bit    整型   
bit数据类型是整型，其值只能是0、1或空值。这种数据类型用于存储只有两种可能值的数据，如Yes 或No、True 或False 、On 或Off.

**注意：很省空间的一种数据类型，如果能够满足需求应该尽量多用。**

tinyint   整型   
tinyint 数据类型能存储从0到255 之间的整数。它在你只打算存储有限数目的数值时很有用。这种数据类型在数据库中占用1 个字节.

**注意：如果bit类型太单调不能满足您的需求，您可以考虑用tinyint类型，因为这个类型相对也是比较安全的，不接受恶意脚本内容的嵌入。**

smallint     整型    
smallint 数据类型可以存储从- 2的15次幂(-32768)到2的15次幂(32767)之间的整数。这种数据类型对存储一些常限定在特定范围内的数值型数据非常有用。这种数据类型在数据库里占用2 字节空间.

**注意：如果tinyint类型太单调不能满足您的需求，您可以考虑用smallint类型，因为这个类型相对也是比较安全的，不接受恶意脚本内容的嵌入。**

int    整型   
int 数据类型可以存储从- 2的31次幂(-2147483648)到2的31次幂 (2147483 647)之间的整数。存储到数据库的几乎所有数值型的数据都可以用这种数据类型。这种数据类型在数据库里占用4个字节.

**注意：如果smallint也不能够满足您的需求，您可以考虑用长度更大的int类型。**  
  
decimal   精确数值型   
decimal 数据类型能用来存储从-10的38次幂-1到10的38次幂-1的固定精度和范围的数值型数据。使用这种数据类型时，必须指定范围和精度。 范围是小数点左右所能存储的数字的总位数。精度是小数点右边存储的数字的位数

numeric   精确数值型   
numeric数据类型与decimal 相似。  
  
smallmoney    货币型   
smallmoney 数据类型用来表示钱和货币值。这种数据类型能存储从-214748.3648 到214748.3647 之间的数据，精确到货币单位的万分之一

money   货币型   
money数据类型用来表示钱和货币值。这种数据类型能存储从-9220亿到9220 亿之间的数据，精确到货币单位的万分之一    
  
float    近似数值型              
float 数据类型是一种近似数值类型，供浮点数使用。说浮点数是近似的，是因为在其范围内不是所有的数都能精确表示。浮点数可以是从-1.79E+308到1.79E+308 之间的任意数   
  
real     近似数值型             
real 数据类型像浮点数一样，是近似数值类型。它可以表示数值在-3.40E+38到3.40E+38之间的浮点数

Smalldatetime   日期时间型            
smalldatetime 数据类型用来表示从1900年1月1日到2079年6月6日间的日期和时间，精确到一分钟

datetime   日期时间型            
datetime数据类型用来表示日期和时间。这种数据类型存储从1753年1月1日到9999年12月31日间所有的日期和时间数据， 精确到三百分之一秒或3.33毫秒 .   
  
cursor    特殊数据型   
cursor 数据类型是一种特殊的数据类型，它包含一个对游标的引用。这种数据类型用在存储过程中，而且创建表时不能用   
  
timestamp   特殊数据型            
timestamp 数据类型是一种特殊的数据类型，用来创建一个数据库范围内的唯一数码。 一个表中只能有一个timestamp列。每次插入或修改一行时，timestamp列的值都会改变。尽管它的名字中有“time”， 但timestamp列不是人们可识别的日期。在一个数据库里，timestamp值是唯一的    
  
Uniqueidentifier    特殊数据型            
Uniqueidentifier数据类型用来存储一个全局唯一标识符，即GUID。GUID确实是全局唯一的。这个数几乎没有机会在另一个系统中被重建。可以使用NEWID 函数或转换一个字符串为唯一  
标识符来初始化具有唯一标识符的列 .  
  
char    字符型                
char数据类型用来存储指定长度的定长非统一编码型的数据。当定义一列为此类型时，你必须指定列长。当你总能知道要存储的数据的长度时，此数据类型很有用。例如，当你按邮政编码加4个字符格式来存储数据时，你知道总要用到10个字符。此数据类型的列宽最大为8000 个字符.  
  
varchar   字符型               varchar数据类型，同char类型一样，用来存储非统一编码型字符数据。与char 型不一样，此数据类型为变长。当定义一列为该数据类型时，你要指定该列的最大长度。 它与char数据类型最大的区别是，存储的长度不是列长，而是数据的长度   .  
  
text    字符型                
text 数据类型用来存储大量的非统一编码型字符数据。这种数据类型最多可以有231-1或20亿个字符.  
  
