Database Programming with PL/SQL

13-2: Creating DML Triggers: Part I

Practice Activities

Vocabulary

Identify the vocabulary word for each definition below.

Row trigger	fires once for each row affected by the triggering event.
DML trigger	A trigger which is automatically fired (executed) whenever a SQL DML statement (INSERT, UPDATE or DELETE) is executed.
Statement trigger	is fired once on behalf of the triggering event, even if no rows are affected at all.

Try It / Solve It

1. When creating a DML statement trigger on a table, what are the components that you must define?

El nombre del trigger

El timing (Before, After, Instead of)

El tipo de evento (Insert, Update, Delete)

El nombre de la tabla o vista asociada al trigger

El cuerp del trigger

2. A business rule states that each time one or more employees are added to the EMPLOYEES table, an audit record must also be created. This rule could be enforced using application code, but we have decided to enforce it using a DML statement trigger.

A. Create the AUDIT_TABLE by executing the following SQL statement:

```
CREATE TABLE audit_table
   (action          VARCHAR2(50),
    user_name       VARCHAR2(30) DEFAULT USER,
last_change_date    TIMESTAMP DEFAULT SYSTIMESTAMP);
```

B. Create a statement-level trigger that inserts a row into the AUDIT_TABLE immediately after one or more rows are added to the EMPLOYEES table. The AUDIT_TABLE row should contain value “Inserting” in the action column. The other two columns should have their default values. Save your trigger code for later.

```
CREATE OR REPLACE TRIGGER tg_ins_audit after INSERT ON EMPLOYEES
BEGIN
Insert into AUDIT_TABLE VALUES('INSERTING', USER, SYSDATE);
END;
```

C. Test your trigger by inserting a row into EMPLOYEES, then querying the AUDIT_TABLE to see that it contains a row.

```
INSERT INTO EMPLOYEES(employee_id,first_name,last_name,email,hire_date,job_id)
VALUES(999,'Jorge','Meza','1730093@upv.edu.mx',SYSDATE,'SA_REP');
```

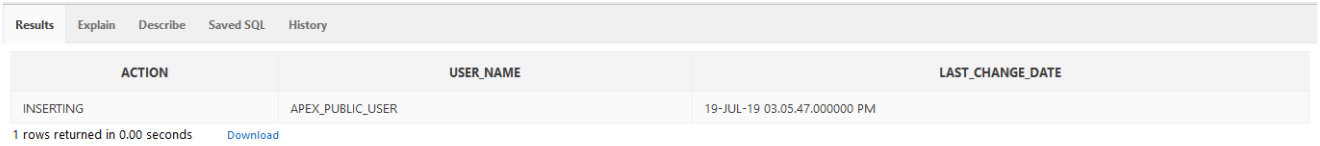
Results | Explain | Describe | Saved SQL | History

ACTION	USER_NAME	LAST_CHANGE_DATE
INSERTING	APEX_PUBLIC_USER	19-JUL-19 03.05.47.000000 PM

1 rows returned in 0.00 seconds Download

D. Make sure the trigger does not fire with a DELETE by deleting the employee you just entered. Recheck the AUDIT_TABLE to make sure that there is not another new row.

El disparador no se volvió a ejecutar cuando se elimino la fila que se inserto.



Results | Explain | Describe | Saved SQL | History

ACTION	USER_NAME	LAST_CHANGE_DATE
INSERTING	APEX_PUBLIC_USER	19-JUL-19 03.05.47.000000 PM

1 rows returned in 0.00 seconds Download

3. True or false? A row trigger fires at least once even if no rows are affected.

FALSE

What is the difference between a statement trigger and a row trigger?

Un row trigger se activa cada que es afectada una fila. En cambio, el statement trigger se activa cada que es ejecutada una sentencia, no importa cuantas filas se vean afectadas por esta sentencia.

4. Imagine that the following DML triggers have been defined on the EMPLOYEES table:
   - A Before Insert statement trigger
   - A Before Update statement trigger
   - An After Delete statement trigger

   An UPDATE statement updates three employee rows. How many times will each trigger fire?

