

MatrimonialBureauDB

Miraslau Alkhovik 248655

Praskouya Horbach 248656

1 . Podstawowe założenia projektu

Cel projektu: stworzenie obiektowo-relacyjnej bazy danych dla biura matrymonialnego, umożliwiającej zarządzanie klientami, ich preferencjami, przypisaniem do opiekunów, organizowaniem randek oraz monitorowaniem działań za pomocą logowania operacji. System ma wspierać funkcjonalności związane z dodawaniem, edytowaniem i przeglądaniem danych w sposób intuicyjny i zgodny z założeniami obiektowego modelowania danych.

Główne założenia:

Założenia dla bazy danych

1. Klienci i opiekunowie:

- Każdy klient jest przypisany dokładnie do jednego opiekuna.
- Jeden opiekun może mieć wielu klientów (relacja jeden-do-wielu).
- Lista podopiecznych danego opiekuna jest przechowywana w zagnieżdżonej tabeli (**KlientRefTab**) w tabeli **OpiekunowieObjTable**.
- Klient nie może umawiać randek sam ze sobą.
- Klient musi być pełnoletni (sprawdzenie daty urodzenia).
- Klient, który był już wcześniej na randce z inną osobą, może być ponownie przypisany tylko wtedy, gdy obie strony wyrażą zgodę na ponowne

spotkanie.

2. Randki:

- Każda randka jest przypisana do dwóch różnych klientów.
- Klient nie może być jednocześnie uczestnikiem dwóch randek w tym samym czasie.
- Randki są przechowywane w tabeli **RandkiObjTable**, a referencje do klientów (**klient1_ref**, **klient2_ref**) wskazują na uczestników.

3. Opinie o randkach:

- Każda opinia jest przypisana do jednej randki oraz jednego klienta, który tę opinię wystawił.
- Ocena randki (**ocena**) musi mieścić się w zakresie od 1 do 5.
- Opinie przechowywane są w tabeli **OpinieRandek** i zawierają szczegółowe komentarze w kolumnie **komentarz**.

4. Logowanie operacji:

- Wszystkie operacje w systemie są logowane w tabeli **LogOperacji**.
- Każdy log zawiera szczegóły operacji, takie jak data wykonania, typ operacji (np. **INSERT**, **UPDATE**, **DELETE**), oraz identyfikator użytkownika, który ją wykonał.
- Logi mogą być używane do audytowania działań w systemie.

5. Proces matchmaking (dobieranie par):

- Matchmaking jest realizowany na podstawie cech klientów, takich jak:
 - Wiek,
 - Preferencje i zainteresowania,
 - Wartości i oczekiwania dotyczące partnera.
- System wyszukuje pary, które mają najwyższy poziom dopasowania według ustalonych kryteriów.

6. Kryteria dopasowania:

Dwustronne dopasowanie:

- Preferencje jednej osoby są porównywane z cechami drugiej osoby i odwrotnie.
- Używane są:
 - **Cechy** (np. zainteresowania, wartości) i **preferencje** (zdefiniowane przez klienta).
 - **Wiek** – zarówno preferowany wiek, jak i wiek osoby.

- **Orientacja seksualna i płeć** – zgodność w tych kryteriach jest kluczowa.

Wykluczenie niedopasowanych par:

- Klienci, którzy mieli negatywne doświadczenia z osobą (opinia < 3) są wykluczani.
- Klienci, którzy byli już wcześniej na randce z daną osobą i nie spełniają kryteriów, nie są rozpatrywani.

Punktacja dopasowania:

- Każda potencjalna para jest oceniana na podstawie punktacji uwzględniającej:
 - Zgodność cech i preferencji (waga: 30%).
 - Zgodność wieku w obu kierunkach (waga: 20%).
 - Zgodność orientacji seksualnej i płci (waga: 50%).

Dynamiczne filtrowanie:

- W pierwszym kroku procedura zlicza potencjalnych kandydatów.
- W drugim kroku zwraca listę kandydatów, posortowaną malejąco według punktacji dopasowania.

Kolejność wyników:

- Kandydaci są zwracani w kolejności od najlepszego dopasowania do najmniej dopasowanego.

