

SPL Base

集算器教案

递归



CONTENTS



01

递归字段

02

递归合并

03

递归目录

04

单条记录的递归引用

05

多条记录的递归引用

06

函数递归

07

脚本递归

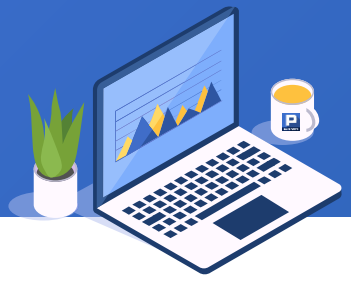
CONTENTS



递归字段



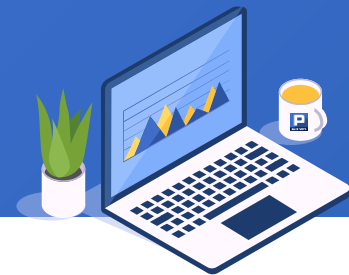
◆ 递归字段



下面是按年月分组的JSON格式销售数据，求2016年的总销售额。

```
[
  {YEAR:2016,MONTH:1,SALES:[
    {ORDERNUMBER:10101, ORDERLINENUMBER:4, SALES:3782, ...},
    {ORDERNUMBER:10102, ORDERLINENUMBER:1, SALES:3773.38, ...},
    ...]
  },
  {YEAR:2016,MONTH:2,SALES:[
    {ORDERNUMBER:10105, ORDERLINENUMBER:2, SALES:7208 ...},
    {ORDERNUMBER:10106, ORDERLINENUMBER:15, SALES:8690.36, ...},
    ...]
  },
  ...]
```

递归字段



SPL如下，其中用到了A.field@r()函数来递归获取字段值：

| | A | B |
|---|----------------------------------|----------------------------|
| 1 | =json(file("sales.json").read()) | /导入JSON数据文件 |
| 2 | =A1.select(YEAR=2016) | /选出2016年的销售数据 |
| 3 | =A2.field@r("SALES") | /使用A.field()函数的@r选项递归获取销售额 |
| 4 | =A3.(~.sum()).sum() | /先把A3每组明细销售额汇总，再求总销售额 |

| A3 | Member |
|-------------------------------|---------|
| [3782,3773.38,1404, ...] | 3782 |
| [7208,8690.36,4566.05,...] | 3773.38 |
| [5265.15,6130.35,3485.82...] | 1404 |
| [2793.86,9264.86,2082.49,...] | ... |

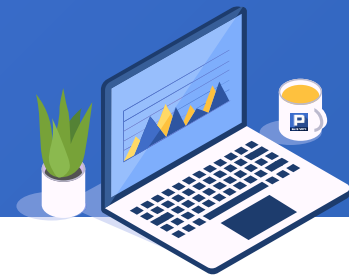
| A4 | Member |
|----|------------|
| | 1252700.43 |

CONTENTS



递归合并

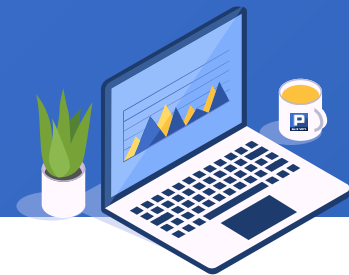
✦ 递归合并



下面是某时刻，新型冠状病毒世界各地确诊人数的JSON数据，要统计世界确诊人数。

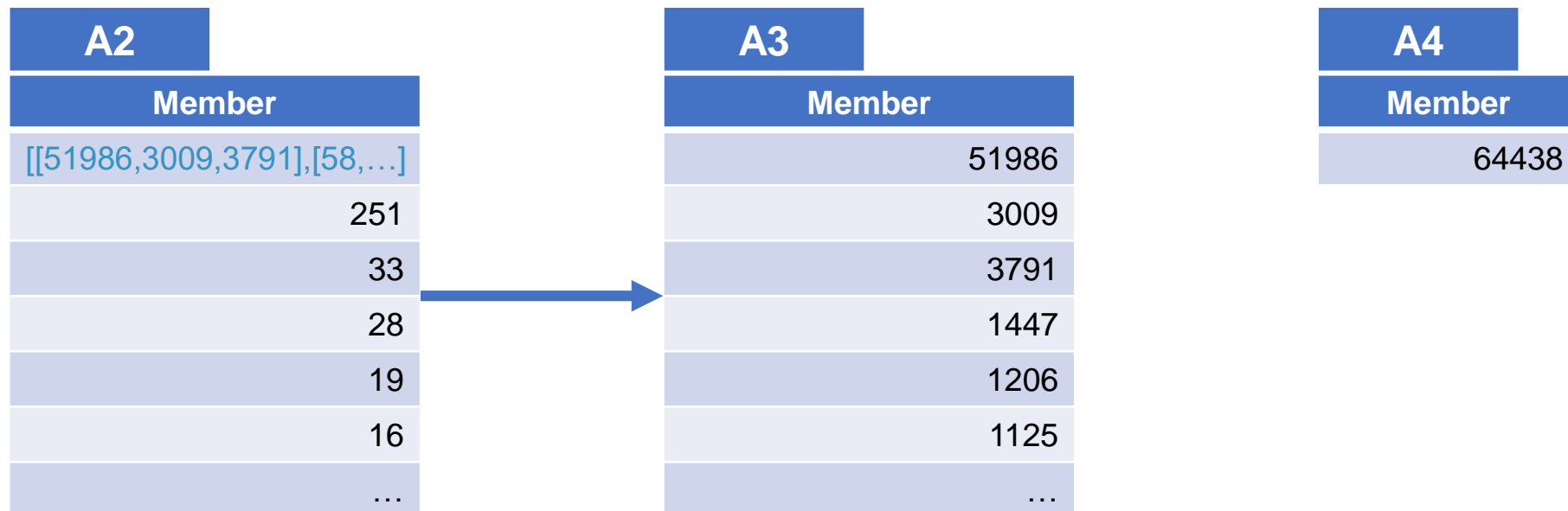
```
[
  {Region:"China",Confirmed:[
    {Region:"Hubei",Confirmed:[
      {Region:"Wuhan",Confirmed:51986},
      {Region:"Xiaogan",Confirmed:3009},
      {Region:"Huanggang",Confirmed:3791},
      ...]
    },
    {Region:"Taiwan",Confirmed:18},
    ...]
  },
  {Region:"Thailand",Confirmed:33},
  ...]
```

