Gagnasafnsfræði

Páll Melsted

28. okt

Sýn

Í SQL er hægt að gefa algengum fyrirspurnum nafn með `CREATE VIEW`. Þá verður til sýn (view) á töfluna sem má nota sem venjulegt töflunafn í fyrirspurnum. Sýnir einfalda flóknar fyrirspurnir og koma í veg fyrir copy/paste.

```
CREATE VIEW ParamountMovies AS
SELECT title,year
FROM Movies
WHERE StudioName = 'Paramount';
```

Sýn

Við getum notað sýnina ParamountMovies eins og hverja aðra töflu

```
SELECT title
FROM ParamountMovies
WHERE year > 1980;
```

Jafnvel þótt ParamountMovies sé hluti af Movie þá megum við bara vísa í þá dálka sem eru skilgreindir í sýninni.

With

Sýnin sem við búum til verður hluti af gagnagrunninum og kemur fram í schemanu. Ef við viljum einfalda flókna fyrirspurn en ekki geyma sýnina þá getum við notað `CREATE TEMPORARY VIEW` sem hverfur þá um leið og við lokum tenginunni.

Einnig er hægt að nota `WITH` á undan fyrirspurn til að einfalda hana

```
WITH M as (SELECT model,price from laptop
           union select model,price from pc
           union select model,price from printer)
SELECT M.model
FROM M
WHERE M.price = (select max(price) from M);
```

Þetta virkar bara í nýrri útgáfum af sqlite.

Sýn og breytingar

Það er hægt að breyta sýn undir vissum kringumstæðum. Almennt séð er það ekki góð hugmynd. Helstu ástæður þess að nota sýnir er að leyfa takmarkaðan aðgang að gögnum og geyma algengar fyrirspurnir.

Með sýn er hægt að veita sumum notendum takmarkaðan aðgang að gögnum sem í heild sinni eiga ekki að vera sýnileg öllum.

Með sýnum er líka hægt að vera með flókið niðurbrot í BCNF töflur þar sem öll gögn eru geymd en sýnin sér um að mynda upprunalegu töfluna.

Vísar

SQL þarf ekki að skila niðurstöðum í ákveðinni röð, nema að `ORDER BY` sé hluti af fyrirspurn. Röðun er dýr, kostar $O(n \log(n))$ fyrir n stök.

Þegar við gerum tengingar á töflum (join) þá þurfum við að framkvæma leit.

Lausnin við báðum vandamálum er að halda utan um gögnin í réttri röð.

INDEX

Við búum til vísi (index) á töflu með

```
CREATE INDEX TitleIndex ON Movies(title);
```

eða

```
CREATE INDEX KeyIndex ON Movies(title,year);
```

Útfærsla

Gagnagrunnurinn notar vísi án þess að við þurfum að biðja um það

- Þegar við gerum join á `(title,year)` þá flýtir vísirinn fyrir því að finna rétt gildi
- Ef við gerum `SELECT * FROM Movie WHERE title='Star Wars';` þarf ekki að lesa alla töfluna
- Það mega vera margar mismunandi raðanir á gögnunum, þau þurfa ekki að vera geymd í neinni röð.

Diskar

Eru hægir, mjög, mjög hægir.

http://www.eecs.berkeley.edu/~rcs/research/interactive_latency.html

---

B-tré

Svipuð hugmynd og tvíleitartré.

- Geymir marga, K lykla í einum hnúti og vísar á $K+1$ undirlykla.
- Hnútarnir mega hafa mest m lykla og minnst $m/2$ (nema rótin)
- Hæðin á trénu verður $\log_m(\frac{n+1}{2})$ fyrir n lykla.

Síður

Diskar geyma gögn í síðum (page). Þegar við biðjum um gögn er náð í heila síðu. Algeng stærð er 4K. Í vinnslu á gagnagrunni getur mestur tími farið í að ná í síður af diski.

Með B-trjám lágmörkum við fjölda lestra sem þarf til að ná í gögnin. Nokkurs konar helmingunarleit sem tekur tillit til þess að diskalestur er dýr.

Vísar

Ef vísar hraða fyrirspurnum og tengingum af hverju búum við ekki til alla mögulega vísa?

- það kostar pláss að geyma vísinn, þarf að lesa af diski
- þegar við setjum inn gögn þarf að uppfæra vísinn, skrifa á disk
- sumar töflur eru svo litlar að það er einfaldara að lesa þær allar í einu

---

Hvernig á að velja vísa?

