**基于大数据平台的电商系统搭建**

**2020年5月**

# 目录

[1 项目概述 3](#_Toc39272580)

[2 建设目标及原则 3](#_Toc39272581)

[2.1 建设目标 3](#_Toc39272582)

[2.2 建设原则 3](#_Toc39272583)

[3 需求分析 4](#_Toc39272584)

[3.1 概述 4](#_Toc39272585)

[3.2 功能需求 4](#_Toc39272586)

[3.2.1 GMV成交总额 4](#_Toc39272587)

[4 概要设计 4](#_Toc39272588)

[4.1 概述 4](#_Toc39272589)

[4.2 系统框架设计 4](#_Toc39272590)

[4.2.1 技术架构设计 4](#_Toc39272591)

[4.2.2 软件功能结构设计 5](#_Toc39272592)

[4.3 应用系统设计 5](#_Toc39272593)

[4.3.1 设计原则 5](#_Toc39272594)

[4.3.2 功能模块设计 5](#_Toc39272595)

[4.4 接口设计 5](#_Toc39272596)

[4.4.1 外部接口 5](#_Toc39272597)

[4.4.2 内部接口 5](#_Toc39272598)

[5 详细设计 5](#_Toc39272599)

[5.1 系统概述 5](#_Toc39272600)

[5.2 系统数据流程 6](#_Toc39272601)

[5.3 功能设计 6](#_Toc39272602)

[5.3.1 GMV成交总额 6](#_Toc39272603)

[6 系统关键技术 6](#_Toc39272604)

[6.1 Hadoop平台 6](#_Toc39272605)

1. **项目概述**

现如今，随着人们对互联网行业的深入研究，互联网技术在近几年得到了极大的普及，电商网站的产生更是让互联网逐步成为人们日常生活中必不可少的一部分，它便利了人们的生活，提高了人们的生活品质。

随着电商网站日益增长的用户体验，数据量增长势头也是只增不减，对这些海量数据进行加工处理使其成为有用的数据成为了一个巨大的挑战，但同时也带来了巨大的技术创新和商业机遇。正确的数据分析结果可以帮助网站能对自身得到更全新的认识，然后对网站的不足之处做出正确的判断，及时纠正，为网站创造更多的收益价值。

1. **建设目标及原则**
   1. **建设目标**

本系统是基于Hadoop框架，围绕Nginx服务器、MySQL数据库、Hive数据仓库、HighCharts图库表、JavaScript脚本语言、CSS样式表等众多前后端技术，实现电商网站用户日志数据的解析处理，最终生成一系列的数据指标，并对此进行可视化展示，为电商网站的改进提供一定的参考。

总之，该系统建设的总体目标是：

* 对电商日志数据进行处理；
* 分析和挖掘电商日志数据信息；
* 为电商网站带来巨大的商业价值；
* 提高电商网站的核心竞争力。
  1. **建设原则**

**开放性原则：**

运用开放的标准规范整合各类系统，为实现系统信息交换和共享提供标准接口。

**集约性原则：**

充分利用已有建设成果，辅以必要的补充完善，逐步提升系统智慧化水平。

**复用性原则：**

采用模块化和平台化研制方法构建应用服务体系，提高软件复用和重用能力，方便建设成果推广。

1. **需求分析**
   1. **概述**
   2. **功能需求**
      1. **GMV成交总额**

GMV（Gross Merchandise Volume），是一定时间段内的成交总额（比如一天、一个月、一年），包括付款和未付款的部分。

* + 1. **用户活跃指标**

一个产品的根本是用户，分析用户的各项活跃指标，可以从根本上反应一个产品的好坏以及质量变化，从而为产品开发和运营提供精确的方向。

电商系统是一个面向用户的产品，是一个服务用户的平台，卖家和买家都是平台的用户，用户带来的流量大小就代表着一个电商系统是否成功。

电商平台每时每刻都会生成用户行为日志，在各种优惠活动期间更是会生成大量的日志，数据仓库通过清洗、抽取、转换用户的应用启动日志，能够整理出用户的活跃度、留存率、新增数量等指标，从不同的角度分析用户的行为，反应不同时间段的产品质量，清晰的表现出产品升级或优惠活动的最终效果如何，比如新增了多少用户，留存了多少用户，有多少用户转换为了忠诚用户等等，依据这些数据，产品研发运营可以抓住用户的核心需求，找到痛点，提升电商平台的核心竞争力。

