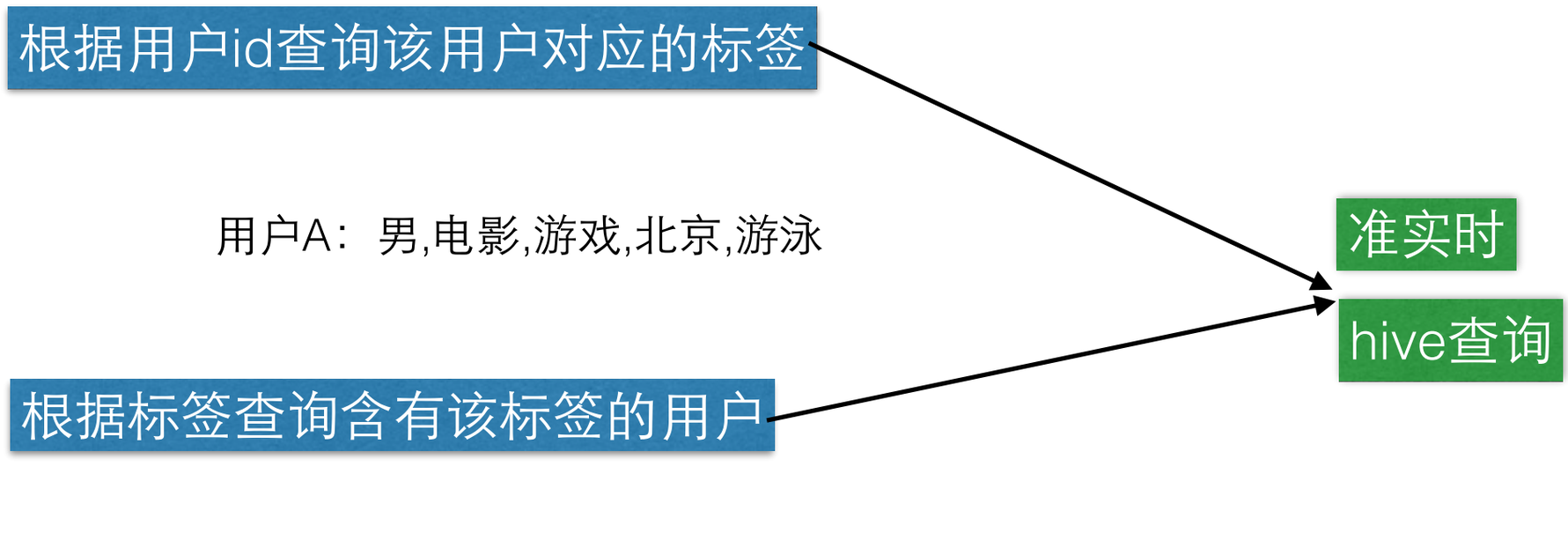
