



TECNOLÓGICO NACIONAL DE MÉXICO  
INSTITUTO TECNOLÓGICO DE TLAXIACO

---

**PRACTICA 3.**

---

SEGURIDAD Y VIRTUALIZACION

**CARRERA:**

INGENIERIA EN SISTEMAS COMPUTACIONALES

**GRUPO:** 7US

**PRESENTA:**

ANA MICHEL LEÓN LEÓN - 21620112

ROSA SALAZAR DOROTEO - 18620216

FERNANDA RUIZ HERAS - 21520151

**DOCENTE**

ING. EDWARD OSORIO SALINA

Tlaxiaco, Oax., 19 de septiembre del 2024.



***“Educación, ciencia y tecnología, progreso día con día” ®***

## Contenido

Introducción .....	3
1. Crea una base de datos en MySQL con una BD que contenga los siguientes campos en tres tablas diferentes .....	3
2. Crea un script en Python que permita insertar los datos del archivo `customers-2000000.csv` .....	4
3. Crea tres usuarios en MySQL .....	7
4. Crea un script en Python que permita realizar una inyección de sql en la tabla 'users' y que muestre los datos de la tabla 'users' en la consola .....	12
5. Crea un backup de la base de datos 'secure_db' y restaura la base de datos en un servidor diferente .....	13
Conclusión .....	17

## Lista de ilustraciones

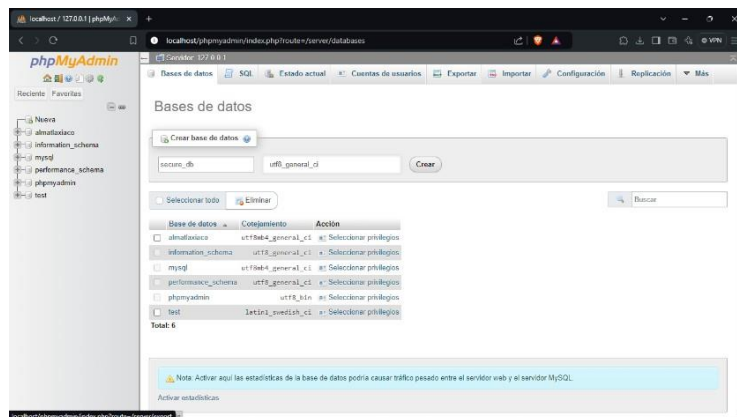
Ilustración 1 Crea base de datos secure_db .....	3
Ilustración 2 Se crea base sql .....	4
Ilustración 3 Consulta de datos .....	4
Ilustración 4 script insert_customers.py .....	5
Ilustración 5 Instalación de librerías .....	5
Ilustración 6 Librerías instaladas .....	5
Ilustración 7 Instalación de librería pandas .....	6
Ilustración 8 Script de conexión a base de datos .....	6
Ilustración 9 Insertar datos del archivo customers a la base secure_db .....	6
Ilustración 10 Ejecuta el script para insertar datos .....	7
Ilustración 11 Datos insertados correctamente a la base .....	7
Ilustración 12 Datos insertados en mysql .....	7
Ilustración 13 Se crea la tabla customers .....	8
Ilustración 14 Permisos al usuario .....	8
Ilustración 15 Consulta SHOW GRANTS .....	9
Ilustración 16 Se crea tabla address para usuario .....	9
Ilustración 17 Se ejecuta GRANT SELECT .....	10
Ilustración 18 Se ejecuta SELECT user .....	10
Ilustración 19 Asignar permisos al usuario 2 .....	11
Ilustración 20 Grant select .....	11
Ilustración 21 Consulta SHOW GRANTS .....	12
Ilustración 22 Insertar datos a la tabla users .....	12
Ilustración 23 Ejecutar script y resultado .....	13
Ilustración 24 Visualizar la tabla users .....	13
Ilustración 25 xampp .....	13
Ilustración 26 Peso de la base de datos .....	14
Ilustración 27 Exporta base de datos .....	14
Ilustración 28 Programa SQL .....	14
Ilustración 29 Selección de base de datos secure_db .....	15
Ilustración 30 dividir la base de datos .....	15
Ilustración 31 Dirección de la base de datos .....	16
Ilustración 32 Visualizar el resultado de dividir la base de datos .....	16
Ilustración 33 Visualiza en mysql .....	17

## Practica 3 - Bases de datos seguras

### INTRODUCCIÓN

Este proyecto tiene como objetivo la creación y administración de una base de datos MySQL llamada `secure_db`, diseñada para gestionar la información de usuarios, direcciones y clientes, distribuidos en tres tablas interrelacionadas: `users`, `address` y `customers`. A lo largo del desarrollo, se implementarán distintos procesos clave, como la inserción masiva de datos desde un archivo CSV a la tabla `customers`, mediante un script en Python. Además, se establecerán tres usuarios con permisos específicos en las tablas, garantizando la seguridad y control de acceso a los datos. Uno de los aspectos más críticos del proyecto será la demostración de una vulnerabilidad de inyección SQL en la tabla `users`, lo que permitirá evidenciar la importancia de aplicar buenas prácticas de seguridad en las aplicaciones que interactúan con bases de datos..

1. Crea una base de datos en MySQL con una BD que contenga los siguientes campos en tres tablas diferentes:
  - Primero para comenzar a crear la base de datos entramos a xampp le damos en el boto star para iniciar el servidor MySQL y hacer clic en el botón admin junto a Mysql para abrir phpMyAdmin. A continuación, le damos clic en la pestaña base de datos, en el campo de crear base de datos escribimos `secure_db` lo que es el nombre de la base de datos y le damos en el botón de crear.



*Ilustración 1 Crea base de datos `secure_db`*

- Posteriormente hacemos clic en la pestaña de `sql` y agregamos el código que vamos a ejecutar para crear la base de datos.

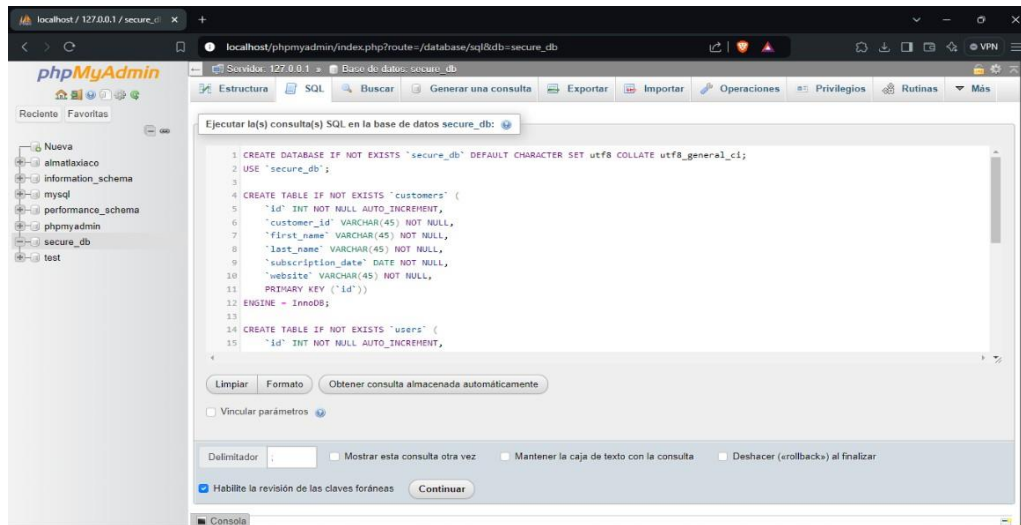


Ilustración 2 Se crea base sql

- En la siguiente captura ya nos muestra que se ha creado correctamente la base de datos con las tablas indicadas en el cual contiene la información del cliente como nombre, apellido, fecha de suscripción y sitio web.

