

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	2
5	Skrypt tworzący obiekty w bazie danych	3
5.1	Model wersjonowania bazy danych	3
5.2	Wynik uruchomienia całego skryptu w trybie wsadowym	3
5.3	Tabele	4
5.3.1	Utworzenie tabeli z obiektami	5
5.3.2	Utworzenie tabeli z najmami	6
5.4	Widoki	6
5.5	Funkcje Skalarne	6
5.6	Funkcje Tabelarne	6
5.7	Triggery	6
5.8	Procedury Składowane	6
5.9	Skrypty w oparciu o kursory	6
5.10	Inne poznane obiekty, własności bazy danych	6
6	Skrypt tworzący dane testowe	6
7	Skrypt tworzący użytkowników i nadający uprawnienia	6
8	Skrypt usuwający obiekty z bazy danych	6
8.1	Wynik uruchomienia całego skryptu w trybie wsadowym	6
9	Skrypt usuwający użytkowników i uprawnienia	7
10	Skrypt usuwający dane testowe	7
11	Przykłady kodu LaTeX	7
11.1	Tytuł podrozdziału	7
11.1.1	Tytuł podpodrozdziału	7
11.1.2	Tytuł podpodrozdziału 2	8

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, imie 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](#)).

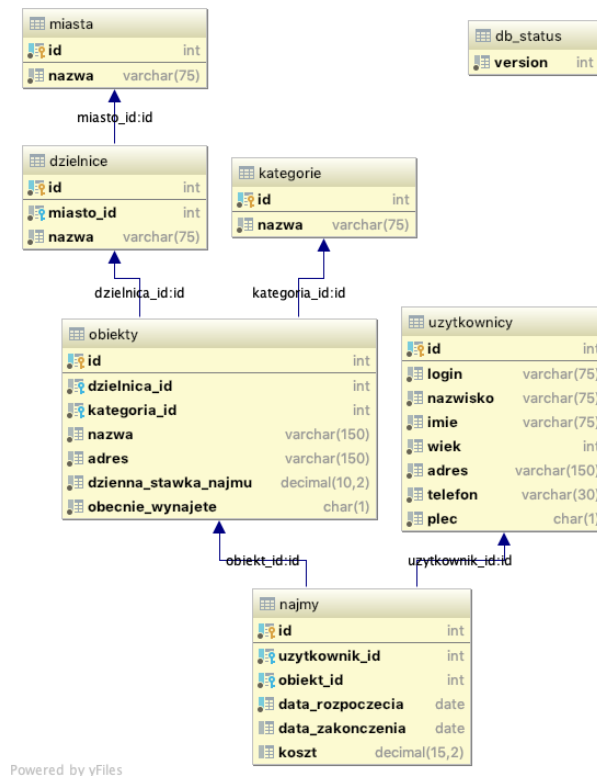
Jako aplikację służącą do łączenia się i wykonywania poleceń wykorzystane zostały aplikacje:

- 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



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`.

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

```

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
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

- Tworzenie relacji pomiędzy kategoriami a obiektami
- Tworzenie tabeli z użytkownikami
- Tworzenie tabeli z najmami
- Tworzenie indeksu unikatowego w tabeli z najemcami
- Tworzenie relacji pomiędzy użytkownikami a najmami
- Tworzenie relacji pomiędzy obiektami a najmami
- Tworzenie 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".

5.3.1 Tworzenie 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.

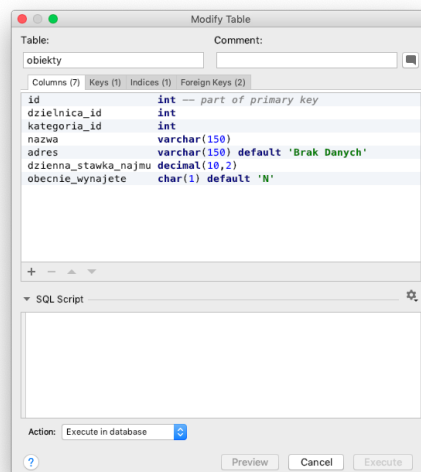
Status obiektu znajdujący się w polu `obecnie_wynajete` może przyjmować dwie wartości T oraz N - odpowiednio dla obiektu wynajętego oraz wolnego.

```

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



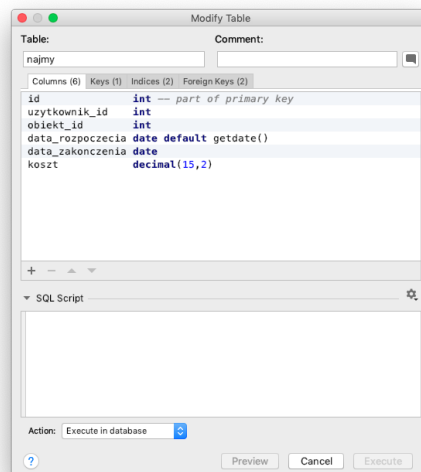
Rysunek 2: Tabela obiekty wyświetlona w programie DataGrip

