiniz ancak sütün takma adlarını kullanamazsınız. Sütün isimlerini WHERE koşulunda kullanabilmek için SELECT bölümünde sütünun yer almasına gerek yoktur.

Oracle'da çekilen kayıtları sınırlandırmak için birçok operatör mevcuttur.

Aşagıda sınırlandırma sorgusunun genel yapısı görülmektedir.

```
SELECT * | { [DISTINCT] sütün | deyim ya da işlem [adi],... }  
FROM tablo;  
[WHERE koşul(lar)];
```

Karşılaştırma Operatörleri

Karşılaştırma operatörleri iki değeri ya da ifadeyi karşılaştırmak için kullanılır. Karşılaştırma işlemi sonucu TRUE, FALSE veya NULL olur.

Aşagıdaki tabloda karşılaştırma operatörlerinin bir listesi verilmiştir.

Karşılaştırma Operatörleri	
Operatör	Amacı
=	Eşitlięi test eder.
!= , <> , ^=	Eşitsizlięi test eder. <> operatörü her platformda çalışır.
<	Küçük operatörü. Deęerin ya da ifadenin küçük olup olmadığını test eder. Küçük ise TRUE sonucunu verir.
>	Büyük operatörü. Deęerin ya da ifadenin büyük olup olmadığını test eder. Büyük ise TRUE sonucunu verir.
<=	Küçükeşit operatörü. Deęerin ya da ifadenin küçük ya da eşit olması durumunda

	TRUE sonucunu verir.
>=	Büyükeşit operatörü. Değerin ya da ifadenin büyük ya da eşit olması durumunda TRUE sonucunu verir.
SOME	Bir listedeki ya da alt sorgudaki değerleri karşılaştırmak için kullanılır. SOME operatöründen önce mutlaka =, !=, <, >, <= ya da >= operatörleri kullanılmalıdır.
ANY	Bir listedeki ya da alt sorgudaki değerleri karşılaştırmak için kullanılır. ANY operatöründen önce mutlaka =, !=, <, >, <= ya da >= operatörleri kullanılmalıdır.
ALL	Bir listedeki ya da alt sorgudaki tüm değerleri karşılaştırmak için kullanılır. ALL operatöründen önce mutlaka =, !=, <, >, <= ya da >= operatörleri kullanılmalıdır.

Mantıksal Operatörler

Mantıksal operatörler iki karşılaştırma işleminin sonucunu belirlemek için kullanılır. Aşağıdaki tabloda mantıksal operatörlerinin bir listesi verilmiştir.

Mantıksal Operatörler	
Operatör	Amacı
NOT	Sonuç değerini tersine çevirir. Sonuç TRUE ise FALSE, FALSE ise TRUE yapar. NULL değerlerde NULL sonucunu verir.
AND	Her iki taraf da TRUE ise TRUE, diğer durumlarda TRUE değerini verir. NULL değerler NULL sonucunu verir.
OR	Her iki taraf da FALSE ise FALSE, diğer durumlarda FALSE değerini verir. NULL değerler NULL sonucunu verir.

Diğer Operatörler

Aşağıdaki tabloda diğer operatörlerinin bir listesi verilmiştir.

Diğer Operatörler	
Operatör	Amacı
IN ({liste}, NOT IN ({liste}))	Listede yer alan değerlerin olup olmadığını test eder.
BETWEEN ... AND...	İki değer arasındaki değerleri test eder. Yani {sütun}>={değer1} AND {sütun}<={değer2} anlamına gelir.
EXISTS	Altsorgular ile birlikte kullanılır. Altsorgu en az bir kayıt dönerse TRUE değeri döner.
IS NULL, IS NOT NULL	NULL değerleri test eder.
LIKE	Belirli karakter kalıbına göre karşılaştırma yapar.

IN ve NOT IN

IN ve NOT IN operatörleri bir liste içerisinde ilgili kaydın olup olmadığını test eder. İlgili kayıt listede var ise IN operatörü TRUE değerini verir. İlgili kayıt yok ise NOT IN operatörü TRUE değerini verir. Daha önce öğrendiklerimiz ile açıklamak gerekirse =ANY operatörü ile ve NOT IN operatörü !=ALL operatörü ile aynı işlevi görür.

