# Strona tytułowa

- Imię nazwisko autorów: Kacper Długosz i Bartłomiej Florek

- Numery albumów: 125111 i 125115

- Temat projektu: Muzeum czołgów

- Nazwę przedmiotu: Bazy danych

- Grupa laboratoryjna: lab 1

Spis treści

[Strona tytułowa 1](#_Toc168649622)

[Opis programu: 2](#_Toc168649623)

[Wykorzystane technologie: 2](#_Toc168649624)

[Opis działań 2](#_Toc168649625)

[Diagram ERD: 3](#_Toc168649626)

[Opis działania aplikacji 5](#_Toc168649627)

[Funkcje języka proceduralnego 18](#_Toc168649628)

[Źródła: 24](#_Toc168649629)

# Opis programu:

Muzeum Czołgów to aplikacja desktopowa, która umożliwia zarządzanie kolekcją czołgów. Zalogowani użytkownicy mogą przeglądać czołgi w danych muzeach, a także wyświetlać wszystkie szczegóły czołgów takie jak: nazwa, opis, zdjęcie, amunicja, kraj produkcji itp.

Aby uruchomić program należy:

- Pobrać PostgreSQL

- W pgAdmin4 utworzyć nową bazę, pod nazwa „tanksdb”.

- Zaimportować bazę z pliku „BAZA\_GIT.sql”, która utworzy wszystkie tabele oraz wprowadzi dane.

- Pobierz kod źródłowy z repozytorium.

- Otwórz projekt w środowisku JavaFX (np. IntelliJ IDEA).

- Skonfiguruj połączenie z bazą danych w pliku konfiguracyjnym „ConnectDB”.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

- Uruchom aplikację.

# Wykorzystane technologie:

- JavaFX version 21: Do tworzenia interfejsu użytkownika.

- Java SDK version 21.0.2: Do pisania funkcjonalności

- PostgreSQL version 16: Do przechowywania danych o czołgach.

- pgAdmin4 version 7.6: Do zarządzania bazą

- <https://dbdiagram.io/d> - do przygotowania bazy danych

# Opis działań

Widoki, funkcje po stronie bazy, wyświetlanie czołgów, dodawanie czołgów, sortowanie, widok dla użytkownika oraz dla admina, baza – Florek

Logowanie, tworzenie nowych użytkowników, walidacje, edycje danych, wyszukiwarki - Długosz

# Diagram ERD:

Obraz zawierający zrzut ekranu, tekst, modelowanie 3D, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Rysunek 1 - Diagram ERD

**Opisy tabel:**

**Tabela Użytkowników (users):**

Kolumny:

- user\_id: Liczba całkowita stanowiąca klucz główny identyfikacji użytkownika.

- user\_login: Pole typu varchar przechowujące dane logowania użytkownika.

- user\_email: Pole typu varchar przechowujące adresy e-mail użytkowników.

- user\_pass: Pole typu varchar przechowujące hasła użytkowników.

- acc\_type: Pole typu varchar wskazujące typ konta (np. admin, zwykły użytkownik).

**Tabela Czołgów (tanks):**

Kolumny:

- tank\_id: Liczba całkowita stanowiąca klucz główny identyfikacji czołgu.

- nation\_id: Liczba całkowita odnosząca się do nation\_id z tabeli nationality.

- name: Pole typu varchar zawierające nazwę czołgu.

- type: Pole typu varchar opisujące typ czołgu (np. ciężki, średni, lekki).

- img: Pole typu bytea służące do przechowywania obrazów czołgów.

- description: Pole typu text zawierające opis czołgu.

- amount: Liczba całkowita wskazująca ilość czołgów.

- data: Pole typu date przechowujące daty związane z czołgami

**Tabela Nationality:**

Kolumny:

- nation\_id: Liczba całkowita stanowiąca klucz główny identyfikacji narodowości.

- nation\_name: Pole varchar przechowujące nazwę narodu (np. Niemcy, USA).

- prod\_place: Pole varchar wskazujące miejsce produkcji (np. lokalizacja fabryki).

