**Universidad Católica Boliviana “San Pablo”**

**“Base de datos de Gimnasio”**



**Facultad:** Facultad de Ciencias Exactas

**Materia:** Base De Datos Il

**Docente:** Marin Salazar Carmen Rosa

**Integrantes:**

* Leandro Joe Jaldin Gutierrez
* Camila Catorceno Orellana

1. **INTRODUCCION**

**Descripción del Problema**

El gimnasio Premier se enfrenta al reto de gestionar de manera eficiente y precisa la información relacionada con sus operaciones diarias. Con una creciente cantidad de clientes y una diversidad de servicios, es esencial contar con un sistema robusto de gestión de datos. El gimnasio ofrece cuatro servicios principales: pesas, zumba, cardio y box, cada uno con horarios específicos y entrenadores dedicados. Además, cada cliente paga una mensualidad fija de 250 Bs.

El sistema debe ser capaz de registrar y mantener la información de los clientes, incluyendo sus datos personales y número de teléfono. También debe administrar los servicios ofrecidos, detallando horarios y responsables. La gestión de inscripciones es crucial, permitiendo registrar el tiempo de suscripción de cada cliente, el costo y calculando automáticamente el total a pagar. Además, el sistema debe llevar un registro detallado de los pagos realizados por los clientes, controlar las asistencias a los diferentes servicios y gestionar la información de los entrenadores.

**Propuesta de Solución**

Nosotros proponemos una solución integral basada en una base de datos relacional que permitirá al Gimnasio Premier gestionar toda la información de manera eficiente y segura. La base de datos diseñada cumplirá con los siguientes objetivos:

**Gestión de Clientes:** Registrar y mantener la información de los clientes, garantizando la unicidad y evitando duplicaciones.

**Administración de Servicios:** Gestionar los detalles de los servicios ofrecidos, incluyendo horarios y entrenadores responsables.

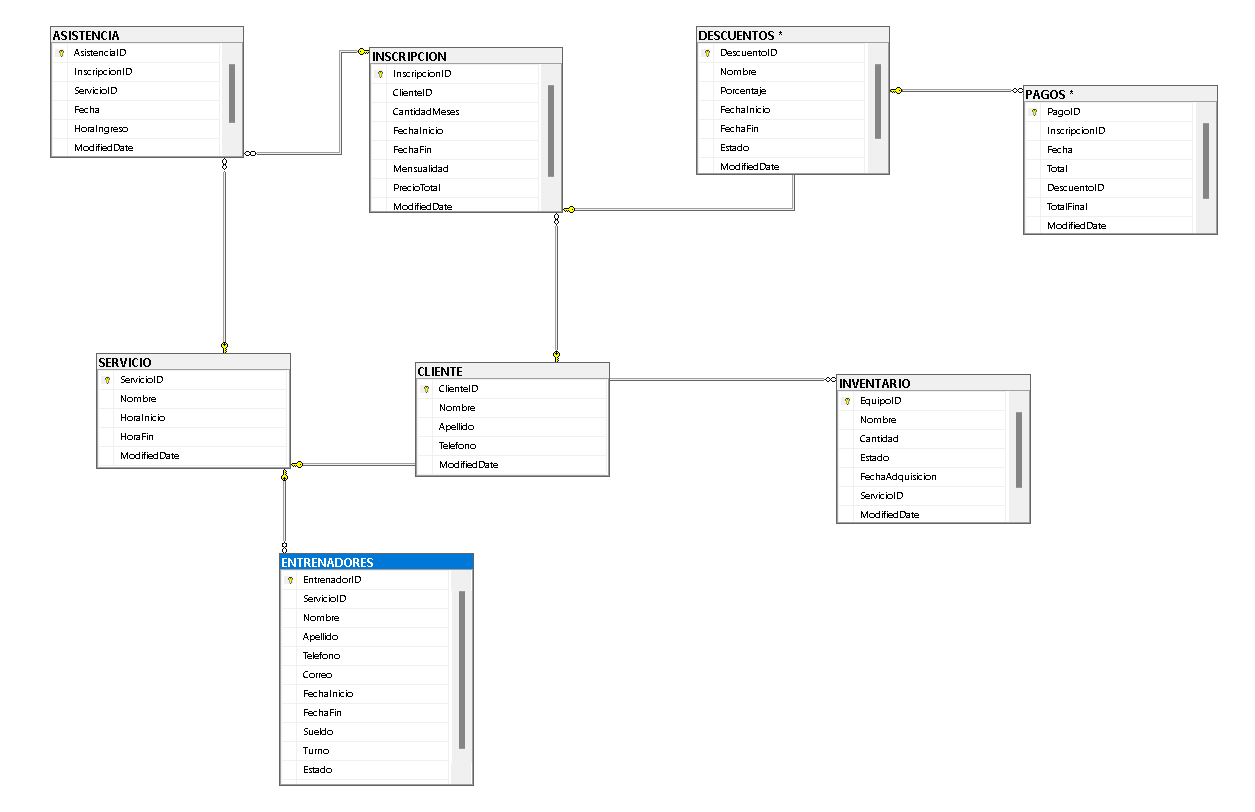
**Registro de Inscripciones:** Permitir la inscripción de clientes, calculando automáticamente las fechas de inicio y fin, así como el total a pagar basado en la duración de la inscripción.

**Control de Pagos:** Mantener un registro detallado de los pagos realizados por los clientes, asegurando precisión en las transacciones.

**Seguimiento de Asistencias:** Rastrear las asistencias de los clientes a los diferentes servicios, asegurando que no se registren asistencias duplicadas.

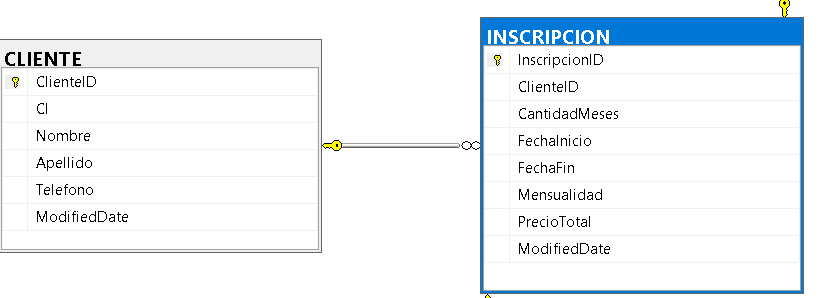
**Gestión de Entrenadores:** Administrar la información de los entrenadores, incluyendo sus datos personales, servicios impartidos y estado de empleo.

Esta solución proporcionará al gimnasio las herramientas necesarias para mejorar la eficiencia operativa, reducir errores y ofrecer un mejor servicio a sus clientes.

1. **Diseño**

**Entidades relaciones y atributos**

**1.Tabla CLIENTE**

****

**ClienteID : Identificador único de cada cliente**

**Ci: Es el carnet de identidad del cliente**

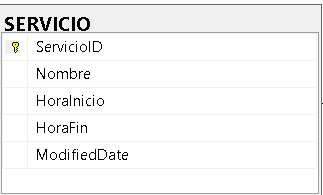
**Nombre: Nombre del cliente**

**Apellido: Apellido del cliente**

**Telefono: Numero de teléfono del cliente, este debe ser único**

**ModifieDate: Fecha de la ultima modificación del registro**

1. **Tabla SERVICIO**

****

**ServicioID: Identificador único de cada servicio**

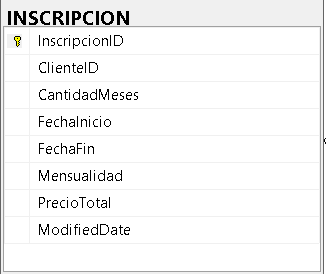
**Nombre: Nombre del servicio**

**HoraInicio: Hora que inicia el servicio**

**HoraFin: Hora que finaliza el servicio**

**ModifeDate: Fecha de la última modificación del registro**

1. **Tabla INSCRIPCION**



**InscripcionID: Identificador único de la inscripción**

**ClienteID: Referencia al cliente que realiza la inscripción**

**CantidadMeses: Numero de meses de la inscripción (por cuantos meses entrenara)**

**FechaInicio: Fecha de inicio de la inscripción**

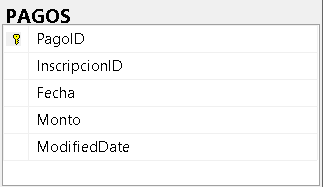
**FechaFin: Fecha de finalización de la inscripción**

**Mensualidad: Monto mensual de la inscripción**

**PrecioTotal: Precio total de la inscripción**

**ModifieDate: fecha de la última modificación del registro**

1. **Tabla PAGOS**

****

**PagoID: Identificador Unico de cada Pago**

**InscripcionID: Referencia al cliente que realiza la inscripción**

**Fecha: Fecha de pago**

**Monto: Monto pagado**

**ModifeDate: Fecha de ultima modificación del registro**

1. **Tabla Asistencia**

****

**AsistenciaID: Identificador único de cada asistencia**

**Inscripcion: Referencia a la inscripción del cliente**

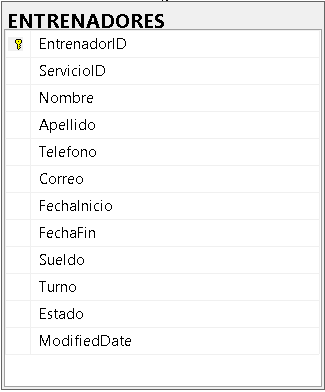
**ServicioID: Referencia al servicio que asiste al cliente**

