

Hluti 1: Útskýring á SQL fyrirspurnum og niðurstöðum

Liður 1

Fyrsta SQL fyrirspurnin notar CTE (Common Table Expression) til að sameina upplýsingar úr töflum atlas.kingdoms og got.houses. Hún notar FULL OUTER JOIN til að tryggja að bæði ríkisupplýsingar og hús séu með í niðurstöðunum, jafnvel þótt þau eigi sér ekki samsvörun. Gögnin eru síðan sett inn í lannister.tables_mapping þar sem ON CONFLICT skilyrðið uppfærir kingdom_id ef house_id er til.

Úttak: Keyrslan setur inní töfluna tables_mapping öll house_id gildi og kingdom_id gildi með viðeigandi tengingu. Eins og þegar house_id er 354, er kingdom_id 1. Þegar house_id er 17 er kingdom_id 7. Niðurstöður sýna þannig samband milli húsa og ríkja. Þess má geta að sumar línur í töflunni eru null, sem þýðir að engin tenging er við ákveðin hús.

Liður 2

Önnur SQL fyrirspurn sameinar gögn úr atlas.locations og got.houses með CTE til að tengja staðsetningar í Norðrinu við hús þeirra. Notað er skilyrði um að húsin séu í The North. Gögnin eru síðan sett inn í lannister.tables_mapping og WHERE NOT EXISTS skilyrðið tryggir að afrit verði ekki til. Taflan sýnir tengsl milli húsa og staða, þar sem house_id og location_id eru skráð.

Úttak: Niðurstöður fyrir Norðrið má sjá í töflunni þar sem dálkarnir eru location_id, location_name, house_id og house_name.

```
135,Karhold,215,House Karstark of Karhold
186,Torrhen's Square,376,House Tallhart of Torrhen's Square
128,Widow's Watch,132,House Flint of Widow's Watch
264,Stony Shore,128,House Fisher of the Stony Shore
129,Hornwood,202,House Hornwood of Hornwood
125,Oldcastle,239,House Locke of Oldcastle
133,Deepwood Motte,150,House Glover of Deepwood Motte
130,The Dreadfort,34,House Bolton of the Dreadfort
184,Flint's Finger,131,House Flint of Flint's Finger
297,Bear Island,271,House Mormont of Bear Island
297,Bear Island,431,House Woodfoot of Bear Island
```

Liður 3

Þriðja SQL fyrirspurnin notar CTE til að safna öllum húsum í Norðrinu og telur fjölda hliðhollra meðlima fyrir hvert hús með CARDINALITY(). Taflan sýnir hús í Norðrinu með fleiri en 5 hliðhollra meðlimi.

Úttak: Úttakið ætti að vera raðað, eins og sagt var áður, eftir fjölda meðlima (stærstu fyrst) og í stafrófsröð. Úttakið er því:

```
House Stark of Winterfell,88
House Bolton of the Dreadfort,15
House Karstark of Karhold,11
House Manderly of White Harbor,10
House Mormont of Bear Island,8
House Ryswell of the Rills,8
House Tallhart of Torrhen's Square,7
House Glover of Deepwood Motte,6
House Umber of the Last Hearth,6
```

Hluti 2: Aðalpersónur í Krúnuleikum

Sýndartaflan:

CREATE OR REPLACE VIEW lannister.v_pov_characters_human_readable
AS Hér er fyrrnefnda sýndartafla búin til og ef hún var nú þegar til þá er hún
uppfærð með replace view.

Útfærsla á kóða:

- Fyrst er notuð `CONCAT(c.titles[1], ' ', c.name)` AS `full_name` til að skilgreina nafn persónunnar.
- Síðan er skilgreint kyn og er því breytt í 'M' eða 'F' eftir því sem á við.
- Næst eru bæði heiti maka og foreldra sótt og passað er að útiloka ekki makalausar og/eða foreldralausar persónur.
- Svo er notuð reglulega segðin `regexp_matches(c.born, '(\d+)(AC|BC)')` AS `born` til að finna fæðingarár úr `born` dálknum. Reglulega segðin finnur því tölustafinn og AC eða BC, en útilokar annan texta eins og í 'Between 230 AC and 260 AC'.
- Síðan er reiknaður aldur persónunnar með því að annaðhvort reikna dánarár-fæðingarár eða -300 ef persónan er á lífi. Einnig er fæðingarárur sem enda á BC breytt í neikvæðatölu. Notuð er regluleg segð fyrir `c.died` ~ `'^\d+$'`. Ef að það eru bara tölustafir og löglegt ár þá skilar segðin gildi `c.died`, ef svarið er NULL eða einungis texti þá skilar segðin NULL, þ.e.a.s þá er ekkert dánarár.
- Næst er athugað hvort persónan sé á lífi með `c.died IS NULL AS alive` og er útkoman annaðhvort TRUE eða FALSE.
- Svo eru bókunum komið fyrir í lista og raðað eftir útgáfuári með eftirfarandi skipun `ARRAY_AGG(b.name ORDER BY b.released)` AS `book_titles`.

