

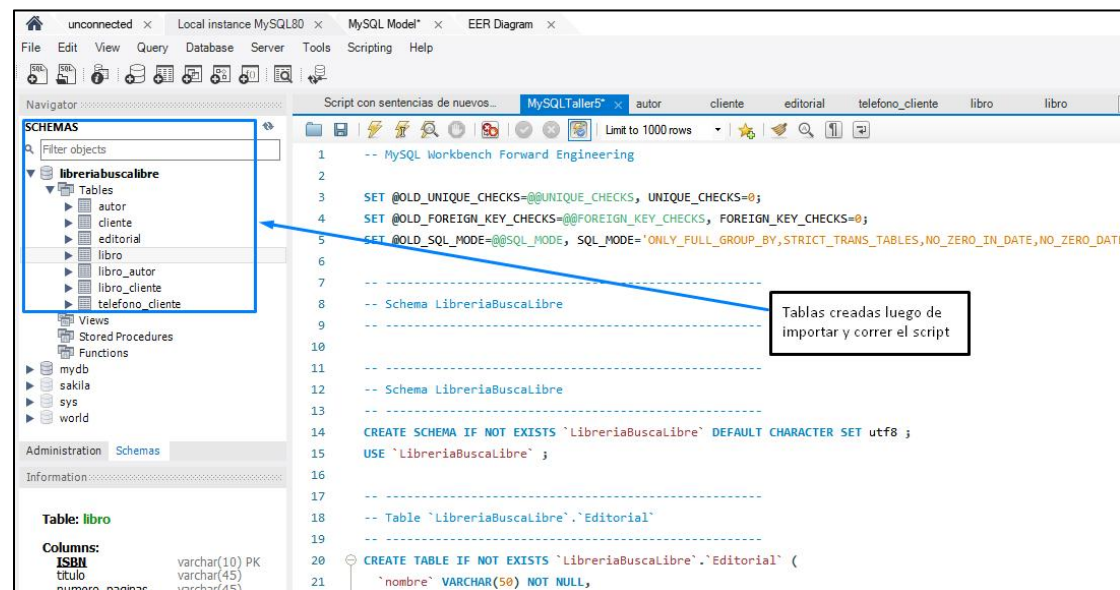
C1-2023-QA-BD-05

Consultas y Vistas

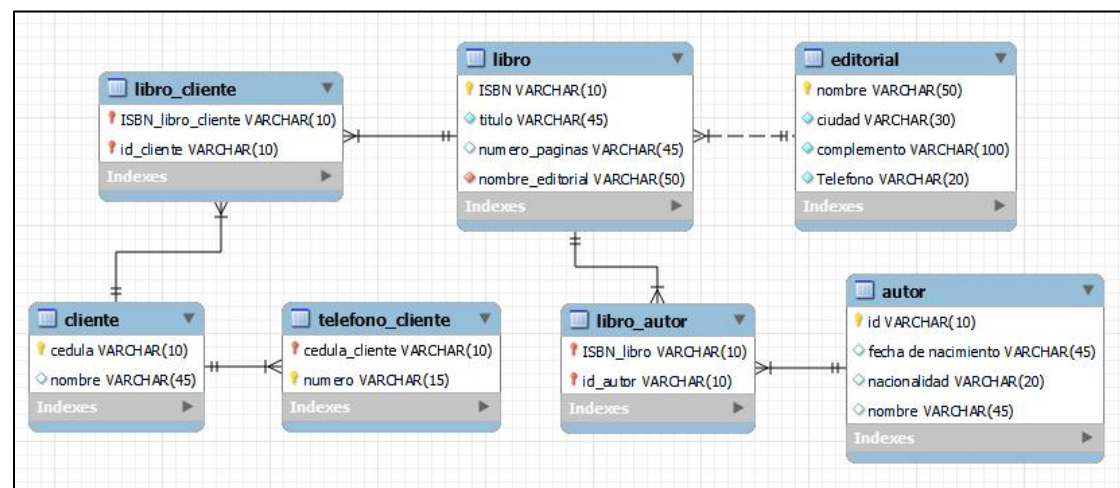
Primera actividad: Utilizando el ejercicio de la Librería realizado en clase (se adjunta script SQL) realice lo siguiente:

- Complete la información para las tablas autor, libro, cliente, editorial, libro_cliente, libro_autor y teléfono_cliente con al menos (5,20,7,4,10,10, 12) registros respectivamente usando unicamente comandos SQL creados por usted.

Se importa el script en MySQL y se corre para crear las tablas como se observa en la siguiente imagen:



Se realiza ingeniería inversa para observar el modelo relacional:



A continuación se ingresan los valores de la tabla “autor”:

The screenshot shows the MySQLTaller5 interface. On the left, the 'libreriaBuscalibre' schema is selected, and the 'autor' table is highlighted. The main window displays an SQL script with an INSERT statement for the 'autor' table. The script is as follows:

```
1 SELECT * FROM libreriaBuscalibre.autor;
2 INSERT INTO `libreriaBuscalibre`.`autor` (`id`, `fecha de nacimiento`, `nacionalidad`, `nombre`)
3 VALUES
4 ('gabrielgm', '06/03/1927', 'colombiano', 'Gabriel García Márquez'),
5 ('hectoraf', '07/09/1958', 'colombiano', 'Héctor Abad Faciolince'),
6 ('joserivera', '19/02/1888', 'colombiano', 'José Eustasio Rivera'),
7 ('laurence', '25/02/1964', 'colombiano', 'Laurence González'),
8 ('juanvas', '24/10/1978', 'colombiano', 'Juan Vasquez');
```

Below the script, the 'Result Grid' shows the data inserted into the 'autor' table:

id	fecha de nacimiento	nacionalidad	nombre
gabrielgm	06/03/1927	colombiano	Gabriel García Márquez
hectoraf	07/09/1958	colombiano	Héctor Abad Faciolince
joserivera	19/02/1888	colombiano	José Eustasio Rivera
juanvas	24/10/1978	colombiano	Juan Vasquez
laurence	25/02/1964	colombiano	Laurence González

On the left, the table structure for 'libro_autor' is shown:

Table: libro_autor

Columns:

- ISBN_libro varchar(10) PK
- id_autor varchar(10) PK

An arrow points from the text 'Se adicionan las sentencias en SQL para ingresar los autores en la tabla.' to the INSERT statement in the script.