递归合并



SPL如下，其中用到了A.conj@r()函数来递归合并序列成员：

| | A | B |
|---|-------------------------------------|------------------------|
| 1 | =json(file("COVID-19.json").read()) | /导入JSON数据文件 |
| 2 | =A1.field@r("Confirmed") | /递归获取所有确诊字段 |
| 3 | =A2.conj@r() | /使用A.conj()函数的@r选项递归合并 |
| 4 | =A3.sum() | /确诊人数求和 |

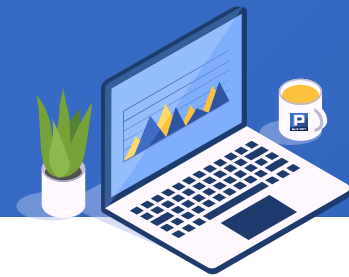


CONTENTS



递归目录

递归目录

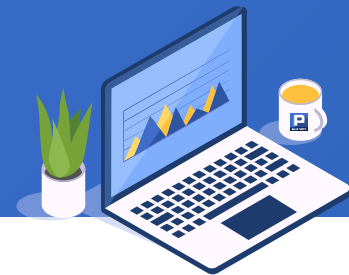


某小学调查学生在线教学可以使用的终端，现在要统计各终端占比。各班问卷及汇总目录如下：

| | |
|---|----------------|
| ▼ | Primary School |
| > | Grade1 |
| > | Grade2 |
| ▼ | Grade3 |
| | Class1 |
| | Class2 |
| | Class3 |
| | Class4 |
| | Class5 |
| | Class6 |
| > | Grade4 |
| > | Grade5 |
| > | Grade6 |

| ID | STUDENT_NAME | TERMINAL |
|-----|-----------------|--------------|
| 1 | Rebecca Moore | Phone |
| 2 | Ashley Wilson | Phone,PC,Pad |
| 3 | Rachel Johnson | Phone,PC,Pad |
| 4 | Emily Smith | PC,Pad |
| 5 | Ashley Smith | PC |
| 6 | Matthew Johnson | Phone |
| 7 | Alexis Smith | Phone,PC |
| 8 | Megan Wilson | Phone,PC,Pad |
| ... | ... | ... |

递归目录



SPL如下，其中用到了directory@s()函数来递归查找文件：

| | A | B | |
|---|---|---------------------------------|--------------------------|
| 1 | =directory@ps("D:/Primary School") | | /递归遍历目录，列出所有文件 |
| 2 | >totalCount=0 | | /定义变量totalCount，用来存储总记录数 |
| 3 | for A1 | =file(A3).xlsimport@t() | /循环导入各班级问卷excel文件 |
| 4 | | =B3.conj(TERMINAL.split@c()) B4 | /将终端按逗号拆分后，合并到B4的序列中 |
| 5 | | >totalCount+=B3.len() | /将每个班级人数加到totalCount中 |
| 6 | =B4.groups(~:TERMINAL;count(~)/totalCount:PERCENTAGE) | | /将B4的序列分组汇总，统计每个终端所占的百分比 |

| A1 |
|--|
| Member |
| D:\Primary School\Grade1\Class1\Investigation.xlsx |
| D:\Primary School\Grade1\Class2\Investigation.xlsx |
| D:\Primary School\Grade1\Class3\Investigation.xlsx |
| ... |

| B4 |
|--------|
| Member |
| Phone |
| Phone |
| PC |
| ... |

groups

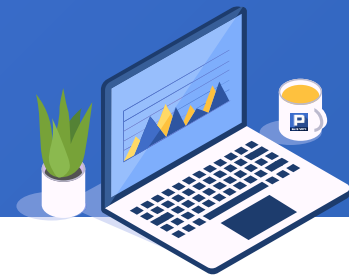
| A6 | TERMINAL | PERCENTAGE |
|----|----------|------------|
| | PC | 0.7 |
| | Pad | 0.567 |
| | Phone | 0.933 |

CONTENTS



单条记录的递归引用

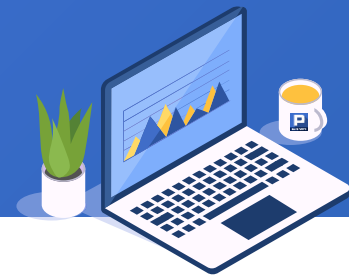
✦ 1. 递归查找引用



下面是某公司组织结构表，查询各部门的级别（总部是0级，分公司1级，依此类推）。

| ID | ORG_NAME | PARENT_ID |
|-----|------------------------|-----------|
| 1 | Head Office | 0 |
| 2 | Beijing Branch Office | 1 |
| 3 | Shanghai Branch Office | 1 |
| 4 | Chengdu Branch Office | 1 |
| 5 | Beijing R&D Center | 2 |
| ... | ... | ... |

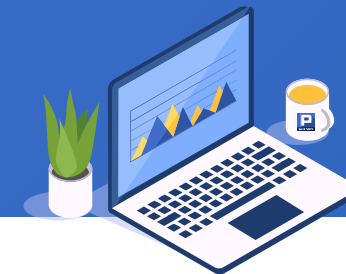
✦ 1. 递归查找引用



SPL如下，其中用到了prior函数来递归查找引用：

| | A | B |
|---|---|---|
| 1 | =connect("db") | /连接数据库 |
| 2 | =A1.query("select * from Organization") | /查询组织结构表 |
| 3 | >A2.switch(PARENT_ID,A2:ID) | /将PARENT_ID外键映射到该ID所在的记录，实现自连接。 |
| 4 | =A2.new(ID,ORG_NAME,~.prior(PARENT_ID).len()-1:LEVEL) | /创建由序号、部门名称和级别构成的新表。其中部门级别，是通过prior函数递归查找引用记录的层次数量计算得出。 |

