

# Interaktive Anwendung und Sichten

## Inhaltsverzeichnis

Interaktive Anwendung und Sichten.....	1
1. Nutzung durch Schüler*innen.....	2
1.1. AG-Übersicht.....	2
1.2. Eigene AGs.....	2
1.3. Besuchte Veranstaltungen.....	3
2. Nutzung durch Lehrer*innen.....	4
2.1. Klassenlisten.....	4
2.2. AG anlegen.....	4
2.3. AG-Verwaltung.....	5
2.3.1. AG-Daten editieren.....	6
2.3.2. AG-Teilnehmer festlegen.....	6
2.3.3. AG-Termin anlegen und Anwesenheit notieren.....	8
2.3.4. AG-Leiter festlegen.....	10
3. Nutzung durch Jedermann.....	12
3.1. Eigenes Profil editieren.....	12
4. Nutzung durch den Administrator.....	13
4.1. Benutzersuche.....	13
4.2. Benutzerdaten editieren.....	14
4.3. Benutzer hinzufügen.....	15

Bei der zugrundeliegenden Miniwelt handelt es sich um die Verwaltung von Arbeitsgemeinschaften im schulischen Kontext. Die Sichten und die interaktive Anwendung sollen genau dies ermöglichen. Dabei sind verschiedene Nutzungsszenarien relevant, die entsprechend angepasste Sichten erfordern:

## 1. Nutzung durch Schüler\*innen

### 1.1. AG-Übersicht

Schüler\*innen sollen sich über die angebotenen Arbeitsgemeinschaften (wgs) einen Überblick verschaffen können. Daher ist es wichtig, dass AG-Titel, Namen der AG-Leiter, Wochentag, Uhrzeit, Dauer, maximale Teilnehmerzahl, Fortsetzbarkeit und eine kurze Beschreibung angezeigt werden.

```
SELECT * FROM wgs
WHERE schoolyear = '2020/21'
ORDER BY day, time, title ASC
```

Alle Arbeitsgemeinschaften auf einen Blick						
Ein Klick auf einen AG-Titel öffnet das zugehörige Infofeld. Ein Klick auf das geöffnete Infofeld schließt dieses wieder. Diese Tabelle ist rein informativ. Es können keine Plätze reserviert werden. Nähere Infos zur Teilnahme erhält man vom jeweiligen AG-Leiter.						
Titel	Leiter	Wochentag	Uhrzeit	Dauer	Maximum	fortsetzbar
Nähen	Adler, Macker	Montag	15:00:00	90 min	10	nein
Wir nähen alle einen Teddybären.						
Schach	Engel, Usus	Montag	15:00:00	90 min	10	ja
Klassisches Schachspiel für jung und alt. Beginn: 29. März 2021						
Karate	Edison	Mittwoch	15:30:00	90 min	keins	ja
Basketball	Brot	Mittwoch	16:00:00	90 min	20	ja
Technik	Jensen	Freitag	14:00:00	90 min	6	ja

(wg\_list.php)

### 1.2. Eigene AGs

Schüler\*innen sollten sich einen Überblick verschaffen können hinsichtlich der AGs, welche sie selbst besuchen. Dieselben Infos wie bei der Kompletliste sind relevant.

```
SELECT * FROM wgs,participate AS p
WHERE p.user_id = :id AND p.wg_id = wgs.wg_id
ORDER BY p.schoolyear DESC, wgs.title ASC
```

Dabei wird :id durch die aktuelle User-Id ersetzt.

## Meine Arbeitsgemeinschaften

Klicke auf den Titel einer AG, um die AG-Beschreibung zu sehen.

Klicke auf das Schuljahresdatum einer AG, um die Liste deiner Teilnahmen zu sehen.

