

Base de datos para Hospital

•Link al repositorio principal del proyecto:

<https://github.com/JuanFunes9/sql-coderHouse>

Introducción

•Sobre la temática elegida y el fin de la base de datos:

En la siguiente base de datos se intenta crear un sistema dinámico que lleve un registro confiable, eficiente y organizado tanto de pacientes como de doctores. Además de llevar un registro sobre las consultas e internaciones (pasadas y actuales) que transcurren en el hospital.

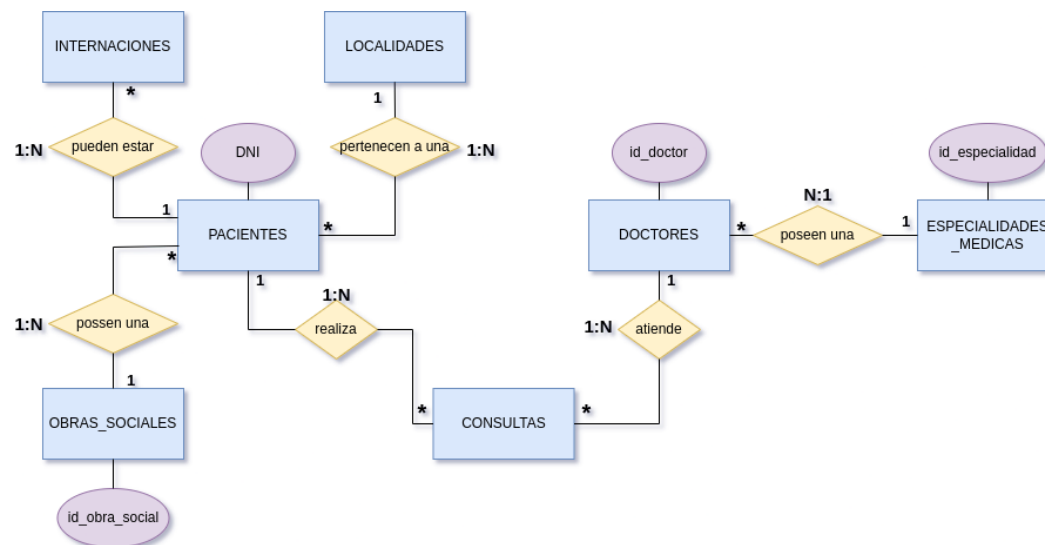
También se guarda información adicional sobre los pacientes (obras sociales, localidad de residencia) y de doctores (qué especialidad médica ejercen en esta institución).

En una institución de la salud es necesario estar ingresando y editando información las 24 hs del día, creando la necesidad de tener un sistema de guardado y gestión de datos que sea eficiente, rápido y confiable. Para conseguir estos requisitos, las bases de datos relacionales son una de las opciones más confiables.

•Objetivo del proyecto y tecnologías usadas:

En el siguiente proyecto mi objetivo es crear una base de datos relacional SQL usando como motor de base de datos MySQL, que tenga las funcionalidades básicas que sean necesarias para este modelo de negocio.

Diagrama entidad-relación



Modelo de tablas y tipos de datos

Tabla:	localidades				
Descripcion tabla:	Contiene los nombres y el codigo postal de cada localidad				
Campo	CLAVE	Tipo	null / not null	Valores restringidos	descripción campo / Observaciones
localidad_id	PK	INT	NOT NULL		id de la tabla, AUTOINCREMENT
nombre		VARCHAR(120)	NOT NULL		Nombre de la localidad
cod_postal		VARCHAR(10)	NOT NULL		codigo postal de la localidad

Tabla:	obras_sociales				
Descripcion tabla:	Contiene informacion sobre las obras sociales				
Campo	CLAVE	Tipo	null / not null	Valores restringidos	descripción campo / Observaciones
obra_social_id	PK	INT	NOT NULL		id de la tabla, AUTOINCREMENT
nombre		VARCHAR(60)	NOT NULL		nombre de la obra social
telefono		VARCHAR(60)	NOT NULL		telefono de la obra social
email		VARCHAR(60)	NOT NULL		email de la obra social

Tabla:	especialidades_medicas				
Descripcion tabla:					
Campo	CLAVE	Tipo	null / not null	Valores restringidos	descripción campo / Observaciones
especialidad_id	PK	INT	NOT NULL		id de la tabla, AUTOINCREMENT
nombre		VARCHAR(60)	NOT NULL		nombre de la especialidad medica

Tabla:	doctores				
Descripcion tabla:					
Campo	CLAVE	Tipo	null / not null	Valores restringidos	descripción campo / Observaciones
id_doctor	PK	INT	NOT NULL		id de la tabla, AUTOINCREMENT
nombre		VARCHAR(60)	NOT NULL		nombre del doctor/a
apellido		VARCHAR(60)	NOT NULL		apellido del doctor/a
matricula		VARCHAR(60)	NOT NULL		matricula oficial de la facultad de medicina
especialidad_id	FK	INT	NOT NULL		especialidad medica

