**INSTITUTO TECNOLÓGICO DE CHIHUAHUA II**

**UNIDAD 2**

**QUINTO SEMESTRE**

Ingeniería en Sistemas Computacionales

Taller de bases de datos

Examen práctico

Ángel Eduardo Arizpe Carrillo 22550298

Armando Sebastián Bueno Núñez 22550336

Miguel Ángel Caballero Peña 22550358

Luis Carlos Meza López 22550299

Maestro: Rubén Alonso Hernández Chávez

Chihuahua, Chih. 9 de noviembre de 2024

**Creación de tablas:**

Tabla pacientes:

create table pacientes(paciente\_id int key auto\_increment, nombre varchar(50) not null, apellido\_pat varchar(50) not null, apellido\_mat varchar(50) not null, fecha\_nacimiento date not null, peso decimal(4,2) not null, altura decimal(4,2) not null, numero\_seguro\_social varchar(11) not null, genero enum("masculino", "femenino"), telefono\_casa varchar(10), telefono\_trabajo varchar(10), telefono\_movil varchar(10), otros varchar(50));

Tabla cuartos:

create table cuartos(cuarto\_id int key auto\_increment, paciente\_id int, numero\_cuarto int not null, inicio\_estadia date not null, fin\_estadia date, foreign key (paciente\_id) references pacientes(paciente\_id) on delete restrict on update cascade);

Tabla direcciones\_pacientes:

create table direcciones\_pacientes(direccion\_paciente\_id int key auto\_increment, paciente\_id int not null, direccion\_id int not null, foreign key (paciente\_id) references pacientes(paciente\_id) on delete restrict on update cascade, foreign key (direccion\_id) references direcciones(direccion\_id) on delete cascade on update cascade);

Tabla empleados:

create table empleados(empleados\_id int key auto\_increment, nombre varchar(50) not null, apellido\_pat varchar(50) not null, apellido\_mat varchar(50) not null, titulo varchar(50), fecha\_nacimiento date not null, otros varchar(50));

Tabla direcciones\_empleados:

create table direcciones\_empleados(direcciones\_empleados\_id int key auto\_increment, empleado\_id int not null, direccion\_id int not null, foreign key (empleado\_id) references empleados(empleados\_id) on delete restrict on update cascade, foreign key (direccion\_id) references direcciones(direccion\_id) on delete cascade on update cascade);

Tabla direcciones:

create table direcciones(direccion\_id int key auto\_increment, calle varchar(50) not null, numero\_exterior int not null, numero\_interior int, colonia varchar(50) not null, codigo\_postal varchar(50) not null, ciudad varchar(50) not null, estado varchar(50), pais varchar(50), otros varchar(50));

Tabla registro\_pacientes:

create table registro\_pacientes(registro\_paciente\_id int key auto\_increment, paciente\_id int not null, cuentas\_pacientes\_id int not null, registro\_por\_empleado\_id int not null, fecha\_admision date not null, condicion\_medica varchar(100) not null, otros varchar(50), foreign key (paciente\_id) references pacientes (paciente\_id) on delete restrict on update cascade, foreign key (cuentas\_pacientes\_id) references cuentas\_pacientes(cuentas\_pacientes\_id) on delete restrict on update restrict, foreign key (registro\_por\_empleado\_id) references empleados(empleados\_id) on delete restrict on update cascade);

Tabla cuentas\_pacientes

create table cuentas\_pacientes(cuentas\_pacientes\_id int key auto\_increment, paciente\_id int not null, forma\_pago\_pac\_id int not null, fecha\_emision date not null, otros varchar(50), foreign key (paciente\_id) references pacientes(paciente\_id) on delete restrict on update cascade, foreign key (forma\_pago\_pac\_id) references formas\_pago\_pacientes(forma\_pago\_pac\_id) on delete restrict on update cascade);

Tabla cuentas\_pacientes\_detalles:

create table cuentas\_pacientes\_detalles(cuentas\_pacientes\_det\_id int key auto\_increment, cuentas\_pacientes\_id int not null, detalle varchar(50) default "Diagnostico", cantidad decimal(10,2) not null, costo\_total decimal(12,2) not null, foreign key (cuentas\_pacientes\_id) references cuentas\_pacientes(cuentas\_pacientes\_id) on delete restrict on update cascade);

Tabla formas\_pago\_pacientes:

create table formas\_pago\_pacientes(forma\_pago\_pac\_id int key auto\_increment, paciente\_id int not null, forma\_pago\_id int not null, detalles varchar(50), foreign key (paciente\_id) references pacientes(paciente\_id) on delete restrict on update cascade, foreign key (forma\_pago\_id) references formas\_pago(forma\_pago\_id) on delete restrict on update cascade);

ALTER TABLE formas\_pago\_pacientes

MODIFY forma\_pago\_id INT NULL;

Tabla formas\_pago:

create table formas\_pago(forma\_pago\_id int key auto\_increment, forma\_pago varchar(30) not null);

