

# **Bazy Danych (Projekt)**

“Aplikacja dla optymalizacji diety”

## **Etap 2 - Projekt, implementacja i testy bazy danych**

Ivan Hancharyk 264511

Kinga Foksińska 255591

Piotr Walerianowicz 264027

# 1. Identyfikacja encji

Tabela 1 Identyfikacja encji wraz z identyfikatorami i atrybutami

Encja	Identyfikator	Atrybuty
Uzytkownik	id_uzytkownika	id_uzytkownika, imie, nazwisko, wiek, wzrost, waga, email, haslo, plec, id_planu
Preferencje	id_preferencji	id_preferencji, cel, alergie_wyjatki, ilosc_posilkow
Posilki	id_posilkow	id_posilku, nazwa_posilku, kcal_posilku, tluszcz_posilku, bialka_posilku, weglowodany_posilku, sol_posilku
Plan_posilkow	id_planu	id_planu, nazwa_planu
Skladniki	id_skladnika	id_skladnika, nazwa_skladnika, kcal, tluszcz, bialka, weglowodany, sol
Osiagniecia	id_osiagniecia	id_osiagniecia, ocena
Rodzaj_diety	id_rodzaju	id_rodzaju, nazwa_diety
Dane_logowania	id_logowania	id_logowania, data_logowania, id_uzytkownika

Tabela 2 Opis atrybutów encji

Encja	Atrybut	Opis
Uzytkownik	id_uzytkownika	Identyfikator użytkownika
	imie	Imię użytkownika
	nazwisko	Nazwisko użytkownika
	wiek	Wiek użytkownika
	wzrost	Wzrost użytkownika
	waga	Waga użytkownika
	email	E-mail użytkownika
	haslo	Hasło użytkownika
	plec	Płeć użytkownika
	id_planu	Identyfikator indywidualnego planu żywienia dla użytkownika

Preferencje	id_preferencji	Identyfikator preferencji
	cel	Nazwa celu diety (np. schudnięcie, utrzymanie wagi)
	alegie_wyjatki	Alergie użytkownika i wyjątki na konkretne składniki
	ilosc_posilkow	Ilość posiłków spożywanych przez użytkownika
Posilki	id_posilku	Identyfikator posiłku
	nazwa_posilku	Nazwa posiłku
	kcal_posilku	Ilość kalorii posiłku
	tluszcz_posilku	Ilość tłuszczu w posiłku (w gramach)
	bialka_posilku	Ilość białka w posiłku (w gramach)
	weglowodany_posilku	Ilość węglowodanów w posiłku (w gramach)
	sol_posilku	Ilość soli w posiłku (w gramach)
Plan_posilkow	id_planu	Identyfikator planu posiłków
	nazwa_planu	Nazwa planu posiłków
Składniki	id_skladnika	Identyfikator składnika
	nazwa_skladnika	Nazwa składnika
	kcal	Ilość kalorii w składniku
	tluszcz	Ilość tłuszczu w składniku (w gramach)
	bialka	Ilość białka w składniku
	weglowodany	Ilość węglowodanów w składniku (w gramach)
	sol	Ilość soli w składniku (w gramach)
Osiagniecia	id_osiagniecia	Identyfikator osiągnięcia
	ocena	Nazwa osiągnięcia
Rodzaj_diety	id_rodzaju	Identyfikator rodzaju diety
	nazwa_diety	Nazwa diety
Dane_logowania	id_logowania	Identyfikator pojedynczego logowania
	data_logowania	Data logowania
	id_uzytkownika	Identyfikator użytkownika, który się zalogował

## 2. Sformułowanie wymagań

Należy precyzyjnie zdefiniować uprawnienia dostępu dla jedynego użytkownika naszej aplikacji do monitorowania diety. Możliwe działania, jakie rozpatrujemy w kontekście sformułowania wymagań to wyświetlanie, modyfikowania, dodawania lub usuwanie rekordów w tabelach.

*Tabela 3 Wymagania dostępu do tabel dla użytkownika*

	Uzytkownik	Osiagniecia	Preferencje	Plan_posilkow	Posilki	Skladniki	Rodzaj_diety	Dane_logowania
wyświetlenie	x	x	x	x	x	x	x	
modyfikowanie	x		x	x			x	
dodawanie			x		x	x		
usuwanie			x					

## 3. Diagram konceptualny (postać 3 normalna)

Trzecia postać normalna (3NF) charakteryzuje się tym, że kolumna informacyjna nie należąca do klucza nie zależy też od innej kolumny informacyjnej, nie należącej do klucza. Czyli każdy niekluczowy argument jest bezpośrednio zależny tylko od klucza głównego, a nie od innej kolumny.

