**EXAMEN INTEGRADOR 2025**

**Base de datos de una clínica médica**

**Materia: Bases de Datos**

**Profesor: Nicolás Quiros**

**Integrantes: Flores Leandro, Aciar Agustin, Vera Santiago, Ramirez Rodrigo, Franco Camila.**

# Instrucciones generales

* Este trabajo debe realizarse en **grupos de 4 a 6 integrantes**.
* Deberán **presentar los resultados en una exposición oral de hasta 10 minutos**, donde expliquen las decisiones tomadas, el diseño de la base de datos, los resultados obtenidos y las justificaciones técnicas.
* El trabajo está basado en un escenario realista y deben resolverlo aplicando los conocimientos adquiridos durante la cursada.

# 

# Contexto del caso

Una clínica médica privada desea informatizar la gestión de atención a pacientes. Se requiere un sistema que permita registrar pacientes, turnos médicos, especialidades, tratamientos realizados y resultados de consultas. Además, debe facilitar el análisis del rendimiento médico, cantidad de turnos por especialidad, y seguimiento de pacientes crónicos. También se necesita filtrar la información según especialidades específicas.

Asumiremos que:

* Cada paciente puede tener una sola obra social, o bien puede no tener ninguna.
* Cada médico puede tener una sola especialidad.

*Todas las asumpciones que no estén listadas arriba se pueden manejar como deseen, siempre y cuando las detallen en su presentación oral.*

# Campos requeridos (a distribuir en tablas)

Se provee una lista de campos obligatorios junto con su tipo de dato que deberán formar parte del modelo. Cada grupo decidirá cómo distribuirlos en diferentes tablas y si debe renombrarlos o no. Podrán agregar campos si lo consideran necesario:

|  |  |  |
| --- | --- | --- |
| **Nombre del campo** | **Tipo de dato** | **Descripción** |
| id\_paciente | INT | Identificador único del paciente. |
| nombre\_paciente | VARCHAR | Primer nombre del paciente. |
| apellido\_paciente | VARCHAR | Apellido nombre del paciente. |
| dni\_paciente | INT | Documento nacional de identidad del paciente. |
| email\_paciente | VARCHAR | Dirección de correo electrónico del paciente. |
| telefono\_paciente | VARCHAR | Número de teléfono del paciente. |
| fecha\_nacimiento\_paciente | DATE | Fecha de nacimiento del paciente. |
| obra\_social\_paciente | VARCHAR | Obra social del paciente. |
| numero\_afiliado\_paciente | VARCHAR | Número de afiliado del paciente en su obra social. |
| historia\_paciente | VARCHAR | Historia clínica del paciente. |
| tiene\_enfermedad\_cronica\_paciente | VARCHAR | Booleano que adquiere valor de “True” si el paciente tiene una enfermedad crónica y “False” en caso contrario. |
| id\_medico | INT | Identificador único del médico. |
| nombre\_medico | VARCHAR | Primer nombre del médico. |
| apellido\_medico | VARCHAR | Apellido nombre del médico. |
| matricula\_medico | VARCHAR | Número de matrícula del médico. |
| especialidad\_medico | VARCHAR | Especialidad del médico. |
| id\_turno | INT | Identificador único del turno médico. |
| fecha\_turno | DATE | Fecha en la que el paciente asistirá o asistió al turno médico. |
| hora\_turno | TIME | Horario en el que el paciente asistirá o asistió al turno médico. |
| estado\_turno | VARCHAR | Estado del turno médico. Sólo puede tomar los valores “Asignado”, “Atendido” o “Cancelado”. |
| motivo\_turno | VARCHAR | Motivo por el cual el paciente asiste al turno médico. |
| id\_secretario | VARCHAR | Identificador único del secretario a cargo de asignar el tuno médico. |
| nombre\_secretario | VARCHAR | Primer nombre del secretario. |
| apellido\_secretario | VARCHAR | Apellido del secretario. |
| sede\_secretario | VARCHAR | Sede de la clínica donde trabaja el secretario (Ciudad de Mendoza, Luján de Cuyo, etc). |

