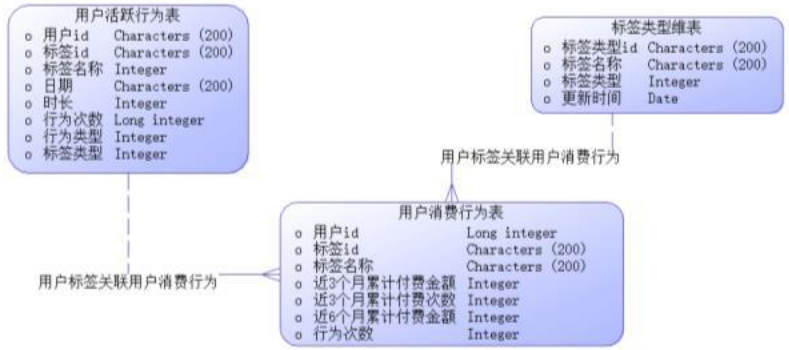
如何进行用户画像建模打标签

Live分享人 — 赵宏田

用户画像应用场景

用户画像的应用场景主要包括业务精细化运营、数据分析与挖掘、精准营销、搜索和广告的个性化定向推送等

业务精细化运营——筛选客群



数据分析挖掘

精准营销：邮件营销、短信营销

个性化推荐：产品站内推荐

产品层面的宏观分析维度（1）

看流量趋势

- ✓ 访客趋势（访客每日的访问量）
- ✓ 浏览趋势（浏览量每日趋势）
- ✓ 新访客趋势（新访客每日访问趋势）
- ✓ 活跃访客趋势（活跃访客每日访问趋势）
- ✓ 访问量（每日、每周、每月）

看页面访问特征

- ✓ 受访画像（各品类页面访问量统计）
- ✓ 进入画像（访客从哪些页面进入网站）
- ✓ 离开画像（访客从哪些页面离开网站）
- ✓ 页面热点图（优化网页设计）
- ✓ 访问标记（访客在页面上点击哪些内容或者id元素）
- ✓ 主机域名（网站子域名访问量）
- ✓ 访问目录（网站子目录访问量）
- ✓ 外链网站（访客点击哪些站外链接离开网站）

看用户整体行为

- ✓ 跳出率（访问行为评估）
- ✓ 忠诚度（访问质量评估）
- ✓ 活跃度（活跃度、流失分析）
- ✓ 用户关联度聚类画像（用户与用户之间的关系）
- ✓ 新用户画像（新用户来源渠道）
- ✓ 访客浏览路径热点画像（用户浏览习惯调研）

看注册会员特征

- ✓ 性别画像（性格的占比）
- ✓ 年龄分布画像（按标准年龄段的正态分布）
- ✓ 教育背景画像（教育背景）
- ✓ 职业分布画像（职业背景）
- ✓ 特征分布画像（多标签特征库，购物狂，游戏迷）
- ✓ 会员游客画像（详细信息画像）
- ✓ 匿名用户画像（会员不详细用户画像）

产品层面的宏观分析维度（2）

看访客特征

- ✓ 地域分析（访客地域位置的分布）
- ✓ 时段分析（访客访问时段分布）
- ✓ 客户端环境（访问客户端分析）
- ✓ 设备属性画像（使用硬件信息）
- ✓ 移动终端（访客上网设备分析）
- ✓ 网络连接画像（不同网络的连接方式运营商）

看用户来源

- ✓ 来源分类（直接输入、搜索引擎、商务合作外链等）
- ✓ 来源网站（网站统计）
- ✓ 来源页面（网站链接）
- ✓ 直接访问（浏览器直接进去）
- ✓ 搜索引擎（具体的搜索引擎画像）
- ✓ 搜索关键词（热点关键词画像）
- ✓ 广告营销（通过广告进入）
- ✓ 移动APP（ASO推广渠道来源）

- ➢ 满足产品层面的宏观分析是画像应用的一个重要方面，而基于画像去做个性化推荐（偏向于站内行为，数据一般做在cookie_id维度）、精准营销（偏向于站外获取，数据一般做在user_id维度）等业务同样是应用的重点；
- ➢ 做好个性化推荐和精准营销的前提是，将用户标签开发的基本功打扎实

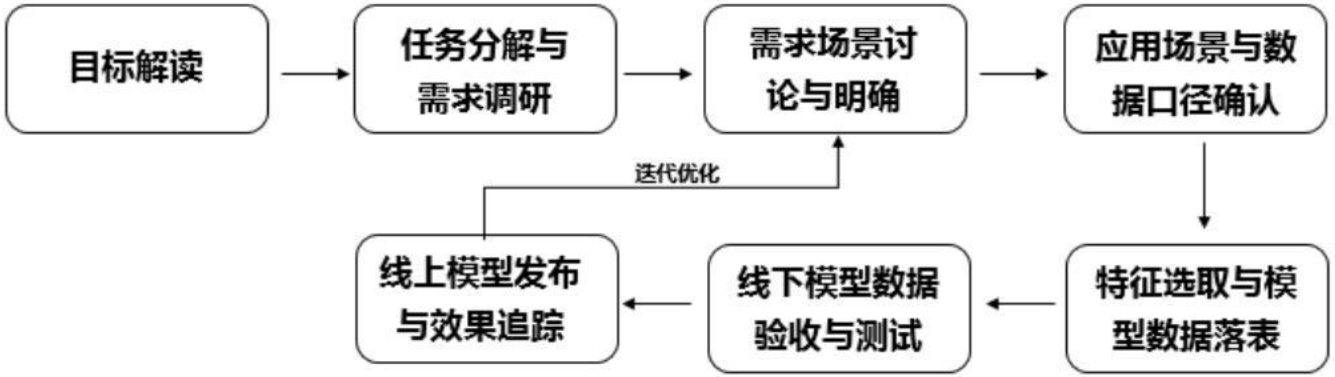
备注：cookie_id是用户访问设备标识，当用户未登录状态下访问时的偏好，记录在cookie_id维度。User_id是用户在产品上的唯一标识，在用户登录状态下时记录

