

Descripción de la temática de la base de datos

La base de datos está diseñada para cubrir las necesidades de una tienda dedicada a la **venta de piezas de artesanías indígenas** en México, contemplando aspectos **contables, logísticos y analíticos**. La información incluye la clasificación de los artículos (obras de arte), su almacenamiento en el inventario, la relación con artistas, los datos de clientes y vendedores, los pedidos, la facturación y los métodos de pago y envío. Al gestionar todos estos elementos de manera integrada, la tienda puede ofrecer un mejor servicio y obtener reportes útiles para la toma de decisiones.

Nota: He de mencionar que yo ya sabía algo de SQLServer; por lo tanto, primero cree la base de datos en SQLServer. No obstante, dado que en el curso se empleó MySQL, también cree la base de datos en MySQL (añado los dos scripts).

Introducción

El presente proyecto final busca sentar las bases para la administración de una **galería de artesanías indígenas en línea** que comercializa diversas categorías de piezas mexicanas (pintura, escultura, fotografía, entre otros). Se han definido entidades y sus relaciones de tal manera que la información fluya de forma coherente, permitiendo conocer la disponibilidad de obras, registrar a los clientes y vendedores, así como manejar todo el proceso de venta (pedido, envío, facturación y pago).

Ahora bien, dado que se busca asegurar la consistencia de la información, se han implementado restricciones (CHECK, FOREIGN KEY, UNIQUE) y se han creado índices adecuados para optimizar la recuperación de datos. Asimismo, la base de datos se ha pensado para escalar y admitir futuras expansiones, como la inclusión de sistemas de subastas, gestión de exposiciones o cualquier otro requerimiento que el negocio requiera con el tiempo.

Modelo de Negocio

La organización detrás de esta solución es una **galería de artesanías indígenas en línea** que ofrece piezas a coleccionistas y público en general a través de diferentes canales de venta (online e incluso algunos presenciales). Por lo tanto, la galería requiere un sistema que administre toda la cadena de valor: desde la adquisición o creación de la obra por parte de un artista, pasando por la clasificación y el control del inventario, hasta la venta final y el cobro efectivo. De esta manera, el modelo de negocio se apoya en la base de datos para garantizar la integridad de la información y respaldar sus operaciones diarias.

Problemática

La galería de arte en línea opera con sistemas dispersos y planillas de cálculo (Excel) que no unifican toda la información relevante, dificultando el seguimiento de inventarios, la facturación y el control de pagos. Así pues, la dispersión genera pérdida de datos, errores en las transacciones y escasa trazabilidad de las obras de arte. Con una base de datos centralizada y bien normalizada, se cubre esta brecha al contar con **una única fuente de verdad**, de fácil consulta y capaz de generar reportes detallados.

Además, al integrar en un mismo modelo la información de clientes, vendedores, pedidos y facturación, se simplifica la gestión administrativa y se mejoran los tiempos de respuesta. Lo anterior repercute positivamente en la satisfacción del cliente y en la optimización de los recursos de la tienda.

Objetivo

El objetivo principal es construir una **solución de base de datos robusta y escalable** que cubra el ciclo de vida de la venta de arte. Lo anterior incluye la gestión de artistas, la clasificación de las obras en categorías y subcategorías, el control de inventario, la emisión de pedidos y facturas, así como la posibilidad de asociar diferentes métodos de pago y envío a cada operación. La intención es que este diseño brinde una plataforma integral para el **análisis transversal** de la actividad comercial (por región, por artista, por tipo de obra, entre otros) y la automatización de tareas contables y logísticas.

