SQL Injection

You join as a software developer at CatChat, primarily a English-French translation service for the word cat but also hosts a message board for cat-related themes. With your CPSC 526 knowledge, you react in horror when observing the following in their codebase:

```
getProfile(username, connection):
 query = "SELECT profile FROM Users WHERE username ='" + username + "' ;"
 statement = connection.createStatement()
 return statement.executeQuery(query)
```

1. Despite your protestations, the team responsible for this feature disagree with you that there is a security vulnerability. Craft a value for username that will result in the table Users being dropped (deleted) to prove yourself correct.

2. The team responsible for this security failure now agrees that this issue is serious. After hours of planning meetings to agree on when to have design meetings to discuss how they can fix it, they then have those meetings and they decide to fix the code by backslash-escaping all single quotes in username. (Assume their SQL engine uses backslashes to escape single quotes.)

   ```
   username = username.replace("'","\\'");
   ```

   This replaces all instances of ' with \'. The code is still vulnerable to SQL injection. Make appropriate modifications to your answer above, so that the table Users will still be deleted. Explain why you made the modifications.

3. After multiple days of nothing but meetings, the team attempts to solve this issue again with the following approach.

   ```
   username = username.replace("\\'","!!!!!");
   username = username.replace("'","\\'");
   username = username.replace("!!!!!","\\'");
   ```

   That is, it replaces \' with a temporary !!!!!, then escapes the remaining quotes, and then returns back the originally correctly escaped quotes. This code is still vulnerable to SQL injection. Make appropriate modifications to your answer above, so that the table Users will still be deleted. Explain why you made the modifications.

4. In frustration after three months of nothing but meetings, you head to your boss to explain that these one-off fixes aren't addressing the real problem. Explain what the real problem is (i.e., code versus data), and how best to fix it.