

National Asset Project Vejledning 3.0

Dette projekt har til formål at indsamle og analysere økonomiske regnskaber fra danske virksomheder i perioden 2012-2023. Projektet har en primær fokus på vurdering af, hvordan virksomheder forvalter deres aktiver. Gennem en omfattende dataindsamling og analyse vil vi generere grafer og visuelle repræsentationer for at belyse forskellige aspekter af årsregnskaberne.

Al data hentes fra Virk.dk

Nyeste version af alle filer findes under mappen *National Asset Project 2023*

Indholdsfortegnelse

01 - Download Virk CVR data	2
02 - Generate CVR-Permanent	3
03 - Generate Ejerregister	4
04 - Download Offentliggørelser	5
05 - Generate Offentliggørelser	6
06 - Download regnskaber	7
07 - Generate Regnskaber	8
08 - Upload to Azure DB.....	11
09 - Supplerende Data	14
10 - Power BI.....	14

01 - Download Virk CVR data (30-35 min)

E:\National Asset Project 2023\01 - Download Virk CVR data

Download de informationsdata for alle virksomheder der findes inde på Virk fra og med år 2012.

Kør curl scriptet:

"curl_query_cvr.bat"

Curl scriptet henter alt data inde fra virk med hjælp fra Query: "curl_query_cvr.json", som sikre at vi kun henter fra 2012 og op:

```
{
  "query": {
    "bool": {
      "filter": [
        {
          "bool": {
            "must": {
              "exists": {
                "field": "Vrvirksomhed.cvrNummer"
              }
            }
          }
        }
      ],
      "should": [
        {
          "range": {
            "Vrvirksomhed.livsforloeb.periode.gyldigtil": { "gte": "2012-01-01" }
          }
        },
        {
          "bool": {
            "must_not": {
              "exists": {
                "field": "Vrvirksomhed.livsforloeb.periode.gyldigtil"
              }
            }
          }
        }
      ],
      "minimum_should_match": 1
    }
  },
  "_source": [ "Vrvirksomhed" ],
  "size": 3000
}
```

Figur 1: curl_query_cvr.json

PROBLEMER: Tidligere har vi haft problemer med syntaks når scriptet "curl_query_cvr.bat" køres:

```
echo off
title Henter Virk CVR-data ud fra curl_query_cvr.json
cd "E:\National Asset Project 2023\01 - Download Virk CVR data"
del VirkAllCVRdata-*.json

curl -XPOST -H "Content-Type: application/json" -o VirkCVRdata-001.json -u Reliasset_CVR_I_SKYEN:522e52b7-32b4-475a-b941-74ef4562b315
http://distribution.virk.dk/cvr-permanent/_search?scroll=5m -d "@curl_query_cvr.json"
```

Screenshot fra curl_query_cvr.bat fil

Her skal anførelsestegn huskes når -d query tilføjes som ses på billedet.

Indsæt output filerne (som har navnet "VirkCVRdata-<SUFFIX>.json") i mappen "Data" hvorefter alle filernes gemmes under ny mappe: "VirkCVRData".

Mappen VirkCVRData kan nu oprettes som database i BaseX (Grundet størrelsen af mappen kan det være nødvendigt at splitte database op i 2: VirkCVRData-1 og VirkCVRData-2)

HVIS: Der eksisterer en database med samme navn skal den gamle database slettes.

PROBLEMER: BaseX skal åbnes igennem bat filen: "**basexgui.bat - Shortcut**" som findes i mappen: E:\National Asset Project 2023. Hvis ikke BaseX åbnes igennem scriptet kan følgende fejl opstå:



1) Åben BaseX igennem bat filen: *"basexgui.bat - Shortcut"*

2) Database/Open and Manage

Drop databases: VirkCVRdata-1 **and** VirkCVRdata-2

3) Database/New

Input file or directory: E:/National Asset Project 2023/Data/VirkCVRData-1/

Name of database: VirkCVRdata-1

Input format: JSON

Parse files in archives: x

Gøres for begge mapper:

Database/New

Input file or directory: E:/National Asset Project 2023/Data/VirkCVRData-2/

Name of database: VirkCVRdata-2

Input format: JSON

Parse files in archives: x

02 - Generate CVR-Permanent

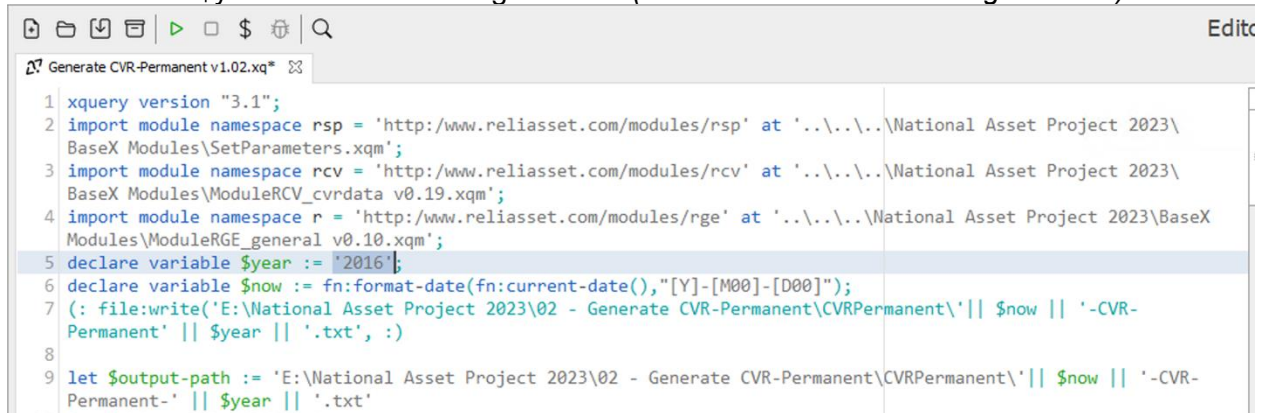
E:\National Asset Project 2023\02 - Generate CVR-Permanent

Baseret på den database der er blevet oprettet i **01** kan vi nu generere txt filer der indeholder basal CVR-information for alle virksomheder pr. år.

1. Åben BaseX igennem bat filen: *"basexgui.bat - Shortcut"*

2. Åben "Generate CVR-Permanent v1.02.xq"

3. Erstat året i \$year til året som skal genereres (Alle år fra 2012-2023 skal genereres).

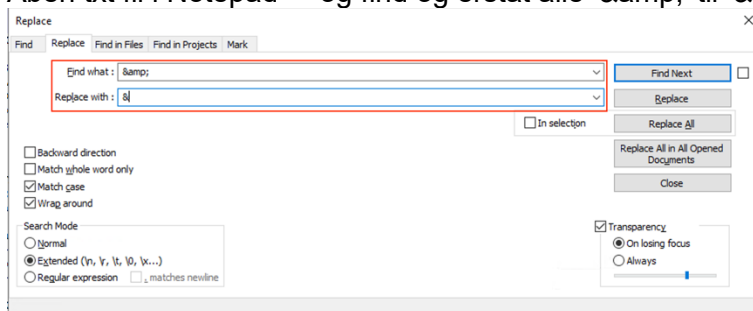


```
1 xquery version "3.1";
2 import module namespace rsp = 'http://www.reliasset.com/modules/rsp' at '..\..\..\National Asset Project 2023\
BaseX Modules\SetParameters.xqm';
3 import module namespace rcv = 'http://www.reliasset.com/modules/rcv' at '..\..\..\National Asset Project 2023\
BaseX Modules\ModuleRCV_cvrdata v0.19.xqm';
4 import module namespace r = 'http://www.reliasset.com/modules/rge' at '..\..\..\National Asset Project 2023\BaseX
Modules\ModuleRGE_general v0.10.xqm';
5 declare variable $year := '2016';
6 declare variable $now := fn:format-date(fn:current-date(),"[Y]-[M00]-[D00]");
7 (: file:write('E:\National Asset Project 2023\02 - Generate CVR-Permanent\CVRPermanent\' || $now || '-CVR-
Permanent' || $year || '.txt', :)
8
9 let $output-path := 'E:\National Asset Project 2023\02 - Generate CVR-Permanent\CVRPermanent\' || $now || '-CVR-
Permanent-' || $year || '.txt'
```

4. Kør BaseX query

5. Output fil findes i *E:\National Asset Project 2023\02 - Generate CVR-Permanent\CVRPermanent* gemt med navnet: *<DATO>-CVR-Permanent-<ÅR>.txt*

Åben txt fil i Notepad++ og find og erstat alle '&' til '&'



6. Gentag for alle år.

03 - Generate Ejerregister

E:\National Asset Project 2023\03 - Generate Ejerregister

Baseret på den database der er blevet oprettet i **01** kan vi nu generere txt filer der indeholder Ejerregister information for alle virksomheder pr. år.