1. **概要设计**
   1. **概述**
   2. **系统框架设计**

* + 1. **技术架构设计**

系统主要由三部分组成，分别为数据采集传输模块、数据存储计算模块和数据展示模块。

数据采集传输模块：主要由nginx、Flume、Sqoop等工具组成，负责将用户操作的各项数据进行收集传输。

数据存储计算模块：主要由Hadoop框架组成，负责过滤、解析上一步收集到的数据信息，并保存计算后的结果。

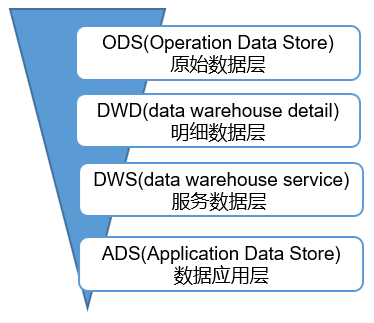
数据展示模块：主要由JavaScript、HTML、CSS组成，负责对平台展示界面的编写以及调取数据api接口，将数据分析结果进行展示。

* + 1. **软件功能结构设计**

系统按照功能可以分为

* 1. **应用**系统设计
     1. **设计原则**
        1. 数据仓库分层

依据业务需求将数据仓库分层为ODS层、DWD层、DWS层和ADS层。

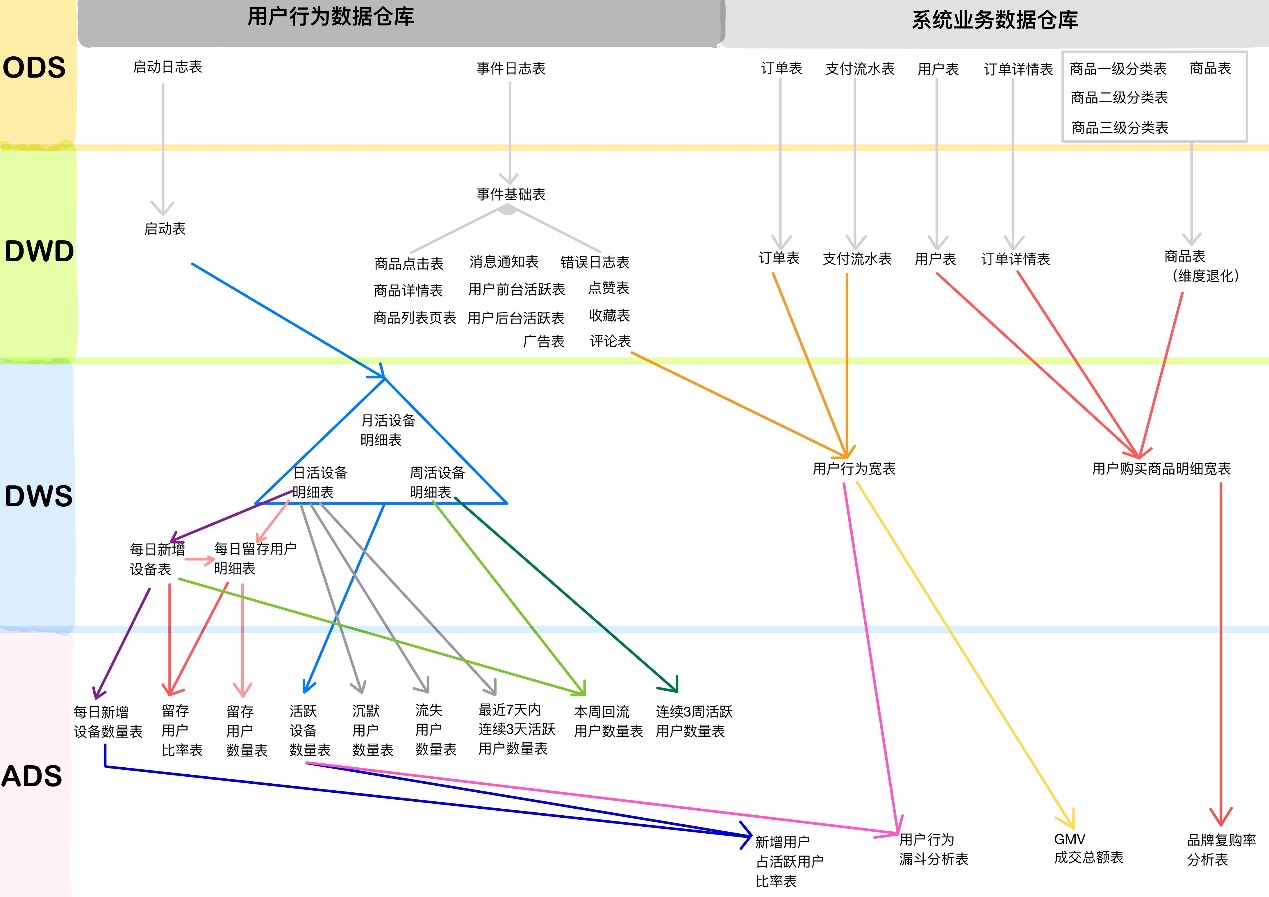


1. ODS层：原始数据层，存放原始数据，直接加载原始日志、数据，数据保持原貌不做处理。
2. DWD层：结构和粒度与原始表保持一致，对ODS层数据进行清洗（去除空值、脏数据、超过极限范围的数据）。
3. DWS层：以DWD层为基础，进行轻度汇总。
4. ADS层：为各种统计报表提供数据

数据仓库通过分层，把复杂问题简单化，将一个复杂的任务分解成多个步骤来完成，每一层只处理单一的步骤，比较简单，容易定位问题；减少重复开发，规范数据分层，通过中间层数据，能够减少大量的重复计算，增加一次计算的可复用性；隔离原始数据，不论是数据的异常还是敏感性，使真实数据与统计数据。

