



1. Condicionales
2. Bucles

Procedimientos.
Condicionales y Bucles

Condicionales

Los comandos condicionales nos permiten poder elegir ejecutar un código dependiendo de que se cumpla una condición o no.

Dentro de este grupo de comandos nos centraremos en dos tipos de condicionales:

IF

CASE

IF

El comando **IF** es el constructor condicional básico.

Sintaxis:

```
IF condicion THEN sentencias_a_ejecutar;  
    [ELSEIF condicion THEN sentencias_a_ejecutar;] ...  
    [ELSE sentencias_a_ejecutar;]  
END IF;
```

IF

- Ejemplo:

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS calculo $$  
CREATE PROCEDURE calculo(IN param1 INTEGER)  
BEGIN  
    DECLARE var1 CHAR(10);  
    //contenido del IF  
END $$  
DELIMITER ;
```

IF

- Ejemplo:

```
IF param1=1 THEN  
    SELECT 'opcion_if';  
    SET var1 = 'opcion_if';  
ELSE  
    SELECT 'opcion_else';  
    SET var1 = 'opcion_else';  
END IF;  
INSERT INTO table1 VALUES (var1);
```

CASE

El comando **CASE** es el constructor condicional complejo.

Sintaxis:

```
CASE variable_de_condicion  
    WHEN condicion THEN sentencias_a_ejecutar;  
    [WHEN condicion THEN sentencias_a_ejecutar;] ...  
    [ELSE sentencias_a_ejecutar;]  
END CASE;
```

CASE

- Ejemplo:

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS calculo2 $$  
CREATE PROCEDURE calculo2(IN param1 INTEGER)  
BEGIN  
    //contenido del CASE  
END $$  
DELIMITER ;
```


CASE

- Ejemplo:

```
CASE param1
  WHEN 1 THEN
    SELECT 'opcion_1';
  WHEN 2 THEN
    SELECT 'opcion_2';
  WHEN 3 THEN
    SELECT 'opcion_3';
  ELSE SELECT 'ninguna_opcion';
END CASE;
```

Bucles

Los comandos de bucles nos permiten poder repetir la ejecución de un código tantas veces como nos sea necesario según que condiciones definamos.

Dentro de este grupo de comandos nos centraremos en dos tipos de bucles:

WHILE

LOOP

WHILE

El comando **WHILE** es el constructor de bucles básico.

Sintaxis:

```
[etiqueta :] WHILE condicion DO  
    sentencias_a_ejecutar;  
END WHILE [etiqueta] ;
```

WHILE

- Ejemplo:

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS calculo3 $$  
CREATE PROCEDURE calculo3()  
BEGIN  
    DECLARE var1 INTEGER DEFAULT 5;  
    //contenido del WHILE  
END $$  
DELIMITER ;
```

WHILE

- Ejemplo:

```
WHILE var1>0 DO  
    SELECT var1;  
    SET var1=var1-1;  
END WHILE;
```

LOOP

El comando **LOOP** es el constructor de bucles complejo.

Sintaxis:

```
etiqueta: LOOP  
    //posible cuerpo del loop-1  
    IF done=1 THEN  
        LEAVE etiqueta;  
    END IF;  
    //posible cuerpo del loop-2  
END LOOP etiqueta;
```

LOOP

- Ejemplo:

```
DELIMITER $$  
DROP PROCEDURE IF EXISTS calculo4 $$  
CREATE PROCEDURE calculo4()  
BEGIN  
    DECLARE incremento;  
    SET incremento = 0;  
    //contenido del LOOP  
END $$  
DELIMITER ;
```


LOOP

- Ejemplo:

```
label4: LOOP  
    SET incremento = incremento + 1;  
    IF incremento > 10 THEN  
        LEAVE label4;  
    END IF;  
END LOOP label4;
```

“Lo más atroz de las cosas malas de la gente mala
es el silencio de la gente buena”

MAHATMA GANDHI

