

Description 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 letters, rejection letters, and departmental information, all while ensuring compliance with institutional standards and constraints.

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

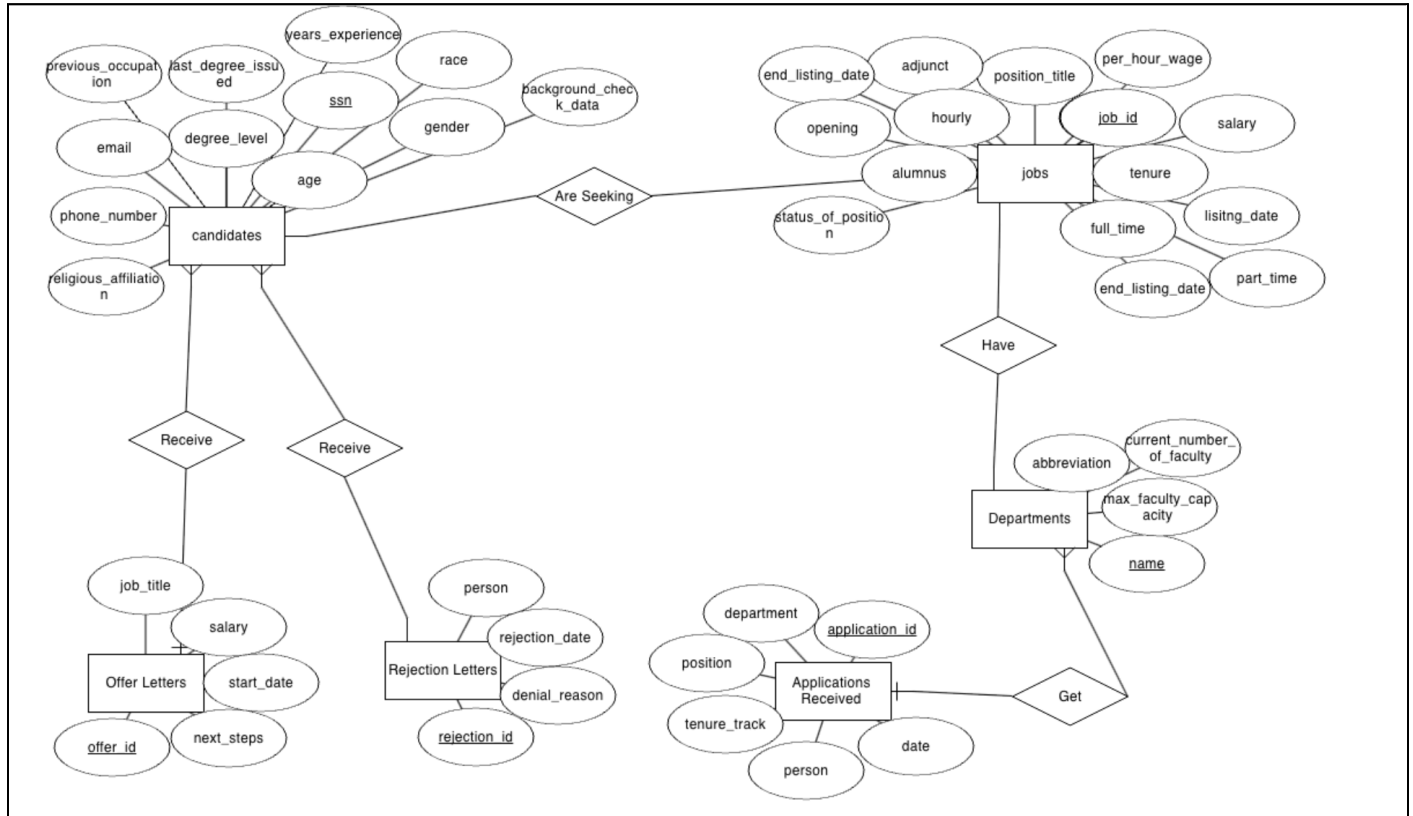
Query for Database Ideas:

- 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/D Diagram:



Conversion to Relations:

- Candidates (Degree Level, Age, Years of Experience, Previous Occupation, Race, Religious Affiliation, Phone Number, Email Address, Last Degree Issued, SSN, Gender, Background Check Data)
- Jobs (Job Id, Position Title, Department, Tenure, Full Time, Part Time, Adjunct, Alumnum, Hourly, Per Hour Wage, Salary, Listing Date, Status of Position, Opening)
- Offer Letters (Offer ID, Salary, Start Date, Next Steps)
- Rejection Letters (Offer Id, Person, Date, Denial Reason)
- Departments (Name, Abbreviation, Current Number of Faculty, Max Faculty Capacity)
- Applications Received (Application id, Department, Position, Tenure Track, Date, Person)

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,  
    age INT,  
    years_experience INT,  
    previous_occupation VARCHAR,  
    race VARCHAR  
    religious_affiliation VARCHAR,  
    phone_number INT,  
    email_address VARCHAR,  
    last_degree_issued VARCHAR,  
    ssn INT,  
    gender VARCHAR,  
    background_check_data VARCHAR,  
    PRIMARY KEY(ssn)  
)
```

```
Jobs (  
    job_id INT,  
    position_title VARCHAR,  
    department VARCHAR,  
    tenure BOOLEAN,  
    full_time BOOLEAN,  
    part_time BOOLEAN,  
    adjunct BOOLEAN,  
    alumnus BOOLEAN,  
    hourly BOOLEAN,  
    per_hour_wage FLOAT,  
    salary INT,  
    lisitng_date VARCHAR,
```

```
end_listing_date VARCHAR,  
status_of_position VARCHAR,  
opening BOOLEAN,  
PRIMARY KEY (job_id)  
)
```

```
Applications_Received(  
    application_id INT,  
    department VARCHAR,  
    position VARCHAR,  
    tenure_track BOOLEAN,  
    date DATE,  
    person PERSON,  
    PRIMARY KEY (application_id)  
)
```

```
Offer_Letters(  
    offer_id INT,  
    job_title VARCHAR,  
    salary INT,  
    start_date DATE,  
    next_steps VARCHAR,  
    PRIMARY KEY (offer_id)  
)
```

```
Rejection_Letters(  
    offer_id INT,  
    person VARCHAR,  
    rejection_date DATE,  
    denial_reason VARCHAR,  
    PRIMARY KEY (offer_id)  
)
```

```
Departments (  
    name VARCHAR,  
    abbreviation VARCHAR,  
    max_faculty_capacity INT,  
    current_number_of_faculty INT,  
    PRIMARY KEY (name)  
)
```

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.