

## Zaawansowane systemy baz danych

### Projekt – etap 1

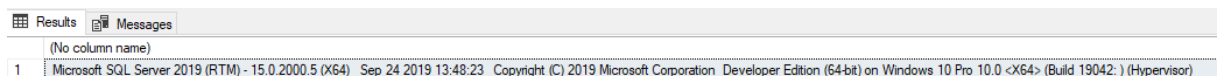
Krzysztof Wyszyński

#### 1. Case study

W celu usprawnienia procesu kwaterunku, zarządzania i komunikacji w Domach Studenckich postanowiono stworzyć system realizujący powyższe funkcjonalności. Podstawową funkcją systemu będzie możliwość zapisywania studentów do poszczególnych akademików. Obejmuje on również pracowników akademików – aktualnie większość portierni w Domach Studenckich bazuje na staromodnym zapisywaniu istotnych informacji na kartkach papieru. Za pomocą systemu uzyskają możliwość dokumentowania swojej pracy w formie elektronicznej – od przyjmowania gości po zgłaszanie incydentów. Każdy Dom Studencki posiadał będzie również administrację, bezpośrednio odpowiedzialną za placówkę oraz jej mieszkańców.

#### 2. System zarządzania bazą danych

W ramach projektu zdecydowałem się wykorzystać SQL Server, ponieważ jest on wykorzystywany w ramach materiałów wykładowych oraz miałem z nim styczność w przeszłości. Microsoft SQL Server pozwala na darmowe korzystanie z usług w fazie developmentu. Za pomocą odpowiednich narzędzi można prosty sposób wygenerować bazę danych przy użyciu interfejsu graficznego.



Results		Messages
(No column name)		
1	Microsoft SQL Server 2019 (RTM) - 15.0.2000.5 (X64) Sep 24 2019 13:48:23 Copyright (C) 2019 Microsoft Corporation Developer Edition (64-bit) on Windows 10 Pro 10.0 <X64> (Build 19042;) (Hypervisor)	

Rys 1. Wersja systemu bazodanowego uzyskana za pomocą komendy „SELECT @@VERSION”

Server memory options

Minimum server memory (in MB):  
0

Maximum server memory (in MB):  
2147483647

Other memory options

Index creation memory (in KB, 0 = dynamic memory):  
0

Minimum memory per query (in KB):  
1024

*Rys 2. Ustawienia pamięci systemu bazodanowego*

Zmieniłem metodę autoryzacji z domyślnej na „SQL Server and Windows Authentication Mode”. Dzięki temu do bazy danych będzie można połączyć się również za pomocą loginu i hasła, co pozwoli na podpinanie bazy pod aplikacje sieciowe i dzielenie dostępu z użytkownikami na innych maszynach.

### **3. Opis projektu ze strony bazodanowej**

#### **3.1. Użytkownicy korzystający z bazy danych**

Według wstępnych założeń baza danych posiadać będzie czterech użytkowników. Jednym z nich będzie pracownik recepcji - będzie dostęp do danych studentów oraz możliwość zgłaszania skarg i rejestrowania noclegów. Drugim użytkownikiem będzie pracownik administracji, zarządzający całym Domem Studenckim. Następny konsument to gość portalu informacyjnego – jego jedynym przywilejem będzie możliwość odczytywania informacji – adresów Domów Studenckich, danych kontaktowych do administracji, ogłoszeń itp. Nie zabraknie również administratora posiadającego pełne uprawnienia do zarządzania systemem, wszystkimi akademikami oraz tabelami w bazie danych.

