

Query 1

Limit to 1000 rows

```
70 FOREIGN KEY (ID_EJEMPLAR) REFERENCES BIBLIOTECA.EJEMPLAR(CODIGO)
71 );
72
73 -- Consultas
74 • SELECT * FROM BIBLIOTECA.AUTOR;
75 • SELECT * FROM BIBLIOTECA.LIBRO;
76 • SELECT * FROM BIBLIOTECA.EJEMPLAR;
77 • SELECT * FROM BIBLIOTECA.LOCALIZACION;
78 • SELECT * FROM BIBLIOTECA.USUARIO;
```

Result Grid

	CODIGO_LIBRO	TITULO	ISBN	PAGINAS	EDITORIAL
▶	1	Cien años de soledad	9780060883287	417	HarperCollins
	2	La casa de los espíritus	9780553383805	481	Plaza & Janes
	3	Harry Potter y la piedra filosofal	9780747532743	309	Bloomsbury
	4	1984	9780451524935	328	Penguin
	5	Don Quijote	9780060934347	992	Vintage

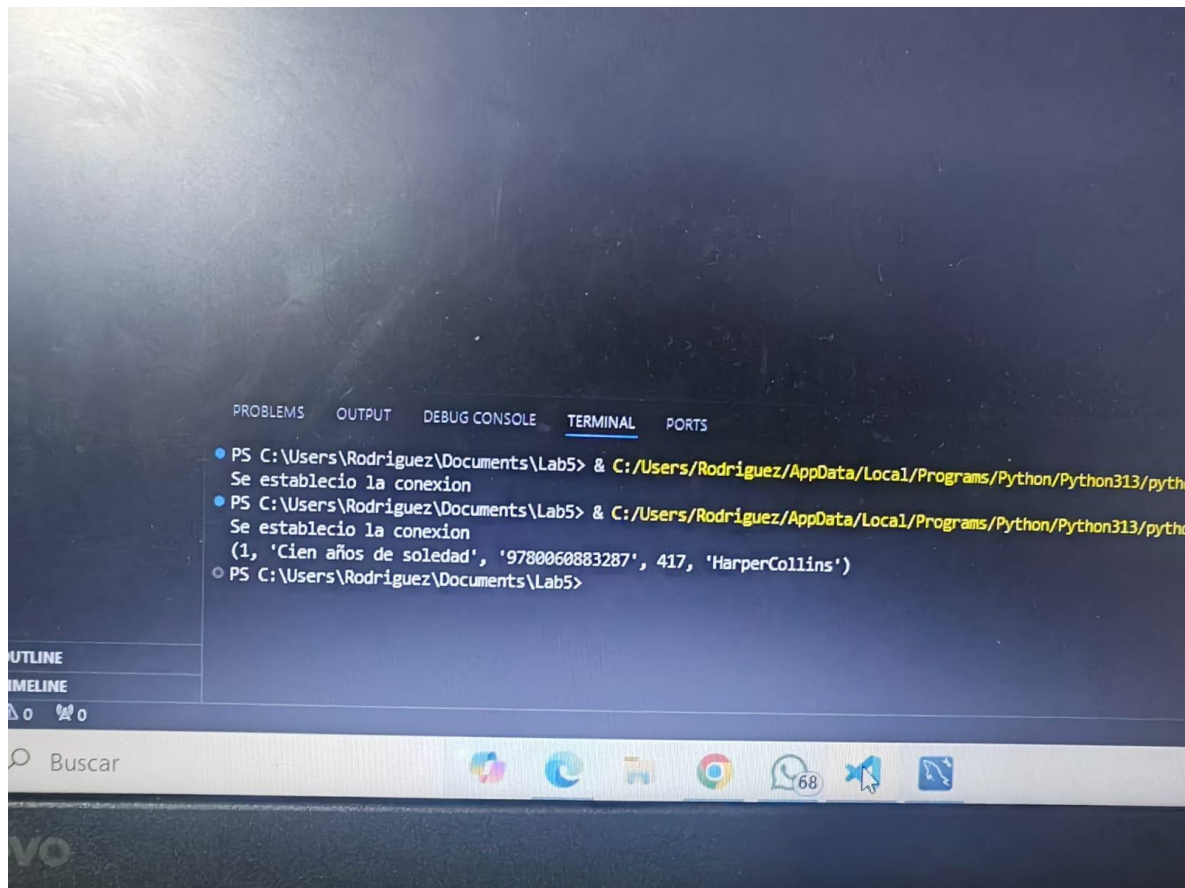
LIBRO 1 x

Output

Action Output

#	Time	Action	Message
✓ 28	00:17:07	INSERT INTO BIBLIOTECA.AUTOR_LIBRO (CODIGO_AUTOR, CODIGO_LIBRO) VALUE...	10 row(s) affected Records: 10 Duplica
