

Projekt Zaliczeniowy - Bazy Danych - Semestr 5

Krystian Duma Grupa - Z501 - Nr. Albumu 7763

23 Grudzień 2018

Spis treści

1	Krótki opis słowny projektu	2
2	Założenia do projektu	2
3	Środowisko Projektowe	2
4	Model fizyczny bazy danych	3
5	Skrypt tworzący obiekty w bazie danych	4
5.1	Model wersjonowania bazy danych	4
5.2	Wynik uruchomienia całego skryptu w trybie wsadowym	4
5.3	Tabele	5
5.3.1	Utworzenie tabeli z obiektami	5
5.3.2	Utworzenie tabeli z najmami i Utworzenie indeksu unikatowego w tabeli z najemcami .	6
5.4	Widoki	7
5.5	Funkcje Skalarne	7
5.6	Funkcje Tabelarne	7
5.7	Triggery	7
5.8	Procedury Składowane	7
5.9	Skrypty w oparciu o kursory	7
5.10	Inne poznane obiekty, własności bazy danych	7
6	Skrypt tworzący dane testowe	8
7	Skrypt tworzący użytkowników i nadający uprawnienia	9
8	Skrypt usuwający obiekty z bazy danych	10
8.1	Wynik uruchomienia całego skryptu w trybie wsadowym	10
9	Skrypt usuwający użytkowników i uprawnienia	11
10	Skrypt usuwający dane testowe	12

1 Krótki opis słowny projektu

Projekt zawiera założenia do bazy danych przechowującej podstawowe informacje o wybranych funkcjach systemu informatycznego wspierającego funkcjonowanie agencji wynajmu krótkoterminowego domów, mieszkań lub innych obiektów.

2 Założenia do projektu

Przyjęte zostały następujące założenia do projektu

1. Podstawowe Obiekty

- **Obiekt** - obiekt najmu - np. konkretny dom lub mieszkanie,
- **Użytkownik** - osoba wynajmująca mieszkanie lub dom,

2. Przechowywane zadania (transakcje)

- **Najem** - transakcja związana z wynajęciem **Obiektu** przez **Użytkownika**.

3. Szczegóły opisu

- **Użytkownik** - potrzeba przechowania informacji: nazwisko klienta, imię klienta, wiek klienta, adres zamieszkania klienta, telefon klienta, płeć klienta oraz login używany do logowania do bazy danych.
- **Obiekt** - potrzeba przechowania informacji: nazwa własna obiektu, adres obiektu, dzienna stawka najmu obiektu, kategoria obiektu, obecny status najmu obiektu (informacja czy dany obiekt jest obecnie wolny lub zajęty), opis obiektu oraz inne atrybuty odpowiednie dla zgromadzonych obiektów.
 - Każdy obiekt może znajdować się w wielu różnych kategoriach,
 - Dla uproszczenia inne atrybuty będą znajdować się w opisie danego obiektu.
- **Najem** - potrzeba przechowania informacji: użytkownika-najemcy, wynajmowany obiekt, data rozpoczęcia najmu, data zakończenia najmu, koszt najmu.
 - Najem to transakcja tylko jednego **Użytkownika** i tylko jednego **Obiektu**,
 - Dla uproszczenia najem jest liczony od godziny 00:00 do godziny 23:59,
 - Jeden **Obiekt** może być w danym czasie wynajęty tylko jednemu użytkownikowi.

4. Użytkownicy i Uprawnienia

- Administrator ma dostęp do danych wszystkich użytkowników,
- Każdy **Użytkownik** ma założone oddzielne konto serwera SQL,
- Użytkownicy nie widzą danych oraz wypożyczeń innych użytkowników.

3 Środowisko Projektowe

Środowiskiem uruchomieniowym jest baza danych [Microsoft SQL Server 2017](#) uruchomiona w kontenerze [Docker](#)'a. Jako obraz bazowy został wybrany obraz [mcr.microsoft.com/mssql/server:2017-latest-ubuntu](#) który zawiera najaktualniejszą obecnie wersję [Microsoft SQL Server 2017](#) uruchomioną na systemie Linux - [Ubuntu Server](#). Do obrazu zostały doinstalowane dodatkowe narzędzia umożliwiające przygotowanie plików wyjściowych: tego dokumentu pdf ([LaTeX](#)) oraz skryptów tworzących i usuwających obiekty z bazy ([PHP](#)).

