

# KYLIN 中支持定义 Logical View 副本

## Background

产品意义如下，KYLIN Temp View 可以帮助客户解决与数仓最后一公里的问题，提高客户的使用体验，还有可以解决现有产品中 CC 列的问题：

1. 需要一些方式对于载入 KYLIN 的表进行一些简单的预处理，如维表过滤、维表在一定程度聚合以适应事实表的粒度（避免多对多），维表去重，等
2. 对于指标输出，我理解有两个需求：
  - a. 将 KYLIN 中预计算的指标与维度导出成数据集，为后续分析提供高速的数据集（提升易用性，不再需要异步导出等方式）
  - b. 在 KYLIN 中通过可视化或 SQL 定义更复杂的数据集，进而向 BI 等分析终端暴露更友好的多维数据接口（减少在 BI 中表达复杂的指标逻辑而带来性能挑战，类似于把一些 BI 工具中的 SQL 数据集功能前置到 KYLIN 中）
  - c. 在 KYLIN 中基于已有 Model 或 2.b 中定义的逻辑数据集进行二次建模，使得 KYLIN 有能力预计算更多更复杂的指标（提升对于复杂指标预计算的支持，同时避免目前拿到基础指标后在 Spark 现算的性能瓶颈）

现有 KYLIN 中已经支持 DDL 语句，可以复用这部分功能，但目前仅允许创建 view 到 Hive 中，这在有些用户的生产环境中并不合规，因此有需要在此之上，建立一个仅存在于 KYLIN 中的视图表。

同时提议希望尽量少的引入新代码逻辑，避免提高整体 KYLIN 逻辑复杂度，在不影响整体功能前提下，希望在产品行为上做一些限制（后续会提到），以最小成本完成本功能。

会上补充背景：可能客户会有码表的过滤，维表的提前筛选，业务变动表后会带来模型的改变，所以有需求使用逻辑视图

## Rationale

Spark 中本身支持定义 Temp View，但调研发现普通的 Temp View 并不符合需求，因为普通 Temp View 是不可以定义 database 的，如果强行适配会与现有 KYLIN 现有逻辑冲突多。

Spark 中支持定义 Global Temp View，这与普通 view 的不同点：

1. 具有唯一指定 database，由 spark.sql.globalTempDatabase 参数控制，show tables 功能可以正常使用（普通 view 无法正常使用）。由于具有 database 属性，所以与现有 KYLIN 的数据源使用适配性好，改动少，Catalog 等信息和 hive 原生表一样使用，不一样的地方只有 show database 不会展示这个特殊 DB。
2. Global Temp View 的生命周期作用于 SparkApplication，Temp View 生命周期作用于 Spark Session，两者对 KYLIN 而言区别不大，可能 Global Temp View 更符合定义。

# Implementation

## 产品行为设计提议

为了整体设计的简洁性，减少对现有逻辑的侵入，同时最小成本完成本功能，提高 ROI，提议限制两种产品行为：

- a. 仅允许创建临时视图至某个固定数据库（global view database），可在配置文件中指定，这是 Spark 行为限制，如果要定义至普通 hive 库中，需要修改 Spark
- b. DDL 暂且仅支持创建(create，或者整体更新 replace)，删除(drop)，展示操作，不支持修改(alter) 操作，支持修改操作意味着需要记录下一系列连续执行语句，在节点同步和启动时需要有依次重放机制，这增加了不少细节的复杂度。

## 元数据改动

~~由于 view 定义需要保存 SQL 文本至 KYLIN 数据库中，考虑到客户写的 SQL text 文本可能会很长，数量也可能很多，可能对现有的元数据写入，同步有性能影响，所以提议元数据使用类似查询历史表一样，使用新表进行保存。~~

~~表的字段暂时设计为：id, tablename, SQL, createtime~~

View SQL 定义还是定义在现有元数据中，是系统级别的元数据(/\_global/tableName)

执行元数据写入的节点为 global owner 节点，其他节点只读操作。这里并没有元数据同步问题，普通节点直接读取元数据库，无需缓存。这里读取 sql 并不是高频操作，仅在 执行 DDL 和重启节点时需要。

