

BADA - Klub lekkoatletyczny

Stanisław Ciszewicz (324 906), Jakub Kuszner (324 924)

Politechnika Warszawska, Cyberbezpieczeństwo

11 grudnia 2023

Spis treści

1. Zakres i cel projektu (opis założeń funkcjonalnych projektowanej bazy danych)	2
2. Definicja systemu	2
2.1. Perspektywy użytkowników	2
3. Model konceptualny	2
3.1. Definicja zbiorów encji określonych w projekcie (decyzje projektowe)	2
3.2. Ustalenie związków między encjami i ich typów	3
3.3. Określenie atrybutów i ich dziedzin	4
3.3.1. Klub	4
3.3.2. Zawodnik	5
3.3.3. Trener	5
3.3.4. Wydarzenie	6
3.3.5. Obiekt	6
3.3.6. Trening	7
3.3.7. Kibic	7
3.4. Dodatkowe reguły integralnościowe (reguły biznesowe)	8
3.5. Klucze kandydujące i główne (decyzje projektowe)	8
3.6. Schemat ER na poziomie konceptualnym	9
3.7. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady	9
3.8. Pułapki wachlarzowe	9
3.9. Pułapki szczelinowe	9
4. Model logiczny	10
4.1. Charakterystyka modelu relacyjnego	10
4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady	10
4.3. Proces normalizacji – analiza i przykłady	11
4.3.1. Pierwsza postać normalna	11
4.3.2. Druga postać normalna	12
4.3.3. Trzecia postać normalna	14
4.4. Schemat ER na poziomie modelu logicznego	14
4.5. Więzy integralności	14
4.6. Proces denormalizacji – analiza i przykłady	15
5. Faza fizyczna	16
5.1. Projekt transakcji i weryfikacja ich wykonalności	16
5.2. Strojanie bazy danych – dobór indeksów	16
5.3. Skrypt SQL zakładający bazę danych	17
5.4. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych	24
5.4.1. Wpisywanie danych do bazy	24
5.4.2. Przykładowe zapytania do bazy	26

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

Celem niniejszego projektu jest zaprojektowanie bazy danych spełniającej wymagania klubu lekkoatletycznego. Klub lekkoatletyczny zajmuje się szkoleniem zawodników, organizacją wydarzeń sportowych i treningów. Klub lekkoatletyczny posiada zawodników, którzy uczestniczą w treningach i wydarzeniach sportowych, odbywających się na posiadanych przez klub obiektach sportowych. Klub lekkoatletyczny ma w swojej ofercie dyscypliny, dla każdej zatrudniając specjalistów w osobach trenerów. Założeniem naszego klubu jest to, że każdy zawodnik może trenować daną dyscyplinę i uczęszczać na wiele treningów, do różnych trenerów. Klub posiada również kibiców, którzy mogą uczestniczyć i dopingować zawodników w wydarzeniach sportowych. W naszym projekcie wyodrębnione zostaną statusy użytkowników z odpowiednimi uprawnieniami, pozwalające korzystać z bazy oraz utrzymywać ją. Co więcej każdy typ użytkownika będzie mógł przeprowadzać określone operacje, które zostaną opisane w dalszej części raportu. Ostateczne rozwiązanie (po uprzednim stworzeniu modelu konceptualnego i logicznego, a następnym zasileniu bazy danych) powinno spełniać wszystkie wyżej wymienione założenia.

2. Definicja systemu

Nasz system w finalnej wersji powinien spełniać następujące wymagania:

- obsługa zawodników:
 - dodawanie, usuwanie zawodników
 - modyfikacja dyscypliny uprawianej przez zawodnika
- obsługa obiektów:
 - dodawanie nowych obiektów
 - modyfikacja pojemności obiektu w razie rozbudowy
 - sprawdzanie jakie wydarzenia i treningi odbywają się na obiekcie
- obsługa wydarzeń:
 - dodawanie wydarzeń,
 - modyfikacja danych uczestników wydarzeń (dodawanie kibiców i zawodników biorących w nim udział)
- obsługa treningów:
 - dodawanie, usuwanie i modyfikacja treningu
 - modyfikacja uczestników treningu (dopisywanie zawodników do konkretnego treningu)
- obsługa trenerów:
 - wypisanie aktualnej listy zawodników trenowanych przez trenera
 - dodawanie i usuwanie trenerów
 - modyfikacja danych trenerów

2.1. Perspektywy użytkowników

W naszym systemie wyodrębniamy następujące perspektywy użytkowników:

- właściciel klubu lekkoatletycznego - posiada on dostęp do wszystkich danych znajdujących się w bazie danych, posiada uprawnienia do realizowania każdej z wyżej wymienionych operacji w bazie danych
- zawodnik - ma on możliwość wglądu w swoje dane osobowe, może również uzyskać dane o treningach lub wydarzeniach, w których będzie uczestniczył
- trener - może zmodyfikować dane treningu (zaplanować, zmienić godzinę lub całkowicie odwołać), zobaczyć listę swoich zawodników
- kibic - ma możliwość wglądu w wydarzenie organizowane przez klub oraz zapisu na nie, ma również wgląd w swoje dane osobowe
- administracja - posiada możliwość dodania, usunięcia zawodnika i trenera (lub modyfikacji danych), dodaje kolejne wydarzenia czy obiekty (czuwa nad systemem)

3. Model konceptualny

3.1. Definicja zbiorów encji określonych w projekcie (decyzje projektowe)

W naszym projekcie na poziomie modelu konceptualnego wyodrębniliśmy siedem encji:

- Klub - encja główna projektu, zawiera dane klubu
- Zawodnik - podopieczny klubu, encja zawiera dane zawodnika (w tym uprawianą przez niego dyscyplinę)
- Trener - pracownik klubu lekkoatletycznego, zawiera dane trenera (w tym dyscyplinę, z której prowadzi treningi)
- Wydarzenie - wydarzenia sportowe organizowane przez klub lekkoatletyczny, zawiera dane o wydarzeniu
- Obiekt - obiekt klubu, na których odbywają się wydarzenia i treningi, zawiera dane obiektu
- Trening - organizowany przez klub lekkoatletyczny, zawiera dane o treningu
- Kibic - uczestnik wydarzeń, zawiera dane kibica

3.2. Ustalenie związków między encjami i ich typów

a) Związek między encjami klub-zawodnik

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Klub	1...1	Posiada_zawodnika	0...n	Zawodnik

Tabela 1. Związek Klub-Zawodnik

Klub (istnieje dokładnie jeden) posiada zawodników - może nie mieć żadnego do maksymalnie wielu.

b) Związek między encjami Klub-Wydarzenie

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Klub	1...1	Klub_organizuje_wydarzenie	0...n	Wydarzenie

Tabela 2. Związek Klub-Wydarzenie

Klub (istnieje dokładnie jeden) - może nie organizować żadnego wydarzenia do maksymalnie wielu.

c) Związek między encjami Klub-Obiekt

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Klub	1...1	Klub_posiada_obiekt	0...n	Obiekt

Tabela 3. Związek Klub-Obiekt

Klub (istnieje dokładnie jeden) - może nie posiadać żadnego obiektu do maksymalnie wielu.

d) Związek między encjami Klub-Trener

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Klub	1...1	Klub_zatrudnia_trenera	0...n	Trener

Tabela 4. Związek Klub-Trener

Klub (istnieje dokładnie jeden) - może nie zatrudniać żadnego trenera do maksymalnie wielu.

e) Związek między encjami Klub-Trening

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Klub	1...1	Klub_organizuje_trening	0...n	Trening

Tabela 5. Związek Klub-Trening

Klub (istnieje dokładnie jeden) - może nie organizować żadnego treningu do maksymalnie wielu.

f) Związek między encjami Zawodnik-Wydarzenie

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Zawodnik	0...n	Bierze_udzial	0...n	Wydarzenie

Tabela 6. Związek Zawodnik-Wydarzenie

Zawodnik (może nie być żadnego zawodnika do wielu) - może brać udział w żadnym do maksymalnie wielu wydarzeń.

g) Związek między encjami Wydarzenie-Obiekt

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Wydarzenie	0...n	Wydarzenie.odbywa.sie.na.obiekcie	1...1	Obiekt

Tabela 7. Związek Zawodnik-Obiekt

Obiekt (istnieje dokładnie jeden) - może się nie odbywać na nim żadne wydarzenie do maksymalnie wielu.

h) Związek między encjami Obiekt-Trening

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Wydarzenie	1...n	Trening.odbywa.sie.na.obiekcie	0...n	Obiekt

Tabela 8. Związek Obiekt-Trening

Obiekt (istnieje dokładnie jeden) - może się nie odbywać na nim żaden trening do maksymalnie wielu.

i) Związek między encjami Trener-Trening

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Trener	1...1	Trener.prowadzi.trening	0...n	Trening

Tabela 9. Związek Trener-Trening

Trener (istnieje dokładnie jeden) - może nie prowadzić żadnego treningu do maksymalnie wielu.

j) Związek między encjami Wydarzenie-Kibic

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Wydarzenie	0...n	Kibic.uczestniczy.w.wydarzeniu	0...n	Kibic

Tabela 10. Związek Wydarzenie-Kibic

Kibic (może nie być żadnego) - może nie uczestniczyć w żadnym wydarzeniu do maksymalnie wielu.

k) Związek między encjami Zawodnik-Trening

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Zawodnik	0...n	Zawodnik.bierze.udzial.w.treningu	0...n	Trening

Tabela 11. Związek Zawodnik-Trening

Zawodnik (może nie być żadnego) - może nie brać udziału w żadnym treningu do maksymalnie wielu.

l) Związek między encjami Trener-Zawodnik

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Trener	1...n	Trener.treduje.zawodnika	0...n	Zawodnik

Tabela 12. Związek Trener-Zawodnik

Trener (może nie być żadnego) - może nie trenować żadnego zawodnika do maksymalnie wielu.

