

Veri Tabanı

WHERE, DISTINCT, LIMIT, AS

Hüseyin Ahmetoğlu

WHERE

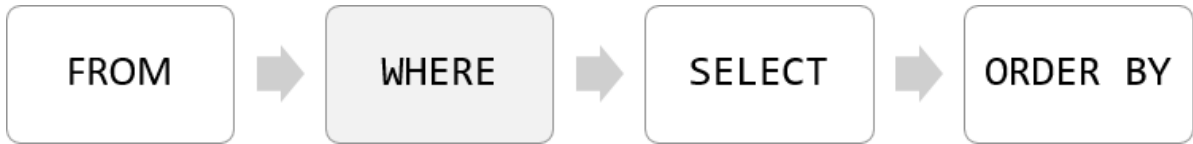
- ▶ WHERE yan tümcesi, bir sorgu tarafından döndürülen satırlar için bir arama koşulu belirtmenize olanak tanır.

```
SELECT
    select_list
FROM
    table_name
WHERE
    search_condition;
```

- ▶ Search_condition, AND, OR ve NOT mantıksal operatörünü kullanan bir veya daha fazla yüklem kombinasyonudur.
- ▶ MySQL'de bir yüklem, DOĞRU, YANLIŞ veya BİLİNMEYEN olarak değerlendirilen bir Boole ifadesidir.

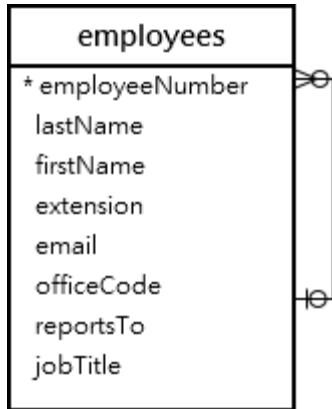
WHERE

- ▶ SELECT deyiminin yanı sıra, hangi satırların güncelleneceğini veya silineceğini belirtmek için UPDATE veya DELETE deyimlerinde de WHERE yan tümcesini kullanabilirsiniz.
- ▶ SELECT deyiminde, WHERE yan tümcesi, FROM yan tümcesinden sonra ve SELECT yan tümcesinden önce değerlendirilir.



WHERE

Örnek

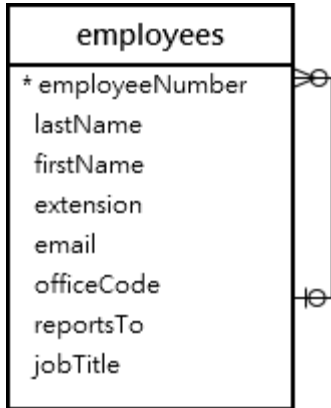


```
SELECT
    lastname,
    firstname,
    jobtitle
FROM
    employees
WHERE
    jobtitle = 'Sales Rep';
```

| | lastname | firstname | jobtitle |
|---|-----------|-----------|-----------|
| ▶ | Jennings | Leslie | Sales Rep |
| | Thompson | Leslie | Sales Rep |
| | Firelli | Julie | Sales Rep |
| | Patterson | Steve | Sales Rep |
| | Tseng | Foon Yue | Sales Rep |
| | Vanauf | George | Sales Rep |
| | Bondur | Loui | Sales Rep |

WHERE

Örnek - AND



```
SELECT
    lastname,
    firstname,
    jobtitle,
    officeCode
FROM
    employees
WHERE
    jobtitle = 'Sales Rep' AND
    officeCode = 1;
```

| | lastname | firstname | jobtitle | officeCode |
|---|----------|-----------|-----------|------------|
| ▶ | Jennings | Leslie | Sales Rep | 1 |
| | Thompson | Leslie | Sales Rep | 1 |

