

Triggery - wyzwalacze

Triggery - wyzwalacze

Trigger - (wyzwalacz) służy do automatycznego wykonania poleceń proceduralnych, po zajściu danego zdarzenia na tabeli w bazie danych.

Zdarzeniem może być (SQL DML - dla MySQL/MariaDB): insert, update lub delete. Dla triggera powinien zostać również określony czas jego aktywacji, tj. przed (before) lub po (after) danej akcji (SQL DML). Wyzwalacze nie są aktywowane przez akcje na referencjach FK-PK.

Wyzwalacz w MySQL/MariaDB posiada wymaganą (tj. nieopcjonalną) klauzulę FOR EACH ROW. Oznacza ona, że wyzwalacz zostanie uruchomiony dla każdego rekordu, który został wywołany powyższym poleceniem SQL DML. Niektóre systemy bazodanowe umożliwiają utworzenie triggera bez powyższej klauzuli (tzn. jest ona opcjonalna, np. Oracle). Wtedy trigger jest wywoływany jednorazowo dla akcji SQL DML, bez znaczenia jest liczba rekordów „dotkniętych” poleceniem SQL DML.

Więcej informacji:

<https://mariadb.com/kb/en/trigger-overview/>

Podstawy baz danych / Artur Niewiarowski

<https://dev.mysql.com/doc/refman/8.0/en/triggers.html>

Triggery - wyzwalacze

Uproszczony schemat tworzenia triggera:

```
CREATE [OR REPLACE]
  TRIGGER [IF NOT EXISTS] trigger_name trigger_time trigger_event
  ON tbl_name FOR EACH ROW
  [{ FOLLOWS | PRECEDES } other_trigger_name ]
  trigger_stmt;
```

trigger_time: before, after

trigger_event: insert, update, delete

W najnowszych SZBD można tworzyć wiele wyzwalaczy o tym samym trigger_time i trigger_event. O kolejności wykonania decyduje czas ich utworzenia lub użycie opcji FOLLOWS ewentualnie PRECEDES przez użytkownika.

Kolejność wykonywania triggerów o tych samych parametrach pokazuje polecenie:

```
mysql> SELECT trigger_name, action_order FROM information_schema.triggers
       WHERE event_object_table='nazwa tabeli';
+-----+-----+
| trigger_name          | action_order |
+-----+-----+
| male_litery_update    |            1 |
| jednak_duze_litery_update |            2 |
| zapamietaj_usuwanie   |            1 |
+-----+-----+
3 rows in set
```

Triggery - wyzwalacze

Schemat usuwania triggera:

```
DROP TRIGGER [IF EXISTS] [schema_name.]trigger_name
```

Przykład:

```
drop trigger if exists t_Klienci_insert;
```

Trigger zostanie usunięty, jeżeli istnieje. Jeżeli nie istnieje, to nie będzie generowanej informacji o błędzie związanym z brakiem wyzwalacza o podanej nazwie.

Triggery - wyzwalacze

Przykład:

```
CREATE TRIGGER male_litery_before_update BEFORE  
UPDATE ON uzytkownicy FOR EACH ROW begin  
  
set NEW.login = lower(NEW.login);  
  
end;
```

Wyzwalacz uruchamia się przed (BEFORE) fizycznym pojawieniem się zmian w tabeli. Dane są podmieniane przed ich aktualizacją (UPDATE).

Triggery - wyzwalacze

Wyzwalacze posiadają następujące odwołania do danych:

NEW.kolumna z wartością - NEW występuje wyłącznie dla wyzwalaczy z akcjami: insert oraz update - ponieważ te polecenia SQL DML umożliwiają wprowadzenie nowych wartości.

OLD.kolumna z wartością - OLD występuje wyłącznie dla wyzwalaczy z akcjami: update oraz delete - ponieważ te polecenia SQL DML operują na starych wartościach.

Próba odwołania do NEW/OLD w ciele triggera niezgodnie z powyższym, wywoła błąd podczas tworzenia wyzwalacza.

Triggery - wyzwalacze

Podmiana danych przed ich fizycznym wprowadzeniem do tabeli jest możliwa wyłącznie z poziomu następujących parametrów triggera:

before update, before insert - wyłącznie dla NEW.

