dolphinDB安装与使用【2.00版】

1. 【股票数据导入完整版】
   1.1. 建库建分区表
   1.2. 股票数据准备：
   1.3. 程序执行：
   1.4. 查询
      1.4.1. 命令行查询
      1.4.2. python查询
2. dolphinDB安装与使用下载安装
   2.1. 下载安装
   2.2. 开启数据库
   2.3. 关闭数据库
   2.4. 连接数据库
      2.4.1. web连接
   2.5. 内存表
   2.6. 建库建分区表
   2.7. table用法
      2.7.1. 写法三：

【可实际使用部分 开始 2023-11-13】

1. 【股票数据导入完整版】

1.1. 建库建分区表

```
heaven@mozart:~/DolphinDB.200/server$ rlwrap -r ./dolphindb -remoteHost 192.168.1.25 -remotePort 8848
DolphinDB Terminal 2.00.10.5 (Build:2023.11.06). Copyright (c) 2011~2023 DolphinDB, Inc.

login(userId=`admin, password=`123456);
>
 dbTime = database("", RANGE,1990.01.01  1991.01.01  1992.01.01  1993.01.01  1994.01.01  1995.01.01  1996.01.01  1997.01.01  1998.01.01  1999.01.01  2000.01.01  2001.01.01  2002.01.01  2003.01.01  2004.01.01  2005.01.01  2006.01.01  2007.01.01  2008.01.01  2009.01.01  2010.01.01  2011.01.01  2012.01.01  2013.01.01  2014.01.01  2015.01.01  2016.01.01  2017.01.01  2018.01.01  2019.01.01  2020.01.01  2021.01.01  2022.01.01  2023.01.01  2024.01.01  2025.01.01  2026.01.01  2027.01.01  2028.01.01  2029.01.01  2030.01.01  2031.01.01  2032.01.01  2033.01.01  2034.01.01  2035.01.01  2036.01.01  2037.01.01  2038.01.01  2039.01.01  2040.01.01  2041.01.01  2042.01.01  2043.01.01  2044.01.01  2045.01.01  2046.01.01  2047.01.01  2048.01.01  2049.01.01  2050.01.01)
dbStockID = database("", HASH, [SYMBOL, 100])
db = database(directory="dfs://stock1f",partitionType=COMPO,partitionScheme=[dbTime,dbStockID],engine="TSDB")
schema = table(1:0, `stockid`market`date`time`open`high`low`close`turnover_volume`turnover_value, [SYMBOL,SYMBOL,DATE,MINUTE,DOUBLE,DOUBLE,DOUBLE,DOUBLE,INT,DOUBLE])
pt = db.createPartitionedTable(table=schema, tableName=`stock1f, partitionColumns=`date`stockid, sortColumns=`stockid`date`time,keepDuplicates=FIRST);
>
schema(pt);
engineType->TSDB
keepDuplicates->FIRST
chunkGranularity->TABLE
sortColumns->["stockid","date","time"]
colDefs->name           typeString typeInt extra comment
--------------- ---------- ------- ----- -------
stockid         SYMBOL     17
market          SYMBOL     17
date            DATE       6
time            MINUTE     9
open            DOUBLE     16
high            DOUBLE     16
low             DOUBLE     16
close           DOUBLE     16
turnover_volume INT        4
turnover_value  DOUBLE     16

```

```
chunkPath->
partitionColumnIndex->[2,0]
partitionColumnName->["date","stockid"]
partitionColumnType->[6,17]
partitionType->[2,5]
partitionTypeName->["RANGE","HASH"]
partitionSchema->([1990.01.01,1991.01.01,1992.01.01,1993.01.01,1994.01.01,
1995.01.01,1996.01.01,1997.01.01,1998.01.01,1999.01.01,2000.01.01,2001.01.
01,2002.01.01,2003.01.01,2004.01.01,2005.01.01,2006.01.01,2007.01.01,2008.
01.01,2009.01.01,2010.01.01,2011.01.01,2012.01.01,2013.01.01,2014.01.01,20
15.01.01,2016.01.01,2017.01.01,2018.01.01,2019.01.01...],100)
partitionSites->

>
```