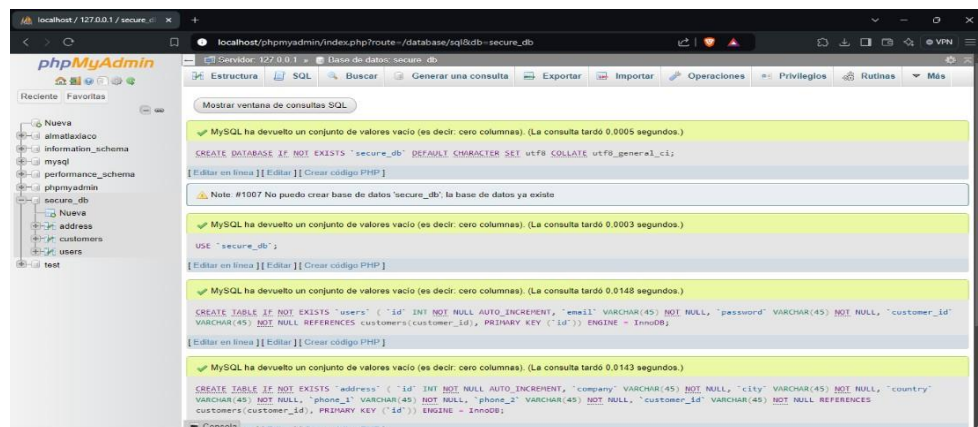


Ilustración 3 Consulta de datos

2. Crea un script en Python que permita insertar los datos del archivo 'customers-2000000.csv'
  - Primero creamos un proyecto para ejecutar los comandos para insertar datos.

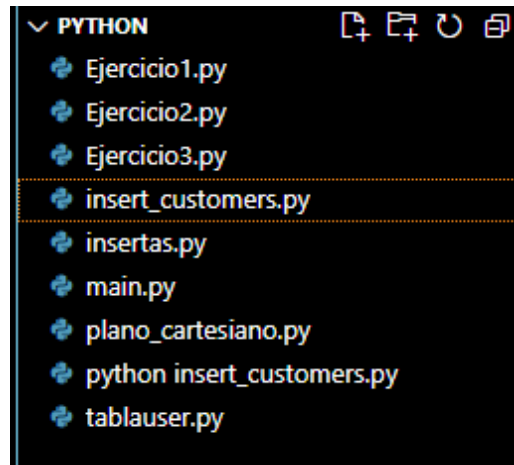


Ilustración 4 script insert\_customers.py

- Para que se conecte correctamente a la base de datos primero instalamos la librería para que se pueda conectar sin errores con el siguiente comando, **pip install mysql-connector-python**

```

C:\Windows\system32>pip install mysql-connector-python
Microsoft Windows [Versión 10.0.22000.2538]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Windows\system32>pip install mysql-connector-python
Collecting mysql-connector-python
  Using cached mysql_connector_python-9.0.0-cp312-cp312-win_amd64.whl.metadata
Downloading mysql_connector_python-9.0.0-cp312-cp312-win_amd64.whl (14.3 MB)
0.0/14.3 MB ? eta --:--
  
```

Ilustración 5 Instalación de librerías

- Posteriormente se verifica que la descarga haya sido exitosa.

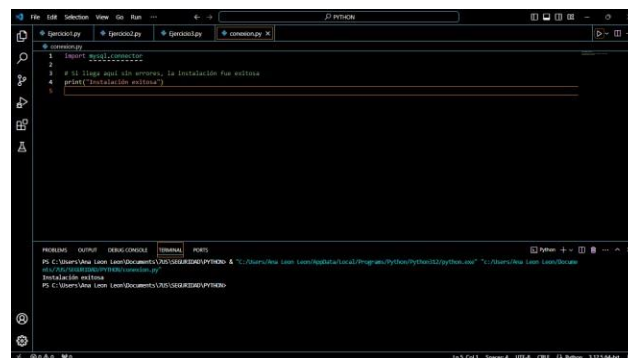


Ilustración 6 Librerías instaladas

- Y también instalamos el siguiente comando **pip install pandas** para que al ejecutar no nos marque ningún error

```
C:\WINDOWS\system32\cmd.exe - pip install pandas
Microsoft Windows [Versión 10.0.22000.2538]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Ana Leon Leon>pip install pandas
Collecting pandas
  Downloading pandas-2.2.2-cp312-cp312-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.26.0 in c:\users\ana leon leon\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2.1.0)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\ana leon leon\appdata\local\programs\python\python312\lib\site-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2024.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in c:\users\ana leon leon\appdata\local\programs\python\python312\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Downloading pandas-2.2.2-cp312-cp312-win_amd64.whl (11.5 MB)
-----
1.6/11.5 MB 56.5 KB/s eta 0:02:56
```

Ilustración 7 Instalación de librería pandas

- Realizamos la conexión con nuestra base de datos con el siguiente código

```
File Edit Selection View Go Run ... PYTHON
EXPLORER
PYTHON
  Ejercicio1.py
  Ejercicio2.py
  Ejercicio3.py
  insert_customers.py
  insertas.py
  main.py
  plano_cartesiano.py
  python insert_customers.py
python insert_customers.py
1 import mysql.connector
2 import pandas as pd
3
4 # Conexión a la base de datos
5 conn = mysql.connector.connect(
6     host="localhost",
7     user="root",
8     password="",
9     database="secure_db"
10 )
11
```

