

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));
```

Examinar

Estructura

SQL

Buscar

Insertar

Exportar

Importar

Privilegios

Más

Estructura de tabla

Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 id	int(10)		UNSIGNED	No	Ninguna		AUTO_INCREMENT	Cambiar Elimina
<input type="checkbox"/>	2 nombre	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Elimina
<input type="checkbox"/>	3 apellido1	varchar(50)	latin1_swedish_ci		No	Ninguna			Cambiar Elimina
<input type="checkbox"/>	4 apellido2	varchar(50)	latin1_swedish_ci		Sí	NULL			Cambiar Elimina
<input type="checkbox"/>	5 email	varchar(128)	latin1_swedish_ci		Sí	NULL			Cambiar Elimina

INSERTAR 10 REGISTROS

				id	nombre	apellido1	apellido2	email
<input type="checkbox"/>	Editar	Copiar	Borrar	1	Erick	Llanos	Ríos	ellanosr@alumnos.imf.com
<input type="checkbox"/>	Editar	Copiar	Borrar	2	Amir	Ghasemi	Shahri	sghasemis@alumnos.imf.com
<input type="checkbox"/>	Editar	Copiar	Borrar	3	Pedro	Rodriguez	Lopez	pedrorodriguez@gmail.com
<input type="checkbox"/>	Editar	Copiar	Borrar	4	Roberto	Martinez	Perez	robertomartinez@gmail.com
<input type="checkbox"/>	Editar	Copiar	Borrar	5	John	Salchichon	Shahri	johnsalchichon@gmail.com
<input type="checkbox"/>	Editar	Copiar	Borrar	6	David	Ghasemi	Garcia	davidghasemi@gmail.com
<input type="checkbox"/>	Editar	Copiar	Borrar	7	Hillary	Rondon	Fuentes	hillaryrondon@gmail.com
<input type="checkbox"/>	Editar	Copiar	Borrar	8	Jorge	Sanchez	Leopoldo	jorgesanchez@gmail.com
<input type="checkbox"/>	Editar	Copiar	Borrar	9	Jim	Halpert	theoffice	jimhalpert@gmail.com
<input type="checkbox"/>	Editar	Copiar	Borrar	10	Michael	Gary	Scott	michaelscott@gmail.com

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

```
1 DELIMITER //
2 CREATE PROCEDURE crear_email(IN nombre VARCHAR(50),
3                               IN apellido1 VARCHAR(50),
4                               IN apellido2 VARCHAR(50),
5                               IN dominio VARCHAR(50),
6                               OUT email VARCHAR(150))
7 BEGIN
8     SET email = CONCAT(LEFT(nombre,1),
9                        LEFT(apellido1,3),
10                       left(apellido2,3),
11                        '@',dominio);
12     SET email = LOWER(email);
13 END
```

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.

```
1 DELIMITER //
```

```
2 CREATE TRIGGER trigger_crear_email_before_insert
```

```
3 BEFORE INSERT
```

```
4 ON alumnos FOR EACH ROW
```

```
5 BEGIN
```

```
6     DECLARE email VARCHAR(150);
```

```
7     DECLARE dominio VARCHAR(50);
```

```
8         SET dominio = 'alumnos.com';
```

```
9     IF NEW.email IS NULL THEN
```

```
10         CALL crear_email(NEW.nombre,
```

```
11                             NEW.apellido1,
```

```
12                             NEW.apellido2,
```

```
13                             dominio,
```

```
14                             @email);
```

```
15         SET NEW.email = @email;
```

```
16     END IF;
```

```
17 END
```

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

```
1 DELIMITER //
```

```
2 CREATE TRIGGER trigger_guardar_email_after_update
```

```
3 AFTER UPDATE
```

```
4 ON alumnos FOR EACH ROW
```

```
5 BEGIN
```

```
6     IF OLD.email <> NEW.email THEN
```

```
7         INSERT INTO log_cambios_email (id_alumno, fecha_hora, old_email, new_email)
```

```
8         VALUES (OLD.id, NOW(), OLD.email, NEW.email);
```

```
9     END IF;
```

```
10 END
```

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)

```
1 DELIMITER //
```

```
2 CREATE TRIGGER trigger_guardar_alumnos_eliminados
```

```
3 AFTER DELETE
```

```
4 ON alumnos FOR EACH ROW
```

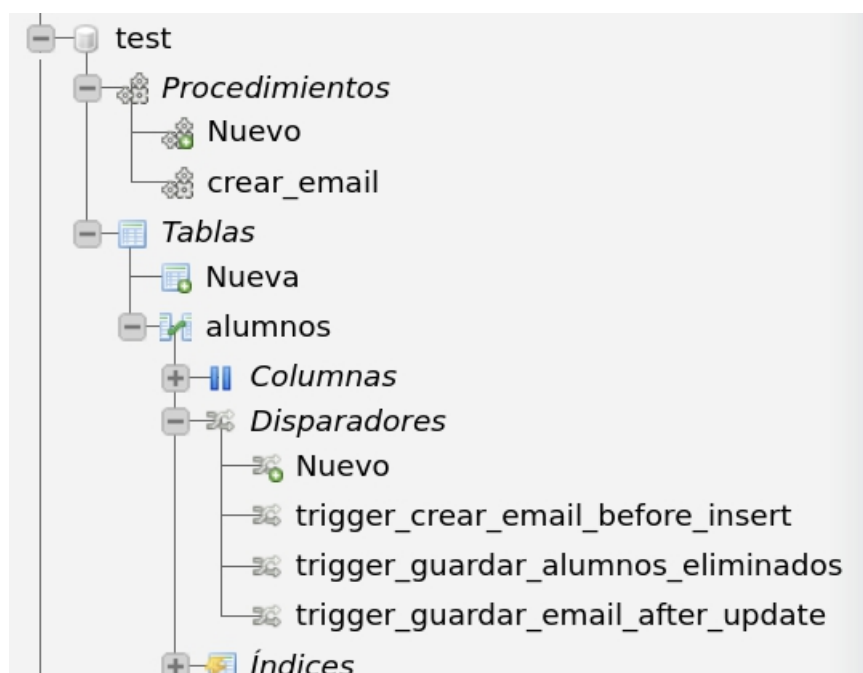
```
5 BEGIN
```

```
6     INSERT INTO log_alumnos_eliminados
```

```
7     VALUES (NULL, OLD.id, NOW(), OLD.nombre, OLD.apellido1, OLD.apellido2, OLD.email);
```

```
8 END
```

RESULTADO FINAL



✓ MySQL ha devuelto un conjunto de valores vacío (es decir: cero columnas). (La consulta tardó 0.0008 segundos.)

```
call crear_email('pedro','jusan','tijeras','alumnos.com',@mail);
```

[[Editar en línea](#)] [[Editar](#)] [[Crear código PHP](#)]

✓ Mostrando filas 0 - 0 (total de 1, La consulta tardó 0.0006 segundos.)

```
SELECT @mail;
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo

Número de filas: 25 ▾

Filtrar filas:

+ Opciones

@mail

pjustij@alumnos.com