

Description/Motivation of Project: The Faculty Hiring Management System is an integrated database-driven solution designed to streamline and enhance the faculty hiring process at educational institutions. This system aims to provide a centralized platform for managing job openings, candidate applications, offer and rejection letters, and departmental information, all while ensuring compliance with institutional standards and constraints. This system will allow the institution to keep track of candidates and job applications to past and present job openings, which should allow for an easier hiring process going forward as well as a centralized database to create future inferences.

Hiring Process: Review and approve faculty job descriptions, candidate selection, and offer letters to ensure they meet institutional standards and budgetary constraints.

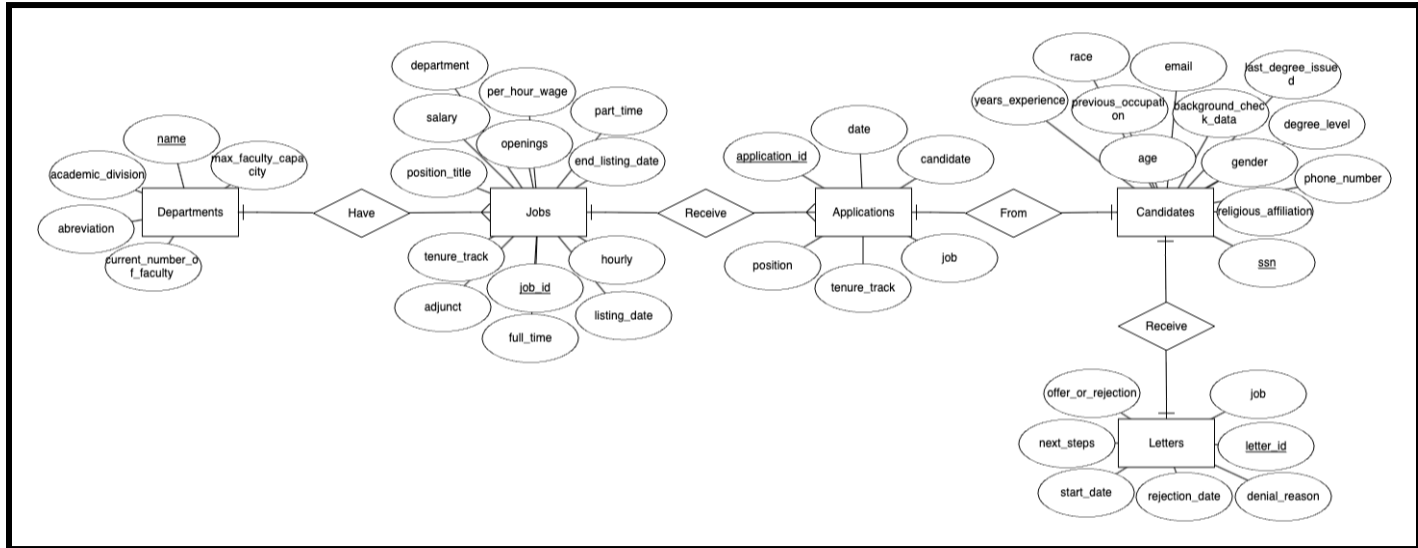
Query for Database Ideas:

- Which departments have the most job openings?
- Which departments have the highest starting salaries?
- Which departments receive the most applications?
- Who has applied for this job and what is their status?
- What attributes historically have increased an applicant's chances at getting a job?

Difficult Aspects and Gaps in Available Data:

- Defining the relationships between entity sets and which sets have a relationship with each other
- Thinking of all the possible relevant attributes of each relation
- Gaps - legal restrictions
 - SSN
 - Background Check

E/R Diagram:



Modified Design Explanation:

There were multiple suggestions given to us and ways we thought of ourselves in how we could improve our er diagram. These include:

- Combining offer and rejection letter entities into one
- Cleaning up the flow of our diagram.
- Renaming relationships
- Removing unnecessary and redundant attributes

This new design allows for a much better flow of one entity into another using accurate logic in how they lead into each other. It becomes much clearer and easier to understand what we are trying to accomplish with this diagram.

Conversion to Relations:

- Candidates (Degree Level, Age, Years of Experience, Previous Occupation, Race, Religious Affiliation, Phone Number, Email, Last Degree Issued, SSN, Gender, Background Check Data)
- Jobs (Job ID, Position Title, Department, Tenure Track, Full Time, Part Time, Adjunct, Hourly, Per Hour Wage, Salary, Listing Date, End Listing Date, Openings)
- Letters (Letter ID, Offer or Rejection, Start Date, Next Steps, Rejection Date, Denial Reason, Job)

- Departments (Name, Abbreviation, Current Number of Faculty, Max Faculty Capacity, Academic Division)
- Applications (Application ID, Job, Position, Tenure Track, Date, Candidate)

Functional Dependencies:

- Candidates
 - ssn →
- Jobs
 - position_title, department →
- Applications_Received
 - Position, person →
- Offer_Letters
 - job_title, start_date →
- Rejection_Letters
 - person, rejection_date →
- Departments
 - name →

***the right side is left blank due to how many attributes are dependent upon each. All the FD's we chose apply to every other attribute in their respective relations.

```
Candidates (  
    degree_level VARCHAR(255),  
    age INT,  
    years_experience INT,  
    previous_occupation VARCHAR(255),  
    race VARCHAR(255),  
    religious_affiliation VARCHAR(255),  
    phone_number INT,  
    email VARCHAR(255),  
    last_degree_issued VARCHAR(255),  
    ssn INT NOT NULL,  
    gender VARCHAR(255),  
    background_check_data VARCHAR(255)  
);
```

```
Jobs (  
    job_id INT NOT NULL,  
    position_title VARCHAR(255),  
    department VARCHAR(255),  
    tenure_track BOOLEAN,  
    full_time BOOLEAN,
```

```
    part_time BOOLEAN,  
    adjunct BOOLEAN,  
    hourly BOOLEAN,  
    per_hour_wage FLOAT,  
    salary INT,  
    listing_date DATETIME,  
    end_listing_date DATETIME,  
    openings INT  
);
```

```
Applications (  
    application_id INT NOT NULL,  
    job VARCHAR(255),  
    position VARCHAR(255),  
    tenure_track BOOLEAN,  
    date DATETIME,  
    candidate VARCHAR(255)  
);
```

```
Letters (  
    letter_id INT NOT NULL,  
    offer_or_rejection VARCHAR(255),  
    job VARCHAR(255),  
    start_date DATETIME,  
    next_steps VARCHAR(255),  
    rejection_date DATETIME,  
    denial_reason VARCHAR(255)  
);
```

```
Departments (  
    name VARCHAR(255) NOT NULL,  
    abbreviation VARCHAR(255),  
    max_faculty_capacity INT,  
    current_number_of_faculty INT,  
    academic_division VARCHAR(255)  
);
```

Final Analysis: The design of our relations and the E/R diagram was based upon relevant and distinct fields within the hiring process at Westmont for the Provost office. Specifically, we made sure to elicit irrelevant information or redundant fields, in order to achieve a level of normalization and organization within the relations field. We created each field in a way that has

commonalities to make sure that each relation table may be queried and combined to find specific attributes and allow for important fields to be found. Lastly, we made sure to separate distinct fields that would cause buildup or lack of clarity in the tables, in order to abide by more normalization standards and cleanliness.