Útskýring á dæmi 1

1. Tíðni nafna á íslandi

Tilgangur vekefnisins er að sameina gögn út tveimur CVS skrám í einn SQLite gagngagrunnþ Í honum verða gögnin í einni töflu sem er síðan hægt að sækja upplýsingar úr og svara spurningum úr grunninum.

Kóðann má finna á names.sql skránni en skoðum nánar virkni hans og ákveðnar skipunir innan hans.

CREATE TABLE IF NOT EXISTS names

Er skipunin sem er notuð til að búa til bæði names og temp töfluna. Sér til þess að hún sé eingöngu búin til ef hún er ekki nú þegar til. Þetta hjæalpar til við villur og veitir skýrleika.

.mode csv .import data/first_names_freq.csv temp

Hér má sjá SQLite skipunina: .import, en hún er notuð til að færa gögn frá skrá inn í töflu.

INSERT INTO names (name, year, frequency, type) SELECT name, year, frequency, 'first' FROM temp;

Hér má sjá dæmi um hvernig SQL skipunin INSERT INTO, er notuð til að setja inn gögn í töflu. Skipunin SELECT velur gögn frá temp töfunni og færir inn í names töfluna. „first" er hér notað sem gildi fyrir type dálkinn, á í þessu tilfelli við eiginöfn.

DELETE FROM temp;

DELETE FROM skipunin eyðir öllum skrám úr temp töflunni. En sú tafla var búin til ofar í kóðanum.

Sama er síðan gert til að sækja gögnin um seinni eiginöfn inn í töfluna. Að lokum er þó notað:

DROP TABLE IF EXISTS temp;

En DROP TABLE IF EXISTS eyðir töflunni en ekki er þörf á henni lengur þar sem gögnin sem á að vinna með eru öll komin yfir í names töfluna. Temp taflan er í raun notið sem tímabundin staðsetning fyrir CVS gögnin. En það er þæginlægt því þá er fyrst hægt að taka gögn af CSV skránum og síðan setja inn í SQLite töfluna.

í spurningu eitt er verið að leita eftir algengasta nafninu milli hópmeðlima.

SELECT name, SUM(frequency) AS total_frequency FROM names WHERE name IN ('Snæfríður', 'Dalrós', 'Erlendur') AND type = 'first' GROUP BY name ORDER BY total_frequency DESC LIMIT 1;

Í fyrstu línu kóðans er valið (SELECT) nafn og heildar algengi hvers nafns. Eru SQL skipanirnar hér nokkuð auðskýranlegar og framkvæma það sem maður ímyndar sér að þær geri. FROM skipunin velur úr töflunni names. WHERE, skipunin takmarkar nöfnin sem leitað er eftir í þessi þrjú nöfn. AND type = 'first, sér til þess að aðeins sé skoðað eignöfn þar sem spurning er að leita eftir því. GROUP BY name skipunin, skiptir niðurstöðunum eftir hverju nafni. ORDER BY total_frequency DESC setur niðurstöðurnar í lækkandi röð með algenasta nafninu efst. Limit 1: sér til þess að aðeins ein röð prentast eða niðurstöður fyrir algengasta nafnið en ekki hin tvö líka.

Svar: Erlendur,234

Af nöfnunum þremur er Erlendur algengasta eiginafnið með 234 niðurstöður.

Spurning tvö snýst um að finna hvenær nönfnin voru vinsælust.

SELECT name, year, MAX(frequency) AS max_frequency FROM names WHERE name IN ('Snæfríður', 'Dalrós', 'Erlendur') GROUP BY name ORDER BY name;

MAX(frequency) skipunuin finnur hámarks algengni fyrir hvert nafn. Finnur í leið árið sem það var algengast. Sama að ofan að skipunin FROM finnur þessi gögn í töflunni names. GROUP BY name skiptir niðurstöðunum eftir nafni. Þannig hvert nafn fær sína niðurstöðu. ORDER BY name, skipunin prentar niðurstöðurnar í stafrósröð.

Svar: "Dalrós",1998,2 Erlendur,1970,11 "Snæfríður",1992,6

Samkvæmt niðurstöðunum var Dalrós vinsælast árið 1998 þegar tvær voru nefndar því nadniþ Árið 1970 var Erlendur vinsælast með ellefu sem voru skírðir því nafni. Snæfríður reyndist svo vinsælast 1992 þegar sex voru skírðir það.