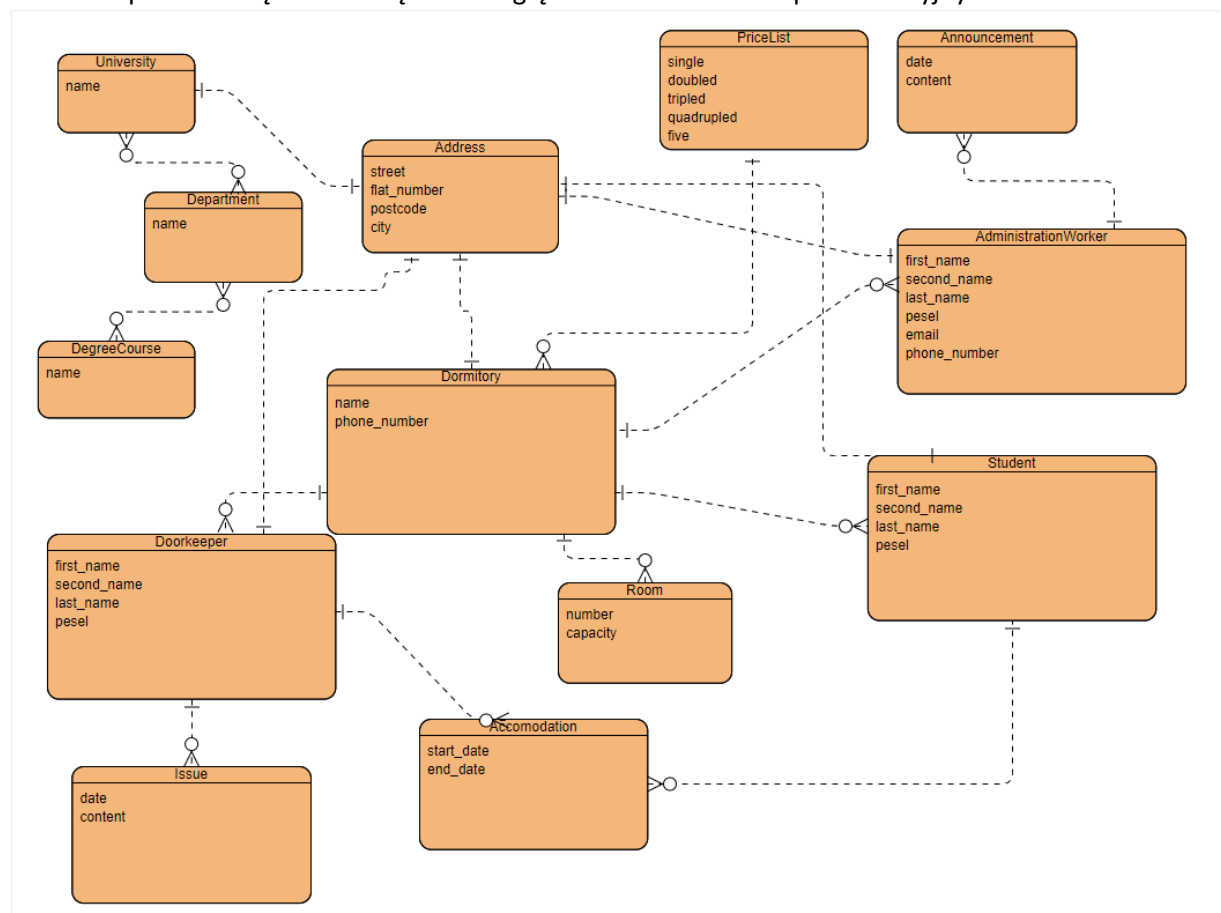
#### **3.2. Dane przechowywane w bazie oraz oferowane usługi**

Baza danych przechowuje dane o Domach Studenckich. W skład każdej instytucji wchodzi pracownicy administracji, portierni oraz studenci. Każdy akademik posiada cennik, adres oraz przypisany jest do konkretnej uczelni. Dana uczelnia może mieć wiele Domów Studenckich. Student zarejestrowany w akademiku posiada swój pokój. Pokój zawiera numer oraz liczbę studentów, którzy mogą w nim mieszkać. Istnieją pokoje jedno-, dwu-, trzy-, cztero- i

pięcioosobowe. System przechowuje dane o każdej z trzech kategorii osób związanych z Domem Studenckim. Są to podstawowe informacje osobowe: imię, drugie imię, nazwisko, PESEL, data urodzenia oraz adres. Student przypisany jest do konkretnego wydziału oraz semestru studiów. Pracownik administracji może rejestrować w systemie ogłoszenia skierowane do społeczności akademickiej. Student ma prawo do przenocowania jednego gościa jednego wieczoru, po uprzedniej konsultacji z pracownikiem portierni. Rejestracja noclegu zawiera informację o dacie przybycia, dacie opuszczenia Domu Studenckiego przez gościa oraz jego danych osobowych. Pracownik portierni może zgłaszać skargi oraz zawsze podczas pracy loguje się do systemu zmieniając status *is\_working*.

### 3.3 Wstępny diagram ERD – konceptualny

Poniższy diagram nie uwzględnia typów danych, generalizacji, rozkładu relacji wiele-do-wiele na tabele pośredniczące. Zostaną one uwzględnione w modelu implementacyjnym.

