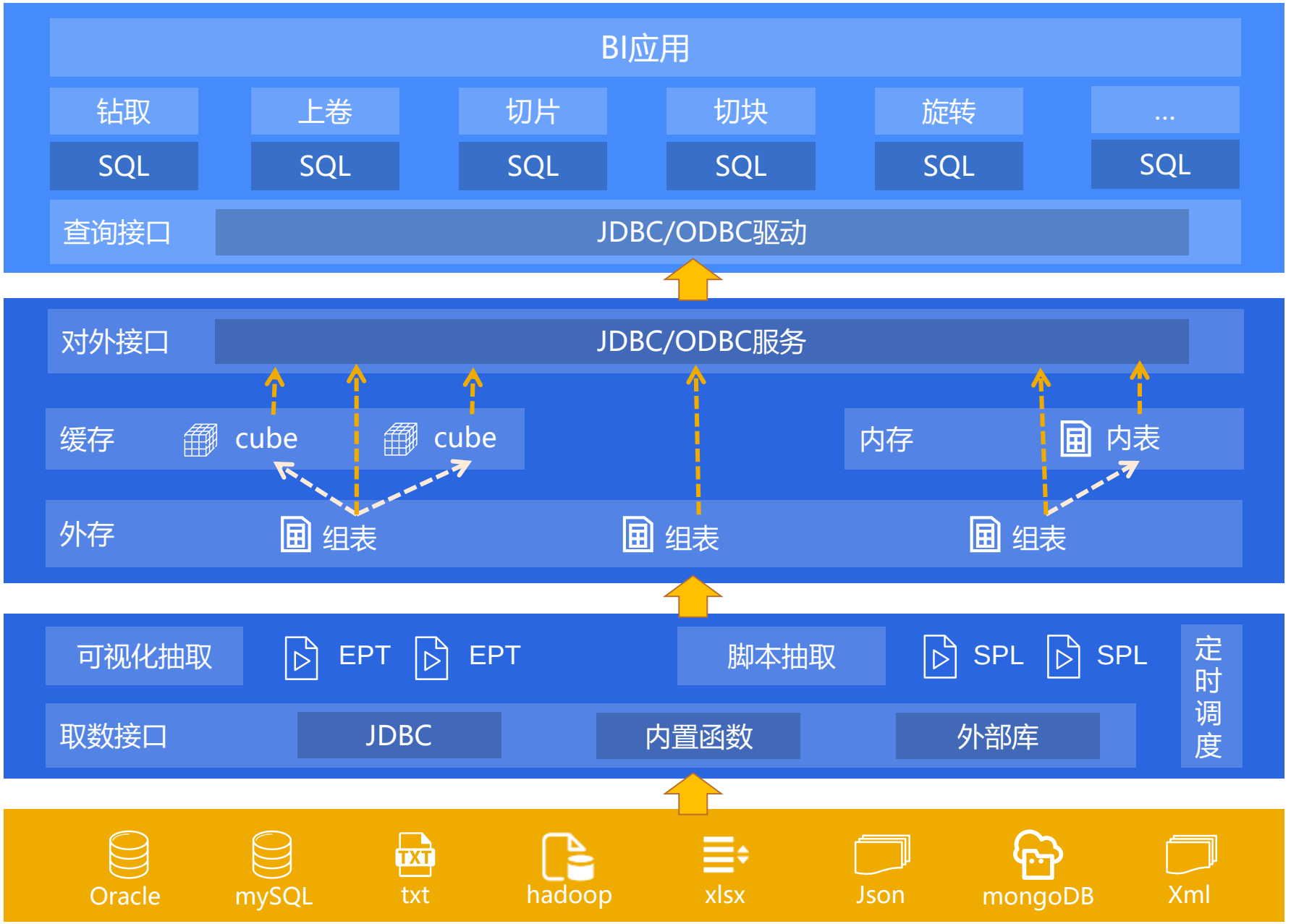
润乾
RAQSOFT

轻量级多维分析后台

集算器应用场景实施方案

www.raqsoft.com.cn

架构


多维分析前台
BI应用
钻取
上卷
切片
切块
旋转
...
SQL
SQL
SQL
SQL
SQL
SQL
查询接口
JDBC/ODBC驱动
多维分析后台
对外接口
JDBC/ODBC服务
缓存
cube
cube
内存
内表
外存
组表
组表
组表
抽取引擎
可视化抽取
EPT
EPT
脚本抽取
SPL
SPL
定时调度
取数接口
JDBC
内置函数
外部库
源数据
Oracle
mySQL
txt
hadoop
xlsx
Json
mongoDB
Xml

与传统RDB对比

相同点

- 支持多维分析相关的SQL语句
- 可实现钻取、切片等多维算法
- 提供JDBC/ODBC驱动
- 数据量远大于MOLAP

主要区别

- **存储格式**

  传统RDB方案：库表

  集算器方案：组表文件

- **计算性能**

  传统RDB方案：中低

  集算器方案：高

优势

集算器底层能力	多维分析后台特性	优势
文件存储	以文件形式存储多维数据	I/O性能高
高效压缩存储格式	存储密度高，占用空间少	硬件成本低
离散数据集模型	基础算法具备更高性能	计算性能高
预汇总、预关联、冷热路由	针对多维计算进行优化	计算性能进一步提高
支持文件系统	可按多级目录管理数据	目录管理灵活方便
多源计算能力	方便抽取特殊数据源 方便实现冷热路由	降低建设成本

目录 CONTENTS

01

常规宽表方案

宽表方案

为了提高计算性能，多维分析后台往往用空间换时间，将事实数据和维度数据存储于同一张表，这种存储方案称为宽表。

一个典型的宽表

sales_wide(事实表)	
key	orderdate(维度)
key	clientname（维度）
key	eid（维度）
	gender（维度）
	deptname（维度）
	clientprovince（维度）
	clientcity（维度）
	amount（测度）
	quantity（测度）

说明：如需显示销售姓名，应在前端用代码表反显。代码表不用于多维计算，不在存储方案的讨论范围。

宽表方案

将RDB中的宽表导出为集算器中的宽表，可使用下面的SPL脚本

	A	B
1	=connect@l("orcl").cursor@x("select orderdate,clientname,eid, gender,deptname,clientprovince,clientcity,amount,quantity from sales_wide order by orderdate,clientname,eid")	/用游标读取RDB表
2	=file("sales_wide.ctx").create@y(#orderdate,#clientname,#eid, gender,deptname,clientprovince,clientcity,amount,quantity)	/构造集算器宽表
3	=A2.append(A1)	/向宽表写入文件

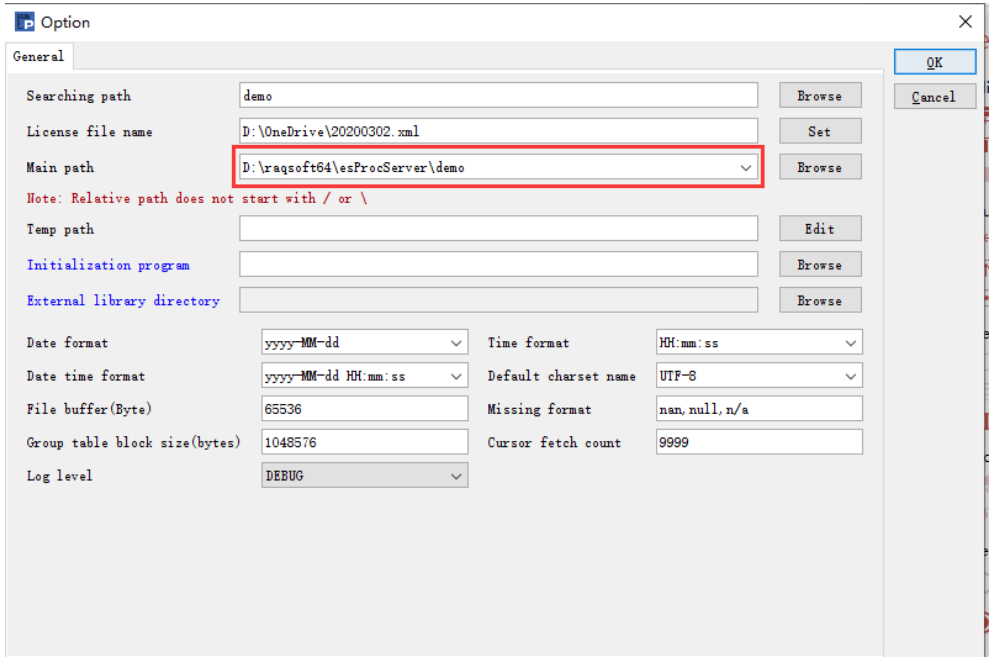
说明：SPL并非本章重点，详情可参考http://c.raqsoft.com.cn/article/1567908371148

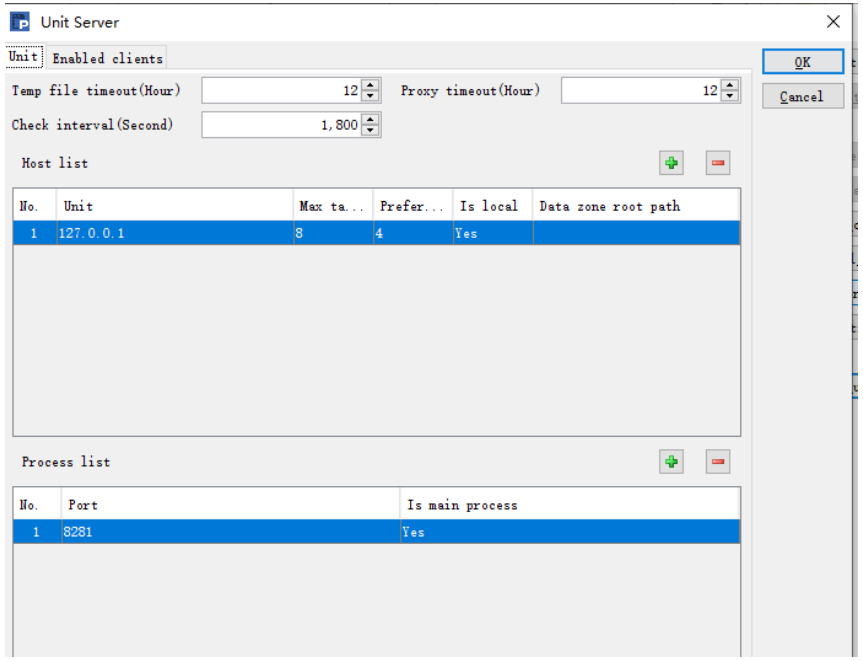
部署步骤

集算器同时支持JAVA（JDBC） BI工具和非JAVA（ODBC） BI工具，下面先讲宽表在前者的部署步骤。

1.配置JDBC集算服务

安装集算服务后，可通过esprocs.exe启动服务管理器（Linux下使用ServerConsole.sh）。管理器内置图形化配置工具，可配置通用选项信息（重点是主目录）和JDBC节点服务（重点是端口号）。





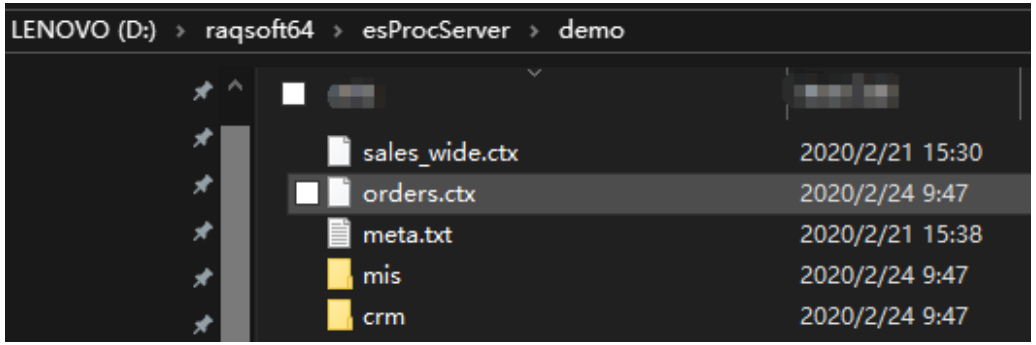
图形配置方法参考：http://doc.raqsoft.com.cn/esproc/tutorial/fuwuqi.html

手工配置方法参考：http://doc.raqsoft.com.cn/esproc/tutorial/pzraqsoftconfig.htm

部署步骤

2.部署数据和别名

在集算服务的主目录放置宽表，较多时可分目录存放。



meta.txt用来存放宽表的别名。对于sales_wide.ctx，meta.txt内容如下：

Table	File	Column	Type
sales_wide	sales_wide.ctx	orderdate	0
sales_wide	salse_wide.ctx	clientname	0
sales_wide	salse_wide.ctx	eid	0
sales_wide	sales_wide.ctx	gender	0
sales_wide	sales_wide.ctx	deptname	0
sales_wide	sales_wide.ctx	provincename	0
sales_wide	sales_wide.ctx	cityname	0
sales_wide	sales_wide.ctx	amount	0
sales_wide	sales_wide.ctx	quantity	0

type为字段类型，代码意义如下

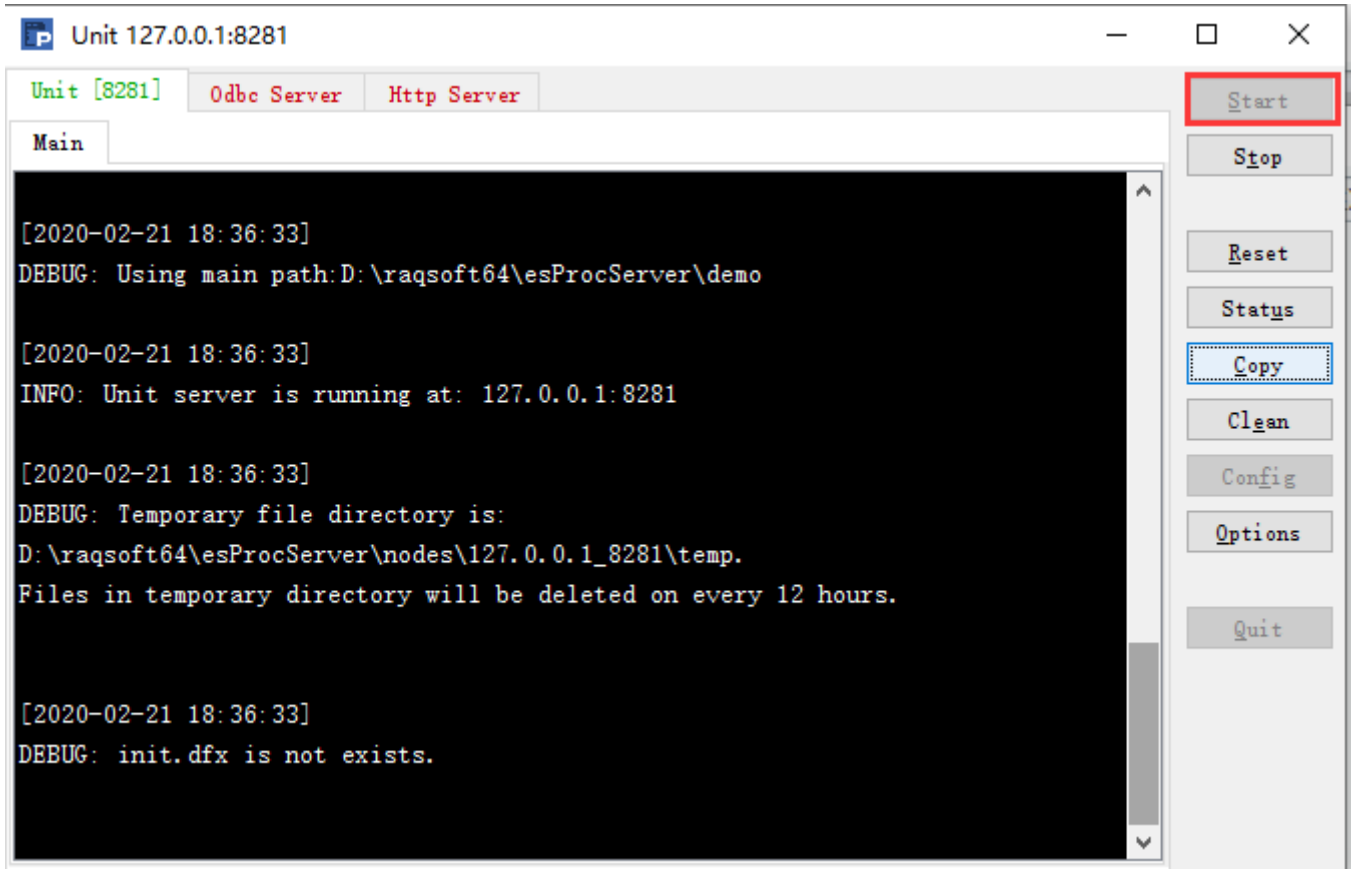
0：默认，自动识别
1：32 位整数
2：64 位长整数
3：16 位短整数
4：大整数
5：32 位浮点数
6：64 位双精度浮点数
7：十进制长实数
8：日期
9：时间
10：日期时间
11：字符串
12：布尔值
62：字节序列