✦ 1. 递归查找引用



A2

| ID | ORG_NAME | PARENT_ID |
|----|------------------------------|-----------|
| 1 | Head Office | 0 |
| 2 | Beijing Branch Office | 1 |
| 3 | Shanghai Branch Office | 1 |
| 4 | Chengdu Branch Office | 1 |
| 5 | Beijing R&D Center | 2 |
| 6 | Beijing Marketing Department | 2 |
| 7 | ... | ... |

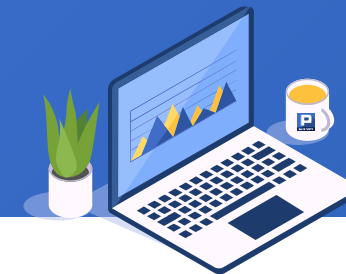
自连接 →

| ID | ORG_NAME | PARENT_ID |
|----|------------------------------|--------------------|
| 1 | Head Office | (null) |
| 2 | Beijing Branch Office | [1,Head Office,] |
| 3 | Shanghai Branch Office | [1,Head Office,] |
| 4 | Chengdu Branch Office | [1,Head Office,] |
| 5 | Beijing R&D Center | [2,Beijing Bra...] |
| 6 | Beijing Marketing Department | [2,Beijing Bra...] |
| 7 | ... | ... |

| ID | ORG_NAME | PARENT_ID |
|----|-----------------------|------------------|
| 2 | Beijing Branch Office | [1,Head Office,] |

| ID | ORG_NAME | PARENT_ID |
|----|-------------|-----------|
| 1 | Head Office | (null) |

✦ 1. 递归查找引用



A4

| ID | ORG_NAME | ~.prior(PARENT_ID) |
|----|------------------------------|-------------------------|
| 1 | Head Office | (null) |
| 2 | Beijing Branch Office | [1,Head Office,] |
| 3 | Shanghai Branch Office | [1,Head Office,] |
| 4 | Chengdu Branch Office | [1,Head Office,] |
| 5 | Beijing R&D Center | [2,Beijing Branch O...] |
| 6 | Beijing Marketing Department | [2,Beijing Branch O...] |
| 7 | ... | ... |

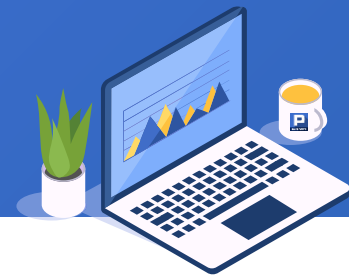


| ID | ORG_NAME | LEVEL |
|----|------------------------------|-------|
| 1 | Head Office | 0 |
| 2 | Beijing Branch Office | 1 |
| 3 | Shanghai Branch Office | 1 |
| 4 | Chengdu Branch Office | 1 |
| 5 | Beijing R&D Center | 2 |
| 6 | Beijing Marketing Department | 2 |
| 7 | ... | ... |



| ID | ORG_NAME | PARENT_ID |
|----|------------------------------|-------------------------|
| 6 | Beijing Marketing Department | [2,Beijing Branch O...] |
| 2 | Beijing Branch Office | [1,Head Office,] |
| 1 | Head Office | (null) |

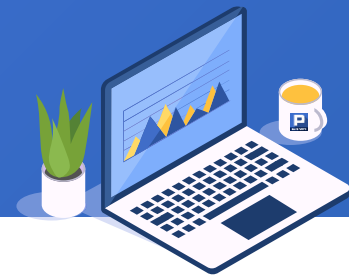
✦ 2. 递归查找引用直到指定值



下面是某公司组织结构表，查询北京分公司的下属机构，并列出其上级机构名称，多层的用逗号分隔。

| ID | ORG_NAME | PARENT_ID |
|-----|------------------------|-----------|
| 1 | Head Office | 0 |
| 2 | Beijing Branch Office | 1 |
| 3 | Shanghai Branch Office | 1 |
| 4 | Chengdu Branch Office | 1 |
| 5 | Beijing R&D Center | 2 |
| ... | ... | ... |

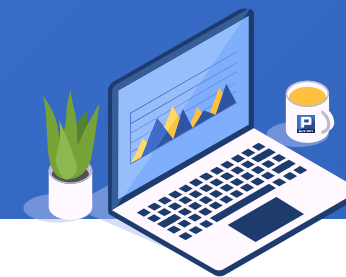
✦ 2. 递归查找引用直到指定值



SPL如下，其中用到了prior(F,r')函数来递归查找引用直到指定值：

| | A | B |
|---|--|---|
| 1 | =connect("db") | /连接数据库 |
| 2 | =A1.query("select * from Organization") | /查询组织结构表 |
| 3 | >A2.switch(PARENT_ID,A2:ID) | /将PARENT_ID外键映射到该ID所在的记录，实现自连接。 |
| 4 | =A2.select@1(ORG_NAME=="Beijing Branch Office") | /选出北京分公司所在记录 |
| 5 | =A2.new(ID,ORG_NAME,~.prior(PARENT_ID,A4):PARENT) | /创建由序号、部门名称和上级组成的新表。其中上级是通过prior函数递归查找北京分公司以下的记录得出。 |
| 6 | =A5.select(PARENT!=null) | /选出上级存在的成员，不存在的说明不在北京分公司下属。 |
| 7 | =A6.run(PARENT=PARENT.(PARENT_ID.ORG_NAME).concat@c()) | /将上级字段中所有的上级名称拼在一起，用逗号分隔。 |

✦ 2. 递归查找引用直到指定值

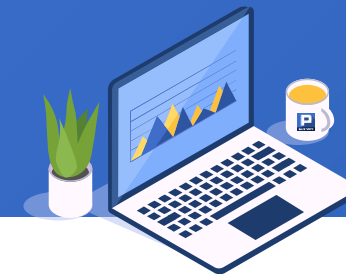


| A5 | | |
|----|---------------------------------|--|
| ID | ORG_NAME | PARENT |
| 1 | Head Office | (null) |
| 2 | Beijing Branch Office | [] |
| 3 | Shanghai Branch Office | (null) |
| 4 | Chengdu Branch Office | (null) |
| 5 | Beijing R&D Center | [[5,Beijing R&D Center,]] |
| 6 | Beijing Marketing Department | [[5,Beijing Marketing Department,]] |
| 7 | Beijing AI R&D Department | [[7,Beijing AI R&D Department,],[5,Beijing Marketing Department,]] |
| 8 | Beijing Internet R&D Department | [[8, Beijing Internet R&D Department,],[5,Beijing R&D Center,]] |
| 9 | ... | ... |

