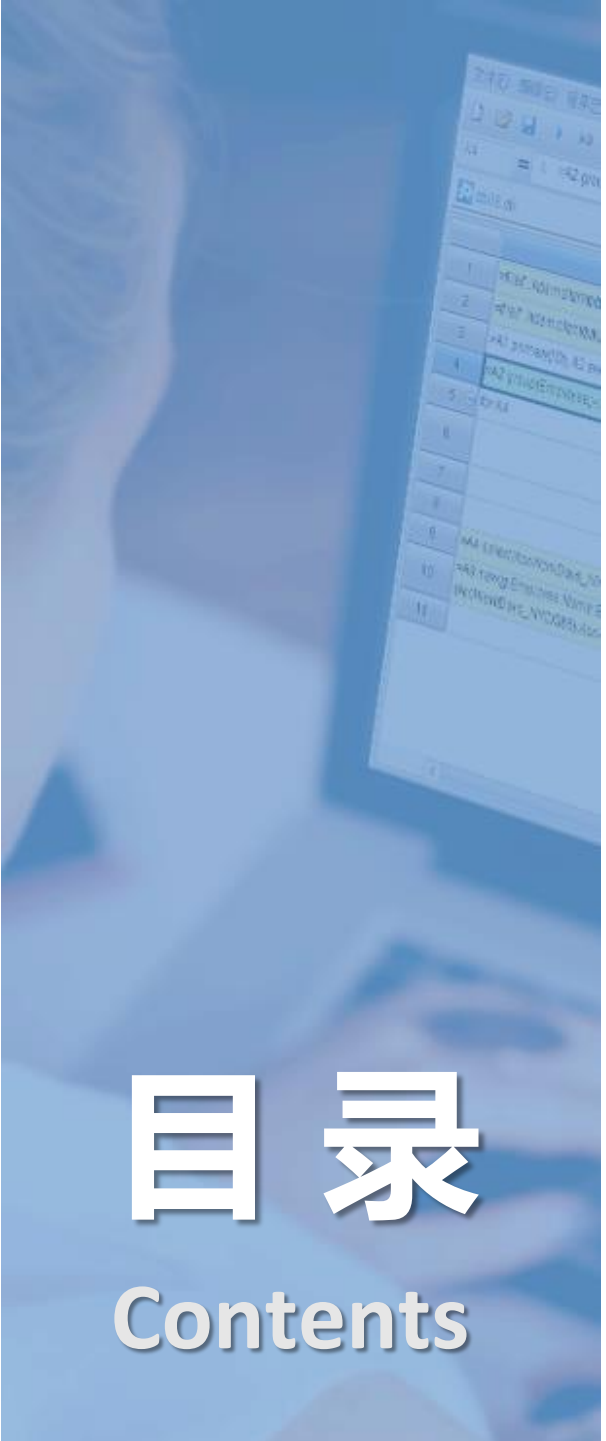
轻量级高性能文件型数据仓库

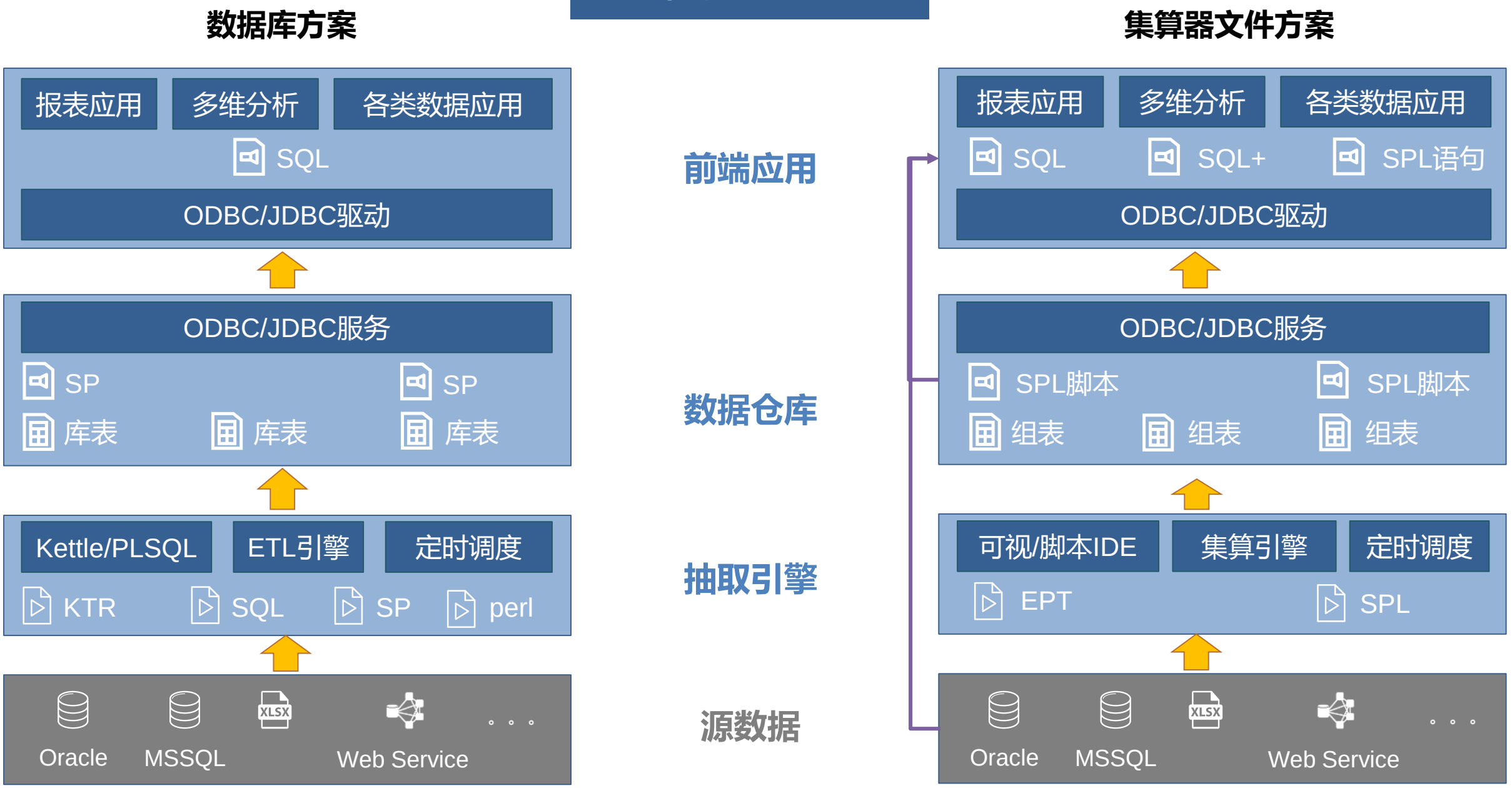
集算器应用场景实施方案



目 录

Contents

架构对比


数据库方案
集算器文件方案
报表应用
多维分析
各类数据应用
SQL
SQL+
SPL语句
ODBC/JDBC驱动
ODBC/JDBC服务
SP
SP
SPL脚本
SPL脚本
库表
库表
库表
组表
组表
组表
Kettle/PLSQL
ETL引擎
定时调度
可视/脚本IDE
集算引擎
定时调度
KTR
SQL
SP
perl
EPT
SPL
前端应用
数据仓库
抽取引擎
源数据
Oracle
MSSQL
XLSX
Web Service
. . .

特点对比

相同点

- 逻辑架构一致
- 搭建过程类似

主要区别

- 存储方式

  集算器方案：文件

  数据库方案：库表

- 查询方式

  集算器方案：SPL语句

  数据库方案：SQL语句

优势

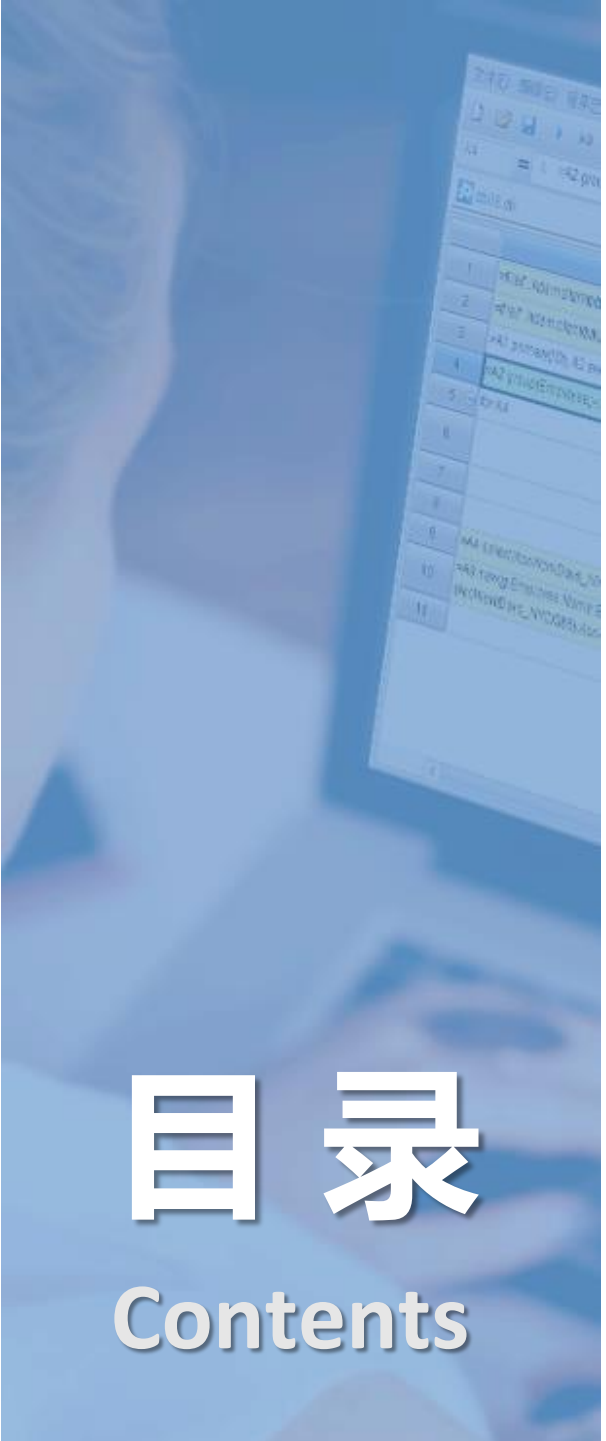
集算器底层能力	数据仓库特性	优势
离散数据集 高效存储	计算性能更高 存储密度更高 可取代昂贵的数据库	降低建设成本
多数据源计算能力	可同时计算内部组表和外部数据 可实现T+0混算	支持开放性存储
支持文件系统	可按多层目录管理脚本和数据	目录管理灵活方便
算法写在脚本中	前端应用可通过名字调用数据仓库算法 修改算法后无须编译无须停机	易于热切换
文件存储 文件计算	文件IO性能远远高于数据库 SPL可以写出高性能算法	计算性能高

