

BD1 PROJEKT

Skład zespołu:

- Mateusz Kielbus
- Alicja Kubiszyn
- Sara Fojt

Temat projektu

Zdecydowaliśmy się zrealizować projekt o tematyce hotelu. Baza danych zawiera informacje o:

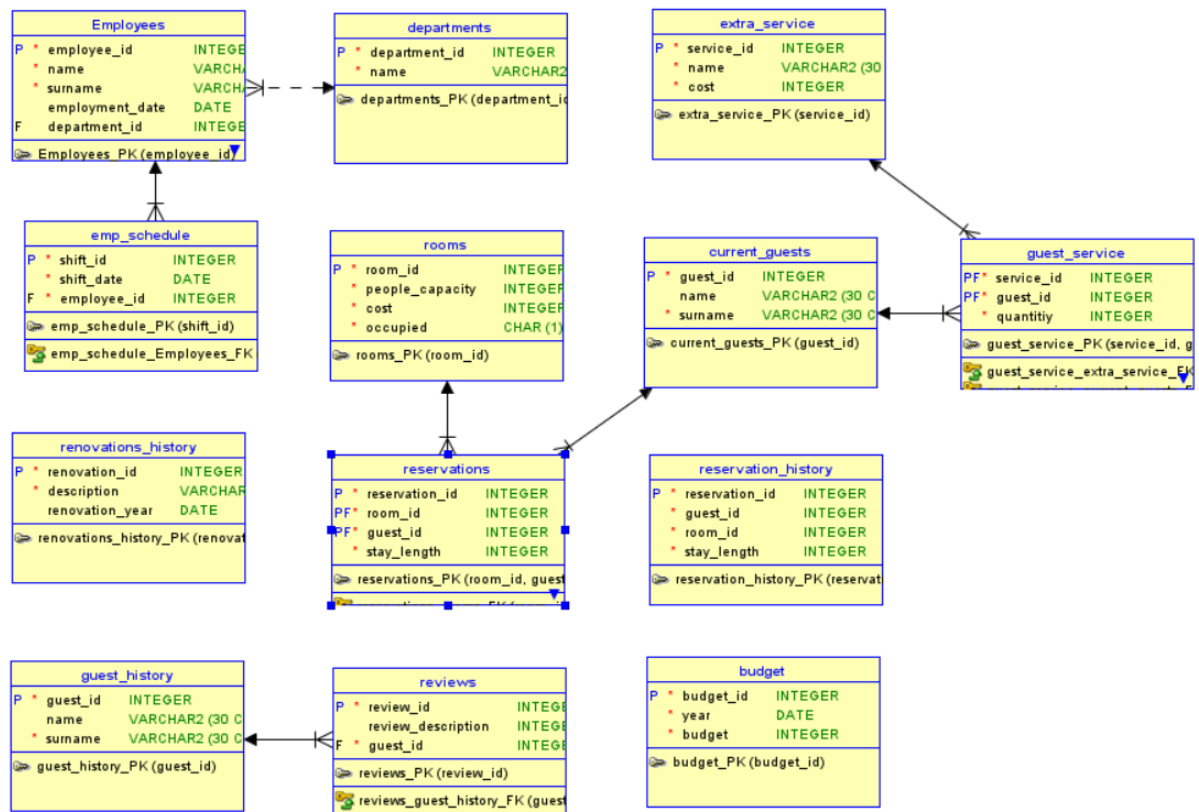
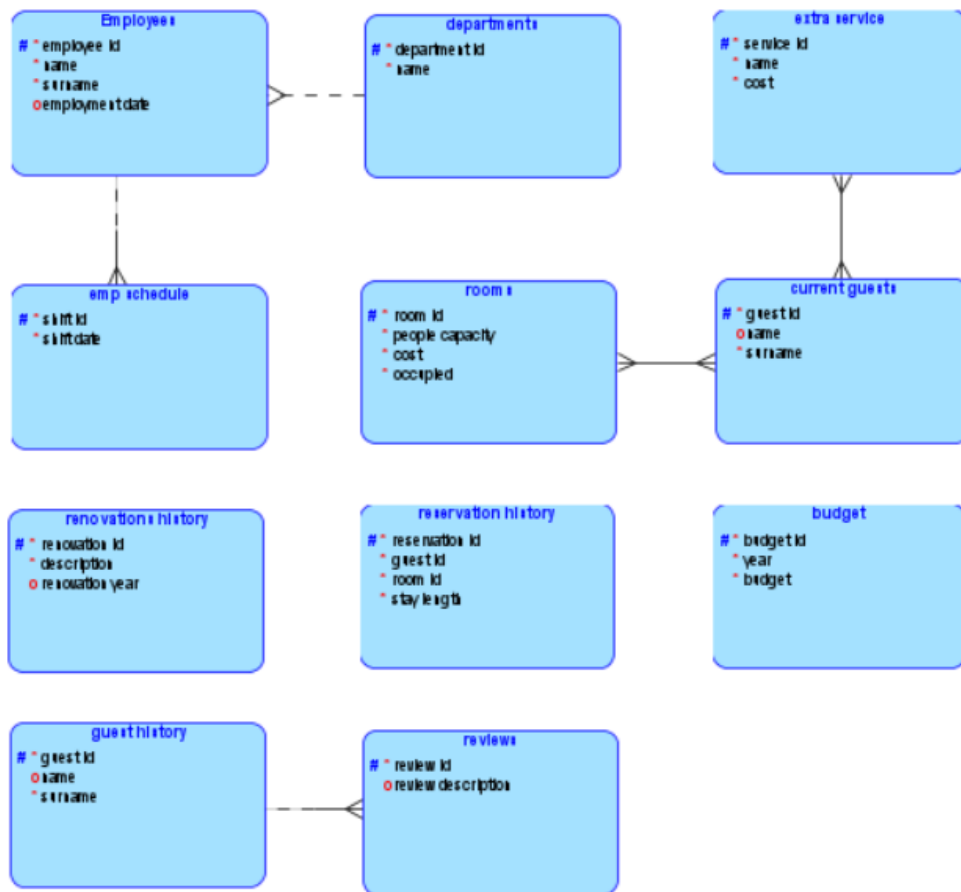
- pracownikach
 - podstawowe dane
 - zmiany
- hotelu
 - działy
 - historia renowacji
 - pokoje
 - budżet
 - dodatkowe usługi
- działaniu hotelu
 - obecni goście
 - historia gości
 - rezerwacje
 - historia rezerwacji
 - usługi wybrane przez gości
 - oceny wystawione przez gości

