**Relation Schemes After eliminating redundancies**

Client(AppStore\_id , client id, name, age, gender, weight, health status, training background, credit card )

Trainer(AppStore\_id , trainer id, name, age, gender, weight, health status, mothertongue, experience, price)

Training Program(TrainingProgram\_id, client\_id, repetition and set amount of exercises, name of the program)

Dietician(Name, age, background, experience, degree of profession, dietician id)

Diet Program(DietProgramID, client\_id, Meals, time period of the meals to be eaten)

Gym(gym id, Weights brand, machines brand, strecthing bands brand, medical equipents amount, address, staff amount)

Trainer Finder Application(name, Platforms which this application is available at, category of the app, price of the app, Publisher of the app, owner of the app, AppStore\_id)

App Admin(AppStore\_id , Tasks, position, Application admin id)

Company(Name, location)

Employ(trainer id, Name)

trainer\_gym(trainer id, gym id)

~~train(trainer id, client id)~~

client\_gym(client id, gym id)

Subscribe(trainer id, client id)

make\_training\_program(trainer id, TrainingProgram\_id)

~~has\_a\_training\_program(TrainingProgram\_id, client id)~~

~~has\_a\_diet\_program(DietProgramID, client id)~~

make\_a\_diet\_program(dietician id, client id)

~~trainer\_app(trainer id, AppStore\_id)~~

~~client\_app(client id, AppStore\_id)~~

~~moderate(Application admin id, AppStore\_id)~~

Changes I had made (at the previous Project proposal file):

-- I changed:

-- "application account ... id" into "...ID"

-- For example: "application account trainer id" into "trainerID"

-- (at dietician table) "dietician\_name" and "client\_name" with "clientID" and "trainerID" (these ID's are foreign keys also) since with referencing those from their table, these can be accesed

-- (at diet program table) “time\_period\_of\_meals “ into “time\_period\_of\_program”

-- (at gym table) "weights, machines and strecthing\_bands" as "...\_brand"

-- (at gym table) "medical-equipments and staff" as "...\_amount"

-- (at the make\_training\_program table) "clientID" into "TrainingProgram\_id"

-- Because, now after the change I make, "make\_training\_program table" is between "training program" and "trainer"

-- I added:

-- (to training program and diet program tables) "clientID" (as a result of adding "clientID" to the training program table, I also added "clientID" to the "make\_training\_program" table too)

-- (to gym tabale) "gymID"

-- little changes at some descriptons

-- "Platforms which this application is available at" into "availablePlatforms"

-- "degree of profession" into "professionDegree" etc.

-- I removed:

-- "application account" from all tables that contain that

-- (at the GYM table) dieticians. From now on, lets think like clients does not reach "dieticians" from "gym" but they reach via the app

-- (at the GYM table) trainers. Because at the "train" relationship, I can see all trainers at the gym better

-- (at client table) subscribed trainer as "trainerID" and made that a foreign key

-- (at trainer table) subscribed client as "clientID" and made that a foreign key

-- (train table) whole train relation (in other words train table).

-- (at dietician table) client\_name

-- (at DietProgram table) clientID and dieticianID

-- (at trainingProgram table) exercises

--(at company table) removed “trainers” since we already have that at the “employ” relation (in other words table)