**PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL**

**ACTIVIDAD BESE DE DATOS**

**ANALISIS Y DESARROLLO DE SOFTWARE 228118**

**ALBERT GODOY CRUZ**

**FICHA 2672883**

**MONICA PENAGOS MARTINEZ**

**BASE DE DATOS**

**BOGOTA 2023**

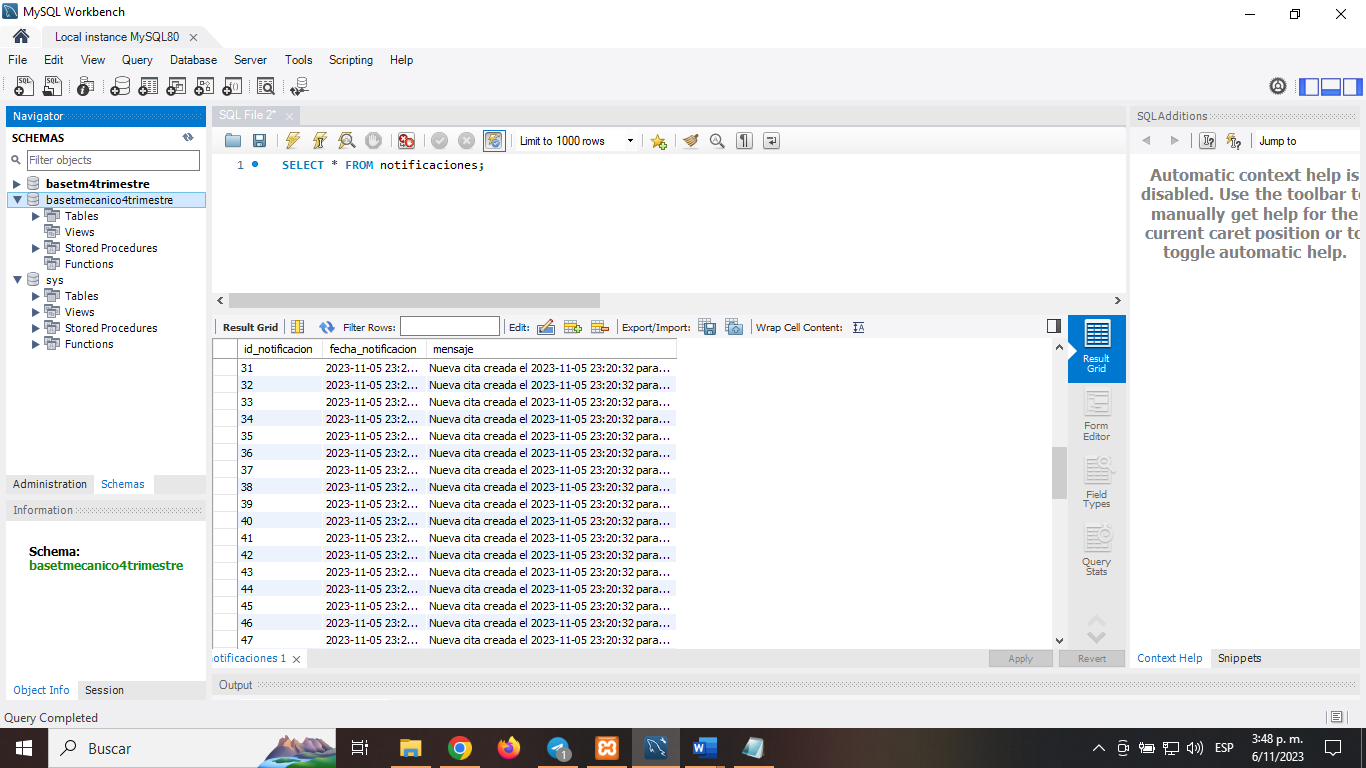
**MODULO CITAS**

1.Este código crea un trigger llamado "after\_insert\_cita" que se activa después de insertar una nueva fila en la tabla

"citas". Cuando se activa, crea un mensaje de notificación que incluye la fecha actual, el cliente y la descripción

del servicio de la nueva cita. Luego, inserta esta notificación en la tabla "notificaciones" con la fecha

de notificación y el mensaje.



