

DATABASE DATATYPES

Numbers

Tinyint	0 to 255
Smallint	+/- 32768
Int	+/- 2.147 billion
Bigint	+/- 9.223 quintillion
Bit	0 or 1
Decimal	+/- 10 ³⁸
Money	+/- 922 trillion + 4 decimal places
Smallmoney	+/- 214,748 + 4 decimal places

Dates & Time

Datetime	Jan 1, 1753 – Dec 31, 9999 + time
Datetime2	Jan 1, 0001 – Dec 31, 9999 + time
Smalldatetime	Jan 1, 1900 – June 6, 2079 + time
Date	Jan 1, 0001 – Dec 31, 9999
Time	00:00:00.0000000 – 23:59:59.9999999

Textual

Char	8000 fixed width character max
Varchar	8000 character max
Varchar(max)	2 ³¹ character max
Nchar	4000 fixed width Unicode character max
Nvarchar	4000 Unicode character max
Nvarchar(max)	2 ³⁰ Unicode character max

Identification

Uniqueidentifier (GUID)	Sql Server data type for a GUID (Globally Unique Identifier) with the following hexadecimal format: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
----------------------------	--

TIPS TO CHOOSE THE APPROPRIATE DATA TYPE

SQL Server stores data in a special structure called data pages that are 8Kb (8192 bytes) in size. Some space on the data pages is used to store system information, which leaves 8060 bytes to store user's data. So, if the table's row size is 4040 bytes, then only one row will be placed on each data page. If you can decrease the row size to 4030 bytes, you can store two rows within a single page. The less space used, the smaller the table and index, and the less the I/O SQL Server has to perform when reading data pages from disk. So, you should design your tables in such a way as to maximize the number of rows that can fit into one data page. To maximize the number of rows that can fit into one data page, you should specify the narrowest columns you can. The narrower the columns, the less data that is stored, and the faster SQL Server is able to read and write data. Try to use the following tips when choosing data types.

1. If you need to store integer data from 0 through 255, use tinyint data type. The columns with tinyint data type use only one byte to store their values, in comparison with two bytes, four bytes and eight bytes used to store the columns with smallint, int and bigint data types accordingly. For example, if you design tables for a small company with 5-7 departments, you can create the departments table with the DepartmentID tinyint column to store the unique number of each department.
2. If you need to store integer data from -32,768 through 32,767, use smallint data type. The columns with smallint data type use only two bytes to store their values, in comparison with four bytes and eight bytes used to store the columns with int and bigint data types accordingly. For example, if you design tables for a company with several hundred employees, you can create an employee table with the EmployeeID smallint column to store the unique number of each employee.
3. If you need to store integer data from -2,147,483,648 through 2,147,483,647, use int data type. The columns with int data type use only four bytes to store their values, in comparison with eight bytes used to store the columns with bigint data types. For example, to design tables for a library with more than 32,767 books, create a books table with a BookID int column to store the unique number of each book.
4. Use smallmoney data type instead of money data type, if you need to store monetary data values from 214,748.3648 through 214,748.3647. The columns with smallmoney data type use only four bytes to store their values, in comparison with eight bytes used to store the columns with money data types. For example, if you need to store the monthly employee payments, it might be possible to use a column with the smallmoney data type instead of money data type.
5. Use smalldatetime data type instead of datetime data type, if you need to store the date and time data from January 1, 1900 through June 6, 2079, with accuracy to the minute. The columns with smalldatetime data type use only four bytes to store their values, in comparison with eight bytes used to store the columns with datetime data types. For example, if you need to store the

employee's hire date, you can use a column with the smalldatetime data type instead of datetime data type.

6. Use **varchar/nvarchar columns instead of text/ntext columns whenever possible**. Because SQL Server stores text/ntext columns on the Text/Image pages separately from the other data, stored on the Data pages, it can take more time to get the text/ntext values.

7. Use **char/varchar columns instead of nchar/nvarchar if you do not need to store unicode data**. The char/varchar value uses only one byte to store one character, the nchar/nvarchar value uses two bytes to store one character, so the char/varchar columns use two times less space to store data in comparison with nchar/nvarchar columns.