1. Åben BaseX igennem bat filen: *"basexgui.bat - Shortcut"*
2. Åben "Generate Ejerregister v1.03.xq"
3. Erstat året i \$year til året som skal genereres.

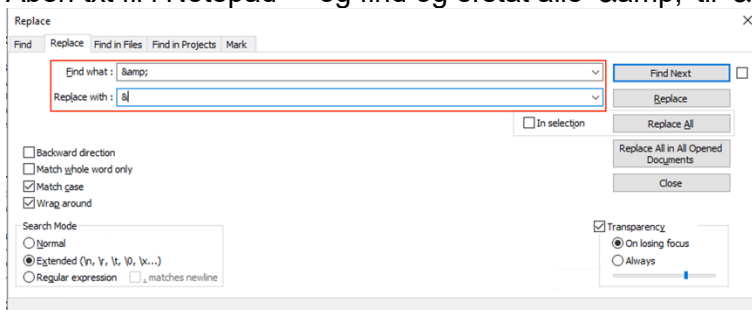


```
1 xquery version "3.1";
2 import module namespace rsp = 'http://www.reliasset.com/modules/rsp' at '..\..\..\National Asset Project 2023\
BaseX Modules\SetParameters.xqm';
3 import module namespace rcv = 'http://www.reliasset.com/modules/rcv' at '..\..\..\National Asset Project 2023\
BaseX Modules\ModuleRCV_cvrdata v0.19.xqm';
4 import module namespace r = 'http://www.reliasset.com/modules/rge' at '..\..\..\National Asset Project 2023\BaseX
Modules\ModuleRGE_general v0.10.xqm';
5 declare variable $year := '2022';
6 declare variable $now := fn:format-date(fn:current-date(), "[Y]-[M00]-[D00]");
7 (: file:write('E:\National Asset Project 2023\02 - Generate CVR-Permanent\CVRPermanent\'|| $now || '-CVR-
PermanentTest2' || $year || '.txt', :)
8
9 let $output-path := 'E:\National Asset Project 2023\03 - Generate Ejerregister\Output\'|| $now || '-Ejerregister-
'|| $year || '.txt'
10 let $dbl := 'VirkCVRdata'
```

4. Kør BaseX query

5. Output fil findes i *E:\National Asset Project 2023\02 - Generate CVR-Permanent\CVRPermanent* gemt med navnet: *<DATO>-Ejerregister-<ÅR>.txt*

Åben txt fil i Notepad++ og find og erstat alle '&' til '&'



6. Gentag for alle år.

04 - Download Offentliggørelser

E:\National Asset Project 2023\04 - Download Offentliggoerelser

Vi downloader nu de offentliggørelser for alle virksomheder der findes inde på Virk fra og med år 2012.

1. Kør "curl_query_offentliggoerelser_all.bat"

2. Indtast offentliggørelsesår

3. Vent på at filerne er downloadet

4. Flyt alle Offentliggoerelser<ÅR>-<SUFFIX>.json til:
"E:\National Asset Project 2023\Data\Offentliggoerelser<ÅR>"
(opret mappen "Ofentliggoerelser<ÅR>" hvis den ikke findes)

5. Åben BaseX og opret ny database:

Database/New

Input file or directory: E:/National Asset Project 2023/Data/Offentliggoerelser<ÅR>/

Name of database: Offentliggoerelser<ÅR>
Input format: JSON
Parse files in archives: x

6. Gentag for alle år

05 - Generate Offentliggoerelser

E:\National Asset Project 2023\05 - Generate Offentliggoerelser

Baseret på den database der er blevet oprettet i **04** kan vi nu generere txt filer der indeholder offentliggørelser information for alle virksomheder pr. år.