用户画像标签类型

用户画像建模其实就是对用户进行打标签，从对用户打标签的方式来看，一般分为三种类型： 1、基于统计类的标签；2、基于规则类的标签、3、基于挖掘类的标签。下面我们介绍这三种类型标签的区别：

- **基于统计类的标签：**这类标签是最为基础也最为常见的标签类型，例如对于某个用户来说，他的性别、年龄、城市、星座、近7日活跃时长、近7日活跃天数、近7日活跃次数等字段可以从用户注册数据、用户访问、消费类数据中统计得出。该类标签构成了用户画像的基础；
- **基于规则类的标签：**该类标签基于用户行为及确定的规则产生。例如对平台上“消费活跃”用户这一口径的定义为近30天交易次数>=2。在实际开发画像的过程中，由于运营人员对业务更为熟悉、而数据人员对数据的结构、分布、特征更为熟悉，因此规则类标签的规则确定由运营人员和数据人员共同协商确定；
- **基于挖掘类的标签：**该类标签通过数据挖掘产生，应用在对用户的某些属性或某些行为进行预测判断。例如根据一个用户的行为习惯判断该用户是男性还是女性，根据一个用户的消费习惯判断其对某商品的偏好程度。该类标签需要通过算法挖掘产生。

画像项目开发流程



第一阶段：目标解读

一般而言，用户画像的服务对象包括运营人员、客服、数据分析人员等。不同业务方对用户画像的需求有不同的侧重点，就运营人员来说，他们需要分析用户的特征、定位用户行为偏好，做商品或内容的个性化推送以提高点击转化率，所以画像的侧重点落在用户个人行为偏好；就数据分析人员来说，他们需要分析用户行为特征，做好用户的流失预警工作，还可根据用户的消费偏好做更有针对性的精准营销。

第二阶段：任务分解与需求调研

经过第一阶段的需求调研和目标解读，我们已经明确了用户画像的服务对象与应用场景，接下来需要针对服务对象的需求侧重点，结合产品现有业务体系和“数据字典”规约实体和标签之间的关联关系，明确分析纬度。一般需要建立用户属性画像、用户行为画像、用户偏好画像、用户群体偏好画像等。

第三、四阶段：

根据需求，输出《产品用户画像需求文档》，在该文档中明确画像应用场景、最终开发出的标签内容与应用方式 。结合业务与数据仓库中已有的相关表，明确与各业务场景相关的数据口径，输出《产品用户画像实施文档》，该文档需要明确应用场景、标签开发的模型、涉及到的数据库与表，应用实施流程。

