TALLER No. 6
TALLER TRIGGER Y CURSOR

OBJETIVO:

- Crear triggers de la base de datos
- Borrar triggers de la base 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)

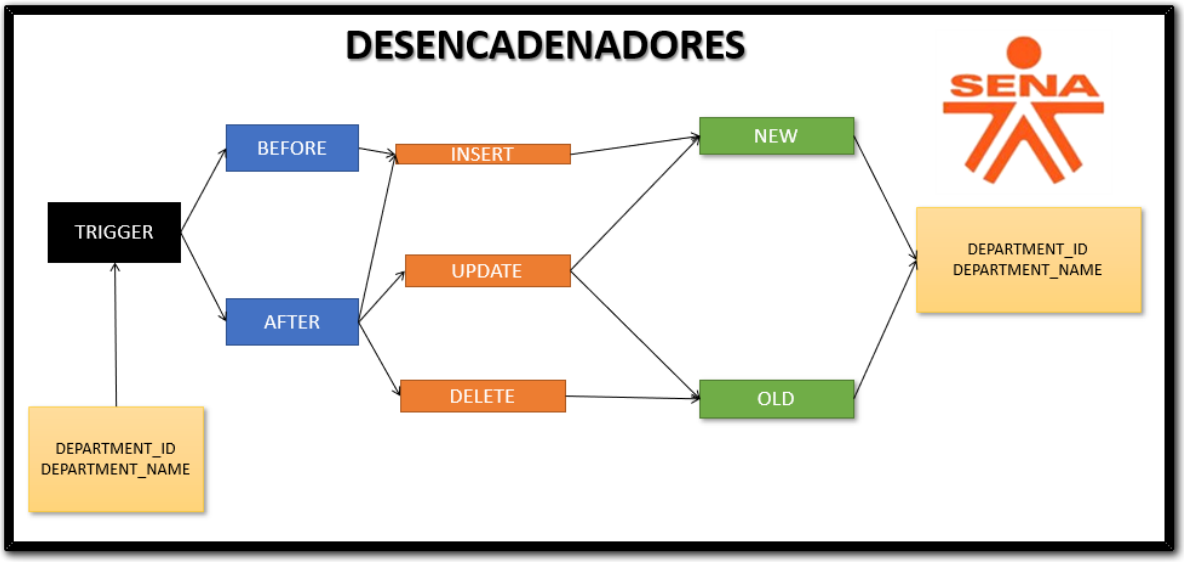
	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					

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** después de grabar el registro en la tabla.

Evento: **INSERT** un nuevo registro, **UPDATE** actualizar el registro, **DELETE** borra un registro. Acceso a los datos de la tabla de acuerdo con 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 vista "**LISTARTRIGGER**" y "**LISTARUTINAS**"

```
CREATE OR REPLACE VIEW LISTARTRIGGER AS
SELECT TRIGGER_SCHEMA BD,TRIGGER_NAME
NOMBRE,EVENT_MANIPULATION EVENTO,EVENT_OBJECT_TABLE
TABLA FROM INFORMATION_SCHEMA.TRIGGERS ;

CREATE OR REPLACE VIEW LISTARUTINAS AS
SELECT ROUTINE_SCHEMA BD,ROUTINE_NAME
NOMBRE,ROUTINE_TYPE TIPO FROM
INFORMATION_SCHEMA.ROUTINES;
```

```
MariaDB [HR]> SELECT * FROM LISTARTRIGGER WHERE BD='HR';
+----+----------------+--------+-------------+
| BD | NOMBRE         | EVENTO | TABLA       |
+----+----------------+--------+-------------+
| hr | TRDEPARTMENTSI | INSERT | departments |
| hr | TRDEPARTMENTSU | UPDATE | departments |
| hr | TRDEPARTMENTSD | DELETE | departments |
| hr | TREMPLEADOI    | INSERT | empleado    |
| hr | TRGSALARIO     | UPDATE | employees   |
+----+----------------+--------+-------------+
5 rows in set (0.08 sec)
```

```
MariaDB [HR]> SELECT * FROM LISTARUTINAS WHERE BD='HR';
+----+------------------+-----------+
| BD | NOMBRE           | TIPO      |
+----+------------------+-----------+
| hr | EJEMPLO1         | PROCEDURE |
| hr | EJEMPLO2         | PROCEDURE |
| hr | MODIFICARSALARIO | PROCEDURE |
| hr | PRDEMOCURSOR     | PROCEDURE |
+----+------------------+-----------+
4 rows in set (0.02 sec)
```