5.3.2 Utworzenie tabeli z najmami

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_rozpoczecia DATE NOT NULL DEFAULT getdate(),  
8   data_zakonczenia DATE NULL,  
9   koszt DECIMAL(15, 2) NULL,  
10 );
```

Listing 4: Skrypt tworzący tabelę 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 EKRANOW Z KOMUNIKATOW PO WSTAWIANIU DANYCH įnp. ILE DANYCH DO KAZDEJ TABELI WPROWADZONO;

7 Skrypt tworzący użytkowników i nadający uprawnienia

8. SKRYPT TWORZACY UZYTKOWNIKOW BAZY DANYCH įKONTA, UPRAWNIENIA itp. W FORMIE: a. OPIS UZYTKOWNIKOW įICH 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 5, 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 5: Wynik uruchomienia całego skryptu usuwającego w trybie wsadowym

9 Skrypt usuwający użytkowników i uprawnienia

10 Skrypt usuwający dane testowe

11 Przykłady kodu LaTeX

Projekt systemu² zawiera założenia do bazy danych przechowującej informacje³ potrzebne

11.1 Tytuł podrozdziału

11.1.1 Tytuł podpodrozdziału

Pogrubiony tekst treść pisma technicznego *tekst do pochylenia* [Something Linky](#)

- pierwszy element
- drugi element
- trzeci element...

11.1.2 Tytuł podpodrozdziału 2

1. pierwszy element
2. drugi element
3. trzeci element...

```
1 BEGIN TRY
2
3
4
5 — Przygotowanie tabeli do przechowywania statusu (wersji) bazy danych
6 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
7     BEGIN
8         PRINT 'Tabela wersjonowania juz istnieje';
9     END
10 ELSE
11     BEGIN
12         CREATE TABLE db_status (
13             version INT NOT NULL DEFAULT (0)
14         )
15
16         INSERT INTO db_status (version) VALUES (0);
17
18         PRINT 'Tabela wersjonowania zostala utworzona';
19     END
20
21
22
23
24
25
26
27
28 PRINT 'Wersja 1: ''Utworzenie tabeli z miastami''
29 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
30     BEGIN
31         IF EXISTS(SELECT * FROM db_status WHERE version = 0)
32             BEGIN
33                 EXEC dbo.sp_executesql @statement = N'
34                     CREATE TABLE miasta (
35                         id INT PRIMARY KEY NOT NULL IDENTITY (1, 1),
36                         nazwa VARCHAR(75) NOT NULL
37                     );
38                 '
39
40                 UPDATE db_status SET version = 1 WHERE version = 0;
```

²footnotes working fine

³footnotes working fine


```

41 PRINT 'Wersja 1: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji
1';
42 END
43 ELSE
44 BEGIN
45 IF EXISTS(SELECT * FROM db_status WHERE version < 0)
46 BEGIN
47 RAISERROR ('Wersja 1: Baza danych jest w za niskiej wersji (wymagana jest wersja
0) aby zainstalować migrację', 11, 2);
48 END
49 ELSE
50 BEGIN
51 PRINT 'Wersja 1: Migracja już została zainstalowana wcześniej';
52 END
53 END
54 END
55 ELSE
56 BEGIN
57 RAISERROR ('Wersja 1: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
58 END
59
60
61
62
63
64
65
66
67 PRINT 'Wersja 2: ''Utworzenie tabeli z dzielnicami''
68 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
69 BEGIN
70 IF EXISTS(SELECT * FROM db_status WHERE version = 1)
71 BEGIN
72 EXEC dbo.sp_executesql @statement = N'
73 CREATE TABLE dzielnice (
74 id INT PRIMARY KEY NOT NULL IDENTITY (1, 1),
75 miasto_id INT NOT NULL,
76 nazwa VARCHAR(75) NOT NULL
77 );
78
79
80 UPDATE db_status SET version = 2 WHERE version = 1;
81 PRINT 'Wersja 2: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji
2';
82 END
83 ELSE
84 BEGIN
85 IF EXISTS(SELECT * FROM db_status WHERE version < 1)
86 BEGIN
87 RAISERROR ('Wersja 2: Baza danych jest w za niskiej wersji (wymagana jest wersja
1) aby zainstalować migrację', 11, 2);
88 END
89 ELSE
90 BEGIN
91 PRINT 'Wersja 2: Migracja już została zainstalowana wcześniej';
92 END
93 END
94 END
95 ELSE
96 BEGIN
97 RAISERROR ('Wersja 2: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
98 END
99
100
101
102
103
104
105
106
107 PRINT 'Wersja 3: ''Utworzenie relacji pomiędzy miastami a dzielnicami''
108 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
109 BEGIN
110 IF EXISTS(SELECT * FROM db_status WHERE version = 2)
111 BEGIN

```

