Group Project

Co-op Database System

CSI 2132 A

Professor Ghada Badr

April 1, 2017

Group7_A01

Qi Ye (7151574)
Mengyun Zheng (7609162)
Geling Hu (7463933)
Tianyu Shi (7875409)
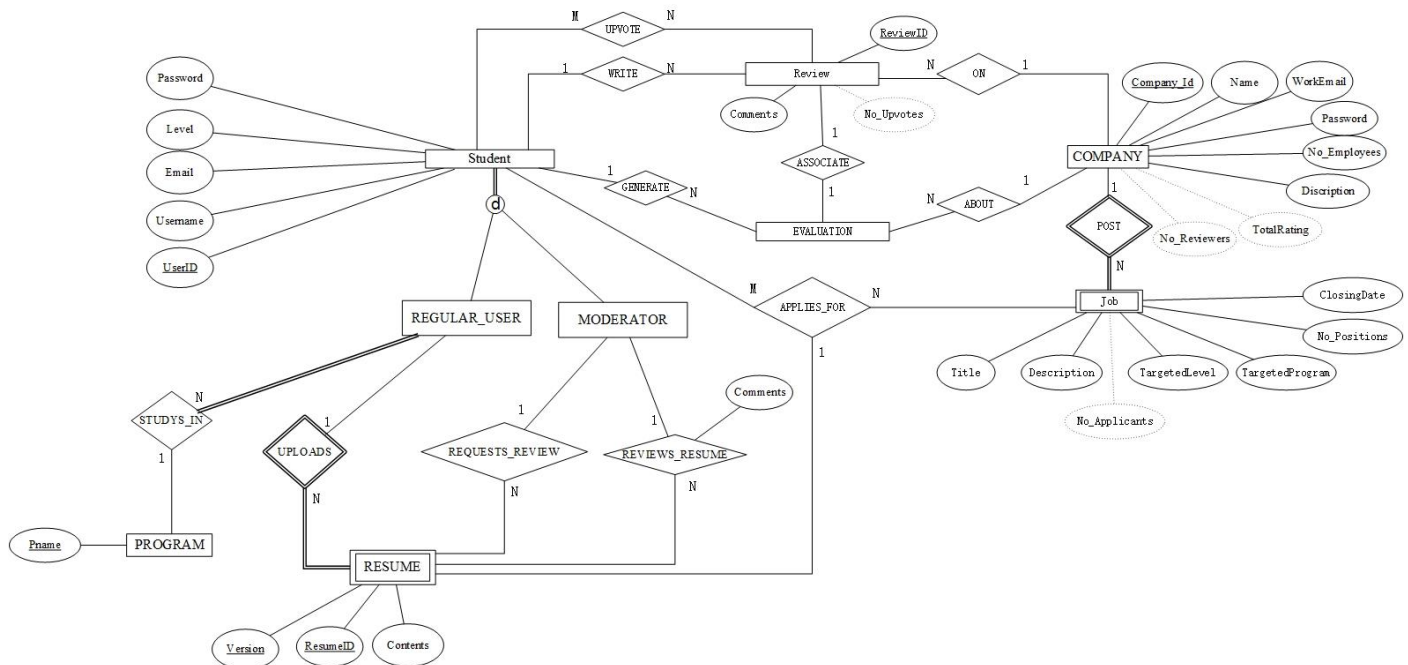
Table of Contents

1. Summary of Work

1.1 ER/ERR Diagram

Figure of ER/ERR Diagram:

Changes from Step 1 and Step 2:

In previous steps, we did not allow entity Admin to have the functionality that regular users have, and we limited its only responsibility is to managing this database itself.

In this step, we decided to delete entity Admin from this database system. We will play the role of admin (i.e. managing this database) when we operate it, but entity Admin is not existed in this diagram.