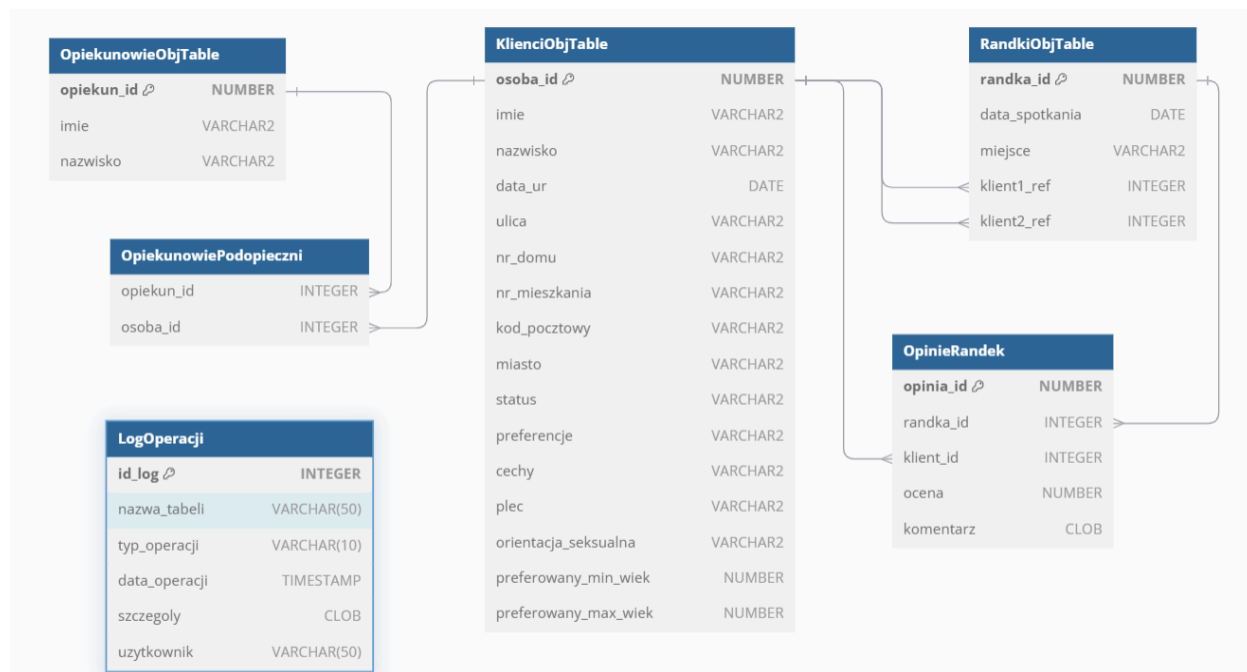
Możliwości:

1. **Dodawanie danych:** Możliwość dodawania nowych klientów, opiekunów oraz randek + Automatyczna walidacja i przetwarzanie danych w oparciu o obiektowe metody i procedury.
2. **Relacje i referencje:** Obsługa złożonych relacji między obiektami (klient-opiekun, klient-randka).
3. **Wyświetlanie danych:** Funkcje i procedury umożliwiające pobieranie danych o klientach, opiekunach i randkach w ustrukturyzowanej formie.
4. **Logowanie operacji:** Śledzenie historii zmian w tabeli klientów, umożliwiające audyt działań.