BETWEEN

BETWEEN operatörü aralık testi yapar.Örneğin A BETWEEN B AND C demek $A \geq B$ AND $A \leq C$ anlamına gelir. B,C değerlerinin karşılaştırma aralığına dahil olduğuna dikkat ediniz.

NULL Değerler

NULL değerleri geçersiz, hükümsüz, değersiz, önemsiz veya yok şeklinde tanımlayabiliriz. Boş bir karakter '' ya da 0 (sıfır) NULL değildir.

IS NULL ve IS NOT NULL

Bu operatörler NULL veya NOT NULL değerleri bulabilmek için kullanılır. = ya da != operatörleri NULL değerlerle çalışmaz. Karşılaştırılan değer NULL ise IS NULL operatörü TRUE döner. Eğer karşılaştırılan değer NULL değil ise NOT NULL operatörü TRUE döner.

LIKE

Bu operatör karakter alıpları ile çalışır. Oracle'da % ve _ karakterleri karakter kalıbı oluşturmak için kullanılır. _ tek karakter anlamına gelir, % ise her ne olursa anlamına gelir. Konuyu daha iyi anlayabilmek için aşağıdaki tabloyu inceleyiniz.

% ve _ Karakterleri ile Oluşturulan Kalıp Örnekleri	
Kalıp	Anlamı
%MET	Başlangıcı önemli değil ancak sonu mutlaka MET olanlar listelenir.
%MET%	İçinde ME geçenler listelenir.
ME%	Başı ME olan sonu önemli olmayanlar listelenir.
_HMET	İlk karakteri önemli olmayan ancak sonu HMET olanlar listelenir.
%HME_	Başı önemli olmayan, sondan 4., 3, ve 2, karakteri H,M,E olan ve sonunda bir karakter olanlar listelenir.

Kayıtların Sıralanması (ORDER BY)

Önceki bölümlerde öğrendiğimiz SQL sorgularında Oracle çekilen kayıtları sıralanmış bir şekilde getirilmesini garanti etmez. SELECT ile çektiğimiz kayıtların istediğimiz şekilde sıralı gelmesi için ORDER BY kelimesi kullanılır.

ORDER BY yazdığımız SELECT cümlelerinde en son kullanılan ifade olmalıdır. ORDER BY varsayılan olarak artan sırada kayıtları sıralar. Azalan şekilde sıralama istediğimiz zaman ORDER BY ile birlikte DESC kelimesi kullanılır.

Aşağıda sıralama sorgusunun genel yapısı görülmektedir.

```
SELECT * | { [DISTINCT] sütun | deyim ya da işlem [adı],... }  
FROM tablo;  
[WHERE koşul(lar)]  
[ORDER BY {sütun, ifade} [ASC|DESC] ]
```

ORDER BY: Çekilen kayıtların sıralanacağını belirtir.

ASC: Kayıtları artan şekilde sıralar (Varsayılan sıralama).

DESC: Kayıtları azalan şekilde sıralar.

NOT: SELECT ifadesinde kullanılmayan sütun isimlerini de ORDER BY da kullanabilirsiniz. Ancak SELECT ifadesinde DISTINCT kelimesi yer alırsa sadece SELECT ifadesinde yer alan sütunları ORDER BY da

kullanabilirsiniz. Eğer DISTINCT yer alan ifade de sütunlar ile birlikte çeşitli operatörler kullanılırsa bu operatörlerde ORDER BY da yer almalıdır. Ayrıca ORDER BY da en fazla 255 sütun ya da ifade kullanabilirsiniz.

Örnek:

```
SELECT DISTINCT 'Maaş' || ücret
FROM personel
WHERE bolumid=6
ORDER BY 'Maaş' || ücret;
```

NULL Değerlerin Sıralanması

Sıralamalarda Oracle'ın varsayılan değeri yani artan sıralama kullanıldığı zaman NULL değerler en altta yer alır. DESC, azalan sıralama yaptığımızda en üstte yer alır. Bu varsayılan sıralama NULLS FIRST ya da NULLS LAST kullanılarak değiştirilir.

NULLS FIRST: İlk önce NULL değerler gelir.

NULLS LAST: En son NULL değerler gelir.