í spurningu þrjú á að finna hvaða ár nöfnin komu fyrst fram.

SELECT name, MIN(year) AS first_appearance FROM names WHERE name IN ('Snæfríður', 'Dalrós', 'Erlendur') GROUP BY name ORDER BY name;

Skipunin MIN(year) finnur lægsta ártalið sem hvert nafn birtist. Sömu SQL skipanir gilda og að ofan. Niðurstöður teknar úr töflunni names. Flokkaðar eftir nafni og settar í stafrósröð.

Svar: "Dalrós",1918 Erlendur,1912 "Snæfríður",1926

Samkvæmt töflunni með gögnunum kemur Dalrós fyrst fram 1918, nafnð Erlendur upp 1912 og Snæfríður 1926.

Útskýring á dæmi 2

Búa til isfolkid.sql

Þessi skrá sýnir hvernig gagnagrunnurinn isfolkid.db er uppbyggður án þess að nota hafa gögnin sjálf með. Hér notuðum við skipuninaCREATE TABLE IF NOT EXISTS nafn á "nafnátöflu" til þess að búa til hverja töflu. Hver tafla inniheldur dálka með mismunandi upplýsingum og kemur svo fram af hvaða gerð þær upplýsingar eru. Sem dæmi inniheldur línan pages INTEGER, í töflunni books, dálk með upplýsingum um blaðsíður bókanna og eru þær upplýsingar af gerðinni INTEGER, eða heiltölu.

isfolkid.sql

Skipanaskráin isfolkid.sql svarar mismunandi spurningum út frá upplýsingum frá skránni isfolkid.db.

Hversu margar aðalpersónur eru í bókabálkinum? SELECT SUM(LENGTH(characters) - LENGTH(REPLACE(characters, ',', '')) + 1) AS total_main_characters reiknar heildarfjölda persóna í dálkinum characters fyrir allar bækur í books töflunni með því að greina fjölda nafna sem eru aðskilin með kommum SVAR: 102

Hversu margar persónur eru í hverri bók? Skipunin telur fjölda persóna í hverri bók með því að reikna fjölda kommna í characters dálkinum, þar sem hver koma aðskildur hverja persónu. Hún skilar titli bókarinnar (is_title) ásamt fjölda persóna (total_characters) sem eru skráðar í characters dálknum. Svar: Álagafjötrar|2 Nornaveiðar|3 Hyldýpið|1 Vonin|1 Dauðasyndin|2 Illur arfur|3 Draugahöllin|1 Dóttir böðulsins|2 Skuggi fortíðar|2 Vetrarhríð|2 Blóðhefnd|3 Ástarfuni|2 Fótspor Satans|3 Síðasti riddarinn|2 Austanvindar|2 Gálgablómið|3 Garður dauðans|2 Gríman fellur|2 Tennur drekans|1 Hrafnsvængir|3 Um óttubil|3 Jómfrúin og vætturin|3 Vorfórn|4 Í iðrum jarðar|2 Guðsbarn eða galdranorn|3 Álagahúsið|1 Hneykslið|2 Ís og eldur|3 Ástir Lúcífers|3 Ókindin|6 Ferjumaðurinn|2 Hungur|2 Martröð|2 Konan á ströndinni|1 Myrkraverk|1 Galdratungl|4 Vágestur|2 Í skugga stríðsins|2 Raddirnar|2 Fangi tímans|3 Djöflafjallið|1 Úr launsátri|1 Í blíðu og stríðu|1 Skapadægur|1 Böðullinn|3 Svarta vatnið|1 Er einhver þarna úti?|1

Hversu oft kemur Þengill fyrir í bókunum? characters LIKE '%Þengill%' Leitar í dálkinum characters eftir öllu sem inniheldur orðið "Þengill" og COUNT(*) Telur fjölda þeirra lína þar sem orðið "Þengill" kemur fyrir. Svar: 2

Hversu margir af Paladín ættinni? Skipunin SELECT COUNT*() FROM family WHERE name LIKE '%Paladín%'; er notuð til að telja hversu margir einstaklingar í family töflunni hafa "Paladín" í nafni sínu. SELECT COUNT()* kallar á alla línurnar sem uppfylla skilyrðin og skilar heildarfjölda þeirra. FROM family leitar í töflunni family og WHERE name LIKE '%Paladín%' leitar að nafni sem inniheldur "Paladín" hvar sem er í strengnum Svar: 9