3.3. Określenie atrybutów i ich dziedzin

3.3.1. Klub

Encja główna projektu, zawiera informacje dotyczące klubu:

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_klubu	Unikalny identyfikator klubu	Integer	T	T	T	
Nazwa	Nazwa klubu	VarChar(100)	T	N	T	
Adres	Adres siedziby klubu	VarChar(200)	T	N	N	Pole segmentowe, obejmujące takie informacje jak miasto, ulica, numer lokalizacji, kod pocztowy,
Data_zalozenia	Data założenia klubu	Date	T	N	T	
Właściciel	Imię i nazwisko właściciela klubu	VarChar(100)	T	N	N	Pole wielowartościowe, segmentowe

Tabela 13. Atrybuty encji Klub

3.3.2. Zawodnik

Encja zawierająca dane zawodnika oraz dyscyplinę, którą uprawia:

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_zawodnika	Unikalny identyfikator zawodnika	Integer	T	T	T	
Imie	Imię zawodnika	VarChar(20)	T	N	T	
Nazwisko	Nazwisko zawodnika	VarChar(30)	T	N	T	
Data_urodzenia	Data urodzenia zawodnika	Date	T	N	T	
PESEL	Numer PESEL zawodnika	Character(11)	N	N	T	Zawodnik może nie posiadać numeru PESEL (np. zawodnik zza granicy)
Adres	Adres zamieszkania zawodnika	VarChar(200)	T	N	N	Pole segmentowe, obejmujące takie informacje jak miasto, ulica, numer lokalizacji, kod pocztowy,
Nr_telefonu	Numer telefonu zawodnika	VarChar(15)	T	N	T	
Email	Adres e-mail zawodnika	VarChar(100)	N	N	T	
Plec	Plec zawodnika (do wyboru z mezczyzna i kobieta)	Character(1)	T	N	T	
Dyscyplina	Dyscypliny uprawiane przez zawodnika	VarChar(30)	T	N	T	

Tabela 14. Atrybuty encji Zawodnik

3.3.3. Trener

Encja zawierająca dane trenera oraz dyscyplinę, którą trenuje:

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_trenera	Unikalny identyfikator trenera	Integer	T	T	T	
Imie	Imię trenera	VarChar(20)	T	N	T	
Nazwisko	Nazwisko trenera	VarChar(30)	T	N	T	
Data_urodzenia	Data urodzenia trenera	Date	T	N	T	
PESEL	Numer PESEL trenera	Character(11)	N	N	T	Trener może nie posiadać numeru PESEL (np. zawodnik zza granicy)
Adres	Adres zamieszkania trenera	VarChar(200)	T	N	N	Pole segmentowe, obejmujące takie informacje jak miasto, ulica, numer lokalizacji, kod pocztowy,
Nr_telefonu	Numer telefonu trenera	VarChar(15)	T	N	T	
Email	Adres e-mail trenera	VarChar(100)	N	N	T	
Plec	Plec trenera (do wyboru z mężczyzna i kobieta)	Character(1)	T	N	T	
Dyscyplina	Dyscyplina, z której trener przygotowuje zawodnika	VarChar(30)	T	N	T	
Data_zatrudnienia	Unikalny identyfikator trenera	Date	T	N	T	

Tabela 15. Atrybuty encji Trener

3.3.4. Wydarzenie

Encja zawierająca dane wydarzenia organizowanego przez klub lekkoatletyczny:

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_wydarzenia	Unikalny identyfikator wydarzenia	Integer	T	T	T	
Nazwa	Nazwa wydarzenia	VarChar(100)	T	N	T	
Data	Data wydarzenia	Date	T	N	T	
Dyscyplina	Dyscypliny, które odbywają się na wydarzeniu	VarChar(30)	T	N	T	

Tabela 16. Atrybuty encji Wydarzenie

3.3.5. Obiekt

Encja zawierająca dane obiektu:

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_obiektu	Unikalny identyfikator obiektu	Integer	T	T	T	
Adres	Adres obiektu	VarChar(200)	T	N	N	Pole segmentowe, obejmujące takie informacje jak miasto, ulica, numer lokalizacji, kod pocztowy,
Nazwa	Nazwa obiektu	VarChar(100)	T	N	T	
Pojemnosc	Pojemność obiektu (ile osób może pomieścić)	Integer	N	N	T	

Tabela 17. Atrybuty encji Obiekt

3.3.6. Trening

Encja zawierająca dane treningu organizowanego przez klub lekkoatletyczny:

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_treningu	Unikalny identyfikator treningu	Integer	T	T	T	
Godzina od	Godzina rozpoczęcia zajęć	DateTime	T	N	T	
Godzina do	Godzina zakończenia zajęć	DateTime	T	N	T	
Dyscyplina	Dyscyplina uprawiana na treningu	VarChar(30)	T	N	T	

Tabela 18. Atrybuty encji Trening

3.3.7. Kibic

Encja zawierająca dane kibica:

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_kibica	Unikalny identyfikator kibica	Integer	T	T	T	
Imie	Imię kibica	VarChar(20)	T	N	T	
Nazwisko	Nazwisko kibica	VarChar(30)	T	N	T	
Data urodzenia	Data urodzenia kibica	Date	N	N	T	
PESEL	Numer PESEL kibica (jeśli posiada)	VarChar(11)	N	N	T	Kibic może nie posiadać numeru PESEL (np. zawodnik zza granicy)
Adres	Adres zamieszkania kibica	VarChar(200)	N	N	N	
Nr_telefonu	Numer telefonu komórkowego kibica	VarChar(15)	T	N	T	
Email	Adres e-mail kibica	VarChar(100)	T	N	T	
Plec	Płeć kibica	PlecD	T	N	T	

Tabela 19. Atrybuty encji Kibic

3.4. Dodatkowe reguły integralnościowe (reguły biznesowe)

W naszej bazie konieczne było zaimplementowanie jednej reguły integralnościowej, która pozwoliła na zdefiniowanie jaki typ danych jest dozwolony w danym atrybucie:

Nazwa reguły	Ograniczenie	Opis	Encje, których dotyczy
PlecR	Plec in ('K','M')	Osoba w naszej bazie danych może być płci męskiej (M) lub żeńskiej (K)	Zawodnik, Trener, Kibic

Tabela 20. Dodatkowe reguły integralnościowe

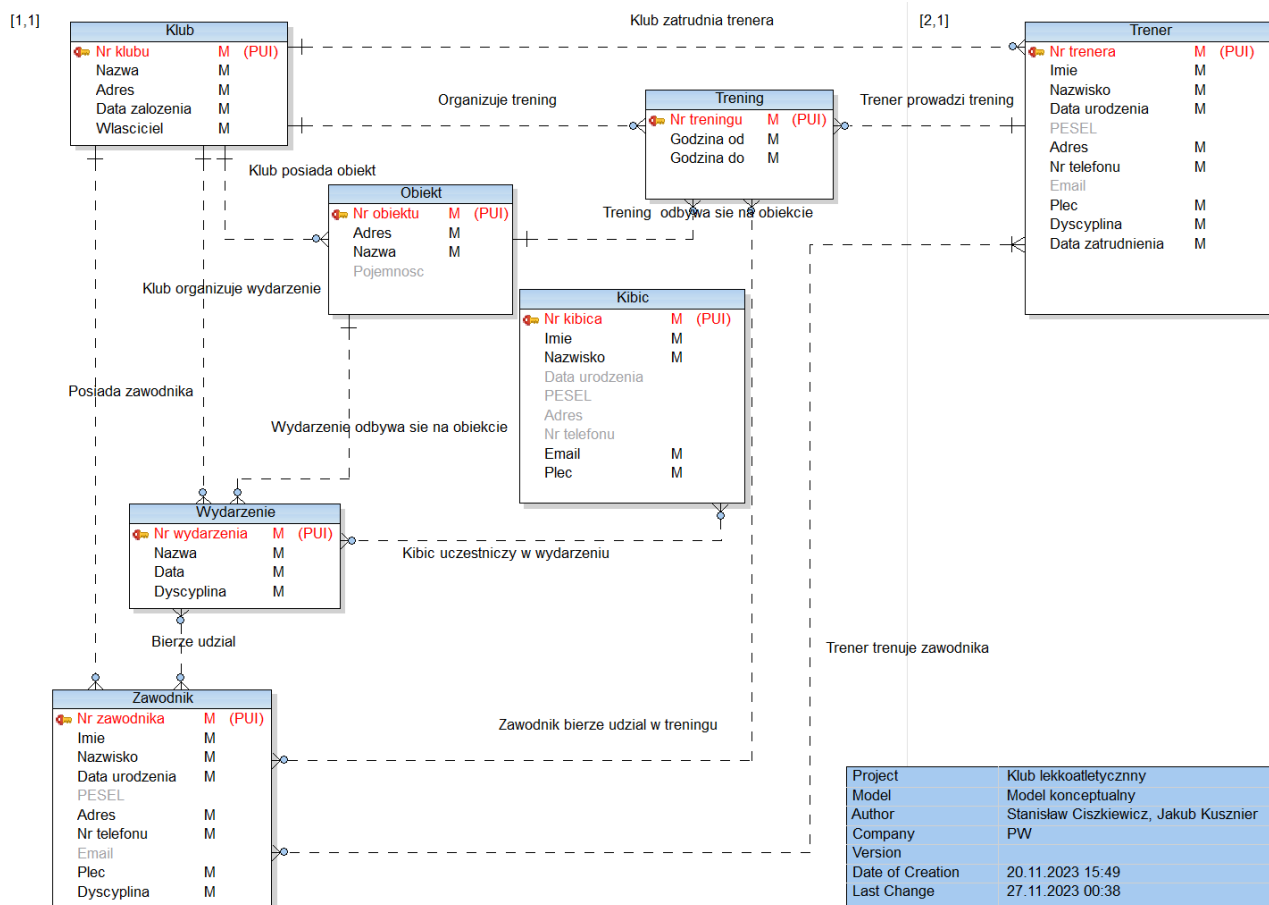
3.5. Klucze kandydujące i główne (decyzje projektowe)

