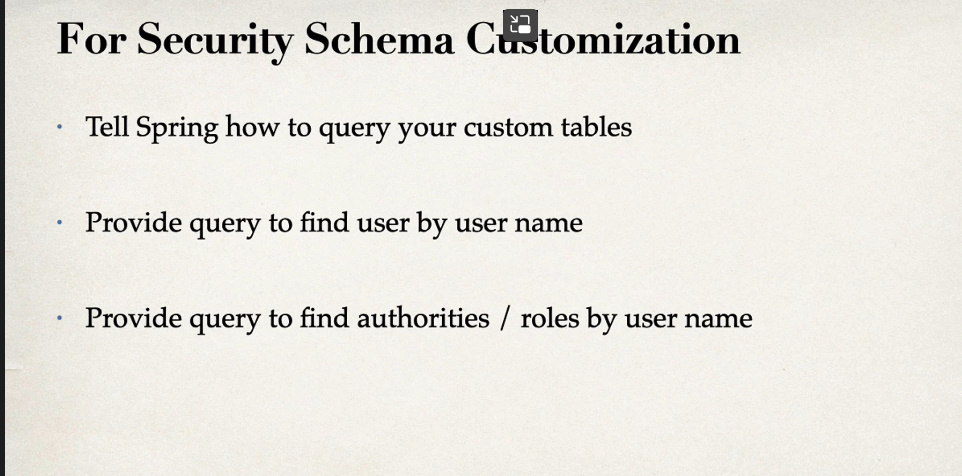
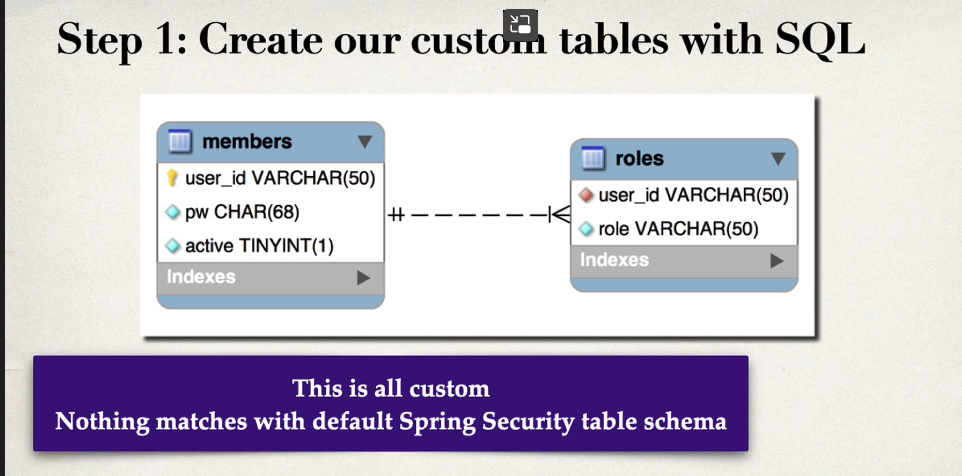
* Spring Security ne ajuta sa cream propria nostra schema pentru tabelele
* Tot ce trebuie sa facem este de a oferi query pentru a agasi userii dupa username si de a gasi rolurile dupa username



**Pasi de creare**

1. Cream tabele custom:



Folosim orice tabel name, orice attribute name, orice coloane, oricate vrem etc.

de ex:

CREATE TABLE custom\_users(

id varchar(50) PRIMARY KEY,

password varchar(68) NOT NULL,

active tinyint NOT NULL DEFAULT 1,

banned tinyint NOT NULL DEFAULT 0

);

CREATE TABLE roles(

user\_id varchar(50) NOT NULL REFERENCES custom\_users(id),

role\_id varchar(50) NOT NULL,

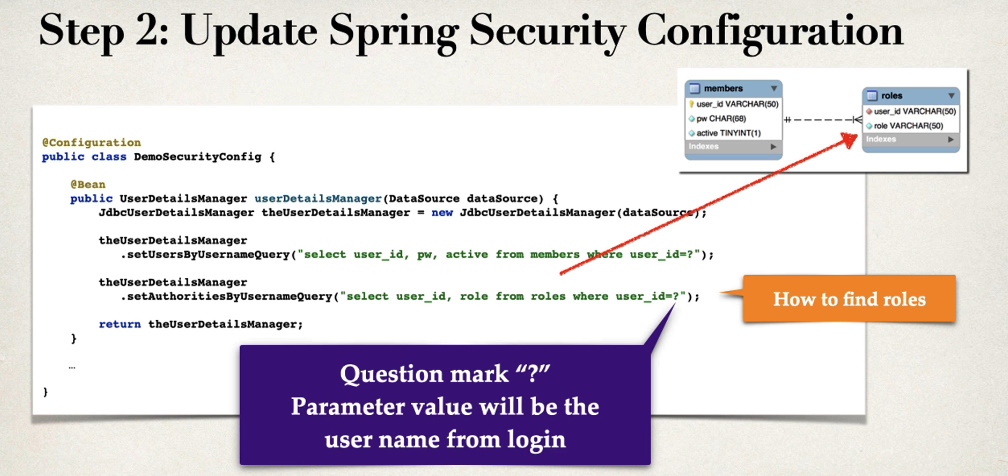
active tinyint DEFAULT 1,

UNIQUE KEY(user\_id,role\_id)

);

1. Modificam Spring Security UserDetailManager Bean

* Oferim un query pentru a gasi userii dupa username
* Oferim un query pentru a gasi rolurile dupa username



.setUsersByUsernameQuery() – gasim cele 3 coloane ce corespund la username, password si active, dar nu conteaza numele lor, doar sa fie in ordinea asta si sa corespunda ca tip de date cerut

.setAuthoritiesByUsernameQuery() – gasim cele 2 coloane ce corespund la username si la role

@Bean  
public JdbcUserDetailsManager jdbcUserDetailsManager(){  
 JdbcUserDetailsManager userDetailsManager = new JdbcUserDetailsManager(securityDataSource());  
 userDetailsManager.setUsersByUsernameQuery("SELECT id,password,active FROM custom\_users WHERE id=?");  
 userDetailsManager.setAuthoritiesByUsernameQuery("SELECT user\_id,role\_id FROM custom\_roles WHERE user\_id=?");  
  
 return userDetailsManager;  
}