

TALLER BASE DE DATOS

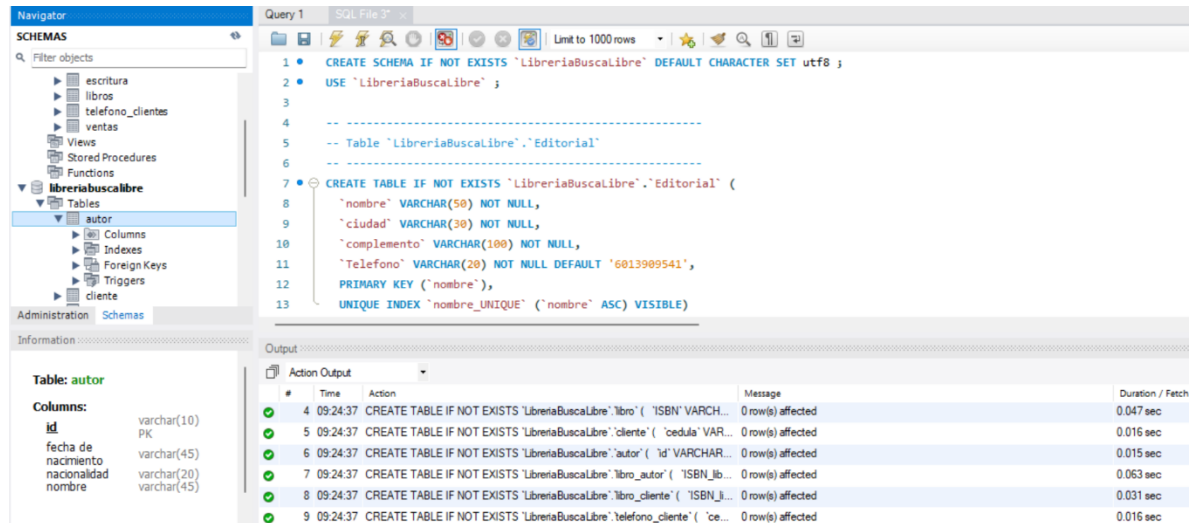
# BD LIBRERÍA Y HOSPITAL

Descripción y paso a paso

## CREACION DE BD LibreriaBuscaLibre

### 1. Creación de las tablas:

- Se crearon las tablas necesarias en la base de datos **LibreriaBuscaLibre** utilizando el lenguaje SQL. Las tablas creadas fueron **Editorial**, **libro**, **cliente**, **autor**, **libro\_autor**, **libro\_cliente** y **telefono\_cliente**. Estas tablas representan entidades como editoriales, libros, clientes, autores y relaciones entre ellas.



```
1 CREATE SCHEMA IF NOT EXISTS 'LibreriaBuscaLibre' DEFAULT CHARACTER SET utf8 ;
2 USE 'LibreriaBuscaLibre' ;
3
4
5 -- Table 'LibreriaBuscaLibre'. 'Editorial'
6
7 CREATE TABLE IF NOT EXISTS 'LibreriaBuscaLibre'. 'Editorial' (
8     'nombre' VARCHAR(50) NOT NULL,
9     'ciudad' VARCHAR(30) NOT NULL,
10    'complemento' VARCHAR(100) NOT NULL,
11    'Telefono' VARCHAR(20) NOT NULL DEFAULT '6013909541',
12    PRIMARY KEY ('nombre'),
13    UNIQUE INDEX 'nombre_UNIQUE' ('nombre' ASC) VISIBLE)
```

#	Time	Action	Message	Duration / Fetch
4	09:24:37	CREATE TABLE IF NOT EXISTS 'LibreriaBuscaLibre'. 'libro' ( 'ISBN' VAR...	0 row(s) affected	0.047 sec
5	09:24:37	CREATE TABLE IF NOT EXISTS 'LibreriaBuscaLibre'. 'cliente' ( 'cedula' VAR...	0 row(s) affected	0.016 sec
6	09:24:37	CREATE TABLE IF NOT EXISTS 'LibreriaBuscaLibre'. 'autor' ( 'Id' VARCHAR...	0 row(s) affected	0.015 sec
7	09:24:37	CREATE TABLE IF NOT EXISTS 'LibreriaBuscaLibre'. 'libro_autor' ( 'ISBN_lib...	0 row(s) affected	0.063 sec
8	09:24:37	CREATE TABLE IF NOT EXISTS 'LibreriaBuscaLibre'. 'libro_cliente' ( 'ISBN_lib...	0 row(s) affected	0.031 sec
9	09:24:37	CREATE TABLE IF NOT EXISTS 'LibreriaBuscaLibre'. 'telefono_cliente' ( 'ce...	0 row(s) affected	0.016 sec

Se crea la base de datos y sus respectivas tablas con el script proporcionado en el ejercicio del repositorio en github







<https://github.com/Training2024SU/2024-C1-QA-BD-T04/blob/main/Script%20%20generado%20libreria.sql>

### 2. Inserción de datos en las tablas:

- Se insertaron registros en las tablas utilizando comandos SQL **INSERT INTO**. Se crearon registros para las editoriales, libros, clientes, autores y relaciones entre ellos. Estos datos ficticios sirven como ejemplo para realizar consultas y pruebas en la base de datos.