第五阶段：特征选取与模型数据落表

本阶段中数据分析挖掘人员需要根据前面明确的需求场景进行业务建模，写好*HQL*逻辑，将相应的模型逻辑写入临时表中，抽取数据校验是否符合业务场景需求。

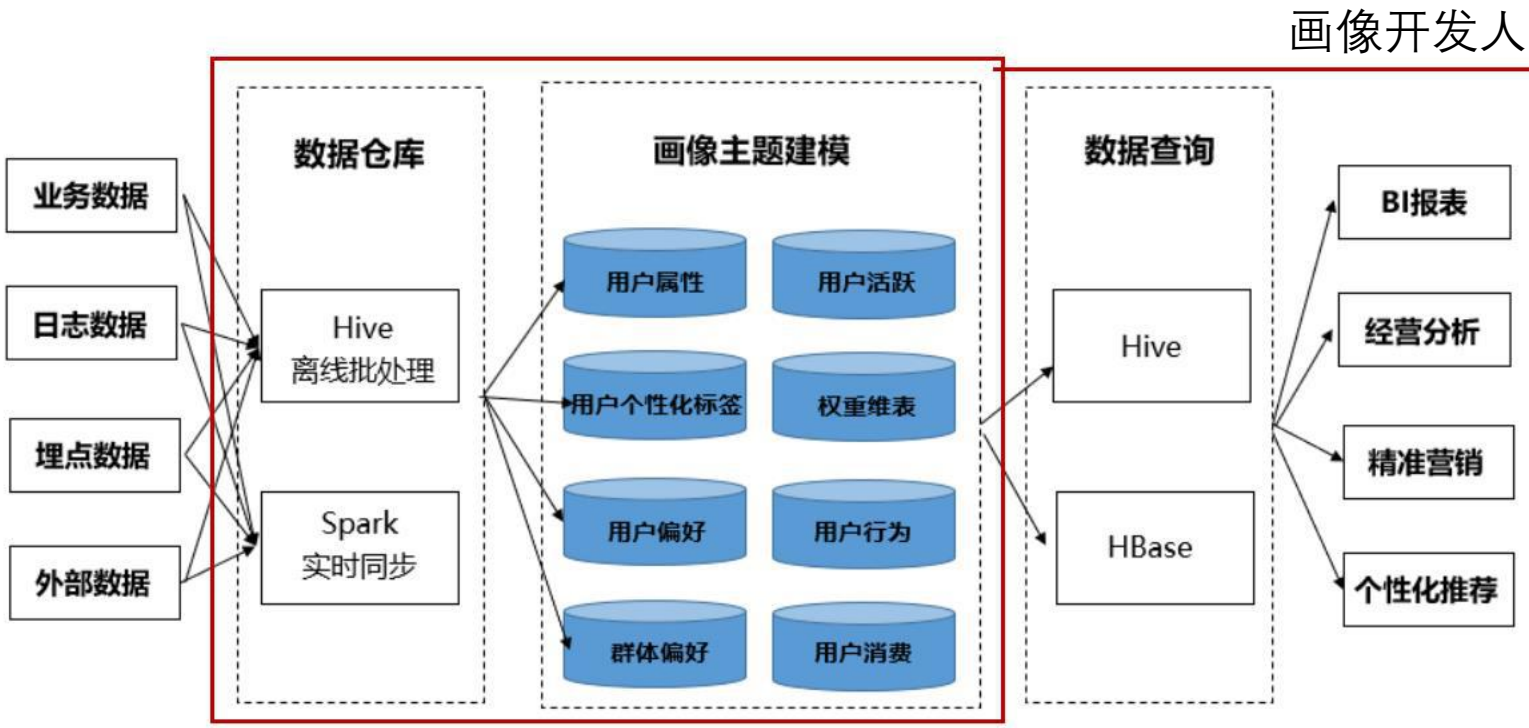
第六阶段：线下模型数据验收与测试

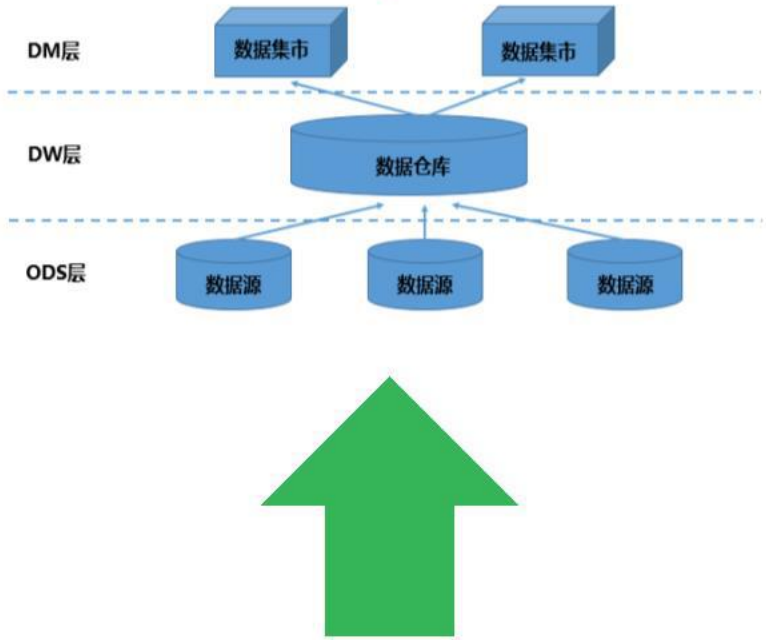
数据仓库团队的人员将相关数据落表后，设置定时调度任务，进行定期增量更新数据。数据运营人员需要验收数仓加工的*HQL*逻辑是否符合需求，根据业务需求抽取查看表中数据范围是否在合理范围内，如果发现问题及时反馈给数据仓库人员调整代码逻辑和行为权重的数值。

第七阶段：线上模型发布与效果追踪

经过第六阶段，数据通过验收之后，就可以将数据接口给到搜索、或技术团队部署上线了。上线后通过对用户点击转化行为的持续追踪，调整优化模型及相关权重配置。

数据仓库介绍





用户画像的应用流程从原始的数据输入到模型应用可分为5个部分

- 将操作型环境数据经ETL后集中存储在数据仓库;
- 经过对数据的建模、挖掘、分析建立用户画像模型;
- 最终将建好用户画像的数据接口调用到BI报表、经营分析、精准营销、个性化推荐等各系统模块

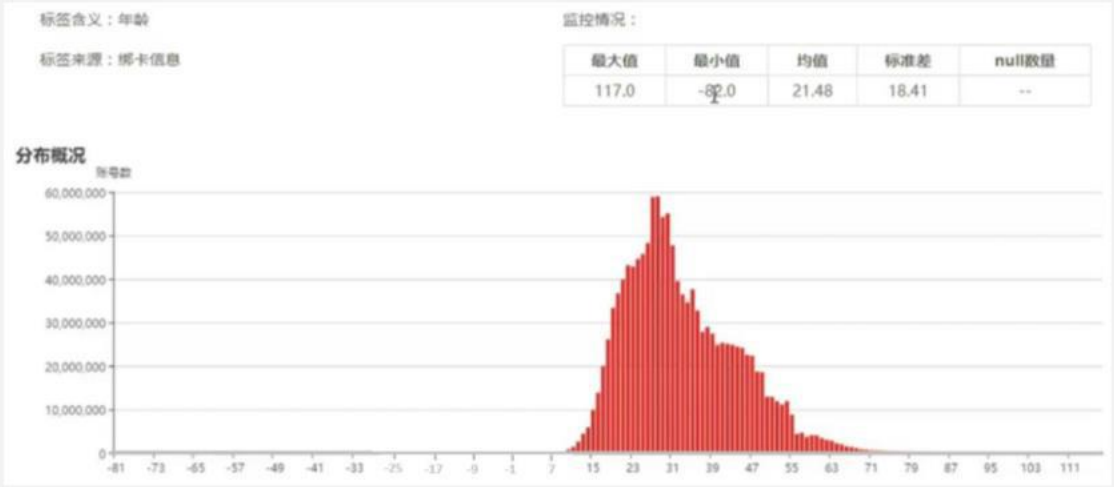
画像开发人员的工作主要是将数据仓库ODS层、DW层的数据建模，加工出数据集市层

画像数据质量管理

数据异常原因

- 在建好用户画像模型后，数据仓库的开发人员通过通过调度任务，每天定时从各业务数据表和日志数据表抽取加工；
- 各种类型的标签每天加工的数据成百上千万条，数据分析人员不会每天对这些数据的质量进行核查，去发现每类标签的数量是否有异常情况

某业务线用户画像用户年龄分布



解决方案

- 为了避免在应用的过程中异常数据导致推荐、分析结论的错误，需要便捷、有效地建立一种对用户画像数据质量进行管理的方式；
- 通过设定画像各业务表各类行为数据监控范围，当每天跑ETL数据出现异常变动时，自动发送邮件给相关人员，然后查找原因进行定位。