1.2. 股票数据准备：

目录：

```
[rabbit @ mozart ~/udata] $ pwd
/home/rabbit/udata
[rabbit @ mozart ~/udata] $
drwxrwxrwx 2 rabbit rabbit 200704 Jun 26 10:32 x_201708
drwxrwxrwx 2 rabbit rabbit 229376 Jun 26 10:38 x_201709
drwxrwxrwx 2 rabbit rabbit 212992 Jun 26 10:48 x_201710
drwxrwxrwx 2 rabbit rabbit 208896 Jun 26 15:29 x_201711
drwxrwxrwx 2 rabbit rabbit 229376 Jun 26 15:42 x_201712
drwxrwxrwx 2 rabbit rabbit 262144 Jun 26 15:24 x_202101
drwxrwxrwx 2 rabbit rabbit 274432 Jun 26 19:14 x_202102
drwxrwxrwx 2 rabbit rabbit 270336 Jun 26 19:01 x_202103
drwxrwxrwx 2 rabbit rabbit 278528 Jun 26 18:48 x_202104
drwxrwxrwx 2 rabbit rabbit 282624 Jun 26 15:48 x_202105
drwxrwxrwx 2 rabbit rabbit 262144 Jun 26 16:31 x_202106
drwxrwxrwx 2 rabbit rabbit 290816 Jun 26 16:11 x_202107
drwxrwxrwx 2 rabbit rabbit 282624 Jun 26 19:09 x_202108
drwxrwxrwx 2 rabbit rabbit 294912 Jun 26 16:45 x_202109
drwxrwxrwx 2 rabbit rabbit 307200 Jun 26 16:17 x_202110
drwxrwxrwx 2 rabbit rabbit 290816 Jun 26 16:24 x_202111
drwxrwxrwx 2 rabbit rabbit 282624 Jun 26 16:38 x_202112
drwxrwxrwx 2 rabbit rabbit 237568 Jun 26 19:55 x_202201
drwxrwxrwx 2 rabbit rabbit 241664 Jun 26 18:54 x_202202
drwxrwxrwx 2 rabbit rabbit 233472 Jun 26 15:56 x_202203
drwxrwxrwx 2 rabbit rabbit 225280 Jun 26 19:49 x_202204
drwxrwxrwx 2 rabbit rabbit 245760 Jun 26 19:42 x_202205
drwxrwxrwx 2 rabbit rabbit 225280 Jun 26 17:01 x_202206
drwxrwxrwx 2 rabbit rabbit 294912 Jun 26 15:37 x_202207
drwxrwxrwx 2 rabbit rabbit 311296 Jun 26 16:54 x_202208
drwxrwxrwx 2 rabbit rabbit 237568 Jun 26 15:18 x_202209
drwxrwxrwx 2 rabbit rabbit 245760 Jun 26 19:20 x_202210
drwxrwxrwx 2 rabbit rabbit 229376 Jun 26 16:04 x_202211
drwxrwxrwx 2 rabbit rabbit 253952 Jun 26 17:08 x_202212
drwxrwxrwx 2 rabbit rabbit 319488 Jun 26 17:15 x_202301
drwxrwxrwx 2 rabbit rabbit 299008 Jun 26 20:03 x_202302
drwxrwxrwx 2 rabbit rabbit 307200 Jun 26 18:42 x_202303
drwxrwxrwx 2 rabbit rabbit 311296 Jun 26 19:27 x_202304
drwxrwxrwx 2 rabbit rabbit 307200 Jun 26 19:35 x_202305

```

文件：

```
[rabbit @ mozart ~/udata/x_201708] $ ll | more
total 1279968
-rwxrwxrwx 1 rabbit rabbit 437546 Jun 26 10:31 x_000001.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 433827 Jun 26 10:30 x_000002.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 413599 Jun 26 10:31 x_000004.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 396244 Jun 26 10:32 x_000005.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 403195 Jun 26 10:29 x_000006.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 415056 Jun 26 10:29 x_000007.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 401142 Jun 26 10:31 x_000008.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 409786 Jun 26 10:29 x_000009.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 396348 Jun 26 10:30 x_000010.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 425444 Jun 26 10:31 x_000011.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 404981 Jun 26 10:29 x_000012.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 412489 Jun 26 10:29 x_000014.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 403311 Jun 26 10:31 x_000016.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 391619 Jun 26 10:30 x_000017.SZ_20170801_20170831.csv
-rwxrwxrwx 1 rabbit rabbit 410164 Jun 26 10:30 x_000019.SZ_20170801_20170831.csv
```

