**Wojskowa Akademia Techniczna**

*Sprawozdanie z projektu zaliczeniowego*

Baza danych hotelu

Wykonała: Julia Słowińska



1. Dane dotyczące zadania

*Wykonać bazę danych na wybrany temat (hotel). Baza powinna spełniać następujące wymagania:*

*-minimum 10 tabel,*

*-minimum 3 perspektywy z czego jedna ma być funkcyjna i jedna zmaterializowana*

*-minimum 3 trigery*

*-minimum 3 procedury wbudowane*

*-minimum 3 funkcje wbudowane*

*-użycie mechanizmów typu autoinkrementacji*

*-minimum 1 własny index*

* 1. *Analiza*

*Utworzona baza danych ma przede wszystkim mieć na celu ułatwienie kierownictwu kontroli i aktualizacji informacji dotyczących wszystkich pracowników, w tym informacji potrzebnych do podpisywania umów i wypłacania wynagrodzenia. Baza danych ma także usprawnić proces rezerwacji pokoi i usług przez pracowników, ponieważ może być używana do śledzenia stanu rezerwacji i dokonywania zmian, jeśli zajdzie taka potrzeba. Ma również zapisywać dane o gościach i ich rezerwacjach*

Model konceptualny bazy:

**

* 1. Dane dotyczące bazy:

*Baza została zaimplementowana w MSSQL oraz MySQL*

*Ta baza danych jest przeznaczona do przechowywania informacji związanych z hotelami, gośćmi, rezerwacjami, usługami i wyposażeniem. Zawiera tabele dla RoomType, Equipment, RoomTypeEquipment, RoomLocation, Hotel, Service, ExecutionRealization, Room, Role, Employee, GuestType, Guest, BookingRealization, Booking, BookingDetails i Execution.*

*Tabela RoomType przechowuje informacje o różnych typach pokoi, takie jak opis rodzaju pokoju i liczbę osób, które może pomieścić pokój.*

*Tabela Wyposażenie zawiera informacje o różnych typach wyposażenia hotelu, takie jak nazwa wyposażenia, marka, data zakupu oraz rodzaj łóżek i toalet.*

*Tabela RoomTypeEquipment przechowuje informacje o tym, jakie typy pomieszczeń mają jakie wyposażenie.*

*Tabela RoomLocation przechowuje informacje o różnych lokalizacjach pokoi w hotelu.*

*Tabela Hotel przechowuje informacje o różnych hotelach, takie jak nazwa, adres i liczba gwiazdek.*

*Tabela Usług zawiera informacje o różnych dostępnych usługach, takie jak opis usługi i koszt usługi.*

*Tabela ExecutionRealization przechowuje informacje o różnych typach realizacji usług takie jak opis i termin realizacji.*

*Tabela Room przechowuje informacje o różnych pokojach, takie jak numer pokoju, typ pokoju, lokalizacja pokoju, hotel, cena pokoju i zajęcie.*

*Tabela Role przechowuje informacje o różnych rolach pracowników, takie jak tytuł roli i opis pokoju.*

*Tabela Emplooy przechowuje informacje o różnych pracownikach, takie jak identyfikator hotelu, identyfikator roli, imię, nazwisko, wynagrodzenie i data zatrudnienia.*

*Tabela GuestType przechowuje informacje o różnych typach gości, takie jak opis typu gościa.*

*Tabela Guest przechowuje informacje o różnych gościach, takie jak identyfikator typu gościa oraz imię i nazwisko gościa.*

*W tabeli BookingRealization przechowywane są informacje o różnych rodzajach rezerwacji takie jak opis realizacji rezerwacji oraz data realizacji rezerwacji.*

*Tabela Reservation przechowuje informacje o różnych rezerwacjach, takie jak identyfikator gościa, identyfikator realizacji rezerwacji i datę rezerwacji.*

*Tabela BookingDetails przechowuje informacje o różnych szczegółach rezerwacji, takich jak identyfikator szczegółów rezerwacji, identyfikator rezerwacji, numer pokoju, data zameldowania, data wymeldowania, cena zakupu i liczba nocy.*

*Tabela Execution przechowuje informacje o wykonaniu różnych usług takie jak ID usługi, ID szczegółów rezerwacji, ID realizacji, data realizacji.*

Model fizyczny bazy:

