# MySQL单表2000万数据查询慢，时间触发器+分区解决

## 事起

当时我们一个项目，要不停地从第三方系统拿一些监控数据存起来，只保留2个月。数据量2千万+。这个数据量，查起来就比较慢了，我们就开始优化。

思路就是将表分区，我们是按照8周，将单表分了8个区，每周一都会将最早一周的分区Drop掉，然后新建下一周的分区。

定时任务这一块，我们当时也没想着用Quartz，直接用MySQL的时间触发器。

## 准备

为写本文档，我是在本机用5.7版本，中间会报一些错误，解决如下：

Error 1418

|  |
| --- |
| set global log\_bin\_trust\_function\_creators=TRUE; |

或配置文件中增加

|  |
| --- |
| log\_bin\_trust\_routine\_creators=1; |



|  |
| --- |
| select @@sql\_mode |

结果如下：

|  |
| --- |
| **ONLY\_FULL\_GROUP\_BY**,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION |

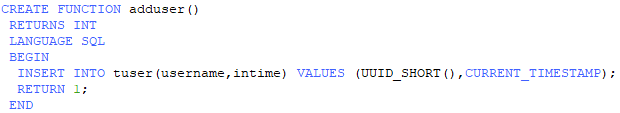
在my.cnf中添加如下，把红色的删掉。

|  |
| --- |
| sql\_mode= STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_AUTO\_CREATE\_USER,NO\_ENGINE\_SUBSTITUTION |

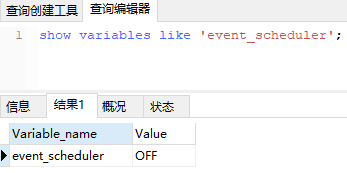
## 时间触发器

先建一张表tuser，id自增，username，intime，建表语句略。

建一个Function，专门insert数据。



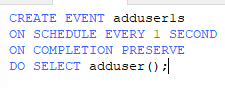
注意，我的5.7默认是关闭event的，如下



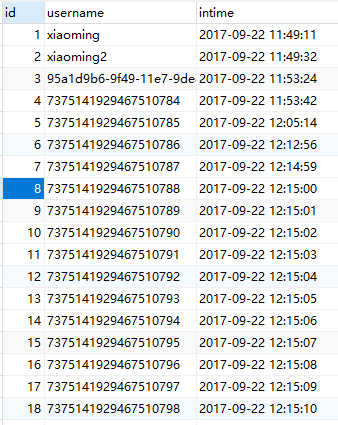
要开启：

|  |
| --- |
| set global event\_scheduler=1; |

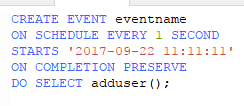
创建一个event，每秒执行一下adduser函数



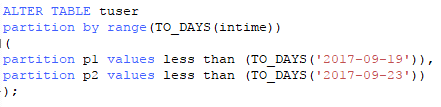
结果如下：



也可以指定开始时间：



## 分区



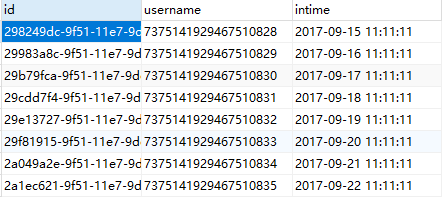
在本例中，我就让表4天一分区了，先建两个分区。



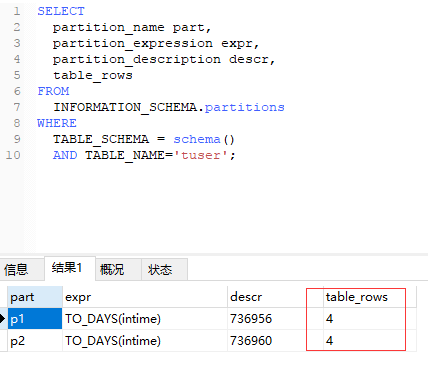
报这个错误是因为我们是按照入表时间这个字段来分的，但是分区要求这个字段必须是主键或主键的一个。

