



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

# Trabajo Práctico 1

[Primera entrega]

30 de septiembre de 2016

Bases de Datos

Integrante	LU	Correo electrónico
Fernandez, Esteban	691/12	esteban.pmf@gmail.com
Marta, Cristian G.	079/12	cristiangmarta@gmail.com
Podavini Rey, Martín Gastón	483/12	marto.rey2006@gmail.com
Wright, Carolina Rocío	876/12	wright.carolina@gmail.com

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. MER y MR</b>	<b>2</b>
2.1. Modelo de Entidad Relación . . . . .	2
2.2. Modelo Relacional . . . . .	2
<b>3. Diseño Físico</b>	<b>4</b>
3.1. Triggers . . . . .	7
<b>4. Código</b>	<b>9</b>
<b>5. Conclusion</b>	<b>10</b>

## 1. Introducción

El problema a resolver mediante el uso de bases de datos relacional es el del registro de los Casos Criminales en un sitio. Para llevar cuenta de ellos se tiene en cuenta los diversos aspectos que los componen, por ejemplo, los oficiales de policía (quienes pertenecen a sus correspondientes departamentos de policía los cuales fueron modelados también) encargados de documentar y resolver los crímenes junto a las demás personas que participaron de alguna manera (con sus respectivos roles) en el caso que se registra. Estos tendrán información asociada que los identifica y describe.

Los casos, como entidad, pueden encontrarse en diferentes estados (congelado, descartados, resuelto etc) y en cada uno de ellos se proveerá diferente tipo de información ligada al estado en sí del caso. La utilización del modelo relacional nos permitirá administrar esta información de la manera más modular y ordenada posible, permitiéndonos así poder asociar, consultar y mantener la información de las diferentes entidades que conforman el registro de los Casos Criminales de una manera eficiente aprovechando el motor de las bases de datos relacionales SQL.

## 2. MER y MR

### 2.1. Modelo de Entidad Relación

En hoja adjunta podemos ver el MER, que representa las entidades más relevantes del sistema que estamos desarrollando.

### 2.2. Modelo Relacional

En esta sección presentamos el diseño lógico, obtenido a partir del Modelo de Entidad Relación.

Testimonio(id, texto, fecha, caso\_id, persona\_dni, nro\_placa\_policia\_a\_cargo)  
PK = CK = {id}  
FK = {nro\_placa\_policia\_a\_cargo, caso\_id, persona\_dni}

Evidencia(id, fecha\_ingreso, descripcion, fecha\_encuentro, fecha\_sellado, caso\_id)  
PK = CK = {id}  
FK = {caso\_id}

Custodia(id, fecha, localizacion, comentario, evidencia\_id, nro\_placa\_policia\_a\_cargo)  
PK = CK = {id}  
FK = {evidencia\_id, nro\_placa\_policia\_a\_cargo}

Domicilio(id, numero, calle)  
PK = CK = {id}

Caso\_Criminal(id, fecha\_ingreso, fecha, lugar, descripcion, estado, nombre\_categoria)  
PK = CK = {id}  
FK = {nombre\_categoria}

Congelado(caso\_id, fecha, comentario)  
PK = CK = FK = {caso\_id}

*Congelado.caso\_id* tiene que estar en *Caso\_Criminal.id*

Descartado(caso\_id, fecha, motivo)  
PK = CK = FK = {(idCaso)}

*Descartado.caso\_id* tiene que estar en *Caso\_Criminal.id*

Resuelto(caso\_id, fecha, descripcion, nro\_placa\_policia\_cerro)

PK = CK = {caso\_id}

FK = {caso\_id, nro\_placa\_policia\_cerro}

*Resuelto.caso\_id* tiene que estar en *Caso\_Criminal.id*

Pendiente(caso\_id)

PK = CK = FK = {caso\_id}

*Pendiente.caso\_id* tiene que estar en *Caso\_Criminal.id*

Investiga(caso\_id, nro\_placa\_policia, es\_investigador\_principal)

