Career Management System Database Design Document

Project Team 07

Database Specification: Purpose, Business Problems Addressed, and Business Rules

Database Purpose: With the recent upsurge in layoffs and the struggle for newer opportunities, the purpose of this database is to maintain the data of candidates, recruiters, as well as jobs across various domains based on skills, requirements, and availability, to provide a platform for interactions and connections to land their dream job

Business Problems Addressed:

- Recruiters spend hours reviewing resumes and scheduling interviews as part of the traditional recruiting process. Several of these processes can be automated by our career management system, expediting the procedure and saving time.
- Keeping track of several applicants throughout the employment process can be challenging. A unified database of candidate data can be provided through the career management system, making it simple to follow each applicant's progress through the hiring process.
- The correct candidates can be attracted via clear and precise job advertisements, which can be achieved with the use of career management tools. Poorly written job postings or ones that do not adequately reflect the job's criteria may attract a large number of unqualified candidates.
- The collaboration and communication between recruiting team members can be facilitated through the career management system. The hiring team may not be on the same page if there is no single mechanism in place for managing the hiring process.

Business Rules:

1. Each Job will belong to one and only one job_Category.
2. Each Organization may have zero or many Jobs.
3. Each Job Position may have zero or many Jobs.
4. Each Job Platform will have one or many Jobs.
5. Each Job may have zero or many Candidate Job Applications associated with it.
6. Each Candidate_Job_Application will have one or many associated Document_Applications.
7. Each Candidate will be allowed to apply to zero or many Candidate_Job_Application.
8. A Candidate Job Application will have one or many Application Status Changes happen.
9. Recruiter and Candidate_Job_Application will have a one to many relationship with Candidate_Evaluation.
10. A Candidate_Job_Application can have one or many Application_Test associated with it.

Design Requirements:

- Use Crow's Foot notation.
- Specify the Primary Keys (PK) for each table.
- Draw lines between entities to show relationships between two entities. The line points to fields in each table that are used to form the relationship.
- In a one-to-many relationship, we need to place a ‘1’ next to the field on the one side of the relationship and a crow’s feet symbol next to the field on the many sides of the relationship.

Design Decisions:

Entity Name	Why Entity Used	How Entity is related to other Entities
Job	The job entity stores all the information related to jobs and provides the database with Job Name,description.	As one of the core entities, it has many foreign keys and can be used to look up the category, position, platform and organization. The primary key is the Job_ID and this entity also contains the actual job description.
Job_Category	Table gives a description of the job category such as technology, management, health, financial, and education.	The job is categorized into different categories, it is connected to the Job entity with a zero-to-many relationship through the Job_Category_ID.
Job_Position	The job position entity will contain the actual job title where we will make a separate database for employment positions since one title can be posted for several positions.	The Job category is further categorized into multiple positions and levels. This entity is also related to a Job entity with a zero-to-many relationship through the Job_position_ID field.
Job_Platform	This entity refers to the channel through which the job opening will be advertised. For instance, a position could be advertised in a local newspaper, online job board, or on social media websites. A link to that job posting can be added in the description field.	The Job_platform is connected to the Job entity with a zero-to-many relationship through the job_platform_id.
Organization	The organization table will contain data on every company that has ever utilized this database for hiring purposes.	Every Job is associated with an organization. An organization has a zero-to-many relationship with a Job entity.
Document_Application	All applications will include an attachment. One document may be added to numerous applications, and numerous supporting documents may be attached to one application. This indicates that the candidate job application and document tables have a many-to-many relationship. To manage this relationship, the lookup table Document_Application has been created.	This entity is an associative entity between Candidate_Job_Application and document, it breaks the many-to-many relationship between Candidate_Job_Application and makes it a zero-to-many relationship.

Entity Name	Why Entity Used	How Entity is related to other Entities
Document	The document table keeps track of any supplementary files the candidate wants to submit. They could include cover letters, resumes, letters of recommendation, and so on. The file will be stored in binary format in this table's document binary column. The name column stores the document's name, last update designates the most recent version uploaded by the applicant, and the url field may be used to store a link to the document. Both the methods of document and url are both nullable; candidates are free to use one or both of them to add information to their applications.	Every Document has a zero-to-many relationship with the Document_Application entity.
Candidate_Job_Application	The information about each candidate's job application, including its date, is listed in the following table. The columns for education and experience are also included in the table. These fields might be present in the candidate table, however a candidate might or might not choose to list a certain educational background or work history on each application they submit. These columns are therefore a part of the table for candidate job applications. Any additional application-related information is kept in the other_info column. The job_id and candidate_id columns, in the candidate job application database, respectively, are foreign keys from the job and candidate tables.	The Candidate_Job_Application is at the heart of this system connecting the Candidate to the Job as well as the Recruiter(through Candidate_Evaluation). It contains multiple Foreign keys and its Primary Key is Candidate_Document_ID. It provides a complete view of the information being provided by the Candidate to be considered for a Job.
Candidate	This entity will store the personal data of candidate's, including their first and last names, email addresses, phone numbers, and summary. The candidate's brief profile will be stored in the summary section.	It is connected to Candidate_Job_Application by its Primary Key id and Foreign Key Candidate_id. It contains all the basic details of the candidate which would be necessary to fill across all job postings.

Entity Name	Why Entity Used	How Entity is related to other Entities
Candidate_Evaluation	Information concerning application evaluations is kept in the Candidate_Evaluation table. It includes recruiter feedback in addition to the Candidate_Document_ID and Recruiter_ID (in notes).	Candidate_Evaluation has one to many relations with Candidate_Job_Application as well as Recruiter via its Primary Key id and Foreign Keys for Recruiter_ID and Candidate_Document_ID contained in it.
Application_Test	A test can be linked to many applications, and a test can be linked to numerous applications. Application_Test is the lookup table we will use to implement this relationship. Since a test may not have a set duration, start time, or end time, the start time and end time columns are nullable.	Application_Test has a one-to-many relationship with Candidate_Job_Application via Primary Key and Foreign Key Candidate_Document_ID. Application_Test has a one-to-many relationship with Online_Assessments via Primary_Key and Foreign Key OA_ID.
Application_Status_Change	All submitted applications' status modifications are kept track of, in the application status change table. The date of the status change is kept in the date_changed column. To evaluate the processing times for each stage of several applications, this table can be useful.	Application_Status_Change has a one-to-many relationship with Candidate_Job_Application via its Primary Key and Foreign Key Candidate_Document_ID. Application_Status_Change has a one-to-many relationship with Application_Status via Primary Key and Foreign Key Application_Status_ID.
Recruiter	Recruiter is used to perform candidate evaluation, storing each recruiter's first_name, last_name, and unique id number.	Recruiter entity is related to one application that can be evaluated by multiple recruiters and one recruiter can evaluate multiple applications, so the recruiter table has a one-to-many relationship with the Candidate_Evaluation table.
Application_Status	An application can have several stages,like "under process", "waiting for decision", "selected", etc. An application will have status of "not_submitted" when a user has started an application but has not submitted it for the recruiters to review.The application_status is used to track the status of the candidate applications.	The final status of the whole process is stored in the entity and it is linked to Aplication_Status_Change through the primary key Application_Status_ID

Entity Name	Why Entity Used	How Entity is related to other Entities
Online_Assesments	The Online_Assesments table stores test details including its unique id, code name, its duration in minutes, and the maximum score possible for that test.	Online_Assessments has a one to many relation with Application_Test by Primary Key id in order to maintain all the information about any assessments the candidate may have to undergo as part of the hiring process.