

# **TÖL303G**

## **Gagnasafnsfræði**

### **Vikublað 10**

Snorri Agnarsson

28. október 2022

#### **Efni vikunnar**

Fjöllum um sýnir (view), vísa (index) og um þýðingu og ágæðun fyrirspurna í gagnasafnskerfum. Lesið kafla 8.1–8.4.

We will discuss views and indexes and how queries are compiled and optimized in database systems. Read chapters 8.1–8.4.

#### **Verkefni – Assignments**

##### **Gradiance**

###### **Íslenska**

Finnið Gradiance verkefni á venjulegum stað, verkefni vikunnar heita „Verkefni 10“.

###### **English**

You will find the Gradiance assignment in the usual place, the weeks assignment is named "Verkefni 10".

##### **Gradescope**

###### **Íslenska**

Verkefnin sem skila skal í Gradescope eru eftirfarandi.

Skrifið SQL fyrirspurnir fyrir eftirfarandi í COMPANY gagnagrunninum. Athugið að þið munuð stundum vilja nota lykilorðið EXISTS og faldaðar fyrirspurnir.

Athugið að, eins og alltaf, til þess að fyrirspurn teljist rétt þá er ekki nóg að hún skili réttu svari miðað við núverandi innihald gagnagrunnsins. Hún þarf að skila réttri niðurstöðu fyrir öll möguleg rétt ástönd gagnagrunnsins.

Þið finnið SQLite útgáfu af gagnagrunninum í skránni `company.db` í forritunar-möppunni í `forritun.zip` í Canvas. Skipunin `.schema` inni í `sqlite3` sýnir ykkur skilgreiningar á töflunum. Þið finnið einnig einindavenslarit fyrir gagnagrunninn á glærum.

1. Finnið eftirnöfn (*last name*) allra millistjórnenda, þ.e. þeirra starfsmanna sem stjórna einhverjum og eru undir stjórn einhvers. Þetta eru þeir starfsmenn sem hafa yfirmann (*supervisor*) og eru einnig yfirmenn einhvers. Þið megið reikna með að enginn starfsmaður sé sinn eigin yfirmaður.
2. Finnið eftirnöfn þeirra deildarstjóra sem vinna að einhverju verkefni (*project*) sem ekki er stýrt af þeirra eigin deild. Athugið að það eru ekki allir yfirmenn deildarstjórar.
3. Finnið eftirnöfn þeirra starfsmanna sem vinna í engu verkefni.
4. Finnið fornöfn (*first name*) og eftirnöfn þeirra starfsmanna sem vinna í öllum verkefnum.

## English

The assignments that should be turned in through Gradescope are the following.

Write SQL queries for the following in the COMPANY database. You may sometimes want to use the keyword EXISTS and nested queries.

As always, in order for a query to be correct it is not enough that it returns the correct result for the current contents of the database. It must always return the correct result for all possible legal states of the database.

You will find an SQLite version of the database in the file `company.db` in the programming folder in `forritun.zip` in Canvas. The command `.schema` in `sqlite3` shows you the table definitions. You can also find E/R diagrams for the database on the slides.

1. Find the last names of all middle managers, i.e. those employees who supervise someone and are supervised by someone. You may assume that no employee supervises himself.
2. Find the last names of those department managers who work on some project that is not controlled by their own department. Not all supervisors are department managers.

3. Find the last names of those employees who work in no project.
4. Find the first and last names of those employees who work in all projects.

## Til hliðsjónar — For Reference

### Íslenska

Til hliðsjónar getum við íhugað eftirfarandi fyrirspurn fyrir kvikmyndagagnagrunninn:

**Verkefni:** Sýnið fyrirspurn sem finnur nöfn allra kvikmyndastjarna sem léku aðalhlutverk í öllum kvikmyndum. (Athugið að það er alls ekki það sama og að finna þær kvikmyndastjörnur sem léku aðalhlutverk í einhverri kvikmynd. Það er heldur ekki það sama og að finna þær kvikmyndastjörnur sem léku aðalhlutverk í öllum þeim kvikmyndum sem hafa aðalhlutverk.)

### Lausn:

```
SELECT name
FROM MovieStar
WHERE NOT EXISTS
  (SELECT *
   FROM Movie
   WHERE NOT EXISTS
     (SELECT *
      FROM StarsIn
      WHERE starName=name
        AND movieTitle=title
        AND movieYear=year
     )
  )
;
```

Þessi fyrirspurn byggist á því að kvikmyndastjarna lék í öllum kvikmyndum þá og því aðeins að að ekki sé til kvikmynd sem stjarnan lék ekki í. Við erum að nota vel þekkta reglu í umsagnareikningi (*predicate calculus*) sem segir að

$$(\forall x : P(x)) \iff \neg(\exists x : \neg P(x))$$

Að segja „Fyrir öll  $x$  gildir  $P(x)$ “ er jafngilt því að segja „Ekki er til  $x$  þannig að  $P(x)$  gildi ekki“.