存储 → 准备 → 使用

- 分析数据结构与特征
- 设计通用存储结构
- 根据特殊需求设计特殊（冗余）存储结构

- 分析源数据特征
- 设计并实现全量抽取
- 设计并实现增量抽取
- 特殊抽取过程（非SQL源、多源、宽表）

- 设计业务计算逻辑
- 前端应用查询数据



目录

Contents

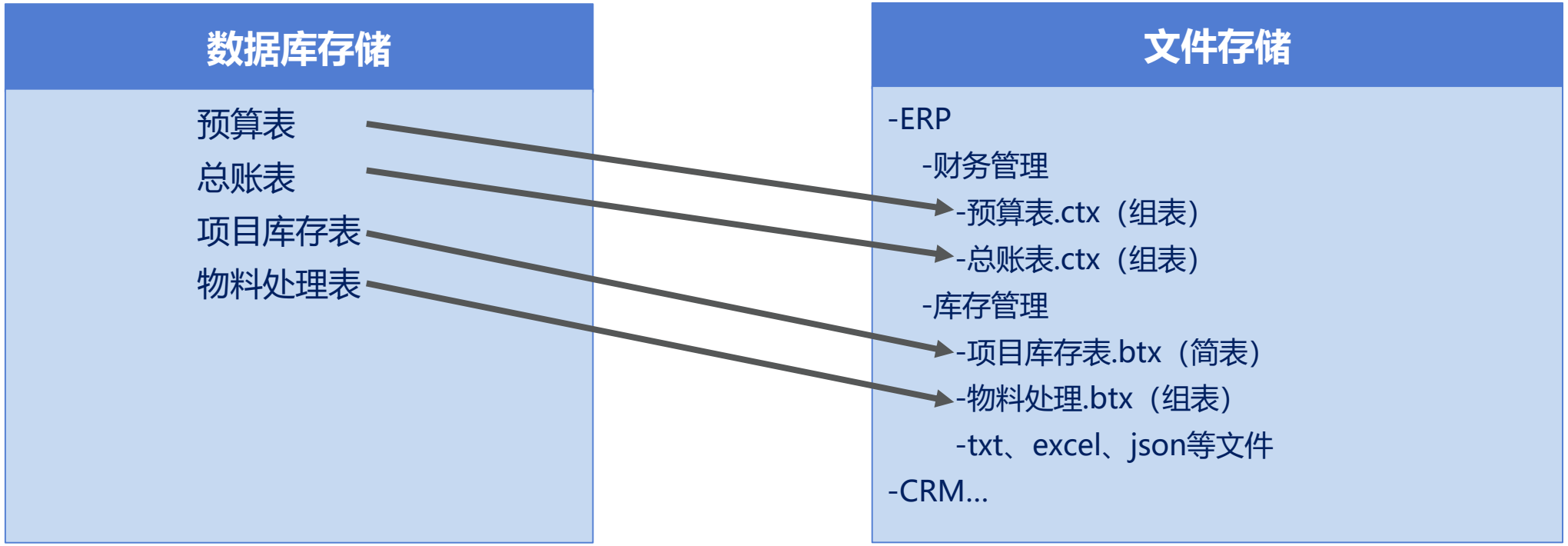
设计基础

一、库表对应集算器文件

数据库型数仓的每个库表，可粗略对应文件型数仓的每个集算器文件。可以用操作系统多级目录管理数据文件。

二、公共格式文件无须入库

txt、excel、json等文件，存入数据库型数仓时必须转为库表，存入文件型数仓时则无须转为集算器文件（除非体积过大等特殊情况）



设计基础

三、根据场景选择合适的文件格式

集算器文件格式有两种：组表、简表，分别适用不同的场景，其中组表适用场景更广。

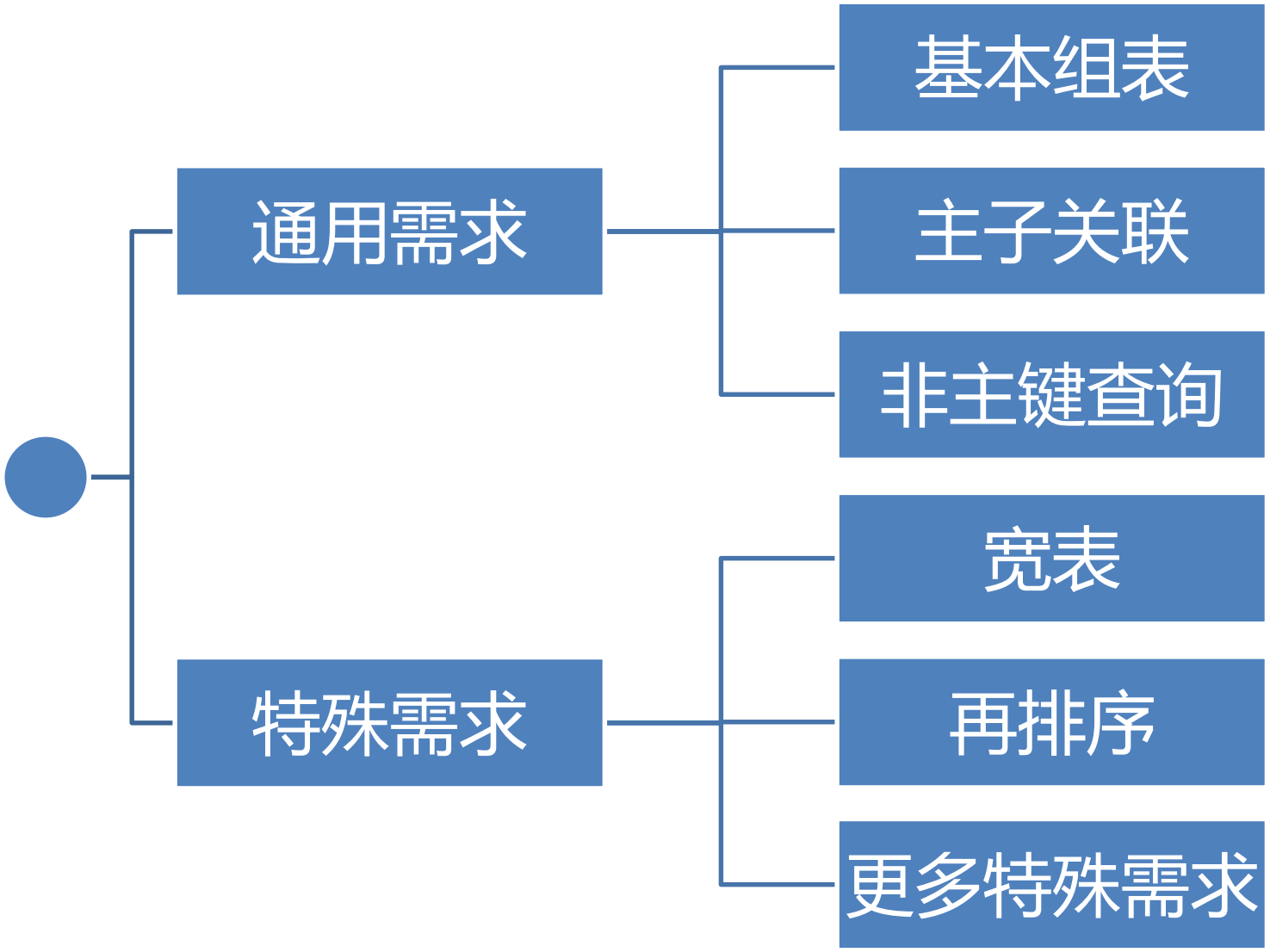
组表ctx

- 常规存储格式
- 压缩存储
- 高性能计算
- 自由列计算（列存）
- 固定字段查询（索引）
- 固定字段汇总（预汇总）
- 可追加可少量删改

简表btx

- 简易存储格式
- 少量快速读取（无压缩）
- 计算大部分字段（行存）
- 资源占用小
- 可追加不可改写

学习路径



基本组表

无特殊需求时，按主键排序建立组表

例子代码：以ID为主键，创建组表students.ctx

```
=file("students.ctx").create(#ID,NAME,GENDER,AGE)
```

将来可存储如下数据，注意数据须按ID排序

ID	NAME	GENDER	AGE
1	Emily	F	17
2	Elizabeth	F	16
3	Sean	M	17
4	Lauren	F	15
5	Michael	M	16

使用场景：通用计算；查询字段自由不固定；计算结果集较大。

说明：如何向组表灌入有序数据，将在后续抽取整理部分讲到。创建简表可用f.export@b()函数，本文不做过多涉及，详情参考http://doc.raqsoft.com.cn/esproc/func/fexportaxfs.html#f_export_A_x_F_s

基本组表

多个主键时，可将重复值较多的键放在前面，以提高压缩效率

压缩效率更高

=file(“stock.ctx”).create(#stockCode,#transTime,price)

stockCode	transTime	price
601988	2015-05-13 10:00:00	4.01
601988	2015-05-13 10:00:01	4.01
601988	2015-05-13 10:00:02	4.02
601988	2015-05-13 10:00:03	4.01
601988	2015-05-13 10:00:04	4.01

占用空间更多

=file(“stock.ctx”).create@y(#transTime,#stockCode,price)

transTime	stockCode	price
2015-05-13 10:00:00	600000	57.52
2015-05-13 10:00:00	601001	3.01
2015-05-13 10:00:00	601982	8.31
2015-05-13 10:00:00	601988	4.01
2015-05-13 10:00:00	602002	18.6

基本组表

无主键的情况

主键由1个或多个排序列组成，可确定组表的唯一记录。组表通常有主键，但有些情况下，可以没有主键只有排序列，甚至没有排序列。排序列即带#的字段（有序字段）

比如日志表源数据，数据按datatime生成且有重复，No虽然形式上是主键，但缺乏实际意义（不参与前端计算）

No	datatime	IP	content
241	2015-05-13 10:00:00	10.1.10.1	…
242	2015-05-13 10:00:00	27.1.3.27	…
243	2015-05-13 10:00:01	192.168.1.4	…
244	2015-05-13 10:00:01	101.4.10.4	…
245	2015-05-13 10:00:01	27.1.3.27	…

可以创建无主键的组表，即去掉No，并指定datatime为排序列

=file(“log.ctx”).create(#datatime,IP,content)

说明：无主键时（排序列组合有重复，或无排序列时），组表只能追加，不能删改。

基本组表

组表与库表的区别

	组表	库表
表之间关系	无物理外键和约束关系	可以有物理外键和约束关系
取数的有序性	顺序确定，天然有序	顺序不确定，须order by指定
小表分段取数	任意分段，方式简单	使用where分段，方式复杂
大表分段取数	任意分段，无须分表	通常用分表实现，维护困难

扩展阅读：组表的生成http://doc.raqsoft.com.cn/esproc/tutorial/zbdsc.html

主子关联

有两种常见的表关系：事实表维表关联、主子表关联

事实表维表关联：无特殊优化

主子关联：可进行同步分段，以提高关联计算性能

主表

存储主记录的表，比如订单、合同、报销单、电信账单。随着业务的发生，主表记录会不断增多。

子表

存储与主表记录对应的明细记录的表，比如订单明细、合同明细、报销明细、通话记录。

随着主表记录的不断增多，相应的子表记录也会增多。

主子关联

关系结构

order.ctx（主表）		
key	orderID	int
	custID	string
	orderDate	date

orderDetail.ctx（子表）		
key	orderID	int
key	productID	int
	price	float
	quantity	int

同步分段例子

=file(“order.ctx”).create(#orderID,custID,orderDate)

=file(“orderDetail.ctx”).create(#orderID,#productID,price,quantit; orderID)

组表数据

orderID	custID	orderDate
10248	VINET	2012-07-04
10249	TOMSP	2012-07-05
10250	HANAR	2012-07-08

orderID	productID	price	quantity
10248	17	14	12
10248	42	9	10
10248	72	34	5
10249	14	17	9
10249	51	42	40

说明：子表必须按主表的主键字段分段，例子中为orderID。

非主键查询

场景：对非主键的固定字段进行查询，且查询结果小时，可建立索引。

注意：针对主键的相关计算无须索引

	A	B
2	=file(“sales.ctx”).create(#orderDate, #region, #orderID, customer, quantity, price ,discount)	/创建组表，指定键
3	=A2.index(i_customer;customer)	/以字段customer建立索引，索引名为i_customer

说明：

1.可指定条件建立索引，可指定索引查询时输出的字段，可指定索引长度，可建立全文索引并进行模糊查询。

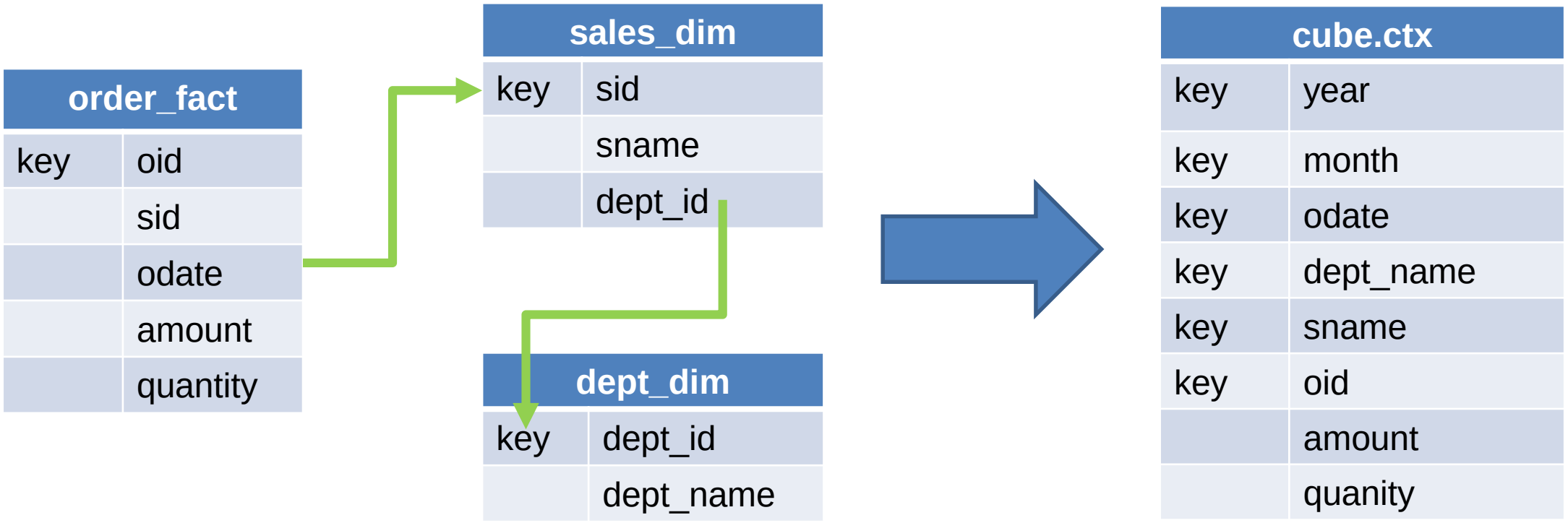
2.再次追加数据时，索引将自动维护。

扩展阅读：为组表创建检索文件http://doc.raqsoft.com.cn/esproc/func/tindexiwch.html#T_index_I

宽表

场景：当固定的join动作影响性能时，比如为OLAP建立cube，可建立宽表以减少join动作

比如：将订单-销售员-部门的三表关联，转为cube宽表



再排序

大部分数据须按时间过滤再计算，组表按时间优先排序可获得最佳计算性能。

特殊需求：按时间优先排序无法获得最佳性能，此时应按合理的字段再排序

比如：源数据是银行流水表，前端按时间查询时，可构造如下组表

bank_transaction.ctx	
key	data_date
key	branchID
key	account
	trans_amount
	balance

