**Entrega inicial SQL avanzado – Coder House / LUNES 10 MAYO PRE \_ENTREGA FINAL**

**Sebastián\_ Garrido\_ Cood / Comisión \_57175**

**Nombre: Sebastián Garrido Cood**

**Temática**

La temática de esta base de datos se centra en analizar la rentabilidad de las sucursales de la empresa Patagonia enfocándose en la relación entre ubicación geográfica y desempeño financiero.

**Objetivo del proyecto**

El objetivo principal es entender cómo factores como la ubicación de las sucursales influyen en sus ingresos, gastos y utilidades, proporcionando información clave para la toma de decisiones estratégicas.

La base de datos contendrá información detallada sobre cada sucursal de Patagonia, incluyendo su nombre, dirección, ciudad, región correspondiente. Estos datos permitirán segmentar el análisis por ubicación, identificando patrones y tendencias específicas para cada área geográfica donde opera la empresa.

Además de los datos de ubicación, se registrarán las ventas realizadas en cada sucursal, detallando el monto de cada transacción y la fecha correspondiente. Esto permitirá calcular los ingresos totales generados por cada sucursal en un período determinado, así como analizar la evolución de las ventas a lo largo del tiempo.

**Modelo entidad relación**

**Diagrama

Descripción generada automáticamente**

**Diagrama entidad relacion EER**

Diagrama

Descripción generada automáticamente

Aquí está el listado de las tablas que comprenden la base de datos, con descripción de cada tabla, listado de campos, nombres completos de campos, tipos de datos y tipo de clave (foránea, primaria):

Por supuesto, aquí está la descripción de cada tabla en la base de datos de la empresa Patagonia sin los asteriscos:

1. Tabla ventas:

- id\_venta (Clave Primaria): Identificador único de cada venta.

- id\_documento (Clave Foránea): Referencia al documento asociado a la venta.

- id\_sucursal (Clave Foránea): Referencia a la sucursal donde se realizó la venta.

- id\_estado\_documento (Clave Foránea): Referencia al estado del documento relacionado.

- id\_vendedor (Clave Foránea): Referencia al vendedor que realizó la venta.

- fecha\_venta: Fecha en la que se realizó la venta.

- total\_venta: Monto total de la venta.

2. Tabla sucursales:

- id\_sucursal (Clave Primaria): Identificador único de cada sucursal.

- id\_ubicacion\_sucursal (Clave Foránea): Referencia a la ubicación de la sucursal.

- nombre\_sucursal: Nombre o identificación de la sucursal.

3. Tabla ubicacion\_sucursales:

- id\_ubicacion\_sucursal (Clave Primaria): Identificador único de cada ubicación de sucursal.

- region: Región geográfica donde se encuentra la sucursal.

- direccion\_sucursal: Dirección física de la sucursal.

4. Tabla vendedores:

- id\_vendedor (Clave Primaria): Identificador único de cada vendedor.

- rut\_vendedor: RUT o identificación fiscal del vendedor.

- nombre\_vendedor: Nombre completo o identificación del vendedor.

5. Tabla reclamos:

- id\_reclamos (Clave Primaria): Identificador único de cada reclamo.

- fecha\_reclamos: Fecha en la que se realizó el reclamo.

- id\_vendedor (Clave Foránea): Referencia al vendedor asociado al reclamo.

- descripcion\_reclamo: Descripción detallada del reclamo.

6. Tabla documentos:

- id\_documento (Clave Primaria): Identificador único de cada documento.

- nombre\_documento: Nombre o tipo de documento.

7. Tabla estado\_documentos:

- id\_estado\_documento (Clave Primaria): Identificador único de cada estado de documento.

- estado\_documento: Estado o condición del documento.

8. Tabla productos\_ventas:

- id\_producto\_venta (Clave Primaria): Identificador único de cada producto vendido.

- id\_producto (Clave Foránea): Referencia al producto asociado a la venta.

- tipo\_venta: Tipo de venta del producto.

9. Tabla productos:

- id\_producto (Clave Primaria): Identificador único de cada producto.

- categoria\_producto: Categoría o tipo de producto.

- genero\_producto: Género o características del producto.

- precio\_producto: Precio unitario del producto.

10. Tabla stock:

- id\_stock (Clave Primaria): Identificador único de cada registro de stock.

- id\_proveedor (Clave Foránea): Referencia al proveedor asociado al stock.