1.2 Relational Scheme

Figure of Relational Scheme:
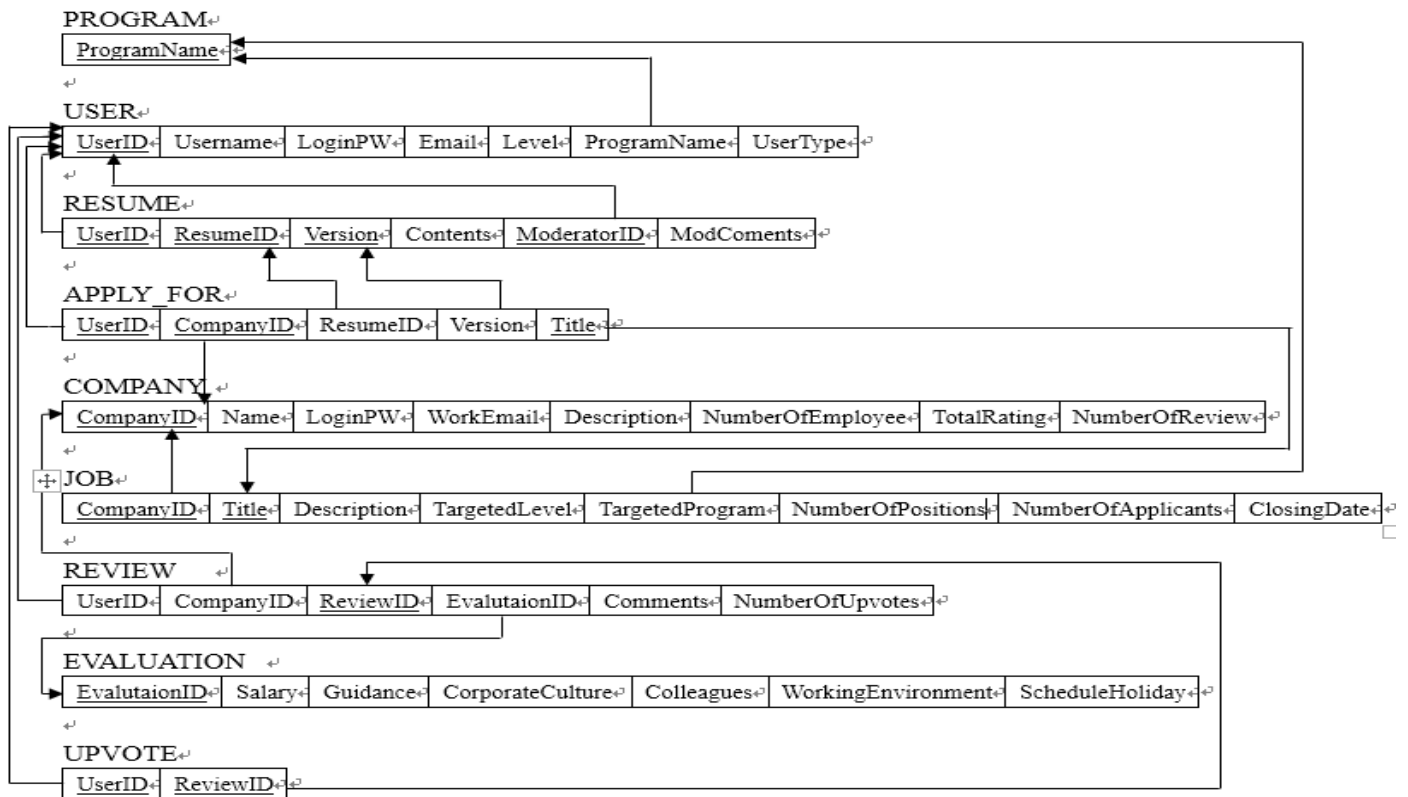
Changes from Step 1 and Step 2:

No changes are made for the relational schema.