Se ingresan los valores de la tabla “cliente”:

The screenshot shows the MySQLTaller5 interface. On the left, the 'libreriaBuscalibre' schema is selected, and the 'cliente' table is highlighted. The main window displays an SQL script with an INSERT statement for the 'cliente' table. The script is as follows:

```
2 INSERT INTO `libreriaBuscalibre`.`cliente` (`cedula`, `nombre`)
3 VALUES
4 ('3456789012', 'Carlos Calero'),
5 ('5678901234', 'Jota Mario Valencia'),
6 ('2345678901', 'Pilar Schmitt'),
7 ('6789012345', 'Alejandro Riaño'),
8 ('1234567890', 'Mabel Cartagena'),
9 ('4567890123', 'Andrea Serna'),
10 ('7890123456', 'Tatán Mejía');
```

Below the script, the 'Result Grid' shows the data inserted into the 'cliente' table:

cedula	nombre
1234567890	Mabel Cartagena
2345678901	Pilar Schmitt
3456789012	Carlos Calero
4567890123	Andrea Serna
5678901234	Jota Mario Valencia
6789012345	Alejandro Riaño
7890123456	Tatán Mejía

On the left, the table structure for 'cliente' is shown:

Table: cliente

Columns:

- cedula varchar(10) PK
- nombre varchar(45)

An arrow points from the text 'Se adicionan la sentencias en SQL para agregar los clientes a la tabla.' to the INSERT statement in the script.

Se ingresan los valores de la tabla “editorial”:

The screenshot shows the MySQLTaller5 interface. On the left, the 'libreriaBuscalibre' schema is selected, and the 'editorial' table is highlighted. The main window displays an SQL script with an INSERT statement for the 'editorial' table. The script is as follows:

```
1 SELECT * FROM libreriaBuscalibre.editorial;
2 INSERT INTO `libreriaBuscalibre`.`editorial` (`nombre`, `ciudad`, `complemento`, `Telefono`)
3 VALUES
4 ('Norma', 'Medellín', 'Editorial de textos para estudiantes', '6012508690'),
5 ('El Planeta', 'Medellín', 'Editorial de textos universitarios', '6012558690'),
6 ('Santillana', 'Bogotá', 'Editorial de textos de ficción', '6012598690'),
7 ('Aguilar', 'Villavicencio', 'Editorial de textos del área de medicina', '6012508869');
```

Below the script, the 'Result Grid' shows the data inserted into the 'editorial' table:

nombre	ciudad	complemento	Telefono
Aguilar	Villavicencio	Editorial de textos del área de medicina	6012508869
El Planeta	Medellín	Editorial de textos universitarios	6012558690
Norma	Medellín	Editorial de textos para estudiantes	6012508690
Santillana	Bogotá	Editorial de textos de ficción	6012598690

On the left, the table structure for 'cliente' is shown (this is a typo in the original image, it should be 'editorial'):

Table: cliente

Columns:

- cedula varchar(10) PK
- nombre varchar(45)

An arrow points from the text 'Se adicionan las sentencias en SQL para agregar las editoriales a la tabla.' to the INSERT statement in the script.

Se ingresan los valores de la tabla "libro", las siguientes son las sentencias para el ingreso de los 20 libros:

```
INSERT INTO `LibreriaBuscaLibre`.`libro` (`ISBN`, `titulo`, `numero_paginas`, `nombre_editorial`)
VALUES
('42-7452-8','Cien años de soledad','522','Norma'),
('42-7453-5','El amor en los tiempos del cólera','302','El Planeta'),
('42-7454-2','Crónica de una muerte anunciada','150','Norma'),
('42-7455-9','El olvido que seremos','204','Santillana'),
('42-7456-6','Del amor y otros demonios','400','Aguilar'),
('42-7467-2','La vorágine','155','Norma'),
('42-7468-9','Ojos de perro azul','99','Aguilar'),
('42-7469-6','La increíble y triste historia de la cándida','80','El Planeta'),
('42-7470-2','El coronel no tiene quien le escriba','204','Norma'),
('42-1598-8','La hojarasca','150','Santillana'),
('42-0682-9','Memoria de mis putas tristes','112','El Planeta'),
('42-1564-3','El amor en los tiempos del cólera','368','Norma'),
('42-1889-7','Del amor y otros demonios','160','Aguilar'),
('42-0299-9','Crónica de una muerte anunciada','120','El Planeta'),
('42-2814-9','El ruido de las cosas al caer','288','Santillana'),
('42-2218-9','Noticia de un secuestro','352','Aguilar'),
('42-1071-1','El otoño del patriarca','416','Norma'),
('42-3073-3','El general en su laberinto','320','El Planeta'),
('42-2993-5','Vivir para contarla','624','Santillana');
```

Script con sentencias de nuevos... MySQLTuner5 libro_autor autor cliente editorial libro

```
1 • SELECT * FROM libreriaescalibre.libro;
2 • INSERT INTO `LibreriaBuscaLibre`.`libro` (`ISBN`, `titulo`, `numero_paginas`, `nombre_editorial`)
3   VALUES
4   ('42-7452-8','Cien años de soledad','522','Norma'),
5   ('42-7453-5','El amor en los tiempos del cólera','302','El Planeta'),
```

ISBN	titulo	numero_paginas	nombre_editorial
42-0299-9	Crónica de una muerte anunciada	120	El Planeta
42-0682-9	Memoria de mis putas tristes	112	El Planeta
42-1071-1	El otoño del patriarca	416	Norma
42-1564-3	El amor en los tiempos del cólera	368	Norma
42-1598-8	La hojarasca	150	Santillana
42-1889-7	Del amor y otros demonios	160	Aguilar
42-2218-9	Noticia de un secuestro	352	Aguilar
42-2814-9	El ruido de las cosas al caer	288	Santillana
42-2993-5	Vivir para contarla	624	Santillana
42-3073-3	El general en su laberinto	320	El Planeta
42-7452-8	Cien años de soledad	522	Norma
42-7453-5	El amor en los tiempos del cólera	302	El Planeta
42-7454-2	Crónica de una muerte anunciada	150	Norma
42-7455-9	El olvido que seremos	204	Santillana
42-7456-6	Del amor y otros demonios	400	Aguilar
42-7467-2	La vorágine	155	Norma
42-7468-9	Ojos de perro azul	99	Aguilar
42-7469-6	La increíble y triste historia de la cándida	80	El Planeta

Se observan los registros ingresados.

Table: cliente
Columns: cedula varchar(10) PK, nombre varchar(45)

Se ingresan los valores de la tabla "libro_autor":

Script con sentencias de nuevos... MySQLTuner5 libro_autor autor cliente editorial libro libro_autor

```
1 • SELECT * FROM libreriaescalibre.libro_autor;
2 • INSERT INTO `LibreriaBuscaLibre`.`libro_autor` (`ISBN_libro`, `id_autor`)
3   VALUES
4   ('42-2814-9','juanvas'),
5   ('42-1889-7','gabrielgm'),
6   ('42-1071-1','gabrielgm'),
7   ('42-7452-8','gabrielgm'),
8   ('42-7467-2','gabrielgm'),
9   ('42-3073-3','gabrielgm'),
10  ('42-7469-6','gabrielgm');
```

ISBN_libro	id_autor
42-1071-1	gabrielgm
42-1564-3	gabrielgm
42-1889-7	gabrielgm
42-2218-9	gabrielgm
42-2993-5	gabrielgm
42-3073-3	gabrielgm

Table: libro_autor

Se ingresan los valores de la tabla “libro_cliente”:

The screenshot shows the MySQLTaller5 interface. On the left, the 'Schemas' panel displays the 'libreriaBuscalibre' database with tables: autor, cliente, editorial, libro, libro_autor, libro_cliente, and telefono_cliente. The 'libro_cliente' table is selected, showing its columns: ISBN_libro (varchar(10) PK) and id_cliente (varchar(10) PK). The main window shows a script with 10 SQL statements to insert data into 'libro_cliente'. The 'Result Grid' displays the results of the insertions.

