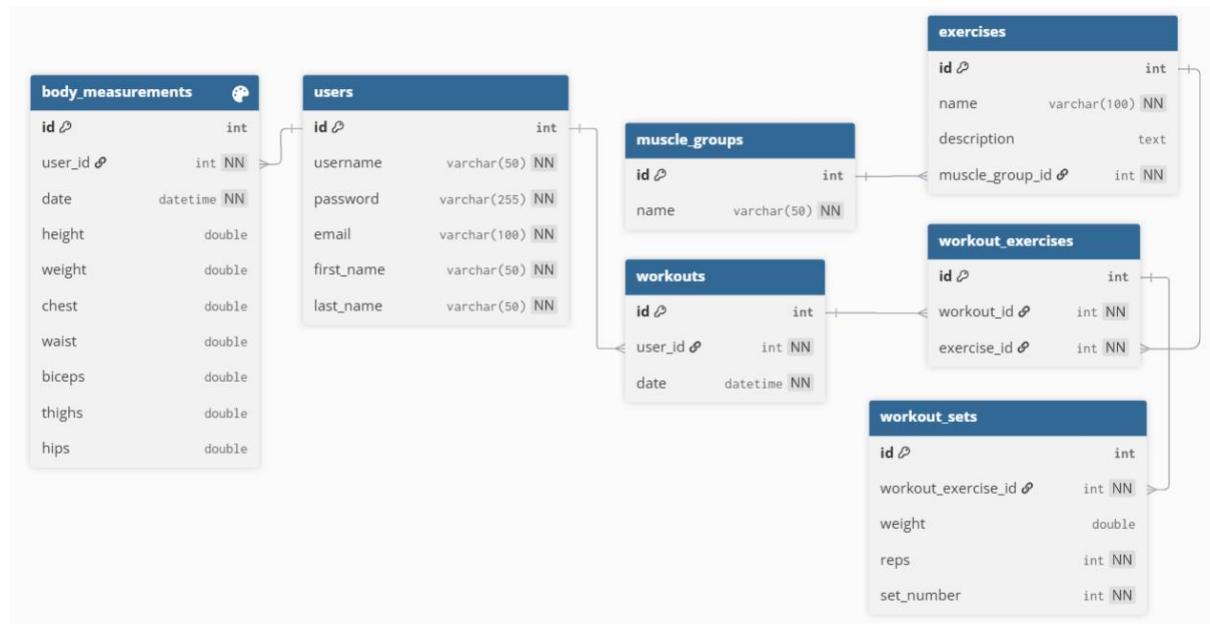


Oliwier Hędrzak [134913] i Michał Janik [134915] Lab3 – proponowane tematy projektu:

1. System zarządzania treningami siłowymi

System umożliwiać ma zalogowanie się użytkownika do systemu i zarządzanie własnym dziennikiem treningów siłowych. Po zalogowaniu się użytkownik może przeglądać historię swoich treningów, dodawać nowe sesje, a także przeglądać listę ćwiczeń pogrupowanych według partii mięśniowych wraz z opisem i sposobem ich wykonania. Zapisywanie wszystkich treningów w poszczególnymi danymi oraz pomiarów ciała umożliwiać będzie również monitorowanie postępu.

Poniżej przedstawiono wstępny diagram ERD bazy danych stworzonej w celu realizacji projektu:



Opis braku opcjonalności:

- **workout_exercises.workout_id → workouts.id** (Dzięki temu nie możemy stworzyć ćwiczenia bez treningu.)
- **exercises.muscle_group_id → muscle_groups.id** (Bo każde ćwiczenie musi należeć do jakieś grupy mięśniowej. Nie ma ćwiczenia „bez partii ciała”.)
- **exercises.muscle_group_id → muscle_groups.id** (Nie dopuszczamy, aby jakieś pomiary ciała były anonimowe. Zawsze muszą być przypisane do użytkownika.)
- **workout_sets.workout_exercise_id** (Zabieramy możliwość, gdzie jakąś seria nie jest przypisana do ćwiczenia. Np. 10 powtórzeń ćwiczenia NULL.)

