一、project

1.录入内容

项目name和描述

产生内容:

private String name;

private String description;

private String owner;//创建人

private ProjectStatusEnum status;//是否可用

private long createTimeUTC;//创建时间

private String lastUpdateTime; //修改时间

以及关联的哪些hive表,以及哪些model,以及哪些RealizationEntry和filter过滤条件

@JsonProperty("tables")

private Set<String> tables = new TreeSet<String>();

@JsonProperty("realizations")

private List<RealizationEntry> realizationEntries;

@JsonProperty("models")

private List<String> models;

@JsonProperty("ext\_filters")

private Set<String> extFilters = new TreeSet<String>();

2.会在指定路径下生成一个json文件,记录这些描述信息

3.因为每一个project是独立的,我猜测应该是api可以不独立,因此每一个project里面包含的model，table等等是不能迁移的.但是代码里面是可以剪切的,但是确实不能复制,可以考虑二次开发这个功能。

这个也解释了为什么project中每次都先删除再添加model等信息

即model可以在project之间迁移么，如果可以，为什么会删除所有model, public ProjectInstance updateModelToProject(String modelName, String newProjectName)

二、ext\_filters字符串对应的是json,可以解析成对象ExternalFilterDesc,属性是以下内容:

1．对象属性

@JsonProperty("name")

private String name;//filter的name

@JsonProperty("filter\_resource\_identifier")

private String filterResourceIdentifier;

@JsonProperty("filter\_table\_type")

private Functions.FilterTableType filterTableType;

@JsonProperty("source\_type")

private int sourceType = ISourceAware.ID\_EXTERNAL;

@JsonProperty("description")

private String description;//描述信息

2.通过代码可以看到filter是全局的,然后可以分配给若干个project

3．Filter应该是通过api的方式才能创建,页面上好像没有

三、model 对应的是DataModelDesc对象

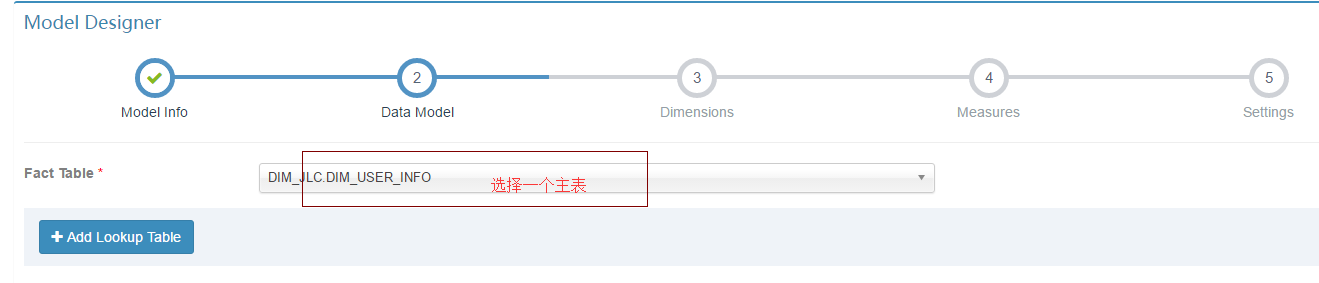
1.获取一组model集合,因为model属于project级别下的,因此要project参数