说明：meta.txt不限于宽表，也适用于关联表其他数据格式，可参考http://c.raqsoft.com.cn/article/1578310507901

部署步骤

3.启动集算器JDBC服务

在服务管理中启动节点服务，即集算器JDBC服务。



部署步骤

4.部署集算器JDBC驱动

将驱动所需jar包/配置文件拷贝到java BI工具的类路径，即下面4个文件：

dm.jar	集算器计算引擎及JDBC驱动包
icu4j_3_4_5.jar	处理国际化
jdom.jar	解析配置文件
raqsoftConfig.xml	运行环境配置文件

raqsoftConfig.xml中需配置集算服务的地址和端口，如下：

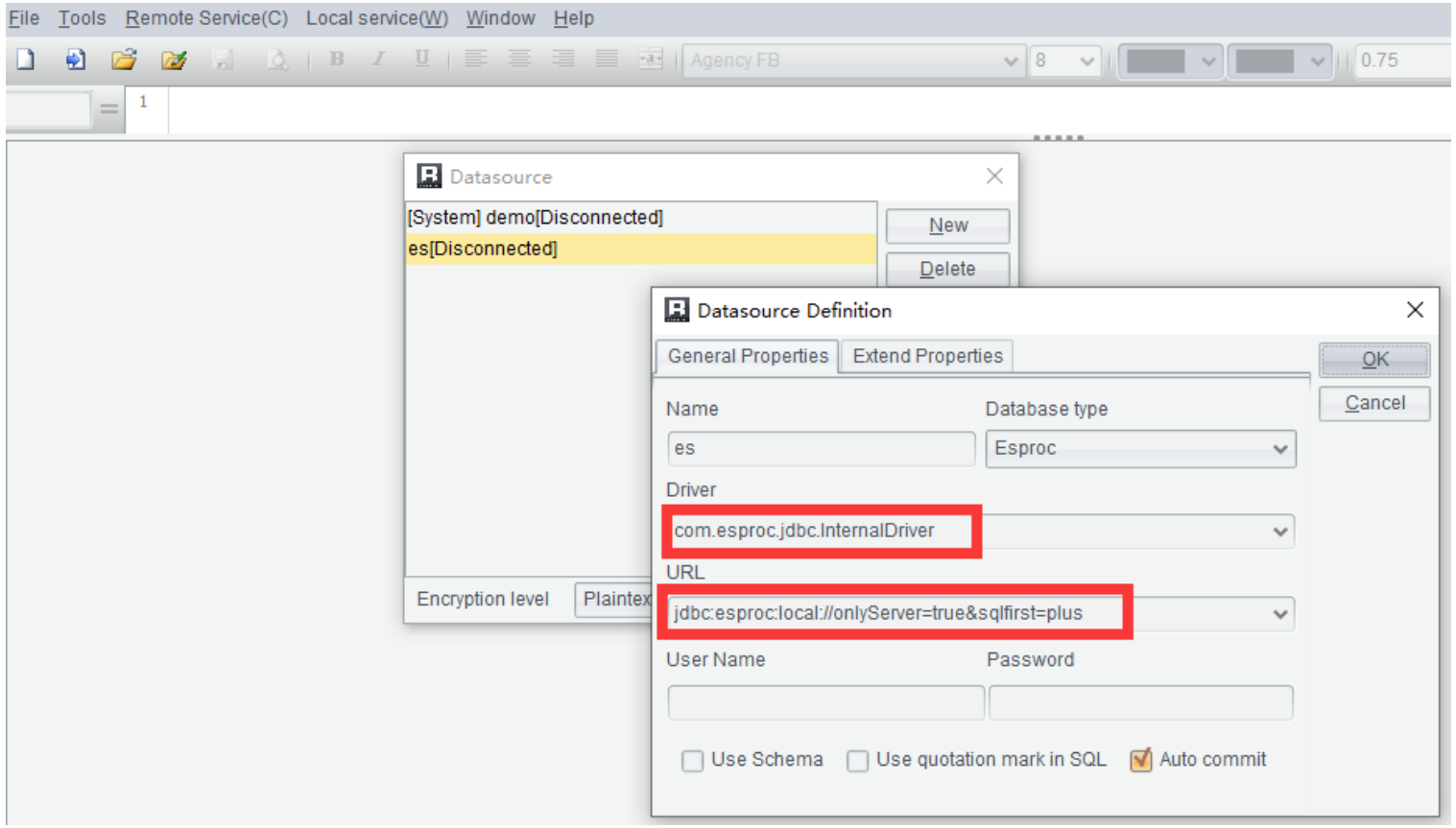
```
<JDBC>
        <load>Runtime,Server</load>
        <Units>
                <Unit>192.168.0.2:8281</Unit>
        </Units>
</JDBC>
```

说明：部署细节请参考http://doc.raqsoft.com.cn/esproc/tutorial/jdbcbushu.html

部署步骤

5.在BI工具中配置集算器JDBC连接

集算器支持主流JAVA BI工具，配置方式与传统ROLAP相同。下图以润乾报表为例：



集算器SQL与RDB SQL

集算器SQL和RDB SQL都符合ANSI SQL92规范，都可进行完备的结构化数据计算，都满足多维分析的需要，区别在于以下方面：

数据表的扩展名不同

RDB SQL：单一存储格式，无须扩展名

集算器SQL：多种存储格式应对不同的场景，其中多维分析使用的组表默认扩展名为ctx，可通过配置去掉扩展名。

对ANSI 92的扩展形式不同（多维分析用不到）

RDB SQL：有些RDB以窗口函数的形式扩展了SQL的功能

集算器SQL：以标记的形式扩展了SQL的功能（称为SQL+），可大幅提高计算性能。

说明：SQL+可参考http://doc.raqsoft.com.cn/esproc/func/sqljia.html

集算器还提供了计算能力更强的结构化计算语言SPL，参考http://c.raqsoft.com.cn/article/1567908371148

更多SQL示例

过滤

```
select ename,gender,amount,quantity from sales_wide.ctx where gender='F'
and amount>24000
```

排序

```
select ename,gender,amount from sales_wide.ctx order by amount
```

分组汇总

```
select year(orderdate),month(orderdate) ,sum(amount)as amount from
sales_wide.ctx group by year(orderdate),month(orderdate)
```

集合

```
select * from sales_wide.ctx where gender='F' and amount>24000
union
select * from sales_wide.ctx where clientname like '*picc*'
```

说明：如果配置了别名，可省略表的扩展名

目录 CONTENTS

02

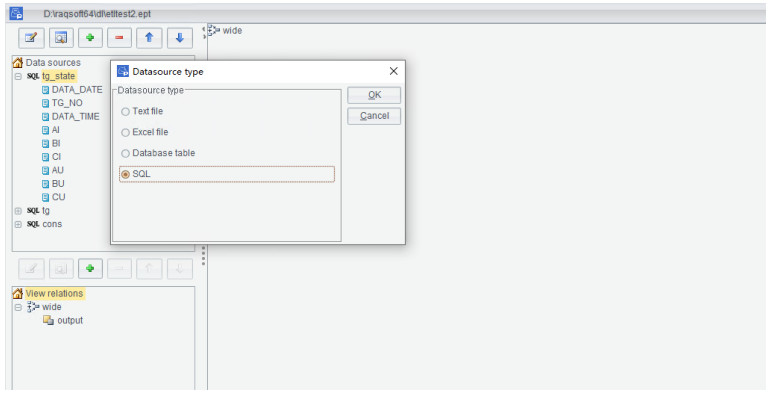
宽表ETL

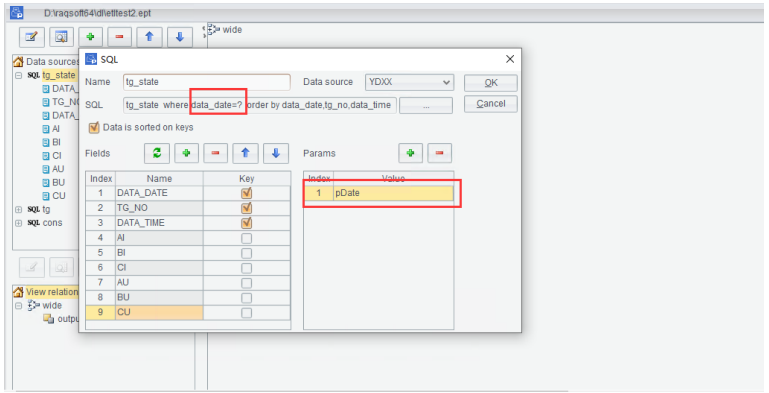
可视化ETL工具

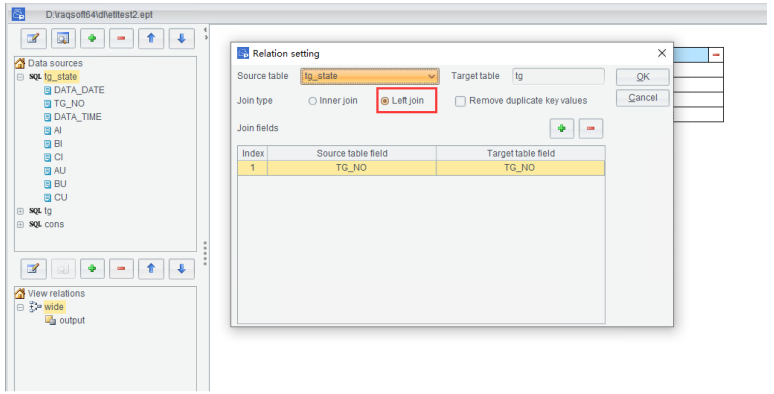
根据源数据的规整程度，集算器提供了2种ETL工具：可视化IDE、脚本IDE

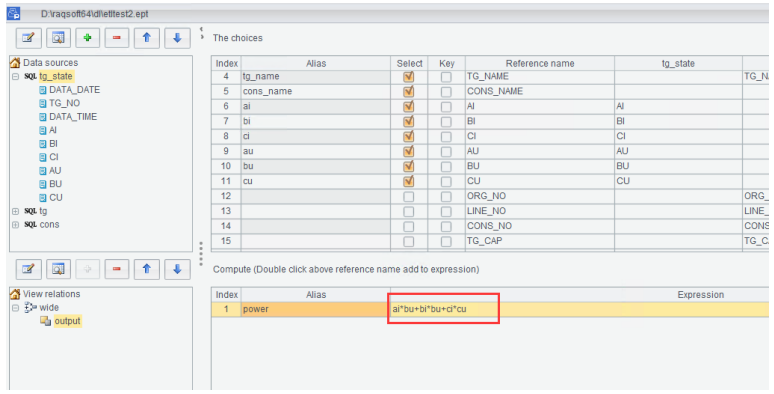
可视化工具适用场景：源数据较规整，ETL算法较简单

- 键值去重
- 类型转换
- 字段合并、拆分
- 计算列
- 跨源关联
- 生成宽表





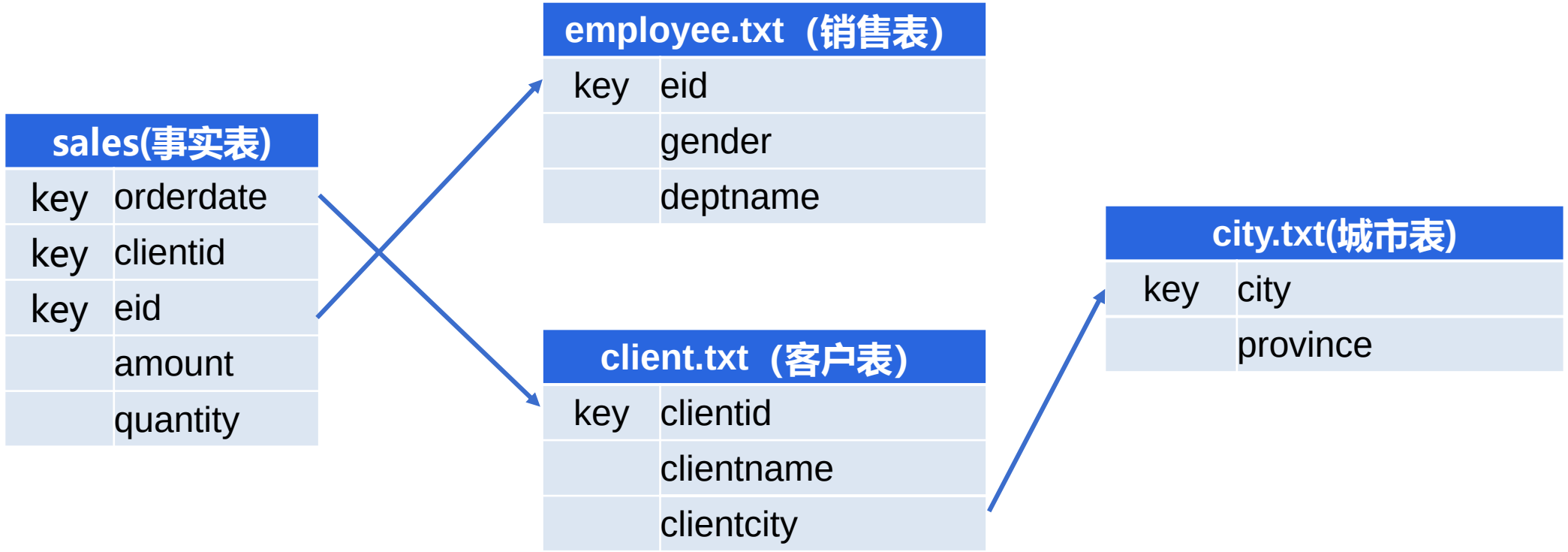




可视化生成宽表

目标：sales表位于mysql数据库，employee.txt、client.txt、city.txt位于文件系统，用集算器可视化ETL工具生成宽表sales_wide.ctx

