

# Baza Danych Szlaki Górskie

Dokumentacja projektu zaliczeniowego z przedmiotu Bazy Danych 1

Alicja Kałuża, 15 stycznia 2024

## 1 Projekt koncepcji, założenia

### 1.1 Temat projektu

Tematem projektu jest baza danych zawierająca informacje o szczytach, schroniskach i szlakach górskich. Aplikacja ma umożliwiać planowanie tras między różnymi szczytami i schroniskami.

### 1.2 Analiza wymagań użytkownika

Użytkownik ma mieć możliwość zobaczyć dostępne szczyty i schroniska oraz informacje o nich. Ma mieć dostęp do listy szlaków zawartych w bazie danych - oprócz punktu startowego-schroniska i końcowego-szczytu, kolor szlaku oraz czas pokonania. Użytkownik aplikacji może także usuwać i dodawać dane z każdej z tabel w bazie danych - szlaki, schroniska oraz szczyty oraz punkt\_punkt. Do tabeli punkt dane dodają się automatycznie za pomocą wyzwalacza. Istnieje możliwość rejestracji i logowania. Główną funkcjonalnością projektu jest możliwość planowania tras: Zalogowany jak i niezalogowany użytkownik może tworzyć trasę dodając do niej kolejne punkty, między którymi istnieją szlaki.

### 1.3 Zaprojektowanie funkcji

Podstawowe funkcje realizowane w bazie danych wraz z opisem:

1. Dodawanie szczytów i schronisk:
  - (a) Funkcja `schronisko_wprowadz` odpowiada za dodawanie nowych schronisk do listy punktów.
  - (b) Funkcja `szczyt_wprowadz` odpowiada za dodawanie nowych szczytów do listy punktów.
2. Usuwanie szczytów i schronisk:
  - (a) Funkcja `schronisko_usun` usuwa schroniska z listy punktów po ich usunięciu z tabeli `schronisko`.
  - (b) Funkcja `szczyt_usun` usuwa szczyty z listy punktów po ich usunięciu z tabeli `szczyt`.
3. Generowanie tras:

Funkcja `do_wprowadzania_punkt_trasa` jest używana jako trigger, który aktualizuje trasę odpowiednio zmieniając jej długość i czas trwania.
4. Wyszukiwanie szlaków na szczyt:

Funkcja `szlak` pozwala na wyszukiwanie dostępnych szlaków na wybrany szczyt na podstawie przypisanego szlaku w tabeli `szlak`.
5. Kolejne punkty w trasie:
  - (a) Funkcja `kolejny_pkt_gora` zwraca kolejne punkty znajdujące się wyżej od poprzednio dodanego punktu do trasy.
  - (b) Funkcja `kolejny_pkt_dol` zwraca kolejne punkty znajdujące się niżej od poprzednio dodanego punktu do trasy.
6. Wypisywanie punktów i połączeń:
  - (a) Funkcja `wypisz_punkty` wypisuje punkty z ich nazwami i wysokościami.

- (b) Funkcja wypisz\_polaczenia wypisuje połączenia między punktami wraz z informacjami o szlaku, czasie i odległości.

#### 7. Generowanie raportów:

Widoki trasa\_odleglosci, trasa\_ilosci i punkty\_ilosci generują raporty na podstawie danych związanych z trasami, ich odległościami i ilościami.

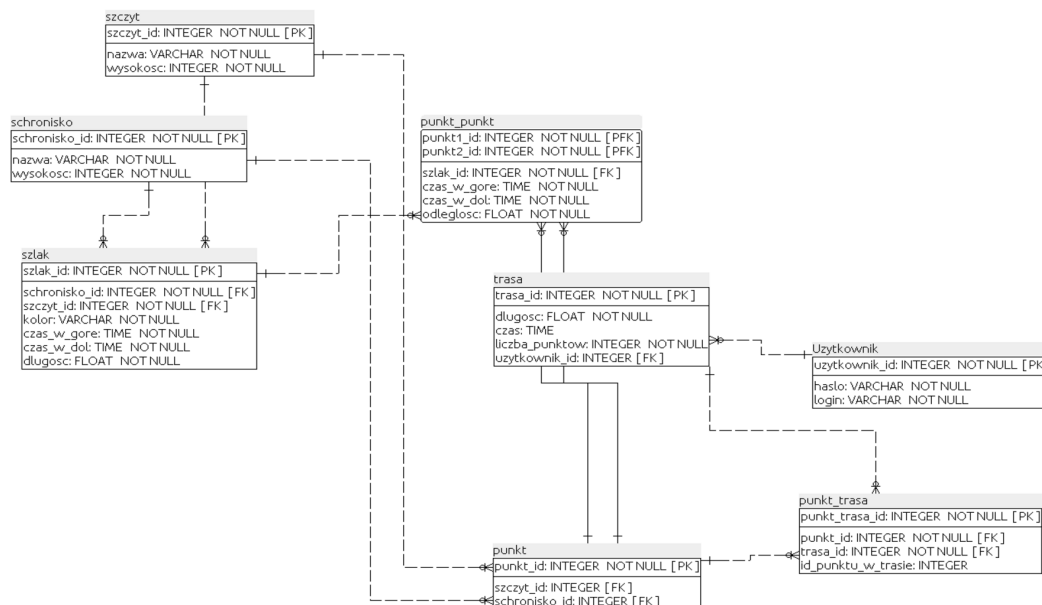
Te funkcje obejmują podstawowe operacje, takie jak dodawanie, usuwanie, generowanie tras, wyszukiwanie informacji o szlakach, wypisywanie danych i generowanie raportów na podstawie danych w bazie danych.

