



EVIDENCIA DE CONOCIMIENTO Y DESEMPEÑO - SQL EXTENDIDO

ACTIVIDADES PARA DESARROLLAR:

Realice cada ejercicio siguiendo las siguientes instrucciones:

- Los ejercicios prácticos deben estar acompañados de explicaciones breves de lo que hace cada parte del código.
- Inserte debajo de cada ejercicio la imagen de cada respuesta con los scripts utilizados para cada ejercicio, puede usar este mismo documento.
- No se puede utilizar ayudas de IA para la solución a menos que se indique en el punto.
- Puede consultar los talleres realizados como ayuda.
- Al terminar el desarrollo de todos los puntos, debe grabar una video-sustentación en el cual se visualice la herramienta de mysql donde realizo cada punto junto con la explicación con sus propias palabras de cómo realizó cada punto y **el resultado de su ejecución según cada ejercicio**. Este vídeo lo debe subir a youtube y pegar el enlace en la columna “enlace vídeo” de [este archivo](#).
- Suba el documento de la evidencia a su portafolio de evidencias.

CASO DE ESTUDIO: Base de datos para un software que permite almacenar el uso de las computadoras por parte estudiantes y profesores de una institución.

1. Creación de la base de datos CitasMedicas

A partir del script cree la base de datos con sus respectivas tablas:

- **Pacientes:** Incluye los datos personales del paciente.
- **Especialidades:** Define las diferentes especialidades médicas.
- **Médicos:** Registra los médicos y su especialidad.
- **Citas:** Registra las citas médicas entre pacientes y médicos, con motivo y fecha de la cita.

```
CREATE DATABASE IF NOT EXISTS CitasMedicas;  
USE CitasMedicas;
```

```
CREATE TABLE Paciente (  
  idPaciente INT PRIMARY KEY AUTO_INCREMENT,
```



Servicio Nacional de Aprendizaje
Formato Taller
Centro de Gestión de Mercados, Logística y Tecnologías de la Información.

```
nombre VARCHAR(50),  
apellido VARCHAR(50),  
fechaNacimiento DATE,  
telefono VARCHAR(15),  
email VARCHAR(50)  
);
```

```
CREATE TABLE Medico (  
    idMedico INT PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(50),  
    apellido VARCHAR(50),  
    idEspecialidad INT,  
    telefono VARCHAR(15),  
    email VARCHAR(50)  
);
```

```
CREATE TABLE Especialidad (  
    idEspecialidad INT PRIMARY KEY AUTO_INCREMENT,  
    nombreEspecialidad VARCHAR(50)  
);
```

```
ALTER TABLE Cita  
ADD idEspecialidad INT;
```

```
CREATE TABLE Cita (  
    idCita INT PRIMARY KEY AUTO_INCREMENT,  
    idPaciente INT,  
    idMedico INT,  
    fechaCita DATETIME,  
    motivo VARCHAR(255),  
    FOREIGN KEY (idPaciente) REFERENCES Paciente(idPaciente),  
    FOREIGN KEY (idMedico) REFERENCES Medico(idMedico)  
);
```

```
ALTER TABLE Cita  
ADD CONSTRAINT fk_especialidad_cita  
FOREIGN KEY (idEspecialidad) REFERENCES Especialidad(idEspecialidad);
```

2. Inserción de Datos coherentes en tablas.



Servicio Nacional de Aprendizaje
Formato Taller
Centro de Gestión de Mercados, Logística y Tecnologías de la Información.

De acuerdo con el script, realice la inserción de los datos en cada tabla:

