TÖL303G

Gagnasafnsfræði

Snorri Agnarsson

Samskipti í námskeiðinu

- Fyrirlestrar á föstudögum
- Dæmatímar einu sinni í viku
- Piazza
  - Signup Link: piazza.com/hi.is/fall2024/tl303g
- Vikublöð (innihalda verkefni)
- Gradescope
- Gradiance

Fyrir næstu viku

- Setjið upp SQLite
- Skráið ykkur á gradiance:
  - Farið á: http://infolab.stanford.edu/~ullman/pub/stud-guide.html
  - Notið háskólatölvupóstfang ykkar í skráningu
  - Notið **class token „1C1E60A4“** til að tengjast TÖL303G
- Leysið dæmi á verkefnablaði
  - Gradescope skil, sjá http://gradescope.com
  - Gradiance skil, sjá http://www.gradiance.com
- Lesið kafla 1, 2.1, 2.2, 6.1, 6.2

Yfirlit

- Hvað eru gagnagrunnar?
- Af hverju gagnagrunnar?
- Venslagagnagrunnar
- Kostir og gallar venslagagnagrunna
- SQLite

Yfirlit

- Hefðbundin notkun gagnagrunna
  - Geymsla talna og texta
- Margmiðlagagnagrunnar
  - Myndir, hljóðskeið, myndskeið
- Landfræðigagnagrunnar
  (GIS – Geographic Information Systems)
  - Geyma og greina landakort, veðurgögn, gervitunglamyndir

Yfirlit

- Vöruhús gagna og greiningarkerfi (OLAP)
  - Sækja og greina gögn s.s. viðskiptaupplýsingar úr stórum gagnagrunnum
  - Oft upplýsingar úr fleiri en einum gagnagrunni og úr fleiri en einu ytra kerfi
  - Styðja ákvarðanatöku
- Rauntímagagnagrunnar
  - Stýra iðnaðar- og framleiðsluferlum

Inngangur

- Gagnagrunnur
  - Safn skyldra gagna
  - Þekktar staðreyndir sem hafa merkingu og unnt er að skrá
- Dæmi um stóra gagnagrunna með viðskiptalegan tilgang
  - amazon.com
  - midi.is
  - Valitor

Inngangur

- Gagnagrunnur er líkan af hluta af raunveruleikanum (Universe of Discourse – UoD)
- Stendur fyrir einhverja ásýnd raunveruleikans
- Gögn sem hægt er að túlka í samhengi
- Byggður í tilteknum tilgangi

Inngangur

- Gagnagrunnskerfi (Database Management System – DBMS)
  - Safn forrita
  - Gerir notendum kleift að smíða gagnagrunna og viðhalda þeim
- Skilgreining gagnagrunns
  - Tilgreina gagnatög, gagnamót og skorður á gögn sem geymd eru
- Gögn um gögn (Meta-data)
  - Gagnagrunnsskilgreining
  - Geymd í gagnagrunninum sem efnisyfirlit

Inngangur

- Aðgerðir á gagnagrunn
  - Fyrirspurnir og uppfærslur
  - Framleiðsla á skýrslum
- Samnýting gagnagrunns
  - Leyfir mörgum notendum samtímis aðgang að gagnagrunninum
- Notendaforrit
  - Senda fyrirspurnir og uppfærslur á gagnagrunninn gegnum gagnagrunnskerfið (DBMS)

Inngangur

- Fyrirspurn
  - Sækir hluta gagnanna úr gagnagrunninum
- Færsla (transaction)
  - Veldur lestri og skrift í gagnagrunninn
- Meðal öryggisþátta eru:
  - Kerfisvernd (System protection)
  - Gagnavernd (Security protection)
- Viðhald gagnagrunns
  - Þróa þarf kerfið eftir því sem kröfur breytast

Hví læra um gagnagrunna?