* + 1. **功能模块设计**
       1. 数据仓库之用户行为

对于用户行为在数据仓库中的清洗、抽取、转换、加载过程的分层结构如下图所示。



* + - * 1. 用户活跃分析

根据行业内对活跃用户的普遍定义，用户打开应用一次记为一个活跃用户。通过数据埋点，收集用户启动行为生成用户启动日志，可以跟踪用户的活跃情况。

根据用户的应用使用情况，将用户活跃统计分为用户活跃主题（日活、周活、月活）、用户新增主题、用户留存主题、沉默用户数、本周回流用户数、流失用户数、最近连续三周中用户数、最近七天内连续三天活跃用户数共8个指标。

1. 用户活跃主题：包括每日活跃设备明细、每周活跃设备明细、每月活跃设备明细，统计当日、当周、当月活动的设备情况。
2. 新增用户主题：指首次连网使用应用的用户。如果一个目标用户首次打开应用，那这个用户定义为新增用户；如果用户卸载后再安装应用，则不计入新增用户。新增用户包括日新增用户、周新增用户、月新增用户。
3. 用户留存主题：某段时间内的新增用户（活跃用户），经过一段时间后，还继续使用应用的视为留存用户。其中，留存用户占当时新增用户（活跃用户）的比例即是留存率。
4. 沉默用户数：指的是只在安装当天启动过应用，且启动时间是在一周前。
5. 本周回流用户数：等于本周活跃-本周新增-上周活跃
6. 流失用户数：最近7天未登录应用的称为流失用户
7. 最近连续三周活跃用户数：通常是周一对前三周的数据作统计，该数据一周计算一次。
8. 最近七天内连续三天活跃用户数。
   1. **接口设计**
      1. **外部接口**
      2. **内部接口**
9. **详细设计**
   1. **系统概述**

