



WYDZIAŁ
ELEKTROTECHNIKI
I INFORMATYKI
POLITECHNIKI RZESZOWSKIEJ

Bazy danych – projekt
System wspomagający działanie klubu fitness

Małgorzata Radomska 173203

Adrianna Rapa 173204

Inżynieria i Analiza Danych

Rzeszów 2024

Spis treści

Projekt konceptualny	3
Zadanie projektowe.....	3
Analiza stanu wyjściowego.....	3
Analiza wymagań użytkownika	3
Określenie scenariuszy użycia, diagram use case	3
Identyfikacja funkcji	5
Propozycja encji, atrybutów oraz powiązań	5
Projekt logiczny	6
Przejście z modelu ERD na model relacyjny	6
Diagram relacyjnej bazy danych.....	6
Analiza zależności funkcyjnych i normalizacja tabel	7
Tabela Adres	7
Tabela Instruktor.....	8
Tabela Specjalizacja.....	8
Tabela Certyfikat	9
Tabela Klient.....	9
Tabela Karnet.....	10
Tabela Zamówienie.....	10
Tabela Grupa zajęciowa.....	11
Tabela Rezerwacja	11
Tabela Sala.....	12
Tabela Zajęcia	12
Diagram relacyjnej bazy danych po normalizacji.....	13
Propozycje operacji na danych	13
Projekt implementacyjny	15
Tworzenie tabel.....	15
Wprowadzenie danych do tabel	20
Utworzone tabele po wprowadzeniu danych	30
Tworzenie kwerend.....	33
Określenie kierunków rozwoju aplikacji	38
Podsumowanie i wnioski	39
Bibliografia	40

Projekt konceptualny

Zadanie projektowe

Celem projektu jest zaplanowanie i stworzenie bazy danych na poziomie konceptualnym (analiza stanu wyjściowego, analiza wymagań użytkownika, określenie scenariuszy użycia, diagram UML, identyfikacja funkcji, diagram ERD), logicznym (model relacyjny, normalizacja, propozycje operacji na danych) i implementacyjnym (skrypty w języku SQL, tworzenie tabel, wprowadzanie, danych, tworzenie kwerend).

Analiza stanu wyjściowego

Tematem projektu jest *System wspomagający działanie klubu fitness*. Klub fitness jest miejscem spotkań klientów i instruktorów na zajęciach sportowych. Tworząc system wspomagający działanie takiego klubu, należy wziąć pod uwagę informacje dotyczące oferowanych usług, osób korzystających z nich oraz usługodawców. Ponadto należy się zastanowić nad technikami umożliwiającymi uczęszczanie na oferowane zajęcia, np. sprzedaż karnetów lub dokonywanie rezerwacji miejsc w grupach zajęciowych. System powinien być dostępny dla klientów, instruktorów oraz administratorów, zarządzającymi informacjami.

Analiza wymagań użytkownika

System powinien być dostępny dla klientów, instruktorów oraz administratorów. Klient powinien mieć dostęp do proponowanej oferty klubu (np. katalogu przeprowadzanych zajęć), możliwość rejestracji w systemie, a później logowania się do niego, możliwość złożenia zamówienia na karnet miesięczny oraz możliwość rezerwacji miejsca w grupie zajęciowej. Instruktor powinien posiadać możliwość logowania do systemu oraz dostęp do informacji na temat prowadzonych zajęć. Rolą administratora jest zarządzanie przechowywanymi w bazie danymi, tj. wprowadzanie informacji na temat instruktorów i zajęć oraz dostęp do informacji na temat klientów, a także zarządzanie płatnościami.

Określenie scenariuszy użycia, diagram use case

Zgodnie z poniższym diagramem use case dla systemu wspomagającego działanie klubu fitness aktorami są klienci (zarejestrowani i niezarejestrowani), instruktor i administrator. Każdy z aktorów ma przypisane role.

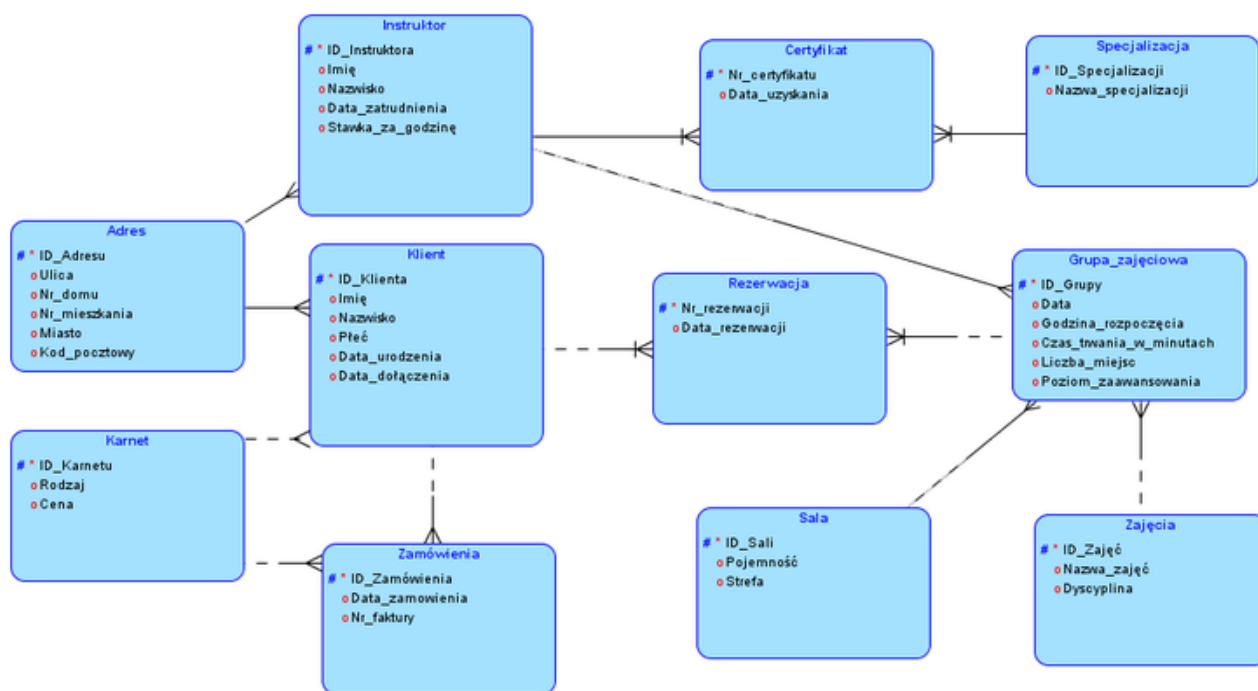
- Klient niezarejestrowany - możliwość przeglądania strony, dostęp do informacji o aktualnościach, ofercie, innych klubach i do danych kontaktowych oraz możliwość rejestracji w systemie.
- Klient zarejestrowany - możliwość zalogowania na swoje konto w systemie, odzyskania hasła, zakupu karnetu, zapisu na zajęcia oraz zmiany danych osobowych. Ponadto klientowi zarejestrowanemu przysługuje taki sam dostęp do informacji zawartych na stronie jak klientowi niezarejestrowanemu.
- Instruktor - dostęp do informacji na temat prowadzonych zajęć, historii zajęć oraz listy uczestników.
- Administrator – posiada największe uprawnienia. Zarządza klientami, instruktorami i ich pracą, informacjami dotyczącymi zajęć oraz płatnościami.

Identyfikacja funkcji

- Przeglądanie strony - każdy użytkownik ma pełen dostęp do informacji zawartych na stronie internetowej, takich jak aktualności, oferta klubu (informacje o zajęciach, grafiku, prowadzących), najczęściej zadawane pytania, informacje o powiązanych klubach oraz możliwość kontaktu przez infolinię lub dostępny formularz.
- Rejestracja - możliwość założenia prywatnego konta w systemie, co w konsekwencji daje możliwość korzystania z oferty klubu już jako użytkownik zarejestrowany.
- Logowanie - dostęp do prywatnego konta mają klienci zarejestrowani oraz instruktorzy. Klienci otrzymują możliwość zmiany danych osobowych, zakupu miesięcznego karnetu dowolnego rodzaju potwierdzonego fakturą, zapisu na zajęcia lub rezygnacji z nich. Każda z operacji kończy się potwierdzeniem.
- Nie pamiętam hasła - opcja dla klientów zarejestrowanych, dająca możliwość odzyskania hasła do prywatnego konta.
- Zarządzanie zajęciami - funkcja dostępna dla administratora, dająca możliwość tworzenia harmonogramu zajęć, odwoływania ich i zmiany informacji dotyczących sali, daty lub godziny planowanych zajęć wraz z potwierdzeniem operacji.
- Zarządzaj klientami – daje administratorowi dostęp do listy klientów i informacji o nich.
- Zarządzaj płatnościami - pozwala na zarządzanie płatnościami przez administratora i sprawdzanie ich statusu. System generuje faktury.
- Zarządzaj instruktorami – funkcja dostępna jedynie dla administratora, która pozwala na zarządzanie grafikami instruktorów, zmianę danych osobowych oraz dodawanie i usuwanie instruktorów.

Propozycja encji, atrybutów oraz powiązań

Zgodnie z poniżej przedstawionym diagramem ERD utworzone zostały następujące encje: adres, klient, karnet, zamówienie, instruktor, specjalizacja, grupa zajęciowa, zajęcia i sala oraz ze względu na związki wiele-wiele dodano tablice asocjacyjne: rezerwacja i certyfikat. Dla każdej encji dobrano odpowiednie atrybuty oraz po wcześniejszej analizie relacji między obiektami zastosowano połączenia 1:N. W zależności od potrzeb część z nich jest wymagana, a część opcjonalna.