DELIMITER //

CREATE TRIGGER after\_insert\_cita

AFTER INSERT ON citas

FOR EACH ROW

BEGIN

DECLARE mensaje\_notificacion TEXT;

SET mensaje\_notificacion = CONCAT('Nueva cita creada el ', NOW(), ' para el cliente ', NEW.cliente, '. Descripción del servicio: ', NEW.descripcion\_servicio);

INSERT INTO notificaciones (fecha\_notificacion, mensaje)

VALUES (NOW(), mensaje\_notificacion);

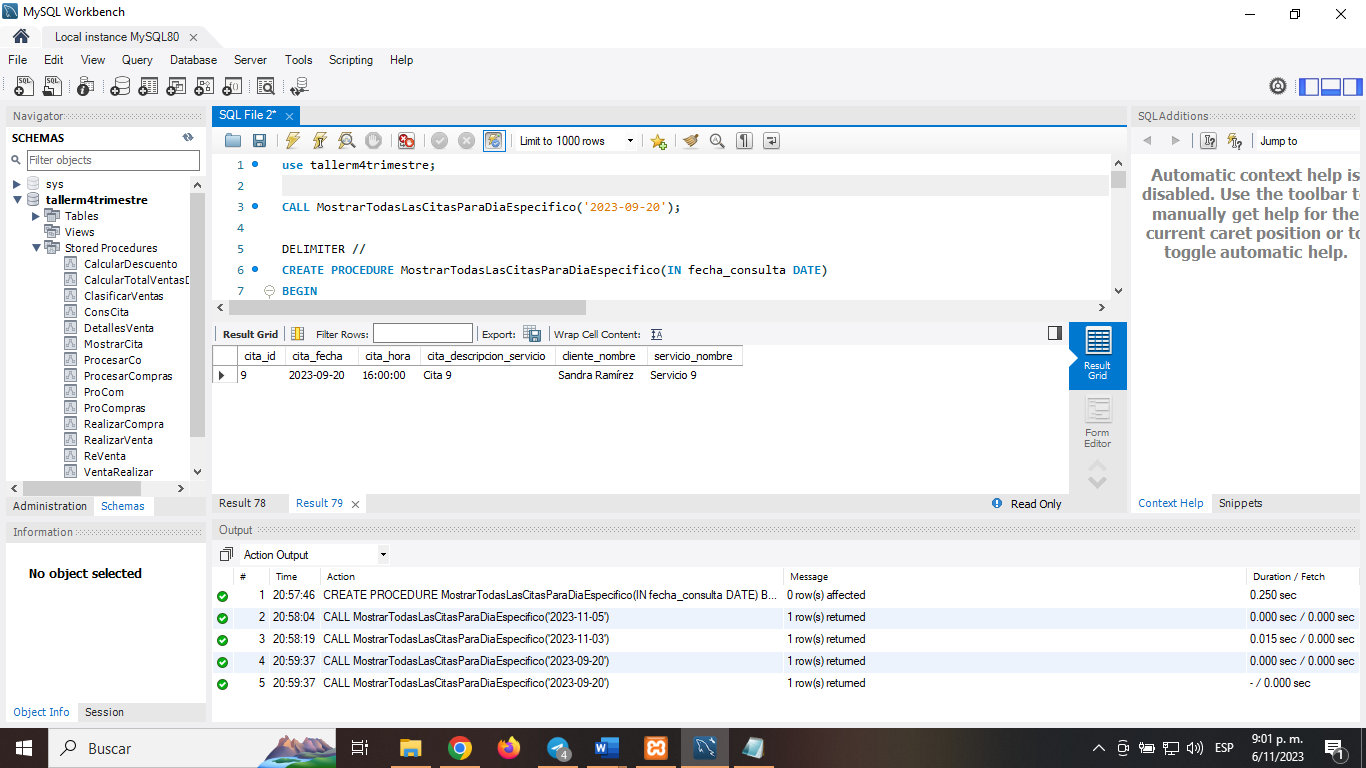
END;

//

DELIMITER ;

SELECT \* FROM notificaciones;

2. Este procedimiento almacenado, llamado "MostrarTodasLasCitasParaDiaEspecifico," toma un parámetro fecha\_consulta, que es la fecha para la cual deseas mostrar las citas. Utiliza un cursor para seleccionar todas las citas que coinciden con la fecha especificada y recorre el conjunto de resultados con un bucle WHILE. Luego, muestra la información de cada cita.