## 2 Projekt diagramów (konceptualny)

### 2.1 Budowa i analiza diagramu przepływu danych (DFD)

- Wejścia:** Źródłem danych w systemie są dane wprowadzane ręcznie przez użytkownika oraz dane wprowadzone przez administratora.
- Przechowywanie:** Dane przechowywane w systemie to informacje o szczytach, szlakach i schroniskach oraz o użytkownikach i utworzonych przez nich trasach.
- Wyjścia:** 3 raporty generowane na podstawie widoków, informacje o utworzonych trasach- ich łącznej długości, łącznym czasie, kolejnych punktach, kolejnych elementów trasy - połączenia między poszczególnymi punktami.
- Procesy:** Tworzenie się trasy i zapisywanie rozpoczyna użytkownik, który chce utworzyć trasę- wybiera punkt startowy, następne punkty są generowane przez odpowiednie funkcje: kolejny\_pkt\_gora oraz kolejny\_pkt\_dol. Trasa i informacje o niej są za każdym dodaniem punktu odświeżane za pomocą wyzwalacza.

### 2.2 Zdefiniowanie encji (obiektów) oraz ich atrybutów, zaprojektowanie relacji pomiędzy encjami



Rysunek 1: Wykres przedstawiający schemat zaprojektowanej bazy danych.

## 3 Projekt logiczny, słowniki danych

- Tabele schronisko, szczyt są tabelami zawierającymi informacje o schroniskach i szczytach- ich atrybuty: wysokosc (INTEGER), nazwa (VARCHAR) oraz odpowiednio schronisko\_id oraz szczyt\_id jako klucze główne, automatycznie inkerementowane.

2. Tabela szlak- jej atrybuty to: szlak\_id jako klucz główny (SERIAL),schronisko\_id, szczyt\_id - klucze obce. Pełnią rolę punktu startowego i końcowego danego szlaku. Relacje pomiędzy tabelami szczyt i szlak to 1:n jak i między tabelami schronisko i szlak. Ze schroniska jak i ze szczytu może wychodzić wiele szlaków. Kolejne atrybuty to kolor jako VARCHAR, czas\_w\_gore, czas\_w\_dol - typ danych TIME (czas pokonania szlaku w górę jest dłuższy niż przy schodzeniu) oraz dlugosc(FLOAT).
3. Tabela punkt- tabela reprezentująca punkt na mapie szlaków- schronisko bądź szczyt. Jego atrybuty to klucz główny: punkt\_id, jeden z atrybutów: schronisko\_id lub szczyt\_id nie jest ustawiony na wartość NULL- jest to informacja czy dany punkt jest schroniskiem, czy szczytem. Atrybuty schronisko\_id oraz szczyt\_id to klucze obce. Relacja między tabelami schronisko-punkt 1:n oraz między szczyt-punkt 1:n.
4. Tabela punkt-punkt reprezentująca połączenie między punktami na mapie szlaków. Klucz główny to (punkt1\_id,punkt2\_id) będące równocześnie kluczami obcymi. Zapobiega to wprowadzaniu połączeń między punktami, między którymi już istnieje połączenie. Tabela zawiera również informacje jaki szlak łączy te punkty: atrybut szlak\_id , odleglosc(FLOAT) - odległość między danymi punktami oraz jaki jest czas pokonania danego odcinka w górę i w dół-atrybuty: czas\_w\_gore i czas\_w\_dol (TIME). Relacja między encjami szlak-punkt\_punkt 1:n- ten sam szlak może łączyć wiele punktów, między punkt i punkt\_punkt występuje relacja 1:n - jeden punkt może występować w różnych połączeniach.
5. Tabela Uzytkownik - zawiera informacje o zarejestrowanych użytkownikach. Informacje niezbędne do autoryzacji:uzytkownik\_id- unikalny identyfikator(SERIAL),login(VARCHAR) oraz haslo(VARCHAR). Relacja między encjami Uzytkownik - trasa 1:n - użytkownik może mieć zapisanych wiele tras.
6. Tabela trasa - klucz główny trasa\_id, atrybut uzytkownik\_id-klucz obcy, ustawiony domyślnie na wartość NULL, ponieważ trasę może utworzyć również użytkownik niezarejestrowany w systemie. Kolejne atrybuty to informacje o tworzonej/utworzonej trasie: łączny czas(TIME), łączna długość-odleglosc(FLOAT) oraz ilosc\_punktow(INTEGER) - ilość dodanych punktów - atrybut niezbędny przy tworzeniu trasy.
7. Tabela punkt\_trasa- jej kluczem głównym jest punkt\_trasa\_id(SERIAL). Jest to tablica asocjacyjna dla tabel punkt i trasa - między którymi występuje relacja wiele do wielu. Ważnym elementem tabeli punkt\_trasa jest atrybut id\_punktu\_w\_trasie(INTEGER) , dzięki któremu znana jest kolejność punktów dodanych do danej trasy.

