Taller Consultas, Visitas, Procedimientos y Trigger.

Jesús Miguel Molina Mendoza

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 únicamente comandos SQL creados por usted.

Para hacer los registros se usa la sentencia INSERT INTO con el nombre de la tabla y entre paréntesis los campos que van a ser llenados, luego se usa la sentencia VALUES y entre paréntesis se colocan los datos para cada campo.

Instrucciones SQL para poblar la tabla autor con 5 registros

```
INSERT INTO autor (id, fecha_de_nacimiento, nacionalidad, nombre)
VALUES ('1','6 de noviembre de 1934','Estadounidense','Carl Sagan'), ('2','8 de enero de 1942','Británico','Stephen Hawking'),
       ('3','8 de abril de 1947','Estadounidense','Robert Kiyosaki'),('4','6 de marzo de 1927','Colombiano','Gabriel García Márquez'),
       ('5','15 de octubre de 1844','Aleman','Friedrich Nietzsche');
```

id	fecha_de_nacimiento	nacionalidad	nombre
1	6 de noviembre de 1934	Estadounidense	Carl Sagan
2	8 de enero de 1942	Británico	Stephen Hawking
3	8 de abril de 1947	Estadounidense	Robert Kiyosaki
4	6 de marzo de 1927	Colombiano	Gabriel García Márquez
5	15 de octubre de 1844	Aleman	Friedrich Nietzsche
NULL	NULL	NULL	NULL

Instrucciones SQL para poblar la tabla libro con 20 registros

```
INSERT INTO libro (ISBN, titulo, numero_paginas,nombre_editorial)
VALUES ('001','Contacto','432','Planeta'),('002','Cosmos','384','Alianza'),('003','El mundo y sus demonios','457','Babel Libros'),('004','Un punto azul pálido','448','Gamma'),
       ('005','Los dragones del Edén','263','Planeta'),

       ('006','Breve historia del tiempo','216','Alianza'),('007','La teoría del todo','152','Babel Libros'),('008','Breves respuestas a las grandes preguntas','288','Gamma'),
       ('009','El gran diseño','208','Planeta'),('010','El universo en una cáscara de nuez','256','Alianza'),

       ('011','Padre Rico, Padre Pobre','232','Babel Libros'),('012','El cuadrante del flujo de dinero','312','Gamma'),('013','Guía para invertir','672','Planeta'),
       ('014','Queremos que seas rico','416','Alianza'),('015','Fake','240','Babel Libros'),

       ('016','Cien años de soledad','496','Gamma'),('017','El amor en los tiempos del cóleras','461','Planeta'),

       ('018','Más allá del bien y del mal','302','Alianza'),('019','El Anticristo','124','Babel Libros'),('020','La gaya ciencia','260','Gamma');
```

ISBN	titulo	numero_paginas	nombre_editorial
001	Contacto	432	Planeta
002	Cosmos	384	Alianza
003	El mundo y sus demonios	457	Babel Libros
004	Un punto azul pálido	448	Gamma
005	Los dragones del Edén	263	Planeta
006	Breve historia del tiempo	216	Alianza
007	La teoría del todo	152	Babel Libros
008	Breves respuestas a las grandes preguntas	288	Gamma
009	El gran diseño	208	Planeta
010	El universo en una cáscara de nuez	256	Alianza
011	Padre Rico, Padre Pobre	232	Babel Libros
012	El cuadrante del flujo de dinero	312	Gamma
013	Guía para invertir	672	Planeta
014	Queremos que seas rico	416	Alianza
015	Fake	240	Babel Libros
016	Cien años de soledad	496	Gamma
017	El amor en los tiempos del cóleras	461	Planeta
018	Más allá del bien y del mal	302	Alianza
019	El Anticristo	124	Babel Libros
020	La gaya ciencia	260	Gamma
NULL	NULL	NULL	NULL

Instrucciones SQL para poblar la tabla cliente con 7 registros

```
INSERT INTO cliente (cedula, nombre)
VALUES ('1010101010','Alejandro Mendez'),('2020202020','Bernardo López'),('3030303030','Carlos Ruiz'),('4040404040','César Morales'),
       ('5050505050','Eliza Saban'),('6060606060','Nancy Calderón'),('7070707070','Vicente Vásquez');
```

cedula	nombre
1010101010	Alejandro Mendez
2020202020	Bernardo López
3030303030	Carlos Ruiz
4040404040	César Morales
5050505050	Eliza Saban
6060606060	Nancy Calderón
7070707070	Vicente Vásquez
NULL	NULL

Instrucciones SQL para poblar la tabla editorial con 4 registros

```
INSERT INTO editorial (nombre, ciudad, complemento, telefono)
VALUES ('Planeta','Bogota','Calle 73','6013909541'),('Alianza','Bogota','Carrera 6','6013909541'),
       ('Babel Libros','Bogota','Calle 39a','6013909541'),('Gamma','Bogota','Calle 85','6013909541');
```

nombre	ciudad	complemento	Telefono
Alianza	Bogota	Carrera 6	6013909541
Babel Libros	Bogota	Calle 39a	6013909541
Gamma	Bogota	Calle 85	6013909541
Planeta	Bogota	Calle 73	6013909541
NULL	NULL	NULL	NULL

Instrucciones SQL para poblar la tabla libro_cliente con 10 registros

```
INSERT INTO libro_cliente (ISBN_libro_cliente, id_cliente)
VALUES ('001','1010101010'),('006','1010101010'),('011','2020202020'),('012','2020202020'),('012','3030303030'),('016','3030303030'),('018','4040404040'),('020','5050505050'),
       ('013','6060606060'),('003','7070707070');
```