WHERE

Örnek - OR

```
SELECT
    lastName,
    firstName,
    jobTitle,
    officeCode
FROM
    employees
WHERE
    jobtitle = 'Sales Rep' OR
    officeCode = 1
ORDER BY
    officeCode ,
    jobTitle;
```

| | lastName | firstName | jobTitle | officeCode |
|---|-----------|-----------|--------------------|------------|
| ▶ | Murphy | Diane | President | 1 |
| | Bow | Anthony | Sales Manager (NA) | 1 |
| | Jennings | Leslie | Sales Rep | 1 |
| | Thompson | Leslie | Sales Rep | 1 |
| | Firrelli | Jeff | VP Marketing | 1 |
| | Hill | Mary | VP Sales | 1 |
| | Firrelli | Julie | Sales Rep | 2 |
| | Patterson | Steve | Sales Rep | 2 |
| | Tseng | Foon Yue | Sales Rep | 3 |
| | Vanauf | George | Sales Rep | 3 |
| | Bondur | Loui | Sales Rep | 4 |
| | Hernandez | Gerard | Sales Rep | 4 |
| | Castillo | Pamela | Sales Rep | 4 |
| | Gerard | Martin | Sales Rep | 4 |
| | Nishi | Mami | Sales Rep | 5 |
| | Kato | Yoshimi | Sales Rep | 5 |
| | Fixter | Andy | Sales Rep | 6 |
| | Marsh | Peter | Sales Rep | 6 |
| | King | Tom | Sales Rep | 6 |
| | Bott | Larry | Sales Rep | 7 |
| | Jones | Barry | Sales Rep | 7 |

WHERE

Örnek - BETWEEN

```
SELECT
    firstName,
    lastName,
    officeCode
FROM
    employees
WHERE
    officeCode BETWEEN 1 AND 3
ORDER BY officeCode;
```

| | firstName | lastName | officeCode |
|---|-----------|-----------|------------|
| ► | Diane | Murphy | 1 |
| | Mary | Hill | 1 |
| | Jeff | Firrelli | 1 |
| | Anthony | Bow | 1 |
| | Leslie | Jennings | 1 |
| | Leslie | Thompson | 1 |
| | Julie | Firrelli | 2 |
| | Steve | Patterson | 2 |
| | Foon Yue | Tseng | 3 |
| | George | Vanauf | 3 |

WHERE

Örnek - IN

```
SELECT
    firstName,
    lastName,
    officeCode
FROM
    employees
WHERE
    officeCode IN (1 , 2, 3)
ORDER BY
    officeCode;
```

| | firstName | lastName | officeCode |
|---|-----------|-----------|------------|
| ► | Diane | Murphy | 1 |
| | Mary | Hill | 1 |
| | Jeff | Firrelli | 1 |
| | Anthony | Bow | 1 |
| | Leslie | Jennings | 1 |
| | Leslie | Thompson | 1 |
| | Julie | Firrelli | 2 |
| | Steve | Patterson | 2 |
| | Foon Yue | Tseng | 3 |
| | George | Vanauf | 3 |

WHERE

Örnek - LIKE

- ▶ LIKE operatörü, bir değer belirtilen bir desene eşleşirse TRUE olarak değerlendirilir. Bir kalıp oluşturmak için % ve _ joker karakterlerini kullanırsınız. % Joker karakteri, sıfır veya daha fazla karakterden oluşan herhangi bir dizeyle eşleşirken, _ joker karakteri herhangi bir tek karakterle eşleşir.

```
SELECT
    firstName,
    lastName
FROM
    employees
WHERE
    lastName LIKE '%son'
ORDER BY firstName;
```

| | firstName | lastName |
|---|-----------|-----------|
| ▶ | Leslie | Thompson |
| | Steve | Patterson |
| | William | Patterson |

WHERE

Örnek – IS NULL

- Bir değerin NULL olup olmadığını kontrol etmek için, eşit işleci (=) değil, IS NULL işlecini kullanırsınız. IS NULL operatörü, bir değer NULL ise TRUE döndürür.

```
SELECT
    lastName,
    firstName,
    reportsTo
FROM
    employees
WHERE
    reportsTo IS NULL;
```

| | lastName | firstName | reportsTo |
|---|----------|-----------|-----------|
| ► | Murphy | Diane | NULL |

WHERE

Karşılaştırma Operatörleri

| Operator | Description |
|----------|--|
| = | Equal to. You can use it with almost any data types. |
| <> or != | Not equal to |
| < | Less than. You typically use it with numeric and date/time data types. |
| > | Greater than. |
| <= | Less than or equal to |
| >= | Greater than or equal to |