- Fræðilegt svar
  - Lærum um áhugaverð reiknirit, rökfræði og tölvunotkun
- Svar fyrir forritara
  - Lærum að kreista upplýsingar úr gagnagrunnum
- Svar fyrir gagnanörda
  - Lærum að vinna með síbreytileg og stór söfn gagna

Markmið námskeiðs

Að öðlast sterkan bakgrunn í gagnagrunnum. Eftir námskeiðið eigið þið að geta útfært meðalstóran venslagagnagrunn, þ.m.t. að geta

- lesið og skrifað SQL fyrirspurnir með helstu fyrirspurnaaðgerðum
- skilgreint venslagagnagrunna út frá fallákveðum á þann hátt að þeir uppfylli BCNF staðalform eða 3NF staðalform
- lesið og skrifað einindavenslarit til að skilgreina kröfur á gagnagrunn og geta út frá einindavenslariti skilgreint samsvarandi venslagagnagrunna
- lesið og skrifað segðir í venslaalgebru sem samsvara tilteknum fyrirspurnum
- lesið og skrifað SQL skipanir til að stýra aðgangi notenda að gagnagrunni

Hvernig eigum við að geyma gögnin?

- Á pappír, í möppum og skjalaskápum?
- Í bókum, í bókasöfnum o.s.frv?
- Í textaskrám (t.d. TSV)?
- Í töflureiknaskrám (t.d. Excel)?
- Í gagnamótum svo sem trjám og listum?
- Á gataspjöldum?

Kostir gagnagrunna

- Leyfa flóknari fyrirspurnir en töflureiknar og textaskrár
- Gerðir fyrir samskeiða vinnslu
- Lifa af náttúruhamfarir
- Allt geymt á stafrænu tölvutæku sniði (á ,,diski``)

- Hvernig eru gagnagrunnar útfærðir?
  - **Röng spurning á þessu stigi.** Við munum eilítið íhuga þessa spurningu seinna, en við viljum geta hugsað um gagnagrunna **án þess** að þessi spurning sé á borðinu.

Venslagagnagrunnar

- Langflestir gagnagrunnar nota venslalíkanið. Við munum fylgja því eftir þar til í lok námskeiðsins. Öll gögn eru geymd í töflum þar sem hver dálkur hefur **nafn**. Töflurnar eru kallaðar **vensl (*relation*)**.

Movie:

title	year	length
Pretty Woman	1990	119
The Man Who Wasn't There	2001	116
Logan's run	1976	
Star Wars	1977	124
Empire Strikes Back	1980	111
Star Trek	1979	132

Venslalíkanið

Fyrst lagt fram af Codd árið 1969

[The relational model] provides a basis for a high level retrieval language which will yield maximal independence between programs on the one hand, and machine representation and organization of data on the other.

Venslalíkanið

- Öll gögn eru geymd í venslum (*relation*, tafla)
- Öll vensl hafa eigindi (*attribute*, dálkur) með nöfnum
- Hvert eigindi hefur óðal (*domain*) af löglegum gildum
- Vensl eru mengi af n-dum (*tuple*)

Venslalíkanið

- Gagnagrunnur = safn af töflum
- Vensl = tafla
- Eigindi = dálkur í töflu
- n-d = röð í töflu (borið fram „ennd“)

Venslalíkanið og SQL

- SQL er mál sem leyfir okkur að skilgreina vensl og framkvæma fyrirspurnir
  - Allir venslagagnagrunnar nota SQL
  - Hægt að nota beint á gagnagrunn eða gegnum forrit
  - Myndar millilag óháð forritunarmáli eða gagnagrunni

Hönnun gagnagrunna

- Ekki er alltaf ljóst hvernig best er að hanna gagnagrunn
  - **Hönnun** þýðir hvernig **venslin** eru skilgreind
- Stór hluti námskeiðsins mun fara í hönnun gagnagrunna þegar við höfum náð tökum á SQL fyrirspurnum

Hvernig er best að geyma eftirfarandi gögn?

