

|  |
| --- |
| Business Template  **Recruitment Agency**  Agency, jobs, recruitment, staffing, temporary icon - Download on Iconfinder |
|  |

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

The database serves as the backbone for a comprehensive recruitment agency that connects job seekers with employment opportunities across various industries. The agency aims to streamline the hiring process for both candidates and companies, ensuring efficient job placements while offering additional services to enhance candidates' employability.

## Problems. Current Situation

Problems:

Inefficient Matching: Difficulty in matching candidates to suitable job openings due to manual screening processes or limited candidate information.

High Turnaround Time: Lengthy hiring cycles leading to increased time-to-fill positions, causing frustration for both candidates and employers.

Limited Candidate Pool Management: Challenges in managing and organizing a vast pool of candidate profiles, skills, and experiences efficiently.

Ineffective Service Offerings: Inability to offer comprehensive services beyond job placements, such as skill development or resume writing, impacting the agency's competitiveness.

Data Disorganization: Difficulty in tracking applications, managing interviews, and placements due to scattered or unorganized candidate and job data.

Current Situation:

The agency might be experiencing delays in filling job openings due to the challenges in identifying suitable candidates quickly.

Limited resources and manual processes might be affecting the agency's ability to scale its services or efficiently manage candidate profiles and job listings.

There might be a lack of integrated systems to handle various aspects of recruitment, such as candidate assessment, interview scheduling, and placement tracking.

The agency might not offer a robust suite of support services to candidates beyond job placements, potentially impacting their competitiveness in the market.

## the Benefits of implementing a database. Project Vision

Efficiency: Streamlines the recruitment process for both candidates and employers, reducing time-to-hire and enhancing candidate-company matches.

Improved Candidate Experience: Provides candidates with support services to enhance their job-seeking skills and prospects.

Better Talent Pool Management: Helps in creating a robust database of qualified candidates and assists in efficient talent management.

Enhanced Service Offerings: Diversification of services beyond placements ensures comprehensive support for candidates, adding value to the agency's offerings.