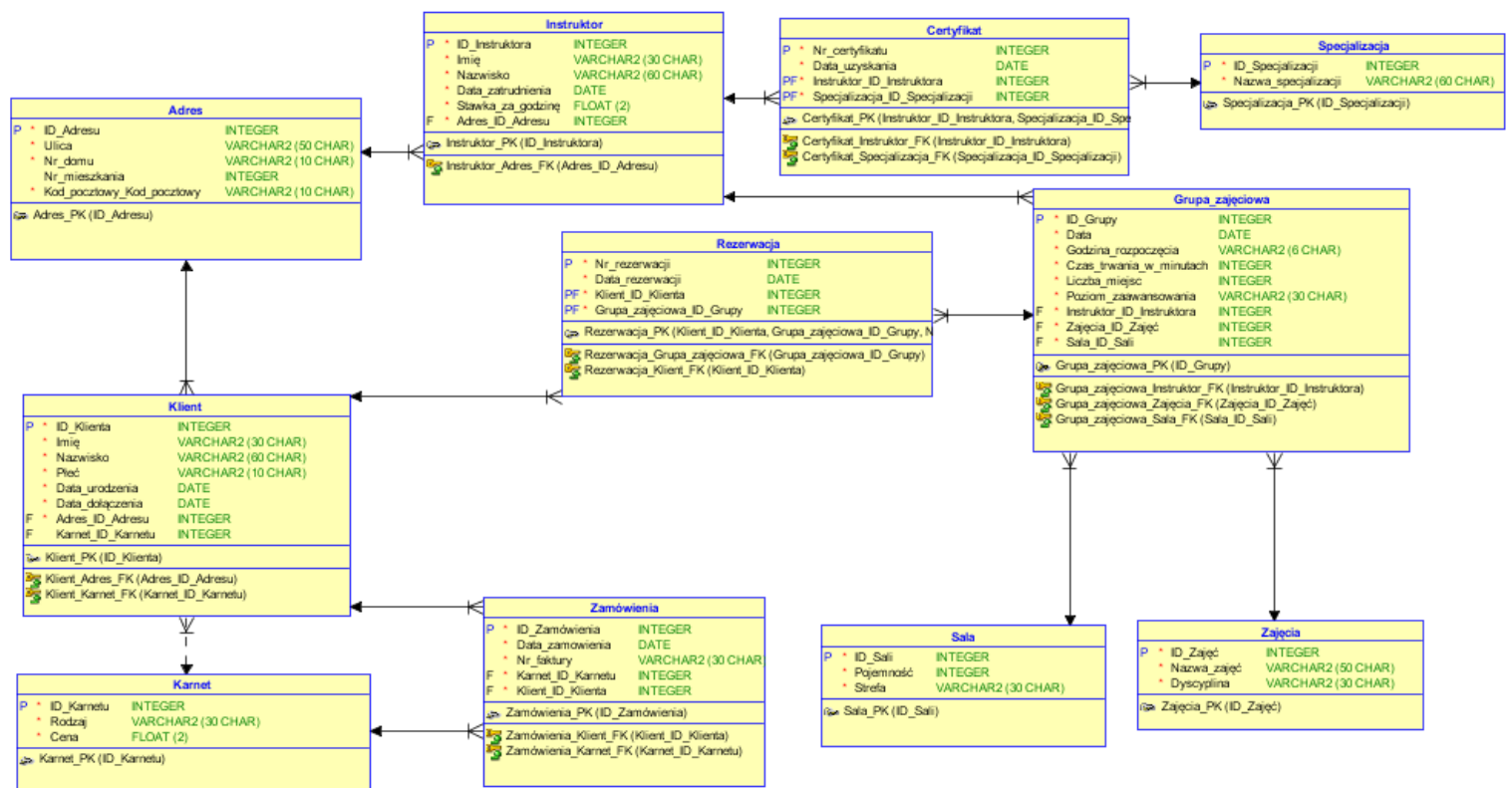
Projekt logiczny

Przejście z modelu ERD na model relacyjny

Na podstawie analizy modelu ERD utworzone zostały tabele o takich samych nazwach zawierające identyczne atrybuty. Dla każdej tabeli ustawiony został unikalny klucz główny. Ponadto ustawiono klucze obce w tabelach: Klient, Zamówienia, Instruktor i Grupa_zajęciowa, co tworzy związki z innymi tabelami. Dodatkowo tabele Rezerwacja i Certyfikat, jako tabele asocjacyjne, posiadają klucze główne-obce.

Diagram relacyjnej bazy danych

Utworzony został diagram relacyjnej bazy danych przedstawiający schematy tabel wraz ze związkami opisanymi wyżej.

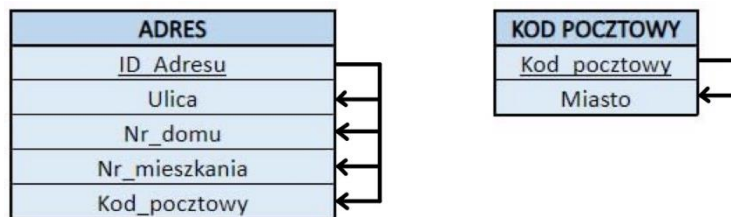


Analiza zależności funkcyjnych i normalizacja tabel

Tabela Adres



1NF	Tabela ADRES spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) ID_Adresu.
2NF	Tabela ADRES spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela ADRES nie spełnia warunków trzeciej postaci normalnej, ponieważ atrybut Kod_pocztowy jest tranzytywnie zależny od klucza głównego ID_Adresu poprzez atrybut Miasto.



3NF	Tabela ADRES spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.
-----	---

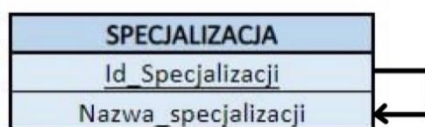
1NF	Tabela KOD POCZTOWY spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) Kod_pocztowy.
2NF	Tabela KOD POCZTOWY spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela KOD POCZTOWY spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.

Tabela Instruktor



1NF	Tabela INSTRUKTOR spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) Id_Instruktora.
2NF	Tabela INSTRUKTOR spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela INSTRUKTOR spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.

Tabela Specjalizacja



1NF	Tabela SPECJALIZACJA spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) ID_Specjalizacji.
2NF	Tabela SPECJALIZACJA spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela SPECJALIZACJA spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.

Tabela Certyfikat



1NF	Tabela CERTYFIKAT spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) Nr_certyfikatu.
2NF	Tabela CERTYFIKAT spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela CERTYFIKAT spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.

Tabela Klient



1NF	Tabela KLIENT spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) ID_Klienta.
2NF	Tabela KLIENT spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela KLIENT spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.

Tabela Karnet



1NF	Tabela KARNET spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) ID_Karnetu.
2NF	Tabela KARNET spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela KARNET spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.

Tabela Zamówienie



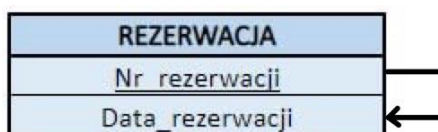
1NF	Tabela ZAMÓWIENIE spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) ID_Zamówienia.
2NF	Tabela ZAMÓWIENIE spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela ZAMÓWIENIE spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.

Tabela Grupa zajęciowa



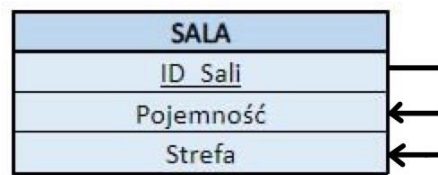
1NF	Tabela GRUPA ZAJĘCIOWA spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) ID_Grupy.
2NF	Tabela GRUPA ZAJĘCIOWA spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela GRUPA ZAJĘCIOWA spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.

Tabela Rezerwacja



1NF	Tabela REZERWACJA spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) Nr_rezerwacji.
2NF	Tabela REZERWACJA spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela REZERWACJA spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.

Tabela Sala



1NF	Tabela SALA spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) ID_Sali.
2NF	Tabela SALA spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela SALA spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.

