**Projekt bazy danych dla administracji hotelu**

Przedmiot: **Bazy danych**

Prowadzący: **dr inż. Arkadiusz Mirakowski**

Autor:

**Piotr Przybyła**

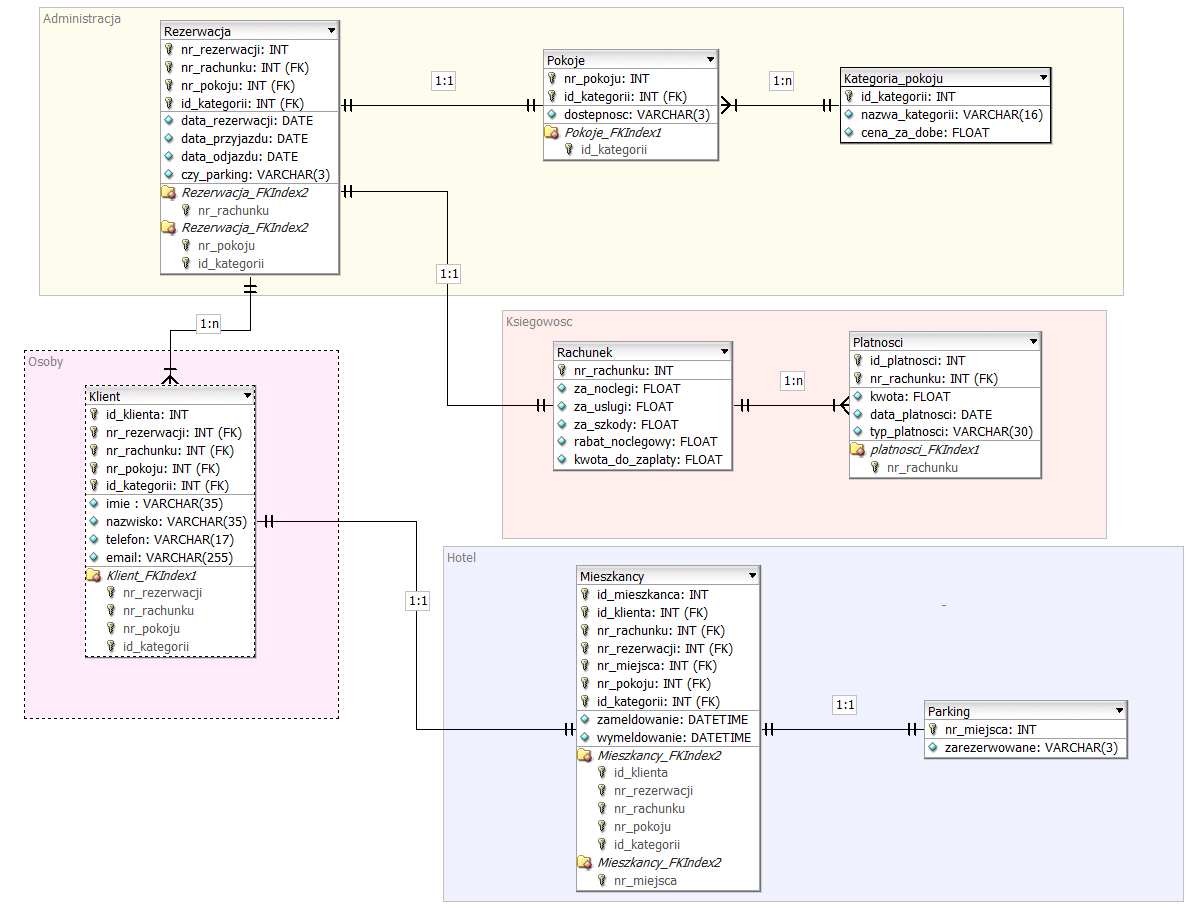
**Grupa VI , Informatyka praktyczna, I rok studiów**

Gdańsk, 2022

1. **Opis bazy danych**

Baza danych pełni rolę części systemu administracyjnego pewnego hotelu. Składa się z tabeli powiązanych ze sobą różnymi relacjami. Służy ona lepszemu zorganizowaniu pracy hotelu. Zawiera informacje niezbędne do łatwej i szybkiej współpracy pomiędzy pracownikami hotelu. Baza posiada wiele zastosowań. Możemy uzyskać z niej informacje na temat zorganizowania i umiejscowienia pokoji hotelowych, informacje na temat klientów oraz dokonywanych przez nich rezerwacji, danych na temat rachunków oraz płatności za dokonaną rezerwację, osób zamieszkujących aktualnie hotel oraz zajmowanych przez nich miejsc parkingowych.