nchar    统一编码字符型     
nchar 数据类型用来存储定长统一编码字符型数据。统一编码用双字节结构来存储每个字符，而不是用单字节(普通文本中的情况)。它允许大量的扩展字符。此数据类型能存储4000种字符，使用的字节空间上增加了一倍.  
  
nvarchar    统一编码字符型      
nvarchar 数据类型用作变长的统一编码字符型数据。此数据类型能存储4000种字符，使用的字节空间增加了一倍.  
  
ntext    统一编码字符型      
ntext 数据类型用来存储大量的统一编码字符型数据。这种数据类型能存储230 -1或将近10亿个字符，且使用的字节空间增加了一倍  
  
binary    二进制数据类型     
binary数据类型用来存储可达8000 字节长的定长的二进制数据。当输入表的内容接近相同的长度时，你应该使用这种数据类型.  
  
varbinary     二进制数据类型     
varbinary 数据类型用来存储可达8000 字节长的变长的二进制数据。当输入表的内容大小可变时，你应该使用这种数据类型    
  
image    二进制数据类型      
image 数据类型用来存储变长的二进制数据，最大可达231-1或大约20亿字节    
  
  
  
－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－  
(1)二进制数据类型   
　　二进制数据包括 Binary、Varbinary 和 Image  
　　Binary 数据类型既可以是固定长度的(Binary),也可以是变长度的。  
　　Binary[(n)] 是 n 位固定的二进制数据。其中，n 的取值范围是从 1 到 8000。其存储窨的大小是 n + 4 个字节。  
　　Varbinary[(n)] 是 n 位变长度的二进制数据。其中，n 的取值范围是从 1 到 8000。其存储窨的大小是 n + 4个字节，不是n 个字节。  
　　在 Image 数据类型中存储的数据是以位字符串存储的，不是由 SQL Server 解释的，必须由应用程序来解释。例如，应用程序可以使用BMP、TIEF、GIF 和 JPEG 格式把数据存储在 Image 数据类型中。   
  
(2)字符数据类型   
  
　　字符数据的类型包括 Char，Varchar 和 Text  
　　字符数据是由任何字母、符号和数字任意组合而成的数据。  
　　Varchar 是变长字符数据，其长度不超过 8KB。Char 是定长字符数据，其长度最多为 8KB。超过 8KB 的ASCII 数据可以使用Text数据类型存储。例如，因为 Html 文档全部都是 ASCII 字符，并且在一般情况下长度超过 8KB，所以这些文档可以 Text 数据类型存储在SQL Server 中。   
  
(3)Unicode 数据类型   
  
　　Unicode 数据类型包括 Nchar,Nvarchar 和Ntext  
　　在 Microsoft SQL Server 中，传统的非 Unicode 数据类型允许使用由特定字符集定义的字符。在 SQL Server安装过程中，允许选择一种字符集。使用 Unicode 数据类型，列中可以存储任何由Unicode 标准定义的字符。在 Unicode 标准中，包括了以各种字符集定义的全部字符。使用Unicode数据类型，所战胜的窨是使用非 Unicode 数据类型所占用的窨大小的两倍。   
　　在 SQL Server 中，Unicode 数据以 Nchar、Nvarchar 和 Ntext 数据类型存储。使用这种字符类型存储的列可以存储多个字符集中的字符。当列的长度变化时，应该使用Nvarchar 字符类型，这时最多可以存储 4000 个字符。当列的长度固定不变时，应该使用 Nchar 字符类型，同样，这时最多可以存储4000 个字符。当使用 Ntext 数据类型时，该列可以存储多于 4000 个字符。   
  
(4)日期和时间数据类型   
  
　　日期和时间数据类型包括 Datetime 和 Smalldatetime两种类型  
　　日期和时间数据类型由有效的日期和时间组成。例如，有效的日期和时间数据包括“4/01/98 12:15:00:00:00 PM”和“1:28:29:15:01AM 8/17/98”。前一个数据类型是日期在前，时间在后一个数据类型是霎时间在前，日期在后。在 Microsoft SQL Server中，日期和时间数据类型包括Datetime 和 Smalldatetime 两种类型时，所存储的日期范围是从 1753 年 1 月 1 日开始，到9999 年12 月 31 日结束(每一个值要求 8 个存储字节)。使用 Smalldatetime 数据类型时，所存储的日期范围是 1900年 1 月 1日 开始，到 2079 年 12 月 31 日结束(每一个值要求 4 个存储字节)。  
　　日期的格式可以设定。设置日期格式的命令如下：  
　　Set DateFormat {format | @format \_var|  
　　其中，format | @format\_var 是日期的顺序。有效的参数包括 MDY、DMY、YMD、YDM、MYD 和 DYM。在默认情况下，日期格式为MDY。  
　　例如，当执行 Set DateFormat YMD 之后，日期的格式为年 月 日 形式；当执行 Set DateFormat DMY 之后，日期的格式为日 月有年 形式   
  
