# Desarrollo de una Base de Datos para Instrumentos Públicos Notariales: Un Enfoque Detallado

## Introducción

El desarrollo de sistemas de información para el ámbito notarial requiere un enfoque meticuloso y detallado, dada la naturaleza crítica de los documentos que gestionan. Los instrumentos públicos notariales son documentos jurídicos de primera importancia que requieren un sistema de gestión de datos robusto, seguro y altamente estructurado. Este documento proporciona una guía exhaustiva para implementar una base de datos relacional basada en el diagrama Entidad-Relación (ER) proporcionado, explicando cada paso del proceso con profundidad y detalle técnico.

## Fase 1: Análisis de Requisitos y Modelado Conceptual

### 1.1 Comprensión del Dominio Notarial

El documento fuente describe la estructura fundamental de los instrumentos públicos notariales en transacciones inmobiliarias, organizada en tres componentes principales:

1. **Encabezamiento**: Metadatos identificativos del documento
2. **Comparecencia**: Personas físicas presentes en el acto
3. **Intervención**: Representación de ausentes o personas jurídicas

Además, introduce conceptos clave como:

* Juicio de capacidad (evaluación de aptitud mental y legal)
* Tipos de representación (voluntaria, orgánica, legal)
* Requisitos de identificación (física y jurídica)
* Consideraciones legales (Ley 10/2010 de prevención de blanqueo)

### 1.2 Identificación de Entidades Principales

Del análisis del documento, identificamos las siguientes entidades principales:

1. **Instrumento Público**: Documento notarial principal
2. **Persona Física**: Individuos que participan (comparecientes o representados)
3. **Persona Jurídica**: Entidades representadas en transacciones
4. **Notario**: Profesional que autoriza el instrumento
5. **Representación**: Mecanismo por el cual se actúa en nombre de otro

### 1.3 Atributos Clave por Entidad

#### Instrumento Público

* Número de protocolo (identificador único)
* Lugar y fecha de otorgamiento
* Notario autorizante (relación)

#### Persona Física

* Identificadores (DNI, NIE, NIF)
* Datos personales (nombre, fecha nacimiento, estado civil)
* Datos de localización (domicilio estructurado)
* Información fiscal y de nacionalidad

#### Persona Jurídica

* Denominación social
* Datos de constitución
* Objeto social y código CNAE
* Identificación fiscal (NIF)

#### Notario

* Datos de identificación
* Colegio notarial
* Datos de habilitación/sustitución

#### Representación

* Tipo (voluntaria, orgánica, legal)
* Atributos de control (veracidad, vigencia, suficiencia)

### 1.4 Relaciones Principales

1. **Instrumento-Encabezamiento**: Relación 1 a 1
2. **Instrumento-Comparecencia**: Relación 1 a N
3. **Instrumento-Intervención**: Relación 1 a N
4. **Persona-Comparecencia**: Relación 1 a N
5. **Representación-Sujeto**: Relación M a 1
6. **Persona Jurídica-Titular Real**: Relación M a N

## Fase 2: Diseño Lógico de la Base de Datos

### 2.1 Transformación del Modelo ER a Esquema Relacional

El diagrama ER se traduce a tablas relacionales siguiendo reglas de mapeo estándar:

1. **Entidades fuertes** se convierten en tablas independientes
2. **Atributos** se convierten en columnas
3. **Relaciones 1:N** se implementan con claves foráneas
4. **Relaciones M:N** requieren tablas de unión
5. **Relaciones 1:1** pueden implementarse con claves foráneas o fusión de tablas

### 2.2 Definición Detallada de Tablas

#### Tabla: INSTRUMENTO\_PUBLICO

CREATE TABLE INSTRUMENTO\_PUBLICO (  
 id\_instrumento INT PRIMARY KEY AUTO\_INCREMENT,  
 numero\_protocolo VARCHAR(20) NOT NULL UNIQUE,  
 lugar\_otorgamiento VARCHAR(255) NOT NULL,  
 fecha\_otorgamiento DATE NOT NULL,  
 id\_notario INT NOT NULL,  
 FOREIGN KEY (id\_notario) REFERENCES NOTARIO(id\_notario)  
);

**Explicación**:

* numero\_protocolo es único por año según requisitos notariales
* Se registra lugar exacto (población, aldea, término municipal)
* Fecha incluye día, mes y año, con hora opcional según casos
* Relación con NOTARIO mediante clave foránea

#### Tabla: ENCABEZAMIENTO

CREATE TABLE ENCABEZAMIENTO (  
 id\_encabezamiento INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_instrumento INT UNIQUE NOT NULL,  
 detalles\_adicionales TEXT,  
 FOREIGN KEY (id\_instrumento) REFERENCES INSTRUMENTO\_PUBLICO(id\_instrumento)  
);

**Explicación**:

* Relación 1:1 con INSTRUMENTO\_PUBLICO
* id\_instrumento es clave foránea y única
* Campos adicionales pueden extenderse según necesidades

#### Tabla: NOTARIO

CREATE TABLE NOTARIO (  
 id\_notario INT PRIMARY KEY AUTO\_INCREMENT,  
 nombre VARCHAR(100) NOT NULL,  
 apellidos VARCHAR(100) NOT NULL,  
 colegio\_notarial VARCHAR(100) NOT NULL,  
 datos\_habilitacion TEXT,  
 datos\_sustitucion TEXT  
);

**Explicación**:

* Almacena información del notario autorizante
* Campos para datos de habilitación/sustitución según requisitos legales
* Relacionada con múltiples instrumentos públicos

#### Tabla: PERSONA\_FISICA

CREATE TABLE PERSONA\_FISICA (  
 id\_persona INT PRIMARY KEY AUTO\_INCREMENT,  
 dni\_nie VARCHAR(20) NOT NULL UNIQUE,  
 nombre VARCHAR(100) NOT NULL,  
 apellido1 VARCHAR(100) NOT NULL,  
 apellido2 VARCHAR(100),  
 fecha\_nacimiento DATE,  
 estado\_civil ENUM('soltero', 'casado', 'separado', 'viudo', 'divorciado'),  
 profesion VARCHAR(100),  
 nif VARCHAR(20) NOT NULL UNIQUE,  
 nacionalidad VARCHAR(50),  
 vecindad\_civil VARCHAR(100),  
 es\_residente BOOLEAN DEFAULT TRUE  
);

**Explicación**:

* dni\_nie como identificador único e inmutable
* Dos apellidos (segundo opcional según normativa española)
* Estado civil con valores predefinidos según Reglamento Notarial
* nif separado de DNI/NIE para fines fiscales
* es\_residente importante para extranjeros