Ilustración 8 Script de conexión a base de datos

- Agregamos el script en Python para insertar datos el cual la primera línea de código establece conexiones a la base de datos secure\_db, y el customers inserta los campos como el customer\_id, nombre apellidos etc.

```
File Edit Selection View Go Run ... PYTHON
EXPLORER
PYTHON
  Ejercicio1.py
  Ejercicio2.py
  Ejercicio3.py
  insert_customers.py
  insertas.py
  main.py
  plano_cartesiano.py
  python insert_customers.py
python insert_customers.py
1 import mysql.connector
2 import pandas as pd
3
4 # Conexión a la base de datos
5 conn = mysql.connector.connect(
6     host="localhost",
7     user="root",
8     password="",
9     database="secure_db"
10 )
11
12 cursor = conn.cursor()
13
14 # Leer el archivo CSV
15 csv_file = "C:\Users\Ana Leon Leon\Documents\PROYECTOS\secure_db\customers_000000.csv"
16 print("Cargando el archivo CSV...")
17 df = pd.read_csv(csv_file)
18
19 # Mostrar las primeras filas para verificar que se leyó correctamente
20 print("Mostrando las primeras filas del CSV:")
21 print(df.head())
22
23 # Insertar datos en la tabla 'customers'
24 print("Insertando datos en la base de datos...")
25 for index, row in df.iterrows():
26     # Crear una lista de valores para insertar
27     values = [row['customer_id'], row['first_name'], row['last_name'], row['subscription_data'], row['website']]
28     # Insertar los datos en la base de datos
29     cursor.execute(f"INSERT INTO customers (customer_id, first_name, last_name, subscription_data, website) VALUES ({values})")
30
31 # Confirmar que los datos se insertaron correctamente
32 print("Datos insertados correctamente.")
33
```

Ilustración 9 Insertar datos del archivo customers a la base secure\_db

- Ahora volvemos a ejecutar el código para insertar datos en el campo y como se observa y está conectado a nuestra base de datos y se insertaron los datos correctamente.

Ilustración 10 Ejecuta el script para insertar datos

- En la siguiente ilustración observamos cómo se va subiendo nuestro archivo a nuestra base de datos y al finalizar nos dirá conexión cerrada y ya estará almacenado la información en nuestra base de datos.

Ilustración 11 Datos insertados correctamente a la base

- Finalmente se visualiza nuestra base de datos con datos del archivo de customers.

id	customer_id	first_name	last_name	email	phone_number	website
1	1	John	Deere	john.deere@deere.com	1-800-447-DEERE	http://www.deere.com
2	2	Case	IH	case@ih.com	1-800-447-DEERE	http://www.caseih.com
3	3	John	Deere	john.deere@deere.com	1-800-447-DEERE	http://www.deere.com
4	4	Case	IH	case@ih.com	1-800-447-DEERE	http://www.caseih.com
5	5	John	Deere	john.deere@deere.com	1-800-447-DEERE	http://www.deere.com
6	6	Case	IH	case@ih.com	1-800-447-DEERE	http://www.caseih.com
7	7	John	Deere	john.deere@deere.com	1-800-447-DEERE	http://www.deere.com
8	8	Case	IH	case@ih.com	1-800-447-DEERE	http://www.caseih.com
9	9	John	Deere	john.deere@deere.com	1-800-447-DEERE	http://www.deere.com
10	10	Case	IH	case@ih.com	1-800-447-DEERE	http://www.caseih.com
11	11	John	Deere	john.deere@deere.com	1-800-447-DEERE	http://www.deere.com
12	12	Case	IH	case@ih.com	1-800-447-DEERE	http://www.caseih.com
13	13	John	Deere	john.deere@deere.com	1-800-447-DEERE	http://www.deere.com
14	14	Case	IH	case@ih.com	1-800-447-DEERE	http://www.caseih.com
15	15	John	Deere	john.deere@deere.com	1-800-447-DEERE	http://www.deere.com
16	16	Case	IH	case@ih.com	1-800-447-DEERE	http://www.caseih.com
17	17	John	Deere	john.deere@deere.com	1-800-447-DEERE	http://www.deere.com
18	18	Case	IH	case@ih.com	1-800-447-DEERE	http://www.caseih.com
19	19	John	Deere	john.deere@deere.com	1-800-447-DEERE	http://www.deere.com
20	20	Case	IH	case@ih.com	1-800-447-DEERE	http://www.caseih.com

Ilustración 12 Datos insertados en mysql

### 3. Crea tres usuarios en MySQL

- Usuario 1: Permisos de lectura en la tabla `customers`
- Usuario 2: Permisos de lectura y escritura en la tabla `address`



### 🚦 Usuario 3: Permisos de lectura, escritura y eliminación en la tabla `users`

Principalmente se creó una base de datos con nombre `secure_db` para después crear sus respectivas tablas como fue la `customers`.



Ilustración 13 Se crea la tabla `customers`

🚦 Para otorgar permisos de solo lectura (SELECT) sobre la tabla `customers` al Usuario1 se utilizó el siguiente código GRANT SELECT ON nombre\_base\_datos.customers TO 'usuario'@'localhost';

🚦 FLUSH PRIVILEGES;

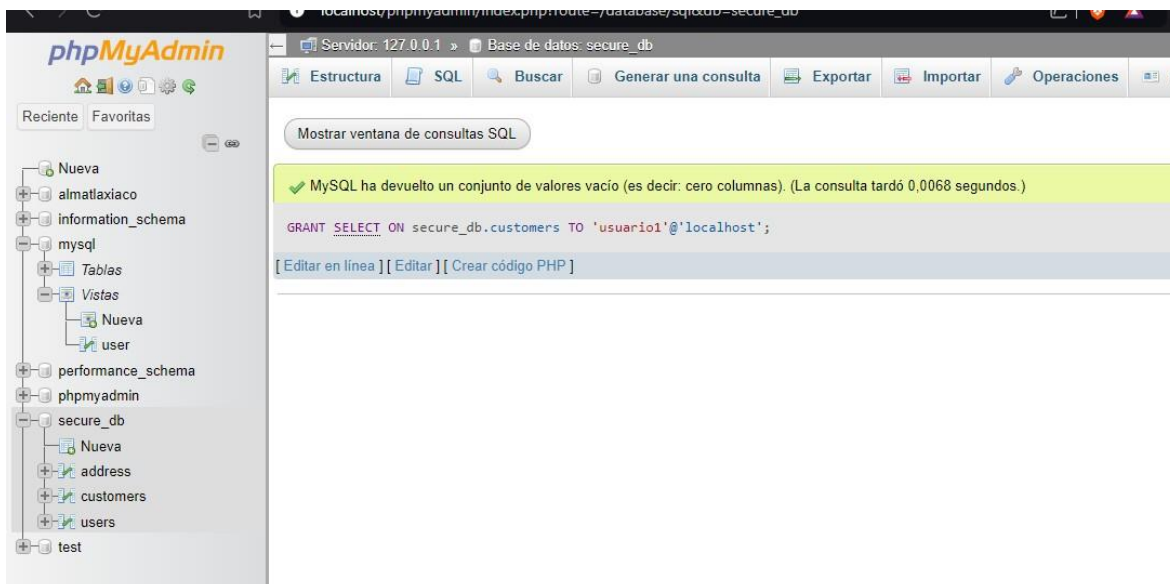


Ilustración 14 Permisos al usuario

En la imagen se está mostrando los permisos de un usuario. La consulta `SHOW GRANTS FOR 'usuario1'@'localhost';` devuelve los privilegios que tiene el usuario `usuario1` en el



servidor localhost.