面向服务的体系结构（SOA）是一个组件模型，具备敏捷性、重用性和低耦合性等特点，使得它能很好地应用于智慧营区“多维立体智能管控平台”，它将应用程序的不同功能单，通过这些服务之间定义良好的接口和契约联系起来。基于SOA架构技术的多维立体智能管控平台划分为感知层、传输层、服务层和应用层构成，通过整合各个服务层，对多维立体智能管控平台的核心业务的变化做出快速反应，呈现出可以支持有机业务框架的能力

在本系统中，基础接入子系统包括卡口子系统、共享单车子系统、摆渡车子系统、绿化灌溉子系统、停车引导子系统、微信公众号和手机APP、公共信息显示子系统、公共区域无线接入子系统、背景音乐子系统。

智慧服务系统通过人、车、物、环境有机的结合在一起，彼此之间数据互通，构建一套从惠民需求到惠民服务的一整套服务体系。

* 1. **系统数据流程**
     1. **用户活跃分析数据流程**



* 1. **功能设计**
     1. **GMV成交总额**
        1. 系统设计
     2. **用户活跃分析**
        1. ODS层
           1. 启动日志表ods\_start\_log

1. 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| action | 状态：成功=1，失败=2 |
| ar | 区域 |
| ba | 手机品牌 |
| detail  en  entry  extend1  g  hw  l  la  ln  loading\_time  md  mid  nw  open\_ad\_type  os  sr  sv  t  uid  vc  vn | 失败码（没有则上报空）  日志类型start（事件类型）  入口：push=1, widget=2, icon=3, notification=4, lockscreen\_widget=5  失败message（没有则上报为空）  gmail  屏幕宽高  系统语言  经度  纬度  加载时长：计算下拉开始到接口返回数据的时间  手机型号  设备唯一标识  网络模式  开屏广告类型：开屏原生广告=1，开屏插屏广告=2  Android系统版本  渠道号，应用从哪个渠道来的  sdk version  客户端日志产生时的时间  用户标识  程序版本号  程序版本名 |

1. 建表语句

hive (gmall)>

drop table if exists gmall.ods\_start\_log;

CREATE EXTERNAL TABLE gmall.ods\_start\_log (`line` string)

PARTITIONED BY (`dt` string)

STORED AS

INPUTFORMAT 'com.hadoop.mapred.DeprecatedLzoTextInputFormat'

OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'

LOCATION '/warehouse/gmall/ods/ods\_start\_log';

* + - 1. DWD层
         1. 启动日志表

1. 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| mid\_id | 设备唯一标识 |
| user\_id | 用户标识 |
| version\_code | 程序版本号 |
| version\_name  lang  source  os  area  model  brand  sdk\_version  gmail  height\_width  app\_time  network  lng  lat  entry  open\_ad\_type  action  loading\_time  detail  extend1 | 程序版本名  系统语言  渠道号，应用从哪个渠道来的  Android系统版本  区域  手机型号  手机品牌  sdk version  gmail  屏幕宽高  客户端日志产生时的时间  网络模式  经度  纬度  入口：push=1, widget=2, icon=3, notification=4, lockscreen\_widget=5  开屏广告类型：开屏原生广告=1，开屏插屏广告=2  状态：成功=1，失败=2  加载时长：计算下拉开始到接口返回数据的时间  失败码（没有则上报空）  失败message（没有则上报为空） |

1. 建表语句：

hive (gmall)>

drop table if exists gmall.dwd\_start\_log;

