

Bazy Danych 1

Dokumentacja projektu

Mateusz Nizio

1. Uruchomienie aplikacji

Aplikacja została napisana w języku Java i łączy się z bazą danych znajdującą się na serwerze Pascal. Aby ją uruchomić należy kolejno (na serwerze Pascal):

-przejsć do folderu w którym jest aplikacja

```
cd ~9nizio/Projekt/src
```

-uruchomić aplikację poleceniem

```
java -cp ../postgresql-42.2.12.jar Main
```

Aplikacja jest już skompilowana, lecz w razie jakichkolwiek problemów z uruchomieniem należy kolejno użyć poleceń:

-rm *.class

-javac Main.java

- java -cp ../postgresql-42.2.12.jar Main

2. Ogólny opis

Aplikacja symuluje system bazy danych hotelu. Po zarejestrowaniu i zalogowaniu jako zwykły użytkownik, możemy dodać nową rezerwację oraz zarządzać swoimi rezerwacjami (opłacić, anulować, wyświetlić szczegóły). Po zalogowaniu jako administrator (login: admin, hasło: admin), oprócz funkcjonalności zwykłego użytkownika możemy także zarządzać obsługą każdego z hoteli (np. usunąć danego pracownika).

3. Projekt diagramów

W projekcie zostały użyte następujące obiekty wraz z ich atrybutami:

-**hotel**(hotel_id, nazwa, adres, kraj, gwiazdki)

-**klient**(klient_id, imie, nazwisko, email, numer, płeć, typ)

-**konto_w_serwisie**(konto_id, login, haslo, czy_istnieje)

-**kategoria_pokoju**(id_kategorii, cena_od_osoby, nazwa)

-**platnosc**(platnosc_id, czy_dokonano, rachunek_id, karta, gotowka)

-**pokoj**(pokoj_id, numer, pietro, ilosc_miejsc, dostepny, hotel_nazwa, kategoria_id)

-**pracownik_hotelu**(pracownik_id, imie, nazwisko, pensja, email, staz, pozycja_id, hotel_id)

-**pozycja_pracownika**(pozycja_id, opis_pozycji)

-**rachunek**(rachunek_id, zamowienie_id, kwota, stan)

-**uslugi**(zamowienie_id, basen, silownia, sauna)

Główną i najważniejszą encją jest encja „zamówienie” która jest łącznikiem pomiędzy zarezerwowaniem wybranego pokoju oraz klientem. Na schemacie widać również dodatkowe tabelę asocjacyjną łączącą zamówienie z pokojem. Znaczna większość tabel jest połączona relacją wiele do wielu np. jeden pracownik może pracować w kilku różnych hotelach. Występują także połączenia 1 do wielu, np. jeden klient może mieć kilka różnych rachunków. W bazie są zdefiniowane także relacje jeden do jeden, np. jedna płatność odpowiada jednemu rachunkowi, użytkownik może mieć tylko jedno konto w serwisie. Schemat projektu znajduje się poniżej:

The diagram illustrates the database structure for a hotel management system. It includes the following tables and their attributes:

