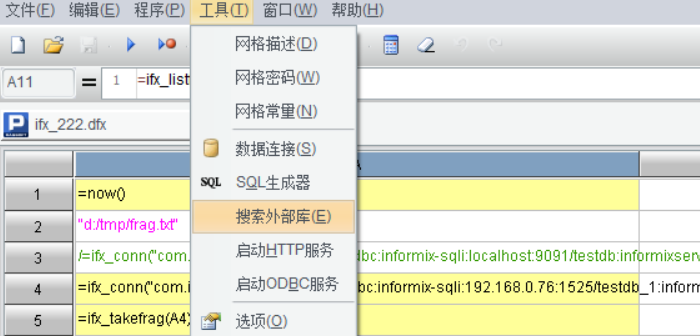
**InformixCli接口使用说明**

informixCli接口实现通过Linux下的本地管道读数据，从Informix数据库分区表中获取数据，可实现并发管道读取，加快数据的导出。对于多表joinx实现时，需要数据为有序的。此接口要求informix创建分区表。

1. InformixCli与集算器接口是在Informix jdbc及Linux下管道开发的基础上开发的，接口包文件为informixcli.jar。
2. 将informixcli.jar包拷贝到集算器安装目录lib下，其它的依赖包也放在lib下。
3. 软件集算器注册后，重启IDE软件在菜单“工具选项”中能看到“搜索外部库”，则进入“选择外部库”。  
   
4. **在集算器下IDE中选择部库界面：**



**5. jdk版本号问题**

informixcli要求运行jdk版本不低于1.7下面，集算器运行的若是jre1.6,在bin/config.txt中改为1.7,如用户的jdk在目录d:\java\jdk1.7.0\_71下

原：

java\_home=C:\Program Files (x86)\raqsoft\common;esproc\_port=41733;jvm\_args=-Xms128m -Xmx4866m -Duser.language=zh

改：

java\_home=d:\java\jdk1.7.0\_71;esproc\_port=41733;jvm\_args=-Xms256m -Xmx4866m -XX:PermSize=256M -XX:MaxPermSize=512M -Duser.language=zh

1. **informixcli接口函数列表：**   
   6.1 ifx\_open(url; fragfile)

url连接informix server的地址信息

fragfile为分片信息文件，记录分片表的分片字段及每个分片中此字段的最值  
返回值: ifxconn;

* 1. ifx\_conn(drv, url; fragfile)

drv,informix jdbc驱动名,常用为： com.informix.jdbc.IfxDriver

url; fragfile同上(6.1).

返回值: ifxconn。

6.3 ifx\_takefrag(ifxconn, table1,table2,...)

从数据库提取指定表的分片信息，并记录在ifxconn中， 若table1及以后省略则提取所有分片表。

6.4 ifx\_setfrag(ifxconn, table, fieldName, min1, min2,....)

设置表table的分片信息，若fieldName以后省略则删除IfxConn中保存的表table的分片信息。

min1, min2为每个分片的最小值。

6.5 ifx\_listfrag(ifxconn, table)

以序表的形式列出指定表的分片信息，若table省略则列出所有分片表的分片信息。

6.6 ifx\_savefrag(ifxconn, file)

内存中的分片信息并存入文件。

6.7 ifx\_cursor(ifxconn, table:where, f:alias,...; k1:k2)

根据where条件从表table的第k1到k2个分片中查询数据，f为表的列名，alias为其别名，where可省略,不支持省略k1时:k2方式。没有k2时，表示只查找k1分片。

**ifx\_cursor选项**@o, 对排序不作要求  
@f, 要求有序

@m, 多管道选项.

6.8 ifx\_cursor(ifxconn, sql; k1:k2)

sql语句从第k1到k2个分片中查询数据。

6.9 ifx\_cursor(ifxconn, table:where, f:alias,...; ifxCursor:f2)

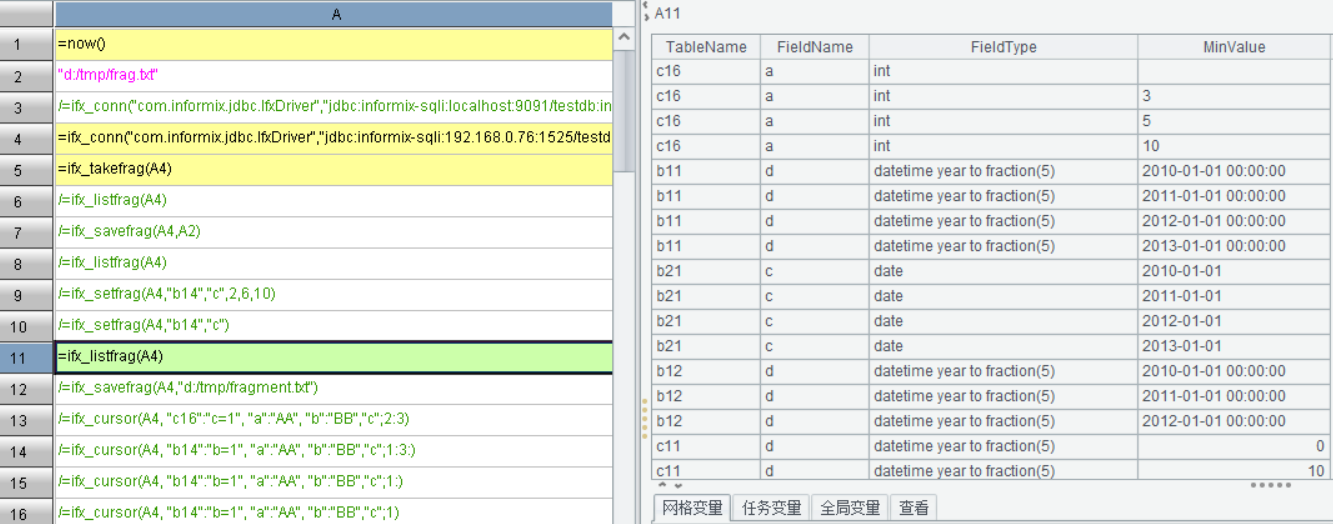
根据where条件从表table查询数据，f为表的列名，alias为其别名，跟ifxCursor取同样的分片数据,f2分片字段，若省略则与ifxCursor分片表的分片字段同名

