**Aggregrating varchar columns**

**By** [**Vikram Takrani**](http://www.sqlservercentral.com/Authors/Scripts/Vikram_Takrani/383776/)**, 2016/05/12**

Assuming you have a table Employee with below schema

CREATE TABLE [dbo].[Employee](  
[id] [int] NOT NULL,  
[status] [varchar](20) NOT NULL,  
[add1] [varchar](20) NOT NULL,  
[add2] [varchar](20) NULL  
)

--Populate Test Data

INSERT INTO [Employee] VALUES (1,'S1','S1Add1','S1Add2')  
INSERT INTO [Employee] VALUES (1,'S2','S2Add1',NULL)  
INSERT INTO [Employee] VALUES (1,'S3','S3Add1',NULL)  
INSERT INTO [Employee] VALUES (1,'S4','S4Add1','S4Add2')  
INSERT INTO [Employee] VALUES (2,'S1','S1Add1','S1Add2')  
INSERT INTO [Employee] VALUES (2,'S2','S2Add1',NULL)  
INSERT INTO [Employee] VALUES (2,'S3','S3Add1','S3Add2')

Requirement :

Columns Status,Add1 and Add2 needs to be grouped for all the rows with the same index as a comma delimited string.

e.g. in the above data we need the output as:

Id CSVs

1 S1,S1Add1,S1Add2,S2,S2Add1,,S3,S3Add1,,S4,S4Add1,S4Add2  
2 S1,S1Add1,S1Add2,S2,S2Add1,,S3,S3Add1,S3Add2

Above SQL uses below approach:

* Get row numbers for a particular index
* Recursively loop thru and append the column values till last row number

Alternative approaches could be writing a stored proc/function to do this logic of concatenation.

;WITH cte2

AS

(

--Assign row numbers to the records

SELECT id,

row\_number() OVER( PARTITION BY id ORDER BY id) col3,

status,

add1,

ISNULL(add2,'') add2

FROM Employee

)

,cte AS

(

--Anchor row will have the id and the rows for that id + 1

-- Max + 1 since this will be a break condition for recursive loop

SELECT id,max(col3)+1 Col3,cast('' AS VARCHAR(100)) Col2

FROM cte2 GROUP BY id

UNION all

SELECT t.id,t.col3,cast(t.status+','+t.add1+','+t.add2+','+c.col2 AS VARCHAR(100)) Col2

FROM cte2 t

JOIN cte c

ON c.id=t.id

WHERE c.col3 = t.col3+1

)

select id,substring(Col2,1,len(col2)-1) CSVs

from cte

where col3=1 order by id;