**Consultas**

**Listado de pacientes que vivan en México:**

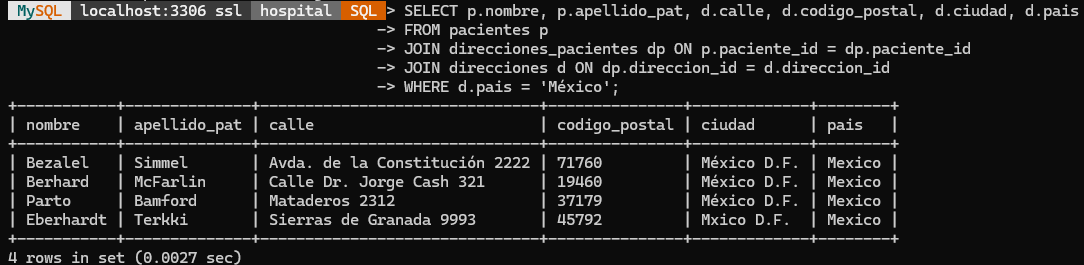
MySQL localhost:3306 ssl hospital SQL > SELECT p.nombre, p.apellido\_pat, d.calle, d.codigo\_postal, d.ciudad, d.pais

-> FROM pacientes p

-> JOIN direcciones\_pacientes dp ON p.paciente\_id = dp.paciente\_id

-> JOIN direcciones d ON dp.direccion\_id = d.direccion\_id

-> WHERE d.pais = 'México';



**Listado de empleados que vivan en Brasil (Brazil):**

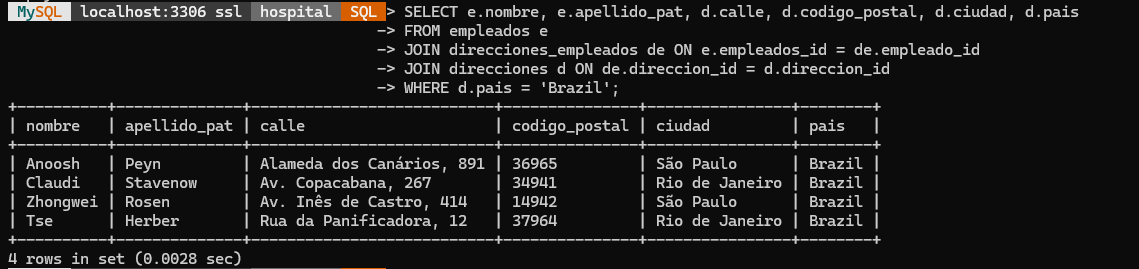
MySQL localhost:3306 ssl hospital SQL > SELECT e.nombre, e.apellido\_pat, d.calle, d.codigo\_postal, d.ciudad, d.pais

-> FROM empleados e

-> JOIN direcciones\_empleados de ON e.empleados\_id = de.empleado\_id

-> JOIN direcciones d ON de.direccion\_id = d.direccion\_id

-> WHERE d.pais = 'Brazil';



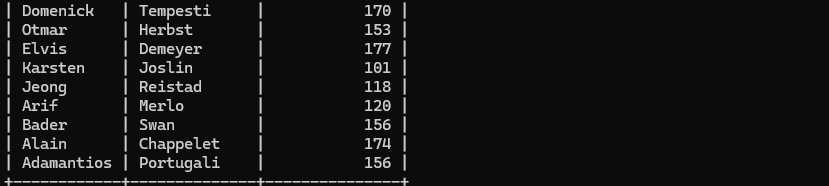
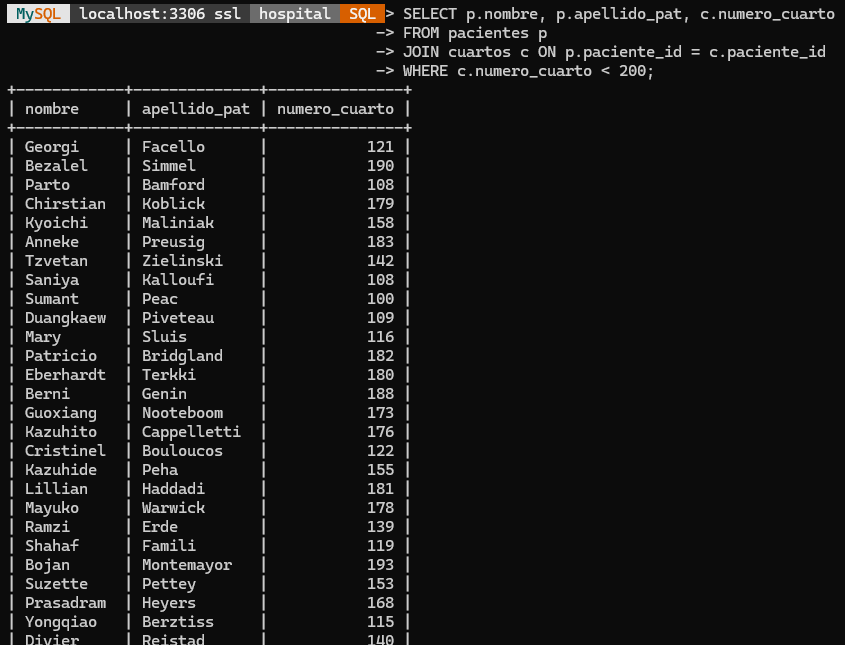
**Listado de pacientes internados en el primer piso del hospital:**

MySQL localhost:3306 ssl hospital SQL > SELECT p.nombre, p.apellido\_pat, c.numero\_cuarto

-> FROM pacientes p

-> JOIN cuartos c ON p.paciente\_id = c.paciente\_id

-> WHERE c.numero\_cuarto < 200;



