

The curious case of CHAR(0)

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser ('FORMSOF (INFLECTIONAL , ''study'') ', 1033, 0, 0)
)
SELECT LEFT(STUFF (@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Over the years I've seen quite a few strange things with SQL Server. This one may not have been the oddest, but it was definitely up there. A user asked me to help them copy a hashed password from one server to another. (*Note: As this was a password I was careful to make sure to get this approved by management before even looking at it.*) This was an application password and was stored in a normal `varchar(100)` column. Nothing exciting about the collation either. (`SQL_Latin1_General_CP1_CI_AS`). But when I looked at the data this is what I saw:

1

```
SELECT MyPass FROM #Password;
```

	MyPass
1	
2	ABC

Now that's odd right? The *MyPass* column in the first row looks empty, and in fact if I copy and paste it I don't get anything. (The second row was added for control so you can see nothing there is nothing special about the table.) But look at this:

1

2

3

```
SELECT MyPass, len(MyPass) AS ColLen,
```

The curious case of CHAR(0)

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser ('FORMSOF (INFLECTIONAL , ''study'') ', 1033, 0, 0)
)
SELECT LEFT(STUFF (@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Over the years I've seen quite a few strange things with SQL Server. This one may not have been the oddest, but it was definitely up there. A user asked me to help them copy a hashed password from one server to another. (*Note: As this was a password I was careful to make sure to get this approved by management before even looking at it.*) This was an application password and was stored in a normal `varchar(100)` column. Nothing exciting about the collation either. (`SQL_Latin1_General_CP1_CI_AS`). But when I looked at the data this is what I saw:

1

```
SELECT MyPass FROM #Password;
```

	MyPass
1	
2	ABC

Now that's odd right? The *MyPass* column in the first row looks empty, and in fact if I copy and paste it I don't get anything. (The second row was added for control so you can see nothing there is nothing special about the table.) But look at this:

1

2

3

```
SELECT MyPass, len(MyPass) AS ColLen,
```

The curious case of CHAR(0)

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser('FORMSOF(INFLECTIONAL, ''study'')', 1033, 0, 0)
)
SELECT LEFT(STUFF(@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Over the years I've seen quite a few strange things with SQL Server. This one may not have been the oddest, but it was definitely up there. A user asked me to help them copy a hashed password from one server to another. (*Note: As this was a password I was careful to make sure to get this approved by management before even looking at it.*) This was an application password and was stored in a normal `varchar(100)` column. Nothing exciting about the collation either. (`SQL_Latin1_General_CP1_CI_AS`). But when I looked at the data this is what I saw:

1

```
SELECT MyPass FROM #Password;
```

	MyPass
1	
2	ABC

Now that's odd right? The *MyPass* column in the first row looks empty, and in fact if I copy and paste it I don't get anything. (The second row was added for control so you can see nothing there is nothing special about the table.) But look at this:

1

2

3

```
SELECT MyPass, len(MyPass) AS ColLen,
```

The curious case of CHAR(0)

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser('FORMSOF(INFLECTIONAL, ''study'')', 1033, 0, 0)
)
SELECT LEFT(STUFF(@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Over the years I've seen quite a few strange things with SQL Server. This one may not have been the oddest, but it was definitely up there. A user asked me to help them copy a hashed password from one server to another. (*Note: As this was a password I was careful to make sure to get this approved by management before even looking at it.*) This was an application password and was stored in a normal `varchar(100)` column. Nothing exciting about the collation either. (`SQL_Latin1_General_CP1_CI_AS`). But when I looked at the data this is what I saw:

1

```
SELECT MyPass FROM #Password;
```

	MyPass
1	
2	ABC

Now that's odd right? The *MyPass* column in the first row looks empty, and in fact if I copy and paste it I don't get anything. (The second row was added for control so you can see nothing there is nothing special about the table.) But look at this:

1

2

3

```
SELECT MyPass, len(MyPass) AS ColLen,
```

The curious case of CHAR(0)

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser('FORMSOF(INFLECTIONAL, ''study'')', 1033, 0, 0)
)
SELECT LEFT(STUFF(@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Over the years I've seen quite a few strange things with SQL Server. This one may not have been the oddest, but it was definitely up there. A user asked me to help them copy a hashed password from one server to another. (*Note: As this was a password I was careful to make sure to get this approved by management before even looking at it.*) This was an application password and was stored in a normal `varchar(100)` column. Nothing exciting about the collation either. (`SQL_Latin1_General_CP1_CI_AS`). But when I looked at the data this is what I saw:

1

```
SELECT MyPass FROM #Password;
```

	MyPass
1	
2	ABC

Now that's odd right? The *MyPass* column in the first row looks empty, and in fact if I copy and paste it I don't get anything. (The second row was added for control so you can see nothing there is nothing special about the table.) But look at this:

1

2

3

```
SELECT MyPass, len(MyPass) AS ColLen,
```

The curious case of CHAR(0)

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser('FORMSOF(INFLECTIONAL, ''study'')', 1033, 0, 0)
)
SELECT LEFT(STUFF(@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Over the years I've seen quite a few strange things with SQL Server. This one may not have been the oddest, but it was definitely up there. A user asked me to help them copy a hashed password from one server to another. (*Note: As this was a password I was careful to make sure to get this approved by management before even looking at it.*) This was an application password and was stored in a normal `varchar(100)` column. Nothing exciting about the collation either. (`SQL_Latin1_General_CP1_CI_AS`). But when I looked at the data this is what I saw:

1

```
SELECT MyPass FROM #Password;
```

	MyPass
1	
2	ABC

Now that's odd right? The *MyPass* column in the first row looks empty, and in fact if I copy and paste it I don't get anything. (The second row was added for control so you can see nothing there is nothing special about the table.) But look at this:

1

2

3

```
SELECT MyPass, len(MyPass) AS ColLen,
```

The curious case of CHAR(0)

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser ('FORMSOF (INFLECTIONAL , ''study'') ', 1033, 0, 0)
)
SELECT LEFT(STUFF (@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Over the years I've seen quite a few strange things with SQL Server. This one may not have been the oddest, but it was definitely up there. A user asked me to help them copy a hashed password from one server to another. (*Note: As this was a password I was careful to make sure to get this approved by management before even looking at it.*) This was an application password and was stored in a normal `varchar(100)` column. Nothing exciting about the collation either. (`SQL_Latin1_General_CP1_CI_AS`). But when I looked at the data this is what I saw:

1

```
SELECT MyPass FROM #Password;
```

	MyPass
1	
2	ABC

Now that's odd right? The *MyPass* column in the first row looks empty, and in fact if I copy and paste it I don't get anything. (The second row was added for control so you can see nothing there is nothing special about the table.) But look at this:

1

2

3

```
SELECT MyPass, len(MyPass) AS ColLen,
```

The curious case of CHAR(0)

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser ('FORMSOF (INFLECTIONAL , ''study'') ', 1033, 0, 0)
)
SELECT LEFT(STUFF (@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Over the years I've seen quite a few strange things with SQL Server. This one may not have been the oddest, but it was definitely up there. A user asked me to help them copy a hashed password from one server to another. (*Note: As this was a password I was careful to make sure to get this approved by management before even looking at it.*) This was an application password and was stored in a normal `varchar(100)` column. Nothing exciting about the collation either. (`SQL_Latin1_General_CP1_CI_AS`). But when I looked at the data this is what I saw:

1

```
SELECT MyPass FROM #Password;
```

	MyPass
1	
2	ABC

Now that's odd right? The *MyPass* column in the first row looks empty, and in fact if I copy and paste it I don't get anything. (The second row was added for control so you can see nothing there is nothing special about the table.) But look at this:

1

2

3

```
SELECT MyPass, len(MyPass) AS ColLen,
```


The curious case of CHAR(0)

SQL STUDIES

```
WITH ToLearn AS (
    SELECT row_number() OVER (ORDER BY keyword) AS RowNumber, *
    FROM sys.dm_fts_parser ('FORMSOF (INFLECTIONAL , ''study'') ', 1033, 0, 0)
)
SELECT LEFT(STUFF (@@VERSION, 2, 8, 'y'), 14) +
    UPPER(LEFT(display_term, 1)) +
    SUBSTRING(display_term, 2, 20) +
    CHAR(46)
FROM ToLearn
WHERE RowNumber = 2
```

Over the years I've seen quite a few strange things with SQL Server. This one may not have been the oddest, but it was definitely up there. A user asked me to help them copy a hashed password from one server to another. (*Note: As this was a password I was careful to make sure to get this approved by management before even looking at it.*) This was an application password and was stored in a normal `varchar(100)` column. Nothing exciting about the collation either. (`SQL_Latin1_General_CP1_CI_AS`). But when I looked at the data this is what I saw:

1

```
SELECT MyPass FROM #Password;
```

	MyPass
1	
2	ABC

Now that's odd right? The *MyPass* column in the first row looks empty, and in fact if I copy and paste it I don't get anything. (The second row was added for control so you can see nothing there is nothing special about the table.) But look at this:

1

2

3

```
SELECT MyPass, len(MyPass) AS ColLen,
```