

[数据库] Navicat for MySQL触发器更新和插入操作

原创 Eastmount 2016-09-18 17:13:19 38142 收藏 7

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

一、触发器概念

触发器 (trigger)：监视某种情况，并触发某种操作，它是提供给程序员和数据分析师来保证数据完整性的一种方法，它是与表事件相关的特殊的存储过程，它的执行不是由程序调用，也不是手工启动，而是由事件来触发，例如当对一个表进行操作 (insert, delete, update) 时就会激活它执行。

触发器经常用于加强数据的完整性约束和业务规则等。触发器创建语法四要素：

1. 监视地点(table)
2. 监视事件(insert/update/delete)
3. 触发时间(after/before)
4. 触发事件(insert/update/delete)

触发器基本语法如下所示：

```
CREATE TRIGGER trigger_name trigger_time trigger_event
ON tbl_name FOR EACH ROW trigger_stmt
```

注意：触发器只能创建在永久表 (Permanent Table) 上，不能对临时表 (Temporary Table) 创建触发器。

其中：trigger_time是触发器的触发事件，可以为before (在检查约束前触发) 或after (在检查约束后触发)；trigger_event是触发器的触发事件，包括insert、update和delete，需注意对同一个表相同触发时间的相同触发事件，只能定义一个触发器；可以使用old和new来引用触发器中发生变化的记录内容。

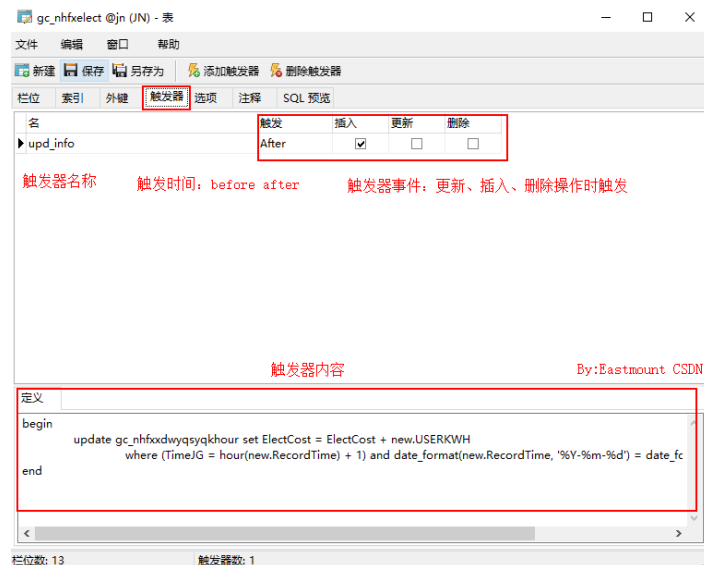
触发器SQL语法：

```
create trigger triggerName
after/before insert/update/delete on 表名
for each row #这句话在mysql是固定的
begin
    sql语句;
end;
```

推荐大家阅读：[mysql之触发器trigger - 郑志伟](#)

同时使用Navicat for MySQL创建触发器方法如下图所示，但是建议使用语句操作。

首先在Navicat for MySQL找到需要建立触发器对应的表，右键“设计表”，然后创建触发器。



二、简单的Insert触发器

假设存在一张学生表（student），包括学生的基本信息，学号（stuid）为主键。

username	password	stuid	birthday
王二	111111	1	2016-08-23
李四	123456	2	2016-07-23
张三	123456	3	2016-08-17
刘五	000000	4	2016-08-18
黄六	666666	5	2016-08-23

另外存在一张成绩表（cj），对应每个学生包括一个值。其中number表示序号为主键，自动递增序列。它在插入过程中默认自增。同时假设成绩表中包括学生姓名和学号。

名	类型	长度	小数点	允许空值 (1)
number	int	8	0	<input checked="" type="checkbox"/>
stu_id	varchar	30	0	<input checked="" type="checkbox"/>
stu_name	varchar	30	0	<input checked="" type="checkbox"/>
math	float	0	0	<input checked="" type="checkbox"/>
chinese	float	0	0	<input checked="" type="checkbox"/>
english	float	0	0	<input checked="" type="checkbox"/>

