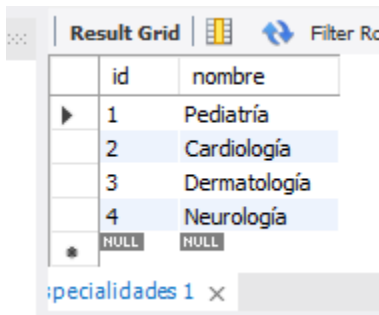


TALLER Sistema con Base de Datos y Java

Parte 1: Configuración de la base de datos

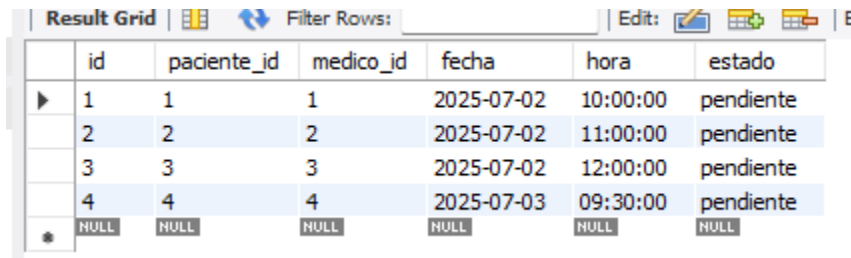
1. Descargar el script SQL proporcionado.
2. Crear la base de datos y las tablas ejecutando el script.
3. Insertar los registros necesarios utilizando INSERT INTO.



Result Grid | Filter Rows:

	id	nombre
▶	1	Pediatría
	2	Cardiología
	3	Dermatología
	4	Neurología
•	NULL	NULL

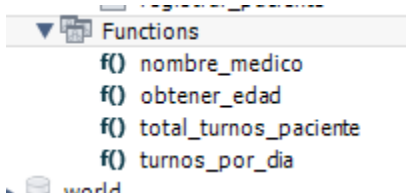
especialidades 1 x



Result Grid | Filter Rows: | Edit: | E

	id	paciente_id	medico_id	fecha	hora	estado
▶	1	1	1	2025-07-02	10:00:00	pendiente
	2	2	2	2025-07-02	11:00:00	pendiente
	3	3	3	2025-07-02	12:00:00	pendiente
	4	4	4	2025-07-03	09:30:00	pendiente
•	NULL	NULL	NULL	NULL	NULL	NULL

4. Crear únicamente las funciones definidas en el script.

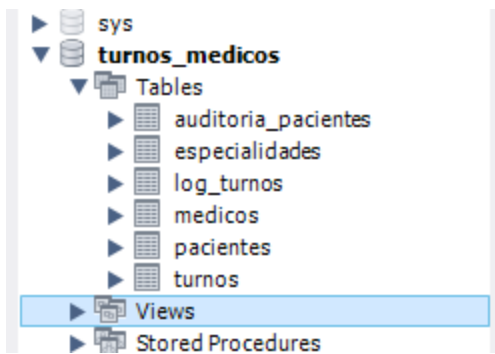


Functions

- f() nombre_medico
- f() obtener_edad
- f() total_turnos_paciente
- f() turnos_por_dia

world

5. Crear únicamente los procedimientos almacenados.



6. Crear únicamente los triggers.

6.1 Verificar que los triggers fueron creados correctamente usando la instrucción:

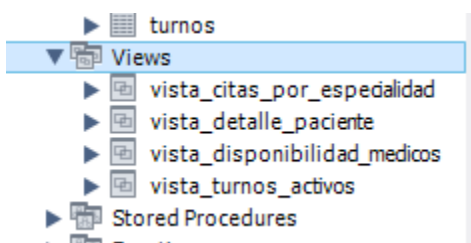
SHOW TRIGGERS;

