Wykład V

Wyzwalacz

- procedura wyzwalana, składowana fizycznie w bazie, uruchamiana automatycznie po nastąpieniu określonego w definicji zdarzenia

Składowe wyzwalacza (ECA):

- określenie zdarzenia (Event)
- określenie warunku (Condiion)
- określenie akcji (Action)

Zastosowanie wyzwalaczy:

- automatyczne generowanie wartości
- zapewnienie niestandardowych więzów integralności, realizacja złożonej logiki biznesowej
- prowadzenie statystyk, monitorowanie działań użytkowników
- zapewnienie dodatkowych reguł bezpieczeństwa
- modyfikacja danych poprzez złożone perspektywy

Typy wyzwalaczy:

- DML – reagujące na polecenia INSERT, UPDATE, DELETE
- zastępujące – zamiast wykonania instrukcji (INSTEAD OF)
- systemowe -uruchamiane w wyniku zajścia zdarzenia systemowego
- przed wykonaniem instrukcji (BEFORE)
- po wykonaniu instrukcji (AFTER)

składnia:

CREATE OR REPLACE TRIGGER [< *schemat.* **>]** < **nazwa** >
{ BEFORE | AFTER | INSTEAD OF }
{ < *zdarzenie_dml* > [**OR** < *zdarzenie_DML*>] . . .
| { < *zdarzenie_ddl* > [**OR** < *zdarzenie_DDL* >] . .
| < *zdarzenie_systemowe* > [OR < *zdarzenie_systemowe* >] . . . }
ON { [<*schemat* . > **SCHEMA** | **DATABASE** } }
[WHEN (< *warunek_logiczny* >) **]**
{ < *blok_PL/SQL* > | < *wywołanie_procedury* >}

Dostępne opcje składni zależą od typu wyzwalacza -
Opis zdarzenia DML

{ **DELETE | INSERT | UPDATE [OR** < *nazwa_kol1* > [, < *nazwa_kol2* >] . . . }
[OR { DELETE | INSERT | UPDATE [OF < *nazwa_kol1* [, < *nazwa_kol2* >] . . .] }] . . .
ON { [< *schemat.* > < *nazwa_tabeli* > | [< *schemat.* > < *nazwa_perspektywy* > }
[FOR EACH ROW]

Użycie klauzuli FOR EACH ROW oznacza, typ wyzwalacza na poziomie wierszy;
wykonują się raz dla każdego wiersza, którego dotyczy polecenie DML
W innym przypadku jest to wyzwalacz poziomu instrukcji

Wykład V

przykłady:

```
CREATE OR REPLACE TRIGGER zmiana_premii
AFTER UPDATE OF premia ON Pracownicy
BEGIN
  INSERT INTO Dziennik VALUES
       (' Pracownicy', Current_date, User, ' zmiana premii' );
END;
```

W wyzwalaczu można uzyskać dostęp do danych (atrybutów) przetwarzanego wiersza.
*Służą do tego dwa identyfikatory korelacji - **:old**, **:new**.*
Identyfikator korelacji to specjalny rodzaj zmiennych powiązanych z tabelą wyzwalającą.
Kompilator PL/SQL traktuje je jako rekordy typu
tabela_wyzwalająca%ROWTYPE

- identyfikator **:old** jest niezdefiniowany dla polecenia **INSERT**
- identyfikator **:new** dla polecenia **DELETE**.

Kompilator PL/SQL nie wygeneruje błędu, jednak wartością pól w obu przypadkach będzie NULL

przykłady:

```
CREATE OR REPLACE TRIGGER Poprawa
BEFORE INSERT ON Etaty
FOR EACH ROW
BEGIN
:new.nazwa:= UPPER(:new.nazwa);
END;

CREATE OR REPLACE TRIGGER Podwyzki
BEFORE UPDATE OF placa ON Pracownicy
FOR EACH ROW
WHEN (new.etat NOT LIKE 'Dyrektor')
BEGIN
 IF :new.placa > :old.placa * 1.2 THEN :new.placa := :old.placa * 1.2;
 END IF;
 END;


CREATE OR REPLACE TRIGGER Kontrola_plac
BEFORE INSERT ON Pracownicy
FOR EACH ROW
DECLARE
V_min NUMBER;
V_max NUMBER;
BEGIN
 SELECT placa_min, placa_max INTO V_min, V_max FROM Etaty
   WHERE nazwa=:new.etat;
  IF :new.placa NOT BETWEEN V_min AND V_max THEN
  RAISE_APPLICATION_ERROR( -20001,' Placa niezgodna z etatem');
  END IF;
END;
CREATE OR REPLACE TRIGGER LIMIT_prac
```

```
BEFORE INSERT OR UPDATE OF Id_zesp  ON Pracownicy
FOR EACH ROW
DECLARE
V_liczba NUMBER;
BEGIN
  SELECT Count(*) INTO V_liczba  FROM  Pracownicy WHERE Id_zesp=:new.id_zesp;
  IF V_liczba >= 5  THEN
  RAISE_APPLICATION_ERROR(-20025,' Za liczny zespól');
  END IF;
END;

UPDATE Pracownicy  SET  Id_zesp = 20;
```

SQL Error: ORA-04091: tabela . . ..PRACOWNICY ulega mutacji, wyzwalacz/funkcja może tego nie widzieć
ORA-06512: przy ". . ..LIMIT_PRAC", linia 4

```
CREATE  OR  REPLACE  TRIGGER Kontrola_egzamin
AFTER  INSERT OR  DELETE  OR  UPDATE ON Egzamin
FOR  EACH  ROW
BEGIN
  IF INSERTING THEN
    INSERT INTO Audyt VALUES ( Current_date, 'Wstawiono dane do tabeli Egzamin');
  END  IF;
  IF  DELETING THEN
    INSERT INTO Audyt VALUES ( Current_date,'Kasowano dane w tabeli Egzamin');
  END  IF;
  IF  UPDATING THEN
    INSERT INTO Audyt  VALUES ( Current_date,'Poprawiano dane w tabeli Egzamin');
  END  IF;
END;
```

