



**Escuela Superior
de Ingeniería y Tecnología**
Universidad de La Laguna

Informe Modelo Lógico Relacional (Disparadores): Viveros

Jaime Pablo Pérez Moro
(alu0101278919@ull.edu.es)

Saúl Pérez García
(alu0101129785@ull.edu.es)

Claudio Néstor Yanes Mesa
(alu0101229942@ull.edu.es)



1. Dada la base de dato de viveros, creamos un procedimiento: crear_email que devuelve una dirección de correo electrónico con el siguiente formato:

- Un conjunto de caracteres del nombre y/o apellidos
- El carácter @
- El dominio pasado como parámetro.

```
CREATE OR REPLACE FUNCTION crear_email() RETURNS trigger AS
$BODY$
BEGIN
    IF NEW.email IS NULL THEN
        NEW.email = lower(NEW.nombre || '@' || TG_ARGV[0]);
    ELSIF NEW.email !~* '^[A-Za-z0-9._%~]+@[A-Za-z0-9.-]+[.][A-Za-z]+$' THEN
        RAISE EXCEPTION 'Error: El email introducido no es valido.';
    END IF;
    RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql;
```

Una vez creado el procedimiento escribimos un disparador con las siguientes características:

- Disparador: trigger_crear_email_before_insert
 - Se ejecuta sobre la tabla clientes.
 - 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 atributo email no es NULL se comprobará que el valor se corresponde con un email.

```
CREATE TRIGGER trigger_crear_email_before_insert BEFORE INSERT ON clientes
FOR EACH ROW EXECUTE PROCEDURE crear_email('gmail.com');
```

2. Crear un disparador que permita verificar que dos viveros diferentes no pertenecen al mismo municipio.



```
CREATE OR REPLACE FUNCTION check_municipio() RETURNS trigger AS
$BODY$
BEGIN
    IF EXISTS (SELECT * FROM viveros WHERE NEW.municipio=municipio) THEN
        RAISE EXCEPTION 'Error: No pueden haber dos viveros en el mismo municipio';
    END IF;
    RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql;
```

Creamos una función que en caso de detectar alguna coincidencia en los municipios de la tabla viveros con el municipio de la nueva fila que queremos insertar nos devuelve un error.

```
CREATE TRIGGER trigger_check_municipio_before_insert BEFORE INSERT ON viveros
FOR EACH ROW EXECUTE PROCEDURE check_municipio();
```

Aquí se ve el trigger que llama a la función creada.

3. Crear el o los disparadores que permitan mantener actualizado el stock de la base de datos de viveros.

```
CREATE OR REPLACE FUNCTION actualizar_stock() RETURNS trigger AS
$BODY$
DECLARE
    new_stock INT;
BEGIN
    IF tg_op = 'UPDATE' THEN
        new_stock = NEW.stock - OLD.stock;
    ELSIF tg_op = 'INSERT' then
        new_stock = NEW.stock;
    END IF;
    UPDATE productos SET stock = stock + new_stock WHERE id = NEW.productos_id;
    RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql;
```

```
CREATE TRIGGER trigger_actualizar_stock_after_insert AFTER INSERT ON stock_en_zona
FOR EACH ROW EXECUTE PROCEDURE actualizar_stock();

CREATE TRIGGER trigger_actualizar_stock_after_update AFTER UPDATE ON stock_en_zona
FOR EACH ROW EXECUTE PROCEDURE actualizar_stock();
```

SELECTS de las tablas:



```
tajinaste=# SELECT * FROM viveros;
      nombre | latitud | longitud | direccion | municipio
-----+-----+-----+-----+-----
SantaCruzViveros.SA | 28.4672612 | -16.2544728 | C/ Castillo, 1 | Los Realejos
(1 row)
```

```
tajinaste=# SELECT * FROM zonas;
      nombre | viveros_latitud | viveros_longitud
-----+-----+-----
Zona1 | 28.4672612 | -16.2544728
(1 row)
```

```
tajinaste=# SELECT * FROM empleados;
  id | nombre
-----+-----
34254 | Pepe Geranios
(1 row)
```

```
tajinaste=# SELECT * FROM productos;
 id | nombre | stock
-----+-----+-----
 3 | Semillas | 0
 4 | Abono | 0
 1 | Macetas | 27
 2 | Rosas | 10
(4 rows)
```

```
tajinaste=# SELECT * FROM stock_en_zona;
 productos_id | zonas_nombre | zonas_viveros_latitud | zonas_viveros_longitud | stock
-----+-----+-----+-----+-----
1 | Zona1 | 28.4672612 | -16.2544728 | 27
2 | Zona1 | 28.4672612 | -16.2544728 | 10
(2 rows)
```

```
tajinaste=# SELECT * FROM clientes;
 dni | email | nombre | telefono
-----+-----+-----+-----
46928192 | fulanito@gmail.com | Fulanito | 622843726
46928195 | email@example.tld | Pepe | 622843732
(2 rows)
```



```
tajinaste=# SELECT * FROM pedidos;
 id |          fecha          | empleados_id | clientes_dni
-----+-----+-----+-----
  1 | 2021-12-04 00:00:00 |      34254 | 46928192
(1 row)
```

```
tajinaste=# SELECT * FROM productos_en_pedido;
 pedidos_id | productos_id | cantidad
-----+-----+-----
          1 |             1 |         3
(1 row)
```

```
tajinaste=# SELECT * FROM empleado_trabaja_en_zona;
 empleados_id | zonas_nombre | zonas_viveros_latitud | zonas_viveros_longitud | fecha_inicio | fecha_final
-----+-----+-----+-----+-----+-----
      34254 | Zona1       |      28.4672612       |      -16.2544728       | 2021-03-04   |
(1 row)
```