

BADA Grupa B Klub Lekkoatletyczny

Mateusz Gawlik, Adam Lipian

Politechnika Warszawska, Wydział Elektroniki i Technik
Informacyjnych

27 listopada 2023

Spis treści

1	Zakres i cel projektu (opis założeń funkcjonalnych projektowanej bazy danych)	3
2	Definicja systemu	3
2.1	Perspektywy użytkowników	4
3	Model konceptualny	5
3.1	Definicja zbiorów encji określonych w projekcie (decyzje projektowe) .	5
3.2	Ustalenie związków między encjami i ich typów	9
3.3	Określenie atrybutów i ich dziedzin	12
3.4	Dodatkowe reguły integralnościowe (reguły biznesowe)	12
3.5	Klucze kandydujące i główne (decyzje projektowe)	12
3.6	Schemat ER na poziomie konceptualnym	13
3.7	Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady	14
3.7.1	Pułapka wachlarzowa	14
3.7.2	Pułapka szczelinowa	15
4	Model logiczny	17
4.1	Charakterystyka modelu relacyjnego	17
4.2	Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady	18
4.3	Proces normalizacji – analiza i przykłady	19
4.3.1	Pierwsza postać normalna	19
4.3.2	Druga postać normalna	21
4.3.3	Trzecia postać normalna	21
4.4	Schemat ER na poziomie modelu logicznego	23
4.5	Więzy integralności	24
4.6	Proces denormalizacji – analiza i przykłady	24

5	Faza fizyczna	25
5.1	Projekt transakcji i weryfikacja ich wykonalności	25
5.2	Strojenie bazy danych – dobór indeksów	25
5.3	Skrypt SQL zakładający bazę danych	26
5.4	Przykłady zapytań i poleceń SQL odnoszących się do bazy danych . .	43

1 Zakres i cel projektu (opis założeń funkcjonalnych projektowanej bazy danych)

Celem tego projektu jest utworzenie bazy danych, które spełnia potrzeby klubu lekkoatletycznego. Klub posiada swoją nazwę, adres, czy datę założenia. Klub prowadzi zajęcia z różnych dyscyplin dla kilku grup wiekowych. Członkowie klubu atletycznego mogą liczyć na wsparcie fizjoterapeutów (w razie kontuzji), czy trenerów z odpowiednimi kwalifikacjami do prowadzenia zajęć. Każdy członek klubu posiada zestaw charakterystycznych cech, na przykład: imię, nazwisko, data urodzenia, itp. Do śledzenia poczynąń zawodników, historia wyników członków, reprezentujących dany klub na zawodach lekkoatletycznych jest zapisywana w bazie danych, wraz ze zdobytym miejscem na zawodach, czy nagrodami.

2 Definicja systemu

System powinien spełniać poniższe wymagania:

- Obsługa pracowników
 - Dodawanie, odczyt, modyfikacja i usuwanie danych pracownika:
 - * Dane osobowe
 - * Dane kontaktowe
 - * Dane o wynagrodzeniu
 - * Dane o stanowisku
- Obsługa członków klubu
 - Dodawanie, odczyt, modyfikacja i usuwanie danych członków klubu:
 - * Dane osobowe
 - * Dane zdrowotne
 - * Dane administracyjne
- Obsługa grup zajęciowych
 - Dodawanie, odczyt, modyfikacja i usuwanie danych grup zajęciowych:
 - * Dane dotyczące opłat
 - * Kategoria wiekowa
 - * Liczba członków grupy
- Obsługa zawodów lekkoatletycznych
 - Dodawanie, odczyt, modyfikacja i usuwanie danych zawodów lekkoatletycznych:

- * Terminy zawodów
- * Nagrody
- * Kategoria wiekowa

2.1 Perspektywy użytkowników

Do prawidłowego zarządzania, baza danych powinna korzystać z mechanizmu kontroli dostępu, który działa na zasadzie przyznawaniu ról poszczególnych użytkowników (role-based access control). Każda rola posiada inne uprawnienia (przyznane ze względu na wykonywaną funkcję) do zarządzania bazą danych. Jeden użytkownik może posiadać wiele ról.

- **Administrator** - posiada pełne uprawnienia do zarządzania bazą danych. Administrator może dodawać, usuwać, czy edytować wszystkie dane w bazie danych, jest też uprawniony do ingerowania w strukturę bazy danych.
- **Trener** - posiada uprawnienia do edycji swoich danych, może dodawać nowe grupy zajęciowe oraz modyfikować i usuwać te grupy do których jest przypisany.
- **Fizjoterapeuta** - posiada uprawnienia do modyfikacji swoich danych, posiada wgląd do danych członków klubu, którzy zostali do niego zarejestrowani.
- **Członek klubu** - posiada uprawnienia do zarządzania swoimi danymi, może edytować swoje dane oraz jest uprawniony do przeglądania dostępnych grup zajęciowych, zawodów lekkoatletycznych, trenerów, czy fizjoterapeutów.
- **Użytkownik zewnętrzny** - posiada jedynie uprawnienia do wglądu dostępnych grup zajęciowych, trenerów i fizjoterapeutów w celu promocji klubu.

3 Model konceptualny

3.1 Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

Klub lekkoatletyczny

Określa dany klub lekkoatletyczny

Nazwa atrybutu	Typ	Obowiązkowość pola	Opis
Nr klubu		Klucz główny - obowiązkowy	Unikatowy identyfikator klubu lekkoatletycznego
Nazwa		Obowiązkowy	Nazwa klubu lekkoatletycznego
Adres siedziby		Obowiązkowy	Adres siedziby klubu lekkoatletycznego
Data założenia		Obowiązkowy	Data założenia klubu lekkoatletycznego

Rys. 1: Tabela opisująca dany klub lekkoatletyczny

Pracownik

Określa pracownika klubu lekkoatletycznego

Nazwa atrybutu	Typ	Obowiązkowość pola	Opis
Nr pracownika		Klucz główny - obowiązkowy	Unikatowy identyfikator pracownika
Imie		Obowiązkowy	Imię pracownika
Nazwisko		Obowiązkowy	Nazwisko pracownika
PESEL		Nieobowiązkowy	Numer PESEL pracownika
Data urodzenia		Obowiązkowy	Data urodzenia pracownika
Data zatrudnienia		Obowiązkowy	Data zatrudnienia pracownika
Data zwolnienia		Nieobowiązkowy	Data zwolnienia pracownika
Wynagrodzenie		Obowiązkowy	Wynagrodzenie pracownika
Numer telefonu		Obowiązkowy	Numer telefonu pracownika
Plec		Obowiązkowy	Płeć pracownika
Adres		Obowiązkowy	Adres pracownika
Stanowisko		Obowiązkowy	Stanowisko pracownika

Rys. 2: Tabela opisująca pracownika

Trener

Określa trenera

Nazwa atrybutu	Typ	Obowiązkowość pola	Opis
Kwalifikacje		Obowiązkowy	Kwalifikacje trenerskie
Dyscyplina		Obowiązkowy	Dyscyplina którą zajmuje się trener

Rys. 3: Tabela opisująca trenera

Fizjoterapeuta

Określa fizjoterapeutę

Nazwa atrybutu	Typ	Obowiązkowość pola	Opis
Specjalizacja		Obowiązkowy	Specjalizacja fizjoterapeuty
Kwalifikacje		Obowiązkowy	Kwalifikacje fizjoterapeutyczne

Rys. 4: Tabela opisująca fizjoterapeutę

Członek

Określa członka klubu lekkoatletycznego

Nazwa atrybutu	Typ	Obowiązkowość pola	Opis
Nr członka		Klucz główny - obowiązkowy	Unikatowy identyfikator członka
Imie		Obowiązkowy	Imię członka
Nazwisko		Obowiązkowy	Nazwisko członka
PESEL		Nieobowiązkowy	Numer PESEL członka
Data urodzenia		Obowiązkowy	Data urodzenia członka
Data zapisania		Obowiązkowy	Data zapisania do klubu lekkoatletycznego
Data wypisania		Nieobowiązkowy	Data wypisania z klubu lekkoatletycznego
Badania lekarskie waznosc		Obowiązkowy	Ważność badań lekarskich członka

Rys. 5: Tabela opisująca członka

Dyscyplina

Określa dyscyplinę

Nazwa atrybutu	Typ	Obowiązkowość pola	Opis
Nr dyscypliny		Klucz główny - obowiązkowy	Unikatowy identyfikator dyscypliny
Nazwa		Obowiązkowy	Nazwa dyscypliny
Opis		Obowiązkowy	Opis dyscypliny

Rys. 6: Tabela opisująca dsycplinę

Grupa

Określa grupę zajęciową

Nazwa atrybutu	Typ	Obowiązkowość pola	Opis
Nr grupy		Klucz główny - obowiązkowy	Unikatowy identyfikator grupy zajęciowej
Cena miesiąc		Obowiązkowy	Cena za miesiąc
Maksymalna liczba członków		Obowiązkowy	Maksymalna liczba członków grupy zajęciowej
Kat wiekowa		Obowiązkowy	Kategoria wiekowa

Rys. 7: Tabela opisująca grupę zajęciową

Zawody

Określa zawody lekkoatletyczne

Nazwa atrybutu	Typ	Obowiązkowość pola	Opis
Nr zawodów		Klucz główny - obowiązkowy	Unikatowy identyfikator zawodów
Miejsce		Obowiązkowy	
Termin		Obowiązkowy	Termin odbywania się zawodów
Nagrody		Nieobowiązkowy	Przyznawane nagrody
Kat wiekowa		Obowiązkowy	Kategoria wiekowa

Rys. 8: Tabela opisująca zawody lekkoatletyczne

3.2 Ustalenie związków między encjami i ich typów

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
-------------	----------	---------	----------	-------------

Rys. 9: Tabela opisująca związek Klub Lekkoatletyczny - Pracownik

Jedna jednostka klubu lekkoatletycznego zatrudnia od 0 do wielu pracowników.

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Klub Lekkoatletyczny	1...1	Zatrudnia	0...n	Pracownik
Klub Lekkoatletyczny	1...1	Ma członka	0...n	Członek
Klub Lekkoatletyczny	1...1	Prowadzi grupę	0...n	Grupa
Klub Lekkoatletyczny	1...1	Prowadzi dyscyplinę	0...n	Dyscyplina
Pracownik	1...1	Specjalizacja	1...1	Trener
Pracownik	1...1	Specjalizacja	1...1	Fizjoterapeuta
Trener	1...n	Pracuje z	0...n	Grupa
Grupa	0...n	Trenuje	1...1	Dyscyplina
Członek	0...n	Uczęszcza do	1...n	Grupa
Członek	0...n	Uczestniczy w	0...n	Zawody
Zawody	0...n	Dotyczące	1...n	Dyscyplina

Rys. 10: Tabela opisująca związek Klub Lekkoatletyczny - Członek

Jedna jednostka klubu lekkoatletycznego posiada od 0 do wielu członków.

