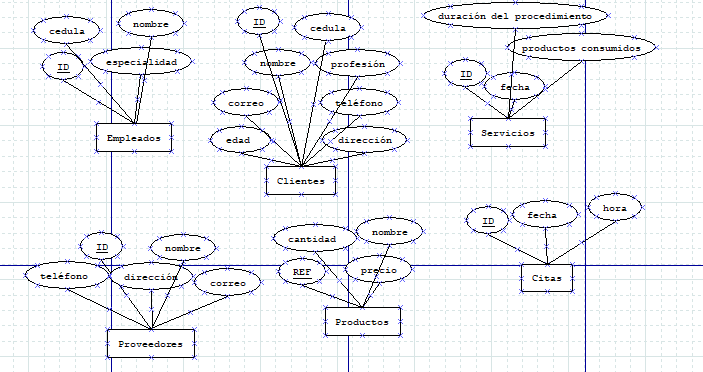
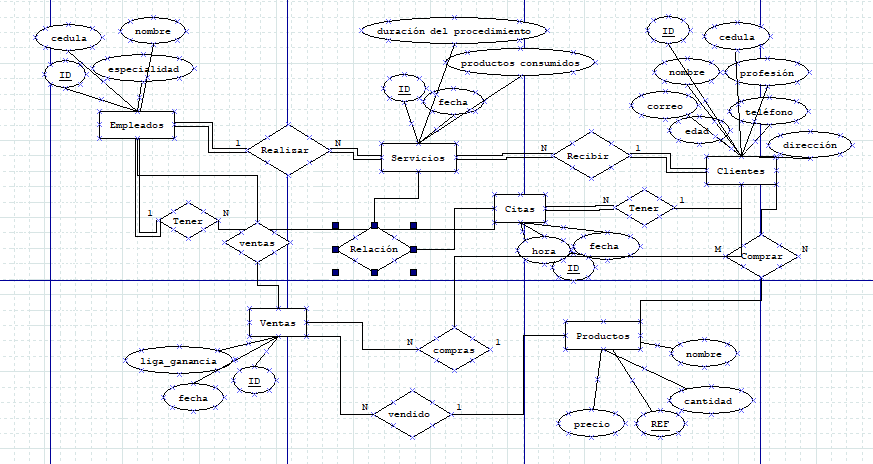
**Barbería (Ejercicio A)**

1. Generando tablas y atributos diagrama E-R



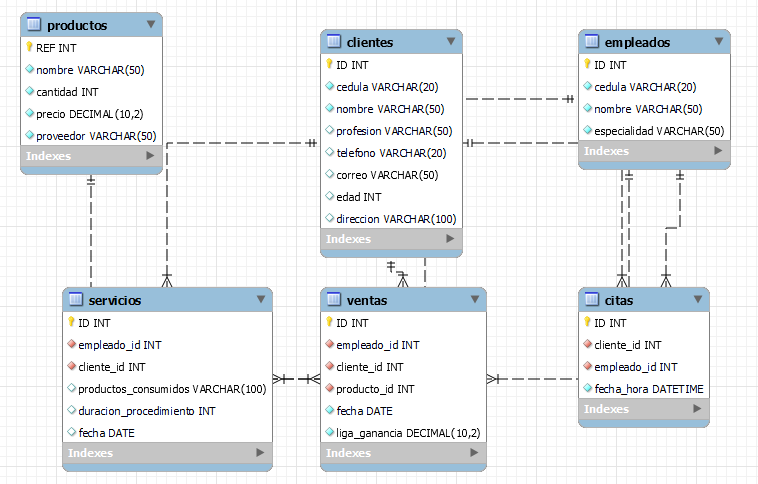
Generando tablas y atributos diagrama E-R y relaciones



