

CSI 2132
Feb. 19, 2017

Step 1 – Project Selection and Assumptions

Group Name: Hao's Group

Muhammed Owais Bawany - 7993647– Lab B03

Hao Luo – 8310767 – B02

Rupsi Kaushik – 8199148 – Lab B01

Sriteja Srivatsavai – 8138963 – Lab B03

1. **Project Title:** Co-op Database System
2. This Co-op database system will have jobs, companies, regarding specific positions which can be reviewed by students and would cater other faculty of engineering students to know about other students interview experience, on-the-job experience as well as salary.
3. **Entities:** User, Moderator, Company, Jobs, Program, Resume. ResumeReview. Upvote

Attributes: (Underline = Primary Key, Italic = Foreign Key)

User - UserID, FName, Minit, Lname, email, level, *ProgramID*,

Moderator - ModeratorID, NumberOfResumesReviewed

Company - CompanyID, CompanyName, Address, Phone, Email, rating, NumberOfEmployees

Jobs - JobID, positionName, job_description, requiredStudentLevel, *CompanyID*, NumberOfPositions, job_level

Program - ProgramID, ProgramName

Resume - ResumeID, *UserID*, VersionID, ResumeDescription

ResumeReview - ReviewID, *ModeratorID*, *ResumeID*, Comments

Upvote - UpvoteID, *UserID*, *ReviewID*, UpvoteRating

Relations:

Users enrolls program N:1

User sends request to moderator to review resume 1:1

Moderator reviews resumes 1:N

User rates a review is useful or not N/1:N

Company offers multiple jobs 1:N

A user applies for multiple jobs 1:N

4. **Assumptions:**

- One account per email address
- There a date for the job posting and a closing date
- Can order the job posting by date, popularity, location, etc
- A company can have multiple job posting
- A student can apply for any number of jobs
- A company must be verified
- Moderator can review multiple resumes
- The user can delete old resumes
- If the user has more than 20 resumes, then the least recent resume will be deleted

- There will be a recent history for jobs you have recently visited
- Jobs will be marked with seen/applied if you have already visited the job posting
- We assume multiple users enroll in one program.
- A user sends a request to a specific moderator to review their resume.
- One user can only have one moderator review their resume but one moderator can review multiple resumes.
- User can rate multiple reviews as either useful or not and one review can be rated by multiple user too.

5. Possible SQL DML:

Deleting information

- ? Users Resume
 - delete resume_id From Resume where Resume.resume_id = "X"
- ? Users Comment
 - Delete review_id From Upvote where Upvote.review_id = "X"
- ? Company job posting
 - Delete job.id from Jobs where job.id = "X"
- ? Resume review request
 - delete resume_id From ResumeReviewRequest, User where user.resume_id = ResumeReviewRequest.resume_id

Modify

- User's personal details (name, email, level, program, etc)
 - Update User, set User.name(or email, level, program, etc) = " ", where user.id = " "
- ? Company(level, number of positions, company information, job description)
 - Update Company set Company.number_of_Employee where Company_id = " "
- ? Update Job
 - Update Job set Job.level(name, description,etc) where Job.job_id = " "
- ? User resume
 - Update Resume set Resume.resume = "X.doc" where Resume.userId in (Select User.resumeid from User, Resume where User.resume = Resume.resumeid)
- ? User comment/vote

- Update Upvote set upvote.comment where Upvote.userid in (select user.userid from User, Upvote where User.userid = upvote.userid

Select

- ? Shows number of job postings per page based of job level sorted by the company address
 - Select J.JobID, C.Address from Jobs as J, Company as C where C.CompanyID = J.CompanyID and j.rank = "X"

Insertion

- ? Inserting a new resume into the resume table
 - Insert into Resume (ResumeID, UserID, VersionID, ResumeDescription) Values (.....)
- ? Inserting a new job posting
 - Insert into Jobs (JobID, positionName, job_description, requiredStudentLevel) Values (....)

6. Triggers/Assertions:

Assertions (Cannot be executed in PostGRE SQL)

- ? Ensures 20 resumes at a time
 - o Create Assertion Restrict20Resume check (select count(*) from Resume) < 20)
- ? Ensures that a comment is only upvoted once by the same user
 - o Create Assertion RestrictComment check(select count(UserId) from Upvote) <= 1)

Triggers

- ? Versioned history of resume uploads of user- track the evolution of their resumes (modify)
 - o Create Trigger New_Resume Before Insert or Update of * on Resume Execute procedure AddtoHistory (New.Resume.*)
- ? Admin notification for job submit
 - o Create Trigger NotifyAdmin After insert Apply for each row when (Old.StudentID is Distinct from New.Studentid) Execute procedure NotifynewApplication(new.id)

- ❓ New job posting notification for students
 - Create Trigger NotifyStudent
After insert Job for each row when (old.JobId is distinct from old.JobId) Execute procedure NotifyNewJob(new.JobId)
- ❓ When a resume is reviewed, deletes the resume review request
 - Create Trigger Reviewed After Insert ResumeReview
For each row when (old.resumeid is distinct from new.resumeid) Execute procedure DeleteResumeReviewRequest(old.resumeid)

7. References:

- [1] "Glassdoor Job Search | Find the Job That Fits Your Life." *Glassdoor*. N.p., n.d. Web. 19 Feb. 2017.
- [2] "Style Guide." *Arctic Anthropology* 39.1/2 (2002): 155-63. Web.
- [3] "If You Know Your Resume Needs a Critique, Here Are Eight Sources of Finding a Resume Reviewer Who Meets Your Needs." *Monster Career Advice*. N.p., n.d. Web. 19 Feb. 2017.