DELIMITER //

CREATE PROCEDURE MostrarTodasLasCitasParaDiaEspecifico(IN fecha\_consulta DATE)

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE cita\_id INT;

DECLARE cita\_fecha DATE;

DECLARE cita\_hora TIME;

DECLARE cita\_descripcion\_servicio TEXT;

DECLARE cliente\_nombre VARCHAR(255);

DECLARE servicio\_nombre VARCHAR(255);

-- Cursor para seleccionar citas para la fecha especificada

DECLARE cita\_cursor CURSOR FOR

SELECT c.id\_cita, c.fecha, c.hora, c.descripcion\_servicio, cl.nombre AS nombre\_cliente, s.nombre AS nombre\_servicio

FROM citas AS c

INNER JOIN clientes AS cl ON c.cliente = cl.id\_cliente

LEFT JOIN servicios AS s ON c.id\_cita = s.cita

WHERE c.fecha = fecha\_consulta;

-- Manejo de errores

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

-- Abrir el cursor

OPEN cita\_cursor;

-- Iniciar el bucle

WHILE done = 0 DO

FETCH cita\_cursor INTO cita\_id, cita\_fecha, cita\_hora, cita\_descripcion\_servicio, cliente\_nombre, servicio\_nombre;

-- Mostrar información de la cita

SELECT cita\_id, cita\_fecha, cita\_hora, cita\_descripcion\_servicio, cliente\_nombre, servicio\_nombre;

END WHILE;

-- Cerrar el cursor

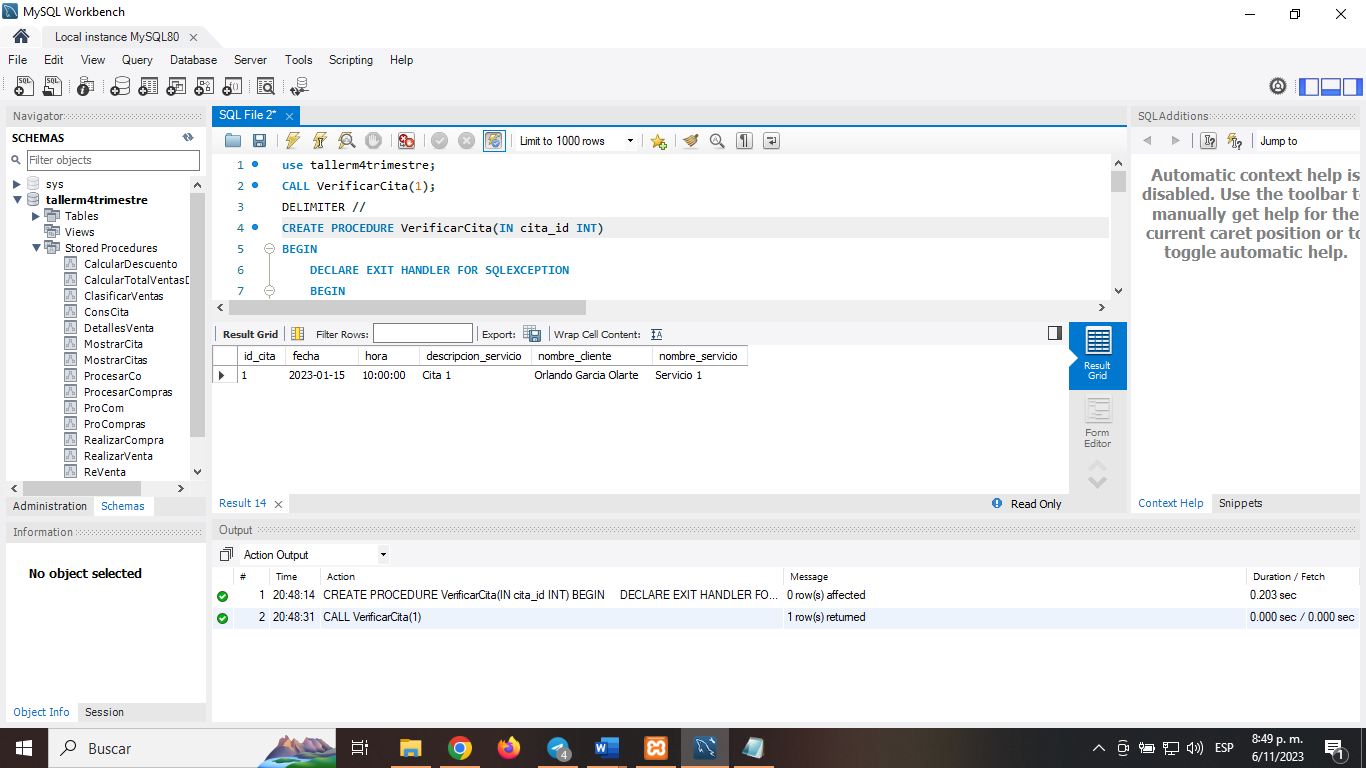
CLOSE cita\_cursor;

END //

DELIMITER ;

CALL MostrarTodasLasCitasParaDiaEspecifico('2023-09-20');

3 Este procedimiento almacenado, llamado "VerificarCita," toma un parámetro cita\_id que es el ID de la cita que deseas verificar. Se utiliza una transacción para evitar errores y se comprueba si la cita existe en la tabla "citas". Si existe, se muestra la información de la cita, el nombre del cliente y el nombre del servicio. Si no existe, se muestra un mensaje de que la cita no existe.