221
222 • **SHOW TRIGGERS;**

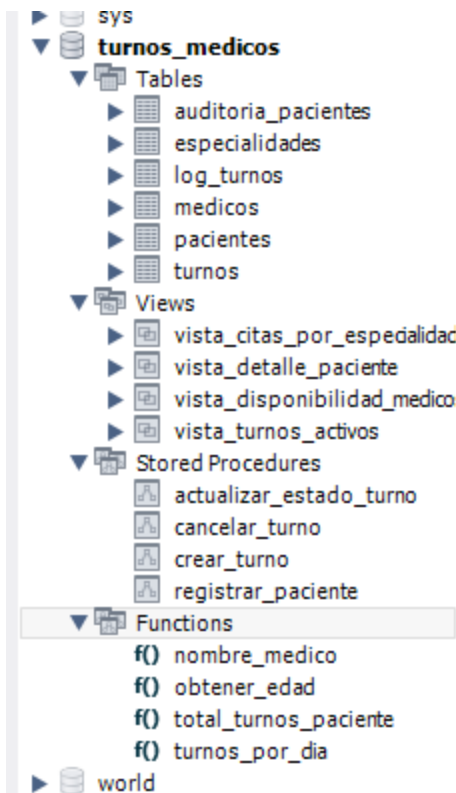
Trigger	Event	Table	Statement	Timing	Created	sql_mode
trg_auditar_paciente_nuevo	INSERT	pacientes	BEGIN INSERT INTO auditoria_pacientes(paci...	AFTER	2025-07-02 11:17:03.14	ONLY_FULL_GROUP_E
trg_validar_fecha_turno	INSERT	turnos	BEGIN IF NEW.fecha < CURDATE() THEN ...	BEFORE	2025-07-02 11:17:03.12	ONLY_FULL_GROUP_E
trg_auto_estado_turno	INSERT	turnos	BEGIN IF NEW.fecha < CURDATE() THEN ...	AFTER	2025-07-02 11:17:03.15	ONLY_FULL_GROUP_E
trg_log_cambios_turnos	UPDATE	turnos	BEGIN INSERT INTO log_turnos(turno_id, acc...	AFTER	2025-07-02 11:17:03.13	ONLY_FULL_GROUP_E

Result 1 x

7. Crear las vistas definidas.



8. Verificar que todos los elementos anteriores (tablas, funciones, procedimientos, triggers y vistas) fueron creados correctamente.



Parte 2: Configuración del proyecto Java en IntelliJ IDEA

1. Abrir IntelliJ IDEA y cargar el proyecto.

2. Restaurar el código Java del sistema.

3. Ir a File → Project Structure → Libraries y añadir el conector MySQL (mysql-connector-java-x.x.xx.jar) previamente descargado.

4. Si aún no lo tienes, descargar el conector MySQL desde el sitio oficial:

<https://dev.mysql.com/downloads/connector/j/>

5. Ejecutar el código Java y comprobar que la conexión con la base de datos funcione.

6. Verificar que el código Java utilice correctamente los elementos de base de datos:

- Vistas

- Funciones

- Procedimientos almacenados

- Triggers

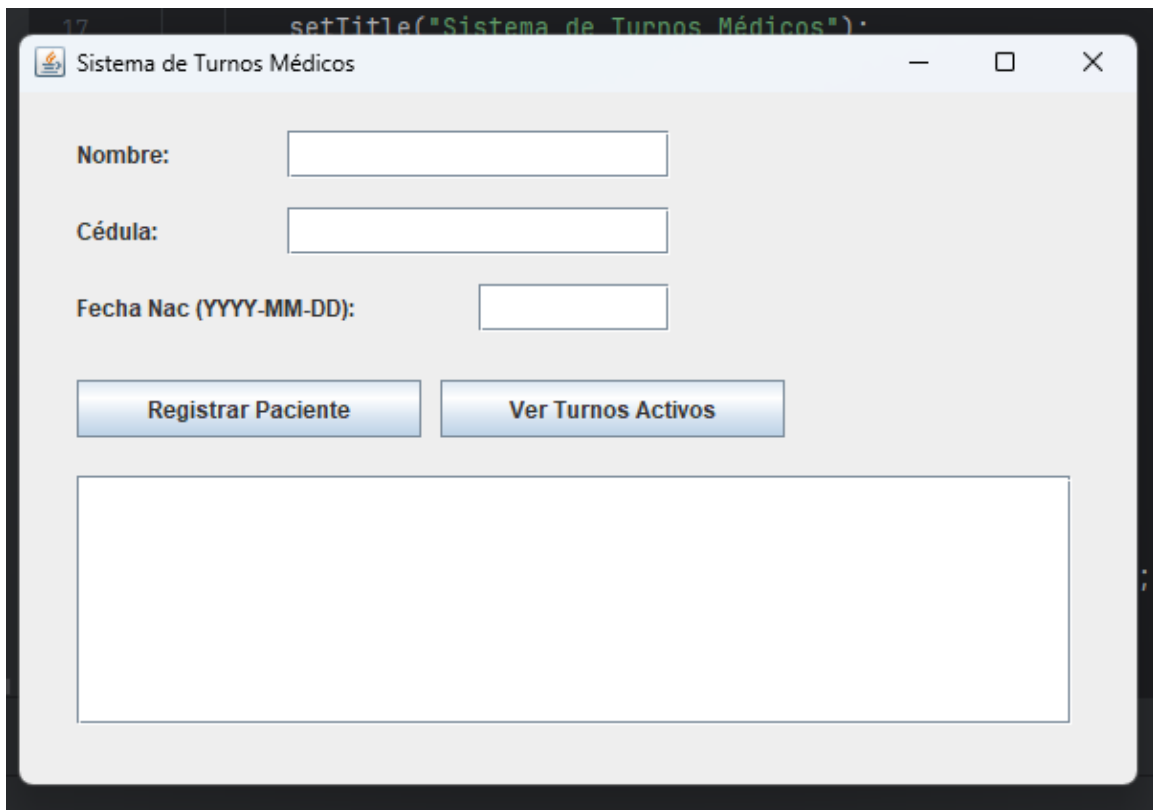
7. Crear usuarios para probar las acciones del trigger

Capturar pantallas.