↓

| ID | ORG_NAME | PARENT_ID |
|----|---------------------------------|----------------------------|
| 8 | Beijing Internet R&D Department | [5,Beijing R&D Center,] |
| 5 | Beijing R&D Center | [2,Beijing Branch Office,] |

✦ 2. 递归查找引用直到指定值



| A6 | ID | ORG_NAME | PARENT |
|----|----|---------------------------------|--|
| | 2 | Beijing Branch Office | [] |
| | 5 | Beijing R&D Center | [[5,Beijing R&D Center,]] |
| | 6 | Beijing Marketing Department | [[5,Beijing Marketing Department,]] |
| | 7 | Beijing AI R&D Department | [[7,Beijing AI R&D Department,],[5,Beijing Marketing Department,]] |
| | 8 | Beijing Internet R&D Department | [[8, Beijing Internet R&D Department,],[5,Beijing R&D Center,]] |
| | 9 | ... | ... |

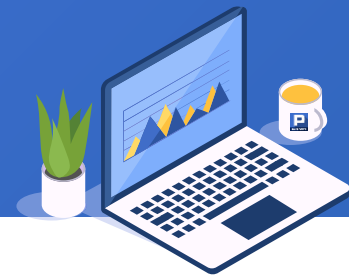
| A7 | ID | ORG_NAME | PARENT |
|----|----|---|--|
| | 2 | Beijing Branch Office | |
| | 5 | Beijing R&D Center | Beijing Branch Office |
| | 6 | Beijing Marketing Department | Beijing Branch Office |
| | 7 | Beijing AI R&D Department | Beijing R&D Center,Beijing Branch Office |
| | 8 | Beijing Internet R&D Department | Beijing R&D Center,Beijing Branch Office |
| | 9 | Beijing Internet Interface R&D department | Beijing Internet R&D Department,Beijing R&D Center,Beijing Branch Office |
| | 10 | Beijing Market Research Team | Beijing Marketing Department,Beijing Branch Office |

CONTENTS



多条记录的递归引用

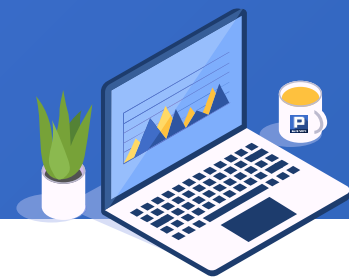
✦ 1. 只找上一级引用



下面是中国行政区划表，查询各行政区域的上级区域名称。

| ID | NAME | PARENT_ID |
|------|--------------|-----------|
| 1 | China | 0 |
| 11 | Beijing | 1 |
| 12 | Tianjin | 1 |
| 13 | Hebei | 1 |
| ... | ... | ... |
| 1301 | Shijiazhuang | 13 |
| 1302 | Tangshan | 13 |
| ... | ... | ... |

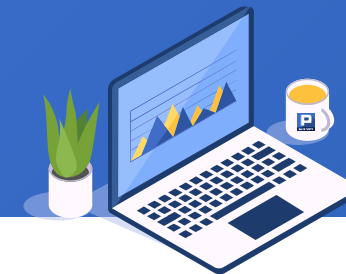
✦ 1. 只找上一级引用



SPL如下，其中用到了P.nodes(F)函数查找记录的上一级引用：

| | A | B |
|---|---|---------------------------------|
| 1 | =connect("db") | /连接数据库 |
| 2 | =A1.query("select * from ChinaRegion") | /查询中国行政区划表 |
| 3 | >A2.switch(PARENT_ID,A2:ID) | /将PARENT_ID外键映射到该ID所在的记录，实现自连接。 |
| 4 | =A2.nodes(PARENT_ID) | /使用nodes函数查找记录的上一级引用 |
| 5 | =A4.new(ID,NAME,PARENT_ID.NAME:PARENT_NAME) | /创建由序号、名称和上一级区域名称组成的序表。 |

✦ 1. 只找上一级引用

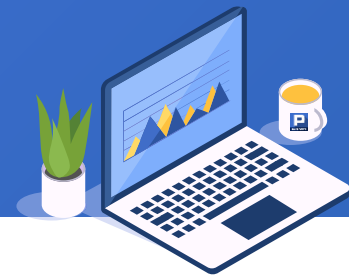


| A4 | | |
|--------|------------------|----------------------|
| ID | NAME | PARENT_ID |
| 1 | China | (null) |
| 11 | Beijing | [1,China,] |
| 12 | Tianjin | [1,China,] |
| 13 | Hebei | [1,China,] |
| ... | ... | ... |
| 1301 | Shijiazhuang | [13,Hebei,] |
| 1302 | Tangshan | [13,Hebei,] |
| 1303 | Qinhuangdao | [13,Hebei,] |
| ... | ... | ... |
| 130102 | Changan District | [1301,Shijiazhuang,] |
| 130104 | Qiaoxi District | [1301,Shijiazhuang,] |
| 130105 | Xinhua District | [1301,Shijiazhuang,] |
| ... | ... | ... |



| A5 | | |
|--------|------------------|--------------|
| ID | NAME | PARENT_NAME |
| 1 | China | (null) |
| 11 | Beijing | China |
| 12 | Tianjin | China |
| 13 | Hebei | China |
| ... | ... | ... |
| 1301 | Shijiazhuang | Hebei |
| 1302 | Tangshan | Hebei |
| 1303 | Qinhuangdao | Hebei |
| ... | ... | ... |
| 130102 | Changan District | Shijiazhuang |
| 130104 | Qiaoxi District | Shijiazhuang |
| 130105 | Xinhua District | Shijiazhuang |
| ... | ... | ... |