默认: []

注释: []

☒ 自动递增

☐ 无符号

☐ 填充零

该成绩表目前没有值，先需要设计一个触发器，当增加新的学生时，需要在成绩表中插入对应的学生信息，至于具体math、chinese、english后面由老师打分更新即可。

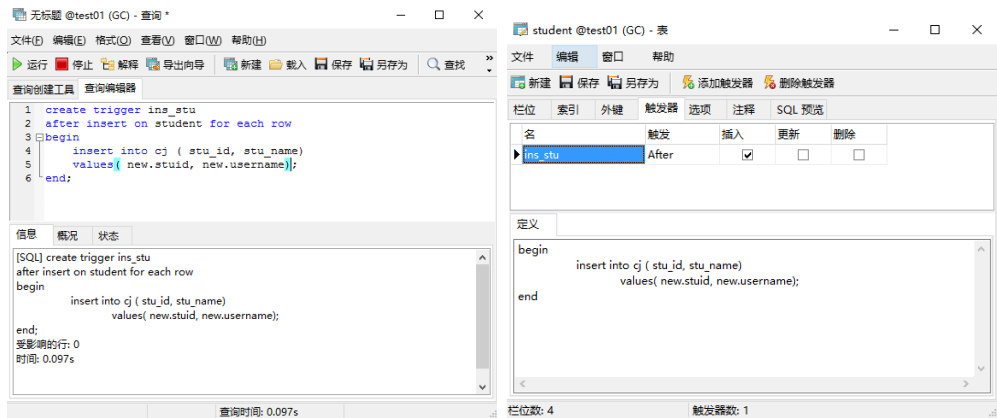
那么，如何设计触发器呢？

- 1.首先它是一个插入Insert触发器，是建立在表student上的；
- 2.然后是after，插入后的事件；

3.事件内容是插入成绩表，主需要插入学生的学号和姓名， number为自增，而成绩目前不需要。
注意： new表示student中新插入的值。

```
create trigger ins_stu
after insert on student for each row
begin
    insert into cj ( stu_id, stu_name)
        values( new.stuid, new.username);
end;
```

创建的触发器如下图所示：



然后插入数据：
`insert student values ('eastmount','111111','6','1991-12-05');`
同时插入两个数据，触发器正确执行了~

number	stu_id	stu_name	math	chinese	english
1	6	eastmount	(Null)	(Null)	(Null)

username	password	stuid	birthday
王二	111111	1	2016-08-23
李四	123456	2	2016-07-23
杨三	123456	3	2016-08-17
刘五	000000	4	2016-08-18
黄六	666666	5	2016-08-23
eastmount	111111	6	1991-12-05

注意：创建触发器和表一样，建议增加判断：DROP TRIGGER IF EXISTS `ins_stu`;

三、判断值后调用触发器

这里简单讲述几个判断插入类型的触发器。
比如触发器调用，当插入时间为20时，对数据进行插入：

```
DROP TRIGGER IF EXISTS `ins_info`;
create trigger ins_info
after insert on nhfxselect for each row
begin
    if HOUR(new.RecordTime)='20' then
```

```

insert into nhfxbyhour (UnitDepName, UnitDepCode, ElectCost, TimeJG, RecordTime)
values( '数统学院', '1', new.USERKWH, '20', new.RecordTime);

end if;

end;

```

这个触发器中，RecordTime为datetime类型，如"2016-08-28 20:10:00"，这时hour()这个值为20才能插入；否则数据不能插入。同时可以date_format(new.RecordTime, '%Y-%m-%d')判断日期为某天或某年某月进行插入。

同时，再如更新触发器，如果设置的值为某个范围，才进行操作或性别为"男"或"女"才进行操作。

基本语法：

```

if 判断条件 then
    sql语句;
end if;