A Before Insert statement trigger No se ejecuta.

A Before Update statement trigger Se ejecuta una sola vez.

An After Delete statement trigger No se ejecuta.

5. Modify your AUDIT_TABLE trigger from question 2B so that it inserts a row into the audit table immediately before one or more employee salaries are updated. The AUDIT_TABLE row should contain value “Updating” in the action column.

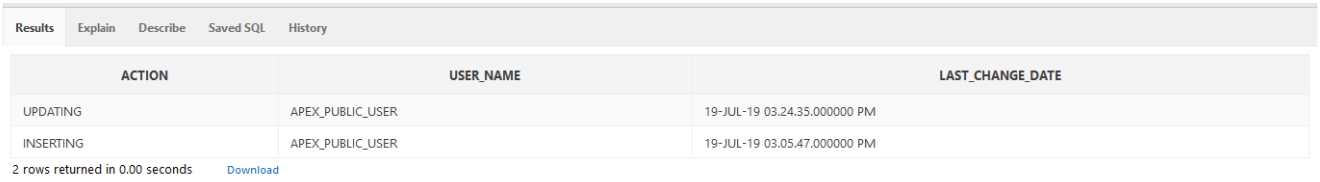
   Test your modified trigger by updating the salary of a non-existent employee (employee_id = 999), then querying the AUDIT_TABLE to see that it contains a new row.

```
CREATE OR REPLACE TRIGGER TG_UP_SALARY BEFORE UPDATE OF SALARY
ON EMPLOYEES

BEGIN

Insert into AUDIT_TABLE VALUES('UPDATING', USER, SYSDATE);

END;
```



Results | Explain | Describe | Saved SQL | History

ACTION	USER_NAME	LAST_CHANGE_DATE
UPDATING	APEX_PUBLIC_USER	19-JUL-19 03.24.35.000000 PM
INSERTING	APEX_PUBLIC_USER	19-JUL-19 03.05.47.000000 PM

2 rows returned in 0.00 seconds Download

A. Modify your trigger so that it prevents employees’ salaries being updated outside working hours. The trigger should allow UPDATEs at other times (and still insert a row into the AUDIT_TABLE), but should raise an application error if an update is attempted before 8:00 am or after 6:00 pm on any day. (HINT: use HH24:MI to extract the time from SYSDATE).

```
CREATE OR REPLACE TRIGGER TG_UP_SALARY BEFORE UPDATE OF SALARY ON
EMPLOYEES

declare

hora NUMBER(2);

BEGIN

hora:=to_char(sysdate,'HH24');

IF hora <= 8 OR hora >= 18 THEN

    RAISE_APPLICATION_ERROR (-20204,'La actualizacion no se puede realizar a estas
horas');

ELSE
```

```
Insert into AUDIT_TABLE VALUES('UPDATING', USER, SYSDATE);

END IF;

END;
```

B. You want to test your modified trigger. However, you need to make sure that right now the database time is outside working hours. Remember that the database could be anywhere in the world and therefore the database may not be in your time zone! Find the current database time by executing:

SELECT TO_CHAR(SYSDATE,'HH24:MI') FROM dual;

If needed, modify your trigger so that it will raise the application error if you try to update a salary within the next hour. For example, if the database time is 10:30, modify the trigger code to include: …BETWEEN '10:30' AND '11:30'…

Test your modified trigger by trying to update the salary of employee_id 100 to a new value of 25000. You should see the ORA-20204 error message.

```
UPDATE employees SET salary = 25000
WHERE employee_id = 100;
```

ACTION	USER_NAME	LAST_CHANGE_DATE
UPDATING	APEX_PUBLIC_USER	19-JUL-19 03.24.35.000000 PM
INSERTING	APEX_PUBLIC_USER	19-JUL-19 03.05.47.000000 PM
UPDATING	APEX_PUBLIC_USER	19-JUL-19 04.01.05.000000 PM
UPDATING	APEX_PUBLIC_USER	19-JUL-19 04.01.47.000000 PM

4 rows returned in 0.00 seconds Download