Ilustración 15 Consulta SHOW GRANTS

Para crear la tabla address para el usuario 2 se tuvo que realizar lo siguiente:

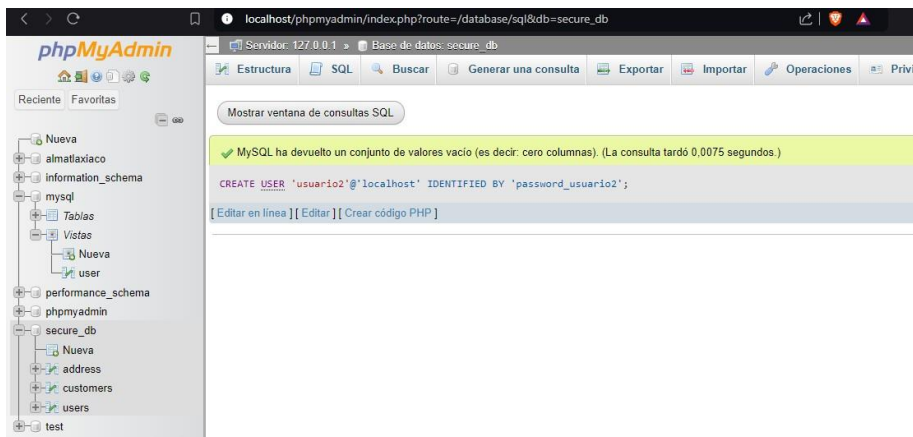


Ilustración 16 Se crea tabla address para usuario

Para verificar la consulta se ejecutó el siguiente comando GRANT SELECT, INSERT, UPDATE ON en la base de datos secure\_db en la tabla de address para darle los permisos al usuario2.

Luego, se ejecuta el comando FLUSH PRIVILEGES, que actualiza los privilegios sin necesidad de reiniciar el servidor.



Ilustración 17 Se ejecuta GRANT SELECT

Como se puede observar SELECT user, host: Este comando indica que deseas seleccionar columnas de la tabla address.

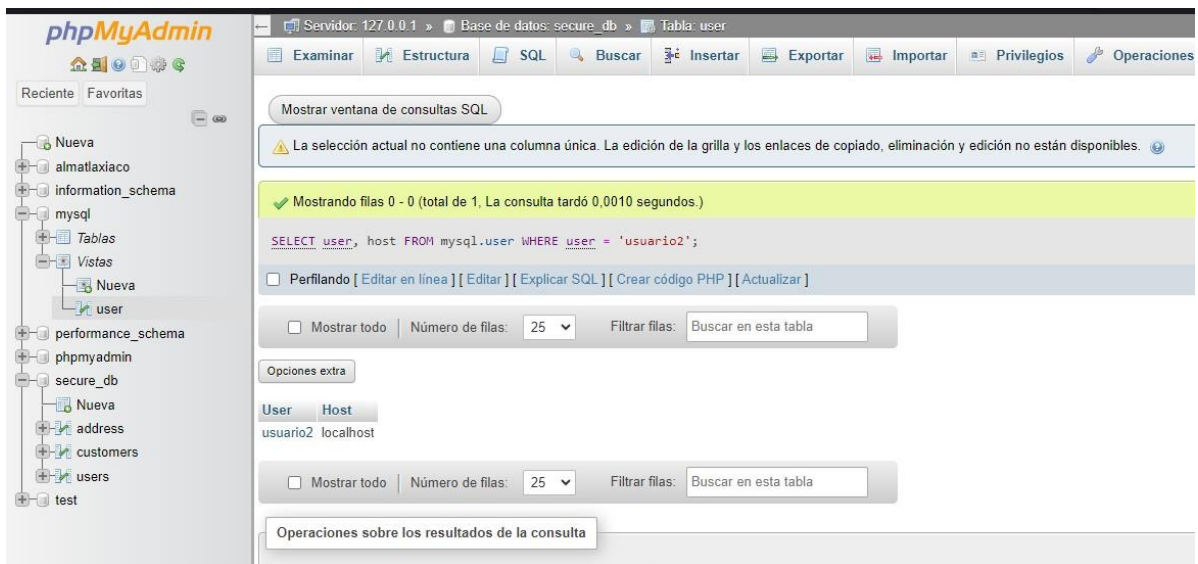


Ilustración 18 Se ejecuta SELECT user

Para el usuario 2 se tiene que darle permisos de lectura, escritura y eliminación en la tabla 'users'

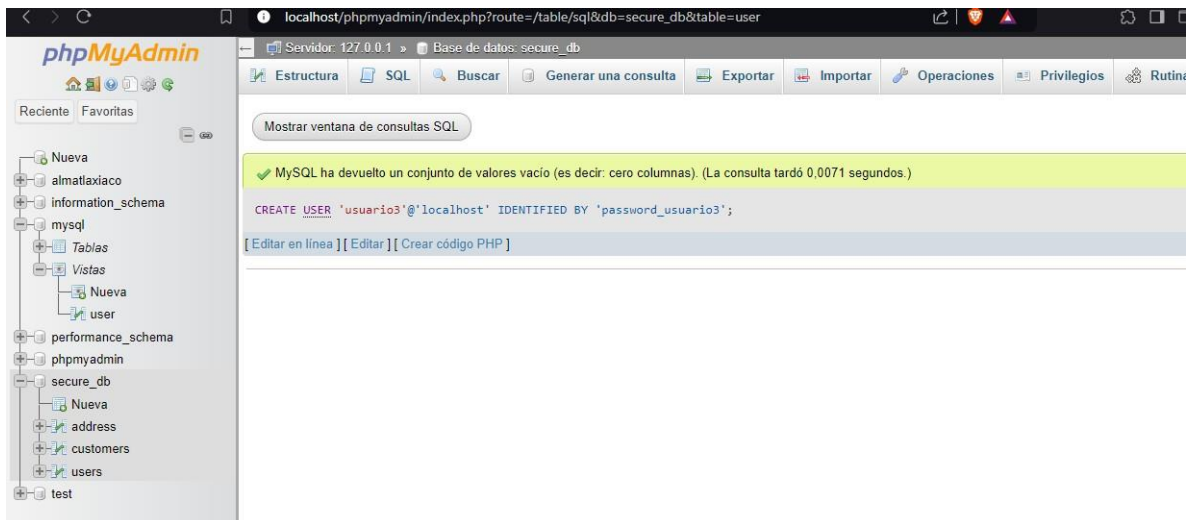


Ilustración 19 Asignar permisos al usuario 2

En la imagen se está ejecutando el siguiente comando **SHOW GRANTS FOR 'username'@'host';** Esto muestra los privilegios que tiene el usuario username en el host host. El resultado indica que el usuario tiene permisos de uso general, así como privilegios específicos de selección, inserción, actualización y eliminación en la base de datos.