CREATE EXTERNAL TABLE gmall.dwd\_start\_log(

`mid\_id` string,

`user\_id` string,

`version\_code` string,

`version\_name` string,

`lang` string,

`source` string,

`os` string,

`area` string,

`model` string,

`brand` string,

`sdk\_version` string,

`gmail` string,

`height\_width` string,

`app\_time` string,

`network` string,

`lng` string,

`lat` string,

`entry` string,

`open\_ad\_type` string,

`action` string,

`loading\_time` string,

`detail` string,

`extend1` string

)

PARTITIONED BY (dt string)

location '/warehouse/gmall/dwd/dwd\_start\_log/';

* + - 1. DWS层
         1. 用户活跃主题

1. 每日活跃设备明细dws\_uv\_detail\_day

a) 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| mid\_id | 设备唯一标识 |
| user\_id | 用户标识 |
| version\_code | 程序版本号 |
| version\_name  lang  source  os  area  model  brand  sdk\_version  gmail  height\_width  app\_time  network  lng  lat | 程序版本名  系统语言  渠道号，应用从哪个渠道来的  Android系统版本  区域  手机型号  手机品牌  sdk version  gmail  屏幕宽高  客户端日志产生时的时间  网络模式  经度  纬度 |

b) 建表语句：

hive (gmall)>

drop table if exists gmall.dws\_uv\_detail\_day;

create external table gmall.dws\_uv\_detail\_day

(

`mid\_id` string COMMENT '设备唯一标识',

`user\_id` string COMMENT '用户标识',

`version\_code` string COMMENT '程序版本号',

`version\_name` string COMMENT '程序版本名',

`lang` string COMMENT '系统语言',

`source` string COMMENT '渠道号',

`os` string COMMENT '安卓系统版本',

`area` string COMMENT '区域',

`model` string COMMENT '手机型号',

`brand` string COMMENT '手机品牌',

`sdk\_version` string COMMENT 'sdkVersion',

`gmail` string COMMENT 'gmail',

`height\_width` string COMMENT '屏幕宽高',

`app\_time` string COMMENT '客户端日志产生时的时间',

`network` string COMMENT '网络模式',

`lng` string COMMENT '经度',

`lat` string COMMENT '纬度'

)

partitioned by(dt string)

stored as parquet

location '/warehouse/gmall/dws/dws\_uv\_detail\_day'

;

1. 第周活跃设备明细dws\_uv\_detail\_wk

a) 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| mid\_id | 设备唯一标识 |
| user\_id | 用户标识 |
| version\_code | 程序版本号 |
| version\_name  lang  source  os  area  model  brand  sdk\_version  gmail  height\_width  app\_time  network  lng  lat  sonday\_date  sunday\_date | 程序版本名  系统语言  渠道号，应用从哪个渠道来的  Android系统版本  区域  手机型号  手机品牌  sdk version  gmail  屏幕宽高  客户端日志产生时的时间  网络模式  经度  纬度  周一日期  周日日期 |

b) 建表语句：

hive (gmall)>

drop table if exists gmall.dws\_uv\_detail\_wk;

create external table gmall.dws\_uv\_detail\_wk(

`mid\_id` string COMMENT '设备唯一标识',

`user\_id` string COMMENT '用户标识',

`version\_code` string COMMENT '程序版本号',

`version\_name` string COMMENT '程序版本名',

`lang` string COMMENT '系统语言',

`source` string COMMENT '渠道号',

`os` string COMMENT '安卓系统版本',

`area` string COMMENT '区域',

`model` string COMMENT '手机型号',

`brand` string COMMENT '手机品牌',

`sdk\_version` string COMMENT 'sdkVersion',

`gmail` string COMMENT 'gmail',

`height\_width` string COMMENT '屏幕宽高',

`app\_time` string COMMENT '客户端日志产生时的时间',

`network` string COMMENT '网络模式',

`lng` string COMMENT '经度',

`lat` string COMMENT '纬度',

`monday\_date` string COMMENT '周一日期',

`sunday\_date` string COMMENT '周日日期'

) COMMENT '活跃用户按周明细'

