

|  |
| --- |
| EliteHire  Elite Talent. Elit Results.  Recruitment Agency |
|  |

Contents

[1 Business Description 3](#_Toc62212630)

[1.1 About EliteHire 3](#_Toc62212631)

[1.2 Challenges in the hiring process 3](#_Toc62212632)

[1.3 The benefits of implementing a database. 3](#_Toc62212633)

[2 Model description 3](#_Toc62212634)

[2.1 Definitions & Acronyms 3](#_Toc62212635)

[2.2 Logical Scheme 3](#_Toc62212636)

[2.3 Objects 3](#_Toc62212637)

# 

# Business Description

## About elitehire

## EliteHire is a recruitment agency that acts as a bridge between job seekers and employers. It streamlines the hiring process by managing job listings, tracking applications, screening candidates, scheduling interviews, and handling final placements. Additionally, EliteHire offers career-enhancing services such as resume writing, interview coaching, and skills development to improve job seekers' chances of securing employment.

## Challenges in the hiring process

* Difficulty in tracking applications and matching candidates to job openings based on skills, experience, and preferences.
* Manual handling of interview scheduling and placement decisions.
* Limited access to structured applicant profiles and job requirements.
* Challenges in tracking employer and recruiter interactions with candidates.
* No structured way to store applicant preferences or job-related services.

## the Benefits of implementing a database

* Improve job matching based on skills, experience, and location.
* Automation of application tracking and interview scheduling, reducing manual work..
* Structured record-keeping for candidates, recruiters, and employers, improving data organization
* Track additional services like resume writing and skills training.
* Improve decision-making by analyzing placements and candidate success rates.

# Model description

## Definitions & Acronyms

* **PK (Primary Key):** A unique identifier for each record in a table.
* **FK (Foreign Key):** A reference to a primary key in another table.
* **M:M (Many-to-Many):** A relationship requiring a bridge table to connect entities.
* **1:M (One-to-Many):** A relationship where one record is associated with multiple records in another table.
* **0..M (Zero to Many):** Indicates that one record in a table can be related to zero or many records in another table.

## Logical Scheme

SORRY FOR INCONVINIENCE THE LOGICAL SCHEME IS ON THE LAST PAGE (16th PAGE)

## Objects

**Table Description**

Stores details of job positions posted by employers, including required experience, status, and application tracking.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| JobOpenings | Job\_ID | Unique identifier for job opening (PK, UNIQUE) | Int (AUTO INCREMENT) |
| Employer\_ID | ID of the employer who posted the job (FK) | Int |
| Job\_name | Title of the job (NOT NULL) | Varchar(255) |
| Num\_of\_applicants | Number of applicants for the job(DEFAULT 0) | Int |
| Opening\_date | Date when job listing was posted (NOT NULL) | Date |
| Closing\_date | Date when the job listing closes (NULLABLE) | Date |
| Status | Status whether the job opening is open/closed | ENUM(Open,Closed) |
| Expected\_experience | Minimum experience required | Float |

**Comments on table relationships**

M:1 with Employees (Each employer posts multiple job openings).

M:M with JobSkills (Each job may require multiple skills). Many-to-Many (M:M) relationships often cause data redundancy, inconsistency, and difficulty in enforcing constraints. Instead of directly linking JobOpenings and JobSkills, we created the JobOpeningSkills table to manage the relationship efficiently.

0:M with JobApplicants (a job can have 0 or many applications from applicants ).

1:1 with JobPreferences ( Each job has unique and specific JobPreference)

0:M with Services (As there can be a JobOpenings with multiple services and there can be a service that is not linked with any job, like resume writing)

1:M with Placements (One job can have multiple placements, but each placement is linked to only one job)

**Example with data**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Job\_ID | Employee\_ID | Job\_name | Num\_of\_applicants | Opening\_date | Closing\_date | Status | Expected\_experience |
| 100 | 1 | Data Analyst | 10 | 2025-03-15 | NULL | Opened | 3.5 |