*Tabela 4 Posilki 3NF*

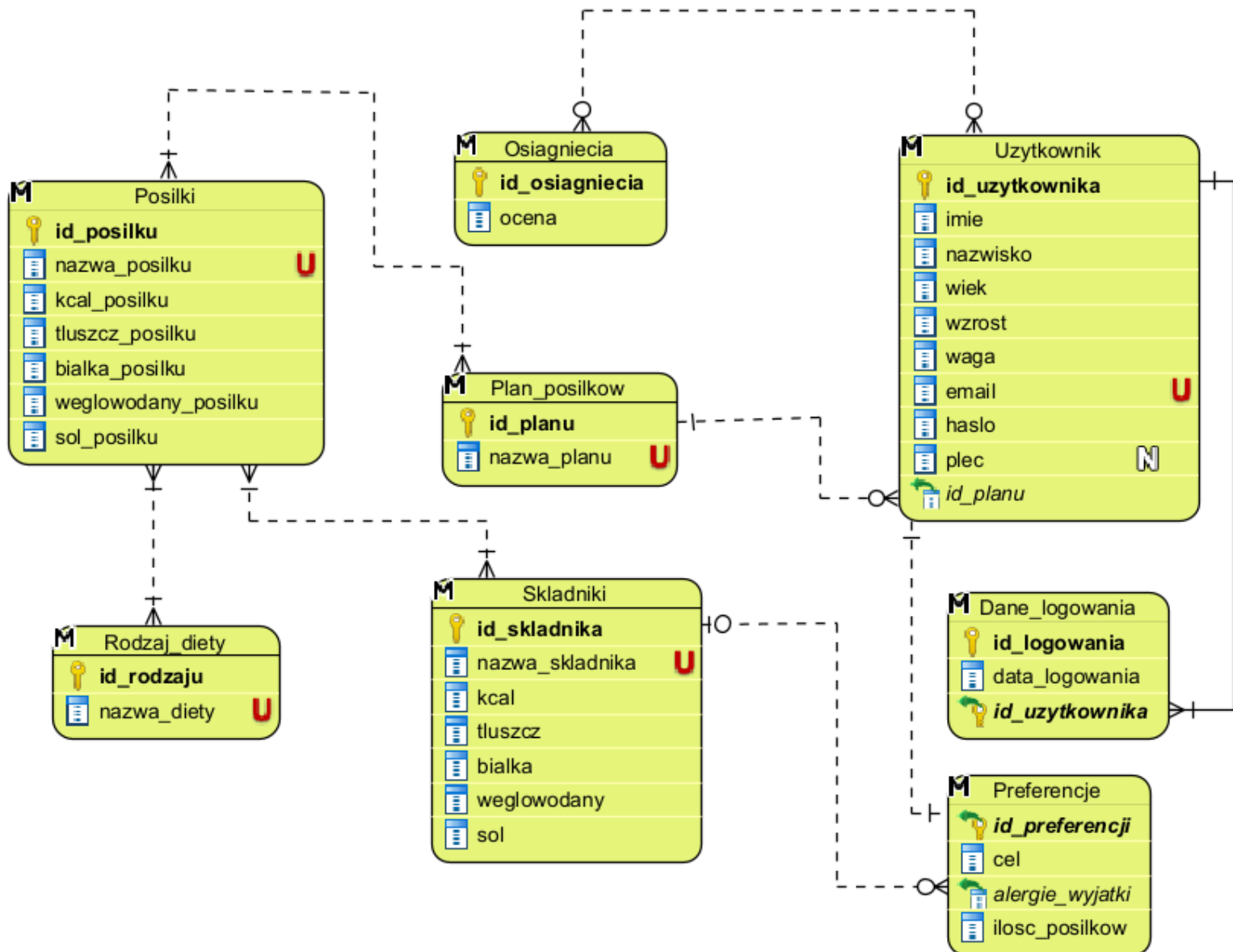
ID_POSILKU	NAZWA_POSILKU	KCAL_POSILKU	TLUSZCZ_POSILKU	BIALKO_POSILKU	WEGLOWODANY_POSILKU	SOL_POSILKU
1	jajecznic	168	11.2	10.9	5.9	1.02
2	serniki	340	20.2	17.4	25.7	1.22
3	carbonara	615	17.3	23	80	2.2
4	barszcz	68	12.2	2.9	13.9	1.02
5	cezar	450	20	20	30	1
6	kotlet z ryby	150	5	25	10	2

*Tabela 5 Składniki 3NF*

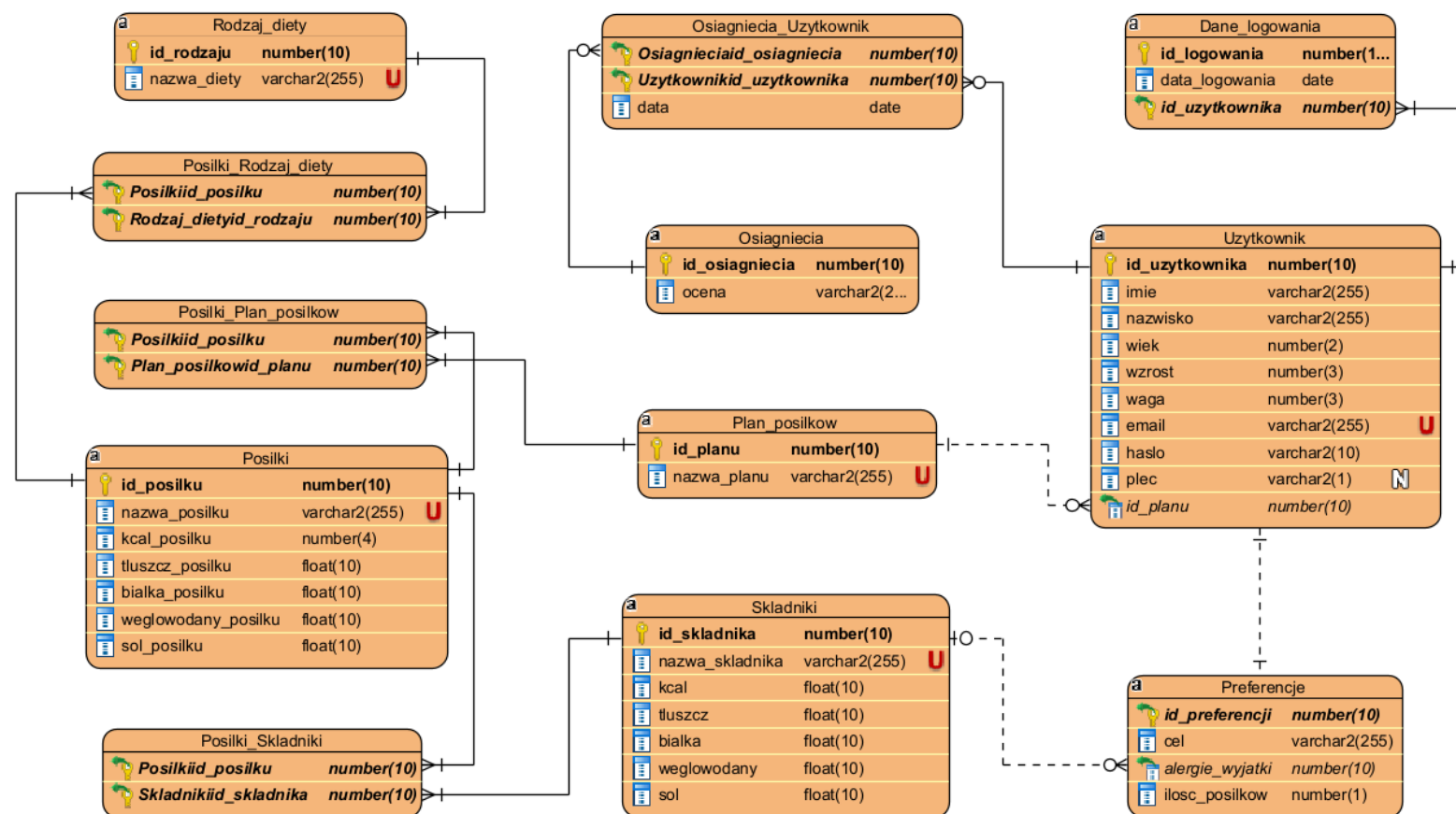
ID_SKLADNIKA	NAZWA_SKLADNIKA	KCAL	TLUSZCZ	BIALKA	WEGLOWODANY	SOL
21	maslo	717	81	.9	.1	.5
1	pomidor	18	.2	.9	3.9	.02
2	salata	13	.2	1.4	2.2	.02
3	bulka	265	2.3	8	50	1.2
4	jajko	143	9.3	12.6	1.1	.4
5	twarog	77	5	11	3.5	.2
6	smietana	205	20	2	3	.1
7	ser parmezan	392	29	38.9	4.1	1.6
8	boczek	471	42	12	.1	2
9	makaron	131	1.3	3.5	25	0
10	burak	43	.2	1.6	9.6	.1
11	czosnek	149	.5	6.4	33.1	0
12	piers z kurczaka	165	1	31	0	.1
13	mleko	61	3.6	3.3	4.8	.1

