

Oracle PL/SQL

Paweł Rajba

pawel@ii.uni.wroc.pl
<http://www.kursy24.eu/>

Zawartość modułu 6

- Wprowadzenie
- Definiowanie wyzwalaczy DML
- Metadane wyzwalaczy
- Inne zagadnienia, tabele mutujące
- Wyzwalacze INSTEAD OF
- Wyzwalacze dla zdarzeń systemowych i zdarzeń użytkownika
- Wyzwalacze i sekwencje

Wyzwalacze

- Co to jest wyzwalacz?
- Do czego możemy ich używać
 - Audyty operacji DDL i DML
 - Wymuszanie reguł poprawności danych
 - Wymuszanie zachowania integralności, których nie można uzyskać w sposób deklaratywny
 - Obsługa zdarzeń
 - Automatyczne generowanie treści,
 - np. auto increment
 - Obsługa zdarzeń systemowych

Wyzwalacze

- Jakie zdarzenia mogą być obsługiwane przez wyzwalacze
 - Operacje DDL (CREATE, ALTER, DROP)
 - Operacje DML (INSERT, UPDATE, DELETE)
 - Zdarzenia systemowe (STARTUP, SHUTDOWN)
 - Zdarzenia użytkownika (LOGON, LOGOFF)
- Nazwa wyzwalacza może być taka sama jak nazwa tabeli, ale nie jest to zalecane
 - Wyzwalacze są w innej przestrzeni nazw

Definiowanie wyzwalaczy DML

- Składnia
 - CREATE [OR REPLACE] TRIGGER nazwa
[BEFORE|AFTER] zdarzenie [OF kolumna]
ON tabela
[klauzula_referencing]
[WHEN warunek]
[FOR EACH ROW]
treść_wyzwalacza

Definiowanie wyzwalaczy DML

- Znaczenie opcji
 - zdarzenie to INSERT, UPDATE, DELETE lub ich kombinacja, np. INSERT OR UPDATE
 - OF kolumna określa, która z kolumn będzie modyfikowana (jeśli zdarzeniem będzie UPDATE)
 - WHEN warunek określa warunki, które muszą być spełnione, żeby wyzwalacz został uruchomiony
 - FOR EACH ROW określa tryb, który może być ROW lub STATEMENT
 - REFERENCING pozwala na niestandardowe nazwanie pseudowierszy OLD i NEW

Definiowanie wyzwalaczy DML

- Tryb ROW i STATEMENT
 - Wyzwalacz może reagować na poziomie wiersza lub na poziomie instrukcji
 - Domyślnie jest na poziomie instrukcji, wstawiając FOR EACH ROW – będzie na poziomie wiersza
 - ROW jest uruchamiany dla każdego wiersza, na którym operuje instrukcja DML, STATEMENT jest uruchamiany raz w odpowiedzi na instrukcję
 - np. w przypadku aktualizacji 10 wierszy ROW będzie uruchamiany 10 razy, STATEMENT – 1 raz

Definiowanie wyzwalaczy DML

- Tryb ROW i STATEMENT – różnice
 - R: wyzwalany dla każdego wiersza
 - S: wyzwalany raz w ramach danej instrukcji
 - w przypadku, gdy operacja DML nie oddziałuje na żaden wiersz: R – nie wykona się w ogóle, S: wykona się raz
 - R: jest dostęp do danych wierszy
 - S: nie ma dostępu do danych wierszy

- R: AFTER ROW blokują wiersze z danymi
- S: nie ma powyższej blokady

Definiowanie wyzwalaczy DML

- Klauzula REFERENCING
 - Domyślnie dostęp do wierszy w wyzwalaczu jest poprzez pseudo wiersze OLD i NEW
 - Można to zmienić używając klauzuli REFERENCING
 - Składnia
 - REFERENCING [OLD AS nazwa] [NEW AS nazwa]
 - Odwołanie jest poprzez :NEW i :OLD
 - Klauzula ma sens tylko w wyzwalaczach typu ROW
 - w przypadku statement klauzula spowoduje błąd kompilacji
 - new i old są dostępne w triggerach AFTER i BEFORE

Definiowanie wyzwalaczy DML

- Klauzula WHEN
 - Określa warunek na wiersze, dla których wyzwalacz zostanie uruchomiony
 - Składnia jest taka jak dla klauzuli WHERE
 - Używamy tylko w wyzwalaczach w trybie ROW
 - Odwołania do NEW i OLD bez dwukropków przed
 - Składnia
 - WHEN (warunek)

