



TP BASE DE DATOS BIBLIOTECA INFORME

Profesor: Lucas Salvatori

Alumnas: Verónica Moroni; Domene, Melisa

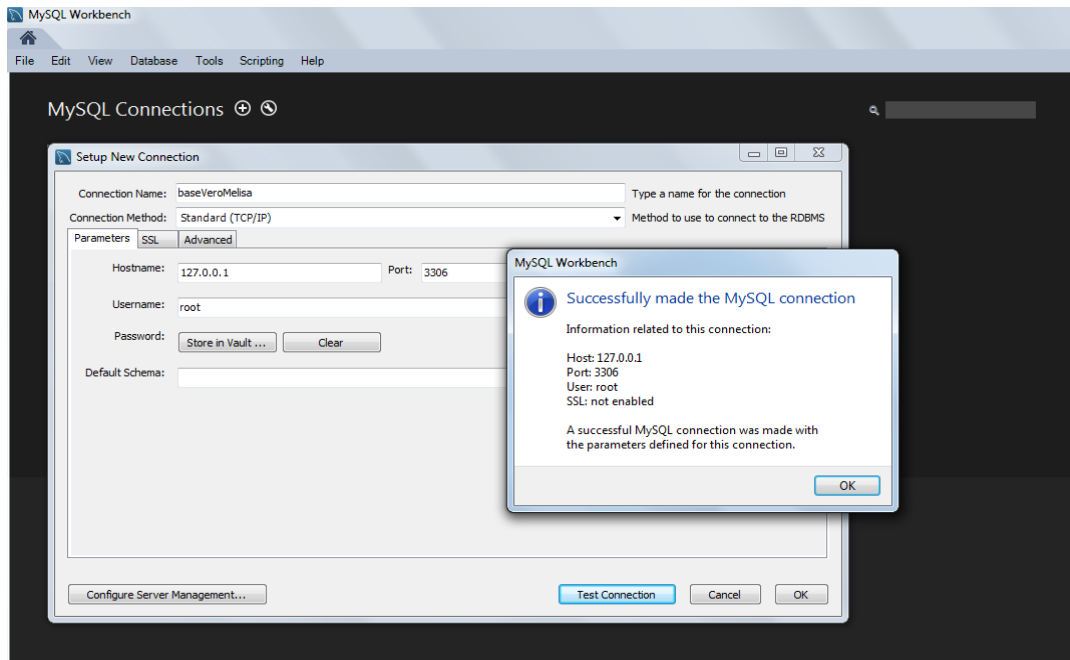
Fecha: 23 de junio de 2024

Índice

Nueva conexión	1
Creación del esquema de la base de datos	3
Creación de tablas dentro del esquema.....	4
Inserción de datos en las tablas.....	6
Procedimientos Almacenados.....	9
Vista.....	10
Disparador.....	12
Transacción.....	14
Bibliografía.....	20

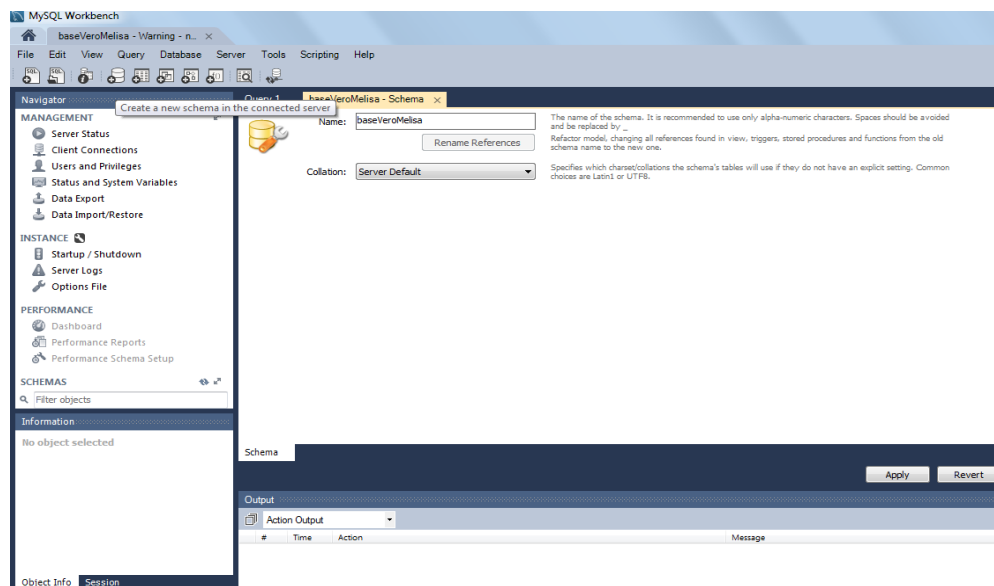
Nueva conexión

En primer lugar, realizamos una nueva conexión en MySQL Workbench. A esta conexión le dimos el nombre: baseVeroMelisa y la creamos sin contraseña. Se probó la conexión y fue exitosa. Guardamos la conexión.

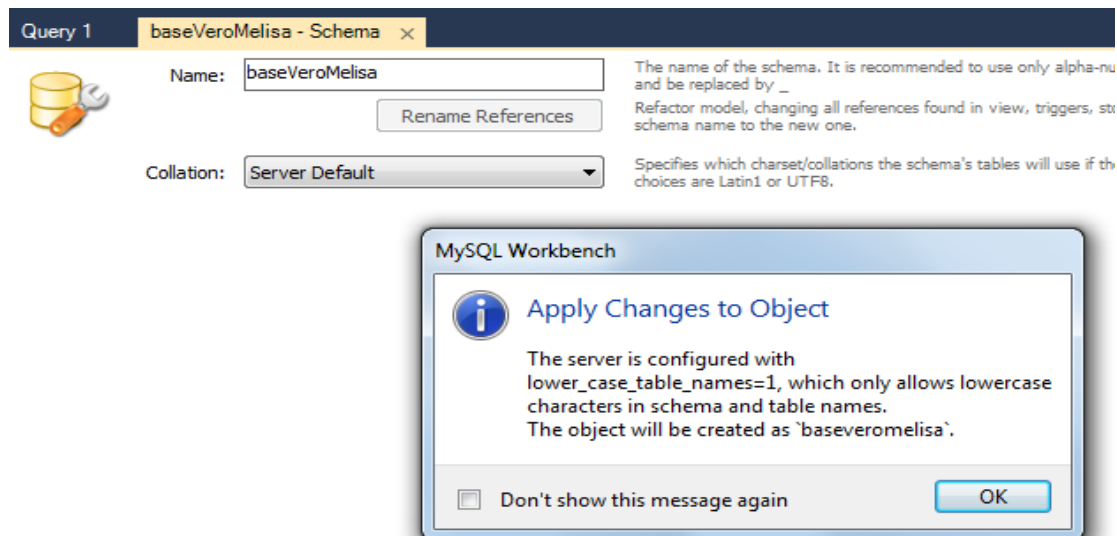


Creación del esquema de la base de datos

Una vez que realizamos la conexión, hicimos doble clic en ella para conectar con el servidor. Hicimos clic en el ícono de nuevo esquema y lo nombramos baseVeroMelisa.



