

Administración de sistemas de venta



- Pre entrega SQL 2024
- Alumno: Lucas Quinteros
- Comisión: 53190
- Proyecto: Creación de bases de dato para sistemas de ventas
- Curso: SQL

Introducción

Un cliente cuenta con un negocio el cual comenzó a expandirse en la cantidad de ventas por lo cual le surge la necesidad de solicitar un sistema para poder automatizar tareas y poder gestionar mejor el negocio.

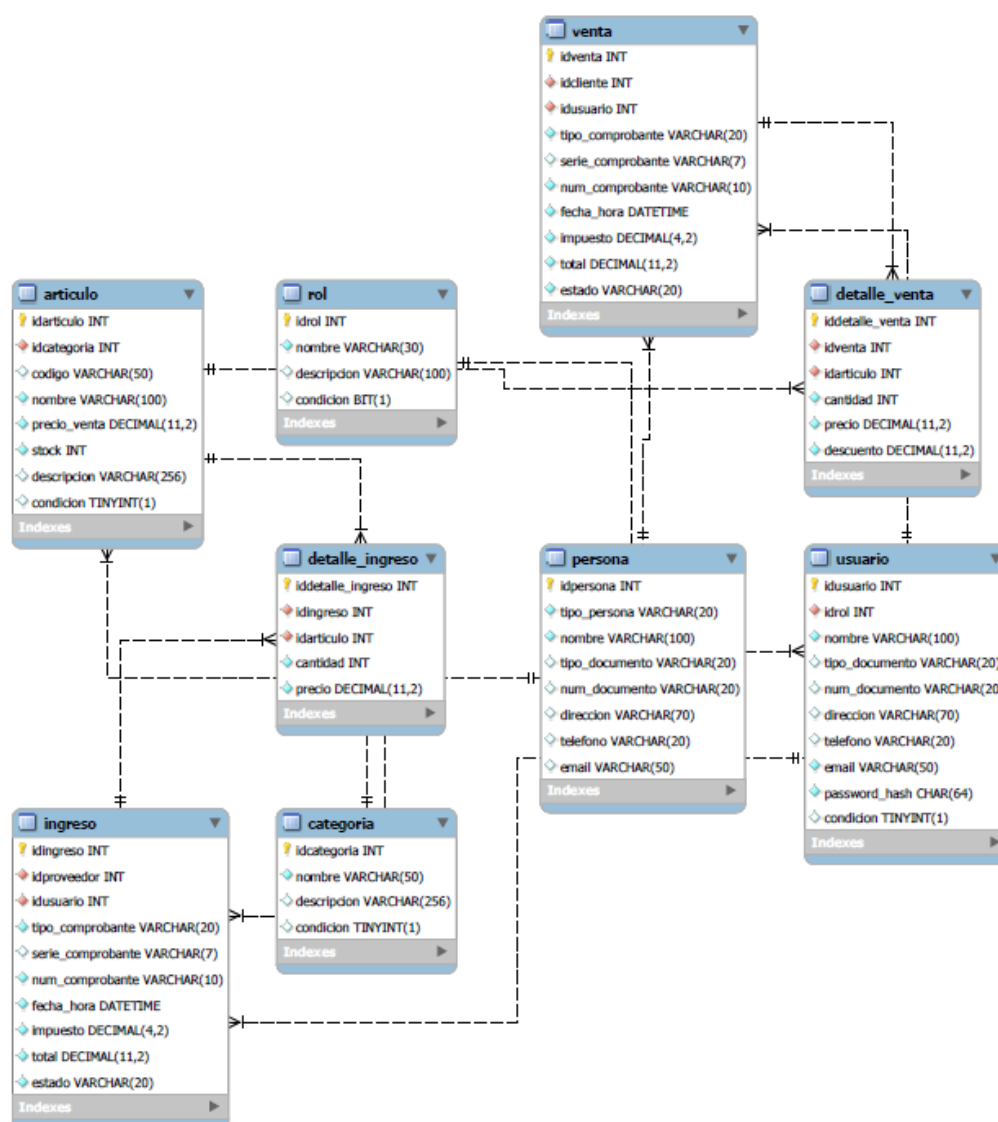
Objetivo

Crear las bases de datos para escalable para que el área de desarrollo pueda a comenzar a trabajar sobre el armado de la codificación para el sistema de ventas personalizado para el cliente.

Proceso

Se trabajo en conjunto con el cliente para relevar datos y planillas de Excel con las cual trabaja actualmente. En base lo recopilado y conversado con las personas que operan en el negocio.

Diagrama entidad relación



Listado de tablas

Tabla categoría: Es una tabla donde se describe la categoría de producto, hace referencia al rubro. La misma no se elimina solo se desactiva.