**Fecha: Fecha de asistencia**

**HoraIngreso: Hora de ingreso del cliente**

**ModifieDate: Fecha de la ultima modificación del registro**

1. **Tabla ENTRENADORES**



**EntrenadorID: Identificador único de cada entrenador.**

**ServicioID: Referencia al servicio que imparte el entrenador.**

**Nombre : Nombre del entrenador.**

**Apellido: Apellido del entrenador.**

**Telefono: Número de teléfono del entrenador.**

**Correo: Correo electrónico del entrenador.**

**FechaInicio: Fecha de inicio del contrato del entrenador.**

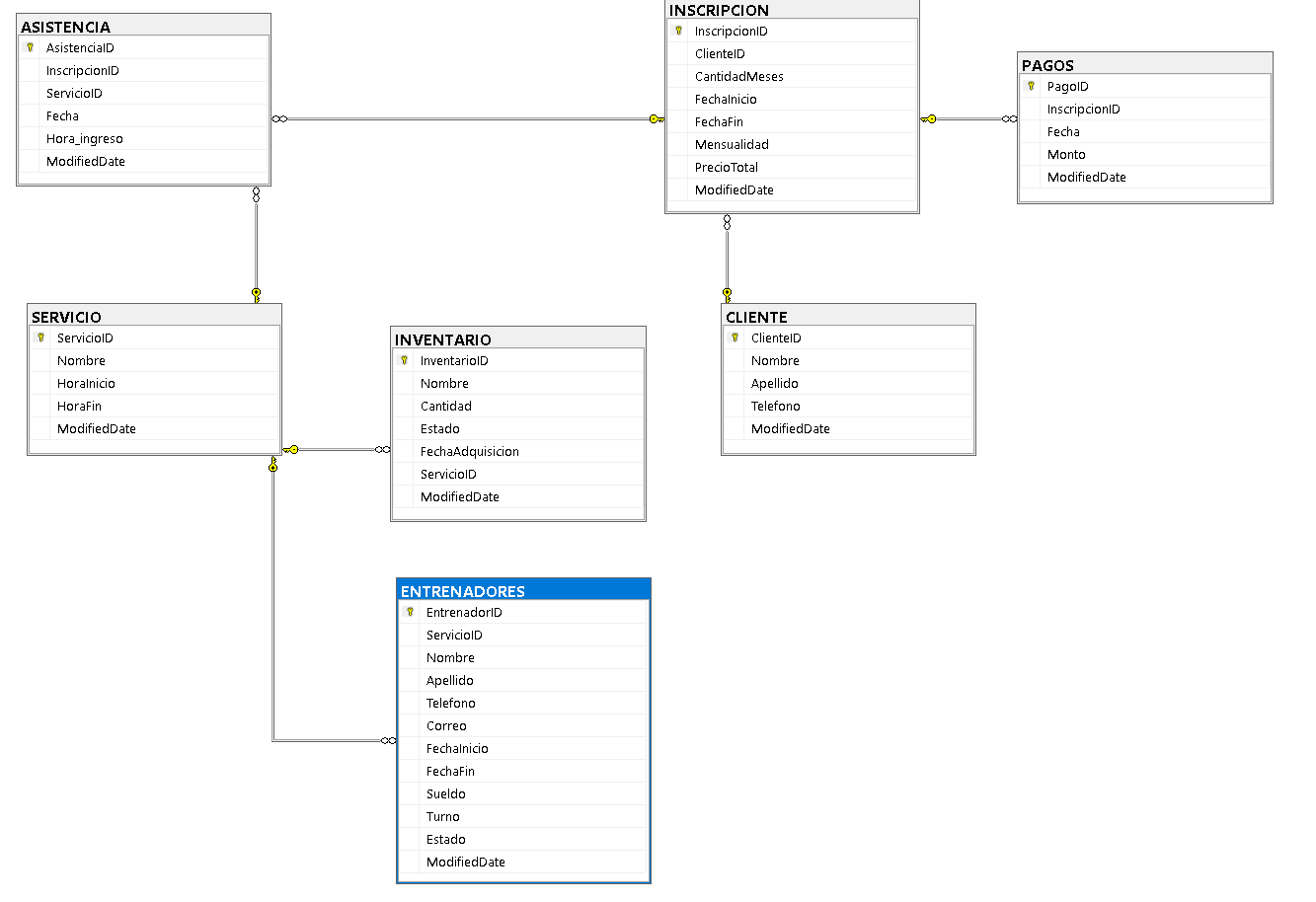
**FechaFin: Fecha de finalización del contrato del entrenador, si aplica.**

**Sueldo: Sueldo del entrenador.**

**Turno: Turno del entrenador (mañana, tarde, noche).**

**Estado: Estado del entrenador (Activo o Despedido), calculado basado en la fecha de finalización.**

**ModifiedDate: Fecha de la última modificación del registro.**

1. **Tabla de inventario**

**Inventarioid – identificador inico de cada maquina**

**Nombre - Nombre de las maquinas del gimnasio**

**Cantidad – El numero de maquinas**

**Estado – el uso y el mantenimiento de cada maquina**

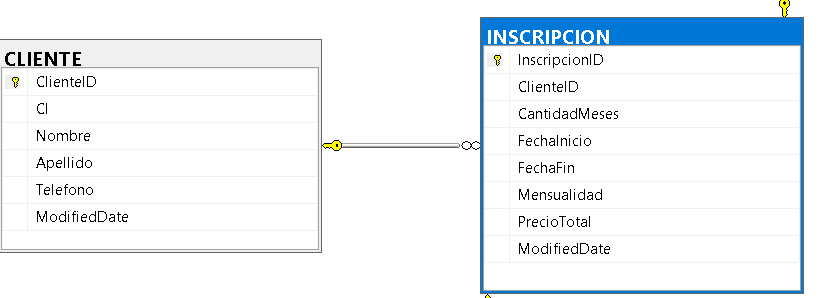
**Fecha de adquisición - la fecha en la que se compro la maquina**

**Servicioid – al servicio donde pertenecen**

**Modifedate – fecha de la ultima modificación de cada registro**

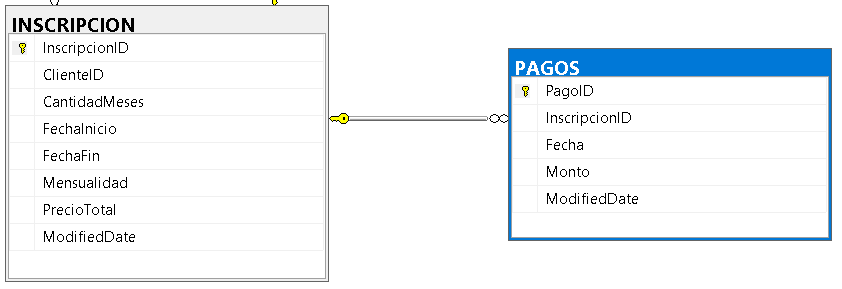
**Relaciones entre las entidades**

1. **Cliente a Inscripciones**

****

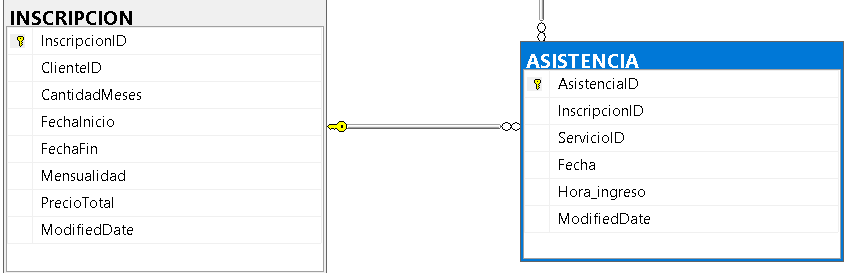
**Un cliente puede tener múltiples inscripciones**

1. **Inscripción a pagos**



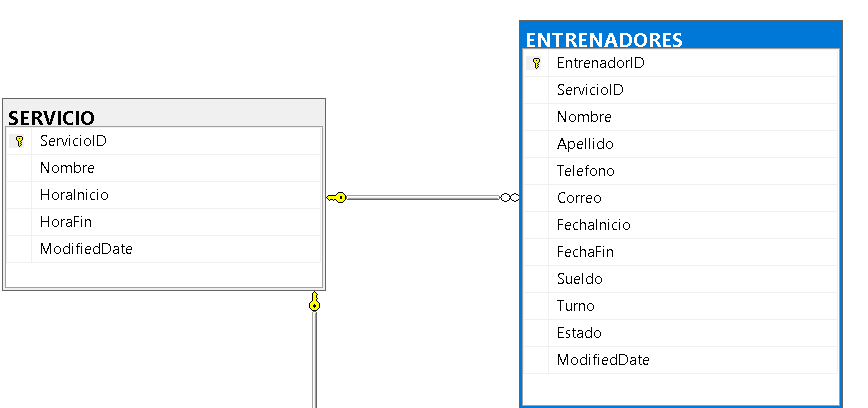
**Una inscripción puede tener múltiples pagos**

1. **Inscripción a Asistencia**



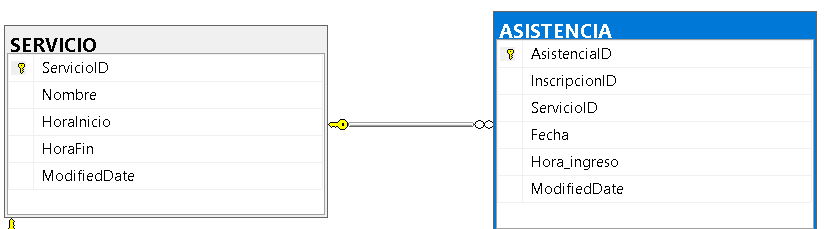
**Una inscripción puede tener múltiples asistencias**