Nombre: Alan Logroño Soria

GITHUB: https://github.com/Alan-Logrono-S/Taller_Complementario_Logro-o.git

Subir información a git HUB y poner conclusión de la práctica realizada



The screenshot shows a Java Swing window titled "Sistema de Turnos Médicos". The window has a light gray background and a standard Windows-style title bar with minimize, maximize, and close buttons. Inside the window, there are three text input fields for user registration: "Nombre:" (Name), "Cédula:" (ID Card), and "Fecha Nac (YYYY-MM-DD):" (Birth Date). Below these fields are two buttons: "Registrar Paciente" (Register Patient) and "Ver Turnos Activos" (View Active Shifts). At the bottom of the window is a large, empty rectangular area, likely intended for displaying a list of active shifts or a confirmation message. The window is set against a dark background, and a portion of a code editor is visible at the top, showing the line `setTitle("Sistema de Turnos Médicos");`.


 Sistema de Turnos Médicos

Nombre:

Cédula:

Fecha Nac (YYYY-MM-DD):

Paciente registrado correctamente.

 Sistema de Turnos Médicos

Nombre:

Cédula:

Fecha Nac (YYYY-MM-DD):

Paciente registrado correctamente.

Sistema de Turnos Médicos

Nombre:

Pepe Rito

Cédula:

1617846259

Fecha Nac (YYYY-MM-DD):

2002-03-05

Registrar Paciente

Ver Turnos Activos

ID: 1 - Paciente: María López - Médico: Dra. Ana Torres - Fecha: 2025-07-02

ID: 2 - Paciente: Pedro González - Médico: Dr. Luis Pérez - Fecha: 2025-07-02

ID: 3 - Paciente: Lucía Martínez - Médico: Dra. Carla Gómez - Fecha: 2025-07-02

ID: 4 - Paciente: Carlos Herrera - Médico: Dr. Jorge Lima - Fecha: 2025-07-03

	id	nombre	cedula	fecha_nacimiento
▶	1	María López	1102233445	1990-04-15
	2	Pedro González	1103344556	1985-06-20
	3	Lucía Martínez	1104455667	2002-09-10
	4	Carlos Herrera	1105566778	1978-12-05
	5	Pepe Rito	1617846259	2002-03-05
	6	Maria Antonieta	0976123292	2003-10-11
✱	NULL	NULL	NULL	NULL

	id	nombre	cedula	fecha_nacimiento
▶	1	María López	1102233445	1990-04-15
	2	Pedro González	1103344556	1985-06-20
	3	Lucía Martínez	1104455667	2002-09-10
	4	Carlos Herrera	1105566778	1978-12-05
	5	Pepe Rito	1617846259	2002-03-05
✱	NULL	NULL	NULL	NULL

	id	paciente_id	fecha_auditoria	descripcion
▶	1	5	2025-07-02 11:43:42	Nuevo paciente registrado: Pepe Rito
	2	6	2025-07-02 11:47:54	Nuevo paciente registrado: Maria Antonieta
✱	NULL	NULL	NULL	NULL

Sistema de Turnos Médicos

Nombre:

Cédula:

Fecha Nac (YYYY-MM-DD):

ID: 1 - Paciente: María López - Médico: Dra. Ana Torres - Fecha: 2025-07-02
ID: 2 - Paciente: Pedro González - Médico: Dr. Luis Pérez - Fecha: 2025-07-02
ID: 3 - Paciente: Lucía Martínez - Médico: Dra. Carla Gómez - Fecha: 2025-07-02
ID: 4 - Paciente: Carlos Herrera - Médico: Dr. Jorge Lima - Fecha: 2025-07-03
ID: 7 - Paciente: Maria Antonieta - Médico: Dr. Jorge Lima - Fecha: 2025-07-21

CONCLUSION:

Se puede concluir la forma en la que trabajan el manejo de las bases de datos en IntelIDE y MySQLWorkbench, nos ayuda con el manejo y seguridad de los mismo para el desarrollo correcto de la actividad simulando un entorno laboral para su correcto entendimiento.