

[数据库] Navicat for MySQL事件Event实现数据每日定期操作

原创 Eastmount 2017-03-10 12:26:10 18779 收藏 5

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法、



Eastmount

¥9.90

订阅

在我们操作数据库过程中，通常会遇到一些某个时间点操作数据库的问题，例如：

- (1).每天凌晨12点对数据库进行定时备份，结算和汇总；
- (2).每天凌晨2点删除数据库前三天的数据；
- (3).插入某个数据超过一定时间改变某个值的状态，比如预警系统。

这里就需要通过Event事件进行简单操作，下面将详细处理。你可能会想到通过触发器实现，但是如果是同一张表Insert插入数据后，但是触发器再进行Update更新操作是不行的，所以需要尝试通过Event事件解决。

在前面讲过一篇文章：[\[数据库\] Navicat for MySQL触发器更新和插入操作](#)

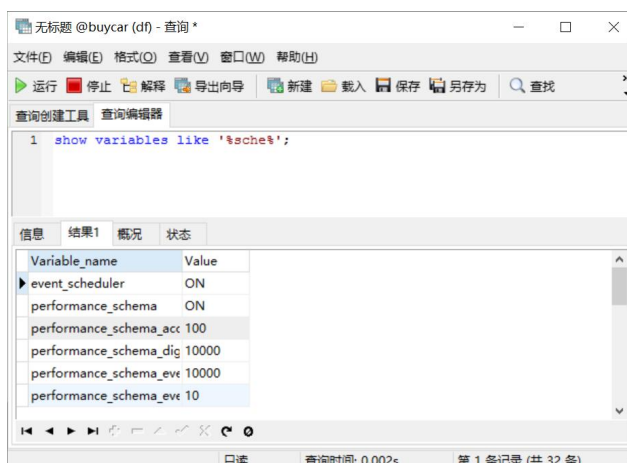
希望这篇文章对你有所帮助，如果文章中出现错误或不足之处，还请海涵~

一. Event实现每日某个时刻删除事件

首先通过Navicat for MySQL查看Event事件是否开启，具体代码：

```
show variables like '%sche%';
```

通常事件是开启的，如下图所示：



如果没有开启，需要数据库超级权限设置。

```
set global event_scheduler =1;
```

其中event_scheduler为ON表示开启，OFF表示关闭，如下：

```
+-----+-----+
|Variable_name|Value|
+-----+-----+
|event_scheduler|OFF|
+-----+-----+
```

下面开始写事件进行操作。

假设现在存在一张表loginuser，包括用户名、密码、权限三个字段。如下图：

loginuser @buycar (df) - 表

文件 编辑 查看 窗口 帮助

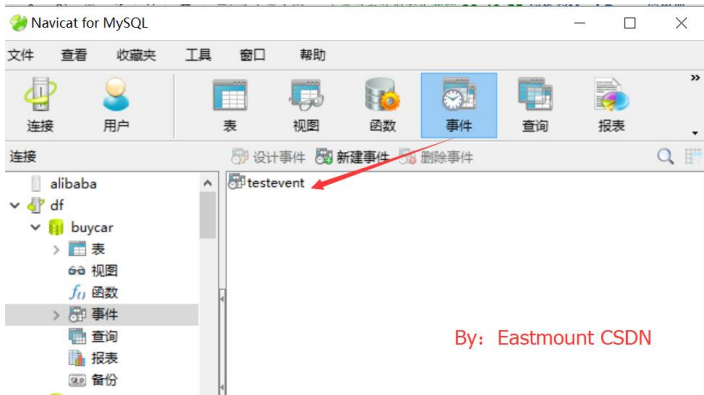
导入向导 导出向导 筛选向导

Username	Password	Role
xiuzhang	123456	1
xiaomin	123456	2
eastmount	000000	2

当前时间点为2017年3月9日 23:48，我们设置23:50执行删除eastmount数据操作。其中SQL语句的Event代码如下：

```
CREATE EVENT testevent
ON SCHEDULE EVERY 1 DAY STARTS '2017-03-09 23:50:00'
DO
    delete from loginuser where Username='eastmount';
```