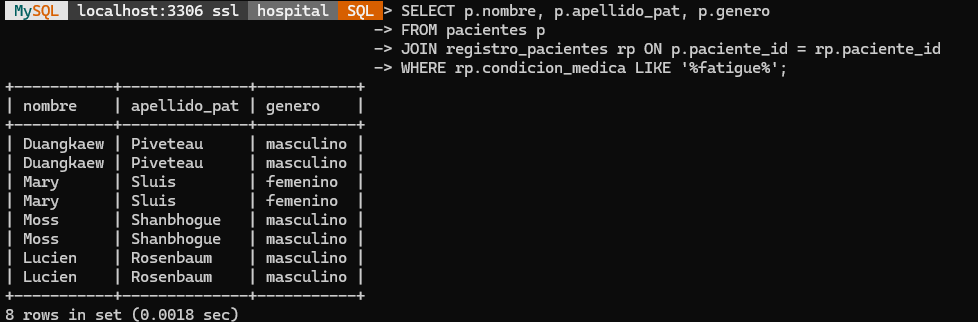
**Listado de pacientes internados por cualquier tipo de fatiga:**

MySQL localhost:3306 ssl hospital SQL > SELECT p.nombre, p.apellido\_pat, p.genero

-> FROM pacientes p

-> JOIN registro\_pacientes rp ON p.paciente\_id = rp.paciente\_id

-> WHERE rp.condicion\_medica LIKE '%fatigue%';

**Nota:** los datos están duplicados porque los pacientes tienen 2 cuentas.

**Realizar un listado de los empleados, y los pacientes que han registrado en la base de datos:**

MySQL localhost:3306 ssl hospital SQL > SELECT e.nombre AS empleado\_nombre, e.apellido\_pat AS empleado\_apellido,

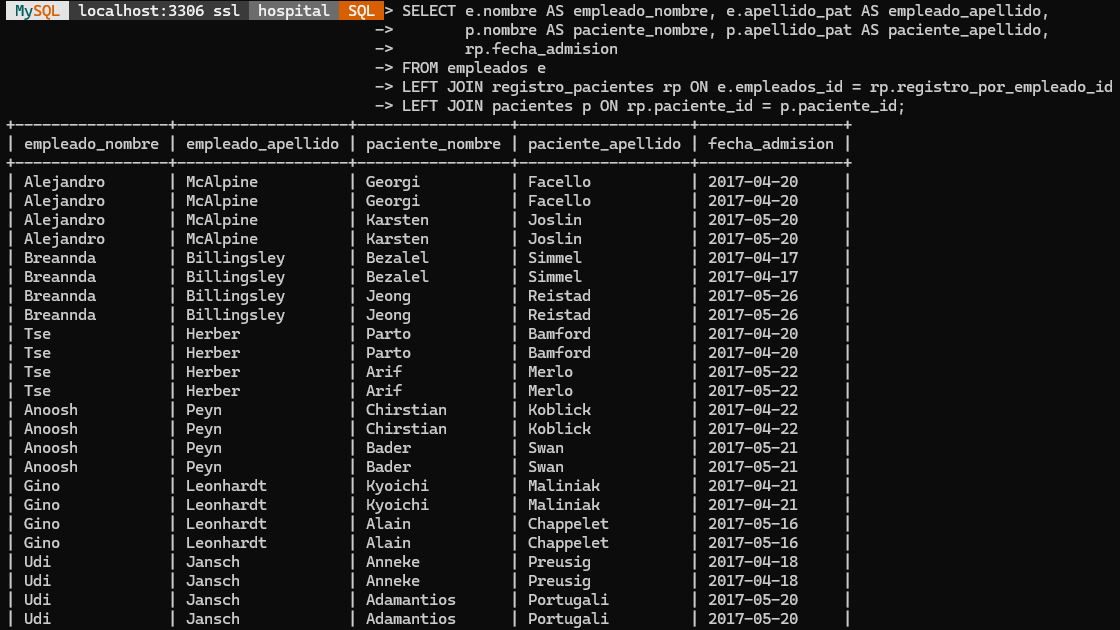
-> p.nombre AS paciente\_nombre, p.apellido\_pat AS paciente\_apellido,

