

# InstaHub

## Tabelle User

Bereits vor deinem ersten Login in deinem InstaHub haben sich 196 User registriert!

Fügst du über einen SQL-Befehl einen neuen User hinzu, kann sich dieser nicht einloggen, da das Passwort gesichert in der Datenbank abgespeichert werden muss und nicht im Klartext verglichen werden kann.

Um Passwörter im Klartext zu hashen, kannst du in deinem Browser die folgende Adresse aufrufen:

<https://hub.instahub.org/dba/cryptPWs>

## Tabelle Photo

```
CREATE TABLE `photos` (  
  `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `user_id` int(10) UNSIGNED NOT NULL,  
  `description` varchar(255) NOT NULL,  
  `url` varchar(255) NOT NULL,  
  `created_at` timestamp NULL DEFAULT NULL,  
  `updated_at` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE  
)
```

1. Verstehe die obigen SQL-Befehl.
2. Führe die obigen SQL-Befehl nacheinander aus.
3. Um die Aktivität deiner Mitglieder zu beschleunigen, kannst du im Ordner *sql* den Datensatz *photos.sql* einfügen.

## Tabelle Tags

```
CREATE TABLE `tags` (  
  `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `photo_id` int(10) UNSIGNED NOT NULL,  
  `name` varchar(255) NOT NULL,  
  `created_at` timestamp NULL DEFAULT NULL,  
  `updated_at` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (id),  
  FOREIGN KEY (`photo_id`) REFERENCES `photos` (`id`) ON DELETE CASCADE  
)
```

1. Verstehe die obigen SQL-Befehl.
2. Führe die obigen SQL-Befehl nacheinander aus.
3. Um aus den bestehenden Bildbeschreibungen die Tags zu extrahieren kannst du folgende Webadresse aufrufen: <https://hub.instahub.org/dba/updateTags>

## Tabelle Comments

```
CREATE TABLE `comments` (  
  `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `user_id` int(10) UNSIGNED NOT NULL,  
  `photo_id` int(10) UNSIGNED NOT NULL,  
  `body` varchar(255) NOT NULL,  
  `created_at` timestamp NULL DEFAULT NULL,  
  `updated_at` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`photo_id`) REFERENCES `photos` (`id`) ON DELETE CASCADE,  
  FOREIGN KEY (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE  
)
```

1. Verstehe die obigen SQL-Befehl.
2. Führe die obigen SQL-Befehl nacheinander aus.
3. Um die Aktivität deiner Mitglieder zu beschleunigen, kannst du im Ordner *sql* den Datensatz *comments.sql* einfügen.

## Tabelle Follower

```
CREATE TABLE `follows` (  
  `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `following_id` int(10) UNSIGNED NOT NULL,  
  `follower_id` int(10) UNSIGNED NOT NULL,  
  `created_at` timestamp NULL DEFAULT NULL,  
  `updated_at` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`following_id`) REFERENCES `users`(`id`) ON DELETE CASCADE,  
  FOREIGN KEY (`follower_id`) REFERENCES `users`(`id`) ON DELETE CASCADE  
)
```

1. Verstehe die obigen SQL-Befehl.
2. Führe die obigen SQL-Befehl nacheinander aus.
3. Um die Aktivität deiner Mitglieder zu beschleunigen, kannst du im Ordner *sql* den Datensatz *follows.sql* einfügen.

## Tabelle Likes

```
CREATE TABLE `likes` (  
  `id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
  `photo_id` int(10) UNSIGNED NOT NULL,  
  `user_id` int(10) UNSIGNED NOT NULL,  
  `created_at` timestamp NULL DEFAULT NULL,  
  `updated_at` timestamp NULL DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  FOREIGN KEY (`photo_id`) REFERENCES `photos`(`id`) ON DELETE CASCADE,  
  FOREIGN KEY (`user_id`) REFERENCES `users`(`id`) ON DELETE CASCADE  
)
```

1. Verstehe die obigen SQL-Befehl.
2. Führe die obigen SQL-Befehl nacheinander aus.
3. Um die Aktivität deiner Mitglieder zu beschleunigen, kannst du im Ordner *sql* den Datensatz *likes.sql* einfügen.