TEMEL SQL SORGULARI BİRİNCİ KISIM

SQL KOMUTLARININ İCRA SIRASI

Keywordslerin yazılış sırasından farklı olarak SQL'in syntaxtaki sıralaması şu şekilde olmalıdır:

"SELECT clause FROM clause"

SELECT, FROM ve * Kullanımı

a) Employee tablosuna ait yalnızca belirttiğim sütunları çağırıyorum:

```
SELECT employee_id, first_name, last_name
FROM employees;
```

b) Employee tablosuna ait tüm sütunları çağırıyorum.

```
SELECT * FROM employees
```

WHERE VE Operatörlerin Kullanımı

a) Belirttiğim sütunları yalnızca employee_id'si 101 olanlardan getirecek.

```
SELECT employee_id, first_name, last_name
FROM employees
WHERE employee_id=101;
```

b) Belirttiğim sütunları yalnızca employee_id'si 1'den büyük veya eşit olanlardan getirecek.

```
SELECT employee_id, first_name, last_name
FROM employees
WHERE employee_id >= 1;
```

c) Burada Taylor kısmında büyük küçük harf uyumu dikkate alınmalı. Çünkü burada belirteceğimiz bilgi doğrudan veri tabanından alınıp getirilir. Bu harf duyarlılıklarını, LOWER ve INITCAP ile ortadan kaldırabiliriz.

```
SELECT first_name, last_name
FROM employees
WHERE last_name='Taylor';
```

KARŞILAŞTIRMA OPERATÖRLERİ

-BETWEEN

-IN

-LIKE

-NULL

-BETWEEN...AND

a) BETWEEN kullanımı:

```
SELECT first_name, salary
FROM employees
WHERE salary BETWEEN 9000 AND 11000;
```

```
SELECT first_name, salary
FROM employees
WHERE salary >= 9000 AND salary <= 11000;
```

NOT: Bu iki cümle de aynı çıktıyı verir.

b) LIKE kullanımı

Veritabanındaki kayıtlı bilgiyi çekip kullanabilmek için verilerin hepsini ezbere bilmemize gerek yok. Birebir eşleşeni bulmak zor olabilir. LIKE ile benzerlikleri sorgulayarak veriye daha kolay ulaşabiliriz. (% ve _)

```
SELECT last_name
FROM employees
WHERE last_name LIKE '_o%'
```

NOT: Verimde o harfi olsun ama 2.harf olsun. %'den sonra kaç harf olduğu beni ilgilendirmez.

```
SELECT last_name FROM employees
WHERE last_name LIKE '_O%'
```

NOT: Büyük küçük harf uyumu burada da söz konusu.

```
SELECT last_name
FROM employees
WHERE last_name LIKE '%A%'
```

Peki ya % veya _ işaretleri benim asıl verimin içinde olursa ne yapacağım?

```
SELECT last_name
FROM employees
WHERE last_name LIKE '%\_R%'  ESCAPE '\'
```

NOT: Bu cümle ile ORACLE EX'e bu cümleyi kurmuş oluyorum: "R ve öncesinde _ olanın veri olduğunu anla ve bu işareti harf sayısı belirtmişim gibi algılama."

NULL

Her erişmeye çalıştığımız veri bir değere karşılık gelmeyebilir. Bu durumlarda o veri NULL değere sahip olmuş olur ve veri tabanı tablosunda değerine karşılık "-" işareti ile belirtilir. Dilersek bu "-" işaretinin yerine kendi belirttiğimiz bir cümleyi veya değeri NVL komutu yazabiliriz. Daha sonraları bu konuya değineceğim.

Eşittir yerine "IS NULL" kullanırız.

Eşit değil yerine "IS NOT NULL" kullanırız.

```
SELECT last_name, manager_id
FROM employees
WHERE manager_id IS NULL;
```

NOT: manager_id değeri NULL olanların last_name ve manager_id'sini getirir.

```
SELECT last_name, commission_pct
FROM employees
WHERE commission_pct IS NOT NULL;
```

NOT: commission_pct değeri NULL olmayanların last_name ve commission_pct'sini getirir.