Możliwości analizy danych (wiele z nich można zrobić po stronie bazy danych za pomocą widoków):

- liczba treningów w danym okresie
- najczęściej wykonywane ćwiczenia / trenowane partie
- średnia liczba serii lub powtórzeń na ćwiczenie
- objętość treningowa (ciężar * powtórzenia * liczba serii)
- zmiana masy ciała
- zmiana pomiarów ciała
- wskaźnik BMI
- objętość treningowa na grupę mięśniową
- progres siły w danym ćwiczeniu

```
Table users {
    id int [pk]
    username varchar(50) [not null]
    password varchar(255) [not null]
    email varchar(100) [not null]
    first_name varchar(50) [not null]
    last_name varchar(50) [not null]
}

Table body_measurements {
    id int [pk]
    user_id int [ref: > users.id, not null]
    date datetime [not null]
    height double
    weight double
    chest double
    waist double
    biceps double
    thighs double
    hips double
}

Table exercises {
    id int [pk]
    name varchar(100) [not null]
    description text
    muscle_group_id int [ref: > muscle_groups.id, not null]
}

Table muscle_groups {
    id int [pk]
    name varchar(50) [not null]
}
```

```
Table workouts {
    id int [pk]
    user_id int [ref: > users.id, not null]
    date datetime [not null]
}

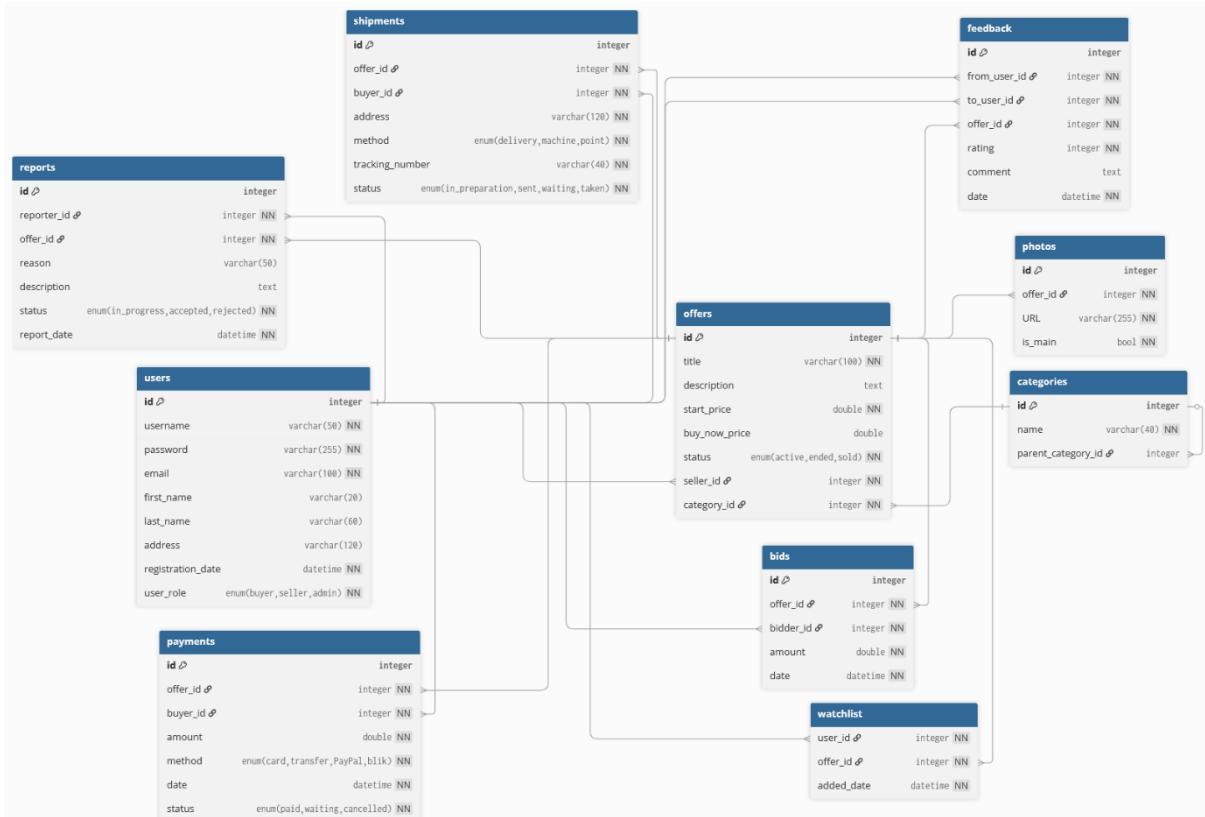
Table workout_exercises {
    id int [pk]
    workout_id int [ref: > workouts.id, not null]
    exercise_id int [ref: > exercises.id, not null]
}

Table workout_sets {
    id int [pk]
    workout_exercise_id int [ref: > workout_exercises.id, not null]
    weight double
    reps int [not null]
    set_number int [not null]
}
```

2. System portalu aukcyjnego

System ma umożliwiać zalogowanie się użytkownika na portal aukcyjny. W zależności od roli (kupujący, sprzedawca, administrator) posiada on oddzielne możliwości. Kupujący może kupować, przeglądać oferty oraz licytacje i wystawiać im opinie i zgłoszenia. Sprzedający może wystawiać oferty oraz licytacje z załączonymi zdjęciami. Administrator może sprawdzać zgodność ogłoszeń z zasadami serwisu i usuwać je w przypadku, kiedy je naruszają. Kupujący ma do wyboru kilka opcji płatności oraz dostawy - prosto do domu kurierem, do paczkomatu lub punktu odbioru.