当某用户画像表数据跑批异常时自动发邮件

常见需要开发的用户画像相关模型

需要开发的画像数据从大类上可划分为用户人口属性画像和用户行为画像，进一步细分的话可在这两个画像基础上开发出用户偏好画像和群体属性、群体偏好画像等

用户人口属性画像

字段	字段类型	字段定义	备注
user_id	Bigint	用户编码	
login_name	String	登录名称	
user_name	String	用户姓名	
user_status_id	Int	用户状态	0未激活，1已激活，2作废，3黄牛禁用
gender_id	Int	用户性别	1男，2女，3未知
birthday	Int	用户生日	
user_age	Int	用户年龄	
constellation_name	String	用户星座名称	白羊座(03.21-04.19)，金牛座(04.20-05.20)，双子座(05.21-06.21)...
zodiac_name	Sting	用户生肖名称	鼠，牛，虎，兔，龙，蛇，马，羊，猴，鸡，狗，猪
cellphone_id	String	用户手机编码	
cert_id	String	用户证件号码	
cert_std_region_id	String	证件归属地标准区域编码	
cert_province_name	String	证件归属地省份名称	根据身份证信息进行解析
cert_std_city_id	String	证件归属地标准市级编码	根据身份证信息进行解析
cert_city_id	String	证件归属地市级编码	根据身份证信息进行解析
cert_city_name	String	证件归属地市级名称	
phone_std_region_id	String	手机归属地标准区域编码	根据手机号信息进行解析
phone_region_name	String	手机归属地区域名称	根据手机号信息进行解析
phone_std_province_id	String	手机归属地标准省份编码	根据手机号信息进行解析
phone_province_name	String	手机归属地省份名称	
phone_std_city_id	String	手机归属地标准市级编码	
phone_city_name	String	手机归属地市级名称	
create_time	Timestamp	注册时间	从用户注册表单获取
create_date	String	注册日期	
cert_region_name	String	证件归属地区域名称	
cert_province_id	String	证件归属地省份编码	

常见需要开发的用户画像相关模型

用户行为标签画像

字段	字段类型	字段定义	备注
user_id	string	用户id	用户唯一id
org_id	string	标签id	各类商品对应id
org_name	string	标签中文名称	标签id对应中文名称
is_valid	string	是否付费	该标签产生过程是否有付费行为
cnt	string	行为次数	用户行为次数
date_id	string	行为日期	产生用户该条标签对应日期
act_type_id	int	用户行为类型	搜索、浏览、收藏、支付等行为
tag_type_id	int	标签类型	可以按商品类型做划分

一些总结

1. 不同公司虽然业务不同，但开发标签的侧重点都大同小异；
2. 用户属性类和用户行为类标签是开发的重点，无论是做偏好或预测类标签也都是以上面两类标签为基础进行深度挖掘；
3. 画像相关数据模型需要从业务需求出发，这两页的画像相关表结构仅供参考

用户偏好画像

字段	字段类型	字段定义	备注
user_id	String	用户编码	用户唯一id
org_id	String	原始编码	标签id
tag_name	String	标签名称	对应标签中文名称
act_weight	Decimal	权重值	用户行为权重
cnt	Int	行为次数	用户行为次数
data_date	String	数据日期	

用户登录活跃信息

字段	字段类型	字段定义	备注
user_id	Int	用户id	用户唯一id
login_city_ration	string	常登陆地	记录用户近一个月常登陆的三个地点及比率
last_online_date	String	最近登录日期	用户最近一次登录日期
online_frequency	Int	登录频次	用户近一个月登录频次
online_time	Int	登录时长	用户近一个月登录时长/秒

某用户行为标签表开发案例—背景介绍（1）

案例背景

某图书电商网站拥有超过千万的网购用户群体，所售各品类图书100余万种。用户在平台上可进行浏览、搜索、收藏、下单、购买等浏览、交易行为。为了更好地运营网站产品，需要进一步了解用户，以便做精准营销和个性化推荐。为此需要开发出用户行为标签表

项目分析

- **目标：**商城自建立以来，数据仓库中积累着大量业务数据、日志数据及埋点数据。如何充分挖掘沉淀在数据仓库中数据的价值，有效支持到用户画像的建设，成为当前的重要工作；
- **数据类型：**在本案例中，可以获取到的数据按其类型可分为：业务类数据、用户行为数据。其中业务类数据是指用户在平台上下单、购买、收藏物品、货物配送等与业务相关的数据；用户行为数据指用户搜索某条信息、访问某个页面、点击某个按钮、提交某个表单等通过操作行为产生（在解析日志的埋点表中）的数据；
- **相关元数据：**涉及到数据仓库中的表主要包括：用户信息表、商品订单表、图书信息表、图书类目表、APP端日志表、WEB端日志表、商品评论表等，下面就用户画像建模过程中一些主要用到的数据表做详细介绍

某用户行为标签表开发案例一元数据介绍（2）

相关元数据介绍

用户信息表(dwd.user_basic_info)：存放有关用户的各种信息，例如用户姓名、年龄、性别、号码、归属地等信息。

用户信息表（dwd.user_basic_info）

字段	字段类型	字段定义	备注
user_id	character varying(50)	用户编码	
user_name	character varying(50)	用户姓名	
user_status_id	integer	用户状态	0:未注册；1:已注册；2:已注销
mail_id	character varying(40)	邮箱编码	
birthday	character varying(40)	用户生日	
gender_id	smallint	性别	0:男 1:女 2:其他
call_phone_id	character varying(64)	电话号码	
is_has_photo	smallint	是否有头像	
gmt_created	timestamp	创建时间	
gmt_created_date	date	注册日期	
province_name	character varying(20)	归属省	用户填写>手机号归属地>身份证归属地
city_name	character varying(20)	归属市	同上
user_address	character varying(320)	详细地址	