```

四、Update触发器-实时更新

假设存在一个实时插入数据的服务器，例如学生的消费金额或用电量等。

StuCost：学生的用电数据，实时插入，Cost为每30秒消费金额，RecordTime为每分钟插入时间，datetime类型；

StuCostbyHour：统计学生一小时的消费金额，HourCost为金额总数，按小时统计，TimeJD时间段，1~24，对应每小时，RecordTime为统计时间。

现在需要设计一个实时更新触发器，当插入消费数据时，按小时统计学生的消费金额，同理，用电量等。

```

DROP TRIGGER IF EXISTS `upd_info`;
create trigger upd_info
after insert on StuCost for each row
begin
    update StuCostbyHour set HourCost = HourCost + new.Cost
    where (TimeJD = hour(new.RecordTime) + 1) and date_format(new.RecordTime, '%Y-%m-%d') =
date_format(RecordTime, '%Y-%m-%d');
end;

```

SQL语句中，需要获取插入的时间，然后通过TimeJD时间段和日期RecordTime找到对应的值，然后进行累加即可。如下图所示：

RecordTime	TimeSpace	Remarks	ElectCost	TimeJG	RecordTime
0000-00-00 00:00:00.000000	60	(Null)	2602.49	1	2016-08-28 00:30:00
2016-08-28 11:32:00.000000	60	(Null)	2646.6	2	2016-08-28 01:30:00
0000-00-00 00:00:00.000000	60	(Null)	2646.6	3	2016-08-28 02:30:00
2016-08-28 11:33:00.000000	60	(Null)	2646.6	4	2016-08-28 03:30:00
0000-00-00 00:00:00.000000	60	(Null)	2646.6	5	2016-08-28 04:30:00
2016-08-28 11:34:00.000000	60	(Null)	2646.6	6	2016-08-28 05:30:00
0000-00-00 00:00:00.000000	60	(Null)	2646.6	7	2016-08-28 06:30:00
2016-08-28 11:35:00.000000	60	(Null)	2602.49	8	2016-08-28 07:30:00
0000-00-00 00:00:00.000000	60	(Null)	2646.6	9	2016-08-28 08:30:00
2016-08-28 11:36:00.000000	60	(Null)	2646.6	10	2016-08-28 09:30:00
0000-00-00 00:00:00.000000	60	(Null)	2646.6	11	2016-08-28 10:30:00
2016-08-28 11:37:00.000000	60	(Null)	2249.61	12	2016-08-28 11:30:00
	60	(Null)	0	13	2016-08-28 12:30:00

上图左边是实时插入数据，右边是触发器更新加和。后面会介绍MySQL实时事件：

<http://blog.csdn.net/zlp5201/article/details/38309095>

五、触发器尽量避免

下面简单参考知乎和CSDN论坛，简单讲解几个内容：

问题一：

[大型系统必须得要存储过程和触发器吗？ - 知乎](#)

回答1：

我们先要弄清楚二个问题：

- 1.什么是大型系统？
- 2.你讨论的是什么领域的应用，可以大致分为二种：互联网、企业内部

接下来给你举一些例子：

- 1.SAP、peopleSoft、ERP等企业级别应用

一般情况下，会使用存储过程和触发器，减少开发成本，毕竟其业务逻辑修改频繁，而且为通用，很多时候会把一些业务逻辑编写成存储过程，像Oracle会写成包，比存储过程更强大。

另外一个原因是服务器的负载是可控，也即系统的访问人数首先是可控的，没有那么大，而且这些数据又非常关键，为此往往使用的设备也比较好，多用存储柜子支撑数据库。

- 2.另外一类互联网行业的

比如淘宝、知乎、微博等，数据库的压力是非常大的，也往往会最容易成为瓶颈，而且多用PC服务器支撑，用户量的增速是不可控的，同时在线访问的用户量也是不可控的，为此肯定会把业务逻辑放到其他语言的代码层，而且可以借助一些LVS等类型软硬件做负载均衡，以及平滑增减Web层的服务器，从而达到线性的增减而支持大规模的访问。