其他模型相关操作与现有实现不变，重载表、查询逻辑不变。

## 节点同步视图改动

- i. 从节点 Sparder 启动、重启时根据客户定义的 SQL 语句，加载所有临时视图
- ii. 构建、抽样时，模型中如果使用临时视图表，需要加载对应临时视图（不需要加载所有临时视图）
- iii. 主节点 DDL 临时视图语句执行时，主动通知从节点同步。
- iv. 从节点定时同步临时表，从节点内存中记录临时视图表 + 创建时间 (防止 ABA 修改问题，即创建-删除-再创建，也能感知)，当有不一致的情况，执行对应表的 DDL 操作
- v. 提供一个接口使从节点主动同步 SQL（类似于 HA 处理方式）

## 数据源展示改动

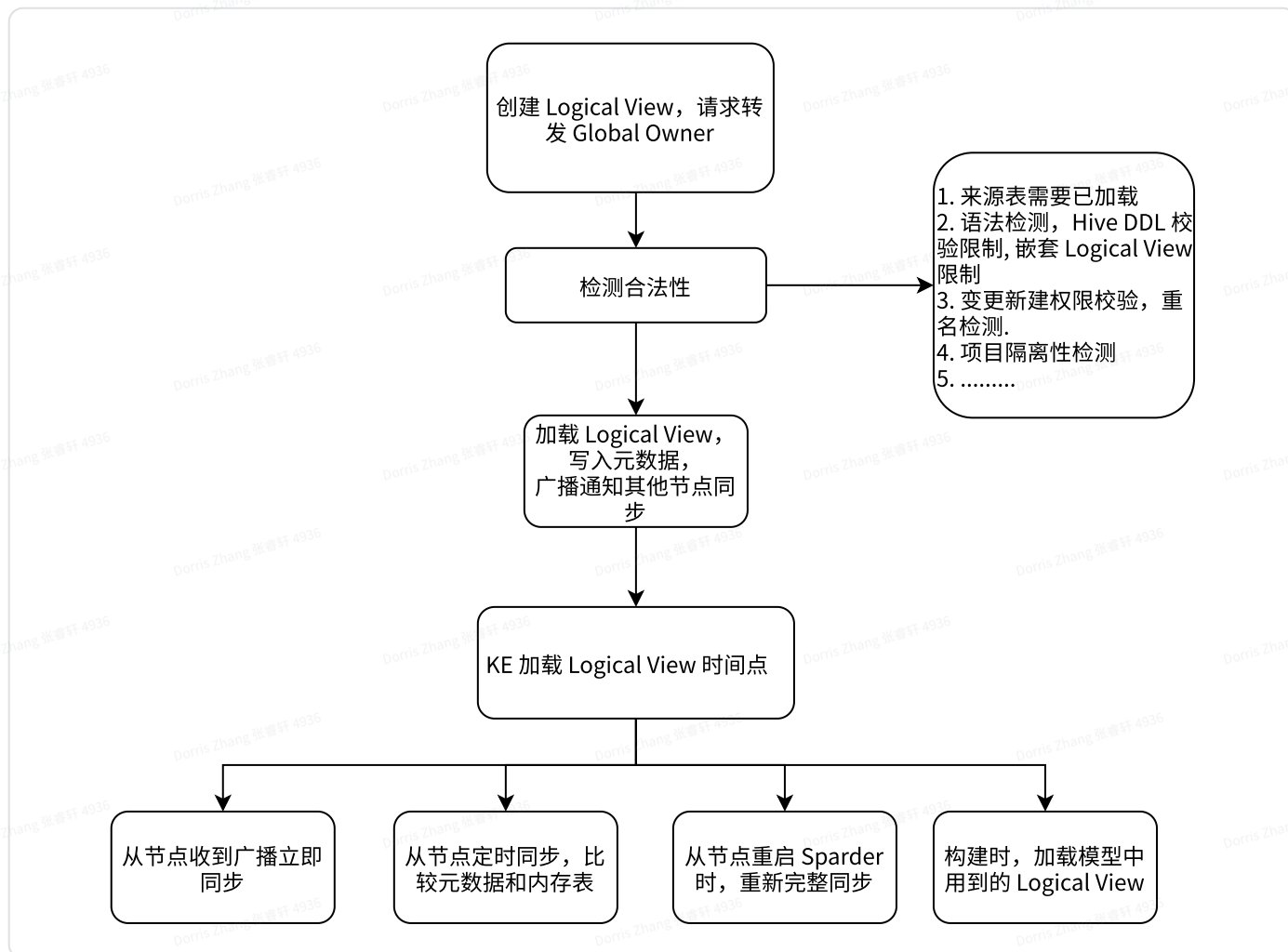
目前看来，与现有 Hive 数据源 Catalog 执行差异小，仅 show databases 时无法列出 global view 的 database，故当启用临时视图功能时，额外多返回一个 global view database，其他语句与 Hive 执行一致（show tables 可以正常返回）

