**Caso Práctico: Gestión de Productos con las 3 Tablas**

Vamos a analizar cómo funcionarían juntas estas tablas en un escenario real de venta de productos con diferentes presentaciones y códigos.

**Escenario: Venta de Lápices**

**Empresa:** "Materiales Escolares S.A." (ID\_Empresa = 1)

**1. Registro del Producto en catalogo**

sql

Copy

Download

INSERT INTO catalogo (

ID\_Empresa, Codigo\_Principal, Descripcion,

Cantidad\_Stock, Unidad\_Base, Precio\_Base

) VALUES (

1, 'LAP-001', 'Lápiz HB premium',

500, 'unidad', 2.50

);

**Resultado en catalogo:**

* ID\_Articulo: 101 (autogenerado)
* Stock: 500 unidades individuales
* Precio base: $2.50 por unidad

**2. Registro de Códigos Alternativos en catalogo\_codigos\_alternativos**

sql

Copy

Download

INSERT INTO catalogo\_codigos\_alternativos (

ID\_Articulo, ID\_Empresa, Tipo\_Codigo, Codigo, Es\_Principal

) VALUES

(101, 1, 'EAN', '1234567890123', 0),

(101, 1, 'UPC', '9876543210987', 0),

(101, 1, 'INTERNO', 'LAP-HB-PACK12', 1);

**Explicación:**

* EAN y UPC son códigos estándar para escaneo
* INTERNO es nuestro código para paquetes de 12
* Es\_Principal=1 marca nuestro código interno como principal

**3. Configuración de Unidades de Venta en catalogo\_unidades**

sql

Copy

Download

INSERT INTO catalogo\_unidades (

ID\_Articulo, ID\_Empresa, Unidad\_Origen, Unidad\_Destino,

Factor\_Conversion, Precio\_Venta, Codigo\_Barras

) VALUES

(101, 1, 'unidad', 'unidad', 1.0000, 2.50, '1234567890123'),

(101, 1, 'unidad', 'paquete', 12.0000, 25.00, 'LAP-HB-PACK12'),

(101, 1, 'unidad', 'caja', 120.0000, 220.00, 'CB-LAPIZ-120');

**Presentaciones configuradas:**

1. **Unidad**: $2.50 (código EAN)
2. **Paquete de 12**: $25.00 (10% descuento)
3. **Caja de 120**: $220.00 (15% descuento)

**Operaciones Comerciales Típicas**

**1. Venta de Productos**

**Caso 1:** Cliente compra 3 paquetes (36 unidades)

sql

Copy

Download

-- Verificar stock suficiente (36 unidades necesarias)

SELECT

Cantidad\_Stock AS Stock\_Actual,

FLOOR(Cantidad\_Stock / 12) AS Paquetes\_Disponibles

FROM catalogo

WHERE ID\_Articulo = 101;

-- Actualizar stock después de la venta

UPDATE catalogo

SET Cantidad\_Stock = Cantidad\_Stock - 36

WHERE ID\_Articulo = 101;

**2. Búsqueda de Productos**

**Por código de barras:**

sql

Copy

Download

SELECT

c.ID\_Articulo, c.Descripcion,

u.Unidad\_Destino AS Presentacion,

u.Factor\_Conversion AS Unidades\_Por\_Presentacion,

u.Precio\_Venta

FROM catalogo c

JOIN catalogo\_unidades u ON c.ID\_Articulo = u.ID\_Articulo

WHERE u.Codigo\_Barras = 'LAP-HB-PACK12' AND c.ID\_Empresa = 1;

**3. Recepción de Mercancía**

**Llegan 5 cajas nuevas (600 unidades):**

sql

Copy

Download

UPDATE catalogo

SET Cantidad\_Stock = Cantidad\_Stock + 600

WHERE ID\_Articulo = 101;

**4. Consulta de Presentaciones Disponibles**

sql

Copy

Download

SELECT

u.Unidad\_Destino AS Presentacion,

u.Factor\_Conversion AS 'Contiene Unidades',

u.Precio\_Venta AS 'Precio',

FLOOR(c.Cantidad\_Stock / u.Factor\_Conversion) AS 'Disponible'

FROM catalogo\_unidades u

JOIN catalogo c ON u.ID\_Articulo = c.ID\_Articulo

WHERE u.ID\_Articulo = 101 AND u.Es\_Vendible = 1;

**Resultado:**

| **Presentacion** | **Contiene Unidades** | **Precio** | **Disponible** |
| --- | --- | --- | --- |
| unidad | 1 | 2.50 | 500 |
| paquete | 12 | 25.00 | 41 |
| caja | 120 | 220.00 | 4 |

**1. Introducción**

**1.1 Objetivo**

Este sistema permite gestionar:  
✅ Artículos con múltiples códigos de identificación  
✅ Diferentes presentaciones de venta (unidad, paquete, caja)  
✅ Precios específicos por presentación  
✅ Control de inventario preciso

