**SQL Server常用函数**

**一、字符转换函数  
1**、ASCII()  
返回字符表达式最左端字符的ASCII 码值。在ASCII（）函数中，纯数字的字符串可不用‘’括起来，但含其它字符的字符串必须用‘’括起来使用，否则会出错。  
  
**2**、CHAR()  
将ASCII 码转换为字符。如果没有输入0 ~ **255** 之间的ASCII 码值，CHAR（） 返回NULL 。  
  
**3**、LOWER()和UPPER()  
LOWER()将字符串全部转为小写；UPPER()将字符串全部转为大写。  
  
**4**、STR()  
把数值型数据转换为字符型数据。  
STR (<float\_expression>[，length[， <decimal>]])  
length 指定返回的字符串的长度，decimal 指定返回的小数位数。如果没有指定长度，缺省的length 值为10， decimal 缺省值为0。  
当length 或者decimal 为负值时，返回NULL；  
当length 小于小数点左边（包括符号位）的位数时，返回length 个\*；  
先服从length ，再取decimal ；  
当返回的字符串位数小于length ，左边补足空格。  
  
**二、去空格函数  
1**、LTRIM() 把字符串头部的空格去掉。  
  
**2**、RTRIM() 把字符串尾部的空格去掉。  
  
**三、取子串函数  
1**、left()   
LEFT (<character\_expression>， <integer\_expression>)  
返回character\_expression 左起 integer\_expression 个字符。  
  
**2**、RIGHT()   
RIGHT (<character\_expression>， <integer\_expression>)  
返回character\_expression 右起 integer\_expression 个字符。  
  
**3**、SUBSTRING()  
SUBSTRING (<expression>， <starting\_ position>， length)  
返回从字符串左边第starting\_ position 个字符起length个字符的部分。  
  
**四、字符串比较函数**  
**1**、CHARINDEX()  
返回字符串中某个指定的子串出现的开始位置。  
CHARINDEX (<’substring\_expression’>， <expression>)  
其中substring \_expression 是所要查找的字符表达式，expression 可为字符串也可为列名表达式。如果没有发现子串，则返回0 值。  
此函数不能用于TEXT 和IMAGE 数据类型。  
  
**2**、PATINDEX()  
返回字符串中某个指定的子串出现的开始位置。  
PATINDEX (<’%substring \_expression%’>， <column\_ name>)其中子串表达式前后必须有百分号“%”否则返回值为0。  
与CHARINDEX 函数不同的是，PATINDEX函数的子串中可以使用通配符，且此函数可用于CHAR、 VARCHAR 和TEXT 数据类型。  
  
**五、字符串操作函数**  
**1**、QUOTENAME()  
返回被特定字符括起来的字符串。  
QUOTENAME (<’character\_expression’>[， quote\_ character]) 其中quote\_ character 标明括字符串所用的字符，缺省值为“[]”。  
  
**2**、REPLICATE()  
返回一个重复character\_expression 指定次数的字符串。  
REPLICATE (character\_expression integer\_expression) 如果integer\_expression 值为负值，则返回NULL 。  
  
**3**、REVERSE()  
将指定的字符串的字符排列顺序颠倒。  
REVERSE (<character\_expression>) 其中character\_expression 可以是字符串、常数或一个列的值。  
  
**4**、REPLACE()  
返回被替换了指定子串的字符串。  
REPLACE (<string\_expression1>， <string\_expression2>， <string\_expression3>) 用string\_expression3 替换在string\_expression1 中的子串string\_expression2。  
  
**5**、SPACE()  
返回一个有指定长度的空白字符串。  
SPACE (<integer\_expression>) 如果integer\_expression 值为负值，则返回NULL 。  
  
**6**、STUFF()  
用另一子串替换字符串指定位置、长度的子串。  
STUFF (<character\_expression1>， <start\_ position>， <length>，<character\_expression2>)  
如果起始位置为负或长度值为负，或者起始位置大于character\_expression1 的长度，则返回NULL 值。  
如果length 长度大于character\_expression1 中 start\_ position 以右的长度，则character\_expression1 只保留首字符。  
  