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
-------------	----------	---------	----------	-------------

Rys. 11: Tabela opisująca związek Klub Lekkoatletyczny - Grupa

Jedna jednostka klubu lekkoatletycznego prowadzi od 0 do wielu grup zajęciowych.

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
-------------	----------	---------	----------	-------------

Rys. 12: Tabela opisująca związek Klub Lekkoatletyczny - Dyscyplina
Jedena jednostka klubu lekkoatletycznego prowadzi od 0 do wielu dyscyplin.

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
-------------	----------	---------	----------	-------------

Rys. 13: Tabela opisująca związek Pracownik - Trener

Pracownik specjalizuje się jako trener.

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
-------------	----------	---------	----------	-------------

Rys. 14: Tabela opisująca związek Pracownik - Fizjoterapeuta

Pracownik specjalizuje się jako fizjoterapeuta.

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
-------------	----------	---------	----------	-------------

Rys. 15: Tabela opisująca związek Trener - Grupa

Przynajmniej jeden trener pracuje z przynajmniej jedną grupą lub nie ma przypisanej żadnej grupy zajęciowej.

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
-------------	----------	---------	----------	-------------

Rys. 16: Tabela opisująca związek Grupa - Dyscyplina

Przynajmniej jedna grupa trenuje jedną dyscyplinę lub nie ma grupy zajęciowej z danej dyscypliny.

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
-------------	----------	---------	----------	-------------

Rys. 17: Tabela opisująca związek Członek - Grupa

Od 0 do wielu członków uczęszcza do przynajmniej jednej grupy zajęciowej

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
-------------	----------	---------	----------	-------------

Rys. 18: Tabela opisująca związek Członek - Zawody

Od 0 do wielu członków uczestniczy w 0 lub wielu zawodach.

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
-------------	----------	---------	----------	-------------

Rys. 19: Tabela opisująca związek Zawody - Dyscyplina

Od 0 do wielu zawodów lekkoatletycznych dotyczy przynajmniej jednej dyscypliny.

3.3 Określenie atrybutów i ich dziedzin

3.4 Dodatkowe reguły integralnościowe (reguły biznesowe)

Kat_wiekowa IN ('U15','U17','U19','U23','Senior')

Kwalifikacje IN ('Trener I','Trener','Trener M','Instruktor')

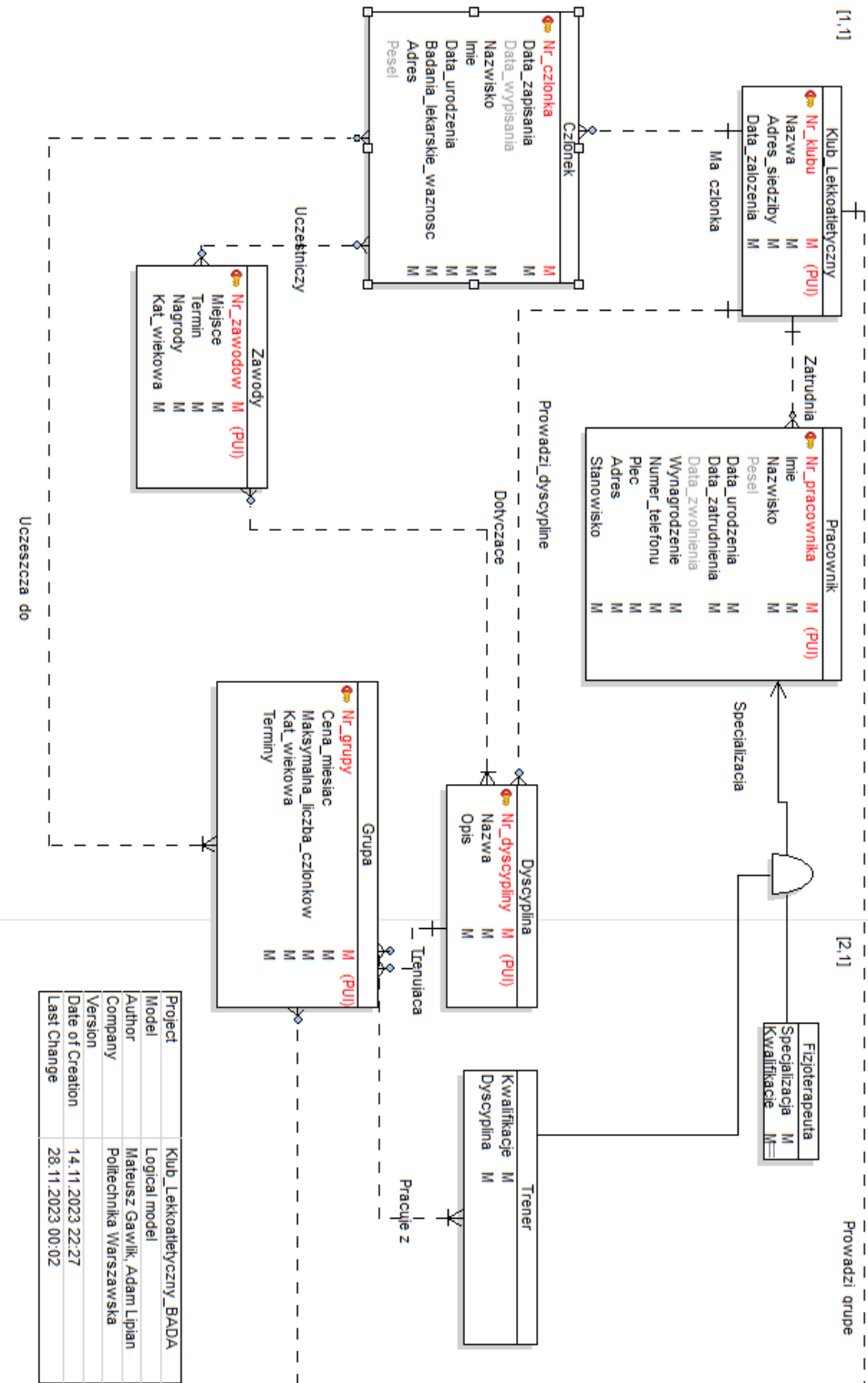
Plec IN ('K','M')

3.5 Klucze kandydujące i główne (decyzje projektowe)

Encja	Klucz główny	Klucze kandydujące	Uzasadnienie
Klub Lekkoatletyczny	Nr klubu	Nazwa	Korzystniejsze jest aby kluczem głównym został Integer
		Adres siedziby	Pole segmentowe nie powinno być kluczem głównym
Pracownik	Nr pracownika	—	—
Członek	Nr członka	—	—
Grupa	Nr grupy	—	—
Zawody	Nr zawodów	—	—
Dyscyplina	Nr dyscypliny	Nazwa	Korzystniejsze jest aby kluczem głównym był Integer
		Opis	Pole Opis jest zbyt długie aby było kluczem głównym