# Actividades

## 1. Diseño conceptual y lógico

1. Definir qué tablas van a construir. Especificar para cada tabla:
   * Su(s) clave(s) primaria(s).
   * Su(s) clave(s) foránea(s).
   * Qué restricciones o contraints se deben utilizar. Por ejemplo: “el campo X no puede ser nulo, el campo Y solo puede tener estos valores determinados, etc”.
   * Su forma normal. Justificar.
   * Decidir si representa un hecho o una dimensión. Justificar.
2. Realizar el diagrama entidad-relación (ER), incluyendo:
   * Entidades.
   * Al menos 3 atributos por entidad.
   * Relaciones, detallando cardinalidad y ordinalidad.

Se permite y sugiere trabajar con datos inventados para ejemplificar el funcionamiento de las estructuras propuestas.

## 1. Diseño conceptual y lógico

### **Tablas principales**

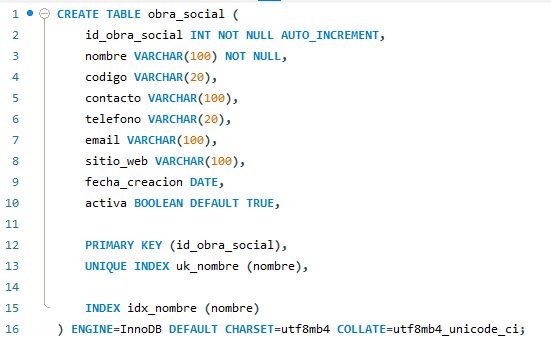
* Clave primaria: id\_paciente.
* Claves foráneas: id\_obra\_social → obra\_social.id\_obra\_social.
* Restricciones: dni y email son UNIQUE, nombre, apellido, dni y fecha\_nac son NOT NULL.
* Forma normal: 3FN (todos los atributos dependen solo de la clave primaria y no entre ellos).
* Tipo: Dimensión (describe entidades, no hechos).

#### 

#### 

#### Tabla**: obra\_social**

* Clave primaria: id\_obra\_social.
* Restricciones: nombre es UNIQUE y NOT NULL.
* Forma normal: 3FN.
* Tipo: Dimensión.

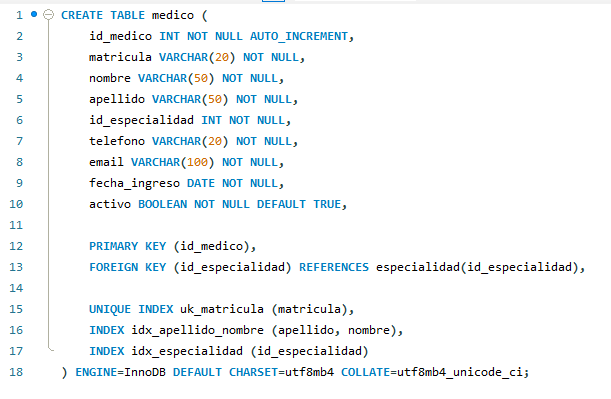


Explicación detallada:

1. **Clave primaria**:
   * id\_obra\_social es autoincremental y NOT NULL.
2. **Restricciones**:
   * nombre tiene restricción UNIQUE y NOT NULL.
   * Se creó un índice único (uk\_nombre) para garantizar la unicidad.
3. **3FN (Tercera Forma Normal)**:
   * Todos los atributos dependen directamente de id\_obra\_social.
   * No hay dependencias transitivas entre atributos no clave.
4. **Tipo dimensión**:
   * Almacena datos descriptivos de entidades (obras sociales).
   * Contiene atributos descriptivos como nombre, contacto, teléfono, etc.
5. **Características adicionales**:
   * Campo activa con valor por defecto TRUE para manejar baja lógica.
   * Índice regular en nombre para optimizar búsquedas.
   * Configuración de charset para soporte completo de Unicode.

#### Tabla: **medico**

* Clave primaria: id\_medico.
* Clave foránea: id\_especialidad → especialidad.id\_especialidad.
* Restricciones: matricula es UNIQUE, todos los campos son NOT NULL.
* Forma normal: 3FN.
* Tipo: Dimensión.