源表逻辑结构



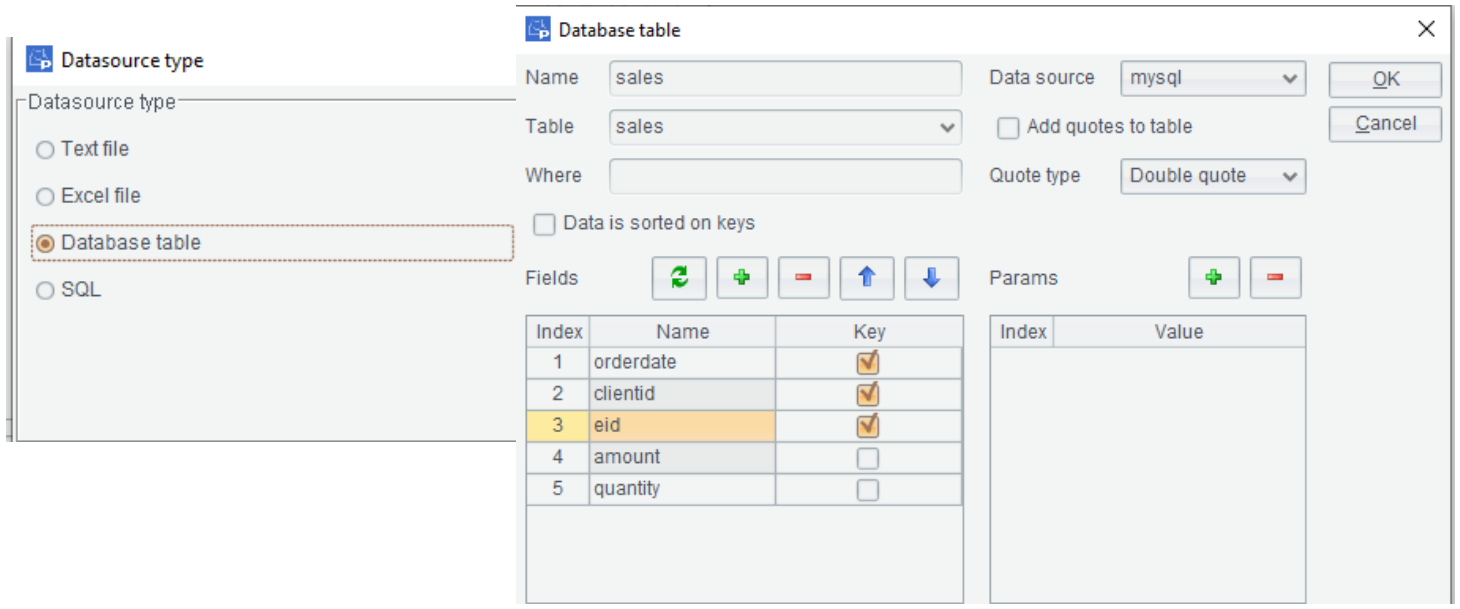
可视化生成宽表

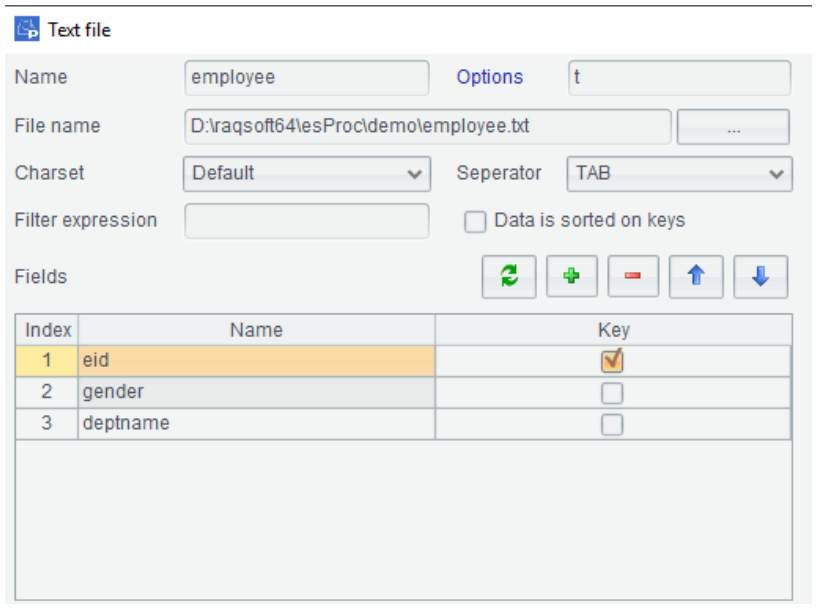
1.配置数据库数据源

使用bctx.exe启动可视化ETL工具，配置直向MySQL的数据源。文本文件无需配置

2.设置源表

在IDE的“Data sources”中设置源表，可以是文本文件、Excel、数据库表、SQL语句，可配置查询条件、外部参数等。依次配置4个源表，其中sales和employee.txt如下：





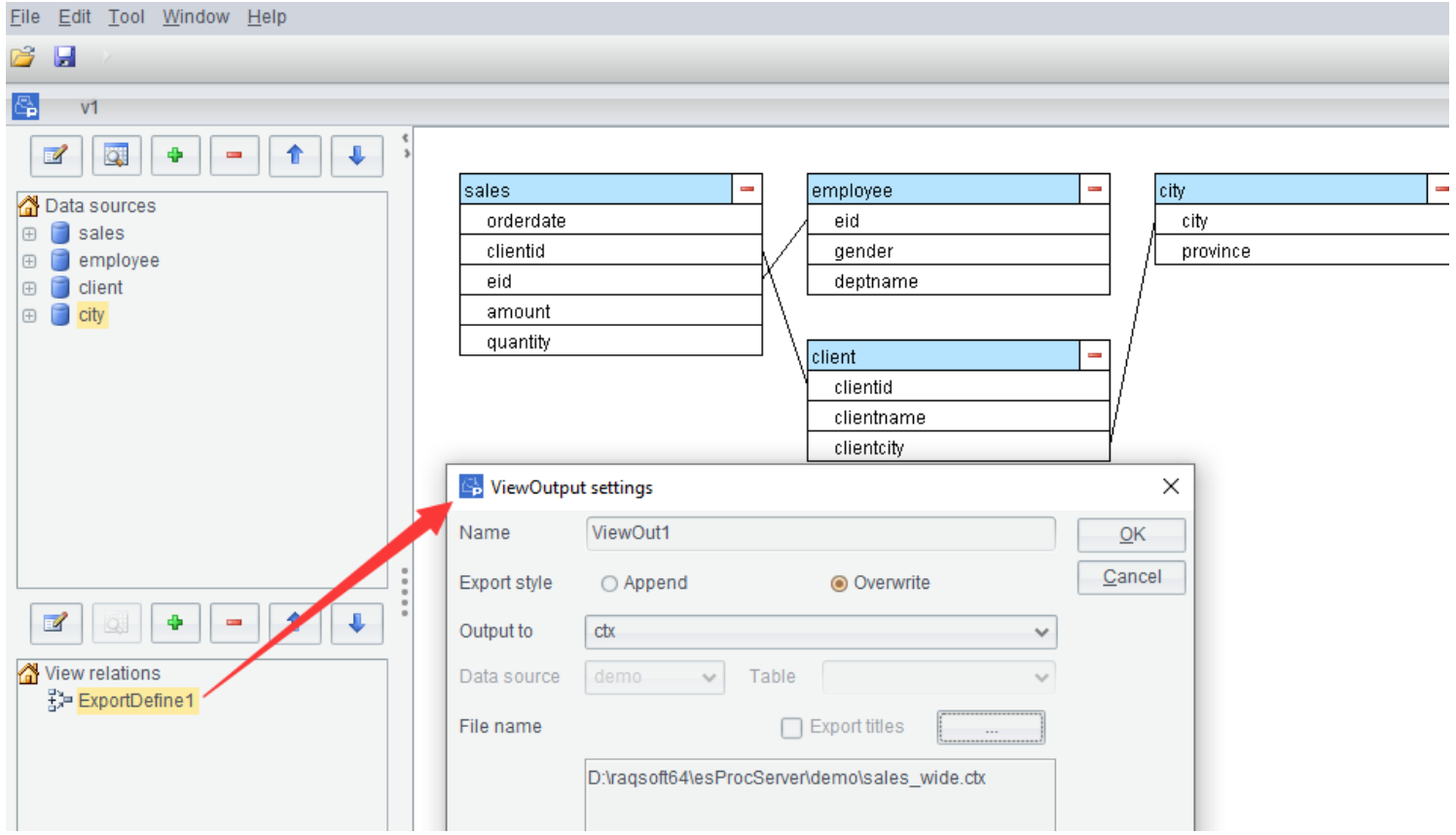
可视化生成宽表

3.设置源表关系

在“view relation”中添加表关系视图，将4个源表依次拖入视图，在弹出界面中设置表之间的关系。

4.设置输出存储格式

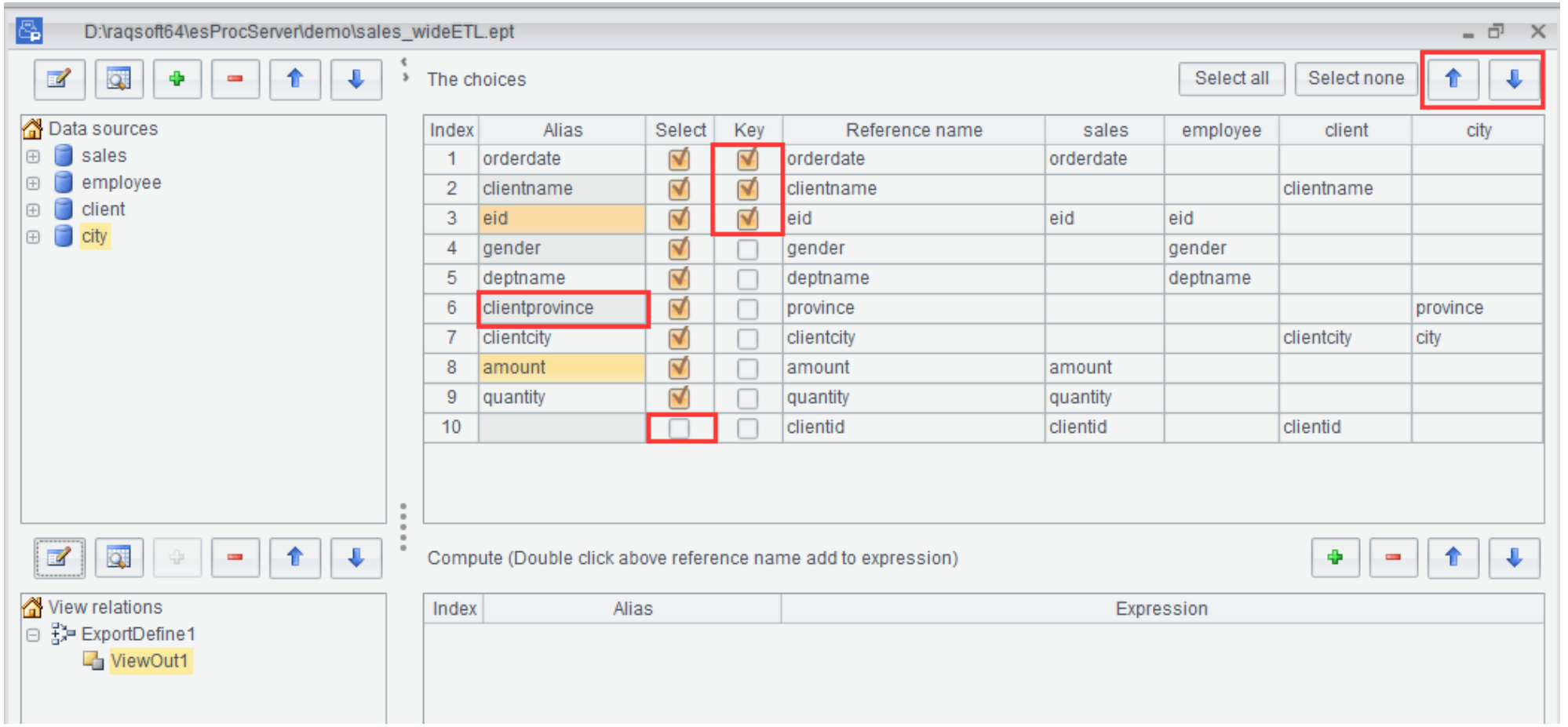
在关系视图的基础上，设置输出存储格式。多维分析一般使用组表ctx格式，数据量不大时覆盖即可。



