Concatenate SQL Server Columns into a String with CONCAT()

By: Chad Churchwell | Read Comments (16) | Related Tips: 1 | 2 | 3 | 4 | 5 | 6 | More > Functions - System

Problem

I need to produce mailing labels from my SQL Server database so I am using the + sign for concatenation the first, middle, and last names together. The issue I see is I get NULL for a lot of rows. This makes me unable to produce the full names. What are some options to address this problem? Check out this tip to learn more about concatenating data in SQL Server.

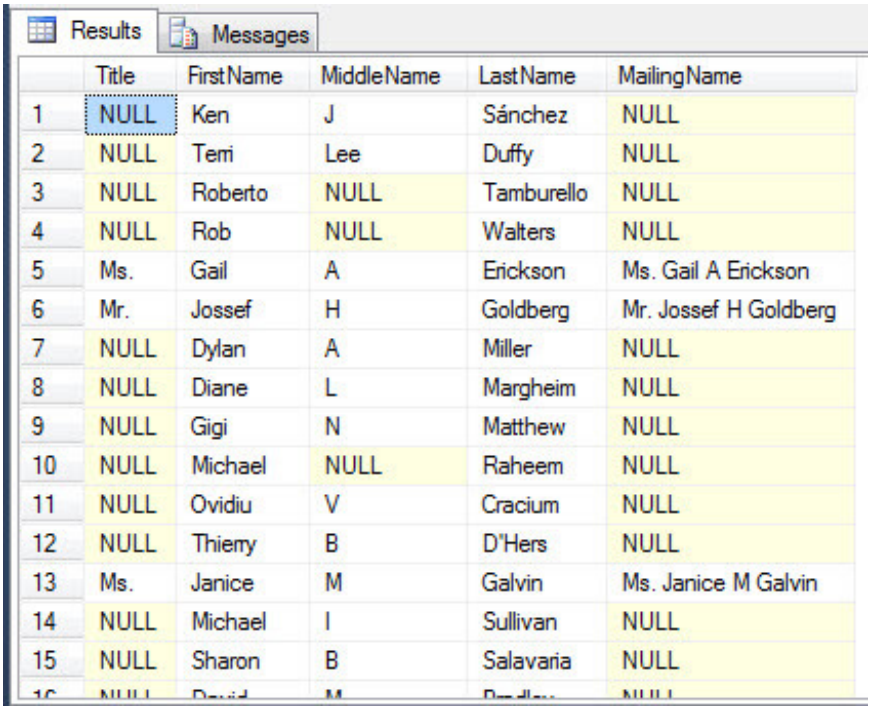
Solution

Prior to SQL Server 2012 concatenation was accomplished by using the plus (+) sign to concatenate fields together. The limitation of this method is if any of the fields you are concatenating are NULL, the entire result is NULL. With the introduction of SQL Server 2012 there is a new CONCAT() function that replaces NULL with an empty string. Take a look at this tip to see how this new function works and how it can be beneficial in your code.

For this demo I am going to use the Person.Person table from the AdventureWorks2012 database to demo having to generate a full name for creating mailing labels. First, the code below is the old technique to perform concatenation using the + sign:

```
SELECT
    Title,
    FirstName,
    MiddleName,
    LastName,
    Title+ ' ' + FirstName + ' ' + MiddleName + ' ' + LastName as MailingName
FROM Person.Person
```

As you can see in the screen shot below the MailingName is NULL for any row that has NULL for any one of the name columns. The only rows that have MailingName filled in have a value for all the title, firstname, middlename, and lastname columns. This could be corrected by wrapping ISNULL(column,'') around all the columns in the concatenated field to account for any values having nulls, but that code gets long, messy, and hard to read.



