# OLAP 项目培训

这次讲的应该算是数据的etl过程，以及olap内容。

## OLAP概念

On-Line Analytical Processing[联机分析处理](https://baike.baidu.com/item/%E8%81%94%E6%9C%BA%E4%BA%8B%E5%8A%A1%E5%A4%84%E7%90%86)。

OLAP可以理解为一种软件技术，它使分析人员能够迅速、一致、交互地从各个方面观察信息，以达到深入理解数据的目的。OLAP展现在用户面前的是一幅幅多维视图。维（Dimension）：是人们观察数据的特定角度，是考虑问题时的一类属性。

## 为什么要改造成olap

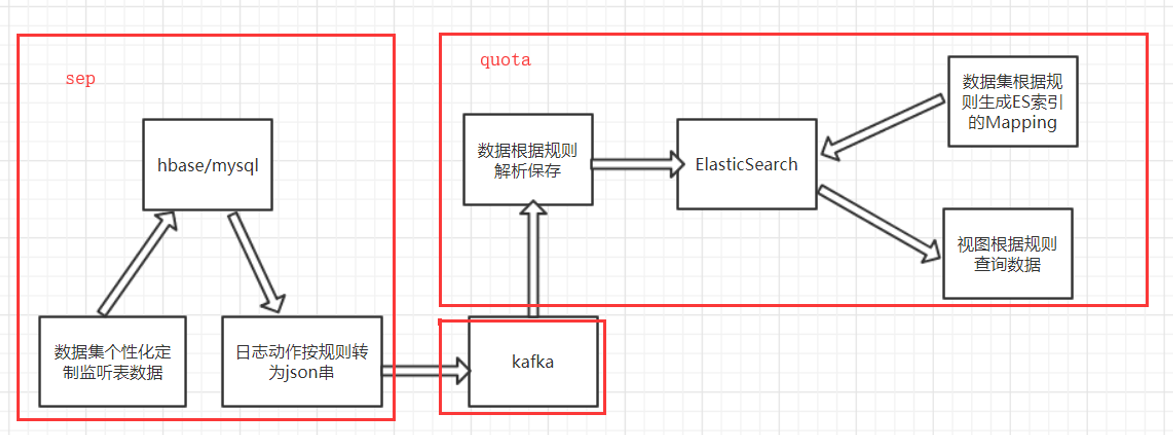
我们原有的项目，如医疗云，数据查询和数据统计时存在很多的不足或者复杂点，如下：

1. 数据获取过程链路长，需要从Hbase->Solr->ES->ES(二次指标)等环节，不容易排查数据问题。
2. 数据分析关注点过多，比如需要关注不同数据来源（Solr，ES，Mysql）。
3. Quartz执行顺序引起的数据不准确问题。
4. 指标多，维护复杂，相同维度指标无法一次性实现。
5. 指标分散，部分数据指标在不同报表（仪表盘）中数据展示不一致。
6. 数据无法实时性，需要同步后在执行指标，最后才能查询统计数据。

基于上面的不足，对比olap的优点有

1. 简化数据链路，将数据从Hbase同步到ES中。
2. 数据分析时只关注ES。
3. 通过多维数据视图和ES的聚合函数，部分指标可以一起实现，不需要多次维护。减少统计接口

## 框架流程

全量触发：完善  
模型 I 健康

### SEP是基于复制集处理来处理

SEP的主要功能是：监听HBase集群上发生的一切数据变化，然后对这些变化做出处理。这里说的数据变化是记录在WAL日志里面的，所以SEP要做的就是监听WAL日志变化。（备注：监听可以指定监听的表和列族和列标识符，即可以根据我们需求定制化监听方案），然后将监听到的数据，增删改等数据信息保存到kafka。

