1.1. Creación de tabla libros

```
-- Crear tabla libros

CREATE TABLE libros (

id SERIAL PRIMARY KEY,

titulo VARCHAR(100),

autor VARCHAR(100),

stock INTEGER

);
```

1.2. Creación de tabla usuarios

```
9 -- Crear tabla usuarios
10 - CREATE TABLE usuarios
11
        id SERIAL PRIMARY KEY,
         nombre VARCHAR(100).
12
13
14 );
       email VARCHAR(100)
15
15
16 -- Crear tabla prestamos con claves foráneas
17 v CREATE TABLE prestamos (
        id SERIAL PRIMARY KEY,
18
         id_libro INT REFERENCES libros(id),
19
         id_usuario INT REFERENCES usuarios(id),
         fecha DATE
22 );
```

Data Output Messages Notifications

CREATE TABLE

Total rows: Query complete 00:00:00.0038

Activar Windows

Query returned successfully in 38 msec. ×

CRLF Ln 9, Col 1

CRLF Ln 16, Col 1

1.3. Creación de tabla prestamos

2.1. Inserción de los datos libros

Total rows: Query complete 00:00:00.046

```
26 -- Insertar libros
27 • INSERT INTO libros (titulo, autor, stock) VALUES
     ('Cien años de soledad', 'Gabriel García Márquez', 5),
('1984', 'George Orwell', 4),
('Don Quijote', 'Miguel de Cervantes', 3),
28
29
30
      ('Rayuela', 'Julio Cortázar', 2),
32
     ('El Principito', 'Antoine de Saint-Exupéry', 6);
33
34 -- Insertar usuarios
35 - INSERT INTO usuarios (nombre, email) VALUES
36 ('Juan Pérez', 'juan@example.com'),
37 ('María López', 'maria@example.com'),
38 ('Carlos Ruiz', 'carlos@example.com');
39
40 -- Insertar préstamos
41 - INSERT INTO prestamos (id_libro, id_usuario, fecha) VALUES
42 (1, 1, '2025-07-15'),
43 (2, 2, '2025-07-16'),
     (3, 3, '2025-07-17');
Data Output Messages Notifications
                                                                                                                  tivar Windows
 INSERT 0 5

✓ Query returned successfully in 43 msec. 

Total rawa: Ouani complete 00:00:00 042
```

2.2. inserción de los datos usuarios

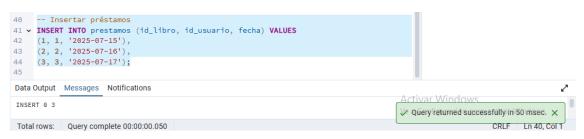
```
34 -- Insertar usuarios
35 V INSERT INTO usuarios (nombre, email) VALUES
36 ('Juan Pérez', 'juan@example.com'),
37 ('María López', 'maria@example.com')
38 ('Carlos Ruiz', 'carlos@example.com');
39
       - Insertar préstamos
40
41 - INSERT INTO prestamos (id_libro, id_usuario, fecha) VALUES
42 (1, 1, '2025-07-15'),
43 (2, 2, '2025-07-16'),
    (3, 3, '2025-07-17');
44
45
Data Output Messages Notifications
INSERT 0 3

✓ Query returned successfully in 42 msec. 

×

Total rows: Query complete 00:00:00.042
                                                                                                                            CRLF Ln 34, Col 1
```

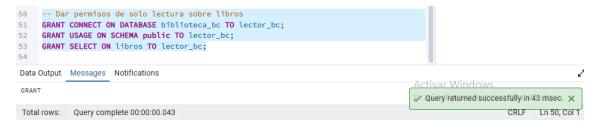
2.3. inserción de los datos prestamos



3.1. creación de roles



3.2. asignación de los permisos al rol



4.1. inicio de transacción con begin

```
--4. inicio de transaccion con begin
    -- Iniciar transacción
56
    BEGIN;
57
58
59 -- Insertar un préstamo válido
60 v INSERT INTO prestamos (id_libro, id_usuario, fecha) VALUES
61
    (1, 2, '2025-07-18');
62
63
    -- Insertar un préstamo con error (libro id 999 no existe)
64 • INSERT INTO prestamos (id_libro, id_usuario, fecha) VALUES
65 (999, 2, '2025-07-18');
66
    -- Como hay error, hacemos rollback
67
68 ROLLBACK;
69
70
    -- Volvemos a iniciar y corregimos
71 BEGIN;
    -- Insertamos uno correcto
74 v INSERT INTO prestamos (id_libro, id_usuario, fecha) VALUES
Data Output Messages Notifications
BEGIN

✓ Query returned successfully in 37 msec. 
X

Total rows: Query complete 00:00:00.037
                                                                                                           CRLF Ln 56, Col 1
```

