SYSTEM REZERWACJI HOTELOWEJ

*DOKUMENTACJA BAZY DANYCH*



Skład grupy:

Natalia Owczarek  
Dominika Sybilska  
Barbara Wolska  
Olga Sadokierska  
Izabela Najder

Spis treści

[1. Opis tła projektowanego i implementowanego zagadnienia 3](#_Toc129686240)

[2. Opis wymagań funkcjonalnych i niefunkcjonalnych 3](#_Toc129686241)

[3. Model fizyczny lub schemat bazy danych w postaci graficznej 3](#_Toc129686242)

[4. Skrypt tworzenia bazy danych 3](#_Toc129686243)

[5. Wykaz utworzonych tabel fizycznych 3](#_Toc129686244)

[6. Skrypt tworzenia tabel i referencji między nimi 3](#_Toc129686245)

[7. Wykaz zaimplementowanych indeksów 3](#_Toc129686246)

[8. Skrypt tworzenia indeksów 3](#_Toc129686247)

[9. Skrypt ładujący przykładowe dane do tabel 3](#_Toc129686248)

[10. Wykaz utworzonych widoków 3](#_Toc129686249)

[11. Skrypt tworzenia widoków 3](#_Toc129686250)

[12. Wykaz utworzonych procedur składowanych 3](#_Toc129686251)

[13. Skrypt tworzenia procedur składowanych 3](#_Toc129686252)

[14. Wykaz utworzonych funkcji użytkownika 3](#_Toc129686253)

[15. Skrypt tworzenia funkcji definiowanych przez użytkownika 3](#_Toc129686254)

[16. Wykaz utworzonych wyzwalaczy 3](#_Toc129686255)

[17. Skrypt tworzenia wyzwalaczy 3](#_Toc129686256)

[18. Wykaz obiektów wartości domyślnej i reguł 3](#_Toc129686257)

[19. Skrypt tworzenia obiektów wartości domyślnej i reguł 3](#_Toc129686258)

[20. Wykaz typów danych tworzonych przez użytkownika 3](#_Toc129686259)

[21. Skrypt tworzenia typów danych definiowanych przez użytkownika 3](#_Toc129686260)

[22. Schemat zaprojektowanego systemu przedstawiający punkty wejścia i wyjścia 3](#_Toc129686261)

[23. Wnioski i podsumowanie 3](#_Toc129686262)

[24. Wykaz/spis ilustracji 3](#_Toc129686263)

[25. Wykaz/spis tabel 3](#_Toc129686264)

# Opis tła projektowanego i implementowanego zagadnienia

Przedmiotem naszego projektu jest sieć hoteli o nazwie „Hotel D&B&N”, która dąży do osiągnięcia wysokiej pozycji na rynku hotelarskim. Przedstawiona w projekcie sieć hotelarska posiada swoje placówki w kilku lokalizacjach na terenie Polski. „Hotel D&B&N” dba o satysfakcje klientów, dlatego w swojej ofercie posiada różnorodne pokoje, takie jak studio, wieloosobowe, czy Superior. System bazy danych oferuje możliwość rezerwacji pokoju w wybranej lokacji i dacie. System zawiera również informacje na temat pracowników sieci hoteli.

Baza danych zawiera kompleksowe informacje na temat placówek hotelu oraz dostępnych pokoi wraz z ich statusem i typem. System pozwala na dodawanie nowych rezerwacji i przechowywanie informacji na temat zrealizowanych i odwołanych rezerwacji. System zawiera również wszelkie informacje na temat gości oraz ich płatności. System pomaga śledzić transakcje, dzięki zbieraniu danych na temat płatności oraz sposobach płatności, dzięki czemu umożliwia szczegółowe kontrolowanie i tworzenie danych do faktur.

Wdrożyliśmy funkcjonalność umożliwiającą składanie rezerwacji z wyprzedzeniem, a także przypisywanie wielu rezerwacji do wielu klientów. Ponadto, istnieje możliwość zbierania informacji potrzebnych do wystawienia faktur, co usprawni proces rozliczeń i ułatwi obsługę klientów. Dzięki tym elementom pracownik jest w stanie lepiej zarządzać rezerwacjami i świadczyć jeszcze lepsze usługi dla klientów.

Wszystkie elementy wymienione powyżej są integralnymi częściami bazy danych sieci hoteli "Hotel D&B&N", która działa na potrzeby zarządzania hotelem, zapewniając kompleksową obsługę klientów oraz usprawniając procesy wewnętrzne.

# Opis wymagań funkcjonalnych i niefunkcjonalnych

Wymagania funkcjonalne:

* Dodawanie rezerwacji
* Dodawanie wypożyczenia
* Dodawanie faktury
* Modyfikacja faktury
* Usunięcie faktury
* Dodawanie klientów
* Modyfikacja klienta
* Dodawanie pokoju
* Modyfikacja pokoju
* Usunięcie pokoju
* Dodanie płatności
* Modyfikacja płatności
* Dodawanie klientów
* Modyfikacja klientów

Wymagania niefunkcjonalne:

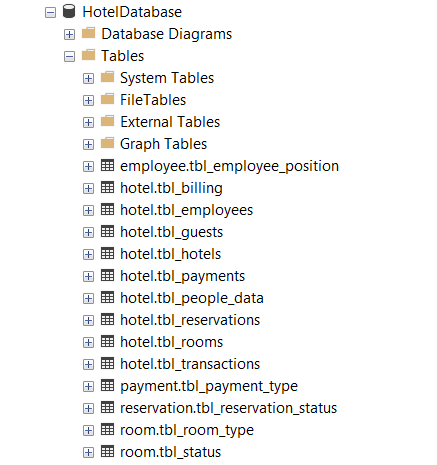
* Brak możliwości usunięcia placówki hotelu
* Brak możliwości usunięcia rezerwacji pokoju
* Nie ma możliwości usunięcia faktury z bieżącego roku
* Nie można zarezerwować już zajętego pokoju
* Nie można usunąć płatności z bieżącego roku

# Model fizyczny lub schemat bazy danych w postaci graficznej

# Skrypt tworzenia bazy danych

CREATE DATABASE HotelDatabase;

# Wykaz utworzonych tabel fizycznych



# Skrypt tworzenia tabel i referencji między nimi

USE HotelDatabase

GO

CREATE SCHEMA hotel

GO

CREATE SCHEMA room

GO

CREATE SCHEMA employee

GO

CREATE SCHEMA payment

GO

CREATE SCHEMA reservation

GO

USE HotelDatabase

GO

CREATE TABLE room.tbl\_status

(

IdRoomStatus int NOT NULL IDENTITY(1,1) PRIMARY KEY,

RoomStatusName varchar(20) NOT NULL

)

ON Dictionaries;

GO

CREATE TABLE room.tbl\_room\_type

(

IdRoomType int NOT NULL IDENTITY(1,1) PRIMARY KEY,

RoomName varchar(30) NOT NULL,

MaxNumberOfGuests int NOT NULL,

RoomPrice decimal NOT NULL,

)

ON Dictionaries;

CREATE TABLE employee.tbl\_employee\_position

(

IdEmployeePosition int NOT NULL IDENTITY(1,1) PRIMARY KEY,

PositionName varchar(18) NOT NULL,

)

ON Dictionaries;

CREATE TABLE payment.tbl\_payment\_type

(

IdPaymentType int NOT NULL IDENTITY(1,1) PRIMARY KEY,

PaymentName varchar(20) NOT NULL,

)

ON Dictionaries;

CREATE TABLE reservation.tbl\_reservation\_status

(

IdReservationStatus int NOT NULL IDENTITY(1,1) PRIMARY KEY,

StatusName varchar(20) NOT NULL,

)

ON Dictionaries;

CREATE TABLE hotel.tbl\_hotels

(

IdHotel int NOT NULL IDENTITY(1,1) PRIMARY KEY,

ZipCode varchar(6) NOT NULL CHECK (ZipCode LIKE '[0-9][0-9]-[0-9][0-9][0-9]'),

NumberOfTheBuilding varchar(6) NOT NULL,

Street varchar(30) NULL,

City varchar(30) NOT NULL,

HotelName varchar(30) NOT NULL,

LandlinePhoneNumber varchar(9) NULL CHECK (LandlinePhoneNumber LIKE '([1-9][0-9]{1,2}) [0-9]{3} [0-9]{3}'),