Rys. 20: Tabela z kluczami głównymi oraz kluczami kandydującymi

3.6 Schemat ER na poziomie konceptualnym

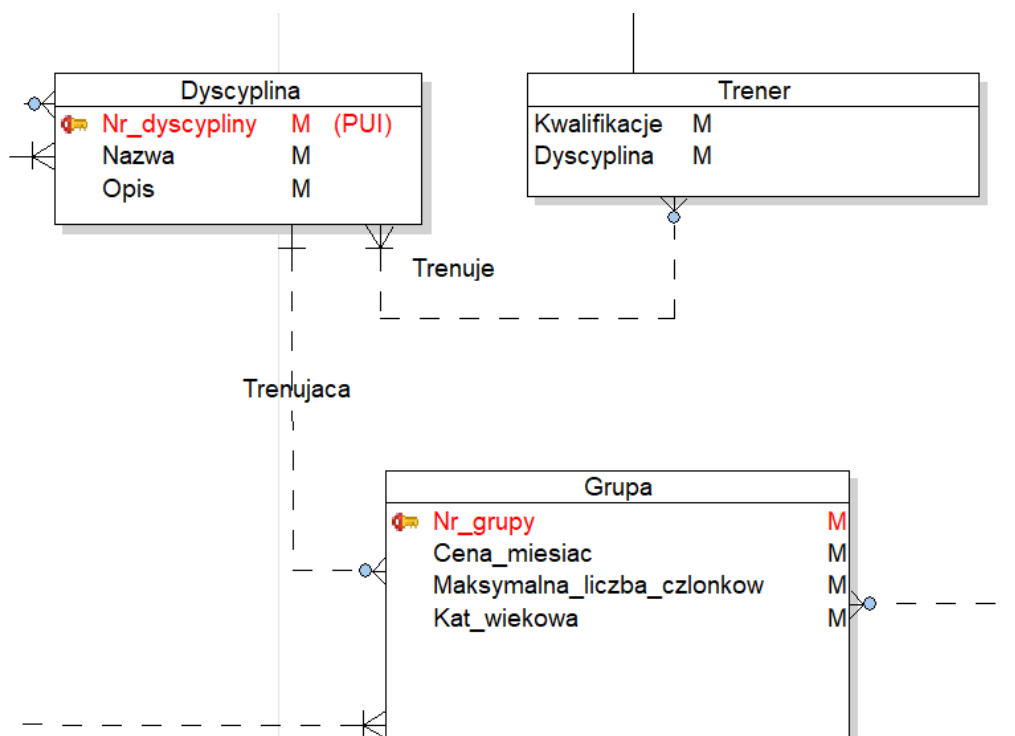


Rys. 21: Model konceptualny

3.7 Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

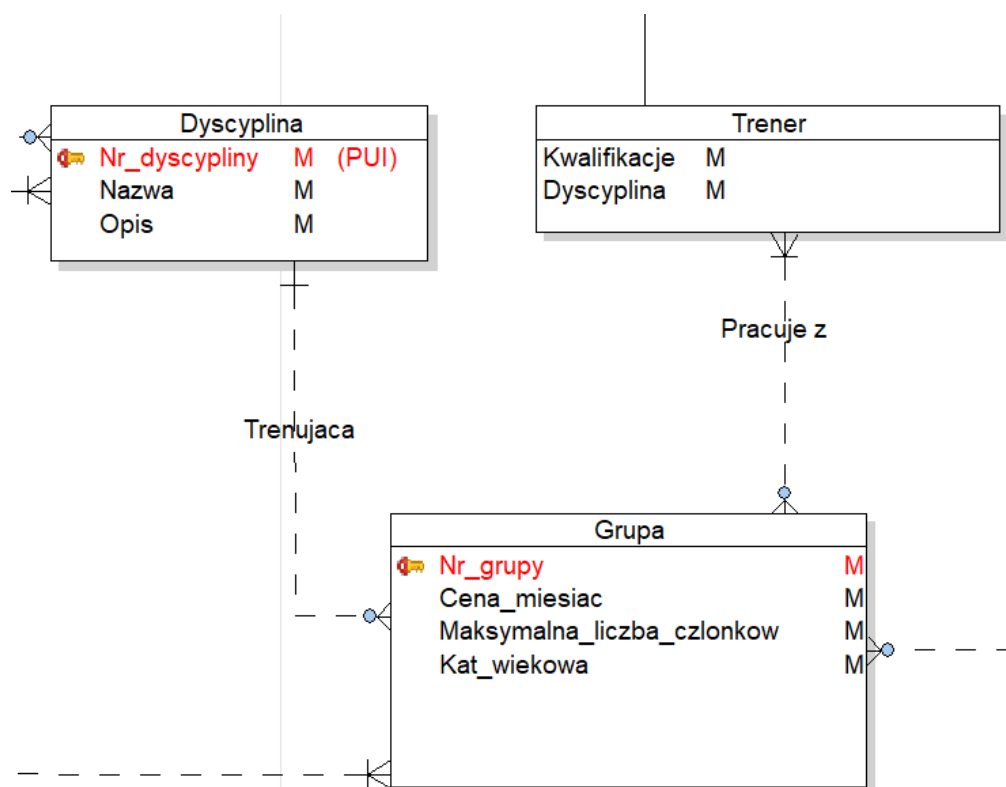
3.7.1 Pułapka wachlarzowa

Podczas tworzenia modelu konceptualnego natrafiliśmy na pułapkę wachlarzową



Rys. 22: Pułapka wachlarzowa

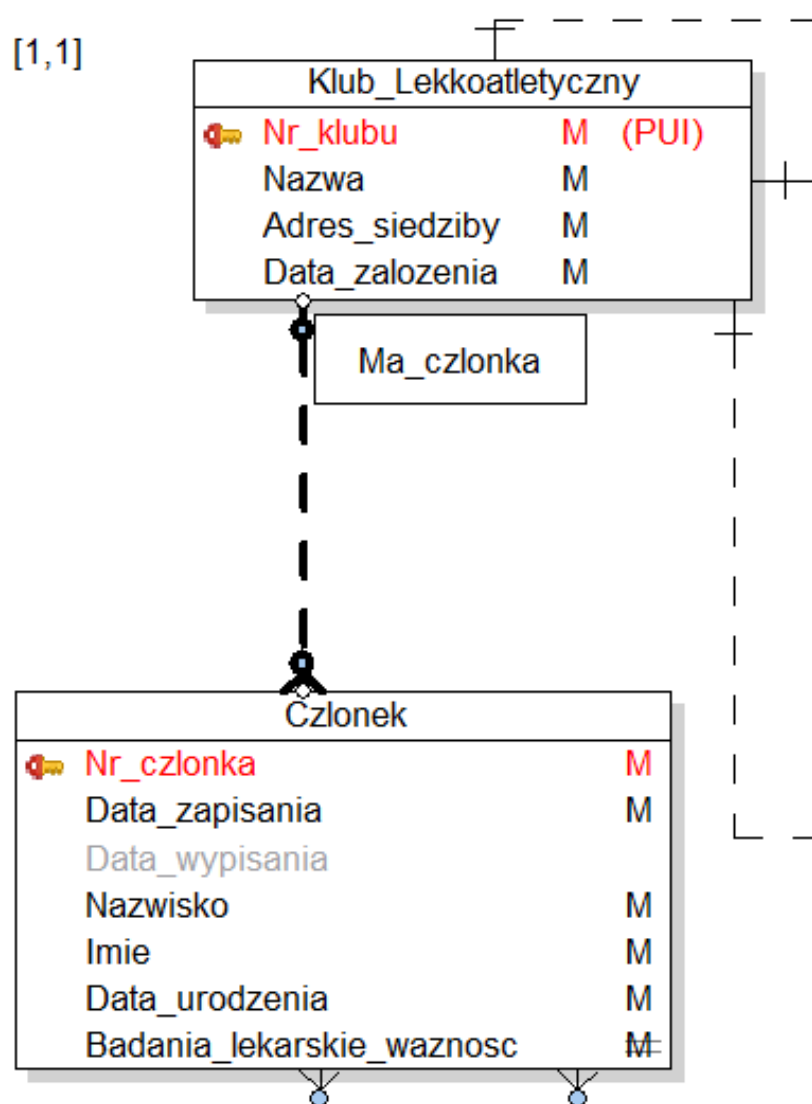
W takiej sytuacji nie jesteśmy w stanie przypisać trenera do grupy. Aby naprawić ten błąd utworzyliśmy związek między trenerem a grupą oraz usunęliśmy związek pomiędzy trenerem a dyscypliną (rys. 23). W ten sposób trener jesteśmy w stanie jednoznacznie określić za grupę jakiej dyscypliny i kategorii wiekowej odpowiedzialny jest trener.



Rys. 23: Poprawiony błąd

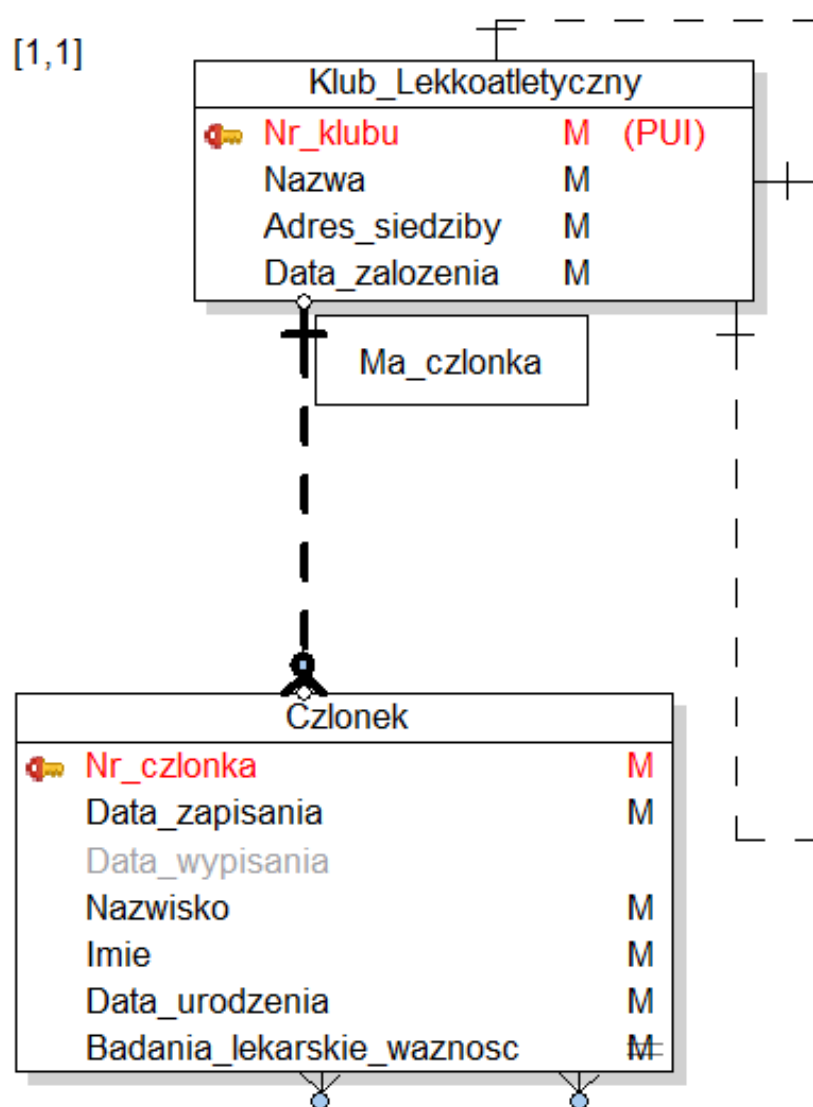
3.7.2 Pułapka szczelinowa

W trakcie modelowania napotkaliśmy również na pułapkę szczelinową.



Rys. 24: Pułapka szczelinowa

Przy takim powiązaniu może istnieć członek, który nie jest zapisany do żadnego klubu lekkoatletycznego, co narusza spójność bazy i może prowadzić do anomalii w procesie wyszukiwania danych. Aby temu zaradzić wprowadziliśmy związek 1..1 do 0:n, dzięki czemu Członek musi mieć przypisany klub.



Rys. 25: Poprawiony błąd

4 Model logiczny

4.1 Charakterystyka modelu relacyjnego

Aby uzyskać model relacyjny, najpierw należy usunąć związki wiele do wielu. Zastępujemy każdy z nich dwoma związkami jeden do wielu. Oprócz tego, encje zastępujemy relacjami – tabelami o nazwach będących liczbą mnogą od nazwy encji (na przykład z encji Pracownik tworzymy tabelę Pracownicy). Związki jeden do wielu realizuje się poprzez dodanie po stronie tabeli “wielu” klucza obcego, którego wartości pochodzą od klucza głównego tabeli “jeden”.

