

SYSTEMY BAZ DANYCH

SQL



TEMAT 23-01: Tworzenie bazy danych i tabel.

Autor dokumentu: Wojciech Galiński

wtorek, 8 listopada 2016 r.

351203 Technik informatyk

ŹRÓDŁA WIEDZY: <http://dev.mysql.com/doc/>.

Zagadnienia obowiązkowe

1. Tworzenie i usuwanie bazy danych – służą do tego polecenia „CREATE DATABASE” oraz „DROP DATABASE” w postaci:

CREATE DATABASE [IF NOT EXISTS] baza_danych

DEFAULT CHARACTER SET kodowanie COLLATE sposob_porzadkowania_danych;

DROP DATABASE [IF EXISTS] baza_danych;

PRZYKŁADY:

-- Baza danych z domyślnym kodowaniem UTF-8 i sortowaniem zgodnym z polskim alfabetem

DROP DATABASE IF EXISTS dziennik; -- usunięcie bazy danych (jeśli istnieje)

CREATE DATABASE dziennik DEFAULT CHARACTER SET utf8 COLLATE utf8_polish_ci;

PAMIĘTAJ! Tworząc nową bazę danych, a także tabele, staraj się, żeby system nie używał wartości domyślnych (może to prowadzić do różnych zachowań w różnych SZBD).

Aby móc wygodnie zarządzać tabelami (tworzenie, edycja, usuwanie) w utworzonej bazie danych, należy ustawić ją jako domyślną bazę danych za pomocą polecenia:

USE baza_danych;

2. Typy danych w kolumnach – zanim utworzymy nową tabelę, dowiedzmy się, jakie typy danych można w nich przechowywać (każda kolumna zawiera dane jednej grupy typów).

Ważniejsze typy danych w SQL to (n – liczba całkowita):

Typ	Rozmiar w bajtach	Wartości bez znaku		Wartości ze znakiem	
		min	max	min	max
TINYINT	1	0	255	-128	127
SMALLINT	2	0	65535	-32768	32767
MEDIUMINT	3	0	16777215	-8388608	8388607
INT	4	0	4294967295	-2147483648	2147483647
BIGINT	8	0	18.4467E18	-9.2233E18	9.2233E18

Typ	Rozmiar w bajtach	Wartość minimalna	Wartość maksymalna
DECIMAL(s,p)	?	-10 ³⁸	10 ³⁸⁻¹
NUMERIC(s,p)			
FLOAT	4	-1,79E+38	1,79E+38
DOUBLE, REAL	8	-3,4E+308	3,4E+308

Typy zmiennoprzecinkowe

Typy całkowitoliczbowe

Typ	Rozmiar w bajtach
CHAR(n), BINARY(n)	n dla n ∈ {0, 2 ⁸ -1}
VARCHAR(n), VARBINARY(n)	n+1 dla n ∈ {0, 2 ⁸ -1}, n+2 dla n ∈ {2 ⁸ , 2 ¹⁶ -1}
TINYTEXT, TINYBLOB	n+1 dla n ∈ {0, 2 ⁸ -1}
TEXT, BLOB	n+2 dla n ∈ {0, 2 ¹⁶ -1}
MEDIUMTEXT, MEDIUMBLOB	n+3 dla n ∈ {0, 2 ²⁴ -1}
LONGTEXT, LONGBLOB	n+4 dla n ∈ {0, 2 ³² -1}

Typ	Rozmiar w bajtach	Zakres wartości
TIMESTAMP	4	1900
DATETIME	8	1000-01-01 00:00:00 – 9999-12-31 23:59:59
DATE	3	1000-01-01 - 9999-12-31
TIME	3	00:00:00 – 23:59:59
YEAR	1	1900-2155 YEAR(4), 1970-2069 YEAR(2)

Typy daty i czasu

Typy tekstowe

Oprócz tego występuje specjalny typ **SERIAL**, który jest prostym sposobem zdefiniowania klucza głównego.

SERIAL = BIGINT UNSIGNED NOT NULL AUTO_INCREMENT UNIQUE

3. Typ wyliczeniowy („ENUM”) – umożliwia przechowywanie indeksu do listy tekstów (1B – maksymalnie do 255 elementów, 2B – maksymalnie do 65535 elementów). Indeks 0 wskazuje na pusty tekst. Elementy tego typu możemy wywoływać:

- **przez wartość** (tekst w apostrofach lub cudzysłowie);
- **przez indeks** (liczba całkowita bez apostrofów lub cudzysłowu).

ZAPAMIĘTAJ! Wielkość liter w definiowanych tekstach MA znaczenie.

PRZYKŁADY:

CREATE TABLE ocena (wartosc ENUM('2+', '3+', '4+', '5+')); -- def. typu w def. Tabeli

-- Odwoływanie się do elementów listy na przykładzie wstawiania rekordów

-- (zwróć uwagę przede wszystkim na pierwszy, drugi i ostatni wśród dodawanych elementów)

INSERT INTO ocena VALUES (0), ('2'), (2), ('4'), (6), (7);

4. **Typ zbiorowy („SET”)** – umożliwia przechowywanie podzbiorów zbioru zdefiniowanego jako typ „SET”. Można zdefiniować maksymalnie 64 elementy (każdy element zajmuje 1 bit).