文件样式：

```
[rabbit @ mozart ~/udata/x_201708] $ head x_603999.SH_20170801_20170831.csv
stockid,market,date,time,open,high,low,close,turnover_volume,turnover_value
603999,SH,2017-08-01,09:31,9.6200,9.6300,9.6100,9.6300,12200,117442.0000
603999,SH,2017-08-01,09:32,9.6300,9.7000,9.6300,9.7000,7000,67885.0000
603999,SH,2017-08-01,09:33,9.7000,9.7000,9.6800,9.6800,600,5816.0000
603999,SH,2017-08-01,09:34,9.6800,9.6800,9.6400,9.6800,5200,50229.0000
603999,SH,2017-08-01,09:35,9.6800,9.6800,9.6700,9.6800,104500,1010560.0000
603999,SH,2017-08-01,09:36,9.6800,9.7000,9.6800,9.7000,14700,142525.0000
603999,SH,2017-08-01,09:37,9.7300,9.7300,9.7000,9.7300,35700,347122.0000
603999,SH,2017-08-01,09:38,9.7300,9.7800,9.7300,9.7800,70500,687467.0000
603999,SH,2017-08-01,09:39,9.7800,9.8900,9.7800,9.8900,163200,1608674.0000
[rabbit @ mozart ~/udata/x_201708] $
```

导入的程序内容：

```
heaven@mozart:~/rundos$ cat import.dos
db = database("dfs://stock1f")

//请注意更换目录dataDir
dataDir="/home/rabbit/udata/x_201708"

xxy=extractTextSchema("/home/rabbit/udata/x_201708/x_603999.SH_20170801_20170831.csv")
update xxy set type=`SYMBOL where name=`stockid;

def importTxtFiles(dataDir, db,sch){
    dataFiles = exec filename from files(dataDir) where isDir=false
    for(f in dataFiles){
        loadTextEx(db, `stock1f, `date`stockid, dataDir + "/" + f, schema=sch, sortColumns=`stockid`date`time)
    }
}
importTxtFiles(dataDir, db,xxy);
heaven@mozart:~/rundos$
```

1.3. 程序执行：

```
heaven@mozart:~/DolphinDB.200/server$ rlwrap -r ./dolphindb -remoteHost 192.168.1.25 -remotePort 8848 -uid admin -pwd 123456  -run ~/rundos/import.dos
```

1.4. 查询

1.4.1. 命令行查询

```
login(userId=`admin, password=`123456);
db = database("dfs://stock1f");
pt=loadTable(db,"stock1f");
select count(*) from pt;
count
--------
17529555

>
select * from pt where stockid="000007";
stockid market date       time   open              high              low               close             ...
------- ------ ---------- ------ ----------------- ----------------- ----------------- ----------------- ---
000007  SZ     2017.08.01 09:31m 16.879999999999999 17                16.769999999999999 16.949999999999999 ...
000007  SZ     2017.08.01 09:32m 16.949999999999999 17.039999999999999 16.949999999999999 17.010000000000001 ...
000007  SZ     2017.08.01 09:33m 17.059999999999998 17.07              16.989999999999998 16.989999999999998 ...
000007  SZ     2017.08.01 09:34m 16.949999999999999 16.989999999999998 16.899999999999998 16.899999999999998 ...
000007  SZ     2017.08.01 09:35m 16.949999999999999 16.949999999999999 16.780000000000001 16.8               ...
000007  SZ     2017.08.01 09:36m 16.8               16.969999999999998 16.8              16.96              ...
000007  SZ     2017.08.01 09:37m 16.96              17.030000000000001 16.96             17                 ...
000007  SZ     2017.08.01 09:38m 17                 17.010000000000001 16.839999999999999 16.949999999999999 ...
000007  SZ     2017.08.01 09:39m 16.949999999999999 17.120000000000001 16.949999999999999 17.109999999999999 ...
000007  SZ     2017.08.01 09:40m 17.100000000000001 17.149999999999998 17.100000000000001 17.149999999999998 ...
000007  SZ     2017.08.01 09:41m 17.199999999999999 17.199999999999999 17.109999999999999 17.179999999999999 ...
000007  SZ     2017.08.01 09:42m 17.170000000000001 17.25              17.170000000000001 17.25              ...
000007  SZ     2017.08.01 09:43m 17.25              17.309999999999998 17.25             17.3               ...
000007  SZ     2017.08.01 09:44m 17.3               17.309999999999998 17.25             17.25              ...
000007  SZ     2017.08.01 09:45m 17.25              17.25              17.149999999999998 17.149999999999998 ...
000007  SZ     2017.08.01 09:46m 17.149999999999998 17.199999999999999 17.129999999999998 17.149999999999998 ...
```

```
29   000007  SZ     2017.08.01 09:47m 17.149999999999998 17.199999999999999 17.
     14              17.149999999999998 ...
30   000007  SZ     2017.08.01 09:48m 17.149999999999998 17.199999999999999 17.
     149999999999998 17.190000000000001 ...
31   000007  SZ     2017.08.01 09:49m 17.190000000000001 17.21              17.
     190000000000001 17.199999999999999 ...
32   000007  SZ     2017.08.01 09:50m 17.199999999999999 17.199999999999999 17.
     129999999999998 17.129999999999998 ...
33   000007  SZ     2017.08.01 09:51m 17.149999999999998 17.149999999999998 17.
     100000000000001 17.100000000000001 ...
34   000007  SZ     2017.08.01 09:52m 17.079999999999998 17.079999999999998 17.
     059999999999998 17.07              ...
35   000007  SZ     2017.08.01 09:53m 17.07              17.109999999999999 17.
     059999999999998 17.089999999999999 ...
36   000007  SZ     2017.08.01 09:54m 17.089999999999999 17.109999999999999 17.
     05              17.05              ...
37   000007  SZ     2017.08.01 09:55m 17.05              17.05              17.
     010000000000001 17.05              ...
38   000007  SZ     2017.08.01 09:56m 17.05              17.129999999999998 17.
     05              17.100000000000001 ...
39   000007  SZ     2017.08.01 09:57m 17.100000000000001 17.100000000000001 17.
     07              17.07              ...
40   000007  SZ     2017.08.01 09:58m 17.07              17.089999999999999 17.
     059999999999998 17.059999999999998 ...
41   000007  SZ     2017.08.01 09:59m 17.059999999999998 17.059999999999998 17.
     019999999999999 17.019999999999999 ...
42   000007  SZ     2017.08.01 10:00m 17.019999999999999 17.100000000000001 17.
     019999999999999 17.100000000000001 ...
43   ...
44
45   >
```