比如：同样的银行流水表，前端按账号查询时，可构造如下组表

bank_transaction.ctx	
key	account
key	data_date
key	branchID
	trans_amount
	balance

再排序

场景：有时源数据和目标组表的排序顺序不同。

比如：源数据报税单按时间生成，但目标组表需按税单号查询

优化方法：设计新、旧两个同构的组表，都按税单号排序，新组表存储近期（比如近一个月）的数据，旧组表存储历史(但不含近一个月)的数据。计算时可将两者归并为文件组，按全量数据查询

taxMon.ctx	
key	cardNo
	declareTime
	area
	declareType
	unit
	network

taxHis.ctx	
key	cardNo
	declareTime
	area
	declareType
	unit
	network

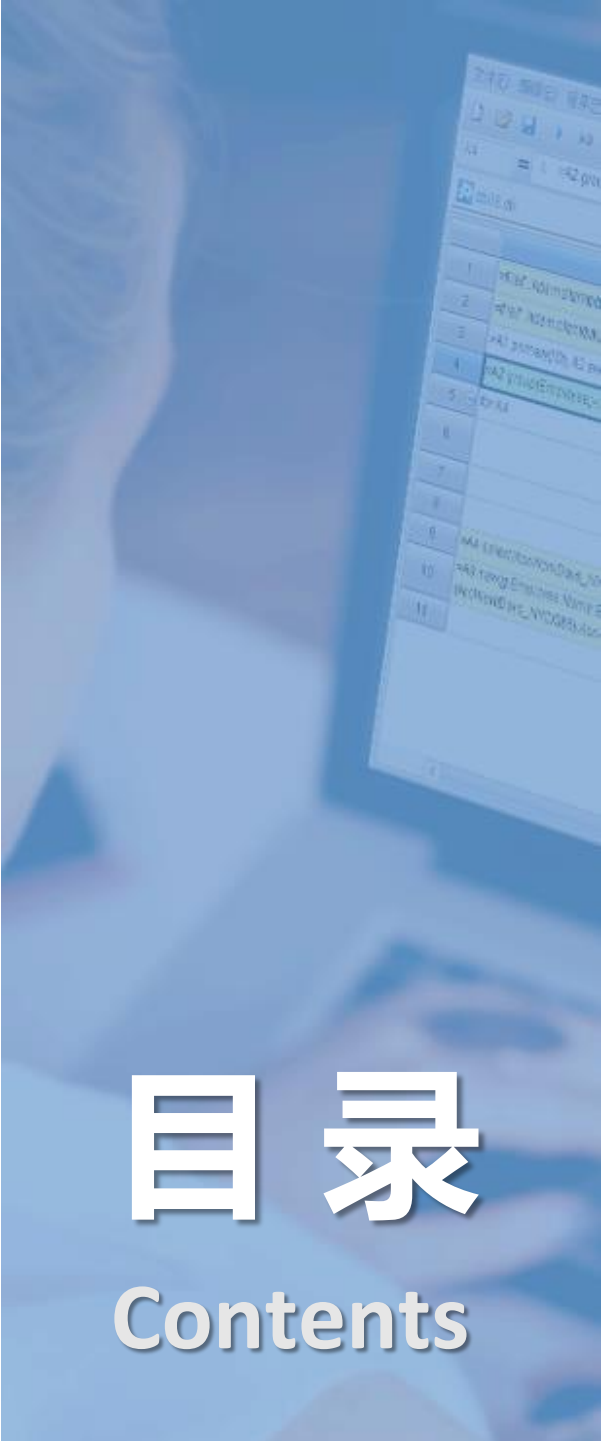
说明：新组表须定期（比如月初）合并到旧组表，计算时新旧组表须归并，这部分内容后面再讲。

更多特殊需求

前面两种特殊需求比较常见，实际应用中存在更多特殊需求，比如特殊join运算、遍历计算、应用排号键等。

针对不同的特殊需求，集算器提供了多种灵活的存储方案，详情参考官方文档

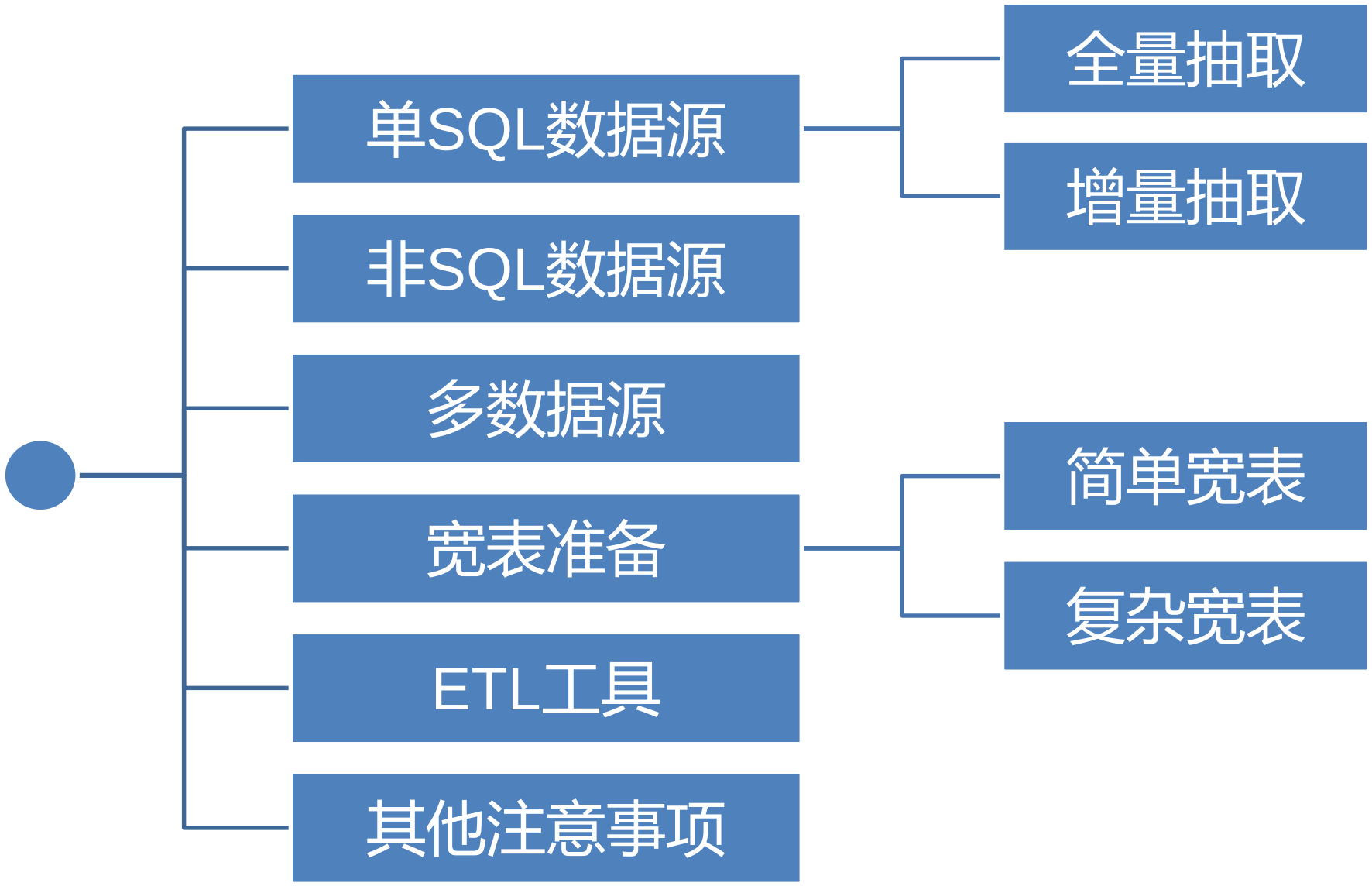
- SPL组表进一步优化 JOIN 性能 http://c.raqsoft.com.cn/article/1545631312033
- 性能优化技巧 - 遍历 http://c.raqsoft.com.cn/article/1551946752372
- 性能优化技巧 - 多层排号键 http://c.raqsoft.com.cn/article/1550218072302
- SPL教案汇总 http://c.raqsoft.com.cn/article/1567908371148



目录

Contents

学习路径

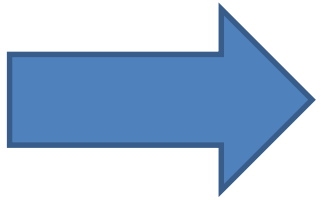


全量抽取

场景：适用于历史数据一次性抽取，或小表更新后抽取。
注意：关系数据库无法保证取出次序，需用order by按主键排序

数据库源表stock

tradeDate	stockCode	price
2015-01-05	792	57.52
2015-01-05	601988	3.01
2015-01-06	62	8.31
2015-01-06	600005	4.01
2015-01-06	2242	18.6



目标组表stock.ctx，键为stockCode、 tradeDate

stockCode	tradeDate	price
601988	2015-05-13	13.3
601988	2015-05-14	12.04
601989	2015-05-15	11.58
601989	2015-04-30	10
601989	2015-09-04	17.54

SPL抽取脚本，在SQL中排序：

	A	B
1	=connect（“db”）@l.cursor（“stockCode,tradeDate,price from stock order by stockCode,tradeDate”）	/排序
2	=file(” stock.ctx”).create(#stockCode,#tradeDate,price)	/创建组表
3	=A2.append(A1)	/加载

也可以在SPL中排序：

=connect（“db”）.cursor（“stockCode,tradeDate,price from stock”）.sortx(stockCode,tradeDate)

全量抽取

场景：对于全量变化的大表，如电商订单状态、人口信息，抽取时应根据数据特点设计抽取方法，从而尽量提高性能。

1.大量更新时，重新生成所有数据，即无优化全量抽取。

2.少量更新(<几十万行），涉及远期数据时（比如1年前），可使用update/delete更新组表，积累一段时间后（比如一个月），再使用reset重整整个表。

3.少量更新(<几十万行），只涉及近期数据时（比如本月），每次使用update/delete更新组表，再用reset@q快速重整数据。

全量抽取

例如：组表contract.ctx存储合同信息，合同30天内允许修改，且变更记录较少，则SPL抽取脚本如下：

	A	B
1	=modify_add=orcl.query("select * from contract where tag='modify_add' and datadate>=? order by orderdate",elapse(now(),-30))	/根据标志查询修改或新增的数据，限30天内
2	=delete=orcl.query("select * from census where tag= 'delete' and datadate>=? order by id",elapse(now(),-30))	/根据标志查询删除的数据，限30天内
3	=file("orders.ctx").create()	/打开现有组表
4	=A3.update(modify_add)	/修改和新增
5	=A3.delete(delete)	/删除
6	=A3.reset@q()	/重整组表

说明：修改的数据不能太多，否则性能变差，合适的数据量不超几十万条

扩展阅读：性能优化技巧 - 组表数据更新http://c.raqsoft.com.cn/article/1550644911917

增量抽取

场景：适用于向历史组表定时追加数据（无修改删除）。

注意：源数据须有时间戳，以便取增量。增量数据须和历史数据同序。

凌晨时抽取昨日数据，追加到组表bank_transaction.ctx

	A	B
1	=connect@l（“db”）.cursor@x（“select data_date,branchID,account,trans_amount,balance from bank_transaction order by data_date,branchID,account where data_date=?“, pDate)	按日期取增量数据
2	=file（“bank_transactio.ctx”）.create(). append(A1)	追加到现有组表

说明：取增量的方法较为灵活，可按时间区间取数，也可按日期取数。上面例子里pData是参数，由外部计算出昨日日期并传入，这样可以方便控制要追加哪天的数据。如果要固定抽取昨日数据，也可用SPL表达式date(elapse(now(),-1))代替pDate。

扩展阅读：Oracle有独特的OGG增量更新方式，集算器支持这一用法，详见OGG 增量采集数据入库

http://c.raqsoft.com.cn/article/1567582134531

增量抽取

场景：增量抽取时，源数据和目标组表的排序顺序不同。

方法1：单组表归并追加

1.从源数据取增量，并按键排序。

2.用归并的方式追加到组表中，此时组表会重写，但比全量重新排序快得多。

源数据报税单按时间生成，但目标组表需按税单号查询，则SPL抽取脚本如下

	A	B
1	=orcl.curosr@x(“select * from tax where declareTime=? order by cardNo”, pData)	/取每日新增数据
2	=file(“taxMon.ctx”).create().append@m(A1)	/归并追加

增量抽取

场景：增量抽取时，源数据和目标组表的排序顺序不同。
方法2：双组表归并重整
1.设计新、旧两个同构的组表，都按键排序。
2.新组表存储近期（比如近一个月）的数据，每日追加归并新数据。
3.旧组表存储历史(但不含近一个月)的数据，定期（比如月初）合并新旧数据。
说明： 因为每次只对小组表归并追加，因此本方法比方法1更快

	A	B	C
1	if day(now())==1	=file(["taxHis.ctx","taxMon.ctx"]).reset@m()	/月初高性能归并重整
2	=orcl.curosr@x(“select * from tax where declareTime=? order by cardNo”, pData)		/取每日新增数据
3	=file("taxMon.ctx").create().append@m(A2)		/每日高性能追加

