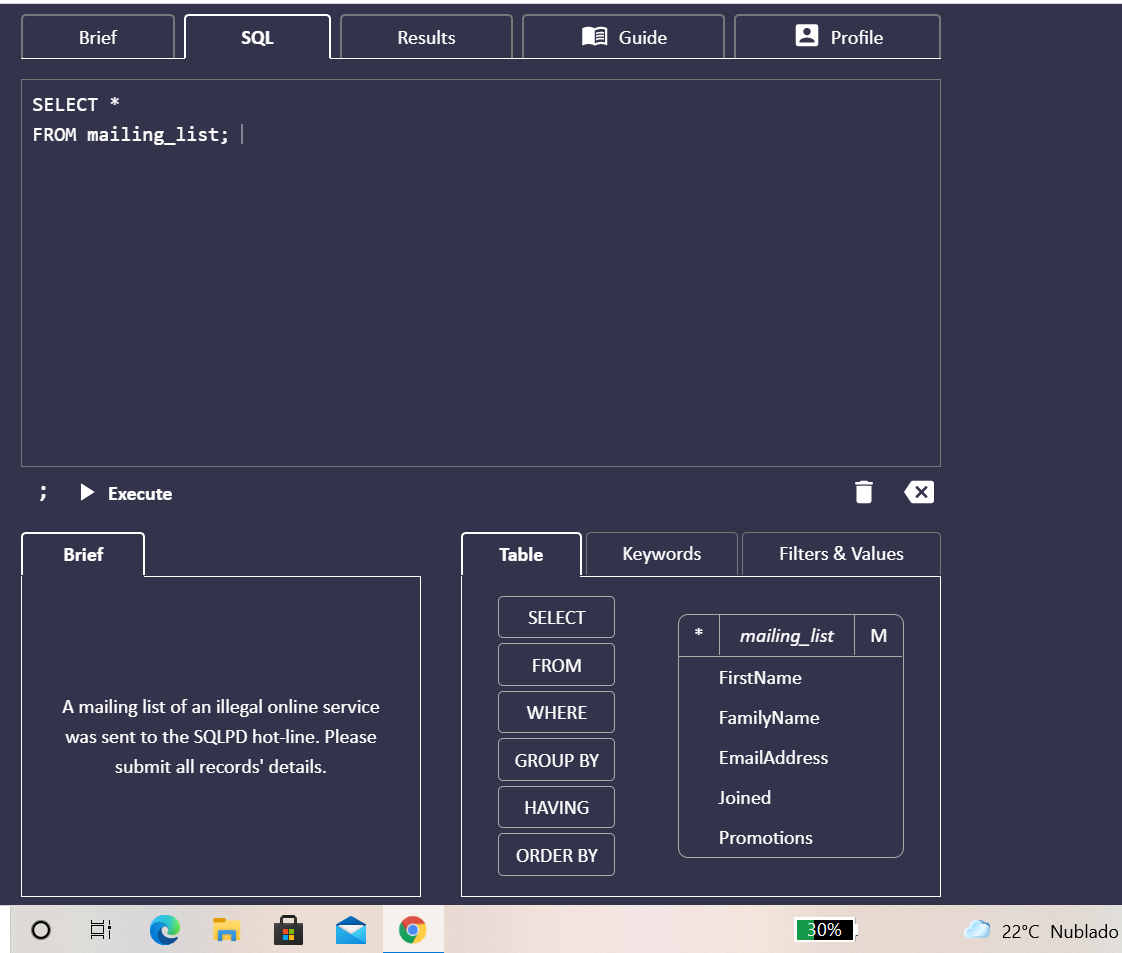
**"Un cracker ha accedido a una información privada y tú, que eres "El inspector SQL", debe tomar una serie de decisiones mediante sentencias SQL para resolver el caso".**

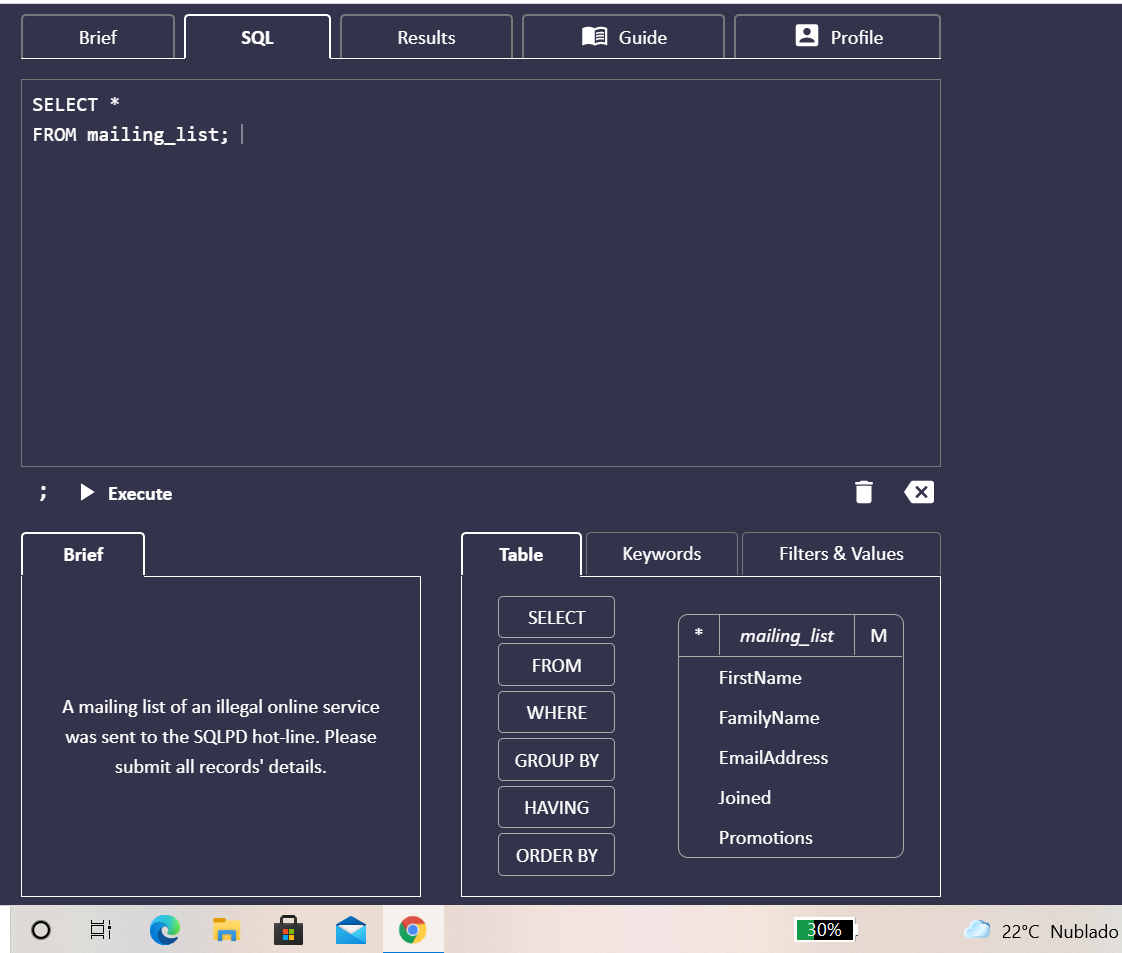
Escribe en un documento de texto (.doc, .txt, ...) la sentencia SQL que has utilizado para cada caso resuelto junto con una breve explicación sobre la sentencia que has utilizado.

El lenguaje SQL está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular la base de datos. A bajo se muestran una serie de casos a las que se les ha dado solución utilizando diferentes sentencias de SQL.