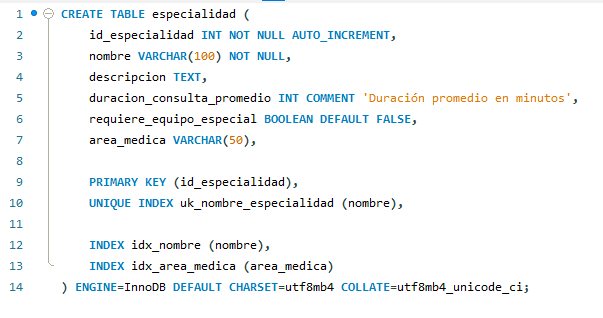


Explicación detallada:

1. **Clave primaria**:
   * id\_medico como INT autoincremental.
2. **Clave foránea**:
   * id\_especialidad referencia a especialidad.id\_especialidad.
3. **Restricciones**:
   * matricula es UNIQUE.
   * Todos los campos son NOT NULL (excepto activo que tiene valor por defecto).
   * Índice único en matrícula (uk\_matricula).
4. **3FN (Tercera Forma Normal)**:
   * Todos los atributos dependen directamente de id\_medico.
   * No hay dependencias transitivas entre atributos no clave.
5. **Optimizaciones**:
   * Índice compuesto en apellido y nombre para búsquedas frecuentes.
   * Índice en especialidad para acelerar JOINs.
   * Campo activo para manejar bajas lógicas.
6. **Tipo dimensión**:
   * Almacena datos descriptivos de entidades (médicos).
   * Contiene todos los atributos identificatorios y descriptivos.

#### Tabla: **especialidad**

* Clave primaria: id\_especialidad.
* Restricciones: nombre es UNIQUE y NOT NULL.
* Forma normal: 3FN.
* Tipo: Dimensión.



Explicación detallada:

1. **Clave primaria**:
   * id\_especialidad como INT autoincremental y NOT NULL.
2. **Restricciones**:
   * nombre es UNIQUE y NOT NULL (garantizado por el índice único uk\_nombre\_especialidad).
   * Todos los demás campos son opcionales (pueden ser NULL).
3. **3FN (Tercera Forma Normal)**:
   * Todos los atributos dependen directamente de id\_especialidad.
   * No hay dependencias transitivas entre atributos no clave.
4. **Tipo dimensión**:
   * Almacena datos descriptivos de entidades (especialidades médicas).
   * Contiene atributos descriptivos como nombre, descripción, duración promedio de consulta, etc.
5. **Optimizaciones**:
   * Índice único en nombre para garantizar unicidad (uk\_nombre\_especialidad).
   * Índice regular en nombre para acelerar búsquedas.
   * Índice en area\_medica para agrupaciones y filtrados.
   * duracion\_consulta\_promedio con comentario para documentación.

Relación con otras tablas:

Esta tabla debe crearse ANTES de la tabla medico ya que esta última contiene una clave foránea que referencia a especialidad.id\_especialidad.

#### Tabla: **turno**

* Clave primaria: id\_turno.
* Claves foráneas: id\_paciente, id\_medico, id\_secretario, id\_estado, id\_sede.
* Restricciones: fecha, hora, estado son NOT NULL, estado con CHECK si no se normaliza.
* Forma normal: 3FN.
* Tipo: Hecho (representa acciones medibles).



