

TALLER No. 1 TALLER TRIGGER Y CURSOR

OBJETIVO:

- Crear triggers de la base de datos
- Borrar triggers de la base de datos
- Declarar y usar cursores.

ACTIVIDADES A DESARROLLAR:

1. Construir desencadenadores (trigger)
2. Construir un cursor

EVIDENCIA(S) A ENTREGAR:

EV1 Desarrollar trigger y cursor de acuerdo a:

Se debe entregar un documento llamado "TALLERTRIGGERCURSOR.sql", que contenga:

- Creación tabla LOGSALARIO
- Creación del trigger TRGSALARIO
- Creación del procedimiento modificarSalario que contenga el cursor llamado CURSOR_SALARIO

CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	JOSE FERNANDO GALINDO S.	Instructor	Teleinformática	28/05/2020

CONTROL DE CAMBIOS (diligenciar únicamente si realizan ajustes al taller)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

PRESENTACIÓN

Los desencadenadores y cursores son sentencias que utilizan para la gestión de la información en una base de datos.

La incorporación de los trigger se comenzó a utilizar desde la versión mysql 5.0.2, son objetos con nombre asociado a una tabla y es activada cuando se realiza un evento INSERT, UPDATE o DELETE.

DESENCADENADORES

Definición:

```
CREATE TRIGGER Nombre Momento Evento ON nomre_tabla
FOR EACH ROW
BEGIN
    ... Cuerpo o llamada
END;
```

Donde:

Momento: **BEFORE** antes de grabar el registro en la tabla, **AFTER** despues de grabar el registro en la tabla.

Evento: **INSERT** un nuevo registro, **UPDATE** actualizar el registro, **DELETE** borrar el registro.

Acceso a los datos de la tabla de acuerdo a la DML del evento.

Evento	Toma de datos
INSERT	new.nombredecampodelatabla
UPDATE	new.nombredecampodelatabla old.nombredecampodelatabla
DELETE	old.nombredecampodelatabla

Borrar un trigger en la base de datos:

```
DROP TRIGGER nombre_trigger;
```

PRACTICA DEL TALLER

Crear la tabla "LOGDEPARTAMENTO"

```
MariaDB [HR]> CREATE TABLE LOGDEPARTAMENTO(ID INTEGER,
-> OPERACION VARCHAR(1),
-> FECHA TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
-> USUARIO VARCHAR(30),
-> VALORN VARCHAR(50),
-> VALORV VARCHAR(50)
-> );
Query OK, 0 rows affected (0.04 sec)

MariaDB [HR]> desc LOGDEPARTAMENTO;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ID    | int(11) | YES |     | NULL    |       |
| OPERACION | varchar(1) | YES |     | NULL    |       |
| FECHA | timestamp | NO |     | CURRENT_TIMESTAMP |       |
| USUARIO | varchar(30) | YES |     | NULL    |       |
| VALORN | varchar(50) | YES |     | NULL    |       |
| VALORV | varchar(50) | YES |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.04 sec)
```

Crear el trigger para el evento INSERT

```
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE TRIGGER TRDEPARTMENTS_I AFTER INSERT ON DEPARTMENTS FOR EACH ROW
-> BEGIN
-> INSERT LOGDEPARTAMENTO VALUES(NEW.DEPARTMENT_ID,'I',NOW(),USER(),NEW.DEPARTMENT_NAME,NULL);
-> END;
-> $
Query OK, 0 rows affected (0.02 sec)

MariaDB [HR]> DELIMITER ;
```

1. Insertamos en la tabla DEPARTMENTS
2. Verificamos que se haya creado en la tabla DEPARTMENTS.
3. Si el desencadenador funcionó, verificamos en la tabla LOGDEPARTAMENTO.

```
MariaDB [HR]> INSERT INTO DEPARTMENTS VALUES(300,'INFORMATICA',103,1400); 1
Query OK, 1 row affected (0.04 sec)

MariaDB [HR]> SELECT * FROM DEPARTMENTS WHERE DEPARTMENT_ID=300; 2
+-----+-----+-----+-----+
| department_id | department_name | manager_id | location_id |
+-----+-----+-----+-----+
| 300 | INFORMATICA | 103 | 1400 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [HR]> SELECT * FROM LOGDEPARTAMENTO; 3
+-----+-----+-----+-----+-----+-----+
| ID | OPERACION | FECHA | USUARIO | VALORN | VALORV |
+-----+-----+-----+-----+-----+-----+
| 300 | I | 2020-05-28 11:11:37 | root@localhost | INFORMATICA | NULL |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Crear el trigger para el evento UPDATE