#### Tabla: DOMICILIO

CREATE TABLE DOMICILIO (  
 id\_domicilio INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_persona INT,  
 id\_juridica INT,  
 tipo ENUM('fisica', 'juridica') NOT NULL,  
 pais VARCHAR(100) NOT NULL,  
 provincia VARCHAR(100),  
 municipio VARCHAR(100) NOT NULL,  
 tipo\_via ENUM('calle', 'avenida', 'paseo', 'plaza', 'otro') NOT NULL,  
 nombre\_via VARCHAR(200) NOT NULL,  
 numero VARCHAR(20) NOT NULL,  
 bloque VARCHAR(20),  
 escalera VARCHAR(20),  
 planta VARCHAR(20),  
 puerta VARCHAR(20),  
 email VARCHAR(100),  
 FOREIGN KEY (id\_persona) REFERENCES PERSONA\_FISICA(id\_persona),  
 FOREIGN KEY (id\_juridica) REFERENCES PERSONA\_JURIDICA(id\_juridica),  
 CHECK ((id\_persona IS NOT NULL AND id\_juridica IS NULL) OR   
 (id\_persona IS NULL AND id\_juridica IS NOT NULL))  
);

**Explicación**:

* Estructura normalizada para domicilios (personas físicas y jurídicas)
* Tipos de vía predefinidos para consistencia
* Restricción CHECK asegura que pertenece a una persona o entidad
* Campos opcionales para detalles adicionales

#### Tabla: PERSONA\_JURIDICA

CREATE TABLE PERSONA\_JURIDICA (  
 id\_juridica INT PRIMARY KEY AUTO\_INCREMENT,  
 denominacion\_social VARCHAR(255) NOT NULL UNIQUE,  
 fecha\_constitucion DATE NOT NULL,  
 lugar\_constitucion VARCHAR(255) NOT NULL,  
 objeto\_social TEXT NOT NULL,  
 codigo\_cnae VARCHAR(10),  
 nif VARCHAR(20) NOT NULL UNIQUE,  
 numero\_registro VARCHAR(50) NOT NULL  
);

**Explicación**:

* denominacion\_social única según requisitos registrales
* codigo\_cnae para clasificación económica (4 dígitos)
* nif con estructura específica (letra + 7 dígitos + carácter control)
* numero\_registro del órgano correspondiente

#### Tabla: COMPARECENCIA

CREATE TABLE COMPARECENCIA (  
 id\_comparecencia INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_instrumento INT NOT NULL,  
 id\_persona INT NOT NULL,  
 rol ENUM('otorgante', 'representante', 'testigo', 'traductor', 'tecnico') NOT NULL,  
 detalles\_rol TEXT,  
 FOREIGN KEY (id\_instrumento) REFERENCES INSTRUMENTO\_PUBLICO(id\_instrumento),  
 FOREIGN KEY (id\_persona) REFERENCES PERSONA\_FISICA(id\_persona)  
);

**Explicación**:

* Registra participación de personas físicas en instrumentos
* rol define la función en el acto (valores predefinidos)
* detalles\_rol para información adicional sobre la participación

#### Tabla: REPRESENTACION

CREATE TABLE REPRESENTACION (  
 id\_representacion INT PRIMARY KEY AUTO\_INCREMENT,  
 tipo ENUM('voluntaria', 'organica', 'legal') NOT NULL,  
 documento\_apoyo VARCHAR(255) NOT NULL,  
 veracidad BOOLEAN NOT NULL,  
 vigencia BOOLEAN NOT NULL,  
 suficiencia BOOLEAN NOT NULL,  
 fecha\_verificacion DATE NOT NULL,  
 id\_notario\_verificador INT NOT NULL,  
 FOREIGN KEY (id\_notario\_verificador) REFERENCES NOTARIO(id\_notario)  
);

**Explicación**:

* Tipos según clasificación legal
* Atributos de control verificados por notario
* Documento que acredita la representación
* Fecha de verificación importante para trazabilidad

#### Tabla: INTERVENCION

CREATE TABLE INTERVENCION (  
 id\_intervencion INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_instrumento INT NOT NULL,  
 id\_representacion INT NOT NULL,  
 id\_representante INT NOT NULL,  
 id\_representado\_fisico INT,  
 id\_representado\_juridico INT,  
 tipo\_sujeto ENUM('fisica', 'juridica') NOT NULL,  
 detalles TEXT,  
 FOREIGN KEY (id\_instrumento) REFERENCES INSTRUMENTO\_PUBLICO(id\_instrumento),  
 FOREIGN KEY (id\_representacion) REFERENCES REPRESENTACION(id\_representacion),  
 FOREIGN KEY (id\_representante) REFERENCES PERSONA\_FISICA(id\_persona),  
 FOREIGN KEY (id\_representado\_fisico) REFERENCES PERSONA\_FISICA(id\_persona),  
 FOREIGN KEY (id\_representado\_juridico) REFERENCES PERSONA\_JURIDICA(id\_juridica),  
 CHECK ((tipo\_sujeto = 'fisica' AND id\_representado\_fisico IS NOT NULL AND id\_representado\_juridico IS NULL) OR  
 (tipo\_sujeto = 'juridica' AND id\_representado\_fisico IS NULL AND id\_representado\_juridico IS NOT NULL))  
);

**Explicación**:

* Registra casos de representación en instrumentos
* tipo\_sujeto indica si el representado es persona física o jurídica
* Restricción CHECK asegura consistencia en el tipo de sujeto
* detalles para información adicional sobre la intervención

#### Tabla: TITULAR\_REAL

CREATE TABLE TITULAR\_REAL (  
 id\_titular\_real INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_juridica INT NOT NULL,  
 id\_persona INT NOT NULL,  
 porcentaje\_participacion DECIMAL(5,2),  
 fecha\_designacion DATE NOT NULL,  
 detalles\_control TEXT,  
 FOREIGN KEY (id\_juridica) REFERENCES PERSONA\_JURIDICA(id\_juridica),  
 FOREIGN KEY (id\_persona) REFERENCES PERSONA\_FISICA(id\_persona),  
 UNIQUE KEY (id\_juridica, id\_persona)  
);

**Explicación**:

* Implementa relación M:N entre personas jurídicas y físicas
* porcentaje\_participacion para indicar nivel de control
* Clave única compuesta para evitar duplicados
* Cumple con requisitos de Ley 10/2010 sobre blanqueo

#### Tabla: JUICIO\_CAPACIDAD

CREATE TABLE JUICIO\_CAPACIDAD (  
 id\_juicio INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_instrumento INT NOT NULL,  
 id\_persona INT NOT NULL,  
 id\_notario INT NOT NULL,  
 capacidad\_natural BOOLEAN NOT NULL,  
 legitimacion BOOLEAN NOT NULL,  
 observaciones TEXT,  
 fecha\_evaluacion DATE NOT NULL,  
 FOREIGN KEY (id\_instrumento) REFERENCES INSTRUMENTO\_PUBLICO(id\_instrumento),  
 FOREIGN KEY (id\_persona) REFERENCES PERSONA\_FISICA(id\_persona),  
 FOREIGN KEY (id\_notario) REFERENCES NOTARIO(id\_notario)  
);