PARTITIONED BY (`wk\_dt` string)

stored as parquet

location '/warehouse/gmall/dws/dws\_uv\_detail\_wk/'

;

1. 每月活跃设备明细dws\_uv\_detail\_mn

a) 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| mid\_id | 设备唯一标识 |
| user\_id | 用户标识 |
| version\_code | 程序版本号 |
| version\_name  lang  source  os  area  model  brand  sdk\_version  gmail  height\_width  app\_time  network  lng  lat | 程序版本名  系统语言  渠道号，应用从哪个渠道来的  Android系统版本  区域  手机型号  手机品牌  sdk version  gmail  屏幕宽高  客户端日志产生时的时间  网络模式  经度  纬度 |

b) 建表语句：

hive (gmall)>

drop table if exists gmall.dws\_uv\_detail\_mn;

create external table gmall.dws\_uv\_detail\_mn(

`mid\_id` string COMMENT '设备唯一标识',

`user\_id` string COMMENT '用户标识',

`version\_code` string COMMENT '程序版本号',

`version\_name` string COMMENT '程序版本名',

`lang` string COMMENT '系统语言',

`source` string COMMENT '渠道号',

`os` string COMMENT '安卓系统版本',

`area` string COMMENT '区域',

`model` string COMMENT '手机型号',

`brand` string COMMENT '手机品牌',

`sdk\_version` string COMMENT 'sdkVersion',

`gmail` string COMMENT 'gmail',

`height\_width` string COMMENT '屏幕宽高',

`app\_time` string COMMENT '客户端日志产生时的时间',

`network` string COMMENT '网络模式',

`lng` string COMMENT '经度',

`lat` string COMMENT '纬度'

) COMMENT '活跃用户按月明细'

PARTITIONED BY (`mn` string)

stored as parquet

location '/warehouse/gmall/dws/dws\_uv\_detail\_mn/'

;

* + - * 1. 用户新增主题

1. 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| mid\_id | 设备唯一标识 |
| user\_id | 用户标识 |
| version\_code | 程序版本号 |
| version\_name  lang  source  os  area  model  brand  sdk\_version  gmail  height\_width  app\_time  network  lng  lat  create\_date | 程序版本名  系统语言  渠道号，应用从哪个渠道来的  Android系统版本  区域  手机型号  手机品牌  sdk version  gmail  屏幕宽高  客户端日志产生时的时间  网络模式  经度  纬度  创建时间 |

1. 建表语句：

hive (gmall)>

drop table if exists gmall.dws\_new\_mid\_day;

create external table gmall.dws\_new\_mid\_day

(

`mid\_id` string COMMENT '设备唯一标识',

`user\_id` string COMMENT '用户标识',

`version\_code` string COMMENT '程序版本号',

`version\_name` string COMMENT '程序版本名',

`lang` string COMMENT '系统语言',

`source` string COMMENT '渠道号',

`os` string COMMENT '安卓系统版本',

`area` string COMMENT '区域',

`model` string COMMENT '手机型号',

`brand` string COMMENT '手机品牌',

`sdk\_version` string COMMENT 'sdkVersion',

`gmail` string COMMENT 'gmail',

`height\_width` string COMMENT '屏幕宽高',

`app\_time` string COMMENT '客户端日志产生时的时间',

`network` string COMMENT '网络模式',

`lng` string COMMENT '经度',

`lat` string COMMENT '纬度',

`create\_date` string comment '创建时间'

) COMMENT '每日新增设备信息'

stored as parquet

location '/warehouse/gmall/dws/dws\_new\_mid\_day/';

* + - * 1. 用户留存主题

1. 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| mid\_id | 设备唯一标识 |
| user\_id | 用户标识 |
| version\_code | 程序版本号 |
| version\_name  lang  source  os  area  model  brand  sdk\_version  gmail  height\_width  app\_time  network  lng  lat  create\_date  retention\_day | 程序版本名  系统语言  渠道号，应用从哪个渠道来的  Android系统版本  区域  手机型号  手机品牌  sdk version  gmail  屏幕宽高  客户端日志产生时的时间  网络模式  经度  纬度  设备新增  截止当前日期留存天数 |