1. Mostrar todos los registros de la tabla

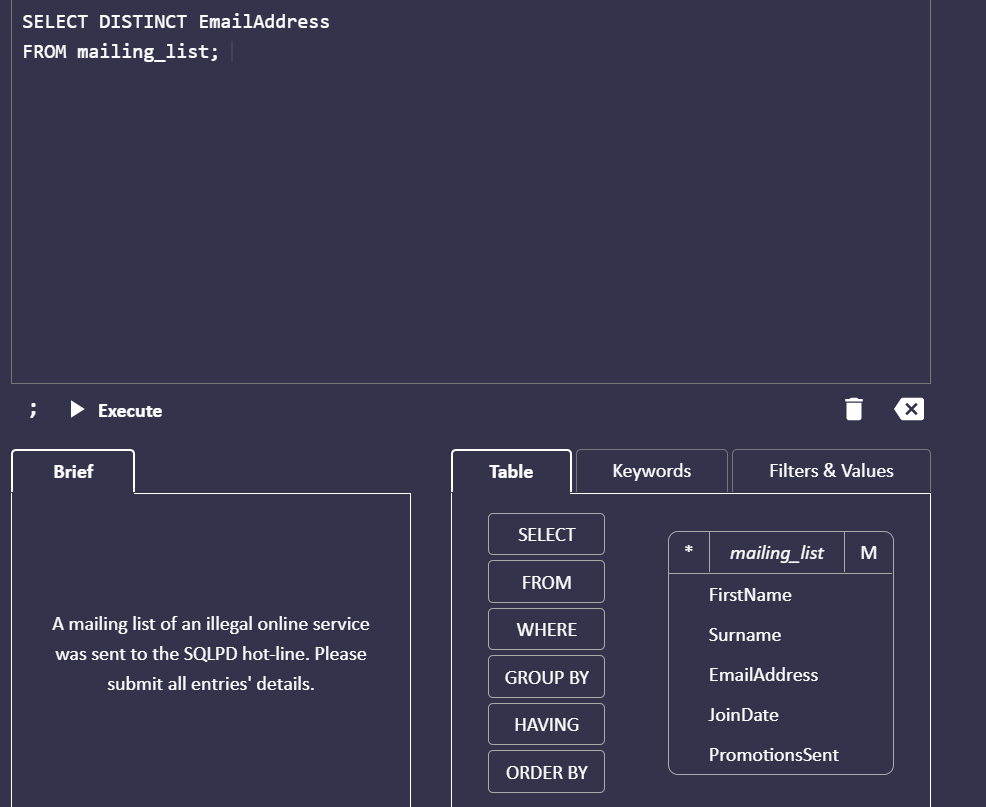


Sentencia SQL utilizada: **SELECT \* FROM mailing\_list;**

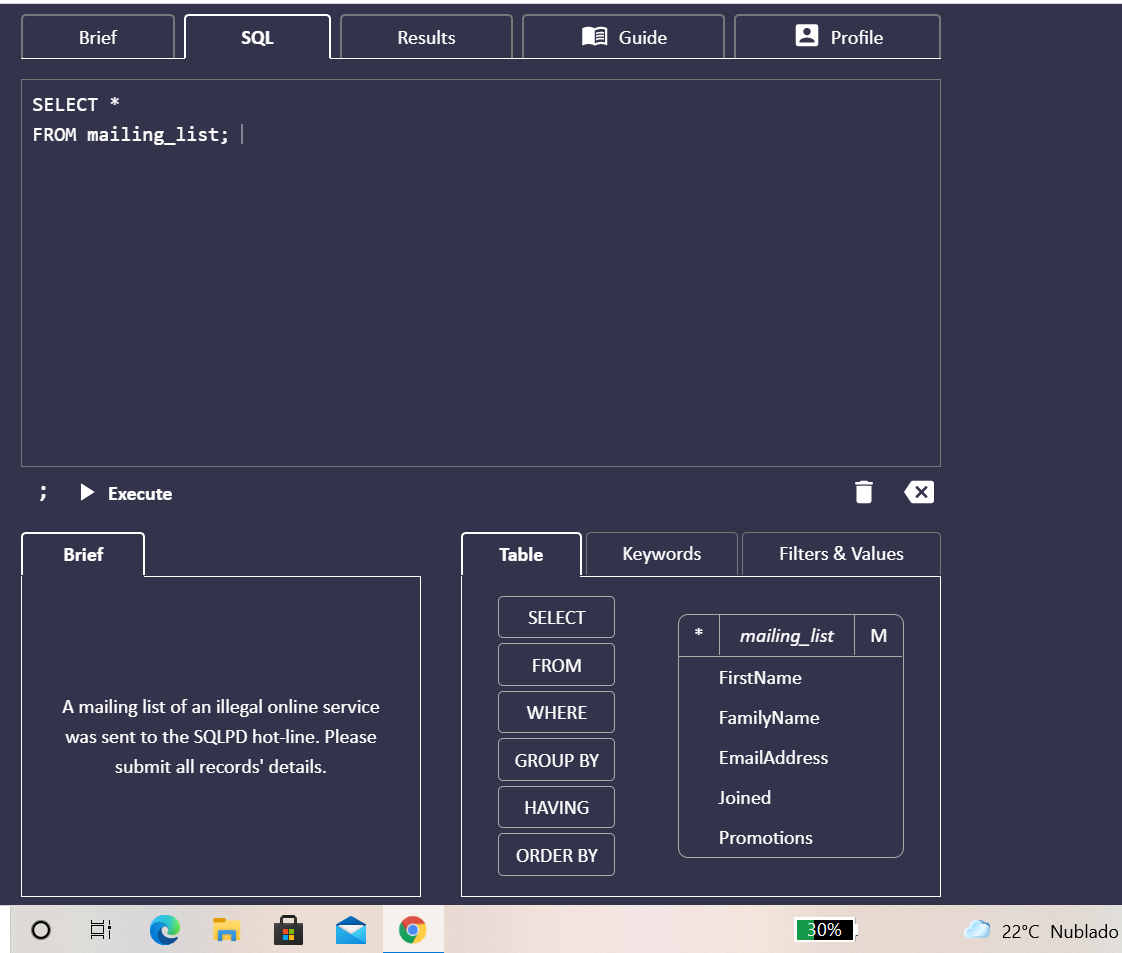


Esta sentencia devuelve todos los registros de la tabla “mailing\_list”. Para que la sentencia devuelva todas las columnas con todos los registros es necesario especificar el signo \* detrás de la SELECT.

