

## Übungsaufgabe 25 - OOP - Datenbank-Migrationsscript für Projekt Benutzerverwaltung

### 25.a) Migrationsscript für Tabelle `users` in Datenbank `user_mangement`

Erstelle ein PHP-Script names `migration.php` mit dem Du die Datenbank-Tabelle `users` in Datenbank `user_mangement` mittels PDO erstellst. Falls die Tabelle bereits vorhanden ist, soll diese zuvor gedropt werden.

Anmerkung: Du benötigst natürlich eine Datenbankverbindung! Hierfür kannst Du unser Script `database.inc.php` aus Aufgabe 21 verwenden, also einfach den Ordner `/inc` bzw. nur die Datei kopieren.

Erzeuge folgende Struktur:

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
firstname	varchar(40)	NO		NULL	
lastname	varchar(40)	NO		NULL	
email	varchar(80)	NO	UNI	NULL	
password	varchar(255)	NO		NULL	
role	varchar(8)	NO		inactive	
created_at	datetime	NO		current_timestamp()	
updated_at	timestamp	NO		current_timestamp()	on update current_timestamp()

wobei

- alle Spalten dürfen nicht `NULL` sein (vgl. WBT Datenbanken & SQL - Lektion 20.4)
- die Spalte `role` soll als Default den Wert "inactive" setzen
- die Spalte `created_at` soll als Default-Wert den `aktuellen Timestamp` setzen
- die Spalte `updated_at` soll mittels eines `ON UPDATE Constraints` so eingeschränkt werden, dass bei einem `UPDATE` eines Datensatzes automatisch der aktuelle Timestamp gesetzt wird

→ Teste Dein Script

## 25.b) Daten seeden

Erstelle ein weiteres Script namens `seeding.php`.

Kopiere Deine Lösung von Aufgabe 24.c und füge diese am besten per Copy&Paste in das Script ein. Da wir die API `fakerphp` benötigen, kannst Du diese entweder mit dem Composer neu installieren, oder Du kopierst dir die beiden JSON-Dateien `composer.json` und `composer.lock` aus Aufgabe 24 in das Stammverzeichnis der Aufgabe (also in das Hauptverzeichnis der Aufgabe und nicht in ein Unterverzeichnis) und führst folgenden Composer-Befehl in der Konsole aus

```
composer install
```

Passe nun den Code so an, dass Deine Funktion `generateUser()` die Werte (d. h. die key-value-Paare) für die Spalten `role`, `created_at` und `updated_at` nicht mehr erzeugt (hier brauchen wir keine Fake-Daten mehr, da wir mittels der Defaultwerte diese in MySQL setzen).

Erzeuge nun manuell 2 User (analog Aufgabe 24.c), deren Daten Du kennst und die Du später zum Testen in einer Applikation z. B. für ein Login nutzen kannst. Einer der User sollte die Rolle `admin` und der andere `user` bekommen. Speichere die beiden User in der Datenbanktabelle.

Erzeuge nun 25 weitere User (analog Aufgabe 24.c) und speichere diese ebenfalls in der Datenbanktabelle (nicht mehr in einer Datei!).

**Wichtig:** Passwörter dürfen niemals in Klartext in der Datenbank gespeichert werden  
⇒ hashe die Passwörter bevor Du die Daten persistierst!

→ Teste Dein Script

## 25.c) Migrieren und Seeden in einen Script

Erstelle ein weiteres Script namens `migrate_n_seed.php`, dass beide Teilaufgaben 25.a und 25.b in einem ausführt.

→ Teste Dein Script