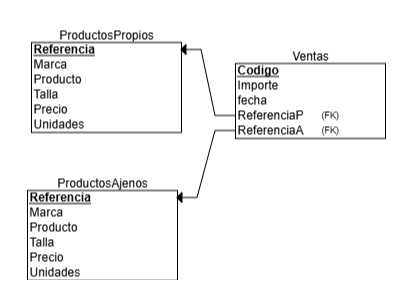
Necesitamos crear una base de datos para nuestra tienda de deportes CansadoSports.

Las tablas que vamos a necesitar para guardar los datos se referencian debajo. Cread las tablas utilizando el editor de SqlServer. **Echad un vistazo a los datos que se mencionan en los ejercicios siguientes, para saber el tipo de datos que debéis emplear en cada campo.**



**Transacciones:**

Durante el día de hoy, hemos realizado las siguientes operaciones en nuestra tienda, que debemos reflejar en la base de datos **mediante una sola transacción**.

Han entrado en el almacen, los siguientes productos nuevos:

Referencia: #321, marca:CansadoSports, Producto: Chándal, Precio 55, Talla 40, Unidades 6

Referencia #557, marca: Nike, Producto: Deportivos, Precio 120, Talla 40, Unidades 2

Referencia #239, marca:CansadoSports, Producto: Forro Polar, Precio 30, Talla XL, Unidades 2

Referencia #211, marca: Adidas, Producto: Camiseta, Precio 30, Talla L, Unidades 4

Referencia #232, marca:CansadoSports, Producto: Pantalones, Precio 40, Talla XL, Unidades 4

BEGIN TRANSACTION

BEGIN TRY

INSERT INTO ProductosPropios (Referencia, Marca, Producto, Precio, Talla, Unidades)

VALUES (321, 'CansadoSports', 'Chándal', 55, 40, 6);

INSERT INTO ProductosAjenos (Referencia, Marca, Producto, Precio, Talla, Unidades)

VALUES (557, 'Nike', 'Deportivos', 120, 40, 2);

INSERT INTO ProductosPropios (Referencia, Marca, Producto, Precio, Talla, Unidades)

VALUES (239, 'CansadoSports', 'Forro Polar', 30, NULL, 2);

INSERT INTO ProductosAjenos (Referencia, Marca, Producto, Precio, Talla, Unidades)

VALUES (211, 'Adidas', 'Camiseta', 30, NULL, 4);

INSERT INTO ProductosPropios (Referencia, Marca, Producto, Precio, Talla, Unidades)

VALUES (232, 'CansadoSports', 'Pantalones', 40, NULL, 4);

COMMIT;

PRINT 'TODO HA SALIDO BIEN'

END TRY

BEGIN CATCH

ROLLBACK;

PRINT 'ERROR AL INSERTAR LOS PRODUCTOS'

END CATCH;

**(PONGO NULL DONDE VA LA TALLA CON LETRAS PORQUE ME DA ERROR AL METERLO)**

Han salido del almacén los siguientes productos:

Una unidad del producto Chándal de marca CansadoSports con referencia #321

Dos unidades del producto Camiseta de marca Adidas con referencia #211

BEGIN TRANSACTION

BEGIN TRY

UPDATE ProductosPropios SET Unidades = 3 WHERE Referencia = 321;

UPDATE ProductosAjenos SET Unidades = 2 WHERE Referencia = 221

COMMIT;

PRINT 'TODO HA SALIDO BIEN'

END TRY

BEGIN CATCH

ROLLBACK;

PRINT 'ERROR AL INSERTAR LOS PRODUCTOS'

END CATCH;

Los productos que han salido del almacén, han generado las siguientes ventas

Código 014, importe 55 fecha 205-03-28 ReferenciaP #321

Código 015, importe 30 fecha 205-03-28 ReferenciaA #211

Código 016, importe 30 fecha 205-03-28 ReferenciaA #211

BEGIN TRANSACTION

BEGIN TRY

INSERT INTO Ventas (Codigo, Importe, Fecha, ReferenciaP)

