Tables

---

1. docAuditLog

- **Description: Stores audit logs for document modifications.**
- **Columns:**
  - **doc_ID (int): The ID of the modified document.**
  - **JS_ID (int): The Job Seeker ID associated with the document.**
  - **actType (varchar): The type of action performed (e.g., 'Modify').**
  - **docTimeStamp (datetime): The timestamp when the action was performed.**
- **Purpose: To track changes made to documents and provide an audit trail.**

---

2. addrs

- **Description: Stores addresses associated with job seekers or employers.**
- **Columns:**
  - **addr_ID (int): Primary key for the address.**
  - **JS_ID or employer_ID (int): Foreign key referencing a job seeker or employer.**
  - **street, city, postal_code, country (varchar): Address details.**
- **Purpose: To store addresses linked to job seekers or employers.**

---

3. job_seeker

- **Description: Stores information about job seekers.**
- **Columns:**
  - **JS_ID (int): Primary key for job seekers.**
  - **F_name, L_name (nvarchar): Job seeker's first and last names.**
  - **citizenship, email (nvarchar): Citizenship and email details.**
- **Purpose: To store and manage job seeker profiles.**

---

4. profile_link

- **Description: Stores links to job seeker profiles on external platforms.**

- **Columns:**
  - **JS_ID (int): Foreign key referencing the job seeker.**
  - **platform_name (varchar): Name of the platform (e.g., LinkedIn).**
  - **profile_url (varchar): URL to the job seeker's profile.**
- **Purpose: To manage job seeker profiles on external platforms.**

---

5. work_exp

- **Description: Stores job seekers' work experience details.**
- **Columns:**
  - **work_exp_ID (int): Primary key for work experience records.**
  - **JS_ID (int): Foreign key referencing the job seeker.**
  - **company_name (nvarchar): Name of the company.**
  - **role, start_date, end_date (nvarchar, date): Role and time period of employment.**
- **Purpose: To store and manage job seekers' work experience history.**

---

6. degree

- **Description: Stores educational degrees held by job seekers.**
- **Columns:**
  - **degree_ID (int): Primary key for degree records.**
  - **JS_ID (int): Foreign key referencing the job seeker.**
  - **degree_name, institution, year_obtained (nvarchar, date): Details of the degree.**
- **Purpose: To store job seekers' educational qualifications.**

---

7. qualification

- **Description: Stores additional qualifications (e.g., certifications) held by job seekers.**
- **Columns:**
  - **qual_ID (int): Primary key for qualification records.**
  - **JS_ID (int): Foreign key referencing the job seeker.**

  - **qual_name (nvarchar): Name of the qualification.**
  - **certifying_body (nvarchar): Organization that issued the qualification.**
- **Purpose: To store certifications and qualifications for job seekers.**

---

8. company

- **Description: Stores company details.**
- **Columns:**
  - **company_ID (int): Primary key for companies.**
  - **company_name, industry, location (nvarchar): Company details.**
- **Purpose: To manage company profiles for job listings.**

---

9. employer

- **Description: Stores employer details.**
- **Columns:**
  - **employer_ID (int): Primary key for employers.**
  - **company_ID (int): Foreign key referencing the associated company.**
  - **F_name, L_name, email (nvarchar): Employer's first name, last name, and email.**
- **Purpose: To store and manage employer information.**

---

10. job_listings

- **Description: Stores job listings posted by employers.**
- **Columns:**
  - **job_ID (int): Primary key for job listings.**
  - **title, job_desc, industry (nvarchar): Job title, description, and industry.**
  - **employer_ID (int): Foreign key referencing the employer.**
- **Purpose: To manage job postings by employers.**

---

11. app

- **Description: Stores applications submitted by job seekers for job listings.**
- **Columns:**
  - **app_ID (int): Primary key for applications.**
  - **JS_ID (int): Foreign key referencing the job seeker.**
  - **job_ID (int): Foreign key referencing the job listing.**
  - **doc_ID (int): Foreign key referencing the submitted document.**
- **Purpose: To manage job applications.**

---

12. team

- **Description: Stores information about teams within a company.**
- **Columns:**
  - **team_ID (int): Primary key for teams.**
  - **company_ID (int): Foreign key referencing the company.**
  - **team_name (nvarchar): Name of the team.**
- **Purpose: To manage teams within companies.**

---

13. documents

- **Description: Stores documents uploaded by job seekers (e.g., resumes, certifications).**
- **Columns:**
  - **doc_ID (int): Primary key for documents.**
  - **JS_ID (int): Foreign key referencing the job seeker.**
  - **qual_ID (int): Foreign key referencing the qualification (optional).**
  - **doc_name (varchar): Document name.**
  - **doc (varbinary): Binary content of the document.**
- **Purpose: To manage documents uploaded by job seekers.**

---

14. team_member

- **Description: Stores members of teams within a company.**
- **Columns:**

  - **team_member_ID (int): Primary key for team members.**
  - **team_ID (int): Foreign key referencing the team.**
  - **employer_ID (int): Foreign key referencing the employer.**
- **Purpose: To manage team memberships within companies.**

---

15. app_workEXP

- **Description: Stores job seeker work experience relevant to specific job applications.**
- **Columns:**
  - **app_workEXP_ID (int): Primary key for work experience records linked to applications.**
  - **app_ID (int): Foreign key referencing the job application.**
  - **work_exp_ID (int): Foreign key referencing the work experience.**
- **Purpose: To link relevant work experience to job applications.**

