目录

[目录 1](#_Toc488431035)

[第一章 关键字 3](#_Toc488431036)

[关键字简介 3](#_Toc488431037)

[关键字详述 4](#_Toc488431038)

[GET\_CONN 4](#_Toc488431039)

[GET\_STAT 4](#_Toc488431040)

[GET\_PSTAT 4](#_Toc488431041)

[GET\_CSTAT 4](#_Toc488431042)

[SQL 5](#_Toc488431043)

[PSQL 5](#_Toc488431044)

[CSQL 5](#_Toc488431045)

[TX 6](#_Toc488431046)

[VERIFY 6](#_Toc488431047)

[SLEEP 6](#_Toc488431048)

[ST 6](#_Toc488431049)

[IMPORT\_DBI 6](#_Toc488431050)

[CLEAR\_DBI 6](#_Toc488431051)

[IMPORT\_IRS 7](#_Toc488431052)

[SYS 7](#_Toc488431053)

[DEF\_PROC 7](#_Toc488431054)

[CONTAIN 7](#_Toc488431055)

[if else 7](#_Toc488431056)

[while 7](#_Toc488431057)

[## 7](#_Toc488431058)

[ERROR 7](#_Toc488431059)

[CAL 7](#_Toc488431060)

[INDEX\_OF 8](#_Toc488431061)

[GRD 8](#_Toc488431062)

[第二章 数据类型 9](#_Toc488431063)

[数据类型 9](#_Toc488431064)

[数据类型详述 9](#_Toc488431065)

[基本数据类型 9](#_Toc488431066)

[Decimal 10](#_Toc488431067)

[List<*var\_category*> 10](#_Toc488431068)

[Connection 10](#_Toc488431069)

[Statement 10](#_Toc488431070)

[PStatement 10](#_Toc488431071)

[CStatement 10](#_Toc488431072)

[ResultSet 10](#_Toc488431073)

[IdealResultSet<*var\_category*> 10](#_Toc488431074)

[第三章 系统功能关键字 12](#_Toc488431075)

[SLEEP 12](#_Toc488431076)

[SYS 12](#_Toc488431077)

[is\_cluster\_available 12](#_Toc488431078)

[await\_available 13](#_Toc488431079)

[merge 13](#_Toc488431080)

[minor\_freeze 13](#_Toc488431081)

[is\_merge\_done 13](#_Toc488431082)

[await\_merge\_done 13](#_Toc488431083)

[reelect 13](#_Toc488431084)

[exist\_master 13](#_Toc488431085)

[kill\_server 13](#_Toc488431086)

[start\_server 14](#_Toc488431087)

[add\_server 14](#_Toc488431088)

[set\_master 15](#_Toc488431089)

[shell\_command 15](#_Toc488431090)

# 关键字

## 关键字简介

表2.1 关键字

|  |  |
| --- | --- |
| 关键字 | 说明 |
| GET\_CONN[ *ms\_category* ] | 获取MS连接 |
| GET\_STAT[ *connection\_name* ] | 获取SQL执行器 |
| GET\_PSTAT[ *connection\_name; “sql”; data\_type* ] | 获取预编译SQL执行器 |
| GET\_CSTAT[ *connection\_name; “sql”; data\_type* ] | 获取存储过程SQL执行器 |
| SQL[ *“sql”; stat\_name; type* ] | SQL语句 |
| PSQL[ *pstat\_name; type; variables|value* ] | 预编译SQL语句 |
| CSQL[ *cstat\_name; type; variable|value param\_type* ] | 存储过程SQL语句 |
| TX[ *connection\_name; tx\_id* ] | 事务 |
| VERIFY[ *variable1; variable2; relation\_operator* [; *set\_type*] ] | 验证结果集 |
| SLEEP[ *number; time\_unit* ] | 暂停时间 |
| ST[ *number1; number2; clean* / *remain; synchronous* / *asynchronous* ] | stress test压力测试 |
| IMPORT\_DBI[ *“dbi\_name”* ] | 导入测试场景 |
| CLEAR\_DBI[ *“dbi\_name”* ] | 清空测试场景 |
| IMPORT\_IRS[ *“result\_name”;* (*variable integer*)+ ] | 结果集导入 |
| SYS[ *sys\_statement; ip\_address* ] | 系统功能 |
| DEF\_PROC[ *str\_variable; stat\_name* ] | 定义存储过程 |
| CONTAIN[ *str\_variable|“string”; str\_variable|“string”; variable|number*] | 检测源字符串中是否有指定数量的字符串匹配 |
| if( *condition* ){} else{} | 分支 |
| while( *number* ){} | 循环 |
| ## | 注释 |
| ERROR | 此案例错误 |
| CAL[ *expression* ] | 计算表达式 |
| INDEX\_OF[ *variable*; [*row*, *column*]; *data\_type* ] | 获取指定位置的数据 |
| GRD[ “sql”; data\_types; rows ] | 生成指定数量的随机数据 |

