# Terms

Chapter 8

1. **string data type** – stores string of 1 or more characters
2. **numeric data type** - stores numbers that can be used in mathematical calculations
3. **temporal data type** - stores dates and times
4. **date/time data type** - The date and time data types are used to store information related to dates and times
    a. dateime – dates and times from Jan 1, 1750 through Dec 31, 99999
    b. smalldatetime – dates and times from Jan 1, 1900 through June 6, 2079
    c. date – stores dates only from Jan 1, 00001 through Dec 31, 99999
    d. time(n) – stores only time (no date) n specifies the precision for fractional second precision
    e. datetime2(n) – dates from Jan 1, 00001 through Dec 31, 99999 with time values from 00:00:00.0000000 to 23:59:59.9999999
    f. datetimeoffset(n) – includes a time zone offset from -14 to 14 and is used with datetime2
5. **integer data type** – whole numbers (an exact numeric data type)
    a. bigint (8 bytes)
    b. int (4 bytes)
    c. smallint (2 bytes)
    d. tinyint (1 byte)
    e. bit ( 1/8 of a byte – value of 0 or 1)
6. **decimal data type** – numbers with decimal places (an exact numeric data type)
    a. decimal - fixed precision (1 to 38, default is 18) and scale (0 to precision, default is 18). Size varies between 5-17 bytes.
    b. numeric (same as decimal)
    c. money – same as Decimal (19,4) where 19 is precision, 4 is decimal places. Size is 8 bytes
    d. small money – same as decimal(10,4) where 10 is precision, 4 is decimal places. Size is 4 bytes
7. **scale** – number of decimal places
8. **precision** – total number of digits in a decimal data type
9. **real data type** – store floating point numbers with a limited number of significant digits (an approximate numeric data type because values could be rounded).
    a. Float – stores single or double precision numbers with a specific scale (number of decimal places). The default scale is 53 (double precision). Size is 4 to 8 bytes depending upon the precision and scale
    b. Real – stores single precision floating point numbers. The default scale is 24
10. **fixed-point number** – decimal data type. Can store a single-precision number up to 7 decimal places
11. **floating-point number** – real data type. Can store a double precision number with up to 15 decimal places

12. **significant digits** – digits of a number used to express it to the degree of accuracy starting with the first non-zero number
13. **single-precision number** - allows for a max of 7 significant digits (real data type)
14. **double-precision number** - allows for a max of 15 significant digits (float data type)
15. **scientific notation** – used to express the value of a floating-point number (typically used for very large or very small numbers.
16. **exact numeric data types** – decimal data types (the precision is exact). Preferred by business
17. **approximate numeric data types** – real data types (the precision is not exact, but approximate). If a number has too many decimal places, it is rounded (that is why the value is approximate).
18. **Unicode character** – standardized character set that includes the characters for all the world's languages (an encoding system like UTF-8 is used to translate the characters into bits). Should only be used in a multi-language environment
19. **Unicode specification** – specifies how to encode characters for all languages. Supports a wider range of characters and data types. Takes up 2X the space per character.
20. **national character** – two-byte characters specified in the ANSI standards
21. **fixed-length string** – strings that have a fixed length defined and will always use the space regardless of the character size.
    a. char(size) – national character, with size values between 1 and 8000
    b. nchar(size) – Unicode character , with size values between 1 and 4000
22. **variable-length string** – only use the amount of space they need rather than always using the maximum space
    a. varchar (size) – national character, size values between 1 and 8000
    b. nvarchar(size) – Unicode character, with size values between 1 and 4000
23. **bit** – value of zero or one
24. **ASCII (American Standard Code for Information Interchange)** – standardized character set developed in the 1960's
25. **large value data types** – created to store image and sound files which take up more space. Can store up to 2 GB of data
    a. varchar(max) – standard variable length data (text), max size approx. 2GB
    b. nvarchar(max) – Unicode variable length data (text), max size approx. 2GB
    c. varbinary(max) – variable length binary data (image files, sound files etc), max size approx. 2GB
26. **implicit conversion** – SQL server converts the data type automatically when you assign data to a column of a different data type
27. **explicit conversion** – you use CAST or CONVERT to convert one data type to another (occurs when SQL cannot do an implicit conversion) Examples: character types to money types, dat types to numeric types, money types to character types
28. **cast** – converts an expression from one data type to another (ANSI-standard that is cross platform)
29. **convert** – converts an expression from one data type to another and allows you to specify a style code for formatting (unique to SQL Server and not part of the ANSI-standard)
30. **Universal Time Coordinate (UTC)** – time standard that is the basis for time zones worldwide
31. **Greenwich Mean Time (GMT)** – time zone used in European and African countries