

Operacje zarządzania tabelami

Model relacyjny zakłada, że dane w bazie zawarte są w tabelach. Oprócz czysto użytkowego zagadnienia obsługi danych zawartych w tabeli (dopisywanie, modyfikacja oraz usuwanie) pojawia się pytanie: co określa postać danych, zawartych w tabeli? Tym czynnikiem jest struktura tabeli, definiowana typami poszczególnych kolumn i ich właściwościami. Język obsługi baz danych powinien umożliwiać wykonanie analogicznych operacji na projekcie tabeli, jak na danych: stworzenia tabeli o założonej strukturze, modyfikację struktury tabeli oraz usuwanie tabeli z bazy danych.

Operacje dotyczące projektu tabeli są silnie związane z konkretnym *RDBMS* – producenci baz danych oferują różne rozszerzenia i ułatwienia, związane z założeniami projektowymi ich produktów. Mimo tych różnic, podstawowym wariantem instrukcji tworzącej tabelę jest

CREATE TABLE nazwa_tabeli (opis_kolumny1, opis_kolumny2, ...)

przy czym opis kolumny to konstrukcja, zawierająca co najmniej parę: *nazwa kolumny typ danych* w opisie kolumny. Pula typów danych, jak również ich modyfikatorów, związana jest z konkretnym *RDBMS*. Przeważnie wyróżnia się trzy podstawowe grupy typów danych w bazach:

1. dane znakowe
2. dane numeryczne
3. dane związane z zapamiętywaniem i obsługą daty oraz czasu

Przykłady typów danych obsługiwanych przez bazę MySQL

Podstawowymi typami znakowymi są typy *CHAR* oraz *VARCHAR*. Dla typów tych wymagane jest podanie maksymalnej długości przechowywanego w kolumnie tekstu. Podane typy różni od siebie to, że wartość typu *CHAR* zajmuje w tabeli zawsze zadeklarowaną, maksymalną ilość znaków, zaś wartość *VARCHAR* zajmuje tylko tyle miejsca, ile jest konieczne, aby zapamiętać wpisany tekst. Maksymalną długość tekstu podaje się w nawiasach okrągłych, stojących po nazwie typu. W bazie *MySQL* maksymalny rozmiar dla typów *CHAR* i *VARCHAR* wynosi 255 znaków. Przykładowa deklaracja kolumny typu tekstowego:

Imię **VARCHAR(50)**

Dla większych typów znakowych konieczne jest użycie specjalizowanych typów z rodzin *TEXT* lub *BLOB* (ang. *Binary Large Object* – duży obiekt binarny). Interesującymi typami są również: typ wyliczeniowy *ENUM*, reprezentujący zbiór cech, z których można posiadać tylko jedną, oraz typ zbiorowy *SET*, reprezentujący zbiór cech, z których każdą z osobna można posiadać albo nie. Oba te typy charakteryzuje to, że prezentują dane w postaci tekstów, podanych w deklaracji, ale w tabeli prowadzonej przez bazę przechowywane są jako liczby całkowite. Przykłady:

nieobecność *ENUM('usprawiedliwiona', 'nieusprawiedliwiona', 'ucieczka')*

typ *SET('starosta_grupy', 'starosta_roku')*

Typy numeryczne, dostępne w bazie *MySQL*, to między innymi: *INTEGER* (lub krócej *INT*), służący do zapisu liczb całkowitych, *FLOAT* służący do zapisu liczb zmiennoprzecinkowych o tzw. pojedynczej precyzji, oraz *DOUBLE PRECISION* (skrótowo *DOUBLE*), służący do obsługi kolumn, zawierających liczby zmiennoprzecinkowe o podwójnej precyzji. Istnieją ponadto typy *NUMERIC* oraz *DECIMAL*, które służą również do przechowywania liczb zmiennoprzecinkowych, jednakże w tabeli są one przechowywane w postaci tekstowej, a nie binarnej.

Typy numeryczne posiadają podobne modyfikatory, jak typy znakowe. Charakterystyczne dla modyfikatorów typów numerycznych jest to, że wpływają jedynie na sposób wyświetlania danych numerycznych, a nie na sposób ich przechowywania w bazie (np. ilość zajmowanej pamięci itp.). Dla typu całkowitego jedyny modyfikator określa, ile znaków ma zająć wyświetlona liczba, dla typów zmiennoprzecinkowych pierwszy parametr jest analogiczny, jak dla typu całkowitego, a drugi

parametr, oddzielony przecinkiem, określa ilość cyfr po separatorze dziesiętnym, jaką należy wyświetlać. W bazie *MySQL* limit ilości znaków wyświetlanych liczb (suma ilości cyfr, znaków, separatora dziesiętnego) wynosi 255, zaś maksymalne ograniczenie na ilość cyfr, wyświetlanych po separatorze dziesiętnym wynosi 30, nie może ono jednak przekroczyć ilości wszystkich wyświetlanych znaków liczby, zmniejszonej o 2. Przykłady deklaracji kolumn typów numerycznych:

wiek INT(3) wiek człowieka w latach jest liczbą najwyżej trzycyfrową
ocena NUMERIC(3,1) ocena studenta to dwie cyfry plus kropka dziesiętna

Typy danych, służące do przechowywania informacji o datach i czasie to: *DATE* – przechowujący datę, *TIME* – przechowujący czas, *DATETIME* – mogący przechować jednocześnie informację o dacie i czasie oraz *YEAR*, pozwalający zapisać rok. Dodatkowo dostępny jest typ *TIMESTAMP*, służący do automatycznego zapamiętywania daty i/lub czasu ostatniej modyfikacji wiersza w tabeli. Podczas modyfikacji danych w tabeli w danym wierszu, pole aktualizowane jest automatycznie na podstawie aktualnej daty i czasu. W zależności od podanego parametru kolumna typu *TIMESTAMP* przechowuje datę, czas lub jedno i drugie, podając 14 otrzymuje się pełny zapis: rok (forma czterocyfrowa), miesiąc, dzień, godzinę, minutę i sekundę, podając mniejszą wartość modyfikatora uzyskuje się mniejszą precyzję zapisu czasu modyfikacji danych (patrz dokumentacja: “MySQL Reference Manual”).

