**Coder House**

**Curso en SQL**

## “Entrega No2.- Vistas-Funciones- Procedimientos almacenados y Triggers

## Jhinner Alejandro Patiño Pasquel

1. **Descripción general de la base de datos:**

La base de datos modela un sistema para una empresa de fabricación de hardware especializado para computo Gamer. Está organizada en cinco entidades principales: Cliente, Tienda, Producto, Centro de distribución y Venta. Estas entidades representan los diferentes componentes del sistema y las relaciones entre ellas.

1. **Descripción de las tablas o campos:**
2. **cliente**:
   * **Campos**: código cliente (PK), nombre, teléfono, tipo (persona o empresa), email,Código\_tienda(FK).
   * **Grado de la tabla**: 6 (número de campos).
   * **Descripción**: Representa a los clientes de la empresa de Hardware. La clave primaria es el código cliente, ya que es único para cada cliente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre campo** | **Tipo dato** | **Llave primaria** | **Índice secundario** | **Llave foránea** |
| **cedula** | int | Sí | No | No |
| **nombre** | text | No | Sí | No |
| **teléfono** | varchar | No | Sí | No |
| **tipo** | enum | No | No | No |
| **email** | varchar | No | No | No |
| **codigo\_tienda** | int | No | No | Sí (Tiendas) |

1. **tienda**:
   * **Atributos**: código de tienda (PK), ciudad, tipología (física u online), tamaño en m² (0 si es online), Numero de la fábrica(FK).
   * **Grado**: 5
   * **Descripción**: Representa las tiendas físicas o virtuales de la empresa. La clave primaria es el código de tienda.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre campo** | **Tipo dato** | **Llave primaria** | **Índice secundario** | **Llave foránea** |
| **cod\_tienda** | int | Sí | No | No |
| **ciudad** | text | No | Sí | No |
| **tipologia** | enum | No | Sí | No |
| **tamano\_mt2** | numeric | No | No | No |
| **numero\_fabrica** | int | No | No | Sí (Centros de distribución) |

1. **producto**:
   * **Atributos**: código de producto (PK), nombre, precio unitario.
   * **Grado**: 3.
   * **Descripción**: Contiene la información de los hardware fabricados por la empresa. La clave primaria es el código del producto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre campo** | **Tipo dato** | **Llave primaria** | **Índice secundario** | **Llave foránea** |
| **cod\_producto** | **int** | **Sí** | **No** | **No** |
| **nombre** | **text** | **No** | **Sí** | **No** |
| **precio\_unitario** | **real** | **No** | **No** | **No** |

1. **centroDistribucion**:
   * **Atributos**: cod\_fabrica(PK), país, ciudad, número de empleados.
   * **Grado**: 4.
   * **Descripción**: Representa los centros de distribución que abastecen a las tiendas. La clave primaria es el número de fábrica.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre campo** | **Tipo dato** | **Llave primaria** | **Índice secundario** | **Llave foránea** |
| **cod\_fabrica** | int | Sí | No | No |
| **pais** | text | No | Sí | No |
| **ciudad** | text | No | Sí | No |
| **numero\_empleados** | numeric | No | No | No |

1. **venta:**
   * **Atributos**: código de venta (PK), Código\_cliente(FK), Código\_producto (FK), cantidad, precio total, fecha del pedido.
   * **Grado**: 6.
   * **Descripción**: Registra las ventas realizadas por los clientes. Es una entidad débil porque depende de las entidades **Clientes** y **Productos**. La clave primaria es el código de transacción.