Schuljahr	Titel	Leiter	Wochentag	Uhrzeit	Dauer	Maximum	fortsetzbar
2020/21	Basketball	Brot	Mittwoch	16:00:00	90 min	20	ja
2020/21	Nähen	Adler, Macker	Montag	15:00:00	90 min	10	nein
2020/21	Technik	Jensen	Freitag	14:00:00	90 min	6	ja
2019/20	Schach	Engel, Usus	Montag	15:00:00	90 min	10	ja

(wg\_my.php)

### 1.3. Besuchte Veranstaltungen

Für Schüler\*innen sind ihre besuchten AG-Veranstaltungen (events) eventuell von Interesse, wenn Sie z.B. ihren Eltern zeigen möchten, was sie in den letzten Monaten dort getan haben. Datum, Uhrzeit, Dauer und eine kurze inhaltliche Beschreibung reichen hierfür aus.

```
SELECT date, time, duration, annotation FROM events AS e, present AS p
WHERE e.wg_id = :wid AND p.user_id = :uid AND e.event_id = p.event_id
ORDER BY e.date DESC, e.time DESC
```

Dabei werden :uid durch die aktuelle User-Id und :wid durch die AG-Id ersetzt.

## Meine Arbeitsgemeinschaften

[Zurück](#)

### Basketball (Terminliste)

Datum	Uhrzeit	Dauer	Bemerkung
2021-05-01	16:00:00	90	Feiertags-AG
2021-04-21	16:00:00	90	Fairplay
2021-04-14	16:00:00	90	Taktik 2
2021-04-07	16:00:00	90	Taktik 1
2021-03-31	16:00:00	90	Wurftechniken

(wg\_my.php)

## 2. Nutzung durch Lehrer\*innen

### 2.1. Klassenlisten

Klassenlehrer\*innen können sich durch Auswahl ihrer Klasse eine Sicht aller AG-Teilnahmen ihrer Schüler\*innen ausgeben lassen. Dies ist praktisch für das Ausfüllen der Zeugnisvermerke hinsichtlich der AG-Teilnahmen. Ausgegeben werden die Namen der Schüler\*innen, die AG-Titel, die Namen der AG-Leiter und die Anzahl der Teilnahmen.

```
SELECT u.firstname, u.lastname, w.title, w.wg_id, COUNT(e.event_id) AS eventcount
FROM students AS s, users AS u, wgs AS w, present AS p, events AS e
WHERE s.class=:cl AND e.date>=:da AND u.user_id=s.user_id AND s.user_id=p.user_id
AND p.event_id=e.event_id AND e.wg_id=w.wg_id
GROUP BY u.user_id, w.wg_id
ORDER BY lastname ASC, firstname ASC, title ASC
```

Dabei werden :cl durch die gewählte Klasse und :da durch das Datum des ersten Schultages ersetzt.

Klassenlisten				
Klassenauswahl				
7a				
Nachname	Vorname	AG	Teilnahmen	AG-Leiter
Bähr	Herbert	Karate	1	Edison
Bähr	Herbert	Technik	1	Jensen
Cham	Wisaam Aram	Technik	1	Jensen
Decker	Doreen	Basketball	1	Brot
Decker	Doreen	Karate	1	Edison
Decker	Doreen	Technik	1	Jensen
Harb	Habudi	Nähen	1	Adler, Macker
Harb	Hamudi	Karate	1	Edison

(classlist.php)

### 2.2. AG anlegen

Die Erfassung neuer AGs muss durch Lehrer selbst möglich sein. Dafür muss eine Sicht zur Verfügung gestellt werden, die es ermöglicht, sämtliche AG-Daten einzugeben, welche anschließend in die Tabelle wgs eingefügt werden. Gleichzeitig muss der/die Lehrer\*in zur AG-Leitungstabelle lead hinzugefügt werden. Beide Insert-Befehle werden in Form einer PDO-Transaktion ausgeführt. Gibt es irgendeinen Fehler, wird ein Rollback durchgeführt.

```
INSERT INTO wgs(title,day,time,duration,max_num,multiple,schoolyear,description)
VALUES (:title,:day,:time,:dur,:max,:mul,:year,:des)

INSERT INTO lead (user_id, wg_id) VALUES (:uid,(SELECT currval('wgs_wg_id_seq')))
```

Da die neue wg\_id vom Datenbanksystem automatisch vergeben wird und somit vorher noch unbekannt ist, muss sie nach dem ersten Insert abgefragt werden. Dies geschieht mittels SELECT currval('wgs\_wg\_id\_seq').

