

表达式生成器中的几种常用符号 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/134/2021_2022__E8_A1_A8_E8_BE_BE_E5_BC_8F_E7_c97_134952.htm “ ”、“ - ”、“ * ”、“ / ”这几个符号就代表数学运算中的“加”、“减”、“乘”、“除”四种运算符号。使用“amp."使用指南"将返回：“数据库使用指南”，也就是将这两个字符串连接在一起，左面的字符串在前面，右面的字符串在后面。如下图所示。

“ = ”、“ > ”、“ ”这四个符号分别表示“等于”、“大于”、“小于”、“不等于”，它们都是用来判断某个条件是否为满足，例如：“ =34 ”表示当某个值等于34时才算满足这个条件。“ "北京" ”表示当某个值不等于字符串“北京”时才算满足了条件。“ and ”、“ or ”、“ not ”这三个逻辑运算符是用来连接上面的这些条件表达式的。比如：“ >100 and 100 or 100 ”这个表达式则表示只要这个值不大于100，这个条件就算满足了。还有一个“ like ”，这个符号又怎么用呢？这个符号常常用在对一个字符型的值进行逻辑判断，是否这个值满足某种格式类型。所以通常“ like ”并不单独使用，常常还要跟一些别的符号：“ ? ”表示任何单一字符；“ * ”表示零个或多个字符；“ # ”表示任何一个数字；“ [字符列表] ”表示字符列表中的任何单一字符；“ [!字符列表] ”表示不在字符列表中的任何单一字符。我们一起看几个例子， like "中国?" 则字符串“中国人”、“中国字”都满足这个条件； like "中国*" 则字符串“中国”、“中国人”、“中国人民银行”这些都满足这个条件； like "表#" 则字符串“表1”、“表2”都满足这个条件； like "[北京,上海,广

州]" 则只有字符串 “北京”、“上海”、“广州” 可以满足条件；like "[!北京,上海,广州]" 则只有字符串 “北京”、“上海”、“广州” 不能满足条件。asc(字符串表达式) 代表字符串中首字母的字符代码，返回一个 integer，。但是这个字符串表达式不能没有，不然会出错。例如：asc("a") 返回 65 asc("a") 返回 97 asc("apple") 返回 65。chr(字符码) 使用 chr 函数来返回指定所代表的字符。0 到 31 之间的数字与标准的非打印 ascii 代码相同。字符码的正常范围为 0 255。chrb 函数作用于包含在 string 中的字节数据。chrb 总是返回一个单字节，而不是返回一个字符，一个字符可能是一个或两个字节。chrw 函数返回包含 unicode 的 string，若在不支持 unicode 的平台上，则其功能与 chr 函数相同。例子：chr(65) 返回 a chr(97) 返回 a。davg (数值数据字段, 表或查询) 可用于计算特定记录集中（一个域）一组数的平均值；例如：davg(产品单价, 产品信息表) 返回 “产品信息表” 中的 “产品单价” 的平均值。dcount(数值数据字段, 表或查询) 可用于确定特定记录集中的记录数（一个域）；例如：dcount (产品单价, 产品信息表) 返回 “产品信息表” 中的 “产品单价” 中的记录数。dlookup(数值数据字段, 表或查询) 用于从指定记录集（一个域）获取特定字段的值；例如：dlookup (产品单价, 产品信息表) 返回 “产品信息表” 中的 “产品单价” 中的值。dmin(数值数据字段, 表或查询)；dmax(数值数据字段, 表或查询)；dmin 和 dmax 函数用于确定指定记录集（一个域）中的最小值和最大值；例如：dmin(产品单价, 产品信息表) 返回 “产品信息表” 中的 “产品单价” 中的最小值。dsum(数值数据字段, 表或查询) 用于计算指定记录集（一个域）中的一组值的

总和；例如：`dmin(产品单价, 产品信息表)` 返回“产品信息表”中的“产品单价”中的和。`dvar(数值数据字段, 表或查询)` `dvarp(数值数据字段, 表或查询)` 用于估算特定记录集（一个域）中一组值的方差，`dvarp`用于计算总体的方差，而 `dvar` 函数用于计算总体抽样的方差；如果引用的记录少于两条，`dvar` 和 `dvarp` 函数将返回 `null`，表示不能计算方差。数值数据字段 可以是用来标识表或查询中字段的字符串表达式，也可以是执行在域合计函数中计算字段的表达式。在 `expr` 中可以使用表中字段的名称、窗体上的控件、常量或函数。函数可以是内置的，也可以是用户自定义的，但不能是另一个域合计函数或 `sql` 合计函数。表或查询：用来标识组成域的记录集的字符串表达式。`ddb(cost, salvage, life, period[, factor])` 指定一笔资产在一特定期间的折旧，可使用双下落收复平衡方法或其它指定的方法进行计算。`cost` 必要。`double` 指定资产的初始成本。`salvage` 必要。`double`.指定使用年限结束时的资产价值。`life` 必要。`double` 指定资产可用的可用年限。`period` 必要。`double` 指定计算资产折旧所用的那一期间。`factor` 可选。`variant` 指定收复平衡下落时的速度。如果省略的话，缺省值为2（双下落方法）。`fv(rate, nper, pmt[, pv[, type]])` 返回一个 `double`，指定未来的定期定额支付且利率固定的年金。`rate` 必要。`double`，指定每一期的利率。例如，如果有一笔贷款年百分率 (`apr`) 为百分之十且按月付款的汽车贷款，则利率为 $0.1/12$ 或 0.0083 。`nper` 必要。`integer`，指定一笔年金的付款总期限。例如，如果对一笔为期四年的汽车贷款选择按月付款方式，则贷款期限共有 $4 * 12$ （或 48）个付款期。

100Test 下载频道开通，各类考试题目直接下载。详细请访

