电商业务常用指标分析之SQL实现

当构建好电商业务数仓之后，需要对业务需要的指标进行计算，从而进一步进行报表的展示，那么，电商的业务知识大概涉及哪些？关于电商业务的常用指标计算都有哪些？这些常用的指标该如何通过Hive数仓进行分析？本文将进行一一梳理.

电商业务领域知识梳理

- **用户**

用户以设备为判断标准，在移动统计中，每个独立设备认为是一个独立用户。Android系统根据IMEI号，IOS系统根据OpenUDID来标识一个独立用户，每部手机一个用户。

- **新增用户**

首次联网使用应用的用户。如果一个用户首次打开某APP，那这个用户定义为新增用户；卸载再安装的设备，不会被算作一次新增。新增用户包括日新增用户、周新增用户、月新增用户。

- **活跃用户**

打开应用的用户即为活跃用户，不考虑用户的使用情况。每天一台设备打开多次会被计为一个活跃用户。

- **周（月）活跃用户**

某个自然周（月）内启动过应用的用户，该周（月）内的多次启动只记一个活跃用户。

- **月活跃率**

月活跃用户与截止到该月累计的用户总和之间的比例。

- **沉默用户**

用户仅在安装当天（次日）启动一次，后续时间无再启动行为。该指标可以反映新增用户质量和用户与APP的匹配程度。

- **版本分布**

不同版本的周内各天新增用户数，活跃用户数和启动次数。利于判断APP各个版本之间的优劣和用户行为习惯。

- **本周回流用户**

上周未启动过应用，本周启动了应用的用户。

- **连续N周活跃用户**

连续n周，每周至少启动一次。

- **忠诚用户**

连续活跃5周以上的用户

- **连续活跃用户**

连续2周及以上活跃的用户

- **近期流失用户**

连续n(2<= n <= 4)周没有启动应用的用户。（第n+1周没有启动过）

- **留存用户**

某段时间内的新增用户，经过一段时间后，仍然使用应用的被认作是留存用户；这部分用户占当时新增用户的比例即是留存率。
例如，5月份新增用户200，这200人在6月份启动过应用的有100人，7月份启动过应用的有80人，8月份启动过应用的有50人；则5月份新增用户一个月后的留存率是50%，二个月后的留存率是40%，三个月后的留存率是25%。

- **用户新鲜度**

每天启动应用的新老用户比例，即新增用户数占活跃用户数的比例。

- **单次使用时长**

每次启动使用的时间长度。

- **日使用时长**

累计一天内的使用时间长度。

- **启动次数计算标准**

IOS平台应用退到后台就算一次独立的启动；Android平台我们规定，两次启动之间的间隔小于30秒，被计算一次启动。用户在使用过程中，若因收发短信或接电话等退出应用30秒又再次返回应用中，那这两次行为应该是延续而非独立的，所以可以被算作一次使用行为，即一次启动。业内大多使用30秒这个标准，但用户还是可以自定义此时间间隔。

常用的日期函数处理

- **date_format函数（根据格式整理日期）**

```
select date_format('2019-12-05','yyyy-MM');
```

输出：2019-12

- **date_add函数（加减日期）**

```
select date_add('2019-12-05',-1);
```

输出：2019-12-04

```
select date_add('2019-12-05',1);
```

输出：2019-12-06

- **next_day函数(返回当前时间的下一个星期X所对应的日期)**

1）取当前天的下一个周一

```
select next_day('2019-12-05','MO')
```

输出：2019-12-09

说明：星期一到星期日的英文（Monday，Tuesday、Wednesday、Thursday、Friday、Saturday、Sunday）

2）取当前周的周一

```
select date_add(next_day('2019-12-05','MO'),-7);
```

输出：2019-12-02

- **last_day函数（返回这个月的最后一天的日期）**

```
select last_day('2019-12-05');
```

输出：2019-12-31

业务指标分析

用户活跃相关指标分析

数仓的DWS层会建立好每日的活跃用户表明细、每周的活跃用户表明细以及每月的活跃用户明细表。

- **每日活跃用户明细表结构**

每天一个分区，存储当天的日活明细，该表根据mid_id进行去重。

```
create external table 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'
;
```

- **每周活跃用户明细表**

根据日用户访问明细，获得周用户访问明细,周明细表按周一日期和周末日期拼接字段进行分区。即每个分区存储的是本周内的活跃用户明细，该表按mid_id进行去重，即一周内获取多次，只记录一条记录。

```
create external table 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/'
;
```

- **每月活跃用户明细**

该表按月进行分区，并按mid_id去重，数据来源与日活明细表

```
create external table 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/'
;
```

- **建立ADS层的活跃用户指标表**

```
create external table 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/'
;
```