1. **Servicio a Entrenadores**



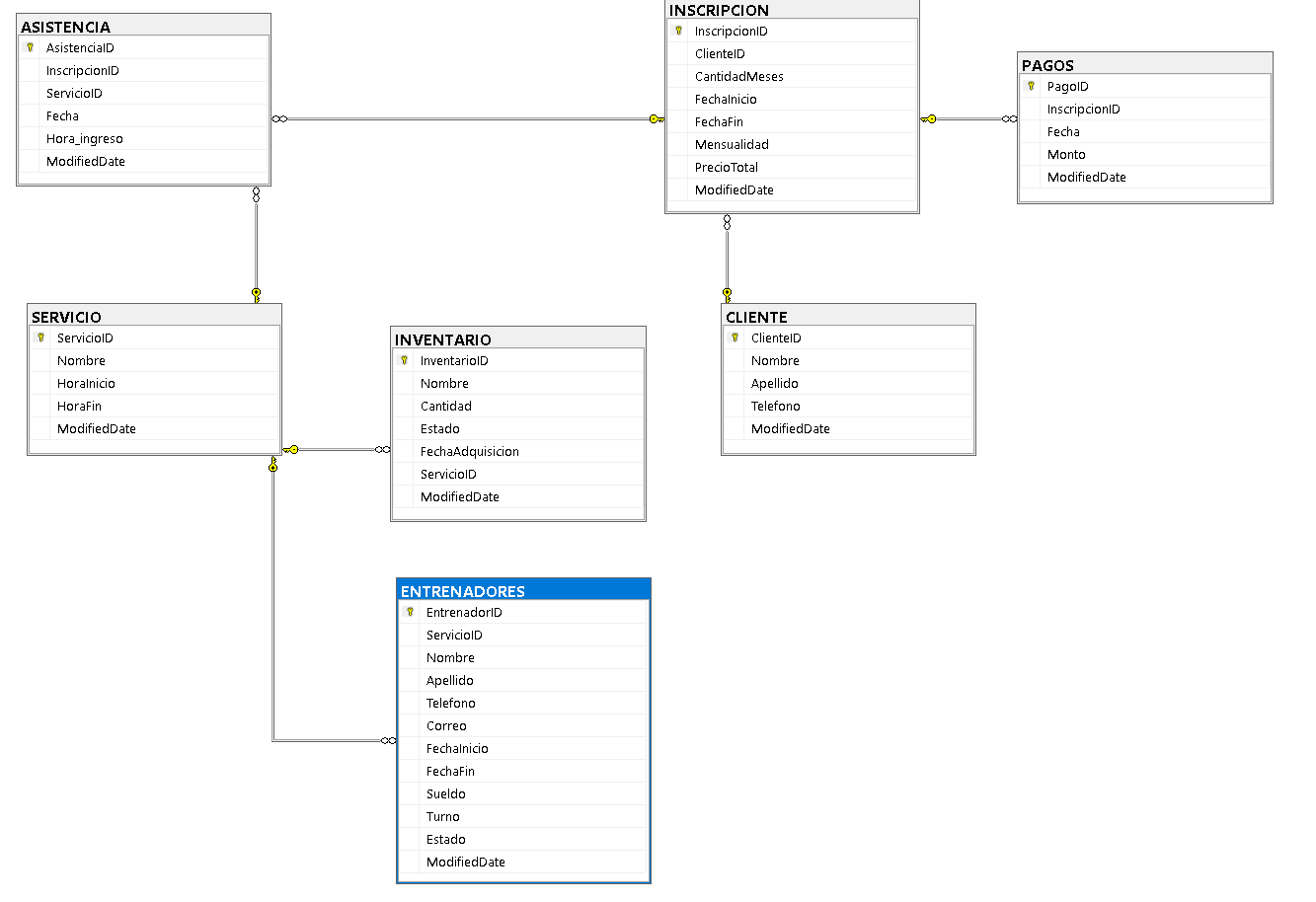
**Un servicio puede ser impartido a múltiples entrenadores**

1. **Servicio a Asistencia**



**Un servicio puede tener múltiples asistencias registradas**

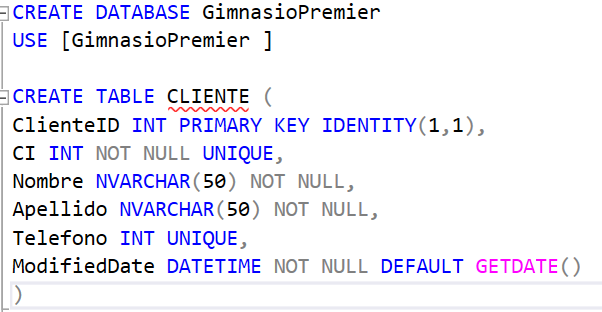
1. **Servicio a inventario**



**Un servicio puede tener múltiples maquinas**

1. Scripts

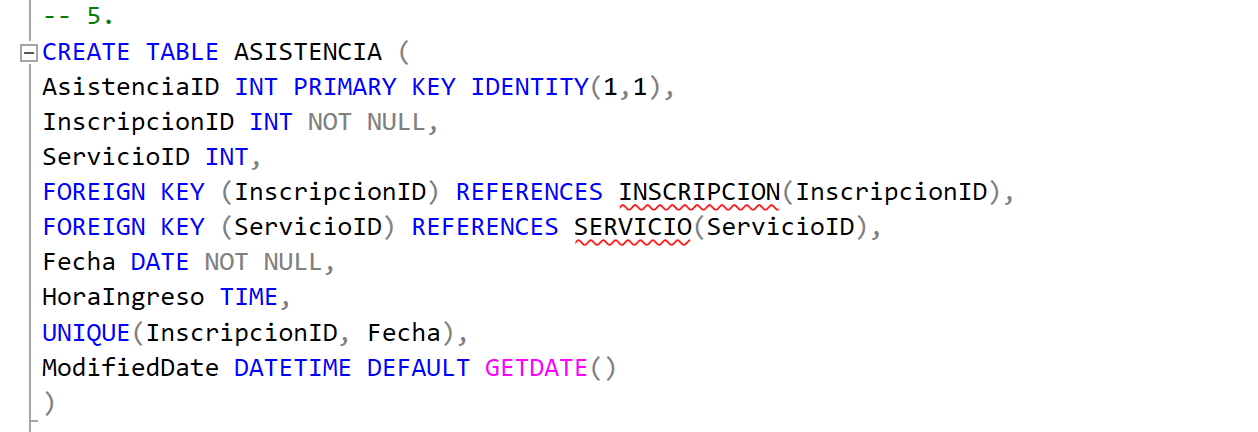
DDL:

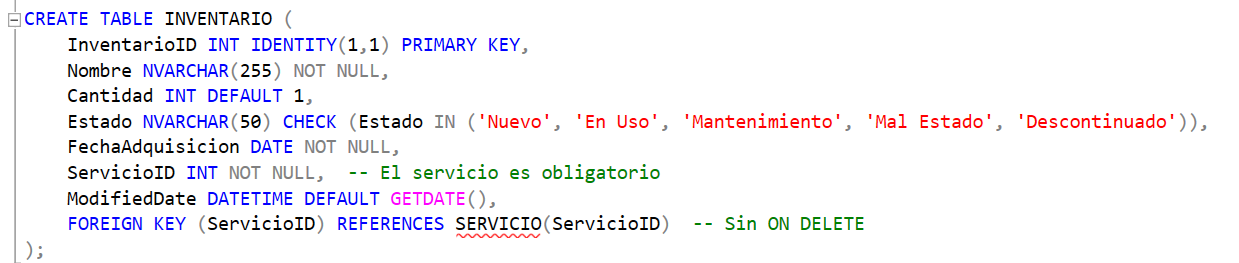


Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza mediaInterfaz de usuario gráfica, Texto, Aplicación

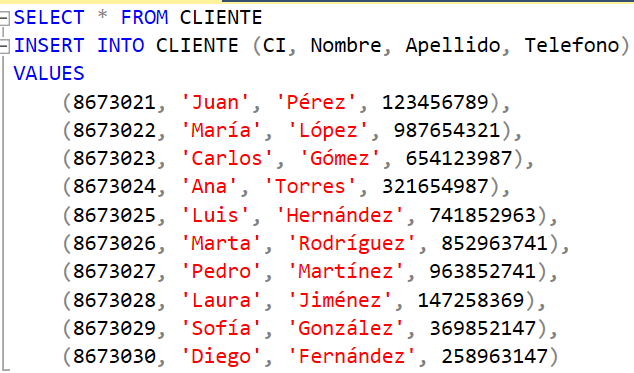
Descripción generada automáticamenteInterfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica

Descripción generada automáticamente

DML:

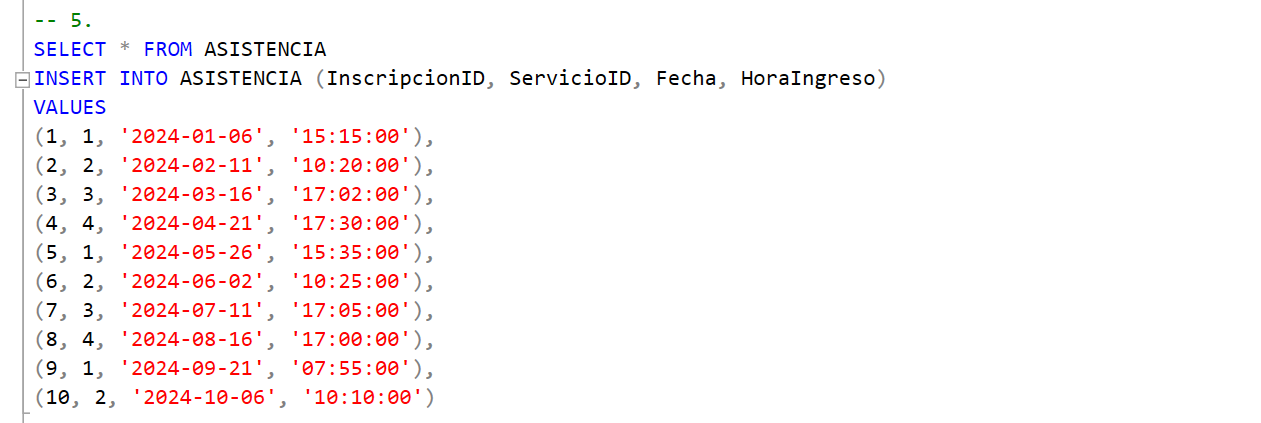


Interfaz de usuario gráfica, Aplicación

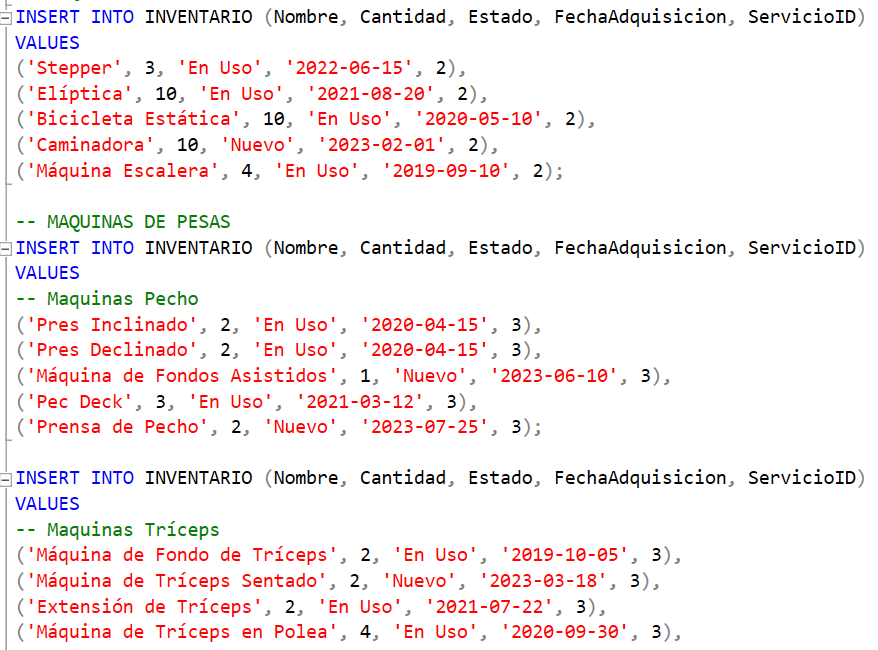
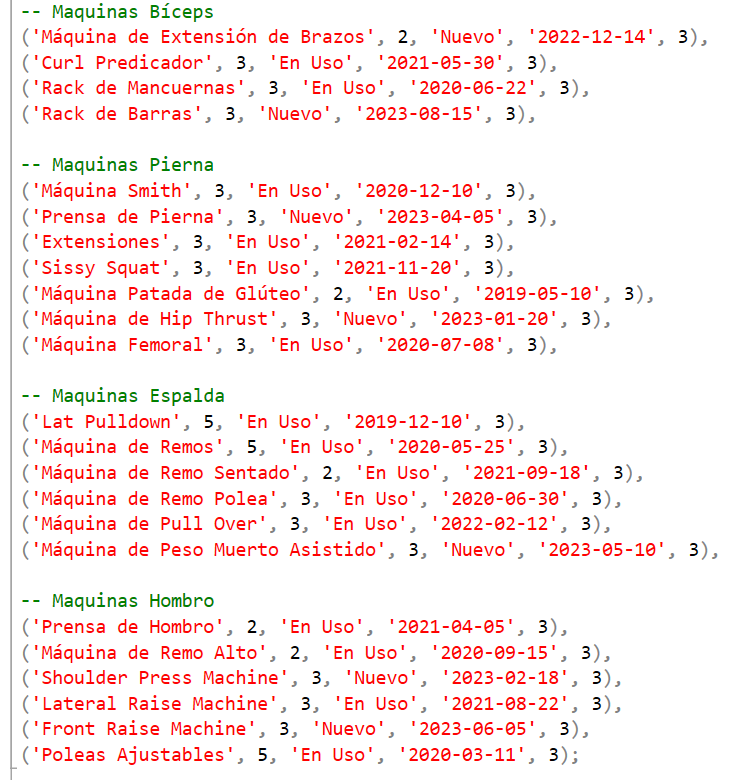
Descripción generada automáticamente

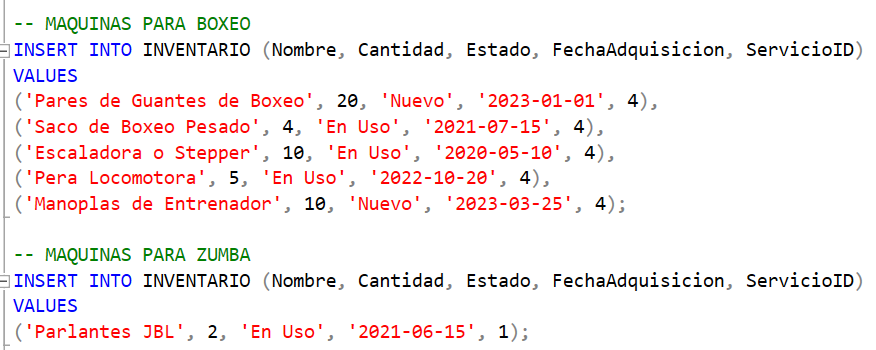
Texto

Descripción generada automáticamenteTexto

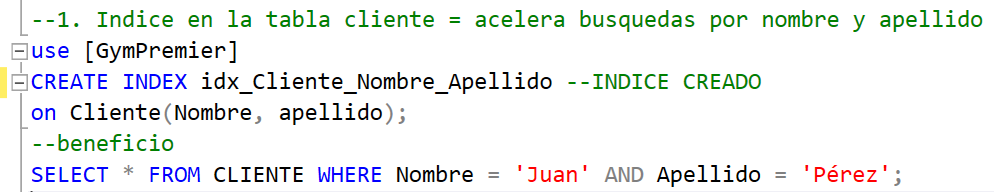
Descripción generada automáticamente con confianza mediaTexto

Descripción generada automáticamente





**Creacion de Indices**

****

**Create index** – indica la creación de un índice

**Idx\_Cliente\_Nombre\_Apellido** – indica el nombre del índice

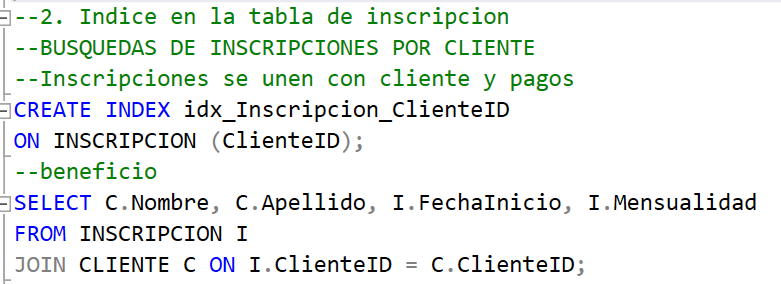
**On Cliente** – indica la tabla de donde se aplicara el índice

**Nombre, apellido** - especifica las columnas que formaran parte del índice

**¿Qué hace?**

Lo que hace es buscar a un cliente por su nombre y apellido

El índice no sirve si solo se busca por apellido



**Create index** - indica la creación de un índice

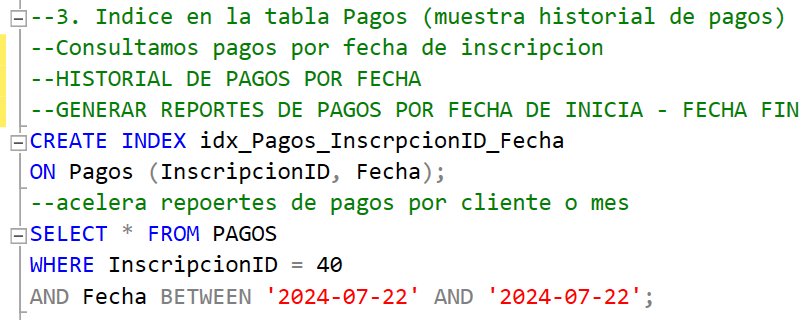
**Idx\_Isncripcion\_ClienteID** – indica el nombre del índice

**On inscripción** - especifica la tabla de donde se hará el índice

**Cliente id** - especifica la columna de donde se aplicar la consulta

**Que hace?**

Lo que hace es a ver búsquedas de inscripciones por el cliente



**Créate index** – indica la creación de un índice

**Idx\_pagos\_inscricpcionid\_fecha** – indica el nombre del índice

**On pagos** - indica en la tabla donde se hará el índice

**InscripcionID, fecha** – indica las columnas de donde se aplicará la consulta

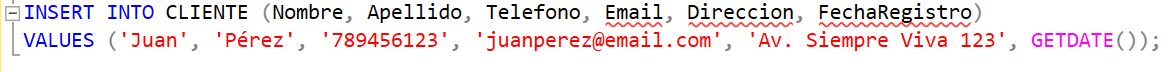
**Que hace**?

Realiza un reporte por la fecha de inscripción

**CREACION DE STORED PROCEDURE (SP)**

**Inscripción de un nuevo cliente sin sp**

**Procedimiento**

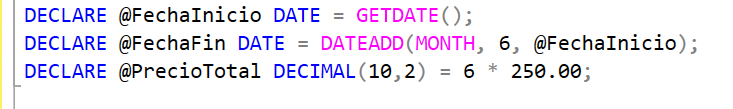
1. **Insertamos un nuevo cliente en la tabla cliente**
2. **Obtenemos el clienteid del nuevo cliente recién insertado**

Sino obtenemos cliente id , no sabra a que cliente inscribir

DECLARE @ClienteID INT = SCOPE\_IDENTITY();

De esta froma obtenemos al ultimo cliente recine genreado

Scope\_identity – se usa para obtener al ultimo cliente reccien añadido

1. **Calculamos las fechas y el precio total**

**DECLARE @FechaInicio DATE = GETDATE();** - declaramos una variable fechainicio de tipo date lo inicializamos con la fecha actual

**Ejemplo:**

**Si hoy es 1 de abril de 2025 entonces**

**@FechaInicio = 2025-05-01**

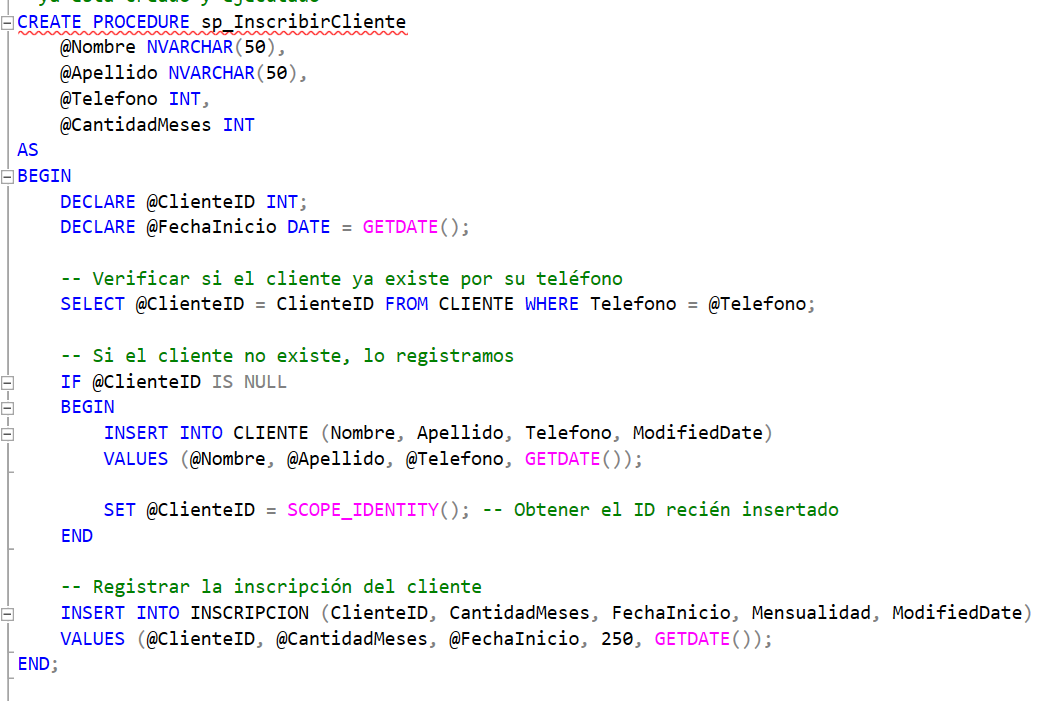
**DECLARE @FechaFin DATE = DATEADD(MONTH, 6, @FechaInicio); -** Declara una variable FechaFin de tipo date

Usa DATEADD(MONTH, 6, @FechaInicio) para sumar 6 meses a la fecha de inicio.