**Table Description**

The Applicants table stores information about job seekers, including their personal details, contact information, and work experience. This table is essential for tracking who is applying for jobs and matching them to suitable positions.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Applicants | Applicant\_ID | Unique identifier for the applicant (PK,UNIQUE) | Int (AUTO INCREMENT) |
| Name | First name of the applicant (NOT NULL) | Varchar (100) |
| Surname | Last name of the applicant (NOT NULL) | Varchar(100) |
| Gender | Gender of the applicant (NOT NULL) | ENUM (‘M’,’F’)  ‘M’ for Male  ‘F’ for Female |
| Email | Email of the applicant | Varchar(255) |
| Opening\_date | Date when job listing was posted (NOT NULL) | Date |
| Experience | Years of experience (NOT NULL) | Float |

**Comments on table relationships**

1:M with ApplicantsPreference each applicant can have different preferences for different jobs, thus 1 row of applicant table can correspond to multiple ones in ApplicantPreference.

0:M with JobApplicants this relation is because one applicant or person can apply for 0 or multiple of jobs

M:M with JobSkills this is for matching applicant skills with skills that jobs require and for handling we created ApplicantSkills table with which it has 1:M relationship as one applicant can have multiple of skills

1:M with Interviews table as one applicant can have multiple of interviews

0:M with Placements table as an applicant can be hired for specific job then if unfortunately he or she gets sacked and we hire him or her for the second time we will have multiple of records in placements table with the same applicant.Also the second scenario is that applicant has never been hired.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Applicant\_ID | Name | Surname | Gender | Email | Opening\_date | Experience |
| 200 | Karo | Tashchyan | Female | tashchyankar@gmail.com | 2025-03-15 | 2.5 |

**Example with data**

|  |  |  |
| --- | --- | --- |
| Field name | Field Description | Data Type |
| Applicant\_ID | ID of the applying candidate (FK) | Int |
| Job\_ID | ID of the applied job (FK) | Int |
| Application\_date | Date of job application (NOT NULL) | Date |
| Status | Status of application (NOT NULL) | ENUM(‘Pending’,’Accepted’,’Rejected’,’Hired’) |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Field name | Field Description | Data Type |
| JobApplicants | Applicant\_ID | ID of the applying candidate (FK)(PK) | Int |
| Job\_ID | ID of the applied job (FK)(PK) | Int |
| Application\_date | Date of job application (NOT NULL) | Date |
| Status | Status of application (NOT NULL) | ENUM(‘Pending’,’Accepted’,’Rejected’,’Hired’) |

The JobApplicants table tracks which applicants have applied for which jobs. It records application details, including the date of application and its current status.

Comments on table relationships

**Comments on table relationships**

JobApplicants has M:0 relationships with Applicants and JobOpenings tables

One JobApplicants row can correspond to only one JobOpenings but one JobOpenings can have multiple JobApplications. Also Job can be left without applicants.

M:0 with Applicants one applicant can apply 0 or many times to different JobOpenings but one JobApplicants record corresponds to only one row or record of JobApplicants table.

**Example with data**

|  |  |  |  |
| --- | --- | --- | --- |
| Applicant\_ID | Job\_ID | Application\_date | Status |
| 200 | 100 | 2025-03-15 | Pending |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| JobSkills | Job\_Skills\_ID | Unique identifier for skill (PK, UNIQUE) | Int |
| Skill\_name | Name of the skill (NOT NULL) | Varchar(255) |

**Table Description**

The JobSkills table stores a list of all skills that can be associated with job openings and applicants. This table ensures that skills are standardized and reused across different job postings and applicants.

**Comments on table relationships**

JobSkills table has **1:M** relationships with bridge tables **JobOpeningSkills** and **ApplicantSkills** tables and if we remove bridge tables we will have M:M relationships with JobOpenings and Applicants tables.

**Example with data**

|  |  |
| --- | --- |
| Job\_Skills\_ID | Skill\_name |
| 1 | SQL |
| 2 | Python |
| 3 | Java |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Interviews | Interview\_ID | Unique identifier for interview (PK, UNIQUE) | Int |
| Job\_ID | Related job opening (FK) | Int |
| Applicant\_ID | Related applicant (FK) | Int |
| Recruiter\_ID | Recruiter conducting the interview (FK) | Int |
| Stage | Interview stage | ENUM (‘HR’,’Technical’,’Final’) |
| Feedback | Feedback after the interview (NULLABLE) | Text |
| Scheduled\_Date | Date of the interview (NOT NULL) | Date |
| Time | Interview time (NOT NULL) | Time |

**Table Description**

The Interviews table tracks all interview rounds conducted between applicants and recruiters for specific job openings. It records the stage of the interview, feedback, and scheduling details.

