**ERD for JOB PORTAL-1**

Creating an Entity-Relationship Diagram (ERD) for a job portal management system is a great way to visualize the database structure. A professional ERD focuses on clarity and key relationships. Below is a detailed breakdown of the entities, their attributes, and the relationships between them.

Here is a professional ERD for a job portal system, broken down into its core components.

**Entities and Attributes**

The ERD for a job portal will typically have several main entities, each representing a key component of the system.

**User**

This entity can represent both job seekers and recruiters, with a role attribute to differentiate them. This is a common practice to avoid duplicate tables and simplify management.

* **user\_id** (Primary Key, unique identifier for each user)
* **email** (Unique, used for login)
* **password\_hash** (Securely stored password)
* **first\_name**
* **last\_name**
* **phone\_number**
* **role** (e.g., 'Job Seeker', 'Recruiter')
* **created\_at** (Timestamp of account creation)
* **last\_login** (Timestamp of the last login)

**Job Seeker Profile**

This entity holds specific information for job seekers. It has a one-to-one relationship with the **User** entity.

* **profile\_id** (Primary Key)
* **user\_id** (Foreign Key, links to the **User** entity)
* **resume\_file\_path** (Path to the uploaded resume)
* **bio** (Short biography or summary)
* **skills** (e.g., as a JSON or a separate skills table)
* **education**
* **experience**
* **location**
* **is\_public** (Boolean, for privacy)

**Company**

Represents the companies that are posting job listings.

* **company\_id** (Primary Key)
* **company\_name** (Unique)
* **industry**
* **website**
* **description**
* **headquarters\_location**

**Recruiter**

Represents the individuals at a company who manage job postings. It has a many-to-one relationship with the **Company** entity.

* **recruiter\_id** (Primary Key)
* **user\_id** (Foreign Key, links to the **User** entity)
* **company\_id** (Foreign Key, links to the **Company** entity)
* **title** (e.g., 'Hiring Manager', 'Recruiter')

**Job Listing**

The core entity for a job portal, containing all the details about an open position.

* **job\_id** (Primary Key)
* **recruiter\_id** (Foreign Key, links to the **Recruiter** entity)
* **company\_id** (Foreign Key, links to the **Company** entity)
* **title** (Job title, e.g., 'Software Engineer')
* **description** (Full job description)
* **requirements**
* **location**
* **job\_type** (e.g., 'Full-time', 'Part-time', 'Contract')
* **salary\_range**
* **posted\_date** (Timestamp)
* **expiration\_date**
* **status** (e.g., 'Open', 'Closed', 'Filled')

**Application**

This entity tracks a job seeker's application to a specific job. It's the linking entity between **Job Seeker Profile** and **Job Listing**.

* **application\_id** (Primary Key)
* **job\_id** (Foreign Key, links to the **Job Listing** entity)
* **job\_seeker\_id** (Foreign Key, links to the **Job Seeker Profile** entity)
* **applied\_date** (Timestamp)
* **status** (e.g., 'Pending', 'In Review', 'Interview', 'Rejected', 'Hired')
* **cover\_letter** (Optional)

**Relationships**

The relationships define how the entities connect to each other.

* **One-to-One (1:1):**
  + **User** and **Job Seeker Profile**: A single user has one unique job seeker profile.
  + **User** and **Recruiter**: A single user has one unique recruiter profile.
* **One-to-Many (1:N):**
  + **Company** and **Recruiter**: One company can have multiple recruiters.
  + **Company** and **Job Listing**: One company can post many job listings.
  + **Recruiter** and **Job Listing**: One recruiter can post many job listings.
  + **Job Seeker Profile** and **Application**: One job seeker can submit many applications.
* **Many-to-Many (M:N):**
  + **Job Seeker Profile** and **Job Listing**: A job seeker can apply to many jobs, and a job can be applied to by many job seekers. This relationship is resolved by the **Application** entity.

**ERD Diagram Visualization**

You can visualize this structure using standard ERD notation.

* **Boxes** for entities.
* **Lines** for relationships.
* **Crow's foot notation** to indicate the type of relationship (one, many).

