POLITECHNIKA WROCŁAWSKA

WYDZIAŁ ELEKTRONIKI

PROJEKT Z BAZ DANYCH

**Dziennik szkolny**

Termin zajęć: Poniedziałek, 11:15

Autor/Autorzy: Prowadzący zajęcia:

Dariusz Tomaszewski 235565 dr inż. Roman Ptak, W4/K9

Alicja Wróbel 238894

E-mail:

235565@student.pwr.edu.pl

238894@student.pwr.edu.pl

Wrocław, 2018 r.

# Wstęp

## Cel projektu

Celem projektu jest baza danych szkolnego dziennika elektronicznego, która będzie wykorzystywana w Szkole Podstawowej w Pcimiu. Ma na celu usprawnić zarządzanie dokumentacją szkolną oraz optymalizacje pracy z dużą ilością danych, co pozwala na zwiększenie efektywności pracowników placówki oraz odciążenie ich od zarządzania dokumentacją.

## Zakres projektu

System ma umożliwiać organizację zajęć szkolnych oraz nadzorowanie postępów ucznia w nauce przez nauczycieli oraz rodziców lub opiekunów prawnych. Zakres projektu będzie obejmował obsługę aplikacji dla nauczycieli i wychowawców.

# Analiza wymagań

## Opis działania i schemat logiczny systemu

W systemie zapisane są informacje o uczniach, klasach, pracownikach, realizowanych przedmiotach. Istnieje możliwość wpisywania ocen oraz nieobecności (ewentualnych usprawiedliwień) uczniów. Baza zawiera listę przedmiotów nauczanych w szkole wraz z jej opisem. Wykorzystywane do spisu zajęć odbywających się w określonym czasie, dniu tygodnia, sali oraz informacja dla której konkretnej klasy mają się odbyć. Dzięki temu jest możliwy widok planu lekcji. Dany przedmiot może odbywać się więcej niż jeden w tym samym czasie lecz w innej sali ponieważ może być prowadzony przez innego nauczyciela. Każdy nauczyciel ma możliwość nauczania więcej niż jednego przedmiotu. System ma również na celu przechowywania informacji o klasach uczących się w szkole. Każda z nich ma przypisanego wychowawcę, rok rozpoczęcia edukacji klasy oraz planowanego ukończenia szkoły. Baza oprócz przechowywania danych osobowych uczniów, będzie zawierać informacje o przynależności ucznia do klasy oraz postępów w nauce. Każdy z uczniów posiada swoją listę ocen oraz nieobecności. Informacje   
o ocenach zawierają datę jej wystawienia, wagę oraz przedmiot, z którego została uzyskana.  
Dzięki temu można uczniowie mają indywidualne dzienniczki. Co więcej lista nieobecności ma datę wystawienia i możliwość jej usprawiedliwienia. System ma również na celu zautomatyzowanie czynności takich jak obliczanie średnich ocen dla konkretnych uczniów jak   
i dla konkretnych klas, automatyczne generowanie danych gotowych do wystawienia świadectwa szkolnego.

## Wymagania funkcjonalne

W zakresie funkcjonalności przeznaczonej dla uczniów

* Sprawdzenie ocen cząstkowych, śródrocznych i końcowych
* Sprawdzenie listy uwag/pochwał
* Sprawdzenie frekwencji
* Sprawdzenie aktualnego planu lekcji
* Wysyłanie/odbieranie komunikatów nauczyciela/wychowawcy

W zakresie funkcjonalności przeznaczonej dla opiekunów:

* Sprawdzenie ocen cząstkowych, śródrocznych i końcowych
* Sprawdzenie listy uwag/pochwał
* Sprawdzenie frekwencji
* Sprawdzenie aktualnego planu lekcji
* Wysyłanie/odbieranie komunikatów nauczyciela/wychowawcy
* Możliwość usprawiedliwienia nieobecności podopiecznego

W zakresie funkcjonalności przeznaczonej dla nauczycieli:

* Możliwość wprowadzania ocen uczniów
* Możliwość wprowadzania uwag i pochwał
* Możliwość rejestrowania frekwencji we wszystkich klasach
* Możliwość wpisania tematu lekcji

W zakresie funkcjonalności przeznaczonej dla wychowawców:

* Dostęp do wszystkich danych swoich wychowanków z możliwością modyfikacji
* Możliwość wydrukowania wyników nauczania ucznia
* Możliwość generowania świadectw szkolnych
* Możliwość wykonania zestawień statystycznych dotyczących wyników nauczania   
  i frekwencji
* Możliwość wysyłania komunikatów do uczniów/opiekunów zarówno pojedynczo jak i do całego oddziału
* Szczegółowa analiza wyników nauczania swoich uczniów (wszystkie dane i zestawienia)
* Możliwość wprowadzania ocen z zachowania

W zakresie funkcjonalności przeznaczonej dla dyrektorów i administratorów:

* Dostęp do wszystkich danych uczniów i możliwość ich modyfikacji
* Analiza uwag wpisanych wszystkim uczniom
* Analiza wyników nauczania, w szczególności ocen końcowych (jak stawiają stopnie nauczyciele, jak wyglądają stopnie z poszczególnych przedmiotów)
* Analiza frekwencji

## Wymagania niefunkcjonalne

### Wykorzystywane technologie i narzędzia

Jako system zarządzania bazą danych zostanie wykorzystany MySQL, natomiast oprogramowanie końcowe zostanie wykonane w technologii Java. Aplikacja zostanie wykonana w wersji desktopowej na system operacyjny Windows.

### Wymagania dotyczące rozmiaru bazy danych

Rozmiar bazy będzie zależał od ilości uczniów w szkole, których przewidywanych jest 300. Do każdego dziecka przypisany jest opiekun. Różnych klas jest 15, przedmiotów 10, nauczycieli 20. W przeciągu semestru uczeń otrzyma ok. 80 ocen, co daje 24000 ocen   
w bazie dla wszystkich uczniów. W przeciągu tygodnia odbędzie się 525 lekcji w całej szkole.

