

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

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](#)).

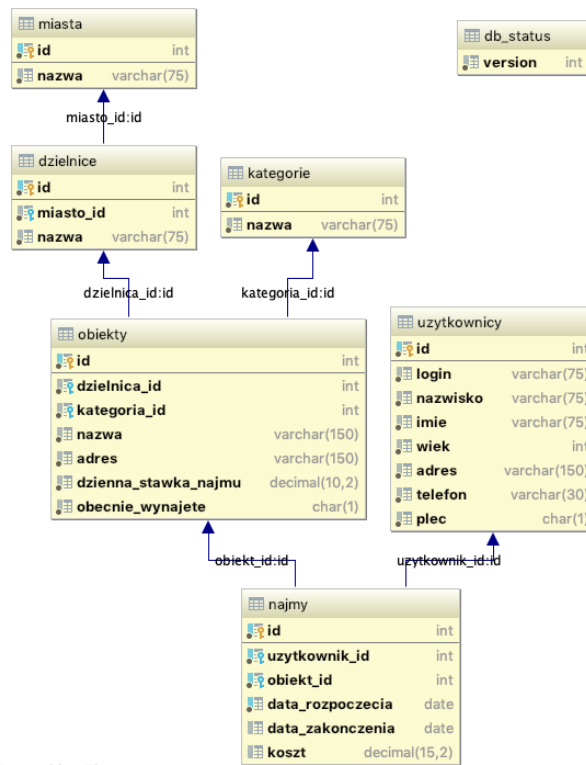
Jako aplikację służącą do łączenia się i wykonywania poleceń wykorzystane zostały aplikacje:

- 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



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`.

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.

```

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

- 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".

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 ;np. ILE DANYCH DO KAZDEJ 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 3, 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 użytkownikami'
2
3 (1 rows affected)
4 Wersja 13: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji 12
5 Wersja 12: 'Utworzenie relacji pomiędzy obiektami a najmami'
6
7 (1 rows affected)
8 Wersja 12: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji 11
9 Wersja 11: 'Utworzenie relacji pomiędzy użytkownikami a najmami'
10
11 (1 rows affected)
12 Wersja 11: Migracja została odinstalowana pomyślnie – 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 została odinstalowana pomyślnie – teraz baza jest w wersji 9
17 Wersja 9: 'Utworzenie tabeli z najmami'
18
19 (1 rows affected)
20 Wersja 9: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji 8
21 Wersja 8: 'Utworzenie tabeli z użytkownikami'
22
23 (1 rows affected)
24 Wersja 8: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji 7
25 Wersja 7: 'Utworzenie relacji pomiędzy kategoriami a obiektami'
26
27 (1 rows affected)
28 Wersja 7: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji 6
29 Wersja 6: 'Utworzenie relacji pomiędzy dzielnicami a obiektami'
30
31 (1 rows affected)
32 Wersja 6: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji 5
33 Wersja 5: 'Utworzenie tabeli z obiektami'
34
35 (1 rows affected)
36 Wersja 5: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji 4
37 Wersja 4: 'Utworzenie tabeli z kategoriami'
38
39 (1 rows affected)
40 Wersja 4: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji 3
41 Wersja 3: 'Utworzenie relacji pomiędzy miastami a dzielnicami'
42
43 (1 rows affected)
44 Wersja 3: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji 2
45 Wersja 2: 'Utworzenie tabeli z dzielnicami'
46
47 (1 rows affected)
48 Wersja 2: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji 1
49 Wersja 1: 'Utworzenie tabeli z miastami'
50
51 (1 rows affected)
52 Wersja 1: Migracja została odinstalowana pomyślnie – teraz baza jest w wersji 0
53 Tabela wersjonowania została skasowana

```

Listing 3: 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

²footnotes working fine

³footnotes working fine

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;
41         PRINT 'Wersja 1: Migracja zostala zainstalowana pomyslnie – teraz baza jest w wersji
42         1';
43     END
44 ELSE
45     BEGIN
46         IF EXISTS(SELECT * FROM db_status WHERE version < 0)
47         BEGIN
48             RAISERROR ('Wersja 1: Baza danych jest w za niskiej wersji (wymagana jest wersja
49             0) aby zainstalowac migracje', 11, 2);
50         END
51         ELSE
52         BEGIN
53             PRINT 'Wersja 1: Migracja ju  zostala zainstalowana wczesniej';
54         END
55     END
56 END
57 ELSE
58 BEGIN
59     RAISERROR ('Wersja 1: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
60 END
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 migracje', 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
3';
121             END
122             ELSE
123                 BEGIN
124                     IF EXISTS(SELECT * FROM db_status WHERE version < 2)
125                         BEGIN
126                             RAISERROR ('Wersja 3: Baza danych jest w za niskiej wersji (wymagana jest wersja
2) aby zainstalować migracje', 11, 2);
127                         END
128                         ELSE
129                             BEGIN
130                                 PRINT 'Wersja 3: Migracja już została zainstalowana wcześniej';
131                             END
132                         END
133                 END
134             ELSE
135                 BEGIN
136                     RAISERROR ('Wersja 3: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
137                 END
138
139
140
141
142

```