W tym kroku przeanalizowaliśmy atrybuty każdej z encji aby zidentyfikować potencjalne klucze kandydujące i podjąć ostateczną decyzję co do wyboru kluczy głównych:

Encja	Klucz główny	Klucz kandydujący	Uzasadnienie
Klub	Nr_klubu	Nazwa klubu	-
Zawodnik	Nr_zawodnika	PESEL	Nie każdy zawodnik musi posiadać numer PESEL (choćby zawodnik zagraniczny)
Trener	Nr_trenera	PESEL	Nie każdy trener musi posiadać numer PESEL (choćby trener zagraniczny)
Wydarzenie	Nr_wydarzenia	Nazwa wydarzenia	-
Obiekt	Nr_obiektu	Nazwa obiektu	-
Trening	Nr_treningu	Nazwa treningu	-
Kibic	Nr_kibica	PESEL	Nie każdy kibic musi posiadać numer PESEL (choćby kibic zagraniczny)

Tabela 21. Klucze kandydujące i główne

3.6. Schemat ER na poziomie konceptualnym



Rys. 1. Schemat konceptualny projektu bazy danych klubu lekkoatletycznego

3.7. Problem pułapek szczelinowych i wachlarzowych – analiza i przykłady

3.8. Pułapki wachlarzowe

W pierwszym kroku przystąpiliśmy do analizy stworzonego przez nas schematu konceptualnego w celu znalezienia potencjalnie występujących pułapek wachlarzowych. Pułapki wachlarzowe występują, gdy na poziomie projektu zakładamy możliwość stworzenia połączenia między dwoma encjami, lecz sama baza danych i wyznaczone związki nie pozwalają na jednoznaczne zdefiniowanie powiązania. Wystąpienie tego typu błędu wynika z błędnego zaprojektowania modelu konceptualnego i wymaga jego modyfikacji w celu uniknięcia dalszych problemów w modelu logicznym i fizycznej implementacji bazy danych. Analizując model konceptualny i definicję naszej bazy danych (zdefiniowanej w punkcie 2) doszliśmy do wniosku, że w naszej bazie danych nie zaistniała pułapka wachlarzowa - nie istnieje funkcjonalność bazy danych, która przy obecnym połączeniu encji (zobrazowanym na schemacie modelu konceptualnego) prowadziłaby do zaistnienia braku jednoznaczności. Potencjalne wystąpienie pułapki wachlarzowej moglibyśmy napotkać w przypadku braku połączenia między encją Obiekt a Wydarzenie - połączenia wiele do jednego z klubem nie pozwalałyby na jednoznaczne określenie i jednocześnie spełnienie wymagania bazy danych dotyczącego określenia na jakim obiekcie odbywają się zawody.

3.9. Pułapki szczelinowe

Kolejnym rodzajem pułapki jaką możemy napotkać po stworzeniu modelu konceptualnego jest pułapka szczelinowa. Występowanie pułapki szczelinowej może zaistnieć w momencie występowania próżni (braku połączenia) pomiędzy encjami, między którymi zgodnie z założeniami definicji systemu powinny istnieć relacje.

Podobnie jak w poprzednim przypadku podczas analizy naszego modelu nie znaleźliśmy tego typu pułapki - wszystkie wymagane połączenia istnieją. Takowy problem mógłby wystąpić, np. gdyby klub posiadał oddziały, które bezpośrednio zatrudniałyby swoich trenerów. W momencie, w którym trener nie był bezpośrednio zatrudniony przez oddział, lecz przez główną jednostkę klubu, a połączenia na poziomie encji występowałyby jedynie między klubem a oddziałem oraz oddziałem a trenerem napotkalibyśmy problem pułapki szczelinowej. W celu usunięcia tej niekompatybilności należałoby dodać dodatkowe połączenie między klubem a trenerem. W przypadku stworzonego przez nas modelu nie znaleźliśmy żadnej z obydwu typów pułapek.

4. Model logiczny

4.1. Charakterystyka modelu relacyjnego

Przechodząc do modelu relacyjnego należy po pierwsze zmienić encje na relacje na liczbę mnogą (tj. np. encja Trener na tabelę Trenerzy). Po drugie należy się pozbyć wszystkich relacji wiele do wielu - zamieniamy je na związki jeden do wielu, pomiędzy którymi tworzona jest tabela łącząca.

4.2. Usunięcie właściwości niekompatybilnych z modelem relacyjnym - przykłady

Tak jak wspomnieliśmy w poprzednim punkcie dokonaliśmy zmiany relacji wiele do wielu na związki jeden do wielu z tabelą łączącą. Poniżej prezentujemy relacje, między którymi należało wstawić table łączące

Relacja 1	Relacja 2	Tabela łącząca
Wydarzenia	Zawodnicy	Wydarzenia_Zawodnicy
Zawodnicy	Treningi	Zawodnicy_Treningi
Kibice	Wydarzenia	Kibice_Wydarzenia
Trenerzy	Zawodnicy	Trenerzy_Zawodnicy

Tabela 22. Stworzone tablice łączące w modelu logicznym

W każdej tablicy łączącej znajdują się klucze główne relacji będących ze sobą w związku:
Wydarzenia_Zawodnicy

Nazwa atrybutu	Typ	Czy obowiązkowy	Czy atrybut prosty
Nr wydarzenia	Integer	T	podstawowy
Nr zawodnika	Integer	T	podstawowy

Tabela 23. Tabela łącząca Wydarzenia z Zawodnikami

Zawodnicy_Treningi

Nazwa atrybutu	Typ	Czy obowiązkowy	Czy atrybut prosty
Nr zawodnika	Integer	T	podstawowy
Nr treningu	Integer	T	podstawowy

Tabela 24. Tabela łącząca Zawodników z Treningami

Kibice_Wydarzenia

Nazwa atrybutu	Typ	Czy obowiązkowy	Czy atrybut prosty
Nr kibica	Integer	T	podstawowy
Nr wydarzenia	Integer	T	podstawowy

Tabela 25. Tabela łącząca Kibiców z Wydarzeniami

Trenerzy_Zawodnicy

Nazwa atrybutu	Typ	Czy obowiązkowy	Czy atrybut prosty
Nr trenera	Integer	T	podstawowy
Nr zawodnika	Integer	T	podstawowy

Tabela 26. Tabela łącząca Trenerów z Zawodnikami

4.3. Proces normalizacji – analiza i przykłady

Po zmianie nazw związków i usunięciu relacji wiele do wielu przystąpiliśmy do procesu normalizacji, czyli procesu mającego na celu doprowadzenie naszego projektu do właściwej postaci (staramy się uniknąć sytuacji, w której w relacyjnej bazie danych mamy powtarzające się dane). Tradycyjny proces normalizacji składa się z trzech etapów, które kolejno przeprowadzimy na naszym modelu.

4.3.1. Pierwsza postać normalna

Przeprowadzanie normalizacji w pierwszej postaci normalnej rozpoczynamy od doprowadzenia wszystkich pól do postaci atomowej. Zgodnie z projektem encji z projektu conceptualnego wiemy, że wartościami atomowymi nie są pola atrybutów Właściciel z relacji Kluby oraz Adres znajdujący się w relacjach: Kluby, Zawodnicy, Trenerzy, Obiekty i Kibice.

W każdej z wyżej wymienionych encji zawierających adres dodaliśmy atrybuty atomowe roz dzielając złożony atrybut adres na: nr_budynku, nr_lokalu, ulica, miasto, kod_pocztowy.

Podobną operację wykonaliśmy dla relacji Kluby, w której konieczne było rozbić złożonego atrybutu właściciela na atomowe atrybuty: imie_wlasciciela, nazwisko_wlasciciela, data_urodzenia_wlasciciela, numer_PESEL_wlasciciela, nr_telefonu_wlasciciela, email_wlasciciela, plec_wlasciciela.

