## 1.设计

1. 每半年迁入一个表中，这样易于业务理解。

2. 从activity表中读取出来2017年的活动id，然后根据活动id去读取activitySku表，将activitySku表中的数据迁移到history表中

3. 循环遍历2017年活动下的pageSize个activitySku数据，若没有写缓存并continue，若有数据insert到history表并delete原表数据，sleep20秒后继续下次for循环，为了使数据库机器压力小点



## 2. 问题

### 2.1 索引

[**mysql索引总结----mysql 索引类型以及创建**](http://www.cnblogs.com/lihuiyong/p/5623191.html)

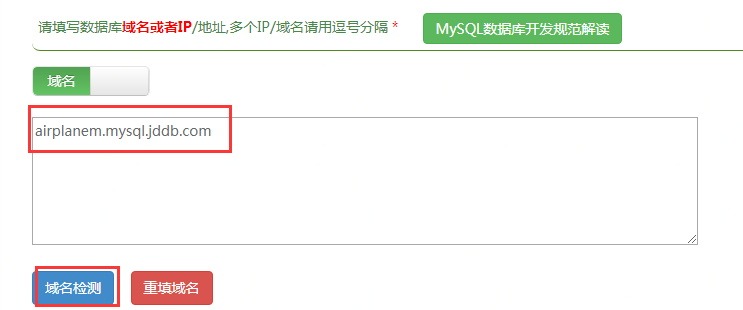
迁移数据时，由于没有在air\_prom\_bdata\_activitysku表中建立索引，导致5条数据的处理速度是300s。在建立索引后处理5条数据的处理速度是300ms，速度提升1000倍。以actId为索引。

#### dbs.jd.com





输入域名后，进行域名检测：



申请建索引：ALTER TABLE air\_prom\_bdata\_activitysku ADD INDEX idx\_act\_id (act\_id)

### 2.2 反射复制属性

由于要复制对象ActivitySku到ActivitySkuHistory，对象属性是private，因此使用反射（性能较差，反射一次需要1S多）。但是不能每次for循环都执行一次反射，这样耗时很长。

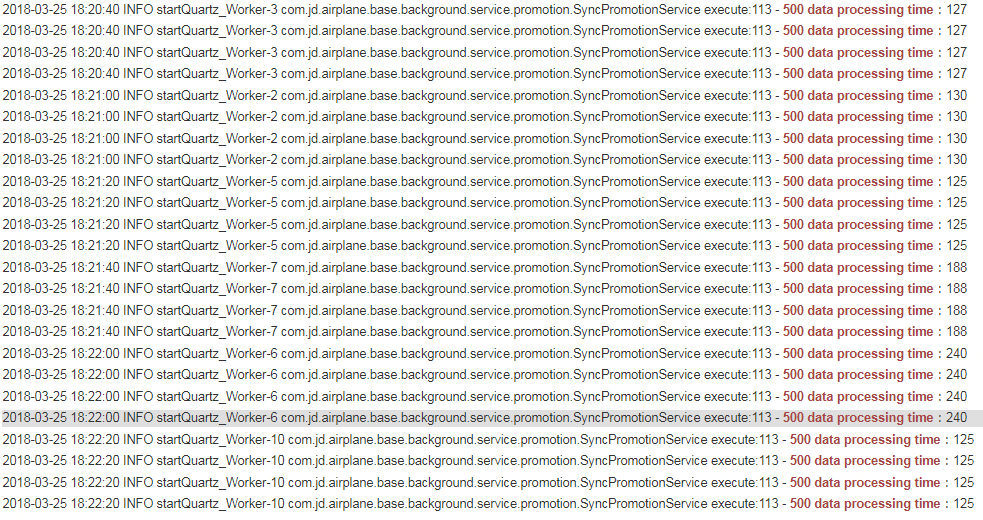
将反射出来的Field存在全局对象Map中，这样程序启动以后只需要反射一次，后面从map中读取field。