Modyfikatory kolumn

Oprócz obowiązkowego podawania nazw i typów danych dla kolumn podczas tworzenia tabeli, systemy baz danych umożliwiają narzucenie pewnych ograniczeń na dane, które będą do tych tabel wprowadzane, jak również wprowadzenie pewnych ułatwień. Możliwości te są ściśle związane z konkretnym systemem zarządzania bazą danych, jakkolwiek dwa poniższe są implementowane praktycznie wszędzie:

1. modyfikatory: *NULL* lub *NOT NULL* umożliwiają zadeklarowanie, czy dopuszczalne (*NULL*) lub niedopuszczalne (*NOT NULL*) jest pominięcie nadania wartości w kolumnie. Pominięcie tego modyfikatora zazwyczaj jest jednoznaczne z podaniem *NULL*.
2. Modyfikator *DEFAULT wartość* pozwala na podanie wartości domyślnej, która musi być stałą (wyrażeniem nie zawierającym funkcji i nazw kolumn), jaka zostanie przypisana w kolumnie w przypadku, gdy nie zostanie podana wartość w instrukcji, wstawiającej wiersz do tabeli.

W bazie danych *MySQL* można użyć dla jednej z kolumn tabeli, która musi być typu *INT*, modyfikatora *AUTO_INCREMENT*, powodującego automatyczne nadawanie kolejnych liczb całkowitych w tej kolumnie poszczególnym wierszom, gdy kolumnie nie nadawana jest jawnie wartość podczas wstawiania wiersza. Jest to dogodne do tworzenia numerycznych kluczy głównych w tabelach.

Przykład instrukcji tworzenia tabeli w bazie *MySQL*:

```
CREATE TABLE personalia  
(  
id INT NOT NULL AUTO_INCREMENT,  
imie VARCHAR(50),  
nazwisko VARCHAR(50) NOT NULL,  
PRIMARY KEY(id)  
)
```

W powyższej instrukcji pojawia się sekcja *PRIMARY KEY(id)*. W ten sposób informuje się bazę *MySQL*, że w tabeli *personalia* klucz główny składa się z kolumny *id* (kolumn może być

więcej, rozdziela się je przecinkami). Pozwala to bazie na odrzucanie instrukcji wprowadzania danych naruszających regułę unikalności klucza głównego, co pozwala na wyeliminowanie części błędów i zredukowanie możliwości zniszczenia danych w tabeli. Od kolumn klucza głównego wymaga się, aby były *NOT NULL* tj. uniemożliwiały pominięcie nadania im wartości w instrukcji wstawiania danych.

Po utworzeniu tabeli czasem okazuje się, że należy ją zmodyfikować z racji błędu podczas tworzenia lub zmiany założeń projektowych. Do tego celu służy instrukcja

ALTER TABLE nazwa tablicy specyfikacja zmiany

Jest to kolejna instrukcja silnie zależna od konkretnego RDBMS, w różnych bazach możliwe jest dokonywanie zmian z szerszego lub węższego repertuaru możliwości, udostępnianych przez bazę, ponadto różna bywa składnia specyfikacji zmiany. Dwie najczęściej oferowane możliwości zmian:

1. **ADD** specyfikacja kolumny – dodanie nowej kolumny do tabeli
2. **MODIFY** specyfikacja kolumny – modyfikacja istniejącej kolumny w tabeli

W bazie danych *MySQL* można dokonać usunięcia (*DROP nazwa kolumny*) kolumny z tabeli – nie wszystkie *RDBMS* obsługują taką możliwość. Istnieje również kilka innych możliwości modyfikacji kolumny (np. zmiana nazwy kolumny: *CHANGE stara_nazwa specyfikacja kolumny*), a nawet zmiana nazwy tabeli (*RENAME nowa_nazwa*).

Przykłady:

ALTER TABLE personalia ADD wiek INT(3)	dodanie kolumny <i>wiek</i> do tabeli
ALTER TABLE personalia MODIFY imie VARCHAR(100) NOT NULL	zmiana specyfikacji kolumny <i>imie</i>
ALTER TABLE personalia DROP nazwisko	usunięcie z tabeli kolumny <i>nazwisko</i>

W bazie *MySQL* w jednej instrukcji *ALTER TABLE* można podać kilka specyfikacji zmian kolumn, oddzielonych przecinkami

Stosunkowo najprostszą instrukcją, dotyczącą tabeli jest usunięcie jej z bazy. Dokonuje tego instrukcja:

DROP TABLE nazwa usuwanej tabeli

Należy pamiętać, że zazwyczaj jest to operacja nieodwracalna i powoduje usunięcie z bazy danych tabeli wraz z jej zawartością.