**电商数据分析平台项目**

**教案**

**文档修订记录**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 文件状态：  [√ ]草稿  [ ]正式发布 | | **当前版本：** | | V1.1 | | | |
| **作 者：** | | 武玉飞 | | | |
| **审 核 人：** | |  | | | |
| **发布日期：** | | 20170518 | | | |
| **编制日期** | **版本** | | **状态** | **简要说明** | **作者** | **审核者** | **审核日期** |
| 20170518 | V1.1 | | A |  |  |  |  |
|  |  | |  |  |  |  |  |
|  |  | |  |  |  |  |  |
|  |  | |  |  |  |  |  |
|  |  | |  |  |  |  |  |
|  |  | |  |  |  |  |  |

说明：

1. 按修改时间先后倒序排列，最新修改的排在第一行。
2. 版本栏中填入版本编号或者更改记录编号。
3. 状态分为三种状态：A­——增加；M——修改；D——删除。
4. 在简要说明栏中填写变更的内容和变更的范围。
5. 表中所有日期格式为：YYYYMMDD。

**目 录**

[第1章 项目概述 1](#_Toc497920156)

[1.1 项目简介 1](#_Toc497920157)

[1.2 项目目标 1](#_Toc497920158)

[1.3 业务需求简介 2](#_Toc497920159)

[1.3.1 用户访问session 2](#_Toc497920160)

[1.3.2 页面单跳转化率统计 2](#_Toc497920161)

[1.3.3 热门商品离线统计 2](#_Toc497920162)

[1.3.4 广告流量实时统计 2](#_Toc497920163)

[第2章 项目主体架构 3](#_Toc497920164)

[2.1 项目架构 3](#_Toc497920165)

[2.2 离线日志采集流程（参考） 4](#_Toc497920166)

[2.3 实时数据采集流程（参考） 5](#_Toc497920167)

[第3章 模拟业务数据源 5](#_Toc497920168)

[3.1 离线数据 5](#_Toc497920169)

[3.1.1 数据模型 5](#_Toc497920170)

[3.1.2 数据说明 6](#_Toc497920171)

[3.2 实时数据 6](#_Toc497920172)

[3.2.1 数据模型 6](#_Toc497920173)

[3.2.2 数据说明 6](#_Toc497920174)

[3.3 查询条件 7](#_Toc497920175)

[3.3.1 数据模型 7](#_Toc497920176)

[3.3.2 数据说明 7](#_Toc497920177)

[第4章 用户访问Session分析 8](#_Toc497920178)

[4.1 模块概述 8](#_Toc497920179)

[4.2 需求一：Session聚合统计 8](#_Toc497920180)

[4.2.1 要做什么？ 8](#_Toc497920181)

[4.2.2 数据结构 9](#_Toc497920182)

[4.2.3 数据处理流程 10](#_Toc497920183)

[4.3 需求二：Session随机抽取 11](#_Toc497920184)

[4.3.1 要做什么？ 11](#_Toc497920185)

[4.3.2 数据模型 11](#_Toc497920186)

[4.3.3 数据处理流程 12](#_Toc497920187)

[4.4 需求三：Top10热门品类 13](#_Toc497920188)

[4.4.1 要做什么？ 13](#_Toc497920189)

[4.4.2 数据模型 13](#_Toc497920190)

[4.4.3 数据处理流程 15](#_Toc497920191)

[4.5 需求四：Top10活跃Session 15](#_Toc497920192)

[4.5.1 要做什么？ 15](#_Toc497920193)

[4.5.2 数据模型 16](#_Toc497920194)

[4.5.3 数据处理流程 16](#_Toc497920195)

[第5章 页面单跳转化率分析 17](#_Toc497920196)

[5.1 模块概述 17](#_Toc497920197)

[5.2 需求一：页面单跳转化率 17](#_Toc497920198)

[5.2.1 数据模型 17](#_Toc497920199)

[5.2.2 数据处理流程 18](#_Toc497920200)

[第6章 各区域热门商品统计分析 18](#_Toc497920201)

[6.1 模块概述 18](#_Toc497920202)

[6.2 需求一：各区域Top3商品统计 19](#_Toc497920203)

[6.2.1 数据模型 19](#_Toc497920204)

[6.2.2 数据处理流程 20](#_Toc497920205)

[第7章 广告点击流量实时统计分析 20](#_Toc497920206)

[7.1 模块概述 20](#_Toc497920207)

[7.2 需求一：广告统计用户黑名单 22](#_Toc497920208)

[7.2.1 要做什么？ 22](#_Toc497920209)

[7.2.2 数据模型 22](#_Toc497920210)

[7.2.3 数据处理流程 23](#_Toc497920211)

[7.3 需求二：广告点击实时统计 23](#_Toc497920212)

[7.3.1 要做什么？ 23](#_Toc497920213)

[7.3.2 数据模型 23](#_Toc497920214)

[7.3.3 数据处理流程 24](#_Toc497920215)

[7.4 需求三：实时统计各省最热广告 24](#_Toc497920216)

[7.4.1 要做什么？ 24](#_Toc497920217)

[7.4.2 数据模型 24](#_Toc497920218)

[7.4.3 数据处理流程 25](#_Toc497920219)

[7.5 需求四：最近一小时广告点击趋势统计 25](#_Toc497920220)

[7.5.1 要做什么？ 25](#_Toc497920221)

[7.5.2 数据模型 25](#_Toc497920222)

[7.5.3 数据处理流程 26](#_Toc497920223)

[第8章 大数据部署建设（选做） 26](#_Toc497920224)

[8.1 通过azkaban将业务进行部署 26](#_Toc497920225)

[8.2 通过Zeppelin将MySQL中的数据进行图表输出 26](#_Toc497920226)

# 项目概述

## 项目简介

本項目主要讲解一个大型电商网站后台的企业级大数据统计分析平台，该平台以Spark为主，对电商网站的流量进行离线和实时的分析。

该大数据分析平台对电商网站的各种用户行为（访问行为、购物行为、广告点击行为等）进行复杂的分析。用统计分析出来的数据，辅助公司中的PM（产品经理）、数据分析师以及管理人员分析现有产品的情况，并根据用户行为分析结果持续改进产品的设计，以及调整公司的战略和业务。最终达到用大数据技术来帮助提升公司的业绩、营业额以及市场占有率的目标。

项目主要使用了Spark技术生态栈中最常用的三个技术框架，Spark Core、Spark SQL和Spark Streaming，进行离线计算和实时计算业务模块的开发。实现了包括用户访问session分析、页面单跳转化率统计、热门商品离线统计、广告流量实时统计4个业务模块。

项目中所有的业务功能模块都是直接从实际企业项目中抽取出来的，业务复杂度没有任何缩水，通过合理的将实际业务模块进行技术整合与改造，该项目几乎完全涵盖了Spark Core、Spark SQL和Spark Streaming这三个技术框架中大部分的功能点、知识点。

## 项目目标

1、掌握电商系统中Spark的主要使用场景以及建设流程。

2、掌握企业级的Spark项目的复杂性能调优、线上故障解决经验、数据倾斜全套处理方案。

3、通过项目实战，完全将Spark所有技术点和知识点都应用在项目中，掌握如何灵活应用Spark各项技术来实现各种复杂业务需求。

## 业务需求简介

### 用户访问session

该模块主要是对用户访问session进行统计分析，包括session的聚合指标计算、按时间比例随机抽取session、获取每天点击、下单和购买排名前10的品类、并获取top10品类的点击量排名前10的session。该模块可以让产品经理、数据分析师以及企业管理层形象地看到各种条件下的具体用户行为以及统计指标，从而对公司的产品设计以及业务发展战略做出调整。主要使用Spark Core实现。

### 页面单跳转化率统计

该模块主要是计算关键页面之间的单步跳转转化率，涉及到页面切片算法以及页面流匹配算法。该模块可以让产品经理、数据分析师以及企业管理层看到各个关键页面之间的转化率，从而对网页布局，进行更好的优化设计。主要使用Spark Core实现。

### 热门商品离线统计

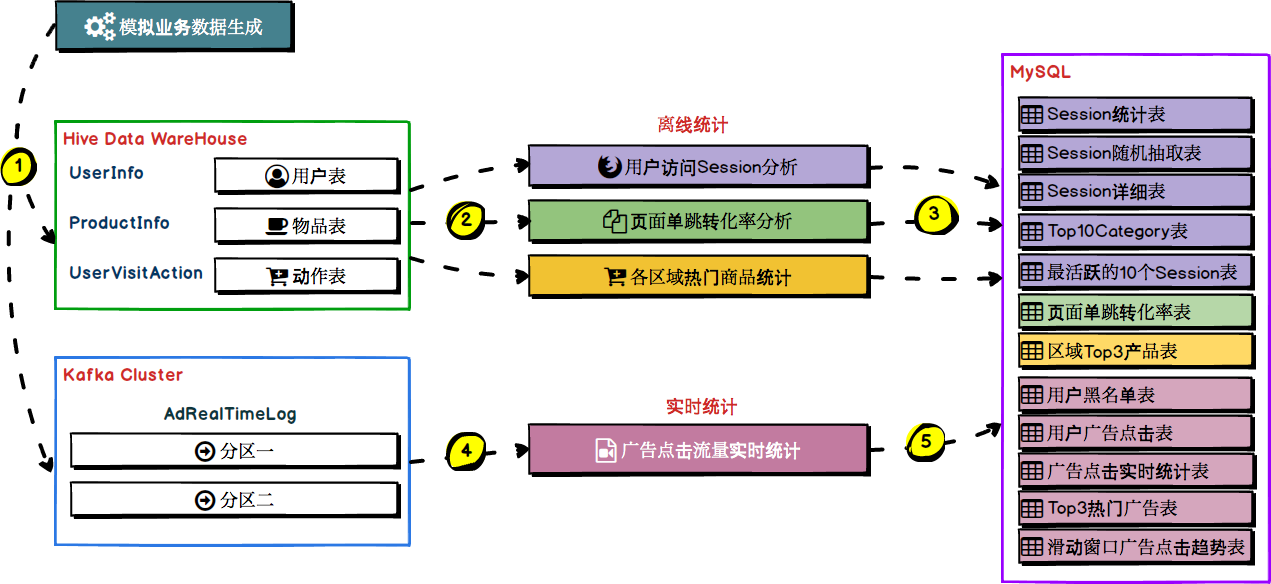
该模块主要实现每天统计出各个区域的top3热门商品。该模块可以让企业管理层看到公司售卖的商品的整体情况，从而对公司的商品相关的战略进行调整。主要使用Spark SQL实现。

