

Understand the impact of wrong choices for column data types

Note:

You are not required to complete the processes, tasks, activities, or steps presented in this example. The various samples provided are for illustrative purposes only and it's likely that if you try this out you will encounter issues in your system.

Dedicated SQL pools support the most commonly used data types. With that, it is essential to keep in mind that minimizing the size of data types shortens the row length, which leads to better query performance. Use the smallest data type that works for your data:

- Avoid defining character columns with a large default length. For example, if the longest value is 25 characters, define your column as VARCHAR(25).
- Avoid using NVARCHAR when you only need VARCHAR to support non-unicode data.
- When possible, use NVARCHAR(4000) or VARCHAR(8000) instead of NVARCHAR(MAX) or VARCHAR(MAX).
- Focus on using the minimal data type as possible. Use the minor data type that can reliably contain all possible values you have. For example, use `tinyint` instead of `smallint` or `int` for exact-numbers between 0-255. `tinyint` would be sufficient for a person's age because no one lives to be more than 255 years old.
- Use DATETIME instead of storing data values in different formats such as string or numeric values. When it comes to querying your data based on date, an additional conversion will put a massive overhead on your queries.
- If you only need to store a date and not the time make sure you use SMALLDATETIME instead of DATETIME. There is a 4 bytes difference between the two.

Note

If you are using PolyBase external tables to load your SQL pool tables, the table row's defined length cannot exceed 1 MB. When a row with variable-length data exceeds 1 MB, you can load the row with BCP but not with PolyBase.

Workarounds for unsupported data types

The following list shows the data types that dedicated SQL pool doesn't support and gives useful alternatives for unsupported data types.

Unsupported data type Workaround

geometry varbinary

geography varbinary

Unsupported data type Workaround

hierarchyid	nvarchar
image	varbinary
text	varchar
ntext	nvarchar
sql_variant	Split column into several strongly typed columns.
table	Convert to temporary tables.
timestamp	Rework code to use datetime2 and the CURRENT_TIMESTAMP function. Only constants are supported as defaults, so current_timestamp can't be defined as a default constraint. If you need to migrate row version values from a timestamp typed column, use BINARY(8) or VARBINARY(8) for NOT NULL or NULL row version values.
xml]	varchar
user-defined type	Convert back to the native data type when possible.
default values	Default values support literals and constants only.