- Crear la tabla “LOGDEPARTAMENTO”

```
MariaDB [HR]> USE HR
Database changed
MariaDB [HR]> CREATE TABLE LOGDEPARTAMENTO(ID INTEGER,
    -> OPERACION VARCHAR(1),
    -> FECHA TIMESTAMP,
    -> USUARIO VARCHAR(30),
    -> VALORN VARCHAR(50),
    -> VALORV VARCHAR(50));
Query OK, 0 rows affected (0.03 sec)

MariaDB [HR]>
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 | on update CURRENT_TIMESTAMP |
| USUARIO   | varchar(30) | YES  |     | NULL              |                             |
| VALORN    | varchar(50) | YES  |     | NULL              |                             |
| VALORV    | varchar(50) | YES  |     | NULL              |                             |
+-----------+-------------+------+-----+-------------------+-----------------------------+
6 rows in set (0.02 sec)
```

- Crear el TRIGGER para el evento INSERT

```
MariaDB [HR]>
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE TRIGGER TRDEPARTMENTSI AFTER INSERT ON DEPARTMENTS FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO LOGDEPARTAMENTO
    ->     VALUES(NEW.DEPARTMENT_ID,'I',NOW(),USER(),NEW.DEPARTMENT_NAME,NULL);
    -> END;
    -> $
Query OK, 0 rows affected (0.03 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]>
MariaDB [HR]>
MariaDB [HR]> INSERT INTO DEPARTMENTS VALUES(280,'ASUNTOS SIN IMPORTANCIA',NULL,NULL); 1
Query OK, 1 row affected (0.02 sec)

MariaDB [HR]> SELECT * FROM DEPARTMENTS WHERE DEPARTMENT_ID=280; 2
+---------------+------------------------+------------+-------------+
| department_id | department_name        | manager_id | location_id |
+---------------+------------------------+------------+-------------+
|           280 | ASUNTOS SIN IMPORTANCIA |       NULL |        NULL |
+---------------+------------------------+------------+-------------+
1 row in set (0.00 sec)

MariaDB [HR]> SELECT * FROM LOGDEPARTAMENTO; 3
+------+-----------+---------------------+----------------+------------------------+--------+
| ID   | OPERACION | FECHA               | USUARIO        | VALORN                 | VALORV |
+------+-----------+---------------------+----------------+------------------------+--------+
|  280 | I         | 2020-09-04 07:52:04 | root@localhost | ASUNTOS SIN IMPORTANCIA | NULL   |
+------+-----------+---------------------+----------------+------------------------+--------+
1 row in set (0.00 sec)
```

- Crear el TRIGGER para el evento UPDATE

```
MariaDB [HR]>
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE TRIGGER TRDEPARTMENTSU AFTER UPDATE ON DEPARTMENTS FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO LOGDEPARTAMENTO
    -> VALUES(OLD.DEPARTMENT_ID,'A',NOW(),USER(),NEW.DEPARTMENT_NAME,OLD.DEPARTMENT_NAME);
    -> END;
    -> $
Query OK, 0 rows affected (0.04 sec)

MariaDB [HR]> DELIMITER ;
```

1. Actualizamos en la tabla DEPARTMENTS el registro 280, cambiamos "ASUNTOS SIN IMPORTANCIA" POR "ASUNTOS CULINARIOS"
2. Verificamos que se modificado en la tabla DEPARTMENTS.
3. Si el desencadenador funcionó, verificamos en la tabla LOGDEPARTAMENTO.

```
MariaDB [HR]> DELIMITER ;
MariaDB [HR]>
MariaDB [HR]>
MariaDB [HR]> UPDATE DEPARTMENTS SET DEPARTMENT_NAME='ASUNTOS CULINARIOS' WHERE DEPARTMENT_ID=280; 1
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [HR]> SELECT * FROM DEPARTMENTS WHERE DEPARTMENT_ID=280; 2
+---------------+--------------------+------------+-------------+
| department_id | department_name    | manager_id | location_id |
+---------------+--------------------+------------+-------------+
|           280 | ASUNTOS CULINARIOS |       NULL |        NULL |
+---------------+--------------------+------------+-------------+
1 row in set (0.00 sec)

MariaDB [HR]> SELECT * FROM LOGDEPARTAMENTO; 3
+------+-----------+---------------------+---------------+------------------------+------------------------+
| ID   | OPERACION | FECHA               | USUARIO       | VALORN                 | VALORV                 |
+------+-----------+---------------------+---------------+------------------------+------------------------+
|  280 | I         | 2020-09-04 07:52:04 | root@localhost | ASUNTOS SIN IMPORTANCIA | NULL                   |
|  280 | A         | 2020-09-04 08:01:04 | root@localhost | ASUNTOS CULINARIOS      | ASUNTOS SIN IMPORTANCIA |
+------+-----------+---------------------+---------------+------------------------+------------------------+
2 rows in set (0.00 sec)
```

