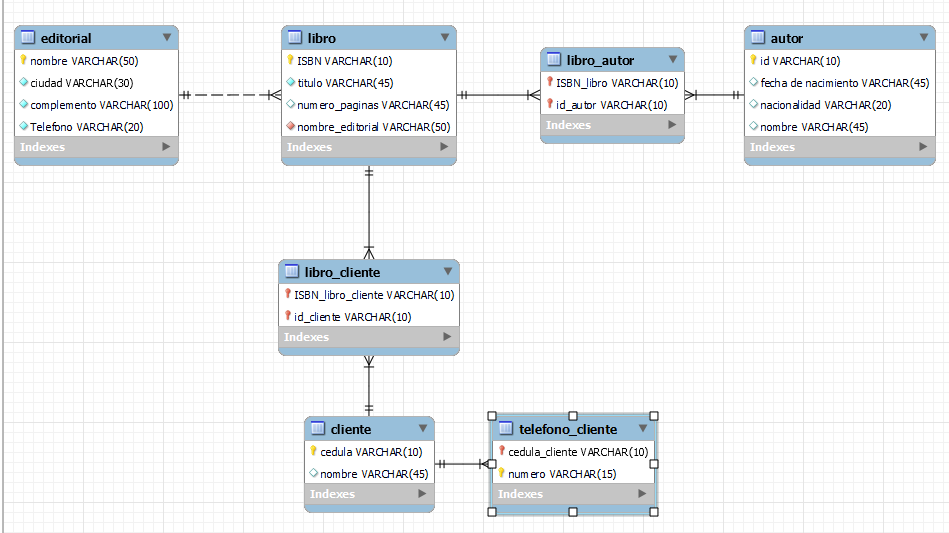
**PRIMERA ACTIVIDAD**

Genero el schema con el script que se encuentra en el repositorio:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Le aplico ingeniería en reversa para tener mejor visibilidad de las tablas, el tipo de datos de los atributos y como se relacionan entre ellas, y así facilitarme la creación de los registros.



Analizando el diagrama se puede observar:

* Las tablas que no tienen llaves foráneas son: editorial, autor y cliente, por ende, en estas se deben crear los primeros registros para evitar problemas con las relaciones.
* Para la tabla libro\_autor, se debe tener registros en las tablas libro y autor.
* Para la tabla libro, se debe tener registros en la tabla editorial.
* Para la tabla libro\_cliente, se debe tener registros en las tablas libro y cliente.
* Para la tabla teléfono\_cliente, se debe tener registros en la tabla cliente.

Con esta información se decide ingresar los registros en el siguiente orden:

Tabla autor:  
Se generan 5 registros

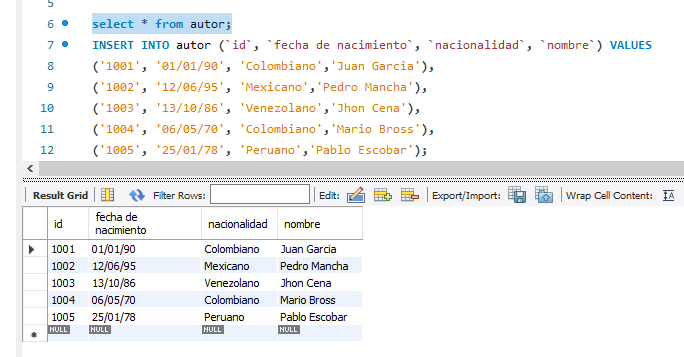


Tabla editorial:  
Se generan 4 registros

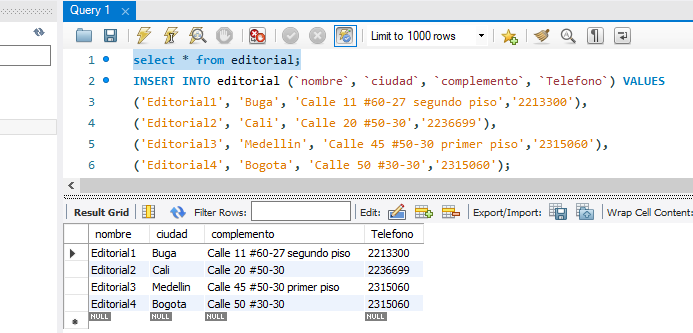


Tabla libro:  
Se generan 20 registros

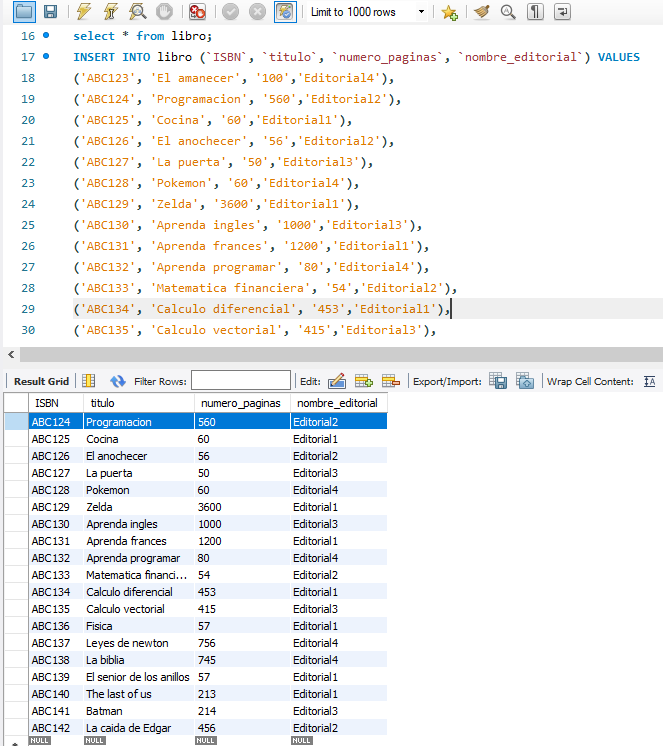


Tabla libro\_autor  
Se generan 10 registros

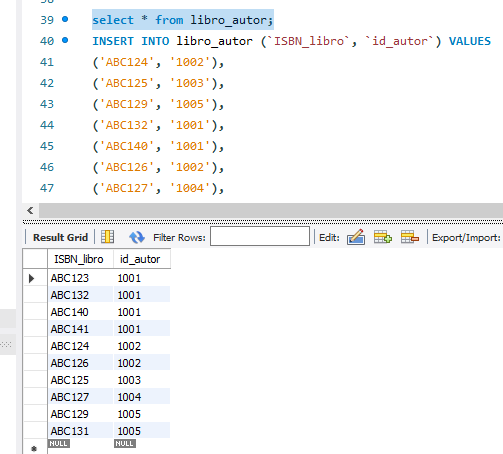


Tabla cliente  
Se crean 7 registros

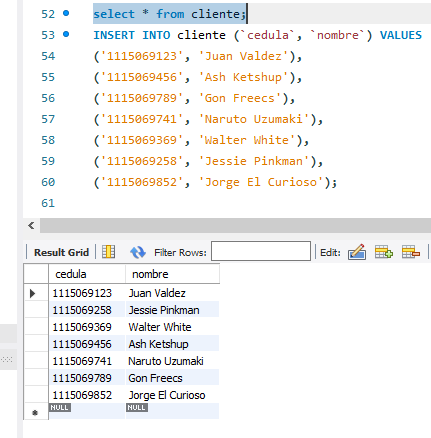


Tabla teléfono\_cliente  
Se crean 12 registros

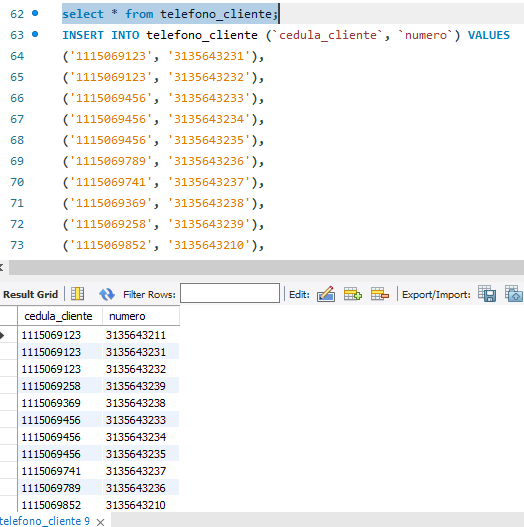
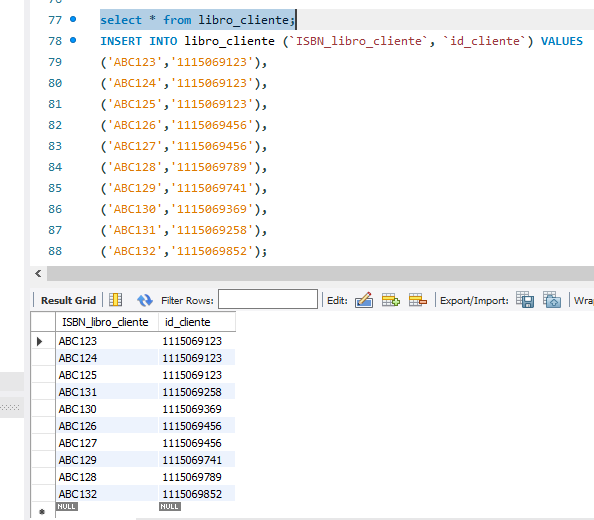
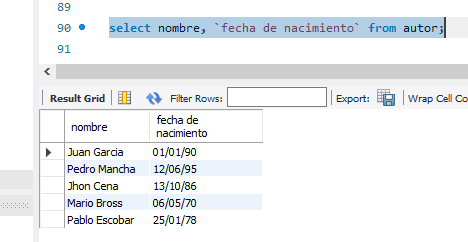


Tabla libro\_cliente  
Se generan 10 registros



**Consultas:**

**Nombre y fecha de nacimiento de los escritores**  
select nombre, `fecha de nacimiento` from autor;



**Cantidad de libros vendidos diferentes**  
select count(distinct ISBN\_libro\_cliente) as "Compras totales" from libro\_cliente;  
Aplicamos **distinct** en el ISBN del libro para que no cuente los libros repetidos.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

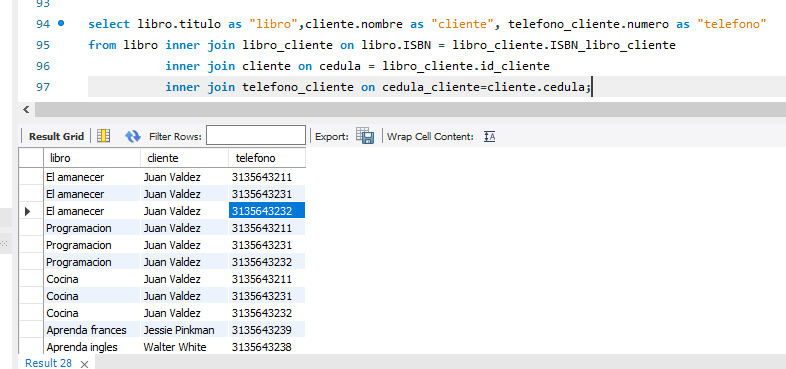
**Cliente y número de teléfono**  
select libro.titulo as "libro",cliente.nombre as "cliente", telefono\_cliente.numero as "telefono"

from libro inner join libro\_cliente on libro.ISBN = libro\_cliente.ISBN\_libro\_cliente

inner join cliente on cedula = libro\_cliente.id\_cliente

inner join telefono\_cliente on cedula\_cliente=cliente.cedula;

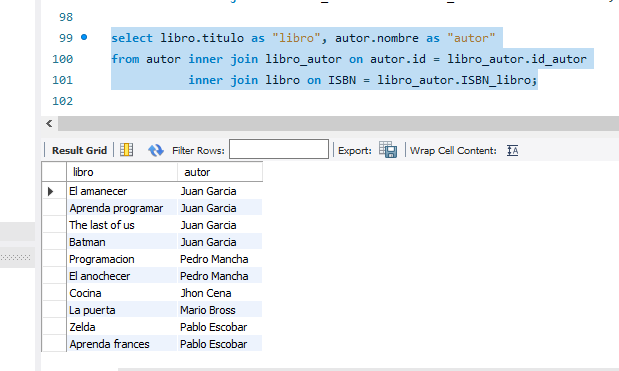
Un cliente al poder tener más de un numero se generan registros repetidos en la búsqueda.