**Explicación**:

* Registra evaluación de capacidad por notario
* Dos dimensiones: capacidad natural y legitimación
* observaciones para detalles de la evaluación
* Fecha importante para efectos legales

### 2.3 Normalización de la Base de Datos

La estructura propuesta cumple con las principales formas normales:

1. **Primera Forma Normal (1NF)**:
   1. Todos los atributos son atómicos
   2. No hay grupos repetitivos
   3. Cada tabla tiene una clave primaria
2. **Segunda Forma Normal (2NF)**:
   1. Cumple con 1NF
   2. Todos los atributos no clave dependen completamente de la clave primaria
3. **Tercera Forma Normal (3NF)**:
   1. Cumple con 2NF
   2. No existen dependencias transitivas (atributos no clave que dependen de otros atributos no clave)

Ejemplo de análisis de normalización para la tabla DOMICILIO:

* **Clave primaria**: id\_domicilio
* **Atributos**: id\_persona, id\_juridica, tipo, pais, provincia, municipio, etc.
* **Dependencias funcionales**:
  + id\_domicilio → todos los demás atributos
  + No hay dependencias parciales (cumple 2NF)
  + No hay dependencias transitivas (cumple 3NF)

### 2.4 Integridad Referencial

El diseño incluye múltiples mecanismos para garantizar integridad:

1. **Claves primarias** en todas las tablas
2. **Claves foráneas** para relaciones
3. **Restricciones CHECK** para validación de datos
4. **UNIQUE constraints** para evitar duplicados
5. **Enumeraciones (ENUM)** para valores predefinidos

Ejemplo de integridad en relación Instrumento-Comparecencia:

* Una COMPARECENCIA no puede existir sin un INSTRUMENTO\_PUBLICO
* Al eliminar un instrumento, se deben eliminar sus comparecencias (ON DELETE CASCADE)
* No se puede insertar una comparecencia con id\_instrumento inexistente

## Fase 3: Implementación Física

### 3.1 Elección del Sistema Gestor de Base de Datos (SGBD)

Para este sistema se recomienda un SGBD relacional robusto como:

1. **Oracle Database**:
   1. Ideal para entornos empresariales críticos
   2. Soporte avanzado para transacciones
   3. Mecanismos de seguridad robustos
2. **PostgreSQL**:
   1. Open-source con capacidades empresariales
   2. Excelente soporte para JSON y datos estructurados
   3. Cumplimiento ACID
3. **Microsoft SQL Server**:
   1. Buena integración con entornos Windows
   2. Herramientas de BI integradas
   3. Soporte para procedimientos almacenados complejos

### 3.2 Script de Creación Completo

*-- Creación de la base de datos*  
CREATE DATABASE notarial\_db CHARACTER SET utf8mb4 COLLATE utf8mb4\_spanish\_ci;  
  
USE notarial\_db;  
  
*-- Tabla NOTARIO*  
CREATE TABLE NOTARIO (  
 id\_notario INT PRIMARY KEY AUTO\_INCREMENT,  
 nombre VARCHAR(100) NOT NULL,  
 apellidos VARCHAR(100) NOT NULL,  
 colegio\_notarial VARCHAR(100) NOT NULL,  
 datos\_habilitacion TEXT,  
 datos\_sustitucion TEXT,  
 fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 fecha\_actualizacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP  
) ENGINE=InnoDB;  
  
*-- Tabla INSTRUMENTO\_PUBLICO*  
CREATE TABLE INSTRUMENTO\_PUBLICO (  
 id\_instrumento INT PRIMARY KEY AUTO\_INCREMENT,  
 numero\_protocolo VARCHAR(20) NOT NULL UNIQUE,  
 lugar\_otorgamiento VARCHAR(255) NOT NULL,  
 fecha\_otorgamiento DATE NOT NULL,  
 hora\_otorgamiento TIME,  
 id\_notario INT NOT NULL,  
 fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 fecha\_actualizacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,  
 FOREIGN KEY (id\_notario) REFERENCES NOTARIO(id\_notario) ON DELETE RESTRICT  
) ENGINE=InnoDB;  
  
*-- Tabla ENCABEZAMIENTO*  
CREATE TABLE ENCABEZAMIENTO (  
 id\_encabezamiento INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_instrumento INT UNIQUE NOT NULL,  
 detalles\_adicionales TEXT,  
 fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (id\_instrumento) REFERENCES INSTRUMENTO\_PUBLICO(id\_instrumento) ON DELETE CASCADE  
) ENGINE=InnoDB;  
  
*-- Tabla PERSONA\_FISICA*  
CREATE TABLE PERSONA\_FISICA (  
 id\_persona INT PRIMARY KEY AUTO\_INCREMENT,  
 dni\_nie VARCHAR(20) NOT NULL UNIQUE,  
 nombre VARCHAR(100) NOT NULL,  
 apellido1 VARCHAR(100) NOT NULL,  
 apellido2 VARCHAR(100),  
 fecha\_nacimiento DATE,  
 estado\_civil ENUM('soltero', 'casado', 'separado', 'viudo', 'divorciado'),  
 profesion VARCHAR(100),  
 nif VARCHAR(20) NOT NULL UNIQUE,  
 nacionalidad VARCHAR(50),  
 vecindad\_civil VARCHAR(100),  
 es\_residente BOOLEAN DEFAULT TRUE,  
 fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 fecha\_actualizacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP  
) ENGINE=InnoDB;  
  
*-- Tabla PERSONA\_JURIDICA*  
CREATE TABLE PERSONA\_JURIDICA (  
 id\_juridica INT PRIMARY KEY AUTO\_INCREMENT,  
 denominacion\_social VARCHAR(255) NOT NULL UNIQUE,  
 fecha\_constitucion DATE NOT NULL,  
 lugar\_constitucion VARCHAR(255) NOT NULL,  
 objeto\_social TEXT NOT NULL,  
 codigo\_cnae VARCHAR(10),  
 nif VARCHAR(20) NOT NULL UNIQUE,  
 numero\_registro VARCHAR(50) NOT NULL,  
 fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 fecha\_actualizacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP  
) ENGINE=InnoDB;  
  