```

112 EXEC dbo.sp_executesql @statement = N'
113     ALTER TABLE dzielnice
114     ADD CONSTRAINT dzielnice_miasta_fk
115     FOREIGN KEY (miasto_id)
116     REFERENCES miasta(id);
117
118
119     UPDATE db_status SET version = 3 WHERE version = 2;
120     PRINT 'Wersja 3: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji
121     3';
122     END
123 ELSE
124     BEGIN
125     IF EXISTS(SELECT * FROM db_status WHERE version < 2)
126     BEGIN
127         RAISERROR ('Wersja 3: Baza danych jest w za niskiej wersji (wymagana jest wersja
128         2) aby zainstalować migrację', 11, 2);
129     END
130     ELSE
131     BEGIN
132         PRINT 'Wersja 3: Migracja już została zainstalowana wcześniej';
133     END
134     END
135 ELSE
136     BEGIN
137     RAISERROR ('Wersja 3: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
138     END
139
140
141
142
143
144
145 PRINT 'Wersja 4: ''Utworzenie tabeli z kategoriami''
146 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
147 BEGIN
148     IF EXISTS(SELECT * FROM db_status WHERE version = 3)
149     BEGIN
150         EXEC dbo.sp_executesql @statement = N'
151             CREATE TABLE kategorie (
152                 id INT PRIMARY KEY NOT NULL IDENTITY (1, 1),
153                 nazwa VARCHAR(75) NOT NULL
154             );
155         ,
156
157
158         UPDATE db_status SET version = 4 WHERE version = 3;
159         PRINT 'Wersja 4: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji
160         4';
161     END
162     ELSE
163     BEGIN
164     IF EXISTS(SELECT * FROM db_status WHERE version < 3)
165     BEGIN
166         RAISERROR ('Wersja 4: Baza danych jest w za niskiej wersji (wymagana jest wersja
167         3) aby zainstalować migrację', 11, 2);
168     END
169     ELSE
170     BEGIN
171         PRINT 'Wersja 4: Migracja już została zainstalowana wcześniej';
172     END
173     END
174     END
175 ELSE
176     BEGIN
177     RAISERROR ('Wersja 4: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
178     END
179
180
181
182

```

```

183
184
185 PRINT 'Wersja 5: ''Utworzenie tabeli z obiektami'''
186 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
187 BEGIN
188     IF EXISTS(SELECT * FROM db_status WHERE version = 4)
189     BEGIN
190         EXEC dbo.sp_executesql @statement = N'
191             CREATE TABLE objekty (
192                 id INT PRIMARY KEY NOT NULL IDENTITY (1, 1),
193
194                 dzielnica_id INT NOT NULL,
195                 kategoria_id INT NOT NULL,
196
197                 nazwa VARCHAR(150) NOT NULL,
198                 adres VARCHAR(150) NOT NULL DEFAULT ''Brak Danych'',
199                 dzienna_stawka_najmu DECIMAL(10, 2) NOT NULL CHECK (dzienna_stawka_najmu > 0),
200
201                 obecnie_wynajete CHAR(1) NOT NULL DEFAULT ''N'' CHECK (obecnie_wynajete IN (''T''
202                 , ''N'')),
203             );
204
205             UPDATE db_status SET version = 5 WHERE version = 4;
206             PRINT 'Wersja 5: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji
207             5';
208         END
209     ELSE
210     BEGIN
211         IF EXISTS(SELECT * FROM db_status WHERE version < 4)
212         BEGIN
213             RAISERROR ('Wersja 5: Baza danych jest w za niskiej wersji (wymagana jest wersja
214             4) aby zainstalować migrację', 11, 2);
215         END
216     ELSE
217     BEGIN
218         PRINT 'Wersja 5: Migracja już została zainstalowana wcześniej';
219     END
220 END
221 ELSE
222 BEGIN
223     RAISERROR ('Wersja 5: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
224 END
225
226
227
228
229
230
231
232 PRINT 'Wersja 6: ''Utworzenie relacji pomiędzy dzielnicami a obiektami'''
233 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
234 BEGIN
235     IF EXISTS(SELECT * FROM db_status WHERE version = 5)
236     BEGIN
237         EXEC dbo.sp_executesql @statement = N'
238             ALTER TABLE objekty
239             ADD CONSTRAINT objekty_dzielnice_fk
240             FOREIGN KEY (dzielnica_id)
241             REFERENCES dzielnice(id);
242         ,
243
244         UPDATE db_status SET version = 6 WHERE version = 5;
245         PRINT 'Wersja 6: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji
246         6';
247     END
248 ELSE
249 BEGIN
250     IF EXISTS(SELECT * FROM db_status WHERE version < 5)
251     BEGIN
252         RAISERROR ('Wersja 6: Baza danych jest w za niskiej wersji (wymagana jest wersja
253         5) aby zainstalować migrację', 11, 2);
254     END
255

```