- Að lokum er Select nitað til að velja gögn úr CTE og er niðurstöðunum raðað svo að lifandi persónur koma fyrst og þeim er raðað í lækkandi röð eftir aldri, svo koma látnu persónurnar einnig raðaðar í lækkandi röð eftir aldri.

Hluti 3: PostGIS og föll í PostgreSQL

Í þessari skrá er nánari útskýring á dæmum fyrir hluta 3 sem finna má á branchinu *Hluti 3* inná Github.

1. Flatarmáls konungsríkja

Búið til PostgreSQL function `.get_kingdom_size(int kingdom_id)` sem reiknar út flatarmál konungsríkis út frá landfræðilegum gögnum. Gefið niðurstöðu í ferkílómetrum (þ.e. km²) með engum aukastöfum.

Fallið lannister `.get_kingdom_size` í PostgreSQL reiknar flatarmál konungsríkis í ferkílómetrum með því að nota landfræðileg gögn. Þegar notandi kallar á fallið með auðkenni konungsríkis (`kingdom_id`), athugar það fyrst hvort konungsríkið sé til í gagnagrunninum. Ef ekki, þá birtir það villu sem segir að ekki hafi fundist konungsríki með því tiltekna auðkenni.

Ef konungsríkið er til, reiknar fallið flatarmálið út frá landfræðilegum gögnum sem eru geymd í dálknum *geog* í töflunni *atlas.kingdoms*. PostgreSQL fallið `ST_Area` er notað til að finna flatarmál svæðisins í fermetrum. Til að umbreyta því í ferkílómetra er niðurstöðunni deilt með einni milljón (1.000.000). Að lokum skilar fallið flatarmálinu sem heiltölu.

Þá skilaði úttakið tíu mismunandi flatarmálum konungsríkja í ferkílómetrum í heiltölum (integer).

Hvað gerist ef þú setur inn ólöglegt gildi fyrir `kingdom_id`? Er hægt að koma í veg fyrir það (þ.e.a.s. kasta villu)?

Ef sett er inn ólöglegt gildi fyrir `kingdom_id`, til dæmis -1, þá mun fallið leita að konungsríki með `kingdom_id = -1` í töflunni *atlas.kingdoms*. Þar sem slíkt `kingdom_id` er ekki til, er eftirfarandi skilyrði satt: *IF NOT EXISTS (SELECT 1 FROM atlas.kingdoms k WHERE k.gid = kingdom_id) THEN*

Þetta leiðir til þess að villan `RAISE EXCEPTION 'Kingdom with id % not found', kingdom_id;` verður kölluð og fallið kastar villu með skilaboðunum *“Kingdom with id -1 not found”*. Þannig verður úttakið villa, og engin úrvinnsla mun eiga sér stað eftir það.

Skrifaðu eina SQL fyrirspurn sem notar fallið til að finna heildar flatarmál þriðja stærsta konungsríkisins.

Eins og sjá má í sjálfri forrituninni inná Github, er konungsríkjunum raðað í röð eftir stærð, frá stærsta til minnsta. Hér er notað fallið lannis-

`ter.get_kingdom_size(gid)` til að reikna flatarmál hvers konungsríkis út frá landfræðilegum gögnum.

Þar sem beðið er um aðeins þriðja stærsta konungsríkið, var notað `OFFSET 2`, sem tók fyrstu tvær línurnar út og þá var þriðja línan sú fyrsta í töflunni. `LIMIT 1` tók síðan allar línur út fyrir utan þessa fyrstu og þá var þriðja stærsta konungsríkið aðeins eftir.

Úttakið er Dorne úr töflunni `railway.atlas.kingdoms`

2. Fjöldi staðsetninga og staðsetningar af ákveðnum tegundum

Skrifaðu eina SQL fyrirspurn sem finnur sjaldgæfustu staðsetningategund (`location_type`) utan *The Seven Kingdoms* (hér ættu einnig að koma staðir eins og í Essos, þ.e.a.s. utan *Westeros*), og hvaða heita staðirnir sem tilheyra þeirri tegund?

Fyrst finnur SQL fyrirspurnin sjaldgæfustu staðsetningategundina með því að skoða staði sem eru ekki innan landfræðilegs svæðis “*The Seven Kingdoms*”. Hún flokkar staðsetningarnar eftir tegund (*l.type*), telur hversu margar tilheyra hverri tegund, og finnur þá tegund sem kemur sjaldnast fyrir. Að lokum birtir hún nöfnin á þeim stöðum sem tilheyra þessari tegund.

Úttakið er Landmark þar sem það var sjaldgæfasta staðsetningartegundin (`type`). Landmark kom 16 sinnum fram í töflunni `railway.atlas.locations 2` og hér að neðan eru staðirnir sem tilheyra þeirri tegund í töflunni:

Stone Mill, Pendric Hills, Nagga-s Bones, Fist of the First Men, Hollow Hill, Old Stone Bridge Inn, Water Gardens, Nunn's Deep, Tumblers Falls, Rushing Falls, Mummer's Ford, Inn og the Kneeling Man, Dagger Lake, God's Eye Long Lake, Crossroads Inn.