PK = CK = FK = {caso\_id, nro\_placa\_policia}

Categoria(nombre)

PK = CK = {nombre}

Rol(nombre)

PK = CK = {nombre}

Persona(dni, fecha\_nacimiento, nombre, apellido, domicilio\_id)

PK = CK = {dni}

FK = {domicilio\_id}

Telefono(persona\_dni, numero, tipo)

PK = CK = {numero}

FK = {persona\_dni}

Evento(id, descripcion, fecha, caso\_id, persona\_dni)

PK = CK = {id}

FK = {caso\_id, persona\_dni}

Oficial\_De\_Policia(numero\_de\_placa, persona\_dni, fecha\_de\_ingreso, numero\_de\_escritorio, nombre\_rango, nombre\_departamento)

PK = CK = {numero\_de\_placa}

FK = {persona\_dni, nombre\_rango, nombre\_departamento}

Servicio(nombre)

PK = CK = {nombre}

Departamento(nombre, linea\_telefonica, nombre\_localidad, supervisado\_por\_departamento)

PK = CK = {nombre}

FK = {nombre\_localidad, supervisado\_por\_departamento}

Linea\_Telefonica(numero, nombre\_departamento)

PK = CK = {numero, nombre\_departamento}

FK = {nombre\_departamento}

Localidad(nombre)

PK = CK = {nombre}

Rango(nombre)

PK = CK = {Nombre}

Involucra(nombre\_rol, caso\_id, persona\_dni)

PK = CK = FK = {nombre\_rol, caso\_id, persona\_dni}

Culpable(persona\_dni, caso\_id)

PK = CK = FK = {persona\_dni, caso\_id}

### 3. Diseño Físico

Creación de las tablas.

```
CREATE TABLE Domicilio(  
    id integer primary key,  
    numero integer not null,  
    calle varchar(30) not null  
);
```

```
CREATE TABLE Persona(  
    dni integer primary key,  
    fecha_nacimiento date not null,  
    nombre varchar(30) not null,  
    apellido varchar(30) not null,  
    domicilio_id integer not null,  
    foreign key (domicilio_id) references Domicilio(id)  
);
```

```
CREATE TABLE Telefono(  
    numero integer primary key,  
    persona_dni integer not null,  
    tipo varchar(30) not null,  
    foreign key (persona_dni) references Persona(dni)  
);
```

```
CREATE TABLE Localidad(  
    nombre varchar(30) primary key  
);
```

```
CREATE TABLE Departamento(  
    nombre varchar(30) primary key,  
    nombre_localidad varchar(30) not null,  
    supervisado_por_departamento varchar(30),  
    foreign key (nombre_localidad) references Localidad(nombre),  
    foreign key (supervisado_por_departamento) references Departamento(nombre)  
);
```

```
CREATE TABLE Linea_Telefonica(  
    numero int,  
    nombre_departamento varchar(30),
```

```

    PRIMARY KEY (numero, nombre_departamento),
    foreign key (nombre_departamento) references Departamento(nombre)
);

CREATE TABLE Rango(
    nombre varchar(30) primary key
);

CREATE TABLE Servicio(
    nombre varchar(30) primary key
);

CREATE TABLE Oficial_De_Policia(
    persona_dni integer not null,
    numero_de_placa integer primary key,
    fecha_de_ingreso date not null,
    numero_de_escritorio integer not null,
    nombre_rango varchar(30) not null,
    nombre_departamento varchar(30) not null,
    foreign key (persona_dni) references Persona(dni),
    foreign key (nombre_rango) references Rango(nombre),
    foreign key (nombre_departamento) references Departamento(nombre)
);

CREATE TABLE Categoria(
    nombre varchar(30) primary key
);

CREATE TABLE Rol(
    nombre varchar(30) primary key
);

CREATE TABLE Caso_Criminal(
    id integer primary key,
    fecha_ingreso date not null,
    fecha datetime not null,
    lugar varchar(30) not null,
    descripcion varchar(255) not null,
    nombre_categoria varchar(30) not null,
    estado varchar(30) not null,
    foreign key (nombre_categoria) references Categoria(nombre),
    CHECK (fecha_ingreso > fecha)
);

CREATE TABLE Involucra(
    caso_id integer,
    persona_dni integer,
    nombre_rol varchar(30) not null,

    PRIMARY KEY (caso_id, persona_dni),
    foreign key (caso_id) references Caso_Criminal(id),
    foreign key (persona_dni) references Persona(dni),
    foreign key (nombre_rol) references Rol(nombre)
);

