作业一、

用拉链表实现核心交易分析中DIM层商家维表，并实现该拉链表的回滚

解答：

1、定义表加载数据

```
-- 商家纬度表（dim层，由商家店铺表、商家地域组织表构成）
drop table if exists dim.dim_trade_shops_org;
create table dim.dim_trade_shops_org(
shopid int,
shopName string,
cityId int,
cityName string ,
regionId int ,
regionName string
)
PARTITIONED BY (dt string)
STORED AS PARQUET;

-- 数据(已存在在表中)
100060,同仁xxx大健康,100211,六安市分公司,100006,华北大区,2020-07-01
100059,乐居xxx日用品,100225,景德镇市分公司,100006,华北大区,2020-07-01
100058,良子xxx铺美食,100329,崇左市分公司,100008,华南大区,2020-07-01
100057,三只xxx鼠零食,100311,东莞市分公司,100008,华南大区,2020-07-01
100056,OPxxx自营店,100050,阜新市分公司,100006,华北大区,2020-07-01
100055,苹果xxx旗舰店,100211,六安市分公司,100006,华北大区,2020-07-01
100054,小米xxx旗舰店,100159,石嘴山市分公司,100007,华西大区,2020-07-01
100053,华为xxx旗舰店,100011,石家庄市分公司,100006,华北大区,2020-07-01
100052,新鲜xxx旗舰店,100236,青岛市分公司,100006,华北大区,2020-07-01
100050,WSxxx营超市,100225,景德镇市分公司,100006,华北大区,2020-07-01

100056,小米xxx自营店,100050,阜新市分公司,100006,华北大区,2020-07-02
100055,小米xxx旗舰店,100211,六安市分公司,100006,华北大区,2020-07-02
100054,苹果xxx旗舰店,100159,石嘴山市分公司,100007,华西大区,2020-07-02
105307,如山xxx旗舰店,100049,营口市分公司,100006,华北大区,2020-07-02
105308,美丽xxx旗舰店,100236,青岛市分公司,100006,华北大区,2020-07-02
105309,Juxxx旗舰店,100063,白城市分公司,100006,华北大区,2020-07-02
105310,兰思xxx旗舰店,100235,济南市分公司,100006,华北大区,2020-07-02

-- 拉链表(存放用户历史信息)
drop table if exists dim.dim_trade_shops_org_zipper;
create table dim.dim_trade_shops_org_zipper(
shopid int,
shopName string,
cityId int,
cityName string ,
```

```
regionId int ,
regionName string,
start_dt string,
end_dt string
) COMMENT '商家地域组织纬度拉链表' STORED AS PARQUET;
```

2、拉链表加载数据

```
-- 初始化拉链表 (2020-07-01)
insert overwrite table dim.dim_trade_shops_org_zipper
select shopid,
shopName,
cityId,
cityName,
regionId,
regionName,
dt as start_date,
'9999-12-31' as end_date
from dim.dim_trade_shops_org
where dt='2020-07-01';

-- 次日新增数据 (2020-07-02)
100056,小米xxx自营店,100050,阜新市分公司,100006,华北大区,2020-07-02
100055,小米xxx旗舰店,100211,六安市分公司,100006,华北大区,2020-07-02
100054,苹果xxx旗舰店,100159,石嘴山市分公司,100007,华西大区,2020-07-02
105307,如山xxx旗舰店,100049,营口市分公司,100006,华北大区,2020-07-02
105308,美丽xxx旗舰店,100236,青岛市分公司,100006,华北大区,2020-07-02
105309,Juxxx旗舰店,100063,白城市分公司,100006,华北大区,2020-07-02
105310,兰思xxx旗舰店,100235,济南市分公司,100006,华北大区,2020-07-02

-- 构建拉链表
    -- 1) 新增数据处理
select
     shopid,
     shopName,
     cityId,
     cityName,
     regionId,
     regionName,
     dt as start_date,
     '9999-12-31' as end_date
from dim.dim_trade_shops_org
where dt='2020-07-02';

    -- 2) 历史更新数据处理
select
     dim.shopid,
     dim.shopName,
     dim.cityId,
     dim.cityName,
     dim.regionId,
```

```
     dim.regionName,
     dim.start_date,
case when dim.end_dt >= '9999-12-31' and B.shopid is not null
     then '2020-07-01'
     else dim.end_date
end end_date
from dim.dim_trade_shops_org_zipper dim
     left join (select * from dim.dim_trade_shops_org where dt='2020-07-02') B
     on dim.shopid=B.shopid;
```

