

Opis modeli Django

Klasa "Exercise" określa ćwiczenia jakie będą do wykonania. Jej kluczem głównym jest id ćwiczenia. Posiada też nazwę ćwiczenia, które mięśnie obejmuje, instrukcje jak poprawnie wykonać te ćwiczenie, trudność (łatwy, trudny, średni itd.) oraz zdjęcie przedstawiające te ćwiczenie. Funkcja str odwołuje się do tego ćwiczenia jako string.

```
class Exercise(models.Model):
    id = models.AutoField(primary_key = True)
    name = models.CharField(max_length = 100)
    muscles = models.CharField(max_length = 100)
    instructions = models.TextField()
    difficulty = models.CharField(max_length = 30)
    image = models.ImageField(upload_to = 'mainapp/images')

    def str(self):
        return f"{self.name} - Difficulty: {self.difficulty}, Instructions: {self.instructions}"
```

Klasa "Challenge" określa wyzwanie. Kluczem głównym tej klasy jest id. Ma nazwę, np. "100 dni przysiadów", "cały tydzień nóg", datę rozpoczęcia oraz **zakończenia i określenie czy jest aktywna??**.

```
class Challenge(models.Model):
    id = models.AutoField(primary_key = True)
    name = models.CharField(max_length = 100)
    date_start = models.DateTimeField(default = timezone.now)
    date_end = models.DateTimeField()
    is_active = models.BooleanField(default = False)

    def str(self):
        return f"{self.name} - Start Date: {self.date_start}, End Date: {self.date_end}"
```

"User" to użytkownik. Kluczem głównym jest id użytkownika. Posiada imię, nazwisko, adres e-mail, datę założenia konta, datę urodzin, płeć, wzrost i wagę oraz nazwę użytkownika wraz z hasłem do konta.

```
class User(models.Model):
    id = models.AutoField(primary_key = True)
    name = models.CharField(max_length = 10)
    surname = models.CharField(max_length = 30)
    email = models.EmailField()
    date_joined = models.DateTimeField(default = timezone.now)
    birthday = models.DateField()
    gender = models.CharField(max_length = 10)
    height = models.IntegerField()
    weight = models.IntegerField()
    username = models.CharField(max_length = 20)
    password = models.CharField(max_length = 100)

    def str(self):
        return f"{self.username}"
```

"Challenge_Goal" określa cele (liczbę) jaką konkretnie chcemy zdobyć podczas wykonywania konkretnego challenge. Łączy klasę "Challenge" z "Exercise", ponieważ "Wyzwanie" może składać się z więcej niż jednego ćwiczenia, i dla każdego ćwiczenia tutaj określamy ilość powtórzeń.

```
class Challenge_Goal(models.Model):
    id_challenge = models.ForeignKey(Challenge, on_delete = models.CASCADE, primary_key = True)
    id_exercise = models.ForeignKey(Exercise, on_delete = models.CASCADE, primary_key = True)
    goal = models.IntegerField()
```

"Personal_Highscore" opisuje osobisty rekord każdego użytkownika. Przyda nam się potem do rywalizacji między userami. Klucze obce: id ćwiczenia (każde ćwiczenie będzie miało osobisty rekord każdego użytkownika), id użytkownika,

który zdobył ten wynik - te klucze obce tworzą klucz główny. Najwyższy wynik uzyskany oraz data zdobycia tego rekordu.

```
class Personal_Highscore(models.Model):
    id_exercise = models.ForeignKey(Exercise, on_delete = models.CASCADE, primary_key = True)
    id_user = models.ForeignKey(User, on_delete=models.CASCADE, primary_key=True)
    highscore = models.IntegerField()
    date_got = models.DateTimeField(default = timezone.now)

    def str(self):
        return f"{self.highscore}"
```

"**Exercise_Highscore**" określa najwyższy zdobyty wynik wykonując dane ćwiczenie, na podstawie rekordów wszystkich użytkowników. Klucze obce: id ćwiczenia - jest w tym samym czasie kluczem głównym, id użytkownika, który zdobył ten wynik. Najwyższy wynik uzyskany oraz data zdobycia tego rekordu.

```
class Exercise_Highscore(models.Model):
    id_exercise = models.ForeignKey(Exercise, on_delete = models.CASCADE, primary_key = True)
    highscore = models.IntegerField()
    id_user = models.ForeignKey(User, on_delete = models.CASCADE)
    date_got = models.DateTimeField(default = timezone.now)

    def str(self):
        return f"{self.highscore}"
```

Klasa "**Exercise_Log**" zapisuje dane dotyczące wykonywania danego ćwiczenia. Na klucz główny składają się klucze obce: id wykonanego ćwiczenia oraz id użytkownika, który wykonał to ćwiczenie. Datę wykonania ćwiczenia, id treningu w którym zawarte było ćwiczenie, jak długo trwało wykonanie go (w przypadku np. deski), ciężar jaki powinno się przyjąć (dla ćwiczeń siłowych), ilość powtórzeń i dystans przebyty (w przypadku biegania lub spaceru)??

```

class Exercise_Log(models.Model):
    id_exercise = models.ForeignKey(Exercise, on_delete = models.CASCADE, primary_key = True)
    id_user = models.ForeignKey(User, on_delete = models.CASCADE, primary_key = True)
    date = models.DateTimeField(default = timezone.now, primary_key = True)
    id_workout_log = models.ForeignKey(Workout_Log, on_delete = models.CASCADE)
    duration = models.IntegerField()
    weight = models.IntegerField()
    reps = models.IntegerField()
    distance = models.IntegerField()

```

"Workout" określa trening. Kluczem głównym jest id konkretnego treningu, posiada też nazwę treningu, jego opis oraz zdjęcie.

```

class Workout(models.Model):
    id = models.AutoField(primary_key = True)
    name = models.CharField(max_length = 100)
    description = models.TextField()
    image = models.ImageField(upload_to = 'mainapp/images')

    def str(self):
        return f"{self.name} - {self.description}"

```

Klasa "Workout_Exercise" łączy klasę "Workout" z "Exercise", aby do treningu móc dodać kilka ćwiczeń. Na klucz główny składają się: id treningu, id ćwiczenia, ilość serii i powtórzeń, ewentualne ciężary potrzebne lub długość wykonywania ćwiczenia.

Tu jest sets a w Exercise_log nie było, a w Exercise-log było "distance" a tu nie ma. Trzeba to zjednoczyć.

```

class Workout_Exercise(models.Model):
    id_workout = models.ForeignKey(Workout, on_delete = models.CA

```

```
SCADE, primary_key = True)
id_exercise = models.ForeignKey(Exercise, on_delete = models.
CASCADE, primary_key = True)
sets = models.IntegerField()
reps = models.IntegerField()
weight = models.IntegerField()
duration = models.IntegerField()
```

"Workout_Log" zapisuje informacje o zrobionym treningu. Id wykonanego treningu oraz id użytkownika, który to zrobił wchodzi w skład klucza głównego. Zawarta jest też data wykonania oraz **czas potrzebny na wykonanie go**.

```
class Workout_Log(models.Model):
    id_workout = models.ForeignKey(Workout, on_delete = models.CA
SCADE, primary_key = True)
    id_user = models.ForeignKey(User, on_delete = models.CASCADE,
primary_key = True)
    date = models.DateTimeField(default = timezone.now, primary_k
ey = True)
    duration = models.IntegerField()
```