**1.2 Diagrama de Relaciones**

Copy

Download

account\_empresa (1) ── (n) catalogo (1) ── (n) catalogo\_codigos\_alternativos

│

└── (n) catalogo\_unidades

**2. Tablas Principales**

**2.1 Tabla CATALOGO**

**Estructura:**

| **Campo** | **Tipo** | **Descripción** |
| --- | --- | --- |
| ID\_Articulo | INT | Clave primaria (autoincremental) |
| ID\_Empresa | BIGINT | Vinculación a empresa |
| Codigo\_Principal | VARCHAR(50) | Código principal del artículo |
| Unidad\_Base | VARCHAR(50) | Siempre "unidad" |

**Ejemplo de Registro:**

sql

Copy

Download

INSERT INTO catalogo

(ID\_Empresa, Codigo\_Principal, Descripcion, Unidad\_Base)

VALUES

(1, 'LAP-001', 'Lápiz HB premium', 'unidad');

**2.2 Tabla CATALOGO\_CODIGOS\_ALTERNATIVOS**

**Casos de Uso:**

* Registrar EAN/UPC para escáneres
* Gestionar códigos internos
* Identificar lotes especiales

**Ejemplo:**

sql

Copy

Download

INSERT INTO catalogo\_codigos\_alternativos

(ID\_Articulo, ID\_Empresa, Tipo\_Codigo, Codigo)

VALUES

(101, 1, 'EAN', '1234567890123');

**2.3 Tabla CATALOGO\_UNIDADES**

**Lógica de Conversión:**

| **Unidad Origen** | **Unidad Destino** | **Factor** | **Ejemplo Real** |
| --- | --- | --- | --- |
| unidad | paquete | 12 | 12 lápices = 1 paquete |
| unidad | caja | 120 | 120 lápices = 1 caja |

**Consulta de Stock por Presentación:**

sql

Copy

Download

SELECT

Unidad\_Destino AS "Presentación",

FLOOR(500/Factor\_Conversion) AS "Disponible"

FROM catalogo\_unidades

WHERE ID\_Articulo = 101;

**3. Flujos de Trabajo**

**3.1 Alta de Nuevo Producto**

1. Insertar en catalogo
2. Registrar códigos alternativos
3. Definir presentaciones de venta

**3.2 Proceso de Venta**

sql

Copy

Download

-- 1. Verificar stock

SELECT Cantidad\_Stock FROM catalogo WHERE ID\_Articulo = 101;

-- 2. Actualizar tras vender 3 paquetes (36 unidades)

UPDATE catalogo SET Cantidad\_Stock = Cantidad\_Stock - 36

WHERE ID\_Articulo = 101;

**4. Anexos**

**4.1 Diccionario de Datos**

| **Campo** | **Valores Permitidos** |
| --- | --- |
| Tipo\_Codigo | EAN, UPC, INTERNO, LOTE |
| Unidad\_Destino | unidad, paquete, caja... |

**4.2 Consultas Útiles**

**Obtener todas las presentaciones de un artículo**:

sql

Copy

Download

SELECT \* FROM catalogo\_unidades

WHERE ID\_Articulo = 101 AND Es\_Vendible = 1;

| **Campo** | **Tipo** | **Descripción** |
| --- | --- | --- |
| ID\_Lote | INT | Identificador único del lote |
| ID\_Articulo | INT | Artículo asociado (relación con catalogo) |
| Numero\_Lote | VARCHAR(50) | Número identificador del lote (Ej: "LOT-2023-001") |
| Fecha\_Fabricacion | DATE | Fecha de producción |
| Fecha\_Caducidad | DATE | Fecha de vencimiento |
| Cantidad\_Inicial | INT | Cantidad inicial de unidades en el lote |
| Cantidad\_Disponible | INT | Unidades disponibles para venta/uso |
| Estado | ENUM | Estado actual del lote |

**7.1 Nuevo Diagrama de Relaciones**

Copy

Download

account\_empresa (1) ── (n) catalogo (1) ── (n) catalogo\_codigos\_alternativos

│

├── (n) catalogo\_unidades

└── (n) catalogo\_lotes

**7.2 Triggers Recomendados**

**Actualización automática de estado al agotarse:**

sql

Copy

Download

DELIMITER //

CREATE TRIGGER actualizar\_estado\_lote

AFTER UPDATE ON catalogo\_lotes

FOR EACH ROW

BEGIN

IF NEW.Cantidad\_Disponible <= 0 THEN

UPDATE catalogo\_lotes SET Estado = 'AGOTADO' WHERE ID\_Lote = NEW.ID\_Lote;

END IF;

END //

DELIMITER ;

**Verificación diaria de caducidad:**

sql

Copy

Download

DELIMITER //

CREATE PROCEDURE verificar\_caducidades()

BEGIN

UPDATE catalogo\_lotes

SET Estado = 'CADUCADO'

WHERE Fecha\_Caducidad < CURDATE()

AND Estado = 'ACTIVO';

END //

DELIMITER ;

-- Programar en Event Scheduler para ejecución diaria