*-- Tabla DOMICILIO*  
CREATE TABLE DOMICILIO (  
 id\_domicilio INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_persona INT,  
 id\_juridica INT,  
 tipo ENUM('fisica', 'juridica') NOT NULL,  
 pais VARCHAR(100) NOT NULL,  
 provincia VARCHAR(100),  
 municipio VARCHAR(100) NOT NULL,  
 tipo\_via ENUM('calle', 'avenida', 'paseo', 'plaza', 'otro') NOT NULL,  
 nombre\_via VARCHAR(200) NOT NULL,  
 numero VARCHAR(20) NOT NULL,  
 bloque VARCHAR(20),  
 escalera VARCHAR(20),  
 planta VARCHAR(20),  
 puerta VARCHAR(20),  
 email VARCHAR(100),  
 fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (id\_persona) REFERENCES PERSONA\_FISICA(id\_persona) ON DELETE CASCADE,  
 FOREIGN KEY (id\_juridica) REFERENCES PERSONA\_JURIDICA(id\_juridica) ON DELETE CASCADE,  
 CHECK ((id\_persona IS NOT NULL AND id\_juridica IS NULL AND tipo = 'fisica') OR   
 (id\_persona IS NULL AND id\_juridica IS NOT NULL AND tipo = 'juridica'))  
) ENGINE=InnoDB;  
  
*-- Tabla REPRESENTACION*  
CREATE TABLE REPRESENTACION (  
 id\_representacion INT PRIMARY KEY AUTO\_INCREMENT,  
 tipo ENUM('voluntaria', 'organica', 'legal') NOT NULL,  
 documento\_apoyo VARCHAR(255) NOT NULL,  
 veracidad BOOLEAN NOT NULL,  
 vigencia BOOLEAN NOT NULL,  
 suficiencia BOOLEAN NOT NULL,  
 fecha\_verificacion DATE NOT NULL,  
 id\_notario\_verificador INT NOT NULL,  
 fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (id\_notario\_verificador) REFERENCES NOTARIO(id\_notario) ON DELETE RESTRICT  
) ENGINE=InnoDB;  
  
*-- Tabla COMPARECENCIA*  
CREATE TABLE COMPARECENCIA (  
 id\_comparecencia INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_instrumento INT NOT NULL,  
 id\_persona INT NOT NULL,  
 rol ENUM('otorgante', 'representante', 'testigo', 'traductor', 'tecnico') NOT NULL,  
 detalles\_rol TEXT,  
 fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (id\_instrumento) REFERENCES INSTRUMENTO\_PUBLICO(id\_instrumento) ON DELETE CASCADE,  
 FOREIGN KEY (id\_persona) REFERENCES PERSONA\_FISICA(id\_persona) ON DELETE RESTRICT  
) ENGINE=InnoDB;  
  
*-- Tabla INTERVENCION*  
CREATE TABLE INTERVENCION (  
 id\_intervencion INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_instrumento INT NOT NULL,  
 id\_representacion INT NOT NULL,  
 id\_representante INT NOT NULL,  
 id\_representado\_fisico INT,  
 id\_representado\_juridico INT,  
 tipo\_sujeto ENUM('fisica', 'juridica') NOT NULL,  
 detalles TEXT,  
 fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (id\_instrumento) REFERENCES INSTRUMENTO\_PUBLICO(id\_instrumento) ON DELETE CASCADE,  
 FOREIGN KEY (id\_representacion) REFERENCES REPRESENTACION(id\_representacion) ON DELETE RESTRICT,  
 FOREIGN KEY (id\_representante) REFERENCES PERSONA\_FISICA(id\_persona) ON DELETE RESTRICT,  
 FOREIGN KEY (id\_representado\_fisico) REFERENCES PERSONA\_FISICA(id\_persona) ON DELETE RESTRICT,  
 FOREIGN KEY (id\_representado\_juridico) REFERENCES PERSONA\_JURIDICA(id\_juridica) ON DELETE RESTRICT,  
 CHECK ((tipo\_sujeto = 'fisica' AND id\_representado\_fisico IS NOT NULL AND id\_representado\_juridico IS NULL) OR  
 (tipo\_sujeto = 'juridica' AND id\_representado\_fisico IS NULL AND id\_representado\_juridico IS NOT NULL))  
) ENGINE=InnoDB;  
  
*-- Tabla TITULAR\_REAL*  
CREATE TABLE TITULAR\_REAL (  
 id\_titular\_real INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_juridica INT NOT NULL,  
 id\_persona INT NOT NULL,  
 porcentaje\_participacion DECIMAL(5,2),  
 fecha\_designacion DATE NOT NULL,  
 detalles\_control TEXT,  
 fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (id\_juridica) REFERENCES PERSONA\_JURIDICA(id\_juridica) ON DELETE CASCADE,  
 FOREIGN KEY (id\_persona) REFERENCES PERSONA\_FISICA(id\_persona) ON DELETE RESTRICT,  
 UNIQUE KEY (id\_juridica, id\_persona)  
) ENGINE=InnoDB;  
  
*-- Tabla JUICIO\_CAPACIDAD*  
CREATE TABLE JUICIO\_CAPACIDAD (  
 id\_juicio INT PRIMARY KEY AUTO\_INCREMENT,  
 id\_instrumento INT NOT NULL,  
 id\_persona INT NOT NULL,  
 id\_notario INT NOT NULL,  
 capacidad\_natural BOOLEAN NOT NULL,  
 legitimacion BOOLEAN NOT NULL,  
 observaciones TEXT,  
 fecha\_evaluacion DATE NOT NULL,  
 fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (id\_instrumento) REFERENCES INSTRUMENTO\_PUBLICO(id\_instrumento) ON DELETE CASCADE,  
 FOREIGN KEY (id\_persona) REFERENCES PERSONA\_FISICA(id\_persona) ON DELETE RESTRICT,  
 FOREIGN KEY (id\_notario) REFERENCES NOTARIO(id\_notario) ON DELETE RESTRICT  
) ENGINE=InnoDB;

### 3.3 Índices y Optimización

Para mejorar el rendimiento, se deben crear índices estratégicos:

*-- Índices para búsquedas frecuentes*  
CREATE INDEX idx\_persona\_nombre ON PERSONA\_FISICA(nombre, apellido1, apellido2);  
CREATE INDEX idx\_juridica\_denominacion ON PERSONA\_JURIDICA(denominacion\_social);  
CREATE INDEX idx\_instrumento\_fecha ON INSTRUMENTO\_PUBLICO(fecha\_otorgamiento);  
CREATE INDEX idx\_instrumento\_notario ON INSTRUMENTO\_PUBLICO(id\_notario, fecha\_otorgamiento);  
  
*-- Índices para claves foráneas*  
CREATE INDEX idx\_comparecencia\_persona ON COMPARECENCIA(id\_persona);  
CREATE INDEX idx\_intervencion\_representante ON INTERVENCION(id\_representante);  
CREATE INDEX idx\_intervencion\_representado\_fisico ON INTERVENCION(id\_representado\_fisico);  
CREATE INDEX idx\_intervencion\_representado\_juridico ON INTERVENCION(id\_representado\_juridico);  
CREATE INDEX idx\_titular\_real\_persona ON TITULAR\_REAL(id\_persona);  
  