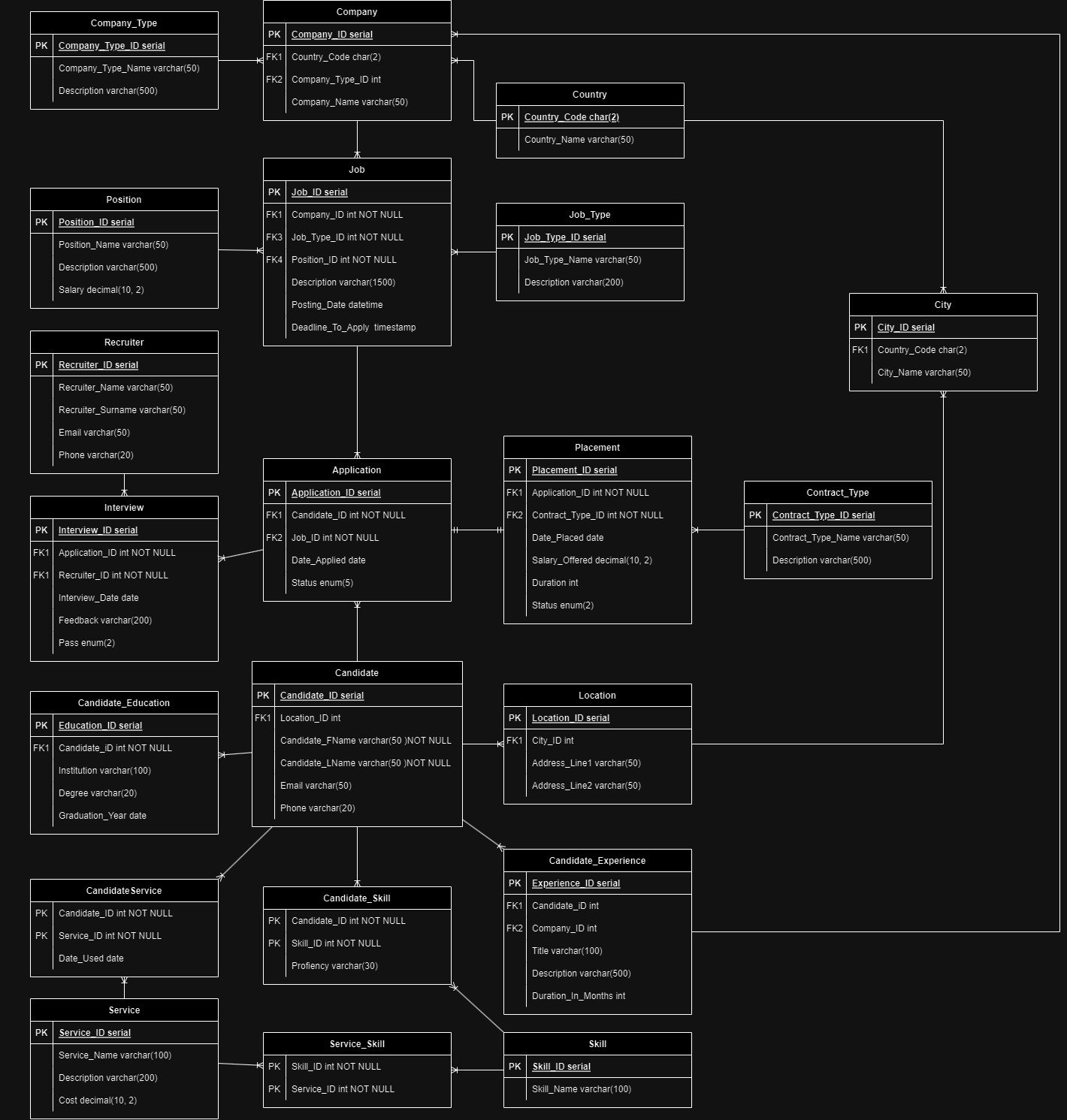
# Model description

## Definitions & Acronyms

PK – Primary Key

FK – Foreign Key

## Logical Scheme



## Objects

**Table Description 1**

< Stores information about available job positions offered by various companies. It includes details such as job descriptions, posting dates, deadlines, and the company associated with each job listing. This table helps manage and organize job postings for recruitment purposes >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Job | Job\_ID | < Unique identifier for each job listing >, PK | Serial |
| Company\_ID | < References the company offering the job >, FK1 | Int |
| Job\_Type\_ID | < References the type of job >, FK2 | Int |
| Position\_ID | < References the specific position for the job > | Int |
| Description | < Description of the job > | Varchar (1500) |
| Posting\_Date | < Date when the job was posted > | Date |
| Deadline\_To\_Apply | < Deadline for job application > | Timestamp |

Comments on table relationships:

1. Connected to the company offering the job. Many-to-one relationship with Company (company\_id).
2. Linked to the specific type of job. Many-to-one relationship with Job\_type (job\_type\_id).
3. Associated with the particular position. Many-to-one relationship with Position (position\_id).

Example with data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| job\_id | company\_id | job\_type\_id | position\_id | description | posting\_date | deadline\_to\_apply |
| 1 | 1 | 1 | 1 | Looking for an experienced Middle Full-stack Developer with skills in Java and Angular to join our development team. Your role will be to develop and support web applications, working both on the client side with Angular and on the server side using Java. | 2023-10-15 | 2023-11-15 |
| 2 | 2 | 2 | 2 | Join Analytics Innovations Inc. as a Data Scientist and drive cutting-edge data analysis projects to make an impact in the field of analytics | 2023-11-01 | 2023-12-01 |
| 3 | 3 | 1 | 3 | HR Connect Corp. is seeking a dedicated Human Resources Manager to lead our HR department and implement effective HR strategies for our organization | 2023-11-10 | 2023-12-10 |

**Table Description 2**

< Manages job applications submitted by candidates. It tracks application details such as candidate IDs, job IDs, application dates, and application statuses. This table helps in monitoring the progress of job applications and their statuses throughout the recruitment process >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Application | Application\_ID | < Unique identifier for each application >, PK | Serial |
| Candidate\_ID | < References the candidate who applied >, FK1 | Int |
| Job\_ID | < References the job the candidate applied for >, FK2 | Int |
| Date\_Applied | < Date of application > | Date |
| Status | < Status of the application, list of possible status: 'Pending', 'Under Review', 'Accepted', 'Rejected', 'Withdrawn' > | Enum ('Pending', 'Under Review', 'Accepted', 'Rejected', 'Withdrawn') |

Comments on table relationships:

1. Associated with the candidate who applied. Many-to-one relationship with Candidate (candidate\_id).
2. Connected to the specific job the candidate applied for. Many-to-one relationship with Job (job\_id).