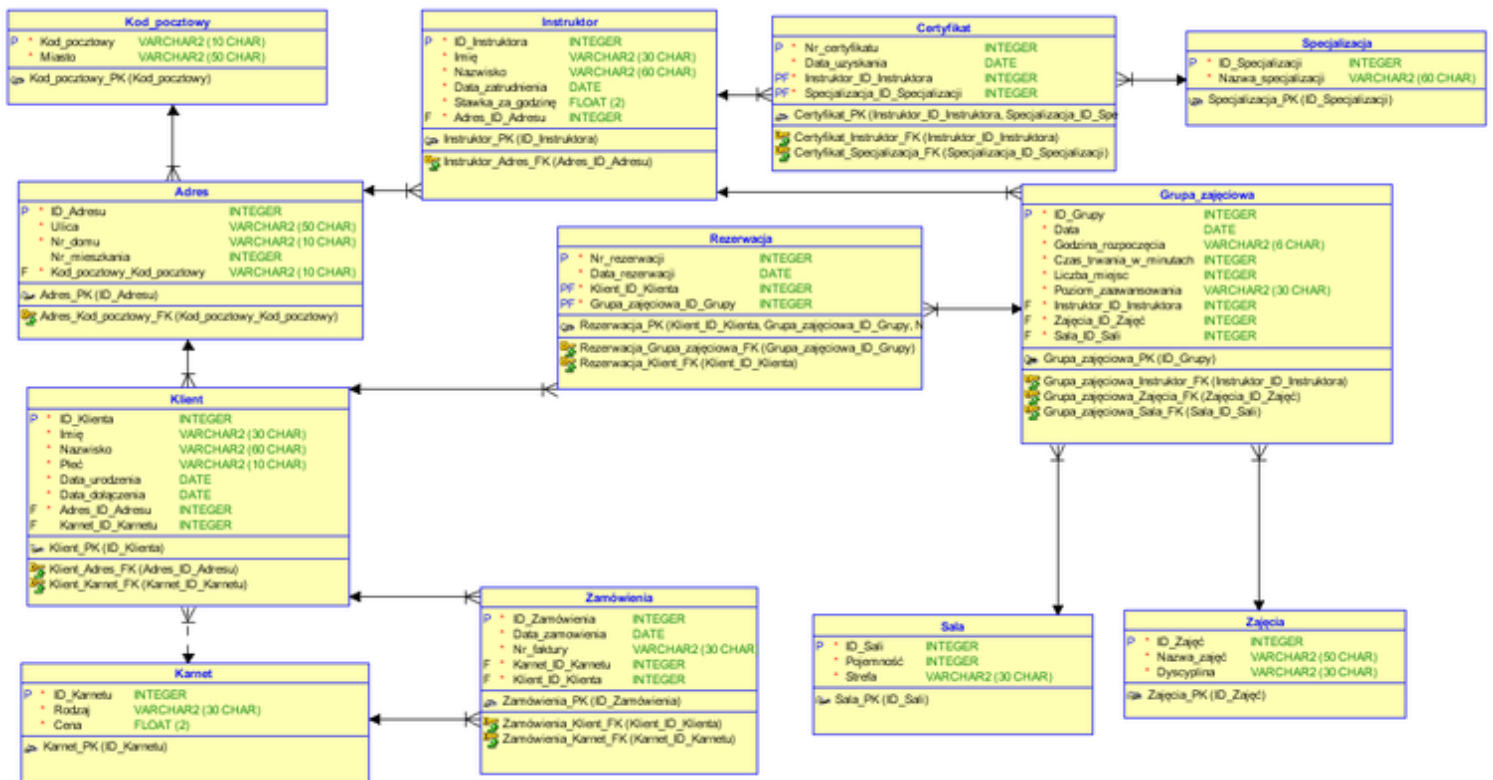
Tabela Zajęcia



1NF	Tabela ZAJĘCIA spełnia warunki pierwszej postaci normalnej, ponieważ każde przecięcie wiersza i kolumny zawiera jedną i tylko jedną wartość oraz posiada unikalny identyfikator (klucz główny) ID_Zajęc.
2NF	Tabela ZAJĘCIA spełnia warunki drugiej postaci normalnej, ponieważ zawiera atrybuty, które odpowiadają tylko temu obiektowi oraz wszystkie atrybuty niekluczowe są w pełni zależne od klucza głównego.
3NF	Tabela ZAJĘCIA spełnia warunki trzeciej postaci normalnej, ponieważ żaden z atrybutów, który nie jest kluczem głównym, nie jest tranzytywnie zależny od klucza głównego poprzez inny atrybut.

Diagram relacyjnej bazy danych po normalizacji

Po dokonaniu normalizacji do diagramu relacyjnej bazy danych została dodana tabela Kod_pocztowy oraz klucz obcy w tabeli Adres.



Propozycje operacji na danych

Przygotowane zostały przykładowe kwerendy z wykorzystaniem notacji operacji algebry relacyjnej. Kwerendy zapisano w języku SQL i przedstawiono uzyskane wyniki.

- Lista klientów, którzy nie mieszkają w Rzeszowie

$$\pi_{\langle \text{imię}, \text{nazwisko}, \text{miasto} \rangle} (\sigma_{\text{miasto} \neq 'Rzeszów'} (\text{klient} \bowtie \text{adres} \bowtie \text{kod_pocztowy}))$$

- Lista sal, w których odbywają się zajęcia w dniach 23-25 stycznia 2024

$$\pi_{\text{data}, \text{id_sali}} (\sigma_{\text{data} \geq 2024-01-23 \wedge \text{data} \leq 2024-01-25} (\text{grupa_zajęciowa} \bowtie \text{sala}))$$

- Lista liczby sprzedanych karmetów dla każdego rodzaju

$$\pi_{\text{rodzaj}, \text{rodzaj}} \Sigma_{\text{liczba_data_zamowienia}} (\sigma_{\text{data_zamowienia} \geq 2023-01-01 \wedge \text{data_zamowienia} \leq 2023-12-31} (\text{karmet} \bowtie \text{zamowienia}))$$

- Lista instruktorów posiadających specjalizację “Sztuki walki”

$$\pi_{\text{imię,nazwisko,stawka_za_godzine,nazwa_specjalizacji}} \left(\sigma_{\text{nazwa_specjalizacji}='Sztuki walki'} (\text{instruktor} \bowtie \text{certyfikat} \bowtie \text{specjalizacja}) \right)$$

- Lista klientek urodzonych po roku 1994 posiadających karnet studencki

$$\pi_{\text{imię,nazwisko,data_urodzenia,rodzaj}} \left(\sigma_{\text{data_urodzenia} \geq 1994-01-01 \wedge \text{płeć}='Kobieta' \wedge \text{rodzaj}='student'} (\text{klient} \bowtie \text{karnet}) \right)$$

- Średnie zarobki instruktorów, którzy posiadają specjalizację "Fitness"

$$\pi_{\text{nazwa_specjalizacji}} \bowtie \text{średnia stawka_za_godzinę} \left(\sigma_{\text{nazwa_specjalizacji}='Fitness'} (\text{instruktor} \bowtie \text{certyfikat} \bowtie \text{specjalizacja}) \right)$$

- Lista faktur z zamówień na karnety VIP złożonych po 2. połowie 2023 roku oraz nazwisk osób, które złożyły te zamówienia

$$\pi_{\text{data_zamowienia,nr_faktury,rodzaj,nazwisko}} \left(\sigma_{\text{rodzaj}='VIP' \wedge \text{data_zamowienia} \geq 2023-07-01} (\text{zamówienia} \bowtie \text{karnet} \bowtie \text{klient}) \right)$$

- Lista karnetów z cenami podstawowymi i cenami po obniżce o 20%

$$\pi_{\text{rodzaj,cena,cena} \cdot 0,8} (\text{karnet})$$

- Lista minimalnych, maksymalnych i średnich pojemności sal w każdej strefie

$$\pi_{\text{strefa,strefa} \bowtie \text{MIN pojemność,MAX pojemność,średnia pojemność}} (\text{sala})$$

- Lista liczby grup dla każdego typu zajęć

$$\pi_{\text{nazwa_zajęć,nazwa_zajęć} \bowtie \text{liczba grupa_zajęciowa}} (\text{zajęcia} \bowtie \text{grupa_zajęciowa})$$

Projekt implementacyjny

Tworzenie tabel

Skrypt SQL, za pomocą którego zostały utworzone tabele wraz z odpowiednimi ograniczeniami:

```
CREATE TABLE adres (  
    id_adresu          INTEGER NOT NULL,  
    ulica              VARCHAR2(50 CHAR) NOT NULL,  
    nr_domu            VARCHAR2(10 CHAR) NOT NULL,  
    nr_mieszkania      INTEGER,  
    kod_pocztowy_kod_pocztowy VARCHAR2(10 CHAR) NOT NULL  
);  
  
ALTER TABLE adres ADD CONSTRAINT adres_pk PRIMARY KEY ( id_adresu );  
  
CREATE TABLE certyfikat (  
    nr_certyfikatu      INTEGER NOT NULL,  
    data_uzyskania      DATE NOT NULL,  
    instruktor_id_instruktora  INTEGER NOT NULL,  
    specjalizacja_id_specjalizacji  INTEGER NOT NULL  
);  
  
ALTER TABLE certyfikat  
    ADD CONSTRAINT certyfikat_pk PRIMARY KEY ( instruktor_id_instruktora,  
                                                specjalizacja_id_specjalizacji,  
                                                nr_certyfikatu );
```

```

CREATE TABLE grupa_zajęciowa (
    id_grupy          INTEGER NOT NULL,
    data              DATE NOT NULL,
    godzina_rozpoczęcia  VARCHAR2(6 CHAR) NOT NULL,
    czas_trwania_w_minutach  INTEGER NOT NULL,
    liczba_miejsc       INTEGER NOT NULL,
    poziom_zaawansowania  VARCHAR2(30 CHAR) NOT NULL,
    instruktor_id_instruktor  INTEGER NOT NULL,
    zajęcia_id_zajęć     INTEGER NOT NULL,
    sala_id_sali        INTEGER NOT NULL
);

ALTER TABLE grupa_zajęciowa ADD CONSTRAINT grupa_zajęciowa_pk PRIMARY KEY (
id_grupy );