4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

Problem związków jeden do wielu został rozwiązany w procesie konwersji modelu konceptualnego na logiczny. Poniżej znajdują się tabele łączące związki wiele do wielu:

Tabela 1	Tabela 2	Tabela łącząca
Członkowie	Grupy	Członkowie_grup
Zawody	Dyscypliny	Konkurencje_na_zawodach
Członkowie	Konkurencje_na_zawodach	Uczestnicy_zawodow
Trenerzy	Grupy	Trenerzy_grup
Terminy	Grupy	Terminy_grup

Tabela 1: Tabele łączące

W każdej tabeli łączącej znajdują się klucze główne relacji wchodzących w skład związku, jak również dodatkowe informacje. Tabele łączące i ich atrybuty:

Nazwa atrybutu	Typ	Obowiązkowość [M,NM]	Klucz
Nr_czlonka	Integer	M	PFK
Nr_grupy	Integer	M	PFK

Tabela 2: Tabela Członkowie_grup

Nazwa atrybutu	Typ	Obowiązkowość [M,NM]	Klucz
Nr_czlonka	Integer	M	PFK
Nr_dyscypliny	Integer	M	FK
Nr_zawodow	Integer	M	FK
Kat_wiekowa	Kat_wiekowaD	M	FK
Wyniki	Varchar2(30)	NM	-
Miejsce	Integer	NM	-

Tabela 3: Tabela Uczestnicy_zawodow

Nazwa atrybutu	Typ	Obowiązkowość [M,NM]	Klucz
Nr_zawodow	Integer	M	PFK
Nr_dyscypliny	Integer	M	PFK
Kat_wiekowa	Kat_wiekowaD	M	PK

Tabela 4: Tabela Konkurencje_na_zawodach

Nazwa atrybutu	Typ	Obowiązkowość [M,NM]	Klucz
Nr_trenera	Integer	M	PFK
Nr_grupy	Integer	M	PFK

Tabela 5: Tabela Trenerzy_grup

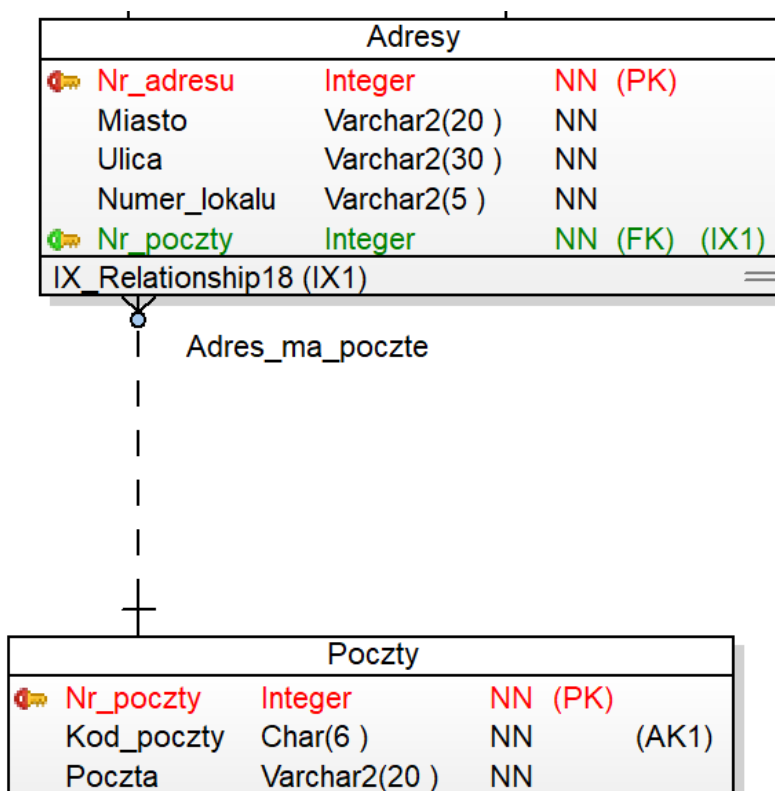
Nazwa atrybutu	Typ	Obowiązkowość [M,NM]	Klucz
Nr_grupy	Integer	M	PFK
Nr_terminu	Integer	M	PFK

Tabela 6: Tabela Terminy_grup

4.3 Proces normalizacji – analiza i przykłady


4.3.1 Pierwsza postać normalna

Aby uzyskać pierwszą postać normalną, wszystkie pola muszą być atomowe oraz nie mogą występować powtarzające się grupy. W naszym modelu konceptualnym w kilku przypadkach tak nie jest. Klub_lekkoatletyczny, Pracownicy, Czlonkowie czy Zawody posiadają adresy, które są polami segmentowymi składającymi się z miasta, ulicy, numeru lokalu i poczty. Z tego powodu utworzyliśmy nową tabelę **Adresy**. Ponieważ atrybut poczty również jest polem segmentowym, postanowiliśmy również utworzyć tabelę **Poczty** zawierającą informację o kodzie pocztowym i Mieście, w którym się znajduje.





Rys. 26: Tabele Poczty i Adresy powiązane relacją 1..1 do 0..n

Pole Stanowisko (Nazwa stanowiska, opis) oraz Wynagrodzenia (Data wynagrodzenia, wynagrodzenie brutto) w tabeli Pracownicy również nie są polami atomowymi. Zastąpiliśmy je tabelami Stanowiska oraz Wynagrodzenia.


Stanowiska		
 Nr_stanowiska	Integer	NN (PK)
Nazwa	Varchar2(20)	NN
Opis	Varchar2(400)	NN

Rys. 27: Tabela stanowiska

Wynagrodzenia		
 Nr_wynagrodzenia	Integer	NN (PK)
Data_wynagrodzenia	Date	NN
Wynagrodzenie_brutto	Number(10,2)	NN
 Nr_pracownika	Integer	NN (FK) (IX1)





Rys. 28: Tabela Wynagrodzenia (Pracownik może mieć kilka wynagrodzeń)

Pole Terminy tabeli Grupy jest polem segmentowym (Dzień tygodnia, godzina rozpoczęcia zajęć, godzina zakończenia zajęć) oraz wielowartościowym (jedna grupa może mieć kilka zajęć w tygodniu). Zastąpiliśmy to pole tabelą Terminy (wspomnianą już w kontekście związków wiele do wielu - sekcja 4.1)

Terminy		
 Nr_terminu	Integer	NN (PK)
Dzien_tygodnia	Dzien_tygodniaD	NN
Godzina_od	Varchar2(5)	NN
Godzina_do	Varchar2(5)	NN

Rys. 29: Tabela Terminy

Pole Nagrody w tabeli Zawody są kolejnym przykładem pola segmentowego wielowartościowego. Każda konkurencja ma osobne nagrody dla zawodników zajmujących miejsca od I do III. Z tego powodu zastąpiliśmy to pole tabelą Nagrody.

Nagrody			
 Nr_nagrody	Integer	NN (PK) (AK0)	
I_miejsce	Number(10,2)	NN	
II_miejsce	Number(10,2)	NN	
III_miejsce	Number(10,2)	NN	
 Nr_zawodow	Integer	NN (FK) (IX1)	
 Nr_dyscypliny	Integer	NN (FK) (IX1)	
 Kat_wiekowa	Kat_wiekowaD	NN (FK) (IX1)	
IX_Relationship9 (IX1)			

Rys. 30: Tabela Nagrody

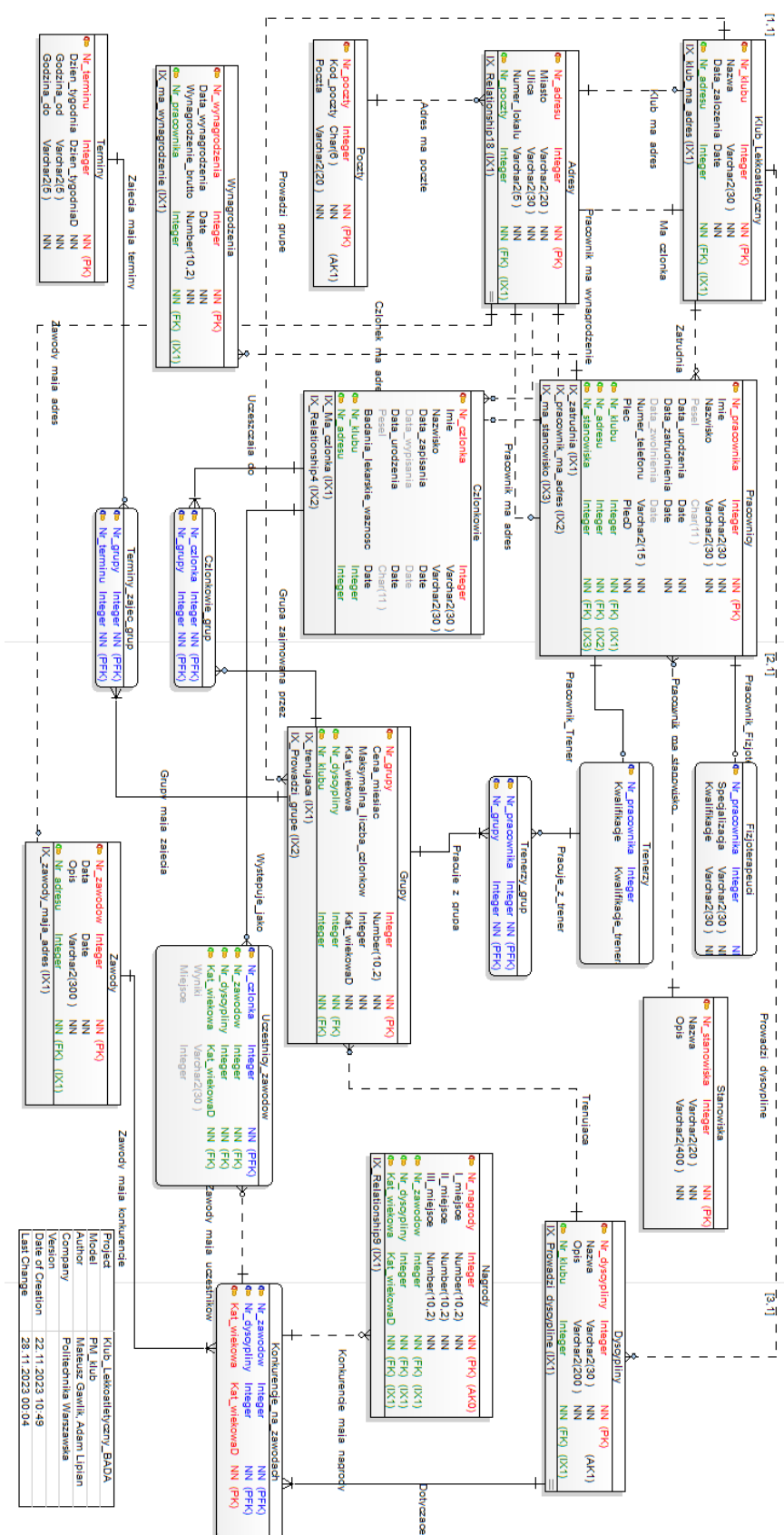
4.3.2 Druga postać normalna

Aby zachodziła, musi być w pierwszej postaci normalnej i wszystkie klucze kandydujące muszą być kluczami prostymi. W naszej bazie ten warunek zachodzi bez wnoszenia poprawek.

4.3.3 Trzecia postać normalna

Aby baza była w trzeciej postaci normalnej, musi być w drugiej postaci normalnej i żaden atrybut niekluczowy nie może zależeć od innego atrybutu niekluczowego. Każdy niekluczowy atrybut zależy tylko od klucza głównego. W naszej bazie ten warunek jest już spełniony, więc jest ona w trzeciej postaci normalnej.