**Tabela Ammunition:**

Kolumny:

- ammo\_id: Pole varchar stanowiące klucz główny identyfikacji amunicji.

- name: Pole varchar przechowujące nazwę amunicji.

- description: Pole varchar opisujące amunicję.

**Tabela Tank-Ammunition:**

Kolumny:

- ammo\_id: Liczba całkowita odnosząca się do ammo\_id z tabeli ammunition.

- tank\_id: Liczba całkowita odnosząca się do tank\_id z tabeli tanks.

**Tabela Museum Tanks:**

Kolumny:

- museum\_id: Liczba całkowita odnosząca się do museum\_id z tabeli museum.

- tank\_id: Liczba całkowita odnosząca się do tank\_id z tabeli tanks.

**Tabela Museum:**

Kolumny:

- museum\_id: Liczba całkowita stanowiąca klucz główny identyfikacji muzeum.

- name: Pole varchar przechowujące nazwę muzeum.

- localization: Pole varchar wskazujące lokalizację muzeum.

Opis działania aplikacji**:**

**Rejestracja i logowanie:**

Widok rejestracji:

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Rysunek 2 - Formularz rejestracji

Program posiada zabezpieczenia przed wprowadzeniem złych wartości i wyświetla odpowiednie komunikaty:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

Rysunek 3 - Walidacja (puste pola) Rysunek 4 - Walidacja (niepoprawne wartości)

Po wprowadzeniu poprawnych danych zostanie wyświetlony komunikat:

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Rysunek 5 - Poprawne dane rejestracji

W przypadku zapomnienia hasła jest możliwość jego odzyskania przez wpisanie maila:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Rysunek 6 - Formularz odzyskiwania hasła

Widok zwykłego użytkownika po zalogowaniu:

Obraz zawierający tekst, zrzut ekranu, statek powietrzny

Opis wygenerowany automatycznie

Rysunek 7 - Widok użytkownika

Widok admina (admin ma dodatkowa opcje widoczne po lewej stronie panelu):

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 8 - Widok admina

Opcje po kliknięciu w wybrany czołg:

Obraz zawierający tekst, zrzut ekranu, pojazd, broń

Opis wygenerowany automatycznie

Rysunek 9 - Szczegóły danego czołgu

Możliwość edycji danych:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, design

Opis wygenerowany automatycznie

Rysunek 10 - Formularz edycji danych o czołgu

Walidacja danych przy edycji:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, design

Opis wygenerowany automatycznie

Rysunek 11 - Walidacja danych podczas edycji

Poprawne dane przy edycji czołgu:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Rysunek 12 - Poprawne dane podczas edycji

Po wybraniu muzeum wyświetlane są tylko czołgi które się w nim znajdują:

Obraz zawierający tekst, zrzut ekranu, broń, pojazd

Opis wygenerowany automatycznie

Rysunek 13 - Wybór muzeum

Możemy także filtrować czołgi po nacji:

Obraz zawierający tekst, zrzut ekranu, pojazd, oprogramowanie

Opis wygenerowany automatycznie

Rysunek 14 - Filtrowanie czołgów po nacji

Przyciskiem po lewej stronie możemy resetować wcześniej wybrane filtry:

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 15 - Resetowanie filtrowania

Możemy użyć wyszukiwarki do znajdowania konkretnych czołgów w tym przypadku zawierające literę „l”

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 16 - Wyszukiwarka czołgów

Widok wszystkich użytkowników z możliwością ich edycji wraz z walidacja:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Strona internetowa

Opis wygenerowany automatycznie

Rysunek 17 - Widok wszystkich użytkowników

Obraz zawierający tekst, oprogramowanie, zrzut ekranu, Ikona komputerowa

Opis wygenerowany automatycznie

Rysunek 18 - Złe dane podczas próby edycji

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Rysunek 19 - Poprawne dane podczas próby edycji

Obraz zawierający tekst, oprogramowanie, zrzut ekranu, Ikona komputerowa

Opis wygenerowany automatycznie

Rysunek 20 - Złe dane podczas próby edycji

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Rysunek 21 - Poprawne dane podczas próby edycji

Możemy wyszukiwać użytkowników:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Strona internetowa