**Nombre de libro y sus autores**  
select libro.titulo as "libro", autor.nombre as "autor"

from autor inner join libro\_autor on autor.id = libro\_autor.id\_autor

inner join libro on ISBN = libro\_autor.ISBN\_libro;



**El nombre de las editoriales que han vendido libros**  
select distinct libro.nombre\_editorial as "editorial"

from libro inner join libro\_cliente on libro.ISBN = libro\_cliente.ISBN\_libro\_cliente;

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

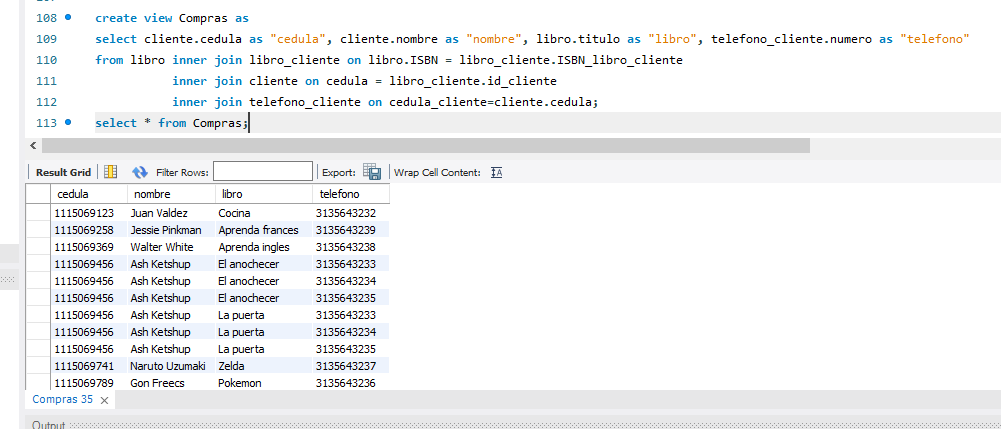
**Vista1**  
**Compras:** Esta vista contiene la información importante de todas las compras que se han realizado en la librería, podría utilizarse para generar una factura.  
create view Compras as

select cliente.cedula as "cedula", cliente.nombre as "nombre", libro.titulo as "libro", telefono\_cliente.numero as "telefono"

from libro inner join libro\_cliente on libro.ISBN = libro\_cliente.ISBN\_libro\_cliente

inner join cliente on cedula = libro\_cliente.id\_cliente

inner join telefono\_cliente on cedula\_cliente=cliente.cedula;



**Vista2  
InfoLibro:** Contiene toda la información necesaria de los libros, sus títulos, autores y editoriales, esta vista puede ser muy utilizada cuando alguien quiera saber donde puede conseguirse el libro buscado.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**SEGUNDA ACTIVIDAD**

Descargo la imagen del modelo relacional para tener una guía con la creación del modelo en SQL (se ve un poco borrosa pero se alcanza a entender).

Diagrama

Descripción generada automáticamente

Analizando el diagrama se pude observar:

* Las tablas que no dependen de otras son: tb\_medicamento, tb\_medico y tb\_procedimiento.
* Las tablas teléfono\_paciente y tb\_factura dependen de tb\_paciente.
* La tabla tb\_paciente necesita de tb\_procedimiento.
* La tabla dll\_paciente\_medicamento necesita de tb\_paciente y tb\_medicamento.
* La tabla dll\_medico\_procedimiento necesita de tb\_procedimiento y tb\_medico.
* Hay un error con la tabla que debería contener el teléfono del médico, este error será corregido en el diagrama creado en SQL, la tabla teléfono\_medico y tb\_enfermero dependen de tb\_medico.
* Pasa el mismo error con la tabla de teléfonos de los enfermeros, el error será corregido en el diagrama creado en SQL, la tabla teléfono\_enfermero depende de enfermero.

Se procede con la creación de tablas y el ingreso de registros al mismo tiempo.

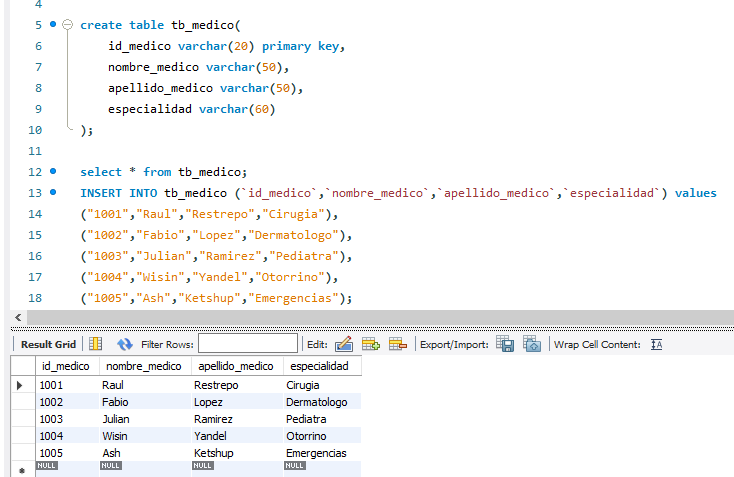
Con este análisis se concluye que el orden de creación de las tablas y de ingresos será el siguiente:

**Se inicia con la creación del schema:**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

**Tabla tb\_medico**

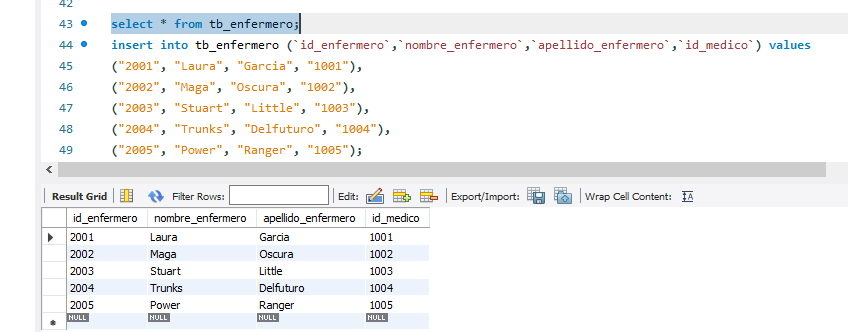


**Tabla teléfono\_medico**

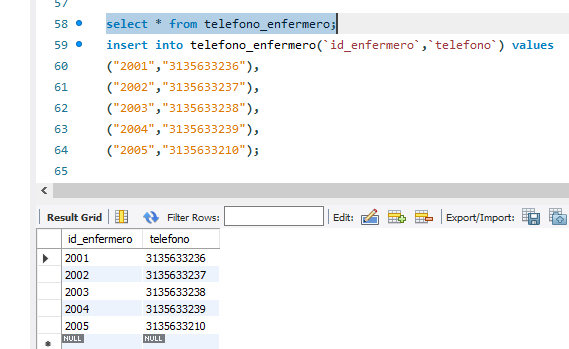
Interfaz de usuario gráfica

Descripción generada automáticamente

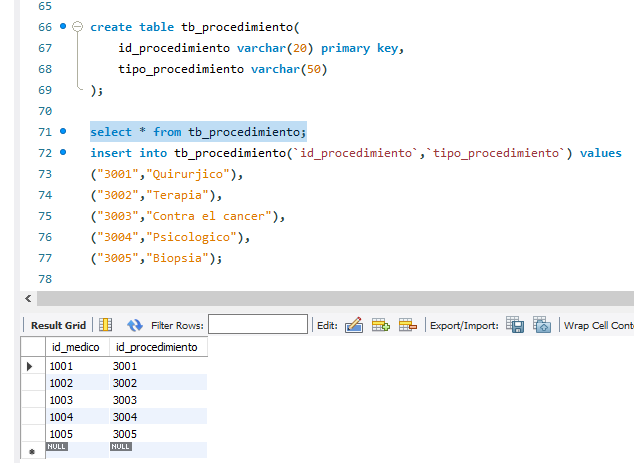
**Tabla tb\_enfermero**



**Tabla tlefono\_enfermero**



**Tabla procedimiento**



**Tabla dll\_\_medico\_procedimiento**

Interfaz de usuario gráfica, Texto

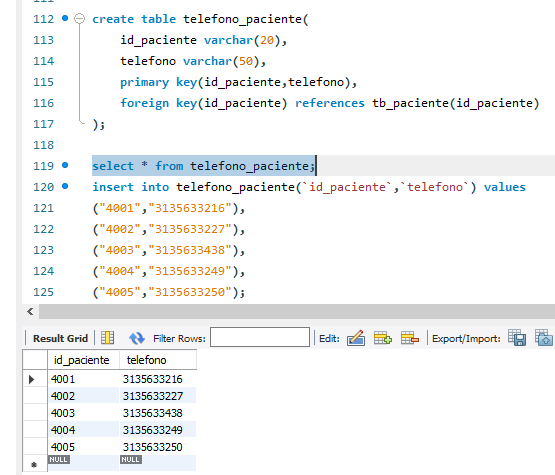
Descripción generada automáticamente

**Tabla tb\_paciente**

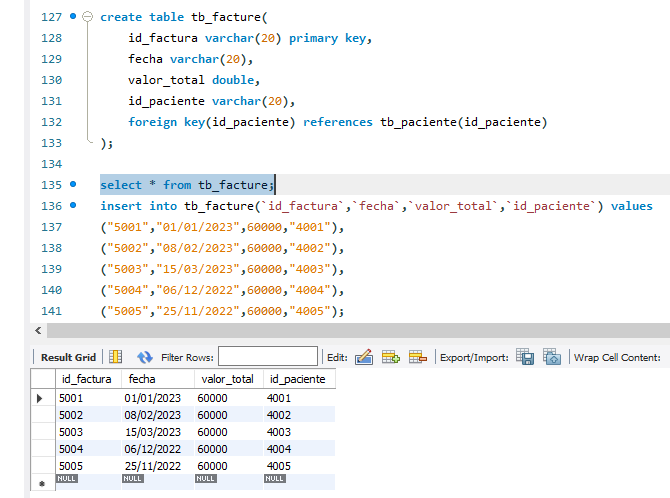
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