✓ 29	00:17:17	INSERT INTO BIBLIOTECA.LOCALIZACION (ID_LOCALIZACION, RECINTO) VALUES (1, '...	10 row(s) affected Records: 10 Duplica
✓ 30	00:17:37	INSERT INTO BIBLIOTECA.USUARIO (CODIGO_NOMBRE, TELEFONO, DIRECCION, LO...	10 row(s) affected Records: 10 Duplica

Aquí podemos evidenciar que la tabla LIBRO ya esta creada con distintos registros.



En esta podemos evidenciar la conexión a la BD en la aplicación de Python y de paso podemos observar que se ejecutó una consulta LIBRO con éxito.

Query 1

```
70 FOREIGN KEY (ID_EJEMPLAR) REFERENCES BIBLIOTECA.EJEMPLAR(CODIGO)
71 );
72
73 -- Consultas
74 • SELECT * FROM BIBLIOTECA.AUTOR;
75 • SELECT * FROM BIBLIOTECA.LIBRO;
76 • SELECT * FROM BIBLIOTECA.EJEMPLAR;
77 • SELECT * FROM BIBLIOTECA.LOCALIZACION;
78 • SELECT * FROM BIBLIOTECA.USUARIO;
```

Result Grid

	CODIGO	NOMBRE
▶	1	Vernon
	2	Isabel Allende
	3	J.K. Rowling
	4	George Orwell
	5	Miguel de Cervantes

