**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 1](#_Toc39533059)

[2 建设目标及原则 1](#_Toc39533060)

[2.1 建设目标 1](#_Toc39533061)

[2.2 建设原则 1](#_Toc39533062)

[3 项目分工及安排 2](#_Toc39533063)

[4 需求分析 1](#_Toc39533064)

[4.1 概述 1](#_Toc39533065)

[4.2 功能需求 1](#_Toc39533066)

[4.2.1 GMV成交总额 1](#_Toc39533067)

[4.2.2 用户新鲜度 1](#_Toc39533068)

[4.2.3 品牌的复购率 1](#_Toc39533069)

[5 概要设计 1](#_Toc39533070)

[5.1 概述 1](#_Toc39533071)

[5.2 系统框架设计 1](#_Toc39533072)

[5.2.1 技术架构设计 1](#_Toc39533073)

[5.2.2 系统功能结构设计 2](#_Toc39533074)

[5.3 应用系统设计 3](#_Toc39533075)

[5.3.1 设计原则 3](#_Toc39533076)

[5.3.2 功能模块设计 4](#_Toc39533077)

[5.4 接口设计 4](#_Toc39533078)

[5.4.1 外部接口 4](#_Toc39533079)

[5.4.2 内部接口 4](#_Toc39533080)

[6 详细设计 4](#_Toc39533081)

[6.1 系统概述 4](#_Toc39533082)

[6.2 系统数据流程 5](#_Toc39533083)

[6.2.1 业务数据生成 5](#_Toc39533084)

[6.2.2 业务数据导入数仓 6](#_Toc39533085)

[6.3 功能设计 12](#_Toc39533086)

[6.3.1 GMV成交总额 12](#_Toc39533087)

[6.3.2 转化率之用户新鲜度及漏斗分析 12](#_Toc39533088)

[6.3.3 品牌复购率 14](#_Toc39533089)

[7 系统关键技术 18](#_Toc39533090)

[7.1 Hadoop平台 18](#_Toc39533091)

[7.2 9.2 .Flume 18](#_Toc39533092)

[7.3 MySQL 18](#_Toc39533093)

[7.4 Hive 19](#_Toc39533094)

[7.5 Sqoop 19](#_Toc39533095)

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. **项目分工及安排**

项目主要分为基础平台搭建、数据采集传输、数仓搭建以及数据可视化四个部分，具体的项目分工如下表：

表 1 项目分工表

|  |  |  |  |
| --- | --- | --- | --- |
| **任务划分** | | | |
| **序号** | **工作内容** | **负责人** | **备注** |
|  | Hadoop基础平台搭建 | 所有人 |  |
|  | 数据采集传输 | 王伟建 |  |
|  | 数仓搭建 | 薛晓玉、杨洁、刘国栋、周擂震、姚磊 |  |
|  | 数据可视化 | 宋希文、况金荣 |  |

项目的进度计划表为：



1. 项目进度计划表
2. **需求分析**
   1. **概述**

通过大数据分析技术，为电商平台商家提供全面的、多维度的分析，并对关键指标进行了直接的可视化展示，可以帮助网站创造更多的收益价值。

* 1. **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. **用户新鲜度**

用户新鲜度主要是从宏观上了解每日用户中的新老用户比，具体指新增用户占日活跃用户的比率。

* + 1. **品牌的复购率**

品牌复购率即为重复购买率，是指消费者对某品牌产品的重复购买次数，重复购买率越高，反应出消费者对品牌的忠诚度就越高，反之则越低。

1. **概要设计**
   1. **概述**

概要设计主要是进一步细化软件设计阶段，得出软件的总体概貌，将软件加工成在程序细节上非常接近于源程序的软件表示。主要面向软件分析员、程序开发员、软件测试员和软件维护员等。

* 1. **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Mysql、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. 技术框架
   * 1. **系统功能结构设计**

