

## Opera surgery data from Medbit Data Warehouse

Document author: anna.hammas@tyks.fi

Kohdat 1-4 on tehty kertaalleen kantaan ktp, skeemaan stage\_dw. Tauluissa nyt vain valideja rivejä. Tarkoitus jatkossa syöttää näihin tauluihin kaikki uudet validit rivit ja päivittää olemassaolevia rivejä invalideiksi, jos näin on lähteessä käynyt. Eli näissä tauluissa tulee tulevaisuudessa olemaan myös invalideja rivejä (valid='n').

1. Aloitamme Opera-datasta (seuraavaksi varmaankin Toti ja Radu)
2. Haemme DataMartista Operaan liittyvät faktataulut (DMF\_LEIKKAUS ja DMF\_TOIMENPIDE) ja niihin liittyvät dimensiot
3. Haemme DW:stä taulun S\_Toimenpiteet\_Opera, josta saamme pre- ja postop-diagnoosit
4. Haku tehdään taulu kerrallaan meidän staging arealle. Medbitin kannassa ei tarvitse tehdä joineja tms. eikä Medbitin serveri kuormitu varmaan kummemmin.
5. DW:ssä käytetään act\_id-sarakkeita pääavaimina ja latpvm-saraketta tiedon uutuuden määrittämiseen.
6. DM:ssä käytetään pääavaimena taulun PK-saraketta ja tiedon uutuuden määrittämiseen sarakkeita DATE\_INSERT ja DATE\_UPDATE. DATE\_UPDATE on null, jos tieto on vain kerran insertoitu eikä vielä kertaakaan päivitetty.
7. DW-tasolla pitää aina huomioida valid kentän arvo. Y = viimeisin voimassaoleva (rivi voi silti olla lähteessä poistettu tai peruttu), N = tietovarastosta löytyy riville uudempi rivi, jossa joku arvoista on päivittynyt. Haetaan vain valideja tietoja joka kohdassa, eli tiedon täytyy olla validi DW:ssä ja lähteessä, ei riitä että jommassakummassa. Varmistutaan siitä, että jos tieto menee ei-validiksi, sekin muutos päivittyy KTP:n kantaan.
8. Tätä ei voi toteuttaa vielä, mutta tulevaisuudessa: Kun alkulataus on tehty, uudet ja muutuneet tiedot haetaan esim. kerran yössä. Tarkistetaan ennen jokaisen taulun hakua, että teidän latauskierron on valmis, taulusta VarSat\_SA\_VSSHPSA\_VSSHPS\_logTable (etlJobName = 'JOB\_Tietovarasto\_VSSHPS' and status = 'ONNISTUI' and endTime is not null).

Tarkoitus on nyt muokata dataa päivittyvästi stage\_dw-skeemasta main\_dev-skeemaan (main\_dev on kehitysversio main-skeemasta). Main\_dev-skeeman taulut leikkaus ja leikkaus\_toimenpide pitäisi täyttää (nyt tyhjä). Koodi\_id-kentät voi alkuvaiheessa jättää tyhjäksi ja laittaa pelkän vastaavan selitteen (esim. leikkaus.paatoimenpide\_koodi\_id tyhjäksi, ja toimenpiteen koodi ja nimi niitä vastaaviin sarakkeisiin). Samoin hoitava\_osasto\_id ja toimenpideosasto\_id tyhjäksi, ja vain selite mukaan.

PDI-transformaation/jobin tekemisen ideoita voi katsoa Arhon tekemistä, tai esim minun ETL/-main/transformations/patologia/insert/insert\_patologia.kjb. Esimerkissä on yritetty täytellä potilas-, henkilön identiteetti- ja potilas asia-taulut myös (main-skeema vaatii tämän). Ainakin jobin/transformaation parametriksi on hyvä laittaa kohdeskeema ja lähdeskeema (kuten patologia-esimerkissä), jotta niitä on helppo muuttaa myöhemmin. Voit käyttää nyt stage\_dw ja main\_dev.

Voisi varmaankin laittaa stage\_dw-cdc\_time-tauluun merkinnän myös stage->main-tiedon päivityksistä. Eli aina kun jobi käynnistyy laittamaan stagesta opera-dataa mainin tauluun leikkaus, tulisi tuohon cdc-tauluun merkintä, jossa taulun nimi on "main\_dev.leikkaus" ja current load timestampiksi tulisi se kyseinen hetki. Kun lataus on ohi, timestamp siirrettäisiin last loadiksi ja currentiin tulisi null. Siten olisi seuraavalla kierroksella helppo katsoa, milloin on viimeksi tehty main\_dev.leikkaukseen päivitys (tuo timestamp), ja ottaa sitä uudemmat datat stage\_dw-tauluista. Ja sitten päivittää näiden päälle uudet timestampit.

