



## Project 2: Recruitment Database

---

### **INTRODUCTION:**

Human Resources (HR) department in any company is involved in developing and administering programs that are designed to increase the effectiveness of the organization. It includes the entire spectrum of creating, managing, and cultivating the employer-employee relationship.

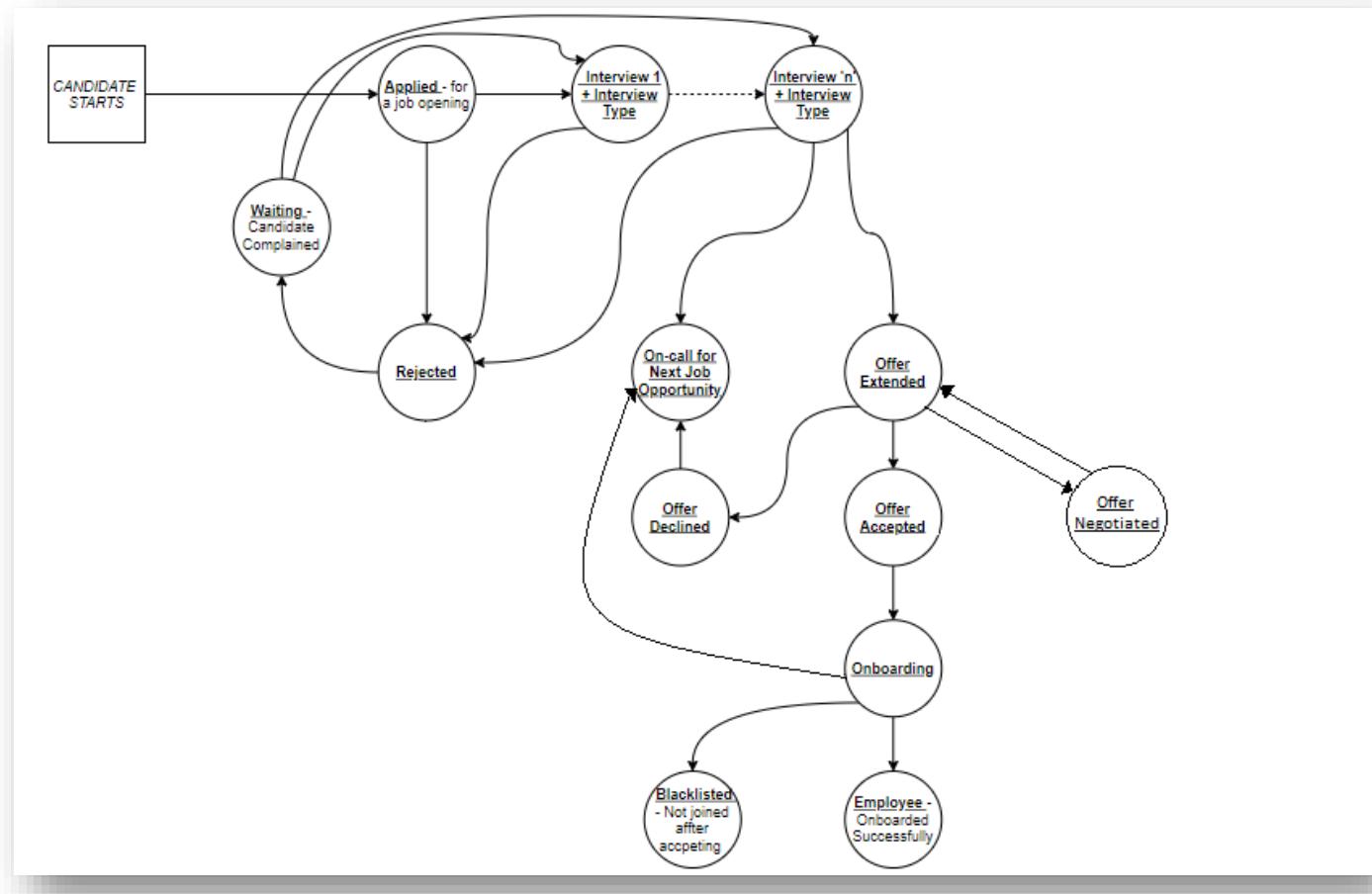
Human resource management is therefore focused on a number of major areas, including:

- *Recruitment*
- Compensation and benefits
- Training and learning
- Labor and employee relations
- Organization development

### **PROBLEM STATEMENT:**

In this project, you will design, implement, and test a database for the *Recruitment* branch of the HR Department in a company called 'XYZ'.