```

```

CREATE TABLE instruktor (
    id_instruktor  INTEGER NOT NULL,
    imię          VARCHAR2(30 CHAR) NOT NULL,
    nazwisko      VARCHAR2(60 CHAR) NOT NULL,
    data_zatrudnienia DATE NOT NULL,
    stawka_za_godzinę NUMBER(5, 2) NOT NULL,
    adres_id_adresu  INTEGER NOT NULL
);

ALTER TABLE instruktor ADD CONSTRAINT instruktor_pk PRIMARY KEY ( id_instruktor );

```

```

CREATE TABLE karnet (
    id_karnetu  INTEGER NOT NULL,
    rodzaj     VARCHAR2(30 CHAR) NOT NULL,
    cena       NUMBER(5, 2) NOT NULL
);

ALTER TABLE karnet ADD CONSTRAINT karnet_pk PRIMARY KEY ( id_karnetu );

```



```

CREATE TABLE klient (
    id_klienta    INTEGER NOT NULL,
    imię         VARCHAR2(30 CHAR) NOT NULL,
    nazwisko      VARCHAR2(60 CHAR) NOT NULL,
    płeć         VARCHAR2(10 CHAR) NOT NULL,
    data_urodzenia DATE NOT NULL,
    data_dołączenia DATE NOT NULL,
    adres_id_adresu INTEGER NOT NULL,
    karnet_id_karnetu INTEGER
);

ALTER TABLE klient ADD CONSTRAINT klient_pk PRIMARY KEY ( id_klienta );

CREATE TABLE kod_pocztowy (
    kod_pocztowy VARCHAR2(10 CHAR) NOT NULL,
    miasto       VARCHAR2(50 CHAR) NOT NULL
);

ALTER TABLE kod_pocztowy ADD CONSTRAINT kod_pocztowy_pk PRIMARY KEY (
kod_pocztowy );

CREATE TABLE rezerwacja (
    nr_rezerwacji    INTEGER NOT NULL,
    data_rezerwacji   DATE NOT NULL,
    klient_id_klienta INTEGER NOT NULL,
    grupa_zajęciowa_id_grupy INTEGER NOT NULL
);

ALTER TABLE rezerwacja
    ADD CONSTRAINT rezerwacja_pk PRIMARY KEY ( klient_id_klienta,
                                                grupa_zajęciowa_id_grupy,
                                                nr_rezerwacji );

```

```

CREATE TABLE sala (
    id_sali INTEGER NOT NULL,
    pojemność INTEGER NOT NULL,
    strefa VARCHAR2(30 CHAR) NOT NULL
);
ALTER TABLE sala ADD CONSTRAINT sala_pk PRIMARY KEY ( id_sali );

```

```

CREATE TABLE specjalizacja (
    id_specjalizacji INTEGER NOT NULL,
    nazwa_specjalizacji VARCHAR2(60 CHAR) NOT NULL
);
ALTER TABLE specjalizacja ADD CONSTRAINT specjalizacja_pk PRIMARY KEY (
id_specjalizacji );

```

```

CREATE TABLE zajęcia (
    id_zajęc INTEGER NOT NULL,
    nazwa_zajęc VARCHAR2(50 CHAR) NOT NULL,
    dyscyplina VARCHAR2(30 CHAR) NOT NULL
);
ALTER TABLE zajęcia ADD CONSTRAINT zajęcia_pk PRIMARY KEY ( id_zajęc );

```

```

CREATE TABLE zamówienia (
    id_zamówienia INTEGER NOT NULL,
    data_zamowienia DATE NOT NULL,
    nr_faktury VARCHAR2(30 CHAR) NOT NULL,
    karnet_id_karnetu INTEGER NOT NULL,
    klient_id_klienta INTEGER NOT NULL
);
ALTER TABLE zamówienia ADD CONSTRAINT zamówienia_pk PRIMARY KEY (
id_zamówienia );

```

```

ALTER TABLE adres
    ADD CONSTRAINT adres_kod_pocztowy_fk FOREIGN KEY ( kod_pocztowy_kod_pocztowy )

```

```

REFERENCES kod_pocztowy ( kod_pocztowy );

ALTER TABLE certyfikat
ADD CONSTRAINT certyfikat_instruktor_fk FOREIGN KEY ( instruktor_id_instruktor )
REFERENCES instruktor ( id_instruktor );

ALTER TABLE certyfikat
ADD CONSTRAINT certyfikat_specjalizacja_fk FOREIGN KEY ( specjalizacja_id_specjalizacji )
REFERENCES specjalizacja( id_specjalizacji );

ALTER TABLE grupa_zajęciowa
ADD CONSTRAINT grupa_zajęciowa_instruktor_fk FOREIGN KEY ( instruktor_id_instruktor )
REFERENCES instruktor ( id_instruktor );

ALTER TABLE grupa_zajęciowa
ADD CONSTRAINT grupa_zajęciowa_sala_fk FOREIGN KEY ( sala_id_sali )
REFERENCES sala ( id_sali );

ALTER TABLE grupa_zajęciowa
ADD CONSTRAINT grupa_zajęciowa_zajęcia_fk FOREIGN KEY ( zajęcia_id_zajęć )
REFERENCES zajęcia ( id_zajęć );

ALTER TABLE instruktor
ADD CONSTRAINT instruktor_adres_fk FOREIGN KEY ( adres_id_adresu )
REFERENCES adres ( id_adresu );

ALTER TABLE klient
ADD CONSTRAINT klient_adres_fk FOREIGN KEY ( adres_id_adresu )
REFERENCES adres ( id_adresu );

ALTER TABLE klient
ADD CONSTRAINT klient_karnet_fk FOREIGN KEY ( karnet_id_karnetu )
REFERENCES karnet ( id_karnetu );

```

ALTER TABLE rezerwacja

ADD CONSTRAINT rezerwacja_grupa_zajęciowa_fk FOREIGN KEY (grupa_zajęciowa_id_grupy)

REFERENCES grupa_zajęciowa (id_grupy);

ALTER TABLE rezerwacja

ADD CONSTRAINT rezerwacja_klient_fk FOREIGN KEY (klient_id_klienta)

REFERENCES klient (id_klienta);

ALTER TABLE zamówienia

ADD CONSTRAINT zamówienia_karnet_fk FOREIGN KEY (karnet_id_karnetu)

REFERENCES karnet (id_karnetu);

ALTER TABLE zamówienia

ADD CONSTRAINT zamówienia_klient_fk FOREIGN KEY (klient_id_klienta)

REFERENCES klient (id_klienta);

Wprowadzenie danych do tabel

Skrypt SQL, za pomocą którego tabele zostały uzupełnione danymi:

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('35-001', 'Rzeszów');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('35-002', 'Rzeszów');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('35-003', 'Rzeszów');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('35-004', 'Rzeszów');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('35-005', 'Rzeszów');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('35-006', 'Rzeszów');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('35-007', 'Rzeszów');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('35-008', 'Rzeszów');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('35-009', 'Rzeszów');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('35-010', 'Rzeszów');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('36-002', 'Jasionka');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('36-047', 'Niechobrz');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('37-100', 'Łańcut');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('38-100', 'Strzyżów');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('36-100', 'Kolbuszowa');

INSERT INTO kod_pocztowy (kod_pocztowy, miasto) VALUES ('36-040', 'Boguchwała');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (1, 'ul. Mickiewicza', '10', 3, '35-001');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (2, 'ul. Piłsudskiego', '22', NULL, '35-002')

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (3, 'ul. Krakowska', '5A', 7, '35-003');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (4, 'ul. Dąbrowskiego', '8', NULL, '35-004');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (5, 'ul. Hetmańska', '12', 15, '35-005');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (6, 'ul. Staromiejska', '3C', 2, '35-006');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (7, 'ul. Okopowa', '7', NULL, '35-007');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (8, 'ul. Rejtana', '14', 4, '35-008');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (9, 'ul. Lubelska', '9', NULL, '35-009');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (10, 'ul. Rycerska', '2', 10, '35-010');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (11, 'Jasionka', '18', NULL, '36-002');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (12, 'Niechobrz', '5', NULL, '36-047');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (13, 'ul. Asnyka', '11', 2, '37-100');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (14, 'ul. Czarnieckiego', '4D', 8, '38-100');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)

```

VALUES (15, 'ul. Akacjowa', '15', NULL, '36-100');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (16, 'ul. Brzozowa', '65', 56, '36-100');

INSERT INTO adres (id_adresu, ulica, nr_domu, nr_mieszkania, kod_pocztowy_kod_pocztowy)
VALUES (17, 'ul. Cicha', '134C', 56, '36-040');


INSERT INTO instruktor (id_instruktor, imię, nazwisko, data_zatrudnienia, stawka_za_godzinę,
adres_id_adresu)
VALUES (1, 'Anna', 'Kowalska', TO_DATE('2017-01-01', 'YYYY-MM-DD'), 50.00, 3);

INSERT INTO instruktor (id_instruktor, imię, nazwisko, data_zatrudnienia, stawka_za_godzinę,
adres_id_adresu)
VALUES (2, 'Jan', 'Nowak', TO_DATE('2018-05-15', 'YYYY-MM-DD'), 50.50, 5);

INSERT INTO instruktor (id_instruktor, imię, nazwisko, data_zatrudnienia, stawka_za_godzinę,
adres_id_adresu)
VALUES (3, 'Magdalena', 'Kaczor', TO_DATE('2019-09-03', 'YYYY-MM-DD'), 45.00, 10);

INSERT INTO instruktor (id_instruktor, imię, nazwisko, data_zatrudnienia, stawka_za_godzinę,
adres_id_adresu)
VALUES (4, 'Piotr', 'Lis', TO_DATE('2020-02-20', 'YYYY-MM-DD'), 48.00, 7);

INSERT INTO instruktor (id_instruktor, imię, nazwisko, data_zatrudnienia, stawka_za_godzinę,
adres_id_adresu)
VALUES (5, 'Anna', 'Kozyra', TO_DATE('2020-04-10', 'YYYY-MM-DD'), 52.00, 1);

INSERT INTO instruktor (id_instruktor, imię, nazwisko, data_zatrudnienia, stawka_za_godzinę,
adres_id_adresu)
VALUES (6, 'Krzysztof', 'Zajac', TO_DATE('2021-06-22', 'YYYY-MM-DD'), 53.50, 14);

INSERT INTO instruktor (id_instruktor, imię, nazwisko, data_zatrudnienia, stawka_za_godzinę,
adres_id_adresu)
VALUES (7, 'Tomasz', 'Kowalski', TO_DATE('2018-11-05', 'YYYY-MM-DD'), 47.50, 3);

INSERT INTO instruktor (id_instruktor, imię, nazwisko, data_zatrudnienia, stawka_za_godzinę,
adres_id_adresu)
VALUES (8, 'Karolina', 'Wójcik', TO_DATE('2019-03-12', 'YYYY-MM-DD'), 56.00, 12);

INSERT INTO instruktor (id_instruktor, imię, nazwisko, data_zatrudnienia, stawka_za_godzinę,
adres_id_adresu)
VALUES (9, 'Jakub', 'Szymanik', TO_DATE('2020-07-08', 'YYYY-MM-DD'), 62.50, 16);

INSERT INTO instruktor (id_instruktor, imię, nazwisko, data_zatrudnienia, stawka_za_godzinę,
adres_id_adresu)
VALUES (10, 'Anna', 'Nowak', TO_DATE('2021-09-15', 'YYYY-MM-DD'), 51.50, 5);

