# 第10章 需求四：沉默用户数

沉默用户：指的是只在安装当天启动过，且启动时间是在一周前

## 10.1 DWS层

使用日活明细表dws\_uv\_detail\_day作为DWS层数据

## 10.2 ADS层



1）建表语句

hive (gmall)>

drop table if exists ads\_slient\_count;

create external table ads\_slient\_count(

`dt` string COMMENT '统计日期',

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

)

stored as parquet

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

2）导入2019-02-20数据

hive (gmall)>

insert into table ads\_slient\_count

select

'2019-02-20' dt,

count(\*) slient\_count

from

(

select mid\_id

from dws\_uv\_detail\_day

where dt<='2019-02-20'

group by mid\_id

having count(\*)=1 and min(dt)<date\_add('2019-02-20',-7)

) t1;

3）查询导入数据

hive (gmall)> select \* from ads\_slient\_count;

## 10.3 编写脚本

1）在hadoop102的/home/atguigu/bin目录下创建脚本

[atguigu@hadoop102 bin]$ vim ads\_slient\_log.sh

在脚本中编写如下内容

#!/bin/bash

hive=/opt/module/hive/bin/hive

APP=gmall

if [ -n "$1" ];then

do\_date=$1

else

do\_date=`date -d "-1 day" +%F`

fi

echo "-----------导入日期$do\_date-----------"

sql="

insert into table "$APP".ads\_slient\_count

select

'$do\_date' dt,

count(\*) slient\_count

from

(

select

mid\_id

from "$APP".dws\_uv\_detail\_day

where dt<='$do\_date'

group by mid\_id

having count(\*)=1 and min(dt)<=date\_add('$do\_date',-7)

)t1;"

$hive -e "$sql"

2）增加脚本执行权限

[atguigu@hadoop102 bin]$ chmod 777 ads\_slient\_log.sh

3）脚本使用

[atguigu@hadoop102 module]$ ads\_slient\_log.sh 2019-02-20

4）查询结果

hive (gmall)> select \* from ads\_slient\_count;

5）脚本执行时间

企业开发中一般在每日凌晨30分~1点

# 第11章 需求五：本周回流用户数

本周回流=本周活跃-本周新增-上周活跃

## 11.1 DWS层

使用日活明细表dws\_uv\_detail\_day作为DWS层数据

## 11.2 ADS层



1）建表语句

hive (gmall)>

drop table if exists 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'

)

stored as parquet

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

2）导入数据：

hive (gmall)>

insert into table ads\_back\_count

select

'2019-02-20' dt,

concat(date\_add(next\_day('2019-02-20','MO'),-7),'\_',date\_add(next\_day('2019-02-20','MO'),-1)) wk\_dt,

count(\*)

from

(

select t1.mid\_id

from

(

select mid\_id

from dws\_uv\_detail\_wk

where wk\_dt=concat(date\_add(next\_day('2019-02-20','MO'),-7),'\_',date\_add(next\_day('2019-02-20','MO'),-1))

)t1

left join

(

select mid\_id

from dws\_new\_mid\_day

where create\_date<=date\_add(next\_day('2019-02-20','MO'),-1) and create\_date>=date\_add(next\_day('2019-02-20','MO'),-7)

)t2

on t1.mid\_id=t2.mid\_id

left join

(

select mid\_id

from dws\_uv\_detail\_wk

where wk\_dt=concat(date\_add(next\_day('2019-02-20','MO'),-7\*2),'\_',date\_add(next\_day('2019-02-20','MO'),-7-1))

)t3

on t1.mid\_id=t3.mid\_id

where t2.mid\_id is null and t3.mid\_id is null

)t4;

3）查询结果

hive (gmall)> select \* from ads\_back\_count;

## 11.3 编写脚本

1）在hadoop102的/home/atguigu/bin目录下创建脚本

[atguigu@hadoop102 bin]$ vim ads\_back\_log.sh

在脚本中编写如下内容

#!/bin/bash

if [ -n "$1" ];then

do\_date=$1

else

do\_date=`date -d "-1 day" +%F`

fi

hive=/opt/module/hive/bin/hive

APP=gmall

echo "-----------导入日期$do\_date-----------"

sql="

insert into table "$APP".ads\_back\_count

select

'$do\_date' dt,

concat(date\_add(next\_day('$do\_date','MO'),-7),'\_',date\_add(next\_day('$do\_date','MO'),-1)) wk\_dt,

count(\*)

from

(

select t1.mid\_id

from

(

select mid\_id

from "$APP".dws\_uv\_detail\_wk

where wk\_dt=concat(date\_add(next\_day('$do\_date','MO'),-7),'\_',date\_add(next\_day('$do\_date','MO'),-1))

