HELPIFY
Database Project Documentation

By

Isaiah Anthony P. Briones

ARJONEL M. MENDOZA, MIT
Lecturer

PROJECT OVERVIEW

Helpify is a very interactive and user-friendly system, Helpify is a straightforward application that lets you convert files like PDF to doxc, gif to png, and PDF to txt, and many other formats.

The system in total provides nine converters, each of which is linked to the database for tracking, log history with real-time tracking, a user statistics counter that keeps track of who has logged in the most and the total number of converted files, sign-in and sign-out functions, and an update account feature. This project, which was constructed using Python, Tkinter, and SQlite3, produced a small, locally compatible converter with database management that makes it easier to access such converters offline and with results comparable to those of online converters.

ENTITY-RELATIONSHIP DIAGRAM (ERD)

The ERD outlines the core entities in the Quantum Dice Roller System, including Users, Dice Rolls, and Roll Statistics. Each entity is defined by attributes essential for tracking user activity, managing dice roll results, and storing roll statistics for individual users.

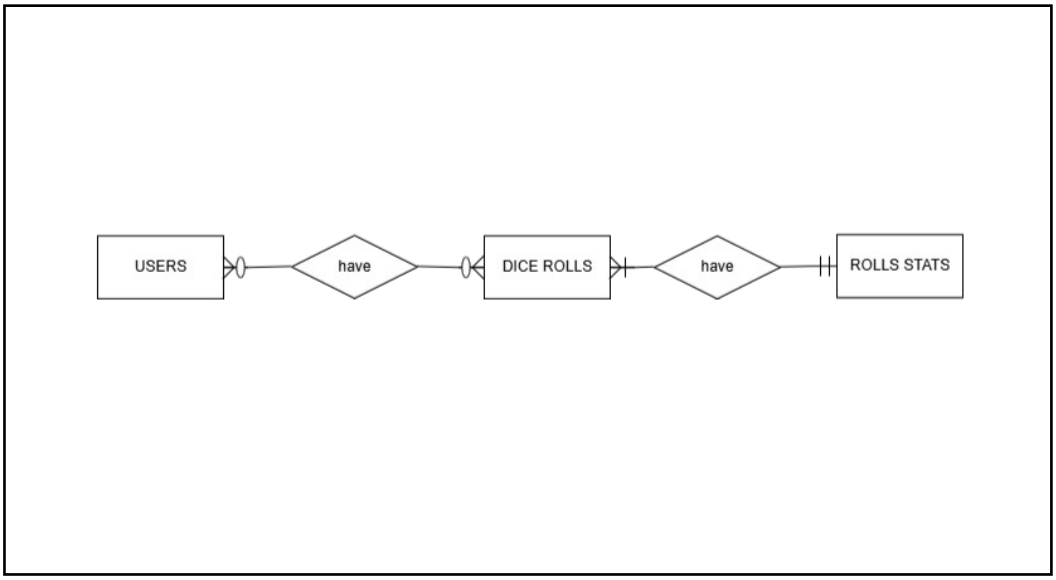

Figure 1. Entity Relationship Diagram

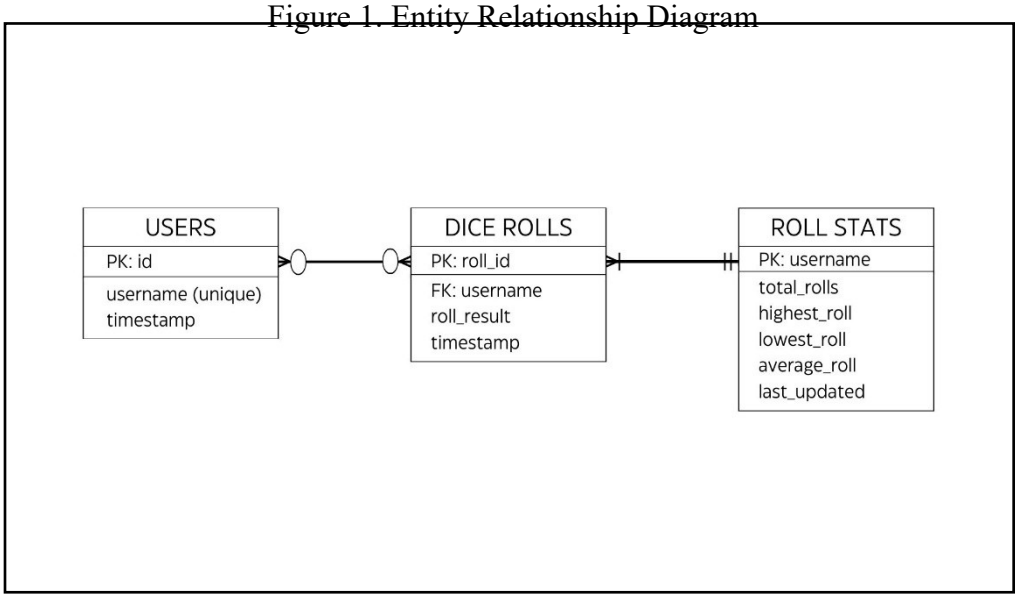

Figure 2. Entity Relationship Diagram

Entities and their Relationships

1. Users and Dice Rolls

- **Type:** One-to-Many (1:N)
- **Scientific Basis:** Each dice roll is an independent event associated with a unique user. This aligns with the principle of independent random events in probability theory, where prior rolls do not influence future outcomes.

2. Users and Roll Statistics

- **Type:** One-to-One (1:1)
- **Scientific Basis:** Statistical summaries provide a snapshot of a user's historical dice rolls, reflecting broader concepts in data aggregation and analysis, such as mean and extrema calculations.

SQL SCRIPTS

```
CREATE TABLE IF NOT EXISTS users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT UNIQUE NOT NULL,
    password TEXT NOT NULL
);
```

```
CREATE TABLE IF NOT EXISTS dice_rolls (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    username TEXT NOT NULL,
    roll_result INTEGER NOT NULL,
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (username) REFERENCES users (username)
);

CREATE TABLE IF NOT EXISTS roll_statistics (
    username TEXT PRIMARY KEY,
    total_rolls INTEGER DEFAULT 0,
    highest_roll INTEGER DEFAULT NULL,
    lowest_roll INTEGER DEFAULT NULL,
    average_roll REAL DEFAULT NULL,
    last_updated DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (username) REFERENCES users (username) ON DELETE CASCADE
);
```

Here are the QUERIES used in the accessing of the database:

Insert New User:

- INSERT INTO users (username, password) VALUES (?, ?)

Check if Username Exists:

- SELECT * FROM users WHERE username = ?

Insert New Dice Roll:

- INSERT INTO dice_rolls (username, roll_result) VALUES (?, ?)

Select All Rolls for User:

SELECT roll_result FROM dice_rolls WHERE username = ?

Insert or Update Roll Statistics:

- INSERT INTO roll_statistics (username, total_rolls, highest_roll, lowest_roll, average_roll, last_updated)

  VALUES (?, ?, ?, ?, ?, ?)

  ON CONFLICT(username) DO UPDATE SET

  total_rolls = ?, highest_roll = ?, lowest_roll = ?, average_roll = ?, last_updated = ?

Select Roll Statistics for User:

- SELECT total_rolls, highest_roll, lowest_roll, average_roll, last_updated

  FROM roll_statistics WHERE username = ?

Select Roll History (All Rolls with Timestamps):

- SELECT roll_result, timestamp FROM dice_rolls WHERE username = ? ORDER BY timestamp DESC

SAMPLE DATA