

# Anleitung Anpassung der Datenbank

20.12.2016

In den folgenden Kapitel wird beschrieben, wie die zu Beginn erhaltene Datenbank angepasst wurde. Es handelt sich dabei um eine Schritt für Schritt Anleitung, mit der die bestehende Datenbank des cnlab-Servers auf die neue Datenbankstruktur umgestellt werden kann.

## 1 Fragen mit mehr als sechs Antwortmöglichkeiten

Da neu die Restriktion gilt, dass maximal fünf Antwortmöglichkeiten erstellt werden können, muss die bestehende Datenbank so angepasst werden, dass keine Fragen mehr mit mehr als fünf Antwortmöglichkeiten bestehen. Mithilfe von phpMyAdmin können die entsprechenden Fragen herausgesucht werden. Bei der übergebenen Mobile Quiz - Datenbank zu Semesterbeginn waren dies die Fragen mit der ID 723, 845, 873 und 1468. Für diese sind die entsprechenden DELETE-Befehle schon unten aufgeführt. Sollten in der Zwischenzeit weitere dazugekommen sein, so sind die Befehle von Hand nachzutragen.

1. Login mit phpMyAdmin
2. Heraussuchen der Fragen mit mehr als 5 Antwortmöglichkeiten

```
SELECT question_id, COUNT(*) count FROM answer_question  
GROUP BY question_id HAVING COUNT(*) > 5;
```

Dabei wurden zusätzlich noch doppelte Fragen gefunden. Folgende Fragen war mehrfach vorhanden (inkl. gleicher Antworten), es wurde jeweils 1 Variante behalten.

- Welchen Teil des Schlüssels muss der Sender einer Nachricht bei der Verschlüsselung mit einem Public Key Verfahren verwenden? (gelöscht ID 876, 948, 915 und 877. behalten ID 875)
- Welche Felder eines Frames verändert ein Switch? (gelöscht ID 1451. behalten ID 1468)

3. Löschung der Beziehung von Quiz und Frage, qunaire\_qu

```
DELETE FROM `qunaire_qu` WHERE question_id = 876;
DELETE FROM `qunaire_qu` WHERE question_id = 948;
DELETE FROM `qunaire_qu` WHERE question_id = 915;
DELETE FROM `qunaire_qu` WHERE question_id = 877;
DELETE FROM `qunaire_qu` WHERE question_id = 1451;
```

#### 4. Löschung der Beziehung von Antwort und Frage, answer\_question

```
DELETE FROM `answer_question` WHERE `answer_question`.`answer_id`
= 2728 AND `answer_question`.`question_id` = 723;
```

```
DELETE FROM `answer_question` WHERE `answer_question`.`answer_id`
= 3188 AND `answer_question`.`question_id` = 845;
```

```
DELETE FROM `answer_question` WHERE `answer_question`.`answer_id`
= 3300 AND `answer_question`.`question_id` = 873;
```

```
DELETE FROM `answer_question` WHERE question_id = 876;
DELETE FROM `answer_question` WHERE question_id = 948;
DELETE FROM `answer_question` WHERE question_id = 915;
DELETE FROM `answer_question` WHERE question_id = 877;
DELETE FROM `answer_question` WHERE question_id = 1451;
```

```
DELETE FROM `answer_question` WHERE `answer_question`.`answer_id`
= 5598 AND `answer_question`.`question_id` = 1468;
```

#### 5. Löschung der Beziehung aus an\_qu\_user

```
DELETE FROM `an_qu_user` WHERE answer_id = 2728;
DELETE FROM `an_qu_user` WHERE answer_id = 3188;
DELETE FROM `an_qu_user` WHERE answer_id = 3300;
DELETE FROM `an_qu_user` where question_id = 876;
DELETE FROM `an_qu_user` where question_id = 948;
DELETE FROM `an_qu_user` where question_id = 915;
DELETE FROM `an_qu_user` where question_id = 877;
DELETE FROM `an_qu_user` where question_id = 1451;
DELETE FROM `an_qu_user` WHERE answer_id = 5598;
```