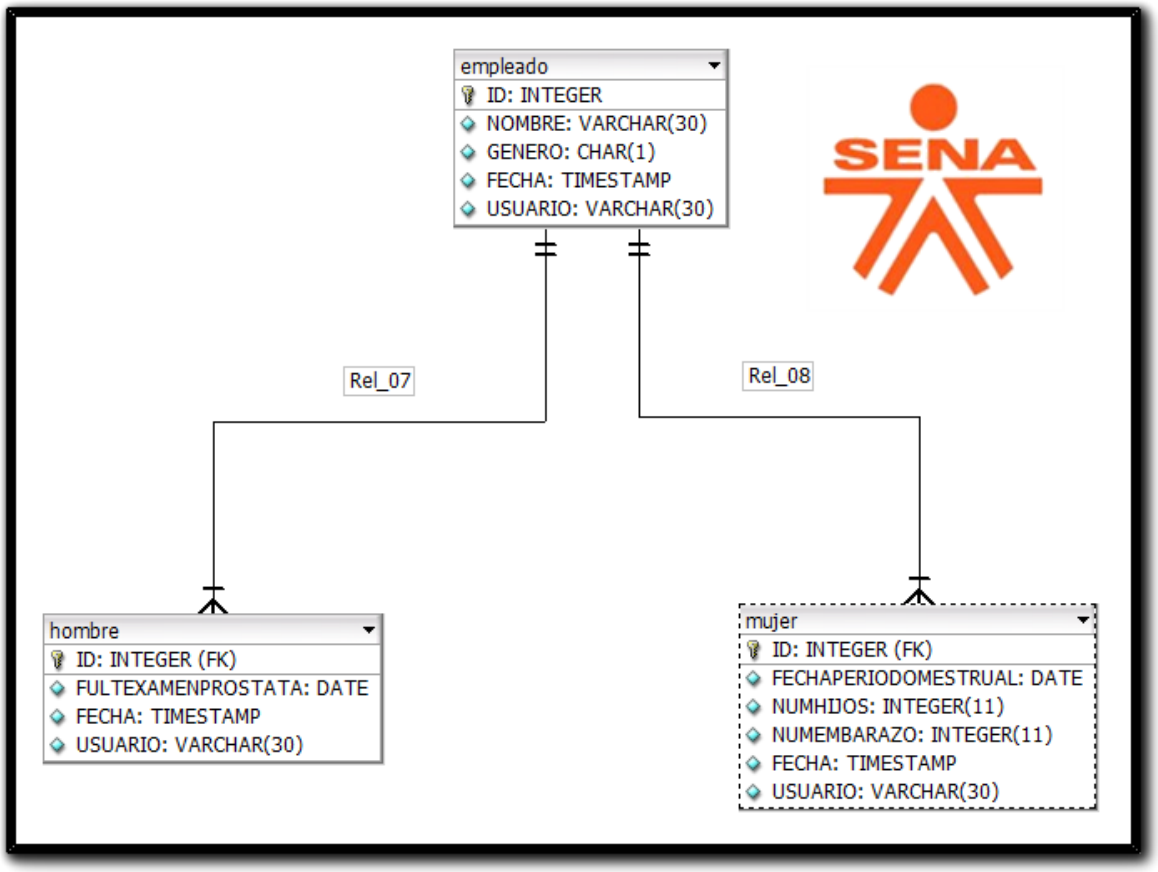
- Crear el TRIGGER para el evento DELETE

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

MariaDB [HR]> DELIMITER ;
```

1. Borramos en la tabla DEPARTMENTS el registro 280.
2. Verificamos que se borró en la tabla DEPARTMENTS.
3. Si el desencadenador funcionó, verificamos en la tabla LOGDEPARTAMENTO.