### 广告流量实时统计

该模块负责实时统计公司的广告流量，包括广告展现流量和广告点击流量。实现动态黑名单机制，以及黑名单过滤；实现滑动窗口内的各城市的广告展现流量和广告点击流量的统计；实现每个区域每个广告的点击流量实时统计；实现每个区域top3点击量的广告的统计。主要使用Spark Streaming实现。

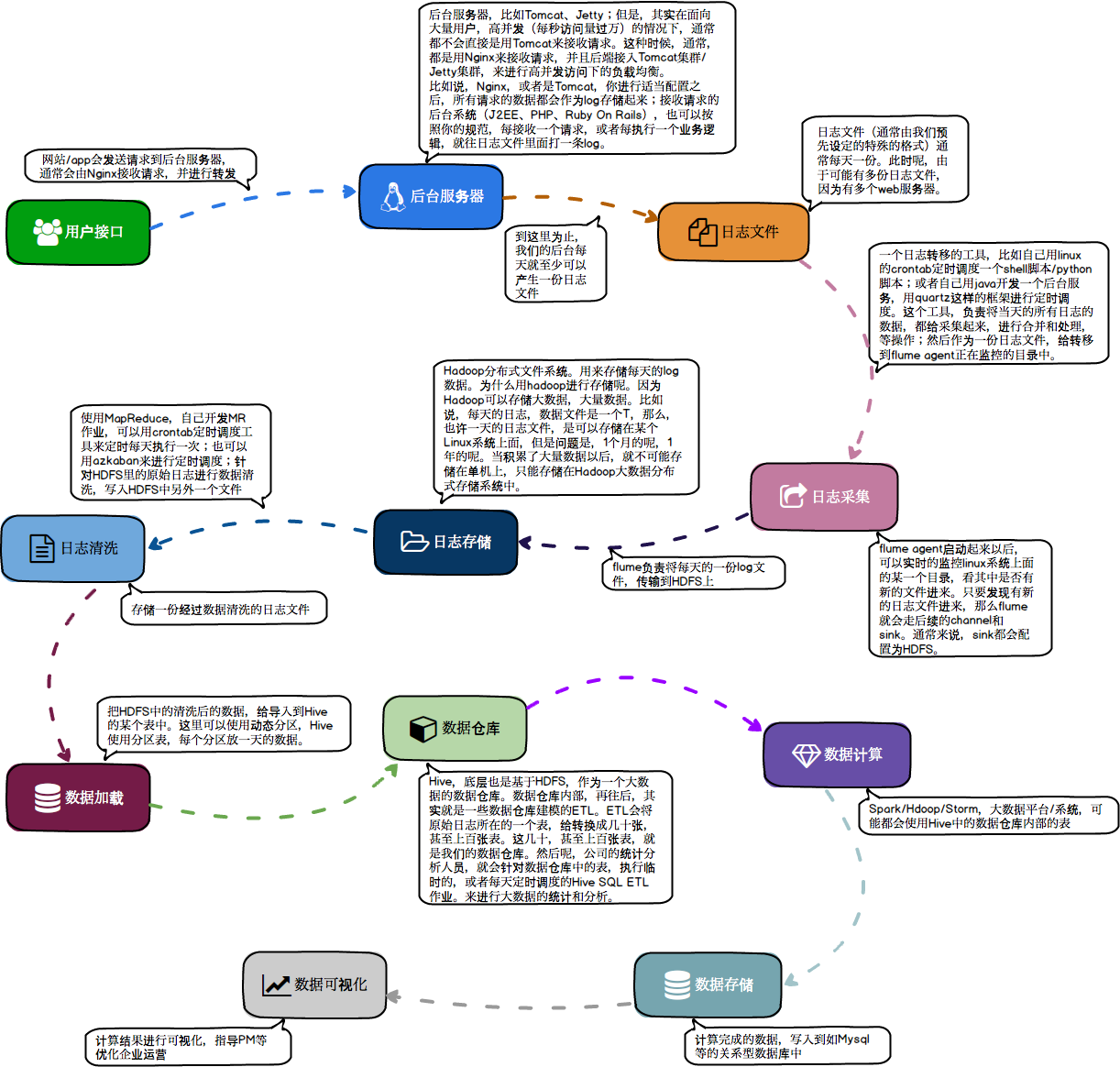
# 项目主体架构

## 项目架构

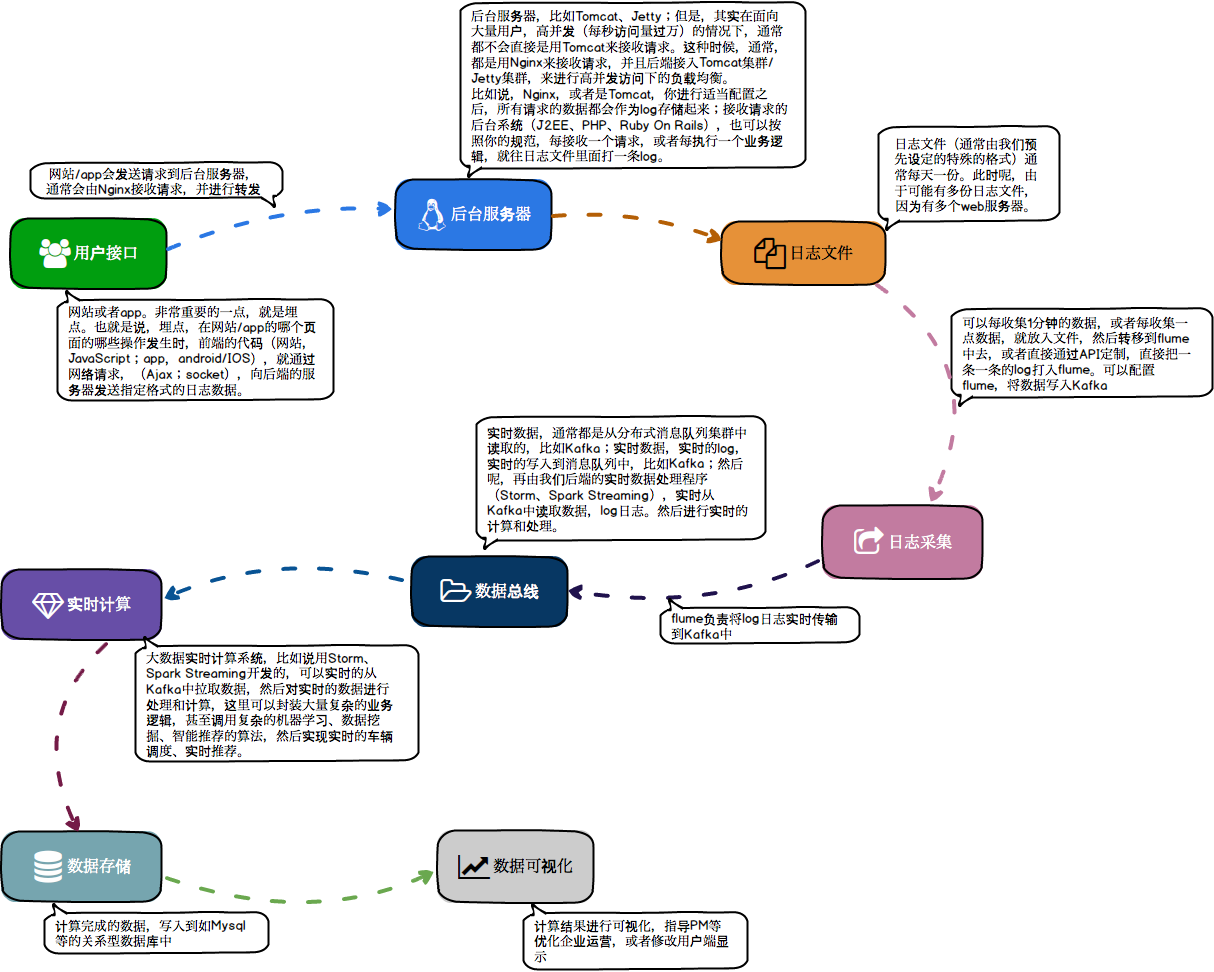


用户行为数据在网站上最简单的存在形式就是日志。网站在运行过程中会产生大量的原始日志RAW LOG，将其存储在文件系统中，企业会将多种原始日志按照用户行为汇总成会话日志SESSION LOG，每一个会话日志表示用户的一种反馈。