### Arbeitsgemeinschaft erstellen

<b>Titel der Arbeitsgemeinschaft</b> <input type="text" value="AG-Titel eingeben..."/>	<b>Wochentag</b> <input type="text" value="Montag oder Dienstag oder ..."/>
<b>Uhrzeit</b> <input type="text" value="15:20"/>	<b>Dauer in Minuten</b> <input type="text" value="90"/>
<b>Maximale Teilnehmeranzahl (0 = beliebig)</b> <input type="text" value="0"/>	<b>Schuljahr</b> <input type="text" value="2020/21"/>
<b>Fortsetzbar?</b> <input checked="" type="checkbox"/>	
<b>Beschreibung</b> <div style="border: 1px solid #ccc; height: 40px; width: 100%;"></div>	
<input type="button" value="speichern"/>	

(wg\_add.php)

## 2.3. AG-Verwaltung



Den AG-Leiter\*innen muss es möglich sein, Änderungen an den AG-Daten vorzunehmen (Tabelle: wgs), Teilnehmer\*innen hinzuzufügen oder zu entfernen (Tabelle: participate), die Anwesenheit bei AG-Terminen zu dokumentieren (Tabellen: events und present) und weitere AG-Leiter festzulegen. Hierfür wird zunächst eine Sicht erzeugt, die alle AGs anzeigt, welche der/die Lehrer\*in leitet. Titel, Wochentag, Uhrzeit und Schuljahr reichen hierfür aus.

```
SELECT title, day, time, schoolyear, wgs.wg_id FROM wgs, lead
WHERE lead.user_id = :uid AND wgs.wg_id = lead.wg_id
ORDER BY schoolyear DESC, title ASC, day ASC, time ASC
```

### Eigene AGs verwalten

Bitte eine AG zum Bearbeiten auswählen.

Sie können Teilnehmer ein- und austragen (✎), die Anwesenheit notieren (✓) und weitere AG-Leiter festlegen (🧠).

Titel	Wochentag	Uhrzeit	Schuljahr			
Nähen	Montag	15:00:00	2020/21		<input checked="" type="checkbox"/>	

(wg\_edit.php)

Einzelne Symbole zum Anklicken ermöglichen die oben geschilderten Funktionalitäten.

### 2.3.1.AG-Daten editieren

Zunächst werden die Daten zu der AG mit der Id :wid abgefragt und angezeigt. Durch einen Klick auf den Speichern-Button werden sie anschließend in der Datenbank gesichert.

```
SELECT * FROM wgs WHERE wg_id=:wid
```

```
UPDATE wgs SET title=:title, day=:day, time=:time, duration=:dur, max_num=:max,  
multiple=:mul, schoolyear=:year, description=:des  
WHERE wg_id=:wid
```

**Eigene AGs verwalten**

Titel

Nähen

Wochentag

Montag

Uhrzeit

15:00:00

Dauer in Minuten

90

Maximale Teilnehmeranzahl (0 = beliebig)

10

Schuljahr

2020/21

Fortsetzbar?

☐

Beschreibung

Wir nähen alle einen Teddybären.

speichern

abbrechen

(wg\_edit.php & wg\_edit\_ajax.php & wg\_edit\_save.php)

### 2.3.2.AG-Teilnehmer festlegen

Damit eine Person der AG hinzugefügt werden kann, muss zunächst einmal erkennbar sein, welche Personen bereits in der AG sind. Zur besseren Unterscheidung von Lehrern und Schülern werden die Infos aus den entsprechenden Tabellen (students / teachers) ebenfalls abgefragt und mittels NATURAL LEFT JOIN angefügt.

```
SELECT * FROM (  
    SELECT user_id, firstname, lastname, email  
    FROM users NATURAL JOIN participate p  
    WHERE p.wg_id = :wid AND p.schoolyear='$schoolyear') AS one  
NATURAL LEFT JOIN students  
NATURAL LEFT JOIN teachers  
ORDER BY lastname, firstname, class, shorthand
```