## 关键字详述

### GET\_CONN

GET\_CONN[ *ms\_category* ]表示获取一个JDBC连接（Connection），*ms\_category*值有两种，值为master表示在主集群MS上建立连接，值为slave表示在备集群merge server上建立连接。与数据类型Connection搭配，形式为：

Connection connection\_name = GET\_CONN[ *ms\_category* ];

### GET\_STAT

GET\_STAT[ *connection\_name* ]表示在指定连接上获取一个SQL执行器，执行器指JDBC中的Statement，*connection\_name*是GET\_CONN返回的某个连接。与数据类型Statement搭配，形式为：

Statement stat\_name = GET\_STAT[ *connection\_name* ];

### GET\_PSTAT

GET\_PSTAT[ *connection\_name; “sql”* ; *(data\_type)\** ]表示在指定连接上获取一个预编译SQL执行器（PreparedStatement），*connection\_name*是GET\_CONN返回的某个连接，*sql*是指预编译的SQL语句，*sql*必须用双引号包含。与PStatement数据类型搭配，形式为：

PStatement pstat\_name = GET\_PSTAT[ *connection\_name; “sql”; data\_type* ];

*data\_type*表示此预编译语句中参数的数据类型，多个参数类型之间以逗号分隔，须与参数一一对应。

例如：

PStatement pstat = GET\_PSTAT[ conn; “insert into t1 values(?,?)”; int, boolean ];

### GET\_CSTAT

GET\_CSTAT[ *connection\_name; “sql”* ; *(data\_type)\** ]表示在指定连接上获取一个存储过程执行器（CallableStatement）。*connection\_name*是GET\_CONN返回的某个连接的名字，*sql*是存储过程，必须用双引号包含。必须与[3.1节](#_3.1_数据类型)中的CStatement数据类型搭配使用，形式为：

CStatement cstat\_name = GET\_CSTAT[ *connection\_name; “sql”; data\_type* ];

*data\_type*表示调用此存储过程参数的数据类型，多个参数类型之间以逗号分隔，须与参数一一对应。

例如：

CStatement cstat = GET\_CSTAT[ conn; “call PR\_Sum(?,?,?)”; double,int,float ];

### SQL

SQL[ *“sql”; stat\_name; type* ]表示在指定SQL执行器（Statement）上以指定执行方式（update或query）执行SQL。关键字中*sql*表示SQL语句，*stat\_name*表示指定SQL执行器的名字，*type*表示执行方式。一个SQL关键字只能包含一句SQL语句。

*sql*用双引号包含，*stat\_name*是GET\_STAT返回的执行器。

*type*值为1，表示此*sql*为读，返回查询结果（ResultSet）；值为2表示写，返回影响行数；-1表示读并会抛异常，不抛则为错误，此时关键字返回预期异常原因的字符串；-2表示写并会抛异常，不抛则为错误，返回预期异常原因的字符串；3表示*sql*是CEDAR的explain语句，关键字返回结果是字符串。-3表示*sql*是CEDAR的explain语句，会抛异常是正确，不抛异常是错误，返回预期异常原因的字符串。

