

## Okidači za potrebe dinamičkog referencijalnog integriteta

Ukoliko DBMS ne podržava deklarativni dinamički referencijalni integritet, onda se isti efekat može postići upotrebom okidača.

Okidač za DELETE nad „parent“ tabelom može poslužiti za RESTRICT, CASCADE, SET NULL i ~~SET-DEFAULT~~ pravila.

Okidač za UPDATE nad „parent“ tabelom može poslužiti za RESTRICT, CASCADE, SET NULL i ~~SET-DEFAULT~~ pravila.

Okidač za INSERT nad „child“ tabelom može poslužiti za RESTRICT pravilo.

Okidač za UPDATE nad „child“ tabelom može poslužiti za RESTRICT pravilo.

Okidač za DELETE nad „child“ tabelom može poslužiti za reverzni referencijalni integritet (tzv. „pendant DELETE“), odnosno pravilo kojim se postiže brisanje reda u „parent“ tabeli nakon što se obriše poslednji strani ključ koji referiše taj red. Ovo pravilo nije podržano deklarativnim referencijalnim integritetom.

Preporučljivo je koristiti deklarativni referencijalni integritet kad god je to moguće.

Okidače treba koristiti:

- a. kada podaci koji se brišu, dodaju ili menjaju moraju biti upotrebljeni da bi se odredilo koju kaciju je potrebno preduzeti.
- b. kada određeno pravilo referencijalnog integriteta nije podržano u DBMS (npr. Neki DBMS podržava samo RESTRICT za DELETE i UPDATE, a neki samo ne podržavaju CASCADE za UPDATE).
- c. kada je potrebno implementirati reverzni referencijalni integritet

Primer okidača za impelmentaciju CASCADE prilikom DELETE iskaza:

```
CREATE TRIGGER Brisanje_naslova
AFTER DELETE ON Naslov
REFERENCING OLD AS stari
FOR EACH ROW
BEGIN
DELETE FROM Autor
WHERE Autor.id_naslova = stari.id_naslova
END;
```

Ili:

```
CREATE TRIGGER Brisanje_naslova
AFTER DELETE ON Naslov
REFERENCING OLD AS stari
FOR EACH STATEMENT
BEGIN
DELETE FROM Autor
WHERE Autor.id_naslova IN (SELECT id_naslova FROM stari)
END;
```

Primer okidača za impelmentaciju RESTRICT prilikom INSERT iskaza:

```
CREATE TRIGGER Dodavanje_naslova
AFTER INSERT ON Autor
REFERENCING NEW AS novi
FOR EACH ROW
BEGIN
IF 1 > ALL (SELECT COUNT(*)
            FROM Naslov
            WHERE id_naslova = novi.id_naslova)
THEN ROLLBACK;
ELSE COMMIT;
END IF;
END;
```

Uvek treba koristiti deklarativni referencijalni interitet kad god je to moguće, jer upotreba okidača zahteva dosta pažnje i detaljnu proveru.

Primer okidača za impelmentaciju SET NULL prilikom UPDATE iskaza:

```
CREATE TRIGGER Dodavanje_naslova
AFTER UPDATE OF id_naslova ON Naslov
REFERENCING NEW AS novi
                OLD AS stari
FOR EACH ROW
BEGIN
IF 0 = ALL (SELECT COUNT(*)
            FROM Naslov
            WHERE id_naslova = stari.id_naslova)
THEN
    UPDATE Autor
    SET id_naslova = NULL
    WHERE id_naslova = stari.id_naslova
END IF;
END;
```

Provera je obavezna kako bi se obezbedilo da je UPDATE iskaz zaista promenio vrednost posmatrane kolone, jer u suprotnom, ako promene nije bilo, nije potrebno izvršiti neutralizaciju (postaviti na NULL).