## AUTOR

```
-- -----  
-- Registro para la tabla Autor  
-- -----  
  
• INSERT INTO `LibreriaBuscaLibre`.`autor` (`id`, `fecha de nacimiento`, `nacionalidad`, `nombre`)  
VALUES  
( '1', '1968-07-10', 'Mexico', 'Jorge Volpi '),  
( '2', '1975-08-20', 'España', 'María Lopez'),  
( '3', '1990-03-15', 'Argentina', 'Pedro Martinez'),  
( '4', '1950-03-07', 'Colombia', 'Laura Restrepo'),  
( '5', '1942-08-02', 'Chile', 'Isabel Allende'),  
( '6', '1927-03-06', 'Colombia', 'Gabriel Garcia Marquez');
```

Result Grid				
Filter Rows: <input type="text"/>				
Edit:   				
Export/Import:  				
Wrap Cell Content: 				
	id	fecha de nacimiento	nacionalidad	nombre
▶	1	1968-07-10	Mexico	Jorge Volpi
	2	1975-08-20	España	María Lopez
	3	1990-03-15	Argentina	Pedro Martinez
	4	1950-03-07	Colombia	Laura Restrepo
	5	1942-08-02	Chile	Isabel Allende
	6	1927-03-06	Colombia	Gabriel Garcia Marquez
*	NULL	NULL	NULL	NULL

## CLIENTE

```
Registros-Tablas-BD-Libreria* x  
Limit to 1000 rows  
  
14  -- -----  
15  -- Registro para la tabla Cliente  
16  -- -----  
17  
18  • INSERT INTO `LibreriaBuscaLibre`.`cliente` (`cedula`, `nombre`)  
19  VALUES  
20  ( '1003049506', 'Juanse Gomez'), ( '50891234', 'Mariana Perez'),  
21  ( '34987687', 'Miguel Hernandez'), ( '1003047685', 'Ana Martinez'),  
22  ( '10987345', 'Luisa Perez'), ( '50897324', 'Carlos Sanchez'),  
23  ( '1067987234', 'Laura Diaz'), ( '309875670', 'Fernando Torres'),  
24  ( '10916347', 'Lucia Ramirez'), ( '1003046324', 'Sofia Gonzalez'),  
25  ( '1068945987', 'Daniel Lopez'), ( '10923485', 'Carmen Garcia'),  
26  ( '34960070', 'Diego Castro'), ( '1003049387', 'Ana Maria Suarez'),  
27  ( '28905432', 'Jorge Fernandez'), ( '50345987', 'Elena Navarro'),  
28  ( '34234109', 'Roberto Morales'), ( '18456890', 'Isabel Ruiz'),  
29  ( '50960987', 'Adriana Jimenez'), ( '1006986954', 'Raul Medina');  
30
```

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
cedula	nombre				
28905432	Jorge Fernandez				
309875670	Fernando Torres				
34234109	Roberto Morales				
34960070	Diego Castro				
34987687	Miguel Hernandez				
50345987	Elena Navarro				
50891234	Mariana Perez				
50897324	Carlos Sanchez				
50960987	Adriana Jimenez				
NULL	NULL				

## EDITORIAL

-- Registro para la tabla Editorial

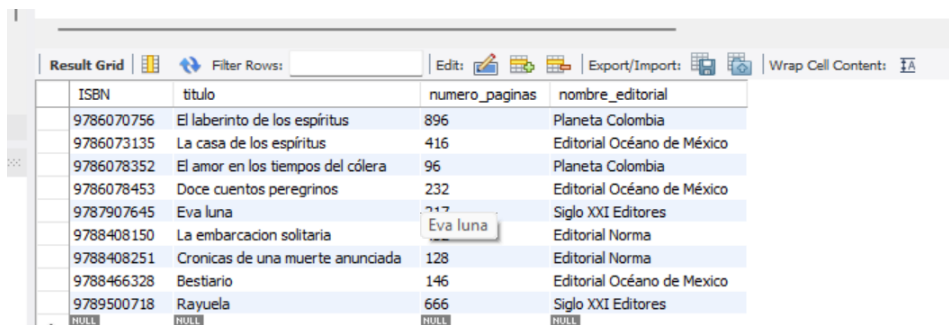
- INSERT INTO** `LibreriaBuscaLibre`.`Editorial` (`nombre`, `ciudad`, `complemento`, `Telefono`)
   
**VALUES**
  
( 'Editorial Norma', 'Bogotá', 'Carrera 58 # 127 - 59', '+57 3004534567'),
   
( 'Planeta Colombia', 'Bogotá', 'Calle 69A # 6 - 61', '+57 3214560987'),
   
( 'Editorial Océano de México', 'Ciudad de México', 'Av. Presidente Masaryk 61', '+5252811141'),
   
( 'Fondo de Cultura Económica', 'Ciudad de México', 'Avenida Universidad 975', '+52 55 4155 0000'),
   
( 'Siglo XXI Editores', 'Buenos Aires', 'Av. Pueyrredón 1539', '+54 11 4821-8611'),
   
( 'Sudamericana', 'Buenos Aires', 'Av. Leandro N. Alem 720', '+54 11 4319-3800'),
   
( 'Grupo Editorial Penguin Random House', 'Buenos Aires', 'Alicia M. de Justo 740', '+54 11 4309-7500');

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
nombre	ciudad	complemento	Telefono		
Editorial Norma	Bogotá	Carrera 58 # 127 - 59	+57 3004534567		
Editorial Océano de México	Ciudad de México	Av. Presidente Masaryk 61	+5252811141		
Fondo de Cultura Económica	Ciudad de México	Avenida Universidad 975	+52 55 4155 0000		
Grupo Editorial Penguin Random House	Buenos Aires	Alicia M. de Justo 740	+54 11 4309-7500		
Planeta Colombia	Bogotá	Calle 69A # 6 - 61	+57 3214560987		
Siglo XXI Editores	Buenos Aires	Av. Pueyrredón 1539	+54 11 4821-8611		
Sudamericana	Buenos Aires	Av. Leandro N. Alem 720	+54 11 4319-3800		
NULL	NULL	NULL	NULL		