SQL具体实现：

```
insert into table ads_uv_count
select
  '2019-02-10' dt,
   daycount.ct,
   wkcount.ct,
```

```
   mncount.ct,
   if(date_add(next_day('2019-02-10','MO'),-1)='2019-02-10','Y','N') , -- 判断跑任务的当天是
否是周末
   if(last_day('2019-02-10')='2019-02-10','Y','N')  -- 判断跑任务的当天是否是月末
from
(
-- 计算当天的日活
   select
      '2019-02-10' dt,
       count(*) ct
   from dws_uv_detail_day
   where dt='2019-02-10'
)daycount join
(
-- 计算当天所属周的周活
   select
     '2019-02-10' dt,
     count (*) ct
   from dws_uv_detail_wk
   where wk_dt=concat(date_add(next_day('2019-02-10','MO'),-7),'_' ,date_add(next_day('2
19-02-10','MO'),-1) )
) wkcount on daycount.dt=wkcount.dt
join
(
-- 计算当天所属月的月活
   select
     '2019-02-10' dt,
     count (*) ct
   from dws_uv_detail_mn
   where mn=date_format('2019-02-10','yyyy-MM')
)mncount on daycount.dt=mncount.dt
;
```

新增用户指标分析

首次联网使用应用的用户。如果一个用户首次打开某APP，那这个用户定义为新增用户；卸载再安装的设备，不会被算作一次新增。新增用户包括日新增用户、周新增用户、月新增用户。

- **每日新增用户明细表**

每日新增用户明细表来源于每天的日活表，使用每天的日活表去LEFT JOIN 每天新增用户明细表，关联的条件是mid_id,筛选条件为，每日新增设备表中为空

```
create external table 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/';
```

- 每日新增用户表

```
create external table 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/';
```

每日新增用户表装载SQL实现

```
insert into table ads_new_mid_count
select
create_date,
count(*)
from dws_new_mid_day
where create_date='2019-02-10'
group by create_date;
```

用户留存指标分析

留存用户：某段时间内的新增用户（活跃用户），经过一段时间后，又继续使用应用的被认作是留存用户；

留存率：留存用户占当时新增用户（活跃用户）的比例即是留存率。

例如，2月10日新增用户100，这100人在2月11日启动过应用的有30人，2月12日启动过应用的有25人，2月13日启动过应用的有32人；

则2月10日新增用户次日的留存率是30/100＝30%，两日留存率是25/100=25%，三日留存率是32/100=32%。

时间	新增用户	1天后	2天后	3天后
2019-02-10	100	30% (2-11)	25% (2-12)	32% (2-13)
2019-02-11	200	20% (2-12)	15% (2-13)	
2019-02-12	100	25% (2-13)		
2019-02-13				

- 每日用户留存明细

该表以天作为分区，每天计算前1天的新用户访问留存明细，

```
create external table 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/'
;
```

每日用户留存明细装载语句

```
insert overwrite table dws_user_retention_day
partition(dt="2019-02-11")
select
    nm.mid_id,
    nm.user_id ,
    nm.version_code ,
    nm.version_name ,
    nm.lang ,
    nm.source,
    nm.os,
    nm.area,
    nm.model,
    nm.brand,
    nm.sdk_version,
    nm.gmail,
    nm.height_width,
    nm.app_time,
    nm.network,
    nm.lng,
    nm.lat,
nm.create_date,
1 retention_day
from dws_uv_detail_day ud join dws_new_mid_day nm   on ud.mid_id =nm.mid_id
where ud.dt='2019-02-11' and nm.create_date=date_add('2019-02-11',-1);
```

- **留存用户数**

```
create external table 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/';
```

留存用户数装载SQL

```
insert into table ads_user_retention_day_count
select
    create_date,
    retention_day,
    count(*) retention_count
from dws_user_retention_day
where dt='2019-02-11'
group by create_date,retention_day;
```

流失用户数分析

流失用户：最近7天未登录我们称之为流失用户

- **流失用户数表**

```
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';
```

装载SQL,如果统计日期为2019-02-20，则7天未登陆的用户数的计算逻辑为：

查询日活表，并按mid_id进行分组，并且设备的最近访问时间小于等于当前时间的一周前，即活跃的最大日期(最近一次访问日期)小于等于2019-02-20

```
insert into table ads_wastage_count
select
     '2019-02-20',
     count(*)
from
(
    select mid_id
from dws_uv_detail_day
    group by mid_id
    having max(dt)<=date_add('2019-02-20',-7)
)t1;
```

最近七天内连续三天活跃用户数指标分析

- **最近七天内连续三天活跃用户数表**

需要使用日活表，来获取最近7天内连续3天活跃用户数

```
create external table 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';
```

装载语句为：

```
insert into table ads_continuity_uv_count
select
    '2019-02-12',
    concat(date_add('2019-02-12',-6),'_','2019-02-12'),
    count(*)
from

        (
        select
        mid_id
        from
                ( -- 筛选出连续3的活跃用户，可能存在重复
                select
                        mid_id

                from
                        (
                                -- 计算活跃用户的活跃日期与其排名的差值
                                select

                                        mid_id,
                                        date_sub(dt,rank) date_dif

                                from
                                        (-- 查询出最近7天的活跃用户，并对活跃日期进行排名
                                                select
                                                        mid_id,
                                                        dt,
                                                        rank() over (partition by mid_id
order by dt) rank

                                                from dws_uv_detail_day
                                                where dt >= date_add('2019-02-12',-6) an
 dt <= '2109-02-12'
                                        ) t1
                        ) t2
                group by mid_id,date_dif -- 对用户设备id和差值进行分组
                having count(*) >=3   -- 统计大于等于3的差值数据筛选出来

                ) t3
        group by mid_id -- 对mid_id进行去重
        ) t4
```