脚本

/root/dw/script/trade/dim_shops_org_zipper.sh

```
#! /bin/bash

source /etc/profile

if [ -n "$1" ]
then
    do_date=$1
else
        do_date=`date -d "-1 day" +%F`
fi

sql="
insert overwrite table dim.dim_trade_shops_org
select
     shopid,
     shopName,
     cityId,
     cityName,
     regionId,
     regionName,
     dt as start_date,
     '9999-12-31' as end_date
from dim.dim_trade_shops_org
where dt='$do_date'

union all

select
     dim.shopid,
     dim.shopName,
     dim.cityId,
     dim.cityName,
     dim.regionId,
     dim.regionName,
     dim.start_date,
case when dim.end_dt >= '$do_date' and B.shopid is not null
     then date_add('$do_date', -1)
```

```
     else dim.end_date
end end_date
from dim_trade_shops_org_zipper dim
     left join (select * from dim_trade_shops_org where dt='$do_date') B
     on dim.shopid=B.shopid;
"

hive -e "$sql"
```

3、拉链表回滚

```
--  1、处理end_date < rollback_date 的数据，保留
select
     shopid,
     shopName,
     cityId,
     cityName,
     regionId,
     regionName,
     start_date,
     end_date,
     '1' as tag
from dim.dim_trade_shops_org_zipper
where end_date < '2020-07-02';

--  2、处理 start_date <= rollback_date <= end_date 的数据，设置 end_date=9999-12-31
select shopid,
     shopName,
     cityId,
     cityName,
     regionId,
     regionName,
     start_date,
     '9999-12-31' as end_date,
     '2' as tag
from dim.dim_trade_shops_org_zipper
where start_date <= '2020-07-02' and end_date >= '2020-07-02';
```

脚本

/root/dw/script/trade/dim_shops_org_zipper_rollback.sh

```
#! /bin/bash

source /etc/profile

if [ -n "$1" ]
then
    do_date=$1
```

```
else
        do_date=`date -d "-1 day" +%F`
fi

sql="
create table dim.dim_trade_shops_org_zipper_rollback as
select
     shopid,
     shopName,
     cityId,
     cityName,
     regionId,
     regionName,
     start_date,
     end_date,
     '1' as tag
from dim.dim_trade_shops_org_zipper
where end_date < '$do_date';

union all

select shopid,
     shopName,
     cityId,
     cityName,
     regionId,
     regionName,
     start_date,
     '9999-12-31' as end_date,
     '2' as tag
from dim.dim_trade_shops_org_zipper
where start_date <= '$do_date' and end_date >= '$do_date';
"

hive -e "$sql"
```

作业二

在会员分析中计算沉默会员数和流失会员数

解答：

1、沉默会员

定义：只在安装当天启动过App，而且安装时间是在7天前

分析：

- 数据来源：dws.dws_member_add_day（每日新增会员明细）

- 安装且启动过APP，表示新增用户
- 安装时间七天前，表示新增用户需统计date_add('$do_date', -7)
- 利用DWD会员每日启动信息明细数据，可以得到DWS明细

创建DWS表（已有）

```
use dws;

--  创建每日沉默会员明细表
drop table if exists dws.dws_member_silence7_day;
create table dws.dws_member_silence7_day
(
`device_id` string,
`uid` string,
`app_v` string,
`os_type` string,
`language` string,
`channel` string,
`area` string,
`brand` string,
`dt` string
) COMMENT '每日沉默会员明细'
stored as parquet;
```