**

*Cel:*

*Podstawowym celem tej bazy danych jest przechowywanie informacji związanych z rezerwacjami hotelowymi, takich jak rodzaje pokoi, wyposażenie, lokalizacje, nazwy hoteli, usługi i role pracowników. Dodatkowo w bazie przechowywane są również informacje o gościach i ich rezerwacjach, a także szczegóły każdej rezerwacji, takie jak numer pokoju, liczba noclegów i cena zakupu. Wreszcie w bazie przechowywane są dane związane z realizacją usług, w tym identyfikator usługi, identyfikator szczegółów rezerwacji, identyfikator realizacji. Wszystkie te dane zostaną wykorzystane w celu umożliwienia kierownictwu hotelu śledzenia rezerwacji, usług i innych informacji związanych z hotelem.*

1. Widoki

*2.1* CorporateGuestViews

*Widok jest wirtualną tabelą, która jest pochodną danych dwóch istniejących tabel "Guest" i "Booking". Widok zawiera trzy kolumny "FirstName", "LastName" i "BookingDate". Dane dla widoku są uzyskiwane poprzez połączenie dwóch tabel na kolumnie "GuestID". Widok wyświetla tylko rekordy, gdzie "GuestTypeID" jest równy 1.*

Kod SQL widoku:

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

Po zrealizowaniu zapytania

**

Ukazują nam się następujące dane

*Obraz zawierający stół

Opis wygenerowany automatycznie*

*2.2 BookedRooms*

*Komenda "CREATE or ALTER VIEW" tworzy lub modyfikuje widok w bazie danych. Widok "BookedRooms" jest tworzony z włączeniem "SCHEMABINDING", co oznacza, że jego schemat jest związany z bazą danych i nie może być zmieniony bez modyfikacji samego widoku. Widok jest tworzony na podstawie danych z dwóch tabel: "Room" i "BookingDetails", w których dane są łączone na podstawie kolumny "RoomNumber". Widok prezentuje dane o numerach pokojów i liczbie ich rezerwacji. Dane są grupowane wg numerów pokojów. Następnie, komenda "CREATE UNIQUE CLUSTERED INDEX" tworzy unikalny, skupiony indeks o nazwie "IDX\_V1" na widoku "dbo.BookedRooms", oparty na kolumnie "RoomNumber". Indeks ten usprawnia wyszukiwanie i sortowanie danych w widoku.*

Kod SQL widoku:

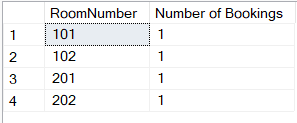
*Obraz zawierający tekst

Opis wygenerowany automatycznie*

Po zrealizowaniu zapytania

**

Ukazują nam się następujące dane



*2.3* GuestBookingBucket

*Komenda "CREATE or ALTER VIEW" tworzy lub modyfikuje widok "GuestBookingBucket". Widok jest tworzony na podstawie danych z dwóch tabel "Guest" i "Booking". Widok zawiera informacje o imieniu i nazwisku gościa oraz jego kategorii, która jest określana za pomocą instrukcji "CASE" na podstawie liczby rezerwacji (pobranych z tabeli "Booking") dla danego gościa (pobranych z tabeli "Guest"). Jeśli dany gość jeszcze nie dokonał żadnej rezerwacji, kategoria będzie wynosić "Not a guest yet". Jeśli gość dokonał mniej niż 3 rezerwacji, kategoria będzie wynosić "Up to 3 bookings". W przeciwnym wypadku, kategoria będzie wynosić "Regular client". Dane w widoku są grupowane wg imienia i nazwiska gościa.*

Kod SQL widoku:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Po zrealizowaniu zapytania

**

Ukazują nam się następujące dane

*Obraz zawierający stół

Opis wygenerowany automatycznie*

1. Wyzwalacze (Triggery)

*3.1* *NumberOfNights*

*Komenda "CREATE TRIGGER" tworzy wyzwalacz o nazwie "NumberOfNights" na tabeli "BookingDetails". Wyzwalacz jest aktywowany przy wstawianiu lub aktualizacji danych w tabeli "BookingDetails". W ciele wyzwalacza, komenda "UPDATE" aktualizuje kolumnę "NumberOfNights" w tabeli "BookingDetails" na podstawie różnicy pomiędzy datami "CheckInDate" i "CheckOutDate". Wszystkie aktualizacje są wykonywane w obrębie tabeli "BookingDetails". Wyzwalacz będzie automatycznie obliczać liczbę nocy, jakie gość spędzi w pokoju, w oparciu o daty "CheckInDate" i "CheckOutDate", i zapisywać ją w kolumnie "NumberOfNights" w tabeli "BookingDetails".*

