**Users Table**

The Users table stores information about all users in the system.

* **user\_id**: A unique identifier for each user (Primary Key).
* **username**: The unique username for the user (required).
* **email**: The unique email address of the user (required).
* **password**: The hashed password for the user (required).
* **full\_name**: The full name of the user (required).
* **role\_id**: A foreign key linking to the Roles table.

**Roles Table**

The Roles table defines different roles that users can have.

* **role\_id**: A unique identifier for each role (Primary Key).
* **name**: The name of the role (e.g., Manager, HR, Team Leader, Applicant) (required).

**Teams Table**

The Teams table represents different teams within the organization

* **team\_id**: A unique identifier for each team (Primary Key).
* **team\_name**: The name of the team (required).

**TeamMemberships Table**

The TeamMemberships table links users to teams and roles within the teams.

* **membership\_id**: A unique identifier for each membership (Primary Key).
* **user\_id**: A foreign key linking to the Users table.
* **team\_id**: A foreign key linking to the Teams table.

**JobPosts Table**

The JobPosts table contains information about job openings.

* **job\_id**: A unique identifier for each job post (Primary Key).
* **title**: The title of the job (required).
* **description**: A detailed description of the job.
* **criteria**: Criteria for the job in JSON format.{
* educational level
* experience
* required employee
* gender
* }
* **category\_id**: A foreign key linking to the Categories table.

**Applications Table**

The Applications table tracks applications submitted by applicants for job postings.

* **application\_id**: A unique identifier for each application (Primary Key).
* **status**: The current status of the application (e.g., Submitted, Reviewed, Interviewed, Accepted, Rejected).
* **interview\_score**: The score given by the team leader after interviewing (nullable).
* **resume\_url**: URL to the applicant's uploaded resume or CV.
* **phone\_number**: The applicant's phone number.
* **education**: Structured data about the applicant's educational background in JSON format.
* **experience**: Structured data about the applicant's work experience in JSON format.
* **skills**: A list of skills or competencies provided by the applicant in JSON format.

**Categories Table**

The Categories table contains different categories for job posts.

* **category\_id**: A unique identifier for each category (Primary Key).
* **category\_name**: The name of the category (required).

**PostedBy Table**

The PostedBy table manages the relationship between job posts, teams, and applicants.

* **post\_id**: A unique identifier for each entry (Primary Key).
* **job\_id**: A foreign key linking to the JobPosts table.
* **team\_id**: A foreign key linking to the Teams table.
* **applicant\_id**: A foreign key linking to the Users table.

**Relationships**

* **Users** table is related to the **Teams** and **Roles** tables through the **TeamMemberships** table.
* **Teams** table has a foreign key leader\_id linking to the **Users** table.
* **TeamMemberships** table has foreign keys user\_id linking to the **Users** table, team\_id linking to the **Teams** table, and role\_id linking to the **Roles** table.
* **JobPosts** table has a foreign key category\_id linking to the **Categories** table.
* **Applications** table has no foreign keys in this definition but can be extended to include foreign keys linking to **Users** and **JobPosts**.
* **PostedBy** table has foreign keys job\_id linking to the **JobPosts** table, team\_id linking to the **Teams** table, and applicant\_id linking to the **Users** table.

### Key Elements of a Permission Table

Here’s what you can include in your Permission table:

1. **permission\_id**: A unique identifier for each permission.
2. **action**: The type of action the permission allows (e.g., create, read, update, delete).
3. **resource**: The resource the action applies to (e.g., job\_post, applicant, team).
4. **description** (optional): A short description of what the permission does (e.g., "Allows a user to create a new job post").
5. **roleId**: A foreign key linking the permission to a specific role. You may also decide to have a many-to-many relationship between roles and permissions through a join table.

### Example of a Permission Table

| **permission\_id** | **action** | **resource** | **description** |
| --- | --- | --- | --- |
| 1 | create | job\_post | Allows creation of job posts |
| 2 | read | applicant | Allows reading applicant data |
| 3 | update | team | Allows updating team details |
| 4 | delete | job\_post | Allows deletion of job posts |

### Many-to-Many Relationship: Roles and Permissions

To implement this efficiently, you would typically create a **join table** (or association table) between Roles and Permissions. This way, you can assign multiple permissions to a single role and assign multiple roles to a single permission.

#### Join Table: Role\_Permissions

| **role\_id** | **permission\_id** |
| --- | --- |
| 1 | 1 |
| 1 | 2 |
| 2 | 2 |
| 3 | 3 |
| 1 | 4 |