语义模型\_预置函数

by边传猛

## 语义函数

|  |  |  |
| --- | --- | --- |
| **函数名称** | **参数** | **功能** |
| **smart**(code) | ***code***:语义模型编码 | 根据语义模型的编码获取到此模型的查询结果 |
| **parameter**(*paramCode*) | ***paramCode***:参数编码 | 根据参数编码，获取参数值 |
| **macro**(*macroCode*) | ***macrocode***:宏变量编码 | 根据宏变量编码，获取参数值 |
| **nc\_metadata**(*entitypath, attributePaths*) | ***entitypath***:实体全路径名***attributePaths***:属性路径列表 | 根据实体路径和属性路径列表获取对应的SQL |
| **meta** (*id*) | ***id***:元定义GUID | 返回该元定义加载数据的sql语句 |
| **Multicolumn(column)** | ***Column:字段表达式*** | 根据当前语种返回相应的多语字段。语法：multicolumn(表别名.字段名) 或 multicolumn(字段名) |
| **Sql(‘sql’)** | ***Sql:入参为标准sql语句*** | sql(' ')函数，入参为标准sql语句，返回该sql语句。  注意：入参使用单引号'引起；如果标准sql中有单引，必须替换为两个单引，即：'替换为''。  另：sql中支持使用参数、宏变量。 |
| **freedef(column)** | ***Column:字段表达式*** | 自定义项函数。将自定义项转档案名称。语法：freedef(表别名.字段名) |
| **ncmdenum(column,枚举类型ID)** | ***Column:字段表达式***  **枚举类型ID: NC元数据组件中定义的枚举类型ID值** | 元数据枚举属性函数。根据枚举属性的 值 取显示名 |
| **timeoffset(timeExp)** | **timeExp:时间类型字段 或 时间戳常量值** | 时区偏移函数。对时间类型数据进行时区差异处理 |
| **multicolumn(column)** | ***Column:字段表达式*** | 多语字段函数。根据当前语种返回相应的多语字段 |

## 其他函数

在使用下面三类函数时，要注意其函数参数的类型以及格式。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **类别** | **函数名称** | **参数** | **功能** | **举例** |
| **字符函数** | **concat**(*arg1*,*args2*) | arg1:拼接字段1；arg2:拼接字段2。arg1,arg2必须是字符类型。 | 拼接arg1,arg2 | concat(name,code) |
| **indexof**(*arg1*,*arg2*,*arg3*) | arg1:被索引字段；arg2:索引子串；arg3:索引起始位置。arg1, arg2为字符类型，arg2为数字。 | 从第arg3的位置起，返回字段arg1中子串arg2第一次出现的位置。 | Indexof(name,’abc’,1)  注意：这里从1开始 |
| **length**(*arg1*) | arg1:被求长字段。 | 获取字段arg1的长度 | length(name) |
| **nvl**(*arg1,arg2*) | arg1:被判空字段；arg2:替换值 | 判断arg1的value是否为null，如果是，则替换为arg2 |  |
| **substr**(*arg1*,*arg2*,*arg3*) | arg1:被截取字段；arg2:起始位置；arg3:截取长度。arg2,arg3为整数型 | 截取arg1,从arg2开始截取arg3长度的子字符串。arg2取值从1开始 | substr(field,1,2) |
| **tostring**(*arg*) | arg:被转换字段 | 使arg转换为字符串类型。 |  |
| **日期函数** | **date**() |  | 获取当前系统时间日期 |  |
| **datediff**(*unit*,*date1*,*date2*) | unit:时间单位；date1:日期1,date2:日期2。其中date1,date2为字符串型，例如’20100101’,’2010-01-01’ | 根据指定的时间单位(年-‘yy’,月-'mm',天-'dd',)，计算两日期之间某部分的差值(date2-date1) | datadiff(‘mm’,substr(toString(field1),0,10), substr(toString(field2),0,10)) |
| **datefmt**(*date*, *pattern*) | date:日期格式的字符型字段，例如’20100101’,’2010-01-01’；pattern:格式化类型 | 用于将时间格式化为期望的字符串,其中pattern为格式化参数,yyyy-年,MM-月,dd-天。支持的格式类型有：yyyy-mm-dd, mm/dd/yyyy, yyyymmdd, dd.mm.yyyy | substr(datefmt(20100101,'yyyy-mm-dd'),0,10)结果为2010-01-01 |
| **todate**(*arg*) | arg:被转换字符串, 例如’20100101’,’2010-01-01’。 | 转换字符串为统一日期类型 | todate(‘20081001’) |
| **dateAdd(fieldchar, num, date)** | **fieldchar：时间单位**  **num：增加的数值**  **date：日期** | 返回在指定日期的年、月或者日上增加某个值num,可增加的时间域fieldchar包括'yy'-增加年;'mm'-增加月;'dd'-增加日. | dateAdd('dd',1,todate('2010-03-10'))表示对前面的日期增加一天 |
| **数学函数** | **ceil(***arg***)** | arg被求值字段 | 返回大于或等于arg的最小整数 |  |
| **max(***arg***)** | 同上 | 获取arg字段中的最大值 |  |
| **min(***arg***)** | 同上 | 获取arg字段中的最小值 |  |
| **round(***arg1,arg2***)** | arg1被求值字段；arg2舍入位 | 对arg1字段值的第arg位执行四舍五入 |  |
| **avg(arg)** | arg:数值类型字段，或 数值 | 获取参数(arg)字段的平均值,其中参数为数值类型。 |  |
| **int(arg1)** | arg1:整数的字符形式 或 浮点数 | 转整形函数 |  |
| **mod(arg1,arg2)** | arg1 被除数,arg2 除数 | 求余运算函数 |  |
| **sum(arg)** | arg:数值型字段 或 数值 | 获取参数(arg)字段的合计值,其中参数为数值类型 |  |