4.4 Schemat ER na poziomie modelu logicznego



4.5 Więzy integralności

- Każda tabela posiada Klucz główny, który jest unikalny i niepusty. Klucz główny identyfikuje jednoznacznie każdy wiersz w tabeli.
- Więzy klucza obcego ustanawiają relacje między tabelami. Wartości w kolumnie klucza obcego muszą odpowiadać wartościom w kolumnie klucza głównego w innej tabeli. Utrzymują one spójność relacyjną między różnymi tabelami w bazie danych.
- Określenie dopuszczonych wartości niektórych pól poprzez wprowadzenie dziedzin (np. CHECK (Plec IN ('M','K'))).
- Zaznaczenie konieczności wystąpienia danego pola. Istotne, aby nie dochodziło do pułapek szczelinowych.

4.6 Proces denormalizacji – analiza i przykłady

Proces denormalizacji to ustalenie, czy wydajność systemu można poprawić, wprowadzając kontrolowaną redundancję poprzez złagodzenie zakresu stosowania reguł normalizacji. Denormalizacja przejawia się w przypadku tabeli **Klub_lekkoatletyczny**. Zawiera się w niej dokładnie jeden rekord z danymi o klubie i jego numerze identyfikacyjnym (Klucz główny), który tworzy klucz obcy w tabelach takich jak Pracownicy (klub ma pracowników), Członkowie (klub ma członków), Grupy (klub ma grupy) i Dyscypliny (klub oferuje dyscypliny). Redundancja w tym przypadku nie jest duża (powtarza się jedynie numer identyfikacyjny klubu), lecz konieczna z perspektywy przynależności danych obiektów do klubu.

5 Faza fizyczna

5.1 Projekt transakcji i weryfikacja ich wykonalności

Transakcja	Wykorzystane tabele	Wykonalne?
Tworzenie, modyfikowanie, odczytywanie, usuwanie pracowników	Pracownicy, Adresy, Wynagrodzenia, Stanowiska, Fizjoterapeuci, Trenerzy, Klub_Lekkoatletyczny	Tak
Tworzenie, usuwanie, odczytywanie, modyfikowanie adresów	Adresy, Poczty	Tak
Tworzenie, usuwanie, odczytywanie, modyfikowanie grup	Grupy, Dyscypliny, Klub_Lekkoatletyczny	Tak
Tworzenie, modyfikowanie, odczytywanie, usuwanie dyscyplin	Dyscypliny, Klub_Lekkoatletyczny	Tak
Tworzenie, modyfikowanie, odczytywanie, usuwanie zawodów	Zawody, Adres	Tak
Tworzenie, modyfikowanie, odczytywanie, usuwanie nagród	Nagrody	Tak
Tworzenie, modyfikowanie, odczytywanie, usuwanie terminów zajęć	Terminy	Tak
Zapisywanie członków klubu do grup	Czlonkowie_grup	Tak
Przypisywanie terminów zajęć grupom	Terminy_zajęć_grup	Tak
Zapisanie uczestników na zawody	Uczestnicy_zawodów, Dyscypliny	Tak
Przypisanie trenerów do grup	Trenerzy_grup	Tak

5.2 Strojenie bazy danych – dobór indeksów

Adresy - poczty

-CREATE INDEX IX_adres_ma_poczte ON Adresy (Nr_poczty)

Nagrody - Konkurencje_na_zawodach

-CREATE INDEX IX_Nagrody_za_konkurencje ON Nagrody (Nr_zawodow, Nr_dyscypliny, Kat_wiekowa)

Wynagrodzenia - pracownicy

-CREATE INDEX XI_Pracownik_dostaje_wynagrodzenie ON Wynagrodzenia (Nr_pracownika)

Klub_lekkoatletyczny - adres

-CREATE INDEX IX_klub_ma_adres ON Klub_Lekkoatletyczny (Nr_adresu)

Pracownicy - klub, adres, stanowisko

-CREATE INDEX IX_zatrudnia ON Pracownicy (Nr_klubu)

-CREATE INDEX IX_pracownik_ma_adres ON Pracownicy (Nr_adresu)

-CREATE INDEX IX_ma_stanowisko ON Pracownicy (Nr_stanowiska)

Dyscypliny - klub

-CREATE INDEX IX_Prowadzi_dyscypline ON Dyscypliny (Nr_klubu)

Grupy - dyscyplina, klub

-CREATE INDEX IX_trenujaca ON Grupy (Nr_dyscypliny)

-CREATE INDEX IX_Prowadzi_grupe ON Grupy (Nr_klubu)

Członkowie - klub, adres

-CREATE INDEX IX_Ma_czlonka ON Czlonkowie (Nr_klubu)

-CREATE INDEX IX_Czlonek_ma_adres ON Czlonkowie (Nr_adresu)

Zawody - adres

-CREATE INDEX IX_zawody_maja_adres ON Zawody (Nr_adresu)

Uczestnicy zawodow - konkurencje na zawodach

-CREATE INDEX IX_biora_udzial ON Uczestnicy_zawodow (Nr_zawodow, Nr_dyscypliny, Kat_wiekowa)

5.3 Skrypt SQL zakładający bazę danych

```
/*
Created: 22.11.2023
Modified: 26.11.2023
Project: Klub_Lekkoatletyczny_BADA
Model: PM_klub
Company: Politechnika Warszawska
Author: Mateusz Gawlik, Adam Lipian
Database: Oracle 18c
*/
```

-- Create sequences section -----

```
CREATE SEQUENCE Adresy_seq1
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
NOCACHE
/
```

```
CREATE SEQUENCE Nagrody_seq1
INCREMENT BY 1
START WITH 1
NOMAXVALUE
```

```

NOMINVALUE
NOCACHE
/

CREATE SEQUENCE Terminy_seq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  NOCACHE
/

CREATE SEQUENCE Stanowisko_seq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  NOCACHE
/

CREATE SEQUENCE Poczty_seq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  NOCACHE
/

CREATE SEQUENCE Pracownicy_seq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE Fizjo_seq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE
  CACHE 20
/

CREATE SEQUENCE Dyscypliny_seq1
  INCREMENT BY 1
  START WITH 1
  NOMAXVALUE
  NOMINVALUE

```

```

    CACHE 20
/

CREATE SEQUENCE Czlonkowie_seq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE Grupy_seq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE Wynagrodzenia_seq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

CREATE SEQUENCE Zawody_seq1
    INCREMENT BY 1
    START WITH 1
    NOMAXVALUE
    NOMINVALUE
    CACHE 20
/

-- Create tables section -----

-- Table Adresy

CREATE TABLE Adresy(
    Nr_adresu Integer NOT NULL,
    Miasto Varchar2(20 ) NOT NULL,
    Ulica Varchar2(30 ) NOT NULL,
    Numer_lokalu Varchar2(5 ) NOT NULL,
    Nr_poczty Integer NOT NULL
)
/

-- Create indexes for table Adresy

```

```

CREATE INDEX XI_adres_ma_poczte ON Adresy (Nr_poczty)
/

-- Add keys for table Adresy

ALTER TABLE Adresy ADD CONSTRAINT PK_Adresy PRIMARY KEY (Nr_adresu)
/

-- Table and Columns comments section

COMMENT ON COLUMN Adresy.Nr_adresu IS 'Unikatowy identyfikator adresu'
/
COMMENT ON COLUMN Adresy.Miasto IS 'Miasto'
/
COMMENT ON COLUMN Adresy.Ulica IS 'Ulica'
/
COMMENT ON COLUMN Adresy.Numer_lokalu IS 'Numer lokalu'
/

-- Table Poczty

CREATE TABLE Poczty(
    Nr_poczty Integer NOT NULL,
    Kod_poczty Char(6 ) NOT NULL,
    Poczta Varchar2(20 ) NOT NULL
)
/

-- Add keys for table Poczty

ALTER TABLE Poczty ADD CONSTRAINT PK_Poczty PRIMARY KEY (Nr_poczty)
/

ALTER TABLE Poczty ADD CONSTRAINT Kod_poczty UNIQUE (Kod_poczty)
/

-- Table and Columns comments section

COMMENT ON COLUMN Poczty.Nr_poczty IS 'Unikalny identyfikator poczty'
/
COMMENT ON COLUMN Poczty.Kod_poczty IS 'Kod poczty w formacie XX-XXX'
/
COMMENT ON COLUMN Poczty.Poczta IS 'Lokalizacja poczty'
/

-- Table Stanowiska

CREATE TABLE Stanowiska(

```

```

    Nr_stanowiska Integer NOT NULL,
    Nazwa Varchar2(20 ) NOT NULL,
    Opis Varchar2(400 ) NOT NULL
)
/

-- Add keys for table Stanowiska

ALTER TABLE Stanowiska ADD CONSTRAINT PK_Stalowiska PRIMARY KEY (Nr_stanowiska)
/

ALTER TABLE Stanowiska ADD CONSTRAINT Nazwa UNIQUE (Nazwa)
/

-- Table and Columns comments section

COMMENT ON COLUMN Stanowiska.Nr_stanowiska IS 'Unikalny numer stanowiska'
/
COMMENT ON COLUMN Stanowiska.Nazwa IS 'Nazwa stanowiska'
/
COMMENT ON COLUMN Stanowiska.Opis IS 'Opis stanowiska'
/

-- Table Terminy

CREATE TABLE Terminy(
    Nr_terminu Integer NOT NULL,
    Dzień_tygodnia Varchar2(3 ) NOT NULL
        CHECK (Dzień_tygodnia IN ('PN','WT','SR','CZW','PT','SB','ND')),
    Godzina_od Varchar2(5 ) NOT NULL,
    Godzina_do Varchar2(5 ) NOT NULL
)
/

-- Add keys for table Terminy

ALTER TABLE Terminy ADD CONSTRAINT PK_Terminy PRIMARY KEY (Nr_terminu)
/

-- Table and Columns comments section

COMMENT ON COLUMN Terminy.Nr_terminu IS 'Numer terminu'
/
COMMENT ON COLUMN Terminy.Dzień_tygodnia IS 'Dzień tygodnia'
/
COMMENT ON COLUMN Terminy.Godzina_od IS 'Godzina rozpoczęcia w formacie XX:XX'
/
COMMENT ON COLUMN Terminy.Godzina_do IS 'Godzina zakończenia w formacie XX:XX'
/