可视化生成宽表

5.设置组表结构

根据关系视图设置组表结构，包括隐藏/输出字段、调整字段顺序、字段改名、选择合适的键字段、增加计算列等。



6.执行生成宽表

将ETL过程保存为sales_wideETL.ept，之后可执行该文件，最终生成宽表。

手工执行：在可视化IDE中选中输出存储格式，点击执行按钮。本方法适合一次性生成组表。

定时执行：使用操作系统的定时调度功能，以命令行的形式执行。本方法适合定时反复刷新组表。

windows命令行如下：

```
集算器安装目\bin\esprocx.exe sales_wideETL.ept
```

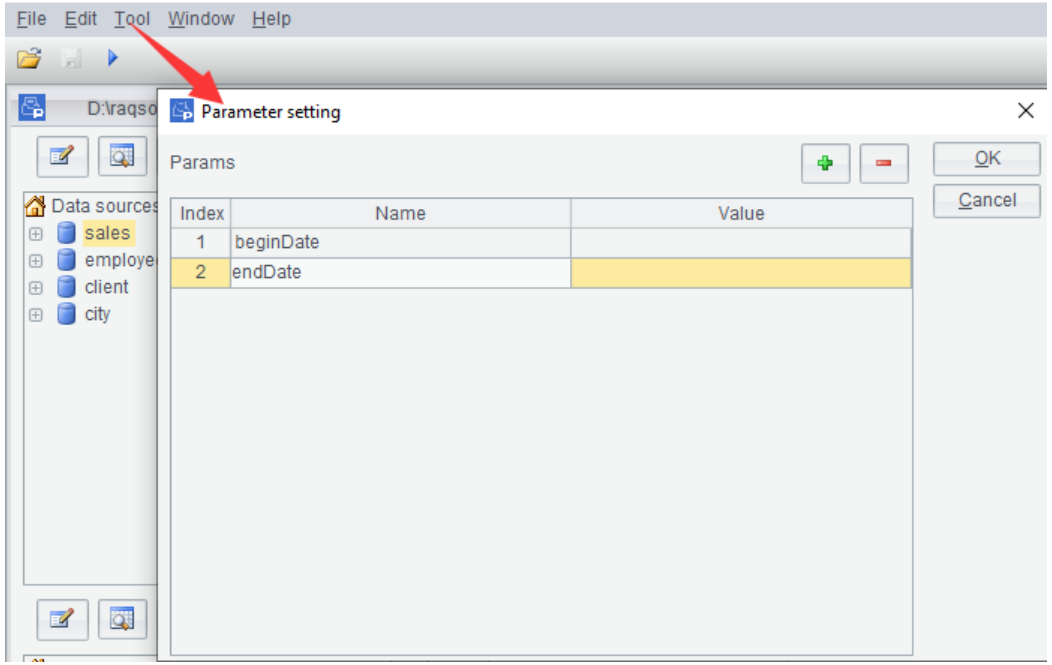
linux命令行如下：

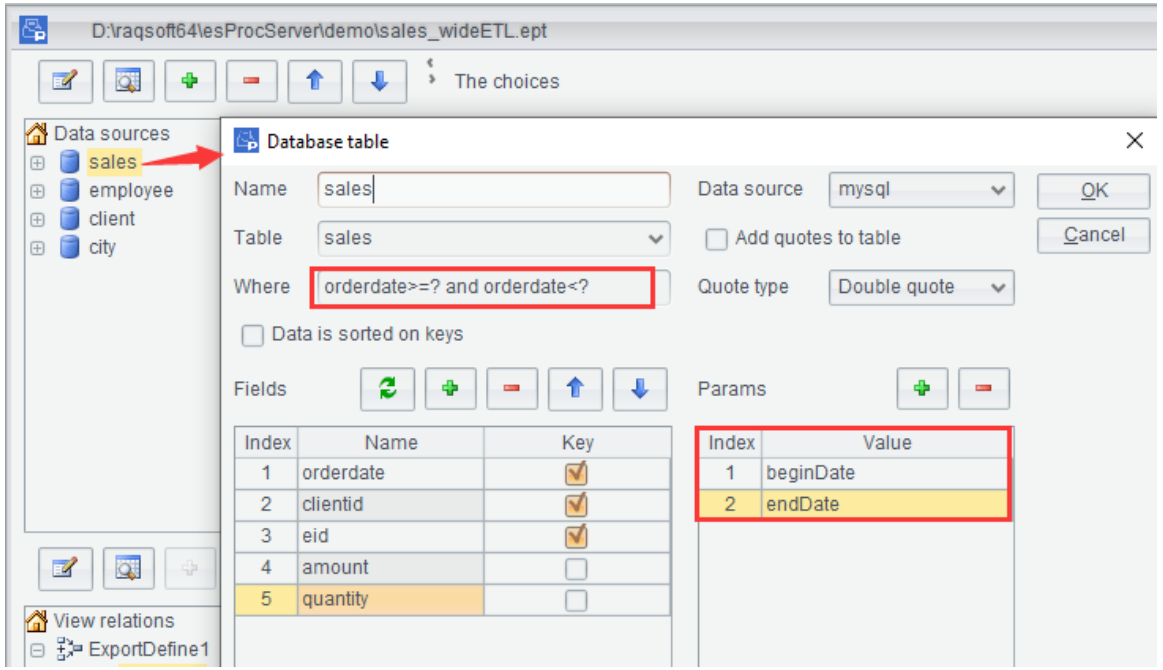
```
集算器安装目\bin\esprocx.sh sales_wideETL.ept
```

增量追加宽表

如果数据量较大，且总有新数据产生，则应当用追加的方式生成宽表

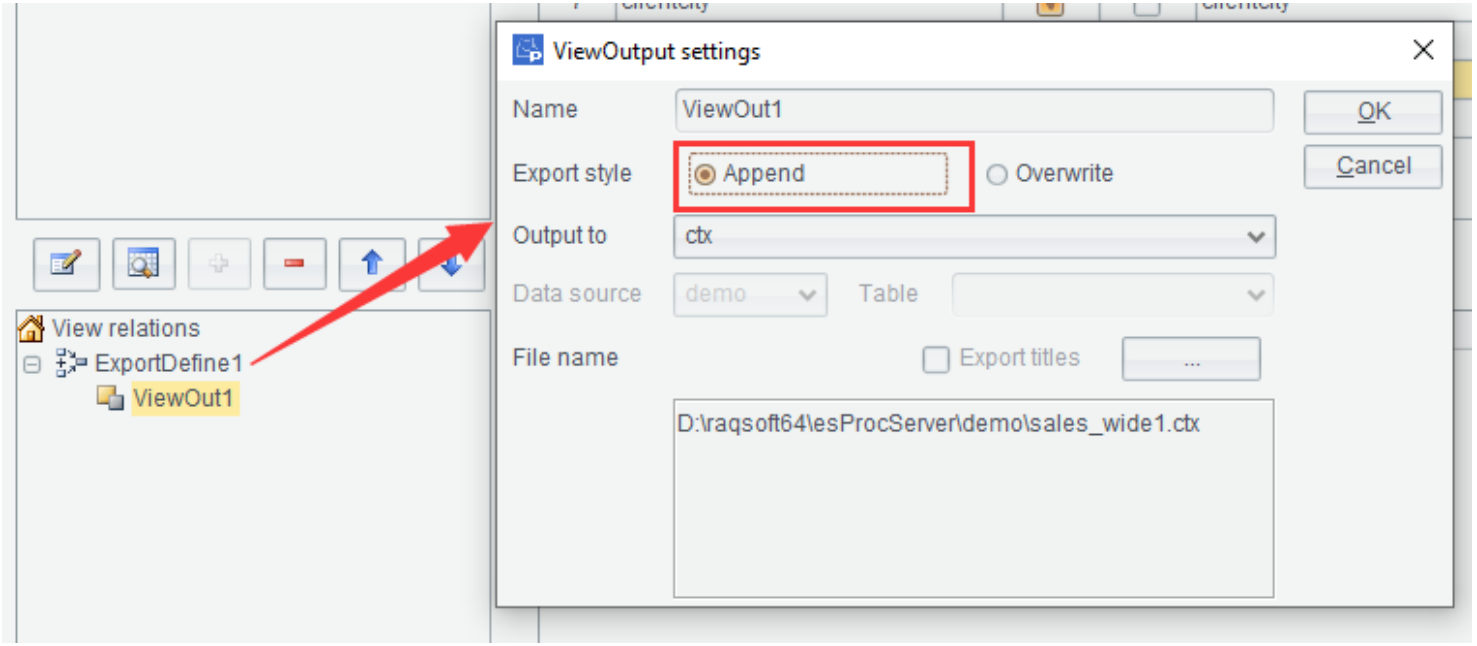
1.新增外部参数，并通过参数取源表数据





增量追加宽表

2.输出存储格式设定为追加



3.输入参数，在IDE或命令中执行ept

> 集算器安装目\bin\esprocx.exe sales_wideETL.ept 2014-01-01 2014-01-02

脚本ETL工具

脚本工具使用SPL语句，适用场景包括：源数据不规整，存在大量计算，ETL过程较复杂

- 分步骤计算
- 计算列关联
- 行列转换
- 并行计算
- 高性能采集
- 外部库、命令行

说明：SPL语言可参考 http://c.raqsoft.com.cn/article/1567908371148

单源ETL

目标：RDB中存储sales、employee、client、city，使用SPL脚本将这些表ETL为宽表。

	A	B
1	=connect@l("mysql")	/连接MySQL
2	=A1.cursor@x("select s.orderdate,c.clientname,e.eid,e.gender,e.deptname,c.province clientprovince,c.clientcity,s.amount,s.quantity from sales s join employee e on s.eid=e.eid join (select c.clientid,c.clientname,c.clientcity,ct.province from client c join city ct on c.clientcity=ct.city) c on s.clientid=c.clientid order by s.orderdate,c.clientname,e.eid ")	/执行SQL，注意数据需排序
3	=file("sales_wide.ctx").create@y(#orderdate,#clientname,#eid,gender,deptname,clientprovince,clientcity,amount,quantity)	/构造宽表
4	=A3.append(A2)	/向宽表追加数据

说明：组表的生成可参考http://doc.raqsoft.com.cn/esproc/tutorial/zbdsc.html

跨源ETL

源数据分2为两个库，Oracle存储sales表，MySQL存储emplyee、client、city表，下面用SPL脚本实现跨库ETL

注意，可视化IDE也具备简单的跨库ETL能力。

	A	B
1	=connect@l("orcl").cursor@x("select * from sales")	/从Oracle取sales
2	=connect@l("mysql")	/连接mysql数据源
3	=A2.query("select * from employee")	/取employee
4	=A2.query@x(" select * from client c join city ct on c.clientcity=ct.city")	/取client\city关联表
5	=A1.join@i(eid,A3:eid,gender,deptname;clientid,A4:clientid,clientname,clientcity,province:clientprovince)	/跨源关联
6	=A5.new(orderdate,clientname,eid,gender,deptname,clientprovince,cientcity,amount,quantity)	/调整字段前后顺序
7	=A6.sortx(orderdate,clientname,eid)	/按键排序
8	=file("sales_cross.ctx").create@y(#orderdate,#clientname,#eid,gender,deptname,clientprovince,clientcity,amount,quantity)	/构造宽表
9	=A8.append@i(A7)	/追加数据

非RDB源

如果sales表存储在kafka、webservice、mongodb等特殊数据源，就需要进行非RDB源的ETL。

使用集算器外部库可抽取非RDB源的数据，并实现异构跨源，这是脚本IDE特有的能力。

例如sales存储于mongodb，只需将取sales游标的代码修改如下（后续关联维表的代码不变）：

	A	B
1	=mongo_open("mongodb://192.168.1.7:27017/mydb")	/从Oracle取sales
2	=mongo_shell@x(A1,"sales.find()")	/连接mysql数据源

说明：集算器支持广泛的数据源，更多外部库用法，参考http://doc.raqsoft.com/esproc/func/wbk.html

复杂数据源

多维分析一般从数仓取数，数据比较规整，但偶尔也会遇到复杂数据源。

作为专业的结构化计算语言，SPL擅长处理复杂数据源，参考http://c.raqsoft.com.cn/article/1567908371148

比如下面代码，可讲多行文本转为二维表

	A	B
1	=file("sales_mult.txt").import@is()	/将小文本读入内存
2	=A1.step(3,1)	/隔3行取第1行
3	=A1.step(3,2)	/隔3行取第2行
4	=A1.step(3,3)	/隔3行取第3行
5	=join@p(A2;A3;A4)	/按序号位置关联
6	=A5.new((m=_1.array("\t"),m(1)):orderdate,m(2):clientid,m(3):eid,_2:amount,_3:quantity)	/组织成规范的二维表

目录 CONTENTS

03

宽表预汇总

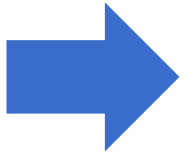
预汇总原理

预先对常用的维度层级或维度组合进行汇总，在实际进行多维分析时就可以直接利用这些汇总数据，从而达到提高性能的目的。

预汇总是冗余的粗粒度cube，形式上是个文件。

预汇总依赖于原宽表，且只对宽表有效。

sales_wide(事实表)	
key	orderdate(维度)
key	clientname（维度）
key	eid（维度）
	gender（维度）
	deptname（维度）
	clientprovince（维度）
	clientcity（维度）
	amount（测度）
	quantity（测度）



clientname(预汇总)
clientname（分组维度）
amount（汇总测度）

clientprovince_deptname(预汇总)
clientprovince（分组维度）
deptname（分组维度）
amount（汇总测度）
quantity（汇总测度）

year_month(预汇总)
year(分组维度)
month（分组维度）
amount（汇总测度）
quantity（汇总测度）

基本用法

生成预汇总

使用SPL脚本对宽表sales_wide.ctx生成预汇总clientname，对clientname分组对amount求和。代码如下：

	A	B
1	=file("sales_wide.ctx").create()	/打开宽表
2	=A1.cuboid()	/删除所有预汇总，非必需步骤
3	=A1.cuboid(clientname,clientname;sum(amount))	/生成预汇总

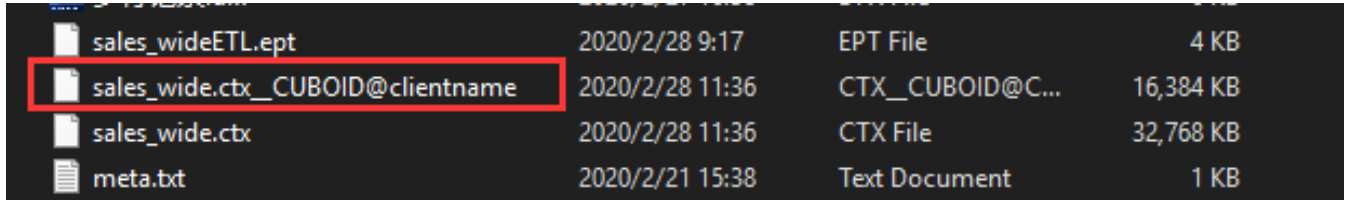
说明：

1.预汇总需指定分组字段和汇总算法，支持的汇总算法有sum/count/max/min/top/iterate，详见：

http://doc.raqsoft.com.cn/esproc/func/cuboid.html

2.如果修改或追加宽表，预汇总会自动更新。

3.预汇总文件会在宽表所在目录生成，如下：



基本用法

使用预汇总：在BI工具中执行SQL访问多维分析后台时，如果参与计算的字段全都在预汇总中，则直接使用预汇总，否则使用原宽表进行计算，判断过程由集算器自动完成。

字段全都在预汇总中，将使用预汇总进行计算：

```
select clientname,sum(amount) from sales_wide group by clientname
```

quantity不在预汇总中，将使用宽表进行计算：

```
select clientname,sum(amount),sum(quantity) from sales_wide group by clientname
```

字段全都在预汇总中，将使用预汇总进行计算：

```
select clientname,sum(amount) from sales_wide where clientname in ('一诠精密工业','万海') group by clientname
```

orderdate不在预汇总中，将使用宽表进行计算：

```
select clientname,sum(amount) from sales_wide where orderdate>=date('2015-01-01') and clientname in ('一诠精密工业','万海') group by clientname
```

多层多维组合预汇总

预汇总可由多个层次多个维度组合而成

用SPL脚本建立多层多维组合预汇总clientprovince_clientcity_deptname

```
=A1.cuboid(clientprovince_clientcity_deptname,clientprovince,clientcity,deptname;sum(amount),sum(quantity))
```

下面的SQL将使用预汇总进行计算

```
select deptname,clientprovince,clientcity, sum(amount),sum(quantity) from sales_wide group by clientname,clientprovince,clientcity
```