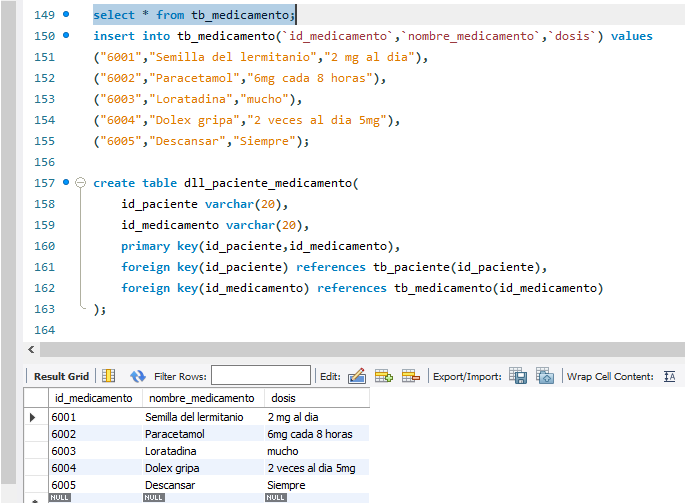
**Tabla teléfono\_paciente**



**Tabla tb\_facture**



**Tabla tb\_medicamento**

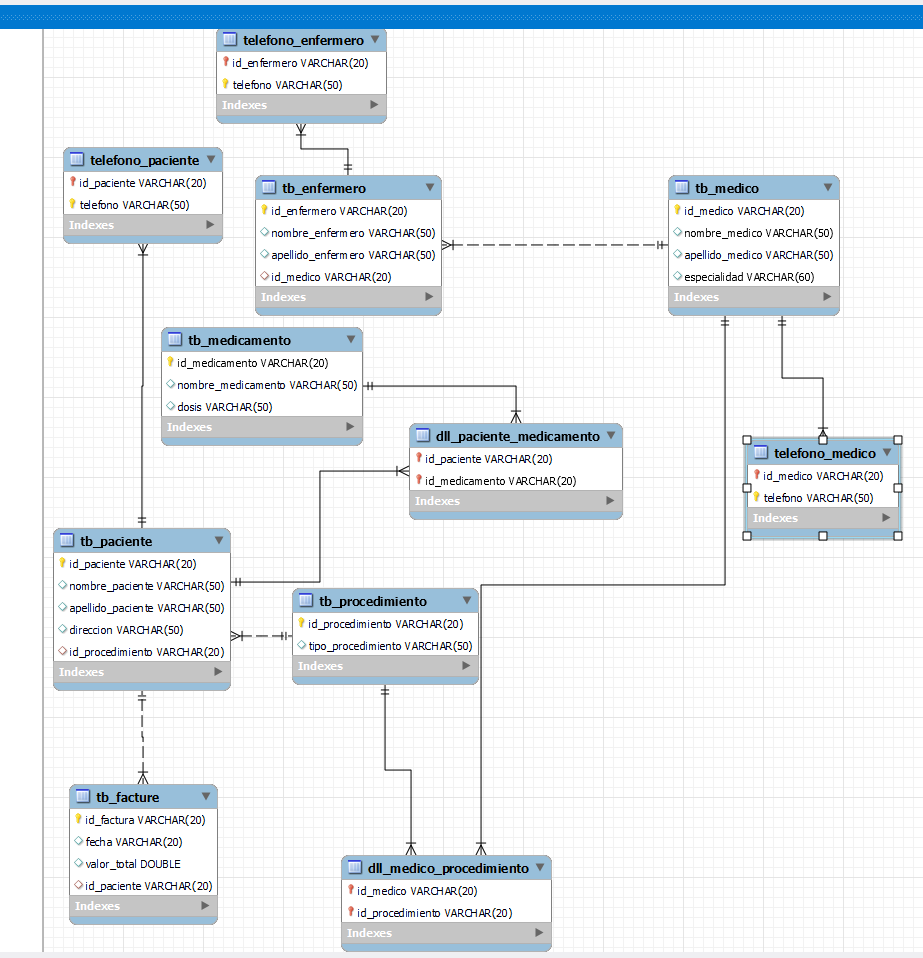


**Tabla dll\_paciente\_medicamento**

Interfaz de usuario gráfica, Texto

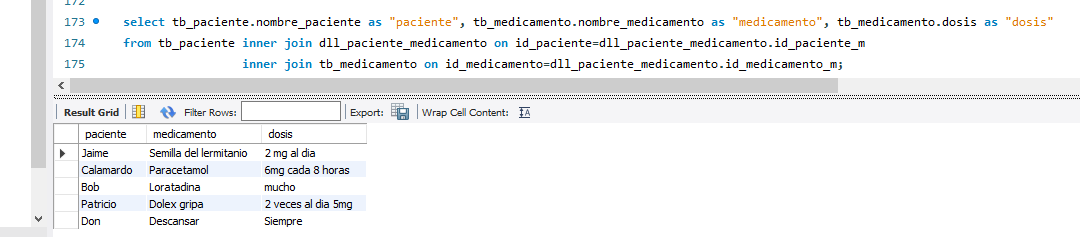
Descripción generada automáticamente con confianza media

Aplicando ingenieria inversa podemos ver el diagrama en workbench



**Consultas:**

**Realice una consulta que me permita conocer que medicamentos a tomado cada paciente y la dosis suministrada.**



Uso inner join para conectar las 3 tablas necesarias y muestro los datos requeridos.  
Conecto la tabla tb\_paciente con dll\_paciente\_medicamento mediante el atributo id\_paciente, y después conecto la consulta generada con la tabla tb\_medicamento mediante el atributo id\_medicamento.  
Se hacen esas conexiones ya que la tabla tb\_paciente contiene el nombre del paciente, la tabla dll\_medicamento\_paciente es necesaria para conectar las dos tablas (tb\_paciente y tb\_medicamento), y la tabla tb\_medicamento contiene el nombre del medicamento y la dosis de este mismo.

**Realice una consulta que me permita conocer que enfermeros estuvieron en los procedimientos de los pacientes.**

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

Para mostrar esta consulta se deben usar 4 inner join, uno que conecte la tabla tb\_pacientes con tb\_procediminientos mediante el atributo id\_procedimiento, después se conecta la tabla generada con dll\_medico\_procedimiento mediante el atributo id\_procedimiento, después se debe conectar con la tabla tb\_medico mediante el id\_medico y por ultimo se conecta con la tabla enfermero mediante el atributo id\_medico.

**Vistas:**

**Vista1 DatosProcedimiento**Esta vista se crea con el fin de generar todos los datos importantes sobre un procedimiento, el doctor que lo realiza con su respectivo enfermero y el tipo de procedimiento que será realizado, y obviamente el paciente que lo va a recibir.

