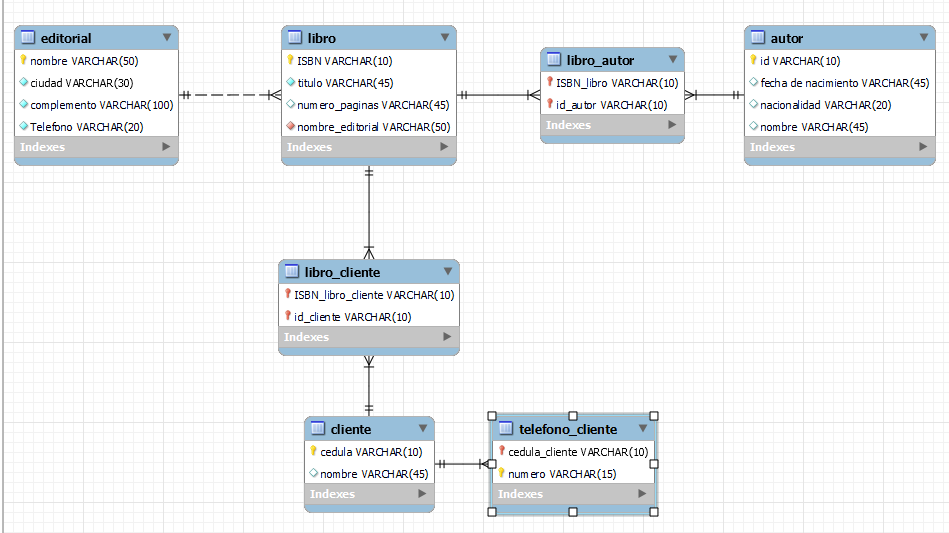
**PRIMERA ACTIVIDAD**

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

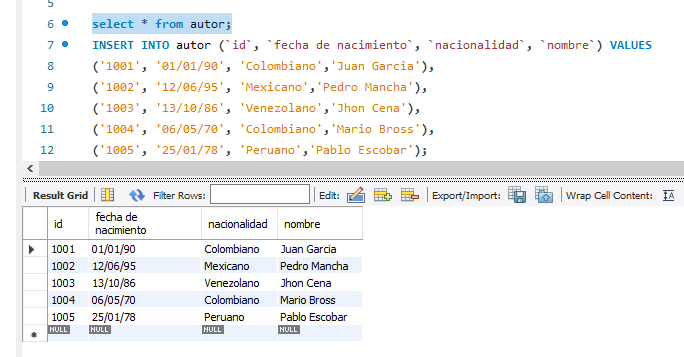


Tabla editorial:  
Se generan 4 registros

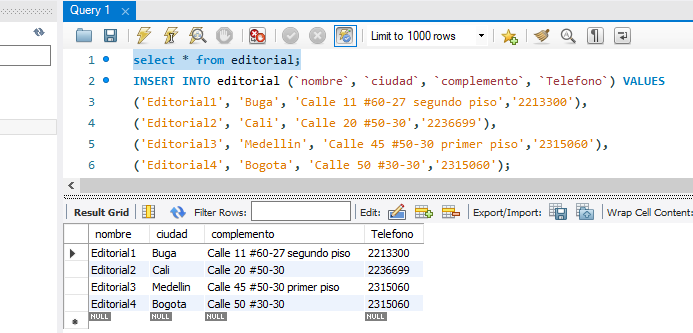


Tabla libro:  
Se generan 20 registros

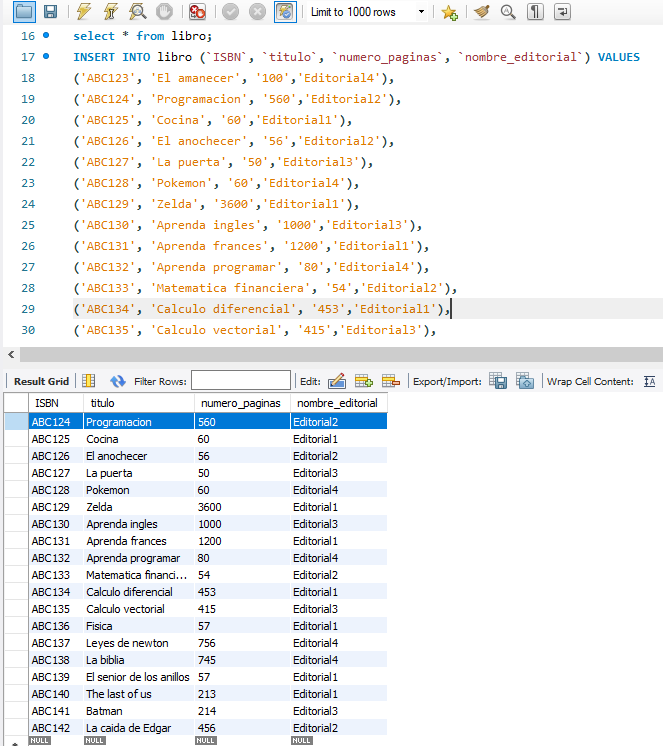