商品订单表(dwd.gdm_ord_order)：存放商品订单的各类信息，包括订单编号、用户 id、用户姓名、订单生成时间、订单状态等信息。

商品订单表（dwd.gdm_ord_order）

字段	字段类型	字段定义	备注
id	bigint	自增主键	
source_id	bigint	订单来源标识	0:app 1:web 2: H5 3:其他
user_id	character varying(50)	用户编码	
user_name	character varying(50)	用户姓名	
order_id	integer	订单号	
std_book_id	bigint	图书编码	
std_book_name	character varying(80)	图书名称	
create_time	timestamp	订单生成时间	
create_date	date	订单日期	
order_remark	character varying(80)	订单备注	
status_id	bigint	订单状态	1:待支付 2:已完成 3:已取消 4:已退款 5:支付失败
status_time	timestamp	订单状态时间	
order_amount	double precision	订单金额	
pay_account	character varying(50)	付款账户	
pay_type_id	character varying(30)	付款方式	

某用户行为标签表开发案例—元数据介绍（3）

相关元数据介绍

图书信息表（dwd.book_base_basic_info）：存放图书名称、作者、出版社、价格、页数、出版时间等信息。

图书信息表（dwd.book_base_basic_info）

字段	字段类型	字段定义	备注
book_id	bigint	图书 id	
book_name	character varying(50)	图书名称	
author	character varying(50)	作者	
republic_name	character varying(50)	出版社	
republic_time	timestamp	出版时间	
book_price	double precision	价格	
book_isbn	character varying(50)	ISBN 编号	
book_pages	bigint	页数	
book_words	bigint	字数	
print_time	timestamp	印刷时间	
book_font	bigint	开本	
book_pattern	character varying(50)	纸张类型	
print_times	bigint	印次	

图书类目表（dwd.book_std_type_df）：存放了图书归属的类别信息，可通过图书 id 与图书信息表建立关联。

图书类目表（dwd.book_std_type_df）

字段	字段类型	字段定义	备注
book_id	bigint	图书 id	
book_name	character varying(50)	图书名称	
book_type_tag	bigint	图书类型编码	
book_type_name	character varying(50)	图书类型名称	
create_time	timestamp	创建时间	
modify_time	timestamp	更新时间	
create_date	date	创建日期	

某用户行为标签表开发案例一元数据介绍（4）

相关元数据介绍

WEB 端日志表（dwd.beacon_web_books_client_pv_log）：存放用户访问 web 页面信息及用户的 LBS 相关信息。通过在客户端做埋点，从日志数据解析出来。

WEB 端日志表（dwd.beacon_web_books_client_pv_log）

字段	字段类型	字段定义	备注
login_id	character varying(50)	设备登录名	设备记录的用户登录名
user_id	character varying(50)	用户 id	
session_id	character varying(50)	设备 id	
visit_time	timestamp	访问时间	本次访问操作在日志表中生成时间
report_time	timestamp	上报时间	终端记录用户点击按钮时间
province	character varying(50)	用户所在省份	通过 IP 地址解析获取用户省份
city	character varying(50)	用户所在城市	通过 IP 地址解析获取用户城市
referer_url	character varying(50)	上一个页面 url	上一个访问页面地址
url	character varying(50)	当前页面 URL	当前访问页面的链接地址
client	character varying(50)	操作系统	mac/windows
date_id	date	登陆日期	YYYY-MM-DD
lon	character varying(50)	经度	用户设备登陆时所在经度
lat	character varying(50)	维度	用户设备登陆时所在维度

APP 端日志表（dwd.beacon_app_books_client_pv_log）：存放用户访问 APP 的相关信息及用户的 LBS 相关信息，通过在客户端埋点，从日志数据解析出来。

APP 端日志表（dwd.beacon_app_books_client_pv_log）

字段	字段类型	字段定义	备注
login_id	character varying(50)	设备登录名	设备记录的用户登录名
user_id	character varying(50)	用户 id	
session_id	character varying(50)	设备 ID	
date_id	date	访问日期	YYYY-MM-DD
visit_time	timestamp	访问时间	本次访问操作在日志表中生成时间
report_time	timestamp	上报时间	终端记录用户点击按钮时间
province	character varying(50)	用户所在省份	通过 IP 地址解析获取用户省份
city	character varying(50)	用户所在城市	通过 IP 地址解析获取用户城市
referer_url	character varying(50)	上一个页面 url	上一个访问页面地址
url	character varying(50)	当前页面 URL	当前访问页面的链接地址
client	character varying(50)	操作系统	android/ios/win
lon	character varying(50)	经度	用户设备登陆时所在经度
lat	character varying(50)	维度	用户设备登陆时所在维度

某用户行为标签表开发案例—元数据介绍（5）

相关元数据介绍

商品评论表（dwd.book_comment）：存放用户对商品的评论信息。

商品评论表（dwd.book_comment）

字段	字段类型	字段定义	备注
user_id	character varying(15)	用户 id	
user_name	character varying(15)	用户姓名	
user_content	character varying(64)	评论内容	
user_images	character varying(15)	评论图片	
status_id	bigint	评论状态	1:待审核 2:已审核 3:已屏蔽
order_code	integer	订单 id,订单对应编号	
create_time	character varying(15)	创建时间	
create_date	date	创建日期	
content_ip	character varying(15)	评论用户 ip	
modify_time	timestamp	更新时间	

搜索日志表（dwd.app_search_log）：存放用户在 APP 端搜索相关的日志数据。

搜索日志表（dwd.app_search_log）

字段	字段类型	字段定义	备注
login_id	character varying(15)	设备登录名	设备记录的用户登录名
user_id	character varying(15)	用户 id	
session_id	character varying(15)	设备 id	
search_rad	character varying(15)	搜索 id	
date_id	date	搜索日期	
visit_time	timestamp	搜索时间	
search_q	character varying(15)	用户搜索的关键词	
tag_name	character varying(15)	标签内容	用户搜索关键词切词后与标签库模糊匹配到的标签内容
random_id	character varying(15)	每个访次的随机数	

