Basics of database systems

Project – Workout Journal Database

Lappeenranta-Lahti University of Technology LUT
Software Engineering

Basics of database systems
Spring 2025

Suvi Vehmaanperä

TABLE OF CONTENTS

1 DEFINITION

The "Workout Journal Database" project is designed to help users efficiently record, track, and manage their workout progress. This database is intended for athletes and fitness enthusiasts who want to systematically log their training plans, associated workouts, exercises, and results. The goal is to provide a structured way to store workout data and analyze training history, allowing users to monitor their strength progression over time.

The database stores essential information, including user profiles, training plans, workouts within those plans, exercises linked to each workout, and performance results such as weights lifted, repetitions, and sets completed. Additionally, users can add personal notes related to their workouts, allowing them to document their experiences and insights.

This database serves as a structured digital solution for logging workouts, tracking past performance, and analyzing progress. It enables users to plan and follow structured training programs. Log workouts and exercises associated with their training plans. Record detailed performance metrics, including weights, repetitions, and sets. Track progress over time and compare results with previous performance. Attach notes to workouts and exercises for personal reflection.

The following five queries will be implemented in the database to allow users to efficiently retrieve essential workout data: 1) List all training plans and their associated workouts for a specific user. 2) List all exercises included in a specific workout for a specific user. 3) List all comments added to a specific workout. 4) List all recorded results for a specific exercise by a specific user, ordered chronologically. 5) List the best recorded performance for each exercise for a specific user.

2 MODELING

2.1 Data model

Figure 1 shows the Workout Journal database model. The model consists of six primary entities: Users, Workout Plans, Exercises, Workouts, Results, and Notes. In addition, an intermediate relationship (Exercises_in_workout) is used to convert the many-to-many (N:M) relationship between exercises and workouts into two one-to-many (1:N) relationships.

The other relationships in the database follow a one-to-many (1:N) structure, while Exercises_in_workout allows multiple exercises to be linked to multiple workouts.

All tables except Users and Exercises contain foreign keys.