Interfaz de usuario gráfica, Aplicación, Word

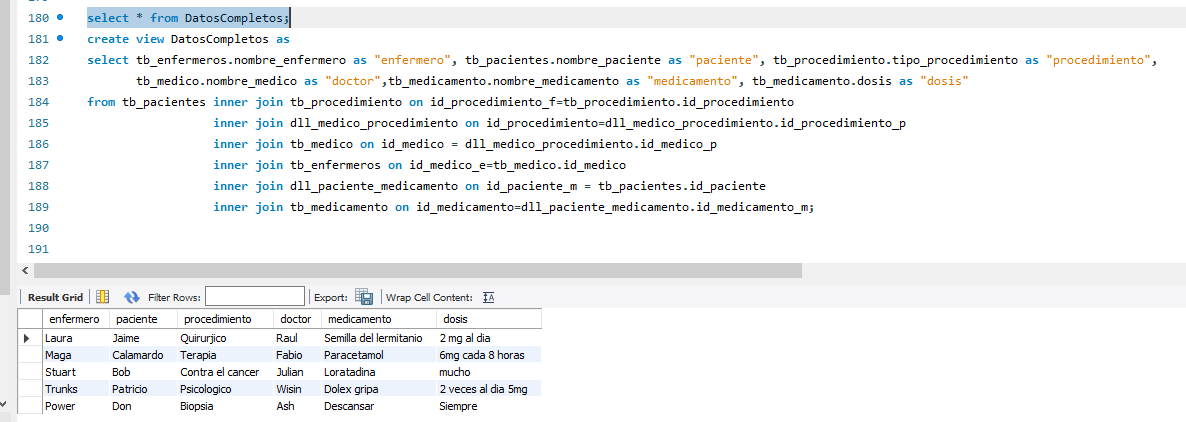
Descripción generada automáticamente

**Vista2 DatosPaciente**Esta vista se crea con la finalidad de tener en una tabla con toda la información sobre un paciente, desde sus doctores, su procedimiento, y sus medicamentos, puede ser útil para llevar una historia clínica del paciente.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

**Vista3 DatosCompletos**Se conectan casi todas las relaciones para guardar toda la información sobre un paciente, enfermeros, médicos, medicamentos y procedimiento.



**Pregunta final**

¿Qué le agregaría al modelo para dar más información y esa información cual sería?

**R)** Creo que lo mas importante al tener una clase factura es agregarle los costos a lo medicamentos y tener una tabla intermedia que podría llamarse pedido, que guarde el total a pagar, y un lista de medicamentos con su respectiva cantidad.

También se podría agregar el salario a los trabajadores, costo a los procedimientos y en realidad se podrían ser muchísimas más cosas que complicarían muchísimo la actividad.

**TERCERA ACTIVIDAD**

**Procedimientos actividad 1**

**Agregar:**

Interfaz de usuario gráfica, Texto, Aplicación

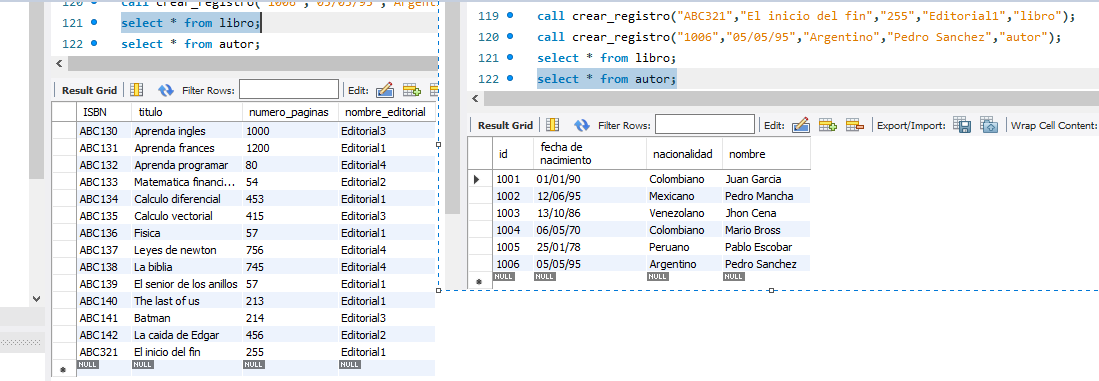
Descripción generada automáticamente con confianza media

El procedimiento crear\_registro recibe 5 parametros, 4 de estos indican los datos que serán ingresados y el ultimo parámetro llamado nombre\_tabla indica en que tabla será agregado el nuevo registro. En este caso solo funciona con las tablas autor y libro.

Imagen que contiene Texto

Descripción generada automáticamente

Al ejecutar estos comandos debería crear el registro con llave primaria ABC321 en la tabla libro, y el registro con llave primaria 1006 en la tabla autor.



**Buscar:**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

El procedimiento buscar\_libro recibe como parámetro el titulo de un libro y hacer la consulta en la tabla libro.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Eliminar:**

Imagen que contiene Texto

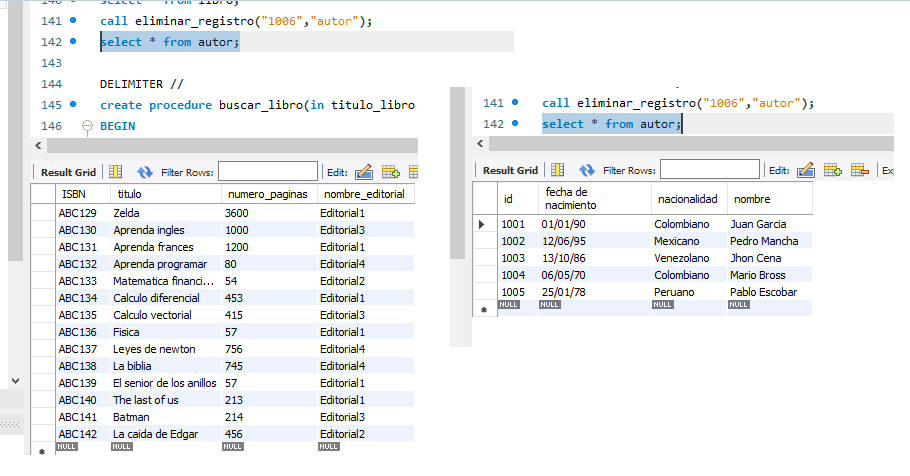
Descripción generada automáticamente

El proceso eliminar\_registro recibe un identificador (la llave primaria de una tabla) y el nombre la tabla de la cual desea eliminar el registro.