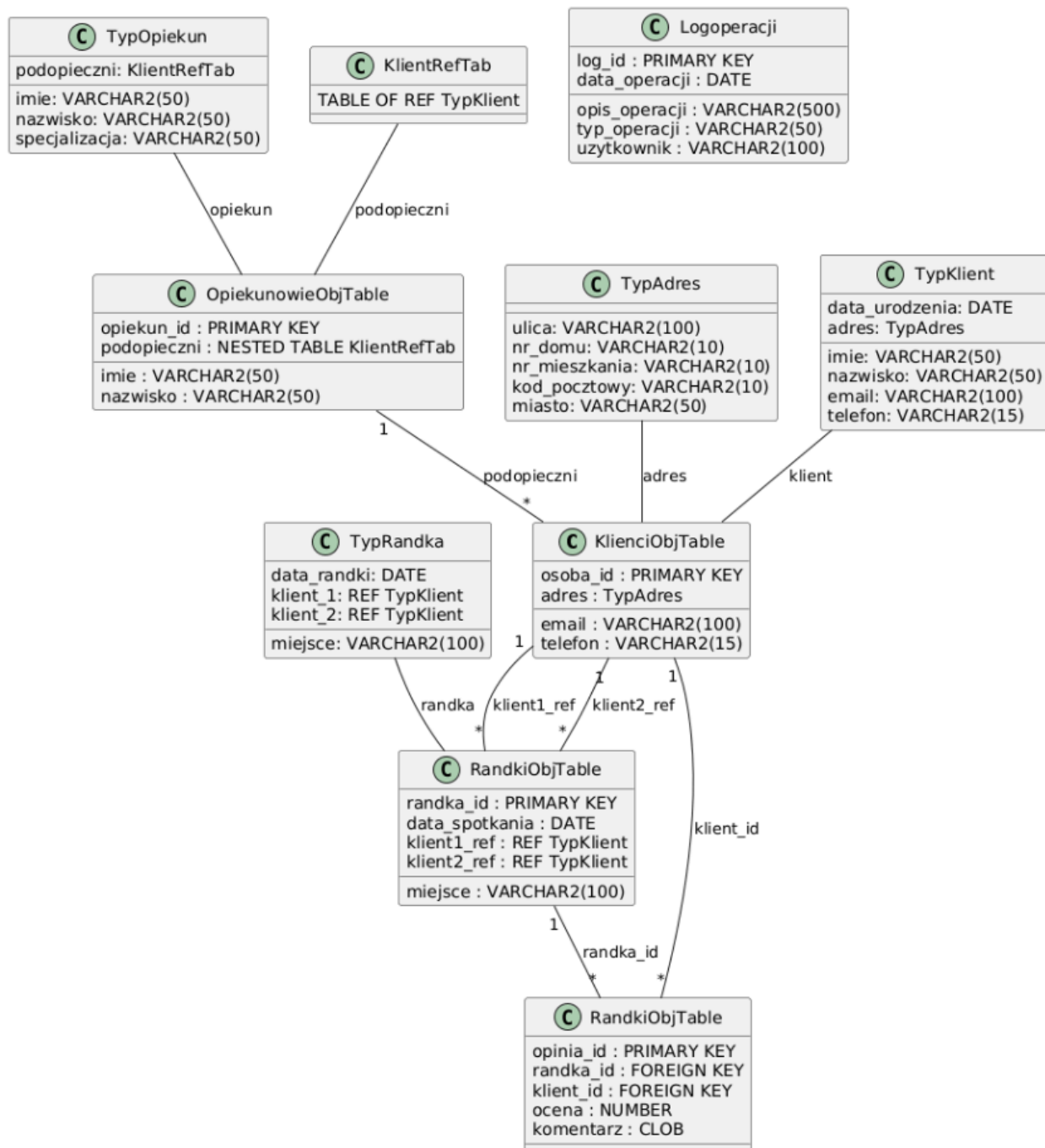
Ograniczenia przyjęte przy projektowaniu:

1. **Wymóg zgodności z Oracle Database:** System wykorzystuje typy obiektowe i mechanizmy specyficzne dla Oracle, co ogranicza przenośność na inne systemy bazodanowe.
2. **Brak złożonych mechanizmów raportowania:** System koncentruje się na podstawowych funkcjach CRUD, logowaniu i walidacji, bez zaawansowanego raportowania.
3. **Brak automatycznego usuwania nieużywanych obiektów:** Referencje (REF) nie są automatycznie usuwane, co wymaga ręcznego zarządzania relacjami przy usuwaniu klientów.

2 . Schemat bazy danych



3. Diagram relacji obiektów



4. Obiekty

Typy danych:

1. TypAdres

- **Opis:** Typ przechowujący dane adresowe.
- **Atrybuty:**
 - `ulica` (VARCHAR2(100)) – nazwa ulicy.
 - `nr_domu` (VARCHAR2(10)) – numer domu.
 - `nr_mieszkania` (VARCHAR2(10)) – numer mieszkania (opcjonalnie).
 - `kod_pocztowy` (VARCHAR2(10)) – kod pocztowy.
 - `miasto` (VARCHAR2(50)) – nazwa miasta.
- **Funkcje:**
 - `pokaz_adres`: Zwraca pełny adres w formie tekstu.

2. TypOsoba

- **Opis:** Typ reprezentujący dane osoby.
- **Atrybuty:**
 - `osoba_id` (NUMBER) – unikalny identyfikator osoby.
 - `imie` (VARCHAR2(50)) – imię osoby.
 - `nazwisko` (VARCHAR2(50)) – nazwisko osoby.
 - `data_urodzenia` (DATE) – data urodzenia osoby.
 - `adres` (TypAdres) – obiekt zawierający dane adresowe osoby.
- **Procedury:**
 - `pokaz_dane`: Wyświetla dane osoby, w tym imię, nazwisko, datę urodzenia i adres.
- **Funkcje:**
 - `get_imie_nazwisko`: Zwraca imię i nazwisko osoby.