**Ejemplo:**

**Si @FechaInicio = '2025-04-01', entonces:  
@FechaFin = '2025-10-01' (6 meses después).**

**Que hace ?**

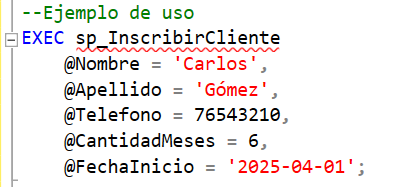
****

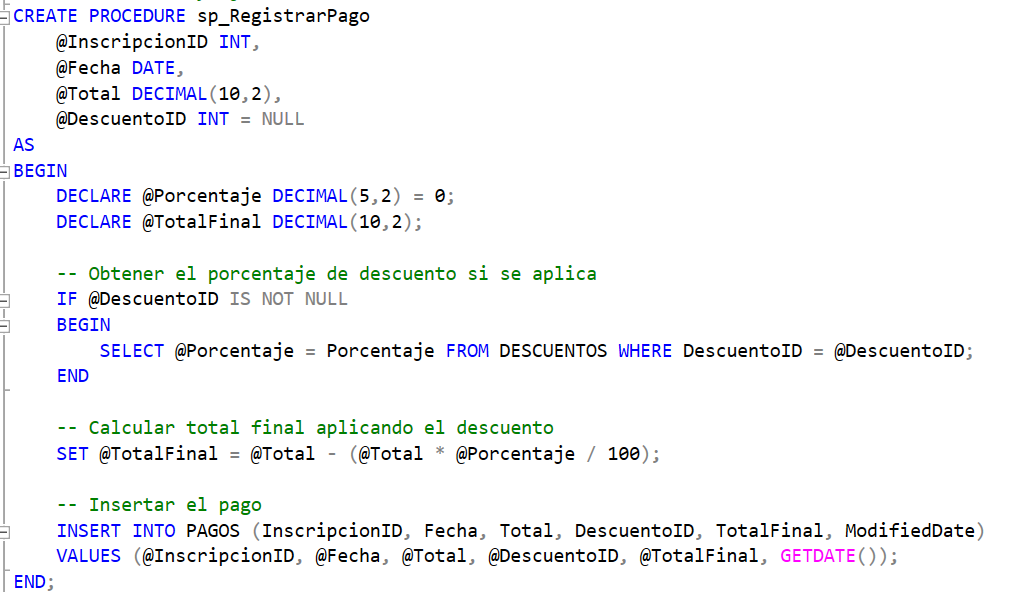
Lo que hace este sp es registrar a un nuevo cliente en la tabla cliente

Inicia una inscripción en la tabla de inscripción

Calcula automáticamente la fecha fin y el precio total

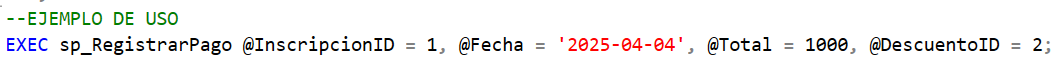
Ejecución del sp



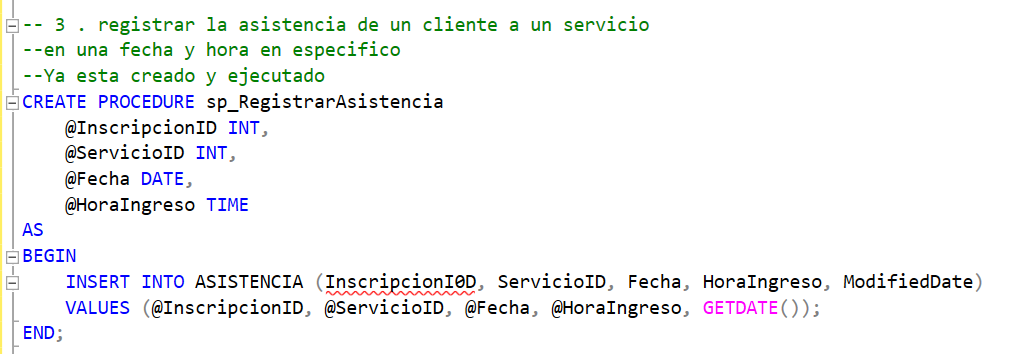
1. Registrar pago de un cliente

En resumen lo que hace este so es registra el pago de un cliente y aplicar el desceunto si es que exite

Primero comparamos si hay un descuento si no hay pasa por nulo y solo se aplica el precio original de 250 si existe algún descuento se pasa por la formula y a atreves de la variable total final se muestra el precio aplicado el descuento



1. Registrar la asistencia de un cliente

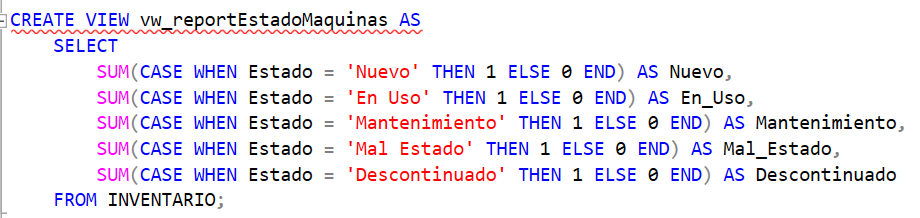
  
En resumen este sp tiene la función de registra la asistencia de de un cliente a un servicio a una fecha y hora determinada

Donde fecha agarra los valores año , mes, dia

Hora de ingreso agarra los la hora , minutos, segundos

E inserta la asistencia con sus respectivos valores a las columnas

**Views**

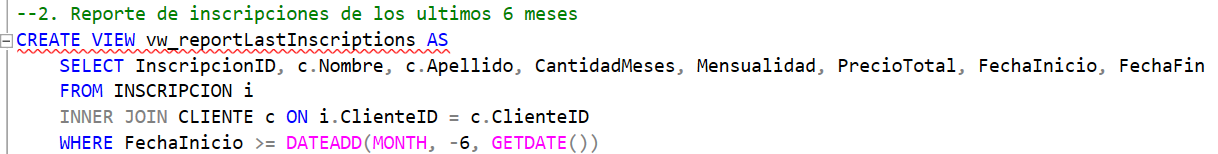
****

**Este view genera un reporte del estado de las maquinas basado en la tabla de inventario**

**SUM(CASE WHEN Estado = '...' THEN 1 ELSE 0 END)**

**Para cada fila en la tabla inventario suma 1, si el estado coincide sale 0 si no**

**Agrupa toda la infromacion en una sola fila , la consulta no tiene group by el resultado saldrá en una sola fila será una única fila con las sumas del estado**

****

**El objetivo de esta vista es generar un reporte de los últimos 6 meses une la rabla inscripción con la tabla clientes usando cliente id**

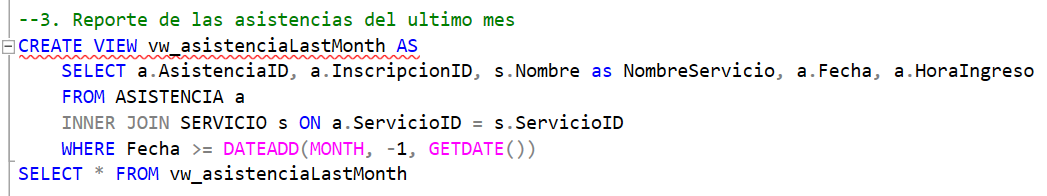
**Asi obtiene los datos del cliente nombre y apellido junto a la información de la inscripción**

**WHERE FechaInicio >= DATEADD(MONTH, -6, GETDATE())**

**DATEADD(MONTH, -6, GETDATE())**

**Calcula la fecha de hace 6 meses a partir de hoy**

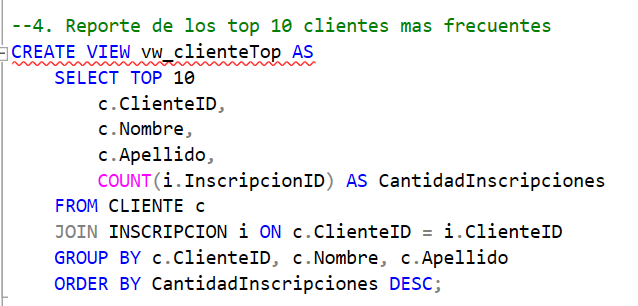
**Solo muestra inscripciones con fecha inicio**

****

**El objetivo de esta vista es generar un reporte de las asistencias registradas en el ultimo mes**

**Se realiza un inner join a la tabla asistencia y la tabla servicio usando la columna de servicioID**

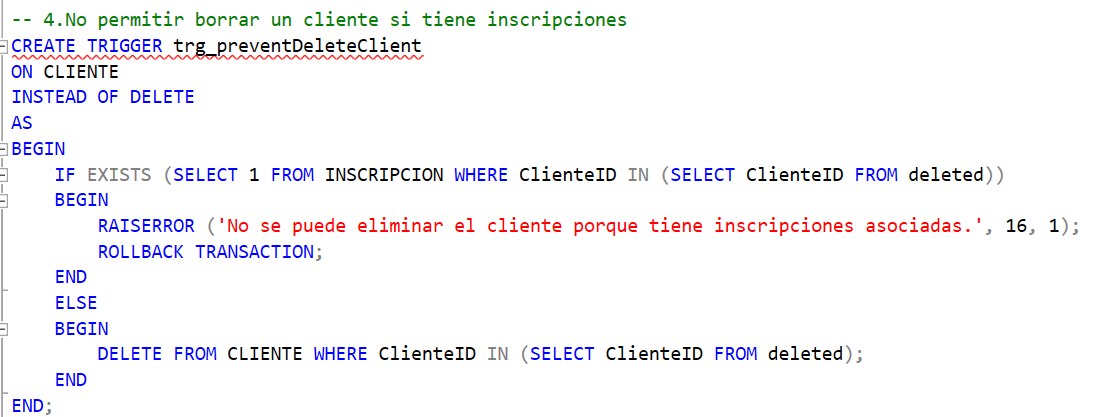
**la sentencia de where filtra los registros par obtener únicamente donde le fecha de asistencia es mayor o igual a la fecha actual menos uno**

****

**Este viee tiene como objetivo mostrar los 10 clientes con mayor cantidad de inscrpciones en el el gimnasio.**

**Usamos la función COUNT(i,inscripcionID) para contar cuantas inscripciones tiene cada cliente ese conteo se muestra en la columna de clienteID esto realciona cada cliente con sus inscripciones**

