Bazy danych

10. SQL — Widoki

P. F. Góra

http://th-www.if.uj.edu.pl/zfs/gora/

semestr letni 2005/06

Widoki, *AKA* Perspektywy

W SQL tabela, którą utworzono za pomocą zapytania `CREATE TABLE`, nazywa się *tabelą podstawową* (*base table*). Jej struktura i dane przechowywane są przez system operacyjny.

Wynik każdego zapytania `SELECT` też, formalnie, jest tabelą. Tabelę taką nazywa się *tabelą pochodną* (*derived table*).

Widokiem (lub perspektywą) nazywam trwałą definicję tabeli pochodnej, która to definicja przechowaywana jest w bazie danych.

Jak tworzymy widoki?

```
CREATE VIEW nazwa
AS SELECT treść_zapytania_select;
```

Przykład

Tworzymy następujące tabele:

```
mysql> CREATE TABLE arabic
    -> (i INT UNSIGNED NOT NULL, b VARCHAR(8)) CHARSET=cp1250;
Query OK, 0 rows affected (0.87 sec)
mysql> CREATE TABLE roman (i INT UNSIGNED NOT NULL, r VARCHAR(4));
Query OK, 0 rows affected (0.42 sec)
mysql> SET CHARSET cp1250;
Query OK, 0 rows affected (0.04 sec)
mysql> INSERT INTO arabic VALUES
    -> (1,'jeden'),(2,'dwa'),(3,'trzy'),(4,'cztery'),(5,'pięć');
Query OK, 5 rows affected (0.13 sec)
Records: 5  Duplicates: 0  Warnings: 0
mysql> INSERT INTO roman VALUES
    -> (1,'I'),(2,'II'),(3,'III'),(4,'IV');
Query OK, 4 rows affected (1.17 sec)
Records: 4  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM arabic;
+---+--------+
| i | b      |
+---+--------+
| 1 | jeden  |
| 2 | dwa    |
| 3 | trzy   |
| 4 | cztery |
| 5 | pięć   |
+---+--------+
5 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM roman;
+---+------+
| i | r    |
+---+------+
| 1 | I    |
| 2 | II   |
| 3 | III  |
| 4 | IV   |
+---+------+
4 rows in set (0.00 sec)
```

Tworzymy najprostszy widok

```
mysql> CREATE VIEW v1
    -> AS SELECT * FROM arabic;
Query OK, 0 rows affected (1.14 sec)

mysql> SELECT * FROM v1;
+---+--------+
| i | b      |
+---+--------+
| 1 | jeden  |
| 2 | dwa    |
| 3 | trzy   |
| 4 | cztery |
| 5 | pięć   |
+---+--------+
5 rows in set (1.13 sec)
```

Tak utworzony widok jest w gruncie rzeczy aliasem tabeli.

Widoki można tworzyć na podstawie widoków.

Wyrażenie `SELECT` w definicji widoku może zawierać klauzulę `WHERE`.

Można brać tylko wybrane kolumny. Można nadawać im aliasy.

```
mysql> CREATE VIEW v2
    -> AS SELECT b AS napis FROM v1
    -> WHERE i > 3;
Query OK, 0 rows affected (0.15 sec)

mysql> SELECT * FROM v2;
+--------+
| napis  |
+--------+
| cztery |
| pięć   |
+--------+
2 rows in set (0.05 sec)
```

Underlying tables

Tabele, których użyto do zdefiniowania widoku, są nazywane *underlying tables*.

Widok `v1` ma jedną underlying table: `arabic`. Widok `v2` ma *dwie* underlying tables: tabelę `arabic` oraz widok `v1`.

Widoki a zmiana *underlying table*

Widoki reagują na zmianę danych w tabelach, których użyto do ich stworzenia:

```
mysql> INSERT INTO arabic VALUES(6,'sześć');
Query OK, 1 row affected (0.43 sec)

mysql> SELECT * FROM v2;
+--------+
| napis  |
+--------+
| cztery |
| pięć   |
| sześć  |
+--------+
3 rows in set (0.38 sec)
```

Widoki nie reagują na zmianę *definicji* tabeli, o ile nie narusza ona integralności widoku.

```
mysql> ALTER TABLE arabic ADD COLUMN (x CHAR(1) DEFAULT 'X');
Query OK, 6 rows affected (4.51 sec)
Records: 6  Duplicates: 0  Warnings: 0
mysql> SELECT * FROM arabic WHERE i>4;
+---+--------+------+
| i | b      | x    |
+---+--------+------+
| 5 | pięć   | X    |
| 6 | sześć  | X    |
+---+--------+------+
2 rows in set (0.89 sec)
mysql> SELECT * FROM v1 WHERE i>4;
+---+--------+
| i | b      |
+---+--------+
| 5 | pięć   |
| 6 | sześć  |
+---+--------+
2 rows in set (0.00 sec)
```

