

Operacje na danych zawartych w tabeli

Podstawowym celem istnienia bazy danych jest przechowywanie informacji. Z tego względu wydaje się oczywiste, że konieczne jest istnienie w języku *SQL* instrukcji, których zadaniem jest wprowadzanie danych do bazy, modyfikacja uprzednio wprowadzonych danych oraz usuwanie niepotrzebnych danych.

Dane zawarte są w bazie w postaci wierszy o ustalonej strukturze, z których składają się tabele. Instrukcje operujące na danych zawsze odnoszą się do jednej tabeli. Przy obsłudze danych w tabeli może pojawić się również zagadnienie predykatu: warunku ograniczającego działanie instrukcji języka *SQL* do pewnego podzbioru wierszy w tabeli.

Instrukcją języka *SQL*, której zadaniem jest dopisywanie danych do tabeli, jest instrukcja *INSERT* (ang. wstaw). Występuje ona w trzech wariantach:

- dwa z nich pozwalają na wprowadzanie danych do tabeli, korzystając z wartości podawanych przez użytkownika – te dwa warianty różnią się składnią
- trzeci wariant pozwala na wprowadzenie do tabeli wyniku zapytania, odnoszącego się do innych tabel

W najprostszym przypadku instrukcja *INSERT* przedstawia się następująco:

INSERT INTO tabela VALUES (wartość_w_kolumnie1, wartość_w_kolumnie2,...)

Dodając w ten sposób wiersz do tabeli, należy podać po kolei wartości dla wszystkich kolumn dla nowego wiersza. Aby móc podać wartości tylko dla niektórych kolumn (np. gdy nie mamy wszystkich potrzebnych danych lub chcemy, aby zostały wpisane wartości domyślne), można wyszczególnić listę kolumn, do których wpisywane będą wartości, w następujący sposób:

INSERT INTO tabela (kol1, kol2) VALUES (wartość_w_kol1, wartość_w_kol2)

Przykłady:

INSERT INTO personalia VALUES (1,'Adam', 'Ślodosy')

INSERT INTO personalia (Imie) VALUES ('Janosik')

Druga instrukcja wypełni w tabeli (w której, wnioskując z pierwszego zapytania, są trzy kolumny: numeryczna oraz dwie tekstowe) tylko kolumnę o nazwie *Imię*, wpisując do niej tekst "*Janosik*".

Drugi z wariantów instrukcji *INSERT* zbliżony jest w działaniu do wersji z podawaniem listy kolumn:

INSERT INTO tabela SET kol1=wartość_w_kol1, kol2=wartość_w_kol2,...

Ostatni przykład, przepisany w tej składni ma postać:

INSERT INTO personalia SET Imie='Janosik'

Ostatni z wariantów instrukcji *INSERT* służy do wpisania do tabeli danych, uzyskanych w wyniku realizacji pewnego zapytania. Najczęściej wykorzystywany jest on do kopiowania danych pomiędzy tabelami. Przy formułowaniu takiej instrukcji należy pamiętać, aby kolumny, otrzymane w wyniku zapytania, odpowiadały specyfikacją kolumnom, do których dane będą zapisane. Ograniczenia te są zależne od konkretnego RDBMS. Składnia tego typu instrukcji *SQL* wygląda następująco:

INSERT INTO tabela SELECT ...

lub

INSERT INTO tabela (kol1, kol2,...) SELECT ...

Przykład:

INSERT INTO personalia (Imie, Opis) SELECT Pseudonim, Alias FROM artysci

Po wprowadzeniu danych do tabeli często okazuje się, że należy je skorygować. Powodem może być pomyłka przy ich wpisywaniu, jak również późniejsza dezaktualizacja danych, na przykład ze względu na zmianę stanowiska pracownika, jego adresu, stanu cywilnego itp. Zmiana danych oznacza, że do pewnych kolumn należy wpisać nowe wartości w jednym lub większej ilości wierszy. Do zmiany danych w tabeli służy w języku *SQL* instrukcja *UPDATE* (ang. uaktualnij) o następującej składni:

UPDATE tabela SET kol1=wartość_w_kol1, kol2=wartość_w_kol2,... WHERE predykat

Jak widać, zmiany tej dokonuje się poprzez podanie listy par *kolumna-nowa wartość* oraz poprzez ograniczenie zbioru wierszy, w których zmiana ma nastąpić, za pomocą predykatu, umieszczonego po klauzuli *WHERE*. Gdy klauzula *WHERE* oraz predykat zostaną pominięte, zmiana wartości dokonana zostanie we wszystkich wierszach tabeli. Aby uzyskać zmianę danych wyłącznie w jednym wierszu, częste jest opieranie predykatu na kluczu głównym tabeli.

Przykład:

UPDATE personalia SET Imie='Adam' WHERE Imie='Adan'

Zapytanie powyższe poprawia błędnie wpisane imię w kolumnie z imionami pracowników, dla każdego wpisu z błędem, polegającym na zamianie litery "m" na "n" w imieniu *Adam*.

Oprócz poprawiania danych w tabeli często zachodzi konieczność usunięcia z niej danych. Do tego celu służy instrukcja *DELETE* (ang. skasuj) o następującej składni:

DELETE FROM tabela WHERE predykat

Zapytanie powyższe spowoduje usunięcie wszystkich wierszy tabeli spełniających predykat. Możliwe jest pominięcie klauzuli *WHERE* wraz z predykatem, ale...

UWAGA!

**Brak predykatu w instrukcji *DELETE* spowoduje
USUNIĘCIE WSZYSTKICH WIERSZY Z TABELI!**