**Triggers DELETED**

****

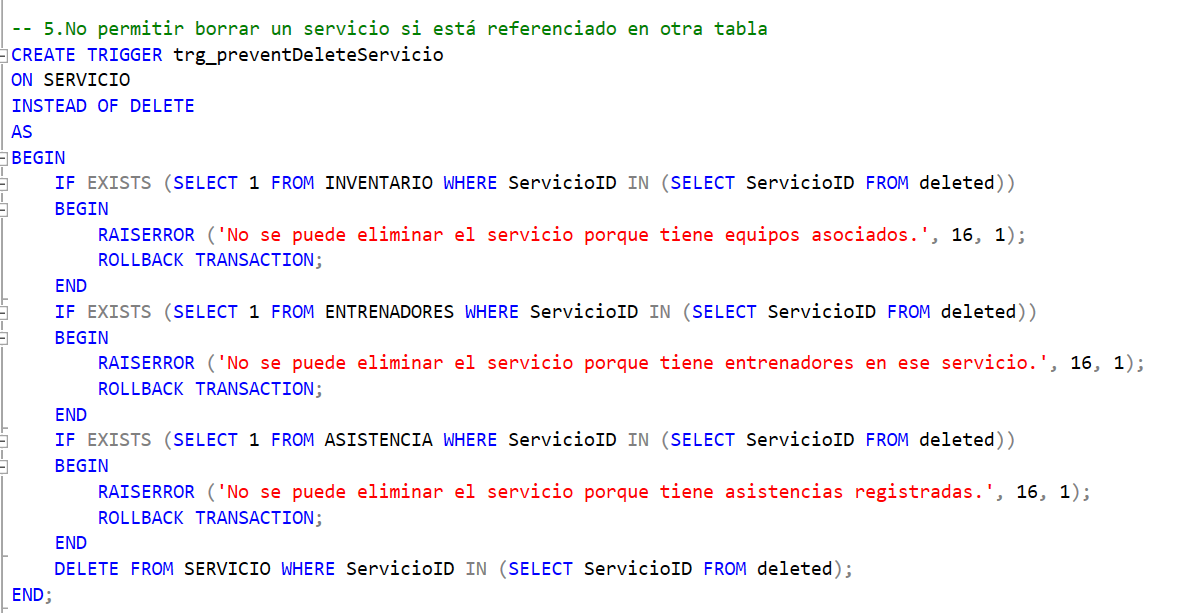
**El objetivo de este trigger es borrar a un cliente ,si tiene almenos una inscripción en la tabla inscripción el cliente no se borra .**

**De manera mas detallada este trigger asegura la integridad referecial de la base de datos**

**Cuando se intenta eliminar un cliente, se revisa si en la tabla inscripción para ver si el cliente tiene inscripciones**

**Si es asi , se muestra un error y se cancela la eliminación**

**Si no , se permite la eliminación de ese cliente**

****

**Este trigger se activa cuando queremos eliminar un servicio**

**Evita que un servicio sea eliminado si esta concectado con otras tablas**

**INSTEAD OF DELETE**

**RAISERROR para lanzar mensajes personalizados**

**ROLLBACK TRANSACTION para evitar cambios no deseados**

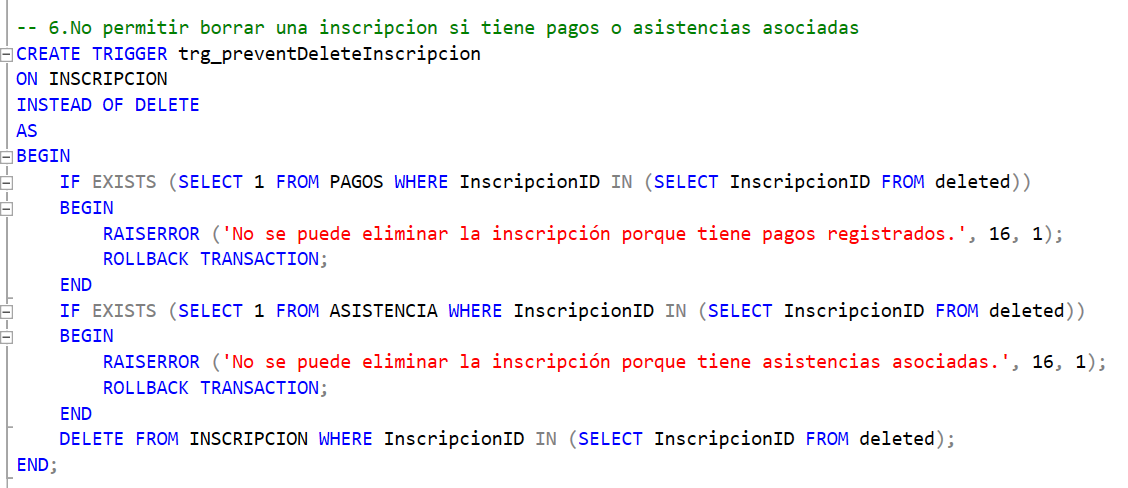
**Como probarlo?**

**DELETE FROM SERVICIO WHERE ServicioID = 2;**

**Msg 50000, Level 16, State 1, Procedure trg\_preventDeleteServicio**

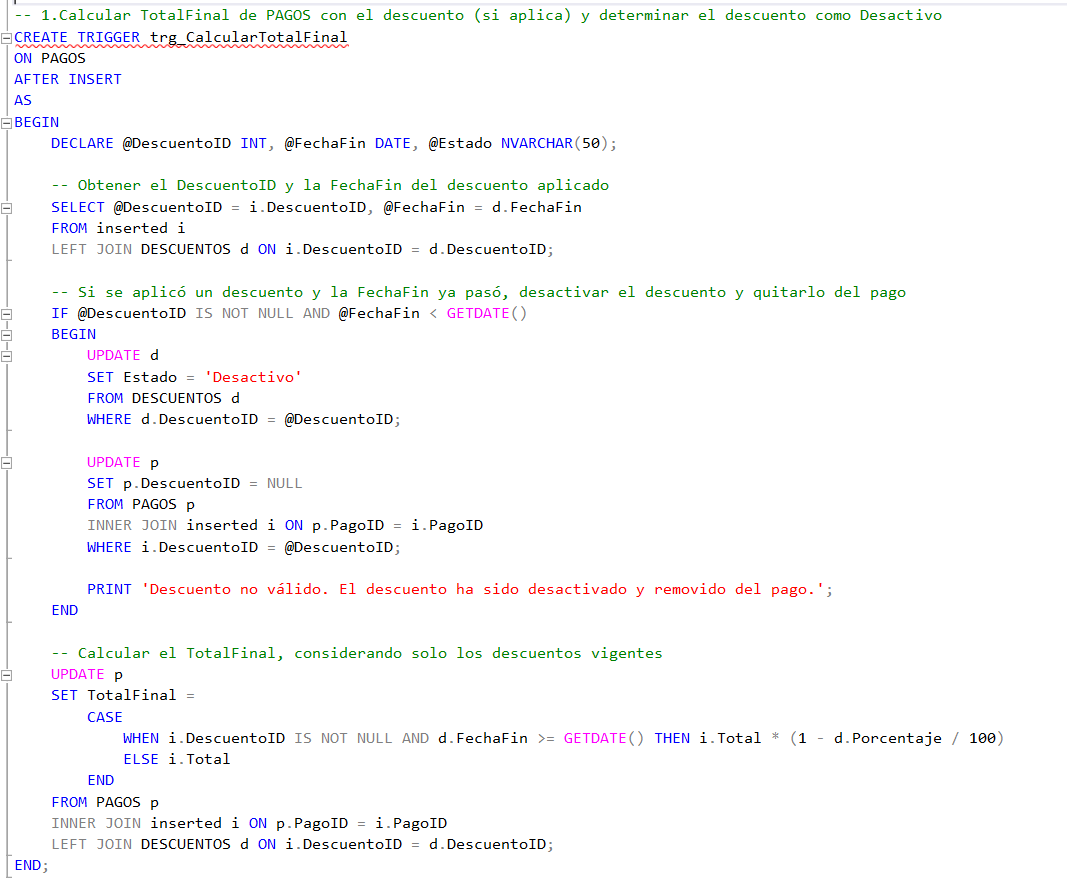
**No se puede eliminar el servicio porque tiene equipos asociados.**

**Si ese servicio tiene máquinas en el inventario, entrenadores o asistencias, aparecerá un error como:**

****

**Este trigger se llama trg\_preventDeleteInscripcion y se ejecuta en lugar de la operación de DELETE sobre la tabla INSCRIPCION. Su objetivo es proteger la integridad de los datos al evitar que se elimine una inscripción que ya tiene pagos o asistencias asociados.**

**Trriggers INSERT**

****

**Este trigger se activa después de insertar un nuevo pago y se encarga de:**

1. **Verificar si el descuento aplicado sigue vigente.**

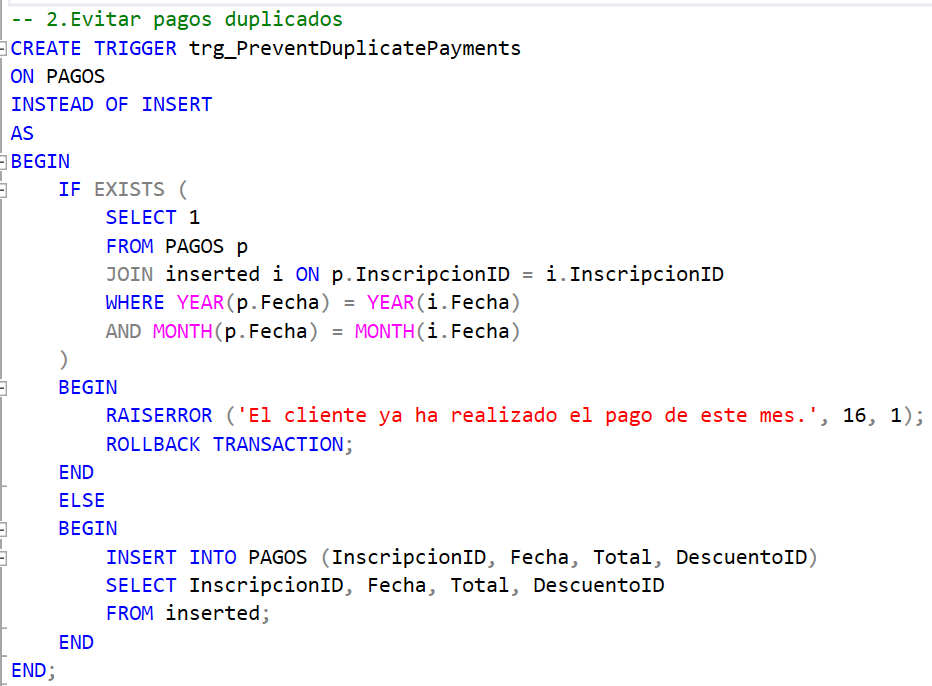
**Si el descuento tiene una FechaFin que ya pasó, lo desactiva (Estado = 'Desactivo') y elimina el descuento del pago (DescuentoID = NULL).**