Sep是hBase-indexer核心功能，具体有兴趣的可以自己去看下。我这里基于他的基础上增加了监听后数据处理的功能。

附加扩展：Mysql 数据变化也可以通过日志监听起来。

### Kafka

消息的发布与订阅（发布监听到的数据）

quota

主要功能有

1、多维数据集创建。维度，及维度成员和对应库中表的字段配置

如：患者就诊档案信息数据集  
 维度：患者姓名，性别，年龄，就诊机构,就诊编号，就诊时间，科室，用药情况(有成员维度) 等。

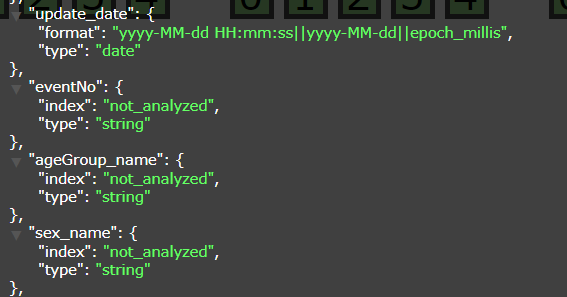
成员维度：用药情况（药名称，价格，型号，用法）

这些维度和成员维度都会由对应的数据源字段对应数据，如 ：患者姓名 由 hbase库中HealthProfile表的 patientName字段对应。

2、根据配置及规则生成多维数据集的ES的Mapping。

ES的mapping 就是创建type中字段和数据类型，字段分词等相关信息。

规则：多维数据集中配置的维度，作为type的一个属性，如果维度存在维度成员，即此时维度成员属于对象类型，需要区分object,nested（子集保存）两种类型。维度成员作为此维度的子属性。如果维度存在数据字典，会扩展出维度\_name属性。



测试：[G](http://laptop-u3dq0u6v:10330/swagger-ui.html" \l "!/cube45controller/getQuotaReportUsingGET)swagger --- 对应数据表相关配置简单介绍

（将数据库中的数据集重命名\_test，避免影响数据）

[/api/v1.0/cube/elasticSearchMapping](http://laptop-u3dq0u6v:10330/swagger-ui.html#!/cube45controller/getQuotaReportUsingGET)

3、根据规则解析kafka中的消息数据并保存到ES。

解析的规则：

数据的动作 就三种 增加，修改，删除。

所以定义好相关的动作，对应的表，列名，列值，如果是子表的话，对应的主表ID

通过这些信息，可以将维度中涉及到字典和函数的，通过相关的字典查询，函数查询获取到对应的扩展值。

最后根据动作修改ES数据。

dataMap.put("dataSource："，"Hbase");

dataMap.put("table"，"表名");*//HealthProfile，HealthProfileSub*

dataMap.put("rowkey"，"唯一键值");

dataMap.put("action"，"Put");

dataMap.put(cloumn, value);*//列名，列值*

dataMap.put("profile\_id"，"关联主表唯一键值");

测试：

put 'Healtharchive', '1001', 'info:eventNo:', 'e\_0001'

put 'Healtharchive', '1001', 'info:age:', '25'

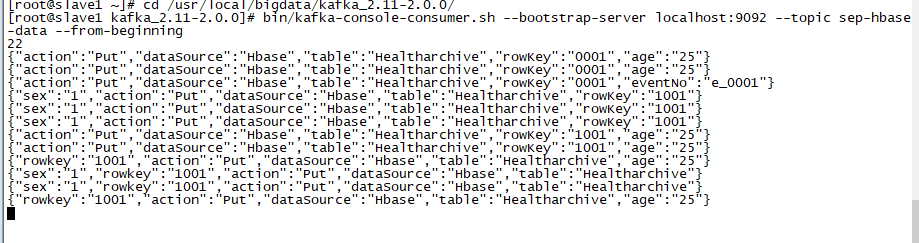
put 'Healtharchive', '1001', 'info:age:', '85'

