

|  |
| --- |
| **Recruitment agency** |
| **Logo / Image** |

Contents

[1 Business Description 3](#_Toc62212630)

[1.1 Business background 3](#_Toc62212631)

[1.2 Problems. Current Situation 3](#_Toc62212632)

[1.3 The benefits of implementing a database. Project Vision 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

## Business background

Recruitment agency manages job listing, candidate registration, application tracking, matches between candidates and job listening based on criteria like skills, experience, location and preferences, interviews, and placements. Moreover, agency offers additional services like resume writing, interview coaching, and skills development to help candidates improve their job prospects

## Problems. Current Situation

An organized database system can answer some crucial business questions, including:

1. Cities or locations with the most job listings (and thus job opportunities)
2. Average years of experience required in a certain job title
3. Job titles that attract the most applications
4. Average years of experience of candidates with a certain job title
5. Most demanded services among the candidates

In the presented model, the above questions could be answered with the following SQL queries:

**1) Cities with the most job listings**

SELECT l.city, COUNT(j.id) AS job\_count

FROM job j

JOIN location l ON j.location = l.id

GROUP BY l.city

ORDER BY job\_count DESC;

**2) Average years of experience required**

SELECT j.title, AVG(j.experience\_in\_numbers) AS avg\_experience\_years

FROM job j GROUP BY j.title;

**3) Job titles that attract the most applications**

SELECT

j.title,

COUNT(a.id) AS application\_count

FROM

application a

JOIN

job j ON a.jobId = j.id

GROUP BY

j.title

ORDER BY

application\_count DESC;

SELECT

j.title,

COUNT(a.id) AS application\_count

FROM

application a

JOIN

job j ON a.jobId = j.id

GROUP BY

j.title

ORDER BY

application\_count DESC;

**4) Average years of experience of candidates with a certain job title**

SELECT

c.title,

AVG(c.experience\_in\_number) AS avg\_experience\_years

FROM

candidate c

GROUP BY

c.title;

**5) Most demanded services among the candidates**

SELECT

s.name AS service\_name,

COUNT(rs.order\_id) AS order\_count

FROM

resume\_service rs

JOIN

services s ON rs.service\_id = s.id

GROUP BY

s.name

ORDER BY

order\_count DESC;

## the Benefits of implementing a database. Project Vision

The expected **benefits** include:

1. Relationship between the tables enables querying and finding answers for key business questions
2. Easy CRUD operations for database tables unlike in the traditional databases like spreadsheets
3. Provides more safety to keep the data in an online database management system
4. Accessible to all employees of the company and seamlessly integrates the data collected by all departments