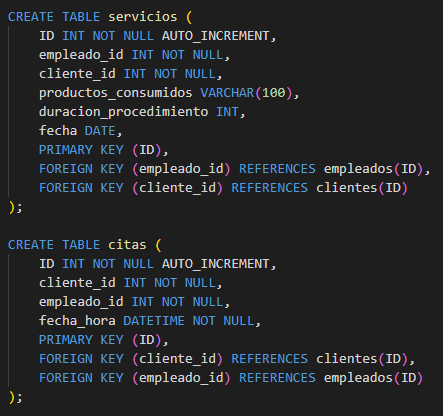
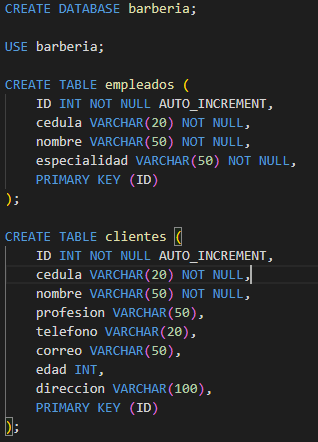
**Las relaciones entre las entidades son las siguientes:**

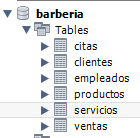
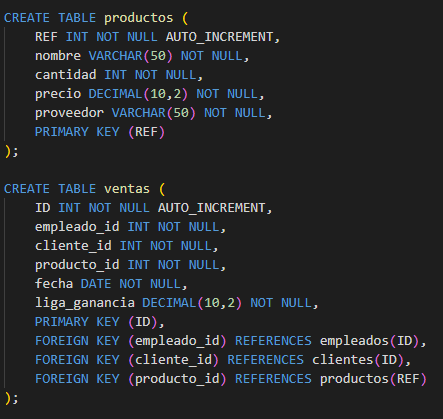
* La entidad "Empleados" tiene una relación de uno a muchos con la entidad "Servicios", ya que un empleado puede prestar muchos servicios, pero un servicio solo puede ser prestado por un empleado.
* La entidad "Clientes" tiene una relación de uno a muchos con la entidad "Citas", ya que un cliente puede tener muchas citas, pero una cita solo puede ser programada para un cliente.
* La entidad "Empleados" tiene una relación de uno a muchos con la entidad "Citas", ya que un empleado puede tener muchas citas, pero una cita solo puede ser asignada a un empleado.
* La entidad "Productos" tiene una relación de uno a muchos con la entidad "Ventas", ya que un producto puede ser vendido muchas veces, pero una venta solo puede incluir un producto.
* La entidad "Empleados" tiene una relación de uno a muchos con la entidad "Ventas", ya que un empleado puede hacer muchas ventas, pero una venta solo puede ser realizada por un empleado.
* La entidad "Clientes" tiene una relación de uno a muchos con la entidad "Ventas", ya que un cliente puede hacer muchas compras, pero una compra solo puede ser realizada por un cliente.

**Modelo relación:**



**Consulta para crear la base datos :**

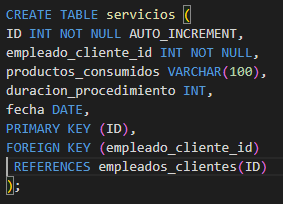
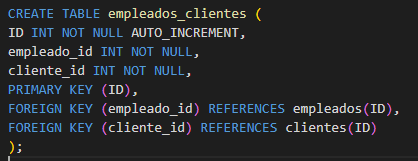


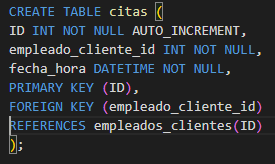


**Normalizando base de datos**

**1FN:**

* Para aplicar la primera forma normal a esta base de datos, se debe revisar cada tabla y asegurarse de que cada columna tenga un valor único y que no haya valores repetidos. En caso de que una tabla contenga valores repetidos, estos se deben eliminar y crear una nueva tabla para almacenarlos.
* En este caso, se puede ver que las tablas empleadas, clientes, productos y ventas cumplen con la primera forma normal, ya que cada columna tiene un valor único. Por lo tanto, no es necesario realizar ninguna modificación en estas tablas.
* Sin embargo, la tabla servicios y citas contienen columnas repetidas. La tabla servicios tiene las columnas empleado\_id y cliente\_id, que se refieren a las mismas entidades que las columnas ID de las tablas empleados y clientes, respectivamente. De manera similar, la tabla citas tiene las columnas cliente\_id y empleado\_id, que se refieren a las mismas entidades que las columnas ID de las tablas clientes y empleados, respectivamente.
* Para cumplir con la primera forma normal, se debe crear una nueva tabla que contenga las columnas ID de las tablas empleados y clientes, y luego referenciar estas tablas desde las tablas servicios y citas.
* **A continuación se muestra el código para crear las nuevas tablas:**

