**UNIVERSIDAD TECNOLÓGICA NACIONAL**
Carrera: Tecnicatura Universitaria en Programación
2do cuatrimestre 2025
Materia: Base de Datos I
Profesor: Gustavo Ramoscelli
Ayudantes:
- Maria Victoria Ruiz
- Fernando Damián Ene

# CheatSheet - Procedimientos almacenados, Triggers y Cursores

## 1. Procedimientos almacenados

```
DROP PROCEDURE IF EXISTS tr_nombre_proc_almc;
DELIMITER $$
CREATE PROCEDURE pa_nombre_proc_almc (IN var1 TIPO, OUT var2 TIPO)
BEGIN

    1 .
    << consulta >>

END$$
DELIMITER ;
```

---

‼Notas:

1. Utilizamos DECLARE para establecer qué variables vamos a utilizar, esto es después del begin pero antes de las consultas.
   *Ejemplo: DECLARE v_nombre VARCHAR(20);*

A tener en cuenta dentro de la consulta:

- @ son variables de sesión, no es necesario declararlas, solo establecer el '@'. Ejemplo: @nombre_var;
- Utilizamos SET nombre_var = valor para asignar un valor a una variable.
  *Ejemplo: SET v_nombre = 'Victoria';*
- Se puede usar un SELECT INTO para que el resultado de una consulta se guarde en una variable. Esto se suele utilizar más que nada en cursores. La sintaxis es SELECT consulta INTO variable.
  *Ejemplo: SELECT nombre FROM Clientes INTO v_nombre;*

## 2. Triggers

```
DROP TRIGGER IF EXISTS tr_nombre_trigger;
DELIMITER $$
CREATE TRIGGER tr_nombre_trigger
 2 . 3 . ON NombreTabla
FOR EACH ROW
BEGIN

    4. << consulta >>

END$$
DELIMITER ;
```

---

### !!Notas:

2. Hay dos palabras reservadas para esta parte: AFTER y BEFORE. La primera es para luego de que suceda algo, y la segunda para el después, por ejemplo, después de actualizar una tabla o antes de borrar un valor de una tabla.
    a. Todo trigger que tenga un 'AFTER' se va a manejar con datos de solo lectura, ya que no se puede modificar algo ya ingresado en una tabla.
    b. Lo mismo ocurre cuando utilizamos 'OLD', es algo que ya estaba en la tabla.
3. Con las palabras reservadas de esta parte determinamos el qué del anterior punto. Se puede utilizar: DELETE, UPDATE o INSERT.

### A tener en cuenta dentro de la consulta:

4. En caso que se utilice un INSERT o UPDATE, se utiliza la palabra reservada **'NEW'** como prefijo para referirnos a la información nueva.
    a. *Ejemplo: saldo = saldo + NEW.ingreso;*

    En caso que se utilice un DELETE o UPDATE, se utiliza la palabra reservada **'OLD'** como prefijo para referirnos a la información que fue borrada o al valor anterior.
    b. *Ejemplo: INSERT INTO clientes (id, nombre, mail) VALUES (OLD.id, OLD.nombre, OLD.mail);*

## 3. Cursores

*DROP TRIGGER IF EXISTS cur_nombre_cursor;*

```
DELIMITER $$
CREATE PROCEDURE cur_nombre_cursor (IN var1 TIPO, OUT var2 TIPO)
BEGIN
```

1. Declaramos las variables (siempre va a estar la variable 'fin'):
   ```
   DECLARE fin BOOLEAN DEFAULT FALSE;
   5. << otras variables >>
   ```

2. Declaramos el cursor y establecemos en base a qué consulta se va a cargar de datos:
   ```
   DECLARE nombre_cursor CURSOR for << consulta >>
   ```

3. Declaramos el HANDLER, que determina cuando no haya más filas en el cursor, siga ejecutando los comandos siguientes y evite un error, además de que fin pasa de FALSE a TRUE:
   ```
   DECLARE CONTINUE HANDLER FOR NOT FOUND SET fin = TRUE;
   ```

4. Abrimos el cursor para poder establecer qué hacemos con los datos:
   ```
   OPEN nombre_cursor;
   ```

5. Utilizamos un loop para iterar por cada fila de los datos guardados:
   ```
   loop_nombre : LOOP
   ```

6. Agarramos la información del cursor dentro de las variables que establecimos al principio. De esta forma, podemos modificar valores:
   ```
   FETCH nombre_cusor INTO 6. << variables >>
   ```

7. Nos aseguramos de que cuando el handler actúe y el cursor no tenga más información, salgamos del loop:
   ```
   7. IF fin THEN
       LEAVE loop_nombre;
   END IF;

   << consultas >>
   ```

8. Cerramos el loop y el cursor:
   ```
   END loop_nombre;
   CLOSE nombre_cursor;
   ```

```
END$$
DELIMITER ;
```

---

‼Notas:

5. Para los nombres de las variables, les recomiendo establecer un 'v_' en el nombre. *Ejemplo: DECLARE v_nombre VARCHAR(20);*
    a. Por cada columna de la consulta que realizamos en el cursor, vamos a tener una variable asociada.
    b. Por ejemplo, si realizamos esto: DELCARE cur CURSOR FOR SELECT nombre, apellido FROM Empleados
    c. Tendríamos que haber declarado 2 variables al principio:
    DECLARE v_nombre VARCHAR(20);
    DECLARE v_apellido VARCHAR(15);
6. Continuando con los ejemplos anteriores, en esta parte tenemos que poner en el mismo orden del SELECT las variables que acabamos de declarar.
    a. Ejemplo:
    FETCH cur INTO v_nombre, v_apellido;
    b. Si lo hacemos de forma inversa, se va a guardar la información del nombre en el apellido, y la del apellido en el nombre; en el caso de fueran dos variables de dos tipos diferentes, tendríamos un error.
    Ejemplo erróneo:
    FETCH cur INTO v_apellido, v_nombre;
7. No es necesario escribir IF fin = TRUE THEN [...], ya que si no establecemos una igualdad, se toma como que el valor que va a tener la variable es TRUE. Entonces, IF fin THEN [...] es lo mismo que IF fin = TRUE THEN [...] .