# Golf Stats Tracker Database Schema

### James Donohue

**1. Introduction**

This document details the schema design for the Golf Stats Tracker database, including all tables, relationships, and sample queries. The purpose of this database is to manage users, golf courses, and rounds played by users, storing detailed statistics for each round.

**2. Database Schema**

**2.1. Tables**

**2.1.1. Users Table**

* **Table Name:** Users
* **Description:** Stores user information.
* **Columns:**
  + user\_id (INT, AUTO\_INCREMENT, PRIMARY KEY)
  + username (VARCHAR(50), NOT NULL)
  + email (VARCHAR(100), NOT NULL)
  + password (VARCHAR(255), NOT NULL)
  + created\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

**2.1.2. Courses Table**

* **Table Name:** Courses
* **Description:** Stores golf course information.
* **Columns:**
  + course\_id (INT, AUTO\_INCREMENT, PRIMARY KEY)
  + course\_name (VARCHAR(100), NOT NULL)
  + tee\_color (VARCHAR(50), NOT NULL)
  + hole1\_yardage (INT, NOT NULL)
  + hole2\_yardage (INT, NOT NULL)
  + hole3\_yardage (INT, NOT NULL)
  + hole4\_yardage (INT, NOT NULL)
  + hole5\_yardage (INT, NOT NULL)
  + hole6\_yardage (INT, NOT NULL)
  + hole7\_yardage (INT, NOT NULL)
  + hole8\_yardage (INT, NOT NULL)
  + hole9\_yardage (INT, NOT NULL)
  + hole10\_yardage (INT, NOT NULL)
  + hole11\_yardage (INT, NOT NULL)
  + hole12\_yardage (INT, NOT NULL)
  + hole13\_yardage (INT, NOT NULL)
  + hole14\_yardage (INT, NOT NULL)
  + hole15\_yardage (INT, NOT NULL)
  + hole16\_yardage (INT, NOT NULL)
  + hole17\_yardage (INT, NOT NULL)
  + hole18\_yardage (INT, NOT NULL)
  + total\_yardage (INT, NOT NULL)
  + hole1\_par (INT, NOT NULL)
  + hole2\_par (INT, NOT NULL)
  + hole3\_par (INT, NOT NULL)
  + hole4\_par (INT, NOT NULL)
  + hole5\_par (INT, NOT NULL)
  + hole6\_par (INT, NOT NULL)
  + hole7\_par (INT, NOT NULL)
  + hole8\_par (INT, NOT NULL)
  + hole9\_par (INT, NOT NULL)
  + front9\_par (INT, NOT NULL)
  + hole10\_par (INT, NOT NULL)
  + hole11\_par (INT, NOT NULL)
  + hole12\_par (INT, NOT NULL)
  + hole13\_par (INT, NOT NULL)
  + hole14\_par (INT, NOT NULL)
  + hole15\_par (INT, NOT NULL)
  + hole16\_par (INT, NOT NULL)
  + hole17\_par (INT, NOT NULL)
  + hole18\_par (INT, NOT NULL)
  + back9\_par (INT, NOT NULL)
  + total\_par (INT, NOT NULL)
  + created\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

**2.1.3. Rounds Table**

* **Table Name:** Rounds
* **Description:** Stores round information for each user.
* **Columns:**
  + round\_id (INT, AUTO\_INCREMENT, PRIMARY KEY)
  + user\_id (INT, NOT NULL, FOREIGN KEY REFERENCES Users(user\_id))
  + course\_id (INT, NOT NULL, FOREIGN KEY REFERENCES Courses(course\_id))
  + date (DATE, NOT NULL)
  + total\_score (INT, NOT NULL)
  + total\_fairways\_hit (INT, NOT NULL)
  + total\_gir (INT, NOT NULL)
  + total\_putts (INT, NOT NULL)
  + created\_at (TIMESTAMP, DEFAULT CURRENT\_TIMESTAMP)