**Vision**

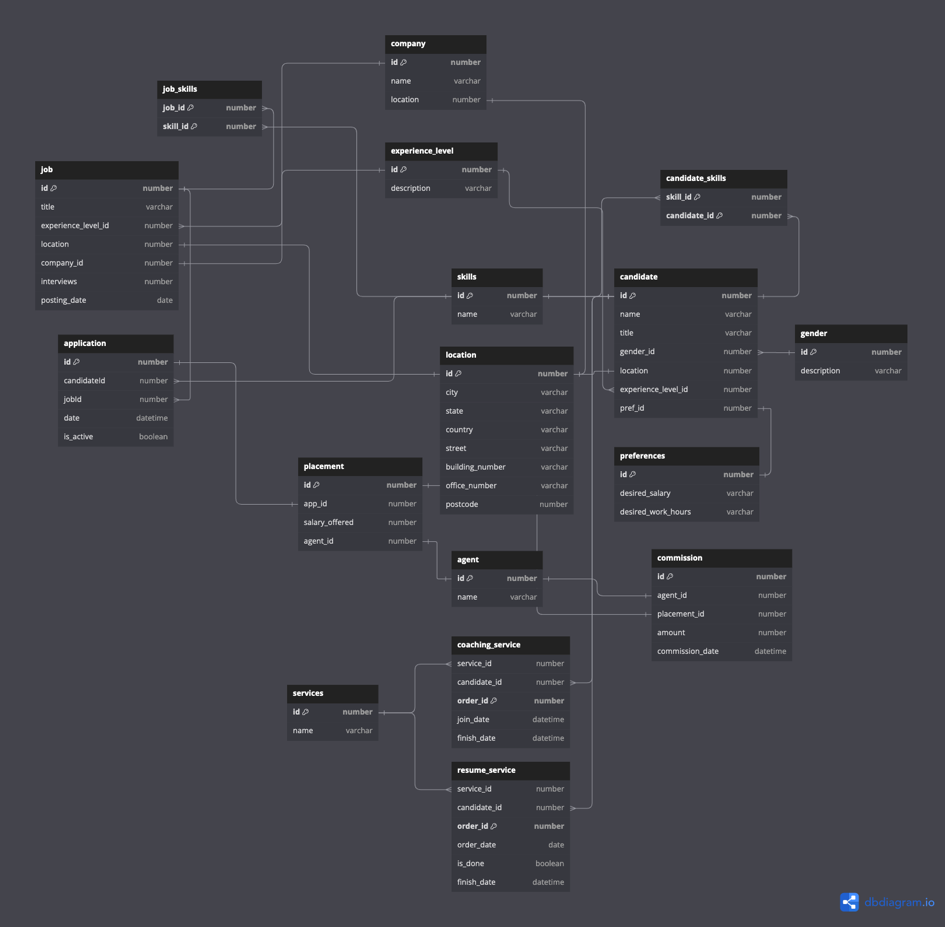
Moving from the traditional clunky spreadsheet system to an online DBMS system for ease of use and better control over the company data

# Model description

## Definitions & Acronyms

* FK – Foreign Key
* PK – Primary Key
* CFK – Compound Foreign Key

## Logical Scheme



## Objects

**Skills table**

Table that displays ID’s and names for the existing skills. E.g. Python coding, SQL database management, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| skills | id | Id of the skill, PK/FK | Int |
| name | Name of the skill, e.g. python coding, SQL database management | Text |

It has one to many relationship with **job\_skills** and **candidate\_skills** tables

Example with data

|  |  |  |
| --- | --- | --- |
| skills.id | job\_skills.id | candidate\_skill.id |
| 1 | 1 | 1 |

**Job\_skills table**

A bridge table that displays correspondence of job\_id’s to skill\_id’s, i.e. connects the table job with skills tables. This helps identify what exact skills are being demanded in a particular job listing.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| job\_skills | job\_id | Id of the job, CFK/FK | Int |
| skill\_id | Id of the skill, CFK/FK | Int |

It has many-to-one relationship with skills and job tables

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| job\_skills.skill\_id | Skills.id | Job\_skills.job\_id | Job.id |
| 1 | 1 | 2 | 2 |

**Candidate\_skills table**

A bridge table that connects the table candidate with skills tables. This helps identify what exact skills a particular candidate has to offer

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| candidate\_skills | job\_id | Id of the job, CFK/FK | Int |
| cand\_id | Id of the candidate, CFK/FK | Int |

It has many-to-one relationship with skills and candidate tables

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| candidate\_skills.skill\_id | Skills.id | Candidate\_skills.cand\_id | Candidate\_id |
| 1 | 1 | 2 | 2 |

**Candidate table**

A table that stores data about all the candidates

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| candidate | id | Id of the job, CFK/FK | Int |
| Name | Name of the candidate | Text |
| Title | Job title | Text |
| Gender | Male or Female | Text |
| Location | Location id | Int |
| Experience | Entry | Intermediate | Senior | varchar |
| Pref\_id | Id of the preferences, FK | Int |
| Experience\_level\_id, FK | Id of the experience | Int |

It has one to many relationship with candidate\_skills and many to many relationship with services tables