1.4.2. python查询

```
heaven@mozart:~/rundos/dolphindb_py$
heaven@mozart:~/rundos/dolphindb_py$ python d2.py
5542
             name typeString  typeInt  extra comment
0         stockid     SYMBOL       17    NaN
1          market     SYMBOL       17    NaN
2            date       DATE        6    NaN
3            time     MINUTE        9    NaN
4            open     DOUBLE       16    NaN
5            high     DOUBLE       16    NaN
6             low     DOUBLE       16    NaN
7           close     DOUBLE       16    NaN
8  turnover_volume        INT        4    NaN
9   turnover_value     DOUBLE       16    NaN
     stockid market       date                time   open   high    low  c
lose  turnover_volume  turnover_value
0     000007     SZ 2017-08-01 1970-01-01 09:31:00  16.88  17.00  16.77  1
6.95           69650       1174936.5
1     000007     SZ 2017-08-01 1970-01-01 09:32:00  16.95  17.04  16.95  1
7.01          136700       2324139.0
2     000007     SZ 2017-08-01 1970-01-01 09:33:00  17.06  17.07  16.99  1
6.99           93300       1590953.0
3     000007     SZ 2017-08-01 1970-01-01 09:34:00  16.95  16.99  16.90  1
6.90           41900        710476.0
4     000007     SZ 2017-08-01 1970-01-01 09:35:00  16.95  16.95  16.78  1
6.80          111000       1875972.0
...      ...    ...        ...                 ...    ...    ...    ...
 ...              ...             ...
5537  000007     SZ 2017-08-31 1970-01-01 14:56:00  15.73  15.73  15.70  1
5.70           36000        565499.0
5538  000007     SZ 2017-08-31 1970-01-01 14:57:00  15.70  15.73  15.70  1
5.70           23900        375486.0
5539  000007     SZ 2017-08-31 1970-01-01 14:58:00  15.70  15.70  15.70  1
5.70               0             0.0
5540  000007     SZ 2017-08-31 1970-01-01 14:59:00  15.70  15.70  15.70  1
5.70               0             0.0
5541  000007     SZ 2017-08-31 1970-01-01 15:00:00  15.70  15.72  15.70  1
5.72           32750        514830.0

[5542 rows x 10 columns]
<dolphindb.session.BlockReader object at 0x7f02cebf3e50>
     stockid market       date                time   open   high    low  c
lose  turnover_volume  turnover_value
0     000007     SZ 2017-08-01 1970-01-01 09:31:00  16.88  17.00  16.77  1
6.95           69650       1174936.5