```

```

CREATE TABLE Evento(
    id integer primary key,
    caso_id integer,
    persona_dni integer,
    descripcion varchar(255) not null,
    fecha datetime not null,

    FOREIGN KEY (caso_id, persona_dni) REFERENCES Involucra (caso_id, persona_dni)
);

CREATE TABLE Evidencia(
    id integer PRIMARY KEY,
    fecha_ingreso date not null,
    descripcion char(128) not null,
    fecha_encuentro datetime not null,
    fecha_sellado datetime not null,
    caso_id integer not null,
    FOREIGN KEY (caso_id) REFERENCES Caso_Criminal(id),
    CHECK (fecha_ingreso > fecha_encuentro),
    CHECK (fecha_sellado > fecha_ingreso)
);

CREATE TABLE Testimonio(
    id integer PRIMARY KEY,
    caso_id integer,
    persona_dni integer,
    texto varchar(255) not null,
    fecha datetime not null,
    nro_placa_policia_a_cargo integer not null,
    FOREIGN KEY (caso_id, persona_dni) REFERENCES Involucra (caso_id, persona_dni),
    FOREIGN KEY (nro_placa_policia_a_cargo) REFERENCES Oficial_De_Policia(numero_de_placa)
);

CREATE TABLE Custodia(
    id integer PRIMARY KEY,
    fecha datetime not null,
    comentario varchar(255) not null,
    localizacion varchar(255) not null,
    evidencia_id integer not null,
    nro_placa_policia_a_cargo integer not null,
    FOREIGN KEY (evidencia_id) REFERENCES Evidencia(id),
    FOREIGN KEY (nro_placa_policia_a_cargo) REFERENCES Oficial_De_Policia(numero_de_placa)
);

CREATE TABLE Congelado(
    caso_id integer primary key,
    fecha date not null,
    comentario varchar(255) not null,
    FOREIGN KEY (caso_id) references Caso_Criminal(id)
);

CREATE TABLE Descartado(
    caso_id integer primary key,

```

```

    fecha date not null,
    motivo varchar(255) not null,
    FOREIGN KEY (caso_id) references Caso_Criminal(id)
);

CREATE TABLE Resuelto(
    caso_id integer primary key,
    fecha date not null,
    descripcion varchar(255) not null,
    nro_placa_policia_cerro integer not null,
    FOREIGN KEY (caso_id) references Caso_Criminal(id),
    FOREIGN KEY (nro_placa_policia_cerro) REFERENCES Oficial_De_Policia(numero_de_placa)
);

CREATE TABLE Pendiente(
    caso_id integer primary key,
    FOREIGN KEY (caso_id) references Caso_Criminal(id)
);

CREATE TABLE Culpable(
    caso_id integer,
    persona_dni integer,

    PRIMARY KEY (caso_id, persona_dni),
    foreign key (caso_id)      references Resuelto(caso_id),
    foreign key (persona_dni) references Persona(dni)
);

CREATE TABLE Investiga(
    caso_id integer not null,
    nro_placa_policia integer not null,
    es_investigador_principal boolean not null,
    FOREIGN KEY (caso_id) REFERENCES Caso_Criminal(id),
    FOREIGN KEY (nro_placa_policia) REFERENCES Oficial_De_Policia(numero_de_placa)
);