```

253         ELSE
254             BEGIN
255                 PRINT 'Wersja 6: Migracja ju została zainstalowana wcześniej';
256             END
257         END
258     END
259 ELSE
260     BEGIN
261         RAISERROR ('Wersja 6: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
262     END
263
264
265
266
267
268
269
270
271 PRINT 'Wersja 7: ''Utworzenie relacji pomiedzy kategoriami a obiektami''
272 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
273     BEGIN
274         IF EXISTS(SELECT * FROM db_status WHERE version = 6)
275             BEGIN
276                 EXEC dbo.sp_executesql @statement = N'
277                     ALTER TABLE objekty
278                     ADD CONSTRAINT objekty_kategorie_fk
279                     FOREIGN KEY (kategoria_id)
280                     REFERENCES kategorie(id);
281                 ,
282
283                 UPDATE db_status SET version = 7 WHERE version = 6;
284                 PRINT 'Wersja 7: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji
285                 7';
286             END
287         ELSE
288             BEGIN
289                 IF EXISTS(SELECT * FROM db_status WHERE version < 6)
290                     BEGIN
291                         RAISERROR ('Wersja 7: Baza danych jest w za niskiej wersji (wymagana jest wersja
292                         6) aby zainstalowac migracje', 11, 2);
293                     END
294                 ELSE
295                     BEGIN
296                         PRINT 'Wersja 7: Migracja ju została zainstalowana wcześniej';
297                     END
298             END
299         END
300     ELSE
301     BEGIN
302         RAISERROR ('Wersja 7: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
303     END
304
305
306
307
308
309
310 PRINT 'Wersja 8: ''Utworzenie tabeli z uzytkownikami''
311 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
312     BEGIN
313         IF EXISTS(SELECT * FROM db_status WHERE version = 7)
314             BEGIN
315                 EXEC dbo.sp_executesql @statement = N'
316                     CREATE TABLE uzytkownicy (
317                         id INT PRIMARY KEY NOT NULL IDENTITY (1, 1),
318
319                         login VARCHAR(75) NOT NULL,
320
321                         nazwisko VARCHAR(75) NOT NULL,
322                         imie VARCHAR(75) NOT NULL,
323                         wiek INT NOT NULL CHECK (wiek BETWEEN 1 AND 100),
324                         adres VARCHAR(150) NOT NULL DEFAULT ''Brak Danych'',
325                         telefon VARCHAR(30) NOT NULL DEFAULT ''Brak Danych'',

```

```

326         plec CHAR(1) NOT NULL CHECK (plec IN ('K', 'M')),
327     );
328
329
330     UPDATE db_status SET version = 8 WHERE version = 7;
331     PRINT 'Wersja 8: Migracja została zainstalowana pomysłnie – teraz baza jest w wersji
8';
332     END
333 ELSE
334     BEGIN
335         IF EXISTS(SELECT * FROM db_status WHERE version < 7)
336             BEGIN
337                 RAISERROR ('Wersja 8: Baza danych jest w za niskiej wersji (wymagana jest wersja
7) aby zainstalować migrację', 11, 2);
338             END
339         ELSE
340             BEGIN
341                 PRINT 'Wersja 8: Migracja już została zainstalowana wcześniej';
342             END
343     END
344 END
345 ELSE
346     BEGIN
347         RAISERROR ('Wersja 8: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
348     END
349
350
351
352
353
354
355
356
357 PRINT 'Wersja 9: ''Utworzenie tabeli z najmami''';
358 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
359     BEGIN
360         IF EXISTS(SELECT * FROM db_status WHERE version = 8)
361             BEGIN
362                 EXEC dbo.sp_executesql @statement = N'
363                 CREATE TABLE najmy (
364                     id INT PRIMARY KEY NOT NULL IDENTITY (1, 1),
365
366                     uzytkownik_id INT NOT NULL,
367                     obiekt_id INT NOT NULL,
368
369                     data_roz poczenia DATE NOT NULL DEFAULT getdate(),
370                     data_zakonczenia DATE NULL,
371                     koszt DECIMAL(15, 2) NULL,
372                 );
373
374
375                 UPDATE db_status SET version = 9 WHERE version = 8;
376                 PRINT 'Wersja 9: Migracja została zainstalowana pomysłnie – teraz baza jest w wersji
9';
377             END
378         ELSE
379             BEGIN
380                 IF EXISTS(SELECT * FROM db_status WHERE version < 8)
381                     BEGIN
382                         RAISERROR ('Wersja 9: Baza danych jest w za niskiej wersji (wymagana jest wersja
8) aby zainstalować migrację', 11, 2);
383                     END
384                 ELSE
385                     BEGIN
386                         PRINT 'Wersja 9: Migracja już została zainstalowana wcześniej';
387                     END
388             END
389     END
390 ELSE
391     BEGIN
392         RAISERROR ('Wersja 9: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
393     END
394
395
396

```

```

397
398
399
400
401
402 PRINT 'Wersja 10: ''Utworzenie indeksu unikatowego w tabeli z najemcami'''
403 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
404 BEGIN
405     IF EXISTS(SELECT * FROM db_status WHERE version = 9)
406     BEGIN
407         EXEC dbo.sp_executesql @statement = N'
408             CREATE UNIQUE INDEX najemcy-ui
409             ON najmy (uzytkownik_id, obiekt_id, data_roz pocze cia);
410         '
411
412         UPDATE db_status SET version = 10 WHERE version = 9;
413         PRINT 'Wersja 10: Migracja zostala zainstalowana pomyslnie – teraz baza jest w wersji
414         10';
415         END
416     ELSE
417     BEGIN
418         IF EXISTS(SELECT * FROM db_status WHERE version < 9)
419         BEGIN
420             RAISERROR ('Wersja 10: Baza danych jest w za niskiej wersji (wymagana jest wersja
421             9) aby zainstalowac migracje', 11, 2);
422             END
423         ELSE
424         BEGIN
425             PRINT 'Wersja 10: Migracja ju  zostala zainstalowana wzczesniej';
426             END
427         END
428     END
429 END
430 ELSE
431 BEGIN
432     RAISERROR ('Wersja 10: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
433 END
434
435
436
437
438
439 PRINT 'Wersja 11: ''Utworzenie relacji pomiedzy uzytkownikami a najmami'''
440 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
441 BEGIN
442     IF EXISTS(SELECT * FROM db_status WHERE version = 10)
443     BEGIN
444         EXEC dbo.sp_executesql @statement = N'
445             ALTER TABLE najmy
446             ADD CONSTRAINT najmy_uzytkownicy_fk
447             FOREIGN KEY (uzytkownik_id)
448             REFERENCES uzytkownicy(id);
449         '
450
451         UPDATE db_status SET version = 11 WHERE version = 10;
452         PRINT 'Wersja 11: Migracja zostala zainstalowana pomyslnie – teraz baza jest w wersji
453         11';
454         END
455     ELSE
456     BEGIN
457         IF EXISTS(SELECT * FROM db_status WHERE version < 10)
458         BEGIN
459             RAISERROR ('Wersja 11: Baza danych jest w za niskiej wersji (wymagana jest wersja
460             10) aby zainstalowac migracje', 11, 2);
461             END
462         ELSE
463         BEGIN
464             PRINT 'Wersja 11: Migracja ju  zostala zainstalowana wzczesniej';
465             END
466         END
467     END
468 END
469 ELSE
470 BEGIN

```