## 离线日志采集流程（参考）



## 实时数据采集流程（参考）



# 模拟业务数据源

## 离线数据

### 数据模型



### 数据说明

|  |
| --- |
| */\*\*  \* 模拟的数据  \* date：是当前日期  \* age: 0 - 59  \* professionals: professional[0 - 59]  \* cities: 0 - 9  \* sex: 0 - 1  \* keywords: ("火锅", "蛋糕", "重庆辣子鸡", "重庆小面", "呷哺呷哺", "新辣道鱼火锅", "国贸大厦", "太古商场", "日本料理", "温泉")  \* categoryIds: 0 - 99  \* ProductId: 0 - 99  \*/*  生成的数据利用Spark SQL保存在Hive表中。 |

## 实时数据

### 数据模型

程序每5秒向Kafka集群写入数据，格式如下：

*格式 ：timestamp province city userid adid*

### 数据说明

|  |
| --- |
| */\*\*  \* 模拟的数据  \* 时间点: 当前时间毫秒  \* userId: 0 - 99  \* 省份、城市 ID相同 ： 1 - 9  \* adid: 0 - 19  \* ((0L,"北京","北京"),(1L,"上海","上海"),(2L,"南京","江苏省"),(3L,"广州","广东省"),(4L,"三亚","海南省"),(5L,"武汉","湖北省"),(6L,"长沙","湖南省"),(7L,"西安","陕西省"),(8L,"成都","四川省"),(9L,"哈尔滨","东北省"))  \* 格式 ：timestamp province city userid adid  \* 某个时间点 某个省份 某个城市 某个用户 某个广告  \*/*  省份的ID和城市的ID相同 |

## 查询条件

### 数据模型

查询对象没有使用MySQL数据库，采用了配置文件，在Commons项目中。格式如下:

|  |
| --- |
| **task.params.json**=**{startDate:"2017-10-20", \  endDate:"2017-10-31", \  startAge: 20, \  endAge: 50, \  professionals: "", \  cities: "", \  sex:"", \  keywords:"", \  categoryIds:"", \  targetPageFlow:"1,2,3,4,5,6,7"}** |

### 数据说明

|  |
| --- |
| *# 可以使用的属性如下： # startDate： 格式： yyyy-MM-DD [必选] # endDate: 格式： yyyy-MM-DD [必选] # startAge: 范围： 0 - 59 # endAge: 范围： 0 - 59 # professionals： 范围：professionals[0 - 59] # cities: 0 - 9 ((0,"北京","华北"),(1,"上海","华东"),(2,"南京","华东"),(3,"广州","华南"),(4,"三亚","华南"),(5,"武汉","华中"),(6,"长沙","华中"),(7,"西安","西北"),(8,"成都","西南"),(9,"哈尔滨","东北")) # sex: 范围： 0 - 1 # keywords: 范围： ("火锅", "蛋糕", "重庆辣子鸡", "重庆小面", "呷哺呷哺", "新辣道鱼火锅", "国贸大厦", "太古商场", "日本料理", "温泉") # categoryIds：0 - 99，以逗号分隔 # targetPageFlow： 0 - 99， 以逗号分隔* |

# 用户访问Session分析

## 模块概述

用户在电商网站上，通常会有很多的点击行为，首先通常都是进入首页；然后可能点击首页上的一些商品；点击首页上的一些品类；也可能随时在搜索框里面搜索关键词；还可能将一些商品加入购物车；对购物车中的多个商品下订单；最后对订单中的多个商品进行支付。

用户的每一次操作，其实可以理解为一个action，比如点击、搜索、下单、支付，用户session，指的就是，从用户第一次进入首页，session就开始了。然后在一定时间范围内，直到最后操作完（可能做了几十次、甚至上百次操作）。离开网站，关闭浏览器，或者长时间没有做操作；那么session就结束了。

以上用户在网站内的访问过程，就称之为一次session。简单理解，session就是某一天某一个时间段内，某个用户对网站从打开/进入，到做了大量操作，到最后关闭浏览器的过程就叫做session。

session实际上就是一个电商网站中最基本的数据和大数据。那么大数据，面向消费者，那么对于电商网站最基本的分析莫过于是面向用户访问行为/用户访问session的分析。

1、可以根据使用者指定的某些条件，筛选出指定的一些用户（有特定年龄、职业、城市）；

2、对这些用户在指定日期范围内发起的session，进行聚合统计，比如，统计出访问时长在0~3s的session占总session数量的比例；

3、按时间比例，比如一天有24个小时，其中12:00~13:00的session数量占当天总session数量的50%，当天总session数量是10000个，那么当天总共要抽取1000个session，ok，12:00~13:00的用户，就得抽取1000\*50%=500。而且这500个需要随机抽取。

4、获取点击量、下单量和支付量都排名10的商品种类

5、获取top10的商品种类的点击数量排名前10的session

## 需求一：Session聚合统计

### 要做什么？

统计出符合条件的session中，访问时长在1s~3s、4s~6s、7s~9s、10s~30s、30s~60s、1m~3m、3m~10m、10m~30m、30m以上各个范围内的session占比；访问步长在1~3、4~6、7~9、10~30、30~60、60以上各个范围内的session占比，并将结果保存到MySQL数据库中。注意在计算之前需要根据查询条件筛选session，查询条件比如搜索过某些关键词的用户、访问时间在某个时间段内的用户、年龄在某个范围内的用户、职业在某个范围内的用户、所在某个城市的用户，发起的session。找到对应的这些用户的session，并进行统计，之所以需要有筛选主要是可以让使用者，对感兴趣的和关系的用户群体，进行后续各种复杂业务逻辑的统计和分析，那么拿到的结果数据，就是只是针对特殊用户群体的分析结果；而不是对所有用户进行分析的泛泛的分析结果。比如说，现在某个企业高层，就是想看到用户群体中，28~35岁的，老师职业的群体，对应的一些统计和分析的结果数据，从而辅助高管进行公司战略上的决策制定。

session访问时长，也就是说一个session对应的开始的action，到结束的action，之间的时间范围；还有，就是访问步长，指的是，一个session执行期间内，依次点击过多少个页面，比如说，一次session，维持了1分钟，那么访问时长就是1m，然后在这1分钟内，点击了10个页面，那么session的访问步长，就是10.

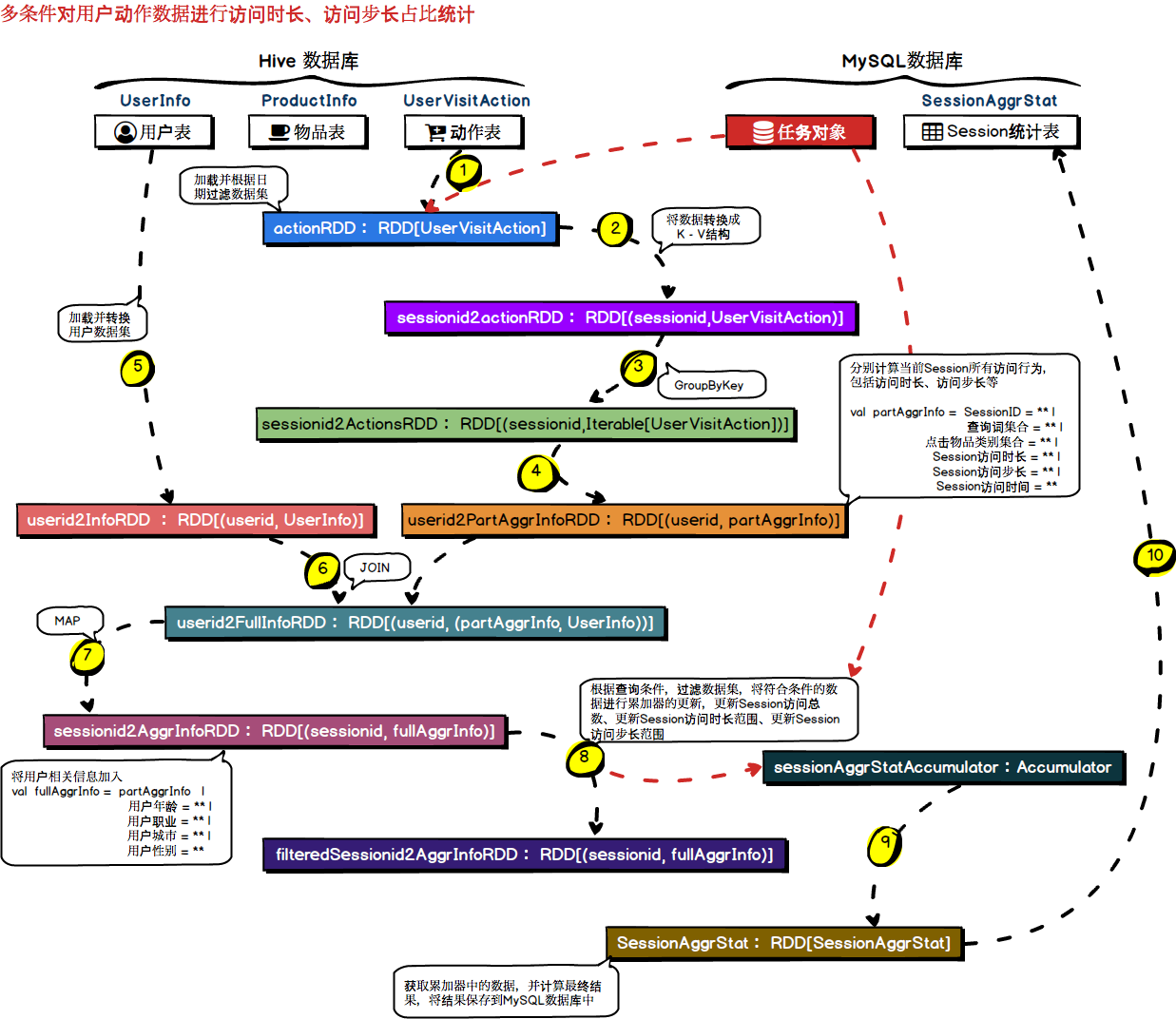
比如说，符合第一步筛选出来的session的数量大概是有1000万个。那么里面，我们要计算出，访问时长在1s~3s内的session的数量，并除以符合条件的总session数量（比如1000万），比如是100万/1000万，那么1s~3s内的session占比就是10%。依次类推，这里说的统计，就是这个意思。

