

**1. Nazwa aplikacji – AwareFit**

**2. Cel:** stworzenie aplikacji powiązanej z bazą danych, która ma za zadanie zapisywać treningi siłowe użytkownika oraz monitorować postępy.

**3. Odzwierciedlenie rzeczywistości:**

- Użytkownik rozpoczyna trening siłowy, wykonuje go oraz zapisuje wszystkie ćwiczenia z podziałem na partie mięśniowe, dokładną ilość serii oraz powtórzeń z konkretnym obciążeniem.
- Wraca do domu i otwiera aplikację, loguje się na swoje konto oraz zapisuje trening:
  - Wybiera dodaj **ćwiczenie** -> wyświetla się lista **partii mięśniowych** (użytkownik wybiera jedną z nich) -> wyświetla się lista ćwiczeń angażujących wybraną partię (użytkownik wybiera konkretne ćwiczenie, które wykonał na treningu)
  - Dla każdego ćwiczenia może dodać **serię** (każda seria to **obciążenie** podane w kilogramach oraz **liczba powtórzeń** wykonanych tym ciężarem)
  - Użytkownik dodaje kolejne serie w takiej samej postaci tak aby odzwierciedlić rzeczywiście wykonany trening
  - Następnie może dodać kolejne ćwiczenia tworząc w ten sposób trening, o konkretnej dacie, który zapisuje się w bazie danych.
- Użytkownik ma również możliwość regularnego zapisywania swoich pomiarów ciała tj. waga, wzrost, obwody poszczególnych partii mięśniowych. Dzięki temu możliwe będzie monitorowanie jak zmienia się jego ciało.
- Dodana zostanie również zakładka, w której użytkownik na podstawie swoich treningów będzie mógł monitorować swój postęp na podstawie różnych statystyk wyliczanych za pomocą danych, i algorytmów zapisywanych w bazie danych.

**4. Działanie bazy:**

**Scenariusz 1:**

**1)** Użytkownik wraca do domu, otwiera aplikację i się loguje.

GUI: Wysyła SELECT do tabeli users, aby sprawdzić hasło.

**2)** Użytkownik klika "Dodaj trening". Wybiera partie , ćwiczenia , a następnie wpisuje serie (np. 100 kg, 5 powtórzeń).

**3)** Użytkownik klika "Zapisz Trening".

GUI: Rozpoczyna transakcję i wysyła serię poleceń INSERT do tabel bazowych:

INSERT INTO workouts (user\_id, date) VALUES (...)

INSERT INTO workout\_exercises (workout\_id, exercise\_id) VALUES (...)

INSERT INTO workout\_sets (workout\_exercise\_id, weight, reps, set\_number) VALUES (...)

INSERT INTO workout\_sets (...) (dla każdej kolejnej serii)

**4)** Baza Danych (PostgreSQL): Po prostu zapisuje te surowe dane w tabelach. Nic nie oblicza, nic nie agreguje. Działa błyskawicznie.

**5)** To samo dotyczy zapisywania pomiarów ciała – to zwykły INSERT do body\_measurements

## Scenariusz 2: Monitorowanie Postępów (Operacje **READ** z użyciem Widoków):

1) Użytkownik klika w zakładkę z monitorem progresu:

**GUI (Interfejs):** Nie wykonuje żadnych obliczeń. Wysyła do bazy bardzo proste zapytanie:

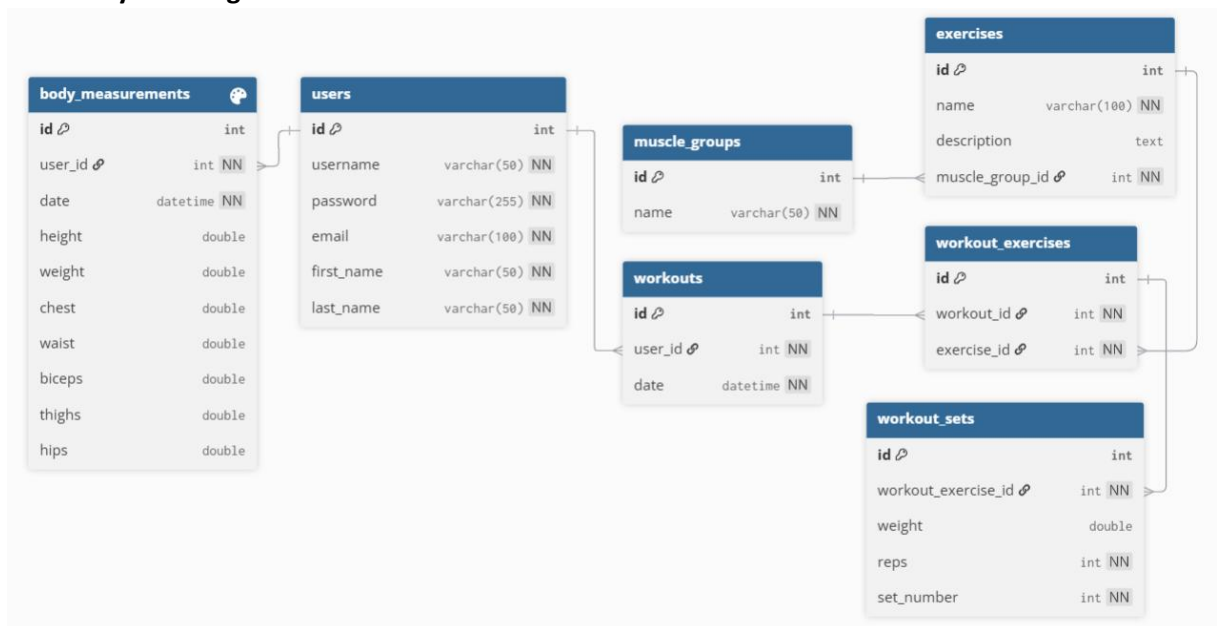
```
SELECT date, total_volume FROM [utworzone widoki] WHERE user_id = ? ORDER BY date;
```