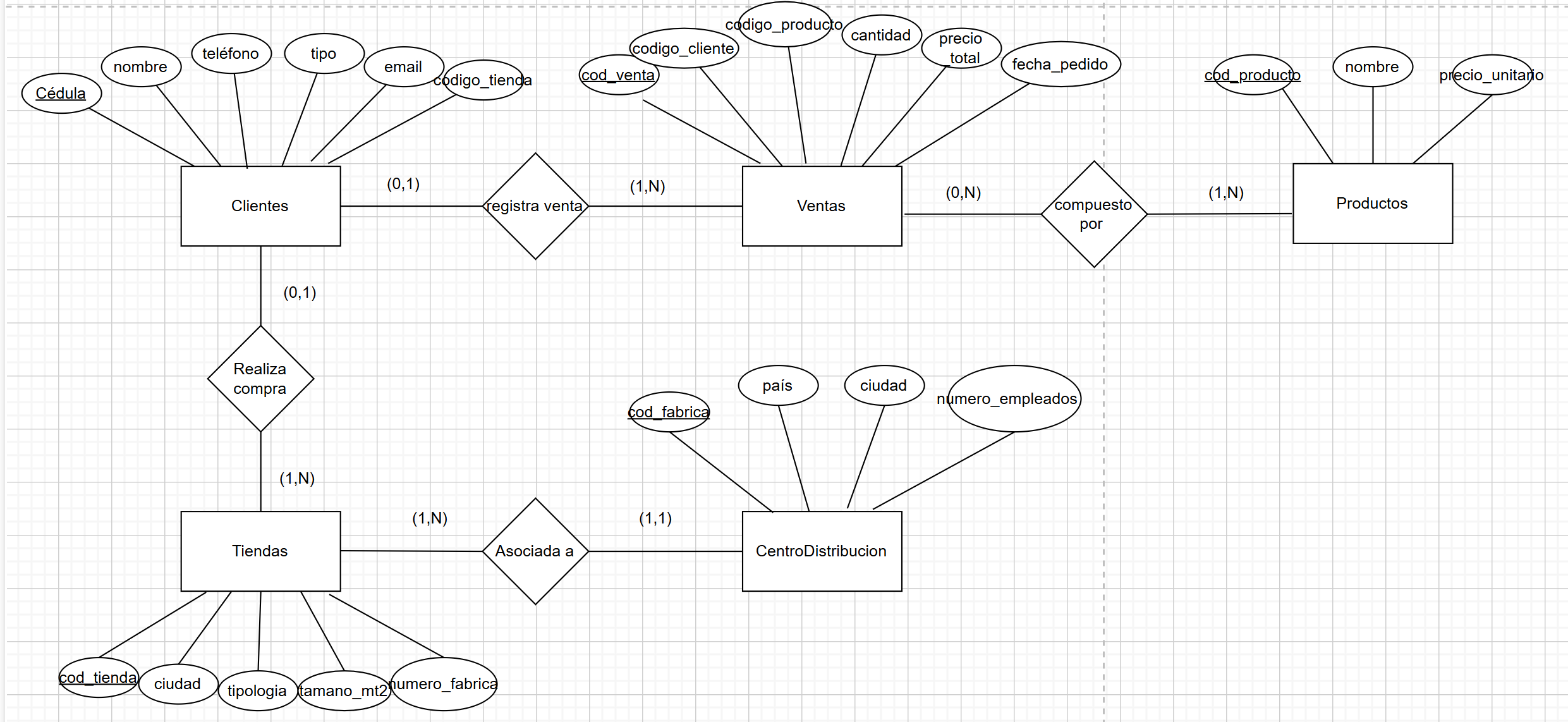
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre campo** | **Tipo dato** | **Llave primaria** | **Llave foránea** | **Índice secundario** |
| cod\_venta | int | Sí | No | No |
| codigo\_cliente | int | No | Sí (Clientes) | No |
| codigo\_producto | int | No | Sí (Productos) | No |
| cantidad | numeric | No | No | No |
| precio\_total | real | No | No | No |
| fecha\_pedido | date | No | No | No |

1. **Relaciones y claves foráneas:**

* **Relación entre Venta y Cliente**:
  + **Clave foránea**: Código\_cliente en Venta
  + Cada venta está asociada a un cliente.
* **Relación entre Venta y Producto**:
  + **Clave foránea**: Código\_producto en Venta
  + Cada venta está asociada a un producto.
* **Relación entre Cliente y Tienda**:
  + **Clave foránea**: Código\_tienda en Cliente
  + Cada cliente realiza sus compras en una tienda específica.
* **Relación entre Tienda y Centro de Distribución**:
  + **Clave foránea**: Número\_fábrica en Tienda
  + Cada tienda está asociada a un centro de distribución único.
* **Relación entre Tienda y Producto**:
  + No se requiere una clave foránea ya que todas las tiendas venden los mismos productos.
* **Relación entre Centros de Distribución y Tiendas**:
  + No hay claves foráneas directas en la tabla de Centros de Distribución, pero la relación se establece a través de la clave primaria Número\_fábrica en la tabla de Tiendas.

1. **Diagrama entidad relación (E-R):**

**Resumen del modelo de datos**

Este diseño proporciona una estructura robusta para gestionar la información de clientes, tiendas, productos, centros de distribución y ventas, con un enfoque en la integridad de los datos y la facilidad para realizar consultas relacionales.