```

```

-- Table Nagrody

CREATE TABLE Nagrody(
    Nr_nagrody Integer NOT NULL,
    I_miejsce Number(10,2) NOT NULL,
    II_miejsce Number(10,2) NOT NULL,
    III_miejsce Number(10,2) NOT NULL,
    Nr_zawodow Integer NOT NULL,
    Nr_dyscypliny Integer NOT NULL,
    Kat_wiekowa Varchar2(6 ) NOT NULL
        CHECK (Kat_wiekowa IN ('U15','U17','U19','U23','Senior'))
)
/

-- Create indexes for table Nagrody

CREATE INDEX XI_Nagrody_za_konkurencje ON Nagrody (Nr_zawodow,Nr_dyscypliny,Kat_wiekowa)
/

-- Add keys for table Nagrody

ALTER TABLE Nagrody ADD CONSTRAINT Nagrody_pk PRIMARY KEY (Nr_nagrody)
/

ALTER TABLE Nagrody ADD CONSTRAINT PK_Nagrody UNIQUE (Nr_nagrody)
/

-- Table and Columns comments section

COMMENT ON TABLE Nagrody IS 'Nagrody za zawody z poszczególnych dyscyplin'
/
COMMENT ON COLUMN Nagrody.Nr_nagrody IS 'Unikalny identyfikator nagrody'
/
COMMENT ON COLUMN Nagrody.I_miejsce IS 'Nagroda za I miejsce'
/
COMMENT ON COLUMN Nagrody.II_miejsce IS 'Nagroda za II miejsce'
/
COMMENT ON COLUMN Nagrody.III_miejsce IS 'Nagroda za III miejsce'
/

-- Table Wynagrodzenia

CREATE TABLE Wynagrodzenia(
    Nr_wynagrodzenia Integer NOT NULL,
    Data_wynagrodzenia Date NOT NULL,
    Wynagrodzenie_brutto Number(10,2) NOT NULL,
    Nr_pracownika Integer NOT NULL
)

```

```

/

-- Create indexes for table Wynagrodzenia

CREATE INDEX IX_Pracownik_dostaje_wynagrodzenie ON Wynagrodzenia (Nr_pracownika)
/

-- Add keys for table Wynagrodzenia

ALTER TABLE Wynagrodzenia ADD CONSTRAINT PK_Wynagrodzenia PRIMARY KEY (Nr_wynagrodzenia)
/

-- Table and Columns comments section

COMMENT ON COLUMN Wynagrodzenia.Nr_wynagrodzenia IS 'Unikalny identyfikator numeru wynagr
/
COMMENT ON COLUMN Wynagrodzenia.Data_wynagrodzenia IS 'Data ważności umowy'
/
COMMENT ON COLUMN Wynagrodzenia.Wynagrodzenie_brutto IS 'Wynagrodzenie_brutto'
/

-- Table Klub_Lekkoatletyczny

CREATE TABLE Klub_Lekkoatletyczny(
    Nr_klubu Integer NOT NULL,
    Nazwa Varchar2(30 ) NOT NULL,
    Data_zalozenia Date NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Klub_Lekkoatletyczny

CREATE INDEX IX_klub_ma_adres ON Klub_Lekkoatletyczny (Nr_adresu)
/

-- Add keys for table Klub_Lekkoatletyczny

ALTER TABLE Klub_Lekkoatletyczny ADD CONSTRAINT Klub_lekkoatletyczny_PK PRIMARY KEY (Nr_k
/

-- Table Pracownicy

CREATE TABLE Pracownicy(
    Nr_pracownika Integer NOT NULL,
    Imie Varchar2(30 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Pesel Char(11 ),
    Data_urodzenia Date NOT NULL,

```



```

    Data_zatrudnienia Date NOT NULL,
    Data_zwolnienia Date,
    Numer_telefonu Varchar2(15 ) NOT NULL,
    Plec Char(1 ) NOT NULL
        CHECK (Plec IN ('K','M')),
    Nr_klubu Integer NOT NULL,
    Nr_adresu Integer NOT NULL,
    Nr_stanowiska Integer NOT NULL
)
/

-- Create indexes for table Pracownicy

CREATE INDEX IX_zatrudnia ON Pracownicy (Nr_klubu)
/

CREATE INDEX IX_pracownik_ma_adres ON Pracownicy (Nr_adresu)
/

CREATE INDEX IX_ma_stanowisko ON Pracownicy (Nr_stanowiska)
/

-- Add keys for table Pracownicy

ALTER TABLE Pracownicy ADD CONSTRAINT Pracownicy_PK PRIMARY KEY (Nr_pracownika)
/

-- Table Trenerzy

CREATE TABLE Trenerzy(
    Nr_pracownika Integer NOT NULL,
    Kwalifikacje Varchar2(20 ) NOT NULL
        CHECK (Kwalifikacje IN ('Trener I','Trener','Trener M','Instruktor'))
)
/

-- Add keys for table Trenerzy

ALTER TABLE Trenerzy ADD CONSTRAINT Unique_Identifier1 PRIMARY KEY (Nr_pracownika)
/

-- Table Fizjoterapeuci

CREATE TABLE Fizjoterapeuci(
    Nr_pracownika Integer NOT NULL,
    Specjalizacja Varchar2(30 ) NOT NULL,
    Kwalifikacje Varchar2(30 ) NOT NULL
)
/

```

```

-- Add keys for table Fizjoterapeuci

ALTER TABLE Fizjoterapeuci ADD CONSTRAINT Unique_Identifier2 PRIMARY KEY (Nr_pracownika)
/

-- Table Dyscypliny

CREATE TABLE Dyscypliny(
    Nr_dyscypliny Integer NOT NULL,
    Nazwa Varchar2(30 ) NOT NULL,
    Opis Varchar2(200 ) NOT NULL,
    Nr_klubu Integer NOT NULL
)
/

-- Create indexes for table Dyscypliny

CREATE INDEX IX_Prowadzi_dyscypline ON Dyscypliny (Nr_klubu)
/

-- Add keys for table Dyscypliny

ALTER TABLE Dyscypliny ADD CONSTRAINT Dyscyplina_PK PRIMARY KEY (Nr_dyscypliny)
/

ALTER TABLE Dyscypliny ADD CONSTRAINT Nazwa_dyscypliny UNIQUE (Nazwa)
/

-- Table Grupy

CREATE TABLE Grupy(
    Nr_grupy Integer NOT NULL,
    Cena_miesiac Number(10,2) NOT NULL,
    Maksymalna_liczba_czlonkow Integer NOT NULL,
    Kat_wiekowa Varchar2(6 ) NOT NULL
        CHECK (Kat_wiekowa IN ('U15','U17','U19','U23','Senior')),
    Nr_dyscypliny Integer NOT NULL,
    Nr_klubu Integer NOT NULL
)
/

-- Create indexes for table Grupy

CREATE INDEX IX_trenujaca ON Grupy (Nr_dyscypliny)
/

CREATE INDEX IX_Prowadzi_grupe ON Grupy (Nr_klubu)
/

```

```

-- Add keys for table Grupy

ALTER TABLE Grupy ADD CONSTRAINT Grupa_PK PRIMARY KEY (Nr_grupy)
/

-- Table Czlonkowie

CREATE TABLE Czlonkowie(
    Nr_czlonka Integer NOT NULL,
    Imie Varchar2(30 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Data_zapisania Date NOT NULL,
    Data_wypisania Date,
    Data_urodzenia Date NOT NULL,
    Badania_lekarskie_waznosc Date NOT NULL,
    Nr_klubu Integer NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Czlonkowie

CREATE INDEX IX_Ma_czlonka ON Czlonkowie (Nr_klubu)
/

CREATE INDEX IX_Czlonka_ma_adres ON Czlonkowie (Nr_adresu)
/

-- Add keys for table Czlonkowie

ALTER TABLE Czlonkowie ADD CONSTRAINT Czlonka_PK PRIMARY KEY (Nr_czlonka)
/

-- Table Zawody

CREATE TABLE Zawody(
    Nr_zawodow Integer NOT NULL,
    Data Date NOT NULL,
    Opis Varchar2(300 ) NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Zawody

CREATE INDEX IX_zawody_maja_adres ON Zawody (Nr_adresu)
/

```

```

-- Add keys for table Zawody

ALTER TABLE Zawody ADD CONSTRAINT Zawody_PK PRIMARY KEY (Nr_zawodow)
/

-- Table and Columns comments section

COMMENT ON COLUMN Zawody.Data IS 'Data zawodów'
/
COMMENT ON COLUMN Zawody.Opis IS 'Opis zawodów'
/

-- Table Uczestnicy_zawodow

CREATE TABLE Uczestnicy_zawodow(
    Nr_czlonka Integer NOT NULL,
    Wyniki Varchar2(30 ),
    Miejsce Integer,
    Nr_zawodow Integer NOT NULL,
    Nr_dyscypliny Integer NOT NULL,
    Kat_wiekowa Varchar2(6 ) NOT NULL
        CHECK (Kat_wiekowa IN ('U15','U17','U19','U23','Senior'))
)
/

-- Create indexes for table Uczestnicy_zawodow

CREATE INDEX IX_biora_udzial ON Uczestnicy_zawodow (Nr_zawodow,Nr_dyscypliny,Kat_wiekowa)
/

-- Table and Columns comments section

COMMENT ON TABLE Uczestnicy_zawodow IS 'Uczestnik zawodów'
/
COMMENT ON COLUMN Uczestnicy_zawodow.Wyniki IS 'Wyniki z zawodów'
/
COMMENT ON COLUMN Uczestnicy_zawodow.Miejsce IS 'Miejsce w zawodach'
/

-- Table Terminy_zajec_grup

CREATE TABLE Terminy_zajec_grup(
    Nr_grupy Integer NOT NULL,
    Nr_terminu Integer NOT NULL
)
/

-- Add keys for table Terminy_zajec_grup

```

```

ALTER TABLE Terminy_zajec_grup ADD CONSTRAINT PK_Terminy_zajec_grup PRIMARY KEY (Nr_grupy
/

-- Table Czlonkowie_grup

CREATE TABLE Czlonkowie_grup(
    Nr_czlonka Integer NOT NULL,
    Nr_grupy Integer NOT NULL
)
/

-- Add keys for table Czlonkowie_grup

ALTER TABLE Czlonkowie_grup ADD CONSTRAINT PK_Czlonkowie_grup PRIMARY KEY (Nr_czlonka,Nr_g
/

-- Table Trenerzy_grup

CREATE TABLE Trenerzy_grup(
    Nr_pracownika Integer NOT NULL,
    Nr_grupy Integer NOT NULL
)
/

-- Add keys for table Trenerzy_grup

ALTER TABLE Trenerzy_grup ADD CONSTRAINT PK_Trenerzy_grup PRIMARY KEY (Nr_pracownika,Nr_g
/

-- Table Konkurencje_na_zawodach

CREATE TABLE Konkurencje_na_zawodach(
    Nr_zawodow Integer NOT NULL,
    Nr_dyscypliny Integer NOT NULL,
    Kat_wiekowa Varchar2(6 ) NOT NULL
        CONSTRAINT Kat_wiekowaCR CHECK (Kat_wiekowa IN ('U15','U17','U19','U23','Senior'))
        CHECK (Kat_wiekowa IN ('U15','U17','U19','U23','Senior'))
)
/

-- Add keys for table Konkurencje_na_zawodach

ALTER TABLE Konkurencje_na_zawodach ADD CONSTRAINT PK_Konkurencje_na_zawodach PRIMARY KEY
/

-- Trigger for sequence Pracownicy_seq1 for column Nr_pracownika in table Pracownicy ----
CREATE OR REPLACE TRIGGER ts_Pracownicy_Pracownicy_seq1 BEFORE INSERT
ON Pracownicy FOR EACH ROW
BEGIN