İfadelerin Kullanımı (Expressions)

Bir veya daha fazla değer, operatörün ve SQL fonksiyonunun birlikte oluşturduğu değere **ifade** denir. Örneğin $2+5*3$ ifadesi 3 tane sayı ve 2 tane aritmetik operatörden oluşmuştur. Sonuç ise 17'dir. İfadelerin sonuç değerlerin veritipi genelde ifadeyi oluşturan elemanların değer tipidir. Bahsedilen ifade de sonuç değerinin veritipi NUMBER olur.

İfadeler bir SQL içerisinde çeşitli bölümlerde olabilir. Örneğin:

- SELECT cümleğinde
- WHERE, ORDER BY ya da HAVING cümleğinde
- INSERT cümleğini VALUES kısmında
- UPDATE cümlesinin SET kısmında

DUAL Tablosu

DUAL tablosu sadece bir sütunu ve bir kaydı olan bir tablodur. Sistem değişkenlerini veya ifadelerini hesaplamak için kullanılır. Veritabanındaki tüm kullanıcılara açıktır.

Bilinmesi Gereken Bazı SQL*Plus Komutları

SQL*Plus veritabanı ile iletişime geçmek için kullanılan Oracle tarafından geliştirilmiş bir araçtır. SQL*Plus'ı kullanarak her çeşit SQL cümleğini ve PL/SQL programlarını çalıştırabilir, sorgu sonuçlarını formatlayabilir veya veritabanını yönetebilirsiniz. Makinenize Oracle yazılımını kurduğunuz zaman, SQL*Plus da kurulmuş olmaktadır.

DESCRIBE Komutu

DESCRIBE (DESC) komutu veritabanı nesnelere hakkında bilgi almak için kullanılır. DESCRIBE komutu kullanılarak tablo ya da görüntülerin sütunları, veri tipleri vs. hakkında bilgi alınabilir. Yine bu komutu kullanarak prosedürlerin ya da fonksiyonların parametreleri öğrenilebilir.

LIST Komutu

LIST komutu SQL ortamında yer alan içeriği görüntülemek için kullanılır. Bu komut (L) şeklinde kısaltılabilir. SQL cümlesinde yer alan * işareti o anki aktif satırı belirtir.

LIST LAST son satırı listeler. LIST m n şeklindeki kullanım ise m'inci satırdan n'inci satıra kadar olan kısmı görüntüler.

APPEND Komutu

APPEND komutu yanına yazılan satırı ortamda yer alan SQL'in sonuna ekler. Kısaltılmış hali A'dır.

CHANGE Komutu

CHANGE komutu yanına yazılan eski değeri yeni değer ile değiştirir. Kullanımı CHANGE /eski/yeni şeklindedir. Eğere yeni değeri yazmazsanız eski değer silinir. Kısaltılmışı C'dir.

INPUT Komutu

INPUT komutu yanına yazılan satırı ya da devam eden satırları yeni bir satır olarak ortamdaki SQL'e ekler.

DEL Komutu

DEL komutu yalnız başına kullanıldığı zaman aktif satırı siler. Aktif satırın * işareti ile belirtildiğini hatırlayınız. DEL m n şeklide kullanıldığı zaman m'inci satırdan n'inci satıra kadar olanları siler. Eğer m ya da n yerine * kullanırsak aktif satırı belirtmiş olursunuzç DEL LAST son satırı siler.

SORGU SONUÇLARININ DOSYAYA KAYDEDİLMESİ

Zaman zaman SQL*Plus'da çalıştırdığımız sorgu sonuçlarını bir dosyaya kaydetmemiz gerekir. Bunun için SQL*Plus'da SPOOL komutu kullanılır. Kullanımı SPOOL *dosya_adi* şeklindedir. Soday uzantısı verilmezse otomatik olarak **.lst** uzantısı verilir.

Sorgu çalıştırıldıktan sonra dosyaya yazma işlemini sonlandırmak için SPOOL OFF ya da SPOOL OUT komutları kullanılır. SPOOL OFF dosyaya yazma işlemini sonlandırır. SPOOL OUT ise dosyaya yazma işlemini sonlandırıp, dosyayı yazıcıya gönderir.