

浙江工业大学

数据库原理及应用实验报告

(2021 级)



实验题目 实验 8 触发器

学生姓名 温家伟

学生学号 202103151422

学科(专业) 大数据分析 2101 班

所在学院 理学院

提交日期 2023 年 5 月 16 日

实验8、触发器

8.1 实验目的

通过实验使学生加深对数据完整性的理解，学会理解、创建和使用触发器。

8.2 实验内容

(1) 为 Tutor 表建立触发器 b4ins_upd_trig，当插入或使更新表中的数据时，保证所操作的纪录的 Tage 值大于0。

(2) 为 Tutor 表建立触发器 b4del_trig，禁止删除编号为00001的 CEO。

- (3) Tutor 表中的人员的编号是唯一且不可变更的，创建触发器 b4upd_trig 实现更新中编号的不可改变性。
- (4) 演示违反 b4ins_upd_trig 触发器的约束的插入操作。
- (5) 演示违反 b4ins_upd_trig 触发器的约束的更新操作。
- (6) 演示违反 b4del_trig 触发器的约束的插入操作。
- (7) 演示违反 b4upd_trig 触发器的约束的更新操作。

8.3 实验步骤

- (1) 仍然使用自定义完整性实验中的 Tutor 表。为此表建立触发器 T1，当插入或使更新表中的数据时，保证所操作的纪录的 Tage 值大于0。

```
ahweidb=# CREATE TABLE Tutor
ahweidb=# (
ahweidb(# Tno CHAR(5),
ahweidb(# Tname CHAR(8) CONSTRAINT con_uni_tname UNIQUE,
ahweidb(# Tsex CHAR(3),
ahweidb(# Tage INT CONSTRAINT con_ulimit_tage CHECK (Tage<=60),
ahweidb(# Tdept CHAR(20),
ahweidb(# CONSTRAINT PK_Tutor PRIMARY KEY(Tno)
ahweidb(# );
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "pk_tutor" for table "tutor"
NOTICE: CREATE TABLE / UNIQUE will create implicit index "con_uni_tname" for table "tutor"
CREATE TABLE
ahweidb=# INSERT INTO Tutor (Tno,Tname,Tsex,Tage,Tdept)
ahweidb=# VALUES('T01','李用','M',24,'后勤部');
INSERT 0 1
ahweidb=# select * from Tutor;
   tno  | tname  | tsex | tage |      tdept
-----+-----+-----+-----+-----
T01    | 李用   | M    | 24   | 后勤部
(1 row)
```

在 openGauss 的控制台中输入如下 SQL 语句

```
ahweidb=# CREATE OR REPLACE FUNCTION b4ins_upd_trig_func() RETURNS TRIGGER AS
ahweidb=# $$
ahweidb$$ DECLARE
ahweidb$$ BEGIN
ahweidb$$ IF NEW.Tage<1 THEN
ahweidb$$ RAISE NOTICE '职工年龄必须是大于0的整数！操作失败！';
ahweidb$$ RETURN NULL;
ahweidb$$ ELSE
ahweidb$$ RETURN NEW;
ahweidb$$ END IF;
ahweidb$$ END
ahweidb$$ $$ LANGUAGE PLPGSQL;
FOR EACH ROW
EXECUTE PROCEDURE b4ins_upd_trig_func();CREATE FUNCTION
ahweidb=#
ahweidb=# CREATE TRIGGER b4ins_upd_trig
ahweidb=# BEFORE INSERT OR UPDATE ON Tutor
ahweidb=# FOR EACH ROW
ahweidb=# EXECUTE PROCEDURE b4ins_upd_trig_func();
CREATE TRIGGER
```

```
ahweidb=# drop trigger b4ins_upd_trig on tutor;
DROP TRIGGER
ahweidb=# Drop function b4ins_upd_trig_func;
DROP FUNCTION
```

- (1') 对每个记录插入到 COMPANY 表中进行审核(审计)。

使用以下查询创建一个名为 `COMPANY` 的表：

```
ahweidb=# CREATE TABLE COMPANY(
ahweidb(# ID INT PRIMARY KEY NOT NULL,
ahweidb(# NAME TEXT NOT NULL,
ahweidb(# AGE INT NOT NULL,
ahweidb(# ADDRESS CHAR(50),
ahweidb(# SALARY REAL);
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "company_pkey" for table "company"
CREATE TABLE
```

为了保存审计/审核，我们将创建一个名为 `audit` 的新表，只要在 `COMPANY` 表中有一个新记录的条目，就会插入日志消息。

使用以下查询语句创建另一个表 `audit`：

```
ahweidb=# CREATE TABLE audit( EMP_ID INT NOT NULL,
ahweidb(# ENTRY_DATE TEXT NOT NULL );
CREATE TABLE
```

在 `COMPANY` 表上创建触发器之前，首先创建一个名为 `auditlogfunc()` 的函数/过程。执行以下查询语句来创建函数/过程：

```
ahweidb=# CREATE OR REPLACE FUNCTION auditlogfunc() RETURNS TRIGGER AS $example_table$
ahweidb$$ BEGIN
ahweidb$$ INSERT INTO AUDIT(EMP_ID, ENTRY_DATE) VALUES (new.ID, current_timestamp);
ahweidb$$ RETURN NEW;
ahweidb$$ END;
ahweidb$$ $example_table$ LANGUAGE plpgsql;
CREATE FUNCTION
ahweidb=#
```

