

Dokumentacja Projektu

Bazy Danych

Jan Pazdan

19 Stycznia 2025

1 Zdefiniowanie tematu projektu

System przeznaczony jest do użytku wewnętrznego placówki i pozwala na przegląd podstawowych informacji o szkole i jej jednostkach organizacji wewnętrznej, dodawanie, usuwanie i aktualizacje danych związanych z jej pracownikami, uczniami i grupami z nich złożonych oraz przegląd harmonogramów i dostępności czasowej ww. grup.

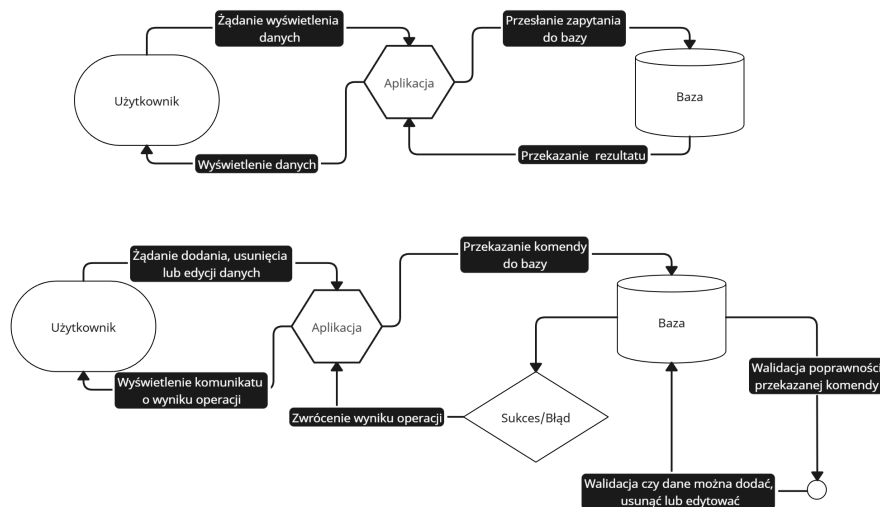
2 Analiza wymagań użytkownika

- Podstawowa - Obserwator (np. Nauczyciel bez praw edycji): Możliwość przeglądania listy uczniów, nauczycieli, sal, uczonych przedmiotów, wychowawstwa, ocen, dziennika klas, nieobecności uczniów oraz rozkłady zajęć klas, nauczycieli i sal.
- Zaawansowana - Edytor (np. Nauczyciel z prawami edycji): Możliwości użytkownika podstawowego wzbogacone o ograniczoną możliwość dodawania, usuwania i zmiany danych wraz z pełnym podglądem do utworzonych relacji.
- Administrator: Funkcjonalność użytkownika zaawansowanego wraz z pełną możliwością wyświetlania, dodawania, usuwania i zmiany danych poprzez dostępny wiersz poleceń.

3 Funkcje aplikacji

- Wyświetlanie dziennika ocen i nieobecności, harmonogram wszystkich jednostek oraz opłaty uczniów
- Podgląd wszystkich klas, przedmiotów, nauczycieli, uczniów i sal
- Dodawanie, usuwanie i edytowanie klas, nauczycieli, sal, uczniów i ich planów lekcji
- Wpisywanie, usuwanie i edytowanie ocen oraz nieobecności

4 Diagram przepływu danych



Rys. 4.1: Diagram przepływu danych

5 Zdefiniowanie encji i atrybutów

- Nauczyciel: ID nauczyciela, imię, nazwisko, lata pracy.
- Sala: ID sali, liczba miejsc, nazwa sali.
- Klasa: ID klasy, numer klasy, litera klasy, ID sali, ID wychowawcy.
- Uczeń: ID ucznia, PESEL, imię, nazwisko, ID klasy, data urodzenia.
- Przedmiot: ID przedmiotu, nazwa przedmiotu.
- Przydział nauczyciela: ID nauczyciela, ID przedmiotu.
- Plan lekcji: ID lekcji, ID przedmiotu, ID klasy, numer lekcji, dzień tygodnia, ID nauczyciela, ID sali.
- Ocena: ID oceny, ID ucznia, ID przedmiotu, data oceny, wartość oceny, waga oceny, komentarz.
- Nieobecność: ID nieobecności, ID ucznia, ID lekcji, data nieobecności.

6 Relacje między encjami

Realizacja 1:1

- Klasa ma tylko jednego wychowawcę a nauczyciel będący wychowawcą ma tylko jedną swoją klasę.
- Klasa ma tylko jedną salę wychowawczą a sala może mieć tylko jedną klasę.

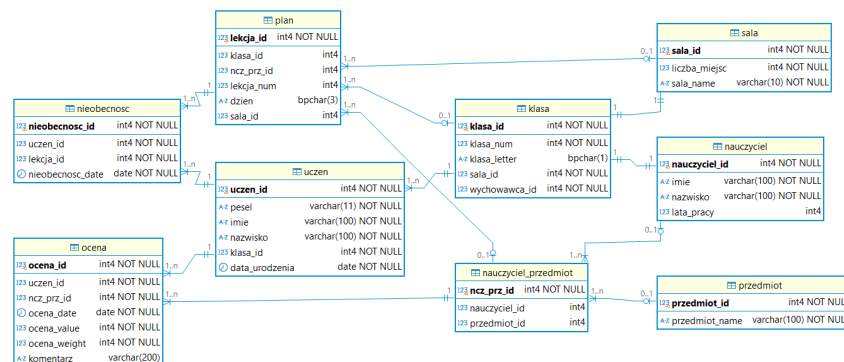
Realizacja 1:N

- Klasa posiada wielu uczniów.
- Klasa może mieć wiele lekcji w ciągu tygodnia.
- W sali może być wiele lekcji w ciągu tygodnia.
- Nauczyciel może mieć wiele lekcji w ciągu tygodnia.
- Uczeń może mieć wiele ocen.

