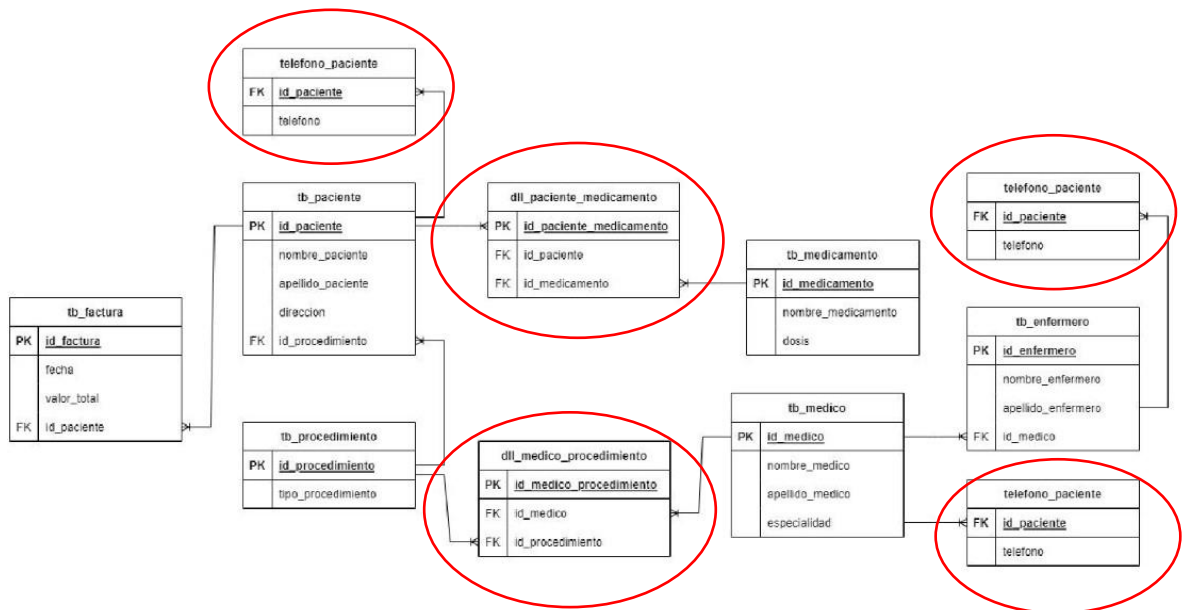


DOCUMENTACION TALLER #5 ACTIVIDAD 2:

Para empezar a realizar esta actividad sobre el Hospital de la mejor manera, lo primero que hice fue revisar y validar el modelo relacional que nos proporcionó el maestro Juan en el repositorio; entonces, empecé por observar que las relaciones estuviesen bien, que las tablas tanto principales como intermedias estuviesen bien construidas y con las llaves correspondientes y que finalmente, los atributos, según fueran su tipo, tuviesen o no las tablas correspondientes, pero no fue así, no del todo estaba bien y dentro de los errores que pude evidenciar en el modelo están estos:

1. Las tablas que se generan intermedias a causa de una cardinalidad N:M, son tablas que heredan las llaves primarias de sus entidades correspondientes pero en dicha tabla quedarán como foráneas, por decirlo de alguna forma creativa, porque están visitando esa tabla. Entonces en aquel diagrama, las tablas `dII_paciente_medicamento` y `dII_medico_procedimiento` fallaron con esto.
2. Cuando tenemos en un modelo entidad relación, un atributo que resulta ser compuesto, cuando normalizamos para hacer un buen modelo relacional, lo que hacemos es crearle a este una nueva tabla, y en ella se hereda la llave primaria de su entidad madre y adicionalmente queda con otra llave primaria correspondiente a la tabla, por ejemplo número o teléfono pero ojo, no se puede olvidar que aquella llave que se hereda de la entidad madre también será foránea, es decir, finalmente será compuesta. Entonces, este fue otro error, porque aquellas tablas en su construcción no eran correctas.
3. Y finalmente, un error un poco de concentración, diría yo, poner el nombre correspondiente a las tablas que estamos creando teniendo en cuenta la entidad de las que estamos derivando.