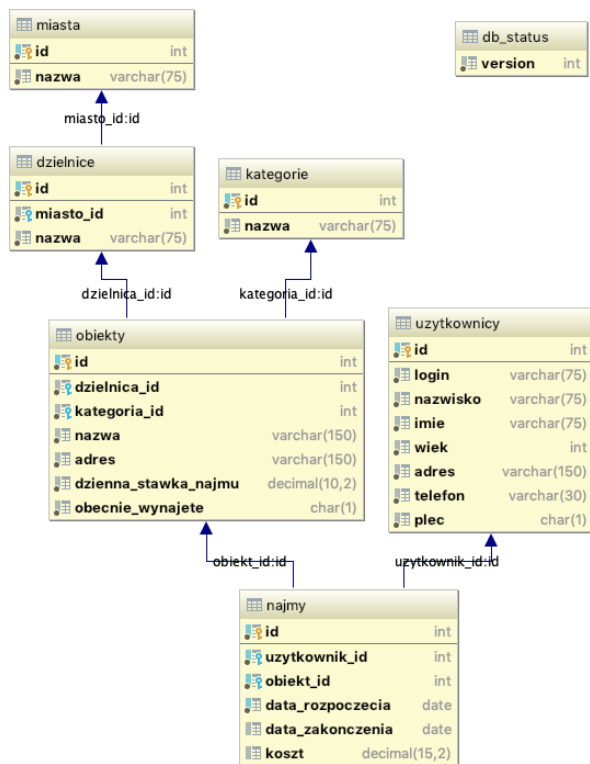
Jako aplikację służącą do łączenia się i wykonywania poleceń wykorzystane zostały aplikacje:

- Dołączona do [SQL Server](#)'a aplikacja wiersza poleceń - [sqlcmd](#)
- Środowisko IDE od czeskiej firmy [JetBrains](#) - [DataGrip](#)
- Środowisko IDE od [Microsoft](#)'u - [SQL Server Management Studio \(SSMS\)](#)

4 Model fizyczny bazy danych

Na Rysunku 1 znajduje się schemat (diagram tabel) wygenerowanej przez skrypt: [skrypt_tworzacy_obiekty_w_bazie_danych.sql](#).

Rysunek 1: Diagram tabel wygenerowanej bazy danych



Powered by yFiles

5 Skrypt tworzący obiekty w bazie danych

5.1 Model wersjonowania bazy danych

Jak można zauważyć na Rysunku 1, w bazie danych znajduje się jedna dodatkowa tabela `db_status` z jednym polem `version` - służy ona do przechowywania wersji bazy danych. Każda operacja w **skrypcie tworzącym** sprawdza i porównuje obecną oraz oczekiwaną wersję dla danej operacji. Dzięki temu zabiegowi nie będzie można uruchomić danej operacji dla jednej bazy danych wielokrotnie. Dodatkowo aktualizacja istniejącej bazy danych do najnowszej wersji będzie uproszczona - wystarczy uruchomić najnowszą wersję skryptu, a wykonane zostaną tylko nowe operacje dodane od ostatniego uruchomienia skryptu instalacyjnego. Dodatkowo w przypadku wystąpienia jakichkolwiek błędów jest przewidziana procedura ich łapania - na listingu 1 widzimy zawartość bloku `CATCH` skryptu instalacyjnego. Skrypt został przygotowany w taki sposób aby w przypadku wystąpienia błędu przerywał działanie¹ i przechodził od razu do bloku `CATCH`.

```
1 BEGIN CATCH
2
3 SELECT
4     ERROR_NUMBER() AS ErrorNumber,
5     ERROR_SEVERITY() AS ErrorSeverity,
6     ERROR_STATE() AS ErrorState,
7     ERROR_PROCEDURE() AS ErrorProcedure,
8     ERROR_LINE() AS ErrorLine,
9     ERROR_MESSAGE() AS ErrorMessage;
10
11 END CATCH;
```

