Basics of database systems

# Project – Database design

Lappeenranta-Lahti University of Technology LUT
Software Engineering

Basics of database systems
Spring 2023

Jere Arokivi
Database implementation for fitness data

**TABLE OF CONTENTS**

# 1 DEFINITION

## Fitness Database

In the project ‘Fitness database’ is developed a comprehensive database for individuals looking to track their fitness and nutrition progress, and for fitness trainers to manage their client’s progress. It provides a comprehensive approach to tracking various aspects of fitness, including workout plans, exercises, nutrition plans, meals, and progress tracking. The triggers added to the database also ensure data integrity, such as preventing duplicate email addresses and creating default nutrition and workout plans for new users.

The database consists of six main tables, including users, workouts, exercises, nutrition plans, meal plans, and progress tracking, each containing specific information relevant to fitness and nutrition tracking.

The users table contains basic information about users, such as their name, email, date of birth, height, weight, and goal weight. The workouts table contains details of different workout plans created by users, and the exercises table contains information on exercises performed in each workout, including exercise name, muscle group targeted, and equipment required.

The nutrition plans table contains data related to users' dietary requirements and targets, while the meal plans table contains details of meals planned according to the nutrition plan. The progress tracking table contains information on users' progress, including weight and body fat percentage.

**Queries implemented.**

1. Retrieves the names and email addresses of all users, which can be useful for checking users' contact information. It can also be used for creating mailing lists or other communication purposes.
2. Filters the results of Query 1 to only show users with a Gmail email address. This can be useful for targeted advertisement campaigns to Gmail users, for example.
3. Lists each user's name and all their associated workouts in a single column-spaced list. This can be useful for analysing users' workout habits or for providing personalized workout recommendations.
4. Retrieves weight and body fat percentage data for each user in a single column-spaced list. This can be useful for tracking users' progress over time or for further analysis of weight-related data.
5. Lists all meal plans for each user based on their associated nutrition plans. This can be useful for recommending additional meal plans to users or for analysis of users' dietary habits.

# 2 MODELING

## 2.1 Concept model

**About the ER Model**

The relationships between these tables are defined using foreign keys. The workouts, exercises, and progress tracking tables are related to the users table through the user identification field. The meal plans and nutrition plans tables are related through the plan identification field, and the meal plans, and nutrition plans tables are related to the users table through the user identification field.

The ER model for this database consists of six entities: users, workouts, exercises, nutrition plans, meal plans and progress tracking. The relationships between these entities are as follows: one user can have many workouts, one workout can have many exercises and exercises can be on multiple workouts, one user can have many nutrition plans, one

nutrition plan can have many meal plans and they can be included in different nutrition plans.

Overall, this database is useful for individuals who are looking to manage their fitness goals. It allows users to track their progress, create and modify workout plans, and manage their nutritional intake. With the use of triggers, the database can automatically create default workout and nutrition plans when a user is added to the system. The database can be further expanded by adding more fields and tables to accommodate additional features or requirements. ER model is presented in the figure 1.