- Baza danych otrzymuje proste zapytanie.
- Widzi że dotyczy ono widoków.
- Uruchamia skomplikowane zapytanie SQL złączające kilka tabel.
- Wykonuje pracę po stronie serwera
- Zwraca do GUI tylko gotowe wyniki poszczególnych zapytań

## 5. Technologie do wykorzystania

- Baza danych: PostgreSQL
- GUI: Java + JavaFX / PHP + HTML + CSS + JS

## 6. Baza danych – Diagram ERD:



## 7. Pytania problemowe:

Pytania te można zaimplementować jako skomplikowane zapytania SQL, funkcje bazodanowe (w PostgreSQL ) lub widoki (VIEWS), aby dostarczyć dane do zakładki ze statystykami.

### Monitorowanie Progresu Treningowego (Objętość i Siła)

Lp.	Pytanie Problemowe	Wymagana Akcja w Bazie Danych (Technika)	Przykład (Wynik)
1.	Jaka jest całkowita objętość treningowa (tonaż) dla konkretnego ćwiczenia w zadanym treningu (np. Wyciskanie na ławce, Trening z 2025-11-17)?	Widok/Funkcja: Zsumowanie iloczynu <code>weight * reps</code> dla wszystkich serii danego ćwiczenia w danym treningu.	Objętość: 3500 kg
2.	Jaki jest szacowany maksymalny ciężar na jedno powtórzenie (e1RM) dla danego ćwiczenia na podstawie najlepszej serii z ostatniego miesiąca?	Funkcja (PL/pgSQL): Wywołanie algorytmu (np. Wzór Epleya) na największej wartości <code>weight</code> zarejestrowanej w serii, gdzie <code>reps</code> było mniejsze lub równe 10.	e1RM: 125 kg
3.	Jak zmieniał się średni ciężar roboczy (Average Working Weight) dla wybranego ćwiczenia (np. Przysiady) w ujęciu miesięcznym przez ostatnie 6 miesięcy?	Widok/Agregacja: Obliczenie średniej ważonej ciężarów ( <code>weight</code> ) dla wszystkich serii danego ćwiczenia w danym miesiącu, pomijając serie rozgrzewkowe (jeśli są oznaczone).	Średni Ciężar (Listopad): 95 kg
4.	Jaka jest całkowita liczba serii wykonanych na daną partię mięśniową (np. Klatka piersiowa) w ciągu ostatniego tygodnia (lub innego okresu)?	Widok/Grupowanie: Zliczenie ( <code>COUNT</code> ) serii z tabeli <code>workout_sets</code> , złączenie z <code>exercises</code> i <code>muscle_groups</code> , filtrowanie po dacie i nazwie partii.	Liczba Serii: 18
5.	Który z dotychczasowych treningów (po dacie) wygenerował największą łączną objętość treningową?	Widok: Obliczenie sumy objętości dla wszystkich ćwiczeń w treningu i posortowanie wyników malejąco.	Data Treningu: 2025-10-28 Objętość: 24 500 kg