### PSQL

PSQL[ *pstat\_name; type; (variables* | *values)\** ]表示在指定预编译SQL执行器上以指定方式执行SQL。预编译SQL的参数值通过*variables*传入。*pstat\_name*表示指定预编译SQL执行器的名字，*type*表示执行方式；*variables*表示变量，*values*表示值，可用其替代*variables*，类型必须与GET\_PSTAT获得的相应PStatement的参数类型一致，可以输入变量或值。

*type*值为1，表示读返回查询结果；值为2表示写，返回影响行数；-1表示读并会抛异常，不抛则为错误，返回预期异常原因的字符串；-2表示写并会抛异常，不抛则为错误，返回预期异常原因的字符串；3表示*sql*是CEDAR的explain语句，关键字返回结果是字符串；-3表示sql是CEDAR的explain语句，会抛异常，不抛则为错误，返回预期异常原因的字符串。

*variables*的个数不唯一，之间以逗号分隔且不得空缺，变量顺序须与预编译SQL的指定顺序保持一致。

### CSQL

CSQL[ *cstat\_name; type; (variable|value param\_type)\** ]表示在存储过程SQL执行器上以指定方式执行SQL。存储过程的输入输出通过变量传递。*cstat\_name*表示指定存储过程SQL执行器的名字，*type*表示执行方式，*variable*表示变量，*param\_type*表示*variable*对应的输入输出类型，*variable*和*param\_type*构成一个变量参数，以空格分隔。*value*表示值，可用其替代*variable*，类型必须与GET\_CSTAT获得的相应CStatement的参数类型一致。

*type*值为1，表示读返回查询结果；值为2表示写返回影响行数。

*variable*和*param\_type*构成的变量参数的个数不唯一，变量参数间以逗号分隔，变量参数顺序须与存储过程的参数顺序保持一致。*param\_type*值有in, out, inout三种。

### TX

TX[ *connection\_name; tx\_id* ]表示事务，*connection\_name*是GET\_CONN返回的连接名，*tx\_id*是事务的三种操作，值为start表示事务开始，值为commit表示事务提交，值为abort表示事务回滚。

### VERIFY

VERIFY[ *variable1*; *variable2*; *relation\_operator* [; *set\_type*] ]表示两个对象的验证，关系运算符支持 ==，!=，>，>=，<，<=，当不是验证相等或不等时，两个对象必须只有一个元素。对象可为查询结果集、理想结果集和基本数据类型。*Set\_type*表示两个结果集按照集合方式比较，行之间不需一一对应。

例如：VERIFY[ rs1; rs2; ==; set\_type];两个结果集按照集合方式对比，VERIFY[ rs1; rs2; == ];两个集合严格按照行顺序对比。

### SLEEP

SLEEP[ *number; time\_unit* ]表示暂停指定的时间，*number*必须是正整数，*time\_unit*表示时间单位，支持microsecond、millisecond、second、minute、hour。

### ST

ST[ *number1; number2; clean/remain; synchronous/asynchronous* ]表示压力测试，包含了功能测试时所需的环境压力和性能测试所需功能。*number1*表示压力负载的编号，*number2*表示此负载的运行次数。当有赋值时则表示性能测试，执行完毕当前案例才能结束，如：

int a = ST[1; 1; clean; synchronous];

当没有赋值时则表示压力测试，案例执行完毕时会结束压力测试，如：

ST[1; 1; clean; synchronous];

*Clean/remain*表示压测产生的数据是否清空，*synchronous* / *asynchronous*表示当前语句同步或异步执行。

### IMPORT\_DBI

IMPORT\_DBI[ *“dbi\_name”* ]表示导入指定的测试场景，即建立表结构并插入数据。*dbi\_name*表示当前测试组别目录下的文件名（不带后缀）。

dbi文件以“[-]+”作为分隔符，分隔符之上为IMPORT\_DBI的执行内容，分隔符之下为CLEAR\_DBI的执行内容。均为SQL语句。

### CLEAR\_DBI

CLEAR\_DBI[ *“dbi\_name”* ]作用与IMPORT\_DBI相反，清空指定的测试场景，即删除表结构和数据。

### IMPORT\_IRS

IMPORT\_IRS[ *“result\_name”;* (*variable integer*)+ ]表示从文件中读取理想结果集到相应的结果集变量中。*result\_name*是当前测试组别目录下的文件名（不带后缀），须用双引号包含；*variable*表示理想结果集变量，integer表示变量对应文件中的位置（从0开始计数），变量不唯一，之间以逗号分隔。理想结果集书写参照第二章IdealResultSet部分。