- Konto_vr_servisie**: id_login (INTEGER), Klient_klient_id (INTEGER (FK)), konto_vv_servisie (BOOLEAN), login (TEXT), haslo (TEXT), login_FKIndex1 (FK to Klient_klient_id).
- Klient**: klient_id (INTEGER), imie (TEXT), nazwisko (TEXT), e-mail (TEXT), numer_telefonu (INTEGER), dorosly (BOOLEAN), plec (BOOLEAN), typ (TEXT).
- Platnosc**: idPlatnosc (INTEGER), Klient_klient_id (INTEGER (FK)), czy_dokonano (BOOLEAN), data_platnosc (DATE), gotowka (BOOLEAN), karta (BOOLEAN), Platnosc_FKIndex1 (FK to Klient_klient_id).
- Rachunek**: rachunek_id (INTEGER), Platnosc_klient_id (INTEGER (FK)), Platnosc_idPlatnosc (INTEGER (FK)), naliczone_mnize (INTEGER), Rachunek_FKIndex1 (FK to Platnosc_klient_id), Platnosc_idPlatnosc (FK to Platnosc_idPlatnosc).
- Zamowienie**: zamowienie_id (INTEGER), Rachunek_rachunek_id (INTEGER (FK)), Dodatkowe_uslugi_id_pakietu (INTEGER (FK)), Klient_klient_id (INTEGER (FK)), data_rezerwacji (DATE), godzina_rezerwacji (TIME), data_przyjazdu (DATE), data_odjazdu (DATE), liczba_dzieci (INTEGER), liczba_doroslych (INTEGER), kczba_zwiera (INTEGER), Zamowienie_FKIndex2 (FK to Klient_klient_id), Rachunek_rachunek_id (FK to Rachunek_rachunek_id), Zamowienie_FKIndex2 (FK to Zamowienie_id), Dodatkowe_uslugi_id_pakietu (FK to Dodatkowe_uslugi_id_pakietu), Zamowienie_FKIndex3 (FK to Klient_klient_id).
- Rezerwacja_Pokoi**: Pokoj_Hotel_hotel_id (INTEGER (FK)), Pokoj_pokoj_id (INTEGER (FK)), Zamowienie_zamowienie_id (INTEGER (FK)), Zamowienie_Rachunek_rachunek_id (INTEGER (FK)), Zamowienie_Dodatkowe_uslugi_id_pakietu (INTEGER (FK)), Pokoj_kategoria_pokoju_kategoria (INTEGER (FK)), Zamowienie_Klient_klient_id (INTEGER (FK)), Rezerwacja_has_Pokoj_FKIndex1 (FK to Zamowienie_zamowienie_id), Zamowienie_zamowienie_id (FK to Zamowienie_id), Zamowienie_Rachunek_rachunek_id (FK to Rachunek_rachunek_id), Zamowienie_Dodatkowe_uslugi_id_pakietu (FK to Dodatkowe_uslugi_id_pakietu), Zamowienie_Klient_klient_id (FK to Klient_klient_id), Rezerwacja_has_Pokoj_FKIndex2 (FK to Pokoj_pokoj_id), Pokoj_pokoj_id (FK to Pokoj_pokoj_id), Pokoj_Hotel_hotel_id (FK to Pokoj_Hotel_hotel_id), Pokoj_kategoria_pokoju_kategoria (FK to Pokoj_kategoria_pokoju_kategoria).
- Pokoj**: pokoj_id (INTEGER), Hotel_hotel_id (INTEGER (FK)), kategoria_pokoju_kategoria (INTEGER (FK)), pietro (INTEGER), numer (INTEGER), ilosc_miejsc (INTEGER), dostepny (BOOLEAN), Pokoj_FKIndex2 (FK to Hotel_hotel_id), Hotel_hotel_id (FK to Hotel_hotel_id), Pokoj_FKIndex2 (FK to Pokoj_id), kategoria_pokoju_kategoria (FK to kategoria_pokoju_kategoria).
- Pracownik_hotelu**: pracownik_id (INTEGER), Hotel_hotel_id (INTEGER (FK)), Pozycja_pozycja_id (INTEGER (FK)), imie (TEXT), nazwisko (TEXT), pensja (INTEGER), email (TEXT), staz (INTEGER), Pracownik_hotelu_FKIndex1 (FK to Hotel_hotel_id), Hotel_hotel_id (FK to Hotel_hotel_id), Pracownik_hotelu_FKIndex2 (FK to Pozycja_pozycja_id), Pozycja_pozycja_id (FK to Pozycja_pozycja_id).
- Pozycja**: pozycja_id (INTEGER), pozycja_nazwa (TEXT), pozycja_opis (TEXT).

Relationships (Rel) are defined as follows:

- Rel_01**: 1:M between Konto_vr_servisie and Klient.
- Rel_13**: 1:M between Rachunek and Zamowienie.
- Rel_15**: 1:M between Zamowienie and Rezerwacja_Pokoi.
- Rel_16**: 1:M between Pokoj and Rezerwacja_Pokoi.
- Rel_18**: 1:M between Platnosc and Rachunek.
- Rel_20**: 1:M between Klient and Platnosc.
- Rel_21**: 1:M between Klient and Zamowienie.
- Rel_25**: 1:M between Pozycja and Pracownik_hotelu.
- Rel_27**: 1:M between Zamowienie and Rezerwacja_Pokoi.
- Rel_28**: 1:M between Pokoj and Rezerwacja_Pokoi.
- Rel_08**: 1:M between Hotel and Pracownik_hotelu.
- Rel_11**: 1:M between Hotel and Pokoj.