ISBN_libro_cliente	id_cliente
001	1010101010
006	1010101010
011	2020202020
012	2020202020
012	3030303030
016	3030303030
018	4040404040
020	5050505050
013	6060606060
003	7070707070
NULL	NULL

Instrucciones SQL para poblar la tabla libro_autor con 10 registros

```
160 •  INSERT INTO libro_autor (ISBN_libro, id_autor)
161    VALUES ('001','1'),('002','1'),('006','2'),('007','2'),('011','3'),('012','3'),('016','4'),('017','4'),('018','5'),('019','5');
```

ISBN_libro	id_autor
001	1
002	1
006	2
007	2
011	3
012	3
016	4
017	4
018	5
019	5
NULL	NULL

Instrucciones SQL para poblar la tabla teléfono_cliente con 12 registros

```
166 •  INSERT INTO telefono_cliente (cedula_cliente, numero)
167    VALUES ('1010101010','+57 3001234567'),('2020202020','+57 3104567890'),('3030303030','+57 3207890123'),('4040404040','+57 3301234567'),
168           ('5050505050','+57 3404567890'),('6060606060','+57 3507890123'),('7070707070','+57 3601234567'),('1010101010','+57 3001111111'),
169           ('2020202020','+57 3102222222'),('3030303030','+57 3203333333'),('4040404040','+57 3304444444'),('5050505050','+57 3405555555');
```

cedula_cliente	numero
1010101010	+57 3001111111
1010101010	+57 3001234567
2020202020	+57 3102222222
2020202020	+57 3104567890
3030303030	+57 3203333333
3030303030	+57 3207890123
4040404040	+57 3301234567
4040404040	+57 3304444444
5050505050	+57 3404567890
5050505050	+57 3405555555
6060606060	+57 3507890123
7070707070	+57 3601234567
NULL	NULL

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 número telefónico, el nombre del libro acompañado por su autor o sus autores, el nombre de las editoriales que han logrado vender libros.

Consulta para conocer el nombre y la fecha de nacimiento de cada autor.

Para esta consulta usamos la instrucción SELECT que nos selecciona los datos que queremos traer de la tabla (en este caso nombre y fecha_de_nacimiento) y la instrucción FROM para seleccionar la tabla de la cual queremos que nos traiga los datos (en este caso de la tabla autor)

```
174 ●   SELECT nombre, fecha_de_nacimiento
175     FROM autor;
```

nombre	fecha_de_nacimiento
Carl Sagan	6 de noviembre de 1934
Stephen Hawking	8 de enero de 1942
Robert Kiyosaki	8 de abril de 1947
Gabriel García Márquez	6 de marzo de 1927
Friedrich Nietzsche	15 de octubre de 1844

Consulta para conocer la cantidad de libros vendidos diferentes.

Para esta consulta usamos la función COUNT() que me cuenta la cantidad de filas y la declaración SELECT DISTINCT para que me devuelva los valores distintos del campo ISBN_libro_cliente, se usa **as** para que me devuelva el resultado en un campo llamado cantidad_libros_diferentes y se unas FROM para decir que viene desde la tabla libro_cliente.

```
185     -- Consulta para conocer la cantidad de libros vendidos diferentes
186 ●   SELECT COUNT(DISTINCT ISBN_libro_cliente) as Cantidad_libros_diferentes
187     FROM libro_cliente;
```

Cantidad_libros_diferentes
9

Consulta para conocer el nombre de su cliente acompañado de su número telefónico.

Para esta consulta usamos la instrucción SELECT para especificar los campos de la consulta que queremos mostrar, en este caso el nombre de la tabla cliente y el numero de la tabla teléfono cliente. La cláusula "FROM" especifica las tablas que se van a utilizar en la consulta, y la cláusula "INNER JOIN" especifica la forma en que se unen estas tablas. Luego usamos un INNER JOIN para mostrar el resultado de aquellas filas en las que haya coincidencia en ambas tablas.

```
190 ●   SELECT cliente.nombre, telefono_cliente.numero
191     FROM cliente
192     INNER JOIN telefono_cliente ON cliente.cedula = telefono_cliente.cedula_cliente;
```

nombre	numero
Alejandro Mendez	+57 3001111111
Alejandro Mendez	+57 3001234567
Bernardo López	+57 3102222222
Bernardo López	+57 3104567890
Carlos Ruiz	+57 3203333333
Carlos Ruiz	+57 3207890123
César Morales	+57 3301234567
César Morales	+57 3304444444
Eliza Saban	+57 3404567890
Eliza Saban	+57 3405555555
Nancy Calderón	+57 3507890123
Vicente Vásquez	+57 3601234567

Consulta para conocer el nombre del libro acompañado por su autor o sus autores

Para esta consulta se debe combinar la información de 3 tablas que son libro, autor y libro_autor. La instrucción INNER JOIN libro_autor une la tabla libro con la tabla libro_autor con la condición ON libro.ISBN = libro_autor.ISBN_libro para traer solo las filas que tengan el mismo valor en las columnas ISBN.

