



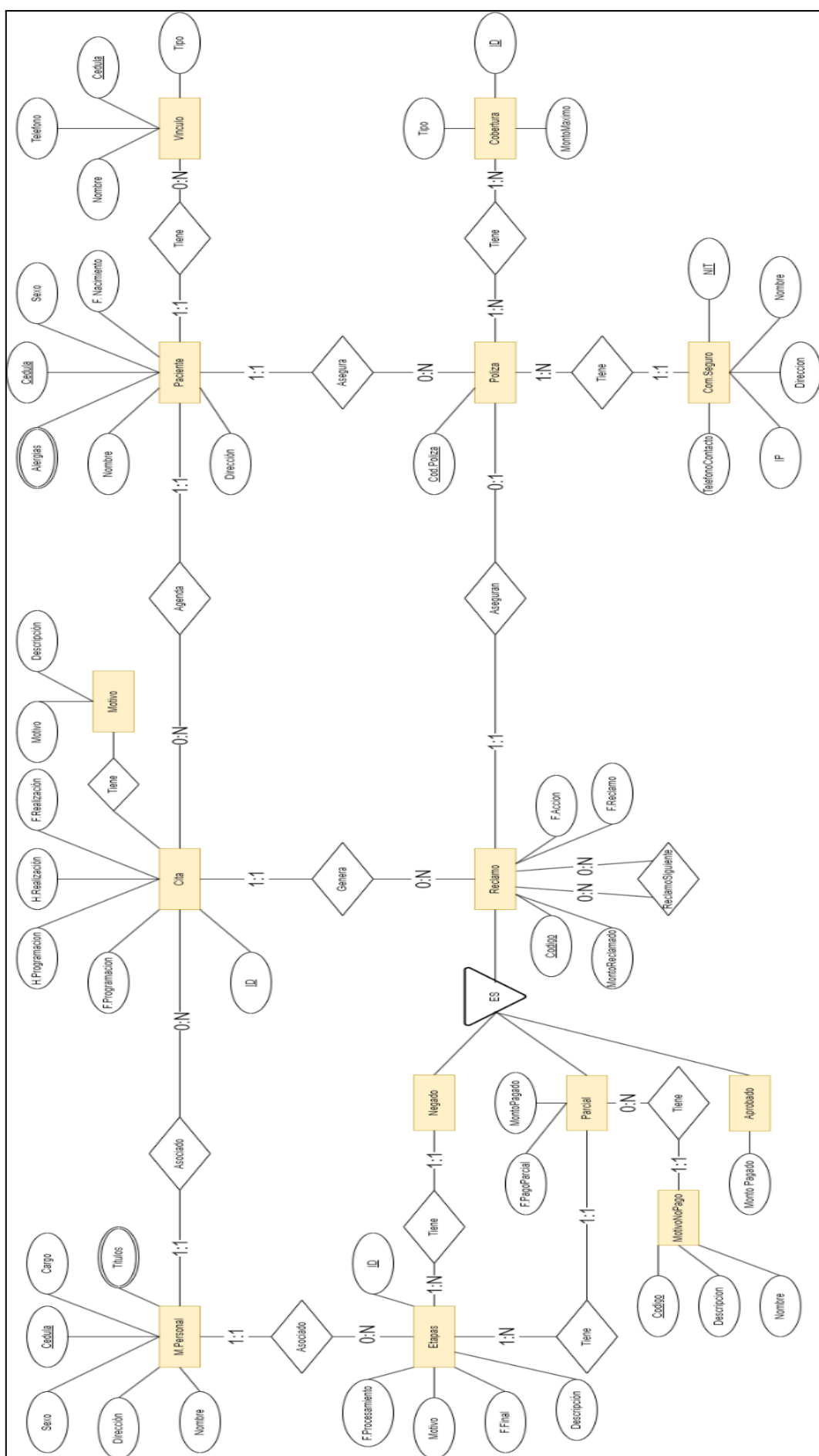
Proyecto 2 - Bases de Datos

Laura Garzón (is360018) - Juan Roza (is360012) - Maria P. Rodríguez (is360019) - Daniel Castro (is360015)

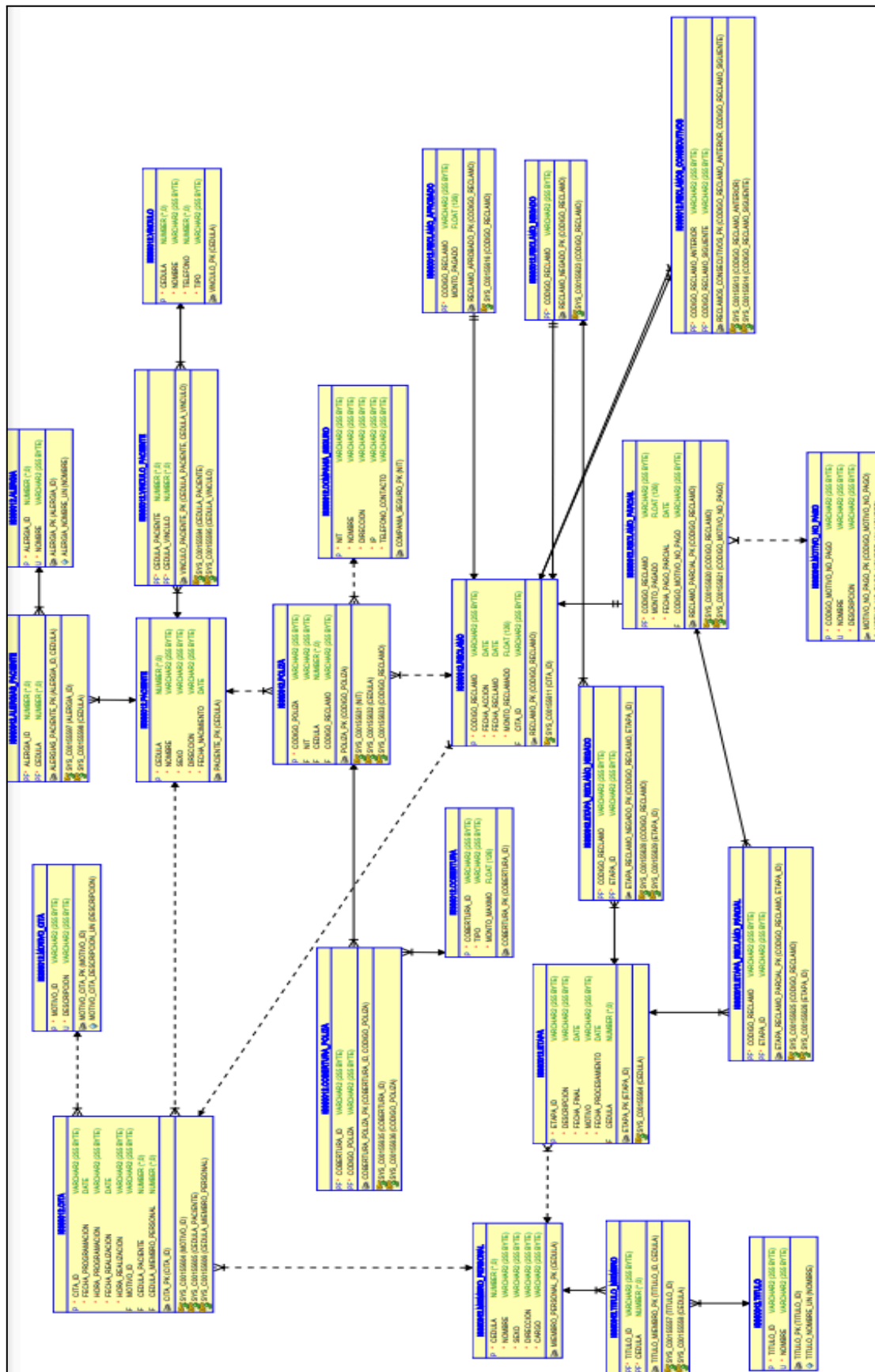
Diseño Entidad / Relación

Para la creación del modelo ER se tomaron las siguientes decisiones:

- Primero se modeló la relación entre Miembro personal, Cita y Paciente tratando a estas 3 como entities con sus respectivos atributos según lo mencionado en el texto. Lo más destacable de este proceso fue:
 1. Tratar como atributo multivaluado a los títulos del miembro de personal pues pueden tener posgrado o pregrado, etc, y a las alergias del paciente pues asimismo pueden ser varias.
 2. Crear un nuevo entity motivo para las citas pues una cita tiene una lista de motivos y un nuevo entity vinculo para aquellos terceros que abogan por el paciente en caso de este estar incapacitado de hacerlo.
- Segundo se modeló la relación entre cita y sus posibles reclamos. Primero se creó una relación entre sí mismo para aquellos reclamos en los que después de su procesamiento la solución es crear un nuevo reclamo. Posteriormente, se manejó herencia de ese entity Reclamo con sus respectivos atributos, todo según los tres posibles casos en los que podría caer un reclamo.
 1. Reclamo Negado: Este fue asociado con un nuevo entity Etapa para cumplir con la medidas de seguimiento de los reclamos en disputa, y es también por esto que esa nueva entity a su vez está asociada con el Miembro de personal que maneja la disputa en esa etapa.
 2. Reclamo Parcial: Este también fue asociado con un nuevo entity etapa para cumplir con la medidas de seguimiento de los reclamos en disputa, que como dijimos anteriormente está asociada con el miembro de personal que maneja la disputa en esa etapa. A su vez está asociada con un nuevo entity MotivoNoPago para manejar correctamente todo el tema de los códigos de razón para no pagar el monto total.
 3. Reclamo Aprobado: A este simplemente se le creó el atributo monto pagado pues ya el resto lo tiene la entity Reclamo.
- Y tercero se modeló la relación entre Póliza, Compañía de seguros y Coberturas. Para ello la lógica fue la siguiente: Una póliza puede tener muchas coberturas y viceversa porque pueden existir como varios “paquetes” ofrecidos por una misma póliza pero pues esos paquetes pueden no ser únicos de ella sino también funcionan para otras pólizas. Ahora bien, una variedad de pólizas si o si tiene que ser ofrecida por una compañía, de esta forma aunque una póliza pueda corresponder a muchas coberturas esta se diferencia porque es ofrecida por X compañía. Así se cumple que cada paciente puede estar asegurado bajo pólizas con muchas compañías de seguros y puede presentar varios reclamos a muchas de estas pólizas con muchas compañías.



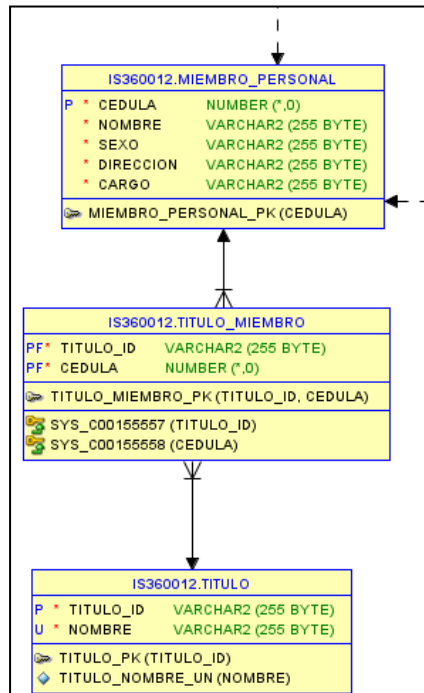
Diseño Relacional



En la anterior imagen se muestra el modelo relacional, en donde se observan las distintas tablas que se tomaron en cuenta para la realización de este modelo, así como sus atributos, llaves primarias y foráneas correspondientes.

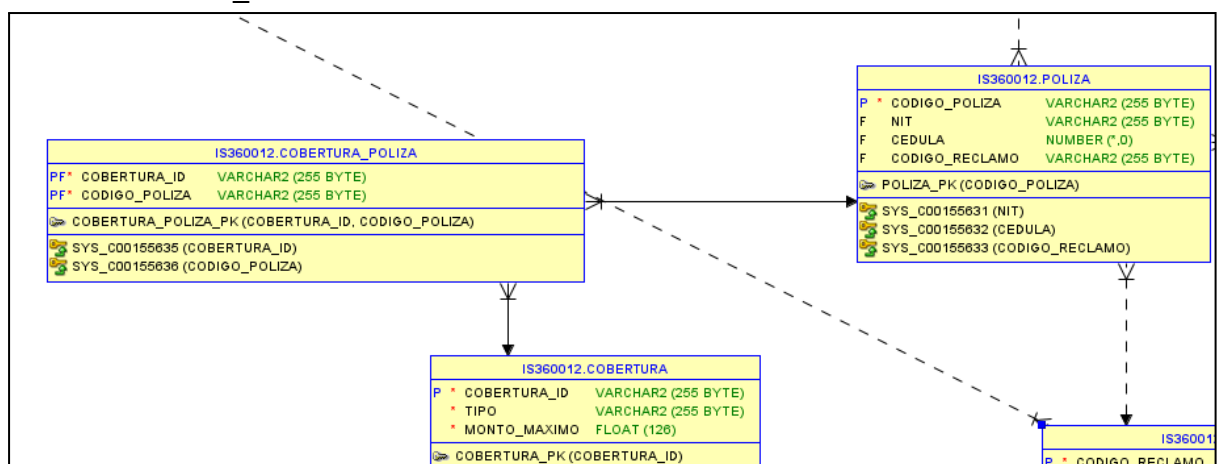
A continuación, se detallan las nuevas tablas de asociación que surgieron debido a relaciones de muchos a muchos entre entidades en el modelo relacional

Tabla Titulo_Miembro



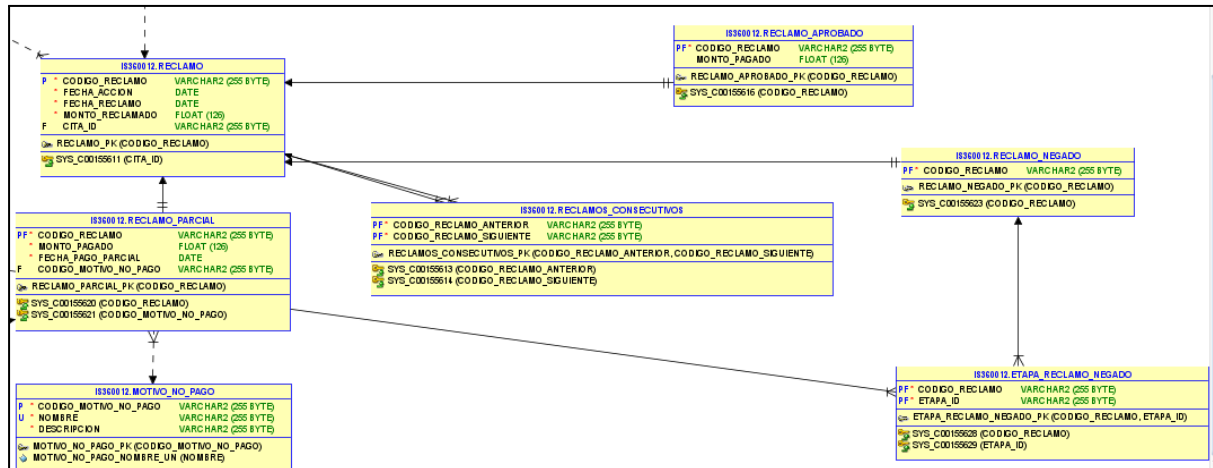
Esta tabla se crea para gestionar la relación entre los títulos o calificaciones y los miembros del personal. Dado que la cardinalidad entre las tablas Miembro_Personal y Titulo es N:N, se requiere una tabla adicional para representar esta relación. La tabla Titulo_Miembro incluye las llaves primarias de ambas tablas mencionadas, es decir, TituloID y Cedula, y se declaran como claves foráneas.