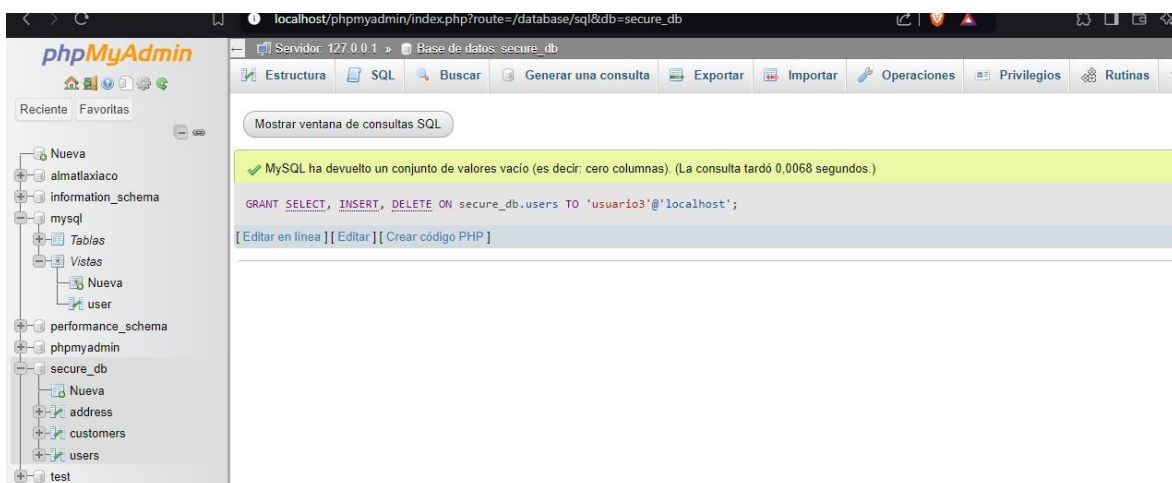


Ilustración 20 Grant select

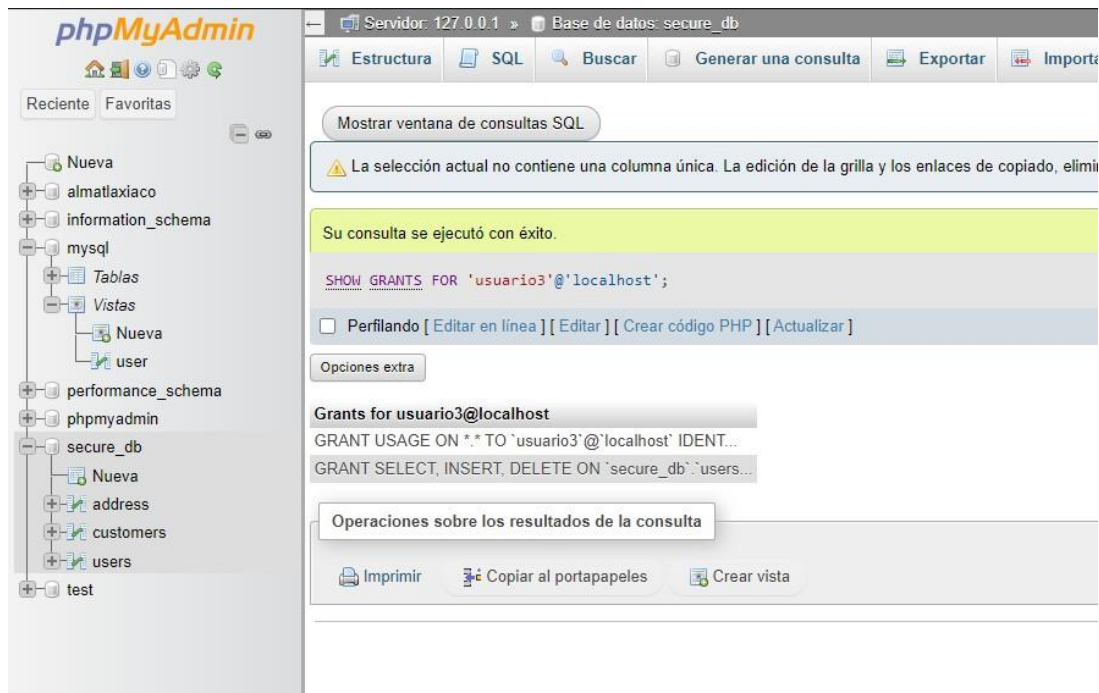


Ilustración 21 Consulta SHOW GRANTS

4. Crea un script en Python que permita realizar una inyección de sql en la tabla 'users' y que muestre los datos de la tabla 'users' en la consola.

Se crea un script en Python para insertar datos el cual la primera línea de código establece conexiones a la base de datos secure\_db, y siguiendo con la tabla users, a la cual se le agregarán los registros.

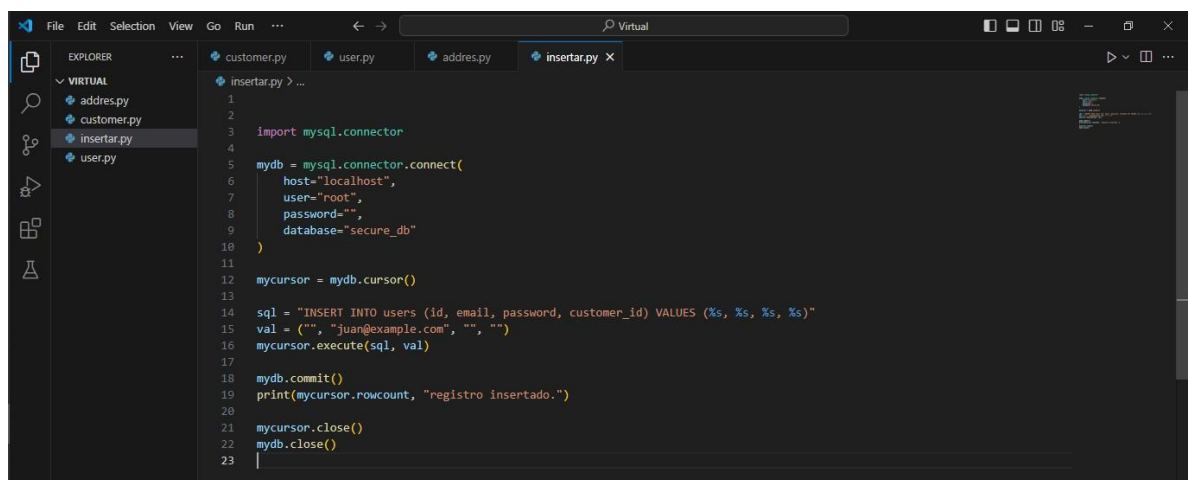
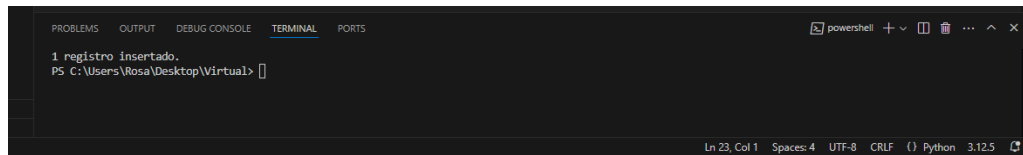