1.3 Primary Keys

Please see attached SQL script for tables and primary keys.
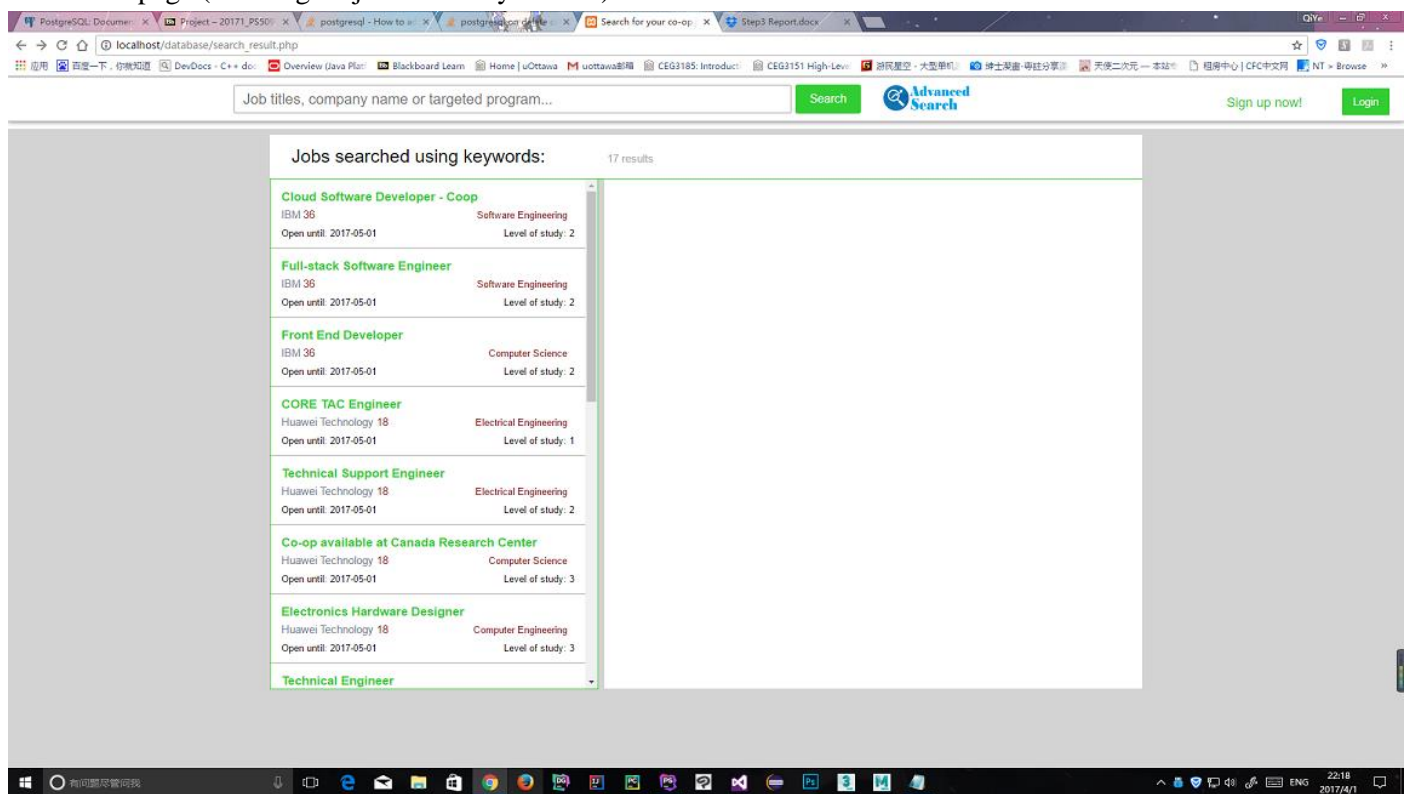
1.4 Added Queries

1.5 Added Triggers and Assertions

None in this step.

1.6 Snapshots of User Interface

1. Index page (showing all job results by default)

2. Sign-up and login pages:

3. Search page with login user:

4. Writing a review for a company:

5. A student can update their resumes:

6.   A company can edit a job:

7.   A company can view the resumes of applicants:

8. Moderator can view and comment on the resumes that has been requested upon:

More functionalities will be demonstrated on Monday lab.

