

Vinoteca El Copihue - Proyecto de Base de Datos Relacional

Autor: Alexei Sandoval

Sumario

1. Introducción
2. Objetivo
3. Situación Problemática
4. Modelo de Negocio
5. Diagrama E-R
6. Listado de Tablas
7. Scripts de Creación
8. Vistas
9. Funciones
10. Procedimientos Almacenados
11. Triggers
12. Scripts de Inserción de Datos
13. Consultas de Análisis
14. Herramientas y Tecnologías Utilizadas
15. Conclusión

1. Introducción

Este proyecto se centra en el diseño y la implementación de una base de datos relacional para "Vinoteca El Copihue", un negocio especializado en la venta de vinos chilenos. El objetivo es centralizar y optimizar la gestión de datos críticos como el inventario, la información de clientes, proveedores y el seguimiento de ventas.

2. Objetivo

El principal propósito de esta base de datos es funcionar como el núcleo operativo del negocio, resolviendo las deficiencias del sistema manual. Se busca específicamente:

- Optimizar el control de stock de vinos.
- Mejorar la gestión de la información de clientes y proveedores.
- Mantener un registro detallado y ordenado de pedidos y ventas.
- Facilitar la generación de reportes para el análisis de negocio.
- Sentar las bases para futuras aplicaciones, como un sistema de punto de venta.

3. Situación Problemática

Actualmente, la vinoteca maneja sus registros en planillas y documentos de texto, lo que genera problemas como la pérdida de información, inconsistencias de datos y una gestión de stock ineficiente. La falta de un sistema centralizado impide realizar análisis de negocio efectivos.

4. Modelo de Negocio

La "Vinoteca El Copihue" opera con un modelo de negocio de venta al por menor. Su cadena de valor incluye:

- **Abastecimiento:** Compra de vinos a viñas y distribuidores (proveedores).
- **Catálogo:** Clasificación y registro de vinos con detalles como tipo, precio y stock.
- **Venta:** Los clientes realizan pedidos que se registran en el sistema.
- **Gestión de Pedidos:** Se actualiza el inventario y se monitorea el estado de cada pedido.

5. Diagrama E-R

El siguiente diagrama representa la estructura lógica de la base de datos, mostrando las entidades (tablas) y sus relaciones.

6. Listado de Tablas

A continuación, se describen las tablas y sus columnas, con sus respectivos tipos de datos y claves.

Tabla: `cliente` *Descripción:* Almacena la información de los clientes.

Abreviatura	Nombre Completo	Tipo de Dato	Clave
<code>cliente_id</code>	ID del Cliente	<code>INT</code> <code>AUTO_INCREMENT</code>	PK
<code>nombre</code>	Nombre del Cliente	<code>VARCHAR(100)</code>	
<code>apellido</code>	Apellido del Cliente	<code>VARCHAR(100)</code>	
<code>email</code>	Email del Cliente	<code>VARCHAR(100)</code>	UNIQUE

telefono	Teléfono	VARCHAR(20)
----------	----------	-------------

direccion	Dirección	VARCHAR(255)
-----------	-----------	--------------

fecha_registro	Fecha de Registro	TIMESTAMP
----------------	-------------------	-----------

Tabla: proveedor Descripción: Contiene los datos de las viñas o distribuidores.

Abreviatura	Nombre Completo	Tipo de Dato	Clave
id_proveedor	ID del Proveedor	INT AUTO_INCREMENT	PK
nombre_proveedor	Nombre del Proveedor	VARCHAR(150)	
contacto_persona	Persona de Contacto	VARCHAR(100)	
telefono	Teléfono	VARCHAR(20)	
email	Email	VARCHAR(100)	UNIQUE

Tabla: categoria Descripción: Clasifica los vinos según su tipo.