Tablas

1. **País:** Registra los países donde se realizan las ventas o se ubican clientes y vendedores.
Campos principales: PaisId (PK), Nombre (UQ), Continente.
2. **Region:** Subdivisiones dentro de un país (por ejemplo, provincias o estados).
Campos: RegionId (PK), Nombre, PaisId (FK).
3. **Ciudad:** Registra las ciudades correspondientes a una región.
Campos: CiudadId (PK), Nombre, RegionId (FK).
4. **Direccion:** Estructura para almacenar direcciones físicas.
Campos: DireccionId (PK), Calle, Numero, CodigoPostal, CiudadId (FK).
5. **CategoriaArte:** Clasificación general de las obras (Pintura, Escultura, etc.).
Campos: CategoriaArteId (PK), NombreCategoria.
6. **SubCategoriaArte:** Subclasificación más específica (Óleo, Mármol, etc.).
Campos: SubCategoriaArteId (PK), NombreSubCategoria, CategoriaArteId (FK).
7. **Artista:** Información de los creadores (Nombre, Apellido, Nacionalidad).
Campos: ArtistaId (PK), Nombre, Apellido, Nacionalidad.
8. **MetodoPago:** Métodos de pago aceptados (Tarjeta, PayPal, Transferencia, etc.).
Campos: MetodoPagoId (PK), Descripcion.
9. **MetodoEnvio:** Distintos tipos de envío (Internacional, Nacional, etc.).
Campos: MetodoEnvioId (PK), Descripcion, TiempoEstimado.
10. **Fecha:** Tabla de fechas para referencia en varios procesos (registro, facturación, envío, etc.).
Campos: FechaId (PK), Fecha.
11. **Articulo:** Representación de una pieza de arte, con su código, título, descripción y autoría.
Campos: ArticuloId (PK), CodigoArticulo (UQ), Titulo, ArtistaId (FK), SubCategoriaArteId (FK), PrecioBase.
12. **Inventario:** Control de stock de cada Articulo, incluyendo su ubicación.
Campos: InventarioId (PK), ArticuloId (FK), Cantidad, Ubicacion.
13. **Cliente:** Registra a los compradores, asociando su dirección y la fecha de registro.
Campos: ClienteId (PK), TipoDocumento, NumeroDocumento (UQ combinado), Nombre, Apellido, DireccionId (FK), FechaRegistro (FK a Fecha).

14. **Vendedor:** Identifica a las personas que gestionan las ventas, con datos personales y fecha de registro.
Campos: VendedorId (PK), TipoDocumento, NumeroDocumento (UQ), Nombre, Apellido, DireccionId (FK), FechaRegistro (FK a Fecha).
15. **Pedido:** Encabezado de la orden de compra (incluye Cliente, Vendedor, Método de Envío y Fecha).
Campos: PedidoId (PK), NumeroPedido, ClienteId (FK), VendedorId (FK), FechaId (FK), MetodoEnvioId (FK).
16. **PedidoDetalle:** Contiene los artículos solicitados en cada Pedido, con cantidades y precio unitario.
Campos: PedidoDetalleId (PK), PedidoId (FK), ArticuloId (FK), Cantidad, PrecioUnitario.
17. **Envio:** Registra detalles concretos del envío (dirección final, fecha de envío, estado).
Campos: EnvioId (PK), PedidoId (FK), DireccionId (FK), FechaEnvio (FK a Fecha), EstadoEnvio.
18. **Factura:** Documento contable que asocia un pedido con un número de factura.
Campos: FacturaId (PK), PedidoId (FK), NroFactura, FechaId (FK).
19. **FacturaDetalle:** Permite desglosar conceptos monetarios de la factura (impuestos, subtotal, etc.).
Campos: FacturaDetalleId (PK), FacturaId (FK), Concepto, Importe.
20. **Pago:** Vincula una factura con un método de pago y la fecha de pago, junto con el importe pagado.
Campos: PagoId (PK), FacturaId (FK), MetodoPagoId (FK), FechaId (FK), ImportePagado.

Detalle de Tablas

A continuación, se listan las columnas, tipos de dato, restricciones y propiedades relevantes de las tablas (como **AUTO_INCREMENT**, **DEFAULT**, entre otros):

1. País: Almacena países y su respectivo continente.

- PaisId INT AUTO_INCREMENT PRIMARY KEY
- Nombre VARCHAR(50) NOT NULL
- Continente VARCHAR(50) NOT NULL
- UNIQUE (Nombre)
- CHECK (Nombre NOT REGEXP '[0-9]')

2. Region: Subdivisiones dentro de un país.

- RegionId INT AUTO_INCREMENT PRIMARY KEY
- Nombre VARCHAR(100) NOT NULL
- PaisId INT NOT NULL (FK a Pais.PaisId, CASCADE)
- CHECK (Nombre NOT REGEXP '[0-9]')

3. Ciudad: Ciudades que pertenecen a una región.

- CiudadId INT AUTO_INCREMENT PRIMARY KEY
- Nombre VARCHAR(100) NOT NULL
- RegionId INT NOT NULL (FK a Region.RegionId, CASCADE)
- CHECK (Nombre NOT REGEXP '[0-9]')

4. Direccion: Direcciones físicas, ligadas a una ciudad.

- DireccionId INT AUTO_INCREMENT PRIMARY KEY
- Calle VARCHAR(200) NOT NULL
- Numero VARCHAR(20) NOT NULL
- CodigoPostal VARCHAR(20) NOT NULL
- CiudadId INT NOT NULL (FK a Ciudad.CiudadId, CASCADE)
- CHECK (Numero REGEXP '^[0-9]+\$')

5. CategoriaArte: Clasificación general (Pintura, Escultura, etc.).

- CategoriaArteId INT AUTO_INCREMENT PRIMARY KEY
- NombreCategoria VARCHAR(100) NOT NULL