1. Mostrar todas las entradas de la tabla

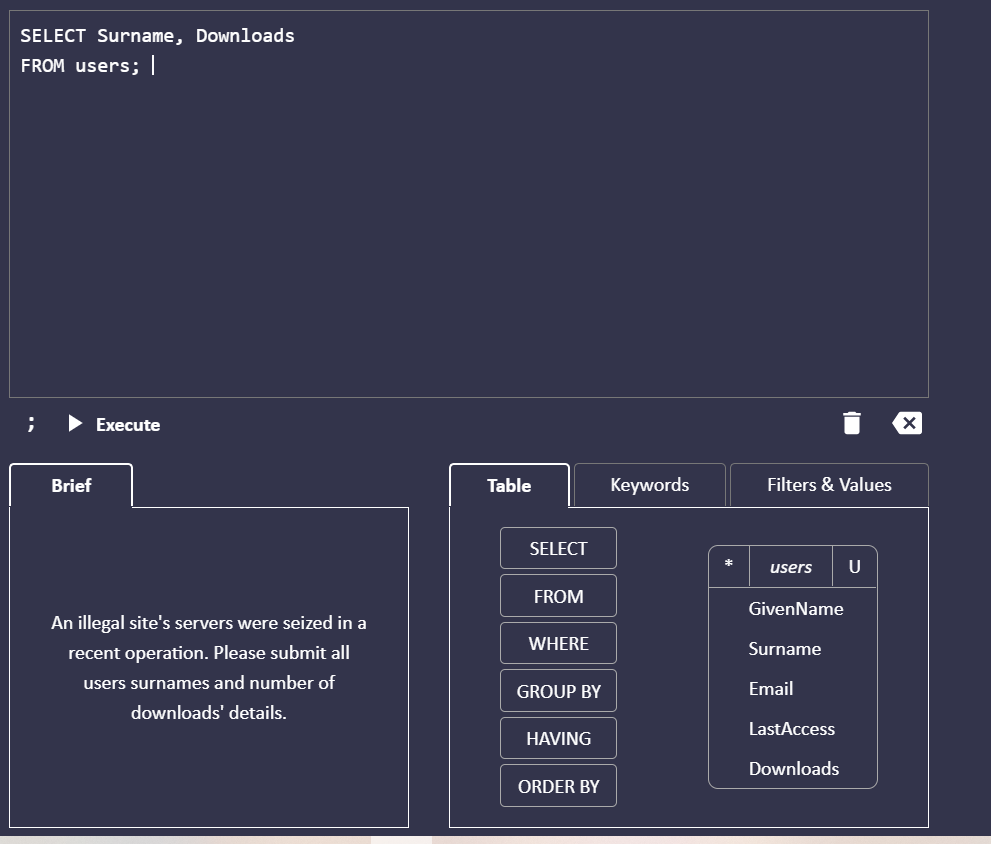


Sentencia SQL utilizada: **SELECT \* FROM mailing\_list;**

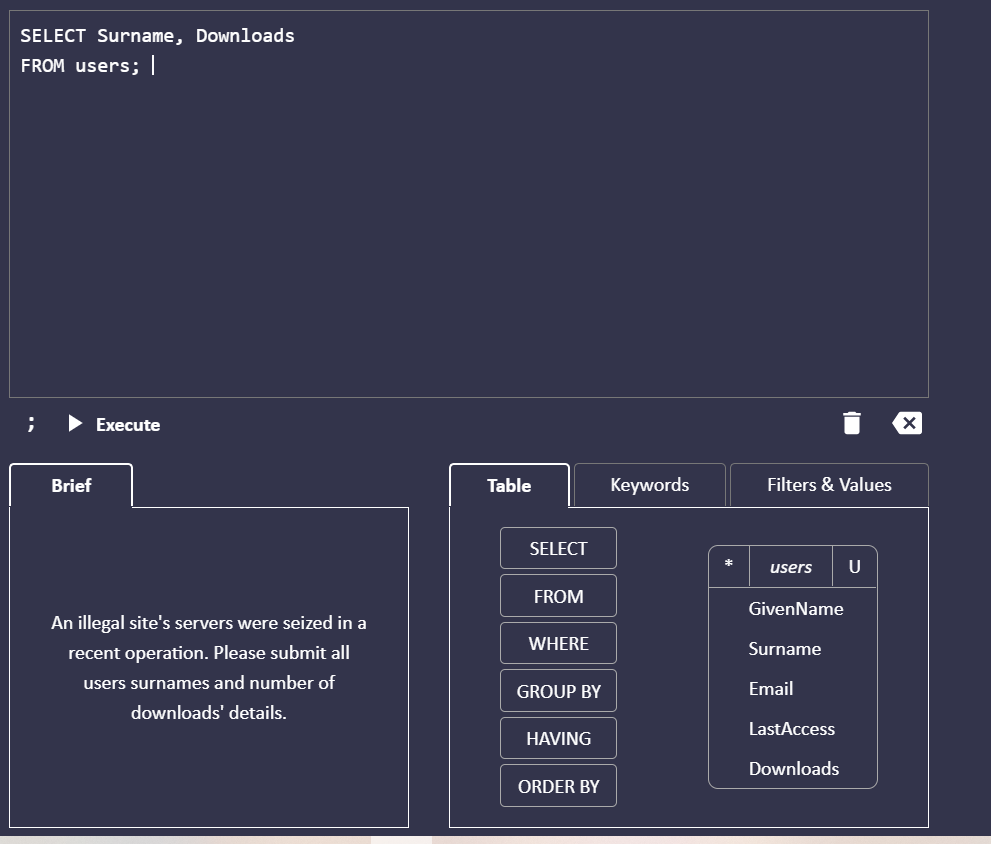


Al igual que en el primer punto la query devuelve todos los registros que tiene la tabla, ya que cada registro es una entrada.

1. Mostrar algunas columnas de la tabla

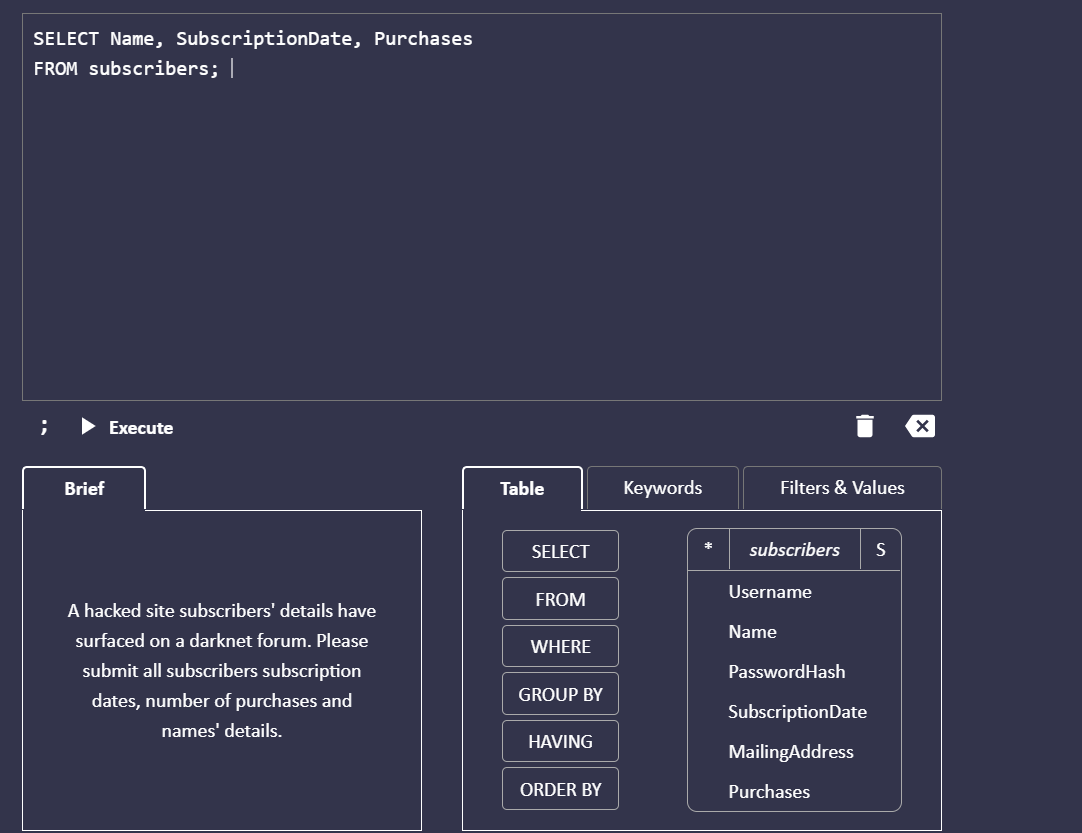


Sentencia SQL utilizada: **SELECT surname, Downloads FROM users;**

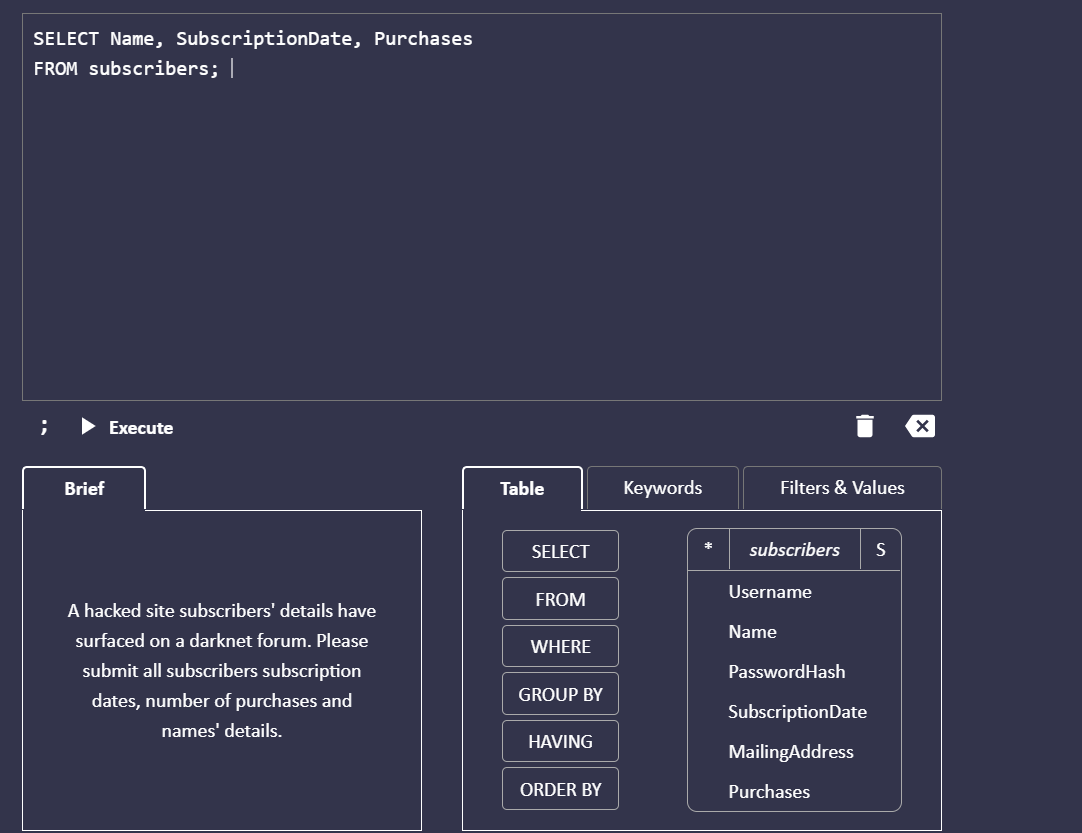


En esta sentencia en vez de el signo \*, hemos utilizado el nombre de las columnas a las que queremos acceder. La query solo va a devolver los registros que contienen las dos columnas especificadas.