Tracking every candidate's status is the focus of this database. The interview process affects a candidate's status from time-to-time. A candidate's status flow diagram is given below, wherein the underlined words are the **statuses**.



#### Interview Type:

- Online
- Onsite (When a candidate has to physically go to the company for an interview)

#### Job Type:

- Summer Internship
- Full-time Job
- Contract-based

#### Design Flow and Requirements for Tracking 'n' number of Candidates' Recruitment:

- For the purpose of this project, you are the database admin and thus the 'point-of-contact' for all candidates be it for any job-opening for any position.
- Candidate applies for Job-Openings at Company 'XYZ'.
- Company rejects or selects the candidate for 1<sup>st</sup> interview, which can be online or onsite.
- Company rejects or selects the candidate for the following interviews till the final 'n<sup>th</sup>' interview.

- Any of these interviews can again be online or onsite.
- Thus, the candidate can be rejected after applying to a job opening, or after any of the 'n' interviews.
  - If a rejected candidate re-applies for any job-opening, a brand-new status flow starts for that candidate. There is no limit on the number of times a candidate can re-apply or the time period after which a candidate can re-apply.
  - If a rejected candidate complains about the hiring or interviewing process, the company puts the candidate on a 'waiting' status. The 'Complaint Handling' branch of the HR department looks into the complaint (with whose working we are not concerned). If the Complaint department finds that the complaint is correct, a re-interview takes place and status is changed accordingly. However, if the Complaint department finds that the complaint is invalid, status is again changed to 'rejected'.
  - If after the final 'n<sup>th</sup>' interview the company likes the candidate but there are no more job openings available for the position interviewed for, the candidate is put on a "on-call for next job opportunity" status. This means that the company will themselves contact this candidate for interviews during the next round of recruitment.
  - If after the final 'n<sup>th</sup>' interview the company likes the candidate and there is a job opening available, an offer is extended to the candidate.
    - Whether a candidate is selected or rejected after (1) the 'application scanning' phase in the beginning or (2) after any of the 'n' interviews, is decided by the interview team of the company. You as a database admin are not concerned with the internal working of this interview team. When someone from the interview team updates the answer for a particular interview, you as admin will change the candidate's status accordingly.
  - The candidate can either decline or accept this offer, or can negotiate 'n' number of times and then decline or accept the negotiated offer. Status is changed accordingly (Declined, Accepted, Negotiating).
  - If the given or negotiated offer is accepted, the onboarding procedure starts. The onboarding procedure involves candidate's background check, and collection and checking of all documents required for employment.
  - If the candidate declines the given or negotiated offer, s/he is again put on the "on-call for next job opportunity" status.
  - If onboarding is successfully completed, the candidate becomes an employee. For the purpose of this project, this is where our recruitment procedure will end.
  - If the onboarding is unsuccessful (some joining requirements were not met) the candidate is again put on the "on-call for next job opportunity" status.
  - If after accepting the offer and/or after successful completion of onboarding, the candidate does not join the company, s/he is blacklisted. This means that the candidate cannot again apply for any job-opening in the company.

- 
- If any interview is onsite, your database needs to keep track of the following things per candidate:
    - Airline reservation details

- Hotel reservation details
  - Car Rental details
  - Reimbursement details (The candidate will be asked to submit their expense receipts - all expenses that the company says it will cover for food, etc during the period of the onsite recruitment process. The reimbursement team will update whether the reimbursement that the candidate is asking for is valid and how much reimbursement will be given.)
  - If any interview is online or at a location close to the candidate's residence, one or more of the above given details would not be needed.
- 

