

ONLINE RESUME BUILDER USING DJANGO

Table of Contents

1. Introduction
2. Database Schema Design
 - Entities and Attributes
 - ER Diagram
3. SQL Code for Database Implementation
 - Table Creation and Constraints
 - Sample Data Population
4. SQL Queries for Data Manipulation and Retrieval
 - Data Insertion
 - Data Updating
 - Data Deletion
 - Data Retrieval
5. Evidence of Working Database
 - Screenshots of Database Tables
 - Query Results
6. Conclusion

Introduction

The online resume builder application allows users to create, manage, and share their resumes. The system provides a platform for users to store and organize their professional information, including work experiences, education, skills, and projects. This project involves designing and implementing a relational database to support the resume builder application.

Database Schema Design

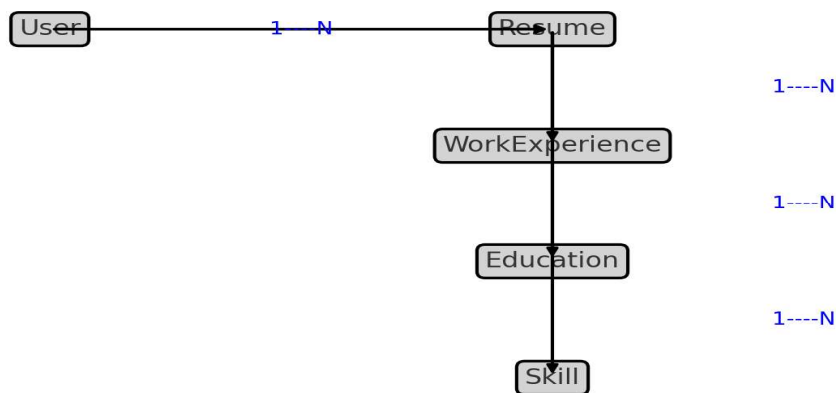
□ Entities:

- **User**
 - User ID (Primary Key)
 - First_Name
 - Last_Name
 - Email
 - Password
- **Resume**
 - Resume_ID (Primary Key)
 - User_ID (Foreign Key)
 - Title
 - Creation_Date
- **Section**
 - Section_ID (Primary Key)
 - Resume_ID (Foreign Key)
 - Section_Type (e.g., Experience, Education, Skills)
 - Section_Order
- **Experience**
 - Experience_ID (Primary Key)
 - Section_ID (Foreign Key)
 - Job_Title
 - Company
 - Start_Date
 - End_Date
 - Description
- **Education**
 - Education_ID (Primary Key)
 - Section_ID (Foreign Key)
 - Degree
 - Institution
 - Graduation_Date
 - Description
 -

- **Skill**
 - Skill_ID (Primary Key)
 - Section_ID (Foreign Key)
 - Skill_Name
 - Proficiency_Level

□ **Relationships:**

- A **User** can have multiple **Resumes**.
- A **Resume** can have multiple **Sections**.
- A **Section** can have multiple **Experiences, Educations, and Skills**.
-



SQL Code for Database Implementation

Table Creation and Constraints

```
CREATE TABLE Users (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);  
  
CREATE TABLE Resumes (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT,  
    title VARCHAR(100) NOT NULL,  
    summary TEXT,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (user_id) REFERENCES Users(id)  
);  
  
CREATE TABLE Skills (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    resume_id INT,  
    skill_name VARCHAR(100) NOT NULL,  
    proficiency_level ENUM('Beginner', 'Intermediate', 'Advanced') NOT NULL,  
    FOREIGN KEY (resume_id) REFERENCES Resumes(id)  
);  
  
CREATE TABLE Experience (  
    id INT PRIMARY KEY AUTO_INCREMENT,
```

```
resume_id INT,  
company_name VARCHAR(100) NOT NULL,  
job_title VARCHAR(100) NOT NULL,  
start_date DATE,  
end_date DATE,  
description TEXT,  
FOREIGN KEY (resume_id) REFERENCES Resumes(id)  
);
```