创建事件可以查看到的，如下图所示：

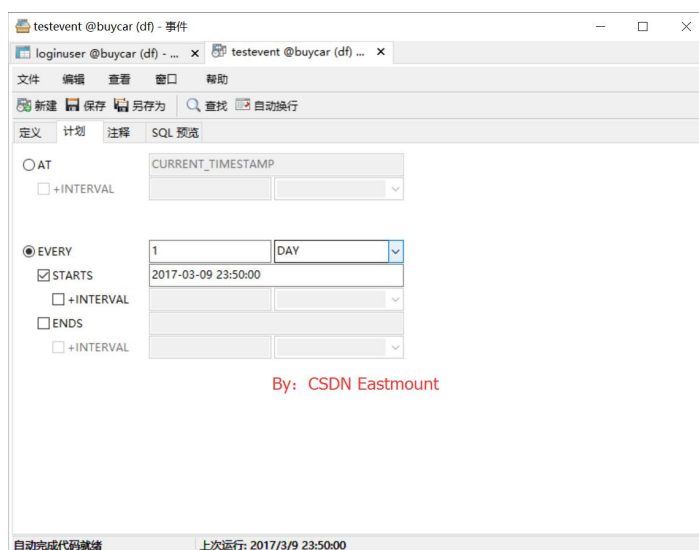
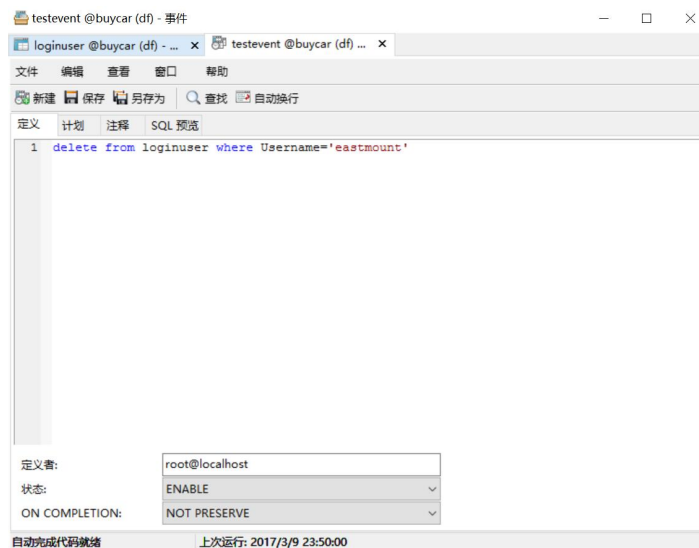