1. **Relación entre Ventas y Productos (compuesto por):**
   * Una venta está compuesta por uno o más productos. Cada producto debe estar asociado a una o más ventas.
   * Esto significa que los productos vendidos en una transacción pueden ser variados y, a su vez, un mismo producto puede estar presente en múltiples ventas.
2. **Relación entre Ventas y Clientes (registra venta):**
   * Un cliente puede registrar como mínimo una venta (si realizó compras) y como máximo N ventas.
   * Una venta puede estar asociada a ningún cliente (cuando se trata de productos reservados o sin comprador específico) o estar asociada como máximo a un cliente que la registre.
3. **Relación entre Tiendas y CentroDistribución (Asociada a):**
   * Una tienda está asociada a un único centro de distribución, ya que cada tienda debe depender de un centro logístico que surte sus productos.
   * Un centro de distribución puede asociarse con una o más tiendas, dependiendo de su capacidad de abastecimiento.
4. **Relación entre Tiendas y Clientes (Realiza compra):**
   * Los clientes realizan compras en las tiendas. Un cliente puede realizar como mínimo una compra en una tienda y como máximo N compras.
   * Una tienda puede realizar ninguna venta (cuando no hay clientes o transacciones en un periodo) y puede realizar como máximo ventas a N clientes.
5. **Creación de Vistas:**
6. **Vista de Ventas Detalladas (vw\_ventas\_detalladas):** Esta vistacombina información de las transacciones (venta), los clientes (cliente) y los productos (producto). Esto se logra a través de sentencias JOIN que relacionan las tablas mediante claves foráneas, permitiendo obtener un reporte consolidado con campos clave como el código de la venta, la fecha, el cliente, el producto, la cantidad vendida y el total de la transacción.

Esta vista simplifica la consulta y el análisis de datos para la empresa, al evitar tener que escribir consultas complejas en cada ocasión y al garantizar que los reportes siempre reflejen la información actualizada de la base de datos.

1. **Vista de Rendimiento por Tienda (vw\_tiendas\_rendimiento):** La Vista de Rendimiento por Tienda es una herramienta que consolida información de distintas fuentes en la base de datos para evaluar el desempeño comercial de cada tienda. Al combinar datos de ventas, detalles de clientes y la relación con los centros de distribución, la vista ofrece un panorama completo del rendimiento de cada tienda, ayudando a identificar oportunidades y desafíos en el proceso de ventas y distribución. Esta vista es útil para análisis gerenciales y para la toma de decisiones estratégicas basadas en datos consolidados y actualizados. Relaciona la tabla de tiendas, centroDistribucion, cliente y venta.
2. **Vista de Compras por Cliente (vw\_clientes\_compra):** tiene como objetivo centralizar información relacionada con los clientes, mostrando datos de contacto y la tienda en la que realizan compras, junto con un resumen de su actividad comercial: la cantidad de ventas registradas y el total acumulado en compras. Esta vista facilita el análisis y la generación de reportes sobre el comportamiento de compra de los clientes. Relaciona las tablas de cliente, tienda y venta.
3. **Procedimiento SP\_REGISTRAR\_CLIENTE:**

El procedimiento SP\_REGISTRAR\_CLIENTE está orientado a incorporar nuevos clientes al sistema; al recibir como parámetros el nombre, teléfono, tipo, email y el código de la tienda asociada, inserta un nuevo registro en la tabla cliente, asegurando una incorporación ordenada y validada de clientes para la gestión comercial.

1. **Creación de funciones**:
2. **Función para Calcular el Total de Compras de un Cliente (FN\_TOTAL\_COMPRAS\_CLIENTE):** Esta función recibe como parámetro el código de un cliente (cedula) y retorna el total acumulado que ese cliente ha gastado en ventas. Es útil para conocer el comportamiento de compra de cada cliente, identificar a los clientes más valiosos y apoyar en análisis financieros o estrategias de fidelización.
3. **Función para Obtener la Cantidad Total Vendida de un Producto (FN\_CANTIDAD\_VENDIDA\_PRODUCTO):** Esta función toma como parámetro el código de un producto y devuelve la suma total de la cantidad vendida para ese producto, según la información registrada en la tabla venta. Es ideal para evaluar el desempeño de cada producto en el mercado**.**
4. **Función para Obtener el Monto de la Venta Más Alta (FN\_VENTA\_MAYOR):** tiene como objetivo determinar y devolver el monto de la ventamás alta registrada en la tabla venta, lo cual permite identificar la transacción de mayor impacto financiero en el sistema; para lograr esto, la función se declara sin parámetros y especifica un valor de retorno de tipo real, declarando internamente una variable local (max\_venta) en la que se almacena el resultado de una consulta que utiliza la función agregada MAX sobre el campo precio\_total de la tabla venta, envuelta en la función COALESCE para asegurar que, en caso de no existir registros, se retorne un 0 en lugar de un valor NULL, y finalmente, tras ejecutar la consulta y asignar el resultado a la variable, la función retorna dicho valor, consolidando en un único bloque todo el proceso de extracción y cálculo del monto de la venta más alta.
5. **Función para Contar el Número de Clientes por Tienda (FN\_NUMERO\_CLIENTES\_POR\_TIENDA):** tiene como objetivo contar y retornar el número total de clientes asociados a una tienda específica; para ello, recibe como parámetro el código de la tienda, declara internamente una variable para almacenar el conteo, y ejecuta una consulta que utiliza la función de agregación COUNT en la tabla cliente, filtrando los registros donde el campo de clave foránea que asocia al cliente con la tienda coincide con el código proporcionado; finalmente, la función devuelve el resultado obtenido, permitiendo conocer de manera rápida y precisa la cantidad de clientes asignados a esa tienda.
6. **Creación de Procedimientos Almacenados:**
7. **ProcedimientoSP\_REGISTRAR\_VENTA:**