ISBN_libro_cliente	id_cliente
42-3073-3	1234567890
42-7452-8	2345678901
42-2814-9	3456789012
42-7452-8	3456789012
42-2814-9	4567890123
42-1071-1	5678901234
42-1889-7	5678901234
42-7467-2	6789012345
42-3073-3	7890123456
42-7467-2	7890123456

Se ingresan los valores de la tabla “telefono_cliente”:

The screenshot shows the MySQLTaller5 interface. On the left, the 'Schemas' panel displays the 'libreriaBuscalibre' database with tables: autor, cliente, editorial, libro, libro_autor, libro_cliente, and telefono_cliente. The 'telefono_cliente' table is selected, showing its columns: cedula_cliente (varchar(10) PK) and numero (varchar(10) PK). The main window shows a script with 8 SQL statements to insert data into 'telefono_cliente'. The 'Result Grid' displays the results of the insertions.

cedula_cliente	numero
1234567890	3116884652
2345678901	3155884652
2345678901	3215884652
3456789012	3115884652
3456789012	3125884652
4567890123	3118887652
5678901234	3135884652
5678901234	3145884652
6789012345	3217884652
6789012345	3225884652
7890123456	3113884652
7890123456	3114884652

- realice 5 consultas que me permitan conocer el nombre y la fecha de nacimiento de cada escritor, la cantidad de libros diferentes vendidos, el nombre de su cliente acompañado de su numero telefónico, el nombre del libro acompañado por su autor o sus autores, el nombre de las editoriales que han logrado vender libros.

Nombre y la fecha de nacimiento de cada escritor

The screenshot shows the MySQLTaller5 interface. On the left, the 'Schemas' panel displays the 'libreriaBuscalibre' database with tables: autor, cliente, editorial, libro, libro_autor, libro_cliente, and telefono_cliente. The 'autor' table is selected, showing its columns: nombre (varchar(100) PK) and fecha de nacimiento (date PK). The main window shows a script with 2 SQL statements. The first statement is 'SELECT * FROM libreriaBuscalibre.autor;'. The second statement is 'SELECT nombre, `fecha de nacimiento` FROM autor;'. The 'Result Grid' displays the results of the second query.

Consulta realizada

nombre	fecha de nacimiento
Gabriel García Márquez	06/03/1927
Héctor Abad Faciolince	07/09/1958
José Eustasio Rivera	19/02/1888
Juan Vasquez	24/10/1978
Laurence González	25/02/1964

Resultado de la consulta realizada

la cantidad de libros diferentes vendidos se asume con la cantidad de registros en la tabla libro_cliente:

The screenshot shows the MySQLTaller5* interface. On the left, the 'SCHEMAS' panel displays the 'libreriaescalibre' database with tables: autor, cliente, editorial, libro, libro_autor, libro_cliente, and telefono_cliente. The main editor shows a SQL script with the following queries:

```
1 • SELECT * FROM libreriaescalibre.autor;  
2 • SELECT COUNT(*) AS cantidad_libros_vendidos  
3 • FROM libro_cliente;  
4 • SELECT nombre, `fecha de nacimiento` FROM autor;  
5 • SELECT l.titulo  
6 • FROM libro l  
7 • JOIN libro_cliente lc ON l.ISBN = lc.ISBN_libro_cliente;
```

Annotations with arrows point to the query and the result grid:

- Arrow to line 2: "Con estas sentencias se cuenta el número de libros vendidos"
- Arrow to the result grid: "Se genera este resultado"

The 'Result Grid' shows a single row with the column 'cantidad_libros_vendidos' and the value '10'.

Table: autor
Columns:

Y los siguientes son los diferentes libros vendidos:

The screenshot shows the MySQLTaller5* interface. The main editor shows a SQL script with the following queries:

```
1 • SELECT * FROM libreriaescalibre.autor;  
2 • SELECT COUNT(*) AS cantidad_libros_vendidos  
3 • FROM libro_cliente;  
4 • SELECT nombre, `fecha de nacimiento` FROM autor;  
5 • SELECT l.titulo  
6 • FROM libro l  
7 • JOIN libro_cliente lc ON l.ISBN = lc.ISBN_libro_cliente;
```

Annotations with arrows point to the query and the result grid:

- Arrow to line 5: "Con estas sentencias se toma el nombre de cada libro vendido"
- Arrow to the result grid: "Estos son los datos de los libros vendidos"

The 'Result Grid' shows a list of book titles:

titulo
El general en su laberinto
Cien años de soledad
El ruido de las cosas al caer
Cien años de soledad
El ruido de las cosas al caer
El otoño del patriarca
Del amor y otros demonios
La vorágine
El general en su laberinto
La vorágine

Table: autor
Columns: id, fecha de nacimiento

El nombre de su cliente acompañado de su numero telefónico:

The screenshot shows the MySQLTaller5* interface. The main editor shows a SQL script with the following queries:

```
1 • SELECT * FROM libreriaescalibre.cliente;  
2 • SELECT c.nombre, tc.numero  
3 • FROM cliente c  
4 • JOIN telefono_cliente tc  
5 • ON c.cedula = tc.cedula_cliente;
```

Annotations with arrows point to the query and the result grid:

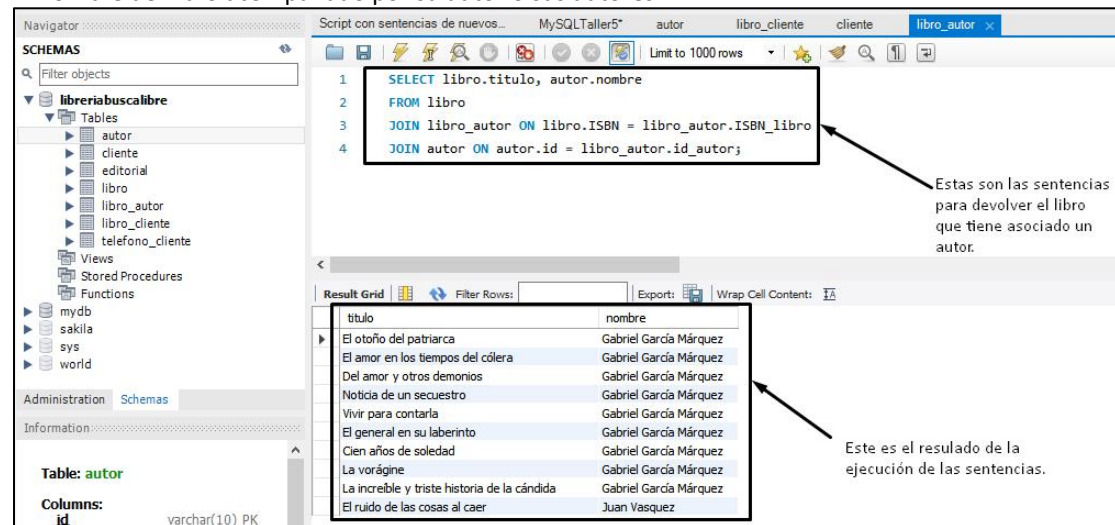
- Arrow to line 2: "Estas sentencias en SQL hacen la unión de las tablas para devolver el nombre y telefono."
- Arrow to the result grid: "Este es el resultado de ejecutar las sentencias."