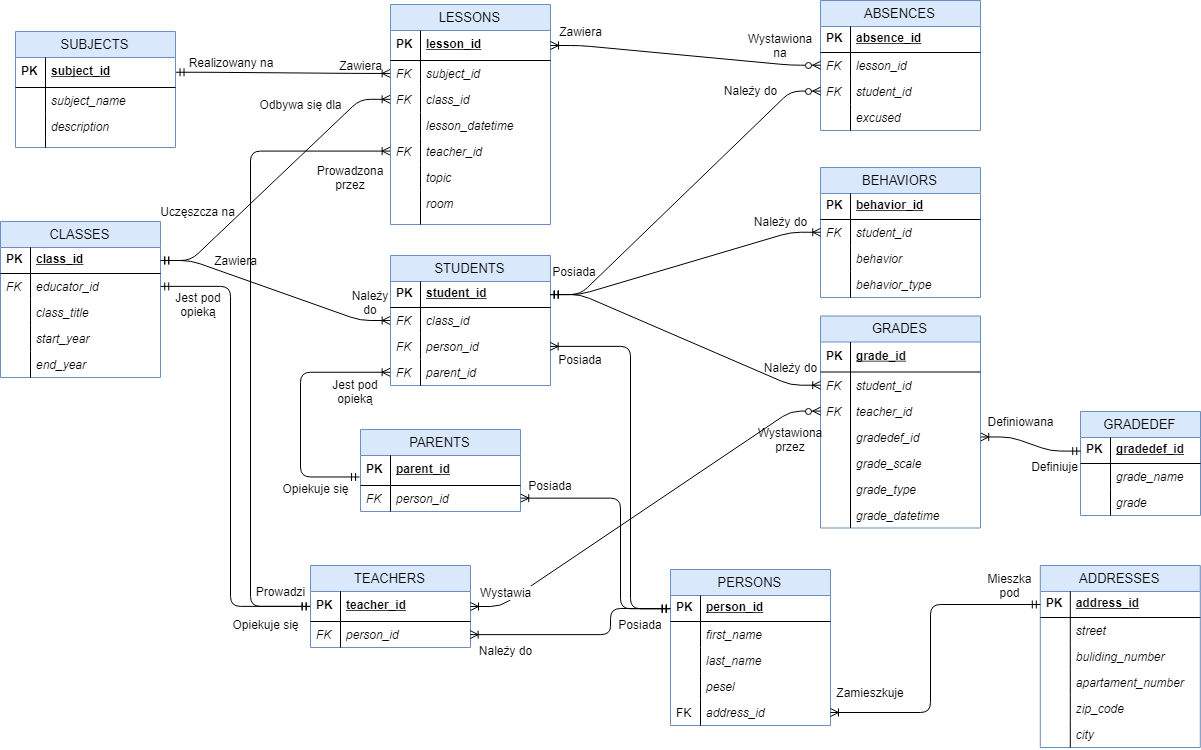
### Wymagania dotyczące bezpieczeństwa systemu

Pracownicy jak i administrator systemu biorą odpowiedzialność za przechowywanie   
i bezpieczeństwo danych. Do aplikacji będzie wykorzystany proces uwierzytelniania przy logowaniu oraz konieczność podania hasła. Przy wpisywaniu ocen końcowych zostaną wykorzystane tokeny pozwalające zatwierdzić wpisaną ocenę.

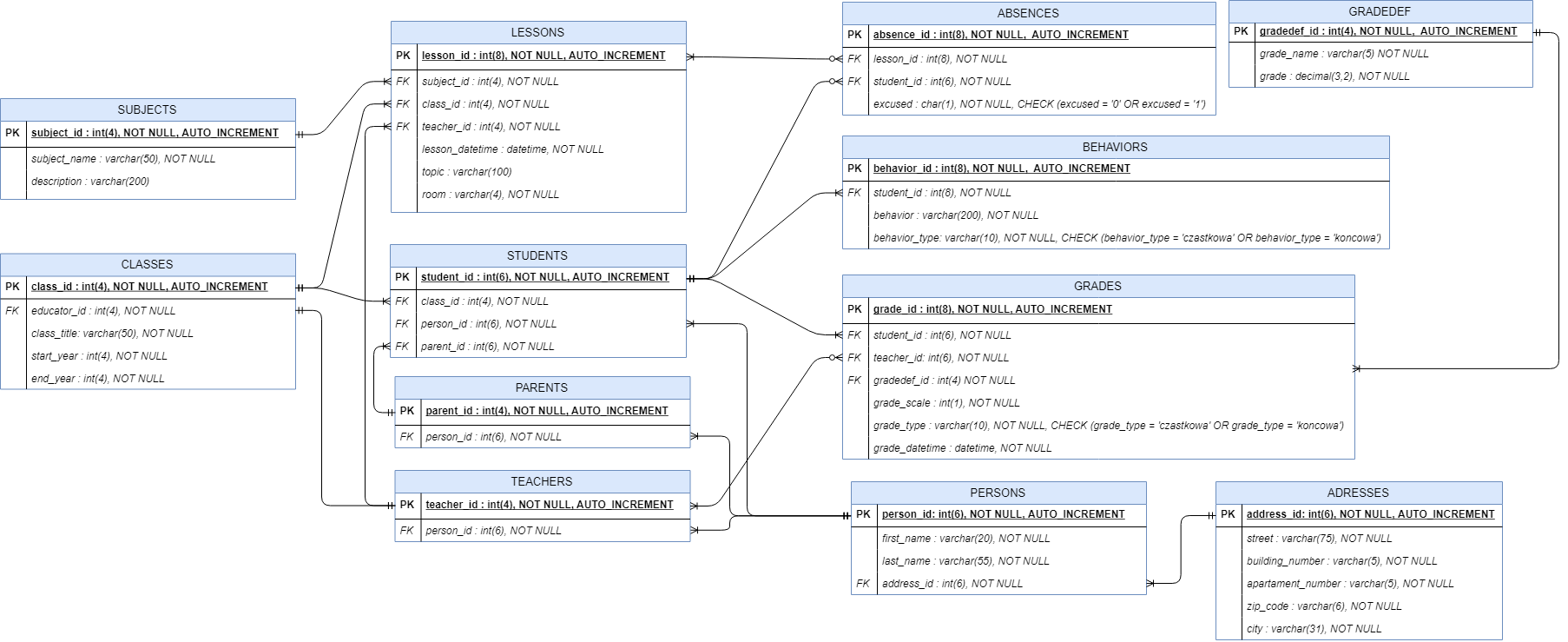
## Projekt systemu

### Analiza rzeczywistości, model logiczny i konceptualny

Szkoła w Pcimiu ma udostępniać uczniom ich własne postępy w nauce oraz kontrole nad nimi przez opiekunów. Nauczyciele i wychowawcy wprowadzają dane postępów podopiecznych, tematu lekcji oraz komunikatów. W systemie po uwierzytelnieniu zostaną wyświetlone funkcje dostępne dla danego użytkownika. W bazie będą znajdować się wszystkie dane dotyczące uczniów, ocen, frekwencji, uwag, planów zajęć, tematów. Dostęp do wglądu i modyfikacji poszczególnych danych zależy od udostępnionych funkcji dla użytkownika.