El procedimiento SP\_REGISTRAR\_VENTA está diseñado para registrar una nueva venta en la base de datos de manera automática , ya que recibe como parámetros el código de venta, código del cliente, el código del producto, la cantidad vendida y la fecha del pedido; internamente, declara variables locales para almacenar el precio unitario del producto y el total de la venta, luego consulta en la tabla producto para obtener el precio unitario correspondiente al código de producto proporcionado, calcula el valor total multiplicando dicha cantidad por el precio unitario, y finalmente inserta un nuevo registro en la tabla venta utilizando todos estos datos, asegurando que cada transacción se almacene con la información precisa y en conformidad con las relaciones definidas en la base de datos. Así:

* Se obtiene el precio unitario del producto desde la tabla producto.
* Se calcula el precio total multiplicando la cantidad por el precio unitario.
* Se inserta la nueva venta en la tabla venta con los datos recibidos y calculados.

1. **Procedimiento SP\_ACTUALIZAR\_PRECIO\_PRODUCTO:** Es un componente clave para la gestión dinámica de precios en la base de datos, ya que permite modificar de manera centralizada el precio unitario de un producto en la tabla producto. Este procedimiento recibe dos parámetros de entrada: uno que identifica de forma única al producto mediante su código (p\_cod\_producto) y otro que especifica el nuevo valor a asignar (p\_nuevo\_precio), lo cual posibilita que, ante cambios en los costos de producción, promociones o ajustes de mercado, se actualice el precio sin tener que reescribir consultas SQL manualmente. Dentro de su estructura, se ejecuta una sentencia UPDATE que altera el campo precio\_unitario del registro correspondiente al código de producto proporcionado, garantizando la integridad de los datos mediante la verificación de la condición en el WHERE. Al agrupar esta operación en un procedimiento almacenado, se asegura la consistencia en la actualización de la información, facilitando su mantenimiento y reutilización en diversas partes de la aplicación o durante procesos batch.
2. **Procedimiento SP\_INFORME\_PRODUCTO:** está diseñado para generar un informe detallado del rendimiento de ventas de un producto específico, recibiendo el código del producto como parámetro de entrada y utilizando parámetros de salida para retornar el total de unidades vendidas, el ingreso total acumulado y el precio promedio ponderado de venta; internamente, el procedimiento ejecuta una consulta en la tabla venta que suma la cantidad y el monto de ventas asociados al producto, almacena estos resultados en variables temporales, y luego asigna esos valores a los parámetros de salida, calculando el precio promedio solo si la cantidad total es mayor a cero, lo que permite obtener de forma rápida y consolidada una visión cuantitativa del desempeño comercial del producto para apoyar decisiones estratégicas
3. **Triggers:**
4. **TRG\_LOG\_cliente:** Su objetivo es únicamente registrar en la tabla LOG\_AUDITORIA que se realizó una inserción en la tabla cliente, guardando la acción ('INSERT'), el nombre de la tabla ('cliente'), el usuario que ejecutó la acción (CURRENT\_USER()) y la fecha y hora (NOW()).
5. **TRG\_LOG\_PROVINCIA\_2**: Se activa después de cada inserción en la tabla cliente de la base de datos EmpresaHardware; para cada nuevo registro (accesible mediante la palabra clave NEW), concatena los valores de sus campos relevantes (como cod\_cliente, nombre, telefono, tipo, email y codigo\_tienda) en una cadena única que se inserta en el campo CAMPONUEVO\_CAMPOANTERIOR de la tabla LOG\_AUDITORIA\_2, junto con información adicional: la acción realizada (INSERT), el nombre de la tabla afectada (cliente), el usuario actual que ejecuta la operación (CURRENT\_USER()), y la fecha y hora de la inserción (NOW()); de esta manera, se registra automáticamente un log detallado para auditoría cada vez que se inserta un nuevo cliente en la tabla.