Opis wygenerowany automatycznie

Rysunek 22 - Wyszukiwanie użytkowników

Widok wszystkich nacji, który działa na tej samej zasadzie co lista użytkowników, tzn. wyszukiwarka oraz edycja wraz walidacja:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Strona internetowa

Opis wygenerowany automatycznie

Rysunek 23 - Lista nacji i wyszukiwanie

Dodawanie nowego czołgu (walidacja):

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Rysunek 24 - Walidacja formularza przy dodawaniu czołgu

W przypadku podania prawidłowych danych: Kiedy nie wybierzemy zdjęcia, aplikacja ustawi zdjęcie domyślne:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Rysunek 25 - Dodanie czołgu

Obraz zawierający tekst, zrzut ekranu, Czcionka, logo

Opis wygenerowany automatycznie

Rysunek 26 - Domyślne zdjęcie czołgu

Dodawanie nowej nacji wraz z walidacją danych:

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Rysunek 27 - Walidacja formularza przy dodawaniu nowej nacji

W przypadku wprowadzenie poprawnych danych:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

Rysunek 28 - Dodanie nowej nacji



Rysunek 29 - Dodana wcześniej nacja w liście

Dodawanie nowego muzeum wraz z walidacja: Miejscowość może być pusta wtedy aplikacja ustawi domyślną wartość na 'Warszawę':

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Rysunek 30 - Walidacja nazwy

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie Obraz zawierający tekst, zrzut ekranu, oprogramowanie, System operacyjny

Opis wygenerowany automatycznie

Rysunek 31 – Komunikat niepoprawnych danych Rysunek 32 - Komunikat dodania muzeum

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 33 - Dowód dodania muzeum

Dodawanie nowego rodzaju amunicji wraz z walidacja: Pole opis może być puste wtedy aplikacja ustawi domyślny opis

Obraz zawierający tekst, zrzut ekranu, design

Opis wygenerowany automatycznie

Rysunek 34 - Walidacja przy dodawaniu amunicji

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 35 - Komunikat przy poprawnym dodaniu amunicji

Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie

Rysunek 36 – Dowód

Edycja zdjęcia czołgu:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Rysunek 37 - Formularz edycji zdjęcia

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 38 - Komunikat

Obraz zawierający tekst, zrzut ekranu

Opis wygenerowany automatycznie

Rysunek 39- Nowe zdjęcie czołgu

Funkcje języka proceduralnego**:**

**Funkcje:**

**- check\_no\_digits:**

CREATE OR REPLACE FUNCTION check\_no\_digits(input\_text TEXT)

RETURNS BOOLEAN AS $$

BEGIN RETURN NOT (regexp\_match(input\_text, '[0-9]') IS NOT NULL);

END;

$$ LANGUAGE plpgsql;

**Opis:** Funkcja sprawdzająca, czy w podanym tekście nie znajdują się żadne cyfry. Argumenty: input\_text TEXT - tekst do sprawdzenia. Zwraca: BOOLEAN - wartość TRUE, jeśli w tekście nie ma cyfr, w przeciwnym razie FALSE.

**- is\_valid\_email**

CREATE OR REPLACE FUNCTION is\_valid\_email(email TEXT)

RETURNS BOOLEAN AS $$

BEGIN RETURN email ~\* '^[A-Za-z0-9. %+-]+@[A-Za-z0-9.-]+.[A-Z|a-z]{2,}$';

END; $$ LANGUAGE plpgsql;

**Opis:** Funkcja sprawdzająca poprawność adresu email. Argumenty: email TEXT - adres email do sprawdzenia. Zwraca: BOOLEAN - wartość TRUE, jeśli email jest poprawny, w przeciwnym razie FALSE

**- is\_valid\_password**

CREATE OR REPLACE FUNCTION is\_valid\_password(password TEXT)

RETURNS BOOLEAN AS $$ BEGIN RETURN LENGTH(password) >= 8;

END;

$$ LANGUAGE plpgsql;

