

Temel SQL Sorguları İkinci Kısım

DUAL

Veri tabanında herhangi bir tablo üzerinde çalışmadığımız zaman, yazdığımız SQL cümlelerini FROM DUAL şeklinde çağırarak, istediğimiz veriyi elde edip görebiliriz.

```
SELECT SYSDATE FROM DUAL;
```

```
SELECT (319/29) + 12 FROM DUAL;
```

UPPER, LOWER ve INITCAP

Daha önce de bahsettiğim üzere veri tabanındaki veriyi çağırırken büyük küçük harf duyarlılığını bu komutlar ile kaldırabiliyoruz.

```
SELECT last_name
```

```
FROM employees
```

```
WHERE LOWER(last_name) = 'abel';
```

NOT: “abel” verisi normalde veri tabanımızda “Abel” olarak kayıtlı fakat ben burada LOWER komutu ile veriyi küçük harflerle çağırarak bu duyarlılığı kaldırmış oldum.

```
SELECT UPPER(last_name) from employees;
```

CHARACTER MANIPULATION FUNCTIONS:

-SUBSTR: Verinin yalnızca belirttiğimiz kısmını çağırmanızı sağlar.

-LENGTH: Verinin karakter boyutunu verir.

-INSTR: Belirttiğimiz karakterin veri üzerinde kaçınıcı sırada olduğunu gösterir.

-LPAD | RPAD: Veriyi solundan veya sağından belirttiğimiz karakter ile doldurur.

-TRIM: Verideki belirttiğimiz karakteri sondan veya baştan kırpmanızı sağlar.

-REPLACE: Verideki belirttiğimiz karakteri kaldırıp tercihen de yerine başka bir değer koymanızı sağlar.

SUBSTR

```
SELECT SUBSTR(last_name, 1, 3)
FROM employees;
```

NOT: Birinci karakterden başlayarak last_name verilerinin ilk üç harflerini getirir.

```
SELECT SUBSTR('HelloWorlddddddd', 6)
FROM DUAL;
```

NOT: Altıncı karakterden başlayarak verinin kalan tüm kısmını getirir.

LENGTH

```
SELECT employee_id, LENGTH(last_name) AS "Boy"
FROM employees ORDER BY Boy;
```

INSTR

```
SELECT last_name, INSTR(last_name, 'a')
FROM employees;
```

NOT: last_name verilerinde ilk 'a' karakterinin kaçınıcı sırada olduğunu gösterir.

LPAD ve RPAD

```
SELECT LPAD('HelloWorld', 15, '-')
FROM DUAL;
```

NOT: Yazmış olduğum verinin sol kısmına “-“ karakteri koyarak verinin karakter sayısını On beşe tamamlar.

```
SELECT RPAD(last_name, 10, '*')
FROM employees;
```

TRIM

```
SELECT TRIM(LEADING 'a' FROM 'abcba')  
FROM DUAL;
```

NOT: Verinin başındaki “a” karakterini kaldırır ve çıktımız “bcba” olur.

```
SELECT TRIM(TRAILING 'a' FROM 'abcba')  
FROM DUAL;
```

NOT: Verinin sonundaki “a” karakterini kaldırır ve çıktımız “abcb” olur.

```
SELECT TRIM(BOTH 'a' FROM 'abcba')  
FROM DUAL;
```

NOT: Verinin her iki tarafındaki “a” karakterini kaldırır ve çıktımız “bcb” olur.

REPLACE

```
REPLACE { string1, string_to_replace, [replacement_string] }
```

```
SELECT last_name, REPLACE(last_name,'a','*') AS "Replaced"  
FROM employees;
```

NOT: last_name verilerinde “a” karakterlerini kaldırıp yerine “*” koyar.

SUBSTITUTION VARIABLES

```
SELECT first_name, last_name, salary, department_id  
FROM employees  
WHERE department_id=10;
```

NOT : Çağırdığımız veri statik olarak gelir.

```
SELECT first_name, last_name, salary, department_id  
FROM employees  
WHERE department_id=:bolum_kodunu_gir;
```

NOT : Çağırdığımız veriyi kullanıcı girer.

ROUND ve TRUNC

`ROUND(45.936, 0)==ROUND(45.936)==46`

`ROUND(45.926, 2)==45.93`

NOT: “926”daki “26” olan ilk iki hane kısmında yuvarlama işlemi yaptı.

