方案一：

问题：拉链表重复跑某一天数据错误

原始 hql：

```
insert overwrite table dwd_dim_user_info_his_tmp
select * from
(
    select
        id,
        name,
        birthday,
        gender,
        email,
        user_level,
        create_time,
        operate_time,
        '2020-06-15' start_date,
        '9999-99-99' end_date
    from ods_user_info where dt='2020-06-15'
    union all
    select
        uh.id,
        uh.name,
        uh.birthday,
        uh.gender,
        uh.email,
        uh.user_level,
        uh.create_time,
        uh.operate_time,
        uh.start_date,
        if(ui.id is not null  and uh.end_date='9999-99-99', date_add(ui.dt,-1), uh.end_date) end_date
    from dwd_dim_user_info_his uh left join
    (
        select
            *
        from ods_user_info
        where dt='2020-06-15'
    ) ui on uh.id=ui.id
)his
order by his.id, start_date;
```

数据错误原因：

多次跑同一天数据，dwd_dim_user_info_his 表中当天数据会被当做历史数据无差别更新

解决思路：

每次跑目标日数据时，dwd_dim_user_info_his 表中数据只取非当前日期以前数据

```
insert overwrite table dwd_dim_user_info_his_tmp
select * from
(
    select
        id,
        name,
        birthday,
        gender,
        email,
        user_level,
        create_time,
        operate_time,
        '2020-06-15' start_date,
        '9999-99-99' end_date
    from ods_user_info where dt='2020-06-15'
    union all
    select
        uh.id,
        uh.name,
        uh.birthday,
        uh.gender,
        uh.email,
        uh.user_level,
        uh.create_time,
        uh.operate_time,
        uh.start_date,
        if(ui.id is not null  and uh.end_date='9999-99-99', date_add(ui.dt,-1), uh.end_date) end_date
    from dwd_dim_user_info_his uh left join
    (
        select
            *
        from ods_user_info
        where dt='2020-06-15'
    ) ui on uh.id=ui.id  where uh.start_date < '2020-06-15'
)his
order by his.id, start_date;
```

目前看暂时是解决了，引发思考 如果跑的是历史中的某一天重跑 就不能用 “<”目标日期作为 where 条件，这样该天后的数据就会丢失而这里如果使用“!=” ，目标日数据的 end_date 无法确定。 所以该思路如果用于重跑历史数据，需要从目标日一直跑到最新一天数据。

方案二：

张走走、：

```
insert overwrite table dwd_dim_user_info_his_tmp
select * from
(
    select
        id,
        name,
        birthday,
        gender,
        email,
        user_level,
        create_time,
        operate_time,
        '2020-06-15' start_date,
        '9999-99-99' end_date
    from ods_user_info where dt='2020-06-15'

    union
    select
        uh.id,
        uh.name,
        uh.birthday,
        uh.gender,
        uh.email,
        uh.user_level,
        uh.create_time,
        uh.operate_time,
        uh.start_date,
        if(ui.id is not null and uh.end_date='9999-99-99' and uh.start_date !='2020-06-15',
date_add(ui.dt,-1), uh.end_date) end_date
    from dwd_dim_user_info_his uh left join
    (
        select
            *
        from ods_user_info
        where dt='2020-06-15'
    ) ui on uh.id=ui.id
)his
```

order by his.id, start_date;

多加了个判断

然后 union 去重 不用 union all

方案三

问题：拉链表重复跑某一天数据错误

原始 hql：

```
insert overwrite table dwd_dim_user_info_his_tmp
select * from
(
    select
        id,
        name,
        birthday,
        gender,
        email,
        user_level,
        create_time,
        operate_time,
        '2020-06-15' start_date,
        '9999-99-99' end_date
    from ods_user_info where dt='2020-06-15'
    union all
    select
        uh.id,
        uh.name,
        uh.birthday,
        uh.gender,
        uh.email,
        uh.user_level,
        uh.create_time,
        uh.operate_time,
        uh.start_date,
        if(ui.id is not null  and uh.end_date='9999-99-99', date_add(ui.dt,-1), uh.end_date) end_date
    from dwd_dim_user_info_his uh left join
    (
        select
            *
        from ods_user_info
        where dt='2020-06-15'
    ) ui on uh.id=ui.id
)his
order by his.id, start_date;
```