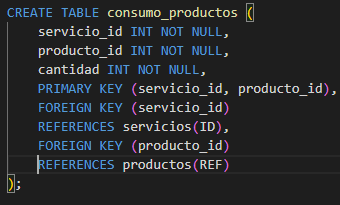
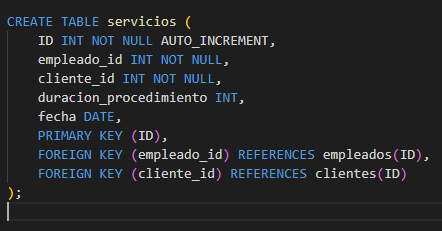




* En la tabla empleados\_clientes, se han eliminado las columnas cedula, nombre y especialidad, ya que estas columnas se encuentran en las tablas empleados y clientes. En su lugar, se ha agregado una nueva columna ID para identificar la relación entre empleados y clientes.
* En las tablas servicios y citas, se ha agregado una nueva columna empleado\_cliente\_id para referenciar a la tabla empleados\_clientes. Además, se han eliminado las columnas empleado\_id y cliente\_id.
* De esta manera, se ha aplicado la primera forma normal a la base de datos barberia.

**2FN:**

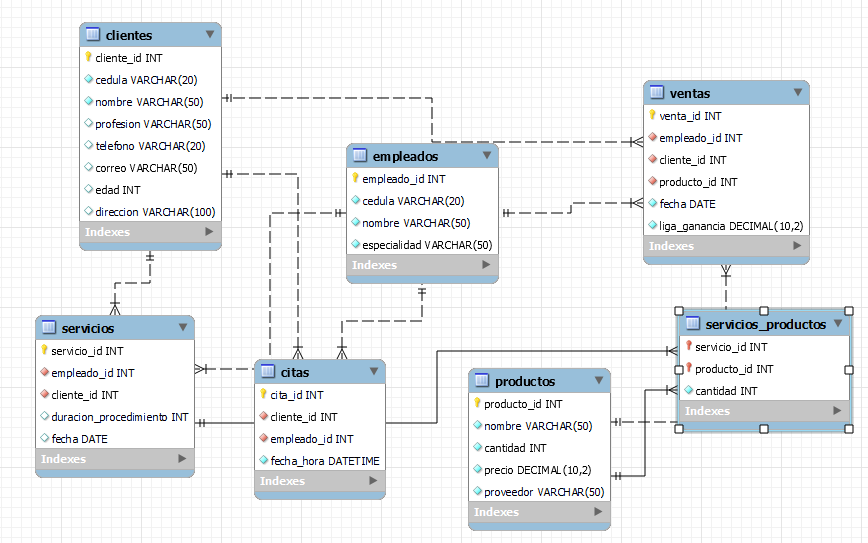
* Para aplicar la Segunda Forma Normal (2FN) a la base de datos "barberia", primero debemos verificar si existe alguna tabla con dependencias parciales en sus columnas. En este caso, la única tabla que puede presentar este problema es la tabla "servicios", ya que la columna "productos\_consumidos" depende funcionalmente del identificador del servicio y no del identificador de producto. Esto puede causar redundancia de datos si un mismo producto se consume en múltiples servicios.
* Por lo tanto, podemos aplicar la 2FN descomponiendo la tabla "servicios" en dos tablas: "servicios" y "consumo\_productos". La tabla "servicios" contendrá solo la información relacionada con el servicio, mientras que la tabla "consumo\_productos" contendrá la información de los productos consumidos en cada servicio.
* **A continuación, mostramos el código SQL para crear las nuevas tablas:**