WHERE

Karşılaştırma Operatörleri

```
SELECT
    lastname,
    firstname,
    jobtitle
FROM
    employees
WHERE
    jobtitle <> 'Sales Rep';
```

| | lastname | firstname | jobtitle |
|---|-----------|-----------|----------------------|
| ► | Murphy | Diane | President |
| | Patterson | Mary | VP Sales |
| | Firrelli | Jeff | VP Marketing |
| | Patterson | William | Sales Manager (APAC) |
| | Bondur | Gerard | Sale Manager (EMEA) |
| | Bow | Anthony | Sales Manager (NA) |

WHERE

Karşılaştırma Operatörleri

```
SELECT
    lastname,
    firstname,
    officeCode
FROM
    employees
WHERE
    officecode > 5;
```

| | lastname | firstname | officeCode |
|---|-----------|-----------|------------|
| ► | Patterson | William | 6 |
| | Bott | Larry | 7 |
| | Jones | Bary | 7 |
| | Fixter | Andy | 6 |
| | Marsh | Peter | 6 |
| | King | Tom | 6 |

WHERE

Karşılaştırma Operatörleri

```
SELECT
    lastname,
    firstname,
    officeCode
FROM
    employees
WHERE
    officecode <= 4;
```

| | lastname | firstname | officeCode |
|---|-----------|-----------|------------|
| ▶ | Murphy | Diane | 1 |
| | Patterson | Mary | 1 |
| | Firelli | Jeff | 1 |
| | Bondur | Gerard | 4 |

DISTINCT

- Bir tablodan veri sorgularken, yinelenen satırlar alabilirsiniz. Bu yinelenen satırları kaldırmak için, SELECT deyiminde DISTINCT yan tümcesini kullanırsınız.

```
SELECT DISTINCT  
    select_list  
FROM  
    table_name;
```

DISTINCT

Örnek

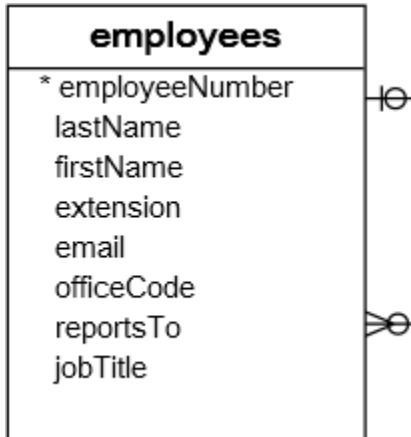
| employees | |
|------------------|--|
| * employeeNumber | |
| lastName | |
| firstName | |
| extension | |
| email | |
| officeCode | |
| reportsTo | |
| jobTitle | |

```
SELECT
    DISTINCT lastname
FROM
    employees
ORDER BY
    lastname;
```

| lastname |
|-----------|
| Bondur |
| Bott |
| Bow |
| Castillo |
| Firrelli |
| Fixter |
| Gerard |
| Hernandez |
| Jennings |
| Jones |

DISTINCT

Örnek



```
SELECT
    lastname
FROM
    employees
ORDER BY
    lastname;
```

| | lastname |
|--|-----------|
| | Bondur |
| | Bondur |
| | Bott |
| | Bow |
| | Castillo |
| | Firrelli |
| | Firrelli |
| | Fixter |
| | Gerard |
| | Hernandez |
| | Jennings |

DISTINCT

Örnek-NULL

- Bir sütunun NULL değerleri varsa ve bu sütun için DISTINCT deyimini kullanırsanız, MySQL yalnızca bir NULL değeri saklar çünkü DISTINCT tüm NULL değerleri aynı değer olarak değerlendirir.

| customers |
|------------------------|
| * customerNumber |
| customerName |
| contactLastName |
| contactFirstName |
| phone |
| addressLine1 |
| addressLine2 |
| city |
| state |
| postalCode |
| country |
| salesRepEmployeeNumber |
| creditLimit |

```
SELECT DISTINCT state  
FROM customers;
```

| state |
|---------------|
| NULL |
| NV |
| Victoria |
| CA |
| NY |
| PA |
| CT |
| MA |
| Osaka |
| BC |
| Québec |
| Isle of Wight |
| NSW |
| NJ |

