

Zusammenfassung der Tabellenstruktur:

- **Tabelle `roles`:**
 - Enthält eine `enum` mit den möglichen Rollen: `ROLE_ADMIN`, `ROLE_MODERATOR`, `ROLE_USER`.
- **Tabelle `users`:**
 - Enthält die Felder `id`, `email`, `password`, `username`, wobei `email` und `username` als einzigartig (UNI) definiert sind.
- **Tabelle `user_roles`:**
 - Enthält die Verknüpfung von Benutzer-IDs (`user_id`) und Rollen-IDs (`role_id`), um die Beziehungen zwischen Benutzern und Rollen zu modellieren.

Nächste Schritte:

1. Füge Rollen in die `roles`-Tabelle ein

Da die `roles`-Tabelle leer ist, müssen wir die vordefinierten Rollen wie `ROLE_USER`, `ROLE_ADMIN`, etc. manuell hinzufügen, damit Benutzer eine Rolle zugewiesen bekommen können. Hier ist das SQL für das Hinzufügen von Rollen:

```
INSERT INTO roles (name) VALUES ('ROLE_USER');
INSERT INTO roles (name) VALUES ('ROLE_ADMIN');
INSERT INTO roles (name) VALUES ('ROLE_MODERATOR');
```

2. Überprüfe die Benutzerregistrierung

Nachdem die Rollen eingefügt wurden, solltest du sicherstellen, dass bei der Benutzerregistrierung auch eine Verknüpfung zwischen dem Benutzer und einer Rolle erstellt wird.

- Bei der Registrierung eines Benutzers (z. B. über `signup`) sollte der Benutzer automatisch einer Rolle zugewiesen werden (z. B. `ROLE_USER`).
- Die Verknüpfung des Benutzers mit einer Rolle erfolgt über die `user_roles`-Tabelle. Das bedeutet, dass nach dem Erstellen eines Benutzers in der `users`-Tabelle auch ein Eintrag in `user_roles` angelegt werden muss, um den Benutzer mit seiner Rolle zu verbinden.

3. Einfügen eines Benutzers und einer Rolle

Um sicherzustellen, dass alles funktioniert, kannst du einen Benutzer manuell einfügen, ihn einer Rolle zuweisen und prüfen, ob die Daten korrekt gespeichert werden.

Beispiel:

1. Füge einen Benutzer in die `users`-Tabelle ein:

```
INSERT INTO users (username, email, password)
VALUES ('newUser7', 'test@mail.de', 'password123');
```

Hinweis: Das Passwort sollte in der Realität mit einem Hashing-Algorithmus (wie BCrypt) gespeichert werden. Im Beispiel wird der Plaintext verwendet, was für Testzwecke akzeptabel ist, aber für die Produktion muss eine sichere Hashing-Methode verwendet werden.

2. Finde die `user_id` des Benutzers:

```
SELECT id FROM users WHERE username = 'newUser7';
```

Angenommen, die `user_id` des Benutzers ist 1, dann kannst du nun den Benutzer einer Rolle zuweisen.

3. Weise den Benutzer der Rolle `ROLE_USER` zu:

```
INSERT INTO user_roles (user_id, role_id)
VALUES (1, (SELECT id FROM roles WHERE name = 'ROLE_USER'));
```

4. Testen der API mit dem `signup`-Endpoint

Wenn du nun den `signup`-Endpoint aufrufst, sollte der Benutzer in die `users`-Tabelle eingefügt werden, und die Zuordnung zur Rolle `ROLE_USER` sollte in der `user_roles`-Tabelle vorgenommen werden.

- **Überprüfe den `signup`-Endpoint** und stelle sicher, dass er wie erwartet funktioniert.
- Wenn alles korrekt konfiguriert ist, solltest du nach der Registrierung eines Benutzers sehen können, dass dieser sowohl in der `users`-Tabelle als auch in der `user_roles`-Tabelle auftaucht.

Zusammenfassung der nächsten Schritte:

1. Füge manuell Rollen in die `roles`-Tabelle ein.
2. Registriere einen Benutzer und stelle sicher, dass er einer Rolle zugewiesen wird.
3. Überprüfe die API und teste den gesamten Flow der Benutzerregistrierung.
4. Wenn alles funktioniert, sollte die `users`-Tabelle den Benutzer enthalten und die `user_roles`-Tabelle die Zuordnung zur Rolle.