1. 建表语句：

hive (gmall)>

drop table if exists gmall.dws\_user\_retention\_day;

create external table gmall.dws\_user\_retention\_day

(

`mid\_id` string COMMENT '设备唯一标识',

`user\_id` string COMMENT '用户标识',

`version\_code` string COMMENT '程序版本号',

`version\_name` string COMMENT '程序版本名',

`lang` string COMMENT '系统语言',

`source` string COMMENT '渠道号',

`os` string COMMENT '安卓系统版本',

`area` string COMMENT '区域',

`model` string COMMENT '手机型号',

`brand` string COMMENT '手机品牌',

`sdk\_version` string COMMENT 'sdkVersion',

`gmail` string COMMENT 'gmail',

`height\_width` string COMMENT '屏幕宽高',

`app\_time` string COMMENT '客户端日志产生时的时间',

`network` string COMMENT '网络模式',

`lng` string COMMENT '经度',

`lat` string COMMENT '纬度',

`create\_date` string comment '设备新增时间',

`retention\_day` int comment '截止当前日期留存天数'

) COMMENT '每日用户留存情况'

PARTITIONED BY (`dt` string)

stored as parquet

location '/warehouse/gmall/dws/dws\_user\_retention\_day/'

;

* + - * 1. 沉默用户数

使用日活明细表dws\_uv\_detail\_day作为DWS层数据

* + - * 1. 本周回流用户数

使用日活明细表dws\_uv\_detail\_day作为DWS层数据

* + - * 1. 流失用户数

使用日活明细表dws\_uv\_detail\_day作为DWS层数据

* + - * 1. 最近连续三周活跃用户数

使用周活明细表dws\_uv\_detail\_day作为DWS层数据

* + - * 1. 最近七天连续三天活跃用户数

使用日活明细表dws\_uv\_detail\_day作为DWS层数据

* + - 1. ADS层
         1. 用户活跃主题

1. 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| dt | 统计日期 |
| day\_count | 当日用户数量 |
| wk\_count  mn\_count | 当周用户数量  当月用户数量 |
| is\_weekend  is\_monthend | Y,N是否是周末，用于得到本周最终结果  Y,N是否是月末，用于得到本月最终结果 |

1. 建表语句：

hive (gmall)>

drop table if exists gmall.ads\_uv\_count;

create external table gmall.ads\_uv\_count(

`dt` string COMMENT '统计日期',

`day\_count` bigint COMMENT '当日用户数量',

`wk\_count` bigint COMMENT '当周用户数量',

`mn\_count` bigint COMMENT '当月用户数量',

`is\_weekend` string COMMENT 'Y,N是否是周末,用于得到本周最终结果',

`is\_monthend` string COMMENT 'Y,N是否是月末,用于得到本月最终结果'

) COMMENT '活跃设备数'

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_uv\_count/'

;

* + - * 1. 用户新增主题

1. 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| create\_date  new\_mid\_count | 创建时间  新增设备数量 |

1. 建表语句：

hive (gmall)>

drop table if exists gmall.ads\_new\_mid\_count;

create external table gmall.ads\_new\_mid\_count

(

`create\_date` string comment '创建时间' ,

`new\_mid\_count` BIGINT comment '新增设备数量'

) COMMENT '每日新增设备信息数量'

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_new\_mid\_count/';

* + - * 1. 用户留存主题

1. 用户留存数

a) 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| create\_date  retention\_day  retention\_count | 设备新增日期  截止当前日期留存天数  留存数量 |

b) 建表语句：

hive (gmall)>

drop table if exists gmall.ads\_user\_retention\_day\_count;

create external table gmall.ads\_user\_retention\_day\_count

(

`create\_date` string comment '设备新增日期',

`retention\_day` int comment '截止当前日期留存天数',

`retention\_count` bigint comment '留存数量'