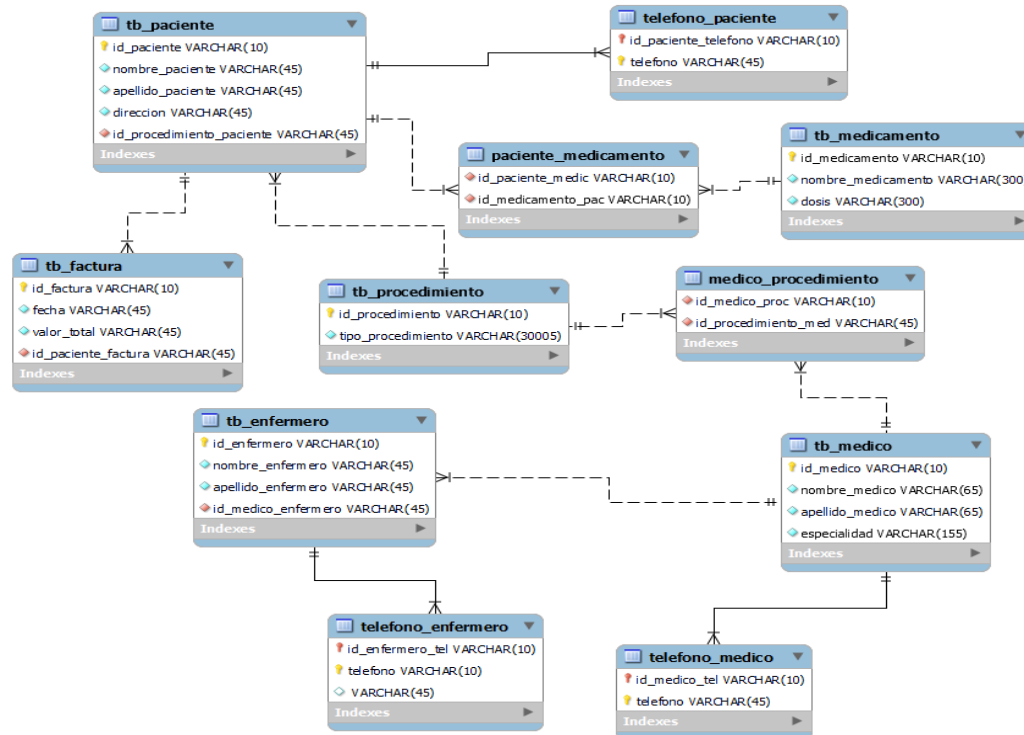
Este es el modelo base dado:



Y, tras hacer las correcciones pertinentes frente a los errores que encontré, este fue el modelo relacional que cree en el programa Workbench:

Pero antes recordar un poco, como hacerlo:

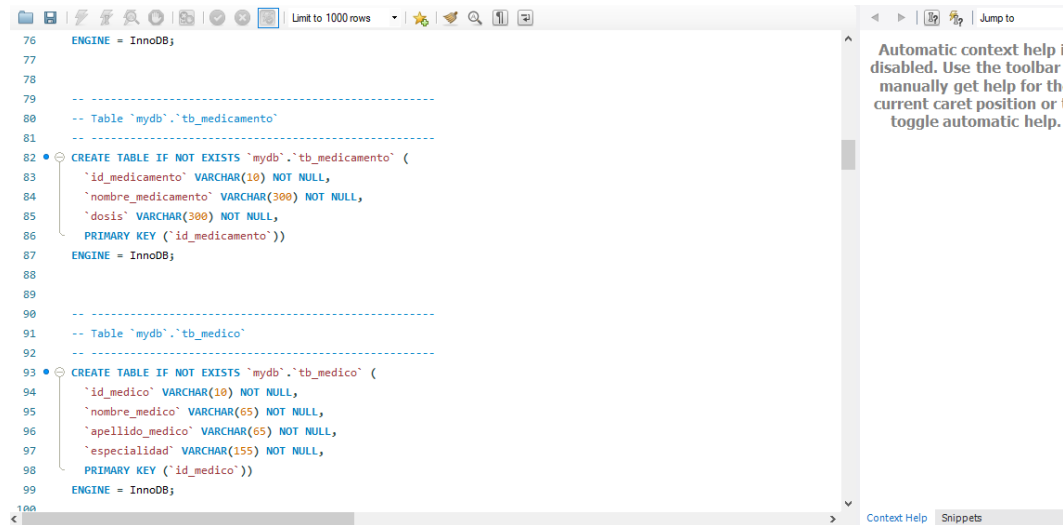
- ★ Identificar las entidades de nuestro MER para transformarlas en tablas
- ★ Identificar los atributos y las relaciones
- ★ Empezar por crear aquellas tablas correspondientes a las entidades con sus respectivas llaves y como recomendación empezar primero por las tablas principales, es decir las madres
- ★ Tras esto, ir creando las demás tablas intermedias, sean por relaciones N:M o por tener atributos compuestos
- ★ Luego, empezar a relacionar dichas tablas ya creadas con las demás , ¿de que forma? Haciendo uso de las llaves primarias y foráneas, conectándolas así y validando las relaciones y cardinalidades correctas
- ★ Finalmente , revisar las veces que desee su modelo final MR para que coincida con lo anteriormente hecho en el MER.



Con el modelo relacional hecho correctamente, hice uso de la ingeniería hacia adelante para generar el Script correspondiente a la base de datos. Aquí adjunto algunas imágenes:

```

1  -- MySQL Workbench Forward Engineering
2
3  • SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4  • SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5  • SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
6
7  --
8  -- Schema mydb
9  --
10
11 --
12 -- Schema mydb
13 --
14 • CREATE SCHEMA IF NOT EXISTS 'mydb' DEFAULT CHARACTER SET utf8 ;
15 • USE 'mydb' ;
16
17 --
18 -- Table 'mydb`.`tb_procedimiento`
19 --
20 • CREATE TABLE IF NOT EXISTS 'mydb`.`tb_procedimiento' (
21   'id_procedimiento' VARCHAR(10) NOT NULL,
22   'tipo_procedimiento' VARCHAR(300) NOT NULL,
23   PRIMARY KEY ('id_procedimiento'))
24 ENGINE = InnoDB;
25
  