- Uczeń może mieć wiele nieobecności.
- Przedmiot może być uczony wiele razy w ciągu tygodnia.
- Lekcja może mieć wiele nieobecności(nieobecnych uczniów).
- Z każdego przedmiotu może być wystawione wiele ocen.

Realizacja (pośrednia) N:M

- Nauczyciel może uczyć wielu przedmiotów a przedmiot może być uczony przez wielu nauczycieli.
- Na danej lekcji może być nieobecnych wielu uczniów i jeden uczeń może być nieobecny na wielu lekcjach.



Rys. 6.1: Diagram związków encji.

7 Projekt tabel i kluczy

Polecenia SQL tworzące tabele, klucze i ograniczenia znajdują się w pliku *table.sql*

7.1 nauczyciel

- Przechowuje dane o nauczycielach.
- **nauczyciel_id** – klucz główny (PRIMARY KEY), automatycznie generowany.
- **imie**, **nazwisko** – imię i nazwisko nauczyciela.
- **lata_pracy** – liczba lat pracy, może być NULL.
- Ograniczenia (CHECK):
 - **imie** i **nazwisko** – mogą zawierać tylko litery.

7.2 sala

- Zawiera informacje o salach w szkole.
- **sala_id** – klucz główny.
- **liczba_miejsc** – liczba dostępnych miejsc.
- **sala_name** – unikalna nazwa sali.

7.3 klasa

- Przechowuje informacje o klasach szkolnych.
- Każda klasa jest przypisana do konkretnej sali i wychowawcy.
- `klasa_id` – klucz główny.
- `klasa_num` – numer klasy.
- `klasa_letter` – jednoliterowy symbol klasy.
- `sala_id` – klucz obcy wskazujący na salę.
- `wychowawca_id` – klucz obcy wskazujący na nauczyciela-wychowawcę .
- Ograniczenie `unique_num_letter_combination` – gwarantuje, że nie będzie dwóch klas o tej samej kombinacji numeru i litery.

7.4 uczen

- Przechowuje informacje o uczniach i ich przynależności do klas.
- `uczen_id` – klucz główny.
- PESEL – unikalny numer identyfikacyjny ucznia.
- `imie, nazwisko` – dane ucznia.
- `klasa_id` – klucz obcy wskazujący na klasę.
- `data_urodzenia` – data urodzenia ucznia.
- Ograniczenia (CHECK):
 - PESEL – musi mieć dokładnie 11 cyfr.
 - `imie` i `nazwisko` – mogą zawierać tylko litery.

7.5 przedmiot

- Przechowuje listę przedmiotów szkolnych.
- `przedmiot_id` – klucz główny.
- `przedmiot_name` – unikalna nazwa przedmiotu.

7.6 nauczyciel__przedmiot (Powiązanie nauczycieli z przedmiotami)

- Przechowuje relacje wielu do wielu między nauczycielami a przedmiotami.
- `ncz_prz_id` - klucz główny
- `nauczyciel_id` – klucz obcy wskazujący na nauczyciela.
- `przedmiot_id` – klucz obcy wskazujący na przedmiot.
- Ograniczenia (CHECK):
 - `nauczyciel_id` ,`przedmiot_id` – para tych dwóch kluczy obcych musi być unikalna

7.7 plan

- Przechowuje harmonogram zajęć dla każdej klasy.
- `lekcja_id` – klucz główny.
- `ncz_prz_id` – klucz obcy wskazujący na parę nauczyciela, który prowadzi lekcje i przedmiot, który jest prowadzony.
- `klasa_id` – klucz obcy wskazujący na klasę.
- `lekcja_num` – numer lekcji w danym dniu.
- `dzien` – dzień tygodnia .
- `sala_id` – klucz obcy wskazujący na salę.
- Ograniczenia (CHECK):
 - `lekcja_num` – musi być w zakresie 0–8.
 - `dzien` – ograniczony do wartości Pon, Wt, Sr, Czw, Pt.

7.8 ocena

- Przechowuje informacje o ocenach uczniów z poszczególnych przedmiotów.
- `ocena_id` – klucz główny.
- `uczen_id` – klucz obcy wskazujący na ucznia.
- `ncz_prz_id` – klucz obcy wskazujący na parę nauczyciela, który prowadzi lekcje i przedmiot, który jest prowadzony.
- `ocena_date` – data wystawienia oceny.
- `ocena_value` – wartość oceny.
- `ocena_weight` – waga oceny.
- `komentarz` – opcjonalny komentarz nauczyciela.
- Ograniczenia (CHECK):
 - `ocena_value` – musi być w zakresie 1–6.
 - `ocena_weight` – musi być w zakresie 1–5.

7.9 nieobecność

- Przechowuje informacje o nieobecnościach uczniów na lekcjach.
- `nieobecnosc_id` – klucz główny.
- `uczen_id` – klucz obcy wskazujący na ucznia.
- `lekcja_id` – klucz obcy wskazujący na lekcję.
- `nieobecnosc_date` – data nieobecności.

8 Słownik danych

8.1 nauczyciel

- `nauczyciel_id` – INT, klucz główny, generowany automatycznie jako IDENTITY.
- `imie` – VARCHAR(100), imię nauczyciela, nie może być puste.
- `nazwisko` – VARCHAR(100), nazwisko nauczyciela, nie może być puste.
- `lata_pracy` – INT, liczba lat pracy, może być NULL.