1. Ef við höfum lykil (PRIMARY KEY) sem verður notaður í join þá viljum við hafa setja vísi á hann (reyndar gerir sqlite þetta alltaf fyrir okkur)
2. Ef við erum með leitarvél, t.d. leitum að nemendum eftir nafni jafnvel þótt það sé ekki lykill
3. Tölur sem er raðað eftir eða gildi eru valin á bili, t.d. ár, dagsetningar o.s.frv.

Sýn geymd á diski

Venjuleg sýn (VIEW) er bara uppskrift af því hvernig mætti búa til vensl og er fyrirpurnin keyrð í hvert skipti sem við skoðum sýnina.

Stundum getur það verið betra að skrifa sýnina beint á disk. Þá er fyrirspurnin fljótari að keyra

```
CREATE MATERIALIZED VIEW MovieProd AS
  SELECT title, year, name
  FROM Movies, MovieExec
  WHERE cert = producerC;
```

Gagnagrunnurinn keyrir þá fyrirspurnina og skrifar á disk.

Sýn geymd á diski

Ef við uppfærum aðra hvora töfluna þá sér gagnagrunnurinn um að uppfæra `MovieProd` sýnina sem við skrifuðum á disk.

Ef við keyrum

```
INSERT INTO Movie(title,year,producerC) VALUES ('Kill Bill', 2003, 23456);
```

þar sem 23456 vísar á Tarantinto þá keyrir gagnagrunnurinn fyrirspurnina

```
SELECT *
FROM PRODUCER
WHERE cert=23456;
```

og setur gögnin inn í `MovieProd` fyrir okkur. Það kostar því mjög lítið að viðhalda gögnunum í sýninni á diskinum.

Endurskrift á fyrirspurnum

Ef einhver fyrirspurn er hæg er hægt að spurja gagnagrunninn hvað hann er að gera með `EXPLAIN QUERY PLAN SELECT ...`

Til að flýta algengum fyrirspurnum er oft nóg að bæta við

- lyklum og ytri lyklum ()
- vísum sem geta hjálpað til við fyrirspurnina, sérstaklega á join

Þegar þetta er ekki nóg er hægt að geyma niðurstöðurnar, t.d. ef verið er að framkvæma join á tveimur töflum og svo leitað í niðurstöðunum.

Þá þarf að klippa út hluta af fyrirspurn og geyma sem `MATERIALIZED VIEW`

Uppskrift af endurskrift

Ef við erum með fyrirspurn, Q, á forminu

```
SELECT LQ FROM RQ WHERE CQ
```

og materialized view, V, á forminu

```
SELECT LV FROM RV WHERE CV
```

þá getum við notað V inni í Q ef

- allar töflur í RV koma fyrir í RQ
- CQ er jafngilt `CV AND C` þar sem C eru einhver skilyrði (eða C tómt ef CQ=CV)
- Ef C er ekki tómt þá eru dálkarnir sem koma fyrir í C líka í töflunum í RV
- dálkarnir í LQ sem eru úr töflum í RV koma líka fyrir í LV

Uppskrift af endurskrift

Ef við erum með fyrirspurn, Q, á forminu

```
SELECT LQ FROM RQ WHERE CQ
```

og materialized view, V, á forminu

```
SELECT LV FROM RV WHERE CV
```

Ef öll skilyrðin eru uppfyllt er hægt að endurskrifa fyrirspurnina með því að

- skipta RQ út fyrir V og þær töflur sem koma ekki fyrir í LV
- skipta CQ út fyrir C (eða sleppa WHERE ef C er tómt)

Dæmi

Við höfum sýnina MovieProd

```
CREATE MATERIALIZED VIEW MovieProd AS
  SELECT title, year, name  FROM Movies, MovieExec
  WHERE cert = producerC;
```

og fyrirspurnina

```
SELECT starName FROM StarsIn, Movies, MovieExec
WHERE movieTitle = title AND movieYear = year
      AND producerC = cert AND name = 'George Lucas';
```

Þá getum við endurskrifað hana sem

```
SELECT starName FROM StarsIn, MovieProd
WHERE movieTitle = title AND movieYear = year
      AND name = 'George Lucas';
```

sem fækkar tengingum úr þremur í tvær.