

- Database Design
 - Tables
 - Quizzes
 - Questions
 - Choices
 - Users
 - QuizRecords
 - QuestionRecords
 - ChoiceRecords
 - Notes
 - About User
 - About Records of History Submissions
 - About How To Create / Modify a Quiz
 - What Information I Can Use

Database Design

Tables

Quizzes

```
CREATE TABLE
IF NOT EXISTS Quizzes (
    id INT not null PRIMARY KEY,
    title VARCHAR ( 255 ), # Quiz title
    tag varchar (36), # Quiz tag, which will not be exposed to users. This field can
    description VARCHAR ( 255 ), # descriptions for the Quiz
    total_points INT, # a positive integer, usually set to 100
    destroyed boolean not null
)
```

Questions

```

CREATE TABLE
IF NOT EXISTS Questions (
    id INT NOT NULL PRIMARY KEY,
    description VARCHAR ( 1023 ) NOT NULL, # the problem descriptions
    percentage INT ( 1 ) NOT NULL, # Percentage in total scores, eg. integer 5 means
    destroyed boolean, # indicates whether this question is destroyed, or whether it
    quiz_id INT,
    FOREIGN KEY ( quiz_id ) REFERENCES Quizzes ( id )
)

```

Choices

```

CREATE TABLE
IF NOT EXISTS Choices (
    id INT NOT NULL PRIMARY KEY,
    description VARCHAR ( 255 ) NOT NULL,
    seq INT ( 1 ), # sequence number of the question, usually from 0 to 3
    question_id INT, # references a question id
    isCorrect boolean NOT NULL, # indicate whether this choice is correct or one of
    UNIQUE KEY ( seq, question_id ), # make sure sequence number is unique in the q
    FOREIGN KEY ( question_id ) REFERENCES Questions ( id )
)

```

Users

```

CREATE TABLE
IF NOT EXISTS Users (
    id INT NOT NULL PRIMARY KEY,
    username VARCHAR ( 255 ), # allow null
    PASSWORD VARCHAR ( 255 ) # allow null since there is no sign in system
)

```

QuizRecords

Records the information once a user submitted a quiz.

```

CREATE TABLE
IF NOT EXISTS QuizRecords (
    id INT NOT NULL PRIMARY KEY,
    quiz_id INT NOT NULL,
    user_id INT NOT NULL,
    total_points INT, # since quiz points may be modified, a snapshot of the original
    title VARCHAR ( 255 ), # same
    description VARCHAR ( 255 ), # same
    submit_time CHAR ( 10 ), # unix timestamp: length 10
    UNIQUE KEY ( user_id, quiz_id ), # since user is created every time a quiz is submitted
    FOREIGN KEY ( quiz_id ) REFERENCES Quizzes ( id ),
    FOREIGN KEY ( user_id ) REFERENCES Users ( id )
)

```

QuestionRecords

Information for each question of the quiz that the user submitted.

```

CREATE TABLE
IF NOT EXISTS QuestionRecords (
    id INT NOT NULL PRIMARY KEY,
    quizrecord_id int not null,
    question_id int not null,
    score int, # Percentage of points the user scored, eg. integer 50 means the user scored 50%
    FOREIGN KEY ( quizrecord_id ) REFERENCES QuizRecords ( id ),
    FOREIGN KEY ( question_id ) REFERENCES Questions ( id ),
    UNIQUE KEY ( quizrecord_id, question_id )
)

```

ChoiceRecords

Specific choices made by users relational to a specific question of the quiz.

```

CREATE TABLE
IF NOT EXISTS ChoiceRecords (
    id INT NOT NULL PRIMARY KEY,
    questionrecord_id int not null,
    choice_id int not null, # the choices that the user selected of the certain question
    FOREIGN KEY ( questionrecord_id ) REFERENCES QuestionRecords ( id ),
    FOREIGN KEY ( choice_id ) REFERENCES Choices ( id ),
    UNIQUE KEY ( questionrecord_id, choice_id )
)

```

Notes

About User

- A user is anonymous in this system. A user is not required to sign in before submit a quiz. Each time a user submitted a quiz, a record of user is created. This table is only for future use. But for now, there is an Admin user which does have a username and password so he/she can log in to the CMS web page.

About Records of History Submissions

I believe the analysis of the received data is the value of the quiz. So there are a few problems I have to mention here.

- Records of history submissions are relational to Quiz ids, Question ids... So any future modification on an existing question / choice may cause unknown errors in the authenticity of the history records.
- To avoid this, a full snapshot of the original quiz should be stored in database which will require much more efforts, and also take too much redundant space for database. For example, if you add a new question to an existing quiz, participants before the modification will not have record of it and previous score percentage may not be reliable compared to those after. Because the new added question is, in a sense, correct for all history submissions, considering cases when you want to calculate the average scores mathematically. And more issues may be discussed...
- A second solution is brutal, which is to create a new quiz rather than making adjustments. And this is the way I strongly recommend.
- In current version of design, I have made snapshots of the total points of the quiz, and percentage of scores of each question. These designs are dependent on how you want to use the data in the future when there is a modification. Do you want to use the original scores or you want to recalculate scores according to current modifications? Anyway, modifications can be really tricky, especially when we are not sure how to judge the data is 'real' or 'valid'.
- To conclude. My suggestion is to create a new quiz once it requires modification. However, some minor changes in texts which won't affect the correctness of a question are tolerant.

About How To Create / Modify a Quiz

- I will build an admin platform so you can view the records of submissions and create a quiz. The detailed functions of interfaces can be added if there is a strong need. Otherwise I will implement it in my way.
- Users are not required to sign in. They can select a quiz and submit it. And records will be stored in database.

What Information I Can Use

Note: No user information is stored! A user can actually take a quiz for multiple times and multiple records will be stored in database and they are recognized as different users because they are anonymous.

Below are data that are stored:

- Total score of the quiz of each submission.
- Score percentage of each question of each quiz of each submission.
- Selected choices of the question of each submission.

Further analytics can be inferred by it:

- Average score of quiz.
- Average score of each question.
- Correctness rate of each question.
- Distribution of selected choices of each question.

If there is data that may be omitted or you believe it is not available with the information provided above, please let me know before I kick start with the code.