系统按照功能可以分为GMV成交总额分析，用户新鲜度分析以及品牌复购率分析三个功能模块：



1. 功能组成图
   1. **应用系统设计**
      1. **设计原则**

* 模型框架

尽量使用星型架构，使用雪花架构的目的是使事实表第一级的维表数量达到最小。

* 设计方法

将常识规范化方法应用于维度表设计。例如，不相关的数据不应组合到单一维度表中，而且在多个维度表中数据不应重复。

* 维表设计

设计维表应包含需要分析的有关事实的有意义信息，例如产品的颜色和大小。

* 事实表设计

不要在事实数据表中进行过度的汇总，以保证在必要时可以进行所需粒度的数据访问。

* 增量更新

必须考虑数据增量和更新的策略，以保证多维数据集中有所需的数据。

* + 1. **功能模块设计**

系统主要分为GMV成交总额分析、用户新鲜度分析以及品牌复购率分析三个功能模块。

其中，GMV成交总额主要是由用户行为宽表选择出来的，它汇总了“下单次数”、“下单金额”以及“支付金额”的总和，由此可以对用户购买意向和用户的粘度进行分析。

用户新鲜度主要是对新增用户占日活跃用户的比率进行分析，同样是由用户行为宽表选择出来的，它包含了“单日活跃设备”、“单日新增设备”以及他们的比率，基于此可以对用户从“访问人数”到“下单人数”再到“支付人数”的行为漏斗进行分析。

品牌复购率主要是用来衡量用户的忠诚度，是对用户表、订单明细表和商品表进行综合分析得出用户购买商品明细表，并在此基础上得出单次复购率和多次复购率指标。

* 1. **接口设计**
     1. **外部接口**
     2. **内部接口**

1. **详细设计**
   1. **系统概述**

通过hadoop、sqoop、hive、mysql等技术对生成的业务数据进行分析处理。ODS层加载原始数据，不对数据进行处理，DWD层对ODS层的数据进行清洗（去除空值、脏数据、超过极限范围的数据），对商品分类表进行维度退化（降维），而在ADS则对数据进行汇总分析，为各种统计报表提供数据。经过数据仓库的分层处理，使得复杂问题实现简单化。

* 1. **系统数据流程**

### 业务数据生成

* + - 1. 导入建表脚本
      2. 生成业务数据

1. 生成业务数据函数

init\_data ( do\_date\_string VARCHAR(20) , order\_incr\_num INT, user\_incr\_num INT , sku\_num INT , if\_truncate BOOLEAN )：

参数一：do\_date\_string生成数据日期

参数二：order\_incr\_num订单id个数

参数三：user\_incr\_num用户id个数

参数四：sku\_num商品sku个数

参数五：if\_truncate是否删除数据

1. 案例测试

CALL init\_data('2019-02-10',1000,200,300,TRUE);

测试需求为生成日期2019年2月10日数据、订单1000个、用户200个、商品sku300个、删除原始数据。

### 业务数据导入数仓

* + - 1. 系统业务数据仓库表结构



* + - 1. ODS层
         1. 订单表

1. 订单表结构

|  |  |  |
| --- | --- | --- |
| 标签 | | 含义 |
| id | 订单编号 | |
| total\_amount | 订单金额 | |
| order\_status | 订单状态 | |
| user\_id | 用户id | |
| payment\_way | 支付方式 | |
| out\_trade\_no | 支付流水号 | |
| create\_time | 创建时间 | |
| operate\_time | 操作时间 | |

1. 创建订单表（ods\_order\_info）

drop table if exists gmall.ods\_order\_info;

create external table gmall.ods\_order\_info (

`id` string COMMENT '订单编号',

`total\_amount` decimal(10,2) COMMENT '订单金额',

`order\_status` string COMMENT '订单状态',

`user\_id` string COMMENT '用户id',

`payment\_way` string COMMENT '支付方式',

`out\_trade\_no` string COMMENT '支付流水号',

`create\_time` string COMMENT '创建时间',

`operate\_time` string COMMENT '操作时间'

) COMMENT '订单表'

PARTITIONED BY (`dt` string)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ods/ods\_order\_info/'

;