```

143
144
145
146 PRINT 'Wersja 4: ''Utworzenie tabeli z kategoriami'''
147 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
148 BEGIN
149     IF EXISTS(SELECT * FROM db_status WHERE version = 3)
150     BEGIN
151         EXEC dbo.sp_executesql @statement = N'
152             CREATE TABLE kategorie (
153                 id INT PRIMARY KEY NOT NULL IDENTITY (1, 1),
154                 nazwa VARCHAR(75) NOT NULL
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 ELSE
175 BEGIN
176     RAISERROR ('Wersja 4: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
177 END
178
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 obiekty (
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
206         UPDATE db_status SET version = 5 WHERE version = 4;
207         PRINT 'Wersja 5: Migracja została zainstalowana pomyślnie – teraz baza jest w wersji
208         5';
209     END
210 ELSE
211 BEGIN
212     IF EXISTS(SELECT * FROM db_status WHERE version < 4)
213     BEGIN
214         RAISERROR ('Wersja 5: Baza danych jest w za niskiej wersji (wymagana jest wersja
215         4) aby zainstalować migrację', 11, 2);

```

```

213         END
214     ELSE
215         BEGIN
216             PRINT 'Wersja 5: Migracja ju została zainstalowana wcześniej';
217         END
218     END
219 END
220 ELSE
221 BEGIN
222     RAISERROR ('Wersja 5: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
223 END
224
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 obiekty
239             ADD CONSTRAINT obiekty_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     ELSE
256     BEGIN
257         PRINT 'Wersja 6: Migracja ju została zainstalowana wcześniej';
258     END
259 END
260 END
261 ELSE
262 BEGIN
263     RAISERROR ('Wersja 6: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
264 END
265
266
267
268
269
270
271 PRINT 'Wersja 7: ''Utworzenie relacji pomiędzy 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 obiekty
278             ADD CONSTRAINT obiekty_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';

```

```

285     END
286 ELSE
287     BEGIN
288         IF EXISTS(SELECT * FROM db_status WHERE version < 6)
289             BEGIN
290                 RAISERROR ('Wersja 7: Baza danych jest w za niskiej wersji (wymagana jest wersja
291                 6) aby zainstalowac migracje', 11, 2);
292             END
293         ELSE
294             BEGIN
295                 PRINT 'Wersja 7: Migracja ju  została zainstalowana wcześniej';
296             END
297     END
298 ELSE
299     BEGIN
300         RAISERROR ('Wersja 7: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
301     END
302
303
304
305
306
307
308
309 PRINT 'Wersja 8: ''Utworzenie tabeli z uzytkownikami''
310 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
311 BEGIN
312     IF EXISTS(SELECT * FROM db_status WHERE version = 7)
313         BEGIN
314             EXEC dbo.sp_executesql @statement = N'
315                 CREATE TABLE uzytkownicy (
316                     id INT PRIMARY KEY NOT NULL IDENTITY (1, 1),
317
318                     login VARCHAR(75) NOT NULL,
319
320                     nazwisko VARCHAR(75) NOT NULL,
321                     imie VARCHAR(75) NOT NULL,
322                     wiek INT NOT NULL CHECK (wiek BETWEEN 1 AND 100),
323                     adres VARCHAR(150) NOT NULL DEFAULT ''Brak Danych'',
324                     telefon VARCHAR(30) NOT NULL DEFAULT ''Brak Danych'',
325                     plec CHAR(1) NOT NULL CHECK (plec IN (''K'', ''M'')),
326                 );
327
328
329
330                 UPDATE db_status SET version = 8 WHERE version = 7;
331                 PRINT 'Wersja 8: Migracja została zainstalowana pomyslnie – teraz baza jest w wersji
332                 8';
333             END
334         ELSE
335             BEGIN
336                 IF EXISTS(SELECT * FROM db_status WHERE version < 7)
337                     BEGIN
338                         RAISERROR ('Wersja 8: Baza danych jest w za niskiej wersji (wymagana jest wersja
339                         7) aby zainstalowac migracje', 11, 2);
340                     END
341                 ELSE
342                     BEGIN
343                         PRINT 'Wersja 8: Migracja ju  została zainstalowana wcześniej';
344                     END
345             END
346         END
347     END
348 ELSE
349     BEGIN
350         RAISERROR ('Wersja 8: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
351     END
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 poczeczia DATE NOT NULL DEFAULT getdate(),
370                 data_zakonczenia DATE NULL,
371                 koszt DECIMAL(15, 2) NULL,
372             );
373
374             UPDATE db_status SET version = 9 WHERE version = 8;
375             PRINT 'Wersja 9: Migracja zostala zainstalowana pomyslnie – teraz baza jest w wersji
376             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
383             8) aby zainstalowac migracje', 11, 2);
384             END
385         ELSE
386         BEGIN
387             PRINT 'Wersja 9: Migracja ju  zostala zainstalowana wczesniej';
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 poczeczia);
410
411             UPDATE db_status SET version = 10 WHERE version = 9;
412             PRINT 'Wersja 10: Migracja zostala zainstalowana pomyslnie – teraz baza jest w wersji
413             10';
414         END
415     ELSE
416     BEGIN
417         IF EXISTS(SELECT * FROM db_status WHERE version < 9)
418         BEGIN
419             RAISERROR ('Wersja 10: Baza danych jest w za niskiej wersji (wymagana jest wersja
420             9) aby zainstalowac migracje', 11, 2);
421             END
422         ELSE
423         BEGIN
424             PRINT 'Wersja 10: Migracja ju  zostala zainstalowana wczesniej';
425             END
426         END
427     ELSE

```