Tabla Cobertura_Poliza



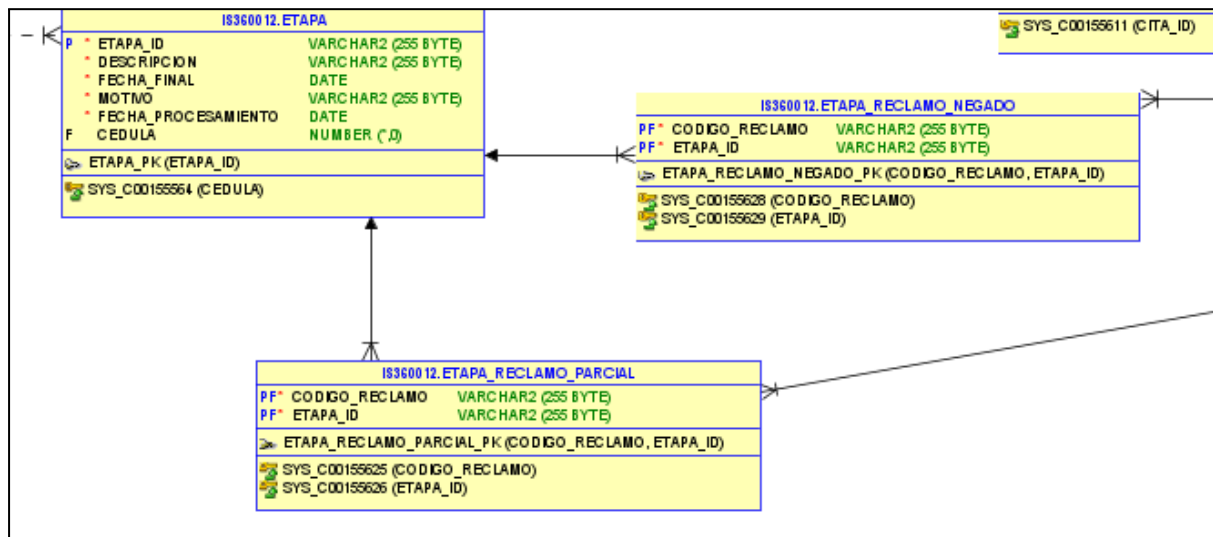
Similarmenre, la tabla Cobertura_Poliza se crea para manejar la relación entre tipos de cobertura y pólizas de seguro. Dado que la cardinalidad entre estas entidades es N:N, se necesita esta tabla intermedia. Contiene las llaves primarias de las tablas Cobertura y Poliza, declaradas como claves foráneas.

Tablas asociadas a Reclamo



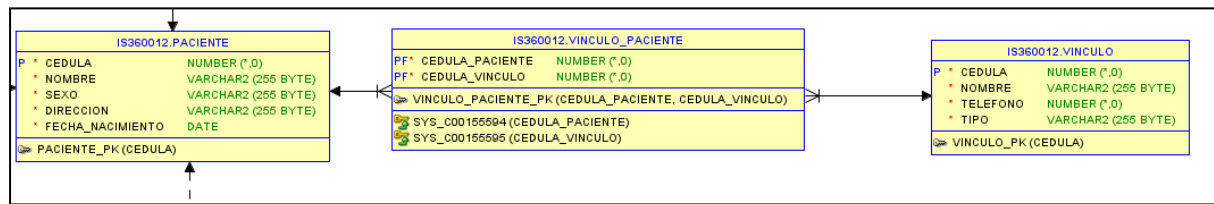
Como se observa en el modelo Entidad-Relación descrito anteriormente, existe una herencia donde las tablas Reclamo_Aprobado, Reclamo_Parcial y Reclamo_Negado heredan de Reclamo. En esta descripción, se pueden observar cada una de las tablas con sus respectivas llaves primarias y foráneas. Además, existe la tabla Reclamos_Consecutivos, la cual se relaciona también con Reclamo para indicar el número total de etapas sucesivas de negociación.

Tablas Etapas_Reclamo_Parcial y Etapas_Reclamo_Negado



Los reclamos negados y parciales son los que tienen etapas, por ello, se realizan las relaciones que se ven en la imagen, donde se crean dos tablas, una Etapas_Reclamo_Parcial, que relaciona las etapas con los reclamos parciales, esta contiene las dos llaves primarias de estas tablas. El mismo procedimiento para los reclamos negados, donde también se incluye la llave primaria de Etapa (Etapa_ID) y el código del reclamo negado (Codigo_Reclamo).

Tabla Vinculo_Paciente



Debido a que un paciente puede tener más de una persona relacionada a él, se creó una nueva tabla (Vinculo_Paciente), en donde se incluye la llave primaria tanto de Paciente como de Vinculo, Cedula_Paciente y Cedula_Vinculo, respectivamente.

Normalización

Es importante resaltar que el modelo se ha desarrollado con el objetivo de alcanzar la tercera forma normal (3FN), al cumplir con este requisito, se asegura una estructura de base de datos eficiente y libre de redundancias, lo que facilita la gestión y el mantenimiento de los datos a largo plazo.

El cumplimiento de la primera forma normal (1FN) se evidencia en cada tabla del modelo, donde los atributos se presentan de manera atómica, sin atributos multivaluados ni duplicados. Por ejemplo, en la tabla de Paciente, los atributos como cédula, sexo, fecha de nacimiento, nombre y dirección son indivisibles, representando valores únicos para cada entidad.

Asimismo, el modelo satisface los criterios de la segunda forma normal (2FN), garantizando que todos los atributos no clave de cada tabla dependan por completo de la clave primaria. Esta disposición previene la redundancia y asegura la integridad de los datos.

En cuanto a la tercera forma normal, además de cumplir con la segunda forma normal, se garantiza la ausencia de dependencias transitivas entre los atributos no clave. Es decir, estos atributos están funcionalmente relacionados exclusivamente con la clave primaria de la tabla y no dependen de otros atributos no clave.

Diseño Físico

```

CREATE TABLE Miembro_Personal (
    cedula INT PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL,
    sexo VARCHAR(255) NOT NULL,
    direccion VARCHAR(255) NOT NULL,
    cargo VARCHAR(255) NOT NULL,
    CONSTRAINT chk_sexo CHECK (sexo IN ('Masculino', 'Femenino'))
);

```

```

CREATE TABLE Titulo (
    titulo_id VARCHAR(255) PRIMARY KEY,
    nombre VARCHAR(255) NOT NULL UNIQUE
);