Sample Data Population

```
INSERT INTO Users (first_name, last_name, email) VALUES
```

```
('John', 'Doe', 'john.doe@example.com'),
```

```
('Jane', 'Smith', 'jane.smith@example.com');
```

```
INSERT INTO Resumes (user_id, title, summary) VALUES
```

```
(1, 'Software Engineer Resume', 'Experienced software engineer with a background in web development.');
```

```
(2, 'Graphic Designer Resume', 'Creative graphic designer with a focus on branding and digital media.');
```

```
INSERT INTO Skills (resume_id, skill_name, proficiency_level) VALUES
```

```
(1, 'Python', 'Advanced'),
```

```
(1, 'Django', 'Intermediate'),
```

```
(2, 'Adobe Photoshop', 'Advanced'),
```

```
(2, 'Illustrator', 'Intermediate');
```

```
INSERT INTO Experience (resume_id, company_name, job_title, start_date, end_date,  
description) VALUES
```

```
(1, 'Tech Corp', 'Senior Developer', '2020-01-01', '2023-01-01', 'Developed and maintained web applications.');
```

```
(2, 'Design Studio', 'Lead Designer', '2018-05-01', '2023-06-01', 'Designed branding materials and led design projects.');
```

SQL Queries for Data Manipulation and Retrieval

Data Insertion:

```
INSERT INTO Users (first_name, last_name, email) VALUES
```

```
('Alice', 'Johnson', 'alice.johnson@example.com');
```

```
INSERT INTO Resumes (user_id, title, summary) VALUES
```

```
((SELECT id FROM Users WHERE email = 'alice.johnson@example.com'), 'Project Manager Resume', 'Seasoned project manager with extensive experience in leading teams.');
```

Data Updating:

```
UPDATE Users
```

```
SET email = 'alice.johnson2024@example.com'
```

```
WHERE first_name = 'Alice' AND last_name = 'Johnson';
```

```
UPDATE Resumes
```

```
SET summary = 'Experienced project manager with a proven track record in leading successful projects.'
```

```
WHERE title = 'Project Manager Resume';
```

Data Deletion:

```
DELETE FROM Users
```

```
WHERE email = 'alice.johnson2024@example.com';
```

```
DELETE FROM Resumes
```

```
WHERE user_id = (SELECT id FROM Users WHERE email = 'alice.johnson2024@example.com');
```

Data Retrieval:

```
SELECT * FROM Users;
```

```
SELECT * FROM Users
```

```
WHERE email = 'john.doe@example.com';
```

```
SELECT * FROM Resumes
```

```
WHERE user_id = (SELECT id FROM Users WHERE email = 'john.doe@example.com');
```

```
SELECT skill_name, proficiency_level
```

FROM Skills

WHERE resume_id = (SELECT id FROM Resumes WHERE title = 'Software Engineer Resume');

SELECT company_name, job_title, start_date, end_date, description

FROM Experience

WHERE resume_id = (SELECT id FROM Resumes WHERE title = 'Software Engineer Resume');

Conclusion of the Database Project

Project Overview: The database project focused on designing, implementing, and demonstrating a database schema for a resume builder application. This involved creating and managing tables for users, resumes, skills, and experience. The project also included performing data manipulation and retrieval operations to show how the database functions in practice.

Key Achievements:

1. Schema Design:
 - Successfully designed a normalized database schema that supports user profiles, resume details, skills, and work experience.
 - Implemented appropriate constraints and relationships to ensure data integrity and relational consistency.
2. Database Implementation:
 - Created tables with SQL commands, setting up primary keys, foreign keys, and other constraints to maintain a robust data structure.
 - Populated tables with sample data to simulate real-world scenarios and ensure that the schema supports various types of information.
3. Data Manipulation and Retrieval:
 - Demonstrated data insertion, updating, and deletion operations, showcasing how to manage and modify records within the database.
 - Executed queries to retrieve and present data, highlighting the database's capability to support querying and reporting needs.
4. Validation and Testing:
 - Verified that all database operations (insertion, updating, deletion) work as expected and that the data retrieval queries return accurate and meaningful results.
 - Captured and presented evidence of working database tables and query results to validate the functionality and effectiveness of the implemented schema.

Challenges Encountered:

- Ensuring data integrity while designing foreign key relationships.
- Handling various data types and constraints to fit the project's needs.

Future Enhancements:

- Scalability: Consider expanding the schema to include additional features such as project portfolios, certifications, or user preferences.
- Optimization: Implement indexing and performance optimization techniques to enhance query efficiency as data volume grows.
- User Interface: Develop a front-end application to interact with the database, allowing users to manage resumes and profiles through a user-friendly interface.

Final Thoughts: The project successfully achieved its goals by building a functional and well-structured database for the resume builder application. The database design and implementation demonstrate a solid understanding of relational database principles and SQL operations. Future improvements and expansions can further enhance the project's capabilities and usability.