

SIMPLE QUIZ:

Quiz maker is an easy app to create very simple multiple choice type quizzes.

See https://en.wikipedia.org/wiki/Multiple_choice

Here you can build a very crude multiple choice type quiz. The coach can create a number of questions and provide multiple answers, of which only one being true. At the end he can provide a percentage necessary to pass the test; a file to be uploaded to a web server will be created. Follow all the steps to create one.

This project is a proof of concept to learn PHP and Mysql, there's room for improvement.

The main goal is

- 1) To implement the building of a DB (using primary and foreign keys)
- 2) Once created, use such DB to build a PHP file
- 3) Export routines directly in the file; results should not be visible

No external DB or JS should be present in the final file, the quiz itself should run on a web server. Results are not present nor visible in the HTML file provided to the student.

Name of the app: simple quiz

Type: open source app, first release

Price: free, open source

MUST HAVE:

- Works on a web server, in single user mode
- The user can create a simple quiz which is being stored on a DB
- Once the quiz is done, th user can create a PHP file containing the quiz, such file can be uploaded on a web server
- The user shouldn't be able to read the answers while is doing the test by checking the code of the HTML file
- The PHP file should be self containing, no extra files needed
- Users can login and change their password or user ID
- Forgotten password utility

SHOULD HAVE:

- Ajax could be used to upgrade the forms while errors are created, while now this does not happen
- Should work not only locally but also on the internet
- User could save a SQL backup-file of the DB before erasing the content of the DB.
- User could select all the previous quizzes, choose one and print out a PHP file from a specific quiz.
- Changing text content and adding other questions once the quiz is done (but this would probably break the logic of the app).
- The PHP file created should not be hard-linked while the app is running on a web server accessible on the internet, but it's ok locally.
- Bootstrap CSS
- Small CSS transitions
- Showing feedback on CSS once the quiz is done, by showing different colors on Divs
- HTML and PHP beautify
- Browser compatibility, Mobile compatibility, Media queries

COULD HAVE:

- Choosing colors and style for the quiz
- Expire password routine
- Strong password hashing
- Using textareas instead of text fields
- Use 2 random variables instead of one: one to randomize the order of the choices of a single question and the other to randomize the order of all the questions
- Create a video tutorial to be integrated inside the app
- Send PHP file via mail
- Send quiz results to the coach
- GIT repository
- Choosing randomly a given number of questions from a bigger number of available questions
- Creating different categories for a single quiz

- Batch script for preparing the project (changing email and SMTP data, could be a batch script, PHP or an INI file)
- SMTP data saved in DB (see file change.php)

WON'T HAVE

- Saving results in a DB
- Having different kinds of tests
- Saving results in a file (for example a PDF file)
- Registration of the students
- Hiding results to the students
- Local DB saved on the students's PC
- Adding images to quiz
- More advanced CSM functions

SECURITY:

- PDO prepared statements
- Catch throw structures for PDO errors
- Catch throw structures for Exceptions
- Catch throw structures generic Errors

This is used for DB connections, queries, Sessions and I/O operations.

- Password and Session protection
- MD5 hashing for passwords
- Sha512 hashing/token (password reset)
- Email control for username/password resetting
- HTML entities on files for characters as <%*\$ and such
- Checking DB consistency once data is being erased

PASSWORDS/SETTINGS:

A field for the administrator's password is saved in the database (see sql.sql) and can be changed using a text editor or directly using PHPMyAdmin or MySql together with the terminal.