```

```

INSERT INTO specjalizacja (id_specjalizacji, nazwa_specjalizacji) VALUES (1, 'Fitness');
INSERT INTO specjalizacja (id_specjalizacji, nazwa_specjalizacji) VALUES (2, 'Sztuki walki');
INSERT INTO specjalizacja (id_specjalizacji, nazwa_specjalizacji) VALUES (3, 'Taniec');
INSERT INTO specjalizacja (id_specjalizacji, nazwa_specjalizacji) VALUES (4, 'Trening siłowy');
INSERT INTO specjalizacja (id_specjalizacji, nazwa_specjalizacji) VALUES (5, 'Stretching');
INSERT INTO specjalizacja (id_specjalizacji, nazwa_specjalizacji) VALUES (6, 'Gimnastyka');

```

```

INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktor,
specjalizacja_id_specjalizacji)

```

```

VALUES ('12345678', TO_DATE('2016-01-01', 'YYYY-MM-DD'), 1, 1);

```

```

INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktor,
specjalizacja_id_specjalizacji)

```

```

VALUES ('98765432', TO_DATE('2018-03-15', 'YYYY-MM-DD'), 2, 2);

```

```

INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktor,
specjalizacja_id_specjalizacji)

```

```

VALUES ('54321098', TO_DATE('2017-02-01', 'YYYY-MM-DD'), 3, 3);

```

```

INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktor,
specjalizacja_id_specjalizacji)

```

```

VALUES ('87654329', TO_DATE('2019-04-10', 'YYYY-MM-DD'), 4, 4);

```

```

INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktor,
specjalizacja_id_specjalizacji)

```

```

VALUES ('13579248', TO_DATE('2020-03-01', 'YYYY-MM-DD'), 5, 5);

```

```

INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktor,
specjalizacja_id_specjalizacji)

```

```

VALUES ('24680137', TO_DATE('2015-05-20', 'YYYY-MM-DD'), 6, 5);

```

```

INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktor,
specjalizacja_id_specjalizacji)

```

```

VALUES ('98765438', TO_DATE('2016-04-05', 'YYYY-MM-DD'), 7, 1);

```

```

INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktor,
specjalizacja_id_specjalizacji)

```

```

VALUES ('12345679', TO_DATE('2018-06-15', 'YYYY-MM-DD'), 8, 2);

```

```

INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktor,
specjalizacja_id_specjalizacji)

```

```

VALUES ('54321091', TO_DATE('2020-05-01', 'YYYY-MM-DD'), 9, 4);

```

```
INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktora,  
specjalizacja_id_specjalizacji)
```

```
VALUES ('87654322', TO_DATE('2017-07-10', 'YYYY-MM-DD'), 10, 6);
```

```
INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktora,  
specjalizacja_id_specjalizacji)
```

```
VALUES ('98765431', TO_DATE('2023-08-01', 'YYYY-MM-DD'), 1, 3);
```

```
INSERT INTO certyfikat (nr_certyfikatu, data_uzyskania, instruktor_id_instruktora,  
specjalizacja_id_specjalizacji)
```

```
VALUES ('12345672', TO_DATE('2022-09-15', 'YYYY-MM-DD'), 9, 1);
```

```
INSERT INTO karnet (id_karnetu, rodzaj, cena)
```

```
VALUES (1, 'podstawowy', 130.00);
```

```
INSERT INTO karnet (id_karnetu, rodzaj, cena)
```

```
VALUES (2, 'student', 80.00);
```

```
INSERT INTO karnet (id_karnetu, rodzaj, cena)
```

```
VALUES (3, 'senior', 100.00);
```

```
INSERT INTO karnet (id_karnetu, rodzaj, cena)
```

```
VALUES (4, 'VIP', 180.00);
```

```
INSERT INTO klient (id_klienta, imię, nazwisko, płeć, data_urodzenia, data_dołączenia,  
adres_id_adresu, karnet_id_karnetu)
```

```
VALUES (1, 'Alicja', 'Kowalska', 'Kobieta', TO_DATE('1990-05-15', 'YYYY-MM-DD'),  
TO_DATE('2021-02-10', 'YYYY-MM-DD'), 3, 2);
```

```
INSERT INTO klient (id_klienta, imię, nazwisko, płeć, data_urodzenia, data_dołączenia,  
adres_id_adresu, karnet_id_karnetu)
```

```
VALUES (2, 'Bartosz', 'Nowak', 'Mężczyzna', TO_DATE('1985-08-22', 'YYYY-MM-DD'),  
TO_DATE('2019-07-05', 'YYYY-MM-DD'), 5, 1);
```

```
INSERT INTO klient (id_klienta, imię, nazwisko, płeć, data_urodzenia, data_dołączenia,  
adres_id_adresu, karnet_id_karnetu)
```

```
VALUES (3, 'Karolina', 'Kita', 'Kobieta', TO_DATE('1978-03-10', 'YYYY-MM-DD'),  
TO_DATE('2020-09-18', 'YYYY-MM-DD'), 8, 1);
```

```
INSERT INTO klient (id_klienta, imię, nazwisko, płeć, data_urodzenia, data_dołączenia,  
adres_id_adresu, karnet_id_karnetu)
```

```
VALUES (4, 'Marcin', 'Kita', 'Mężczyzna', TO_DATE('1982-12-05', 'YYYY-MM-DD'),  
TO_DATE('2018-06-30', 'YYYY-MM-DD'), 8, 4);
```

```
INSERT INTO klient (id_klienta, imię, nazwisko, płeć, data_urodzenia, data_dołączenia,  
adres_id_adresu, karnet_id_karnetu)
```



```

VALUES (5, 'Anna', 'Zielińska', 'Kobieta', TO_DATE('1995-09-18', 'YYYY-MM-DD'),
TO_DATE('2022-01-15', 'YYYY-MM-DD'), 17, 2);

INSERT INTO klient (id_klienta, imię, nazwisko, płeć, data_urodzenia, data_dołączenia,
adres_id_adresu, karnet_id_karnetu)

VALUES (6, 'Mateusz', 'Pawłowski', 'Mężczyzna', TO_DATE('1965-11-30', 'YYYY-MM-DD'),
TO_DATE('2023-03-22', 'YYYY-MM-DD'), 6, 3);

INSERT INTO klient (id_klienta, imię, nazwisko, płeć, data_urodzenia, data_dołączenia,
adres_id_adresu, karnet_id_karnetu)

VALUES (7, 'Katarzyna', 'Zych', 'Kobieta', TO_DATE('1988-07-12', 'YYYY-MM-DD'),
TO_DATE('2019-10-05', 'YYYY-MM-DD'), 13, 1);

INSERT INTO klient (id_klienta, imię, nazwisko, płeć, data_urodzenia, data_dołączenia,
adres_id_adresu, karnet_id_karnetu)

VALUES (8, 'Piotr', 'Szymański', 'Mężczyzna', TO_DATE('1980-01-25', 'YYYY-MM-DD'),
TO_DATE('2022-04-15', 'YYYY-MM-DD'), 8, 4);

INSERT INTO klient (id_klienta, imię, nazwisko, płeć, data_urodzenia, data_dołączenia,
adres_id_adresu, karnet_id_karnetu)

VALUES (9, 'Magdalena', 'Kowalczyk', 'Kobieta', TO_DATE('1959-06-08', 'YYYY-MM-DD'),
TO_DATE('2020-11-08', 'YYYY-MM-DD'), 9, 3);

INSERT INTO klient (id_klienta, imię, nazwisko, płeć, data_urodzenia, data_dołączenia,
adres_id_adresu, karnet_id_karnetu)

VALUES (10, 'Łukasz', 'Duda', 'Mężczyzna', TO_DATE('1973-04-14', 'YYYY-MM-DD'),
TO_DATE('2023-07-25', 'YYYY-MM-DD'), 11, 1);

INSERT INTO klient (id_klienta, imię, nazwisko, płeć, data_urodzenia, data_dołączenia,
adres_id_adresu, karnet_id_karnetu)

VALUES (11, 'Wiktor', 'Czajka', 'Mężczyzna', TO_DATE('2002-04-19', 'YYYY-MM-DD'),
TO_DATE('2023-12-12', 'YYYY-MM-DD'), 15, NULL);

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

VALUES (1, TO_DATE('2022-01-10', 'YYYY-MM-DD'), 'A1B2C3D4E5', 1, 1);

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

VALUES (2, TO_DATE('2023-04-05', 'YYYY-MM-DD'), 'F6G7H8I9J0', 1, 2)

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

VALUES (3, TO_DATE('2022-07-20', 'YYYY-MM-DD'), 'K1L2M3N4O5', 1, 3);

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