3. TypKlient (rozszerza TypOsoba)

- **Opis:** Typ reprezentujący klienta biura matrymonialnego. Rozszerza typ `TypOsoba`, dodając specyficzne atrybuty dla klientów.
- **Atrybuty:**
 - `status_klienta` (VARCHAR2(20)) – status klienta (np. "aktywny",

"nieaktywny").

- **preferencje** (VARCHAR2(200)) – preferencje klienta dotyczące drugiej osoby.
- **cechy** (VARCHAR2(500)) – cechy preferowane przez klienta.
- **plec** (VARCHAR2(10)) – płeć klienta.
- **orientacja_seksualna** (VARCHAR2(30)) – orientacja seksualna klienta.
- **preferowany_min_wiek** (NUMBER) – minimalny wiek preferowanej osoby.
- **preferowany_max_wiek** (NUMBER) – maksymalny wiek preferowanej osoby.
- **Funkcje:**
 - **pelne_dane**: Zwraca pełne dane klienta, w tym imię, nazwisko, datę urodzenia, miasto, status, preferencje, cechy, płeć, orientację seksualną i wiek.

4. TypRandka

- **Opis**: Typ reprezentujący randkę pomiędzy dwoma klientami.
- **Atrybuty**:
 - **randka_id** (NUMBER) – unikalny identyfikator randki.
 - **data_spotkania** (DATE) – data spotkania.
 - **miejsce** (VARCHAR2(100)) – miejsce spotkania.
 - **klient1_ref** (REF TypKlient) – referencja do pierwszego klienta.
 - **klient2_ref** (REF TypKlient) – referencja do drugiego klienta.
- **Funkcje**:
 - **opis**: Zwraca krótki opis randki, zawierający ID randki, datę i miejsce spotkania.

5. KlientRefTab

- **Opis**: typ tabeli osadzonej (nested table), którego elementami są referencje (REF) do obiektów typu **TypKlient**. Pozwala to na przechowywanie listy odwołań do instancji klientów w sposób logicznie powiązany z innymi obiektami.
- **Typ**: **TABLE OF REF TypKlient**.

6. TypOpiekun

- **Opis**: Typ reprezentujący opiekuna, który zarządza grupą klientów.
- **Atrybuty**:

- `opiekun_id` (NUMBER) – unikalny identyfikator opiekuna.
 - `imie` (VARCHAR2(50)) – imię opiekuna.
 - `nazwisko` (VARCHAR2(50)) – nazwisko opiekuna.
 - `podopieczni` (KlientRefTab) – kolekcja referencji do klientów, którymi opiekun się opiekuje.
 - **Funkcje:**
 - `pokaz_opiekuna`: Zwraca krótki opis opiekuna, zawierający jego imię, nazwisko oraz identyfikator.
-

Tabele:

1. KlienciObjTable

- **Opis:** Tabela przechowująca dane obiektów typu **TypKlient**.
- **Kolumny:**
 - `osoba_id` (PRIMARY KEY) – unikalny identyfikator osoby (klienta).
 - Pozostałe kolumny przechowują atrybuty typu **TypKlient**.

2. RandkiObjTable

- **Opis:** Tabela przechowująca dane obiektów typu **TypRandka**.
- **Kolumny:**
 - `randka_id` (PRIMARY KEY) – unikalny identyfikator randki.
 - `data_spotkania` (DATE) – data spotkania.
 - `miejsce` (VARCHAR2(100)) – miejsce spotkania.
 - `klient1_ref` (REF TypKlient) – referencja do pierwszego klienta.
 - `klient2_ref` (REF TypKlient) – referencja do drugiego klienta.

3. OpiekunowieObjTable

- **Opis:** Tabela przechowująca dane obiektów typu **TypOpiekun**.
- **Kolumny:**
 - `opiekun_id` (PRIMARY KEY) – unikalny identyfikator opiekuna.
 - `imie` (VARCHAR2(50)) – imię opiekuna.
 - `nazwisko` (VARCHAR2(50)) – nazwisko opiekuna.
 - `podopieczni` (NESTED TABLE) – przechowuje dane podopiecznych (klientów) w formie tabeli.

4. PodopieczniStorage

- **Opis:** Tabela przechowująca dane zagnieżdżonych klientów w ramach tabeli **OpiekunowieObjTable**.
- **Kolumny:**
 - **NESTED_TABLE_ID** (PRIMARY KEY) – unikalny identyfikator zagnieżdżonego wpisu.
 - Pozostałe kolumny przechowują dane klientów powiązanych z opiekunem.