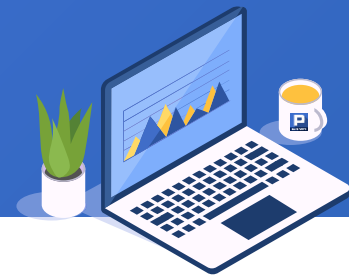
✦ 2. 递归查找引用直到指定值



下面是中国行政区划表，查询河北省有哪些行政区。

| ID | NAME | PARENT_ID |
|------|--------------|-----------|
| 1 | China | 0 |
| 11 | Beijing | 1 |
| 12 | Tianjin | 1 |
| 13 | Hebei | 1 |
| ... | ... | ... |
| 1301 | Shijiazhuang | 13 |
| 1302 | Tangshan | 13 |
| ... | ... | ... |

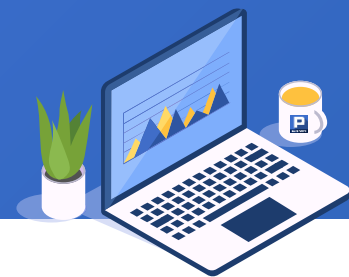
✦ 2. 递归查找引用直到指定值



SPL如下，其中用到了P.nodes(F,r)函数递归查找引用直到指定记录：

| | A | B |
|---|---|----------------------------------|
| 1 | =connect("db") | /连接数据库 |
| 2 | =A1.query("select * from ChinaRegion") | /查询中国行政区划表 |
| 3 | >A2.switch(PARENT_ID,A2:ID) | /将PARENT_ID外键映射到该ID所在的记录，实现自连接。 |
| 4 | =A2.select@1(name=="Hebei") | /查找河北省所在记录 |
| 5 | =A2.nodes(PARENT_ID,A4) | /使用nodes函数递归查找引用直到PARENT_ID指向河北省 |
| 6 | =A5.new(ID,NAME,PARENT_ID.NAME:PARENT_NAME) | /创建由序号、名称和上一级区域名称组成的序表。 |

✦ 2. 递归查找引用直到指定值

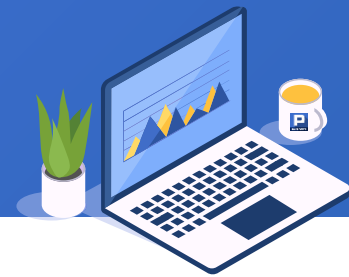


| A5 | | |
|--------|----------------------|----------------------|
| ID | NAME | PARENT_ID |
| 1301 | Shijiazhuang | [13,Hebei,] |
| 1302 | Tangshan | [13,Hebei,] |
| 1303 | Qinhuangdao | [13,Hebei,] |
| 1304 | Handan | [13,Hebei,] |
| 1305 | Xingtai | [13,Hebei,] |
| ... | ... | ... |
| 130102 | Changan District | [1301,Shijiazhuang,] |
| 130104 | Qiaoxi District | [1301,Shijiazhuang,] |
| 130105 | Xinhua District | [1301,Shijiazhuang,] |
| 130107 | Jingxing mining area | [1301,Shijiazhuang,] |
| 130108 | Yuhua District | [1301,Shijiazhuang,] |
| 130109 | Gaocheng District | [1301,Shijiazhuang,] |
| ... | ... | ... |



| A6 | | |
|--------|----------------------|--------------|
| ID | NAME | PARENT_NAME |
| 1301 | Shijiazhuang | Hebei |
| 1302 | Tangshan | Hebei |
| 1303 | Qinhuangdao | Hebei |
| 1304 | Handan | Hebei |
| 1305 | Xingtai | Hebei |
| ... | ... | ... |
| 130102 | Changan District | Shijiazhuang |
| 130104 | Qiaoxi District | Shijiazhuang |
| 130105 | Xinhua District | Shijiazhuang |
| 130107 | Jingxing mining area | Shijiazhuang |
| 130108 | Yuhua District | Shijiazhuang |
| 130109 | Gaocheng District | Shijiazhuang |
| ... | ... | ... |

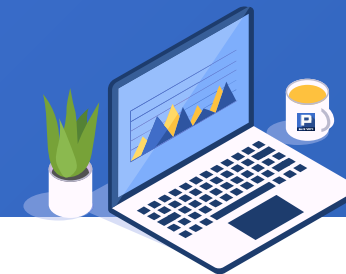
✦ 3. 只找叶子



下面是中国行政区划表，查询河北省有哪些区县。

| ID | NAME | PARENT_ID |
|------|--------------|-----------|
| 1 | China | 0 |
| 11 | Beijing | 1 |
| 12 | Tianjin | 1 |
| 13 | Hebei | 1 |
| ... | ... | ... |
| 1301 | Shijiazhuang | 13 |
| 1302 | Tangshan | 13 |
| ... | ... | ... |

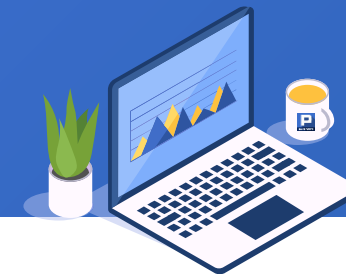
✦ 3. 只找叶子



SPL如下，其中用到了P.nodes@d(F,r)函数递归所有叶子（没有被其他记录引用的记录）：

| | A | B |
|---|---|---|
| 1 | =connect("db") | /连接数据库 |
| 2 | =A1.query("select * from ChinaRegion") | /查询中国行政区划表 |
| 3 | >A2.switch(PARENT_ID,A2:ID) | /将PARENT_ID外键映射到该ID所在的记录，实现自连接。 |
| 4 | =A2.select@1(name=="Hebei") | /查找河北省所在记录 |
| 5 | =A2.nodes@d(PARENT_ID,A4) | /使用nodes函数递归查找引用直到PARENT_ID指向河北省；使用@d选项只取叶子，在本例中会选出所有区县 |
| 6 | =A5.new(ID,NAME,PARENT_ID.NAME:PARENT_NAME) | /创建由序号、名称和上一级区域名称组成的序表。 |