现在通过使用以下查询语句在 `COMPANY` 表上创建一个触发器：

```
ahweidb=# CREATE TRIGGER example_trigger AFTER INSERT ON COMPANY
ahweidb-# FOR EACH ROW EXECUTE PROCEDURE auditlogfunc();
CREATE TRIGGER
```

向 `COMPANY` 表中插入一些数据记录，以验证触发器执行情况。

在执行上面两条插入语句后，现我们来看 `AUDIT` 表是否有自动插入两条审核记录。

```
ahweidb=# INSERT INTO COMPANY VALUES(1, '小米科技', 8, '北京市朝阳区', 9999);
INSERT 0 1
ahweidb=# INSERT INTO COMPANY VALUES(2, '京东中科', 6, '广州市天河区', 8999);
INSERT 0 1
ahweidb=# select * from COMPANY;
 id |  name  | age |          address          | salary
-----+-----+-----+-----+-----
  1 | 小米科技 |   8 | 北京市朝阳区          |   9999
  2 | 京东中科 |   6 | 广州市天河区          |   8999
(2 rows)
```

(2) 为 `Tutor` 表建立触发器 `T2`，禁止删除编号为 `S01` 的 CEO。

在 `OpenGauss` 的控制台中输入如下 SQL 语句

一种正确的做法：

在 `BEFORE DELETE` 触发的情况下，返回值没有直接影响，但它必须为非空才能继续触发操作。返回 `null` 以通知触发器管理器跳过该行的其余操作（即，不触发后续触发器，并且该行不发生 `INSERT/UPDATE/DELETE`）。请注意，`DELETE` 触发器中的 `NEW` 为 `null`，因此返回它通常是不明智的。`DELETE` 触发器中的惯用法是返回 `OLD`。

```

ahweidb=# CREATE OR REPLACE FUNCTION b4del_trig_func() RETURNS TRIGGER AS
ahweidb-#      $$
ahweidb$#      DECLARE
ahweidb$#      BEGIN
ahweidb$#          IF OLD.Tno='T01' THEN
ahweidb$#              RAISE NOTICE '此人是CEO! 删除操作失败!';
ahweidb$#              RETURN NULL; --返回空,
ahweidb$#          ELSE
ahweidb$#              RETURN OLD;
ahweidb$#          END IF;
ahweidb$#      END
ahweidb$#      $$ LANGUAGE PLPGSQL;
ahweidb=# EXECUTE PROCEDURE b4del_trig_func();CREATE FUNCTION
ahweidb=#
ahweidb=#      CREATE TRIGGER b4del_trig
ahweidb-#      BEFORE DELETE ON Tutor
ahweidb-#      FOR EACH ROW
ahweidb-#      EXECUTE PROCEDURE b4del_trig_func();
ahweidb=# CREATE TRIGGER

```

(3) Tutor表中的人员的编号是唯一且不可变更的，创建触发器T3实现更新中编号的不可改变性。

在OpenGauss的控制台中输入如下SQL语句

```

ahweidb=# CREATE OR REPLACE FUNCTION b4upd_trig_func() RETURNS TRIGGER AS
ahweidb-#      $$
ahweidb$#      DECLARE
ahweidb$#      BEGIN
ahweidb$#          RAISE NOTICE '职工编号不能修改!';
ahweidb$#          Return null;
ahweidb$#      END
ahweidb$#      $$ LANGUAGE PLPGSQL;
ahweidb=# CREATE FUNCTION
ahweidb=#
ahweidb=#      CREATE TRIGGER b4upd_trig
ahweidb-#      BEFORE UPDATE OF Tno ON Tutor
ahweidb-#      FOR EACH ROW
ahweidb-#      EXECUTE PROCEDURE b4upd_trig_func();
ahweidb=# CREATE TRIGGER

```

(4) 在OpenGauss的控制台中输入如下SQL语句

```

ahweidb=# INSERT INTO Tutor VALUES('T06','李宏','F',-8,'开发部');
NOTICE:  职工年龄必须是大于0的整数! 操作失败!
INSERT 0 0
ahweidb=# INSERT INTO Tutor VALUES('T05','李宏','F',8,'开发部');
INSERT 0 1
ahweidb=# select * from Tutor;
   tno  |  tname  | tsex |  tage  |      tdept
-----+-----+-----+-----+-----
T01    | 李用    | M    |    24  | 后勤部
T05    | 李宏    | F    |     8  | 开发部
(2 rows)

```

(5) 在OpenGauss的控制台中输入如下SQL语句

```

ahweidb=# UPDATE Tutor SET Tage=-7 WHERE Tno='T01';
NOTICE:  职工年龄必须是大于0的整数! 操作失败!
UPDATE 0

```

(6) 在OpenGauss的控制台中输入如下SQL语句

```
ahweidb=# DELETE FROM Tutor WHERE Tname='李用';
NOTICE:  此人是CEO! 删除操作失败!
DELETE 0
```

(7) 在OpenGauss的控制台中输入如下SQL语句

```
mysql> select count(*) as 总数, avg(Score) as 平均分, max(Score) as 最高分, min(Score) as 最低分 from STC where Score < 0;
+-----+-----+-----+-----+
| 总数 | 平均分 | 最高分 | 最低分 |
+-----+-----+-----+-----+
| 0 | NULL | NULL | NULL |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

ahweidb=# UPDATE Tutor SET Tno='T07' WHERE Tsex='F';
NOTICE:  职工编号不能修改!
UPDATE 0
```