VALUES (014, 55, '2025-03-28', 321);

INSERT INTO Ventas (Codigo, Importe, Fecha, ReferenciaA)

VALUES (015, 30, '2025-03-28', 211);

INSERT INTO Ventas (Codigo, Importe, Fecha, ReferenciaA)

VALUES (016, 30, '2025-03-28', 211);

COMMIT;

PRINT 'TODO HA SALIDO BIEN'

END TRY

BEGIN CATCH

ROLLBACK;

PRINT 'ERROR AL INSERTAR LOS PRODUCTOS'

END CATCH;

**Procedimientos almacenados**

1. Necesitamos dos procedimientos almacenados (uno para cada una de las tablas de productos), que nos permitan insertar un nuevo registro con los datos de un nuevo producto en la tabla. **El procedimiento recibirá todos los datos y realizará la inserción.**

CREATE PROCEDURE InsertarNuevosProductosPropios

@REFERENCIAPP INT,

@MARCAPP VARCHAR(50),

@PRODUCTOPP VARCHAR(50),

@TALLAPP VARCHAR(50),

@PRECIOPP INT,

@UNIDADESPP INT

AS

BEGIN

INSERT INTO ProductosPropios (Referencia, Marca, Producto, Talla, Precio, Unidades)

VALUES (@REFERENCIAPP, @MARCAPP, @PRODUCTOPP, @TALLAPP, @PRECIOPP, @UNIDADESPP);

END;

EXEC InsertarNuevosProductosPropios @REFERENCIAPP, @MARCAPP, @PRODUCTOPP, @TALLAPP, @PRECIOPP, @UNIDADESPP

**(EN EL EXEC SUSTITUIMOS LAS VARIABLES POR LOS DATOS QUE QUEREMOS METER PARA EL PRODUCTO)**

CREATE PROCEDURE InsertarNuevosProductosAjenos

@REFERENCIAPP INT,

@MARCAPP VARCHAR(50),

@PRODUCTOPP VARCHAR(50),

@TALLAPP VARCHAR(50),

@PRECIOPP INT,

@UNIDADESPP INT

AS

BEGIN

INSERT INTO ProductosPropios (Referencia, Marca, Producto, Talla, Precio, Unidades)

VALUES (@REFERENCIAPP, @MARCAPP, @PRODUCTOPP, @TALLAPP, @PRECIOPP, @UNIDADESPP);

END;

EXEC InsertarNuevosProductosAjenos @REFERENCIAPP, @MARCAPP, @PRODUCTOPP, @TALLAPP, @PRECIOPP, @UNIDADESPP

**(EN EL EXEC SUSTITUIMOS LAS VARIABLES POR LOS DATOS QUE QUEREMOS METER PARA EL PRODUCTO)**

1. También necesitamos dos procedimientos almacenados (uno procedimiento para cada una de las tablas de productos) que nos permita extraer una unidad del producto que le indiquemos (descontarla del número de unidades que tenemos) y generar una línea en la tabla de ventas en que se indique el precio, la referencia del producto y la fecha actual como fecha de la venta (podéis usar la función predefinida de SqlServer GETDATE() para obtener la fecha actual). **El procedimiento recibirá sólo la referencia del producto.**

CREATE PROCEDURE CreacionDeVentasConProductosPropios

@CODVENTA INT,

@IMPORTEVENTA INT,

@FECHAVENTA DATE,

@REFERENCIAPP INT

AS

BEGIN

SELECT ProductosPropios.Referencia FROM ProductosPropios

WHERE ProductosPropios.Referencia = @REFERENCIAPP

UPDATE ProductosPropios SET Unidades = Unidades - 1 WHERE Referencia = @REFERENCIAPP

INSERT INTO Ventas (Ventas.Codigo, Ventas.Importe, Ventas.Fecha, Ventas.ReferenciaP)

