

Declanșatori compuși în Oracle 11g

Un trigger compus reprezintă patru declanșatori diferiți definiți ca fiind unul singur. Nu numai că se economisește foarte mult cod de scris, dar, de asemenea, rezolvă problema în cazul în care același set de variabile de sesiune împart aceleași date.

La ce sunt folosiți?

Cele mai importante două motive pentru a utiliza declanșatorul compus sunt: acumularea rândurilor pentru inserarea în bloc și evitarea erorii de tabel mutant.

Declanșatoarele compuse pot avea până la patru secțiuni:

1. Secțiunea BEFORE
2. Secțiunea BEFORE EACH ROW
3. Secțiunea AFTER EACH ROW
4. Secțiunea AFTER

Sintaxa generală:

```
CREATE OR REPLACE TRIGGER compound_trigger_name
FOR [INSERT|DELETE]UPDATE [OF column] ON table
COMPOUND TRIGGER
  -- Sectiune declarativa optionala

  --Se executa inaintea instructiunilor LMD
  BEFORE STATEMENT IS
  BEGIN
    NULL;
  END BEFORE STATEMENT;

  --Se executa inaintea schimbarii fiecarei linii
  -- :NEW, :OLD valabile
  BEFORE EACH ROW IS
  BEGIN
    NULL;
  END BEFORE EACH ROW;
```

```
--Se executa dupa fiecare schimbare de linie
-- :NEW, :OLD valabile
AFTER EACH ROW IS
BEGIN
    NULL;
END AFTER EACH ROW;
```

--Se executa dupa instructiunile LMD

```
AFTER STATEMENT IS
BEGIN
    NULL;
END AFTER STATEMENT;
```

```
END compound_trigger_name;
```

Restricții:

- Un declanșator compus trebuie să fie un declanșator LMD.
- Un declanșator compus trebuie definit fie pe o masă, fie pe o vizualizare.
- Partea declarativă nu poate include PRAGMA AUTONOMOUS_TRANSACTION.
- Un corp declanșator compus nu poate avea un bloc de inițializare; prin urmare, nu poate avea o secțiune de excepție
- O excepție care apare într-o secțiune trebuie tratată în acea secțiune. Nu poate transfera controlul către o altă secțiune.
- Dacă o secțiune include o instrucțiune GOTO, ținta declarației GOTO trebuie să fie în aceeași secțiune.
- Numai secțiunea BEFORE EACH ROW poate modifica valoarea: NEW.

Exemplu tabela copie emp_odi a tabeli employees de la laborator:

Vreau să updatez comisionul angajaților din departamentul cu id-ul care începe cu litera ,s' , afișând cel mai mare comision înainte de updată, după fiecare linie pe cel nou și pe cel vechi și cel mai mare comision după updată.

```
CREATE OR REPLACE TRIGGER exemplu
FOR UPDATE ON emp_odi
COMPOUND TRIGGER
```

```
-- Global declaration.
```

```
TYPE t_tab IS TABLE OF VARCHAR2(500);
```

```
l_tab t_tab := t_tab();
```

```
BEFORE STATEMENT IS
```

```
maxim emp_odi.commission_pct%type;
```

```
BEGIN
```

```
l_tab.extend;
```

```
--afisez cel mai mare comision inainte de marire
```

```
select max(commission_pct)
```

```
into maxim
```

```
from emp_odi;
```

```
l_tab(l_tab.last) := 'Cel mai mare comision BEFORE UPDATE este :| | maxim;
```

```
END BEFORE STATEMENT;
```

```
BEFORE EACH ROW IS
```

```
BEGIN
```

```
l_tab.extend;
```

```
l_tab(l_tab.last) := 'BEFORE EACH ROW:| |'
```

```
noul comision= ' | | :new.commission_pct | | ' vechiul comision= ' | |
```

```
nvl(:old.commission_pct,0);
```

```
END BEFORE EACH ROW;
```

```
AFTER EACH ROW IS
```

```
BEGIN
```

```
l_tab.extend;
```

```
l_tab(l_tab.last) := 'AFTER EACH ROW:| |'noul comision=' | |
```

```
:new.commission_pct | | ' vechiul comision=' | | nvL(:old.commission_pct,0);
```

```
END AFTER EACH ROW;
```

```
AFTER STATEMENT IS
```

```
    maxim emp_odi.commission_pct%type;
```

```
BEGIN
```

```
    l_tab.extend;
```

```
    --afisez cel mai mare comision dupa update
```

```
    select max(commission_pct)
```

```
    into maxim
```

```
    from emp_odi;
```

```
    l_tab(l_tab.last) := 'Cel mai mare comision AFTER UPDATE este :'| | maxim;
```

```
FOR i IN l_tab.first .. l_tab.last LOOP
```

```
    DBMS_OUTPUT.put_line(l_tab(i));
```

```
END LOOP;
```

```
    l_tab.delete;
```

```
END AFTER STATEMENT;
```

```
END exemplu;
```

```
/
```

```
select * from emp_odi;
```

```
update emp_odi
```

```
set commission_pct=0.9
```

```
where lower(job_id) like's%';
```

```
select * from emp_odi
```

```
where lower(job_id) like's%';
```

```
rollback;
```

```
drop trigger exemplu;
```

The image displays two side-by-side screenshots of the 'Dbms Output' window from an Oracle database client. Both windows show the output of a PL/SQL program named 'gr241'. The top-left window shows the first part of the execution, starting with 'Cel mai mare comision BEFORE UPDATE este :.4' and followed by several iterations of 'BEFORE EACH ROW:' and 'AFTER EACH ROW:noul comision=.9 vechiul comision=0'. The top-right window shows the continuation of the same execution, ending with 'Cel mai mare comision AFTER UPDATE este :.9'. Both windows have a 'Buffer Size: 20000' setting.