TDT4145 Databases and Database Management Systems

> Database project Delivery 2 by Ola Moen, Peder Smith, Sigrid Kvamme, Sigurd Totland

Setup

You can either have a `mysql` server running on your local machine at port `3306` with the stock root/root as username and password, or run the attached `docker-compose` file using the following command.

```
$ docker-compose up -d
```

Extract the `dist` folder from the `dist.zip`. It comes bundeled witht the `jdbc-mysl.jar` file and the fixtures and models folders. Run the program using

```
java -jar jdbc-mysql.jar.
```

Description

The program have the following classes.

- `JDBC.java` - Main Entrypoint, holds constants and keeps asking the user for input.
- `BaseController` - Abstract controller, holds variables and methods that are needed for other controllers.
- `DatabaseController` - Main database controller, is responsible for creating/termination connections.
- `ScriptRunner` - Class for running external `sql` scripts on the .sql format.
- `Output` - Class for printing menus and sql output to the terminal.
- `[TableName]Controller` - Classes for controlling their respective fields in the database.

The progam uses the following models and fixtures

- `models/models.sql` - Creates the database schema.
- `fixtures/[Table].sql` - Populates the rows with fake data.

Tasks

Task 1 Register equipment, exercises and workout with associated data.

- Start program with empty database
- Press 1 to migrate database schema

> Register equipment

- Press 4 to access the Equipment menu
- Press 1 to fetch all rows (should be 0)
- Press 2 to add new Equipment (enter name and description)
- Press 1 to see that the Equipment is there

> Register exercise

- Press 5 to access the Exercise menu
- Press 1 to fetch all rows (should be 0)
- Press 2 to add new Exercise (select equipment id = 1, the one you just created)
- Press 1 to see that the Exercise is there

> Register workout

- Press 7 to access the Workout menu
- Press 1 to fetch all rows (should be 0)
- Press 2 to add new Workout (select exercise id = 1, the one you just created)
- Press 1 to see that the Workout is there

> Register note to workout

- Press 6 to access the Note menu
- Press 1 to fetch all notes (should be 0)
- Press 2 to add new Note (select workout id = 1, the one you just created)
- Press 1 to see that the note is created
- Navigate to the workout menu and see that the note is added

Task 2 Get n latest workouts with their notes, where n is chosen by the user.

- Press 3 to drop database if you entered rows in task 1
- Press 1 to migrate database schema
- Press 2 to load fixtures
- Press 7 to access the Workout menu
- Press 1 to see the current Workouts
- Press 3 to and enter the number 3 to see the 3 last Workouts with there notes

Task 3 For every exercise, retrieve a result log for a given time interval where the endpoints of the interval is specified by the user.

- Press 5 to access the Exercise menu
- Press 1 to see all exercise in the system
- Press 4 to fetch the result log
  - Enter 1 to fetch all Joggings
  - Enter startTime *2019-03-09 13:00:00*
  - Enter endTime *2019-03-10 13:00:00*

Task 4 Create new exercise groups and find similar exercises.

> Create new exercise groups

- Press 8 to access the Group menu
- Press 1 to see the current Groups
- Press 2 to create a new Group
- Press 1 to see that the new Group is there

> Find similar exercises (Method 1)

- Press 8 to access the Group menu
- Press 1 to all Groups
- Press 3 to find exercises in Group
- Enter "Arms" to get all exercises under this group

> Find similar exercises (Method 2)

- Press 5 to access the Exercise Menu
- Press 1 to see all exercises
- Press 3 to find similar exercises
- Enter "Pull-up" to get all similar exercises

Task 5 A self-chosen use case.

> UseCase 1

- Ability to `fetchAll` for every table in the system

> UseCase 2

- Ability to `drop`, `migrate`, and `load` SQL from the menu