The 'Result Grid' shows a list of client names and phone numbers:

nombre	numero
Mabel Cartagena	3116884652
Pilar Schmitt	3115884652
Pilar Schmitt	3215884652
Carlos Calero	3115884652
Carlos Calero	3125884652
Andrea Serna	3118887652
Jota Mario Valencia	3135884652
Jota Mario Valencia	3145884652
Alejandro Riaño	3217884652
Alejandro Riaño	3225884652
Tatán Mejía	3113884652
Tatán Mejía	3114884652

Table: autor
Columns: id, fecha de nacimiento

El nombre del libro acompañado por su autor o sus autores



Script con sentencias de nuevos...

```
1 SELECT libro.titulo, autor.nombre
2 FROM libro
3 JOIN libro_autor ON libro.ISBN = libro_autor.ISBN_libro
4 JOIN autor ON autor.id = libro_autor.id_autor;
```

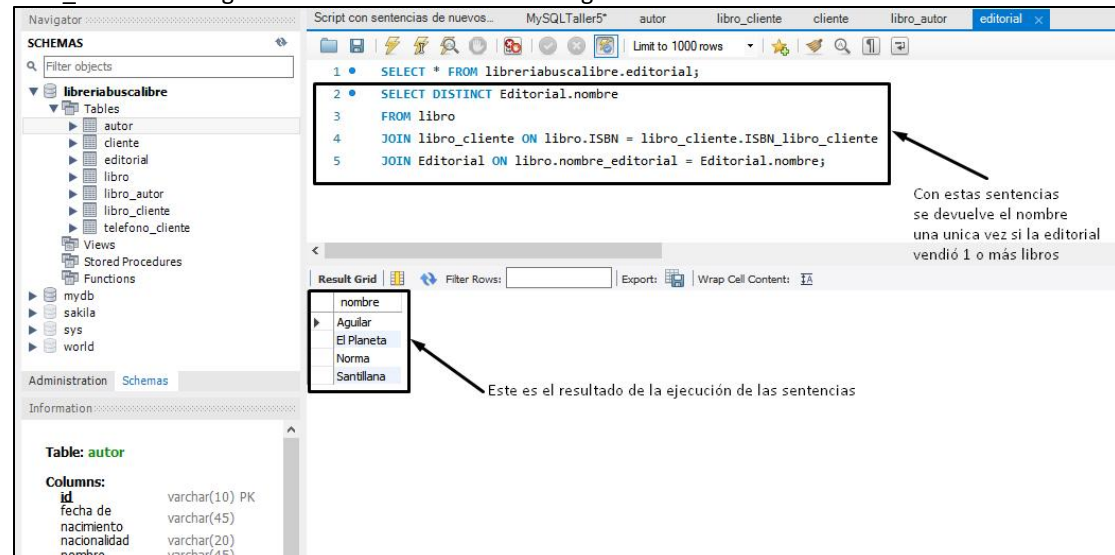
Estas son las sentencias para devolver el libro que tiene asociado un autor.

Result Grid

titulo	nombre
El otoño del patriarca	Gabriel García Márquez
El amor en los tiempos del cólera	Gabriel García Márquez
Del amor y otros demonios	Gabriel García Márquez
Noticia de un secuestro	Gabriel García Márquez
Vivir para contarla	Gabriel García Márquez
El general en su laberinto	Gabriel García Márquez
Cien años de soledad	Gabriel García Márquez
La vorágine	Gabriel García Márquez
La increíble y triste historia de la cándida	Gabriel García Márquez
El ruido de las cosas al caer	Juan Vazquez

Este es el resultado de la ejecución de las sentencias.

El nombre de las editoriales que han logrado vender libros. Asumiendo que los registros en la tabla libro_cliente son registros de venta. Se obtiene los siguiente:



Script con sentencias de nuevos...

```
1 SELECT * FROM libreriaescalibre.editorial;
2 SELECT DISTINCT Editorial.nombre
3 FROM libro
4 JOIN libro_cliente ON libro.ISBN = libro_cliente.ISBN_libro_cliente
5 JOIN Editorial ON libro.nombre_editorial = Editorial.nombre;
```

Con estas sentencias se devuelve el nombre una unica vez si la editorial vendió 1 o más libros

Result Grid

nombre
Agullar
El Planeta
Norma
Santillana

Este es el resultado de la ejecución de las sentencias

- Realice las dos vistas que considere sean las más importantes y explique el motivo de su selección.

En función de la estrategia del negocio es importante conocer cuales son los libros más vendidos en ese contexto crearía una vista para que el dueño del negocio analice la conveniencia de tener en el inventario más cantidad de libros de un determinado título como sigue:

Se crea la vista: vista_ventas_libros2 para identificar cuales son los libros más vendidos.

```
1  /*
2  Vista 1
3  */
4  CREATE VIEW vista_ventas_libros2 AS
5  SELECT lc.ISBN_libro_cliente, l.titulo, COUNT(*) as cantidad_ventas
6  FROM libro_cliente lc
7  JOIN libro l ON lc.ISBN_libro_cliente = l.ISBN
8  GROUP BY lc.ISBN_libro_cliente, l.titulo;
```

El resultado de la vista creada para los datos ingresados es el siguiente:

Resultado de la vista creada para identificar la cantidad de libros vendidos de un determinado título

ISBN_libro_cliente	titulo	cantidad_ventas
42-1071-1	El otoño del patriarca	1
42-1889-7	Del amor y otros demonios	1
42-2814-9	El ruido de las cosas al caer	2
42-3073-3	El general en su laberinto	2
42-7452-8	Cien años de soledad	2
42-7467-2	La vorágine	2

Por otra parte como parte de un eventual programa de fidelización y comunicación con los clientes es conveniente identificar cuales son los clientes que más compran en la tienda y así mismo tener con fácil acceso sus números de teléfono; por lo anterior crearía una vista como sigue:

Con estas sentencias se crea una vista para tener rapido acceso a la información de cuantos libros a comprado cada cliente y su número telefónico.