## LIBRO

```
-----  
-- Registro para la tabla Libro  
-----  
  
• INSERT INTO `LibreriaBuscaLibre`.`libro` (`ISBN`, `titulo`, `numero_paginas`, `nombre_editorial`)  
VALUES  
( '9788408150', 'La embarcacion solitaria', '432', 'Editorial Norma'),  
( '9786070756', 'El laberinto de los espíritus', '896', 'Planeta Colombia'),  
( '9786073135', 'La casa de los espíritus', '416', 'Editorial Océano de México'),  
( '9789500718', 'Rayuela', '666', 'Siglo XXI Editores');
```

Posteriormente se ingresaron nuevos registros para completar la tabla



ISBN	titulo	numero_paginas	nombre_editorial
9786070756	El laberinto de los espíritus	896	Planeta Colombia
9786073135	La casa de los espíritus	416	Editorial Océano de México
9786078352	El amor en los tiempos del cólera	96	Planeta Colombia
9786078453	Doce cuentos peregrinos	232	Editorial Océano de México
9787907645	Eva luna	217	Siglo XXI Editores
9788408150	La embarcacion solitaria	Eva luna	Editorial Norma
9788408251	Cronicas de una muerte anunciada	128	Editorial Norma
9788466328	Bestiario	146	Editorial Océano de Mexico
9789500718	Rayuela	666	Siglo XXI Editores

## LIBRO-AUTOR

```
-----  
-- Registro para la tabla Libro_autor  
-----  
  
• INSERT INTO `LibreriaBuscaLibre`.`libro_autor` (`ISBN_libro`, `id_autor`)  
VALUES  
( '9788408150', '1'), -- la embarcacion solitaria - Jorge Volpi  
( '9786070756', '1'), -- El laberinto de los espíritus - Jorge Volpi  
( '9786073135', '5'), -- La casa de los espíritus - Isabel Allende  
( '9789500718', '3'), -- Rayuela - Pedro Martinez  
( '9788408251', '6'), -- Crónica de una muerte anunciada - Gabriel García Márquez  
( '9786078352', '6'), -- El amor en los tiempos del cólera - Gabriel García Márquez  
( '9786078453', '6'), -- Doce cuentos peregrinos - Gabriel García Márquez  
( '9787907645', '5'), -- Eva Luna - Isabel Allende  
( '9778094564', '5'), -- Hija de la fortuna - Isabel Allende  
( '9788466328', '2'); -- Bestiario - Maria lopez
```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

ISBN_libro	id_autor
9786070756	1
9788408150	1
9788466328	2
9789500718	3
9778094564	5
9786073135	5
9787907645	5
9786078352	6
libro_autor 1	

## LIBRO\_CLIENTE

```
--
-- Registro para la tabla Libro_cliente
--
```

- INSERT INTO** `LibreriaBuscaLibre`.`libro\_cliente` (`ISBN\_libro\_cliente`, `id\_cliente`)
 **VALUES**
  
 ('9788408251', '1003046324'), -- Crónicas de una muerte anunciada - Sofia Gonzalez
 ('9786070756', '1003047685'), -- El laberinto de los espíritus - Ana Martinez
 ('9786073135', '1006986954'), -- La casa de los espíritus - Raul Medina
 ('9789500718', '1067987234'), -- Rayuela - Laura Diaz
 ('9788408251', '28905432'), -- Crónica de una muerte anunciada - Jorge Hernandez
 ('9786078352', '34987687'), -- El amor en los tiempos del cólera - Miguel Hernandez
 ('9786078453', '50897324'), -- Doce cuentos peregrinos - Carlos Sanchez
 ('9787907645', '34960070'), -- Eva Luna - Diego Castro
 ('9788466328', '309875670'), -- Bestiario - Fernando Torres
 ('9786078352', '10923485'); -- Crónica de una muerte anunciada - Carmen Garcia

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

cedula	nombre
28905432	Jorge Fernandez
309875670	Fernando Torres
34234109	Roberto Morales
34960070	Diego Castro
34987687	Miguel Hernandez
50345987	Elena Navarro
50891234	Mariana Perez
50897324	Carlos Sanchez
50960987	Adriana Jimenez
NULL	NULL

## TELEFONO\_CLIENTE

```

90  -- -----
91  -- Registro para la tabla Telefono_cliente
92  -- -----
93  • INSERT INTO `LibreriaBuscaLibre`.`telefono_cliente` (`cedula_cliente`, `numero`)
94  VALUES
95  ('10923485', '3145763458'), -- Carmen Garcia
96  ('34987687', '3119783457'), -- Miguel Hernandez
97  ('28905432', '3246328712'), -- Jorge Hernandez
98  ('1003046324', '3156943723'), -- Sofia Gonzalez
99  ('1006986954', '3023610274'), -- Raul Medina
100 ('50897324', '3023785469'), -- Carlos Sanchez
101 ('1067987234', '3233974634'), -- Laura Diaz
102 ('309875670', '3135018999'), -- Fernando Torres
103 ('34960070', '3135258285'), -- Diego Castro
104 ('1003046324', '3134567843'), -- Sofia Gonzalez
105 ('1003047685', '3162095489'), -- Ana Martinez
106 ('1003047685', '3024673219'); -- Ana Martinez

```

Result Grid			Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	cedula_cliente	numero				
▶	1003046324	3134567843				
	1003046324	3156943723				
	1003047685	3024673219				
	1003047685	3162095489				
	1006986954	3023610274				
	1067987234	3233974634				
	10923485	3145763458				
	28905432	3246328712				

telefono\_cliente 1 x

### 3. Creación de consultas SQL:

- Se realizaron consultas SQL para satisfacer diferentes requisitos, como obtener información de las tablas, realizar consultas de selección con filtros y realizar consultas de agregación para obtener datos resumidos. Estas consultas se utilizaron para explorar y comprender los datos en la base de datos.