Kolejnym warunkiem, który musi zostać spełniony podczas pierwszego etapu normalizacji jest usunięcie powtarzających się grup. Redundancję możemy zaobserwować w relacjach Zawodnicy i Trenerzy - powtarzającą się wartością będzie w tym przypadku wartość atrybutu Dyscyplina. W celu rozwiązania tego problemu i tym samym spełnienia wszystkich wymagań pierwszej postaci normalnej stworzyliśmy relację Dyscypliny, która wygląda następująco:

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_dyscypliny	Unikalny identyfikator dyscypliny	Integer	T	T	T	
Nazwa_dyscypliny	Nazwa dyscypliny	VarChar2(30)	T	N	T	

Tabela 27. Atrybuty relacji Dyscypliny

Dodane zostały również następujące związki między relacjami:

a) Związek między relacjami Zawodnicy-Dyscyplina

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Zawodnicy	0...n	Zawodnik_uprawia_dyscypline	1...1	Dyscyplina

Tabela 28. Związek Zawodnicy-Dyscyplina

b) Związek między relacjami Trenerzy-Dyscyplina

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Trenerzy	0...n	Trener_treduje_dyscypline	1...1	Dyscyplina

Tabela 29. Związek Trenerzy-Dyscyplina

c) Związek między relacjami Wydarzenia-Dyscyplina

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Dyscyplina	1...1	Dyscyplina_jest_uprawiana_na_wydarzeniu	0...n	Wydarzenia

Tabela 30. Związek Wydarzenia-Dyscyplina

d) Związek między relacjami Treningi-Dyscyplina

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Dyscyplina	1...1	Dyscyplina_jest_uprawiana_na_treningu	0...n	Treningi

Tabela 31. Związek Treningi-Dyscyplina

4.3.2. Druga postać normalna

Następnym krokiem była analiza wymagań drugiej postaci normalnej. Aby model znajdował się w drugiej postaci normalnej musi być w pierwszej postaci normalnej oraz każdy atrybut musi zależeć jedynie od klucza. Tym samym zauważyliśmy, że stworzona przez nas struktura podczas pierwszej postaci normalnej rozbijająca pole adres na atrybuty atomowe powinna trafić do oddzielnej relacji (dając przykład np. atrybut ulica w relacji klub nie jest zależny jedynie od klucza głównego nr_klubu). Tym samym zdecydowaliśmy się na stworzenie nowej relacji Adresy, do której trafiły wszystkie pola atomowe powstałe z złożonego pola Adres (przed pierwszą postacią normalną). Poniżej prezentujemy strukturę relacji Adresy wraz z jej atrybutami:

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_adresu	Unikalny identyfikator adresu	Integer	T	T	T	
Nr_budynku	Numer budynku	VarChar2(5)	T	N	T	
Nr_lokalu	Numer lokalu	VarChar2(4)	N	N	T	Numer lokalu jest nieobowiązkowy, dotyczy bloków
Ulica	Ulica	VarChar2(20)	T	N	T	
Miasto	Miasto	VarChar2(20)	T	N	T	
Kod_pocztowy	Kod pocztowy	VarChar2(6)	T	N	T	

Tabela 32. Atrybuty relacji Adresy

Poniżej prezentujemy związki, jakie zostały dodane między nowoutworzoną relacją Adresy a innymi relacjami:

a) Związek między relacjami Kluby-Adresy

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Kluby	0...1	Klub_posiada_adres	1...1	Adresy

Tabela 33. Związek Kluby-Adresy

b) Związek między relacjami Zawodnicy-Adresy

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Zawodnicy	0...1	Zawodnik_posiada_adres	1...1	Adresy

Tabela 34. Związek Zawodnicy-Adresy

c) Związek między relacjami Trenerzy-Adresy

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Trenerzy	0...1	Trener_posiada_adres	1...1	Adresy

Tabela 35. Związek Trenerzy-Adresy

d) Związek między relacjami Kibice-Adresy

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Kibice	0...1	Kibic_posiada_adres	1...1	Adresy

Tabela 36. Związek Kibice-Adresy

e) Związek między relacjami Obiekty-Adresy

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Obiekty	0...1	Obiekt_posiada_adres	1...1	Adresy

Tabela 37. Związek Obiekty-Adresy

Z tego samego powodu zmuszeni byliśmy na stworzenie relacji Właściciele, która zawiera dane osobowe właściciela klubu lekkoatletycznego (wszystkie atomowe atrybuty powstałe z rozbicia złożonego atrybutu Właściciele podczas 1NF):

Nazwa atrybutu	Opis znaczenia atrybutu	Dziedzina	Czy obowiązkowy	Klucz główny	Czy atrybut prosty	Dodatkowe informacje
Nr_wlasciciela	Unikalny identyfikator właściciela	Integer	T	T	T	
Imie	Imię właściciela	VarChar2(20)	T	N	T	
Nazwisko	Nazwisko właściciela	VarChar2(30)	T	N	T	
Data_urodzenia	Data urodzenia właściciela	Date	T	N	T	
PESEL	Numer PESEL właściciela	Char(11)	N	N	T	Zawodnik może nie posiadać numeru PESEL (np. zawodnik zza granicy)
Numer_telefonu	Numer telefonu właściciela	VarChar2(15)	T	N	N	
Email	Adres e-mail właściciela	VarChar(100)	T	N	T	
Plec	Płeć właściciela	PlecD	T	N	T	
Nr_adresu	Numer adresu właściciela	Integer	T	N	T	

Tabela 38. Atrybuty relacji Wlasciciel

Poniżej prezentujemy związki, jakie zostały dodane między nowoutworzoną relacją Właściciele a innymi relacjami:

a) Związek między relacjami Wlasciciele-Adresy

Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Wlasciciele	0...1	Wlasciciel_posiada_adres	1...1	Adresy

Tabela 39. Związek Wlasciciele-Adresy

b) Związek między relacjami Kluby-Wlasciciele;

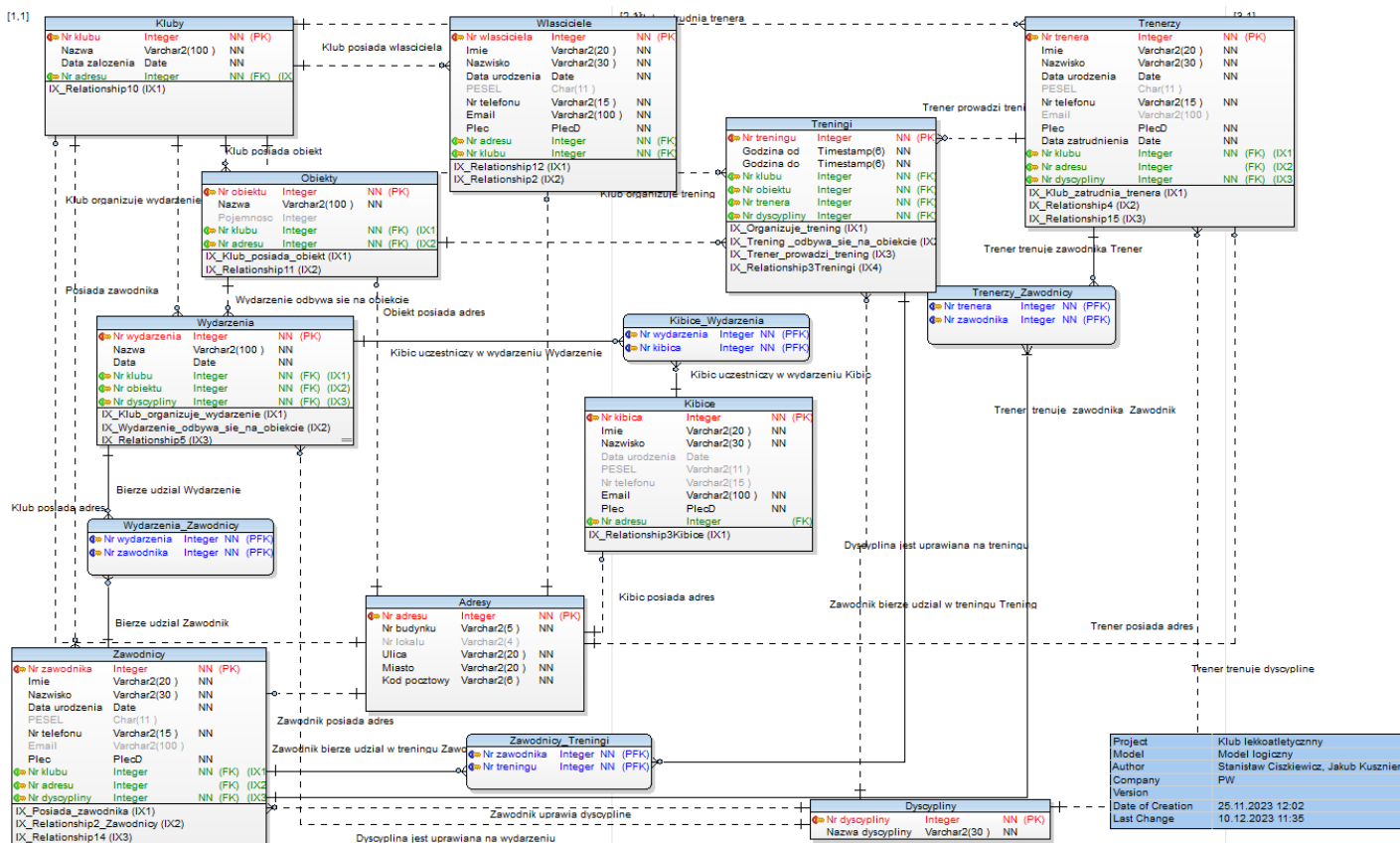
Nazwa encji	Krotność	Związek	Krotność	Nazwa encji
Kluby	1..1	Klub_posiada_wlasciciela	0..n	Wlasciciele

Tabela 40. Związek Kluby-Wlasciciele

4.3.3. Trzecia postać normalna

Ostatnim etapem procesu normalizacji jest trzecia postać normalna (istnieje więcej, ale zgodnie z zaleceniem prowadzącego skupiamy się jedynie na trzech pierwszych). Aby model logiczny znajdował się w trzeciej postaci normalnej musi być w drugiej postaci normalnej oraz żaden niekluczowy atrybut relacji nie może zależeć od innego atrybutu niekluczowego relacji. Idąc tym tropem zaczęliśmy analizować obecny stan naszej bazy danych i znaleźliśmy potencjalne miejsce, które moglibyśmy rozważyć podczas doprowadzania modelu do trzeciej postaci normalnej. Jest to atrybut płeć, który w przypadku imion jest zależny od imienia (a konkretnie jego ostatniej litery - litera "a" wskazuje na imię damskie). W tym celu można byłoby stworzyć oddzielną tabelę z imionami oraz płcią. Rozwiązanie to po głębszej analizie okazało się całkowicie nieadekwatne, gdyż rozwiązanie stałoby się błędne w przypadku coraz częściej pojawiających się imion pochodzenia zagranicznego (osoby mogłyby również pochodzić z zagranicy, co wtedy?). Zrezygnowaliśmy tym samym z tego pomysłu, jednak analizując resztę modelu nie znaleźliśmy podobnych zależności między atrybutami niekluczowymi - model był już w trzeciej postaci normalnej.

4.4. Schemat ER na poziomie modelu logicznego



Rys. 2. Schemat ER na poziomie modelu logicznego

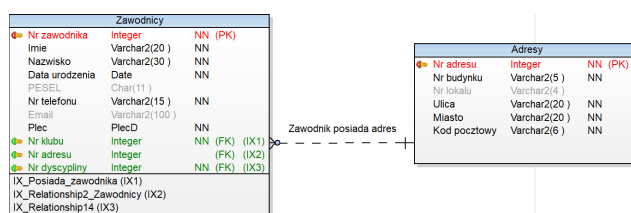
4.5. Wieży integralności

- każdy klucz główny jest postaci NOT_NULL oraz PRIMARY_KEY

- nazwy wszystkich relacji są różne, a żadna relacja nie posiada dwóch takich samych atrybutów
- dane muszą być wpisywane zgodnie z zadeklarowanym typem (podanym w modelu konceptualnym) - w szczególności wartość atrybutu płeć (Plec in('K','M')).
- określenie obowiązkowości występowania pól (musimy się liczyć z tym, że nie każdy musi posiadać elementy nieobowiązkowe, trzeba być gotowym na takie sytuacje)

4.6. Proces denormalizacji – analiza i przykłady

Ostatecznym etapem była analiza możliwości przeprowadzenia denormalizacji naszego modelu. Tym samym mieliśmy zastanowić się nad potencjalnym cofnięciem pewnych kroków poczynionych w ramach przeprowadzania procesu normalizacji w celu usprawnienia działania naszej bazy danych (niejako złamanie zasad normalizacji). Pierwszym krokiem była analiza czy w przypadku relacji Adres nie lepiej byłoby pozostać przy jednym, złożonym atrybucie w każdej relacji zawierającej pole adres.



Rys. 3. Podział na dwie relacje

Zawodnicy		
	Nr zawodnika	Integer NN (PK)
	Imie	Varchar2(20) NN
	Nazwisko	Varchar2(30) NN
	Data urodzenia	Date NN
	PESEL	Char(11) NN
	Nr telefonu	Varchar2(15) NN
	Email	Varchar2(100) NN
	Plec	PlecD NN
	Adres	Varchar2(200) NN
	Nr klubu	Integer NN (FK) (IX1)
	Nr dyscypliny	Integer NN (FK) (IX2)
IX_Posiada_zawodnika (IX1)		
IX_Relationship14 (IX2)		

Rys. 4. Możliwy efekt denormalizacji

Szybko zrezygnowaliśmy z tego pomysłu - byłoby to bardzo mało kompatybilne by przechowywać w każdym z obiektów oddzielnie dane adresowe (w dodatku w złożonej, nieatomowej formie - trudności z odwołaniem się do pewnych danych) - teoretycznie uzyskiwalibyśmy dane szybciej (operacje wyszukiwania trwałyby krócej), jednak ciężko byłoby odpowiednio zinterpretować fragmenty tak bardzo złożonego atrybutu.. Ten sam problem zidentyfikowaliśmy analizując relację Dyscypliny - cofnięcie poczynionych rozwiązań byłoby mało pomocne, efekty byłyby niezauważalne. Ostatecznie podjęliśmy decyzję o zaniechaniu próby denormalizowania naszego modelu - normalizacja nie skomplikowała go do takiego stopnia, by trzeba było kosztem łamania zasad usprawniać jego działanie. Proces denormalizacji zdałby egzamin w przypadku dużych baz danych, w których po normalizacji dochodzimy do niekompatybilności, trudności otrzymania odpowiedzi na zapytania (skomplikowana struktura itp.) - wtedy właśnie należy zastanowić się czy w przypadku bazy nie warto byłoby odstąpić od zasad normalizacyjnych.

5. Faza fizyczna

5.1. Projekt transakcji i weryfikacja ich wykonalności

Transakcja	Tabele	Czy możliwe?
Dodawanie, modyfikowanie i usuwanie zawodników	Zawodnicy, Treningi	Tak
Dodawanie, modyfikowanie i usuwanie trenerów	Trenerzy, Treningi	Tak
Dodawanie, modyfikowanie i usuwanie adresów	Adresy, Zawodnicy, Obiekty, Wlasciele, Kibice	Tak
Dodawanie, modyfikowanie i usuwanie obiektów	Obiekty, Wydarzenia, Treningi	Tak
Dodawanie, modyfikowanie i usuwanie treningów	Treningi	Tak
Dodawanie, modyfikowanie i usuwanie wydarzeń	Wydarzenia	Tak
Dodawanie, modyfikowanie i usuwanie kibiców	Kibice, Dyscypliny, Zawodnicy, Treningi, Trenerzy	Tak
Dodawanie, modyfikowanie i usuwanie właścicieli	Wlasciele, Klub	Tak

Tabela 41. Transakcje

5.2. Strojanie bazy danych – dobór indeksów

Kluby (Nr_adresu, Nr_wlasciciela):

```
CREATE INDEX IX_Relationship10 ON Kluby (Nr_adresu)
/
```

Trenerzy (Nr_klubu, Nr_adresu, Nr_dyscypliny)

```
CREATE INDEX IX_Klub_zatrudnia_trenera ON Trenerzy (Nr_klubu)
/
```

```
CREATE INDEX IX_Relationship4 ON Trenerzy (Nr_adresu)
/
```

```
CREATE INDEX IX_Relationship15 ON Trenerzy (Nr_dyscypliny)
/
```

Treningi (Nr_klubu, Nr_obiektu, Nr_trenera, Nr_dyscypliny)

```
CREATE INDEX IX_Organizuje_trening ON Treningi (Nr_klubu)
/
```

```
CREATE INDEX IX_Trening_odbywa_sie_na_obieckie ON Treningi (Nr_obiektu)
/
```

```
CREATE INDEX IX_Trener_prowadzi_trening ON Treningi (Nr_trenera)
/
```

```
CREATE INDEX IX_Relationship3Treningi ON Treningi (Nr_dyscypliny)
/
```

Zawodnicy (Nr_klubu, Nr_adresu, Nr_dyscypliny)

```
CREATE INDEX IX_Posiada_zawodnika ON Zawodnicy (Nr_klubu)
/
```

```
CREATE INDEX IX_Relationship2_Zawodnicy ON Zawodnicy (Nr_adresu)
/
```

```
CREATE INDEX IX_Relationship14 ON Zawodnicy (Nr_dyscypliny)
/
```

Wydarzenia (Nr_klubu, Nr_obiektu, Nr_dyscypliny)

```
CREATE INDEX IX_Klub_organizuje_wydarzenie ON Wydarzenia (Nr_klubu)
/
```

```
CREATE INDEX IX_Wydarzenie_odbywa_sie_na_obieckie ON Wydarzenia (Nr_obiektu)
/
```

```
CREATE INDEX IX_Relationship5 ON Wydarzenia (Nr_dyscypliny)
/
```


Obiekty (Nr_klubu, Nr_adresu)

```
CREATE INDEX IX_Klub_posiada_obiekt ON Obiekty (Nr_klubu)
/
```

```
CREATE INDEX IX_Relationship11 ON Obiekty (Nr_adresu)
/
```

Kibice (Nr_adresu)

```
CREATE INDEX IX_Relationship3Kibice ON Kibice (Nr_adresu)
/
```

Wlasciciele (Nr_adresu)

```
CREATE INDEX IX_Relationship12 ON Wlasciciele (Nr_adresu)
/
```

```
CREATE INDEX IX_Relationship2 ON Wlasciciele (Nr_klubu)
/
```

5.3. Skrypt SQL zakładający bazę danych

```
/*
Created: 25.11.2023
Modified: 10.12.2023
Project: Klub lekkoatletyczny
Model: Model logiczny
Company: PW
Author: Stanisław Ciszewicz, Jakub Kusznierek
Database: Oracle 18c
*/
```

