Scraping data from the web

Data about the municipalities of Sweden

Let’s say we want to create a municipal database

We want to keep track of:

- Municipal name (e.g. Stockholms Stad)
- Municipal web URL
- Whether the municipality supports HTTPS

Where could we gather such data?

I found a web page with all names and URLs to the 290 municipalities:

http://skl.se/tjanster/kommunerlandsting/adressuppgifterkommuner.1246.html

```
<p class="normal" id="A"><a href="http://www.ale.se/">Ale kommun<br/></a>449
80 ALAFORS<br/>0303-33 00 00<a
href="mailto:kommun@ale.se"><br/>kommun@ale.se</a></p><p class="normal"><a
href="http://www.alingsas.se/">Alingsås kommun</a><br/>441 81
ALINGSÅS<br/>0322-61 60 00<a
href="mailto:kommunstyrelsen@alingsas.se"><br/>kommunstyrelsen@alingsas.se</
a></p><p class="normal"><a href="http://www.alvesta.se/">Alvesta
kommun</a><br/>342 80 ALVESTA<br/>0472-150 00<a
href="mailto:klk@alvesta.se"><br/>klk@alvesta.se</a></p>.....
```

Parsing out the URLs

```
$ cat kommuner-raw|tr '=' '\n'|tr '"' '\n'|grep http
http://www.ale.se/
http://www.alingsas.se/
http://www.alvesta.se/
http://www.aneby.se/
http://www.arboga.se/
http://www.arjeplog.se/
.....
# Did we get all 290?
$ cat kommuner-raw|tr '=' '\n'|tr '"' '\n'|grep http | wc -l
290
$ cat kommuner-raw|tr '=' '\n'|tr '"' '\n'|grep http > KommunURLer.txt
```

Parsing out the names

```
$ cat kommuner-raw|tr '=' '\n'|tr '"' '\n'|grep '<br/>'|egrep -v '^$' \
   |cut -d '>' -f2|cut -d '<' -f1|egrep -v '^$' > Names.txt


# Yes there are other ways to achieve that...
```

Checking URLs for HTTPS support

Idea is to loop throught the URLs and perform a HTTP HEAD request for the URL but with “https” instead of “http” and filter out the ones without support for HTTPS:

```
$ for URL in $(cat KommunURLer.txt)
do
 HTTPS=$(echo $URL|sed -e 's/http/https/')
 HEAD $HTTPS -t 2 &> /dev/null || echo $HTTPS
done > HTTP-kommuner.txt

# Tip: To install HEAD command on mac os: http://lwp.interglacial.com/ch01_03.htm
# Tip: Install lwp-request for Windows with cygwin if you want to try this
https://cygwin.com/cgi-bin2/package-grep.cgi?grep=lwp-request&arch=x86_64
In cygwin, the following is the same as HEAD: lwp-request -m HEAD
```

Creating a table

Let’s start with a not so normalised table:

```
CREATE TABLE municipalities(MunicipalityID INTEGER PRIMARY KEY NOT NULL,
 Name text, URL text, HTTPS boolean);
```

Populating the table

The plan here is to create a text file with the SQL INSERT statements we need to populate the table. Let’s start with adding the URLs:

```
for URL in `cat KommunURLer.txt`
  do echo "INSERT INTO municipalities(URL) VALUES('"$URL"');"
done > INSERT_URLS.sql
$ sqlite3 my_municipalities < INSERT_URLS.sql
sqlite> SELECT * FROM municipalities LIMIT 5;
Mun  Name              URL                   HTTPS
---  ----------------  --------------------  -----
1                      http://www.ale.se/
2                      http://www.alingsas.
3                      http://www.alvesta.s
4                      http://www.aneby.se/
5                      http://www.arboga.se
```

Adding Names to the table

Using the fact that names and URLs are in the same order (they were extracted in the same order), we can create an index and increase it in the loop:

```
i=1
cat Names.txt|while read NAME
 do echo "UPDATE municipalities SET NAME='"$NAME"' WHERE MunicipalityID=$((i++));"
done > INSERT_NAMES.sql
```

```
sqlite> SELECT * FROM municipalities LIMIT 5;
MunicipalityID  Name                      URL                       HTTPS
--------------  ------------------------  ------------------------  -----
1               Ale kommun                http://www.ale.se/
2               Alingsås kommun           http://www.alingsas.se/
3               Alvesta kommun            http://www.alvesta.se/
4               Aneby kommun              http://www.aneby.se/
5               Arboga kommun             http://www.arboga.se/
```