```



```
76 ENGINE = InnoDB;
77
78
79 -----
80 -- Table 'mydb`.`tb_medimento`
81 -----
82 CREATE TABLE IF NOT EXISTS `mydb`.`tb_medimento` (
83   `id_medimento` VARCHAR(10) NOT NULL,
84   `nombre_medimento` VARCHAR(300) NOT NULL,
85   `dosis` VARCHAR(300) NOT NULL,
86   PRIMARY KEY (`id_medimento`))
87 ENGINE = InnoDB;
88
89
90 -----
91 -- Table 'mydb`.`tb_medico`
92 -----
93 CREATE TABLE IF NOT EXISTS `mydb`.`tb_medico` (
94   `id_medico` VARCHAR(10) NOT NULL,
95   `nombre_medico` VARCHAR(65) NOT NULL,
96   `apellido_medico` VARCHAR(65) NOT NULL,
97   `especialidad` VARCHAR(155) NOT NULL,
98   PRIMARY KEY (`id_medico`))
99 ENGINE = InnoDB;
```

Luego de tener listo el Script , inserté los registros en mi base de datos, de la siguiente manera, haciendo uso de las sentencias INSERT TO, y VALUES:

INSERT INTO tb_procedimiento (id_procedimiento, tipo_procedimiento)

VALUES ('P1', 'Extracción cordales'),

('P2', 'Limpieza dental'),

('P3', 'Endodoncia'),

('P4', 'Implante dental'),

('P5', 'Ortodoncia');

Hasta ahí , todo bien, pero al tratar de ingresar los datos tuve un error , porque ya estaban duplicados ,es decir que ya , por alguna razón, había registrado esos datos en la tabla , es decir que ya existían y estaba tratando de decirle a MySQL que los volviera a ingresar, entonces para corregir esto usé esta sentencia que lo que quiere decir es que me borra de (nombre de la tabla) donde se cumpla una condición , en mi caso borrar los registros que tenían esos id, pero aclaro que lo hice para dejarla nuevamente vacía y luego volverla a llenar correctamente.

DELETE FROM tb_procedimiento WHERE id_procedimiento = 'PROC1';

Imagen correspondiente al insert de los registros para la tabla procedimiento:

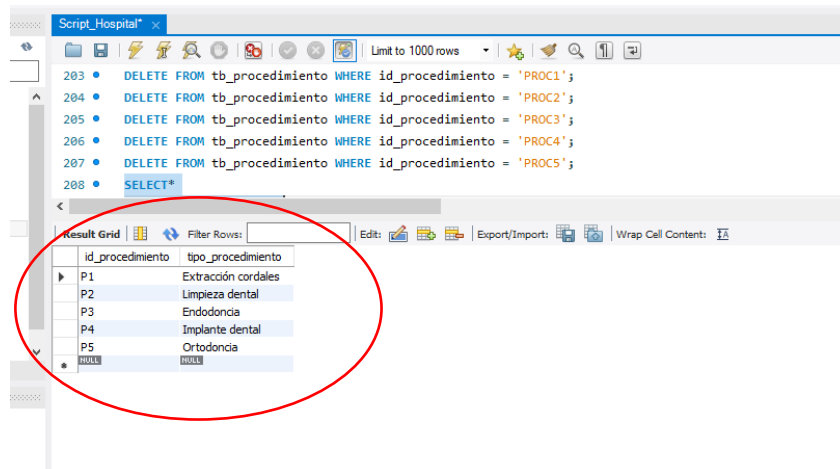


Imagen correspondiente al insert de los registros para la tabla paciente:

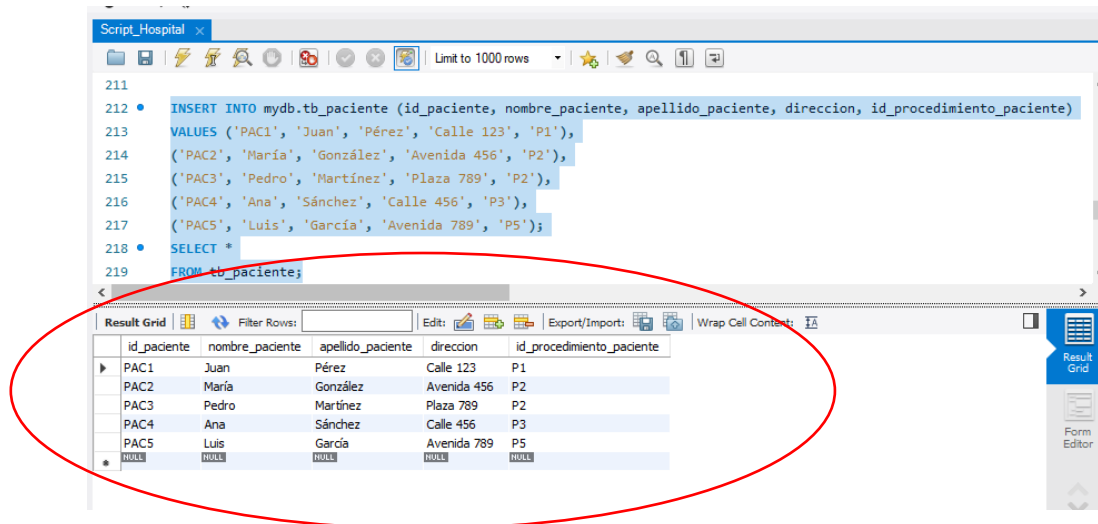


Imagen correspondiente al insert de los registros para la tabla telefono_paciente, teniendo en cuenta que debía registrar un telefono con un id de paciente ya existente, sino aparecería un error:

Script_Hospital

```

218 • SELECT *
219 FROM tb_paciente;
220
221 • INSERT INTO mydb.telefono_paciente (id_paciente_telefono, telefono)
222 VALUES ('PAC1', '3156203563'),
223 ('PAC1', '604274896'),
224 ('PAC2', '3112569985'),
225 ('PAC3', '3217884789'),
226 ('PAC5', '3214567891'),

```

Result Grid

id_paciente_telefono	telefono
PAC1	3156203563
PAC1	604274896
PAC2	3112569985
PAC3	3217884789
PAC4	3005226630
PAC5	3214567891
NULL	NULL

telefono_paciente7

Imagen correspondiente al insert de los registros para la tabla factura, teniendo en cuenta la llave foránea correspondiente al id del paciente:

```

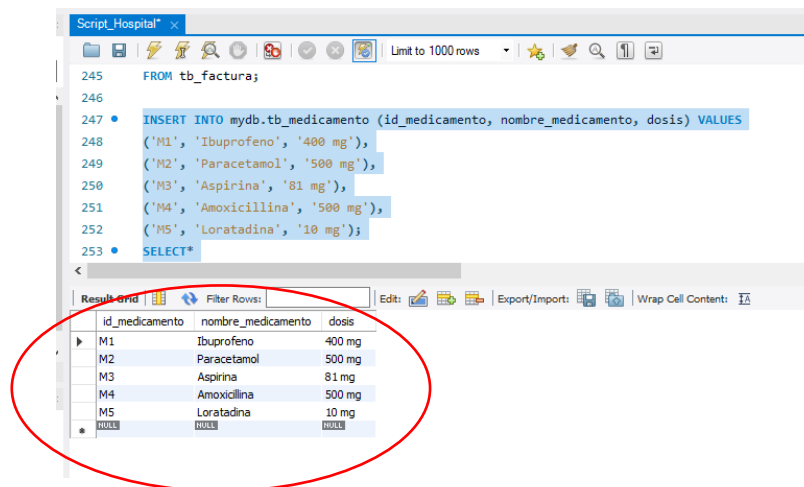
237
238 • INSERT INTO mydb.tb_factura (id_factura, fecha, valor_total, id_paciente_factura)
239 VALUES ('FAC1', '2022-01-01', '100', 'PAC1'),
240 ('FAC2', '2022-02-01', '200', 'PAC2'),
241 ('FAC3', '2022-03-01', '300', 'PAC3'),
242 ('FAC4', '2022-04-01', '400', 'PAC4'),
243 ('FAC5', '2022-05-01', '500', 'PAC5');
244 • SELECT *

```

Result Grid

id_factura	fecha	valor_total	id_paciente_factura
FAC1	2022-01-01	100	PAC1
FAC2	2022-02-01	200	PAC2
FAC3	2022-03-01	300	PAC3
FAC4	2022-04-01	400	PAC4
FAC5	2022-05-01	500	PAC5
NULL	NULL	NULL	NULL

Imagen correspondiente al insert de los registros para la tabla medicamento:



```
245 FROM tb_factura;
```

```
246
```

```
247 • INSERT INTO mydb.tb_medicamento (id_medicamento, nombre_medicamento, dosis) VALUES
```

```
248 ('M1', 'Ibuprofeno', '400 mg'),
```

```
249 ('M2', 'Paracetamol', '500 mg'),
```

```
250 ('M3', 'Aspirina', '81 mg'),
```

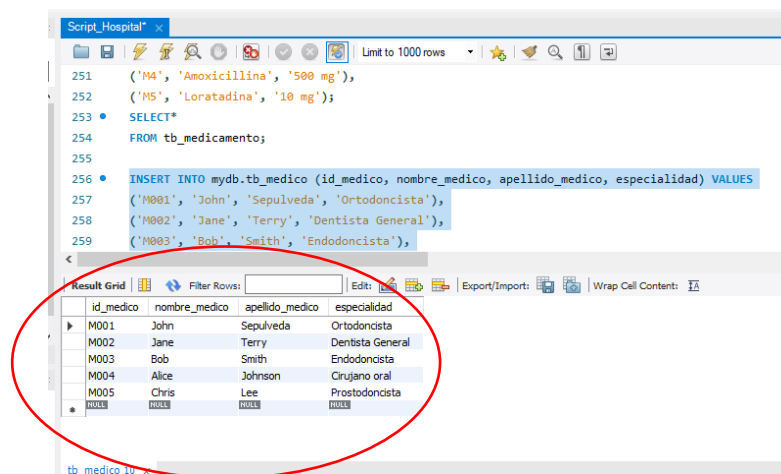
```
251 ('M4', 'Amoxicilina', '500 mg'),
```

```
252 ('M5', 'Loratadina', '10 mg');
```

```
253 • SELECT*
```

id_medicamento	nombre_medicamento	dosis
M1	Ibuprofeno	400 mg
M2	Paracetamol	500 mg
M3	Aspirina	81 mg
M4	Amoxicilina	500 mg
M5	Loratadina	10 mg