```

```
CREATE TABLE Titulo_Miembro (  
    titulo_id VARCHAR(255) NOT NULL,  
    cedula INT NOT NULL,  
    PRIMARY KEY (titulo_id, cedula),  
    FOREIGN KEY (titulo_id) REFERENCES Titulo(titulo_id),  
    FOREIGN KEY (cedula) REFERENCES Miembro_Personal(cedula)  
);
```

```
CREATE TABLE Etapas (  
    etapa_id VARCHAR(255) PRIMARY KEY,  
    descripcion VARCHAR(255) NOT NULL,  
    fecha_final DATE NOT NULL,  
    motivo VARCHAR(255) NOT NULL,  
    fecha_procesamiento DATE NOT NULL,  
    cedula INT,  
    FOREIGN KEY (cedula) REFERENCES Miembro_Personal(cedula)  
);
```

```
CREATE TABLE Cobertura (  
    cobertura_id VARCHAR(255) PRIMARY KEY,  
    tipo VARCHAR(255) NOT NULL,  
    monto_maximo FLOAT NOT NULL  
);
```

```
CREATE TABLE Alergia (  
    alergia_id INT PRIMARY KEY,  
    nombre VARCHAR(255) UNIQUE NOT NULL  
);
```

```
CREATE TABLE Motivo_No_Pago (  
    codigo_motivo_no_pago VARCHAR(255) PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL UNIQUE,  
    descripcion VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Compania_Seguro (  
    nit VARCHAR(255) PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    direccion VARCHAR(255) NOT NULL,  
    ip VARCHAR(255) NOT NULL,  
    telefono_contacto VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Motivo_Cita (  
    motivo_id VARCHAR(255) PRIMARY KEY,  
    descripcion VARCHAR(255) UNIQUE NOT NULL  
);
```

```
CREATE TABLE Vinculo (  
    cedula INT PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    telefono INT NOT NULL,  
    tipo VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Paciente (  
    cedula INT PRIMARY KEY,  
    nombre VARCHAR(255) NOT NULL,  
    sexo VARCHAR(255) NOT NULL,  
    direccion VARCHAR(255) NOT NULL,  
    fecha_nacimiento DATE NOT NULL,  
    CONSTRAINT check_sexo CHECK (sexo IN ('Masculino', 'Femenino'))  
);
```

```
CREATE TABLE Vinculo_Paciente (  
    cedula_paciente INT,  
    cedula_vinculo INT,  
    PRIMARY KEY (cedula_paciente, cedula_vinculo),  
    FOREIGN KEY (cedula_paciente) REFERENCES Paciente(cedula),  
    FOREIGN KEY (cedula_vinculo) REFERENCES Vinculo(cedula)  
);
```

```
CREATE TABLE Alergias_Paciente (  
    alergia_id INT,  
    cedula INT,  
    PRIMARY KEY (alergia_id, cedula),  
    FOREIGN KEY (alergia_id) REFERENCES Alergia(alergia_id),  
    FOREIGN KEY (cedula) REFERENCES Paciente(cedula)  
);
```

```
CREATE TABLE Cita (  
    cita_id VARCHAR(255) PRIMARY KEY,  
    fecha_programacion DATE NOT NULL,  
    hora_programacion VARCHAR(255) NOT NULL,  
    fecha_realizacion DATE NOT NULL,  
    hora_realizacion VARCHAR(255) NOT NULL,  
    motivo_id VARCHAR(255),  
    cedula_paciente INT,  
    cedula_miembro_personal INT,  
    FOREIGN KEY (motivo_id) REFERENCES Motivo_Cita(motivo_id),  
    FOREIGN KEY (cedula_paciente) REFERENCES Paciente(cedula),  
    FOREIGN KEY (cedula_miembro_personal) REFERENCES Miembro_Personal(cedula)  
);
```



```
CREATE TABLE Reclamo (  
    codigo_reclamo VARCHAR(255) PRIMARY KEY,  
    fecha_accion DATE NOT NULL,  
    fecha_reclamo DATE NOT NULL,  
    monto_reclamado FLOAT NOT NULL,  
    cita_id VARCHAR(255),  
    FOREIGN KEY (cita_id) REFERENCES Cita(cita_id)  
);
```

```
CREATE TABLE Reclamos_Consecutivos (  
    codigo_reclamo_anterior VARCHAR(255),  
    codigo_reclamo_siguiente VARCHAR(255),  
    PRIMARY KEY (codigo_reclamo_anterior, codigo_reclamo_siguiente),  
    FOREIGN KEY (codigo_reclamo_anterior) REFERENCES Reclamo,  
    FOREIGN KEY (codigo_reclamo_siguiente) REFERENCES Reclamo  
);
```

```
CREATE TABLE Reclamo_Aprobado (  
    codigo_reclamo VARCHAR(255) PRIMARY KEY,  
    monto_pagado FLOAT,  
    FOREIGN KEY (codigo_reclamo) REFERENCES Reclamo(codigo_reclamo)  
);
```

```
CREATE TABLE Reclamo_Parcial (  
    codigo_reclamo VARCHAR(255) PRIMARY KEY,  
    monto_pagado FLOAT NOT NULL,  
    fecha_pago_parcial DATE NOT NULL,  
    codigo_motivo_no_pago VARCHAR(255),  
    FOREIGN KEY (codigo_reclamo) REFERENCES Reclamo,  
    FOREIGN KEY (codigo_motivo_no_pago) REFERENCES  
Motivo_No_Pago(codigo_motivo_no_pago)  
);
```

```
CREATE TABLE Reclamo_Negado (  
    codigo_reclamo VARCHAR(255) PRIMARY KEY,  
    FOREIGN KEY (codigo_reclamo) REFERENCES Reclamo  
);
```

```
CREATE TABLE Etapas_Reclamo_Parcial (  
    codigo_reclamo VARCHAR(255),  
    etapa_id VARCHAR(255),  
    PRIMARY KEY (codigo_reclamo, etapa_id),  
    FOREIGN KEY (codigo_reclamo) REFERENCES Reclamo_Parcial,  
    FOREIGN KEY (etapa_id) REFERENCES Etapa  
);
```

```
CREATE TABLE Etapla_Reclamo_Negado (
    codigo_reclamo VARCHAR(255),
    etapa_id VARCHAR(255),
    PRIMARY KEY (codigo_reclamo, etapa_id),
    FOREIGN KEY (codigo_reclamo) REFERENCES Reclamo_Negado,
    FOREIGN KEY (etapa_id) REFERENCES Etapa
);
```

```
CREATE TABLE Poliza (
    codigo_poliza VARCHAR(255) PRIMARY KEY,
    nit VARCHAR(255),
    cedula INT,
    codigo_reclamo VARCHAR(255),
    FOREIGN KEY (nit) REFERENCES Compania_Seguro(nit),
    FOREIGN KEY (cedula) REFERENCES Paciente(cedula),
    FOREIGN KEY (codigo_reclamo) REFERENCES Reclamo
);
```

```
CREATE TABLE Cobertura_Poliza (
    cobertura_id VARCHAR(255),
    codigo_poliza VARCHAR(255),
    PRIMARY KEY (cobertura_id, codigo_poliza),
    FOREIGN KEY (cobertura_id) REFERENCES Cobertura,
    FOREIGN KEY (codigo_poliza) REFERENCES Poliza
);
```

Inserción de valores

-- Insertar datos en la tabla Miembro_Personal

```
INSERT INTO Miembro_Personal (cedula, nombre, sexo, direccion, cargo)
VALUES (123456, 'Juan Pérez', 'Masculino', 'Av. Siempre Viva 123', 'Doctor');
INSERT INTO Miembro_Personal (cedula, nombre, sexo, direccion, cargo)
VALUES (789012, 'Ana López', 'Femenino', 'Calle Falsa 456', 'Enfermera');
INSERT INTO Miembro_Personal (cedula, nombre, sexo, direccion, cargo)
VALUES (102715, 'Daniel Castro', 'Masculino', 'Calle 127 #59-69', 'Medico');
```

-- Insertar datos en la tabla Titulo

```
INSERT INTO Titulo (titulo_id, nombre)
VALUES ('T1', 'Doctorado');
INSERT INTO Titulo (titulo_id, nombre)
VALUES ('T2', 'Maestría');
```

-- Insertar datos en la tabla Titulo_Miembro

```
INSERT INTO Titulo_Miembro (titulo_id, cedula)
VALUES ('T1', 123456);
INSERT INTO Titulo_Miembro (titulo_id, cedula)
VALUES ('T2', 789012);
INSERT INTO Titulo_Miembro (titulo_id, cedula)
VALUES ('T2', 102715);
```

-- Insertar datos en la tabla Vinculo

```
INSERT INTO Vinculo (cedula, nombre, telefono, tipo)
VALUES (123, 'Juan Erasmo', 123456789, 'Familiar');
INSERT INTO Vinculo (cedula, nombre, telefono, tipo)
VALUES (456, 'María Gómez', 987654321, 'Amiga');
INSERT INTO Vinculo (cedula, nombre, telefono, tipo)
VALUES (789, 'Juan Rozo', 019283746, 'Mejor Amigo');
```

-- Insertar datos en la tabla Paciente

```
INSERT INTO Paciente (cedula, nombre, sexo, direccion, fecha_nacimiento)
VALUES (111, 'Carlos Pérez', 'Masculino', 'Calle 1, Ciudad 1', TO_DATE('1980-01-01',
'YYYY-MM-DD'));
INSERT INTO Paciente (cedula, nombre, sexo, direccion, fecha_nacimiento)
VALUES (222, 'Laura Martínez', 'Femenino', 'Calle 2, Ciudad 2', TO_DATE('1990-02-02',
'YYYY-MM-DD'));
INSERT INTO Paciente (cedula, nombre, sexo, direccion, fecha_nacimiento)
VALUES (333, 'Mapa Ruiz', 'Femenino', 'Calle 19, Carrera 91', TO_DATE('2005-12-10',
'YYYY-MM-DD'));
INSERT INTO Paciente (cedula, nombre, sexo, direccion, fecha_nacimiento)
VALUES (444, 'Carlos Ferrer', 'Masculino', 'Calle 3, Carrera 20', TO_DATE('1995-03-03',
'YYYY-MM-DD'));
```