```

```
1     000007     SZ 2017-08-01 1970-01-01 09:32:00  16.95  17.04  16.95  17.01          136700       2324139.0
2     000007     SZ 2017-08-01 1970-01-01 09:33:00  17.06  17.07  16.99  16.99           93300       1590953.0
3     000007     SZ 2017-08-01 1970-01-01 09:34:00  16.95  16.99  16.90  16.90           41900        710476.0
4     000007     SZ 2017-08-01 1970-01-01 09:35:00  16.95  16.95  16.78  16.80          111000       1875972.0
...      ...    ...        ...                 ...    ...    ...    ...     ...             ...             ...
5537  000007     SZ 2017-08-31 1970-01-01 14:56:00  15.73  15.73  15.70  15.70           36000        565499.0
5538  000007     SZ 2017-08-31 1970-01-01 14:57:00  15.70  15.73  15.70  15.70           23900        375486.0
5539  000007     SZ 2017-08-31 1970-01-01 14:58:00  15.70  15.70  15.70  15.70               0             0.0
5540  000007     SZ 2017-08-31 1970-01-01 14:59:00  15.70  15.70  15.70  15.70               0             0.0
5541  000007     SZ 2017-08-31 1970-01-01 15:00:00  15.70  15.72  15.70  15.72           32750        514830.0

[5542 rows x 10 columns]
total= 5542
heaven@mozart:~/rundos/dolphindb_py$
heaven@mozart:~/rundos/dolphindb_py$ cat d2.py
import os
import dolphindb.settings as keys
import dolphindb as db
s = db.session()
s.connect("localhost", 8848)
s.login("admin","123456")
s.database(dbPath="dfs://stock1f")
#trade = s.loadTable(tableName="stock1f",dbPath="dfs://stock1f")
#trade = s.loadTable(tableName="stock1f",dbPath="dfs://stock1f",sql="select count(*) from stock1f")
trade = s.loadTableBySQL(tableName="stock1f", dbPath="dfs://stock1f", sql="select * from stock1f where stockid='000007' and  date>2010.01.01")
print(trade.rows)
print(trade.schema)
print(trade.toDF())

script1='''
db=database("dfs://stock1f");
pt=loadTable(db,"stock1f");
select * from pt where stockid="000007";
'''
block= s.run(script1, fetchSize = 8192)
```

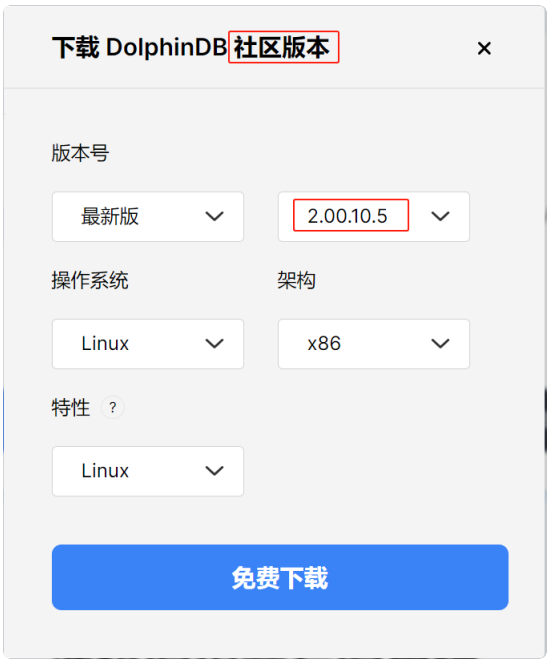
```
69   total = 0
70   while block.hasNext():
71       print(block)
72       tem = block.read()
73       print(tem)
74       total+=len(tem)
75
```