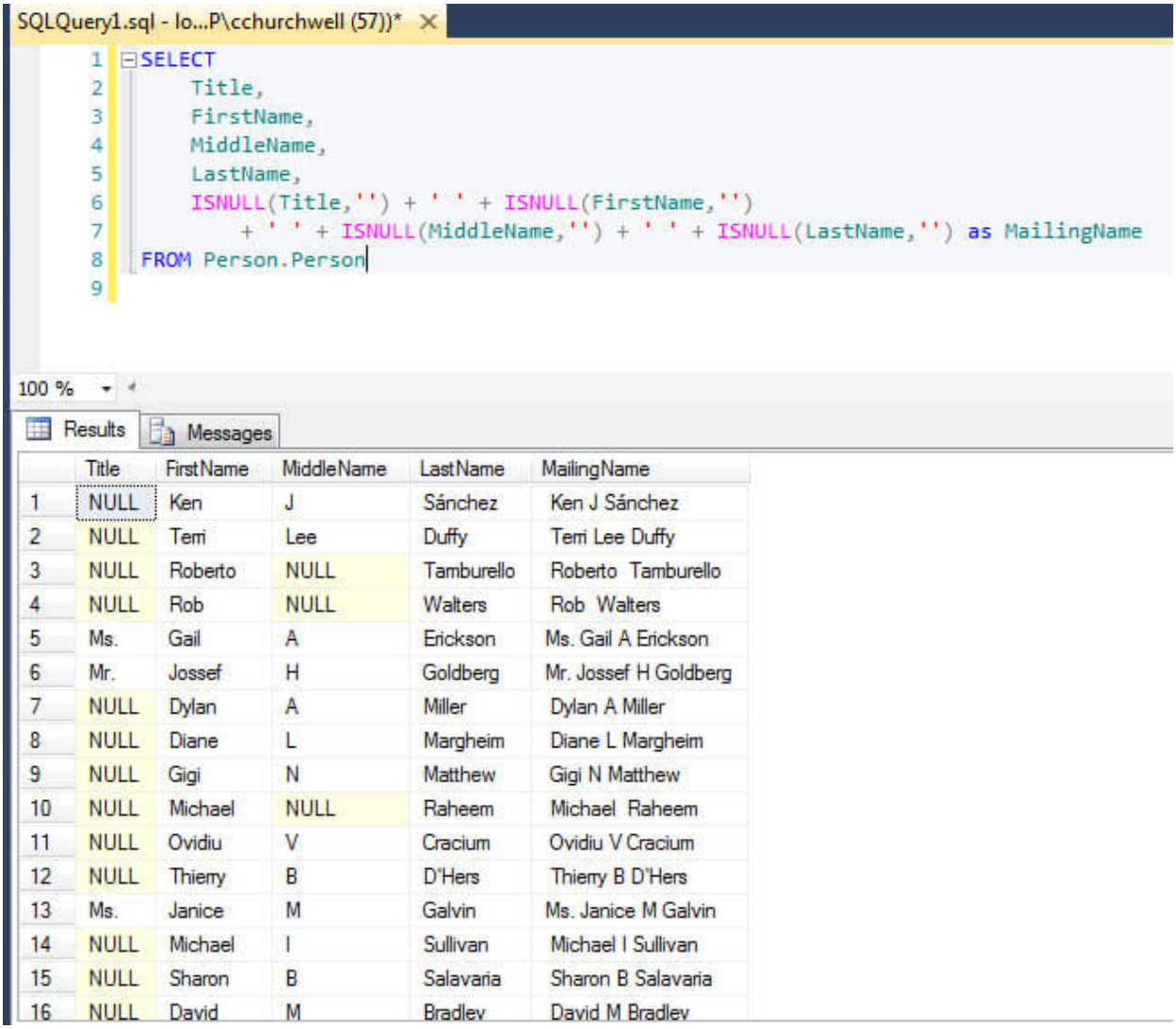
Results | Messages

	Title	FirstName	MiddleName	LastName	MailingName
1	NULL	Ken	J	Sánchez	NULL
2	NULL	Terri	Lee	Duffy	NULL
3	NULL	Roberto	NULL	Tamburello	NULL
4	NULL	Rob	NULL	Walters	NULL
5	Ms.	Gail	A	Erickson	Ms. Gail A Erickson
6	Mr.	Jossef	H	Goldberg	Mr. Jossef H Goldberg
7	NULL	Dylan	A	Miller	NULL
8	NULL	Diane	L	Margheim	NULL
9	NULL	Gigi	N	Matthew	NULL
10	NULL	Michael	NULL	Raheem	NULL
11	NULL	Ovidiu	V	Cracium	NULL
12	NULL	Thierry	B	D'Hers	NULL
13	Ms.	Janice	M	Galvin	Ms. Janice M Galvin
14	NULL	Michael	I	Sullivan	NULL
15	NULL	Sharon	B	Salavaria	NULL

