**TRIM() FUNCTION:**

The TRIM() function removes the space character OR other specified characters from the start or end of a string.

By default, the TRIM() function removes leading and trailing spaces from a string.

Syntax:

**TRIM([*characters*FROM]*string*)**

Parameters:

***characters* FROM** Optional. Specific characters to remove.

***String***Required. The string to remove spaces or characters from

**RTRIM() FUNCTION:**

The RTRIM() function removes trailing spaces from a string.

Syntax:

**RTRIM(*string*)**

Parameters:

***String*** Required. The string to remove trailing spaces from

**LTRIM() FUNCTION:**

The LTRIM() function removes leading spaces from a string.

Syntax:

**LTRIM(*string*)**

Parameters:

***String*** Required. The string to remove trailing spaces from

**COUNT() FUNCTION:**

The COUNT() function returns the number of records returned by a select query.

**Note:** NULL values are not counted.

Syntax:

COUNT(expression)

Parameters:

expression Required. A field or a string value

Example1:

Return the number of products in the "Products" table:

SELECT COUNT(ProductID) AS NumberOfProducts FROM Products;

Example2:

(From “Sample Retail” Database)

-- How many customers have yahoo mail?

SELECT COUNT (\*)

FROM sale.customer

WHERE PATINDEX('%yahoo%', email) > 0;

OR;

SELECT COUNT(email)

FROM sale.customer

WHERE email LIKE '%yahoo.com';

**CHARINDEX() FUNCTION:**

Search for "t" in string "Customer", and return position:

SELECT CHARINDEX('t', 'Customer') AS MatchPosition;

**The CHARINDEX()** function:

searches for a **substring in a string,** and returns the **position.**

If the substring is not found, this function returns 0.

**Note:** This function performs a **case-insensitive** search.

Syntax:

CHARINDEX(substring, string, start)

Parameters:

substring Required. The substring to be searched

string Required. The string to be searched

start Optional. The position where the search will start (if you do not want to start at the beginning of string). The first position in string is 1

Example1:

SELECT CHARINDEX('OM', 'Customer') AS MatchPosition;

Example2:

(From “Sample Retail” Database)

--Write a query that returns the characters before the '@' character in the email column.

SELECT email, LEFT(email,CHARINDEX('@',email)-1) AS Chars  
FROM sale.customer

**COALESCE() FUNCTION:**

The COALESCE() function returns the first non-null value in a list.

Syntax:

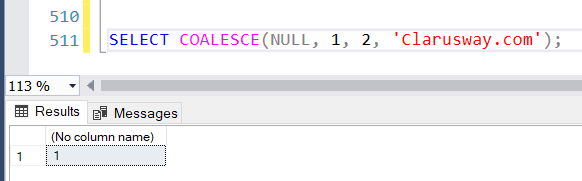
COALESCE(val1, val2, ...., val\_n)

Parameters:

***val1, val2, val\_n***Required. The values to test

Example1:

SELECT COALESCE(NULL, 1, 2, 'Clarusway.com');



**İlk non-null değeri döndürdü.**

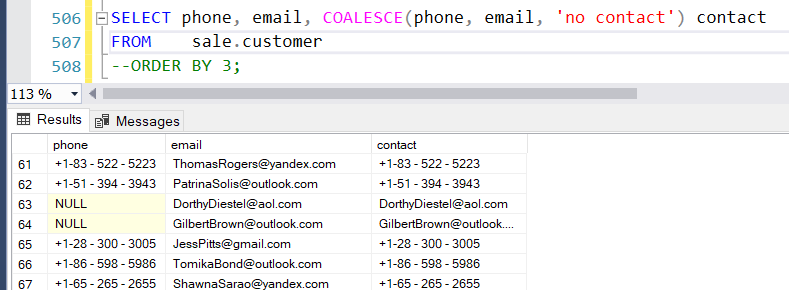
Example2:

(From “Sample Retail” Database)

---Add a new column to the customers table that contains the customers' contact information.  
--If the phone is not null, the phone information will be printed, if not, the email information will be printed.

SELECT phone, email, COALESCE(phone, email, 'no contact') contact

FROM sale.customer;



**JOINS**

**INNER JOIN**

The INNER JOIN keyword selects records that have matching values in both tables.

Syntax:

SELECT column\_name(s)  
FROM table1  
INNER JOIN table2ON table1.column\_name = table2.column\_name;

Example1:

SELECT Orders.OrderID, Customers.CustomerName  
FROM Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID;

**Example2:**

SELECT A.product\_name, A.category\_id,

B.category\_id, B.category\_name

FROM product.product A,

product.category B

WHERE A.category\_id = B.category\_id

ORDER BY

A.category\_id

**JOIN Three Tables:**

SELECT Orders.OrderID, Customers.CustomerName, Shippers.ShipperName  
FROM ((Orders  
INNER JOIN Customers ON Orders.CustomerID = Customers.CustomerID)  
INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID);