

Klub Tenisowy

DOKUMENTACJA

Maciej Rytel

Spis treści

1. Zakres i cel projektu
 - 1.1 Założenia funkcjonalne
2. Definicja Systemu
 - 2.1 Perspektywy użytkowników
 - 2.2 Obszary systemu
3. Model konceptualny
 - 3.1 Definicja zbiorów encji określonych w projekcie oraz określenie ich atrybutów i dziedzin
 - 3.2 Ustalenie związków między encjami i ich typów
 - 3.3 Klucze kandydujące i główne
 - 3.4 Schemat ER na poziomie konceptualnym
 - 3.5 Problemy pułapek szczelinowych i wachlarzowych
4. Model logiczny
 - 4.1 Charakterystyka modelu relacyjnego
 - 4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym
 - 4.3 Proces normalizacji
 - 4.4 Schemat ER na poziomie modelu logicznego
 - 4.5 Więzy integralności
 - 4.6 Proces denormalizacji
5. Faza fizyczna
 - 5.1 Projekt transakcji i weryfikacja ich wykonalności
 - 5.2 Strojenie bazy danych
 - 5.3 Skrypt SQL zakładający bazę danych
 - 5.4 Przykłady zapytań SQL do bazy danych

1. Zakres i cel projektu

Celem projektu jest zaprojektowanie, na poziomie konceptualnym i logicznym, a także fizyczna implementacja relacyjnej bazy danych

Baza danych umożliwia obsługę i zarządzanie klubem tenisowym. Zostanie ona oparta o system zarządzania bazami danych firmy Oracle.

Oprogramowanie użyte podczas realizacji projektu:

- Oracle Database 19c
- Toad Data Modeler
- Oracle SQL Developer

1.1 Założenia funkcjonalne

Projekt dotyczy klubu tenisowego i placówek, które się na niego składają. Każda z placówek oferuje usługi, takie jak wynajem kortu tenisowego czy lekcja gry w tenisa. Klient korzystając z usługi udostępnia klubowi potrzebne dane, które trafiają do bazy danych, a w szczególnym przypadku może zostać on zawodnikiem klubu. Wtedy jego dane są uzupełniane o dodatkowe informacje.

Każda z placówek zatrudnia pracowników, takich jak trener czy pracownik administracyjny. Ich dane trafiają do bazy danych i są uzupełniane o wynagrodzenie. Trenerzy posiadają swój grafik i uprawnienia trenerskie, natomiast pracownicy administracyjni mają określone stanowiska.

2. Definicja systemu

2.1 Perspektywy użytkowników

1. Administrator – ma dostęp do wszystkich danych
2. Prezes – ma wgląd we wszystkie dane, ale nie ma możliwości ingerencji w strukturę bazy
3. Kierownik placówki – ma dostęp do danych związanych z placówką. Nie ma dostępu do danych klienta
4. Trener – ma dostęp do swoich danych, wgląd do danych zawodnika klubu oraz usług oferowanych przez placówkę
5. Pracownik administracyjny – ma dostęp do danych związanych z pracownikami oraz klientami
6. Klient – ma wgląd do swoich danych oraz do usług oferowanych przez placówkę

2.2 Obszary systemu

Funkcjonalność	Administrator	Prezes	Kierownik placówki	Trener	Pracownik administracyjny	Klient
Lista pracowników						
Podgląd, dodawanie, usuwanie, modyfikowanie danych pracownika	Tak	Tak	Tak	Tylko podgląd	Tak	Nie
Podgląd, dodawanie, usuwanie, modyfikowanie wynagrodzeń pracownika	Tak	Tak	Tak	Tylko podgląd	Tak	Nie
Podgląd, dodawanie, usuwanie, modyfikowanie grafiku trenera	Tak	Tylko podgląd	Tylko podgląd	Tak	Tak	Tylko podgląd
Lista placówek						
Podgląd, dodawanie, usuwanie, modyfikowanie danych placówki	Tak	Tak	Tak	Nie	Tylko podgląd	Nie
Lista Klientów						
Podgląd, dodawanie, usuwanie, modyfikowanie danych klienta	Tak	Tylko podgląd	Nie	Nie	Tak	Tylko podgląd
Podgląd, dodawanie, usuwanie, modyfikowanie danych zawodnika	Tak	Tylko podgląd	Nie	Tylko podgląd	Tak	Tylko podgląd
Usługi						
Podgląd, dodawanie, usuwanie, modyfikowanie oferowanej usługi	Tak	Tylko podgląd	Tak	Tylko podgląd	Tak	Tylko podgląd
Podgląd, dodawanie, usuwanie przypisania pracownika do usługi	Tak	Tylko podgląd	Nie	Tylko podgląd	Tak	Tylko podgląd
Administracja bazy danych						
dodawanie, modyfikacja, usuwanie danych o rodzajach stanowisk	Tak	Tak	Tak	Nie	Tak	Nie
dodawanie, modyfikacja, usuwanie danych o klubie	Tak	Tak	Nie	Nie	Nie	Nie

3. Model konceptualny

3.1 Definicja zbiorów encji określonych w projekcie oraz określenie ich atrybutów i dziedzin

Encja Klub

Atrybuty tej encji przedstawiają dane dotyczące firmy, jej nazwę, numer KRS, numer NIP, dane adresowe.

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
YES	Id_klubu		Integer	YES	ID biura - unikatowy identyfikator
NO	Nazwa		VarChar(20)	YES	Nazwa klubu
NO	Data_zalozenia		Date	YES	Data założenia klubu
NO	KRS		Integer	YES	Numer krajowego rejestru sądowego
NO	NIP		Integer	YES	Numer identyfikacji podatkowej
NO	Miasto		VarChar(20)	YES	Miasto w którym znajduje się klub
NO	Ulica		VarChar(20)	YES	Ulica przy której znajduje się klub
NO	Nr_budynku		VarChar(10)	YES	Numer budynku w którym znajduje się klub
NO	Nr_lokalu		VarChar(10)	NO	Numer lokalu w którym znajduje się klub
NO	Kod_pocztowy		VarChar(6)	YES	Kod pocztowy miasta w którym znajduje się klub
NO	Telefon		Integer	YES	Telefon kontaktowy
NO	Email		VarChar(30)	YES	Adres e-mail

Encja Placowka

W tej encji zawarte są podstawowe dane na temat placówki, czyli adres, dane kontaktowe czy np. ilość kortów jaka znajdują się przy danej placówce

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
YES	Id_placowki		Integer	YES	ID placówki - unikalny identyfikator
NO	Miasto		VarChar(20)	YES	Miasto w którym znajduje się kort
NO	Ulica		VarChar(20)	YES	Ulica przy której znajduje się kort
NO	Nr_budynku		VarChar(10)	YES	Numer budynku kortu
NO	Kod_pocztowy		VarChar(6)	YES	Kod pocztowy miasta w którym znajduje się kort
NO	Liczba_kortow		Integer	YES	Liczba kortów posiadanych przez placówkę
NO	Odnowa_biologiczna	Odnowa_biologiczna	Character(1)	YES	Czy w danej placówce znajduje się aparatura do odnowy biologicznej
NO	Silownia	Silownia	Character(1)	YES	Czy w placówce znajduje się siłownia
NO	Nr_lokalu		VarChar(10)	NO	Numer lokalu w którym znajduje się placówka
NO	Telefon		Integer	YES	Telefon
NO	Email		VarChar(30)	YES	Adres e-mail

Encja Pracownik

Encja ta przechowuje dane osobowe pracownika danej placówki

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
YES	Id_pracownika		Integer	YES	ID pracownika - unikalny identyfikator
NO	Imie		VarChar(20)	YES	Imie pracownika
NO	Nazwisko		VarChar(20)	YES	Nazwisko pracownika
NO	Drugie_Imie		VarChar(20)	NO	Drugie imie pracownika
NO	PESEL		Integer	NO	Numer PESEL pracownika
NO	Data_Urodzenia		Date	YES	Data urodzenia pracownika
NO	Miasto		VarChar(20)	YES	Miasto zamieszkania pracownika
NO	Ulica		VarChar(20)	YES	Ulica zamieszkania pracownika
NO	Nr_budynku		VarChar(10)	YES	Numer budynku w którym mieszka pracownik
NO	Nr_lokalu		VarChar(10)	NO	Numer lokalu w którym mieszka pracownik
NO	Plec	Plec	Character(1)	YES	Plec pracownika
NO	Kod_pocztowy		VarChar(6)	YES	Kod pocztowy miejsca zamieszkania pracownika
NO	Stopien_naukowy	Stopien_naukowy	Character(1)	YES	Stopien naukowy pracownika

Encja Klient

Atrybuty tej encji stanowią dane klienta, takie jak adres, imię, nazwisko czy PESEL

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
YES	Id_klienta		Integer	YES	ID klienta - unikalny identyfikator
NO	Imie		VarChar(20)	YES	Imie zawodnika
NO	Nazwisko		VarChar(20)	YES	Nazwisko zawodnika
NO	Drugie_Imie		VarChar(20)	NO	Drugie imie pracownika
NO	PESEL		Integer	NO	Numer PESEL zawodnika
NO	Data_Urodzenia		Date	YES	Data urodzenia pracownika
NO	Miasto		VarChar(20)	YES	Miasto zamieszkania zawodnika
NO	Ulica		VarChar(20)	YES	Ulica zamieszkania zawodnika
NO	Nr_budynku		VarChar(10)	YES	Numer budynku w którym mieszka zawodnik
NO	Nr_lokalu		VarChar(10)	NO	Numer lokalu w którym mieszka zawodnik
NO	Plec	Plec	Character(1)	YES	Plec zawodnika
NO	Kod_pocztowy		VarChar(6)	YES	Kod pocztowy miejsca zamieszkania zawodnika

Encja Zawodnik

Encja zawiera dane zawodnika, które są rozszerzeniem dla danych klienta

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
NO	Data_przyjecia		Date	YES	Data przyjęcia do klubu
NO	Nr_karty		Integer	YES	Numer karty zawodniczej
NO	Reka_wiodaca	Reka_wiodaca	Character(1)	YES	Reka wiodaca zawodnika przy uderzeniach rakietą
NO	Wzrost		Integer	YES	Wzrost zawodnika
NO	Waga		Integer	YES	Waga zawodnika

Encja Trener

W ten encji znajdują się informację na temat trenera, będące uzupełnieniem danych pracownika

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
NO	Reka_wiodaca	Reka_wiodaca	Character(1)	YES	Reka wiodaca przy uderzeniach rakietą
NO	Wzrost		Integer	YES	Wzrost trenera
NO	Waga		Integer	YES	Waga trenera

Encja Grafik

Encja przechowuje podstawowe dane na temat grafiku trenera

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
YES	Id_grafiku		Integer	YES	ID grafiku - unikalny identyfikator
NO	Dzien_tygodnia		VarChar(30)	YES	Dzien tygodnia zajec trenera
NO	Godz_od		Character(4)	YES	Godzina rozpoczecia zajec
NO	Godz_do		Character(4)	YES	Godzina zakonczenia zajec

Encja Uprawnienia_trenerskie

Atrybuty tej encji przechowują podstawowe dane na temat uprawnień trenerskich trenera klubu

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
YES	Id_uprawnien		Integer	YES	ID uprawnień - unikalny identyfikator
NO	Nr_licencji		Integer	YES	Numer licencji
NO	Data_uzyskania		Date	YES	Data uzyskania licencji
NO	Placowka_wystawiajaca		VarChar(30)	YES	Placowka wystawiajaca licencje

Encja Wynagrodzenie

Encja zawiera informacje na temat wynagrodzenia pracownika, np. płaca brutto czy premia

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
YES	Id_wynagrodzenia		Integer	YES	ID Wynagrodzenia - unikalny identyfikator
NO	Placa_brutto		Money	YES	Kwota pensji brutto
NO	Premia		Money	NO	Premia dla pracownika

Encja Stanowisko

W tej encji przechowywane są informacje na temat stanowiska pracownika administracyjnego

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
YES	Id_stanowiska		Integer	YES	ID Stanowiska - unikalny identyfikator
NO	Nazwa		VarChar(20)	YES	Nazwa stanowiska
NO	Opis_stanowiska		VarChar(400)	YES	Opis stanowiska

Encja Administracja

Encja zawiera dane pracownika biurowego, czy też administracyjnego

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
NO	Stanowisko		VarChar(30)	YES	Stanowisko
NO	Nr_pokoju		Integer	NO	Numer pokoju w którym pracuje pracownik
NO	Nr_stanowiska		Integer	YES	Numer stanowiska przy którym pracuje pracownik

Encja Inne_kluby

Atrybutami tej encji są informacje na temat innych klubów, w których występował zawodnik

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
YES	Id_innego_klubu		Integer	YES	ID Innego klubu - unikatowy identyfikator
NO	Nazwa_klubu		VarChar(20)	YES	Nazwa klubu

NO	Uczestnictwo_od		Date	YES	Początek uczestnictwa w innym klubie
NO	Uczestnictwo_do		Date	YES	Koniec uczestnictwa w innym klubie

Encja Usługa

Encja ta przechowuje informacje na temat usługi, np. cena czy rodzaj usługi

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
YES	Id_uslugi		Integer	YES	ID Usługi - unikalny identyfikator
NO	Rodzaj_uslugi		VarChar(30)	YES	Rodzaj usługi
NO	Cena		Money	YES	Usługa ma cenę

Encja Lekcja

Atrybutami tej encji są informacje na temat lekcji gry w tenisa, z której może skorzystać klient

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
NO	Od_godz		Character(4)	YES	Początek lekcji
NO	Do_godz		Character(4)	YES	Godzina zakończenia lekcji
NO	Data		Date	YES	Data lekcji

Encja Wynajem

W tej encji znajdują się informacje na temat wynajmu kortu przez klienta

Identyfikujący	Nazwa atrybutu	Dziedzina	Typ danych	Obowiązkowy	Opis
NO	Od_godz		Character(4)	YES	Godzina rozpoczęcia rezerwacji
NO	Do_godz		Character(4)	YES	Godzina do której została wykonana rezerwacja
NO	Data		Date	YES	Data rezerwacji
NO	Liczba_osob		Integer	YES	Ile osób przybędzie w ramach rezerwacji