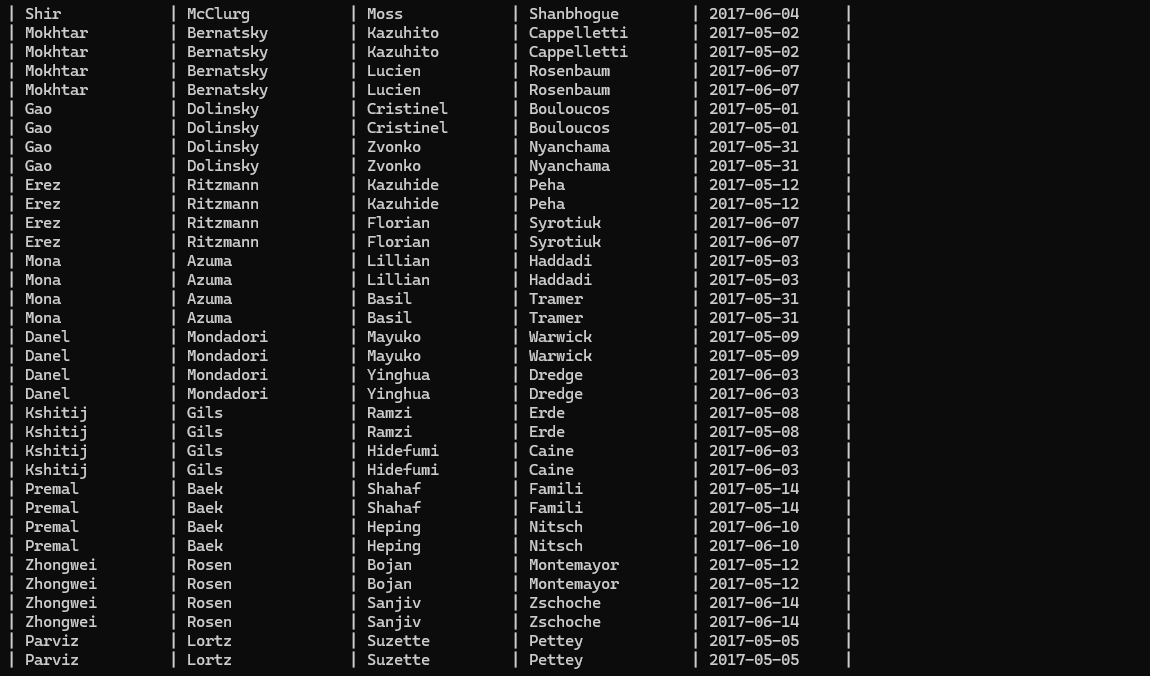
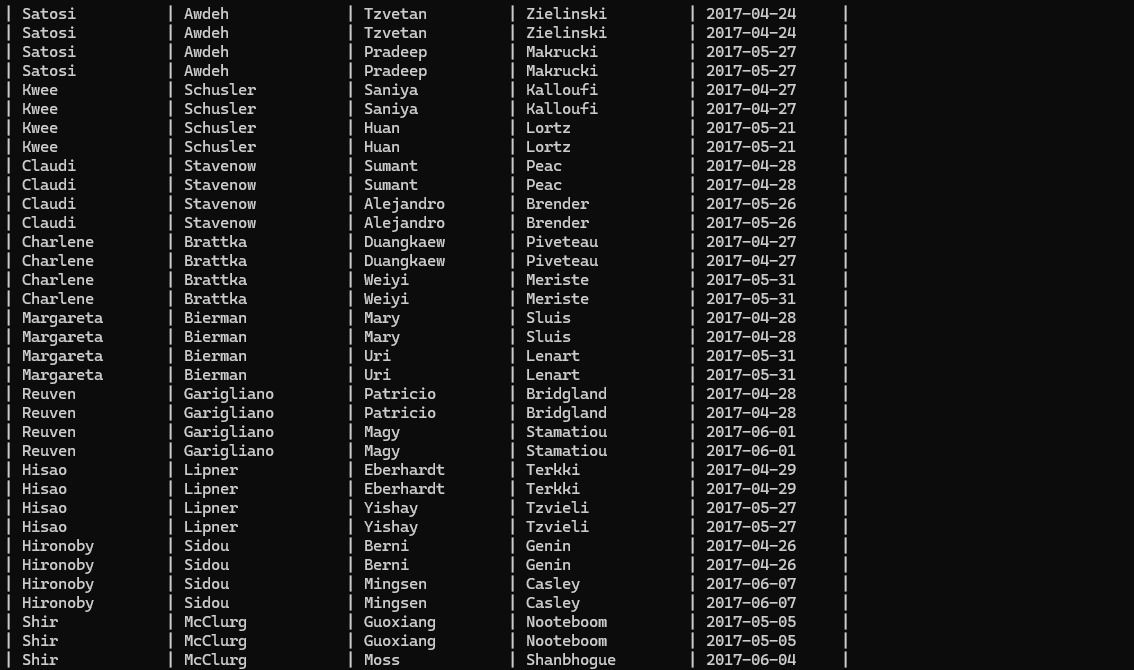
-> rp.fecha\_admision

-> FROM empleados e

-> LEFT JOIN registro\_pacientes rp ON e.empleados\_id = rp.registro\_por\_empleado\_id

-> LEFT JOIN pacientes p ON rp.paciente\_id = p.paciente\_id;





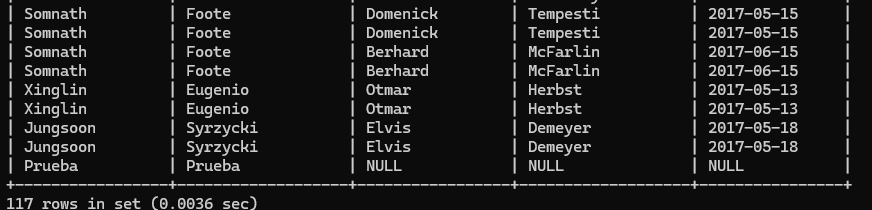
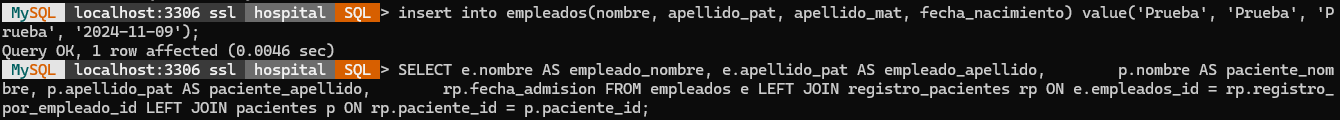


Agregue un empleado llamado “Prueba” (con datos ficticios, los que usted elija), y repita la búsqueda. El empleado debe aparecer, pero sin pacientes:

MySQL localhost:3306 ssl hospital SQL > insert into empleados(nombre, apellido\_pat, apellido\_mat, fecha\_nacimiento) value('Prueba', 'Prueba', 'Prueba', '2024-11-09');

