



# TEMA 6.

PROGRAMACIÓN EN BASES DE  
DATOS.

SESIÓN 3



### 3.- Desarrollo de procedimientos almacenados

- Vamos a ver ahora instrucciones de control de flujo dentro de nuestras rutinas
- De decisión:
  - **If**
  - **Case**
- De control de flujo con bucles o repetitivas
  - **Loop**
  - **While**
  - **Repeat**

### 3.- Desarrollo de procedimientos almacenados

- Instrucciones de control de flujo - IF
- Si una condición se cumple, se realizan las instrucciones entre IF y ELSE o entre IF y END IF cuando no hay cláusula ELSE.
- Si no se cumple, se realizan las acciones bajo ELSE (si lo hay).
- Sintaxis:

```
IF condición THEN
    instruccion1;
    instruccion2;
    .....
ELSE
    instruccionA;
    instruccionB;
    .....
END IF;
```

```
IF condición THEN
    instruccion1;
    instruccion2;
    .....
END IF;
```

### 3.- Desarrollo de procedimientos almacenados

- **Ejemplo 1:** Realizar un procedimiento llamado par que recibe un número entero y escribe un texto “Es un número par” o “Es un número impar” según sea el número par o impar.
- **Ejemplo 2:** Realizar un procedimiento llamado es\_par que devuelve true si un número entero recibido en un parámetro es par y false si es impar.

# 3.- Desarrollo de procedimientos almacenados

- Ejemplo 1:

```
CREATE PROCEDURE par (IN numero INT)
BEGIN
  IF numero%2=0 THEN
    SELECT "Es un número par";
  ELSE
    SELECT "Es un número impar";
  END IF;
END
```

- Ejemplo 2:

```
CREATE PROCEDURE es_par (IN numero INT, OUT par BOOLEAN)
BEGIN
  IF numero%2=0 THEN
    SET par=true;
  ELSE
    SET par=false;
  END IF;
END
```

```
CREATE PROCEDURE es_par (IN numero INT, OUT par BOOLEAN)
BEGIN
  SET par=false;
  IF numero%2=0 THEN
    SET par=true;
  END IF;
END
```

### 3.- Desarrollo de procedimientos almacenados

- Instrucciones de control de flujo – IF y ELSEIF
- La cláusula ELSEIF dentro de un IF permite que se evalúe otra condición si no se cumple la condición IF u otra condición ELSEIF anterior.
- Sintaxis:

```
IF condición1 THEN
    instruccion1;
ELSEIF condición2 THEN
    instruccion2;
ELSEIF condición3 THEN
    instruccion3;
    .....
ELSE
    instrucciónN;
END IF;
```

### 3.- Desarrollo de procedimientos almacenados

- **Ejemplo 3A:** Realizar un procedimiento que recibe un número de día de semana laboral y devuelve el nombre de ese día de la semana. Hay que hacerlo con ELSEIF.
- **Ejemplo 3B:** Realizar un procedimiento que recibe un número de día de semana laboral y devuelve el nombre de ese día de la semana. Hay que hacerlo con IF anidados.

## 3.- Desarrollo de procedimientos almacenados

- Ejemplo 3A:

```
CREATE PROCEDURE ejemplo3A(IN numdia INT, OUT nomdia VARCHAR(15))
BEGIN
  IF numdia=1 THEN set nomdia='lunes';
    ELSEIF numdia=2 THEN SET nomdia='martes';
    ELSEIF numdia=3 THEN SET nomdia='miércoles';
    ELSEIF numdia=4 THEN SET nomdia='jueves';
    ELSEIF numdia=5 THEN SET nomdia='viernes';

  ELSE
    SET nomdia='dia incorrecto';
  END IF;
END
```



# 3.- Desarrollo de procedimientos almacenados

- Ejemplo 3B:

```
CREATE PROCEDURE ejemplo3(IN numdia INT, OUT nomdia VARCHAR(15))
BEGIN
  IF numdia=1 THEN
    SET nomdia='lunes';
  ELSE
    IF numdia=2 THEN
      SET nomdia='martes';
    ELSE
      IF numdia=3 THEN
        SET nomdia='miércoles';
      ELSE
        IF numdia=4 THEN
          SET nomdia='jueves';
        ELSE
          IF numdia=5 THEN
            SET nomdia='viernes';
          ELSE
            SET nomdia='dia incorrecto';
          END IF;
        END IF;
      END IF;
    END IF;
  END IF;
END IF;
END
```

### 3.- Desarrollo de procedimientos almacenados

- **Ejemplo 4:**
- Realizar un procedimiento que crea un nuevo contrato de alquiler para el coche de la matrícula que se pase como parámetro y para el cliente cuyo nombre y apellidos se pasen como parámetros.
- El procedimiento debe comprobar que el cliente y el coche existen y que el coche está disponible para alquilar.
- Si se puede crear el contrato se devuelve true en un parámetro, si no se puede crear el contrato, se devuelve false.