3.2 Ustalanie związków między encjami i ich typów

1. Relacja Klub – Placowka

Encja klub może posiadać wiele placówek, oferujących usługi klientom i zatrudniających pracowników. W chwili założenia, klub musi posiadać placówkę

2. Relacja Klub – Klient

Klub może posiadać wielu klientów, ale na początku swojego funkcjonowania może nie mieć żadnego

3. Relacja Placowka – Usługa

Placówka oferuje wiele usług i nie może nie oferować żadnej

4. Relacja Klient – Usługa

Klient może korzystać z wielu usług, tak jak jedna usługa może mieć wielu klientów. Dana usługa może nie mieć klienta, a żaden klient może nie korzystać z jakiejś usługi

5. Relacja Placowka – Pracownik

Placówka może zatrudniać wielu pracowników, zaraz po otwarciu może nie mieć żadnego. Natomiast pracownik musi być zatrudniony w jeden z placówek

6. Relacja Administracja – Stanowisko

Pracownik administracyjny musi być zatrudniony na jednym stanowisku, ale dane stanowisko może posiadać wakat lub mieć paru pracowników

7. Relacja Pracownik – Wynagrodzenie

Każdy pracownik otrzymuje wynagrodzenie

8. Relacja Trener - Grafiki

Trener posiada grafiki pracy

9. Relacja Trener – Uprawnienia_trenerskie

Trener posiada uprawnienia trenerskie. Może mieć tylko jedno

10. Relacja Zawodnik – Inne_kluby

Zawodnik mógł występować w kilku innych klubach przed przystąpieniem do naszego klubu i dany był klub mógł mieć kilku naszych obecnych zawodników

11. Relacja Placowka – Stanowisko

Placówka może utrzymywać wiele stanowisk, na samym początku może to być zero, ale stanowisko może być przypisane tylko do jednej placówki

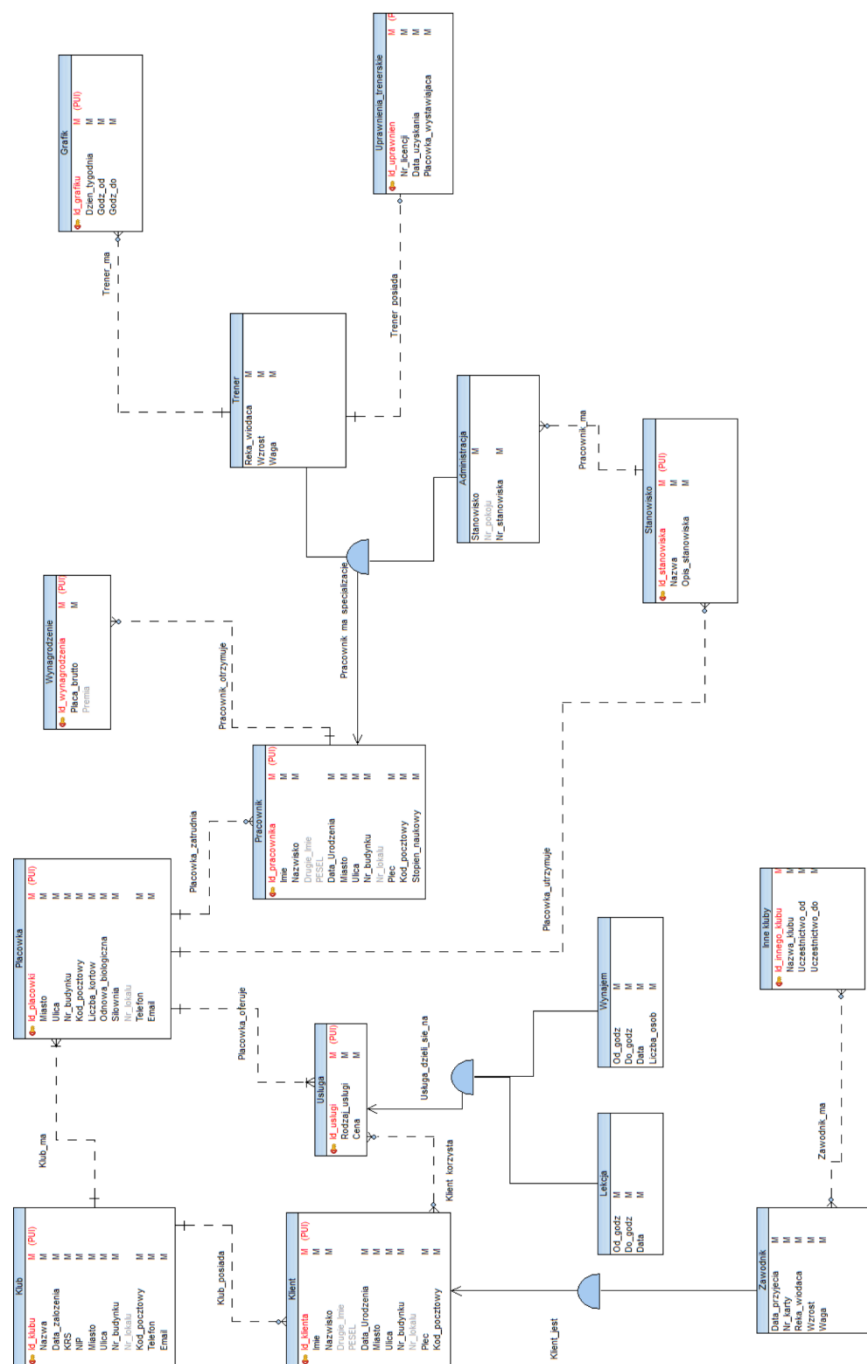
Tabela określająca typy relacji i ich krotności:

Relacja	Typ związku	Typ uczestnictwa	Stopień
Klient - Usługa	n : m	Opcjonalna – opcjonalna	binarny
Klub - Placowka	1 : n	Obowiązkowa – obowiązkowa	binarny
Klub - Klient	1 : n	Obowiązkowa – opcjonalna	binarny
Placowka - Usługa	1 : n	Obowiązkowa – obowiązkowa	binarny
Placowka - Pracownik	1 : n	Obowiązkowa – opcjonalna	binarny
Administracja - Stanowisko	n : 1	Opcjonalna – obowiązkowa	binarny
Pracownik - Wynagrodzenie	1 : n	Obowiązkowa – opcjonalna	binarny
Trener - Grafiki	1 : n	Obowiązkowa – opcjonalna	binarny
Trener - Uprawnienia_trenerskie	1 : 1	Obowiązkowa – opcjonalna	binarny
Zawodnik - Inne_kluby	n : n	Opcjonalna – opcjonalna	binarny
Placowka - Stanowisko	1 : n	Obowiązkowa – opcjonalna	binarny

3.3 Klucze kandydujące i główne

Nazwa Encji	Klucz główny	Klucz kandydujący
Grafik	Id_grafiku	-
Inne_kluby	Id_innego_klubu	Nazwa_klubu
Klient	Id_klienta	-
Klub	Id_klubu	KRS
Placowka	Id_placowki	Email
Pracownik	Id_pracownika	-
Stanowisko	Id_stanowiska	-
Uprawnienia_trenerskie	Id_uprawnien	Nr_licencji
Usługa	Id_uslugi	-
Wynagrodzenie	Id_wynagrodzenia	-