8.2 sala

- `sala_id` – INT, klucz główny, generowany automatycznie jako IDENTITY.
- `liczba_miejsc` – INT, liczba miejsc w sali, nie może być puste.
- `sala_name` – VARCHAR(10), unikalna nazwa sali, nie może być pusta.

8.3 klasa

- `klasa_id` – INT, klucz główny, generowany automatycznie jako IDENTITY.
- `klasa_num` – INT, numer klasy, nie może być puste.
- `klasa_letter` – CHAR(1), literowy symbol klasy, może być NULL.
- `sala_id` – INT, klucz obcy wskazujący na salę, nie może być NULL.
- `wychowawca_id` – INT, klucz obcy wskazujący na nauczyciela-wychowawcę, nie może być NULL.

8.4 uczen

- `uczen_id` – INT, klucz główny, generowany automatycznie jako IDENTITY.
- `PESEL` – VARCHAR(11), unikalny numer identyfikacyjny ucznia, nie może być pusty.
- `imie` – VARCHAR(100), imię ucznia, nie może być puste.
- `nazwisko` – VARCHAR(100), nazwisko ucznia, nie może być puste.
- `klasa_id` – INT, klucz obcy wskazujący na klasę, nie może być NULL.
- `data_urodzenia` – DATE, data urodzenia ucznia, nie może być pusta.

8.5 przedmiot

- `przedmiot_id` – INT, klucz główny, generowany automatycznie jako IDENTITY.
- `przedmiot_name` – VARCHAR(100), unikalna nazwa przedmiotu, nie może być pusta.

8.6 nauczyciel_przedmiot (Powiązanie nauczycieli z przedmiotami)

- `ncz_prz_id` – INT, klucz główny, generowany automatycznie jako IDENTITY.
- `nauczyciel_id` – INT, klucz obcy wskazujący na nauczyciela, nie może być NULL.
- `przedmiot_id` – INT, klucz obcy wskazujący na przedmiot, nie może być NULL.

8.7 plan

- `lekcja_id` – INT, klucz główny, generowany automatycznie jako IDENTITY.
- `ncz_prz_id` – INT, klucz obcy, wskazujący na parę nauczyciela i przedmiot.
- `klasa_id` – INT, klucz obcy wskazujący na klasę, nie może być NULL.
- `lekcja_num` – INT, numer lekcji w danym dniu, może być NULL.
- `dzien` – CHAR(3), dzień tygodnia, może być NULL.
- `sala_id` – INT, klucz obcy wskazujący na salę, nie może być NULL.

8.8 ocena

- `ocena_id` – INT, klucz główny, generowany automatycznie jako IDENTITY.
- `uczen_id` – INT, klucz obcy wskazujący na ucznia, nie może być NULL.
- `ncz_prz_id` – INT, klucz obcy, wskazujący na parę nauczyciela i przedmiot.
- `ocena_date` – DATE, data wpisania oceny, nie może być pusta.
- `ocena_value` – INT, wartość oceny (musi być w zakresie 1–6).
- `ocena_weight` – INT, waga oceny (musi być w zakresie 1–5, domyślnie 1).
- `komentarz` – VARCHAR(200), opcjonalny komentarz nauczyciela.

8.9 nieobecnosc

- `nieobecnosc_id` – INT, klucz główny, generowany automatycznie jako IDENTITY.
- `uczen_id` – INT, klucz obcy wskazujący na ucznia, nie może być NULL.
- `lekcja_id` – INT, klucz obcy wskazujący na lekcję, nie może być NULL.
- `nieobecnosc_date` – DATE, data nieobecności, nie może być pusta.

9 Analiza zależności funkcyjnych

9.1 Tabela nauczyciel

- Zależności funkcyjne:
 - $nauczyciel_id \rightarrow imie, nazwisko, lata_pracy$
- Normalizacja:
 - Tabela jest w 1NF, ponieważ wszystkie kolumny zawierają atomowe wartości.
 - Tabela jest w 2NF, ponieważ wszystkie kolumny są w pełni zależne od klucza głównego ($nauczyciel_id$).
 - Tabela jest w 3NF, ponieważ nie ma żadnych zależności przechodnich między kolumnami.

9.2 Tabela sala

- Zależności funkcyjne:
 - $sala_id \rightarrow liczba_miejsc, sala_name$
- Normalizacja:
 - Tabela jest w 1NF, ponieważ wszystkie kolumny zawierają atomowe wartości.
 - Tabela jest w 2NF, ponieważ wszystkie kolumny są w pełni zależne od klucza głównego ($sala_id$).
 - Tabela jest w 3NF, ponieważ nie ma żadnych zależności przechodnich między kolumnami.

9.3 Tabela klasa

- Zależności funkcyjne:
 - $klasa_id \rightarrow klasa_num, klasa_letter, sala_id, wychowawca_id$
 - $klasa_num, klasa_letter \rightarrow klasa_id$
- Normalizacja:
 - Tabela jest w 1NF, ponieważ wszystkie kolumny zawierają atomowe wartości.
 - Tabela jest w 2NF, ponieważ wszystkie kolumny są w pełni zależne od klucza głównego ($klasa_id$).
 - Tabela jest w 3NF, ponieważ nie ma żadnych zależności przechodnich między kolumnami.