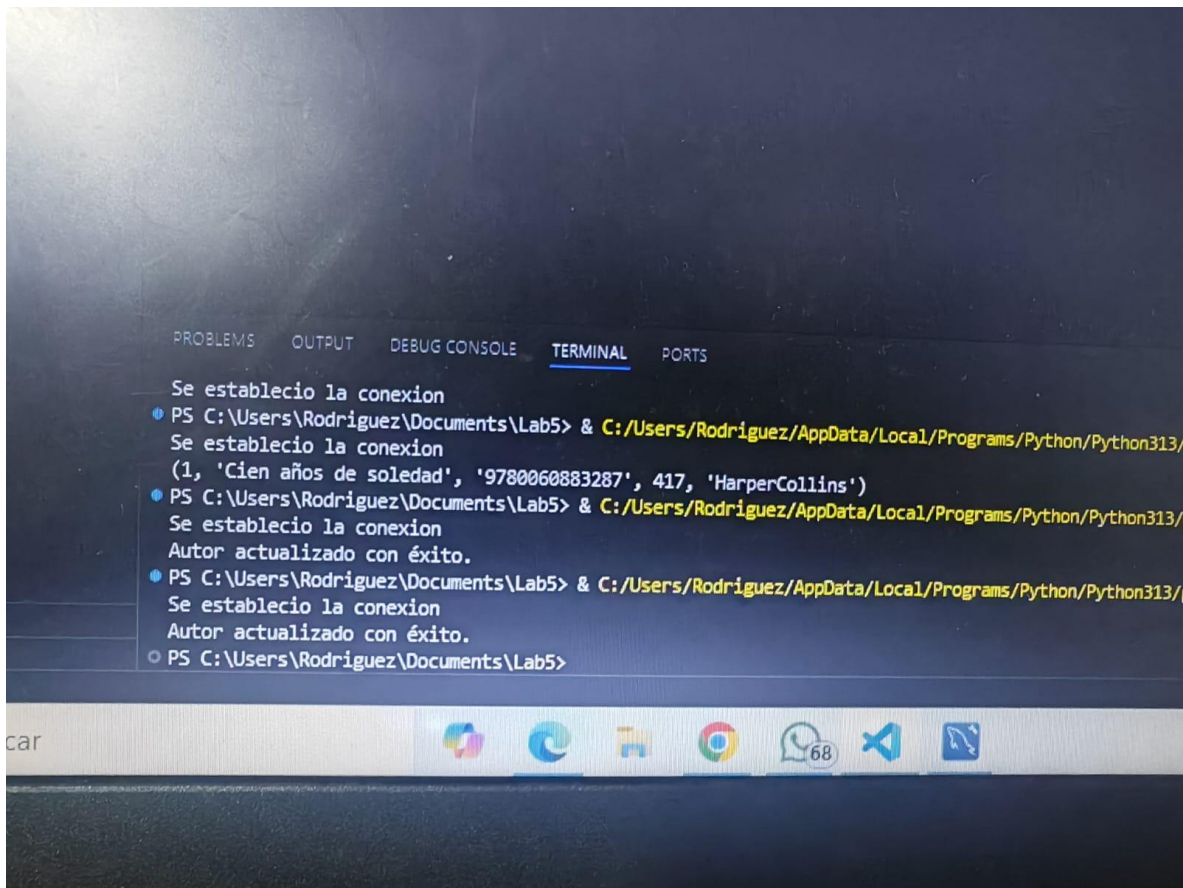
AUTOR 2 x

Output

Action Output

#	Time	Action	Message
✓ 29	00:17:17	INSERT INTO BIBLIOTECA.LOCALIZACION (ID_LOCALIZACION, RECINTO) VALUES (1, '...'	10 row(s) af
✓ 30	00:17:37	INSERT INTO BIBLIOTECA.USUARIO (CODIGO, NOMBRE, TELEFONO, DIRECCION, LO...	10 row(s) af

Aquí tenemos en evidencia la tabla Autor y podemos observar los registros insertados.



En la imagen anterior podemos observar con éxito qué se actualizaron los datos de autor.


```
71 );
72
73 -- Consultas
74 • SELECT * FROM BIBLIOTECA.AUTOR;
75 • SELECT * FROM BIBLIOTECA.LIBRO;
76 • SELECT * FROM BIBLIOTECA.EJEMPLAR;
77 • SELECT * FROM BIBLIOTECA.LOCALIZACION;
78 • SELECT * FROM BIBLIOTECA.USUARIO;
```

Result Grid

	CODIGO	NOMBRE
▶	1	Barbie
	2	Isabel Allende
	3	J.K. Rowling
	4	George Orwell
	5	Miguel de Cervantes

AUTOR 2 x

Output

Action Output

#	Time	Action
✓ 29	00:17:17	INSERT INTO BIBLIOTECA.LOCALIZACION (ID_LOCALIZACION, RECINTO) VALUES (1, '... 10
✓ 30	00:17:37	INSERT INTO BIBLIOTECA.USUARIO (CODIGO, NOMBRE, TELEFONO, DIRECCION, LO... 10

Aquí el autor con ID 1 actualizado.

Query 1

Limit to 1000 rows

```
71 );
72
73 -- Consultas
74 • SELECT * FROM BIBLIOTECA.AUTOR;
75 • SELECT * FROM BIBLIOTECA.LIBRO;
76 • SELECT * FROM BIBLIOTECA.EJEMPLAR;
77 • SELECT * FROM BIBLIOTECA.LOCALIZACION;
78 • SELECT * FROM BIBLIOTECA.USUARIO;
79 • SELECT * FROM BIBLIOTECA.USUARIO_EJEMPLAR;
```

Result Grid

	CODIGO	NOMBRE	TELEFONO	DIRECCION	LOCALIZACION
▶ 1		Juan Pérez	123456789	Calle 1	1
2		María López	987654321	Calle 2	2
3		Pedro Ramírez	456789123	Calle 3	3
4		Ana Gómez	789123456	Calle 4	4
5		Luis Torres	321654987	Calle 5	5