---

16. app_qual

- **Description: Stores job seeker qualifications relevant to specific job applications.**
- **Columns:**
  - **app_qual_ID (int): Primary key for qualification records linked to applications.**
  - **app_ID (int): Foreign key referencing the job application.**
  - **qual_ID (int): Foreign key referencing the qualification.**
- **Purpose: To link relevant qualifications to job applications.**

---

17. job_qual

- **Description: Stores qualifications required for specific job listings.**
- **Columns:**
  - **job_qual_ID (int): Primary key for job qualification records.**
  - **job_ID (int): Foreign key referencing the job listing.**
  - **qual_ID (int): Foreign key referencing the qualification.**
- **Purpose: To link required qualifications to job listings.**

---

18. job_req

- **Description: Stores job requirements for specific job listings.**
- **Columns:**
  - **job_req_ID (int): Primary key for job requirement records.**
  - **job_ID (int): Foreign key referencing the job listing.**
  - **requirement (nvarchar): The requirement description.**
- **Purpose: To manage job requirements for job listings.**

Stored Procedures

---

1. GetApplicantDocuments

- **Description**: Retrieves the documents for a specific job seeker based on their ID.
- **Parameters**:
  - @JS_ID (int): The ID of the job seeker.
- **Functionality**: Selects document information (doc_ID, doc_name, and doc content) for the given job seeker from the documents table.
- **Result**: A list of documents linked to the job seeker.

---

2. GetApplicants

- **Description**: Retrieves all applicant information along with associated job seeker details.
- **Parameters**: None.
- **Functionality**: Joins the app table (applications) and job_seeker table to select the applicant details, including app_ID, job seeker information (JS_ID, first name, last name, citizenship, and email).
- **Result**: A list of all applicants with job seeker details.

---

3. GetJobListings

- **Description**: Retrieves all available job listings.
- **Parameters**: None.
- **Functionality**: Selects job listing details such as job_ID, title, job_desc, industry, and employer_ID from the job_listings table and orders the results by job_ID.
- **Result**: A list of all job listings ordered by job ID.

---

4. GetJobSeekerInfo

- **Description**: Retrieves personal information for a specific job seeker.
- **Parameters**:
  - @JS_ID (int): The ID of the job seeker.
- **Functionality**: Selects the first name, last name, citizenship, and email of the job seeker based on the provided JS_ID.
- **Result**: Personal details of the job seeker.

---

5. sp_CreateJobSeeker

- **Description**: Inserts a new job seeker into the system.
- **Parameters**:
  - @F_name (nvarchar): First name of the job seeker.
  - @L_name (nvarchar): Last name of the job seeker.
  - @citizenship (nvarchar): Citizenship of the job seeker.
  - @email (nvarchar): Email address of the job seeker.
  - @JS_ID (int, OUTPUT): Job seeker ID that is output after the insertion.
- **Functionality**: Checks if the job seeker already exists based on their first name, last name, and email. If not, inserts a new job seeker record and returns the new JS_ID.
- **Result**: A new job seeker is added if they don't already exist, and their ID is returned.

---

6. sp_UploadDocument

- **Description**: Uploads a document for a job seeker and associates it with a specific qualification.
- **Parameters**:

  - @JS_ID (int): The ID of the job seeker.
  - @qual_ID (int): The qualification ID (optional).
  - @doc_name (varchar): The name of the document.
  - @doc (varbinary): The binary content of the document.
  - @doc_type (varchar): The type of the document (e.g., Resume, Certification).
- **Functionality**: Checks if a document with the same name has already been uploaded for the job seeker. If not, inserts the document into the documents table.
- **Result**: A message indicating whether the document upload was successful or if it already exists.

---

7. SubmitJobApplication

- **Description**: Submits a job application for a job seeker to a specific job.
- **Parameters**:
  - @JS_ID (int): The ID of the job seeker.
  - @job_ID (int): The ID of the job listing.
- **Functionality**: Checks if the job seeker has already applied for the job. If not, inserts a new record in the app table.
- **Result**: A job application is submitted if it doesn't already exist.

---

8. UpdateApplicantStatus

- **Description**: Updates the application status for a specific applicant.
- **Parameters**:
  - @app_ID (int): The ID of the application.
  - @new_status (nvarchar): The new status of the application (e.g., Approved, Rejected).
- **Functionality**: Updates the app_status of the specified application in the app table.
- **Result**: The status of the application is updated

Triggers

---

1. trig_afterJSModOnDoc

- **Description: Trigger that logs modifications to the documents table after an INSERT, UPDATE, or DELETE operation.**
- **Event: AFTER INSERT, UPDATE, DELETE on the documents table.**
- **Functionality:**
    - **Captures the doc_ID and JS_ID from the modified record(s).**
    - **Inserts an entry into the docAuditLog table, recording the doc_ID, JS_ID, action type ('Modify'), and the current timestamp using GETDATE().**
- **Purpose: To maintain a log of changes made to the documents table for audit purposes.**
- **Tables Involved:**
    - **documents: The table being monitored for changes.**
    - **docAuditLog: The table where change logs are recorded.**

---

2. trg_AfterJobApplicationInsert

- **Description: Trigger that updates the app table with the corresponding doc_ID after a job application is inserted.**
- **Event: AFTER INSERT on the app table.**
- **Functionality:**
    - **Updates the doc_ID in the app table by joining the newly inserted application record with the documents table on the JS_ID (Job Seeker ID).**
    - **The doc_ID from the documents table is assigned to the matching application record in the app table.**
- **Purpose: Ensures that the correct document is linked to the application by updating the doc_ID field after a new application is inserted.**
- **Tables Involved:**
    - **app: The table where job applications are stored.**
    - **documents: The table containing job seeker documents.**