Tabla libro\_autor  
Se generan 10 registros

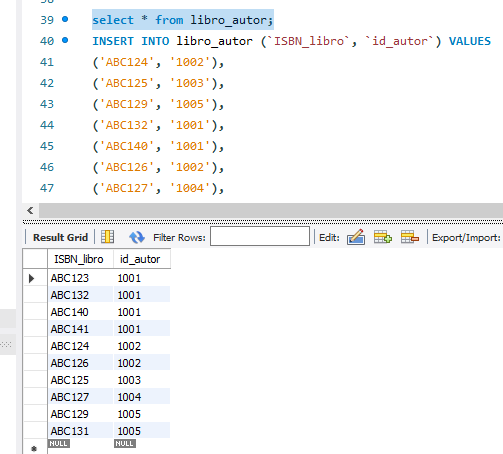


Tabla cliente  
Se crean 7 registros

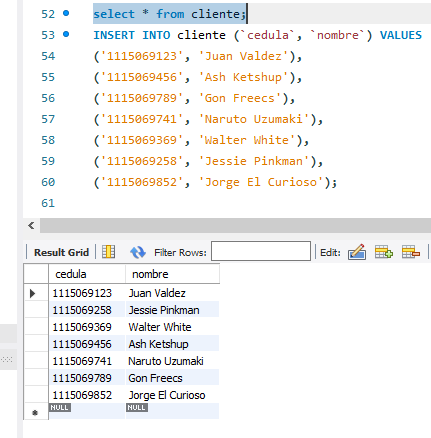


Tabla teléfono\_cliente  
Se crean 12 registros

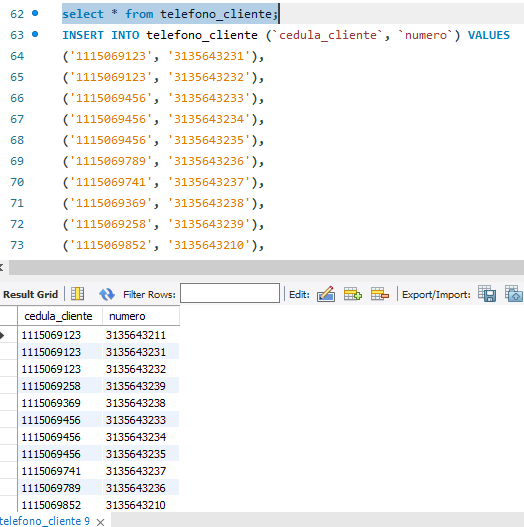
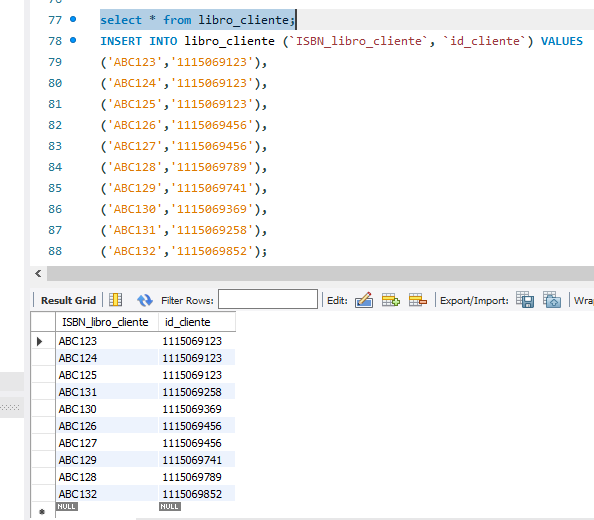
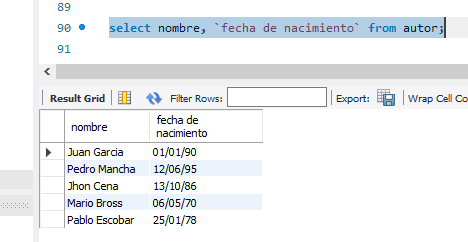