Wie viele Plätze noch frei sind, ist ebenfalls wichtig:

```
SELECT count(user_id) AS num
FROM participate
WHERE wg_id = :wid AND schoolyear = :sy
```

**Eigene AGs verwalten**

**Nähen - Teilnehmer einschreiben und entfernen**

Es können noch 4 Teilnehmer hinzugefügt werden.

Teilnehmerauswahl

neuen Teilnehmer auswählen

Zur AG hinzufügen

abbrechen

**Eingeschriebene Personen:**

Nachname	Vorname	Klasse/Kürzel	E-Mail	entfernen
Abt	Stephan	8b		×
Flanell	Fidora	8a		×
Harb	Habudi	7a	habudiharb@ag-manager.de	×
Jabłoński	Jakub	7b	jakubjablonski@ag-manager.de	×
Nadel	Nina	7b		×
Usus	Uschi	7a		×

(wg\_edit.php & wg\_participate.php)

Durch ein einfaches Klicken auf ein x ist es möglich, Schüler\*innen leicht aus der AG zu entfernen:

```
DELETE FROM participate WHERE user_id=:uid AND wg_id=:wid AND schoolyear=:sy
```

Für ein einfaches Hinzufügen von Personen steht eine Auswahlliste bereit, welche alle Personen enthält, die nicht bereits in die AG eingeschrieben wurden:

```
SELECT * FROM (
    SELECT user_id, firstname, lastname
    FROM users
    EXCEPT (
        SELECT u.user_id, firstname, lastname
        FROM users u, participate p
        WHERE p.wg_id = :wid AND u.user_id=p.user_id)) AS noparticipant
NATURAL LEFT JOIN students
NATURAL LEFT JOIN teachers
ORDER BY lastname, firstname, class, shorthand
```

Wurde eine Person ausgewählt, kann sie nun zur AG hinzugefügt werden:

```
INSERT INTO participate (user_id, wg_id, schoolyear) VALUES (:uid, :wid, :sy)
```

### 2.3.3.AG-Termin anlegen und Anwesenheit notieren

Für eine einfache Dokumentation der AG-Inhalte und der Anwesenheit existiert ein View, der Datum, Zeit, Dauer sowie ein Anmerkungsfeld zur Verfügung stellt. Beim Speichern werden diese Daten in die Tabelle events eingetragen.

```
INSERT INTO events (time, date, duration, annotation, wg_id)
VALUES (:ti, :da, :du, :an, :id)
```

Ebenso kann man durch das Setzen von Häkchen auf einfache Weise die Anwesenheit der Schüler\*innen dokumentieren.

Abfrage der AG-Teilnehmer für das Bereitstellen der Häkchen:

```
SELECT * FROM
    (SELECT user_id, firstname, lastname FROM participate p NATURAL JOIN users
     WHERE p.wg_id=:wid AND p.schoolyear=:sy) AS one
NATURAL LEFT JOIN (SELECT students.user_id, class FROM students) AS s
NATURAL LEFT JOIN teachers
ORDER BY lastname, firstname, class
```

Speichern der Anwesenheit:

```
INSERT INTO present (user_id, event_id)
VALUES (u1,e1), (u2,e2), ...
```

Die einzufügenden Wertepaare erzeugt eine PHP-Schleife durch das systematische Abfragen der Kästchen, ob diese abgehakt wurden oder nicht. Da das Einfügen des Termins in die Tabelle events und das Eintragen der Teilnehmer in die Tabelle present voneinander abhängig sind, werden diese in Form einer Transaktion realisiert. Auf diese Weise bleibt die Datenbank konsistent, falls ein Fehler in einem der beiden INSERT-Befehle auftritt.

