Transact-SQL conversion instructions

SQL Data Types

	Object_type	Interval	Storage	Comments
Exact Numerics	bigint	-2^{63} (-9E+18) to 2^{63}-1 (9E+18)	8 octets	For representing whole numbers.
	int	-2^{31} (-2E+9) to 2^{31}-1 (2E+9)	4 octets	
	smallint	-2^{15} (-32 768) to 2^{15}-1 (32 767)	2 octets	
	tinyint	0 to 255	1 octet	
	bit	0, 1 or NULL	1 bit	
	decimal [(p[, s])] or numeric [(p[, s])]	- 10^{38}+1 to 10^{38}-1	5, 9, 13, or 17 octets according to precision p	p : maximum total number of digits (18 by default) s : number of digits after the decimal (0 by default)
	money	-922 337 203 685 477.5808 to 922 337 203 685 477.5807	8 octets	
	smallmoney	- 214 748.3648 to 214 748.3647	4 octets	
Approximate numerics	float [(n)]	- 1.79E+308 to -2.23E-308, 0 and 2.23E-308 to 1.79E+308	4 or 8	n : number of bits to store the mantissa (53 by default)
	real	- 3.40E+38 to -1.18E-38, 0 and 1.18E-38 to 3.40E+38	4 octets	
Date and Time	datetime	January 1st 1753 to December 31st 9999	8 octets	
	smalldatetime	January 1st 1900 to June 6th 2079	4 octets	
Character Strings	char [(n)]	String of fixed length n character maximum	n octets	Fixed length (1 ≤ n ≤ 8000)
	varchar [(n)]	String of variable length n character maximum	n+2 octets	Variable length (1 ≤ n ≤ 8000)
	text	String of 2^{31}-1 characters maximum		
Unicode Character Strings	nchar [(n)]	Same as above in Unicode coding		
	nvarchar [(n)]			
	ntext			
Binary Strings	binary [(n)]	binary number of n bits, with fixed length	n octets	Fixed length (1 ≤ n ≤ 8000)
	varbinary [(n)]	binary number of n bits, with variable length	n+2 octets	Variable length (1 ≤ n ≤ 8000)
	image	binary data from 0 to 231-1 bytes		

Type Conversion Instructions

To convert an expression from one type to another :

```
CAST ( expression AS Object_type [ ( size ) ])
CONVERT ( Object_type [ ( size ) ] , expression [ , style ] )
```

To convert a number to a string of characters :

```
STR ( number [ , size [ , digit ] ] )
```

converts the number to a string of characters size, with the most digit after the decimal point.
(by default, size is 10 and digit is 0)

Conversion examples :

	value			**comments**
	67.459	**13.57234**	**164.34567**	
CONVERT(int, *value*)	67	13	164	truncation
CONVERT(decimal(3, 1), *value*)	67.5	13.6	error	3 digit max. + 1 decimal
CONVERT(decimal(5, 1), *value*)	67.5	13.6	164.3	5 digit max. + 1 decimal
CONVERT(decimal(5, 2), *value*)	67.46	13.57	164.35	5 digit max. + 2 decimals
STR(*value*, 5, 3)	'67.46'	'13.57'	'164.3'	5 characters + 3 decimals or more
STR(*value*, 5, 2)	'67.46'	'13.57'	'164.3'	5 characters + 2 decimals or more
STR(*value*, 4, 3)	'67.5'	'13.6'	' 164'	4 characters + 3 decimals or more
STR(*value*, 4, 2)	'67.5'	'13.6'	' 164'	4 characters + 2 decimals or more
STR(*value*, 3, 3)	' 67'	' 14'	'164'	3 characters + 3 decimals or more
STR(*value*, 2, 1)	'67'	'14'	'**'	2 characters + 1 decimal or more
STR(*value*, 6, 4)	'67.459'	'13.572'	'164.35'	6 characters + 4 decimals or more

NB: [...] means that what is inside is optional.