-- Insertar datos en la tabla Vinculo_Paciente

```
INSERT INTO Vinculo_Paciente (cedula_vinculo, cedula_paciente)
VALUES (123, 111);
INSERT INTO Vinculo_Paciente (cedula_vinculo, cedula_paciente)
VALUES (456, 222);
INSERT INTO Vinculo_Paciente (cedula_vinculo, cedula_paciente)
VALUES (789, 444);
```

-- Insertar datos en la tabla Alergia

```
INSERT INTO Alergia (alergia_id, nombre)
VALUES (1, 'Penicilina');
INSERT INTO Alergia (alergia_id, nombre)
VALUES (2, 'Mariscos');
```

-- Insertar datos en la tabla Alergias_Paciente

```
INSERT INTO Alergias_Paciente (alergia_id, cedula)
VALUES (1, 111);
INSERT INTO Alergias_Paciente (alergia_id, cedula)
VALUES (2, 222);
```

-- Insertar datos en la tabla Motivo_Cita

```
INSERT INTO Motivo_Cita (motivo_id, descripcion)
VALUES ('M1', 'Consulta general');
INSERT INTO Motivo_Cita (motivo_id, descripcion)
VALUES ('M2', 'Revisión de resultados');
```

-- Insertar datos en la tabla Cita

```
INSERT INTO Cita (cita_id, fecha_programacion, hora_programacion, fecha_realizacion,
hora_realizacion, motivo_id, cedula_paciente, cedula_miembro_personal)
VALUES ('C1', TO_DATE('2024-05-01', 'YYYY-MM-DD'), '10:00:00', TO_DATE('2024-05-01',
'YYYY-MM-DD'), '10:30:00', 'M1', 111, 123456);
INSERT INTO Cita (cita_id, fecha_programacion, hora_programacion, fecha_realizacion,
hora_realizacion, motivo_id, cedula_paciente, cedula_miembro_personal)
VALUES ('C2', TO_DATE('2024-05-02', 'YYYY-MM-DD'), '11:00:00', TO_DATE('2024-05-02',
'YYYY-MM-DD'), '11:30:00', 'M2', 222, 789012);
INSERT INTO Cita (cita_id, fecha_programacion, hora_programacion, fecha_realizacion,
hora_realizacion, motivo_id, cedula_paciente, cedula_miembro_personal)
VALUES ('C3', TO_DATE('2024-04-30', 'YYYY-MM-DD'), '12:00:00', TO_DATE('2024-05-02',
'YYYY-MM-DD'), '12:30:00', 'M1', 333, 102715);
```

-- Insertar datos en la tabla Etapa

```
INSERT INTO Etapa (etapa_id, descripcion, fecha_final, motivo, fecha_procesamiento, cedula)
VALUES ('E1', 'Primera revisión', TO_DATE('2024-05-02', 'YYYY-MM-DD'), 'Datos incompletos',
TO_DATE('2024-05-01', 'YYYY-MM-DD'), 123456);
INSERT INTO Etapa (etapa_id, descripcion, fecha_final, motivo, fecha_procesamiento, cedula)
VALUES ('E2', 'Segunda revisión', TO_DATE('2024-05-03', 'YYYY-MM-DD'), 'Administración de
medicamentos', TO_DATE('2024-05-01', 'YYYY-MM-DD'), 789012);
INSERT INTO Etapa (etapa_id, descripcion, fecha_final, motivo, fecha_procesamiento, cedula)
VALUES ('E3', 'Tercera revisión', TO_DATE('2024-05-05', 'YYYY-MM-DD'), 'Sello aprobacion',
TO_DATE('2024-05-04', 'YYYY-MM-DD'), 102715);
```

-- Insertar datos en la tabla Cobertura

```
INSERT INTO Cobertura (cobertura_id, tipo, monto_maximo)
VALUES ('C1', 'Salud', 10000);
INSERT INTO Cobertura (cobertura_id, tipo, monto_maximo)
VALUES ('C2', 'Vida', 50000);
```

-- Insertar datos en la tabla Motivo_No_Pago

```
INSERT INTO Motivo_No_Pago (codigo_motivo_no_pago, nombre, descripcion)
VALUES ('MNP1', 'Sin autorización', 'Pago no autorizado');
INSERT INTO Motivo_No_Pago (codigo_motivo_no_pago, nombre, descripcion)
VALUES ('MNP2', 'Datos incompletos', 'Información insuficiente');
```

-- Insertar datos en la tabla Compania_Seguro

```
INSERT INTO Compania_Seguro (nit, nombre, direccion, ip, telefono_contacto)
VALUES ('NIT123', 'Seguros Generales', 'Av. Principal 123', '192.168.5.1', '555-1234');
INSERT INTO Compania_Seguro (nit, nombre, direccion, ip, telefono_contacto)
VALUES ('NIT456', 'Seguros Avanzados', 'Calle Grande 789', '192.168.5.2', '555-5678');
```

-- Insertar datos en la tabla Reclamo

```
INSERT INTO Reclamo (codigo_reclamo, fecha_accion, fecha_reclamo, monto_reclamado, cita_id)
VALUES ('R1', TO_DATE('2024-05-02', 'YYYY-MM-DD'), TO_DATE('2024-05-03',
'YYYY-MM-DD'), 5000, 'C1');
```

```
INSERT INTO Reclamo (codigo_reclamo, fecha_accion, fecha_reclamo, monto_reclamado, cita_id)
VALUES ('R2', TO_DATE('2024-05-03', 'YYYY-MM-DD'), TO_DATE('2024-05-04',
'YYYY-MM-DD'), 7000, 'C2');
```

```
INSERT INTO Reclamo (codigo_reclamo, fecha_accion, fecha_reclamo, monto_reclamado, cita_id)
VALUES ('R3', TO_DATE('2024-05-04', 'YYYY-MM-DD'), TO_DATE('2024-05-05',
'YYYY-MM-DD'), 10000, 'C3');
```

```
INSERT INTO Reclamo (codigo_reclamo, fecha_accion, fecha_reclamo, monto_reclamado, cita_id)
VALUES ('R4', TO_DATE('2024-05-05', 'YYYY-MM-DD'), TO_DATE('2024-05-06',
'YYYY-MM-DD'), 14000, 'C3');
```

-- Insertar datos en la tabla Reclamos_Consecutivos

```
INSERT INTO Reclamos_Consecutivos (codigo_reclamo_anterior, codigo_reclamo_siguiente)
VALUES ('R3', 'R4');
```

-- Insertar datos en la tabla Reclamo_Aprobado

```
INSERT INTO Reclamo_Aprobado (codigo_reclamo, monto_pagado)
VALUES ('R1', 5000);
```

-- Insertar datos en la tabla Reclamo_Parcial

```
INSERT INTO Reclamo_Parcial (codigo_reclamo, monto_pagado, fecha_pago_parcial,
codigo_motivo_no_pago)
VALUES ('R2', 3000, TO_DATE('2024-05-05', 'YYYY-MM-DD'), 'MNP1');
```

-- Insertar datos en la tabla Reclamo_Negado

```
INSERT INTO Reclamo_Negado (codigo_reclamo)
VALUES ('R3');
```

-- Insertar datos en la tabla Etapa_Reclamo_Parcial

```
INSERT INTO Etapa_Reclamo_Parcial (codigo_reclamo, etapa_id)
VALUES ('R2', 'E1');
```

-- Insertar datos en la tabla Etapa_Reclamo_Negado

```
INSERT INTO Etapa_Reclamo_Negado (codigo_reclamo, etapa_id)
VALUES ('R3', 'E2');
INSERT INTO Etapa_Reclamo_Negado (codigo_reclamo, etapa_id)
VALUES ('R3', 'E3');
```

-- Insertar datos en la tabla Poliza

```
INSERT INTO Poliza (codigo_poliza, nit, cedula, codigo_reclamo)
VALUES ('P1', 'NIT123', 111, 'R1');
INSERT INTO Poliza (codigo_poliza, nit, cedula, codigo_reclamo)
VALUES ('P2', 'NIT456', 222, 'R2');
```