Imagen correspondiente al insert de los registros para la tabla medico:



```
251 ('M4', 'Amoxicilina', '500 mg'),
```

```
252 ('M5', 'Loratadina', '10 mg');
```

```
253 • SELECT*
```

```
254 FROM tb_medicamento;
```

```
255
```

```
256 • INSERT INTO mydb.tb_medico (id_medico, nombre_medico, apellido_medico, especialidad) VALUES
```

```
257 ('M001', 'John', 'Sepulveda', 'Ortodoncista'),
```

```
258 ('M002', 'Jane', 'Terry', 'Dentista General'),
```

```
259 ('M003', 'Bob', 'Smith', 'Endodoncista'),
```

id_medico	nombre_medico	apellido_medico	especialidad
M001	John	Sepulveda	Ortodoncista
M002	Jane	Terry	Dentista General
M003	Bob	Smith	Endodoncista
M004	Alice	Johnson	Cirujano oral
M005	Chris	Lee	Prostodoncista

Imagen correspondiente al insert de los registros para la tabla enfermero:

```
263 FROM tb_medico;
264
265 • INSERT INTO mydb.tb_enfermero (id_enfermero, nombre_enfermero, apellido_enfermero, id_medico_enfermero)
266 VALUES
267 ('E001', 'Ana', 'Garcia', 'M001'),
268 ('E002', 'Pedro', 'Lopez', 'M002'),
```

id_enfermero	nombre_enfermero	apellido_enfermero	id_medico_enfermero
E001	Ana	Garcia	M001
E002	Pedro	Lopez	M002
E003	Lucia	Fernandez	M003
E004	Juan	Perez	M001
E005	Luisa	Gonzalez	M002

Imagen correspondiente al insert de los registros para la tabla paciente_medico, como resultado de una relación N:M entre paciente y medico:

```
275 • INSERT INTO mydb.paciente_medicamento (id_paciente_medic, id_medimento_pac) VALUES
276 ('PAC1', 'M2'),
277 ('PAC2', 'M3'),
```

id_paciente_medic	id_medimento_pac
PAC1	M2
PAC2	M3
PAC3	M1
PAC4	M5
PAC5	M4

Imagen correspondiente al insert de los registros para la tabla medico_procedimiento, como un resultado de la relación N:M entre ambas entidades:

```
281 • SELECT*
282 FROM paciente_medimento;
283
284 • INSERT INTO mydb.medico_procedimiento (id_medico_proc, id_procedimiento_med) VALUES
285 ('M001', 'P1'),
286 ('M002', 'P2'),
```

id_medico_proc	id_procedimiento_med
M001	P1
M002	P2
M003	P5
M004	P3
M005	P4

Imagen correspondiente al insert de los registros para la tabla telefono_enfermero, como el resultado de ser un atributo compuesto:

293 • INSERT INTO mydb.telefono_enfermero (id_enfermero_tel, telefono) VALUES

id_enfermero_tel	telefono
E001	3112758962
E002	6042747915
E003	3225447788
E004	3103589620
E005	5558960
M001	M001

Imagen correspondiente al insert de los registros para la tabla telefono_medico, como el resultado de ser un atributo compuesto:

Script_Hospital

```
306 ('M005', '3225441199');
307 ('M004', '6045263320');
308
309 DELETE FROM telefono_medico WHERE id_medico_tel = 'M001';
310 DELETE FROM telefono_medico WHERE id_medico_tel = 'M002';
311 DELETE FROM telefono_medico WHERE id_medico_tel = 'M003';
312 DELETE FROM telefono_medico WHERE id_medico_tel = 'M004';
313 DELETE FROM telefono_medico WHERE id_medico_tel = 'M005';
314 SELECT
```

id_medico_tel	telefono
M001	3154123620
M002	3207467796
M003	3103562111
M004	6045263320
M005	3225441199
M001	M001

Luego de registrar la información en mi base de datos, procedí a realizar las consultas indicadas:

✓ **Consulta para saber que medicamentos ha tomado cada paciente y la dosis suministrada:**

Para comenzar tuve en cuenta que debía utilizar tres tablas de mi base de datos, tb_paciente, tb_medicamento y paciente_medicamento, y así mismo que debía usar la sentencia JOIN para unir las tres tablas a través de sus claves primarias y foráneas.

En particular, la tabla paciente_medicamento tiene dos llaves foráneas id_paciente_medic e id_medicamento_pac, que se corresponden con las claves primarias id_paciente e id_medicamento, respectivamente, en las tablas

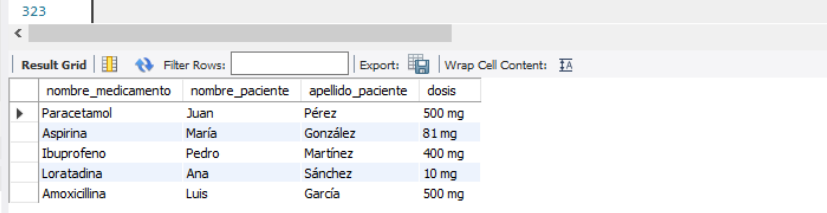
tb_paciente y tb_medimento. Entonces, para hacer mi consulta seleccioné los campos nombre_paciente, apellido_paciente, nombre_medimento y dosis de las tablas unidas, para que el resultado final fuera una tabla que muestra qué medicamentos ha tomado cada paciente y la dosis correspondiente.

