永洪产品使用最佳实践（上）

引言

YonghongTech
Talk with Data

- ？ 报表做完了需要修改需求，咦，组件在哪呢，为什么找不到了呢？
- ？ 为什么我的报表做完了，每天打开的时候这么慢呢？
- ？ 我什么都没做呀，为什么突然报表就打不开了？
- ？ 服务器的内存占用这么多，磁盘突然间就满了，为什么啊？
- ？ 报表不能这么慢呀，领导不乐意了，怎么加快报表打开的速度呢？

为什么？ 为什么？ 为什么？ 为什么？ 为什么？ 为什么？ 为什么？ 为什么？ 为什么？ 为什么？

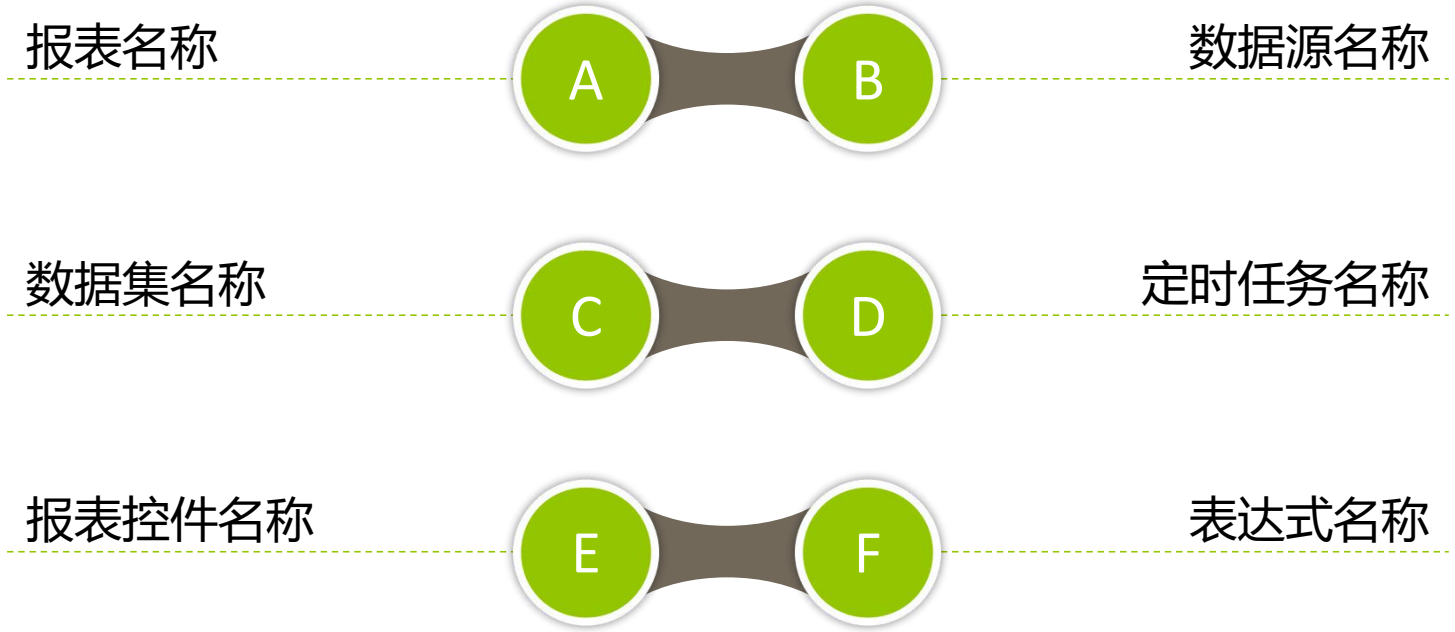
目录

YonghongTech

Talk with Data

命名最佳实践

命名最佳实践



命名最佳实践

YonghongTech
Talk with Data

1) 报表名称

