

**T.C.**  
**ATATÜRK ÜNİVERSİTESİ**  
**İKTİSADİ VE İDARİ BİLİMLER FAKÜLTESİ**  
**YÖNETİM BİLİŞİM SİSTEMLERİ BÖLÜMÜ**

**MySQL SQL DEYİMLERİ ARAŞTIRILMASI**

**Hazırlayan**  
**İbrahim Ethem RONA**  
**200307063**

**Dersin Öğretim Üyesi**  
**Dr. Öğr. Üyesi Muhammed Fatih ALAEDDİNOĞLU**

**Erzurum 2022**

## İçindekiler

SQL Nedir?.....	3
SQL SELECT Statement (Deyimi).....	3
SQL SELECT DISTINCT Statement (Deyimi) .....	4
WHERE Clause (Tümce) .....	5
AND, OR ve NOT Operators .....	6
ORDER BY Keyboard .....	9
INSERT INTO Statement (Deyimi).....	11
NULL Value (Değeri) .....	13
UPDATE Statement (Deyimi) .....	14
DELETE Statement (Deyimi).....	16
LIMIT Clause (Deyimi) .....	17
MIN() ve MAX() Functions (İşlevleri) .....	17
COUNT(), SUM() ve AVG() Functions (İşlevleri) .....	19
LIKE Operator .....	20
IN Operator.....	23
BETWEEN Operator .....	24
Aliases (AS).....	26
JOIN TABLES (Birleştirme Tabloları).....	28
INNER JOIN .....	29
LEFT JOIN.....	30
RIGHT JOIN .....	31
CROSS JOIN.....	32
SELF JOIN .....	34
UNION Operator .....	34
GROUP BY Statement (Deyimi).....	37
HAVING Clause.....	38
EXISTS Operator.....	39
ANY and ALL Operators .....	40
ANY Operator .....	41
ALL Operator .....	41
INSERT INTO SELECT Statement (Deyimi).....	44
CASE Statement (Deyimi) .....	45
COMMENTS (Yorum Satırları).....	46
Arithmetich Operators (Aritmetik Operatörler).....	47
Bitwise Operators (Bit Operatörler).....	47

## SQL Nedir?

SQL, ilişkisel veritabanları ile başa çıkmak için kullanılan standart bir dildir.

SQL, veritabanı kayıtlarını eklemek, aramak, güncelleştirmek ve silmek için kullanılır.

**Not :** *SQL anahtar sözcükleri büyük/küçük harf duyarlı değildir: **select – SELECT***

## SQL SELECT Statement (Deyimi)

Veritabanından veri seçmek için kullanılır. Döndürülen veriler sonuç kümesi adı verilen bir sonuç tablosunda depolanır.

### SELECT Sözdizimi;

```
SELECT sütun1, sütun2, ...  
FROM tablo_adi
```

Burada **sütun1, sütun2, ...** veri seçmek istediğiniz tablonun alan adlarıdır. Tabloda kullanılabilen tüm alan adlarını seçmek istiyorsanız aşağıdaki sözdizimine bakın.

```
SELECT *  
FROM tablo_adi
```

Bu öğreticide **customers** veritabanını kullanacağız.

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
	3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
	4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
	5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
	8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
	9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
	10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## SELECT Sütunlar Örneği

Aşağıdaki SQL deyimi, **customers** tablosundan **first\_name** ve **last\_name** sütunlarını seçer:

```
SELECT first_name, last_name  
FROM customers
```

	first_name	last_name
▶	Babara	MacCaffrey
	Ines	Brushfield
	Freddi	Boagey
	Ambur	Roseburgh
	Clemmie	Betchley
	Elka	Twiddell
	Ilene	Dowson
	Thacher	Naseby
	Romola	Rumgay
	Levy	Mynett

## SELECT (\*) Örneği

Yukarıdaki SQL deyimi *customers* tablosundan tüm sütunları seçer:

```
SELECT *
FROM customers
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
	3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
	4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
	5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
	8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
	9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
	10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## SQL SELECT DISTINCT Statement (Deyimi)

Yalnızca farklı değerler döndürmek için kullanılır. Tablonun içinde, bir sütunda birçok yinelenen (tekrarlanan) değerler içerir. Birbirinden farklı değerleri listelemek isterseniz: ***SELECT DISTINCT***

### SELECT DISTINCT Sözdizimi;

```
SELECT DISTINCT sütun1, sütun2, ...
FROM tablo_adi
```

Aşağıda görmüş olduğunuz gibi *state (eyalet)* sütununda yinelenen veriler var. *state (eyalet)* sütununu seçmek için:

```
SELECT state
FROM customers
```

	state
▶	VA
	VA
	CO
	FL
	TX
	IL
	TN
	FL
	CA
	GA

Yukarıda yinelenen iki tane “VA” eyaleti var. Buna ***SELECT DISTINCT*** uygulayalım. Aşağıdaki SQL deyimi yalnızca ***customers*** tablosundaki ***state (eyalet)*** sütunundan ***DISTINCT*** değerlerini seçer:

```
SELECT DISTINCT state
FROM customers
```

	state
▶	VA
	CO
	FL
	TX
	IL
	TN
	CA
	GA

## WHERE Clause (Tümce)

Kayıtları filtrelemek ve yalnızca belirli bir koşulu yerine getiren kayıtları ayıklamak için kullanılır. ***WHERE***

## WHERE Sözdizimi ;

```
SELECT sütun1, sütun2, ...
FROM tablo_adi
WHERE koşul
```

**Not :** ***WHERE*** sadece ifadelerde kullanılmaz. Aynı zamanda ***SELECT***, ***UPDATE***, ***DELETE*** gibi deyimlerde kullanılır.

## WHERE Örneği

```
SELECT *  
FROM customers  
WHERE city = 'Chicago'
```

Yukarıdaki SQL deyimi *customers* tablosunda *city* = 'Chicago' olanların tüm kaydını verir.

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Metin alanları ve Sayısal alanları Karşılaştırma

SQL, metin değerleri etrafında tek tırnak işaretleri gerektirir. Çoğu veritabanı sistemi çift tırnak işaretine izin verir. Ancak, sayısal alanlar tırnak içine alınmamalıdır.

```
SELECT *  
FROM customers  
WHERE customer_id = 1
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## WHERE İşleçleri

Operatörler	Tanım
=	Eşittir.
>	Büyüktür.
<	Küçüktür.
>=	Büyük eşittir.
<=	Küçük eşittir.
!=	Eşit değildir.
BETWEEN	Belirli aralıklar
LIKE	İfade arama
IN	Bir sütunda birden çok olası durum belirtmek için kullanılır.

## AND, OR ve NOT Operators

**AND**, **OR**, **NOT** işleçleri, kayıtları birden fazla koşula göre filtrelemek için kullanılır.

- Ayrılmış tüm koşullar **DOĞRU** ise, işleç bir kayıt görüntüler. **AND**
- Ayrılmış koşullardan herhangi biri **DOĞRU** ise, işleç bir kayıt görüntüler. **OR**
- Koşul **DOĞRU DEĞİL** ise, işleç bir kayıt görüntüler. **NOT**

### AND Sözdizimi;

```
SELECT sütun1, sütun2, ...
FROM tablo_adi
WHERE koşul AND koşul AND koşul
```

### OR Sözdizimi;

```
SELECT sütun1, sütun2, ...
FROM tablo_adi
WHERE koşul OR koşul OR koşul
```

### NOT Sözdizimi;

```
SELECT sütun1, sütun2, ...
FROM tablo_adi
WHERE NOT koşul
```

### AND Örnek

Aşağıdaki SQL deyimi, *state* = “VA” ve *points* > 1000 olduğu *customers* tablosundaki tüm alanları seçer:

```
SELECT *
FROM customers
WHERE state = 'VA' AND points > 1000
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL	HULL

### OR Örnek

Aşağıdaki SQL deyimi, *birth\_date* = “1990-01-01” veya *points* > 1000 olduğu *customers* tablosundaki tüm alanları seçer:

```
SELECT *
FROM customers
WHERE birth_date = '1990-01-01' OR points > 1000
```

Aşağıdaki görüntüde fark ettiğiniz gibi herhangi bir koşul doğru ise tüm kayıtları getirir. Buradaki ifadede 1990 doğumlu herhangi bir kayıt yok fakat puanı 1000' den büyük olan kayıtlar vardır.

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
	3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
	5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
	9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## NOT Örnek

Aşağıdaki SQL deyimi, **state = "VA"** olmadığı **customers** tablosundaki tüm alanları seçer:

```
SELECT *
FROM customers
WHERE NOT state = 'VA'
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
	4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
	5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
	8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
	9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
	10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## AND, OR, NOT Birleştirilmesi

**Not :** Matematiksel operatörler gibi AND, OR, NOT operatörlerinde de işlem önceliği vardır. Öncelik çarpma ve bölmede olduğu gibi AND operatörüne aittir.