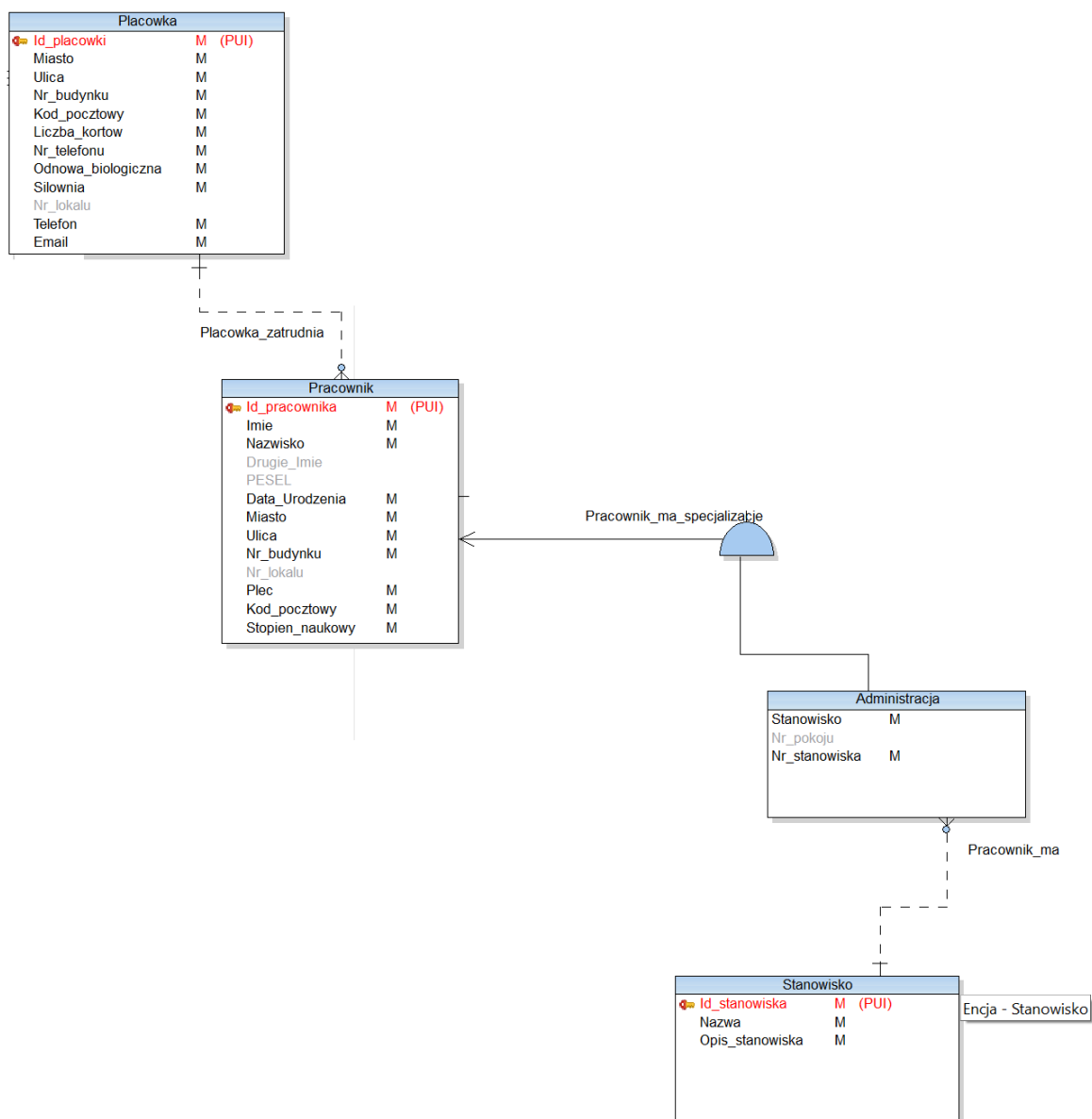
3.4 Schemat ER na poziomie konceptualnym



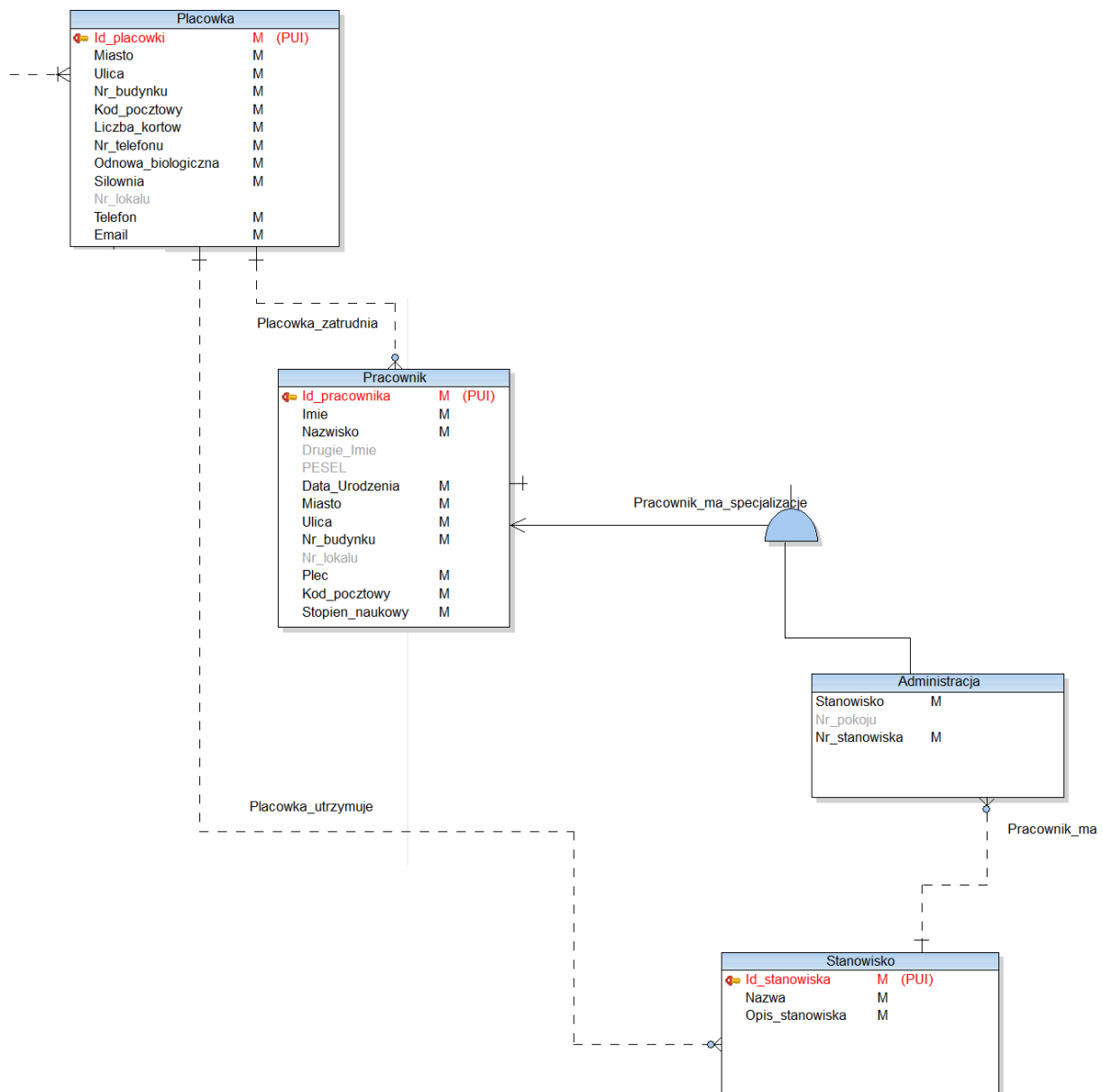
3.5 Problem pętalek szczelinowych i wachlarzowych

Szczelinowa

Powstało ryzyko, iż jeżeli stanowisko będzie uzależnione od pracownika, to usuwając pracownika stracilibyśmy również zajmowane przez niego stanowisko.