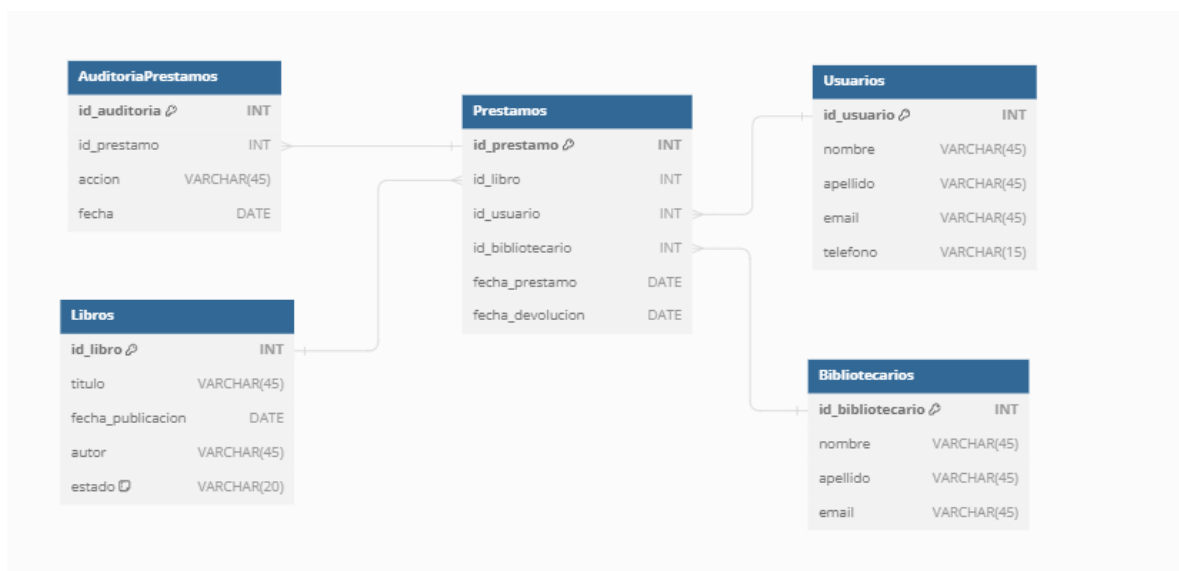
Al hacer clic en aplicar, el programa mostró un mensaje avisándonos que el servidor está configurado para usar la notación snake_case, por lo que cambió el nombre que habíamos elegido a: **baseveromelisa**.



Revisamos el script e hicimos nuevamente clic en aplicar.

Creación de tablas dentro del esquema

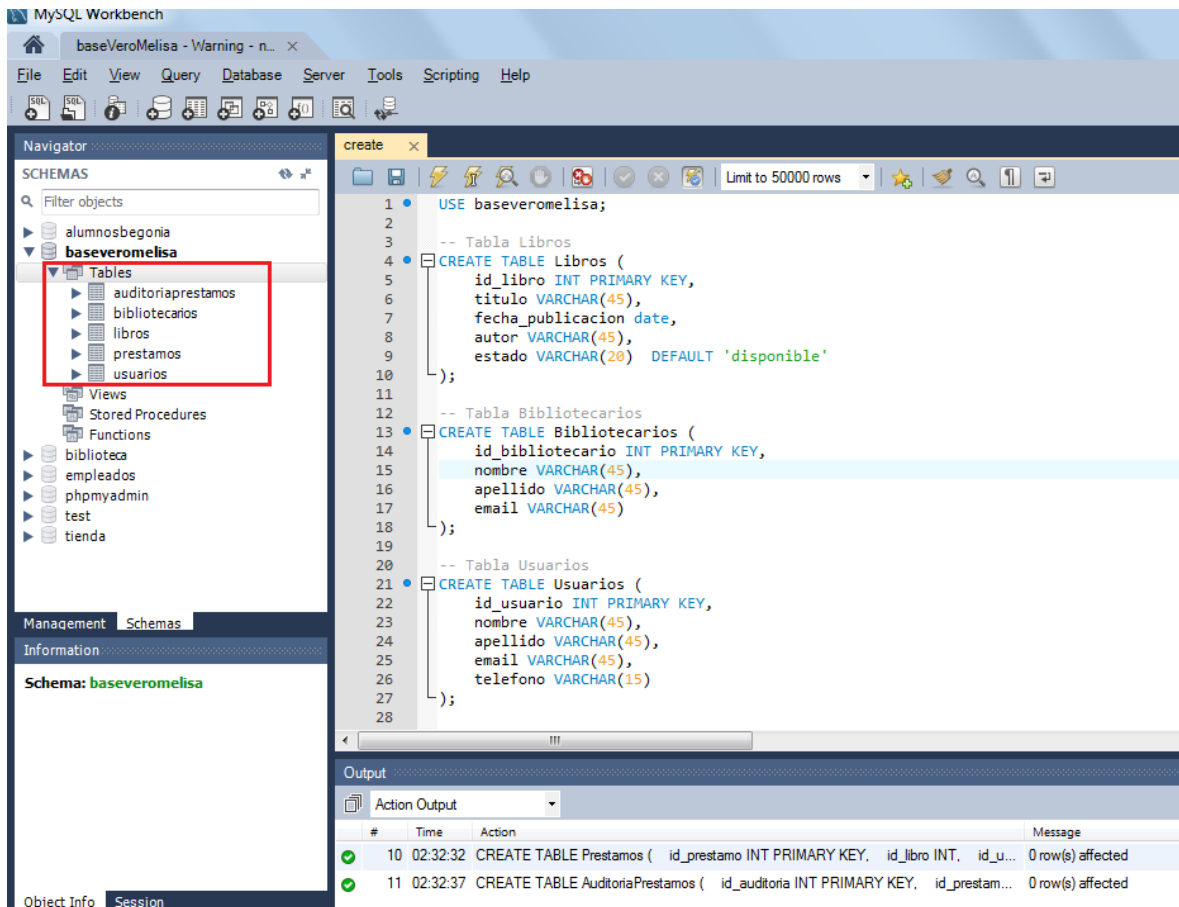
Como se pedía en el enunciado presentado por la cátedra, la creación de las tablas se llevó a cabo por medio de un script. Para este punto, primero pensamos cuáles serían las tablas en lápiz y papel y luego lo hicimos de manera digital usando la página dbdiagram.



Usando baseveromelisa se crean **cinco tablas**.