Próba odwołania w ciele triggera niezgodnie z powyższym, wywoła błąd podczas tworzenia wyzwalacza.

Triggery - wyzwalacze

Przykłady zastosowania momentu wywołania triggera dla:

- before - podmiana danych, automatyczne uzupełnienie pozostałych wartości w wierszu na bazie wprowadzanych danych; sprawdzenie, czy wprowadzane dane mogą zostać umieszczone i ewentualnie wygenerowanie wyjątku uniemożliwiającego ich wprowadzenie;
- after - automatyczne uzupełnienie innych tabel danymi na podstawie wprowadzonych; np. zapamiętywanie zmian dokonywanych na rekordach tabeli źródłowej w tabeli archiwalnej poprzez przekopiowanie rekordu z danymi przed zmianą

Triggery - wyzwalacze

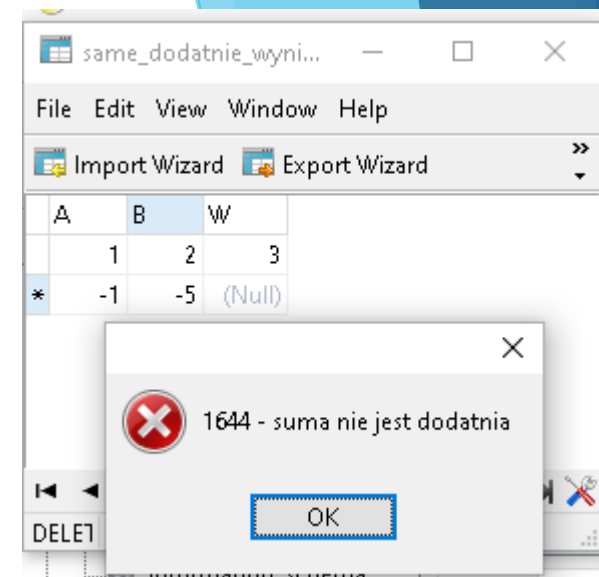
Przykłady zastosowania momentu wywołania triggera

odmowa umieszczenia rekordu z poziomu triggera

```
CREATE TABLE `same_dodatnie_wyniki` (  
  `A` int(11) DEFAULT NULL,  
  `B` int(11) DEFAULT NULL,  
  `W` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
CREATE TRIGGER `before_insert_sdw` BEFORE INSERT ON `same_dodatnie_wyniki`  
  FOR EACH ROW trigger1:  
begin  
  if (NEW.A+NEW.B < 0) then  
    SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'suma nie jest dodatnia';  
  else  
    set NEW.W = NEW.A+NEW.B;  
  LEAVE trigger1;  
end if;  
end;
```

Polecenie LEAVE - umożliwia opuszczenie bloku o nazwie trigger1 (nazwanego przez programistę), czyli natychmiastowe wyjście z triggera



Triggery - wyzwalacze

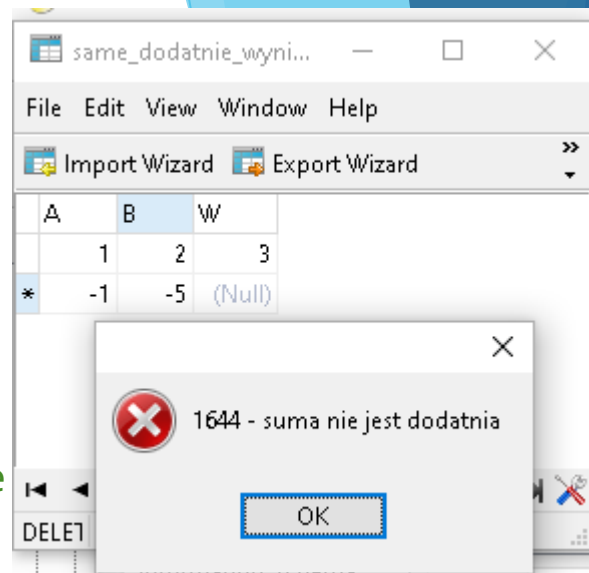
Przykłady zastosowania momentu wywołania triggera

odmowa umieszczenia rekordu z poziomu triggera

```
SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'suma nie
```

SIGNAL - generuje wyjątek (błąd), który przerywa działanie polecenia SQL i propaguje informację o błędzie dalej, w tym do środowiska programistycznego, które łączy się z SZBD.