这个功能的作用，其实就是，可以让人从全局的角度看到，符合某些条件的用户群体，使用我们的产品的一些习惯。比如大多数人，到底是会在产品中停留多长时间，大多数人，会在一次使用产品的过程中，访问多少个页面。那么对于使用者来说，有一个全局和清晰的认识。

### 数据结构

|  |
| --- |
| *-- ---------------------------- -- Table structure for `session\_aggr\_stat` -- ----------------------------* **DROP TABLE** IF **EXISTS** `session\_aggr\_stat`; **CREATE TABLE** `session\_aggr\_stat` (  **`taskid` varchar**(255) **DEFAULT NULL**,  **`session\_count` int**(11) **DEFAULT NULL**,  **`visit\_length\_1s\_3s\_ratio` double DEFAULT NULL**,  **`visit\_length\_4s\_6s\_ratio` double DEFAULT NULL**,  **`visit\_length\_7s\_9s\_ratio` double DEFAULT NULL**,  **`visit\_length\_10s\_30s\_ratio` double DEFAULT NULL**,  **`visit\_length\_30s\_60s\_ratio` double DEFAULT NULL**,  **`visit\_length\_1m\_3m\_ratio` double DEFAULT NULL**,  **`visit\_length\_3m\_10m\_ratio` double DEFAULT NULL**,  **`visit\_length\_10m\_30m\_ratio` double DEFAULT NULL**,  **`visit\_length\_30m\_ratio` double DEFAULT NULL**,  **`step\_length\_1\_3\_ratio` double DEFAULT NULL**,  **`step\_length\_4\_6\_ratio` double DEFAULT NULL**,  **`step\_length\_7\_9\_ratio` double DEFAULT NULL**,  **`step\_length\_10\_30\_ratio` double DEFAULT NULL**,  **`step\_length\_30\_60\_ratio` double DEFAULT NULL**,  **`step\_length\_60\_ratio` double DEFAULT NULL**,  **KEY** `idx\_task\_id` (`taskid`) ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8; |

### 数据处理流程



## 需求二：Session随机抽取

### 要做什么？

在符合条件的session中，按照时间比例随机抽取1000个session

这个按照时间比例是什么意思呢？随机抽取本身是很简单的，但是按照时间比例，就很复杂了。比如说，这一天总共有1000万的session。那么我现在总共要从这1000万session中，随机抽取出来1000个session。但是这个随机不是那么简单的。需要做到如下几点要求：首先，如果这一天的12:00~13:00的session数量是100万，那么这个小时的session占比就是1/10，那么这个小时中的100万的session，我们就要抽取1/10 \* 1000 = 100个。然后再从这个小时的100万session中，随机抽取出100个session。以此类推，其他小时的抽取也是这样做。

这个功能的作用，是说，可以让使用者，能够对于符合条件的session，按照时间比例均匀的随机采样出1000个session，然后观察每个session具体的点击流/行为，比如先进入了首页、然后点击了食品品类、然后点击了雨润火腿肠商品、然后搜索了火腿肠罐头的关键词、接着对王中王火腿肠下了订单、最后对订单做了支付。

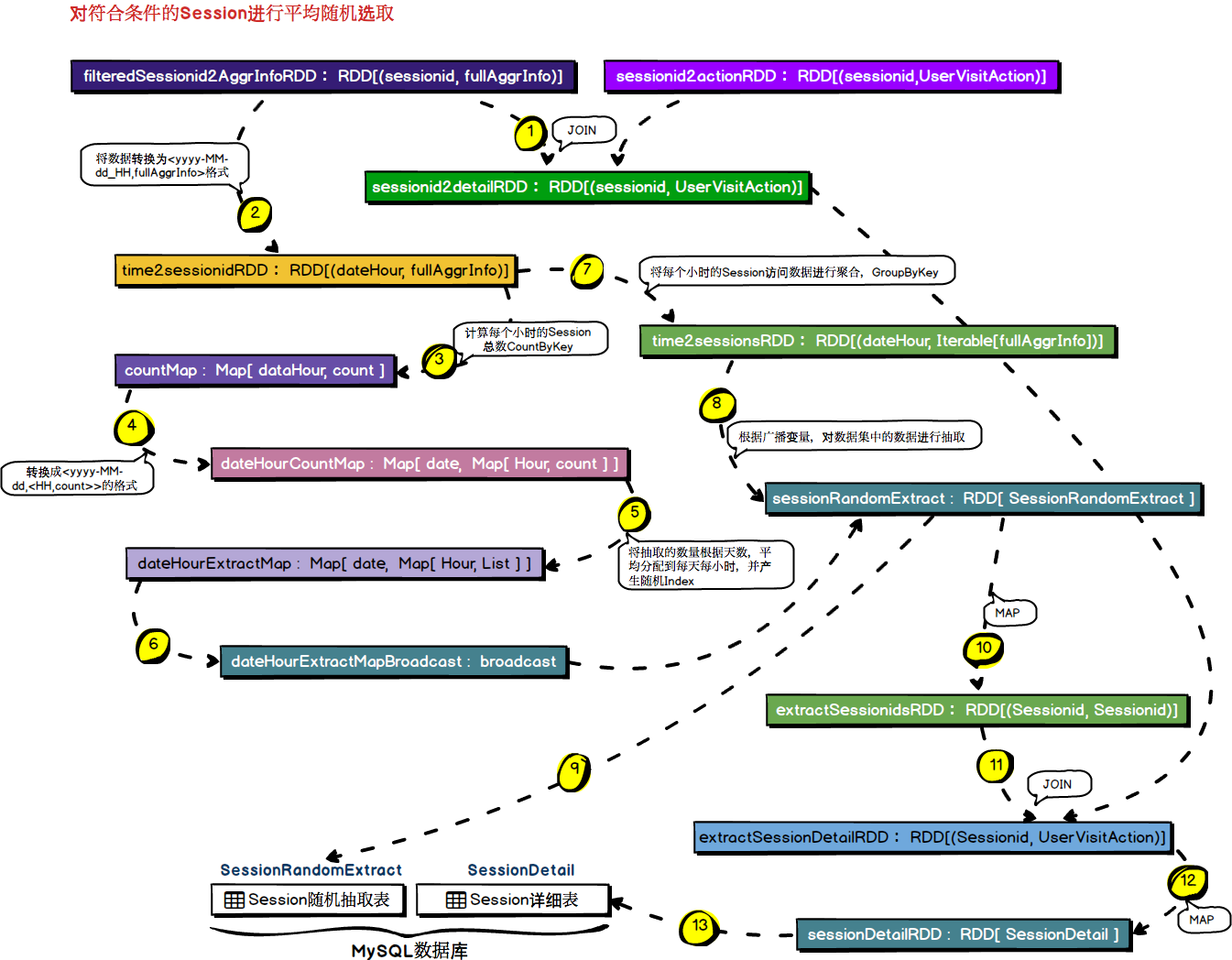
之所以要做到按时间比例随机采用抽取，就是要做到，观察样本的公平性。

抽取完毕之后，需要将Session的相关信息和详细信息保存到MySQL数据库中。

### 数据模型

|  |
| --- |
| *-- ---------------------------- -- Table structure for `session\_detail` -- ----------------------------* **DROP TABLE** IF **EXISTS** `session\_detail`; **CREATE TABLE** `session\_detail` (  **`taskid` varchar**(255) **DEFAULT NULL**,  **`userid` int**(11) **DEFAULT NULL**,  **`sessionid` varchar**(255) **DEFAULT NULL**,  **`pageid` int**(11) **DEFAULT NULL**,  **`actionTime` varchar**(255) **DEFAULT NULL**,  **`searchKeyword` varchar**(255) **DEFAULT NULL**,  **`clickCategoryId` int**(11) **DEFAULT NULL**,  **`clickProductId` int**(11) **DEFAULT NULL**,  **`orderCategoryIds` varchar**(255) **DEFAULT NULL**,  **`orderProductIds` varchar**(255) **DEFAULT NULL**,  **`payCategoryIds` varchar**(255) **DEFAULT NULL**,  **`payProductIds` varchar**(255) **DEFAULT NULL**,  **KEY** `idx\_task\_id` (`taskid`),  **KEY** `idx\_session\_id` (`sessionid`) ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;  *-- ---------------------------- -- Table structure for `session\_random\_extract` -- ----------------------------* **DROP TABLE** IF **EXISTS** `session\_random\_extract`; **CREATE TABLE** `session\_random\_extract` (  **`taskid` varchar**(255) **DEFAULT NULL**,  **`sessionid` varchar**(255) **DEFAULT NULL**,  **`startTime` varchar**(50) **DEFAULT NULL**,  **`searchKeywords` varchar**(255) **DEFAULT NULL**,  **`clickCategoryIds` varchar**(255) **DEFAULT NULL**,  **KEY** `idx\_task\_id` (`taskid`) ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8; |