Data to be changed:

- [email](#) in users table (see email row)

in sql.sql file:
UPDATE users
SET email = "email";

Change SMTP connection data
In change.php file:

- mail->Host = 'host'; // Specify main and backup SMTP servers
- \$mail->SMTPAuth = true; // Enable SMTP authentication
- \$mail->Username = 'username'; // SMTP username
- \$mail->Password = 'password'; // SMTP password
- \$mail->Port = 2525; // TCP port

Change also the data in php/lib/pdo.inc.php to connect to the MySQL server:

host
port
user
password

First run/default data:

user id: **user**

password: **password**

the app asks the user to change default data.

standard MySQL userid/password can be used

LIMITATIONS:

- Data type for numbers: check sql.sql
- No answers can be repeated
- No quiz with one answer, a quiz can have minimum two answers, but it's ok to have a quiz containing only one question and two answers
- Max length for a single question: 600 characters
- Max length for titles and user name: 50 characters
- Max 25 answers for a single question
- Max 50 questions for 1 document (a single quiz)
- Max 50 documents/quizzes in the DB

So for 1 document there are 50×25 (1250) possible answers and there's a maximum limit of 50 documents, so 1250×50 (62500) answers before the DB is full, then one needs to empty it before using again the app.
Folders containing the quizzes are not erased.

DATABASE:

MySql type

Server version: 5.5.42/ PHP: 7.0.8

Name: simple_quiz

Access DB method: PDO

At the moment there's no backup utility, data is saved directly in the PHP documents but an sql file is available to build the project.

EVOLUTION:

Ajax, a tutorial, backup utility, more CRUD functions available
More control and testing, Browser compatibility, Mobile compatibility,
Media queries

LIBRARIES/EXTERNAL CODE/FILES:

- Bootstrap 4
- myExceptionHandler.inc.php (Micky De Pauw MIT License)
- PHPMailer-5.2 (GNU Lesser General Public License v2.1)
- Radio buttons: <http://jsfiddle.net/xykPT/> (Digital Ocean, no licence)

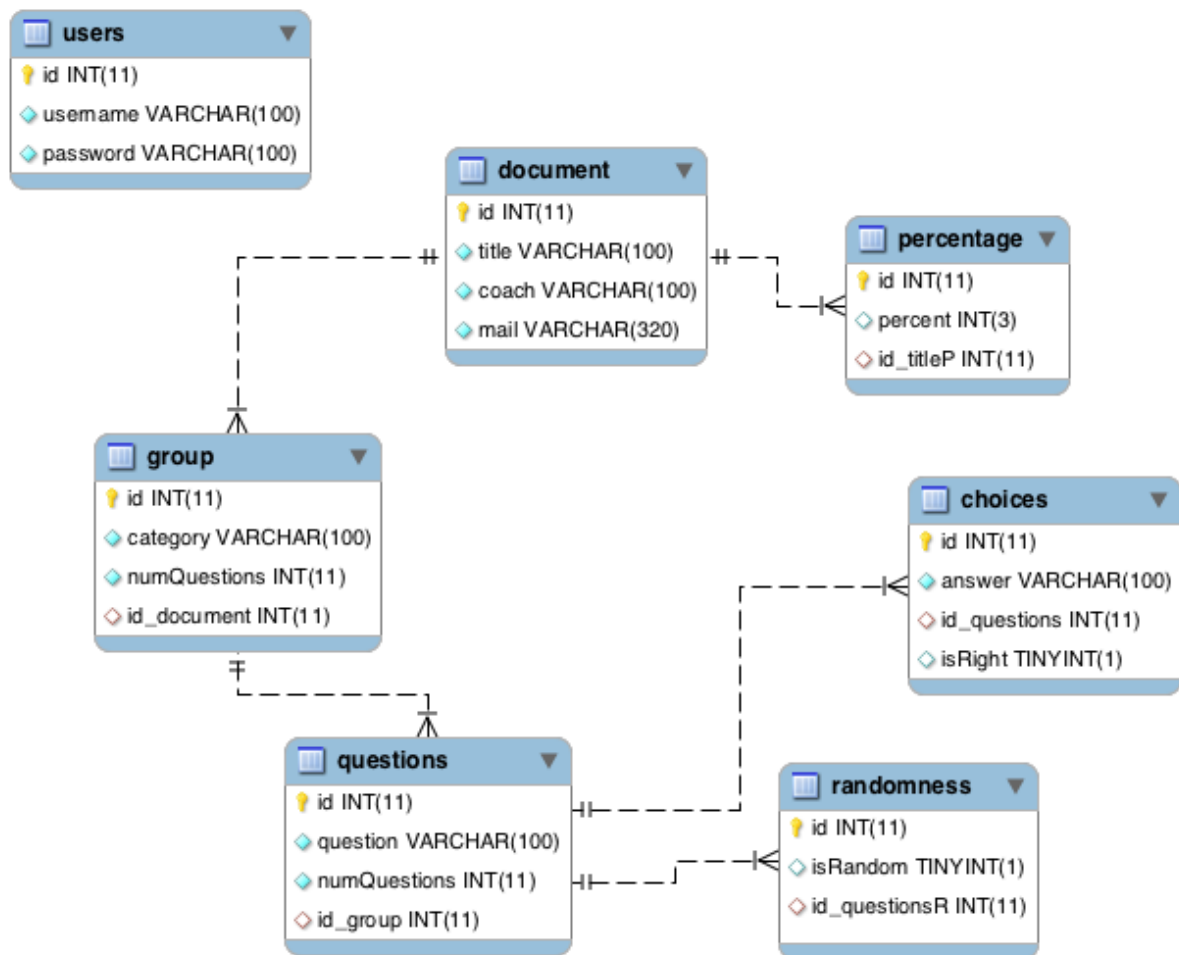
Snippets and ideas from different websites:

- <https://stackoverflow.com/questions/35419235/customizing-html-range-input-with-a-floating-div-that-contains-current-value>
- <https://css-tricks.com/value-bubbles-for-range-inputs/>
- <https://codepen.io/chriscoyier/pen/imdrE>
- <https://stackoverflow.com/questions/38773520/move-element-with-range-directly-javascript>
- <https://codepen.io/abhisharma2/pen/wMOpqz>
- <http://thenewcode.com/757/Playing-With-The-HTML5-range-Slider-Input>

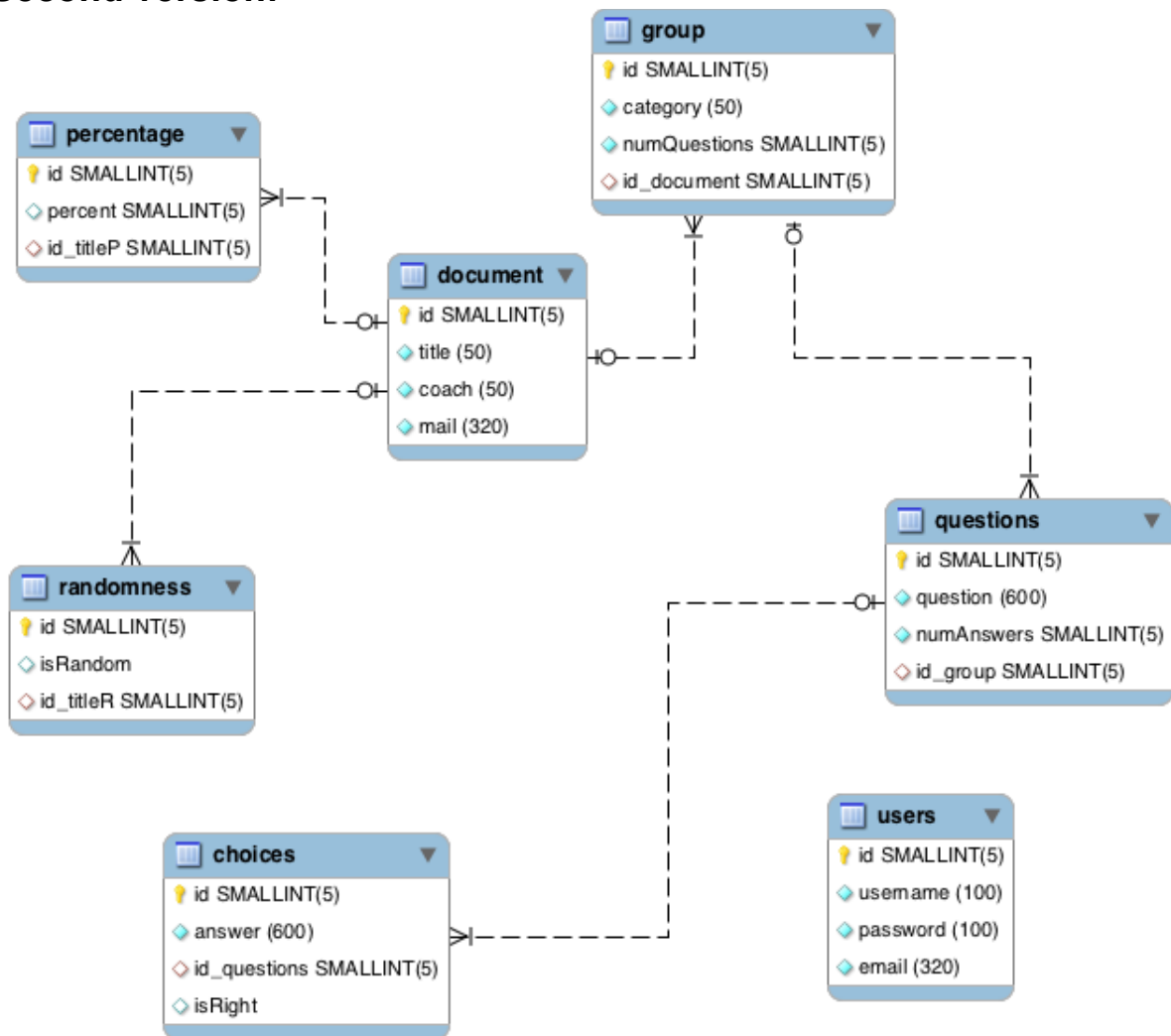
IMAGES:

- <https://pixabay.com/it/colore-triangolo-geometrico-2174065/>
- <https://victimoutreach.org/images/icn/onebit/>

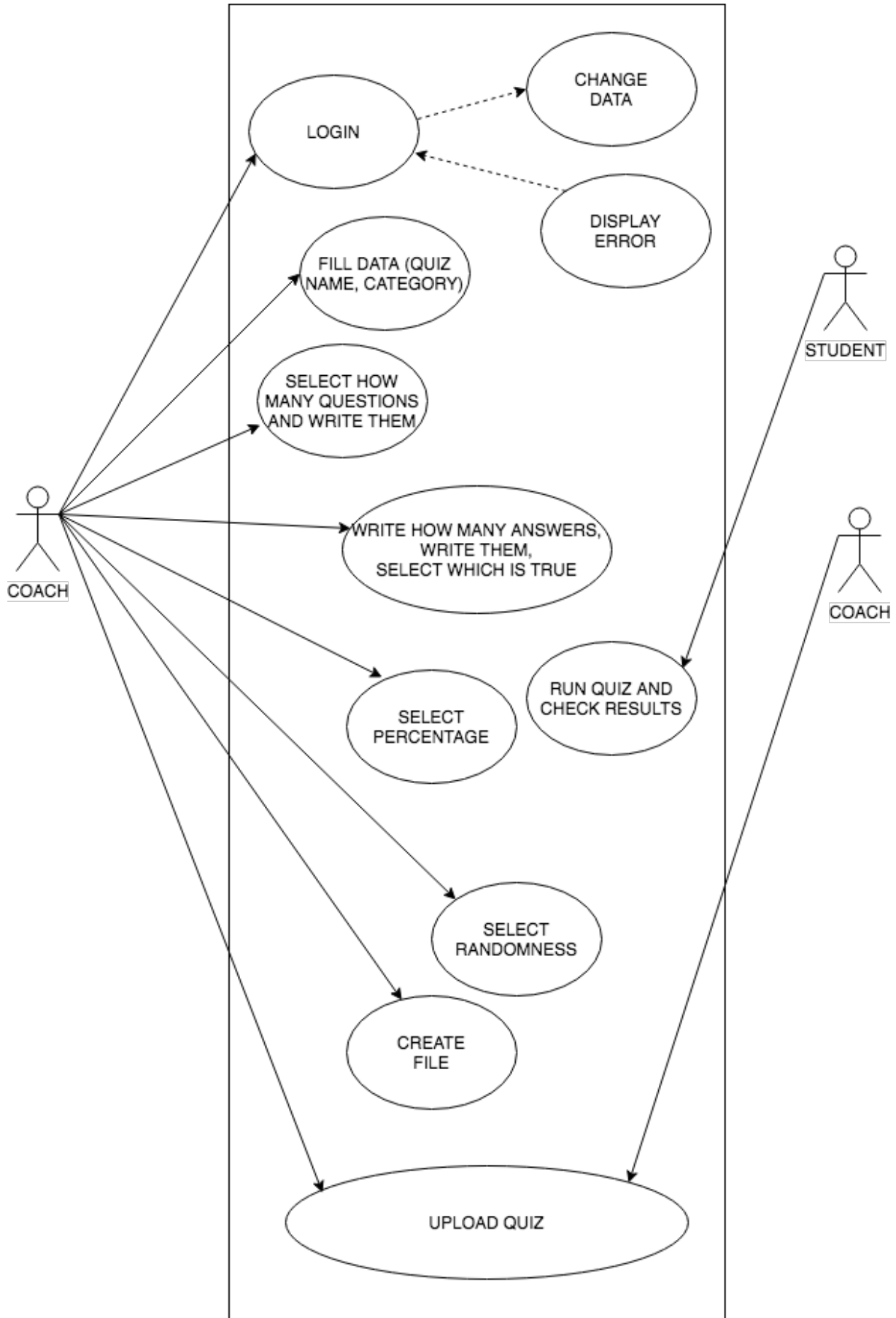
First version ERD DIAGRAM (MySql workbench) data type is then being changed