MANTIKSAL KARŞILAŞTIRMALAR VE ÖNCELİK KURALLARI

-NOT

-IN

-NOT IN

-AND

-OR

a) OR kullanımı

```
SELECT city, state_province, country_id
FROM locations
WHERE country_id IN('UK','CA');
```

```
WHERE country_id = 'UK' OR country_id = 'CA';
```

NOT: Bu iki cümle de aynı çıktıyı verir.

```
SELECT first_name, department_id, salary
FROM employees
WHERE department_id> 50 AND salary > 12000;
```

```
SELECT first_name, department_id, salary
FROM employees
WHERE department_id > 50 OR salary > 12000;
```

NOT: OR komutunda tek tarafı doğrulamak yeterli olacaktır.

```
SELECT department_name, dept_manager_id, location_id
FROM departments
WHERE location_id=2500 OR dept_manager-id=124;
```

b) NOT VE IN kullanımı

SELECT department_name, location_id

FROM departments

WHERE location_id NOT IN (1700,1800);

NOT: location_id'leri 1700 ve 1800 olmayanların belirttiğim satırlarını getirir.

SELECT department_name, location_id

FROM departments

WHERE location_id IN (1700,1800);

NOT: location_id'leri 1700 ve 1800 olanların belirttiğim satırlarını getirir.

c) AND kullanımı

SELECT last_name, department_id, first_name, last_name FROM employees

WHERE department_id IN(50,80) AND first_name LIKE 'C%'

OR last_name LIKE '%m%';

NOT : AND kısmında iki tarafı da birlikte okur, eğer koşul sağlanmıyorsa OR kısmı çalışır.

ORDER BY

Verileri veri tabanımızdan çekip çağırırken SQL 'e bu işlemi belirli bir kısıta göre sıralama yaparak getirmesini ORDER BY komutu ile söyleyebiliriz.

Çıktı üzerinde çalıştığı için ORDER BY cümlenin sonuna yazılır.

Aynı zamanda SELECT edilmemiş sütunlara da ORDER_BY komutunu verebiliriz.

SELECT last_name, hire_date

FROM employees

ORDER BY hire_date;

NOT: Burada kısıtı hire_date olarak belirledik ve satırdaki verileri hire_date'e göre sıralama yaparak çağırdı

Aynı şekilde iç içe bir sıralama yapmak da mümkün;

```
SELECT last_name, hire_date
FROM employees
ORDER BY hire_date, last_name;
```

NOT: Önce hire_date göre, onun içinde de last_name e göre sıralama yapar.

ORDEY BY kullanırken varsayılanın dışında ters sıralama yapmamız da DESC komutu ile mümkün;

```
SELECT last_name, hire_date AS “işegiriş”
FROM employees
ORDER BY işegiriş desc, last_name;
```

NOT:

ASCENDİNG: Küçükten büyüge, default.

DESCENDİNG: Büyükten küçüğe, ayrıca belirtilmeli.

İNTRODUCTİON of FUNCTİONS

Sinlge-Row Functionlar: Girdisi bir tane ise çıktısı da bir tane olması gerekir. Yani her sütun için yalnızca bir çıktı elde ederiz. (Money --> Drink Machine --> Drink)

Multi-Row Functionlar: Girdi birden fazladır ama çıktı yalnızca bir tanedir.

CONCATENATİON

Sütunları birleştirme işlemlerini bu komut ile gerçekleştiririz.

string1 || string2 || string_n

```
SELECT department_id ||' '|| department-name as DEPARTMENT
FROM departments;
```

```
SELECT last_name || ' Has a monthly salary of '|| salary*2 || ' dollars .' AS PAY
 FROM employees;
```

DISTINCT

Tüm departmanlarda çalışan olmak zorunda değil. Departmanlardan birinde hiçbir çalışan olmayabilir. Çıktıdaki tekrar eden verileri bu komut ile ortadan kaldırırız. Yani çıktıdaki tekrarları önleriz.

```
SELECT DISTINCT department_id
FROM employees;
```

KISA BİR ÖZET

1)FROM ile alanı belirle.

2)WHERE ile alanın belli bir kısmını belirtirerek alanı daralt.

3)SELECT ile o alanda arama yap.

4)ORDER BY ile de bunu bana getirirken sıralayarak getir dedik.

5) İNTRODUCTİON of FUNCTİONS