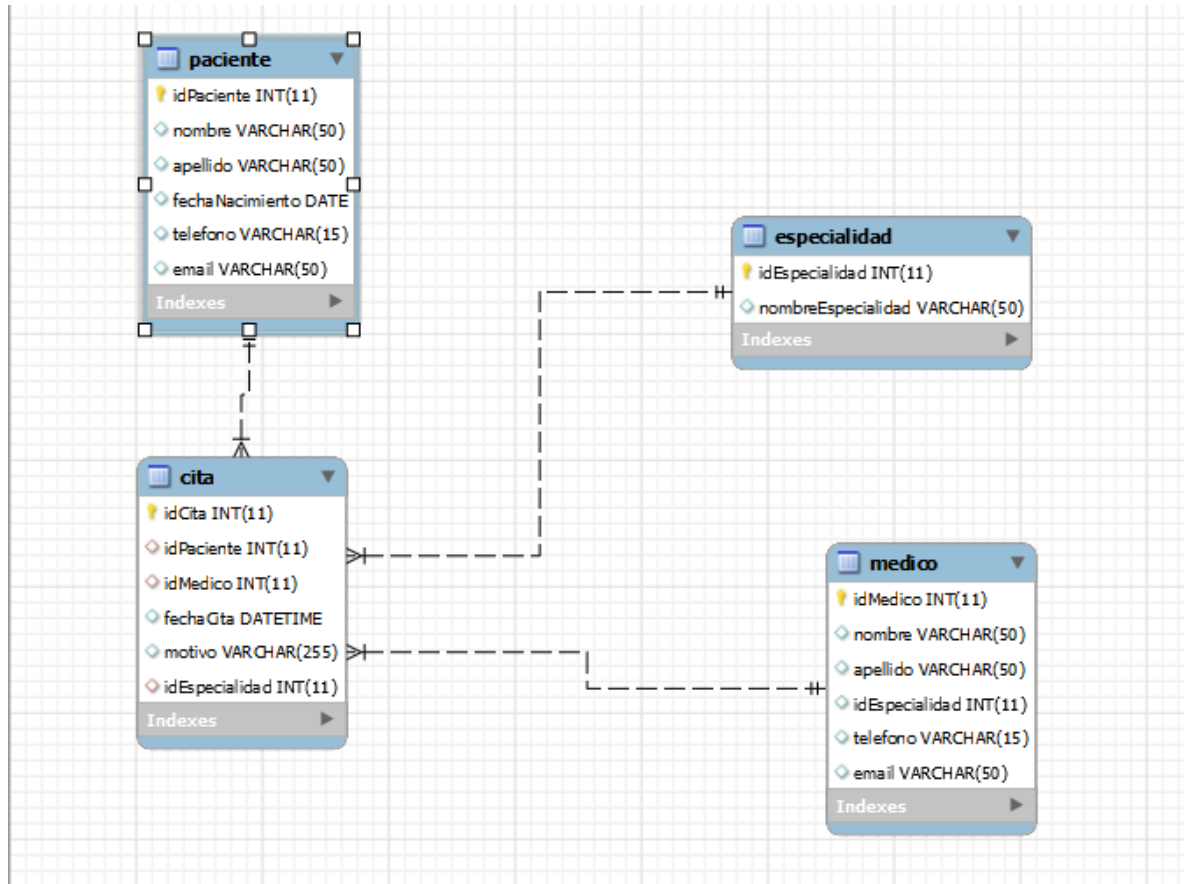
```
INSERT INTO Paciente (nombre, apellido, fechaNacimiento, telefono, email)
VALUES
('Juan', 'Pérez', '1990-05-10', '555-1234', 'juan.perez@mail.com'),
('Ana', 'Gómez', '1985-08-20', '555-5678', 'ana.gomez@mail.com'),
('Carlos', 'Rodríguez', '1992-12-15', '555-9012', 'carlos.rodriguez@mail.com'),
('Laura', 'Fernández', '1978-04-05', '555-3456', 'laura.fernandez@mail.com'),
('Pedro', 'Martínez', '1980-10-30', '555-7890', 'pedro.martinez@mail.com');
```

```
INSERT INTO Especialidad (nombreEspecialidad)
VALUES
('Cardiología'),
('Dermatología'),
('Pediatría'),
('Neurología'),
('Odontología');
```

```
INSERT INTO Medico (nombre, apellido, idEspecialidad, telefono, email)
VALUES
('Dr. Fernando', 'Salazar', 1, '555-4321', 'f.salazar@medico.com'),
('Dra. Isabel', 'López', 2, '555-8765', 'i.lopez@medico.com'),
('Dr. Rafael', 'García', 3, '555-6543', 'r.garcia@medico.com'),
('Dra. Patricia', 'Santos', 4, '555-9876', 'p.santos@medico.com'),
('Dr. Javier', 'Moreno', 5, '555-3459', 'j.moreno@medico.com');
```

```
INSERT INTO Cita (idPaciente, idMedico, fechaCita, motivo, idEspecialidad)
VALUES
(1, 1, '2024-10-01 08:00:00', 'Chequeo de rutina', 1), -- Cardiología
(2, 2, '2024-10-02 10:00:00', 'Revisión de la piel', 2), -- Dermatología
(3, 3, '2024-10-03 14:00:00', 'Control pediátrico', 3), -- Pediatría
(4, 4, '2024-10-04 09:30:00', 'Consulta neurológica', 4), -- Neurología
(5, 5, '2024-10-05 16:00:00', 'Limpieza dental', 5); -- Odontología
```

