

Operacje wyboru danych z pojedynczej tabeli, cz.2

Na poprzednich zajęciach mówiliśmy o zapytaniach, korzystających z projekcji: ograniczenia ilości danych, otrzymanych w wyniku zapytania, poprzez wybór kolumn, z których zostaną włączone wartości do tabeli wynikowej. Tak skonstruowane zapytanie zwracało wartości ze wszystkich wierszy tabeli. Nie zawsze jednak interesują nas wartości we wszystkich wierszach, przeciwnie – często potrzebujemy ograniczenia wyniku wręcz do jednego wiersza!

Operację wybrania pewnych wierszy z tabeli nazywamy OGRANICZENIEM (albo SELEKCJA)

Oczywiście musi istnieć możliwość określenia, które wiersze mają znaleźć się w tabeli wynikowej. Realizowane jest to poprzez podanie warunku, zwanego *predykatem*, którego spełnienie powoduje uwzględnienie danego wiersza w tabeli wynikowej. Składnia polecenia *SQL* przedstawia się następująco:

SELECT kolumny FROM tablica WHERE warunek

Predykat jest konstruowany w oparciu o operatory porównania, operatory logiczne oraz specjalizowane klauzule.

Operatory porównania:

- = równy
- < mniejszy
- <= mniejszy lub równy
- > większy
- >= większy lub równy
- <> różny (w niektórych bazach jest obsługiwany również jako !=)

Operatory porównania mogą porównywać ze sobą teksty, a nie tylko wartości typów liczbowych. Działają wtedy następująco: porównywane są parami kody znaków kolejnych liter w obu łańcuchach tekstowych, do momentu natrafienia na różne od siebie znaki na danej pozycji lub koniec któregoś z tekstów – za napis “większy” uznaje się ten, w którym pierwszy różniący napisy znak ma większy kod (w niektórych bazach przy porównaniach nie jest rozróżniana wielkość liter, tak jest w bazie *MySQL*), lub dłuższy z napisów (co realizowane jest w ten sposób, że krótszy z napisów jest uzupełniany przed serią porównań par znaków znakami odstępu, których kod jest mniejszy od kodu dowolnej litery). W wypadku, gdy chcemy zapewnić niezależność wyniku sortowania od wielkości liter, można zastosować do obu porównywanych tekstów jedną z dwóch funkcji: *upper(tekst)*, która zamienia wszystkie litery w tekście na duże, lub *lower(tekst)*, która zamienia wszystkie litery na małe.

Podczas dodawania wiersza do tabeli, nie zawsze trzeba podawać wartości dla wszystkich pól. Wartości, które zostały pominięte, zostają oznaczone w specjalny sposób – jako tzw. *NULL*. Operatory porównań nie są w stanie dokonywać porównań na pustych (tj. nie wpisanych do bazy) wartościach w wierszu. Do sprawdzenia, czy w wierszu istnieje wpis w danym polu, czy nie, służą dwa specjalne warunki:

- kol ***IS NULL*** prawdziwy, gdy nie istnieje wpis w kolumnie *kol*.
- kol ***IS NOT NULL*** prawdziwy, gdy podano wartość w kolumnie *kol*.

Przykłady zapytań z operatorem porównania:

- Wyświetlenie listy osób pełnoletnich
SELECT imie, nazwisko, wiek FROM personalia WHERE wiek >= 18
- Wyświetlenie listy imion i nazwisk osób, z wyjątkiem osób z nazwiskami na litery A, B, oraz C
SELECT imie, nazwisko FROM personalia WHERE UPPER(nazwisko) >= 'D'

Operatory logiczne

W języku *SQL* istnieją trzy operatory logiczne służące do konstruowania złożonych predykatów. Są to, w kolejności malejącego priorytetu:

1. **NOT** logiczne *NIE* (operator jednoargumentowy)
2. **AND** logiczne *I* (operator dwuargumentowy)
3. **OR** logiczne *LUB* (operator dwuargumentowy)

Przykład zapytania ze złożonym predykatem:

SELECT imie, nazwisko, wiek FROM personalia WHERE wiek IS NULL OR wiek >= 10 AND wiek <= 20

Powyższe zapytanie wyświetli listę wszystkich nastolatków, zapisanych w bazie, a ponadto osoby nie posiadające wpisu, dotyczącego wieku (proszę zwrócić uwagę na priorytety operatorów: najpierw *AND*, potem *OR*).

Specjalne klauzule sprawdzające

1. Szukanie wartości z zadanego przedziału - BETWEEN ... AND ...

Klauzula ta powoduje wybranie wierszy, w których wartości w podanej kolumnie należą do zadanego przedziału, wraz z wartościami granicznymi. Składnia predykatu przedstawia się następująco:

kolumna **BETWEEN** wartość_min **AND** wartość_max

Przykład zapytania:

SELECT * FROM personalia WHERE wiek BETWEEN 10 AND 20

2. Szukanie w zbiorze podanych wartości - IN ()

Do wyszukiwania wierszy, w których w jednej z kolumn wartości należy do pewnego zbioru, służy klauzula *IN* (*zbiór*) o składni:

kolumna **IN** (wartosc1, wartosc2, ...)

Przykład zapytania:

SELECT * FROM personalia WHERE zajecie IN ('student', 'asystent')

3. Szukanie względem wzorca tekstowego - LIKE ...

Do wyszukiwania w kolumnach tekstowych wartości, spełniających reguły założone przez użytkownika, służy klauzula *LIKE* o składni:

kolumna **LIKE** 'tekst_wzorca'

W skład tekstu wzorca mogą wchodzić następujące znaki specjalne:

- **%** zastępuje dowolny (także pusty) ciąg znaków
- **_** zastępuje dokładnie jeden, dowolny znak

Przykłady wzorców tekstowych i zgodnych z nimi tekstów:

- **'A%'** dowolny napis, rozpoczynający się na literę *A*
- **'%D%'** dowolny napis, zawierający literę *D*
- **'_K%'** dowolny napis, zawierający literę *K* na drugiej pozycji
- **'%L'** dowolny napis, kończący się na literę *L*
- **'Adam'** napis, ściśle odpowiadający tekstowi *Adam*

Aby uzyskać możliwość wyszukania tekstów, zawierających w treści znaki % oraz _ stosuje się zapis rozszerzony

kolumna **LIKE** 'wzorzec' **ESCAPE** 'znak_ucieczki'

Przykład:

```
SELECT Tytuł FROM Fonoteka WHERE Tytuł LIKE '%100/%%' ESCAPE '/'
```

Powyższe zapytanie odnajdzie wszystkie wiersze z tabeli *Fonoteka*, które mają w kolumnie *Tytuł* wpisany tekst, zawierający fragment *100%*. Znak % zostanie potraktowany jako litera, ponieważ jest we wzorcu poprzedzony znakiem /, zadeklarowanym jako „znak ucieczki”. W bazie danych *MySQL* domyślnym znakiem ucieczki jest backslash (znak \) i nie trzeba go deklarować klauzulą *ESCAPE*.