El siguiente es el resultado de la vista creada previamente:

Este es el resultado se observa el total de compras realizadas por cada cliente y como en su mayoría cada cliente tiene 2 número entonces se muestra el total de compras y sus dos números asociados.

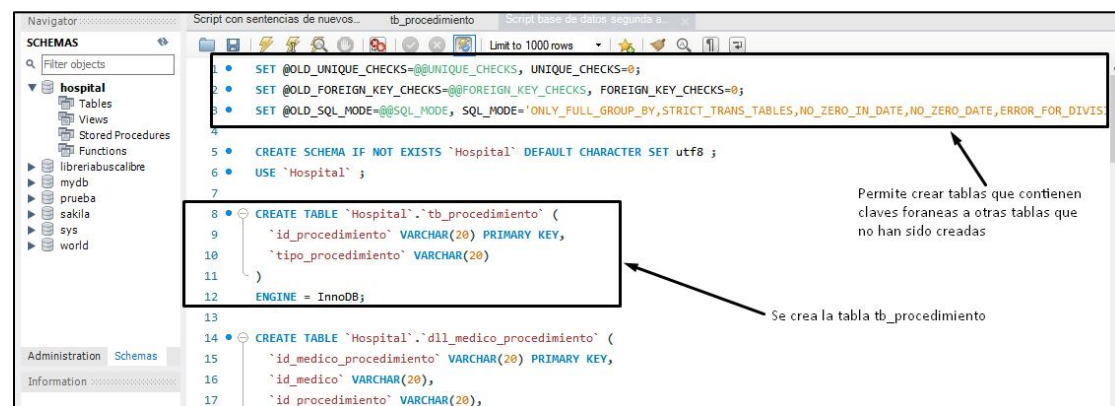
nombre_cliente	num_compras	telefono
Mabel Cartagena	1	3116884652
Pilar Schmitt	1	3155884652
Pilar Schmitt	1	3215884652
Carlos Calero	2	3115884652
Carlos Calero	2	3125884652
Andrea Serna	1	3118887652
Jota Mario Valencia	2	3135884652
Jota Mario Valencia	2	3145884652
Alejandro Riaño	1	3217884652
Alejandro Riaño	1	3225884652
Tatán Mejía	2	3113884652
Tatán Mejía	2	3114884652

Segunda actividad

<https://github.com/Ococho/Hospital-GNECJ.git>. Utilizando el ejercicio del hospital realizado por sus compañeros realice lo siguiente:

- Convierta el MR en una base de datos en MySQL utilizando sentencias SQL o el diagrama EER.

Se usaron sentencias en SQL para la creación de las tablas como sigue, se crea la tabla `tb_procedimiento`:



```
1 • SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
2 • SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
3 • SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVIS
4
5 • CREATE SCHEMA IF NOT EXISTS `Hospital` DEFAULT CHARACTER SET utf8 ;
6 • USE `Hospital` ;
7
8 • CREATE TABLE `Hospital`.`tb_procedimiento` (
9   `id_procedimiento` VARCHAR(20) PRIMARY KEY,
10  `tipo_procedimiento` VARCHAR(20)
11 )
12 ENGINE = InnoDB;
13
14 • CREATE TABLE `Hospital`.`dll_medico_procedimiento` (
15   `id_medico_procedimiento` VARCHAR(20) PRIMARY KEY,
16   `id_medico` VARCHAR(20),
17   `id_procedimiento` VARCHAR(20),
```

Permite crear tablas que contienen claves foráneas a otras tablas que no han sido creadas

Se crea la tabla `tb_procedimiento`

Se crea la tabla `dll_medico_procedimiento`:

```
15
16 • CREATE TABLE `Hospital`.`dll_medico_procedimiento` (
17   `id_medico_procedimiento` VARCHAR(20) PRIMARY KEY,
18   `id_medico` VARCHAR(20),
19   `id_procedimiento` VARCHAR(20),
20   FOREIGN KEY (`id_medico`)
21     REFERENCES `Hospital`.`tb_medico` (`id_medico`),
22   FOREIGN KEY (`id_procedimiento`)
23     REFERENCES `Hospital`.`tb_procedimiento` (`id_procedimiento`)
24 )
25 ENGINE = InnoDB;
26
```

Se crea la tabla `tb_paciente`:

```
26
27 • CREATE TABLE `Hospital`.`tb_paciente` (
28   `id_paciente` VARCHAR(50) NOT NULL,
29   `nombre_paciente` VARCHAR(50) NOT NULL,
30   `apellido_paciente` VARCHAR(50) NOT NULL,
31   `direccion` VARCHAR(100) NOT NULL,
32   `id_procedimiento` VARCHAR(50),
33   PRIMARY KEY (`id_paciente`),
34   CONSTRAINT `fk_id_procedimiento`
35     FOREIGN KEY (`id_procedimiento`)
36     REFERENCES `tb_procedimiento` (`id_procedimiento`)
37 ) ENGINE = InnoDB;
38
```

Se crea la tabla tb_factura:

```
38
39 • CREATE TABLE `Hospital`.`tb_factura` (
40     `id_factura` VARCHAR(20) PRIMARY KEY,
41     `fecha` VARCHAR(20),
42     `valor_total` VARCHAR(20),
43     `id_paciente` VARCHAR(20),
44     FOREIGN KEY (`id_paciente`)
45     REFERENCES `Hospital`.`tb_paciente` (`id_paciente`)
46 )
47 ENGINE = InnoDB;
```

Se crea la tabla telefono_paciente:

```
48
49 • CREATE TABLE `Hospital`.`telefono_paciente` (
50     `id_paciente` VARCHAR(20),
51     `telefono` VARCHAR(20),
52     FOREIGN KEY (`id_paciente`)
53     REFERENCES `Hospital`.`tb_paciente` (`id_paciente`)
54 )
55 ENGINE = InnoDB;
```

Se crea la tabla tb_medico:

```
57
58 • CREATE TABLE `Hospital`.`tb_medico` (
59     `id_medico` VARCHAR(20) PRIMARY KEY,
60     `nombre_medico` VARCHAR(50),
61     `apellido_medico` VARCHAR(50),
62     `especialidad` VARCHAR(50)
63 )
64 ENGINE = InnoDB;
65
```

Se crea la tabla tb_enfermero:

```
66 • CREATE TABLE `Hospital`.`tb_enfermero` (
67     `id_enfermero` VARCHAR(20) PRIMARY KEY,
68     `nombre_enfermero` VARCHAR(50),
69     `apellido_enfermero` VARCHAR(50),
70     `id_medico` VARCHAR(20),
71     FOREIGN KEY (`id_medico`)
72     REFERENCES `Hospital`.`tb_medico` (`id_medico`)
73 )
74 ENGINE = InnoDB;
```

Se crea la tabla telefono_paciente2:

```
75
76 • CREATE TABLE `Hospital`.`telefono_paciente2` (
77     `id_paciente2` VARCHAR(20),
78     `telefono2` VARCHAR(20),
79     FOREIGN KEY (`id_paciente2`)
80     REFERENCES `Hospital`.`tb_medico` (`id_medico`)
81 )
82 ENGINE = InnoDB;
```