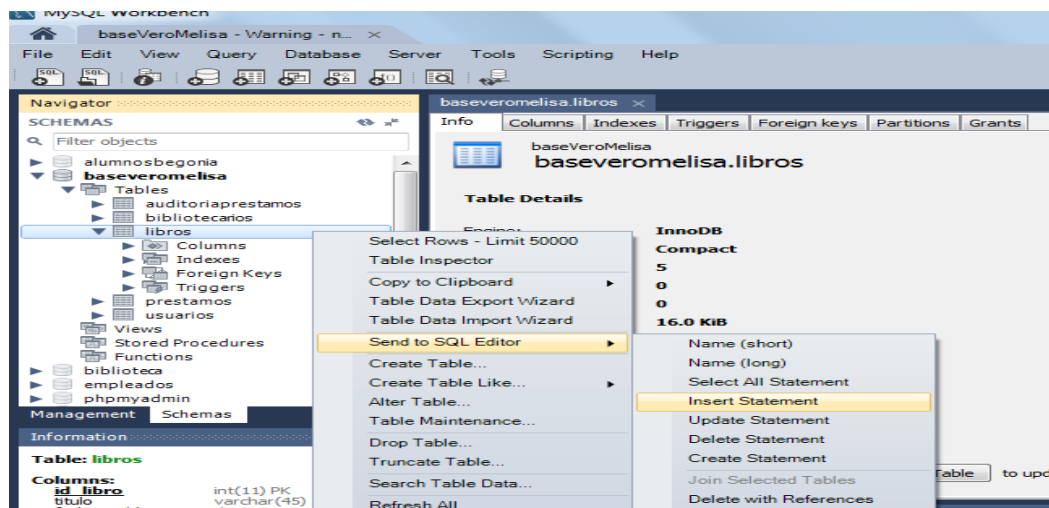
1. La tabla **libros** tiene como llave primaria `id_libro` y luego tiene cuatro atributos más (columnas). `Titulo` es un carácter de longitud variable (de hasta 45 caracteres) que representa el título del libro. `Fecha_publicacion` es del tipo `date` y representa cuál es la fecha en la que se publicó el libro. `Autor` es un carácter de longitud variable de hasta 45 caracteres que describe el nombre del autor del libro. Finalmente, el atributo `estado` es un carácter de longitud variable que se usa para saber si el libro está disponible.
2. La tabla **prestamos** tiene seis columnas. `Id_prestamo` es un número entero que es llave primaria de la tabla. `Id_libro` es un número entero que identifica el libro prestado. `id_usuario` es un número entero que identifica el número de usuario que toma prestado el libro de la biblioteca. `Id_bibliotecario` es un número entero que representa al bibliotecario que recibió al usuario. `Fecha_prestamo` es la fecha en la que se registró el préstamo y `fecha_devolución` es la fecha en que se devuelve el libro. Las fechas son del tipo `date`. `Id_libro`, `id_bibliotecario` y `id_usuario` son llaves foráneas. Ésta tabla tiene el disparador que se activa en caso de una actualización en ella. Este punto se explica en detalle más adelante.
3. La tabla **bibliotecarios** consta de cuatro columnas. `Id_bibliotecario` es la llave primaria que es un número entero que identifica a cada bibliotecario. `Nombre` es un carácter de longitud variable como así también lo es `apellido` y `email` que son los datos personales de cada bibliotecario.
4. La tabla **usuarios** tiene cinco columnas. `Id_usuario` es un número entero que es la llave primaria de la tabla. Identifica a cada usuario de forma única. `Nombre`, `apellido`, `teléfono` y `email` son caracteres de longitud variable que se usan para almacenar información personal de cada usuario de la biblioteca.
5. La tabla **auditoriaprestamos** está formada por cuatro columnas. `Id_auditoria` es su llave primaria que es un número entero. `Id_prestamo` es una llave foránea a la tabla de préstamos. `Acción` es un carácter de longitud variable que puede ser préstamo o devolución y `fecha` es del tipo `date` ya que es la fecha en la que se registró una actividad.

Un bibliotecario puede gestionar muchos préstamos. Un usuario puede pedir muchos libros prestados. Un libro puede prestarse muchas veces. Los préstamos se registran en varias auditorías.



Inserción de datos en las tablas

Para insertar datos en las tablas (para luego poder probar el correcto funcionamiento de lo pedido) se puede hacer clic derecho en la tabla en la que se desea insertar datos e ir al editor con una sentencia de insert:



Debido a que se pedía la entrega del archivo sql insertions, decidimos realizar los inserts desde un nuevo script.

The screenshot shows a database management interface with a 'Navigator' pane on the left displaying the 'baseveromelisa' schema. The main editor shows SQL scripts for inserting data into three tables: 'libros', 'bibliotecarios', and 'usuarios'. The 'Output' pane at the bottom shows the execution results of the first two queries.

```

1 USE baseveromelisa;
2
3 -- Insertar libros
4 INSERT INTO libros (id_libro, titulo, fecha_publicacion, autor, estado) VALUES
5 (1, 'Cien Años de Soledad', '1967-06-05', 'Gabriel Garcia Marquez', 'disponible'),
6 (2, 'Don Quijote de la Mancha', '1605-01-16', 'Miguel de Cervantes', 'prestado'),
7 (3, '1984', '1949-06-08', 'George Orwell', 'disponible'),
8 (4, 'El Principito', '1943-04-06', 'Antoine de Saint-Exupery', 'disponible'),
9 (5, 'La Divina Comedia', '1472-06-08', 'Dante Alighieri', 'prestado'),
10 (6, 'Orgullo y Prejuicio', '1813-01-28', 'Jane Austen', 'disponible'),
11 (7, 'Moby Dick', '1851-10-18', 'Herman Melville', 'disponible'),
12 (8, 'Hamlet', '1600-01-01', 'William Shakespeare', 'disponible'),
13 (9, 'Ulices', '1922-02-02', 'James Joyce', 'prestado'),
14 (10, 'En busca del tiempo perdido', '1913-11-14', 'Marcel Proust', 'disponible');
15
16 -- Insertar bibliotecarios
17 INSERT INTO bibliotecarios (id_bibliotecario, nombre, apellido, email) VALUES
18 (1, 'Ana', 'Perez', 'ana.perez@biblioteca.com'),
19 (2, 'Juan', 'Martinez', 'juan.martinez@biblioteca.com'),
20 (3, 'Lucia', 'Garcia', 'lucia.garcia@biblioteca.com');
21
22 -- Insertar usuarios
23 INSERT INTO usuarios (id_usuario, nombre, apellido, email, telefono) VALUES
24 (1, 'Carlos', 'Gonzalez', 'carlos.gonzalez@usuario.com', '555-1234'),
25 (2, 'Maria', 'Lopez', 'maria.lopez@usuario.com', '555-5678'),
26 (3, 'Jose', 'Hernandez', 'jose.hernandez@usuario.com', '555-9101'),
27 (4, 'Laura', 'Fernandez', 'laura.fernandez@usuario.com', '555-1122');
  
```

Output:

#	Time	Action	Message
73	05:01:12	INSERT INTO prestamos (id_prestamo, id_libro, ...)	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0
74	05:01:12	INSERT INTO AuditoriaPrestamos (id_auditori...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0

Comprobamos que los datos se cargaron de forma correcta, revisando el contenido de cada tabla.

Libros con datos:

The screenshot shows the 'libros' table in the database. The table contains 10 rows of data, including titles like 'Cien Años de Soledad', 'Don Quijote de la Mancha', '1984', 'El Principito', 'La Divina Comedia', 'Orgullo y Prejuicio', 'Moby Dick', 'Hamlet', 'Ulices', and 'En busca del tiempo perdido'.

id_libro	titulo	fecha_publicacion	autor	estado
1	Cien Años de Soledad	1967-06-05	Gabriel Garcia Marquez	disponible
2	Don Quijote de la Mancha	1605-01-16	Miguel de Cervantes	prestado
3	1984	1949-06-08	George Orwell	disponible
4	El Principito	1943-04-06	Antoine de Saint-Exupery	disponible
5	La Divina Comedia	1472-06-08	Dante Alighieri	prestado
6	Orgullo y Prejuicio	1813-01-28	Jane Austen	disponible
7	Moby Dick	1851-10-18	Herman Melville	disponible

Prestamos con datos:

	id_prestamo	id_libro	id_usuario	id_bibliotecario	fecha_prestamo	fecha_devolucion
1	1	2	1	1	2024-06-01	2024-06-15
2	2	5	2	2	2024-06-05	2024-06-19
3	3	9	3	3	2024-06-10	NULL

