# DolphinDB快速入门

DolphinDB开发文档http://www.dolphindb.com/cn/help/

# 启动单机版DolphinDB

进入DolphinDB\_Linux\_V0.6/server目录

在终端执行./dolphindb

随后就可以通过web或GUI访问localhost:8848来进行一些基本操作

停止运行只需要在终端输入quit即可

# 加载数据集

使用loadText可以快速将文本文件读入内存中，如

t = loadText(“/path/to/file”, “|”) //第二个参数”|”用于指定分隔符，默认为”,”

schema(t).colDefs //查看表的模式

----参考返回结果----

|  |  |  |
| --- | --- | --- |
| name | typeString | typeInt |
| Time | NANOTIME | 13 |
| Exchange | CHAR | 2 |
| … | … | … |

正常载入后不会有返回结果，可以通过一些简单的SQL操作来查看是否载入成功

关于更多的数据加载方法可以参阅DolphinDB开发文档第9章：文件I/O

注意：loadText直接读入内存，因此不能超过内存限制，可以使用loadTextEx来读入分布式表，参考DolphinDB开发文档第10章：数据库和分布式计算

# 执行一些简单的SQL操作

DolphinDB支持SQL操作，并能直接在脚本语言中使用，如

select top 100 \* from t //返回前100条记录，t就是2中加载的数据集

select count(\*) from t where Time>=11:00:00

关于更多的数据类型可以参阅DolphinDB开发文档第2章：数据类型和结构

关于支持的SQL及相关语法可以参阅DolphinDB开发文档第8章：SQL语句

注意：DolphinDB作为一体化大数据分析平台，更长于数据的批量添加与查询

# 脚本语言的用法

DolphinDB支持的脚本语言语法与python类似，支持自定义函数等更多特性，与SQL结合可以方便编写一些复杂函数

def sumPrice(tb, tm) { //使用def来定义函数，另外也支持lambda

tmp = exec Trade\_Price from tb where Time>=tm //脚本语言与SQL结合使用

s = 0

for(x in tmp) { //支持常见的编程方式，如for/if-else等

s = s + x

}

return s

}

sumPrice(t, second(11:00:00)) //调用函数，整个函数等价于下面这一行代码的功能

sum(exec Trade\_Price from t where Time>=11:00:00) //使用sum等内置函数

SQL与脚本语言的配合使用可以极大简化编程与数据操作的过程

关于更多的脚本语言的语法可以参阅DolphinDB开发文档第5章：编程语句

关于更多的内置函数和命令可以参阅DolphinDB开发文档第13章：函数和命令