Se crea la tabla dll_paciente_medicamento:

```
83
84 ● CREATE TABLE `Hospital`.`dll_paciente_medicamento` (
85     `id_paciente_medicamento` INT PRIMARY KEY,
86     `id_paciente` VARCHAR(20),
87     `id_medicamento` VARCHAR(20),
88     FOREIGN KEY (`id_paciente`)
89     REFERENCES `Hospital`.`tb_paciente` (`id_paciente`),
90     FOREIGN KEY (`id_medicamento`)
91     REFERENCES `Hospital`.`tb_medicamento` (`id_medicamento`)
92 )
93 ENGINE = InnoDB;
```

Se crea la tabla tb_medicamento:

```
95 ● CREATE TABLE `Hospital`.`tb_medicamento` (
96     `id_medicamento` VARCHAR(20) PRIMARY KEY,
97     `nombre_medicamento` VARCHAR(50),
98     `dosis` VARCHAR(50)
99 )
100 ENGINE = InnoDB;
101
102 ● CREATE TABLE `Hospital`.`telefono_paciente3` (
103     `id_paciente` VARCHAR(20),
104     `telefono` VARCHAR(20),
105     FOREIGN KEY (`id_paciente`)
106     REFERENCES `Hospital`.`tb_enfermero` (`id_enfermero`)
107 )
108 ENGINE = InnoDB;
109
110 ● SET SQL_MODE=@OLD_SQL_MODE;
111 ● SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
112 ● SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

- Complete la información para las tablas realizadas con al menos 5 registros por tabla.

Ingreso de registros para la tabla tb_procedimiento:

The screenshot shows a database management interface with a Navigator pane on the left displaying the 'hospital' schema. The main window shows a script editor with SQL commands for creating and inserting data into the 'tb_procedimiento' table. Below the script editor, a 'Result Grid' displays the data inserted into the table.

Script con sentencias de nuevos...

```
1 ● SELECT * FROM hospital.tb_procedimiento;
2 ● INSERT INTO `Hospital`.`tb_procedimiento` (`id_procedimiento`, `tipo_procedimiento`)
3     VALUES
4     ('P006', 'Cirugía cadera'),
5     ('P007', 'Radiografía tórax'),
6     ('P008', 'Colonoscopia'),
7     ('P009', 'Trat de conducto'),
8     ('P010', 'Extracción muela');
```

Result Grid

id_procedimiento	tipo_procedimiento
P006	Cirugía de cadera
P007	Radiografía de tórax
P008	Colonoscopia
P009	Trat de conducto
P010	Extracción de muela
NULL	NULL

Resultados del ingreso de los registros

Sentencias para el ingreso de los 5 nuevos registros.

Ingreso de registros para la tabla dll_medico_procedimiento

The screenshot shows a database management interface with a schema named 'hospital'. The 'Tables' folder is expanded, showing various tables including 'dll_medico_procedimiento'. The 'Scripts' tab is active, displaying a SQL script for inserting data into the 'dll_medico_procedimiento' table. The script consists of a SELECT statement to view existing data and an INSERT INTO statement to add new records. The 'Result Grid' at the bottom shows the data after the insert operation.

```
1 SELECT * FROM hospital.dll_medico_procedimiento;
2 INSERT INTO `Hospital`.`dll_medico_procedimiento` (`id_medico_procedimiento`,`id_medico`,`id_procedimiento`)
3 VALUES
4 ('MP001','M004','P006'),
5 ('MP002','M005','P007'),
6 ('MP003','M002','P008'),
7 ('MP004','M001','P009'),
8 ('MP005','M003','P010');
```

id_medico_procedimiento	id_medico	id_procedimiento
MP001	M004	P006
MP002	M005	P007
MP003	M002	P008
MP004	M001	P009
MP005	M003	P010
NULL	NULL	NULL

Ingreso de registros para la tabla dll_paciente_medicamento

The screenshot shows a database management interface with a schema named 'hospital'. The 'Scripts' tab is active, displaying a SQL script for inserting data into the 'dll_paciente_medicamento' table. The script consists of a SELECT statement to view existing data and an INSERT INTO statement to add new records. The 'Result Grid' at the bottom shows the data after the insert operation.

```
1 SELECT * FROM hospital.dll_paciente_medicamento;
2 INSERT INTO `Hospital`.`dll_paciente_medicamento` (`id_paciente_medicamento`,`id_paciente`,`id_medimento`)
3 VALUES
4 ('907','P005','M002'),
5 ('932','P008','M003'),
6 ('931','P003','M001'),
7 ('901','P009','M002'),
8 ('902','P001','M004');
```

id_paciente_medicamento	id_paciente	id_medimento
901	P009	M002
902	P001	M004
931	P003	M001
932	P008	M003
907	P005	M002
NULL	NULL	NULL

Ingreso de registros para la tabla tb_enfermero

The screenshot shows a database management interface with a schema named 'hospital'. The 'Scripts' tab is active, displaying a SQL script for inserting data into the 'tb_enfermero' table. The script consists of a SELECT statement to view existing data and an INSERT INTO statement to add new records. The 'Result Grid' at the bottom shows the data after the insert operation.

```
1 SELECT * FROM hospital.tb_enfermero;
2 INSERT INTO `Hospital`.`tb_enfermero` (`id_enfermero`,`nombre_enfermero`,`apellido_enfermero`,`id_medico`)
3 VALUES
4 ('E006','José','Pérez','M004'),
5 ('E007','Ana','Castro','M005'),
6 ('E008','Elena','Ramírez','M002'),
7 ('E009','Pedro','Gómez','M001'),
8 ('E010','Sara','González','M003');
```

id_enfermero	nombre_enfermero	apellido_enfermero	id_medico
E006	José	Pérez	M004
E007	Ana	Castro	M005
E008	Elena	Ramírez	M002
E009	Pedro	Gómez	M001
E010	Sara	González	M003
NULL	NULL	NULL	NULL

Ingreso de registros para la tabla tb_factura:

The screenshot shows the SQL Studio interface with the 'tb_factura' script selected. The script contains an INSERT statement with 10 rows of data. The 'Result Grid' at the bottom displays the inserted records.

```
1 • SELECT * FROM hospital.tb_factura;  
2 • INSERT INTO `Hospital`.`tb_factura` (`id_factura`, `fecha`, `valor_total`, `id_paciente`)  
3   VALUES  
4   ('F006', '2022-01-20', '4000', 'P003'),  
5   ('F007', '2022-02-10', '3500', 'P005'),  
6   ('F008', '2022-02-12', '2900', 'P002'),  
7   ('F009', '2022-02-13', '8100', 'P004'),  
8   ('F010', '2022-02-14', '3200', 'P001');
```

id_factura	fecha	valor_total	id_paciente
F006	2022-01-20	4000	P003
F007	2022-02-10	3500	P005
F008	2022-02-12	2900	P002
F009	2022-02-13	8100	P004
F010	2022-02-14	3200	P001
NULL	NULL	NULL	NULL

Ingreso de registros para la tabla tb_medicamento

The screenshot shows the SQL Studio interface with the 'tb_medicamento' script selected. The script contains an INSERT statement with 5 rows of data. The 'Result Grid' at the bottom displays the inserted records.