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Application\_ID | Candidate | Job\_ID | Date\_Applied | Status |
| 1 | 301 | 1 | 2023-10-18 | In Progress |
| 2 | 302 | 2 | 2023-11-03 | Rejected |
| 3 | 303 | 3 | 2023-11-12 | Pending |

**Table Description 3**

< Stores details about scheduled interviews, including the interview dates and the associated job applications. This table assists in organizing and tracking interview schedules for different job applications as a bridge table between Application and Recruiter >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Interview | Interview \_ID | < Unique identifier for each interview >, PK | Serial |
| Application\_ID | < References the application associated with the interview >, FK1 | Int |
| Recruiter\_ID | < Unique identifier for each recruiter >, FK2 | Int |
| Interview\_Date | < Date of the interview > | Date |
| Feedback | < Feedback from the interview > | Varchar (200) |
| Pass | < Indicator if the candidate passed the interview > | Enum (‘Passed’, ‘Failed’) |

Comments on table relationships:

1. Linked to a specific application for an interview. Many-to-one relationship with Application (application\_id).
2. Linked to a specific recruiter involved in the interview. Many-to-one relationship with Recruiter(recruiter\_id.)

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Interview \_ID | Application\_ID | Recruiter\_ID | Interview\_Date | Feedback | Pass |
| 1 | 1 | 1 | 2023-11-05 | Feedback text | Passed |
| 2 | 3 | 2 | 2023-11-15 | Feedback text | Passed |
| 3 | 2 | 3 | 2023-11-10 | Feedback text | Failed |

**Table Description 4**

< Contains information about recruiters involved in the hiring process. It includes details like names, surnames, email addresses, and phone numbers of the recruiters. This table helps in maintaining a record of recruiters associated with various hiring activities >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Recruiter | Recruiter \_ID | < Unique identifier for each recruiter >, PK | Serial |
| Recruiter \_Name | < First name of the recruiter > | Varchar (50) |
| Recruiter \_Surname | < Last name of the recruiter > | Varchar (50) |
| Email | < Email address of the recruiter > | Varchar (50) |
| Phone | < Phone number of the recruiter > | Varchar (20) |

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Recruiter \_ID | Recruiter \_Name | Recruiter \_Surname | Email | Phone |
| 1 | John | Smith | john@example.com | +1234567890 |
| 2 | Emily | Johnson | emily@example.com | +1987654321 |
| 3 | Michael | Brown | michael@example.com | +1654321890 |

**Table Description 5**

< Records details about companies offering job positions. It includes information such as company names, associated country IDs, and company type IDs. This table serves as a repository for company data and aids in organizing company-related information for recruitment purposes >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Company | Company \_ID | < Unique identifier for each company >, PK | Serial |
| Country\_Code | < References the country where the company is located >, FK1 | Char(2) |
| Company\_Type\_ID | < References the type of company >, FK2 | int |
| Company\_Name | < Name of the company > | Varchar (50) |

Comments on table relationships:

1. Associated with the country it operates in. Many-to-one relationship with Country (country\_code).
2. Connected to a specific type of company. Many-to-one relationship with Company\_type (company\_type\_id).

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| Company \_ID | Country\_Code | Company\_Type\_ID | Company\_Name |
| 1 | KZ | 1 | ABC Solutions |
| 2 | RU | 2 | XYZ Marketing |
| 3 | TK | 1 | DataTech Innovations |

**Table Description 6**

< Stores details of candidates' work experiences, including job titles, descriptions, durations, and associated candidate IDs. This table helps in documenting candidates' professional experiences for better matching with job requirements >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Candidate\_Experience | Experience \_ID | < Unique identifier for each experience record >, PK | Serial |
| Candidate\_ID | < References the candidate with the experience >, FK1 | Int |
| Company\_iD | < References the company in which the candidate have worked >, FK2 | Int |
| Title | < Title or position in the company. > | Varchar (100) |
| Description | < Description of the experience > | Varchar (500) |
| Duration\_In\_Months | < Duration of experience in months > | Int |

Comments on table relationships:

1. Connected to the candidate who has the experience. Many-to-one relationship with Candidate (candidate\_id).
2. Connected to the company which related to the experience. One-to-many relationship with Company (company\_id).

