

Define Column Data Types

Database Design Goal:

Data types that allow '*right*' data in **least amount of space**

Seven Categories of SQL Server Data Types

- Exact numeric
- Approximate numeric
- Monetary
- Date and Time
- Character
- Binary
- Special Purpose

Example 1: Integers

- Bigint

- Integer (whole number) data from -2^{63} (-9,223,372,036,854,775,808) through $2^{63}-1$ (9,223,372,036,854,775,807)

- INT

- Integer (whole number) data from -2^{31} (-2,147,483,648) through $2^{31} - 1$ (2,147,483,647)

- Smallint

- Integer data from -2^{15} (-32,768) through $2^{15} - 1$ (32,767)

- TinyINT

- Integer data from 0 through 255

Example 2: Date & Time

- **datetime**

- Date and time data from January 1, 1753, through December 31, 9999, with an accuracy of three-hundredths of a second, or 3.33 milliseconds

- **smalldatetime**

- Date and time data from January 1, 1900, through June 6, 2079, with an accuracy of one minute

- **date**

- Date values from 0001-01-01 through 9999-12-31

- **time**

- The default accuracy is 100 nanoseconds (customizable)
- zero to seven places to the right of the decimal (3 – 5 bytes)

Example 3: Character Strings

- `char(n)`
 - **Fixed-length**, non-Unicode string data. `n` defines the string length and must be a value from **1 through 8,000**, such as `char(5)`.
- **Varchar(n)**
 - **Variable-length**, non-Unicode string data. `n` defines the string length and can be a value from **1 through 8,000**. **max indicates that the maximum storage size is $2^{31}-1$ bytes (2 GB)**. An example is `varchar(100)`.

Best Practice

- Choosing 'correct' data type is important
- Go for smallest workable data type to save space on disk and in memory. It also increases performance.