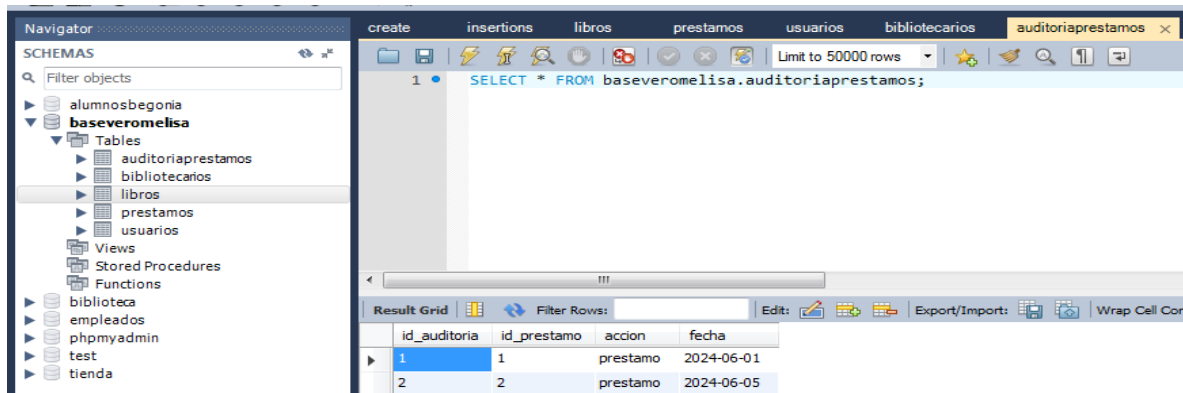
Usuarios con datos:

	id_usuario	nombre	apellido	email	telefono
1	1	Carlos	Gonzalez	carlos.gonzalez@usuario.com	555-1234
2	2	Maria	Lopez	maria.lopez@usuario.com	555-5678
3	3	Jose	Hernandez	jose.hernandez@usuario.com	555-9101
4	4	Laura	Fernandez	laura.fernandez@usuario.com	555-1122

Bibliotecarios con datos:

	id_bibliotecario	nombre	apellido	email
1	1	Ana	Perez	ana.perez@biblioteca.com
2	2	Juan	Martinez	juan.martinez@biblioteca.com
3	3	Lucia	Garcia	lucia.garcia@biblioteca.com
4	NULL	NULL	NULL	NULL

AuditoriaPrestamos con datos:



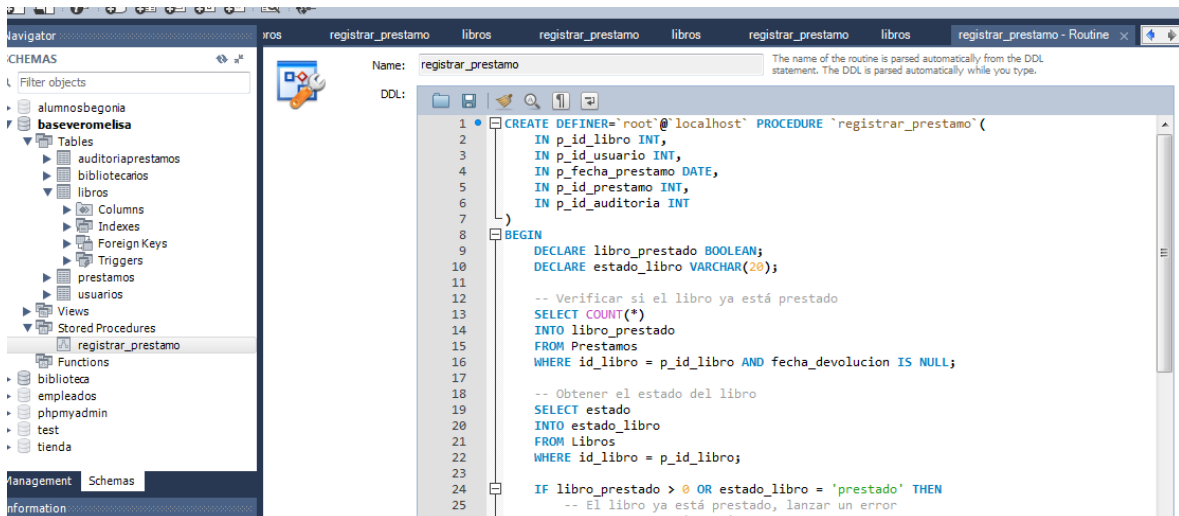
The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view with 'baseveromelisa' selected, showing tables like 'auditoriaprestamos', 'bibliotecarios', 'libros', 'prestamos', and 'usuarios'. The main editor window shows a SQL query: `SELECT * FROM baseveromelisa.auditoriaprestamos;`. Below the query, the 'Result Grid' displays two rows of data:

id_auditoria	id_prestamo	accion	fecha
1	1	prestamo	2024-06-01
2	2	prestamo	2024-06-05

Procedimientos Almacenados (Stored Procedures)

Un stored procedure es un conjunto de sentencias SQL que se pueden ejecutar como una sola unidad en el servidor de bases de datos.

Para crear el procedimiento almacenado registrar_prestamo, hicimos clic derecho en Stored Procedures de baseveromelisa y luego hicimos clic en Create Store Procedure.



The screenshot shows the MySQL Workbench interface with the 'registrar_prestamo' stored procedure being created. The 'DDL' pane shows the following code:

```
1 CREATE DEFINER='root'@'localhost' PROCEDURE `registrar_prestamo`(  
2     IN p_id_libro INT,  
3     IN p_id_usuario INT,  
4     IN p_fecha_prestamo DATE,  
5     IN p_id_prestamo INT,  
6     IN p_id_auditoria INT  
7 )  
8 BEGIN  
9     DECLARE libro_prestado BOOLEAN;  
10    DECLARE estado_libro VARCHAR(20);  
11  
12    -- Verificar si el libro ya está prestado  
13    SELECT COUNT(*)  
14    INTO libro_prestado  
15    FROM Prestamos  
16    WHERE id_libro = p_id_libro AND fecha_devolucion IS NULL;  
17  
18    -- Obtener el estado del libro  
19    SELECT estado  
20    INTO estado_libro  
21    FROM Libros  
22    WHERE id_libro = p_id_libro;  
23  
24    IF libro_prestado > 0 OR estado_libro = 'prestado' THEN  
25        -- El libro ya está prestado, lanzar un error  
26        SIGNAL SQLSTATE '45000';
```

