



UNIVERSIDAD DE BUENOS AIRES

Departamento de Computación

Facultad de Ciencias Exactas y Naturales

Bases de Datos

Trabajo Práctico 1

9 de agosto de 2015

Integrante	LU	Correo electrónico
Maurizio, Miguel Sebastián	635/11	miguelmaurizio.92@gmail.com
Prillo, Sebastián	616/11	sebastianprillo@gmail.com
Tagliavini Ponce, Guido	783/11	guido.tag@gmail.com

Índice

1. Introducción	1
2. Diagrama Entidad Relación	2
2.1. Restricciones en lenguaje natural	4
3. Modelo Relacional	4
4. Implementación	10

1. Introducción

En este trabajo práctico diseñamos e implementamos una base de datos para el Registro Único de Accidentes de Tránsito (RUAT), sistema que está que está preparando el Gobierno Nacional, para registrar y analizar información sobre accidentes e infracciones de tránsito ocurridos en el país.

Por un lado, el sistema registra todos los datos relacionados con siniestros de tránsito, lo cual abarca:

- vehículos involucrados;
- conductores involucrados;
- testigos;
- localización;
- modalidad del siniestro (atropello, vuelco, etc.);
- tipo de colisión;
- denuncia radicada por el hecho;
- estudios y peritajes hechos.

Por otro lado, registra infracciones de tránsito, y más específicamente:

- vehículo involucrado;
- conductor involucrado;
- localización;
- tipo de infracción;

Además, el sistema registra datos sobre los vehículos, personas y las vías nacionales. Sobre los vehículos, almacena:

- categoría de coche (gama media, gama alta, etc.);
- tipo de vehículo (auto, camión, moto, etc.);
- seguro automotor y su tipo;

Sobre las personas, almacena:

- datos personales;
- autos de los cuales es dueña;
- cédulas (verdes y azules) que posee;
- licencias de conducir que posee;
- antecedentes penales;

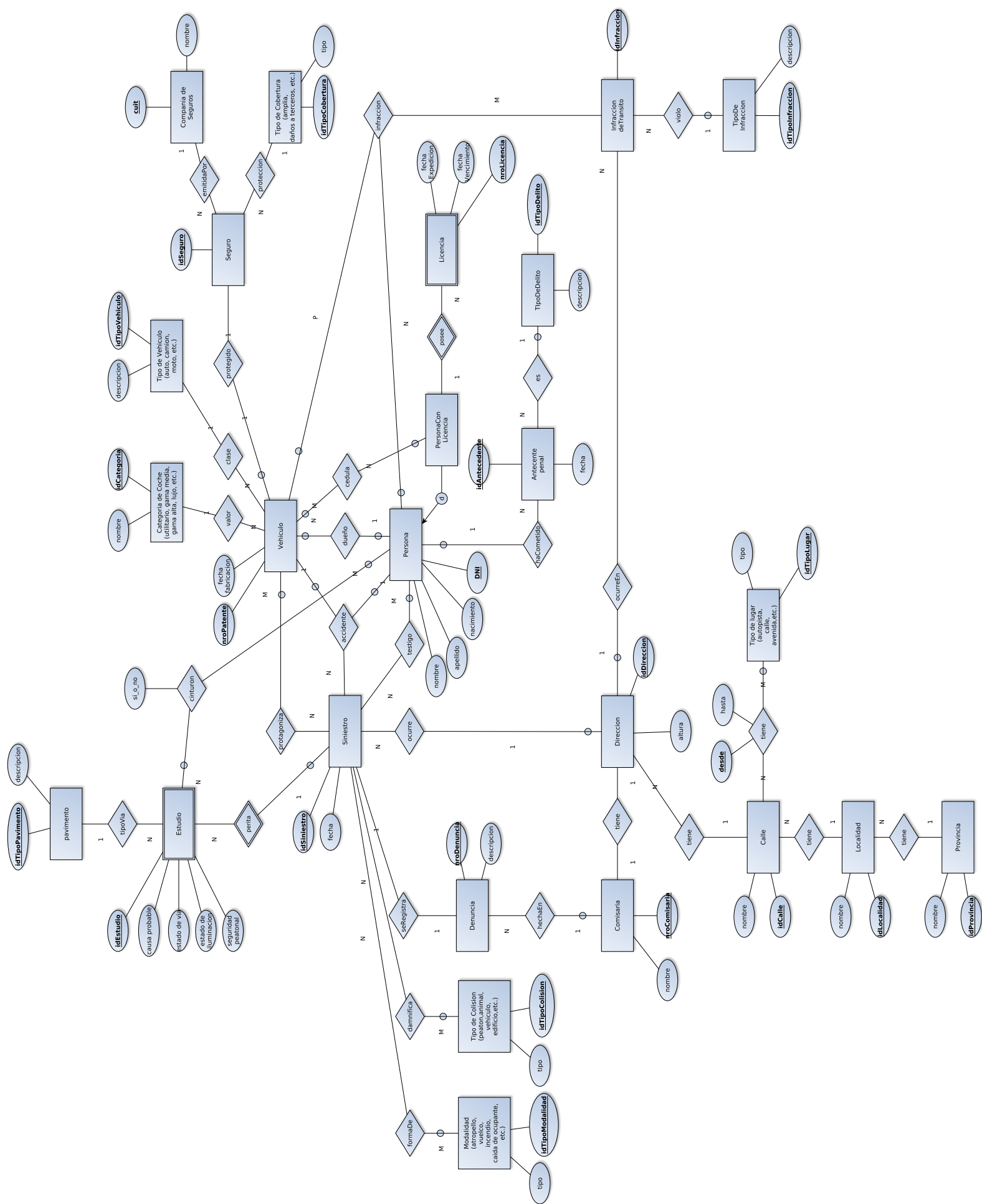
Finalmente, sobre las vías nacionales, el sistema almacena:

- tipo de vía según el tramo (calle, avenida, autopista, etc.);
- extensión del tramo;

La elicitación de estos requerimientos proviene, principalmente, de la lectura del enunciado del trabajo práctico, que contiene toda esta información.

2. Diagrama Entidad Relación

A partir de los requerimientos recabados, elaboramos un diagrama de entidad-relación, que presentamos a continuación.



2.1. Restricciones en lenguaje natural

Las restricciones que el DER no puede capturar son las siguientes:

1. Las personas relacionadas con un estudio, deben estar involucradas en el siniestro correspondiente a ese estudio.
2. Si un vehículo sin conductor forma parte de un siniestro (es decir, está relacionado con un siniestro vía la relación binaria *protagoniza*), entonces no era conducido por nadie en ese siniestro (es decir, no aparece en la relación ternaria *accidente* con ese siniestro).
3. Las personas que aparecen relacionadas con el estudio en la relación *cinturon*, deben ser personas que participaron del siniestro correspondiente al estudio, conduciendo uno de los vehículos de ese siniestro.
4. Cada persona aparece a lo sumo una vez por siniestro. En otras palabras, un siniestro no puede estar relacionado dos veces con la misma persona vía la ternaria *accidente*.
5. Cada vehículo aparece a lo sumo una vez por siniestro. En otras palabras, un siniestro no puede estar relacionado dos veces con el mismo vehículo vía la ternaria *accidente*.
6. No hay solapamiento entre los rangos de alturas de las calles. Por ejemplo, no puede ser que Monroe del 0 al 3000 sea calle y del 2500 al 4000 sea avenida.

3. Modelo Relacional

A partir del DER, derivamos el siguiente modelo relacional. Lo presentamos directamente en sintaxis SQL.

Algunas de las relaciones N-M del DER tenían nombres poco ilustrativos al ser sacados de contexto. Por ejemplo, la relación binaria entre *Siniestro* y *Tipo de Colisión* fue llamada *damnifica*, nombre que no tiene mucho sentido aisladamente. Esto representaba un problema, dado que en, el MR, esas relaciones pasan a ser tablas. En estos casos, decidimos transformar el nombre de estas relaciones, usando los nombres de las entidades relacionadas, del siguiente modo. Una relación de nombre *R* entre dos entidades de nombres *X* e *Y*, se pasa a llamar *X_R_Y*. Por ejemplo, en el caso de la relación *damnifica*, el nombre de la tabla asociada pasó a ser *siniestro_damnifica_tipo_de_colision*.

