

## Práctica 18

1.- Restaura la base de datos **ACME** a partir de la copia de seguridad

**BD\_ACME.sql**.

| Tabla                                  | Acción  | Filas | Tipo   | Cotejamiento      | Tamaño | Residuo a depurar |
|--|---|-------|--------|-------------------|--------|-------------------|
| <input type="checkbox"/> clientes      | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 8     | InnoDB | utf8_general_ci   | 32 KB  | -                 |
| <input type="checkbox"/> departamentos | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 7     | InnoDB | utf8_general_ci   | 16 KB  | -                 |
| <input type="checkbox"/> empleados     | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 10    | InnoDB | utf8_general_ci   | 48 KB  | -                 |
| <input type="checkbox"/> pedidos       | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 17    | InnoDB | utf8_general_ci   | 48 KB  | -                 |
| <input type="checkbox"/> productos     | ★ Examinar Estructura Buscar Insertar Vaciar Eliminar | 8     | InnoDB | utf8_general_ci   | 16 KB  | -                 |
| 5 tablas                               | Número de filas                                       | 50    | InnoDB | latin1_swedish_ci | 160 KB | 0 B               |

2.- Usaremos en nuestros ejercicios de esta práctica las tablas

**Departamentos** y **Empleados**, así que muestra la descripción y contenidos esas dos tablas creadas.

```
MariaDB [acme]> DESC departamentos;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| DEP_NO     | int(2)        | NO   | PRI | NULL    |       |
| DNOMBRE    | varchar(14)   | YES  |     | NULL    |       |
| LOCALIDAD  | varchar(10)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.014 sec)

MariaDB [acme]> DESC empleados;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| EMP_NO     | int(4)        | NO   | PRI | NULL    |       |
| APELLIDO   | varchar(8)    | YES  |     | NULL    |       |
| OFICIO     | varchar(10)   | YES  |     | NULL    |       |
| DIRECTOR   | int(4)        | YES  | MUL | NULL    |       |
| FECHA_ALTA | date          | YES  |     | NULL    |       |
| SALARIO    | float(6,2)    | YES  |     | NULL    |       |
| COMISION   | float(6,2)    | YES  |     | NULL    |       |
| DEP_NO     | int(2)        | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.012 sec)
```

## Práctica 18

3.- Crea dentro de **ACME** un procedimiento llamado '**ins**' sin parámetros que introduzca el número de Departamento **66** en la tabla con nombre **AUDITORIA** y localidad **TORRELAVEGA** en **Departamentos**. Comprueba su funcionamiento.

```
MariaDB [acme]> DELIMITER //
```

```
MariaDB [acme]> CREATE PROCEDURE ins()  
-> BEGIN  
->     INSERT INTO departamentos VALUES(66,"Auditoria","Torrelavega");  
-> END//
```

```
Query OK, 0 rows affected (0.005 sec)
```

```
MariaDB [acme]> DELIMITER ;
```

```
MariaDB [acme]> CALL ins();  
Query OK, 1 row affected, 1 warning (0.003 sec)
```

```
MariaDB [acme]> SELECT * FROM departamentos;
```

| DEP_NO | DNOMBRE       | LOCALIDAD   |
|--------|---------------|-------------|
| 10     | CONTABILIDAD  | BARCELONA   |
| 20     | INVESTIGACION | VALENCIA    |
| 30     | VENTAS        | MADRID      |
| 40     | PRODUCCION    | SEVILLA     |
| 50     | SEGURIDAD     | NULL        |
| 66     | Auditoria     | Torrelavega |
| 77     | CALIDAD       | NULL        |

```
7 rows in set (0.000 sec)
```

4.-Crea el procedimiento **ins\_dep** con dos parámetros de entrada **pnum** y **pnom** que permita hacer el ejercicio anterior indicando el número y el nombre del departamento a introducir. El campo **Lugar** quedará a **Null**.

```
MariaDB [acme]> DELIMITER //
```

```
MariaDB [acme]> CREATE PROCEDURE ins_dep(pnum INT(2), pnom VARCHAR(14))  
-> BEGIN  
->     INSERT INTO departamentos VALUES(pnum, pnom, null);  
-> END//
```

```
Query OK, 0 rows affected (0.009 sec)
```

```
MariaDB [acme]> DELIMITER ;
```

## Práctica 18

5.- Prueba si funciona con **Call Ins\_dep(77,'CALIDAD')**

```
MariaDB [acme]> Call Ins_dep(77,'CALIDAD');
Query OK, 1 row affected (0.002 sec)

MariaDB [acme]> SELECT * FROM departamentos;
+-----+-----+-----+
| DEP_NO | DNOMBRE      | LOCALIDAD |
+-----+-----+-----+
| 10     | CONTABILIDAD | BARCELONA |
| 20     | INVESTIGACION | VALENCIA  |
| 30     | VENTAS        | MADRID    |
| 40     | PRODUCCION    | SEVILLA   |
| 50     | SEGURIDAD     | NULL      |
| 66     | Auditoria     | Torrelaveg |
| 77     | CALIDAD       | NULL      |
+-----+-----+-----+
7 rows in set (0.000 sec)
```