Luego se utiliza la clausula INNER JOIN autor que une la tabla libro_autor con la tabla autor con la condición ON libro_autor.id_autor = autor.id para traer solo las filas que tengan el mismo valor en las columnas id

La instrucción SELECT libro.titulo, autor.nombre donde se especifican las columnas que se quieren traer.

```
194     -- Consulta para conocer el nombre del libro acompañado por su autor o sus autores
195 •   SELECT libro.titulo, autor.nombre
196     FROM libro
197     INNER JOIN libro_autor ON libro.ISBN = libro_autor.ISBN_libro
198     INNER JOIN autor ON libro_autor.id_autor = autor.id;
```

titulo	nombre
Contacto	Carl Sagan
Cosmos	Carl Sagan
Breve historia del tiempo	Stephen Hawking
La teoría del todo	Stephen Hawking
Padre Rico, Padre Pobre	Robert Kiyosaki
El cuadrante del flujo de dinero	Robert Kiyosaki
Cien años de soledad	Gabriel García Márquez
El amor en los tiempos del cóleras	Gabriel García Márquez
Más allá del bien y del mal	Friedrich Nietzsche
El Anticristo	Friedrich Nietzsche

Consulta para conocer el nombre de las editoriales que han logrado vender libros.

Para realizar esta consulta se deben combinar la tabla libro, con la tabla editorial y la tabla libro con la tabla libro_cliente. Para este se una una instrucción INNER JOIN editorial con la condición ON editorial.nombre = libro.nombre_editorial para traer las editoriales que tengan libros

Luego de esto se usa el INNER JOIN libro_cliente con la condición ON libro_cliente.ISBN_libro_cliente = libro.ISBN para mostrar los libros que tenga ventas. De esta forma solo se muestran las librerías que tienen libros con ventas.

Y se usa la sentencia SELECT DISTINCT nombre_editorial para que el campo que te muestre sea el del nombre_editorial y que no se repita.

```
201 ●    SELECT DISTINCT nombre_editorial
202      FROM libro
203      INNER JOIN editorial ON editorial.nombre = libro.nombre_editorial
204      INNER JOIN libro_cliente ON libro_cliente.ISBN_libro_cliente = libro.ISBN
```

nombre_editorial
Alianza
Babel Libros
Gamma
Planeta

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

Se crea una vista con la consulta gustos_clientes que nos permite ver los libros que ha comprado cada cliente, me parece que puede ser importante para una librería porque de esta forma puede conocer los gustos literarios de cada cliente y así poderle recomendar otros libros con la ayuda de esos gustos.

```
206 ●    CREATE VIEW gustos_clientes AS
207      SELECT cliente.nombre, libro.titulo
208      FROM libro_cliente
209      INNER JOIN libro ON libro_cliente.ISBN_libro_cliente = libro.ISBN
210      INNER JOIN cliente ON cliente.cedula = libro_cliente.id_cliente;
211
212 ●    SELECT *
213      FROM gustos_clientes
```

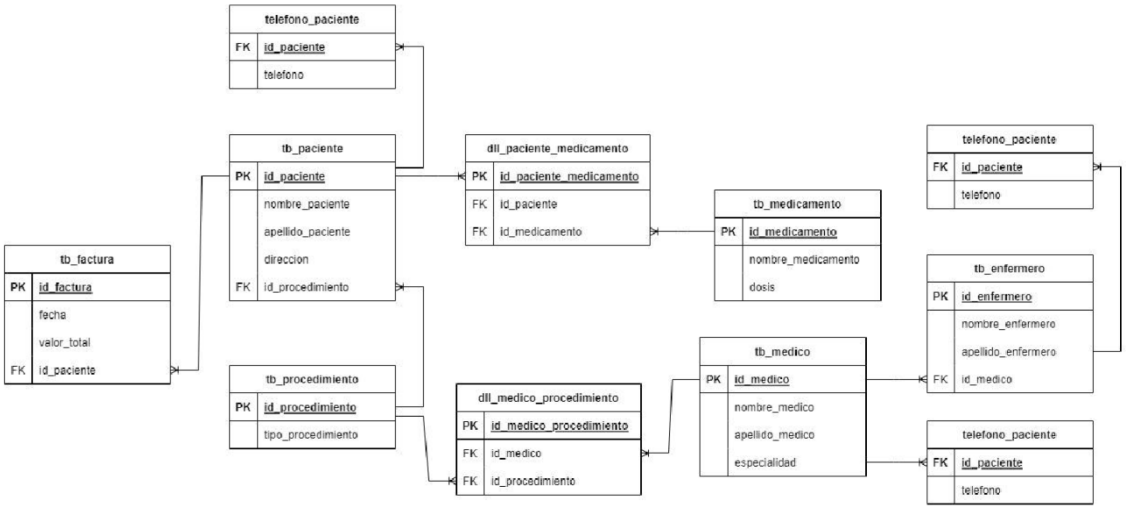
	nombre	titulo
▶	Alejandro Mendez	Contacto
	Alejandro Mendez	Breve historia del tiempo
	Bernardo López	Padre Rico, Padre Pobre
	Bernardo López	El cuadrante del flujo de dinero
	Carlos Ruiz	El cuadrante del flujo de dinero
	Carlos Ruiz	Cien años de soledad
	César Morales	Más allá del bien y del mal
	Eliza Saban	La gaya ciencia
	Nancy Calderón	Guía para invertir
	Vicente Vásquez	El mundo y sus demonios

Se crea la vista contacto cliente usando la consulta para conocer el nombre de su cliente acompañado de su número telefónico, elegí esta consulta para crear la vista porque para una librería puede ser importante tener el contacto de sus clientes para ofrecerle nuevos libros.