说明：前端计算时，新旧组表需归并再计算，这部分内容后面会讲到。

非SQL数据源

场景：对于文本文件、Excel、json等公共格式的非SQL数据源，可用SPL内置函数直接抽取

源数据students.txt，分隔符为tab ，已按ID字段排序

ID	NAME	GENDER	AGE
1	Emily	F	17
2	Elizabeth	F	16
3	Sean	M	17
4	Lauren	F	15
5	Michael	M	16

目标组表students.ctx，键为ID(与源同序) ，则SPL抽取脚本如下：

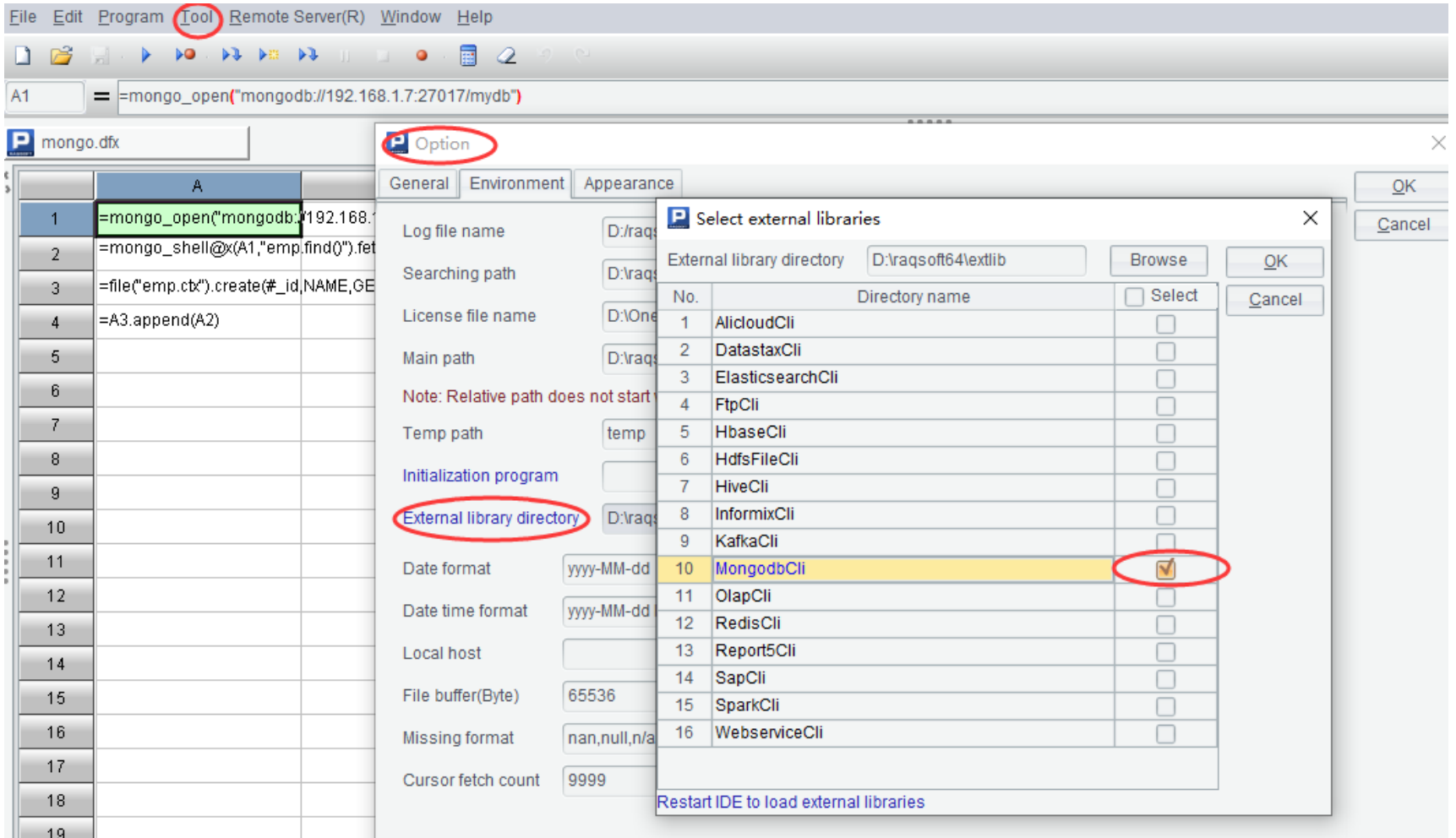
	A	B
1	=file(“stdutents.txt”).cursor@t(ID,NAME,GNDER,AGE)	/游标取数
2	=file(“students.ctx”).create(#ID,NAME,GENDER,AGE)	/创建组表
3	=A2.append(A1)	/加载

如果目标与源不同序，可使用sortx在SPL中排序

非SQL数据源

场景：针对webService、MongoDB等非SQL数据源，以及Hive等支持高性能接口的数据源，集算器提供了各类外部库，以方便抽取整理。

配置外部库



非SQL数据源

访问MongoDB，使用shell命令查询emp collection，并写入组表

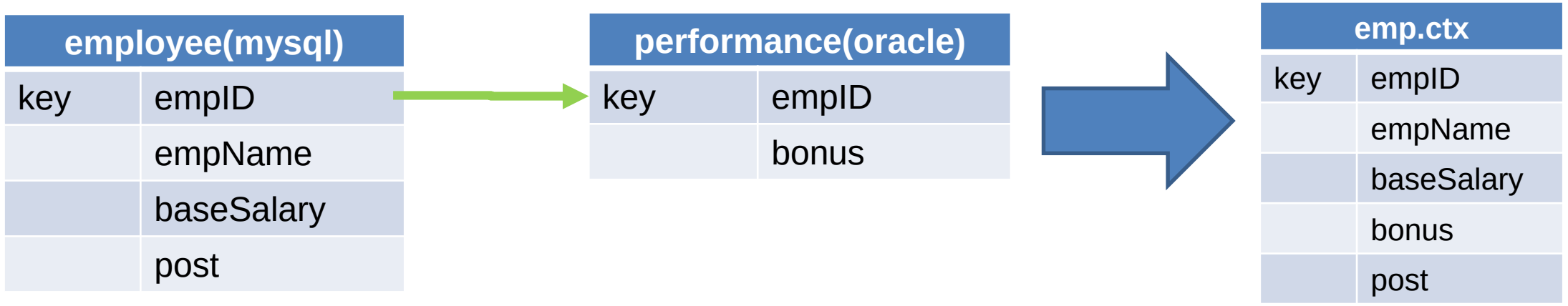
	A	B
1	=mongo_open("mongodb://192.168.1.7:27017/mydb")	/连接
2	=mongo_shell@x(A1,"emp.find()").sortx(#_id)	/查询
3	=file("emp.ctx").create(#_id,NAME,GENDER,BIRTHDAY,DEPT,SALARY)	/创建组表
4	=A3.append(A2)	/加载数据

扩展阅读：更多外部库用法，参考http://doc.raqsoft.com/esproc/func/wbk.html

多源数据

场景： 从不同的数据源抽取数据，进行关联计算后，将计算结果加载到组表

例如：将Oracle的绩效表和mysql中的员工表经过关联计算，加载到组表emp.tct



多源数据

SPL脚本

	A	B
1	=connect（“orcl”）.cursor@x（“select * from performance”）	/绩效
2	=connect（“mysql”）.query@x（“select * from employee order by empID”）	/员工
3	=A2.switch(empID,A1:empID)	/跨库关联
4	=A3.new(empID.empID:empID,empName,baseSalay,empID.bonus:bonus,post)	/取字段
5	=file(“ emp.ctx”）.create(#empID,empName,baseSalary,bonus,post)	/建组表
6	=A5.append(A4)	/加载数据

数据库数仓

将异库数据预先抽取到同库，再进行ETL。

或使用缺乏结构化计算函数的高级语言进行硬编码跨库计算。

非数据库的多源ETL将更复杂

文件数仓

使用结构化计算函数，直接进行多源ETL

扩展阅读：

跨库数据表的运算 http://c.raqsoft.com.cn/article/1536666621882

协助报表开发之 MongoDB join mysql http://c.raqsoft.com.cn/article/1568710911993

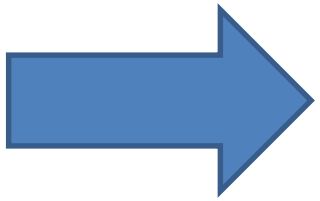
简单宽表

简单的单源宽表可用SQL取数生成

order_fact	
key	oid
	sid
	odate
	amount
	quantity

sales_dim	
key	sid
	sname
	dept_id

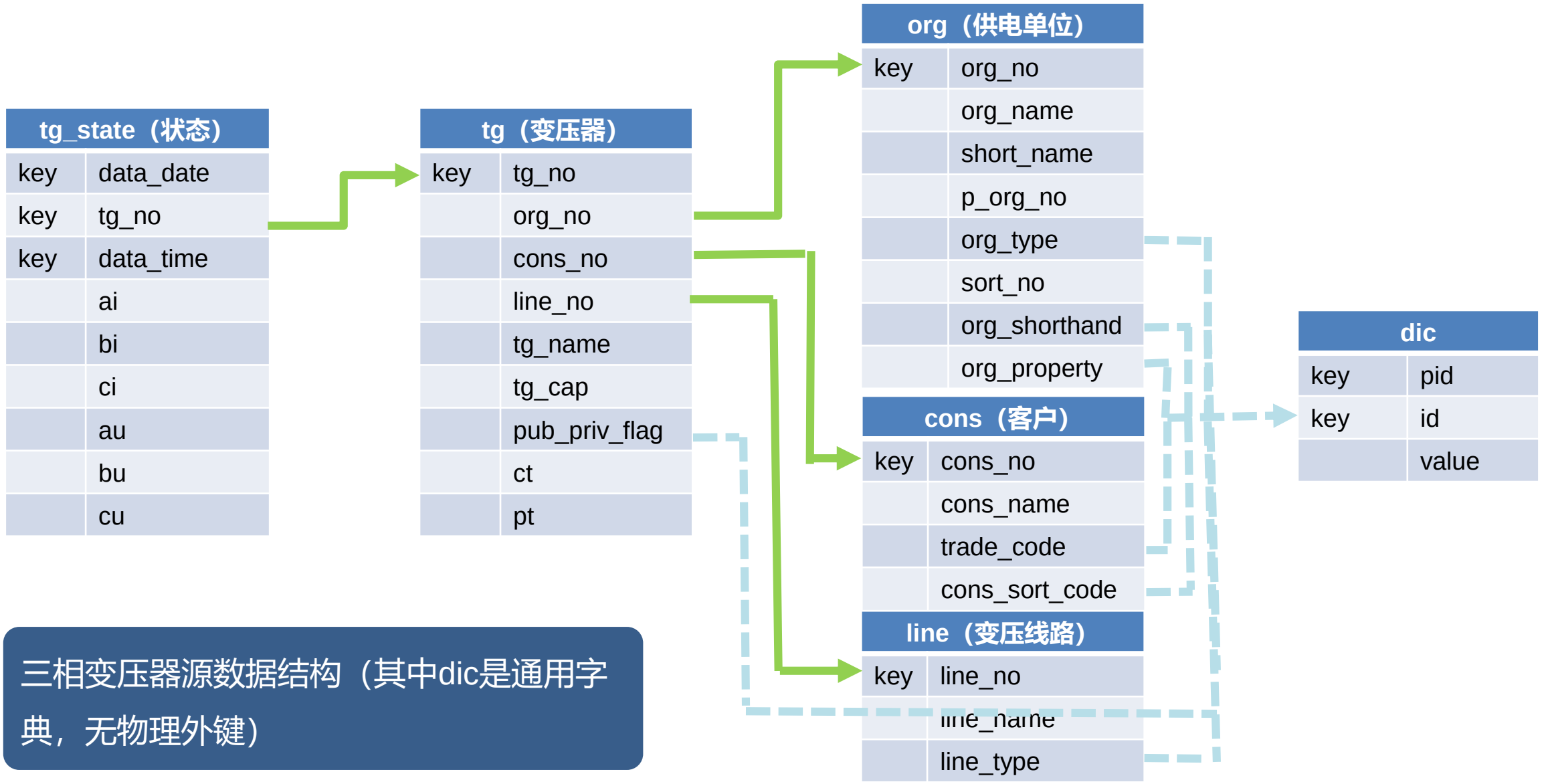
dept_dim	
key	dept_id
	dept_name



cube.ctx	
key	year
key	month
key	odate
key	dept_name
key	sname
key	oid
	amount
	quanity

	A	B
1	=connect(“db”).cursor@x(“select year(o.odate) year, month(o.odate) month, o.odate, d.dept_name,s.sname,o.oid,o.amount,o.quantity from order_fact o, sales_dim,s, dept_dim,d where o.sid=s.sid and s.dept_id=d.dept_id”)	/游标取数
2	=file(“students.ctx”).create(#year,#month,#odate,#dept_name,#sname,#oid,amount,quantity)	/创建组表
3	=A2.append(A1)	/加载