### 数据处理流程



## 需求三：Top10热门品类

### 要做什么？

在符合条件的session中，获取点击、下单和支付数量排名前10的品类

什么意思呢，对于这些session，每个session可能都会对一些品类的商品进行点击、下单和支付等等行为。那么现在就需要获取这些session点击、下单和支付数量排名前10的最热门的品类。也就是说，要计算出所有这些session对各个品类的点击、下单和支付的次数，然后按照这三个属性进行排序，获取前10个品类。

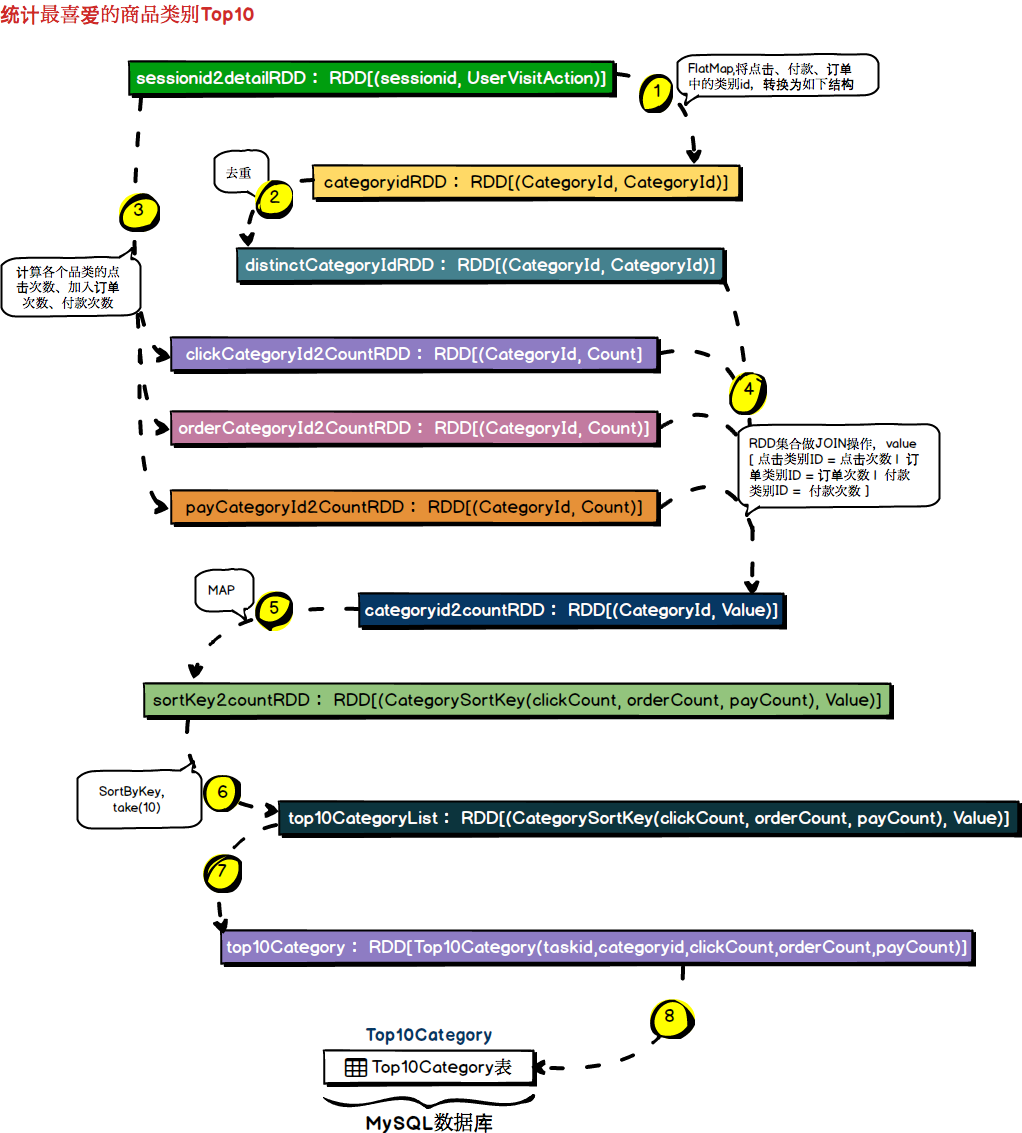
这个功能，很重要，就可以让我们明白，就是符合条件的用户，他最感兴趣的商品是什么种类。这个可以让公司里的人，清晰地了解到不同层次、不同类型的用户的心理和喜好。

计算完成之后，将数据保存到MySQL数据库中。

### 数据模型

|  |
| --- |
| *-- ---------------------------- -- Table structure for `top10\_category` -- ----------------------------* **DROP TABLE** IF **EXISTS** `top10\_category`; **CREATE TABLE** `top10\_category` (  **`taskid` varchar**(255) **DEFAULT NULL**,  **`categoryid` int**(11) **DEFAULT NULL**,  **`clickCount` int**(11) **DEFAULT NULL**,  **`orderCount` int**(11) **DEFAULT NULL**,  **`payCount` int**(11) **DEFAULT NULL**,  **KEY** `idx\_task\_id` (`taskid`) ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8; |

### 数据处理流程



## 需求四：Top10活跃Session

### 要做什么？

对于排名前10的品类，分别获取其点击次数排名前10的session

这个就是说，对于top10的品类，每一个都要获取对它点击次数排名前10的session。

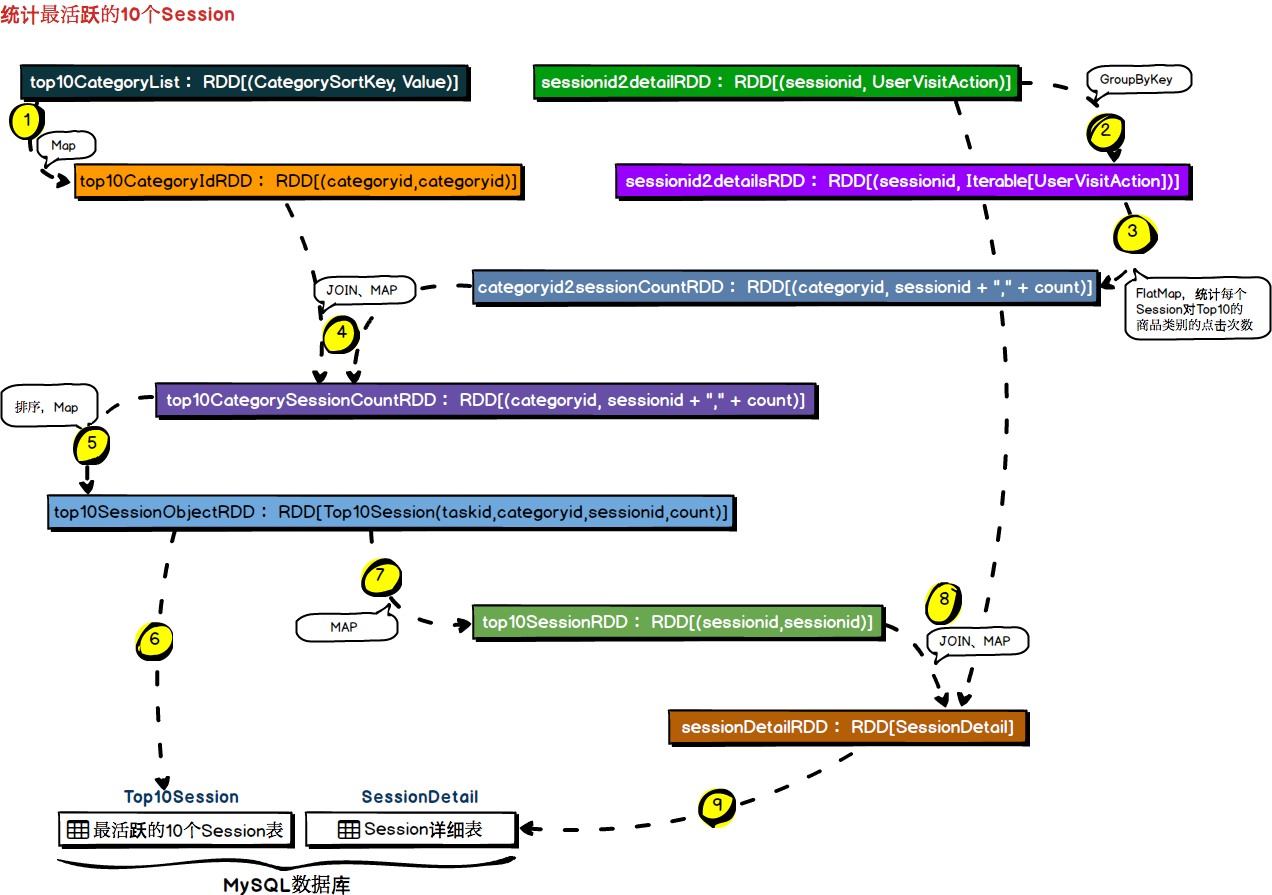
这个功能，可以让我们看到，对某个用户群体最感兴趣的品类，各个品类最感兴趣最典型的用户的session的行为。

计算完成之后，将数据保存到MySQL数据库中。

### 数据模型

|  |
| --- |
| *-- ---------------------------- -- Table structure for `top10\_session` -- ----------------------------* **DROP TABLE** IF **EXISTS** `top10\_session`; **CREATE TABLE** `top10\_session` (  **`taskid` varchar**(255) **DEFAULT NULL**,  **`categoryid` int**(11) **DEFAULT NULL**,  **`sessionid` varchar**(255) **DEFAULT NULL**,  **`clickCount` int**(11) **DEFAULT NULL**,  **KEY** `idx\_task\_id` (`taskid`) ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8; |