Here's a simplified text representation of the diagram:

+------------------+ +--------------------------+

| User |-----| Job Seeker Profile |

| (PK) user\_id | 1:1 | (PK) profile\_id |

| email | | (FK) user\_id |

| ... | | ... |

+------------------+ +--------------------------+

| 1:1

|

| +------------------+

|------| Recruiter |

| | (PK) recruiter\_id|

| | (FK) user\_id |

| | (FK) company\_id |

| | ... |

+------+------------------+

| 1:N

|

+------------------+ +--------------------------+

| Company |---| Job Listing |

| (PK) company\_id | 1:N| (PK) job\_id |

| company\_name | | (FK) recruiter\_id |

| ... | | (FK) company\_id |

+------------------+ | ... |

+--------------------------+

| N:M

|

+-------------------+--------------------+

| Application |

| (PK) application\_id |

| (FK) job\_id |

| (FK) job\_seeker\_id |

| applied\_date, status, ... |

+----------------------------------------+

This model is a solid, professional foundation for a job portal system. It's scalable and clearly defines the relationships between the key components, which is crucial for database design.

**ERD for JOB PORTAL-2**

Here’s a professional ERD (Entity‑Relationship Diagram) outline for a **Job Portal Management System**, plus a suggested visual layout (you can draw it with tools like draw.io, Lucidchart, etc.). You can adapt it to your exact needs.

**🧱 Key Entities & Relationships**

Here are the main entities you might need, with their essential attributes and the relationships among them.

|  |  |  |
| --- | --- | --- |
| Entity | Key Attributes | Notes |
| User | user\_id (PK), name, email, password\_hash, role (e.g. job\_seeker, employer, admin), is\_active, created\_at, updated\_at | Roles control permissions. One person might be both employer & seeker (if allowed). |
| Employer | employer\_id (PK), user\_id (FK → User), company\_name, address, website, contact\_number, logo, is\_verified, created\_at, updated\_at | Only for users that are employers. |
| Job\_Seeker\_Profile | seeker\_id (PK), user\_id (FK → User), resume, bio, skills, experience\_years, education, profile\_photo, created\_at, updated\_at | Only for users that are job seekers. |
| Job\_Category | category\_id (PK), name, description, is\_active |  |
| Job\_Location | location\_id (PK), city, state, country, postal\_code |  |
| Job | job\_id (PK), employer\_id (FK → Employer), category\_id (FK → Job\_Category), location\_id (FK → Job\_Location), title, description, job\_type (full‑time / part‑time / contract), salary\_min, salary\_max, posted\_date, application\_deadline, num\_vacancies, status (open / closed / paused) |  |
| Application | application\_id (PK), job\_id (FK → Job), seeker\_id (FK → Job\_Seeker\_Profile), applied\_date, status (pending / reviewed / accepted / rejected), resume\_submitted, cover\_letter, created\_at, updated\_at |  |
| Interview | interview\_id (PK), application\_id (FK → Application), scheduled\_date, mode (online / onsite), interviewer\_details, result (pass / fail / pending), notes |  |
| Skill | skill\_id (PK), name |  |
| Seeker\_Skill | seeker\_id (FK → Job\_Seeker\_Profile), skill\_id (FK → Skill) → Composite PK (seeker\_id, skill\_id) | Many‑to‑many: a seeker can have many skills. |
| Employer\_Package | package\_id (PK), employer\_id (FK → Employer), name, price, duration\_days, features, start\_date, end\_date, status |  |
| Notification (optional) | notification\_id (PK), user\_id (FK → User), type, message, is\_read, created\_at |  |

**🔗 Relationships**

* A **User** can have one **Employer** profile *or* one **Job\_Seeker\_Profile**, depending on their role (or possibly both, if your system allows).
* An **Employer** can post many **Jobs**.
* A **Job** belongs to exactly one **Job\_Category** and one **Job\_Location**.
* A **Job\_Seeker\_Profile** can submit many **Applications**, each referring to a particular **Job**.
* Each **Application** can lead to zero or more **Interviews**.
* **Skills** are many‑to‑many with **Job\_Seeker\_Profile** via **Seeker\_Skill**.
* **Employer\_Package** is associated with **Employer** to track paid status/features.
* **Notifications** connect to **User**.

**📐 Visual Layout Suggestion**

When drawing it, here’s a suggested layout:

[User]──< 1 ‑ to ‑ 1 >──[Job\_Seeker\_Profile]──< \* ‑ to ‑ \* >──[Skill]

│

│──< 1 ‑ to ‑ 1 >──[Employer]──< 1 ‑ to ‑ \* >──[Job]──< \* ‑ to ‑ \* >──[Application]──< \* ‑ to ‑ \* >──[Interview]

│ │

│ └──< part of >──[Job\_Category]

│ └──< part of >──[Job\_Location]

│

└──< 1 ‑ to ‑ \* >──[Employer\_Package]

[User]──< 1 ‑ to ‑ \* >──[Notification]

* “1 to 1” means one record in one table maps to at most one in the other.
* “1 to \*” means one → many.
* “\* to \*” indicates many‑to‐many (resolved via a join table like Seeker\_Skill).