Nombre de la columna	Tipo de dato	LEN	Prinart key	Unique	auto increment	not null
idcategoria	integer		true	true	true	
nombre	varchar	50		true		true
descripcion	varchar	256				
Condición	Tinyint	1				

Tabla personas: Almacena información en personas siendo estas diferenciadas en clientes o proveedores

Nombre de la columna	Tipo de dato	LEN	Prinart key	foreign key	Unique	auto increment	not null
idpersona	integer		true			true	
tipo_persona	varchar	20					true
nombre	varchar	100					true
tipo_documento	varchar	20					
num_documento	varchar	20					
direccion	varchar	70					
telefono	varchar	20					
email	varchar	50					

Tablas usuario: Hace referencia a las personas que utilizan el sistema y almacena sus datos personales.

Nombre de la columna	Tipo de dato	LEN	Prinart key	foreign key	Unique	auto increment	not null
idusuario	integer		true	true		true	true
idrol	integer						
nombre	varchar	100					
tipo_documento	varchar	20					
num_documento	varchar	20					
direccion	varchar	70					
telefono	varchar	20					
email	varchar	50					
password_hash	char	64					
condicion	tinyint	1					

Tabla rol: Hace referencia el tipo de rol de acceso y estado que tendrán los usuarios que accederán al sistema.

Nombre de la columna	Tipo de dato	LEN	Prinart key	foreign key	Unique	auto increment	not null
idrol	integer		true			true	
nombre	varchar	30					true
descripcion	varchar	100					
condicion	bit default	1					

Tabla detalle ingreso: Hace referencia a las características de artículo que ingreso, cantidad, precio.

Nombre de la columna	Tipo de dato	LEN	Primary key	foreign key	Unique	auto increment	not null
iddetalle	integer		true			true	
idingreso	integer						true
idarticulo	integer						true
cantidad	integer			true			true
precio	decimal	11.2		true			true

Tabla ingreso: Hace referencia para registrar los ingresos de mercadería a al depósito.

Nombre de la columna	Tipo de dato	LEN	Prinart key	foreign key	Unique	auto increment	not null
idingreso	integer		true			true	
idproveedor	integer			true			true
idusuario	integer			true			true
tipo_comprobante	varchar	20					true
serie_comprobante	varchar	7					
num_comprobante	varchar	10					true
fecha_hora	datetime						
impuesto	decimal	4.2					true
total	varchar	11.2					true
estado		20					true

Tabla articulo: Hace referencia donde se registran el código de identificación del artículo, precio, stock, categoría.

Nombre de la columna	Tipo de dato	LEN	Primary key	foreign key	Unique	auto increment	not null
idarticulo	integer		tue			true	
idcategoria	integer			true			true
codigo	varchar	50					
nombre	varchar	100			true		true
precio_venta	decimal						
stock	integer						true
descripcion	varchar	256					
condicion	tinyint	1					

Tabla detalle venta: Hace referencia a los detalles por los cuales se realizaron las ventas.

Nombre de la columna	Tipo de dato	LEN	Primary key	foreign key	Unique	auto increment	not null
iddetalle	integer		true			true	
idventas	integer			true			true
idarticulo	integer			true			true
cantidad	integer						true
precio	decimal	11.2					true
descuento	decimal	11.2					true

Tabla Ventas: Donde se registran las ventas, comprobantes, cliente, usuario, etc.

Nombre de la columna	Tipo de dato	LEN	Primary key	foreign key	Unique	auto increment	not null
idventas	integer		true			true	
idecliente	integer			true			true
ideusuario	integer			true			true
tipo_comprobante	varchar						true
serie_comprobante	varchar	20					
num_comprobante	varchar	7					
fecha_hora	datetime	10					true
impuesto	decimal	4.2					true
total	decimal	11.2					true
estado	varchar	20					true

Segunda entrega

Vistas

La vista ventas

Esta vista muestra información detallada sobre las ventas, incluyendo detalles de los artículos vendidos y sus categorías.

- Joins: Une las tablas venta, detalle_venta, articulo y categoria.
- Columnas Seleccionadas: Información de la venta (ID, cliente, usuario, tipo de comprobante, fecha, total, etc.), información del artículo (nombre y precio de venta) y la categoría del artículo.
- Orden: Los resultados se ordenan por la fecha y hora de la venta.

detalle ingreso articulo

Esta vista proporciona detalles sobre los ingresos de artículos, incluyendo información detallada de cada artículo ingresado.

- Joins: Une las tablas detalle_ingreso y articulo.
- Columnas Seleccionadas: Detalles del ingreso (ID del detalle, ID del ingreso, cantidad, precio), información del artículo (código, nombre, precio de venta, stock, descripción y condición).
- Orden: Los resultados se ordenan por el ID del detalle de ingreso.

vista_detalle_venta_usuario

Esta vista combina detalles de ventas con la información del usuario que realizó la venta.

- Joins: Une las tablas detalle_venta, articulo, venta y usuario.
- Columnas Seleccionadas: Detalles de la venta (ID del detalle, ID de la venta, cantidad, precio, descuento), información del artículo (código, nombre, precio de venta, stock, descripción y condición) y detalles del usuario (ID, nombre, tipo de documento, número de documento, dirección, teléfono, email y condición).
- Orden: Los resultados se ordenan por el ID del detalle de venta.

vista_articulos_categorias

Esta vista muestra detalles de los artículos junto con su información de categoría.