Abreviatura	Nombre Completo	Tipo de Dato	Clave
-------------	-----------------	--------------	-------

<code>id_categoria</code>	ID de la Categoría	<code>INT</code> <code>AUTO_INCREMENT</code>	PK
<code>nombre_categoria</code>	Nombre de la Categoría	<code>VARCHAR(100)</code>	UNIQUE
<code>descripcion</code>	Descripción	<code>TEXT</code>	

Tabla: `vino` *Descripción:* Almacena el catálogo de vinos.

Abreviatura	Nombre Completo	Tipo de Dato	Clave
<code>vino_id</code>	ID del Vino	<code>INT</code> <code>AUTO_INCREMENT</code>	PK
<code>nombre_vino</code>	Nombre del Vino	<code>VARCHAR(200)</code>	INDEX
<code>descripcion</code>	Descripción	<code>TEXT</code>	
<code>cosecha</code>	Año de Cosecha	<code>YEAR</code>	
<code>precio</code>	Precio Unitario	<code>DECIMAL(10, 2)</code>	
<code>proveedor_id</code>	ID del Proveedor	<code>INT</code>	FK

Tabla: `uva` *Descripción:* Contiene las variedades de uva.

Abreviatura	Nombre Completo	Tipo de Dato	Clave
uva_id	ID de la Uva	INT AUTO_INCREMENT	PK
nombre_uva	Nombre de la Uva	VARCHAR(100)	UNIQUE
descripcion	Descripción	TEXT	

Tabla: bodega Descripción: Almacena información sobre las bodegas.

Abreviatura	Nombre Completo	Tipo de Dato	Clave
bodega_id	ID de la Bodega	INT AUTO_INCREMENT	PK
nombre_bodega	Nombre de la Bodega	VARCHAR(100)	
region	Región	VARCHAR(100)	
pais	País	VARCHAR(100)	

Tabla: stock Descripción: Mantiene el stock de vinos por sucursal.

Abreviatura	Nombre Completo	Tipo de Dato	Clave
stock_id	ID del Stock	INT AUTO_INCREMENT	PK
sucursal_id	ID de la Sucursal	INT	FK
vino_id	ID del Vino	INT	FK
cantidad	Cantidad en Stock	INT	

Tabla: pedido Descripción: Registra cada pedido realizado por un cliente.

Abreviatura	Nombre Completo	Tipo de Dato	Clave
id_pedido	ID del Pedido	INT AUTO_INCREMENT	PK
id_cliente	ID del Cliente	INT	FK
id_empleado	ID del Empleado	INT	FK
id_sucursal	ID de la Sucursal	INT	FK

fecha_pedido	Fecha del Pedido	TIMESTAMP	INDEX
--------------	------------------	-----------	-------

total	Total del Pedido	DECIMAL(10, 2)
-------	------------------	----------------

estado	Estado del Pedido	VARCHAR(50)
--------	-------------------	-------------

Tabla: detalle_pedido Descripción: Detalles de los productos en cada pedido.

Abreviatura	Nombre Completo	Tipo de Dato	Clave
detalle_id	ID del Detalle	INT AUTO_INCREMENT	PK
pedido_id	ID del Pedido	INT	FK
vino_id	ID del Vino	INT	FK
cantidad	Cantidad de Productos	INT	
precio_unitario	Precio en el momento de la compra	DECIMAL(10, 2)	

7. Scripts de Creación

Los scripts completos para la creación de la base de datos y sus tablas están disponibles en el siguiente repositorio:

- [GitHub: AlexeiSandoval23/bdcoderhouse](#)

Un fragmento destacado del script de creación de la base de datos es:

```
SET autocommit=0;
```

```
SET FOREIGN_KEY_CHECKS=0;
```

```
DROP DATABASE IF EXISTS vinoteca_el_copihue;
```

```
CREATE DATABASE vinoteca_el_copihue CHARACTER SET utf8mb4 COLLATE  
utf8mb4_spanish_ci;
```

```
USE vinoteca_el_copihue;
```

```
CREATE TABLE cliente (
```

```
    cliente_id INT AUTO_INCREMENT PRIMARY KEY,
```

```
    nombre VARCHAR(100) NOT NULL,
```

```
    apellido VARCHAR(100) NOT NULL,
```

```
    email VARCHAR(100) NOT NULL UNIQUE,
```

```
    telefono VARCHAR(20),
```

```
    direccion VARCHAR(255),
```

```
    fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
    INDEX (apellido, nombre)
```

```
) ENGINE=InnoDB;
```

```
-- Otros CREATE TABLE...
```