例如：IMPORT\_IRS[ “function”; irs 3, irs1 2]；

### SYS

SYS[ *sys\_statement; ip\_address* ]表示系统功能，*sys\_statement*是相应系统功能命令，详见第三章，*ip\_address*只有命令是shell时才存在。

### DEF\_PROC

DEF\_PROC[ *str\_variable; stat\_name* ]表示定义存储过程，*str\_variable*是存放存储过程定义语句的变量名，且这个变量必须是String类型。*stat\_name*是SQL执行器（Statement）。

### CONTAIN

CONTAIN[*str\_variable|“string”; str\_variable|“string”; variable|number*]表示在源字符串中匹配到指定数目的字符串，不符合时返回ERROR，案例错误。第一个参数是源字符串，第二个参数是匹配字符串，第三个参数是数目。三个参数都可以用变量指定，也可以直接输入字符串或数字。

### if else

if( *condition* ){} else{}表示分支结构，*condition*表示条件表达式

### while

while( *number* ){}表示循环结构，*number*表示循环的次数。

### ##

一行以##开头表示此行为注释，以CR LF（换行）作为注释结束。

### ERROR

当执行到此关键字时表示此案例错误，直接结束当前案例的执行，用法是添加在某些分支流程中。

### CAL

计算关键字里的表达式。例如：CAL[ a = a\*3 ]。不支持其他语言里的缩写语法糖，如+=, \*=，++等。

### INDEX\_OF

INDEX\_OF[ *variable*; *row*[, *column*; *data\_type* ]]，获取List、ResultSet或IdealResultSet变量中指定位置的数据，列表只有行*row*，理想结果集和实际结果集行列和数据类型都必须有，*data\_type*是指定位置的数据类型。行和列都是从0开始计数。

### GRD（暂未实现）

GRD[ “*sql*”; *data\_types*; *rows*; *synchronous/asynchronous* ]，生成指定数量的随机数据。*sql*是预编译插入操作；*data\_types*是*sql*中一一对应参数的数据类型，其中可包含主键主要标志PK，varchar长度；*rows*是随机数据的行数。synchronous表示插入完成后再执行下一条命令，asynchronous表示边插入边执行下一条命令。

如：GRD[“insert into t1(c1,c2,c3) values(?,?,?)”; int PK, int, varchar 20; 200000; synchronous]；

注：用户需要自行先创建插入数据所需要的表。

# 数据类型

本章介绍Woodpecker支持的数据类型及其含义。

## 数据类型

表2.1 数据类型

|  |  |
| --- | --- |
| 数据类型 | 说明 |
| int | 整型 |
| long | 长整形 |
| boolean | 布尔型 |
| char | 字符 |
| float | 单精度浮点 |
| double | 双精度浮点 |
| String | 字符串 |
| Decimal | 小数 |
| List<var\_category> | 链表 |
| Connection | JDBC连接 |
| Statement | SQL执行器 |
| PStatement | 预编译SQL执行器 |
| CStatement | 存储过程SQL执行器 |
| ResultSet | 查询结果集 |
| IdealResultSet<var\_category> | 理想结果集 |
| Timestamp | 时间戳 |

## 数据类型详述

### 基本数据类型

int和long与Java语言中int类型和long类型含义一致，分别是4字节和8字节。boolean与Java语言中boolean类型含义一致。char与Java语言中char类型含义一致，2字节。float与Java语言中float类型含义一致，4个字节。double与Java语言中double类型含义一致，8个字节。String与Java语言中String类型含义一致。