Listing 1: Blok `CATCH` w skrypcie tworzącym

5.2 Wynik uruchomienia całego skryptu w trybie wsadowym

Jak widać na listingu 2, skrypt podaje bardzo dokładne informacje na temat aktualnie wykonywanej operacji. W większości przypadków wystąpienia ciągu tekstowego `(1 rows affected)`, następuje zmiana aktualnej wersji bazy danych w tabeli wersjonowania - `db_status`.

```
1
2 (1 rows affected)
3 Tabela wersjonowania została utworzona
4 Wersja 1: 'Utworzenie tabeli z miastami'
5
6 (1 rows affected)
7 Wersja 1: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 1
8 Wersja 2: 'Utworzenie tabeli z dzielnicami'
9
10 (1 rows affected)
11 Wersja 2: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 2
12 Wersja 3: 'Utworzenie relacji pomiędzy miastami a dzielnicami'
13
14 (1 rows affected)
15 Wersja 3: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 3
16 Wersja 4: 'Utworzenie tabeli z kategoriami'
17
18 (1 rows affected)
19 Wersja 4: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 4
20 Wersja 5: 'Utworzenie tabeli z obiektami'
21
22 (1 rows affected)
23 Wersja 5: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 5
24 Wersja 6: 'Utworzenie relacji pomiędzy dzielnicami a obiektami'
25
26 (1 rows affected)
27 Wersja 6: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 6
28 Wersja 7: 'Utworzenie relacji pomiędzy kategoriami a obiektami'
29
30 (1 rows affected)
31 Wersja 7: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 7
32 Wersja 8: 'Utworzenie tabeli z użytkownikami'
33
34 (1 rows affected)
35 Wersja 8: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 8
```

¹Aby wywołanie funkcji `RAISERROR` przekazało kontrolę do bloku `CATCH`, parametr `severity` musi mieć wartość z zakresu od 11 do 19. Wartości poniżej nie powodują przerywania skryptu, a wartości powyżej terminują połączenie z bazą danych.

```

36 Wersja 9: 'Utworzenie tabeli z najmami'
37
38 (1 rows affected)
39 Wersja 9: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 9
40 Wersja 10: 'Utworzenie indeksu unikatowego w tabeli z najemcami'
41
42 (1 rows affected)
43 Wersja 10: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 10
44 Wersja 11: 'Utworzenie relacji pomiędzy użytkownikami a najmami'
45
46 (1 rows affected)
47 Wersja 11: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 11
48 Wersja 12: 'Utworzenie relacji pomiędzy obiektami a najmami'
49
50 (1 rows affected)
51 Wersja 12: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 12
52 Wersja 13: 'Utworzenie indeksu unikatowego w tabeli z użytkownikami'
53
54 (1 rows affected)
55 Wersja 13: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji 13

```

Listing 2: Wynik uruchomienia całego skryptu tworzącego w trybie wsadowym

5.3 Tabele

Wszystkie tabele są tworzone przez 13 skryptów SQL:

- Utworzenie tabeli z miastami
- Utworzenie tabeli z dzielnicami
- Utworzenie relacji pomiędzy miastami a dzielnicami
- Utworzenie tabeli z kategoriami
- Utworzenie tabeli z obiektami
- Utworzenie relacji pomiędzy dzielnicami a obiektami
- Utworzenie relacji pomiędzy kategoriami a obiektami
- Utworzenie tabeli z użytkownikami
- Utworzenie tabeli z najmami
- Utworzenie indeksu unikatowego w tabeli z najemcami
- Utworzenie relacji pomiędzy użytkownikami a najmami
- Utworzenie relacji pomiędzy obiektami a najmami
- Utworzenie indeksu unikatowego w tabeli z użytkownikami

Tworzenie relacji pomiędzy tabelami oraz indeksów zostało oddzielone od operacji tworzenia poszczególnych tabel - celem tego działania jest lepsza organizacja skryptów. Dodatkowo oddzielając te operacje, w przypadku wystąpienia jakiegoś błędu jesteśmy w stanie określić co i gdzie się "wysypało".