加载DWS数据（已有）

```
#! /bin/bash

source /etc/profile

if [ -n "$1" ]
then
    do_date=$1
else
        do_date=`date -d "-1 day" +%F`
fi

#-- 加载每日新增会员明细表
sql="
insert into table dws.dws_member_silence7_day
select t1.device_id,
t1.uid,
t1.app_v,
t1.os_type,
t1.language,
t1.channel,
t1.area,
t1.brand,
t1.dt
```

```
from(
(select * from dws.dws_member_add_day where dt=date_add('$do_date', -7)) t1
left join
(select * from dws.dws_member_add_day where dt > date_add('$do_date', -6)) t2
on t1.device_id=t2.device_id)
where t2.device_id is null;
"
hive -e "$sql"
```

创建ADS层表

```
drop table if exists ads.ads_new_member_silence7_cnt;
create table ads.ads_new_member_silence7_cnt
(
`cnt` string
)
partitioned by(dt string)
row format delimited fields terminated by ',';
```

加载ADS层数据

```
#! /bin/bash

source /etc/profile

if [ -n "$1" ]
then
    do_date=$1
else
                do_date=`date -d "-1 day" +%F`
fi

#--     加载每日新增会员明细表
sql="
insert into table dws.ads_new_member_silence7_cnt
partition (dt='$do_date')
select count(*) from dws.dws_member_silence7_day
where dt='$do_date';
"
hive -e "$sql"
```

2、流失会员

定义：最近30天未登录的会员

分析：

- 数据来源：dws.dws_member_start_day（会员日启动汇总表）
- 30天未登陆，即统计30天前一天date_add('$do_date', -30)
- 利用DWD会员每日启动信息明细数据，可以得到DWS明细

创建DWS表（已有）

```
use dws;

--      创建每日新增会员明细表
drop table if exists dws.dws_member_lose_day;
create table dws.dws_member_lose_day
(
`device_id` string,
`uid` string,
`app_v` string,
`os_type` string,
`language` string,
`channel` string,
`area` string,
`brand` string,
`dt` string
) COMMENT '每日流失会员明细'
stored as parquet;
```

加载DWS数据（已有）

```
#! /bin/bash

source /etc/profile

if [ -n "$1" ]
then
    do_date=$1
else
                do_date=`date -d "-1 day" +%F`
fi

#--     加载每日新增会员明细表
sql="
insert into table dws.dws_member_lose_day
select t1.device_id,
t1.uid,
t1.app_v,
t1.os_type,
t1.language,
t1.channel,
t1.area,
t1.brand,
t1.dt
from(
(select * from dws.dws_member_start_day where dt=date_add('$do_date', -31)) t1
left join
(select * from dws.dws_member_start_day where dt > date_add('$do_date', -30)) t2
```

```
on t1.device_id=t2.device_id)
where t2.device_id is null;
"
hive -e "$sql"
```

创建ADS层表

```
drop table if exists ads.ads_new_member_lose_cnt;
create table ads.ads_new_member_lose_cnt
(
`cnt` string
)
partitioned by(dt string)
row format delimited fields terminated by ',';
```

加载ADS层数据

```
#! /bin/bash

source /etc/profile

if [ -n "$1" ]
then
    do_date=$1
else
                do_date=`date -d "-1 day" +%F`
fi

#--     加载每日新增会员明细表
sql="
insert into table dws.ads_new_member_lose_cnt
partition (dt='$do_date')
select count(*) from dws.dws_member_lose_day
where dt='$do_date';
"
hive -e "$sql"
```

作业三

在核心交易分析中完成如下指标的计算

需求：

- 统计2020年每个季度的销售订单笔数、订单总额
- 统计2020年每个月的销售订单笔数、订单总额
- 统计2020年每周（周一到周日）的销售订单笔数、订单总额

- 统计2020年国家法定节假日、休息日、工作日的订单笔数、订单总额

分析：

- 现已建成DWS层订单明细表（dws_trade_orders）和订单明细宽表（dws_trade_orders_w）
- 获取当前第几度开始日期：select trunc('2017-05-04','Q')
- 获取当前周一至周日日期：
- 法定节假日需单独建表