— Create sequences section —

```
CREATE SEQUENCE Nr_klubuS
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```
CREATE SEQUENCE Nr_obiektuS
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```
CREATE SEQUENCE Nr_wydarzeniaS
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```
CREATE SEQUENCE Nr_wlascicielaS
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```
CREATE SEQUENCE Nr_zawodnikaS
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```
CREATE SEQUENCE Nr_adresuS
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```
CREATE SEQUENCE Nr_kibicaS
NOMAXVALUE
NOMINVALUE
CACHE 20
/
```

```

CREATE SEQUENCE Nr_dyscyplinyS
NOMAXVALUE
NOMINVALUE
CACHE 20
/

CREATE SEQUENCE Nr_treninguS
NOMAXVALUE
NOMINVALUE
CACHE 20
/

CREATE SEQUENCE Nr_treneraS
NOMAXVALUE
NOMINVALUE
CACHE 20
/

-- Create tables section -----

-- Table Kluby

CREATE TABLE Kluby(
    Nr_klubu Integer NOT NULL,
    Nazwa Varchar2(100 ) NOT NULL,
    Data_zalozenia Date NOT NULL,
    Nr_adresu Integer NOT NULL
)
/

-- Create indexes for table Kluby

CREATE INDEX IX_Relationship10 ON Kluby (Nr_adresu)
/

-- Add keys for table Kluby

ALTER TABLE Kluby ADD CONSTRAINT PK_Kluby PRIMARY KEY (Nr_klubu)
/

-- Table Trenerzy

CREATE TABLE Trenerzy(
    Nr_trenera Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Data_urodzenia Date NOT NULL,
    PESEL Char(11 ),
    Nr_telefonu Varchar2(15 ) NOT NULL,
    Email Varchar2(100 ),
    Plec Char(256 ) NOT NULL
    CHECK (Plec in ('K','M')),
    Data_zatrudnienia Date NOT NULL,
    Nr_klubu Integer NOT NULL,
    Nr_adresu Integer,
    Nr_dyscypliny Integer NOT NULL
)
/

-- Create indexes for table Trenerzy

CREATE INDEX IX_Klub_zatrudnia_trenera ON Trenerzy (Nr_klubu)
/

CREATE INDEX IX_Relationship4 ON Trenerzy (Nr_adresu)
/

CREATE INDEX IX_Relationship15 ON Trenerzy (Nr_dyscypliny)
/

-- Add keys for table Trenerzy

ALTER TABLE Trenerzy ADD CONSTRAINT PK_Trenerzy PRIMARY KEY (Nr_trenera)
/

-- Table Treningi

CREATE TABLE Treningi(
    Nr_treningu Integer NOT NULL,
    Godzina_od Timestamp(6) NOT NULL,
    Godzina_do Timestamp(6) NOT NULL,
    Nr_klubu Integer NOT NULL,
    Nr_obiektu Integer NOT NULL,

```

```

    Nr_trenera Integer NOT NULL,
    Nr_dyscypliny Integer NOT NULL
)
/

-- Create indexes for table Treningi

CREATE INDEX IX_Organizuje_trening ON Treningi (Nr_klubu)
/

CREATE INDEX IX_Trening_odbywa_sie_na_obiekcie ON Treningi (Nr_obiektu)
/

CREATE INDEX IX_Trener_prowadzi_trening ON Treningi (Nr_trenera)
/

CREATE INDEX IX_Relationship3Treningi ON Treningi (Nr_dyscypliny)
/

-- Add keys for table Treningi

ALTER TABLE Treningi ADD CONSTRAINT PK_Treningi PRIMARY KEY (Nr_treningu)
/

-- Table Zawodnicy

CREATE TABLE Zawodnicy(
    Nr_zawodnika Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Data_urodzenia Date NOT NULL,
    PESEL Char(11 ),
    Nr_telefonu Varchar2(15 ) NOT NULL,
    Email Varchar2(100 ),
    Plec Char(256 ) NOT NULL
        CHECK (Plec in ( 'K', 'M' ) ),
    Nr_klubu Integer NOT NULL,
    Nr_adresu Integer,
    Nr_dyscypliny Integer NOT NULL
)
/

-- Create indexes for table Zawodnicy

CREATE INDEX IX_Posiada_zawodnika ON Zawodnicy (Nr_klubu)
/

CREATE INDEX IX_Relationship2_Zawodnicy ON Zawodnicy (Nr_adresu)
/

CREATE INDEX IX_Relationship14 ON Zawodnicy (Nr_dyscypliny)
/

-- Add keys for table Zawodnicy

ALTER TABLE Zawodnicy ADD CONSTRAINT PK_Zawodnika PRIMARY KEY (Nr_zawodnika)
/

-- Table Wydarzenia

CREATE TABLE Wydarzenia(
    Nr_wydarzenia Integer NOT NULL,
    Nazwa Varchar2(100 ) NOT NULL,
    Data Date NOT NULL,
    Nr_klubu Integer NOT NULL,
    Nr_obiektu Integer NOT NULL,
    Nr_dyscypliny Integer NOT NULL
)
/

-- Create indexes for table Wydarzenia

CREATE INDEX IX_Klub_organizuje_wydarzenie ON Wydarzenia (Nr_klubu)
/

CREATE INDEX IX_Wydarzenie_odbywa_sie_na_obiekcie ON Wydarzenia (Nr_obiektu)
/

CREATE INDEX IX_Relationship5 ON Wydarzenia (Nr_dyscypliny)
/

-- Add keys for table Wydarzenia

ALTER TABLE Wydarzenia ADD CONSTRAINT PK_Wydarzenia PRIMARY KEY (Nr_wydarzenia)

```

```

/
— Table Obiekty
CREATE TABLE Obiekty(
    Nr_obiektu Integer NOT NULL,
    Nazwa Varchar2(100 ) NOT NULL,
    Pojemnosc Integer,
    Nr_klubu Integer NOT NULL,
    Nr_adresu Integer NOT NULL
)
/
— Create indexes for table Obiekty
CREATE INDEX IX_Klub_posiada_obiekt ON Obiekty (Nr_klubu)
/
CREATE INDEX IX_Relationship11 ON Obiekty (Nr_adresu)
/
— Add keys for table Obiekty
ALTER TABLE Obiekty ADD CONSTRAINT PK_Obiekty PRIMARY KEY (Nr_obiektu)
/
— Table Kibice
CREATE TABLE Kibice(
    Nr_kibica Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Data_urodzenia Date,
    PESEL Varchar2(11 ),
    Nr_telefonu Varchar2(15 ),
    Email Varchar2(100 ) NOT NULL,
    Plec Char(256 ) NOT NULL
    CHECK (Plec in ('K','M')),
    Nr_adresu Integer
)
/
— Create indexes for table Kibice
CREATE INDEX IX_Relationship3Kibice ON Kibice (Nr_adresu)
/
— Add keys for table Kibice
ALTER TABLE Kibice ADD CONSTRAINT PK_Kibice PRIMARY KEY (Nr_kibica)
/
— Table Trenerzy_Zawodnicy
CREATE TABLE Trenerzy_Zawodnicy(
    Nr_trenera Integer NOT NULL,
    Nr_zawodnika Integer NOT NULL
)
/
— Table Zawodnicy_Treningi
CREATE TABLE Zawodnicy_Treningi(
    Nr_zawodnika Integer NOT NULL,
    Nr_treningu Integer NOT NULL
)
/
— Table Kibice_Wydarzenia
CREATE TABLE Kibice_Wydarzenia(
    Nr_wydarzenia Integer NOT NULL,
    Nr_kibica Integer NOT NULL
)
/
— Table Wydarzenia_Zawodnicy
CREATE TABLE Wydarzenia_Zawodnicy(
    Nr_wydarzenia Integer NOT NULL,
    Nr_zawodnika Integer NOT NULL
)
/

```

```

— Table Adresy

CREATE TABLE Adresy(
    Nr_adresu Integer NOT NULL,
    Nr_budynku Varchar2(5 ) NOT NULL,
    Nr_lokalu Varchar2(4 ),
    Ulica Varchar2(20 ) NOT NULL,
    Miasto Varchar2(20 ) NOT NULL,
    Kod_pocztowy Varchar2(6 ) NOT NULL
)
/

— Add keys for table Adresy

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

— Table Wlasciciele

CREATE TABLE Wlasciciele(
    Nr_wlasciciela Integer NOT NULL,
    Imie Varchar2(20 ) NOT NULL,
    Nazwisko Varchar2(30 ) NOT NULL,
    Data_urodzenia Date NOT NULL,
    PESEL Char(11 ),
    Nr_telefonu Varchar2(15 ) NOT NULL,
    Email Varchar2(100 ) NOT NULL,
    Plec Char(256 ) NOT NULL
    CHECK (Plec in ('K','M')),
    Nr_adresu Integer NOT NULL,
    Nr_klubu Integer NOT NULL
)
/

— Create indexes for table Wlasciciele

CREATE INDEX IX_Relationship12 ON Wlasciciele (Nr_adresu)
/

CREATE INDEX IX_Relationship2 ON Wlasciciele (Nr_klubu)
/

— Add keys for table Wlasciciele

ALTER TABLE Wlasciciele ADD CONSTRAINT PK_Wlasciciele PRIMARY KEY (Nr_wlasciciela)
/

— Table Dyscypliny

CREATE TABLE Dyscypliny(
    Nr_dyscypliny Integer NOT NULL,
    Nazwa_dyscypliny Varchar2(30 ) NOT NULL
)
/

— Add keys for table Dyscypliny

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

— Trigger for sequence Nr_klubuS for column Nr_klubu in table Kluby —————
CREATE OR REPLACE TRIGGER ts_Kluby_Nr_klubuS BEFORE INSERT
ON Kluby FOR EACH ROW
BEGIN
    :new.Nr_klubu := Nr_klubuS.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Kluby_Nr_klubuS AFTER UPDATE OF Nr_klubu
ON Kluby FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(−20010,'Cannot -update -column -Nr_klubu -in -table -Kluby -as -it -uses -sequence. ');
END;
/

— Trigger for sequence Nr_treneraS for column Nr_trenera in table Trenerzy —————
CREATE OR REPLACE TRIGGER ts_Trenerzy_Nr_treneraS BEFORE INSERT
ON Trenerzy FOR EACH ROW
BEGIN
    :new.Nr_trenera := Nr_treneraS.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Trenerzy_Nr_treneraS AFTER UPDATE OF Nr_trenera
ON Trenerzy FOR EACH ROW
BEGIN