* + - * 1. 订单详情表

1. 订单详情表结构

|  |  |  |
| --- | --- | --- |
| 标签 | | 含义 |
| id | 订单编号 | |
| order\_id | 订单号 | |
| user\_id | 用户id | |
| sku\_id | 商品id | |
| sku\_name | 商品名称 | |
| order\_price | 商品价格 | |
| sku\_num | 商品数量 | |
| create\_time | 创建时间 | |

1. 创建订单详情表

drop table if exists gmall.ods\_order\_info;

create external table gmall.ods\_order\_info (

`id` string COMMENT '订单编号',

`total\_amount` decimal(10,2) COMMENT '订单金额',

`order\_status` string COMMENT '订单状态',

`user\_id` string COMMENT '用户id',

`payment\_way` string COMMENT '支付方式',

`out\_trade\_no` string COMMENT '支付流水号',

`create\_time` string COMMENT '创建时间',

`operate\_time` string COMMENT '操作时间'

) COMMENT '订单表'

PARTITIONED BY (`dt` string)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ods/ods\_order\_info/'

;

* + - * 1. 商品表

1. 商品表结构

| 标签 | | 含义 |
| --- | --- | --- |
| id | skuId | |
| spu\_id | spuid | |
| price | 价格 | |
| sku\_name | 商品名称 | |
| sku\_desc | 商品描述 | |
| weight | 重量 | |
| tm\_id | 品牌id | |
| category3\_id | 品类id | |
| create\_time | 创建时间 | |

1. 创建商品表

drop table if exists gmall.ods\_sku\_info;

create external table gmall.ods\_sku\_info(

`id` string COMMENT 'skuId',

`spu\_id` string COMMENT 'spuid',

`price` decimal(10,2) COMMENT '价格',

`sku\_name` string COMMENT '商品名称',

`sku\_desc` string COMMENT '商品描述',

`weight` string COMMENT '重量',

`tm\_id` string COMMENT '品牌id',

`category3\_id` string COMMENT '品类id',

`create\_time` string COMMENT '创建时间'

) COMMENT '商品表'

PARTITIONED BY (`dt` string)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ods/ods\_sku\_info/'

;

* + - * 1. 用户表

1. 用户表结构

|  |  |  |
| --- | --- | --- |
| 标签 | | 含义 |
| id | 用户id | |
| name | 姓名 | |
| birthday | 生日 | |
| gender | 性别 | |
| email | 邮箱 | |
| user\_level | 用户等级 | |
| create\_time | 创建时间 | |

1. 创建用户表

drop table if exists gmall.ods\_user\_info;

create external table gmall.ods\_user\_info(

`id` string COMMENT '用户id',

`name` string COMMENT '姓名',

`birthday` string COMMENT '生日',

`gender` string COMMENT '性别',

`email` string COMMENT '邮箱',

`user\_level` string COMMENT '用户等级',

`create\_time` string COMMENT '创建时间'

) COMMENT '用户信息'

PARTITIONED BY (`dt` string)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ods/ods\_user\_info/'

;

* + - * 1. 商品一级分类表

1. 商品一级分类表结构

|  |  |  |
| --- | --- | --- |
| 标签 | | 含义 |
| id | id | |
| name | 名称 | |

1. 创建商品一级分类表

drop table if exists gmall.ods\_base\_category1;

create external table gmall.ods\_base\_category1(

`id` string COMMENT 'id',

`name` string COMMENT '名称'

) COMMENT '商品一级分类'

PARTITIONED BY (`dt` string)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ods/ods\_base\_category1/'

;

* + - * 1. 商品二级分类表

1. 商品二级分类表结构

|  |  |  |
| --- | --- | --- |
| 标签 | | 含义 |
| id | id | |
| name | 名称 | |
| category1\_id | 一级品类id | |

1. 创建商品二级分类表

drop table if exists gmall.ods\_base\_category2;

create external table gmall.ods\_base\_category2(

`id` string COMMENT ' id',

`name` string COMMENT '名称',

category1\_id string COMMENT '一级品类id'

