

Dokumentacja projektu wykonywanego w ramach zajęć

Bazy Danych 1

Olga Kubiszyn

10.02.2021

I. Projekt koncepcji, założenia

1. Zdefiniowanie tematu projektu

Tematem projektu jest aplikacja okienkowa dająca dostęp do bazy danych zawierającej informacje o hotelu – gościach, pokojach, rezerwacjach itp.

2. Analiza wymagań użytkownika

Aplikacja umożliwiała odczyt zawartości bazy oraz wprowadzanie nowych danych.

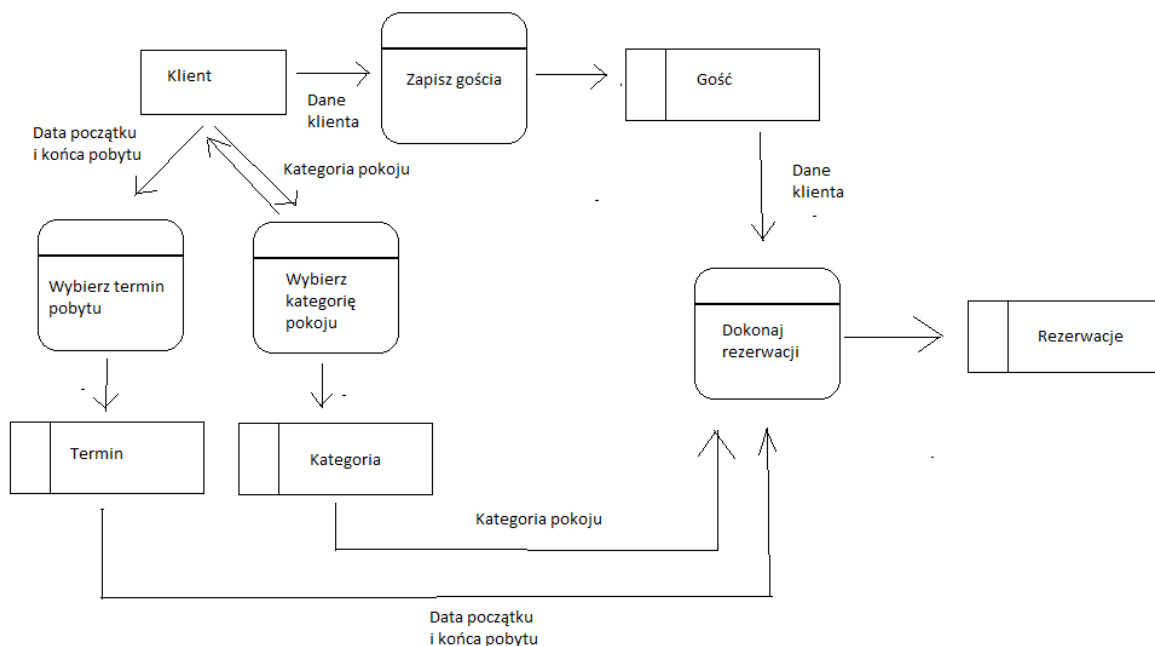
3. Zaprojektowanie funkcji:

- dodawanie nowych gości i ich rezerwacji oraz dodatkowych usług – sprzątania i śniadań,
- dodawanie nowych danych uruchamia odpowiednie funkcje – kontrola poprawności i spójności,
- wybranie danego raportu do wyświetlenia jest umożliwione, dzięki odpowiednim widokom.

II. Projekt diagramów (konceptualny)

1. Budowa i analiza diagramu przepływu danych

Najważniejszym zadaniem bazy jest przetwarzanie rezerwacji klientów na dany termin i kategorię pokoju.