**Esto evita que se apliquen descuentos caducados.**

1. **Calcular el TotalFinal del pago.**

**Si hay un descuento vigente, aplica el porcentaje.**

**Si no hay descuento o ya está vencido, el TotalFinal es igual al Total.**

****

**Evita que se realice más de un pago por mes para la misma inscripción.**

**Antes de insertar un nuevo pago, verifica si ya existe un pago en el mismo mes y año para el mismo InscripcionID.**

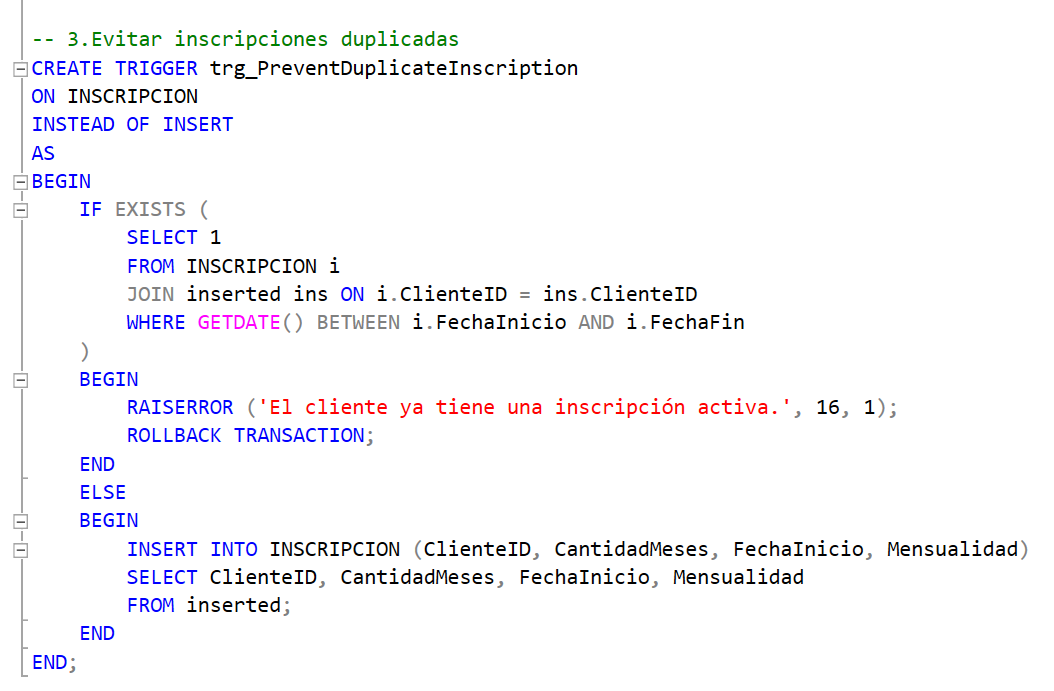
**Si ya existe uno:**

**Lanza un error: 'El cliente ya ha realizado el pago de este mes.'**

**Hace un ROLLBACK, así que el nuevo pago no se guarda.**

**Si no existe:**

**Inserta normalmente el pago.**

****

**Evita que un cliente tenga más de una inscripción activa al mismo tiempo.**

**Verifica si el cliente del inserted ya tiene una inscripción cuya FechaInicio y FechaFin abarquen la fecha actual (GETDATE()).**

**Si tiene una inscripción activa:**

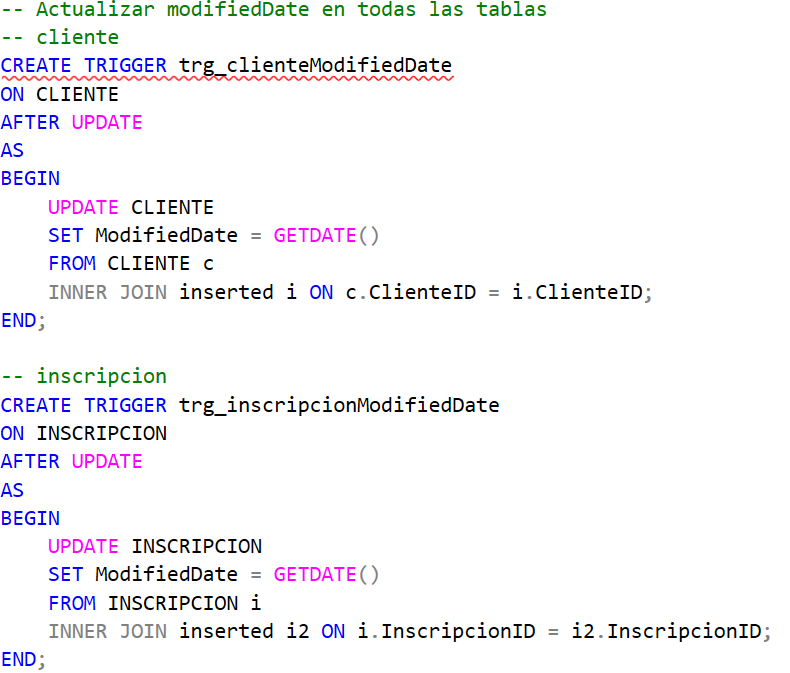
**Lanza un error: 'El cliente ya tiene una inscripción activa.'**

**Cancela la inserción (ROLLBACK).**

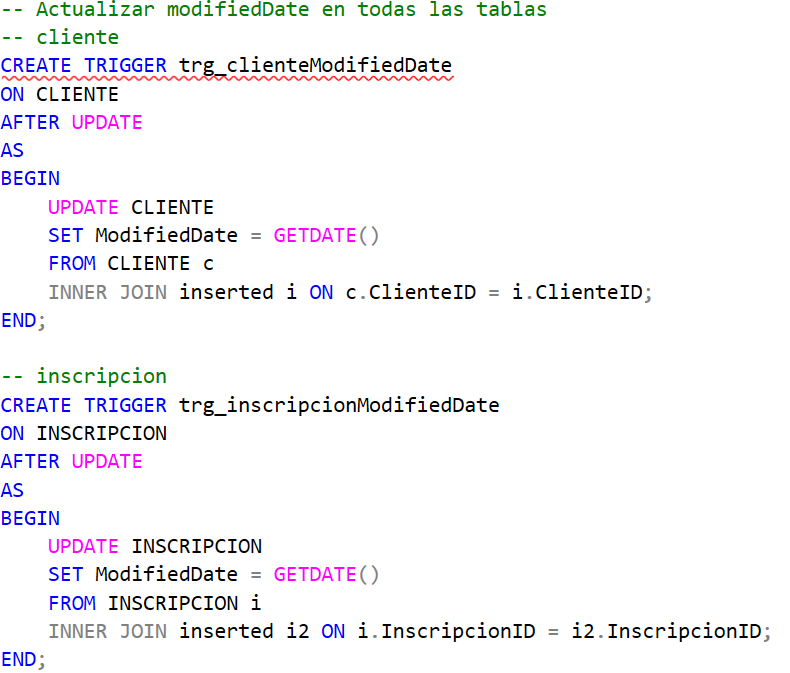
**Si no tiene una inscripción activa:**

**Inserta normalmente la nueva inscripción.**

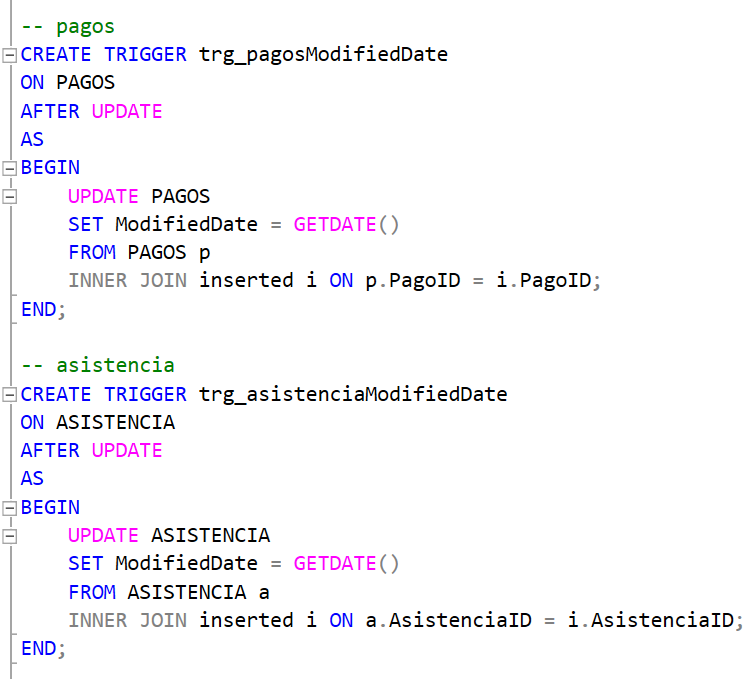
**Triggers UPDATE**

****

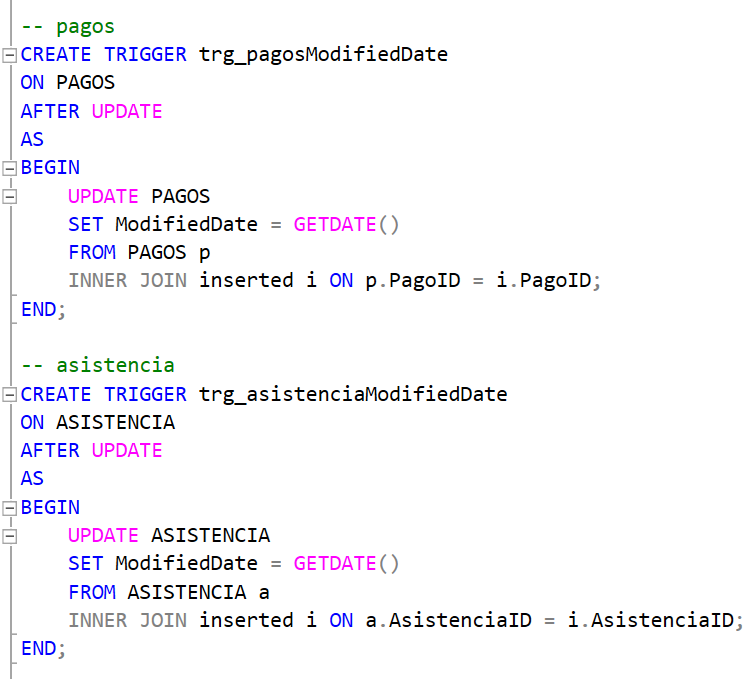
**Este trigger se activa después de actualizar un cliente.  
Actualiza automáticamente el campo ModifiedDate con la fecha y hora actual.  
Permite llevar control de cuándo fue modificado un registro de cliente.  
Utiliza la tabla inserted para identificar los clientes afectados.  
Es útil para auditoría y trazabilidad de cambios.**