#### Explicación detallada:

#### **Clave primaria**:

#### id\_turno autoincremental como identificador único.

#### **Claves foráneas**:

#### Relaciones completas con paciente, médico, secretario, estado y sede.

#### id\_secretario es opcional (puede ser NULL).

#### **Restricciones NOT NULL**:

#### fecha, hora e id\_estado son obligatorios.

#### motivo también es requerido.

#### **Campos adicionales**:

#### duracion\_estimada: Duración predeterminada de 30 minutos.

#### Campos de auditoría: fecha\_registro y fecha\_actualizacion.

#### observaciones para notas adicionales.

#### **Índices**:

#### Optimizados para búsquedas frecuentes.

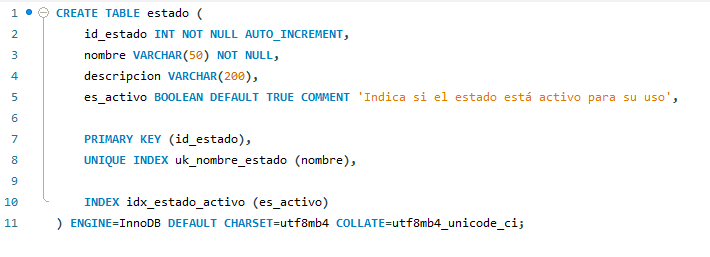
#### Índice compuesto en fecha+hora para gestión de agenda.

#### **Restricción CHECK**:

#### Valida que la hora del turno esté entre las 8 AM y 8 PM.

#### Tabla: **estado**

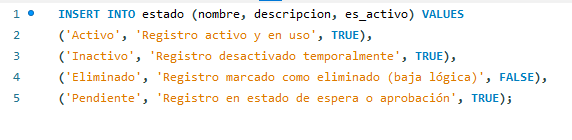
* Clave primaria: id\_estado.
* Restricciones: nombre es UNIQUE y NOT NULL.
* Tipo: Dimensión (catálogo para evitar errores y normalizar).



Explicación detallada:

1. **Estructura básica**:
   * Tabla catálogo/dimensión para normalizar los estados del sistema.
   * Diseño minimalista pero completo para su propósito.
2. **Clave primaria**:
   * id\_estado como entero autoincremental (mejor para JOINs que VARCHAR)
3. **Restricciones**:
   * nombre es UNIQUE y NOT NULL (garantiza consistencia).
   * Índice único para asegurar nombres no repetidos.
4. **Campos adicionales útiles**:
   * descripcion: Para explicar el propósito del estado.
   * es\_activo: Permite desactivar estados sin borrarlos (baja lógica).
5. **Optimizaciones**:
   * Índice en es\_activo para filtrar estados activos eficientemente.
   * Charset UTF8MB4 para soportar caracteres especiales.

EJEMPLO DE DATOS INICIALES

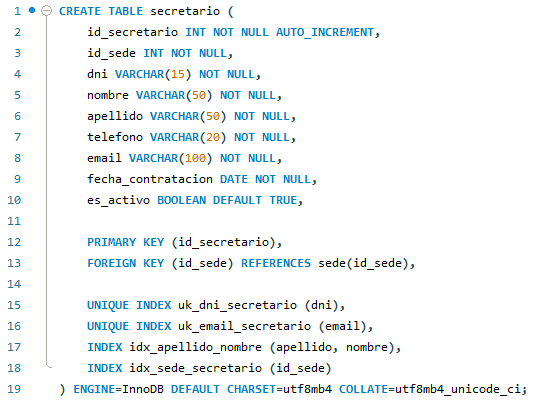


Esta tabla es ideal para:

* Normalizar estados de pacientes, citas, usuarios, etc.
* Evitar errores de tipeo en los estados.
* Facilitar mantenimiento y reportes.
* Permitir internacionalización futura.

#### Tabla: **secretario**

* Clave primaria: id\_secretario.
* Clave foránea: id\_sede.
* Tipo: Dimensión

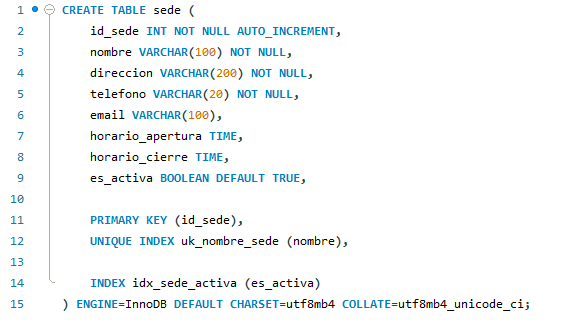


Para la tabla secretario:

* **Clave primaria**: id\_secretario autoincremental.
* **Clave foránea**: id\_sede que referencia a sede.id\_sede.
* **Restricciones**:
  + Campos obligatorios para datos básicos.
  + DNI y email únicos para evitar duplicados.
* **Índices**: Optimizados para búsquedas comunes (por sede, apellido+nombre).