### Model fizyczny i ograniczenia integralności danych



## Inne elementy schematu – mechanizmy przetwarzania danych

Triggery

Triggery zostaną utworzone w odniesieniu do trzech tabel ABSENCES, GRADES oraz BEHAVIORS.   
Do każdej z nich zostanie przypisany trigger aktywowany po modyfikacji tabeli, który będzie zawierał informacje o typie zdarzenia, dacie oraz osobie modyfikującej.

* T\_GRADE\_LOGS – trigger wyzwalany przy dodawaniu/edycji/usunięciu wiersza w tabeli GRADES.
* T\_ABSENCES\_LOGS – trigger wyzwalany przy dodawaniu/edycji/usunięciu wiersza w tabeli ABSENCES
* T\_BEHAVIORS\_LOGS – trigger wyzwalany przy dodawaniu/edycji/usunięciu wiersza w tabeli ABSENCES

Widoki

Widoki będą utworzone na podstawie łączenia różnych tabel, poniżej w nawiasach zostały zapisane dokładnie jakich.

TIMETABLE – widok zawierający siatkę zajęć dla danej klasy(CLASSES), o konkretnej godzinie, dniu tygodnia(LESSONS) i z konkretnym nauczycielem(TEACHERS).

STUDENTS ABSENCES – lista nieobecności(ABSENCES) uczniów(STUDENTS) danej klasy(CLASSES).

REPORT CARD – świadectwo z uzyskanymi dotychczas średnimi ocen(GRADES) z danych przedmiotów(LESSONS), zachowaniem(BEHAVIORS) oraz dane osobowe ucznia(STUDENTS).

W celu uzyskania dokładnych danych nauczycieli oraz uczniów tabele te zostaną połączone   
z tabelą PERSONS.

Indeksy

Zostanie utworzony indeks do tabeli STUDENTS na kolumnach first\_name oraz last\_name pobranych z PERSONS.

## Bezpieczeństwo na poziomie baz danych

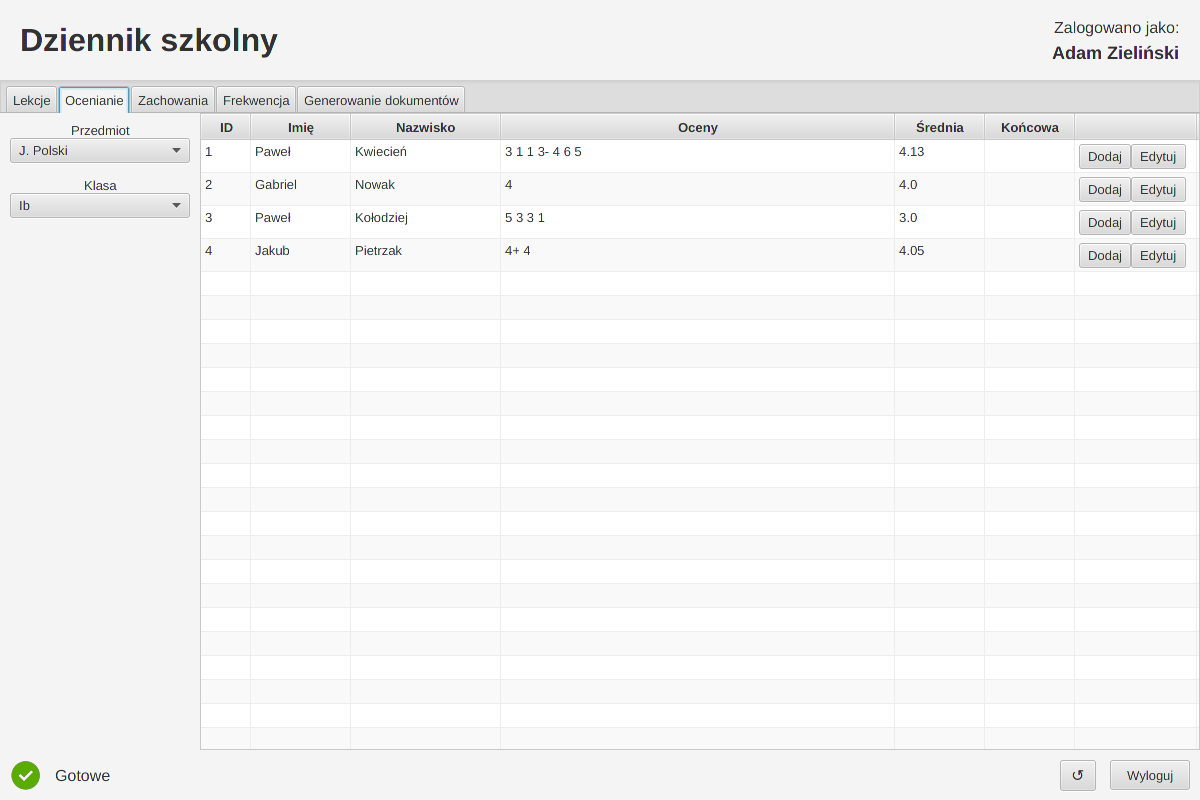
Podział ze względu na dostęp do danych. Tabele, które nie zostały wymienione są wyłącznie do odczytu dla poniżej wymienionych użytkowników.

