BRAINSTER

Workshop: PDO & Session storage

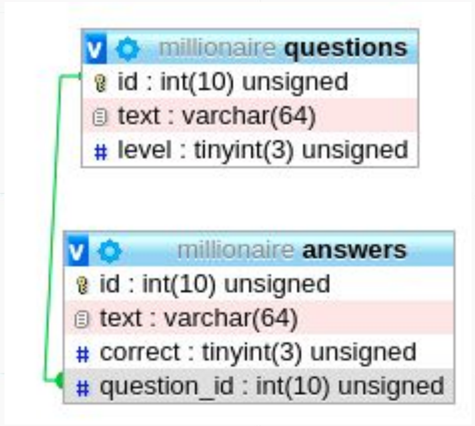
Академија за програмирање

The database

In this workshop, we will create the 'Who wants to be a millionaire' game.

To accomplish that, first thing we'll need is a database to store the questions and the answers.

Create the database. There are multiple possible solutions. For example:



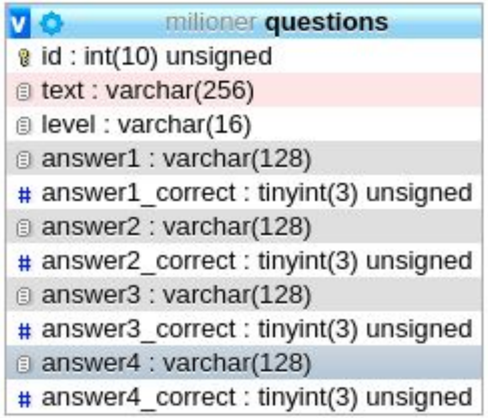
millionaire questions

- id : int(10) unsigned
- text : varchar(64)
- level : tinyint(3) unsigned

millionaire answers

- id : int(10) unsigned
- text : varchar(64)
- correct : tinyint(3) unsigned
- question_id : int(10) unsigned

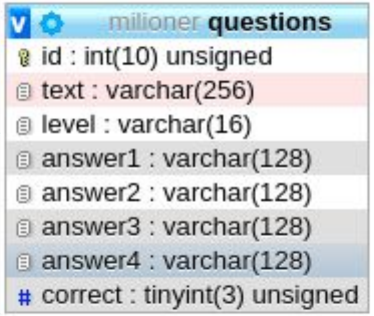
vs.



milioner questions

- id : int(10) unsigned
- text : varchar(256)
- level : varchar(16)
- answer1 : varchar(128)
- answer1_correct : tinyint(3) unsigned
- answer2 : varchar(128)
- answer2_correct : tinyint(3) unsigned
- answer3 : varchar(128)
- answer3_correct : tinyint(3) unsigned
- answer4 : varchar(128)
- answer4_correct : tinyint(3) unsigned

vs.



milioner questions

- id : int(10) unsigned
- text : varchar(256)
- level : varchar(16)
- answer1 : varchar(128)
- answer2 : varchar(128)
- answer3 : varchar(128)
- answer4 : varchar(128)
- correct : tinyint(3) unsigned

The database

The first approach would be useful if we have variable number of answers for each question, for example if one question can have 3 answers, other 4, some 5 etc.

Since the millionaire game is played with 4 possible answers for each question, we can simplify things with the second or third approach.

Second approach stores 0 or 1 for each answer's 'correct' column.

Third approach has 1 column that will have one of the following values [1,2,3,4] depending on which answer is correct.

Since the third approach is the simplest one, let's go with it. Create the table.

CRUD

Create CRUD for managing quiz questions. Although against everything we did until now, try to fit all 4 functionalities in a single file: dashboard.php. Create and edit functionalities should be handled with the same form.

There should be a form to create a question. When submitted there should be validation whether all 4 possible answers are provided and whether one is marked as correct.

All the questions should be listed in a table and next to each one there should be buttons to Delete or Edit a question.

When Delete button is clicked it deletes the appropriate question.

When Edit button is clicked it prepopulates the form used to create a question. The ‘Add’ becomes ‘Update’ button and a ‘Cancel’ button appears (Hint: google `javascript reload page get request`).

Every action should be accompanied with appropriate message upon success and failure. For example: Question created, Question deleted, Question updated, Question can not be deleted etc.

Authentication

Limit the dashboard to only signed-in users.

If someone is not signed-in, he should be redirected to our home page.

On the homepage there should be a link to our Signin page and a Start quiz button.

Authentication should be implemented through database. In the database create table admins, where all administrators will be stored. Beware: don’t store passwords in plain text, use md5 hashing at least.

Admins will be able to sign in providing password and either their username or email.

When the admin attempts to sign in, we first check their credentials. If they are correct, we setup a session for that admin, so that their subsequent visits are authenticated. We then take him to the dashboard page where he can edit the quiz questions. On the dashboard we need a logout button which will clear the session and redirect the admin to our home page.

On the dashboard we also need a ‘Add user’ button which will take the signed in user to a page where he can add/edit/delete the admins that will be able to access the dashboard (able to sign-in).

The Quiz

If the user is on the homepage and hits the ‘Start Quiz’ button, he should be taken to a quiz.php page.

Here we have 2 options, easy one and hard one.

Try to complete the easy one during the workshop.

Try to complete the hard one as homework.

The easy one

On the quiz page, from database choose 5 questions with level 'easy', 5 questions with level 'medium' and 5 questions with level 'hard'.

Print them all to the user in the appropriate order, first the easy ones, then the medium ones and then the hard ones. Below each question there should be 4 radio buttons representing the 4 possible answers.

The user needs to answer all the questions and hit a button 'Check if I am a millionaire'.

We validate if he has all 15 questions answered and check if they are all correct. If they are all correct we print him 'Congratulations, you are a millionaire'. If at least one is incorrect we print 'Sorry, you got X/15 correct', where X is the number of correct answers given.

The hard one

Using session storage to track user’s progress, show him the questions one by one. Start by 5 with level easy, then 5 with level medium and finally 5 with level hard. Choose the questions randomly.

Show him next question as long as he gives correct answer to the current one.

If he sequentially gives 15 correct answers, print `Congratulations, you are a millionaire’.

If he gives wrong answer to any question, the quiz ends for him.
Print `Sorry, better luck next time`, clear the session and print a ‘Start over’ button.

The result

Click the above text to see this workshop in action