Hversu algengur er illi arfurinn af þeim sem eru útvalin? SELECT COUNT*() FROM family WHERE chosen_one='evil' SELECT COUNT()* kallar á allar línurnar sem uppfylla skilyrðin og skilar heildarfjölda þeirra, FROM family leitar í töflunni family og WHERE chosen_one='evil' finnur þá einstaklinga sem hafa gildið 'evil' í dálknum chosen_one Svar: 18

Hver er fæðingatíðni kvenna? Skipunin SELECT birth, COUNT(*) AS total_females FROM family WHERE gender = 'F' AND birth IS NOT NULL GROUP BY birth ORDER BY birth ASC; er notuð til að finna fæðingatíðni kvenna í family töflunni. GROUP BY birth skiptir niðurstöðunum eftir fæðingarárinu, þannig að fjöldi kvenna í hverju ári sé talinn saman og ORDER BY birth ASC raðar niðurstöðunum í hækkandi röð eftir fæðingarárinu Svar: 1556|1 1564|1 1579|1 1583|1 1587|1 1601|2 1602|1 1610|1 1616|1 1627|1 1628|1 1629|1 1634|1 1638|1 1652|1 1655|1 1656|1 1674|1 1677|1 1678|1 1679|1 1694|1 1697|1 1698|1 1699|1 1716|1 1720|1 1750|1 1752|1 1769|1 1775|1 1777|1 1779|1 1788|1 1796|1 1800|2 1820|1 1825|1 1829|1 1830|1 1836|1 1842|1 1848|1 1853|1 1855|1 1872|2 1880|1 1884|1 1894|1 1902|1 1909|1 1910|1 1922|1 1926|1 1929|1 1937|1 1938|1 1942|1 1943|1 1948|1

Hvað er Ísfólkið margar blaðsíður samanlagt? SELECT SUM(pages) FROM books; SELECT SUM(pages) sækir summu allra gilda í dálkinum pages FROM books Þetta segir að notar gögn úr töflunni books Svar: 8023

Hvað er meðallengd hvers þáttar af Ískisum? SELECT AVG(length) sækir meðaltal allra gilda í dálkinum length og FROM storytel_iskisur sækir gögnin úr töflunni storytel_iskisur+ Svar: 116.40206185567

Útskýringar á dæmi 3

Útskýring á RQL skjalinu

Búa til töfluna hlaup:

- CREATE TABLE IF NOT EXISTS hlup: Býr til töflu sem heitir hlaup ef hún er þegar ekki til.
- id INTEGER PRIMARY KEY AUTOINCREMENT: id er aðalskýringar (primary key) sem verður sjálfkrafa aukið fyrir hvern nýjan þáttakanda. Þetta tryggir að hver þátttakandi hefur einstakt auðkenni.

- upphaf DATETIME og endir DATETIME: upphaf og endir geyma dagsetningu og tíma sem tengist byrjun og endi hlaupsins.
- nafn TEXT: nafn geymdir nafn hlaupsins sem texta.
- fjoldi INTEGER: fjoldi geymir fjölda þátttakenda sem taka þátt í hlaupinu.
- startadur INTEGER og lokadur INTEGER: Geyma upplýsingar um staðsetningu (eða önnur töluleg gildi) sem tengjast hlaupi.

Búa til töfluna timataka:

- CREATE TABLE IF NOT EXISTS timataka: Býr til töflu sem heitir timataka ef hún er ekki þegar til.
- id INTEGER PRIMARY KEY AUTOINCREMENT: id er aðalskýringar sem verður sjálfkrafa aukið.
- hlaup_id INTEGER: hlaup_id er útlit á id í hlaup töflunni, sem tengir þátttakandann við tiltekið hlaup.
- bib INTEGER: bib geymir númer þátttakanda (t.d. númerið á keppnistímanum).
- nafn TEXT: nafn geymir nafn þátttakanda.
- timi TEXT: timi geymir tímann sem þátttakandi lauk hlaupi.
- aldur TEXT: aldur geymir aldur þátttakanda.
- club TEXT: club geymir upplýsingarnar um klúbbinn sem þátttakandi tilheyrir.
- split TEXT, behind TEXT, chiptime TEXT: Þessar dálkar geyma frekari upplýsingar um tímaskiptingu, eftir, og klukku-tíma.
- FOREIGN KEY (hlaup_id) REFERENCES hlaup(id): Þetta segir að hlaup_id í timataka tengist id í hlaup. Þetta tryggir að aðeins þátttakendur sem tengjast skráðri hlaupi séu leyfðir.