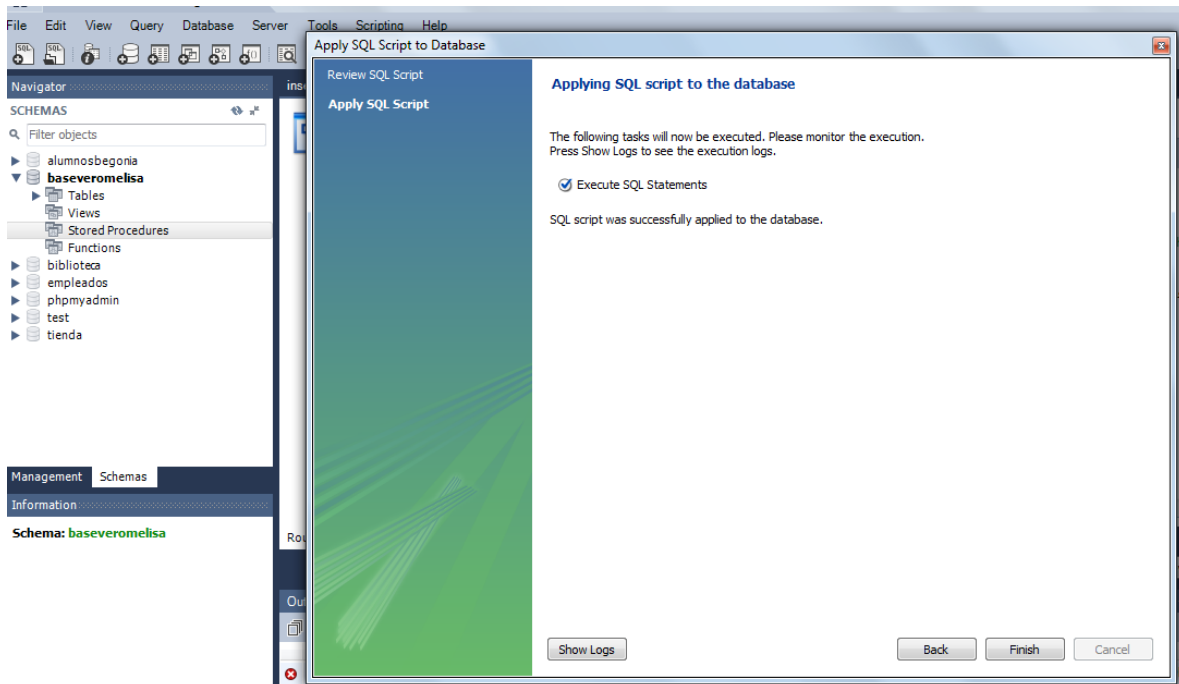
Escribimos el código del procedimiento e hicimos clic en Apply.

Declaración de la Variable estado_libro: Se declara una variable para almacenar el estado del libro.

Selección del Estado del Libro: Se selecciona explícitamente el estado del libro y se almacena en estado_libro.

Se verifica si libro_prestado es mayor que 0 o si estado_libro es 'prestado' para lanzar un error si el libro ya está prestado.

Revisamos el código e hicimos clic en aplicar.



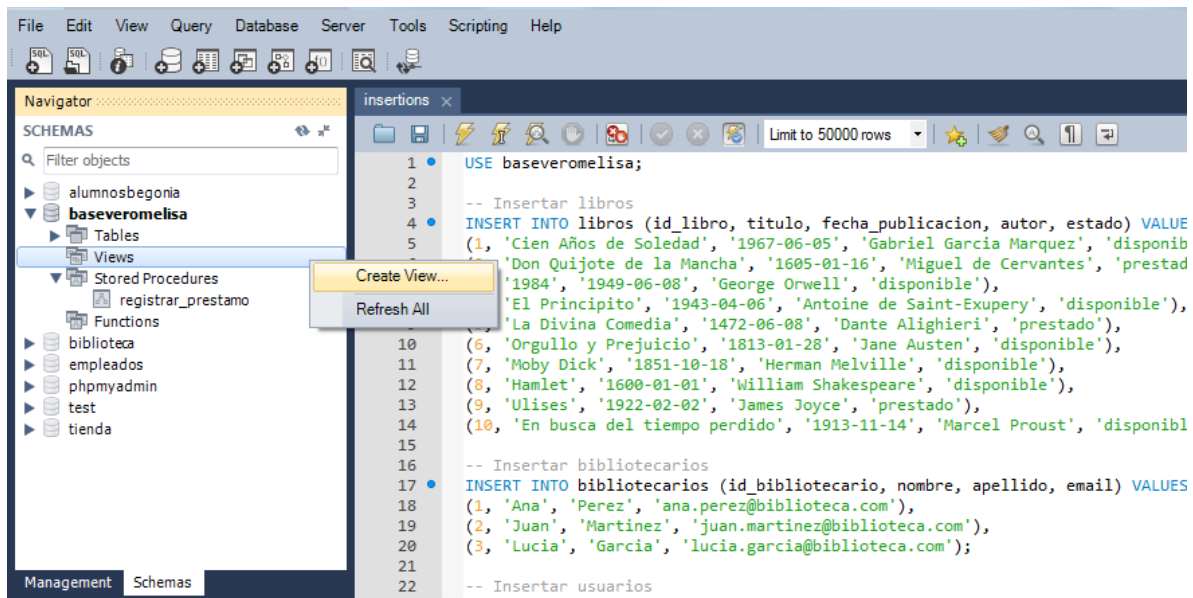
Hicimos clic en terminar, para tener nuestro procedimiento almacenado.

Para probar el procedimiento se usa la sentencia call (llamamos al procedimiento).

Vista

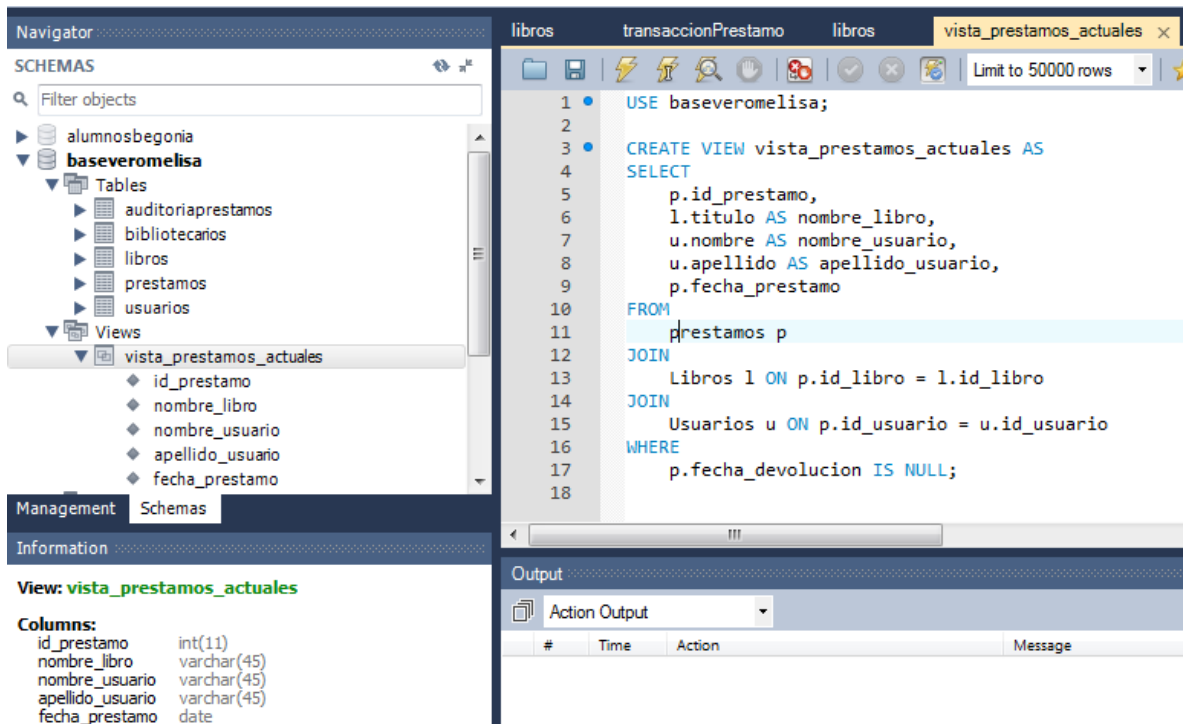
Una vista es una tabla virtual cuyo contenido está definido por una consulta.

