

Database Systems

David Chehet

Fitness Application 3

1. last_workout_day

```
CREATE DEFINER=`chehet25`@`%` FUNCTION `last_workout_day`(User_Input Integer)
RETURNS int
    DETERMINISTIC
BEGIN
    DECLARE WorkoutDay Integer;
    SELECT wo.WorkoutDayID_WorksOut
    INTO WorkoutDay
    FROM WorksOut wo
    JOIN User u ON u.UserID = wo.UserID_WorksOut
    WHERE u.UserID = User_Input
    ORDER BY wo.Date DESC
    LIMIT 1;

    RETURN IFNULL(WorkoutDay, 0);
END
```

2. muscles_worked

```
CREATE DEFINER=`chehet25`@`%` FUNCTION `muscles_worked` (WorkoutDay Integer)
RETURNS varchar(30) CHARSET utf8mb4
    DETERMINISTIC
BEGIN
    DECLARE muscles VARCHAR(30);

    SELECT GROUP_CONCAT(DISTINCT fa.PrimaryMuscleGroup SEPARATOR '/')
    INTO muscles
    FROM FitnessActivity fa
    JOIN Scheduled s ON fa.FitnessActivityID = s.FitnessActivityID_Scheduled
    JOIN WorkoutDay wd ON s.WorkoutDayID_Scheduled = wd.WorkoutDayID
    WHERE wd.WorkoutDayID = WorkoutDay
        AND fa.PrimaryMuscleGroup NOT IN ('multiple', 'shoulders');

    RETURN muscles;
END
```

3. **next_workout_day** -> For this, I refactored my **last_workout_day** code to return a **CASE** in which I add 1 to the **last_workout_day**, and use the **CASE** to handle someone who hasn't worked out yet or is on day 6.

```
CREATE DEFINER=`chehet25`@`%` FUNCTION `next_workout_day` (User_Input Integer)
RETURNS int
DETERMINISTIC
BEGIN
    DECLARE WorkoutDay Integer;

    SELECT wo.WorkoutDayID_WorksOut
    INTO WorkoutDay
    FROM WorksOut wo
    JOIN User u ON u.UserID = wo.UserID_WorksOut
    WHERE u.UserID = User_Input
    ORDER BY wo.Date DESC
    LIMIT 1;

    RETURN
    CASE
        WHEN WorkoutDay + 1 = 7 THEN 1
        WHEN WorkoutDay IS NULL THEN 1
        ELSE WorkoutDay + 1
    END;

END
```

Procedures

build_workout :

```
CREATE DEFINER=`chehet25`@`%` PROCEDURE `build_workout`(User_ID INT,  
Operation INT)  
BEGIN  
  
INSERT INTO WorksOut (UserID_WorksOut, WorkoutDayID_WorksOut,  
FitnessActivityID_WorksOut, Date)  
  
SELECT DISTINCT  
  (SELECT UserID FROM User u WHERE u.UserID = User_ID),  
  wd.WorkoutDayID, fa.FitnessActivityID AS 'Activity', curdate()  
FROM WorksOut wo  
JOIN WorkoutDay wd ON wo.WorkoutDayID_WorksOut = WorkoutDayID  
JOIN FitnessActivity fa ON fa.FitnessActivityID = wo.FitnessActivityID_WorksOut  
  WHERE NOT EXISTS  
    (SELECT wo.Date  
     FROM WorksOut wo  
     WHERE wo.Date = curdate()  
     AND wo.UserID_WorksOut = User_ID)  
  
AND wd.WorkoutDayID = Operation;  
  
END
```

fetch_current_workout:

```
CREATE DEFINER=`chehet25`@`%` PROCEDURE `fetch_current_workout`(User_ID INT)
BEGIN
```

```
    SELECT fa.Name AS 'Activity', wo.Date, wo.Minutes, wo.Reps, wo.MaxWeight
    FROM WorksOut wo
    JOIN FitnessActivity fa ON fa.FitnessActivityID = wo.FitnessActivityID_WorksOut
    WHERE wo.UserID_WorksOut = User_ID
           AND wo.Date = (
                        SELECT MAX(wo.Date)
                        FROM WorksOut wo
                        WHERE UserID_WorksOut = User_ID
                        );
```

```
END
```