-- Tabla Provincia

```
CREATE TABLE provincia (  
  idProvincia INTEGER NOT NULL,  
  nombre VARCHAR(255) NOT NULL,  
  PRIMARY KEY(idProvincia)  
);
```

-- Tabla Localidad

```
CREATE TABLE localidad (  
  idLocalidad INTEGER NOT NULL,  
  nombre VARCHAR(255) DEFAULT NULL,  
  idProvincia INTEGER NOT NULL,  
  PRIMARY KEY(idLocalidad),  
  FOREIGN KEY(idProvincia) REFERENCES provincia(idProvincia)  
);
```

-- Tabla Calle

```
CREATE TABLE calle (  

```

```

idCalle INTEGER NOT NULL,
nombre VARCHAR(255) DEFAULT NULL,
idLocalidad INTEGER NOT NULL,
PRIMARY KEY(idCalle),
FOREIGN KEY(idLocalidad) REFERENCES localidad(idLocalidad)
);

-- Tabla Direccion
CREATE TABLE direccion (
idDireccion INTEGER NOT NULL,
altura INTEGER NOT NULL,
idCalle INTEGER NOT NULL,
PRIMARY KEY(idDireccion),
FOREIGN KEY(idCalle) REFERENCES calle(idCalle)
);

-- Tabla Tipo de Lugar
CREATE TABLE tipo_de_lugar (
idTipoLugar INTEGER NOT NULL,
tipo VARCHAR(255) DEFAULT NULL,
PRIMARY KEY(idTipoLugar)
);

-- Tabla relacion tiene, entre Calle y Tipo de lugar
CREATE TABLE calle_tiene_tipo_de_lugar (
idCalle INTEGER NOT NULL,
idTipoLugar INTEGER NOT NULL,
desde INTEGER NOT NULL,
hasta INTEGER NOT NULL,
PRIMARY KEY(idCalle, idTipoLugar, desde),
FOREIGN KEY(idCalle) REFERENCES calle(idCalle),
FOREIGN KEY(idTipoLugar) REFERENCES tipo_de_lugar(idTipoLugar)
);

-- Tabla Comisaria
CREATE TABLE comisaria (
nroComisaria INTEGER NOT NULL,
nombre VARCHAR(255) DEFAULT NULL,
idDireccion INTEGER NOT NULL,
PRIMARY KEY(nroComisaria),
FOREIGN KEY(idDireccion) REFERENCES direccion(idDireccion)
);

-- Tabla Denuncia
CREATE TABLE denuncia (
nroDenuncia INTEGER NOT NULL,
descripcion VARCHAR(255) DEFAULT NULL,
nroComisaria INTEGER NOT NULL,
PRIMARY KEY(nroDenuncia),
FOREIGN KEY(nroComisaria) REFERENCES comisaria(nroComisaria)
);

```

```

-- Tabla Tipo de Colision
CREATE TABLE tipo_de_colision (
    idTipoColision INTEGER NOT NULL,
    tipo VARCHAR(255) DEFAULT NULL,
    PRIMARY KEY(idTipoColision)
);

-- Tabla Modalidad
CREATE TABLE modalidad (
    idTipoModalidad INTEGER NOT NULL,
    tipo VARCHAR(255) DEFAULT NULL,
    PRIMARY KEY(idTipoModalidad)
);

-- Tabla Siniestro
CREATE TABLE siniestro (
    idSiniestro INTEGER NOT NULL,
    fecha DATETIME DEFAULT NULL,
    nroDenuncia INTEGER NOT NULL,
    idDireccion INTEGER NOT NULL,
    PRIMARY KEY(idSiniestro),
    FOREIGN KEY(nroDenuncia) REFERENCES denuncia(nroDenuncia),
    FOREIGN KEY(idDireccion) REFERENCES direccion(idDireccion)
);

-- Tabla relacion damnifica, entre Siniestro y Tipo de Colision
CREATE TABLE siniestro_damnifica_tipo_de_colision (
    idSiniestro INTEGER NOT NULL,
    idTipoColision INTEGER NOT NULL,
    PRIMARY KEY(idSiniestro, idTipoColision),
    FOREIGN KEY(idSiniestro) REFERENCES siniestro(idSiniestro),
    FOREIGN KEY(idTipoColision) REFERENCES tipo_de_colision(idTipoColision)
);

-- Tabla relacion formaDe, entre Siniestro y Modalidad
CREATE TABLE siniestro_forma_de_modalidad (
    idSiniestro INTEGER NOT NULL,
    idTipoModalidad INTEGER NOT NULL,
    PRIMARY KEY(idSiniestro, idTipoModalidad),
    FOREIGN KEY(idSiniestro) REFERENCES siniestro(idSiniestro),
    FOREIGN KEY(idTipoModalidad) REFERENCES modalidad(idTipoModalidad)
);

-- Tabla Pavimento
CREATE TABLE pavimento (
    idPavimento INTEGER NOT NULL,
    descripcion VARCHAR(255) DEFAULT NULL,
    PRIMARY KEY(idPavimento)
);

