# Recruitment Agency Database Design

Author: Niko Tchitchinadze

File Name: DB\_Niko\_Tchitchinadze\_HW\_RecruitmentAgency\_descriptions.docx

## 1. Business Description

### 1.1 Business Background

The Recruitment Agency database is designed to support and simplify the operations of an agency that connects job-seeking individuals (candidates) with their potential employers (client companies). The agency manages job listings, candidate registrations, application tracking, interviews and placements. On top of this, it provides additional services such as CV tailoring, interview coaching, and professional skill development to make the candidate more competitive. This system captures every stage of the recruitment process, ensuring data consistency, traceability, and efficiency in matching the right candidates to the right positions.

### 1.2 Problems

Before implementing a structured database, the agency relied on spreadsheets and manual tracking systems, leading to frequent data duplication, incomplete candidate information, and difficulties managing job postings and interview schedules. Recruiters faced challenges when it came to locating relevant candidates quickly, while clients lacked visibility into application progress, and reporting required extensive manual work. These inefficiencies caused delays in placements and reduced overall productivity.

### 1.3 The Benefits of Implementing a Database

The new database centralizes all recruitment data into a single, reliable system. It allows recruiters to manage clients, candidates, and job applications efficiently, enabling automated matching based on skills, experience, and preferences. The system ensures data integrity, supports detailed reporting, and enhances collaboration across branches. In the long term, the database will serve as the foundation for analytical tools to improve business performance and optimize placement success rates.