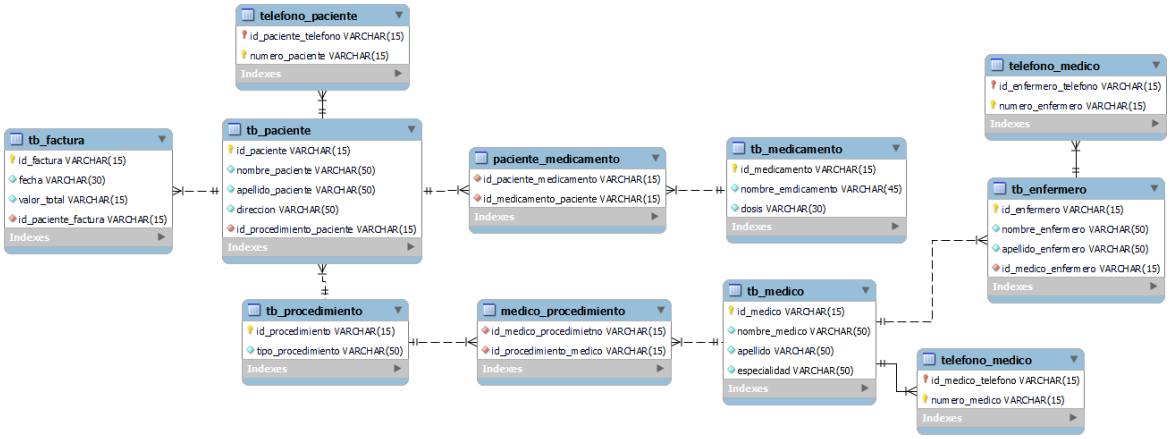
```
216     -- Vista para buscar el contacto del cliente
217 ●   CREATE VIEW contacto_cliente AS
218     SELECT cliente.nombre, telefono_cliente.numero
219     FROM cliente
220     INNER JOIN telefono_cliente ON cliente.cedula = telefono_cliente.cedula_cliente;
221
222 ●   SELECT *
223     FROM contacto_cliente;
```

	nombre	numero
▶	Alejandro Mendez	+57 3001111111
	Alejandro Mendez	+57 3001234567
	Bernardo López	+57 3102222222
	Bernardo López	+57 3104567890
	Carlos Ruiz	+57 3203333333
	Carlos Ruiz	+57 3207890123
	César Morales	+57 3301234567
	César Morales	+57 3304444444
	Eliza Saban	+57 3404567890
	Eliza Saban	+57 3405555555
	Nancy Calderón	+57 3507890123
	Vicente Vásquez	+57 3601234567

Segunda actividad:



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



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

Registro tabla medico

```
183 •   INSERT INTO tb_medico (id_medico, nombre_medico, apellido, especialidad)
184     VALUES('M001', 'Juan', 'García', 'Pediatría'),('M002', 'María', 'González', 'Oftalmología'),('M003', 'Santiago', 'Martínez', 'Cardiología'),
185           ('M004', 'Ana', 'Hernández', 'Dermatología'),('M005', 'Luis', 'Pérez', 'Neurología');
```

id_medico	nombre_medico	apellido	especialidad
M001	Juan	García	Pediatría
M002	María	González	Oftalmología
M003	Santiago	Martínez	Cardiología
M004	Ana	Hernández	Dermatología
M005	Luis	Pérez	Neurología
NULL	NULL	NULL	NULL

Registro tabla teléfono_medico

```
189     -- Se hacen 5 registros para la tabla telefono_medico
190 •   INSERT INTO telefono_medico (id_medico_telefono, numero_medico)
191     VALUES('M001', '3101234567'),('M002', '3219876543'),('M003', '3014567890'),('M004', '3003216549'),('M005', '3127418529');
```

id_medico_telefono	numero_medico
M001	3101234567
M002	3219876543
M003	3014567890
M004	3003216549
M005	3127418529
NULL	NULL

Registro tabla tb_procedimiento

```
196 •   INSERT INTO tb_procedimiento (id_procedimiento, tipo_procedimiento)
197     VALUES('P001', 'Extracción de sangre'),('P002', 'Radiografía'),('P003', 'Endoscopia'),('P004', 'Tomografía'),('P005', 'Cirugía de cataratas');
```

id_procedimiento	tipo_procedimiento
P001	Extracción de sangre
P002	Radiografía
P003	Endoscopia
P004	Tomografía
P005	Cirugía de cataratas
NULL	NULL

Registro tabla tb_procedimiento

```
203     -- Se hacen 5 registros para la tablas procedimiento
204 •   INSERT INTO medico_procedimiento (id_medico_procedimietno, id_procedimiento_medico)
205     VALUES('M001', 'P001'),('M001', 'P003'),('M002', 'P002'),('M003', 'P004'),('M004', 'P005');
```

id_medico_procedimietno	id_procedimiento_medico
M001	P001
M001	P003
M002	P002
M003	P004
M004	P005

Registro tabla tb_paciente

```
211 • INSERT INTO tb_paciente (id_paciente, nombre_paciente, apellido_paciente, direccion, id_procedimiento_paciente)
212   VALUES('1', 'María', 'García', 'Calle 123', 'P001'),('2', 'José', 'Rodríguez', 'Carrera 45', 'P002'),('3', 'Ana', 'Martínez', 'Avenida 67', 'P003'),
213         ('4', 'Pedro', 'Fernández', 'Calle 23', 'P004'),('5', 'Mónica', 'Hernández', 'Carrera 78', 'P005');
```

id_paciente	nombre_paciente	apellido_paciente	direccion	id_procedimiento_paciente
1	María	García	Calle 123	P001
2	José	Rodríguez	Carrera 45	P002
3	Ana	Martínez	Avenida 67	P003
4	Pedro	Fernández	Calle 23	P004
5	Mónica	Hernández	Carrera 78	P005
NULL	NULL	NULL	NULL	NULL