Ilustración 22 Insertar datos a la tabla users

Una vez ejecutado el script, si es correcto el registro nos arroja el mensaje “registro insertado”



```
1 registro insertado.
PS C:\Users\Rosa\Desktop\Virtual>
```

Ilustración 23 Ejecutar script y resultado

Se visualiza el dato insertado en la tabla ‘users’ con la consola. En este caso seria el numero 11.

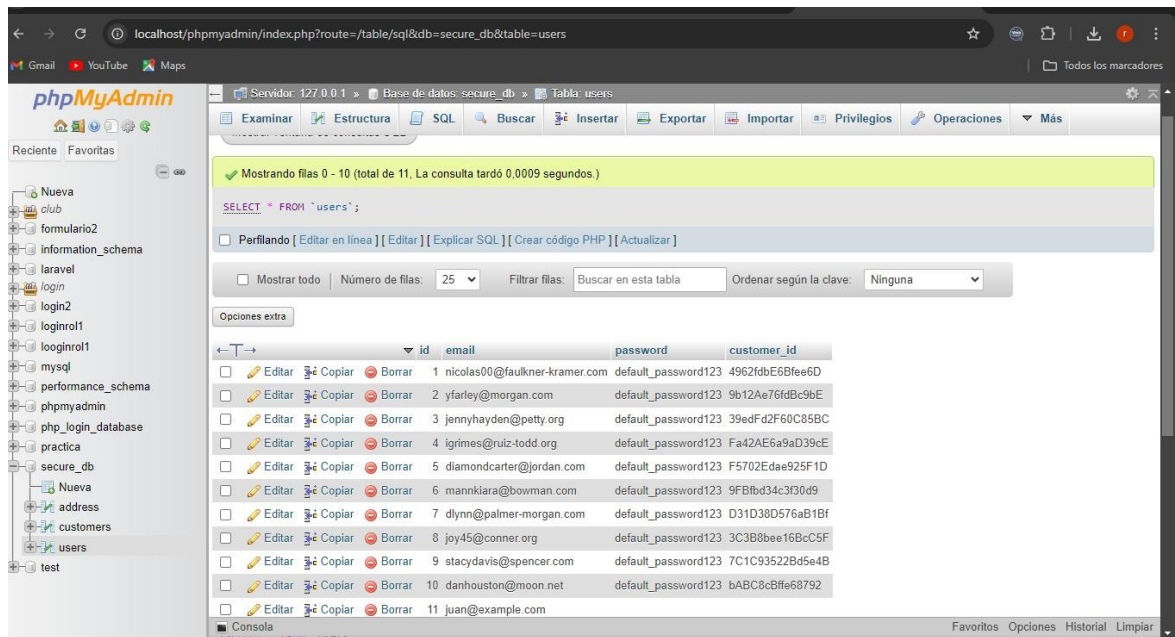


Ilustración 24 Visualizar la tabla users

5. Crea un backup de la base de datos ‘secure\_db’ y restaura la base de datos en un servidor diferente.
- Primero se abre Xampp y asegurarnos de que MySQL se esté ejecutando.

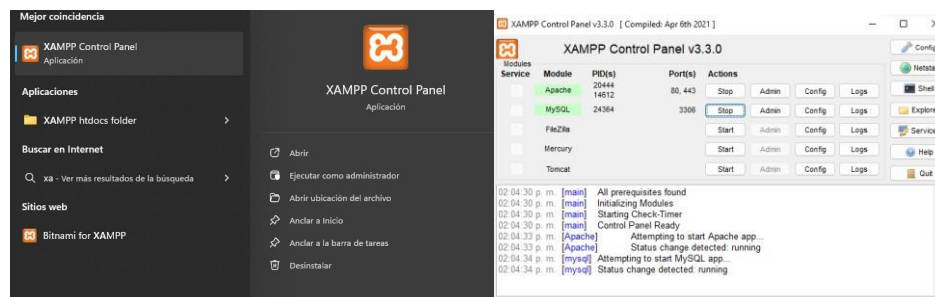
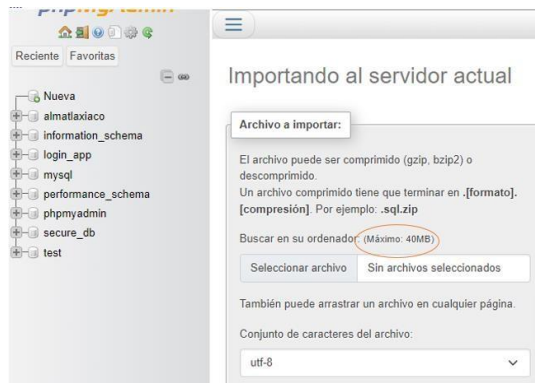


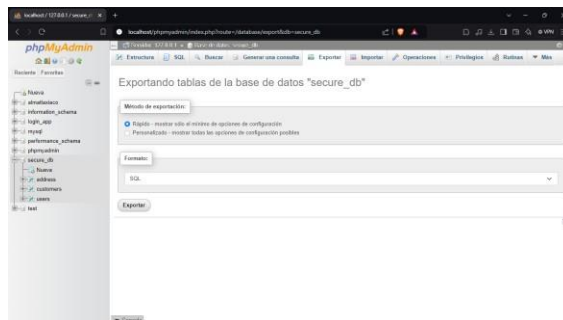
Ilustración 25 xampp

- Posteriormente se verifica cuando pesa la base de datos en este caso tiene un máximo de 40MB.



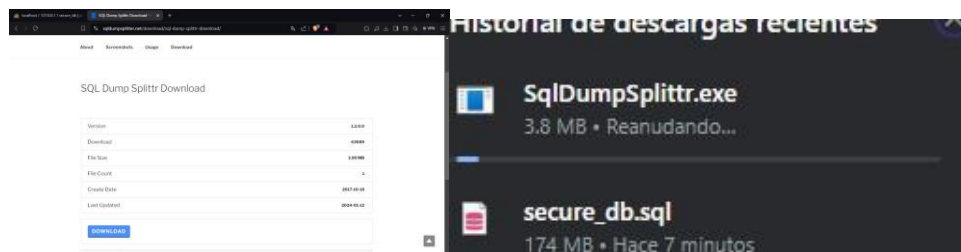
*Ilustración 26 Peso de la base de datos*

- A continuación, exportamos la base de datos.