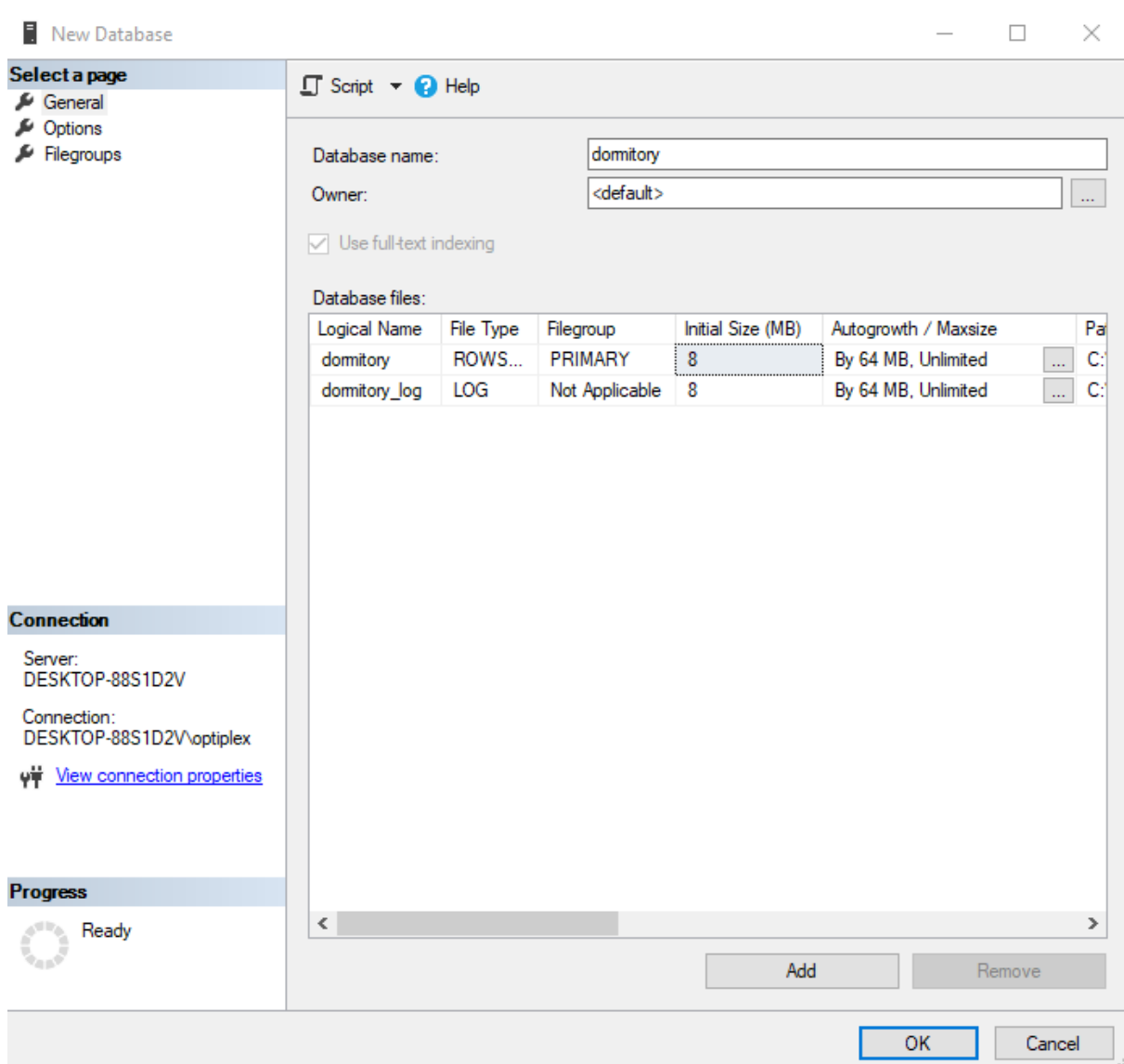


Rys 3. Wstępny diagram ERD

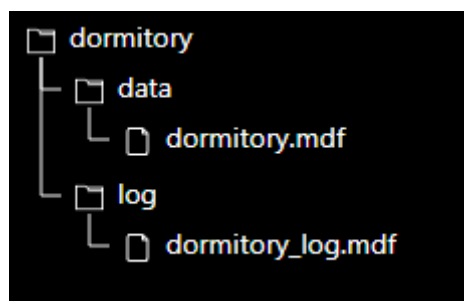
## 4. Implementacja bazy danych

### 4.1. Struktura plików

Przyjąłem domyślne wartości pamięci przeznaczone stronom bazodanowym – 8MB na jeden plik ROWS DATA, z możliwością Autogrowth. Zgodnie z wiedzą nabytą podczas zajęć wykładowych, oddzieliłem fizycznie log transakcyjny od folderu zawierającego dane.

