Gaurav Sheni & Grant McGovern
Computer Science 321
Database Project
December 11, 2015

CareerHack

Background

People who are interviewing for tech companies generally rely on Glassdoor to know what questions they could be asked. Glassdoor allows anonymous users to post their application experience. People write about what the interviews were like, the difficulty of the interviews, and any questions they were asked. While this is a great resource, there a few flaws. People cannot see what specific person at the company asked the question, and their role. Many times, the interview questions depend heavily on who is conducting the interview. Another flaw is that answers cannot be posted. Users are unable to post their solutions to the questions. Our aim with this project was to solve this problem.

CareerHack is a resource for people who are interviewing in tech companies. This website allows job hunters to find what questions they could be asked when they are interviewing. They can not only see the questions that have been asked before but also see by what company they were asked by, what specific person asked them the question, the role of that person at the company, possible answers to the question they asked, and when the question was asked. They can figure out the question they might be asked selecting the company name, or topic, or job type.

Furthermore, CareerHack allows people to anonymously post questions they have been asked. After specifying the necessary details, it becomes part of the CareerHack database. The anonymity encourages users to questions without fear of prosecution.

Setup

All instructions to setup the website to run locally can be found in README.md or README.pdf file. The website is also live at ttps://careerhack.herokuapp.com/

Database

A database was used because there is a great deal of information. It allows us to store a great deal of efficiently. Information from the database can be retrieved quickly, and easily. If new data needs to be added, or old data needs to be deleted, that is also done quickly. Our database also allows more than one person needs to access the database at the same time. These benefits, along with numerous others makes it ideal for CareerHack to use a database.

We elected to use a PostgreSQL database primarily because of the compatibility/ease of setup on Heroku. All native Heroku database instances (ones which run on the same dyno/server as the application) automatically default to PostgreSQL. This, coupled with our previous familiarity with PostgreSQL, and some added benefits like shortcut commands (e.g. "\dt;" to list all tables) made it an ideal choice.

To create this database tables, we used PostgreSQL commands entered in terminal in a Unix environment. We created all 6 tables with Companies, Interviewers, Questions, Answers, JobTypes, Questions, and Topics. One such example can be seen below.

```
CREATE TABLE ANSWERS(
        ANSWERID              SERIAL PRIMARY KEY              NOT NULL,
        ANSWER                TEXT                           NOT NULL,
        QUESTIONID   INT                      references QUESTIONS(QUESTIONID)
                                                                    NOT NULL,
        CREATED               TIMESTAMP                      DEFAULT NOW(),
        UPDATED               TIMESTAMP                      DEFAULT NOW()
);
```
It can be seen the primary key is set to be the AnswerID. Furthermore, the AnswerID will be auto incremented upon each new entry, starting with zero. This is seen by the Serial constraint. In addition, some attributes are not allowed to be null, such as Answer, QuestionID. The timestamps for the columns Created and Updated will be the time the entry is inserted. All of these create Table commands can be found in the PostgreSQL_Commands.txt file.

To insert data, commands were again entered through terminal. For example, a insert command for the Answers table is found below.

```
INSERT INTO ANSWERS(ANSWER, QUESTIONID)
VALUES
('A circular manhole prevents one from falling into the hole', 7);
```

In this command, the AnswerID does not need to specified since it is going to auto increment due to the SERIAL constraint given during the creation of the table. Furthermore, the Created or Updated values do not need to be give because they will be the time at which the data was inserted. All of these INSERT commands can be found in the PostgreSQL_Commands.txt file.

In order to retrieve data to fill the website page, commands were tested through terminal and then placed in the app.py file to be run when the user clicked a link. For example, the following command was used to find the interview questions and answers at a company.

'SELECT * FROM "COMPANIES" INNER JOIN (
        SELECT * FROM "INTERVIEWERS" INNER JOIN (
            SELECT * FROM "QUESTIONS" LEFT OUTER JOIN (
                SELECT * FROM "ANSWERS")
            AS JOINC USING ("QUESTIONID")) AS JOINB USING ("INTERVIEWERID"))
        AS JOINA USING ("COMPANYID") WHERE "NAME"=\'%S\';' % COMPANY_NAME

The '\%S\' is where the name of the company the user selected would go. This query uses inner join and left outer joins to gather all the necessary information about a company, and the interview questions they ask, as well as who asked them. If there is an answer in the database, the query will return it. If not, then it will be blank. Both of this is possible because of the left outer join of question with answers.