```

### 3.1. Triggers

```

CREATE TRIGGER chk_in_descartado_fecha
BEFORE INSERT ON 'descartado'
FOR EACH ROW
BEGIN
    IF (NEW.fecha < (select fecha from caso_criminal cc where NEW.caso_id = cc.id)) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser menor a la fecha del caso';
    END IF;
END //

CREATE TRIGGER chk_up_descartado_fecha
BEFORE UPDATE ON 'descartado'
FOR EACH ROW
BEGIN
    IF (NEW.fecha < (select fecha from caso_criminal cc where NEW.caso_id = cc.id)) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser menor a la fecha del caso';
    END IF;

```



```
END //
```

```
CREATE TRIGGER chk_in_congelado_fecha
```

```
BEFORE INSERT ON 'congelado'
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    IF (NEW.fecha < (select fecha from caso_criminal cc where NEW.caso_id = cc.id)) THEN
```

```
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser menor a la fecha del caso';
```

```
    END IF;
```

```
END //
```

```
CREATE TRIGGER chk_up_congelado_fecha
```

```
BEFORE UPDATE ON 'congelado'
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    IF (NEW.fecha < (select fecha from caso_criminal cc where NEW.caso_id = cc.id)) THEN
```

```
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser menor a la fecha del caso';
```

```
    END IF;
```

```
END //
```

```
CREATE TRIGGER chk_in_resuelto_fecha
```

```
BEFORE INSERT ON 'resuelto'
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    IF (NEW.fecha < (select fecha from caso_criminal cc where NEW.caso_id = cc.id)) THEN
```

```
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser menor a la fecha del caso';
```

```
    END IF;
```

```
END //
```

```
CREATE TRIGGER chk_up_resuelto_fecha
```

```
BEFORE UPDATE ON 'resuelto'
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    IF (NEW.fecha < (select fecha from caso_criminal cc where NEW.caso_id = cc.id)) THEN
```

```
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser menor a la fecha del caso';
```

```
    END IF;
```

```
END //
```

```
CREATE TRIGGER chk_in_evento_fecha
```

```
BEFORE INSERT ON 'evento'
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    IF (NEW.fecha > (select fecha from caso_criminal cc where NEW.caso_id = cc.id)) THEN
```

```
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser mayor a la fecha del caso';
```

```
    END IF;
```

```
END //
```

```
CREATE TRIGGER chk_up_evento_fecha
```

```
BEFORE UPDATE ON 'evento'
```

```
FOR EACH ROW
```

```
BEGIN
```

```
    IF (NEW.fecha > (select fecha from caso_criminal cc where NEW.caso_id = cc.id)) THEN
```

```
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser mayor a la fecha del caso';
```

```
    END IF;
```

```

END //

CREATE TRIGGER chk_in_testimonio_fecha
BEFORE INSERT ON 'testimonio'
FOR EACH ROW
BEGIN
    IF (NEW.fecha < (select fecha from caso_criminal cc where NEW.caso_id = cc.id)) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser menor a la fecha del caso';
    END IF;
END //

CREATE TRIGGER chk_up_testimonio_fecha
BEFORE UPDATE ON 'testimonio'
FOR EACH ROW
BEGIN
    IF (NEW.fecha < (select fecha from caso_criminal cc where NEW.caso_id = cc.id)) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser menor a la fecha del caso';
    END IF;
END //

CREATE TRIGGER chk_in_custodia_fecha
BEFORE INSERT ON 'custodia'
FOR EACH ROW
BEGIN
    IF (NEW.fecha < (select fecha_ingreso from evidencia e where NEW.evidencia_id = e.id)) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser menor a la fecha de ingreso';
    END IF;
END //

CREATE TRIGGER chk_up_custodia_fecha
BEFORE UPDATE ON 'custodia'
FOR EACH ROW
BEGIN
    IF (NEW.fecha < (select fecha_ingreso from evidencia e where NEW.evidencia_id = e.id)) THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'La fecha no puede ser menor a la fecha de ingreso';
    END IF;
END //

```

## 4. Código

Presentamos las funcionalidades que soporta nuestra base de datos.