Um vorausgegangene Termine im Blick zu haben und diese ggf. noch einmal zu ändern oder zu ergänzen, wird eine entsprechende Tabelle angezeigt.

```
SELECT * FROM events WHERE wg_id = :wid ORDER BY date DESC, time ASC
```



Eigene AGs verwalten

Nähen - einen neuen AG-Termin anlegen

Datum

15.05.2021

Zeit

15:00:00

Dauer in Minuten

90

Anmerkung

☐ Stephan Abt (8b)
☐ Fidora Flanell (8a)
☐ Habudi Harb (7a)
☐ Jakub Jabłoński (7b)
☐ Nina Nadel (7b)
☐ Uschi Usus (7a)

speichern

abbrechen

Bisherige Termine:

Datum	Uhrzeit	Dauer	Anmerkung
2021-05-03	15:00:00	90	Datum in der Zukunft
2021-04-05	15:00:00	60	Stoff ausschneiden
2021-03-28	15:00:00	90	Schnittmuster vorbereiten
2021-03-22	15:00:00	90	Gemeinsames Kennenlernen

(wg\_edit.php & wg\_presence\_new.php)

Ein Klick auf einen alten Termin fragt dessen Daten ab.

```
SELECT * FROM events WHERE event_id = :eid
```

Außerdem werden die damals anwesenden Teilnehmer als Checkboxes angezeigt. Da zwischenzeitlich ein Teilnehmer aus der AG ausgetreten sein kann, gestaltet sich die Abfrage etwas schwieriger:

```
SELECT * FROM (
    SELECT user_id, firstname, lastname FROM present pr NATURAL JOIN users
    WHERE pr.event_id=:eid
    UNION
    SELECT user_id, firstname, lastname FROM participate pa NATURAL JOIN users
    WHERE pa.wg_id=:wid AND pa.schoolyear=:sy) as one
NATURAL LEFT JOIN (
    SELECT user_id, 'y' AS waspresent FROM present WHERE event_id=:eid) AS two
NATURAL LEFT JOIN (
    SELECT students.user_id, class FROM students) AS three
NATURAL LEFT JOIN teachers
ORDER BY lastname, firstname, class
```

**Eigene AGs verwalten**

**Nähen - einen AG-Termin bearbeiten**

Datum:  Zeit:  Dauer in Minuten:

Anmerkung:

☒ Stephan Abt (8b)
 ☒ Fidora Flanell (8a)
 ☒ Habudi Harb (7a)
 ☒ Jakub Jabłoński (7b)
 ☒ Nina Nadel (7b)
 ☒ Uschi Usus (7a)

**Bisherige Termine:**

Datum	Uhrzeit	Dauer	Anmerkung
2021-05-03	15:00:00	90	Datum in der Zukunft
2021-04-05	15:00:00	60	Stoff ausschneiden
2021-03-28	15:00:00	90	Schnittmuster vorbereiten
2021-03-22	15:00:00	90	Gemeinsames Kennenlernen

(wg\_edit.php & wg\_presence\_edit.php):

### 2.3.4.AG-Leiter festlegen

Da es durchaus AGs gibt, welche von mehreren Lehrer\*innen oder auch von Schülern selbst geleitet werden, ist es sinnvoll, weitere AG-Leiter festlegen zu können. Ein AG-Leiter, der selbst kein Schüler ist, kann diese Sicht aufrufen. Da Schüler\*innen noch minderjährig sind, wird ihnen diese Funktion verwehrt.

