

01 April SQL

30 Mart tarihindeki dersin sorgu kaydı ⇒

```
/=====
SELECT
=====/
/* tracks tablosundaki track isimlerini (name) sorgulayınız */
SELECT name FROM tracks;
/* tracks tablosundaki besteci(Composer) ve şarkı
imlerini (name) sorgulayınız*/
SELECT Composer,name FROM tracks;
/*album tablosundaki tüm bilgileri sorgulayınız*/
select * from albums;
/*=====
DISTINCT
=====*/
/*tracks tablosundaki composer bilgileri sorgulayınız
(TEKRARLI OLABİLİR)*/
SELECT Composer FROM tracks;
/*tracks tablosundaki composer bilgileri sorgulayınız
TEKRARSIZ) */
select DISTINCT Composer from tracks;
```

DISTINCT ⇒ ilk bulduğunu alır. İkili olarak alınca ikisine birlikte bakıyor. İkisinin tekrarsızlarına bakıyor. örneğin 1/1 varsa bir defa alır diğerlerini göstermez.

WHERE ⇒ koşul belirtirken kullanılır. Sırası önemli.

```
SELECT column_name(s) FROM table_name WHERE condition(s);
```

▼ WHERE Clause - Operator

- = Equal to
- > Greater than
- < Less than
- >= Greater than or equal
- <= Less than or equal
- <> Not equal

- BETWEEN test if a value is between a certain range of values
- LIKE Determine if a character string matches a predefined pattern
- IN Test whether or a value matches any value in a list

```
SELECT * FROM student_table WHERE lesson = 'Mathematics';
/* matematik dersini alan öğrencileri sıralar. Büyük küçük harf duyarlıdır. */
```

```
SELECT column_name(s) FROM table_name LIMIT number_rows
/* Limit sorgusu için genel komut. Sıra önemli */
```

ORDER BY Clause

```
SELECT column_name(s) FROM table_name ORDER BY column_name(s) ASC|DESC;
/* where koşulu olsaydı order by where'den sonra gelmesi gerekirdi.
SQL de belli bir sıra takip edilmesi gerekir. Order by ' de yazılan
sütunların sıraya göre tabloyu düzenler. Birden fazla sütun yazılırsa
ilk sütunda sıralandığında eşitlik olması durumunda ikinci sütundakini göre sıralar
eşitlikleri.*/
/* Default değeri ASC dir. Yani artarak sıralar.*/
/* büyük küçük harf karışık ise önce büyük harfleri sıralar sonra küçük harfleri
sıralar.*/
```

Komut

sırası

⇒

```
1 SELECT title
2 FROM SERIES
3 INNER JOIN Characters
4 ON Characters.series_id = Series.id
5 WHERE Characters.species = 'human'
6 GROUP BY Series.title
7 ORDER BY COUNT (Characters.species) DESC
8 LIMIT 1
```

```
SELECT * FROM employees ORDER BY gender DESC, first_name ASC;
/* önce gender'a azalan sırayla sıralar sonra first_name'e azalan sırayla
sıralar. */
```

```
WHERE left_conditon AND right_condition
/* örnek olarak adı ahmet yaşı 30 ise. İki sorguyuda sağlaması gerekir. */
```

```
WHERE left_conditon OR right_condition
/* iki koşuldan birisi sağlıyor ise listeler */
```

```
WHERE NOT first_condition
/* NOT bütün şartlarda kullanılabilir. Tam tersini yaptırır. Örneğin yaşı 30 değilse..*/
```

```
WHERE test_expression BETWEEN low_expression AND high_expression
/* low_expression(inclusive) high_expression (inclusive) ikiside dahil.*/
```

```
WHERE test_expression NOT BETWEEN low_expression AND high_expression
/* NOT genellikle kafa karıştırıcı olduğundan kullanmamakta fayda var. En iyisi pozitif yapı kurmak.*/
```