Definiowanie wyzwalaczy DML

- Wiele zdarzeń wyzwalających
 - Możemy utworzyć wyzwalacz, który reaguje na kilka zdarzeń (np. UPDATE i DELETE)
 - Żeby wewnątrz wyzwalacza odróżnić, które zdarzenie miało miejsce, mamy predykaty:
 - INSERTING, UPDATING, DELETING
- Przykład użycia
 - IF INSERTING THEN ...
ELSIF UPDATING THEN ...
ELSIF DELETING THEN ...
END IF;

Definiowanie wyzwalaczy DML

- Wiele wyzwalaczy na jednej tabeli
 - Można na danej tabeli robić kombinacje
 - BEFORE, AFTER
 - ROW, STATEMENT
 - INSERT, UPDATE, DELETE
co daje 12 różnych wyzwalaczy na jednej tabeli
 - A co jeśli zrobimy dwa takie wyzwalacze na jednej tabeli?
 - Po pierwsze możemy tak zrobić
 - Przykłady pokazują, że wyzwalacze będą uruchamiane w kolejności odwrotnej do tej, w której były tworzone

Metadane wyzwalaczy

- Mamy cztery perspektywy
 - USER_OBJECTS, ALL_OBJECTS
 - Info o wersji skompilowanej
 - USER_SOURCE, ALL_SOURCE
 - Kody źródłowe
 - W obu tabelach OBJECT_TYPE jest TRIGGER
- Odpalamy zapytania
 - `select object_name, object_type, status from user_objects where object_name=nazwa;`
 - `select text from user_source where name=nazwa;`

Inne zagadnienia

- Włączenia i wyłączenie
 - ALTER TRIGGER nazwa ENABLE|DISABLE [ALL TRIGGERS]
 - Kolumna user_triggers.status (all_triggers.status) pozwala ustalić, czy wyzwalacz jest włączony
- Usunięcie wyzwalacza
 - DROP TRIGGER nazwa;
 - usunięcie tabeli powoduje automatyczne usunięcie powiązanych z nim wyzwalaczy
- Przykład: wyzwalacze-dml.txt

Tabele mutujące

- Co to jest tabela mutująca?
- Kiedy tabela jest mutująca
 - Jest modyfikowana przez instrukcję DML lub wyzwalacz
 - Może być modyfikowana przez zależność klucza obcego (np. cascade)
- Dotyczy głównie wyzwalaczy typu ROW
 - W niektórych przypadkach także typu STATEMENT
- Przykład: `wyzwalacze-mutujace.txt`

Wyzwalacze INSTEAD OF

- Perspektywa może być modyfikowana, gdy w jej zapytaniu definiującym nie ma:
 - złączeń
 - operatorów SET
 - funkcji agregujących
 - klauzul GROUP BY, CONNECT BY i START WITH
 - operatora DISTINCT
 - pseudokolumn i wyrażeń
- W pozostałych przypadkach przydatne są wyzwalacze typu INSTEAD OF

Definiowanie wyzwalaczy INSTEAD OF

- Składnia
 - CREATE [OR REPLACE] TRIGGER nazwa INSTEAD OF zdarzenia [OF kolumna] ON widok [klauzula_referencing] [FOR EACH ROW] treść_wyzwalacza
- Kilka uwag
 - Tego typu wyzwalacze są zawsze typu ROW
 - Nie ma AFTER, BEFORE, WHEN
 - Znaczenie pozostałych parametrów bez zmian
- Przykład: wyzwalacze-insteadof.txt

Wyzwalacze systemowe i użytkownika

- W Oracle'u możemy także tworzyć wyzwalacze na zdarzenia systemowe
 - Uruchomienie i zatrzymanie bazy danych
 - Logowanie do i wylogowanie z bazy danych
 - zwykle aplikacji klienckiej
 - Błędy po stronie serwera
- oraz na zdarzenia użytkownika
 - Początek i koniec sesji
 - Operacje języka DDL

Wyzwalacze systemowe i użytkownika

- Definiowanie takich wyzwalaczy
 - CREATE [OR REPLACE] TRIGGER nazwa [BEFORE|AFTER] zdarzenie ON {DATABASE|[schemat.]SCHEMA} [WHEN warunki] treść_wyzwalacza
- Wyzwalacze takie tworzymy albo na poziomie bazy danych, albo na poziomie schematu