-- Insertar datos en la tabla Cobertura_Poliza

```
INSERT INTO Cobertura_Poliza (cobertura_id, codigo_poliza)
VALUES ('C1', 'P1');
INSERT INTO Cobertura_Poliza (cobertura_id, codigo_poliza)
VALUES ('C2', 'P2');
```

Diccionario de datos

Tabla Miembro_personal

Llaves	Campo	Tipo	Descripción	Dominio
	nombre	VARCHAR2(255)	Nombre del miembro del personal	NOT NULL
PK	cedula	NUMBER(38)	Cédula de ciudadanía del miembro del personal	
	sexo	VARCHAR2(255)	Sexo del miembro del personal	NOT NULL ['Masculino', 'Femenino']
	dirección	VARCHAR2(255)	Dirección de la residencia del miembro del personal	NOT NULL
	cargo	VARCHAR2(255)	Función del miembro del personal dentro de la organización	NOT NULL

Tabla Título

Llaves	Campo	Tipo	Descripción	Dominio
PK	titulo_id	VARCHAR2(255)	Identificador del título	
	nombre	VARCHAR2(255)	Nombre del título	NOT NULL UNIQUE

Tabla Título_miembro

Llaves	Campo	Tipo	Descripción	Dominio
PK FK	titulo_id	VARCHAR2(255)	Identificador del título	NOT NULL
PK FK	cedula	NUMBER(38)	Cédula de ciudadanía del miembro del personal	NOT NULL

Tabla Etapa

Llaves	Campo	Tipo	Descripción	Dominio
PK	etapa_id	VARCHAR2(255)	Identificador de la etapa	
	descripción	VARCHAR2(255)	Descripción de la etapa	NOT NULL
	fecha_final	DATE	Fecha en la que finaliza la etapa	NOT NULL
	motivo	VARCHAR2(255)	Motivo de disputa que causa el paso de procesamiento	NOT NULL
	fecha_procesamiento	DATE	Fecha del procesamiento	NOT NULL
FK	cedula	NUMBER(38)	Cédula del miembro personal asociado	

Tabla Cobertura

Llaves	Campo	Tipo	Descripción	Dominio
PK	cobertura_id	VARCHAR2(255)	Identificador de la cobertura	
	tipo	VARCHAR2(255)	Tipo de cobertura	NOT NULL
	monto_maximo	FLOAT(126)	Monto máximo por el cual se dará la cobertura	NOT NULL

Tabla Alergia

Llaves	Campo	Tipo	Descripción	Dominio
PK	alergia_id	NUMBER(38)	Identificador de la alergia	
	nombre	VARCHAR2(255)	Nombre de la alergia	NOT NULL

Tabla Motivo_no_pago

Llaves	Campo	Tipo	Descripción	Dominio
PK	codigo_motivo_no_pago	VARCHAR2(255)	Código que identifica al motivo para no pagar	
	nombre	VARCHAR2(255)	Nombre del motivo	NOT NULL UNIQUE
	descripción	VARCHAR2(255)	Descripción del motivo	NOT NULL

Tabla Compania_seguro

Llaves	Campo	Tipo	Descripción	Dominio
PK	nit	VARCHAR2(255)	Número de Identificación Tributaria	
	nombre	VARCHAR2(255)	Nombre de la compañía	NOT NULL
	direccion	VARCHAR2(255)	Dirección de la compañía	NOT NULL
	ip	VARCHAR2(255)	Dirección única que identifica a un dispositivo en Internet o en una red local, perteneciente a la compañía.	NOT NULL
	telefono_contacto	VARCHAR2(255)	Teléfono de la persona de contacto de la empresa	NOT NULL

Tabla Motivo_cita

Llaves	Campo	Tipo	Descripción	Dominio
PK	motivo_id	NUMBER(38)	Identificador del motivo	
	descripción	VARCHAR2(255)	Descripción del motivo	NOT NULL UNIQUE

Tabla Vinculo

Llaves	Campo	Tipo	Descripción	Dominio
PK	cedula	NUMBER(38)	Cédula de ciudadanía que identifica a la persona vinculada	
	nombre	VARCHAR2(255)	Nombre de la persona vinculada	NOT NULL
	teléfono	NUMBER(38)	Teléfono de la persona vinculada	NOT NULL
	tipo	VARCHAR2(255)	Tipo de vínculo con el paciente	NOT NULL

Tabla Paciente

Llaves	Campo	Tipo	Descripción	Dominio
PK	cedula	NUMBER(38)	Cédula de ciudadanía del paciente	
	nombre	VARCHAR2(255)	Nombre del paciente	NOT NULL
	sexo	VARCHAR2(255)	Sexo del paciente	NOT NULL ['Masculino', 'Femenino']
	direccion	VARCHAR2(255)	Dirección de residencia del paciente	NOT NULL
	fecha_nacimiento	DATE	Fecha de nacimiento del paciente	NOT NULL

Tabla Vinculo_Paciente

Llaves	Campo	Tipo	Descripción	Dominio
PK FK	cedula_paciente	NUMBER(38)	Cédula de ciudadanía del paciente	
PK FK	cedula_vinculo	NUMBER(38)	Cédula de ciudadanía que identifica a la persona vinculada	

Tabla Alergias_paciente

Llaves	Campo	Tipo	Descripción	Dominio
PK FK	alergia_id	NUMBER(38)	Identificador de la alergia	
PK FK	cedula	NUMBER(38)	Cédula de ciudadanía del paciente	

Tabla cita

Llaves	Campo	Tipo	Descripción	Dominio
PK	cita_id	VARCHAR2(255)	Identificador de la cita	
	fecha_programacion	DATE	Fecha en la cual se programó la cita	NOT NULL
	hora_programacion	VARCHAR2(255)	Hora en la que se programó la cita	NOT NULL
	fecha_realizacion	DATE	Fecha en la que se realizó la cita	NOT NULL
	hora_realizacion	VARCHAR2(255)	Hora en la que se realizó la cita	NOT NULL
FK	motivo_id	VARCHAR2(255)	Identificador del motivo de la cita	
FK	cedula_paciente	NUMBER(38)	Cédula de ciudadanía del paciente	
FK	cedula_miembro_personal	NUMBER(38)	Cédula de ciudadanía del miembro del personal	

Tabla Reclamo

Llaves	Campo	Tipo	Descripción	Dominio
PK	codigo_reclamo	VARCHAR2(255)	Código que identifica el reclamo	
	fecha_accion	DATE	Fecha en la que se realizó la acción a reclamar	NOT NULL
	fecha_reclamo	DATE	Fecha del reclamo	NOT NULL
	monto_reclamado	FLOAT(126)	Monto del reclamo	NOT NULL
FK	cita_id	VARCHAR2(255)	Identificador de la cita asociada al reclamo	

Tabla Reclamos_Consecutivos

Llaves	Campo	Tipo	Descripción	Dominio
PK FK	codigo_reclamo_anterior	VARCHAR2(255)	Código que identifica el reclamo	
PK FK	codigo_reclamo_siguiente	VARCHAR2(255)	Código que identifica el reclamo	

Tabla Reclamo_Aprobado

Llaves	Campo	Tipo	Descripción	Dominio
PK FK	codigo_reclamo	VARCHAR2(255)	Código que identifica el reclamo	
	monto_pagado	FLOAT(126)	Monto pagado por la poliza respecto a el reclamo	

Tabla Reclamo_Parcial

Llaves	Campo	Tipo	Descripción	Dominio
PK FK	codigo_reclamo	VARCHAR2(255)	Código que identifica el reclamo	
	monto_pagado	FLOAT(126)	Monto pagado por la poliza respecto a el reclamo	NOT NULL
	fecha_pago_parcial	DATE	Fecha en la que se efectuó el pago parcial	NOT NULL
FK	codigo_motivo_no_pago	VARCHAR2(255)	Identificador del motivo para no pago total	

Tabla Reclamo_Negado

Llaves	Campo	Tipo	Descripción	Dominio
PK FK	codigo_reclamo	VARCHAR2(255)	Código que identifica el reclamo	

Tabla Etapa_Reclamo_Negado

Llaves	Campo	Tipo	Descripción	Dominio
PK FK	codigo_reclamo	VARCHAR2(255)	Código que identifica el reclamo	
PK FK	etapa_id	VARCHAR2(255)	Identificador de la etapa	

