#### **Сущности**

1. **Пользователь (User)**
   * id: SERIAL PRIMARY KEY
   * username: VARCHAR(50) UNIQUE NOT NULL
   * password: VARCHAR(255) NOT NULL
   * role: VARCHAR(20) NOT NULL (например, 'admin' или 'recruiter')
2. **Типы вакансий (Job\_Type)**
   * id: SERIAL PRIMARY KEY
   * type\_name: VARCHAR(50) NOT NULL (например, ‘QA jun’, Python middle)
3. **Вакансия (Job)**
   * id: SERIAL PRIMARY KEY
   * title: VARCHAR(100) NOT NULL
   * description: TEXT NOT NULL
   * job\_type\_id: INTEGER REFERENCES Job\_Type(id)
   * status: VARCHAR(20) NOT NULL (например, 'open', 'closed')
   * created\_at: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
   * closed\_at: TIMESTAMP
   * recruiter\_id: INTEGER REFERENCES User(id)
4. **Кандидат (Candidate)**
   * id: SERIAL PRIMARY KEY
   * name: VARCHAR(100) NOT NULL
   * email: VARCHAR(100) NOT NULL
   * resume: TEXT
   * vacancy\_id: INTEGER REFERENCES Job(id)
5. **Этап (Stage)**
   * id: SERIAL PRIMARY KEY
   * name: VARCHAR(50) NOT NULL (например, 'prescreening', 'interview', 'offer')
   * job\_id: INTEGER REFERENCES Job(id)
6. **Кандидат на этапе (Candidate\_Stage)**
   * id: SERIAL PRIMARY KEY
   * candidate\_id: INTEGER REFERENCES Candidate(id)
   * stage\_id: INTEGER REFERENCES Stage(id)
   * updated\_at: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
7. **Аналитика (Analytics)**
   * id: SERIAL PRIMARY KEY
   * job\_id: INTEGER REFERENCES Job(id)
   * stage\_id: INTEGER REFERENCES Stage(id)
   * candidate\_count: INTEGER NOT NULL
8. **Лог действий (Action\_Log)**
   * id: SERIAL PRIMARY KEY
   * user\_id: INTEGER REFERENCES User(id)
   * action: VARCHAR(255) NOT NULL
   * created\_at: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
9. **Отчет (Report)**
   * id: SERIAL PRIMARY KEY
   * user\_id: INTEGER REFERENCES User(id)
   * job\_id: INTEGER REFERENCES Job(id)
   * created\_at: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP
   * content: TEXT NOT NULL (отчет о производительности, количестве кандидатов и т.д.)
10. **Комментарий к кандидату (Candidate\_Comment)**
    * id: SERIAL PRIMARY KEY
    * candidate\_id: INTEGER REFERENCES Candidate(id)
    * recruiter\_id: INTEGER REFERENCES User(id)
    * comment: TEXT (отзывы рекрутера о кандидате)
    * created\_at: TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

### **ER**

Вот описание ER-структуры, которое охватывает все предложенные сущности и их связи между собой. Эта структура основывается на задачах автоматизации системы подбора персонала (ATS), где каждая сущность выполняет свою роль.

