

## Liður 1

### Taflan `names.sql`:

Taflan `names.sql` geymir upplýsingar um bæði eiginnöfn og millinöfn ásamt tíðni þeirra. Dálkarnir í töflunni eru skilgreindir á eftirfarandi hátt: \* **name**: Þetta er nafn einstaklingsins, tekið er tillit til bæði eiginnafns og millinafns. Gagnategundin er text vegna þess að nafn er alltaf strengur. \* **year**: Ár sem nafnið kemur fyrir, skilgreint sem int. \* **frequency**: Tíðni nafnsins það árið, sem er einnig int. \* **type**: Tegund nafns, þar sem gildin eru annaðhvort “eiginnafn” eða “millinafn”.

### Greining á gögnum:

#### Hvaða hópmeðlimur á algengasta eiginnafnið?

Hér er leitað að nafni með hæsta gildi í dálkinum frequency þar nafnið hefur tegundina (type) „eiginnafn“ vegna þess að við viljum bara fá eiginnafn ekki millinafn. ### Hvenær voru öll nöfnin vinsælust? Hér er leitað eftir árinu þar sem samanlagður fjöldi allra nafna var hæstur. Þetta er gert með því að summu upp frequency fyrir hvert einasta ár. ### Hvenær komu þau fyrst fram? Til að finna hvenær nöfnin komu fyrst fram, leitum við að lægsta gildi í year dálknum fyrir hvert nafn.

## Liður 2

### Hér að neðan eru nánari útskýringar um skráanna `isfolkid.sql`

**Hversu margar aðalpersónur eru í bókabálkinum?** Í skránni má sjá tiltölulega flókna forritun en í stuttu máli er tekið streng úr dálkinum *characters* í töflunni *books*, skipt honum í undirstrengi með því að nota kommur sem skilgreinda skiptanalínu og telur síðan fjölda mismunandi strengja.

**Hversu margar persónur eru í hverri bók?** Byrjar á því að lesa dálkinn *is\_title* úr töflunni *books*. Hugsunin á bakvið forritunina hér var að taka annars vegar nöfn með kommu og hins vegar nöfn án komma, finna mismuninn á þeim og þá var hægt að sjá hversu fjölda komma. Þá er hægt að vita fjölda persóna með því að bæta +1 við fjölda komma. **Sem dæmi:** Strengurinn Alda, Benni, Elísabet Hér eru tvær kommur en nöfnin eru þrjú og þess vegna bætum við +1 við fjölda komma til að vita fjölda persóna.

**Hversu oft kemur Þengill fyrir í bókunum?** Hér er notað endurtekna CTE (Common Table Expression) með “WITH RECURSIVE” til að skipta streng sem er geymdur í dálkinum *characters* í töflunni *books*. Hún brýtur strenginn upp í einingar aðskildar með kommu og leitar síðan að tilteknu nafni, í þessu

tilfelli Þengill. Forritsbúturinn mun síðan skila fjölda raða þar sem nafnið Þengill finnst í dálkinum `characters` í töflunni `books`.

**Sem dæmi:** Ef við værum með strenginn `Alda, Benni, Elísabet`. Þá myndi strengurinn verða að: `Alda, Benni Elísabet` Forritsbúturinn telur þannig fjölda ákveðins nafns.

**Hvað eru margir af Paladín ættinni?** Hér telur forritsbúturinn allar raðir í töflunni `family` þar sem `name` dálkurinn inniheldur strenginn “Paladín”. Hún skilar heildarfjölda þeirra raða sem innihalda þetta nafn.

**Hversu algengur er illi arfurinn af þeim sem eru útvalin?** Hér eru allar raðir taldar í töflunni `family` þar sem dálkurinn `chosen_one` hefur gildið “evil”. Hún skilar fjölda þeirra raða sem uppfylla þetta skilyrði.

**Hver er fæðingartíðni kvenna í bókabálkinum?** Eins og við skiljum spurninguna, er verið að spurja um fjölda kvenna sem fæðast á gefnum árum í bókabálkinum. Forritunin finnur hversu margar konur (`gender = 'F'`) eru fæddar á hverju ári í gagnagrunninum. Hún skilar fjölda fæðinga fyrir hvert ár og birtir það í tímaröð, frá elsta til nýjasta.

**Hvað er Ísfólkið margar blaðsíður samanlagt?** Tekur summu allra blaðsíðna (`pages`) frá töflunni `books`

**Hvað er meðallengd hvers þáttar af Ískisum?** Þessi fyrirspurn skilar meðallengd(AVG) allra raða í dálkinum `length` úr töflunni `storytel_iskisur` og skilar þá meðallengd hvers þáttar af Ískisum.

## Til að keyra hina skrána, `create_isfolkid.sql`

Það eina sem þarf að skrifa í terminal, eftir að hafa opnað gagnagrunninn, er `.schema` og þá birtust töflurnar sem unnið var með í liði 2

## Liður 3

Í þessum liði var unnið með tvær töflur; `hlaupog timataka` sem tók saman upplýsingar sérhvert hlaup, t.d.: `* id: Auðkenni * upphaf: Tímasetning hlaups (dagsetning og tími dags) * endir: Áætluð lok hlaups (dagsetning og tími dags) * nafn: Nafn hlaups * fjoldi: Fjöldi þátttakenda`

`createTables.sql` býr til töflu uppsetninguna til þess að vista gögnin seinna eins og kom fram í README.

`insertData.py` tekur csv skrána sem var búin til af `Lidur3.py` og vistar það inn í gagnagrunninn.

Skráin `Lidur3.py` sá um að ná í upplýsingarnar á <https://timataka.net> og færa það yfir í CSV form. Það var gert með því að sækja fyrst forsíðuna á síðunni. Þar notar skráin segð (regex) til þess að ná í ár-mánuð og lista af hlaupum. Næst tekur það lista af hlaupunum og nær í hlekkinn að því, nafn og dagsetningu. Eftir það var náð í öll hlaupin undir ágúst 2022 (þar sem við erum Lannister). Loks voru þau öll tekin og náð í grunnupplýsingar um hlaup, eins og upphaf, endir, nafn og fjölda. Það er svo loks vistað í CSV skrá.

### **Segðir**

Þær segðir sem útbúnaðar voru má finna í skránni `Lidur3.py`. Þær voru fundnar með því að taka html kóðann af síðunni og finna segðir sem gátu lesið inn sem dæmi má nefna öll hlaup og linka sem stóðu undir ágúst 2022, eða alla dálka í töflunum (rank,nafn,upphafstíma,lokatíma os.frv.). Notast var við [Regex101.com](https://regex101.com) til að sjá sjónrænt hvort segðin ætti við.

### **Niðurstaða:**

Spurningin í þessu dæmi var hvort fjöldi keppenda stemmir nákvæmlega við skráð hlaup. Því miður getum við ekki fullyrt það með vissu þar sem ekki var nægur tími til að klára töfluna `timataka`. En vonandi var skráin hlaup fullnægjandi og skýr.