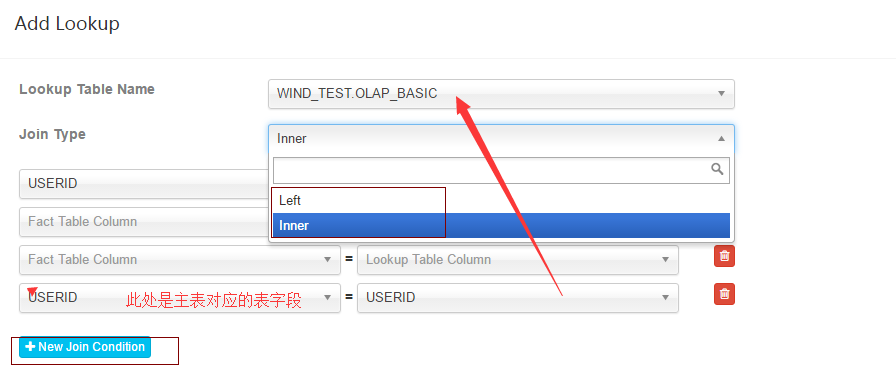
2.model创建的请求页面

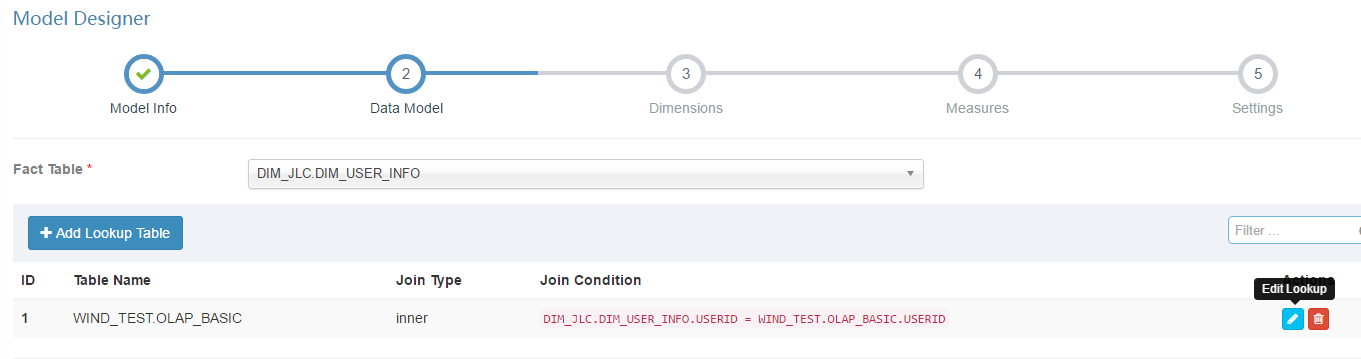
a.包含model的name和描述信息

b.选择fact表以及设置lookup

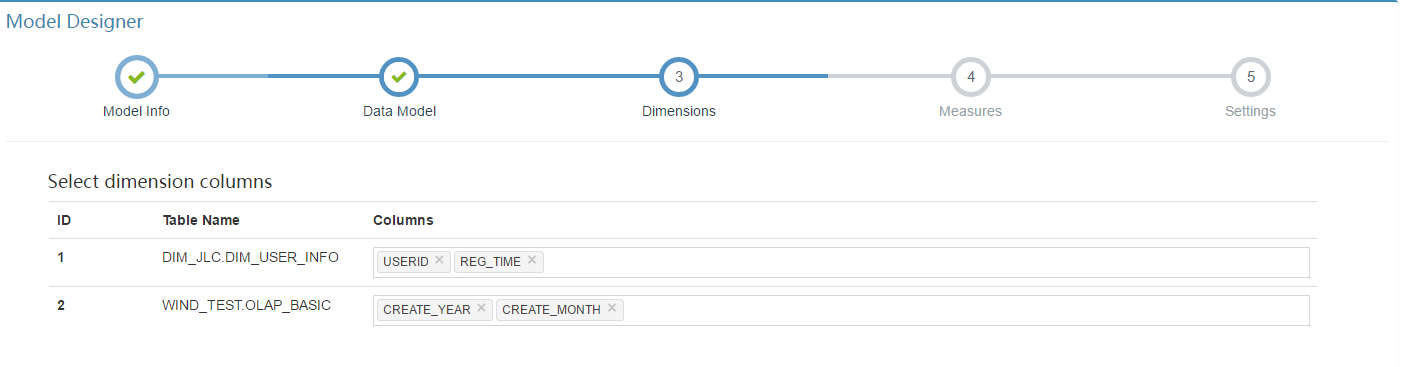
即关联N多个表组成一个大宽表的过程



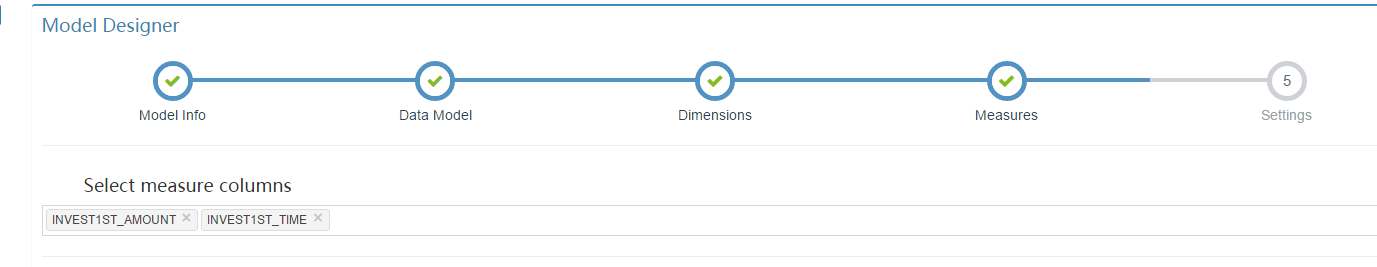




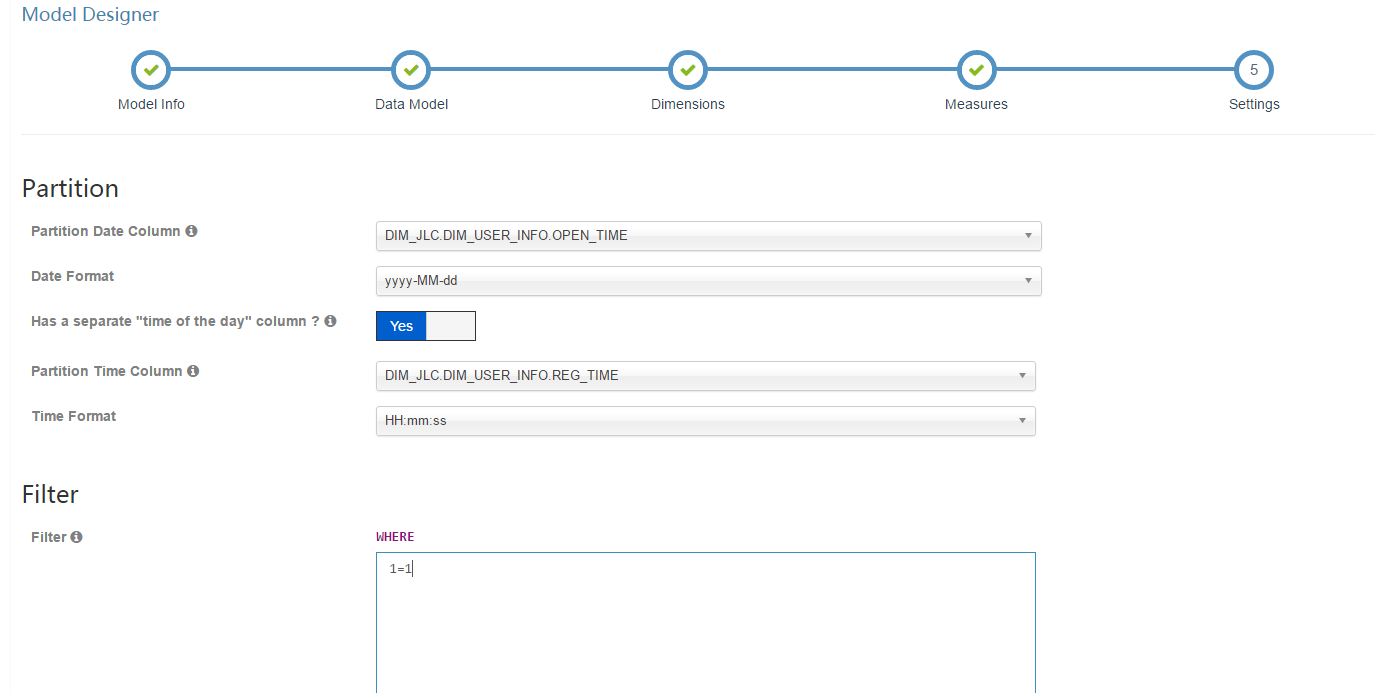
c.从对应的表中选择要的维度



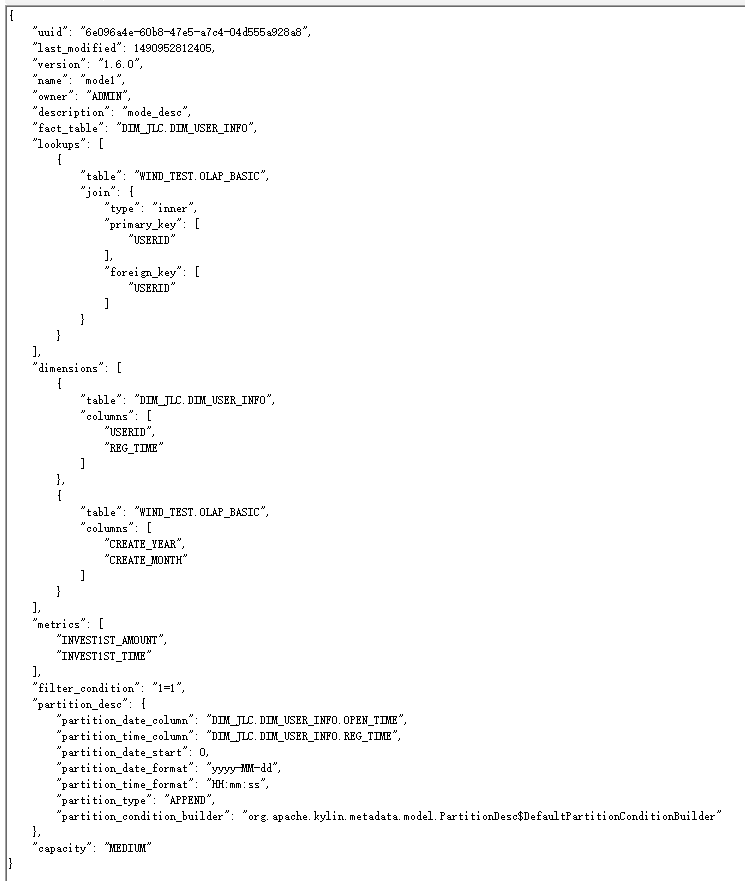
d.选择度量的列



e.设置分区以及过滤条件



f.最终创建的json



3. DataModelDesc对象属性内容:

private String name;

private String description;