### **ER-структура и связи между сущностями:**

1. **Пользователь (User)**
   * **Поля**:
     + id (PK)
     + username
     + password
     + role (например, 'admin', 'recruiter')
   * **Связи**:
     + **Вакансии**: Пользователь может быть рекрутером для вакансий. Связь: User → Job (recruiter\_id)
     + **Лог действий**: Пользователь осуществляет действия, которые записываются в журнале. Связь: User → Action\_Log (user\_id)
     + **Комментарии к кандидату**: Пользователь (рекрутер) может оставлять комментарии к кандидатам. Связь: User → Candidate\_Comment (recruiter\_id)
     + **Отчеты**: Пользователь может создавать отчеты по вакансиям. Связь: User → Report (user\_id)
2. **Типы вакансий (Job\_Type)**
   * **Поля**:
     + id (PK)
     + type\_name
   * **Связи**:
     + **Вакансии**: Тип вакансии определяет категорию для каждой вакансии. Связь: Job\_Type → Job (job\_type\_id)
3. **Вакансия (Job)**
   * **Поля**:
     + id (PK)
     + title
     + description
     + job\_type\_id (FK на Job\_Type)
     + status
     + created\_at
     + closed\_at
     + recruiter\_id (FK на User)
   * **Связи**:
     + **Тип вакансии**: Вакансия относится к определённому типу вакансий. Связь: Job → Job\_Type
     + **Рекрутер**: Вакансия назначается определённому рекрутеру (пользователю). Связь: Job → User (recruiter\_id)
     + **Кандидаты**: Вакансия связана с кандидатами, поданными на неё. Связь: Job → Candidate (vacancy\_id)
     + **Этапы процесса рекрутинга**: Вакансия проходит через различные этапы. Связь: Job → Stage (job\_id)
     + **Аналитика**: Вакансия используется для расчета аналитики по этапам и кандидатам. Связь: Job → Analytics (job\_id)
     + **Отчеты**: Вакансия может быть предметом отчёта. Связь: Job → Report (job\_id)
4. **Кандидат (Candidate)**
   * **Поля**:
     + id (PK)
     + name
     + email
     + resume
     + vacancy\_id (FK на Job)
   * **Связи**:
     + **Вакансия**: Кандидат подан на определённую вакансию. Связь: Candidate → Job (vacancy\_id)
     + **Этапы**: Кандидат может находиться на определённых этапах рекрутингового процесса. Связь: Candidate → Candidate\_Stage (candidate\_id)
     + **Комментарии**: Кандидат может иметь комментарии от рекрутеров. Связь: Candidate → Candidate\_Comment (candidate\_id)
5. **Этап (Stage)**
   * **Поля**:
     + id (PK)
     + name
     + job\_id (FK на Job)
   * **Связи**:
     + **Вакансия**: Этап привязан к конкретной вакансии. Связь: Stage → Job (job\_id)
     + **Кандидаты на этапе**: Каждый этап содержит информацию о кандидатах на нём. Связь: Stage → Candidate\_Stage (stage\_id)
     + **Аналитика**: Этапы участвуют в аналитике по вакансиям. Связь: Stage → Analytics (stage\_id)
6. **Кандидат на этапе (Candidate\_Stage)**
   * **Поля**:
     + id (PK)
     + candidate\_id (FK на Candidate)
     + stage\_id (FK на Stage)
     + updated\_at
   * **Связи**:
     + **Кандидаты**: Сущность отслеживает на каком этапе находится кандидат. Связь: Candidate\_Stage → Candidate (candidate\_id)
     + **Этапы**: Сущность хранит информацию, на каком этапе находится кандидат. Связь: Candidate\_Stage → Stage (stage\_id)
7. **Аналитика (Analytics)**
   * **Поля**:
     + id (PK)
     + job\_id (FK на Job)
     + stage\_id (FK на Stage)
     + candidate\_count
   * **Связи**:
     + **Вакансия**: Сущность аналитики связывается с вакансиями для сбора данных по каждому этапу и количеству кандидатов. Связь: Analytics → Job (job\_id)
     + **Этап**: Сущность связывается с конкретными этапами, чтобы вести учёт количества кандидатов на каждом этапе. Связь: Analytics → Stage (stage\_id)
8. **Лог действий (Action\_Log)**
   * **Поля**:
     + id (PK)
     + user\_id (FK на User)
     + action
     + created\_at
   * **Связи**:
     + **Пользователь**: Лог записывает действия, совершённые пользователем (например, создание вакансии или этапа). Связь: Action\_Log → User (user\_id)
9. **Отчет (Report)**
   * **Поля**:
     + id (PK)
     + user\_id (FK на User)
     + job\_id (FK на Job)
     + created\_at
     + content
   * **Связи**:
     + **Пользователь**: Отчеты создаются пользователями. Связь: Report → User (user\_id)
     + **Вакансия**: Отчеты могут содержать информацию по конкретным вакансиям. Связь: Report → Job (job\_id)
10. **Комментарий к кандидату (Candidate\_Comment)**

* **Поля**:
  + id (PK)
  + candidate\_id (FK на Candidate)
  + recruiter\_id (FK на User)
  + comment
  + created\_at
* **Связи**:
  + **Кандидаты**: Комментарии относятся к конкретному кандидату. Связь: Candidate\_Comment → Candidate (candidate\_id)
  + **Рекрутер**: Комментарии оставляются рекрутерами. Связь: Candidate\_Comment → User (recruiter\_id)