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Experience \_ID | Candidate\_ID | Company\_ID | Title | Description | Duration\_In\_Months |
| 1 | 1 | 1 | Developer | Description 1 | 24 |
| 2 | 2 | 2 | Analyst | Description 2 | 18 |
| 3 | 3 | 3 | Designer | Description 3 | 36 |

**Table Description 7**

< Contains information about countries referenced in the database. It includes country codes and country names. This table serves as a reference to associate countries with specific entities, such as companies or candidates >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Country | Country \_Code | < Unique identifier for each country >, PK | Char(2) |
| Country\_Name | < Name of the country > | Varchar (50) |

Example with data

|  |  |
| --- | --- |
| Country \_ID | Country\_Name |
| KZ | Kazakhstan |
| RU | Russia |
| TK | Turkey |

**Table Description 8**

< Manages information about placements resulting from job applications. It includes details such as placement IDs, application IDs, contract type IDs, and placement dates. This table assists in tracking successful placements after the hiring process >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Placement | Placement \_ID | < Unique identifier for each placement >, PK | Serial |
| Application \_ID | < References the application leading to the placement >, FK1 | Int |
| Contract\_Type\_ID | < References the type of contract >, FK2 | Int |
| Date\_Placed | < Date when the candidate was placed > | Date |
| Salary\_Offered | < Salary offered in the contract > | Decimal (10,2) |
| Duration | < Duration of the contract, in months> | Int |
| Status | < Current status of the contract > | Enum (‘active’,‘inactive’) |

Comments on table relationships:

1. Associated with the application leading to placement. One-to-one relationship with Application (application\_id).
2. Linked to a specific type of contract. One-to-one relationship with Contract\_type (contract\_type\_id).

Example with data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Placement \_ID | Application \_ID | Contract\_Type\_ID | Date\_Placed | Salary\_Offered | Duration | Status |
| 1 | 1 | 1 | 2023-11-20 | 70000 | 12 | Active |
| 2 | 2 | 6 | 2023-11-25 | 84000 | 6 | Active |
| 3 | 3 | 8 | 2022-12-02 | 105000 | 12 | Inactive |

**Table Description 9**

< Stores details of candidates applying for jobs, including candidate names, surnames, email addresses, and phone numbers. This table serves as a repository for candidate information involved in the recruitment process >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Candidate | Candidate \_ID | < Unique identifier for each candidate >, PK | Serial |
| Location\_ID | < References the location of the candidate,FK1 | Int |
| Candidate\_FName | < First name of the candidate > | Varchar (50) |
| Candidate\_LName | < Last name of the candidate > | Varchar (50) |
| Email | < Email address of the candidate (unique) > | Varchar (50) |
| Phone | < Phone number of the candidate > | Varchar (20) |

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Candidate \_ID | Location\_ID | Candidate\_FName | Candidate\_LName | Email | Phone |
| 1 | 1 | Alice | Johnson | alice@example.com | +1234567890 |
| 2 | 2 | Bob | Smith | bob@example.com | +1987654321 |
| 3 | 3 | Clara | Brown | clara@example.com | +1654321890 |

**Table Description 10**

< Facilitates the many-to-many relationship between candidates and their skills. It links candidate IDs with skill IDs and includes proficiency levels. This table enables the association of multiple skills with multiple candidates >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Candidate\_Skill | Candidate \_ID | < References the candidate >, PK, FK | Int |
| Skill\_ID | < References the skill >, PK, FK | Int |
| Proficiency | < Level of proficiency in the skill for the candidate > | Varchar (30) |

Comments on table relationships:

1. Connects candidates to their respective skills. Many-to-many relationship between Candidate and Skill through Candidate\_skill (candidate\_id, skill\_id).