DISTINCT

Örnek-Çoklu sütun

- ▶ DISTINCT yan tümcesini birden fazla sütunla kullanabilirsiniz. Bu durumda MySQL, sonuç kümesindeki satırın benzersizliğini belirlemek için bu sütunlardaki değerlerin kombinasyonunu kullanır.

```
SELECT DISTINCT
    state, city
FROM
    customers
WHERE
    state IS NOT NULL
ORDER BY
    state,
    city;
```

| state | city |
|----------|---------------|
| BC | Tsawassen |
| BC | Vancouver |
| CA | Brisbane |
| CA | Burbank |
| CA | Burlingame |
| CA | Glendale |
| CA | Los Angeles |
| CA | Pasadena |
| CA | San Diego |
| CA | San Francisco |
| CA | San Jose |
| CA | San Rafael |
| Co. Cork | Cork |
| CT | Bridgewater |
| CT | Glendale |

DISTINCT

Örnek-Çoklu sütun

```
SELECT
    state, city
FROM
    customers
WHERE
    state IS NOT NULL
ORDER BY
    state ,
    city;
```

| state | city |
|----------|---------------|
| BC | Tsawassen |
| BC | Vancouver |
| CA | Brisbane |
| CA | Burbank |
| CA | Burlingame |
| CA | Glendale |
| CA | Los Angeles |
| CA | Pasadena |
| CA | San Diego |
| CA | San Francisco |
| CA | San Francisco |
| CA | San Jose |
| CA | San Rafael |
| Co. Cork | Cork |
| CT | Bridgewater |
| CT | Glendale |

DISTINCT

Örnek-COUNT

- ▶ Sonuç kümesine toplama işlevleri uygulanmadan önce yinelenen satırları kaldırmak için DISTINCT yan tümcesini bir toplama işleviyle birlikte, örneğin SUM, AVG ve COUNT kullanabilirsiniz.
- ▶ Örneğin, ABD'deki müşterilerin benzersiz bölgelerini saymak için aşağıdaki sorguyu kullanırsınız:

```
SELECT
    COUNT(DISTINCT state)
FROM
    customers
WHERE
    country = 'USA';
```

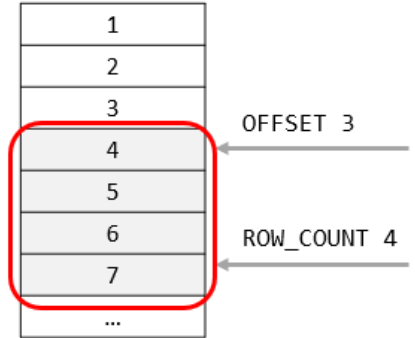
| | COUNT(DISTINCT state) |
|---|-----------------------|
| ▶ | 8 |

LIMIT

- ▶ LIMIT yan tümcesi, döndürülecek satır sayısını sınırlamak için SELECT deyiminde kullanılır. LIMIT yan tümcesi bir veya iki bağımsız değişkeni kabul eder. Her iki bağımsız değişkenin değerleri sıfır veya pozitif tamsayılar olmalıdır.

```
SELECT
    select_list
FROM
    table_name
LIMIT [offset,] row_count;
```

```
SELECT n FROM t
ORDER BY n
LIMIT 3, 4;
```



LIMIT

- ▶ ORDER BY yan tümcesi olmayan SELECT deyimi, satırları belirtilmemiş bir sırayla döndürür. Bu, satırların herhangi bir sırada olabileceği anlamına gelir. LIMIT yan tümcesini bu sıralanmamış sonuç kümesine uyguladığınızda, sorgunun hangi satırları döndüreceğini bilemezsiniz.

```
SELECT select_list  
FROM table_name  
ORDER BY order_expression  
LIMIT offset, row_count;
```