*-- Índices para campos únicos*  
CREATE UNIQUE INDEX idx\_persona\_dni ON PERSONA\_FISICA(dni\_nie);  
CREATE UNIQUE INDEX idx\_persona\_nif ON PERSONA\_FISICA(nif);  
CREATE UNIQUE INDEX idx\_juridica\_nif ON PERSONA\_JURIDICA(nif);

### 3.4 Vistas para Consultas Comunes

*-- Vista para instrumentos con información completa*  
CREATE VIEW vista\_instrumentos\_completos AS  
SELECT   
 ip.id\_instrumento,  
 ip.numero\_protocolo,  
 ip.lugar\_otorgamiento,  
 ip.fecha\_otorgamiento,  
 n.nombre AS nombre\_notario,  
 n.apellidos AS apellidos\_notario,  
 n.colegio\_notarial,  
 COUNT(DISTINCT c.id\_comparecencia) AS total\_comparecientes,  
 COUNT(DISTINCT i.id\_intervencion) AS total\_intervenciones  
FROM INSTRUMENTO\_PUBLICO ip  
JOIN NOTARIO n ON ip.id\_notario = n.id\_notario  
LEFT JOIN COMPARECENCIA c ON ip.id\_instrumento = c.id\_instrumento  
LEFT JOIN INTERVENCION i ON ip.id\_instrumento = i.id\_instrumento  
GROUP BY ip.id\_instrumento;  
  
*-- Vista para personas con sus roles en instrumentos*  
CREATE VIEW vista\_personas\_roles AS  
SELECT   
 pf.id\_persona,  
 pf.dni\_nie,  
 pf.nombre,  
 pf.apellido1,  
 pf.apellido2,  
 c.rol,  
 COUNT(DISTINCT c.id\_instrumento) AS total\_instrumentos,  
 GROUP\_CONCAT(DISTINCT ip.numero\_protocolo ORDER BY ip.fecha\_otorgamiento DESC SEPARATOR ', ') AS protocolos  
FROM PERSONA\_FISICA pf  
LEFT JOIN COMPARECENCIA c ON pf.id\_persona = c.id\_persona  
LEFT JOIN INSTRUMENTO\_PUBLICO ip ON c.id\_instrumento = ip.id\_instrumento  
GROUP BY pf.id\_persona, c.rol;  
  
*-- Vista para representaciones verificadas*  
CREATE VIEW vista\_representaciones\_verificadas AS  
SELECT   
 r.id\_representacion,  
 r.tipo,  
 r.documento\_apoyo,  
 n.nombre AS nombre\_notario,  
 n.apellidos AS apellidos\_notario,  
 pf\_rep.nombre AS nombre\_representante,  
 pf\_rep.apellido1 AS apellido1\_representante,  
 CASE   
 WHEN i.tipo\_sujeto = 'fisica' THEN CONCAT(pf\_rep.nombre, ' ', pf\_rep.apellido1)  
 WHEN i.tipo\_sujeto = 'juridica' THEN pj.denominacion\_social  
 END AS sujeto\_representado,  
 i.id\_instrumento,  
 ip.numero\_protocolo  
FROM REPRESENTACION r  
JOIN NOTARIO n ON r.id\_notario\_verificador = n.id\_notario  
JOIN INTERVENCION i ON r.id\_representacion = i.id\_representacion  
JOIN PERSONA\_FISICA pf\_rep ON i.id\_representante = pf\_rep.id\_persona  
LEFT JOIN PERSONA\_FISICA pf\_repd ON i.id\_representado\_fisico = pf\_repd.id\_persona AND i.tipo\_sujeto = 'fisica'  
LEFT JOIN PERSONA\_JURIDICA pj ON i.id\_representado\_juridico = pj.id\_juridica AND i.tipo\_sujeto = 'juridica'  
JOIN INSTRUMENTO\_PUBLICO ip ON i.id\_instrumento = ip.id\_instrumento;

### 3.5 Procedimientos Almacenados

*-- Procedimiento para registrar nuevo instrumento*  
DELIMITER *//*  
CREATE PROCEDURE sp\_registrar\_instrumento(  
 IN p\_numero\_protocolo VARCHAR(20),  
 IN p\_lugar\_otorgamiento VARCHAR(255),  
 IN p\_fecha\_otorgamiento DATE,

IN p\_hora\_otorgamiento TIME,  
 IN p\_id\_notario INT,  
 OUT p\_id\_instrumento INT  
)  
BEGIN  
 INSERT INTO INSTRUMENTO\_PUBLICO (  
 numero\_protocolo,  
 lugar\_otorgamiento,  
 fecha\_otorgamiento,  
 hora\_otorgamiento,  
 id\_notario  
 ) VALUES (  
 p\_numero\_protocolo,  
 p\_lugar\_otorgamiento,  
 p\_fecha\_otorgamiento,  
 p\_hora\_otorgamiento,  
 p\_id\_notario  
 );  
   
 SET p\_id\_instrumento = LAST\_INSERT\_ID();  
   
 *-- Crear encabezamiento automáticamente*  
 INSERT INTO ENCABEZAMIENTO (id\_instrumento) VALUES (p\_id\_instrumento);  
END *//*  
DELIMITER ;  
  
*-- Procedimiento para registrar comparecencia*  
DELIMITER *//*  
CREATE PROCEDURE sp\_registrar\_comparecencia(  
 IN p\_id\_instrumento INT,  
 IN p\_dni\_nie VARCHAR(20),  
 IN p\_rol ENUM('otorgante', 'representante', 'testigo', 'traductor', 'tecnico'),  
 IN p\_detalles\_rol TEXT,  
 OUT p\_id\_comparecencia INT  
)  
BEGIN  
 DECLARE v\_id\_persona INT;  
   
 *-- Obtener ID de persona*  
 SELECT id\_persona INTO v\_id\_persona FROM PERSONA\_FISICA WHERE dni\_nie = p\_dni\_nie;  
   
 IF v\_id\_persona IS NULL THEN  
 SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Persona no encontrada';  
 END IF;  
   
 *-- Insertar comparecencia*  
 INSERT INTO COMPARECENCIA (  
 id\_instrumento,  
 id\_persona,  
 rol,  
 detalles\_rol  
 ) VALUES (  
 p\_id\_instrumento,  
 v\_id\_persona,  
 p\_rol,  
 p\_detalles\_rol  
 );  
   
 SET p\_id\_comparecencia = LAST\_INSERT\_ID();  
END *//*  
DELIMITER ;  
  
*-- Procedimiento para evaluar capacidad*  
DELIMITER *//*  
CREATE PROCEDURE sp\_evaluar\_capacidad(  
 IN p\_id\_instrumento INT,  
 IN p\_dni\_nie VARCHAR(20),  
 IN p\_id\_notario INT,  
 IN p\_capacidad\_natural BOOLEAN,  
 IN p\_legitimacion BOOLEAN,  
 IN p\_observaciones TEXT,  
 OUT p\_id\_juicio INT  
)  
BEGIN  
 DECLARE v\_id\_persona INT;  
   