- id\_producto (Clave Foránea): Referencia al producto relacionado al stock.

- id\_promocion (Clave Foránea): Referencia a la promoción asociada al stock.

- fecha\_stock: Fecha en la que se registró el stock.

- cantidad\_stock: Cantidad de productos en stock.

11. Tabla promociones:

- id\_promocion (Clave Primaria): Identificador único de cada promoción.

- descuento\_promocion: Descripción o detalle del descuento de la promoción.

- genero\_promocion: Género o tipo de productos incluidos en la promoción.

- fecha\_promocion: Fecha de inicio o vigencia de la promoción.

12. Tabla proveedores:

- id\_proveedor (Clave Primaria): Identificador único de cada proveedor.

- nombre\_proveedor: Nombre o identificación del proveedor.

- email\_proveedor: Correo electrónico de contacto del proveedor.

- compra\_proveedor: Monto de compra o transacciones realizadas con el proveedor.

- fecha\_compra\_proveedor: Fecha de la última compra realizada al proveedor.

13. Tabla documento\_proveedor:

- id\_documento\_proveedor (Clave Primaria): Identificador único de cada documento asociado al proveedor.

- id\_proveedor (Clave Foránea): Referencia al proveedor asociado al documento.

- fecha\_emision: Fecha de emisión del documento.

- total\_factura: Monto total de la factura asociada al documento.

**VISTAS**

1. **VISTA\_FACTURAS\_DETALLADAS**

Descripción:

Esta vista ofrece un detalle de las facturas emitidas, mostrando información sobre el proveedor, la fecha de emisión y el monto total de la factura.

Objetivo:

Proporcionar una visión detallada de las facturas para análisis financiero y operativo, facilitando la revisión de transacciones específicas y el monitoreo de la actividad de facturación.

Tablas Compuestas:

- documento\_proveedor: Registra los documentos emitidos por los proveedores, con detalles como id\_documento\_proveedor, id\_proveedor, fecha\_emision, y total\_factura.

- proveedores: Contiene información sobre los proveedores, incluyendo el nombre\_proveedor.

1. **VISTA\_PROVEEDORES\_ACTIVOS**

Descripción:

Esta vista proporciona una lista de proveedores que han emitido documentos en el último año, junto con el conteo total de documentos emitidos por cada proveedor en ese período.

Objetivo:

Identificar a los proveedores activos en el último año y cuantificar su actividad en términos de documentos emitidos. Esto puede ser útil para análisis de proveedores, evaluación de desempeño y decisiones de negocios relacionadas con la gestión de proveedores.

Tablas Compuestas:

- proveedores: Contiene información sobre los proveedores, incluyendo el id\_proveedor y nombre\_proveedor.

- documento\_proveedor: Contiene los documentos emitidos por los proveedores, con campos como id\_documento\_proveedor, id\_proveedor, y fecha\_emision.

1. **VISTA\_RESUMEN\_ANUAL**

Descripción:

Esta vista resume el total facturado anualmente, agrupando por año de emisión de las facturas.

Objetivo:

Proveer un resumen anual de las ventas totales, lo cual es útil para análisis de tendencias financieras, reportes anuales, y planificación estratégica.

Tablas Compuestas:

- documento\_proveedor: Contiene los registros de las facturas emitidas, con campos como fecha\_emision y total\_factura.

1. **VISTA\_VENDEDORES\_RECLAMOS**

Descripción:

Esta vista muestra la relación entre los vendedores y los reclamos asociados a ellos, detallando información del vendedor y del reclamo.

Objetivo:

Facilitar el análisis de reclamos relacionados con vendedores específicos, permitiendo evaluar el desempeño del vendedor y gestionar problemas de servicio al cliente.

Tablas Compuestas:

- vendedores: Contiene información sobre los vendedores, incluyendo id\_vendedor, rut\_vendedor, y nombre\_vendedor.

- reclamos: Registra los reclamos, con campos como id\_reclamo, fecha\_reclamo, descripcion\_reclamo, estado\_reclamo, y id\_vendedor (relación con vendedores).

1. **VISTA\_VENTAS\_SUCURSALES**

Descripción:

Esta vista detalla las ventas por sucursal y vendedor, mostrando la suma total de ventas por cada combinación de vendedor y sucursal, junto con la información de la ubicación de la sucursal.

Objetivo:

Proporcionar un análisis de ventas desglosado por vendedor y sucursal, permitiendo identificar patrones de ventas, evaluar el desempeño de los vendedores y las sucursales, y tomar decisiones informadas sobre la gestión de ventas y la distribución de recursos.