#### Tabla: **sede**

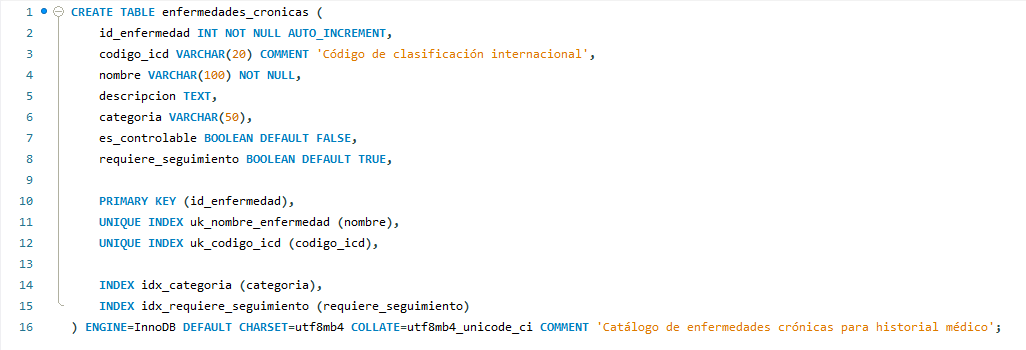
* Clave primaria: id\_sede
* Tipo: Dimensión



Para la tabla sede:

* **Clave primaria**: id\_sede autoincremental.
* **Diseño**: Tabla dimensión con datos completos de ubicaciones físicas.
* **Campos**: Información de contacto, horarios y estado.
* **Índices**: Optimización para búsquedas por nombre y estado.

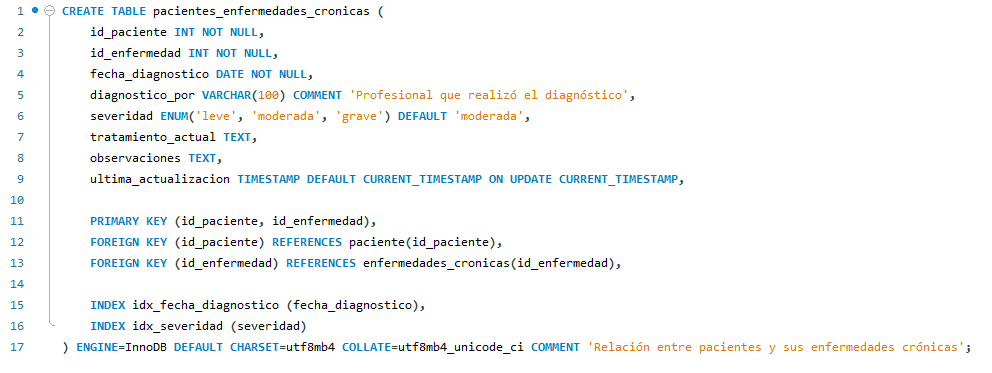
#### Tabla: **enfermedades\_cronicas**

* Clave primaria: id\_enfermedad.
* Tipo: Dimensión.

Explicación detallada:

1. **Estructura básica**:
   * Tabla catálogo/dimensión para enfermedades crónicas.
   * Diseño normalizado para integrarse con historiales médicos.
2. **Clave primaria**:
   * id\_enfermedad como entero autoincremental.
3. **Campos principales**:
   * codigo\_icd: Para clasificación médica estándar (opcional).
   * nombre: Nombre completo de la enfermedad (único).
   * descripcion: Detalles clínicos relevantes.
   * categoria: Grupo de enfermedades (ej. cardiovasculares, metabólicas).
4. **Campos booleanos**:
   * es\_controlable: Si puede controlarse con tratamiento.
   * requiere\_seguimiento: Si necesita monitoreo constante.
5. **Índices**:
   * Únicos para nombre y código ICD.
   * Índices para búsquedas por categoría y seguimiento