Są to tabele spełniające wszystkie 3 postaci normalne. Dzięki temu, wszystkie dane znajdujące się w każdej z tabel są zależne tylko od klucza głównego.

Diagram konceptualny bazy danych 3NF



## 4. Diagram fizyczny



## 5. Wdrożenie bazy

```

-- TWORZENIE TABEL
-- Zdefiniowanie typów kolumn zgodnie z wymaganiami serwera.
-- Ustawienie odpowiednich kolumn jako unikalne i/lub opcjonalne.
-- Zdefiniowanie kluczy głównych i obcych dla powiązania tabel.
CREATE TABLE Dane_logowania (
    id_logowania  number(10) NOT NULL,
    data_logowania date NOT NULL,
    id_uzytkownika number(10) NOT NULL,
    PRIMARY KEY (id_logowania, id_uzytkownika));

CREATE TABLE Osiagniecia (
    id_osiagniecia number(10) GENERATED AS IDENTITY,
    ocena          varchar2(255) NOT NULL,
    PRIMARY KEY (id_osiagniecia));

CREATE TABLE Osiagniecia_Uzytkownik (

```

```

    Osiagnieciaid_osiagniecia number(10) NOT NULL,
    Uzytkownikid_uzytkownika number(10) NOT NULL,
    data date NOT NULL,
    PRIMARY KEY (Osiagnieciaid_osiagniecia,
    Uzytkownikid_uzytkownika));

CREATE TABLE Plan_posilkow (
    id_planu number(10) GENERATED AS IDENTITY,
    nazwa_planu varchar2(255) NOT NULL UNIQUE,
    PRIMARY KEY (id_planu));

CREATE TABLE Posilki (
    id_posilku number(10) GENERATED AS IDENTITY,
    nazwa_posilku varchar2(255) NOT NULL UNIQUE,
    kcal_posilku number(4) NOT NULL,
    tluszcz_posilku float(10) NOT NULL,
    bialka_posilku float(10) NOT NULL,
    weglowodany_posilku float(10) NOT NULL,
    sol_posilku float(10) NOT NULL,
    PRIMARY KEY (id_posilku));

CREATE TABLE Posilki_Plan_posilkow (
    Posilkiid_posilku number(10) NOT NULL,
    Plan_posilkowid_planu number(10) NOT NULL,
    PRIMARY KEY (Posilkiid_posilku,
    Plan_posilkowid_planu));

CREATE TABLE Posilki_Rodzaj_diety (
    Posilkiid_posilku number(10) NOT NULL,
    Rodzaj_dietyid_rodzaju number(10) NOT NULL,
    PRIMARY KEY (Posilkiid_posilku,
    Rodzaj_dietyid_rodzaju));

CREATE TABLE Posilki_Skladniki (
    Posilkiid_posilku number(10) NOT NULL,
    Skladnikiid_skladnika number(10) NOT NULL,
    PRIMARY KEY (Posilkiid_posilku,
    Skladnikiid_skladnika));

CREATE TABLE Preferencje (
    id_preferencji number(10) NOT NULL,
    cel varchar2(255) NOT NULL,
    alergie_wyjatki number(10) NOT NULL,
    ilosc_posilkow number(1) DEFAULT 5 NOT NULL,
    PRIMARY KEY (id_preferencji));

CREATE TABLE Rodzaj_diety (
    id_rodzaju number(10) GENERATED AS IDENTITY,