Aşağıdaki SQL deyimi, **state = 'VA'** veya **points > 1000** ve **birth\_date > '1990-01-01'** olan tüm alanları seçer:

```
SELECT *
FROM customers
WHERE state = 'FL' OR points > 1000 AND birth_date > '1990-01-01'
```



	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
	9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Dediğimiz gibi işlem önceliği AND' e ait işlem önceliğine göre işlemleri kullanmak istiyorsak parantezler yardımıyla ayırabiliriz:

```
SELECT *
FROM customers
WHERE (state = 'FL' OR points > 1000) AND birth_date > '1990-01-01'
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
	9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Aşağıdaki SQL deyimi, *city* = 'Atlanta' ve *city* = 'Sarasota' olmayan tüm alanları seçer:

```
SELECT *
FROM customers
WHERE NOT city = 'Atlanta' AND NOT city = 'Sarasota'
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
	3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
	4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
	5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
	9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## ORDER BY Keyboard

Artan ve azalan düzende sıralama yapmak için kullanılır. **ORDER BY**

Bu anahtar sözcük kelimeyi var sayılan olarak artan sırada sıralar fakat kayıtları artan düzende sıralama yapmak için **ASC** anahtar sözcüğünü kullanırız.

Kayıtları azalan düzende sıralamak için **DESC** anahtar sözcüğünü kullanırız.

## ORDER BY Sözdizimi;

```
SELECT sütun1, sütun2, ...  
FROM tablo_adi  
ORDER BY sütun1, sütun2, ... ASC | DESC
```

Aşağıdaki SQL deyimi, *first\_name* sütununa göre sıralanmış *customers* tablosunda olan tüm kayıtları seçer:

```
SELECT first_name  
FROM customers  
ORDER BY first_name
```

Tablonun ilk hali

	first_name
▶	Babara
	Ines
	Freddi
	Ambur
	Clemmie
	Elka
	Ilene
	Thacher
	Romola
	Levy

Tablonun artan olarak sıralanmış hali

	first_name
▶	Ambur
	Babara
	Clemmie
	Elka
	Freddi
	Ilene
	Ines
	Levy
	Romola
	Thacher

## ORDER BY DESC Örneği

Aşağıdaki SQL deyimi, *points* sütununa göre azalan düzende sıralanmış *customers* tablosundaki tüm kayıtları seçer:

```
SELECT points  
FROM customers  
ORDER BY points DESC
```

Tablonun ilk hali

	points
▶	2273
	947
	2967
	457
	3675
	3073
	1672
	205
	1486
	796

Tablonun azalana olarak sıralanmış hali

	points
▶	3675
	3073
	2967
	2273
	1672
	1486
	947
	796
	457
	205

## Birkaç Sütun Örneği

Aşağıdaki SQL deyimi, *city* ve *points* sütununa göre sıralanmış *customers* tablosundaki tüm kayıtları seçer:

```
SELECT *  
FROM customers  
ORDER BY city, points
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
	10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
	7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
	4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
	8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
	9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## INSERT INTO Statement (Deyimi)

Tabloya yeni kayıtlar eklemek için kullanılır. **INSERT INTO**

### INSERT INTO Sözdizimi;

İfadeyi iki şekilde yazmak mümkündür:

1. Hem sütun adlarını hem de eklenecek değerleri belirtin:

```
INSERT INTO tablo_adi (sütun1, sütun2, ...)  
VALUES (value1, value2, ...)
```

2. Tablonun tüm sütunları için değer ekliyorsanız, SQL sorgusunda sütun adlarını belirtmeniz gerekmez. Ancak değerlerin sırasını tablodaki sırasıyla aynı olduğuna emin olun:

```
INSERT INTO tablo_adi  
VALUES (value1, value2, ...)
```

## ÖRNEK INSERT INTO

Aşağıdaki SQL deyimi, *customers* tablosuna yeni veri ekle:

```
INSERT INTO customers  
VALUES (12, 'İbrahim', 'Rona', '1998-02-08', '424-265-3350', 'Atatürk  
Üniversitesi', 'Erzurum', 'ER', 2000)
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
	3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
	4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
	5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
	8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
	9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
	10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
	12	İbrahim	Rona	1998-02-08	424-265-3350	Atatürk Üniversitesi	Erzurum	ER	2000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Yalnızca Belirtilen Sütunlara Veri Ekleme

Aşağıdaki SQL deyimi, *customers* tablosundan yalnızca belirtilen sütunlarına yeni veri ekle:

```
INSERT INTO customers (first_name, last_name, address, city, state)  
VALUES ('Orlando', 'Black', '305 - 14th Ave. S. Suite 3B', 'Seattle', 'US')
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
	3	Freddi	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Colorado Springs	CO	2967
	4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457
	5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
	6	Elka	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	7	Ilene	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
	8	Thacher	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	205
	9	Romola	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
	10	Levy	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
	12	İbrahim	Rona	1998-02-08	424-265-3350	Atatürk Üniversitesi	Erzurum	ER	2000
	13	Orlando	Black	NULL	NULL	305 - 14th Ave. S. Sui...	Seattle	US	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## NULL Value (Değeri)

Boş değeri olan bir alan veya değeri olmayan bir alandır.

Tablodaki bir alan isteğe bağlıysa, bu alana değer eklemekten yeni bir kayıt eklemek veya bir kaydı güncelleştirmek mümkündür. Ardından, alan NULL değeriyle kaydedilir.

**Not :** NULL değeri sıfır değerinden veya boşluk içeren bir alandan farklıdır. NULL değeri olan bir alana, kayıt oluşturma sırasında boş bırakılmış bir alandır.

NULL değerleri =, < veya > gibi karşılaştırma operatörleriyle sınanması mümkün değildir.

Bunun yerine **IS NULL** ve **IS NOT NULL** operatörlerini kullanacağız.

### IS NULL Sözdizimi;

```
SELECT sütun_adi  
FROM tablo_adi  
WHERE sütun_adi IS NULL
```

### IS NOT NULL Sözdizimi;

```
SELECT sütun_adi  
FROM tablo_adi  
WHERE sütun_adi IS NOT NULL
```

### IS NULL Örneği

Aşağıdaki SQL deyimi, **phone** sütununda boş olan tüm müşterileri listeler:

```
SELECT first_name, last_name, phone  
FROM customers  
WHERE phone IS NULL
```

	first_name	last_name	phone
▶	Clemmie	Betchley	NULL
	Orlando	Black	NULL



## IS NULL Örneği

Aşağıdaki SQL deyimi, **address** sütununda değeri olan tüm müşterileri listeler:

```
SELECT first_name, last_name, address
FROM customers
WHERE address IS NOT NULL
```

	first_name	last_name	address
▶	Babara	MacCaffrey	0 Sage Terrace
	Ines	Brushfield	14187 Commercial Trail
	Freddi	Boagey	251 Springs Junction
	Ambur	Roseburgh	30 Arapahoe Terrace
	Clemmie	Betchley	5 Spohn Circle
	Elka	Twiddell	7 Manley Drive
	İlene	Dowson	50 Lillian Crossing
	Thacher	Naseby	538 Mosinee Center
	Romola	Rumgay	3520 Ohio Trail
	Levy	Mynett	68 Lawn Avenue
	İbrahim	Rona	Atatürk Üniversitesi
	Orlando	Black	305 - 14th Ave. S. Sui...

## UPDATE Statement (Deyimi)

Bir tablodaki varolan kayıtları değiştirmek için kullanılır. **UPDATE**

### UPDATE Sözdizimi;

```
UPDATE tablo_adi
SET sütun1 = value1, sütun2 = value2, ...
WHERE koşul
```

**Not :** Tablodaki kayıtları güncelleştirirken dikkatli olun! İfadedeki maddeye dikkat edin **WHERE** güncelleştirmesi gereken kayıtları belirtir. **WHERE** atlatırsanız, tablodaki tüm kayıtlar güncelleştirilir.

## Tabloyu Güncelleştir

Aşağıdaki SQL deyimi, **customer\_id = 3** yeni bir kişi ve yeni bir şehir ile güncelleştirin:

```
UPDATE customers
SET first_name = 'Jack', city = 'Frankfurt'
WHERE customer_id = 3
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	1	Babara	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
	2	Ines	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
	3	Jack	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Frankfurt	CO	2967
	4	Ambur	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	457

## Birden Çok Kaydı Güncelleştir

Aşağıdaki SQL deyimi, **phone IS NULL** olduğu tüm kayıtlar için **phone = 0** olarak güncelleştir:

```
UPDATE customers
SET phone = 0
WHERE phone IS NULL
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	5	Clemmie	Betchley	1973-11-07	NULL	5 Spohn Circle	Arlington	TX	3675
	13	Orlando	Black	NULL	NULL	305 - 14th Ave. S. Suite 3B	Seattle	US	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	5	Clemmie	Betchley	1973-11-07	0	5 Spohn Circle	Arlington	TX	3675
	13	Orlando	Black	NULL	0	305 - 14th Ave. S. Suite 3B	Seattle	US	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Güncelleme Uyarısı!

Kayıtları güncelleştirirken dikkatli olun! **WHERE** atlarsanız, tüm kayıtlar güncelleştirilir.