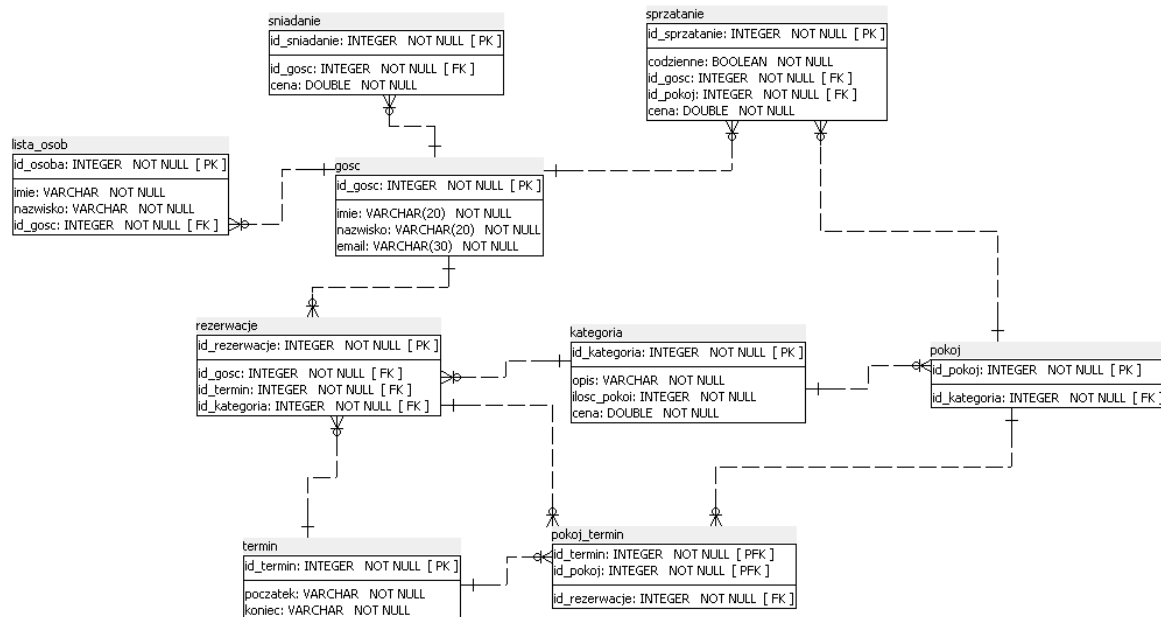
Rysunek 1. Diagram DFD.

2. Zdefiniowanie encji (obiektów) oraz ich atrybutów

- gosc – osoba dokonująca rezerwacji:
 - id_gosc – klucz główny, unikalna wartość
 - imię – imię gościa
 - nazwisko – nazwisko gościa
 - email – email gościa
- lista_osob – lista osób, które przyjechały z danym gościem
 - id_osoba - klucz główny, unikalna wartość