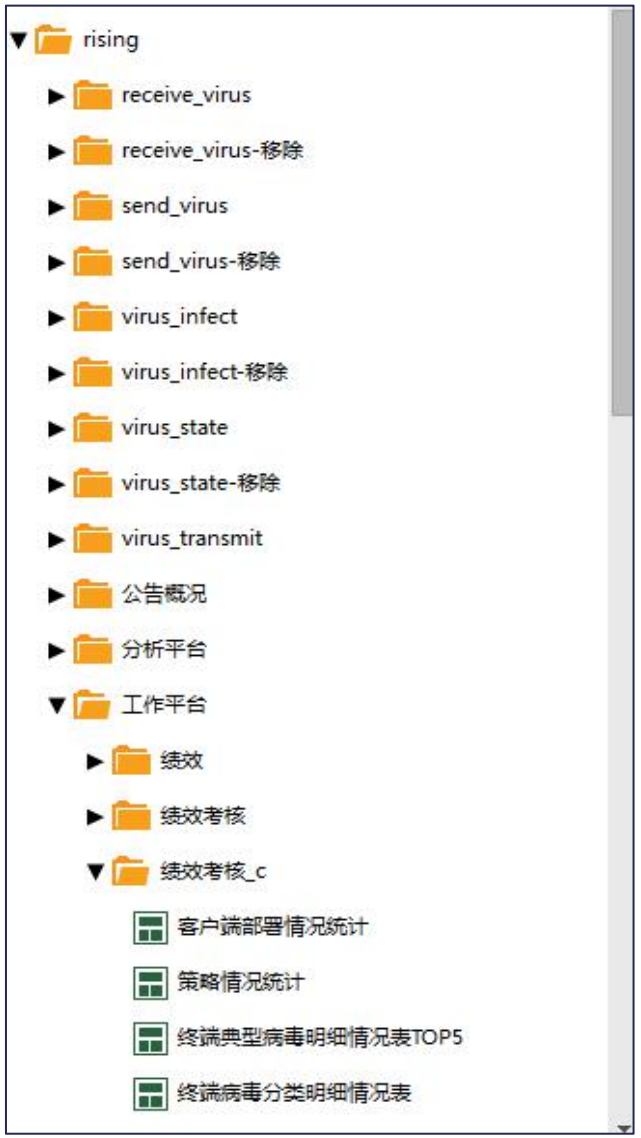
- 按照主题、子主题、报表名称的层级去设定报表分布。
- 若存在历史版本报表，要单独创建文件夹和正式版本区分开。
- 报表文件夹的层级不能超过**8层**。

命名最佳实践

较好的案例

VS

较差的案例



命名最佳实践

YonghongTech
Talk with Data

2) 数据源名称

- 数据源名称要**准确**，避免产生误导。
- 如果一个类型的数据源有多个ip，建议在名称后添加ip。

例如：mysql139；mysql137。

命名最佳实践

YonghongTech
Talk with Data

3) 数据集名称

- 数据集命名的原则同报表一致，按照主题、子主题、数据集名称的层级去设定分布，名称和报表互相对应。
- 如果有集市，创建一个sql文件夹和一个集市文件夹分别存储。