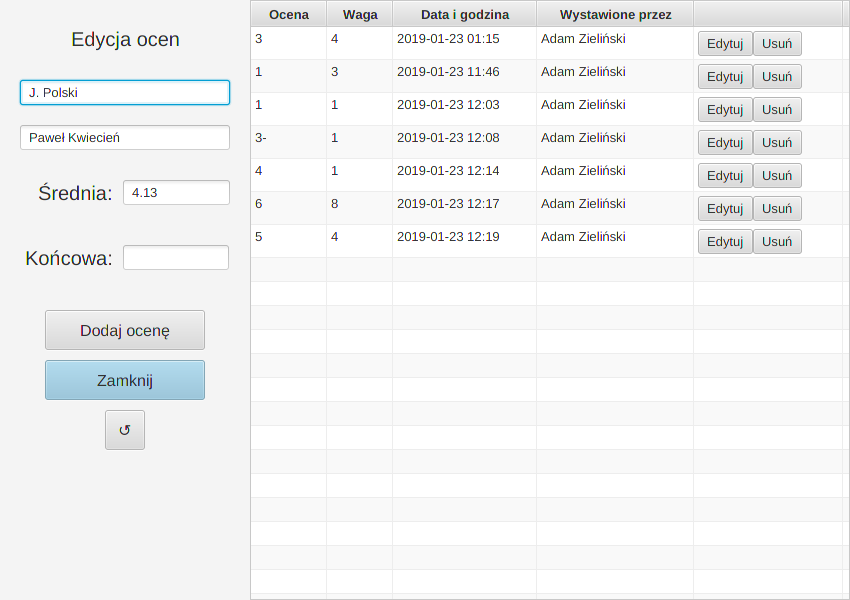
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| WYCHOWACA | | | NAUCZYCIEL | | |
| ADD | UPADE | DELETE | ADD | UPADE | DELETE |
| GRADES | GRADES | GRADES | GRADES | GRADES | - |
| ABSENCES | ABSENCES | ABSENCES | ABSENCES | - | - |
| BEHAVORS | BEHAVORS | BEHAVORS | BEHAVORS (behavour\_type=’czastkowa’) | BEHAVORS | - |
| - | LESSONS (topic) | - | - | LESSONS (topic) | - |
| - | PERSONS | - | - | - | - |

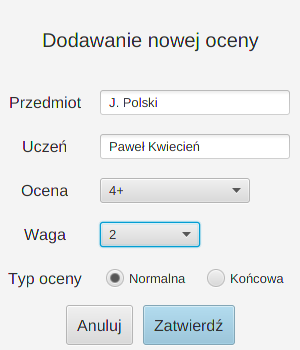
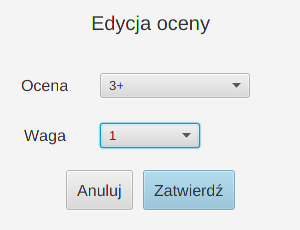
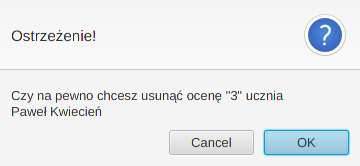
## Projekt aplikacji użytkownika

## Diagram przypadków użycia

### 3.2.2. Interfejs graficzny i struktura menu







### Metoda podłączania do bazy danych – integracja z bazą danych

Aplikacja do bazy danych zostanie podłączona z pomocą biblioteki Hibernate. Hibernate jest najpopularniejszą biblioteką służącą do mapowania obiektowo-relacyjnego w Javie (ORM / Object Relational Mapping. Zapewnia on przede wszystkim translację danych pomiędzy relacyjną bazą danych a światem obiektowym, dzięki czemu można rzutować obiekty, stosowane w obiektowych językach programowania, takich jak Java bezpośrednio na istniejące tabele bazy danych. Dodatkowo Hibernate zwiększa wydajność operacji na bazie danych dzięki buforowaniu  
i minimalizacji liczby przesyłanych zapytań. Hibernate potrzebuje określenia sterownika jakiego chcemy używać (w tym przypadku będzie to *com.mysql.jdbc.Driver*), adresu URL do bazy danych, nazwy użytkownika mającego dostęp do bazy oraz jego hasło.

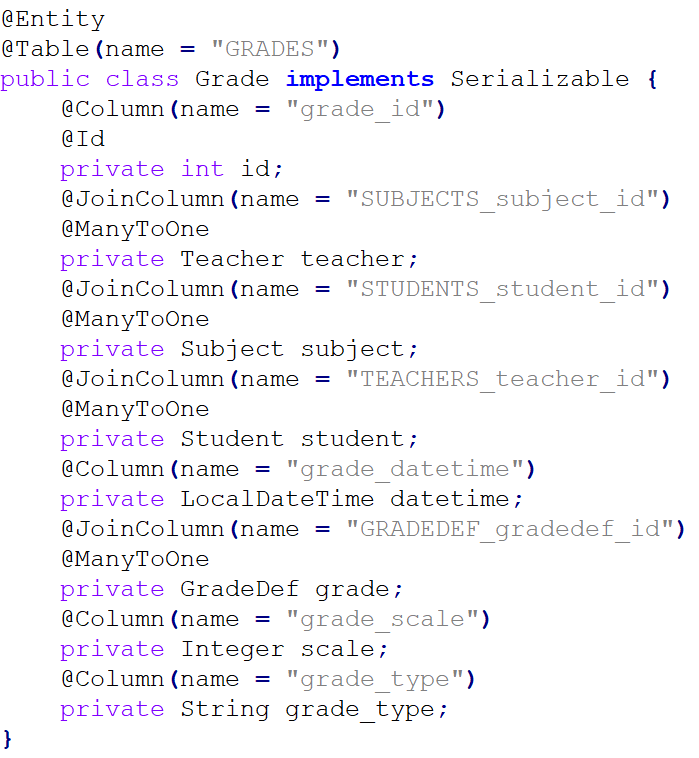
Klasa zawierająca obiekty statyczne będące interfejsami dla komunikacji z bazą danych.



Przykład klasy zawierający metody obsługujące transakcje z bazą danych oraz umożliwiające operacje na tej bazie. (Tutaj mały wycinek ważniejszych funkcjonalności).

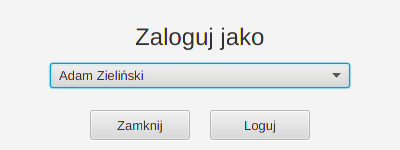


Przykładowa klasa będąca odpowiednikiem tabeli „Grade”.