### Queries to work on PostgreSQL database ktp, schema stage\_dw

Join query to get (roughly) the necessary data for **main.leikkaus**:

```
select *
```

```
from stage_dw.dm_dmf_leikkaus as l
inner join stage_dw.dm_dmd_erikoisala as e1 on l.FK_DMD_ERIKOISALA = e1.PK_DMD_ERIKOISALA
inner join stage_dw.dm_dmd_erikoisala as e2 on l.FK_DMD_ERIKOISALA_PAATOIMENPIDE =
    e2.PK_DMD_ERIKOISALA
inner join stage_dw.dm_dmd_toimenpide as t1 on l.FK_DMD_TOIMENPIDE_PAA = t1.PK_DMD_TOIMENPIDE
inner join stage_dw.dm_dmd_jonottamisyys as j on l.FK_DMD_JONOTTAMISENSYY =
    j.PK_DMD_JONOTTAMISENSYY
inner join stage_dw.dm_dmd_osasto as o1 on l.FK_DMD_OSASTO_HOITAVA = o1.PK_DMD_OSASTO
inner join stage_dw.dm_dmd_laitos as la on o1.FK_DMD_LAITOS = la.PK_DMD_LAITOS
inner join stage_dw.dm_dmd_osasto as o2 on l.FK_DMD_OSASTO_TOIMENPIDE = o2.PK_DMD_OSASTO
inner join stage_dw.dm_dmd_toimenpide as t2 on l.FK_DMD_TOIMENPIDE_PUOLISUUS = t2.PK_DMD_TOIMENPIDE
inner join stage_dw.dw_S_Person as p on (l.potilas = p.md5Tunniste and p.valid = 'y')
inner join stage_dw.dw_S_Henkilötunnus as a on (p.entity_id = a.entity_id and a.valid = 'y')
where l.lippu_tmpValid = 1;
```

Join query to get (roughly) the necessary data for **main.leikkaus\_toimenpide**:

```
select s.act_id as S_Toimenpiteet_Opera_act_id,
s.preopDiagnoosiCodeIn, s.preopDiagnoosiValue,
s.postopDiagnoosiCodeIn, s.postopDiagnoosiValue,
t.*
from
stage_dw.dm_dmf_toimenpide as t
inner join stage_dw.dm_dmf_leikkaus as l on t.tmpActID = l.H_tmp_act_id
inner join stage_dw.dm_dmd_toimenpide as t1 on t.FK_TOIMENPIDE_TOTEUTUNUT = t1.PK_DMD_TOIMENPIDE
inner join stage_dw.dm_dmd_toimenpide as t2 on t.FK_TOIMENPIDE_SUUNNITELTU = t2.PK_DMD_TOIMENPIDE
inner join stage_dw.dm_dmd_toimenpide as t3 on t.FK_TOIMENPIDE_PUOLISUUS = t3.PK_DMD_TOIMENPIDE
inner join stage_dw.dm_dmd_erikoisala as e on t.FK_DMD_ERIKOISALA = e.PK_DMD_ERIKOISALA
inner join stage_dw.dw_S_Toimenpiteet_Opera as s on t.toimenpideActID = s.act_id
where s.valid = 'y'
and s.tmpValid = 1
and s.toimenpideValid = 1;
```

## Query versions to work on DM and DW

### main.leikkaus

```
select *
from Toimintatilasto.dmf_leikkaus as l
inner join yhteinen.dmd_erikoisala as e1 on l.FK_DMD_ERIKOISALA = e1.PK_DMD_ERIKOISALA
inner join yhteinen.dmd_erikoisala as e2 on l.FK_DMD_ERIKOISALA_PAATOIMENPIDE = e2.PK_DMD_ERIKOISALA
inner join yhteinen.dmd_toimenpide as t1 on l.FK_DMD_TOIMENPIDE_PAA = t1.PK_DMD_TOIMENPIDE
inner join yhteinen.dmd_jonottamisyys as j on l.FK_DMD_JONOTTAMISENSYY = j.PK_DMD_JONOTTAMISENSYY
inner join yhteinen.dmd_osasto as o1 on l.FK_DMD_OSASTO_HOITAVA = o1.PK_DMD_OSASTO
inner join yhteinen.dmd_laitos as la on o1.FK_DMD_LAITOS = la.PK_DMD_LAITOS
inner join yhteinen.dmd_osasto as o2 on l.FK_DMD_OSASTO_TOIMENPIDE = o2.PK_DMD_OSASTO
inner join yhteinen.dmd_toimenpide as t2 on l.FK_DMD_TOIMENPIDE_PUOLISUUS = t2.PK_DMD_TOIMENPIDE
inner join Varsat_DW_VSSHP.dbo.S_Person as p on (l.potilas = p.md5Tunniste and p.valid = 'y')
inner join Varsat_DW_VSSHP.dbo.S_Henkilötunnus as a on (p.entity_id = a.entity_id and a.valid = 'y')
where l.lippu_tmpValid = 1;
```

### main.leikkaus\_toimenpide

```
select s.act_id as S_Toimenpiteet_Opera_act_id,
s.preopDiagnoosiCodeIn, s.preopDiagnoosiValue,
s.postopDiagnoosiCodeIn, s.postopDiagnoosiValue,
t.*
from
Toimintatilasto.dmf_toimenpide as t
inner join Toimintatilasto.dmf_leikkaus as l on t.tmpActID = l.H_tmp_act_id
inner join Yhteinen.DMD_TOIMENPIDE as t1 on t.FK_TOIMENPIDE_TOTEUTUNUT = t1.PK_DMD_TOIMENPIDE
inner join Yhteinen.DMD_TOIMENPIDE as t2 on t.FK_TOIMENPIDE_SUUNNITELTU = t2.PK_DMD_TOIMENPIDE
inner join Yhteinen.DMD_TOIMENPIDE as t3 on t.FK_TOIMENPIDE_PUOLISUUS = t3.PK_DMD_TOIMENPIDE
inner join Yhteinen.DMD_ERIKOISALA as e on t.FK_DMD_ERIKOISALA = e.PK_DMD_ERIKOISALA
inner join Varsat_DW_VSSHP.dbo.S_Toimenpiteet_Opera as s on t.toimenpideActID = s.act_id
where s.valid = 'y'
and s.tmpValid = 1
and s.toimenpideValid = 1;
```