The following tables have been created in SQL:

1. Questions
   * id = primary key
   * type = question type (short answer, multiple choice, multiple answer, or true/false)
   * answer = answer to the question (will be the letter for multiple choice, letters in alphabetical order for multiple answer, t or f for true/false, and the phrase for short answer)
   * a = multiple choice or multiple answer a option (otherwise should be blank)
   * b = multiple choice or multiple answer b option (otherwise should be blank)
   * c = multiple choice or multiple answer c option (otherwise should be blank)
   * d = multiple choice or multiple answer d option (otherwise should be blank)
2. Results
   * id = primary key
   * grade = decimal out of 1 for questions gotten right
   * link = auto-generated key link
   * Foreign keys:
     1. Quizzes\_id = quiz taken
     2. Employers\_id = employer who created the quiz
     3. Candidates\_id = candidate who took quiz
3. Quizzes
   * id = primary key
   * title = title of quiz
   * time = length of time to take quiz
   * Foreign keys:
     1. Employers\_id = id of employer who made quiz
4. Employers
   * id = primary key
   * name = name of employer
   * email = email of employer
   * company = company employer works at
   * password\_hash = password of employer put through a hashmap to store in order to make more secure
5. Candidates
   * id = primary key
   * name = name of candidate
   * email = email of candidate

**Note: See DDL.sql and DML.sql for all SQL queries**

**Setup queries to create all tables in database**

* See DDL.sql

**Employer should be able to CRUD an account and profile and login**

* Will need 4 REST endpoints
  + post for create
  + get for retrieve
  + put for update
  + delete for delete

**Variable number of questions, True/False, multiple choice, check all that apply, free-form answers**

Questions table accommodates all types. All questions have a primary key id. They have a type to specify the type. They have an answer that accommodates all question types. This answer will be the letter for multiple choice, the string of letters in alphabetical order for multiple answer, t or f for true/false, and the answer string for short answer. The a, b, c, and d parameters will be empty for t/f and short answer questions and store those options for multiple answer and multiple choice questions.

**For the employer, to create a quiz and then email a unique key link that authorizes a person to take the quiz. The app should auto-email the key link to the candidate.**

**User has configurable amount of time to complete the quiz after clicking the key link and confirming the start of the quiz.**

* Creating a quiz requires the following one endpoint (post)
  + Query to create a quiz
  + Query to create an individual question
* Emailing the key link requires a post endpoint to post the results entry
  + create a candidate (if candidate not in database)
  + select a candidate by email (to choose candidate if in database)
  + Create a results entry in Results table with the key link and foreign keys linking it to employer, candidate, and quiz

**For candidates, given the key link, allows them to take the timed quiz.**

* Query to update the results entry with the grade the student received (PUT endpoint)

**An email should be sent to the employer after a candidate has completed a quiz.**

* Query to get employer email by using employer\_id foreign key associated with quiz result (get endpoint)

**For the employer, to see the quiz respondents ranked, and also individual results and stats.**

* Select results by quiz and sort based on grade (get endpoint)
* Select results by individual (get endpoint)