```

468 RAISERROR ( 'Wersja 11: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
469 END
470
471
472
473
474
475
476
477
478 PRINT 'Wersja 12: ''Utworzenie relacji pomiedzy obiektami a najmami'''
479 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
480 BEGIN
481     IF EXISTS(SELECT * FROM db_status WHERE version = 11)
482     BEGIN
483         EXEC dbo.sp_executesql @statement = N'
484             ALTER TABLE najmy
485             ADD CONSTRAINT najmy_obiekty_fk
486             FOREIGN KEY (obiett_id)
487             REFERENCES obiekty(id);
488         '
489
490         UPDATE db_status SET version = 12 WHERE version = 11;
491         PRINT 'Wersja 12: Migracja zostala zainstalowana pomyslnie – teraz baza jest w wersji
12';
492     END
493 ELSE
494     BEGIN
495         IF EXISTS(SELECT * FROM db_status WHERE version < 11)
496         BEGIN
497             RAISERROR ( 'Wersja 12: Baza danych jest w za niskiej wersji (wymagana jest wersja
11) aby zainstalowac migracje', 11, 2);
498         END
499     ELSE
500     BEGIN
501         PRINT 'Wersja 12: Migracja ju  zostala zainstalowana wczesniej';
502     END
503 END
504 END
505 ELSE
506 BEGIN
507     RAISERROR ( 'Wersja 12: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
508 END
509
510
511
512
513
514
515
516
517 PRINT 'Wersja 13: ''Utworzenie indeksu unikatowego w tabeli z uzytkownikami'''
518 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
519 BEGIN
520     IF EXISTS(SELECT * FROM db_status WHERE version = 12)
521     BEGIN
522         EXEC dbo.sp_executesql @statement = N'
523             CREATE UNIQUE INDEX najemcy_ui
524             ON uzytkownicy (login);
525         '
526
527         UPDATE db_status SET version = 13 WHERE version = 12;
528         PRINT 'Wersja 13: Migracja zostala zainstalowana pomyslnie – teraz baza jest w wersji
13';
529     END
530 ELSE
531     BEGIN
532         IF EXISTS(SELECT * FROM db_status WHERE version < 12)
533         BEGIN
534             RAISERROR ( 'Wersja 13: Baza danych jest w za niskiej wersji (wymagana jest wersja
12) aby zainstalowac migracje', 11, 2);
535         END
536     ELSE
537     BEGIN
538         PRINT 'Wersja 13: Migracja ju  zostala zainstalowana wczesniej';

```

```

539         END
540     END
541 END
542 ELSE
543 BEGIN
544     RAISERROR ( 'Wersja 13: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
545 END
546
547
548
549
550
551
552
553
554
555
556
557
558 END TRY
559 BEGIN CATCH
560
561     SELECT
562         ERROR_NUMBER() AS ErrorNumber,
563         ERROR_SEVERITY() AS ErrorSeverity,
564         ERROR_STATE() AS ErrorState,
565         ERROR_PROCEDURE() AS ErrorProcedure,
566         ERROR_LINE() AS ErrorLine,
567         ERROR_MESSAGE() AS ErrorMessage;
568
569 END CATCH;

```

Listing 6: skrypt_tworzacy_obiekty_w_bazie_danych.sql

```

1 BEGIN TRY
2
3
4
5 PRINT 'Wersja 13: ''Utworzenie indeksu unikatowego w tabeli z uzytkownikami''
6 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
7 BEGIN
8     IF EXISTS(SELECT * FROM db_status WHERE version = 13)
9     BEGIN
10         EXEC dbo.sp_executesql @statement = N'
11             DROP INDEX uzytkownicy.najemcy_ui;
12         ,
13
14         UPDATE db_status SET version = 12 WHERE version = 13;
15         PRINT 'Wersja 13: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji
16         12';
17     END
18 ELSE
19 BEGIN
20     IF EXISTS(SELECT * FROM db_status WHERE version > 13)
21     BEGIN
22         RAISERROR ( 'Wersja 13: Baza danych jest w za wysokiej wersji (wymagana jest
23         wersja 13) aby odinstalowac migracje', 11, 2);
24     END
25     ELSE
26     BEGIN
27         PRINT 'Wersja 13: Migracja nie by a wzczesniej zainstalowana lub zostala juz
28         odinstalowana';
29     END
30 END
31 END
32 ELSE
33 BEGIN
34     RAISERROR ( 'Wersja 13: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
35 END
36
37
38
39

```