#### Tabla: **pacientes\_enfermedades\_cronicas**

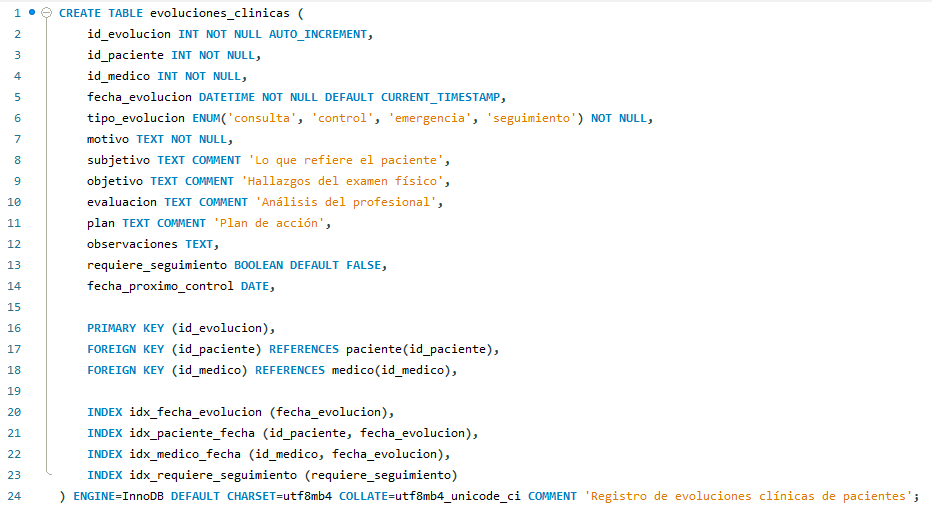
* Clave primaria compuesta: (id\_paciente, id\_enfermedad)
* Tipo: Relación N:M

Para pacientes\_enfermedades\_cronicas:

* **Clave primaria compuesta**: Relación muchos-a-muchos entre pacientes y enfermedades.
* **Campos adicionales**:
  + Fecha de diagnóstico y profesional que lo realizó.
  + Nivel de severidad con valores predefinidos.
  + Tratamiento actual y observaciones.
  + Auditoría con timestamp de última actualización.

#### Tabla: **evoluciones\_clinicas**

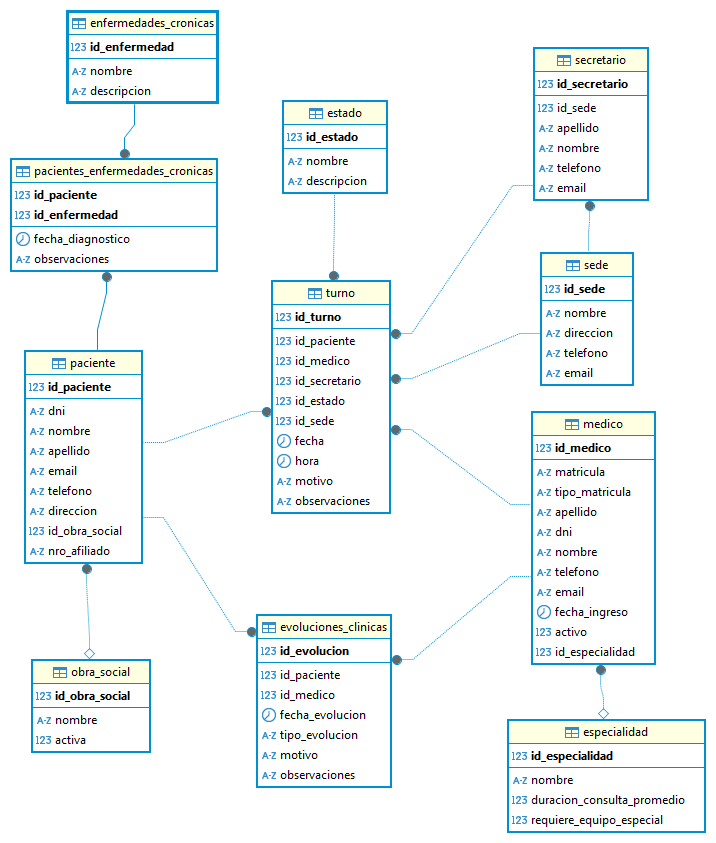
* Clave primaria: id\_evolucion.
* Tipo: Hecho (seguimiento médico).