6. SubCategoriaArte: Subclasificación (ej. 'óleo', 'mármol', etc.).

- SubCategoriaArteId INT AUTO_INCREMENT PRIMARY KEY
- NombreSubCategoria VARCHAR(100) NOT NULL
- CategoriaArteId INT NOT NULL (FK a CategoriaArte.CategoriaArteId)
- CHECK (NombreSubCategoria NOT REGEXP '[0-9]')

7. Artista: Información de creadores: nombre, apellido, nacionalidad.

- ArtistaId INT AUTO_INCREMENT PRIMARY KEY
- Nombre VARCHAR(100) NOT NULL
- Apellido VARCHAR(100) NOT NULL
- Nacionalidad VARCHAR(100) NOT NULL
- CHECK (Nombre NOT REGEXP '[0-9]')
- CHECK (Apellido NOT REGEXP '[0-9]')

8. MetodoPago: Métodos de pago aceptados.

- MetodoPagoId INT AUTO_INCREMENT PRIMARY KEY
- Descripcion VARCHAR(50) NOT NULL
- CHECK (Descripcion NOT REGEXP '[0-9]')

9. MetodoEnvio: Distintas modalidades de envío.

- MetodoEnvioId INT AUTO_INCREMENT PRIMARY KEY
- Descripcion VARCHAR(50) NOT NULL
- TiempoEstimado VARCHAR(50) NOT NULL
- CHECK (Descripcion NOT REGEXP '[0-9]')

10. Fecha: Almacena fechas para registros, envíos, etc.

- FechaId INT AUTO_INCREMENT PRIMARY KEY
- Fecha DATE NOT NULL

11. Artículo: Representa una pieza de arte a la venta.

- ArtículoId INT AUTO_INCREMENT PRIMARY KEY
- CodigoArticulo VARCHAR(50) UNIQUE NOT NULL
- Titulo VARCHAR(200) NOT NULL
- Descripcion TEXT NOT NULL
- ArtistaId INT NOT NULL (FK a Artista.ArtistaId)
- SubCategoriaArteId INT NOT NULL (FK a SubCategoriaArte.SubCategoriaArteId)
- PrecioBase DECIMAL(18,2) NOT NULL CHECK (PrecioBase >= 0)
- AnoCreacion INT NULL

12. Inventario: Control de stock y ubicación física.

- InventarioId INT AUTO_INCREMENT PRIMARY KEY
- ArtículoId INT NOT NULL (FK a Artículo.ArticuloId)
- Cantidad INT NOT NULL CHECK (Cantidad >= 0)
- Ubicacion VARCHAR(200) NOT NULL

13. Cliente: Información de compradores.

- ClienteId INT AUTO_INCREMENT PRIMARY KEY
- TipoDocumento VARCHAR(10) NOT NULL
- NumeroDocumento VARCHAR(20) NOT NULL
- Nombre VARCHAR(50) NOT NULL
- Apellido VARCHAR(50) NOT NULL
- DireccionId INT NOT NULL (FK a Direccion.DireccionId)
- FechaRegistro INT NOT NULL (FK a Fecha.FechaId)
- UNIQUE (TipoDocumento, NumeroDocumento)
- CHECK (Nombre NOT REGEXP '[0-9]')
- CHECK (Apellido NOT REGEXP '[0-9]')

14. Vendedor: Datos de quienes gestionan las ventas.

- VendedorId INT AUTO_INCREMENT PRIMARY KEY
- TipoDocumento VARCHAR(10) NOT NULL
- NumeroDocumento VARCHAR(20) NOT NULL
- Nombre VARCHAR(50) NOT NULL
- Apellido VARCHAR(50) NOT NULL
- DireccionId INT NOT NULL (FK a Direccion.DireccionId)
- FechaRegistro INT NOT NULL (FK a Fecha.FechaId)
- UNIQUE (TipoDocumento, NumeroDocumento)
- CHECK (Nombre NOT REGEXP '[0-9]')
- CHECK (Apellido NOT REGEXP '[0-9]')

15. Pedido: Encabezado de una orden de compra.

- PedidoId INT AUTO_INCREMENT PRIMARY KEY
- NumeroPedido INT NOT NULL
- ClienteId INT NOT NULL (FK a Cliente.ClienteId)
- VendedorId INT NOT NULL (FK a Vendedor.VendedorId)
- FechaId INT NOT NULL (FK a Fecha.FechaId)
- MetodoEnvioId INT NOT NULL (FK a MetodoEnvio.MetodoEnvioId)