1. Åben BaseX igennem bat filen: *"basexgui.bat - Shortcut"*

2. Åben "Generate Offentliggoerelser v1.03.xq"

3. Erstat året i \$year til året som skal genereres



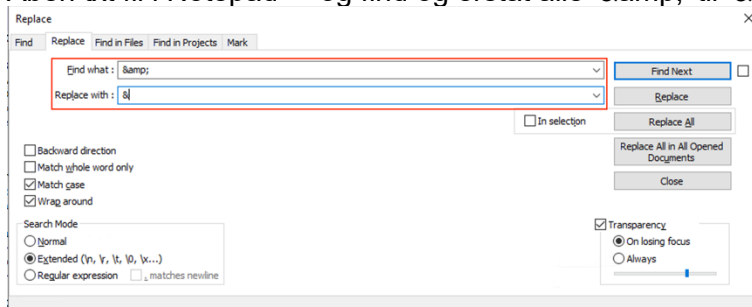
```
1 xquery version "3.1";
2
3 import module namespace rsp = 'http://www.reliasset.com/modules/rsp' at
4 '..\..\..\National Asset Project 2023\BaseX Modules\SetParameters.xqm';
5 import module namespace r = 'http://www.reliasset.com/modules/rge' at '..\..\..\National Asset Project 2023\BaseX
6 Modules\ModuleRGE_general v0.10.xqm';
7 declare variable $year := '2023';
8
9 let $now := fn:format-date(fn:current-date(), "[Y]-[M00]-[D00]");
10 let $output-path :=
11 'E:\National Asset Project 2023\06 - Generate Offentliggoerelser\Offentliggoerelser\' || $now || '-
12 Offentliggoerelse' || $year || '.txt'
```

4. Kør BaseX query

5. Output fil kan findes:

E:\National Asset Project 2023\06 - Generate Offentliggoerelser\Offentliggoerelser
gemt med navnet: <DATO>-Offentliggoerelse<ÅR>.txt

Åben txt fil i Notepad++ og find og erstat alle '&,' til '&'

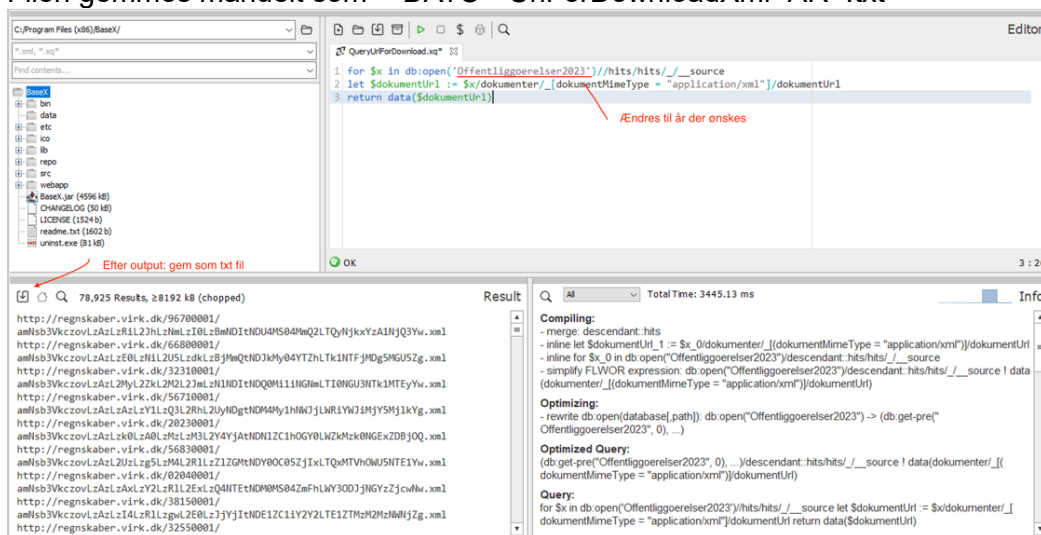


6. Gentag for alle år

06 - Download regnskaber

E:\National Asset Project 2023\06 - Download regnskaber

1. Kør BaseX filen "QueryUrlForDownload.xq" for at hente regnskabs url.
2. Database der skal åbnes (oprettet i **04**) ændres til år ønsket ('Offentliggoerelser<ÅR>')
3. Filen gemmes manuelt som "<DATO>-UrlForDownloadXml<ÅR>.txt"