```
MariaDB [HR]> DELETE FROM DEPARTMENTS WHERE DEPARTMENT_ID=280; 1
Query OK, 1 row affected (0.02 sec)

MariaDB [HR]> SELECT * FROM DEPARTMENTS WHERE DEPARTMENT_ID=280; 2
Empty set (0.00 sec)

MariaDB [HR]> SELECT * FROM LOGDEPARTAMENTO; 3
+------+-----------+---------------------+---------------+------------------------+------------------------+
| ID   | OPERACION | FECHA               | USUARIO       | VALORN                 | VALORV                 |
+------+-----------+---------------------+---------------+------------------------+------------------------+
|  280 | I         | 2020-09-04 07:52:04 | root@localhost | ASUNTOS SIN IMPORTANCIA | NULL                   |
|  280 | A         | 2020-09-04 08:01:04 | root@localhost | ASUNTOS CULINARIOS      | ASUNTOS SIN IMPORTANCIA |
|  280 | B         | 2020-09-04 08:07:59 | root@localhost | NULL                   | ASUNTOS CULINARIOS      |
+------+-----------+---------------------+---------------+------------------------+------------------------+
3 rows in set (0.00 sec)
```

Otro ejemplo:



- Creamos la tabla EMPLEADO

```
MariaDB [HR]> CREATE TABLE EMPLEADO(
    -> ID INTEGER PRIMARY KEY AUTO_INCREMENT,
    -> NOMBRE VARCHAR(30),
    -> GENERO CHAR(1),
    -> FECHA TIMESTAMP,
    -> USUARIO VARCHAR(30)
    -> );
Query OK, 0 rows affected (0.03 sec)

MariaDB [HR]>
MariaDB [HR]> DESC EMPLEADO;
+---------+-------------+------+-----+-------------------+-----------------------------+
| Field   | Type        | Null | Key | Default           | Extra                       |
+---------+-------------+------+-----+-------------------+-----------------------------+
| ID      | int(11)     | NO   | PRI | NULL              | auto_increment              |
| NOMBRE  | varchar(30) | YES  |     | NULL              |                             |
| GENERO  | char(1)     | YES  |     | NULL              |                             |
| FECHA   | timestamp   | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
| USUARIO | varchar(30) | YES  |     | NULL              |                             |
+---------+-------------+------+-----+-------------------+-----------------------------+
5 rows in set (0.02 sec)
```

- Creamos la tabla MUJER

```
MariaDB [HR]> CREATE TABLE MUJER(
    -> ID INTEGER PRIMARY KEY,
    -> FECHAPERIODOMESTRUAL DATE,
    -> NUMHIJOS INTEGER,
    -> NUMEMBARAZO INTEGER,
    -> FECHA TIMESTAMP,
    -> USUARIO VARCHAR(30)
    -> );
ERROR 1050 (42S01): Table 'mujer' already exists
MariaDB [HR]> DESC MUJER;
+----------------------+-------------+------+-----+-------------------+-----------------------------+
| Field                | Type        | Null | Key | Default           | Extra                       |
+----------------------+-------------+------+-----+-------------------+-----------------------------+
| ID                   | int(11)     | NO   | PRI | NULL              |                             |
| FECHAPERIODOMESTRUAL | date        | YES  |     | NULL              |                             |
| NUMHIJOS             | int(11)     | YES  |     | NULL              |                             |
| NUMEMBARAZO          | int(11)     | YES  |     | NULL              |                             |
| FECHA                | timestamp   | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
| USUARIO              | varchar(30) | YES  |     | NULL              |                             |
+----------------------+-------------+------+-----+-------------------+-----------------------------+
6 rows in set (0.01 sec)
```

- Creamos la tabla HOMBRE

```
MariaDB [HR]> CREATE TABLE HOMBRE(
    -> ID INTEGER PRIMARY KEY,
    -> FULTEXAMENPROSTATA DATE,
    -> FECHA TIMESTAMP,
    -> USUARIO VARCHAR(30));
Query OK, 0 rows affected (0.03 sec)