```



```

-- Tabla Estudio
CREATE TABLE estudio (
  idEstudio INTEGER NOT NULL,
  causaProbable VARCHAR(255) DEFAULT NULL,
  estadoVia VARCHAR(255) DEFAULT NULL,
  estadoIluminacion VARCHAR(255) DEFAULT NULL,
  seguridadPeatonal BOOLEAN DEFAULT NULL,
  idTipoPavimento INTEGER NOT NULL,
  idSiniestro INTEGER NOT NULL,
  PRIMARY KEY (idEstudio, idSiniestro),
  FOREIGN KEY(idTipoPavimento) REFERENCES pavimento(idTipoPavimento),
  FOREIGN KEY(idSiniestro) REFERENCES siniestro(idSiniestro)
);

-- Tabla Persona
CREATE TABLE persona (
  dni INTEGER NOT NULL,
  nombre VARCHAR(255) NOT NULL,
  apellido VARCHAR(255) NOT NULL,
  nacimiento DATE NOT NULL,
  PRIMARY KEY(dni)
);

-- Tabla relacion Testigo, entre Siniestro y Persona
CREATE TABLE siniestro_testigo_persona (
  idSiniestro INTEGER NOT NULL,
  dni INTEGER NOT NULL,
  PRIMARY KEY(idSiniestro, dni),
  FOREIGN KEY(idSiniestro) REFERENCES siniestro(idSiniestro),
  FOREIGN KEY(dni) REFERENCES persona(dni)
);

-- Tabla relacion Cinturon, entre Estudio y Persona
CREATE TABLE estudio_cinturon_persona (
  idEstudio INTEGER NOT NULL,
  dni INTEGER NOT NULL,
  si_o_no BOOLEAN NOT NULL,
  PRIMARY KEY(idEstudio, dni),
  FOREIGN KEY(idEstudio) REFERENCES estudio(idEstudio),
  FOREIGN KEY(dni) REFERENCES persona(dni)
);

-- Tabla Tipo de Delito
CREATE TABLE tipo_de_delito (
  idTipoDelito INTEGER NOT NULL,
  descripcion VARCHAR(255) DEFAULT NULL,
  PRIMARY KEY(idTipoDelito)
);

-- Tabla Antecedente penal
CREATE TABLE antecedente_penal (

```

```

idAntecedente INTEGER NOT NULL,
fecha Date NOT NULL,
dni INTEGER NOT NULL,
idTipoDelito INTEGER NOT NULL,
FOREIGN KEY(dni) REFERENCES persona(dni),
FOREIGN KEY(idTipoDelito) REFERENCES tipo_de_delito(idTipoDelito),
PRIMARY KEY(idAntecedente)
);

-- Tabla Tipo Infraccion
CREATE TABLE tipo_de_infraccion (
    idTipoInfraccion INTEGER NOT NULL,
    descripcion VARCHAR(255) DEFAULT NULL,
    PRIMARY KEY(idTipoInfraccion)
);

-- Tabla Antecedente penal
CREATE TABLE infraccion_de_transito (
    idInfraccion INTEGER NOT NULL,
    idDireccion INTEGER NOT NULL,
    idTipoInfraccion INTEGER NOT NULL,
    FOREIGN KEY(idDireccion) REFERENCES direccion(idDireccion),
    FOREIGN KEY(idTipoInfraccion) REFERENCES tipo_infraccion(idTipoInfraccion),
    PRIMARY KEY(idInfraccion)
);

-- Tabla Persona con Licencia
CREATE TABLE persona_con_licencia (
    dni INTEGER NOT NULL,
    FOREIGN KEY(dni) REFERENCES persona(dni),
    PRIMARY KEY(dni)
);

-- Licencia
CREATE TABLE licencia (
    nroLicencia INTEGER NOT NULL,
    dni INTEGER NOT NULL,
    expedicion DATE NOT NULL,
    expiracion DATE NOT NULL,
    FOREIGN KEY(dni) REFERENCES persona_con_licencia(dni),
    PRIMARY KEY (nroLicencia, dni)
);

-- Tabla Compania de Seguros
CREATE TABLE compania_de_seguro (
    cuit INTEGER NOT NULL,
    nombre VARCHAR(255) NOT NULL,
    PRIMARY KEY(cuit)
);

-- Tabla Tipo de Cobertura