Below is an example is using ISNULL along with the plus sign for concatenation. The ISNULL function will replace null values with the value noted in the second parameter, which in this example is an empty string.

```
SELECT
    Title,
    FirstName,
    MiddleName,
    LastName,
    ISNULL(Title,'') + ' ' + ISNULL(FirstName,'') + ' '
  + ISNULL(MiddleName,'') + ' ' + ISNULL(LastName,'') as MailingName
FROM Person.Person
```

As you can see in the example below, the MailingName is no longer NULL as it replaced the NULL values with an empty string. This achieves the same as using the CONCAT() function, but requires a lot more code and readability.



	Title	FirstName	MiddleName	LastName	MailingName
1	NULL	Ken	J	Sánchez	Ken J Sánchez
2	NULL	Terri	Lee	Duffy	Terri Lee Duffy
3	NULL	Roberto	NULL	Tamburello	Roberto Tamburello
4	NULL	Rob	NULL	Walters	Rob Walters
5	Ms.	Gail	A	Erickson	Ms. Gail A Erickson
6	Mr.	Jossef	H	Goldberg	Mr. Jossef H Goldberg
7	NULL	Dylan	A	Miller	Dylan A Miller
8	NULL	Diane	L	Margheim	Diane L Margheim
9	NULL	Gigi	N	Matthew	Gigi N Matthew
10	NULL	Michael	NULL	Raheem	Michael Raheem
11	NULL	Ovidiu	V	Cracium	Ovidiu V Cracium
12	NULL	Thierry	B	D'Hers	Thierry B D'Hers
13	Ms.	Janice	M	Galvin	Ms. Janice M Galvin
14	NULL	Michael	I	Sullivan	Michael I Sullivan
15	NULL	Sharon	B	Salavaria	Sharon B Salavaria
16	NULL	David	M	Bradley	David M Bradley

The next set of code is using the new CONCAT() function, new to SQL Server 2012. It replaces NULL values with an empty string of type VARCHAR(1). This code is much easier to read and write when you need to have NULL code handling in place.

```
SELECT
    Title,
    FirstName,
    MiddleName,
    LastName,
    CONCAT(Title,' ',FirstName,' ',MiddleName,' ',LastName) as MailingName
FROM Person.Person
```

If you see the results of this, all MailingName values are present, even if they have some of the columns set to NULL.

Title	FirstName	MiddleName	LastName	MailingName
NULL	Ken	J	Sánchez	Ken J Sánchez
NULL	Terri	Lee	Duffy	Terri Lee Duffy
NULL	Roberto	NULL	Tamburello	Roberto Tamburello
NULL	Rob	NULL	Walters	Rob Walters
Ms.	Gail	A	Erickson	Ms. Gail A Erickson
Mr.	Jossef	H	Goldberg	Mr. Jossef H Goldberg
NULL	Dylan	A	Miller	Dylan A Miller
NULL	Diane	L	Margheim	Diane L Margheim
NULL	Gigi	N	Matthew	Gigi N Matthew
NULL	Michael	NULL	Raheem	Michael Raheem
NULL	Ovidiu	V	Cracium	Ovidiu V Cracium
NULL	Thierry	B	D'Hers	Thierry B D'Hers
Ms.	Janice	M	Galvin	Ms. Janice M Galvin
NULL	Michael	I	Sullivan	Michael I Sullivan
NULL	Sharon	B	Salavaria	Sharon B Salavaria
NULL	David	M	Pmdlow	David M Pmdlow

As you can see this new function is very handy and behaves much different that the old form of concatenation. Instead of evaluating to NULL if any if the columns contain NULL values, the CONCAT() function replaces it with an empty string. This is very useful for coding around NULL values.

Next Steps

Check out these additional resources:

Last Update: 2013-07-22

About the author

Chad Churchwell is a SQL Server professional specializing in High Availability, Disaster Recovery, and Replication.
View all my tips

Related Resources