所以我们把原来单个id主键索引删掉，改成id、intime为主键即可。我重新建了表，id是uuid，与intime一起为主键。

向表里插入数据如下：

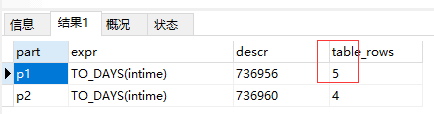


按照分区的原则，本表一共两个分区，每个分区有4条数据，查询结果如下：



正好对应。

插入一条14号的数据：

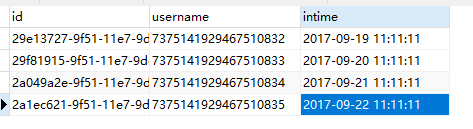


也没问题。

接着我们删除第一个分区：

|  |
| --- |
| alter table tuser drop partition p1; |

分区内的数据一并删除：

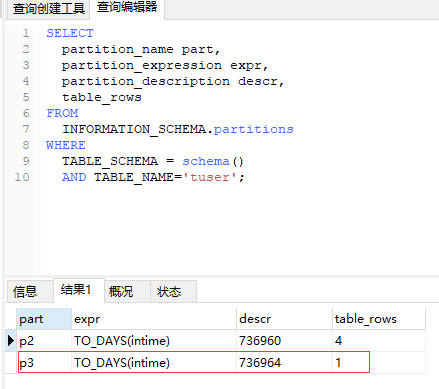


再创建一个分区

|  |
| --- |
| alter table tuser add partition (partition **p3** values less than (TO\_DAYS('**2017-09-27**'))); |

注意红色粗体，我们新建的分区必须要在原分区之后追加，虽然我们删除了p1分区，但是这里也不能建p1，必须要大于p2，日期也要大于p2的。

我们增加一条23号的数据



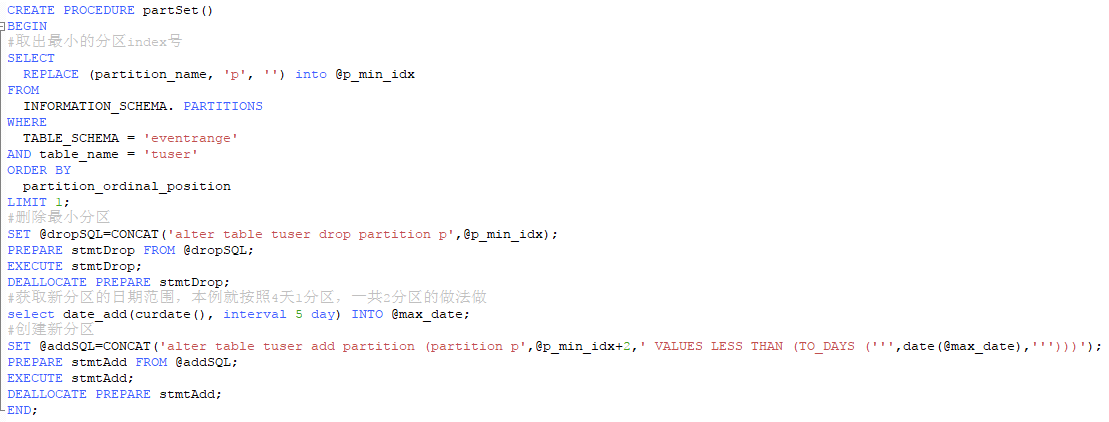
没有问题。

至此，解决方案的各项技术点都已经测试通过了，可以整合了。

## 整合方案

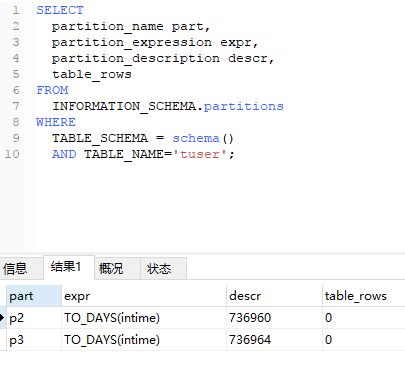
思路很简单了，每7天某个时间点触发Event，执行一个存储过程，先取出最早的分区，drop，然后创建一个最大分区号+1，日期也+7的分区。

本文为了演示，就按照4天1分区，一共2个分区的要求来写了。



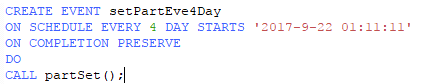
大家在实际工作中，主要是注意分区的序号和日期，要+上合适的数字。

执行结果：



记录为0是我之前把数据都清了一下。

接着是创建Event



至此完结。