Query OK, 1 row affected (0.0046 sec)

MySQL localhost:3306 ssl hospital SQL > SELECT e.nombre AS empleado\_nombre, e.apellido\_pat AS empleado\_apellido, p.nombre AS paciente\_nombre, p.apellido\_pat AS paciente\_apellido, rp.fecha\_admision FROM empleados e LEFT JOIN registro\_pacientes rp ON e.empleados\_id = rp.registro\_por\_empleado\_id LEFT JOIN pacientes p ON rp.paciente\_id = p.paciente\_id;

**Nota:** los datos están duplicados porque los pacientes tienen 2 cuentas.

**Listado de pacientes (ordenado alfabéticamente por el apellido) y sus cuentas, mostrando el detalle del adeudo para cada cliente:**

MySQL localhost:3306 ssl hospital SQL > SELECT p.nombre, p.apellido\_pat, cp.fecha\_emision,

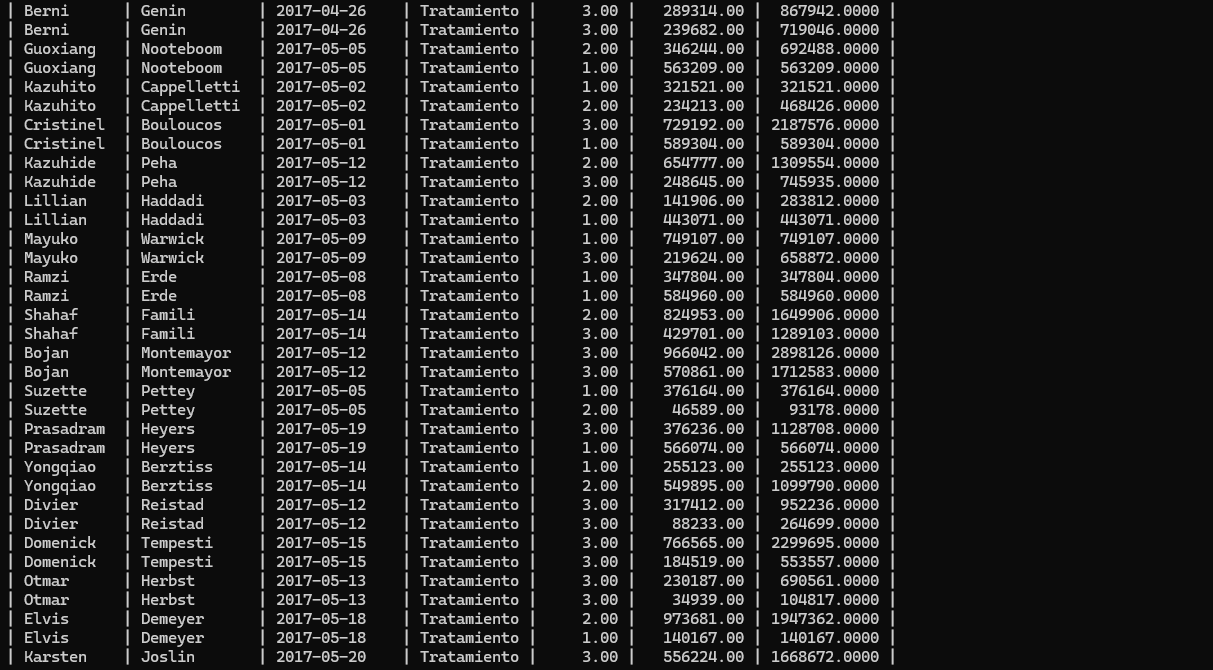
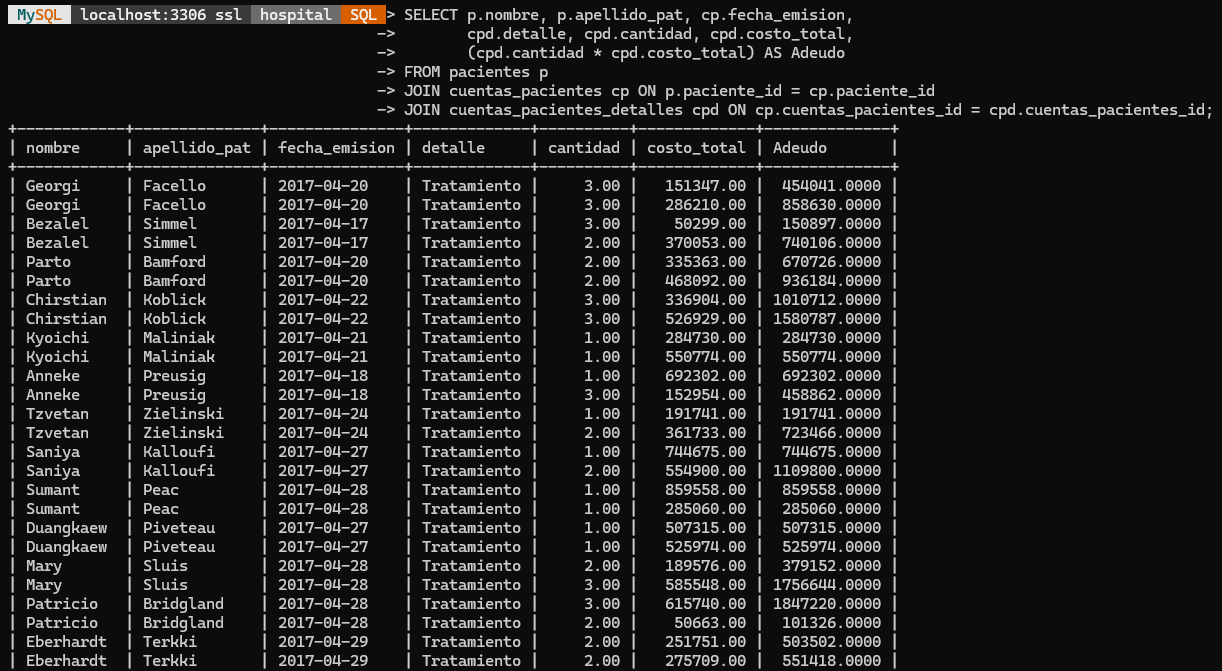
-> cpd.detalle, cpd.cantidad, cpd.costo\_total,

