

Triger / Wyzwalacz

Triger, zwany także wyzwalaczem jest to skrypt (fragment kodu) wykonywany w przypadku zajścia jakiegoś zdarzenia w bazie danych (np. dodania danych, ich modyfikacji, czy usunięcia). Istnieje kilka typów wyzwalaczy. Zadania dla BEFORE i AFTER.

Dla każdego typu istnieją trzy zdarzenia powodujące wykonanie wyzwalacza i są to:

- AFTER DELETE – wykonanie wyzwalacza po operacji usunięcia rekordu
- AFTER INSERT - wykonanie wyzwalacza po dodaniu rekordu
- AFTER UPDATE - wykonanie wyzwalacza po zmodyfikowaniu rekordu
- BEFORE DELETE - wykonanie wyzwalacza przed operacji usunięcia rekordu
- BEFORE INSERT - wykonanie wyzwalacza przed dodaniem rekordu
- BEFORE UPDATE - wykonanie wyzwalacza przed zmodyfikowaniem rekordu

Wyzwalacz uruchamiany jest zawsze po wykonaniu wywołującej go instrukcji. Jest jednak uruchamiany w obrębie tej samej transakcji, co wywołująca go instrukcja. Dzięki temu możemy wycofać efekty tej instrukcji poleceniem **ROLLBACK**.

Odwołania do starych i nowych wartości modyfikowanych danych są możliwe dzięki wirtualnym tabelom o nazwach **INSERTED** i **DELETED**, zawierających odpowiednio nowe i stare wartości. Aby się do nich odwołać, musimy użyć instrukcji **SELECT** tak, jakby były to normalne tabele bazy danych. W przypadku, gdy wyzwalacz został uruchomiony przez instrukcję **INSERT**, tylko tabela **INSERTED** będzie zawierała rekordy. Tabela **DELETED** będzie pusta. Gdy uruchamiającą wyzwalacz instrukcją jest **DELETE**, tylko **DELETED** zawiera rekordy. W przypadku **UPDATE** obie tabele zawierają rekordy: **INSERTED** zawiera nowe wartości (te, na które zmieniamy), a **DELETED** stare (nadpisane).

Tabele te są tylko do odczytu – modyfikacja danych bezpośrednio w nich nie jest możliwa. Jeśli chcemy wewnątrz wyzwalacza wprowadzić modyfikację danych, musimy wykonywać instrukcje na rzeczywistych tabelach. Dane w tych tabelach są danymi już zaktualizowanymi przez instrukcję uruchamiającą wyzwalacz.

Istniejące wyzwalacze mogą być włączane i wyłączane instrukcją:

```
ENABLE / DISABLE TRIGGER nazwa_wyzwalacza ON nazwa_tabeli
```

Budowa

```
DELIMITER || // OPCJONALNIE DELIMITER $$
CREATE TRIGGER nazwa_trigera
BEFORE INSERT ON // zdarzenie określające kiedy triger zostanie wyzwolony
nazwa_tabeli // tabela na której triger zostanie założony
FOR EACH ROW BEGIN
... // skrypt wykonywany przez triger
END || //OPCJONALNIE $$
```

Ćwiczenie.1 Utworzyć bazę zatrudnienie a w niej tabelę pracownik:

```
CREATE
TABLE pracownik
(
    id_pracownika INT(5) NOT NULL,
    imie CHAR(20) NOT NULL,
    nazwisko CHAR(50) NOT NULL,
    pensja FLOAT(6,2) NOT NULL,
    data_dodania DATETIME,
    data_mod DATETIME,
    pop_pensja FLOAT,
    PRIMARY KEY (id_pracownika)
);
```

Ćwiczenie.2 Utworzyć triggera który automatycznie uzupełni nam datę dodania rekordu do bazy.

```
DELIMITER ||
CREATE TRIGGER data_dodania
BEFORE INSERT ON pracownik
FOR EACH ROW BEGIN
SET NEW.data_dodania = now();
END ||
```

Opis: NEW.data_dodania oznacza nową wartość data_dodania. Trigger wywołany przed operacją INSERT spowoduje, że przy dodaniu rekordu do tabeli pracownik wartość pola data_dodania przyjmie aktualną (funkcja now()) datę + czas.

Ćwiczenie.3 Utworzyć trigger, który będzie uzupełniał datę ostatniej modyfikacji rekordu, wartość pola pop_pensja (poprzednia pensja) - w momencie kiedy pensja pracownika ulegnie zmianie.

```
DELIMITER ||
CREATE TRIGGER modyfikacja
BEFORE UPDATE ON pracownik
FOR EACH ROW BEGIN
SET NEW.data_mod = now();
IF NEW.pensja!=OLD.pensja THEN
SET NEW.pop_pensja = OLD.pensja;
END IF;
END ||
```

Opis: Wykorzystana została funkcja warunkowa IF która sprawdza czy nowa pensja (NEW.pensja) pracownika jest różna od poprzedniej (OLD.pensja). Jeżeli warunek jest spełniony, pole pop_pesja przyjmie wartość poprzedniej pensji, zaś pole pensja przyjmie wartość aktualną.

Ćwiczenie.4 Sprawdzić działanie triggera z Ćwiczenie.3

- INSERT INTO pracownik (id_pracownika, imie, nazwisko, pensja) VALUES (1, 'John', 'Smith', '2500');
- select * from pracownik;
- UPDATE pracownik SET pensja = 3400 WHERE id_pracownika=1;
- select * from pracownik;

Zad.1 Do bazy zatrudnienie stworzyć (kwerendą) tabelę stanowiska zawierającą pola:

- id_pracownika (klucz główny, autonumerowanie),
- stanowisko (text, 40 znaków),
- numer_pokoju (liczba naturalna)
- data_stanowisko (datetime)
- data_mod_stanowisko (datetime)

Zad.2 Uzupełnić (kwerendami) dane w tabelach pracownik i stanowiska - po 4 rekordy.

Zad.3 Stworzyć trigger, który sprawdzi kiedy dodano stanowisko.

Zad.4 Stworzyć trigger, który będzie sprawdzał kiedy zmodyfikowano stanowisko pracownika.

Zad.5 Stworzyć trigger, które sprawdzi kiedy usunięto pracownika.