23. CONCAT & Split a String

In this article, we are going to learn about how we use the CONCAT() function to add values or columns together. In addition, we will also take a look at how to split a string using STRING_SPLIT() and SUBSTRING() functions. Examples in this article are available in both SQL Server and MySQL.

CONCAT() Function

CONCAT function() is used to concatenate all expressions together.

Syntax
SELECT CONCAT(Expression1, Expression2, Expression3, ...)

Table 1: CONCAT Function

We will get the below result.

Result:
Expression1Expression2Expression3 ...

Table 2. Concatenated Result

The expression can be a value or a column, and it will be cast to string before adding them together. In MySQL, the function returns NULL when one of the expressions is equal to NULL. However, in SQL Server, NULLs will be ignored, and remaining expressions will be added together.

Refer below tables for the following examples.

Employees Table:

Id	Code	DepartmentId	JobId	EmployeeCode
1	0001	01	01	NULL
2	0002	02	01	NULL
3	0003	01	02	NULL
4	0004	01	04	NULL
5	0005	03	01	NULL

6	0006	02	02	NULL

Table 3. Sample Data for Employees

Departments Table:

Id	Name
01	Production
02	HR
03	Marketing
04	IT
05	Accounting

Table 4. Sample Data for Departments

Jobs Table:

Id	Description
01	Management
02	Executive
03	Non-Executive
04	Training

Table 5. Sample Data for Jobs

Use SQL Server or MySQL to run the following query to create sample tables for given examples.

Sample Data Script:

```
CREATE TABLE Employees(
    Id INT NOT NULL,
    Code VARCHAR(4) NOT NULL,
    DepartmentId VARCHAR(2) NOT NULL,
    JobId VARCHAR(2) NOT NULL,
    EmployeeCode VARCHAR(10) NULL,
    PRIMARY KEY (Id)
);
```

```
CREATE TABLE Departments(
    Id VARCHAR(2) NOT NULL,
    Name VARCHAR(50) NOT NULL,
    PRIMARY KEY (Id)
);

CREATE TABLE Jobs(
    Id VARCHAR(2) NOT NULL,
    Description VARCHAR(100) NOT NULL,
    PRIMARY KEY (Id)
);

INSERT Employees(Id, Code, DepartmentId, JobId) VALUES (1, '0001', '01',
'01');
INSERT Employees(Id, Code, DepartmentId, JobId) VALUES (2, '0002', '02',
'01');
INSERT Employees(Id, Code, DepartmentId, JobId) VALUES (3, '0003', '01',
'02');
INSERT Employees(Id, Code, DepartmentId, JobId) VALUES (4, '0004', '01',
'04');
INSERT Employees(Id, Code, DepartmentId, JobId) VALUES (5, '0005', '03',
'01');
INSERT Employees(Id, Code, DepartmentId, JobId) VALUES (6, '0006', '02',
'02');

INSERT Departments(Id, Name) VALUES ('01', 'Production');
INSERT Departments(Id, Name) VALUES ('02', 'HR');
INSERT Departments(Id, Name) VALUES ('03', 'Marketing');
INSERT Departments(Id, Name) VALUES ('04', 'IT');
INSERT Departments(Id, Name) VALUES ('05', 'Accounting');

INSERT Jobs(Id, Description) VALUES ('01', 'Management');
INSERT Jobs(Id, Description) VALUES ('02', 'Executive');
INSERT Jobs(Id, Description) VALUES ('03', 'Non-Executive');
INSERT Jobs(Id, Description) VALUES ('04', 'Training');
```

Script 1. Sample Data

Example 1: CONCAT() Function

The following query will update the EmployeeCode column.

SQL Script:

```
UPDATE Employees SET EmployeeCode = CONCAT(Code, '-', DepartmentId, '-',
JobId);
```

Script 2. Example 1 for CONCAT

The following query retrieves Id and “Employee Code” from the Employee table. The "Employee Code" is a concatenation between Code, dash, DepartmentId, dash again, and JobId.

SQL Script:

```
SELECT Id, CONCAT(Code, '-', DepartmentId, '-', JobId) AS 'Employee Code'
FROM Employees;
```

Script 3. Example 2 for CONCAT

Following is the result displaying Id and "Employee Code".

Result:

Id	Employee Code
1	0001-01-01
2	0002-02-01
3	0003-01-02
4	0004-01-04
5	0005-03-01
6	0006-02-02

Table 6. Concatenated Result

Split a String in SQL Server using STRING_SPLIT()

STRING_SPLIT() function is used to split a string in SQL Server.

Syntax
STRING_SPLIT(E, S);

Table 7 . STRING_SPLIT Function

Where E is an expression. S is a single character symbol as the separator.

Example 2: Split a String in SQL Server using STRING_SPLIT()

The following example shows how to split a string using STRING_SPLIT().

SQL Server Script:

```
DECLARE @EmployeeCode VARCHAR(10) = (SELECT EmployeeCode FROM Employees
WHERE Id = 2);
SELECT * FROM STRING_SPLIT(@EmployeeCode, '-');
```

Script 4. STRING_SPLIT in SQL Server

Result:

value
0002
02
02

Table 8. Split Data

Split a string in SQL using SUBSTRING()

SUBSTRING() function can be used to split a string in SQL.

Syntax
SUBSTRING(E, N, L);

Table 9. SUBSTRING Function

Where E is an expression. N is the length to splitting point. L is the length of the returned expression.

Example 3: Split a Column in SQL

The following query will split the EmployeeCode column and retrieve data as a table in both SQL Server and MySQL

SQL Script:

```
SELECT Id, SUBSTRING(EmployeeCode, 1, 4) AS Code
, SUBSTRING(EmployeeCode, 6, 2) AS 'Department Id'
, SUBSTRING(EmployeeCode, 9, 2) AS 'Job Id'
FROM Employees;
```

Script 5. Split using SUBSTRING

Result:

Id	Code	Department Id	Job Id
1	0001	01	01
2	0002	02	01
3	0003	01	02
4	0004	01	04
5	0005	03	01
6	0006	02	02

Table 10. Split Data