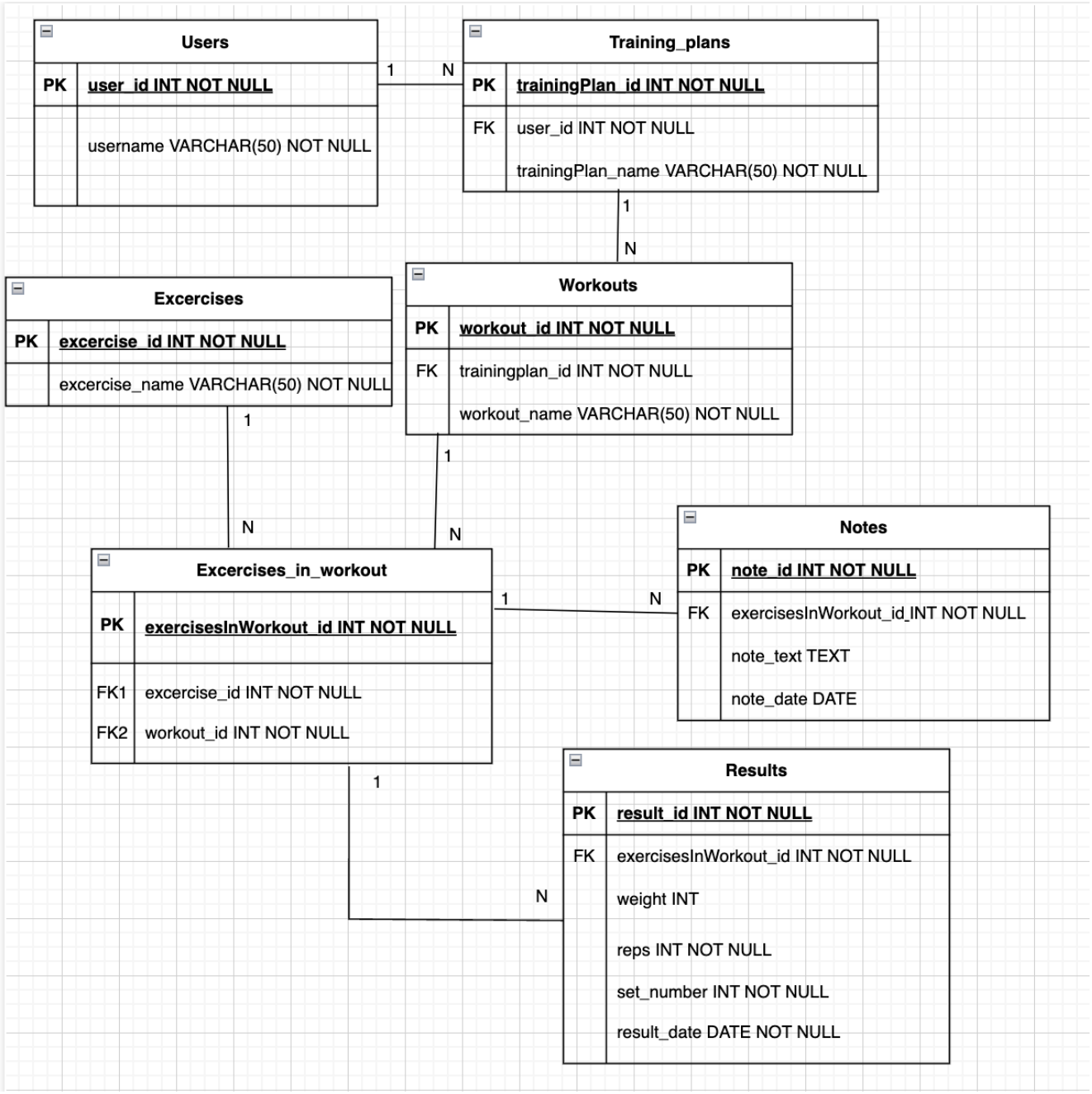

Figure 1: Data model

3 DATABASE IMPLEMENTATION

During implementation, the following constraints are created for the relations:

- **Users**:
  - user_id is primary key and cannot be null (NOT NULL)
  - username cannot be null (NOT NULL) and must be unique (UNIQUE)

- **Training_plans**:
  - trainingPlan_id is primary key and cannot be null (NOT NULL)
  - trainingPlan_name cannot be null (NOT NULL)
  - user_id is a foreign key reference to Users, cannot be null (NOT NULL)
    - ON UPDATE RESTRICT
    - ON DELETE CASCADE

- **Workouts**:
  - workout_id is primary key and cannot be null (NOT NULL)
  - workout_name cannot be null (NOT NULL)
  - trainingPlan_id is a foreign key reference to Training_plans, cannot be null (NOT NULL)
    - ON UPDATE RESTRICT
    - ON DELETE CASCADE

- **Exercises**:
  - exercise_id is primary key and cannot be null (NOT NULL)
  - exercise_name cannot be null (NOT NULL)

- **Exercises_in_workout**:
  - exercisesInWorkout_id is primary key and cannot be null (NOT NULL)
  - exercise_id is a foreign key reference to Exercises, cannot be null (NOT NULL)
    - ON UPDATE RESTRICT

        - ON DELETE CASCADE
    - workout_id is a foreign key reference to Workouts, cannot be null (NOT NULL)
        - ON UPDATE RESTRICT
        - ON DELETE CASCADE

- **Notes**:
    - note_id is primary key and cannot be null (NOT NULL)
    - note_date cannot be null (NOT NULL) and DEFAULT value is CURRENT_DATE
    - exercisesInWorkout_id is a foreign key reference to Exercises_in_workouts, cannot be null (NOT NULL)
        - ON DELETE CASCADE
        - ON UPDATE RESTRICT

- **Results**:
    - result_id is primary key and cannot be null (NOT NULL)
    - weight is DEFAULT value of 0
    - reps cannot be null (NOT NULL) and must be at least 1 CHECK (reps >= 1)
    - set_number cannot be null (NOT NULL) and must be at least 1 CHECK (sets >= 1)
    - result_date cannot be null (NOT NULL) and DEFAULT value is CURRENT_DATE
    - exercisesInWorkout_id is a foreign key reference to Exercises_in_workouts cannot be null (NOT NULL)
        - ON UPDATE RESTRICT
        - ON DELETE CASCADE

In addition to the integrity constraints listed above, the database also includes two indices to improve query performance.

Index (index_results_exercisesInWorkout) is created on the exercisesInWorkout_id column in the Results table to speed up searches when retrieving performance data for specific exercises within workouts.

The index (index_exercisesInWorkout) is created on the Exercises_in_workout table for the combination of workout_id and exercise_id, allowing for faster lookups when determining which exercises belong to specific workouts

These indices enhance database efficiency by optimizing workout results tracking and improving search queries for exercises included in workouts.