```



```
nazwa_diety varchar2(255) NOT NULL UNIQUE,  
PRIMARY KEY (id_rodzaju));
```

```
CREATE TABLE Skladniki (  
  id_skladnika    number(10) GENERATED AS IDENTITY,  
  nazwa_skladnika varchar2(255) NOT NULL UNIQUE,  
  kcal            float(10) NOT NULL,  
  tluszcz        float(10) NOT NULL,  
  bialka         float(10) NOT NULL,  
  weglowodany    float(10) NOT NULL,  
  sol            float(10) NOT NULL,  
  PRIMARY KEY (id_skladnika));
```

```
CREATE TABLE Uzytkownik (  
  id_uzytkownika number(10) GENERATED AS IDENTITY,  
  imie           varchar2(255) NOT NULL,  
  nazwisko       varchar2(255) NOT NULL,  
  wiek           number(2) NOT NULL CHECK (wiek BETWEEN 18 AND 99),  
  wzrost         number(3) NOT NULL,  
  waga           number(3) NOT NULL,  
  email          varchar2(255) NOT NULL UNIQUE,  
  haslo          varchar2(10) NOT NULL,  
  plec           varchar2(1),  
  id_planu       number(10) NOT NULL,  
  PRIMARY KEY (id_uzytkownika));
```

```
ALTER TABLE Preferencje ADD CONSTRAINT FKPreferencj735605 FOREIGN KEY  
(alergie_wyjatki) REFERENCES Skladniki (id_skladnika);  
ALTER TABLE Uzytkownik ADD CONSTRAINT FKUzytkownik939445 FOREIGN KEY  
(id_planu) REFERENCES Plan_posilkow (id_planu);  
ALTER TABLE Dane_logowania ADD CONSTRAINT FKDane_logow532698 FOREIGN KEY  
(id_uzytkownika) REFERENCES Uzytkownik (id_uzytkownika);  
ALTER TABLE Posilki_Rodzaj_diety ADD CONSTRAINT FKPosilki_Ro433362 FOREIGN  
KEY (Posilkiid_posilku) REFERENCES Posilki (id_posilku);  
ALTER TABLE Posilki_Rodzaj_diety ADD CONSTRAINT FKPosilki_Ro143797 FOREIGN  
KEY (Rodzaj_dietyid_rodzaju) REFERENCES Rodzaj_diety (id_rodzaju);  
ALTER TABLE Osiagniecia_Uzytkownik ADD CONSTRAINT FK0siagnieci428199 FOREIGN  
KEY (Osiagnieciaid_osiagniecia) REFERENCES Osiagniecia (id_osiagniecia);  
ALTER TABLE Posilki_Skladniki ADD CONSTRAINT FKPosilki_Sk910290 FOREIGN KEY  
(Posilkiid_posilku) REFERENCES Posilki (id_posilku);  
ALTER TABLE Posilki_Skladniki ADD CONSTRAINT FKPosilki_Sk516588 FOREIGN KEY  
(Skladnikiid_skladnika) REFERENCES Skladniki (id_skladnika);  
ALTER TABLE Posilki_Plan_posilkow ADD CONSTRAINT FKPosilki_P1337303 FOREIGN  
KEY (Posilkiid_posilku) REFERENCES Posilki (id_posilku);  
ALTER TABLE Posilki_Plan_posilkow ADD CONSTRAINT FKPosilki_P1675965 FOREIGN  
KEY (Plan_posilkowid_planu) REFERENCES Plan_posilkow (id_planu);  
ALTER TABLE Osiagniecia_Uzytkownik ADD CONSTRAINT FK0siagnieci511642 FOREIGN  
KEY (Uzytkownikid_uzytkownika) REFERENCES Uzytkownik (id_uzytkownika);
```

```

ALTER TABLE Preferencje ADD CONSTRAINT FKPreferencj244422 FOREIGN KEY
(id_preferencji) REFERENCES Uzytkownik (id_uzytkownika);

-- Zdefiniowanie użytkowników bazy danych i nadanie im odpowiednich
uprawnień.
-- Tworzenie użytkownika
CREATE USER klient IDENTIFIED BY haslo;

-- Nadawanie uprawnień dla tabeli "Uzytkownik"
GRANT SELECT, UPDATE ON Uzytkownik TO klient;

-- Nadawanie uprawnień dla tabeli "Osiagniecia"
GRANT SELECT ON Osiagniecia TO klient;

-- Nadawanie uprawnień dla tabeli "Preferencje"
GRANT SELECT, INSERT, UPDATE, DELETE ON Preferencje TO klient;

-- Nadawanie uprawnień dla tabeli "Plan_posilkow"
GRANT SELECT, UPDATE ON Plan_posilkow TO klient;

-- Nadawanie uprawnień dla tabeli "Posilki"
GRANT SELECT, INSERT ON Posilki TO klient;

-- Nadawanie uprawnień dla tabeli "Skladniki"
GRANT SELECT, INSERT ON Skladniki TO klient;

-- Nadawanie uprawnień dla tabeli "Rodzaj diety"
GRANT SELECT, UPDATE ON Rodzaj_diety TO klient;

-- Odjęcie wszystkich uprawnień dla tabeli "Dane_logowania"
REVOKE ALL ON Dane_logowania FROM klient;

-- WYZWALACZE

-- Wyzwalacz zwiększający wartosci posilku (kcal, białko, sól, eglowodany i
tłuszcze), po dodaniu mu kolejnego składnika
CREATE OR REPLACE TRIGGER zwieksz_wartosci_po_dodaniu_skladnika
AFTER INSERT ON Posilki_Skladniki
FOR EACH ROW
DECLARE
    v_kcal_skladnika Skladniki.kcal%TYPE;
    v_bialko_skladnika Skladniki.bialka%TYPE;
    v_sol_skladnika Skladniki.sol%TYPE;
    v_weglowodany_skladnika Skladniki.weglowodany%TYPE;
    v_tluszcz_skladnika Skladniki.tluszcz%TYPE;
BEGIN
    -- Pobranie wartości składnika dodanego do posiłku