【可实际使用部分 结束】

2. dolphinDB安装与使用下载安装

2.1. 下载安装

首页 – DolphinDB 【下载地址】



```
heaven@mozart:~/DolphinDB.200/server$ ll
total 134852
drwxrwxr-x  7 heaven heaven     4096 Nov 13 14:38 ./
drwxrwxr-x  3 heaven heaven     4096 Nov 13 14:38 ../
drwxrwxr-x  5 heaven heaven     4096 Nov 13 14:38 clusterDemo/
-rwxrwxr-x  1 heaven heaven 33728128 Nov 13 14:38 dolphindb*
-rw-rw-r--  1 heaven heaven      280 Nov 13 14:38 dolphindb.cfg
-rw-rw-r--  1 heaven heaven     3212 Nov 13 14:38 dolphindb.dos
-rw-rw-r--  1 heaven heaven      387 Nov 13 14:38 dolphindb.lic
-rw-rw-r--  1 heaven heaven 61995220 Nov 13 14:38 libDolphinDB.so
-rw-rw-r--  1 heaven heaven  5620077 Nov 13 14:38 libgfortran.so.3
-rw-rw-r--  1 heaven heaven 28902143 Nov 13 14:38 libopenblas.so.0
-rw-rw-r--  1 heaven heaven   925835 Nov 13 14:38 libquadmath.so.0
-rw-rw-r--  1 heaven heaven  6635846 Nov 13 14:38 libstdc++.so.6
-rw-rw-r--  1 heaven heaven   214993 Nov 13 14:38 libtcmalloc_minimal.so.4
drwxrwxr-x  2 heaven heaven     4096 Nov 13 14:38 marketHoliday/
drwxrwxr-x  3 heaven heaven     4096 Nov 13 14:38 modules/
drwxrwxr-x  8 heaven heaven     4096 Nov 13 14:38 plugins/
-rwxrwxr-x  1 heaven heaven       74 Nov 13 14:38 startSingle.sh*
drwxrwxr-x 14 heaven heaven     4096 Nov 13 14:38 web/
heaven@mozart:~/DolphinDB.200/server$
```

```
chmod +x dolphindb
chmod +x startSingle.sh
```

2.2. 开启数据库

```
$ ./startSingle.sh
```

检查：

```
heaven@mozart:~/DolphinDB.200/server$ ps -ef | grep dolphindb
heaven     59896       1  2 14:44 pts/0    00:00:00 ./dolphindb -console 0 -mo
de single
heaven     60289   58332  0 14:44 pts/0    00:00:00 grep --color=auto dolphind
b
heaven@mozart:~/DolphinDB.200/server$
```

2.3. 关闭数据库

```
pkill -15 dolphindb
```

安全关机 — DolphinDB 2.0 documentation

2.4. 连接数据库

IP地址需要使用实际的IP，不能是127.0.0.1

```
~/DolphinDB.200/server$ rlwrap -r ./dolphindb -remoteHost 192.168.1.25 -remote
Port 8848
DolphinDB Terminal 2.00.10.5 (Build:2023.11.06). Copyright (c) 2011~2023 Dolph
inDB, Inc.

>login(userId=`admin, password=`123456);
>quit
~/DolphinDB.200/server$
```

standalone_server.md · 浙江智臾科技有限公司/Tutorials_CN – Gitee.com

管理员身份（默认账号：admin，默认密码：123456）

```
heaven@mozart:~/DolphinDB.200/server$  cat dolphindb.cfg
localSite=localhost:8848:local8848
mode=single
maxMemSize=32
maxConnections=512
workerNum=4
localExecutors=3
maxBatchJobWorker=4
dataSync=1
OLAPCacheEngineSize=2
TSDBCacheEngineSize=1
newValuePartitionPolicy=add
maxPubConnections=64
subExecutors=4
perfMonitoring=true
lanCluster=0heaven@mozart:~/DolphinDB.200/server$
```

2.4.1. web连接

`http://192.168.1.25:8848/` 管理员身份（默认账号：admin，默认密码：123456）

必须使用 实际IP，不能使用127.0.0.1

2.5. 内存表

```
>
n=1000000
ID=rand(10, n)
x=rand(1.0, n)
t=table(ID, x)
select * from t;
select count(*) from t;
```

```
heaven@mozart:~/DolphinDB.200/server$ rlwrap -r ./dolphindb -remoteHost 192.16
8.1.25 -remotePort 8848
DolphinDB Terminal 2.00.10.5 (Build:2023.11.06). Copyright (c) 2011~2023 Dolph
inDB, Inc.

login(userId=`admin, password=`123456);
>
n=1000000
ID=rand(10, n)
x=rand(1.0, n)
t=table(ID, x)
select * from t;
ID x
-- -----------------
7  0.554596869042143
3  0.162759877974167
5  0.041018647374585
3  0.438487786334008
1  0.061486455844715
9  0.859343684511259

select count(*) from t;
count
-------
1000000

>
```

2.6. 建库建分区表

创建数据库和表 — DolphinDB 2.0 documentation

一个库，多个分区表

- 连接数据库：`rlwrap -r ./dolphindb -remoteHost 192.168.1.25 -remotePort 8848`
- admin用户登录：`login(userId=`admin, password=`123456)`
- 创建**内存表**：`t=table(ID, x)`
- 创建**数据库**：`db=database("dfs://rangedb", RANGE,  0 5 10)`

