

Group Members: Garrett Buchanan, Griffin Brown, Samuel An, Andy Chung, Carson Brase

Description of Database:

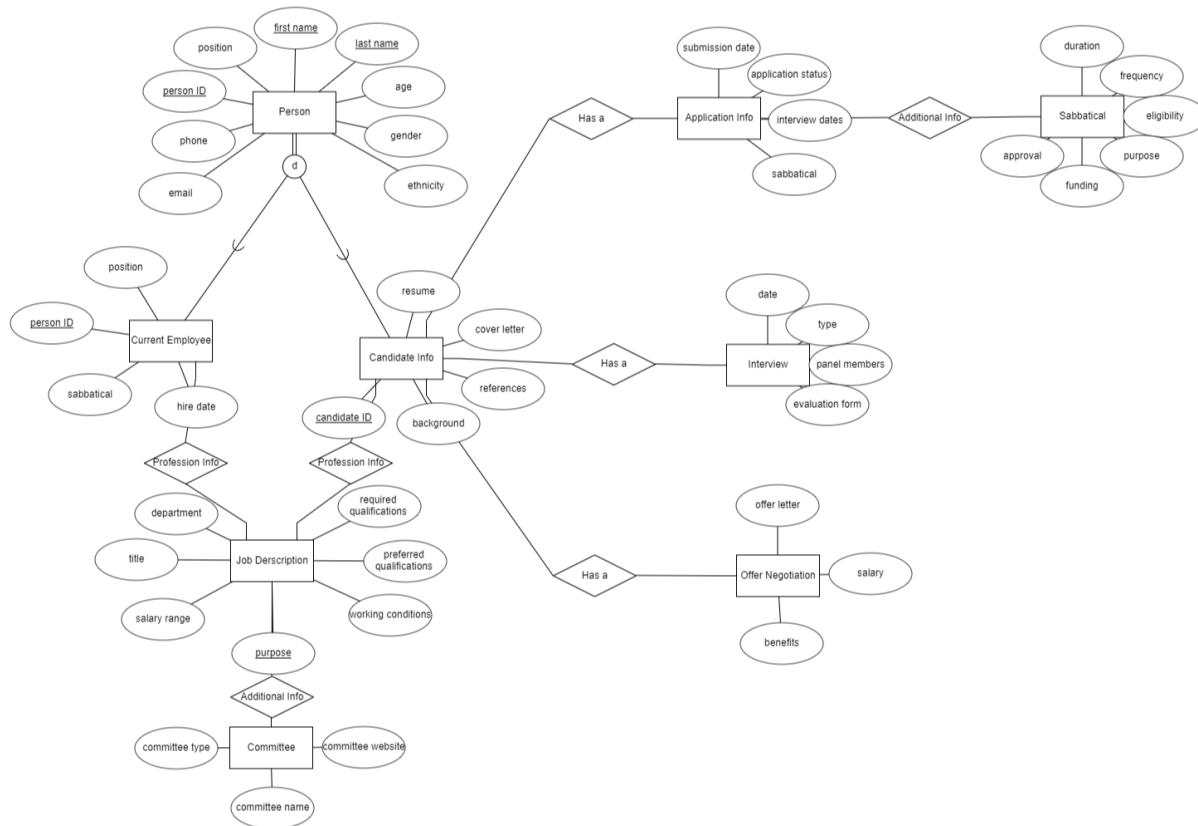
The POD database we created has the overarching goal of simplifying the hiring process for a small college, such as Westmont. The basis for our database is based on the ability to answer simple questions to decipher which candidates are better suited for the open job. In our database, we included several broad entities regarding potential candidates that narrow down into attributes that allow the employer to have more knowledge about the to be employee. The attributes we included begin with a general outline of the person going through or been through the hiring process, then specifies whether the person being researched is a current employee of a hiring candidate. The thought process behind this initial step was to create a broad classification for two main groups of people within the database, giving each individual key attributes that allow for personal identification, access to basic information and contact information. Furthermore, both groups of people, current employees and candidates are required to fill out a description of the job they are currently working or applying for. As an extension of this job description, there is an entity attached to it labeled committee, which places employees that are eligible for a committee into one, as well as finds a potential committee for a candidate moving through the hiring process.