MobilelinePhoneNumber varchar(9) NULL CHECK (MobilelinePhoneNumber LIKE '48[5-8][0-9]{8}'),

Email varchar(30) NULL,

);

GO

CREATE TABLE hotel.tbl\_rooms

(

IdRoom int NOT NULL IDENTITY(1,1) PRIMARY KEY,

IdRoomType int NOT NULL FOREIGN KEY REFERENCES room.tbl\_room\_type(IdRoomType) ON UPDATE CASCADE,

IdRoomStatus int NOT NULL FOREIGN KEY REFERENCES room.tbl\_status(IdRoomStatus) ON UPDATE CASCADE,

RoomNumber int NOT NULL,

FloorNumber int NOT NULL,

IdHotel int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_hotels(IdHotel) ON UPDATE CASCADE,

);

GO

CREATE TABLE hotel.tbl\_reservations

(

IdReservation int NOT NULL IDENTITY(1,1) PRIMARY KEY,

IdRoom int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_rooms(IdRoom),

CheckInDate date NOT NULL,

CheckOutDate date NOT NULL, --CHECK (CheckOutDate >= CheckInDate)

NumberOfGuests varchar(3) NOT NULL,

IdReservationStatus int NOT NULL FOREIGN KEY REFERENCES reservation.tbl\_reservation\_status(IdReservationStatus),

IdHotel int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_hotels(IdHotel),

);

GO

CREATE TABLE hotel.tbl\_people\_data

(

IdPersonData int NOT NULL IDENTITY(1,1) PRIMARY KEY,

ZipCode varchar(6) NOT NULL CHECK (ZipCode LIKE '[0-9][0-9]-[0-9][0-9][0-9]'),

ApartmentNumber varchar(4) NULL,

NumberOfTheBuilding varchar(6) NOT NULL,

Street varchar(30) NULL,

City varchar(30) NOT NULL,

Country varchar(30) NOT NULL,

DateOfBirth date NOT NULL CHECK ((YEAR(GETDATE()) - YEAR(DateOfBirth)) >= 18),

MobilelinePhoneNumber varchar(9) NULL CHECK (MobilelinePhoneNumber LIKE '48[5-8][0-9]{8}'),

Email varchar(30) NULL,

);

GO

CREATE TABLE hotel.tbl\_guests

(

IdGuest int NOT NULL IDENTITY(1,1) PRIMARY KEY,

IdPersonData int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_people\_data(IdPersonData) ON UPDATE CASCADE,

[Name] varchar(15) NOT NULL,

LastName varchar(30) NOT NULL,

IDNumber varchar(30) NOT NULL CHECK (IDNumber LIKE '[A-Z][A-Z][A-Z][0-9][0-9][0-9][0-9][0-9][0-9]') UNIQUE,

PassportNumber varchar(9) NOT NULL CHECK (PassportNumber LIKE '[A-Z][A-Z][0-9][0-9][0-9][0-9][0-9][0-9]') UNIQUE,

);

GO

CREATE TABLE hotel.tbl\_employees

(

IdEmployee int NOT NULL IDENTITY(1,1) PRIMARY KEY,

IdPersonData int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_people\_data(IdPersonData) ON UPDATE CASCADE,

[Name] varchar(15) NOT NULL,

LastName varchar(30) NOT NULL,

IdEmployeePosition int NOT NULL FOREIGN KEY REFERENCES employee.tbl\_employee\_position(IdEmployeePosition) ON UPDATE CASCADE,

DateOfEmployment date NOT NULL CHECK (DateOfEmployment <= GETDATE()),

IdHotel int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_hotels(IdHotel) ON UPDATE CASCADE,

);

GO

CREATE TABLE hotel.tbl\_payments

(

IdPayment int NOT NULL IDENTITY(1,1) PRIMARY KEY,

IdPaymentType int NOT NULL FOREIGN KEY REFERENCES payment.tbl\_payment\_type(IdPaymentType) ON UPDATE CASCADE,

PaymentDate date NOT NULL,

);

GO

CREATE TABLE hotel.tbl\_transactions

(

IdTransaction int NOT NULL IDENTITY(1,1) PRIMARY KEY,

IdReservation int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_reservations(IdReservation),

IdPayment int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_payments(IdPayment),

IdEmployee int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_employees(IdEmployee),

IdGuest int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_guests(IdGuest),