- The job openings can be for any department's any position.
  - These job openings can be for three types:
    - Summer Internship
    - Full-time Job
    - Contract-based
  - The number of job openings per position are fixed and need to be tracked. Every time a position is filled by a candidate (that is when the status becomes 'offer extended'), the number of job openings for that position should be reduced by 1).
  - And every time an offer is declined or somebody is blacklisted, the corresponding job opening should be increased by 1.
  - Although here we have only one db admin, the database has other roles of interviewers, onboarding team specialist, etc., with the power to change a few things in the database. Thus, when the reducing or increasing of the number of job-openings by 1 is taking place, make sure that nobody else can do that at the same time, else there would be a synchronization issue.
- 

- Per interview, the candidate can give a review for the interviewer, and the interviewer can give a review for the candidate.
  - Make the status changes due to the on-going interview process per candidate as automatic as possible as the design of your database can allow.
- 

Main Tables to start your Database Design off with:

- Candidates
- Interviewer
- Interview
- Job Openings
- AirlineReservation
- CarRental

- HotelReservation
- Reimbursement
- Onboarding

## **Technical Requirements:**

### Design:

- 1) Include **everything** mentioned in the **Project Statement**. Once everything asked is taken care of, you can add your own design constraints if you want.
- 2) Determine what tables to form.
- 3) Determine what information to store per table
- 4) Determine the logic and thus queries you would be needing for testing your design and implementation.
- 5) Address potential integrity and security issues.
- 6) Design a descriptive **E/R diagram** (please use MS Visio, draw.io, or any similar tool)
- 7) Normalize your design into its **3<sup>rd</sup> Normal Form**. Your final database design should have **at least 16 to 20 tables**.
- 8) Please do **NOT** cross lines and please make your design easy to read. A **full screenshot** of your design is required, which should also be readable.
- 9) **CamelCase** naming standard is required (If a name is formed of multiple words that are joined together as a single word, the first letter of each of the multiple words should be capitalized and no underscore is allowed). Incorrect capitalization will lose points.
- 10) Please use **full word** to name your tables/columns. Do **NOT** use shortcuts.

### Implement:

- 1) Determine and implement tables, columns, primary keys, foreign keys, data types, nullabilities and relationships reasonably. These should follow the **real world**. For example, do **NOT** use "Table1" or "Person2" as your database entry content. Every record of data should have real meaning, like "PersonInformation" etc.
- 2) Please **code by yourself** the creation of your:
  1. database,
  2. tables (according to your design),
  3. constraints (at least 1, as needed by your business logic),
  4. triggers (at least 1, as needed by your business logic)

**No** auto-generation of SQL code is allowed.

### Test:

- 1) For testing, populate your database with test data (**3~5** records or rows) with real meaning (same as Requirement 1 in Implementation part). Proper datatypes are required. Use meaningful data to demonstrate your database works properly.
- 2) After data insertions, demonstrate your database's reliability and logic through **several scenarios** (**All requirements mentioned in the problem statement should be proved as working**. After that you can add your own scenarios if needed or wanted) showcased using:
  - 4 views

- 2 stored procedures
  - 2 functions
  - 2 scripts (at least 1 of them should create users with various security levels, password assignments, roles, encryptions).
  - 1 transaction (to avoid synchronization issues). If you are incorporating your transaction as a 'trigger' during the 'Implementation' phase, please mention so.
- 3) These scenarios should be meaningful. Any scenario like "Create a view to show all candidates' information" will **NOT** be accepted.
  - 4) For every scenario, please add a sentence before to demonstrate the **purpose** of that scenario.
  - 5) To test your database, the design and data everyone loaded must be **different**. Teamwork is **NOT** allowed.

Please submit your typed report as a single file (.doc or .pdf) on Blackboard.

Your report should include following sections:

1. [5] Cover page: a descriptive title, a short abstract, and your information
2. [50] Design: your introduction, design considerations and choices, and E/R diagram's screenshot
3. [10] Implementation: source codes with your comments, and screenshots.
4. [10] Testing: testcases and screenshots with your comments
5. [5] Conclusions: your project analysis, and your remarks on this project

All reports that satisfy all the following basic requirements will have a base score of 20 points. The maximum points deducted for noncompliance is indicated in parentheses.

1. General report guidelines followed, i.e. sections labeled, pages numbered [10]
2. Report adheres to basic standards of grammar and spelling [10]