1. Mostrar más de dos columnas de la tabla

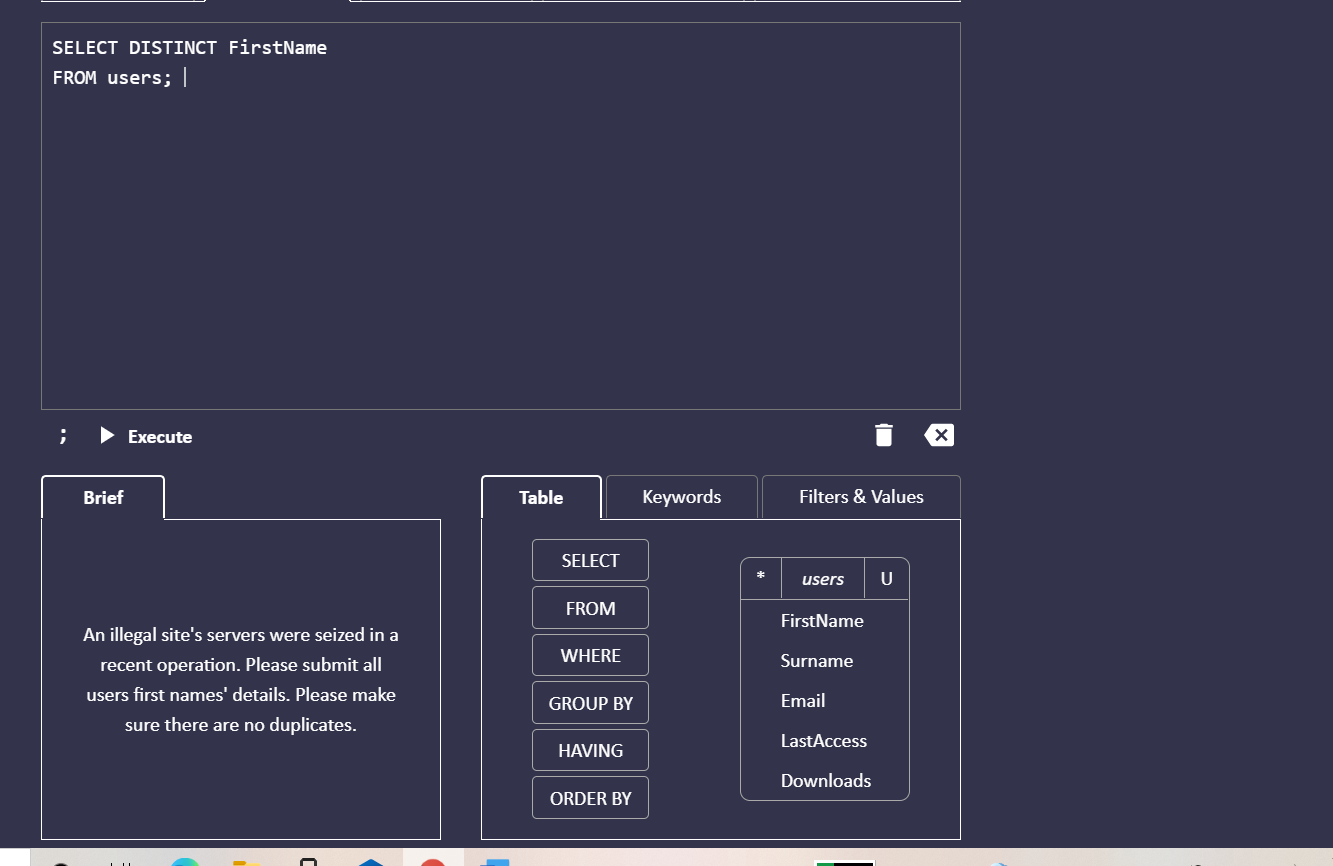


Sentencia SQL utilizada: **SELECT name, SubscriptionDate, Purchases FROM suscribers;**

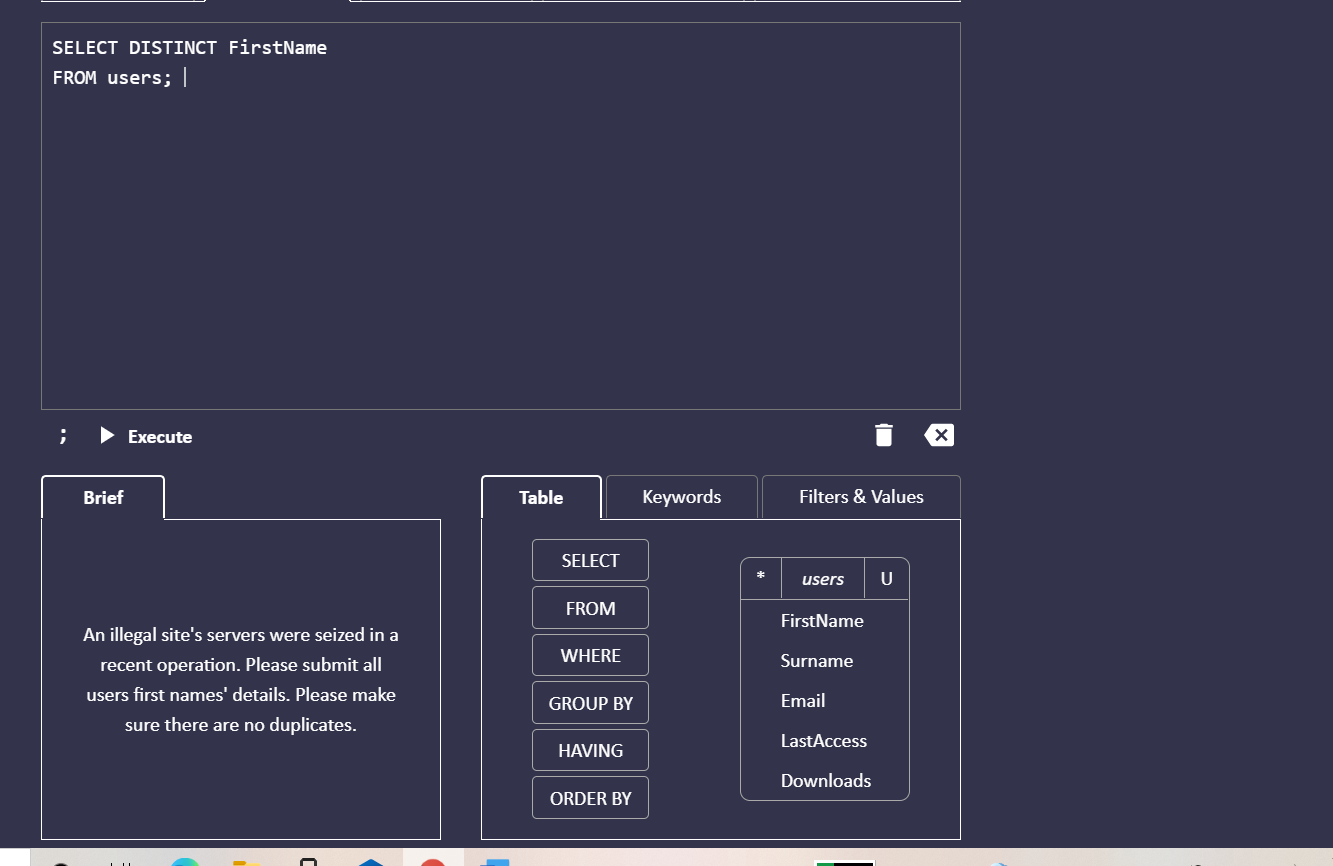


Esta sentencia devuelve los registros almacenados de la tabla “subscribes”, pero solamente en las columnas “Name”, “SubscriptipnDate” y “Purchaises”.