Para evoluciones\_clinicas:

* **Estructura SOAP**:
  + Subjetivo, Objetivo, Evaluación, Plan (metodología clínica).
* **Relaciones**:
  + Con paciente y médico (ambos obligatorios).
* **Índices**:
  + Optimizados para búsquedas por fechas y profesionales.
  + Índice compuesto para historiales por paciente.

**Diagrama ER**



## 2. Modelado dimensional

1. Partiendo del modelo de entidad-relación realizado en el punto anterior, realizar el modelo de hechos y dimensiones de la base de datos. Incluir en el diagrama:
   * Clave(s) primaria(s) de cada tabla.
   * Clave(s) foránea(s) de cada tabla.
   * Relaciones, detallando cardinalidad y ordinalidad.
2. Explicar si el modelo resultante sigue el esquema de estrella o de copo de nieve. Justificar.
3. Incluir el código SQL de creación de al menos dos tablas relevantes, con sus respectivos tipos de datos y restricciones (claves primarias, claves foráneas, NOT NULL, etc.).

## 

## 2. a) Modelo de hechos y dimensiones:

**Hechos (fact tables):**

* turno
* evoluciones\_clinicas

Estas tablas contienen los eventos o transacciones, con claves foráneas a las dimensiones y atributos como fechas, motivos, observaciones.

**Dimensiones (dimension tables):**

* paciente
* medico
* secretario
* estado
* especialidad (relacionada con médico)

Cada dimensión tiene atributos descriptivos que permiten filtrar, agrupar o analizar los hechos.

Imagen que contiene Escala de tiempo

El contenido generado por IA puede ser incorrecto.

b) **Estructura en estrella:**  
Las tablas de hechos están en el centro, rodeadas por dimensiones directamente conectadas, sin más niveles de normalización en las dimensiones

3. Vistas

1. Crear un código de SQL que genere dos vistas específicas:
   * Una destinada a los médicos, donde se puedan ver todos los datos del paciente.

Vista de turnos para médicos

CREATE OR REPLACE VIEW vista\_turnos\_medicos AS

SELECT

    t.id\_turno,

    t.fecha,

    t.hora,

    t.motivo,

    t.observaciones,

    p.id\_paciente,

    p.dni,

    p.nombre AS paciente\_nombre,

    p.apellido AS paciente\_apellido,

    p.email AS paciente\_email,

    p.telefono AS paciente\_telefono,

    p.direccion AS paciente\_direccion,

    p.nro\_afiliado,

    p.id\_obra\_social,

    m.id\_medico,

    m.nombre AS medico\_nombre,

    m.apellido AS medico\_apellido,

    m.matricula,

    m.id\_especialidad

FROM turno t

JOIN paciente p ON t.id\_paciente = p.id\_paciente

JOIN medico m ON t.id\_medico = m.id\_medico;

Vista de turnos para secretarios

CREATE OR REPLACE VIEW vista\_turnos\_secretarios AS

SELECT

    t.id\_turno,

    t.fecha\_id,

    t.hora,

    p.id\_paciente,

    p.dni,

    p.nombre AS paciente\_nombre,

    p.apellido AS paciente\_apellido,

    p.nro\_afiliado,

    m.id\_medico,

    m.nombre AS medico\_nombre,

    m.apellido AS medico\_apellido,

    m.id\_especialidad

FROM turno t

JOIN paciente p ON t.id\_paciente = p.id\_paciente

JOIN medico m ON t.id\_medico = m.id\_medico;

Vista evoluciones clínicas para médicos

CREATE VIEW vista\_evoluciones\_medicas AS

SELECT

    ec.id\_evolucion,

    ec.fecha\_evolucion,

    ec.tipo\_evolucion,

    ec.motivo,

    ec.observaciones,

    p.id\_paciente,

    p.dni,

    p.nombre AS paciente\_nombre,

    p.apellido AS paciente\_apellido,

    p.email AS paciente\_email,

    p.telefono AS paciente\_telefono,

    p.direccion AS paciente\_direccion,

    p.nro\_afiliado,

    p.id\_obra\_social,

    m.id\_medico,

    m.nombre AS medico\_nombre,

    m.apellido AS medico\_apellido,

    m.matricula,

    m.id\_especialidad

FROM evoluciones\_clinicas ec

JOIN paciente p ON ec.id\_paciente = p.id\_paciente

JOIN medico m ON ec.id\_medico = m.id\_medico;