### 数据处理流程



# 页面单跳转化率分析

## 模块概述

计算页面单跳转化率，什么是页面单跳转换率，比如一个用户在一次Session过程中访问的页面路径3,5,7,9,10,21，那么页面3调到页面5叫一次单跳，7-9也叫一次单跳，那么单跳转化率就是要统计页面点击的概率，比如：计算3-5的单跳转化率，先获取符合条件的Session对于页面3的访问次数（PV）为A，然后获取符合条件的Session中访问了页面3又紧接着访问了页面5的次数为B，那么B/A 就是3-5的页面单跳转化率，我们记为C；那么页面5-7的转化率怎么求呢？先需要求出符合条件的Session中访问了页面3又紧接着访问了页面5又紧接着访问了页面7的次数为D，那么D/C即为5-7的单跳转化率。

产品经理，可以根据这个指标，去尝试分析，整个网站，产品，各个页面的表现怎么样，是不是需要去优化产品的布局；吸引用户最终可以进入最后的支付页面；

数据分析师，可以基于咱们的这个数据，做更深一步的计算和分析

企业管理层，可以看到整个公司的网站，各个页面的之间的跳转的表现如何？心里有数，可以适当调整公司的经营战略或策略



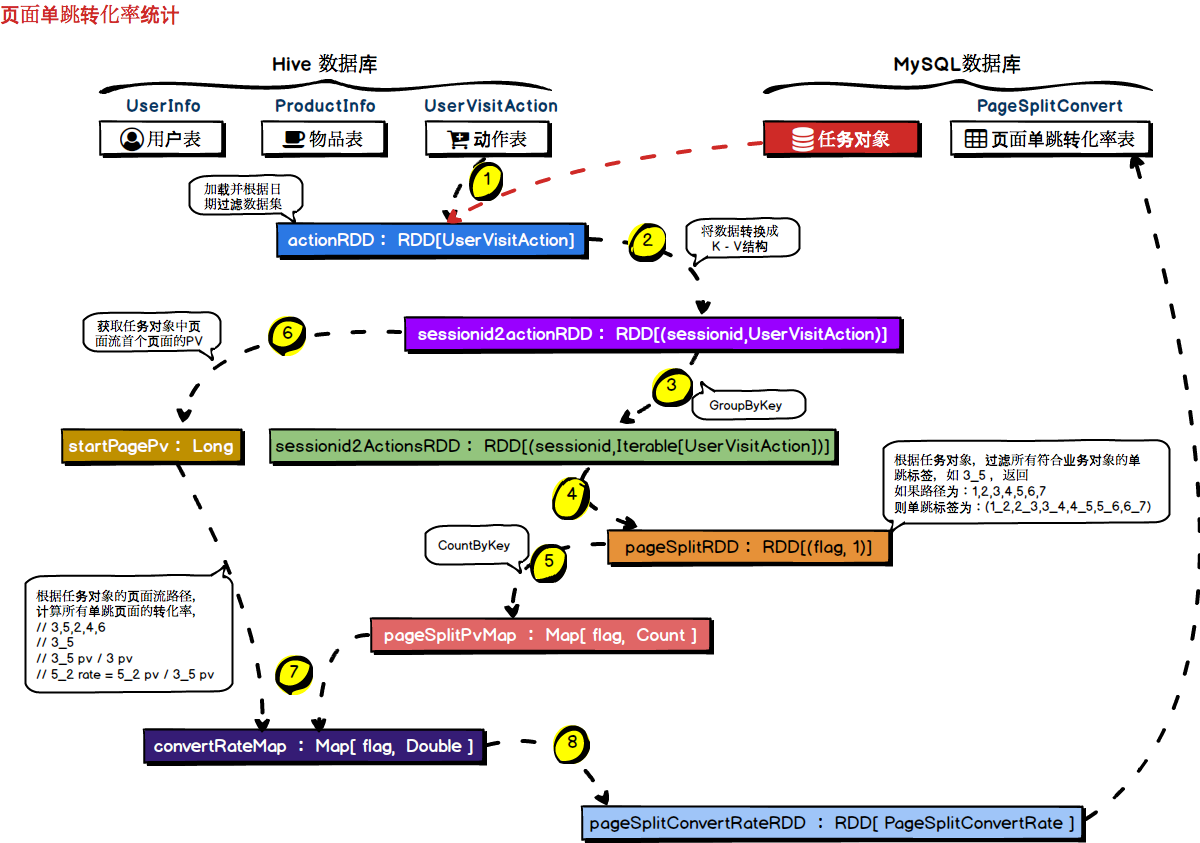
在该模块中，需要根据查询对象中设置的Session过滤条件，先将对应得Session过滤出来，然后根据查询对象中设置的页面路径，计算页面单跳转化率，比如查询的页面路径为：3、5、7、8，那么就要计算3-5、5-7、7-8的页面单跳转化率。需要注意的一点是，页面的访问时有先后的。

## 需求一：页面单跳转化率

### 数据模型

|  |
| --- |
| *-- ---------------------------- -- Table structure for `page\_split\_convert\_rate` -- ----------------------------* **DROP TABLE** IF **EXISTS** `page\_split\_convert\_rate`; **CREATE TABLE** `page\_split\_convert\_rate` (  **`taskid` varchar**(255) **DEFAULT NULL**,  **`convertRate` varchar**(255) **DEFAULT NULL** ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8; |

### 数据处理流程



# 各区域热门商品统计分析

## 模块概述

根据用户指定的日期查询条件范围，统计各个区域下的最热门的top3商品，区域信息、各个城市的信息在项目中用固定值进行配置，因为不怎么变动。

技术方案设计：

1、查询task，获取日期范围，通过Spark SQL，查询user\_visit\_action表中的指定日期范围内的数据，过滤出，商品点击行为，click\_product\_id is not null；click\_product\_id != 'NULL'；click\_product\_id != 'null'；city\_id，click\_product\_id

2、使用Spark SQL从MySQL中查询出来城市信息（city\_id、city\_name、area），用户访问行为数据要跟城市信息进行join，city\_id、city\_name、area、product\_id，RDD，转换成DataFrame，注册成一个临时表

3、Spark SQL内置函数（case when），对area打标记（华东大区，A级，华中大区，B级，东北大区，C级，西北大区，D级），area\_level

4、计算出来每个区域下每个商品的点击次数，group by area, product\_id；保留每个区域的城市名称列表；自定义UDAF，group\_concat\_distinct()函数，聚合出来一个city\_names字段，area、product\_id、city\_names、click\_count

5、join商品明细表，hive（product\_id、product\_name、extend\_info），extend\_info是json类型，自定义UDF，get\_json\_object()函数，取出其中的product\_status字段，if()函数（Spark SQL内置函数），判断，0 自营，1 第三方；（area、product\_id、city\_names、click\_count、product\_name、product\_status）

6、开窗函数，根据area来聚合，获取每个area下，click\_count排名前3的product信息；area、area\_level、product\_id、city\_names、click\_count、product\_name、product\_status