1

```
5 -- Nombre y fecha de nacimiento de cada autor
6 • SELECT nombre, `fecha de nacimiento`
7 FROM autor;
8
```

nombre	fecha de nacimiento
Jorge Volpi	1968-07-10
María Lopez	1975-08-20
Pedro Martinez	1990-03-15
Laura Restrepo	1950-03-07
Isabel Allende	1942-08-02
Gabriel Garcia Marquez	1927-03-06

2

```
-- -----
-- cantidad de libros diferentes vendidos
-- -----
• SELECT COUNT(DISTINCT ISBN_libro_cliente) AS cantidad_libros_vendidos
FROM libro_cliente;
```

Para obtener la cantidad de libros diferentes vendidos, debemos contar la cantidad de ISBN únicos que aparecen en la tabla **libro\_cliente**, ya que cada ISBN representa un libro diferente vendido.

cantidad_libros_vendidos
8

Esta consulta cuenta la cantidad de ISBN únicos en la columna **ISBN\_libro\_cliente** de la tabla **libro\_cliente**, lo que nos dará la cantidad de libros diferentes vendidos, como se puede observar hasta ese momento la cantidad de libros diferentes vendidos es de 8.

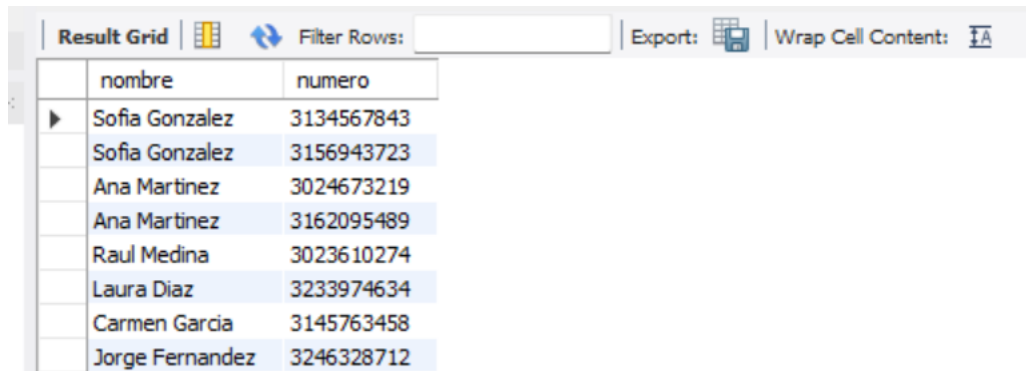


3

```
15  -- -----
16  -- Nombre de su cliente acompañado de su numero telefónico
17  -- -----
18
19  •  SELECT c.nombre, tc.numero
20     FROM cliente c
21     JOIN telefono_cliente tc ON c.cedula = tc.cedula_cliente;
```

Para obtener el nombre de los clientes acompañado de su número de teléfono, se debe combinar la información de las tablas **cliente** y **telefono\_cliente** utilizando la columna común que las relaciona, que es **cedula** en la tabla **cliente** y **cedula\_cliente** en la tabla **telefono\_cliente**

La cláusula **ON c.cedula = tc.cedula\_cliente** especifica que las filas de la tabla cliente se relacionan con las filas de la tabla telefono\_cliente cuando el valor de la columna **cedula** en la tabla **cliente** es igual al valor de la columna **cedula\_cliente** en la tabla **telefono\_cliente**



The screenshot shows a database interface with a 'Result Grid' tab. The grid displays the results of a SQL query. The columns are 'nombre' and 'numero'. The data rows are as follows:

nombre	numero
Sofia Gonzalez	3134567843
Sofia Gonzalez	3156943723
Ana Martinez	3024673219
Ana Martinez	3162095489
Raul Medina	3023610274
Laura Diaz	3233974634
Carmen Garcia	3145763458
Jorge Fernandez	3246328712

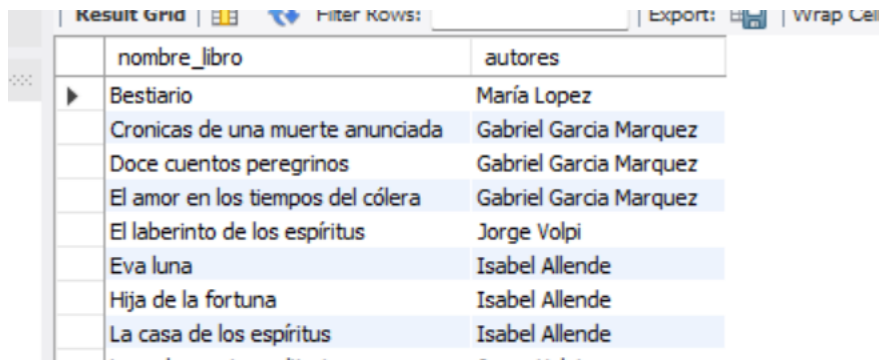
Esta consulta utiliza la cláusula **JOIN** para combinar las filas de las tablas **cliente** y **telefono\_cliente** donde los valores de **cedula** en la tabla **cliente** coinciden con los valores de **cedula\_cliente** en la tabla **telefono\_cliente**. Luego selecciona el nombre del cliente de la tabla **cliente** y el número de teléfono de la tabla **telefono\_cliente**