9.4 Tabela uczen

- Zależności funkcyjne:
 - $uczen_id \rightarrow PESEL, imie, nazwisko, klasa_id, data_urodzenia$
 - $PESEL \rightarrow imie, nazwisko, klasa_id, data_urodzenia$
- Normalizacja:
 - Tabela jest w 1NF, ponieważ wszystkie kolumny zawierają atomowe wartości.
 - Tabela jest w 2NF, ponieważ wszystkie kolumny są w pełni zależne od klucza głównego ($uczen_id$).
 - Tabela jest w 3NF, ponieważ nie ma żadnych zależności przechodnich między kolumnami.

9.5 Tabela przedmiot

- Zależności funkcyjne:
 - $przedmiot_id \rightarrow przedmiot_name$
- Normalizacja:
 - Tabela jest w 1NF, ponieważ wszystkie kolumny zawierają atomowe wartości.
 - Tabela jest w 2NF, ponieważ wszystkie kolumny są w pełni zależne od klucza głównego ($przedmiot_id$).
 - Tabela jest w 3NF, ponieważ nie ma żadnych zależności przechodnich między kolumnami.

9.6 Tabela nauczyciel__przedmiot

- Zależności funkcyjne:
 - $(nauczyciel_id, przedmiot_id) \rightarrow nauczyciel_id, przedmiot_id$
- Normalizacja:
 - Tabela jest w 1NF, ponieważ wszystkie kolumny zawierają atomowe wartości.
 - Tabela jest w 2NF, ponieważ wszystkie kolumny są w pełni zależne od klucza głównego ($nauczyciel_id, przedmiot_id$).
 - Tabela jest w 3NF, ponieważ nie ma żadnych zależności przechodnich między kolumnami.

9.7 Tabela plan

- Zależności funkcyjne:
 - $lekcja_id \rightarrow ncz_prz_id, klasa_id, lekcja_num, dzien, sala_id$
 - $(przedmiot_id, klasa_id, dzien, lekcja_num) \rightarrow lekcja_id$
- Normalizacja:
 - Tabela jest w 1NF, ponieważ wszystkie kolumny zawierają atomowe wartości.
 - Tabela jest w 2NF, ponieważ wszystkie kolumny są w pełni zależne od klucza głównego ($lekcja_id$).
 - Tabela jest w 3NF, ponieważ nie ma żadnych zależności przechodnich między kolumnami.

9.8 Tabela ocena

- Zależności funkcyjne:
 - $ocena_id \rightarrow uczen_id, ncz_prz_id, ocena_date, ocena_value, ocena_weight, komentarz$
- Normalizacja:
 - Tabela jest w 1NF, ponieważ wszystkie kolumny zawierają atomowe wartości.
 - Tabela jest w 2NF, ponieważ wszystkie kolumny są w pełni zależne od klucza głównego ($ocena_id$).
 - Tabela jest w 3NF, ponieważ nie ma żadnych zależności przechodnich między kolumnami.

9.9 Tabela nieobecnosc

- Zależności funkcyjne:
 - $nieobecnosc_id \rightarrow uczen_id, lekcja_id, nieobecnosc_date$
- Normalizacja:
 - Tabela jest w 1NF, ponieważ wszystkie kolumny zawierają atomowe wartości.
 - Tabela jest w 2NF, ponieważ wszystkie kolumny są w pełni zależne od klucza głównego ($nieobecnosc_id$).
 - Tabela jest w 3NF, ponieważ nie ma żadnych zależności przechodnich między kolumnami.

10 Zaprojektowanie operacji na danych

Polecenia SQL realizujące poniższe funkcje znajdują się w pliku *function.sql*.

10.1 fun_plan_klasy(INT)

- Funkcja generuje plan lekcji dla podanej klasy ($klasa_id$). Zwraca tabelę z harmonogramem zajęć, gdzie każda kolumna reprezentuje dzień tygodnia, a każdy wiersz odpowiada numerowi lekcji. Wypełnia plan lekcji informacją o nazwie przedmiotu i przypisanej sali.

10.2 fun_plan_nauczyciela(INT)

- Funkcja zwraca plan lekcji dla nauczyciela ($nauczyciel_id$). Wskazuje, w której klasie i w jakiej sali nauczyciel prowadzi zajęcia w danym dniu i godzinie lekcyjnej.

10.3 fun_plan_sali(INT)

- Funkcja zwraca plan zajęć dla danej sali ($sala_id$). Pokazuje, które klasy mają lekcje w danej sali oraz który nauczyciel prowadzi zajęcia.

10.4 fun_sklad_klasy(INT)

- Funkcja zwraca listę uczniów zapisanych do konkretnej klasy (**klasa_id**). Każdy uczeń ma przypisany numer w dzienniku, nazwisko i imię.

10.5 fun_uczen_info(INT)

- Funkcja zwraca informacje o uczniu (**uczen_id**), w tym jego nazwisko, imię, klasę oraz wychowawcę. Jeśli jako parametr podano 0, zwraca listę wszystkich uczniów.

10.6 fun_dziennik(INT,INT)

- Funkcja zwraca dziennik ocen dla konkretnej klasy (pierwszy argument - **klasa_id**) z konkretnego przedmiotu (drugi argument - **przedmiot_id**). Zawiera nazwiska i imiona uczniów oraz listę ich ocen w formie tekstowej (zagregowane oceny względem daty wystawienia) oraz średnią ocen.

10.7 fun_nieobecnosc(INT)

- Funkcja zwraca listę nieobecności dla konkretnego ucznia (**uczen_id**). Uwzględnia numer lekcji, dzień tygodnia, przedmiot oraz datę nieobecności.