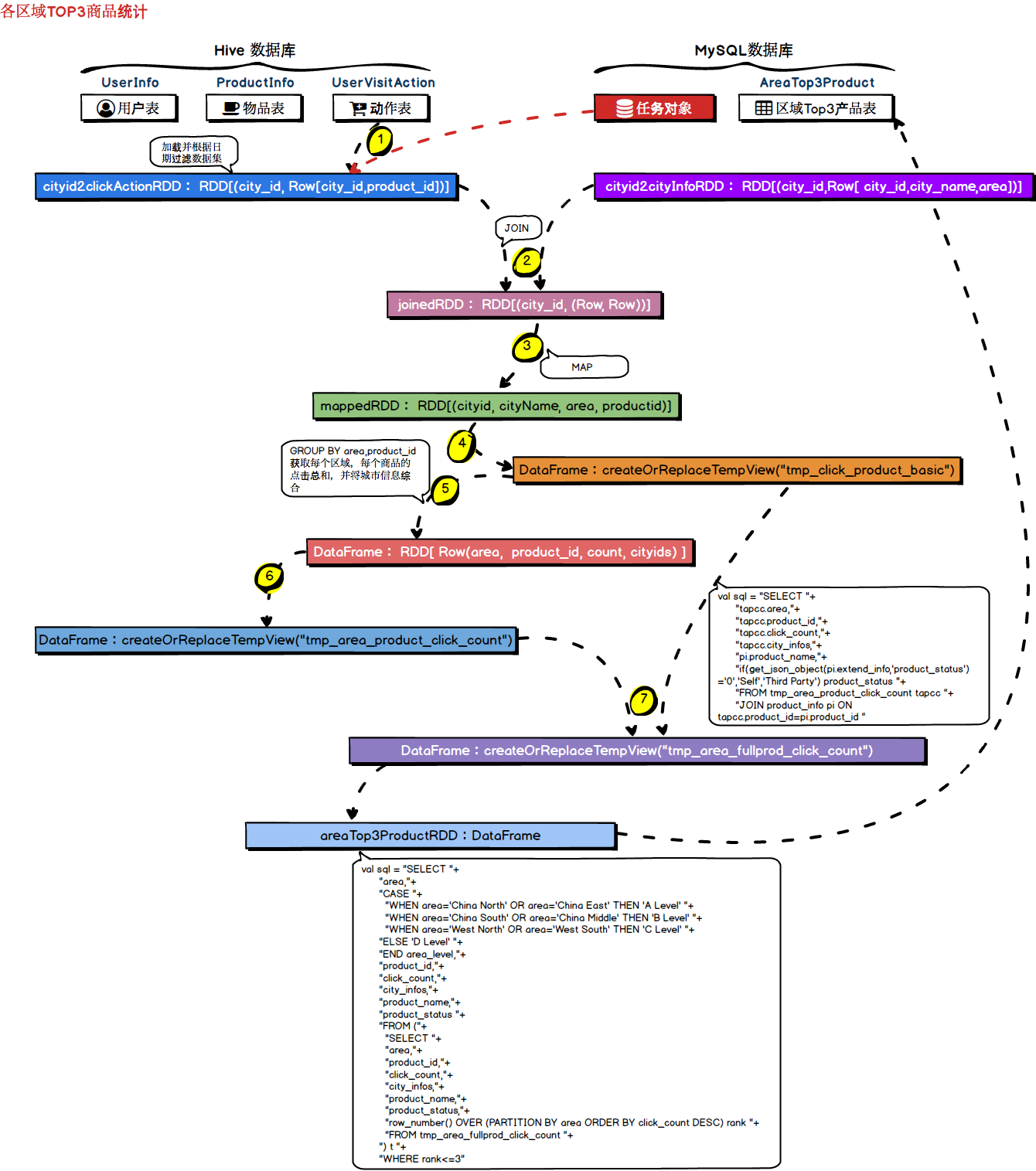
7、结果写入MySQL表中

## 需求一：各区域Top3商品统计

### 数据模型

|  |
| --- |
| *-- ---------------------------- -- Table structure for `area\_top3\_product` -- ----------------------------* **DROP TABLE** IF **EXISTS** `area\_top3\_product`; **CREATE TABLE** `area\_top3\_product` (  **`taskid` varchar**(255) **DEFAULT NULL**,  **`area` varchar**(255) **DEFAULT NULL**,  **`areaLevel` varchar**(255) **DEFAULT NULL**,  **`productid` int**(11) **DEFAULT NULL**,  **`cityInfos` varchar**(255) **DEFAULT NULL**,  **`clickCount` int**(11) **DEFAULT NULL**,  **`productName` varchar**(255) **DEFAULT NULL**,  **`productStatus` varchar**(255) **DEFAULT NULL** ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8; |

### 数据处理流程



# 广告点击流量实时统计分析

## 模块概述

网站/app，通常会给一些第三方的客户，打一些广告；也是一些互联网公司的核心收入来源；广告在网站/app某个广告位打出去，在用户来使用网站/app的时候，广告会显示出来；此时，有些用户可能就会去点击那个广告。

广告被点击以后，实际上，我们就是要针对这种用户行为（广告点击行为），实时数据，进行实时的大数据计算和统计。

每次点击一个广告以后，通常来说，网站/app中都会有埋点（前端的应用中，比如JavaScript Ajax；app中的socket网络请求，往后台发送一条日志数据）；日志数据而言，通常，如果要做实时统计的话，那么就会通过某些方式将数据写入到分布式消息队列中（Kafka）；

日志写入到后台web服务器（nginx），nginx产生的实时的不断增加 / 更新的本地日志文件，就会被日志监控客户端（比如flume agent），写入到消息队列中（kafka），我们要负责编写实时计算程序，去从消息队列中（kafka）去实时地拉取数据，然后对数据进行实时的计算和统计。

这个模块的意义在于，让产品经理、高管可以实时地掌握到公司打的各种广告的投放效果。以便于后期持续地对公司的广告投放相关的战略和策略，进行调整和优化；以期望获得最好的广告收益。

需要实现以下几个需求：

1、实现实时的动态黑名单机制：将每天对某个广告点击超过100次的用户拉黑

2、基于黑名单的非法广告点击流量过滤机制

3、每天各省各城市各广告的点击流量实时统计

4、统计每天各省top3热门广告

5、统计各广告最近1小时内的点击量趋势：各广告最近1小时内各分钟的点击量

6、使用高性能方式将实时统计结果写入MySQL

流式数据的格式为：timestamp province city userid adid

timestamp 1450702800

province Jiangsu

city Nanjing

userid 100001

adid 100001

## 需求一：广告统计用户黑名单

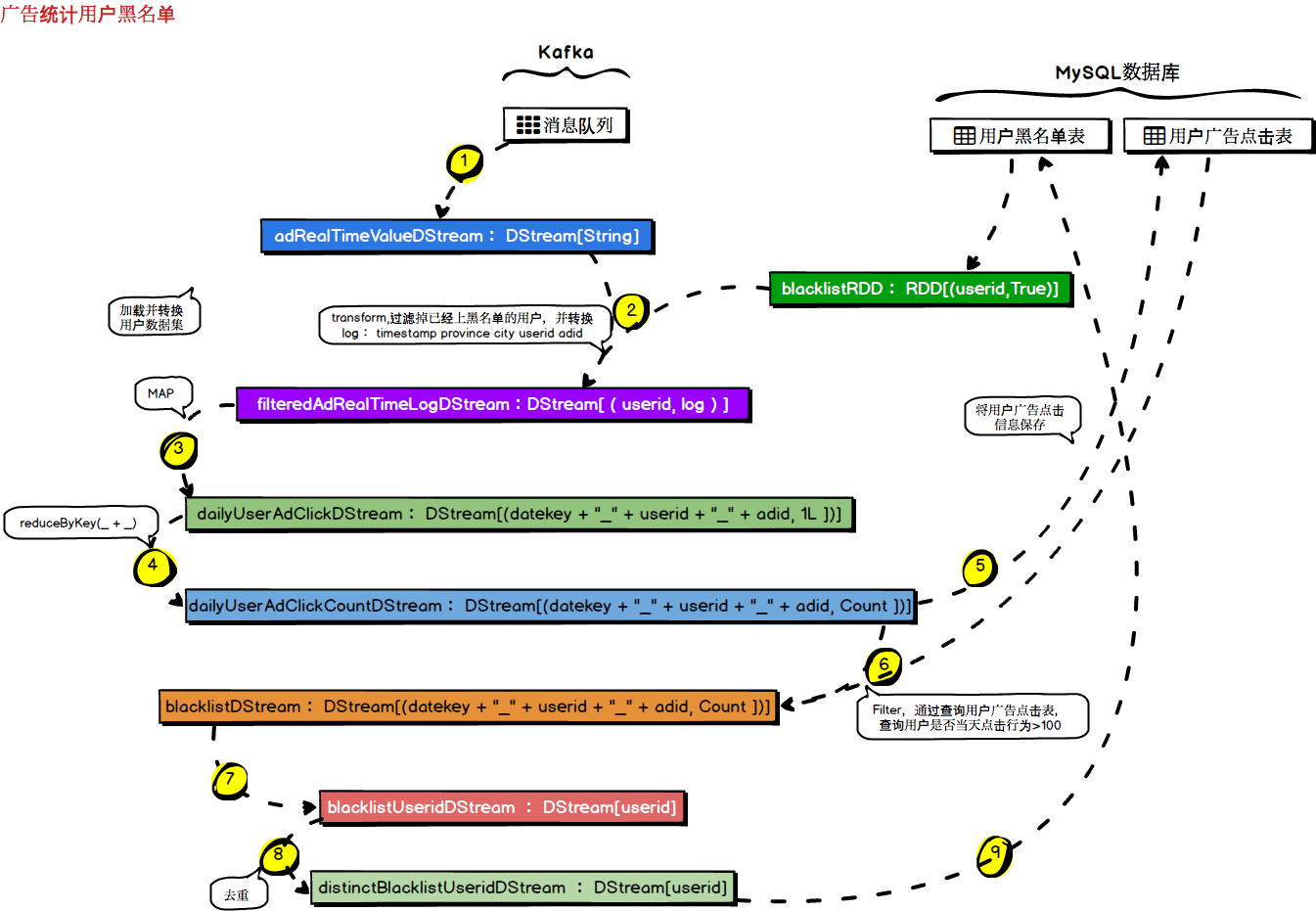
### 要做什么？

实现实时的动态黑名单机制：将每天对某个广告点击超过100次的用户拉黑

### 数据模型

|  |
| --- |
| *-- ---------------------------- -- Table structure for `ad\_blacklist` -- ----------------------------* **DROP TABLE** IF **EXISTS** `ad\_blacklist`; **CREATE TABLE** `ad\_blacklist` (  **`userid` int**(11) **DEFAULT NULL** ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8;  *-- ---------------------------- -- Table structure for `ad\_user\_click\_count` -- ----------------------------* **DROP TABLE** IF **EXISTS** `ad\_user\_click\_count`; **CREATE TABLE** `ad\_user\_click\_count` (  **`date` varchar**(30) **DEFAULT NULL**,  **`userid` int**(11) **DEFAULT NULL**,  **`adid` int**(11) **DEFAULT NULL**,  **`clickCount` int**(11) **DEFAULT NULL** ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8; |

