

# Applied Database Technology | Spring 2024

## Final Project Stage-2

### Group 2: DevPath Insight: A Stack Overflow Data-Driven Career Tool

#### Database Design

**Minju Kim**  
MS in Data Science  
[mk159@iu.edu](mailto:mk159@iu.edu)

**FNU Aditi**  
MS in Data Science  
[fnuadit@iu.edu](mailto:fnuadit@iu.edu)

**Jeet Rakesh Patel**  
MS in Data Science  
[patejeet@iu.edu](mailto:patejeet@iu.edu)

#### 1. Schema for database

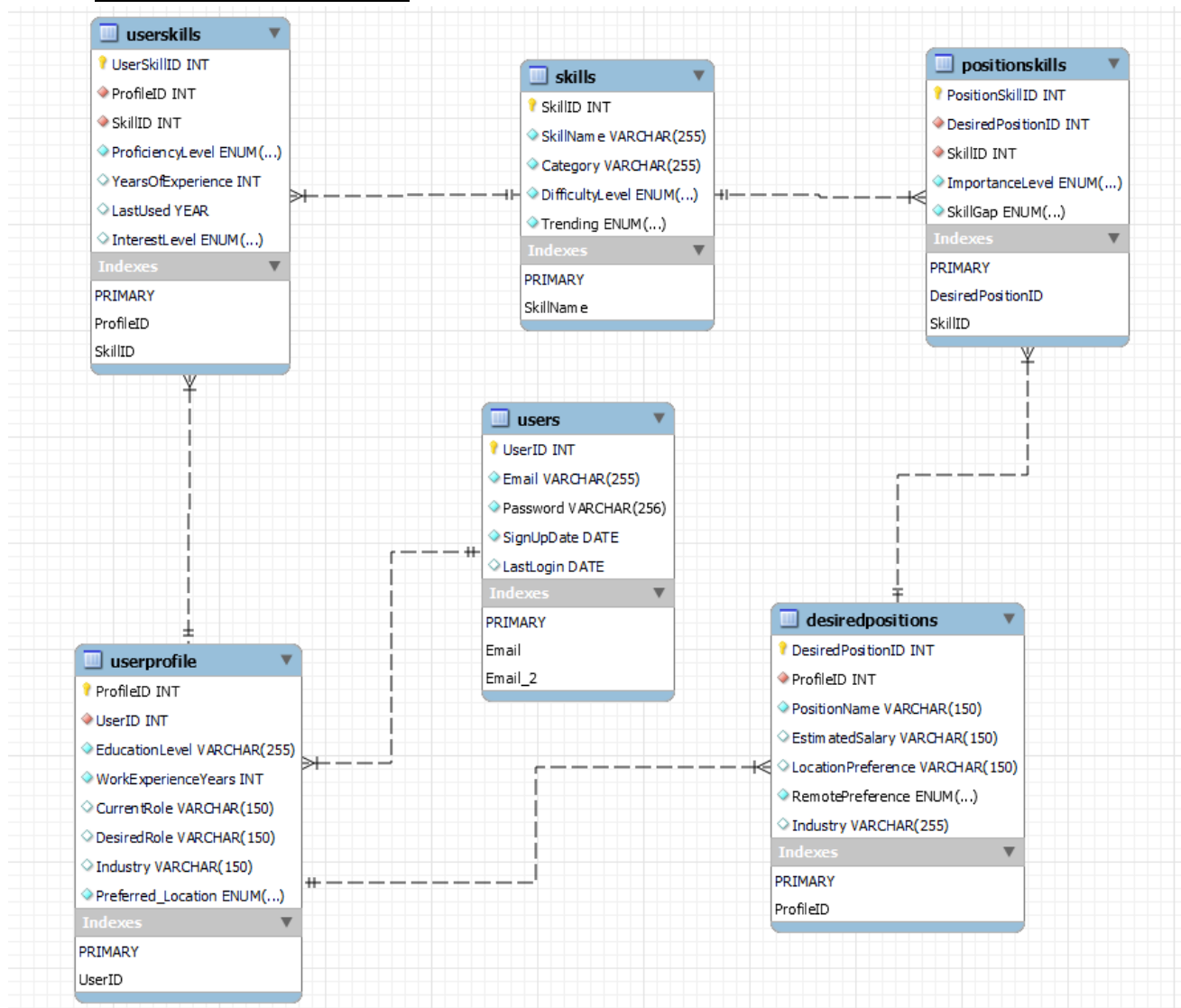


Figure 1: Schema Design

Our database began with a vast amount of data from the Stack Overflow Annual Developer Survey, featuring 84 columns and 90,000 rows. Recognizing the need for efficiency, we carefully analyzed the survey to select data useful for delivering personalized career insights. Through thoughtful discussions and prioritization, we decomposed the dataset to essential columns supporting our application's core functionalities. Our normalization process involved segmenting the data into well-structured tables **Users**, **UserProfile**, **Skills**, **UserSkills**, **DesiredPositions**, and **PositionSkills** eliminating redundancy and optimizing for query performance.