) COMMENT '商品二级分类'

PARTITIONED BY (`dt` string)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ods/ods\_base\_category2/'

;

* + - * 1. 商品三级分类表

1. 商品三级分类表结构

|  |  |  |
| --- | --- | --- |
| 标签 | | 含义 |
| id | id | |
| name | 名称 | |
| Category2\_id | 二级品类id | |

1. 创建商品三级分类表

drop table if exists gmall.ods\_base\_category3;

create external table gmall.ods\_base\_category3(

`id` string COMMENT ' id',

`name` string COMMENT '名称',

category2\_id string COMMENT '二级品类id'

) COMMENT '商品三级分类'

PARTITIONED BY (`dt` string)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ods/ods\_base\_category3/'

;

* + - * 1. 支付流水表

1. 支付流水表结构

|  |  |  |
| --- | --- | --- |
| 标签 | | 含义 |
| id | 编号 | |
| out\_trade\_no | 对外业务编号 | |
| order\_id | 订单编号 | |
| user\_id | 用户编号 | |
| alipay\_trade\_no | 支付宝交易流水编号 | |
| total\_amount | 支付金额 | |
| subject | 交易内容 | |
| payment\_type | 支付类型 | |
| payment\_time | 支付时间 | |

1. 创建支付流水表

drop table if exists gmall.ods\_payment\_info;

create external table gmall.ods\_payment\_info(

`id` bigint COMMENT '编号',

`out\_trade\_no` string COMMENT '对外业务编号',

`order\_id` string COMMENT '订单编号',

`user\_id` string COMMENT '用户编号',

`alipay\_trade\_no` string COMMENT '支付宝交易流水编号',

`total\_amount` decimal(16,2) COMMENT '支付金额',

`subject` string COMMENT '交易内容',

`payment\_type` string COMMENT '支付类型',

`payment\_time` string COMMENT '支付时间'

)COMMENT '支付流水表'

PARTITIONED BY (`dt` string)

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ods/ods\_payment\_info/'

;

* + - 1. DWD层

创建商品表（增加分类）

1. 商品表结构

|  |  |  |
| --- | --- | --- |
| 标签 | | 含义 |
| id | id | |
| spu\_id | spuid | |
| price | 价格 | |
| sku\_name | 商品名称 | |
| sku\_desc | 商品描述 | |
| weight | 重量 | |
| tm\_id | 品牌id | |
| category3\_id | 品类id | |
| category2\_id | 二级品类id | |
| category1\_id | 一级品类id | |
| category3\_name | 品类名称 | |
| category2\_name | 二级品类名称 | |
| category1\_name | 一级品类名称 | |
| create\_time | 创建时间 | |

1. 建表语句：

drop table if exists gmall.dwd\_sku\_info;

create external table gmall.dwd\_sku\_info(

`id` string COMMENT 'skuId',

`spu\_id` string COMMENT 'spuid',

`price` decimal(10,2) COMMENT '',

`sku\_name` string COMMENT '',

`sku\_desc` string COMMENT '',

`weight` string COMMENT '',

`tm\_id` string COMMENT 'id',

`category3\_id` string COMMENT '1id',

`category2\_id` string COMMENT '2id',

`category1\_id` string COMMENT '3id',

`category3\_name` string COMMENT '3',

`category2\_name` string COMMENT '2',

`category1\_name` string COMMENT '1',

`create\_time` string COMMENT ''

)

PARTITIONED BY (`dt` string)

stored as parquet

location '/warehouse/gmall/dwd/dwd\_sku\_info/'

tblproperties ("parquet.compression"="snappy")

;

* 1. **功能设计**
     1. **GMV成交总额**

1. 表结构

|  |  |
| --- | --- |
| 标签 | 含义 |
| dt | 统计日期 |
| gmv\_count | 当日gmv订单个数 |
| gmv\_amount | 当日gmv订单总金额 |
| gmv\_payment | 当日支付金额 |