#### 6. Löschung der Antwort aus answer

```
DELETE FROM `answer` WHERE `answer`.`id` = 2728;
DELETE FROM `answer` WHERE `answer`.`id` = 3188;
DELETE FROM `answer` WHERE `answer`.`id` = 3300;
```

```

DELETE FROM `answer` WHERE `answer`.`id` = 3311;
DELETE FROM `answer` WHERE `answer`.`id` = 3312;
DELETE FROM `answer` WHERE `answer`.`id` = 3313;
DELETE FROM `answer` WHERE `answer`.`id` = 3314;
DELETE FROM `answer` WHERE `answer`.`id` = 3315;
DELETE FROM `answer` WHERE `answer`.`id` = 3316;
DELETE FROM `answer` WHERE `answer`.`id` = 3573;
DELETE FROM `answer` WHERE `answer`.`id` = 3574;
DELETE FROM `answer` WHERE `answer`.`id` = 3575;
DELETE FROM `answer` WHERE `answer`.`id` = 3576;
DELETE FROM `answer` WHERE `answer`.`id` = 3577;
DELETE FROM `answer` WHERE `answer`.`id` = 3457;
DELETE FROM `answer` WHERE `answer`.`id` = 3458;
DELETE FROM `answer` WHERE `answer`.`id` = 3459;
DELETE FROM `answer` WHERE `answer`.`id` = 3460;
DELETE FROM `answer` WHERE `answer`.`id` = 3461;
DELETE FROM `answer` WHERE `answer`.`id` = 3317;
DELETE FROM `answer` WHERE `answer`.`id` = 3318;
DELETE FROM `answer` WHERE `answer`.`id` = 3319;
DELETE FROM `answer` WHERE `answer`.`id` = 3320;
DELETE FROM `answer` WHERE `answer`.`id` = 3321;
DELETE FROM `answer` WHERE `answer`.`id` = 5520;
DELETE FROM `answer` WHERE `answer`.`id` = 5521;
DELETE FROM `answer` WHERE `answer`.`id` = 5522;
DELETE FROM `answer` WHERE `answer`.`id` = 5523;
DELETE FROM `answer` WHERE `answer`.`id` = 5524;
DELETE FROM `answer` WHERE `answer`.`id` = 5525;
DELETE FROM `answer` WHERE `answer`.`id` = 5598;

```

#### 7. Löschung der Frage aus question

```

DELETE FROM `question` WHERE `question`.`id` = 876;
DELETE FROM `question` WHERE `question`.`id` = 948;
DELETE FROM `question` WHERE `question`.`id` = 915;
DELETE FROM `question` WHERE `question`.`id` = 877;
DELETE FROM `question` WHERE `question`.`id` = 1451;

```

#### 8. Überprüfung ob alles geklappt hat, das Ergebnis der folgenden Abfrage sollte leer sein.

```

SELECT question_id, COUNT(*) count FROM answer_question
GROUP BY question_id HAVING COUNT(*) > 5;

```

## 2 Request-Update

Es können ebenfalls Anfragen für neue Sprachen und Themengebiete aus einer Frage entstehen. Dies wurde bisher noch nicht von der Datenbank unterstützt. Mit den folgenden Änderungen kann diese Unterstützung erreicht werden.

### 1. Login mit phpMyAdmin

### 2. Anpassen von language\_request

Bei folgendem Befehl muss beachtet werden, dass der Datenbankname auf dem cnlab-Server von unserem gewählten Datenbankname abweicht. Es müssen also sämtliche Vorkommnisse von mobilequizdb durch den auf dem cnlab-Server gewählten Namen ersetzt werden.

```
ALTER TABLE `language_request` ADD `question_id` INT(10)
UNSIGNED NULL DEFAULT NULL AFTER `questionnaire_id`;
```

```
ALTER TABLE `mobilequizdb`.`language_request` ADD INDEX
`fk_language_request_question` (`question_id`) USING BTREE;
```

```
ALTER TABLE `language_request` ADD CONSTRAINT
`fk_language_request_question` FOREIGN KEY (`question_id`)
REFERENCES `mobilequizdb`.`question`(`id`) ON DELETE
RESTRICT ON UPDATE RESTRICT;
```

### 3. Anpassen von topic\_request

Auch bei dem hier folgenden Befehl muss beachtet werden, dass der Datenbankname auf dem cnlab-Server von unserem gewählten Datenbankname abweicht. Es müssen also sämtliche Vorkommnisse von mobilequizdb durch den auf dem cnlab-Server gewählten Namen ersetzt werden.

```
ALTER TABLE `topic_request` ADD `question_id` INT(10)
UNSIGNED NULL DEFAULT NULL AFTER `questionnaire_id`;
```

```
ALTER TABLE `mobilequizdb`.`topic_request` ADD INDEX
`fk_topic_request_question` (`question_id`) USING BTREE;
```

```
ALTER TABLE `topic_request` ADD CONSTRAINT
`fk_topic_request_question` FOREIGN KEY (`question_id`)
REFERENCES `mobilequizdb`.`question`(`id`) ON DELETE
RESTRICT ON UPDATE RESTRICT;
```