某用户行为标签表开发案例—元数据介绍（6）

相关元数据介绍

用户收藏表（dwd.book_collection_df）：记录用户收藏图书数据。

用户收藏表（dwd.book_collection_df）

字段	字段类型	字段定义	备注
user_id	character varying(15)	用户 id	
create_date	date	收藏日期	
creat_time	timestamp	收藏时间	
book_id	bigint	图书 id	
book_name	character varying(50)	图书名称	
status_id	bigint	收藏状态	1：收藏　0：取消收藏
modify_date	date	修改日期	
modify_time	timestamp	修改时间	

购物车信息表（dwd. book_shopping_cart_df）：记录用户将图书加入购物车的数据。

购物车信息表（dwd. book_shopping_cart_df）

字段	字段类型	字段定义	备注
user_id	character varying(15)	用户 id	
book_id	bigint	图书 id	
book_name	character varying(50)	图书名称	
quantity	bigint	图书数量	
create_date	date	创建日期	
creat_time	timestamp	创建时间	
status_id	bigint	图书状态	1:加入购物车　0:移出购物车
modify_date	date	修改日期	
modify_time	timestamp	修改时间	

本案例相关元数据汇总

1. 用户信息表 (dwd.user_basic_info)
2. 商品订单表 (dwd.gdm_ord_order)
3. 图书信息表 (dwd.book_base_basic_info)
4. 图书类目表 (dwd.book_std_type_df)
5. WEB端日志表 (dwd.beacon_web_books_client_pv_log)
6. APP端日志表 (dwd.beacon_app_books_client_pv_log)
7. 商品评论表 (dwd.book_comment)
8. 用户收藏表 (dwd.book_collection_df)
9. 购物车信息表 (dwd.book_shopping_cart_df)
10. 搜索日志表 (dwd.app_search_log)

某用户行为标签表开发案例一标签开发方式

[**数据来源**]：用户行为属性标签主要来自用户属性标签表、商品订单表、图书信息表、图书类目表、WEB端日志表、APP端日志表、商品评论表、用户收藏表、购物车信息表、搜索日志表、行为权重配置表

开发表依赖关系

用户行为标签表结构

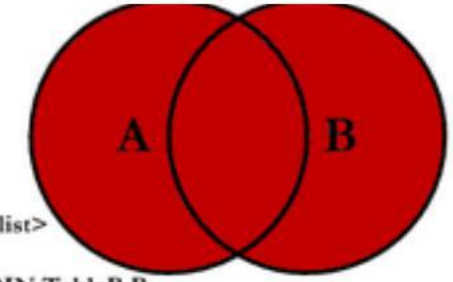
标签名	标签解释	示例	数据来源
用户 id	用户 id	xxxxxxxxx	用户属性标签表
标签 id	标签 id	xxxxxxxxx	图书信息表
标签名称	标签名称	钢铁是怎样炼成的	图书信息表
用户行为次数	用户当日与该标签相关行为次数	3	WEB 端日志表、APP 端日志表、商品评论表等
日期	用户行为产生该标签的日期	2017-08-01	WEB 端日志表、APP 端日志表、商品评论表
用户行为类型	用户通过哪些行为带来的标签	浏览、搜索、收藏、购买、评论等	商品订单表、WEB 端日志表、APP 端日志表、搜索日志表、商品评论表、用户收藏表等
标签类型	标签类型	图书、作者、出版社等	标签类型维表
标签权重	用户该标签权重值	0.7877	使用综合打分法计算该标签的权重值

某用户行为标签表开发案例—数据开发（1）

表结构、数据存储结构

确定画像相关的表结构，包含哪些字段，这些字段都是什么数据类型，例如用户行为标签表创建：

```
drop table if exists dwd.persona_user_tag_relation_public;
create table dwd.persona_user_tag_relation_public    --用户行为标签表
    (
      user_id    string comment '用户编码',
      tag_id   string comment '标签id',
      tag_name string comment '标签名称',
      cnt    int comment '行为次数',
      date_id    timestamp comment '行为日期',
      tag_type_id    int comment '标签类型',
      act_type_id int  comment '行为类型'
    )
comment '用户画像-用户行为标签表';
```




```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```

- 数据按日期维度分区存储，例如用户行为类标签需要每天跑批昨日产生的数据，并增量更新到已有数据的基础上；
- 一般设定日期分区，以历史数据所在表作为主表，*FULL OUT JOIN*（如左图）昨日新产生的数据，将新数据追加到历史数据上。并将追加后的数据写入到当日所在的分区。

举个例子：今天是2018-04-22日，昨日为21日，今日跑批任务时，需要将21日产生的增量数据与20日所在的日期分区数据做全连接（20日数据为历史全量，作为主表），将全连接后的得到的新的数据写入到21日所在的日期分区下。

某用户行为标签表开发案例—数据开发（2）

开发过程中需要建立一些中间表

在开发用户行为标签表，将用户与图书相关行为的标签打在用户身上的过程中，需要建立系列临时表，下面分5步详细介绍用户个性化标签表的建立过程

Step1：建立临时表获取图书和图书类型的信息

从图书信息表和图书类型表中抽取图书id、图书名称、图书类型等信息建立图书相关信息的临时表1，其中图书信息表和图书类型维表通过图书id相关联

```
drop table if exists dwd.persona_user_tag_relation_public_01;
create table dwd.persona_user_tag_relation_public_01
as
select t1.book_id,          --图书编码
       t1.book_name,        --图书名称
       t2.book_type_tag,    --图书类型编码
       t2.book_type_name    --图书类型名称
  from dwd.book_base_basic_info  t1     -- 图书信息表
inner join dwd.book_std_type_df  t2       -- 图书类目表
   on t1.book_id = t2.book_id     -- 通过图书id两表相关联
 where t2.book_type_name not in  ('未定义', '其他')
group by t1.book_id,
         t1.book_name,
         t2.book_type_tag,
         t2.book_type_name
```

某用户行为标签表开发案例—数据开发（3）

Step2：建立临时表从日志数据中提取用户浏览信息

日志数据对用户的每一次操作行为都进行了记录，对于探究用户行为偏好具有非常重要的意义。例如用户在WEB端浏览过某个图书详情页，对应的在WEB日志表中记录该次浏览行为，包括时间、日期、页面url、来源页url、用户id、设备id、ip地址等数据。

一般商品页面链接中包含有该商品id的参数，通过对url进行解析可以找到该商品的id。