*Ilustración 27 Exporta base de datos*

Como siguiente paso utilizamos el programa de SQL, este programa nos ayuda a dividir la base de datos en varias partes, a continuación, descargamos la aplicación para posterior utilizarla de manera adecuada



*Ilustración 28 Programa SQL*

- Seleccionamos nuestra base de datos llamada secure\_db.sql

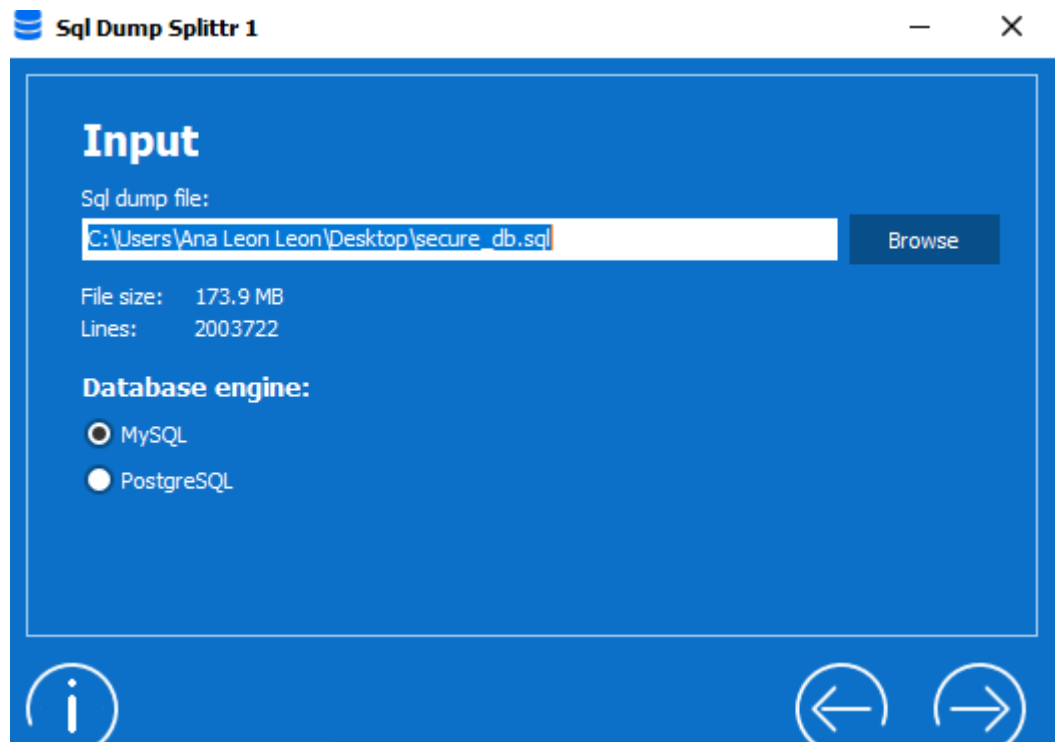


Ilustración 29 Selección de base de datos secure\_db

- En este apartado se muestra en cuantas partes se requiere dividir la base de datos en este caso lo dejamos en kilobyte esto dependiendo del tamaño que contenga el servidor

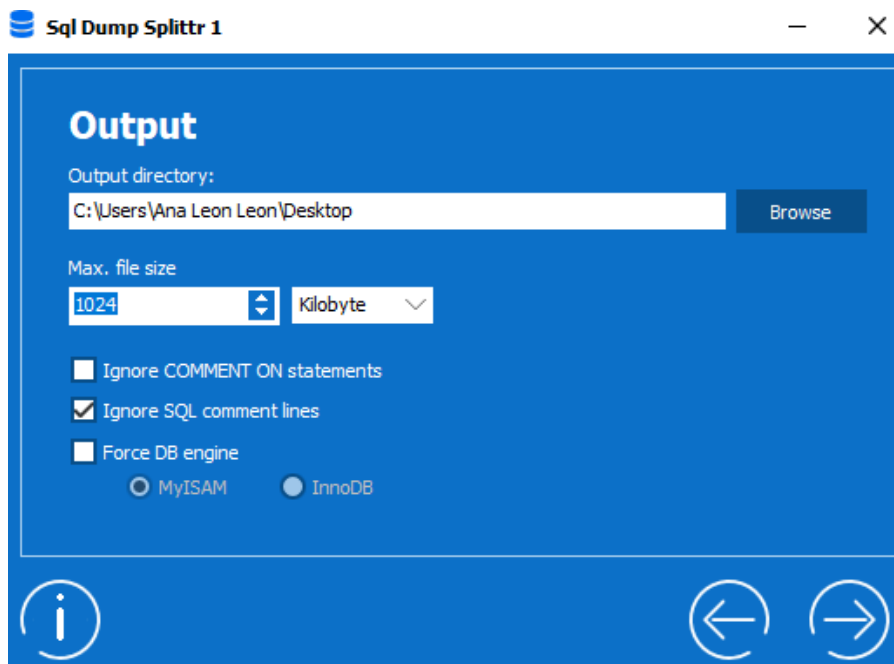


Ilustración 30 dividir la base de datos



En este apartado nos está indicando en donde se va guardar nuestra base de datos en nuestro caso lo colocamos en escritorio como se puede visualizar en la imagen.