### 3 Schreibfehler korrigieren

1. Login mit phpMyAdmin
2. Anpassung von Schreibfehlern

Es wurden in der Datenbank Schreibfehler bei Single-Choice und Multiple-Choice gefunden. Diese können wie folgt behoben werden.

```
UPDATE `question_type` SET `type` = 'singlechoice'  
WHERE `question_type`.`id` = 1;
```

```
UPDATE `question_type` SET `type` = 'multiplechoice'  
WHERE `question_type`.`id` = 2;
```

```
ALTER TABLE `questionnaire` CHANGE `singlechoise_multiplier`  
`singlechoice_multiplier` INT(3) NULL DEFAULT '1';
```

### 4 Korrektur der user\_data

- Login mit phpMyAdmin
- Es wurde festgestellt, dass die Fremdschlüsselbeziehung der user\_id in der Tabelle user\_data auf die Tabelle user mit dem Feld id nicht korrekt umgesetzt wurde. Es wurde deshalb versucht diese Fremdschlüsselbeziehung herzustellen. Auch bei diesen Befehlen ist zu beachten, dass der Datenbankname angepasst werden muss. Die dafür benötigten Befehle lauten wie folgt.

```
ALTER TABLE `mobilequizDB`.`user_data` ADD INDEX  
`fk_user_data_user` (`user_id`) USING BTREE;
```

```
ALTER TABLE `user_data` ADD CONSTRAINT `fk_user_data_user`  
FOREIGN KEY (`user_id`) REFERENCES `user` (`id`) ON DELETE  
RESTRICT ON UPDATE RESTRICT;
```

- Leider tritt bei der Ausführung des zweiten Befehls folgende Fehlermeldung auf: Cannot add or update a child row: a foreign key constraint fails. Es muss also noch überprüft werden, ob in der Tabelle user noch sämtliche Elemente vorhanden sind, welche mit user\_data verknüpft werden sollen.

### 5 Korrektur der user\_activation

- Login mit phpMyAdmin

- Es wurde festgestellt, dass die Fremdschlüsselbeziehung der user\_id in der Tabelle user\_activation auf die Tabelle user mit dem Feld id nicht korrekt umgesetzt wurde. Es wurde deshalb versucht diese Fremdschlüsselbeziehung herzustellen. Auch bei diesen Befehlen ist zu beachten, dass der Datenbankname angepasst werden muss. Die dafür benötigten Befehle lauten wie folgt.

```
ALTER TABLE `mobilequizDB`.`user_activation` ADD INDEX
`fk_user_activation_user` (`user_id`) USING BTREE;
```

```
ALTER TABLE `user_activation` ADD CONSTRAINT
`fk_user_activation_user` FOREIGN KEY (`user_id`) REFERENCES
`user`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

- Leider tritt bei der Ausführung des zweiten Befehls folgende Fehlermeldung auf: Cannot add or update a child row: a foreign key constraint fails. Es muss also noch überprüft werden, ob in der Tabelle user noch sämtliche Elemente vorhanden sind, welche mit user\_activation verknüpft werden sollen.

## 6 Umstellung auf Durchführung / execution

### 1. Login mit phpMyAdmin

Auch hier muss bei sämtlichen Befehlen beachtet werden, dass der Datenbankname von mobilequizdb auf den tatsächlich gewählten Namen des cnlab-Servers passt.

### 2. Erstellen der neuen Tabellen