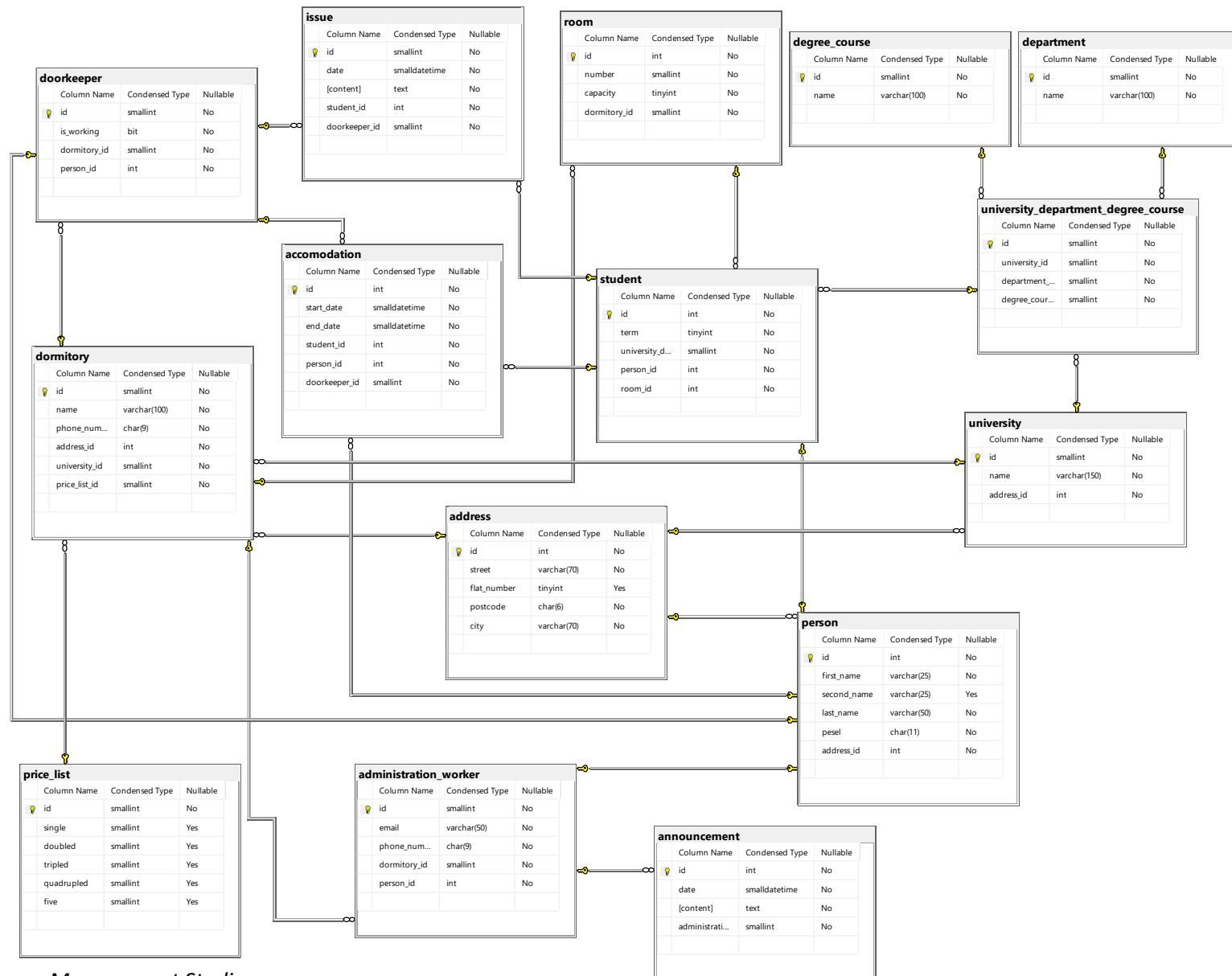


Rys 4. Struktura plików bazy danych



Rys 5. Fizyczny rozkład plików bazy danych

## 4.2. Układ logiczny bazy danych



Rys 6. Schemat logiczny bazy danych

wygenerowany przez Microsoft SQL Server Management Studio

#### 4.3. Istotne decyzje implementacyjne

- a) Encje *Student*, *Doorkeeper* oraz *AdministrationWorker* współdzielą dane osobowe. W związku z tym dokonałem generalizacji i wydzieliłem wspólne atrybuty do oddzielnej encji *Person*, poprzez utworzenie relacji „one-to-one” i przyjęciu klucza obcego jako UNIQUE
- b) Z racji, że zarówno między wydziałem a kierunkiem studiów, jak i między wydziałem a uczelnią występuje relacja „many-to-many”, stworzyłem tabelę asocjacyjną łączącą wszystkie trzy składowe
- c) Podczas doboru typu danych starałem się możliwie ograniczyć zakres zajmowanej przez nie pamięci, według zasad przedstawionych na wykładzie:

- Starać się używać:
  - varchar zamiast char
  - tinyint / smallint zamiast int / bigint
  - bit zamiast int
  - smallmoney zamiast money
  - smalldatetime zamiast datetime
  - char / varchar zamiast nchar / nvarchar

#### 5. Dane wprowadzone do bazy

Do bazy danych wprowadziłem 5 uczelni:

- a) Politechnika Warszawska,
- b) Uniwersytet Warszawski,
- c) Warszawski Uniwersytet Medyczny,
- d) Szkoła Główna Gospodarstwa Wiejskiego w Warszawie,
- e) Szkoła Główna Handlowa,

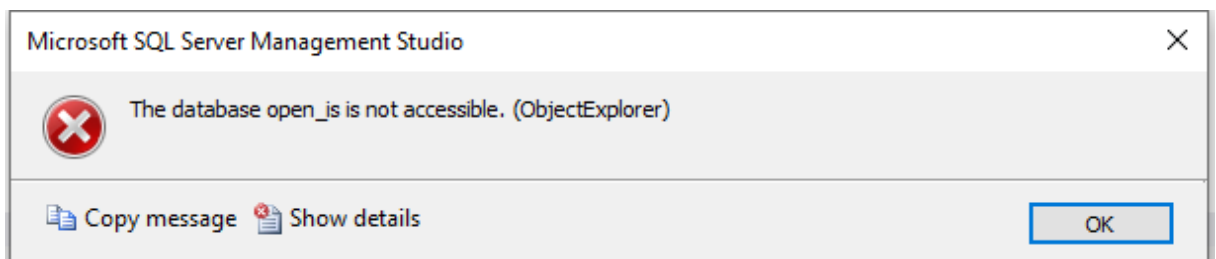
16 wydziałów, 12 kierunków, 13 Domów Studenckich. W skład każdego Domu Studenckiego wchodzi 6 pokoi, 2 osoby pracujące na portierni, jeden pracownik administracji oraz 4 studentów.

Liczba danych nie jest duża, lecz dosyć starannie odzwierciedlają rzeczywistość, dlatego ich generacja wymagała dużego nakładu pracy.

#### 6. Użytkownicy

W celu ograniczenia dostępu do zasobów bazy danych utworzyłem użytkowników.

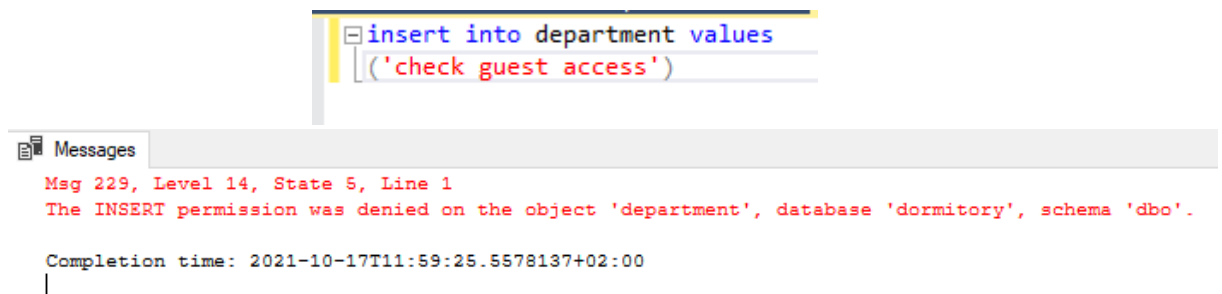
Każdy z wymienionych poniżej użytkowników ma dostęp tylko do bazy danych dotyczącej projektu. Próba uzyskania dostępu do innej bazy skutkuje następującym komunikatem:



Rys 7. Ograniczenie dostępu do innych baz dla użytkowników

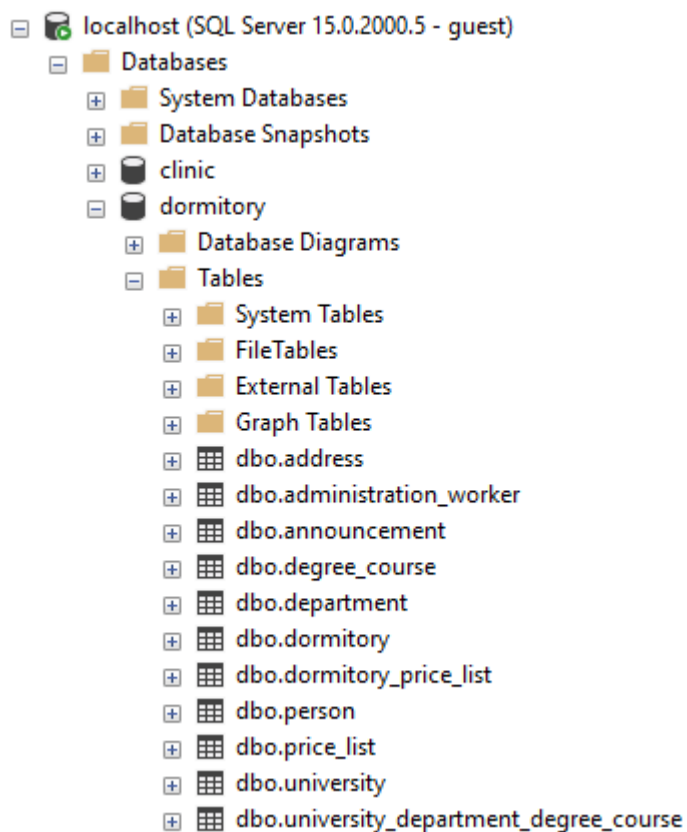
## 6.1. Gość

Podstawowym użytkownikiem bazy danych jest gość (guest). Posiada on dostęp do danych o uczelniach, Domach Studenckich wchodzących w ich skład, pracownikach administracji oraz cenniku pomieszczeń. Nie ma praw do edycji bądź dodawania jakichkolwiek rekordów.



*Rys 7. Ograniczenie dostępu do edycji danych w bazie dla gościa*

Poniżej przedstawiono listę tabel dostępnych do odczytu dla gościa:



*Rys 8. Dostępne tabele dla gościa*

## 6.2. Pracownik portierni

Kolejny użytkownik bazy danych to pracownik portierni (dormitory\_doorkeeper). Posiada on dostęp do wszystkich danych w obrębie danego Domu Studenckiego, lecz nie ma praw do

edycji bądź dodawania pomieszczeń, studentów, pracowników administracji lub innych pracowników portierni.

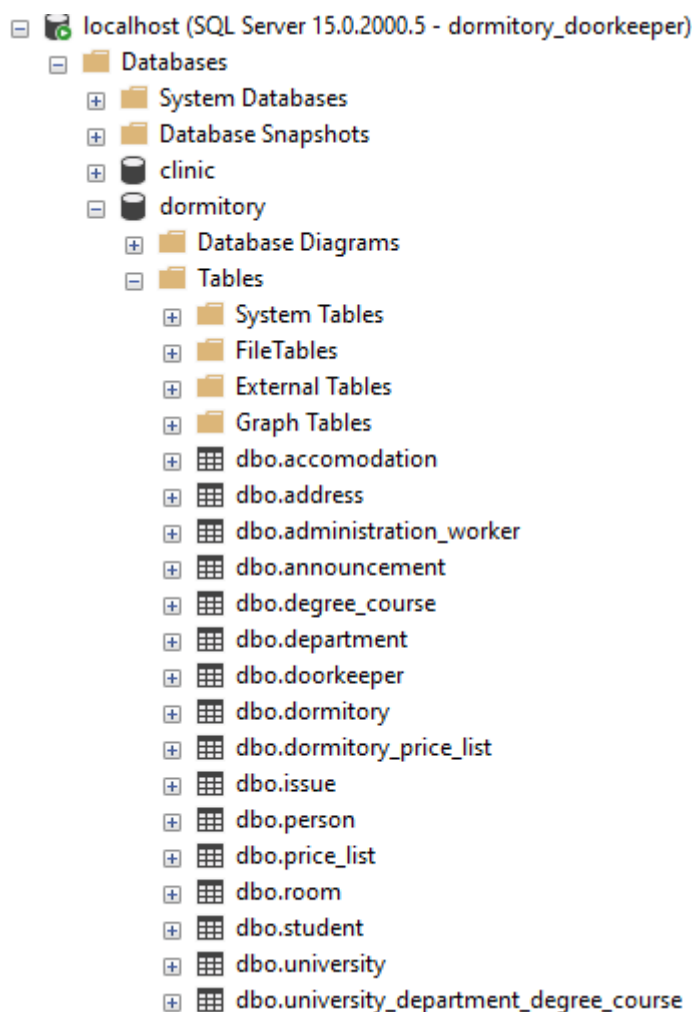
```
insert into administration_worker values
('administracja@bratniak.com', '111222333', 1, 79)
```

Messages

Msg 229, Level 14, State 5, Line 1  
The INSERT permission was denied on the object 'administration\_worker', database 'dormitory', schema 'dbo'.