### 2.3 多线程

开10个线程，多线程去迁移数据。将insert和delete逻辑放在*executor*.submit(new ActSKUCallable(history, activitySKU, tablePostFix, countDownLatch)); 的call()方法中。

Callable有返回值。结合使用countDownLatch统计执行时间。

结果：500条数据的首次执行时间1s（有反射），后面的执行时间200ms左右。



### 2.4 model中字段枚举

Model的枚举字段与数据表的int字段的映射转换

Mybatis中的typeHandler

EnumTypeHandler

## 3. 表名设计

表名后缀以活动的开始时间计算，开始时间在6月前的数据表名后缀是“YYYY06”

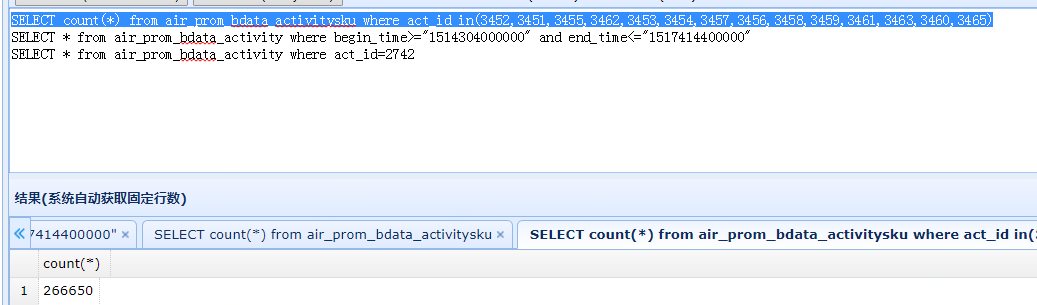
开始时间在6月后的数据表名后缀是“YYYY12”



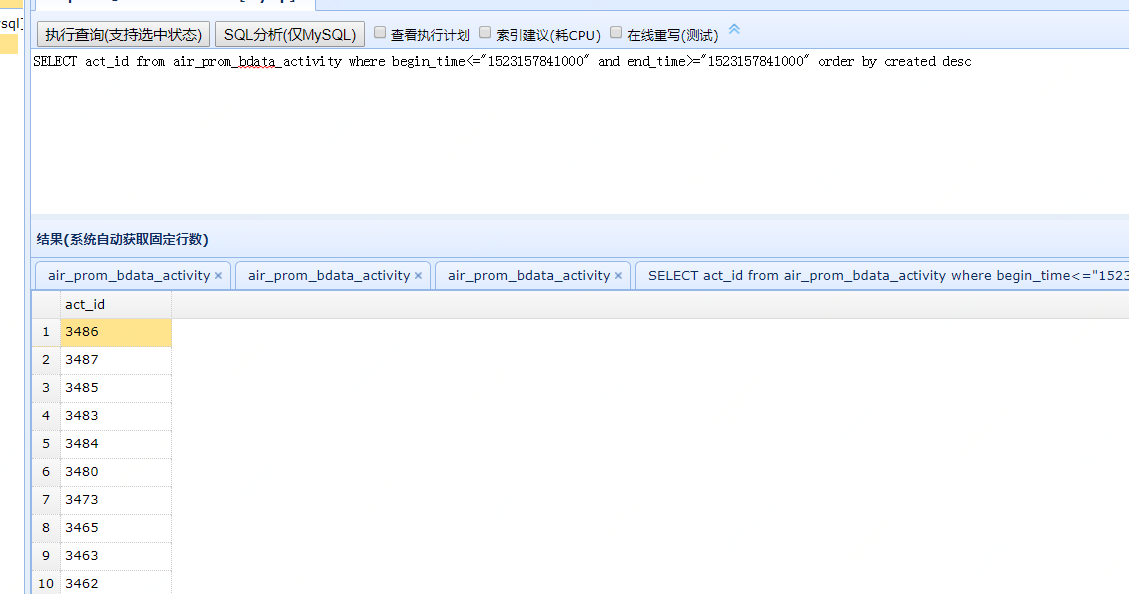
## Sql语句：

SELECT count(\*) from air\_prom\_bdata\_activitysku where act\_id in(3452,3451,3455,3462,3453,3454,3457,3456,3458,3459,3461,3463,3460,3465)

