Final Project Milestone 1

Inas El Ouazguiti,
Luis Urdaneta,
Amir Aslamov

Relational Schemas:

Applicant(ID, University, HighestDegreeEarned, Gender, FirstName, LastName, Ethnicity, DisabilityStatus, Major, GPA, Password, Location,Email)

- PK: ID
- Not Null: Email, Password
- Unique: Email

Company(ID, CompanyName, Industry, Size, Location, ServiceType, CompanyType, AnnualRevenue, HREmail, AcceptsOPT/CPT)

- PK: ID
- Unique: HREmail
- Not Null: HREmail

Interview(ID, CompanyID, ApplicantID, Location,MeetingTime)

- PK's: ID, CompanyID, ApplicantID
- FK's: CompanyID(Company), ApplicantID(Applicant)

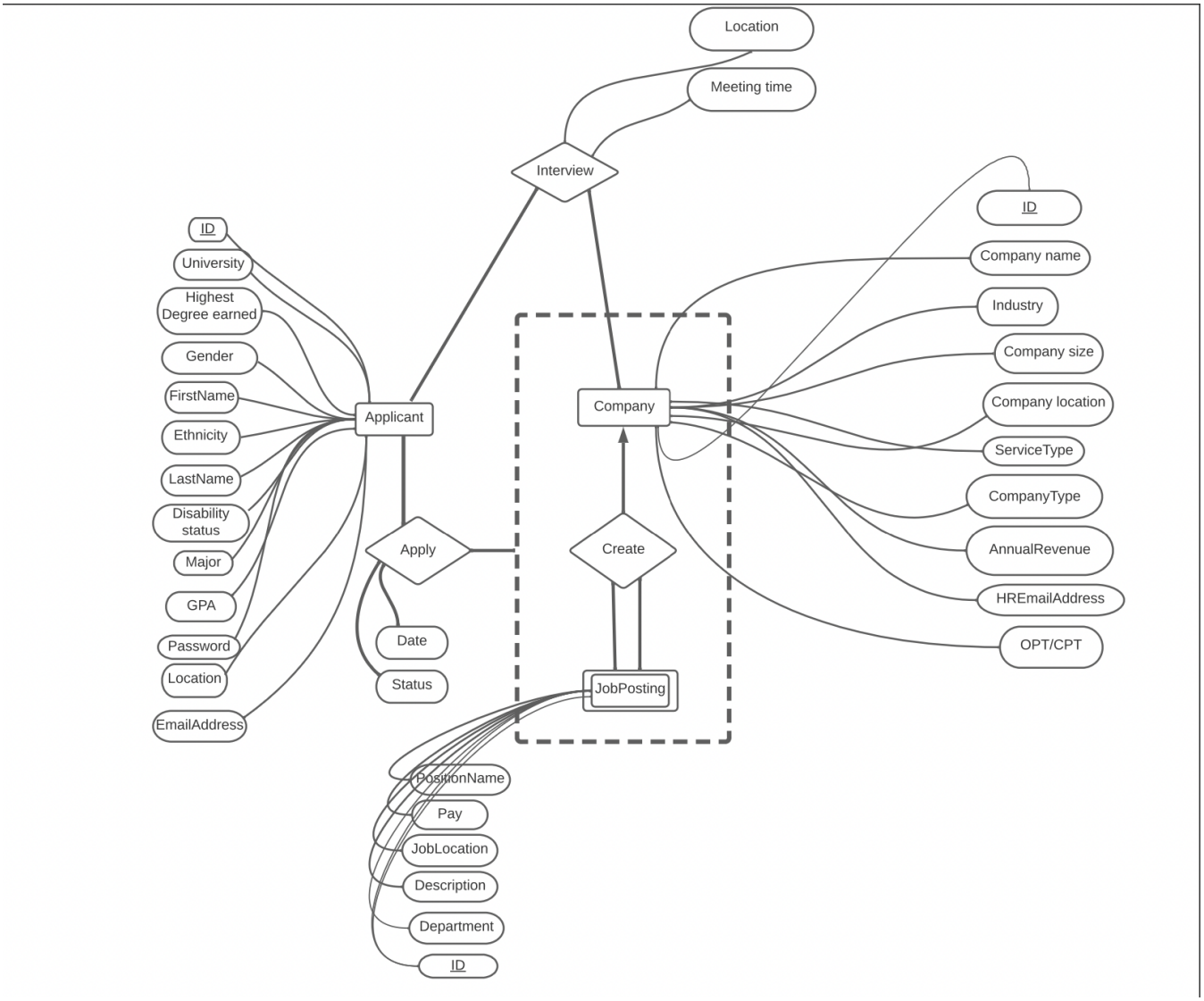
JobPosting(ID, CompanyID, PositionName, Pay, JobLocation, Description, Department)