### Monitorowanie Mierników Ciała i Statystyk (Body Metrics)

Lp.	Pytanie Problemowe	Wymagana Akcja w Bazie Danych (Technika)	Przykład (Wynik)
6.	Jak zmieniła się masa ciała (wartość bezwzględna i procentowa) użytkownika od momentu rejestracji lub w zadanym okresie (np. ostatnie 90 dni)?	Funkcja/Agregacja: Pobranie pierwszej i ostatniej wartości wagi z <code>body_measurements</code> w danym okresie i obliczenie różnicy.	Zmiana: -4.5 kg (-5.2%)
7.	Jaka jest aktualna wartość wskaźnika BMI użytkownika na podstawie najnowszych pomiarów wagi i wzrostu?	Funkcja (PL/pgSQL): Obliczenie $BMI = \frac{\text{waga (kg)}}{\text{wzrost (m)}^2}$ na podstawie najnowszych wpisów z <code>body_measurements</code> .	Aktualne BMI: 24.8
8.	Jak zmieniała się w czasie wartość wybranego pomiaru obwodu (np. Obwód Pasa lub Udo) i kiedy nastąpiła największa zmiana (wzrost lub spadek) między kolejnymi pomiarami?	Widok/Analiza Czasowa: Zastosowanie funkcji okienkowych (np. <code>LAG()</code> ) w PostgreSQL do porównania wartości obwodu z poprzednim pomiarem w tabeli <code>body_measurements</code> .	Zmiana w Pasiu (Listopad): -1.5 cm

## Statystyki Konsekwencji i Dodatkowe Wskaźniki

Lp.	Pytanie Problemowe	Wymagana Akcja w Bazie Danych (Technika)	Przykład (Wynik)
9.	Jaka jest średnia liczba dni treningowych w miesiącu w ostatnim kwartale?	<b>Agregacja:</b> Zliczenie unikalnych dat treningów ( <code>workouts.date</code> ) i podzielenie przez liczbę miesięcy w okresie.	<b>Średnio Dni:</b> 14
10.	Jakie są trzy najczęściej i najrzadziej wykonywane ćwiczenia w całej historii treningowej użytkownika?	<b>Widok/Grupowanie:</b> Zliczenie wystąpień <code>exercise_id</code> w tabeli <code>workout_exercises</code> i sortowanie malejąco/rosnąco.	<b>Najczęściej:</b> 1. Wyciskanie 2. Przysiady 3. Martwy Ciąg
11.	Jaki jest szacowany "współczynnik regeneracji" mierzony jako średni czas przerwy (w godzinach) między treningami trenującymi tę samą partię mięśniową (np. Nogii)?	<b>Analiza Czasowa (Funkcja):</b> Obliczenie różnicy dat między treningiem, a kolejnym treningiem zawierającym to samo ćwiczenie/partię.	<b>Średni czas regeneracji nóg:</b> 96 godzin

### 1. Obliczenia Agregacyjne i Raportowanie (Widoki i SQL)

Większość pytań dotyczących **historii, sumowania i średnich** (np. objętość, historia treningów, liczba serii, średni ciężar, najczęściej/najrzadziej trenowane) może być rozwiązana za pomocą zaawansowanych zapytań SQL, które zostaną **zapisane jako Widoki (Views)**.

- **Objętość Treningowa:** Obliczana za pomocą prostej agregacji: `SUM(weight * reps)`.
- **Historia Treningów i Serii na Partię:** Wymaga tylko złożonych złączeń (JOINS) i grupowania (GROUP BY).
- **Zmiana Pomiarów Ciała:** Wymaga użycia zaawansowanych funkcji okienkowych (np. `LAG()`), które pozwalają porównać wiersz (obecny pomiar) z wierszem poprzednim (poprzedni pomiar).

### 2. Algorytmy i Wskaźniki (Funkcje PL/pgSQL)

Pytania wymagające zastosowania wzorów matematycznych muszą zostać zaimplementowane w postaci **Funkcji (Stored Functions)** napisanych w języku PL/pgSQL (proceduralne rozszerzenie SQL w PostgreSQL).

- **1RM:** Implementacja wzoru (np. Epleya) do przeliczania `weight` i `reps` na szacowany maks.
- **Wskaźnik BMI:** Implementacja wzoru `waga/wzrost2`.
- **Procentowa Zmiana Masy Ciała:** Funkcja pobierająca dwie wartości wagi z różnych dat i obliczająca procentową różnicę.
- **Współczynnik regeneracji:** Złożona funkcja, która oblicza różnicę czasu między dwoma wydarzeniami (treningami) w szeregu czasowym.

**Jaka to będzie obsługiwane:**

Wymaganie	Przykład Techniki PostgreSQL
<b>Objętość treningowa (Pkt 1, 5)</b>	<b>Widok + Agregacja SQL</b>
<b>1RM, BMI (Pkt 2, 7)</b>	<b>Funkcja (PL/pgSQL)</b>
<b>Średni ciężar, Liczba serii (Pkt 3, 4, 10)</b>	<b>Widok + Grupowe funkcje SQL</b>
<b>Zmiana pomiarów ciała (Pkt 6, 8)</b>	<b>Widok + Funkcje Okienkowe (LAG)</b>
<b>Historia treningów (Pkt 4)</b>	<b>Proste Zapytanie SELECT (Widok)</b>
<b>Współczynnik regeneracji (Pkt 11)</b>	<b>Funkcja + Funkcje Okienkowe</b>