命名最佳实践

YonghongTech
Talk with Data

4）定时任务名称

- 建议任务文件夹名称与数据集文件夹一致；任务名称与数据集一致。

命名最佳实践

YonghongTech
Talk with Data

5）报表控件名称

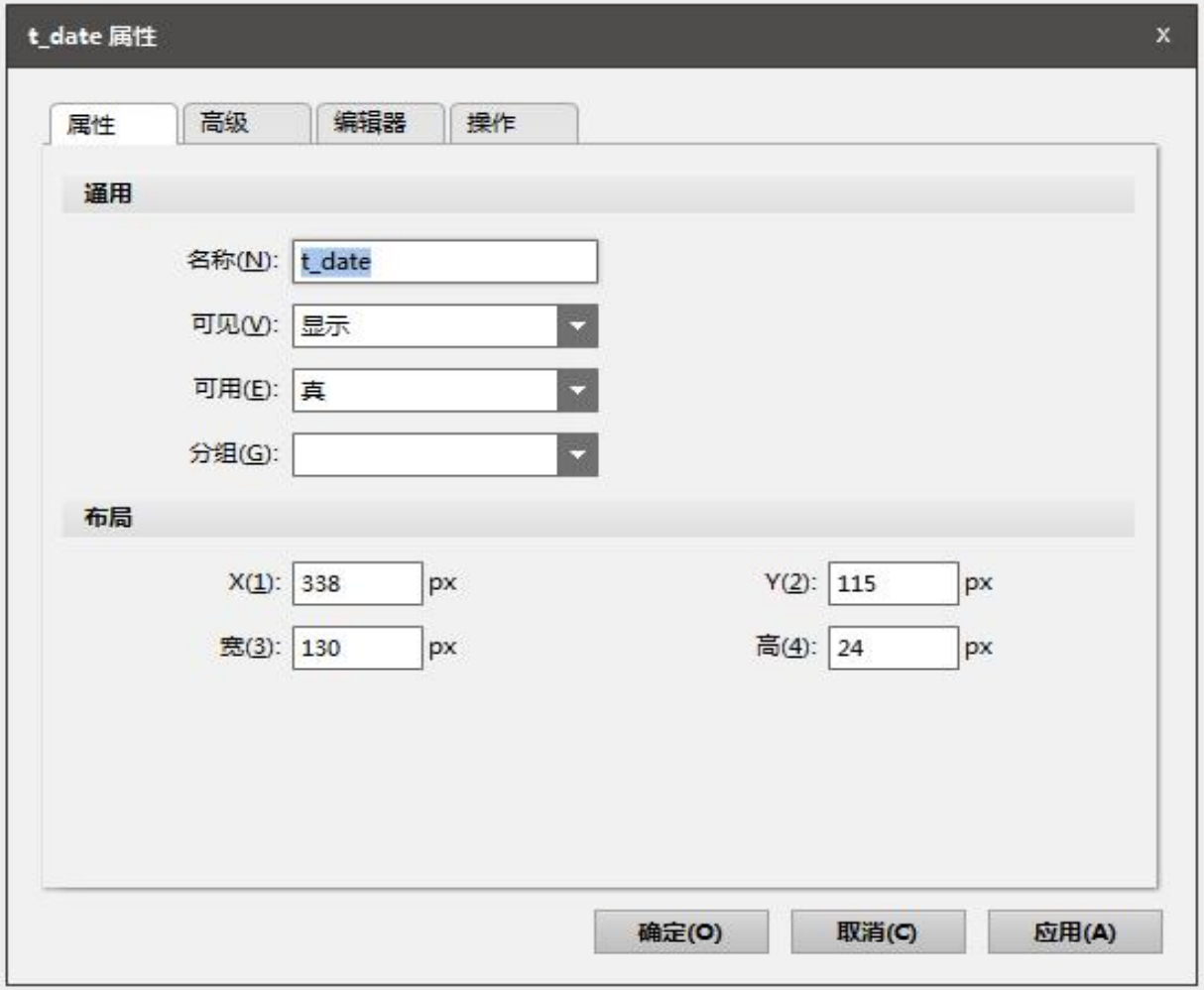
- 控件要重命名，名称要有意义。

  例如：文本输入框-t_date；table1-客户端部署情况统计。尤其是页面控件很多，又需要写脚本的时候，不命名的话，过滤器和脚本会看不明白。

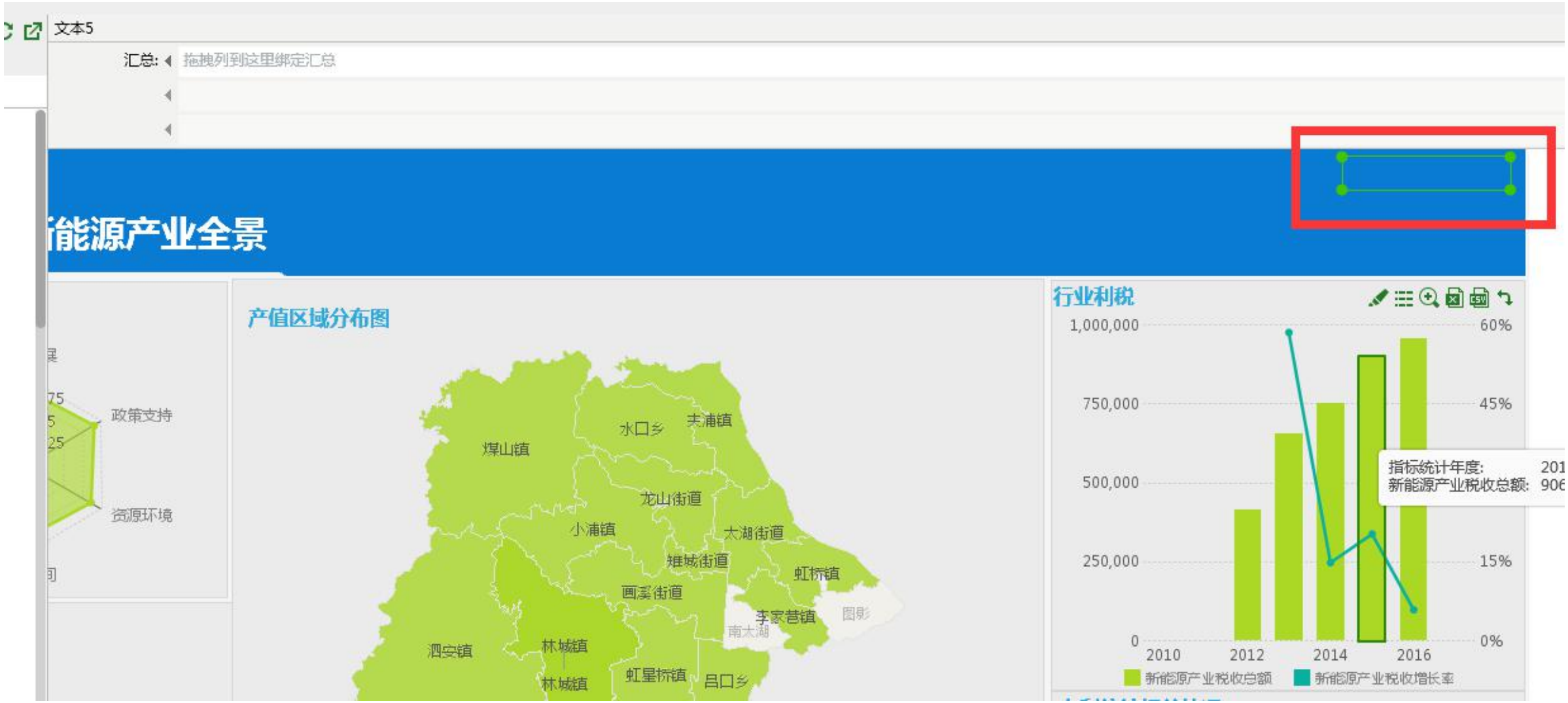
- 隐藏控件的名称以hidden_开头，位置统一放到右上角，采用设置背景色、字体颜色、透明、去边框、设置边框颜色的方式隐藏，**置于顶层**，不要被已有控件遮挡。

命名最佳实践


防病毒安全管理系统监控情况通报
2017年6月1日至2017年6月28日全省业务专网
开始时间：
2017-06-01
结束时间：
2017-06-28
一、防病毒管理系统 应用总体情况
1）客户端部署情况统计
通过部署的瑞星防病毒软件系统查询发现全省软件部署率平均为: 100.00%。
序号 单位名称 总终端数 已安装数 杀毒软件安装率
1 省局 382 382 100%
2 南京 2253 2253 100%
3 无锡 1907 1907 100%
4 徐州 1710 1710 100%
5 常州 1568 1568 100%
6 苏州 2953 2953 100%
7 南通 1772 1772 100%
8 连云港 1011 1011 100%
9 淮安 1051 1051 100%
10 盐城 1957 1957 100%
11 扬州 1018 1018 100%
12 镇江 985 985 100%
13 泰州 1331 1331 100%


t_date 属性
x
属性
高级
编辑器
操作
通用
名称(N): t_date
可见(V): 显示
可用(E): 真
分组(G):
布局
X(1): 338 px
Y(2): 115 px
宽(3): 130 px
高(4): 24 px
确定(O)
取消(C)
应用(A)