1. Mostrar datos no duplicados

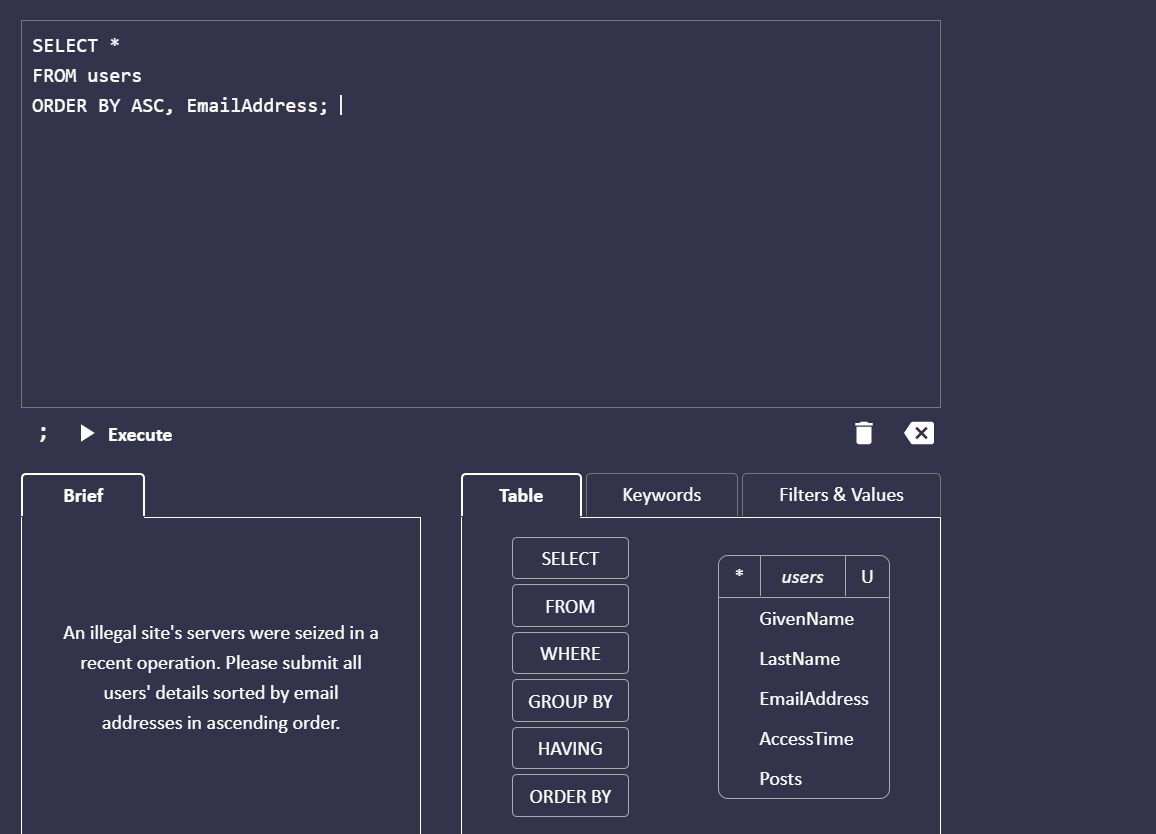


Sentencia SQL utilizada: **SELECT DISTINCT FirstName FROM users;**



Esta sentencia devuelve todos los nombres registrados únicos y sin duplicados en la tabla users. La cláusula “DISTICT” es la palabra clave que nos permite eliminar todos los duplicados a la hora de mostrar los datos al usuario. Esta palabra clave siempre se coloca al principio de la sentencia después de la SELECT. Elimina todos los registros duplicados que devuelve la query definida.

1. Mostrar datos ordenados ascendentemente.



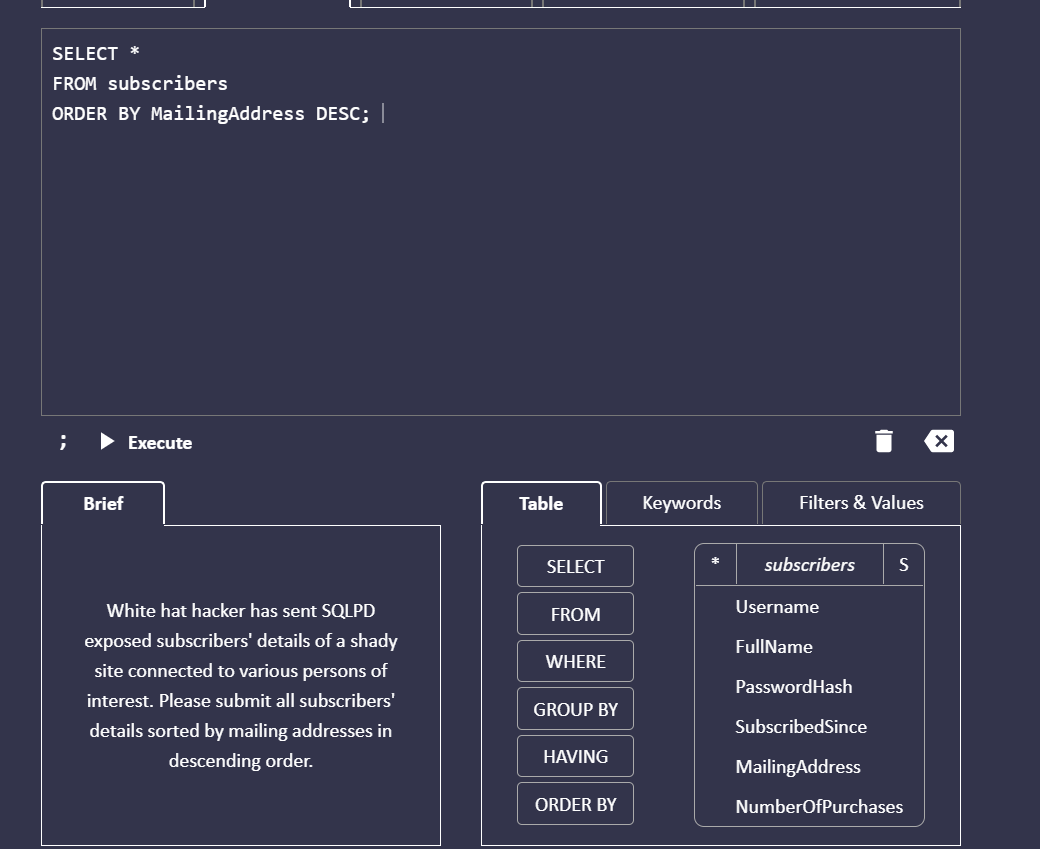
Sentencia SQL utilizada: **SELECT \* FROM users ORDER BY EmailAddress ASC;**



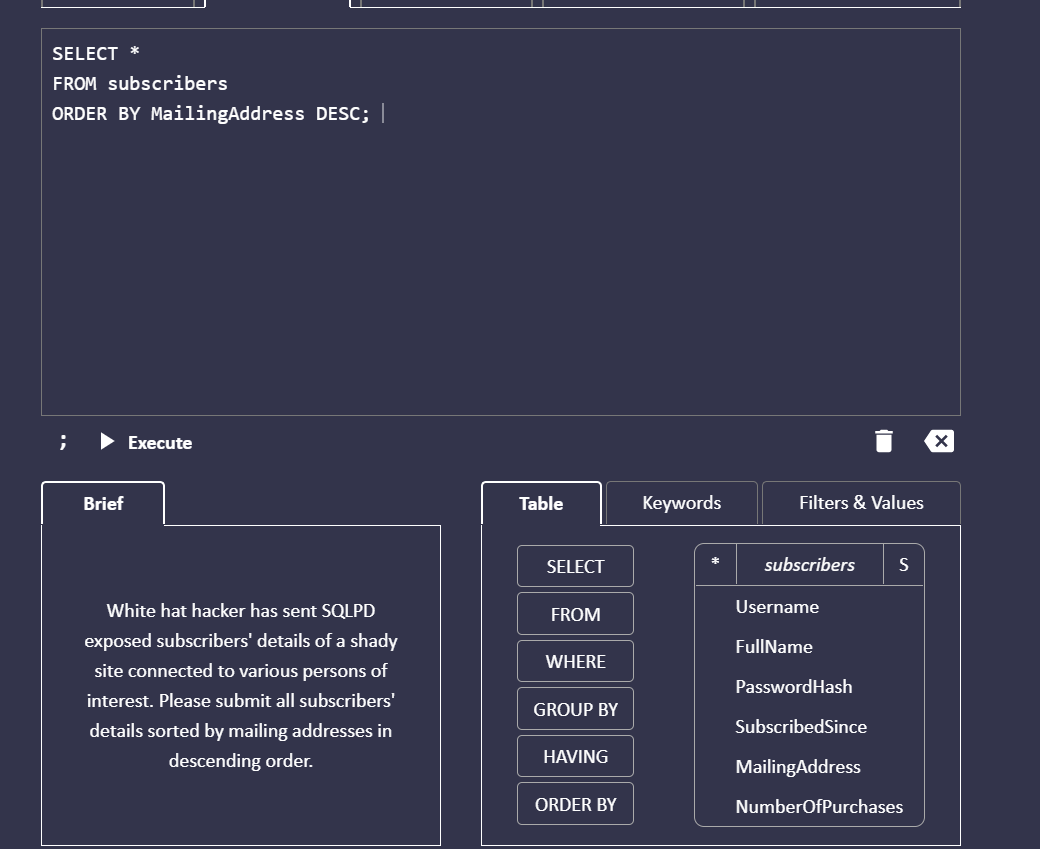
Esta sentencia devuelve todos los registros de la tabla ordenados por la columna EmailAddress de forma ascendente.