Para crear la vista vista_prestamos_actuales hicimos clic en Views en baseveromelisa y en Create View:

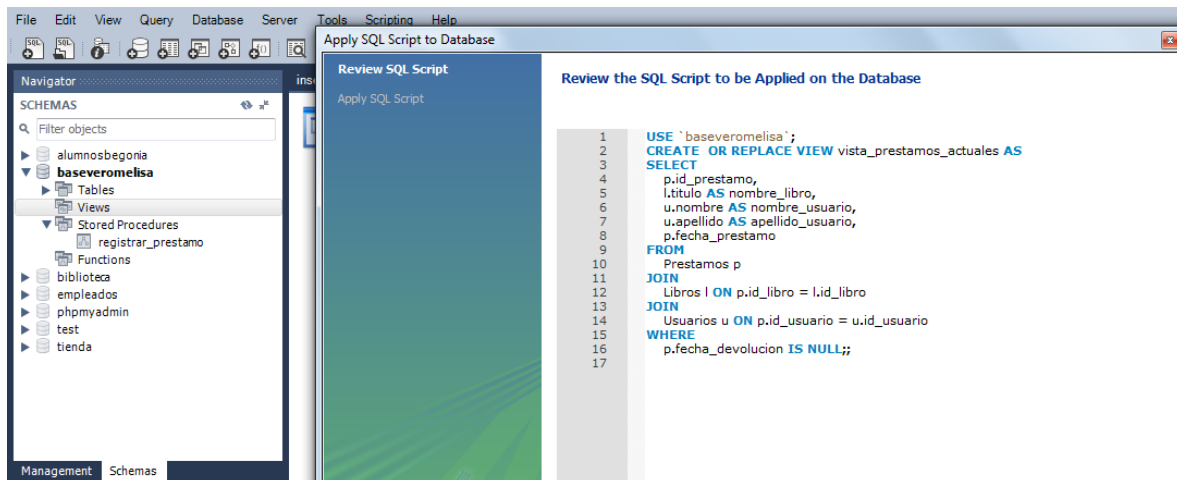


En nuestro caso la vista se crea para poder “ver” cuáles son los préstamos activos de nuestra biblioteca.

Se toman los datos de las tablas préstamos, libros y usuarios pero sólo se consideran aquellos préstamos en los que la fecha de devolución es nula. Esto quiere decir que el libro prestado aún no ha sido devuelto (un préstamo activo).



Revisamos y aplicamos el script en nuestra base de datos:



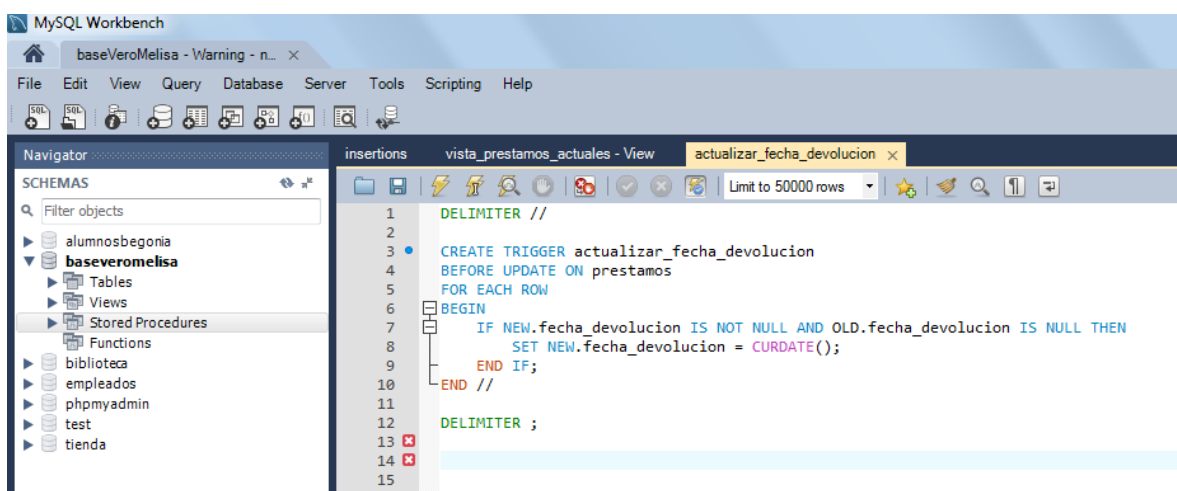
Quedó creada la vista vista_prestamos_actuales.

Disparador

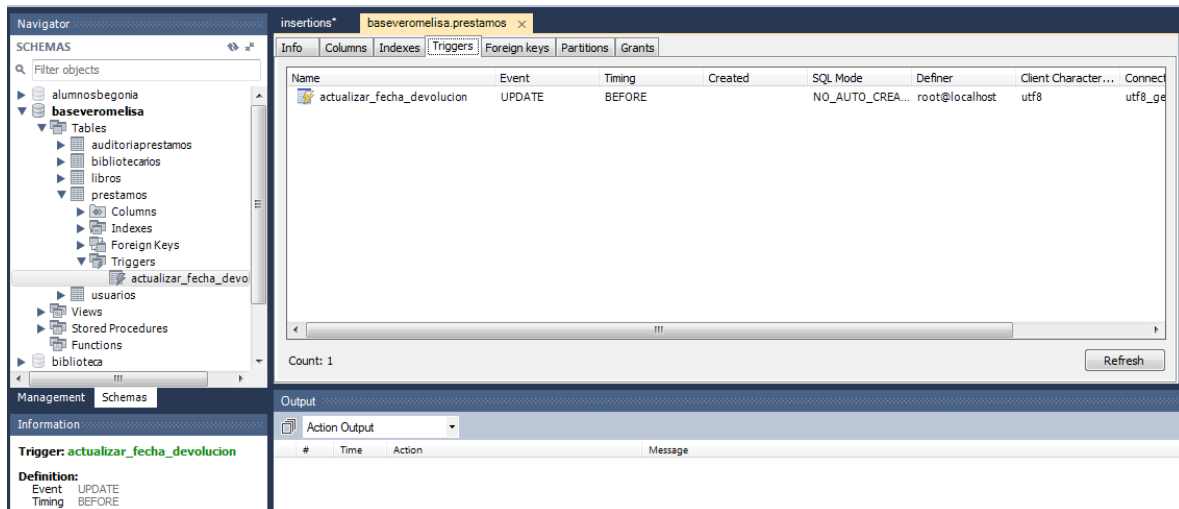
Un trigger es una funcionalidad que la base de datos ejecuta de forma automática cuando se realiza una operación de tipo insert, update o delete.

En nuestro caso el disparador actualizar_fecha_devolucion se ejecuta antes de una actualización en la tabla de prestamos.

Lo que hace esta función al dispararse ante una actualización de prestamos, es establecer la fecha de devolución del libro como la fecha actual detectada por el sistema.