 *-- Obtener ID de persona*  
 SELECT id\_persona INTO v\_id\_persona FROM PERSONA\_FISICA WHERE dni\_nie = p\_dni\_nie;  
   
 IF v\_id\_persona IS NULL THEN  
 SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Persona no encontrada';  
 END IF;  
   
 *-- Insertar evaluación*  
 INSERT INTO JUICIO\_CAPACIDAD (  
 id\_instrumento,  
 id\_persona,  
 id\_notario,  
 capacidad\_natural,  
 legitimacion,  
 observaciones,  
 fecha\_evaluacion  
 ) VALUES (  
 p\_id\_instrumento,  
 v\_id\_persona,  
 p\_id\_notario,  
 p\_capacidad\_natural,  
 p\_legitimacion,  
 p\_observaciones,  
 CURDATE()  
 );  
   
 SET p\_id\_juicio = LAST\_INSERT\_ID();  
END *//*  
DELIMITER ;

Fase 4: Seguridad y Control de Accesos (Continuación)

4.2 Políticas de Seguridad Detalladas

4.2.1 Autenticación y Encriptación

1. **Autenticación Fuerte**:
   * Implementar autenticación multifactor para usuarios con privilegios elevados
   * Requerir certificados digitales para notarios que accedan remotamente
   * Configurar políticas de complejidad de contraseñas:

sql

Copy

Download

SET GLOBAL validate\_password.policy = STRONG;

SET GLOBAL validate\_password.length = 12;

SET GLOBAL validate\_password.mixed\_case\_count = 1;

SET GLOBAL validate\_password.number\_count = 1;

SET GLOBAL validate\_password.special\_char\_count = 1;

1. **Encriptación de Datos**:
   * Encriptación de datos en tránsito (SSL/TLS):

sql

Copy

Download

SHOW VARIABLES LIKE '%ssl%';

* + Encriptación de datos en reposo:

sql

Copy

Download

CREATE TABLE datos\_sensibles (

id INT PRIMARY KEY,

datos VARBINARY(255) NOT NULL

) ENCRYPTION='Y';

1. **Máscara de Datos**:
   * Implementar funciones de enmascaramiento para consultas de usuarios con roles limitados:

sql

Copy

Download

CREATE FUNCTION mask\_dni(dni VARCHAR(20)) RETURNS VARCHAR(20) DETERMINISTIC

RETURN CONCAT('\*\*\*', RIGHT(dni, 4));

4.2.2 Control de Acceso Basado en Roles (RBAC)

Estructura detallada de privilegios por rol:

| **Rol** | **Tablas/Objetos** | **Permisos Específicos** | **Restricciones** |
| --- | --- | --- | --- |
| rol\_notario | Todas | SELECT, INSERT, UPDATE | No DELETE |
|  | INSTRUMENTO\_PUBLICO | ALTER (solo campos específicos) | Solo propios registros |
|  | JUICIO\_CAPACIDAD | INSERT, UPDATE |  |
| rol\_administrativo | COMPARECENCIA, INTERVENCION | SELECT, INSERT |  |
|  | PERSONA\_FISICA, PERSONA\_JURIDICA | SELECT |  |
|  | Vistas | SELECT |  |
| rol\_consulta | Vistas | SELECT | Solo datos no sensibles |
| rol\_auditor | Todas | SELECT | Solo acceso desde IPs autorizadas |

Implementación en SQL:

sql

Copy

Download

*-- Permisos específicos para notarios sobre sus propios instrumentos*

CREATE VIEW vista\_notario\_instrumentos AS

SELECT \* FROM INSTRUMENTO\_PUBLICO

WHERE id\_notario = SUBSTRING\_INDEX(USER(), '@', 1);

GRANT SELECT ON vista\_notario\_instrumentos TO rol\_notario;

*-- Restricción para evitar eliminaciones*

REVOKE DELETE ON notarial\_db.\* FROM rol\_notario;

4.3 Seguridad a Nivel de Fila (Row-Level Security)

Implementación para que los notarios solo vean sus propios instrumentos:

sql

Copy

Download

*-- Creación de políticas en PostgreSQL (ejemplo)*

CREATE POLICY notario\_policy ON INSTRUMENTO\_PUBLICO

FOR ALL

TO rol\_notario

USING (id\_notario = current\_setting('app.current\_notario\_id')::integer);

*-- En MySQL se puede simular con vistas:*

CREATE VIEW vista\_instrumentos\_notario AS

SELECT \* FROM INSTRUMENTO\_PUBLICO

WHERE id\_notario = (SELECT id\_notario FROM notario\_session WHERE usuario = CURRENT\_USER());

4.4 Auditoría y Trazabilidad

4.4.1 Tablas de Auditoría

sql

Copy

Download

CREATE TABLE auditoria (

id\_auditoria BIGINT PRIMARY KEY AUTO\_INCREMENT,

tabla\_afectada VARCHAR(50) NOT NULL,

id\_registro INT NOT NULL,

tipo\_operacion ENUM('INSERT', 'UPDATE', 'DELETE') NOT NULL,

usuario VARCHAR(100) NOT NULL,

fecha\_hora TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

valores\_anteriores JSON,

valores\_nuevos JSON,

ip\_conexion VARCHAR(45)

) ENGINE=InnoDB;

*-- Índices para búsquedas de auditoría*

CREATE INDEX idx\_auditoria\_tabla ON auditoria(tabla\_afectada);

CREATE INDEX idx\_auditoria\_fecha ON auditoria(fecha\_hora);

CREATE INDEX idx\_auditoria\_usuario ON auditoria(usuario);

4.4.2 Triggers para Auditoría Automática

Ejemplo para la tabla INSTRUMENTO\_PUBLICO:

sql

Copy

Download

DELIMITER *//*

CREATE TRIGGER trig\_audit\_instrumento\_insert

AFTER INSERT ON INSTRUMENTO\_PUBLICO

FOR EACH ROW

BEGIN

INSERT INTO auditoria (

tabla\_afectada,

id\_registro,

tipo\_operacion,

usuario,

valores\_nuevos,

ip\_conexion

) VALUES (

'INSTRUMENTO\_PUBLICO',

NEW.id\_instrumento,

'INSERT',

CURRENT\_USER(),

JSON\_OBJECT(

'numero\_protocolo', NEW.numero\_protocolo,

'lugar\_otorgamiento', NEW.lugar\_otorgamiento,

'fecha\_otorgamiento', NEW.fecha\_otorgamiento,

'id\_notario', NEW.id\_notario

),

CONNECTION\_ID()

);

END *//*

DELIMITER ;