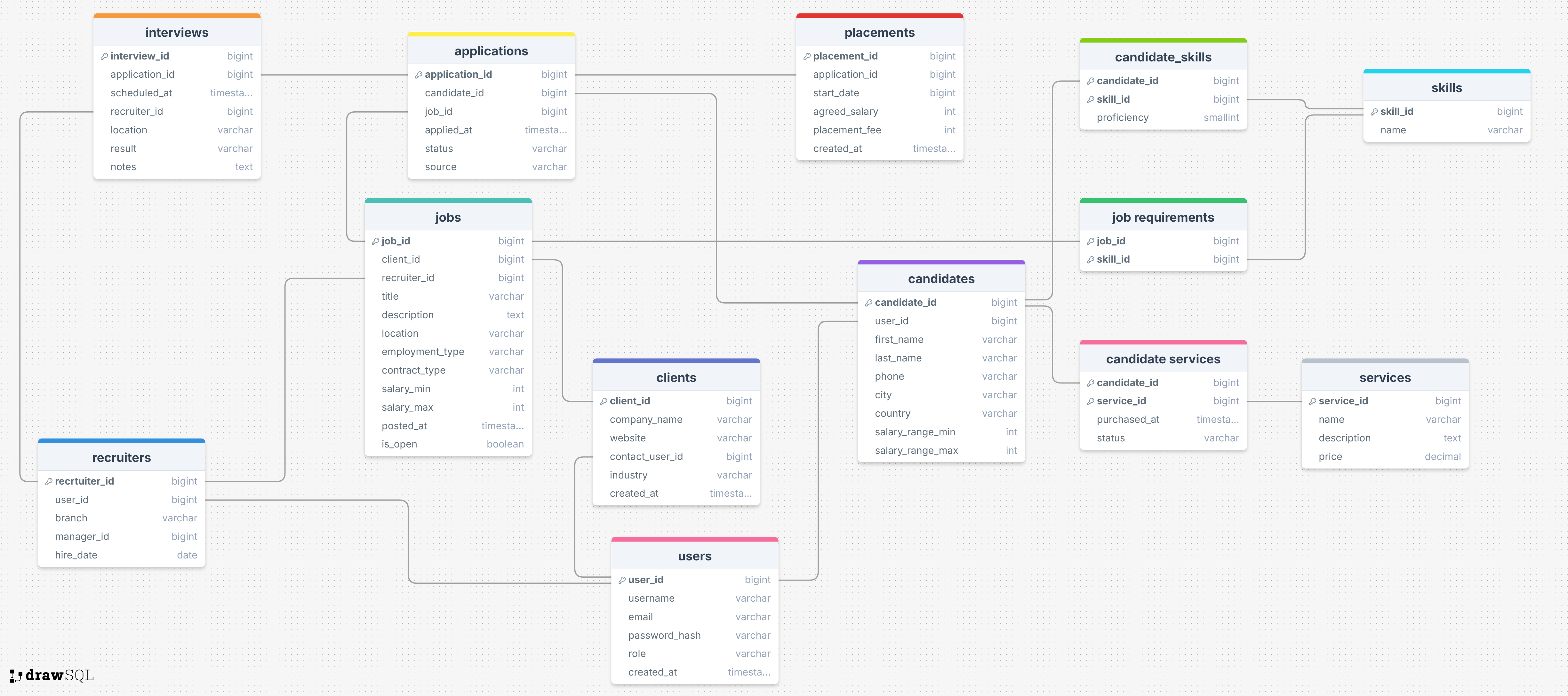
## 2. Model Description

### 2.1 Definitions

PK – Primary Key  
FK – Foreign Key  
M:N – Many-to-Many Relationship  
3NF – Third Normal Form  
DDL – Data Definition Language  
ERD – Entity Relationship Diagram

### 2.2 Logical Scheme

The following diagram represents the logical structure of the Recruitment Agency database, showing tables, relationships, and constraints:



### 2.3 Objects

The model consists of thirteen tables that represent the main entities and relationships within the recruitment agency. Each table stores data relevant to one specific entity or process, maintaining a clear separation of concerns and eliminating redundancy (3NF compliance).

#### Table Name: Candidates

|  |  |  |
| --- | --- | --- |
| Field Name | Description | Data Type |
| candidate\_id | Unique candidate identifier (PK) | bigint |
| user\_id | References users table (FK) | bigint |
| first\_name | Candidate’s first name | varchar(100) |
| last\_name | Candidate’s last name | varchar(100) |
| phone | Contact phone number | varchar(30) |
| city | Candidate’s city of residence | varchar(100) |
| country | Candidate’s country of residence | varchar(100) |
| desired\_salary | Preferred salary amount | int |
| resume\_url | Link to uploaded resume file | varchar(2048) |

#### Table Name: Jobs

|  |  |  |
| --- | --- | --- |
| Field Name | Description | Data Type |
| job\_id | Unique job identifier (PK) | bigint |
| client\_id | References clients table (FK) | bigint |
| recruiter\_id | References recruiters table (FK) | bigint |
| title | Job title | varchar(200) |
| description | Detailed job description | text |
| location | Job location | varchar(200) |
| employment\_type | Type of employment (full-time, part-time, etc.) | varchar(50) |
| salary\_min | Minimum salary offered | int |
| salary\_max | Maximum salary offered | int |

#### Table Name: Applications

|  |  |  |
| --- | --- | --- |
| Field Name | Description | Data Type |
| application\_id | Unique application identifier (PK) | bigint |
| candidate\_id | References candidates table (FK) | bigint |
| job\_id | References jobs table (FK) | bigint |
| applied\_at | Date and time of application | timestamp |
| status | Current status of the application | varchar(50) |
| source | Origin of the application (portal, referral, etc.) | varchar(100) |

#### Table Name: Recruiters

|  |  |  |
| --- | --- | --- |
| Field Name | Description | Data Type |
| recruiter\_id | Unique recruiter identifier (PK) | bigint |
| user\_id | References users table (FK) | bigint |
| manager\_id | Self-reference to recruiters table (FK) | bigint |
| branch | Recruitment branch or office | varchar(100) |
| hire\_date | Recruiter’s hire date | date |

#### Table Name: Skills

|  |  |  |
| --- | --- | --- |
| Field Name | Description | Data Type |
| skill\_id | Unique skill identifier (PK) | bigint |
| name | Skill name | varchar(150) |

#### Comments on Table Relationships

The Recruitment Agency database contains multiple one-to-many and many-to-many relationships.   
Applications represent the link between Candidates and Jobs, forming the core of the recruitment process.   
Candidate\_Skills and Job\_Requirements bridge Candidates ↔ Skills and Jobs ↔ Skills relationships respectively, supporting skill-based matching.   
The Recruiters table contains a self-referential foreign key (manager\_id) to represent hierarchy within the agency.   
Placements link to successful Applications, while Candidate\_Services link Candidates to Services purchased.   
These relationships ensure referential integrity and logical representation of real-world recruitment interactions.

#### Example with Data

|  |  |  |  |
| --- | --- | --- | --- |
| candidate\_id | job\_id | status | applied\_at |
| 101 | 5001 | interview | 2025-10-10 09:00 |
| 102 | 5002 | rejected | 2025-10-11 14:30 |