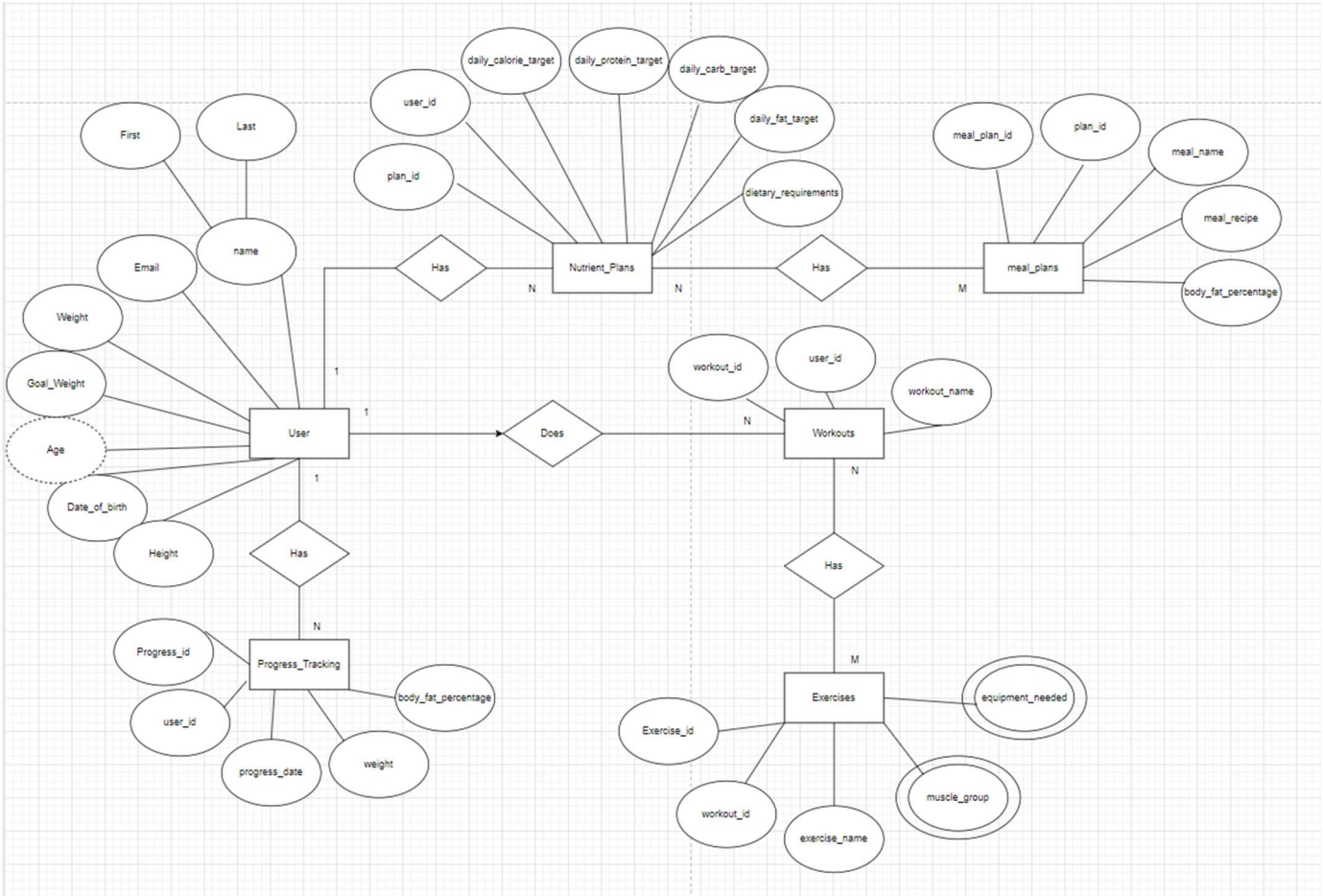


**Figure 1:** ER model

## 2.2 Relational model

In order to transform the ER model of the presented database into a relation model, the entities and their relationships were identified and transformed into tables.

The resulting relation model consists of six tables: users, workouts, exercises, nutrition plans, meal plans, and progress tracking. These tables represent the main entities of the database and their relationships, such as users having multiple workouts, workouts having multiple exercises, users having nutrition plans, and users having progress tracking.