Zunächst einmal wird eine Liste aller bereits existierenden AG-Leiter angezeigt:

```
SELECT * FROM (
    SELECT user_id, firstname, lastname FROM users NATURAL JOIN lead l
    WHERE l.wg_id = :wid
    EXCEPT
    SELECT user_id, firstname, lastname FROM users WHERE user_id = :uid) AS one
NATURAL LEFT JOIN students
NATURAL LEFT JOIN teachers
ORDER BY lastname, firstname, class, shorthand
```

Durch einen Klick auf ein x hinter dem Namen und dem Kürzel, kann ein AG-Leiter euch leicht wieder entfernt werden:

```
DELETE FROM lead WHERE user_id=:uid AND wg_id=:wid
```

Damit man einen neuen AG-Leiter leicht festlegen kann, werden alle Personen abgefragt, die noch kein Leiter der AG sind und in eine Auswahlliste eingefügt:

```
SELECT * FROM
  (SELECT user_id, firstname, lastname FROM users
  EXCEPT
  SELECT u.user_id, firstname, lastname FROM users u, lead l
  WHERE l.wg_id = :wid AND u.user_id=l.user_id) AS noleader
NATURAL LEFT JOIN (SELECT user_id, class FROM students) AS s
NATURAL LEFT JOIN (SELECT user_id, shorthand FROM teachers) AS t
ORDER BY lastname, firstname, class, shorthand
```

Hat man einen neuen AG-Leiter ausgewählt, wird dieser durch Klick auf den grünen Button in der Datenbank gesichert:

```
INSERT INTO lead (user_id, wg_id) VALUES (:uid, :wid)
```

**Eigene AGs verwalten**

**Nähen - AG-Leiter festlegen**

Personenauswahl

neuen AG-Leiter auswählen

als AG-Leiter festlegen

abbrechen

Sie selbst werden in der folgenden Tabelle nicht angezeigt.

**Existierende AG-LeiterInnen:**

Nachname	Vorname	Klasse/Kürzel	entfernen
Adler	Adam	ADL	×

(wg\_edit.php & wg\_leaders.php)

## 3. Nutzung durch Jedermann

### 3.1. Eigenes Profil editieren

Damit sich die Benutzer leicht zu merkende Benutzernamen aussuchen, Schreibfehler korrigieren, ein neues Passwort aussuchen oder bei Bedarf ihre E-Mailadresse hinterlegen können, ist es sinnvoll, eine Möglichkeit zu schaffen, die eigenen Daten editieren zu können.

Daten aus der Datenbank auslesen:

```
SELECT * FROM users
```

Daten in der Datenbank aktualisieren:

```
UPDATE users SET username = :un, password = :pw, firstname = :fn, lastname = :ln, email = :em
```

**Ihr Profil:**

Sie sind als Lehrer registriert.

Benutzername	Passwort	Passwort (Wiederholung)
<input type="text" value="regenhimmel"/>	<input type="password" value="*****"/>	<input type="password" value="Passwort..."/>
Vorname	Nachname	E-Mail
<input type="text" value="Moritz"/>	<input type="text" value="Macker"/>	<input type="text" value="moritzmacker@ag-manager.de"/>

Speichern

(profile.php)

## 4. Nutzung durch den Administrator

Eine Software, welche von einer Vielzahl von Benutzern eingesetzt wird und deren Daten zentral verwaltet benötigt einen Administrator. Schnell ist einmal ein Passwort vergessen, so dass ein Nutzer keinen Zugriff mehr auf seine Daten hat. Oder ein neuer Kollege muss in das System eingepflegt werden. Hierfür sind mehrere Funktionalitäten vonnöten.

### 4.1. Benutzersuche

Müssen Änderungen an den Daten eines Benutzers vorgenommen werden, so müssen diese erst einmal gefunden werden. Hierfür ist eine Suchfunktion unerlässlich. Eine Suche nach dem Benutzernamen, dem Vornamen oder dem Nachnamen ist ausreichend.

```
SELECT * FROM users
```

```
WHERE username LIKE :user AND firstname LIKE :fn AND lastname LIKE :ln
```

```
ORDER BY lastname ASC, firstname ASC, username ASC
```

**Verwaltungsbereich für Administratoren**

Wildcards:

- beliebig viele beliebige Zeichen: %
- ein beliebiges Zeichen: \_

Benutzer

Vorname

Nachname

%

%

%

suchen

(user\_search.php)

Die Ausgabe erfolgt in Form einer einfachen Tabelle, welche bis auf den Passworthash alle Daten aus der Tabelle "users" enthält.