)t1

left join

(

select mid\_id

from "$APP".dws\_new\_mid\_day

where create\_date<=date\_add(next\_day('$do\_date','MO'),-1) and create\_date>=date\_add(next\_day('$do\_date','MO'),-7)

)t2

on t1.mid\_id=t2.mid\_id

left join

(

select mid\_id

from "$APP".dws\_uv\_detail\_wk

where wk\_dt=concat(date\_add(next\_day('$do\_date','MO'),-7\*2),'\_',date\_add(next\_day('$do\_date','MO'),-7-1))

)t3

on t1.mid\_id=t3.mid\_id

where t2.mid\_id is null and t3.mid\_id is null

)t4;

"

$hive -e "$sql"

2）增加脚本执行权限

[atguigu@hadoop102 bin]$ chmod 777 ads\_back\_log.sh

3）脚本使用

[atguigu@hadoop102 module]$ ads\_back\_log.sh 2019-02-20

4）查询结果

hive (gmall)> select \* from ads\_back\_count;

5）脚本执行时间

企业开发中一般在每周一凌晨30分~1点

# 第12章 需求六：流失用户数

流失用户：最近7天未登录我们称之为流失用户

## 12.1 DWS层

使用日活明细表dws\_uv\_detail\_day作为DWS层数据

## 12.2 ADS层



1）建表语句

hive (gmall)>

drop table if exists 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'

)

stored as parquet

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

2）导入2019-02-20数据

hive (gmall)>

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;

## 12.3 编写脚本

1）在hadoop102的/home/atguigu/bin目录下创建脚本

[atguigu@hadoop102 bin]$ vim ads\_wastage\_log.sh

在脚本中编写如下内容

#!/bin/bash

if [ -n "$1" ];then

do\_date=$1

else

do\_date=`date -d "-1 day" +%F`

fi

hive=/opt/module/hive/bin/hive

APP=gmall

echo "-----------导入日期$do\_date-----------"

sql="

insert into table "$APP".ads\_wastage\_count

select

'$do\_date',

count(\*)

from

(

select mid\_id

from "$APP".dws\_uv\_detail\_day

group by mid\_id

having max(dt)<=date\_add('$do\_date',-7)

)t1;

"

$hive -e "$sql"

2）增加脚本执行权限

[atguigu@hadoop102 bin]$ chmod 777 ads\_wastage\_log.sh

3）脚本使用

[atguigu@hadoop102 module]$ ads\_wastage\_log.sh 2019-02-20

4）查询结果

hive (gmall)> select \* from ads\_wastage\_count;

5）脚本执行时间

企业开发中一般在每日凌晨30分~1点

# 第13章 需求七：最近连续3周活跃用户数

最近3周连续活跃的用户：通常是周一对前3周的数据做统计，该数据一周计算一次。

## 13.1 DWS层

使用周活明细表dws\_uv\_detail\_wk作为DWS层数据

## 13.2 ADS层



1）建表语句

hive (gmall)>

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

create external table 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

)

stored as parquet

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

2）导入2019-02-20所在周的数据

hive (gmall)>

insert into table ads\_continuity\_wk\_count

select

'2019-02-20',

concat(date\_add(next\_day('2019-02-20','MO'),-7\*3),'\_',date\_add(next\_day('2019-02-20','MO'),-1)),

count(\*)

from

(

select mid\_id

from dws\_uv\_detail\_wk

where wk\_dt>=concat(date\_add(next\_day('2019-02-20','MO'),-7\*3),'\_',date\_add(next\_day('2019-02-20','MO'),-7\*2-1))

and wk\_dt<=concat(date\_add(next\_day('2019-02-20','MO'),-7),'\_',date\_add(next\_day('2019-02-20','MO'),-1))

group by mid\_id

having count(\*)=3

)t1;