DELIMITER //

CREATE PROCEDURE VerificarCita(IN cita\_id INT)

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

ROLLBACK;

SELECT 'Ocurrió un error durante la operación.';

END;

START TRANSACTION;

-- Verificar si la cita existe

SELECT COUNT(\*) INTO @cita\_count FROM citas WHERE id\_cita = cita\_id;

IF @cita\_count = 1 THEN

-- La cita existe, obtener la información

SELECT c.id\_cita, c.fecha, c.hora, c.descripcion\_servicio, cl.nombre AS nombre\_cliente, s.nombre AS nombre\_servicio

FROM citas AS c

INNER JOIN clientes AS cl ON c.cliente = cl.id\_cliente

LEFT JOIN servicios AS s ON c.id\_cita = s.cita

WHERE c.id\_cita = cita\_id;

ELSE

-- La cita no existe

SELECT 'La cita con el ID ' + CAST(cita\_id AS CHAR) + ' no existe.';

END IF;

COMMIT;

END //

DELIMITER ;

CALL VerificarCita (1);

4. Este procedimiento almacenado se llama "ObtenerCitasCanceladas" y muestra las citas canceladas junto con la fecha de cancelación. La consulta se une a las tablas "citas", "servicios", "estados" y "citas" nuevamente para obtener la fecha de cancelación. Utiliza un LEFT JOIN para obtener todas las citas, incluso las que no han sido canceladas, y muestra la fecha de cancelación cuando esté disponible.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

DELIMITER //

CREATE PROCEDURE ObtenerCitasCanceladas()

BEGIN

SELECT

c.id\_cita,

c.fecha AS fecha\_cita,

c.hora AS hora\_cita,

c.descripcion\_servicio,

cs.fecha AS fecha\_cancelacion

FROM citas c

JOIN servicios s ON c.id\_cita = s.cita

JOIN estados e ON s.venta = e.id\_estado

LEFT JOIN citas cs ON e.en\_espera = 'S' AND cs.id\_cita = c.id\_cita;

END//

DELIMITER ;

CALL ObtenerCitasCanceladas();

5. Este procedimiento almacenado toma el ID de la cita y la nueva fecha como parámetros de entrada. Luego, busca el nombre del cliente asociado a la cita y reprograma la cita en la próxima fecha disponible si cumple con las condiciones. La salida incluye el nombre del cliente, el ID de la cita y la nueva fecha programada.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

DELIMITER //

CREATE PROCEDURE ReprogramarCita(

IN cita\_id INT,

IN nueva\_fecha DATE

)

BEGIN

DECLARE fecha\_disponible DATE;

DECLARE cliente\_nombre VARCHAR(255);

-- Obtener el nombre del cliente asociado a la cita

SELECT c.nombre INTO cliente\_nombre

FROM citas ci

JOIN clientes c ON ci.cliente = c.id\_cliente

WHERE ci.id\_cita = cita\_id;

-- Comprobar si la nueva fecha está dentro del horario disponible para citas

IF DAYOFWEEK(nueva\_fecha) IN (2, 3, 4, 5, 6) AND nueva\_fecha >= CURDATE() THEN

-- Buscar la próxima fecha disponible para la cita

SET fecha\_disponible = nueva\_fecha;

-- Verificar si la fecha ya está ocupada por otra cita

WHILE (SELECT COUNT(\*) FROM citas WHERE fecha = fecha\_disponible) > 0 DO

SET fecha\_disponible = DATE\_ADD(fecha\_disponible, INTERVAL 1 DAY);

END WHILE;

-- Actualizar la fecha de la cita

UPDATE citas

SET fecha = fecha\_disponible

WHERE id\_cita = cita\_id;

SELECT CONCAT('La cita de ', cliente\_nombre, ' con ID ', cita\_id, ' se ha reprogramado para el ', fecha\_disponible) AS mensaje;

ELSE

SELECT 'La fecha seleccionada no está dentro del horario disponible para citas.' AS mensaje;

END IF;

END //

DELIMITER ;

-- Ejecuta el procedimiento para reprogramar la cita con ID 1

CALL ReprogramarCita(1, '2023-11-14');