1. 建表语句

drop table if exists gmall.ads\_gmv\_sum\_day;

create external table gmall.ads\_gmv\_sum\_day(

`dt` string COMMENT '统计日期',

`gmv\_count` bigint COMMENT '当日gmv订单个数',

`gmv\_amount` decimal(16,2) COMMENT '当日gmv订单总金额',

`gmv\_payment` decimal(16,2) COMMENT '当日支付金额'

) COMMENT 'GMV'

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_gmv\_sum\_day/'

;

1. 数据导入

insert into table gmall.ads\_gmv\_sum\_day

select

'2019-02-10' dt,

sum(order\_count) gmv\_count,

sum(order\_amount) gmv\_amount,

sum(payment\_amount) payment\_amount

from gmall.dws\_user\_action

where dt ='2019-02-10'

group by dt

;

### 转化率之用户新鲜度及漏斗分析

* + - 1. 新增用户占日活跃用户比率（用户新鲜度）

1. 表结构

|  |  |
| --- | --- |
| 标签 | 含义 |
| dt | 统计日期 |
| uv\_m\_count | 当日活跃设备 |
| new\_m\_count | 当日新增设备 |
| new\_m\_ratio | 当日新增占日活的比率 |

1. 建表语句

drop table if exists gmall.ads\_user\_convert\_day;

create external table gmall.ads\_user\_convert\_day(

`dt` string COMMENT '统计日期',

`uv\_m\_count` bigint COMMENT '当日活跃设备',

`new\_m\_count` bigint COMMENT '当日新增设备',

`new\_m\_ratio` decimal(10,2) COMMENT '当日新增占日活的比率'

) COMMENT '转化率'

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_user\_convert\_day/'

;

1. 数据导入

insert into table gmall.ads\_user\_convert\_day

select

'2019-02-10',

sum(uc.dc) sum\_dc,

sum(uc.nmc) sum\_nmc,

cast(sum( uc.nmc)/sum( uc.dc)\*100 as decimal(10,2)) new\_m\_ratio

from

(

select

day\_count dc,

0 nmc

from gmall.ads\_uv\_count

where dt='2019-02-10'

union all

select

0 dc,

new\_mid\_count nmc

from gmall.ads\_new\_mid\_count

where create\_date='2019-02-10'

)uc;

* + - 1. 用户行为漏斗分析

1. 表结构

|  |  |
| --- | --- |
| 标签 | 含义 |
| dt | 统计日期 |
| order\_u\_count | 下单人数 |
| total\_visitor\_m\_count | 总访问人数 |
| visitor2order\_convert\_ratio | 访问到下单转化率 |
| order2payment\_convert\_ratio | 下单到支付的转化率 |

1. 建表语句

drop table if exists gmall.ads\_user\_action\_convert\_day;

create external table gmall.ads\_user\_action\_convert\_day(

`dt` string COMMENT '统计日期',

`total\_visitor\_m\_count` bigint COMMENT '总访问人数',

`order\_u\_count` bigint COMMENT '下单人数',

`visitor2order\_convert\_ratio` decimal(10,2) COMMENT '访问到下单转化率',

`payment\_u\_count` bigint COMMENT '支付人数',

`order2payment\_convert\_ratio` decimal(10,2) COMMENT '下单到支付的转化率'

) COMMENT '用户行为漏斗分析'

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_user\_action\_convert\_day/'

;

1. 数据导入

insert into table gmall.ads\_user\_action\_convert\_day

select

'2019-02-10',

uv.day\_count,

ua.order\_count,

cast(ua.order\_count/uv.day\_count as decimal(10,2)) visitor2order\_convert\_ratio,

ua.payment\_count,

cast(ua.payment\_count/ua.order\_count as decimal(10,2)) order2payment\_convert\_ratio

from

(

select

dt,

sum(if(order\_count>0,1,0)) order\_count,

sum(if(payment\_count>0,1,0)) payment\_count

from gmall.dws\_user\_action

where dt='2019-02-10'

group by dt

)ua join gmall.ads\_uv\_count uv on uv.dt=ua.dt

;