6.- Borra el procedimiento **ins**.

```
MariaDB [acme]> DROP PROCEDURE ins;
Query OK, 0 rows affected (0.007 sec)
```

## Práctica 18

7.- Ejecuta y explica que hace este procedimiento:

```
Create Procedure Crea_Logistica()  
Begin  
Declare num int default 0;  
Select emp_no into num from empleados order by emp_no desc limit 1;  
Set num=num+1;  
Insert into departamentos Values (80,'LOGISTICA','RENEDO');  
Insert into empleados(emp_no,apellido,dep_no)  
Values (num,'ROMERALES',80);  
Set num=num+1;  
Insert into empleados(emp_no,apellido,dep_no)  
Values (num,'LOBATO',80);  
Update empleados set salario = 999 where dep_no=80 and salario is null;  
End//
```

```
MariaDB [acme]> DELIMITER //  
MariaDB [acme]> CREATE PROCEDURE Crea_Logistica()  
-> BEGIN  
-> DECLARE num INT DEFAULT 0;  
-> SELECT emp_no INTO num FROM empleados ORDER BY emp_no DESC LIMIT 1;  
-> SET num=num+1;  
-> INSERT INTO departamentos VALUES (80,'LOGISTICA','RENEDO');  
-> INSERT INTO empleados(emp_no,apellido,dep_no) VALUES (num,'ROMERALES',80);  
-> SET num=num+1;  
-> INSERT INTO empleados(emp_no,apellido,dep_no) VALUES (num,'LOBATO',80);  
-> UPDATE empleados SET salario = 999 WHERE dep_no=80 AND salario IS NULL;  
-> END//  
Query OK, 0 rows affected (0.009 sec)  
  
MariaDB [acme]> DELIMITER ;
```

```
MariaDB [acme]> CALL crea_logistica();  
Query OK, 6 rows affected (0.008 sec)
```

1. Guarda en la variable num el número del último empleado
2. Suma 1 a num
3. Inserta el departamento de logística Renedo
4. Inserta un empleado para el departamento anterior con el numero de la variable num
5. Suma 1 a num
6. Inserta otro empleado de la misma forma anterior
7. Actualiza el salario de los empleados del departamento de Renedo a 999 si son nulos

## Práctica 18

8.- Explica que hace este procedimiento, corrige los errores y ejecútalo para comprobarlo.

```
CREATE PROCEDURE s_altos()  
BEGIN  
SELECT nombre, salario FROM empleados WHERE salario > 2000;  
SELECT count(*) 'Total' FROM empleados WHERE salario > 2000;  
END//
```

```
MariaDB [acme]> DELIMITER //  
MariaDB [acme]> CREATE PROCEDURE s_altos()  
-> BEGIN  
->     SELECT apellido, salario FROM empleados WHERE salario > 2000;  
->     SELECT count(*) 'Total' FROM empleados WHERE salario > 2000;  
-> END//  
Query OK, 0 rows affected (0.011 sec)  
  
MariaDB [acme]> DELIMITER ;
```

Realiza dos consultas, la primera muestra el apellido y salario de los empleados con más de 2000 de salario, la segunda muestra el número de empleados con más de 2000 de salario.

```
MariaDB [acme]> CALL s_altos();  
+-----+-----+  
| apellido | salario |  
+-----+-----+  
| GARRIDO  | 3850.12 |  
| MARTINEZ | 2450.00 |  
| REY      | 6000.00 |  
| GIL      | 3350.00 |  
+-----+-----+  
4 rows in set (0.001 sec)  
  
+-----+  
| Total |  
+-----+  
|      4 |  
+-----+  
1 row in set (0.007 sec)  
  
Query OK, 0 rows affected (0.008 sec)
```

## Práctica 18

9.- Modifica el procedimiento anterior para que el valor con que se compara el sueldo se pase como parámetro de entrada y el número de empleados lo devuelva en un parámetro de salida.