```
UPDATE customers
SET first_name = 'Freud'
```

	customer_id	first_name	last_name	birth_date	phone	address	city	state	points
▶	1	Freud	MacCaffrey	1986-03-28	781-932-9754	0 Sage Terrace	Waltham	VA	2273
	2	Freud	Brushfield	1986-04-13	804-427-9456	14187 Commercial Trail	Hampton	VA	947
	3	Freud	Boagey	1985-02-07	719-724-7869	251 Springs Junction	Frankfurt	CO	2967
	4	Freud	Roseburgh	1974-04-14	407-231-8017	30 Arapahoe Terrace	Orlando	FL	3000
	5	Freud	Betchley	1973-11-07	0	5 Spohn Circle	Arlington	TX	3675
	6	Freud	Twiddell	1991-09-04	312-480-8498	7 Manley Drive	Chicago	IL	3073
	7	Freud	Dowson	1964-08-30	615-641-4759	50 Lillian Crossing	Nashville	TN	1672
	8	Freud	Naseby	1993-07-17	941-527-3977	538 Mosinee Center	Sarasota	FL	3000
	9	Freud	Rumgay	1992-05-23	559-181-3744	3520 Ohio Trail	Visalia	CA	1486
	10	Freud	Mynett	1969-10-13	404-246-3370	68 Lawn Avenue	Atlanta	GA	796
	12	Freud	Rona	1998-02-08	424-265-3350	Atatürk Üniversitesi	Erzurum	ER	2000
	13	Freud	Black	NULL	0	305 - 14th Ave. S. Sui...	Seattle	US	0
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## DELETE Statement (Deyimi)

Bir tablodaki varolan kayıtları silmek için kullanılır. **DELETE**

### DELETE Sözdizimi;

```
DELETE FROM tablo_adi  
WHERE koşul
```

**Not :** Tablodaki kayıtları silerken dikkatli olun! İfadedeki maddeye dikkat edin **WHERE** hangi kayıtların silinmesi gerektiğini belirtir. **WHERE** atlatırsanız, tablodaki tüm kayıtlar silinir.

Aşağıda *customers* tablosundan bir seçim verilmiştir:

	customer_id	name	surname	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
	4	Thomas	Hardy	120 Hanover Sq.	London	WA1 1DP	UK
	5	Christina	Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## DELETE Örneği

Aşağıdaki SQL deyimi, *customers* tablosundan name = 'Ana' müşterisini siler:

```
DELETE FROM customers  
WHERE name = 'Ana'
```

	customer_id	name	surname	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	Obere Str 57	Berlin	12209	Germany
	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
	4	Thomas	Hardy	120 Hanover Sq.	London	WA1 1DP	UK
	5	Christina	Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Tüm Kayıtları Sil

Aşağıdaki SQL deyimi, *customers* tablosundaki tüm kayıtları siler:

```
DELETE FROM customers
```

	customer_id	name	surname	address	city	postalCode	country
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL



## LIMIT Clause (Deyimi)

Döndürülecek kayıt sayısını belirtmek için kullanılır. **LIMIT**

Binlerce kaydı olan büyük tablolarda yararlıdır. Çok sayıda kayıt döndürmek performansı etkileyebilir.

### LIMIT Sözdizimi;

```
SELECT sütun1, sütun2, ...  
FROM tablo_adi  
WHERE koşul  
LIMIT number
```

### LIMIT Örneği

Aşağıdaki SQL deyimi, *customers* tablosundan ilk 3 kaydı seçer:

```
SELECT *  
FROM customers  
LIMIT 3
```

	customer_id	name	surname	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## WHERE EKLEME

Aşağıdaki SQL deyimi, *customers* tablosundan *city* = 'Mexico' olan ilk 2 kaydı seçer:

```
SELECT *  
FROM customers  
WHERE city = 'Mexico'  
LIMIT 2
```

	customer_id	name	surname	address	city	postalCode	country
▶	2	Ana	Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## MIN() ve MAX() Functions (İşlevleri)

Seçili sütunun en küçük değerini döndürür. **MIN()**

Seçili sütunun en büyük değerini döndürür. **MAX()**

### MIN() Sözdizimi;

```
SELECT MIN (sütun_adi)  
FROM tablo_adi  
WHERE koşul
```

## MAX() Sözdizimi;

```
SELECT MAX (sütun_adi)
FROM tablo_adi
WHERE koşul
```

Aşağıda, örnek veritabanındaki *products* tablosundan bir seçim verilmiştir:

	product_id	name	quantity_in_stock	unit_price
▶	1	Foam Dinner Plate	70	1.21
	2	Pork - Bacon,back Peameal	49	4.65
	3	Lettuce - Romaine, Heart	38	3.35
	4	Brocolinni - Gaylan, Chinese	90	4.53
	5	Sauce - Ranch Dressing	94	1.63
	6	Petit Baguette	14	2.39
	7	Sweet Pea Sprouts	98	3.29
	8	Island Oasis - Raspberry	26	0.74
	9	Longan	67	2.26
	10	Broom - Push	6	1.09
*	NULL	NULL	NULL	NULL

## MIN() Örneği

Aşağıdaki SQL deyimi, *products* tablosundan en küçük stok miktarını verir:

```
SELECT MIN(quantity_in_stock)
FROM products
```

	MIN(quantity_in_stock)
▶	6

## MAX() Örneği

Aşağıdaki SQL deyimi, *products* tablosundan en büyük stok miktarını verir:

```
SELECT MAX(quantity_in_stock)
FROM products
```

	MAX(quantity_in_stock)
▶	98

## COUNT(), SUM() ve AVG() Functions (İşlevleri)

Belirtilen ölçütle eşleşen satır sayısını döndürür. **COUNT()**

### COUNT() Sözdizimi;

```
SELECT COUNT(sütun_adi)
FROM tablo_adi
WHERE koşul
```

Sayısal bir sütunun toplam toplamını döndürür. **SUM()**

### SUM() Sözdizimi;

```
SELECT SUM(sütun_adi)
FROM tablo_adi
WHERE koşul
```

Sayısal bir sütunun ortalama değerini döndürür. **AVG()**

### AVG() Sözdizimi;

```
SELECT AVG(sütun_adi)
FROM tablo_adi
WHERE koşul
```

## COUNT() Örneği

Aşağıdaki SQL deyimi, *products* tablosundan *products\_id* sayısını bulur:

```
SELECT COUNT(products_id)
FROM products
```

	COUNT(product_id)
▶	10

**Not :** *NULL* değerleri sayılmaz.

## SUM() Örneği

Aşağıdaki SQL deyimi, *products* tablosundan *quantity\_in\_stock* toplamını bulur:

```
SELECT SUM(quantity_in_stock)
FROM products
```

	SUM(quantity_in_stock)
▶	552

**Not :** *NULL değerleri sayılmaz.*

### AVG() Örneği

Aşağıdaki SQL deyimi, **products** tablosundan **price** alanının ortalamasını bulur:

<b>SELECT</b> AVG(unit_price) <b>FROM</b> products	
	AVG(unit_price)
▶	2.514000

**Not :** *NULL değerleri sayılmaz.*

### LIKE Operator

Bir sütunda belirtilen ifadeyi aramak için **WHERE** tümcesinde kullanılır. **LIKE**

Sıklıkla kullanılan iki joker karakterleri vardır:

1. Yüzde işareti (%) sıfır, bir veya birden çok karakteri temsil eder.
2. Altı çizgi işareti (\_) tek bir karakteri temsil eder.

### LIKE Sözdizimi;

<b>SELECT</b> sütun1, sütun2, ... <b>FROM</b> tablo_adi <b>WHERE</b> sütun_adi <b>LIKE</b> ifade
--

**Not :** **AND , OR** işleçleri ile birden çok ifadeyi bağlayabilirsiniz.

LIKE Operatörü	Tanım
WHERE name LIKE 'a%'	Başlangıç karakteri "a" olan
WHERE name LIKE '%a'	Bitiş karakteri "a" olan
WHERE name LIKE '%or%'	Herhangi bir konumda "or" olan
WHERE name LIKE '_a%'	İkinci konumu "a" olan
WHERE name LIKE 'a_%'	Başlangıç karakteri "a" ve 2 karakter uzunluğunda olan
WHERE name LIKE 'a__%'	Başlangıç karakteri "a" ve 3 karakter uzunluğunda olan
WHERE name LIKE 'a%o'	"a" ile başlayan "o" ile biten

## LIKE Örnekleri

Aşağıdaki SQL deyimi, *customers* tablosunda bulunan *name* sütunundan başında “a%” harfine sahip tüm müşterileri seçer:

```
SELECT *  
FROM customers  
WHERE name LIKE 'a%'
```

	customer_id	name	surname	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Aşağıdaki SQL deyimi, *customers* tablosunda bulunan *name* sütunundan sonunda “%a” harfine sahip tüm müşterileri seçer:

```
SELECT *  
FROM customers  
WHERE name LIKE '%a'
```

	customer_id	name	surname	address	city	postalCode	country
▶	2	Ana	Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	5	Christina	Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Aşağıdaki SQL deyimi, *customers* tablosunda bulunan *surname* sütunundan herhangi bir konumunda “%or%” harflerine sahip tüm müşterileri seçer:

```
SELECT *  
FROM customers  
WHERE surname LIKE '%or%'
```

	customer_id	name	surname	address	city	postalCode	country
▶	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Aşağıdaki SQL deyimi, *customers* tablosunda bulunan *name* sütunundan ikinci konumunda “\_n%” harfine sahip tüm müşterileri seçer:

```
SELECT *  
FROM customers  
WHERE name LIKE '_n%'
```

	customer_id	name	surname	address	city	postalCode	country
▶	2	Ana	Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Aşağıdaki SQL deyimi, *customers* tablosunda bulunan **name** sütunundan “a\_ %” harfiyle başlayıp 2 karakter uzunluğuna sahip tüm müşterileri seçer:

```
SELECT *
FROM customers
WHERE name LIKE 'a_ %'
```

	customer_id	name	surname	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Aşağıdaki SQL deyimi, *customers* tablosunda bulunan **name** sütunundan “a\_\_ %” harfiyle başlayıp 3 karakter uzunluğuna sahip tüm müşterileri seçer:

```
SELECT *
FROM customers
WHERE name LIKE 'a__ %'
```

	customer_id	name	surname	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Aşağıdaki SQL deyimi, *customers* tablosunda bulunan **name** sütunundan “a%o” “a” harfiyle başlayıp “o” harfiyle biten tüm müşterileri seçer:

```
SELECT *
FROM customers
WHERE name LIKE 'a%o'
```

	customer_id	name	surname	address	city	postalCode	country
▶	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## IN Operator

Birden fazla değer belirtmemizi sağlar. **IN**  
Operatör birden fazla için stenodur. **IN**

### IN Sözdizimi;

```
SELECT sütun_adi  
FROM tablo_adi  
WHERE sütun_adi IN (value1, value2, ...)
```

### Veya

```
SELECT sütun_adi  
FROM tablo_adi  
WHERE sütun_adi IN (SELECT Statement)
```

## IN Örnekleri

Aşağıdaki SQL deyimi, *customers* tablosundan ülkeleri “UK”, “Sweden”, “Germany” bulunan tüm müşterileri seçer:

```
SELECT *  
FROM customers  
WHERE country IN ('UK', 'Sweden', 'Germany')
```

	customer_id	name	surname	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	Obere Str 57	Berlin	12209	Germany
	4	Thomas	Hardy	120 Hanover Sq.	London	WA1 1DP	UK
	5	Christina	Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Aşağıdaki SQL deyimi, *customers* tablosundan ülkeleri “UK”, “Sweden”, bulunmayan tüm müşterileri seçer:

```
SELECT *  
FROM customers  
WHERE country NOT IN ('UK', 'Sweden')
```

	customer_id	name	surname	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
★	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## BETWEEN Operator

Belirli bir aralıktaki değerleri seçer. Değerler sayı, metin veya tarih olabilir.

### BETWEEN

Başlangıç ve bitiş değerleri dahildir. **BETWEEN**

### BETWEEN Sözdizimi;

```
SELECT sütun_adi
FROM tablo_adi
WHERE sütun_adi BETWEEN value1 AND value2
```

### BETWEEN Örneği

Aşağıdaki SQL deyimi, *products* tablosundan *stok miktarı 5 ve 30* arasında olan tüm müşterileri seçer:

```
SELECT *
FROM products
WHERE quantity_in_stock BETWEEN 5 AND 30
```

	product_id	name	quantity_in_stock	unit_price
▶	6	Petit Baguette	14	2.39
	8	Island Oasis - Raspberry	26	0.74
	10	Broom - Push	6	1.09
★	NULL	NULL	NULL	NULL

### NOT BETWEEN Örneği

Aşağıdaki SQL deyimi, *products* tablosundan *stok miktarı 10 ve 90* arasında olan ve *ürün kimliği (1, 2, 3)* olanları göstermeyin:

```
SELECT *
FROM products
WHERE quantity_in_stock BETWEEN 10 AND 90
AND product_id NOT IN (1, 2, 3)
```



	product_id	name	quantity_in_stock	unit_price
▶	1	Foam Dinner Plate	70	1.21
	2	Pork - Bacon,back Peameal	49	4.65
	3	Lettuce - Romaine, Heart	38	3.35
	4	Brocolinni - Gaylan, Chinese	90	4.53
	6	Petit Baguette	14	2.39
	8	Island Oasis - Raspberry	26	0.74
	9	Longan	67	2.26
*	NULL	NULL	NULL	NULL

	product_id	name	quantity_in_stock	unit_price
▶	1	Foam Dinner Plate	70	1.21
	2	Pork - Bacon,back Peameal	49	4.65
	3	Lettuce - Romaine, Heart	38	3.35
	6	Petit Baguette	14	2.39
*	NULL	NULL	NULL	NULL

## BETWEEN Metin Değerleri Örneği

Aşağıdaki SQL deyimi, *customers* tablosunda *Antonio* ve *Christina* arasında *name* içeren tüm müşterileri seçer:

```
SELECT *
FROM customers
WHERE name BETWEEN 'Antonio' AND 'Christina'
ORDER BY name
```

	customer_id	name	surname	address	city	postalCode	country
▶	3	Antonio	Moreno	Mataderos 2312	México D.F.	05023	Mexico
	5	Christina	Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## NOT BETWEEN Metin Değerleri Örneği

Aşağıdaki SQL deyimi, *customers* tablosunda *Antonio* ve *Christina* arasında olmayan *name* sahip tüm müşterileri seçer:

```
SELECT *
FROM customers
WHERE name NOT BETWEEN 'Antonio' AND 'Christina'
ORDER BY name
```

	customer_id	name	surname	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	4	Thomas	Hardy	120 Hanover Sq.	London	WA1 1DP	UK
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## BETWEEN Tarih Örneği

Aşağıdaki SQL deyimi, *customers* tablosunda “1980-10-02” ve “1990-12-12” arasında *birth\_date* sahip tüm müşterileri seçer:

```
SELECT *  
FROM customers  
WHERE birth_date BETWEEN '1980-10-02' AND '1990-12-12'
```

	customer_id	name	surname	birth_date	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	1985-10-02	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	1990-12-12	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	1990-05-15	Mataderos 2312	México D.F.	05023	Mexico
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Aliases (AS)

Bir tabloya ve tablodaki bir sütuna geçici bir ad vermek için kullanılır. **AS**  
Genellikle sütun adlarını daha okunabilir hale getirmek için kullanılır. **AS**

### AS Sütun Sözdizimi;

```
SELECT sütun_adi AS takma_ad  
FROM tablo_adi
```

### AS Tablo Sözdizimi;

```
SELECT sütun_adi  
FROM tablo_adi AS takma_ad
```

## AS Sütun Örnekleri

Aşağıdaki SQL deyimi, biri *customer\_id* sütunu ve diğeri *name* sütunu için olmak üzere iki takma ad oluşturur:

```
SELECT customer_id AS ID, name AS customer  
FROM customers
```

	customer_id	name
▶	1	Alfreds
	2	Ana
	3	Antonio
	4	Thomas
	5	Christina
*	NULL	NULL

	ID	customer
▶	1	Alfreds
	2	Ana
	3	Antonio
	4	Thomas
	5	Christina

**Not :** **AS** eğer boşluk içeriyorsa tek tırnak veya çift tırnak işaretleri kullanılması gereklidir.

Aşağıdaki SQL deyimi, biri **name** sütunu ve diğeri **birth\_date** sütunu için olmak üzere iki diğer ad oluşturur:

```
SELECT name AS customer, birth_date AS 'birth date'
FROM customers
```

name	birth_date	customer	birth date
Alfreds	1985-10-02	Alfreds	1985-10-02
Ana	1990-12-12	Ana	1990-12-12
Antonio	1990-05-15	Antonio	1990-05-15
Thomas	1998-02-08	Thomas	1998-02-08
Christina	2000-11-20	Christina	2000-11-20

Aşağıdaki SQL deyimi, dört sütunu (**address**, **city**, **postalCode** ve **country**) birleştiren "**Address**" adlı bir takma ad oluşturur:

```
SELECT CONCAT_WS(' ', address, city, postalCode, country) AS Address
FROM customers
```

Address
Obere Str 57, Berlin, 12209, Germany
Avda. de la Constitución 2222, México D.F., 05021, Mexico
Mataderos 2312, México D.F., 05023, Mexico
120 Hanover Sq., London, WA1 1DP, UK
Berguvsvägen 8, Luleå, S-958 22, Sweden

## AS Tablo Örnekleri

Aşağıdaki SQL deyimi, **CustomerID** ile müşteriden gelen tüm siparişleri seçer. "**Müşteriler**" ve "**Siparişler**" tablolarını kullanıyoruz ve onlara sırasıyla "**c**" ve "**o**" tablo takma adlarını veriyoruz (Burada SQL'i kısaltmak için takma adlar kullanıyoruz):

```
SELECT o.OrderID, o.OrderDate, c.CustomerName
FROM Customers AS c, Orders AS o
WHERE c.CustomerName = 'Around the Horn' AND c.customerID = o.customerID
```

OrderID	OrderDate	CustomerName
10355	1996-11-15	Around the Horn
10383	1996-12-16	Around the Horn
10453	1997-02-21	Around the Horn
10558	1997-06-04	Around the Horn

**AS** şunlar olduğunda kullanılması yararlı olabilir:

- Sorguda birden fazla tablo varsa
- Sorguda işlevler kullanılıyorsa
- Sütun adları büyük veya çok okunabilir değilse
- İki veya daha fazla sütun bir araya getirilirse

## JOIN TABLES (Birleştirme Tabloları)

Aralarındaki ilgili bir sütuna bağlı olarak iki veya daha fazla tablodaki sütunları birleştirmek için kullanılır. **JOIN**

*Orders* tablosundan bir seçime bakalım:

	order_id	customer_id	order_date
▶	10308	2	1996-09-18
	10309	37	1996-09-19
	10310	77	1996-09-20
•	NULL	NULL	NULL

Ardından *Customers* tablosundan bir seçime bakalım:

	customer_id	customer_name	surname	birth_date	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	1985-10-02	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	1990-12-12	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	1990-05-15	Mataderos 2312	México D.F.	05023	Mexico
	4	Thomas	Hardy	1998-02-08	120 Hanover Sq.	London	WA1 1DP	UK
	5	Christina	Berglund	2000-11-20	Berguvsvägen 8	Luleå	S-958 22	Sweden
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

*Orders* tablosundaki “customer\_id” sütununun *Customers* tablosundaki “customer\_id” sütununa atıfta bulunduğuna dikkat edin. Yukarıdaki iki tablo arasındaki ilişki “customer\_id” sütunudur.

Daha sonra, her iki tabloda eşleşen değerleri seçen aşağıdaki SQL deyimini içeren bir birleştirme işlemi uygulayacağız: **INNER JOIN**

```
SELECT orders.order_id, customers.name, orders.order_date
FROM orders
INNER JOIN customers ON orders.customer_id = customers.customer_id
```

	order_id	name	order_date
▶	10308	Ana	1996-09-18
	10309	Antonio	1996-09-19
	10310	Thomas	1996-09-20

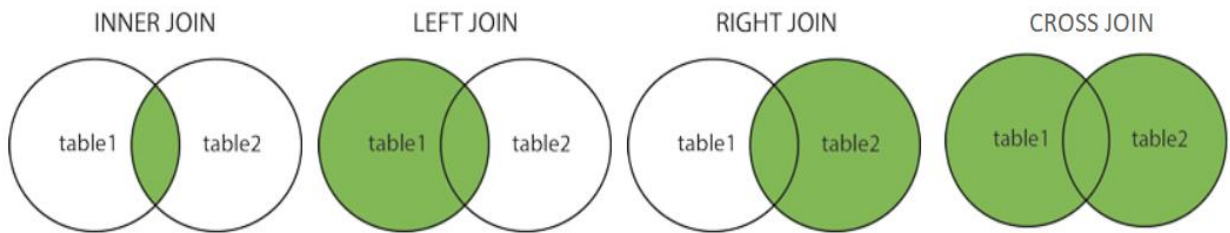
## MySQL’de Desteklenen Birleşim Türleri

**INNER JOIN:** Her iki tabloda da eşleşen değerlere sahip kayıtlar döndürür.

**LEFT JOIN:** Sol tablodaki tüm kayıtları sağ tablodaki eşleşen kayıtlara döndürür.

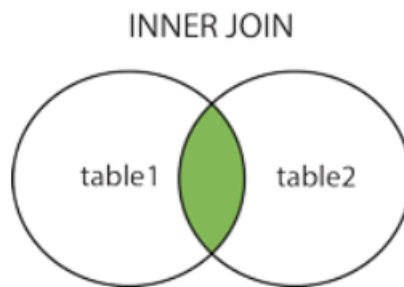
**RIGHT JOIN:** Sağ tablodaki tüm kayıtları sol tablodaki eşleşen kayıtlara döndürür.

**CROSS JOIN:** Her iki tablodaki tüm kayıtları döndürür.



## INNER JOIN

Her iki tabloda da eşleşen değerlere sahip kayıtlar seçer: **INNER JOIN**



## INNER JOIN Sözdizimi;

```
SELECT sütun_adi  
FROM tablo1  
INNER JOIN tablo2 ON tablo1.sütun_adi = tablo2.sütun_adi
```

*Orders* tablosundan bir seçime bakalım:

	order_id	customer_id	order_date
▶	10308	2	1996-09-18
	10309	37	1996-09-19
	10310	77	1996-09-20
•	NULL	NULL	NULL

Ardından **Customers** tablosundan bir seçime bakalım:

	customer_id	customer_name	surname	birth_date	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	1985-10-02	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	1990-12-12	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	1990-05-15	Mataderos 2312	México D.F.	05023	Mexico
	4	Thomas	Hardy	1998-02-08	120 Hanover Sq.	London	WA1 1DP	UK
	5	Christina	Berglund	2000-11-20	Berguvsvägen 8	Luleå	S-958 22	Sweden
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Orders** tablosundaki “customer\_id” sütununun **Customers** tablosundaki “customer\_id” sütununa atıfta bulunduğuna dikkat edin. Yukarıdaki iki tablo arasındaki ilişki “customer\_id” sütunudur.

Daha sonra, her iki tabloda eşleşen değerleri seçen aşağıdaki SQL deyimini içeren bir birleştirme işlemi uygulayacağız: **INNER JOIN**

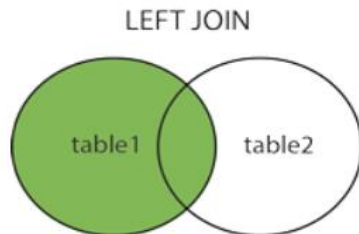
```
SELECT orders.order_id, customers.name
FROM orders
INNER JOIN customers ON orders.customer_id = customers.customer_id
```

	order_id	name
▶	10308	Ana
	10309	Antonio
	10310	Thomas

**Not : INNER JOIN**, sütunlar arasında bir eşleşme olduğu sürece her iki tablodaki tüm satırları seçer. “Orders” tablosundan “Customers” tablosu içinde eşleşme olmayan kayıtlar varsa, bu kayıtlar gösterilmez!

## LEFT JOIN

Sol tablodaki tablo1’in tüm kayıtlar ve sağ tablodaki tablo2’nin eşleşen kayıtlar varsa döndürür: **LEFT JOIN**



## LEFT JOIN Sözdizimi;