存在多维多层预汇总时，如果SQL只对其中部分维度进行计算，则对预汇总自动进行二次汇总，而不是从原宽表汇总。如此可提高计算性能。

针对预汇总clientprovince_clientcity_deptname，下面的SQL将对预汇总进行二次汇总。

```
select clientprovince, sum(amount),sum(quantity) from sales_wide group by clientname
```

```
select deptname,clientprovince, sum(amount),sum(quantity) from sales_wide group by deptname,clientname
```

多个预汇总

存在多个不相干的预汇总时，集算器将根据SQL的需要，自动使用匹配的预汇总。

多维后台存在两个预汇总：clientname、clientprovince_clientcity_deptname，则下面的两个SQL将自动使用不同的预汇总进行计算

```
select clientname,sum(amount) from sales_wide group by clientname
```

```
select deptname,clientprovince,clientcity, sum(amount),sum(quantity) from sales_wide group by clientname,clientprovince,clientcity
```

SQL可利用多个预汇总进行二次汇总时，集算器将自动使用数据量最少的预汇总，最大限度提高性能。

多维后台存在两个预汇总： clientprovince_clientcity_deptname、clientprovince_clientcity

下面的SQL将使用数据量较少的clientprovince_clientcity进行二次汇总。

```
select clientprovince,sum(amount) from sales_wide group by clientprovince
```

时间维度预汇总

与普通维度不同，日期维度具有固定的汇总关系，可使用year、month函数建立预汇总。

说明：普通维度不支持按函数建立预汇总。

使用SPL脚本对宽表sales_wide.ctx生成预汇总yearmonth，对orderdate字段按年、月分组，对amount和quantity求和。代码如下：

```
=A1.cuboid(yearmonth,year(orderdate),month(orderdate);sum(amount),sum(quantity))
```

时间维度预汇总具备普通维度预汇总的全部功能。比如下面的SQL可利用预汇总进行计算

```
select year(orderdate) y,sum(amount) m,sum(quantity) from sales_wide group by  year(orderdate)
```

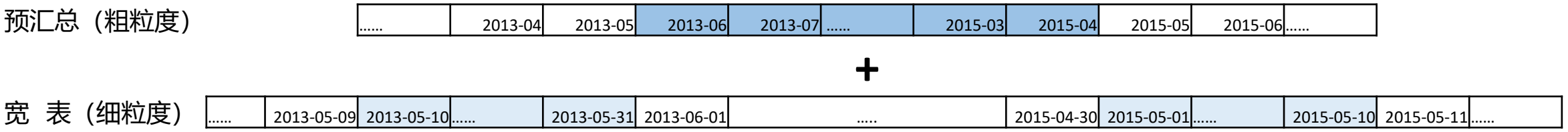
```
select year(orderdate) y,month(orderdate) m,sum(amount),sum(quantity) from sales_wide
where year(orderdate)  in(2013,2015) group by  year(orderdate) ,month(orderdate)
```

时间维度预汇总

对于时间维度预汇总，当SQL按日期段查询时，集算器将分别从预汇总和宽表查询数据，从而最大限度提高性能。

比如SQL语句为：
select orderdate, sum(amount), sum(quantity) from sales_wide
where orderdate>=date('2013-05-10') and orderdate<=date('2015-05-10')

则集算器会将区间分为:
[2013-06,2015-04]预汇总查询
[2013-05-10,2013-05-31]、 [2015-05-01,2013-05-10]宽表查询



说明：
1.时间维度预汇总支持跨年跨月
2.如果同时还存在对年的预汇总，则集算器分别从年预汇总、年-月预汇总、宽表查询。
3. 时间预汇总只支持两种汇总方式：年单层、年-月两层。
4.除了日期字段外，日期时间字段也支持时间维度预汇总

组合预汇总

预汇总可以极大提高计算性能，在分组字段较少且常用字段不确定时，可对每种字段的组合建立预汇总，即全组合预汇总。如果分组字段较多（超过11个），全组合预汇总容易占满空间，此时须按常用字段生成预汇总，或选取部分字段生成组合预汇总。

宽表sales_wide有9个可分组字段，针对其中6个常用字段生成组合预汇总，共2^6-2个：

	A	B	C	D	E
1	=file("sales_wide.ctx").create()				/打开宽表
2	=["year(orderdate)","month(orderdate)","orderdate","clientname","clientcity","clientprovince"]				/所有可分组字段
3	for bits@n("1"),bits@n("111110")				/从000001循环到111110
4		=[]			/准备空的分组字段集合
5		for 1,A2.len()			/循环A6的每一位
6			if and(shift(A3,B5-1),1)==1	=B4=B4|A2(B5)	/如果当前位为1，则将对应的字段加入分组字段集合
7		=A1.cuboid(${A3},${B4.string()};sum(amount),sum(quantity))			/建立预汇总

说明：000001到111110之间的每个数对应一种字段组合情况，当前位为1表示组合中有该字段，为0表示没有。

目录 CONTENTS

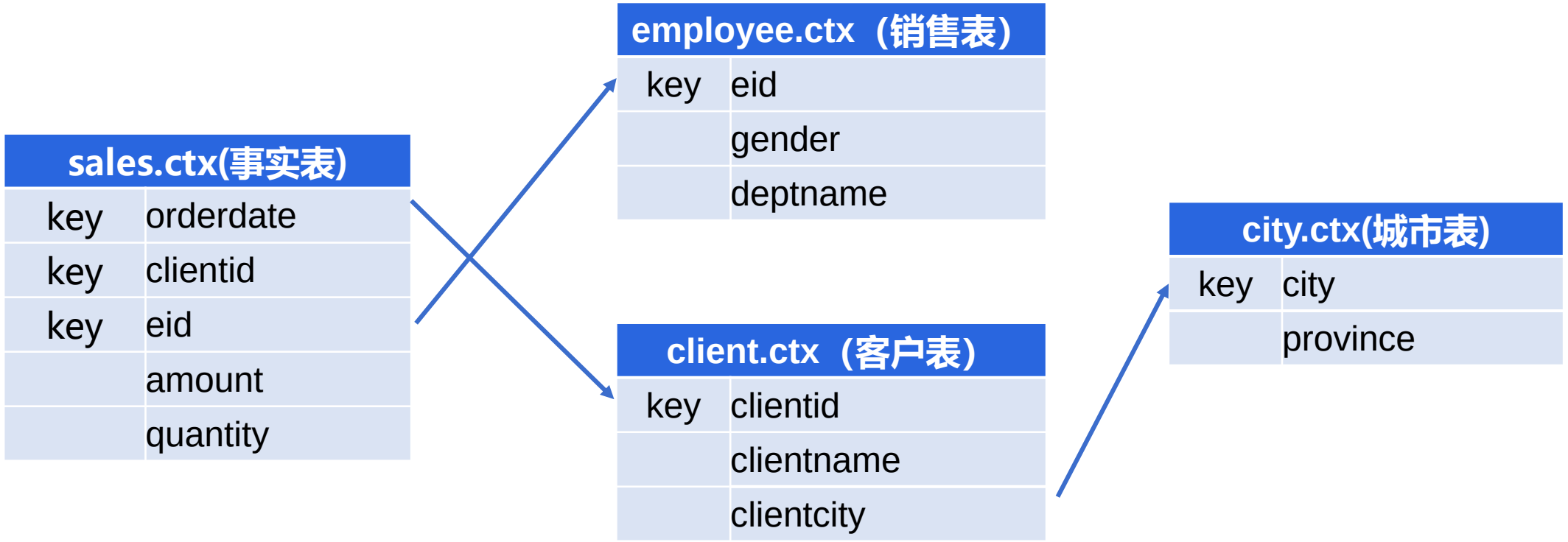
04

关联表方案

关联表

事实表和维表可以分表存放，如此可消除冗余节省空间，缺点是关联计算性能较低。

集算器提供了多种优化手段，用来提升关联计算性能。



原始关联计算

不加优化，只进行原始的关联计算。

典型的关联SQL如下：

```
select year(s.orderdate) ,month(s.orderdate),e.deptname ,ct.province,sum(s.amount) ,sum(s.quantity)
from sales.ctx s  join employee.ctx e on s.eid=e.eid
join client.ctx c on s.clientid=c.clientid
join city.ctx ct on c.clientcity=ct.city
where s.orderdate>=date('2013-05-10') and s.orderdate<date('2015-05-10')
group by  year(s.orderdate) ,month(s.orderdate),e.deptname,ct.province
```

说明：

1.集算器SQL只支持join语句显式表达关联关系，不支持在where中用条件过滤

2.关联表的生成、部署、SQL和宽表相同。

3.可通过meta.txt设置别名，从而在SQL中使用无扩展名的表名。

维表内存化

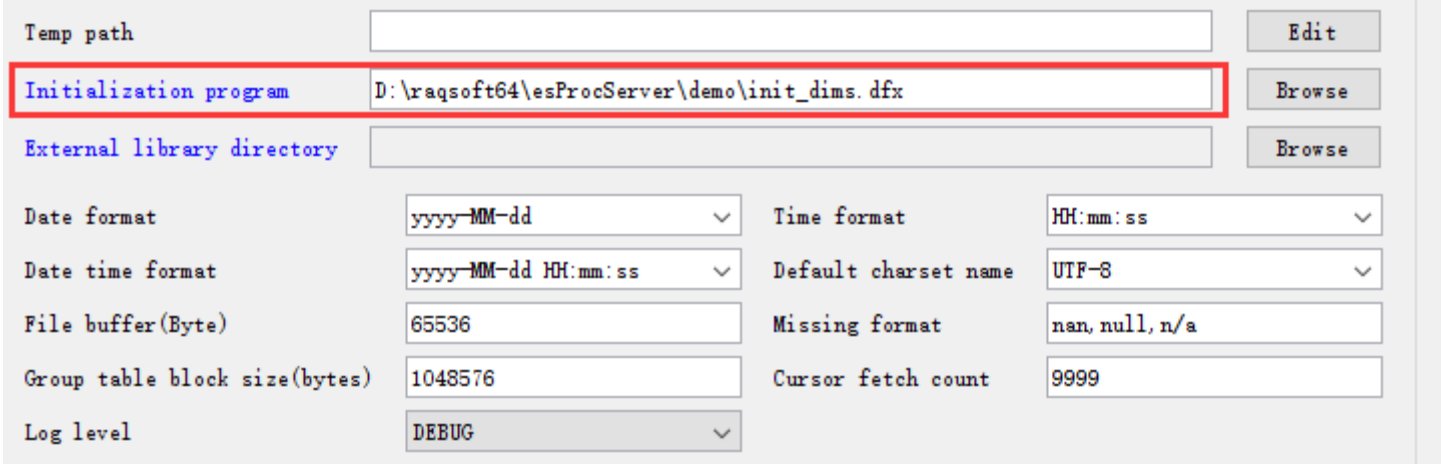
将维表加载进内存并常驻，关联计算时可显著提升性能。

1.编写内存加载脚本

	A	B	C
1	=file("employee.ctx").create().memory()	=env(employee,A1)	/将外存维表employee.ctx加载到内存，并以名字employee共享
2	=file("client.ctx").create().memory()	=env(client,A2)	
3	=file("city.ctx").create().memory()	=env(city,A3)	

2.配置内存加载脚本

修改raqsoftConfig.xml，或通过可视化界面配置内存加载脚本



维表内存化

3.启动集算器JDBC\ODBC服务，在客户端使用内存共享名正常写SQL

对比：关联查询

```
select year(s.orderdate) ,month(s.orderdate),e.deptname ,ct.province,sum(s.amount) ,sum(s.quantity)
from sales.ctx s  join employee e on s.eid=e.eid
join client c on s.clientid=c.clientid
join city ct on c.clientcity=ct.city
where s.orderdate>=date('2013-05-10') and s.orderdate<date('2015-05-10')
group by  year(s.orderdate) ,month(s.orderdate),e.deptname,ct.province
```

说明：用内存共享名访问维表时，meta.txt中没必要再配别名

外键序号化

将关联字段改造成1开始的整数序号，可大幅提升关联性能

sales.ctx及其维表共有3个关联关系

sales.ctx和employee.ctx：原本就以序号为关联字段，无需改造

client.ctx和city.ctx：以字符串为关联字段，但这两张表数据量小关联较快，无需改造

sales.ctx和client.ctx：以字符串为关联字段，且sales.ctx是大事实表，关联字段应改造成序号。部分数据如下：

sales.ctx

orderdate	clientid	eid	amount	quantity
2013-07-04	b0001	5	2440	1
2013-07-05	c0004	6	1863.41	1
2013-07-08	c0020	4	1813	1
2013-07-08	a0023	3	670.8	1
2013-07-09	a0011	4	3730	1

client.ctx

clientid	clientname	clientcity
a0001	三川实业有限公司	天津
a0002	东南实业	天津
a0003	坦森行贸易	石家庄
a0004	国顶有限公司	深圳
b0001	通恒机械	南京

外键序号化

1.执行外键序号化脚本 。

	A	B
1	=file("client.ctx").create().memory()	/将内表形式打开客户维表
2	=A1.derive(#:newid)	/新加字段newid，值为序号
3	=file("sales.ctx").create().cursor()	/以游标形式打开事实表
4	=A3.switch(clientid,A2:clientid)	/以原字段关联事实表和维表
5	=A4.new(orderdate,clientid.newid:clientid,eid,amount,quantity)	/将事实表中的外键改为序号
6	=file("sales_seq.ctx").create@y(#orderdate,#clientid,#eid,amount,quantity)	/构建新事实表，字段名不变
7	=A6.append@i(A5.sortx(orderdate,clientid,eid))	/向新事实表写入有序数据
8	=A2.new(newid:clientid,clientname,clientcity)	/将维表中的主键改为序号
9	=file("client_seq.ctx").create(#clientid,clientname,clientcity)	/构建新维表，字段名不变
10	=A9.append@i(A8.cursor())	/向维表写入数据

改造之后，新的事实表和客户表如下：

sales_seq.ctx

orderdate	clientid	eid	amount	quantity
2013-07-04	5	5	2440	1
2013-07-05	79	6	1863.41	1
2013-07-08	82	4	1813	1
2013-07-08	32	3	670.8	1
2013-07-09	28	4	3730	1

client_seq.ctx

clientid	clientname	clientcity
1	三川实业有限公司	天津
2	东南实业	天津
3	坦森行贸易	石家庄
4	国顶有限公司	深圳
5	通恒机械	南京

外键序号化

2.编写维表的内存加载脚本

	A	B	C
1	=file("employee.ctx").create().memory()	=env(employee,A1)	
2	=file("client_seq.ctx").create().memory()	=env(client,A2)	/读入新维表，并在内存共享
3	=file("city.ctx").create().memory()	=env(city,A3)	

说明：维表内存化不是外键序号化的必须步骤，但两者配合性能更佳。

如果使用维表内存化，则外键序号化脚本中不必生成新维表（A9A10），只需在内存加载脚本中将原维表主键临时修改为序号，A2代码如下：

```
=file("client.ctx").create().memory().run(#:clientid)
```