```

```

    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_trenera in table
--Trenerzy as it uses sequence. ');
END;
/

-- Trigger for sequence Nr_treninguS for column Nr_treningu in table Treningi -----
CREATE OR REPLACE TRIGGER ts_Treningi_Nr_treninguS BEFORE INSERT
ON Treningi FOR EACH ROW
BEGIN
    :new.Nr_treningu := Nr_treninguS.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Treningi_Nr_treninguS AFTER UPDATE OF Nr_treningu
ON Treningi FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_treningu in table
--Treningi as it uses sequence. ');
END;
/

-- Trigger for sequence Nr_zawodnikaS for column Nr_zawodnika in table Zawodnicy -----
CREATE OR REPLACE TRIGGER ts_Zawodnicy_Nr_zawodnikaS BEFORE INSERT
ON Zawodnicy FOR EACH ROW
BEGIN
    :new.Nr_zawodnika := Nr_zawodnikaS.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Zawodnicy_Nr_zawodnikaS AFTER UPDATE OF Nr_zawodnika
ON Zawodnicy FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_zawodnika in table
--Zawodnicy as it uses sequence. ');
END;
/

-- Trigger for sequence Nr_wydarzeniaS for column Nr_wydarzenia in table Wydarzenia -----
CREATE OR REPLACE TRIGGER ts_Wydarzenia_Nr_wydarzeniaS BEFORE INSERT
ON Wydarzenia FOR EACH ROW
BEGIN
    :new.Nr_wydarzenia := Nr_wydarzeniaS.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wydarzenia_Nr_wydarzeniaS AFTER UPDATE OF Nr_wydarzenia
ON Wydarzenia FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_wydarzenia in table
--Wydarzenia as it uses sequence. ');
END;
/

-- Trigger for sequence Nr_obiektuS for column Nr_obiektu in table Obiekty -----
CREATE OR REPLACE TRIGGER ts_Obiekty_Nr_obiektuS BEFORE INSERT
ON Obiekty FOR EACH ROW
BEGIN
    :new.Nr_obiektu := Nr_obiektuS.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Obiekty_Nr_obiektuS AFTER UPDATE OF Nr_obiektu
ON Obiekty FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_obiektu in table Obiekty as it uses sequence. ');
END;
/

-- Trigger for sequence Nr_kibicaS for column Nr_kibica in table Kibice -----
CREATE OR REPLACE TRIGGER ts_Kibice_Nr_kibicaS BEFORE INSERT
ON Kibice FOR EACH ROW
BEGIN
    :new.Nr_kibica := Nr_kibicaS.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Kibice_Nr_kibicaS AFTER UPDATE OF Nr_kibica
ON Kibice FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_kibica in table Kibice as it uses sequence. ');
END;
/

-- Trigger for sequence Nr_adresuS for column Nr_adresu in table Adresy -----
CREATE OR REPLACE TRIGGER ts_Adresy_Nr_adresuS BEFORE INSERT
ON Adresy FOR EACH ROW
BEGIN
    :new.Nr_adresu := Nr_adresuS.nextval;
END;

```

```

/
CREATE OR REPLACE TRIGGER tsu_Adresy_Nr_adresuS 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 uses sequence. ');
END;
/

-- Trigger for sequence Nr_wlascicielaS for column Nr_wlasciciela in table Wlasciciela
CREATE OR REPLACE TRIGGER ts_Wlasciciela_Nr_wlascicielaS BEFORE INSERT
ON Wlasciciela FOR EACH ROW
BEGIN
    :new.Nr_wlasciciela := Nr_wlascicielaS.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Wlasciciela_Nr_wlascicielaS AFTER UPDATE OF Nr_wlasciciela
ON Wlasciciela FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_wlasciciela in table Wlasciciela as
-- it uses sequence. ');
END;
/

-- Trigger for sequence Nr_dyscyplinyS for column Nr_dyscypliny in table Dyscypliny
CREATE OR REPLACE TRIGGER ts_Dyscypliny_Nr_dyscyplinyS BEFORE INSERT
ON Dyscypliny FOR EACH ROW
BEGIN
    :new.Nr_dyscypliny := Nr_dyscyplinyS.nextval;
END;
/
CREATE OR REPLACE TRIGGER tsu_Dyscypliny_Nr_dyscyplinyS AFTER UPDATE OF Nr_dyscypliny
ON Dyscypliny FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20010, 'Cannot update column Nr_dyscypliny in table Dyscypliny
-- as it uses sequence. ');
END;
/

-- Create foreign keys (relationships) section
ALTER TABLE Zawodnicy ADD CONSTRAINT Posiada_zawodnika FOREIGN KEY (Nr_klubu) REFERENCES Kluby (Nr_klubu)
/

ALTER TABLE Treningi ADD CONSTRAINT Klub_organizuje_trening FOREIGN KEY (Nr_klubu) REFERENCES Kluby (Nr_klubu)
/

ALTER TABLE Trenerzy ADD CONSTRAINT Klub_zatrudnia_trenera FOREIGN KEY (Nr_klubu) REFERENCES Kluby (Nr_klubu)
/

ALTER TABLE Treningi ADD CONSTRAINT Trening_odbywa_sie_na_obiekcie FOREIGN KEY (Nr_obiektu) REFERENCES
Obiekty (Nr_obiektu)
/

ALTER TABLE Treningi ADD CONSTRAINT Trener_prowadzi_trening FOREIGN KEY (Nr_trenera)
REFERENCES Trenerzy (Nr_trenera)
/

ALTER TABLE Obiekty ADD CONSTRAINT Klub_posiada_obiekt FOREIGN KEY (Nr_klubu)
REFERENCES Kluby (Nr_klubu)
/

ALTER TABLE Wydarzenia ADD CONSTRAINT Klub_organizuje_wydarzenie FOREIGN KEY (Nr_klubu)
REFERENCES Kluby (Nr_klubu)
/

ALTER TABLE Wydarzenia ADD CONSTRAINT Wydarzenie_odbywa_sie_na_obiekcie FOREIGN KEY (Nr_obiektu)
REFERENCES Obiekty (Nr_obiektu)
/

```

```
ALTER TABLE Zawodnicy ADD CONSTRAINT Zawodnik_posiada_adres FOREIGN KEY (Nr_adresu)
REFERENCES Adresy (Nr_adresu)
/
```

```
ALTER TABLE Kibice ADD CONSTRAINT Kibic_posiada_adres FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)
/
```

```
ALTER TABLE Trenerzy ADD CONSTRAINT Trener_posiada_adres FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)
/
```

```
ALTER TABLE Kluby ADD CONSTRAINT Klub_posiada_adres FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)
/
```

```
ALTER TABLE Obiekty ADD CONSTRAINT Obiekt_posiada_adres FOREIGN KEY (Nr_adresu) REFERENCES Adresy (Nr_adresu)
/
```

```
ALTER TABLE Wlasciciele ADD CONSTRAINT Wlasciciel_posiada_adres FOREIGN KEY (Nr_adresu)
REFERENCES Adresy (Nr_adresu)
/
```

```
ALTER TABLE Zawodnicy ADD CONSTRAINT Zawodnik_uprawia_dyscypline FOREIGN KEY (Nr_dyscypliny)
REFERENCES Dyscypliny (Nr_dyscypliny)
/
```

```
ALTER TABLE Trenerzy ADD CONSTRAINT Trener_trenuje_dyscypline FOREIGN KEY (Nr_dyscypliny)
REFERENCES Dyscypliny (Nr_dyscypliny)
/
```

```
ALTER TABLE Treningi ADD CONSTRAINT Dyscyplina_jest_uprawiana_na_treningu FOREIGN KEY (Nr_dyscypliny)
REFERENCES Dyscypliny (Nr_dyscypliny)
/
```

```
ALTER TABLE Wydarzenia ADD CONSTRAINT Dyscyplina_jest_uprawiana_na_wydarzeniu FOREIGN KEY (Nr_dyscypliny)
REFERENCES Dyscypliny (Nr_dyscypliny)
/
```

```
ALTER TABLE Wlasciciele ADD CONSTRAINT Klub_posiada_wlasciciela FOREIGN KEY (Nr_klubu)
REFERENCES Kluby (Nr_klubu)
/
```