3. Genere el Diagrama relacional base de datos e insértelo como imagen debajo de este punto



4. Desarrolle cada punto siguiente:

1. Consultas Multitabla

Crea una consulta que muestre el nombre completo de los pacientes, el nombre del médico que los atendió, la especialidad del médico y la fecha de la cita.

Debe ejecutar el código para comprobar el funcionamiento.



Servicio Nacional de Aprendizaje
Formato Taller
Centro de Gestión de Mercados, Logística y Tecnologías de la Información.

```
74  /*Consultas Multitabla*/
75  •  SELECT
76      P.nombre AS Nombre_Paciente,
77      P.apellido AS Apellido_Paciente,
78      M.nombre AS Nombre_Medico,
79      M.apellido AS Apellido_Medico,
80      E.nombreEspecialidad AS Especialidad,
81      C.fechaCita AS Fecha_Cita
82  FROM
83      Paciente P, Medico M, Especialidad E, Cita C
84  WHERE
```

Nombre_Paciente	Apellido_Paciente	Nombre_Medico	Apellido_Medico	Especialidad	Fecha_Cita
Juan	Pérez	Dr. Fernando	Salazar	Cardiología	2024-10-01 08:00:00
Ana	Gómez	Dra. Isabel	López	Dermatología	2024-10-02 10:00:00
Carlos	Rodríguez	Dr. Rafael	García	Pediatría	2024-10-03 14:00:00
Laura	Fernández	Dra. Patricia	Santos	Neurología	2024-10-04 09:30:00
Pedro	Martínez	Dr. Javier	Moreno	Odontología	2024-10-05 16:00:00

Result 1 x

Output

#	Time	Action	Message
✓ 15	16:30:58	INSERT INTO Medico (nombre, apellido, idEspecialidad, telefono, email) VALUES (Dr. Fern...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
✓ 16	16:31:12	INSERT INTO Cita (idPaciente, idMedico, fechaCita, motivo, idEspecialidad) VALUES (1, 1, '...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0
✓ 17	16:37:31	SELECT P.nombre AS Nombre_Paciente, P.apellido AS Apellido_Paciente, M.nomb...	5 row(s) returned

2. Vistas

Crea una vista que muestre las citas pendientes, incluyendo el nombre del paciente, el nombre del médico y la especialidad. Quizás deba agregar un campo estado a la tabla Cita. El campo estado puede tener valores como 'Pendiente', 'Confirmada', 'Cancelada', 'Finalizada', o cualquier otro valor que quiera usar para indicar el estado de una cita.

Debe ejecutar el código para comprobar el funcionamiento.

```
107 •  CREATE VIEW CitasPendientes AS
108  SELECT
109      P.nombre AS Nombre_Paciente,
110      P.apellido AS Apellido_Paciente,
111      M.nombre AS Nombre_Medico,
112      M.apellido AS Apellido_Medico,
113      E.nombreEspecialidad AS Especialidad,
114      C.fechaCita AS Fecha_Cita,
115      C.estado AS Estado
```

Nombre_Paciente	Apellido_Paciente	Nombre_Medico	Apellido_Medico	Especialidad	Fecha_Cita	Estado
Juan	Pérez	Dr. Fernando	Salazar	Cardiología	2024-10-01 08:00:00	Pendiente
Carlos	Rodríguez	Dr. Rafael	García	Pediatría	2024-10-03 14:00:00	Pendiente



3. Procedimientos Almacenados

Crea un procedimiento almacenado llamado ContarCitasPorMedico que, dado el ID de un médico, devuelva el número total de citas que ha atendido.