3.后续操作不变：配置内存加载脚本，启动集算服务，正常写SQL。

说明：有关外键序号化的进一步技巧，详见：http://c.raqsoft.com.cn/article/1575263621672

外键排号键化

将关联字段改造为序号时需要先关联再替换，过程较复杂。为了简化改造过程，可略微损失性能，将关联字段改造为排号键。

1.事实表排号键化脚本 。

sales.ctx的clientid字段格式为1位字母+4位数字的形式，可改造成3位排号键（每位最大255）

	A	B
1	=file("sales.ctx").create().cursor()	/打开事实表
2	=A1.run(clientid=k(asc(mid(clientid,1,1)),int(mid(clientid,2,2)),int(mid(clientid,4,2))))	/排号建化
3	=file("sales_k.ctx").create@y(#orderdate,#clientid,#eid,amount,quantity)	/构建新事实表
4	=A3.append@i(A2.sortx(orderdate,clientid,eid))	/写入有序数据

2.编写维表的内存加载脚本（后续操作不变）

	A	B	C
1	=file("employee.ctx").create().memory()	=env(employee,A1)	
2	=file("client.ctx").create().memory().run(clientid=k(asc(mid(clientid,1,1)),int(mid(clientid,2,2)),int(mid(clientid,4,2))))	=env(client,A2)	/读入client维表，临时改造排号键
3	=file("city.ctx").create().memory()	=env(city,A3)	

说明：排号键还有更广泛的用法，比如组织机构代码、身份证，详见：性能优化技巧 - 多层排号键http://c.raqsoft.com.cn/article/1550218072302

预关联

服务端将事实表和维表读入内存并预先关联起来，客户端再执行关联SQL时，服务端将直接使用关联后的数据，从而大幅提升计算性能。

优点：在保持关联表数据结构的前提下，达到与宽表类似的高计算性能

缺点：事实表不能太大

下面以sales及其相关维表为例，说明预关联的用法

1.服务端启动时执行内存加载脚本

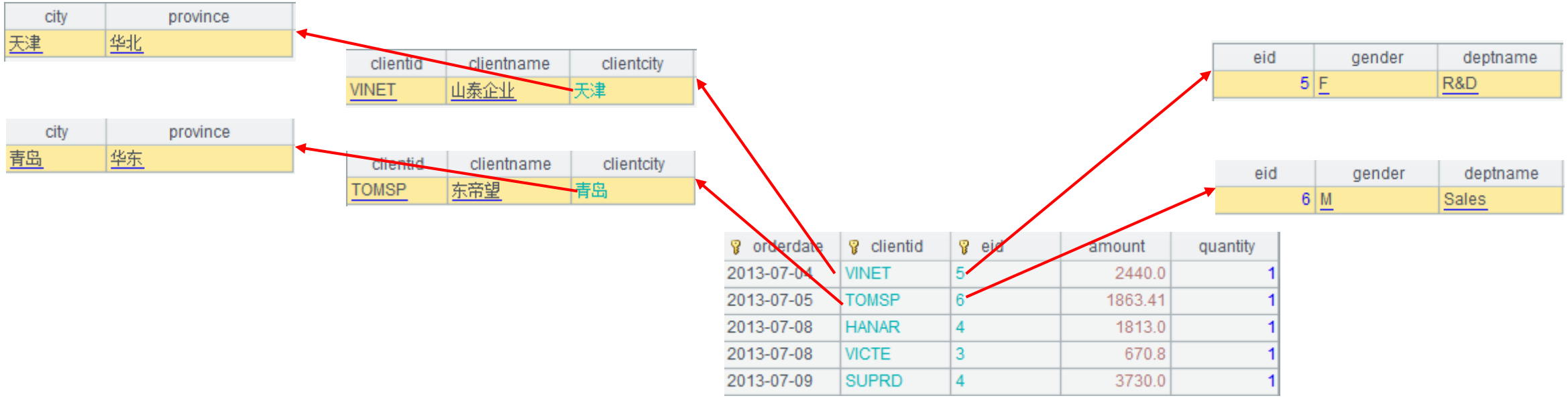
	A	B	C
1	=file("sales.ctx").create().memory()	=env(sales,A1)	/内存化事实表，并共享
2	=file("employee.ctx").create().memory()	=env(employee,A2)	/销售维
3	=file("client.ctx").create().memory()	=env(client,A3)	/客户维
4	=file("city.ctx").create().memory()	=env(city,A4)	/地区维
5	>client.switch(clientcity,city:city)		/客户关联地区
6	>sales.switch(eid,employee:eid; clientid,client:clientid)		/事实关联销售、客户

说明：关于SPL函数switch，详情参阅http://doc.raqsoft.com.cn/esproc/func/aswitchfiaix.html

关于SPL关联优化技巧，详情参阅http://c.raqsoft.com.cn/article/1550816037603

预关联

执行内存加载脚本后，全局变量sales的数据结构如下，可以看到事实表已通过指针与各级维表关联起来。



2.客户端正常执行SQL语句

```
select year(s.orderdate) ,month(s.orderdate),e.deptname ,ct.province,sum(s.amount) ,sum(s.quantity)
from sales join employee e on s.eid=e.eid
join client c on s.clientid=c.clientid
join city ct on c.clientcity=ct.city
where s.orderdate>=date('2013-05-10') and s.orderdate<date('2015-05-10')
group by  year(s.orderdate) ,month(s.orderdate),e.deptname,ct.province
```

说明：参与预关联计算的表使用内存共享名，无须meta.txt设定别名

3.更新内存数据

与维表不同，事实表通常会定期生成或追加，这种情况下需要将硬盘上的事实表更新到内存中

方法一：冷更新

通过重启集算服务，让内存加载脚本重新执行，从而达到更新内存的目的。

方法二：热更新

在集算器客户端，使用命令行远程执行集算服务器上的加载脚本。具体命令如下：

Windows： esprocx.exe –r =callx(\"d:/raqsoft64/temp/init/initData.dfx\";[\"127.0.0.1:8281\"])

Linux： esprocx.sh –r =callx(\"/opt/raqsoft64/temp/init/initData.dfx\";[\"127.0.0.1:8281\"])

说明：

1.自动热切换即定时执行加载脚本，可使用操作系统自带调度工具，如Windows计划任务、Linux Crontab命令，或第三方可视化工具如opencron。

1.执行热更新的集算器环境，应当与集算服务不同（可在另一个目录或另一台机器上），否则执行命令前会额外执行一遍内存加载脚本，在新进程中生成无用内存数据。

2.热更新时，服务器在短期内会占用2倍内存。冷更新则无此问题。

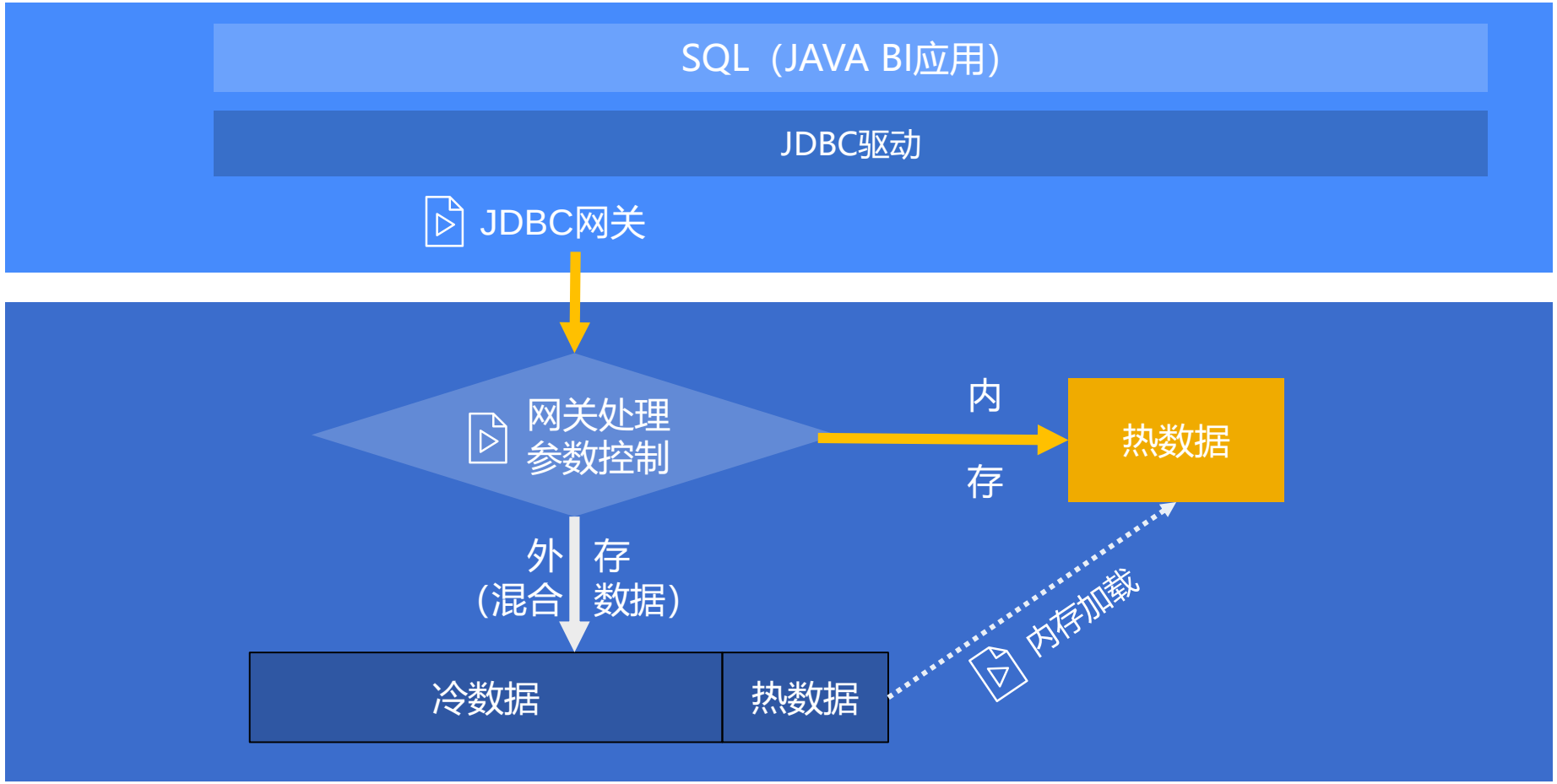
目录 CONTENTS

05

冷热路由

冷热路由

当体较大的外存事实表含有部分常用热数据时，可以把热数据常驻内存，此时通过JDBC网关判断SQL的取值区间，就可以路由到内存热数据或外存冷热混合数据。



部署步骤

sales.ctx包含部分热数据（orderdate>=2015-01-01）,以此为例讲解冷热路由的实现方法和部署步骤。

1.部署客户端JDBC网关脚本

JDBC网关是集算器的专有机制，其特点包括：

- 以SPL脚本的形式存在，截获并处理所有流经JDBC驱动的SQL语句
- 有固定参数sql和args，分别代表SQL语句和参数集合，由JDBC驱动自动传入
- 须部署在JDBC驱动所在的计算机，不适用ODBC

针对本案例，JDBC网关脚本router.dfx须调用服务器上的网关处理脚本routerprocess.dfx，并向客户端返回计算结果：

	A	B
1	=callx("routerprocess.dfx",[sql],[args];["192.168.0.2:8281"])	/调用服务器网关处理脚本
2	return A1.conj()	/返回服务器计算结果

在JDBC驱动的raqsoftConfig.xml中配置router.dfx，使其生效：

```
<JDBC>
<load>Runtime</load>
<gateway>router.dfx</gateway>
</JDBC>
```

说明：JDBC网关用法详见http://doc.raqsoft.com.cn/esproc/tutorial/jdbcwangguan.html

部署步骤

2.部署服务端网关处理脚本

网关处理脚本routerprocess.dfx部署在服务端，其逻辑算法为：判断SQL所查询的数据范围，如果只查热数据，则直接执行SQL；如果包含部分冷数据，则将SQL中的事实表由内存表替换为外存表，最后执行SQL。

	A	B	C	D
1	=connect()			/连接服务端本地数据源
2	=from=sql.sqlparse@f()			/原Sql from部分
3	if substr(from,"sales")!=null			/如果from中包含sales
4		=hotcoldLine=date("2015-01-01")		/冷热分界线
5		if args(1)>=hotcoldLine	>B1=A1.query(sql,args(1),args(2))	/如果查询区间只有热数据，则直接执行SQL
6		else	=sqlhot=replace(sql,"sales","sales.ctx")	/否则将SQL里的表名替换为全量组表
7			>B1=A1.query(sqlhot,args(1),args(2))	/用全量组表执行SQL
8	else			/如果from部分没有要处理的表
9		>B1=A1.query(sql,${args.len().(concat("args(",~,")"))})		/按原参数执行原SQL
10	return B1			/返回计算结果

说明：

1.脚本须设定两个参数：sql和args，用来接收router.dfx传入的参数值

2.内存表名为sales，外存表名为sales.ctx，原SQL默认用内存表名

3.部署服务端内存加载脚本

从全量组表中查询当期热数据，内存只加载这部分数据

预关联不是冷热路由必须的步骤，但两者配合性能更佳

	A	B	C
1	=file("sales.ctx").create().memory(;orderdate>=date("2015-01-01"))	=env(sales,A1)	/查询当期热数据，并共享内表
2	=file("employee.ctx").create().memory()	=env(employee,A2)	/销售维
3	=file("client.ctx").create().memory()	=env(client,A3)	/客户维
4	=file("city.ctx").create().memory()	=env(city,A4)	/地区维
5	>client.switch(clientcity,city:city)		/客户关联地区
6	>sales.switch(eid,employee:eid; clientid,client:clientid)		/事实关联销售、客户

4.客户端通过JDBC正常执行SQL

对于下面的SQL，当参数区间完全处于热数据区时，自动路由到内存表计算；当部分或全部区间落入冷数据区时，自动路由到外存表计算。

```
select year(s.orderdate) ,month(s.orderdate),e.deptname ,ct.province,sum(s.amount) ,sum(s.quantity)
from sales join employee e on s.eid=e.eid
join client c on s.clientid=c.clientid
join city ct on c.clientcity=ct.city
where s.orderdate>=? and s.orderdate<?
group by  year(s.orderdate) ,month(s.orderdate),e.deptname,ct.province
```

目录 CONTENTS

06

应用接口

应用接口

集算器提供了符合JAVA SE6规范的JDBC驱动，以及符合Microsoft标准的ODBC驱动

- 连接打开/关闭
- 取数据库名/默认数据库名
- 表名列表
- 字段列表/字段类型
- SQL语句/存储过程
- 结果集

……

如果定制开发基于集算器的前端BI工具，可参考

JDBC接口规范https://docs.oracle.com/javase/6/docs/technotes/guides/jdbc/

ODBC接口规范https://docs.microsoft.com/en-us/sql/odbc/reference/syntax/odbc-api-reference