MariaDB [HR]> DESC HOMBRE;
+--------------------+-------------+------+-----+-------------------+-----------------------------+
| Field              | Type        | Null | Key | Default           | Extra                       |
+--------------------+-------------+------+-----+-------------------+-----------------------------+
| ID                 | int(11)     | NO   | PRI | NULL              |                             |
| FULTEXAMENPROSTATA | date        | YES  |     | NULL              |                             |
| FECHA              | timestamp   | NO   |     | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
| USUARIO            | varchar(30) | YES  |     | NULL              |                             |
+--------------------+-------------+------+-----+-------------------+-----------------------------+
4 rows in set (0.01 sec)
```

- Creamos el TRIGGER “TREMPLEADOI”

```
MariaDB [HR]> DELIMITER $
MariaDB [HR]> CREATE OR REPLACE TRIGGER TREMPLEADOI AFTER INSERT ON EMPLEADO FOR EACH ROW
    -> BEGIN
    ->     IF( NEW.GENERO ='F') THEN
    ->     INSERT INTO MUJER(ID,USUARIO) VALUES(NEW.ID,USER());
    ->     END IF;
    ->     IF( NEW.GENERO ='M') THEN
    ->     INSERT INTO HOMBRE(ID,USUARIO) VALUES(NEW.ID,USER());
    ->     END IF;
    -> END;
    -> $
Query OK, 0 rows affected (0.03 sec)
```

- Insertamos el registro de un empleado que es mujer.

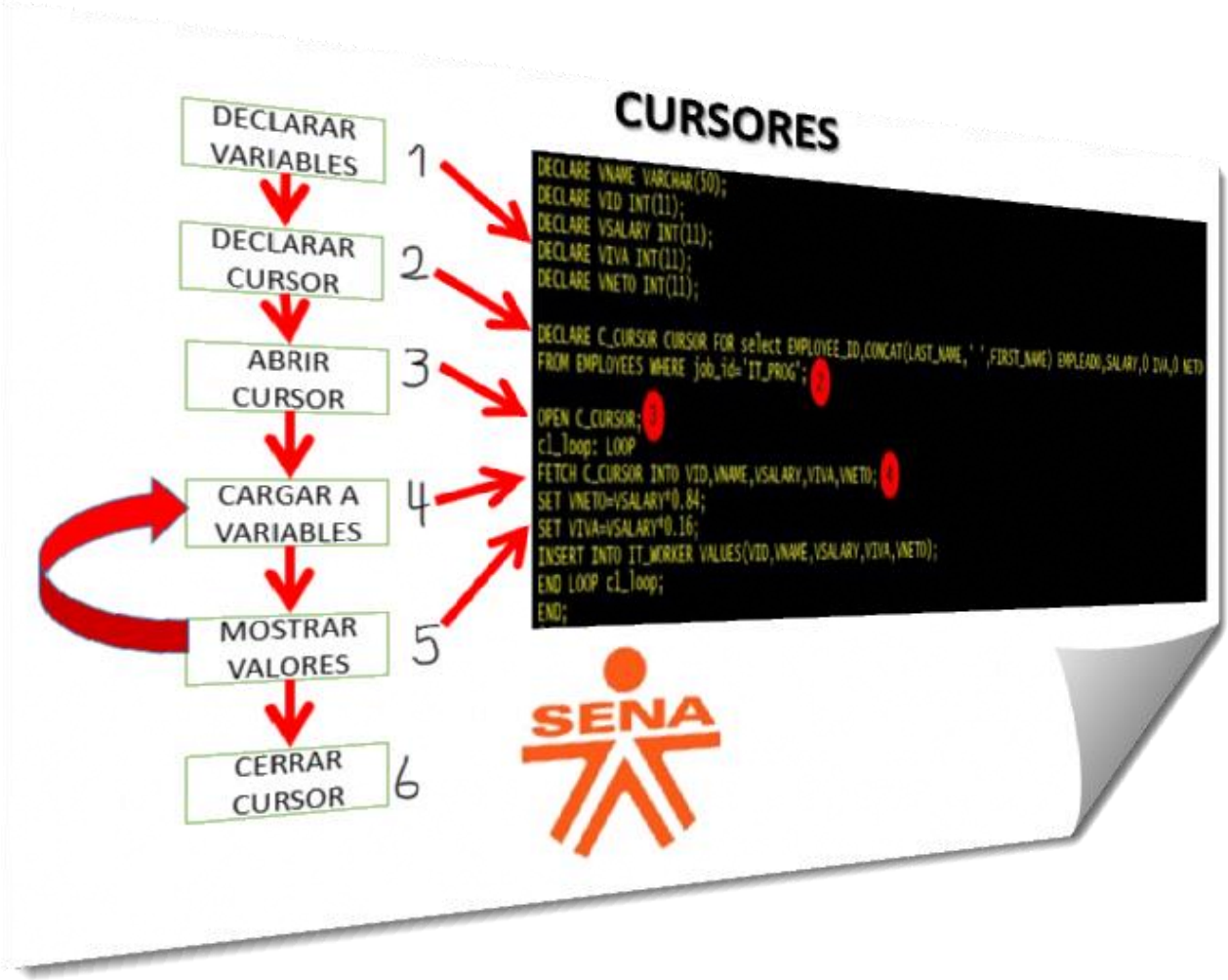
```
MariaDB [HR]> INSERT INTO EMPLEADO(NOMBRE,GENERO) VALUES('ZOILA VACA','F'); 1
Query OK, 1 row affected (0.01 sec)