Ze względu na to aby nie zajmować zbyt dużo miejsca, poniżej zostaną przedstawione tylko najważniejsze z powyższych skryptów.

5.3.1 Utworzenie tabeli z obiektami

Ponieważ polecenia `CREATE DEFAULT` oraz `CREATE RULE` zostały zdeprecjonowane i w kolejnych wersjach SQL Serwera prawdopodobnie zostaną usunięte zdecydowałem się umieścić wartości domyślne oraz reguły sprawdzające w definicjach konkretnych tabel.

W wyniku projektowania zostało dodatkowo ustalone że `dzienna_stawka_najmu` musi być większa od 0.

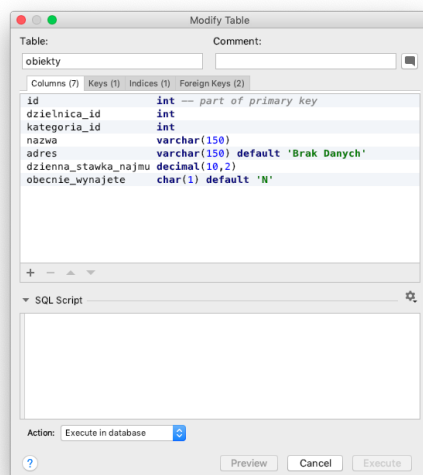
Status obiektu znajdujący się w polu `obecnie_wynajete` może przyjmować dwie wartości T oraz N - odpowiednio dla obiekty wynajętego oraz wolnego.

```

1 CREATE TABLE obiekty (
2   id INT PRIMARY KEY NOT NULL IDENTITY (1, 1),
3
4   dzielnica_id INT NOT NULL,
5   kategoria_id INT NOT NULL,
6
7   nazwa VARCHAR(150) NOT NULL,
8   adres VARCHAR(150) NOT NULL DEFAULT 'Brak Danych',
9   dzienna_stawka_najmu DECIMAL(10, 2) NOT NULL CHECK (dzienna_stawka_najmu > 0),
10
11   obecnie_wynajete CHAR(1) NOT NULL DEFAULT 'N' CHECK (obecnie_wynajete IN ('T', 'N')),
12 );

```

Listing 3: Skrypt tworzący tabelę obiekty



Rysunek 2: Tabela obiekty wyświetlona w programie [DataGrip](#)

5.3.2 Utworzenie tabeli z najmami i Utworzenie indeksu unikatowego w tabeli z najemcami

Przyjmujemy że domyślną datą rozpoczęcia najmu jest data jego dodania do bazy.

```

1 CREATE TABLE najmy (
2   id INT PRIMARY KEY NOT NULL IDENTITY (1, 1),
3
4   uzytkownik_id INT NOT NULL,
5   obiekt_id INT NOT NULL,
6
7   data_roz poczecie DATE NOT NULL DEFAULT getdate(),
8   data_zakonczenia DATE NULL,
9   koszt DECIMAL(15, 2) NULL,
10 );

```

Listing 4: Skrypt tworzący tabelę obiekty

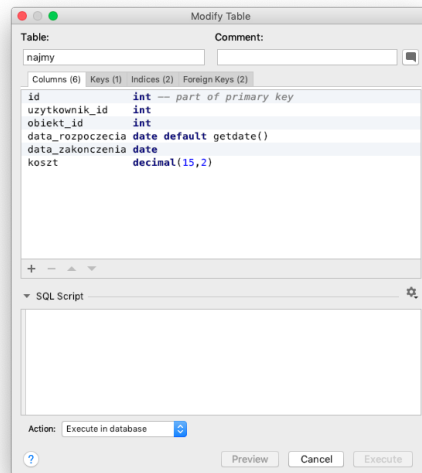
Ponieważ w założeniach projektowych przyjeśliśmy że najem następuje od północy do godziny 23:59, to zostało wykorzystane pole typu DATE. Więc na przykład jeśli klient wynajmie dany obiekt tylko na jeden dzień to w polach data_roz poczecie oraz data_zakonczenia wartość będzie ta sama.

```

1 CREATE UNIQUE INDEX najemcy-ui
2 ON najmy (uzytkownik_id, obiekt_id, data_roz poczecie);