- Nemendaskráning í HÍ
- Pantanir hjá Dominos
- Pakkasendingar hjá Póstinum
- Sparisjóðsreikningar hjá Landsbankanum
- Þjóðskrá Íslands

Hve flókin eru gögnin? Hve mikið af gögnum? Hve ört breytast gögnin? Hvaða aðilar fá aðgang að gögnunum? Hvernig tökum við öryggisafrit? Hvar geymum við öryggisafrit? Hvernig notum við öryggisafrit?

Fyrirspurnir í gagnagrunnum

Movie gagnagrunnurinn:

```
title                          year length
------------------------------ ---- ------
Pretty Woman                   1990 119
The Man Who Wasn't There       2001 116
Logan's run                    1976
...
```

SQL fyrirspurn til að ná í gögnin:

```
SELECT *
FROM Movie;
```

Meira SQL

```
SELECT title, year
FROM Movie
WHERE length > 120;
```

```
SELECT *
FROM Movie
WHERE year > 15*length;
```

SELECT þekking

- Hve margir hafa séð

```
SELECT title, year
FROM Movie
WHERE length > 120;
```

eða svipað, áður?

- En hvað með þetta?

$$\pi_{\text{title,year}}\left(\sigma_{\text{length}>120}(\text{Movie})\right)$$

Uppbygging SELECT skipunar

```
SELECT name1,name2,...
FROM relation1,relation2,...
WHERE condition;
```

- Við notum * sem styttingu á að velja alla dálka
- Í næstu viku munum við tala um hvernig við vinnum með mörg vensl (margar töflur) samtímis í einni SELECT skipun

Málfræði og merking SQL

- SQL er læsilegt, yfirleitt er hægt að skilja einfaldar SELECT fyrirspurnir
- Er við lærum meira um SQL þurfum við að íhuga
  - Málfræði SQL. Hvernig **má skrifa** SQL setningar?
  - Merkingarfræði SQL. Hvað **þýðir** SQL setningin?
- Málfræðin er einföld
- Til að ræða merkingarfræðina þurfum við **líkan**

Venslalíkanið

- SQL er byggt á venslalíkaninu (relational model) og venslaalgebru (relational algebra)
- Venslalíkanið gefur okkur nákvæma merkingu á SQL setningum
- Venslaalgebran gerir okkur kleift að tala um jafngildar aðgerðir og endurskrifa fyrirspurnir

Önnur hlutverk SQL

- SQL er ekki aðeins fyrirspurnamál
- Auk fyrirspurna notum við SQL til að
  - Skilgreina ný vensl
  - Breyta gögnum
  - Setja upp skorður (constaints) og gikki (triggers)
  - Halda utan um notendur og öryggi
  - Stýra hreyfingum á gagnagrunnum fyrir marga notendur

Dálkaval

- Með ofanvarpi/dálkavali (*projection*) getum við valið hluta af eigindum úr venslum, endurnefnt (*rename*) og endurraðað (*reorder*)

```
SELECT title AS name, length AS duration
FROM Movies;
```

- Oft er þægilegt að nota slíkt til að stytta SQL fyrirspurnir

Dálkaval

- Við getum líka reiknað út einfaldar segðir sem nýja dálka í útkomunni

```
SELECT
        vara AS nafn,
        1.255*kostnadur AS heimsendingarkostnadur
FROM ...;
```

Val á röðum, skilyrði (*condition*)

- Í **WHERE** hluta getum við sett hvaða Boolean segð sem er, svipað og í Java, C, C++, C#, Pascal o.s.frv.
  - Notum = fyrir samanburð til að athuga hvort tvö gildi eru jöfn, ekki ==
  - Notum <> í stað !=
  - Strengir hafa einfaldar gæsalappir, skrifum ’abc’, ekki ”abc”
  - Skeytum saman strengjum með ||
- SQL skipanir gera ekki greinarmun á há- og lágstöfum, en strengjasamanburður gerir það