命名最佳实践


文本5
汇总:
拖拽列到这里绑定汇总
能源产业全景
政策支持
资源环境
产值区域分布图
煤山镇
水口乡
夫浦镇
龙山街道
小浦镇
太湖街道
雉城街道
虹桥镇
画溪街道
李家巷镇
南太湖
泗安镇
林城镇
林城镇
虹星桥镇
吕口乡
行业利税
1,000,000
750,000
500,000
250,000
0
60%
45%
15%
0%
2010
2012
2014
2016
指标统计年度:
新能源产业税收总额:
新能源产业税收总额
新能源产业税收增长率

命名最佳实践

YonghongTech
Talk with Data

6) 表达式名称

- 表达式要体现数据的特征，避免**过长过短**。
- 区分维度表达式、细节表达式以及聚合表达式，避免创建混乱，一般情况下，字符串类型的数据创建维度表达式，数值类型的数据创建细节表达式，需要对某个数据段中的数据进行汇总时，创建聚合表达式。

(7.5.2版本自带帮助手册，**脚本-计算器脚本**)

YonghongTech

Talk with Data

性能最佳实践

性能最佳实践

YonghongTech
Talk with Data

01

数据集

02

制作报告

03

定时任务

04

集市使用

05

脚本和表达式的使用

性能最佳实践

YonghongTech
Talk with Data

1. 数据集

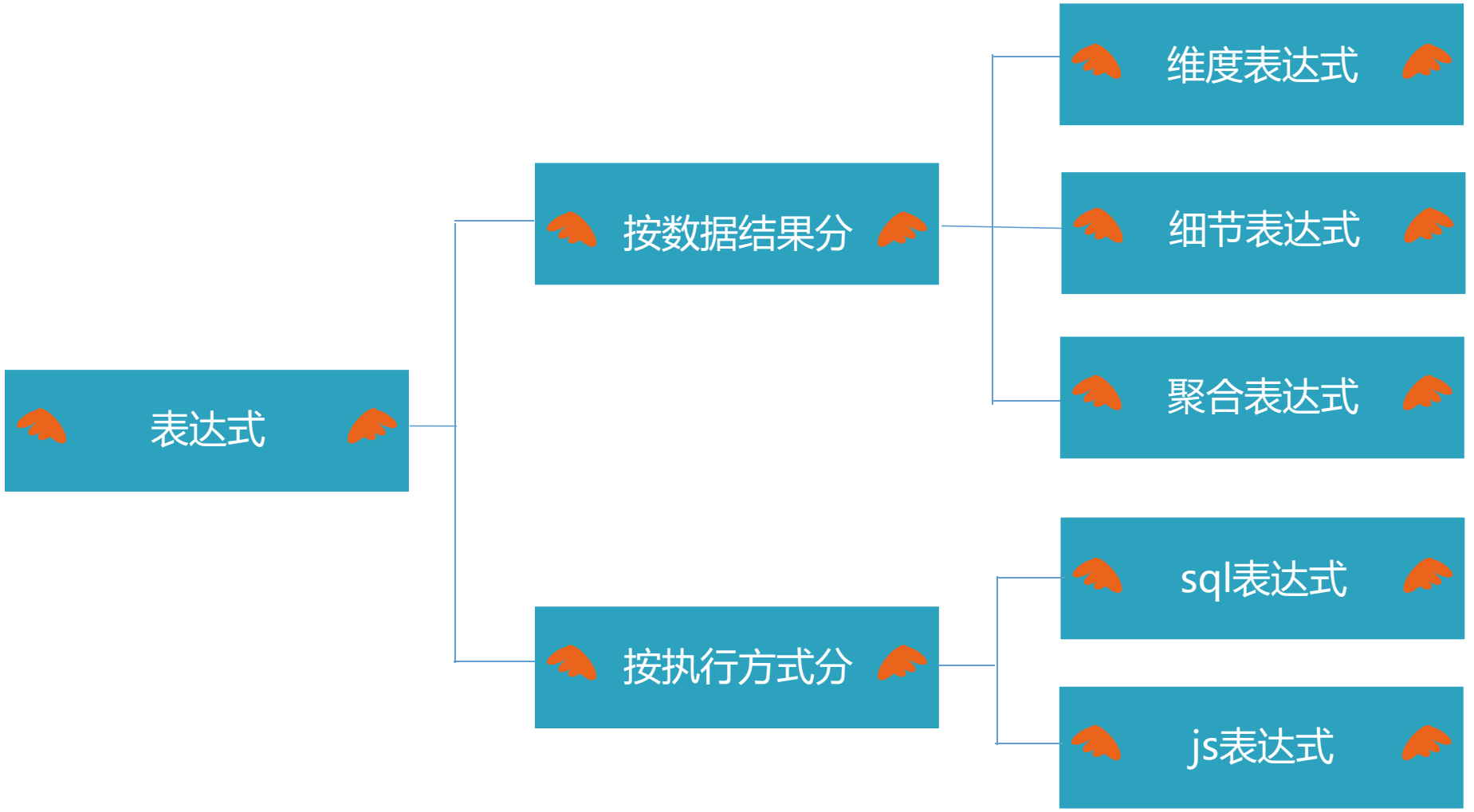
1）能在**创建数据集**处理的字段，尽量不要在**报表端**处理。

风险：在报表中创建的表达式无法给其他的报表共用，造成**重复性工作**。

2）能使用**sql函数**处理的字段，尽量不要用**js语法**处理。

风险：永洪支持库内计算，使用sql函数（sql表达式），可以将计算逻辑下推到数据库执行，而使用js语法（js表达式）处理的字段，会将数据抓到内存后，再进行计算，**占用大量内存**的同时，也有可能会在服务器上生成**大量临时文件**。