IdHotel int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_hotels(IdHotel),

DataTransaction Date NOT NULL,

);

GO

CREATE TABLE hotel.tbl\_billing

(

IdBilling int NOT NULL IDENTITY(1,1) PRIMARY KEY,

IdGuest int NOT NULL FOREIGN KEY REFERENCES hotel.tbl\_guests(IdGuest) ON UPDATE CASCADE,

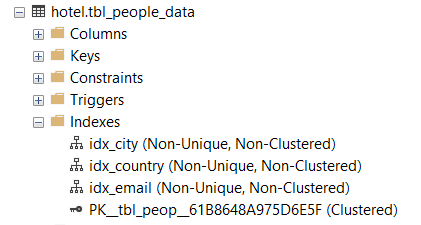
CompanyName varchar(30) NOT NULL,

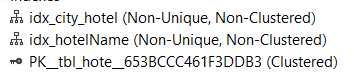
NIPNumber varchar(10) NOT NULL,

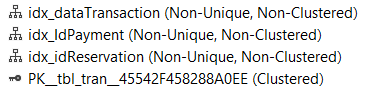
);

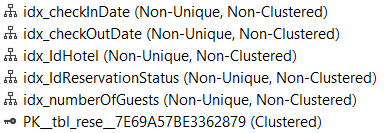
GO

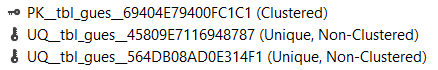
# Wykaz zaimplementowanych indeksów

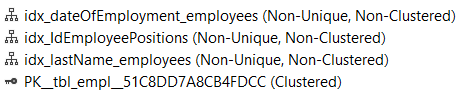


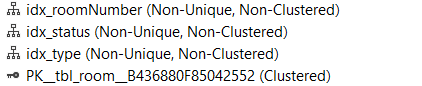


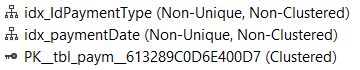


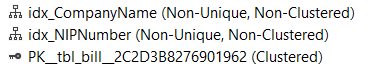












# Skrypt tworzenia indeksów

CREATE NONCLUSTERED INDEX [idx\_email] ON hotel.tbl\_people\_data

(

[Email] ASC

)

CREATE NONCLUSTERED INDEX [idx\_country] ON hotel.tbl\_people\_data

(

[Country] ASC

)

CREATE NONCLUSTERED INDEX [idx\_city] ON hotel.tbl\_people\_data

(

[City] ASC

)

CREATE NONCLUSTERED INDEX [idx\_city\_hotel] ON hotel.tbl\_hotels

(

[City] ASC

)

CREATE NONCLUSTERED INDEX [idx\_hotelName] ON hotel.tbl\_hotels

(

[HotelName] ASC

)

CREATE NONCLUSTERED INDEX [idx\_dataTransaction] ON hotel.tbl\_transactions

(

[DataTransaction] ASC

)

CREATE NONCLUSTERED INDEX [idx\_idReservation] ON hotel.tbl\_transactions

(

[IdReservation] ASC

)

CREATE NONCLUSTERED INDEX [idx\_IdPayment] ON hotel.tbl\_transactions

(

[IdPayment] ASC

)

CREATE NONCLUSTERED INDEX [idx\_checkInDate] ON hotel.tbl\_reservations

(

[CheckInDate] ASC

)

CREATE NONCLUSTERED INDEX [idx\_checkOutDate] ON hotel.tbl\_reservations

(

[CheckOutDate] ASC

)

CREATE NONCLUSTERED INDEX [idx\_numberOfGuests] ON hotel.tbl\_reservations

(

[NumberOfGuests] ASC

)

CREATE NONCLUSTERED INDEX [idx\_IdReservationStatus] ON hotel.tbl\_reservations

(

[IdReservationStatus] ASC

)

CREATE NONCLUSTERED INDEX [idx\_IdHotel] ON hotel.tbl\_reservations

(

[IdHotel] ASC

)

CREATE NONCLUSTERED INDEX [idx\_lastName] ON hotel.tbl\_guests

(

[LastName] ASC

)

CREATE NONCLUSTERED INDEX [idx\_lastName\_employees] ON hotel.tbl\_employees

(

[LastName] ASC

)