private String owner;//创建人

private String factTable;//主要的事实表fact\_table

private TableDesc factTableDesc;//事实表的元数据对象

//kylin提供了两种join的方式：left join和inner join，本案例采用的是inner join的方式。以及关联条件

private LookupDesc[] lookups;

后续初始化的过程中会将表名映射成对象

private TableDesc factTableDesc;//事实表的元数据对象

//添加lookup表的元数据对象

private List<TableDesc> lookupTableDescs = Lists.newArrayList();

//第三页面添加数据

//factTable表和LookupDesc表数量集合,比如有5个表join,则在这个页面会出现5行内容,每一个表对应一行数据,可以选择该表的一些属性,组装成总维度,即按照这些维度进行group by操作

private List<ModelDimensionDesc> dimensions;

//第四个页面 选择度量的列集合

//表示页面添加的Measures属性集合,即哪些列是可以作为维度值进行统计分析的

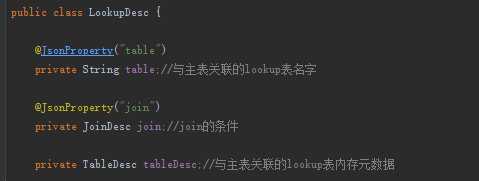
private String[] metrics;

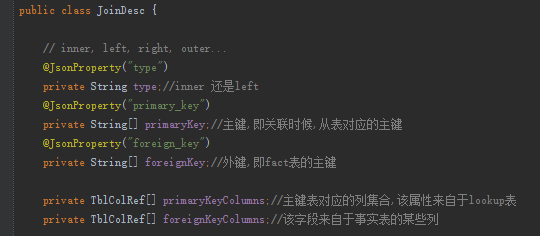
//第5个页面

private String filterCondition;//自定义的where条件

PartitionDesc partitionDesc;//如何分区,查询不同日期区间的数据

详细说明

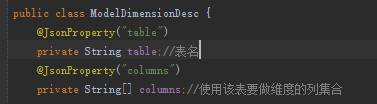




可以看到lookup表示的是其他表与主表fact\_table的关联关系,因此其他表就用lookup表示,因此该对象持有其他表的表名字以及表对象,还有join关系。