Example with data

|  |  |  |
| --- | --- | --- |
| Candidate \_ID | Skill\_ID | Proficiency |
| 1 | 1 | Expert |
| 2 | 2 | Intermediate |
| 2 | 3 | Advanced |

**Table Description 11**

< Contains a list of skills required for job positions. It includes skill IDs and skill names. This table serves as a reference for various skills that candidates possess or jobs require >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Skill | Skill \_ID | <description>, PK | Serial |
| Skill\_Name | < Name of the skill > | Varchar (100) |

Comments on table relationships:

1. Linked to candidates possessing the skill. One-to-many relationship with Candidate\_Skill (skill\_id).

Example with data

|  |  |
| --- | --- |
| Skill \_ID | Skill\_Name |
| 1 | Java |
| 2 | Django |
| 3 | Python |

**Table Description 12**

<Manages the relationship between candidates and additional services they've utilized, such as resume writing or interview coaching. It links candidate IDs with service IDs and dates when services were used>

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Candidate\_Service | Candidate\_ID | < References the candidate using a particular service >, PK, FK | Int |
| Service \_ID | < References the service >, PK, FK | Int |
| Date\_Used | < Date when the service was used > | Date |

Comments on table relationships:

1. Connects candidates to their acquired services. Many-to-many relationship between Candidate and Service through Candidate\_Service (candidate\_id, service\_id).

Example with data

|  |  |  |
| --- | --- | --- |
| Candidate\_ID | Service \_ID | Date\_Used |
| 1 | 1 | 2023-11-10 |
| 2 | 2 | 2023-11-15 |
| 3 | 1 | 2023-11-20 |

**Table Description 13**

< Stores information about additional services offered by the recruitment agency, including service IDs, associated skill IDs, service names, descriptions, and costs. This table aids in providing and managing additional services for candidates to enhance their job prospects >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Service | Service \_ID | < Unique identifier for each service >, PK | Serial |
| Service\_Name | < Name of the service > | Varchar (100) |
| Description | < Description of the service > | Varchar (200) |
| Cost | < Cost or pricing for the service, in USD > | Decimal (10,2) |

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| Service \_ID | Service\_Name | Description | Cost |
| 1 | Resume Writing | Description 1 | 50 |
| 2 | Interview Prep | Description 2 | 70 |
| 3 | Skill Training | Description 3 | 100 |

**Table Description 14**

< Contains various job types, such as full-time, part-time, or contract-based positions. It includes type IDs, job type names, and descriptions. This table categorizes different types of job positions available >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Job\_Type | Job\_Type\_ID | < Unique identifier for each job type >, PK | Serial |
| Job\_Type\_Name | < Name of the job type > | Varchar (50) |
| Description | < Description of the job type > | Varchar (200) |

Comments on table relationships:

1. Linked to various jobs falling under this type. One-to-many relationship with Job (job\_type\_id).

Example with data

|  |  |  |
| --- | --- | --- |
| Job\_Type\_ID | Job\_Type\_Name | Description |
| 1 | Full-time | Full-time employment |
| 2 | Part-time | Part-time employment |
| 3 | Contract | Contract-based work |

**Table Description 15**

< Manages specific positions within job listings, such as developer, marketer, or analyst. It includes position IDs, descriptions, and associated salary information. This table categorizes job positions based on their specific roles >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Position | Position\_ID | < Unique identifier for each position >, PK | Serial |
| Position \_Name | < Name of the position >, FK1 | Varchar (50) |
| Description | < Description of the position > | Varchar (500) |
| Salary | < Salary associated with the position > | Decimal (10,2) |

Comments on table relationships:

1. Linked to the jobs for specific positions. One-to-many relationship with Job (position\_id).

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| Position\_ID | Position \_Name | Description | Salary |
| 1 | Developer | Description 1 | 80000 |
| 2 | Marketing Specialist | Description 2 | 70000 |
| 3 | Data Scientist | Description 3 | 90000 |

**Table Description 16**

< Stores different types of companies, such as corporations, agencies, or startups. It includes company type IDs, type names, and descriptions. This table categorizes companies based on their organizational structure >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Company\_Type | Company\_Type\_ID | < Unique identifier for each company type >, PK | Serial |
| Company\_Type\_Name | < Name of the company type >, FK1 | Int |
| Description | < Description of the company type >, FK2 | Varchar (500) |