LIMIT

Örnek

| customers |
|------------------------|
| * customerNumber |
| customerName |
| contactLastName |
| contactFirstName |
| phone |
| addressLine1 |
| addressLine2 |
| city |
| state |
| postalCode |
| country |
| salesRepEmployeeNumber |
| creditLimit |

```
SELECT
    customerNumber,
    customerName,
    creditLimit
FROM
    customers
ORDER BY creditLimit DESC
LIMIT 5;
```

| | customernumber | customename | creditlimit |
|---|----------------|------------------------------|-------------|
| ► | 141 | Euro+ Shopping Channel | 227600 |
| | 124 | Mini Gifts Distributors Ltd. | 210500 |
| | 298 | Vida Sport, Ltd | 141300 |
| | 151 | Muscle Machine Inc | 138500 |
| | 187 | AV Stores. Co | 136800 |

LIMIT

Örnek

```
SELECT
    customerNumber,
    customerName,
    creditLimit
FROM
    customers
ORDER BY
    creditLimit,
    customerNumber
LIMIT 5;
```

| | customerNumber | customerName | creditLimit |
|---|----------------|----------------------------|-------------|
| ▶ | 125 | Havel & Zbyszek Co | 0.00 |
| | 168 | American Souvenirs Inc | 0.00 |
| | 169 | Porto Imports Co. | 0.00 |
| | 206 | Asian Shopping Network, Co | 0.00 |
| | 223 | Natürlich Autos | 0.00 |

LIMIT

Örnek-Sayfalama

- Uygulamalarda veri görüntülediğinizde, genellikle satırları, her sayfanın 5, 10 veya 20 gibi belirli sayıda satır içerdiği sayfalara bölmek istersiniz.
- Sayfa sayısını hesaplamak için, toplam satırların sayfa başına satır sayısına bölünmesini elde edersiniz. Belirli bir sayfanın satırlarını getirmek için LIMIT yan tümcesini kullanabilirsiniz.

```
SELECT COUNT(*) FROM customers;
```

```
+-----+  
| COUNT(*) |  
+-----+  
|      122 |  
+-----+  
1 row in set (0.00 sec)
```

LIMIT

Örnek-Sayfalama

```
SELECT
    customerNumber,
    customerName
FROM
    customers
ORDER BY customerName
LIMIT 10;
```

```
SELECT
    customerNumber,
    customerName
FROM
    customers
ORDER BY customerName
LIMIT 10, 10;
```

| | customerNumber | customerName |
|---|----------------|------------------------------|
| ▶ | 242 | Alpha Cognac |
| | 168 | American Souvenirs Inc |
| | 249 | Amica Models & Co. |
| | 237 | ANG Resellers |
| | 276 | Anna's Decorations, Ltd |
| | 465 | Anton Designs, Ltd. |
| | 206 | Asian Shopping Network, Co |
| | 348 | Asian Treasures, Inc. |
| | 103 | Atelier graphique |
| | 471 | Australian Collectables, Ltd |

| | customerNumber | customerName |
|---|----------------|------------------------------------|
| ▶ | 114 | Australian Collectors, Co. |
| | 333 | Australian Gift Network, Co |
| | 256 | Auto Associés & Cie. |
| | 406 | Auto Canal+ Petit |
| | 198 | Auto-Moto Classics Inc. |
| | 187 | AV Stores, Co. |
| | 121 | Baane Mini Imports |
| | 415 | Bavarian Collectables Imports, Co. |
| | 293 | BG&E Collectables |
| | 128 | Blauer See Auto, Co. |

LIMIT

Örnek-Max ya da Min

- N'inci en yüksek veya en düşük değeri elde etmek için yandaki LIMIT cümlesi kullanılabilir.

```
SELECT
    customerName,
    creditLimit
FROM
    customers
ORDER BY
    creditLimit DESC
LIMIT 1,1;
```

| | customerName | creditLimit |
|---|------------------------------|-------------|
| ► | Mini Gifts Distributors Ltd. | 210500.00 |

```
SELECT select_list
FROM table_name
ORDER BY sort_expression
LIMIT n-1, 1;
```

| | customerName | creditLimit |
|---|------------------------------|-------------|
| ► | Euro+ Shopping Channel | 227600.00 |
| | Mini Gifts Distributors Ltd. | 210500.00 |
| | Vida Sport, Ltd | 141300.00 |
| | Muscle Machine Inc | 138500.00 |
| | AV Stores, Co. | 136800.00 |
| | Saveley & Henriot, Co. | 123900.00 |
| | Marta's Replicas Co. | 123700.00 |
| | L'ordine Souvenirs | 121400.00 |
| | Heintze Collectables | 120800.00 |
| | Toms Spezialitäten, Ltd | 120400.00 |
| | Rovelli Gifts | 119600.00 |

