Fitness Tracker Database Schema - Complete Guide

Overview

This database is designed to track everything about a user's fitness journey, from planning workouts to recording actual performance. Think of it as the backend for a comprehensive fitness app like MyFitnessPal or Strong.

🏗️ Database Structure Overview

The system is built around **7 core tables** that work together to create a complete fitness tracking ecosystem:

1. **USERS** - Who is using the system
2. **MUSCLE_GROUPS** - What body parts we're targeting
3. **EXERCISES** - What movements we can do
4. **WORKOUT_PLANS** - Structured workout programs
5. **PLAN_EXERCISES** - Which exercises go in which plans
6. **WORKOUT_SESSIONS** - Individual workout instances
7. **SETS** - Individual sets performed during workouts

📋 Detailed Table Breakdown

1. USERS Table

Purpose: Stores information about each person using the fitness tracker.

```sql
USERS {
  user_id (Primary Key)    - Unique identifier for each user
  username                 - Display name (e.g., "FitJohn2024")
  email                    - Login credential and contact
  password_hash            - Encrypted password for security
  created_at               - When they joined
  fitness_level            - Beginner/Intermediate/Advanced
  goals                    - Their fitness objectives (lose weight, gain muscle, etc.)
}
```

Real-world example: John Smith signs up, gets user_id=1, wants to "build muscle and lose 20 lbs"

---

2. MUSCLE_GROUPS Table

Purpose: Categorizes exercises by what body parts they work.

```sql
MUSCLE_GROUPS {
    muscle_group_id (PK)     - Unique ID for each muscle group
    name                     - "Chest", "Biceps", "Quadriceps", etc.
    description              - Detailed explanation of the muscle
    body_region              - "Upper Body", "Lower Body", "Core"
}
```

Real-world example:

- muscle_group_id=1, name="Chest", body_region="Upper Body"
- muscle_group_id=2, name="Quadriceps", body_region="Lower Body"

---

3. EXERCISES Table

Purpose: Stores all available exercises, both system-provided and user-created.

```sql
EXERCISES {
    exercise_id (PK)         - Unique ID for each exercise
    user_id (FK)             - NULL for system exercises, user_id for custom ones
    muscle_group_id (FK)     - Which muscle group this targets
    exercise_name            - "Bench Press", "Squats", "Custom Bicep Curl"
    description              - Brief overview of the exercise
    instructions             - Step-by-step how-to perform it
    equipment_needed         - "Barbell", "Dumbbells", "Bodyweight"
    difficulty_level         - Beginner/Intermediate/Advanced
    is_custom                - TRUE if user-created, FALSE if system exercise
}
```

Real-world example:

- System exercise: "Bench Press" targets Chest, needs Barbell
- Custom exercise: John creates "Incline Dumbbell Press variation" for his home gym

---

4. WORKOUT_PLANS Table

Purpose: Structured workout programs that users follow.

```sql
WORKOUT_PLANS {
    plan_id (PK)          - Unique ID for each workout plan
    user_id (FK)          - Who created/owns this plan
    plan_name             - "Push/Pull/Legs", "John's Home Workout"
    description           - What the plan is about
    difficulty_level      - Beginner/Intermediate/Advanced
    duration_weeks        - How long the program lasts
    created_at            - When it was made
    is_active             - Whether user is currently following it
}
```

Real-world example: John creates "Beginner Mass Builder" - 12 weeks, focuses on compound movements

---

5. PLAN_EXERCISES Table (Junction Table)

Purpose: Links exercises to workout plans and defines how they should be performed.

```sql
PLAN_EXERCISES {
    plan_exercise_id (PK)   - Unique ID for this plan-exercise combination
    plan_id (FK)            - Which workout plan this belongs to
    exercise_id (FK)        - Which exercise to perform
    sets_planned            - How many sets to do (e.g., 3)
    reps_planned            - How many reps per set (e.g., 8-12)
    weight_planned          - Target weight to use
    rest_seconds            - How long to rest between sets
    order_in_workout        - Exercise #1, #2, #3 in the workout
    notes                   - Special instructions ("Focus on form")
}
```

Real-world example: In John's plan, Bench Press is exercise #1, do 3 sets of 8 reps at 135 lbs, rest 90 seconds

---

6. WORKOUT_SESSIONS Table

Purpose: Records individual workout instances - each time someone works out.

```sql
```

```
WORKOUT_SESSIONS {
  session_id (PK)       - Unique ID for each workout session
  user_id (FK)          - Who did this workout
  plan_id (FK)          - Which plan they followed (optional - can be freestyle)
  workout_date          - When they worked out
  start_time            - What time they started
  end_time              - What time they finished
  notes                 - How they felt, what went well/poorly
  session_type          - "Planned", "Freestyle", "Cardio", etc.
  completed             - Did they finish the whole workout?
}
```

Real-world example: John works out on Monday 3/15/2024 from 6:00-7:30 PM, following his "Beginner Mass Builder" plan

---

7. SETS Table

Purpose: Records every single set performed during workouts - the most granular data.

sql

```
SETS {
  set_id (PK)           - Unique ID for each set
  session_id (FK)       - Which workout session this set belongs to
  exercise_id (FK)      - Which exercise was performed
  set_number            - Set #1, #2, #3, etc. for this exercise
  reps_completed        - How many reps they actually did
  weight_used           - How much weight they actually used
  rest_seconds          - How long they actually rested
  difficulty_rating     - "Easy", "Perfect", "Too Hard"
  notes                 - "Felt strong today", "Form broke down"
  completed_at          - Exact timestamp when set was finished
}
```

Real-world example: John's first set of bench press: 8 reps at 135 lbs, felt "Perfect", rested 85 seconds

---

🔗 How The Tables Connect (Relationships)

One-to-Many Relationships:

1. **USERS → WORKOUT_PLANS**: One user can create many workout plans
2. **USERS → EXERCISES**: One user can create many custom exercises

3. **USERS → WORKOUT_SESSIONS**: One user performs many workout sessions
4. **MUSCLE_GROUPS → EXERCISES**: One muscle group contains many exercises
5. **WORKOUT_PLANS → PLAN_EXERCISES**: One plan contains many exercises
6. **EXERCISES → PLAN_EXERCISES**: One exercise can be in many plans
7. **WORKOUT_PLANS → WORKOUT_SESSIONS**: One plan can be used for many sessions
8. **WORKOUT_SESSIONS → SETS**: One session contains many sets
9. **EXERCISES → SETS**: One exercise can be performed in many sets

---

🎯 Real-World Usage Scenarios

Scenario 1: Creating a New Workout Plan

1. John (user_id=1) creates "Upper Body Strength" plan
2. He adds Bench Press, Rows, and Shoulder Press to the plan
3. Each exercise gets specific sets/reps/weight targets in PLAN_EXERCISES
4. Plan is ready to use!

Scenario 2: Doing a Workout

1. John starts a new workout session following his "Upper Body Strength" plan
2. He performs Bench Press: 3 sets recorded in SETS table
3. He performs Rows: 3 sets recorded in SETS table
4. Session is marked complete

Scenario 3: Tracking Progress

1. Query SETS table to see John's bench press weight over time
2. Compare planned vs. actual reps/weight from PLAN_EXERCISES vs. SETS
3. Generate progress charts and statistics

Scenario 4: Custom Exercise Creation

1. John's gym doesn't have regular barbells, only dumbbells
2. He creates custom exercise "Dumbbell Bench Press Variation"
3. Links it to "Chest" muscle group
4. Adds it to his workout plans