所以不管你的这个系统是否庞大，首先要分业务支持的对象，系统最可能容易出现瓶颈的地方在那？

当然也不是说互联网行业的应用就绝对不用存储过程，这个也不对，曾在阿里做的Oracle迁移MySQL系统确实用了，因为历史的原因，另外还有一些新系统也有用，比如晚上进行定期的数据统计的一些操作，不过有量上的控制。存储过程是好东西，要分场景，分业务类型来用就可以把握好。

回答2：

肯定不能一刀切的说能用或者不能用，不同类型的系统、不同的规模、不同的历史原因都会有不同的解决方案。

一般情况下，Web应用的瓶颈常在DB上，所以会尽可能的减少DB做的事情，把耗时的服务做成Scale Out，这种情况下，肯定不会使用存储过程；而如果只是一般的应用，DB没有性能上的问题，在适当的场景下，也可以使用存储过程。

至于触发器，我是知道有这东西但从来没用过。我希望风险可控，遇到问题能够快速找到原因，尽可能不会去使用触发器。

回答3：

- 1.PLSQL可以大大降低parse/exec 百分比；
- 2.存储过程可以自动完成静态SQL variable bind；
- 3.存储过程大大减少了JDBC网络传输与交互，速度快；
- 4.oracle 中存储过程内部commit为异步写，一定程度上减少了等redo日志落地时间；
- 5.存储过程最大问题就是给数据库开发工作压力太大，另外架构升级时候会比较难解耦；
- 6.触发器不推荐使用，触发操作能在业务层解决就在业务层解决，否则很难维护，而且容易产生死锁。

问题2：

[为什么大家都不推荐使用MySQL触发器而用存储过程？ - segmentfault](#)

回答1：

1.存储过程和触发器二者是有很大的联系的，我的一般理解就是触发器是一个隐藏的存储过程，因为它不需要参数，不需要显示调用，往往在你不知情的情况下已经做了很多操作。从这个角度来说，由于是隐藏的，无形中增加了系统的复杂性，非DBA人员理解起来数据库就会有困难，因为它不执行根本感觉不到它的存在。

2.再有，涉及到复杂的逻辑的时候，触发器的嵌套是避免不了的，如果再涉及几个存储过程，再加上事务等等，很容易出现死锁现象，再调试的时候也会经常性的从一个触发器转到另外一个，级联关系的不断追溯，很容易使人头大。其实，从性能上，触发器并没有提升多少性能，只是从代码上来说，可能在coding的时候很容易实现业务，所以我的观点

是：**摒弃触发器！触发器的功能基本都可以用存储过程来实现。**

3.在编码中存储过程显示调用很容易阅读代码，触发器隐式调用容易被忽略。

4.存储过程的致命伤在于移植性，存储过程不能跨库移植，比如事先是在mysql数据库的存储过程，考虑性能要移植到oracle上面那么所有的存储过程都需要被重写一遍。

回答2:

这种东西只有在并发不高的项目，管理系统中用。如果是面向用户的高并发应用，都不要使用。

触发器和存储过程本身难以开发和维护，不能高效移植。触发器完全可以用事务替代。存储过程可以用后端脚本替代。

回答3:

我觉得来自两方面的因素：

1.存储过程需要显式调用，意思是阅读源码的时候你能知道存储过程的存在，而触发器必须在数据库端才能看到，容易被忽略。

2.Mysql的触发器本身不是很好，比如after delete无法链式反应的问题。

我认为性能上其实还是触发器占优势的，但是基于以上原因不受青睐。

最后希望这篇文章对你有所帮助，尤其是学习MySQL触发器的同学，你可以通过触发器实现一些功能，同时需要注意合理的使用触发器，但这个过程需要你不断的去积累和开发，才能真正理解它的用法和使用场所。

(By:Eastmount 2016-08-28 下午2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