Można tworzyć całkiem skomplikowane widoki

```
mysql> CREATE VIEW lewy
    -> AS SELECT arabic.i AS i, b, r FROM arabic LEFT JOIN roman
    -> ON arabic.i=roman.i;
Query OK, 0 rows affected (1.23 sec)
mysql> SELECT * FROM lewy;
+---+--------+------+
| i | b      | r    |
+---+--------+------+
| 1 | jeden  | I    |
| 2 | dwa    | II   |
| 3 | trzy   | III  |
| 4 | cztery | IV   |
| 5 | pięć   |      |
| 6 | sześć  |      |
+---+--------+------+
6 rows in set (0.18 sec)
```

Po co widoki?

```
mysql> CREATE TABLE Ceny
    -> (ProducentId SMALLINT UNSIGNED NOT NULL,
    -> ProduktId SMALLINT UNSIGNED NOT NULL,
    -> Cena DECIMAL(10,2) NOT NULL,
    -> PRIMARY KEY(ProducentId, ProduktId));
mysql> CREATE TABLE Dostawy
    -> (NrDostawy INT UNSIGNED NOT NULL PRIMARY KEY,
    -> ProducentId SMALLINT UNSIGNED NOT NULL,
    -> ProduktId SMALLINT UNSIGNED NOT NULL,
    -> Ilosc INT UNSIGNED NOT NULL);

mysql> CREATE VIEW ksiegowosc
    -> AS SELECT ProducentID, Ilosc*Cena AS Kwota
    -> FROM Dostawy NATURAL JOIN Ceny;
mysql> CREATE VIEW produkcja
    -> AS SELECT ProduktId, SUM(Ilosc) AS Zapas
    -> FROM Dostawy
    -> GROUP BY ProduktId;
```

W bazie zapamiętane zostaną tylko *definicje* poszczególnych kolumn widoków, nie zaś wartości!

Różni użytkownicy bazy chcą/mogą/powinni mieć różny dostęp do tych samych danych źródłowych.

Definiowanie widoków pozwala na przechowywanie całkiem złożonych definicji w bazie danych i późniejsze wykorzystywanie ich w zapytaniach. Jest to szczególnie wygodne gdy z tą samą bazą współpracują *różne* aplikacje.

Widoki modyfikowalne

Niektóre widoki są *modyfikowalne* — zmiana danych w widoku powoduje zmianę danych w underlying tables.

```
mysql> UPDATE v2 SET napis="six" WHERE napis LIKE "sz%";
Query OK, 1 row affected (1.33 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM arabic;
+---+--------+------+
| i | b      | x    |
+---+--------+------+
| 1 | jeden  | X    |
| 2 | dwa    | X    |
| 3 | trzy   | X    |
| 4 | cztery | X    |
| 5 | pięć   | X    |
| 6 | six    | X    |
+---+--------+------+
6 rows in set (0.00 sec)
```

Ale...

```
mysql> INSERT INTO v2 VALUES ('siedem');
ERROR 1423 (HY000): Field of view 'widoki.v2' underlying table
doesn't have a default value
```

Nie można dodać lub zmienić wiersza widoku modyfikowalnego, jeżeli zmiana taka naruszałaby więzy integralności underlying tables.

Gdyby kolumna `i` tabeli `arabic` była określona jako `AUTO_INCREMENT` lub też gdyby nie była określona jako `NOT NULL`, powyższe wstawienie wiersza do widoku zadziałałoby.

Jeżeli zatem chcemy udostępniać użytkownikom (aplikacjom) widoki, które mają być modyfikowalne, trzeba *dobrze* przemyśleć strukturę całej bazy.

Widoki niemodyfikowalne

Niektóre widoki są z założenia niemodyfikowalne: Niemodyfikowalne są te, które w liście `SELECT` zawierają `UNION`, `UNION ALL`, `DISTINCT`, `DISTINCTROW`, inny widok niemodyfikowalny lub podzapytanie. W MySQL widoki w ogóle nie mogą zawierać podzapytań.

Usuwanie wierszy z widoków

Niekiedy z widoków modyfikowalnych można usuwać wiersze — mianowicie wtedy, gdy SQL potrafi “przetłumaczyć” zapytanie usuwające wiersze z widoku na zapytanie (zapytania) usuwające wiersze z underlying table(s).

```
mysql> DELETE FROM v2 WHERE napis="pięć";
Query OK, 1 row affected (1.24 sec)

mysql> SELECT * FROM arabic;
+---+--------+------+
| i | b      | x    |
+---+--------+------+
| 1 | jeden  | X    |
| 2 | dwa    | X    |
| 3 | trzy   | X    |
| 4 | cztery | X    |
| 6 | six    | X    |
+---+--------+------+
5 rows in set (0.00 sec)
```

Ale...

```
mysql> SELECT * FROM lewy;
+---+--------+------+
| i | b      | r    |
+---+--------+------+
| 1 | jeden  | I    |
| 2 | dwa    | II   |
| 3 | trzy   | III  |
| 4 | cztery | IV   |
| 6 | six    |      |
+---+--------+------+
5 rows in set (1.10 sec)

mysql> DELETE FROM LEWY WHERE r='I';
ERROR 1395 (HY000): Can not delete from join view 'widoki.lewy'
```

Więz CHECK w widokach

```
CREATE VIEW nazwa
AS SELECT treść_zapytania_select WHERE warunek
[WITH [CASCADED | LOCAL] CHECK OPTION];
```

Jeśli w zapytaniu tworzącym widok jest klauzula WHERE, więz CHECK uniemożliwi takie dodanie/zmodyfikowanie danych *do widoku*, które naruszałoby tę klauzulę.