**Opis**: Funkcja sprawdzająca, czy hasło spełnia określone wymagania (długość, obecność cyfr, małych i wielkich liter oraz znaków specjalnych). Argumenty: password TEXT - hasło do sprawdzenia. Zwraca: BOOLEAN - wartość TRUE jeśli hasło spełnia wymagania, w przeciwnym razie FALSE

**-is\_valid\_acc\_type**

CREATE OR REPLACE FUNCTION is\_valid\_acc\_type(acc\_type VARCHAR)

RETURNS BOOLEAN AS $$

BEGIN RETURN acc\_type = 'admin'

OR acc\_type = 'user';

END; $$

LANGUAGE plpgsql;

**Opis:** Ta funkcja is\_valid\_acc\_type sprawdza, czy podany typ konta użytkownika jest prawidłowy. Przyjmuje jeden argument acc\_type typu VARCHAR, który reprezentuje typ konta. Funkcja zwraca wartość logiczną TRUE, jeśli typ konta jest równy "admin" lub "user", a w przeciwnym razie zwraca FALSE.

**Procedury:**

**- add\_ammunition**

CREATE OR REPLACE PROCEDURE add\_ammunition( p\_name VARCHAR, p\_description VARCHAR ) LANGUAGE plpgsql AS $$

BEGIN INSERT INTO ammunition(name, description) VALUES (p\_name, p\_description);

END; $$;

**Opis**: Procedura add\_ammunition służy do dodawania nowej amunicji do bazy danych. Procedura przyjmuje dwa argumenty: nazwę amunicji (p\_name) oraz jej opis (p\_description). Następnie wstawia te dane do tabeli ammunition. Argumenty: p\_name (VARCHAR): Nazwa amunicji, która ma być dodana do bazy danych. p\_description (VARCHAR): Opis amunicji, który ma być dodany do bazy danych. Działanie: Procedura wykonuje operację INSERT, aby dodać nowy rekord do tabeli ammunition z podanymi wartościami nazwy i opisu.

**- add\_museum**

CREATE OR REPLACE PROCEDURE add\_museum( p\_name VARCHAR, p\_localization VARCHAR ) LANGUAGE plpgsql AS $$

BEGIN INSERT INTO museum(name, localization) VALUES (p\_name, p\_localization);

END; $$;

**Opis**: Procedura add\_museum dodaje nowe muzeum do bazy danych. Przyjmuje dwa argumenty: nazwę muzeum i jego lokalizację, które są następnie wstawiane do tabeli museum. Argumenty: p\_name VARCHAR - nazwa muzeum. p\_localization VARCHAR - lokalizacja muzeum. Działanie: Procedura wstawia nowy rekord do tabeli museum z podanymi wartościami dla kolumn name i localization.

**- add\_nation**

CREATE OR REPLACE PROCEDURE add\_nation( p\_nation\_name VARCHAR, p\_prod\_place VARCHAR ) LANGUAGE plpgsql AS $$ BEGIN INSERT INTO nationality(nation\_name, prod\_place) VALUES (p\_nation\_name, p\_prod\_place); END; $$;

**Opis**: Procedura add\_nation służy do dodawania nowej nacji do bazy danych. Procedura przyjmuje dwa argumenty: nazwę nacji (p\_nation\_name) oraz miejsce produkcji (p\_prod\_place). Następnie wstawia te dane do tabeli nationality. Argumenty: p\_nation\_name (VARCHAR): Nazwa nacji, która ma być dodana do bazy danych. p\_prod\_place (VARCHAR): Miejsce produkcji związane z nacją, które ma być dodane do bazy danych. Działanie: Procedura wykonuje operację INSERT, aby dodać nowy rekord do tabeli nationality z podanymi wartościami nazwy nacji i miejsca produkcji.