复杂宽表



三相变压器源数据结构（其中dic是通用字典，无物理外键）

复杂宽表

用SPL脚本生成三项变压器宽表，思路如下：

1.从各数据库（或非库）分别取5张表

2.根据表之间的关系特点，由下而上建立关联

3.使用对象引用式语法，简化多级关联查询，生成目标宽表。比如单级机构需展开为市、县、区三级

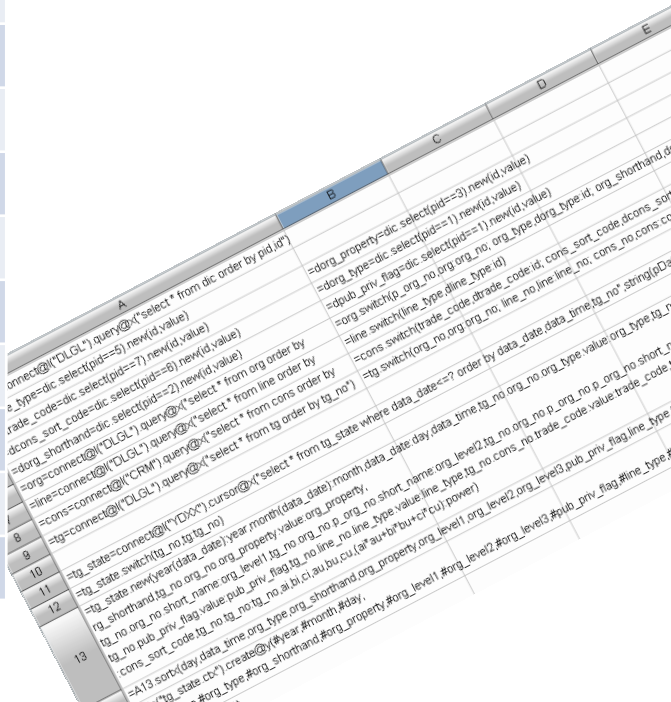
4.增加冗余维度：year(year(data_date))、month(month(data_date))

5.增加冗余测度：power（ai*au+bi*bu+ci*cu）

参考资料
SPL 简化 SQL 案例详解：多级关联
http://c.raqsoft.com.cn/article/1554168677971
SPL组表进一步优化 JOIN 性能
http://c.raqsoft.com.cn/article/1545631312033

tg_state.ctx	
key	year
key	month
key	day
key	data_time
key	org_type
key	org_shorthand
key	org_property
key	org_level1
key	org_level2
key	org_leve3
key	pub_priv_flag
key	line_type
key	trade_code
key	cons_sort_code
key	tg_no
	ai
	bi

	ci
	au
	bu
	cu
	du
	power



ETL工具

脚本IDE：适用于复杂ETL过程，参见前面例子

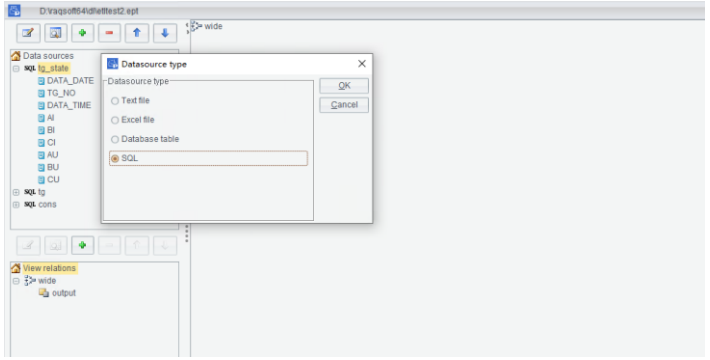
- 分步骤计算
- 计算列关联
- 行列转换
- 并行计算
- 高性能采集
- 外部库、命令行

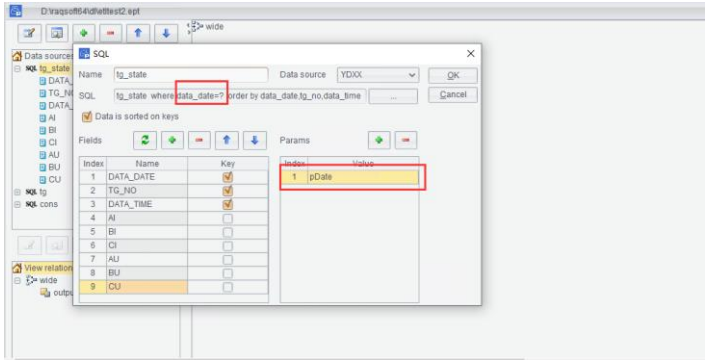
扩展阅读： SPL教案汇总 http://c.raqsoft.com.cn/article/1567908371148

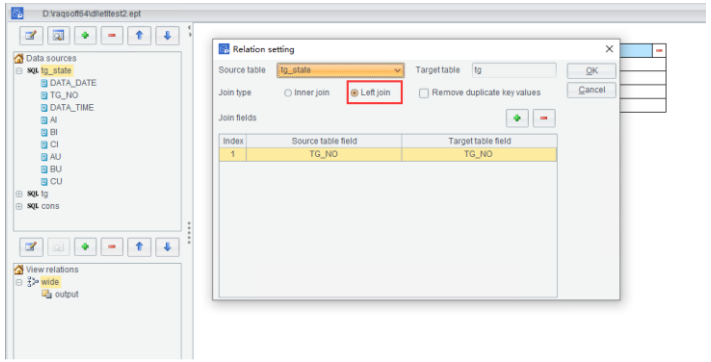
ETL工具

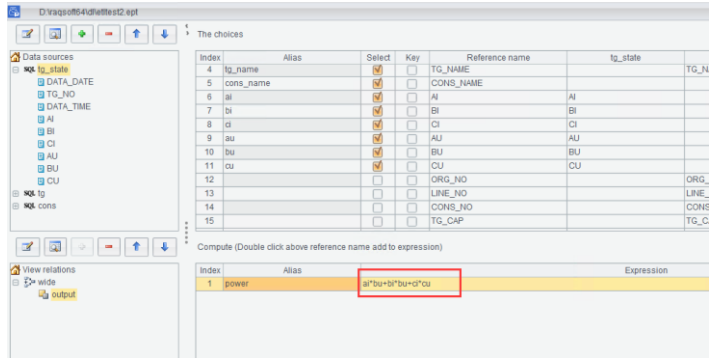
可视化IDE：适用于简单ETL过程，技术要求较低

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







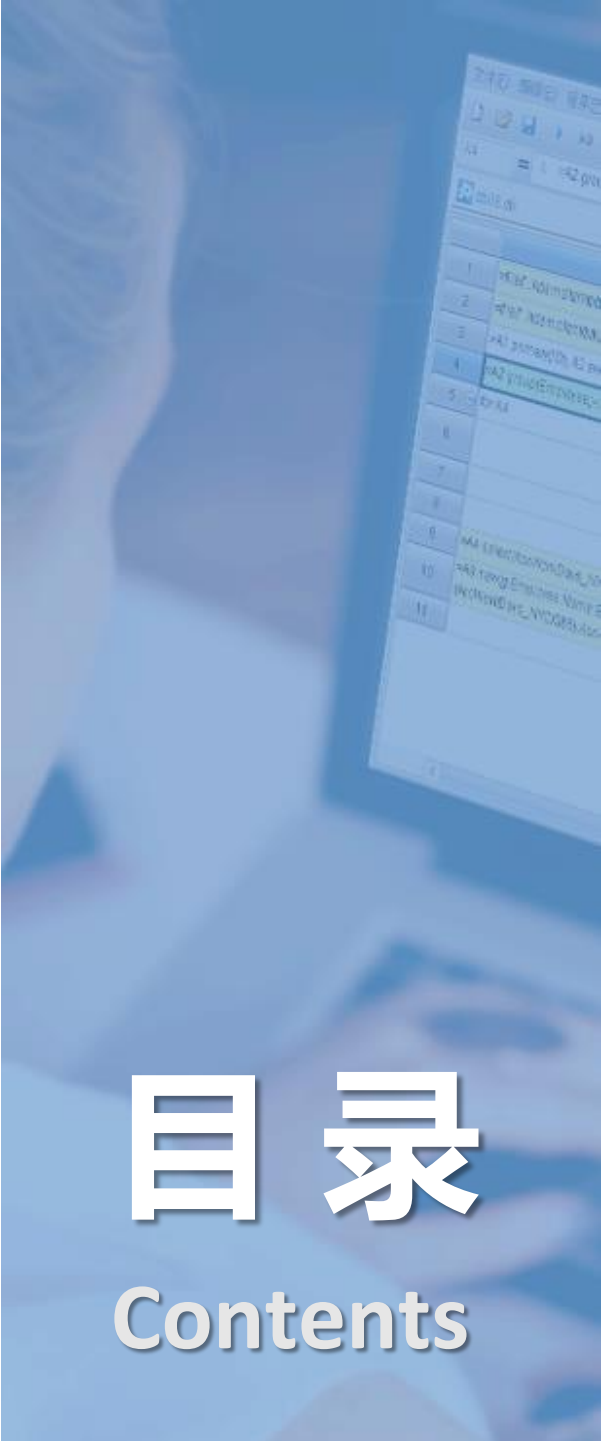


其他注意事项

ETL脚本的定时执行，可使用操作系统自带调度工具，如Crontab，或第三方可视化工具如opencron。

更新组表如果出错，可以使用rollback函数恢复之前的状态，详见 http://doc.raqsoft.com/esproc/func/frollback.html