Tablas Compuestas:

- ventas: Registra las ventas realizadas, incluyendo monto\_total, id\_vendedor, y id\_sucursal.

- vendedores: Contiene información sobre los vendedores, como nombre\_vendedor.

- sucursales: Registra las sucursales, incluyendo nombre\_sucursal y id\_ubicacion.

- ubicacion\_sucursales: Detalla la ubicación de las sucursales, con campos como region y direccion\_sucursal.

**FUNCIONES**

**Descripción de la Función promedio\_ventas\_por\_anio\_y\_region**

La función promedio\_ventas\_por\_anio\_y\_region está diseñada para calcular el promedio de ventas de un año y región específicos en la base de datos patagonia. Esta función toma dos parámetros de entrada: anio\_param, que corresponde al año de interés, y region\_param, que especifica la región geográfica. La función devuelve un valor decimal que representa el promedio de las ventas totales para la combinación de año y región proporcionada.

**Objetivo**

El objetivo principal de esta función es proporcionar una manera eficiente y precisa de calcular el promedio de ventas para un año y región determinados. Esto es útil para realizar análisis financieros y de rendimiento, permitiendo a los usuarios y analistas obtener rápidamente información agregada sobre el desempeño de ventas en distintas regiones y períodos.

**Beneficios**

Simplicidad y usabilidad: Centraliza la lógica de cálculo del promedio en una única función, permitiendo su reutilización en diferentes partes del sistema y en múltiples consultas SQL sin necesidad de repetir el código.

Mantenimiento: Facilita el mantenimiento del código, ya que cualquier cambio en la lógica de cálculo del promedio solo necesita realizarse en la función almacenada, simplificando las actualizaciones y modificaciones.

Eficiencia: Mejora la eficiencia de las operaciones de cálculo al ejecutarse directamente en el servidor de la base de datos, lo que reduce el tráfico de red y acelera el proceso de obtención de resultados.

Consistencia: Asegura que el cálculo del promedio de ventas se realice de manera coherente en todas las consultas y reportes, proporcionando resultados precisos y uniformes.

**Datos y Tablas Relacionadas**

Vista vista\_ventas\_sucursales

La función interactúa principalmente con la vista vista\_ventas\_sucursales. Esta vista contiene los registros de las ventas realizadas en distintas sucursales y sus campos incluyen el total de la venta, el año de la venta y la región de la sucursal. La estructura simplificada de la vista puede incluir los siguientes campos:

- total\_venta: Representa el monto total de cada venta.

- anio: Indica el año en que se realizó la venta.

- region: Especifica la región geográfica de la sucursal donde se realizó la venta.

**Proceso de Cálculo**

1. Declaración de Variables: La función declara una variable `promedio` de tipo decimal para almacenar el resultado del cálculo.

2. Cálculo del Promedio: Utiliza una consulta SQL para calcular el promedio de las ventas (total\_venta) para el año (anio\_param) y la región (region\_param) especificados. El resultado de esta consulta se almacena en la variable promedio.

3.Retorno del Resultado: La función devuelve el valor calculado almacenado en la variable promedio.

Esta función es esencial para análisis detallados y reportes financieros, proporcionando a los usuarios una herramienta eficiente para evaluar el desempeño de ventas en diferentes regiones y períodos. Esto, a su vez, facilita la toma de decisiones informadas y estratégicas basadas en datos precisos y agregados.

**Descripción de la Función obtener\_documentos\_por\_proveedor**

La función obtener\_documentos\_por\_proveedor está diseñada para contar el número de documentos asociados a un proveedor específico en la tabla documento\_proveedor de la base de datos `patagonia`. Esta función toma un parámetro de entrada, id\_proveedor\_parametro, que corresponde al identificador único del proveedor, y devuelve un valor entero que representa la cantidad de documentos relacionados con ese proveedor.

**Objetivo**

El objetivo principal de esta función es proporcionar una manera eficiente y precisa de contar el número de documentos asociados a un proveedor específico. Esto facilita el análisis y la gestión de la información de los proveedores, permitiendo a los usuarios obtener rápidamente el número de documentos emitidos por cada proveedor.

**Beneficios**

Simplicidad y usabilidad: Centraliza la lógica de conteo en una única función, permitiendo su reutilización en diferentes partes del sistema y en múltiples consultas SQL sin necesidad de repetir el código.