ER Diagram

The ER diagram of the database below shows the relationships between Companies, Interviewers, Questions, Answers, JobTypes, and Topics. These relationships required that some inferences be made about.

- A company will have 1 interviewer or more.
- An interviewer, if in the database, will belong to 1 and only company.
- An interview will ask 1 or more questions.
- A questions belongs to 1 interviewer.
- A question may have 0 or many answers.
- A answer will have 1 question.
- A question will have 1 job type, and 1 topic.
- A job type or topic may belong to 1 or multiple questions.
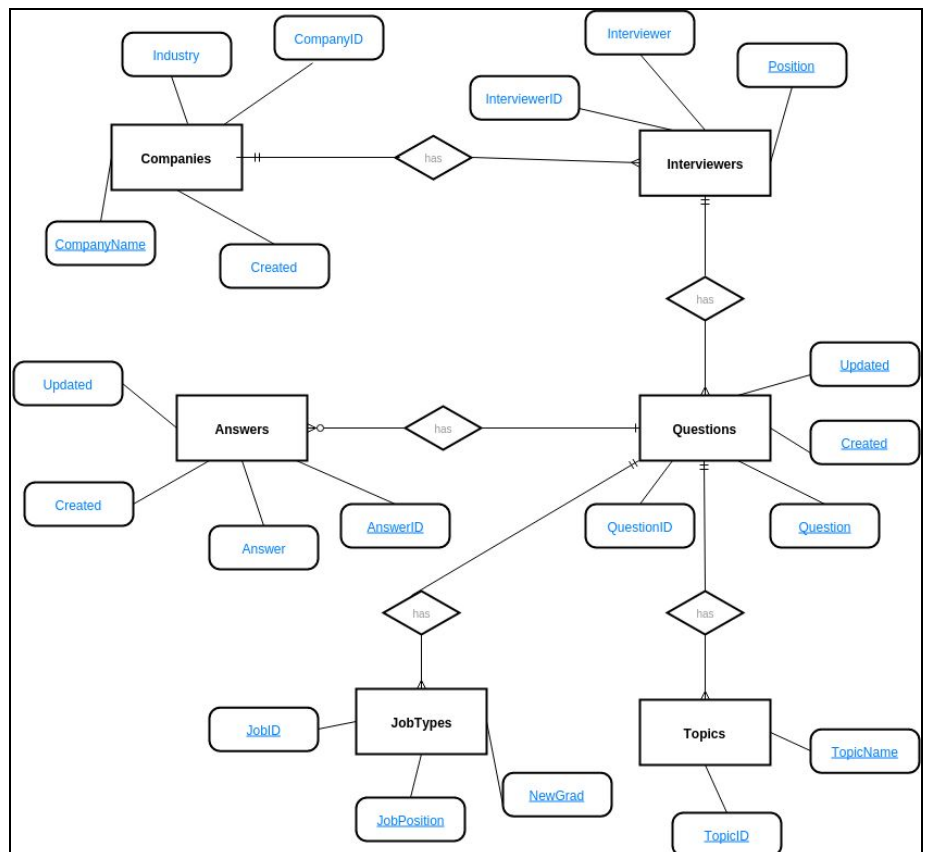
Database Schema

The ER diagram was then converted to a database schema seen below.
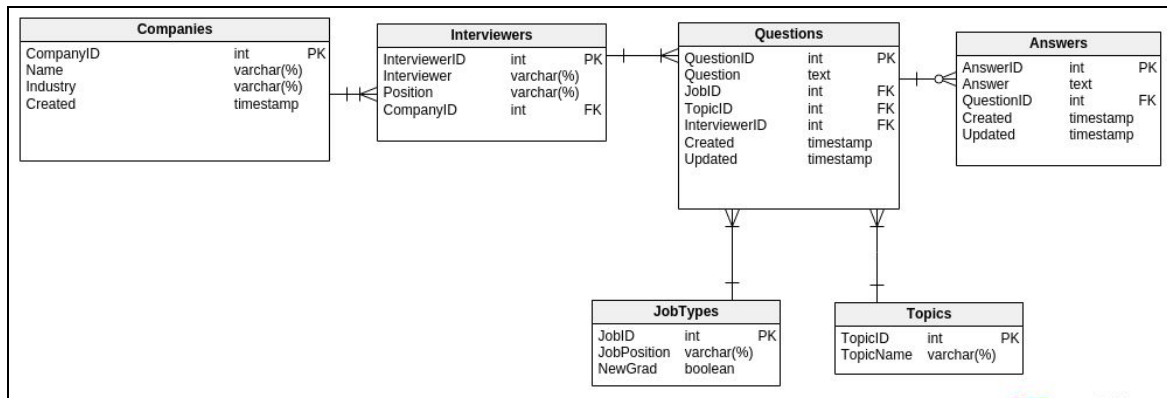
Access