```
1 • SELECT * FROM hospital.tb_medicamento;  
2 • INSERT INTO `Hospital`.`tb_medicamento` (`id_medicamento`, `nombre_medicamento`, `dosis`)  
3   VALUES  
4   ('M004', 'Amoxicilina', '1 al día'),  
5   ('M003', 'Ibuprofeno', '2 al día'),  
6   ('M002', 'Paracetamol', '1 al día'),  
7   ('M005', 'Lorazepam', '2 al día'),  
8   ('M001', 'Aspirina', '3 al día');
```

id_medicamento	nombre_medicamento	dosis
M001	Aspirina	3 al día
M002	Paracetamol	1 al día
M003	Ibuprofeno	2 al día
M004	Amoxicilina	1 al día
M005	Lorazepam	2 al día
NULL	NULL	NULL

Ingreso de registros para la tabla tb_medico

The screenshot shows the SQL Studio interface with the 'tb_medico' script selected. The script contains an INSERT statement with 10 rows of data. The 'Result Grid' at the bottom displays the inserted records.

```
1 • SELECT * FROM hospital.tb_medico;  
2 • INSERT INTO `Hospital`.`tb_medico` (`id_medico`, `nombre_medico`, `apellido_medico`, `especialidad`)  
3   VALUES  
4   ('M006', 'Laura', 'Gómez', 'Cardiología'),  
5   ('M007', 'Jorge', 'Salazar', 'Oftalmología'),  
6   ('M008', 'Lucas', 'Torres', 'Pediatria'),  
7   ('M009', 'María', 'Fernández', 'Ginecología'),  
8   ('M010', 'Sofía', 'Ortiz', 'Neurología');
```

id_medico	nombre_medico	apellido_medico	especialidad
M006	Laura	Gómez	Cardiología
M007	Jorge	Salazar	Oftalmología
M008	Lucas	Torres	Pediatria
M009	María	Fernández	Ginecología
M010	Sofía	Ortiz	Neurología
NULL	NULL	NULL	NULL

Ingreso de registros para la tabla telefono_paciente

The screenshot shows the SQL Enterprise Manager interface. On the left, the 'Schemas' pane displays the 'hospital' schema with various tables. The main pane shows a script with the following SQL commands:

```
1 • SELECT * FROM hospital.telefono_paciente;
2 • INSERT INTO `Hospital`.`telefono_paciente` (`id_paciente`, `telefono`)
3 VALUES
4 ('P001', '555-1234'),
5 ('P002', '555-4321'),
6 ('P003', '555-6789'),
7 ('P004', '555-9876'),
8 ('P005', '555-1111');
```

Below the script, the 'Result Grid' displays the data inserted into the 'telefono_paciente' table:

id_paciente	telefono
P001	555-1234
P002	555-4321
P003	555-6789
P004	555-9876
P005	555-1111

Ingreso de registros para la tabla telefono_paciente2

The screenshot shows the SQL Enterprise Manager interface. On the left, the 'Schemas' pane displays the 'hospital' schema. The main pane shows a script with the following SQL commands:

```
1 • SELECT * FROM hospital.telefono_paciente2;
2 • INSERT INTO `Hospital`.`telefono_paciente2` (`id_paciente2`, `telefono2`)
3 VALUES
4 ('P006', '555-2468'),
5 ('P007', '555-8642'),
6 ('P008', '555-3972'),
7 ('P009', '555-7654'),
8 ('P010', '555-1029');
```

Below the script, the 'Result Grid' displays the data inserted into the 'telefono_paciente2' table:

id_paciente2	telefono2
P006	555-2468
P007	555-8642
P008	555-3972
P009	555-7654
P010	555-1029

Ingreso de registros para la tabla telefono_paciente3

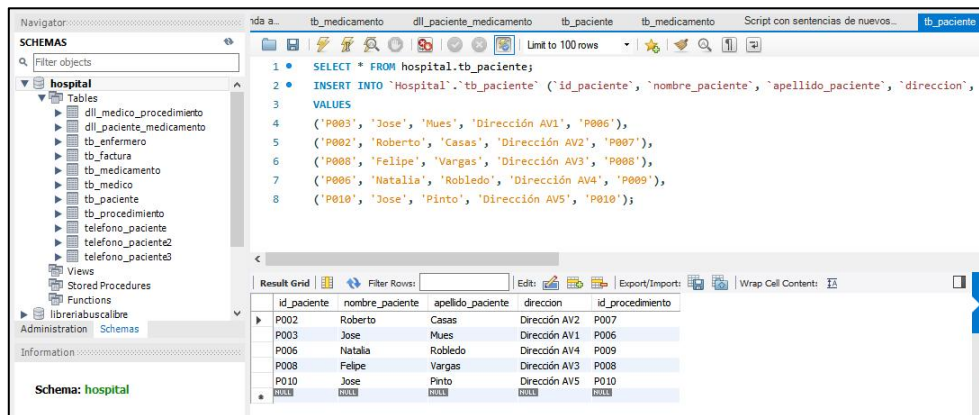
The screenshot shows the SQL Enterprise Manager interface. On the left, the 'Schemas' pane displays the 'hospital' schema. The main pane shows a script with the following SQL commands:

```
1 • SELECT * FROM hospital.telefono_paciente3;
2 • INSERT INTO `Hospital`.`telefono_paciente3` (`id_paciente`, `telefono`)
3 VALUES
4 ('P003', '555-111-3456'),
5 ('P006', '555-222-7890'),
6 ('P010', '555-333-4567'),
7 ('P008', '555-444-1234'),
8 ('P002', '555-555-5678');
```

Below the script, the 'Result Grid' displays the data inserted into the 'telefono_paciente3' table:

id_paciente	telefono
P003	555-111-3456
P006	555-222-7890
P010	555-333-4567
P008	555-444-1234
P002	555-555-5678

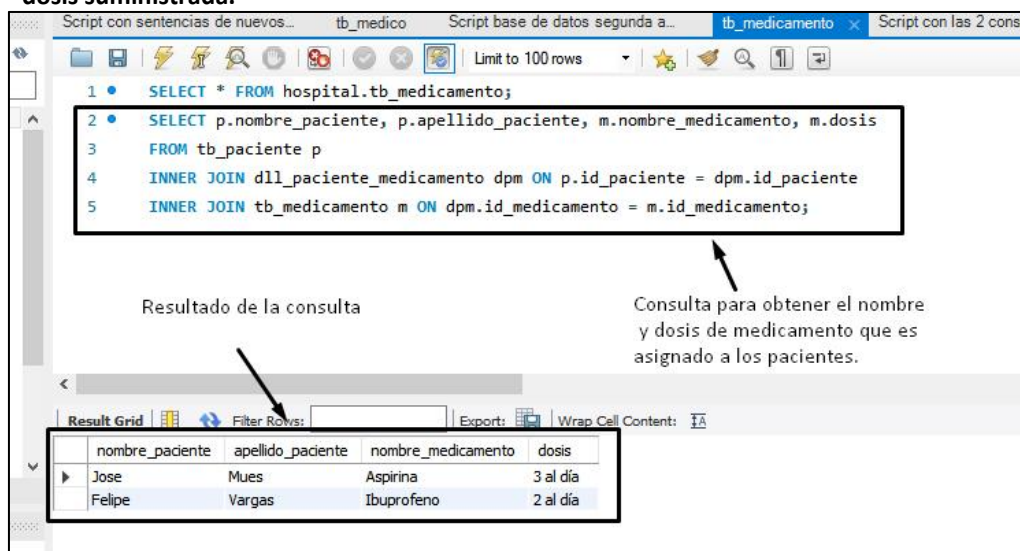
Ingreso de registros en la tabla tb_paciente:



```
1 • SELECT * FROM hospital.tb_paciente;
2 • INSERT INTO "Hospital"."tb_paciente" ('id_paciente', 'nombre_paciente', 'apellido_paciente', 'direccion', 'id_procedimiento')
3 VALUES
4 ('P003', 'Jose', 'Mues', 'Dirección AV1', 'P006'),
5 ('P002', 'Roberto', 'Casas', 'Dirección AV2', 'P007'),
6 ('P008', 'Felipe', 'Vargas', 'Dirección AV3', 'P008'),
7 ('P006', 'Natalia', 'Robledo', 'Dirección AV4', 'P009'),
8 ('P010', 'Jose', 'Pinto', 'Dirección AV5', 'P010');
```

id_paciente	nombre_paciente	apellido_paciente	direccion	id_procedimiento
P002	Roberto	Casas	Dirección AV2	P007
P003	Jose	Mues	Dirección AV1	P006
P006	Natalia	Robledo	Dirección AV4	P009
P008	Felipe	Vargas	Dirección AV3	P008
P010	Jose	Pinto	Dirección AV5	P010
NULL	NULL	NULL	NULL	NULL

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



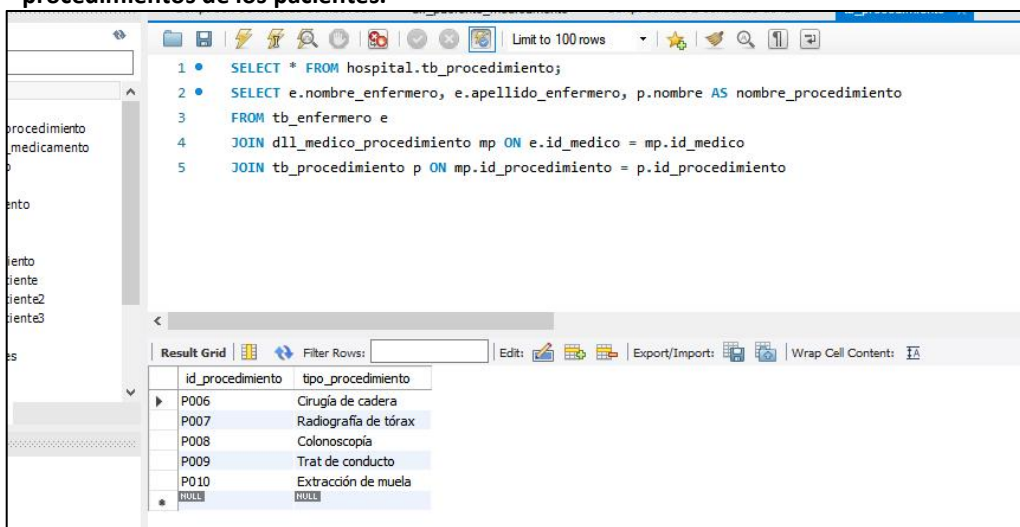
```
1 • SELECT * FROM hospital.tb_medimento;
2 • SELECT p.nombre_paciente, p.apellido_paciente, m.nombre_medimento, m.dosis
3 FROM tb_paciente p
4 INNER JOIN dll_paciente_medimento dpm ON p.id_paciente = dpm.id_paciente
5 INNER JOIN tb_medimento m ON dpm.id_medimento = m.id_medimento;
```

Resultado de la consulta

nombre_paciente	apellido_paciente	nombre_medimento	dosis
Jose	Mues	Aspirina	3 al día
Felipe	Vargas	Ibuprofeno	2 al día

Consulta para obtener el nombre y dosis de medicamento que es asignado a los pacientes.

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



```
1 • SELECT * FROM hospital.tb_procedimiento;
2 • SELECT e.nombre_enfermero, e.apellido_enfermero, p.nombre AS nombre_procedimiento
3 FROM tb_enfermero e
4 JOIN dll_medico_procedimiento mp ON e.id_medico = mp.id_medico
5 JOIN tb_procedimiento p ON mp.id_procedimiento = p.id_procedimiento
```

id_procedimiento	tipo_procedimiento
P006	Cirugía de cadera
P007	Radiografía de tórax
P008	Colonoscopia
P009	Trat de conducto
P010	Extracción de muela
NULL	NULL

- Realice las tres vistas que considere sean las más importantes y explique el motivo de su selección.

Es útil y conveniente conocer rapidamente el nombre de cada paciente, el procedimiento que se le practicó, el medicamento que esta tomando y su dosis respectiva.

The screenshot shows the SQL Developer interface. On the left, the 'SCHEMAS' pane lists various tables and views. The main editor displays the following SQL code:

```
1 CREATE VIEW `vista_paciente_procedimiento_medimento` AS
2 SELECT p.nombre_paciente, pr.tipo_procedimiento, m.nombre_medimento, m.dosis
3 FROM tb_paciente p
4 JOIN tb_factura f ON p.id_paciente = f.id_paciente
5 JOIN tb_procedimiento pr ON p.id_procedimiento = pr.id_procedimiento
6 JOIN dll_paciente_medimento dpm ON p.id_paciente = dpm.id_paciente
7 JOIN tb_medimento m ON dpm.id_medimento = m.id_medimento;
```

An arrow points to the code with the text: "Sentencias para crear la vista e identificar el paciente con su procedimiento, medicamento y dosis asociada."

The screenshot shows the SQL Developer interface with the query executed. The main editor displays the following SQL code:

```
1 SELECT * FROM hospital.vista_paciente_procedimiento_medimento;
```

An arrow points to the 'Result Grid' pane with the text: "Resultado de la vista". The 'Result Grid' pane shows the following data:

nombre_paciente	tipo_procedimiento	nombre_medimento	dosis
Jose	Cirugía de cadera	Aspirina	3 al día