Fyrirspurnin er kannski frekar flókin en ekki endilega mjög óvanalega flókin. Flestar fyrirspurnir eru kannski einfaldari en þessi, en það eru sennilega margar fyrirspurnir í sífelldri notkun sem eru flóknari en þessi.

Við erum eðlilega að gera ráð fyrir að allar kvikmyndastjörnur séu skráðar í MovieStar, allar kvikmyndir séu skráðar í Movie og að öll aðalhlutverk í kvikmyndum séu skráð í StarsIn og að aðeins aðalhlutverk séu skráð þar.

Ef við setjum spurninguna upp í umsagnarökfræði (*predicate calculus*), sem flest ykkar þekkja úr Stærðfræðimynstrum, þá felst vandamálið í að finna öll  $x$  sem uppfylla umsagnarökfræðisegðina  $\forall y : L(x, y)$  þar sem óðalið (*domain*) fyrir  $x$  er allar kvikmyndastjörnur (lyklar í MovieStar), óðalið fyrir  $y$  er allar kvikmyndir (lyklar í Movie) og  $L(x, y)$  hefur merkinguna „kvikmyndastjarna  $x$  lék í kvikmynd  $y$ “.

Í SQL er ekki til bein samsvörun fyrir umsagnarökfræðisegðina  $\forall y : L(x, y)$ , en hins vegar er það nær að til sé bein samsvörun fyrir jafngildu segðina  $\neg(\exists y : \neg L(x, y))$ . Málið flækist eilítið vegna þess að við þurfum einnig að endurrita innri segðina  $\neg L(x, y)$  yfir í  $\neg(\exists u, v : L(u, v) \wedge x = u \wedge v = y)$  svo unnt sé að komast að sanngildi hennar í gagnagrunninum vegna þess að við geymum ekki neikvæðar fullyrðingar í gagnagrunnum. Þess í stað þurfum við að „leita“ í gagnagrunninum að mótdæmi og erum aðeins viss um sanngildi  $\neg L(x, y)$  ef ekkert mótdæmi er að finna.

## English

For reference we can consider the following query for the movies database:

**Problem:** Show a query that finds the names of all movie stars who starred in all movies. (This is definately not the same as finding the stars who starred in some movie. It is also not the same as finding those who starred in all movies that have a star.)

## Solution:

```
SELECT name
FROM MovieStar
WHERE NOT EXISTS
  (SELECT *
   FROM Movie
   WHERE NOT EXISTS
     (SELECT *
      FROM StarsIn
      WHERE starName=name
        AND movieTitle=title
        AND movieYear=year
     )
  )
;
```

This query is based on the fact that a movie star starred in all movies if and only if there does not exist a movie that the star did not star in. We are using a well known

rule in predicate calculus which states that

$$(\forall x : P(x)) \iff \neg(\exists x : \neg P(x))$$

Saying "For all  $x$  we have  $P(x)$ " is equivalent to saying "There does not exist an  $x$  such that  $P(x)$  does not hold".

The query may be a tad complicated but not necessarily unusually complicated. Most queries may be simpler than this one, but there are probably many queries in constant use that are more complex than this one.

We are naturally assuming that all movie stars are registered in MovieStar, all movies are registered in Movie, and all star roles (and only star roles) are registered in StarsIn.

If we state the question in predicate calculus, which most of you know from Discrete Mathematics, the problem consists of finding all  $x$  that fulfil the predicate calculus formula  $\forall y : L(x, y)$  where the domain of  $x$  is all movie stars (keys in MovieStar), the domain for  $y$  is all movies (keys in Movie) and  $L(x, y)$  has the meaning "movie star  $x$  starred in movie  $y$ ".

In SQL there is no direct correspondence for the predicate calculus formula  $\forall y : L(x, y)$ , but on the other hand we are closer to having a direct correspondence with the equivalent formula  $\neg(\exists y : \neg L(x, y))$ . Matters become a little complicated because we also have to rewrite the inner formula  $\neg L(x, y)$  to  $\neg(\exists u, v : L(u, v) \wedge x = u \wedge v = y)$  in order to ascertain its truth value because we do not register negative facts (falseness) in databases. Instead we have to "search" the database for counterexamples and are only convinced of the truth of  $\neg L(x, y)$  if no counterexample is to be found.