1. **Diagram związków encji – DBDesigner**



Rys.1. Diagram związków encji – DBDesigner4

1. **Omówienie tabel**
   1. **Klient**  – tabela zawiera dane personalne oraz kontaktowe klientów hotelu
   2. **Rezerwacja** – tabela zawiera dane na temat: daty dokonania rezerwacji, daty chęci przyjazdu oraz odjazdu przez klienta hotelu, ilości dni planowanych przez klienta na spędzenie w hotelu oraz czy potrzebne jest miejsce parkingowe.
   3. **Pokoje** – tabela zawiera dane dotyczące rezerwowanych pokoi: numer pokoju, cenę za spędzoną dobę oraz czy dany pokój jest dostępny
   4. **Kategoria\_pokoju** – tabela zawiera dane na temat kategorii pokojów
   5. **Rachunek** – tabela zawiera dane odnoszące się do aktualnych należności klienta, informację o rabacie i kwocie do zapłaty.
   6. **Płatnosci** – tabela zawiera dane na temat daty oraz sposobu płatności dokonanych przez klientów
   7. **Mieszkancy** – tabela zawiera dane na temat osób aktualnie mieszkających w hotelu
   8. **Parking** - tabela zawiera dane dotyczące miejsc parkingowych hotelu
2. **Definicje tabel**

**Klient**

|  |  |  |
| --- | --- | --- |
| nazwa atrybutu | typ atrybutu | inne cechy atrybutu |
| id\_klienta | int | PK |
| nr\_rezerwacji | int | FK |
| nr\_rachunku | int | FK |
| nr\_pokoju | int | FK |
| id\_kategorii | int | FK |
| imie | varchar(35) | - |
| nazwisko | varchar(35) | - |
| telefon | varchar(17) | - |
| e-mail | varchar(255) | - |

**Rezerwacja**

|  |  |  |
| --- | --- | --- |
| nazwa atrybutu | typ atrybutu | inne cechy atrybutu |
| nr\_rezerwacji | int | PK |
| nr\_rachunku | int | FK |
| nr\_pokoju | int | FK |
| id\_kategorii | int | FK |
| data\_rezerwacji | date | - |
| data\_przyjazdu | date | - |
| data\_odjazdu | date | - |
| czy\_parking | varchar(3) | - |

**Kategoria\_pokoju**

|  |  |  |
| --- | --- | --- |
| nazwa atrybutu | typ atrybutu | inne cechy atrybutu |
| id\_kategorii | int | PK |
| nazwa\_kategorii | varchar(16) | - |
| cena\_za\_dobe | float | - |

**Pokoje**

|  |  |  |
| --- | --- | --- |
| nazwa atrybutu | typ atrybutu | inne cechy atrybutu |
| nr\_pokoju | int | PK |
| id\_kategorii | int | FK |
| dostepnosc | varchar(3) | - |

**Rachunek**

|  |  |  |
| --- | --- | --- |
| nazwa atrybutu | typ atrybutu | inne cechy atrybutu |
| nr\_rachunku | int | PK |
| za\_noclegi | float | - |
| za\_uslugi | float | - |
| za\_szkody | float | - |
| rabat\_noclegowy | float | - |
| kwota\_do\_zaplaty | float | - |

**Platnosci**

|  |  |  |
| --- | --- | --- |
| nazwa atrybutu | typ atrybutu | inne cechy atrybutu |
| id\_platnosci | int | PK |
| nr\_rachunku | int | FK |
| kwota | float | - |
| data\_ platnosci | date | - |
| typ\_platnosci | varchar(30) | - |

**Mieszkancy**

|  |  |  |
| --- | --- | --- |
| nazwa atrybutu | typ atrybutu | inne cechy atrybutu |
| id\_mieszkanca | int | PK |
| id\_klienta | int | FK |
| nr\_rachunku | int | FK |
| nr\_rezerwacji | int | FK |
| nr\_miejsca | int | FK |
| nr\_pokoju | int | FK |
| id\_kategorii | int | FK |
| zameldowanie | datetime | - |
| wymeldowanie | datetime | - |

**Parking**

|  |  |  |
| --- | --- | --- |
| nazwa atrybutu | typ atrybutu | inne cechy atrybutu |
| nr\_miejsca | int | PK |
| zarezerwowane | varchar(3) | - |

1. **Zawartość tabel**

**Klient**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |



**Rezerwacja**



**Kategoria\_pokoju**



**Pokoje**



**Rachunek**



**Platnosci**



**Mieszkancy**



**Parking**



1. **Relacje bazodanowe**

**Relacja 1 – 1:n** – Kategoria\_pokoju - Pokoje

* 1. Alter table Pokoje
  2. add CONSTRAINT FK\_Kat\_pok\_id\_kategorii foreign key (id\_kategorii)
  3. references Kategoria\_pokoju(id\_kategorii)