Comprueba que funciona correctamente.

```
MariaDB [acme]> DELIMITER //
```

```
MariaDB [acme]> CREATE PROCEDURE s_altos2(sueldo FLOAT(6,2),OUT c_emp INT)
```

```
  -> BEGIN
```

```
  ->     SELECT apellido, salario FROM empleados WHERE salario > sueldo;
```

```
  ->     SELECT count(*) INTO c_emp FROM empleados WHERE salario > sueldo;
```

```
  -> END//
```

```
Query OK, 0 rows affected (0.009 sec)
```

```
MariaDB [acme]> DELIMITER ;
```

```
MariaDB [acme]> CALL s_altos2(1900,@x);
```

```
+-----+-----+
| apellido | salario |
+-----+-----+
| GARRIDO  | 3850.12 |
| MARTINEZ | 2450.00 |
| REY      | 6000.00 |
| GIL      | 3350.00 |
+-----+-----+
4 rows in set (0.000 sec)
```

```
Query OK, 1 row affected (0.007 sec)
```

```
MariaDB [acme]> SELECT @x;
```

```
+-----+
| @x    |
+-----+
|      4 |
+-----+
1 row in set (0.000 sec)
```

## Práctica 18

10.- Hacer una **función** utilizando **CASE** que reciba como parámetro de entrada la edad de un vendedor y devuelva el texto:

- **Veterano**                      si la edad es  $\geq 60$
- **Adulto**                        si la edad es  $\geq 30$  y  $< 60$
- **Joven**                         si la edad es  $< 30$

### Ejemplo de uso de la estructura CASE:

CASE

```
WHEN Quantity > 30 THEN "The quantity is greater than 30"
WHEN Quantity = 30 THEN "The quantity is 30"
ELSE "The quantity is under 30"
```

END

```
MariaDB [acme]> DELIMITER //
MariaDB [acme]> CREATE FUNCTION case_edad(edad INT) RETURNS VARCHAR(10)
-> BEGIN
->     DECLARE text VARCHAR(10);
->     SET text =
->         CASE
->             WHEN edad > 59 THEN "Veterano"
->             WHEN edad > 29 AND edad < 60 THEN "Adulto"
->             ELSE "Joven"
->         END;
->     RETURN text;
-> END//
Query OK, 0 rows affected (0.007 sec)
```

```
MariaDB [acme]> DELIMITER ;
```

```
MariaDB [acme]> SELECT case_edad(21);
+-----+
| case_edad(21) |
+-----+
| Joven          |
+-----+
1 row in set (0.001 sec)
```

## Práctica 18

11.- Explica que hace este procedimiento y prueba los distintos casos que pueden darse, valores correctos y errores.

```
delimiter //
Create Procedure INS_EMPLEADO(nro int, ape varchar(10), f_alt date, dep int)
Begin
Declare nume int;
Declare numd int;
Declare err int default 1;
if f_alt > curdate() then set err = 2;
else
    Select emp_no into nume from empleados where emp_no=nro limit 1;
    /* Nos aseguramos de que no de el error 1172 poniendo limit 1*/
    if nume is not null then
        set err=3;
    else
        Select empleados.dep_no into numd from empleados where
dep_no=dep limit 1;
        if numd is null then
            set err=4;
        end if;
    end if;
end if;
if err=1 then
    Insert into empleados(emp_no,apellido,fecha_alta,dep_no) Values
(nro,ape,f_alt,dep);
end if;
    Select  elt(err,"Empleado insertado","Fecha incorrecta","Ese nro de empleado
ya existe","Departamento inexistente") As 'Resultado';
End//
delimiter ;
```

1. Compara que la fecha pasada no sea mayor que la actual. Si es mayor mostrara un mensaje de error.
2. Guarda en la variable nume el valor pasado como parámetro en nro.
3. Si el número guardado está ocupado por otro empleado mostrara un mensaje de error.
4. Guarda en la variable numd el número del departamento, si el número del departamento existe.
5. Si el número de departamento guardado es null mostrara un mensaje de error.
6. Si el valor de la variable err es 1 guarda los datos en la tabla empleados
7. Muestra el valor de la variable err



## Práctica 18

Caso err=1

```
MariaDB [acme]> CALL ins_empleado(21,"Garcia","2023-01-18",50);
+-----+
| Resultado |
+-----+
| Empleado insertado |
+-----+
1 row in set (0.007 sec)

Query OK, 2 rows affected (0.009 sec)
```

Caso err=2

```
MariaDB [acme]> CALL ins_empleado(23,"Callejo","2024-01-18",50);
+-----+
| Resultado |
+-----+
| Fecha incorrecta |
+-----+
1 row in set (0.000 sec)

Query OK, 0 rows affected (0.003 sec)
```

Caso err=3

```
MariaDB [acme]> CALL ins_empleado(21,"Garcia","2023-01-18",50);
+-----+
| Resultado |
+-----+
| Ese nro de empleado ya existe |
+-----+
1 row in set (0.001 sec)

Query OK, 1 row affected (0.004 sec)
```

Caso err=4

```
MariaDB [acme]> CALL ins_empleado(25,"Murillo","2023-01-18",52);
+-----+
| Resultado |
+-----+
| Departamento inexistente |
+-----+
1 row in set (0.001 sec)

Query OK, 0 rows affected, 1 warning (0.004 sec)
```