**Job table**

A table that stores data about all the job listings

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| job | Id | Id of the job, PK, FK | Int |
| Title | Title of the job | Text |
| Experience\_level\_id | Id of the experience, FK | Int |
| Location | Location id, FK | Int |
| Interviews | Number of interviews | Int |
| Company\_id | Id of the company, FK | Int |
| Posting\_date | Date | Datetime |

It has one to many relationship with job\_skills and application tables, as well as one to one relationship with location table

**Location table**

A table that stores data about locations

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| location | Id | Id of the location, PK, FK | Int |
| City |  | Text |
| State |  | Text |
| Country |  | Text |
| Postcode |  | Int |
|  | Building\_number |  | Int |
|  | Office\_number |  | Int |
|  | Street |  | Text |

It has one to one relationship with job and company tables

**Application table**

A table that stores data about job applications made by the candidates

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Application | Id | Id of the application, PK | Int |
| candidate\_id | Id of the candidate who applied, FK | Int |
| Job\_id | Id of the job application made, FK | Int |
| date | date | Datitime |
| Is\_active | Is the application active or no | Boolean |

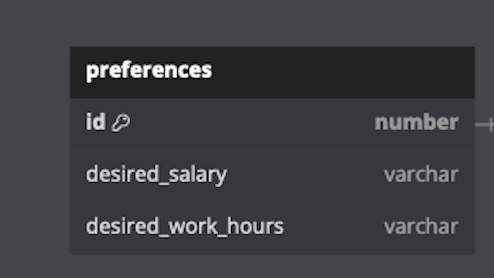
It has many-to-one relationship with job and candidate tables

**Placement table**

A table that stores data about job placemenets following an application

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Placement | Id | Id of the placement, PK | Int |
| App\_id | Id of the application, FK | Int |
| Salary\_offered | Salary offered | Int |
| Agent\_id | FK to Agent | int |

It has one-to-one relationship with application table



**Preferences table**

A table that stores data about candidate preferences

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Preferences | Id | Id of the preference, PK/FK | Int |
| Desired\_salary | Amount of salary range | Text |
| Desired\_work\_hours | Desired work hours in range | Int |

It has one-to-one relationship with candidate table

**Services table**

A table that stores data about services provided to candidates. Services include resume writing, coaching and skills development

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Services | Id | Id of the service, PK/FK | Int |
| Name | Name of the service | Text |

It has many-to-many relationship with candidate, and one-to-many relationship with resume\_service and coaching\_service tables

**Resume service table**

A table that stores data resume services

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Resume\_service | Service\_id | Id of the service, FK | Int |
| Order\_id | Id of the order, PK | Text |
| Order\_date | Date ordered | datetime |
| Is\_done | Whether finished | Boolean |
| Finish\_date | Date of finishing | datetime |

It has many-to-one relationship with services table

**Coaching service table**

A table that stores services like coaching or skills development

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Coaching\_service | Order\_id | Id of the order, PK | Int |
| Service\_id | Id of the service, FK | Text |
| Join\_date | Date when the candidate joined the service | datetime |
| Finish\_date | Date of finishing | datetime |

It has many-to-one relationship with the services table

**Company table**

A table that stores services like coaching or skills development

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Company | Id | Id of the company, PK | Int |
| name | Id of the service, FK | Text |
| Location | FK to location | int |

It has one-to-many relationship with job and one to one relationship with the location tables

**Agent table**

A table that stores data about the agent

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Gender | Id | Id of the agent | Int |
| Name |  | Text |

**Gender table**

A table that stores data about genders

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Gender | Id | Id of the gender, e.g. 0 or 1 | Int |
| Description | Male or Female | Text |

**Experience level**

A table that stores data about experience levels

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Experience\_level | Id | Id of the experience, e.g. 0, 1, 2 | Int |
| Description | e.g. Junior, Middle, Senior | Text |

**Comission table**

A table that stores data about the comissions

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Comission | Id | Id of the comission, PK | Int |
| Agent\_id | Id of the admin, FK | Int |
| Placement\_id | Id of the placement, FK | Int |
| Amount | Amount of commission | Int |
| Comission\_date | Date when commission is issued | datetime |