Skoða þátttakendafjölda:

- SELECT h.nafn, h.fjoldi, COUNT(t.id) AS participants: Velur nafn hlaups (h.nafn), fjölda þátttakenda (h.fjoldi), og fjölda þátttakenda sem hafa skráð sig í hlaup (COUNT(t.id)).
- FROM hlaup h: Velur gögnin úr hlaup töflunni og skýrir hana sem h.
- LEFT JOIN timataka t ON h.id = t.hlaup_id: Sameinar hlaup og timataka töflur þar sem hlaup_id í timataka samsvarar id í hlaup. LEFT JOIN tryggir að allar skráningar úr hlaup eru sýndar, jafnvel þótt engir þátttakendur séu skráðir í timataka.
- GROUP BY h.id: Skiptir niður gögnunum eftir hlaupi (h.id), svo að hægt sé að telja þátttakendur í hverju hlaupi.
- HAVING h.fjoldi != participants: Sýnir aðeins hlaupin þar sem fjöldi skráðra þátttakenda (frá timataka) er ekki sá sami og skráð fjöldi í hlaup.

Útksýring á REGEX

Upplýsingar um hlaupið:

- `Start time\s*</small>\s*<h4>(.*?)</h4>`:
  - `Start time`: Leitar að textanum `"Start time"`.
  - `\s*`: Leyfir að vera hvaða fjöldi bil (whitespace) á eftir `"Start time"`.
  - `</small>`: Leitar að `</small>` HTML merkinu.
  - `\s*`: Leyfir aftur bil á eftir.

    - `<h4>(.*?)<\/h4>`: Leitar að tíma í `</h4>` merkinu. `(.*?)` er grípandi hópur sem dregur út tímann (start time) og leyfir hvaða fjölda af stöfum sem er á milli `<h4>` og `</h4>`.

- `.?Finished<\/small>\s*<h4>(\d+) / (\d+)<\/h4>`:

    - `.*?`: Leyfir hvaða fjölda af stöfum á milli fyrri og seinni hluta, en stoppar þegar `"Finished"` kemur.

    - `Finished<\/small>`: Leitar að `"Finished"` og `</small>` merkinu.

    - `\s*<h4>(\d+) / (\d+)<\/h4>`: Leitar að talnaskiptum í `<h4>` merki. `(\d+)` er grípandi hópur sem dregur út tölur (fjöldi þátttakenda sem byrjaði og lokið) á undan og á eftir `"/"`.

- `.?Est\. finish time<\/small>\s*<h4>(.*?)<\/h4>`:

    - `.*?`: Leyfir aftur hvaða fjölda af stöfum.

    - `Est\. finish time<\/small>`: Leitar að `"Est. finish time"` og `</small>` merkinu.

    - `\s*<h4>(.?)<\/h4>`: Leitar aftur að áætlaðra tímans í `<h4>` merki og dregur út ímann (end time) með `(.*?)`.

Upplýsingar um hvern þátttakanda:

- `<tr>`: Leitar að byrjun á nýrri röð í HTML töflu.

- `.*?<td class_"hidden-xs">(\d+)</td>`:

    - `.*?`: Leyfir hvaða fjölda af stöfum áður en næsta merki kemur.

    - `<td class"hideen-xs">(\d+)</td>`: Leitar að `<td>` merki sem hefur klasa `"hidden-xs"`. `(\d+)` dregur út númer þátttakanda (rank).

- `.*?<td>(\d+)</td>`: Leitar að næsta `<td>` merki og dregur út bib númer þátttakanda.

- `.*?<td>(.*?)<\/td>`: Leitar að næsta `<td>` merki sem inniheldur nafn þátttakanda. `(.*?)` dregur út nafnið.

- `.*?<td class="hidden-xs">(.*?)<\/td>`: Leitar að næsta `<td>` merki með klasa `"hidden-xs"` og dregur út fæðingarár.

- `.*?<td>(.*?)<\/td>`: Leitar að næsta `<td>` merki og dregur út klúbbinn.

- `.*?<td>(.*?)<\/td>`: Leitar að næsta `<td>` merki og dregur út tímann (finish time) sem þátttakandi lauk á.