3）查询

hive (gmall)> select \* from ads\_continuity\_wk\_count;

## 13.3 编写脚本

1）在hadoop102的/home/atguigu/bin目录下创建脚本

[atguigu@hadoop102 bin]$ vim ads\_continuity\_wk\_log.sh

在脚本中编写如下内容

#!/bin/bash

if [ -n "$1" ];then

do\_date=$1

else

do\_date=`date -d "-1 day" +%F`

fi

hive=/opt/module/hive/bin/hive

APP=gmall

echo "-----------导入日期$do\_date-----------"

sql="

insert into table "$APP".ads\_continuity\_wk\_count

select

'$do\_date',

concat(date\_add(next\_day('$do\_date','MO'),-7\*3),'\_',date\_add(next\_day('$do\_date','MO'),-1)),

count(\*)

from

(

select mid\_id

from "$APP".dws\_uv\_detail\_wk

where wk\_dt>=concat(date\_add(next\_day('$do\_date','MO'),-7\*3),'\_',date\_add(next\_day('$do\_date','MO'),-7\*2-1))

and wk\_dt<=concat(date\_add(next\_day('$do\_date','MO'),-7),'\_',date\_add(next\_day('$do\_date','MO'),-1))

group by mid\_id

having count(\*)=3

)t1;"

$hive -e "$sql"

2）增加脚本执行权限

[atguigu@hadoop102 bin]$ chmod 777 ads\_continuity\_wk\_log.sh

3）脚本使用

[atguigu@hadoop102 module]$ ads\_continuity\_wk\_log.sh 2019-02-20

4）查询结果

hive (gmall)> select \* from ads\_continuity\_wk\_count;

5）脚本执行时间

企业开发中一般在每周一凌晨30分~1点

# 第14章 需求八：最近七天内连续三天活跃用户数

说明：最近7天内连续3天活跃用户数

## 14.1 DWS层

使用日活明细表dws\_uv\_detail\_day作为DWS层数据

## 14.2 ADS层



1）建表语句

hive (gmall)>

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

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'

stored as parquet

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

2）写出导入数据的SQL语句

hive (gmall)>

insert into table ads\_continuity\_uv\_count

select

'2019-02-20',

concat(date\_add('2019-02-20',-6),'\_','2019-02-20') dt,

count(\*)

from

(

select mid\_id

from

(

select

mid\_id,

date\_sub(dt,rank) date\_diff

from

(

select

mid\_id,

dt,

rank() over(partition by mid\_id order by dt) rank

from dws\_uv\_detail\_day

where dt>=date\_add('2019-02-20',-6) and dt<='2019-02-20'

)t1

)t2

group by mid\_id,date\_diff

having count(\*)>=3

)t3;

(5)查询

hive (gmall)> select \* from ads\_continuity\_uv\_count;

## 14.3 编写脚本

1）在hadoop102的/home/atguigu/bin目录下创建脚本

[atguigu@hadoop102 bin]$ vim ads\_continuity\_log.sh

在脚本中编写如下内容

#!/bin/bash

if [ -n "$1" ];then

do\_date=$1

else

do\_date=`date -d "-1 day" +%F`

fi

hive=/opt/module/hive/bin/hive

APP=gmall

echo "-----------导入日期$do\_date-----------"

sql="

insert into table "$APP".ads\_continuity\_uv\_count

select

'$do\_date',

concat(date\_add('$do\_date',-6),'\_','$do\_date') dt,

count(\*)

from

(

select

mid\_id

from

(

select

mid\_id,

date\_sub(dt,rank) date\_diff

from

(

select

mid\_id,

dt,

rank() over(partition by mid\_id order by dt) rank

from "$APP".dws\_uv\_detail\_day

where dt>=date\_add('$do\_date',-6) and dt<='$do\_date'

)t1

)t2

group by mid\_id,date\_diff

having count(\*)>=3

)t3;

"

$hive -e "$sql"

2）增加脚本执行权限

[atguigu@hadoop102 bin]$ chmod 777 ads\_continuity\_log.sh

3）脚本使用

[atguigu@hadoop102 module]$ ads\_continuity\_log.sh 2019-02-20

4）查询结果

hive (gmall)> select \* from ads\_continuity\_uv\_count;

5）脚本执行时间

企业开发中一般在每日凌晨30分~1点