Debe probar el funcionamiento del procedimiento llamándolo.

```
130      SELECT COUNT(*)
131      INTO totalCitas
132      FROM Cita
133      WHERE idMedico = medicoID;
134
135 •      SET @totalCitas = 0;
136 •      CALL ContarCitasPorMedico(2, @totalCitas);
137 •      SELECT @totalCitas AS Total_Citas_Atendidas;
138
139
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
Total_Citas_Atendidas				
▶	1			

4. Disparadores (Triggers)

Crea un trigger que registre en una tabla llamada RegistroCitas cada vez que se inserta una nueva cita. Debe registrar el ID del paciente, el ID del médico, la fecha de la cita y la fecha en que se registró la cita.

Debe probar el disparador y las tablas relacionadas antes y después.



ANTES

```
149 • CREATE TRIGGER after_insert_cita
150 AFTER INSERT ON Cita
151 FOR EACH ROW
152
153 INSERT INTO RegistroCitas (idPaciente, idMedico, fechaCita)
154 VALUES (NEW.idPaciente, NEW.idMedico, NEW.fechaCita);
155
156 • SELECT * FROM RegistroCitas;
157
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap
idRegistro	idPaciente	idMedico	fechaCita	fechaRegistro
*	NULL	NULL	NULL	NULL

DESPUES

```
.57
.58
.59 • INSERT INTO Cita (idPaciente, idMedico, fechaCita, motivo, idEspecialidad, estado)
.60 VALUES (2, 1, '2024-10-10 09:00:00', 'Consulta de seguimiento', 1, 'Pendiente');
.61
.62 • SELECT * FROM RegistroCitas;
.63
```

Result Grid	Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
idRegistro	idPaciente	idMedico	fechaCita	fechaRegistro
1	2	1	2024-10-10 09:00:00	2024-09-12 16:54:44
NULL	NULL	NULL	NULL	NULL

5. Eventos

Crea un evento que actualice el estado de las citas a "Finalizada" cuando la fecha de la cita sea anterior a la fecha actual.

Debe probar el evento para que se pueda verificar como estaban los datos antes y después.



Servicio Nacional de Aprendizaje
Formato Taller
Centro de Gestión de Mercados, Logística y Tecnologías de la Información.

ANTES

```
54
55 • CREATE EVENT IF NOT EXISTS actualizar_estado_citas
56 ON SCHEDULE EVERY 1 DAY
57 DO
58     UPDATE Cita
59     SET estado = 'Finalizada'
60     WHERE fechaCita < NOW()
61     AND estado != 'Finalizada';
62
63 • SELECT idCita, fechaCita, estado FROM Cita;
```

idCita	fechaCita	estado
2	2024-10-02 10:00:00	Confirmada
3	2024-10-03 14:00:00	Pendiente
4	2024-10-04 09:30:00	Confirmada
5	2024-10-05 16:00:00	Cancelada
6	2024-10-10 09:00:00	Pendiente

DESPUES

```
175
176 • UPDATE Cita
177 SET estado = 'Finalizada'
178 WHERE idCita IS NOT NULL
179 AND fechaCita < NOW()
180 AND estado != 'Finalizada';
181
182
183 • SELECT idCita, fechaCita, estado FROM Cita;
```

idCita	fechaCita	estado
1	2024-10-01 08:00:00	Pendiente
2	2024-10-02 10:00:00	Confirmada
3	2024-10-03 14:00:00	Pendiente
4	2024-10-04 09:30:00	Confirmada
5	2024-10-05 16:00:00	Cancelada

6. Carga Masiva

Realiza un script para cargar de forma masiva una lista de 20 pacientes desde un archivo CSV generado en chatGpt o herramienta similar que contenga las siguientes columnas: nombre, apellido, fecha de nacimiento, teléfono y email.

Debe probar la carga masiva consultando antes y después de la inserción masiva de datos.

ANTES



Servicio Nacional de Aprendizaje
Formato Taller
Centro de Gestión de Mercados, Logística y Tecnologías de la Información.