Tabla Etapa_Reclamo_Parcial

Llaves	Campo	Tipo	Descripción	Dominio
PK FK	codigo_reclamo	VARCHAR2(255)	Código que identifica el reclamo	
PK FK	etapa_id	VARCHAR2(255)	Identificador de la etapa	

Tabla Poliza

Llaves	Campo	Tipo	Descripción	Dominio
PK	codigo_poliza	VARCHAR2(255)	identificador de la póliza	
FK	nit	VARCHAR2(255)	Número de Identificación Tributaria de la compañía de seguros	
FK	cedula	NUMBER(38)	Cédula de ciudadanía del paciente	
FK	codigo_reclamo	VARCHAR2(255)	Código que identifica el reclamo	

Tabla Cobertura_poliza

Llaves	Campo	Tipo	Descripción	Dominio
PK FK	cobertura_id	VARCHAR2(255)	Identificador de la cobertura	
PK FK	codigo_poliza	VARCHAR2(255)	Identificador de la póliza	

Consultas**-- Primera consulta**

CREATE VIEW Reclamaciones_Paciente AS

SELECT

p.nombre AS "Nombre Paciente",

p.sexo AS "Genero Paciente",

COALESCE(COUNT(DISTINCT c.cita_id), 0) AS "Número de Citas",

COALESCE(SUM(r.monto_reclamado), 0) AS "Valor Total Monto Reclamado",

COALESCE(SUM(COALESCE(ra.monto_pagado, 0) + COALESCE(rp.monto_pagado, 0)), 0) AS

"Valor Total Monto Pagado",

COALESCE(SUM(r.monto_reclamado) - SUM(COALESCE(ra.monto_pagado, 0) +

COALESCE(rp.monto_pagado, 0)), 0) AS "Diferencia Reclamado-Pagado"

FROM

Paciente p

LEFT JOIN

Cita c ON p.cedula = c.cedula_paciente

LEFT JOIN

Reclamo r ON c.cita_id = r.cita_id

LEFT JOIN

Reclamo_Aprobado ra ON r.codigo_reclamo = ra.codigo_reclamo

LEFT JOIN

Reclamo_Parcial rp ON r.codigo_reclamo = rp.codigo_reclamo

GROUP BY

p.nombre, p.sexo

UNION ALL

SELECT

```
'Total' AS "Nombre Paciente",
NULL AS "Genero Paciente",
COUNT(DISTINCT c.cita_id) AS "Número de Citas",
COALESCE(SUM(r.monto_reclamado), 0) AS "Valor Total Monto Reclamado",
COALESCE(SUM(COALESCE(ra.monto_pagado, 0) + COALESCE(rp.monto_pagado, 0)), 0) AS
"Valor Total Monto Pagado",
COALESCE(SUM(r.monto_reclamado) - SUM(COALESCE(ra.monto_pagado, 0) +
COALESCE(rp.monto_pagado, 0)), 0) AS "Diferencia Reclamado-Pagado"
```

FROM

Paciente p

LEFT JOIN

Cita c ON p.cedula = c.cedula_paciente

LEFT JOIN

Reclamo r ON c.cita_id = r.cita_id

LEFT JOIN

Reclamo_Aprobado ra ON r.codigo_reclamo = ra.codigo_reclamo

LEFT JOIN

Reclamo_Parcial rp ON r.codigo_reclamo = rp.codigo_reclamo;

<pre>SELECT p.nombre AS "Nombre Paciente", p.sexo AS "Genero Paciente", COALESCE(COUNT(DISTINCT c.cita_id), 0) AS "Número de Citas", COALESCE(SUM(r.monto_reclamado), 0) AS "Valor Total Monto Reclamado", COALESCE(SUM(COALESCE(ra.monto_pagado, 0) + COALESCE(rp.monto_pagado, 0)), 0) AS "Valor Total Monto Pagado", COALESCE(SUM(r.monto_reclamado) - SUM(COALESCE(ra.monto_pagado, 0) + COALESCE(rp.monto_pagado, 0)), 0) AS "Diferencia Reclamado-Pagado" FROM Paciente p LEFT JOIN Cita c ON p.cedula = c.cedula_paciente LEFT JOIN Reclamo r ON c.cita_id = r.cita_id LEFT JOIN Reclamo_Aprobado ra ON r.codigo_reclamo = ra.codigo_reclamo LEFT JOIN Reclamo_Parcial rp ON r.codigo_reclamo = rp.codigo_reclamo GROUP BY p.nombre, p.sexo UNION ALL SELECT 'Total' AS "Nombre Paciente", NULL AS "Genero Paciente", COUNT(DISTINCT c.cita_id) AS "Número de Citas", COALESCE(SUM(r.monto_reclamado), 0) AS "Valor Total Monto Reclamado", COALESCE(SUM(COALESCE(ra.monto_pagado, 0) + COALESCE(rp.monto_pagado, 0)), 0) AS "Valor Total Monto Pagado", COALESCE(SUM(r.monto_reclamado) - SUM(COALESCE(ra.monto_pagado, 0) + COALESCE(rp.monto_pagado, 0)), 0) AS "Diferencia Reclamado-Pagado" FROM Paciente p LEFT JOIN Cita c ON p.cedula = c.cedula_paciente LEFT JOIN Reclamo r ON c.cita_id = r.cita_id LEFT JOIN Reclamo_Aprobado ra ON r.codigo_reclamo = ra.codigo_reclamo LEFT JOIN Reclamo_Parcial rp ON r.codigo_reclamo = rp.codigo_reclamo;</pre>					
Script Output x Query Result x					
All Rows Fetched: 5 in 0,102 seconds					
Nombre Paciente	Genero Paciente	Número de Citas	Valor Total Monto Reclamado	Valor Total Monto Pagado	Diferencia Reclamado-Pagado
1 Carlos Ferrer	Masculino	0	0	0	0
2 Mapa Ruiz	Femenino	1	24000	0	24000
3 Laura Martinez	Femenino	1	7000	3000	4000
4 Carlos Pérez	Masculino	1	5000	5000	0
5 Total	(null)	3	36000	8000	28000

	Nombre Paciente	Genero Paciente	Número de Citas	Valor Total Monto Reclamado	Valor Total Monto Pagado	Diferencia Reclamado-Pagado
1	Carlos Ferrer	Masculino	0	0	0	0
2	Mapa Ruiz	Femenino	1	24000	0	24000
3	Laura Martínez	Femenino	1	7000	3000	4000
4	Carlos Pérez	Masculino	1	5000	5000	0
5	Total	(null)	3	36000	8000	28000

-- Segunda consulta

CREATE VIEW Reclamaciones_No_Reembolsadas AS

SELECT

mp.cargo AS "Clase",

SUM(CASE WHEN mp.sexo = 'Masculino' AND (r.monto_reclamado - COALESCE(rp.monto_pagado, 0)) <> 0 THEN 1 ELSE 0 END) AS "Número de reclamaciones masculino",

SUM(CASE WHEN mp.sexo = 'Femenino' AND (r.monto_reclamado - COALESCE(rp.monto_pagado, 0)) <> 0 THEN 1 ELSE 0 END) AS "Número de reclamaciones femenino",

SUM(CASE WHEN (r.monto_reclamado - COALESCE(rp.monto_pagado, 0)) <> 0 THEN r.monto_reclamado - COALESCE(rp.monto_pagado, 0) ELSE 0 END) AS "Valor de reclamaciones no pagadas totalmente"

FROM

Reclamo_Parcial rp

LEFT JOIN

Reclamo r ON rp.codigo_reclamo = r.codigo_reclamo

LEFT JOIN

Cita c ON r.cita_id = c.cita_id

LEFT JOIN

Miembro_Personal mp ON c.cedula_miembro_personal = mp.cedula

GROUP BY

mp.cargo;