1.7 Testing Results

The data used for testing are as follows:

Students (including 2 moderators and others as regular users):

userid	email	username	password	study_level	program	usertype
23	site@site.ca	site	site1234	2	Software Engineering	R
17	nanaya2024@gmail.com	oppai	qwe1234	4	Electrical Engineering	R
18	johnsmith@asd.com	jsmith	123456	1	Civil Engineering	R
19	what@baidu.com	whoever	who1234	2	Mechanical Engineering	R
20	soft@gmail.com	softguy	soft1234	3	Software Engineering	R
21	bm@hotmail.com	bmeniac	bm1234	4	Biomedical Mechanical Engineering	R
22	cb@sina.com	cby	cb1234	1	Chemical and Biological Engineering	R
15	qye079@uottawa.ca	qirito	qi1234	3	Computer Engineering	M
16	yaeqarts@gamil.com	nanaya	asd1234	2	Computer Science	M

Companies:

companyid	cname	workemail	password	numberofemployees	totalrating	numberofreviewwer
6	Synopsys	synop@sy.com	synop1234	10000	0	0
7	Thales	thales@tha.ca	thales1234	10000	0	0
5	Huawei Technology	huawei@huawei.cn	huawei1234	10000	18	1
9	intel	intel@intel.ca	intel1234	20000	0	0
4	IBM	ibm@ibm.com	ibm1234	10000	57	3

Jobs:

companyid	title	jobdescription	targetedprogram	targetedlevel	numberofpositions	closingdate	numberofapplicants
4	Cloud Software Developer - Coop	IBM is a globally integrated technology and consulting company headquartered in A…	Software Engineering	2	2	2017-05-01	0
4	Full-stack Software Engineer	IBM is a globally integrated technology and consulting company headquartered in A…	Software Engineering	2	1	2017-05-01	0
4	Front End Developer	IBM is a globally integrated technology and consulting company headquartered in A…	Computer Science	2	5	2017-05-01	1
5	CORE TAC Engineer	Co-Op Placement: CORE TAC Engineer◊Ref #: 577◊Division: Infrastructure◊Location: …	Electrical Engineering	1	5	2017-05-01	0
5	Technical Support Engineer	Co-Op Placement Technical Support Engineer: UMTS,LTE RAN◊Ref #: 578◊Division: Inf…	Electrical Engineering	2	2	2017-05-01	0
5	Co-op available at Canada Research Center	Full-time and Co-op/intern positions available at Canada Research Center at Markh…	Computer Science	3	5	2017-05-01	0
6	Electronics Hardware Designer	Electronics Hardware Designer◊Ref #: 763◊Division: R&D◊Location: Waterloo, Ontari…	Computer Engineering	3	5	2017-05-01	0
6	Technical Engineer	Business Title Intern (Technical Engineering)◊Requisition Number 11917BR◊Hiring L…	Computer Engineering	2	1	2017-05-01	0
6	GUI Automation Coop	Business Title GUI Automation Intern◊Requisition Number 13062BR◊Hiring Location(s…	Computer Engineering	2	3	2017-05-01	0
4	Back-end Developer	IBM is a globally integrated technology and consulting company headquartered in A…	Software Engineering	3	2	2017-05-01	0
4	Compiler Optimization Developer	Company Description◊null Job Description◊◊IBM is a globally integrated technology…	Software Engineering	3	1	2017-05-01	1
9	Intel Job Posting Test	Description of Job From Intel	Electrical Engineering	2	10	2018-12-31	0
4	DB2 Software Innovator	IBM is a globally integrated technology and consulting company headquartered in A…	Computer Science	3	4	2017-05-01	2
6	Coop Software Developer	Business Title Intermediate Software Developer◊Requisition Number 13057BR◊Hiring …	Computer Science	3	1	2017-05-01	0
7	Site Deployment	Thales Canada, Transportation Solutions (Thales) is one of the world's leading su…	Electrical Engineering	3	1	2017-05-01	0
7	Software Tester Co op	Thales in Canada has 1,800 employees in Quebec City, Montreal, Ottawa, Toronto an…	Software Engineering	1	2	2017-05-01	0
4	Test update	updated?	Electrical Engineering	2	3	2018-06-30	0

These 3 types are the main data set to begin with to test the functionalities of the database. Most queries used in this project are provided later in this report and they all function correctly, and details will be demonstrated in the lab. Below will provide a few of the results by some queries. You can test with my codes to verify the functionality of this project.

Note: to test my codes, be sure to always start from search_page.php. This is the default index page for this project.

1.8 Results of Assertions

There's no assertions in our project, only triggers that will update the values of certain columns in some of the tables. The result of working triggers will be demonstrated in the lab.