```

40 PRINT 'Wersja 12: ''Utworzenie relacji pomiedzy obiektami a najmami'''
41 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
42 BEGIN
43     IF EXISTS(SELECT * FROM db_status WHERE version = 12)
44     BEGIN
45         EXEC dbo.sp_executesql @statement = N'
46             ALTER TABLE najmy DROP CONSTRAINT najmy_obiekty_fk;
47         ',
48
49
50         UPDATE db_status SET version = 11 WHERE version = 12;
51         PRINT 'Wersja 12: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji
52             11';
53     END
54 ELSE
55 BEGIN
56     IF EXISTS(SELECT * FROM db_status WHERE version > 12)
57     BEGIN
58         RAISERROR ('Wersja 12: Baza danych jest w za wysokiej wersji (wymagana jest
59             wersja 12) aby odinstalowac migracje', 11, 2);
60     END
61     ELSE
62     BEGIN
63         PRINT 'Wersja 12: Migracja nie by a wzczesniej zainstalowana lub zostala juz
64             odinstalowana';
65     END
66 END
67 END
68 ELSE
69 BEGIN
70     RAISERROR ('Wersja 12: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
71 END
72
73
74
75
76
77 PRINT 'Wersja 11: ''Utworzenie relacji pomiedzy uzytkownikami a najmami'''
78 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
79 BEGIN
80     IF EXISTS(SELECT * FROM db_status WHERE version = 11)
81     BEGIN
82         EXEC dbo.sp_executesql @statement = N'
83             ALTER TABLE najmy DROP CONSTRAINT najmy_uzytkownicy_fk;
84         ',
85
86
87         UPDATE db_status SET version = 10 WHERE version = 11;
88         PRINT 'Wersja 11: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji
89             10';
90     END
91     ELSE
92     BEGIN
93         IF EXISTS(SELECT * FROM db_status WHERE version > 11)
94         BEGIN
95             RAISERROR ('Wersja 11: Baza danych jest w za wysokiej wersji (wymagana jest
96                 wersja 11) aby odinstalowac migracje', 11, 2);
97             END
98             ELSE
99             BEGIN
100                 PRINT 'Wersja 11: Migracja nie by a wzczesniej zainstalowana lub zostala juz
101                     odinstalowana';
102             END
103         END
104     END
105 ELSE
106 BEGIN
107     RAISERROR ('Wersja 11: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
108 END

```

```

109
110
111
112
113 PRINT 'Wersja 10: ''Utworzenie indeksu unikatowego w tabeli z najemcami''
114 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
115 BEGIN
116     IF EXISTS(SELECT * FROM db_status WHERE version = 10)
117     BEGIN
118         EXEC dbo.sp_executesql @statement = N'
119             DROP INDEX najmy.najemcy-ui;
120         '
121
122         UPDATE db_status SET version = 9 WHERE version = 10;
123         PRINT 'Wersja 10: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji
124             9';
125     END
126 ELSE
127 BEGIN
128     IF EXISTS(SELECT * FROM db_status WHERE version > 10)
129     BEGIN
130         RAISERROR ('Wersja 10: Baza danych jest w za wysokiej wersji (wymagana jest
131             wersja 10) aby odinstalować migrację', 11, 2);
132     END
133     ELSE
134     BEGIN
135         PRINT 'Wersja 10: Migracja nie była wcześniej zainstalowana lub została już
136             odinstalowana';
137     END
138 END
139 ELSE
140 BEGIN
141     RAISERROR ('Wersja 10: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
142 END
143
144
145
146
147
148
149 PRINT 'Wersja 9: ''Utworzenie tabeli z najmami''
150 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
151 BEGIN
152     IF EXISTS(SELECT * FROM db_status WHERE version = 9)
153     BEGIN
154         EXEC dbo.sp_executesql @statement = N'
155             DROP TABLE najmy;
156         '
157
158         UPDATE db_status SET version = 8 WHERE version = 9;
159         PRINT 'Wersja 9: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji
160             8';
161     END
162 ELSE
163 BEGIN
164     IF EXISTS(SELECT * FROM db_status WHERE version > 9)
165     BEGIN
166         RAISERROR ('Wersja 9: Baza danych jest w za wysokiej wersji (wymagana jest wersja
167             9) aby odinstalować migrację', 11, 2);
168     END
169     ELSE
170     BEGIN
171         PRINT 'Wersja 9: Migracja nie była wcześniej zainstalowana lub została już
172             odinstalowana';
173     END
174 END
175 ELSE
176 BEGIN
177     RAISERROR ('Wersja 9: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
178 END

```