**六、数据类型转换函数**  
**1**、CAST()  
CAST (<expression> AS <data\_ type>[ length ])  
  
**2**、CONVERT()  
CONVERT (<data\_ type>[ length ]， <expression> [， style])  
  
**1**）data\_type为SQL Server系统定义的数据类型，用户自定义的数据类型不能在此使用。  
**2**）length用于指定数据的长度，缺省值为30。  
**3**）把CHAR或VARCHAR类型转换为诸如INT或SAMLLINT这样的INTEGER类型、结果必须是带正号或负号的数值。  
**4**）TEXT类型到CHAR或VARCHAR类型转换最多为8000个字符，即CHAR或VARCHAR数据类型是最大长度。  
**5**）IMAGE类型存储的数据转换到BINARY或VARBINARY类型，最多为8000个字符。  
**6**）把整数值转换为MONEY或SMALLMONEY类型，按定义的国家的货币单位来处理，如人民币、美元、英镑等。  
**7**）BIT类型的转换把非零值转换为1，并仍以BIT类型存储。  
**8**）试图转换到不同长度的数据类型，会截短转换值并在转换值后显示“+”，以标识发生了这种截断。  
**9**）用CONVERT（）函数的style 选项能以不同的格式显示日期和时间。style 是将DATATIME 和SMALLDATETIME 数据转换为字符串时所选用的由SQL Server 系统提供的转换样式编号，不同的样式编号有不同的输出格式。  
  
**七、日期函数  
1**、day(date\_expression)   
返回date\_expression中的日期值  
  
**2**、month(date\_expression)  
返回date\_expression中的月份值  
  
**3**、year(date\_expression)  
返回date\_expression中的年份值  
  
**4**、DATEADD()  
DATEADD (<datepart>， <number>， <date>)  
返回指定日期date 加上指定的额外日期间隔number 产生的新日期。  
  
**5**、DATEDIFF()  
DATEDIFF (<datepart>， <date1>， <date2>)  
返回两个指定日期在datepart 方面的不同之处，即date2 超过date1的差距值，其结果值是一个带有正负号的整数值。  
  
**6**、DATENAME()  
DATENAME (<datepart>， <date>)  
以字符串的形式返回日期的指定部分此部分。由datepart 来指定。  
  
**7**、DATEPART()  
DATEPART (<datepart>， <date>)  
以整数值的形式返回日期的指定部分。此部分由datepart 来指定。  
DATEPART (dd， date) 等同于DAY (date)  
DATEPART (mm， date) 等同于MONTH (date)  
DATEPART (yy， date) 等同于YEAR (date)  
  