) COMMENT '每日用户留存情况'

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_user\_retention\_day\_count/';

1. 留存用户比率

a) 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| stat\_date  create\_date  retention\_day  retention\_count  new\_mid\_count  retention\_ratio | 统计日期  设备新增日期  截止当前日期留存天数  留存数量  当日设备新增数量  留存率 |

b) 建表语句：

hive (gmall)>

drop table if exists gmall.ads\_user\_retention\_day\_rate;

create external table gmall.ads\_user\_retention\_day\_rate

(

`stat\_date` string comment '统计日期',

`create\_date` string comment '设备新增日期',

`retention\_day` int comment '截止当前日期留存天数',

`retention\_count` bigint comment '留存数量',

`new\_mid\_count` bigint comment '当日设备新增数量',

`retention\_ratio` decimal(10,2) comment '留存率'

) COMMENT '每日用户留存情况'

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_user\_retention\_day\_rate/';

* + - * 1. 沉默用户数

1. 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| dt  silent\_count | 统计日期  沉默设备数 |

1. 建表语句：

hive (gmall)>

drop table if exists gmall.ads\_silent\_count;

create external table gmall.ads\_silent\_count(

`dt` string COMMENT '统计日期',

`silent\_count` bigint COMMENT '沉默设备数'

)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_silent\_count';

* + - * 1. 本周回流用户数

1. 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| dt  wk\_dt  wastage\_count | 统计日期  统计日期所在周  回流设备数 |

1. 建表语句：

hive (gmall)>

drop table if exists gmall.ads\_back\_count;

create external table ads\_back\_count(

`dt` string COMMENT '统计日期',

`wk\_dt` string COMMENT '统计日期所在周',

`wastage\_count` bigint COMMENT '回流设备数'

)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_back\_count';

* + - * 1. 流失用户数

1. 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| dt  wastage\_count | 统计日期  流失设备数 |

1. 建表语句：

hive (gmall)>

drop table if exists gmall.ads\_wastage\_count;

create external table ads\_wastage\_count(

`dt` string COMMENT '统计日期',

`wastage\_count` bigint COMMENT '流失设备数'

)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_wastage\_count';

* + - * 1. 最近连续三周活跃用户数

1. 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| dt  wk\_dt | 统计日期，一般用结束周周日日期，如果每天计算一次，可用当天日期  持续时间 |

1. 建表语句：

hive (gmall)>

drop table if exists gmall.ads\_continuity\_wk\_count;

create external table gmall.ads\_continuity\_wk\_count(

`dt` string COMMENT '统计日期,一般用结束周周日日期,如果每天计算一次,可用当天日期',

`wk\_dt` string COMMENT '持续时间',

`continuity\_count` bigint

)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_continuity\_wk\_count';

* + - * 1. 最近七天内连续三天活跃用户数

1. 表结构：

|  |  |
| --- | --- |
| 标签 | 含义 |
| dt  wk\_dt  continuity\_count | 统计日期  最近7天日期  活跃设备数 |

1. 建表语句：

hive (gmall)>

drop table if exists gmall.ads\_continuity\_uv\_count;

create external table gmall.ads\_continuity\_uv\_count(

`dt` string COMMENT '统计日期',

`wk\_dt` string COMMENT '最近7天日期',

`continuity\_count` bigint

) COMMENT '连续活跃设备数'

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_continuity\_uv\_count';

1. **系统关键技术**
   1. **Hadoop平台**

Hadoop是一个由Apache基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。Hadoop实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。HDFS有高容错性的特点，并且设计用来部署在低廉的（low-cost）硬件上；而且它提供高吞吐量（high throughput）来访问应用程序的数据，适合那些有着超大数据集（large data set）的应用程序。HDFS放宽了（relax）POSIX的要求，可以以流的形式访问（streaming access）文件系统中的数据。Hadoop的框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，而MapReduce则为海量的数据提供了计算