```

178
179
180
181
182
183
184
185 PRINT 'Wersja 8: ''Utworzenie tabeli z uzytkownikami'''
186 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
187 BEGIN
188     IF EXISTS(SELECT * FROM db_status WHERE version = 8)
189     BEGIN
190         EXEC dbo.sp_executesql @statement = N'
191             DROP TABLE uzytkownicy;
192         '
193
194         UPDATE db_status SET version = 7 WHERE version = 8;
195         PRINT 'Wersja 8: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji
196             7';
197     END
198 ELSE
199 BEGIN
200     IF EXISTS(SELECT * FROM db_status WHERE version > 8)
201     BEGIN
202         RAISERROR ('Wersja 8: Baza danych jest w za wysokiej wersji (wymagana jest wersja
203             8) aby odinstalowac migracje', 11, 2);
204     END
205 ELSE
206 BEGIN
207     PRINT 'Wersja 8: Migracja nie by a wzczesniej zainstalowana lub zostala juz
208     odinstalowana';
209 END
210 END
211 ELSE
212 BEGIN
213     RAISERROR ('Wersja 8: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
214 END
215
216
217
218
219
220
221 PRINT 'Wersja 7: ''Utworzenie relacji pomiedzy kategoriami a obiektami'''
222 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
223 BEGIN
224     IF EXISTS(SELECT * FROM db_status WHERE version = 7)
225     BEGIN
226         EXEC dbo.sp_executesql @statement = N'
227             ALTER TABLE obiekty DROP CONSTRAINT obiekty_kategorie_fk;
228         '
229
230         UPDATE db_status SET version = 6 WHERE version = 7;
231         PRINT 'Wersja 7: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji
232             6';
233     END
234 ELSE
235 BEGIN
236     IF EXISTS(SELECT * FROM db_status WHERE version > 7)
237     BEGIN
238         RAISERROR ('Wersja 7: Baza danych jest w za wysokiej wersji (wymagana jest wersja
239             7) aby odinstalowac migracje', 11, 2);
240     END
241 ELSE
242 BEGIN
243     PRINT 'Wersja 7: Migracja nie by a wzczesniej zainstalowana lub zostala juz
244     odinstalowana';
245 END
246 END
247 ELSE
248 BEGIN

```

```

247 RAISERROR ( 'Wersja 7: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
248 END
249
250
251
252
253
254
255
256
257 PRINT 'Wersja 6: ''Utworzenie relacji pomiedzy dzielnicami a obiektami'''
258 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
259 BEGIN
260     IF EXISTS(SELECT * FROM db_status WHERE version = 6)
261     BEGIN
262         EXEC dbo.sp_executesql @statement = N'
263             ALTER TABLE obiekty DROP CONSTRAINT obiekty_dzielnice_fk;
264         ,
265
266         UPDATE db_status SET version = 5 WHERE version = 6;
267         PRINT 'Wersja 6: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji
268         5';
269     END
270 ELSE
271 BEGIN
272     IF EXISTS(SELECT * FROM db_status WHERE version > 6)
273     BEGIN
274         RAISERROR ( 'Wersja 6: Baza danych jest w za wysokiej wersji (wymagana jest wersja
275         6) aby odinstalowac migracje', 11, 2);
276     END
277 ELSE
278 BEGIN
279     PRINT 'Wersja 6: Migracja nie by a wzczesniej zainstalowana lub zostala juz
280     odinstalowana';
281 END
282 END
283 END
284 ELSE
285 BEGIN
286     RAISERROR ( 'Wersja 6: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
287 END
288
289
290
291
292
293 PRINT 'Wersja 5: ''Utworzenie tabeli z obiektami'''
294 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
295 BEGIN
296     IF EXISTS(SELECT * FROM db_status WHERE version = 5)
297     BEGIN
298         EXEC dbo.sp_executesql @statement = N'
299             DROP TABLE obiekty;
300         ,
301
302         UPDATE db_status SET version = 4 WHERE version = 5;
303         PRINT 'Wersja 5: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji
304         4';
305     END
306 ELSE
307 BEGIN
308     IF EXISTS(SELECT * FROM db_status WHERE version > 5)
309     BEGIN
310         RAISERROR ( 'Wersja 5: Baza danych jest w za wysokiej wersji (wymagana jest wersja
311         5) aby odinstalowac migracje', 11, 2);
312     END
313 ELSE
314 BEGIN
315     PRINT 'Wersja 5: Migracja nie by a wzczesniej zainstalowana lub zostala juz
316     odinstalowana';
317 END
318 END

```

```

316 END
317 ELSE
318 BEGIN
319     RAISERROR ('Wersja 5: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
320 END
321
322
323
324
325
326
327
328
329 PRINT 'Wersja 4: ''Utworzenie tabeli z kategoriami'''
330 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
331 BEGIN
332     IF EXISTS(SELECT * FROM db_status WHERE version = 4)
333     BEGIN
334         EXEC dbo.sp_executesql @statement = N'
335             DROP TABLE kategorie;
336         ',
337
338         UPDATE db_status SET version = 3 WHERE version = 4;
339         PRINT 'Wersja 4: Migracja została odinstalowana pomyslnie – teraz baza jest w wersji
340 3';
341     END
342 ELSE
343 BEGIN
344     IF EXISTS(SELECT * FROM db_status WHERE version > 4)
345     BEGIN
346         RAISERROR ('Wersja 4: Baza danych jest w za wysokiej wersji (wymagana jest wersja
347 4) aby odinstalowac migracje', 11, 2);
348     END
349 ELSE
350 BEGIN
351     PRINT 'Wersja 4: Migracja nie by a wzczesniej zainstalowana lub zostala juz
352 odinstalowana';
353     END
354 END
355 END
356 ELSE
357 BEGIN
358     RAISERROR ('Wersja 4: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
359 END
360
361
362
363
364
365 PRINT 'Wersja 3: ''Utworzenie relacji pomiedzy miastami a dzielnicami'''
366 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
367 BEGIN
368     IF EXISTS(SELECT * FROM db_status WHERE version = 3)
369     BEGIN
370         EXEC dbo.sp_executesql @statement = N'
371             ALTER TABLE dzielnice DROP CONSTRAINT dzielnice_miasta_fk;
372         ',
373
374         UPDATE db_status SET version = 2 WHERE version = 3;
375         PRINT 'Wersja 3: Migracja została odinstalowana pomyslnie – teraz baza jest w wersji
376 2';
377     END
378 ELSE
379 BEGIN
380     IF EXISTS(SELECT * FROM db_status WHERE version > 3)
381     BEGIN
382         RAISERROR ('Wersja 3: Baza danych jest w za wysokiej wersji (wymagana jest wersja
383 3) aby odinstalowac migracje', 11, 2);
384     END
385 ELSE
386 BEGIN
387     PRINT 'Wersja 3: Migracja nie by a wzczesniej zainstalowana lub zostala juz

```