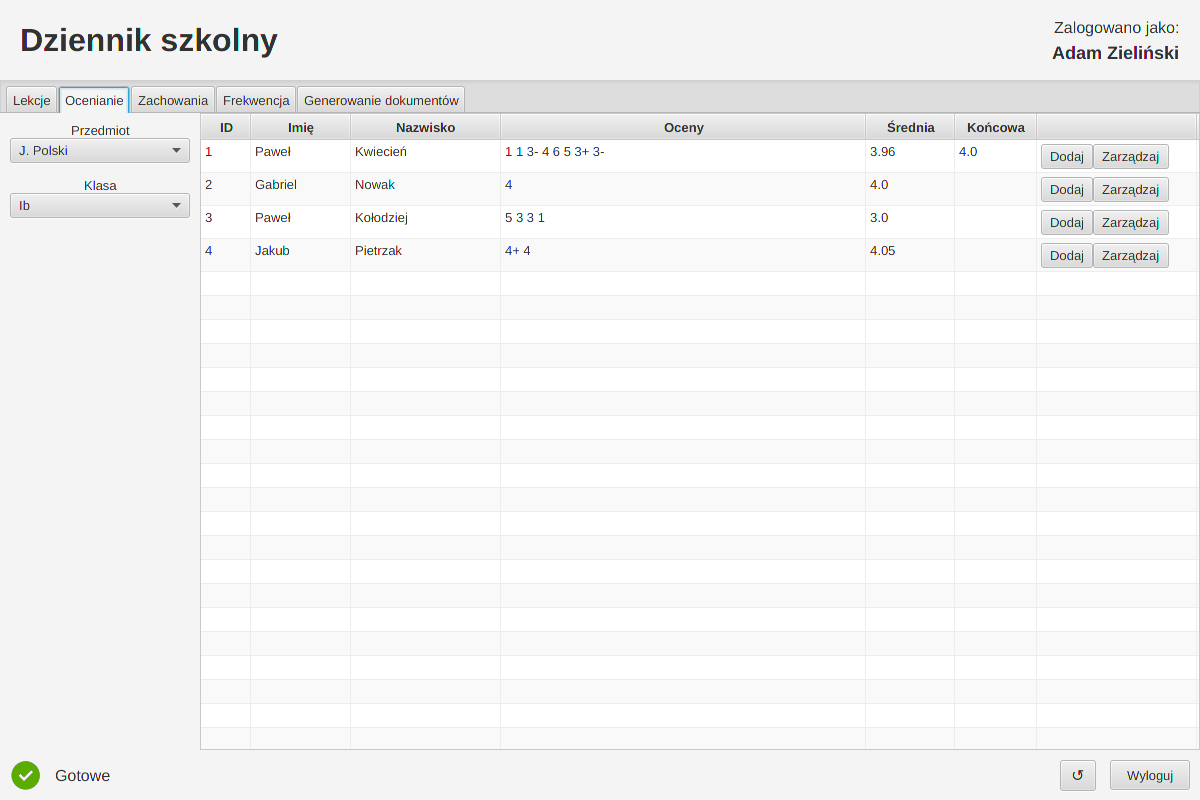


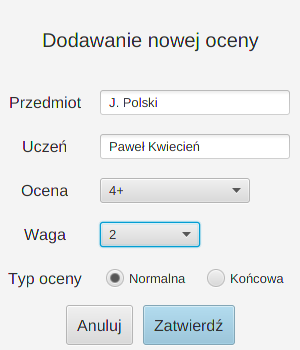
### Obsługa aplikacji

Po uruchomieniu aplikacji wita nas okno logowana w którym możemy wybrać osobę za pomocą której chcemy się zalogować do aplikacji.

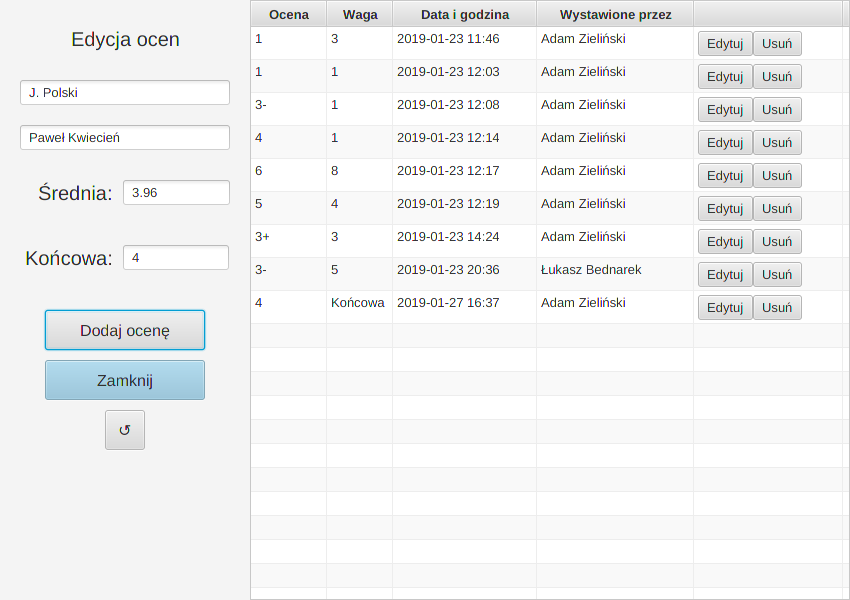


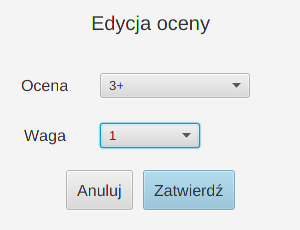
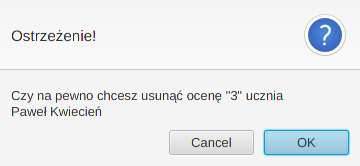
Po zalogowaniu wyświetla się okno główne programu. Okno aplikacji podzielone jest na pasek tytułowy zawierający nazwę aplikacji oraz imię i nazwisko osoby obecnie zalogowanej, część główną zawierającą strony przełączane za pomocą kart oraz stopkę, która zawiera status wykonanej ostatnio operacji na bazie i jakie wystąpiły błędy w połączeniu, przycisk odświeżenia danych (ponowne pobranie z bazy danych) oraz przycisk wylogowania. Część główna aplikacji zawiera wcześniej wspomniane strony przełączane za pomocą kart. Każda karta zawiera logicznie uporządkowane odpowiednie funkcjonalności. Dla przykładu przyjrzymy się karcie „Ocenianie” która zawiera interfejs do sprawdzania ocen uczniów oraz umożliwia zarządzanie ocenami uczniów. Po lewej stronie znajduje się belka z atrybutami, które można modyfikować i za ich pomocą wyświetlać różne dane w tabeli znajdującej się po prawej stronie. Na karcie „Ocenianie” możliwe jest wyświetlanie ocen uczniów z konkretnego przedmiotu oraz dla konkretnej klasy. Tabela zawiera kolumny z ID, imieniem i nazwiskiem ucznia, poszczególnymi jego ocenami, średnią ważoną ocen oraz oceną końcową.