### 3.1 Analiza zależności funkcyjnych i normalizacja tabel (dekompozycja do 3NF)

Pierwsza postać normalna zostaje spełniona, ponieważ w każdej z relacji pozostałe atrybuty są zależne od unikalnego klucza głównego, druga postać normalna, ponieważ atrybuty nie wchodzące w skład klucza są w pełni funkcjonalnie zależne od całego klucza głównego (nie od jego części) oraz trzecia Relacja jest trzeciej postaci normalnej (3NF) wtedy i tylko wtedy, gdy w każdej chwili każda krotka składa się z wartości klucza głównego identyfikującego pewną encję oraz ze zbioru zero lub więcej wzajemnie niezależnych wartości atrybutów charakteryzujących ją w jakiś sposób. co wyjaśniono poniżej:

1. Relacje szlak,schronisko - klucz główny został wybrano jak opisano poprzednim punkcie, ponieważ mogą się zdarzyć schroniska o tej samej nazwie i wysokości, to samo dotyczy szczytów, id jest unikalne i do niego zależą nazwa i wysokość.
2. W relacji punkt\_punkt unikalny klucz główny od którego zależne są pozostałe atrybuty do punkt1\_id oraz punkt2\_id - w założeniu działania aplikacji zakładamy, że nie istnieje więcej niż jedno połączenie między dwoma tymi samymi punktami. Może takie istnieć, ale po złączeniu różnych odcinków.
3. W relacji trasa trasa\_id jest unikalnym kluczem głównym- wartości pozostałych atrybutów nie są unikalne i są zależne od klucza głównego.
4. W relacji punkt\_trasa punkt\_trasa\_id jest unikalnym kluczem głównym- wartości pozostałych atrybutów nie są unikalne i są zależne od klucza głównego.
5. W relacji Uzytkownik uzytkownik\_id jest unikalnym kluczem głównym- nazwa użytkownika login nie jest unikalne a hasło jest zaszyfrowane.

### 3.2 Zaprojektowanie operacji na danych

Dla realizacji funkcji wyspecyfikowanych w założeniach projektu zdefiniowano funkcje składowane umożliwiające dostęp do danych (opisane w punkcie 1.3) , których nie można zrealizować za pomocą prostych zapytań SELECT - te zostały zdefiniowane w kodzie aplikacji. Te proste zapytania obejmują wypisywanie wszystkich dostępnych szlaków, schronisk, szczytów i tras.

## 4 Projekt funkcjonalny

### 4.1 Panel sterowania

Do wyboru odpowiednich formularzy oraz dostępu do danych służy menu widoczne dla użytkownika w każdej podstronie.