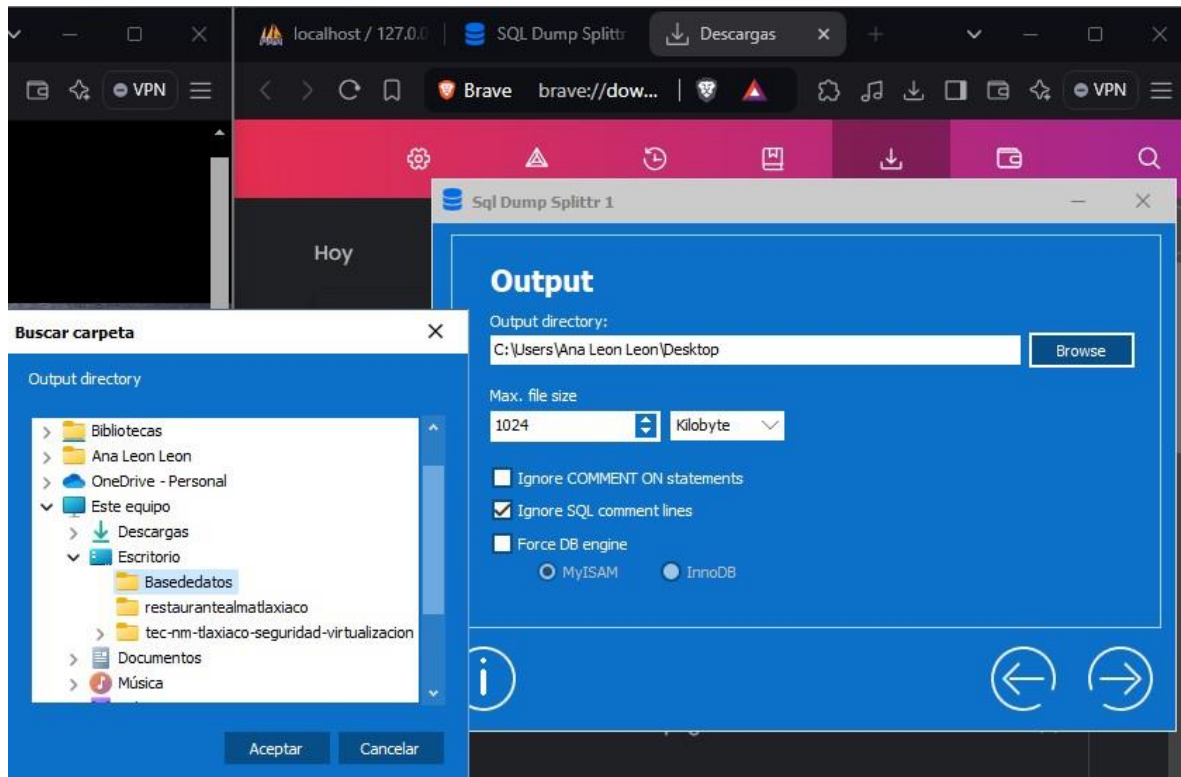


Ilustración 31 Dirección de la base de datos

Una vez terminando su ejecución, ingresamos a la carpeta Basededatos nos arroja a los resultados y como se puede visualizar la página nos la dividió en todas estas partes como se muestra en la siguiente imagen.

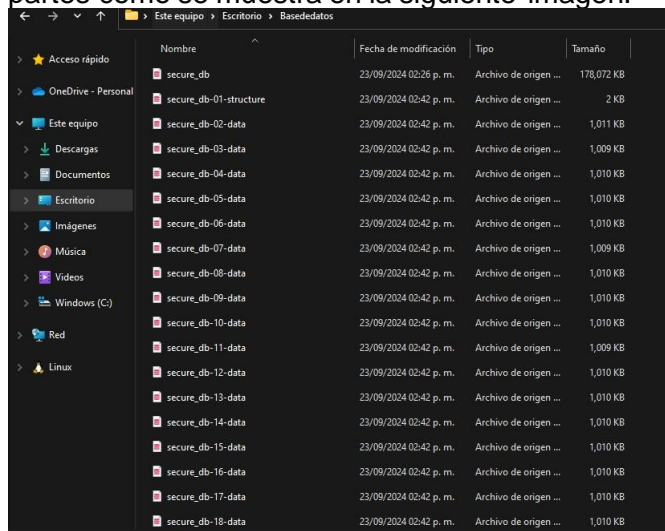
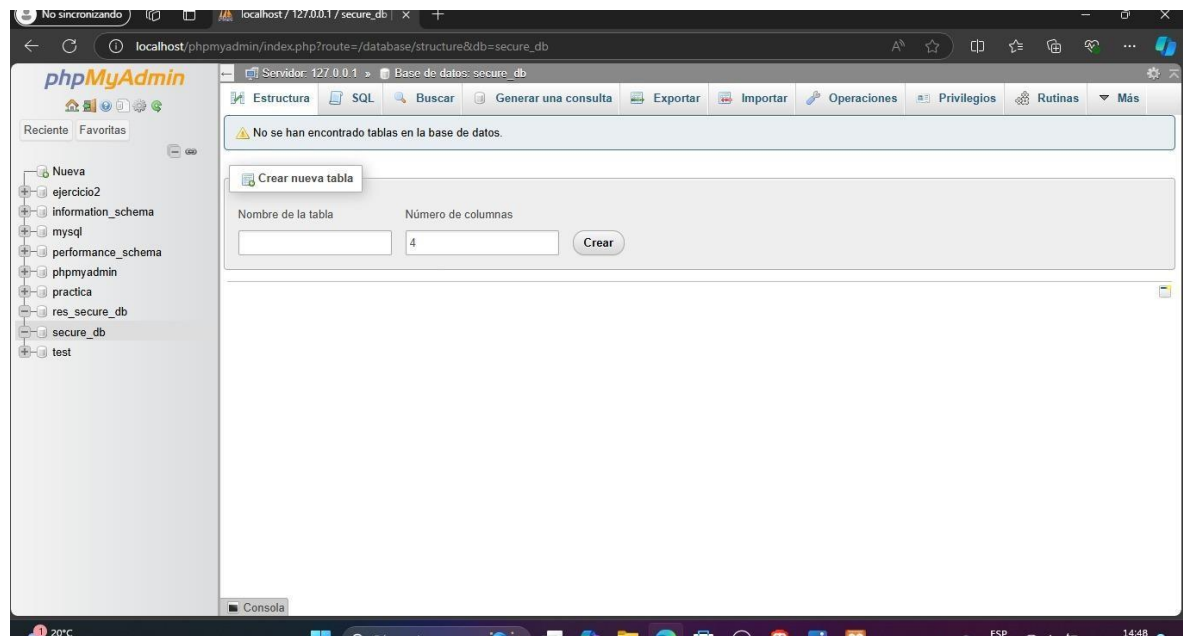


Ilustración 32 Visualizar el resultado de dividir la base de datos

- A continuación, a nuestro nuevo servidor creamos una base de datos con el nombre `sucere_db`.



*Ilustración 33 Visualiza en mysql*

## CONCLUSIÓN

Este proyecto ha permitido desarrollar una base de datos estructurada y eficiente en MySQL, demostrando la importancia de gestionar adecuadamente la información de usuarios, direcciones y clientes mediante el uso de tablas interrelacionadas. La automatización de la inserción de datos masivos desde un archivo CSV mostró la capacidad de manejar grandes volúmenes de información de manera eficiente. Además, la implementación de usuarios con permisos específicos reforzó las buenas prácticas de seguridad, limitando el acceso según las necesidades de cada usuario. La demostración de una vulnerabilidad de inyección SQL subrayó la importancia de proteger las aplicaciones contra este tipo de ataques. Finalmente, el respaldo y la restauración de la base de datos garantizaron la continuidad operativa y la seguridad de los datos en caso de fallos o migraciones, consolidando las mejores prácticas en la gestión y administración de bases de datos.