Timestamp类型含义与java.sql.Timestamp一致。只能用字符串创建，且格式必须符合*yyyy-[m]m-[d]d [h]h:[m]m:[s]s[.f...]*。如：2017-06-1 10:5:47，日期和时间之间必须且只能有一个空格。

### Decimal

与Java语言中BigDecimal类型含义一致。

### List<*var\_category*>

与Java语言中List<E>含义一致，*var\_category*表示数据类型，包括int、long、boolean、char、float、double、String、Decimal。一次声明*var\_category*只能选一种类型。使用方式如：

List<int> list = [1,2,3];

### Connection

Connection表示JDBC连接。

### Statement

含义与JDBC中的Statement接口相同。

### PStatement

含义与JDBC中的PreparedStatement接口相同。

### CStatement

含义与JDBC中的CallableStatement接口相同。

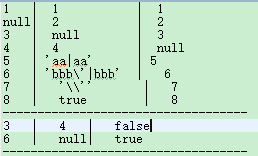
### ResultSet

表示查询结果集，含义与JDBC中的ResultSet接口相同。

### IdealResultSet<*var\_category*>

用于存储理想结果集，*var\_category*表示数据类型，包括int、boolean、char、float、double、String、Decimal。*var\_category*允许有多个，多个类型间以逗号分隔，一个类型对应理想结果集中一列的数据类型，类型的顺序与理想结果集中列的顺序一致。

每个理想结果集以分隔符：“--.\*”表示结束；不同记录之间的分隔符为：“\n”或者“\r\n”；不同元素之间的分隔符为：“|”。某个元素为空时用“null”表示。元素是字符串则用单引号包含，转义字符是“\”；第一行不能是分隔符，每个结果集以分隔符结尾。示例如下。



# 系统功能关键字

## SLEEP

语法格式：SLEEP[ *number; time\_unit* ]，暂停当前线程指定的时间，*number*是正整数，*time\_unit*是时间单位，支持microsecond、millisecond、second、minute、hour。无返回结果。

## SYS

语法格式：SYS[ *sys\_statement; ip\_address* ]，SYS关键字目前支持的系统功能命令如表3.1所示：

表3.1 SYS系统功能命令

|  |  |
| --- | --- |
| 命令sys\_statement | 说明 |
| is\_cluster\_available | 集群是否可服务 |
| await\_available n | 在n秒内等待所有集群可服务 |
| merge | 对主集群发起每日合并后返回 |
| minor\_freeze | 对UPS上的memtable冻结 |
| is\_merge\_done | 集群合并是否完成 |
| await\_merge\_done n | 在n秒内等待所有集群合并完成 |
| reelect | 对主集群换主 |
| await\_reelect\_done n | 在n秒内等待换主完成 |
| exist\_master | 集群是否有主 |
| kill\_server -option | 杀CEDAR server |
| start\_server variable | 启server |
| add\_server –option ip | 添加server |
| set\_master variable | 在所有集群中设定主集群 |
| shell\_command | shell命令 |
| gather\_statistics | 开启统计信息收集任务 |
| is\_gather\_done | 统计信息收集是否完成 |

### is\_cluster\_available

SYS[ is\_cluster\_available ]表示集群是否可服务，命令返回结果为整型数字，0代表所有集群可服务，1代表有一个备集群不可服务，其余集群可服务，2代表两个备集群不可服务，其余集群可服务，3代表主集群不可服务，备集群都可服务，4代表一个备集群可服务，其余集群不可服务，5代表所有集群不可服务。

如：int ava\_result = SYS[ is\_cluster\_available ];

### await\_available

SYS[ await\_available n ]表示在n秒内等待所有集群可服务，超出指定时间仍有集群不可服务则表示当前案例失败。n为非正整数表示直到可用才返回。