“ORDER BY” es la palabra clave que nos permitirá devolver datos ordenados de la manera que se especifique. Esta palabra clave siempre se coloca al final de la sentencia, detrás de la tabla y seguido de la columna/s que se quiera/n ordenar. Los datos se pueden ordenar de forma ascendente o descendente. ASC permite ordenar los datos de forma ascendente. Estas siglas se colocan detrás de la columna que se quiera ordenar. Si no se especifica nada detrás de la columna, por defecto ORDER BY ordena los datos de forma ascendente.

1. Mostrar datos ordenados descendentemente.



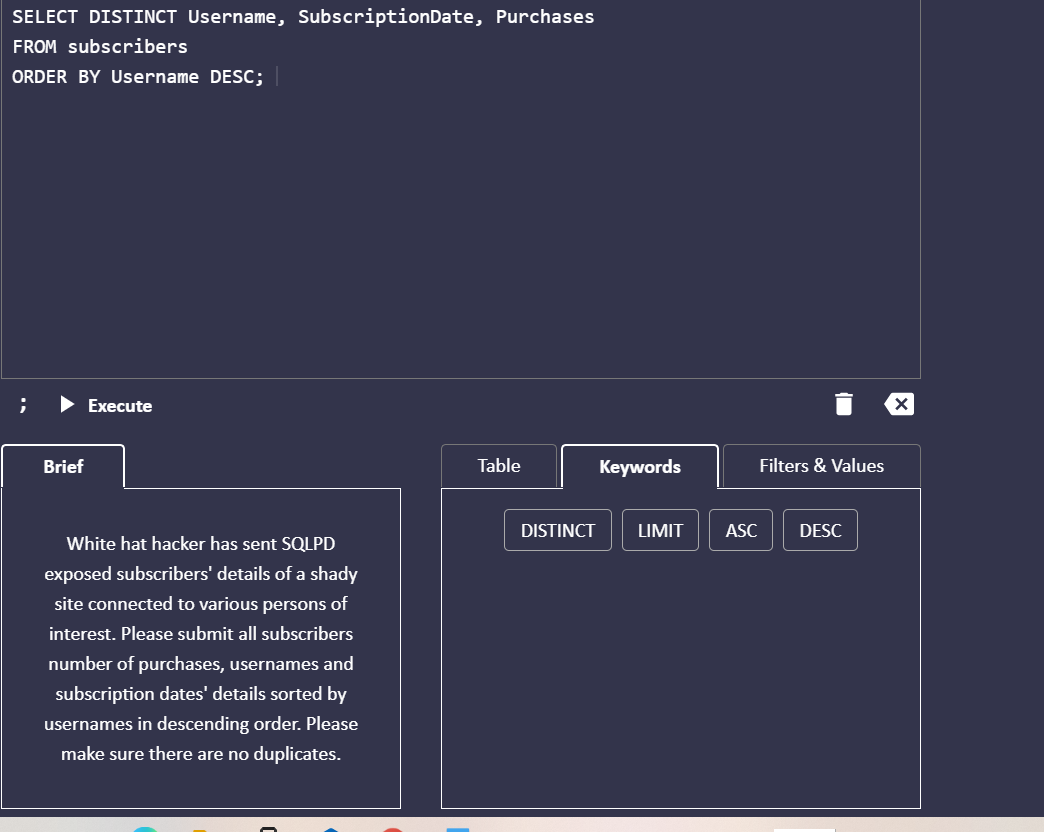
Sentencia SQL utilizada: **SELECT \* FROM subscribers ORDER BY MailingAddress DESC;**



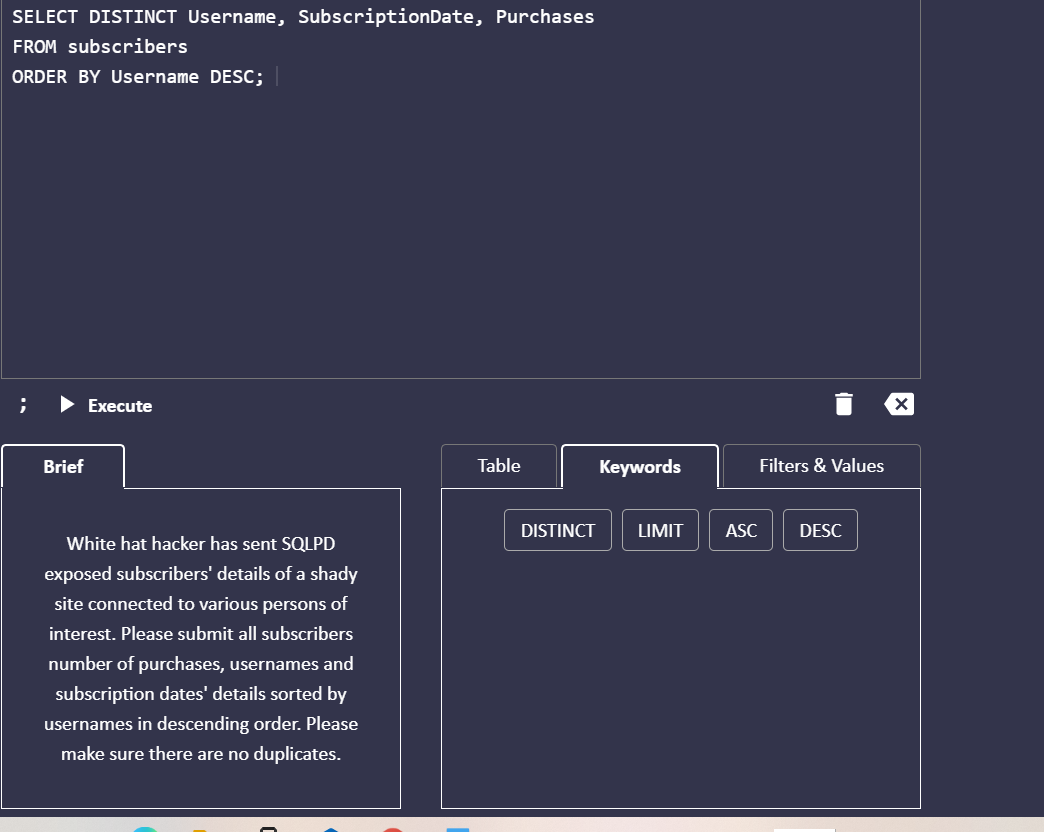
Esta sentencia devuelve todos los registros de la tabla ordenados por la columna MailingAddressde forma descendente.

Si se quiere ordenar los datos de forma descendente es necesario especificar la palabra “DESC” detrás de la columna.