Completion time: 2021-10-17T13:24:04.3637522+02:00

Rys 9. Ograniczenie dostępu do edycji danych w bazie dla pracownika portierni



Rys 10. Dostępne tabele dla pracownika portierni

### 6.3. Pracownik administracji

Trzecim użytkownikiem jest pracownik administracji (dormitory\_administrator). Tak jak pracownik portierni ma on wgląd do wszystkich danych w Domu Studenckim. Nie może jednak zgłaszać incydentów ani rejestrować noclegów (są to zadania pracowników portierni), bądź też dodawać nowych pracowników administracji.



```
SQLQuery3.sql - loc...administrator (64))* X
insert into accomodation values
('2021-10-19 18:00:00', '2021-10-20 18:00:00', 1, 19, 1)
```

Messages

Msg 229, Level 14, State 5, Line 1  
The INSERT permission was denied on the object 'accomodation', database 'dormitory', schema 'dbo'.

Completion time: 2021-10-17T13:37:15.5016709+02:00

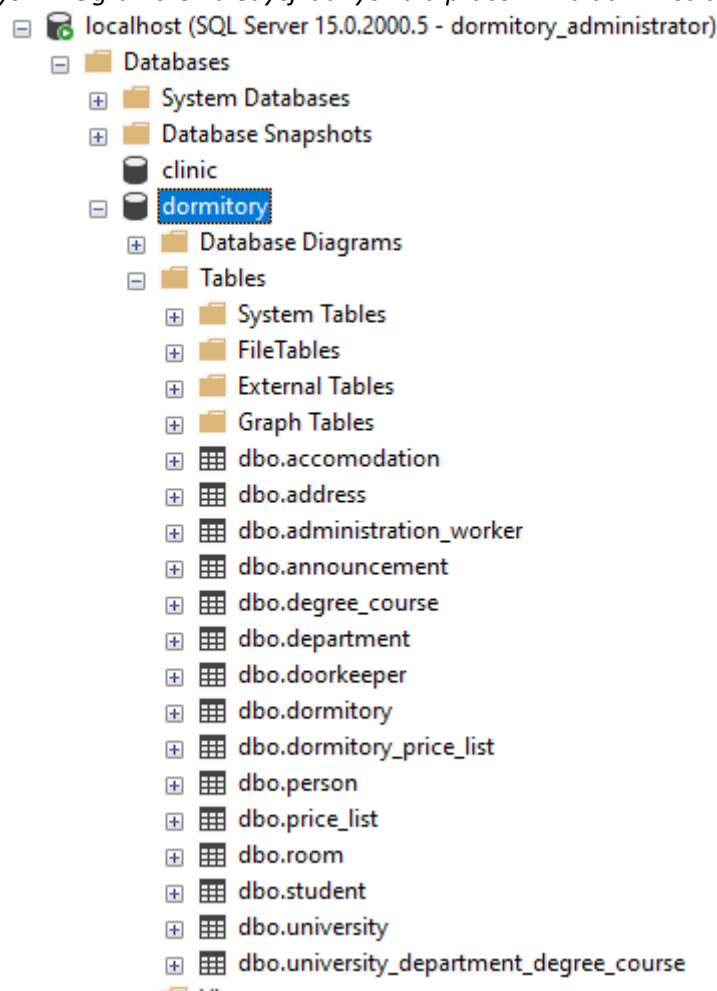
```
SQLQuery3.sql - loc...administrator (64))* X
insert into issue values
('2021-10-19 18:00:00', N'Student narozrabiał', 1, 1)
```

Messages

Msg 229, Level 14, State 5, Line 1  
The INSERT permission was denied on the object 'issue', database 'dormitory', schema 'dbo'.

Completion time: 2021-10-17T13:38:57.2834828+02:00

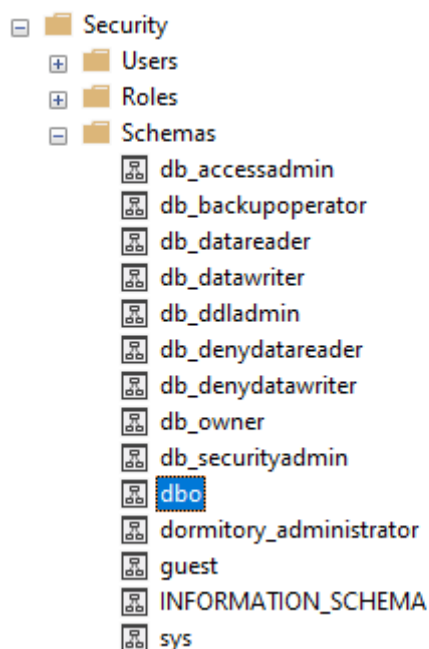
Rys 11. Ograniczenia edycji danych dla pracownika administracji



Rys 12. Dostępne tabele dla pracownika administracji

#### 6.4. Schematy bazodanowe

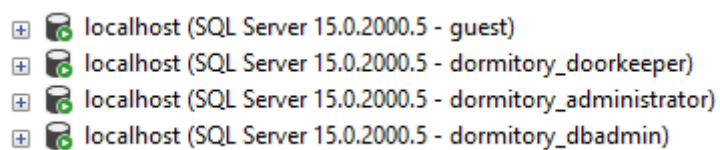
Poniższy zrzut ekranu przedstawia schematy zawarte w bazie:



Wszystkie tabele przypisane są domyślnie do schematu dbo.