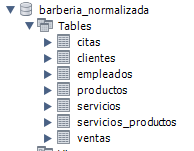
**3FN:**

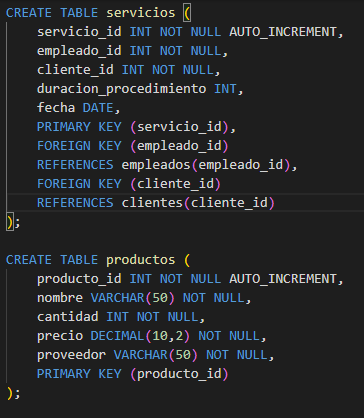
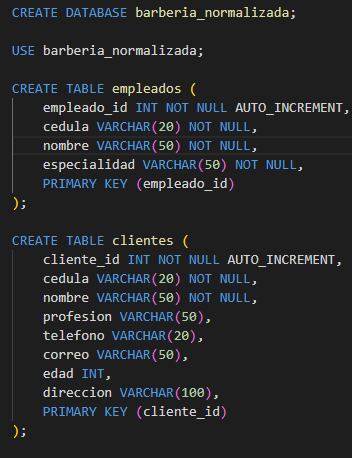
Después de aplicar la segunda forma normal, en este caso no es necesario aplicar la tercera forma normal ya que todas las tablas cumplen con las condiciones de esta forma normal. Las tablas tienen claves primarias únicas y no hay dependencias transitivas.

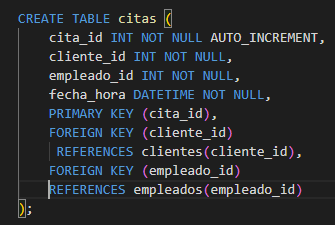
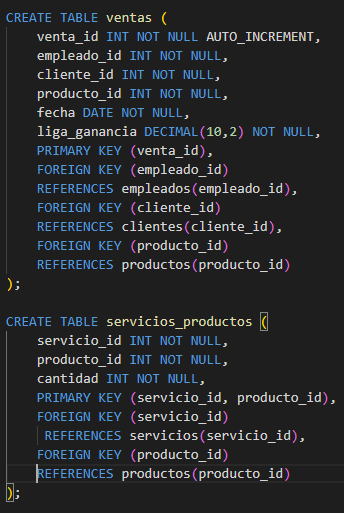
**Diagrama de Barberia normalizada:**



Consultas para crear la base de datos normalizada



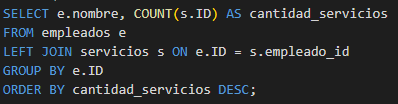




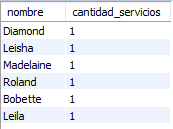
**Escribir consultas que me permitan ver la información de cada tabla o de varias tablas (10):**

1. Obtener el nombre de los empleados y la cantidad de servicios que han prestado, ordenados de mayor a menor cantidad de servicios:

* Consulta

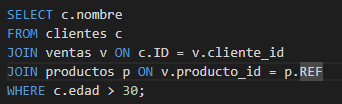


* Resultado



1. Obtener el nombre de los clientes que tienen más de 30 años y que han comprado algún producto vendido por la barbería:

* Consulta

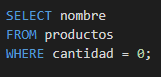


* Resultado



1. Obtener el nombre de los productos vendidos por la barbería que se han agotado:

* Consulta

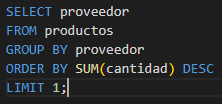


* Resultado



1. Obtener el nombre del proveedor que ha suministrado la mayor cantidad de productos a la barbería:

* Consulta

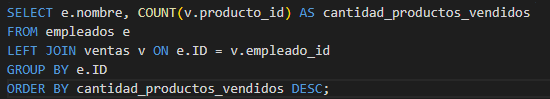


* Resultado

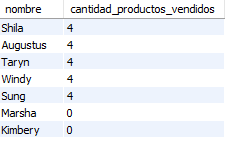


1. Obtener el nombre de los empleados y la cantidad de productos que han vendido, ordenados de mayor a menor cantidad de productos:

* Consulta

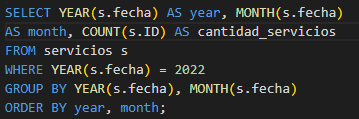


* Resultado



1. Obtener el número de servicios prestados por mes, para el año 2022:

* Consulta

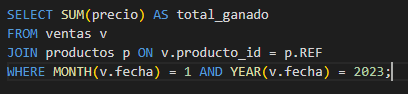


* Resultado



1. Obtener la cantidad total de dinero ganado por la barbería por la venta de productos, en el mes de enero de 2023:

* Consulta

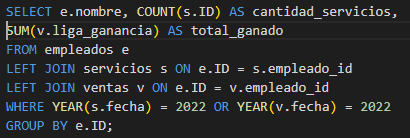


* Resultado



1. Obtener la cantidad de servicios prestados por cada empleado, junto con el total de dinero ganado por ellos a través de la venta de productos, en el año 2022:

* Consulta

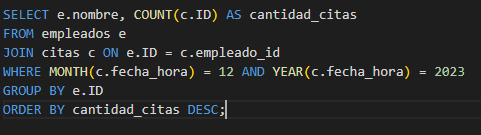


* Resultado



1. Obtener el número de citas por empleado, ordenados de mayor a menor número de citas, para el mes de diciembre de 2023:

* Consultado

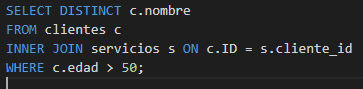


* Resultado



1. La consulta SQL para obtener el nombre de los clientes que tienen más de 50 años y que han recibido servicios en la barbería sería la siguiente:

* Consulta



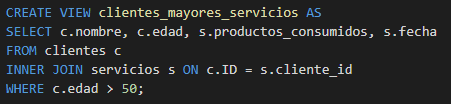
* Resultado



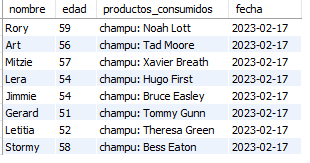
**Generar de 4 a 6 vistas donde se evidencie lo más importante de cada ejercicio (haga una selección muy responsable de la información realmente importante según el contexto):**