1. Mostrar datos únicos y ordenados

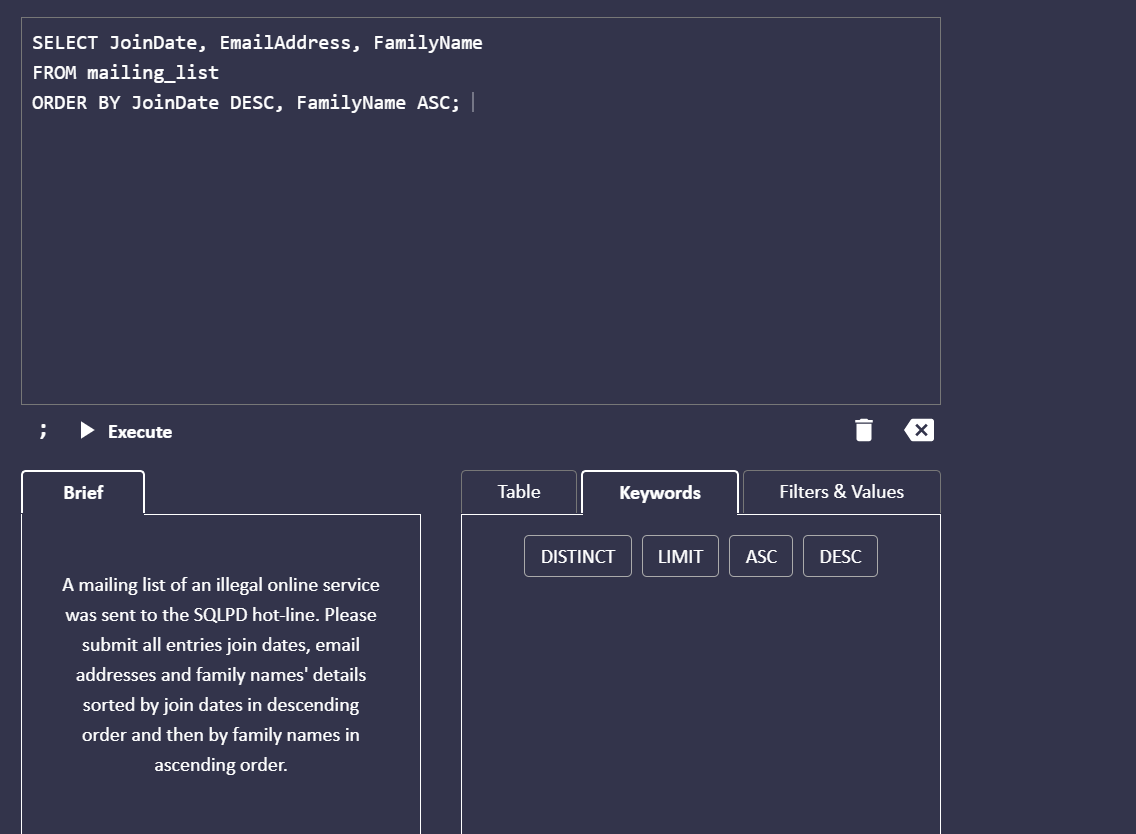


Sentencia SQL utilizada: **SELECT DISTINCT Username, SubscriotionDate, Purchases FROM subscribers ORDER BY Username DESC;**

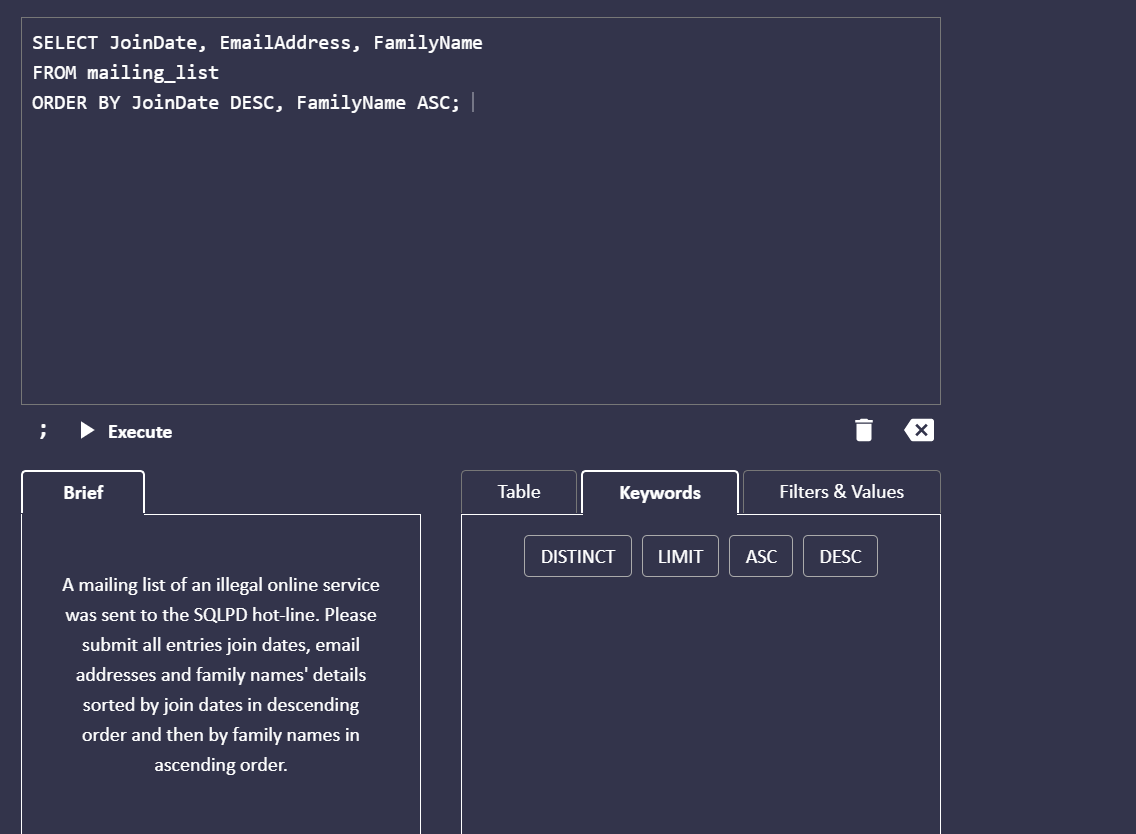


Esta sentencia devuelve datos no repetidos de las columnas “Username”, “SubscriptionDate” y “Purchases”, es decir, datos únicos de la combinación de las tres columnas. Estos datos se presentarán de manera ordenada descendentemente según la columna “Username”.

1. Mostrar datos ordenados por dos columnas distintas

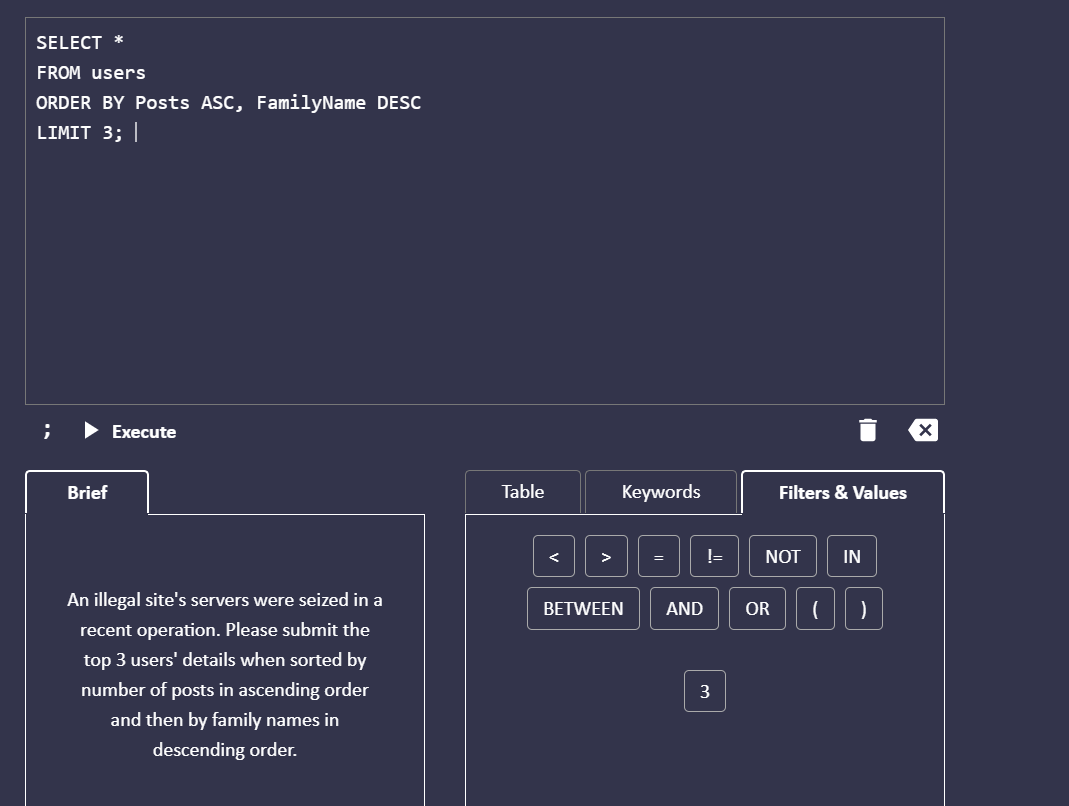


Sentencia SQL utilizada: **SELECT JoinDate, EmailAdress, FamilyName FROM mailing\_list ORDER BY JoinDate DESC, FamilyName ASC;**