#### 4

Para obtener el nombre del libro acompañado por su autor o autores, debemos combinar la información de las tablas **libro** y **autor**, utilizando la tabla de relación **libro\_autor** para relacionar los libros con sus autores

```
23  -- -----
24  -- Nombre del libro acompañado por su autor o autores
25  -- -----
26  • SELECT l.titulo AS nombre_libro, GROUP_CONCAT(a.nombre SEPARATOR ', ') AS autores
27  FROM libro l
28  JOIN libro_autor la ON l.ISBN = la.ISBN_libro
29  JOIN autor a ON la.id_autor = a.id
30  GROUP BY l.titulo;
```

Esta consulta utiliza las cláusulas **JOIN** para combinar las filas de las tablas **libro**, **libro\_autor** y **autor**, relacionando los libros con sus autores, luego, utiliza **GROUP\_CONCAT** para concatenar los nombres de los autores en una sola cadena, separados por comas, y finalmente, **GROUP BY** para agrupar los resultados por título del libro.



nombre_libro	autores
Bestiario	María Lopez
Cronicas de una muerte anunciada	Gabriel Garcia Marquez
Doce cuentos peregrinos	Gabriel Garcia Marquez
El amor en los tiempos del cólera	Gabriel Garcia Marquez
El laberinto de los espíritus	Jorge Volpi
Eva luna	Isabel Allende
Hija de la fortuna	Isabel Allende
La casa de los espíritus	Isabel Allende

## 5

Para obtener el nombre de las editoriales que han logrado vender más libros, se debe consultar las editoriales que tienen libros asociados en la tabla **libro**

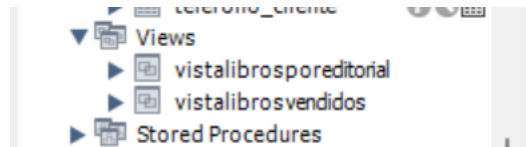
```
32  -- -----  
33  -- Nombre de la editoriales que han logrado vender mas libros  
34  -- -----  
35  • SELECT DISTINCT e.nombre  
36  FROM Editorial e  
37  JOIN libro l ON e.nombre = l.nombre_editorial;
```

En esta consulta se utiliza la cláusula **JOIN** para combinar las filas de las tablas **Editorial** y **libro**, basándose en el nombre de la editorial que se encuentra en la columna **nombre\_editorial** de la tabla **libro**, después utilizando **DISTINCT**, nos aseguramos de que cada nombre de editorial aparezca solo **una vez** en los resultados, evitando duplicados.

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
nombre			
▶ Editorial Norma			
Editorial Océano de México			
Planeta Colombia			
Siglo XXI Editores			

#### 4. CREACIÓN DE VISTAS:

- Se crearon dos vistas en la base de datos utilizando la cláusula **CREATE VIEW** en SQL. Las vistas creadas fueron **VistaLibrosVendidos** y **VistaLibrosPorEditorial**. Estas vistas proporcionan diferente información de los datos almacenados en la base de datos y simplifican consultas comunes al organizar la información.



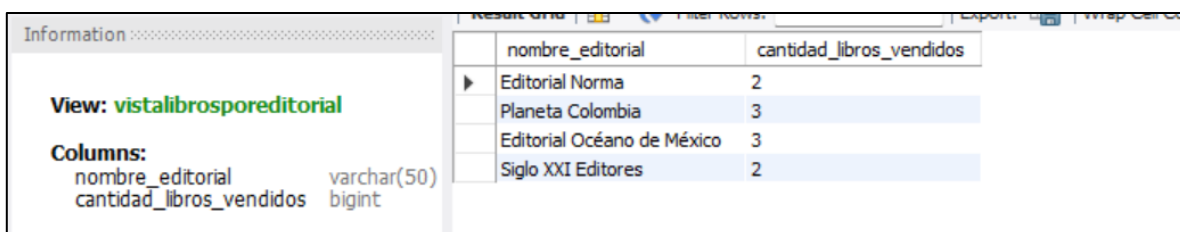
##### VISTA 1

```
6  -- VISTA LIBROS VENDIDOS CON DETALLE DE CLIENTES
7  -----
8  -- Esta vista proporciona una lista de todos los libros vendidos junto con detalles de los clientes que los compraron
9  -- incluyendo su nombre y número de teléfono. Es importante esta vista ya que es útil para consultas frecuentes que
10 -- involucran detalles de ventas y clientes debido a que simplifica la consulta al centralizar
11 -- la información relevante en una sola vista.
12 -----
13 • CREATE VIEW VistaLibrosVendidos AS
14 SELECT lc.ISBN_libro_cliente, l.titulo AS titulo_libro, c.cedula, c.nombre AS nombre_cliente, tc.numero AS telefono_cliente
15 FROM libro_cliente lc
16 JOIN libro l ON lc.ISBN_libro_cliente = l.ISBN
17 JOIN cliente c ON lc.id_cliente = c.cedula
18 JOIN telefono_cliente tc ON c.cedula = tc.cedula_cliente;
```