Mantenimiento: Facilita el mantenimiento del código, ya que cualquier cambio en la lógica de conteo solo necesita realizarse en la función almacenada, simplificando las actualizaciones y modificaciones.

Eficiencia: Mejora la eficiencia de las operaciones de conteo al ejecutarse directamente en el servidor de la base de datos, lo que reduce el tráfico de red y acelera el proceso de obtención de resultados.

Consistencia: Asegura que el conteo de documentos se realice de manera coherente en todas las consultas y reportes, proporcionando resultados precisos y uniformes.

**Datos y Tablas Relacionadas**

Tabla documento\_proveedor

La función interactúa principalmente con la tabla documento\_proveedor. Esta tabla contiene los registros de los documentos emitidos por los proveedores y sus campos incluyen:

- `id\_documento\_proveedor`: Identificador único del documento.

- `id\_proveedor`: Identificador del proveedor asociado al documento.

- `fecha\_emision`: Fecha de emisión del documento.

- `total\_factura`: Monto total de la factura.

**Proceso de Cálculo**

1.Declaración de Variables: La función declara una variable `total` de tipo entero para almacenar el resultado del conteo.

2. Conteo de Documentos: Utiliza una consulta SQL para contar el número de documentos COUNT(\*) donde el id\_proveedor coincida con el parámetro de entrada id\_proveedor\_parametro. El resultado de esta consulta se almacena en la variable total.

3. Retorno del Resultado: La función devuelve el valor almacenado en la variable `total`, que representa el número de documentos asociados al proveedor especificado.

**Descripción del Procedimiento Almacenado obtener\_facturas\_por\_fecha**

El procedimiento almacenado obtener\_facturas\_por\_fecha está diseñado para obtener todas las facturas emitidas dentro de un rango de fechas específico en la tabla documento\_proveedor en la base de datos patagonia. Este procedimiento toma dos parámetros de entrada: fecha\_inicio, que corresponde a la fecha inicial del rango, y fecha\_fin, que es la fecha final del rango.

**Objetivo**

El objetivo principal de este procedimiento es proporcionar una manera eficiente y precisa de recuperar todas las facturas emitidas entre dos fechas específicas. Esto facilita la gestión y el análisis de los datos de facturación, permitiendo a los usuarios obtener rápidamente información relevante sobre las facturas emitidas en un período determinado.

**Beneficios**

Simplicidad y usabilidad: Centraliza la lógica de consulta en un solo lugar, permitiendo su reutilización en diferentes partes del sistema sin necesidad de repetir el código SQL.

Mantenimiento: Facilita el mantenimiento del código, ya que cualquier cambio en la lógica de consulta solo necesita realizarse en el procedimiento almacenado, simplificando las actualizaciones y modificaciones.

Eficiencia: Mejora la eficiencia de las operaciones de consulta al ejecutarse directamente en el servidor de la base de datos, lo que reduce el tráfico de red y acelera el proceso de recuperación de datos.

Flexibilidad: Permite a los usuarios especificar un rango de fechas personalizado, proporcionando una herramienta flexible para la obtención de datos específicos de facturación según las necesidades del análisis.

**Datos y Tablas Relacionadas**

Tabla documento\_proveedor

La tabla documento\_proveedor es la principal afectada por este procedimiento. Contiene los registros de las facturas emitidas por los proveedores y sus campos incluyen:

- id\_documento\_proveedor: Identificador único del documento.

- id\_proveedor: Identificador del proveedor asociado al documento.

- fecha\_emision: Fecha de emisión del documento.

- total\_factura: Monto total de la factura.

**STORED PRCEDURES**

**Descripción del Stored Procedure \_eliminar\_documento\_proveedor**

El procedimiento almacenado eliminar\_documento\_proveedor está diseñado para eliminar un registro específico de la tabla documento\_proveedor en la base de datos patagonia. Este procedimiento toma un parámetro de entrada, id\_documento, que corresponde al identificador único del documento que se desea eliminar.

**Objetivo**

El objetivo principal de este procedimiento es proporcionar una manera eficiente y segura de eliminar registros específicos de la tabla documento\_proveedor mediante el uso de un identificador único. Esto facilita la gestión de los documentos de proveedores y asegura que las eliminaciones se realicen de manera controlada y coherente.