更新时组表不能查询，需要停止服务或在空闲时段更新，为改善这一状况，集算器将支持“允许同时读写”的组表。



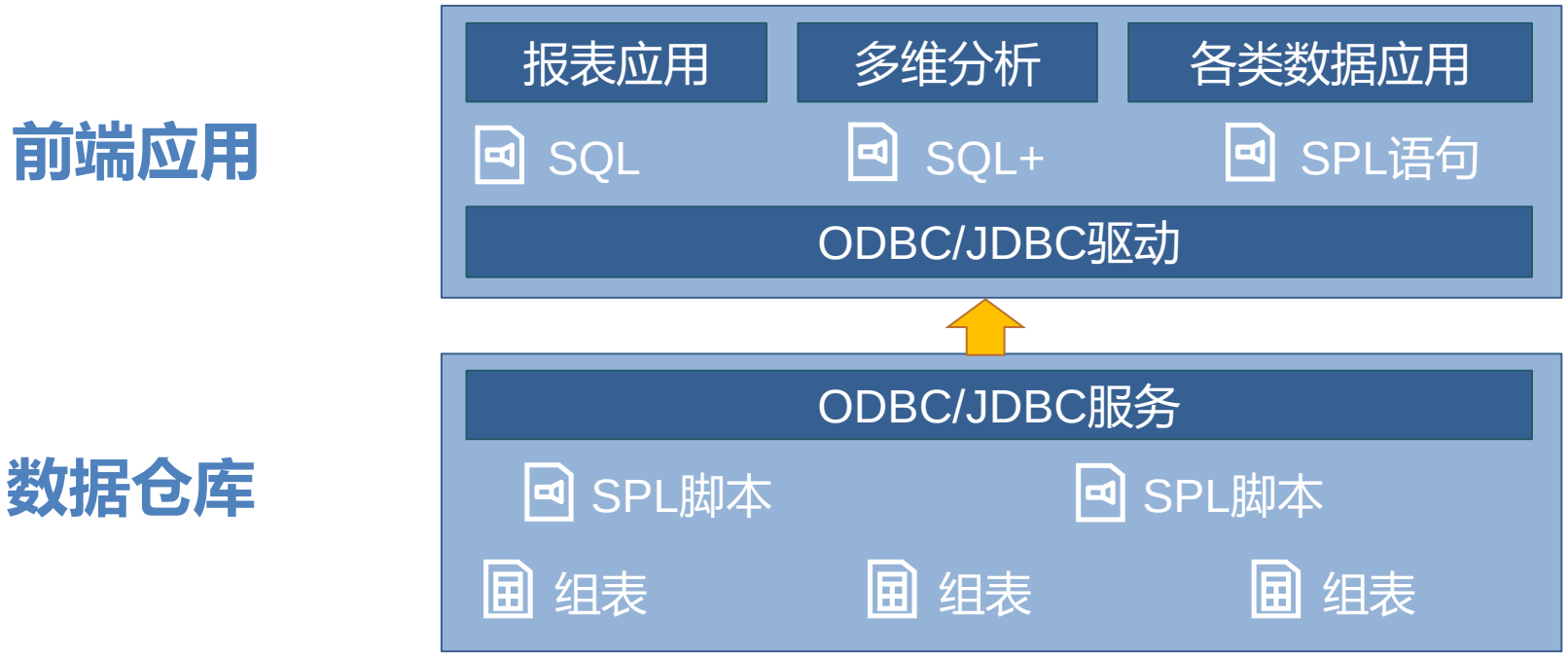
目录

Contents

对外接口

文件型数仓对外提供ODBC/JDBC接口

使用独立JDBC/ODBC驱动访问网络数据
耦合性低，可扩展为多台前端应用



说明：除了上述独立式服务架构，文件数仓也支持嵌入式服务架构，即前端应用与数仓服务合并

两种查询方式

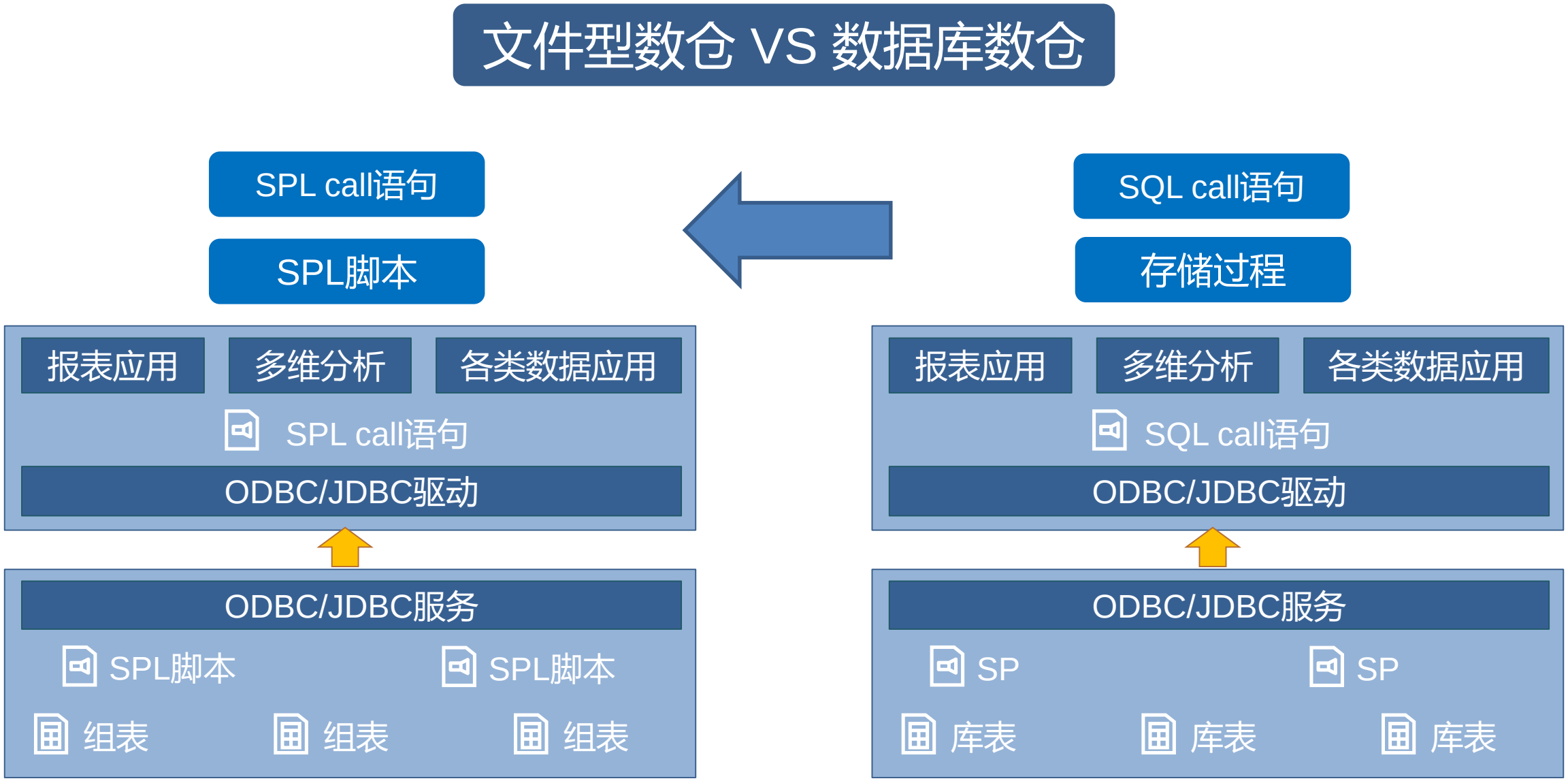
存储过程方式

适用于算法复杂、性能要求很高的场景

查询语句方式

适用于算法较简单、性能要求较高的场景

存储过程方式


文件型数仓 VS 数据库数仓
SPL call语句
SPL脚本
SQL call语句
存储过程
报表应用
多维分析
各类数据应用
SPL call语句
ODBC/JDBC驱动
ODBC/JDBC服务
SPL脚本
SPL脚本
组表
组表
组表
报表应用
多维分析
各类数据应用
SQL call语句
ODBC/JDBC驱动
ODBC/JDBC服务
SP
SP
库表
库表
库表

存储过程方式

	文件数仓
java调用	Class.forName("com.esproc.jdbc.InternalDriver"); con =DriverManager.getConnection("jdbc:esproc:local://onlyServer=true"); ResultSet rs = con.executeQuery(“call calc(?)");
SPL call语句	call calc(?)
SPL脚本文件名	calc.dfx

	数据库数仓
java调用	Class.forName("oracle.jdbc.driver.OracleDriver"); con =DriverManager.getConnection("jdbc:oracle:thin:@//192.168.0.2:1521/orcl"); ResultSet rs = con.executeQuery(“call calc(?)");
SP call 语句	call calc(?)
SP名	calc

说明：访问数据仓库时，需在前端应用配置服务器地址，详见 http://doc.raqsoft.com.cn/esproc/tutorial/pzraqsoftconfig.html 。报表工具应使用存储过程数据集，类似 “call calc(?)”，其用法与访问普通存储过程相同，参见集算器助力ireport http://c.raqsoft.com.cn/article/1560233466960

存储过程方式

与数据文件一样，SPL文件也存放于操作系统目录

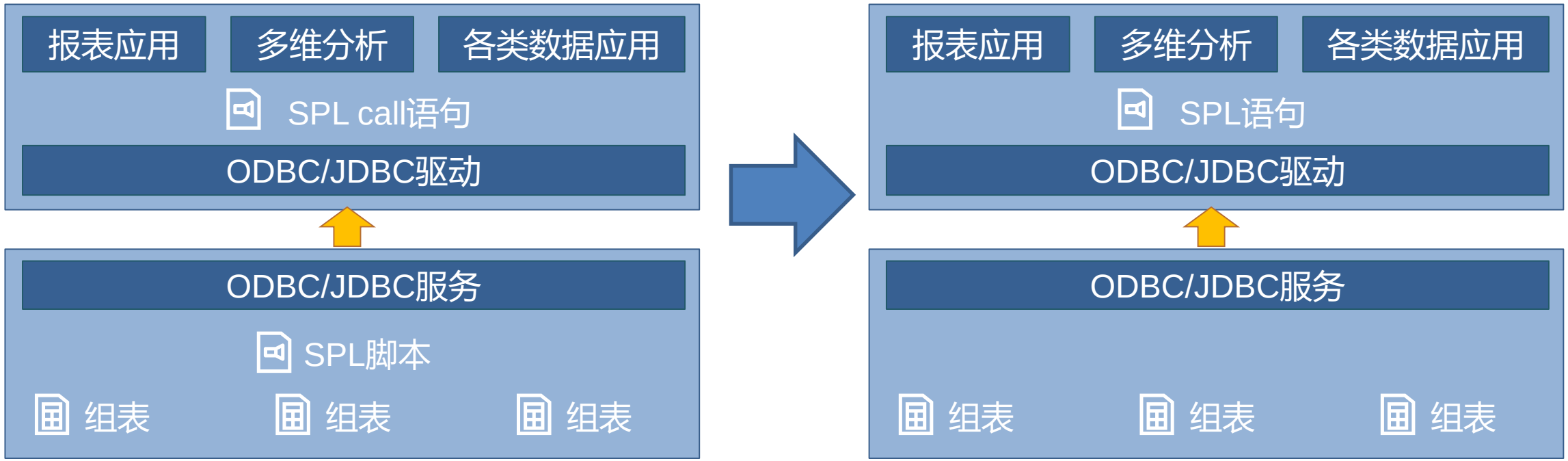
通常情况下建议按功能模块划分目录，也可按应用类型、时间、版本划分目录

```
-主目录
    -客户管理
    -考勤绩效
    -企业资源管理
        -财务管理
            -固定资产统计.dfx
            -现金流量查询.dfx
            -呆账坏账预警分析.dfx
        -库存管理
            …
```

前端报表调用示例： call 企业资源管理/财务管理/现金流量查询(?,?)

查询语句方式

如果SPL脚本文件较简单，可用前端SPL语句替代服务端SPL脚本



查询语句方式

	存储过程方式
java调用	Class.forName("com.esproc.jdbc.InternalDriver"); con=DriverManager.getConnection("jdbc:esproc:local://onlyServer=true"); ResultSet rs = con.executeQuery(“call run()");
SPL call语句	call run()

	SPL查询语句方式
java调用	Class.forName("oracle.jdbc.driver.OracleDriver"); con=DriverManager.getConnection("jdbc:oracle:thin:@//192.168.0.2:1521/orcl"); ResultSet rs=con.executeQuery(“=file(\”emp_index.ctx\“).create().icursor(;DEPT==\"HR\").fetch()");
SPL语句	=file(\”emp_index.ctx\“).create().icursor(;DEPT==\"HR\").fetch()

脚本文件

	A
1	=file(”emp_index.ctx“).create()
2	=A1().icursor(;DEPT=="HR").fetch()

计算举例

高性能键值查询（SPL脚本）

组表stock存储每日股价，主键为stockCode、tradeDate。请查询出stockCode=601且tradeDate=2018-01-01的记录

```
=file("stock.ctx").create().find([601,date("2018-01-01")])
```

非键字段索引查询（SPL脚本）

组表sales存储销售数据，主键为orderDate、region、orderID，并额外建立针对customer字段的索引（名为i_customer）。请查询出customer="GE"的记录

```
=file("sales.ctx").create().icursor(;customer=="GE", i_customer)
```

说明：索引查询时可省略索引名，集算器将自动使用合适的索引。

扩展阅读：通过索引与条件来过滤实表http://doc.raqsoft.com.cn/esproc/func/icursor2.html

计算举例

大主子表关联查询（SPL脚本）

order.ctx和orderDetail是主子关系，且都按orderID分段。请计算每个订单的金额。

order.ctx（主表）		
key	orderID	int
	custID	string
	orderDate	date

orderDetail.ctx（子表）		
key	orderID	int
key	productID	int
	price	float
	quantity	int

	A	B
1	=file（“order.ctx”）.create().cursor@m(orderID)	/并行游标
2	=file（“orderDetail.ctx”）.create().cursor@m(orderID,price,quantity;;A1)	/按主表分段并行
3	=joinx(A1:main,orderID; A2:detail,orderID)	/关联
4	=A3.groups(orderID;sum(detail.price*detail.quantity):total)	/分组汇总

扩展阅读：　大主子表关联的性能优化方法http://c.raqsoft.com.cn/article/1545619124373

SPL组表进一步优化 JOIN 性能http://c.raqsoft.com.cn/article/1545631312033

计算举例

源数据和目标组表的排序顺序不同时，对组表的计算（SPL脚本）

源数据报税单按时间生成，目标组表需按税单号查询，使用新（近一个月）、旧（历史）两个组表存储税单，前端计算时将两个组表视为整体进行查询。

	A	B
1	=file(["taxHis.ctx","taxMon.ctx"])	/创建文件组
2	=A1.create().cursor(;cardNo=="010319760818002X")	/查询

计算举例

T+0实时混算（SPL脚本）

数仓存储订单历史数据，生产系统存储实时订单数据。请统计目前为止每个客户的订单数。

	A	B
1	=connect（“db”）.query@x（“select customer,count(1) total from sales where orderDate=?”,now() ）	/数据库当日订单
2	=connect().query@x（“select customer,count(1) total from sales.ctx”）	/组表历史订单
3	=[A1,A2].conj().groups(customer,sum(total):total)	/二次分组汇总

扩展阅读：

上述例子说明了T+0实时混算的基本用法，关于更全面更细节的情况请参考分库后的统计查询

http://c.raqsoft.com.cn/article/1567657792653

查询语句方式

除了SPL语句之外，文件数仓还支持SQL语句、SQL+语句

语法	举例比较	特点
SQL语句	select * from emp.ctx where DEPT='HR'	用法简单，无法自由优化
SQL+语句	select * from emp.ctx where /*+index*/ DEPT='HR'	提供基于数据特征的标记语法，可进行高性能计算
SPL语句	=file("emp.ctx“).create().icursor(;DEPT=="HR").fetch()	可自由优化性能

说明：使用SPL语句和SQL语句，URL要写成jdbc:esproc:local://onlyServer=true。使用SQL+语句，url要写成jdbc:esproc:local://onlyServer=true&sqlfirst=plus。