性能最佳实践



性能最佳实践

YonghongTech
Talk with Data

3）**sql尽量优化**，效率整体提升。

pushsql规则.xlsx

常见sql优化方式：

union代替join

尽量使用select 字段名1，字段名2，... from 表名，避免select *

避免用 in 或 not in，在子查询中，用 exists 代替 in ……

影响：直连数据库的报表，依赖sql在数据库的执行效率，*as soon as possible ,*

sql的效率越高，报表打开的越快！

性能最佳实践

YonghongTech
Talk with Data

4）可以转化为sql的查询，尽量**不使用组合查询。**

风险：多层嵌套的组合查询，难免会遇到无法下推到数据库的情况，层层计算，会整体**拖慢报表的性能**。

5）尽量**避免上万条数据以上的跨数据源组合查询。**

风险：跨数据源的组合查询，会将两个数据源的数据都抓到内存中，然后再进行计算，**占用大量内存**的同时，也有可能会在服务器上生成**大量临时文件**。

2. 报表制作

1）通过业务分离、逻辑分层，分为多页面显示，通过超链接进行逐层展示，同时尽量保障一报表**刚好一屏**，能少用一个组件尽量不多用。

风险：有的不同业务分析内容，放在一个页面，或为了达到美化效果，单页面组件过多，导致**页面展示缓慢，问题排查困难**。

性能最佳实践

YonghongTech
Talk with Data

2）尽量避免大量**明细数据**导出（超过1W条）的场景，明细数据导出往往需要线下二次加工，可以把这部分需求整理好放到BI上。

风险：导出明细数据，占用**大量的内存**。举个例子，假设给永洪分配了20G内存，存1T的数据。那么在每次用户访问报表的时候，会把报表涉及的数据加载到C节点的内存里，假设现在有一张表，涉及到1个G的明细数据。那么10个用户同时导出这个数据，最少也有10G，考虑到并发，这个占用的内存还会更大。