Rozwiązaniem było powiązanie stanowiska z placówką



4. Model logiczny

4.1 Charakterystyka modelu relacyjnego

Żeby dokonać konwersji modelu konceptualnego na logiczny model relacyjny, należało zastąpić związki wielu do wielu z wykorzystaniem tablic łączących. W moim projekcie wystąpiły dwie takie sytuacje, gdzie należało połączyć dwie encję będące w związku wielu do wielu, związkami jeden do wielu z tablicą łączącą.

4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym

Usunięcie związków wielu do wielu

Następujące związki wielu do wielu zostały usunięte

1. Zawodnicy – Inne_kluby

Zawodnicy mogli w przeszłości występować w wielu innych klubach, tak samo jak w danym klubie mogło trenować wielu z naszych obecnych zawodników. W celu usunięcia niekompatybilności stworzono tablicę łączącą Historia_kariery, która gromadzi powiązania zawodników z innymi klubami.

2. Klient – Usługa

Każdy klient może korzystać z wielu usług, a każda usługa może być świadczona dla więcej niż jednego klienta. Niekompatybilność zastąpiono tablicą łączącą Klient_Usluga

4.3 Proces normalizacji

Pierwsza postać normalna

Aby otrzymać pierwszą postać normalną, należy posiadać w swojej bazie same pola atomowe. W moim przypadku, już na samym początku, w fazie projektowania konceptualnego, starałem się unikać pól segmentowych, dlatego takie dane jak *adres* rozbiłem na pola *miasto*, *ulica*, *numer domu*, itd.

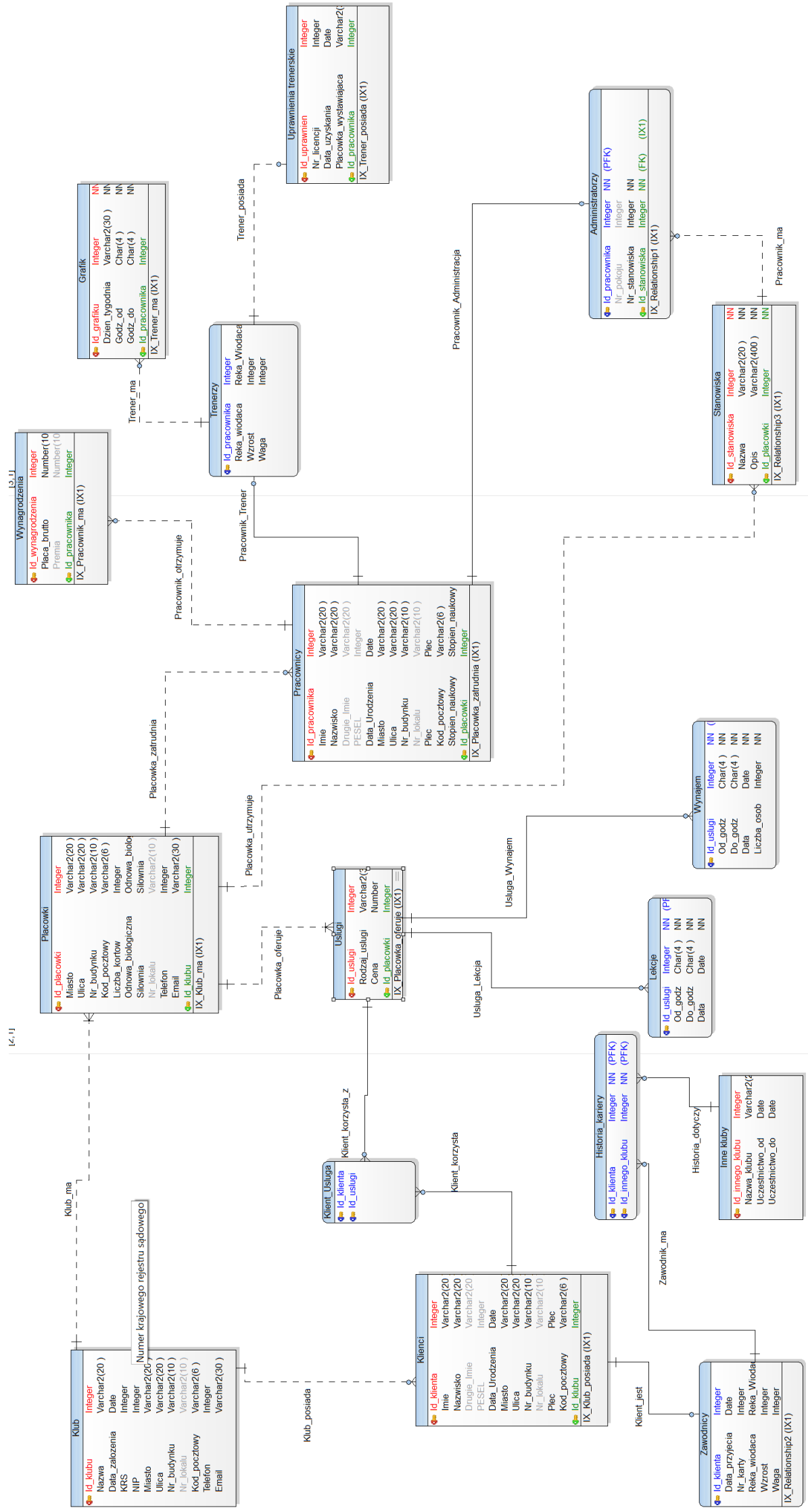
Druga postać normalna

Żeby spełnić tę postać, wszystkie klucze kandydujące musiały być kluczami prostymi. Przy przejściu z modelu konceptualnego na model relacyjny, również ta postać była spełniona.

Trzecia postać normalna

W celu uzyskania postaci normalnej, z Encji *Administratorzy* została wydzielona encja *Stanowiska* z atrybutami *Id_Stnowiska*, *Opis*, *Nazwa*.

4.4 Schemat ER na poziomie modelu logicznego



4.5 Więzy integralności

Wszystkie pola są polami atomowymi. Każda encja posiada klucz główny, który nie przyjmuje wartości pustych (NOT NULL) i jest unikalny (UNIQUE)

4.6 Proces denormalizacji

Moim zdaniem nie ma potrzeby denormalizacji bazy danych, która w znaczący sposób przyspieszyłaby dostęp do niej.