✦ 3. 只找叶子

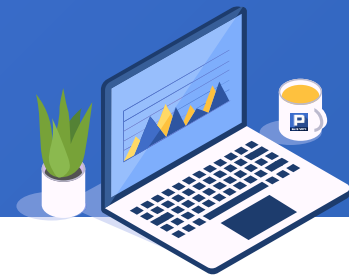


| A5 | | |
|--------|----------------------|----------------------|
| ID | NAME | PARENT_ID |
| 130102 | Changan District | [1301,Shijiazhuang,] |
| 130104 | Qiaoxi District | [1301,Shijiazhuang,] |
| 130105 | Xinhua District | [1301,Shijiazhuang,] |
| 130107 | Jingxing mining area | [1301,Shijiazhuang,] |
| 130108 | Yuhua District | [1301,Shijiazhuang,] |
| 130109 | Gaocheng District | [1301,Shijiazhuang,] |
| 130110 | Luquan District | [1301,Shijiazhuang,] |
| 130111 | Luancheng District | [1301,Shijiazhuang,] |
| 130121 | Jingxing County | [1301,Shijiazhuang,] |
| 130123 | Zhengding County | [1301,Shijiazhuang,] |
| 130125 | Xingtang County | [1301,Shijiazhuang,] |
| 130621 | Lingshou County | [1301,Shijiazhuang,] |
| ... | ... | ... |



| A6 | | |
|--------|----------------------|--------------|
| ID | NAME | PARENT_NAME |
| 130102 | Changan District | Shijiazhuang |
| 130104 | Qiaoxi District | Shijiazhuang |
| 130105 | Xinhua District | Shijiazhuang |
| 130107 | Jingxing mining area | Shijiazhuang |
| 130108 | Yuhua District | Shijiazhuang |
| 130109 | Gaocheng District | Shijiazhuang |
| 130110 | Luquan District | Shijiazhuang |
| 130111 | Luancheng District | Shijiazhuang |
| 130121 | Jingxing County | Shijiazhuang |
| 130123 | Zhengding County | Shijiazhuang |
| 130125 | Xingtang County | Shijiazhuang |
| 130621 | Lingshou County | Shijiazhuang |
| ... | ... | ... |

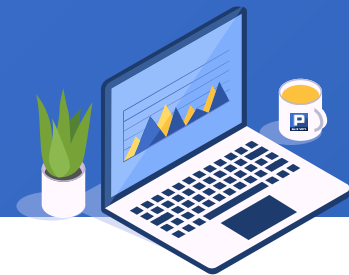
✦ 4. 查找所有上级引用



下面是中国行政区划表，将行政区名称拼上上级区域名称输出，多级用逗号分隔。例如石家庄输出：China,Hebei,Shijiazhuang

| ID | NAME | PARENT_ID |
|------|--------------|-----------|
| 1 | China | 0 |
| 11 | Beijing | 1 |
| 12 | Tianjin | 1 |
| 13 | Hebei | 1 |
| ... | ... | ... |
| 1301 | Shijiazhuang | 13 |
| 1302 | Tangshan | 13 |
| ... | ... | ... |

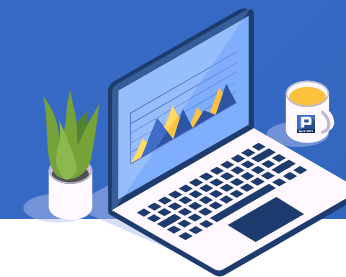
✦ 4. 查找所有上级引用



SPL如下，其中用到了P.nodes@p(F,r)函数查找所有上级引用：

| | A | B |
|---|--|---------------------------------|
| 1 | =connect("db") | /连接数据库 |
| 2 | =A1.query("select * from ChinaRegion") | /查询中国行政区划表 |
| 3 | >A2.switch(PARENT_ID,A2:ID) | /将PARENT_ID外键映射到该ID所在的记录，实现自连接。 |
| 4 | =A2.nodes@p(PARENT_ID) | /使用nodes函数的@p选项递归查找所有上级引用 |
| 5 | =A4.run(~=~.(NAME).concat@c()) | /将上级引用的名称拼在一起，使用逗号分隔 |

✦ 4. 查找所有上级引用



| A4 |
|---|
| Member |
| [[1,China,]] |
| [[1,China,],[11,Beijing,]] |
| [[1,China,],[12,Tianjin,]] |
| [[1,China,],[13,Hebei,]] |
| ... |
| [[1,China,],[13,Hebei,],[1301,Shijiazhuang,]] |
| [[1,China,],[13,Hebei,],[1302,Tangshan,]] |
| [[1,China,],[13,Hebei,],[1303,Qinhuangdao]] |
| ... |
| [[1,China,],[13,Hebei,],[1301,Shijiazhuang,],...] |
| [[1,China,],[13,Hebei,],[1301,Shijiazhuang,],...] |
| [[1,China,],[13,Hebei,],[1301,Shijiazhuang,],...] |
| ... |



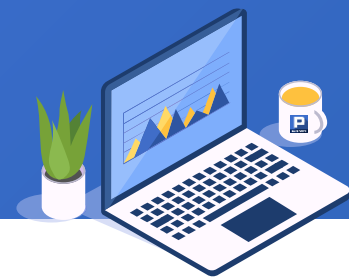
| A5 |
|---|
| Member |
| China |
| China,Beijing |
| China,Tianjin |
| China,Hebei |
| ... |
| China,Hebei,Shijiazhuang |
| China,Hebei,Tangshan |
| China,Hebei,Qinhuangdao |
| ... |
| China,Hebei,Shijiazhuang,Changan District |
| China,Hebei,Shijiazhuang,Qiaoxi District |
| China,Hebei,Shijiazhuang,Xinhua District |
| ... |