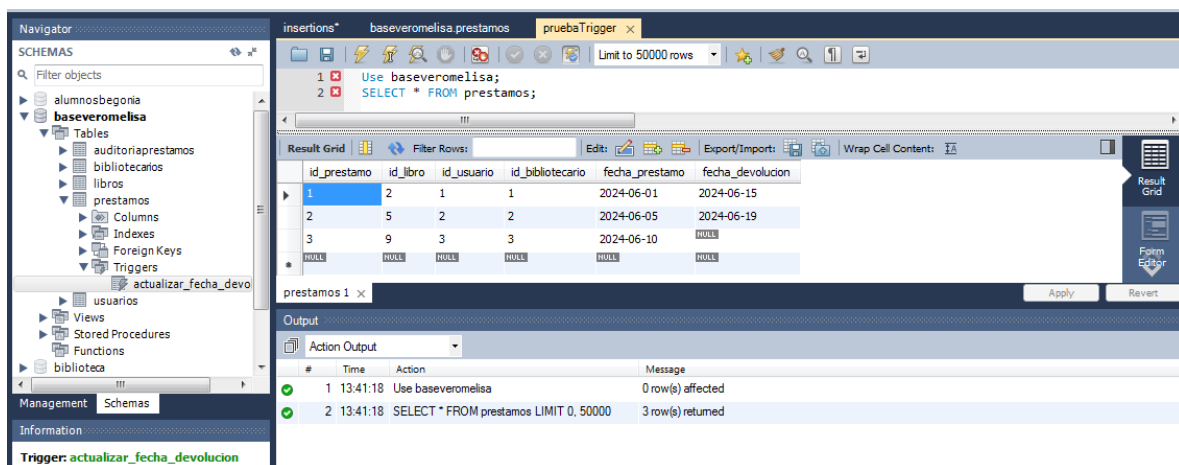


Luego de ejecutar el código, se puede observar que la tabla prestamos, tiene ahora el disparador actualizar_fecha_devolucion.



Para verificar que el trigger funciona como debe ser, realizamos una operación de actualización en la tabla prestamos y le damos una fecha de devolución (distinta a la fecha de hoy).

Vemos qué datos tiene la tabla prestamos antes de actualizarla:

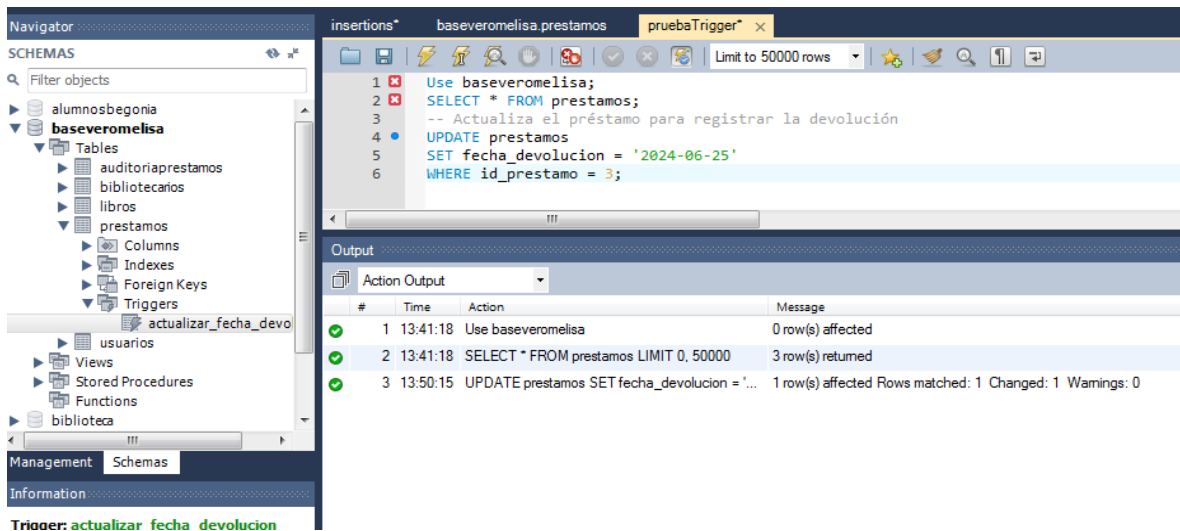


Al inicio hay tres préstamos realizados, dos con fechas de devolución. O sea que dos libros se prestaron y se devolvieron. Uno de los libros “Ulises” se prestó el día 10 y no se devolvió aún ya que el atributo fecha_devolucion contiene NULL.

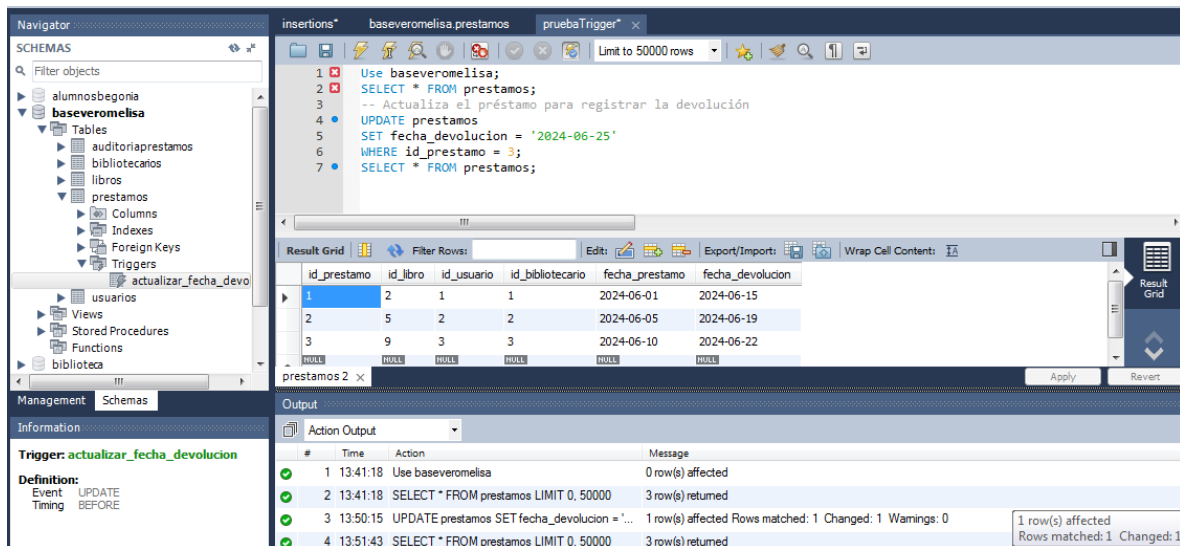
Aclaramos que la fecha en la que se realiza este párrafo de éste informe es 22 de junio de 2024.

Hacemos un script SQL para hacer como que el libro con id de préstamo igual a tres fue devuelto, pero establecemos como fecha de devolución el veinticinco de junio de 2024 (no la fecha de hoy).

Ejecutamos el script para actualizar los datos de la tabla préstamos.



Volvemos a escribir el script para ver el contenido total de la tabla prestamos:



Como se puede observar en la captura anterior, el trigger funcionó perfectamente ya que la fecha que se estableció para la devolución del libro fue la fecha actual.

Transacción

Para realizar una transacción que registre un nuevo préstamo, actualice el estado del libro a "prestado" y registre la operación en la tabla de auditoría, primero observamos qué datos contiene la tabla auditoriaprestamos.