## DDL 操作限制改动

复用现有 DDL 实现，包括前端，检测来源表逻辑不变，DDL 语句类型如上额外支持 CREATE Logical VIEW, Drop Logical View

提供一个 tab 页面专门查询所有 Logical View 列表，可分项目展示，并且可以修改 (replace) Logical View 语句

## 流程图



## 接口文档

废除原先 ddl 的参数 `kylin.source.ddl.enabled`, 改为分别控制:

`kylin.source.ddl.hive.enabled=false`

`kylin.source.ddl.logical-view.enabled=false`

额外的新增参数:

Logical View database 名称, 不能与普通 Hive 库重名:

`kylin.source.ddl.logical-view.database=KYLIN_LOGICAL_VIEW`

Logical View 定时同步频率 (单位秒):

`kylin.source.ddl.logical-view-catchup-interval=60`

Logical View DDL :

## 支持 create\drop\replace logical view

```
1
2 curl -X POST \
3 http://10.1.3.35:18082/kylin/api/spark_source/ddl \
4 -H 'accept: application/vnd.apache.kylin-v4+json' \
5 -H 'accept-language: en' \
6 -H 'authorization: Basic QURNSU46S1lMSU4=' \
7 -H 'content-type: application/json' \
8 -d '{"sql": "create logical view logical_view_table1 AS select * from SSB.Cust
```

其中 restrict 可选为 logic、hive、replaceLogicalView（修改 logical view 时传入该参数），如果不填则代表不限制

注意原先 hive ddl 接口需要改变的是 project 参数修改为 ddl\_project 参数，并且新增 restrict 字段为 hive

### DDL 接口描述：

Hive、Logical View 不同处是 page\_type= hive 或者 logic：

```
1 curl -X GET \
2 'http://10.1.3.35:18082/kylin/api/spark_source/ddl/description?project=lcl&
3 -H 'accept: application/vnd.apache.kylin-v4+json' \
4 -H 'accept-language: en' \
5 -H 'authorization: Basic QURNSU46S1lMSU4=' \
6 -H 'content-type: application/json'
```

### Logical View 列表

会返回所有 logical view，如果不属于该项目，created sql 为 \*\*\*

table 参数代表模糊搜索，可以不加该参数

```
1 curl -X GET \
2 'http://10.1.3.35:18082/kylin/api/spark_source/ddl/view_list?project=lcl&table
3 -H 'accept: application/vnd.apache.kylin-v4+json' \
4 -H 'accept-language: en' \
5 -H 'authorization: Basic QURNSU46S1lMSU4=' \
6 -H 'cache-control: no-cache'
```

返回体示例：

```
1 {
2   "code": "000",
3   "data": [
4     {
5       "uuid": "e89424d2-2d84-126d-b6d1-0934e36e60dd",
6       "last_modified": 1671074540881,
7       "create_time": 1671074540881,
8       "version": "4.0.0.0",
9       "mvcc": 2,
10      "table_name": "LOGICAL_VIEW_TABLE1",
11      "created_sql": "replace logical VIEW logical_view_table1 AS select
12      "modified_user": "ADMIN",
13      "created_project": "\c\l"
14    },
15    {
16      "uuid": "e2e38850-9e2b-0899-ce34-6716f8c7ceee",
17      "last_modified": 1671075074215,
18      "create_time": 1671075074215,
19      "version": "4.0.0.0",
20      "mvcc": 0,
21      "table_name": "LOGICAL_VIEW_TABLE2",
22      "created_sql": "***",
23      "modified_user": "ADMIN",
24      "created_project": "ssb"
25    }
26  ],
27  "msg": ""
28 }
```