Dla każdego ucznia istnieje możliwość dodania oceny zwykłej lub końcowej. W tym celu należy kliknąć przycisk „Dodaj” w wierszu ucznia, któremu ocenę chcemy dodać, a następnie wybrać odpowiednie parametry tj. ocenę, wagę oceny oraz typ oceny i zatwierdzić dodanie.

Dla każdego ucznia istnieje również możliwość wejścia w tryb zarządzania ocenami osobno.   
W tym celu należy wybrać przycisk „Zarządzaj” w wierszu ucznia, którego chcemy sprawdzić   
i ewentualnie zmodyfikować oceny.



Wyświetli nam się nowe okno ze szczegółowymi ocenami danego ucznia z danego przedmiotu który wybraliśmy we wcześniejszym oknie. W tym oknie mamy możliwość podejrzenia ocen, ich wag, daty   
i godziny wystawienia oraz nauczyciela wystawiającego. Okno pokazuje też przedmiot na którym wykonujemy operacje, imię i nazwisko ucznia, średnią ważoną ocen oraz ocenę końcową. Mamy również możliwość dodania nowej oceny za pomocą przycisku po lewej stronie okna, zamknięcia okna, odświeżenia danych w tym oknie oraz możemy zmodyfikować konkretną ocenę lub ją usunąć poprzez przyciski umieszczone w każdym wierszu tabeli. Edycja oceny wymaga jedynie podania nowej oceny oraz nowej wagi i zatwierdzenia zmian, a usuwanie oceny poprzedzone jest komunikatem ostrzegawczym zapobiegającemu przypadkowemu usunięciu oceny ucznia.

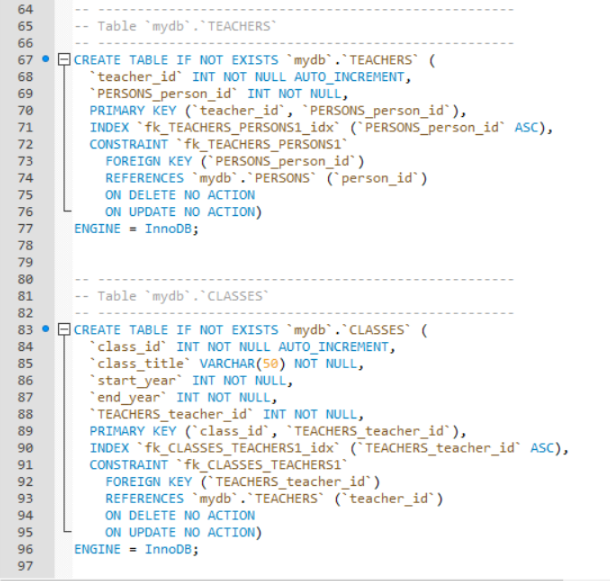
### Dystrybucja aplikacji

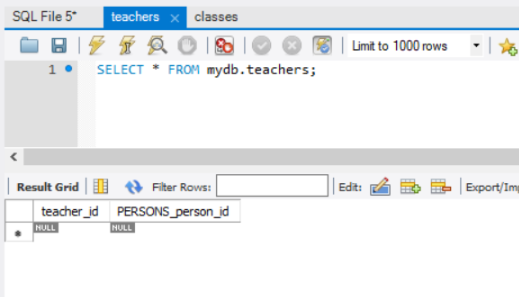
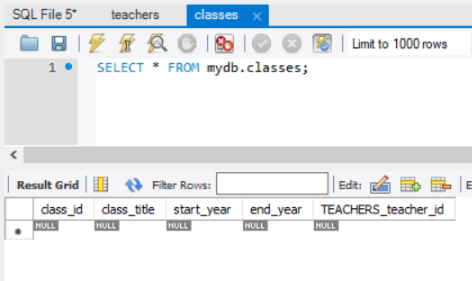
Aplikacja została napisana z użyciem języka programowania Java, więc może być dystrybuowana jako plik wykonywalny Java (plik z rozszerzeniem .jar) dzięki czemu aplikację będzie można uruchomić na każdej maszynie ze wsparciem dla technologii Java (Windows, GNU/Linux, Mac OS). Komputer klienta w takim razie musi zostać wyposażony w oprogramowanie umożliwiające obsługę i uruchamianie aplikacji stworzonych w technologii Java tj. Java Runtime Environment (JRE) które można pobrać ze strony producenta pod linkiem <https://www.java.com/pl/download/> .

# Implementacja sytemu baz danych

## 4.1 Tworzenie tabel i definiowanie ograniczeń

Tabele, relacje oraz ograniczenia zostały wygenerowane automatycznie na podstawie stworzonego modelu fizycznego. Przykład dla tabeli TEACHERS oraz CLASSES.