- imię – imię osoby
 - nazwisko – nazwisko osoby
 - id_gosc – id gościa, z którym przyjechała dana osoba
- c) kategoria – kategoria pokoju
- id_kategoria - klucz główny, unikalna wartość
 - opis – opis pokoju np. „1 łóżko”, „1 podwójne łóżko”, „2 łóżka”
 - ilosc_pokoi – ilość pokoi z danej kategorii
 - cena – cena pokoju z danej kategorii
- d) pokój
- id_pokoj - klucz główny, unikalna wartość, równocześnie jest to numer pokoju
 - id_kategoria – kategoria do jakiej należy dany pokój
- e) termin – termin, w jakim przyjeżdża dany gość
- id_termin - klucz główny, unikalna wartość
 - poczatek – data początkowa zapisana w odpowiednim formacie (sprawdzone przez wyzwalacz)
 - koniec – data końcowa zapisana w odpowiednim formacie (sprawdzone przez wyzwalacz)
- a) pokoj_termin – tabela asocjacyjna łącząca pokój i termin oraz posiadająca informację o rezerwacji danego pokoju w danym terminie
- id_pokoj - klucz główny, unikalna wartość
 - id_termin - klucz główny, unikalna wartość
 - id_rezerwacje
- b) rezerwacje – rezerwacja dla danego gościa w danym terminie danej kategorii pokoju
- id_rezerwacje - klucz główny, unikalna wartość
 - id_gosc
 - id_termin
 - id_kategoria
- b) śniadanie – śniadanie wykupione na każdy dzień pobytu, osobno dla każdej osoby, która przyjechała z gościem – jeden gość może zamówić kilka śniadań
- id_sniadanie - klucz główny, unikalna wartość
 - id_gosc – gość, który zamówił śniadanie
 - cena – cena śniadania (kilka opcji do wyboru – różne ceny)
- c) sprzątanie – usługa sprzątania – może być wykupiona na każdy dzień pobytu lub jednorazowo (sprzątanie po wymeldowaniu gościa jest wliczone w cenę pobytu)
- id_sprzatanie - klucz główny, unikalna wartość
 - codzienne – wartość określająca, czy sprzątanie jest wykupione na każdy dzień pobytu czy jednorazowo
 - id_gosc – gość, który wykupił sprzątanie
 - id_pokoj – pokój, który ma być sprzątnięty
 - cena – cena sprzątania (zależna od kategorii pokoju i innych czynników)
3. Zaprojektowanie relacji pomiędzy encjami
- a) gosc – lista_osob 1:N,
 - b) kategoria – pokój 1:N,
 - c) pokój – pokoj_termin 1:N,
 - d) termin - pokoj_termin 1:N,
 - e) gosc – rezerwacje 1:N,
 - f) termin – rezerwacje 1:N,
 - g) rezerwacje – pokoj_termin 1:1,
 - h) kategoria – rezerwacje 1:N,
 - i) gosc – śniadanie 1:N,

- j) gośc – sprzątanie 1:N,
k) pokój – sprzątanie 1:N.



Rysunek 2. Projekt ERD.

III. Projekt logiczny

1. Projektowanie tabel, kluczy, indeksów

Najważniejszą relacją w bazie jest ta pomiędzy tabelą gośc i rezerwacje. Każdy gość może dokonać wielu rezerwacji. Jedna rezerwacja jest zapisywana na jednego gościa. Reszta gości hotelu jest zapisywana w tabeli lista_osob. Każda osoba z listy jest przypisana do jednego gościa, każdy gość może przyjechać z wieloma osobami.

Jedna rezerwacja jest dokonywana na jeden termin, ale w danym terminie może być wiele rezerwacji.

Rezerwacja jest dokonywana na jedną kategorię pokoju. Jedna kategoria może pojawić się w wielu rezerwacjach.

Z każdej kategorii istnieje wiele pokoi, ale jeden pokój ma tylko jedną kategorię.

Jeden pokój może być zarezerwowany w wielu terminach (nienachodzących na siebie), w jednym terminie może być zarezerwowanych wiele pokoi (tabela asocjacyjna). Do tabeli asocjacyjnej dołączono także relację z tabelą rezerwacje 1:1, ponieważ pokój w danym terminie może być raz zarezerwowany – rezerwacja jest na jeden pokój w danym terminie.

Projekt w języku SQL znajduje się w pliku sql/create.sql.

2. Słowniki danych

nazwa_atrybutu	typ danych	własności	informacje
id_gosc	INTEGER	PRIMARY_KEY, NOT NULL	wartość generowana automatycznie przez sekwencje

imie	VARCHAR	NOT NULL	max długość 20
nazwisko	VARCHAR	NOT NULL	max długość 20
email	VARCHAR	NOT NULL	max długość 30

Tabela 1. Słownik dla tabeli gosc

nazwa_atrybutu	typ danych	własności	informacje
id_osoba	INTEGER	PRIMARY_KEY, NOT NULL	wartość generowana automatycznie przez sekwencje
imie	VARCHAR	NOT NULL	
nazwisko	VARCHAR	NOT NULL	
id_gosc	INTEGER	FOREIGN KEY, NOT NULL	klucz główny do tabeli gosc

Tabela 2. Słownik dla tabeli lista_osob.