1. Vista de los clientes que tienen más de 50 años y que han recibido servicios:

* Consulta



* Resultado

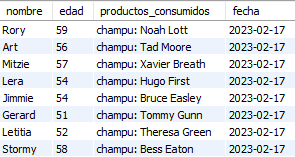


1. Vista de los empleados y sus especialidades:

* Consulta

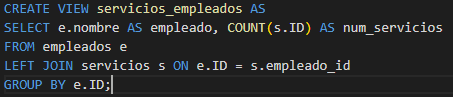


* Resultado



1. Vista de los servicios realizados por cada empleado:

* Consulta



* Resultado

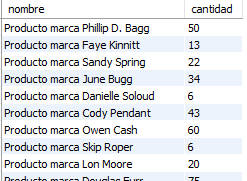


1. Vista de los productos con su cantidad disponible:

* Consulta

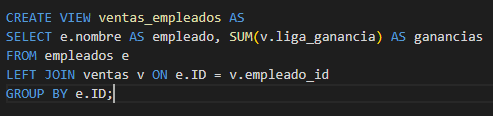


* Resultado

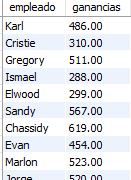


1. Vista de las ventas realizadas por cada empleado:

* Consulta

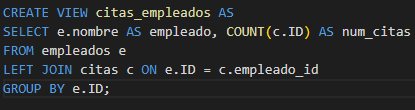


* Resultado

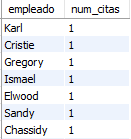


1. Vista de las citas agendadas para cada empleado:

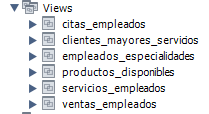
* Consulta



* Resultado



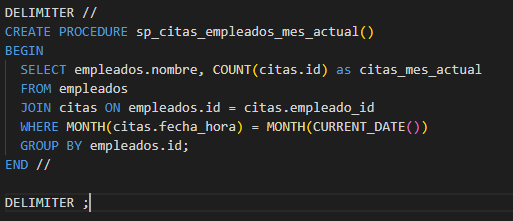
* Resultados totales



**Generar al menos 4 procedimientos almacenados:**

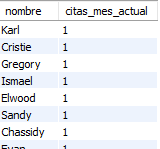
1. Obtener el listado de los empleados con la cantidad de citas que han tenido durante el mes actual:

* Consulta



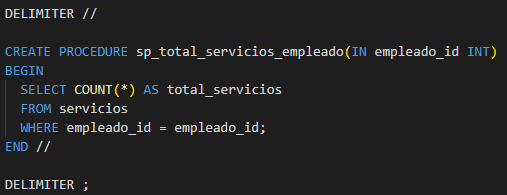


* Resultado



1. Obtener el número total de servicios realizados por un empleado en particular:

* Consulta



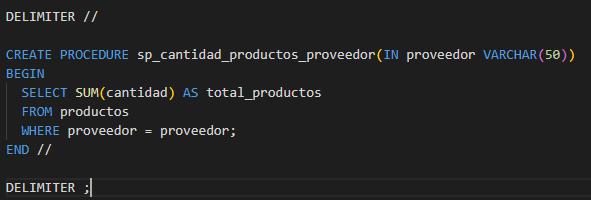


* Resultado



1. Obtener la cantidad de productos en existencia de un proveedor en particular:

* Consulta



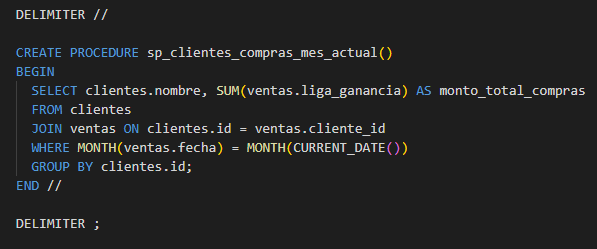


* Resultado



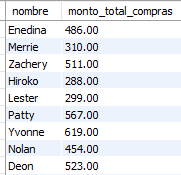
1. Obtener la lista de clientes que han comprado productos durante el mes actual y el monto total de sus compras:

* Consulta

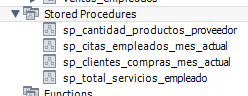




* Resultado



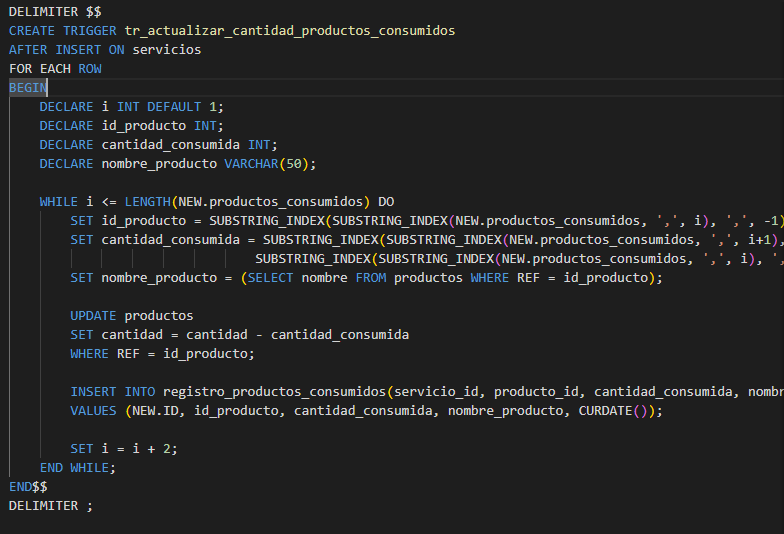
* Resultados totales



**Generar al menos 4 triggers:**

1. Este trigger utiliza la tabla productos y crea una tabla adicional llamada registro\_productos\_consumidos

* Consulta

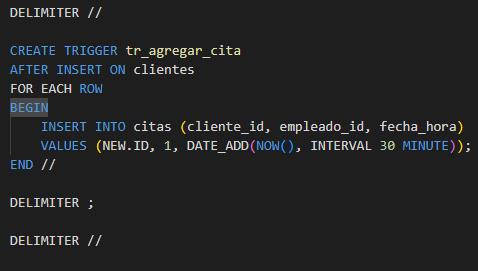


Resultado



1. Trigger para agregar una cita por defecto de 30 minutos cuando se inserta un cliente en la tabla de citas:

* Consulta

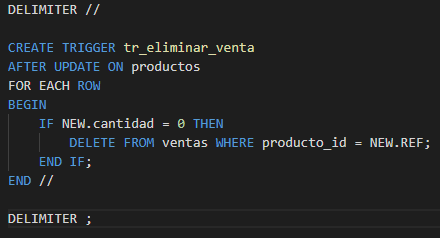


* Resultado



1. Trigger para eliminar una venta cuando se actualiza la cantidad a cero en la tabla de productos:

* Consulta

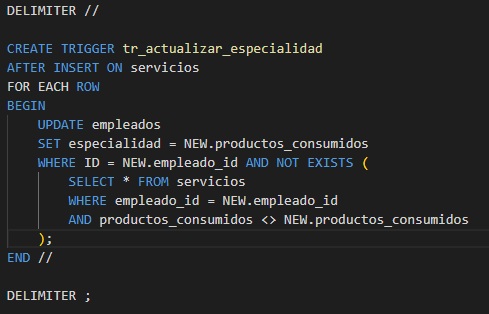


* Resultado



1. Trigger para actualizar la especialidad de un empleado si ha realizado un servicio de una nueva especialidad:

* Consulta



* Resultado

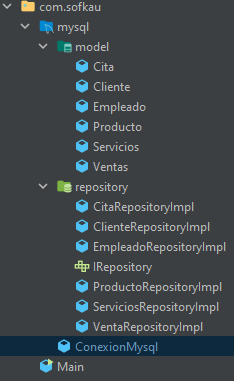


**Poblar la base de datos (50 registros por tabla) utilizando una conexión desde Java:**

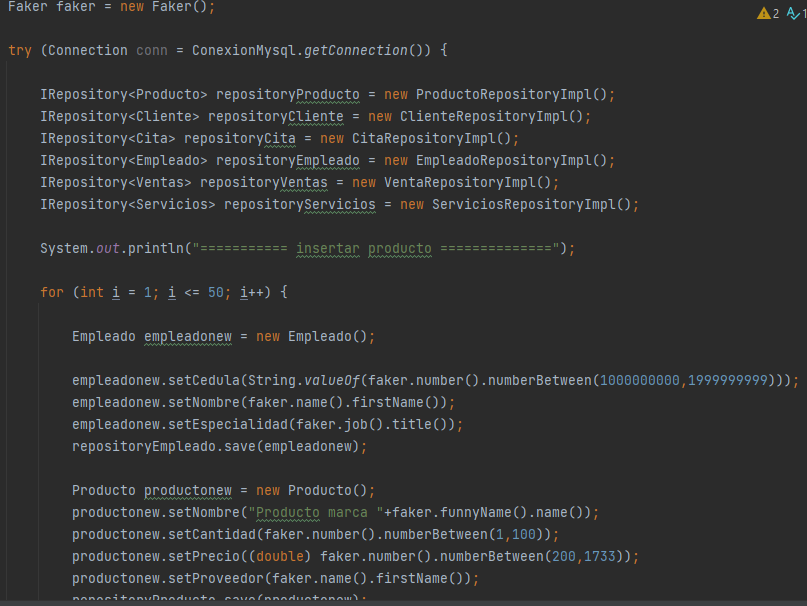
* **Conexión mysql**

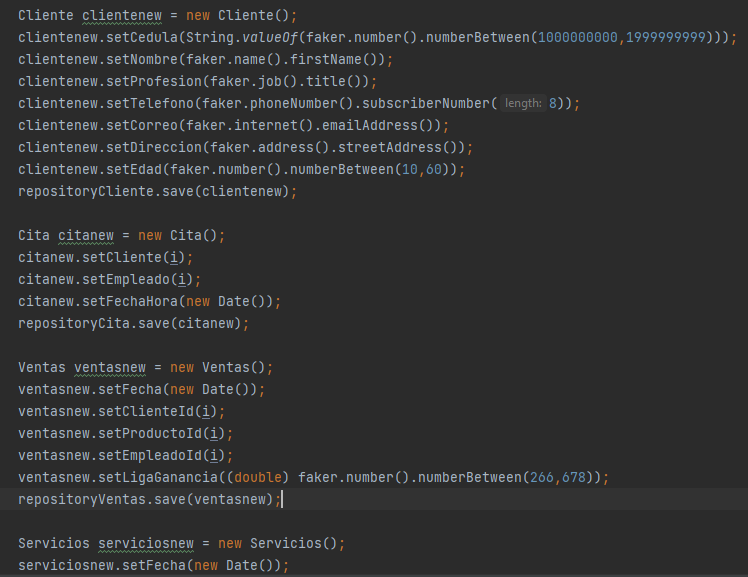


* **Creación de una clase y repositorio por cada tabla**



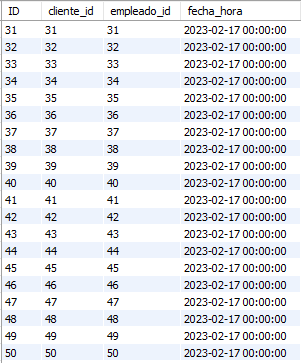
* **Iteración de 50 registros**



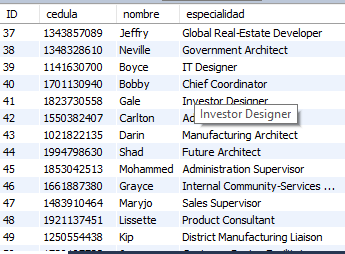


* **Resultados por cada tabla**

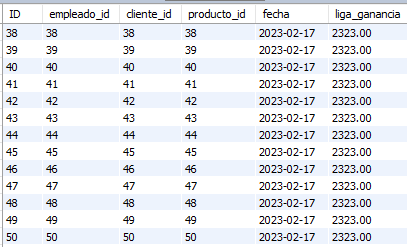