扩展阅读：SQL+ http://doc.raqsoft.com.cn/esproc/func/sqljia.html

计算举例

过滤（SQL语句）

select ID,NAME,GENDER,AGE from students.ctx where GENDER='F' and AGE>24

排序（SQL语句）

select ID,NAME,GENDER,AGE from students.ctx order by AGE

分组汇总（SQL语句）

select GENDER,count(1) as total from students.ctx group by GENDER

提示：不限于组表，也可对简表btx、Excel、txt等数据格式进行计算

宽表OLAP分析

对数据库数仓的宽表进行OLAP分析时，一般使用JDBC/ODBC接口访问数据，使用SQL语句取数，文件型数仓亦同样。

支持文件数仓的OLAP工具：凡支持ODBC\JDBC数据源的OLAP工具

查询语法：SQL语句、SQL+语句、SPL语句（包括脚本文件）

特别地，润乾报表OLAP分析控件既支持数据库数仓，也支持文件型数仓，并直接支持Excel、TXT等文件数据源（其他OLAP工具需入库再使用）

宽表多维分析

以润乾报表为例，对宽表进行多维分析

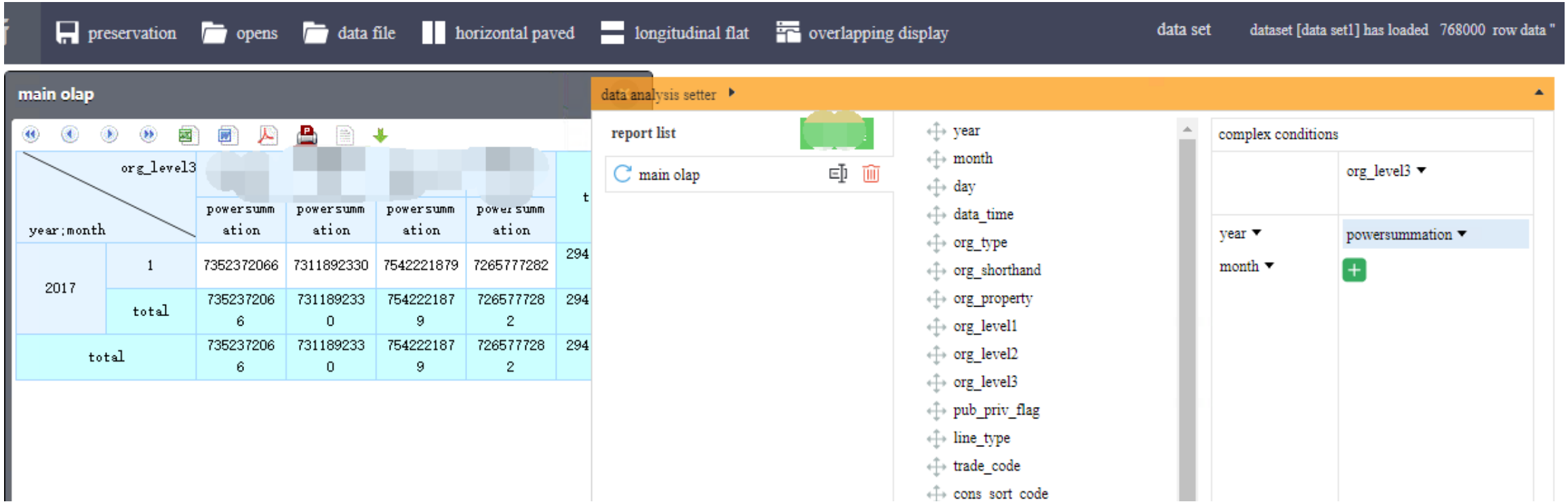
1：在JSP页面嵌入OLAP分析控件，输入SPL、SQL（如下面例子）、SQL+，或SPL脚本文件名

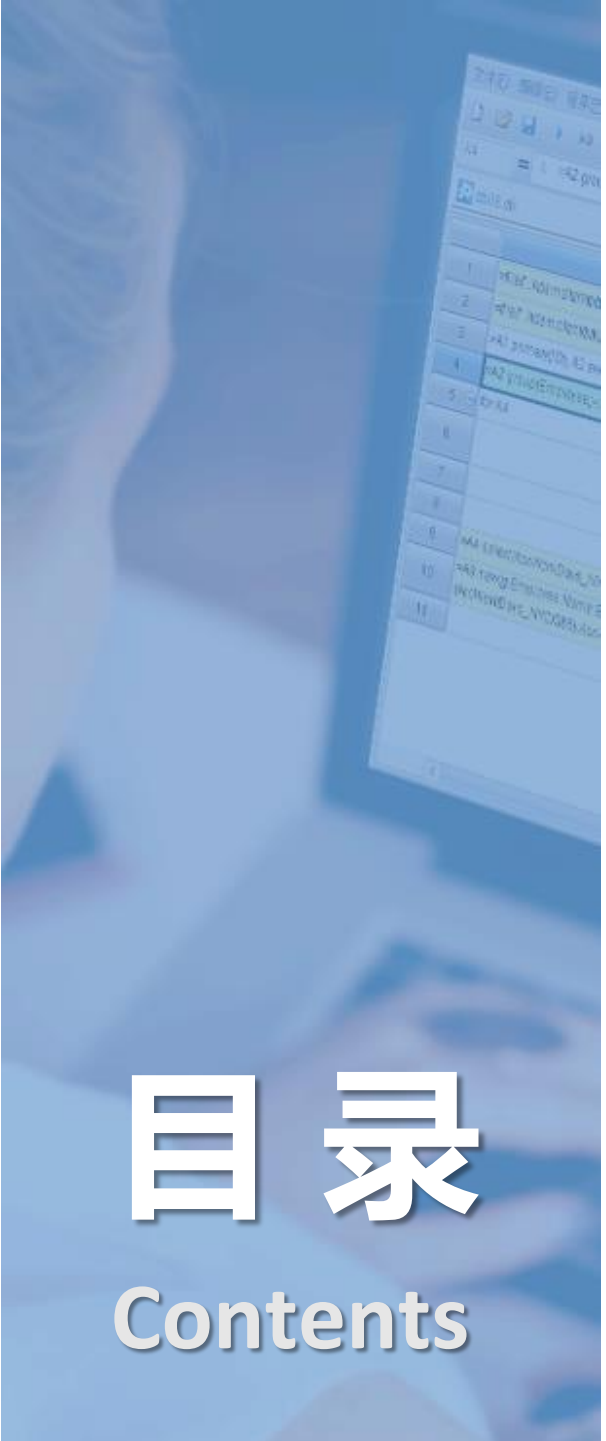
```
…..
<%@ taglib uri="/WEB-INF/raqsoftAnalyse.tld" prefix="raqsoft" %>
<raqsoft:analysev2
                dataSource= "filewh"
                ql="selct * from sales.ctx"
/>
…..
```

更多OLAP控件用法，参见http://doc.raqsoft.com.cn/report/dql/

宽表OLAP分析

2.执行JSP，进行OLAP分析，形如：



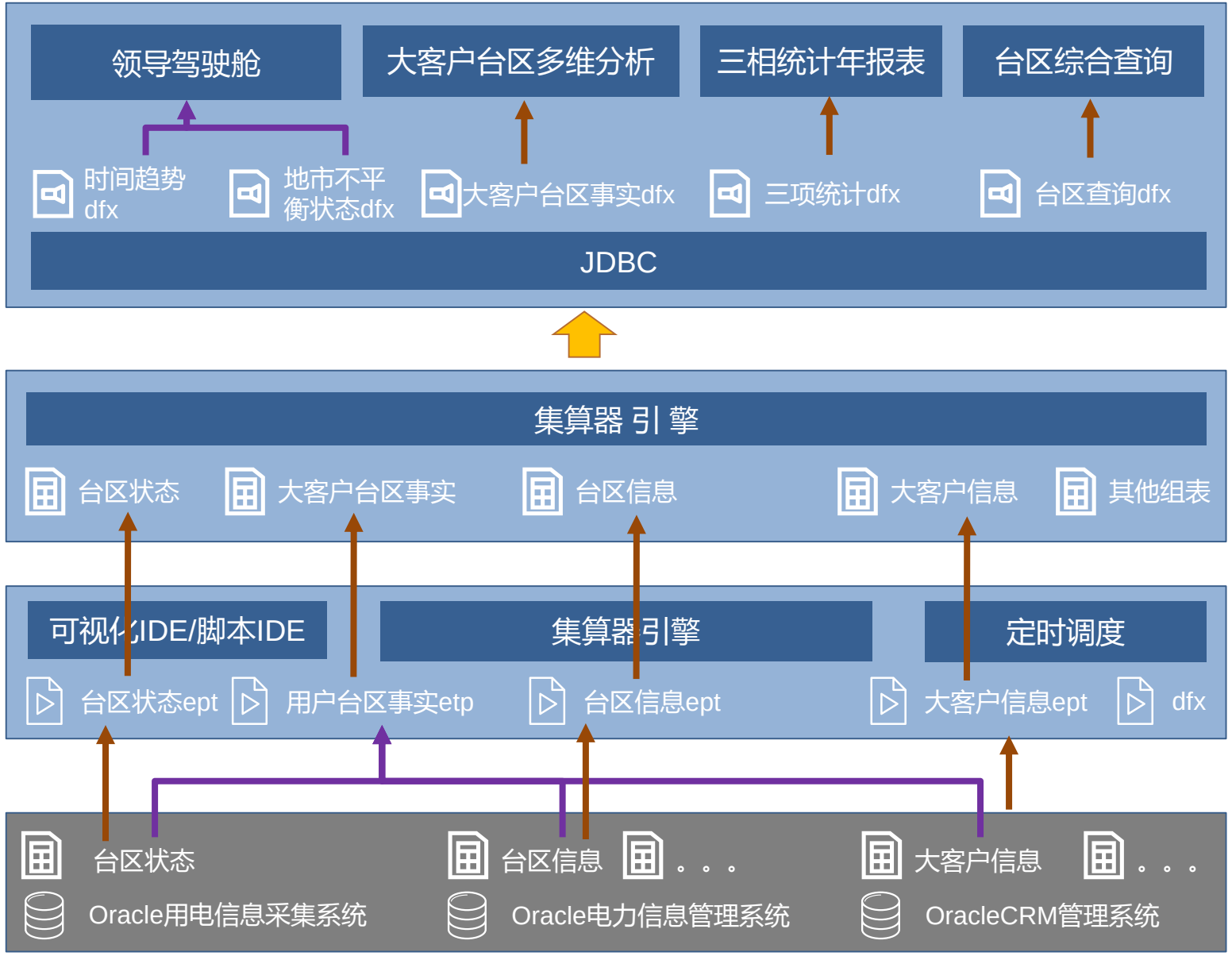


目 录

Contents

电力台区数据仓库

以电力行业的三个业务系统为源数据，建立变压器为主题的数据仓库，向上提供Dashboard、报表查询、多维分析等前端应用



存储设计

源数据

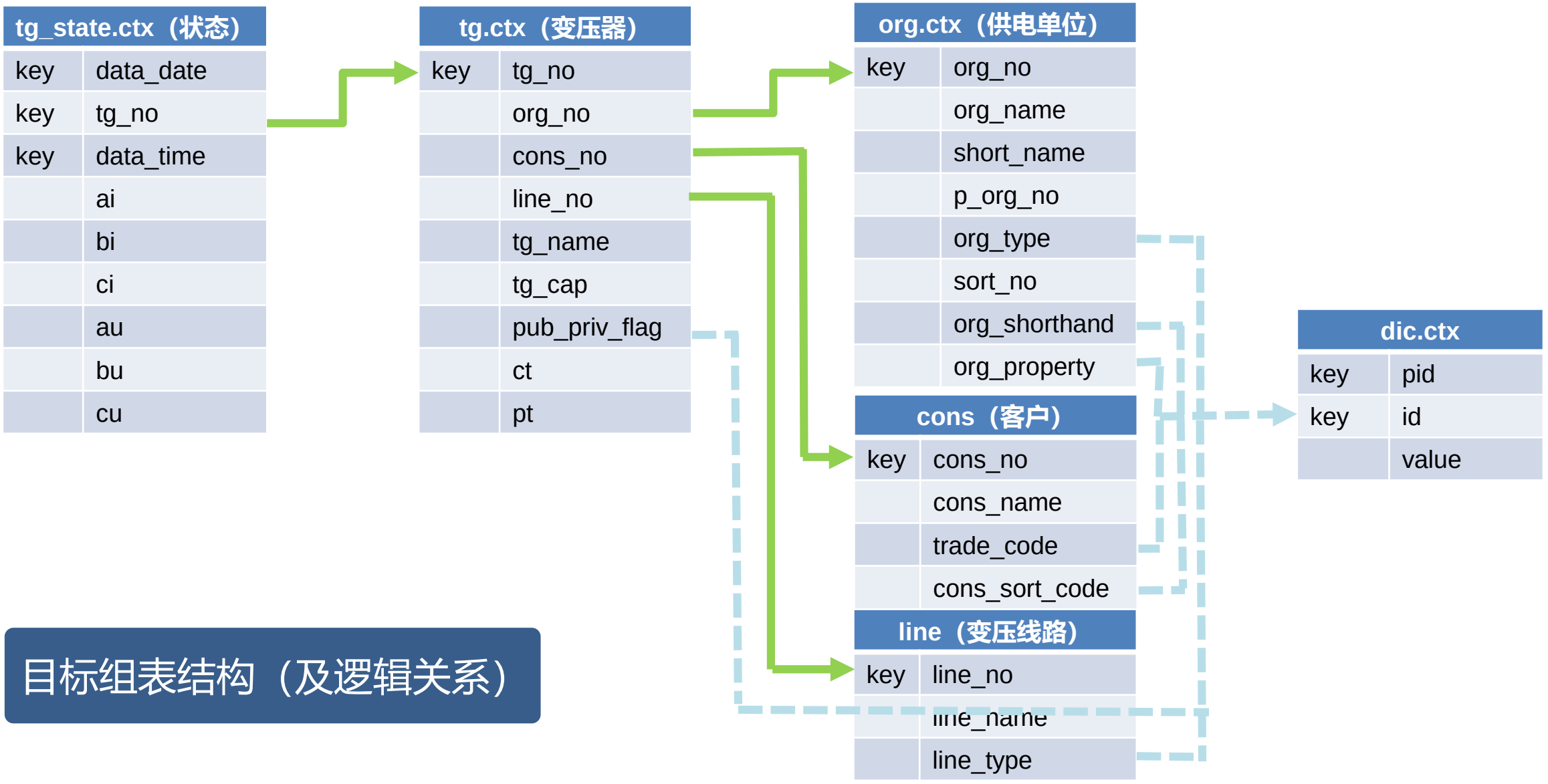
业务库	库表	类型	更新/追加频率
用电信息采集系统	台区状态	事实表	每15分钟追加
电力信息管理系统	台区信息	维表	相对固定
	供电单位	维表	-
	线路信息	维表	-
	字典表	维表	-
CRM管理系统	大客户信息	维表	相对固定

