

PRIMER: PRIVATNA LEKARSKA ORDINACIJA 1

Opis

Posmatrani sistem je jedna privatna lekarska ordinacija koja opslužuje pacijente u kojoj spolja angažovani lekari-specijalisti pružaju usluge. Ordinacija radi tako što pacijenti prethodno zakazuju termine dolaska. Svaki zakazani termin može biti ostvaren ili neostvaren (ako pacijent ne dođe). Prilikom svakog ostvarenog termina pacijent može imati jedan ili više tretmana prilikom kojih mu se pruža usluga od strane određenog lekara.

Šema RBP

```
LEKAR ( IDLek, Ime )
USLUGA ( IDUsl, Naziv, Cena )
PACIJENT ( IDPac, Naziv, Adresa )

TERMIN_ZAK ( IDTer, Datum, Vreme, IDPac )
TERMIN_OST ( IDTer, Trajanje )
TRETMAN ( IDTre, MedPod, IDLek, IDUsl, IDTer )
```

Potrebe za analizom

Usluge su grupisane po podgrupama i grupama. Pacijenti se klasifikuju po polu i uzrastu. Lekari se klasifikuju po oblasti.

- Praćenje tretmana po datumu, pacijentima, lekatima, uslugama i iznosu, unutar termina i globalno;
- Praćenje termina po ostvarenosti, datum-vremenu, pacijentima i ukupnom iznosu.

Zadatak

- konceptualno projektovati skladište podataka;
- Sastaviti DMQL definiciju po šemi “pahuljica-zvezda” bez razrade vremenske dimenzije.

a

```
TRETMAN (IDDan, IDPac, IDlek, IDUsl, IDTer, Iznos)
```

```
    DAN (IDDan, Datum, ...)
```

```
    VREME (IDVre, DatumVreme, ...)
```

```
    PACIJENT (IDPac, Ime, Pol, Uzrast)
```

```
    LEKAR (IDLek, Ime, Oblast)
```

```
    USLUGA (IDUsl, Naziv, IDPod)
```

```
        PODOBLAST (IDPod, Naziv, IDObl)
```

```
        OBLAST (IDObl, Naziv)
```

```
    OSTVARENOST (IDOst, Indikator)
```

```
    TERMIN_VEZA (IDTer)
```

```
TERMIN (IDVre, IDPac, IDOst, IDTer, Iznos)
```

b

```

define cube Tretman [Dan, Pacijent, Lekar, Usluga, Termin]:
    Iznos=sum(Cena)

define dimension Dan as (IDDan, Datum, ...)

define dimension Pacijent as (IDPac, Ime, Pol, Uzrast)

define dimension Lekar as (IDLek, Ime, Oblast)

define dimension Usluga as (IDUsl, Naziv,
    Podoblast(IdPod, Naziv,
        Oblast(IdObl, Naziv)
    )
)

define dimension Termin_veza as (IDTer)

-

define cube Termin [Vreme, Pacijent, Ostvarenost, Termin_veza]:
    Iznos=sum(Iznos)

define dimension Vreme as (IDVre, DatumVreme, ...)

define dimension Pacijent as Pacijent in cube Tretman

define dimension Ostvarenost as (IDOst, Indikator)

define dimension Termin_veza as (IDTer)

```

Operativna baza podataka

LEKAR (IDLek, Ime)
USLUGA (IDUsl, Naziv, Cena)
PACIJENT (IDPac, Naziv, Adresa)
TERMIN_ZAK (IDTer, Datum, Vreme, IDPac)
TERMIN_OST (IDTer, Trajanje)
TRETMAN (IDTre, MedPod, IDLek, IDUsl, IDTer)

Skladište podataka

TRETMAN (IDDan, IDPac, IDLek, IDUsl, IDTer, Iznos)
DAN (IDDan, Datum, ...)
VREME (IDVre, DatumVreme, ...)
PACIJENT (IDPac, Ime, Pol, Uzrast)
LEKAR (IDLek, Ime, Oblast)
USLUGA (IDUsl, Naziv, IDPod)
 PODOBLAST (IDPod, Naziv, IDObl)
 OBLAST (IDObl, Naziv)
OSTVARENOST (IDost, Indikator)
TERMIN_VEZA (IDTer)
TERMIN (IDVre, IDPac, IDost, IDTer, Iznos)