5. OpinieRandek

- **Opis:** Tabela przechowująca opinie o randkach.
- **Kolumny:**
 - **opinia_id** (PRIMARY KEY) – unikalny identyfikator opinii.
 - **randka_id** (FOREIGN KEY) – identyfikator randki.
 - **klient_id** (FOREIGN KEY) – identyfikator klienta.
 - **ocena** (NUMBER) – ocena randki w skali od 1 do 5.
 - **komentarz** (CLOB) – tekstowy komentarz dotyczący randki.

6. LogOperacji

- **Opis:** Tabela przechowująca logi operacji wykonywanych w systemie, używana do audytowania działań i monitorowania aktywności.
- **Kolumny:**
 1. **log_id** (PRIMARY KEY) – unikalny identyfikator logu.
 2. **data_operacji** (DATE) – data i czas wykonania operacji.
 3. **opis_operacji** (VARCHAR2(500)) – krótki opis wykonanej operacji.
 4. **typ_operacji** (VARCHAR2(50)) – typ operacji (np. "INSERT", "UPDATE", "DELETE").
 5. **uzytkownik** (VARCHAR2(100)) – identyfikator użytkownika lub systemu, który wykonał operację.

Pakiety i triggerzy:

1. Pakiet **PakietBiuro**

1.Procedury dotyczące klientów:

- `dodaj_klienta` – Dodaje nowego klienta do systemu.
- `wyswietl_klienta` – Wyświetla informacje o kliencie.
- `pobierz_klientow` – Pobiera listę wszystkich klientów.

2.Procedury dotyczące opiekunów:

- `dodaj_opiekuna` – Dodaje nowego opiekuna do systemu.
- `dodaj_klienta_do_opiekuna` – Przypisuje klienta do opiekuna.
- `pobierz_opiekunow` – Pobiera listę wszystkich opiekunów.
- `pokaz_podopiecznych` – Wyświetla podopiecznych przypisanych do opiekuna.

3. Procedury dotyczące randek:

- `dodaj_randke` – Dodaje nową randkę do systemu.
- `pobierz_randki` – Pobiera listę wszystkich randek.
- `znajdz_pare` – Wyszukuje potencjalną parę dla klienta.

4. Procedury dotyczące opinii i historii:

- `dodaj_opinie` – Dodaje opinię dotyczącą randki.
- `historia_randek` – Wyświetla historię randek dla klienta.

5. Procedura wyszukiwania klientów:

- `wyszukaj_klienta_po_cechach` – Wyszukuje klienta na podstawie podanych cech.

2. Triggery:

- `trg_check_randka_self`: Sprawdza, czy klient nie jest przypisany jako oba uczestnicy tej samej randki.

- **trg_check_data_urodzenia**: Weryfikuje, czy data urodzenia klienta nie jest w przyszłości.
- **trg_log_operacji**: Rejestruje operacje na tabeli **KlienciObjTable** w **LogOperacji**.
- **trg_check_double_booking**: Sprawdza, czy klient nie ma podwójnej rezerwacji randki w tym samym czasie.

5 . Role i uprawnienia

1. **Administrator**:

- **Zakres uprawnień:**
 - **Pełna kontrola nad systemem:**
 1. Dostęp do wszystkich tabel, typów i procedur.
 2. Możliwość przeglądania, modyfikowania, dodawania i usuwania danych.
 - **Zarządzanie strukturą bazy danych:**
 1. Tworzenie i usuwanie tabel, typów oraz użytkowników.
 2. Modyfikacja struktury obiektów w bazie (np. dodawanie nowych kolumn do tabel).
 - **Wywoływanie procedur i funkcji:**
 1. Dostęp do wszystkich procedur i funkcji z pakietu **PakietBiuro**.

2. **Employee:**

- **Zakres uprawnień:**

- **Praca z danymi:**

- 1. Dodawanie i aktualizacja danych klientów oraz randek.
 - 2. Przeglądanie danych klientów, randek i opiekunów.

- **Dostęp do procedur:**

- 1. Wywoływanie określonych procedur z pakietu **PakietBiuro**.

- **Ograniczenia:**

- 1. Brak możliwości usuwania danych.
 - 2. Brak możliwości modyfikacji struktury bazy danych (np. tworzenia tabel, modyfikacji typów).