```

```

        SELECT kcal, bialka, sol, weglowodany, tluszcz
        INTO v_kcal_skladnika, v_bialko_skladnika, v_sol_skladnika,
v_weglowodany_skladnika, v_tluszcz_skladnika
        FROM Skladniki
        WHERE id_skladnika = :NEW.Skladnikiid_skladnika;

-- Aktualizacja wartości składników w Posilki
UPDATE Posilki
SET kcal_posilku = kcal_posilku + v_kcal_skladnika,
    bialka_posilku = bialka_posilku + v_bialko_skladnika,
    sol_posilku = sol_posilku + v_sol_skladnika,
    weglowodany_posilku = weglowodany_posilku + v_weglowodany_skladnika,
    tluszcz_posilku = tluszcz_posilku + v_tluszcz_skladnika
WHERE id_posilku = :NEW.Posilkiid_posilku;
END;
/

-- Wyzwalacz zmieniający wartości w posiłkach do których należy składnik po
zmianie wartosci tego skladnika
CREATE OR REPLACE TRIGGER
aktualizuj_wartosci_w_posilkach_po_zmianie_skladnika
AFTER UPDATE OF kcal, bialka, tluszcz, weglowodany, sol ON Skladniki
FOR EACH ROW
BEGIN
    UPDATE Posilki P
    SET P.kcal_posilku = P.kcal_posilku + (:NEW.kcal - :OLD.kcal),
        P.bialka_posilku = P.bialka_posilku + (:NEW.bialka - :OLD.bialka),
        P.tluszcz_posilku = P.tluszcz_posilku + (:NEW.tluszcz -
:OLD.tluszcz),
        P.weglowodany_posilku = P.weglowodany_posilku + (:NEW.weglowodany -
:OLD.weglowodany),
        P.sol_posilku = P.sol_posilku + (:NEW.sol - :OLD.sol)
    WHERE P.id_posilku IN (
        SELECT PS.Posilkiid_posilku
        FROM Posilki_Skladniki PS
        WHERE PS.Skladnikiid_skladnika = :NEW.id_skladnika
    );
END;
/

-- Wyzwalacz dodający dane do tabeli Dane_logowania po każdym zalogowaniu
się klienta
CREATE OR REPLACE TRIGGER dodaj_logowanie_po_zalogowaniu
AFTER LOGON ON klient.schema
BEGIN
    INSERT INTO Dane_logowania (id_logowania, data_logowania,
id_uzytkownika)
    VALUES (seq_dane_logowania.NEXTVAL, SYSDATE, (SELECT id_uzytkownika FROM

```

```

Uzytkownik WHERE username = USER));
END;
/
-- Wyzwalacze odpowiedzialne za usuwanie danych z tabeli łączących
(zastosowanych w relacjach wiele do wielu), jeśli któryś z rekordów został
usunięty
CREATE OR REPLACE TRIGGER usun_powiazane_dane_po_usunieciu_posilku
AFTER DELETE ON Posilki
FOR EACH ROW
BEGIN
    -- Usunięcie rekordów z Posilki_Rodzaj_diety dla usuwanego posilku
    DELETE FROM Posilki_Rodzaj_diety
    WHERE Posilkiid_posilku = :OLD.id_posilku;

    -- Usunięcie rekordów z Posilki_Skladniki dla usuwanego posilku
    DELETE FROM Posilki_Skladniki
    WHERE Posilkiid_posilku = :OLD.id_posilku;

    -- Usunięcie rekordów z Posilki_Plan_posilkow dla usuwanego posilku
    DELETE FROM Posilki_Plan_posilkow
    WHERE Posilkiid_posilku = :OLD.id_posilku;
END;
/

CREATE OR REPLACE TRIGGER usun_powiazane_dane_po_usunieciu_skladnika
BEFORE DELETE ON Skladniki
FOR EACH ROW
BEGIN
    -- Usunięcie rekordów z Posilki_Skladniki dla usuwanego skladnika
    DELETE FROM Posilki_Skladniki
    WHERE Skladnikiid_skladnika = :OLD.id_skladnika;
END;
/

CREATE OR REPLACE TRIGGER usun_powiazane_dane_po_usunieciu_uzytkownika
AFTER DELETE ON Uzytkownik
FOR EACH ROW
BEGIN
    -- Usunięcie rekordów z Osiagniecia_Uzytkownik dla usuwanego uzytkownika
    DELETE FROM Osiagniecia_Uzytkownik
    WHERE Uzytkownikid_uzytkownika = :OLD.id_uzytkownika;

    -- Usunięcie preferencji dla usuwanego uzytkownika
    DELETE FROM Preferencje
    WHERE id_preferencji = :OLD.id_uzytkownika;
END;
/
-- Funkcja sprawdzająca, czy posiłek spełnia kryteria diety

```

```

CREATE OR REPLACE FUNCTION CzyPosilekSpelniaKryteriaDiety (p_id_posilku
NUMBER, p_id_rodzaju_diety NUMBER) RETURN BOOLEAN
AS
    v_liczba_posilkow NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_liczba_posilkow
    FROM Posilki_Rodzaj_diety
    WHERE Posilkiid_posilku = p_id_posilku AND Rodzaj_dietyid_rodzaju =
p_id_rodzaju_diety;

    RETURN v_liczba_posilkow > 0;
END;
/

-- Funkcja zwracająca liczbę osiągnięć danego użytkownika
CREATE OR REPLACE FUNCTION
PobierzLiczbeOsiagniecUzytkownika(p_id_uzytkownika NUMBER) RETURN NUMBER
AS
    v_liczba_osiagniec NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_liczba_osiagniec
    FROM Osiagniecia_Uzytkownik
    WHERE Uzytkownikid_uzytkownika = p_id_uzytkownika;

    RETURN v_liczba_osiagniec;
END;
/

-- Funkcja zwracająca liczbę składników w danym posiłku
CREATE OR REPLACE FUNCTION PobierzLiczbeSkladnikowWPosilku(p_id_posilku
NUMBER) RETURN NUMBER
AS
    v_liczba_skladnikow NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_liczba_skladnikow
    FROM Posilki_Skladniki
    WHERE Posilkiid_posilku = p_id_posilku;

    RETURN v_liczba_skladnikow;
END;
/

-- Procedura aktualizująca preferencje danego użytkownika
CREATE OR REPLACE PROCEDURE AktualizujPreferencjeUzytkownika(
    p_id_uzytkownika NUMBER,

```

```

    p_cel VARCHAR2,
    p_alergie_wyjatki NUMBER,
    p_ilosc_posilkow NUMBER
)
AS
BEGIN
    UPDATE Preferencje
    SET cel = p_cel,
        alergie_wyjatki = p_alergie_wyjatki,
        ilosc_posilkow = p_ilosc_posilkow
    WHERE id_preferencji = p_id_uzytkownika;

    COMMIT;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Błąd podczas aktualizacji preferencji.');
```