**Comments on table relationships**

M:1 Relationship with JobOpenings. Each interview must belong to a single job opening but one job opening can have multiple interviews.

M:1 Relationship with Applicants Each interview must belong to one applicant, but applicants can have multiple interviews for different jobs.

M:1 Relationship with Employees Each interview must belong to a single recruiter, but one recruiter can conduct multiple interviews.

**Example with data**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Interview\_ID | Job\_ID | Applicant\_ID | Recruiter\_ID | Stage | Feedback | Scheduled\_date | Time |
| 1 | 100 | 200 | 10 | HR | Great | 2025-03-15 | 16:00 |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Placements | Placement\_ID | Unique Identifier for placement (PK, UNIQUE) | Int |
| Applicant\_ID | Placed applicant ID (FK) | Int |
| Job\_ID | Job the applicant was hired for (FK) | Int |
| Offer\_status | Status of the job offer | ENUM(‘Accepted’,’Rejected’) |
| Start\_date | Start of the job(when the employee will start) | Date |
| Salary | Salary employee will get (NOT NULL) | Int |

Comments on table relationships

**Table Description**

The Placements table stores records of successful job hires, meaning an applicant has been offered and accepted a job. It tracks which applicant was hired, for which job, the start date, salary, and whether the offer was accepted or rejected.

**Comments on table relationships**

M:1 Relationship with JobOpenings Each placement must belong to a single job opening but one job opening can have multiple placements.

M:1 Relationship with Applicants Each placement must belong to a single applicant, but applicants can have multiple placements if they get hired for different jobs over time.

**Example with data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Placement\_ID | Applicant\_ID | Job\_ID | Offer\_status | Start\_date | Salary |
| 2 | 200 | 100 | Accepted | 2025-03-15 | 150000 (USD?) |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Employees | Employee\_ID | Unique Identifier for Employee (PK UNIQUE) | Int |
| Name | Employee name (NOT NULL) | Varchar(255) |
| Surname | Employee surname (NOT NULL) | Varchar(255) |
| Type | Employee role (NOT NULL) | ENUM (‘Employer’,’Recruiter’) |
| Email | Employer email (UNIQUE, NOT NULL) | Varchar (255) |

Comments on table relationships

**Table Description**

The Employees table stores information about both recruiters and employers who are involved in the recruitment process. It keeps track of their roles, contact details, and unique identifiers.

**Comments on table relationships**

1:M Relationship with JobOpenings. Each job posting belongs to a single employer.

1:M Relationship with Interviews. Each interview is linked to one recruiter, but recruiters can conduct multiple interviews.

**Example with data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Employee\_ID | Name | Surname | | Type | Email |
| 1 | Henrik | Mkhitaryan | Recruiter | | HenMkhitaryan@gmail.com |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| JobOpeningSkills | Job\_ID | (FK),(PK)Unique Identifier of the job opening that requires a specific skill | Int |
| Job\_Skills\_ID | (FK),(PK)Unique Identifier of the skill that is required for the specific job. | Int |

Comments on table relationships

**Table Description**

The JobOpeningSkills table is a bridge table used to manage the many-to-many (M:M) relationship between JobOpenings and JobSkills. Since each job can require multiple skills and each skill can be required for multiple jobs, this table helps establish the connection without redundancy.

**Comments on table relationships**

M:1 Relationship with JobOpenings Each skill listed for a job is part of the job's requirements

M:1 Relationship with JobSkills Each record connects one skill to one job opening.