```
drop table if exists dwd.persona_user_tag_relation_public_02;
create table dwd.persona_user_tag_relation_public_02
as
select  t1.user_id,
        t1.date_id,
        t1.book_id,
       count(1) as cnt
  from (
        select user_id as user_id,
               date_id as date_id,
               regexp_extract(parse_url(url,'PATH','.*/(.*?)$',1)) as book_id
          from dwd.beacon_web_books_client_pv_log    --web页面访问表
         where date_id = date_sub(from_unixtime(unix_timestamp(),'yyyy-MM-dd'),1)    --昨日
           and url like '%books.com/detail/%'
           and user_id <> ''
           and user_id <> '-'
      union all   --
        select user_id as user_id,
               date_id as date_id,
               regexp_extract(parse_url(url,'PATH','.*/(.*?)$',1)) as book_id
          from dwd.beacon_app_books_client_pv_log    --app页面访问表
         where date_id = date_sub(from_unixtime(unix_timestamp(),'yyyy-MM-dd'),1)    --昨日
           and url like '%books.com/detail/%'
           and user_id <> ''
           and user_id <> '-'
       ) t1
 where t1.book_id <> ''
group by t1.user_id,
        t1.date_id,
        t1.book_id
```

这里从日志相关的表中（包括APP日志表和WEB日志表）获取用户浏览图书对应的页面链接，通过正则表达式匹配出用户浏览图书页面链接所对应的图书id

某用户行为标签表开发案例—数据开发（4）

Step3：将用户行为产生的图书标签插入到用户行为标签表中

用户行为标签表记录了用户在平台上购买、浏览、评论、收藏、取消收藏、放入购物车、搜索等各种行为过程所带来的标签。
需要开发的表结构包括用户编码、标签id、标签名称、用户行为类型、标签类型、行为日期、行为次数共计7个字段。
开发过程中需要将相关表的数据经过抽取、清洗后插入到用户行为标签表中。

关于行为标签表的7个字段，各字段的释义如下：

- 用户id（user_id）：用户唯一id；
- 标签id（tag_id）：图书id；
- 标签名称（tag_name）：图书名称；
- 用户行为次数（cnt）：用户当日产生该标签的次数，如用户当日浏览一本图书4次，则记录4；
- 行为日期（date_id）：产生该条标签对应日期；
- 标签类型（tag_type_id）：在本案例中通过与图书类型表相关联，取出每本图书对应的类型，如《钢铁是怎么炼成的》对应“名著”；
- 用户行为类型（act_type_id）：即用户的购买、浏览、评论等操作行为，在本例中通过预设数值1~7来定义用户对应的行为类型。1：购买行为，2：浏览行为，3：评论行为，4：收藏行为，5：取消收藏行为，6：加入购物车行为，7：搜索行为；

下面通过7段代码来讲述如何将用户每一种行为所带来的标签插入到用户标签表中。

某用户行为标签表开发案例—数据开发（5）

行为类型1：用户购买图书行为带来的标签

这里将用户不同的行为类型，通过不同的数字标识写死

```
行为类型1：用户购买图书行为带来的标签，代码执行如下：
insert into dwd.persona_user_tag_relation_public     --建立用户行为标签表
     select t1.user_id as user_id,
            t2.book_id as tag_id,          --购买图书对应的图书id作为标签id
            t2.book_name as tag_name,
            count(1) as cnt,
            t1.create_date as date_id,
            t3.book_type_tag as tag_type_id,
            1 as act_type_id                --行为类型1
       from dwd.gdm_ord_order t1            --商品订单表
 inner join dwd.book_base_basic_info t2     --图书信息表
         on t1.std_book_id = t2.book_id
 inner join dwd.book_std_type_df t3         -- 图书类目表
         on t2.book_id = t3.book_id
      where t1.date_id  = date_sub(from_unixtime(unix_timestamp(),'yyyy-MM-dd'),1)   --昨日行为
        and t1.user_id <> ''
        and t1.user_id <> '-'
  group by t1.user_id
           t2.book_id,
           t2.book_name,
           t1.create_date,
           t3.book_type_tag,
           1
```

排除用户id为空的脏数据

每天跑定时任务时，自动获取前一期的日期

某用户行为标签表开发案例—数据开发（6）

行为类型2：用户浏览图书行为带来的标签

这里将用户不同的行为类型，通过不同的数字标识写死

```
行为类型2：用户浏览图书行为带来的标签，代码执行如下：
insert into dwd.persona_user_tag_relation_public
      select t1.user_id as user_id,
             t1.book_id as tag_id,
             t2.book_name as tag_name,
             count(1) as cnt,
             t1.date_id as date_id,
             t2.book_type_tag as tag_type_id,
             2 as act_type_id                    -- 行为类型2
        from dwd.persona_user_tag_relation_public_02 t1   --用户浏览图书信息表
  inner join dwd.persona_user_tag_relation_public_01 t2   --获取图书信息临时表
          on t1.book_id = t2.book_id
       where t1.date_id = date_sub(from_unixtime(unix_timestamp(),'yyyy-MM-dd'),1)
         and t1.user_id <> ''              --过滤用户id为空的脏数据
         and t1.user_id <> '-'             --过滤用户id为-的脏数据
    group by t1.user_id,
             t1.book_id,
             t2.book_name,
             t1.create_date,
             t2.book_type_tag,
             2
```

这里group by标签聚合，防止重复数据

某用户行为标签表开发案例—数据开发（7）

行为类型3：用户评论图书行为带来的标签