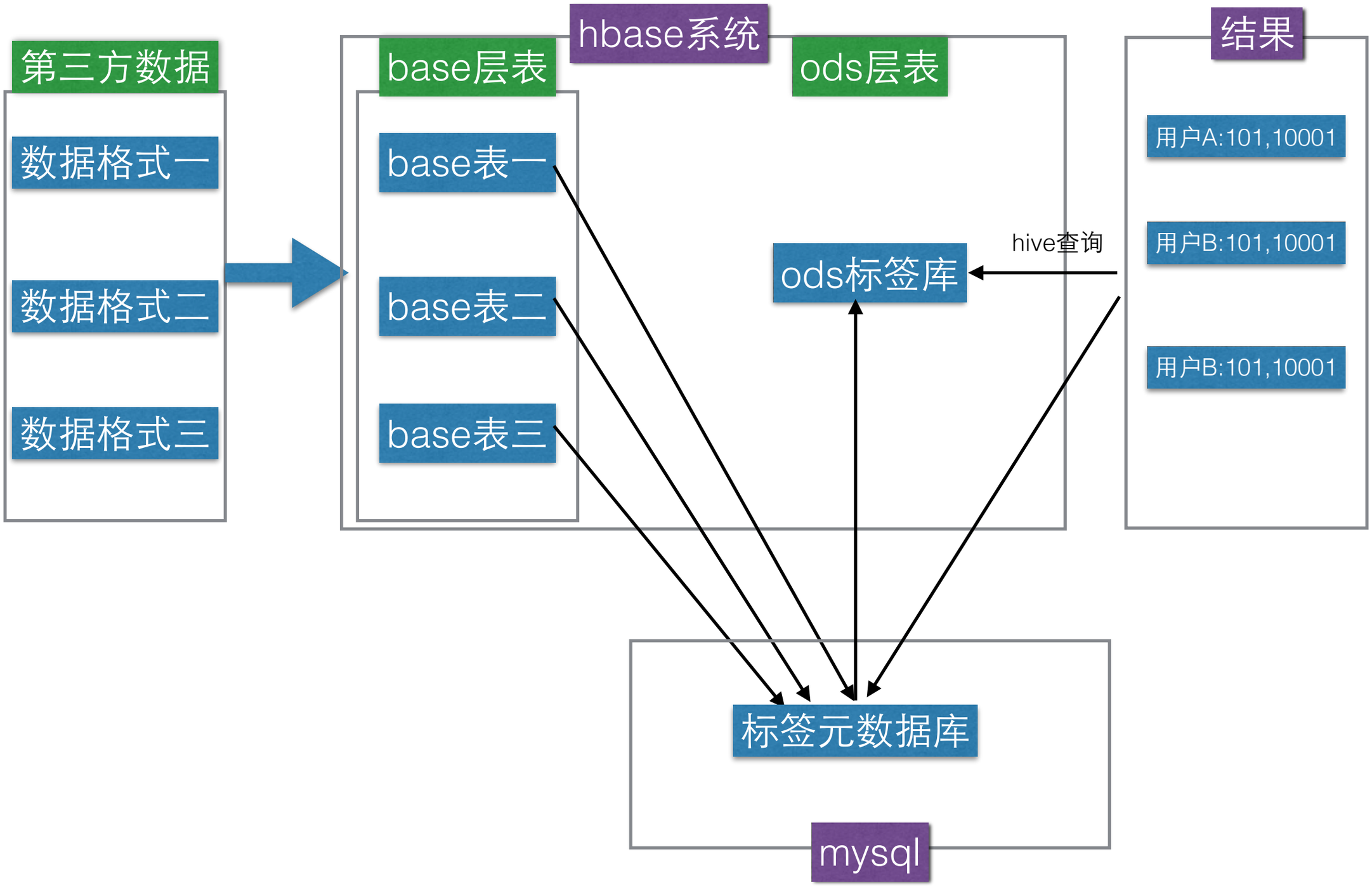
DMP 项目总结V1.0

