<http://www.cnblogs.com/kissdodog/archive/2013/03/09/2951523.html>

[**SQL Server聚合函数**](http://www.cnblogs.com/kissdodog/archive/2013/03/09/2951523.html)

　　聚合函数对一组值计算后返回单个值。除了count(统计项数)函数以外，其他的聚合函数在计算式都会忽略空值(null)。所有的聚合函数均为确定性函数。即任何时候使用一组相同的输入值调用聚合函数执行后的返回值都是相同的，无二义性。T-SQL提供的聚合函数一共有13个之多。

　　聚合函数通常会在下列场合使用：

　　　1、select语句的选择列表，包括子查询和外部查询。

　　　2、使用compute或compute by产生汇总列时。

　　　3、having子句对分组的数据记录进行条件筛选。

# 聚合函数

**1、平均值AVG**

　AVG函数用于计算精确型或近似型数据类型的平均值，bit类型除外，忽略null值。AVG函数计算时将计算一组数的总和，然后除以为null的个数，得到平均值。

　语法结构：

　　avg( [ all | distinct ] expression )

　all：为默认值，表示对所用的数据都计算平均值。

　distinct：每个值的唯一值计算平均值，不管相同的值出现多次，多个行相同的值仅仅出现一次作为计算。

　expression：精确或近似值的表达式。表达式内部不允许使用子查询和其他聚合函数。

　示例：

　　select avg(distinct age) from person -- 查询person表里的年龄的平均值，相同值只计算一次

**2、最小值MIN**

　　　　MIN函数用于计算最小值，MIN函数可以适用于numeric、char、varchar或datetime、money或smallmoney列，但不能用于bit列。不允许使用聚合函数和子查询，忽略null值。

　　　　语法结构：

　　min( [ all | distinct ] expression )

　　　　示例：

　　select min(age) from person -- 查询person表里的年龄的最小值

**3、最大值MAX**

　　　MAX函数用于计算最大值，忽略null值。max函数可以使用于numeric、char、varchar、money、smallmoney、或datetime列，但不能用于bit列。不允许使用聚合函数和子查询。

　　　语法结构：

MAX ( [ all | distinct ] expression )

　　　示例：

　　　　select max(age) from person -- 查询person表里的年龄的最大值

**4、求和值SUM**

　　　SUM函数用于求和，只能用于精确或近似数字类型列(bit类型除外)，忽略null值，不允许使用聚合函数和子查询。

　　　1、语法结构：

　　SUM ( [ all | distinct ] expression )

　　　　示例:

　　　select sum(age) from person -- 查询person表里的年龄的总和

**5、统计项数值count(count\_big)**

　　　　count函数用于计算满足条件的数据项数，返回int数据类型的值。

　　　　1、语法结构：

　　count( {[[ all | distinct] expression ] | \* } )

　　这里的表达式是除text、image或ntext以外任何数据类型的表达式。但不允许使用聚合函数和子查询。

　　　　2、常见使用方法

　　　　　　count(\*) ： 返回所有的项数，**包括null值**和重复项。而除了count(\*)外，其他任何形式的count()函数都会忽略Null行。除了Count(\*)函数外，其他任何聚合函数都会忽略NULL值，也就是说，AVG()参数里的值如果为Null则这一行会被忽略如计算平均值，这点要注意。

　　　　　　count(all表达式)：返回非空的项数。

　　　　　　count(distinct表达式)：返回唯一非空的项数

　　　　　　count\_big的语法与count完全一样，只不过返回值为bigint数据类型，这样返回的数值范围就可以大大超过count。

　　　　注意：count(字段名)，如果字段名为NULL，则count函数不会统计。例如count(name)，如果name为空，则不会统计到结果。

　　　　示例：

　　　select count(distinct age) from person -- 查询person表里的年龄唯一且非空的项数

**6、计算标准偏差值STDEV**

　　　语法结构：

　　STDEV( [all | distinct ] expression )

　　　　这里的expression必须是一个数值表达式，不允许使用聚合函数和子查询。表达式的值是精确或近似数值类型，但不包括bit数据类型。将忽略null值。

　　　　标准偏差是高中的东西，忘记干什么用的了，用到的时候再学回来吧。

　　　　示例：

　　select stdev(age) from person -- 查询person表里的年龄的标准偏差

**7、计算方差VAR**

　　　VAR函数用于计算指定表达式中所有值的方差。

　　　语法结构：

　　VAR( [ all | distinct ] expression )

　　　这里的expression表达式必须是一个数值表达式，不允许使用聚合函数和子查询。表达式的值是精确或近似数值类型，但不包括bit数据类型，将忽略null值。

　　　方差也是高中学的东西，忘了。

　　　示例：

　　select var(age) from person -- 查询person表里的年龄的方差

**8、CHECKSUM\_AGG**

　　返回组中各值的校验和。 将忽略 Null 值。CHECKSUM\_AGG 可用于检测表中的更改。表中行的顺序不影响 CHECKSUM\_AGG 的结果。此外，CHECKSUM\_AGG 函数还可与 DISTINCT 关键字和 GROUP BY 子句一起使用。如果表达式列表中的某个值发生更改，则列表的校验和通常也会更改。但只在极少数情况下，校验值会保持不变。

　　语法如下：

　　CHECKSUM\_AGG ( [ ALL | DISTINCT ] expression )

　　参数说明：

　　ALL：对所有的值进行聚合函数运算。 ALL 为默认值。

　　DISTINCT ：指定 CHECKSUM\_AGG 返回唯一校验值。

　　expression ：一个整数表达式。 不允许使用聚合函数和子查询。

SELECT CHECKSUM\_AGG(Account\_Age) FROM Account

GO

UPDATE Account SET Account\_Age = **30** WHERE Account\_Id = **6**

GO

SELECT CHECKSUM\_AGG(Account\_Age) FROM Account

　　显示结果如下：

  
　　可见随着表的更改，该系统函数返回的值也变了。此函数的作用正在于此，检测表的更改。

**9、COUNT\_BIG**

 　　返回组中的项数。 COUNT\_BIG 的用法与 COUNT 函数类似。 两个函数唯一的差别是它们的返回值。 COUNT\_BIG 始终返回 bigint 数据类型值。 COUNT 始终返回 int 数据类型值。

　　10、STDEVP  
　　11、GROUPING  
　　12、GROUPING\_ID  
　　13、VARP