CREATE NONCLUSTERED INDEX [idx\_IdEmployeePositions] ON hotel.tbl\_employees

(

[IdEmployeePosition] ASC

)

CREATE NONCLUSTERED INDEX [idx\_dateOfEmployment\_employees] ON hotel.tbl\_employees

(

[DateOfEmployment] ASC

)

CREATE NONCLUSTERED INDEX [idx\_roomNumber] ON hotel.tbl\_rooms

(

[RoomNumber] ASC

)

CREATE NONCLUSTERED INDEX [idx\_status] ON hotel.tbl\_rooms

(

[IdRoomStatus] ASC

)

CREATE NONCLUSTERED INDEX [idx\_type] ON hotel.tbl\_rooms

(

[IdRoomType] ASC

)

CREATE NONCLUSTERED INDEX [idx\_paymentDate] ON hotel.tbl\_payments

(

[PaymentDate] ASC

)

CREATE NONCLUSTERED INDEX [idx\_IdPaymentType] ON hotel.tbl\_payments

(

[IdPaymentType] ASC

)

CREATE NONCLUSTERED INDEX [idx\_NIPNumber] ON hotel.tbl\_billing

(

[NIPNumber] ASC

)

CREATE NONCLUSTERED INDEX [idx\_CompanyName] ON hotel.tbl\_billing

(

[CompanyName] ASC

)

# Skrypt ładujący przykładowe dane do tabel

# Wykaz utworzonych widoków

# Skrypt tworzenia widoków

# Wykaz utworzonych procedur składowanych

# Skrypt tworzenia procedur składowanych

# Wykaz utworzonych funkcji użytkownika

# Skrypt tworzenia funkcji definiowanych przez użytkownika

# Wykaz utworzonych wyzwalaczy

# Skrypt tworzenia wyzwalaczy

# Wykaz obiektów wartości domyślnej i reguł

# Skrypt tworzenia obiektów wartości domyślnej i reguł

# Wykaz typów danych tworzonych przez użytkownika

# Skrypt tworzenia typów danych definiowanych przez użytkownika

# Schemat zaprojektowanego systemu przedstawiający punkty wejścia i wyjścia

# Wnioski i podsumowanie

Baza danych sieci hoteli "Hotel D&B&N" została zaprojektowana zgodnie z zasadami projektowania baz danych, które zapewniają efektywne i niezawodne funkcjonowanie systemu. Baza ta spełnia wszystkie wymagania i funkcjonalności, które zostały wymienione wcześniej. Dodatkowo, wypełniono ją przykładowymi danymi, co pozwala na przetestowanie jej działania i weryfikację poprawności wykonanych implementacji. Dzięki temu użytkownicy systemu mogą mieć pewność, że dane, które są w niej przechowywane, są dokładne i aktualne, co wpływa na jakość świadczonych usług i zadowolenie klientów.

Baza danych sieci hoteli "Hotel D&B&N" została wyposażona w wiele widoków, które usprawniają proces analizy danych klientów, rezerwacji oraz pokoi. Dzięki nim użytkownicy mogą łatwo przeglądać i monitorować dane z różnych perspektyw, co przyspiesza podejmowanie decyzji biznesowych. Ponadto, baza ta umożliwia dodawanie i modyfikowanie danych związanych z klientami, rezerwacjami, pracownikami i placówkami hotelu, co zwiększa elastyczność i dostosowanie systemu do potrzeb użytkowników. Baza ta posiada także wiele funkcjonalności ułatwiających pracę z danymi, w tym zarządzanie fakturami i płatnościami, co usprawnia proces rozliczeń i ułatwia obsługę klientów.

Projekt sieci hoteli został stworzony z myślą o obsłudze wszystkich operacji związanych z prowadzeniem hoteli, tak aby był w pełni funkcjonalny i dostosowany do potrzeb użytkowników. Obejmuje on kompleksowe rozwiązania dla zarządzania rezerwacjami, obsługą klientów, finansami, pracownikami i innymi aspektami związanymi z funkcjonowaniem sieci hoteli. Dzięki temu użytkownicy systemu mogą łatwo zarządzać i monitorować procesy biznesowe, co przyczynia się do poprawy efektywności i rentowności biznesu.

# Wykaz/spis ilustracji

# Wykaz/spis tabel