Val

- Útkoma úr Boolean segð í WHERE er yfirleitt TRUE eða FALSE. Á þessi gildi má beita AND, OR og NOT aðgerðum, eins og í öðrum forritunarmálum. Svigar hafa venjuleg áhrif, t.d.:

```
SELECT title
FROM Movie
WHERE (year>1970 OR length<90) AND studioName='MGM';
```

Strengjaleit

- SQL hefur sérstaka aðgerð fyrir strengjaleit: LIKE
- s LIKE p skilar TRUE ef strengurinn s passar við mynstrið p
- Tveir stafir hafa sérstaka merkingu í mynstri
  - _ passar við hvaða staf sem er
  - % passar við hvaða streng sem er, jafnvel tóman

```
SELECT title
FROM Movie
WHERE title LIKE 'Star ____ '; -- fjögur _ í röð
```

Strengjaleit

- Til að leita að ’ eða % þarf að skrifa stafina tvisvar
- Í SQLite þarf að nota ESCAPE undirklausu í LIKE klausunni fyrir % í mynstri

```
SELECT title FROM Movie WHERE title LIKE '% ' 's% ';
SELECT 'a%b' LIKE 'a\%b' ESCAPE '\';
SELECT 'axb' LIKE 'a\%b' ESCAPE '\';
```

NULL „gildið“

- NULL er löglegt gildi í SQL sem hefur mismunandi merkingu eftir samhengi
  - Notum NULL þegar gildi vantar
  - Notum NULL þegar ekkert gildi á við
  - Notum NULL þegar við vitum ekki viðeigandi gildi
- NULL má nota í hvaða segð sem er
  - 1+NULL er NULL, 1<NULL er NULL, 1 AND NULL er NULL, 0 OR NULL er NULL, NOT NULL er NULL (prófið þetta í sqlite)
  - Hins vegar: 0 AND NULL er 0, 1 OR NULL er 1 (hvers vegna er það rökrétt?)

NULL

- NULL getur verið gildi í n-d
- Til að athuga hvort gildi sé NULL þarf að nota IS NULL aðgerðina

SELECT title, length FROM Movie WHERE length=NULL;

skilar engu (hvers vegna?), en

SELECT title, length FROM Movie WHERE length IS NULL;

skilar röðinni:

Logan's run|

(eða sömu röð á öðru sniði, eftir stillingum)

UNKNOWN vs NULL

- Í SQLite getur útkoman úr Boolean segð verið ein af þrennu:
  - 0 (sem þýðir ósatt, FALSE)
  - 1 (sem þýðir satt, TRUE)
  - NULL (sem þýðir óþekkt, UNKNOWN)
- Í sumum gagnagrunnum er útkoman úr Boolean segð ein af þrennu:
  - FALSE
  - TRUE
  - UNKNOWN
- Í sumum gagnagrunnum er reynt að gera greinarmun á gildi sem vantar (NULL) og rökgildi sem er óþekkt (UNKNOWN)
- Í þessu námskeiði höfum við oftast ekki áhyggjur af þessu

Meira NULL í SQLite – prófið þetta

```
.nullvalue NULL
SELECT NULL=NULL;
SELECT NULL=NULL OR NULL<>NULL;
SELECT 1<NULL OR 1>=NULL;
SELECT NULL OR NOT NULL;
SELECT NOT NULL;
SELECT NULL AND 0;
SELECT NULL OR 1;
SELECT NULL AND 1;
SELECT NULL AND 0;
```

NULL / UNKNOWN

- Úr SELECT fyrirspurn með WHERE klausu birtast **aðeins** raðir þar sem skilyrðið er TRUE (1 í SQLite)
- Raðir þar sem skilyrðið er FALSE (0 í SQLite) eða NULL eða UNKNOWN **birtast ekki**

```
SELECT * FROM Movie WHERE length<0 OR length>=0;

SELECT * FROM Movie WHERE (length<0 OR length>=0) IS NULL;
```