SYSTEMY BAZ DANYCH
SQL

TEMAT 25-06: Transakcje.

Autor dokumentu: Wojciech Galiński	czwartek, 9 lutego 2017 r.	351203 Technik informatyk
ŹRÓDŁA WIEDZY:	http://dev.mysql.com/doc/, http://webmaster.helion.pl/index.php/kurs-mysql.	

Zagadnienia obowiązkowe

1. **Przetwarzanie transakcyjne** – to przetwarzanie wielu instrukcji tak, jakby to była pojedyncza instrukcja języka SQL. Transakcje są dostępne tylko w niektórych silnikach baz danych, np. „InnoDB”. Nie są dostępne w „phpMyAdmin”.
2. **Cechy transakcji** – każda transakcja powinna być zgodna z zasadą ACID. Są to:
   - ***niepodzielność*** – (ang. **A**tomicity) wszystkie instrukcje wchodzące w skład transakcji muszą być poprawne (gdy tak nie jest – CAŁA transakcja jest odrzucana);
   - ***spójność*** – (ang. **C**onsistency) to gwarancja, że spójność jest zawsze zachowana (nawet w przypadku awarii serwera);
   - ***izolacja*** – (ang. **I**solation) oznacza zabezpieczanie danych przed modyfikacją przez innych użytkowników podczas trwania transakcji (poprzez blokowanie albo kopiowanie danych, do których odwołujemy się podczas transakcji).

     Niestety, całkowite spełnienie tego warunku spowoduje bardzo duże obniżenie przepustowości serwera, dlatego, w zależności od potrzeb, stosuje się różne poziomy izolowania danych (patrz: następny temat);
   - ***trwałość*** – (ang. **D**urability) jeśli transakcja zostanie zatwierdzona, to mamy gwarancję, że zmiany zostaną zapisane nawet w przypadku awarii serwera.
3. **Transakcje w MySQL** – domyślnie, każde zapytanie SQL traktowane jest jako osobna transakcja składająca się z pojedynczej instrukcji języka SQL.

   Sytuacja zmienia się diametralnie po użyciu instrukcji:

   ```
   START TRANSACTION;
   ```
   albo
   ```
   BEGIN [WORK];
   ```

   Po wywołaniu jednej z powyższych instrukcji przejdziemy do trybu, w którym transakcję trzeba zakończyć samodzielnie poprzez użycie jednej z następujących instrukcji:

   ```
   COMMIT;   -- zatwierdzenie CAŁEJ transakcji
   ```
   albo
   ```
   ROLLBACK;   -- odrzucenie CAŁEJ transakcji
   ```

   Transakcje umożliwiają traktowanie wielu zapytań SQL jako jednego zapytania, które jest wykonywane albo w całości, albo wcale (wyjątkiem jest użycie punktu zachowania).

   PRZYKŁAD:

   Najpierw przygotowujemy tabelę:

   ```
   CREATE TABLE pory_roku (tekst VARCHAR(30));
   INSERT INTO pory_roku VALUES ('?'), ('wszystko budzi się do życia');
   SELECT * FROM pory_roku;
   ```

   Następnie wykonujemy transakcję:

   ```
   START TRANSACTION; -- rozpoczynamy transakcję
     INSERT INTO pory_roku VALUES ('gorące wakacje');
     INSERT INTO pory_roku VALUES ('flora ma różne kolory');
     INSERT INTO pory_roku VALUES ('biało wszędzie');
     INSERT INTO pory_roku VALUES ('takiej pory roku nie ma');
     SELECT * FROM pory_roku;
   ROLLBACK;    -- wybieramy, bo chcemy cofnąć całą transakcję
   SELECT * FROM pory_roku;
   ```

4. **Punkty zachowania** – to swego rodzaju zakładki z nazwą pozwalająca cofnąć fragment transakcji (zamiast całej).

   Aby utworzyć punkt zachowania, stosujemy polecenie:

   ```
   SAVEPOINT nazwa_punktu_zachowania;
   ```

   W celu przywrócenia transakcji do punktu zachowania stosujemy polecenie:

   ```
   ROLLBACK TO SAVEPOINT nazwa_punktu_zachowania;
   ```

   PRZYKŁAD:

   Wykonujemy transakcję korzystając z tabeli z poprzedniego przykładu:

```
START TRANSACTION; -- rozpoczynamy transakcję
  INSERT INTO pory_roku VALUES ('gorące wakacje');
  SAVEPOINT lato;
  INSERT INTO pory_roku VALUES ('flora ma różne kolory');
  SAVEPOINT jesien;
  INSERT INTO pory_roku VALUES ('biało wszędzie');
  SAVEPOINT zima;
  INSERT INTO pory_roku VALUES ('takiej pory roku nie ma');
  SELECT * FROM pory_roku;
  ROLLBACK TO SAVEPOINT lato;  -- wybieramy, bo chcemy cofnąć fragment transakcji
  SELECT * FROM pory_roku;
COMMIT;    -- wybieramy, bo chcemy zatwierdzić tą część transakcji, która nie została wycofana
```

   Na koniec sprzątamy bazę danych:

```
DROP TABLE pory_roku;
```

5. **Przykład transakcji przelewu z użyciem transakcyjnego przetwarzania danych** – oto kod:

   Przygotowanie bazy danych do transakcji:

```
CREATE TABLE konta
(
  id SERIAL,
  nr_konta CHAR(24),
  wn DECIMAL(12, 2),
  ma DECIMAL(12, 2)
);

CREATE VIEW konto_001 AS
  SELECT * FROM konta WHERE nr_konta='001';
CREATE VIEW stan_konta_001 AS
  SELECT SUM(wn)-SUM(ma) AS Stan_konta_001 FROM konta WHERE nr_konta='001';

CREATE VIEW konto_002 AS
  SELECT * FROM konta WHERE nr_konta='002';
CREATE VIEW stan_konta_002 AS
  SELECT SUM(wn)-SUM(ma) AS Stan_konta_002 FROM konta WHERE nr_konta='002';

INSERT INTO konta(nr_konta, wn, ma) VALUES ('001', 1000, 0), ('002', 1000, 0);

SELECT * FROM konto_001;    SELECT * FROM stan_konta_001;
SELECT * FROM konto_002;    SELECT * FROM stan_konta_002;
```

   Przelew kwoty 500 zł z konta „001” na konto „002” i wyświetlenie wyników tej transakcji:

```
START TRANSACTION;
  INSERT INTO konta(nr_konta, wn, ma) VALUES ('001', 0, 500);
  INSERT INTO konta(nr_konta, wn, ma) VALUES ('002', 500, 0);
COMMIT;    -- wybieramy, bo chcemy zatwierdzić całą transakcję

SELECT * FROM konto_001;    SELECT * FROM stan_konta_001;
SELECT * FROM konto_002;    SELECT * FROM stan_konta_002;
```

   Posprzątanie bazy danych:

```
DROP VIEW konto_001;    DROP VIEW stan_konta_001;
DROP VIEW konto_002;    DROP VIEW stan_konta_002;
DROP TABLE konta;
```

Pytania kontrolne

1. Na czym polega przetwarzanie transakcyjne?
2. Jakie cechy powinna mieć każda transakcja?
3. W jaki sposób wykonać transakcję w MySQL?
4. W jaki sposób można anulować fragment transakcji?
5. Podczas jakich operacji serwer MySQL używa przetwarzania transakcyjnego?

Zadania

1. Wykonaj i przeanalizuj przykłady z punktów: 3 i 4.
2. Wykonaj i przeanalizuj przykład przelewu z punktu 5.