Herefter kan vi via curl trække alle url ud separat:

4. Kør script:
"curl_download_all.bat"
- a. Input fil gives som følgende:
"<DATO>-UrlForDownloadXml<ÅR>.txt"
- b. Outputfolder skal oprettes for gældende år og gives som output folder i script:
E:\National Asset Project 2023\06 - Download regnskaber\Output\<ÅR>

Herefter oprettes outputfolderen som en database i BaseX med navn: "XBRL<ÅR>":

5. Åben BaseX og opret ny database:

Database/New

Input file or directory: E:\National Asset Project 2023\06 - Download regnskaber\Output\<ÅR>

Name of database: XBRL<ÅR>

Input format: XML

Parse files in archives: x

PROBLEMER: Hvis mappen er for stor til at blive oprettet i databasen splittes mappen op i flere

mapper og gemmes på følgende vis:

XBRL2022-1

XBRL2022-2

XBRL2022-3

XBRL2022-4

For at splitte mappen op kan scriptet, 'split_folder_copy.py' bruges:

```
35 if __name__ == "__main__":
36     # Define source and destination paths
37     source_folder = r"E:\National Asset Project 2023\06 - Download regnskaber\Output\2023"
38     destination_folders = [
39         r"E:\National Asset Project 2023\Data\XBRL2023-1",
40         r"E:\National Asset Project 2023\Data\XBRL2023-2",
41         r"E:\National Asset Project 2023\Data\XBRL2023-3",
42         r"E:\National Asset Project 2023\Data\XBRL2023-4",
43         r"E:\National Asset Project 2023\Data\XBRL2023-5"
44     ]
```

Her ændres 'source_folder' og 'destination_folders' til ønsket.

Scriptet køres på følgende måde fra terminalen (terminal sat til lokationen af python scriptet):

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4780]
(c) Microsoft Corporation. All rights reserved.