然后过了23:50执行查询语句就会发现eastmount用户已经被删除。

```
1 select * from loginuser;
```

信息	结果1	概况	状态
Username	Password	Role	
xiaomin	123456	2	
xiuzhang	123456	1	

点开事件如下图所示，但我更推荐大家写SQL语句。



通过上面这段SQL语句，我们知道了最简单的Event事件，下面进一步加深介绍。备份数据加个备份SQL语句执行即可。

参考官网：<https://dev.mysql.com/doc/refman/5.7/en/event-scheduler.html>

二. Event实现每隔某段时间定时插入事件

在使用事件Event中，需要开启和关闭事件，使用的代码如下：

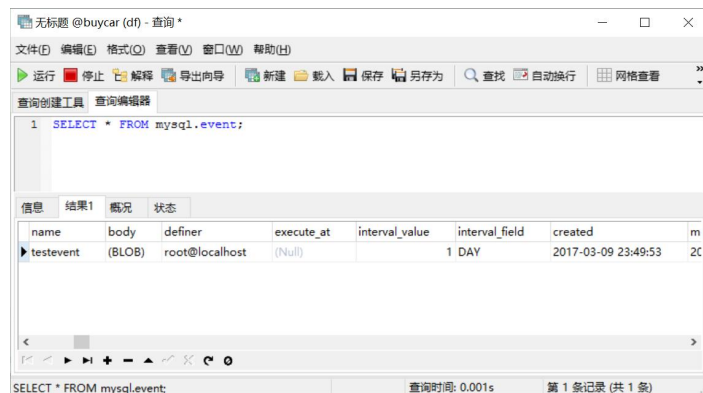
关闭事件：ALTER EVENT testevent DISABLE;

开启事件：ALTER EVENT testevent ENABLE;

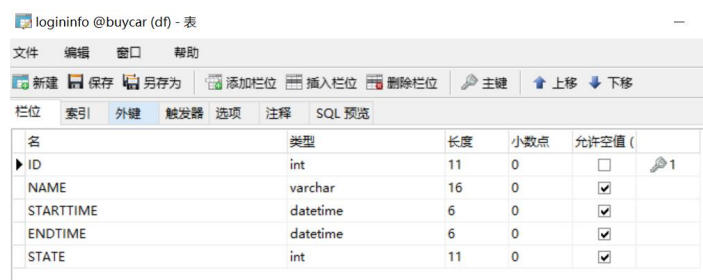
查看事件是否开启的代码如下：

```
SELECT * FROM mysql.event;
```

输出如下图所示：



下面定义每隔10秒插入一行数据的事件，比如实时接收交通数据信息等实时处理，通常会使用该方法进行插入操作。创建一个表logininfo，记录登录信息，字段包括：ID序号（主键、递增INT型）、NAME用户名、STARTTIME（登录时间）、ENDTIME（登出时间）、STATE（状态 0-离线 1-在线）。



SQL语句创建EVENT事件如下：

```
CREATE EVENT insertevent
ON SCHEDULE EVERY 10 SECOND
DO
    INSERT INTO logininfo(NAME,STARTTIME,STATE) values('test01',now(),'1');
```

运行结果如下图所示：

ID	NAME	STARTTIME	ENDTIME	STATE
2	test01	2017-03-10 09:07:07.000000	(Null)	1
3	test01	2017-03-10 09:07:17.000000	(Null)	1
4	test01	2017-03-10 09:07:27.000000	(Null)	1
5	test01	2017-03-10 09:07:37.000000	(Null)	1
6	test01	2017-03-10 09:07:47.000000	(Null)	1
7	test01	2017-03-10 09:07:57.000000	(Null)	1
8	test01	2017-03-10 09:08:07.000000	(Null)	1
9	test01	2017-03-10 09:08:17.000000	(Null)	1
10	test01	2017-03-10 09:08:27.000000	(Null)	1
11	test01	2017-03-10 09:08:37.000000	(Null)	1

从图中可以看到，每隔10秒就插入一组数据，该事件正确执行。

注意：在真实的开发过程中，会遇到mysql服务重启或断电等情况，此时则会出现时间调度器被关闭的情况，所有事件都不起作用，解决方法，需要在mysql.ini文件中加入 "event_scheduler = ON;" 语句。

三. Event实现超过某个时间更新状态

假设现在有一个报警系统，当用户进入房间开始计时，当用户在放在中待了2小时就开始报警，设置状态。数据如下：

logininfo @buycar (df) - 表

ID	NAME	STARTTIME	ENDTIME	STATE
41	test01	2017-03-10 12:04:44.000000	(Null)	1
42	test01	2017-03-10 12:04:54.000000	(Null)	1

SQL语句查询返回结果如下：

```
select current_timestamp, starttime, timestampdiff(SECOND,starttime,current_timestamp) as T from logininfo;
```