### merge

SYS[ merge ]表示对主集群发起每日合并，发起合并后返回不等待合并完成。

### minor\_freeze

SYS[ minor\_freeze ]表示对主集群UPS上的memtable进行冻结。

### is\_merge\_done

SYS[ is\_merge\_done ]表示集群合并是否完成，命令返回结果为整型数字，0代表所有集群合并完成，1代表有一个备集群未合并完成，其余集群合并完，2代表两个备集群未合并完成，其余集群合并完，3代表主集群未合并完成，其余备集群合并完，4代表一个备集群合并完成，其余集群未合并完成，5代表所有集群都未合并完成。

### await\_merge\_done

SYS[ await\_merge\_done n ]表示在n秒内等待所有集群合并完成，超出指定时间仍有集群未合并完成则表示当前案例失败。n为非正整数表示直到每日合并结束才返回。

### reelect

SYS[ reelect ]表示对主集群发送重新选举命令，进行换主选举。发起选举后返回不需等待选举完成。

### await\_reelect\_done

SYS[ await\_reelect\_done ] 表示在n秒内等待选举完成，超出指定时间仍无新的主集群产生则表示当前案例失败。n为非正整数表示直到选举完成才返回。

### exist\_master

SYS[ exist\_master ]表示当前集群是否有主，当存在主时返回true。

### kill\_server

SYS[ kill\_server -*option* ] 表示杀CEDAR的server，返回被杀server的信息。*option*的具体值见表3.2：

表3.2 kill\_server -option参数值

|  |  |
| --- | --- |
| 参数（-option） | 说明 |
| -master | 主集群所有server |
| -slave | 备集群所有server |
| -slave\_all | 所有备集群所有server |
| -all | 所有集群所有server |
| -ups\_master | 主集群UPS |
| -ups\_slave | 备集群一个UPS |
| -ups\_slave\_all | 备集群所有UPS |
| -ups\_all | 所有集群UPS |
| -rs\_master | 主集群RS |
| -rs\_slave | 备集群一个RS |
| -rs\_slave\_all | 备集群所有RS |
| -rs\_all | 所有集群RS |
| -lms\_master | 主集群LMS |
| -lms\_slave | 备集群一个LMS |
| -lms\_slave\_all | 备集群所有LMS |
| -lms\_all | 所有集群LMS |
| -ms\_master n | 主集群随机n个MS |
| -ms\_master\_all | 主集群所有MS |
| -ms\_slave n | 某个备集群中随机n个MS |
| -ms\_slave\_all n | 所有备集群中随机n个MS |
| -cs\_master n | 主集群随机n个CS |
| -cs\_master\_all | 主集群所有CS |
| -cs\_slave n | 某个备集群中随机n个CS |
| -cs\_slave\_all n | 所有备集群中随机n个CS |

例如：

1、杀死主集群所有server

String b= SYS[kill\_server -master];

2、杀死某个备集群中随机n个MS

String c= SYS[kill\_server -ms\_slave n];

3、所有备集群中随机n个CS

String d= SYS[kill\_server -cs\_slave\_all n];

### start\_server

SYS[ start\_server variable ]表示启某个被杀server，variable是存放kill\_server返回字符串信息的String变量名，存储有被杀server的信息。

例如：SYS[ start\_server b];表示重启刚被杀死的主集群

### add\_server

SYS[ add\_server –*option* ip\_address ]表示在指定ip上添加一个server，server类型由*option*决定，见表3.3：

表3.3 add\_server option参数值

|  |  |
| --- | --- |
| 参数 | 说明 |
| -ms\_master | 在主集群上添加MS |
| -cs\_master | 在主集群上添加CS |
| -ms\_slave | 在备集群上添加MS |
| -cs\_slave | 在备集群上添加CS |

### set\_master

SYS[ set\_master variable ]表示设定variable所代表的server所在集群为主集群，前提是当前所有集群中不存在主集群，variable是kill\_server返回的字符串信息，存储有被杀server的信息。

### shell\_command

SYS[ *“shell\_command”*; *ip\_address* ]表示在指定IP地址上执行shell命令，默认路径是用户主目录，命令必须用双引号包含。

### gather\_statistics和is\_gather\_done

SYS[ gather\_statistics ]和SYS[ is\_gather\_done ]表示开启统计信息收集任务和查看统计信息收集状态。