ImpactCV Database Schema Documentation

Entity Relationship Diagram (ERD)

```
┌─────────────────────┐       ┌─────────────────────────────────┐
│       users         │       │             resumes             │
├─────────────────────┤       ├─────────────────────────────────┤
│ id (PK)             │       │ id (PK)                         │
│ email               │       │ user_id (FK)                    │
│ password_hash       │       │ title                           │
│ created_at          │       │ theme                           │
└─────────────────────┘       │ data (JSONB)                    │
         │                    │ created_at                      │
         │                    │ updated_at                      │
         │                    └─────────────────────────────────┘
         │                                 ▲
         └─────────────────────────────────┘

                     1:N
```

Tables Description

Users Table

The `users` table stores user account information.

Column	Type	Constraints	Description
id	SERIAL	PRIMARY KEY	Unique identifier for each user
email	VARCHAR	UNIQUE, NOT NULL	User's email address
password_hash	VARCHAR	NOT NULL	Bcrypt-hashed password
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Account creation timestamp

Resumes Table

The `resumes` table stores all resume data for users.

Column	Type	Constraints	Description
id	SERIAL	PRIMARY KEY	Unique identifier for each resume
user_id	INTEGER	FOREIGN KEY, ON DELETE CASCADE	Reference to users.id
title	VARCHAR	NOT NULL	Resume title
theme	VARCHAR		Selected theme identifier
data	JSONB	NOT NULL	Complete resume data structure
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Resume creation timestamp

updated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	Last update timestamp

Relationships

- **One-to-Many**: A user can have multiple resumes (1:N relationship)
  - The `user_id` in the `resumes` table is a foreign key referencing the `id` in the `users` table
  - When a user is deleted, all associated resumes are automatically deleted (CASCADE)

JSONB Data Structure

The `data` column in the `resumes` table uses PostgreSQL's JSONB type to store the complete resume structure. This allows for flexible schema evolution without database migrations.

Example structure:

```
{
  "basics": {
    "name": "John Doe",
    "label": "Software Engineer",
    "email": "john@example.com",
    "phone": "+1 (555) 123-4567",
    "picture": "/uploads/profile.jpg",
    "summary": "Experienced software engineer with 5+ years...",
    "location": {
      "city": "San Francisco",
      "region": "CA"
    },
    "profiles": [
      {
        "network": "LinkedIn",
        "url": "https://linkedin.com/in/johndoe"
      }
    ]
  },
  "work": [
    {
      "company": "Tech Company",
      "position": "Senior Developer",
      "startDate": "2020-01",
      "endDate": "Present",
      "summary": "Led development of...",
      "highlights": [
        "Increased performance by 40%",
        "Implemented CI/CD pipeline"
      ]
    }
  ],
  "education": [
    {
      "institution": "University of Technology",
      "area": "Computer Science",
      "studyType": "Bachelor",
      "startDate": "2012-09",
      "endDate": "2016-06",
      "gpa": "3.8"
    }
  ],
  "skills": [
    {
      "name": "Web Development",
      "level": "Advanced",
      "keywords": ["JavaScript", "React", "Node.js"]
    }
  ],
  "projects": [
    {
      "name": "Portfolio Website",
      "description": "Personal portfolio showcasing projects",
      "url": "https://example.com",
      "technologies": ["React", "Tailwind CSS"]
    }
  ]
}
```

Indexing Strategy

For optimal performance, the following indexes are recommended:

1. Index on `users.email` for fast login lookups
2. Index on `resumes.user_id` for quick filtering of resumes by user
3. Index on `resumes.updated_at` for sorting by last modified

Data Integrity

The database schema enforces the following integrity constraints:

1. User emails must be unique
2. Passwords must be stored as hashes, never in plain text
3. Every resume must be associated with a valid user
4. When a user is deleted, all their resumes are automatically deleted