Second version:



USE CASE DIAGRAM:

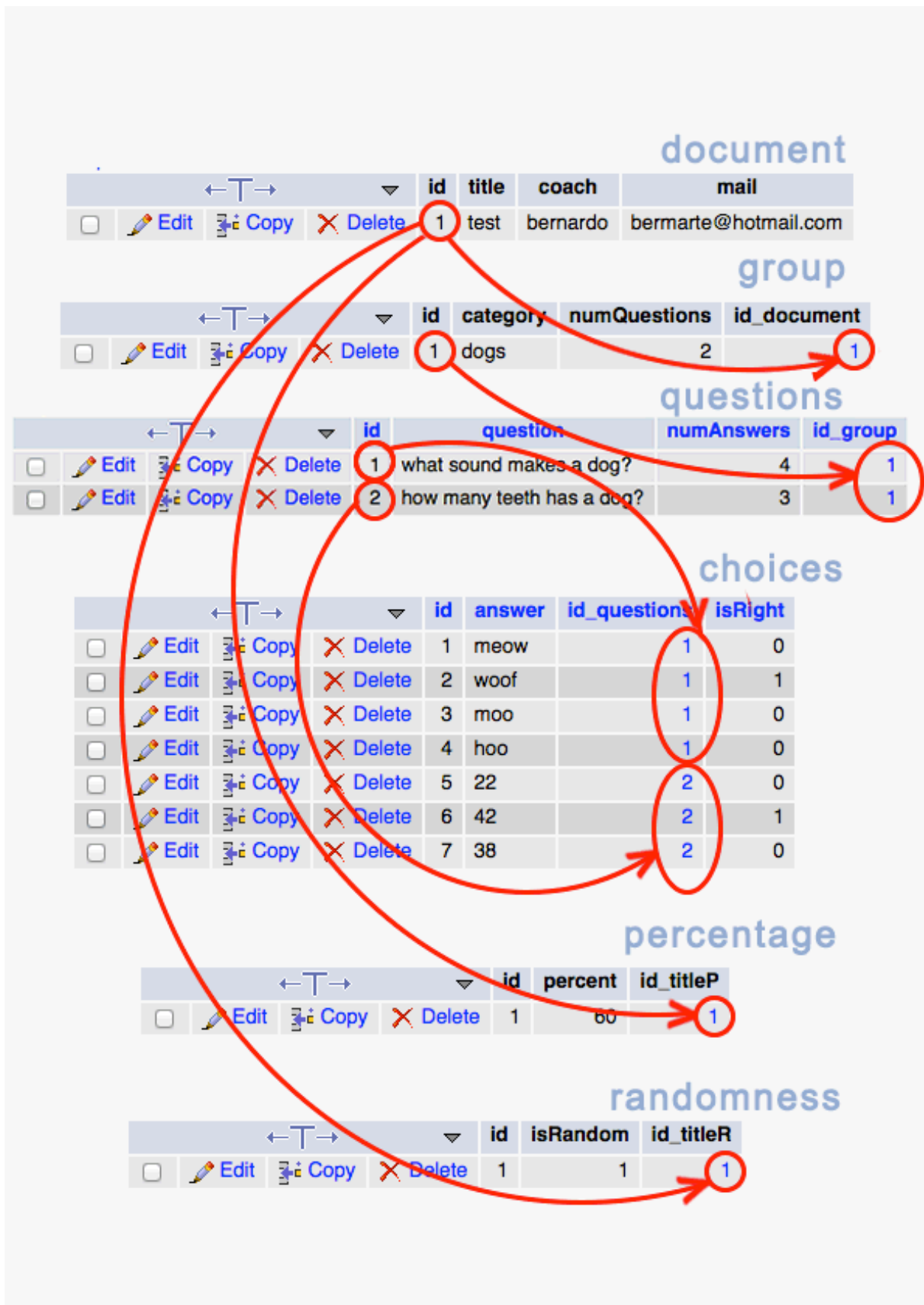


INSERT DATA EXAMPLE:

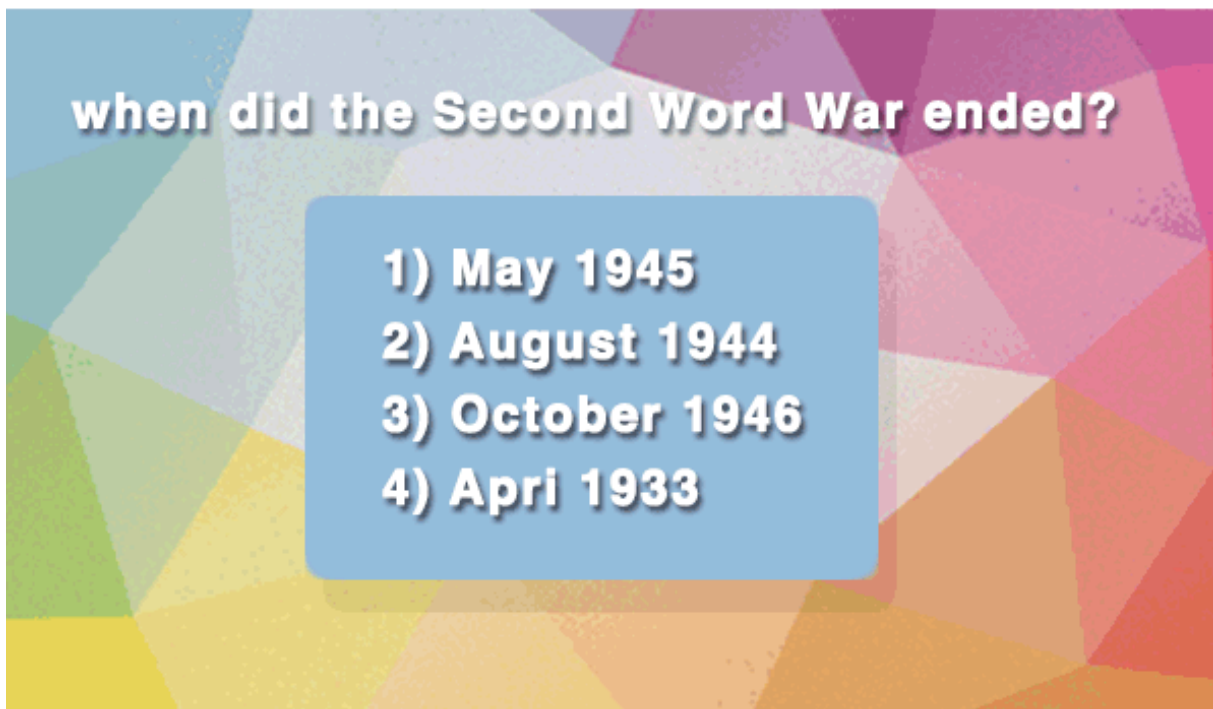
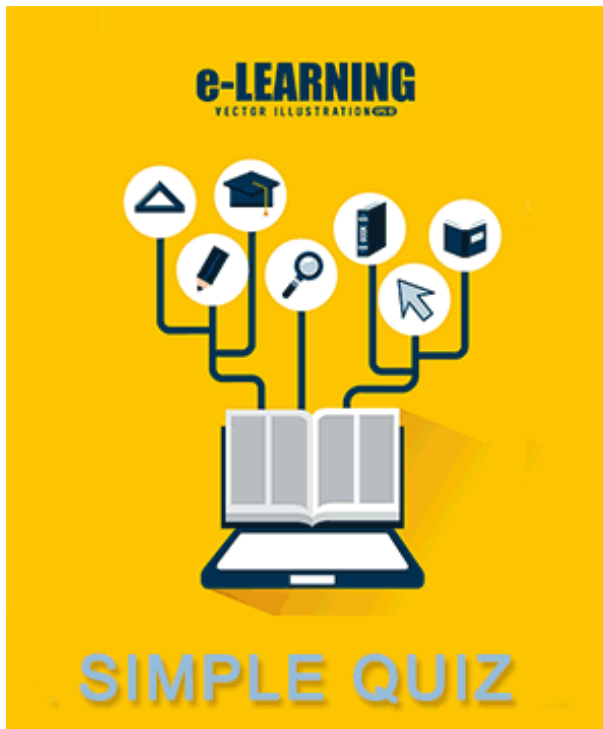
```
-- insert a complete quiz
INSERT INTO simple_quiz.document (title,coach,mail)
VALUES ('test', 'bermarte', 'bermarte@website.com');
INSERT INTO simple_quiz.group(category, numQuestions,id_document)
VALUES ('dogs',2,1);
INSERT INTO questions(question, id_group, numAnswers)
VALUES ('what sound makes a dog?', 1, 4);
INSERT INTO choices (answer, id_questions, isRight)
VALUES ('meow', 1, false);
INSERT INTO choices (answer, id_questions, isRight)
VALUES ('woof', 1, true);
INSERT INTO choices (answer, id_questions, isRight)
VALUES ('moo', 1, false);
INSERT INTO choices (answer, id_questions, isRight)
VALUES ('hoo', 1, false);
-- insert second question
```

```
INSERT INTO questions(question, id_group, numAnswers)
VALUES ('how many teeth has a dog?', 1, 3);
INSERT INTO choices (answer, id_questions, isRight)
VALUES ('22', 2, false);
INSERT INTO choices (answer, id_questions, isRight)
VALUES ('42', 2, true);
INSERT INTO choices (answer, id_questions, isRight)
VALUES ('38', 2, false);
INSERT INTO randomness(isRandom, id_titleR)
VALUES(true, 1);
INSERT INTO percentage(percent, id_titleP)
VALUES (60, 1);
```

F-KEYS/RELATIONS EXAMPLE:



MOODBOARDS:





SIMPLE QUIZ



NOTA:

This app is the eind-project for the course as Web-Developer at ENCORA given in 2018 by teacher Micky De Pauw.

De cursus webontwikkeling wordt gegeven aan het SCVO-ENCORA te Berchem. Meer informatie vind je op de website van ENCORA (<http://www.encora.be>)