Tabla libro\_cliente  
Se generan 10 registros



**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

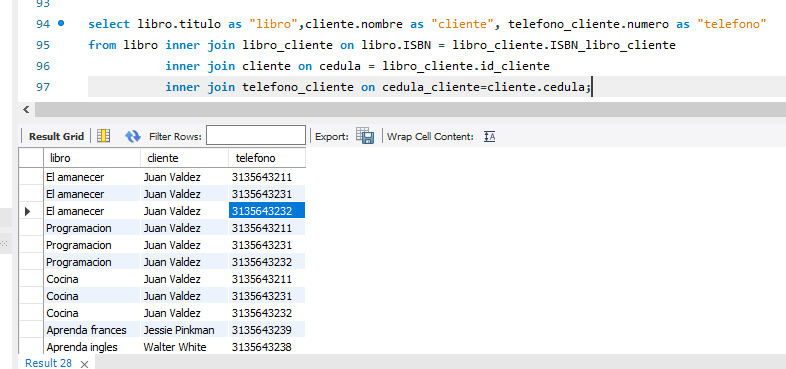
**Cliente y número de teléfono**  
select libro.titulo as "libro",cliente.nombre as "cliente", telefono\_cliente.numero as "telefono"

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;

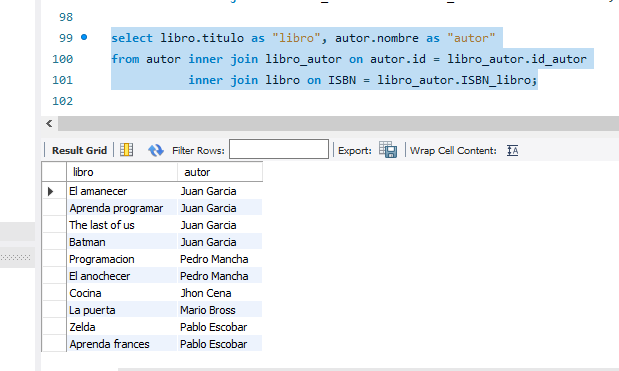
Un cliente al poder tener más de un numero se generan registros repetidos en la búsqueda.



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

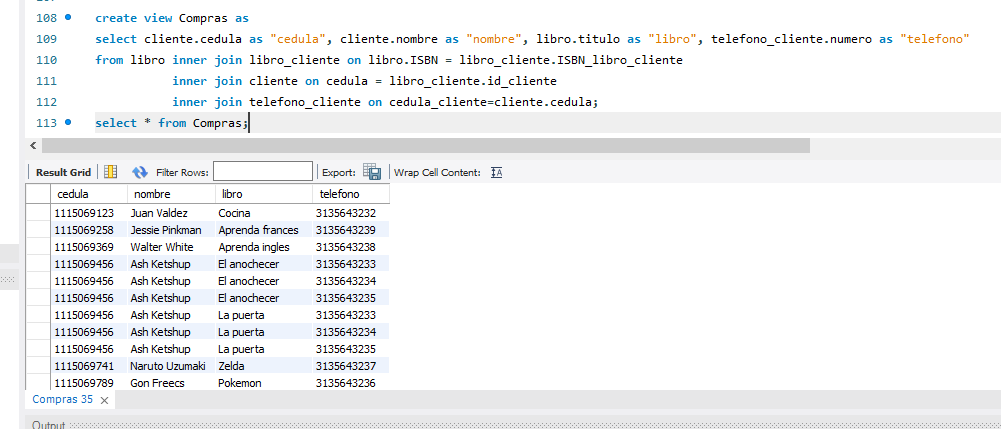
**Vista1**  
**Compras:** Esta vista contiene la información importante de todas las compras que se han realizado en la librería, podría utilizarse para generar una factura.  
create view Compras as