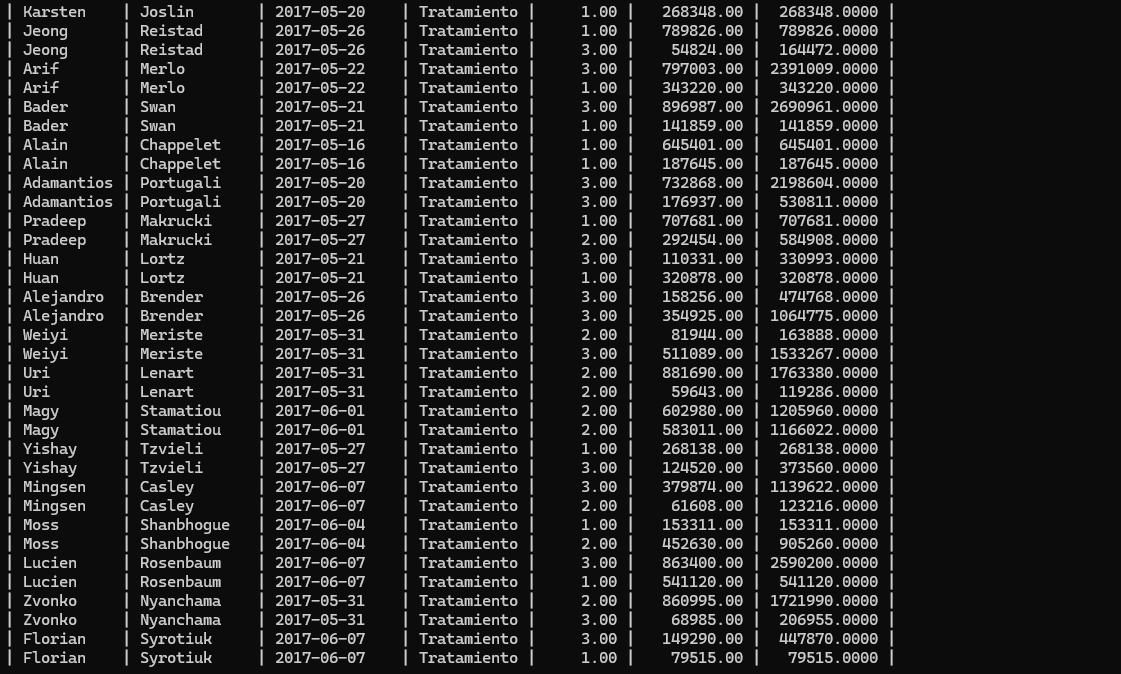
-> (cpd.cantidad \* cpd.costo\_total) AS Adeudo

-> FROM pacientes p

-> JOIN cuentas\_pacientes cp ON p.paciente\_id = cp.paciente\_id

-> JOIN cuentas\_pacientes\_detalles cpd ON cp.cuentas\_pacientes\_id = cpd.cuentas\_pacientes\_id;