### 品牌复购率

* + - 1. 用户购买商品明细表

1. 表结构

|  |  |
| --- | --- |
| 标签 | 含义 |
| user\_id | 用户 id |
| sku\_id | 商品 Id |
| user\_gender | 用户性别 |
| user\_age | 用户年龄 |
| user\_level | 用户等级 |
| order\_price decimal | 商品价格 |
| sku\_name | 商品名称 |
| sku\_tm | 品牌id |
| sku\_category3 | 商品三级品类id |
| sku\_category2 | 商品二级品类id |
| sku\_category1 | 商品一级品类id |
| sku\_category3 | 商品三级品类名称 |
| sku\_category2 | 商品二级品类名称 |
| sku\_category1 | 商品一级品类名称 |
| spu\_id | 商品 spu |
| sku\_num | 购买个数 |
| order\_count | 当日下单单数 |
| order\_amount | 当日下单金额 |

1. 建表语句

drop table if exists gmall.dws\_sale\_detail\_daycount;

create external table gmall.dws\_sale\_detail\_daycount

(

user\_id string comment '用户 id',

sku\_id string comment '商品 Id',

user\_gender string comment '用户性别',

user\_age string comment '用户年龄',

user\_level string comment '用户等级',

order\_price decimal(10,2) comment '商品价格',

sku\_name string comment '商品名称',

sku\_tm\_id string comment '品牌id',

sku\_category3\_id string comment '商品三级品类id',

sku\_category2\_id string comment '商品二级品类id',

sku\_category1\_id string comment '商品一级品类id',

sku\_category3\_name string comment '商品三级品类名称',

sku\_category2\_name string comment '商品二级品类名称',

sku\_category1\_name string comment '商品一级品类名称',

spu\_id string comment '商品 spu',

sku\_num int comment '购买个数',

order\_count string comment '当日下单单数',

order\_amount string comment '当日下单金额'

) COMMENT '用户购买商品明细表'

PARTITIONED BY (`dt` string)

stored as parquet

location '/warehouse/gmall/dws/dws\_user\_sale\_detail\_daycount/'

tblproperties ("parquet.compression"="snappy");

1. 数据导入

品牌复购率with

tmp\_detail as

(

select

user\_id,

sku\_id,

sum(sku\_num) sku\_num,

count(\*) order\_count,

sum(od.order\_price\*sku\_num) order\_amount

from gmall.dwd\_order\_detail od

where od.dt='2019-02-10'

group by user\_id, sku\_id

)

insert overwrite table gmall.dws\_sale\_detail\_daycount partition(dt='2019-02-10')

select

tmp\_detail.user\_id,

tmp\_detail.sku\_id,

u.gender,

months\_between('2019-02-10', u.birthday)/12 age,

u.user\_level,

price,

sku\_name,

tm\_id,

category3\_id,

category2\_id,

category1\_id,

category3\_name,

category2\_name,

category1\_name,

spu\_id,

tmp\_detail.sku\_num,

tmp\_detail.order\_count,

tmp\_detail.order\_amount

from tmp\_detail

left join gmall.dwd\_user\_info u on tmp\_detail.user\_id =u.id and u.dt='2019-02-10'

left join gmall.dwd\_sku\_info s on tmp\_detail.sku\_id =s.id and s.dt='2019-02-10'

;

* + - 1. 品牌复购率

1. 表结构

|  |  |
| --- | --- |
| 标签 | 含义 |
| tm\_id | 一级品类id |
| category1\_id | 一级品类名称 |
| buycount | 购买人数 |
| buy\_twice\_last bigint | 两次以上购买人数 |
| buy\_twice\_last\_ratio | 单次复购率 |
| buy\_3times\_last | 三次以上购买人数 |
| buy\_3times\_last\_ratio | 多次复购率 |
| stat\_mn | 统计月份 |
| category1\_id | 一级品类id |
| category3\_name | 品类名称 |
| stat\_date | 统计日期 |
| category1\_name | 一级品类名称 |