```

```

VALUES (4, TO_DATE('2023-09-15', 'YYYY-MM-DD'), 'P6Q7R8S9T0', 4, 4);

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

VALUES (5, TO_DATE('2022-12-01', 'YYYY-MM-DD'), 'U1V2W3X4Y5', 2, 5);

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

VALUES (6, TO_DATE('2023-05-10', 'YYYY-MM-DD'), 'Z6A7B8C9D0', 3, 6);

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

VALUES (7, TO_DATE('2022-10-15', 'YYYY-MM-DD'), 'E1F2G3H4I5', 1, 7);

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

VALUES (8, TO_DATE('2023-03-22', 'YYYY-MM-DD'), 'J6K7L8M9N0', 4, 8);

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

VALUES (9, TO_DATE('2022-08-03', 'YYYY-MM-DD'), 'E3F4G5H6I', 3, 9);

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

VALUES (10, TO_DATE('2023-07-01', 'YYYY-MM-DD'), 'A1B2C3D4E5', 1, 10);

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

VALUES (11, TO_DATE('2023-07-05', 'YYYY-MM-DD'), 'A1B2C3F45D', 2, 1);

INSERT INTO zamówienia (id_zamówienia, data_zamowienia, nr_faktury, karnet_id_karnetu,
klient_id_klienta)

VALUES (12, TO_DATE('2023-03-09', 'YYYY-MM-DD'), 'M75DF3D4E5', 2, 10);


INSERT INTO zajęcia (id_zajęć, nazwa_zajęć, dyscyplina)

VALUES (1, 'Aerobik', 'Fitness');

INSERT INTO zajęcia (id_zajęć, nazwa_zajęć, dyscyplina)

VALUES (2, 'Zumba', 'Taniec');

INSERT INTO zajęcia (id_zajęć, nazwa_zajęć, dyscyplina)

VALUES (3, 'Gimnastyka', 'Gimnastyka');

INSERT INTO zajęcia (id_zajęć, nazwa_zajęć, dyscyplina)

VALUES (4, 'Joga', 'Stretching');

INSERT INTO zajęcia (id_zajęć, nazwa_zajęć, dyscyplina)

```

```
VALUES (5, 'Boks', 'Sztuki walki');
INSERT INTO zajęcia (id_zajęc, nazwa_zajęc, dyscyplina)
VALUES (6, 'Judo', 'Sztuki walki');
INSERT INTO zajęcia (id_zajęc, nazwa_zajęc, dyscyplina)
VALUES (7, 'Zdrowy kręgosłup', 'Stretching');
INSERT INTO zajęcia (id_zajęc, nazwa_zajęc, dyscyplina)
VALUES (8, 'Tabata', 'Fitness');
INSERT INTO zajęcia (id_zajęc, nazwa_zajęc, dyscyplina)
VALUES (9, 'Kettlebell', 'Trening siłowy');
```

```
INSERT INTO sala (id_sali, pojemność, strefa)
VALUES (1, 30, 'Fitness');
INSERT INTO sala (id_sali, pojemność, strefa)
VALUES (2, 25, 'Calm');
INSERT INTO sala (id_sali, pojemność, strefa)
VALUES (3, 15, 'Strong');
INSERT INTO sala (id_sali, pojemność, strefa)
VALUES (4, 20, 'Dance');
INSERT INTO sala (id_sali, pojemność, strefa)
VALUES (5, 10, 'Fight');
INSERT INTO sala (id_sali, pojemność, strefa)
VALUES (6, 28, 'Fitness');
INSERT INTO sala (id_sali, pojemność, strefa)
VALUES (7, 32, 'Calm');
INSERT INTO sala (id_sali, pojemność, strefa)
VALUES (8, 20, 'Strong');
INSERT INTO sala (id_sali, pojemność, strefa)
VALUES (9, 40, 'Dance');
INSERT INTO sala (id_sali, pojemność, strefa)
VALUES (10, 5, 'Fight');
```

```
INSERT INTO grupa_zajęciowa (id_grupy, data, godzina_rozpoczęcia, czas_trwania_w_minutach,
liczba_miejsc, poziom_zaaawansowania, instruktor_id_instruktor, zajęcia_id_zajęc, sala_id_sali)
```

```

VALUES (1, TO_DATE('2024-01-20', 'YYYY-MM-DD'), '10:00', 60, 25, 'Początkujący', 1, 1, 6);

INSERT INTO grupa_zajęciowa (id_grupy, data, godzina_rozpoczęcia, czas_trwania_w_minutach,
liczba_miejsc, poziom_zaaawansowania, instruktor_id_instruktor, zajęcia_id_zajęć, sala_id_sali)

VALUES (2, TO_DATE('2024-01-21', 'YYYY-MM-DD'), '15:30', 75, 30, 'Średniozaawansowany', 3,
2, 9);

INSERT INTO grupa_zajęciowa (id_grupy, data, godzina_rozpoczęcia, czas_trwania_w_minutach,
liczba_miejsc, poziom_zaaawansowania, instruktor_id_instruktor, zajęcia_id_zajęć, sala_id_sali)

VALUES (3, TO_DATE('2024-01-22', 'YYYY-MM-DD'), '12:45', 90, 40, 'Zaawansowany', 2, 5, 5);

INSERT INTO grupa_zajęciowa (id_grupy, data, godzina_rozpoczęcia, czas_trwania_w_minutach,
liczba_miejsc, poziom_zaaawansowania, instruktor_id_instruktor, zajęcia_id_zajęć, sala_id_sali)

VALUES (4, TO_DATE('2024-01-23', 'YYYY-MM-DD'), '18:00', 60, 20, 'Zaawansowany', 5, 4, 2);

INSERT INTO grupa_zajęciowa (id_grupy, data, godzina_rozpoczęcia, czas_trwania_w_minutach,
liczba_miejsc, poziom_zaaawansowania, instruktor_id_instruktor, zajęcia_id_zajęć, sala_id_sali)

VALUES (5, TO_DATE('2024-01-24', 'YYYY-MM-DD'), '11:30', 75, 5, 'Początkujący', 8, 5, 10);

INSERT INTO grupa_zajęciowa (id_grupy, data, godzina_rozpoczęcia, czas_trwania_w_minutach,
liczba_miejsc, poziom_zaaawansowania, instruktor_id_instruktor, zajęcia_id_zajęć, sala_id_sali)

VALUES (6, TO_DATE('2024-01-25', 'YYYY-MM-DD'), '17:15', 90, 10, 'Zaawansowany', 8, 6, 5);

INSERT INTO grupa_zajęciowa (id_grupy, data, godzina_rozpoczęcia, czas_trwania_w_minutach,
liczba_miejsc, poziom_zaaawansowania, instruktor_id_instruktor, zajęcia_id_zajęć, sala_id_sali)

VALUES (7, TO_DATE('2024-01-26', 'YYYY-MM-DD'), '14:45', 60, 32, 'Dla wszystkich', 6, 7, 7);

INSERT INTO grupa_zajęciowa (id_grupy, data, godzina_rozpoczęcia, czas_trwania_w_minutach,
liczba_miejsc, poziom_zaaawansowania, instruktor_id_instruktor, zajęcia_id_zajęć, sala_id_sali)

VALUES (8, TO_DATE('2024-01-27', 'YYYY-MM-DD'), '09:30', 45, 25, 'Średniozaawansowany', 9,
8, 6);

INSERT INTO grupa_zajęciowa (id_grupy, data, godzina_rozpoczęcia, czas_trwania_w_minutach,
liczba_miejsc, poziom_zaaawansowania, instruktor_id_instruktor, zajęcia_id_zajęć, sala_id_sali)

VALUES (9, TO_DATE('2024-01-28', 'YYYY-MM-DD'), '16:00', 90, 15, 'Zaawansowany', 9, 9, 3);


INSERT INTO rezerwacja (nr_rezerwacji, data_rezerwacji, klient_id_klienta,
grupa_zajęciowa_id_grupy)

VALUES (1, TO_DATE('2024-01-10', 'YYYY-MM-DD'), 1, 1);

INSERT INTO rezerwacja (nr_rezerwacji, data_rezerwacji, klient_id_klienta,
grupa_zajęciowa_id_grupy)

VALUES (2, TO_DATE('2024-01-07', 'YYYY-MM-DD'), 2, 3);

INSERT INTO rezerwacja (nr_rezerwacji, data_rezerwacji, klient_id_klienta,
grupa_zajęciowa_id_grupy)

VALUES (3, TO_DATE('2024-01-12', 'YYYY-MM-DD'), 3, 2);