```
SELECT sütun_adi
FROM tablo1
LEFT JOIN tablo2 ON tablo1.sütun_adi = tablo2.sütun_adi
```

## LEFT JOIN Örneği

Aşağıdaki SQL deyimi, tüm müşterileri ve sahip olacağı değerleri seçer:

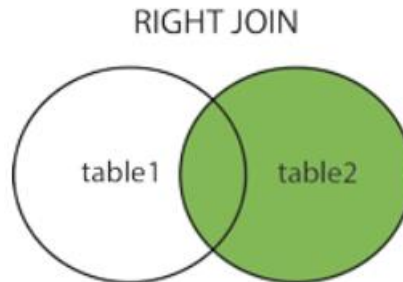
```
SELECT customers.name, orders.order_id
FROM customers
LEFT JOIN orders ON customers.customer_id = orders.customer_id
ORDER BY customers.name
```

	name	order_id
▶	Alfreds	NULL
	Ana	10308
	Antonio	10309
	Christina	NULL
	Thomas	10310

Not : **LEFT JOIN**, sağ tabloda eşleşme olmasa bile sol tablodaki “customers” tablosu “orders” tablosuna ait tüm kayıtları döndürür.

## RIGHT JOIN

Sağ tablodaki tablo2’in tüm kayıtlar ve sol tablodaki tablo1’in eşleşen kayıtlar varsa döndürür: **RIGHT JOIN**



## RIGHT JOIN Sözdizimi;

```
SELECT sütun_adi
FROM tablo1
RIGHT JOIN tablo2 ON tablo1.sütun_adi = tablo2.sütun_adi
```

*Orders* tablosundan bir seçime bakalım:

	order_id	customer_id	employee_id	order_date	shipper_id
	10308	2	5	1996-09-18	3
	10309	3	2	1996-09-19	2
	10310	4	4	1996-09-20	1
	10311	1	3	1996-09-21	4
▶	10312	5	1	1996-09-22	5
*	NULL	NULL	NULL	NULL	NULL



Ardından **Employees** tablosundan bir seçime bakalım:

	employee_id	surname	name	birth_date	photo
▶	1	Davolio	Nancy	1968-8-12	EmpID1.pic
	2	Fuller	Andrew	1952-2-19	EmpID2.pic
	3	Leverling	Janet	196-8-30	EmpID3.pic
*	NULL	NULL	NULL	NULL	NULL

## RIGHT JOIN Örneği

Aşağıdaki SQL deyimi, tüm “**employees**” tablosundaki ve “**orders**” tablosunda sahip olacağı değerleri seçer:

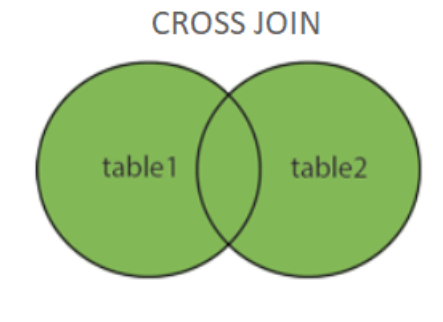
```
SELECT orders.order_id, employees.surname, employees.name
FROM orders
RIGHT JOIN employees ON orders.employee_id = employees.employee_id
ORDER BY orders.order_id
```

	order_id	surname	name
▶	10309	Fuller	Andrew
	10311	Leverling	Janet
	10312	Davolio	Nancy

**Not :** **RIGHT JOIN**, sol tabloda eşleşme olmasa bile sağ tablodaki “**employees**” tablosu “**orders**” tablosuna ait tüm kayıtları döndürür.

## CROSS JOIN

Her iki tabloda, tablo1 ve tablo2’deki tüm kayıtları döndürür: **CROSS JOIN**



## CROSS JOIN Sözdizimi;

```
SELECT sütun_adi
FROM tablo1
CROSS JOIN tablo2
```

**Not :** **CROSS JOIN**, çok büyük sonuç kümeleri döndürebilir.



**Orders** tablosundan bir seçime bakalım:

	order_id	customer_id	order_date
▶	10308	2	1996-09-18
	10309	37	1996-09-19
	10310	77	1996-09-20
•	NULL	NULL	NULL

Ardından **Customers** tablosundan bir seçime bakalım:

	customer_id	customer_name	surname	birth_date	address	city	postalCode	country
▶	1	Alfreds	Futterkiste	1985-10-02	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	1990-12-12	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	1990-05-15	Mataderos 2312	México D.F.	05023	Mexico
	4	Thomas	Hardy	1998-02-08	120 Hanover Sq.	London	WA1 1DP	UK
	5	Christina	Berglund	2000-11-20	Berguvsvägen 8	Luleå	S-958 22	Sweden
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## CROSS JOIN Örneği

Aşağıdaki SQL deyimi, tüm “**customers**” tablosunu ve “**orders**” tablosundaki kayıtları seçer:

```
SELECT customers.name, orders.order_id
FROM customers
CROSS JOIN orders
WHERE customers.customer_id = orders.customer_id
```

	name	order_id
▶	Alfreds	10311
	Ana	10308
	Antonio	10309
	Thomas	10310
	Christina	10312

**Not : CROSS JOIN**, diğer tablo eşleşse de eşleşmese de her iki tablodan eşleşen tüm kayıtları döndürür. Bu nedenle “**customers**” tablosunda, “**orders**” tablosunda eşleşmesi olmayan satırlar varsa veya “**orders**” tablosunda, “**customers**” tablosunda eşleşmesi olmayan varsa, bu satırlar da listelenir.

## SELF JOIN

Kendi kendine birleştirme düzenli birleşimdir, ancak tablo kendisiyle birleştirilir: **SELF JOIN**

### SELF JOIN Sözdizimi;

```
SELECT sütun_adi  
FROM tablo1 T1, tablo2 T2  
WHERE koşul
```

**Not :** *T1 ve T2 aynı tablo için farklı tablo adlarıdır.*

Aşağıda “customers” tablosundan bir seçim yapılmıştır:

	customer_id	customer_name	surname	birth_date	address	city	postalCode	country
►	1	Alfreds	Futterkiste	1985-10-02	Obere Str 57	Berlin	12209	Germany
	2	Ana	Trujillo	1990-12-12	Avda. de la Constitución 2222	México D.F.	05021	Mexico
	3	Antonio	Moreno	1990-05-15	Mataderos 2312	México D.F.	05023	Mexico
	4	Thomas	Hardy	1998-02-08	120 Hanover Sq.	London	WA1 1DP	UK
	5	Christina	Berglund	2000-11-20	Berguvsvägen 8	Luleå	S-958 22	Sweden
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### SELF JOIN Örneği

```
SELECT A.name AS customerName1, B.name AS customerName2, A.city  
FROM customers A, customers B  
WHERE A.customer_id <> B.customer_id AND A.city = B.city  
ORDER BY A.city
```

Aşağıdaki SQL deyimi, aynı şehirden gelen müşterilerle eşleşir:

	customerName1	customerName2	city
►	Antonio	Ana	México D.F.
	Ana	Antonio	México D.F.

## UNION Operator

İki veya daha fazla deyimin sonuç kümesini birleştirmek için kullanılır: **UNION SELECT**

- İçindeki her deyim aynı sayıda sütuna sahip olmalıdır.
- Sütunlar benzer veri yapılarına sahip olmalıdır.
- Her deyimdeki sütunlar da aynı sırada olmalıdır.

## UNION Sözdizimi;

```
SELECT sütun_adi FROM tablo1
UNION
SELECT sütun_adi FROM tablo2
```

## UNION ALL Sözdizimi;

İşleç Varsayılan olarak yalnızca farklı değerler seçer. Yinelenen değerlere izin vermek için: **UNION ALL**

```
SELECT sütun_adi FROM tablo1
UNION ALL
SELECT sütun_adi FROM tablo2
```

**Not :** Sonuç kümesindeki sütun adları genellikle ilk deyimdeki sütun adlarına eşittir.

Aşağıda "**customers**" tablosundan bir seçim verilmiştir:

	customer_id	name	surname	adress	city	postalCode	country
▶	1	Alfreds	Futterkiste	Obere Str...	Berlin	12209	Germany
	2	Ana	Trujillo	Avda. de l...	México D.F.	05021	Mexico
	3	Antonio	Moreno	Matadero...	México D.F.	05023	Mexico
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Aşağıda "**supperliers**" tablosundan bir seçim verilmiştir:

	supperlier_id	name	surname	address	city	postalCode	country
	1	Charlotte	Cooper	49 Gilb...	Lon...	EC1 4SD	UK
	2	Shelley	Burke	P.O. B...	Ne...	70117	USA
▶	3	Regina	Murphy	707 Ox...	Ann...	48104	USA
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## UNION Örneği

Aşağıdaki SQL deyimini, hem "**customers**" hem de "**supperliers**" tablosundan şehirleri yalnızca farklı değerlere döndürür:

```
SELECT city FROM customers
UNION
SELECT city FROM supperliers
ORDER BY city
```

	city
▶	Ann Arbor
	Berlin
	London
	México D.F.
	New Orleans

**Not :** Bazı “*customers*” ve “*supperliers*” tablosundan kayıtlar aynı şehre sahipse, yalnızca farklı değerler seçtiğinden, her şehir yalnızca bir kez listelenir. Yinelenen değerleri seçmek için ise **UNION ALL** deyimi kullanılır.

### UNION ALL Örneği

Aşağıdaki SQL deyimi, hem “*customers*” hem de “*supperliers*” tablosundan şehirleri yinelenen değerlerde döndürür:

```
SELECT city FROM customers
UNION ALL
SELECT city FROM supperliers
ORDER BY city
```

city
Ann Arbor
Berlin
London
México D.F.
México D.F.
New Orleans

### UNION With WHERE

Aşağıdaki SQL deyimi, hem “*customers*” hem de “*supperliers*” tablosundan Alman şehirlerini yalnızca farklı değerlere döndürür:

```
SELECT city, country FROM customers
WHERE city = 'Germany'
UNION
SELECT city, country FROM supperliers
WHERE city = 'Germany'
ORDER BY city
```

City	Country
Aachen	Germany
Berlin	Germany
Brandenburg	Germany

Aşağıdaki SQL deyimi, hem “*customers*” hem de “*supperliers*” tablosundan Alman şehirlerini yinelenen değerlerde döndürür:

```
SELECT city, country FROM customers
WHERE city = 'Germany'
UNION ALL
SELECT city, country FROM supperliers
WHERE city = 'Germany'
ORDER BY city
```

City	Country
Aachen	Germany
Berlin	Germany
Berlin	Germany
Brandenburg	Germany

Aşağıdaki SQL deyimi, hem *"customers"* hem de *"suppliers"* tablosu listelenir:

```
SELECT 'customer' AS Type , name, city, country
FROM customers
UNION
SELECT 'supplier', name, city, country
FROM suppliers
```

	Type	name	city	country
▶	customer	Alfreds	Berlin	Germany
	customer	Ana	México D.F.	Mexico
	customer	Antonio	México D.F.	Mexico
	supplier	Charlotte	London	UK
	supplier	Shelley	New Orleans	USA
	supplier	Regina	Ann Arbor	Mexico

**Not:** Yukarıdaki **AS** türüne dikkat edin! SQL takma adları, bir tabloya veya sütuna geçici bir ad vermek için kullanır. **AS** sadece sorgu süresince bulunur. Bu nedenle buradaki ilgili tablonun *"customers"* mi yoksa *"suppliers"* mi olduğunu listeleyen **"Tür"** adlı geçici bir sütun oluşturduk.

## GROUP BY Statement (Deyimi)

Aynı değerlere sahip satırları **"her ülkedeki kişi sayısını bul"** gibi özet satırlara gruplar: **GROUP BY**

## GROUP BY Sözdizimi;

```
SELECT sütun_adi
FROM tablo_adi
WHERE koşul
GROUP BY sütun_adi
ORDER BY sütun_adi
```

## GROUP BY Örnekleri

Aşağıdaki SQL deyimi, her ülkedeki müşteri sayısını verir:

```
SELECT COUNT(customer_id), country
FROM customers
GROUP BY country
```

	COUNT(customer_id)	country
▶	1	Germany
	2	Mexico

Aşağıdaki SQL deyimi, her ülkedeki yüksek ila düşük arasında sıralanmış müşteri sayısını listeler:

```
SELECT COUNT(customer_id), country
FROM customers
GROUP BY country
ORDER BY COUNT(customer_id) DESC
```

	COUNT(customer_id)	country
▶	2	Mexico
	1	Germany

## HAVING Clause

Yapısı temeliyle **WHERE** ile aynı işlevi görmektedir. **GROUP BY** ile kullanılır. **WHERE** ifadesi ile belirtilen kriter, **GROUP BY** uygulanmadan önce geçerli olurken, **HAVING** ifadesi ile belirtilen kriter ise **GROUP BY** uygulandıktan sonra ortaya çıkan verileri filtrelemek için kullanılır: **HAVING**

## HAVING Sözdizimi;

```
SELECT sütun_adi
FROM tablo_adi
WHERE koşul
GROUP BY sütun_adi
HAVING koşul
ORDER BY sütun_adi
```

## HAVING Örnekleri

Aşağıdaki SQL deyiminde her ülkedeki müşteri sayısı listeler. Yalnızca 1'ten fazla müşterisi olan ülkeleri dahil edin:

```
SELECT COUNT(customer_id), country
FROM customers
GROUP BY country
HAVING COUNT(customer_id) > 1
```

	COUNT(customer_id)	country
▶	2	Mexico

Aşağıdaki SQL deyimi, her ülkedeki yüksekten düşüğe göre sıralanmış müşteri sayısını listeler (Yalnızca 1'den fazla müşterisi olan ülkeleri içerir):

```
SELECT COUNT(customer_id), country
FROM customers
GROUP BY country
HAVING COUNT(customer_id) > 1
ORDER BY customer_id DESC
```

	COUNT(customer_id)	country
▶	2	Mexico
	2	Germany

## EXISTS Operator

İşleç, bir alt sorguda herhangi bir kaydın varlığını test etmek için kullanılır: **EXISTS**  
Alt sorgu bir veya daha fazla kayıt döndürürse, işleç **TRUE** döndürür: **EXISTS**

## EXISTS Sözdizimi;

```
SELECT sütun_adi
FROM tablo_adi
WHERE EXISTS
(SELECT sütun_adi FROM tablo_adi WHERE koşul)
```

Aşağıda “*products*” tablosundan bir seçim:

	product_id	name	supplier_id	category_id	unit	price
▶	1	Chais	1	1	10 boxes x 20 bags	18
	2	Chang	1	1	24 - 12 oz bottles	19
	3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
	5	Chef Anton's Gumbo Mix	2	2	36 boxes	23
*	NULL	NULL	NULL	NULL	NULL	NULL

Ve “*supperliers*” tablosundan bir seçim:

	supperlier_id	name	surname	address	city	postalCode	country
▶	1	Charlotte	Cooper	49 Gilbert St.	London	EC1 4SD	UK
	2	Shelley	Burke	P.O. Box 78934	New Orleans	70117	USA
	3	Regina	Murphy	707 Oxford Rd.	Ann Arbor	48104	Mexico
	4	Yoshi	Nagase	9-8 Sekimai Musashino-shi	Tokyo	100	Japan
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## EXISTS Örnekleri

Aşağıdaki SQL deyimi, DOĞRU döndürür ve ürün fiyatı 20'den az olan tedarikçileri listeler:

```
SELECT name AS SupplierName
FROM supperliers
WHERE EXISTS (SELECT name FROM products WHERE
products.supperlier_id = supperliers.supperlier_id AND price < 20)
```

### SupplierName

Exotic Liquid

New Orleans Cajun Delights

Tokyo Traders

Mayumi's

Aşağıdaki SQL deyimi, DOĞRU döndürür ve tedarikçileri 22'ye eşit olan ürün fiyatı listeler:

```
SELECT name AS SupplierName
FROM supperliers
WHERE EXISTS (SELECT name FROM products WHERE
products.supperlier_id = supperliers.supperlier_id AND price = 22)
```

### SupplierName

New Orleans Cajun Delights

## ANY and ALL Operators

İşleçler, tek bir sütun değeri ile diğer değerler arasında bir karşılaştırma gerçekleştirmemizi sağlar: **ANY, ALL**



## ANY Operator

Bir boole değeri döndürür.

Alt sorgu değerlerinden herhangi bir koşulu sağlıyorsa **TRUE** döndürür: **ANY**  
**ANY** işlemi, aralıktaki değerlerden herhangi biri için doğruysa koşulun doğru olacağı anlamına gelir.

### ANY Sözdizimi;

```
SELECT sütun_adi  
FROM tablo_adi  
WHERE sütun_adi ANY  
(SELECT sütun_adi FROM tablo_adi WHERE koşul)
```

**Not:** İşleç standart bir karşılaştırma işleci olmalıdır. (=, <>, !=, < veya >).

## ALL Operator

Bir boole değeri döndürür.

Alt sorgu değerlerinden tümü koşulu sağlıyorsa **TRUE** döndürür: **ALL**  
**ALL** işlemi, koşulun yalnızca işlem aralığındaki tüm değerler için doğruysa koşulun doğru olacağı anlamına gelir.

**SELECT, WHERE, HAVING** deyimleriyle kullanılır.

### ALL Sözdizimi;

```
SELECT ALL sütun_adi  
FROM tablo_adi  
WHERE koşul
```

### ALL WHERE or HAVING Sözdizimi;