2. Design of User Interface

2.1 Initial Design

Figure of Initial Design (from Step 2):

Upload Resume
Or Post a job for Employer after signing in)

User sign in   Employer Sign in
User sign in for regular user,
moderator and Admin

Keyword Search Bar

Search Filter
(By job titles, Company name, etc)

List of Jobs searched

Description of selected Job and
Company Information

(Note: we generally followed the layout of www.glassdoor.ca)

2.2 Implementation of User Interface

All of the following functionality are achieved by GUI of websites.

(1) Sign In / Sign Up

When a new user signs up, simply click the "sign in" or "sign up" button on the top-right corner of the index page. Then he/she will be asked to enter username, email, password, level and program. Among them, username and email must be unique. When an existing user signs in, a correct combination of his username/email and password must match. This is also true for company(Employer) to signs up and in.

(2) Search Jobs
The user can search for jobs by keywords in job title or company name, or choose to display the result by ascending or descending order by the jobs' number of applicants (popularity), or by company's average rating.

(3) Upload Resume
Users can upload resumes online by clicking the button on the top-left corner, and will be able to see the details of resumes in his/her account. After uploading the resume, a review request on a certain resume can be made, and a moderator is randomly assigned to this resume, and the assigned moderator will have temporary access to the content of this resume. After the moderator finishes review, the moderator will no longer be able to access this resume.

(4) Apply Jobs
Users can also apply for jobs using their uploaded resume. Then the company that posted this job will be able to access this resume.

(5) Write Review
Users can write a review for a company by simply searching for the company, get the result and click a button called "write review".

(6) Edit User Profile
In Users' account, users will be able to edit his email, password, or username. He can also edit a review within 30 days of its posting, and delete a review at any time, all just by clicking on the corresponding buttons and enter required information. Companies share the similar GUI to manage its own information and job postings.

(7) Review and comment on resumes
Moderators can have access to the contents of resumes submitted for review. The moderator can only add comments on the resume, but cannot change the original contents. After the moderator has finished the review, a newer version of the resume with comments will be generated for the author.

3. Description and Implementation of All Queries
All the queries are declared as function across many of PHP files. The source of all the queries will be given below. To see the full implementation of those functions, please refer to the given PHP files.

```
-- A new Student signs up; violation of unique key constraint will be handled by PHP
script (within function declared in signup.php)
INSERT INTO student(Email, username, password, level, program) VALUES (...);

-- A student or company can update their profiles(Both in function declared in
profile.php)
UPDATE student
SET username=$q_name, password=$q_password, study_level=$num, program=$q_program
```

```
WHERE userid = $id;
-- A similar query can follow for Company updating its Company Name/work email/password

-- A student can upload a resume; This UserID will be acquired from PHP session, and
-- contents will be text input from User (within function in upload.php)
INSERT INTO resume(userid, contents) VALUES ([this UserID], [Contents]);

-- A student can apply for a job that has the same level as student with an uploaded
resume.
-- All the values to insert are from websites with GUI.
-- A trigger will check if the user's level is the same as the targeted level of this job.
(within function declared in apply_job.php)
INSERT INTO applyfor(UserID, comid, jobtitle, resumeid, resumever) VALUES
  ([this UserID], [Company's ID], [job title], [resume's id], [resume's version']);

-- A student can post review associated with an evaluation for a company.
-- Evaluation ID will be from the last added record in Evaluation, because an insert into
Review always follows
-- an insert into Evaluation in our design.
(within function declared in submit_review.php)
INSERT INTO evaluation(salary, guidance, we, culture, SandH, Colleagues) VALUES (...);
INSERT INTO review(UserID, companyid, EvaID, Comments) VALUES
  ([This UserID], [Company's ID], [Evaluation' ID], [Comments]);

-- A student can upvote a Review (only once)
(within function declared in upvote.php)
INSERT INTO upvote(userid, reviewid) VALUES([This UserID], [Reviews ID]);

-- A student can make a reviwe request for one of his/her resumes to a moderator
(within function declared in resume_disposal.php)
UPDATE resume
SET ModeratorID = [the ID of Moderator who is randomly assigned];
WHERE UserId = [this Userid];

-- A student can edit his review for a Company
(within function declared in review_disposal.php)
UPDATE Review
SET Comments = [newly typed comments from user]
WHERE UserId = [this Userid] and CompanyId = [this Company ID];

-- A student can delete his review for a Company at any time
(within function declared in review_disposal.php)
DELETE FROM review WHERE review=[the review's ID];

-- A moderator can make comments on a resume that has requested review
```

```
(within function declared in resume_review_disposal.php)
UPDATE Resume
SET comments = [added comments from moderator]
WHERE UserId = [the resumes UserID] AND ResumeId = [The resumes ID] AND version = [The
resumes current version];


-- A company can post or delete a job
(within function declared in upload.php)
INSERT INTO Job(CompanyId, title, description, targetedProgram, targetedLevel,
numberOfPositions, ClosingDate) VALUES(...)
DELETE FROM Job WHERE CompanyId = [this Company Id] AND Title = [This jobs title];


-- Student can search for a job using keywords for job title, company name, or targeted
program.
■  We've also added a filter functionality to allow user to select jobs according to its
   popularity (number of applicants) or total rating of the company. Details will be
   demonstrated in the lab.
(within function declared in search_result.php)
SELECT Title FROM Job WHERE Title[or targeted level or program] LIKE '%' || [search
keywords] || '%';
```