5.4. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych

5.4.1. Wpisywanie danych do bazy

```
INSERT INTO ADRESY (NR_BUDYNKU, NR_LOKALU, ULICA, MIASTO, KOD_POCZTOWY)
VALUES (50, 17, 'Mysia', 'Warszawa', '02-650');
```

```
INSERT INTO ADRESY (NR_BUDYNKU, NR_LOKALU, ULICA, MIASTO, KOD_POCZTOWY)
VALUES (40, 7, 'Kocia', 'Bia Cystok', '15-651');
```

```
INSERT INTO KLUBY (NAZWA, DATA_ZALOZENIA, NR_ADRESU)
VALUES ('GRUPA-LAMPART', TO_DATE('1980-10-17', 'YYYY-MM-DD'), 2);
```

```
INSERT INTO WLASCICIELE (IMIE, NAZWISKO, DATA_URODZENIA, PESEL, NR_TELEFONU, EMAIL, PLEC, NR_ADRESU, NR_KLUBU)
VALUES ('Jan', 'Kowalski', TO_DATE('2000-11-25', 'YYYY-MM-DD'), '12345678912', '48111222333',
'jan.kowalski@wp.pl', 'M', 1, 1);
```



```

INSERT INTO DYSCYPLINY (NAZWA_DYSCYPLINY)
VALUES ('Rzut młotem');

INSERT INTO ADRESY (NR_BUDYNKU, NR_LOKALU, ULICA, MIASTO, KOD_POCZTOWY)
VALUES (30, 1, 'Dzicza', 'Maćkinia', '12-421'); --adres trenera

INSERT INTO TRENERZY (IMIE, NAZWISKO, DATA_URODZENIA, PESEL, NR_TELEFONU, EMAIL, PLEC, DATA_ZATRUDNIENIA,
NR_KLUBU, NR_ADRESU, NR_DYSCYPLINY)
VALUES ('Wojciech', 'Piłkowski', TO_DATE('1972-09-06', 'YYYY-MM-DD'), '27380461736', '+48963481204',
'woj.pia@wp.pl', 'M', TO_DATE('2012-04-18', 'YYYY-MM-DD'), 1, 3, 1);

INSERT INTO ADRESY (NR_BUDYNKU, NR_LOKALU, ULICA, MIASTO, KOD_POCZTOWY)
VALUES (35, NULL, 'Główna', 'Kraków', '30-500');

INSERT INTO TRENERZY (IMIE, NAZWISKO, DATA_URODZENIA, PESEL, NR_TELEFONU, EMAIL, PLEC, DATA_ZATRUDNIENIA,
NR_KLUBU, NR_ADRESU, NR_DYSCYPLINY)
VALUES ('Adrian', 'Siemieniec', TO_DATE('1978-04-11', 'YYYY-MM-DD'), '27673461149', '+48347481764',
'adi.si@wp.pl', 'M', TO_DATE('2010-05-16', 'YYYY-MM-DD'), 1, 4, 1);

--3 trener
INSERT INTO DYSCYPLINY (NAZWA_DYSCYPLINY)
VALUES ('Bieg na 100m');

INSERT INTO ADRESY (NR_BUDYNKU, NR_LOKALU, ULICA, MIASTO, KOD_POCZTOWY)
VALUES (15, 3, 'Mickiewicza', 'Wrocław', '50-101');

INSERT INTO TRENERZY (IMIE, NAZWISKO, DATA_URODZENIA, PESEL, NR_TELEFONU, EMAIL, PLEC, DATA_ZATRUDNIENIA,
NR_KLUBU, NR_ADRESU, NR_DYSCYPLINY)
VALUES ('Grzegorz', 'Nowak', TO_DATE('1985-12-10', 'YYYY-MM-DD'), '85121056789', '+48666778899',
'grzegorz.nowak@coach.com', 'M', TO_DATE('2015-07-01', 'YYYY-MM-DD'), 1, 5, 2);

INSERT INTO ADRESY (NR_BUDYNKU, NR_LOKALU, ULICA, MIASTO, KOD_POCZTOWY)
VALUES (40, 5, 'Słowackiego', 'Warszawa', '00-001');

INSERT INTO ZAWODNICY (IMIE, NAZWISKO, DATA_URODZENIA, PESEL, NR_TELEFONU, EMAIL, PLEC, NR_KLUBU, NR_ADRESU,
NR_DYSCYPLINY)
VALUES ('Tomasz', 'Nowak', TO_DATE('1997-7-2'), '56783947401', '+48346952389',
'tomek.nowak@wp.pl', 'M', 1, 6, 1);

INSERT INTO ADRESY (NR_BUDYNKU, NR_LOKALU, ULICA, MIASTO, KOD_POCZTOWY)
VALUES (63, 15, 'Mickiewicza', 'Szczecin', '13-671');

INSERT INTO ZAWODNICY (IMIE, NAZWISKO, DATA_URODZENIA, PESEL, NR_TELEFONU, EMAIL, PLEC, NR_KLUBU, NR_ADRESU,
NR_DYSCYPLINY)
VALUES ('Robert', 'Lewangolski', TO_DATE('1999-8-31'), '06284717303', '+48638012745',
'robercik.strzelagole@wp.pl', 'M', 1, 7, 2);

INSERT INTO ADRESY (NR_BUDYNKU, NR_LOKALU, ULICA, MIASTO, KOD_POCZTOWY)
VALUES (1, NULL, 'Słoneczna', 'Warszawa', '00-852');

INSERT INTO OBIEKTY (NAZWA, POJEMNOSC, NR_KLUBU, NR_ADRESU)
VALUES ('PGE Narodowy', 60000, 1, 8);

INSERT INTO TRENINGI (GODZINA_OD, GODZINA_DO, NR_KLUBU, NR_OBIEKTU, NR_TRENERA, NR_DYSCYPLINY)
VALUES ('2023-11-26 14:30:00', '2023-11-26 16:30:00', 1, 1, 1, 1);

INSERT INTO WYDARZENIA (NAZWA, DATA, NR_KLUBU, NR_OBIEKTU, NR_DYSCYPLINY)
VALUES ('Mistrzostwa lekkoatletyczne', TO_DATE('2023-9-15'), 1, 1, 1);

INSERT INTO WYDARZENIA_ZAWODNICY (NR_WYDARZENIA, NR_ZAWODNIKA)
VALUES (1, 1);

INSERT INTO ADRESY (NR_BUDYNKU, NR_LOKALU, ULICA, MIASTO, KOD_POCZTOWY)
VALUES (2, NULL, 'Wojskowa', 'Warszawa', '00-325');

INSERT INTO KIBICE (IMIE, NAZWISKO, DATA_URODZENIA, PESEL, NR_TELEFONU, EMAIL, PLEC, NR_ADRESU)
VALUES ('Mariusz', 'Tomaszewski', TO_DATE('1983-7-12'), '98638310541', '+48764312890',
'maru.tomek@wp.pl', 'M', 9);

INSERT INTO KIBICE_WYDARZENIA (NR_WYDARZENIA, NR_KIBICA)
VALUES (1, 1);

INSERT INTO TRENERZY_ZAWODNICY (NR_TRENERA, NR_ZAWODNIKA)
VALUES (1, 1);

INSERT INTO TRENERZY_ZAWODNICY (NR_TRENERA, NR_ZAWODNIKA)
VALUES (3, 2);

INSERT INTO ZAWODNICY_TRENINGI (NR_ZAWODNIKA, NR_TRENINGU)
VALUES(1,1);

```

5.4.2. Przykładowe zapytania do bazy

Wypisz liczbę adresów zapisanych w bazie:

```
SELECT COUNT(*) FROM ADRESY;
```

	COUNT(*)
1	9

Rys. 5. Rezultat dla powyższego zapytania

Wypisz imię i nazwisko wszystkich zawodników biorących udział w wydarzeniu:

```
SELECT Z.Imie , Z.Nazwisko
FROM Zawodnicy Z
JOIN Wydarzenia_Zawodnicy WZ ON Z.Nr_zawodnika = WZ.Nr_zawodnika
WHERE WZ.Nr_wydarzenia = '1';
```

	IMIE	NAZWISKO
1	Tomasz	Nowak

Rys. 6. Rezultat dla powyższego zapytania

Wypisz imię, nazwisko oraz numer PESEL trenerów (i zawodników), którzy mają przypisanych zawodników.

```
SELECT
Trenerzy.Imie AS Trener_Imie ,
Trenerzy.Nazwisko AS Trener_Nazwisko ,
Trenerzy.PESEL AS PESEL_Trenera ,
Zawodnicy.Imie AS Zawodnik_Imie ,
Zawodnicy.Nazwisko AS Zawodnik_Nazwisko ,
Zawodnicy.PESEL AS PESEL_Zawodnika
FROM
Trenerzy_Zawodnicy
JOIN
Trenerzy ON Trenerzy_Zawodnicy.Nr_trenera = Trenerzy.Nr_trenera
JOIN
Zawodnicy ON Trenerzy_Zawodnicy.Nr_zawodnika = Zawodnicy.Nr_zawodnika;
```

	TRENER_IMIE	TRENER_NAZWISKO	PESEL_TRENERA	ZAWODNIK_IMIE	ZAWODNIK_NAZWISKO	PESEL_ZAWODNIKA
1	Wojciech	Piątkowski	27380461736	Tomasz	Nowak	56783947401
2	Grzegorz	Nowak	85121056789	Robert	Lewangolski	06284717303

Rys. 7. Rezultat dla powyższego zapytania

Policz ile adresów ma ustawione miasto na Warszawa:

```
SELECT COUNT(*)
FROM Adresy
WHERE Miasto = 'Warszawa';
```

	COUNT(*)
1	4

Rys. 8. Rezultat dla powyższego zapytania

Wypisz zawodników biorących udział w treningu oraz godzinę rozpoczęcia i zakończenia treningu:

```
SELECT
Zawodnicy.Nr_zawodnika ,
Zawodnicy.Imie ,
Zawodnicy.Nazwisko ,
Treningi.Nr_treningu ,
Treningi.Godzina_od ,
```

```

Treningi.Godzina_do
FROM
Zawodnicy_Treningi
JOIN
Zawodnicy ON Zawodnicy_Treningi.Nr_zawodnika = Zawodnicy.Nr_zawodnika
JOIN
Treningi ON Zawodnicy_Treningi.Nr_treningu = Treningi.Nr_treningu;

```

	NR_ZAWODNIKA	IMIE	NAZWISKO	NR_TRENINGU	GODZINA_OD	GODZINA_DO
1	1	Tomasz	Nowak	1	23/11/26 14:30:00,000000000	23/11/26 16:30:00,000000000

Rys. 9. Rezultat dla powyższego zapytania