

**autor:     Michał Jaworowski**

## **SKOKI NARCIARSKIE**

### **Dlaczego taka tematyka?**

Od czasu sukcesów Adama Małysza skoki narciarskie są moim hobby. Śledzę na bieżąco wszystkie wyniki( zarówno konkursów jak i serii treningowych i kwalifikacji ), analizuje je, nieraz wielokrotnie, ale śledzę również wszystko wokół z tym związane: przebudowy skoczni, metody treningowe, nowinki techniczne, artykuły zagraniczne i wiele innych.

### **Co zawiera moja baza danych?**

Moja baza danych zawiera przede wszystkim:

- wyniki zawodów indywidualnych skoczków, które będą się składały na wyliczane klasyfikacje( indywidualna, drużynowa ),
- sezony i ilość przeprowadzonych w nich konkursów,
- konkursy odbyte i zaplanowane,
- skocznie i ich historie przebudowań,
- skoczków i ich przynależności do klubów,
- reprezentacje i ich główni trenerzy,
- historię rekordów danych skoczni,
- inne, mniej istotne informacje( np. sponsorzy klubów czy wykonawców przebudowań ), lecz dopełniające całość.

### **Scenariusz:**

Witryna na wzór strony skijumping.pl. Są dwie kategorie użytkowników: admin i zwykły użytkownik. Aby mieć dostęp do informacji, należy posiadać konto. Jeśli takowego jeszcze nie posiadamy, dostępna jest rejestracja, w której możemy utworzyć konto( tylko użytkownika ). Po zalogowaniu, w części głównej, admin ma dostęp do Panelu administratora, w którym znajduje się możliwość edycji danych w bazie( możliwość dokonania UPDATE, DELETE, INSERT ), a także do wszystkiego tego, co ma użytkownik. Dodatkowo Admin ma możliwość dodawania, usuwania i aktualizowania użytkowników( tylko tych zwykłych ). Użytkownicy mają dostęp jedynie do przeglądania przygotowanych informacji na temat sezonów, konkursów, wyników poszczególnych zawodów, klasyfikacji danego sezonu, informacji o zawodnikach, reprezentacjach, trenerach, skoczniach, pogodzie w danym konkursie i innych. Mają możliwość zaznaczenia względem której tabeli chcą posortować przygotowane dane. Przygotowany jest także kalendarz zawodów oraz wykres częstości wystawianych not. Użytkownik ma jeszcze możliwość zgłaszania błędów za pomocą Feedback.

## Jakie funkcjonalności zawarłem w bazie?

W zaprojektowanej przeze mnie bazie zawarłem:

### a) Widoki:

- i. **wyniki\_konkursow\_id** -> Obliczanie wyników konkursów( wszystkich ) - zawiera id zawodnika, id konkursu oraz wynik ( posortowane względem daty konkursu, a potem wyniku ) – widok ten wykorzystuje procedura do obliczenia klasyfikacji generalnej danego lub wszystkich sezonów
- ii. **wyniki\_konkursow** -> również oblicza wyniki konkursów, lecz zawiera imię i nazwisko skoczka, jego wynik, miejscowość, w której znajduje się skocznia, sezon oraz datę konkursu
- iii. **skoki\_poza\_rozmiarHS** -> wyszukuje wszystkie skoki dalsze bądź równe niż punkt bezpieczeństwa( punkt HS ) – zawiera punkt konstrukcyjny skoczni, miejscowość skoczni, numer serii konkursu, odległość, imię i nazwisko skoczka, sezon oraz datę konkursu