**Triggery:**

**- check\_input\_trigger\_function**

CREATE OR REPLACE FUNCTION check\_input\_trigger\_function()

RETURNS TRIGGER AS $$

BEGIN IF NEW.nation\_name IS NOT NULL AND NOT check\_no\_digits(NEW.nation\_name::TEXT) THEN RAISE EXCEPTION 'Nazwa nacji nie może zawierać cyfr.'; END IF; IF NEW.prod\_place IS NOT NULL AND NOT check\_no\_digits(NEW.prod\_place::TEXT) THEN RAISE EXCEPTION 'Miejsce produkcji nie może zawierać cyfr.';

END IF; RETURN NEW;

END; $$ LANGUAGE plpgsql;

**Opis**: Funkcja check\_input\_trigger\_function jest funkcją wyzwalającą (trigger function), która służy do walidacji danych wprowadzanych do tabeli nationality. Jej głównym celem jest zapewnienie, że pola nation\_name i prod\_place nie zawierają cyfr. Jeśli którakolwiek z tych kolumn zawiera cyfry, funkcja zgłasza wyjątek i zapobiega wstawieniu lub aktualizacji danych. Działanie: Funkcja sprawdza, czy kolumna nation\_name w nowym rekordzie (NEW) jest różna od NULL i czy nie zawiera cyfr, korzystając z funkcji check\_no\_digits. Jeśli nation\_name zawiera cyfry, funkcja zgłasza wyjątek z komunikatem "Nazwa nacji nie może zawierać cyfr." Podobnie, funkcja sprawdza, czy kolumna prod\_place w nowym rekordzie (NEW) jest różna od NULL i czy nie zawiera cyfr, korzystając z funkcji check\_no\_digits. Jeśli prod\_place zawiera cyfry, funkcja zgłasza wyjątek z komunikatem "Miejsce produkcji nie może zawierać cyfr." Jeśli oba pola są prawidłowe, funkcja zwraca nowy rekord (NEW), pozwalając na wstawienie lub aktualizację danych w tabeli nationality.

**- set\_default\_acc\_type**

CREATE OR REPLACE FUNCTION set\_default\_acc\_type()

RETURNS TRIGGER AS $$

BEGIN NEW.acc\_type := 'user';

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

**Opis**: Funkcja set\_default\_acc\_type jest funkcją wyzwalającą (trigger function), której celem jest automatyczne ustawienie domyślnej wartości kolumny acc\_type na 'user' dla nowych rekordów wstawianych do tabeli. Jest używana w kontekście wyzwalacza, który uruchamia się przed operacją INSERT na określonej tabeli. Działanie: Funkcja przypisuje wartość 'user' do kolumny acc\_type nowego rekordu (NEW). Funkcja zwraca zmodyfikowany nowy rekord (NEW), co pozwala na dalsze przetwarzanie rekordu przez operację INSERT.

**- set\_default\_description**

CREATE OR REPLACE FUNCTION set\_default\_description()

RETURNS TRIGGER AS $$

BEGIN IF NEW.description IS NULL OR NEW.description = '' THEN NEW.description := 'opis zostanie jeszcze dodany';

END IF; RETURN NEW;

END; $$ LANGUAGE plpgsql;

**Opis**: Funkcja set\_default\_description ma na celu automatyczne ustawianie wartości domyślnej dla kolumny description w przypadku, gdy nowo wstawiany rekord nie zawiera wartości dla tej kolumny lub gdy wartość ta jest pusta.

**- set\_default\_localization**

CREATE OR REPLACE FUNCTION set\_default\_localization()

RETURNS TRIGGER AS $$ BEGIN IF NEW.localization IS NULL OR NEW.localization = '' THEN NEW.localization := 'Warszawa';

END IF;

RETURN NEW;

END; $$ LANGUAGE plpgsql;

**Opis:** Funkcja set\_default\_localization ma na celu automatyczne ustawianie wartości domyślnej dla kolumny localization w przypadku, gdy nowo wstawiany rekord nie zawiera wartości dla tej kolumny lub gdy wartość ta jest pusta.