Tabla:	pacientes				
Descripción tabla:	Contiene los pacientes de la base, con información sobre los mismos como nombre, edad, genero, etc.				
Campo	CLAVE	Tipo	null / not null	Valores restringidos	descripción campo / Observaciones
id_paciente	PK	INT	NOT NULL		id de la tabla, AUTOINCREMENT
dni		VARCHAR(60)	NOT NULL		
nombre		VARCHAR(60)	NOT NULL		Nombre del px
apellido		VARCHAR(60)	NOT NULL		Apellido del px
fecha_nacimiento		DATE	NOT NULL		Este campo sirve para calcular la edad actual del px al momento de generar una nueva vista
genero		VARCHAR(20)	NOT NULL		
telefono		VARCHAR(60)	NULL		Puede no tener
email		VARCHAR(60)	NULL		Puede no tener
direccion		VARCHAR(120)	NOT NULL		
obra_social_id	FK	INT	NULL		Puede no tener
localidad_id	FK	INT	NOT NULL		Localidad de residencia

Tabla:	internaciones				
Descripción tabla:	Contiene el registro de todos los pacientes que estuvieron o estan internados				
Campo	CLAVE	Tipo	null / not null	Valores restringidos	descripción campo / Observaciones
internacion_id	PK	INT	NOT NULL		id de la tabla, AUTOINCREMENT
id_paciente	FK	INT	NOT NULL		Id que hace referencia a la tabla de px
ingreso		DATE	NOT NULL		Fecha en que se ingreso al px
cama		INT	NOT NULL		N° de cama en donde estuvo/esta
egreso		DATE	NULL		Fecha del alta. Puede ser null si esta internado actualmente
motivo_ingreso		VARCHAR(255)	NOT NULL		Breve desc del motivo de ingreso (proc. quirurgico, enf. grave, etc.)

Tabla:	consultas				
Descripción tabla:					
Campo	CLAVE	Tipo	null / not null	Valores restringidos	descripción campo / Observaciones
consulta_id	PK	INT	NOT NULL		id de la tabla, AUTOINCREMENT
fecha_consulta		DATE	NOT NULL		
id_paciente	FK	INT	NOT NULL		
id_doctor	FK	INT	NOT NULL		
motivo_consulta		VARCHAR(255)	NOT NULL		Breve desc del motivo de consulta

Vistas

1) Vista **Pacientes_completos**

Esta vista trae toda la información de cada paciente registrado.
Hace el inner join de las tablas de obras_sociales, y localidades

2) Vista de **Consultas_completas**

Esta vista que todas las consultas ordenadas por fecha descendiente.
Se realiza una query a la tabla de consultas y se incluye el inner join de las tablas de pacientes, especialidades_medicas y doctores

3) Vista de **Internaciones_completas**

Esta vista muestra los registros de pacientes internados actualmente. También muestra su edad, género y cuantos días llevan ingresados.
Se realiza la query sobre la tabla de internaciones y el inner join con las tablas de pacientes y obras_sociales

4) Vista de **Doctores_completos**

Vista que muestra la información de todos los doctores
Los ordena según la cantidad de consultas que atendieron en el último tiempo
Hace el inner join con la tabla de especialidades_medicas y una subconsulta a la tabla de consultas.

5) Vista de **Obras_sociales_completas**

Vista que devuelve información de las obras sociales y cuáles son las más elegidas por los pacientes
Hace la query a la tabla de obras_sociales y una subconsulta a la tabla de pacientes.

Funciones

1) Funcion **get_edad_paciente_by_fecha_nac**

Esta función recibe como parámetro el ID del paciente y la fecha de nacimiento del mismo. Luego realiza el cálculo de la edad a la fecha actual y retorna un INTEGER con el valor correspondiente.

2) Funcion **get_cantidad_consultas**

Esta función recibe como parámetro el ID de un doctor y retorna la cantidad de consultas que fueron atendidas por el mismo. Puede ser usada para crear un ranking de los doctores que más consultas atendieron en el último mes, por ejemplo.

Stored procedures

1) Stored procedure: **sp_listado_pacientes_ordenado**

Este SP devuelve la tabla de pacientes ordenada según 2 parámetros de entrada: la columna a ordenar y en que orden ("ASC" o "DESC").

2) Stored procedure: **sp_nueva_internacion**

Este SP realiza el insert de una nueva internación. Antes del INSERT, realiza las siguientes validaciones:

- Que el paciente que se quiere ingresar, es un paciente ya registrado en la DB.
- Que el paciente que se quiere ingresar NO sea un paciente que ya se encuentra internado actualmente.
- Que la cama en que se ingresa al paciente NO se encuentre ya ocupada por otro paciente.

Recibe como parámetros el ID del paciente, el número de cama y el motivo de ingreso.

Triggers

Para la creación de los Triggers, primero cree 2 nuevas tablas para almacenar en ellas los logs: "logs_consultas" y "logs_internaciones".

1) Trigger: **logs_consultas**

Este trigger inserta la información y el detalle de cada consulta a una tabla de logs_consultas

2) Trigger: **update_consultas**

Este trigger deja el registro de las consultas que fueron editadas. Especificando los nuevos valores y que usuario administrador las rea

3) Trigger: **logs_internaciones**

Este trigger inserta la información y el detalle de cada internación a una tabla de logs_internaciones

4) Trigger: **update_internaciones**

El siguiente trigger se ejecuta cuando se le da de alta a un paciente. También inserta la información en la tabla de logs_internaciones. Calcula la cantidad de días que el paciente estuvo ingresado.

Ejemplo de sentencia update:

```
UPDATE internaciones  
SET egreso = curdate()  
WHERE internacion_id = 23;
```