Benutzer

Vorname

Nachname

%

%

N%

suchen

ID	Benutzer	Vorname	Nachname	E-Mail	Rolle	aktiv	Registrierungsdatum	Letztes Update
<a href="#">5</a>	schneeflocke	Nina	Nadel		user	ja	2021-04-27 17:42:40.364454	2021-04-27 17:42:40.364454
<a href="#">26</a>	tonnhi	Tôn Ngoc	Nhi	tonnhi@ag-manager.de	user	ja	2021-04-27 18:07:49.996247	2021-04-27 18:07:49.996247

(user\_search.php)

Durch einen Klick auf die ID ist es möglich, die Daten des Nutzers zu editieren.

## 4.2. Benutzerdaten editieren

Neben den Daten in der Tabelle "users" sind bei Lehrern auch das Kürzel und bei Schülern die Schülernummer und die Klasse relevant. Daher werden diese in diesen View mit einbezogen.

Abruf der Daten:

```
SELECT * FROM users WHERE (user_id = :id)
```

```
SELECT * FROM students WHERE (user_id = :id)
```

```
SELECT * FROM teachers WHERE (user_id = :id)
```

Änderung der Daten bei leerem Passwortfeld:

```
UPDATE users SET username = :na, enabled = :en, firstname = :fn, lastname = :ln, email = :email,  
roll = :roll WHERE user_id = :id
```

Änderung der Daten bei ausgefülltem Passwortfeld:

```
UPDATE users SET username = :na, password = :pwd, enabled = :en, firstname = :fn, lastname  
= :ln, email = :email, roll = :roll WHERE user_id = :id
```

Änderung der Daten in den Tabellen "teachers" und "students":

```
UPDATE teachers SET shorthand = :sh WHERE user_id = :uid
```

oder

```
INSERT INTO teachers (user_id, shorthand) VALUES (:id, :sh)
```

```
DELETE FROM students WHERE user_id = :uid
```

```
UPDATE students SET class = :cl, studentnumber = :sn WHERE user_id = :uid
```

oder

```
INSERT INTO students (user_id, class, studentnumber) VALUES (:id, :cl, :sn)
```

```
DELETE FROM teachers WHERE user_id = :uid
```

Für den Fall, dass beispielsweise ein Lehrer versehentlich als Schüler eingetragen wurde, und dies nun korrigiert wird, ist kein "UPDATE" auszuführen, sondern die obige Kombination aus "INSERT" und "DELETE".

**Benutzerdaten editieren**

Benutzername schneeflocke	Passwort Passwort...	Passwort (Wiederholung) Passwort...	Vorname Nina
Nachname Nadel	E-Mail mailadresse@irgendwo.de	Rolle user	Konto aktiv <input checked="" type="checkbox"/>

☐ Lehrerin 
 ☒ SchülerIn 
 ☐ Andere(r)

Klasse  
7b

Schülernummer  
nina01

(user\_edit.php nutzt account\_class.php)

### 4.3. Benutzer hinzufügen

Für das Hinzufügen eines neuen Benutzers wird eine leere Eingabemaske bereitgestellt, die eine Eingabe aller Benutzerinfos ermöglicht.

```
INSERT INTO users (username, password, firstname, lastname, email, roll, enabled)
VALUES (:na, :pw, :fn, :ln, :em, :ro, :en)
```

ggf.

```
INSERT INTO teachers (user_id, shorthand) VALUES (:id, :sh)
```

oder

```
INSERT INTO students (user_id, class, studentnumber) VALUES (:id, :cl, :sn)
```

**Benutzer hinzufügen**

Benutzername Benutzername...	Passwort Passwort...	Vorname Vorname...
Nachname Nachname...	E-Mail mailadresse@irgendwo.de	Rolle admin, editor, viewer oder user

☒ Lehrerin 
 ☐ SchülerIn 
 ☐ Andere(r)

Konto aktiv  
☒

Lehrerkürzel  
Lehrerkürzel...

(user\_add.php nutzt account\_class.php)