```

```

CREATE TABLE tipo_de_cobertura (
    idTipoCobertura INTEGER NOT NULL,
    descripcion VARCHAR(255) NOT NULL,
    PRIMARY KEY(idTipoCobertura)
);

-- Tabla Tipo Vehiculo
CREATE TABLE tipo_de_vehiculo (
    idTipoVehiculo INTEGER NOT NULL,
    descripcion VARCHAR(255) NOT NULL,
    PRIMARY KEY(idTipoVehiculo)
);

-- Tabla Categoria Vehiculo
CREATE TABLE categoria_de_vehiculo (
    idCategoria INTEGER NOT NULL,
    descripcion VARCHAR(255) NOT NULL,
    PRIMARY KEY(idCategoria)
);

-- Tabla Vehiculo
CREATE TABLE vehiculo (
    nroPatente CHARACTER(6) NOT NULL,
    fechaFabricacion DATE NOT NULL,
    idCategoria INTEGER NOT NULL,
    idTipoVehiculo INTEGER NOT NULL,
    dni INTEGER NOT NULL,
    FOREIGN KEY(idCategoria) REFERENCES categoria_de_vehiculo(idCategoria),
    FOREIGN KEY(idTipoVehiculo) REFERENCES tipo_de_vehiculo(idTipoVehiculo),
    FOREIGN KEY(dni) REFERENCES persona(dni),
    PRIMARY KEY(nroPatente)
);

-- Tabla Seguros
CREATE TABLE seguro (
    idSeguro INTEGER NOT NULL,
    cuit INTEGER NOT NULL,
    idTipoCobertura INTEGER NOT NULL,
    nroPatente INTEGER NOT NULL,
    FOREIGN KEY(cuit) REFERENCES compania_de_seguro(cuit),
    FOREIGN KEY(idTipoCobertura) REFERENCES cobertura(idTipoCobertura),
    FOREIGN KEY(nroPatente) REFERENCES vehiculo(nroPatente),
    PRIMARY KEY(idSeguro)
);

-- Tabla Cedula
CREATE TABLE cedula (
    nroPatente CHARACTER(6) NOT NULL,
    dni INTEGER NOT NULL,
    FOREIGN KEY(nroPatente) REFERENCES vehiculo(nroPatente),
    FOREIGN KEY(dni) REFERENCES persona_con_licencia(dni),

```

```

    PRIMARY KEY(nroPatente, dni)
);

-- Tabla Protagoniza
CREATE TABLE protagoniza (
    nroPatente CHARACTER(6) NOT NULL,
    idSiniestro INTEGER NOT NULL,
    FOREIGN KEY(nroPatente) REFERENCES vehiculo(nroPatente),
    FOREIGN KEY(idSiniestro) REFERENCES siniestro(idSiniestro),
    PRIMARY KEY(nroPatente, idSiniestro)
);

-- Tabla Accidente
CREATE TABLE accidente (
    nroPatente CHARACTER(6) NOT NULL,
    idSiniestro INTEGER NOT NULL,
    dni INTEGER NOT NULL,
    FOREIGN KEY(nroPatente) REFERENCES vehiculo(nroPatente),
    FOREIGN KEY(idSiniestro) REFERENCES siniestro(idSiniestro),
    FOREIGN KEY(dni) REFERENCES persona(dni),
    PRIMARY KEY(nroPatente, idSiniestro)
);

-- Tabla Infraccion
CREATE TABLE infraccion (
    nroPatente CHARACTER(6) NOT NULL,
    dni INTEGER NOT NULL,
    idInfraccion INTEGER NOT NULL,
    FOREIGN KEY(nroPatente) REFERENCES vehiculo(nroPatente),
    FOREIGN KEY(dni) REFERENCES persona(dni),
    FOREIGN KEY(idInfraccion) REFERENCES infraccion_de_transito(idInfraccion),
    PRIMARY KEY(nroPatente, dni, idInfraccion)
);

```

4. Implementación

Implementamos la base de datos usando el motor SQLite. Las tablas mostradas en la sección del modelo relacional son precisamente el código DDL utilizado para definir las tablas en SQLite.

Referencias