16. PedidoDetalle: Línea detallada de artículos solicitados.

- PedidoDetalleId INT AUTO_INCREMENT PRIMARY KEY
- PedidoId INT NOT NULL (FK a Pedido.PedidoId)
- ArtículoId INT NOT NULL (FK a Artículo.ArticuloId)
- Cantidad INT NOT NULL CHECK (Cantidad > 0)
- PrecioUnitario DECIMAL(18,2) NOT NULL CHECK (PrecioUnitario >= 0)

17. Envio: Gestión concreta del envío.

- EnvioId INT AUTO_INCREMENT PRIMARY KEY
- PedidoId INT NOT NULL (FK a Pedido.PedidoId)
- DireccionId INT NOT NULL (FK a Direccion.DireccionId)
- FechaEnvio INT NULL (FK a Fecha.FechaId)
- EstadoEnvio VARCHAR(50) NOT NULL

18. Factura: Documento contable asociado a un pedido.

- FacturaId INT AUTO_INCREMENT PRIMARY KEY
- PedidoId INT NOT NULL (FK a Pedido.PedidoId)
- NroFactura INT NOT NULL
- FechaId INT NOT NULL (FK a Fecha.FechaId)

19. FacturaDetalle: Desglose de conceptos monetarios en la factura.

- FacturaDetalleId INT AUTO_INCREMENT PRIMARY KEY
- FacturaId INT NOT NULL (FK a Factura.FacturaId)
- Concepto VARCHAR(200) NOT NULL
- Importe DECIMAL(18,2) NOT NULL

20. Pago: Registra cómo se paga la factura y su importe.

- PagoId INT AUTO_INCREMENT PRIMARY KEY
- FacturaId INT NOT NULL (FK a Factura.FacturaId)
- MetodoPagoId INT NOT NULL (FK a MetodoPago.MetodoPagoId)
- FechaId INT NOT NULL (FK a Fecha.FechaId)
- ImportePagado DECIMAL(18,2) NOT NULL

Vistas

1. vw_ArteGeneral

- **Objetivo:** Combinar la información principal de pedidos, clientes, vendedores, artículos, facturas y pagos, para obtener un panorama unificado de las ventas.
- **Tablas involucradas:** Pedido, PedidoDetalle, Articulo, Cliente, Vendedor, Factura, Pago, MetodoPago, MetodoEnvio, Envio.
- **Uso:** Facilita reportes globales de ventas, mostrando el pedido, el artículo, el cliente, forma de pago, envío, etc.

2. vw_ResumenClientes

- **Objetivo:** Mostrar el total de pedidos por cliente y su última fecha de pedido.
- **Tablas involucradas:** Cliente, Pedido, Fecha.
- **Uso:** Ayudar en el análisis de hábitos de compra y para identificar clientes frecuentes o inactivos.

3. vw_StockArticulos

- **Objetivo:** Visualizar la disponibilidad (stock) y ubicación de los artículos, junto con su precio base.
- **Tablas involucradas:** Articulo, Inventario.
- **Uso:** Proporcionar una consulta rápida de inventario y precios.

Funciones

FN_CalcularIVA

- **Descripción:** Calcula el IVA (21%) de un valor base.
- **Objetivo:** Ayudar en cálculos de precio final o reporte de montos con IVA.
- **Tablas manipuladas:** No manipula tablas. Es únicamente una función aritmética.

Stored Procedures

SP_ActualizarStock

- **Descripción:** Dado un **ArticuloId** y una **CantidadVendida**, descuenta dicha cantidad de la tabla **Inventario**.
- **Objetivo:** Automatizar la reducción de stock al concretar una venta o registrar un **PedidoDetalle**.
- **Tablas con las que interactúa:** **Inventario**.

Triggers

TR_RestarStockDespuesPedido

- **Descripción:** Después de insertar un nuevo registro en **PedidoDetalle**, llama al SP **SP_ActualizarStock** para descontar la cantidad vendida del inventario.
- **Objetivo:** Mantener sincronizado el stock sin que el usuario deba hacerlo manualmente.
- **Tablas involucradas:** Evento en **PedidoDetalle**; actualiza **Inventario** mediante el Stored Procedure.

Conclusión

La presente estructura ofrece una base sólida para la administración de la galería. No obstante, es posible incorporar módulos adicionales (como sistemas de subastas, exhibiciones o suscripciones) conforme la galería vaya evolucionando y sus necesidades operativas se amplíen. Gracias a la normalización y a la implementación de restricciones, la ampliación de la base de datos puede hacerse de forma organizada y sin comprometer la integridad de la información.

Instrucciones finales

Ejecutar primero el archivo **DBArte_estructura_inserciones.sql** para crear la BD, sus tablas y cargar los datos iniciales (y la vista **vw_ArteGeneral**). Luego, ejecutar el archivo **DBArte_objetos_avanzados.sql** para crear las vistas complementarias, la función, el stored procedure y el trigger.