**Relacja 2 – 1:1**  – Rezerwacja – Pokoje

1. Alter table Rezerwacja
2. add CONSTRAINT FK\_Pok\_nr\_pokoju foreign key (nr\_pokoju)
3. references Pokoje(nr\_pokoju)
5. Alter table Rezerwacja
6. add CONSTRAINT FK\_Kat\_pok\_id\_kategorii foreign key (id\_kategorii)
7. references Kategoria\_pokoju(id\_kategorii)

**Relacja 3 – 1:1** – Rachunek - Rezerwacja

1. Alter table Rezerwacja
2. add CONSTRAINT FK\_Rach\_ nr\_rachunku foreign key (nr\_rachunku)
3. references Rachunek(nr\_rachunku)

**Relacja 4 – 1:n** – Rachunek – Platnosci

1. Alter table Platnosci
2. add CONSTRAINT FK\_Rach\_ nr\_rachunku foreign key (nr\_rachunku)
3. references Rachunek(nr\_rachunku)

**Relacja 5 – 1:n** – Rezerwacja – Klient

1. Alter table Klient
2. add CONSTRAINT FK\_Rez\_nr\_rezerwacji foreign key (nr\_rezerwacji)
3. references Rezerwacja(nr\_rezerwacji)
5. Alter table Klient
6. add CONSTRAINT FK\_Kat\_pok\_id\_kategorii foreign key (id\_kategorii)
7. references Kategoria\_pokoju(id\_kategorii)
9. Alter table Klient
10. add CONSTRAINT FK\_Rach\_ nr\_rachunku foreign key (nr\_rachunku)
11. references Rachunek(nr\_rachunku)
13. Alter table Klient
14. add CONSTRAINT FK\_Pok\_nr\_pokoju foreign key (nr\_pokoju)
15. references Pokoje(nr\_pokoju)

**Relacja 6 – 1:1** – Klient - Mieszkancy

1. Alter table Mieszkancy
2. add CONSTRAINT FK\_Klient\_ id\_klienta foreign key (id\_klienta)
3. references Klient(id\_klienta)
5. Alter table Klient
6. add CONSTRAINT FK\_Kat\_pok\_id\_kategorii foreign key (id\_kategorii)
7. references Kategoria\_pokoju(id\_kategorii)
9. Alter table Klient
10. add CONSTRAINT FK\_Rach\_ nr\_rachunku foreign key (nr\_rachunku)
11. references Rachunek(nr\_rachunku)
13. Alter table Klient
14. add CONSTRAINT FK\_Rez\_nr\_rezerwacji foreign key (nr\_rezerwacji)
15. references Rezerwacja(nr\_rezerwacji)
17. Alter table Klient
18. add CONSTRAINT FK\_Pok\_nr\_pokoju foreign key (nr\_pokoju)
19. references Pokoje(nr\_pokoju)

**Relacja 7 – 1:1** – Mieszkancy - Parking

1. Alter table Mieszkancy
2. add CONSTRAINT FK\_Park\_ nr\_miejsca foreign key (nr\_miejsca)
3. references Parking(nr\_miejsca)
4. **Omówienie i prezentacja widoków**

W tym punkcie umieszczamy informacje dotyczące 3 opracowanych widoków. Funkcjonalność widoków musi być utrzymana na poziomie uniwersyteckim. Dokumentację dotyczącą widoków tworzymy zgodnie z poniższym wzorem:

jakie\_dostępne – jest to widok, który umożliwia sprawdzenie numerów oraz kategorii dostępnych pokoji

DROP VIEW IF EXISTS jakie\_dostępne

GO

CREATE VIEW jakie\_dostępne AS

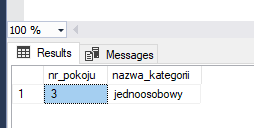
SELECT nr\_pokoju, nazwa\_kategorii FROM Pokoje, Kategoria\_pokoju

WHERE Pokoje.id\_kategorii=Kategoria\_pokoju.id\_kategorii

AND dostepnosc='TAK'

GO

SELECT\*FROM jakie\_dostępne

****

Rys.2. Wynik działania widoku jakie\_dostępne

wandalizm – jest to widok, który umożliwia zobaczenia jacy klienci dopuścili się wandalizmu, w jakim pokoju, na jaki koszt były straty oraz na jaki numer rachunku zostały one naliczone.