**2.1.4. HoleStats Table**

* **Table Name:** HoleStats
* **Description:** Stores detailed statistics for each hole in a round.
* **Columns:**
  + holestats\_id (INT, AUTO\_INCREMENT, PRIMARY KEY)
  + round\_id (INT, NOT NULL, FOREIGN KEY REFERENCES Rounds(round\_id))
  + hole\_number (INT, NOT NULL)
  + score (INT, NOT NULL)
  + fairways\_hit (INT, NOT NULL)
  + gir (INT, NOT NULL)
  + putts (INT, NOT NULL)

**2.2. Relationships**

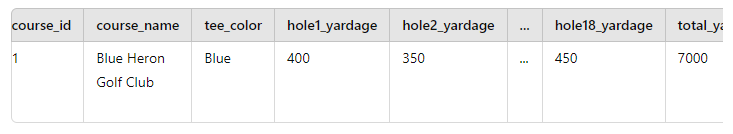
* Each User can have multiple Rounds.
* Each Round is associated with one Course.
* Each Round can have multiple HoleStats.

**3. Sample Data**

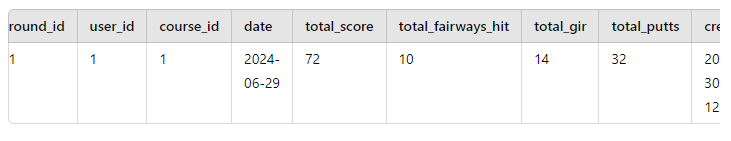
**3.1. Users Table**



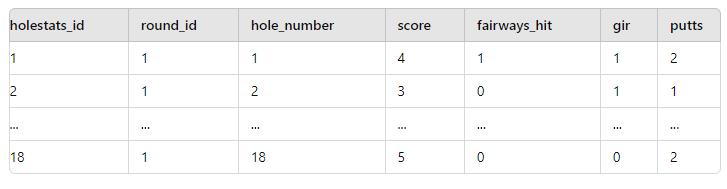
**3.2. Courses Table**



**3.3. Rounds Table**



**3.4. HoleStats Table**



**4. Queries**

**4.1. Insert a New User**

INSERT INTO Users (username, email, password) VALUES ('john\_doe', '[john@example.com](mailto:john@example.com)', 'hashed\_password');

**4.2. Insert a New Course**

INSERT INTO Courses (course\_name, tee\_color, hole1\_yardage, hole2\_yardage, ..., hole18\_yardage, total\_yardage, hole1\_par, ..., hole18\_par, front9\_par, back9\_par, total\_par) VALUES ('Blue Heron Golf Club', 'Blue', 400, 350, ..., 450, 7000, 4, ..., 5, 36, 36, 72);

**4.3. Insert a New Round**

INSERT INTO Rounds (user\_id, course\_id, date, total\_score, total\_fairways\_hit, total\_gir, total\_putts) VALUES (1, 1, '2024-06-29', 72, 10, 14, 32);

**4.4. Insert New HoleStats**

INSERT INTO HoleStats (round\_id, hole\_number, score, fairways\_hit, gir, putts) VALUES (1, 1, 4, 1, 1, 2);  
INSERT INTO HoleStats (round\_id, hole\_number, score, fairways\_hit, gir, putts) VALUES (1, 2, 3, 0, 1, 1);  
INSERT INTO HoleStats (round\_id, hole\_number, score, fairways\_hit, gir, putts) VALUES (1, 18, 5, 0, 0, 2);

**4.5. Fetch All Rounds for a User**

SELECT \* FROM Rounds WHERE user\_id = 1;

**4.6. Fetch Course Details**

SELECT \* FROM Courses WHERE course\_id = 1;

**4.7. Fetch HoleStats for a Round**

SELECT \* FROM HoleStats WHERE round\_id = 1;

**5. Testing**

To ensure the database is functioning as expected, perform the following tests:

* **Insert Users:** Use the insert query to add sample users and verify they appear in the Users table.
* **Insert Courses:** Use the insert query to add sample courses and verify they appear in the Courses table.
* **Insert Rounds:** Use the insert query to add sample rounds and verify they appear in the Rounds table.
* **Insert HoleStats:** Use the insert query to add sample hole stats and verify they appear in the HoleStats table.
* **Fetch Data:** Execute select queries to fetch data and ensure the results are as expected.