Result Grid						
Filter Rows:						
Edit:						
Export/Import:						
	idPaciente	nombre	apellido	fechaNacimiento	telefono	email
▶	1	Juan	Pérez	1990-05-10	555-1234	juan.perez@mail.com
	2	Ana	Gómez	1985-08-20	555-5678	ana.gomez@mail.com
	3	Carlos	Rodríguez	1992-12-15	555-9012	carlos.rodriguez@mail.com
	4	Laura	Fernández	1978-04-05	555-3456	laura.fernandez@mail.com
	5	Pedro	Martínez	1980-10-30	555-7890	pedro.martinez@mail.com

DESPUES

Result Grid						
Filter Rows:						
Edit:						
Export/Import:						
	idPaciente	nombre	apellido	fechaNacimiento	telefono	email
▶	1	Juan	Pérez	1990-05-10	555-1234	juan.perez@mail.com
	2	Ana	Gómez	1985-08-20	555-5678	ana.gomez@mail.com
	3	Carlos	Rodríguez	1992-12-15	555-9012	carlos.rodriguez@mail.com
	4	Laura	Fernández	1978-04-05	555-3456	laura.fernandez@mail.com
	5	Pedro	Martínez	1980-10-30	555-7890	pedro.martinez@mail.com
	106	Juan	Pérez	0000-00-00	555-1234	juan.perez@mail.com
	107	Ana	Gómez	0000-00-00	555-5678	ana.gomez@mail.com
	108	Carlos	Rodríguez	0000-00-00	555-9012	carlos.rodriguez@mail.com
	109	Laura	Fernández	0000-00-00	555-3456	laura.fernandez@mail.com
	110	Pedro	Martínez	0000-00-00	555-7890	pedro.martinez@mail.com

7. Creación de Usuarios

Crea un usuario llamado secretaria con contraseña secretaria123. Dale permisos para insertar y consultar datos en la tabla Cita.

Probar los permisos dados y los que no tiene

CONSULTA.



Servicio Nacional de Aprendizaje
Formato Taller
Centro de Gestión de Mercados, Logística y Tecnologías de la Información.

```
MySQL Model* x EER Diagram x Conexión Secretaria - Warnin... x
Server Tools Scripting Help
Query 1 x
Limit to 1000 rows
1 • use citasmedicas;
2
3 • SELECT * FROM Cita;
```

Result Grid

idCita	idPaciente	idMedico	fechaCita	motivo	idEspecialidad	estado
1	1	1	2024-10-01 08:00:00	Chequeo de rutina	1	Pendiente
2	2	2	2024-10-02 10:00:00	Revisión de la piel	2	Confirmada
3	3	3	2024-10-03 14:00:00	Control pediátrico	3	Pendiente
4	4	4	2024-10-04 09:30:00	Consulta neurológica	4	Confirmada
5	5	5	2024-10-05 16:00:00	Limpieza dental	5	Cancelada
6	2	1	2024-10-10 09:00:00	Consulta de seguimiento	1	Pendiente
NULL	NULL	NULL	NULL	NULL	NULL	NULL

INSERCIÓN DE DATOS



Servicio Nacional de Aprendizaje
Formato Taller
Centro de Gestión de Mercados, Logística y Tecnologías de la Información.

```
3 SELECT * FROM Cita;  
4  
5 INSERT INTO Cita (idPaciente, idMedico, fechaCita, motivo)  
6 VALUES (1, 1, '2024-10-15 10:00:00', 'Prueba de inserción');  
7
```

Output			
Action Output			
#	Time	Action	Message
✓ 1	18:34:10	use citasmedicas	0 row(s) affected
✓ 2	18:34:15	SELECT * FROM Cita LIMIT 0, 1000	6 row(s) returned
✓ 3	18:35:40	INSERT INTO Cita (idPaciente, idMedico, fechaCita, motivo) VALUES (1, 1, '2024-10-15 10:0...	1 row(s) affected

PRUEBA CON UN PERMISO QUE NO TIENE

```
8 DELETE FROM Cita WHERE idCita = 1;  
9
```

Output			
Action Output			
#	Time	Action	Message
✓ 2	18:34:15	SELECT * FROM Cita LIMIT 0, 1000	
✓ 3	18:35:40	INSERT INTO Cita (idPaciente, idMedico, fechaCita, motivo) VALUES (1, 1, '2024-10-15 10...	
✗ 4	18:37:26	DELETE FROM Cita WHERE idCita = 1	