Wyzwalacze systemowe i użytkownika

- Zdarzenia systemowe mogą być:
 - STARTUP (AFTER), SHUTDOWN (BEFORE), SERVERERROR (AFTER), LOGON (AFTER), LOGOUT (BEFORE)
- Zdarzenia użytkownika mogą być:
 - LOGON (AFTER), LOGOFF (BEFORE), SERVERERROR (AFTER), CREATE (BEFORE, AFTER), ALTER (BEFORE, AFTER), DROP (BEFORE, AFTER)

Wyzwalacze systemowe i użytkownika

- Atrybuty na poziomie bazy danych
 - `sys.sysevent`, `varchar(20)` – zdarzenie systemowe wyzwalające trigger
 - `sys.instance_num`, `number` – numer instancji
 - `sys.database_name`, `varchar2(50)` – nazwa bazy d.
 - `sys.server_error`, `number` – zwraca numer błędu
 - `sys.is_servererror`, `boolean` – czy błąd jest błędem serwera
 - `sys.login_user`, `varchar2(30)` – nazwa użytkownika

Wyzwalacze systemowe i użytkownika

- Na poziomie użytkownika, oprócz atrybutów systemowych mamy następujące atrybuty:
 - `sys.dictionary_obj_type`, `varchar2(20)` – typ obiektu, na którym wykonano operację DDL
 - `sys.dictionary_obj_name`, `varchar2(30)` – nazwa obiektu, na którym wykonano operację DDL
 - `sys.dictionary_obj_owner`, `varchar2(30)` – właściciel obiektu, na którym wykonano operację DDL

Wyzwalacze systemowe i użytkownika

- Atrybuty zdarzeń użytkownika
 - CREATE, ALTER, DROP (BEFORE, AFTER)
 - sys.sysevent, sys.login_user, sys.instance_num, sys.database_name, sys.dictionary_obj_type, sys.dictionary_obj_name, sys.dictionary_obj_owner
 - LOGON, LOGOFF
 - sys.sysevent, sys.login_user, sys.instance_num, sys.database_name

Wyzwalacze systemowe i użytkownika

- Atrybuty zdarzeń systemowych
 - STARTUP, SHUTDOWN
 - sys.sysevent, sys.login_user, sys.instance_num, sys.database_name
 - SERVERERROR
 - sys.sysevent, sys.login_user, sys.instance_num, sys.database_name, sys.server_error, sys.is_servererror
- Przykład:
 - wyzwalacze-systemowe-uzytkownika.txt

Wyzwalacze i sekwencje

- Co to są sekwencje?
- Tworzenie sekwencji
 - CREATE SEQUENCE nazwa
[INCREMENT BY liczba]
[START WITH liczba]
[MAXVALUE liczba|NOMAXVALUE]
[MINVALUE liczba|NOMINVALUE]
[CYCLE|NOCYCLE]
[CACHE liczba|NOCACHE]
[ORDER|NOORDER]

Wyzwalacze i sekwencje

- Tworzenie sekwencji, znaczenie opcji
 - INCREMENT BY liczba – określa skok
 - START WITH liczba – wartość początkowa
 - MAXVALUE liczba – górne ograniczenie
 - NOMAXVALUE – brak ograniczenia (domyślnie)
 - MINVALUE liczba – dolne ograniczenie
 - NOMINVALUE – brak ograniczenia (domyślnie)
 - CYCLE – po osiągnięciu ograniczenia, generowanie będzie od początku
 - NOCYCLE – przeciwne do CYCLE (domyślnie)

Wyzwalacze i sekwencje

- Tworzenie sekwencji, znaczenie opcji c.d.
 - CACHE liczba – ile następnych liczb sekwencji ma być przygotowana w pamięci, domyślnie jest 20
 - NOCACHE – liczby będą generowane on-line
- Nie podając żadnych opcji stworzymy sekwencję, która
 - zaczyna się o 1
 - skok jest o 1
 - nie ma ograniczenia górnego
 - nie ma generowania cyklicznego

Wyzwalacze i sekwencje

- Modyfikowanie sekwencji
 - ALTER SEQUENCE nazwa [opcje_takie_jak_przy_tworzeniu]
 - Jeśli chcemy zrestartować sekwencję od innej liczby, trzeba go usunąć i utworzyć ponownie
- Usunięcie sekwencji
 - DROP SEQUENCE nazwa
- Metody sekwencji
 - CURRVAL – pobranie bieżącej wartości
 - NEXTVAL – pobranie następnej wartości

Wyzwalacze i sekwencje

- Przykład: wyzwalacze-sekwencje.txt