Saving Query Output to a String

Occasionally I have come across the need to save the output from a query to a varchar value, for instance varchar(max). There are lots of ways to do this, could you use a cursor and just append to a string each loop through the cursor, but there is an easier way.

Examples of a need to do this would be to write the output to a file, or to build an email message that is going to be sent, or many other possible scenarios.

Lets start with a simple query on SQL Server 2019, grabbing a list of databases, and with a where clause to keep the list short for now.

```
--start with a simple query SELECT *    FROM sys.databases   WHERE databa
se_id
```

```
Which produces output something like this.
```

Now lets trim it down to just the data we want to build into string output.

```
-- trim it down to just the data you want SELECT name, create_date, state
_desc   FROM sys.databases   WHERE database_id
```

But still we are no closer to getting the output into a varchar or nvarchar, so we do this:

```
-- you could use a cursor, or just concatenate a string.  DECLARE @myOutp
```

```
utString AS VARCHAR(MAX) = ''; SELECT @myOutputString += concat(name, ' '
, create_date, ' ', state_desc, char(13), char(10))   FROM sys.databases
  WHERE database_id
```

When we run this we get a list like this:

So what are these char(13) and char(10) things in the query? Those are just the the carriage return and linefeed character as needed to break to a new line. Without those the results would just all be on one line.

Notice the Plus Equal syntax to add on to the string for each iteration through the rows.

That all works on SQL Server 2012 and newer when the concat function was introduced. But what about if you are running on something older that SQL Server 2012 like SQL Server 2005 , 2008 or 2008R2?

```
--for older versions it works like this. DECLARE @myOutputString AS VARCH
AR(MAX) = ''; SELECT @myOutputString += name + ' ' + cast(create_date as
varchar(100)) +  ' ' + state_desc + char(13) + char(10)   FROM sys.databa
ses   WHERE database_id
```

For the older versions 2008 and newer, the above query works, but it breaks on 2005 due to the += operator not being avialable. You may see an error about "cannot assign to a default value to a local variable. or Incorrect syntax near '='. Then try this one on SQL 2005.

```
--What about SQL Server 2005? -- Cannot assign a default value to a local
 variable. -- Incorrect syntax near '='.  DECLARE @myOutputString AS VARC
HAR(MAX); SET @myOutputString = ''; SELECT @myOutputString = @myOutputStr
ing + name + ' ' + cast(create_date as varchar(100)) +  ' ' + state_desc
+ char(13) + char(10)   FROM sys.databases   WHERE database_id
```

What we have done to get it to work on SQL 2005 is 2 things. First we added a line to initialize the @myOutputString to an empty string, rather than doing that on the declaration line. Then we replaced the += with the assignment as shown in the red box above. That now works on SQL Server 2005.

Lets make it more interesting, since I occasionally still work on SQL Server 2000 with some clients, to do the similar thing on SQL Server 2000, what would it take? Well, you don't have varchar(max), and you don't have the sys.databases table that I used in the example. With a little bit of modification it will even work on SQL Server 2000.

```
--What about SQL Server 2000? -- Line 2: Incorrect syntax near 'MAX'. --
Invalid object name 'sys.databases'. -- Invalid column name 'create_date'
. -- Invalid column name 'database_id'.  DECLARE @myOutputString AS VARCH
AR(4000); SET @myOutputString = ''; SELECT @myOutputString = @myOutputStr
ing + name + ' ' + cast(crdate as varchar(100)) +  ' ' + cast(status as v
archar(100)) + char(13) + char(10)   FROM master.dbo.sysdatabases   WHERE
 dbid
```

Takeaways from this post:

It is quick and easy to take the output from a query and build it into an string.

This can be done on all versions of SQL Server over the last 20 years, but the syntax may vary slightly.

You don't always have to use a cursor to get the results from a query.