```

428 BEGIN
429     RAISERROR ( 'Wersja 10: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
430 END
431
432
433
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 wczesniej';
465     END
466 END
467 ELSE
468 BEGIN
469     RAISERROR ( 'Wersja 11: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
470 END
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 (obiekt_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
492         12';
493     END
494 ELSE
495 BEGIN
496     IF EXISTS(SELECT * FROM db_status WHERE version < 11)
497     BEGIN
498         RAISERROR ( 'Wersja 12: Baza danych jest w za niskiej wersji (wymagana jest wersja
499         11) aby zainstalowac migracje', 11, 2);
500     END

```

```

499         ELSE
500             BEGIN
501                 PRINT 'Wersja 12: Migracja ju została zainstalowana wcześniej';
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 została zainstalowana pomyslnie – teraz baza jest w wersji
529                 13';
530             END
531         ELSE
532             BEGIN
533                 IF EXISTS(SELECT * FROM db_status WHERE version < 12)
534                     BEGIN
535                         RAISERROR ('Wersja 13: Baza danych jest w za niskiej wersji (wymagana jest wersja
536                         12) aby zainstalowac migracje', 11, 2);
537                     END
538                 ELSE
539                     BEGIN
540                         PRINT 'Wersja 13: Migracja ju została zainstalowana wcześniej';
541                     END
542             END
543         END
544     ELSE
545     BEGIN
546         RAISERROR ('Wersja 13: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
547     END
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 4: 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
41 PRINT 'Wersja 12: ''Utworzenie relacji pomiedzy obiektami a najmami''
42 IF EXISTS(SELECT * FROM sys.tables WHERE name = N'db_status')
43 BEGIN
44     IF EXISTS(SELECT * FROM db_status WHERE version = 12)
45     BEGIN
46         EXEC dbo.sp_executesql @statement = N'
47             ALTER TABLE najmy DROP CONSTRAINT najmy_obiekty_fk;
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

```

```

70
71
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         UPDATE db_status SET version = 10 WHERE version = 11;
87         PRINT 'Wersja 11: Migracja zostala odinstalowana pomyslnie – teraz baza jest w wersji
10';
88     END
89 ELSE
90     BEGIN
91         IF EXISTS(SELECT * FROM db_status WHERE version > 11)
92         BEGIN
93             RAISERROR ('Wersja 11: Baza danych jest w za wysokiej wersji (wymagana jest
wersja 11) aby odinstalowac migracje', 11, 2);
94         END
95     ELSE
96     BEGIN
97         PRINT 'Wersja 11: Migracja nie by a wzczesniej zainstalowana lub zostala juz
odinstalowana';
98     END
99 END
100 END
101 ELSE
102 BEGIN
103     RAISERROR ('Wersja 11: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
104 END
105
106
107
108
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 zostala odinstalowana pomyslnie – teraz baza jest w wersji
9';
124     END
125 ELSE
126     BEGIN
127         IF EXISTS(SELECT * FROM db_status WHERE version > 10)
128         BEGIN
129             RAISERROR ('Wersja 10: Baza danych jest w za wysokiej wersji (wymagana jest
wersja 10) aby odinstalowac migracje', 11, 2);
130         END
131     ELSE
132     BEGIN
133         PRINT 'Wersja 10: Migracja nie by a wzczesniej zainstalowana lub zostala juz
odinstalowana';
134     END
135 END
136 END
137 ELSE
138 BEGIN

```