These are the main queries that will be implemented. Some queries with the same type are not listed here.

4. Description and Implementation of All Triggers and Assertions

```
-- Trigger when a student upvotes a review, the number of upvotes for this review
-- is incremented by 1 using function increment_no_upvote()
CREATE TRIGGER Upvoted
    AFTER INSERT ON upvote
    FOR EACH ROW
    EXECUTE PROCEDURE increment_no_upvote();

create function "CoopDatabase".increment_no_upvote() returns trigger
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE review
    SET numberofupvotes = numberofupvotes + 1
    WHERE reviewid = NEW.reviewid;
    return NEW;
END;
$$;



-- Trigger after a new review is posted for a Company: the number of reviewers for this
company is incremented by 1, and the total rating is incremented by the rating of new
evaluation.
```

```
CREATE TRIGGER reviewed
    AFTER INSERT ON review
    FOR EACH ROW
    EXECUTE PROCEDURE increment_no_review_and_rating();


CREATE OR REPLACE FUNCTION update_no_review_and_rating () RETURNS trigger
LANGUAGE plpgsql
AS $$
DECLARE
    newRating INTEGER;
BEGIN
    -- Increment number of reviews
    UPDATE company
    SET numberofreviewwer = numberofreviewwer + 1
    WHERE companyid = NEW.companyid;
    -- Add total ratings to this company
    SELECT (e.salary+e.guidance+e.we+e.culture+e.SandH+e.Colleagues) INTO newRating
    FROM evaluation e
    WHERE NEW.evaid = e.evaluationid;

    UPDATE company
    SET totalrating = totalrating + newRating
    WHERE companyid = NEW.companyid;
    RETURN NEW;
END;
$$


-- Trigger after a student has applied for a job. The number of applicants of this job
will be incremented by 1.
CREATE TRIGGER applied
    AFTER INSERT ON applyfor
    FOR EACH ROW
    EXECUTE PROCEDURE increment_no_applicants();


create OR REPLACE function "CoopDatabase".increment_no_applicants() returns trigger
LANGUAGE plpgsql
AS $$
BEGIN
    UPDATE job
    SET numberofapplicants = numberofapplicants + 1
    WHERE companyid = NEW.comid AND title = NEW.jobtitle;
  RETURN NEW;
  END;
$$;
```

14

Assertions and Triggers that check if the student has the same level with the job has been removed, because we think it makes more sense for students to be able to apply for all jobs, and it's up to the company if they will hire the student with different levels.

5. Distribution Table

Student Name	Student Number	Work
Qi Ye	7151574	Code related
Mengyun Zheng	7609162	Prepared final report, Chart drawing
Geling Hu	7463933	Prepared final report, Chart drawing
Tianyu Shi	7875409	Prepared final report, Chart drawing

6. Reference

1. Website: *glassdoor*. Available at www.glassdoor.ca. [Retrieved at April 1, 2017]