Here the database follows 3NF, because here in our database there are no transitive dependencies, meaning no non-key attribute depends on another non-key attribute.

### **Entities:**

- **Users:** Each record represents an individual user, uniquely identified and authenticated by their email.
- **UserProfile:** Contains detailed, non-repetitive information about the user, with a one-to-one linkage to Users.
- **Skills:** Acts as a centralized repository for skill data, serving multiple UserSkills records without duplication.
- **UserSkills:** Links users' profiles to their skills, highlighting the diversity of skill sets and experience levels within our user base.
- **DesiredPositions:** Reflects the career aspirations of our users, connecting their profiles to various potential job roles.
- **PositionSkills:** Bridges desired job roles with necessary skills, crucial for matching users to their preferred positions and guiding skill development

### **Relationships between the table**

- **Users to UserProfile:** One-to-One relationship, the foreign key in UserProfile referencing Users enforces referential integrity, ensuring that every profile is associated with an existing user.
- **UserProfile to UserSkills:** One-to-Many relationship, a profile can have many associated skills.
- **Skills to UserSkills:** One-to-Many relationship, a skill can be associated with many user profiles (though this is not directly shown in the diagram). Here we kept skills in a separate table while referencing them in UserSkills is an example of **normalization**, which reduces redundancy and improves data consistency.
- **UserProfile to DesiredPositions:** One-to-Many relationship, a profile can aspire to multiple desired positions.
- **DesiredPositions to PositionSkills:** One-to-Many relationship, a desired position can require multiple skills.
- **Skills to PositionSkills:** One-to-Many relationship, indicating that a skill can be required for multiple positions (not shown directly but implied).

## **2. Database Constraints:**

The entity relation diagram includes several types of constraints. Primary key constraints enforce uniqueness within the data. Foreign key constraints maintain referential integrity between related tables. Not null constraints ensure columns cannot have empty values. Unique constraints guarantee uniqueness of values or combinations across specified columns in the table.

### **USERS TABLE**

- UserID: Primary Key (ensuring each user in the Users table has a unique identifier) with an AUTO\_INCREMENT attribute.
- Email: A VARCHAR(255) column set to NOT NULL, each user must have a UNIQUE email address, which is required and cannot be shared with others.
- Password: Stored in a VARCHAR(256) field and marked as NOT NULL, which means that a password must be provided upon user creation for security purposes.
- SignUpDate: A DATETIME column set to NOT NULL, which records the exact date and time when the user account was created, and must be filled out for every user.
- LastLogin: Another DATETIME field, but this one allows for NULL values, accommodating the fact that new users might not have logged in immediately after account creation.
- **INDEX (Email)**: This index is created on the Email field to increase the efficiency of search queries that use the email address, which is a common identifier for login and account management functions.

### **USER PROFILE TABLE**

- ProfileID: Primary Key with an Auto\_Increment attribute.
- UserID: INT datatype that establishes a link to the corresponding user in the Users table. The UNIQUE and NOT NULL constraint ensures that each user has only one profile, **enforcing a one-to-one relationship between the Users table and the UserProfile table.**
- EducationLevel: A VARCHAR(255) field set to NOT NULL
- WorkExperienceYears: INT, NOT NULL
- CurrentRole: A VARCHAR(150) field for the user's current or most recent job title, which can be left empty if not applicable.
- DesiredRole: Another VARCHAR(150) field for the job title the user aspires to obtain next, which is optional.
- Industry: Stored in a VARCHAR(150) field, this column represents the current or desired industry of work for the user and is optional.

- Preferred Location: An ENUM type set to NOT NULL, restricting the user's preferred work location to 'Remote', 'Hybrid', or 'Offsite', thereby requiring a selection upon profile creation.
- **FOREIGN KEY (UserID)**: A foreign key that references the UserID in the Users table, enforcing referential integrity by ensuring that each UserProfile is tied to an existing and valid user.

## **SKILLS TABLE**

- SkillID: Primary Key, NOT NULL, UNIQUE
- SkillName: NOT NULL, ensuring that every skill entered into the database is given a name. The UNIQUE constraint on the SkillName field is crucial as it prevents duplicate entries.
- Category: Denotes the broad category or area that the skill is associated with, for example, "Programming" or "Data Science." This non-null field assists in the structured organization and efficient retrieval of skills.
- DifficultyLevel: An ENUM type that categorizes the skill's difficulty into predefined options: 'Beginner', 'Intermediate', or 'Advance'.
- Trending: ENUM to convey the current market demand status for a skill, with possible values being 'Not Trending', 'Stable', 'Rising', or 'Hot'.