数据错误原因：

多次跑同一天数据，dwd_dim_user_info_his 表中当天数据会被当做历史数据无差别更新

解决思路：

因为 mysql 中,同时每个 user 只会记录一条记录,所以同步到 hive 时，相同 user 的 start_time 是唯一的。所以同步到临时表时，有错误的数据无所谓。只需要利用开窗，只取一条的方法，去重处理即可。

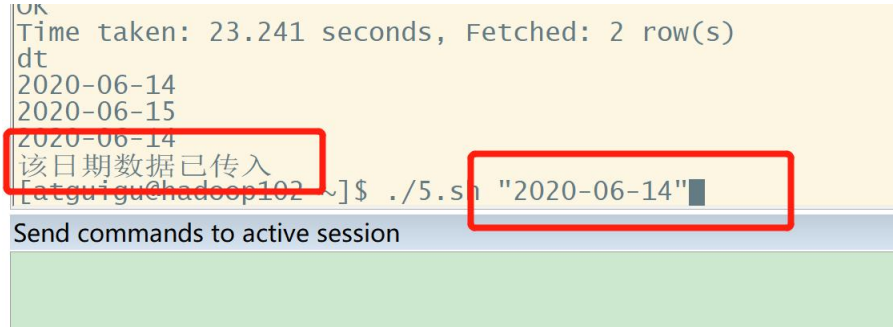
```
        if(ui.id is not null  and uh.end_date='9999-99-99' , date_add(ui.dt,-1), uh.end_date) end_date
    from dwd_dim_user_info_his uh left join
    (
        select
            *
        from ods_user_info
        where dt='2020-06-15'
    ) ui on uh.id=ui.id
)his
order by his.id, start_date;

insert overwrite table dwd_dim_user_info_his
select id,name,birthday,gender,email,user_level,create_time,operate_time,start_date,end_date from
(select *,row_number() over (partition by id,start_date order by end_date desc ) rn from dwd_dim_user_info_his_tmp)t1 where t1.rn = 1;
```

目前看暂时是解决了，引发思考 如果跑的是历史中的某一天重跑　就不能用 “<”目标日期作为 where 条件，这样该天后的数据就会丢失而这里如果使用“!=” ，目标日数据的 end_date 无法确定。 所以该思路如果用于重跑历史数据，需要从目标日一直跑到最新一天数据。

方案四：
达达:
海哥，这算一个吗，先取出每一个 start_date，再和要传入的时间进行判断，如果有，则退出程序，没有，程序继续运行

```
#!/bin/bash

APP=gmall
hive=/opt/module/hive/bin/hive

#查找这一天是否已经倒入过数据，这一天导入了数据的话，start_date就是这一天
already_date=`hive -e "select distinct dt from ${APP}.ods_user_info";`


#echo "$already_date"

# 如果是输入的日期按照取输入日期；如果没输入日期取当前时间的前一天
if [ -n "$1" ] ;then
    do_date=$1
else
    do_date=`date -d "-1 day" +%F`
fi

#echo "$do_date"



result=$(echo $already_date | grep "${do_date}")
if [[ "$result" != "" ]]
then
    echo "该日期数据已传入"
    exit
fi

echo "zhengchang"
```

```
Time taken: 24.462 seconds, Fetched
dt
2020-06-14
2020-06-15
2020-08-06
正常执行
[atguigu@hadoop102 ~]$ ./5.sh
```


OK
Time taken: 23.241 seconds, Fetched: 2 row(s)
dt
2020-06-14
2020-06-15
2020-06-14
该日期数据已传入
[atguigu@hadoop102 ~]$ ./5.sh "2020-06-14"
Send commands to active session