```

```

        :new.Nr_pracownika := Pracownicy_seq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Pracownicy_Pracownicy_seq1 AFTER UPDATE OF Nr_pracownika
ON Pracownicy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_pracownika in table Pracownicy
END;
/

-- Trigger for sequence Dyscypliny_seq1 for column Nr_dyscypliny in table Dyscypliny ----
CREATE OR REPLACE TRIGGER ts_Dyscypliny_Dyscypliny_seq1 BEFORE INSERT
ON Dyscypliny FOR EACH ROW
BEGIN
    :new.Nr_dyscypliny := Dyscypliny_seq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Dyscypliny_Dyscypliny_seq1 AFTER UPDATE OF Nr_dyscypliny
ON Dyscypliny FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_dyscypliny in table Dyscypliny
END;
/

-- Trigger for sequence Grupy_seq1 for column Nr_grupy in table Grupy -----
CREATE OR REPLACE TRIGGER ts_Grupy_Grupy_seq1 BEFORE INSERT
ON Grupy FOR EACH ROW
BEGIN
    :new.Nr_grupy := Grupy_seq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Grupy_Grupy_seq1 AFTER UPDATE OF Nr_grupy
ON Grupy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_grupy in table Grupy as it uses
END;
/

-- Trigger for sequence Czlonkowie_seq1 for column Nr_czlonka in table Czlonkowie -----
CREATE OR REPLACE TRIGGER ts_Czlonkowie_Czlonkowie_seq1 BEFORE INSERT
ON Czlonkowie FOR EACH ROW
BEGIN
    :new.Nr_czlonka := Czlonkowie_seq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Czlonkowie_Czlonkowie_seq1 AFTER UPDATE OF Nr_czlonka
ON Czlonkowie FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_czlonka in table Czlonkowie as

```

```

END;
/

-- Trigger for sequence Zawody_seq1 for column Nr_zawodow in table Zawody -----
CREATE OR REPLACE TRIGGER ts_Zawody_Zawody_seq1 BEFORE INSERT
ON Zawody FOR EACH ROW
BEGIN
    :new.Nr_zawodow := Zawody_seq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Zawody_Zawody_seq1 AFTER UPDATE OF Nr_zawodow
ON Zawody FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_zawodow in table Zawody as it u
END;
/

-- Trigger for sequence Adresy_seq1 for column Nr_adresu in table Adresy -----
CREATE OR REPLACE TRIGGER ts_Adresy_Adresy_seq1 BEFORE INSERT
ON Adresy FOR EACH ROW
BEGIN
    :new.Nr_adresu := Adresy_seq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Adresy_Adresy_seq1 AFTER UPDATE OF Nr_adresu
ON Adresy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_adresu in table Adresy as it us
END;
/

-- Trigger for sequence Stanowisko_seq1 for column Nr_stanowiska in table Stanowiska ----
CREATE OR REPLACE TRIGGER ts_Stalowiska_Stalowisko_seq1 BEFORE INSERT
ON Stanowiska FOR EACH ROW
BEGIN
    :new.Nr_stanowiska := Stanowisko_seq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Stalowiska_Stalowisko_seq1 AFTER UPDATE OF Nr_stanowiska
ON Stanowiska FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_stanowiska in table Stanowiska
END;
/

-- Trigger for sequence Wynagrodzenia_seq1 for column Nr_wynagrodzenia in table Wynagrodz
CREATE OR REPLACE TRIGGER ts_Wynagrodzenia_Wynagrodzenia_seq1 BEFORE INSERT
ON Wynagrodzenia FOR EACH ROW
BEGIN

```

```

        :new.Nr_wynagrodzenia := Wynagrodzenia_seq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wynagrodzenia_Wynagrodzenia_seq1 AFTER UPDATE OF Nr_wynagro
ON Wynagrodzenia FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_wynagrodzenia in table Wynagrod
END;
/

-- Trigger for sequence Terminy_seq1 for column Nr_terminu in table Terminy -----
CREATE OR REPLACE TRIGGER ts_Terminy_Terminy_seq1 BEFORE INSERT
ON Terminy FOR EACH ROW
BEGIN
    :new.Nr_terminu := Terminy_seq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Terminy_Terminy_seq1 AFTER UPDATE OF Nr_terminu
ON Terminy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_terminu in table Terminy as it
END;
/

-- Trigger for sequence Nagrody_seq1 for column Nr_nagrody in table Nagrody -----
CREATE OR REPLACE TRIGGER ts_Nagrody_Nagrody_seq1 BEFORE INSERT
ON Nagrody FOR EACH ROW
BEGIN
    :new.Nr_nagrody := Nagrody_seq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Nagrody_Nagrody_seq1 AFTER UPDATE OF Nr_nagrody
ON Nagrody FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_nagrody in table Nagrody as it
END;
/

-- Trigger for sequence Poczty_seq1 for column Nr_poczty in table Poczty -----
CREATE OR REPLACE TRIGGER ts_Poczty_Poczty_seq1 BEFORE INSERT
ON Poczty FOR EACH ROW
BEGIN
    :new.Nr_poczty := Poczty_seq1.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Poczty_Poczty_seq1 AFTER UPDATE OF Nr_poczty
ON Poczty FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010,'Cannot update column Nr_poczty in table Poczty as it us

```



```
END;  
/
```

```
-- Create foreign keys (relationships) section -----
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Zatrudnia FOREIGN KEY (Nr_klubu) REFERENCES Klub_Le  
/
```

```
ALTER TABLE Grupy ADD CONSTRAINT Trenujaca FOREIGN KEY (Nr_dyscypliny) REFERENCES Dyscypl  
/
```

```
ALTER TABLE Grupy ADD CONSTRAINT Prowadzi_grupe FOREIGN KEY (Nr_klubu) REFERENCES Klub_Le  
/
```

```
ALTER TABLE Czlonkowie ADD CONSTRAINT Ma_czlonka FOREIGN KEY (Nr_klubu) REFERENCES Klub_L  
/
```

```
ALTER TABLE Dyscypliny ADD CONSTRAINT Prowadzi_dyscypline FOREIGN KEY (Nr_klubu) REFERENC  
/
```

```
ALTER TABLE Klub_Lekkoatletyczny ADD CONSTRAINT Klub_ma_adres FOREIGN KEY (Nr_adresu) REF  
/
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_ma_adres FOREIGN KEY (Nr_adresu) REFERENC  
/
```

```
ALTER TABLE Czlonkowie ADD CONSTRAINT Czloniek_ma_adres FOREIGN KEY (Nr_adresu) REFERENCES  
/
```

```
ALTER TABLE Wynagrodzenia ADD CONSTRAINT Pracownik_ma_wynagrodzenie FOREIGN KEY (Nr_praco  
/
```

```
ALTER TABLE Adresy ADD CONSTRAINT Adres_ma_poczte FOREIGN KEY (Nr_poczty) REFERENCES Poczty (Nr_poczty)
/
```

```
ALTER TABLE Zawody ADD CONSTRAINT Zawody_maja_adres FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)
/
```

```
ALTER TABLE Pracownicy ADD CONSTRAINT Pracownik_ma_stanowisko FOREIGN KEY (Nr_stanowiska) REFERENCES Stanowiska (Nr_stanowiska)
/
```

```
ALTER TABLE Terminy_zajec_grup ADD CONSTRAINT Grupy_maja_zajecia FOREIGN KEY (Nr_grupy) REFERENCES Grupy (Nr_grupy)
/
```

```
ALTER TABLE Terminy_zajec_grup ADD CONSTRAINT Zajecia_maja_terminy FOREIGN KEY (Nr_terminu) REFERENCES Terminy (Nr_terminu)
/
```

```
ALTER TABLE Czlonkowie_grup ADD CONSTRAINT Uczeszcza_do FOREIGN KEY (Nr_czlonka) REFERENCES Czlonkowie (Nr_czlonka)
/
```

```
ALTER TABLE Czlonkowie_grup ADD CONSTRAINT Grupa_zajmowana_przez FOREIGN KEY (Nr_grupy) REFERENCES Grupy (Nr_grupy)
/
```

```
ALTER TABLE Trenerzy_grup ADD CONSTRAINT Pracuje_z_trener FOREIGN KEY (Nr_pracownika) REFERENCES Pracownicy (Nr_pracownika)
/
```

```
ALTER TABLE Trenerzy_grup ADD CONSTRAINT Pracuje_z_grupa FOREIGN KEY (Nr_grupy) REFERENCES Grupy (Nr_grupy)
/
```

```
ALTER TABLE Konkurencje_na_zawodach ADD CONSTRAINT Zawody_maja_konkurencje FOREIGN KEY (Nr_zawodu) REFERENCES Zawody (Nr_zawodu)
/
```

```
ALTER TABLE Konkurencje_na_zawodach ADD CONSTRAINT Dotyczace FOREIGN KEY (Nr_dyscypliny)
/
```

```
ALTER TABLE Nagrody ADD CONSTRAINT Konkurencje_maja_nagrody FOREIGN KEY (Nr_zawodow, Nr_d
/
```

```
ALTER TABLE Uczestnicy_zawodow ADD CONSTRAINT Zawody_maja_uczestnikow FOREIGN KEY (Nr_zaw
/
```

5.4 Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

1) Wyszukanie trenerów i ich grup treningowych

```
Select Imie, Nazwisko, Data_zatrudnienia, Nazwa as Dyscyplina, Kat_wiekowa
From Pracownicy
Natural join Trenerzy
Natural Join Trenerzy_grup
Natural join Grupy
Natural Join Dyscypliny
```