```
SELECT sütun_adi  
FROM tablo_adi  
WHERE sütun_adi ALL  
(SELECT sütun_adi FROM tablo_adi WHERE koşul)
```

**Not:** İşleç standart bir karşılaştırma işleci olmalıdır. (=, <>, !=, < veya >).

Aşağıda **“products”** tablosundan bir seçim:

	product_id	product_name	supplier_id	category_id	unit	price
▶	1	Chais	1	1	10 boxes x 20 bags	18
	2	Chang	1	1	24 - 12 oz bottles	19
	3	Aniseed Syrup	1	2	12 - 550 ml bottles	10
	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22
	5	Chef Anton's Gumbo Mix	2	2	36 boxes	23
	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25
	7	Uncle Bob's Organic Dried Pears	3	2	12 - 1 lb pkgs.	30
	8	Northwoods Cranberry Sauce	3	7	12 - 12 oz jars	40
	9	Mishi Kobe Niku	3	2	18 - 500 g pkgs.	97
*	NULL	NULL	NULL	NULL	NULL	NULL

Ve **“orderdetails”** tablosundan bir seçim:

	orderdetails_id	order_id	product_id	quantity
▶	1	10248	5	12
	2	10248	6	10
	3	10248	7	5
	4	10249	9	9
	5	10249	8	40
	6	10250	1	10
	7	10250	2	35
	8	10250	3	15
	9	10251	4	6
	10	10251	5	15
*	NULL	NULL	NULL	NULL

## ANY Örnekleri

Aşağıdaki SQL deyimi, **“orderdetails”** tablosunda Herhangi bir kayıt bulursa **“products\_name”** sütununu listeler. Miktar 10'a eşittir (Miktar sütununda 10 olan bazı değerleri DOĞRU döndürür):

```
SELECT products_name
FROM products
WHERE products_id = ANY
(SELECT products_id FROM orderdetails WHERE quantity = 10)
```

	product_name
▶	Grandma's Boysenberry Spread
	Chais

Aşağıdaki SQL deyimi, **“orderdetails”** tablosunda herhangi bir kayıt bulursa **“products\_name”** sütununu listeler. 10'dan büyük Miktar vardır. (Miktar sütunu 99'dan büyük bazı değerlere sahip olduğundan bunu DOĞRU döndürür):

```
SELECT products_name
FROM products
WHERE products_id = ANY
(SELECT products_id FROM orderdetails WHERE quantity > 10)
```

	product_name
▶	Chef Anton's Gumbo Mix
	Northwoods Cranberry Sauce
	Chang
	Aniseed Syrup

Aşağıdaki SQL deyimi, “*orderdetails*” tablosunda 1000'den büyük Miktarla sahip herhangi bir kayıt bulursa “*products\_name*” sütununu listeler. (Miktar sütununda 1000'den büyük değer olmadığından bu FALSE döndürür):

```
SELECT product_name
FROM products
WHERE product_id = ANY
(SELECT product_id FROM orderdetails WHERE quantity > 1000)
```

## ALL Örnekleri

Aşağıdaki SQL deyimi, tüm “*products*” adlarını listeler:

```
SELECT ALL product_name
FROM products
WHERE TRUE
```

	product_name
--	--------------

Aşağıdaki SQL deyimi, “*orderdetails*” tablosundaki tüm kayıtların miktarı 8'a eşitse, “*products\_name*” sütununu listeler. Miktar sütunu birçok farklı değere sahip olduğundan (yalnızca 8 değeri değil):

```
SELECT product_name
FROM products
WHERE product_id = ALL
(SELECT product_id FROM orderdetails WHERE quantity = 8)
```

	product_name
►	Chais
	Chang
	Aniseed Syrup
	Chef Anton's Cajun Seasoning
	Chef Anton's Gumbo Mix
	Grandma's Boysenberry Spread
	Uncle Bob's Organic Dried Pears
	Northwoods Cranberry Sauce
	Mishi Kobe Niku

## INSERT INTO SELECT Statement (Deyimi)

Verileri bir tablodan kopyalar ve başka bir tabloya ekler: **INSERT INTO SELECT**

**Not:** Hedef tablodaki varolan kayıtlar eklenmez.

### INSERT INTO SELECT Sözdizimi;

```
INSERT INTO tablo2
SELECT * FROM tablo1
WHERE koşul
```

Bir tablodan başka bir tabloya yalnızca bazı sütunları kopyalayın:

```
INSERT INTO tablo2 (sütun1, sütun2, ...)
SELECT sütun1, sütun2, ...
FROM tablo1
WHERE koşul
```

Aşağıdaki SQL deyimi, "**supperliers**" tablosunu "**customers**" olarak kopyalar (verilerle doldurulmayan sütunlar NULL içerir):

```
INSERT INTO customers (name, city, country)
SELECT name, city, country FROM supperliers
```

Aşağıdaki SQL deyimi, yalnızca Alman tedarikçileri "**customers**" e kopyalar:

```
INSERT INTO customers (name, city, country)
SELECT name, city, country FROM supperliers
WHERE country = 'Germany'
```

## CASE Statement (Deyimi)

Deyim koşullardan geçer ve ilk koşul karşılandığında bir değer döndürür (if-then-else deyimi gibi). Bu nedenle, bir koşul doğru olduğunda, okumayı durdurur ve sonucu döndürür. Hiçbir koşul doğru değilse, yan tümcedeki değeri döndürür: **CASE ELSE**

Hiçbir parça yoksa ve koşul doğru değilse, NULL döndürür: **ELSE**

## CASE Sözdizimi;

```
CASE customers (name, city, country)
  WHEN koşul THEN sonuç1
  WHEN koşul2 THEN sonuç2
  WHEN koşulN THEN sonuçN
  ELSE sonuç
END
```

Aşağıda, "orderdetails" tablosundan bir seçim verilmiştir:

	orderdetails_id	order_id	product_id	quantity
▶	1	10248	5	12
	2	10248	6	10
	3	10248	7	5
	4	10249	9	9
	5	10249	8	40
	6	10250	1	10
	7	10250	2	35
	8	10250	3	15
	9	10251	4	6
	10	10251	5	15
•	NULL	NULL	NULL	NULL

Aşağıdaki SQL deyimi, koşullardan geçer ve ilk koşul karşılandığında bir değer döndürür:

```
SELECT order_id, quantity,
CASE
  WHEN quantity > 30 THEN 'miktar 30'dan büyük'
  WHEN quantity = 30 THEN 'miktar 30'a eşit'
  ELSE 'miktar 30'u altında'
END AS 'Miktar metni'
FROM orderdetails
```

	order_id	quantity	Miktar metni
▶	10248	12	miktar 30'u altında
	10248	10	miktar 30'u altında
	10248	5	miktar 30'u altında
	10249	9	miktar 30'u altında
	10249	40	miktar 30'dan büyük
	10250	10	miktar 30'u altında
	10250	35	miktar 30'dan büyük
	10250	15	miktar 30'u altında
	10251	6	miktar 30'u altında
	10251	15	miktar 30'u altında

## COMMENTS (Yorum Satırları)

Açıklamalar, SQL deyimlerinin bölümlerini açıklamak veya SQL deyimlerinin yürütülmesini önlemek için kullanılır.

### Tek Satırlı Açıklamalar

Tek satırlı yorumlar ile başlar: **--**

Satırın sonu ile arasındaki herhangi bir metin yürütülmez.

Aşağıdaki örnek açıklama satırı olarak, tek satırlık bir açıklama kullanır:

```
-- SELECT ALL  
SELECT * FROM customers
```

Aşağıdaki örnek, bir satırın sonunu yok saymak için tek satırlı bir açıklama kullanır:

```
SELECT * FROM customers  
-- WHERE city = 'Berlin'
```

### Çok Satırlı Açıklamalar

Çok satırlı yorumlar ile başlar ve biter: **/\*\*/**

**/\*** ve **\*/** arasındaki herhangi bir metin yürütülmez.

Aşağıdaki örnek, açıklama olarak çok satırlı bir açıklama kullanır:

```
/*Müşteri tablosundaki tüm kayıtları seçin*\  
SELECT * FROM customers
```

Aşağıdaki örnek, birçok deyim yok saymak için çok satırlı bir açıklama kullanır:

```
/*SELECT * FROM suppliers  
SELECT * FROM employees  
SELECT * FROM orders*\  
SELECT * FROM customers
```

Aşağıdaki örnek, deyim bir bölümünü yok saymak için bir açıklama kullanır:

```
SELECT * FROM customers WHERE  
(name LIKE 'L%' /*OR name LIKE  
'R%' OR name LIKE 'S%' OR name  
LIKE 'T%']* OR name LIKE 'W%'  
AND country = 'USA'  
ORDER BY name
```

### Arithmetich Operators (Aritmetik Operatörler)

Operatörler	Tanım
+	Toplama işlemi yapar.
-	Çıkarma işlemi yapar.
*	Çarpma işlemi yapar.
/	Bölme işlemi yapar.
%	Mod alma işlemi yapar.

### Bitwise Operators (Bit Operatörler)

Operatörler	Tanım
&	Bitsel ve.
	Bitsel veya.
^	Bitsel özel veya.