8. Vistas

Las vistas simplifican la complejidad de consultas recurrentes, pre-calculando uniones y agregaciones de datos.

vw_catálogo_vinos *Descripción:* Combina la información de vinos con su proveedor, categorías, uvas y bodega. *Objetivo:* Facilitar la consulta del catálogo completo de vinos con una sola tabla virtual. *Script:*

```
CREATE VIEW vw_catálogo_vinos AS
```



```

SELECT

    v.vino_id,

    v.nombre_vino,

    v.cosecha,

    v.precio,

    p.nombre_proveedor AS proveedor,

    GROUP_CONCAT(DISTINCT c.nombre_categoria SEPARATOR ', ') AS categorias,

    GROUP_CONCAT(DISTINCT u.nombre_uva SEPARATOR ', ') AS uvas,

    b.nombre_bodega AS bodega,

    b.region,

    b.pais

FROM vino v

JOIN proveedor p ON v.proveedor_id = p.proveedor_id

LEFT JOIN vino_categoria vc ON v.vino_id = vc.vino_id

LEFT JOIN categoria c ON vc.categoria_id = c.categoria_id

LEFT JOIN vino_uva vu ON v.vino_id = vu.vino_id

LEFT JOIN uva u ON vu.uva_id = u.uva_id

LEFT JOIN vino_bodega vb ON v.vino_id = vb.vino_id

LEFT JOIN bodega b ON vb.bodega_id = b.bodega_id

GROUP BY v.vino_id, v.nombre_vino, v.cosecha, v.precio, proveedor, bodega, region, pais

ORDER BY v.nombre_vino;

```

vw_resumen_ventas_cliente *Descripción:* Muestra el total de pedidos y gasto de cada cliente. *Objetivo:* Permitir un análisis rápido del comportamiento de compra de los clientes.

vw_stock_sucursal *Descripción:* Muestra el stock actual de vinos por sucursal. *Objetivo:* Facilitar la gestión del inventario por ubicación.

vw_productos_mas_vendidos *Descripción:* Lista los 10 vinos más vendidos. *Objetivo:* Identificar los productos con mejor desempeño.

vw_detalle_pedido_completo *Descripción:* Proporciona una vista detallada de los pedidos, incluyendo información del cliente, sucursal y vinos. *Objetivo:* Simplificar la generación de reportes de ventas detallados.

9. Funciones

Las funciones permiten encapsular lógica de negocio y reutilizarla en consultas o en otros objetos de la base de datos.

fn_calcular_total_pedido(p_pedido_id INT) *Descripción:* Calcula el total de un pedido sumando los subtotales de sus productos. *Script:*

```
DELIMITER //

CREATE FUNCTION fn_calcular_total_pedido(p_pedido_id INT)

RETURNS DECIMAL(10, 2)

DETERMINISTIC

BEGIN

    DECLARE v_total DECIMAL(10, 2);

    SELECT SUM(cantidad * precio_unitario) INTO v_total

    FROM detalle_pedido

    WHERE pedido_id = p_pedido_id;

    RETURN IFNULL(v_total, 0.00);

END //

DELIMITER ;
```

fn_obtener_stock_disponible(p_sucursal_id INT, p_vino_id INT)

Descripción: Devuelve el stock disponible de un vino en una sucursal específica. *Objetivo:* Útil para validaciones en transacciones de venta.

