



TEMA 6.

PROGRAMACIÓN EN BASES DE
DATOS.

SESIÓN 4



3.- Desarrollo de procedimientos almacenados

- Después de haber visto las estructuras condicionales, vamos a ver ahora las estructuras de control de flujo o repetitivas:
 - **Loop**
 - **While**
 - **Repeat**

3.- Desarrollo de procedimientos almacenados

▪ Bucle Loop

- LOOP es un bucle que 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 loop:** Realiza un procedimiento para insertar en una tabla números los números del 1 al 10. El procedimiento se hace en la base de datos prueba. Muestra la tabla al finalizar.

```
CREATE ejemplo8_loop()  
BEGIN  
declare i int;  
create table numeros(  
num int);  
set i=1;  
etiq1: loop  
    insert into numeros values (i);  
    set i=i+1;  
    if i>10 then  
        leave etiq1;  
    end if;  
end loop etiq1;  
select * from numeros;  
END
```

3.- Desarrollo de procedimientos almacenados

- **Ejemplo 9 loop:** Realiza un procedimiento que devuelve en una variable de texto los números de contrato pares de la base de datos alquileres.

```
CREATE PROCEDURE ejemplo9_loop(OUT texto
VARCHAR(100))
BEGIN
DECLARE ncontrato INT;
DECLARE i INT;
DECLARE aux INT;
SET i=1;
SET texto='resultado: ';
SELECT max(numcontrato) INTO ncontrato
FROM contratos;

etiq1: LOOP
    if i>cont THEN
        LEAVE etiq1;
    ELSEIF i%2=0 then
        SELECT count(*) INTO aux FROM
        contratos WHERE numcontrato=i;
        IF aux=1 THEN
            set texto=concat(texto,' ', i);
        END IF;
    END IF;
    SET i=i+1;
END LOOP etiq1;
END
```

3.- Desarrollo de procedimientos almacenados

- 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 8 repeat:** Realiza un procedimiento para insertar en una tabla números los números del 1 al 10. El procedimiento se hace en la base de datos prueba. Muestra la tabla al finalizar.

```
CREATE PROCEDURE ejemplo8_repeat()  
BEGIN  
  DECLARE i INT;  
  CREATE TABLE numeros(  
    num INT);  
  SET i=1;  
  REPEAT  
    INSERT INTO numeros VALUES (i);  
    SET i=i+1;  
  UNTIL i>10 END REPEAT;  
  SELECT * FROM numeros;  
END
```

3.- Desarrollo de procedimientos almacenados

- **Ejemplo 9 repeat:** Realiza un procedimiento que devuelve en una variable de texto los números de contrato pares de la base de datos alquileres.

```
CREATE PROCEDURE ejemplo9_repeat(OUT texto
VARCHAR(100))
BEGIN
DECLARE ncontratos INT;
DECLARE i INT;
DECLARE aux INT;
SET i=1;
SET texto='resultado: ';
SELECT MAX(numcontrato) INTO ncontratos FROM
contratos;
```

```
REPEAT
    IF i%2=0 THEN
        SELECT count(*) INTO aux FROM contratos
        WHERE numcontrato=i;
        IF aux=1 THEN
            set texto=concat(texto,' ', i);
        END IF;
    END IF;
    SET i=i+1;
UNTIL i>ncontratos END REPEAT;
END
```


3.- Desarrollo de procedimientos almacenados

- 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 8 while:** Realiza un procedimiento para insertar en una tabla números los números del 1 al 10. El procedimiento se hace en la base de datos prueba. Muestra la tabla al finalizar.

```
CREATE PROCEDURE ejemplo8_while()
BEGIN
DECLARE i INT;
CREATE TABLE numeros(
num INT);
SET i=1;
WHILE i<=10 DO
    INSERT INTO numeros VALUES (i);
    SET i=i+1;
END WHILE;
SELECT * FROM numeros;
END
```

3.- Desarrollo de procedimientos almacenados

- **Ejemplo 9 repeat:** Realiza un procedimiento que devuelve en una variable de texto los números de contrato pares de la base de datos alquileres.

```
CREATE PROCEDURE ejemplo9_repeat(OUT texto
VARCHAR(100))
BEGIN
DECLARE ncontratos INT;
DECLARE i INT;
DECLARE aux INT;
SET i=1;
SET texto='resultado: ';
SELECT MAX(numcontrato) INTO ncontratos FROM
contratos;
```

```
REPEAT
    IF i%2=0 THEN
        SELECT count(*) INTO aux FROM contratos
        WHERE numcontrato=i;
        IF aux=1 THEN
            set texto=concat(texto,' ', i);
        END IF;
    END IF;
    SET i=i+1;
UNTIL i>ncontratos END REPEAT;
END
```