```

139 RAISERROR ('Wersja 10: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
140 END
141
142
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 migracje', 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
179
180
181
182
183
184
185 PRINT 'Wersja 8: ''Utworzenie tabeli z użytkownikami'''
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 została odinstalowana pomyślnie – 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 odinstalować migracje', 11, 2);
204     END
205 ELSE
206 BEGIN
207     PRINT 'Wersja 8: Migracja nie była wcześniej zainstalowana lub została już
208     odinstalowana';
209 END
210 END

```

```

208 END
209 ELSE
210 BEGIN
211     RAISERROR ('Wersja 8: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
212 END
213
214
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 objekty DROP CONSTRAINT objekty_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
249     RAISERROR ('Wersja 7: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
250 END
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 objekty DROP CONSTRAINT objekty_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

```

```

278         odinstalowana';
279         END
280     END
281 ELSE
282 BEGIN
283     RAISERROR ('Wersja 6: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
284 END
285
286
287
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 została odinstalowana pomyślnie – 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 odinstalować migracje', 11, 2);
312     END
313 ELSE
314 BEGIN
315     PRINT 'Wersja 5: Migracja nie była wcześniej zainstalowana lub została już
316         odinstalowana';
317     END
318 END
319 ELSE
320 BEGIN
321     RAISERROR ('Wersja 5: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
322 END
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 pomyślnie – 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 odinstalować migracje', 11, 2);
348     END
349 END

```

```

347         ELSE
348             BEGIN
349                 PRINT 'Wersja 4: Migracja nie by a wzczesniej zainstalowana lub zostala juz
odinstalowana';
350             END
351         END
352     END
353 ELSE
354     BEGIN
355         RAISERROR ('Wersja 4: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
356     END
357
358
359
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 zostala odinstalowana pomyslnie – teraz baza jest w wersji
2';
376             END
377         ELSE
378             BEGIN
379                 IF EXISTS(SELECT * FROM db_status WHERE version > 3)
380                     BEGIN
381                         RAISERROR ('Wersja 3: Baza danych jest w za wysokiej wersji (wymagana jest wersja
3) aby odinstalowac migracje', 11, 2);
382                     END
383                 ELSE
384                     BEGIN
385                         PRINT 'Wersja 3: Migracja nie by a wzczesniej zainstalowana lub zostala juz
odinstalowana';
386                     END
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 zostala odinstalowana pomyslnie – teraz baza jest w wersji
1';
412             END
413         ELSE
414             BEGIN
415                 IF EXISTS(SELECT * FROM db_status WHERE version > 2)
416                     BEGIN

```

```

417         RAISERROR ('Wersja 2: Baza danych jest w za wysokiej wersji (wymagana jest wersja
418         2) aby odinstalowac migracje', 11, 2);
419     END
420 ELSE
421     BEGIN
422         PRINT 'Wersja 2: Migracja nie by a wzczesniej zainstalowana lub zostala juz
423         odinstalowana';
424     END
425 END
426 ELSE
427 BEGIN
428     RAISERROR ('Wersja 2: Nie znaleziono tabeli wersjonowania bazy danych', 11, 1);
429 END
430
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 zostala odinstalowana pomyslnie – 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 odinstalowac migracje', 11, 2);
456         END
457     ELSE
458     BEGIN
459         PRINT 'Wersja 1: Migracja nie by a wzczesniej zainstalowana lub zostala juz
460         odinstalowana';
461     END
462 END
463 END
464
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,
484         2);
485     END
486 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 5: 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	Wynik uruchomienia całego skryptu usuwającego w trybie wsadowym	5
4	skrypt_tworzacy_obiekty_w_bazie_danych.sql	7
5	skrypt_usuwajacy_obiekty_z_bazy.sql.sql	15

Spis rysunków

1	Diagram tabel wygenerowanej bazy danych	3
---	---	---