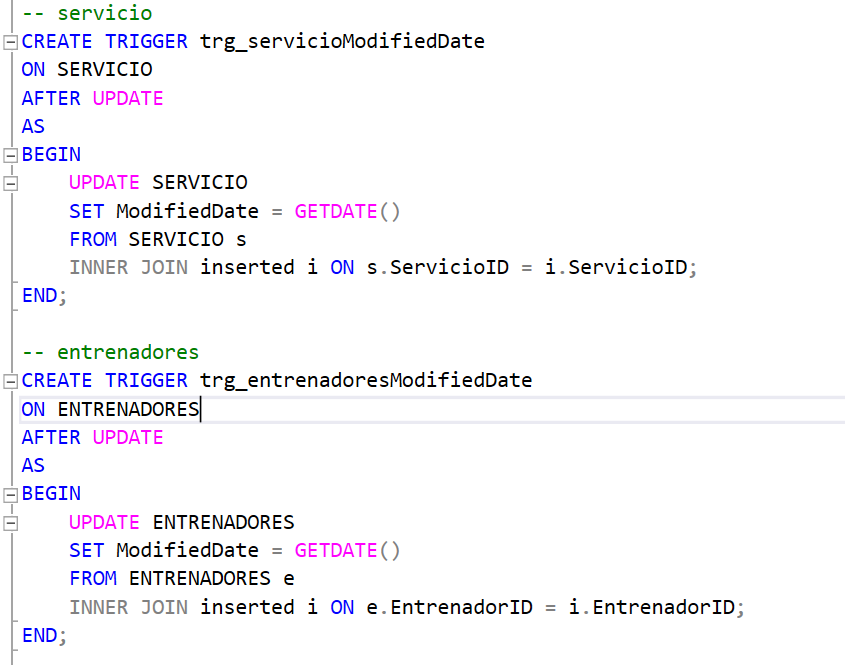
**Se ejecuta luego de una actualización en la tabla INSCRIPCION.  
Actualiza el campo ModifiedDate con la fecha actual (GETDATE()).  
Permite saber cuándo una inscripción fue modificada.  
Compara la tabla inserted para identificar los registros modificados.  
Ayuda en el control de cambios en inscripciones activas.**

****

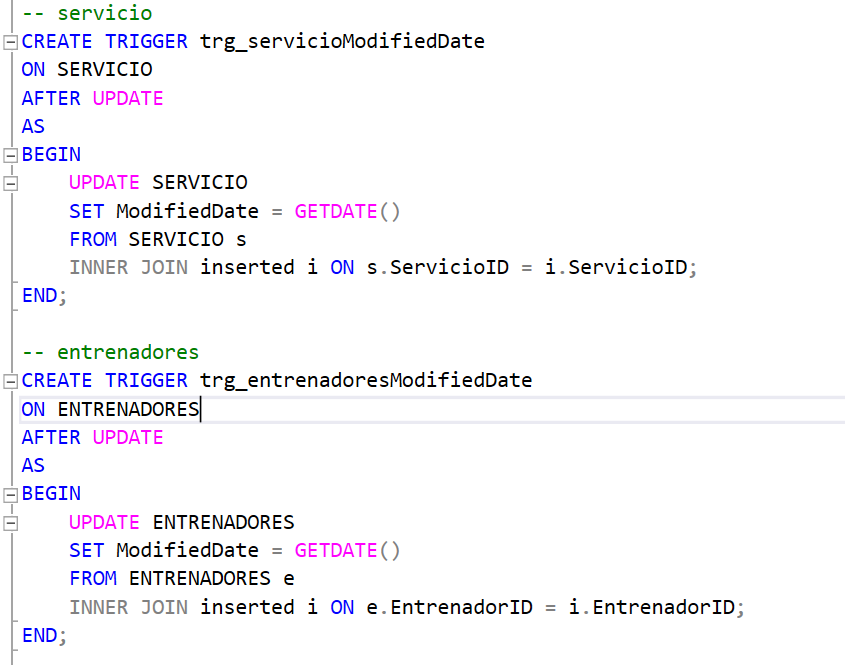
**Este trigger se dispara después de modificar un pago.  
Actualiza la columna ModifiedDate en la tabla PAGOS.  
Usa la tabla inserted para aplicar el cambio solo al pago afectado.  
Es útil para saber cuándo fue alterado un registro de pago.  
Contribuye a la integridad y trazabilidad de los datos financieros.**

****

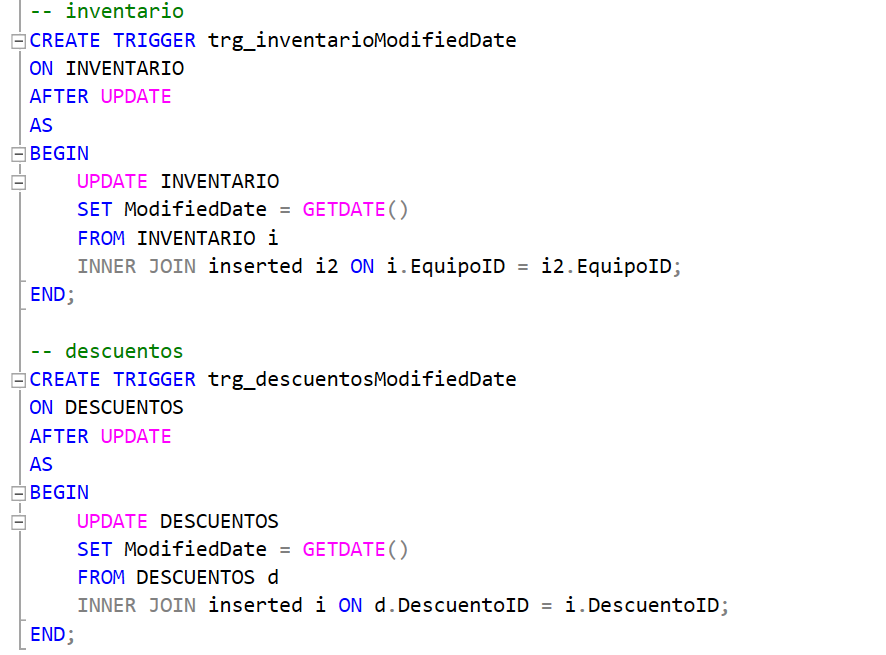
**Se ejecuta luego de un UPDATE en la tabla ASISTENCIA.  
Cambia la columna ModifiedDate con la fecha y hora actual.  
Sirve para saber la última vez que se modificó una asistencia.  
Aplica únicamente a los registros modificados.  
Es útil para control de registro de entrada y salida en el gimnasio.**

****

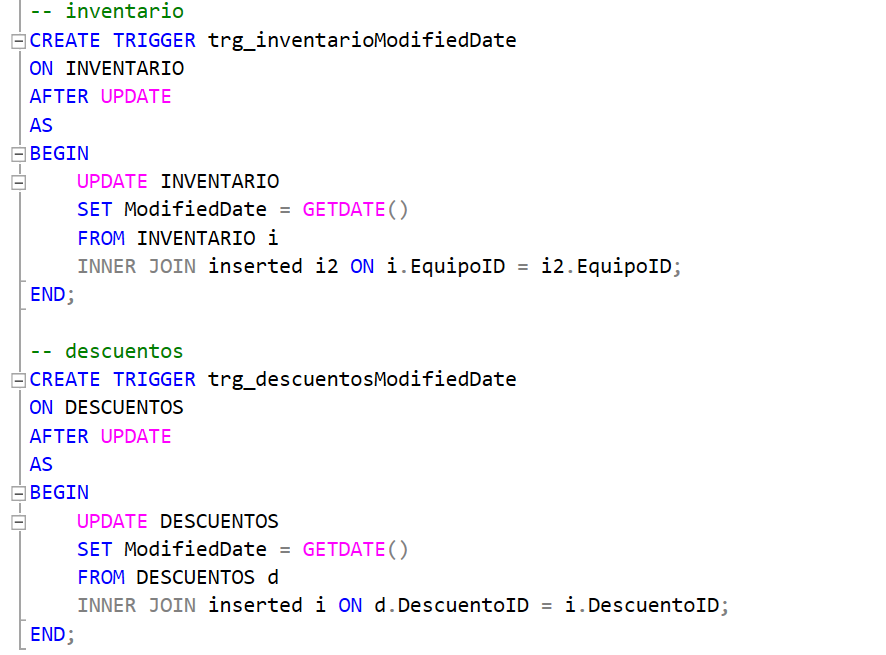
**Se activa tras una modificación en la tabla SERVICIO.  
Actualiza automáticamente el campo ModifiedDate.  
Utiliza GETDATE() para establecer la fecha actual.  
Solo modifica los servicios que hayan sido actualizados.  
Ayuda a mantener un historial de cambios en los servicios ofrecidos.**

****

**Se ejecuta luego de modificar un registro en ENTRENADORES.  
Actualiza la columna ModifiedDate con la fecha y hora actual.  
Identifica al entrenador modificado mediante la tabla inserted.  
Sirve para registrar cuándo se actualizó la información de un entrenador.  
Es útil para seguimiento de personal.**

****

**Este trigger se dispara después de actualizar un equipo del inventario.  
Modifica la columna ModifiedDate con la fecha actual.  
Afecta solamente los registros modificados (por EquipoID).  
Sirve para auditar los cambios en el estado o información del equipo.  
Muy útil para control de mantenimiento de máquinas.**

****

**Se ejecuta luego de actualizar un descuento.  
Modifica el campo ModifiedDate en la tabla DESCUENTOS.  
Solo actualiza los descuentos afectados por el cambio.  
Permite saber cuándo se modificó un descuento o su vigencia.  
Apoya el control y gestión de promociones activas.**