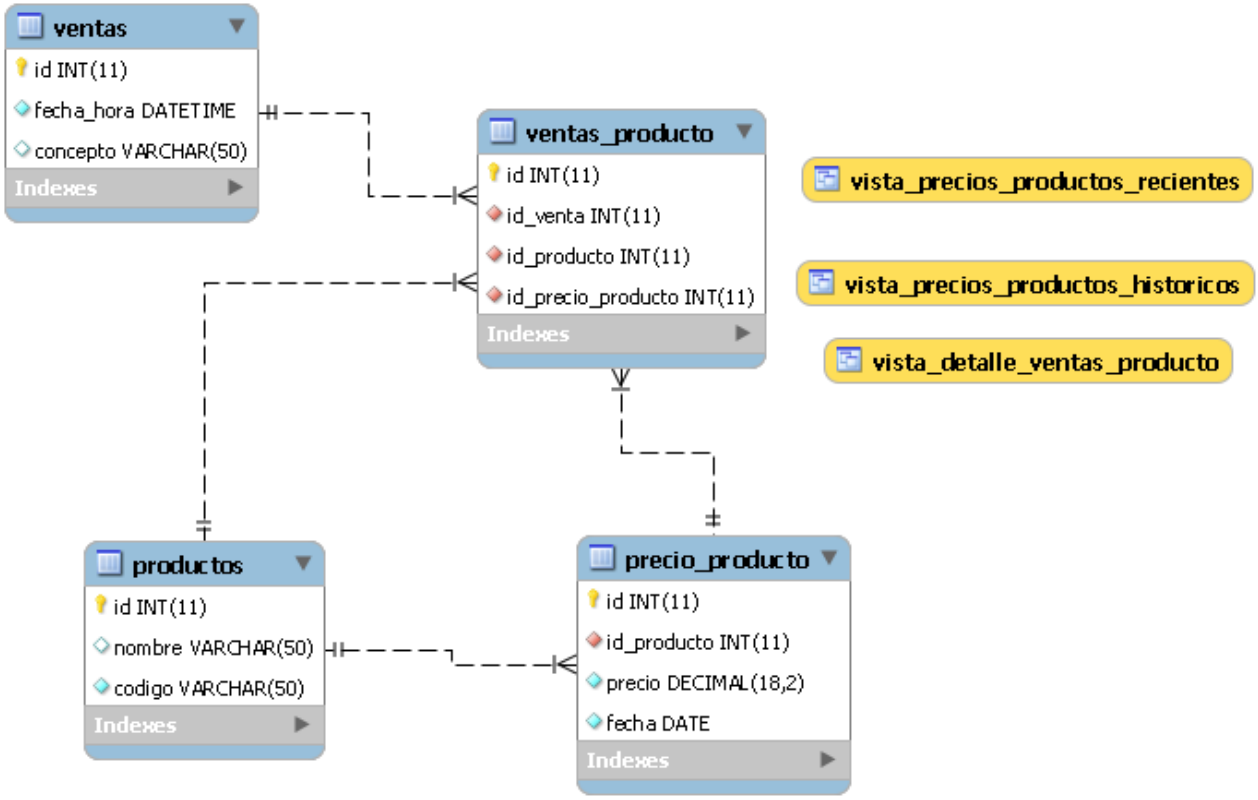
Trabajo final Bases de datos
Segunda Parte

introducción:

Para la realización del trabajo final se creó una base de datos que simula un negocio de venta de productos con 4 tablas 3 vistas y 4 store procedures.

Diagrama entidad relación:



Para la DB se pensó en un modelo de negocio de ventas el cual necesita tener, un listado de precio de los productos y un listado de ventas de artículos.

Para el listado de productos y precios se crearon dos tablas relacionadas entre si. La idea de las misas es que se puedan actualizar los precios de los productos manteniendo un historial de precios y sus cambios en el tiempo.

En la tabla Productos se almacena el nombre del producto, el código que debe ser único y el id para poder relacionarlo con la tabla de precios.

La tabla de precio_producto, tiene la finalidad de ir guardando los datos de precios, si se actualizar el precio de un producto se almacena la fecha en la que se hizo el cambio de precio

Para la visualización de los productos se crearon 2 vistas Vista_precios_productos_historicos y

- La Vista_precios_productos_historicos trae el listado completo de de precios y productos.
- La Vista_precios_productos_recientes, trae los productos y el ultimo precio agregado por producto.