性能最佳实践

YonghongTech
Talk with Data

3）细节数据展现，在页面上加参数，根据**参数过滤**后，再进行**少量数据**展现。

风险：大量的明细数据从数据库执行出来后，通过网络带宽传输到永洪，报表的打开速度，基本上取决于sql在数据库执行时间以及网络对数据的传送速度，影响报表的打开速度，慢且占用内存，大量的内存占用会影响永洪的整体性能，并可能引发永洪**宕机**的风险。

性能最佳实践

4）有多个筛选条件时，尽量新增一个按钮做**批量提交**。在页面属性中勾选批量提交，可以减少报表前端向后台发请求的次数，**提升性能**。

风险：批量提交的作用是，在有多个选项的时候，只有当点击提交的时候，才会执行数据的过滤。当没有设置批量提交的时候，每改变一个选择条件，都会发出一条sql申请到数据库，当数据量较大的时候，会**占用线程时间较长**，那么后面再发出的申请只能等待线程释放。最终的后果，可能造成其他的用户在看报表的时候，**一直加载，无法显示报表内容**。

性能最佳实践

5）**筛选控件较多**时，在页面属性中去掉“过滤组件之间是否关联”。

风险：过滤组件之间是否关联，默认是关联的，当多个筛选组件的时候，勾选其中一个，其他的组件也会跟着进行数据的过滤，这个时候就会执行sql重新抓取数据，在报表页面直接感受到的是**报表卡顿，无法再次操作。**

6）尽量避免**大量的**使用文本框、仪表组件。

风险：一个组件相当于一个查询，过多查询影响**报表打开的速度**。

小结

尽量避免同时发出过多的请求给数据库，避免请求排队。

最佳实践主张优先库内计算，后内存计算，合理使用内存。

对于sql的要求，越快越好，越优越好！

下期预告

YonghongTech
Talk with Data

A 定时任务最佳实践

B 数据集市最佳实践

C 脚本表达式最佳实践

D 组件使用最佳实践

E 认证授权最佳实践

THANKS

永洪产品使用最佳实践（下）

目录

YonghongTech

Talk with Data

性能最佳实践

性能最佳实践

YonghongTech
Talk with Data

01 数据集

02 制作报告

03 **定时任务**

04 **集市使用**

05 **脚本和表达式的使用**

http://192.168.1.137:8010/bi/viewer

3. 定时任务

1）所有任务的执行时间尽量安排在**闲时**，并且把**时间分散开来**，避免同一时间执行多个任务，以免在忙时影响用户使用。如果在忙时导数，可以采用**单独节点导数**，和用户访问区分开。

风险：导数的定时任务，非常耗用资源，占用线程以及内存，如果在忙时导数，用户所发出的请求会**出现排队**的情况，内存不足也会导致**数据处理速度慢**，直接导致**永洪访问慢**，甚至是**宕机**。

性能最佳实践

YonghongTech
Talk with Data

2）**增量导入数据**和**同步数据集**的区别和适用场景

	同步数据集	增量导入数据
区别	a. 全量入集市，数据更新周期长 b. 创建数据集位置可操作 c. 无需新建集市数据集即可使用	a. 可传参，勾选追加增量导入数据 b. 可设置维度表，用于分布式join c. 可分割列入集市 d. 需新建集市数据集引用集市数据
场景	a. 没有支持增量导入的传参字段 b. 数据全量更新，历史数据在变化	a. 数据量大，历史数据不变化 b. 实时性要求不高 c. 数据中存在可识别增量部分的字段，如日期

4. 集市使用

1）入集市时，尽量避免随意将所有业务数据抽取到集市，建议先**确定要分析的内容**，然后**选取**数据分析过程中，可能会用到的字段，需要分析的范围的数据抽取到数据集市中。

风险：数据量大，列数多，长字符串列，都会导致抽取到集市**时间增加**，浪费资源且消耗性能。每个数据文件存储100W行数据，**列数多或长字符串列**的时候，单个文件就会**大**；**行数多**的时候，文件**数量会增加**，读取数据的时间就会**增长**。