`TRUNC(45.936, 0)==TRUNC(45.936)==45`

SYSDATE

`SELECT last_name, hire_date, hire_date + 60 as "eklenmiş_tarih"`

`FROM employees;`

NOT: hire_date verisinin üstüne 60 gün ekler. Fakat dikkat edilmesi gereken nokta, numerik olarak 60 eklenmez.

`SELECT last_name, hire_date, (SYSDATE - hire_date)/7 AS “Kıdem”`

`FROM employees;`

NOT: Hafta cinsinden last_name verisine sahip çalışanın kıdem süresini verir.

DATE FUNCTIONS

-MONTHS_BETWEEN: Numerik gelir

-ADD_MONTHS

-NEXT_DAY

-LAST_DAY

-ROUND

-TRUNC

```
SELECT last_name, hire_date, MONTHS_BETWEEN (SYSDATE, hire_date)
FROM employees;
```

NOT: Şu anki tarih ile hire_date verisi arasındaki tarihi numerik olarak ay cinsinden getirir.

```
SELECT last_name, hire_date, MONTHS_BETWEEN (SYSDATE, hire_date)
FROM employees;
```

```
WHERE MONTHS_BETWEEN
```

```
(SYSDATE, hire_date) > 340;
```

NOT: Şu anki tarih ile hire_date verisi arasında 340 aydan daha fazla olan verileri getirir.

```
SELECT ADD_MONTHS (SYSDATE, 6) as "Next Year"
```

```
FROM DUAL;
```

NOT: Şu anki tarihe altı ay ekleyip getirir.

```
SELECT NEXT_DAY (SYSDATE, 'Friday') as "Next Friday"
```

```
FROM DUAL;
```

NOT: Şu anki tarihten itibaren gelecek Cuma gününü tarih formatında getirir.

```
SELECT LAST_DAY (SYSDATE) as "End of the Month"
```

```
FROM DUAL;
```

NOT: Şu anki tarihte bulunduğumuz ayın son gününü getirir.

```
SELECT hire_date, ROUND(hire_date, 'Month') AS "Rounded HIRE_DATE"
```

```
FROM employees
```

```
WHERE department_id=50;
```

NOT: Diğer ayın ilk gününü date olarak yuvarlayıp getirir.

```
SELECT hire_date, ROUND(hire_date, 'Year')
```

```
FROM employees
```

```
WHERE department_id=50;
```

NOT: Burada dikkat edilmesi gereken nokta aşağı yuvarladığımızda yılın değişmediği fakat yukarı yuvarlayınca değişebildiğidir.

```
SELECT hire_date, TRUNC(hire_date, 'Month')
```

```
FROM employees
```

```
WHERE department_id=50;
```

```
SELECT hire_date, TRUNC(hire_date, 'Year')
```

```
FROM employees
```

```
WHERE department_id=50;
```

```
SELECT employee_id, hire_date,
```

```
ROUND(MONTHS_BETWEEN(SYSDATE, hire_date) AS "TENURE"
```

```
ADD_MONTHS(hire_date, 6) AS "REVIEW"
```

```
NEXT_DAY(hire_date, 'FRIDAY'), LAST_DAY(hire_date)
```

```
FROM employees
```

```
WHERE MONTHS_BETWEEN (SYSDATE, hire_date) > 36;
```

DATA CONVERSION

```
SELECT hire_date, TO_CHAR(hire_date, 'Month dd, YYYY') AS "Biçimlendirildi"  
FROM employees;
```

```
SELECT hire_date, TO_CHAR(hire_date, 'fmMonth dd, YYYY')  
FROM employees;
```

```
SELECT hire_date, TO_CHAR(hire_date, 'fmDAYddthsp Mon, YYYY')  
FROM employees;
```

```
SELECT TO_CHAR(SYSDATE, 'hh:mm:sspm')  
FROM DUAL;
```

```
SELECT TO_CHAR(4500, '99,999') FROM DUAL;  
NOT: Eğer 450000 olsa "#####" çıktısı verirdi.
```

NOT: to_char output/çıktının formatıdır. to_number ve to_date ise inputun/girdinin formatıdır.

```
SELECT last_name, bonus, TO_NUMBER(bonus, '9999') As "Bonus"  
FROM employees  
WHERE department_id=80;
```

NOT: Bonus sütunu içerisinde içinde sayı olmasına rağmen tipi varchar2. O yüzden numb'a çevirmem gerek.

```
SELECT TO_DATE('November 3, 2001', 'Month dd, yyyy') as "Tarihe_Cevirildi"  
FROM DUAL;
```

```
SELECT TO_DATE('25 March, 2022', 'dd Month, yyyy')  
FROM DUAL;
```

NOT: Girdiyi formatlar.

```
SELECT TO_CHAR(SYSDATE, 'dd Month, yyyy')  
FROM DUAL;
```

NOT: Çıktıyı formatlar.