**- set\_negative\_amount\_to\_zero**

CREATE OR REPLACE FUNCTION set\_negative\_amount\_to\_zero()

RETURNS TRIGGER AS $$ BEGIN IF NEW.amount < 0 THEN NEW.amount := 0;

END IF;

RETURN NEW; END; $$ LANGUAGE plpgsql;

**Do tej funkcji utworzone są triggery: przed dodaniem nowego czołgu oraz podczas edycji:**

CREATE TRIGGER set\_negative\_amount\_to\_zero\_before\_insert BEFORE INSERT ON tanks FOR EACH ROW EXECUTE FUNCTION set\_negative\_amount\_to\_zero();

CREATE TRIGGER set\_negative\_amount\_to\_zero\_before\_update BEFORE UPDATE ON tanks FOR EACH ROW EXECUTE FUNCTION set\_negative\_amount\_to\_zero();

**Opis:** Funkcja set\_negative\_amount\_to\_zero ma na celu zapobieganie ustawieniu ujemnej wartości dla kolumny amount w tabeli tanks. Jeśli nowa wartość dla tej kolumny jest ujemna, zostanie automatycznie zmieniona na zero.

**- validate\_user\_update**

CREATE OR REPLACE FUNCTION validate\_user\_update()

RETURNS TRIGGER AS $$

BEGIN IF NEW.user\_email IS NOT NULL AND NOT is\_valid\_email(NEW.user\_email) THEN RAISE EXCEPTION 'Niepoprawny adres e-mail: %', NEW.user\_email;

END IF; IF NEW.user\_pass IS NOT NULL AND NOT is\_valid\_password(NEW.user\_pass) THEN RAISE EXCEPTION 'Hasło musi mieć co najmniej 8 znaków';

END IF; RETURN NEW; END; $$ LANGUAGE plpgsql;

**Do tej funkcji utworzony jest trigger, który wywoła funkcje przed dodaniem nowego użytkownika albo przed jego edycja:**

CREATE TRIGGER check\_input\_trigger BEFORE INSERT OR UPDATE ON nationality FOR EACH ROW EXECUTE FUNCTION check\_input\_trigger\_function();

**Opis**: Funkcja validate\_user\_update ma na celu sprawdzenie poprawności aktualizacji danych użytkownika w tabeli users. Sprawdza, czy nowy adres e-mail jest w prawidłowym formacie oraz czy nowe hasło spełnia minimalne wymagania dotyczące długości.

**- validate\_and\_set\_acc\_type**

CREATE OR REPLACE FUNCTION validate\_and\_set\_acc\_type()

RETURNS TRIGGER AS $$ BEGIN IF NEW.acc\_type IS NULL

OR NOT is\_valid\_acc\_type(NEW.acc\_type)

THEN NEW.acc\_type := 'user';

END IF;

RETURN NEW;

END; $$ LANGUAGE plpgsql;

**Do tej funkcji utworzony jest trigger, który wywoła te funkcje przed każda operacja update**

CREATE TRIGGER validate\_and\_set\_acc\_type\_before\_update

BEFORE UPDATE ON users FOR EACH ROW

EXECUTE FUNCTION validate\_and\_set\_acc\_type()

**Opis:** Ta funkcja validate\_and\_set\_acc\_type służy do walidacji i ustawienia domyślnego typu konta użytkownika w przypadku, gdy podany typ jest nieprawidłowy lub nullowy. Funkcja ta jest często używana w kontekście tworzenia nowych rekordów użytkowników lub aktualizacji istniejących rekordów, aby zapewnić spójność danych w bazie danych.

**Sekwencje są użytkowane przy inkrementowaniu kluczy głównych.**

CREATE SEQUENCE IF NOT EXISTS public.ammunition\_ammo\_id\_seq

INCREMENT 1

START 1

MINVALUE 1

MAXVALUE 2147483647

CACHE 1

OWNED BY ammunition.ammo\_id;

# Źródła:

Ikony pobrane ze strony: <https://icons8.com/>

Grafika muzeum utworzona przez: <https://copilot.microsoft.com/>

Dynamiczna siatka z czołgami: <https://www.youtube.com/watch?v=L3PLDAZWU9s>

Drugi materiał dotyczący siatki: <https://www.youtube.com/watch?v=XlAzQ170kzM>

Wzór: <https://www.youtube.com/watch?v=Aliw3lNRzfc>

Dodawanie zdjęć do bazy: <https://www.tutorialspoint.com/how-to-insert-an-image-in-to-mysql-database-using-java-program>

Również dodawanie: <https://stackoverflow.com/questions/30447539/uploading-image-to-a-mysql-database-using-java>