E:\National Asset Project 2023\06 - Download regnskaber>python split_folder_copy.py
```

6. Gentag for alle år.

07 - Generate Regnskaber

E:\National Asset Project 2023\07 - Generate Regnskaber

1. Inden Regnskaber kan genereres skal **ExchangeRates2023.xlsx** dokumentet opdateres for gældende år. (Dokumentet er på nuværende tidspunkt opdateret til og med 2023 - 01/03 2024)

Kopier ark fra tidligere år og opdater alle valuta kurser for gældende år fra følgende sider:

<https://www.ofx.com/en-au/forex-news/historical-exchange-rates/yearly-average-rates/>

<https://www.x-rates.com/average/?from=QAR&to=EUR&amount=1&year=2023>

2. Efter at have opdateret valuta kurser kan følgende kolonner kopieres over i "ModuleREX_exchangerate v0.13.xqm" som findes:

E:\National Asset Project 2023\BaseX modules

<pre>else if (\$year = 2023 and \$currOut = "DKK") then (switch (string(\$curr)) case "iso4217:AUD" return (fn:round(data(\$value) * 4.578738) cast as xs:integer) case "iso4217:CAD" return (fn:round(data(\$value) * 5.104454) cast as xs:integer) case "iso4217:CHF" return (fn:round(data(\$value) * 7.670737) cast as xs:integer) case "iso4217:DKK" return (fn:round(data(\$value) * 1) cast as xs:integer) case "iso4217:EEK" return (fn:round(data(\$value) * 0.47585) cast as xs:integer) case "iso4217:EUR" return (fn:round(data(\$value) * 7.456463) cast as xs:integer) case "iso4217:GBP" return (fn:round(data(\$value) * 8.568279) cast as xs:integer) case "iso4217:QAR" return (fn:round(data(\$value) * 1.883864) cast as xs:integer) case "iso4217:SEK" return (fn:round(data(\$value) * 0.649833) cast as xs:integer) case "iso4217:TWD" return (fn:round(data(\$value) * 0.220417) cast as xs:integer) case "iso4217:USD" return (fn:round(data(\$value) * 6.890194) cast as xs:integer) case "iso4217:NOK" return (fn:round(data(\$value) * 0.65293) cast as xs:integer) case "iso4217:ISK" return (fn:round(data(\$value) * 0.04979) cast as xs:integer) case "iso4217:RON" return (fn:round(data(\$value) * 1.514525) cast as xs:integer) default return ("Unknown currency - 2023"))</pre>	<pre>else if (\$year = 2023 and \$currOut = "EUR") then (switch (string(\$curr)) case "iso4217:AUD" return (fn:round(data(\$value) * 0.613903) cast as xs:integer) case "iso4217:CAD" return (fn:round(data(\$value) * 0.684665) cast as xs:integer) case "iso4217:CHF" return (fn:round(data(\$value) * 1.028955) cast as xs:integer) case "iso4217:DKK" return (fn:round(data(\$value) * 0.134113) cast as xs:integer) case "iso4217:EEK" return (fn:round(data(\$value) * 0.06391) cast as xs:integer) case "iso4217:EUR" return (fn:round(data(\$value) * 1) cast as xs:integer) case "iso4217:GBP" return (fn:round(data(\$value) * 1.149226) cast as xs:integer) case "iso4217:QAR" return (fn:round(data(\$value) * 0.253854) cast as xs:integer) case "iso4217:RUB" return (fn:round(data(\$value) * 0.010921) cast as xs:integer) case "iso4217:SEK" return (fn:round(data(\$value) * 0.087132) cast as xs:integer) case "iso4217:TWD" return (fn:round(data(\$value) * 0.029795) cast as xs:integer) case "iso4217:USD" return (fn:round(data(\$value) * 0.924224) cast as xs:integer) case "iso4217:NOK" return (fn:round(data(\$value) * 0.087568) cast as xs:integer) case "iso4217:ISK" return (fn:round(data(\$value) * 0.006596) cast as xs:integer) case "iso4217:RON" return (fn:round(data(\$value) * 0.202072) cast as xs:integer) default return ("Unknown currency - 2023"))</pre>	<pre>else if (\$year = 2023 and \$currOut = "USD") then (switch (string(\$curr)) case "iso4217:AUD" return (fn:round(data(\$value) * 0.664352) cast as xs:integer) case "iso4217:CAD" return (fn:round(data(\$value) * 0.740708) cast as xs:integer) case "iso4217:CHF" return (fn:round(data(\$value) * 1.113583) cast as xs:integer) case "iso4217:DKK" return (fn:round(data(\$value) * 0.145167) cast as xs:integer) case "iso4217:EEK" return (fn:round(data(\$value) * 0.07071) cast as xs:integer) case "iso4217:EUR" return (fn:round(data(\$value) * 1.082245) cast as xs:integer) case "iso4217:GBP" return (fn:round(data(\$value) * 1.243779) cast as xs:integer) case "iso4217:QAR" return (fn:round(data(\$value) * 0.094298) cast as xs:integer) case "iso4217:RUB" return (fn:round(data(\$value) * 0.011809) cast as xs:integer) case "iso4217:SEK" return (fn:round(data(\$value) * 0.094298) cast as xs:integer) case "iso4217:TWD" return (fn:round(data(\$value) * 0.03267) cast as xs:integer) case "iso4217:USD" return (fn:round(data(\$value) * 1) cast as xs:integer) case "iso4217:NOK" return (fn:round(data(\$value) * 0.094761) cast as xs:integer) case "iso4217:ISK" return (fn:round(data(\$value) * 0.007179) cast as xs:integer) case "iso4217:RON" return (fn:round(data(\$value) * 0.218865) cast as xs:integer) default return ("Unknown currency - 2023"))</pre>
---	---	--

ModuleREX_exchangerate v0.13.xqm:

```
Generate Regnskaber v1.3.xqm | ModuleRNS_namespace v0.15.xqm | ModuleREX_exchangerate v0.13.xqm
553 case "iso4217:RON" return (fn:round(data($value) * 1.514525) cast as xs:integer)
554 default return ("Unknown currency - 2023"))
555
556 else if ($year = 2023 and $currOut = "EUR") then (
557   switch (string($curr))
558   case "iso4217:AUD" return (fn:round(data($value) * 0.613903) cast as xs:integer)
559   case "iso4217:CAD" return (fn:round(data($value) * 0.684665) cast as xs:integer)
560   case "iso4217:CHF" return (fn:round(data($value) * 1.028955) cast as xs:integer)
561   case "iso4217:DKK" return (fn:round(data($value) * 0.134113) cast as xs:integer)
562   case "iso4217:EEK" return (fn:round(data($value) * 0.06391) cast as xs:integer)
563   case "iso4217:EUR" return (fn:round(data($value) * 1) cast as xs:integer)
564   case "iso4217:GBP" return (fn:round(data($value) * 1.149226) cast as xs:integer)
565   case "iso4217:QAR" return (fn:round(data($value) * 0.253854) cast as xs:integer)
566   case "iso4217:RUB" return (fn:round(data($value) * 0.010921) cast as xs:integer)
567   case "iso4217:SEK" return (fn:round(data($value) * 0.087132) cast as xs:integer)
568   case "iso4217:TWD" return (fn:round(data($value) * 0.029795) cast as xs:integer)
569   case "iso4217:USD" return (fn:round(data($value) * 0.924224) cast as xs:integer)
570   case "iso4217:NOK" return (fn:round(data($value) * 0.087568) cast as xs:integer)
571   case "iso4217:ISK" return (fn:round(data($value) * 0.006596) cast as xs:integer)
572   case "iso4217:RON" return (fn:round(data($value) * 0.202072) cast as xs:integer)
573   default return ("Unknown currency - 2023"))
574
575 else if ($year = 2023 and $currOut = "USD") then (
576   switch (string($curr))
577   case "iso4217:AUD" return (fn:round(data($value) * 0.664352) cast as xs:integer)
578   case "iso4217:CAD" return (fn:round(data($value) * 0.740708) cast as xs:integer)
579   case "iso4217:CHF" return (fn:round(data($value) * 1.113583) cast as xs:integer)
580   case "iso4217:DKK" return (fn:round(data($value) * 0.145167) cast as xs:integer)
581   case "iso4217:EEK" return (fn:round(data($value) * 0.07071) cast as xs:integer)
582   case "iso4217:EUR" return (fn:round(data($value) * 1.082245) cast as xs:integer)
583   case "iso4217:GBP" return (fn:round(data($value) * 1.243779) cast as xs:integer)
584   case "iso4217:QAR" return (fn:round(data($value) * 0.094298) cast as xs:integer)
585   case "iso4217:RUB" return (fn:round(data($value) * 0.011809) cast as xs:integer)
586   case "iso4217:SEK" return (fn:round(data($value) * 0.094298) cast as xs:integer)
587   case "iso4217:TWD" return (fn:round(data($value) * 0.03267) cast as xs:integer)
588   case "iso4217:USD" return (fn:round(data($value) * 1) cast as xs:integer)
589   case "iso4217:NOK" return (fn:round(data($value) * 0.094761) cast as xs:integer)
590   case "iso4217:ISK" return (fn:round(data($value) * 0.007179) cast as xs:integer)
591   case "iso4217:RON" return (fn:round(data($value) * 0.218865) cast as xs:integer)
592   default return ("Unknown currency - 2023"))
```

Efter exchange rates er opdateret, kan følgende punkter begyndes på:

Baseret på den database der er blevet oprettet i 06 kan vi nu generere txt filer der indeholder regnskaber for alle virksomheder pr. år.

1. Åben BaseX igennem bat filen: "basexgui.bat - Shortcut"

2. Åben "Generate Regnskaber v1.4.xqm"

3. Erstat året i \$year til året som skal genereres

