**Actividad de los Triggers**

Erick Llanos Ríos | 1º DAW

**PARTE 0.**

Crea una base de datos llamada test que contenga una tabla llamada alumnos con las siguientes columnas.

Tabla alumnos:

* id (entero sin signo)
* nombre (cadena de caracteres)
* apellido1 (cadena de caracteres)
* apellido2 (cadena de caracteres)
* email (cadena de caracteres)

Inserta unos 10 registros en esa tabla.

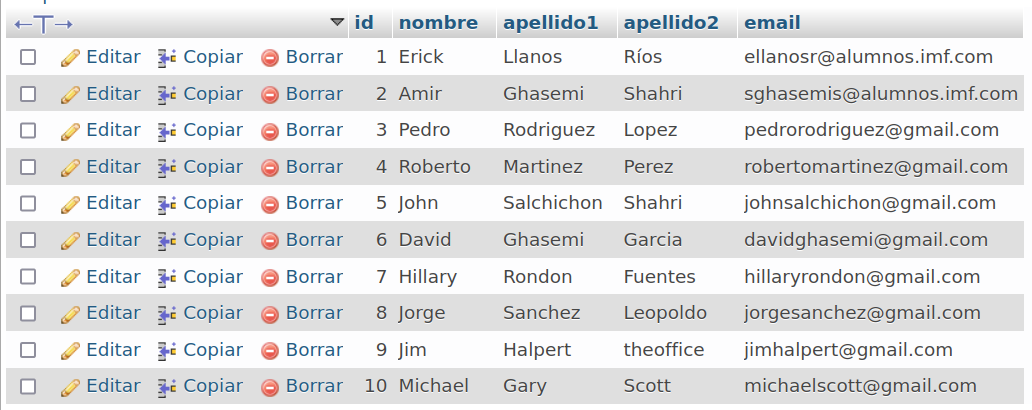
CREATE DATABASE test;

USE test;

CREATE TABLE alumnos (id INT UNSIGNED AUTO\_INCREMENT PRIMARY KEY, nombre VARCHAR (50) NOT NULL, apellido1 VARCHAR (50) NOT NULL, apellido2 VARCHAR (50), email VARCHAR(128));



INSERTAR 10 REGISTROS



**PARTE 1.**

1.- Escriba un procedimiento llamado **crear\_email** que dados los parámetros de entrada: nombre, apellido1, apellido2 y dominio, cree una dirección de email y la devuelva como salida.

Procedimiento: crear\_email

Entrada:

* nombre (cadena de caracteres)
* apellido1 (cadena de caracteres)
* apellido2 (cadena de caracteres)
* dominio (cadena de caracteres)

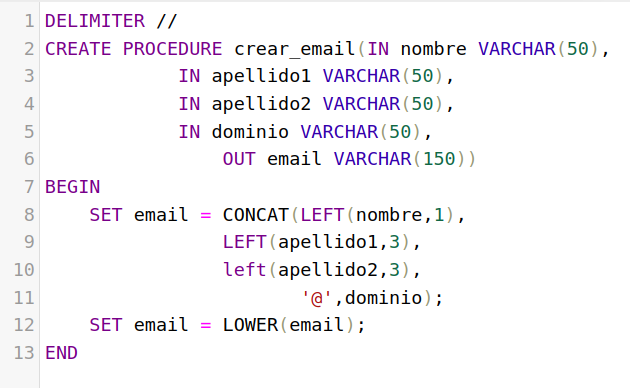
Salida:

* email (cadena de caracteres)

devuelva una dirección de correo electrónico con el siguiente formato:

* El primer carácter del parámetro nombre.
* Los tres primeros caracteres del parámetro apellido1.
* Los tres primeros caracteres del parámetro apellido2.
* El carácter @.
* El dominio pasado como parámetro.

Ejemplo: Entrada: María Arriaga Caballero, hotmail.com. Salida: marrcab@hotmail.com

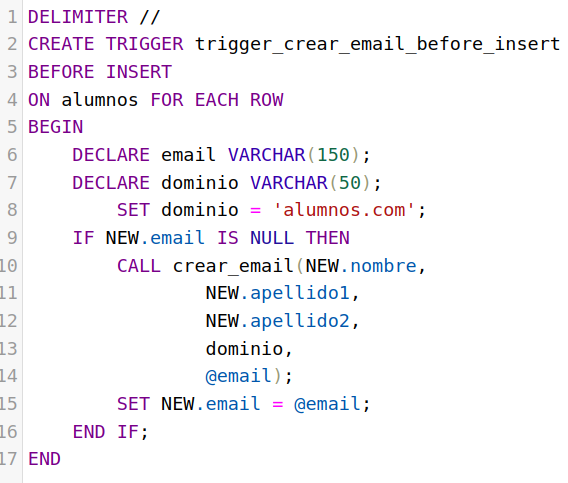


**PARTE 2.**

2.- Una vez creada la tabla escriba un trigger con las siguientes características:

* Trigger: trigger\_crear\_email\_before\_insert
* Se ejecuta sobre la tabla alumnos.
* Se ejecuta antes de una operación de inserción.
* Si el nuevo valor del email que se quiere insertar es NULL, entonces se le creará automáticamente una dirección de email y se insertará en la tabla.
* Si el nuevo valor del email no es NULL se guardará en la tabla el valor del email.

Nota: Para crear la nueva dirección de email se deberá hacer uso del procedimiento crear\_email.



**PARTE 3.**

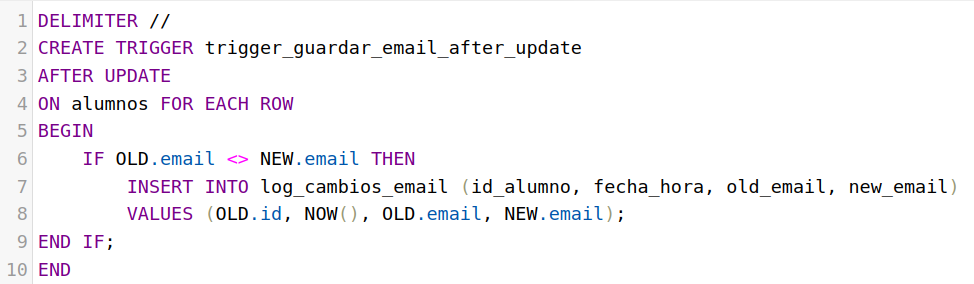
3.- Modifica el ejercicio anterior y añade un nuevo trigger que las siguientes características:

Trigger: trigger\_guardar\_email\_after\_update:

* Se ejecuta sobre la tabla alumnos.
* Se ejecuta después de una operación de actualización.
* Cada vez que un alumno modifique su dirección de email se deberá insertar un nuevo registro en una tabla llamada log\_cambios\_email.

La tabla log\_cambios\_email contiene los siguientes campos:

* id: clave primaria (entero autonumérico)
* id\_alumno: id del alumno (entero)
* fecha\_hora: marca de tiempo con el instante del cambio (fecha y hora)
* old\_email: valor anterior del email (cadena de caracteres)
* new\_email: nuevo valor con el que se ha actualizado



**PARTE 4**

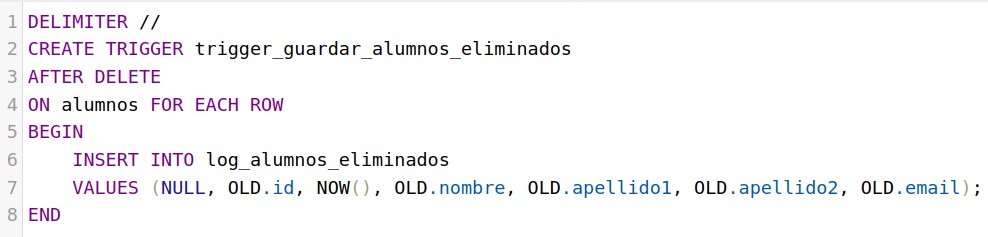
4.- Modifica el ejercicio anterior y añade un nuevo trigger que tenga las siguientes características:

Trigger: trigger\_guardar\_alumnos\_eliminados:

* Se ejecuta sobre la tabla alumnos.
* Se ejecuta después de una operación de borrado.
* Cada vez que se elimine un alumno de la tabla alumnos se deberá insertar un nuevo registro en una tabla llamada log\_alumnos\_eliminados.

La tabla log\_alumnos\_eliminados contiene los siguientes campos:

* id: clave primaria (entero autonumérico)
* id\_alumno: id del alumno (entero)
* fecha\_hora: marca de tiempo con el instante del cambio (fecha y hora)
* nombre: nombre del alumno eliminado (cadena de caracteres)
* apellido1: primer apellido del alumno eliminado (cadena de caracteres)
* apellido2: segundo apellido del alumno eliminado (cadena de caracteres)
* email: email del alumno eliminado (cadena de caracteres)



**RESULTADO FINAL**