10.8 fun_oceny(INT)

- Funkcja zwraca szczegółowe informacje o ocenach ucznia (**uczen_id**). Uwzględnia przedmiot, nauczyciela wystawiającego ocenę, wartość oceny, wagę, datę oraz ewentualny komentarz.

10.9 fun_nauczyciel_info()

- Funkcja zwraca listę nauczycieli wraz z informacją, czy pełnią rolę wychowawcy klasy.

10.10 fun_klasa_info()

- Funkcja zwraca informacje o klasach, takie jak liczba uczniów, wychowawca i przypisana sala wychowawcza.

10.11 fun_sala_info()

- Funkcja zwraca listę sal lekcyjnych, ich pojemność oraz informację, czy są przypisane do konkretnej klasy jako sala wychowawcza.

10.12 fun_przedmiot_info()

- Funkcja zwraca listę nauczanych przedmiotów oraz nazwiska nauczycieli prowadzących zajęcia.

10.13 licz_srednia(INT,INT)

- Funkcja zwraca średnią ocen ucznia (pierwszy argument - **uczen_id**) z danego przedmiotu (drugi argument - **przedmiot_id**).

11 Interfejs do obsługi aplikacji i bazy

Polecenia w języku SQL realizujące poniższe widoki znajdują się w pliku *view.sql*.

Widoki bazy danych

W systemie bazy danych dla szkoły zostały utworzone widoki, które umożliwiają wygodne pobieranie informacji dotyczących nauczycieli, klas, sal lekcyjnych oraz przedmiotów. Widoki te służą jako interfejsy dla funkcji zwracających ogólne informacje bez konieczności operowania na surowych danych tabelarycznych. Poniżej znajduje się ich opis:

- **nauczyciel_info** – widok zawierający listę nauczycieli wraz z informacją, czy pełnią funkcję wychowawcy klasy.
- **klasa_info** – widok prezentujący dane dotyczące klas, w tym liczbę uczniów, wychowawcę oraz przypisaną salę wychowawczą.
- **sala_info** – widok przedstawiający listę sal lekcyjnych wraz z ich pojemnością oraz informacją, czy są przypisane do konkretnej klasy jako sala wychowawcza.
- **przedmiot_info** – widok dostarczający informacji o nauczanych przedmiotach oraz nauczycielach je prowadzących.
- **klasa_przedmiot_info** – widok zawiera informacje nauczycielach uczących daną klasę oraz jakiego przedmiotu.

Ze względu na potrzebę ujednolicenia formy pobieranych danych, powyższe widoki oprócz **klasa_przedmiot_info** są realizowane przez funkcje bezargumentowe o tej samej nazwie poprzedzone członem *fun_*. Poza zostały funkcje realizują podobne zadania jednak wymagają uszczegółowienia o kim lub o czym należy przekazać dane.

12 Weryfikacja wprowadzonych danych

Polecenia w języku SQL realizujące poniższe wyzwalacze znajdują się w pliku *trigger.sql*.

12.1 trigger_plan_klasy

- **Zdarzenie wywołujące:** Wyzwalany przed operacją INSERT lub UPDATE na tabeli **plan**.
- **Opis działania:**
 - Trigger sprawdza, czy nie istnieje już zajęcia w tej samej sali, z tym samym nauczycielem i w tym samym czasie.
 - Jeżeli taki konflikt istnieje, zgłaszany jest wyjątek z informacjami o istniejących zajęciach.
 - W przeciwnym razie rekord jest dodawany lub aktualizowany.

12.2 trigger_plan_sali

- **Zdarzenie wywołujące:** Wyzwalany przed operacją INSERT lub UPDATE na tabeli **plan**.
- **Opis działania:**
 - Trigger sprawdza, czy sala jest już zajęta w określonym dniu i godzinie.
 - W przypadku konfliktu, zgłaszany jest wyjątek z informacjami o zajętych salach.
 - Jeżeli nie ma konfliktu, zapis jest dodawany lub aktualizowany.

12.3 trigger_plan_nauczyciela

- **Zdarzenie wywołujące:** Wyzwalany przed operacją INSERT lub UPDATE na tabeli **plan**.
- **Opis działania:**
 - Trigger sprawdza, czy nauczyciel ma już zaplanowane zajęcia w danym czasie.
 - Jeżeli nauczyciel jest już zajęty, zgłaszany jest wyjątek z informacjami o konfliktujących zajęciach.
 - W przeciwnym razie, rekord jest dodawany lub aktualizowany.

12.4 trigger_enforce_uppercase_klasa_letter

- **Zdarzenie wywołujące:** Wyzwalany przed operacją INSERT lub UPDATE na tabeli `klasa`.
- **Opis działania:**
 - Trigger przekształca wartość kolumny `klasa_letter` na wielką literę przed zapisaniem jej w tabeli.
 - Pozwala to na zachowanie jednolitego formatu danych.

12.5 trigger_ocena

- **Zdarzenie wywołujące:** Wyzwalany przed operacją INSERT lub UPDATE na tabeli `ocena`.
- **Opis działania:**
 - Trigger sprawdza czy istnieje identyfikator pary (nauczyciel, przedmiot) wśród tych, których jest uczony dany uczeń korzystając z widoku `klasa_przedmiot_info`
 - Jeżeli dana para nie zawiera się w widoku dla danego ucznia to zgłaszany jest wyjątek.
 - W przeciwnym razie, rekord jest dodawany lub aktualizowany.