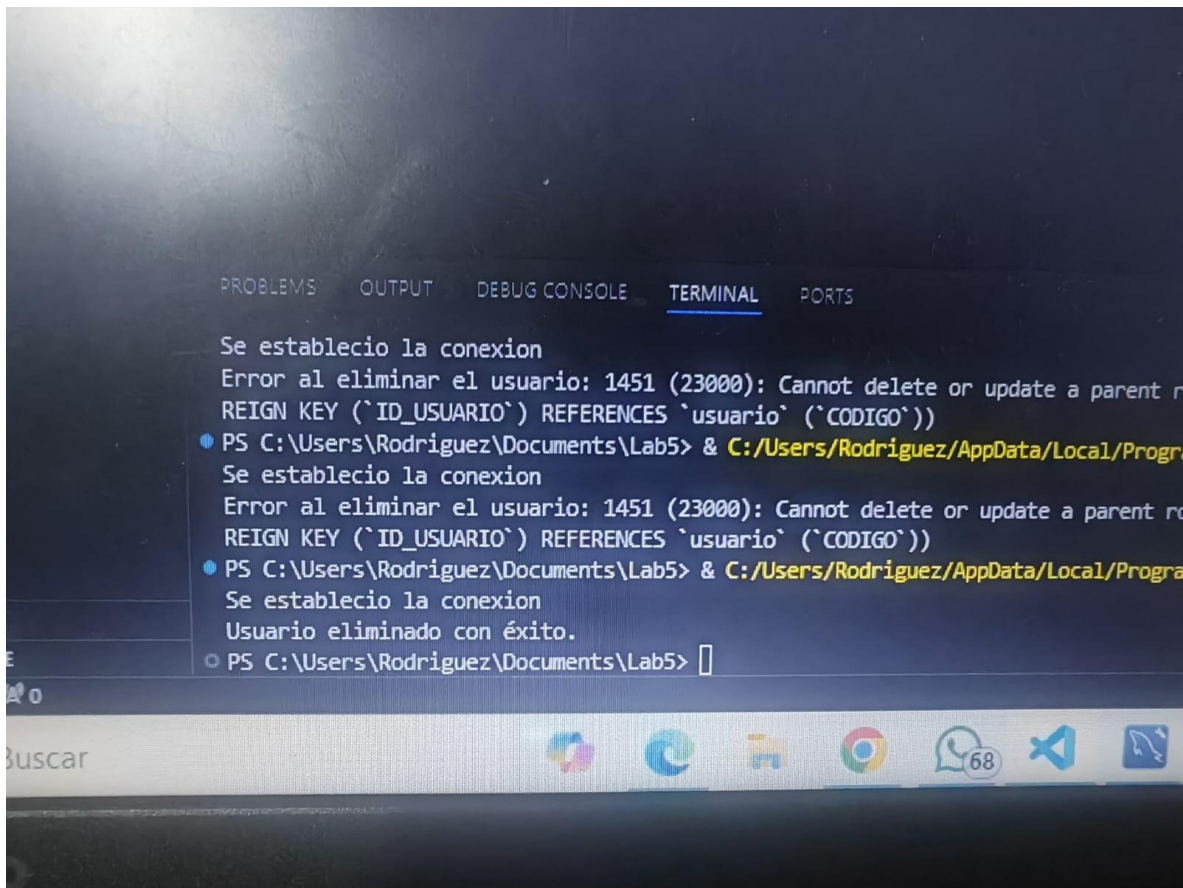
USUARIO 3

Output

Action Output

#	Time	Action	Message
✓ 30	00:17:37	INSERT INTO BIBLIOTECA.USUARIO (CODIGO, NOMBRE, TELEFONO, DIRECCION, LO...	10 row(s) affected Rec
✓ 31	00:17:52	INSERT INTO BIBLIOTECA.EJEMPLAR (CODIGO, LOCALIZACION, CODIGO_LIBRO) VAL...	10 row(s) affected Rec
✓ 32	00:18:11	INSERT INTO BIBLIOTECA.USUARIO_EJEMPLAR (ID_USUARIO, ID_EJEMPLAR, PK_U...	10 row(s) affected Rec
✓ 33	00:24:05	SELECT * FROM BIBLIOTECA.LIBRO LIMIT 0, 1000	10 row(s) returned

Acá podemos ver la tabla Usuario creada con sus registros.