10. Procedimientos Almacenados

Los procedimientos almacenados permiten automatizar y ejecutar transacciones complejas de forma segura y eficiente.

sp_crear_pedido(IN p_cliente_id INT, IN p_empleado_id INT, IN p_sucursal_id INT, IN p_items_json JSON) *Descripción:* Inserta un nuevo pedido y sus detalles de manera atómica, garantizando la consistencia de los datos. *Objetivo:* Centralizar la lógica de negocio para la creación de pedidos.

sp_obtener_historial_cliente(IN p_cliente_id INT) *Descripción:* Genera un reporte del historial de pedidos de un cliente, mostrando los productos comprados y sus detalles. *Objetivo:* Proporcionar una herramienta para el análisis del comportamiento de compra de los clientes.

11. Triggers

Los triggers son disparadores que ejecutan código de forma automática ante eventos específicos en las tablas.

tr_disminuir_stock_pedido *Descripción:* Disminuye automáticamente el stock de un vino al insertar un nuevo detalle de pedido. También valida que haya suficiente stock disponible antes de la inserción. *Evento:* BEFORE INSERT en la tabla `detalle_pedido`.

tr_actualizar_total_pedido *Descripción:* Actualiza el campo `total` en la tabla `pedido` cada vez que se inserta un `detalle_pedido`. *Evento:* AFTER INSERT en la tabla `detalle_pedido`.

12. Scripts de Inserción de Datos

Los scripts de inserción de datos poblan las tablas con información inicial para pruebas y uso real. Un fragmento destacado del script es:

```
INSERT INTO empleado (nombre, apellido, puesto, fecha_contratacion, email) VALUES
```

```
('Juan', 'Pérez', 'Gerente', '2023-01-15', 'juan.perez@vinoteca.com'),
```

```
('María', 'Gómez', 'Asistente de Ventas', '2023-02-20', 'maria.gomez@vinoteca.com');
```

```
INSERT INTO cliente (nombre, apellido, email, telefono, direccion) VALUES
```

```
('Javier', 'González', 'javier.gonzalez@correo.cl', '+56987654321', 'Los Cerezos 123, Ñuñoa, Santiago'),
```

```
('Carolina', 'Soto', 'caro.soto@correo.cl', '+56912345678', 'Av. Alemania 45, Concepción');
```

13. Consultas de Análisis

Estas consultas permiten extraer información valiosa para la toma de decisiones.

- **Ventas Totales por Mes y Año:** Muestra el desempeño de las ventas a lo largo del tiempo.
- **Los 10 Vinos Más Vendidos por Cantidad:** Identifica los productos estrella.
- **Resumen de Ventas por Sucursal:** Analiza el rendimiento de cada punto de venta.
- **Distribución de Vinos por Categoría:** Muestra la variedad de vinos en el catálogo.
- **Clientes Más Frecuentes y con Mayor Gasto:** Identifica a los clientes más valiosos.
- **Vinos con Stock Bajo:** Ayuda en la gestión del inventario y la planificación de compras.
- **Historial de Pagos por Método:** Analiza las preferencias de pago de los clientes.
- **Vinos por Uva:** Muestra la popularidad de las distintas variedades de uva.
- **Pedidos que no han sido entregados:** Permite monitorear el estado de los envíos.
- **Movimientos de Stock Recientes:** Proporciona un registro de las entradas y salidas de inventario.

14. Herramientas y Tecnologías Utilizadas

- **MySQL:** Sistema de gestión de bases de datos relacionales.
- **SQL:** Lenguaje de consulta para la manipulación de datos.
- **Git y GitHub:** Control de versiones y repositorio de código.

15. Conclusión

La implementación de esta base de datos relacional para "Vinoteca El Copihue" representa una mejora significativa en la gestión operativa del negocio. Al centralizar y estructurar la información, se eliminan los problemas inherentes a los sistemas manuales. La base de datos no solo optimiza el control de inventario y el seguimiento de pedidos, sino que también proporciona una plataforma sólida para el análisis de datos. La capacidad de generar reportes detallados y obtener información en tiempo real sobre las ventas, el stock y el comportamiento de los clientes, permitirá a la vinoteca tomar decisiones más informadas y estratégicas, impulsando así su crecimiento y competitividad en el mercado.