## 6. Przetestowanie bazy

```

-- wprowadzenie danych dla tabeli Osiagniecia
INSERT INTO Osiagniecia (ocena)
VALUES ('Tydzien bez slodyczy!');

INSERT INTO Osiagniecia (ocena)
VALUES ('Strata 2 kilogramow w 30 dni!');

INSERT INTO Osiagniecia (ocena)
VALUES ('Jedzenie 5 porcji warzyw dziennie przez tydzien!');

INSERT INTO Osiagniecia (ocena)
VALUES ('Trzymasz się planu posilkow od miesiaca!');

INSERT INTO Osiagniecia (ocena)
VALUES ('Redukcja spozycia cukru o 50% przez miesiac!');

-- wprowadzenie danych dla tabeli Plan_posilkow
INSERT INTO Plan_posilkow (nazwa_planu)
VALUES ('Plan wegetrianski 1600 kcal');

INSERT INTO Plan_posilkow (nazwa_planu)
VALUES ('Plan wegetariański 2000 kcal');
```

```
INSERT INTO Plan_posilkow (nazwa_planu)
VALUES ('Plan paleo 1800 kcal');

INSERT INTO Plan_posilkow (nazwa_planu)
VALUES ('Plan redukcyjny 1500 kcal');

INSERT INTO Plan_posilkow (nazwa_planu)
VALUES ('Plan ketogeniczny 2200 kcal');

-- wprowadzenie danych dla tabeli Rodzaj_diety
INSERT INTO Rodzaj_diety (nazwa_diety)
VALUES ('Wegetariańska');

INSERT INTO Rodzaj_diety (nazwa_diety)
VALUES ('Wegańska');

INSERT INTO Rodzaj_diety (nazwa_diety)
VALUES ('Mięsna');

INSERT INTO Rodzaj_diety (nazwa_diety)
VALUES ('Paleo');

INSERT INTO Rodzaj_diety (nazwa_diety)
VALUES ('Wysokokaloryczna');

-- wprowadzenie danych dla tabeli Posilki
INSERT INTO Posilki (nazwa_posilku, kcal_posilku, tluszcz_posilku,
bialka_posilku, weglowodany_posilku, sol_posilku)
VALUES ('jajecznicza', 150, 11, 10, 2, 1);

INSERT INTO Posilki (nazwa_posilku, kcal_posilku, tluszcz_posilku,
bialka_posilku, weglowodany_posilku, sol_posilku)
VALUES ('serniki', 250, 15, 5, 20, 1);

INSERT INTO Posilki (nazwa_posilku, kcal_posilku, tluszcz_posilku,
bialka_posilku, weglowodany_posilku, sol_posilku)
VALUES ('carbonara', 350, 15, 15, 30, 1);

INSERT INTO Posilki (nazwa_posilku, kcal_posilku, tluszcz_posilku,
bialka_posilku, weglowodany_posilku, sol_posilku)
VALUES ('barszcz', 50, 12, 2, 10, 1);

INSERT INTO Posilki (nazwa_posilku, kcal_posilku, tluszcz_posilku,
bialka_posilku, weglowodany_posilku, sol_posilku)
VALUES ('cezar', 450, 20, 20, 30, 1);
```

```
INSERT INTO Posilki (nazwa_posilku, kcal_posilku, tluszcz_posilku,
bialka_posilku, weglowodany_posilku, sol_posilku)
VALUES ('kotlet z ryby', 150, 5, 25, 10, 2);

-- wprowadzenie danych dla tabeli Skladniki
INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('pomidor', 18, 0.2, 0.9, 3.9, 0.02);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('salata', 13, 0.2, 1.4, 2.2, 0.02);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('bulka', 265, 2.3, 8, 50, 1.2);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('jajko', 143, 9.3, 12.6, 1.1, 0.4);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('maslo', 717, 81, 0.9, 0.1, 0.5);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('twarog', 77, 5, 11, 3.5, 0.2);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('smietana', 205, 20, 2, 3, 0.1);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('ser parmezan', 392, 29, 38.9, 4.1, 1.6);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('boczek', 471, 42, 12, 0.1, 2);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('makaron', 131, 1.3, 3.5, 25, 0);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
```



```
weglowodany, sol)
VALUES ('burak', 43, 0.2, 1.6, 9.6, 0.1);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('czosnek', 149, 0.5, 6.4, 33.1, 0);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('piers z kurczaka', 165, 1, 31, 0, 0.1);

INSERT INTO Skladniki (nazwa_skladnika, kcal, tluszcz, bialka,
weglowodany, sol)
VALUES ('mleko', 61, 3.6, 3.3, 4.8, 0.1);

-- wprowadzenie danych dla tabeli Uzytkownik
INSERT INTO Uzytkownik (imie, nazwisko, wiek, wzrost, waga, email,
haslo, id_planu)
VALUES ('Jan', 'Kowalski', 30, 180, 80, 'jan.kowalski@gmail.com',