Registros tabla telefono_paciente

```
219 •  INSERT INTO telefono_paciente (id_paciente_telefono, numero_paciente)
220    VALUES('1','3124587698'),('2','3168457958'),('3','3188456587'),('4','3205845657'),('5','322584857');
```

id_paciente_telefono	numero_paciente
1	3124587698
2	3168457958
3	3188456587
4	3205845657
5	322584857
NULL	NULL

Registros tabla tb_factura

```
226 • INSERT INTO tb_factura (id_factura, fecha, valor_total, id_paciente_factura)
227   VALUES ('FAC-0001', '2022-02-01', '350000', '1'),('FAC-0002', '2022-02-10', '450000', '2'),('FAC-0003', '2022-02-15', '800000', '3'),
228         ('FAC-0004', '2022-02-20', '250000', '4'),('FAC-0005', '2022-03-01', '900000', '5');
```

id_factura	fecha	valor_total	id_paciente_factura
FAC-0001	2022-02-01	350000	1
FAC-0002	2022-02-10	450000	2
FAC-0003	2022-02-15	800000	3
FAC-0004	2022-02-20	250000	4
FAC-0005	2022-03-01	900000	5
NULL	NULL	NULL	NULL

Registros tabla tb_medicamento

```
233 • INSERT INTO tb_medicamento (id_medicamento, nombre_emdicamento, dosis)
234   VALUES('ME001', 'Ibuprofeno', '200mg'),('ME002', 'Paracetamol', '500mg'),('ME003', 'Omeprazol', '20mg'),('ME004', 'Aspirina', '100mg'),('ME005', 'Amoxicilina', '500mg');
```

id_medicamento	nombre_emdicamento	dosis
ME001	Ibuprofeno	200mg
ME002	Paracetamol	500mg
ME003	Omeprazol	20mg
ME004	Aspirina	100mg
ME005	Amoxicilina	500mg
NULL	NULL	NULL

Registros tabla paciente_medicamento

```
239 •  INSERT INTO paciente_medicamento (id_paciente_medicamento, id_medicamento_paciente)
240    VALUES ('ME001', '1'),('ME002', '2'),('ME003', '3'),('ME004', '4'),('ME005', '5');
```

	id_paciente_medicamento	id_medicamento_paciente
▶	ME001	1
	ME002	2
	ME003	3
	ME004	4
	ME005	5

Registros tabla tb_enfermero

```
248 •  INSERT INTO tb_enfermero (id_enfermero, nombre_enfermero, apellido_enfermero, id_medico_enfermero)
249    VALUES ('E001', 'Luisa', 'García', 'M001'),('E002', 'Gabriel', 'Rodríguez', 'M002'),('E003', 'Ana', 'Fernández', 'M003'),
250           ('E004', 'Carlos', 'González', 'M004'),('E005', 'María', 'Martínez', 'M005');
```

	id_enfermero	nombre_enfermero	apellido_enfermero	id_medico_enfermero
	E001	Luisa	García	M001
	E002	Gabriel	Rodríguez	M002
	E003	Ana	Fernández	M003
	E004	Carlos	González	M004
	E005	María	Martínez	M005
▶*	NULL	NULL	NULL	

Registro tabla teléfono_enfermero

```
256    -- Se hacen 5 registros para la tabla telefono_enfermero
257 •  INSERT INTO telefono_enfermero (id_enfermero_telefono, numero_enfermero)
258    VALUES('E001', '3111278543'),('E002', '3205875972'),('E003', '3011258768'),('E004', '3008756842'),('E005', '3125875457');
```

	id_enfermero_telefono	numero_enfermero
▶	E001	3111278543
	E002	3205875972
	E003	3011258768
	E004	3008756842
	E005	3125875457
*	NULL	NULL

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

Para esta consulta usamos la instrucción SELECT tb_paciente.nombre_paciente AS Nombre, tb_paciente.apellido_paciente AS Apellido, tb_medicamento.nombre_emdicamento AS Medicamento, tb_medicamento.dosis AS Dosis, de esta forma mostraremos en la tabla solo los campos que nos interesan y hacemos INNER JOIN desde la tabla paciente_medicamento y tabla tb_paciente con la condición tb_paciente ON tb_paciente.id_paciente = paciente_medicamento.id_paciente_medicamento para traer solo las filas que tengan similitud en los ids,

Además, hacemos un INNER JOIN desde la tabla paciente_medicamento y la tabla tb_medicamento con la condición tb_medicamento ON tb_medicamento.id_medicamento = paciente_medicamento.id_medicamento_paciente y así solo traemos las filas que tengan similitud en los ids

```
SELECT tb_paciente.nombre_paciente AS Nombre, tb_paciente.apellido_paciente AS Apellido, tb_medicamento.nombre_emdicamento AS Medicamento, tb_medicamento.dosis AS Dosis
FROM paciente_medicamento
INNER JOIN tb_paciente ON tb_paciente.id_paciente = paciente_medicamento.id_paciente_medicamento
INNER JOIN tb_medicamento ON tb_medicamento.id_medicamento = paciente_medicamento.id_medicamento_paciente
```

