

SYSTEMY BAZ DANYCH

SQL



TEMAT 21-06: Podzapytania w zapytaniach SELECT.

Autor dokumentu: Wojciech Galiński

czwartek, 3 listopada 2016 r.

351203 Technik informatyk

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

Zagadnienia obowiązkowe

SELECT imie
FROM uczen
WHERE imie **IN**

Zapytanie zewnętrzne

Zapytanie wewnętrzne (podzapytanie)

**(SELECT DISTINCT imie
FROM nauczyciel)** ;

1. **Podzapytanie języka SQL (zapytanie zagnieżdżone)** – to osobne zapytanie SQL (w praktyce jest to zapytanie „SELECT”) (zapytanie wewnętrzne) znajdujące się wewnątrz innego zapytania (zapytania zewnętrznego).

ZAPAMIĘTAJ! Podzapytania muszą być umieszczone w okrągłych nawiasach.

Przypomina to trochę stosowanie zmiennych w proceduralnym języku programowania. Podzapytania wykonywane są w pierwszej kolejności, by ich wyniki zostały wykorzystane do wykonania zapytania głównego.

Zapytania mogą się wielokrotnie zagnieżdżać, ale lepiej rozbić to na kilka zapytań, stosując np. procedury składowane (dzięki temu MySQL szybciej znajdzie wynik).

ZAPAMIĘTAJ! Częstym błędem jest niedopasowanie liczby elementów zwracanych przez podzapytanie do wymagań zapytania zewnętrznego.

Podzapytania wykorzystywane są często w skryptach PHP (podzapytanie może być wówczas wstawiane dynamicznie).

2. **Wielokrotne zagnieżdżanie zapytań** – w MySQL zapytania można wielokrotnie zagnieżdżać, tzn. w zapytaniu może znajdować się podzapytanie, a w nim następne podzapytanie, w którym może znajdować się kolejne podzapytanie.

PRZYKŁADY:

```
-- Lista osób, które kiedykolwiek kupiły jakąkolwiek kurtkę
SELECT nazwisko, imie FROM klient JOIN zakup ON (klient.id=zakup.id_klient)
WHERE zakup.id IN
    ( SELECT id FROM zakup WHERE id_towar IN (SELECT id FROM towar WHERE nazwa LIKE 'koc%') );
```

3. **Podzapytania zamiast łączenia tabel** – podzapytania możemy użyć w miejsce łączenia tabel.

PRZYKŁAD:

Dane są tabele „uczen” i „ocena”. Jeśli chcemy dowiedzieć się np. którzy uczniowie otrzymali oceny niedostateczne, można połączyć tabele za pomocą zapytania:

```
SELECT * FROM uczen INNER JOIN ocena ON ocena.id_uczen = uczen.id WHERE ocena.wartosc = 1;
```

lub wybrać tych uczniów za pomocą podzapytania:

```
SELECT * FROM uczen WHERE id IN (SELECT DISTINCT id_uczen FROM ocena WHERE ocena.wartosc = 1);
```

4. **Miejsca wstawiania podzapytań w kwerendach** – opisuje to poniższa tabela:

Po jakiej klauzuli	Podzapytanie musi zwracać	Dodatkowe wymagania
SELECT, WHERE	pojedynczą wartość	
WHERE ... IN	pojedynczą wartość, listę wartości	
FROM	pojedynczą wartość, listę wartości, tablicę wartości	Podzapytanie musi mieć zdefiniowany alias.

5. **Podzapytania występujące po słowie kluczowym „SELECT”** – to podzapytania niepowiązane zwracające pojedynczą wartość – pełnią rolę zmiennych używanych razem z operatorami relacji. Mogą zawierać funkcję agregującą. Oto wybrane wzorce:

```
SELECT (podzapytanie) FROM tabela;
SELECT pole, (podzapytanie) FROM tabela;
SELECT pole OPERATOR (podzapytanie) FROM tabela;
```

PRZYKŁAD: SELECT Nazwa, Cena, cena-(SELECT Min(cena) FROM towar) AS 'O ile droższy' FROM towar;

6. **Podzapytania po klauzuli „FROM”** – pozwalają na utworzenie źródła przetwarzanych danych. W tym przypadku wymagane jest zdefiniowanie aliasu dla powstałej tabeli. Oto wzorzec:

```
SELECT * FROM (podzapytanie) AS alias;
```

PRZYKŁAD: SELECT * FROM (SELECT imie FROM klasa_4ti) AS k;

7. Podzapytania po klauzuli „WHERE” – dzielimy je na:

- **podzapytania zwracające pojedynczą wartość** – wstawiane są w miejsce pojedynczej wartości albo wyrażenia, którego wynik to pojedyncza wartość;