```
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE TRIGGER TRDEPARTMENTS_U AFTER UPDATE ON DEPARTMENTS FOR EACH ROW
-> BEGIN
-> INSERT LOGDEPARTAMENTO VALUES(NEW.DEPARTMENT_ID,'U',NOW(),USER(),NEW.DEPARTMENT_NAME,OLD.DEPARTMENT_NAME);
-> END;
-> $
Query OK, 0 rows affected (0.02 sec)

MariaDB [HR]> DELIMITER ;
```

1. Actualizamos en la tabla DEPARTMENTS el registro 300, cambiamos "INFORMATICA" POR "IT & SISTEMAS"
2. Verificamos que se modificó en la tabla DEPARTMENTS.

- Si el desencadenador funcionó, verificamos en la tabla LOGDEPARTAMENTO.

```
MariaDB [HR]> UPDATE DEPARTMENTS SET DEPARTMENT_NAME='IT Y & SISTEMAS' WHERE DEPARTMENT_ID=300; 1
Query OK, 1 row affected (0.06 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [HR]> SELECT * FROM DEPARTMENTS WHERE DEPARTMENT_ID=300; 2
+-----+-----+-----+-----+
| department_id | department_name | manager_id | location_id |
+-----+-----+-----+-----+
| 300 | IT Y & SISTEMAS | 103 | 1400 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [HR]> SELECT * FROM LOGDEPARTAMENTO; 3
+-----+-----+-----+-----+-----+-----+
| ID | OPERACION | FECHA | USUARIO | VALORN | VALORV |
+-----+-----+-----+-----+-----+-----+
| 300 | I | 2020-05-28 11:11:37 | root@localhost | INFORMATICA | NULL |
| 300 | U | 2020-05-28 11:21:54 | root@localhost | IT Y & SISTEMAS | INFORMATICA |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Crear el trigger para el evento UPDATE

```
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE TRIGGER TRDEPARTMENTS AFTER DELETE ON DEPARTMENTS FOR EACH ROW
-> BEGIN
-> INSERT LOGDEPARTAMENTO VALUES(OLD.DEPARTMENT_ID,'D',NOW(),USER(),NULL,OLD.DEPARTMENT_NAME);
-> END;
-> $
Query OK, 0 rows affected (0.03 sec)

MariaDB [HR]> DELIMITER ;
```

- Borramos en la tabla DEPARTMENTS el registro 300.
- Verificamos que se borró en la tabla DEPARTMENTS.
- Si el desencadenador funcionó, verificamos en la tabla LOGDEPARTAMENTO.

```
MariaDB [HR]> DELETE FROM DEPARTMENTS WHERE DEPARTMENT_ID=300; 1
Query OK, 1 row affected (0.03 sec)

MariaDB [HR]> SELECT * FROM DEPARTMENTS WHERE DEPARTMENT_ID=300; 2
Empty set (0.00 sec)

MariaDB [HR]> SELECT * FROM LOGDEPARTAMENTO; 3
+-----+-----+-----+-----+-----+-----+
| ID | OPERACION | FECHA | USUARIO | VALORN | VALORV |
+-----+-----+-----+-----+-----+-----+
| 300 | I | 2020-05-28 11:11:37 | root@localhost | INFORMATICA | NULL |
| 300 | U | 2020-05-28 11:21:54 | root@localhost | IT Y & SISTEMAS | INFORMATICA |
| 300 | D | 2020-05-28 11:42:15 | root@localhost | NULL | IT Y & SISTEMAS |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

CURSORES

Para la práctica de cursores debemos tener en cuenta las siguientes partes:

- 1- DECLARAR EL CURSOR
- 2- ABRIR EL CURSOR
- 3- CARGAR DATOS
- 4- MOSTRAR DATOS
- 5- CERRAR EL CURSOR

Para esto, crearemos un procedimiento llamado "PR_DEMOCURSOR" que contiene un cursor que carga las filas de la tabla EMPLOYEES que su JOB_ID pertenezcan a "IT_PROG", así:

```
MariaDB [HR]> select EMPLOYEE_ID,CONCAT(LAST_NAME,' ',FIRST_NAME) EMPLEADO,SALARY from employees where job_id='IT_PROG';
```

EMPLOYEE_ID	EMPLEADO	SALARY
103	Hunold Alexander	9000.00
104	Ernst Bruce	6000.00
105	Austin David	4800.00
106	Pataballa Valli	4800.00
107	Lorentz Diana	4200.00

5 rows in set (0.00 sec)

Creamos la tabla "IT_WORKER" solo la estructura

```
MariaDB [HR]> CREATE TABLE IT_WORKER AS
-> select EMPLOYEE_ID,CONCAT(LAST_NAME,' ',FIRST_NAME) EMPLEADO,SALARY,0 IVA,0 NETO
-> from employees where 1=2;
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [HR]> DESC IT_WORKER;
```

Field	Type	Null	Key	Default	Extra
EMPLOYEE_ID	int(11) unsigned	NO		NULL	
EMPLEADO	varchar(46)	YES		NULL	
SALARY	decimal(8,2)	NO		NULL	
IVA	int(1)	NO		NULL	
NETO	int(1)	NO		NULL	

5 rows in set (0.04 sec)

```
MariaDB [HR]> SELECT * FROM IT_WORKER;
Empty set (0.00 sec)
```


La idea es de correr el procedimiento “PR_DEMOCURSOR” y utilizar el cursor llamado “C_CURSOR” que calcula el IVA y el valor neto de su salario.

```
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE PROCEDURE PR_DEMOCURSOR() 1
-> BEGIN
-> DECLARE VNAME VARCHAR(50);
-> DECLARE VID INT(11);
-> DECLARE VSALARY INT(11);
-> DECLARE VIVA INT(11);
-> DECLARE VNETO INT(11);
->
-> DECLARE C_CURSOR CURSOR FOR select EMPLOYEE_ID,CONCAT(LAST_NAME,' ',FIRST_NAME) EMPLEADO,SALARY,0 IVA,0 NETO
-> FROM EMPLOYEES WHERE job_id='IT_PROG'; 2
->
-> OPEN C_CURSOR; 3
-> c1_loop: LOOP
-> FETCH C_CURSOR INTO VID,VNAME,VSALARY,VIVA,VNETO; 4
-> SET VNETO=VSALARY*0.84;
-> SET VIVA=VSALARY*0.16;
-> INSERT INTO IT_WORKER VALUES(VID,VNAME,VSALARY,VIVA,VNETO);
-> END LOOP c1_loop;
-> END;
-> $
Query OK, 0 rows affected (0.05 sec)

MariaDB [HR]> DELIMITER ;
MariaDB [HR]>
MariaDB [HR]> CALL PR_DEMOCURSOR(); 5
ERROR 1329 (02000): No data - zero rows fetched, selected, or processed
MariaDB [HR]> SELECT * FROM IT_WORKER; 6
```

EMPLOYEE_ID	EMPLEADO	SALARY	IVA	NETO
103	Hunold Alexander	9000.00	1440	7560
104	Ernst Bruce	6000.00	960	5040
105	Austin David	4800.00	768	4032
106	Pataballa Valli	4800.00	768	4032
107	Lorentz Diana	4200.00	672	3528

5 rows in set (0.00 sec)

ACTIVIDAD A REALIZAR

Realizar un procedimiento llamado modificarSalario que utilice un cursor con el id, el salario y el cargo que ocupa.

De acuerdo al cargo se hará un incremento así:

CARGO	INCREMENTO
IT_PROG	1.5%
HR_REP	1.6%
SA_REP	1.7%
ST_CLERK	1.5%
SH_CLERK	1.5%
AC_ACCOUNT	1.7%
FI_ACCOUNT	1.6%
El resto de cargos	1.2%

Cuando se actualice el salario en la tabla EMPLOYEES se debe construir un trigger llamado TRGSALARIO donde se guardará el id, el salario antiguo, el nuevo salario, el cargo, la fecha y el usuario que realizó la transacción en la tabla LOGSALARIO.

Realizado por el instructor José Fernando Galindo Suárez

jgalindos@sena.edu.co 2020