  会在`/local8848/storage/CHUNKS`目录下创建一个`rangedb`的目录

  分为`[0,5) [5,10)`两个`range`
- 创建**数据库**：`db=database("dfs://rangedb", RANGE,  0..10)`
- 创建**分区表**：`pt=db.createPartitionedTable(t, `pttt, `ID)`以表`t`按`RANGE`方式建分区表，以`ID`进行`range`区分，分区表名在磁盘上为`pttt`.

```
>
n=10
ID=rand(10, n)
x=rand(1.0, n)
t=table(ID, x)
db=database("dfs://rangedb", RANGE,  0..10)
pt=db.createPartitionedTable(t, `ptbbbb, `ID)
pt.append!(t)
select * from pt;
```

```
heaven@mozart:~/DolphinDB.200/server/local8848/storage/CHUNKS/rangedb$ ll
total 68
drwxrwxr-x 11 heaven heaven 4096 Nov 13 17:13 ./
drwxrwxr-x  3 heaven heaven 4096 Nov 13 16:58 ../
drwxrwxr-x  7 heaven heaven 4096 Nov 13 17:13 0_1/
drwxrwxr-x  4 heaven heaven 4096 Nov 13 17:05 1_2/
drwxrwxr-x  4 heaven heaven 4096 Nov 13 17:05 2_3/
drwxrwxr-x  7 heaven heaven 4096 Nov 13 17:13 3_4/
drwxrwxr-x  7 heaven heaven 4096 Nov 13 17:13 4_5/
drwxrwxr-x  5 heaven heaven 4096 Nov 13 17:13 6_7/
drwxrwxr-x  7 heaven heaven 4096 Nov 13 17:13 7_8/
drwxrwxr-x  7 heaven heaven 4096 Nov 13 17:13 8_9/
drwxrwxr-x  5 heaven heaven 4096 Nov 13 17:13 9_10/
-rw-rw-r--  1 heaven heaven    0 Nov 13 17:13 dolphindb.lock
-rw-rw-r--  1 heaven heaven  105 Nov 13 16:58 domain
-rw-rw-r--  1 heaven heaven   48 Nov 13 17:12 ptaaaa.tbl
-rw-rw-r--  1 heaven heaven   48 Nov 13 17:13 ptbbbb.tbl
-rw-rw-r--  1 heaven heaven   48 Nov 13 17:13 ptcccc.tbl
-rw-rw-r--  1 heaven heaven   48 Nov 13 16:58 pt.tbl
-rw-rw-r--  1 heaven heaven   48 Nov 13 17:05 ptxxxx.tbl
heaven@mozart:~/DolphinDB.200/server/local8848/storage/CHUNKS/rangedb$
```

```
heaven@mozart:~/DolphinDB.200/server/local8848/storage/CHUNKS/rangedb$ tree
.
├── 0_1
│   ├── 2
│   │   └── pt_2
│   │       ├── ID.col
│   │       └── x.col
│   ├── 5
│   │   └── ptxxxx_3
│   │       ├── ID.col
│   │       └── x.col
│   ├── 7
│   │   └── ptaaaa_4
│   │       ├── ID.col
│   │       └── x.col
│   ├── 9
│   │   └── ptbbbb_5
│   │       ├── ID.col
│   │       └── x.col
│   └── b
│       └── ptcccc_6
│           ├── ID.col
│           └── x.col
├── 1_2
│   ├── 2
│   │   └── pt_2
│   │       ├── ID.col
│   │       └── x.col
│   └── 5
│       └── ptxxxx_3
│           ├── ID.col
│           └── x.col
├── 2_3
│   ├── 2
│   │   └── pt_2
│   │       ├── ID.col
│   │       └── x.col
│   └── 5
│       └── ptxxxx_3
│           ├── ID.col
│           └── x.col
├── 3_4
│   ├── 2
│   │   └── pt_2
│   │       ├── ID.col
│   │       └── x.col
│   │
```

```
│   ├── 5
│   │   └── ptxxxx_3
│   │       ├── ID.col
│   │       └── x.col
│   ├── 7
│   │   └── ptaaaa_4
│   │       ├── ID.col
│   │       └── x.col
│   ├── 9
│   │   └── ptbbbb_5
│   │       ├── ID.col
│   │       └── x.col
│   └── b
│       └── ptcccc_6
│           ├── ID.col
│           └── x.col
├── 4_5
│   ├── 2
│   │   └── pt_2
│   │       ├── ID.col
│   │       └── x.col
│   ├── 5
│   │   └── ptxxxx_3
│   │       ├── ID.col
│   │       └── x.col
│   ├── 7
│   │   └── ptaaaa_4
│   │       ├── ID.col
│   │       └── x.col
│   ├── 9
│   │   └── ptbbbb_5
│   │       ├── ID.col
│   │       └── x.col
│   └── b
│       └── ptcccc_6
│           ├── ID.col
│           └── x.col
├── 6_7
│   ├── 7
│   │   └── ptaaaa_4
│   │       ├── ID.col
│   │       └── x.col
│   ├── 9
│   │   └── ptbbbb_5
│   │       ├── ID.col
│   │       └── x.col
│   └── b
```

```
│       └── ptcccc_6
│           ├── ID.col
│           └── x.col
├── 7_8
│   ├── 2
│   │   └── pt_2
│   │       ├── ID.col
│   │       └── x.col
│   ├── 5
│   │   └── ptxxxx_3
│   │       ├── ID.col
│   │       └── x.col
│   ├── 7
│   │   └── ptaaaa_4
│   │       ├── ID.col
│   │       └── x.col
│   ├── 9
│   │   └── ptbbbb_5
│   │       ├── ID.col
│   │       └── x.col
│   └── b
│       └── ptcccc_6
│           ├── ID.col
│           └── x.col
├── 8_9
│   ├── 2
│   │   └── pt_2
│   │       ├── ID.col
│   │       └── x.col
│   ├── 5
│   │   └── ptxxxx_3
│   │       ├── ID.col
│   │       └── x.col
│   ├── 7
│   │   └── ptaaaa_4
│   │       ├── ID.col
│   │       └── x.col
│   ├── 9
│   │   └── ptbbbb_5
│   │       ├── ID.col
│   │       └── x.col
│   └── b
│       └── ptcccc_6
│           ├── ID.col
│           └── x.col
├── 9_10
│   ├── 7
```

```
│   │   └── ptaaaa_4
│   │       ├── ID.col
│   │       └── x.col
│   ├── 9
│   │   └── ptbbbb_5
│   │       ├── ID.col
│   │       └── x.col
│   └── b
│       └── ptcccc_6
│           ├── ID.col
│           └── x.col
├── dolphindb.lock
├── domain
├── ptaaaa.tbl
├── ptbbbb.tbl
├── ptcccc.tbl
├── pt.tbl
└── ptxxxx.tbl

79 directories, 77 files
```

```
db = database("dfs://rangedb", VALUE, 2000.01M .. 2016.12M) => It is not allow
ed to overwrite an existing database.
```