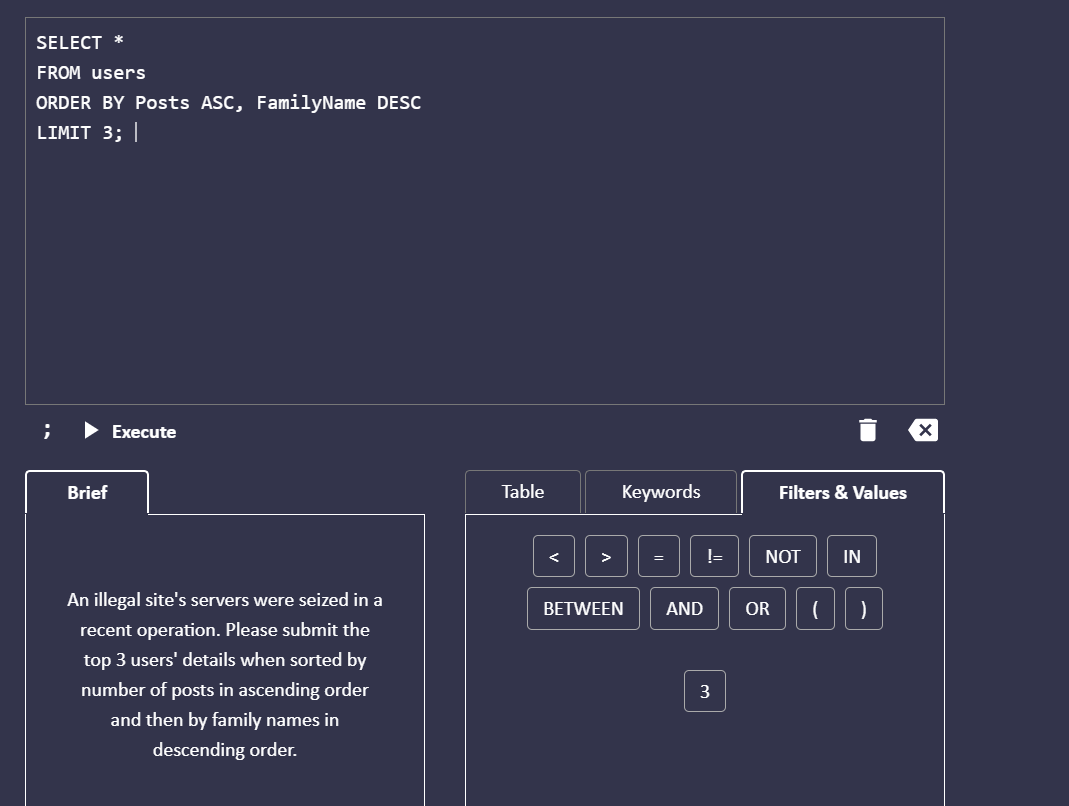


Esta sentencia devulve los registros de las tres columnas seleccionadas ordenadas primeramente por la columna “JoinDate” de forma descendente y con una segunda ordenación de la columna “FamilyName” ascendentemente.

1. Limitar registros

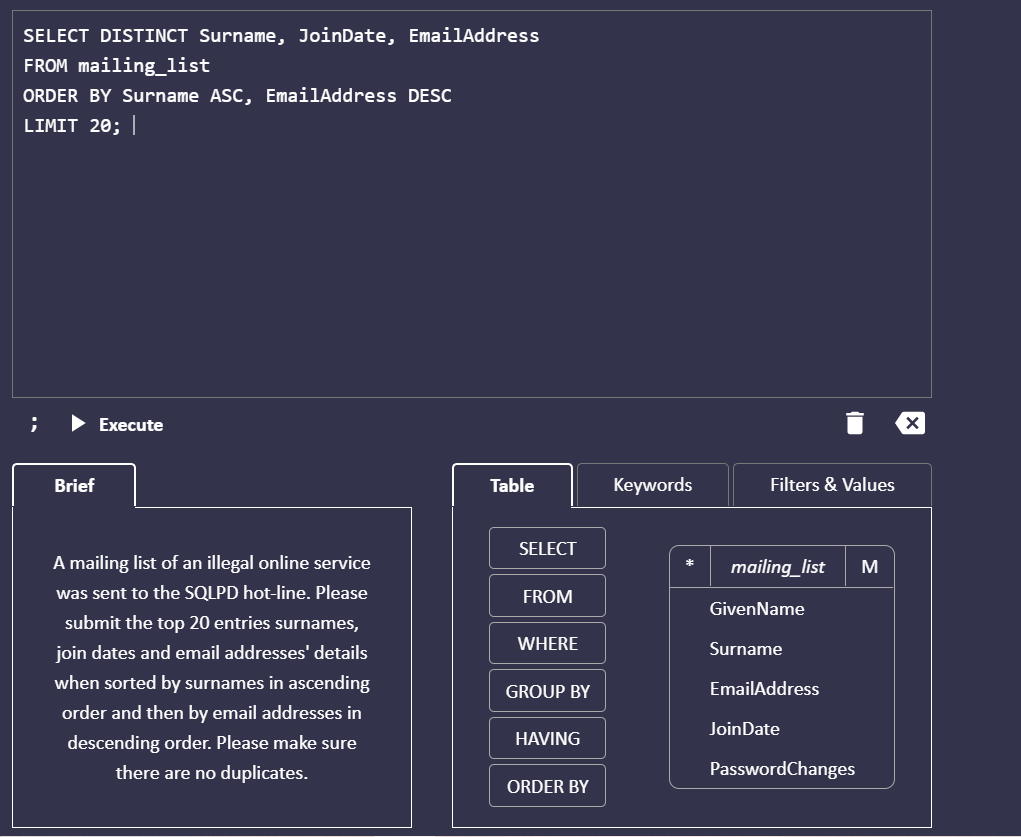


Sentencia SQL utilizada: **SELECT \* FROM users ORDER BY Posts ASC, FamilyName DESC LIMIT 3;**

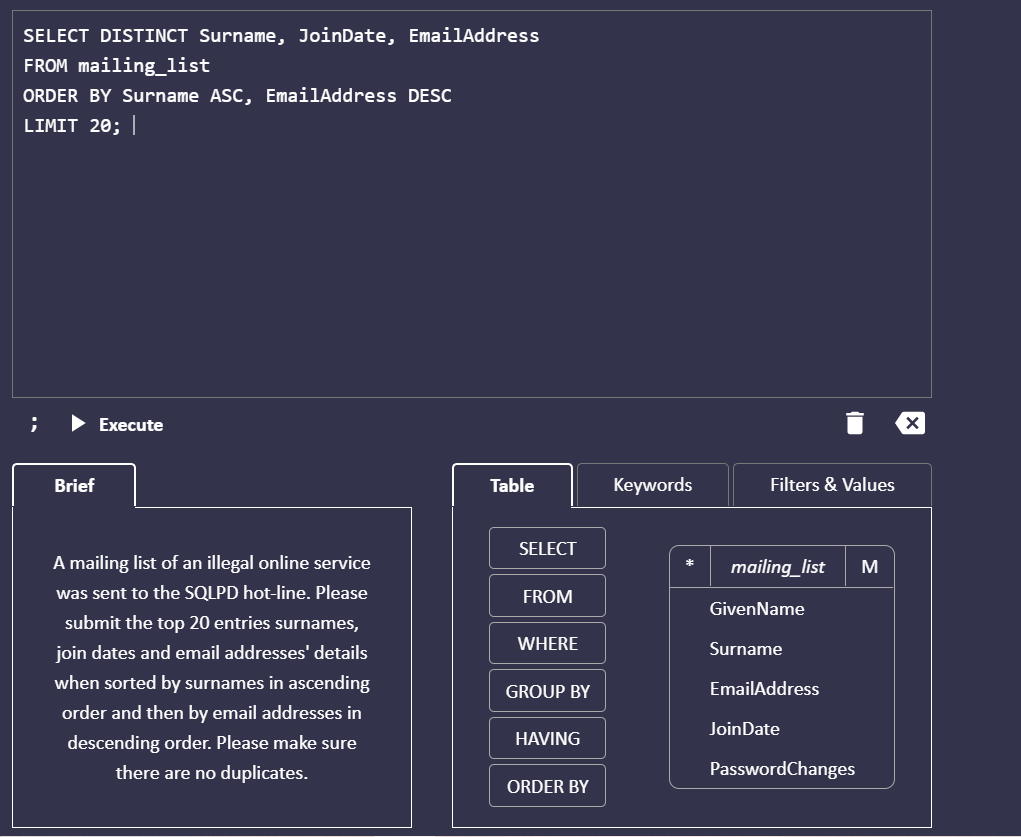


Esta sentencia devuelve todos los registros de la tabla ordenados por dos columnas y además limita el resultado en 3. Por ello, se mostrarán solamente los 3 primeros resultados que tengan el mayor número de “Posts”, con una segunda ordenación por “FamilyName” ordenada de manera descendente. La palabra clave que permite limitar registros es “LIMIT” y se coloca a final de la sentencia.

1. Mostrar resultados no duplicados, ordenados y limitados



Sentencia SQL utilizada: **SELECT DISTINCT Surname, JoinDate, EmailAddress FROM mailing\_lits ORDER BY Surname ASC, EmailAddress DESC LIMIT 20;**



Esta sentencia devuelve los 20 primeros registros no duplicados de las columnas especificadas ordenadas ascendentemente por la columna “Surname” y descendentemente por la columna “EmailAddress.