SELECT act\_id from air\_prom\_bdata\_activity where begin\_time<="1522631893000" and end\_time>="1522631893000"



4月8日



4月16日

SELECT act\_id from air\_prom\_bdata\_activity where begin\_time<="1523841299000" and end\_time>="1523841299000" order by created

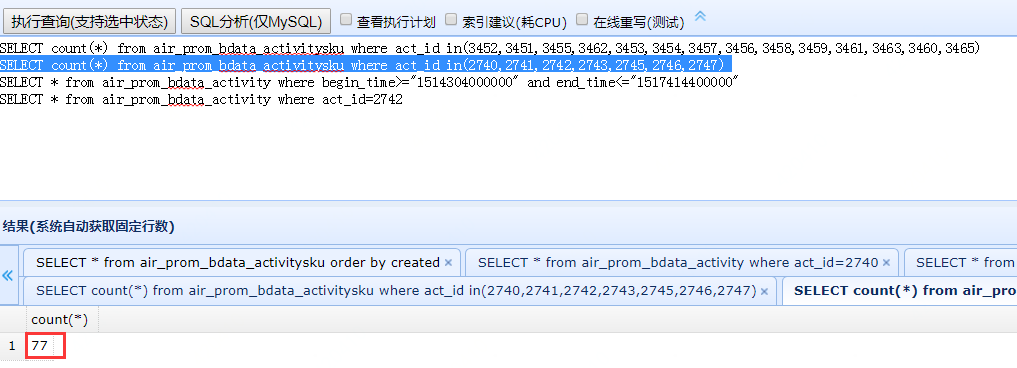
## 迁移进度：

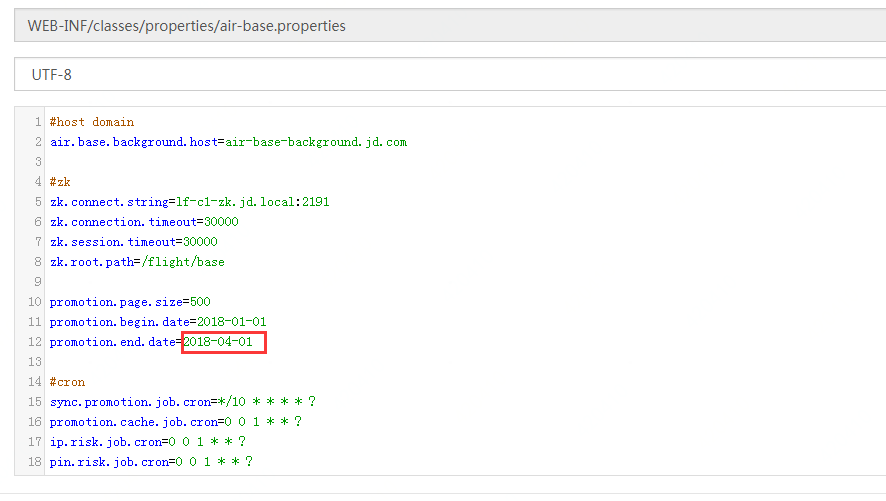
### 2018年4月1日之前全部迁完

备注：表中有开始时间是2017年的activitySku数据

2740,2741,2742,2743,2745,2746,2747活动对应的数据

有迁错的数据：将2018年数据迁到2017年表中





### 2018年6月19日前

promotion.page.size=200

promotion.begin.date=2018-03-31

promotion.end.date=2018-06-19

### 2018年6月25日前

## 改成定时任务

## 0 0 9 ? \* MON：每周一上午九点，每执行pageSize条sleep