nazwa_atrybutu	typ danych	własności	informacje
id_kategoria	INTEGER	PRIMARY_KEY, NOT NULL	wartość generowana automatycznie przez sekwencje
opis	VARCHAR	NOT NULL	
ilosc_pokoi	INTEGER	NOT NULL	
cena	DOUBLE	NOT NULL	

Tabela 3. Słownik dla tabeli lista_osob.

nazwa_atrybutu	typ danych	własności	informacje
id_pokoj	INTEGER	PRIMARY_KEY, NOT NULL	Może nie powinna być automatycznie
id_kategoria	INTEGER	FOREIGN KEY, NOT NULL	klucz główny do tabeli kategoria

Tabela 4. Słownik dla tabeli pokoj.

nazwa_atrybutu	typ danych	własności	informacje
id Termin	INTEGER	PRIMARY_KEY, NOT NULL	wartość generowana automatycznie przez sekwencje
poczatek	VARCHAR	NOT NULL	
koniec	VARCHAR	NOT NULL	

Tabela 5. Słownik dla tabeli termin.

nazwa_atrybutu	typ danych	własności	informacje
id_pokoj	INTEGER	PRIMARY_KEY, FOREIGN KEY, NOT	klucz główny do tabeli pokoj

		NULL	
id_termin	VARCHAR	PRIMARY_KEY, FOREIGN KEY, NOT NULL	klucz główny do tabeli termin
id_kategoria	VARCHAR	FOREIGN KEY, NOT NULL	klucz główny do tabeli kategoria

Tabela 6. Słownik dla tabeli pokoj_termin.

nazwa_atrybutu	typ danych	własności	informacje
id_rezerwacje	INTEGER	PRIMARY_KEY, NOT NULL	wartość generowana automatycznie przez sekwencje
id_gosc	INTEGER	FOREIGN KEY, NOT NULL	klucz główny do tabeli gosc
id_termin	INTEGER	FOREIGN KEY, NOT NULL	klucz główny do tabeli termin
id_kategoria	INTEGER	FOREIGN KEY, NOT NULL	klucz główny do tabeli kategoria

Tabela 2. Słownik dla tabeli rezerwacje.

nazwa_atrybutu	typ danych	własności	informacje
id_sniadanie	INTEGER	PRIMARY_KEY, NOT NULL	wartość generowana automatycznie przez sekwencje
id_gosc	INTEGER	FOREIGN KEY, NOT NULL	klucz główny do tabeli gosc
cena	DOUBLE	NOT NULL	

Tabela 6. Słownik dla tabeli sniadanie.

nazwa_atrybutu	typ danych	własności	informacje
id_sprzatanie	INTEGER	PRIMARY_KEY, NOT NULL	wartość generowana automatycznie przez sekwencje
codzienne	BOOLEAN	NOT NULL	
id_gosc	INTEGER	FOREIGN KEY, NOT NULL	klucz główny do tabeli gosc
id_pokoj	INTEGER	FOREIGN KEY, NOT NULL	klucz główny do tabeli pokoj
cena	DOUBLE	NOT NULL	

Tabela 2. Słownik dla tabeli sprzatanie.

3. Analiza zależności funkcyjnych i normalizacja tabel (dekompozycja do 3NF ewentualnieBCNF)

Tabele spełniają założenia trzeciej postaci normalnej. Wartości unikalne są wymuszane tylko dla kluczy głównych.

4. Zaprojektowanie operacji na danych

Kontrola poprawności i spójności danych jest realizowana, dzięki zastosowaniu wyzwalaczy:

- a) rezerwacje_insert (funkcja check_rezerwacje_insert) – sprawdza, czy w danym terminie jest dostępny pokój z danej kategorii,
- b) rezerwacje_insert_pokoj Termin (funkcja rezerwacje_insert_pokoj_termin) – po dokonaniu rezerwacji wprowadza odpowiednie dane do tabeli pokoj_termin – informacje o danym pokoju w danym terminie dla danej rezerwacji,
- c) pokoj_insert (funkcja pokoj_insert_kategoria_update) – uaktualnia ilość pokoi w danej kategorii,
- d) termin_insert (funkcja termin_insert_check) – sprawdza poprawność wprowadzanego terminu.