**Example with data**

|  |  |
| --- | --- |
| Job\_ID | Job\_Skills\_ID |
| 100 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| ApplicantSkills | Applicant\_ID | Unique identifier of the applicant that has specific skill (NOT NULL)(FK,PK) | Int |
| Job\_Skills\_ID | Unique identifier of Skill that specific applicant has.(NOT NULL)(FK,PK) | Int |

Comments on table relationships

**Table Description**

The ApplicantSkills table is a bridge table that manages the many-to-many (M:M) relationship between Applicants and JobSkills. Since each applicant can have multiple skills and each skill can be possessed by multiple applicants, this table helps establish the connection in a normalized way.

**Comments on table relationships**

**M:1 Relationship with Applicants** Each record in this table links a skill to a specific applicant.

**M:1 Relationship with JobSkills** Each record connects one skill to one applicant.

**Example with data**

|  |  |
| --- | --- |
| Applicant\_ID | Job\_Skills\_ID |
| 200 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Services | Service\_ID | Unique Identifier for the service provided to the applicants. (PK,UNIQUE) | Int |
| Job\_opening\_ID | Unique identifier of the job for what the service is provided can be null as service can be provided before applying for specific job (FK, NULLABLE) | Int |
| Person\_email | Email address who will get the service (UNIQUE, NOT NULL) | Varchar(255) |
| Service\_type | Type of service that should be provided | ENUM('Resume Writing', 'Interview Coaching', 'Skills Development') |
| Start\_date | Date when particular service begins (NOT NULL) | Date |
| End\_date | Date when particular service ends (NOT NULL) | Date |
| Status | Status whether the service begined, canceled or completed | ENUM(‘Begined’,’Canceled’,’Completed’) |

Comments on table relationships

**Table Description**

The Services table stores records of career-related services provided to job seekers. These services can be offered before or after an applicant applies for a job, which is why Job\_opening\_ID is nullable.

**Comments on table relationships**

**M:0 Relationship with JobOpenings** A service can belong to a single job opening and can not as there can be services that are not linked with any job, and one job opening can have multiple services provided to different applicants.

**Example with data**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Service\_ID | Job\_opening\_ID | Person\_email | Service\_type | Start\_date | End\_date | Status |
| 23 | NULL | guest@gmail.com | Resume Writing | 2025-02-04 | 2025-02-04 | Completed |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| JobPreferences | Job\_ID | Unique identifier for job opening (PK, UNIQUE) (FK) | Int |
| Expected\_sallary | Salary that is expected to give for specific job | Int |
| Location | Location of the office from where the job is posted | Varchar(255) |
| Remote | Can job be remote or not | Boolean |

Comments on table relationships

**Table Description**

The JobPreferences table stores predefined job-related preferences, such as expected salary, office location, and whether the job is remote or not. This information helps recruiters and applicants understand the specific job conditions before applying

**Comments on table relationships**

**1:1 Relationship with JobOpenings** Each row in JobPreferences corresponds to a specific Job\_ID

**Example with data**

|  |  |  |  |
| --- | --- | --- | --- |
| Job\_ID | Expected\_sallary | Location | Remote |
| 100 | 120000 | Yerevan | False |

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| ApplicantPreferences | Preference\_ID (PK) | Unique identifier for Preference (PK, UNIQUE) | Int |
| Applicant\_ID (FK) | The applicant who has the preference | Int |
| Job\_ID (FK) | The specific job that the preference refers to | Int |
| Expected\_salary | Salary expectation of applicant for specific job | int |
| Location | Prefered location of the applicant | Varchar(255) |
| Remote | Preference to work remote or not | Boolean |

Comments on table relationships

**Table Description**

The ApplicantPreferences table stores job preference details for each applicant, such as expected salary, preferred job location, and remote work preference. Since an applicant may have different preferences for different jobs, this table helps in matching applicants with suitable job openings.

**Comments on table relationships**

**M:1 Relationship with Applicants** Each preference is specific to one applicant, but an applicant can have multiple preferences for different jobs.

**M:1 Relationship with JobOpenings** Each job preference is linked to one job, but jobs can have multiple applicants with different expectations.

**Example with data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Preference\_ID | Applicant \_ID | Job\_ID | Expected\_salary | Location | Remote |
| 12 | 200 | 100 | 150000 | Yerevan | TRUE |

## Logical Scheme

