

Zadatak 1:

Za primer Agencije za drumski prevoz tereta potrebno je:

a) Sastaviti jedan ili više SQL okidača koji u tabeli VOZNJA održavaju redundantne podatke o nazivima mesta (NazivOd, NazivDo) i imenu vozača (Ime) u trenutku izmene podatka o statusu vožnje (kada status dobije vrednost Z).

```
CREATE TRIGGER DetaljiVoznja
AFTER UPDATE OF Status ON VOZNJA
REFERENCING NEW ROW AS Novi
FOR EACH ROW
WHEN Novi.Status = 'Z'
BEGIN
    DECLARE
        vIDMesOd ...,
        vIDMesDo ...,
        vNazivOd ...,
        vNazivDo ...,
        vlme ...

    SELECT IDMesOd, IDMesDo
    INTO vIDMesOd, vIDMesDo
    FROM LINIJA
    WHERE IDLin = Novi.IDLin;

    SELECT Naziv
    INTO vIDNazivOd
    FROM MESTO
    WHERE IDMes = vIDMesOd;

    SELECT Naziv
    INTO vIDNazivDo
    FROM MESTO
    WHERE IDMes = vIDMesDo;

    SELECT Ime
    INTO vlme
    FROM VOZAC
    WHERE IDVoz = Novi.IDVoz;

    UPDATE VOZNJA
    SET NazivOd = vNazivOd, NazivDo = vNazivDo, Ime = vlme
    WHERE IDVzn = Novi.IDVzn ;
END
```

b) Sastaviti jedan ili više SQL okidača koji u tabeli KAMION održavaju redundantne podatke o ukupnom broju vožnji i ukupnoj pređenoj kilometraži (UkupnoVoznji, UkupnoKm) u trenutku izmene podatka o statusu vožnje (kada status dobije vrednost Z).

```
CREATE TRIGGER DetaljiKamion
AFTER UPDATE OF Status ON VOZNJA
REFERENCING NEW ROW AS Novi
FOR EACH ROW
WHEN Novi.Status = 'Z'
BEGIN
    DECLARE
        vDuzina ...,

    SELECT Duzina
    INTO vDuzina
    FROM LINIJA
    WHERE IDLin = Novi.IDLin;

    UPDATE KAMION
    SET UkupnoKm = Duzina +
                (SELECT UkupnoKm FROM KAMION WHERE IDKam = Novi.IDKam),
        UkupnoVoznji = 1 + (SELECT UkupnoVoznji FROM KAMION WHERE IDKam = Novi.IDKam)
    WHERE IDKam = Novi.IDKam;
END
```

c) Sastaviti jedan ili više SQL okidača koji u tabeli LINIJA održavaju redundantne podatke o nazivima mesta (NazivOd, NazivDo) i ukupnom broju vožnji (BrojVoznji) na posmatranoj liniji.

```
CREATE TRIGGER DetaljiLinija1
AFTER INSERT IN LINIJA
REFERENCING NEW ROW AS Novi
FOR EACH ROW
BEGIN
    DECLARE
        vNazivOd ...,
        vNazivDo ...,

        SELECT Naziv
        INTO vIDNazivOd
        FROM MESTO
        WHERE IDMes = Novi.IDMesOd;

        SELECT Naziv
        INTO vIDNazivDo
        FROM MESTO
        WHERE IDMes = Novi.IDMesDo;

        UPDATE LINIJA
        SET NazivOd = vNazivOd, NazivDo = vNazivDo, BrojVoznji = 0
        WHERE IDLin = Novi.IDLin ;
END

CREATE TRIGGER DetaljiLinija2
AFTER UPDATE OF Status ON VOZNJA
REFERENCING NEW ROW AS Novi
FOR EACH ROW
WHEN Novi.Status = 'Z'
BEGIN
    UPDATE LINIJA
    SET BrojVoznji = 1 + (SELECT BrojVoznji FROM LINIJA WHERE IDLin = Novi.IDLin)
    WHERE IDLin = Novi.IDLin;
END
```

d) Sastaviti jedan ili više SQL okidača koji u tabeli MESTO održavaju redundantne podatke o broju i težini poslatih pošiljki (BrojPoslato, TezinaPoslato) iz posmatranog mesta.

```
CREATE TRIGGER DetaljiMesto
AFTER DELETE ON PREVOZI_SE
REFERENCING OLD ROW AS Stari
FOR EACH ROW
BEGIN
    DECLARE
        vTezina ...,
        vIDMesOd ...

    SELECT IDMesOd, Tezina
    INTO vIDMesOd, vTezina
    FROM POSILJKA
    WHERE IDPos = Stari.IDPos;

    UPDATE MESTO
    SET BrojPoslato = 1 + (SELECT BrojPoslato FROM MESTO WHERE IDMes = vIDMesOd),
    TezinaPoslato = vTezina +
        (SELECT TezinaPoslato FROM MESTO WHERE IDMes = vIDMesOd)
    WHERE IDMes = vIDMesOd ;
END
```

e) Sastaviti jedan ili više SQL okidača koji u tabeli VOZAC_KOMITENT održavaju redundantne podatke za potrebe pregleda 'ImeVozača - IDVozača - NazivKomitenta - IDKomitenta'.

```
CREATE TRIGGER VOZAC-KOMITENT
AFTER INSERT IN PREVOZ
REFERENCING NEW ROW AS Novi
FOR EACH ROW
BEGIN
    DECLARE
        vIDVoz ...,
        vIDKom ...,
        vlme ...,
        vNaziv ...

    SELECT IDKom
    INTO vIDKom
    FROM POSILJKA
    WHERE IDPos = Novi.IDPos;

    SELECT IDVoz
    INTO vIDVoz
    FROM VOZNJA
    WHERE IDVzn = Novi.IDVzn;

    IF NOT EXISTS (SELECT *
                   FROM VOZAC_KOMITENT
                   WHERE IDVoz = vIDVoz AND IDKom = vIDKom) THEN

        SELECT lme
        INTO vlme
        FROM VOZAC
        WHERE IDVoz = vIDVoz;

        SELECT Naziv
        INTO vNaziv
        FROM KOMITENT
        WHERE IDKom = vIDKom;

        INSERT INTO VOZAC_KOMITENT (IDVoz, lme, IDKom, Naziv)
        VALUES (vIDVoz, vlme, vIDKom, vNaziv)

    END IF
END
```

Za obradu situacija promene imena vozača i naziva komitenta trebali bi još i okidači koji reaguju na UPDATE u tabelama VOZAC i KOMITENT.