'haslo123', 1);

INSERT INTO Uzytkownik (imie, nazwisko, wiek, wzrost, waga, email,
haslo, plec, id_planu)
VALUES ('Anna', 'Nowak', 25, 165, 60, 'anna.nowak@gmail.com',
'password12', 'K', 2);

INSERT INTO Uzytkownik (imie, nazwisko, wiek, wzrost, waga, email,
haslo, plec, id_planu)
VALUES ('Piotr', 'Wiśniewski', 35, 175, 70,
'piotr.wisniewski@gmail.com', 'securepass', 'M', 3);

INSERT INTO Uzytkownik (imie, nazwisko, wiek, wzrost, waga, email,
haslo, plec, id_planu)
VALUES ('Ewa', 'Kowalczyk', 28, 160, 55, 'ewa.kowalczyk@gmail.com',
'qwerty123', 'K', 4);

INSERT INTO Uzytkownik (imie, nazwisko, wiek, wzrost, waga, email,
haslo, plec, id_planu)
VALUES ('Krzysztof', 'Lis', 32, 185, 90, 'krzysztof.lis@gmail.com',
'letmein123', 'M', 5);

-- wprowadzenie danych dla tabeli Preferencje
INSERT INTO Preferencje (id_preferencji, cel, alergije_wyjatki,
ilosc_posilkow)
VALUES (1, 'Utrata wagi', 1, 3);
```

```
INSERT INTO Preferencje (id_preferencji, cel, alergie_wyjatki,
ilosc_posilkow)
VALUES (2, 'Zwiększenie masy mięśniowej', 1, 5);

INSERT INTO Preferencje (id_preferencji, cel, alergie_wyjatki,
ilosc_posilkow)
VALUES (3, 'Poprawa kondycji fizycznej', 3, 4);

INSERT INTO Preferencje (id_preferencji, cel, alergie_wyjatki,
ilosc_posilkow)
VALUES (4, 'Redukcja stresu', 2, 3);

INSERT INTO Preferencje (id_preferencji, cel, alergie_wyjatki,
ilosc_posilkow)
VALUES (5, 'Poprawa odporności', 2, 4);

-- wprowadzenie danych dla tabeli Osiagniecia_Uzytkownik
INSERT INTO Osiagniecia_Uzytkownik (Osiagnieciaid_osiagniecia,
Uzytkownikid_uzytkownika, data)
VALUES (1, 1, TO_DATE('2023-01-01', 'YYYY-MM-DD'));

INSERT INTO Osiagniecia_Uzytkownik (Osiagnieciaid_osiagniecia,
Uzytkownikid_uzytkownika, data)
VALUES (2, 2, TO_DATE('2023-02-15', 'YYYY-MM-DD'));

INSERT INTO Osiagniecia_Uzytkownik (Osiagnieciaid_osiagniecia,
Uzytkownikid_uzytkownika, data)
VALUES (3, 1, TO_DATE('2023-03-21', 'YYYY-MM-DD'));

INSERT INTO Osiagniecia_Uzytkownik (Osiagnieciaid_osiagniecia,
Uzytkownikid_uzytkownika, data)
VALUES (2, 3, TO_DATE('2023-04-05', 'YYYY-MM-DD'));

INSERT INTO Osiagniecia_Uzytkownik (Osiagnieciaid_osiagniecia,
Uzytkownikid_uzytkownika, data)
VALUES (5, 2, TO_DATE('2023-05-12', 'YYYY-MM-DD'));

-- wprowadzenie danych dla tabeli Posilki_Plan_posilkow
INSERT INTO Posilki_Plan_posilkow (Posilkiid_posilku,
Plan_posilkowid_planu)
VALUES (1, 1);

INSERT INTO Posilki_Plan_posilkow (Posilkiid_posilku,
Plan_posilkowid_planu)
VALUES (2, 2);
```

```
INSERT INTO Posilki_Plan_posilkow (Posilkiid_posilku,  
Plan_posilkowid_planu)  
VALUES (3, 2);  
  
INSERT INTO Posilki_Plan_posilkow (Posilkiid_posilku,  
Plan_posilkowid_planu)  
VALUES (4, 1);  
  
INSERT INTO Posilki_Plan_posilkow (Posilkiid_posilku,  
Plan_posilkowid_planu)  
VALUES (2, 3);  
  
-- wprowadzenie danych dla tabeli Posilki_Rodzaj_diety  
INSERT INTO Posilki_Rodzaj_diety (Posilkiid_posilku,  
Rodzaj_dietyid_rodzaju)  
VALUES (1, 1);  
  
INSERT INTO Posilki_Rodzaj_diety (Posilkiid_posilku,  
Rodzaj_dietyid_rodzaju)  
VALUES (2, 2);  
  
INSERT INTO Posilki_Rodzaj_diety (Posilkiid_posilku,  
Rodzaj_dietyid_rodzaju)  
VALUES (3, 3);  
  
INSERT INTO Posilki_Rodzaj_diety (Posilkiid_posilku,  
Rodzaj_dietyid_rodzaju)  
VALUES (4, 1);  
  
INSERT INTO Posilki_Rodzaj_diety (Posilkiid_posilku,  
Rodzaj_dietyid_rodzaju)  
VALUES (2, 5);  
  
-- wprowadzenie danych dla tabeli Posilki_Skladniki  
INSERT INTO Posilki_Skladniki (Posilkiid_posilku, Skladnikiid_skladnika)  
VALUES (1, 1);  
  
INSERT INTO Posilki_Skladniki (Posilkiid_posilku, Skladnikiid_skladnika)  
VALUES (2, 2);  
  
INSERT INTO Posilki_Skladniki (Posilkiid_posilku, Skladnikiid_skladnika)  
VALUES (3, 3);
```

```
INSERT INTO Posilki_Skladniki (Posilkiid_posilku, Skladnikiid_skladnika)
VALUES (4, 1);
```

```
INSERT INTO Posilki_Skladniki (Posilkiid_posilku, Skladnikiid_skladnika)
VALUES (2, 5);
```