2.7. table用法

table命令用法： table — DolphinDB 2.0 documentation

写法一：

```
id=`XOM`GS`AAPL
x=102.1 33.4 73.6
tt=table(id, x);
```

写法二：

```
tt=table(`XOM`GS`AAPL as id, 102.1 33.4 73.6 as x);
```

```
select * from tt;
id   x
---- ------------------
XOM  102.099999999999994
GS   33.399999999999998
AAPL 73.599999999999994
```

2.7.1. 写法三：

`table(capacity:size, colNames, colTypes)`

第二种用法中：

capacity 是正整数，表示建表时系统为该表分配的内存（以记录数为单位）。当记录数超过 capacity 时，系统首先会分配 capacity 1.2~2倍的新的内存空间，然后复制数据到新的内存空间，最后释放原来的内存。对于规模较大的表，此类操作的内存占用会很高。因此，建议创建内存表时预先分配一个合理的 capacity。

size 是正整数，表示该表新建时的行数。若 size =0，创建一个空表。 注意如果 colTypes 指定为数组向量， size 必须为0。

colNames 是一个向量，表示列名。

colTypes 是一个向量，表示每列的数据类型。可使用表示数据类型的系统保留字或相应的字符串。

```
tt=table(100:5, `name`id`value, [STRING,INT,DOUBLE]);
```

```
select * from tt;
name id value
---- -- -----
     0  0
     0  0
     0  0
     0  0
     0  0

select count(id) from tt;
count_id
--------
5
```

数据类型：数据类型 — DolphinDB 2.0 documentation

`SYMBOL` 是特殊的字符串类型。某个表字段定义为 SYMBOL 类型时，必须保证该字段的不同取值小于2097152(2^21)个，否则会报错“One symbase’s size can’t exceed 2097152”。

可以用 `SYMBOL` 来表示**股票代码。**

```
colNames=`stockid`date`time`open`high`low`close`turnover_volume`turnover_value
colTypes=[SYMBOL,DATE,MINUTE,DOUBLE,DOUBLE,DOUBLE,DOUBLE,INT,DOUBLE]
t=table(1:0,colNames,colTypes)
```