Wszystkie te informacje są zawarte w poniższych tabelach:

- employees
 - employee_id PK
 - name VARCHAR(30) not null
 - surname VARCHAR(30) not null
 - department_id FK
 - employment_date DATE
- emp_schedule
 - shift_id PK unique
 - employee_id FK
 - date DATE not null
- departments
 - department_id PK
 - name VARCHAR(30) unique not null
- current_guests
 - guest_id PK
 - name VARCHAR(30)
 - surname VARCHAR(30) not null
- reservations
 - reservation_id PK
 - guest_id FK unique not null
 - room_id FK unique not null
 - stay_length integer not null

- rooms
 - room_id (room number) PK
 - people_capacity integer not null
 - cost integer not null
 - occupied boolean not null
- extra_service
 - service_id PK
 - name VARCHAR(30) unique not null
 - cost Integer not null
- guest_service
 - guest_id FK not null
 - service_id FK not null
 - quantity Integer not null
- renovations_history
 - renovation_id PK
 - description VARCHAR(255) not null
 - renovation_year DATE
- reservation_history (jak się tu doda to się usuwa z current stay, więc nie mogą być fk)
 - reservation_id PK
 - guest_id not null
 - room_id not null
 - stay_length Integer not null
- guest_history
 - guest_id
 - name
 - surname
- budget
 - budget_id PK
 - year DATE not null
 - budget Integer not null
- reviews
 - review_id PK
 - guest_id FK (do guest_history) not null
 - review_description Integer (niech potem ktoś zrobi wyzwalacz czy coś że sprawdza czy jest od 1 do 5)

Po zaplanowaniu struktury encji i atrybutów stworzyliśmy modele (konceptyjny i relacyjny).
Prezentują się one następująco:



Aby baza danych działała poprawnie oraz żeby dało się obsłużyć zapytania, stworzyliśmy następujące elementy:

- funkcje:
 - `getCostForGuest(guest_id_param current_guests.guest_id%TYPE)` - jest to funkcja obliczająca cenę za pobyt w hotelu dla podanego gościa na podstawie jego rezerwacji
 - `number_of_rooms_occupied` - zwraca liczbę zajętych pokoi w hotelu
- procedury:
 - `show_employees` - wyświetla dane o pracownikach hotelu
 - `show_shifts(employee_id employees.employee_id%TYPE)` - wyświetla zmiany dla podanego pracownika
- wyzwalacze:
 - `add_reservation` - po dodaniu rezerwacji, zmienia pole "occupied" w tabeli ROOMS w zarezerwowanym pokoju na 1
 - `add_to_guest_history` - po usunięciu gościa z tabeli CURRENT_GUESTS dodaje go do tabeli GUEST_HISTORY
 - `add_to_reservation_history` - po usunięciu rezerwacji z tabeli RESERVATIONS, dodaje ją do RESERVATIONS_HISTORY oraz zmienia atrybut danego pokoju "occupied" na 0
 - `check_add_to_guest_history` - jeśli próbujemy dodać gościa do tabeli GUEST_HISTORY, a on już tam jest, rzucany jest wyjątek
 - `check_delete_rooms` - przy próbie usunięcia pokoju, który aktualnie jest zarezerwowany rzuca błąd
 - `extra_service_check` - rzuca błąd przy próbie zmiany lub dodania kosztu ujemnego kosztu usługi
 - `review_description_check` - rzuca wyjątek przy próbie dodania oceny do tabeli REVIEWS, która ma review_desciptions spoza przedziału 1 i 5
- kursory, które są wykorzystywane w powyższych elementach

Na sam koniec stworzyliśmy prostą, konsolową aplikację, która pozwala użytkownikowi na korzystanie z bazy danych i przeprowadzenie symulacji działania hotelu na podstawowym poziomie.