- PK: ID, CompanyID
- FK: CompanyID(Company)
- Not Null: PositionName, Department

Apply(PostingID, CompanyID, ApplicantID, Date, Status)

- PK's: PostingID, CompanyID, ApplicantID, Date
- FK's: PostingID(JobPositing), CompanyID(Company), ApplicantID(Applicant)
- Not Null: Status

EER Diagram


Location
Meeting time
Interview
ID
University
Highest Degree earned
Gender
FirstName
Ethnicity
LastName
Disability status
Major
GPA
Password
Location
EmailAddress
Applicant
Apply
Date
Status
Company
Create
JobPosting
PositionName
Pay
JobLocation
Description
Department
ID
ID
Company name
Industry
Company size
Company location
ServiceType
CompanyType
AnnualRevenue
HREmailAddress
OPT/CPT

Project Summary

The project we have decided to develop is a web application designed to serve 2 types of users - Applicants and Companies. The idea is based on similar apps like HandShake designed to serve these types of users for the purposes of facilitating the job/internship applications, centralizing and managing such applications and job posting for applicants and companies respectively, helping applicants easily find information about other companies while helping companies have easy overview of applicant's background and sharing their contact information for easy connection. Our application is also designed to help manage status of applications and correspondingly manage creation and management of interviews to be scheduled for the matches among the job/internship posting, applicants and companies.

In this project, we have decided to create separate database entities for Applicant, Company and Job Posting, each with their relevant attributes. For these, we require that email addresses be required, unique and not null for applicants and companies because our application would have authentication for sign up and sign in each time the application is used, for the reasons of personalizing some of the features for these distinct entities - for example, when an applicant signs in, we would show a dashboard with the features of "See the list of companies applied" or "Search job postings" while when a company HR member sign in, the application would display a dashboard with specific features such as "Create a job posting" or "Search applicants in a particular university".

For the Job Posting, we would require that the name of the position and the department entries be not null, because this would be necessary information for the applicants to be able to filter job posting by position name and department.

We have decided that when the company creates a job posting, job posting becomes available for applicants to apply - therefore, we have decided that this relationship of job posting creation be an aggregation and correspondingly linked to applicants to be able to apply for it. As the applicant applies for a job posting created by the company, relevant information such as date and status are also stored, with the constraint that date be unique for each applicant that applies to a job posting created by a company - so, we have decided to include date as part of the primary key to ensure no duplicate applications are stored in our database.

When the status of the application changes to approved, we can turn to the Interview table where we store applicant, and company information as well as location and meeting time to help both parties centrally manage their interviews.

Overall, the project would have 3 distinct entities, 3 relationships and utilize the following types of queries: 1) SELECT - to list all job postings 2) WHERE - to filter job postings 3) ORDER BY - to order job postings or the list of applied companies 4) INSERT - to enter a new job posting 5) UPDATE - to update the status of application 6) DELETE - to delete a job posting after it got filled 7) CREATE TABLE - to create all the necessary table for storage 8)

FROM - to specify which table to retrieve from for the listing of job postings and applied companies