Poniżej przedstawiono wstępny diagram ERD bazy danych stworzonej w celu realizacji projektu:



Opis braku opcjonalności:

- **users.id → reports.reporter_id**
- **users.id → shipments.buyer_id**
- **users.id → payments.buyer_id**
- **users.id → feedback.from_user_id**
- **users.id → feedback.to_user_id**
- **users.id → offer.seller_id**
- **users.id → bids.bidder_id**
- **users.id → watchlist.user_id**

(Raport, zamówienie, płatność, opinia, oferta, licytacja ani lista obserwacji nie mogą istnieć bez istniejącego użytkownika, nieważne o jakiej roli.)

- **offer.id → reports.offer_id**
- **offer.id → shipments.offer_id**
- **offer.id → payments.offer_id**
- **offer.id → bids.offer_id**
- **offer.id → watchlist.offer_id**
- **offer.id → feedback.offer_id**
- **offer.id → photos.offer_id**

(Raport, zamówienie, płatność, licytacja, lista obserwacji, opinia i załączone zdjęcie nie mogą istnieć bez oferty.)

- **categories.id → categories.parent_category_id**
- **categories.id → offers.category_id**

(Kategoria poboczna ani oferta nie mogą istnieć bez kategorii.)

Możliwości analizy danych (wiele z nich można zrobić po stronie bazy danych za pomocą widoków):

- panel sprzedawcy – sprzedane aukcje, stan magazynu
- koszyk zakupów użytkownika
- wyświetlanie najczęściej kupowanych i najlepiej ocenianych produktów
- wyszukiwanie i sortowanie produktów po określonych kategoriach i filtrach
- wykres rozkładu opinii
- historia (oś czasu) licytacji
- lista obserwowanych ogłoszeń
- śledzenie przesyłki
- podsumowanie transakcji

```

Table users {
    id integer [primary key]
    username varchar(50) [not null]
    password varchar(255) [not null]
    email varchar(100) [not null]
    first_name varchar(20)
    last_name varchar(60)
    address varchar(120)
    registration_date datetime [not null]
    user_role enum(buyer,seller,admin) [not null]
}

Table offers{
    id integer [primary key]
    title varchar(100) [not null]
    description text
    start_price double [not null]
    buy_now_price double
    status enum(active,ended,sold) [not null]
    seller_id integer [not null]
    category_id integer [not null]
}

Table shipments{
    id integer [primary key]
    offer_id integer [not null]
    buyer_id integer [not null]
    address varchar(120) [not null]
    method enum(delivery,machine,point) [not null]
    tracking_number varchar(40) [not null]
    status enum(in_preparation,sent,waiting,taken) [not null]
}

Table categories{
    id integer [primary key]
    name varchar(40) [not null]
    parent_category_id integer
}

Table feedback{
    id integer [primary key]
    from_user_id integer [not null]
    to_user_id integer [not null]
    offer_id integer [not null]
    rating integer [not null]
    comment text
    date datetime [not null]
}

```

```

Table photos{
    id integer [primary key]
    offer_id integer [not null]
    URL varchar(255) [not null]
    is_main bool [not null]
}

Table watchlist{
    user_id integer [not null]
    offer_id integer [not null]
    added_date datetime [not null]
}

Table reports{
    id integer [primary key]
    reporter_id integer [not null]
    offer_id integer [not null]
    reason varchar(50)
    description text
    status enum(in_progress, accepted, rejected) [not null]
    report_date datetime [not null]
}

Table payments{
    id integer [primary key]
    offer_id integer [not null]
    buyer_id integer [not null]
    amount double [not null]
    method enum(card,transfer,PayPal,blik) [not null]
    date datetime [not null]
    status enum(paid,waiting,cancelled) [not null]
}

Table bids{
    id integer [primary key]
    offer_id integer [not null]
    bidder_id integer [not null]
    amount double [not null]
    date datetime [not null]
}

```