业务场景



游泳:用户A，用户B

架构设计图


第三方数据
数据格式一
数据格式二
数据格式三
hbase系统
base层表
base表一
base表二
base表三
ods层表
ods标签库
hive查询
结果
用户A:101,10001
用户B:101,10001
用户B:101,10001
标签元数据库
mysql

详细设计

数据清洗模块

第三方过来的数据，根据不同的数据源建不同的base表，这部分作为dmp的基础数据，保持完整性，同步到hbase表中，
第三方的数据有几种格式，就需要建几张base层表，所以数据清洗加载数据到hbase 的base层表，较易实现

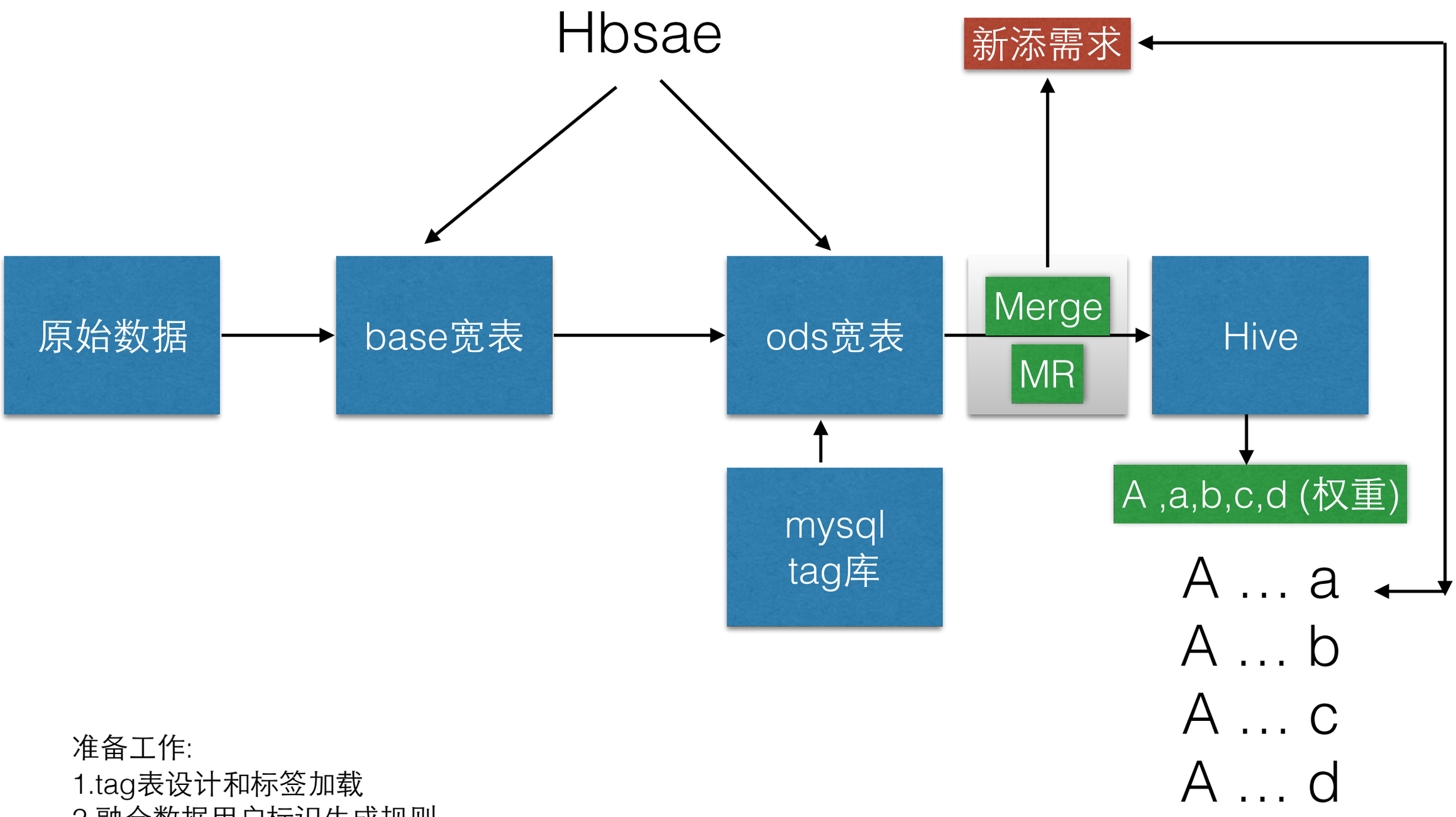
mysql标签元数据库

针对标签可能扩展为几万、十几万、甚至百万级的量，所以考虑用mysql 来存储标签定义，每一个主键id作为一个标签号，
例如:
一级标签号从1-999
二级标签用它对应的一级标签id为前缀，例如标签1的下级标签是1001-1999，标签10的下级标签是10001-10999
同理，三级标签用它对应的二级标签id为前缀
……
也就是说如果一个用户都含有这三种标签，那么应该是:用户A:2,1003,1001001