View: vistalibrosponsoreditorial					
Columns:					
nombre_editorial	varchar(50)				
cantidad_libros_vendidos	bigint				
ISBN_libro_cliente	titulo_libro	cedula	nombre_cliente	telefono_cliente	
9788408251	Cronicas de una muerte anunciada	1003046324	Sofia Gonzalez	3134567843	
9788408251	Cronicas de una muerte anunciada	1003046324	Sofia Gonzalez	3156943723	
9786070756	El laberinto de los espíritus	1003047685	Ana Martinez	3024673219	
9786070756	El laberinto de los espíritus	1003047685	Ana Martinez	3162095489	
9786073135	La casa de los espíritus	1006986954	Raul Medina	3023610274	
9789500718	Rayuela	1067987234	Laura Diaz	3233974634	
9786078352	El amor en los tiempos del cólera	10923485	Carmen Garcia	3145763458	
9788408251	Cronicas de una muerte anunciada	1003046324	Sofia Gonzalez	3134567843	

## VISTA 2

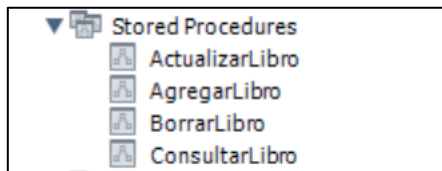
```
20  -- -----
21  -- VISTA CANTIDAD DE LIBROS VENDIDOS POR EDITORIAL
22  -- -----
23  -- Esta vista proporciona una perspectiva de la cantidad de libros vendidos por cada editorial. Es útil para el análisis
24  -- de ventas y la toma de decisiones empresariales, ya que permite identificar rápidamente las editoriales más exitosas en
25  -- términos de ventas.
26  • CREATE VIEW VistaLibrosPorEditorial AS
27  SELECT l.nombre_editorial, COUNT(l.ISBN) AS cantidad_libros_vendidos
28  FROM libro l
29  JOIN libro_cliente lc ON l.ISBN = lc.ISBN_libro_cliente
30  GROUP BY l.nombre_editorial;
```



nombre_editorial	cantidad_libros_vendidos
Editorial Norma	2
Planeta Colombia	3
Editorial Océano de México	3
Siglo XXI Editores	2

## PROCEDIMIENTOS ALMACENADOS:

- Se crearon cuatro procedimientos almacenados: **AgregarLibro**, **ActualizarLibro**, **ConsultarLibro** y **BorrarLibro**.



- Estos procedimientos permiten realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en la tabla **libro** de la base de datos LibreriaBuscaLibre.

## AGREGAR UN LIBRO

```
SQL File 8* x SQL File 9* libro
-- AGREGAR LIBRO
-----
DELIMITER //
CREATE PROCEDURE AgregarLibro(
    IN p_ISBN VARCHAR(10),
    IN p_titulo VARCHAR(45),
    IN p_numero_paginas VARCHAR(45),
    IN p_nombre_editorial VARCHAR(50)
)
BEGIN
    INSERT INTO libro (ISBN, titulo, numero_paginas, nombre_editorial)
    VALUES (p_ISBN, p_titulo, p_numero_paginas, p_nombre_editorial);
END //
DELIMITER ;
```

## CONSULTAR UN LIBRO

```
-----
-- CONSULTAR INFORMACION DE UN LIBRO
-----
DELIMITER //
CREATE PROCEDURE ConsultarLibro(
    IN p_ISBN VARCHAR(10)
)
BEGIN
    SELECT * FROM libro WHERE ISBN = p_ISBN;
END //
DELIMITER ;
```

## ACTUALIZAR LIBRO

```
SQL File 8 x SQL File 9 libro libro
Limit to 1000 rows
22 DELIMITER //
23 • CREATE PROCEDURE ActualizarLibro(
24     IN p_ISBN VARCHAR(10),
25     IN p_titulo VARCHAR(45),
26     IN p_numero_paginas VARCHAR(45),
27     IN p_nombre_editorial VARCHAR(50)
28 )
29 BEGIN
30     UPDATE libro
31     SET titulo = p_titulo,
32         numero_paginas = p_numero_paginas,
33         nombre_editorial = p_nombre_editorial
34     WHERE ISBN = p_ISBN;
35 END //
36 DELIMITER ;
```

## ELIMINAR UN LIBRO

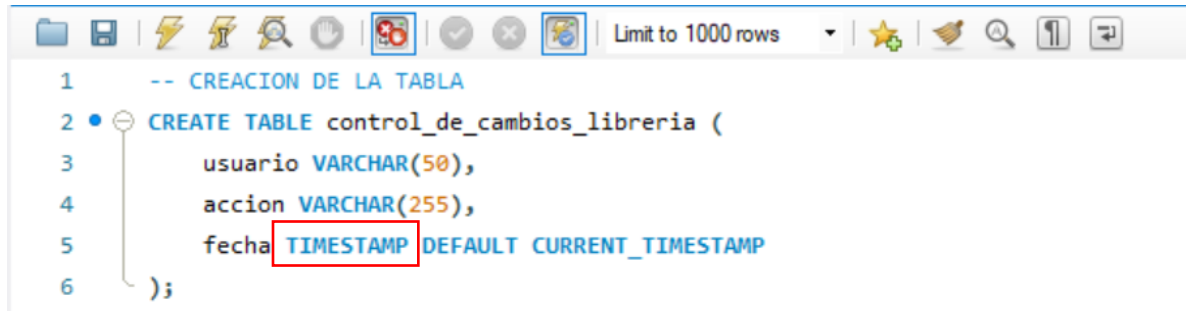
```
51 -- -----
52 -- ELIMINAR LIBRO
53 -- -----
54 DELIMITER //
55 • CREATE PROCEDURE BorrarLibro(
56     IN p_ISBN VARCHAR(10)
57 )
58 BEGIN
59     DELETE FROM libro WHERE ISBN = p_ISBN;
60 END //
61 DELIMITER ;
```

- Cada procedimiento toma parámetros de entrada relevantes para la operación que realiza y ejecuta consultas SQL correspondientes en la tabla libro.