### b) Funkcje:

- i. **f\_punkty\_za\_pkt\_k** -> przyjmuje jako parametr punkt konstrukcyjny( wartość INTEGER ), funkcja jest wykorzystywana do obliczeń wyników konkursów, w zależności od punktu konstrukcyjnego( punkt k ) zwraca odpowiednią bonifikatę punktową za osiągnięcie punktu konstrukcyjnego( INTEGER ),
- ii. **f\_punkty\_za\_metr** -> przyjmuje jako parametr punkt konstrukcyjny ( wartość INTEGER ). Funkcja jest wykorzystywana do obliczeń wyników konkursów. W zależności od punktu konstrukcyjnego zwraca odpowiednią wartość należną za każdy metr( NUMBER ),
- iii. **f\_sprawdz\_nazwe\_sezonu** -> przyjmuje jako parametr nazwę sezonu ( VARCHAR2 ) i sprawdza poprawność formatu. Zwraca na wyjściu czy nazwa jest poprawna, a jeśli nie, to zwraca opis dlaczego( VARCHAR2 ). Wykorzystywana jest w zapytaniu SELECT do sprawdzenia i zestawienia poprawności nazw.
- iv. **f\_logowanie** -> przyjmuje parametry nazwa użytkownika( VARCHAR2 ) oraz hasło ( VARCHAR2 ) i sprawdza czy podany użytkownik jest w bazie i czy hasło jest poprawne. Jeśli podane wartości się zgadzają, to funkcja zwraca wartość BOOLEAN ( true ), w przeciwnym wypadku false.

### c) Procedury:

- i. **p\_oblicz\_klasyf\_ind** -> nie przyjmuje parametrów. Służy do obliczenia klasyfikacji indywidualnej na podstawie wyników konkursów zapisanych do widoku wyniki\_konkursow\_id. Korzysta z tabeli punktacja, w której zawarta jest metodologia przyznawania punktów za odpowiednie lokaty. Oblicza klasyfikacje dla każdego sezonu oddzielnie,
- ii. **p\_oblicz\_klasyf\_druz** -> nie przyjmuje parametrów. Służy do obliczenia klasyfikacji drużynowej na podstawie wyników indywidualnych konkursów zapisanych do widoku wyniki\_konkursow\_id. Korzysta z tabeli punktacja, w

której zawarta jest metodologia przyznawania punktów za odpowiednie lokaty zawodnikom. Punkty zawodników z tego samego kraju są sumowane i przedstawione jako wynik reprezentacji. Oblicza klasyfikacje dla każdego sezonu oddzielnie,

- iii. **p\_wypisz\_zwyciezcow** -> jako parametr przyjmuje nazwę sezonu(VARCHAR2). Służy do wyszukania i wypisania wszystkich zwycięzców z wpisanego sezonu na podstawie widoku wyniki\_konkursow\_id. Zwycięzcy są zapisywani do tabeli zwyciezcy wraz z miejscowością, w której odbył się konkurs, nazwą sezonu oraz datą,
- iv. **p\_aktualizuj\_rekordy** -> nie przyjmuje parametrów. Procedura ta służy do zaktualizowania historii rekordów skoczni. Przeszukuje odbyte sezony, konkursy i sprawdza wyniki czy skok był dłuższy niż obowiązujący rekord( rozróżnianie na letni i zimowy rekord ) oraz czy skok został uznany( nota łączna za skok przynajmniej 35 punktów ). Aktualizuje rekordy skoczni, dodając nowy, a stary aktualizując jako nieaktualny,
- v. **p\_aktualizuj\_przebudowy** -> nie przyjmuje parametrów. Procedura ta sprawdza, czy termin ukończenia przebudowy już minął. Wtedy ustawia pole ukończono jako 't', w przeciwnym wypadku 'n'.