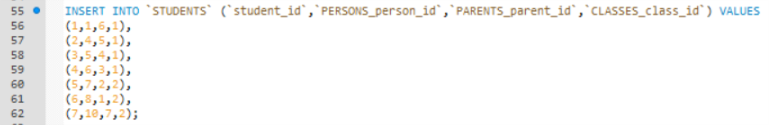


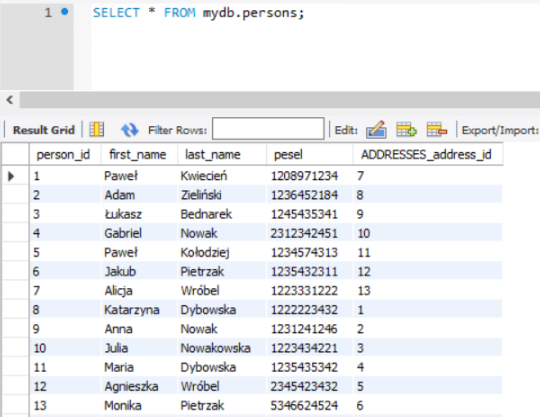
Efekt implementacji kodu

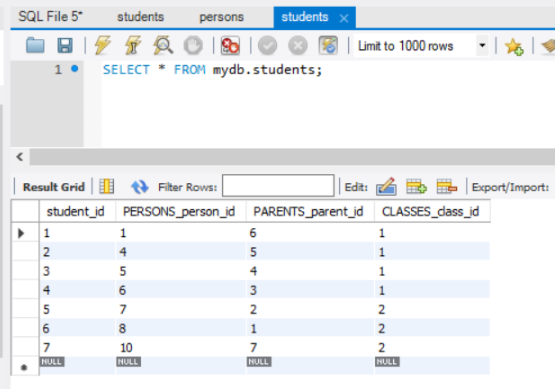
## 4.2 Przykładowe dane oraz ich test

Dodanie przykładowych danych do tabeli PERSONS oraz STUDENTS

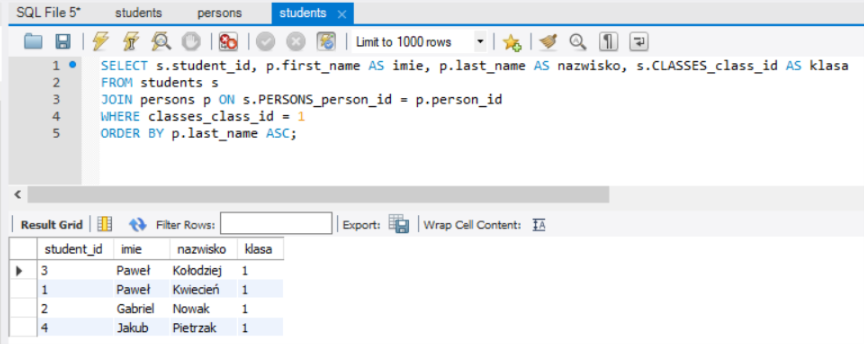
## 

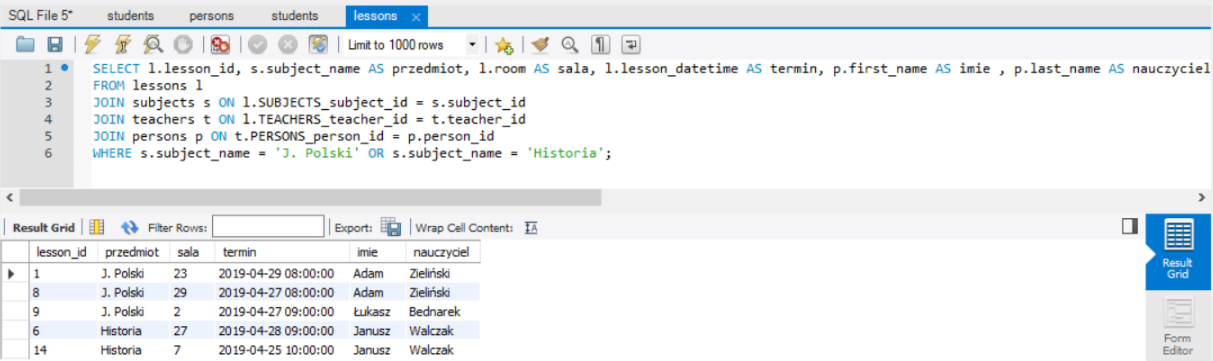


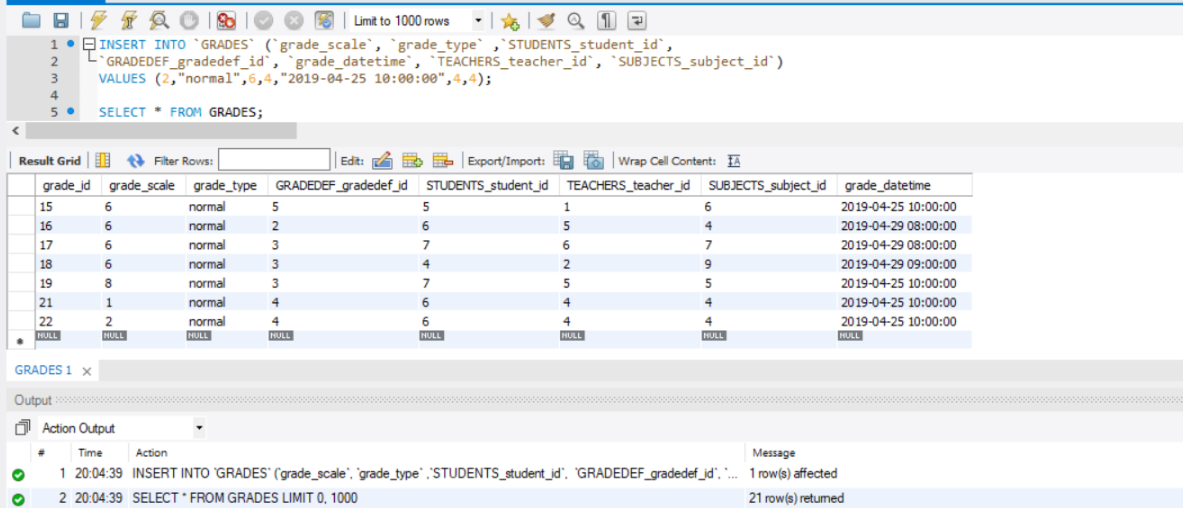
Efekt implementacji

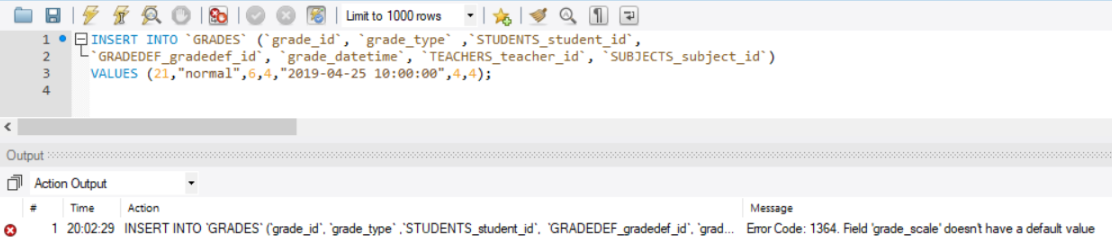


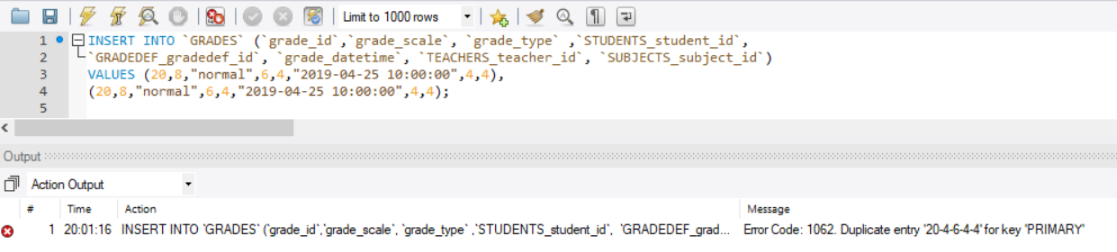
## Testy bazy

Wyszukanie wszystkich uczniów klasy o numerze id 1

Wyszukanie wszystkich zajęć z języka polskiego oraz historii

Auto increment

Not null

Unique

## 4.3. Implementacja mechanizmów przetwarzania danych

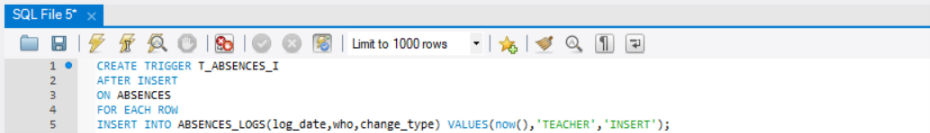
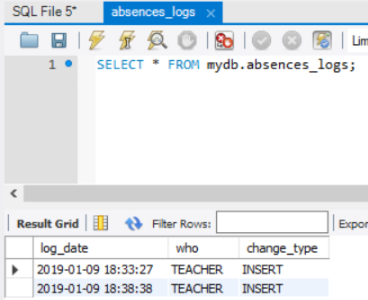
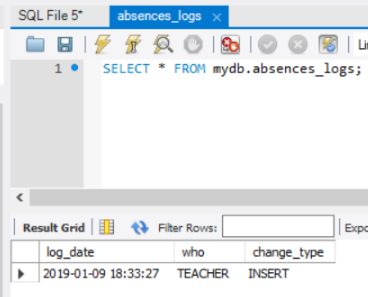
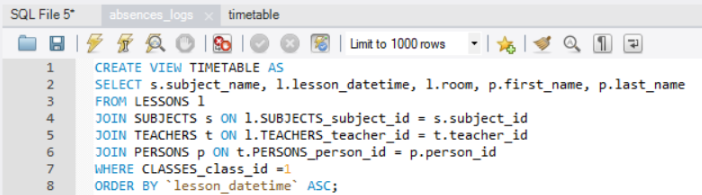