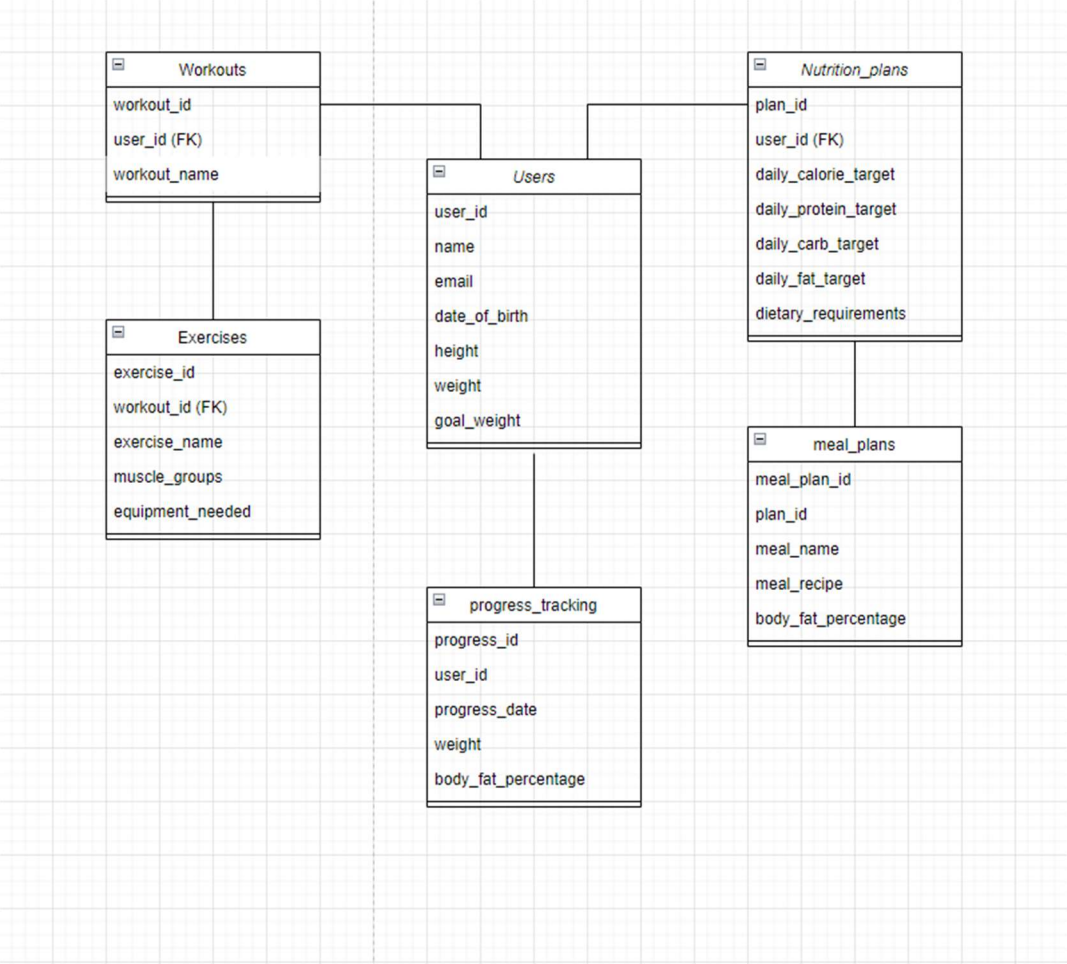


**Figure 2:** Relational model from the ER model

# 3 DATABASE IMPLEMENTATION

**During the implementation process, the first step was to determine the necessary components for the database. In this case, we needed to identify the required tables. Once the tables were defined, we identified the connections and properties of each table to determine the attributes and their properties. As a result of this process, the following database was created, with all identifiers implemented using autoincrement functionality.**

- **Users**:
  - Name cannot be null (NOT NULL)
  - Height, Weight, Goal Weight need to be positive numbers (CHECK >0)
  - Email can only have unique value (UNIQUE)
- **Workouts**:
  - Foreign key reference to User.
  - Workout Name cannot be null (NOT NULL)
  - ON DELETE CASCADE
- **Exercises**:
  - Reference of workout identifier and exercise name cannot be null (NOT NULL)
  - Default values were implemented on muscle groups and equipment needed
  - ON DELETE CASCADE
- **Nutrition plan**:
  - Foreign key reference to user identifier and cannot be null (NOT NULL)
  - ON DELETE CASCADE
- **Meal plan**:
  - Foreign key reference to plan identifier and cannot be null (NOT NULL)
  - Meal name cannot be null (NOT NULL)
  - Default value implemented on meal recipe
- **Progress Tracking**:
  - Foreign key reference to user and cannot be null (NOT NULL)
  - Date cannot be null and defaults to current date (NOT NULL, DEFAULT)
  - ON DELETE CASCADE

- **Following Triggers were implemented**
    - Default Nutrition plan
        - To insert a default nutrition plan when user data is being inserted
    - Default Workouts plan
        - To insert a default Workout plan when user data is being inserted