CONTENTS

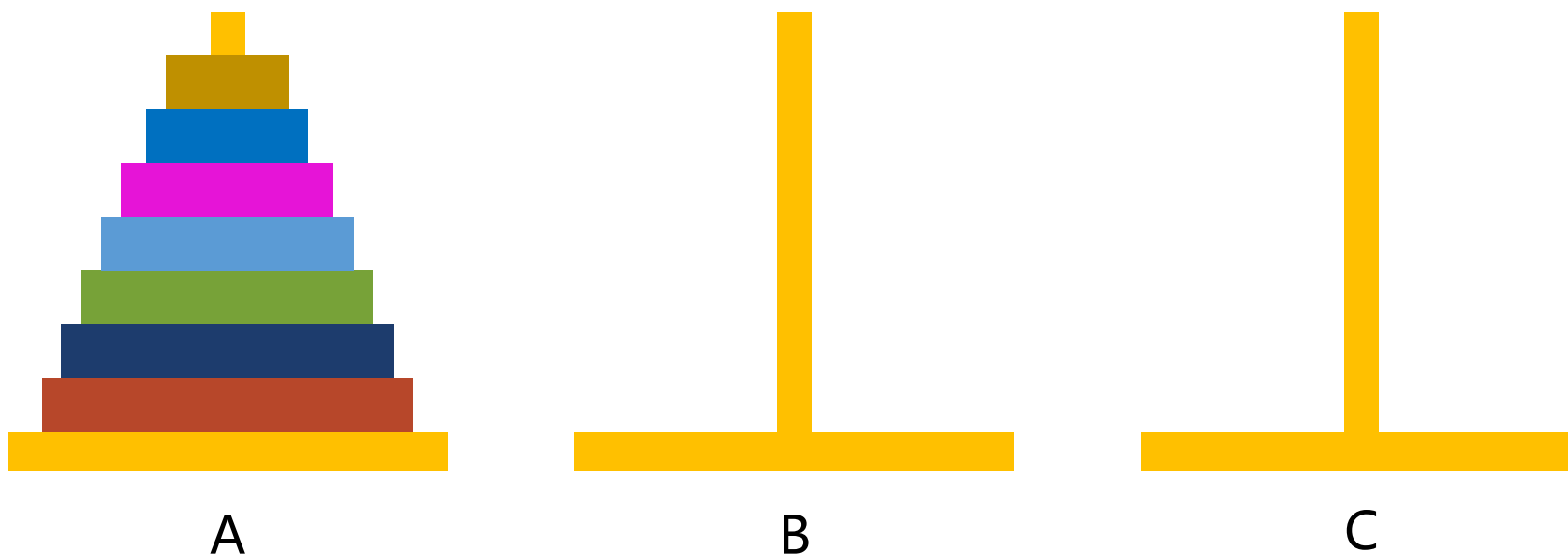


函数递归

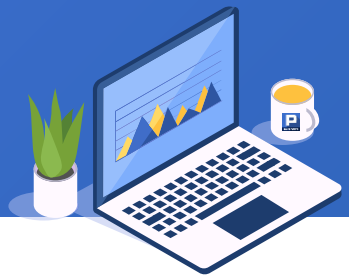
✦ 函数递归



汉诺塔问题是递归的经典问题。把A杆上的圆盘全部移到C杆上，并仍保持原有顺序叠好。每次移动一个圆盘，移动时要始终保持大盘在下，小盘在上。



✦ 函数递归

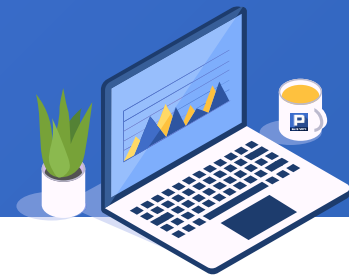


盘子从小到大依次命名为1到n。我们总是把n个盘子看成第n个盘子和n-1个盘子两组。把n-1个盘子移动到B，把第n个盘子移动到C，再把n-1个盘子移动到C即可。

SPL如下，其中用到了func(c,...)函数进行递归运算：

| | A | B | C | D |
|---|-------------------------|--------------|--|-------------------------------------|
| 1 | func | | | /定义函数 |
| 2 | | if(A1(1)==1) | >output("move dish " + string(A1) + " from " + B1 + " to " + D1) | /圆盘只有一个时移动到C |
| 3 | | else | >func(A1,[A1-1,B1,D1,C1]) | /将A上的n-1个盘子移动到B |
| 4 | | | >output("move dish " + string(A1) + " from " + B1 + " to " + D1) | /将A最下面的盘子移动到C |
| 5 | | | >func(A1,[A1-1,C1,B1,D1]) | /将B上的n-1个盘子移动到A |
| 6 | >func(A1,5,"A","B","C") | | | /函数有4个参数，分别是盘子个数（也是名称）、初始塔、中介塔和目标塔。 |

✦ 函数递归

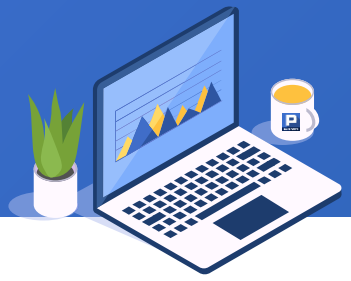


当A上有5个盘子时,
输出结果如右图:

```
move dish 1 from A to C
move dish 2 from A to B
move dish 1 from C to B
move dish 3 from A to C
move dish 1 from B to A
move dish 2 from B to C
move dish 1 from A to C
move dish 4 from A to B
move dish 1 from C to B
move dish 2 from C to A
move dish 1 from B to A
move dish 3 from C to B
move dish 1 from A to C
move dish 2 from A to B
move dish 1 from C to B
move dish 5 from A to C
move dish 1 from B to A
move dish 2 from B to C
move dish 1 from A to C
```

```
move dish 3 from B to A
move dish 1 from C to B
move dish 2 from C to A
move dish 1 from B to A
move dish 4 from B to C
move dish 1 from A to C
move dish 2 from A to B
move dish 1 from C to B
move dish 3 from A to C
move dish 1 from B to A
move dish 2 from B to C
move dish 1 from A to C
```

函数递归



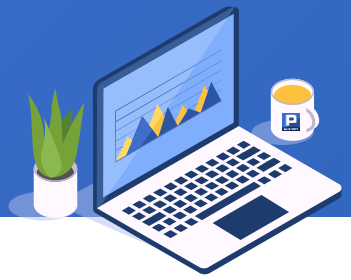
再来看一个海盗分金问题：

5个海盗抢得100枚金币，他们按抽签的顺序依次提方案：首先由1号提出分配方案，然后5人表决，投票要超过半数同意方案才被通过，否则他将被扔入大海喂鲨鱼，依此类推。



当只有两个海盗时，海盗5会否决海盗4的任何方案，独吞100金币。所以海盗4为了保命，会无条件同意海盗3方案。知道了海盗4的想法，贪婪的海盗3一定会给出[100,0,0]的分配方案。依此类推。