**Listado de pacientes y el total (costo monetario) de todos los servicios que se le han otorgado:**

MySQL localhost:3306 ssl hospital SQL > SELECT p.nombre, p.apellido\_pat,

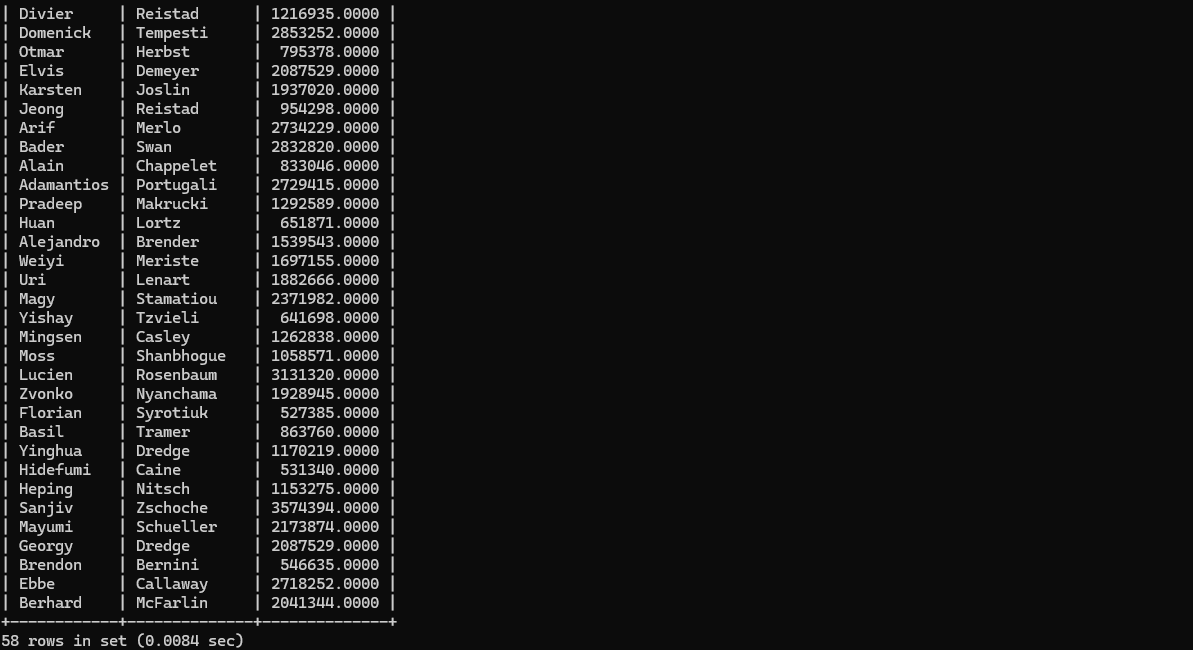
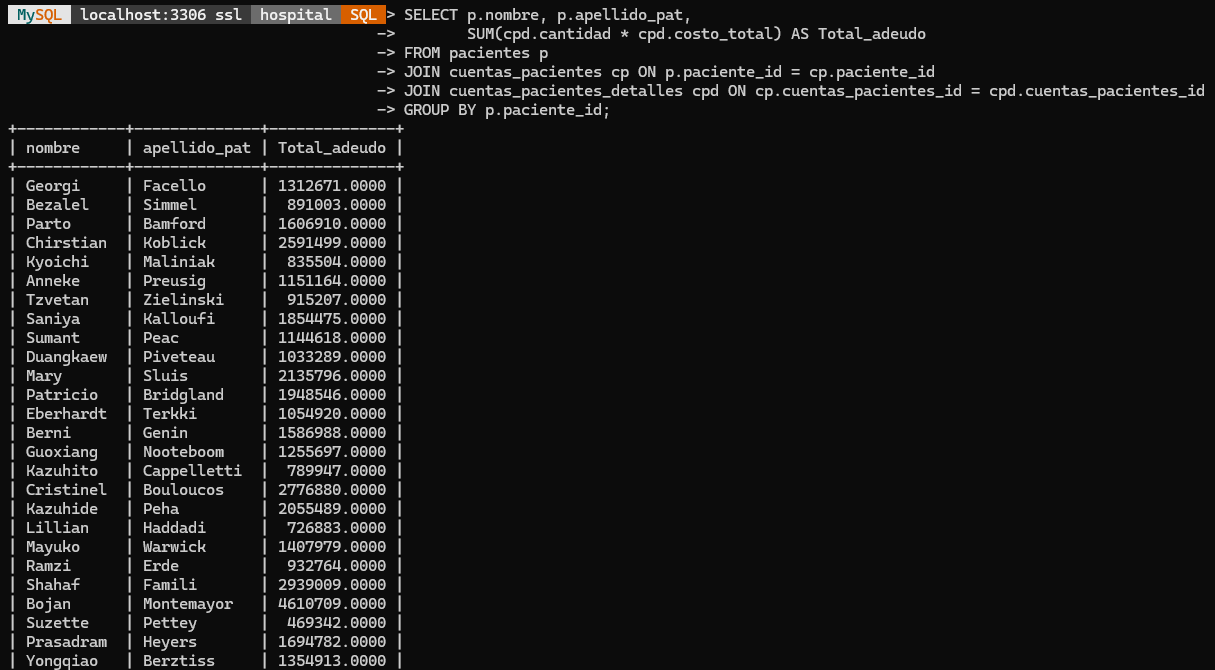
-> SUM(cpd.cantidad \* cpd.costo\_total) AS Total\_adeudo

-> FROM pacientes p

-> JOIN cuentas\_pacientes cp ON p.paciente\_id = cp.paciente\_id

-> JOIN cuentas\_pacientes\_detalles cpd ON cp.cuentas\_pacientes\_id = cpd.cuentas\_pacientes\_id

-> GROUP BY p.paciente\_id;