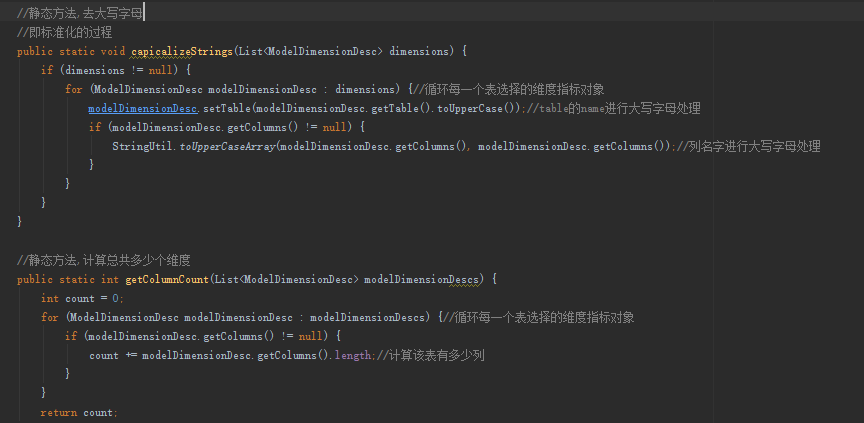
而JoinDesc中有连接类型type以及主键和外键,其中主键表示的是lookup表中对应的字段,外键就是fact\_table表字段,以及属性对象,因为关联on可以有多个字段,因此设置了多个字段使用数组。

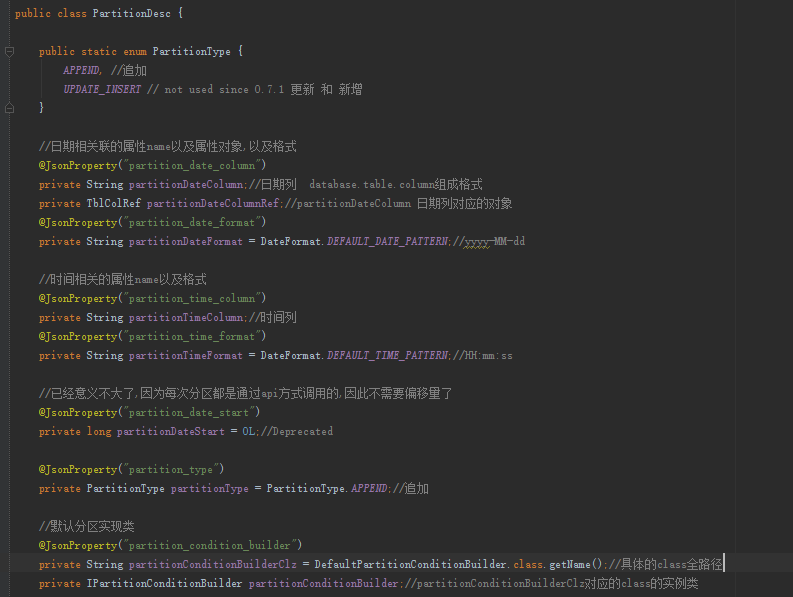
当然该对象有方法可以交换主外键关系。



表示每一个表中选择了哪些列

因此还提供了两个有用的静态方法,



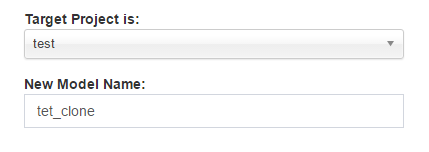


4.程序可以得到一个table是否在model中被使用,或者在某一个project下的model中被使用。

5.model可以进行clone

对已经存在的model,即modelName进行复制,所有内容都一样,只是要重新赋予一个name即可,name在请求里面

页面:



注意:

个人觉得model好鸡肋啊

Model中的Capacity 应该表示数据源的大小,但是不知道有什么意义,后续看代码再看吧