Vista evoluciones clínicas para secretarios

CREATE VIEW vista\_evoluciones\_secretarios AS

SELECT

    ec.id\_evolucion,

    ec.fecha\_evolucion,

    ec.tipo\_evolucion,

    p.id\_paciente,

    p.dni,

    p.nombre AS paciente\_nombre,

    p.apellido AS paciente\_apellido,

    m.id\_medico,

    m.nombre AS medico\_nombre,

    m.apellido AS medico\_apellido

FROM evoluciones\_clinicas ec

JOIN paciente p ON ec.id\_paciente = p.id\_paciente

JOIN medico m ON ec.id\_medico = m.id\_medico;

Es necesario usar vista en este caso porque estas permiten controlar el acceso a la información sensible (como tratamiento, patologías y notas adicionales medicas) dependiendo del rol del usuario, evitando errores o accesos indebidos a las tablas base.

## 4. Consultas analíticas

Elegir 3 de las siguientes preguntas y creen el código de SQL para responderlas:

* ¿Cuál fue la especialidad con más turnos cancelados en el año 2023?

SELECT

e.nombre AS especialidad,

COUNT(\*) AS total\_cancelados

FROM

turno t

JOIN

estado es ON t.id\_estado = es.id\_estado

JOIN

medico m ON t.id\_medico = m.id\_medico

JOIN

especialidad e ON m.id\_especialidad = e.id\_especialidad

WHERE

es.nombre = 'Cancelado' AND

YEAR(t.fecha) = 2023

GROUP BY

e.nombre

ORDER BY

total\_cancelados DESC

LIMIT 1;

* ¿Qué médicos atendieron a más de 100 pacientes distintos?

SELECT

m.id\_medico,

m.nombre AS medico\_nombre,

m.apellido AS medico\_apellido,

COUNT(DISTINCT t.id\_paciente) AS pacientes\_atendidos

FROM

turno t

JOIN

medico m ON t.id\_medico = m.id\_medico

JOIN estado e ON t.id\_estado = t.id\_estado

WHERE e.nombre = “Confirmado”

GROUP BY

m.id\_medico, m.nombre, m.apellido

HAVING

COUNT(DISTINCT t.id\_paciente) > 100

ORDER BY

pacientes\_atendidos DESC;

* ¿Cuál es el porcentaje de pacientes con enfermedades crónicas?

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

## 5. Visualización de resultados (extra)

Sugerir **qué tipos de visualizaciones gráficas** podrían utilizar para representar los resultados obtenidos en la sección 4. Justificar la elección. Algunos ejemplos de visualización pueden ser:

* Gráfico de barras.
* Gráfico de torta.
* Gráfico de línea.
* Serie temporal.
* Mapa de calor.
* Tabla dinámica.

Para el primer resultado obtenido de la sección 4 sugerimos el gráfico de barras horizontales, permite ver cuál es la especialidad con más turnos cancelados y también las que le siguen con poca diferencia.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Para el segundo resultado sugerimos usar gráfico de columnas o barras verticales:

Gráfico

El contenido generado por IA puede ser incorrecto.

En este se puede ver claramente como todos los médicos superan los 100 pacientes distintos. Además, se aprecia el orden descendente de pacientes únicos según cada médico.

Al igual que en el primer resultado, este tipo de gráficos son muy buenos para mostrar la información de manera sencilla.

Para el último resultado de ejemplo sugerimos utilizar un gráfico de torta:

Gráfico, Gráfico circular

El contenido generado por IA puede ser incorrecto.

Es una forma simple e intuitiva de mostrar porcentajes, y permite ver claramente la diferencia entre los pacientes con enfermedades crónicas y los que no.