This website will be open to all users. Any person who is browing the Internet can simply go to careerhack.herokuapp.com to see CareerHack. It is meant for anyone who is looking for a job at tech companies but works for internships as well. When a user clicks the companies present on the website, they will also be given the option to add questions to the database. They will need to specify the necessary information, but once they do, it will be added to the database.

Website Screenshots
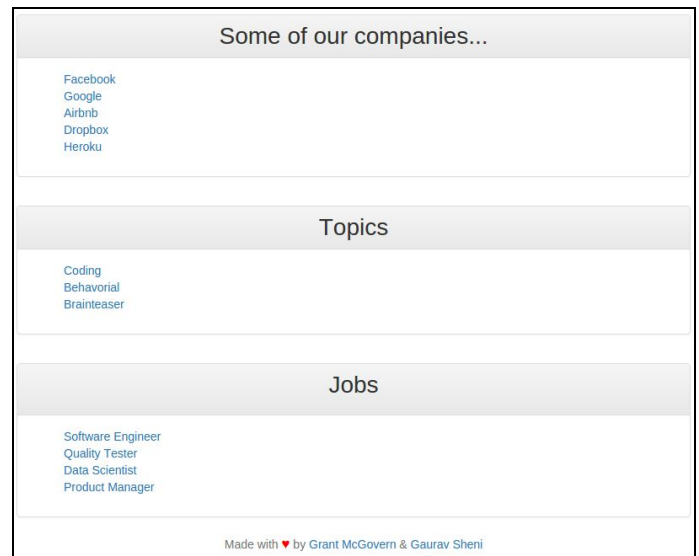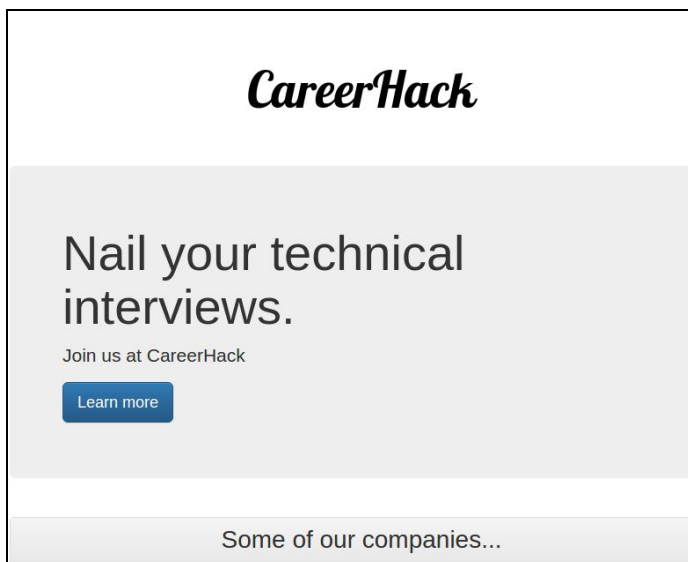
Google

Given a list of words, return the pairs of words that when concatenated form a palindrome in linear time.

Asked by Gaurav Sheni

The idea is to use bitwise operators for a solution that is O(n) time and uses O(1) extra space. The solution is not easy like other XOR based solutions, because all elements appear odd number of times here.

Add Question +

Coding

Given a list of words, return the pairs of words that when concatenated form a palindrome in linear time.

The idea is to use bitwise operators for a solution that is O(n) time and uses O(1) extra space. The solution is not easy like other XOR based solutions, because all elements appear odd number of times here.

Asked by Gaurav Sheni at Google who is a Software Engineer

What is the JRE and what does it have to do with .jar files?

JRE is the environment within which the virtual machine runs. Java Runtime Environment.

Asked by Teddy Beachley at Heroku who is a Software Engineer

Division of Work

This project allowed both of us to a great deal. We learned how to design a database, implement it, and make queries to the database. Grant was responsible for creating the website with Flask, setting up the Heroku to host PostgreSQLdatabase controller, and setting up PostgreSQL to interface with Flask. Gaurav was responsible for designing the database, and determining the necessary queries for the website.