NOT: “fx” mutlaka bunu ara, fixle demektir.

```
SELECT TO_DATE ('27-Oct-95', 'DD-Mon-RR') AS "Date"  
FROM DUAL;
```

NOT: Şu anki tarihte yıl olarak 22 deyim. Oracle buradaki 95’i olsa olsa 1995’dir diye yorumlar. 95 yerine 49 olsa 2049, 50 olsa 1950 olurdu.

NOT: Bulunduğum yıl ve aldığım değer 0-49 ise mevcut yüzyıl. Bulunduğum yıl 0-49 ve aldığım değer 50-99 ise önceki yüzyıl olarak yorumlanır.

```
SELECT last_name, TO_CHAR (hire_date, 'DD-Mon-YY'), TO_DATE ('01-Jan-  
90', 'DD-Mon-YY')  
FROM employees  
WHERE hire_date < TO_DATE ('01-Jan-90', 'DD-Mon-YY');
```

NOT: 1990’dan küçük değerleri getirmesi lazımdı fakat çıktıda büyük değerler de var. Çünkü burada 2090’dan küçükleri kabul etti.

EKSİ NE İFADE EDER?

'-' Olduğu zaman tam sayı kısmına bakılır. -1 ise sondan bi basamağı yuvarla demek olur.

```
SELECT ROUND(30695.348,-1) as "R-1" FROM DUAL; --30700
```

```
SELECT ROUND(30695.999,-2) as "R-2" FROM DUAL; --30700
```

```
SELECT ROUND(30645.999,-1) as "R-1" FROM DUAL; --30650
```

```
SELECT ROUND(30650.999,-2) as "R-2" FROM DUAL; --30700
```

```
SELECT ROUND(30649.999,-2) as "R-2" FROM DUAL; --30600
```

Date Conversion to Character Data

- The tables show the different format models that can be used.
- When specifying time elements, note that hours (HH), minutes (MI), seconds (SS), and AM or PM can also be formatted.

YYYY	Full year in numbers
YEAR	Year spelled out
MM	Two-digit value for month
MONTH	Full name of the month
MON	Three-letter abbreviation of the month
DY	Three-letter abbreviation of the day of the week
DAY	Full name of the day of the week
DD	Numeric day of the month
DDspth	FOURTEENTH
Ddspt	Fourteenth
ddspt	fourteenth
DDD or DD or D	Day of year, month or week
HH24:MI:SS AM	5:45:32 PM
DD "of" MONTH	12 of October

Number Conversion to Character Data (VARCHAR2)

- The table illustrates some of the format elements available to use with TO_CHAR functions.

```
SELECT TO_CHAR(salary,
'$99,999') AS "Salary"
FROM employees;
```

Salary
\$24,000
\$17,000

ELEMENT	DESCRIPTION	EXAMPLE	RESULT
9	Numeric position (# of 9's determine width)	999999	1234
0	Display leading zeros	099999	001234
\$	Floating dollar sign	\$999999	\$1234
L	Floating local currency symbol	L999999	FF1234
.	Decimal point in position specified	999999.99	1234.00
,	Comma in position specified	999,999	1,234
MI	Minus signs to right (negative values)	999999MI	1234-
PR	Parenthesize negative numbers	999999PR	<1234>
EEEE	Scientific notation (must have four EEEE)	99.999999E	1,23E+03
V	Multiply by 10 n times (n= number of 9's after V)	9999V99	9999V99
B	Display zero values as blank, not 0	899999.99	1234.00

Function	Description
MONTHS_BETWEEN	Number of months between two dates
ADD_MONTHS	Add calendar months to date
NEXT_DAY	Date of the next occurrence of day of the week specified
LAST_DAY	Last day of the month
ROUND	Round date
TRUNC	Truncate date