Estas fueron las sentencias que usé para la consulta:

```
SELECT nombre_medimento, nombre_paciente,apellido_paciente,dosis
FROM tb_paciente
INNER JOIN paciente_medimento ON id_paciente=id_paciente_medic
INNER JOIN tb_medimento ON id_medimento= id_medimento_pac;
```

Esta es el resultado:

```
317 -- Consulta para saber que medicamentos ha tomado cada paciente y la dosis suministrada
318
319 • SELECT nombre_medimento, nombre_paciente,apellido_paciente,dosis
320 FROM tb_paciente
321 INNER JOIN paciente_medimento ON id_paciente=id_paciente_medic
322 INNER JOIN tb_medimento ON id_medimento= id_medimento_pac;
323
```



nombre_medimento	nombre_paciente	apellido_paciente	dosis
Paracetamol	Juan	Pérez	500 mg
Aspirina	María	González	81 mg
Ibuprofeno	Pedro	Martínez	400 mg
Loratadina	Ana	Sánchez	10 mg
Amoxicilina	Luis	García	500 mg

- ✓ Consulta para saber que enfermeros, y que médicos estuvieron en los procedimientos de los pacientes:

Para empezar a hacer la consulta , empecé por seleccionar los nombres y apellidos de los enfermeros, los nombres y apellidos de los médicos, la especialidad de los médicos y el tipo de procedimiento realizado en una tabla de procedimientos. Luego, usé la sentencia INNER JOIN para combinar las tablas tb_enfermero, tb_medico y tb_procedimiento mediante sus respectivas claves primarias y foráneas, y medico_procedimiento para relacionar las tablas tb_medico y tb_procedimiento.

Así fue como quedó la consulta, que realmente era solo conocer que enfermeros estuvieron en los procedimientos de los pacientes, pero que para aprender más decidí modificar un poco, aunque lo pedido se cumple:

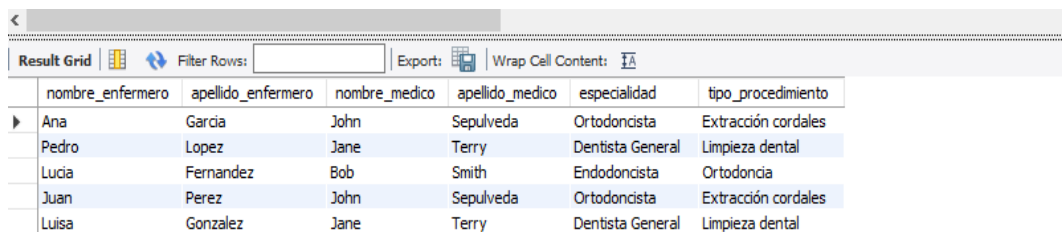
```
SELECT tb_enfermero.nombre_enfermero, tb_enfermero.apellido_enfermero,
tb_medico.nombre_medico, tb_medico.apellido_medico, tb_medico.especialidad,
tb_procedimiento.tipo_procedimiento

FROM tb_enfermero

INNER JOIN tb_medico ON tb_enfermero.id_medico_enfermero = tb_medico.id_medico

INNER JOIN medico_procedimiento ON tb_medico.id_medico =
medico_procedimiento.id_medico_proc

INNER JOIN tb_procedimiento ON medico_procedimiento.id_procedimiento_med =
tb_procedimiento.id_procedimiento;
```



The screenshot shows a database query result grid with the following data:

	nombre_enfermero	apellido_enfermero	nombre_medico	apellido_medico	especialidad	tipo_procedimiento
▶	Ana	Garcia	John	Sepulveda	Ortodoncista	Extracción cordales
	Pedro	Lopez	Jane	Terry	Dentista General	Limpieza dental
	Lucia	Fernandez	Bob	Smith	Endodoncista	Ortodoncia
	Juan	Perez	John	Sepulveda	Ortodoncista	Extracción cordales
	Luisa	Gonzalez	Jane	Terry	Dentista General	Limpieza dental

Finalmente, procedí a crear las vistas:

-- creación de una vista para saber que medicamentos debe tomar cada paciente y que dosis--

La vista que planteé incluirá los siguientes campos para cada paciente y medicamento:

- ★ nombre_medimento: el nombre del medicamento que el paciente está tomando.
- ★ nombre_paciente: el nombre del paciente.
- ★ apellido_paciente: el apellido del paciente.

dosis: la dosis del medicamento que el paciente está tomando.

Para obtener esta información, creé la vista que realiza una consulta utilizando tres tablas: "tb_paciente", "paciente_medimento" y "tb_medimento". Luego, utilicé una cláusula INNER JOIN para unir estas tablas en función de sus claves primarias y foráneas.

El primer INNER JOIN que decidí usar, une las tablas "tb_paciente" y "paciente_medicamento" utilizando las columnas "id_paciente" de ambas tablas. Esto me permitió relacionar a cada paciente con los medicamentos que está tomando.