**(Citas) (Clientes**



**(Empleados) (Productos)**



**(Ventas) (Servicios)**



**¿ Está conforme con el resultado obtenido según el contexto o cree que hubiera obtenido un mejor resultado con una base de datos no relacional?**

En cuanto a la implementación del modelo de base de datos para la barbería, considero que el modelo relacional es adecuado y cumple con los requisitos planteados en el ejercicio. La estructura de las tablas y sus relaciones permite una gestión eficiente de la información y la realización de consultas y operaciones de forma sencilla.

Sin embargo, en algunos casos una base de datos no relacional puede ser una mejor opción, por ejemplo, en situaciones en las que se requiera un almacenamiento escalable y flexible, o en sistemas que manejen grandes cantidades de datos no estructurados. En este caso, si la barbería llegara a manejar una cantidad de datos significativamente mayor, una base de datos no relacional podría ser más adecuada.

En resumen, considero que el modelo relacional utilizado en este ejercicio es adecuado para las necesidades actuales de la barbería, pero podría ser necesario considerar una base de datos no relacional en caso de que las necesidades de almacenamiento y procesamiento de datos evolucionen en el futuro.

**Anexos:**

* + **barberia.sql**
  + **consultas10.sql**
  + **normalizacion\_barberia.sql**
  + **procedimiento\_barberia.sql**
  + **trigger\_barberia.sql**
  + **view\_barberia.sql**
  + **Barbería\_E-R.jpeg**