Łamie zasady normalizacji relacyjnych baz danych. Używana, aby zwiększyć wydajność SZBD.

PRZYKŁADY:

```
CREATE TABLE osoba (jezyki SET('pl','de','fr','gb','usa')); -- def. typu w def. Tabeli
INSERT INTO osoba VALUES (0), ('1'), (2), ('pl,de'), (7), ('gb'), (31), (63);
```

5. **Silniki w MySQL** – wyróżniamy m. in. (źródło: blog.zeromski.com.pl):

- **MyISAM** – główną zaletą jest stosowanie indeksów „FULLTEXT”. Pozwala to na bardzo szybkie przeszukiwanie danych za pomocą „MATCH ... AGAINST” (zamiast „LIKE”);
- **InnoDB** – nie można tworzyć indeksu „FULLTEXT”, natomiast można definiować klucze obce. Tylko w „InnoDB” można wykonywać transakcję;
- **Memory** – dane są przechowywane w pamięci komputera i po wyłączeniu serwera znikają z bazy. Dodatkowo nie można na tym silniku tworzyć pól typu TEXT, BLOB itp.;
- **Archive** – służy do archiwizacji (składowanie i wyciąganie danych), nie można tworzyć kluczy.

6. **Tworzenie nowej tabeli** – zadanie to realizujemy za pomocą polecenia „CREATE TABLE”. Słowo kluczowe „TEMPORARY” oznacza tabelę tymczasową kasowaną automatycznie po zerwaniu połączenia z bazą danych (zasłoni zwykłą tabelę o tej samej nazwie). Oto wzorzec:

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tabela
(
    -- Definicje kolumn tabeli
)
[ENGINE=typ_tabeli] [DEFAULT CHARSET=kodowanie]
[COLLATE=rodzaj_sortowania];
```

gdzie:

- **typ tabeli** – będziemy używać:
 - „InnoDB” – obsługuje transakcje, umożliwia stosowanie kluczy obcych;
 - „MyISAM” – wysoka wydajność dla operacji „SELECT”, łatwa archiwizacja danych.
 - **kodowanie znaków** – będziemy używać „utf8” (uwzględnia on wszystkie języki świata);
 - **rodzaj sortowania** – będziemy używać:
 - „utf8_polish_ci”, „utf8_polish_cs” – uwzględnia polski alfabet: AĄBCĆDEEFGHIJKL...;
 - „utf8_general_ci”, „utf8_general_cs” – nie uwzględnia polskiego alfabetu (polskie znaki są po „Z”).
- W powyższych pozycjach występują przyrostki „ci” oraz „cs”. Oto ich znaczenie:
- „ci” – (ang. case insensitive) bez uwzględniania wielkości liter;
 - „cs” – (ang. case sensitive) uwzględnianie wielkości liter.

PRZYKŁAD:

```
CREATE TABLE IF NOT EXISTS klasa
( id INT PRIMARY KEY,
  nazwa VARCHAR(5) COLLATE utf8_polish_ci UNIQUE NOT NULL,
  opis VARCHAR(5) COLLATE DEFAULT NULL, -- jakim sposobem jest sortowana ta kolumna?
  data_wpisu TIMESTAMP DEFAULT NOW()
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_polish_ci;
```

7. **Usuwanie tabeli** – służy do tego polecenie „DROP TABLE” w postaci:

```
DROP TABLE [IF EXISTS] tabela;
```

PRZYKŁAD: DROP TABLE IF EXISTS towar; -- usunięcie istniejącej bazy danych

8. **Wyświetlanie polecenia tworzenia tabeli** – po utworzeniu tabeli możemy chcieć przypomnieć sobie, jak tworzyliśmy tabelę. Z pomocą przychodzi polecenie „SHOW”. Oto wzorzec:

```
SHOW CREATE DATABASE baza_danych;
SHOW CREATE TABLE tabela;
```

PRZYKŁAD: SHOW CREATE TABLE klasa;

Pytania kontrolne

1. Jak tworzy się i usuwa bazę danych? Na co należy zwrócić szczególną uwagę?
2. Wymień i opisz typy liczbowe z podziałem na dwie podstawowe kategorie.
3. Wymień i opisz typy tekstowe oraz typy daty i czasu,
4. Wymień i opisz rodzaje silników w MySQL. Z których silników będziemy korzystać najczęściej i dlaczego?
5. Podaj różnice pomiędzy typem wyliczeniowym, a zbiorowym.
6. Jak utworzyć tabelę za pomocą języka SQL? Na co należy zwrócić szczególną uwagę?

Zadania

1. Utwórz bazę danych 'dziennik' i tabele umożliwiające przechowywanie końcowych ocen ucznia i wyliczenie średniej ocen.