*Kod SQL wyzwalacza:*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Po zrealizowaniu zapytania*

**

*Ukazują nam się następujące dane:*

*Obraz zawierający stół

Opis wygenerowany automatycznie*

*3.2 NoSalaryupdate*

*Komenda "CREATE TRIGGER" tworzy wyzwalacz o nazwie "NoSalaryUpdate" na tabeli "Employee". Wyzwalacz jest aktywowany tylko podczas aktualizacji danych w tabeli "Employee". W ciele wyzwalacza, instrukcja "IF UPDATE (salary)" sprawdza, czy aktualizowana jest kolumna "salary". Jeśli tak, wyzwalacz wykonuje instrukcję "ROLLBACK", co unieważnia transakcję aktualizacji i zwraca komunikat błędu "Salary update not allowed" z poziomem ostrzeżenia 16 i numerze błędu 1. Wyzwalacz zapobiega aktualizacji kolumny "salary" w tabeli "Employee", uniemożliwiając modyfikację wynagrodzenia pracownika.*

*Kod SQL wyzwalacza:*

*Obraz zawierający tekst

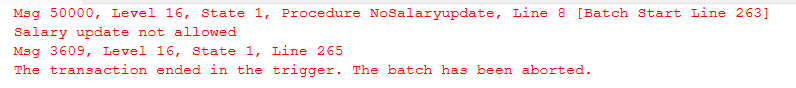
Opis wygenerowany automatycznie*

*Po zrealizowaniu zapytania*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Ukazują nam się następujące dane:*

**

*3.3 LastbookingDate*

*Komenda "CREATE TRIGGER" tworzy wyzwalacz o nazwie "LastBookingDate" na tabeli "Booking". Wyzwalacz jest aktywowany podczas operacji wstawiania, aktualizacji lub usuwania danych w tabeli "Booking". W ciele wyzwalacza, instrukcja "UPDATE" aktualizuje kolumnę "LastBookingDate" w tabeli "Guest", ustawiając ją na najnowszą datę rezerwacji dla danego gościa. Jest to osiągane poprzez dołączenie tabeli "Guest" z subzapytaniem, które zwraca dla każdego gościa najnowszą datę rezerwacji z tabeli "Booking". Wyzwalacz jest wykorzystywany do aktualizacji informacji o ostatniej rezerwacji gościa po każdej operacji na tabeli "Booking".*

*Kod SQL wyzwalacza:*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Po zrealizowaniu któregoś z tych zapytań*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Ukazują nam się następujące dane:*

Obraz zawierający stół

Opis wygenerowany automatycznie

1. *Index*

**

1. *Procedury wbudowane*

*5.1 InsertNewGuest*

*Procedura służąca do wstawiania nowego gościa do bazy danych. Procedura składa się z trzech głównych bloków kodu:*

1. *Ustawienie NOCOUNT ON: ustawia opcję NOCOUNT na ON, co oznacza, że nie będzie wyświetlanych dodatkowych wyników zapytania.*
2. *Blok transakcji: blok transakcji jest używany do zapewnienia integralności danych w przypadku wystąpienia błędu w trakcie wykonywania zapytania. W bloku transakcji znajdują się dwie części:*

*a. Blok TRY: główny kod, który jest wykonywany. W tym przypadku jest to instrukcja INSERT do tabeli "Guest".*

*b. Blok CATCH: w przypadku wystąpienia błędu w bloku TRY, kod w bloku CATCH zostanie wykonany. W tym przypadku jest to instrukcja ROLLBACK, która anuluje wszystkie zmiany wprowadzone przez instrukcje SQL w bloku TRY. Procedura wstawia nowego gościa do tabeli "Guest" z wykorzystaniem parametrów @FirstName, @LastName i @GuestType. Zmiany są automatycznie wstrzymywane i anulowane w przypadku wystąpienia błędu.*

*Kod SQL procedury:*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Po wykonaniu zapytania*

**

*I wyświetleniu tabeli Guest ukazuje nam się rezultat:*

*Obraz zawierający stół

Opis wygenerowany automatycznie*

*5.2* ListOfEquipment

*Ta procedura składowana pobiera i wyświetla informacje z tabeli "Equipment" w bazie danych. Procedura używa pętli, aby iterować przez każdy EquipmentID w tabeli, a dla każdej iteracji wykonuje instrukcję SELECT, aby pobrać i wyświetlić odpowiadający rekord z tabeli "Equipment".*