Para la segunda INNER JOIN uní las tablas "paciente_medicamento" y "tb_medicamento" utilizando las columnas "id_medicamento_pac" de ambas tablas. Esto me permite relacionar cada medicamento con su información correspondiente en la tabla "tb_medicamento".

Y finalmente esta fue la vista que creé, que a decir verdad me pareció muy importante hacerla porque hablando en un contexto hospitalario, mas enfocado a la vida real, poder tener estos registros facilita muchas cosas y lleva un buen control tanto para el sistema como para los pacientes.

```
CREATE VIEW info_medicamento_pacientes AS
```

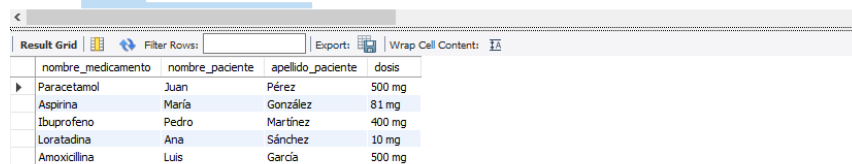
```
SELECT nombre_medicamento, nombre_paciente, apellido_paciente, dosis
```

```
FROM tb_paciente
```

```
INNER JOIN paciente_medicamento ON id_paciente=id_paciente_medic
```

```
INNER JOIN tb_medicamento ON id_medicamento= id_medicamento_pac;
```

```
330
331 -- creacion de una vista para saber que medicamentos debe tomar cada paciente y que dosis
332 • CREATE VIEW info_medicamento_pacientes AS
333 SELECT nombre_medicamento, nombre_paciente, apellido_paciente, dosis
334 FROM tb_paciente
335 INNER JOIN paciente_medicamento ON id_paciente=id_paciente_medic
336 INNER JOIN tb_medicamento ON id_medicamento= id_medicamento_pac;
337 • SELECT *
```



The screenshot shows a database query result grid with the following data:

nombre_medicamento	nombre_paciente	apellido_paciente	dosis
Paracetamol	Juan	Pérez	500 mg
Aspirina	María	González	81 mg
Ibuprofeno	Pedro	Martínez	400 mg
Loratadina	Ana	Sánchez	10 mg
Amoxicilina	Luis	García	500 mg

-- Creación de una vista para tener registro e información completa de los procedimientos en los pacientes—

Para comenzar con la creación de esta vista, lo primero que hice fue tener en cuenta los campos que incluiría :

- ★ nombre_enfermero: el nombre del enfermero que participó en el procedimiento.
- ★ apellido_enfermero: el apellido del enfermero que participó en el procedimiento.
- ★ nombre_medico: el nombre del médico que realizó el procedimiento.
- ★ apellido_medico: el apellido del médico que realizó el procedimiento.
- ★ especialidad: la especialidad del médico que realizó el procedimiento.
- ★ tipo_procedimiento: el tipo de procedimiento realizado.

Y para hacer esta vista un hecho, empecé por utilizar tres tablas: "tb_enfermero", "tb_medico" y "medico_procedimiento", luego usé una cláusula INNER JOIN para unir estas tablas en función de sus claves primarias y foráneas.

El primer INNER JOIN me unió las tablas "tb_enfermero" y "tb_medico" utilizando la columna "id_medico_enfermero" de la tabla "tb_enfermero" y la columna "id_medico" de la tabla "tb_medico". Esto con el fin de poder relacionar a cada enfermero con el médico que supervisó el procedimiento.

El segundo INNER JOIN unió las tablas "tb_medico" y "medico_procedimiento" utilizando la columna "id_medico" de ambas tablas, para así relacionar a cada médico con los procedimientos que ha realizado.

Y una vez que se han unido las tres tablas, la vista selecciona los campos necesarios para crear la vista "info_procedimientos".

Esta fue la vista creada, que me pareció muy importante porque, retomando el contexto hospitalario, llevar un buen registro de los procedimientos que se le hacen a los pacientes y quienes (enfermero y medico) lo hacen, es un indicador de un buen orden y aunque sean solo registros en este caso, en la vida real podría ser sinónimo de organización, de calidad hacia los clientes:

```
CREATE VIEW info_procedimientos AS
```

```
SELECT tb_enfermero.nombre_enfermero, tb_enfermero.apellido_enfermero,  
tb_medico.nombre_medico, tb_medico.apellido_medico, tb_medico.especialidad,  
tb_procedimiento.tipo_procedimiento
```

```
FROM tb_enfermero
```

```
INNER JOIN tb_medico ON tb_enfermero.id_medico_enfermero = tb_medico.id_medico
```