**8**、GETDATE()  
以DATETIME 的缺省格式返回系统当前的日期和时间。  
 **9、Sql Server 中一个非常强大的日期格式化函数**Select CONVERT(varchar(100), GETDATE(), 0): 05 16 2006 10:57AM  
Select CONVERT(varchar(100), GETDATE(), 1): 05/16/06  
Select CONVERT(varchar(100), GETDATE(), 2): 06.05.16  
Select CONVERT(varchar(100), GETDATE(), 3): 16/05/06  
Select CONVERT(varchar(100), GETDATE(), 4): 16.05.06  
Select CONVERT(varchar(100), GETDATE(), 5): 16-05-06  
Select CONVERT(varchar(100), GETDATE(), 6): 16 05 06  
Select CONVERT(varchar(100), GETDATE(), 7): 05 16, 06  
Select CONVERT(varchar(100), GETDATE(), 8): 10:57:46  
Select CONVERT(varchar(100), GETDATE(), 9): 05 16 2006 10:57:46:827AM  
Select CONVERT(varchar(100), GETDATE(), 10): 05-16-06  
Select CONVERT(varchar(100), GETDATE(), 11): 06/05/16  
Select CONVERT(varchar(100), GETDATE(), 12): 060516  
Select CONVERT(varchar(100), GETDATE(), 13): 16 05 2006 10:57:46:937  
Select CONVERT(varchar(100), GETDATE(), 14): 10:57:46:967  
Select CONVERT(varchar(100), GETDATE(), 20): 2006-05-16 10:57:47  
Select CONVERT(varchar(100), GETDATE(), 21): 2006-05-16 10:57:47.157  
Select CONVERT(varchar(100), GETDATE(), 22): 05/16/06 10:57:47 AM  
Select CONVERT(varchar(100), GETDATE(), 23): 2006-05-16  
Select CONVERT(varchar(100), GETDATE(), 24): 10:57:47  
Select CONVERT(varchar(100), GETDATE(), 25): 2006-05-16 10:57:47.250  
Select CONVERT(varchar(100), GETDATE(), 100): 05 16 2006 10:57AM  
Select CONVERT(varchar(100), GETDATE(), 101): 05/16/2006  
Select CONVERT(varchar(100), GETDATE(), 102): 2006.05.16  
Select CONVERT(varchar(100), GETDATE(), 103): 16/05/2006  
Select CONVERT(varchar(100), GETDATE(), 104): 16.05.2006  
Select CONVERT(varchar(100), GETDATE(), 105): 16-05-2006  
Select CONVERT(varchar(100), GETDATE(), 106): 16 05 2006  
Select CONVERT(varchar(100), GETDATE(), 107): 05 16, 2006  
Select CONVERT(varchar(100), GETDATE(), 108): 10:57:49  
Select CONVERT(varchar(100), GETDATE(), 109): 05 16 2006 10:57:49:437AM  
Select CONVERT(varchar(100), GETDATE(), 110): 05-16-2006  
Select CONVERT(varchar(100), GETDATE(), 111): 2006/05/16  
Select CONVERT(varchar(100), GETDATE(), 112): 20060516  
Select CONVERT(varchar(100), GETDATE(), 113): 16 05 2006 10:57:49:513  
Select CONVERT(varchar(100), GETDATE(), 114): 10:57:49:547  
Select CONVERT(varchar(100), GETDATE(), 120): 2006-05-16 10:57:49  
Select CONVERT(varchar(100), GETDATE(), 121): 2006-05-16 10:57:49.700  
Select CONVERT(varchar(100), GETDATE(), 126): 2006-05-16T10:57:49.827  
Select CONVERT(varchar(100), GETDATE(), 130): 18 ???? ?????? 1427 10:57:49:907AM  
Select CONVERT(varchar(100), GETDATE(), 131): 18/04/1427 10:57:49:920AM

常用：  
Select CONVERT(varchar(100), GETDATE(), 8): 10:57:46  
Select CONVERT(varchar(100), GETDATE(), 24): 10:57:47  
Select CONVERT(varchar(100), GETDATE(), 108): 10:57:49  
Select CONVERT(varchar(100), GETDATE(), 12): 060516  
Select CONVERT(varchar(100), GETDATE(), 23): 2006-05-16

**八、统计函数**

AVG （ ） -返回的平均价值   
count（ ） -返回的行数   
first（ ） -返回第一个值   
last（ ） -返回最后一个值   
max（ ） -返回的最大价值   
min（ ） -返回最小的价值   
total（ ） -返回的总和  
  
**九、数学函数**  
abs(numeric\_expr)                   求绝对值  
ceiling(numeric\_expr)                取大于等于指定值的最小整数  
exp(float\_expr)                      取指数  
floor(numeric\_expr)                   小于等于指定值得最大整数  
pi()                                  3.1415926.........  
power(numeric\_expr,power)            返回power次方         
rand([int\_expr])                     随机数产生器  
round(numeric\_expr,int\_expr)         安int\_expr规定的精度四舍五入  
sign(int\_expr)                      根据正数,0,负数,,返回+1,0,-1  
sqrt(float\_expr)                     平方根  
  
**十、系统函数**suser\_name()       用户登录名  
user\_name()        用户在数据库中的名字  
user               用户在数据库中的名字  
show\_role()        对当前用户起作用的规则  
db\_name()          数据库名                       
object\_name(obj\_id) 数据库对象名        
col\_name(obj\_id,col\_id) 列名          
col\_length(objname,colname)  列长度   
valid\_name(char\_expr)     是否是有效标识符