

SYSTEMY BAZ DANYCH SQL



TEMAT 23-06: Widoki, czyli perspektywy.

Autor dokumentu: Wojciech Galiński

sobota, 19 września 2015 r.

351203 Technik informatyk

ŹRÓDŁA WIEDZY: <http://dev.mysql.com/doc/>, <http://webmaster.helion.pl/index.php/kurs-mysql>.

Zagadnienia obowiązkowe

1. **Pojęcie widoku** – to swego rodzaju kopia zapytania, uruchamiana za każdym razem, gdy odwołujemy się do niego, dzięki czemu pracujemy zawsze na aktualnych danych z tabeli bazowej (to tabela, do której odwołujemy się w definicji widoku).
2. **Korzyści płynące ze stosowania widoków** – można tu wyróżnić:
 - **prostsze odwoływanie się do danych** – w widoku wybieramy interesujące nas pola (kolumny), rekordy (wiersze) z wybranych przez nas tabel;
 - **ukrywanie efektów normalizacji** – za pomocą widoku możemy np. złączyć kilka tabel w jedną tabelę;
 - **ograniczanie danych dostępnych dla użytkowników** – w widoku udostępniamy tylko te pola i rekordy, które powinny być dostępne i nic poza tym;
 - **tworzenie dodatkowej warstwy abstrakcji** – jeśli np. w skrypcie PHP odwołujemy się do widoku, a nie bezpośrednio do zapytania, to możemy zmodyfikować widok bez potrzeby zmian w skrypcie PHP.
3. **Tworzenie widoku** – służy do tego „CREATE OR REPLACE VIEW”. Oto wzorzec:

CREATE [OR REPLACE] VIEW *widok* AS (*zapytanie*) [WITH CHECK OPTION];

PRZYKŁAD:

```
CREATE VIEW v_uczniowie AS
  SELECT CONCAT(imie, ' ', nazwisko) AS imie_i_nazwisko, klasa FROM uczen;
CREATE VIEW v_towary_na_stanie AS
  SELECT nazwa, kategoria FROM towar WHERE stan_w_magazynie>0;
CREATE VIEW v_osoby AS
  SELECT imie,nazwisko FROM uczen
  UNION
  SELECT imie,nazwisko FROM nauczyciel;
```

4. **Modyfikowanie danych poprzez widoki** – jest możliwe po spełnieniu następujących warunków:
 - każdy wiersz widoku odpowiada wierszowi w tabeli bazowej (zatem nie mogą w zapytaniu występować: wyrażenia matematyczne, funkcje grupujące, operator „DISTINCT”, klauzule „GROUP BY” oraz „HAVING”);
 - w pojedynczym zapytaniu modyfikujemy pojedynczą tabelę bazową (ale sam widok może odwoływać się do wielu tabel);
 - widok nie narusza dodatkowych ograniczeń ustalonych w definicji tabeli (np. „NOT NULL”, „UNIQUE”, „WITH CHECK OPTION”);
 - widok może odwoływać się tylko do widoków, które można również modyfikować.
5. **Warunek „WITH CHECK OPTION”** – opcja ta ma zadanie pilnować pozornej spójności danych w widokach, np. nie zezwala na modyfikowanie danych w widoku zawierającym warunek w taki sposób, że zmodyfikowane dane przestają spełniać ten warunek.

PRZYKŁAD:

```
CREATE VIEW v_stefany AS
  SELECT id,imie,nazwisko FROM osoba WHERE imie='Stefan';
UPDATE v_stefany SET imie='Franciszek' WHERE id<10;
```

6. **Modyfikowanie widoku** – służy do zmiany zapytania z zachowaniem uprawnień użytkowników do widoku. Używamy do tego instrukcji:

ALTER VIEW *widok* AS (*zapytanie*);

PRZYKŁAD:

```
ALTER VIEW v_uczniowie AS  
SELECT imie, nazwisko, klasa FROM uczen ORDER BY klasa,nazwisko,imie;
```

7. **Usuwanie widoku** – służy do tego następujące zapytanie SQL:

DROP VIEW *widok*;

PRZYKŁAD:

```
DROP VIEW v_stefany;
```

Pytania kontrolne

1. Czym jest widok (perspektywa) w języku SQL?
2. Jakie korzyści płyną ze stosowania widoków?
3. W jaki sposób utworzyć widok w języku SQL?
4. Jakie są zasady modyfikowania danych poprzez widoki?
5. Co oznacza warunek „WITH CHECK OPTION”?
6. W jaki sposób w języku SQL modyfikujemy i usuwamy widoki?

Zadania

1. Dodać 2 nowe widoki, jeden z nich zmodyfikować, a drugi – usunąć tak, żeby w konsoli pozostał ślad umożliwiający ocenę (zapytania i wyniki tych zapytań).
2. Do własnej witryny utworzyć co najmniej 3 widoki konsultując to z nauczycielem prowadzącym.