Dopunjena operativna baza podataka

LEKAR (IDLek, Ime, *Oblast*)
USLUGA (IDUsl, Naziv, Cena, *IDPod*)
PACIJENT (IDPac, Naziv, Adresa, *Pol, Uzrast*)
TERMIN_ZAK (IDTer, Datum, Vreme, IDPac)
TERMIN_OST (IDTer, Trajanje)
TRETMAN (IDTre, MedPod, IDLek, IDUsl, IDTer)
PODOBLAST (IDPod, Naziv)
PODOBLAST_NOVO (IDPod, Naziv)
LEKAR_NOVO (IDLek, Ime, Oblast)
USLUGA_NOVO (IDUsl, Naziv, Cena, IDPod)
PACIJENT_NOVO (IDPac, Naziv, Adresa, Pol, Uzrast)

Paketno inkrementalno punjenje skladišta podataka

```
INSERT INTO USLUGA_NOVO (IDUs1,Naziv,Cena, IDPod)
SELECT IDUs1,Naziv,Cena, IDPod
FROM USLUGA U
WHERE NOT EXISTS (SELECT IDUs1
                   FROM DW.USLUGA
                   WHERE IDUs1 = C.IDUs1)
```

```
INSERT INTO LEKAR_NOVO (IDLek,Ime, Oblast)
SELECT IDLek,Ime, Oblast
FROM LEKAR L
WHERE NOT EXISTS (SELECT IDLek
                  FROM DW.LEKAR
                  WHERE IDLek = L.IDLek)
```

```
INSERT INTO PACIJENT_NOVO (IDPac,Naziv,Adresa, Pol, Uzrast)
SELECT IDPac,Naziv,Adresa, Pol, Uzrast
FROM PACIJENT P
WHERE NOT EXISTS (SELECT IDPac
                  FROM DW.PACIJENT
                  WHERE IDPac = P.IDPac)
```

```
INSERT INTO PODOBLAST_NOVO (IDPod, Naziv)
SELECT IDPod, Naziv
FROM PODOBLAST P
WHERE NOT EXISTS (SELECT IDPod
                  FROM DW.PODOBLAST
                  WHERE IDPod = P.IDPod)
```

```

FOR EACH Red IN TERMIN_ZAK
IF DATE ( Red.Datum ) > DatumPrethodnogPunjenja
  IF Red.IDTer EXISTS IN TRETMAN
    Red.IDTer > TRETMAN > vIDLek, vIDUs1
    vIDUs1 > USLUGA > vIznos
    Red.Datum -f1> vIDDan
    INSERT INTO DW.Dan
      VALUES ( vIDDan,Red.Datum )
    INSERT INTO DW.TRETMAN
      VALUES ( vIDDan,Red.IDPac,vIDLek,vIDUs1,Red.IDTer, vIznos )
  ENDIF
ENDIF

FOR EACH Red IN TERMIN_ZAK
IF DATE ( Red.Datum ) > DatumPrethodnogPunjenja
  Red.Datum, Red.Vreme -f2> vIDVre
  Red.Datum, Red.Vreme -f3> vDatumVreme
  INSERT INTO DW.Vreme
    VALUES ( vIDVre, vDatumVreme )
  Red.IDTer > -f1 > vIDOST
  IF Red.IDTer EXISTS IN TERMIN_OST
    INSERT INTO DW.OSTVARENOST
      VALUES ( vIDOST, 1 )
    Red.IDTer > TRETMAN > vIDUs1
    vIDUs1 > USLUGA > vIznos
    INSERT INTO DW.TERMIN
      VALUES ( vIDVre, Red.IDPac, vIDOST, Red.IDTer, vIznos )
  ELSE
    INSERT INTO DW.OSTVARENOST
      VALUES ( vIDOST, 0 )
    INSERT INTO DW.TERMIN
      VALUES ( vIDVre, Red.IDPac, vIDOST, Red.IDTer, NULL )
  ENDIF
ENDIF

```