

Bases de Datos

Unidad 8:
Programación de bases de datos
Sesión 4

Instrucciones de control de flujo

De control de flujo de bucle o iterativas

☐ **LOOP**

☐ **WHILE**

☐ **REPEAT**

bucle LOOP

LOOP no tiene ninguna condición de salida. Si se quiere salir de un bucle LOOP, hay que usar dentro de él una instrucción **LEAVE**. La sintaxis para la instrucción LOOP es:

```
[etiqueta:]LOOP  
    instrucciones;  
END LOOP [etiqueta];
```

La etiqueta es una marca que sirve para que se pueda saltar al comienzo o al final del bucle con las instrucciones LEAVE o ITERATE. La etiqueta que hay al principio y al final del bucle debe ser la misma.

NO ES RECOMENDABLE USAR LOOP. SE DEBEN USAR LOS BUCLES WHILE O REPEAT.

3.- Desarrollo de procedimientos almacenados

Ejemplo 8: Realiza un procedimiento para obtener cuantos divisores tiene un número entero.

```
CREATE PROCEDURE ejemplo8 (IN num INT, OUT c INT)
BEGIN
  DECLARE d INT;
  DECLARE n INT;
  SET c=0;
  SET n=num;
  IF n<0 THEN
    SET n=-n;
  END IF;
  SET d=n;
  etiq1: LOOP
    IF d=0 THEN
      LEAVE etiq1;
    END IF;
    IF n%d=0 THEN
      SET c=c+1;
    END IF;
    SET d=d-1;
  END LOOP etiq1;
END
```

¿Qué ocurrirá cuando haya que dividir entre 0?

¿Cuántos divisores tiene 5 según este procedimiento? ¿Es correcto?

3.- Desarrollo de procedimientos almacenados

Ejemplo 9: Realiza un procedimiento que obtiene el primer número de contrato a partir del contrato número 1 que no exista en la tabla contratos.

```
CREATE PROCEDURE ejemplo9 (OUT n INT)
BEGIN
  DECLARE cont INT;
  SET n=1;
  etiq1: LOOP
    SELECT count(*) INTO cont FROM contratos WHERE numcontrato=n;
    IF cont=0 THEN
      LEAVE etiq1;
    END IF;
    SET n=n+1;
  END LOOP etiq1;
END
```

¿Qué utilidad podría tener este procedimiento?

bucle REPEAT

Permite implementar una estructura repetitiva del tipo **repetir...hasta**. En esta estructura repetitiva se empieza ejecutando las instrucciones que están dentro de REPEAT y, al final, se analiza si se cumple la condición indicada en **UNTIL**. Si la condición es verdadera, se sale del bucle y, si es falsa, se vuelve al comienzo del bucle.

La sintaxis de REPEAT es:

REPEAT

instrucciones

UNTIL condicion

END REPEAT;

3.- Desarrollo de procedimientos almacenados

Ejemplo 10: Realiza un procedimiento para obtener cuantos divisores tiene un número entero.

```
CREATE PROCEDURE ejemplo10 (IN num INT, OUT c INT)
BEGIN
  DECLARE d INT;
  DECLARE n INT;
  DECLARE contador INT;
  SET contador=0;
  SET n=num;
  IF n<0 THEN
    SET n=-n;
  END IF;
  SET d=n;
  IF d>0 THEN
    REPEAT
      IF n%d=0 THEN
        SET contador=contador+1;
      END IF;
      SET d=d-1;
    UNTIL d=0 END REPEAT;
  END IF;
  SET c=contador;
END
```

3.- Desarrollo de procedimientos almacenados

Ejemplo 11: Realiza un procedimiento que crea una tabla con los nombres y apellidos de 10 clientes de la tabla clientes elegidos al azar y sin repetir.

```
CREATE PROCEDURE ejemplo11 ()
BEGIN
  DECLARE n INT default 0;
  DECLARE c INT;
  DECLARE nom VARCHAR(15);
  DECLARE ape VARCHAR(40);
  DROP TABLE IF EXISTS temporal;
  CREATE TABLE temporal ( nombre VARCHAR(25), apellidos VARCHAR(40));
  REPEAT
    SELECT nombre, apellidos INTO nom,ape FROM clientes ORDER BY rand() LIMIT 1;
    SELECT count(*) INTO c FROM temporal WHERE nombre=nom AND apellidos=ape;
    IF c=0 THEN
      INSERT INTO temporal VALUES (nom,ape);
    END IF;
    SET n=n+1;
  UNTIL n=10 END REPEAT;
END
```

¿Para qué se comprueba que c=0? ¿Qué error puedes identificar en el código?

bucle WHILE

En este bucle, se evalúa inicialmente una condición y, si esta se cumple, se ejecutan las instrucciones que hay dentro del bucle. Cuando se llega al final del bucle while (END WHILE) se vuelve al principio del bucle para evaluar la condición del WHILE, repitiéndose el proceso anterior si la condición se cumple.

Cuando la condición del WHILE no se cumpla, se produce la salida del bucle.

La sintaxis de WHILE es:

```
WHILE condicion DO  
    instrucciones  
END WHILE;
```

3.- Desarrollo de procedimientos almacenados

Ejemplo 12: Realiza un procedimiento para obtener cuantos divisores tiene un número entero.

```
CREATE PROCEDURE ejemplo12 (IN num INT, OUT c INT)
BEGIN
  DECLARE d INT;
  DECLARE n INT;
  SET c=0;
  SET n=num;
  IF n<0 THEN
    SET n=-n;
  END IF;
  SET d=n;
  WHILE d>0 DO
    IF n%d=0 THEN
      SET c=c+1;
    END IF;
    SET d=d-1;
  END WHILE;
END
```

3.- Desarrollo de procedimientos almacenados

Ejemplo 13: Realiza un procedimiento que crea una tabla en la base de datos ALQUILERES con los nombres y apellidos de tantas personas como se indique en un parámetro. Los nombres y apellidos se obtendrán al azar barajando los nombres y apellidos de todos los usuarios de la tabla usuarios de la base de datos CONCURSOMUSICA.

```
CREATE PROCEDURE ejemplo13 (IN numero INT)
BEGIN
    DECLARE c INT DEFAULT 0;
    DECLARE nom VARCHAR(15);
    DECLARE ape VARCHAR(40);
    DROP TABLE IF EXISTS temporal;
    CREATE TABLE temporal (
        nombre VARCHAR(25),
        apellidos VARCHAR(40));
    WHILE c<numero DO
        SELECT nombre INTO nom from concursomusica.usuarios ORDER BY rand() LIMIT 1;
        SELECT apellidos INTO ape FROM concursomusica.usuarios ORDER BY rand() LIMIT 1;
        SET c=c+1;
        INSERT INTO temporal VALUES (nom,ape);
    END WHILE;
END
```