### 3.- Desarrollo de procedimientos almacenados

- Ejemplo 4:

```
CREATE PROCEDURE ejemplo4(IN mat CHAR(7), nom VARCHAR(15),ape VARCHAR(25),OUT hecho BOOLEAN)
BEGIN
    DECLARE na INT;
    DECLARE ncli INT;
    DECLARE kil INT;
    DECLARE d CHAR(9);
    SELECT count(*) INTO ncli FROM clientes WHERE nombre=nom AND apellidos=ape;
    SELECT count(*) INTO na FROM automoviles WHERE matricula=mat AND alquilado=false;
    IF na=1 AND ncli=1 THEN
        SELECT kilometros INTO kil FROM automoviles WHERE matricula=mat;
        SELECT dni INTO d FROM clientes WHERE nombre=nom AND apellidos=ape;
        INSERT INTO contratos(matricula,dnicliente,fini,kini) VALUES (mat,d,curdate(),kil);
        UPDATE automoviles SET alquilado=true WHERE matricula=mat;
        SET hecho=true;
    ELSE
        SET hecho=false;
    END IF;
END
```

### 3.- Desarrollo de procedimientos almacenados

- **Ejemplo 5:**
- Realizar un procedimiento que, partiendo de la matrícula de un coche, devuelve el texto 'A estrenar' cuando el coche tiene menos de 5000 Km, 'nuevo' cuando tiene entre 5000 y 25000, 'bastante rodado' cuando tiene entre 25000 y 100000 y 'muy rodado' en otro caso.
- Si no existiera coche con la matrícula pasada al procedimiento, se devolvería el texto 'No existe'.

### 3.- Desarrollo de procedimientos almacenados

- Ejemplo 5:

```
CREATE PROCEDURE ejemplo5 (IN mat CHAR(7), OUT estado TEXT)
BEGIN
    DECLARE km INT;
    DECLARE n INT DEFAULT 0;
    SET estado='No existe';
    SELECT count(*) INTO n FROM automoviles WHERE matricula=mat;
    IF n=1 THEN
        SELECT kilometros INTO km FROM automoviles WHERE matricula=mat;
        IF km<5000 THEN
            SET estado='A estrenar';
            ELSEIF km<25000 THEN
                SET estado='nuevo';
                ELSEIF km<100000 THEN
                    SET estado='bastante rodado';
                    ELSE
                        SET estado='muy rodado';
                    END IF;
        END IF;
    END IF;
END
```

### 3.- Desarrollo de procedimientos almacenados

- Instrucciones de control de flujo – CASE
- CASE es una estructura de decisión múltiple. Tiene dos sintaxis.
- **Sintaxis 1:** Se ejecutan las instrucciones correspondientes al primer valor que sea igual a la expresión. Cada uno de los valores posibles se evalúa con la cláusula WHEN. Si ninguno de los valores es igual a la expresión, se ejecutan las instrucciones que hay dentro de ELSE, caso de que hubiera ELSE.

```
CASE expresion
WHEN valor1 THEN instrucciones1
  [WHEN valor2 THEN instrucciones2]
.....
  [WHEN valorN THEN instruccionesN]
  [ELSE instrucciones_else]
END CASE;
```

## 3.- Desarrollo de procedimientos almacenados

- ***Ejemplo 6:***
- Realizar un procedimiento para obtener la fecha actual en formato:
- D de mes de AAAA (donde mes es el nombre del mes en español

### 3.- Desarrollo de procedimientos almacenados

- Ejemplo 6:

```
CREATE PROCEDURE ejemplo6 (OUT dia TEXT)
BEGIN
  DECLARE fecha DATE;
  DECLARE mes text;
  SET dia='';
  SELECT curdate() INTO fecha;
  SET dia=concat(dia,dayofmonth(fecha),' de ');
  CASE month(fecha)
    WHEN 1 THEN
      SET mes='enero';
    WHEN 2 THEN SET mes='febrero';
    WHEN 3 THEN SET mes='marzo';
    WHEN 4 THEN SET mes='abril';
```

```
    WHEN 5 THEN SET mes='mayo';
    WHEN 6 THEN SET mes='junio';
    WHEN 7 THEN SET mes='julio';
    WHEN 8 THEN SET mes='agosto';
    WHEN 9 THEN SET mes='septiembre';
    WHEN 10 THEN SET mes='octubre';
    WHEN 11 THEN SET mes='noviembre';
    ELSE
      SET mes='diciembre';
  END CASE;
  SET dia=concat(dia,mes,' de ',year(fecha));
END
```



### 3.- Desarrollo de procedimientos almacenados

- Instrucciones de control de flujo – CASE
- CASE es una estructura de decisión múltiple. Tiene dos sintaxis.
- **Sintaxis 2:** Se ejecutan las instrucciones correspondientes a la primera condición que se cumpla y si no se cumpliera ninguna de las condiciones, se ejecutarían las instrucciones que hay dentro del ELSE, caso de que haya ELSE.

```
CASE
WHEN condicion1 THEN instrucciones1
[WHEN condicion2 THEN instrucciones2]
.....
[WHEN condicionN THEN instruccionesN]
[ELSE instrucciones_else]
END CASE;
```

### 3.- Desarrollo de procedimientos almacenados

- **Ejemplo 7:**
- Realizar un procedimiento que, partiendo de la matrícula de coche, devuelve el texto 'A estrenar' cuando el coche tiene menos de 5000 Km, 'nuevo' cuando tiene entre 5000 y 25000, 'bastante rodado' cuando tiene entre 25000 y 100000 y 'muy rodado' en otro caso.
- Si no existiera coche con la matrícula pasada al procedimiento, se devolvería el texto 'No existe'.

# 3.- Desarrollo de procedimientos almacenados

- Ejemplo 7:

```
CREATE PROCEDURE ejemplo7 (IN mat CHAR(7), OUT estado TEXT)
BEGIN
  DECLARE km INT;
  DECLARE n INT DEFAULT 0;
  SET estado='No existe';
  SELECT count(*) INTO n FROM automoviles WHERE matricula=mat;
  IF n=1 THEN
    SELECT kilometros INTO km FROM automoviles WHERE matricula=mat;
    CASE
      WHEN km<5000 THEN
        SET estado='A estrenar';
      WHEN km<25000 THEN
        SET estado='nuevo';
      WHEN km<100000 THEN
        SET estado='bastante rodado';
      ELSE
        SET estado='muy rodado';
    END CASE;
  END IF;
END
```