VALUES (@CODVENTA, @IMPORTEVENTA, @FECHAVENTA, @REFERENCIAPP)

END;

CREATE PROCEDURE CreacionDeVentasConProductosAjenos

@CODVENTA INT,

@IMPORTEVENTA INT,

@FECHAVENTA DATE,

@REFERENCIAPA INT

AS

BEGIN

SELECT ProductosAjenos.Referencia FROM ProductosAjenos

WHERE ProductosAjenos.Referencia = @REFERENCIAPA

UPDATE ProductosAjenos SET Unidades = Unidades - 1 WHERE Referencia = @REFERENCIAPA

INSERT INTO Ventas (Ventas.Codigo, Ventas.Importe, Ventas.Fecha, Ventas.ReferenciaP)

VALUES (@CODVENTA, @IMPORTEVENTA, @FECHAVENTA, @REFERENCIAPA)

END;

1. Por otro lado, necesitamos un procedimiento que nos informe de las ventas realizadas en un intervalo de fechas determinado. El procedimiento recibirá dos fechas y nos mostrará la información de los productos vendidos en ese intervalo de tiempo.

CREATE PROCEDURE VentasRealizadasEnDeterminadasFechas

@FECHAINICIO DATE,

@FECHAFINAL DATE,

@TOTALVENTAS INT OUTPUT

AS

BEGIN

SELECT @TOTALVENTAS = COUNT(\*) FROM Ventas

END;

DECLARE @NUMEROVENTAS INT

EXEC VentasRealizadasEnDeterminadasFechas 'fechainicio', 'fechafinal', @NUMEROVENTAS OUTPUT

PRINT 'EL NUMERO DE VENTAS EN ESAS FEHCAS ES DE' + @NUMEROVENTAS

1. También queremos un procedimiento que, dado el nombre de un producto (por ejemplo, camiseta), nos muestre las unidades de ese producto que tenemos en total (las de nuestra marca propia, más las de otras marcas).

CREATE PROCEDURE UnidadesProductoPorSuNombre

@PRENDAROPA VARCHAR(50),

@UNIDADESROPA INT OUTPUT

AS

BEGIN

SELECT ProductosPropios.Producto, ProductosAjenos.Producto

FROM ProductosPropios, ProductosAjenos

WHERE ProductosPropios.Producto = @PRENDAROPA AND ProductosAjenos.Producto = @PRENDAROPA

END;

DECLARE @NUMEROUNIDADES INT

EXEC UnidadesProductoPorSuNombre 'prendaropa', @NUMEROUNIDADES OUTPUT

PRINT @NUMEROUNIDADES

1. Por último, necesitamos un procedimiento que nos muestre un listado de todos los productos de los que tenemos stock. Tanto propios, como ajenos. El procedimiento mostrará la marca, el producto, la talla y las unidades.

CREATE PROCEDURE MostrarListadoDeProductosEnStock

@MARCAROPA VARCHAR(50) OUTPUT,

@PRODUCTOROPA VARCHAR(50) OUTPUT,

@TALLAROPA VARCHAR(50) OUTPUT,

@UNIDADESROPA INT OUTPUT

AS

BEGIN

IF @MARCAROPA = 'CansadoSports'

BEGIN

SELECT ProductosPropios.Producto, ProductosPropios.Talla, ProductosPropios.Unidades

FROM ProductosPropios

END

ELSE

BEGIN

SELECT ProductosAjenos.Producto, ProductosAjenos.Talla, ProductosAjenos.Unidades

FROM ProductosAjenos

END

IF @UNIDADESROPA < 0

BEGIN

PRINT 'NO HAY STOCK DE ESTA PRENDA'

END

END;

DECLARE @MARCA VARCHAR(50)

DECLARE @PRODUCTO VARCHAR(50)

DECLARE @TALLA VARCHAR(50)

DECLARE @UNIDADES VARCHAR(50)

EXEC MostrarListadoDeProductosEnStock @MARCA, @PRODUCTO, @TALLA, @UNIDADES