```

Listing 5: Skrypt tworzący indeks unikatowy w tabeli obiekty



Rysunek 3: Tabela `najmy` wyświetlona w programie [DataGrip](#)

5.4 Widoki

5.5 Funkcje Skalarne

5.6 Funkcje Tabelarne

5.7 Triggery

5.8 Procedury Składowane

5.9 Skrypty w oparciu o kursory

5.10 Inne poznane obiekty, własności bazy danych

6 Skrypt tworzący dane testowe

ZRZUT EKRAŃÓW Z KOMUNIKATÓW PO WSTAWIANIU DANYCH [np. ILE DANYCH DO KAŻDEJ TABELI WPROWADZONO];

7 Skrypt tworzący użytkowników i nadający uprawnienia

8. SKRYPT TWORZACY UZYTKOWNIKOW BAZY DANYCH ;KONTA, UPRAWNIENIA itp;, W FORMIE: a. OPIS UZYTKOWNIKOW ;ICH ZADANIA;, b. SKŁADNIA SKRYPTU c. ZRZUTY EKRANU Z: UTWORZENIA UZYTKOWNIKOW, UPRAWNIENIA, TEST MOZLIWOSCI/NIE MOZLIWOSCI KAZDEGO Z NICH

8 Skrypt usuwający obiekty z bazy danych

8.1 Wynik uruchomienia całego skryptu w trybie wsadowym

Jak widać na listingu 6, skrypt podaje bardzo dokładne informacje na temat aktualnie wykonywanej operacji. W przypadku tego skryptu, operacje są wykonywane w odwrotnej kolejności niż w skrypcie tworzącym z listingu 2.

```
1 Wersja 13: 'Utworzenie indeksu unikatowego w tabeli z uzytkownikami'
2
3 (1 rows affected)
4 Wersja 13: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 12
5 Wersja 12: 'Utworzenie relacji pomiedzy obiektami a najmami'
6
7 (1 rows affected)
8 Wersja 12: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 11
9 Wersja 11: 'Utworzenie relacji pomiedzy uzytkownikami a najmami'
10
11 (1 rows affected)
12 Wersja 11: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 10
13 Wersja 10: 'Utworzenie indeksu unikatowego w tabeli z najemcami'
14
15 (1 rows affected)
16 Wersja 10: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 9
17 Wersja 9: 'Utworzenie tabeli z najmami'
18
19 (1 rows affected)
20 Wersja 9: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 8
21 Wersja 8: 'Utworzenie tabeli z uzytkownikami'
22
23 (1 rows affected)
24 Wersja 8: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 7
25 Wersja 7: 'Utworzenie relacji pomiedzy kategoriami a obiektami'
26
27 (1 rows affected)
28 Wersja 7: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 6
29 Wersja 6: 'Utworzenie relacji pomiedzy dzielnicami a obiektami'
30
31 (1 rows affected)
32 Wersja 6: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 5
33 Wersja 5: 'Utworzenie tabeli z obiektami'
34
35 (1 rows affected)
36 Wersja 5: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 4
37 Wersja 4: 'Utworzenie tabeli z kategoriami'
38
39 (1 rows affected)
40 Wersja 4: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 3
41 Wersja 3: 'Utworzenie relacji pomiedzy miastami a dzielnicami'
42
43 (1 rows affected)
44 Wersja 3: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 2
45 Wersja 2: 'Utworzenie tabeli z dzielnicami'
46
47 (1 rows affected)
48 Wersja 2: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 1
49 Wersja 1: 'Utworzenie tabeli z miastami'
50
51 (1 rows affected)
52 Wersja 1: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji 0
53 Tabela wersjonowania zostala skasowana
```

Listing 6: Wynik uruchomienia całego skryptu usuwającego w trybie wsadowym

9 Skrypt usuwający użytkowników i uprawnienia

10 Skrypt usuwający dane testowe