```
CREATE TABLE `mobilequizdb`.`execution` ( `id` INT(10) UNSIGNED
NOT NULL AUTO_INCREMENT , `priority_id` INT(10) UNSIGNED NOT NULL ,
`name` VARCHAR(255) NULL DEFAULT NULL , `exec_token` VARCHAR(15) NULL
DEFAULT NULL , `creation_date` INT(25) NULL DEFAULT NULL ,
`last_modified` INT(25) NULL DEFAULT NULL , `starttime` INT(25) NULL
DEFAULT NULL , `endtime` INT(25) NULL DEFAULT NULL ,
`noParticipationPeriod` TINYINT(1) NULL DEFAULT '0' , `limited_time`
INT(25) NULL DEFAULT NULL , `amount_of_questions` INT(4) NULL
DEFAULT '0' , `amount_participations` INT(5) NULL DEFAULT '0' ,
`quiz_passed` INT(3) NULL DEFAULT '0' , `random_questions` TINYINT(1)
NULL DEFAULT '1' , `random_answers` TINYINT(1) NULL DEFAULT '1' ,
`singlechoice_multiplier` INT(3) NULL DEFAULT '1' , `public` TINYINT(1)
NULL DEFAULT '0' , `result_visible_points` TINYINT(1) NULL DEFAULT '1' ,
`result_visible` TINYINT(1) NULL DEFAULT NULL , `showTaskPaper`
TINYINT(1) NULL DEFAULT '0' , PRIMARY KEY (`id`)) ENGINE = InnoDB;
```

```
CREATE TABLE `mobilequizdb`.`priority` ( `id` INT(10) UNSIGNED NOT
```

```
NULL AUTO_INCREMENT , `name` VARCHAR(255) NULL DEFAULT NULL ,
PRIMARY KEY (`id`)) ENGINE = InnoDB;
```

```
CREATE TABLE `mobilequizdb`.`priority_settings` ( `priority_id`
INT(10) UNSIGNED NOT NULL , `user_id` INT(6) UNSIGNED NOT NULL ,
`starttime` INT(25) NULL DEFAULT NULL , `endtime` INT(25) NULL
DEFAULT NULL , `noParticipationPeriod` TINYINT(1) NULL DEFAULT '0' ,
`limited_time` INT(25) NULL DEFAULT NULL , `amount_of_questions` INT(4)
NULL DEFAULT '0' , `amount_participations` INT(5) NULL DEFAULT '0' ,
`quiz_passed` INT(3) NULL DEFAULT NULL , `random_questions` TINYINT(1)
NULL DEFAULT '1' , `random_answers` TINYINT(1) NULL DEFAULT '1' ,
`singlechoice_multiplier` INT(3) NULL DEFAULT '1' , `public` TINYINT(1)
NULL DEFAULT '0' , `result_visible_points` TINYINT(1) NULL DEFAULT '1' ,
`result_visible` TINYINT(1) NULL DEFAULT NULL , `showTaskPaper`
TINYINT(1) NULL DEFAULT '0' , PRIMARY KEY (`priority_id`, `user_id`))
ENGINE = InnoDB;
```

```
CREATE TABLE `mobilequizdb`.`user_group` ( `user_id` INT(6) UNSIGNED
NOT NULL , `group_id` INT(6) UNSIGNED NOT NULL , PRIMARY KEY
(`user_id`, `group_id`)) ENGINE = InnoDB;
```

```
CREATE TABLE `mobilequizdb`.`group_exec` ( `group_id` INT(6) UNSIGNED
NOT NULL , `execution_id` INT(10) UNSIGNED NOT NULL , PRIMARY KEY
(`group_id`, `execution_id`)) ENGINE = InnoDB;
```

```
CREATE TABLE `mobilequizdb`.`qunaire_exec` ( `questionnaire_id` INT(10)
UNSIGNED NOT NULL , `execution_id` INT(10) UNSIGNED NOT NULL , PRIMARY
KEY (`questionnaire_id`, `execution_id`)) ENGINE = InnoDB;
```

```
CREATE TABLE `mobilequizdb`.`user_exec` ( `user_id` INT(6) UNSIGNED
NOT NULL , `execution_id` INT(10) UNSIGNED NOT NULL , PRIMARY KEY
(`user_id`, `execution_id`)) ENGINE = InnoDB;
```

### 3. Fremdschlüssel-Verbindungen herstellen

```
ALTER TABLE `mobilequizdb`.`execution` ADD INDEX
`fk_execution_priority` (`priority_id`) USING BTREE;
```

```
ALTER TABLE `execution` ADD CONSTRAINT `fk_execution_priority`
FOREIGN KEY (`priority_id`) REFERENCES `mobilequizdb`.`priority`(`id`)
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```

ALTER TABLE `mobilequizdb`.`user_exec` ADD INDEX `fk_user_exec_user`
(`user_id`) USING BTREE;

ALTER TABLE `mobilequizdb`.`user_exec` ADD INDEX
`fk_user_exec_execution` (`execution_id`) USING BTREE;