Updating the HTTPS boolean

We have a list of HTTP municipalities (non-https cities). Let’s set all cities to TRUE (1) and then set the non-https cities to FALSE (0):

```
sqlite> UPDATE municipalities SET HTTPS=1;
for HTTPS in $(cat HTTP-kommuner.txt)
 do echo "UPDATE municipalities SET HTTPS=0 WHERE URL='"$HTTPS"';"
done > UPDATE_HTTPS.sql
sqlite3 my_municipalities < UPDATE_HTTPS.sql

sqlite> SELECT * FROM municipalities LIMIT 5;
MunicipalityID  Name                      URL                       HTTPS
--------------  ------------------------  ------------------------  -----
1               Ale kommun                http://www.ale.se/        0
2               Alingsås kommun           http://www.alingsas.se/   1
3               Alvesta kommun            http://www.alvesta.se/    0
4               Aneby kommun              http://www.aneby.se/      0
5               Arboga kommun             http://www.arboga.se/     0
```

Some statistical reports

How many municipalities support HTTPS and how many don’t?

```
sqlite> SELECT COUNT(*), HTTPS FROM municipalities GROUP BY HTTPS;
COUNT(*)    HTTPS
----------  ----------
222         0
68          1
```

Some statistical reports

How many municipalities are called “Something Stad” and how many “Something Kommun”?

```
sqlite> SELECT COUNT(*) FROM municipalities WHERE Name LIKE '% stad';
COUNT(*)
----------
14
sqlite> SELECT COUNT(*) FROM municipalities WHERE Name LIKE '% kommun';
COUNT(*)
----------
275
```

Some statistical reports

Who are these “Stad” municipalities?

```
sqlite> SELECT Name FROM municipalities WHERE Name LIKE '% stad';
Borås stad
Göteborgs stad
Haparanda stad
Helsingborgs stad
Landskrona stad
Lidingö stad
Malmö stad
Mölndals stad
Solna stad
Stockholms stad
Sundbybergs stad
Trollhättans stad
Vaxholms stad
Västerås stad
```

Twilight zone - is there a connection? O_o

```
sqlite> SELECT Name,HTTPS FROM municipalities WHERE Name LIKE '% stad';
Name                  HTTPS
--------------------  ------
Borås stad            1
Göteborgs stad        1
Haparanda stad        0
Helsingborgs stad     0
Landskrona stad       0
Lidingö stad          1
Malmö stad            1
Mölndals stad         1
Solna stad            1
Stockholms stad       1
Sundbybergs stad      1
Trollhättans stad     0
Vaxholms stad         1
Västerås stad         0
```

Using a boolean column in the where clause

```
sqlite> SELECT Name FROM municipalities WHERE Name LIKE '% stad' AND HTTPS;
Name
--------------------
Borås stad
Göteborgs stad
Lidingö stad
Malmö stad
Mölndals stad
Solna stad
Stockholms stad
Sundbybergs stad
Vaxholms stad
```

Nested query example

Nested queries will not be tested in the exam. This is just for show.

```
sqlite> SELECT Name,MunicipalityID FROM municipalities
 WHERE MunicipalityID=(SELECT MAX(MunicipalityID) FROM municipalities);
Name                  MunicipalityID
--------------------  --------------
Övertorneå kommun     290
```

What server are they running?

Most municipalities give away a hint of what web server they are running:

```
$ HEAD http://www.almhult.se/
200 OK
Connection: close
Date: Thu, 05 Nov 2015 11:40:42 GMT
Server: Apache-Coyote/1.1
Vary: Accept-Encoding
Content-Type: text/html;charset=UTF-8
Client-Date: Thu, 05 Nov 2015 11:40:38 GMT
Client-Peer: 194.103.158.6:80
Client-Response-Num: 1
Client-Transfer-Encoding: chunked
Set-Cookie: JSESSIONID=605A0936C203C9A96EE04A29ED5DDCF4; Path=/
X-UA-Compatible: IE=EDGE
```

What server are they running?

Using egrep to filter out only Server:

```
$ HEAD http://www.almhult.se/|egrep '^Server'
Server: Apache-Coyote/1.1
```

Putting it all together:

```
$ for url in $(cat KommunURLer.txt)
   do server="$url $(HEAD $url -t 2|grep '^Server:')"
   echo $server
 done > URLvsServer.txt
# Filtering out only those who has given us Server:

$ grep Server URLvsServer.txt > Servers.txt
```

