Sqlcmd

Sqlcmd命令执行描述文件中的定义的SQL语句，并按照描述文件要求，SQL结果可以保存到CSV，JSON，EXCEL，SQL类型的文件中。

描述文件是一个JSON格式的文件，定义如下：

{

"sql\_string":"select …",

"sql\_file":"d:/sql/cmd.sql",

"db":{

…

},

"writer":{

…

}

}

说明：

“sql\_string”，定义一个可带变量的SQL语句，请看后文关于可带变量的SQL语句描述

“sql\_file”,定义一个文件，文件内容为可带变量的SQL语句

一个描述文件中至少要定义一个“sql\_string”或”sql\_file”，如果两个都定义了，则只会选用“sql\_file”。

“db”必选项，定义SQL语句将执行的DBAPI接口，即SQL在那个数据库上执行。“db”定义数据库的DRIVER和建立数据库连接的参数，sqlcmd理论上支持python所支持的所有数据库，但前支持：mssql，sqlite3，db2，postgresql，有需要可以增其他接口。

如连接MS SQL Server：

”db“:{

"driver":"pymssql",

"kwargs":{

"host":"192.168.161.1",

"user":"someone",

"password":”password",

"database":"mydb"

}

}

连接postgresql的：

“db”: {

"driver":"psycopg2",

"kwargs":{

"host":"192.168.161.128",

"user":"someone",

"password":"password",

"database":"mydb"

}

}

Writer 把SQL的结果输出，目前支持输出到csv，excel，json，sql四种文件中。

每种输出格式的writer定义略有不同。

输出到csv的格式为：

”writer”:{

“filetype”:”csv”,

“kwargs”:{

“outfile”:”d:/data/data1.csv”,

“write\_mode”:”w”

“delimiter”:”|+|”

}

}

如果“outfile”没有定义，则生成数据输出到标准输出上；

Write\_mode有两个选择项：’w’,’a’

‘w’如果文件存在将覆盖

‘a’如果文件存在则追加

Delimiter定义分隔符，缺省分隔符是逗号“,”

输出到json格式的文件为：

”writer”:{

“filetype”:”json”,

“kwargs”:{

“outfile”:”d:/data/data1.json”

}

}

输出到excel文件格式为：

“writer”:{

“filetype”:”excel”,

“kwargs”:{

“outfile”:”d:/data/data1.xlsx”,

“patternfile”:”d:/pattern/mydata.xlsx”,

“sheet”:”Sheet1”,

“start”:”A1”

}

}

输出到sql文件

”writer”:{

“filetype”:”sql”,

“kwargs”:{

“outfile”:”d:/data/data1.sql”,

“database”:”mydb”,

“table”:”dbo.table1”,

“coding”:”utf8”

}

}

可带参数的SQL语句

支持多语句，在一个session中执行多条语句，语句间用分号“;”隔离，最后一个SQL后面不能有分号。

可带参数，参数形式为${xxxx}，${\_\_mainsql\_\_}参数有特殊含义，可放在语句的任何地方，说明此句SQL语句需要输出结果，如果所有的SQL语句中都没有定义${\_\_main\_\_}变量，那么即使定义的writer，也不会有真正的输出。${\_\_mainsql\_\_}变量在整个可带参数的SQL语句中只能定义一次。

变量仅可用来替代数据，不能用来替代sql关键字、表，视图，字段，以及其他的数据库对象

变量获取，可通过环境变量，命令行指定方法给SQL指定参数，环境变量中的参数优先级低于命令行参数，即命令行可以改写环境变量中定义的变量值。

例子：

Create tabel1 as

select \* from table2 where f1 >0;

select \* from table1${\_\_mainsql\_\_}