5. Faza fizyczna

5.1 Projekt transakcji i weryfikacja ich wykonalności

Transakcja	Wymagane zasoby (tabele)
Podgląd danych personalnych pracowników	Pracownicy
Dodawanie, modyfikacja, usuwanie danych pracownika	Pracownicy
Podgląd danych personalnych trenerów	Pracownicy, Trenerzy
Dodawanie, modyfikacja, usuwanie danych trenera	Pracownicy, Trenerzy, Uprawnienia_trenerskie, Grafiki
Dodawanie, modyfikacja, usuwanie danych klientów	Klienci
Podgląd danych personalnych zawodników	Klienci, Zawodnicy
Dodawanie, modyfikacja, usuwanie danych wynajmu	Wynajem
Podgląd danych placówki	Placowka
Dodawanie, modyfikacja, usuwanie uprawnień trenerskich	Uprawnienia_trenerskie
Podgląd danych o stanowiskach	Stanowiska
Dodawanie, modyfikacja, usuwanie danych wynagrodzeń	Wynagrodzenia

Wymienione wyżej transakcje przeszły pozytywnie weryfikację ich wykonalności

5.2 Strojanie bazy danych

Pisząc zapytania do mojej bazy danych, zaobserwowałem kilka, które powtarzały się często i warto założyć dla nich indeksy

```
CREATE INDEX Dzień_tygodnia_grafiku ON GRAFIK(Dzień_tygodnia);
```

```
CREATE INDEX Miasto_placowki ON PLACOWKA(Miasto);
```

```
CREATE INDEX Nazwa_stanowiska ON STANOWISKA(Nazwa);
```

```
CREATE INDEX Numer_stanowiska ON ADMINISTRATORZY(Nr_stanowiska);
```

```
CREATE INDEX Cena_uslugi ON USLUGI(Cena);
```

5.3 Skrypt SQL zakładający bazę danych

```
-- CREATE TABLES SECTION -----  
  
-- TABLE KLUB  
  
CREATE TABLE KLUB(  
  ID_KLUBU INTEGER NOT NULL,  
  NAZWA VARCHAR2(20 ) NOT NULL,  
  DATA_ZALOZENIA DATE NOT NULL,  
  KRS INTEGER NOT NULL,  
  NIP INTEGER NOT NULL,  
  MIASTO VARCHAR2(20 ) NOT NULL,  
  ULICA VARCHAR2(20 ) NOT NULL,  
  NR_BUDYNKU VARCHAR2(10 ) NOT NULL,  
  NR_LOKALU VARCHAR2(10 ),  
  KOD_POCZTOWY VARCHAR2(6 ) NOT NULL,  
  TELEFON INTEGER NOT NULL,  
  EMAIL VARCHAR2(30 ) NOT NULL  
)  
/  
  
-- ADD KEYS FOR TABLE KLUB  
  
ALTER TABLE KLUB ADD CONSTRAINT KLUBPK PRIMARY KEY (ID_KLUBU)  
/  
  
-- TABLE PRACOWNICY  
  
CREATE TABLE PRACOWNICY(  
  ID_PRACOWNIKA INTEGER NOT NULL,  
  IMIE VARCHAR2(20 ) NOT NULL,  
  NAZWISKO VARCHAR2(20 ) NOT NULL,  
  DRUGIE_IMIE VARCHAR2(20 ),  
  PESEL INTEGER,  
  DATA_URODZENIA DATE NOT NULL,  
  MIASTO VARCHAR2(20 ) NOT NULL,
```

```

        ULICA VARCHAR2(20 ) NOT NULL,
        NR_BUDYNKU VARCHAR2(10 ) NOT NULL,
        NR_LOKALU VARCHAR2(10 ),
        PLEC CHAR(1 ) NOT NULL
            CONSTRAINT CHECKCONSTRAINTD1 CHECK (PLEC IN ('K', 'M')),
        KOD_POCZTOWY VARCHAR2(6 ) NOT NULL,
        STOPIEN_NAUKOWY CHAR(1 ) NOT NULL
            CONSTRAINT CHECKCONSTRAINTD2 CHECK (STOPIEN_NAUKOWY IN ('L', 'I', 'M', 'D', 'P')),
        ID_PLACOWKI INTEGER NOT NULL
    )
/

-- CREATE INDEXES FOR TABLE PRACOWNICY

CREATE INDEX IX_PLACOWKA_ZATRUDNIA ON PRACOWNICY (ID_PLACOWKI)
/

-- ADD KEYS FOR TABLE PRACOWNICY

ALTER TABLE PRACOWNICY ADD CONSTRAINT PRACOWNIKPK PRIMARY KEY (ID_PRACOWNIKA)
/

-- TABLE PLACOWKI

CREATE TABLE PLACOWKI(
    ID_PLACOWKI INTEGER NOT NULL,
    MIASTO VARCHAR2(20 ) NOT NULL,
    ULICA VARCHAR2(20 ) NOT NULL,
    NR_BUDYNKU VARCHAR2(10 ) NOT NULL,
    KOD_POCZTOWY VARCHAR2(6 ) NOT NULL,
    LICZBA_KORTOW INTEGER NOT NULL,
    ODNOWA_BIOLOGICZNA CHAR(1 ) NOT NULL
        CONSTRAINT CHECKCONSTRAINTD3 CHECK (ODNOWA_BIOLOGICZNA IN ('Y', 'N')),
    SILOWNIA CHAR(1 ) NOT NULL
        CONSTRAINT CHECKCONSTRAINTD4 CHECK (SILOWNIA IN ('Y', 'N')),
    NR_LOKALU VARCHAR2(10 ),
    TELEFON INTEGER NOT NULL,
    EMAIL VARCHAR2(30 ) NOT NULL,
    ID_KLUBU INTEGER NOT NULL
)
/

-- CREATE INDEXES FOR TABLE PLACOWKI

CREATE INDEX IX_KLUB_MA ON PLACOWKI (ID_KLUBU)
/

-- ADD KEYS FOR TABLE PLACOWKI

ALTER TABLE PLACOWKI ADD CONSTRAINT PLACOWKAPK PRIMARY KEY (ID_PLACOWKI)
/

-- TABLE TRENERZY

CREATE TABLE TRENERZY(
    ID_PRACOWNIKA INTEGER NOT NULL,
    REKA_WIODACA CHAR(1 ) NOT NULL,
    WZROST INTEGER NOT NULL,
    WAGA INTEGER NOT NULL
)
/

-- ADD KEYS FOR TABLE TRENERZY

ALTER TABLE TRENERZY ADD CONSTRAINT TRENERPK PRIMARY KEY (ID_PRACOWNIKA)
/

```

-- TABLE UPRAWNIENIA_TRENEERSKIE

```
CREATE TABLE UPRAWNIENIA_TRENEERSKIE(  
  ID_UPRAWNIEN INTEGER NOT NULL,  
  NR_LICENCJI INTEGER NOT NULL,  
  DATA_UZYSKANIA DATE NOT NULL,  
  PLACOWKA_WYSTAWIAJACA VARCHAR2(30 ) NOT NULL,  
  ID_PRACOWNIKA INTEGER  
)  
/
```

-- CREATE INDEXES FOR TABLE UPRAWNIENIA_TRENEERSKIE

```
CREATE INDEX IX_TRENER_POSIADA ON UPRAWNIENIA_TRENEERSKIE (ID_PRACOWNIKA)  
/
```