```
行为类型3：用户评论图书行为带来的标签，代码执行如下：
insert into dwd.persona_user_tag_relation_public
     select t1.user_id,
            t3.book_id as tag_id,
            t3.book_name as tag_name,
            count(1) as cnt,
            t1.create_date as date_id,
            t2.book_type_tag as tag_type_id,
            3 as act_type_id
       from dwd.book_comment t1       --商品评论表
 inner join dwd.gdm_ord_order t2      --商品订单表
         on t1.order_code = t2.order_id      --订单id相关联
 inner join dwd.persona_user_tag_relation_public_01 t3    --图书信息临时表
         on t2.std_book_id = t3.book_id
      where t1.create_date = date_sub(from_unixtime(unix_timestamp(),'yyyy-MM-dd'),1)
        and t1.status_id = 2          --评论状态:已审核
        and t1.user_id <> ''          --过滤用户id为空的脏数据
        and t1.user_id <> '-'         --过滤用户id为-的脏数据
  group by t1.user_id,
            t3.book_id,
            t3.book_name,
            t1.create_date,
            t2.book_type_tag,
            3
```

上述代码中在解析评论带来的图书标签时，首先需要将商品评论表和商品订单表关联(通过订单*id*)，然后从商品订单表中找到对应的图书*id*。

某用户行为标签表开发案例—数据开发（8）

行为类型4：用户收藏图书行为带来的标签

```
行为类型4：用户收藏图书行为带来的标签，代码执行如下：
insert into dwd.persona_user_tag_relation_public
     select t1.user_id as user_id,
            t1.book_id as tag_id,
            t2.book_name as tag_name,
            count(1) as cnt,
            t1.create_date as date_id,
            t2.book_type_tag as tag_type_id,
            4 as act_type_id
       from dwd.book_collection_df t1       --用户收藏表
 inner join dwd.persona_user_tag_relation_public_01 t2     --获取图书信息临时表
         on t1.book_id = t2.book_id
      where t1.date_id = date_sub(from_unixtime(unix_timestamp(),'yyyy-MM-dd'),1)
        and t1.status_id = 1    --状态 1 收藏
        and t1.user_id <> ''
        and t1.user_id <> '-'
  group by t1.user_id,
           t1.book_id,
           t2.book_name,
           t1.create_date,
           t2.book_type_tag,
           4
```

某用户行为标签表开发案例—数据开发（9）

行为类型5：用户取消收藏图书行为带来的标签

```
行为类型5：用户取消收藏图书行为带来的标签，代码执行如下：
insert into  dwd.persona_user_tag_relation_public
      select t1.user_id as user_id,
             t1.book_id as tag_id,
             t2.book_name as tag_name,
             count(1) as cnt,
             t1.create_date as date_id,
             t2.book_type_tag as tag_type_id,
             5 as act_type_id
        from dwd.book_collection_df t1        --用户收藏标签表
  inner join dwd.persona_user_tag_relation_public_01 t2     --获取图书信息临时表
          on t1.book_id = t2.book_id
       where t1.date_id = date_sub(from_unixtime(unix_timestamp(),'yyyy-MM-dd'),1)
         and t1.status_id = 0    --状态 0 取消收藏
         and t1.user_id <> ''
         and t1.user_id <> '-'
    group by t1.user_id
             t1.book_id,
             t2.book_name,
             t1.create_date,
             t2.book_type_tag,
             5
```

某用户行为标签表开发案例—数据开发（10）

行为类型6：用户加入购物车行为带来的标签

```
行为类型6：用户加入购物车行为带来的标签，代码执行如下：
insert into dwd.persona_user_tag_relation_public
      select t1.user_id as user_id,
             t1.book_id as tag_id,
             t2.book_name as tag_name,
             count(1) as cnt,
             t1.create_date as date_id,
             t2.book_type_tag as tag_type_id,
             6 as act_type_id                    --用户行为类型固定写死
        from dwd.book_shopping_cart_df t1    --购物车信息表
  inner join dwd.persona_user_tag_relation_public_01 t2     --获取图书信息临时表
          on t1.book_id = t2.book_id
       where t1.date_id = date_sub(from_unixtime(unix_timestamp(),'yyyy-MM-dd'),1)
         and t1.user_id <> ''
         and t1.user_id <> '-'
         and t1.status_id = 1    --状态 1 加入购物车
    group by t1.user_id
             t1.book_id,
             t2.book_name,
             t1.create_date,
             t2.book_type_tag,
             6
```

某用户行为标签表开发案例—数据开发（11）

行为类型7：用户搜索图书行为带来的标签

```
行为类型7：用户搜索行为带来的标签，代码执行如下：
insert into dwd.persona_user_tag_relation_public
    select t.user_id,
           t.tag_id,
           t.tag_name,
           t.cnt,
           t.date_id,
           t.tag_type_id,
           t.act_type_id
     from (
          select t1.user_id,
                 t2.book_id as tag_id,
                 t2.book_name as tag_name,
                 count(1) as cnt
                 t1.date_id,    --搜索日期
                 t3.book_type_tag as tag_type_id,
                 7 as act_type_id
            from dwd.app_search_log t1     --搜索日志表
           inner join dwd.book_base_basic_info t2   --图书信息表
              on t1.tag_name = t2.book_name   --搜索匹配到的标签与图书名称相关联
           inner join dwd.persona_user_tag_relation_public_01 t3     --图书信息临时表
              on t2.book_id = t3.book_id
           where t1.date_id = date_sub(from_unixtime(unix_timestamp(),'yyyy-MM-dd'),1)
             ) t
   group by t.user_id,
           t.tag_id,
           t.tag_name,
           t.cnt,
           t.date_id,
           t.tag_type_id,
           t.act_type_id
```

将业务数据与日志数据相关联，进行挖掘

到这里，用户个性化标签表的创建工作就完成了。用户个性化标签表记录了用户在WEB/APP端每一次操作带来标签的明细数据，并且该表每天增量更新昨天产生的数据（数仓T+1天更新数据）