1. 建表语句

drop table if exists gmall.ads\_sale\_tm\_category1\_stat\_mn;

create external table gmall.ads\_sale\_tm\_category1\_stat\_mn

(

tm\_id string comment '品牌id',

category1\_id string comment '1级品类id ',

category1\_name string comment '1级品类名称 ',

buycount bigint comment '购买人数',

buy\_twice\_last bigint comment '两次以上购买人数',

buy\_twice\_last\_ratio decimal(10,2) comment '单次复购率',

buy\_3times\_last bigint comment '三次以上购买人数',

buy\_3times\_last\_ratio decimal(10,2) comment '多次复购率',

stat\_mn string comment '统计月份',

stat\_date string comment '统计日期'

) COMMENT '复购率统计'

row format delimited fields terminated by '\t'

location '/warehouse/gmall/ads/ads\_sale\_tm\_category1\_stat\_mn/'

;

1. 数据导入

insert into table gmall.ads\_sale\_tm\_category1\_stat\_mn

select

mn.sku\_tm\_id,

mn.sku\_category1\_id,

mn.sku\_category1\_name,

sum(if(mn.order\_count>=1,1,0)) buycount,

sum(if(mn.order\_count>=2,1,0)) buyTwiceLast,

sum(if(mn.order\_count>=2,1,0))/sum( if(mn.order\_count>=1,1,0)) buyTwiceLastRatio,

sum(if(mn.order\_count>=3,1,0)) buy3timeLast ,

sum(if(mn.order\_count>=3,1,0))/sum( if(mn.order\_count>=1,1,0)) buy3timeLastRatio ,

date\_format('2019-02-10' ,'yyyy-MM') stat\_mn,

'2019-02-10' stat\_date

from

(

select

user\_id,

sd.sku\_tm\_id,

sd.sku\_category1\_id,

sd.sku\_category1\_name,

sum(order\_count) order\_count

from gmall.dws\_sale\_detail\_daycount sd

where date\_format(dt,'yyyy-MM')=date\_format('2019-02-10' ,'yyyy-MM')

group by user\_id, sd.sku\_tm\_id, sd.sku\_category1\_id, sd.sku\_category1\_name

) mn

group by mn.sku\_tm\_id, mn.sku\_category1\_id, mn.sku\_category1\_name

;

1. **系统关键技术**
   1. **MySQL**

MySQL是一个[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511)，由瑞典MySQL AB 公司开发，目前属于 [Oracle](https://baike.baidu.com/item/Oracle) 旗下产品。MySQL 是最流行的[关系型数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E5%9E%8B%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/696511" \t "_blank)之一，在 WEB 应用方面，MySQL是最好的 [RDBMS](https://baike.baidu.com/item/RDBMS/1048260) (Relational Database Management System，关系数据库管理系统) 应用软件之一。

MySQL是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，而不是将所有数据放在一个大仓库内，这样就增加了速度并提高了灵活性。

MySQL所使用的 SQL 语言是用于访问[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728)的最常用标准化语言。

* 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。

* 1. **Flume**

Flume 是一种分布式，可靠且可用的服务，用于有效地收集，聚

合和移动大量日志数据。它具有基于流数据流的简单灵活的体系 结构。它具有可调整的可靠性机制以及许多故障转移和恢复机制， 具有强大的功能和容错能力。它使用一个简单的可扩展数据模型， 允许在线分析应用程序。

* 1. **Sqoop**

Sqoop是一款开源的工具，主要用于在Hadoop(Hive)与传统的数据库(mysql、postgresql...)间进行数据的传递，可以将一个关系型数据库（例如 ： MySQL ,Oracle ,Postgres等）中的数据导进到Hadoop的HDFS中，也可以将HDFS的数据导进到关系型数据库中。

* 1. **Hive**

Apache Hive数据仓库软件有助于使用SQL读取，写入和管理驻留在分布式存储中的大型数据集。可以将结构投影到已经存储的数据上。提供了命令行工具和JDBC驱动程序以将用户连接到Hive。