Powyższy wyjątek jest wyjątkiem użytkownika z jego własnym komunikatem.



Triggery - wyzwalacze

W najnowszych systemach MariaDB wprowadzono możliwość tworzenia sekwencji na wzór sekwencji z szbd Oracle. Sekwencje umożliwiają inkrementację wartości w celu późniejszego wykorzystania np. w PK.

Sekwencje w MariaDB oparte są o specjalną tabelę, która tworzona jest automatycznie po wykonaniu polecenia „create sequence”.

Tego typu mechanizm można z powodzeniem wywołać w wyzwalaczu, jako alternatywę do auto_increment.

```
mysql> CREATE SEQUENCE moja_sekwencja START WITH 1 INCREMENT BY 2;  
Query OK, 0 rows affected
```

```
mysql> select NEXTVAL(moja_sekwencja);
```

```
+-----+  
| NEXTVAL(moja_sekwencja) |  
+-----+  
| 1 |  
+-----+
```

```
1 row in set
```

```
mysql> select NEXTVAL(moja_sekwencja);
```

```
+-----+  
| NEXTVAL(moja_sekwencja) |  
+-----+  
| 3 |  
+-----+
```

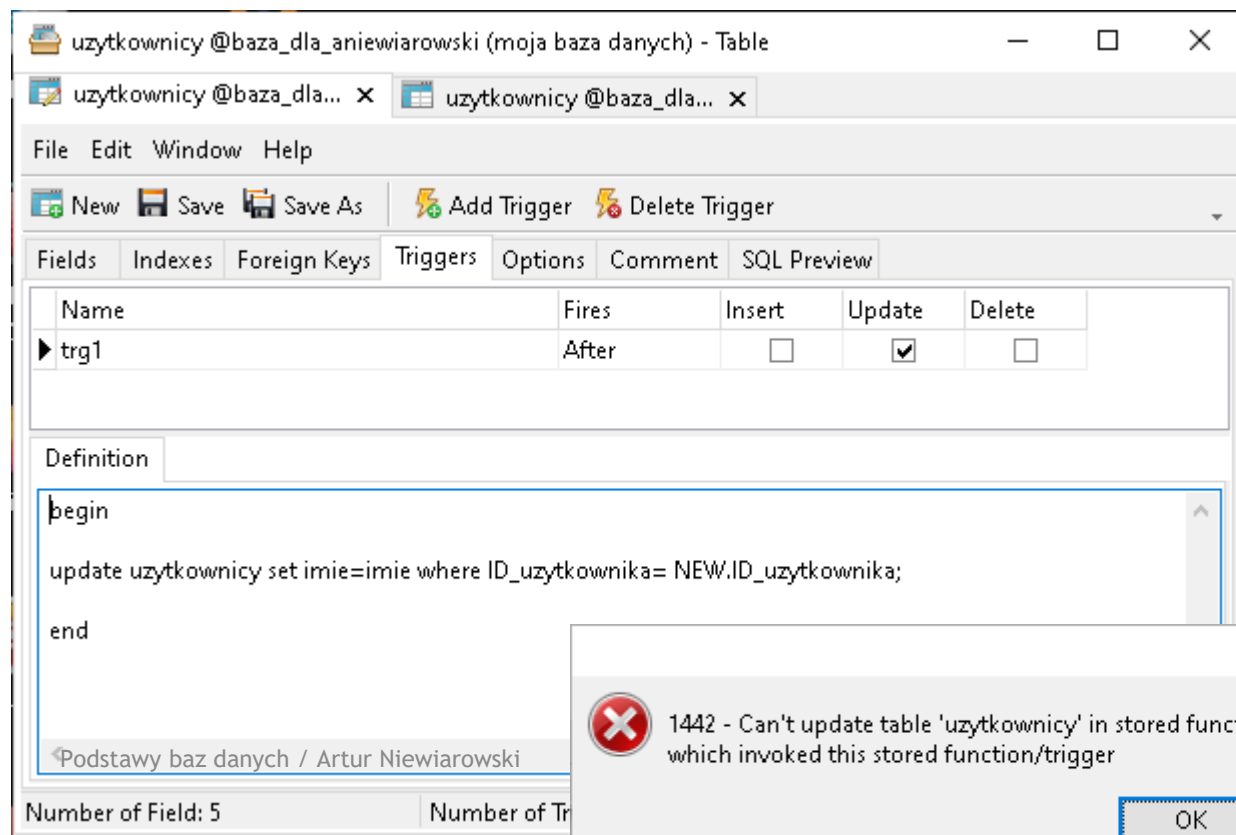
Podstawy baz danych / Artur Niewiarowski

```
1 row in set
```