存储设计

目标组表与源数据关系

组表	类型	源数据表
台区状态	事实表	用电信息采集系统.台区状态
台区信息	维表	电力信息管理系统.台区信息
供电单位	维表	电力信息管理系统.供电单位
线路信息	维表	电力信息管理系统.线路信息
字典表	维表	电力信息管理系统.字典表
大客户信息	维表	CRM管理系统.大客户信息
大客户台区事实表	事实表	用电信息采集系统.台区状态 电力信息管理系统.台区信息 电力信息管理系统.供电单位 CRM管理系统.大客户信息 电力信息管理系统.线路信息

存储设计


tg_state.ctx（状态）
key data_date
key tg_no
key data_time
ai
bi
ci
au
bu
cu
tg.ctx（变压器）
key tg_no
org_no
cons_no
line_no
tg_name
tg_cap
pub_priv_flag
ct
pt
org.ctx（供电单位）
key org_no
org_name
short_name
p_org_no
org_type
sort_no
org_shorthand
org_property
cons（客户）
key cons_no
cons_name
trade_code
cons_sort_code
line（变压线路）
key line_no
line_name
line_type
dic.ctx
key pid
key id
value
目标组表结构（及逻辑关系）

存储设计

目标组表结构：

大客户台区事实表（OLAP宽表）

cube.ctx	
key	year
key	month
key	day
key	data_time
key	org_type
key	org_shorthand
key	org_property
key	org_level1
key	org_level2
key	org_leve3
key	pub_priv_flag
key	line_type
key	trade_code
key	cons_sort_code
key	tg_no
	ai
	bi

	ci
	au
	bu
	cu
	du
	power

准备整理

供电单位表

处理方法：全量抽取

说明：供电单位是小维表，更新频率很低，更新后可全量抽取。

注意：取数时需按主键排序。

部分代码如下：

B	C	D	E
=connect("DLHIVE")			
=B2.query@x("select org_no , org_name , short_name , p_org_no ,org_type ,sort_no ,org_shorthand , org_property from org order by org_no")			
=fileName="testData" /A2/ "/SC_SB_YDXX_ORG.ctx"			
=file(fileName).create@y(#org_no,org_name,short_name,p_org_no,org_type,sort_no,org_s			
=B5.append@i(B3.sort(org_no).cursor())			

准备整理

台区状态表

处理方法：增量抽取

说明：台区状态是大事实表，需每日增量更新。

注意：需按日期去增量，需按主键排序。

部分代码如右图：

B	C	D	E
/begin can not be null	if begin==null ||	=output("need	end
=connect("DLHIVE")			
=B2.cursor@x("select data_date ,tg_no ,data_time ,ai ,bi ,ci ,au ,bu ,cu from tg_state where mod(tg_no+1,2)=? and data_date>=? and data_date<? order by data_date,tg_no,data_time ",A2-1,string(begin),string(end))			
=file("testData" /A2/ "/SC_SB_YDXX_TG_STATE.ctx")			
=dfile=B4.create@y(#data_date,#tg_no,#data_time ,ai ,bi ,ci ,au ,bu ,cu;data_date)			
=dfile.append(B3.run(data_date=date(data_date),data_time=time(data_time)))			

说明：大客户台区事实表属于多源复杂宽表，见前面例子。其他组表不再赘述。

查询计算

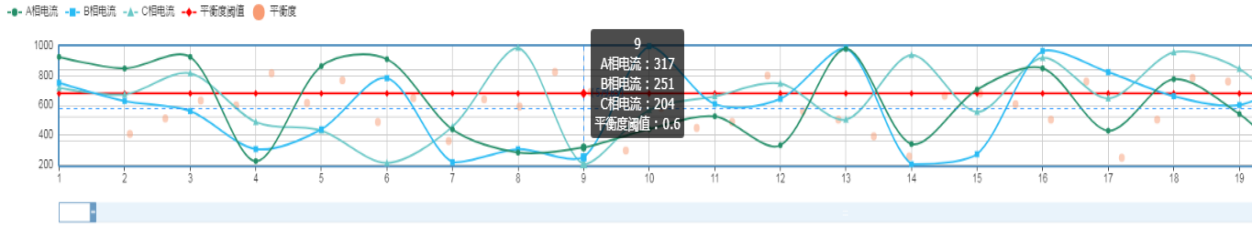
台区综合查询

以各维度为参数，查询一批台区某日的三相统计数据；以台区编号为参数，钻取查询某台区的详情

起始时间 2019-07-03 终止时间 2019-07-10 城市 全部 区县 全部 所站 全部 查询
线路编码 线路名称 台区编码 台区名称 客户名称 全部
不平衡状态 全部 三相不平衡 平衡

城市	区县	所站	线路编号	线路名称	台区编号	台区名称	日期	不平衡状态	电流曲线
唐山供电公司	脱敏11109供电单位名称	脱敏1110901供电单位名称	4	脱敏4线路名称	5	脱敏5线路名称	2017-01-01	三相不平衡	查看
唐山供电公司	脱敏11109供电单位名称	脱敏1110901供电单位名称	4	脱敏4线路名称	5	脱敏5线路名称	2017-01-04	三相不平衡	查看
唐山供电公司	脱敏11109供电单位名称	脱敏1110901供电单位名称	4	脱敏4线路名称	5	脱敏5线路名称	2017-01-06	三相不平衡	查看
唐山供电公司	脱敏11109供电单位名称	脱敏1110901供电单位名称	4	脱敏4线路名称	5	脱敏5线路名称	2017-01-08	三相不平衡	查看
唐山供电公司	脱敏11109供电单位名称	脱敏1110901供电单位名称	4	脱敏4线路名称	5	脱敏5线路名称	2017-01-10	三相不平衡	查看
唐山供电公司	脱敏11109供电单位名称	脱敏1110901供电单位名称	4	脱敏4线路名称	5	脱敏5线路名称	2017-01-09	三相不平衡	查看
唐山供电公司	脱敏11109供电单位名称	脱敏1110901供电单位名称	4	脱敏4线路名称	5	脱敏5线路名称	2017-01-07	三相不平衡	查看
唐山供电公									
唐山供电公									
张家口供电									
张家口供电									

时序点	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
A相电流	925	848	925	225	863	910	438	283	317	441	526	332	979	340	706	849	429	776	542	203	585	350	675	252	377	992	419	905	286	989	577	470	719	41
B相电流	751	629	561	306	439	783	218	305	251	996	609	644	985	204	272	965	823	661	602	809	966	392	862	254	449	713	612	652	622	950	925	246	475	64
C相电流	717	673	815	488	430	213	457	987	204	568	660	746	504	937	556	919	647	956	845	470	527	244	886	236	904	394	298	560	939	934	906	910	215	86
平衡度	0.22	0.26	0.39	0.54	0.50	0.77	0.52	0.71	0.36	0.56	0.20	0.55	0.49	0.78	0.61	0.12	0.48	0.31	0.36	0.75	0.45	0.38	0.24	0.07	0.58	0.60	0.51	0.38	0.70	0.06	0.38	0.73	0.70	0.5
平衡度阈值	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6



	A	B	C	D
12	=A11.cursor@z(cons_no,cons_name).fetch().keys(	//cons表取数		
13				
14	=file@0("SC_SB_YDXX_LINE.ctx", A1).create()			
15	=A14.cursor@z(line_no,line_name).fetch().keys(line	//line表取数		
16				
17	=file@0z("SC_SB_YDXX_TG_STATE.ctx",A1).create()			
18	=A17.cursor@m(data_date,tg_no,data_time,ai,bi,ci;d	//按照时间过滤数据	//cursor@m中的线程数应根据CPU物理核心	
19	=A18.derive((maxi=max(ai,ci,bi),mini=min(ai,bi,ci),平衡度=(maxi-mini)/maxi,if		//添加时点状态	
20	=A19.groups@o(data_date,tg_no;if(max(连续不平	//计算当日状态		
21	=if(string(phzt)=="" || phzt==null,"1==1","当日平衡	=A20.select(${A21})	////按照平衡状态过滤数据，0不平衡1平衡	
22	=B21.sort(data_date,tg_no)			
23	=file@0("SC_SB_YDXX_TG.ctx", A1).create()			
24	=A23.cursor@0z(tg_no,org_no,cons_no,line_no,tg_	//tg表取数		
25	=join(A22:aa,tg_no;A24:bb,tg_no)			
26	=A25.new(aa.data_date:data_date,aa.当日平			
27	>A26.switch(org_no,A9:zhan_no).switch(cons_no,A12:cons_no).switch(line_no,A15:line_no)			
28	=A26.new(data_date,drphzt,tg_no,tg_name,cons_no	cons_no:cons_no,cons_no.cons_name:cons_name,line_no.line_no:line		
29	=if(string(xlbh)=="" ||	=if(xlmc=="" ||	=if(tqbh=="" || tqbh==null," && 1==1"," &&	
30	=if(tqmc=="" || tqmc==null," && 1==1"," &&	=if(khbh=="" || khbh==null," && 1==1"," && cons_no==khbh")		
31	=A29+B29+C29+A30+B30			

查询计算

台区驾驶舱

计算关键指标，为统计图准备准据，体现本月台区重要和宏观的状态

	A	B
15	=A14.sort(data_date,tg_no)	=now()
16	=file@0("SC_SB_YDXX_TG.ctx", A1).create()	
17	=A16.cursor@0z(tg_no,org_no,cons_no,line_no,tg_name).fetch()	//tg表取数
18	=join(A15:aa,tg_no;A17:bb,tg_no)	
19	=A18.new(aa.data_date:data_date,aa.当日平衡:drphzt,bb.tg_no:tg_no,bb,bb.org_no:org_no)	
20	=A19.switch(org_no,A9:zhan_no)	
21	=A20.new(data_date,drphzt,tg_no,org_no.city_no:city_no,org_no.city_name:city_name)	
22	=A21.sort(city_name).groups(city_name,data_date;count(drphzt==0):bph,count(drphzt==1):ph)	
23	=A21.select(data_date>=elapse@m(rq,-3) && data_date<=rq)	//近三个月数据
24	=A23.sort(city_name).groups(city_name;count(drphzt==0):bphs,count(drphzt=	//近三个月平衡数据
25		

方案比较

原系统使用Oracle数仓，因客户体验和维护性等原因，现改用集算器文件数仓

下面以算法“某月份每天每个台区的平衡状态（组内计算）”为例，对比两套方案的指标差异

硬件环境

CPU：8核，Intel i7

内存：16G

硬盘：1TB

网络：千兆

软件环境

操作系统：centOS 7.4

JDK：version 1.8, 64bit

ORACLE:11.2g

集算器:version 2018

数据环境

总记录条数：463,484,582

库表体积：42.4G

组表体积：9.2G

对比指标	Oracle数仓	文件数仓
执行时间	85,413ms	8,318ms
代码开发	难以调试	易于调试
应用耦合	高	低
维护性	较差	较好
客户体验	较差	较好

方案比较

算法对比

原方案：Oracle数仓

```
select data_date,tg_no, case when max(连续不平衡)>=3
then 0 else 1 end 当日平衡 from(
  select data_date,tg_no,count(平衡累计)-1 连续不平衡 from(
        select  data_date,tg_no,sum(时点不平衡) over
(partition  by data_date,tg_no order by
data_date,tg_no,data_time)平衡累计 from(
                select  data_date,tg_no,data_time ,case when
(greatest(ai,bi,ci)-least(ai,bi,ci))/greatest(ai,bi,ci)>0.6 then 0
else 1 end 时点不平衡
                        from tg_state   where data_date>=’ 2017-
01-01’   and data_date<’ 2017-02-01’
                        order by data_date,tg_no,data_time
                )
        )
  group by data_date,tg_no,平衡累计
)
group by data_date,tg_no order by data_date,tg_no
```

现方案：文件数仓

	A
1	=file("nodes.txt").read@n()
2	=file@0z("SC_SB_YDXX_TG_STATE.ctx",A1).create()
3	
4	=A2.cursor@m(data_date,tg_no,data_time,ai,bi,ci;data_date>=begin && data_date<=end;4)
5	=A4.derive((maxi=max(ai,ci,bi),mini=min(ai,bi,ci),平衡度=(maxi-mini)/maxi,if(平衡度>0.6,false,true)):时点平衡,t=if(data_date==data_date[-1] && tg_no==tg_no[-1] && 时点平衡[-1]==false,t+1,0):连续不平衡)
6	=A5.groups@o(data_date,tg_no;if(max(连续不平衡)>=3,0,1):当日平衡)
7	=A6.sort(data_date,tg_no)

想要了解更多 请联系我们

更多技术内容请移步 乾学院

http://c.raqsoft.com.cn/login

优惠价购买请加入好多乾

http://sys.misdiy.com/hdq.html

THANK YOU!