Acá podemos ver que se ejecutó con éxito el proceso de eliminar Usuario.


```

76 • SELECT * FROM BIBLIOTECA.USUARIO;
77 • SELECT * FROM BIBLIOTECA.USUARIO_EJEMPLAR;
78
79 -- Consultas de joins
80 • SELECT al.CODIGO_AUTOR, al.CODIGO_LIBRO
81 FROM BIBLIOTECA.AUTOR_LIBRO al;
82

```

Result Grid

	CODIGO	NOMBRE	TELEFONO	DIRECCION	LOCALIZACION
PK	2	María López	987654321	Calle 2	2
rchar(30)	3	Pedro Ramírez	456789123	Calle 3	3
rchar(30)	4	Ana Gómez	789123456	Calle 4	4
rchar(30)	5	Luis Torres	321654987	Calle 5	5
	6	Sofía Rodríguez	654987321	Calle 6	6

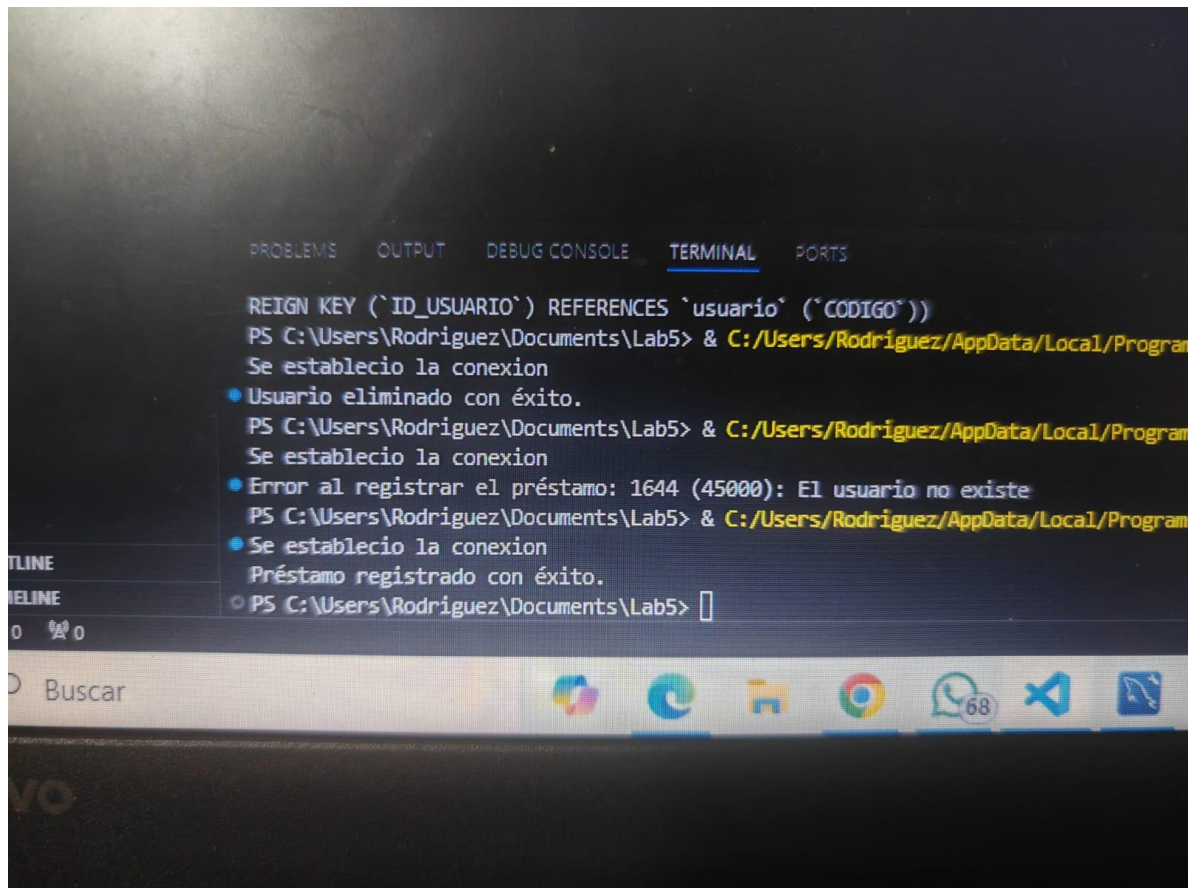
USUARIO 5 ✕

Output

Action Output

#	Time	Action	Message
40	01:38:46	REATE PROCEDURE RegistrarPrestamo(IN id_usuario INT, IN id_ejemplar INT) BEGIN	Error
41	01:39:42	CREATE PROCEDURE RegistrarPrestamo(IN id_usuario INT, IN id_ejemplar INT) BEGIN	0 row
42	01:43:14	DROP TABLE 'biblioteca'. 'usuario_ejemplar'	0 row

Usuario #1 eliminado.



The image shows a screenshot of a Visual Studio Code terminal window. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. The output shows a SQL command: `REIGN KEY (`ID_USUARIO`) REFERENCES `usuario` (`CODIGO`))`. Below this, a PowerShell command is executed: `PS C:\Users\Rodriguez\Documents\Lab5> & C:/Users/Rodriguez/AppData/Local/Program`. The output continues with `Se establecio la conexion`. A blue bullet point indicates `• Usuario eliminado con éxito.`. Another PowerShell command is shown: `PS C:\Users\Rodriguez\Documents\Lab5> & C:/Users/Rodriguez/AppData/Local/Program`, followed by `Se establecio la conexion`. A blue bullet point indicates `• Error al registrar el préstamo: 1644 (45000): El usuario no existe`. Another PowerShell command is shown: `PS C:\Users\Rodriguez\Documents\Lab5> & C:/Users/Rodriguez/AppData/Local/Program`, followed by `Se establecio la conexion` and `Préstamo registrado con éxito.