

## Triggers

1. Crea una **base de datos** llamada `clases` 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)
- `nota` (número real)

Una vez creada la tabla escriba **dos triggers** con las siguientes características:

- Trigger 1: `trigger_check_nota_before_insert`
  - Se ejecuta sobre la tabla `alumnos`.
  - Se ejecuta *antes* de una operación de *inserción*.
  - Si el nuevo valor de la nota que se quiere insertar es negativo, se guarda como 0.
  - Si el nuevo valor de la nota que se quiere insertar es mayor que 10, se guarda como 10.
- Trigger2 : `trigger_check_nota_before_update`
  - Se ejecuta sobre la tabla `alumnos`.
  - Se ejecuta *antes* de una operación de *actualización*.
  - Si el nuevo valor de la nota que se quiere actualizar es negativo, se guarda como 0.
  - Si el nuevo valor de la nota que se quiere actualizar es mayor que 10, se guarda como 10.

Una vez creados los triggers escriba varias sentencias de inserción y actualización sobre la tabla `alumnos` y verifica que los *triggers* se están ejecutando correctamente.

2. Crea una **base de datos** llamada `clases` que contenga una **tabla** llamada `alumnos` con las siguientes co-lumnas.

Tabla `alumnos`:

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

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.

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`.

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

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)

5. Haz un disparador que cree un registro en la tabla `nros` de la base ebanca con los campos cliente, cuenta, fecha y saldo cada vez que algún cliente se quede en números rojos en alguna de sus cuentas

6. Crea un disparador que cada vez que se borre una noticia de la base de datos nmotor, registre en la tabla log\_borrados el título de la noticia, el usuario y la fecha y hora.
7. Realiza un procedimiento en la base de datos nmotor, que inserte un nuevo campo en la tabla autores denominado numnews, que indicará el número de noticias publicada por el autor. Crea un disparador que cada vez que se inserte una noticia nueva, actualice este nuevo campo creado.