1 Nisan tarihindeki ders sorgu kayıtları ⇒

```
select * from tracks;
/* Birden fazla kriter varsa o kriterlere göre DISTINCTine bakar.*/
SELECT DISTINCT AlbumId, MediaTypeId FROM tracks;
/* =====
WHERE
=====*/
/* Composer'ı Jimi Hendrix olan şarkıları sorgulayınız*/
SELECT name
FROM tracks
WHERE Composer = 'Jimi Hendrix';
/* Tek satır kullanmaktaysan alt alta daha kullanışlı olur.Daha şık görünür.*/

/* invoices tablosunda Total değeri 10$ dan büyük olan faturaların tüm
bilgilerini sorgulayınız*/

SELECT *
FROM invoices
WHERE total > 10;
```

```

/*=====
LIMIT
=====*/
/*invoices tablosunda Total değeri 10$'dan büyük olan
ilk 4 kayıt'ın InvoiceId,InvoiceDate ve total
bilgilerini sorgulayınız */

SELECT InvoiceId, InvoiceDate, total
FROM invoices
WHERE total > 10
LIMIT 4;
/*=====
ORDER BY
=====*/
/*invoices tablosunda Total değeri 10$'dan büyük olan kayıtlardan işlem tarihi
(InvoiceDate) en yeni olan 10 kaydın tüm bilgilerini listeyiniz */

SELECT *
FROM invoices
WHERE total > 10
ORDER by InvoiceDate DESC
LIMIT 10;
/* invoices tablosundan CustomerId, InvoiceDate, BillingCity, total
değerlerini sorgulayınız. BillingCity artan sırayla ve total değerlerini
azalan sırayla sıralayınız.*/
SELECT CustomerId, InvoiceDate, BillingCity, total
FROM invoices
ORDER by BillingCity ASC, total DESC;

/*=====
LOGICAL OPERATORS (AND,OR,NOT)
=====*/

/* invoices tablosunda ülkesi (BillingCountry) USA olmayan kayıtları total
değerine göre AZALAN sırada listeyiniz */

SELECT *
FROM invoices
WHERE NOT BillingCountry='USA'
/* BillingCountry <> 'USA' da olabilir veya BillingCountry !='USA' da olabilir.*/
ORDER by total DESC;

/* invoices tablosunda, ülkesi (BillingCountry) USA veya Germany olan
kayıtları, InvoiceId sırasına göre artan sırada listeyiniz */

SELECT *
FROM invoices
WHERE BillingCountry = 'USA' or BillingCountry = 'Germany'
ORDER by InvoiceId ASC;

/* invoices tablosunda fatura tarihi (InvoiceDate) 01-01-2012 ile 02-01-2013
tarihleri arasındaki faturaların tüm bilgilerini listeleyiniz */

```

```

SELECT *
FROM invoices
WHERE InvoiceDate >= '2012-01-01' and InvoiceDate <= '2013-01-02';
/* 2012-01-01 00:00:00 tarihinden başlayıp 2013-01-01 23:59:59 e kadar sıralar.
Hem date hem time var ona dikkat edilmesi gerekir. Bundan kurtulmak için time
bilgisi 00:00:00 dahil edilebilir.*/

SELECT *
FROM invoices
WHERE InvoiceDate BETWEEN '2012-01-01' AND '2013-01-02';

/* invoices tablosunda fatura tarihi (InvoiceDate) 2009 ila 2011 tarihleri
arasındaki en yeni faturayı listeleyen sorguyu yazınız */

SELECT *
FROM invoices
WHERE InvoiceDate BETWEEN '2009-01-01' AND '2011-12-31'
ORDER BY InvoiceDate DESC
LIMIT 1;

/*=====
IN
=====*/
/* customers tablosunda Belgium, Norway veya Canada ,USA ülkelerinden
sipariş veren müşterilerin FirstName, LastName, country bilgilerini listeyiniz */

SELECT FirstName, LastName, country
FROM customers
WHERE country IN('Belgium', 'Norway', 'Canada', 'USA')

```