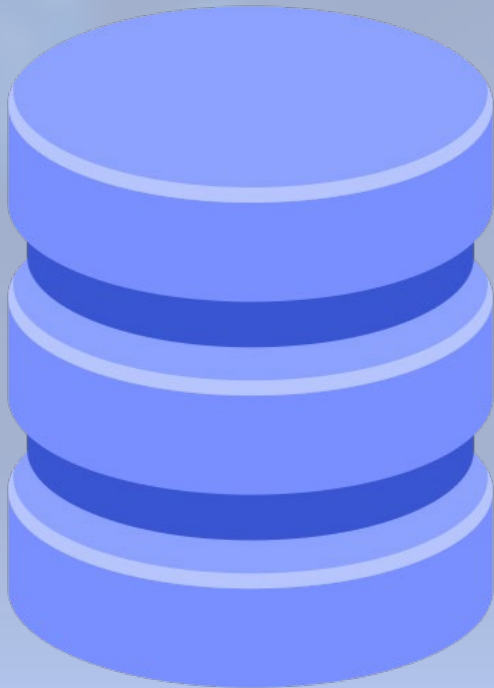


PROCEDIMIENTOS IV: MYSQL



**GESTIÓN DE BASES DE DATOS
JUAN CARLOS NAVIDAD GARCÍA**

1. Crear una rutina para realizar una copia de seguridad de la tabla artículos y ventas. Habrá que hacer una copia para los días pares y otra para los días impares.

```
CREATE PROCEDURE copia ()
BEGIN
IF day(curdate()) mod 2 = 0 THEN
DROP TABLE IF EXISTS `articulosp_bk`,`ventasp_bk`;
CREATE TABLE articulosp_bk as (select * from articulos);
CREATE TABLE ventasp_bk as (select * from ventas);
SELECT "Se ha creado la copia de seguridad del día par" as "Mensaje: ";
ELSE
DROP TABLE IF EXISTS `articulosi_bk`,`ventas_i_bk`;
CREATE TABLE articulosi_bk as (select * from articulos);
CREATE TABLE ventas_i_bk as (select * from ventas);
SELECT "Se ha creado la copia de seguridad del día impar" as "Mensaje: ";
END IF;
END //
```

```
mysql> call copia();
+-----+
| Mensaje: |
+-----+
| Se ha creado la copia de seguridad del día impar |
+-----+
1 row in set (0.06 sec)

Query OK, 0 rows affected (0.06 sec)
```

2. Crear una rutina para insertar una venta ahora. Comprobar que el código de artículo no existe previamente y que el código de proveedor que se va a añadir, si exista. Si no, informar de los errores.

```
CREATE PROCEDURE añadir (IN xart VARCHAR(4), IN xdesc VARCHAR(40), IN xprecio
DECIMAL, IN xpvp DECIMAL, IN xstock INT, IN xprov VARCHAR(3))
BEGIN
DECLARE cart INT;
DECLARE cprov INT;
SELECT COUNT(*) INTO cart FROM articulos WHERE codart=xart;
IF cart>0 THEN
SELECT "El código de artículo ya existe." AS "Mensaje: ";
ELSE
SELECT COUNT(*) INTO cprov FROM proveedores WHERE codigo=xprov;
IF cprov>0 THEN
INSERT INTO articulos VALUES (xart, xdesc, xprecio, xpvp, CURDATE(), xstock,
xprov);
SELECT "Artículo añadido." AS "Mensaje: ";
ELSE
SELECT "El proveedor introducido no existe." AS "Mensaje: ";
END IF;
END IF;
END //
```

```
mysql> CALL añadir("H003","Zapatillas Running",35.00,69.99,30,"ADI");
+-----+
| Mensaje: |
+-----+
| El codigo de articulo ya existe. |
+-----+
1 row in set (0.00 sec)
```

```
mysql> CALL añadir("H004","Zapatillas Running",35.00,69.99,30,"JVK");
+-----+
| Mensaje: |
+-----+
| El proveedor introducido no existe. |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected, 1 warning (0.00 sec)
```

```
mysql> CALL añadir("H004","Zapatillas Running",35.00,69.99,30,"ADI");
+-----+
| Mensaje: |
+-----+
| Articulo añadido. |
+-----+
1 row in set (0.00 sec)
```

