Kamil Winczewski 245958

Martyna Szymańska 245938

**Temat projektu**

System rezerwacji hoteli

**Cel projektu**

Celem projektu było stworzenie systemu rezerwacji dla hoteli, który umożliwi:

* Zarządzanie rezerwacjami pokoi przez pracowników hotelu.
* Ułatwienie weryfikacji dostępności pokoi i ich rezerwacji dla gości.
* Obsługę płatności oraz zarządzanie wydarzeniami hotelowymi.

**Wymagania projektu**

**Podstawowe funkcjonalności:**

* Zarządzanie rezerwacjami pokoi, w tym sprawdzanie dostępności.
* Obsługa płatności za rezerwacje.
* Możliwość zarządzania wydarzeniami hotelowymi oraz zapisów na nie.

**Bezpieczeństwo:**

* Wprowadzenie ról użytkowników i przypisanie minimalnych uprawnień.
* Ustawienie Modelu odzyskiwania danych na FULL RECOVERY.

**Dane i walidacje:**

* Pełnoletniość klientów dokonujących rezerwacji.
* Zgodność wprowadzanych danych (np. format e-mail, numer dokumentu).
* Brak możliwości rezerwacji jednego pokoju przez wielu klientów w tym samym czasie.

**Opis założeń**

* Klient może mieć wiele rezerwacji
* Nie można zarezerwować jednego pokoju w jednym czasie przez 2 i więcej klientów
* Hotel może mieć więcej niż jednego managera
* Jeśli anulowana została płatność, status rezerwacji zmienia się na „Canceled”
* Klienci rezerwujący pokój muszą być pełnoletni
* Wprowadzone dane klientów, takie jak numer dokumentu tożsamości, muszą być zgodne z odpowiednim formatem.

**Operacje na tabelach**

* **sp\_InsertHotel** – dodawanie hoteli.
* **sp\_InsertRoom** – dodawanie pokoi z walidacją danych (np. cena > 0).
* **sp\_InsertClient**: Procedura sprawdza poprawność e-maila oraz numeru telefonu przed wstawieniem danych klienta. Walidacja obejmuje:
  + E-mail w formacie @ i domeny końcowej.
  + Numer telefonu zawierający tylko cyfry i mieszczący się w zakresie 9-15 znaków.
* **sp\_InsertReservation** – obsługa rezerwacji z walidacją dostępności pokoi.
* **sp\_InsertPayment** – przypisywanie płatności do rezerwacji.
* **trg\_UpdateModifiedDate**– automatyczna aktualizacja pola **ModifiedDate** w tabeli **Hotels** po każdej modyfikacji danych.
* **trg\_UpdateTotalRooms** – automatyczna aktualizacja liczby pokoi w hotelu po dodaniu lub usunięciu pokoju w tabeli **Rooms**.
* **trg\_CheckRoomAvailability** – walidacja dostępności pokoju przed wstawieniem nowej rezerwacji:
  + Uniemożliwia rezerwację pokoju, który jest już zajęty w danym terminie.
  + Wyświetla odpowiedni komunikat błędu w przypadku konfliktu.
* **trg\_UpdatePaymentStatus** – automatyczna zmiana statusu rezerwacji na „Paid” po dokonaniu płatności w tabeli **Payments**.

**Przykładowe zapytania**

* Wyświetlenie dostępnych pokoi w danym terminie.
* Liczba rezerwacji dla każdego klienta.
* Liczba uczestników wydarzenia
* Liczba rezerwacji konkretnego pokoju
* Sprawdzanie poprawności dotyczące ról i grup użytkowników

**Co udało się zrealizować**

1. **Struktura bazy danych:**
   * Wszystkie tabele zostały znormalizowane do 3NF.
   * Dodano odpowiednie klucze główne i obce, a także ograniczenia takie jak CHECK i UNIQUE.
   * Stworzono pełną strukturę bazy danych z mechanizmami walidacji i wyzwalaczami.
   * Obsłużono podstawowe funkcjonalności systemu rezerwacji oraz płatności.
   * Wykonano backup bazy danych i przygotowano skrypt jej przywracania.
2. **Mechanizmy bezpieczeństwa:**
   * Utworzono loginy, użytkowników i przypisano role zgodnie z zasadami minimalnych uprawnień.
   * Przeprowadzono testy weryfikujące działanie przypisanych uprawnień.
3. **Backup i odzyskiwanie danych:**
   * Zrealizowano pełny backup bazy danych oraz dostarczono skrypt do jej przywracania.
4. **Walidacja danych:**
   * Stworzono zestaw zapytań do weryfikacji integralności danych, takich jak sprawdzenie wieku klientów, poprawności adresów e-mail oraz unikania konfliktów w rezerwacjach.
5. **Funkcjonalność:**
   * Obsłużono scenariusze, takie jak zarządzanie rezerwacjami, wydarzeniami oraz przypisywanie płatności.

**Największe wyzwania**

1. **Zarządzanie ograniczeniami:**
   * Złożoność walidacji danych, takich jak poprawne daty rezerwacji czy brak konfliktów między rezerwacjami, wymagała zaawansowanego SQL.
2. **Testowanie ról i uprawnień:**
   * Zapewnienie, że każda rola ma dokładnie tyle uprawnień, ile potrzeba, wymagało wielokrotnych testów.
3. **Wydajność zapytań:**
   * Przy większej liczbie danych wydajność niektórych zapytań, takich jak sprawdzanie dostępnych pokoi, może być wyzwaniem.

**Możliwości rozbudowy**

1. **Moduł raportowania:**
   * Dodanie zaawansowanych raportów analitycznych (np. liczba rezerwacji według regionów, najbardziej popularne wydarzenia).
2. **Automatyzacja powiadomień:**
   * System powiadomień e-mail lub SMS dla klientów (np. przypomnienia o rezerwacjach).
3. **Zarządzanie sezonowością:**
   * Dodanie funkcji sezonowych cen pokoi lub rabatów dla klientów.
4. **Integracja z systemami zewnętrznymi:**
   * Synchronizacja z systemami płatności online lub platformami rezerwacyjnymi.
5. **Rozbudowa o nowe moduły:**
   * Moduł obsługi restauracji w hotelach lub zarządzania pracownikami.

**Podsumowanie**

Projekt bazy danych został zrealizowany z uwzględnieniem podstawowych zasad normalizacji do trzeciej postaci normalnej (3NF). Stworzona struktura obejmuje kluczowe tabele, takie jak hotele, pokoje, klienci, rezerwacje, wydarzenia oraz płatności, co zapewnia spójność i minimalizuje redundancję danych. Wdrożono również mechanizmy zabezpieczeń, w tym role użytkowników z odpowiednimi uprawnieniami oraz przeprowadzono pełny backup bazy danych.