#### 6.5. Administrator bazy danych

Ostatni użytkownik to administrator bazy danych (dormitory\_dbowner), który ma wszystkie uprawnienia w bazie danych.



Rys 13. Wykaz użytkowników bazy danych

## 7. Przykładowe zapytania

### 7.1. Zapytanie pierwsze

Wyświetl dane osobowe studentów ze wszystkich Domów Studenckich mieszkających w pokojach dwuosobowych, wraz z informacją o numerze pokoju, posortowane alfabetycznie według nazwy Domu Studenckiego.

```
select person.first_name, person.last_name, person.pesel, room.number as 'Room number',  
       room.capacity as 'Room capacity', dormitory.name as 'Dormitory name' from student  
inner join room on room.id = room_id  
inner join dormitory on dormitory.id = room.dormitory_id  
inner join person on person.id = person_id where room_id in (  
    select id from room where capacity = 2  
) order by dormitory.name ASC
```

Results		Messages				
	first_name	last_name	pesel	Room number	Room capacity	Dormitory name
1	Michał	Faleks	97127338427	138	2	Dom Studencki Akademik
2	Stefan	Burczymucha	99022305427	137	2	Dom Studencki Bratniak-Musze...
3	Michał	Broda	99012305427	137	2	Dom Studencki Bratniak-Musze...
4	Piotr	Peks	95021305427	138	2	Dom Studencki Bratniak-Musze...
5	Piotr	Feks	95029305427	138	2	Dom Studencki Bratniak-Musze...
6	Michał	Fleks	97127305427	137	2	Dom Studencki Mikrus
7	Jakub	Ojeks	95027305427	137	2	Dom Studencki Mikrus
8	Antoni	Steks	95027085427	138	2	Dom Studencki Mikrus
9	Kamil	Zalateks	92027310427	138	2	Dom Studencki Mikrus
10	Łukasz	Deks	96070994189	138	2	Dom Studencki nr 1
11	Piotr	Peks	99081577875	138	2	Dom Studencki nr 2
12	Piotr	Feks	97110496845	138	2	Dom Studencki nr 2
13	Piotr	Geks	94028305427	137	2	Dom Studencki Ustronie
14	Piotr	Neks	99027305427	138	2	Dom Studencki Ustronie
15	Kamil	Zeks	92027305427	221	2	Dom Studencki Ustronie
16	Antoni	Steks	02120174297	138	2	Dom studenta Dendryt
17	Piotr	Fonal	01081577875	138	2	Dom studenta Dendryt
18	Michał	Ofleks	02310174097	138	2	Dom Studenta nr 5
19	Jakub	Bolojeks	02017174207	138	2	Dom Studenta nr 5
20	Jakub	Ojeks	02010174297	138	2	Dom Studenta nr 6
21	Kamil	Zeks	02031462646	138	2	Grosik
22	Łukasz	Deks	03070994189	138	2	Grosik
23	Piotr	Donal	01309522187	138	2	Krokus Dom Studenta
24	Jakub	Momal	01010147197	138	2	Oaza Dom Studenta
25	Piotr	Peks	01089577875	138	2	Sabinki

## 7.2. Zapytanie drugie

Wyświetl sumę wszystkich skarg w każdym z Domów Studenckich w roku 2021, z podziałem na Domy Studenckie.

```
select count(room.id) as 'Issues', dormitory.name as 'Dormitory name' from issue
inner join student on student.id=student_id
inner join room on room.id=student.room_id
inner join dormitory on dormitory.id=room.dormitory_id
where year(date) = '2021'
group by dormitory.name
```

Results		Messages
	Issues	Dormitory name
1	6	Dom Studencki Akademik
2	3	Dom Studencki Bratniak-Musze...
3	4	Dom Studencki Mikrus
4	2	Dom Studencki nr 1
5	2	Dom Studencki nr 2
6	8	Dom Studencki Ustronie
7	2	Dom studenta Dendryt
8	6	Dom Studenta nr 5
9	2	Dom Studenta nr 6
10	2	Grosik
11	2	Krokus Dom Studenta
12	2	Oaza Dom Studenta
13	2	Sabinki

