

## Okidači za potrebe ograničenja

Upotrebom okidača može se nadoknaditi nedostatak podrške za očuvanje opšteg integriteta (CREATE ASSERTION).

Na primer:

```
CREATE TABLE Prodavnice
(id_prodavnice INTEGER NOT NULL PRIMARY KEY,
 naziv_prodavnice CHAR(35) NOT NULL,
 ..);

CREATE TABLE Radnici
(id_radnika CHAR(9) NOT NULL PRIMARY KEY,
 ime CHAR(15) NOT NULL,
 prezime CHAR(15) NOT NULL,
 ..);

CREATE TABLE DodelaPosla
(id_prodavnice INTEGER NOT NULL REFERENCES Prodavnice(id_prodavnice)
                                ON UPDATE CASCADE
                                ON DELETE CASCADE,
 id_radnika CHAR(9) NOT NULL PRIMARY KEY REFERENCES Radnici(id_radnika)
                                ON UPDATE CASCADE
                                ON DELETE CASCADE,
 datum_pocetka TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
 datum_kraja TIMESTAMP,
 CHECK (datum_pocetka <= datum_kraja),
 tip_posla INTEGER DEFAULT 0 NOT NULL CHECK (tip_posla BETWEEN 0 AND 99) -- menadzer = 99
);
```

Potrebno je podržati opšti intergitet definisan u vidu ogrničenja, da svaka prodavnica ima najviše jednog menadžera:

```
CREATE ASSERTION ProveraMenadzera
CHECK (1 => ALL (SELECT COUNT(*)
                FROM DodelaPosla
                WHERE tip_posla = 99
                GROUP BY id_prodavnice));
```

### Rešenje 1: U nedostatku podrške za ASSERTION, kreiran je okidač:

```
CREATE TRIGGER ProveraManadzera
AFTER UPDATE ON DodelaPosla -- potreban je istovetan okidač i za INSERT
IF 1 < ANY (SELECT COUNT(*)
            FROM DodelaPosla
            WHERE tip_posla = 99
            GROUP BY id_prodavnice)
THEN ROLLBACK;
ELSE COMMIT;
END IF;
```

### Rešenje 2: U nedostatku podrške za ASSERTION, kreirane su dve disjunktne tabele, potom pogled koji ih objedinjuje, kao i odgovarajući okidač nad pogledom:

```
CREATE TABLE Dodela_99_Posla
(id_prodavnice INTEGER NOT NULL PRIMARY KEY REFERENCES Prodavnice(id_prodavnice)
                                ON UPDATE CASCADE
                                ON DELETE CASCADE,
id_radnika CHAR(9) NOT NULL REFERENCES Radnici(id_radnika)
                                ON UPDATE CASCADE
                                ON DELETE CASCADE,
datum_pocetka TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
datum_kraja TIMESTAMP,
CHECK (datum_pocetka <= datum_kraja),
tip_posla INTEGER DEFAULT 99 NOT NULL CHECK (tip_posla = 99)
);
```

```
CREATE TABLE Dodela_ne99_Posla
(id_prodavnice INTEGER NOT NULL REFERENCES Prodavnice(id_prodavnice)
                                ON UPDATE CASCADE
                                ON DELETE CASCADE,
id_radnika CHAR(9) NOT NULL PRIMARY KEY REFERENCES Radnici(id_radnika)
                                ON UPDATE CASCADE
                                ON DELETE CASCADE,
datum_pocetka TIMESTAMP DEFAULT CURRENT_TIMESTAMP NOT NULL,
datum_kraja TIMESTAMP,
CHECK (datum_pocetka <= datum_kraja),
tip_posla INTEGER DEFAULT 0 NOT NULL CHECK (tip_posla BETWEEN 0 AND 98) - nema tipa 99
);
```

```
CREATE VIEW DodelaPosla (id_prodavnice, id_radnika, datum_pocetka, datum_kraja, tip_posla)
AS
(SELECT id_prodavnice, id_radnika, datum_pocetka, datum_kraja, tip_posla
 FROM Dodela_ne99_Posla
 UNION ALL
 SELECT id_prodavnice, id_radnika, datum_pocetka, datum_kraja, tip_posla
 FROM Dodela_99_Posla
);
```

```
CREATE TRIGGER NovaDodelaPosla
INSTEAD OF INSERT ON DodelaPosla
REFERENCING NEW AS DP
BEGIN
IF DP.tip_posla <> 99
THEN INSERT INTO Dodela_ne99_Posla
VALUES (DP.id_prodavnice, DP.id_radnika, DP.datum_pocetka, DP.datum_kraja, DP.tip_posla);
ELSE INSERT INTO Dodela_99_Posla
VALUES (DP.id_prodavnice, DP.id_radnika, DP.datum_pocetka, DP.datum_kraja, DP.tip_posla);
END IF;
END;
```

```
CREATE TRIGGER NovaDodelaPosla
INSTEAD OF DELETE ON DodelaPosla
REFERENCING OLD AS DP
BEGIN
IF DP.tip_posla <> 99
THEN DELETE FROM Dodela_ne99_Posla
WHERE id_radnika = DP.id_radnika;
ELSE DELETE FROM Dodela_99_Posla
WHERE id_radnika = DP.id_radnika;
END IF;
END;
```