MariaDB [HR]> SELECT * FROM EMPLEADO; 2
+----+------------+--------+---------------------+---------+
| ID | NOMBRE     | GENERO | FECHA               | USUARIO |
+----+------------+--------+---------------------+---------+
|  1 | ZOILA VACA | F      | 2020-09-04 09:36:26 | NULL    |
+----+------------+--------+---------------------+---------+
1 row in set (0.00 sec)

MariaDB [HR]> SELECT * FROM MUJER; 3
+----+----------------------+----------+-------------+---------------------+----------------+
| ID | FECHAPERIODOMESTRUAL | NUMHIJOS | NUMEMBARAZO | FECHA               | USUARIO        |
+----+----------------------+----------+-------------+---------------------+----------------+
|  1 | NULL                 |     NULL |        NULL | 2020-09-04 09:36:26 | root@localhost |
+----+----------------------+----------+-------------+---------------------+----------------+
1 row in set (0.00 sec)

MariaDB [HR]> SELECT * FROM HOMBRE; 4
Empty set (0.00 sec)
```

- Insertamos el registro de un empleado que es hombre.

```
MariaDB [HR]> INSERT INTO EMPLEADO(NOMBRE,GENERO) VALUES('SIMON TOLOMEO','M'); 1
Query OK, 1 row affected (0.01 sec)

MariaDB [HR]> SELECT * FROM EMPLEADO; 2
+----+---------------+--------+---------------------+---------+
| ID | NOMBRE        | GENERO | FECHA               | USUARIO |
+----+---------------+--------+---------------------+---------+
|  1 | ZOILA VACA    | F      | 2020-09-04 09:36:26 | NULL    |
|  2 | SIMON TOLOMEO | M      | 2020-09-04 09:40:24 | NULL    |
+----+---------------+--------+---------------------+---------+
2 rows in set (0.00 sec)

MariaDB [HR]> SELECT * FROM MUJER;
+----+----------------------+----------+-------------+---------------------+----------------+
| ID | FECHAPERIODOMESTRUAL | NUMHIJOS | NUMEMBARAZO | FECHA               | USUARIO        |
+----+----------------------+----------+-------------+---------------------+----------------+
|  1 | NULL                 |     NULL |        NULL | 2020-09-04 09:36:26 | root@localhost |
+----+----------------------+----------+-------------+---------------------+----------------+
1 row in set (0.00 sec)

MariaDB [HR]> SELECT * FROM HOMBRE; 3
+----+--------------------+---------------------+----------------+
| ID | FULTEXAMENPROSTATA | FECHA               | USUARIO        |
+----+--------------------+---------------------+----------------+
|  2 | NULL               | 2020-09-04 09:40:24 | root@localhost |
+----+--------------------+---------------------+----------------+
1 row in set (0.00 sec)
```

CURSORES

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



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()
    -> 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';
    ->
    -> OPEN C_CURSOR;
    -> c1_loop: LOOP
    -> FETCH C_CURSOR INTO VID,VNAME,VSALARY,VIVA,VNETO;
    -> 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();
ERROR 1329 (02000): No data - zero rows fetched, selected, or processed
MariaDB [HR]> SELECT * FROM IT_WORKER;
+-------------+------------------+---------+------+------+
| 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)
```

RETO POR 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 guardar el id, el salario antiguo, el nuevo salario, el cargo, la fecha y el usuario que realizo la transacción en la tabla LOGSALARIO.

Realizado por el instructor José Fernando Galindo Suárez
jgalindos@sena.edu.co 2020