函数递归



SPL如下，其中用到了func(c,...)函数进行递归运算：

| | A | B | C | D | E |
|---|-----------------|----------------------------|-----------------------------------|----------------|---------------------|
| 1 | func | | | | |
| 2 | | if(A1==2) | return [-1,B1] | /只有2个海盗时 | /最后一个海盗全拿 |
| 3 | | =func(A1,A1-1,B1) | /使用func()函数，递归计算自己被否决后，剩下海盗会采用的方案 | | |
| 4 | | =B3.psort() | =A1/2 | /对下一个方案排序 | /除了自己还需要几个海盗支持 |
| 5 | | for B4.len() | if (B5 <= C4) | =B3(B4(B5))+=1 | /争取分配最少的海盗们支持，多给1金币 |
| 6 | | | | >B1-=D5 | /总金币里扣除分配的金币，剩下是自己的 |
| 7 | | | else | >B3(B4(B5))=0 | /剩下的海盗分配0金币 |
| 8 | | return B1 B3 | /返回自己剩下的金币和修改后的下一个方案，就是自己的分配方案。 | | |
| 9 | =func(A1,5,100) | /执行func()函数，参数是5个海盗，100个金币 | | | |

| A9 | Member |
|----|--------|
| | 97 |
| | 0 |
| | 1 |
| | 2 |
| | 0 |

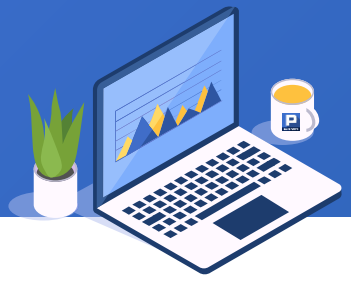
CONTENTS



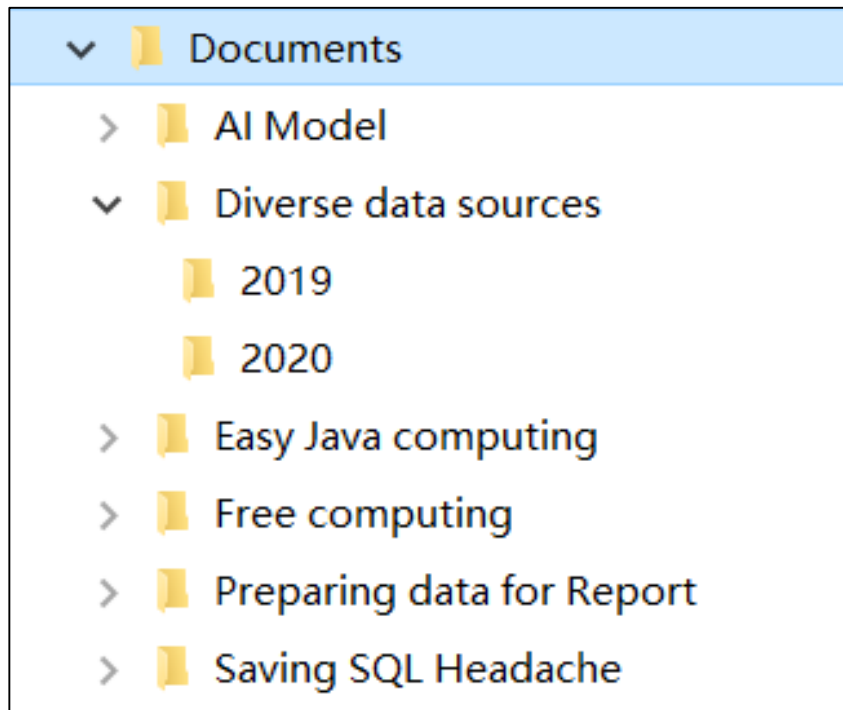
脚本递归



脚本递归



指定目录下有多级子目录以及上万个文本类型文件（txt、csv等）。想把每个文件中的指定行（第17行）汇总导出到一个文本文件中。下面是部分文件目录和文件：



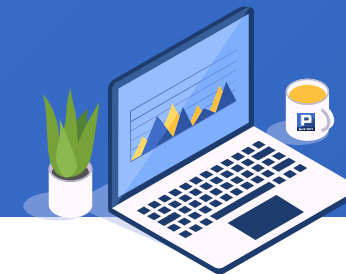
文本类型文件示例：

```
16 ...  
17 Middleware for report source data computing  
18 ...
```

汇总后result.txt：

```
Middleware for report source data computing  
SPL parsing and exporting Excel  
SQL Headaches Therapies – For Loop Operations  
The skill of updating database with esProc  
...
```

脚本递归



首先定义参数:

| ID | Name | Value | Remarks |
|----|------|-------|---------|
| 1 | path | | 文件目录 |

下面脚本用来实现读取文件指定行并输出到文本文件，保存为readfile.dfx。SPL如下:

| | A | B | C |
|---|----------------------------------|------------------------------|---------------------|
| 1 | =directory@p(path) | | /列出当前目录文件列表 |
| 2 | =A1.(file(~).cursor@s()) | | /循环打开文件游标 |
| 3 | =A2.((~.skip(16),~.fetch@x(1))) | | /每个文件游标跳过16行，取第17行 |
| 4 | =A3.union() | | /合并取出来的记录 |
| 5 | >file("result.txt").export@a(A4) | | /将结果写出到result.txt文件 |
| 6 | =directory@dp(path) | | /列出当前目录的子目录列表 |
| 7 | if A6.len()==0 | return | /如果没有子目录，程序返回 |
| 8 | else | =A6.(call("readfile.dfx",~)) | /如果有子目录，递归执行脚本 |

使用call()函数执行编辑好的readfile.dfx。SPL如下:

| | A | B |
|---|--------------------------------------|----------------|
| 1 | =call("readfile.dfx","D:/Documents") | /执行网格程序，参数指定目录 |

通过递归遍历目录和游标式读取文件，几乎不占用内存，适合大量文件和大数据场景。

THANKS

感谢观看