Hbase ods层表

ods表设计

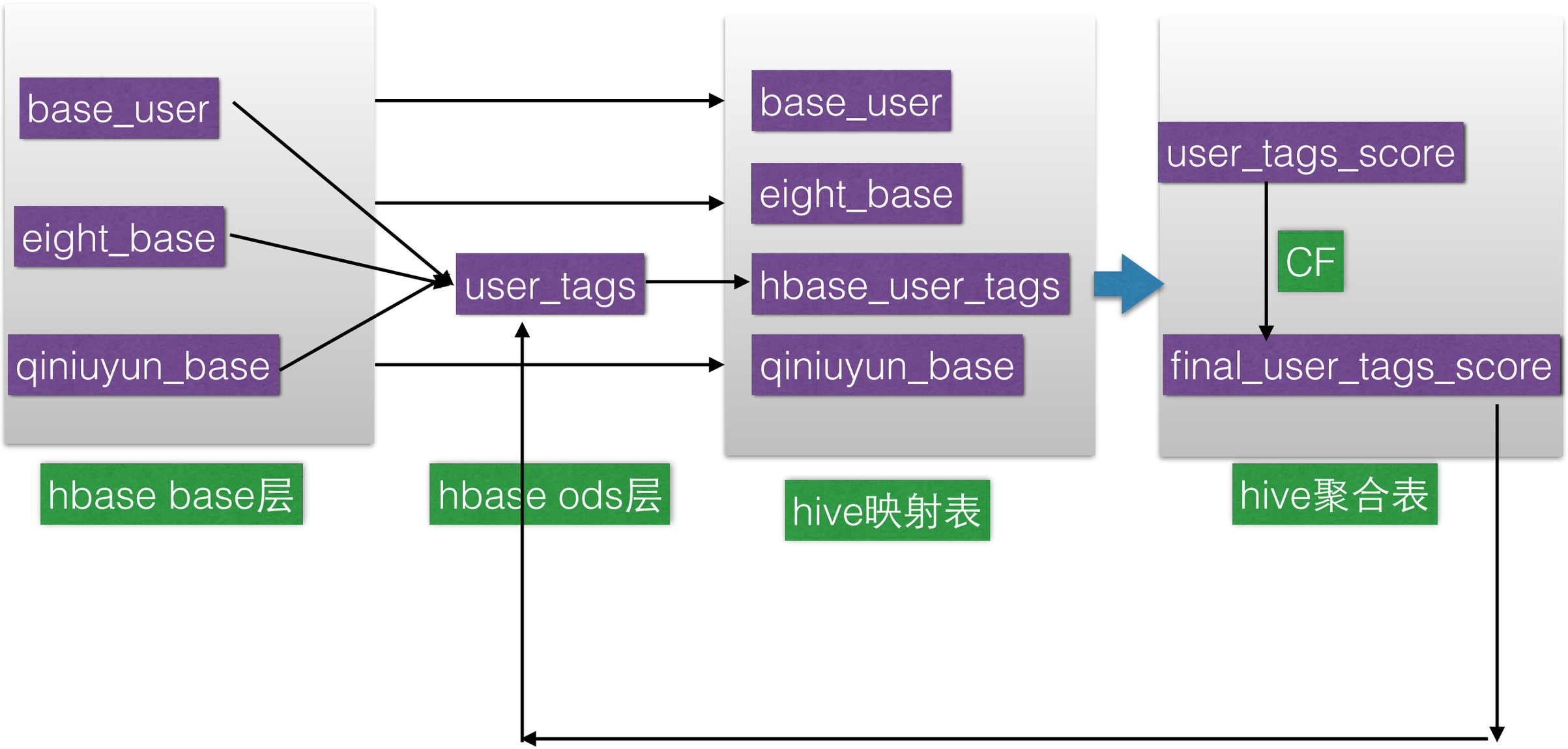
rowkey(用户标识)	列一(唯一标识)	列二(标签id)
A	A1	2,1003,1001001
B	B1	3,1001



准备工作:
1.tag表设计和标签加载
2.融合数据用户标识生成规则

时间0609

实践过程


base_user
eight_base
qiniuyun_base
user_tags
base_user
eight_base
hbase_user_tags
qiniuyun_base
user_tags_score
CF
final_user_tags_score
hbase base层
hbase ods层
hive映射表
hive聚合表

数据融合和映射方案

1.根据用户id查询标签
2.根据标签查询用户id

hbase_user_tags

final_user_tags_score

uid(Int)

LF平台id

id_mapping

user_tags_score

id	bigint
phone	string
idfa	string
qiniu_id	string
tag_id	bigint
tag_name	string
score	double

cf

a)根据lf平台用户id查询标签
b)根据第三方平台用户id查询标签
c)根据phone查询标签
d)根据idfa查询标签
e)针对id查询标签id或者中文标签