Wynik testu - sukcesywne wprowadzanie wszystkich danych do tabel:

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

```
-- Przykład próby naruszenia unikalności kolumny 'email' w tabeli
'Uzytkownik'
```

```
INSERT INTO Uzytkownik (imie, nazwisko, wiek, wzrost, waga, email,
haslo, plec, id_planu)
```

```
VALUES ('Nowy', 'Uzytkownik', 25, 175, 70, 'anna.nowak@gmail.com',
'haslo124', 'M', 6);
```

## Wynik:

ORA-00001: unique constraint (SQL\_QVCFDOBBEKLRLVIFYCYITPMH.SYS\_C00142665357) violated ORA-06512: at "SYS.DBMS\_SQL", line 1721

More Details: <https://docs.oracle.com/error-help/db/ora-00001>

```
-- Przykład próby naruszenia warunku CHECK w tabeli 'Uzytkownik'(  
wartość powinna być między 18 a 99)  
INSERT INTO Uzytkownik (imie, nazwisko, wiek, wzrost, waga, email,  
haslo, plec, id_planu)  
VALUES ('Naruszajacy', 'Warunki', 15, 160, 55, 'nowy@uzytkownik.pl',  
'haslo123', 'K', 1);
```

## Wynik:

ORA-02290: check constraint (SQL\_QVCFDOBBEKLRLVIFYCYITPMH.SYS\_C00142665355) violated ORA-06512: at "SYS.DBMS\_SQL", line 1721

More Details: <https://docs.oracle.com/error-help/db/ora-02290>

```
-- Przykład próby naruszenia unikalności kombinacji kluczy obcych w  
tabeli 'Posilki_Plan_posilkow'  
INSERT INTO Posilki_Plan_posilkow (Posilkiid_posilku,  
Plan_posilkowid_planu)  
VALUES (1, 1);  
  
-- Druga próba dodania tego samego wpisu  
INSERT INTO Posilki_Plan_posilkow (Posilkiid_posilku,  
Plan_posilkowid_planu)  
VALUES (1, 1);
```

## Wynik:

ORA-00001: unique constraint (SQL\_QVCFDOBBEKLRLVIFYCYITPMH.SYS\_C00142666828) violated ORA-06512: at "SYS.DBMS\_SQL", line 1721

More Details: <https://docs.oracle.com/error-help/db/ora-00001>

```
-- Dodanie składnika bez podania wymaganego pola(brakuje pola  
'nazwa_skladnika')  
INSERT INTO Skladniki (kcal, tluszcz, bialka, weglowodany, sol)  
VALUES (30, 0.2, 1.5, 6.3, 0.1);
```

## Wynik:

ORA-01400: cannot insert NULL into ("SQL\_XHDOLCWGBVTMPCQYZKPYXFULJ"."SKLADNIKI"."NAZWA\_SKLADNIKA") ORA-06512: at "SYS.DBMS\_SQL", line 1721

More Details: <https://docs.oracle.com/error-help/db/ora-01400>

```
--Wyswietlenie danych z tablicy uzytkownik  
SELECT * FROM Uzytkownik;
```

Wynik:

ID_UZYTKOWNIKA	IMIE	NAZWISKO	WIEK	WZROST	WAGA	EMAIL	HASLO	PLEC	ID_PLANU
1	Jan	Kowalski	30	180	80	jan.kowalski@gmail.com	haslo123	-	1
2	Anna	Nowak	25	165	60	anna.nowak@gmail.com	password12	K	2
3	Piotr	Wiśniewski	35	175	70	piotr.wisniewski@gmail.com	securepass	M	3
4	Ewa	Kowalczyk	28	160	55	ewa.kowalczyk@gmail.com	qwerty123	K	4
5	Krzysztof	Lis	32	185	90	krzysztof.lis@gmail.com	letmein123	M	5

```
--Wyswietlenie czesci danych z tablicy Posilki  
SELECT id_posilku, nazwa_posilku, kcal_posilku,  
weglowodany_posilku FROM Posilki;
```

Wynik:

ID_POSILKU	NAZWA_POSILKU	KCAL_POSILKU	WEGLOWODANY_POSILKU
1	jajecznicza	168	5.9
2	serniki	980	22.3
3	carbonara	615	80
4	barszcz	68	13.9
5	cezar	450	30
6	kotlet z ryby	150	10

```
--Aktualizacja danych uzytkownika
```

```
UPDATE Uzytkownik
SET imie = 'NoweImie'
WHERE id_uzytkownika = 1;
```

Wynik:

ID_UZYTKOWNIKA	IMIE	NAZWISKO	WIEK	WZROST	WAGA	EMAIL	HASLO	PLEC	ID_PLANU
1	NoweImie	Kowalski	30	180	80	jan.kowalski@gmail.com	haslo123	-	1
2	Anna	Nowak	25	165	60	anna.nowak@gmail.com	password12	K	2
3	Piotr	Wiśniewski	35	175	70	piotr.wisniewski@gmail.com	securepass	M	3
4	Ewa	Kowalczyk	28	160	55	ewa.kowalczyk@gmail.com	qwerty123	K	4
5	Krzysztof	Lis	32	185	90	krzysztof.lis@gmail.com	letmein123	M	5

```
--Aktualizacja danych rodzaju_diety
UPDATE Rodzaj_diety
SET NAZWA_DIETY = 'NowaDieta'
WHERE id_rodzaju = 1;
```

Wynik:

ID_RODZAJU	NAZWA_DIETY
1	NowaDieta
2	Wegańska
3	Mięsna
4	Paleo
5	Wysokokaloryczna

```
--Usuwanie użytkownika i posilku
```

```
DELETE FROM Uzytkownik
WHERE id_uzytkownika = 1;
```

```
DELETE FROM Posilki
WHERE id_posilku = 1;
```

## Wyniki:

ID_UZYTEKOWNIKA	IMIE	NAZWISKO	WIEK	WZROST	WAGA	EMAIL	HASLO	PLEC	ID_PLANU
2	Anna	Nowak	25	165	60	anna.nowak@gmail.com	password12	K	2
3	Piotr	Wiśniewski	35	175	70	piotr.wisniewski@gmail.com	securepass	M	3
4	Ewa	Kowalczyk	28	160	55	ewa.kowalczyk@gmail.com	qwerty123	K	4
5	Krzysztof	Lis	32	185	90	krzysztof.lis@gmail.com	letmein123	M	5

Download CSV

4 rows selected.

ID_POSILKU	NAZWA_POSILKU	KCAL_POSILKU	TLUSZCZ_POSILKU	BIALKA_POSILKU	WEGLOWODANY_POSILKU	SOL_POSILKU
2	serniki	980	96.2	7.3	22.3	1.52
3	carbonara	615	17.3	23	80	2.2
4	barszcz	68	12.2	2.9	13.9	1.02
5	cezar	450	20	20	30	1
6	kotlet z ryby	150	5	25	10	2

Download CSV

5 rows selected.