Fixing the table - adding Server

```
sqlite> CREATE TABLE municipalities2(MunicipalityID INTEGER PRIMARY KEY NOT
NULL,Name text, URL text, HTTPS boolean,Server text);

sqlite> INSERT INTO municipalities2(MunicipalityID,Name,URL,HTTPS) SELECT
MunicipalityID,Name,URL,HTTPS FROM municipalities;


sqlite> DROP TABLE municipalities;
sqlite> ALTER TABLE municipalities2 RENAME TO municipalities;
sqlite>
```

Updating the Server columns

```
$ cat Servers.txt|while read line
 do URL=$(echo $line|cut -d ' ' -f1)
 SERVER=$(echo $line|cut -d ' ' -f3-)
 echo "UPDATE municipalities SET Server=\""$SERVER"\" WHERE \
       URL='"$URL"';"
done > UPDATE_SERVERS.txt
$ sqlite3 my_municipalities < UPDATE_SERVERS.txt
```

What's the correlation between server and https?

```
sqlite> SELECT COUNT(*),HTTPS FROM municipalities WHERE Server LIKE
'Apache%' GROUP BY HTTPS;
97|0
40|1
sqlite> SELECT COUNT(*),HTTPS FROM municipalities WHERE Server LIKE
'Microsoft%' GROUP BY HTTPS;
95|0
24|1
/* Just speculation off course ;-) */
```

Discussion - how could we normalise the DB?

Revisiting the table design:

```
CREATE TABLE "municipalities"(MunicipalityID INTEGER PRIMARY KEY NOT NULL,
  Name text, URL text, HTTPS boolean,Server text);

/* Idea: keep municipalities short, e.g. MunicipalityID,Name */

/* What other tables could we have? */

/* Tradeoff: The more tables we have, the more complicated queries ;-) */
```

What municipalities only support www.-adresses?

```
CREATE TABLE municipality_id_www_only
    (MunicipalityID INTEGER PRIMARY KEY NOT NULL, AllowsOnlyWWW boolean);
```

What municipalities only support www.-adresses?

```
$ for url in $(cat KommunURLer.txt)
  do short=$(echo $url|sed -e 's/www\.//')
  HEAD $short -t4 &> /dev/null || echo $url
done > OnlyWWWCitites.txt

$ for url in $(cat OnlyWWWCitites.txt)
  do echo "UPDATE municipality_id_www_only SET AllowsOnlyWWW=1 \
          WHERE MunicipalityID= \
          (SELECT MunicipalityID FROM municipalities \
           WHERE URL='"$url"');"
done > UPDATE_WWW_ONLY.sql
```

Using the new table

```
sqlite> SELECT Name FROM municipalities m JOIN municipality_id_www_only mo
    ON m.MunicipalityID = mo.MunicipalityID WHERE mo.AllowsOnlyWWW;
Askersunds kommun
Eksjö kommun
Eslövs kommun
Essunga kommun
Grums kommun
Gävle kommun
Habo kommun
Hudiksvalls kommun
Laholms kommun
Lycksele kommun
...
...
```

More correlations?

```
sqlite> SELECT Name,HTTPS,Server FROM municipalities m JOIN municipality_id_www_only mo ON
m.MunicipalityID = mo.MunicipalityID WHERE mo.AllowsOnlyWWW;
Askersunds kommun|1|Apache-Coyote/1.1
Eksjö kommun|0|Microsoft-IIS/8.0
Eslövs kommun|0|Apache-Coyote/1.1
Essunga kommun|0|Microsoft-IIS/7.5
Grums kommun|0|Unknown
Gävle kommun|1|
Habo kommun|0|Microsoft-IIS/8.0
Hudiksvalls kommun|0|
Laholms kommun|0|Microsoft-IIS/8.0
Lycksele kommun|0|Microsoft-IIS/6.0
Munkfors kommun|0|Microsoft-IIS/7.5
Ockelbo kommun|0|
Osby kommun|0|Microsoft-IIS/7.5
Robertsfors kommun|0|Apache/2.2.16
(Debian)
Smedjebackens kommun|0|Apache-Coyote/1.1
Storumans kommun|0|Microsoft-IIS/8.0
Svenljunga kommun|0|Apache-Coyote/1.1
Sävsjö kommun|0|Apache-Coyote/1.1
Timrå kommun|0|
Tingsryds kommun|0|
Tranås kommun|0|Apache-Coyote/1.1
Upplands Väsby
kommun|1|Apache-Coyote/1.1
Vansbro kommun|0|Apache
Varbergs kommun|0|Apache-Coyote/1.1
Vingåkers kommun|0|Microsoft-IIS/6.0
Vårgårda kommun|0|Apache-Coyote/1.1
Öckerö kommun|1|Apache-Coyote/1.1
Övertorneå kommun|0|Microsoft-IIS/7.5
```