

# SQL BOOTCAMP İKİNCİ HAFTA ÖDEVİ

## a. Tüm değerleri NULL olan kayıtların sayısı

```
SELECT COUNT(*) AS nullkayitlar  
  
FROM Invoice  
  
WHERE invoice_id IS NULL  
  
AND customer_id IS NULL  
  
AND invoice_date IS NULL  
  
AND billing_address IS NULL  
  
AND billing_city IS NULL  
  
AND billing_state IS NULL  
  
AND billing_country IS NULL  
  
AND billingpostal_code IS NULL  
  
AND total IS NULL; /* Row sayısı: 1 */
```

- COUNT(\*) ile şartı sağlayan satırları sayıyoruz.
- WHERE koşulunda **tüm sütunların NULL olup olmadığını** kontrol ediyoruz.
- Eğer bir satırdaki **tüm sütunlar NULL ise**, bu satır sayılır.

```
1 SELECT COUNT(*) AS nullkayitlar  
2 FROM Invoice  
3 WHERE invoice_id IS NULL  
4 AND customer_id IS NULL  
5 AND invoice_date IS NULL  
6 AND billing_address IS NULL  
7 AND billing_city IS NULL  
8 AND billing_state IS NULL  
9 AND billing_country IS NULL  
10 AND billingpostal_code IS NULL  
11 AND total IS NULL; /* Row sayısı: 1 */  
12  
13  
14  
15
```

Data Output Messages Notifications

	nullkayitlar bigint
1	0

## b. Total değer karşılaştırma

SELECT

invoice\_id,

customer\_id,

invoice\_date,

total AS eskitoplam,

total \* 2 AS yenitoplam

FROM Invoice

ORDER BY yenitoplam DESC;

- total AS **eskitoplam**: Eski Total değerini saklar.
- total \* 2 AS **yenitoplam**: Total değerinin iki katını alır.
- ORDER BY **yenitoplam** DESC: Yeni Total sütununa göre **büyükten küçüğe** sıralama yapar.

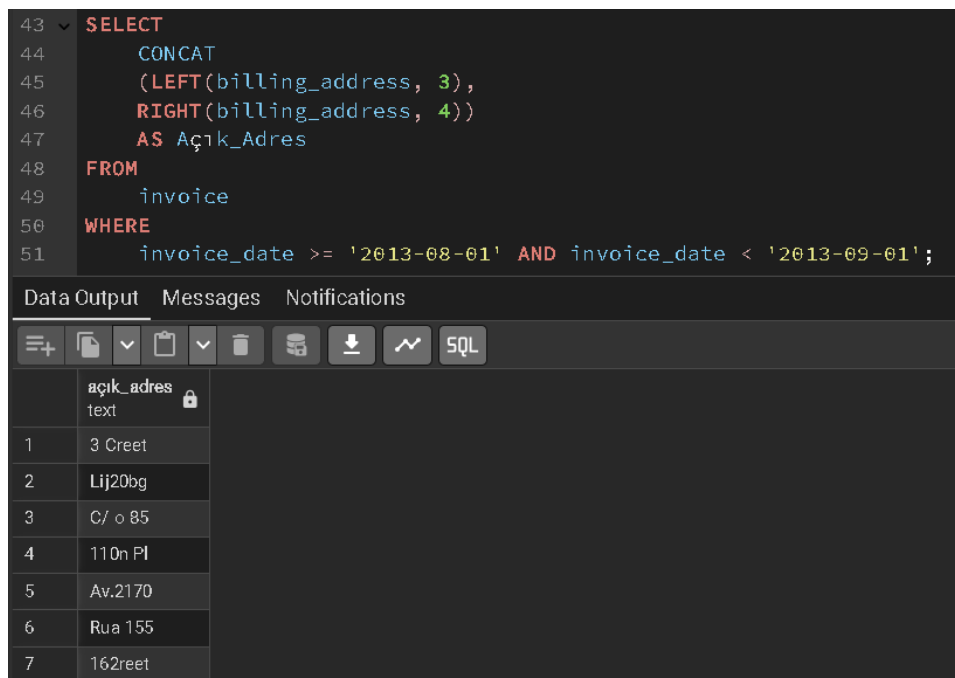
```
15
16 SELECT
17     invoice_id,
18
19     customer_id,
20
21     invoice_date,
22
23     total AS eskitoplam,
24
25     total * 2 AS yenitoplam
26
27 FROM Invoice
28 ORDER BY yenitoplam DESC;
29
```

	invoice_id [PK] integer	customer_id integer	invoice_date timestamp without time zone	eskitoplam numeric (10,2)	yenitoplam numeric
1	404	6	2013-11-13 00:00:00	25.86	51.72
2	299	26	2012-08-05 00:00:00	23.86	47.72
3	194	46	2011-04-28 00:00:00	21.86	43.72
4	96	45	2010-02-18 00:00:00	21.86	43.72
5	89	7	2010-01-18 00:00:00	18.86	37.72
6	201	25	2011-05-29 00:00:00	18.86	37.72
7	88	57	2010-01-13 00:00:00	17.91	35.82
8	313	43	2012-10-06 00:00:00	16.86	33.72
9	306	5	2012-09-05 00:00:00	16.86	33.72
Total rows: 412    Query complete 00:00:00.158					

### c. 2013 8.ay için Faturaları: Açık Adres Hesaplama

```
SELECT CONCAT(LEFT(billing_address, 3), RIGHT(billing_address, 4)) AS Açık_Adres  
FROM invoice WHERE invoice_date >= '2013-08-01' AND invoice_date < '2013-09-01';
```

- LEFT(billing\_address, 3) ve RIGHT(billing\_address, 4) ile adresin ilk 3 ve son 4 karakteri alınır.
- CONCAT ile bu iki kısım birleştirilir.
- invoice\_date >= '2013-08-01' AND invoice\_date < '2013-09-01':  
Bu koşul, 2013 yılının Ağustos ayındaki tüm faturaları alır.



The screenshot shows a SQL IDE interface. The top pane displays a SQL query. The bottom pane shows the results of the query in a table format. The table has two columns: 'açık\_adres' (text) and an unlabeled column. The results are as follows:

	açık_adres	
1	3 Creet	
2	Lij20bg	
3	C/ o 85	
4	110n Pl	
5	Av.2170	
6	Rua 155	
7	162reet	