Nombre	Apellido	Medicamento	Dosis
María	García	Ibuprofeno	200mg
José	Rodríguez	Paracetamol	500mg
Ana	Martínez	Omeprazol	20mg
Pedro	Fernández	Aspirina	100mg
Mónica	Hernández	Amoxicilina	500mg

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

Para esta consulta se usa la instrucción SELECT con los campos de cada tabla que quiero ver, y se hacen varios INNER JOIN para evaluando los ids de cada tabla desde la tabla enfermero hasta la tabla paciente.

```
SELECT tb_enfermero.nombre_enfermero, tb_enfermero.apellido_enfermero, tb_procedimiento.tipo_procedimiento, tb_paciente.nombre_paciente, tb_paciente.apellido_paciente
FROM tb_enfermero
INNER JOIN tb_medico ON tb_medico.id_medico = tb_enfermero.id_medico_enfermero
INNER JOIN medico_procedimiento ON medico_procedimiento.id_medico_procedimietno = tb_medico.id_medico
INNER JOIN tb_procedimiento ON tb_procedimiento.id_procedimiento = medico_procedimiento.id_procedimiento_medico
INNER JOIN tb_paciente ON tb_paciente.id_procedimiento_paciente = tb_procedimiento.id_procedimiento
```

nombre_enfermero	apellido_enfermero	tipo_procedimiento	nombre_paciente	apellido_paciente
Luisa	García	Extracción de sangre	María	García
Luisa	García	Endoscopia	Ana	Martínez
Gabriel	Rodríguez	Radiografía	José	Rodríguez
Ana	Fernández	Tomografía	Pedro	Fernández
Carlos	González	Cirugía de cataratas	Mónica	Hernández

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

Vista para ver el medicamento de cada paciente

Elegí esta vista porque para un hospital es importante tener a la mano saber que medicamento consume cada paciente.

```
-- Vista para ver el medicamento de cada paciente
CREATE VIEW medicamento_pacientes AS
SELECT tb_paciente.nombre_paciente AS Nombre, tb_paciente.apellido_paciente AS Apellido, tb_medicamento.nombre_emdicamento AS Medicamento, tb_medicamento.dosis AS Dosis
FROM paciente_medicamento
INNER JOIN tb_paciente ON tb_paciente.id_paciente = paciente_medicamento.id_paciente_medicamento
INNER JOIN tb_medicamento ON tb_medicamento.id_medicamento = paciente_medicamento.id_medicamento_paciente;

SELECT *
FROM medicamento_pacientes;
```

	Nombre	Apellido	Medicamento	Dosis
▶	María	García	Ibuprofeno	200mg
	José	Rodríguez	Paracetamol	500mg
	Ana	Martínez	Omeprazol	20mg
	Pedro	Fernández	Aspirina	100mg
	Mónica	Hernández	Amoxicilina	500mg

Vista para ver los procedimientos de cada paciente

También elegí esta vista porque para un hospital además de conocer el medicamento de sus pacientes, también es importante conocer el procedimiento que lleva cada paciente

```
289 •  CREATE VIEW procedimientos_pacientes AS
290    SELECT tb_paciente.nombre_paciente AS Nombre, tb_paciente.apellido_paciente AS Apellido, tb_procedimiento.tipo_procedimiento
291    FROM tb_paciente
292    INNER JOIN tb_procedimiento ON tb_procedimiento.id_procedimiento = tb_paciente.id_procedimiento_paciente;
```

	Nombre	Apellido	tipo_procedimiento
▶	María	García	Extracción de sangre
	José	Rodríguez	Radiografía
	Ana	Martínez	Endoscopia
	Pedro	Fernández	Tomografía
	Mónica	Hernández	Cirugía de cataratas

Vista para ver las facturas de cada cliente

Esta vista permite observar el nombre del cliente con el valor total a pagar en su factura, elegí esta vista porque para un hospital es importante observar el valor que debe pagar cada paciente.

```
297    -- Vista para ver las facturas de cada cliente
298 •  CREATE VIEW paciente_factura AS
299    SELECT  tb_paciente.nombre_paciente AS Nombre, tb_paciente.apellido_paciente AS Apellido, tb_factura.valor_total
300    FROM tb_paciente
301    INNER JOIN tb_factura ON tb_factura.id_paciente_factura = tb_paciente.id_paciente;
302
303 •  SELECT *
304    FROM paciente_factura;
```

	Nombre	Apellido	valor_total
▶	María	García	350000
	José	Rodríguez	450000
	Ana	Martínez	800000
	Pedro	Fernández	250000
	Mónica	Hernández	900000

Despúes de realizar el trabajo responda ¿Qué le agregaría al modelo para dar mas información y esa información cúal seria?

Agregaría una nueva tabla llamada Citas, con la información de la hora de la cita y fecha, para que el paciente deba sacar una cita antes de hacerle un procedimiento.

Tercera Actividad:

Elabore 4 procedimientos almacenados que me permitan agregar, actualizar, consultar y borrar, en una de las tablas de la librería (primera actividad).

Procedimiento para agregar un número de teléfono a un cliente

Este procedimiento tienen dos parámetros de entrada "cedula_cliente_procedimiento" y "numero_cliente", ambos de tipo VARCHAR.

