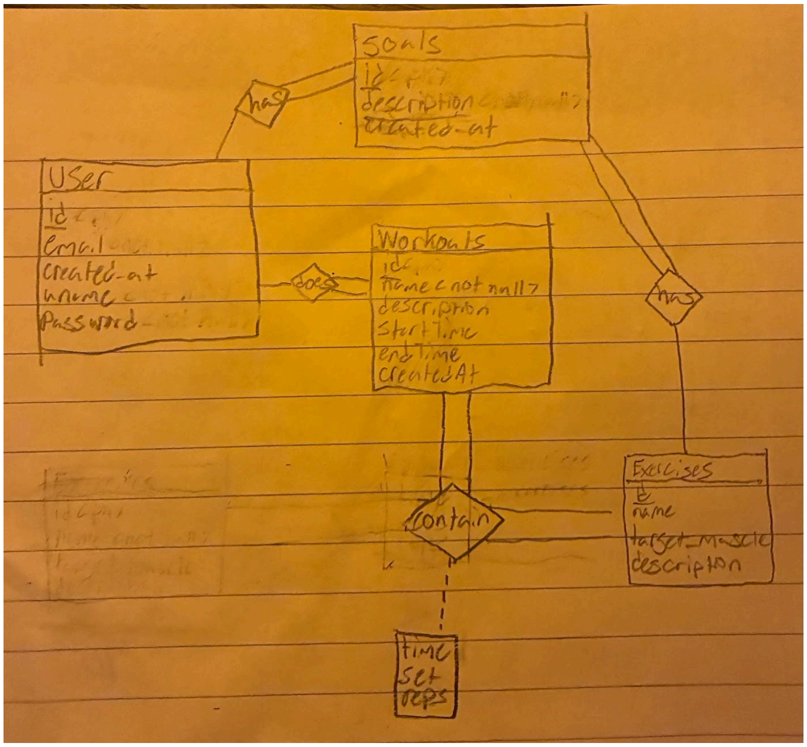
- First, use **ER modeling** to capture the data aspects for your application.
- The ER model must have **at least 4 meaningful entity sets** and it should capture relationships among those entities.



- Then, apply **'ER to table' conversion** rules to generate the schema for the database with appropriate constraints.

users(id, email, created_at, uname, password)

Pk: (id)

Not null: (uname, password, email)

Unique: (uname, email)

exercises(id, name, target_muscle, description)

Pk: (id)

Not null: (name)

Unique: (name)

goals(id, description, created_at, user_id, exercise_id)

Pk: (id)

Not null: (description, user_id)

Fk: (exercise_id) refers to exercises(id)

Fk: (user_id) refers to users(id)

workouts(id, name, user_id, description, startTime, endTime, created_at)

Pk: (id)

Not null: (name, user_id)

Unique: (name)

Fk: (user_id) refers to users(id)

workout_exercises(workout_id, exercise_id, time, sets, reps)

Pk: (workout_id, exercise_id)

Fk: (workout_id) refers to workouts(id)

Fk: (exercise_id) refers to exercises(id)

- Apart from the ER model, use normalization theory to come up with tables. Start with a wide table with all of your application attributes, then **Normalize this table** into BCNF or 3NF normal forms. In order to do this, identify the **functional dependencies** of your application.

Wide_Table(user_id, email, uname, password, user_created_at, workout_id, workout_name, workout_desc, start_time, end_time, workout_created_at, exercise_id, exercise_name, target_muscle, exersice_desc, sets, reps, time, goal_id, goal_desc, goal_created_at)

FD’s:

User_id → email, uname, password, user_created_at

Email → user_id

Uname → user_id

Workout_id → workout_name, workout_desc, start_time, end_time, workout_created_at, user_id

Exercise_id → exercise_id → exercise_name, target_muscle, exercise_desc

Workout_id, exercise_id → sets, reps, time

Goal_id → user_id, exercise_id, goal_desc, goal_created_at

Convert each into single-RHS:

User_id → email

User_id → uname

User_id → password

User_id → user_created at

Email → user_id

Uname → user_id

Workout_id → workout_name

Workout_id → workout_desc

Workout_id → start_time

Workout_id → end_time

Workout_id → workout_created_at

Workout_id → user_id

Exercise_id → exercise_name

Exercise_id → target_muscle

Exercise_id → exercise_desc

Workout_id, exercise_id → sets

Workout_id, exercise_id → reps

Workout_id, exercise_id → time

Goal_id → user_id

Goal_id → exercise_id

Goal_id → goal_desc

Goal_id → goal_created_at

Merge FDs with Same LHS:

User_id → email, uname, password, user_created_at

Email → user_id

Uname → user_id

Workout_id → workout_name, workout_desc, start_time, end_time, workout_created_at, user_id

Exercise_id → exercise_name, target_muscle, exercise_desc

Workout_id, exercise_id → sets, reps, time

Goal_id → user_id, exercise_id, goal_desc, goal_created_at

Create One relation for each FD

R1(user_id, email, uname, password, user_created_at)

R2(email, user_id)

R3(uname, user_id)

R4(workout_id, workout_name, workout_desc, start_time, end_time, workout_created_at, user_id)

R5(exercise_id, exercise_name, target_muscle, exercise_desc)

R6(workout_id, exercise_id, sets, reps, time)

R7(goal_id, user_id, exercise_id, goal_desc, goal_created_at)

Remove Subset Relations:

R1(user_id, email, uname, password, user_created_at)

R4(workout_id, workout_name, workout_desc, start_time, end_time, workout_created_at, user_id)

R5(exercise_id, exercise_name, target_muscle, exercise_desc)

R6(workout_id, exercise_id, sets, reps, time)

R7(goal_id, user_id, exercise_id, goal_desc, goal_created_at)

Rename:

users(user_id, email, uname, password, user_created_at)

workouts(workout_id, workout_name, workout_desc, start_time, end_time, workout_created_at, user_id)

exercises(exercise_id, exercise_name, target_muscle, exercise_desc)

workout_exercises(workout_id, exercise_id, sets, reps, time)

goals(goal_id, user_id, exercise_id, goal_desc, goal_created_at)

- After that, use the **docker based MySQL instance** we use in the class to create your database schema. If the tables you found from your ER model are different from the tables you found from normalization, **explain the differences briefly** then make a decision on which one to go with. **Justify your choice** with a brief comment in the **db_design.pdf** file (see deliverables).

We actually get the same relations through ER modeling and 3NF synthesis, showing that our relations are very strongly chosen.