4. Projektowanie tabel, kluczy indeksów

Poniżej wypisane są wszystkie użyte tabele wraz z ich atrybutami oraz kluczami:

4.1 konto_w_serwisie

- konto_id – klucz główny INT NOT NULL
- login VARCHAR
- hasło VARCHAR

4.2 hotel

- hotel_id – klucz główny INT NOT NULL
- nazwa VARCHAR
- adres VARCHAR
- kraj VARCHAR
- gwiazdki INT
- wolne_miejsca BOOLEAN

4.3 klient

- klient_id klucz główny INT NOT NULL
- imie VARCHAR
- nazwisko VARCHAR
- email VARCHAR
- numer INT
- plec VARCHAR
- typ VARCHAR
- konto_id klucz obcy INT

4.4 pokój

- pokoj_id klucz główny INT NOT NULL
- numer INT
- pietro INT
- ilość_miejsc INT
- dostępny BOOLEAN
- hotel_nazwa klucz obcy INT
- kategoria_id klucz obcy INT

4.5 kategoria_pokoju

- kategoria_id klucz główny INT NOT NULL
- cena_od_osoby INT NOT NULL

4.6 platnosc

- platnosc_id klucz główny INT NOT NULL
- czy_dokonano BOOLEAN
- rachunek_id klucz obcy

4.7 pracownik_hotelu

- pracownik_id klucz główny INT NOT NULL
- imie VARCHAR
- nazwisko VARCHAR
- pensja INT
- email VARCHAR
- staz INT
- pozycja_id klucz obcy
- hotel_id klucz obcy

4.8 pozycja_pracownika

- pozycja_id klucz główny INT NOT NULL
- opis VARCHAR

4.9 rachunek

- rachunek_id klucz główny INT NOT NULL
- zamówienie_id klucz obcy
- kwota INT
- stan INT

4.10 usługi

- usługi_id klucz główny NOT NULL
- zamówienie_id klucz obcy
- basen BOOLEAN
- siłownia BOOLEAN
- sauna BOOLEAN

4.11 zamówienie

- zamówienie_id klucz główny INT NOT NULL
- data_przyjazdu DATE
- data_odjazdu DATE
- liczba_dzieci INT
- liczba_dorosłych INT
- pokój_id klucz obcy
- klient_id klucz obcy

Niektóre tabele zostały uprzednio wypełnione danymi na potrzeby programu (np. tabela hotel, pracownicy_hotelu, pokoje). Inserty dokonane do tabel dostępne są w pliku sql/inserty_do_tablic.sql

Oprócz wyżej wymienionych encji, na potrzeby projektu zostało stworzonych kilka widoków które usprawniły pracę i pomogły wprowadzać dane do tabeli. Wszystkie widoki zawarte są w sql/widoki.sql. Przykładowym może być widok PokojAndZamowienie który łączy w sobie tabele pokój oraz zamówienie przypisane do tego pokoju na podstawie klucza głównego pokoju. Innym widokiem jest widok HotelAndPokoj który wypisuje wszystkie pokoje znajdujące się w danym hotelu. HotelAndPracownicy wypisuje wszystkich pracowników którzy pracują w danym hotelu.