-- ADD KEYS FOR TABLE UPRAWNIENIA_TRENEERSKIE

```
ALTER TABLE UPRAWNIENIA_TRENEERSKIE ADD CONSTRAINT UPRAWNIENIAPK PRIMARY KEY (ID_UPRAWNIEN)  
/
```

-- TABLE GRAFIK

```
CREATE TABLE GRAFIK(  
  ID_GRAFIKU INTEGER NOT NULL,  
  DZIEEN_TYGODNIA VARCHAR2(30 ) NOT NULL,  
  GODZ_OD CHAR(4 ) NOT NULL,  
  GODZ_DO CHAR(4 ) NOT NULL,  
  ID_PRACOWNIKA INTEGER  
)  
/
```

-- CREATE INDEXES FOR TABLE GRAFIK

```
CREATE INDEX IX_TRENER_MA ON GRAFIK (ID_PRACOWNIKA)  
/
```

-- ADD KEYS FOR TABLE GRAFIK

```
ALTER TABLE GRAFIK ADD CONSTRAINT GRAFIKPK PRIMARY KEY (ID_GRAFIKU)  
/
```

-- TABLE KLIENCI

```
CREATE TABLE KLIENCI(  
  ID_KLIENTA INTEGER NOT NULL,  
  IMIE VARCHAR2(20 ) NOT NULL,  
  NAZWISKO VARCHAR2(20 ) NOT NULL,  
  DRUGIE_IMIE VARCHAR2(20 ),  
  PESEL INTEGER,  
  DATA_URODZENIA DATE NOT NULL,  
  MIASTO VARCHAR2(20 ) NOT NULL,  
  ULICA VARCHAR2(20 ) NOT NULL,  
  NR_BUDYNKU VARCHAR2(10 ) NOT NULL,  
  NR_LOKALU VARCHAR2(10 ),  
  PLEC CHAR(1 ) NOT NULL  
  CONSTRAINT CHECKCONSTRAINTD5 CHECK (PLEC IN ('K', 'M'))),  
  KOD_POCZTOWY VARCHAR2(6 ) NOT NULL,  
  ID_KLUBU INTEGER NOT NULL  
)  
/
```

-- CREATE INDEXES FOR TABLE KLIENCI

```
CREATE INDEX IX_KLUB_POSIADA ON KLIENCI (ID_KLUBU)
```

```

/

-- ADD KEYS FOR TABLE KLIENCI

ALTER TABLE KLIENCI ADD CONSTRAINT KLIENTPK PRIMARY KEY (ID_KLIENTA)
/

-- TABLE USLUGI

CREATE TABLE USLUGI(
  ID_USLUGI INTEGER NOT NULL,
  RODZAJ_USLUGI VARCHAR2(30 ) NOT NULL,
  CENA NUMBER NOT NULL
  ID_PLACOWKI INTEGER NOT NULL,
)
/

-- CREATE INDEXES FOR TABLE USLUGI

CREATE INDEX IX_PLACOWKA_OFERUJE ON USLUGI (ID_PLACOWKI)
/

-- ADD KEYS FOR TABLE USLUGI

ALTER TABLE USLUGI ADD CONSTRAINT USLUGAPK PRIMARY KEY (ID_USLUGI)
/

-- TABLE AND COLUMNS COMMENTS SECTION

COMMENT ON COLUMN USLUGI.CENA IS 'CENA USŁUGI'
/

-- TABLE ADMINISTRATORZY

CREATE TABLE ADMINISTRATORZY(
  ID_PRACOWNIKA INTEGER NOT NULL,
  NR_POKOJU INTEGER,
  NR_STANOWISKA INTEGER NOT NULL,
  ID_STANOWISKA INTEGER NOT NULL
)
/

-- CREATE INDEXES FOR TABLE ADMINISTRATORZY

CREATE INDEX IX_RELATIONSHIP1 ON ADMINISTRATORZY (ID_STANOWISKA)
/

-- ADD KEYS FOR TABLE ADMINISTRATORZY

ALTER TABLE ADMINISTRATORZY ADD CONSTRAINT UNIQUE_IDENTIFIER1 PRIMARY KEY (ID_PRACOWNIKA)
/

-- TABLE WYNAGRODZENIA

CREATE TABLE WYNAGRODZENIA(
  ID_WYNAGRODZENIA INTEGER NOT NULL,
  PLACA_BRUTTO NUMBER(10,2) NOT NULL,
  PREMIA NUMBER(10,2),
  ID_PRACOWNIKA INTEGER NOT NULL
)
/

-- CREATE INDEXES FOR TABLE WYNAGRODZENIA

CREATE INDEX IX_PRACOWNIK_MA ON WYNAGRODZENIA (ID_PRACOWNIKA)
/

```

-- ADD KEYS FOR TABLE WYNAGRODZENIA

ALTER TABLE WYNAGRODZENIA ADD CONSTRAINT WYNAGRODZENIEPK PRIMARY KEY (ID_WYNAGRODZENIA)
/

-- TABLE LEKCJE

CREATE TABLE LEKCJE(
ID_USLUGI INTEGER NOT NULL,
OD_GODZ CHAR(4) NOT NULL,
DO_GODZ CHAR(4) NOT NULL,
DATA DATE NOT NULL
)
/

-- ADD KEYS FOR TABLE LEKCJE

ALTER TABLE LEKCJE ADD CONSTRAINT UNIQUE_IDENTIFIER2 PRIMARY KEY (ID_USLUGI)
/

-- TABLE WYNAJEM

CREATE TABLE WYNAJEM(
ID_USLUGI INTEGER NOT NULL,
OD_GODZ CHAR(4) NOT NULL,
DO_GODZ CHAR(4) NOT NULL,
DATA DATE NOT NULL,
LICZBA_OSOB INTEGER NOT NULL
)
/

-- ADD KEYS FOR TABLE WYNAJEM

ALTER TABLE WYNAJEM ADD CONSTRAINT UNIQUE_IDENTIFIER3 PRIMARY KEY (ID_USLUGI)
/

-- TABLE KLIENT_USLUGA

CREATE TABLE KLIENT_USLUGA(
ID_KLIENTA INTEGER NOT NULL,
ID_USLUGI INTEGER NOT NULL
)
/

-- TABLE STANOWISKA

CREATE TABLE STANOWISKA(
ID_STANOWISKA INTEGER NOT NULL,
NAZWA VARCHAR2(20) NOT NULL,
OPIS VARCHAR2(400) NOT NULL,
ID_PLACOWKI INTEGER NOT NULL
)
/