Triggery - wyzwalacze

Często popełniane błędy podczas tworzenia/użytkowania triggerów

- utworzenie triggera z momentem aktywacji after i z poleceniem SQL, dla którego ten trigger jest wywoływany



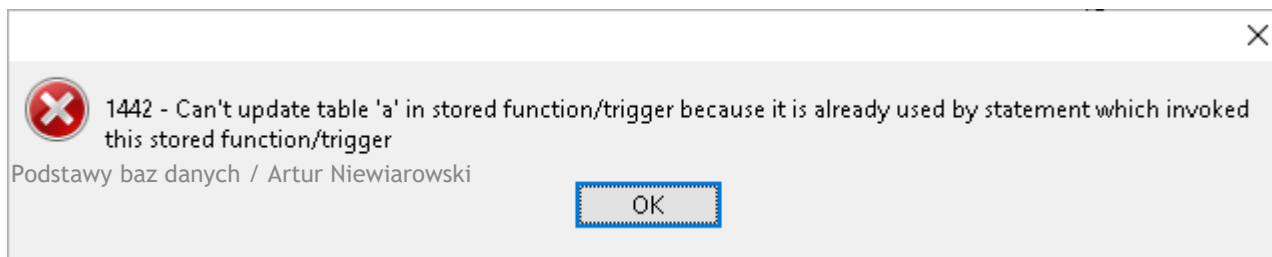
Triggery - wyzwalacze

Często popełniane błędy podczas tworzenia/użytkowania triggerów

- utworzenie triggera, który odwołuje się do tabeli, w której jest drugi trigger, który odwołuje się do tej pierwszej tabeli poleceniem, które aktywuje pierwszego triggera

```
CREATE TRIGGER `trg1` AFTER INSERT  
ON `A` FOR EACH ROW begin  
insert into B(v) values (NEW.v);  
end;
```

```
CREATE TRIGGER `trg2` AFTER INSERT  
ON `B` FOR EACH ROW begin  
insert into A(v) values (NEW.v);  
end;
```



Triggery - wyzwalacze

Często popełniane błędy podczas tworzenia/użytkowania triggerów

- błędne odczytanie źródła błędu z powodu triggera

```
CREATE TABLE `c` (  
  `v` int(11) unsigned DEFAULT NULL CHECK (`v` < 10)  
)
```

```
CREATE TRIGGER `trg4` AFTER INSERT ON `A` FOR EACH ROW  
begin  
insert into C(v) values (NEW.v);  
end;
```

Źródłem poniższych błędów jest tabela C, nie tabela A:

```
mysql> insert into A (v) values (-1);
```

```
1264 - Out of range value for column 'v' at row 1
```

```
mysql> insert into A (v) values (100);
```

```
4025 - CONSTRAINT `c.v` failed for `baza_dla_aniewiarowski`.`c`
```

Triggery - wyzwalacze

Wybrane ograniczenia dotyczące wyzwalaczy:

- Nie można wywołać poleceń: „alter table”, „load data”, „load table”
- Nie można wywołać poleceń: „lock tables”, „unlock tables”
- Polecenie „begin” rozpoczyna blok begin-end, więc jawne uruchomienie transakcji musi zostać wywołane poleceniem „start transaction”
- Nie można wywołać polecenia select zwracającego wyniki (wyjątkiem jest składnia: „select into”)
- Nie zwracają wartości (brak RETURN, zamiast tego LEAVE - opuszczenie wyzwalacza)
- Wyzwalacz nie jest aktywowany przez akcję na referencji FK-PK

Więcej informacji:

<https://mariadb.com/kb/en/trigger-limitations/>