- Joins: Une las tablas articulo y categoria.
- Columnas Seleccionadas: Información del artículo (ID, ID de categoría, código, nombre, precio de venta, stock, descripción y condición) y detalles de la categoría (nombre, descripción y condición).
- Orden: Los resultados se ordenan por el nombre del artículo.

vista_articulos_detalle_venta

Esta vista combina detalles de ventas de artículos con información de los roles (asumiendo una relación específica).

- Joins: Une las tablas detalle_venta, articulo y rol.
- Columnas Seleccionadas: Detalles de la venta (ID del detalle, ID de la venta, cantidad, precio, descuento), información del artículo (código, nombre, precio de venta, stock, descripción y condición) y detalles del rol (ID, nombre, descripción y condición).
- Orden: Los resultados se ordenan por el nombre del artículo.

Funciones

1. Función calcular_stock

Esta función calcula el stock total de todos los artículos en la tabla artículo.

- **DELIMITER //**: Cambia el delimitador estándar de MySQL (;) a // para permitir la definición de la función. Esto es necesario porque el cuerpo de la función contiene múltiples declaraciones.
- **CREATE FUNCTION calcular_stock()**: Define una nueva función llamada **calcular_stock** que no toma ningún parámetro y devuelve un valor de tipo **INT**.
- **RETURNS INT**: Especifica que la función devuelve un valor entero.
- **BEGIN ... END**: Delimita el cuerpo de la función.

- **DECLARE total_stock INT:** Declara una variable local **total_stock** de tipo entero.
- **SELECT SUM(stock) INTO total_stock FROM articulo:** Calcula la suma del campo **stock** de todos los registros en la tabla **articulo** y almacena el resultado en la variable **total_stock**.
- **RETURN total_stock:** Devuelve el valor de **total_stock**.
- **DELIMITER ;:** Restaura el delimitador estándar (;).
- **SELECT calcular_stock() AS stock_total:** Llama a la función **calcular_stock** y muestra el resultado con el alias **stock_total**.

Explicación

Esta función es útil para obtener rápidamente el total de todos los artículos en inventario. Suma los valores del campo **stock** en la tabla **artículo** y devuelve el resultado.

2. Función calcular_iva

Esta función calcula el IVA (Impuesto sobre el Valor Añadido) de un artículo dado su precio.

- **DELIMITER //:** Cambia el delimitador estándar de MySQL (;) a // para permitir la definición de la función.
- **CREATE FUNCTION calcular_iva(precio DECIMAL(11,2)):** Define una nueva función llamada **calcular_iva** que toma un parámetro **precio** de tipo **DECIMAL(11,2)** y devuelve un valor de tipo **DECIMAL(11,2)**.
- **RETURNS DECIMAL(11,2):** Especifica que la función devuelve un valor decimal con una precisión de hasta 11 dígitos y 2 decimales.
- **BEGIN ... END:** Delimita el cuerpo de la función.
- **DECLARE iva DECIMAL(11,2):** Declara una variable local **iva** de tipo decimal.
- **SET iva = precio * 0.21:** Calcula el IVA como el 21% del precio proporcionado y almacena el resultado en la variable **iva**.
- **RETURN iva:** Devuelve el valor de **iva**.
- **DELIMITER ;:** Restaura el delimitador estándar (;).

Explicación

Esta función es útil para calcular el IVA de un artículo dado su precio. Multiplica el precio del artículo por la tasa de IVA (21% en este caso) y devuelve el resultado.

Procedimientos almacenados

calcular_precio_total

Este procedimiento calcula el precio total de un artículo dado su ID, junto con el IVA.

Parámetros de Entrada: p_idarticulo (INT) - ID del artículo del cual se calculará el precio total.

Parámetros de Salida:

p_precio_venta (DECIMAL(11,2)) - Precio de venta del artículo.

p_iva (DECIMAL(11,2)) - Valor del IVA calculado.

p_precio_total (DECIMAL(11,2)) - Precio total del artículo (precio de venta + IVA).

registrar_venta

Este procedimiento registra una venta en la tabla venta.

Parámetros de Entrada: Datos necesarios para registrar una venta: ID de cliente, ID de usuario, tipo de comprobante, serie de comprobante, número de comprobante, fecha y hora, impuesto, total y estado.

Acción: Inserta una nueva fila en la tabla venta con los datos proporcionados.

Disparador after_detalle_venta_insert

Este disparador se activa después de insertar una fila en la tabla detalle_venta y actualiza el stock del artículo correspondiente.

Acción: Resta la cantidad vendida del stock del artículo correspondiente en la tabla articulo.

Disparador before_detalle_ingreso_insert

Este disparador se activa antes de insertar una fila en la tabla detalle_ingreso y verifica que la cantidad de artículos sea mayor que cero.

Acción: Verifica que la cantidad de artículos a ingresar sea mayor que cero. Si la condición no se cumple, se emite una señal de error con un mensaje personalizado.

Links de github

<https://github.com/lucas2351/segundaentrega.git>