**Beneficios**

Simplicidad y usabilidad: Centraliza la lógica de eliminación en un solo lugar, permitiendo su reutilización en diferentes partes del sistema sin necesidad de repetir el código SQL.

Mantenimiento: Facilita el mantenimiento del código, ya que cualquier cambio en la lógica de eliminación solo necesita realizarse en el procedimiento almacenado, simplificando las actualizaciones y modificaciones.

Seguridad: Reduce el riesgo de errores en la eliminación de datos al encapsular la lógica en un procedimiento seguro. Esto asegura que solo se eliminen los registros cuando se cumplan las condiciones especificadas.

Eficiencia: Mejora la eficiencia de las operaciones de eliminación al ejecutarse directamente en el servidor de la base de datos, lo que reduce el tráfico de red y acelera el proceso de eliminación.

**Tablas Relacionadas**

**Tabla\_ documento\_proveedor**

La tabla documento\_proveedor es la principal afectada por este procedimiento. Contiene los registros de los documentos de proveedores y sus campos incluyen el identificador del documento, el identificador del proveedor, la fecha de emisión y el total de la factura.

**Tabla \_proveedor**

Esta tabla contiene información sobre los proveedores y se relaciona con documento\_proveedor a través del campo id\_proveedor. Al eliminar un documento de proveedor, es importante asegurar que la eliminación no afecte negativamente las relaciones de integridad referencial con la tabla `proveedor`.

**Interacción con Otras Tablas**

Este procedimiento puede afectar indirectamente otras tablas que tengan relaciones de clave foránea con documento\_proveedor. Si existen tablas que referencian id\_documento\_proveedor como clave foránea, es crucial revisar las restricciones de integridad referencial antes de eliminar un registro para evitar problemas o datos huérfanos.

**Descripción del Stored Procedure actualizar\_proveedor**

El procedimiento almacenado actualizar\_proveedor está diseñado para actualizar el nombre de un proveedor específico en la tabla proveedores en la base de datos patagonia. Este procedimiento toma dos parámetros de entrada: id\_proveedor\_parametro, que corresponde al identificador único del proveedor cuyo nombre se desea actualizar, y nuevo\_nombre, que es el nuevo nombre del proveedor.

**Objetivo**

El objetivo principal de este procedimiento es proporcionar una manera eficiente y segura de actualizar el nombre de un proveedor específico en la tabla proveedores. Esto facilita la gestión de los datos de los proveedores y asegura que las actualizaciones se realicen de manera controlada y coherente.

**Beneficios**

Al igual que el procedimiento anterior la lógica del procedimiento genera los mismos beneficios.

Simplicidad y usabilidad: Centraliza la lógica de actualización en un solo lugar, permitiendo su reutilización en diferentes partes del sistema sin necesidad de repetir el código SQL.

Mantenimiento: Facilita el mantenimiento del código, ya que cualquier cambio en la lógica de actualización solo necesita realizarse en el procedimiento almacenado, simplificando las actualizaciones y modificaciones.

Seguridad: Reduce el riesgo de errores en la actualización de datos al encapsular la lógica en un procedimiento seguro. Esto asegura que solo se actualicen los registros cuando se cumplan las condiciones especificadas.

Eficiencia: Mejora la eficiencia de las operaciones de actualización al ejecutarse directamente en el servidor de la base de datos, lo que reduce el tráfico de red y acelera el proceso de actualización.

**Tablas Relacionadas**

Tabla proveedores

La tabla proveedores es la principal afectada por este procedimiento. Contiene los registros de los proveedores y sus campos incluyen el identificador del proveedor y el nombre del proveedor.

**Descripción del Stored Procedure obtener\_facturas\_por\_fecha**

El procedimiento almacenado obtener\_facturas\_por\_fecha está diseñado para obtener todas las facturas emitidas dentro de un rango de fechas específico en la tabla documento\_proveedor en la base de datos patagonia. Este procedimiento toma dos parámetros de entrada: fecha\_inicio, que corresponde a la fecha inicial del rango, y fecha\_fin, que es la fecha final del rango.

**Objetivo**

El objetivo principal de este procedimiento es proporcionar una manera eficiente y precisa de recuperar todas las facturas emitidas entre dos fechas específicas. Esto facilita la gestión y el análisis de los datos de facturación, permitiendo a los usuarios obtener rápidamente información relevante sobre las facturas emitidas en un período determinado.

**Beneficios**

Simplicidad y Reusabilidad: Centraliza la lógica de consulta en un solo lugar, permitiendo su reutilización en diferentes partes del sistema sin necesidad de repetir el código SQL.

