Proyecto Final “Tina Cafe”

Coderhouse - SQL

Anaclara Soria
Profesor: Anderson Ocaña
Tutor: Nicolás Maugeri
Comisión: 59430

Introducción

Mi nombre es Anaclara Soria, curso en Coderhouse Base de datos en SQL, y presentare como proyecto una base de datos basado en una cafetería (ficticio) pet friendly libre de gluten dedicada a ofrecer una experiencia única en un ambiente cálido y moderno con sabores de calidad

Situación Problemática y Objetivo

Actualmente, mi cafetería enfrenta varios problemas en la administración de sus ventas y pedidos. A raíz del incremento de la demanda, se ha vuelto difícil medir la rentabilidad real del negocio, realizar un correcto seguimiento de ventas, controlar el inventario y actualizar la información referida a clientes, órdenes y empleados.

Veo necesario crear un sistema que permita almacenar y manipular toda esta información con el objetivo de tenemos mejor visibilidad de los resultados del negocio y tomar decisiones efectivas.

Descripción de la base de datos:

Tablas

Proveedor: Almacena información sobre los proveedores de productos. Campos: id_proveedor (PK), id_producto, nombre, telefono, direccion, email.

Producto: Almacena información sobre los productos ofrecidos en el café. Campos: id_producto (PK), id_categoria_producto (FK), tipo_producto, nombre, precio, stock.

Categoria_Producto: Define las categorías a las que pertenecen los productos. Campos: id_categoria_producto (PK), nombre, descripción, fecha _ actualizacion, subcategoria.

Cliente: Almacena información sobre los clientes. Campos: id_cliente (PK), nombre, DNI, email, teléfono.

Empleado: Almacena información sobre los empleados del café. Campos: id_empleado (PK), nombre, direccion, email, telefono, puesto.

Orden: Almacena información sobre las órdenes realizadas por los clientes. Campos: id_orden (PK), id_cliente (FK), id_producto (FK), id_empleado (FK), fecha, estado.

Descripción de la base de datos:

Tablas

Pagos: Almacena información sobre los pagos realizados por las órdenes. Campos: id_pagos (PK), id_detalle_orden (FK), fecha, descuento, metodo de pago, estado de pago.

Detalle_orden: Almacena los detalles de cada orden, incluyendo los productos y cantidades. Campos: id_detalle_orden (PK), id_orden (FK), id_producto (FK), comentarios, cantidades, metodo_entrega.

Calificaciones: Almacena las calificaciones tanto positivas como negativas de cada cliente en cada orden.

id_calificacion (PK), id_cliente (FK), id_orden (FK), comentarios, tipo_comentario, calificacion.

Envíos: Almacena información referida a los envíos a domicilio. Campos: id_envio (PK), id_orden (FK), id_empleado (FK), direccion, fecha, estado.

Inventario: Almacena información referida al stock de productos por categoría con el objetivo de mantener una buena administración y anticiparse a realizar nuevas ordenes a tiempo. Campos: id_inventario (PK), id_producto (FK), stock, fecha ingreso, fecha vencimiento, estado producto.

Importación de datos

- Se importaron datos, a través del comando INSERT INTO, desde la terminal de GitHub
- Los datos fueron creados a partir de información ficticia obtenida desde la pagina web Mockaroo y ChatGPT.

Dichos datos se encuentran en repositorio de GitHub.
Su orden esta establecido en el documento ‘population’.

Objetos SQL - Funciones

Estas funciones tienen el objetivo de realizar una operación y devolver un valor. A continuación se muestra las dos funciones creadas para esta DB.

Función: tina_cafe.fx_total_pedidos

Descripción: Función que declara la cantidad de pedidos tomados por empleado.

Objetivo: Proporcionar la cantidad de pedidos tomados por cada empleado a raíz de la información proporcionada en las tablas de orden y empleados con el objetivo de monitorear que la carga de trabajo se encuentre correctamente balanceada.

Parámetros: id_empleado

Retorno: cantidad de órdenes por empleado (INT).

Función: tina_cafe.fx_producto_mas_vendido

Descripción: Función que declara los productos más vendidos.

Objetivo: Proporcionar el nombre del producto con más ventas con el objetivo de tener visibilidad de la rentabilidad del negocio y poder realizar estrategias más eficientes.

Parámetros: id_producto

Retorno: los dos productos con mayores unidades vendidas (VARCHAR).

Objetos SQL - Store Procedures

Los SP tienen como objetivo almacenar un conjunto de instrucciones, para agilizar aquellos procedimientos que se manejan de manera repetitiva. Se han creado dos SP para esta DB.

Procedimiento: registrar_proveedor

.

Objetivo: Permite insertar nuevos proveedores en la base de datos proveedor.

Tabla involucrada: proveedor.

Procedimiento: registrar_orden

.

Objetivo: Permite insertar nuevas ordenes en la base de datos orden.

Tabla involucrada: orden

Objetos SQL - Vistas

El objetivo de las vistas, es crear tablas virtuales a partir de tablas reales, para mostrar la información de las mismas, sin que sean alterados los datos. Esta DB posee dos vistas:

vw_ventas_por_periodo

Descripción: Sumatoria de los precios de los productos vendidos. Objetivo: identificar los periodos con mayores y menores ventas para tomar medidas al respecto.

vw_ventas_por_producto

Descripción: Total de unidades vendidas y la suma de ingresos por cada producto con el objetivo de visualizar qué productos son más populares.

vw_ventas_por_categoria

Descripción: Total de ventas y unidades por cada categoría de producto (ej., bebidas, comidas, snacks).

vw_ventas_por_empleado

Descripción: Analizar qué empleados están generando más ingresos de acuerdo a las órdenes colocadas

Objetos SQL - Triggers

El objetivo de los Triggers es desencadenar una acción específica cuando se realiza algún determinado evento. Los Triggers utilizados en esta DB son:

Trigger: Actualizar_stock_producto

Descripción: Este trigger tiene como objetivo actualizar el stock de los productos en la tabla producto cuando se crea una nueva orden. Cuando se inserta un registro en la tabla detalle_orden, que está relacionada con la orden, el stock de los productos que están siendo comprados debe ser disminuido en la cantidad especificada en Cantidades_orden.

Acción: Insert

Trigger: Prevenir_pago_para_orden_cancelada

Descripción: Este trigger tiene como objetivo prevenir la inserción de un pago en la tabla pagos si la orden asociada está en estado "cancelada". Si el estado de la orden es "cancelada", el pago no debería ser procesado.

Acción: Insert