## TRIGGERS

### TABLA DE CONTROL DE CAMBIOS:

- Se creó una nueva tabla llamada **control\_de\_cambios\_libreria** con las columnas **usuario**, **accion** y **fecha**.
- Esta tabla se utiliza para registrar cambios en la tabla libro, incluyendo quién realizó la acción (usuario), qué acción se realizó (accion) y cuándo se realizó (fecha).

A screenshot of a SQL IDE interface. The top toolbar contains various icons for file operations, execution, and search. Below the toolbar, a SQL query is displayed in a text editor. The query is for creating a table named 'control\_de\_cambios\_libreria' with three columns: 'usuario' of type VARCHAR(50), 'accion' of type VARCHAR(255), and 'fecha' of type TIMESTAMP with a default value of CURRENT\_TIMESTAMP. The 'TIMESTAMP' keyword is highlighted with a red rectangle. The query is numbered 1 through 6 on the left margin.

```
1  -- CREACION DE LA TABLA
2  CREATE TABLE control_de_cambios_libreria (
3      usuario VARCHAR(50),
4      accion VARCHAR(255),
5      fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP
6  );
```

- El tipo **TIMESTAMP** permite almacenar fechas y horas con una precisión de hasta segundos, lo que es adecuado para este caso.
- Se implementaron dos triggers: **registro\_insercion** y **registro\_eliminacion**.  
(se pueden observar en el documento “TRIGGER-insercion-eliminacion-LIBRERÍA.sql”)



- El trigger **registro\_insercion** se activa después de una inserción en la tabla **libro** y registra el nombre de usuario, la acción "Inserción" y la fecha en la tabla **control\_de\_cambios\_libreria**.

```

7 DELIMITER //
8 • CREATE TRIGGER registro_insercion
9 AFTER INSERT ON cliente
10 FOR EACH ROW
11 BEGIN
12     INSERT INTO control_de_cambios_hospital (usuario, accion)
13     VALUES (NEW.nombre, 'Inserción');
14 END;
15 //
16 DELIMITER ;

```

Output

Action Output

#	Time	Action	Message
✓ 20	09:33:16	DROP TABLE `libreriaescalibre`.`control_de_cambios_libreria`	0 row(s) affected
✓ 21	09:33:34	CREATE TABLE control_de_cambios_hospital ( usuario VARCHAR(50), ...	0 row(s) affected
✓ 22	09:34:01	DROP TABLE `libreriaescalibre`.`control_de_cambios_hospital`	0 row(s) affected
✓ 23	09:34:53	CREATE TABLE control_de_cambios_libreria ( usuario VARCHAR(50), a...	0 row(s) affected
✓ 24	09:35:49	SELECT * FROM libreriaescalibre.control_de_cambios_libreria LIMIT 0, 1000	0 row(s) returned
✓ 25	09:46:01	CREATE TRIGGER registro_insercion AFTER INSERT ON cliente FOR EAC...	0 row(s) affected

- El trigger **registro\_eliminacion** se activa después de una eliminación en la tabla **libro** y registra el nombre de usuario y la acción "Eliminación" en la tabla **control\_de\_cambios\_libreria**.

```

23 DELIMITER //
24 • CREATE TRIGGER registro_eliminacion
25 AFTER DELETE ON cliente
26 FOR EACH ROW
27 BEGIN
28     INSERT INTO control_de_cambios_libreria (usuario, accion)
29     VALUES (OLD.nombre, 'Eliminación');
30 END;
31 //
32 DELIMITER ;
33

```

Output			
Action Output			
#	Time	Action	Message
✓ 32	09:52:29	INSERT INTO cliente(cedula,nombre) VALUES('1003049508','Devon Alvarez')	1 row(s) affected
✓ 33	09:52:39	SELECT * FROM libreriaescalibre.cliente LIMIT 0, 1000	21 row(s) returned
✓ 34	09:52:46	SELECT * FROM libreriaescalibre.control_de_cambios_libreria LIMIT 0, 1000	1 row(s) returned
✗ 35	09:55:31	CREATE TRIGGER registro_eliminacion AFTER DELETE ON cliente FOR EA...	Error Code: 1363. There is no NEW row in on DELETE trigger
✗ 36	09:58:03	CREATE TRIGGER registro_insercion AFTER INSERT ON cliente FOR EAC...	Error Code: 1363. There is no OLD row in on INSERT trigger
✓ 37	09:58:33	CREATE TRIGGER registro_eliminacion AFTER DELETE ON cliente FOR EA...	0 row(s) affected