5. Działanie programu na poziomie tabel

Po włączeniu aplikacji użytkownik ma możliwość zalogowania się lub rejestracji. Gdy wybierze rejestrację, wprowadzając dane przez formularz, są zapisywane do tabeli `konto_w_serwisie` oraz `klient`. Następnie użytkownik może zalogować się do programu- następuje walidacja czy login oraz hasło istnieją w bazie(sprawdzone w encji `konto_w_serwisie`). Kolejnym krokiem jest np. dokonanie nowej rezerwacji. Tutaj po wybraniu interesujących użytkownika parametrów(data wyjazdu, odjazdu, liczba osób), funkcje sprawdzają czy w danym hotelu są wolne miejsca, a jeżeli tak to czy w wybranym przez użytkownika terminie są wolne pokoje dla wybranej liczby osób(liczba miejsc dostępnych w pokoju> liczba miejsc zadeklarowanych przez użytkownika). Następnie użytkownik może wybrać usługi dodatkowe. Po każdej zmianie wyzwalana jest także funkcja która oblicza aktualną cenę za pobyt w hotelu przy zadanych parametrach. Po dokonaniu rezerwacji dane są zapisywane do tabeli `zamowienie`, a następnie wyzwalacze zapisują kolejne dane na podstawie zamówienia do tabel `rachunek` oraz `płatność`. Użytkownik może teraz przeglądać swoje zamówienia. Po wybraniu jednego z nich można np. wyświetlić szczegóły, gdzie dzięki przygotowanym widokom, widzimy wszystkie interesujące informacje na temat hotelu, rachunku, zamówienia i płatności. Dane te są pobierane z tabeli `zamowienie`, `uslugi`, `hotel`, `pokoj`, `rachunek`. Użytkownik może także usunąć swoje zamówienie, co wiąże się z usunięciem go z tabeli `zamowienie` oraz uruchomieniem wyzwalaczy aby usunęły także `rachunek` oraz `płatność` dotyczącą danego zamówienia. Administrator ma także możliwość przeglądania wszystkich pracowników danego hotelu oraz zwalnianie każdego z nich(tabele `hotel`, `pracownik_hotelu`).

6. Funkcje oraz wyzwalacze

W projekcie użyto wielu funkcji które miały na celu odczyt konkretnych danych z tabel, usuwanie lub aktualizowanie rekordów. Funkcje zostały umieszczone w pliku `sql/funkcje.sql`. Użyto także triggerów które wyzwały się automatycznie po dokonaniu pewnych operacji. Na przykład, po złożeniu nowego zamówienia przez użytkownika, program automatycznie tworzy nowy rachunek przypisany do danego zamówienia. Innym przykładem może być automatyczna zmiana statusu w encji „`płatność`” oraz „`rachunek`” po dokonaniu opłaty przez użytkownika.

7. Interfejs graficzny

Program został napisany w języku Java, z wykorzystaniem frameworku Java Swing. Każde nowe okno jakie można otworzyć to nowy `JPanel`(panel z java swing). Między panelami są przesyłane informacje takie jak: panel poprzedni(aby można było powrócić do poprzedniego okna) oraz dane użytkownika(unikalne id), tak aby w każdej chwili z bazy można było wybrać interesujące nas informacje np. na temat zamówienia. Program rozpoczyna się panelem startowym, a następnie mamy do wyboru panel rejestracji lub logowania. Potem głównym panelem jest Menu z którego można przejść do rezerwacji, szczegółów lub panelu administratora. Istnieją także panele potwierdzające lub odrzucające rezerwację. Najważniejszym panelem który łączy w sobie wszystkie inne jest panel `db_connection`, który nawiązuje połączenie z bazą danych oraz wysyła do niej zapytania o uruchomienie konkretnych funkcji. W programie istnieją także klasy reprezentujące obiekty takie jak `hotel` czy `pokoj`. Zostały one stworzone w celu łatwiejszego przekazywania informacji między funkcjami. Na przykład instancja klasy `pokoj_wrapper` przechowuje informacje na temat aktualnie wybranego pokoju, takie jak `pietro`, `numer` itd.

8. Źródła

<https://www.javatpoint.com/java-swing>

<https://www.guru99.com/java-swing-gui.html#:~:text=Swing%20in%20Java%20is%20a,GUI%20programing%20that%20provide%20GUI>

<https://docs.oracle.com/javase/tutorial/uiswing/>

<https://www.youtube.com/watch?v=mDxEgtMNPtA>

<https://www.javaguides.net/2019/07/java-swing-application-with-database-connection.html>

<https://stackoverflow.com/questions/10952117/swing-form-database-connection>

Oraz pdfy z zajęć od dr. Grażyny Krupińskiej z upela.