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.

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 medico

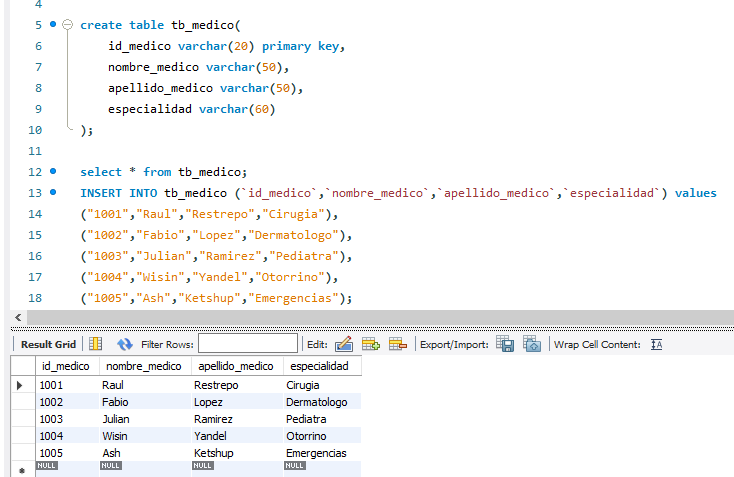


Tabla teléfono\_medico

Interfaz de usuario gráfica

Descripción generada automáticamente

Tabla tb\_enfermero

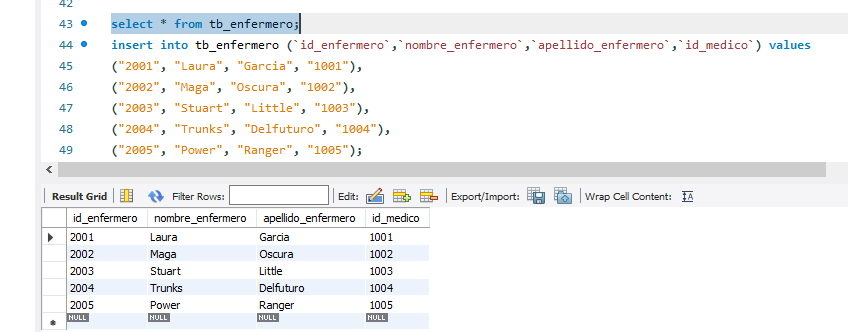


Tabla tlefono\_enfermero

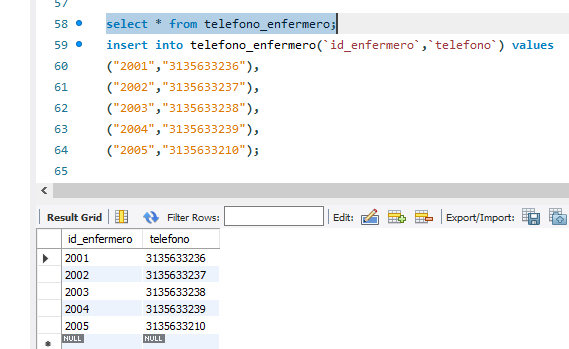


Tabla procedimiento

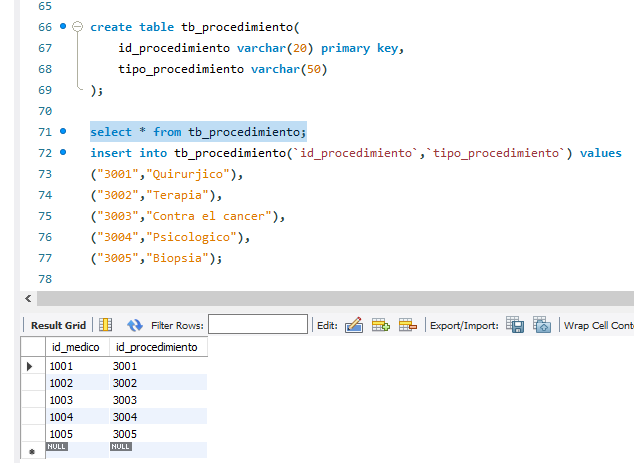