The screenshot shows a database management interface with a 'Navigator' pane on the left displaying the 'baseveromelisa' schema. The main pane shows a query window with the SQL statement: `SELECT * FROM baseveromelisa.auditoriaprestamos;`. Below the query, a 'Result Grid' displays the data from the 'auditoriaprestamos' table.

id_auditoria	id_prestamo	accion	fecha
1	1	prestamo	2024-06-01
2	2	prestamo	2024-06-05
NULL	NULL	NULL	NULL

Tiene registradas dos auditorías, las cuales son ambas de préstamos realizados.

También observamos el contenido de la tabla préstamos:

The screenshot shows the same database management interface, but now the 'prestamos' table is selected. The query window shows: `SELECT * FROM baseveromelisa.prestamos;`. The 'Result Grid' displays the data from the 'prestamos' table.

id_prestamo	id_libro	id_usuario	id_bibliotecario	fecha_prestamo	fecha_devolucion
1	2	1	1	2024-06-01	2024-06-15
2	5	2	2	2024-06-05	2024-06-19
3	9	3	3	2024-06-10	2024-06-22
NULL	NULL	NULL	NULL	NULL	NULL

Hay tres préstamos registrados de tres libros diferentes que ya fueron devueltos.

Luego observamos el estado de la tabla libros.

baseveromelisa.prestamos auditoriaprestamos libros

1 • SELECT * FROM baseveromelisa.libros;

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

id_libro	titulo	fecha_publicacion	autor	estado
1	Cien Años de Soledad	1967-06-05	Gabriel Garcia Marquez	disponible
2	Don Quijote de la Mancha	1605-01-16	Miguel de Cervantes	prestado
3	1984	1949-06-08	George Orwell	disponible
4	El Principito	1943-04-06	Antoine de Saint-Exupery	disponible
5	La Divina Comedia	1472-06-08	Dante Alighieri	prestado
6	Orgullo y Prejuicio	1813-01-28	Jane Austen	disponible
7	Moby Dick	1851-10-18	Herman Melville	disponible
8	Hamlet	1600-01-01	William Shakespeare	disponible
9	Ulises	1922-02-02	James Joyce	prestado
10	En busca del tiempo perdido	1913-11-14	Marcel Proust	disponible

Trigger: actualizar_fecha_devolucion

Ya que el libro Hamlet figura como disponible, vamos a hacer como que es éste libro el que viene a retirar un usuario de nuestra biblioteca. El libro Hamlet tiene como id 8.

En un nuevo archivo.sql vamos a crear el script necesario para realizar esta transacción. El archivo de la transacción es: transaccionPrestamo.sql.

En este archivo se comienza una nueva transacción.

Establecemos las variables para los valores necesarios. El id del libro es el 8 ya que el usuario quiere leer Hamlet. El usuario es Laura Fernandez ya que es la usuaria con id 4. Fue recibida en nuestra biblioteca por la bibliotecaria Ana Perez con id igual a 1. La fecha del préstamo es el 22 de junio (nuestra biblioteca abre los sábados ☺)

Insertamos los datos correspondientes en la tabla de préstamos.

Hacemos la actualización de la tabla libros. El libro Hamlet debe tener como estado “prestado”.

Insertamos los datos necesarios en la tabla auditorioprestamos para registrar el nuevo préstamo tomando la fecha actual.

Confirmamos la transacción (los cambios son permanentes).


```

4
5
6 -- Variables
7 SET @id_libro = 8;
8 SET @id_usuario = 4;
9 SET @id_bibliotecario = 1;
10 SET @fecha_prestamo = '2024-06-22';
11
12 -- Insertar el nuevo préstamo
13 INSERT INTO prestamos (id_libro, id_usuario, id_bibliotecario, fecha_prestamo, fecha_devolucion)
14 VALUES (@id_libro, @id_usuario, @id_bibliotecario, @fecha_prestamo, NULL);
15
16 -- Obtener el id del préstamo recién insertado
17 SET @id_prestamo = LAST_INSERT_ID();
18
19 -- Actualizar el estado del libro a 'prestado'
20 UPDATE Libros
21 SET estado = 'prestado'
22 WHERE id_libro = @id_libro;
23
24 -- Registrar la operación en AuditoriaPrestamos
25 INSERT INTO AuditoriaPrestamos (id_prestamo, accion, fecha)
26 VALUES (@id_prestamo, 'prestamo', CURDATE());
27
28 COMMIT;

```

Vamos a ver ahora si todo salió como debía ser.

Luego de la transacción, el libro Hamlet (id 8) figura en la tabla libros como prestado 😊

id_libro	titulo	fecha_publicacion	autor	estado
1	Cien Años de Soledad	1967-06-05	Gabriel Garcia Marquez	disponible
2	Don Quijote de la Mancha	1605-01-16	Miguel de Cervantes	prestado
3	1984	1949-06-08	George Orwell	disponible
4	El Principito	1943-04-06	Antoine de Saint-Exupery	disponible
5	La Divina Comedia	1472-06-08	Dante Alighieri	prestado
6	Orgullo y Prejuicio	1813-01-28	Jane Austen	disponible
7	Moby Dick	1851-10-18	Herman Melville	disponible
8	Hamlet	1600-01-01	William Shakespeare	prestado
9	Ulises	1922-02-02	James Joyce	prestado
10	En busca del tiempo perdido	1913-11-14	Marcel Proust	disponible
NULL	NULL	NULL	NULL	NULL

Revisamos la tabla prestamos para asegurarnos de que se haya registrado un nuevo préstamo:

baseveromelisa.prestamos auditoriaprestamos libros transaccionPrestamo libros **prestamos**

1 • `SELECT * FROM baseveromelisa.prestamos;`

id_prestamo	id_libro	id_usuario	id_bibliotecario	fecha_prestamo	fecha_devolucion
0	8	4	1	2024-06-22	NULL
1	2	1	1	2024-06-01	2024-06-15
2	5	2	2	2024-06-05	2024-06-19
3	9	3	3	2024-06-10	2024-06-22
NULL	NULL	NULL	NULL	NULL	NULL

Se agregó a la tabla prestamos, el préstamo con id 0 del libro 8 que es Hamlet 😊

No tiene fecha de devolución porque el usuario recién lo llevó y todavía no lo leyó.

Chequeamos la tabla auditoriaprestamos:

libros transaccionPrestamo libros prestamos **auditoriaprestamos**

1 • `SELECT * FROM baseveromelisa.auditoriaprestamos;`

id_auditoria	id_prestamo	accion	fecha
0	0	prestamo	2024-06-22
1	1	prestamo	2024-06-01
2	2	prestamo	2024-06-05
NULL	NULL	NULL	NULL

Contiene un nuevo registro con el nuevo préstamo 😊

Finalmente pudimos comprobar que la transacción funciona correctamente.

Bibliografía

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). *Fundamentos de bases de datos* (6ª ed.). McGraw-Hill.

Elmasri, R., & Navathe, S. B. (2011). *Sistemas de bases de datos* (6ª ed.). Pearson Educación.

Date, C. J. (2007). *Introducción a los sistemas de bases de datos* (8ª ed.). Pearson Educación.

dbdiagram.io. (n.d.). Retrieved June 22, 2024, from <https://dbdiagram.io/home>

Google Classroom. (n.d.). Retrieved June 22, 2024, from <https://classroom.google.com/c/NjczMDgwOTUxNjE0>