### 数据处理流程



## 需求二：广告点击实时统计

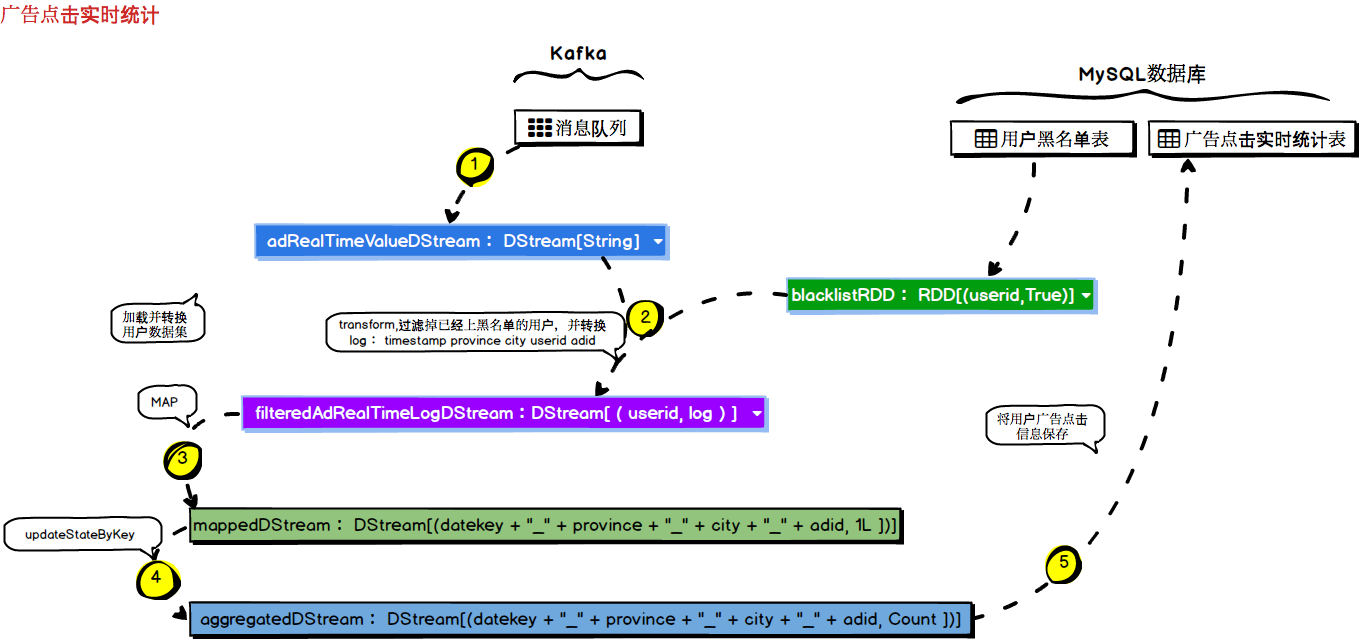
### 要做什么？

每天各省各城市各广告的点击流量实时统计

### 数据模型

|  |
| --- |
| *-- ---------------------------- -- Table structure for `ad\_stat` -- ----------------------------* **DROP TABLE** IF **EXISTS** `ad\_stat`; **CREATE TABLE** `ad\_stat` (  **`date` varchar**(30) **DEFAULT NULL**,  **`province` varchar**(100) **DEFAULT NULL**,  **`city` varchar**(100) **DEFAULT NULL**,  **`adid` int**(11) **DEFAULT NULL**,  **`clickCount` int**(11) **DEFAULT NULL** ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8; |

### 数据处理流程



## 需求三：实时统计各省最热广告

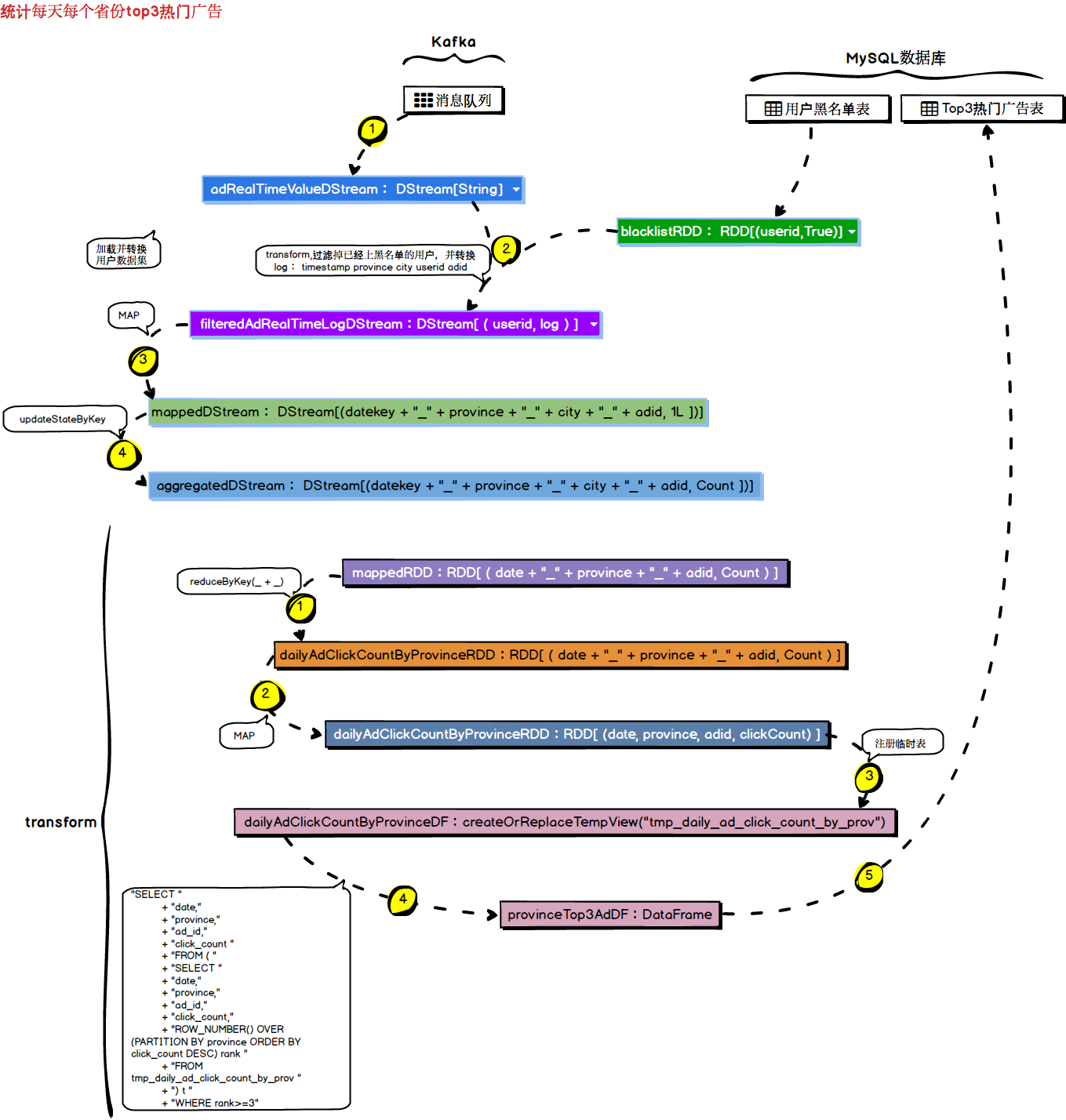
### 要做什么？

统计每天各省top3热门广告

### 数据模型

|  |
| --- |
| *-- ---------------------------- -- Table structure for `ad\_province\_top3` -- ----------------------------* **DROP TABLE** IF **EXISTS** `ad\_province\_top3`; **CREATE TABLE** `ad\_province\_top3` (  **`date` varchar**(30) **DEFAULT NULL**,  **`province` varchar**(100) **DEFAULT NULL**,  **`adid` int**(11) **DEFAULT NULL**,  **`clickCount` int**(11) **DEFAULT NULL** ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8; |

### 数据处理流程



## 需求四：最近一小时广告点击趋势统计

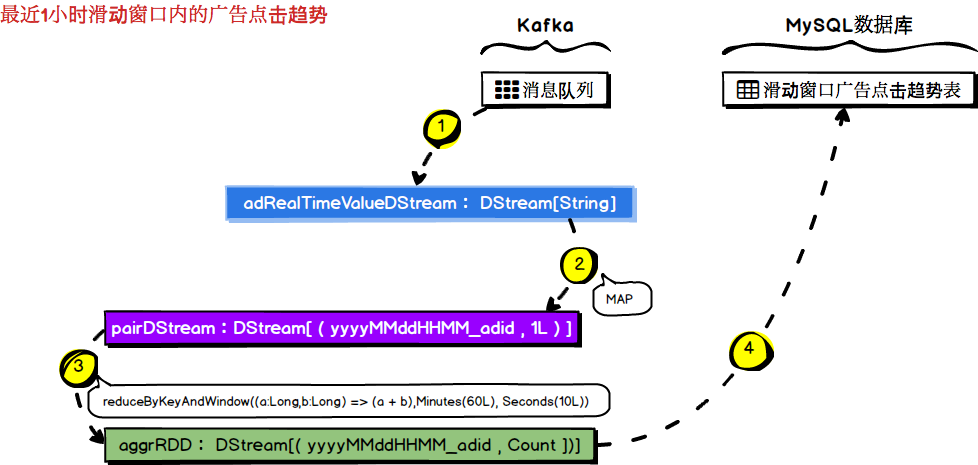
### 要做什么？

统计各广告最近1小时内的点击量趋势：各广告最近1小时内各分钟的点击量

### 数据模型

|  |
| --- |
| *-- ---------------------------- -- Table structure for `ad\_click\_trend` -- ----------------------------* **DROP TABLE** IF **EXISTS** `ad\_click\_trend`; **CREATE TABLE** `ad\_click\_trend` (  **`date` varchar**(30) **DEFAULT NULL**,  **`hour` varchar**(30) **DEFAULT NULL**,  **`minute` varchar**(30) **DEFAULT NULL**,  **`adid` int**(11) **DEFAULT NULL**,  **`clickCount` int**(11) **DEFAULT NULL** ) ENGINE=InnoDB **DEFAULT** CHARSET=utf8; |

### 数据处理流程



# 大数据部署建设（选做）

## 通过azkaban将业务进行部署

## 通过Zeppelin将MySQL中的数据进行图表输出