LAB - CHAR Data Types

In this lab, you will learn how to use the SQL Server `CHAR` data type to store the fixed-length, non-Unicode character strings in the database.

If you want to store fixed length, non-Unicode string data, you use the SQL Server `CHAR` data type:

```
CHAR(n)
```

In this syntax, n specifies the string length which ranges from 1 to 8,000.

Because n is optional, if don't specify it in a data definition or variable declaration statement, its default value is 1.

You should use the `CHAR` data type only when the sizes of values in the column are fixed.

When you insert a string value into a `CHAR` column. If the length of the string value is less than the length specified in the column, SQL Server will add trailing spaces to the string value to the length declared in the column. However, when you select this string value, SQL Server removes the trailing spaces before returning it.

On the other hand, if you insert a value whose length exceeds the column length, SQL Server issues an error message.

Note that the ISO synonym for `CHAR` is `CHARACTER` so you can use them interchangeably.

Examples

The following statement creates a new table that contains a `CHAR` column:

```
CREATE TABLE sql_server_char (
    val CHAR(3)
);
```

To insert a fixed-length character string into the `CHAR` column, you use the `INSERT` statement as follows:

```
INSERT INTO sql_server_char (val)
VALUES
    ('ABC');
```

The statement worked as expected.

The following statement attempts to insert a new character string whose length exceeds the column length:

```
INSERT INTO sql_server_char (val)
VALUES
    ('XYZ1');
```

SQL Server issued the following error:

```
String or binary data would be truncated.
The statement has been terminated.
```

The following statement inserts a single character into the `val` column of the test. `sql_server_char` table:

```
INSERT INTO sql_server_char (val)
VALUES
    ('A');
```

In SQL Server, `LEN` function returns the number of characters in a specified column that excludes the trailing blanks and the `DATALENGTH` function returns the number of bytes.

See the following statement:

```
SELECT
    val,
    LEN(val) len,
    DATALENGTH(val) data_length
FROM
    sql_server_char;
```

val	len	data_length
ABC	3	3
A	1	3

Even though the character 'A' is only one character, the number of bytes of the column is fixed which is three.

In this lab, you have learned how to use the SQL Server `CHAR` data type to store fixed-length, non-Unicode character strings in the database.