**Listado de pacientes que NO HAN pagado sus cuentas y el monto que adeudan:**

MySQL localhost:3306 ssl hospital SQL > SELECT p.nombre, p.apellido\_pat,

-> SUM(cpd.cantidad \* cpd.costo\_total) AS Total\_adeudo

-> FROM pacientes p

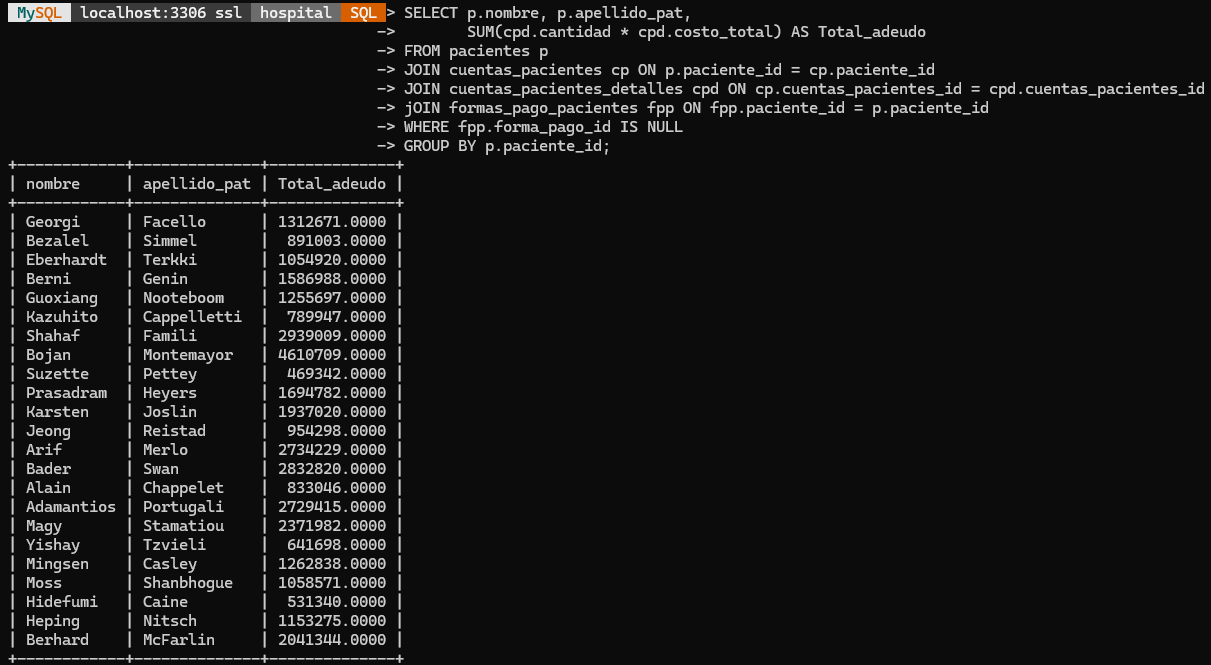
-> JOIN cuentas\_pacientes cp ON p.paciente\_id = cp.paciente\_id

-> JOIN cuentas\_pacientes\_detalles cpd ON cp.cuentas\_pacientes\_id = cpd.cuentas\_pacientes\_id

-> jOIN formas\_pago\_pacientes fpp ON fpp.paciente\_id = p.paciente\_id

-> WHERE fpp.forma\_pago\_id IS NULL

-> GROUP BY p.paciente\_id;



**Listado de pacientes que HAN pagado sus cuentas en EFECTIVO:**

MySQL localhost:3306 ssl hospital SQL > SELECT p.nombre, p.apellido\_pat,

-> SUM(cpd.cantidad \* cpd.costo\_total) AS Total\_adeudo

-> FROM pacientes p

-> JOIN cuentas\_pacientes cp ON p.paciente\_id = cp.paciente\_id

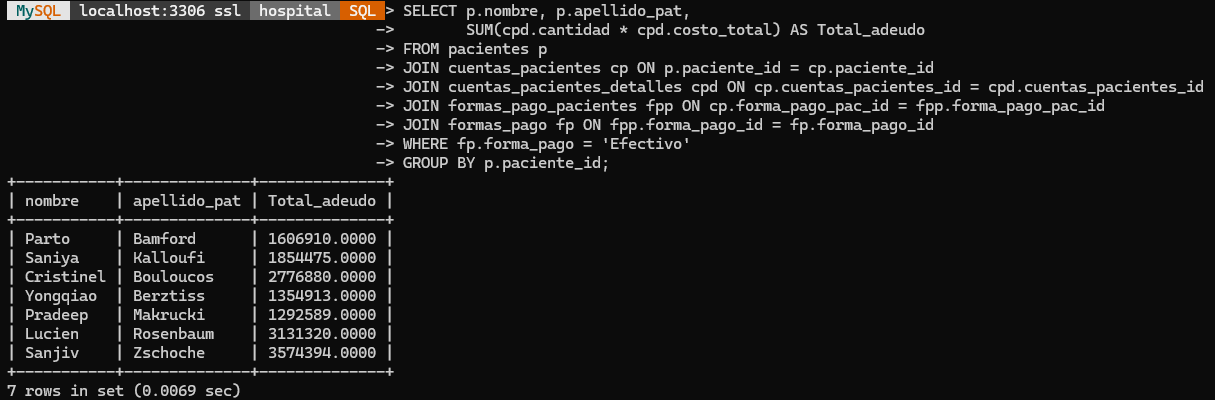
-> JOIN cuentas\_pacientes\_detalles cpd ON cp.cuentas\_pacientes\_id = cpd.cuentas\_pacientes\_id

-> JOIN formas\_pago\_pacientes fpp ON cp.forma\_pago\_pac\_id = fpp.forma\_pago\_pac\_id

-> JOIN formas\_pago fp ON fpp.forma\_pago\_id = fp.forma\_pago\_id

-> WHERE fp.forma\_pago = 'Efectivo'

-> GROUP BY p.paciente\_id;



**Listado de pacientes que HAN pagado sus cuentas y el monto que adeudan:**

MySQL localhost:3306 ssl hospital SQL > SELECT p.nombre, p.apellido\_pat,

-> SUM(cpd.cantidad \* cpd.costo\_total) AS Total\_adeudo

-> FROM pacientes p

-> JOIN cuentas\_pacientes cp ON p.paciente\_id = cp.paciente\_id

-> JOIN cuentas\_pacientes\_detalles cpd ON cp.cuentas\_pacientes\_id = cpd.cuentas\_pacientes\_id

-> JOIN formas\_pago\_pacientes fpp ON fpp.paciente\_id = p.paciente\_id

-> WHERE fpp.forma\_pago\_id IS NOT NULL

-> GROUP BY p.paciente\_id;