Wyzwalacze zastępujące INSTEAD OF

- stosowane wyłącznie dla perspektyw
- wykonywane zamiast instrukcji aktywującej wyzwalacz
- ograniczenia typu CHECK nie są sprawdzane
- zawsze typu wierszowego

```
CREATE  OR  REPLACE  VIEW Zespoly_pracownicze
AS
SELECT  nazwa, count(numer) liczba_prac, sum(placa+nvl(dodatek,0))  budzet FROM
Pracownicy RIGHT JOIN Zespoly USING( id_zesp)  GROUP  BY  nazwa;

CREATE  OR  REPLACE  TRIGGER Mod_Zespoly_pracownicze
INSTEAD OF INSERT  ON  Zespoly_pracownicze
BEGIN
   INSERT  INTO Zespoly VALUES (Numeruj.nexval, :new.nazwa, null);
END;
```

Wykład V

Wyzwalacze systemowe są aktywowane przez
- zdarzenia zmiany stanu systemu
- zdarzenia wywołane pracą użytkowników

mogą być wywołane dla danego schematu (jednego użytkownika),
bądź dla całej bazy danych (dla wszystkich użytkowników)

Zdarzenia **DDL** i bazy danych
BEFORE | AFTER ALTER – przed/ po zmianie definicji obiektu
BEFORE | AFTER DROP – przed / po usunięciu obiektu
BEFORE | AFTER CREATE -przed / po utworzeniu obiektu
BEFORE | AFTER GRANT – przed / po nadaniu uprawnień
BEFORE | AFTER REVOKE – przed / po odebraniu uprawnień
AFTER LOGON – po zalogowaniu
BEFORE LOGOFF – przed wylogowaniem
AFTER STARTUP – po otwarciu bazy danych
AFTER SERVERERROR – po wystąpieniu błędu
BEFORE SHUTDOWN – przed zatrzymaniem instancji bazy

```
CREATE OR REPLACE TRIGGER Blokada_drop
BEFORE DROP ON SCHEMA
BEGIN
   RAISE_APPLICATION_ERROR
              ( -20123,' próba usuwania obiektów przez ' || user|| ' numer ' || uid );
END;
```

```
CREATE OR REPLACE TRIGGER Log_errors
AFTER SERVERERROR ON DATABASE
BEGIN
  IF ( IS_SERVERERROR(1017) THEN
    INSERT INTO sever_audit( Current_timestamp, 'Error – 1017');
 END IF;
END;
```

Ograniczenia stosowania wyzwalaczy

- nie mogą wykonywać instrukcji DDL
- nie mogą wykonywać instrukcji kontrolnych COMMIT, ROLLBACK itp.
- wyzwalacze wierszowe nie mogą modyfikować ani wykonywać zapytania do obiektu, dla którego zostały wywołane
- w ciele wyzwalacza nie można deklarować zmiennych typu LONG ani LONG RAW
- wyzwalacze systemowe STARTUP i SHUTDOWN nie mogą mieć żadnych warunków

Inne polecenia PL/SQL związane z wyzwalaczami

ALTER TRIGGER < *nazwa* > **{ DISABLE | ENABLE};**
DROP TRIGGER < *nazwa* >;

Wykład V

Dynamiczny SQL – technika programowania umożliwiająca generowanie instrukcji SQL dynamicznie w trakcie wykonywania kodu źródłowego

- instrukcja taka jest zapisana w postaci łańcucha znaków
- wykonanie wewnątrz bloku PL/SQL umożliwia polecenie **EXECUTE IMMEDIATE**

przyklady:

```
CREATE OR REPLACE PROCEDURE Dowolne_polecenie( p_text VARCHAR2 )
AS
BEGIN
  EXECUTE IMMEDIATE( p_text );
END;

EXEC Dowolne_polecenie('CREATE TABLE Opis( kol VARCHAR2(30))');
```

ORA-01031: niewystarczające uprawnienia
ORA-06512: przy "STUD_UZUP.DOWOLNE_POLECENIE", linia 4

```
EXEC Dowolne_polecenie('INSERT INTO Egzamin VALUES (12348,95)');

EXEC Dowolne_polecenie('DROP TABLE Robo');

CREATE OR REPLACE PROCEDURE Kasuj_tabele ( p_nazwa VARCHAR2)
AS
BEGIN
  EXECUTE IMMEDIATE ' DROP TABLE '||p_nazwa;
END;


EXEC Kasuj_tabele( 'AUDYT')
```

Dynamiczny SQL- parametryzacja

```
CREATE OR REPLACE PROCEDURE Dodaj_zespol
                        (p_nazwa VARCHAR2, p_adres VARCHAR2 )
AS
V_id zespoly.id_zesp%TYPE;
BEGIN
  EXECUTE IMMEDIATE 'SELECT Max(id_zesp) FROM zespoly' INTO V_id  ;
  EXECUTE IMMEDIATE 'INSERT INTO Zespoly VALUES (:1, :2, :3) '
    USING V_id + 10, p_nazwa, p_adres;
  COMMIT;
END Dodaj_zespol;
```

Wykorzystano
Wykłady dr inż. Olga Siedlecka-Lamch – Systemy baz danych z roku 2012