DELIMITER *//*

CREATE TRIGGER trig\_audit\_instrumento\_update

AFTER UPDATE ON INSTRUMENTO\_PUBLICO

FOR EACH ROW

BEGIN

IF NEW.numero\_protocolo != OLD.numero\_protocolo OR

NEW.lugar\_otorgamiento != OLD.lugar\_otorgamiento OR

NEW.fecha\_otorgamiento != OLD.fecha\_otorgamiento OR

NEW.id\_notario != OLD.id\_notario THEN

INSERT INTO auditoria (

tabla\_afectada,

id\_registro,

tipo\_operacion,

usuario,

valores\_anteriores,

valores\_nuevos,

ip\_conexion

) VALUES (

'INSTRUMENTO\_PUBLICO',

NEW.id\_instrumento,

'UPDATE',

CURRENT\_USER(),

JSON\_OBJECT(

'numero\_protocolo', OLD.numero\_protocolo,

'lugar\_otorgamiento', OLD.lugar\_otorgamiento,

'fecha\_otorgamiento', OLD.fecha\_otorgamiento,

'id\_notario', OLD.id\_notario

),

JSON\_OBJECT(

'numero\_protocolo', NEW.numero\_protocolo,

'lugar\_otorgamiento', NEW.lugar\_otorgamiento,

'fecha\_otorgamiento', NEW.fecha\_otorgamiento,

'id\_notario', NEW.id\_notario

),

CONNECTION\_ID()

);

END IF;

END *//*

DELIMITER ;

DELIMITER *//*

CREATE TRIGGER trig\_audit\_instrumento\_delete

BEFORE DELETE ON INSTRUMENTO\_PUBLICO

FOR EACH ROW

BEGIN

INSERT INTO auditoria (

tabla\_afectada,

id\_registro,

tipo\_operacion,

usuario,

valores\_anteriores,

ip\_conexion

) VALUES (

'INSTRUMENTO\_PUBLICO',

OLD.id\_instrumento,

'DELETE',

CURRENT\_USER(),

JSON\_OBJECT(

'numero\_protocolo', OLD.numero\_protocolo,

'lugar\_otorgamiento', OLD.lugar\_otorgamiento,

'fecha\_otorgamiento', OLD.fecha\_otorgamiento,

'id\_notario', OLD.id\_notario

),

CONNECTION\_ID()

);

END *//*

DELIMITER ;

4.5 Cumplimiento Legal (Ley 10/2010)

Implementación de controles específicos para prevención de blanqueo:

sql

Copy

Download

*-- Tabla para registrar información de dueños beneficiarios*

CREATE TABLE titular\_real\_verificacion (

id\_verificacion INT PRIMARY KEY AUTO\_INCREMENT,

id\_titular\_real INT NOT NULL,

documento\_identidad VARCHAR(50) NOT NULL,

origen\_fondos ENUM('ahorros', 'ingresos', 'prestamo', 'herencia', 'otros') NOT NULL,

documento\_fondos VARCHAR(255),

id\_notario\_verificador INT NOT NULL,

fecha\_verificacion DATE NOT NULL,

observaciones TEXT,

FOREIGN KEY (id\_titular\_real) REFERENCES TITULAR\_REAL(id\_titular\_real),

FOREIGN KEY (id\_notario\_verificador) REFERENCES NOTARIO(id\_notario)

);

*-- Procedimiento para verificación de titular real*

DELIMITER *//*

CREATE PROCEDURE sp\_verificar\_titular\_real(

IN p\_id\_titular\_real INT,

IN p\_documento\_identidad VARCHAR(50),

IN p\_origen\_fondos VARCHAR(50),

IN p\_documento\_fondos VARCHAR(255),

IN p\_id\_notario INT,

IN p\_observaciones TEXT,

OUT p\_id\_verificacion INT

)

BEGIN

DECLARE v\_id\_juridica INT;

DECLARE v\_id\_persona INT;

*-- Obtener datos del titular real*

SELECT id\_juridica, id\_persona INTO v\_id\_juridica, v\_id\_persona

FROM TITULAR\_REAL WHERE id\_titular\_real = p\_id\_titular\_real;

IF v\_id\_juridica IS NULL OR v\_id\_persona IS NULL THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Titular real no encontrado';

END IF;

*-- Insertar verificación*

INSERT INTO titular\_real\_verificacion (

id\_titular\_real,

documento\_identidad,

origen\_fondos,

documento\_fondos,

id\_notario\_verificador,

fecha\_verificacion,

observaciones

) VALUES (

p\_id\_titular\_real,

p\_documento\_identidad,

p\_origen\_fondos,

p\_documento\_fondos,

p\_id\_notario,

CURDATE(),

p\_observaciones

);

SET p\_id\_verificacion = LAST\_INSERT\_ID();

*-- Registrar en auditoría*

INSERT INTO auditoria (

tabla\_afectada,

id\_registro,

tipo\_operacion,

usuario,

valores\_nuevos,

ip\_conexion

) VALUES (

'titular\_real\_verificacion',

p\_id\_verificacion,

'INSERT',

CURRENT\_USER(),

JSON\_OBJECT(

'id\_titular\_real', p\_id\_titular\_real,

'id\_notario', p\_id\_notario,

'origen\_fondos', p\_origen\_fondos

),

CONNECTION\_ID()

);

END *//*

DELIMITER ;

4.6 Backup y Recuperación

4.6.1 Estrategia de Backup

1. **Backups completos diarios**:

bash

Copy

Download

mysqldump -u root -p --opt --routines --triggers --events notarial\_db > /backups/notarial\_completo\_$(date +%Y%m%d).sql

1. **Backups incrementales horarios**:

sql

Copy

Download

*-- Habilitar binary logging*

SET GLOBAL log\_bin = ON;

SET GLOBAL binlog\_format = 'ROW';

1. **Backup de registros críticos**:

sql

Copy

Download

*-- Exportar datos sensibles con encriptación*

SELECT id\_persona, nombre, apellido1, apellido2, dni\_nie

INTO OUTFILE '/secure\_backup/personas.csv'

FIELDS TERMINATED BY ',' ENCLOSED BY '"'

LINES TERMINATED BY '\n'

FROM PERSONA\_FISICA;

4.6.2 Política de Retención

| **Tipo Backup** | **Frecuencia** | **Retención** | **Localización** | **Encriptación** |
| --- | --- | --- | --- | --- |
| Completo | Diario | 30 días | On-site + Cloud | AES-256 |
| Incremental | Cada 4 horas | 7 días | On-site | AES-128 |
| Logs transaccionales | Continuo | 48 horas | Servidor dedicado | TLS 1.2+ |