SELECT * FROM tabela WHERE kolumna OPERATOR (podzapytanie);

PRZYKŁADY:

-- Wyświetla informację, o ile droższe są produkty od najtańszego produktu

SELECT nazwa, cena FROM towar WHERE cena < (SELECT AVG(cena) FROM towar);

-- Wyświetla imiona i nazwiska osób z klasy 1 mi

SELECT imie,nazwisko FROM uczen WHERE id_klasa=(SELECT id FROM klasa WHERE nazwa = '1 mi');

- **podzapytania zwracające listę wartości** – w praktyce ten przypadek sprowadza się do podzapytań z wyrażeniem „IN” oraz „NOT IN”. Operator „IN” / „NOT IN” sprawdza, czy badana wartość **jest / nie jest** jedną z wartości z zadanego zbioru. Podzapytanie jest wykorzystywane do wygenerowania zbioru zadanych wartości (który można podać także bezpośrednio) i musi zwracać pojedynczą wartość albo wartości z jednej kolumny (pola) lub rekordu (wiersza) w tabeli. Oto wzorzec:

SELECT pole1, ... FROM tabela WHERE poleK [NOT] IN (zapytanie_zagnieżdżone_zwracające_co_najwyżej_wektor_danych);

PRZYKŁADY:

-- Wyświetla imiona i nazwiska osób o imiona bezpośrednio wymienionych na liście

SELECT imie,nazwisko FROM uczen WHERE imie IN ("Ewa", "Adam", "Anna");

-- Wyświetla imiona i nazwiska osób o imionach takich, jak imiona uczniów z klasy 4 TI

SELECT imie,nazwisko FROM uczen WHERE imie IN (SELECT DISTINCT imie FROM uczen,klasa WHERE uczen.id_klasa=klasa.id AND klasa.nazwa="4 TI");

-- Osoby, które nie mają na imię tak, jak imiona w tabeli "znane_imiona"

SELECT imie,nazwisko FROM osoba WHERE imie NOT IN (SELECT imie FROM znane_imiona);

8. [4] Podzapytania z operatorami „EXISTS”, „NOT EXISTS” – wyrażenie to służy do sprawdzenia, czy podzapytanie zwraca jakieś wyniki (co najmniej 1 rekord). Oto wzorzec:

SELECT * FROM tabela WHERE EXISTS (dowolne_podzapytanie);

PRZYKŁADY:

-- Zwraca te rekordy, w których imię i nazwisko powtarza się dla różnych id

SELECT id, nazwisko, imie FROM klient K1

WHERE EXISTS

(SELECT id FROM klient K2 WHERE K1.imie=K2.imie AND K1.nazwisko=K2.nazwisko AND K1.id<>K2.id);

9. [4] Podzapytania z operatorami „ANY”, „ALL” – sprawdza, czy relacja jest spełniona dla co najmniej 1 wartości („ANY”) albo dla wszystkich wartości („ALL”) w wyniku podzapytania, co pozwala na sprawdzanie różnych relacji pojedynczej wartości z listą wartości.

Oto wzorzec:

SELECT * FROM tabela WHERE pole_K OPERATOR ANY (podzapytanie);

SELECT * FROM tabela WHERE pole_K OPERATOR ALL (podzapytanie);

gdzie **OPERATOR** to jeden z operatorów: „=”, „>”, „>=”, „<”, „<=”.

PRZYKŁADY:

SELECT * FROM towar WHERE koszt > ANY (SELECT zysk FROM towar);

SELECT * FROM towar WHERE zysk <= ALL (SELECT koszt FROM towar);

Pytania kontrolne

1. Po jakich klauzulach i na jakich zasadach można umieszczać podzapytania?
2. W jaki sposób wykorzystujemy słowa kluczowe „EXISTS”, „NOT EXISTS”, „ANY”, „ALL” stosując podzapytania?
3. Po jakich słowach kluczowych stosujemy podzapytania zwracające **pojedynczą wartość / listę wartości / tablicę wartości**?

Zadania

1. Znajdź różnice pomiędzy aktualnymi i średnimi cenami towarów w magazynie.
2. Znajdź te towary, których cena jest większa od średniej ceny towarów zwiększonej o 1 odchylenie standardowe.
3. Znajdź uczniów ze wszystkich klas 4 (wykonaj 2 zapytania: za pomocą złączenia tabel oraz za pomocą podzapytania). Które zapytanie wykonuje się szybciej na Twoim komputerze?
4. Znajdź klientów biblioteki, którzy wypożyczyli co najmniej 1 książkę oraz takich, którzy nic nie wypożyczyli (użyj podzapytania oraz poznanych operatorów).