## **USERSKILLS TABLE:**

- UserSkillID: An INT with AUTO\_INCREMENT serves as the Primary Key.
- ProfileID: An INT column, marked as NOT NULL and UNIQUE, linking each skill entry to a specific user profile in the `UserProfile` table.
- SkillID: INT column that connects each skill entry to a specific skill in the Skills table. Marked as NOT NULL and UNIQUE to ensure every user skill entry is linked to a predefined skill without duplication.
- ProficiencyLevel: An ENUM type that is set as NOT NULL and have the proficiency level to 'Beginner', 'Intermediate', 'Advanced', or 'Expert'.
- YearsOfExperience: An INT to represent the number of years a user has worked with the skill.
- LastUsed: A YEAR field to record the most recent year the user used the skill
- InterestLevel: An ENUM to indicate the user's interest in continuing to use or develop the skill, with options like 'Low', 'Medium', or 'High'.
- **FOREIGN KEY (ProfileID) REFERENCES UserProfile(ProfileID)**: This ensures that each entry in UserSkills is tied to an existing user profile.
- **FOREIGN KEY (SkillID) REFERENCES Skills(SkillID)**: This maintains the relationship between each user skill and the general skill definitions.

The decision to have two separate tables for skills, namely the Skills table and the UserSkills table, is to maintain clear separation of concerns and database efficiency.

- Having a dedicated Skills table makes it easier to scale the application as new skills can be added without affecting the user-specific data in the UserSkills table.

### **DESIRED POSITION TABLE:**

- DesiredPositionID: Primary Key with AUTO\_INCREMENT attribute.
- ProfileID: Links desired positions to specific user profiles, establishing a one-to-many relationship. Marked as NOT NULL with a FOREIGN KEY constraint referencing the 'UserProfile' table.
- PositionName: Identifies the job position. Marked as NOT NULL.
- EstimatedSalary: INT, stores the estimated salary range for the desired position.
- LocationPreference: Captures the geographic location preference for the job.
- RemotePreference: Enumerates the user's preference regarding remote work. Marked as NOT NULL.
- Industry: Specifies the industry or sector of the desired position.
- **Foreign Key Constraint**: Ensures referential integrity by linking the 'ProfileID' field to the ProfileID field in the UserProfile table. This constraint guarantees that each entry in the DesiredPositions table corresponds to an existing user profile.

### **POSITION TABLES:**

- PositionSkillID: Primary Key with AUTO\_INCREMENT attribute, uniquely identifying each entry in the table.
- DesiredPositionID: Links to the 'DesiredPositions' table, indicating the desired position to which the skills entry corresponds. Ensures a one-to-many relationship between desired positions and skills. Marked as NOT NULL with a FOREIGN KEY constraint referencing the 'DesiredPositions' table.
- SkillID: Links to the 'Skills' table, specifying the required skill for the desired position. Multiple entries can reference the same skill. Marked as NOT NULL with a FOREIGN KEY constraint referencing the 'Skills' table.
- ImportanceLevel: Indicates the significance of the skill for the desired position, categorized as 'Essential' or 'Desirable'. Marked as NOT NULL.
- SkillGap: Reflects the user's proficiency gap in the skill needed for the desired position. Utilizes an ENUM with options like 'None', 'Minor', 'Moderate', or 'Major'. Marked as NOT NULL with a DEFAULT value of 'None'.
- **Foreign Key Constraints**: For referential integrity by linking the DesiredPositionID field to the 'DesiredPositionID' field in the 'DesiredPositions' table and the 'SkillID' field to the 'SkillID' field in the Skills table. These constraints guarantee that each entry in the PositionSkills table corresponds to an existing desired position and skill, maintaining data consistency.

### **3. Write code to create a database and build queries. Your task is to create a reproducible code.**

The SQL code for database creation, table creation, constraint enforcement, and value insertion has been documented and is available in the attached files on Github.

Github repository: <https://github.iu.edu/patejeet/ADT-Project-DevPath-Insight>

### **4. Group Contribution**

<b>Name</b>	<b>Tasks</b>	<b>Contribution</b>	<b>AVR Time Spent (hrs)</b>
Minju Kim	<ul style="list-style-type: none"><li>- Conceptual Schema</li><li>- MySQL Code</li><li>- Word Document</li></ul>	<ul style="list-style-type: none"><li>- Provided ideas and discussed how to make schema</li><li>- Made MySQL code for creating tables and inserting data into "Users" and "Users Profile" tables.</li><li>- Wrote "Project Plan Part2" Report.</li></ul>	10
Jeet Rakesh Patel	<ul style="list-style-type: none"><li>- Conceptual Schema</li><li>- MySQL Code and Database Constraints</li><li>- Word Document</li></ul>	<ul style="list-style-type: none"><li>- Participated in the design of the conceptual database schema by defining the entities.</li><li>- Contributed to establishing database constraints, including primary key, foreign key, not null, and unique constraints, for each entity.</li><li>- Made MySQL code for creating tables and inserting data into "Skills" and "UserSkills" tables.</li><li>- Specifically responsible for Skills and UserSkills Table and writing all the SQL queries related to this table.</li><li>- Wrote "Project Plan Part2" Report.</li></ul>	10
FNU Aditi	<ul style="list-style-type: none"><li>- Github repository</li><li>- MySQL Code</li><li>- Word document</li></ul>	<ul style="list-style-type: none"><li>- Updated the github repository with the updated table queries and schema diagram</li><li>- Made MySQL code for the Desiredpositions and SkillPosition table</li><li>- Wrote "Project Plan Part2" Report</li></ul>	10