- Datos de las personas que fueron sospechosas.

```

select p.dni, p.fecha_nacimiento, p.nombre, p.apellido
from persona p
inner join involucra inv
on p.dni = inv.persona_dni
where inv.nombre_rol ="Sospechoso";

```

- Direcciones donde convivieron personas sospechosas de diferentes casos

```

select id, numero, calle, count(dni) as dniInvolucrados
from domicilio d

```

```

inner join persona p on d.id = p.domicilio_id
inner join involucra inv on inv.persona_dni = p.dni
where inv.nombre_rol = "Sospechoso"
group by d.id
having COUNT(dniInvolucrados) > 1

```

- Oficiales que participaron en la cadena de custodia de evidencias para más de un caso.

```

select numero_de_placa, count(caso_id) as casosInvolucrados
from oficial_de_policia op
inner join custodia c on c.nro_placa_policia_a_cargo = op.numero_de_placa
inner join evidencia e on e.id = c.evidencia_id
group by op.numero_de_placa HAVING casosInvolucrados > 1;

```

- La sucesión de eventos de personas involucradas en un caso.

```

select e.fecha, inv.persona_dni as dni, e.descripcion
from evento e
inner join involucra inv
where inv.persona_dni = e.persona_dni and inv.caso_id = "ID DEL CASO"
order by e.fecha;

```

- Un ranking de oficiales exitosos, es decir los que cerraron mayor cantidad de casos (resueltos).

```

select numero_de_placa, count(caso_id) as nroCasosResueltos
from resuelto
inner join oficial_de_policia on nro_placa_policia_cerro = numero_de_placa
group by numero_de_placa
order by nroCasosResueltos desc;

```

- Las ubicaciones de todas las evidencias de un caso.

```

select e.id as evidencia_id, c.localizacion
from custodia c
inner join evidencia e on e.id = c.evidencia_id
where e.caso_id = "ID DEL CASO" and c.fecha =
(select max(fecha)
from custodia c2
where c2.evidencia_id = e.id);

```

- La lista de oficiales involucrados en un caso.

```

select op.numero_de_placa, inv.nombre_rol
from oficial_de_policia op
inner join persona pe on pe.dni = op.persona_dni
inner join involucra inv on inv.persona_dni = pe.dni
where inv.caso_id = "ID DEL CASO";

```

- Las categorías de casos ordenadas por cantidad de casos

```

select cat.nombre, count(cc.id) as cantidadDeCasos
from categoria cat
inner join caso_criminal cc on cc.nombre_categoria = cat.nombre
group by cat.nombre
order by cantidadDeCasos desc;

```

- Todos los testimonios de un caso dado

```
select t.fecha, t.persona_dni, t.texto
from testimonio t
where t.caso_id = "ID DEL CASO";
```

- Para una categoría en particular listar, para cada uno de los casos, los testimonios asociados

```
select cc.id as casoId, te.texto as Testimonio
from categoria cat
inner join caso_criminal cc on cc.nombre_categoria = cat.nombre
inner join involucra inv on inv.caso_id = cc.id
inner join testimonio te on te.caso_id = inv.caso_id and te.persona_dni = inv.persona_dni
where cat.nombre = "NOMBRE CATEGORIA";
```

## 5. Conclusion

A lo largo del trabajo práctico pudimos aplicar conocimientos adquiridos durante la cursada. Tuvimos como gran desafío la realización del DER, con distintos intercambios de ideas. También pudimos tener un primer acercamiento formal a la creación de una base de datos desde cero.