<pre>SELECT mp.cargo AS "Clase", SUM(CASE WHEN mp.sexo = 'Masculino' AND (r.monto_reclamado - COALESCE(rp.monto_pagado, 0)) <> 0 THEN 1 ELSE 0 END) AS "Número de reclamaciones masculino", SUM(CASE WHEN mp.sexo = 'Femenino' AND (r.monto_reclamado - COALESCE(rp.monto_pagado, 0)) <> 0 THEN 1 ELSE 0 END) AS "Número de reclamaciones femenino", SUM(CASE WHEN (r.monto_reclamado - COALESCE(rp.monto_pagado, 0)) <> 0 THEN r.monto_reclamado - COALESCE(rp.monto_pagado, 0) ELSE 0 END) AS "Valor de reclamaciones no pagadas totalmente" FROM Reclamo_Parcial rp LEFT JOIN Reclamo r ON rp.codigo_reclamo = r.codigo_reclamo LEFT JOIN Cita c ON r.cita_id = c.cita_id LEFT JOIN Miembro_Personal mp ON c.cedula_miembro_personal = mp.cedula GROUP BY mp.cargo;</pre>			
Script Output x Query Result x			
All Rows Fetched: 1 in 0,021 seconds			
Clase	Número de reclamaciones masculino	Número de reclamaciones femenino	Valor de reclamaciones no pagadas totalmente
1 Enfermera	0	1	4000

Clase	Número de reclamaciones masculino	Número de reclamaciones femenino	Valor de reclamaciones no pagadas totalmente
1 Enfermera	0	1	4000

-- Tercera consulta

CREATE VIEW Paciente_Personas_Relacionadas AS

SELECT

p.nombre AS "Nombre Paciente",
COALESCE(v.nombre, 'N/A') AS "Nombre Persona Relacionada",
COALESCE(v.tipo, 'N/A') AS "Naturaleza de la Relación"

FROM

Paciente p

LEFT JOIN

Vinculo_Paciente vp ON p.cedula = vp.cedula_paciente

LEFT JOIN

Vinculo v ON vp.cedula_vinculo = v.cedula;

```
SELECT
    p.nombre AS "Nombre Paciente",
    COALESCE(v.nombre, 'N/A') AS "Nombre Persona Relacionada",
    COALESCE(v.tipo, 'N/A') AS "Naturaleza de la Relación"
FROM
    Paciente p
LEFT JOIN
    Vinculo_Paciente vp ON p.cedula = vp.cedula_paciente
LEFT JOIN
    Vinculo v ON vp.cedula_vinculo = v.cedula;
```

Script Output x Query Result x

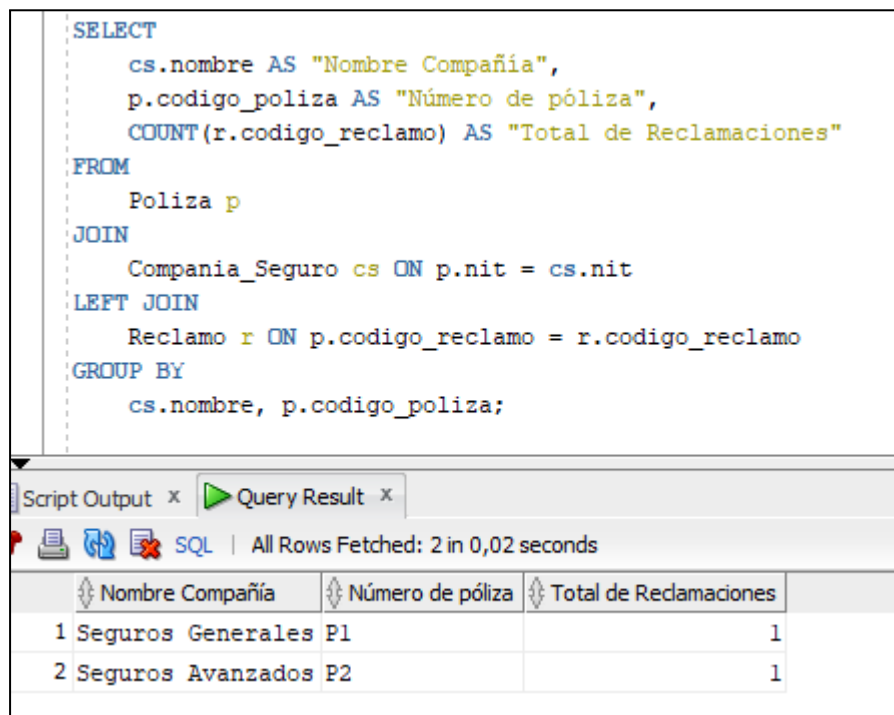
SQL | All Rows Fetched: 4 in 0,03 seconds

	Nombre Paciente	Nombre Persona Relacionada	Naturaleza de la Relación
1	Carlos Pérez	Juan Erasmo	Familiar
2	Laura Martínez	María Gómez	Amiga
3	Carlos Ferrer	Juan Rozo	Mejor Amigo
4	Mapa Ruiz	N/A	N/A

Nombre Paciente	Nombre Persona Relacionada	Naturaleza de la Relación
Carlos Pérez	Juan Erasmo	Familiar
Laura Martínez	María Gómez	Amiga
Carlos Ferrer	Juan Rozo	Mejor Amigo
Mapa Ruiz	N/A	N/A

-- Cuarta consulta

```
CREATE VIEW Reclamaciones_Poliza_Compania AS
SELECT
    cs.nombre AS "Nombre Compañía",
    p.codigo_poliza AS "Número de póliza",
    COUNT(r.codigo_reclamo) AS "Total de Reclamaciones"
FROM
    Poliza p
JOIN
    Compania_Seguro cs ON p.nit = cs.nit
LEFT JOIN
    Reclamo r ON p.codigo_reclamo = r.codigo_reclamo
GROUP BY
    cs.nombre, p.codigo_poliza;
```



The screenshot shows a SQL IDE interface. The top pane displays the SQL query for creating a view. The bottom pane shows the 'Query Result' tab with a table of 2 rows and 3 columns. The status bar indicates 'All Rows Fetched: 2 in 0,02 seconds'.

	Nombre Compañía	Número de póliza	Total de Reclamaciones
1	Seguros Generales P1		1
2	Seguros Avanzados P2		1

Nombre Compañía	Número de póliza	Total de Reclamaciones
Seguros Generales P1		1
Seguros Avanzados P2		1

-- Quinta consulta

CREATE VIEW Etapas_Reclamo AS

SELECT

 r.codigo_reclamo AS "ID de Reclamo",

 COUNT(DISTINCT ep.etapa_id) + COUNT(DISTINCT en.etapa_id) AS "Número Total de Etapas de Negociación",

 rc.codigo_reclamo_siguiente AS "ID de Nuevo Reclamo"

FROM

 Reclamo r

LEFT JOIN

 Etapas_Reclamo_Parcial ep ON r.codigo_reclamo = ep.codigo_reclamo

LEFT JOIN

 Etapas_Reclamo_Negado en ON r.codigo_reclamo = en.codigo_reclamo

LEFT JOIN

 Reclamos_Consecutivos rc ON r.codigo_reclamo = rc.codigo_reclamo_anterior

GROUP BY

 r.codigo_reclamo, rc.codigo_reclamo_siguiente;

<pre>SELECT r.codigo_reclamo AS "ID de Reclamo", COUNT(DISTINCT ep.etapa_id) + COUNT(DISTINCT en.etapa_id) AS "Número Total de Etapas de Negociación", rc.codigo_reclamo_siguiente AS "ID de Nuevo Reclamo" FROM Reclamo r LEFT JOIN Etapas_Reclamo_Parcial ep ON r.codigo_reclamo = ep.codigo_reclamo LEFT JOIN Etapas_Reclamo_Negado en ON r.codigo_reclamo = en.codigo_reclamo LEFT JOIN Reclamos_Consecutivos rc ON r.codigo_reclamo = rc.codigo_reclamo_anterior GROUP BY r.codigo_reclamo, rc.codigo_reclamo_siguiente; -- Dar acceso a todas las vistas</pre>	
Script Output x Query Result x	
All Rows Fetched: 4 in 0,035 seconds	
ID de Reclamo	Número Total de Etapas de Negociación ID de Nuevo Reclamo
1 R1	0 (null)
2 R2	1 (null)
3 R3	2 R4
4 R4	0 (null)

ID de Reclamo	Número Total de Etapas de Negociación	ID de Nuevo Reclamo
R1		0 (null)
R2		1 (null)
R3		2 R4
R4		0 (null)