解答：

1、建表

```
-- ADS
DROP TABLE IF EXISTS ads.ads_trade_order_analysis_week;
create table if not exists ads.ads_trade_order_analysis_week(
totalcount_week bigint,     --  每周订单笔数
totalmoney_week double,     --  每周订单总额
week int                        --  本年度第几周 weekofyear('2017-12-04')
)partitioned by (dt string)
row format delimited fields terminated by ',';

create table if not exists ads.ads_trade_order_analysis_month(
totalcount_month bigint,    --  每月订单笔数
totalmoney_month double    --  每月订单总额
)partitioned by (dt string)
row format delimited fields terminated by ',';

create table if not exists ads.ads_trade_order_analysis_quarter(
totalcount_quarter bigint,  --  季度订单笔数
totalmoney_quarter double  --  季度订单总额
)partitioned by (dt string)
row format delimited fields terminated by ',';

create table if not exists ads.ads_trade_order_analysis_vacate(
totalcount_vacate bigint,   --  假期订单笔数
totalmoney_vacate double   --  假期订单总额
)partitioned by (dt string)
row format delimited fields terminated by ',';

--      假期表
create table if not exists dim.dim_date(
dt      date,                           --      日期
yearmonth       int,            --      年月
year    smallint,       --      年
month   tinyint,                --      月
day     tinyint,                        --      日
week    tinyint,                --      周几
weeks   tinyint,                --      第几周
quat    tinyint,                --      季度
```

```
vacate  smallint        --      节假日标示，是--1，否--0
) row format delimited fields terminated by ',';
```

2、实现

```
--      周
select count(distinct orderid) as totalcount_week,
       sum(paymoney) as totalmoney_week,
       weekofyear('2017-12-04') week
from dws.dws_trade_orders_w where dt>= date_add(next_day('2020-10-15','mo'),-1) and
dt<='2020-10-15';

--      月
select count(distinct orderid) as totalcount_month,
       sum(paymoney) as totalmoney_month
from dws.dws_trade_orders_w where month(dt) = month('2020-10-15');

--      季度
select count(distinct orderid) as totalcount_quarter,
       sum(paymoney) as totalmoney_quarter
from dws.dws_trade_orders_w where dt > trunc('2020-10-15','Q') and dt <= '2020-10-15';

--      假期
with mid_orders as (
select od.orderid as orderid,od.paymoney as paymoney from (
(select * from dws.dws_trade_orders_w where dt='2020-10-15') od
  left join dim.dim_date d
  on od.dt=d.dt and d.vacate=1)
  where d.dt is not null
)
select count(distinct orderid) as totalcount_vacate,
       sum(paymoney) as totalmoney_vacate
from mid_orders
```

```
#! /bin/bash

source /etc/profile

if [ -n "$1" ]
then
    do_date=$1
else
                do_date=`date -d "-1 day" +%F`
fi

#--     加载每日新增会员明细表
sql="
insert into table ads.ads_trade_order_analysis_week
partition (dt='$do_date')
select count(distinct orderid) as totalcount_week,
```

```
       sum(paymoney) as totalmoney_week,
       weekofyear('$do_date') week
from dws.dws_trade_orders_w where dt>= date_add(next_day('$do_date','mo'),-1) and
dt<='$do_date';

insert into table ads.ads_trade_order_analysis_month
partition (dt='$do_date')
select count(distinct orderid) as totalcount_month,
       sum(paymoney) as totalmoney_month
from dws.dws_trade_orders_w where month(dt) = month('$do_date');


insert into table ads.ads_trade_order_analysis_quarter
partition (dt='$do_date')
select count(distinct orderid) as totalcount_quarter,
       sum(paymoney) as totalmoney_quarter
from dws.dws_trade_orders_w where dt > trunc('$do_date','Q') and dt <= '$do_date';


with mid_orders as (
select od.orderid as orderid,od.paymoney as paymoney from (
(select * from dws.dws_trade_orders_w where dt='$do_date') od
  left join dim.dim_date d
  on od.dt=d.dt and d.vacate=1)
  where d.dt is not null
)
insert into table ads.ads_trade_order_analysis_vacate
partition (dt='$do_date')
select count(distinct orderid) as totalcount_vacate,
       sum(paymoney) as totalmoney_vacate
from mid_orders;

"
hive -e "$sql"
```