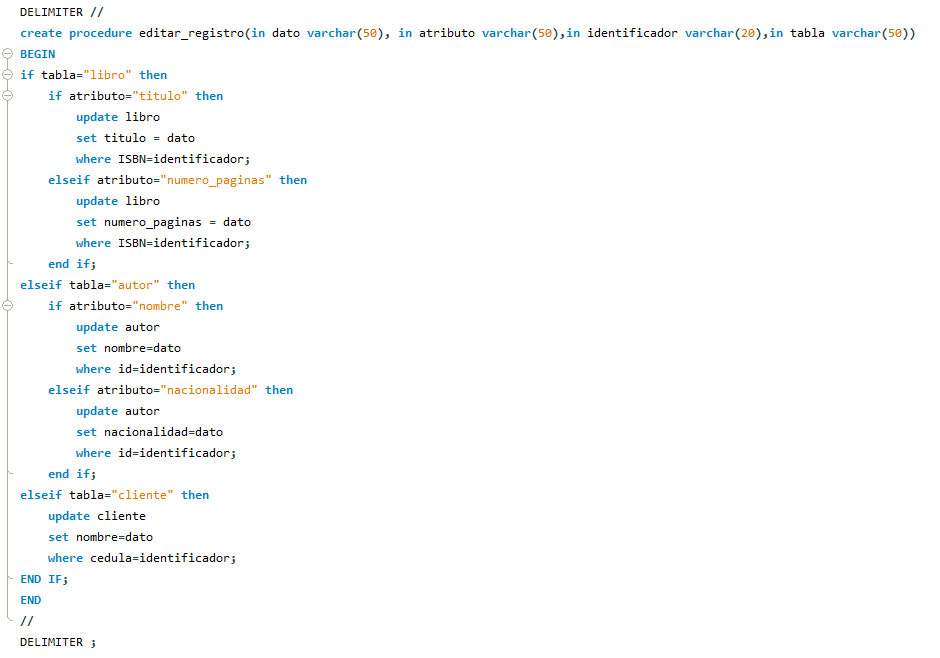
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

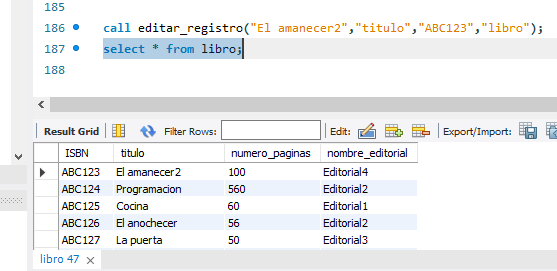
Al ejecutar el procedimiento con esos datos, eliminaremos los registros creados en el ejemplo anterior.



**Editar:**

****

Este procedimiento recibe 4 parámetros, la tabla que se va a modificar, la columna que se va a modificar, el nuevo dato que será ingresado y el identificador del registro que se desea modificar.



Se puede observar como cambio el titulo del libro (ABC123), antes era el amanecer y ahora es el amanecer2.

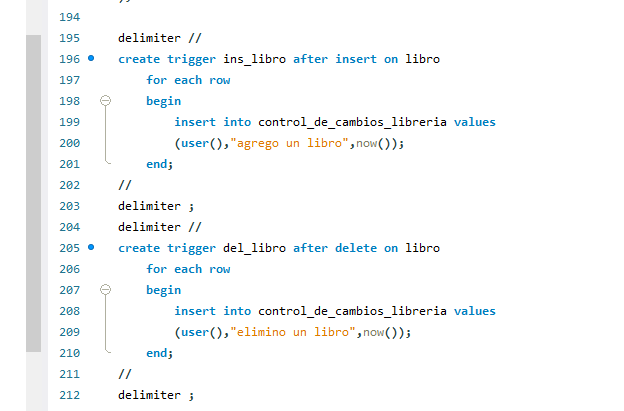
**Triggers**

Inicio creando la tabla control\_de\_cambio\_libreria

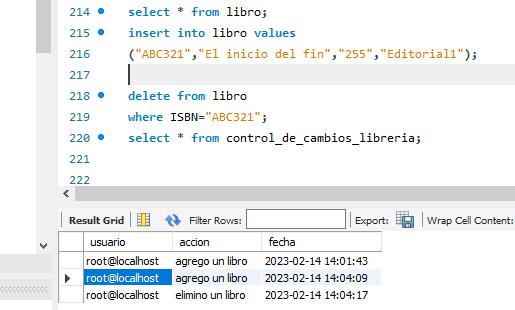
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Creación de triggers



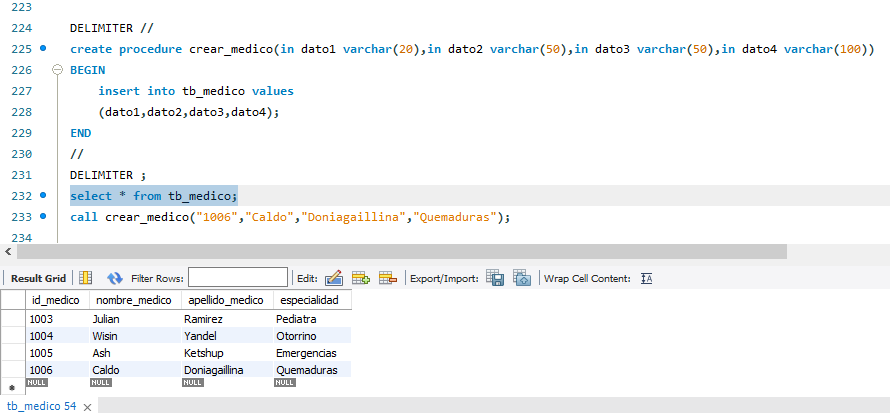
Se crean los triggers ins\_libro y del\_libro, uno se encarga de los registros de creación y el otro de los registros de eliminación respectivamente, y estos registros son guardados en la tabla control\_de\_cambio\_libreria.