ALTER TABLE `user_exec` ADD CONSTRAINT `fk_user_exec_user`
FOREIGN KEY (`user_id`) REFERENCES `mobilequizdb`.`user`(`id`)
ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE `user_exec` ADD CONSTRAINT `fk_user_exec_execution`
FOREIGN KEY (`execution_id`) REFERENCES `mobilequizdb`.`execution`(`id`)
ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE `mobilequizdb`.`qunaire_exec` ADD INDEX
`fk_qunaire_exec_questionnaire` (`questionnaire_id`) USING BTREE;

ALTER TABLE `mobilequizdb`.`qunaire_exec` ADD INDEX
`fk_qunaire_exec_execution` (`execution_id`) USING BTREE;

ALTER TABLE `qunaire_exec` ADD CONSTRAINT
fk_qunaire_exec_questionnaire` FOREIGN KEY (`questionnaire_id`)
REFERENCES `mobilequizdb`.`questionnaire`(`id`) ON DELETE RESTRICT
ON UPDATE RESTRICT;

ALTER TABLE `qunaire_exec` ADD CONSTRAINT `fk_qunaire_exec_execution`
FOREIGN KEY (`execution_id`) REFERENCES `mobilequizdb`.`execution`(`id`)
ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE `mobilequizdb`.`group_exec` ADD INDEX `fk_group_exec_group`
(`group_id`) USING BTREE;

ALTER TABLE `mobilequizdb`.`group_exec` ADD INDEX
`fk_group_exec_execution` (`execution_id`) USING BTREE;

ALTER TABLE `group_exec` ADD CONSTRAINT `fk_group_exec_group`
FOREIGN KEY (`group_id`) REFERENCES `mobilequizdb`.`group`(`id`)
ON DELETE RESTRICT ON UPDATE RESTRICT;

ALTER TABLE `group_exec` ADD CONSTRAINT `fk_group_exec_execution`
FOREIGN KEY (`execution_id`) REFERENCES `mobilequizdb`.`execution`(`id`)

```



```
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `mobilequizdb`.`priority_settings` ADD INDEX  
`fk_priority_settings_priority` (`priority_id`) USING BTREE;
```

```
ALTER TABLE `mobilequizdb`.`priority_settings` ADD INDEX  
`fk_priority_settings_user` (`user_id`) USING BTREE;
```

```
ALTER TABLE `priority_settings` ADD CONSTRAINT  
`fk_priority_settings_priority` FOREIGN KEY (`priority_id`)  
REFERENCES `mobilequizdb`.`priority`(`id`) ON DELETE RESTRICT  
ON UPDATE RESTRICT;
```

```
ALTER TABLE `priority_settings` ADD CONSTRAINT  
`fk_priority_settings_user` FOREIGN KEY (`user_id`) REFERENCES  
`mobilequizdb`.`user`(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `mobilequizdb`.`user_group` ADD INDEX `fk_user_group_user`  
(`user_id`) USING BTREE;
```

```
ALTER TABLE `mobilequizdb`.`user_group` ADD INDEX `fk_user_group_group`  
(`group_id`) USING BTREE;
```

```
ALTER TABLE `user_group` ADD CONSTRAINT `fk_user_group_user`  
FOREIGN KEY (`user_id`) REFERENCES `mobilequizdb`.`user`(`id`)  
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

```
ALTER TABLE `user_group` ADD CONSTRAINT `fk_user_group_group`  
FOREIGN KEY (`group_id`) REFERENCES `mobilequizdb`.`group`(`id`)  
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

#### 4. Migration der Daten

Beim Einfügen in die neue Tabelle execution kann der Name dieser default erstellen Durchführungen festgelegt werden. Im unten stehenden zweiten Befehl wurde ,default.execution' gewählt, dies kann allerdings beliebig angepasst werden.

```
INSERT INTO `priority` VALUES (0, 'learning_help');  
UPDATE `priority` SET `id` = '0' WHERE `priority`.`id` = 1;  
INSERT INTO `priority` VALUES (1, 'exam_requirement');  
INSERT INTO `priority` VALUES (2, 'exam');  
  