```

```
INSERT INTO rezerwacja (nr_rezerwacji, data_rezerwacji, klient_id_klienta,  
grupa_zajęciowa_id_grupy)  
VALUES (4, TO_DATE('2024-01-15', 'YYYY-MM-DD'), 4, 4);  
  
INSERT INTO rezerwacja (nr_rezerwacji, data_rezerwacji, klient_id_klienta,  
grupa_zajęciowa_id_grupy)  
VALUES (5, TO_DATE('2024-01-08', 'YYYY-MM-DD'), 5, 5);  
  
INSERT INTO rezerwacja (nr_rezerwacji, data_rezerwacji, klient_id_klienta,  
grupa_zajęciowa_id_grupy)  
VALUES (6, TO_DATE('2024-01-11', 'YYYY-MM-DD'), 6, 6);  
  
INSERT INTO rezerwacja (nr_rezerwacji, data_rezerwacji, klient_id_klienta,  
grupa_zajęciowa_id_grupy)  
VALUES (7, TO_DATE('2024-01-09', 'YYYY-MM-DD'), 7, 7);  
  
INSERT INTO rezerwacja (nr_rezerwacji, data_rezerwacji, klient_id_klienta,  
grupa_zajęciowa_id_grupy)  
VALUES (8, TO_DATE('2024-01-14', 'YYYY-MM-DD'), 8, 8);  
  
INSERT INTO rezerwacja (nr_rezerwacji, data_rezerwacji, klient_id_klienta,  
grupa_zajęciowa_id_grupy)  
VALUES (9, TO_DATE('2024-01-06', 'YYYY-MM-DD'), 9, 9);
```

Utworzone tabele po wprowadzeniu danych

- Tabela Kod pocztowy

KOD_POCZTOWY	MIASTO
35-001	Rzeszów
35-002	Rzeszów
35-003	Rzeszów
35-004	Rzeszów
35-005	Rzeszów
35-006	Rzeszów
35-007	Rzeszów
35-008	Rzeszów
35-009	Rzeszów
35-010	Rzeszów
36-002	Jasionka
36-047	Niechobrz
37-100	Łańcut
38-100	Strzyżów
36-100	Kolbuszowa
36-040	Boguchwała

- Tabela Adres

ID_ADRESU	ULICA	NR_DOMU	NR_MIESZKANIA	KOD_POCZTOWY_KOD_POCZTOWY
1	ul. Mickiewicza	10		3 35-001
2	ul. Piłsudskiego	22	(null)	35-002
3	ul. Krakowska	5A		7 35-003
4	ul. Dąbrowskiego	8	(null)	35-004
5	ul. Hetmańska	12		15 35-005
6	ul. Staromiejska	3C		2 35-006
7	ul. Okopowa	7	(null)	35-007
8	ul. Rejtana	14		4 35-008
9	ul. Lubelska	9	(null)	35-009
10	ul. Rycerska	2		10 35-010
11	Jasionka	18	(null)	36-002
12	Niechobrz	5	(null)	36-047
13	ul. Asnyka	11		2 37-100
14	ul. Czarnieckiego	4D		8 38-100
15	ul. Akacjowa	15	(null)	36-100
16	ul. Brzozowa	65		56 36-100
17	ul. Cicha	134C		56 36-040

- Tabela Instruktor

ID_INSTRUKTORA	IMIĘ	NAZWISKO	DATA_ZATRUDNIENIA	STAWKA_ZA_GODZINĘ	ADRES_ID_ADRESU
1	Anna	Kowalska	01-JAN-17	50	3
2	Jan	Nowak	15-MAY-18	50.5	5
3	Magdalena	Kaczor	03-SEP-19	45	10
4	Piotr	Lis	20-FEB-20	48	7
5	Anna	Kozyra	10-APR-20	52	1
6	Krzysztof	Zajac	22-JUN-21	53.5	14
7	Tomasz	Kowalski	05-NOV-18	47.5	3
8	Karolina	Wójcik	12-MAR-19	56	12
9	Jakub	Szymanik	08-JUL-20	62.5	16
10	Anna	Nowak	15-SEP-21	51.5	5

- Tabela Specjalizacja

ID_SPECJALIZACJI	NAZWA_SPECJALIZACJI
1	Fitness
2	Sztuki walki
3	Taniec
4	Trening siłowy
5	Stretching
6	Gimnastyka

- Tabela Certyfikat

NR_CERTYFIKATU	DATA_UZYSKANIA	INSTRUKTOR_ID_INSTRUKTORA	SPECJALIZACJA_ID_SPECJALIZACJI
12345678	01 - JAN -16	1	1
98765432	15 - MAR -18	2	2
54321098	01 - FEB -17	3	3
87654329	10 - APR -19	4	4
13579248	01 - MAR -20	5	5
24680137	20 - MAY -15	6	5
98765438	05 - APR -16	7	1
12345679	15 - JUN -18	8	2
54321091	01 - MAY -20	9	4
87654322	10 - JUL -17	10	6
98765431	01 - AUG -23	1	3
12345672	15 - SEP -22	9	1

- Tabela Karnet

ID_KARNETU	RODZAJ	CENA
1	podstawowy	130
2	student	80
3	senior	100
4	VIP	180

- Tabela Klient

ID_KLIENTA	IMIĘ	NAZWISKO	PLEĆ	DATA_URODZENIA	DATA_DOŁĄCZENIA	ADRES_ID_ADRESU	KARNET_ID_KARNETU
1	Alicja	Kowalska	Kobieta	15-MAY-90	10-FEB-21	3	2
2	Bartosz	Nowak	Mężczyzna	22-AUG-85	05-JUL-19	5	1
3	Karolina	Kita	Kobieta	10-MAR-78	18-SEP-20	8	1
4	Marcin	Kita	Mężczyzna	05-DEC-82	30-JUN-18	8	4
5	Anna	Zielińska	Kobieta	18-SEP-95	15-JAN-22	17	2
6	Mateusz	Pawłowski	Mężczyzna	30-NOV-65	22-MAR-23	6	3
7	Katarzyna	Zych	Kobieta	12-JUL-88	05-OCT-19	13	1
8	Piotr	Szymański	Mężczyzna	25-JAN-80	15-APR-22	8	4
9	Magdalena	Kowalczyk	Kobieta	08-JUN-59	08-NOV-20	9	3
10	Łukasz	Duda	Mężczyzna	14-APR-73	25-JUL-23	11	1
11	Wiktor	Czajka	Mężczyzna	19-APR-02	12-DEC-23	15	(null)

- Tabela Zamówienia

ID_ZAMÓWIENIA	DATA_ZAMOWIENIA	NR_FAKTURY	KARNET_ID_KARNETU	KLIENT_ID_KLIENTA
1	10 - JAN - 22	A1B2C3D4E5	1	1
2	05 - APR - 23	F6G7H8I9J0	1	2
3	20 - JUL - 22	K1L2M3N4O5	1	3
4	15 - SEP - 23	P6Q7R8S9T0	4	4
5	01 - DEC - 22	U1V2W3X4Y5	2	5
6	10 - MAY - 23	Z6A7B8C9D0	3	6
7	15 - OCT - 22	E1F2G3H4I5	1	7
8	22 - MAR - 23	J6K7L8M9N0	4	8
9	03 - AUG - 22	E3F4G5H6I	3	9
10	01 - JUL - 23	A1B2C3D4E5	1	10
11	05 - JUL - 23	A1B2C3F45D	2	1
12	09 - MAR - 23	M75DF3D4E5	2	10

- Tabela Zajęcia

ID_ZAJĘĆ	NAZWA_ZAJĘĆ	DYSCYPLINA
1	Aerobik	Fitness
2	Zumba	Taniec
3	Gimnastyka	Gimnastyka
4	Joga	Stretching
5	Boks	Sztuki walki
6	Judo	Sztuki walki
7	Zdrowy kręgosłup	Stretching
8	Tabata	Fitness
9	Kettlebell	Trening siłowy

- Tabela Sala

ID_SALI	POJEMNOŚĆ	STREFA
1	30	Fitness
2	25	Calm
3	15	Strong
4	20	Dance
5	10	Fight
6	28	Fitness
7	32	Calm
8	20	Strong
9	40	Dance
10	5	Fight

- Tabela Grupa zajęciowa

ID_GRUPY	DATA	GODZINA_ROZPOCZĘCIA_FORMATTED	CZAS_TRWANIA_W_MINUTACH	LICZBA_MIEJSC	POZIOM_ZAAWANSOWANIA	INSTRUKTOR_ID_INSTRUKTORA	ZAJĘCIA_ID_ZAJĘĆ	SALA_ID_SALI
1	20 - JAN - 24	2024-01-20 10:00:00	60	25	Początkujący	1	1	6
2	21 - JAN - 24	2024-01-21 15:30:00	75	30	Średniozaawansowany	3	2	9
3	22 - JAN - 24	2024-01-22 12:45:00	90	40	Zaawansowany	2	5	5
4	23 - JAN - 24	2024-01-23 18:00:00	60	20	Zaawansowany	5	4	2
5	24 - JAN - 24	2024-01-24 11:30:00	75	5	Początkujący	8	5	10
6	25 - JAN - 24	2024-01-25 17:15:00	90	10	Zaawansowany	8	6	5
7	26 - JAN - 24	2024-01-26 14:45:00	60	32	Dla wszystkich	6	7	7
8	27 - JAN - 24	2024-01-27 09:30:00	45	25	Średniozaawansowany	9	8	6
9	28 - JAN - 24	2024-01-28 16:00:00	90	15	Zaawansowany	9	9	3