性能最佳实践

2）尽量避免**在数据集市中**进行千万级别记录数据集的**组合查询**。

3）如果必须做Join，在数据库里做Join，结果再入集市，而不是直接在集市里Join，如果是一个大表和多个小表的关联，可以使用**分布式Join**，也就是在入集市的时候，将小表的入集市任务勾选**维度表**。

风险：**内存计算**，耗用访问节点（C节点）大量内存，易导致**卡顿或宕机**。

【知识点】分布式Jion原理：在分布式系统中，当有星形数据（一个大表，若干个小表）需要 join 的时候，可以将小表的数据复制到每个 Map 节点，执行 Map Side Join, 而无须到 Reduce 节点进行连接操作，从而提升表连接的效率。

性能最佳实践

YonghongTech
Talk with Data

性能最佳实践

YonghongTech
Talk with Data

4）尽量避免入集市后在报表侧创建**细节/维度表达式，**尽量避免入集市后在数据集侧创建**表达式**。

风险：入集市后创建的表达式（除了报表侧创建的聚合表达式外），都会基于内存计算，集市将所有的数据文件都抓取到内存中，然后**耗用内存**进行计算。

5. 脚本表达式

1）表达式嵌套使用**最多不能超过两层**。

风险：表达式嵌套超过**两层**会提示错误，**字段不存在**。如，时间字段-->日期粒度拆分-->利用日期字段再新生成字段，会提示时间字段不存在。

2）尽量避免**逻辑复杂**的，一环扣一环的脚本使用。

风险：如果更改引用字段，或进行名称变更，**增加排查问题难度**。如，脚本从组件获取数据->将数据赋给参数->参数传递给表达式->表达式绑定到组件中。

性能最佳实践

3）脚本尽量写在**报表级别**上，没有特殊原因，尽量避免写在组件上。

风险：在报表制作过程中，经常会有隐藏某个组件的情况，如果组件上有脚本，那么一旦出现问题，将**增加问题的排查难度**。

【知识点】脚本的执行顺序（简要理解）：

- **报表打开时，定义在仪表盘上的装载时运行的脚本，是最先执行的，且只在报表第一次打开的时候执行。**
- **当仪表盘上有任何变化时，设定在变化时运行的脚本被执行。**
- **最后组件上的脚本被执行。**

性能最佳实践

YonghongTech
Talk with Data

4）报表完成后，如果带有脚本或者表达式，尽量**整理个文档**，注明用了什么脚本，原因，作用等，编写脚本时，尽量带有代码注释。

风险：在进行报表后期运维的时候，如果没有文档记录，制作理念，变更原因等情况，无法了解制作报表的逻辑，增加了运维的**难度**。

脚本

一、招商部看板涉及到的脚本：

（1）

页面空白处右击脚本，装载时运行：

```
var a=getViewData("文本 36")
var b=getViewData("文本 34")
if(a==null){文本 6.visible=true}else{文本 6.visible=false}
if(b==null){文本 42.visible=true}else{文本 42.visible=false}
```

写脚本原因：昨日签单量可能为空，数据库没有此数据，空着显得很候需要显示为 0.

```
var a=getViewData("文本 36")      //获取文本 36 的值（实际签单量）
if(a==null){文本 6.visible=true}else{文本 6.visible=false}   //两个文本一
```

定为 0，叠放在一起，根据实际情况显示。当实际签单量为空的时候文本）显示出来。

（2）

装载时运行：

```
param['shiji']=getViewData("文本 32")
```

原因：要制作出目标完成度。

脚本是获取文本 32（本月时间签单量）

然后，新建聚合表达式，名字完成度。（招商部_计划签单量此数据源后，右击新建聚合表达式来创建的，查看的话，右击这个聚合

YonghongTech

Talk with Data

其他最佳实践

性能最佳实践

YonghongTech
Talk with Data

1）组件的使用：门户、门户组件使用时，尽量使用永洪推荐的IE11以上，谷歌48以上，火狐浏览器。

2）认证授权：查看用户，开发用户，创建数据集用户，权限严格区分，只开放对应权限，避免误操作。

THANKS