INSERT INTO `execution` (`priority_id`, `name`, `exec_token`,
```

```

`creation_date`, `last_modified`, `starttime`, `endtime`,
`noParticipationPeriod`, `limited_time`, `amount_of_questions`,
`amount_participations`, `quiz_passed`, `random_questions`,
`random_answers`, `singlechoice_multiplier`, `public`,
`result_visible_points`, `result_visible`, `showTaskPaper`) SELECT
`priority`, `default_execution`, `qnaire_token`, `creation_date`,
`last_modified`, `starttime`, `endtime`, `noParticipationPeriod`,
`limited_time`, `amount_of_questions`, `amount_participations`,
`quiz_passed`, `random_questions`, `random_answers`,
`singlechoice_multiplier`, `public`, `result_visible_points`,
`result_visible`, `showTaskPaper` FROM `questionnaire`;

INSERT INTO `qnaire_exec` (`questionnaire_id`, `execution_id`)
SELECT questionnaire.id, execution.id FROM `questionnaire`
INNER JOIN `execution` ON questionnaire.qnaire_token =
execution.exec_token;

INSERT INTO `group_exec` SELECT group_id, qnaire_exec.execution_id
FROM `assign_group_qnaire` INNER JOIN `qnaire_exec` ON
qnaire_exec.questionnaire_id = assign_group_qnaire.questionnaire_id;

INSERT INTO `user_group` (`user_id`, `group_id`) SELECT id, group_id
FROM `user` WHERE group_id IS NOT NULL;

ALTER TABLE `user_qnaire_session` ADD `execution_id` INT(10)
UNSIGNED NOT NULL AFTER `questionnaire_id`;

UPDATE user_qnaire_session uqs, qnaire_exec qe SET uqs.execution_id =
qe.execution_id WHERE uqs.questionnaire_id = qe.questionnaire_id;

ALTER TABLE `mobilequizdb`.`user_qnaire_session` ADD INDEX
`fk_user_qnaire_session_execution` (`execution_id`) USING BTREE;

ALTER TABLE user_qnaire_session DROP FOREIGN KEY
fk_questionnaire_session;

ALTER TABLE user_qnaire_session DROP INDEX fk_questionnaire_session;

ALTER TABLE `user_qnaire_session` ADD CONSTRAINT
`fk_user_qnaire_session_execution` FOREIGN KEY (`execution_id`)
REFERENCES `mobilequizdb`.`execution`(`id`) ON DELETE RESTRICT

```

```
ON UPDATE RESTRICT;
```

```
ALTER TABLE `user_qunaire_session` DROP `questionnaire_id`;
```

```
RENAME TABLE `user_qunaire_session` TO `user_exec_session`;
```

```
DROP TABLE `assign_group_qunaire`;
```

```
ALTER TABLE `questionnaire` DROP `starttime`, DROP `endtime`, DROP  
`qnaire_token`, DROP `random_questions`, DROP `random_answers`, DROP  
`limited_time`, DROP `result_visible`, DROP `result_visible_points`,  
DROP `amount_of_questions`, DROP `public`, DROP `picture_link`, DROP  
`priority`, DROP `amount_participations`, DROP `quiz_passed`, DROP  
`singlechoice_multiplier`, DROP `noParticipationPeriod`, DROP  
`showTaskPaper`;
```

## 7 Interessen-Gebiete

### 1. Login mit Putty

### 2. Fremdschlüsselbeziehungen herstellen

Auch hier muss darauf geachtet werden, dass der Datenbankname auf den Namen geändert wird, der auf dem cnlab-Server gewählt wurde.

```
ALTER TABLE `group` ADD `subject_id` INT(6) UNSIGNED NULL  
DEFAULT NULL AFTER `token`;
```

```
ALTER TABLE `mobilequizdb`.`group` ADD INDEX `fk_group_subject`  
(`id`) USING BTREE;
```

```
ALTER TABLE `mobilequizdb`.`group` DROP INDEX `fk_group_subject`,  
ADD INDEX `fk_group_subject` (`subject_id`) USING BTREE;
```

```
ALTER TABLE `group` ADD CONSTRAINT `fk_group_subject` FOREIGN KEY  
(`subject_id`) REFERENCES `mobilequizdb`.`subjects`(`id`) ON DELETE  
RESTRICT ON UPDATE RESTRICT;
```

### 3. Manuelles Erstellen der Einträge für die bestehenden Interessen-Gebiete.

Der unten aufgeführte Befehl muss für alle bereits bestehenden Interessen-Gebiete durchgeführt werden. Es muss dabei darauf geachtet werden, dass der token immer unterschiedlich ist.

```
INSERT INTO `group` (`id`, `name`, `owner_id`, `token`,  
`subject_id`) VALUES ('22',  
'interest_group_Informationssicherheit', '260', '324f79', '11');
```