- Tabela Rezerwacja

NR_REZERWACJI	DATA_REZERWACJI	KLIENT_ID_KLIENTA	GRUPA_ZAJĘCIOWA_ID_GRUPY
1	10 - JAN - 24	1	1
2	07 - JAN - 24	2	3
3	12 - JAN - 24	3	2
4	15 - JAN - 24	4	4
5	08 - JAN - 24	5	5
6	11 - JAN - 24	6	6
7	09 - JAN - 24	7	7
8	14 - JAN - 24	8	8
9	06 - JAN - 24	9	9

Tworzenie kwerend

Dla każdej z kwerend zaproponowanych w części logicznej projektu stworzono skrypt SQL oraz wyświetlono wyniki:

- **Lista klientów, którzy nie mieszkają w Rzeszowie**

```
SELECT imię, nazwisko, kod_pocztowy.miasto
FROM klient
LEFT JOIN adres
ON klient.adres_id_adresu = adres.id_adresu
LEFT JOIN kod_pocztowy
ON adres.kod_pocztowy_kod_pocztowy = kod_pocztowy.kod_pocztowy
WHERE kod_pocztowy.miasto <> 'Rzeszów';
```

	IMIĘ	NAZWISKO	MIASTO
1	Anna	Zielińska	Boguchwała
2	Katarzyna	Zych	Łańcut
3	Łukasz	Duda	Jasionka
4	Wiktor	Czajka	Kolbuszowa

- **Lista sal, w których odbywają się zajęcia w dniach 23-25 stycznia 2024**

```
SELECT grupa_zajęciowa.data, sala.id_sali
FROM grupa_zajęciowa
LEFT JOIN sala
ON grupa_zajęciowa.sala_id_sali = sala.id_sali
WHERE grupa_zajęciowa.data BETWEEN TO_DATE('2024-01-23', 'YYYY-MM-DD')
AND TO_DATE('2024-01-25', 'YYYY-MM-DD');
```

	DATA	ID_SALI
1	23-JAN-24	2
2	24-JAN-24	10
3	25-JAN-24	5

- **Lista liczby sprzedanych karnetów dla każdego rodzaju**

```
SELECT karnet.rodzaj, COUNT(karnet.rodzaj) AS "Liczba sprzedanych"
FROM zamówienia
LEFT JOIN karnet
ON karnet.id_karnetu = zamówienia.karnet_id_karnetu
WHERE zamówienia.data_zamówienia BETWEEN TO_DATE('2023-01-01', 'YYYY-MM-DD')
AND TO_DATE('2023-12-31', 'YYYY-MM-DD')
GROUP BY karnet.rodzaj;
```

RODZAJ	Liczba sprzedanych
1 podstawowy	2
2 VIP	2
3 senior	1
4 student	2

- **Lista instruktorów posiadających specjalizację “Sztuki walki” posortowana malejąco względem zarobków**

```
SELECT instruktor.imię, instruktor.nazwisko, instruktor.stawka_za_godzinę,
specjalizacja.nazwa_specjalizacji
FROM instruktor
LEFT JOIN certyfikat
ON instruktor.id_instruktora = certyfikat.instruktor_id_instruktora
LEFT JOIN specjalizacja
ON certyfikat.specjalizacja_id_specjalizacji = specjalizacja.id_specjalizacji
WHERE specjalizacja.nazwa_specjalizacji = 'Sztuki walki'
ORDER BY instruktor.stawka_za_godzinę DESC;
```

IMIĘ	NAZWISKO	STAWKA_ZA_GODZINĘ	NAZWA_SPECJALIZACJI
1 Karolina	Wójcik	56	Sztuki walki
2 Jan	Nowak	50.5	Sztuki walki

- **Lista klientek urodzonych po roku 1994 posiadających karnet studencki**

```
SELECT klient.imię, klient.nazwisko, klient.płeć, klient.data_urodzenia, karnet.rodzaj
FROM klient
LEFT JOIN karnet
ON klient.karnet_id_karnetu = karnet.id_karnetu
WHERE karnet.rodzaj = 'student'
AND klient.płeć = 'Kobieta'
AND klient.data_urodzenia > TO_DATE('01-01-1994', 'DD-MM-YYYY');
```

	IMIĘ	NAZWISKO	PŁEĆ	DATA_URODZENIA	RODZAJ
1	Anna	Zielińska	Kobieta	18-SEP-95	student

- **Średnie zarobki instruktorów (zaokrąglone do całości), którzy posiadają specjalizację "Fitness"**

```
SELECT round(AVG(instruktor.stawka_za_godzinę)) AS "Średnie zarobki - fitness"
FROM instruktor
LEFT JOIN certyfikat
ON instruktor.id_instruktora = certyfikat.instruktor_id_instruktora
LEFT JOIN specjalizacja
ON certyfikat.specjalizacja_id_specjalizacji = specjalizacja.id_specjalizacji
WHERE specjalizacja.nazwa_specjalizacji = 'Fitness';
```

	Średnie zarobki - fitness
1	53

- **Lista faktur z zamówień na karnety VIP złożonych po 2. połowie 2023 roku oraz nazwisk osób, które złożyły te zamówienia**

```
SELECT zamówienia.data_zamowienia,
       zamówienia.nr_faktury, karnet.rodzaj, klient.nazwisko
FROM zamówienia
LEFT JOIN karnet
ON zamówienia.karnet_id_karnetu = karnet.id_karnetu
LEFT JOIN klient
ON zamówienia.klient_id_klienta = klient.id_klienta
WHERE karnet.rodzaj = 'VIP'
AND zamówienia.data_zamowienia >= TO_DATE('01-07-2023', 'DD-MM-YYYY');
```

DATA_ZAMOWIENIA	NR_FAKTURY	RODZAJ	NAZWISKO
1 15-SEP-23	P6Q7R8S9T0	VIP	Kita

- **Lista karnetów z cenami podstawowymi i cenami po obniżce o 20%**

```
SELECT rodzaj, cena, 0.8*cena AS "Cena po obnizce"
FROM karnet;
```

RODZAJ	CENA	Cena po obnizce
1 podstawowy	130	104
2 student	80	64
3 senior	100	80
4 VIP	180	144

- **Lista minimalnych, maksymalnych i średnich pojemności sal w każdej strefie**

```
SELECT sala.strefa,
       MIN(sala.pojemność) AS minimalna_pojemność,
       MAX(sala.pojemność) AS maksymalna_pojemność,
       ROUND(AVG(sala.pojemność)) AS średnia_pojemność
FROM sala
GROUP BY sala.strefa;
```

STREFA	MINIMALNA_POJEMNOŚĆ	MAKSYMALNA_POJEMNOŚĆ	ŚREDNIA_POJEMNOŚĆ
1 Fitness	28	30	29
2 Calm	25	32	29
3 Strong	15	20	18
4 Dance	20	40	30
5 Fight	5	10	8

- **Lista liczby grup dla każdego typu zajęć posortowana malejąco**

```
SELECT zajęcia.nazwa_zajęć, COUNT(grupa_zajęciowa.id_grupy) AS "Liczba grup"
FROM zajęcia
LEFT JOIN grupa_zajęciowa ON zajęcia.id_zajęć = grupa_zajęciowa.zajęcia_id_zajęć
GROUP BY nazwa_zajęć
ORDER BY "Liczba grup" DESC;
```

NAZWA_ZAJĘĆ	Liczba grup
1 Boks	2
2 Zumba	1
3 Joga	1
4 Kettlebell	1
5 Judo	1
6 Zdrowy kręgosłup	1
7 Tabata	1
8 Aerobik	1
9 Gimnastyka	0

Określenie kierunków rozwoju aplikacji

- System powiadomień - klienci otrzymywaliby powiadomienia w sytuacji odwołania zajęć, zmiany daty, godziny, sali lub instruktora.
- System oceny instruktorów - klienci mogliby oceniać pracę instruktorów i prowadzone przez nich zajęcia oraz dodawać komentarze.
- System zarządzania sprzętem - administratorzy mogliby zarządzać informacjami na temat zasobów sprzętowych posiadanych przez klub.
- System integracji z mediami społecznościowymi - administratorzy mogliby udostępniać informacje na temat zajęć i wydarzeń w klubie w powiązanych mediach społecznościowych.
- Program lojalnościowy - stali klienci otrzymywaliby zniżki lub inne korzystne oferty.
- System treningów personalnych – klienci dostaliby możliwość umówienia się na trening personalny z danym instruktorem.

Podsumowanie i wnioski

W trakcie realizacji projektu udało się zaplanować i zaimplementować bazę danych, przechowującą informacje zbierane przez klub fitness. Opracowany system usprawnia pracę klubu oraz pozwala na korzystanie klientom, instruktorom i administratorom. Do stworzenia schematu use case wykorzystano program UMLet, a do diagramu ERD i modelu relacyjnego – Oracle Data Modeler. Baza danych została zaimplementowana w języku SQL z wykorzystaniem SQL Developer.

Realizacja projektu pozwoliła nam zaznajomić z kolejnymi etapami projektowania i implementacji bazy danych.

Bibliografia

- Wykłady
- Instrukcje z laboratorium
- sqlpedia.pl