输出结果如下图所示，每隔10秒会插入一个数据，所以不断更新。

current_timestamp	starttime	T
2017-03-10 12:06:14	2017-03-10 12:04:44.0000	90
2017-03-10 12:06:14	2017-03-10 12:04:54.0000	80
2017-03-10 12:06:14	2017-03-10 12:05:04.0000	70
2017-03-10 12:06:14	2017-03-10 12:05:14.0000	60
2017-03-10 12:06:14	2017-03-10 12:05:24.0000	50
2017-03-10 12:06:14	2017-03-10 12:05:34.0000	40
2017-03-10 12:06:14	2017-03-10 12:05:44.0000	30
2017-03-10 12:06:14	2017-03-10 12:05:54.0000	20

现在的更新UPDATE的语句如下所示：

```
CREATE EVENT updateevent
ON SCHEDULE EVERY 10 SECOND
DO
    update logininfo set STATE=0 WHERE
        timestampdiff(SECOND,starttime,current_timestamp)>300 and STATE=1;
```

然后查询实时更新状态的SQL语句如下：

```
select ID, STATE, current_timestamp, starttime,
        timestampdiff(SECOND,starttime,current_timestamp) as T
from logininfo;
```

输出如下图所示，发现302可能是查过300秒，可能实时更新的延误，但不影响效果的。

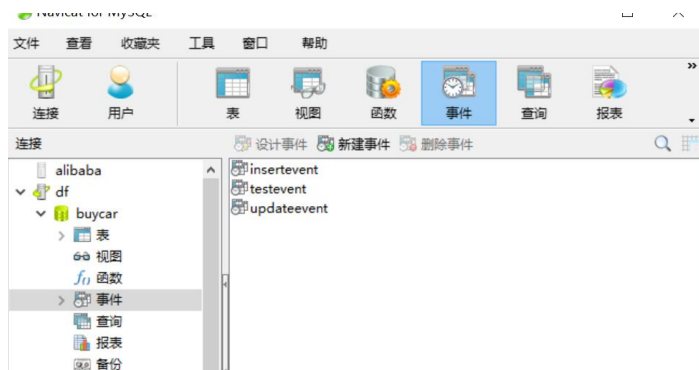
```

1 select ID, STATE, current_timestamp, starttime,
2       timestampdiff(SECOND,starttime,current_timestamp) as T
3 from logininfo;
4

```

ID	STATE	current_timestamp	starttime	T
41	0	2017-03-10 12:10:16	2017-03-10 12:04:44.0000	332
42	0	2017-03-10 12:10:16	2017-03-10 12:04:54.0000	322
43	0	2017-03-10 12:10:16	2017-03-10 12:05:04.0000	312
44	1	2017-03-10 12:10:16	2017-03-10 12:05:14.0000	302
45	1	2017-03-10 12:10:16	2017-03-10 12:05:24.0000	292
46	1	2017-03-10 12:10:16	2017-03-10 12:05:34.0000	282
47	1	2017-03-10 12:10:16	2017-03-10 12:05:44.0000	272
48	1	2017-03-10 12:10:16	2017-03-10 12:05:54.0000	262

讲到这里，一个简单预警系统的数据库就设置完成，总共我们设置了3个事件，如下图所示：



补充一段代码：

```

delimiter //
CREATE EVENT e
ON SCHEDULE
    EVERY 5 SECOND
DO
BEGIN
    DECLARE v INTEGER;
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION BEGIN END;
    SET v = 0;
    WHILE v < 5 DO
        INSERT INTO t1 VALUES (0);
        UPDATE t2 SET s1 = s1 + 1;
        SET v = v + 1;
    END WHILE;
END //
delimiter ;

```

最后希望文字对您有所帮助，基础性文章，如果存在错误或不足之处，还请海涵~
最近自己和她太操劳，自己需要多付出点，多学点，多为她做点，forever。

(By:Eastmount 2017-03-10 中午12点 <http://blog.csdn.net/eastmount/>)



Eastmount



博客专家

原创文章 462

获赞 6724

访问量 525万+

关注

他的留言板