d) **Wyzwalacze:**

- i. **t\_klasyf\_ind\_ilosc\_konkursow** -> wyzwalacz typu AFTER instrukcji INSERT lub DELETE na tabeli konk\_zaw. Służy do automatycznego zwiększania bądź zmniejszania liczby występów zawodnika w przypadku dodania go lub usunięcia z konkursu( również dodanie lub usunięcie z klasyfikacji danego sezonu, jeśli nie brał udziału w żadnym konkursie sezonu ),
- ii. **t\_sezon\_liczba\_konkursow** -> wyzwalacz typu AFTER instrukcji INSERT lub DELETE na tabeli konkurs . Służy do automatycznego zwiększania bądź zmniejszania liczby konkursów zgłoszonych w sezonie( tabela sezon ), po dodaniu lub usunięciu konkursu,
- iii. **t\_sezon\_ustal\_rodzaj** -> wyzwalacz typu BEFORE instrukcji INSERT na tabeli sezon. Służy do automatycznego ustalania rodzaju sezonu( zimowy/letni ) na podstawie długości nazwy sezonu( dł. 9 to 'zimowy', dł. 4 to 'letni' ),
- iv. **t\_edytuj\_wynik\_ind** -> wyzwalacz typu BEFORE instrukcji DELETE na tabeli wynik\_ind. Służy do zapisu zmian wykonywanych na tabeli wynik\_ind i zapisywaniu wersji rekordów sprzed edycji w tabeli wynik\_ind\_hist z adnotacją jaka operacja została wykonana i kiedy. W razie usunięcia wyniku, usuwa także powiązane z wynikiem noty,
- v. **t\_usun\_refer\_zawodnika** -> wyzwalacz typu BEFORE instrukcji DELETE na tabeli zawodnik. Jeśli zdecydujemy się( chociaż niezalecane ) na usunięcie zawodnika, to usuną się wszystkie jego referencje w tabelach: klasyf\_ind, konk\_zaw i zaw\_wyn,

- vi. **t\_usun\_wynik** -> wyzwalacz typu AFTER instrukcji DELETE na tabeli zaw\_wyn. Jeśli usunięta zostaje powiązanie zawodnika z wynikiem, to wynik archiwizujemy.

e) **Ograniczenia pól tabel:**

i. **unique:**

- klub( nazwa ) <- nazwa klubu,
- kraj\_zaw( nazwa ) <- nazwa kraju zawodnika,
- sponsorzy( nazwa ) <- nazwa sponsora klubu,
- skocznia( nazwa ) <- nazwa skoczni,
- wykonawca( nazwa ) <- nazwa wykonawcy przebudowy.

ii. **check:**

- konkurs( rodzaj ) <- 'indywidualny', 'drużynowy' lub 'kwalifikacje',
- kraj\_zaw( limit\_start ) <- limit startowy kraju między 2 a 7,
- noty( nota ) <- wartość między 0 a 20 oraz tylko podzielne przez 0.5,
- przebudowa( ukończono ) <- 't' jako tak, 'n' jako nie,
- rekord( typ ) <- 'zimowy' lub 'letni' typ rekordu,
- rekord( odległosc ) <- wartość między 0.0 a 260.0 oraz podzielne przez 0.5,
- skocznia( punkt\_k ) <- punkt konstrukcyjny skoczni między 90 a 200,
- skocznia( punkt\_hs ) <- rozmiar skoczni między 97 a 225,
- skocznia( przebudowana ) <- 't' jako tak, 'n' jako nie,
- wynik\_ind( nr\_serii ) <- 0 – kwalifikacje, 1 i 2 – serie konkursowe,
- wynik\_ind\_hist( rodzaj\_zmiany ) <- typ zmiany 'update' lub 'delete',
- sezon( liczba\_konk ) <- musi być liczbą dodatnią lub 0,
- sezon( nazwa ) <- musi być długości 9 lub 4, bo format zimowy to np. '2014/2015' lub format letni to np. '2015',
- sezon( rodzaj ) <- 'letni' lub 'zimowy' rodzaj sezonu.

iii. **default:**

- sezon( liczba\_konk ) <- 0,
- skocznia( przebudowana ) <- 'n',
- przebudowa( ukończono ) <- 'n',
- i wiele innych.

f) **Pakiet:**

- i. **pakiet** -> zawierający wszystkie opisane wcześniej procedury i funkcje.

## Wizualizacja danych

Po zaprojektowaniu i organizacji bazy danych utworzyłem interfejs graficzny w APEX 5. Scenariusz użytkownika opisany został wcześniej.

## Wymagana kolejność wykonywania skryptów:

Plik	Dlaczego
3_skryptDDL.sql	
6c_wyzwalacze.sql	wykorzystywane przy instrukcjach INSERT
6a_funkcje.sql	wykorzystywane w procedurach i SELECT do tworzenia widoków
4_insert.sql	
5_select.sql	wykorzystywane w procedurach
6_procedury.sql	
7_pakiet.sql	
8_APEX.sql	