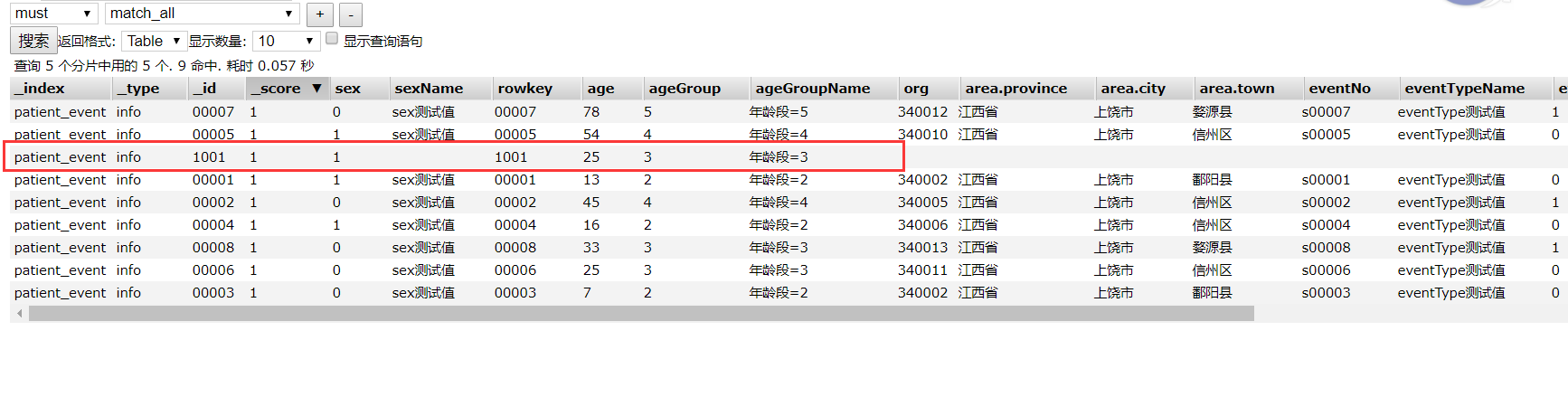
put 'Healtharchive', '1001', 'info:sex:', '1'

put 'Healtharchive', '1001', 'info:org:', 'org0001'

delete 'Healtharchive','1001','info:age' -- 删除单个字段中的值 属于修改为空值

deleteall 'Healtharchive','1001'





4、视图配置的及根据视图规则查询解析数据。

视图数据可以是数值，表格列表，图表型数据。

针对于这些类型的视图数据，定义好规则，生成对应的es查询语法。

视图查询配置相关规则说明：

视图查询需要配置对应的数据集（有es库索引名和type），及维度（行，列，下钻），指标，维度的顺序，视图的条件，指标的条件等规则。参照数据库表设计。Olap\_view\_\*\*\*\*

具体实例；



组装后的es 查询结构：

{

"size" : 0,

"query" : {

"bool" : {

"must" : {

"term" : {

"sex" : "0"

}

}

}

},

"aggregations" : {

"Age50Count-filter" : {

"filter" : {

"bool" : {

"must" : {

"range" : {

"age" : {

"from" : "50",

"to" : null,

"include\_lower" : true,

"include\_upper" : true

}

}

}

}

},

"aggregations" : {

"Age50Count-count-result" : {

"value\_count" : {

"field" : "age"

}

}

}

},

"ManCount-cardinality-result" : {

"cardinality" : {

"field" : "eventNo"

}

}

}

}

## 测试

swagger

[/api/v1.0/cube/](http://laptop-u3dq0u6v:10330/swagger-ui.html#!/cube45controller/getQuotaReportUsingGET)GetViewStatisticData

数值型数据

列表型数据

图表型数据

### 模型

* 就诊地理位置（省，市，区/县，街道，社区）
* 就诊医疗机构（性质，等级）
* 就诊类型（挂号类型?）
* 就诊科室（科室，科室类别）
* 就诊事件号
* 患者（ID，姓名）
* 出生日期（年，季度，月，日）
* 健康档案注册日期（年，季度，月，日）
* 年龄（年龄段）
* 性别
* 就诊（门诊、住院、体检）时间（年，季度，月，日）
* 出院（门诊、住院、体检）时间（年，季度，月，日），门诊，体检可按就诊时间+24小时
* 就诊天数，（门诊，体检默认一天，住院为出院-入院+1）
* 就诊费用，采用子表形式（费用类型，支付方式，费用）
* 出院主诊断（疾病分类），采用子表形式（诊断类型，诊断名，诊断编码，疾病名）？
* 入院主诊断
* 检查，采用子表形式（检查名，检查时间，费用）
* 检验，采用子表形式（检验名，检验时间，费用）
* 手术，采用子表形式（手术名，手术时间，手术等级，切口愈合等，费用）
* 用药，采用子表形式（药品名，药品分类，用量，规格，单价，费用，医保）
* 卫生材料，采用子表形式（卫生材料，类别，费用）
* 死亡标志
* 医生（ID，职称，职务，性别）
* 标签（）
* 更新时间