4.2. inserción de un préstamo valido

```
59 -- Insertar un préstamo válido
60 - INSERT INTO prestamos (id_libro, id_usuario, fecha) VALUES
61 (1, 2, '2025-07-18');
     -- Insertar un préstamo con error (libro id 999 no existe)
64 - INSERT INTO prestamos (id_libro, id_usuario, fecha) VALUES
65 (999, 2, '2025-07-18');
66
    -- Como hay error, hacemos rollback
68 ROLLBACK;
69
     -- Volvemos a iniciar y corregimos
70
71 BEGIN;
    -- Insertamos uno correcto
74 v INSERT INTO prestamos (id_libro, id_usuario, fecha) VALUES
Data Output Messages Notifications
INSERT 0 1

✓ Query returned successfully in 44 msec. 

Total rows: Query complete 00:00:00.044
                                                                                                                  Ln 59, Col 1
```

4.3. inserción de préstamo con error intencional

```
-- Insertar un préstamo con error (libro id 999 no existe)

1NSERT INTO prestamos (id_libro, id_usuario, fecha) VALUES

(999, 2, '2025-07-18');
```

Messages

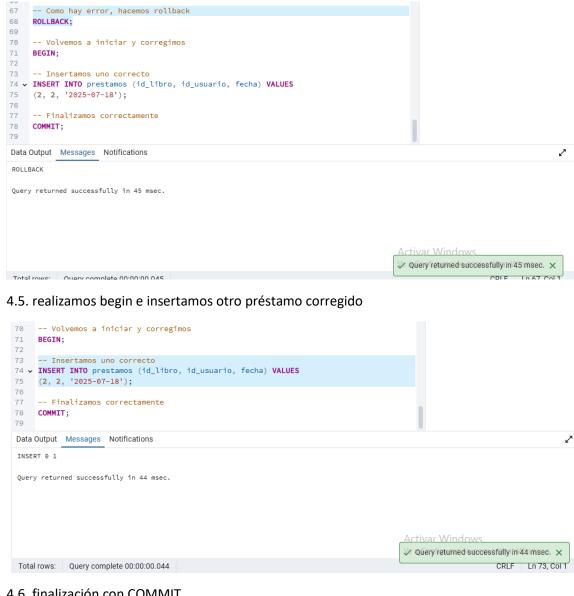
ERROR: inserción o actualización en la tabla «prestamos» viola la llave foránea «prestamos_id_libro_fkey»

La llave (id_libro)=(999) no está presente en la tabla «libros».

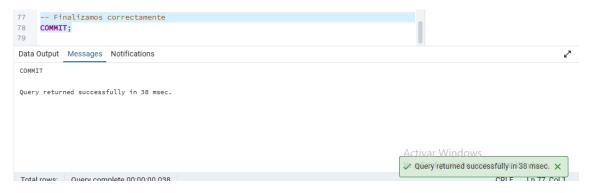
SQL state: 23503

Detail: La llave (id_libro)=(999) no está presente en la tabla «libros».

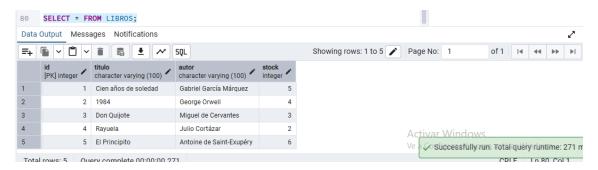
4.4. uso del ROLLBACK



4.6. finalización con COMMIT



CONCLUSION:



1. ¿Qué aprendiste al aplicar todos los sublenguajes SQL en un solo ejercicio?

Aprendí cómo se integran los sublenguajes DDL (definición de tablas), DML (inserciones), DCL (permisos) y TCL (transacciones) en una solución real, mostrando cómo funcionan juntos para garantizar consistencia y seguridad en una base de datos.

2. ¿Qué errores lograste evitar usando ROLLBACK?

Evité que se guardara un préstamo con un id_libro inválido (999), lo que habría generado inconsistencia. Con ROLLBACK, todo lo que se intentó guardar después del error fue descartado.

3. ¿Para qué sirve crear usuarios con permisos limitados?

Sirve para proteger la base de datos de accesos no autorizados o acciones no deseadas, como borrar o modificar información crítica. El rol lector solo puede consultar datos, ideal para usuarios que solo necesitan visualizar.