Wyświetlenie raportu jest umożliwione, dzięki odpowiednim widokom:

- e) rezerwacje_info – zwraca dane gościa, termin, kategorię pokoju i numer pokoju,
- f) rachunek_pokoje – zwraca cenę, ilość dni, gościa i jego rezerwację,
- g) rachunek_suma_pokoje – zwraca sumę opłaty za pokoje danego gościa,
- h) rachunek_sniadania – zwraca sumę opłat za wszystkie śniadania zamówione na danego gościa,
- i) rachunek_sprzatanie – zwraca sumę opłat za wszystkie usługi sprzątania dla danego gościa,
- j) rachunki – zwraca opłaty za poszczególne elementy i całkowitą opłatę dla danego gościa,
- k) gosc_rezerwacje_voucher – zwraca listę gości, którzy mieli więcej niż jedną rezerwację w hotelu,
- l) lista_gosci – zwraca listę osób, które przyjechały z danym gościem,
- m) gosc_oplata_voucher – zwraca listę gości, którzy zapłacili ponad 500 za pokoje.

Widoki ze słowem voucher w nazwie oznaczają, że mogłyby to być przyznawana zniżka dla danych gości. Taka funkcjonalność nie została zrealizowana, ale mogłaby być dodana w przyszłości.

Widoki, funkcje i wyzwalacze znajdują się w pliku sql/widoki_funkcje.sql.

IV. Projekt funkcjonalny

1. Interfejsy do prezentacji, edycji i obsługi danych

W aplikacji dostępne są formularze umożliwiające wprowadzanie danych do tabel. Pola każdego formularza to kolumny danej tabeli oprócz kluczy głównych generowanych automatycznie. Aby wiedzieć jaką wartość kluczy obcych wpisać w formularz, można w trakcie wpisywania wyświetlić odpowiednią tabelę, aby zobaczyć wartość interesującego nas klucza.

2. Wizualizacja danych

Wyświetlanie danych dzieli się na dwie kategorie – wyświetlanie zawartości poszczególnych tabel oraz raportów opartych na widokach. Obie kategorie danych wyświetlane są w tabelach.

3. Zdefiniowanie panelu sterowania aplikacji

Początkowy interfejs aplikacji zawiera 3 przyciski służące do wprowadzania danych, wyświetlania zawartości tabel oraz wyświetlania raportów. Po kliknięciu wybranego przycisku pojawia się możliwość wybrania interesującej nas tabeli/ raportu. Po zatwierdzeniu wyboru w nowym oknie pojawia się odpowiedni formularz/ tabela.

V. Dokumentacja

1. Wprowadzanie danych

Dane w aplikacji wprowadzane są ręcznie przez formularze.

2. Dokumentacja użytkownika

Polecenia do kompilacji i uruchomienia znajdują się w pliku app/readme.md.

Możliwości obsługi aplikacji zostały opisane w punkcie IV dokumentacji.

3. Opracowanie dokumentacji technicznej

Projekt został zrealizowany w języku PostgreSQL – baza oraz Java – GUI. Użyta została baza ElephantSQL. Połączenie jest realizowane dzięki interfejsowi programowania JDBC. GUI zostało wykonane z użyciem biblioteki graficznej Swing. W folderze app znajdują się pliki aplikacji w Javie. Plik Main.java to główny plik aplikacji. Główne okno to JFrame z głównym panelem JPanel. JPanel służy do obsługi wyświetlania tabel i raportów. FormFrame i FormPanel odpowiadają za wyświetlanie formularzy. Połączenie z bazą danych jest realizowane w pliku DB.java.

4. Wykaz literatury

- wykłady i pliki z laboratoriów do przedmiotu Bazy danych 1,
- dokumentacja biblioteki Swing, PostgreSQL.