应用接口

得益于规范的JDC/ODBC接口，集算器可被大部分BI工具集成

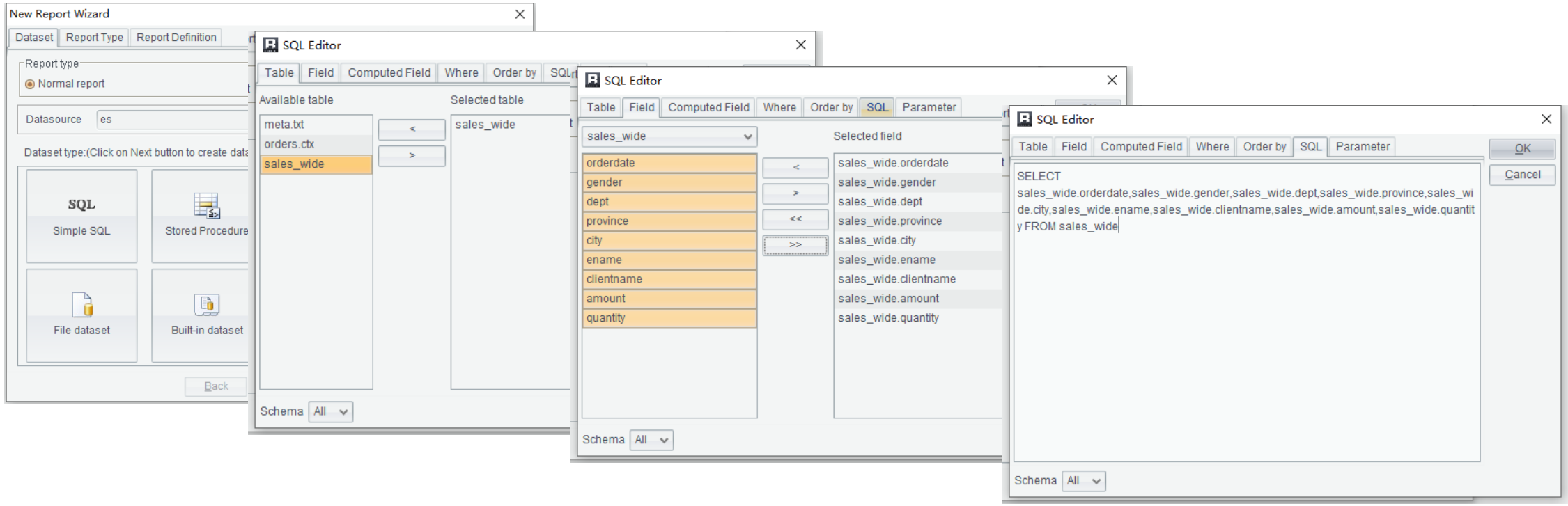
- BIRT report
- JASPER report
- SMART BI
- FINE BI
- RUNQIAN report
- MICROSOFT excel
- SAP BO
- ……

说明：少数特殊的BI工具比如tableau，需要驱动程序申报注册，暂不支持集算器。

JDBC接口

以润乾报表为例，在JAVA BI工具中获取集算服务元数据

集算器JDBC驱动支持元数据接口，可通过主流BI工具的向导生成SQL语句，如下图：

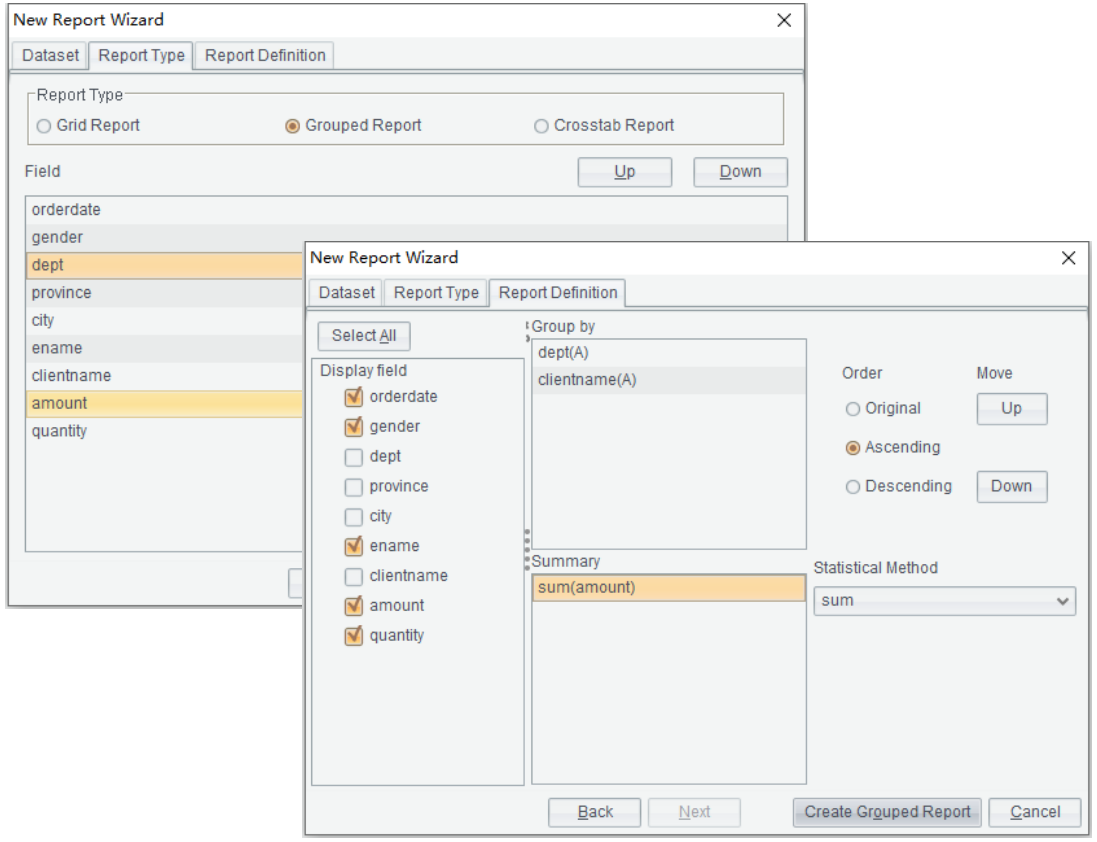


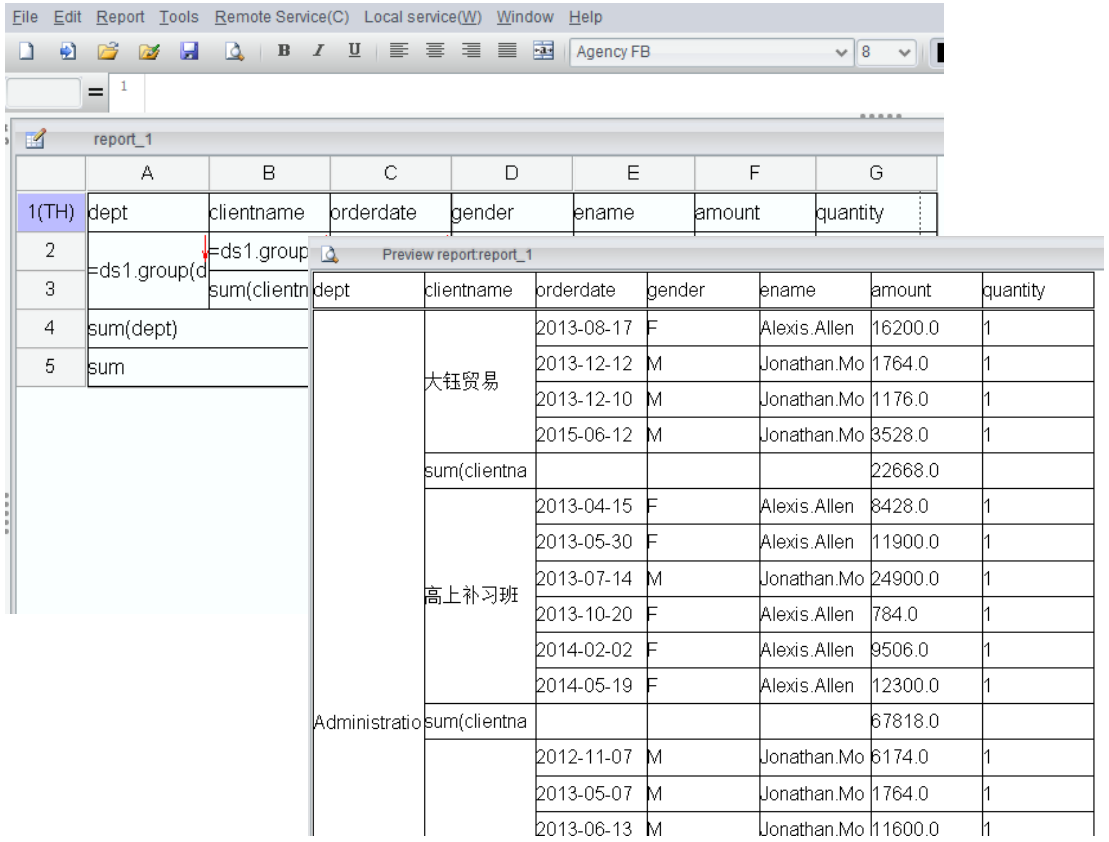
说明：如果没有别名文件meta.txt，有些BI向导会生成形如sales_wide.ctx.orderdate的字段名，需要手工添加引号，变成"sales_wide.ctx".orderdate才能正确执行。有些BI工具会自动添加引号，比如润乾报表。

JDBC接口

数据的呈现分析

生成标准SQL后，可在OLAP分析或报表中使用集算器数据。如下图：



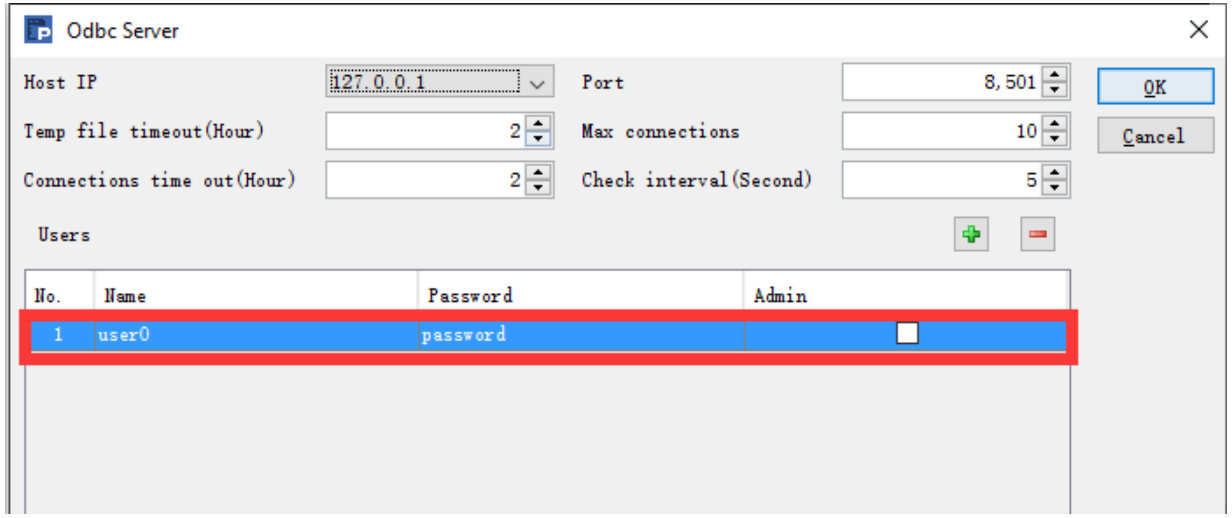


ODBC接口

下面以Excel为例，说明非JAVA BI工具使用ODBC驱动的过程。

1.配置集算器ODBC服务

通用选项信息与JDBC部署时相同，重点是主目录。ODBC服务信息的重点是端口号（与JDBC节点服务的端口不同）。还需额外配置用访问账号（类似下图）：



ODBC接口

2.部署数据和别名

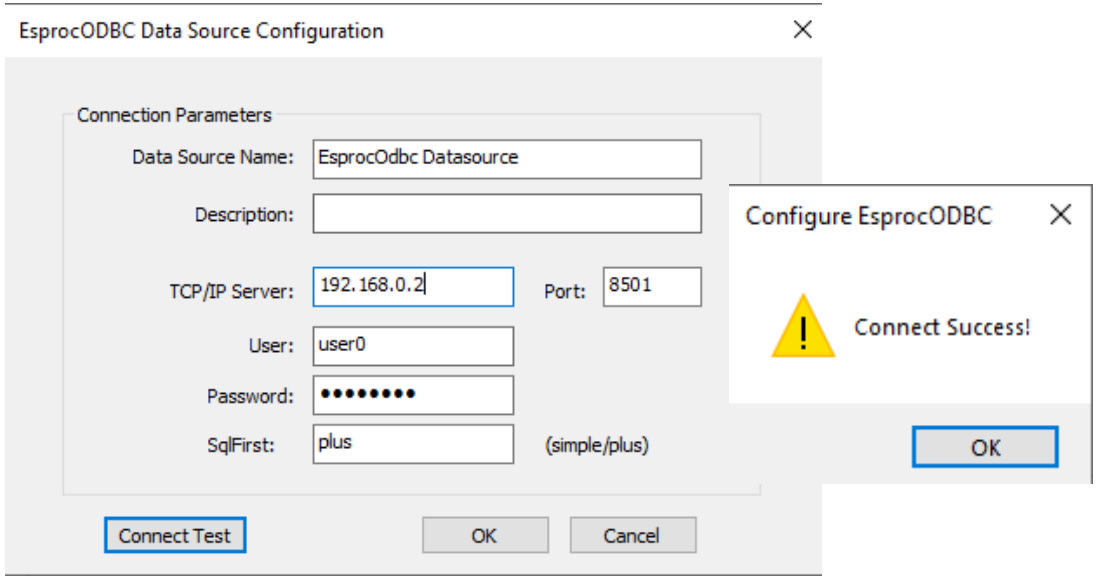
3.启动集算器ODBC服务

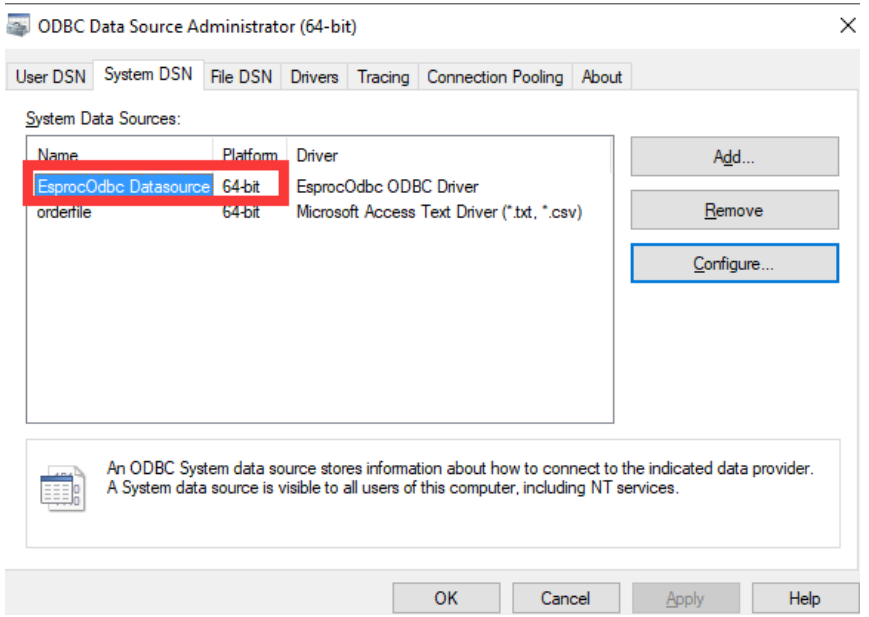
4.安装ODBC驱动

在BI工具（Excel）所在的计算机上，以管理员身份执行“集算器安装路径\bin\esprocOdbcinst.exe “，注意操作系统位数、集算器位数、Excel位数须一致。