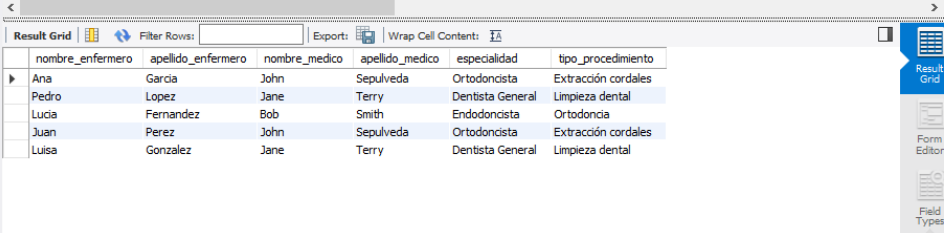
```
INNER JOIN medico_procedimiento ON tb_medico.id_medico =  
medico_procedimiento.id_medico_proc
```

```
INNER JOIN tb_procedimiento ON medico_procedimiento.id_procedimiento_med =  
tb_procedimiento.id_procedimiento;
```

```

339
340 -- Creacion de una vista para tener resgistro e informacion completa de los procedimientos en los pacientes
341 • CREATE VIEW info_procedimientos AS
342 SELECT tb_enfermero.nombre_enfermero, tb_enfermero.apellido_enfermero, tb_medico.nombre_medico, tb_medico.apellido_med.
343 FROM tb_enfermero
344 INNER JOIN tb_medico ON tb_enfermero.id_medico_enfermero = tb_medico.id_medico

```



nombre_enfermero	apellido_enfermero	nombre_medico	apellido_medico	especialidad	tipo_procedimiento
Ana	García	John	Sepulveda	Ortodoncista	Extracción cordales
Pedro	Lopez	Jane	Terry	Dentista General	Limpieza dental
Lucia	Fernandez	Bob	Smith	Endodoncista	Ortodoncia
Juan	Perez	John	Sepulveda	Ortodoncista	Extracción cordales
Luisa	Gonzalez	Jane	Terry	Dentista General	Limpieza dental

-- Creacion de una vista para mostrar la información del paciente, la factura correspondiente y el total de la factura—

Lo primero que hice fue elegir los campos para la creacion de mi vista:

- ★ nombre_paciente: el nombre del paciente.
- ★ apellido_paciente: el apellido del paciente.
- ★ direccion: la dirección del paciente.
- ★ fecha: la fecha en que se generó la factura correspondiente.
- ★ valor_total: el valor total de la factura.

Luego de tener claro, que columnas usar, utilicé dos tablas: "tb_paciente" y "tb_factura" y luego el JOIN para unir estas tablas en función de sus claves primarias y foráneas: la columna "id_paciente" de la tabla "tb_paciente" y la columna "Id_paciente_factura" de la tabla "tb_factura". Esto me permitió relacionar a cada paciente con sus facturas correspondientes.

Y antes de las sentencias, cree esta vista porque me parece que es bastante importante en un sistema hospitalario, pues llevar un registro organizado de los pacientes , sus facturas y su valor pagado , para la vida real , la practica resulta muy beneficioso y en el sentido de nuestro curso, porque además es un registro que contiene muchos datos y construyéndolos así aprendemos mucho más.

Estas fueron las sentencias para la creacion de la vista:

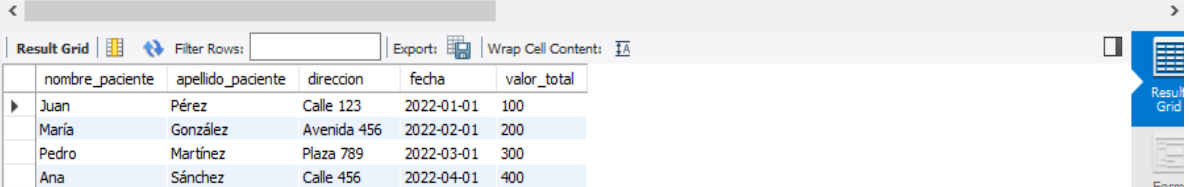
```
CREATE VIEW info_paciente AS
```

```
SELECT p.nombre_paciente, p.apellido_paciente, p.direccion, f.fecha, f.valor_total
```

```
FROM `mydb`.`tb_paciente` p
```

```
JOIN `mydb`.`tb_factura` f ON p.id_paciente = f.Id_paciente_factura;
```

```
351 -- Creacion de una vista para mostrar la información del paciente, la factura correspondiente y el total de la factura
352
353 • CREATE VIEW info_paciente AS
354 SELECT p.nombre_paciente, p.apellido_paciente, p.direccion, f.fecha, f.valor_total
355 FROM `mydb`.`tb_paciente` p
356 JOIN `mydb`.`tb_factura` f ON p.id_paciente = f.Id_paciente_factura;
357 • SELECT *
```



nombre_paciente	apellido_paciente	direccion	fecha	valor_total
Juan	Pérez	Calle 123	2022-01-01	100
María	González	Avenida 456	2022-02-01	200
Pedro	Martínez	Plaza 789	2022-03-01	300
Ana	Sánchez	Calle 456	2022-04-01	400

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

Podríamos agregarle a nuestra base de datos algunas otras tablas de importancia como por ejemplo: agendar cita y que el paciente elija con que doctor le gustaría y en que horario, también se podría agregar el correo tanto del paciente como del enfermero y del doctor para que los medios de comunicación sean mas accesibles , incluso podríamos también crear una tabla para el servicio al cliente, con los nombres de los asesores , sus teléfonos y horarios de trabajo. Y finalmente, podríamos agregar como atributos para que la información sea más completa , duración de los procedimientos , recomendaciones para antes y después del procedimiento, contactos de emergencia del paciente y en medicamentos , por cuanto tiempo tomarlos y sus efectos para tenerlos en cuenta.