DROP VIEW IF EXISTS wandalizm

GO

CREATE VIEW wandalizm AS

SELECT imie, nazwisko, Rachunek.nr\_rachunku, za\_szkody, Pokoje.nr\_pokoju,

CASE

WHEN za\_szkody>0 THEN 'Wandal'

ELSE 'Nie wandal'

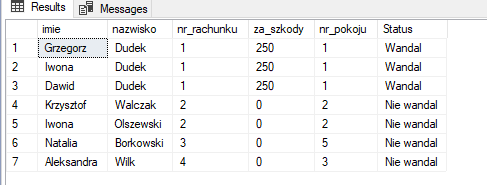
END AS 'Status'

FROM Rachunek, Pokoje, Klient

WHERE Rachunek.nr\_rachunku=Klient.nr\_rachunku and Pokoje.nr\_pokoju=Klient.nr\_pokoju

GO

SELECT\*FROM wandalizm

****

Rys.3. Wynik działania widoku wandalizm

przyjezdni – jest to widok, który umożliwia zobaczenie obecnej i przyszłej daty przyjazdów klientów do zarezerwowanych pokoji, numerów tych pokoji, nr\_rezerwacji, na ile dni mają Ci klienci zostać oraz czy potrzebują miejsca parkingowego.

DROP VIEW IF EXISTS przyjezdni

GO

CREATE VIEW przyjezdni AS

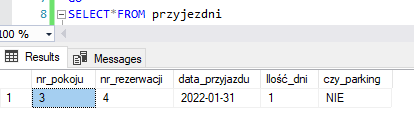
SELECT nr\_pokoju, Rezerwacja.nr\_rezerwacji, data\_przyjazdu, DATEDIFF(DAY ,data\_przyjazdu, data\_odjazdu) AS Ilość\_dni, czy\_parking

FROM Rezerwacja

WHERE data\_przyjazdu >= GETDATE()

GO

SELECT\*FROM przyjezdni



Rys.4. Wynik działania widoku przyjezdni

1. **Omówienie i prezentacja funkcji**

W tym punkcie umieszczamy informacje dotyczące 3 opracowanych funkcji. Funkcjonalność funkcji musi być utrzymana na poziomie uniwersyteckim. Dokumentację dotyczącą funkcji tworzymy zgodnie z poniższym wzorem:

cały\_rachunek – jest to funkcja, która zwraca zestawienie wszystkich kosztów dla danego nr rachunku.

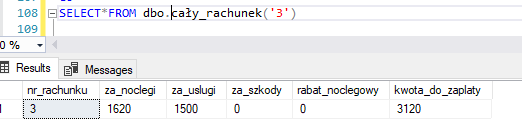
DROP FUNCTION IF EXISTS dbo.cały\_rachunek

GO

CREATE FUNCTION cały\_rachunek(@nr\_rachunku int)

RETURNS table AS

RETURN (SELECT\*FROM Rachunek WHERE nr\_rachunku=@nr\_rachunku)

****

Rys. 5. Wynik działania funkcji **cały\_rachunek**

ilosc\_rabatow – jest to funkcja, która zwraca ile razy korzystano z rabatu noclegowego.

DROP FUNCTION IF EXISTS dbo.ilosc\_rabatow

GO

CREATE FUNCTION dbo.ilosc\_rabatow()

RETURNS int

BEGIN

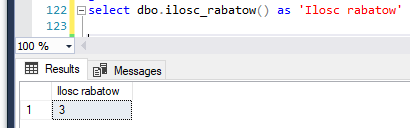
DECLARE @ile\_rabatow int

SET @ile\_rabatow=(SELECT COUNT(\*) FROM Rachunek

WHERE rabat\_noclegowy>0)

RETURN @ile\_rabatow

END



Rys. 6. Wynik działania funkcji ilosc\_rabatow

kalkulator\_noclegow – jest to funkcja, która liczy jaka jest cena wynajęcia pokoju o danej kategorii od daty przyjazdu do daty odjazdu.

DROP FUNCTION IF EXISTS dbo.kalkulator\_noclegow

GO

CREATE FUNCTION dbo.kalkulator\_noclegow(@data\_przyjazdu date,@data\_odjazdu date, @id\_kategorii\_pokoju int)

RETURNS float

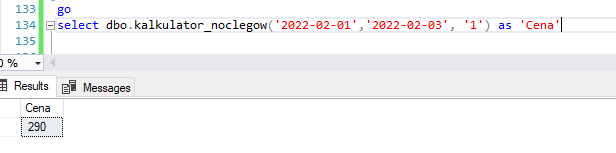
BEGIN

DECLARE @cena float

SET @cena=((SELECT cena\_za\_dobe from Kategoria\_pokoju WHERE id\_kategorii=@id\_kategorii\_pokoju) \* CAST(DATEDIFF(DAY,@data\_przyjazdu, @data\_odjazdu) AS float))

RETURN @cena

END



Rys. 6. Wynik działania funkcji kalkulator\_noclegow

**Punktacja (max 10 pkt):**

1. **0,5 pkt**
2. **2,5 pkt**
3. **0,5 pkt**
4. **0,5 pkt**
5. **0,5 pkt**
6. **1 pkt**
7. **1,5 pkt**
8. **3 pkt**

**Termin oddania – 14 blok (przedostatni)**

**Sposób oddania: droga mailowa, format PDF.**