6.10 ifx\_cursor(ifxconn, sql; ifxCursor:f2)

从sql语句查询数据，跟ifxCursor取同样的分片数据,f2分片字段，若省略则与ifxCursor分片表的分片字段同名

6.11 ifx\_close(ifxconn)

关闭连接



7．依赖包文件列表：

ifxjdbc.jar

ifxlang.jar

ifxsqlj.jar

8．样例1：

|  |  |
| --- | --- |
|  | **A** |
| 1 | "d:/tmp/frag.txt" |
| 2 | =ifx\_conn("com.informix.jdbc.IfxDriver","jdbc:informix-sqli:192.168.0.76:1525/testdb\_1:informixserver=tramsserver&user=informix&pwd=informix";A1) |
| 3 | =ifx\_takefrag(A2) |
| 4 | =ifx\_listfrag(A2) |
| 5 | =ifx\_savefrag(A3,A2) |
| 6 | =ifx\_listfrag(A3) |
| 7 | =ifx\_setfrag(A3,"b14","c",2,6,10) |
| 8 | =ifx\_setfrag(A3,"b14","c") |
| 9 | =ifx\_listfrag(A3) |
| 10 | =ifx\_cursor(A3, "b11":"c=1", "a":"AA", "b":"BB","c";2:3) |
| 11 | =ifx\_cursor(A3, "c11":"c=1",;A10:"d") |
| 12 | =[A11].mcursor().groups(;count(1):count\_order) |
| 13 | =ifx\_close(A3) |
|  |  |

样例2：

|  |  |
| --- | --- |
|  | **A** |
| 1 | "d:/tmp/frag.txt" |
| 2 | =ifx\_conn("com.informix.jdbc.IfxDriver","jdbc:informix-sqli:192.168.0.76:1525/testdb\_1:informixserver=tramsserver&user=informix&pwd=informix";A2) |
| 3 | /=ifx\_conn("com.informix.jdbc.IfxDriver","jdbc:informix-sqli://192.168.0.7:8000/tpch:informixserver=runqian&user=informix&pwd=informix";"/tmp/tpch\_fragment.txt") |
| 4 | =ifx\_takefrag(A2) |
| 5 | =ifx\_cursor(A2,"select L\_Returnflag,L\_Linestatus,L\_Quantity,L\_Extendedprice,L\_Discount,L\_Tax From Lineitem Where L\_Shipdate<=Date('1998-12-01') - interval(90) day to day";1:2) |
| 6 | =ifx\_cursor(A2,"select L\_Returnflag,L\_Linestatus,L\_Quantity,L\_Extendedprice,L\_Discount,L\_Tax From Lineitem Where L\_Shipdate<=Date('1998-12-01') - interval(90) day to day";3:4) |
| 7 | =ifx\_cursor(A2,"select L\_Returnflag,L\_Linestatus,L\_Quantity,L\_Extendedprice,L\_Discount,L\_Tax From Lineitem Where L\_Shipdate<=Date('1998-12-01') - interval(90) day to day";5:6) |
| 8 |  |
| 9 | /=[A4,A5,A6,A7,A8,A9,A11].mcursor().groups(;sum(1):TotalScore) |
| 10 | =[A5,A6,A7].groups@m(l\_returnflag,l\_linestatus; sum(l\_quantity):sum\_qty,sum(l\_extendedprice):sum\_base\_price, sum(l\_extendedprice \* (1 - l\_discount)):sum\_disc\_price,sum(l\_extendedprice \* (1 - l\_discount) \* (1 + l\_tax)):sum\_charge, avg(l\_quantity):avg\_qty,avg(l\_extendedprice):avg\_price,avg(l\_discount):avg\_disc,count(1):count\_order) |
| 11 |  |
| 12 |  |
| 13 | =ifx\_close(A2) |
|  |  |

**问题说明：**

1. 对于字符串解析可能出现乱码，它需要与db系统的字符集信息绑定解析。管道导出数据的信息与jdbc的不一样，暂时没有实现。
2. 查询的数据量太少，管道写入数据很快结束退出，java没有来得及读取，此时程序会报异常.
3. **Informix分片规则：**  
   A round-robin 轮转法    
   B表达式expression   
   C remainder

D 列表 LIST  
E 固定间隔 INTERVAL  
除了规则A不适合本接口使用的外，其它采用实现了基本的功能。使用中若发现与规则不同时，可自己写sql语句与规则保持一致.

1. 表分片与读取管道没有关系. 每个ifx\_cursor对应一个管道，与分片无关。每个ifx\_cursor可以对应一个或多个表分片. ifx\_cursor可组成多游标来访问
2. 开了Informix PDQ之后，不管是一个管道，还是多个管道，数据都是无序的。ifx\_cursor@o参数能打开PDQ选项
3. 对于大表数据来说，加上order by可能出现临时表空间不足问题，执行带order by的sql可能出错.