（5）数字数据类型   
  
　　数字数据只包含数字。数字数据类型包括正数和负数、小数（浮点数）和整数  
　　整数由正整数和负整数组成，例如 39、25、0-2 和 33967。在 Micrsoft SQL Server 中，整数存储的数据类型是 　　 Int，Smallint和 Tinyint。Int 数据类型存储数据的范围大于 Smallint 数据类型存储数据的范围，而 Smallint 据类型存储数据的范围大于Tinyint 数据类型存储数据的范围。使用 Int 数据狗昔存储数据的范围是从 -2 147 483 648 到 2 147 483 647（每一个值要求4个字节存储空间）。使用 Smallint 数据类型时，存储数据的范围从 -32 768 到 32 767（每一个值要求2个字节存储空间）。使用Tinyint 数据类型时，存储数据的范围是从0 到255（每一个值要求1个字节存储空间）。  
　　精确小娄数据在 SQL Server 中的数据类型是 Decimal 和 Numeric。这种数据所占的存储空间根据该数据的位数后的位数来确定。  
　　在SQL Server 中，近似小数数据的数据类型是 Float 和 Real。例如，三分之一这个分数记作。3333333，当使用近似数据类型时能准确表示。因此，从系统中检索到的数据可能与存储在该列中数据不完全一样。   
  
（6）货币数据表示正的或者负的货币数量 。   
  
　　在 Microsoft SQL Server 中，货币数据的数据类型是Money 和 Smallmoney   
　　Money数据类型要求 8 个存储字节，Smallmoney 数据类型要求 4 个存储字节。   
  
（7）特殊数据类型   
  
　　特殊数据类型包括前面没有提过的数据类型。特殊的数据类型有3种，即 　　 Timestamp、Bit 和 Uniqueidentifier。  
　　Timestamp 用于表示SQL Server 活动的先后顺序，以二进投影的格式表示。Timestamp 数据与插入数据或者日期和时间没有关系。  
　　Bit 由 1 或者 0 组成。当表示真或者假、ON 或者 OFF 时，使用 Bit 数据类型。例如，询问是否是每一次访问的客户机请求可以存储在这种数据类型的列中。  
　　Uniqueidentifier 由 16 字节的十六进制数字组成，表示一个全局唯一的。当表的记录行要求唯一时，GUID是非常有用。例如，在客户标识号列使用这种数据类型可以区别不同的客户。   
  
2.用户定义的数据类型   
  
　　用户定义的数据类型基于在 Microsoft SQL Server 中提供的数据类型。当几个表中必须存储同一种数据类型时，并且为保证这些列有相同的数据类型、长度和可空性时，可以使用用户定义的数据类型。例如，可定义一种称为 　　postal\_code 的数据类型，它基于 Char 数据类型。  
　　当创建用户定义的数据类型时，必须提供三个数：数据类型的名称、所基于的系统数据类型和数据类型的可空性。   
  
（1）创建用户定义的数据类型   
  
　　创建用户定义的数据类型可以使用 Transact-SQL 语句。系统存储过程 sp\_addtype 可以来创建用户定义的数据类型。其语法形式如下：  
　　sp\_addtype {type},[,system\_data\_bype][,'null\_type']  
　　其中，type 是用户定义的数据类型的名称。system\_data\_type 是系统提供的数据类型，例如 Decimal、Int、Char 　　等等。 null\_type 表示该数据类型是如何处理空值的，必须使用单引号引起来，例如'NULL'、'NOT NULL'或者'NONULL'。  
　　例子：  
　　Use cust  
　　Exec sp\_addtype ssn,'Varchar(11)',"Not Null'  
　　创建一个用户定义的数据类型 ssn，其基于的系统数据类型是变长为11 的字符，不允许空。  
　　例子：  
　　Use cust  
　　Exec sp\_addtype birthday,datetime,'Null'  
　　创建一个用户定义的数据类型 birthday，其基于的系统数据类型是 DateTime，允许空。  
　　例子：  
　　Use master  
　　Exec sp\_addtype telephone,'varchar(24),'Not Null'  
　　Eexc sp\_addtype fax,'varchar(24)','Null'  
　　创建两个数据类型，即 telephone 和 fax   
  
（2）删除用户定义的数据类型   
  
　　当用户定义的数据类型不需要时，可删除。删除用户定义的数据类型的命令是 sp\_droptype {'type'}。  
　　例子：  
　　Use master  
　　Exec sp\_droptype 'ssn'  
　　注意：当表中的列还正在使用用户定义的数据类型时，或者在其上面还绑定有默认或者规则时，这种用户定义的数据类型不能删除。   
  