- Estos triggers proporcionan un mecanismo automatizado para registrar cambios en la tabla **libro** y mantener un registro histórico de las acciones realizadas por los usuarios.
- Se puede observar que se ingreso un registro en la tabla cliente y una eliminación en esa misma tabla y observamos que esas acciones quedan registradas en la tabla **control\_de\_cambios\_libreria**.

```

1 • SELECT * FROM libreriaescalibre.control_de_cambios_libreria;
2
3 -- INSERCIÓN
4 • INSERT INTO cliente(cedula,nombre) VALUES('1003049508','Devon Alvarez');
5 -- ELIMINACIÓN
6 • DELETE FROM cliente WHERE cedula = '50345987';

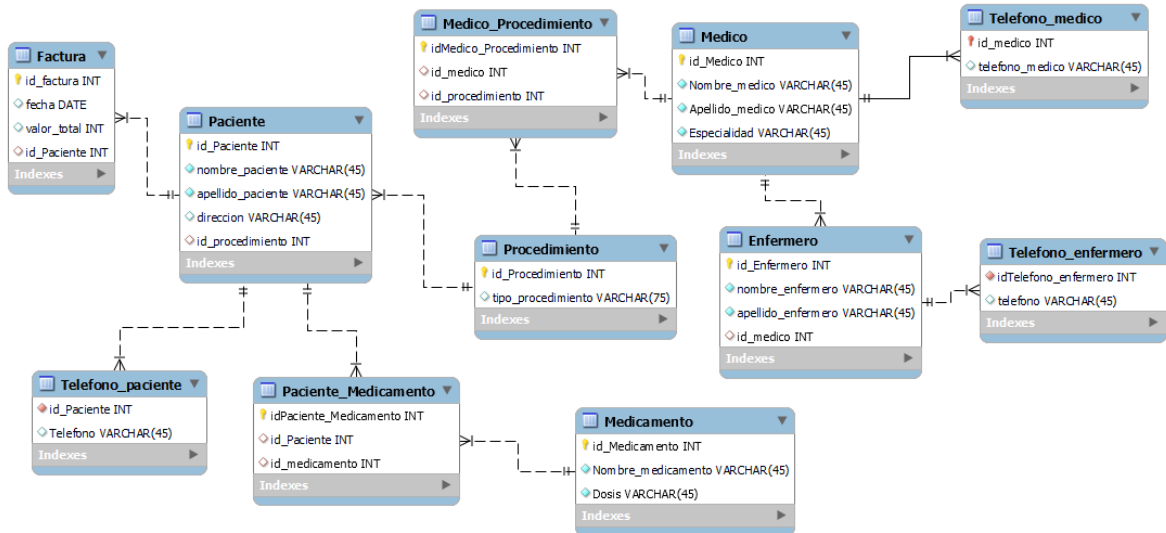
```

Result Grid			
Filter Rows:			
Export:   Wrap Cell Content:			
usuario	accion	fecha	
Devon Alvarez	Inserción	2024-03-30 09:52:29	
Elena Navarro	Eliminación	2024-03-30 10:03:30	

## CREACION DE BD Hospital\_bd

### Diseño del Diagrama de la Base de Datos:

- Se comienza diseñando un diagrama de la base de datos "Hospital" utilizando el MER Y MR proporcionado en GitHub para realizar la actividad.



- Se determinan los atributos de cada entidad y las claves primarias y foráneas necesarias para establecer las relaciones entre las tablas.

### Traducción del Diagrama a SQL:

- Se traduce el diseño del diagrama de la base de datos a sentencias SQL para crear las tablas y definir las relaciones entre ellas.
- Se definen las tablas necesarias, sus columnas y restricciones, como las claves primarias y foráneas.

```
-- Schema Hospital_BD
-----

CREATE SCHEMA IF NOT EXISTS `Hospital_BD` DEFAULT CHARACTER SET utf8 ;
USE `Hospital_BD` ;

-----

-- Table `Hospital_BD`.`Procedimiento`
-----

CREATE TABLE IF NOT EXISTS `Hospital_BD`.`Procedimiento` (
  `id_Procedimiento` INT NOT NULL AUTO_INCREMENT,
  `tipo_procedimiento` VARCHAR(75) NULL,
  PRIMARY KEY (`id_Procedimiento`))
ENGINE = InnoDB;

-----

-- Table `Hospital_BD`.`Paciente`
-----

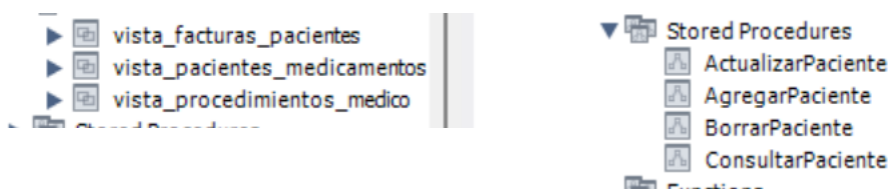
CREATE TABLE IF NOT EXISTS `Hospital_BD`.`Paciente` (
  `id Paciente` INT NOT NULL AUTO INCREMENT,
```

(Mirar script completo en el archivo: "Script\_BD\_Hospital.sql")

## CREACIÓN DE LAS TABLAS:

Se utiliza la sentencia SQL (**Mirar script completo en el archivo: "Script\_BD\_Hospital.sql"**) para crear cada una de las tablas necesarias en la base de datos

Se realiza la creación de las vistas en la base de datos al igual que los procedimientos almacenados



En los documentos .sql adjuntos en el repositorio en GITHUB se observan los detalles

## Creación de Triggers:

- Se identifican los eventos que activarán los triggers, como la inserción, actualización o eliminación de registros en ciertas tablas.
- Se utilizan las sentencias SQL **CREATE TRIGGER** para definir los triggers y sus acciones asociadas.