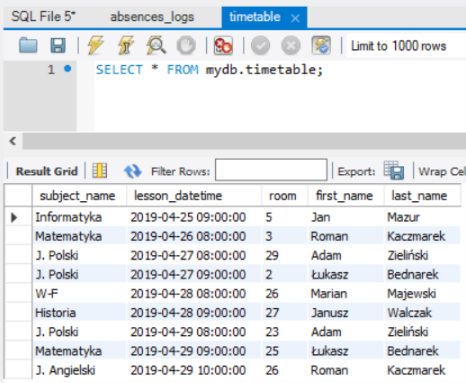
Trigger dodający informacje do tabeli logów, wyzwalany przy dodaniu wiersza w tabeli ABSENCES

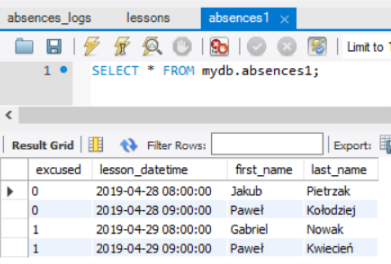
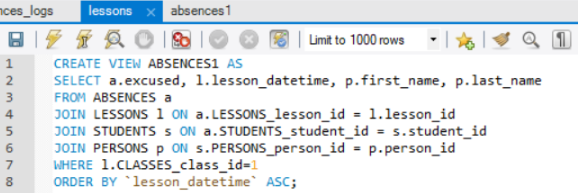
Tabela logów przed i po poleceniu INSERT na tabeli ABSENCES

Widok lekcji dla klasy o numerze id 1

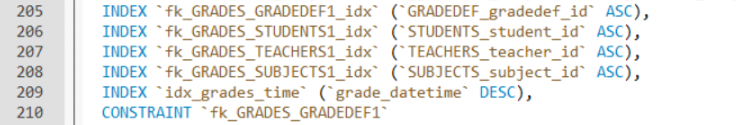


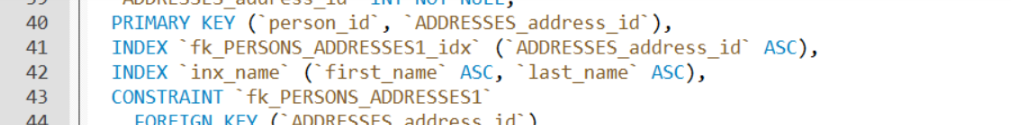


Widok nieobecności klasy o numerze id 1

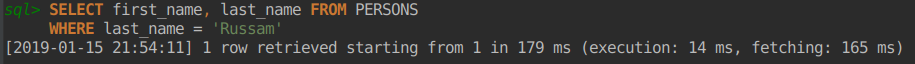


Indeksy

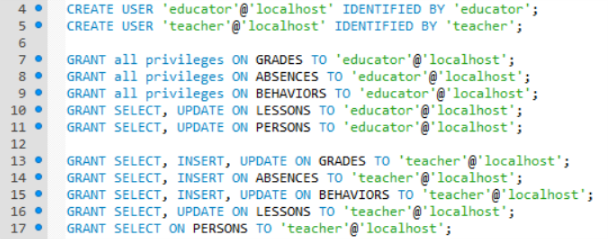
Indeksy na kluczach obcych zostały wygenerowane automatycznie. Przykłady indeksów dla tabeli GRADES, PERSONS oraz ABSENCES



Czas działania kodu bez zastosowania indeksów oraz z indeksami dla tabeli PERSONS



Uprawnienia użytkowników nauczyciel oraz wychowawca



Zgody i odmowy dla poleceń wykonanych przez użytkownika educator i teacher