Para el almacenamiento de las ventas se crearon dos tablas, una tabla principal llamada ventas donde se almacena el id y el concepto de la venta, y una tabla ventas_producto la cual tiene 4

columnas, una columna ID que actúa como clave principal y 3 columnas que son claves foraneas: id_venta, id_producto, id_precio_producto.

Para la visualización de los datos se creó una vista llamada vista_detalle_ventas_producto que muestra el detalle de las ventas.

Listado de Store Procedures

1. InsertarProductoConPrecio:

```
CREATE PROCEDURE  InsertarProductoConPrecio(
   IN p_nombre VARCHAR(50),
   IN p_codigo VARCHAR(50),
   IN p_precio DECIMAL(18, 2),
   IN p_fecha DATE
)
BEGIN

   DECLARE v_producto_id INT;


   SELECT id INTO v_producto_id
   FROM productos
   WHERE codigo = p_codigo;
   IF v_producto_id IS NOT NULL THEN
      UPDATE productos
      SET nombre = p_nombre
      WHERE id = v_producto_id;
      INSERT INTO precio_producto (id_producto, precio, fecha)
      VALUES (v_producto_id, p_precio, p_fecha);
   ELSE
      INSERT INTO productos (nombre, codigo)
      VALUES (p_nombre, p_codigo);
      SET v_producto_id = LAST_INSERT_ID();
      INSERT INTO precio_producto (id_producto, precio, fecha)
      VALUES (v_producto_id, p_precio, p_fecha);

   END IF;

END //

DELIMITER ;
```

Este SP tiene por objetivo almacenar el nombre del producto y en simultaneo el precio, si el producto ya fue ingresado anteriormente pero tiene un nuevo precio, se procede a actualizar los datos del mismo en ambas tablas

2. ObtenerProductoConUltimoPrecioPorCodigo

```
DELIMITER //
```

```
CREATE PROCEDURE ObtenerProductoConUltimoPrecioPorCodigo(
   IN p_codigo VARCHAR(50)
)
BEGIN
   SELECT
      *
   FROM
      sistema_ventas.vista_precios_productos_recientes
   WHERE codigo = p_codigo;
END //

DELIMITER ;
```

Me permite traer el precio mas actual del producto

3. InsertarVenta.

```
DELIMITER //

CREATE PROCEDURE InsertarVenta(
   IN p_concepto VARCHAR(50),
   OUT p_id_venta INT
)
BEGIN
   INSERT INTO sistema_ventas.ventas(
      fecha_hora,
      concepto
   )
   VALUES
   (NOW(), p_concepto);

   -- Asignar el último ID insertado al parámetro OUT
   SELECT LAST_INSERT_ID() INTO p_id_venta;

END //

DELIMITER ;
```

Permite guardar la venta y trae el ultimo ID almacenado

4. InsertarVentaProducto:

```
DELIMITER //

CREATE PROCEDURE InsertarVentaProducto(
   IN in_id_venta INT,
   IN in_codigo_producto VARCHAR(50)
)
BEGIN

   DECLARE v_id_producto INT;
   DECLARE v_id_precio_producto INT;
   DECLARE v_precio_venta DECIMAL(18,2);
```

```
    SELECT id INTO v_id_producto
    FROM sistema_ventas.productos
    WHERE codigo = in_codigo_producto;
    IF v_id_producto IS NULL THEN

        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error: Producto no
encontrado con el código proporcionado.';
    ELSE

        SELECT pp.id, pp.precio
        INTO v_id_precio_producto, v_precio_venta
        FROM sistema_ventas.precio_producto pp
        WHERE pp.id_producto = v_id_producto
        ORDER BY pp.fecha DESC, pp.id DESC
        LIMIT 1;
        IF v_id_precio_producto IS NULL THEN
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Error: No se encontró un
precio para el producto.';
        ELSE
            INSERT INTO sistema_ventas.ventas_producto
            (
                id_venta,
                id_producto,
                id_precio_producto
            )
            VALUES
            (
                in_id_venta,
                v_id_producto,
                v_id_precio_producto
            );
        END IF;
    END IF;

END //
```

Guarda uno por uno cada producto vendido

https://github.com/schvemler/IdeaSchvemler.git