

1. Diseño Conceptual y Modelo Entidad-Relación (MER)

El diseño conceptual está enfocado en digitalizar los seis requisitos funcionales principales del Hospital "San Juan". El modelo se compone de 10 entidades, superando el mínimo de 8 tablas requeridas.

1.1 Entidades y Relaciones Clave

Las entidades principales son:

Pacientes: Almacena datos personales y registra su ubicación en el hospital, manteniendo una relación uno a muchos (1:N) con las Citas que programan y una relación muchos a uno (N:1) con las Habitaciones a la que están asignados.

Doctores: Gestiona al personal médico y tiene una relación uno a muchos (1:N) con Citas.

Especialidades: Es un catálogo de áreas médicas que se relaciona con Doctores en una relación de muchos a muchos (M:N), resuelta por la tabla intermedia Doctor_Especialidad.

Citas: Es el centro del flujo de atención, uniéndose en relaciones muchos a uno (N:1) con Pacientes y Doctores.

Diagnosticos: Registra la conclusión médica. Tiene una relación uno a uno (1:1 lógica) con Citas.

Tratamientos: Define el plan de acción, derivado de un Diagnostico (relación N:1).

Medicamentos: Controla el inventario. Se relaciona con Tratamientos en una relación de muchos a muchos (M:N), resuelta por la tabla intermedia Tratamiento_Medicamento, que especifica la dosis y frecuencia.

Habitaciones: Permite el control de camas disponibles y su estado.

2. Esquema Relacional Normalizado (Tercera Forma Normal - 3FN)

El diseño del esquema relacional asegura que todas las tablas cumplan con la Tercera Forma Normal (3FN). Esto elimina dependencias parciales y transitivas, garantizando la integridad de los datos y minimizando la redundancia.

Para alcanzar la 3FN, se realizó lo siguiente:

Tablas de Unión (N:M): Se crearon Doctor_Especialidad y Tratamiento_Medicamento para eliminar dependencias parciales y que los atributos (como la dosis) dependan únicamente de la clave compuesta completa (ej. id_tratamiento y id_medicamento).

Eliminación de Transitividad: Se aseguró que los atributos no clave dependan directamente de la clave primaria. Por ejemplo, los detalles de Medicamentos (como el stock) residen solo en esa tabla, y solo el id_medicamento se usa como clave foránea en la tabla de unión, evitando duplicar datos de stock cada vez que se usa una medicina en un tratamiento.

3. Implementación y Requisitos Técnicos

3.1 Integridad Referencial (ON DELETE/ON UPDATE)

Se implementó integridad referencial con el uso estratégico de acciones ON DELETE y ON UPDATE:

ON DELETE RESTRICT: Se utiliza en la mayoría de las FK (ej. id_doctor en Citas). Esto impide la eliminación de un doctor si todavía tiene citas registradas, protegiendo los datos históricos esenciales. También se usa para proteger catálogos (ej. Especialidades).

ON DELETE CASCADE: Se aplica a las tablas de unión (Doctor_Especialidad, Tratamiento_Medicamento) y a Tratamientos. Si se borra un tratamiento, se eliminan automáticamente sus líneas de medicación asociadas.

ON DELETE SET NULL: Se implementó en el campo id_habitacion de la tabla Pacientes. Si una habitación es eliminada del sistema, el paciente que estaba en ella simplemente queda sin habitación asignada, sin perder el registro del paciente.

3.2 Restricciones CHECK (Mínimo 3 por Tabla)

Cada tabla incluye restricciones CHECK para validar la lógica del negocio. Ejemplos clave:

Citas: Valida que el estado sea uno de la lista ('Pendiente', 'Realizada', 'Cancelada') y que las citas se programen dentro del horario de atención (8:00 a 18:00 horas).

Medicamentos: Asegura que el stock_actual sea siempre positivo (≥ 0) y que el stock_minimo sea mayor a cero (> 0).

Pacientes: Restringe la fecha_nacimiento a ser anterior a la fecha actual y usa expresiones regulares (CHECK (telefono ~ '^[0-9]+\$')) para garantizar que el teléfono contenga solo dígitos.

Tratamiento_Medicamento: Utiliza expresiones regulares para validar el formato de la dosis (ej. 500mg, 10ml), forzando la inclusión de una unidad de medida válida.

4. Consultas Requeridas

Se diseñaron cinco consultas complejas que responden a requisitos de negocio específicos, utilizando JOIN, funciones de agregación (COUNT), y condiciones de tiempo (NOW() e INTERVAL).

Consulta 1 (Pacientes con > 3 citas): Combina Pacientes y Citas, filtrando por la fecha de las citas (último mes) y utilizando GROUP BY y HAVING para contar y seleccionar solo aquellos con más de tres ocurrencias.

Consulta 2 (Doctores y Citas por Especialidad): Une cuatro tablas (Doctores, Citas, Doctor_Especialidad, Especialidades) para agregar la carga de trabajo por área médica.

Consulta 3 (Stock Bajo): Consulta directa a Medicamentos con la condición WHERE stock_actual < stock_minimo.

Consulta 4 (Habitaciones Disponibles): Consulta a Habitaciones, filtrando por estado = 'Disponible' y agrupando por tipo para obtener el conteo de disponibilidad por categoría.

Consulta 5 (Historial Médico Completo): Utiliza múltiples LEFT JOIN (para incluir citas sin diagnóstico) y une todas las tablas de atención (Citas, Diagnosticos, Tratamientos, Tratamiento_Medicamento, Medicamentos) para reconstruir el historial completo de un paciente específico.

5. Manual de Instalación y Uso

Requisitos Previos

Tener instalado PostgreSQL 14+.

Tener acceso a una herramienta de cliente SQL (psql o pgAdmin).

Pasos de Ejecución

Los scripts deben ejecutarse en el siguiente orden para garantizar que las dependencias de claves foráneas se cumplan:

Creación de la Base de Datos: Conectarse al servidor PostgreSQL y crear la base de datos de destino.

SQL

```
CREATE DATABASE hospital_sanjuan;
```

```
\c hospital_sanjuan;
```

Estructura de la BD: Ejecutar creacion_bd.sql. Este script define todas las tablas, claves primarias, claves foráneas, la integridad referencial y las 3 restricciones CHECK por tabla.

SQL

```
\i [ruta_del_archivo]/01_implementacion/creacion_bd.sql
```

Población de Datos: Ejecutar datos_prueba.sql. Este script inserta datos coherentes, incluyendo el mínimo de 3 citas pasadas para el paciente de prueba, que son esenciales para la Consulta 1.

SQL

```
\i [ruta_del_archivo]/01_implementacion/datos_prueba.sql
```

Ejecución de Consultas: Ejecutar consultas.sql. Este script mostrará los resultados de las cinco consultas obligatorias en la salida estándar.

SQL

```
\i [ruta_del_archivo]/01_implementacion/consultas.sql
```