SQL SERVER的字段类型说明   
  
　　以下为SQL SERVER7.0以上版本的字段类型说明。SQL SERVER6.5的字段类型说明请参考SQL SERVER提供的说明。  
  
字段类型 描述   
bit   0或1的整型数字   
int   从-2^31(-2,147,483,648)到2^31(2,147,483,647)的整型数字   
smallint   从-2^15(-32,768)到2^15(32,767)的整型数字   
tinyint   从0到255的整型数字   
      
decimal   从-10^38到10^38-1的定精度与有效位数的数字   
numeric   decimal的同义词   
      
money   从-2^63(-922,337,203,685,477.5808)到2^63-1(922,337,203,685,477.5807)的货币数据，最小货币单位千分之十   
smallmoney   从-214,748.3648到214,748.3647的货币数据，最小货币单位千分之十   
      
float   从-1.79E+308到1.79E+308可变精度的数字   
real   从-3.04E+38到3.04E+38可变精度的数字   
      
datetime   从1753年1月1日到9999年12日31的日期和时间数据，最小时间单位为百分之三秒或3.33毫秒   
smalldatetime   从1900年1月1日到2079年6月6日的日期和时间数据，最小时间单位为分钟   
      
timestamp   时间戳，一个数据库宽度的唯一数字   
uniqueidentifier   全球唯一标识符GUID   
      
char   定长非Unicode的字符型数据，最大长度为8000   
varchar   变长非Unicode的字符型数据，最大长度为8000   
text   变长非Unicode的字符型数据，最大长度为2^31-1(2G)   
      
nchar   定长Unicode的字符型数据，最大长度为8000   
nvarchar   变长Unicode的字符型数据，最大长度为8000   
ntext   变长Unicode的字符型数据，最大长度为2^31-1(2G)   
      
binary   定长二进制数据，最大长度为8000   
varbinary   变长二进制数据，最大长度为8000   
image   变长二进制数据，最大长度为2^31-1(2G)   
  
  
－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－－   
Bit   
     1位，值为0或1   
     Int   
     Integer   
     4字节，值为-2^31~2^31-1   
     Smallint   
     2字节，值为-2^15~2^15-1   
     Tinyint   
     1字节，值为0~255   
     Decimal (p,s)   
     数字数据，固定精度为P，宽度为S   
     Numeric   
     Money   
     8字节，存放货币类型，值为-2^63~2^63-1   
     Small money   
     4字节，存放货币类型，值为-214748.3648~+214748.3647近似数值数据类型   
     Float (n)   
     N在1~24之间，4字节，7位精度   
     N=1~7为real   
     N在25~53之间，8字节，15位精度   
     =8~15为float   
Datetime   
     8字节，描述某天的日期和时刻，值的精确度为1/300秒   
     Smalldatetime   
     4字节，描述某天的日期和时刻，精度为分钟   
     Cursor    
     对游标的引用   
     Timestamp   
     8字节，存放在数据库内唯一的数据   
     Uniqueidentifier   
     16字节，存放全局唯一标识（GUID）   
     Char (n)   
     非unicode字符串的固定长度，n=1~8000   
     Character (n)  
Varchar (n)   
     可变长度，非unicode字符串n=1~8000   
     Char varying(n)   
Text   
     服务器代码页中可变长度非unicode数据。最大长度为231-1个字符   
     Nchar   
     固定长度unicode字符串n=1~4000   
     National character (n),   
     National char(n)   
Nvarchar   
     固定长度unicode字符串n=1~4000   
     National character varying(n)   
Ntext   
     可变长度unicode数据，最大长度为230-1个字符   
     National text   
Binary (n)   
     固定长度二进制数据，n在1~8000之间，存储空间为n+4字节   
     Varbinary (n)   
     可变长度二进制数据，n=1~8000   
     Binary varying (n)   
Tmage   
     可变长度二进制数据，大小为0~231-1   
     注意：   
     1） 对于数值型数据类型，宽度（scale）是指存储在小数点后的数字位数，而精度（precision）是指能存储的包含小数点在内的所有数字位数。   
     2） money和small money的存储宽度为4。   
     3） 时间戳列值在每一行更新时系统自动更新，时间戳列不能是关键字或关键字的一部分。   
     4） 唯一标识数据类型不能使用算术操作符（如+、-等），这种数据类型只能使用相等比较操作。Unicode是所有字符集的一致存储数据的标准。它要使用两倍于非Unicode数据存储的存储空间。

http://www.cnblogs.com/andy\_tigger/archive/2011/08/21/2147745.html