`. A final PowerShell command is shown: `PS C:\Users\Rodriguez\Documents\Lab5>` . The bottom of the image shows a Windows taskbar with a search bar labeled 'Buscar' and several application icons including File Explorer, Edge, and Chrome.

```
REIGN KEY (`ID_USUARIO`) REFERENCES `usuario` (`CODIGO`))
PS C:\Users\Rodriguez\Documents\Lab5> & C:/Users/Rodriguez/AppData/Local/Program
Se establecio la conexion
• Usuario eliminado con éxito.
PS C:\Users\Rodriguez\Documents\Lab5> & C:/Users/Rodriguez/AppData/Local/Program
Se establecio la conexion
• Error al registrar el préstamo: 1644 (45000): El usuario no existe
PS C:\Users\Rodriguez\Documents\Lab5> & C:/Users/Rodriguez/AppData/Local/Program
Se establecio la conexion
Préstamo registrado con éxito.
PS C:\Users\Rodriguez\Documents\Lab5> 
```

Acá tenemos evidenciado la ejecución de asignación de prestamos con éxito.

```
72 • SELECT * FROM BIBLIOTECA.AUTOR;
73 • SELECT * FROM BIBLIOTECA.LIBRO;
74 • SELECT * FROM BIBLIOTECA.EJEMPLAR;
75 • SELECT * FROM BIBLIOTECA.LOCALIZACION;
76 • SELECT * FROM BIBLIOTECA.USUARIO;
77 • SELECT * FROM BIBLIOTECA.USUARIO_EJEMPLAR;
78 • SELECT * FROM BIBLIOTECA.PRESTAMO;
79
```

Result Grid | Filter Rows: | Edit: | Export/Import: |

	ID_PRESTAMO	CODIGO_USUARIO	CODIGO_EJEMPLAR	FECHA_PRESTAMO
r(30) ▶	2	3	1	2024-11-18 02:05:51
r(30) *	NULL	NULL	NULL	NULL
r(30)				

PRESTAMO 8 x

Output

Action Output

#	Time	Action
✓ 44	01:46:32	SELECT * FROM BIBLIOTECA.EJEMPLAR LIMIT 0, 1000

Esta la tabla de prestamos con su primer registro ingresado desde la aplicación.