*Kod SQL procedury:*

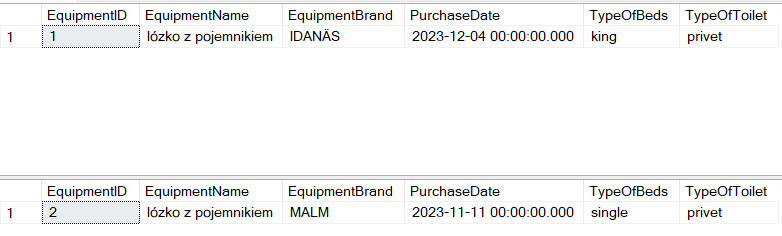
*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Po wykonaniu zapytania*

**

*Ukazuje nam się rezultat:*

**

*5.3 RoomPriceRise*

*Ta procedura składowana przyjmuje jeden parametr wejściowy o nazwie kwota i umożliwia podniesienie ceny pokoju w bazie danych. Wewnątrz procedury wykonywana jest instrukcja UPDATE, aby zaktualizować wszystkie rekordy w tabeli "Room", dodając wartość kwota do każdej z obecnych cen pokoju.*

*Kod SQL procedury:*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Po wykonaniu zapytania*

**

**

*Ukazuje nam się rezultat:*

*Obraz zawierający stół

Opis wygenerowany automatycznie*

1. *Funkcje wbudowane*

*6.1 isoccupied*

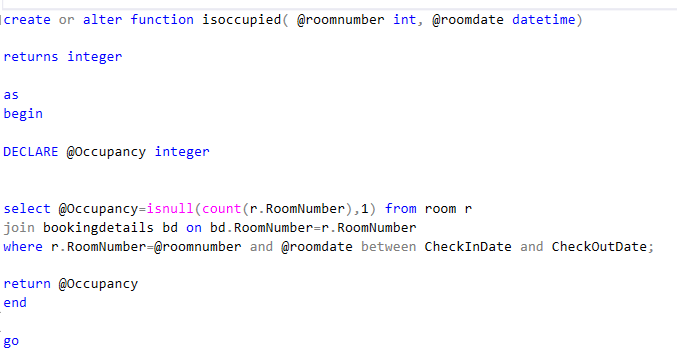
*Kod tworzy lub modyfikuje funkcję w bazie danych SQL Server o nazwie "isoccupied".*

*Funkcja przyjmuje dwa parametry jako wejście:*

*roomnumber: liczba całkowita reprezentująca numer pokoju roomdate: datetime reprezentujący datę Funkcja zwraca liczbę całkowitą, która wskazuje, czy pokój jest zajęty w danym dniu. Funkcja wykorzystuje polecenie SELECT do zliczenia rezerwacji dla danego numeru pokoju i daty. Jeśli nie ma rezerwacji, zwraca 1. Rezultat zliczenia jest przechowywany w zmiennej Occupancy.*

*Na końcu funkcja zwraca zmienną Occupancy.*

*Kod SQL funkcji:*

**

*Po wykonaniu zapytania*

**

*Ukazuje nam się rezultat:*

**

*6.2 bookedbyguest*

*tworzy lub modyfikuje funkcję o nazwie „bookedroomsbyguest”, która przyjmuje jeden argument - identyfikator gościa - i zwraca ciąg znaków (VARCHAR (1000)). Kod utworzy kursor zapytania dla wszystkich rezerwacji dla tego gościa. Kursor pobiera kolejne elementy z kolumny numeru pokoju i zapisuje je do zmiennej @Bookedrooms jako ciąg znaków oddzielonych przecinkami, a następnie zwraca go jako wynik funkcji.*

*Kod SQL funkcji:*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Po wykonaniu zapytania*

**

*Ukazuje nam się rezultat:*

**

*6.3* employesalary

Utworzenie lub zmienienie funkcji o nazwie „employeesalary”, która przyjmuje identyfikator pracownika jako parametr i zwraca jego wynagrodzenie jako liczbę całkowitą. Kod zaczyna się od zadeklarowania zmiennej o nazwie „wynagrodzenie”, a następnie wysyła zapytanie do tabeli Pracownik, aby znaleźć pracownika o podanym identyfikatorze i przypisać jego wynagrodzenie do zmiennej. Na koniec funkcja zwraca wartość zmiennej wynagrodzenia.

*Kod SQL funkcji:*

*Obraz zawierający tekst

Opis wygenerowany automatycznie*

*Po wykonaniu zapytania*

**

*Ukazuje nam się rezultat:*

**