You have learned how to retrieve data from SQL Server tables in a number of ways: through simple queries, through joins, with functions, and more. You have learned to manipulate data, write scripts, and create database objects. Essentially, you have learned the T-SQL basics. Not only have you learned these skills, but you have learned to think about the best way to solve a problem, not just the easy way.

This chapter introduces some of the more interesting and complex data types available in SQL Server 2012. You will learn about sparse columns, CLR data types (HIERARCHYID, GEOMETRY, and GEOGRAPHY), enhanced date and time data types, large-value data types (MAX), and FILESTREAM data. Some of these, such as the CLR data types, are nothing like the traditional data types you have been using throughout this book. This chapter provides a glimpse of these interesting new data types.

Chapters 1 through 9 covered the important skills you need to become a proficient T-SQL developer. Since this chapter covers “bonus material,” it doesn’t contain exercises. I encourage you to practice working with any of the new data types that interest you or that you think will be beneficial in your job.

Large-Value String Data Types (MAX)

Older versions of SQL Server used NTEXT and TEXT data types to represent large values. Microsoft has deprecated those types, which means that in some future release of SQL Server, NTEXT and TEXT will no longer work. For now, however, the deprecated data types still work in SQL Server 2012. Going forward, you should replace these data types with VARCHAR(MAX) and NVARCHAR(MAX).

The TEXT and NTEXT data types have many limitations. For example, you can’t declare a variable of type TEXT or NTEXT, use them with most functions, or use them within most search criteria. The MAX data types represent the benefits of both the regular string data types and the TEXT and NTEXT data types when storing large strings. They allow you to store large amounts of data and offer the same functionality of the traditional data types.

When creating string data types, you supply a number of characters. Instead of supplying a number, use the word MAX when the data is going to surpass the maximum normally allowed. Table 10-1 lists the differences between the string value data types.
Table 10-1. The String Data Types

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Maximum Characters</th>
<th>Character Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR</td>
<td>Fixed width</td>
<td>8,000</td>
<td>ASCII</td>
</tr>
<tr>
<td>NCHAR</td>
<td>Fixed width</td>
<td>4,000</td>
<td>Unicode</td>
</tr>
<tr>
<td>VARCHAR</td>
<td>Variable width</td>
<td>8,000</td>
<td>ASCII</td>
</tr>
<tr>
<td>NVARCHAR</td>
<td>Variable width</td>
<td>4,000</td>
<td>Unicode</td>
</tr>
<tr>
<td>TEXT</td>
<td>Variable width</td>
<td>$2^{31} - 1$</td>
<td>ASCII</td>
</tr>
<tr>
<td>NTEXT</td>
<td>Variable width</td>
<td>$2^{30} - 1$</td>
<td>Unicode</td>
</tr>
<tr>
<td>VARCHAR(MAX)</td>
<td>Variable width</td>
<td>$2^{31} - 1$</td>
<td>ASCII</td>
</tr>
<tr>
<td>NVARCHAR(MAX)</td>
<td>Variable width</td>
<td>$2^{30} - 1$</td>
<td>Unicode</td>
</tr>
</tbody>
</table>

You work with the MAX string data types just like you do with the traditional types for the most part. Type in and execute Listing 10-1 to learn how to work with the MAX types.

Listing 10-1. Using VARCHAR(MAX)

```
--1
CREATE TABLE #maxExample (maxCol VARCHAR(MAX),
    line INT NOT NULL IDENTITY PRIMARY KEY);
GO

--2
INSERT INTO #maxExample(maxCol)
VALUES ('This is a varchar(max)');

--3
INSERT INTO #maxExample(maxCol)
VALUES (REPLICATE('aaaaaaaaaa',9000));

--4
INSERT INTO #maxExample(maxCol)
VALUES (REPLICATE(CONVERT(VARCHAR(MAX) , 'bbbbbbbbbb'),9000));

--5
SELECT LEFT(MaxCol,10) AS Left10,LEN(MaxCol) AS varLen
FROM #maxExample;
GO
DROP TABLE #maxExample;
```