Alias

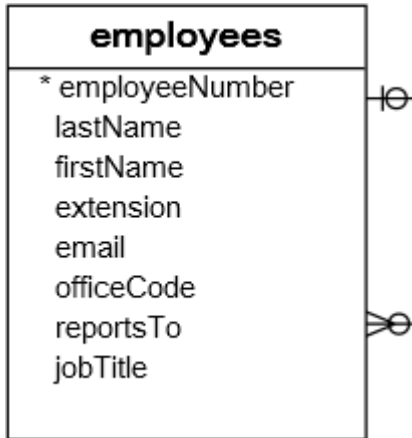
- Bazen sütun adları o kadar tekniktir ki, sorgunun çıktısının anlaşılmasını çok zorlaştırır. Bir sütuna açıklayıcı bir ad vermek için sütun takma adı kullanabilirsiniz.

```
SELECT  
    [column_1 | expression] AS descriptive_name  
FROM table_name;
```

```
SELECT  
    [column_1 | expression] AS `descriptive name`  
FROM  
    table_name;
```

Alias

Örnek

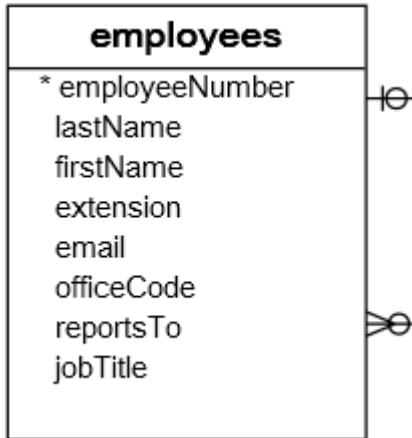


```
SELECT
    CONCAT_WS(', ', lastName, firstName)
FROM
    employees;
```

| | CONCAT_WS(', ', lastName, firstName) |
|--|--------------------------------------|
| | Murphy, Diane |
| | Patterson, Mary |
| | Firrelli, Jeff |
| | Patterson, William |
| | Bondur, Gerard |
| | Bow, Anthony |
| | Jennings, Leslie |
| | Thompson, Leslie |
| | Firrelli, Julie |
| | Patterson, Steve |

Alias

Örnek



```
SELECT
    CONCAT_WS(' ', lastName, firstName) AS `Full name`
FROM
    employees;
```

| Full name |
|--------------------|
| Murphy, Diane |
| Patterson, Mary |
| Firrelli, Jeff |
| Patterson, William |
| Bondur, Gerard |
| Bow, Anthony |
| Jennings, Leslie |
| Thompson, Leslie |
| Firrelli, Julie |
| Patterson, Steve |

Alias

Örnek

```
SELECT
    CONCAT_WS(', ', lastName, firstname) `Full name`
FROM
    employees
ORDER BY
    `Full name`;
```

| | Full name |
|--|-------------------|
| | Bondur, Gerard |
| | Bondur, Loui |
| | Bott, Larry |
| | Bow, Anthony |
| | Castillo, Pamela |
| | Firrelli, Jeff |
| | Firrelli, Julie |
| | Fixter, Andy |
| | Gerard, Martin |
| | Hernandez, Gerard |

Alias

Tablo Adları

```
SELECT * FROM employees e;
```

```
SELECT
    e.firstName,
    e.lastName
FROM
    employees e
ORDER BY e.firstName;
```

KAYNAKLAR

- ▶ MySQL Tutorial - Learn MySQL Fast, Easy and Fun. (2020, March 30). Retrieved from <https://www.mysqltutorial.org>