12.6 trigger_sprawdz_nieobecnosc

- **Zdarzenie wywołujące:** Wyzwalany przed operacją INSERT lub UPDATE na tabeli `ocena`.
- **Opis działania:**
 - Trigger sprawdza czy data wpisywana do nieobecności jest poprawna, tzn. czy w danym dniu tygodnia znajdowała się dana lekcja.
 - Jeżeli data jest nieprawidłowa to zgłaszany jest wyjątek.
 - W przeciwnym razie, rekord jest dodawany lub aktualizowany.

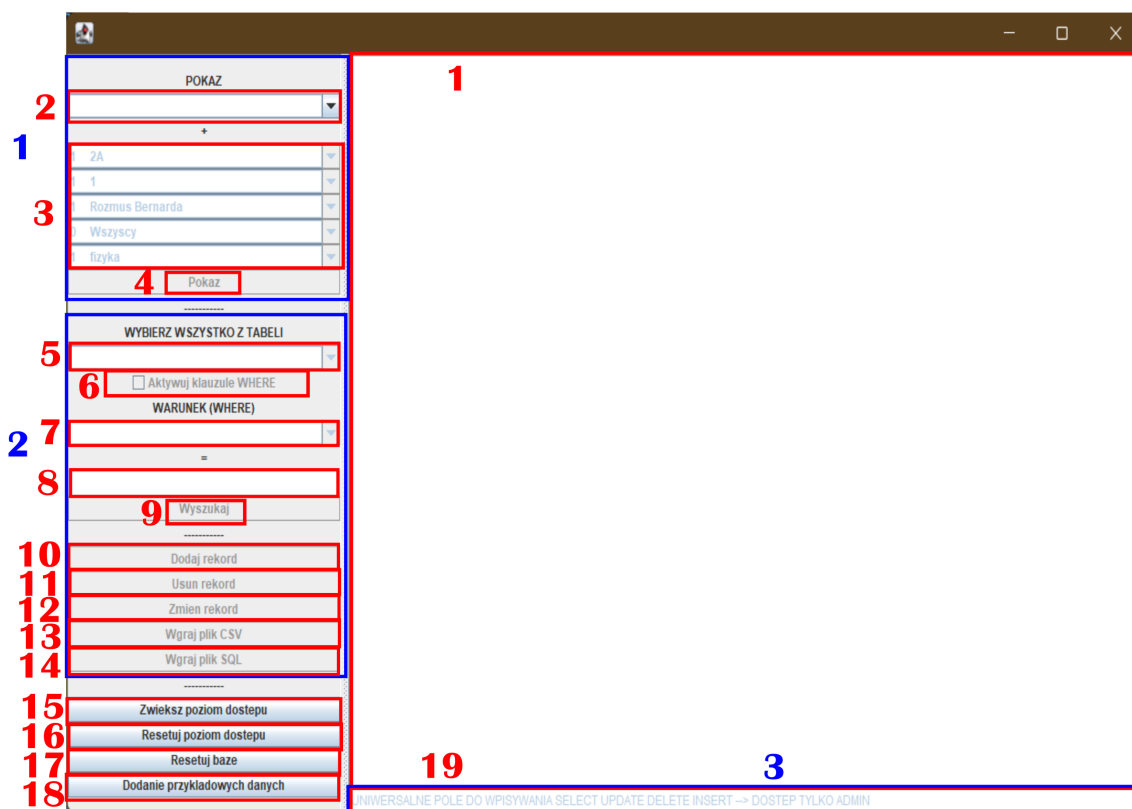
12.7 trigger_sprawdz_ucznia_lekcje

- **Zdarzenie wywołujące:** Wyzwalany przed operacją INSERT lub UPDATE na tabeli `ocena`.
- **Opis działania:**
 - Trigger sprawdza czy lekcja wpisywana do nieobecności istnieje w planie lekcji danego ucznia.

13 Wizualizacja danych

- Wyświetlane raporty i odpowiedzi bazy są w postaci podstawowych zmiennych znakowych typu *String*. W przypadku poprawnego zapytania wyświetla się tabela złożona z przynajmniej jednej kolumny oraz przynajmniej jednego wiersza. Część zapytań domyślnie realizuje tylko jedno wierszowe zapytanie inne zaś tylko wielowierszowe.
- W przypadku poprawnej realizacji edycji tabel (dodawanie, usuwanie, edytowanie rekordów) wyświetlany jest komunikat o nowej lub byłej zawartości rekordu, który brał udział w tej procedurze.
- W przypadku błędnej realizacji zapytania lub edycji wyświetlany jest komunikat o tym co spowodowało błąd, w przypadku braku obsługi danego błędu wyświetli się prosty komunikat o sugestii zmiany wpisanego zapytania lub pola edycji.

14 Panel sterowania aplikacji oraz Makropolecenia



Rys. 14.1: Panel użytkownika

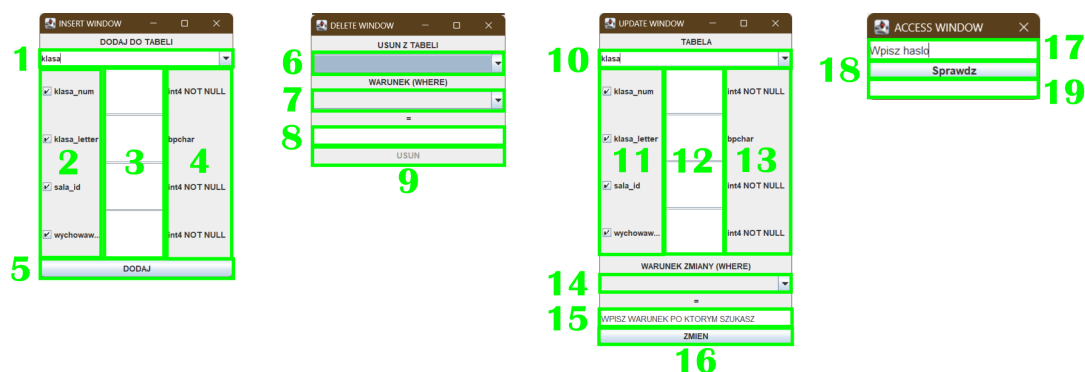
Niebieskie - każdy z zaznaczonych obszarów realizowany jest przez różny poziom dostępu zmieniający za pomocą przycisków (Czerwone) **nr 13** i **nr 14**.