Comments on table relationships:

1. Linked to various companies falling under this type. One-to-many relationship with Company (company\_type\_id).

Example with data

|  |  |  |
| --- | --- | --- |
| Company\_Type\_ID | Company\_Type\_Name | Description |
| 1 | Corporation | Large-scale companies… |
| 2 | Agency | Marketing/HR agencies… |
| 3 | Startup | Newly established firms… |

**Table Description 17**

< Manages contract types associated with placements. It includes contract type IDs, contract names. Tracks different contract terms for placements >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Contract\_Type | Contract\_Type\_ID | < Unique identifier for each contract type >, PK | Serial |
| Contract\_Type\_Name | < Name of the contract type> | Varchar (50) |
| Description | < Description of the contract > | Varchar(200) |

Comments on table relationships:

1. Linked to the application associated with the contract. One-to-one relationship with Application (application\_id).

Example with data

|  |  |  |
| --- | --- | --- |
| Contract\_Type\_ID | Contract\_Type\_Name | Description |
| 1 | Full-time Contract | Description 1 |
| 2 | Part-time Contract | Description 2 |
| 3 | Temp Contract | Description 3 |

**Table Description 18**

< Stores educational background information of candidates, including education IDs, candidate IDs, institutions attended, degrees earned, and graduation years. This table records candidates' educational history for employment consideration >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Candidate\_Education | Education\_ID | < Unique identifier for each education record >, PK | Serial |
| Candidate\_ID | < References the candidate's education record >, FK1 | Int |
| Institution | < Name of the institution > | Varchar (100) |
| Degree | < Degree obtained > | Varchar (20) |
| Graduation\_Year | < Graduation year > | Date |

Comments on table relationships:

1. Connected to the candidate's educational history. Many-to-one relationship with Candidate (candidate\_id).

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Education\_ID | Candidate\_ID | Institution | Degree | Graduation\_Year |
| 1 | 1 | ABC University | BSc IT | 2018 |
| 2 | 2 | XYZ College | BBA | 2019 |
| 3 | 3 | DataTech Institute | MSc CompSci | 2017 |

**Table Description 19**

< Represents the relationship between services and skills, indicating which skills are associated with specific services >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Service\_Skill | Skill\_ID | < References the skill associated with a particular service >, PK, FK | Int |
| Service\_ID | < References the service related to a specific skill >, PK, FK | Int |

Comments on table relationships:

1. Many-to-many relationship acting as a bridge table between Service (service\_id) and Skill (skill\_id).

Example with data

|  |  |
| --- | --- |
| Skill\_ID | Service\_ID |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

**Table Description 20**

< Contains information about cities referenced in the database. It includes country codes and city names. This table serves as a reference to associate cities with specific locations >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| City | City\_ID | < Unique identifier for each city >, PK | Int |
| Country\_Code | < References the country related to a city >, FK | Char (2) |
| City\_Name | < Stores the name of the city > | Varchar (50) |

Comments on table relationships:

1. Many-to-one relationship referenced with the Country(country\_code).

Example with data

|  |  |  |
| --- | --- | --- |
| City\_ID | Country\_Code | City\_Name |
| 1 | KZ | Astana |
| 2 | RU | Moscow |
| 3 | TK | Istanbul |

**Table Description 21**

< Stores specific location details for candidates. It includes location IDs, associated cities, and address names >

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Location | Location\_ID | < Unique identifier for each location entry >, PK | Int |
| City\_ID | < References the city related to the location >, FK1 | Int |
| Address Line 1 | < Represents the primary line of the address for a specific location > | Varchar (50) |
| Address Line 2 | < Represents a secondary line of the address, providing additional details if needed > | Varchar (50) |

Comments on table relationships:

1. Linked to the specific city of the location. One-to-many relationship with City(city\_id).

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| Location\_ID | City\_ID | Address\_Line1 | Address\_Line2 |
| 1 | 1 | 123 Main Street | Apt 301 |
| 2 | 2 | 4A Park Avenue | Suite 500 |
| 3 | 3 | 17 Baker Street | Block C |