Tabla dll\_\_medico\_procedimiento

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Tabla tb\_paciente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Tabla teléfono\_paciente

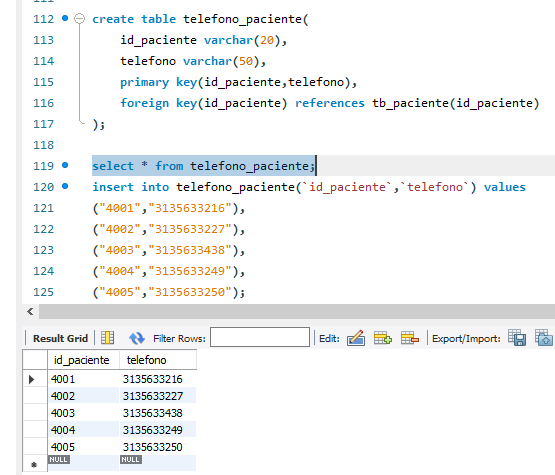


Tabla tb\_facture

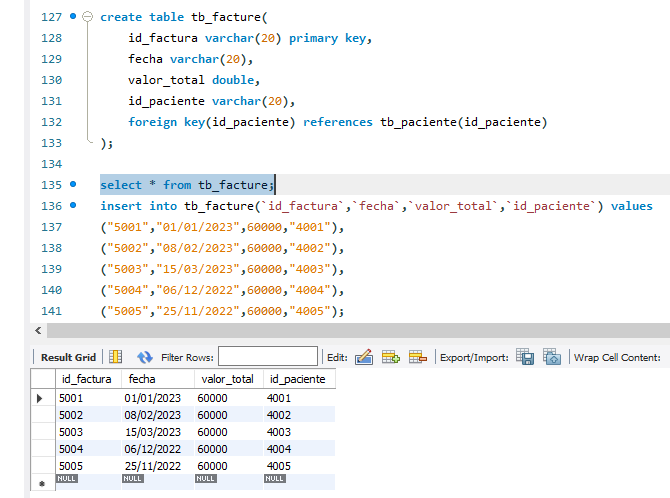


Tabla tb\_medicamento

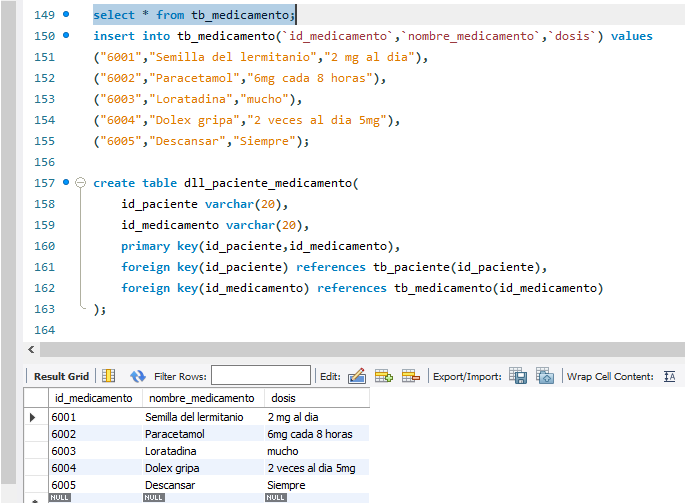


Tabla dll\_paciente\_medicamento

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media