Mantenimiento: Facilita el mantenimiento del código, ya que cualquier cambio en la lógica de consulta solo necesita realizarse en el procedimiento almacenado, simplificando las actualizaciones y modificaciones.

Eficiencia: Mejora la eficiencia de las operaciones de consulta al ejecutarse directamente en el servidor de la base de datos, lo que reduce el tráfico de red y acelera el proceso de recuperación de datos.

Flexibilidad: Permite a los usuarios especificar un rango de fechas personalizado, proporcionando una herramienta flexible para la obtención de datos específicos de facturación según las necesidades del análisis.

**Tablas Relacionadas**

Tabla documento\_proveedor

La tabla documento\_proveedor es la principal afectada por este procedimiento. Contiene los registros de las facturas emitidas por los proveedores y sus campos incluyen el identificador del documento, el identificador del proveedor, la fecha de emisión y el total de la factura.

Interacción con Otras Tablas

Este procedimiento interactúa principalmente con la tabla documento\_proveedor. Sin embargo, los datos recuperados pueden ser utilizados en combinación con otras tablas para análisis más profundos, como la relación con la tabla proveedores para obtener detalles adicionales sobre los proveedores que emitieron las facturas en el rango de fechas especificado. Es importante asegurar que las consultas a la tabla documento\_proveedor mantengan la integridad y coherencia de los datos en el sistema.

**Descripción Detallada de Cada Trigger**

1. Trigger after\_update\_ventas

**Objetivo**

El objetivo de este trigger es registrar cualquier actualización en la tabla `ventas`. Cuando se modifica una fila en esta tabla, el trigger compara los valores antiguos y nuevos de ciertas columnas clave. Si detecta un cambio, inserta un registro en la tabla `auditoria\_ventas` con los detalles del cambio.

**Beneficio**

Proporciona un historial detallado de las modificaciones en la tabla `ventas`, facilitando auditorías y seguimiento de cambios. Esto es crucial para mantener la integridad de los datos y realizar análisis históricos de las transacciones.

**Interacciones**

-Tabla ventas:

La tabla donde ocurren las actualizaciones.

- Tabla auditoria\_ventas: La tabla que almacena los registros de auditoría de las actualizaciones.

2. Trigger before\_insert\_ventas

**Objetivo**

El objetivo de este trigger es prevenir la inserción de registros con un `total\_venta` negativo en la tabla `ventas`. Antes de insertar una nueva fila, el trigger verifica el valor de `total\_venta` y, si es negativo, lanza un error y bloquea la inserción.

**Beneficio**

Garantiza que no se ingresen datos inválidos en la tabla `ventas`, manteniendo la integridad y coherencia de los datos financieros.

**Interacciones**

- Tabla ventas: La tabla donde se intenta insertar el nuevo registro.

3. Trigger after\_insert\_ventas

**Objetivo**

El objetivo de este trigger es registrar cada inserción en la tabla `ventas`. Después de que se inserte una nueva fila, el trigger crea un registro en la tabla `auditoria\_ventas` con los detalles de la inserción.

**Beneficio**

Facilita el seguimiento y la auditoría de todas las nuevas ventas registradas, proporcionando un historial claro de las inserciones.

**Interacciones**

- Tabla ventas: La tabla donde se inserta el nuevo registro.

- Tabla auditoria\_ventas: La tabla que almacena los registros de auditoría de las inserciones.

4. Trigger after\_delete\_ventas

**Objetivo**

El objetivo de este trigger es registrar cada eliminación en la tabla `ventas`. Después de que se elimine una fila, el trigger crea un registro en la tabla `auditoria\_ventas` con los detalles de la eliminación.

**Beneficio**

Proporciona un historial de todas las eliminaciones realizadas en la tabla `ventas`, lo que es útil para auditorías y para comprender el historial completo de transacciones.

**Interacciones**

- Tabla ventas: La tabla de la cual se elimina el registro.

- Tabla `auditoria\_ventas`: La tabla que almacena los registros de auditoría de las eliminaciones.

**Conclusión**

Cada uno de estos triggers aporta un valor significativo al proyecto, asegurando la integridad y el seguimiento detallado de los datos en la tabla `ventas`. Al registrar todas las actualizaciones, inserciones y eliminaciones, se facilita la auditoría y se garantiza que los datos sean precisos y consistentes.