5.配置Windows ODBC数据源

与普通ROLAP类似，注意服务器地址、端口号、访问账号，最后别忘测试连接。

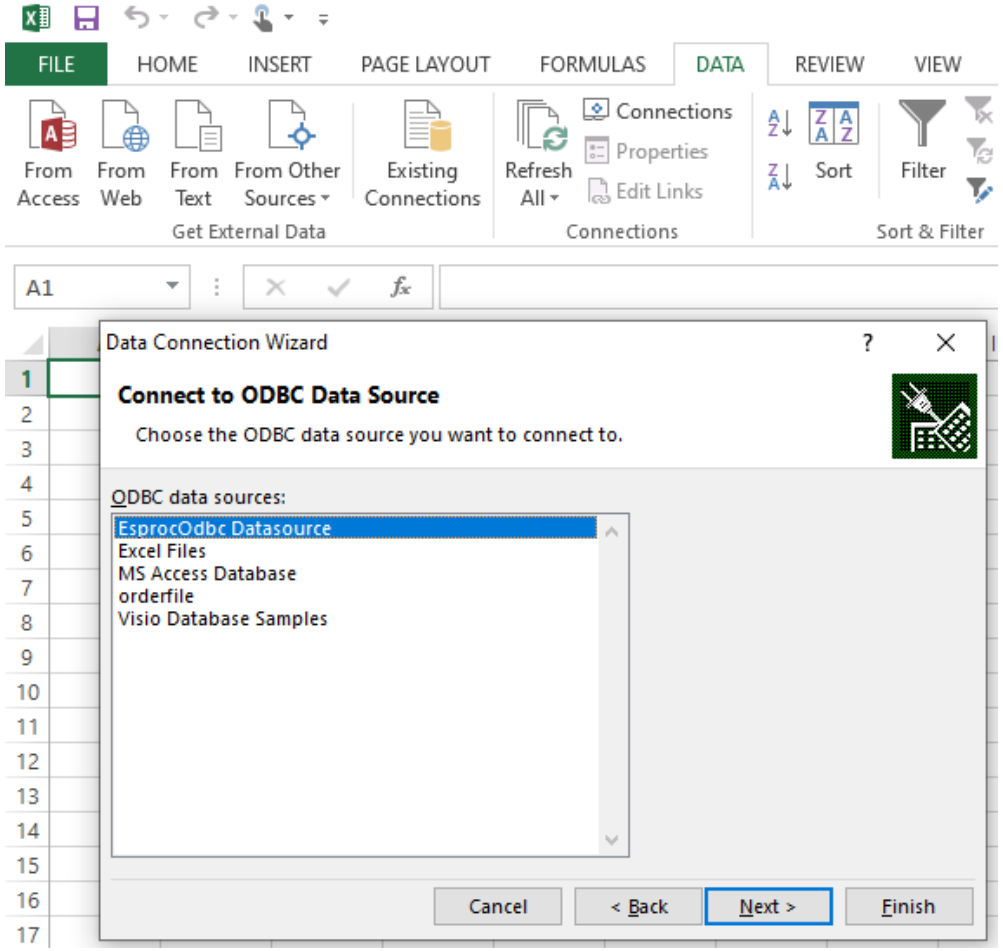


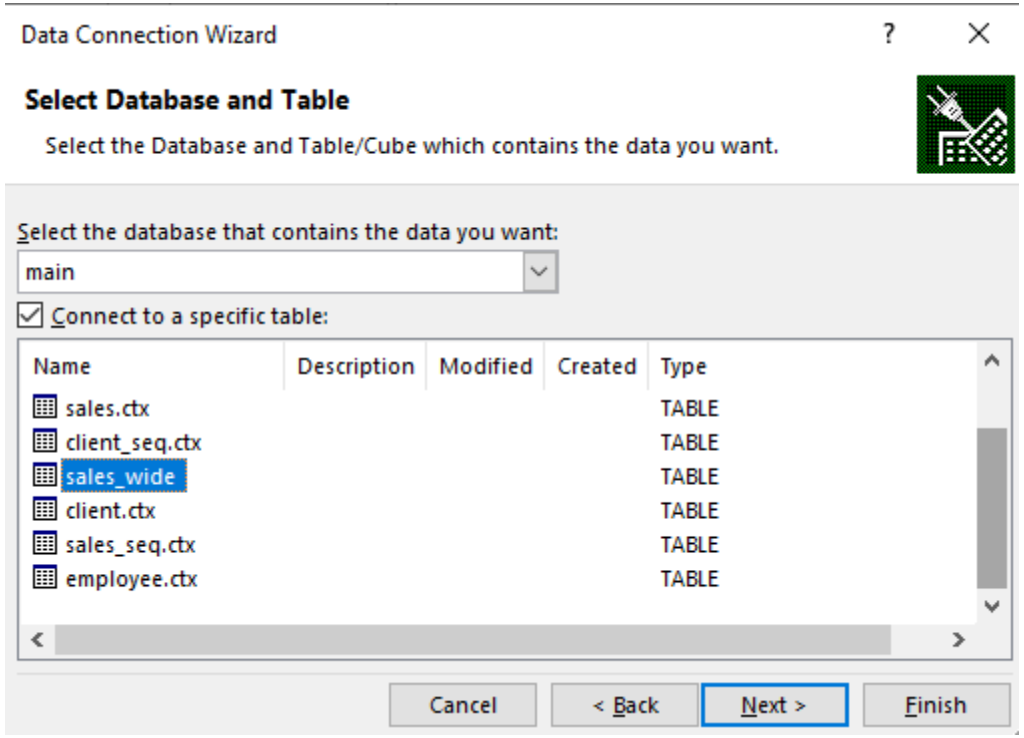


说明：ODBC配置请参考http://doc.raqsoft.com.cn/esproc/tutorial/odbcbushu.html

ODBC接口

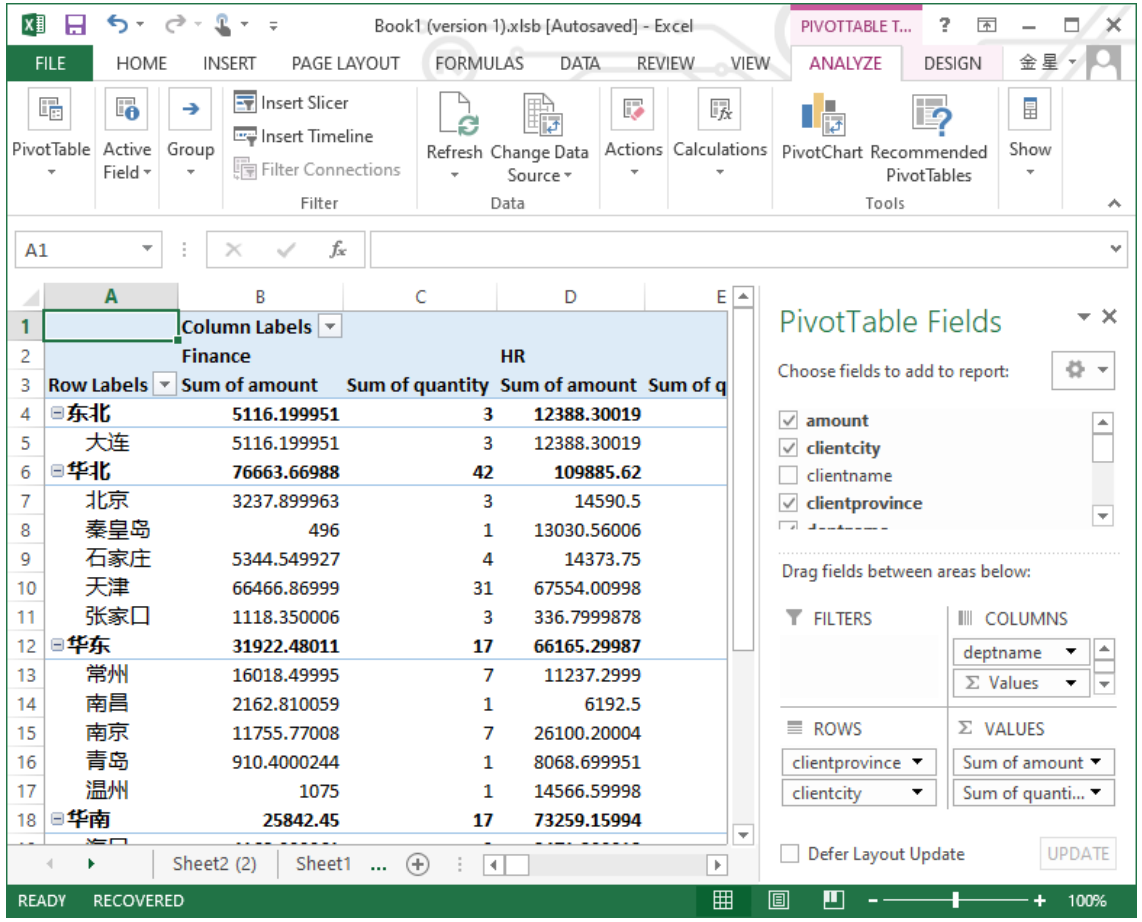
6. 在Excel中获取集算服务元数据





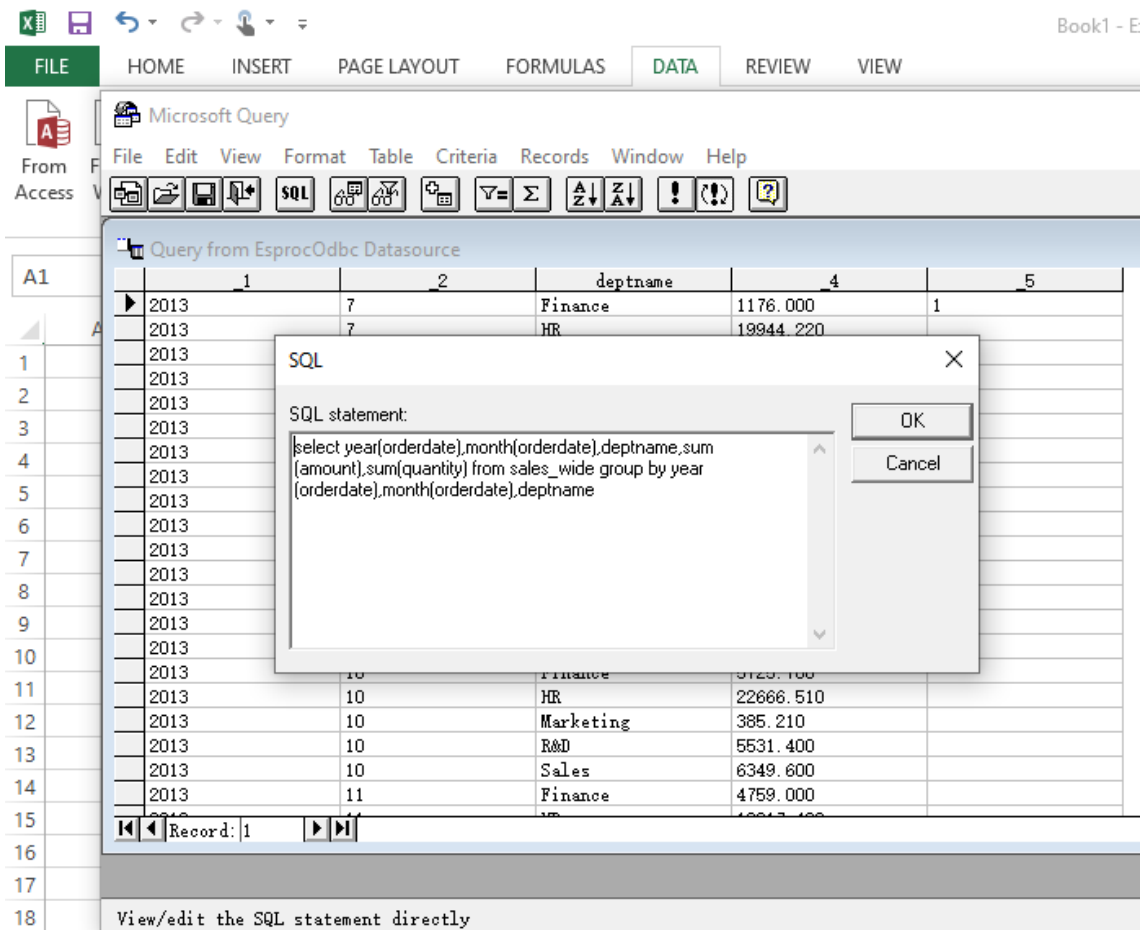
ODBC接口

7.以pivot Table或sheet的形式使用集算器数据



	A	B	C	D	E	F
1	orderdate	clientname	eid	gender	deptname	clientprovince
2	2013/7/4	山泰企业	5	F	R&D	华北
3	2013/7/5	东帝望	6	M	Sales	华东
4	2013/7/8	千固	3	F	Sales	华北
5	2013/7/8	实翼	4	F	HR	华东
6	2013/7/9	福星制衣厂股份有限公司	4	F	HR	华北
7	2013/7/10	实翼	3	F	Sales	华东
8	2013/7/11	浩天旅行社	5	F	R&D	华北
9	2013/7/12	永大企业	9	F	HR	华东
10	2013/7/15	凯诚国际顾问公司	3	F	Sales	华南
11	2013/7/16	远东开发	4	F	HR	华南
12	2013/7/17	正人资源	1	F	R&D	华南
13	2013/7/18	三捷实业	4	F	HR	东北
14	2013/7/19	一诠精密工业	4	F	HR	华南
15	2013/7/19	兰格英语	4	F	HR	华北

8.Excel也可通过SQL直接获取集算器数据



更多应用

凡支持JDBC\ODBC的应用，都可以通过集算器获得更高的计算性能

- ETL工具
  - Kettle
  - Apache Camel
  - informatica
- Web中间件
  - WebSphere
  - Weblogic
  - Tomcat
  - Microsoft IIS
- ……

润乾 RAQSOFT

想要了解更多 请联系我们

技术内容请移步 乾学院
http://c.raqsoft.com.cn

优惠价购买请加入 好多乾
http://sys.misdiy.com/hdq.html

www.raqsoft.com.cn