Como se puede observar, funcionan perfectamente.

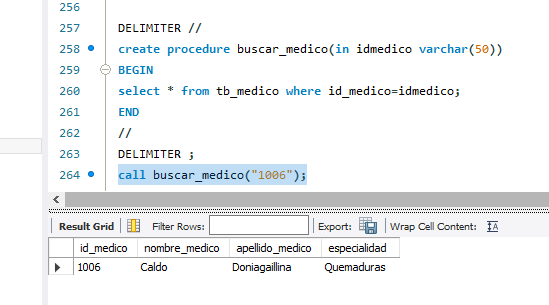
**Procedimientos actividad 2:**

**Agregar:**

****

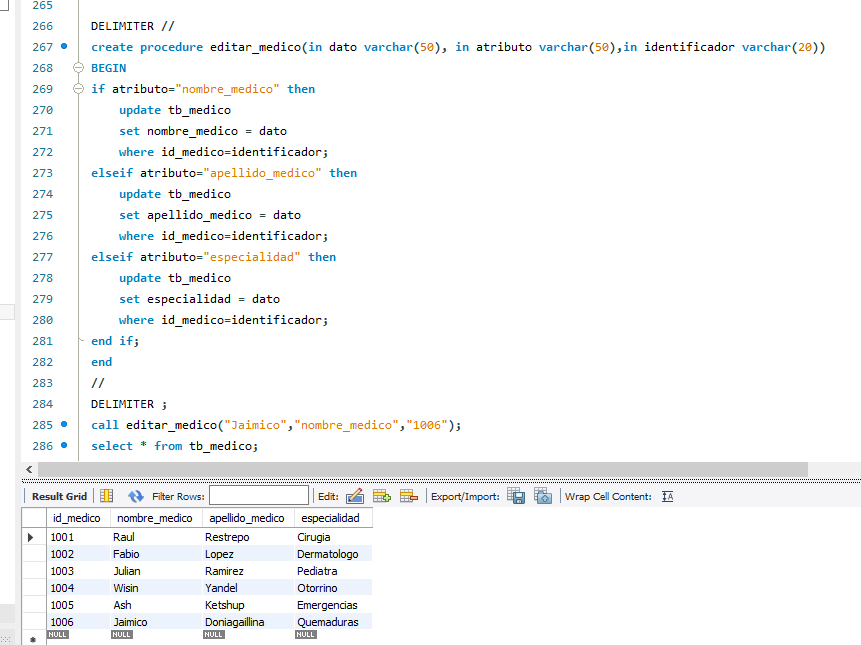
Se crea el procedimiento agregar\_medico y se genera un registro utilizándolo.

**Buscar:**

****

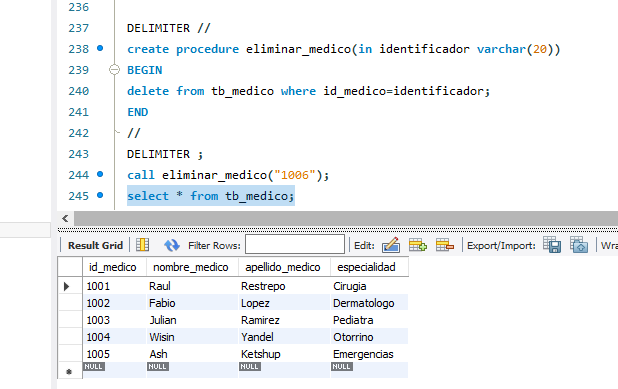
Se crea el procedimiento buscar\_medico y se consulta al medico creado en el ejemplo anterior, utilizando el procedimiento.

**Editar:**

****

Este procedimiento puede editar una de las columnas que no sea el identificador de la tabla tb\_medico, recibe como parámetros el identificador del médico, el atributo a modificar y el dato con el que reemplazara la información que estaba antes.

**Eliminar:**



Se crea el procedimiento eliminar\_medico y se ejecuta para eliminar al medico con id=1006 que fue creado en los ejemplos anteriores.

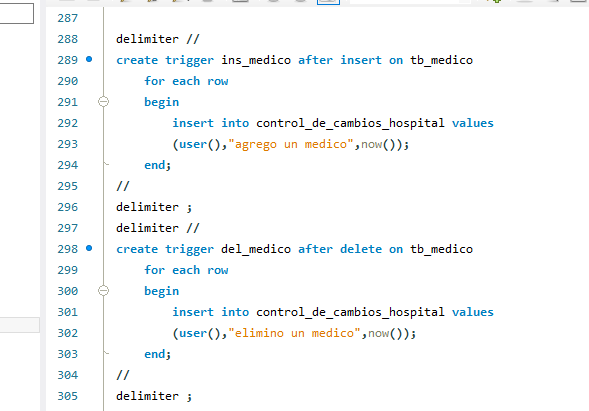
**Triggers**

Se crea la tabla control\_de\_cambios\_hospital

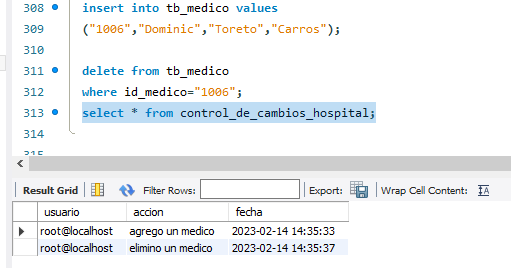
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Se crean ambos triggers



Estos trisggers guardan un registro en la tabla de control\_de\_cambios\_hospital, cuando un usuario crea un nuevo médico o lo elimina.



Se crea y se elimina al medico con id=1006 para ejemplificar el funcionamiento correcto de los triggers.