```

386         odinstalowana';
387         END
388     END
389 ELSE
390 BEGIN
391     RAISERROR ('Wersja 3: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
392 END
393
394
395
396
397
398
399
400
401 PRINT 'Wersja 2: ''Utworzenie tabeli z dzielnicami'''
402 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
403 BEGIN
404     IF EXISTS(SELECT * FROM db_status WHERE version = 2)
405     BEGIN
406         EXEC dbo.sp_executesql @statement = N'
407             DROP TABLE dzielnice;
408         ',
409
410         UPDATE db_status SET version = 1 WHERE version = 2;
411         PRINT 'Wersja 2: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji
412             1';
413     END
414 ELSE
415 BEGIN
416     IF EXISTS(SELECT * FROM db_status WHERE version > 2)
417     BEGIN
418         RAISERROR ('Wersja 2: Baza danych jest w za wysokiej wersji (wymagana jest wersja
419             2) aby odinstalować migracje', 11, 2);
420     END
421     ELSE
422     BEGIN
423         PRINT 'Wersja 2: Migracja nie była wcześniej zainstalowana lub została już
424             odinstalowana';
425     END
426 END
427 ELSE
428 BEGIN
429     RAISERROR ('Wersja 2: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
430 END
431
432
433
434
435
436
437 PRINT 'Wersja 1: ''Utworzenie tabeli z miastami'''
438 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
439 BEGIN
440     IF EXISTS(SELECT * FROM db_status WHERE version = 1)
441     BEGIN
442         EXEC dbo.sp_executesql @statement = N'
443             DROP TABLE miasta;
444         ',
445
446         UPDATE db_status SET version = 0 WHERE version = 1;
447         PRINT 'Wersja 1: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji
448             0';
449     END
450 ELSE
451 BEGIN
452     IF EXISTS(SELECT * FROM db_status WHERE version > 1)
453     BEGIN
454         RAISERROR ('Wersja 1: Baza danych jest w za wysokiej wersji (wymagana jest wersja
455             1) aby odinstalować migracje', 11, 2);
456     END
457 END

```

```

455         ELSE
456             BEGIN
457                 PRINT 'Wersja 1: Migracja nie by a wzczesniej zainstalowana lub zostala juz
odinstalowana';
458             END
459         END
460     END
461 ELSE
462     BEGIN
463         RAISERROR ( 'Wersja 1: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
464     END
465
466
467
468
469
470
471
472
473 — Usuni cie tabeli do przechowywania statusu (wersji) bazy danych
474 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
475 BEGIN
476     IF EXISTS(SELECT * FROM db_status WHERE version = 0)
477     BEGIN
478         DROP TABLE db_status;
479         PRINT 'Tabela wersjonowania zostala skasowana';
480     END
481     ELSE
482     BEGIN
483         RAISERROR ( 'Nie mozna usunac tabeli wersjonowania poniewaz ma niezerowa wersje', 11,
2);
484     END
485 END
486
487
488
489 END TRY
490 BEGIN CATCH
491
492     SELECT
493         ERROR_NUMBER() AS ErrorNumber ,
494         ERROR_SEVERITY() AS ErrorSeverity ,
495         ERROR_STATE() AS ErrorState ,
496         ERROR_PROCEDURE() AS ErrorProcedure ,
497         ERROR_LINE() AS ErrorLine ,
498         ERROR_MESSAGE() AS ErrorMessage ;
499
500 END CATCH;

```

Listing 7: skrypt_usuwajacy_obiekty_z_bazy.sql.sql

Listings

1	Blok CATCH w skrypcie tworzącym	3
2	Wynik uruchomienia całego skryptu tworzącego w trybie wsadowym	4
3	Skrypt tworzący tabelę obiekty	5
4	Skrypt tworzący tabelę obiekty	6
5	Wynik uruchomienia całego skryptu usuwającego w trybie wsadowym	6
6	skrypt_tworzacy_obiekty_w_bazie_danych.sql	8
7	skrypt_usuwajacy_obiekty_z_bazy.sql.sql	16

Spis rysunków

1	Diagram tabel wygenerowanej bazy danych	3
2	Tabela obiekty wyświetlona w programie DataGrip	5
3	Tabela najmy wyświetlona w programie DataGrip	6