### 4.2 Interfejsy do prezentacji, edycji i obsługi danych

**Wprowadzanie i usuwanie danych:**

1. Użytkownik może dodać dane do każdej z tabeli: szczyt,schronisko,szlak za pomocą formularzy, w których użytkownik podaje niezbędne dane.
2. Do tabeli punkt\_punkt można wprowadzanie danych jest możliwe dzięki następującemu formularzowi - wypisywane są wszystkie dostępne punkty w postaci tabeli wraz z ich nazwami(jest zaimplementowana w bazie funkcja która wyprowadza te informacje : wypisz\_punkty()). Użytkownik zaznacza dwa punkty z wszystkich dostępnych oraz uzupełnia formularz potrzebnymi danymi o połączeniu : czas, odległość. Gdy wysyła formularz wprowadzane jest połączenie między wybranymi punktami wraz z pozostałymi atrybutami: czas i odległość.
3. Do tabeli trasa dodawany jest kolejny wiersz w momencie kiedy użytkownik klika przycisk : 'Stwórz trasę'. W zależności czy użytkownik jest zalogowany czy nie dodawana jest trasa z id użytkownika lub bez.
4. Tabela punkt\_trasa jest uzupełniana gdy użytkownik wybiera kolejne punkty do swojej trasy. Formularz wyboru kolejnych punktów pojawia się za każdym razem z dostępnymi punktami z poprzednio wybranego punktu. Wybieranie punktu odbywa się znowu dzięki 'checkbox' znajdującemu się obok każdego wypisanego punktu.
5. Formularz logowania i rejestracji to prosty formularz, w którym użytkownik podaje login i hasło.
6. Usuwanie danych jest możliwe dzięki formularzom, w którym są wypisywane wszystkie dane z danej tabeli a obok każdego elementu znajduje się 'checkbox', wysłanie formularza powoduje usunięcie zaznaczonych elementów z danej tabeli.

### 4.3 Wizualizacja danych

1. Dane z wszystkich relacji pokazywane są w tabelach.
2. Po udanym logowaniu/rejestracji użytkownik ma dostęp do utworzonych przez siebie tras. Informacje o długości i czasie oraz kolejnych punktach i odcinkach w danej trasie.

#### Raporty

W aplikacji są generowane trzy raporty w postaci tabel.

1. Użytkownicy z największą liczbą zapisanych tras. Użytkownik widzi login i ilość tras.
2. Użytkownicy z największą łączną długością zapisanych tras. Użytkownik widzi login i łączną długość.
3. Punkty najczęściej dodawane do planowanych tras. Użytkownik widzi nazwę punktu i ile razy był dany punkt zapisany.

## 4.4 Makropolecenia

Po wysłaniu formularza wywoływane są odpowiednie skrypty, które wykonują zestaw działań w bazie danych. Wybrane makropolecenia:

1. Usuwanie danych
2. Wywoływanie zdefiniowanych funkcji w bazie danych, czyli np: `kolejny_pkt_gora(id_trasy)` - na podstawie tworzonej w danym momencie trasy
3. Dodawanie danych

## 5 Dokumentacja

### 5.1 Wprowadzanie danych

Część danych została wprowadzona ręcznie. Jak zostało opisane w poprzednich punktach dane może także ręcznie wprowadzać użytkownik.

### 5.2 Dokumentacja użytkownika

Użytkownik niezalogowany porusza się po aplikacji za pomocą menu:

1. Schroniska - podstrona zawiera listę dostępnych schronisk, formularz usuwania i dodawania.
2. Szlaki - podstrona zawiera listę dostępnych szlaków oraz połączeń między punktami formularz usuwania i dodawania. Formularze opisane w punkcie 4.2.
3. Szczyty - podstrona zawiera listę dostępnych schronisk, formularz usuwania i dodawania.
4. Trasy - podstrona zawiera dwa przyciski:
  - (a) wybór szczytu - użytkownik wybiera szczyt, który chciałby zdobyć. Rezultatem żądania jest lista wszystkich dostępnych szlaków prowadzących na wybrany szczyt.
  - (b) Stwórz trasę - użytkownik wybiera punkt startowy z listy dostępnych punktów. Następnie po sukcesywnym dodaniu do trasy pojawia się przycisk, po którego naciśnięciu wypisują się punkty do których użytkownik ma możliwość się udać z poprzedniego punktu. W każdym momencie użytkownik może zakończyć tworzenie trasy - gdy użytkownik jest zalogowany trasa zapisuje się, w przeciwnym razie jest usuwana. Wyświetla się raport dotyczący trasy: ile punktów ma, jaką ma długość, czas oraz kolejne połączenia.
5. Logowanie/Rejestracja - podstrona zawiera formularz rejestracji i logowania
6. Rankingi - podstrona zawiera tabelę z raportami opisanymi w punkcie 4.3

Użytkownik zalogowany ma możliwość wyboru dodatkowych podstron:

1. Moje trasy - lista zapisanych tras wraz z informacjami o nich z możliwością ich usunięcia
2. Wyloguj

### 5.3 Dokumentacja techniczna

Wygenerowana automatycznie. Zamieszczona w pliku.

### 5.4 Wykaz literatury

1. Przykłady wykorzystania języka SQL przedstawione na platformie upel
2. <https://www.php.net/manual/>
3. <https://stackoverflow.com/>
4. <https://www.postgresql.org/docs/current/>
5. <https://chat.openai.com/>
6. <http://natatry.pl/>
7. <https://www.tatry.info.pl/>
8. <https://tatromaniak.pl/szlaki/>