El procedimiento empieza con la declaración BEGIN y finaliza con END y dentro se realiza un INSERT en la tabla teléfono_cliente para las columnas cedula_cliente" y "numero.

```
DELIMITER //
CREATE PROCEDURE agregar_telefono_cliente (
 IN cedula_cliente_procedimiento VARCHAR(10),
 IN numero_cliente VARCHAR(15)
)
BEGIN
INSERT INTO telefono_cliente (cedula_cliente, numero)
VALUES (cedula_cliente_procedimiento, numero_cliente);
END //
//DELIMITER ;

CALL agregar_telefono_cliente ('7070707070','+57 3605485751')
```

cedula_cliente	numero
4040404040	+57 3301234567
4040404040	+57 3304444444
5050505050	+57 3404567890
5050505050	+57 3405555555
6060606060	+57 3507890123
7070707070	+57 3601234567
7070707070	+57 3605485751
NULL	NULL

Procedimiento para actualizar un número de teléfono

Este procedimiento tienen tres parámetros de entrada "cedula_cliente_procedimiento", "numero_cliente" y "nuevo_numero_cliente", todos de tipo VARCHAR.

El procedimiento empieza con la declaración BEGIN y finaliza con END y dentro se realiza un UPDATE en la tabla teléfono_cliente Y un SET al campo numero para darle el nuevo valor. Luego se usa un WHERE para dar la condiciones para buscar el campo que se le va dar el nuevo valor

```
DELIMITER //
CREATE PROCEDURE actualizar_telefono_cliente (
IN cedula_cliente_procedimiento VARCHAR(10),
IN numero_cliente VARCHAR(15),
IN nuevo_numero_cliente VARCHAR(15)
)
BEGIN
UPDATE telefono_cliente
SET numero = nuevo_numero_cliente
WHERE cedula_cliente = cedula_cliente_procedimiento AND numero = numero_cliente;
END //
//DELIMITER ;

CALL actualizar_telefono_cliente ('7070707070','+57 3605485751','+57 3606060606')
```

cedula_cliente	numero
4040404040	+57 3301234567
4040404040	+57 3304444444
5050505050	+57 3404567890
5050505050	+57 3405555555
6060606060	+57 3507890123
7070707070	+57 3601234567
7070707070	+57 3606060606
NULL	NULL

Procedimiento para consultar número de teléfonos del cliente

Este procedimiento tienen un parámetros de entrada "cedula_cliente_procedimiento", de tipo VARCHAR.

El procedimiento empieza con la declaración BEGIN y finaliza con END y dentro se realiza un SELECT numero Y un FROM a la tabla teléfono_cliente. Luego se usa un WHERE para dar la condiciones para buscar el campo.

```
CREATE PROCEDURE consultar_telefono_cliente (
IN cedula_cliente_procedimiento VARCHAR(10)
)
BEGIN
SELECT numero
FROM telefono_cliente
WHERE cedula_cliente = cedula_cliente_procedimiento;
END //
//DELIMITER ;

CALL consultar_telefono_cliente ('7070707070')
```

numero
+57 3601234567
+57 3606060606

Procedimiento para eliminar numero de telefonos del cliente

Este procedimiento tienen dos parámetros de entrada "cedula_cliente_procedimiento", "numero_cliente" todos de tipo VARCHAR.

El procedimiento empieza con la declaración BEGIN y finaliza con END y dentro se realiza un DELETE desde teléfono_cliente Y Luego se usa un WHERE para dar la condiciones para buscar el campo que se desea eliminar.

```
48    DELIMITER //
49    CREATE PROCEDURE eliminar_telefono_cliente (
50    IN cedula_cliente_procedimiento VARCHAR(10),
51    IN numero_cliente VARCHAR(15)
52    )
53    BEGIN
54    DELETE FROM telefono_cliente
55    WHERE cedula_cliente = cedula_cliente_procedimiento AND numero = numero_cliente;
56    END //
57    //DELIMITER ;
```

Elabore una nueva tabla llamada "control_de_cambios_librería" la cual debe contener 3 columnas (usuario, accion, fecha) y guarde utilizando 2 Triggers el nombre del usuario que agrego o elimino un registro en la tabla seleccionada en el punto anterior.

Se crea una tabla con el nombre de control_de_cambios_librería, esta tabla contiene 3 columnas llamadas usuario, acción y fecha.

```
62    CREATE TABLE IF NOT EXISTS control_de_cambios_librería (
63        usuario VARCHAR(50),
64        accion VARCHAR(50),
65        fecha DATETIME DEFAULT current_timestamp
66    );
```

Se crea un trigger con el nombre agregar_teléfono que se dispara cuando inserten información en la tabla teléfono_cliente y dejara el registro en la tabla control_de_cambios_librería

```
69    DELIMITER //
70    CREATE TRIGGER agregar_telefono AFTER INSERT ON telefono_cliente
71        FOR EACH ROW
72        BEGIN
73            INSERT INTO control_de_cambios_librería VALUES(user(), 'agregar', now());
74        END;
75    // DELIMITER ;
```

Lo mismo pasa con el trigger con el nombre eliminar_telefono pero este se disprar cuando se elimine información de la tabla teléfono y también deja registro en la tabla control_de_cambios_librería.

```
78    DELIMITER //
79    CREATE TRIGGER eliminar_telefono AFTER DELETE ON telefono_cliente
80        FOR EACH ROW
81        BEGIN
82            INSERT INTO control_de_cambios_librería VALUES(user(), 'eliminar', now());
83        END;
84    // DELIMITER ;
```

Elabore 4 procedimientos almacenados que me permitan agregar, actualizar, consultar y borrar, en una de las tablas del Hospital (segunda actividad).

Procedimiento para agregar un número de teléfono a un medico

Este procedimiento tienen dos parámetros de entrada " id_medico_procedimiento" y "numero_medico_procedimiento", ambos de tipo VARCHAR.