-- CREATE INDEXES FOR TABLE STANOWISKA

CREATE INDEX IX_RELATIONSHIP3 ON STANOWISKA (ID_PLACOWKI)
/

-- ADD KEYS FOR TABLE STANOWISKA

ALTER TABLE STANOWISKA ADD CONSTRAINT PK_STANOWISKA PRIMARY KEY (ID_STANOWISKA)
/

-- TABLE AND COLUMNS COMMENTS SECTION

```

COMMENT ON COLUMN STANOWISKA.ID_STANOWISKA IS 'ID STANOWISKA - UNIKALNY IDENTYFIKATOR'
/
COMMENT ON COLUMN STANOWISKA.NAZWA IS 'NAZWA STANOWISKA'
/
COMMENT ON COLUMN STANOWISKA.OPIS IS 'OPIS STANOWISKA'
/

-- TABLE ZAWODNICY

CREATE TABLE ZAWODNICY(
  ID_KLIENTA INTEGER NOT NULL,
  DATA_PRZYJECIA DATE NOT NULL,
  NR_KARTY INTEGER NOT NULL,
  REKA_WIODACA CHAR(1 ) NOT NULL,
  WZROST INTEGER NOT NULL,
  WAGA INTEGER NOT NULL
)
/

-- CREATE INDEXES FOR TABLE ZAWODNICY

CREATE INDEX IX_RELATIONSHIP2 ON ZAWODNICY (ID_KLIENTA)
/

-- ADD KEYS FOR TABLE ZAWODNICY

ALTER TABLE ZAWODNICY ADD CONSTRAINT PK_ZAWODNICY PRIMARY KEY (ID_KLIENTA)
/

-- TABLE AND COLUMNS COMMENTS SECTION

COMMENT ON TABLE ZAWODNICY IS 'ZAWODNIK KLUBU'
/
COMMENT ON COLUMN ZAWODNICY.DATA_PRZYJECIA IS 'DATA PRZYJECIA DO KLUBU'
/
COMMENT ON COLUMN ZAWODNICY.NR_KARTY IS 'NUMER KARTY ZAWODNICZEJ'
/
COMMENT ON COLUMN ZAWODNICY.REKA_WIODACA IS 'REKA WIADACA ZAWODNIKA'
/
COMMENT ON COLUMN ZAWODNICY.WZROST IS 'WZROST ZAWODNIKA'
/
COMMENT ON COLUMN ZAWODNICY.WAGA IS 'WAGA ZAWODNIKA'
/

-- TABLE INNE_KLUBY

CREATE TABLE INNE_KLUBY(
  ID_INNEGO_KLUBU INTEGER NOT NULL,
  NAZWA_KLUBU VARCHAR2(20 ) NOT NULL,
  UCZESTNICTWO_OD DATE NOT NULL,
  UCZESTNICTWO_DO DATE NOT NULL
)
/

-- ADD KEYS FOR TABLE INNE_KLUBY

ALTER TABLE INNE_KLUBY ADD CONSTRAINT PK_INNE_KLUBY PRIMARY KEY (ID_INNEGO_KLUBU)
/

-- TABLE AND COLUMNS COMMENTS SECTION

COMMENT ON COLUMN INNE_KLUBY.ID_INNEGO_KLUBU IS 'ID INNEGO KLUBU - UNIKALNY IDENTYFIKATOR'
/
COMMENT ON COLUMN INNE_KLUBY.NAZWA_KLUBU IS 'NAZWA KLUBU'
/

```

```

COMMENT ON COLUMN INNE_KLUBY.UCZESTNICTWO_OD IS 'POCZATEK UCZESTNICTWA W DANYM KLUBIE'
/
COMMENT ON COLUMN INNE_KLUBY.UCZESTNICTWO_DO IS 'KONIEC UCZESTNICTWA W DANYM KLUBIE'
/

-- TABLE HISTORIA_KARIERY

CREATE TABLE HISTORIA_KARIERY(
  ID_KLIENTA INTEGER NOT NULL,
  ID_INNEGO_KLUBU INTEGER NOT NULL
)
/

-- ADD KEYS FOR TABLE HISTORIA_KARIERY

ALTER TABLE HISTORIA_KARIERY ADD CONSTRAINT PK_HISTORIA_KARIERY PRIMARY KEY (ID_KLIENTA, ID_INNEGO_KLUBU)
/

-- CREATE FOREIGN KEYS (RELATIONSHIPS) SECTION -----

ALTER TABLE PLACOWKI ADD CONSTRAINT KLUB_MA FOREIGN KEY (ID_KLUBU) REFERENCES KLUB (ID_KLUBU)
/

ALTER TABLE UPRAWNIENIA_TREnerskie ADD CONSTRAINT TRENER_POSIADA FOREIGN KEY (ID_PRACOWNIKA) REFERENCES TRENERZY (ID_PRACOWNIKA)
/

ALTER TABLE GRAFIK ADD CONSTRAINT TRENER_MA FOREIGN KEY (ID_PRACOWNIKA) REFERENCES TRENERZY (ID_PRACOWNIKA)
/

ALTER TABLE KLIENCI ADD CONSTRAINT KLUB_POSIADA FOREIGN KEY (ID_KLUBU) REFERENCES KLUB (ID_KLUBU)
/

ALTER TABLE USLUGI ADD CONSTRAINT PLACOWKA_OFERUJE FOREIGN KEY (ID_PLACOWKI) REFERENCES PLACOWKI (ID_PLACOWKI)
/

ALTER TABLE PRACOWNICY ADD CONSTRAINT PLACOWKA_ZATRUDNIA FOREIGN KEY (ID_PLACOWKI) REFERENCES PLACOWKI (ID_PLACOWKI)
/

ALTER TABLE WYNAGRODZENIA ADD CONSTRAINT PRACOWNIK_OTRZYMUJE FOREIGN KEY (ID_PRACOWNIKA) REFERENCES PRACOWNICY (ID_PRACOWNIKA)
/

ALTER TABLE ADMINISTRATORZY ADD CONSTRAINT PRACOWNIK_MA FOREIGN KEY (ID_STANOWISKA) REFERENCES STANOWISKA (ID_STANOWISKA)
/

ALTER TABLE ZAWODNICY ADD CONSTRAINT KLIENT_JEST FOREIGN KEY (ID_KLIENTA) REFERENCES KLIENCI (ID_KLIENTA)
/

ALTER TABLE HISTORIA_KARIERY ADD CONSTRAINT ZAWODNIK_MA FOREIGN KEY (ID_KLIENTA) REFERENCES ZAWODNICY (ID_KLIENTA)
/

ALTER TABLE HISTORIA_KARIERY ADD CONSTRAINT HISTORIA_DOTYCZY FOREIGN KEY (ID_INNEGO_KLUBU) REFERENCES INNE_KLUBY (ID_INNEGO_KLUBU)
/

ALTER TABLE STANOWISKA ADD CONSTRAINT PLACOWKA_UTRZYMUJE FOREIGN KEY (ID_PLACOWKI) REFERENCES PLACOWKI (ID_PLACOWKI)
/

```

5.4 Przykłady zapytań SQL do bazy danych

Przykładowy insert do zasilenia tabeli Usługi

```
INSERT INTO USLUGI VALUES (1, 'KONSULTACJE', 200, 2)
```

Przykład wyświetlania danych trenerów mieszkających w Warszawie

```
SELECT * FROM PRACOWNICY INNER JOIN TRENERZY ON PRACOWNICY.Id_Pracownika =  
TRENERZY.Id_pracownika  
WHERE PRACOWNICY.MIASTO = 'WARSZAWA';
```

Wyświetlanie danych placówki, w której znajduje się siłownia

```
SELECT * FROM PLACOWKI WHERE SILOWNIA = 'Y'
```

Wyświetlanie danych pracowników, którzy zarabiają więcej niż 1500 zł brutto

```
SELECT * FROM PRACOWNICY INNER JOIN WYNAGRODZENIA ON PRACOWNICY.Id_pracownika =  
WYNAGRODZENIA.Id_pracownika  
WHERE Placa_brutto > 1500
```