4.7 Monitorización y Alertas

Configuración de monitoreo para seguridad:

sql

Copy

Download

*-- Creación de tabla para eventos de seguridad*

CREATE TABLE eventos\_seguridad (

id\_evento BIGINT PRIMARY KEY AUTO\_INCREMENT,

tipo\_evento ENUM('intento\_acceso', 'modificacion\_masiva', 'acceso\_privilegiado', 'violacion\_politica') NOT NULL,

descripcion TEXT NOT NULL,

usuario VARCHAR(100),

ip\_origen VARCHAR(45),

fecha\_hora TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

severidad ENUM('baja', 'media', 'alta', 'critica') NOT NULL,

accion\_tomada VARCHAR(255)

);

*-- Trigger para detectar modificaciones masivas*

DELIMITER *//*

CREATE TRIGGER trig\_deteccion\_modificacion\_masiva

AFTER UPDATE ON PERSONA\_FISICA

FOR EACH ROW

BEGIN

DECLARE v\_count INT;

SELECT COUNT(\*) INTO v\_count FROM INFORMATION\_SCHEMA.PROCESSLIST

WHERE INFO LIKE '%UPDATE PERSONA\_FISICA%' AND USER = CURRENT\_USER();

IF v\_count > 5 THEN

INSERT INTO eventos\_seguridad (

tipo\_evento,

descripcion,

usuario,

ip\_origen,

severidad

) VALUES (

'modificacion\_masiva',

CONCAT('Actualización masiva de registros en PERSONA\_FISICA por ', CURRENT\_USER()),

CURRENT\_USER(),

CONNECTION\_ID(),

'alta'

);

END IF;

END *//*

DELIMITER ;

Fase 5: Integración y Operaciones

5.1 Integración con Sistemas Externos

5.1.1 Conexión con Índice Único Informatizado

sql

Copy

Download

*-- Tabla para registro de intercambios*

CREATE TABLE integracion\_indice\_unico (

id\_integracion INT PRIMARY KEY AUTO\_INCREMENT,

id\_instrumento INT NOT NULL,

fecha\_envio DATETIME NOT NULL,

estado ENUM('pendiente', 'enviado', 'confirmado', 'error') NOT NULL,

codigo\_respuesta VARCHAR(10),

mensaje\_respuesta TEXT,

intentos TINYINT DEFAULT 0,

FOREIGN KEY (id\_instrumento) REFERENCES INSTRUMENTO\_PUBLICO(id\_instrumento)

);

*-- Procedimiento para enviar datos*

DELIMITER *//*

CREATE PROCEDURE sp\_enviar\_indice\_unico(IN p\_id\_instrumento INT)

BEGIN

DECLARE v\_count INT;

*-- Verificar si ya fue enviado*

SELECT COUNT(\*) INTO v\_count FROM integracion\_indice\_unico

WHERE id\_instrumento = p\_id\_instrumento AND estado = 'confirmado';

IF v\_count > 0 THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Instrumento ya fue enviado y confirmado';

END IF;

*-- Marcar como enviado (la integración real dependerá de API externa)*

INSERT INTO integracion\_indice\_unico (

id\_instrumento,

fecha\_envio,

estado

) VALUES (

p\_id\_instrumento,

NOW(),

'enviado'

);

*-- Aquí iría la lógica real de integración con el sistema externo*

END *//*

DELIMITER ;

5.2 Operaciones Automatizadas

5.2.1 Eventos Programados

sql

Copy

Download

*-- Evento para limpieza de sesiones expiradas*

DELIMITER *//*

CREATE EVENT evt\_limpieza\_sesiones

ON SCHEDULE EVERY 1 DAY

STARTS CURRENT\_TIMESTAMP

DO

BEGIN

DELETE FROM sesiones\_activas

WHERE fecha\_expiracion < NOW();

INSERT INTO auditoria (

tabla\_afectada,

tipo\_operacion,

usuario,

descripcion

) VALUES (

'sistema',

'mantenimiento',

'system',

'Limpieza automática de sesiones expiradas'

);

END *//*

DELIMITER ;

*-- Evento para backups diarios*

DELIMITER *//*

CREATE EVENT evt\_backup\_diario

ON SCHEDULE EVERY 1 DAY

STARTS '2023-01-01 02:00:00'

DO

BEGIN

*-- Lógica simplificada, en la práctica se ejecutaría un script externo*

INSERT INTO registro\_backups (

tipo\_backup,

fecha\_hora,

tamaño\_estimado,

estado

) VALUES (

'completo',

NOW(),

0,

'iniciado'

);

END *//*

DELIMITER ;

5.3 Mantenimiento y Optimización

5.3.1 Rutinas de Mantenimiento

sql

Copy

Download

*-- Procedimiento para reindexación periódica*

DELIMITER *//*

CREATE PROCEDURE sp\_mantenimiento\_reindexacion()

BEGIN

DECLARE done INT DEFAULT FALSE;

DECLARE tabla\_name VARCHAR(100);

DECLARE cur\_tablas CURSOR FOR

SELECT TABLE\_NAME FROM INFORMATION\_SCHEMA.TABLES

WHERE TABLE\_SCHEMA = 'notarial\_db';

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

OPEN cur\_tablas;

read\_loop: LOOP

FETCH cur\_tablas INTO tabla\_name;

IF done THEN

LEAVE read\_loop;

END IF;

SET @sql = CONCAT('ANALYZE TABLE ', tabla\_name);

PREPARE stmt FROM @sql;

EXECUTE stmt;

DEALLOCATE PREPARE stmt;

INSERT INTO auditoria (

tabla\_afectada,

tipo\_operacion,

usuario,

descripcion

) VALUES (

'sistema',

'mantenimiento',

'system',

CONCAT('Reindexación de tabla ', tabla\_name)

);

END LOOP;

CLOSE cur\_tablas;

END *//*

DELIMITER ;

*-- Evento para mantenimiento semanal*

DELIMITER *//*

CREATE EVENT evt\_mantenimiento\_semanal

ON SCHEDULE EVERY 1 WEEK

STARTS '2023-01-01 03:00:00'

DO

BEGIN

CALL sp\_mantenimiento\_reindexacion();

OPTIMIZE TABLE PERSONA\_FISICA, PERSONA\_JURIDICA, INSTRUMENTO\_PUBLICO;

END *//*

DELIMITER ;

Conclusión

Este diseño detallado de base de datos para instrumentos públicos notariales cumple con todos los requisitos legales y técnicos expuestos en los documentos fuente. La implementación propuesta:

1. **Estructura completa** para gestionar encabezamientos, comparecencias e intervenciones
2. **Sólidos controles de seguridad** que cumplen con la normativa de protección de datos
3. **Mecanismos de auditoría** exhaustivos para garantizar trazabilidad
4. **Procesos automatizados** para mantenimiento e integración
5. **Estrategias de backup** robustas para garantizar continuidad

El sistema está preparado para manejar los volúmenes de datos esperados en un entorno notarial, con especial atención a los requisitos legales como la Ley 10/2010 de prevención de blanqueo de capitales. La arquitectura propuesta permite futuras expansiones y adaptaciones según evolucionen las necesidades del Consejo General del Notariado.