CASCADED i LOCAL mają znaczenie jeśli widok tworzony jest na podstawie innych widoków. Jeżeli wybierzemy CASCADED, klauzule WHERE "podwidoków" też są sprawdzane, nawet jeśli nie nałożono na nie jawnie więzu CHECK.

Przykład

```
mysql> CREATE VIEW v3
    -> AS SELECT i, b FROM arabic
    -> WHERE i < 6;
Query OK, 0 rows affected (1.13 sec)
```

```
mysql> CREATE VIEW v4
    -> AS SELECT i, b FROM arabic
    -> WHERE i < 6
    -> WITH CHECK OPTION;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> SELECT * FROM v3;
+---+--------+
| i | b      |
+---+--------+
| 1 | jeden  |
| 2 | dwa    |
| 3 | trzy   |
| 4 | cztery |
+---+--------+
4 rows in set (0.00 sec)
```

```
mysql> SELECT * FROM v4;
+---+--------+
| i | b      |
+---+--------+
| 1 | jeden  |
| 2 | dwa    |
| 3 | trzy   |
| 4 | cztery |
+---+--------+
4 rows in set (0.00 sec)
```

Na tym etapie nie ma różnicy — klauzula `WHERE` wybrała odpowiednie wiersze.

```
mysql> INSERT INTO v3 VALUES (7,'siedem');
Query OK, 1 row affected (1.10 sec)

mysql> SELECT * FROM v3;
+---+--------+
| i | b      |
+---+--------+
| 1 | jeden  |
| 2 | dwa    |
| 3 | trzy   |
| 4 | cztery |
+---+--------+
4 rows in set (0.01 sec)

mysql> INSERT INTO v4 VALUES (8,'osiem');
ERROR 1369 (HY000): CHECK OPTION failed 'widoki.v4'
```

Zawartość widoku `v3` nie zmieniła się, ale pozwolił on na modyfikację underlying table, co możnaby sprawdzić wykonując zapytanie `SELECT * FROM arabic`. Widok `v4`, z klauzulą `CHECK`, na to nie pozwolił.

Widoki “dyndające”

Jeżeli usuniemy tabelę, nie usuniemy zaś opartych na niej widoków, lub też tak zmodyfikujemy tabelę, że definicja widoku straci sens, dostaniemy widoki “dyndające” (*dangling views*). Widoki są wypisywane na liście `SHOW TABLES`;

```
mysql> SHOW TABLES;
+------------------+
| Tables_in_widoki |
+------------------+
| arabic           |
| ceny             |
| dostawy          |
| ksiegowosc       |
| lewy             |
| produkcja        |
| roman            |
| v1               |
| v2               |
| v3               |
| v4               |
+------------------+
11 rows in set (1.12 sec)
```

Aby dowiedzieć się czy dana tabela lub widok nie są uszkodzone, dajemy polecenie

```
mysql> CHECK TABLE roman;
+--------------+-------+----------+----------+
| Table        | Op    | Msg_type | Msg_text |
+--------------+-------+----------+----------+
| widoki.roman | check | status   | OK       |
+--------------+-------+----------+----------+
1 row in set (1.23 sec)

mysql> CHECK TABLE lewy;
+-------------+-------+----------+----------+
| Table       | Op    | Msg_type | Msg_text |
+-------------+-------+----------+----------+
| widoki.lewy | check | status   | OK       |
+-------------+-------+----------+----------+
1 row in set (0.16 sec)
```

W tym momencie wszystko jest w porządku.

Ale...

```
mysql> ALTER TABLE roman DROP COLUMN r;
Query OK, 4 rows affected (1.88 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> CHECK TABLE lewy;
+-------------+-------+----------+-----------------------------------------+
| Table       | Op    | Msg_type | Msg_text                                |
+-------------+-------+----------+-----------------------------------------+
| widoki.lewy | check | error    | View 'widoki.lewy' references invalid   |
|             |       |          | table(s) or column(s) or function(s) or |
|             |       |          | definer/invoker of view lack rights to use |
|             |       |          | them                                    |
+-------------+-------+----------+-----------------------------------------+
1 row in set (0.02 sec)

mysql> DROP VIEW lewy;
Query OK, 0 rows affected (0.01 sec)
```