1. Poziom dostępu podstawowy - Obserwator
2. Poziom dostępu zaawansowany - Edytor
3. Poziom dostępu Administrator

Czerwone:

1. Główny panel - element podający informację zwrotną na temat wykonywanej akcji.
2. Rozwijane menu funkcji - zawiera listę poleceń do wykonania.
3. Rozwijane menu argumentów - w przypadku gdy polecenie z menu **nr 2** wymaga argumentu, odpowiednie menu będzie dostępne i będzie możliwy wybór argumentu.
4. Przycisk *Pokaż* - realizuje zapytanie wybrane w menu **nr 2** i **nr 3**.
5. Rozwijane menu tabel - zawiera listę tabel w bazie danych.
6. Pole wyboru klauzuli warunku - po zaznaczeniu menu **nr 7**, pole tekstowe **nr 8** oraz przycisk **nr 9** zostają aktywowane.
7. Rozwijane menu atrybutów - zawiera listę atrybutów tabeli wybranej w menu **nr 5**.
8. Pole tekstowe warunku - wpisanie jest równoznaczne ze sprawdzeniem czy istnieje atrybut, który jest równy wartości wpisanej w pole tekstowe.
9. Przycisk *Wyszukaj* - realizuje zapytanie i przekazuje na główny panel rekordy z wybranej tabeli w menu **nr 5** i opcjonalnie realizuje warunek z menu **nr 7** i pola tekstowego **nr 8**.
10. Przycisk *Dodaj rekord* - wyświetla nowe okno, które realizuje opcje dodania rekordu do bazy.

11. Przycisk *Usuń rekord* - wyświetla nowe okno, które realizuje opcje usunięcia rekordu lub rekordów z bazy.
12. Przycisk *Dodaj rekord* - wyświetla nowe okno, które realizuje opcje edycji rekordu lub rekordów w bazie.
13. Przycisk *Wgraj plik CSV* - wyświetla okno systemowego przeglądania plików oraz możliwość dokonania wyboru. Po dokonaniu wyboru pojawi się komunikat.
14. Przycisk *Wgraj plik SQL* - wyświetla okno systemowego przeglądania plików oraz możliwość dokonania wyboru. Po dokonaniu wyboru pojawi się komunikat.
15. Przycisk *Zwiększ poziom dostępu* - wyświetla nowe okno, w którym jest możliwość zwiększenia poziomu dostępu.
16. Przycisk *Resetuj poziom dostępu* - resetuje poziom dostępu do podstawowego.
17. Przycisk *Resetuj bazę* - resetuje bazę do danych początkowych. Wszystkie akcje modyfikujące bazę są cofane i baza zostaje przywrócona do stanu początkowego.
18. Przycisk *Dodanie przykładowych danych* - funkcjonalność realizowana jednokrotnie na każdy reset bazy. Dodaje nie proceduralne wcześniej utworzone przykładowe dane do bazy, w celu nagłego zwiększenia ilości danych. W przypadku ponownego użycia wyświetli się komunikat o błędzie.
19. Wiersz poleceń - realizuje dowolne zapytanie wyszukujące, dodające, usuwające i edytujące dane. Po wpisaniu komendy należy wcisnąć klawisz *ENTER*.



Rys. 14.2: Interfejs pojawiających się okien. Kolejno od lewej okno dodawania rekordu, okno usuwania rekordu, okno edycji rekordu oraz okno zwiększenia dostępu w aplikacji.

Zielone - interfejs pojawiających się okien:

- Okno dodawania rekordu - podczas gdy jest otwarte, główne okno pozostaje aktywne.
 1. Rozwijane menu tabel - zob. **Czerwone nr 5**.
 2. Pola atrybutów - każdy wiersz obszaru nr 2 jest osobnym atrybutem wybranej tabeli w menu **nr 1**, który po zaznaczeniu jest uwzględniany przy tworzeniu komendy dodawania.
 3. Pola tekstowe atrybutów - każde z pól jest miejscem do wprowadzania danych, które użytkownik chce dodać jako atrybut rekordu.
 4. Informacja o atrybutach - ten nie interaktywny obszar zawiera informacje o tym jakiego typu jest dany atrybut i czy może być pusty tzn. nie zaznaczony w polu **nr 2**.
 5. Przycisk *DODAJ* - przycisk finalizujący dodawanie rekordu. Niezależnie do powodzenia operacji dodawania okno nadal pozostaje otwarte.
- Okno usuwania rekordu - podczas gdy jest otwarte, główne okno pozostaje aktywne.
 6. Rozwijane menu tabel - zob. **Czerwone nr 5**.
 7. Rozwijane menu atrybutów - zob. **Czerwone nr 7**.