3. Crear una rutina para obtener un listado de los artículos de un determinado proveedor.

```
CREATE PROCEDURE obt_art (IN xprov VARCHAR(3))
BEGIN
DECLARE cprov INT;
SELECT COUNT(*) INTO cprov FROM proveedores WHERE codigo=xprov;
IF cprov>0 THEN
SELECT * FROM articulos where proveedor=xprov;
ELSE
SELECT "El proveedor introducido no existe." AS "Mensaje: ";
END IF;
END //
```

```
mysql> CALL obt_art("ADI");
+-----+-----+-----+-----+-----+-----+
| CodArt | Descripcion | PrecioCompra | PVP | FechaCompra | Stock | Proveedor |
+-----+-----+-----+-----+-----+-----+
| A020 | Chancas Hombre Adilette | 3.95 | 24.95 | 2019-07-15 | 6 | ADI |
| A103 | Gorra con visera | 2.25 | 14.75 | 2019-03-15 | 12 | ADI |
| H004 | Zapatillas Running | 35.00 | 70.00 | 2023-03-03 | 30 | ADI |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> CALL obt_art("JVK");
+-----+
| Mensaje: |
+-----+
| El proveedor introducido no existe. |
+-----+
1 row in set (0.00 sec)
```

4. Crear una rutina para actualizar precios de un determinado proveedor, decrementándolos con un determinado porcentaje.

```
CREATE PROCEDURE menos_precio (IN xprov VARCHAR(3), IN xpor INT)
BEGIN
DECLARE cprov INT;
SELECT COUNT(*) INTO cprov FROM proveedores WHERE codigo=xprov;
IF cprov>0 THEN
UPDATE articulos SET pvp=pvp-((pvp*xpor)/100) WHERE proveedor=xprov;
SELECT "Precio actualizado." AS "Mensaje: ";
ELSE
SELECT "El proveedor introducido no existe." AS "Mensaje: ";
END IF;
END //
```

```
mysql> CALL menos_precio("ADI",10);
+-----+
| Mensaje:          |
+-----+
| Precio actualizado. |
+-----+
1 row in set (0.00 sec)
```

```
mysql> CALL menos_precio("JVK",10);
+-----+
| Mensaje:          |
+-----+
| El proveedor introducido no existe. |
+-----+
1 row in set (0.00 sec)
```

5. Crear una rutina para borrar los artículos no vendidos de la copia de seguridad creada en el primer ejercicio.

```
CREATE PROCEDURE borrar ()
BEGIN
DELETE FROM articulosi_bk WHERE codart NOT IN (SELECT codart FROM ventasi_bk);
DELETE FROM articulosp_bk WHERE codart NOT IN (SELECT codart FROM ventasp_bk);
SELECT "Registros no vendidos eliminados." AS "Mensaje: ";
END //
```

```
mysql> call borrar();
+-----+
| Mensaje:          |
+-----+
| Registros no vendidos eliminados. |
+-----+
1 row in set (0.00 sec)
```

6. Crear una rutina para calcular el total vendido hoy. Utilizarla para obtener un listado de las ventas diarias del último mes, por días. El formato del listado debería ser algo similar a:

Fecha(formato dd-mm-aaaa) Total Ventas

```
CREATE PROCEDURE caja (IN xfecha DATE)
BEGIN
SELECT DATE(fechaventa) AS "Fecha", SUM(pvp)-descuento AS "Total Ventas" FROM
articulos,ventas WHERE DATE(fechaventa)=xfecha;
END //
```

```
mysql> CALL caja (curdate());
+-----+-----+
| Fecha      | Total Ventas |
+-----+-----+
| 2023-03-03 |      4027.53 |
+-----+-----+
1 row in set (0.00 sec)
```