DML Select

```

1 Select Imie, Nazwisko, Data_zatrudnienia, Nazwa as Dyscyplina, Kat_wiekowa
2 From Pracownicy
3 Natural join Trenerzy
4 Natural Join Trenerzy_grup
5 Natural join Grupy
6 Natural Join Dyscypliny
7

```

Data Grid

Messages | Data Grid | Trace | DBMS Output (disabled) | Query Viewer | Explain Plan | Script Output

Default

IMIE	NAZWISKO	DATA_ZATRUDNIENIA	DYSCYPLINA	KAT_WIEKOWA
▶ Kasia	Kuchta	12/12/2022	bieg na 100m	U15
Kasia	Kuchta	12/12/2022	bieg na 100m	U17
Marcin	Nayman	13/05/2019	bieg na 100m	U17
Michał	Probież	06/09/2020	bieg na 400m	U19
Anna	Kowalska	15/03/2015	skok o tyczce	U23
Piotr	Nowak	20/07/2010	skok w dal	U23
Piotr	Nowak	20/07/2010	skok w dal	Senior

2) Wyszukanie uczestników zawodów

```

Select c.Imie, c.Nazwisko, c.Data_urodzenia, k.Nr_zawodow
From Czlonkowie c
Join Uczestnicy_zawodow u On u.Nr_czlonka = c.Nr_czlonka
Join Konkurencje_na_zawodach k On k.Nr_dyscypliny = u.Nr_dyscypliny and
k.Nr_zawodow = u.Nr_zawodow and k.Kat_wiekowa = u.Kat_wiekowa;

```

DML Select

```

1 Select c.Imie, c.Nazwisko, c.Data_urodzenia, k.Nr_zawodow From Czlonkowie c
2 Join Uczestnicy_zawodow u On u.Nr_czlonka = c.Nr_czlonka
3 Join Konkurencje_na_zawodach k On k.Nr_dyscypliny = u.Nr_dyscypliny and k.Nr_zawodow = u.Nr_zawodow and k.Kat_wiekowa = u.Kat_wiekowa;
4

```

Data Grid

Messages | Data Grid | Trace | DBMS Output (disabled) | Query Viewer | Explain Plan | Script Output

Default

IMIE	NAZWISKO	DATA_URODZENIA	NR_ZAWODOW
▶ Jan	Kowalski	15/04/2005	1
Jan	Kowalski	15/04/2005	2
Anna	Nowak	20/07/2000	1
Anna	Nowak	20/07/2000	2
Mateusz	Wiśniewski	10/11/2002	1
Katarzyna	Dąbrowska	25/03/1999	1
Paweł	Lewandowski	01/12/2003	1
Martyna	Kaczmarek	08/05/2001	2

3) Wyszukanie terminów zajęć treningowych

```

SELECT t.Dzien_tygodnia, t.Godzina_od, t.Godzina_do, g.Nr_grupy
FROM Terminy t
Left JOIN Terminy_zajec_grup tzg ON t.Nr_terminu = tzg.Nr_terminu
Left Join Grupy g on g.Nr_grupy = tzg.NR_GRUPY;

```

1	SELECT t.Dzien_tygodnia, t.Godzina_od, t.Godzina_do, g.Nr_grupy
2	FROM Terminy t
3	Left JOIN Terminy_zajec_grup tzg ON t.Nr_terminu = tzg.Nr_terminu
4	Left Join Grupy g on g.Nr_grupy = tzg.NR_GRUPY;

DZIEN_TYGODNIA	GODZINA_OD	GODZINA_DO	NR_GRUPY
PN	14:00	15:00	1
WT	15:00	16:00	2
SR	20:00	21:00	3
CZW	17:00	19:00	4
PT	19:00	21:00	5
SB	11:00	13:00	5
ND	13:00	14:00	

4) Wyszukanie wszystkich pracowników, ich wynagrodzeń, kwalifikacji (dla trenera) oraz specjalizacji (dla fizjoterapeutów)

```

Select p.Imie, p.Nazwisko, st.Nazwa, w.Wynagrodzenie_brutto,
t.Kwalifikacje As Kwalifikacje_trenera, f.Specjalizacja As
Specjalizacja_fizjoterapeuty
From Pracownicy p
Left join Fizjoterapeuci f on f.Nr_pracownika = p.Nr_pracownika
Left Join Trenerzy t On t.NR_PRACOWNIKA = p.Nr_pracownika
Join Stanowiska st on st.Nr_stanowiska = p.Nr_stanowiska
Join Wynagrodzenia w on w.Nr_pracownika = p.Nr_pracownika

```

1	Select p.Imie, p.Nazwisko, s.Nazwa, w.Wynagrodzenie_brutto, t.Kwalifikacje As Kwalifikacje_trenera, f.Specjalizacja As Specjalizacja_fizjoterapeuty
2	From Pracownicy p
3	Left join Fizjoterapeuci f on f.Nr_pracownika = p.Nr_pracownika
4	Left Join Trenerzy t On t.NR_PRACOWNIKA = p.Nr_pracownika
5	Join Stanowiska s on s.Nr_stanowiska = p.Nr_stanowiska
6	Join Wynagrodzenia w on w.Nr_pracownika = p.Nr_pracownika
7	

IMIE	NAZWISKO	NAZWA	WYNAGRODZENIE_BRUTTO	KWALIFIKACJE_TRENERA	SPECJALIZACJA_FIZJOTERAPEUTY
Kasia	Kuchta	Trener	3000	Trener M	
Marcin	Nayman	Trener	3200	Trener I	
Michał	Probiez	Trener	2900	Trener	
Anna	Kowalska	Trener	2600	Instruktor	
Piotr	Nowak	Trener	3500	Trener	
Monika	Wiśniewska	Fizjoterapeuta	3100		Kończyny dolne
Marek	Dąbrowski	Kierownik	2700		
Marta	Lewandowska	Fizjoterapeuta	3250		Kończyny górne

5) Zliczenie ilu trenerów ma grupa

```

Select tg.Nr_grupy, Count(tg.Nr_pracownika) from Trenerzy_grup tg
Group by tg.Nr_grupy;

```

The screenshot shows a database query tool interface. The top pane contains a SQL query:

```
1 Select tg.Nr_grupy, Count(tg.Nr_pracownika) from Trenerzy_grup tg
2 Group by tg.Nr_grupy;
3
4
```

The bottom pane, titled "Data Grid", shows the results of the query in a table with two columns: NR_GRPY and COUNT(TG.NR_PRACOWNIKA). The table has 6 rows of data.

NR_GRPY	COUNT(TG.NR_PRACOWNIKA)
1	1
2	2
3	1
4	1
5	1
6	1

6) Zliczenie ilu uczestników na poszczególnych zawodach

```
Select z.Nr_zawodow, Count(u.Nr_czlonka) From Zawody z
Left Join Konkurencje_na_zawodach knz on knz.Nr_zawodow = z.Nr_zawodow
Left Join Uczestnicy_zawodow u on knz.Nr_zawodow = u.Nr_zawodow and
knz.Nr_dyscypliny = u.Nr_dyscypliny and knz.Kat_wiekowa = u.Kat_wiekowa
Group by z.Nr_zawodow
```

The screenshot shows a database query tool interface. The top pane contains a SQL query:

```
1 Select z.Nr_zawodow, Count(u.Nr_czlonka) From Zawody z
2 Left Join Konkurencje_na_zawodach knz on knz.Nr_zawodow = z.Nr_zawodow
3 Left Join Uczestnicy_zawodow u on knz.Nr_zawodow = u.Nr_zawodow and knz.Nr_dyscypliny = u.Nr_dyscypliny and knz.Kat_wiekowa = u.Kat_wiekowa
4 Group by z.Nr_zawodow
```

The bottom pane, titled "Data Grid", shows the results of the query in a table with two columns: NR_ZAWODOW and COUNT(U.NR_CZLONKA). The table has 4 rows of data.

NR_ZAWODOW	COUNT(U.NR_CZLONKA)
1	5
2	3
3	0
4	0

7) Zliczenie ilu członków w poszczególnych grupach

```
Select Nr_grupy, Count(Nr_czlonka) From Grupy
Natural Join Czlonkowie_grup
Natural Join Czlonkowie
Group by Nr_grupy
Order by Nr_grupy asc;
```

1	Select Nr_grupy, Count(Nr_czlonka) From Grupy
2	Natural Join Czlonkowie_grup
3	Natural Join Czlonkowie
4	Group by Nr_grupy
5	Order by Nr_grupy asc;
6	
7	

NR_GRUPY	COUNT(NR_CZLONKA)
1	3
2	2
3	2
4	1
5	2
6	1

8) Wyszukanie adresu i kodu pocztowego klubu lekkoatletycznego

```
Select Nazwa, Data_zalozenia, Ulica, Numer_lokalu, Kod_poczty
From Klub_lekkoatletyczny
Natural Join Adresy
Natural Join Poczty
```

1	Select Nazwa, Data_zalozenia, Ulica, Numer_lokalu, Kod_poczty From Klub_lekkoatletyczny
2	Natural Join Adresy
3	Natural Join Poczty

NAZWA	DATA_ZALOZENIA	ULICA	NUMER_LOKALU	KOD_PO CZTY
Lekki Bucik	05/11/2003	Krochmalna	4	01-001

9) Łączna pula nagród na zawodach

```
Select Nr_zawodow, Sum(I_miejsce) + Sum(II_miejsce) + Sum(III_miejsce)
Wydatki_na_nagrody From Konkurencje_na_zawodach
Natural join Zawody
Natural Join Nagrody
Group by Nr_zawodow
```

SQL Query Editor (DML Select):

```

1 ► Select Nr_zawodow, Sum(I_miejsce) + Sum(II_miejsce) + Sum(III_miejsce) Wydatki_na_nagrody From Konkurencje_na_zawodach
2 Natural join Zawody
3 Natural Join Nagrody
4 Group by Nr_zawodow

```

Data Grid:

NR_ZAWODOW	WYDATKI_NA_NAGRODY
1	17000
2	7300
3	6350
4	12000

10) Wyszukanie zwolnionych pracowników

Select * From Pracownicy Where Data_zwolnienia is not null

SQL Query Editor (DML Select):

```

1 ► Select * From Pracownicy Where Data_zwolnienia is not null

```

Data Grid:

NR_PRACOWNIKA	IMIE	NAZWISKO	PESEL	DATA_URODZENIA	DATA_ZATRUDNIENIA	DATA_ZWOLNIENIA	NUMER_TELEFONU	PLEC	NR_KLUBU	NR_ADRESU	NR_STANOWISKA
21	Piotr	Stan	45434635645	11/12/1990	11/01/2022	12/01/2022	342354345	M	1	10	1