- 8. Pole tekstowe warunku - zob. **Czerwone nr 8**.
- 9. Przycisk *USUŃ* - przycisk finalizujący usuwanie rekordu. Niezależnie do powodzenia operacji usuwania okno nadal pozostaje otwarte.
- Okno edycji rekordu - podczas gdy jest otwarte, główne okno pozostaje aktywne.
- 10. Rozwijane menu tabel - zob. **Czerwone nr 5**.
- 11. Pola atrybutów - zob. **Zielone nr 2**. Zaznaczone pola są uwzględniane przy tworzeniu komendy edycji.
- 12. Pola tekstowe atrybutów - zob. **Zielone nr 3**.
- 13. Informacja o atrybutach - zob. **Zielone nr 4**.
- 14. Rozwijane menu atrybutów - zob. **Czerwone nr 7**.
- 15. Pole tekstowe warunku - zob. **Czerwone nr 8**.
- 16. Przycisk *ZMIEN* - przycisk finalizujący edycję rekordu. Niezależnie do powodzenia operacji edytowania okno nadal pozostaje otwarte.
- Okno zwiększenia dostępu do aplikacji - podczas gdy jest otwarte, główne okno jest nieaktywne.
- 17. Pole tekstowe hasła - pole tekstowe przeznaczone do wpisania hasła.
- 18. Przycisk *Sprawdź* - przycisk sprawdzający czy wpisane hasło jest poprawne. W przypadku poprawnego hasła okno znika.
- 19. Informacyjne pole tekstowe - w przypadku podania złego hasła w tym polu pojawia się komunikat.

15 Wprowadzanie danych

Wprowadzanie danych jest możliwe poprzez wgranie pliku CSV lub SQL oraz poprzez ręczne ich wpisanie. Plik CSV oraz SQL nie muszą posiadać ich odpowiedniego rozszerzenia, wymagania są co do zawartości. Plik SQL powinien zawierać podstawowe komendy do wprowadzania lub edycji danych w języku PSQL. Plik typu CSV w pierwszym wierszu powinien zawierać nazwę tabeli, do której użytkownik chce dodać dane. Drugi wiersz zawiera nazwy kolumn rozdzielone przecinkiem. Reszta wierszy powinna zawierać dane, które użytkownik ma zamiar dodać, jednakże kolejność odpowiedniej danej powinna odpowiadać kolejności wpisanych kolumn.

16 Dokumentacja użytkownika

Przykładowe ścieżki działań dla każdego z rodzajów użytkowników:

- Użytkownik podstawowy:
 1. Wybranie odpowiedniej procedury *pokazującej* z menu **nr 2** w głównym oknie.
 2. Zależnie od procedury, należy wybrać z odpowiadającego jej menu argumentów (**nr 3** w głównym oknie).
 3. Wciśnięcie przycisku **nr 4** w głównym oknie.
- Użytkownik zaawansowany - poprawnie przeszedł formę weryfikacji.
 1. Wciśnięcie przycisku **nr 10** w głównym panelu.
 2. Wybranie z nowo wyświetlonego okna odpowiedniej tabeli z menu **nr 1** z elementów interfejsu wyskakujących okien.
 3. Zaznaczenie odpowiednich pól, które użytkownik chce aby zawierały się w nowym rekordzie.
 4. Wpisanie wartości odpowiadających typom w aktywne pola tekstowe.
 5. Wciśnięcie przycisku *DODAJ*.

lub

1. Wybranie tabeli z menu **nr 5** w głównym oknie.
2. Zaznaczenie lub odznaczenie aktywacji klauzuli warunku *WHERE*.
3. (W przypadku zaznaczenia) Należy wybrać, z menu **nr 7** w głównym oknie, atrybut względem którego ma być rozpatrzony warunek.
4. (W przypadku zaznaczenia) Należy wpisać w pole tekstowe **nr 8** wartość atrybutu, który użytkownik uważa za słuszny w kontekście przeszukiwanej tabeli.
5. Wcisnąć przycisk **nr 9**, aby wyszukać wszystkie lub odpowiednie rekordy.

lub

1. Wciśnięcie przycisku **nr 13** lub **nr 14**.
 2. Wyświetli się okno wyboru pliku, należy wybrać odpowiedni mający logiczny sens w kontekście działania programu.
- Administrator - najwyższy stopień dostępu.
 1. Najeżdżenie myszką na aktywny wiersz poleceń, w którym użytkownik może wpisać dowolne polecenie dodające, usuwające, edytujące lub wyświetlające zgodnie z składnią języka PSQL.
 2. Po wpisaniu komendy, wcisnąć klawisz *ENTER*.

17 Opracowanie dokumentacji technicznej

Hierarchia plików:

- *CSV.csv* - przykładowy plik do sprawdzenia funkcjonalności dodawania CSV.
- *SQL.sql* - przykładowy plik do sprawdzenia funkcjonalności dodawania SQL.
- *exec*
 - *szkola_BD.jar*
 - *src* - wewnętrzne pliki potrzebne do resetu danych.
- *database*
 - *all.sql*
 - *function.sql*
 - *table.sql*
 - *trigger.sql*
 - *view.sql*
 - *insert0.sql*
 - *insert1.sql*
- *dokumentacja*
 - *DOKUMENTACJA.pdf* - ten plik
 - *function.sql*
 - *table.sql*
 - *trigger.sql*
 - *view.sql*
 - *Java_doc*
 - * ...
 - * *index.html* - dokumentacja aplikacji
 - * ...
- *app_src*
 - ... - pliki źródłowe aplikacji w kodzie *Java*.

18 Literatura

Dokumentacja biblioteki Java Swing

- <https://docs.oracle.com/en/java/javase/17/docs/api/java.desktop/javax.swing/package-summary.html>

Dokumentacja biblioteki Java JDBC

- <https://docs.oracle.com/javase/8/docs/api/java/sql/package-summary.html>