El procedimiento empieza con la declaración BEGIN y finaliza con END y dentro se realiza un INSERT en la tabla telefono_medico para las columnas id_medico_telefono" y numero_medico.

```
3     -- Procedimiento para agregar un numero de telefono
4     DELIMITER //
5     CREATE PROCEDURE agregar_telefono_medico (
6      IN id_medico_procedimiento VARCHAR(15),
7      IN numero_medico_procedimiento VARCHAR(15)
8     )
9     BEGIN
10    INSERT INTO telefono_medico (id_medico_telefono, numero_medico)
11    VALUES (id_medico_procedimiento, numero_medico_procedimiento);
12    END //
13    //DELIMITER ;
14
15    CALL agregar_telefono_medico('M001','3150254875')
```

id_medico_telefono	numero_medico
M001	3101234567
M001	3150254875
M002	3219876543
M003	3014567890
M004	3003216549
M005	3127418529
NULL	NULL

Procedimiento para actualizar un número de teléfono

Este procedimiento tienen tres parámetros de entrada " id_medico_procedimiento", "numero_medico_procedimiento" y " nuevo_numero_medico", todos de tipo VARCHAR.

El procedimiento empieza con la declaración BEGIN y finaliza con END y dentro se realiza un UPDATE en la tabla telefono_medico Y un SET al campo numero_medico para darle el nuevo valor. Luego se usa un WHERE para dar la condiciones para buscar el campo que se le va dar el nuevo valor.

```sql
-- Procedimiento para actualizar un numero de telefono
DELIMITER //
CREATE PROCEDURE actualizar_telefono_medico (
IN id_medico_procedimiento VARCHAR(15),
IN numero_medico_procedimiento VARCHAR(15),
IN nuevo_numero_medico VARCHAR(15)
)
BEGIN
UPDATE telefono_medico
SET numero_medico = nuevo_numero_medico
WHERE id_medico_telefono = id_medico_procedimiento AND numero_medico = numero_medico_procedimiento;
END //
//DELIMITER ;

CALL actualizar_telefono_medico ('M001','3150254875','3208565873')
```

id_medico_telefono	numero_medico
M001	3101234567
M001	3208565873
M002	3219876543
M003	3014567890
M004	3003216549
M005	3127418529
NULL	NULL

Procedimiento para consultar número de teléfonos

Este procedimiento tienen un parámetros de entrada " id_medico_procedimiento", de tipo VARCHAR.

El procedimiento empieza con la declaración BEGIN y finaliza con END y dentro se realiza un SELECT numero_medico Y un FROM a la tabla telefono_medico. Luego se usa un WHERE para dar la condiciones para buscar el campo.

```sql
-- Procedimiento para consultar numero de telefonos
DELIMITER //
CREATE PROCEDURE consultar_telefono_medico (
IN id_medico_procedimiento VARCHAR(15)
)
BEGIN
SELECT numero_medico
FROM telefono_medico
WHERE id_medico_telefono = id_medico_procedimiento;
END //
//DELIMITER ;

CALL consultar_telefono_medico('M001')
```

Procedimiento para eliminar numero de teléfonos del cliente

Este procedimiento tiene dos parámetros de entrada " id_medico_procedimiento", " numero_medico_procedimiento" todos de tipo VARCHAR.

El procedimiento empieza con la declaración BEGIN y finaliza con END y dentro se realiza un DELETE desde telefono_medico Y Luego se usa un WHERE para dar la condiciones para buscar el campo que se desea eliminar.

```
48    DELIMITER //
49    CREATE PROCEDURE eliminar_telefono_medico (
50    IN id_medico_procedimiento VARCHAR(15),
51    IN numero_medico_procedimiento VARCHAR(15)
52    )
53    BEGIN
54    DELETE FROM telefono_medico
55    WHERE id_medico_telefono = id_medico_procedimiento AND numero_medico = numero_medico_procedimiento;
56    END //
57    //DELIMITER ;
```

Elabore una nueva tabla llamada "control_de_cambios_hospital" la cual debe contener 3 columnas (usuario, accion, fecha) y guarde utilizando 2 Triggers el nombre del usuario que agrego o elimino un registro en la tabla seleccionada en el punto anterior.

Se crea una tabla con el nombre de control_de_cambios_hospital, esta tabla contiene 3 columnas llamadas usuario, acción y fecha.

```
62  CREATE TABLE IF NOT EXISTS control_de_cambios_hospital (
63      usuario VARCHAR(50),
64      accion VARCHAR(50),
65      fecha DATETIME DEFAULT current_timestamp
66  );
```

Se crea un trigger con el nombre agregar_teléfono que se dispara cuando inserten información en la tabla teléfono_cliente y dejara el registro en la tabla control_de_cambios_hospital

```
69    DELIMITER //
70    CREATE TRIGGER agregar_telefono AFTER INSERT ON telefono_medico
71        FOR EACH ROW
72        BEGIN
73            INSERT INTO control_de_cambios_hospital VALUES(user(), 'agregar', now());
74        END;
75    // DELIMITER ;
```

Lo mismo pasa con el trigger con el nombre eliminar_telefono pero este se disprar cuando se elimine información de la tabla teléfono y también deja registro en la tabla control_de_cambios_hospital.

```
78    DELIMITER //
79    CREATE TRIGGER eliminar_telefono AFTER DELETE ON telefono_medico
80        FOR EACH ROW
81        BEGIN
82            INSERT INTO control_de_cambios_hospital VALUES(user(), 'eliminar', now());
83        END;
84    // DELIMITER ;
```