The main function of this database is to simplify the hiring process, which is centered around the candidates applying for open positions. To expand the description for those people who are deemed qualified candidates, there are three separate entities that our database dives into: application information, basics of their interview and offer negotiations. Each of these topics have several attributes that create a clear and concise way to filter through applicants who have completed specific criteria, such as making it past the application process and being accepted for the position after the interview. This database also covers the topic of a potential sabbatical for new candidates, describing what their sabbatical terms will look like, if the candidate was accepted for their applied job. When all of these dates, descriptions and additional bits of information are placed into the database, the Provost of a college would be able to operate this system with clarity because of the simplicity of our outline, on top of the necessary details it addresses.

Challenges or difficult aspects of DB: (gaps in data or relationships, etc)

1. If there are to be multiple rounds of interviews, how would we organize the data of this? (e.g. if an individual had positive feedback after a first interview and passed to the second round, but the second interview had negative feedback and failed.

Entity-Relationship Diagram:



Questions to ask of database:

1. Which current employees are a part of a particular committee?
2. How many candidates have applications accepted by the employer? How many have been through the interview process and offered a job?
3. Which of the current employees work in a specific department, as well as a specific committee?
4. Which panel of interviewers saw the most accepted candidates?
5. How many female applicants were accepted for a given position? How many males?

Flaws within the E/R diagram:

1. Lineage of entities: The application information, interview and offer negotiation entities should have some sort of relation between themselves, since they are all related. Instead of having a lot of null values for applicants who apply but do not have their application accepted, there should be a way to delete them from the table based off of an attribute that describes if the application was accepted.
2. Redundancy: The sabbatical attribute and entity are slightly redundant, and could be condensed. On top of that, there are repeated key attributes in different entities. We deemed those attributes necessary for the database, but there may be ways to limit the number of occurrences throughout the database and diagram.

Examples of non-BCNF, 3NF, or 4NF Relation Schemas:

Our functional dependencies are already combined in the most efficient and effective ways, resulting in no need for decomposition or combination

Expected Queries:

Our schema will expect queries that will ask for information about potential future employees or current faculty and staff. The specifics of these queries can include information about job description, committee assignments, application information, interview status, and offer negotiations. Each of these categories contains multiple subcategories that leads to more precise information retrieval.

Conversions to Relations and Functional Dependencies:

```
Person (  
    firstName,  
    lastName,  
    position,  
    personID,  
    phone,  
    email,  
    age,  
    gender,  
    ethnicity,  
)
```

Functional Dependencies:

- $\text{personID} \rightarrow \{\text{firstName}, \text{lastName}, \text{position}, \text{phone}, \text{email}, \text{age}, \text{gender}, \text{ethnicity}\}$
- $\text{email} \rightarrow \{\text{firstName}, \text{lastName}, \text{position}, \text{personID}, \text{phone}, \text{age}, \text{gender}, \text{ethnicity}\}$
- $\text{Phone} \rightarrow \{\text{firstName}, \text{lastName}, \text{position}, \text{personID}, \text{email}, \text{age}, \text{gender}, \text{ethnicity}\}$

```
JobDescription (  
    title,  
    department,  
    purpose,  
    requiredQualifications,  
    preferredQualifications,  
    salaryRange,  
    workingConditions,  
)
```

Functional Dependencies:

- purpose → {title, department, requiredQualifications, preferredQualifications, salaryRange, workingConditions}

Final Analysis:

The normalization of this database came about when our group determined that the best approach would be to start broad with the personal information of an individual in the database. From there, we would determine whether this person is already employed in a position, or if they are a potential candidate to take over a position. If a candidate, data would then be directed to the information regarding their application, interview, and offer negotiations. If instead a current employee, we would grab the current data regarding the work and position of the individual. Ultimately, whether the individual is a candidate or current employee, the specific job description would be pulled from the database, along with the committee associated with this specific job.