```
10 declare default element namespace "http://www.xbrl.org/2003/instance";
11 declare namespace cvr      = 'http://cvr.dk/report';
12 declare namespace cmn      = "http://xbrl.dcca.dk/cmn";
13 declare namespace fsa      = "http://xbrl.dcca.dk/fsa";
14 declare namespace gsd      = "http://xbrl.dcca.dk/gsd";
15 declare namespace xbrldi    = "http://xbrl.org/2006/xbrldi";
16 declare namespace xbrli     = "http://www.xbrl.org/2003/instance";
17 declare namespace xlink    = "http://www.w3.org/1999/xlink";
18 declare namespace link      = "http://www.xbrl.org/2003/linkbase";
19 declare namespace ifrs      = "http://xbrl.ifrs.org/taxonomy/2011-03-25/ifrs";
20 declare namespace ifrs-dk   = "http://xbrl.dcca.dk/ifrs-dk-cor_2014-12-20";
21 (: declare namespace ifrs-full = "http://xbrl.ifrs.org/taxonomy/2014-03-05/ifrs-full"; :)
22 (: declare namespace ifrs-full="http://xbrl.ifrs.org/taxonomy/2021-03-24/ifrs-full"; :) (:NEW NAMESPACE:)
23
24
25
26 declare variable $year := 2021;
27 declare variable $currOut := 'DKK';
28
29 (: for $year in 2020 to 2022 :)
30 let $now := fn:format-date(fn:current-date(), "[Y]-[M00]-[D00]")
31 let $output-path :=
32   "E:\National Asset Project 2023\07 - Generate Regnskaber\Regnskaber\" || $now || '-Regnskab' || $year || '.txt'
33
34 let $result := (
35 let $xbrlIDBPart := concat('XBRL', $year)
36 let $ext := ('', '-1', '-2', '-3', '-4', '-5', '-6', '-7', '-8', '-9')
```

4. Kør BaseX query

Eksternt script forståelse

BaseX-queryen **Generate Regnskaber v1.4.xq** benytter et eksternt script til at hente CVR-numre for virksomheder, der anvender deres LEI-nummer som ID i deres regnskaber. Dette sikrer, at alle relevante CVR-numre bliver korrekt matchet og inkluderet i analysen.

Et snipped af hvor det eksterne script køres kan ses for neden:

```
let $LEI :=
if (string-length($IdentificationNumberCvrOfReportingEntity) > 8)
then $IdentificationNumberCvrOfReportingEntity
else ()

let $CVR :=
if (string-length($IdentificationNumberCvrOfReportingEntity) <= 8)
then $IdentificationNumberCvrOfReportingEntity
else (
let $input := string($LEI)
let $batchScript := "E:\National Asset Project 2023\07 - Generate Regnskaber\get_cvr_from_lei.bat";
(: Construct the command as a single string :)
let $command := string-join(($batchScript, $input), ' ')
(: Run the batch script using the proc:system function :)
let $result := proc:system($command)
(: Save the output as a variable or process it as needed :)
return fn:normalize-space($result)
)
```

Screenshot fra *Generate Regnskaber v1.4.xq*

Queryen kalder først batchfilen **get_cvr_from_lei.bat**, som derefter eksekverer Python-scriptet **get_cvr_from_lei.py**.

Vejledning til Python-scriptet

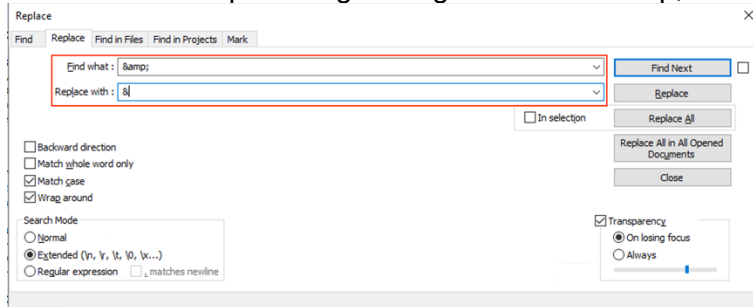
For detaljeret information om, hvordan Python-scriptet fungerer, henvises til den separate vejledning:

'Vejledning til brug af det eksterne Python-script.docx'

5. Output fil kan findes:

E:\National Asset Project 2023\07 - Generate Regnskaber\Regnskaber med navn: <DATO>-Regnskab<ÅR>.txt

Åben txt fil i Notepad++ og find og erstat alle '&' til '&'



5. Gentag for alle år

08 - Upload to Azure DB

E:\National Asset Project 2023\08 - Upload to Azure DB

1. Åben SQL Server Management Studio Management Studio 20

2. Log ind på Azure Databasen

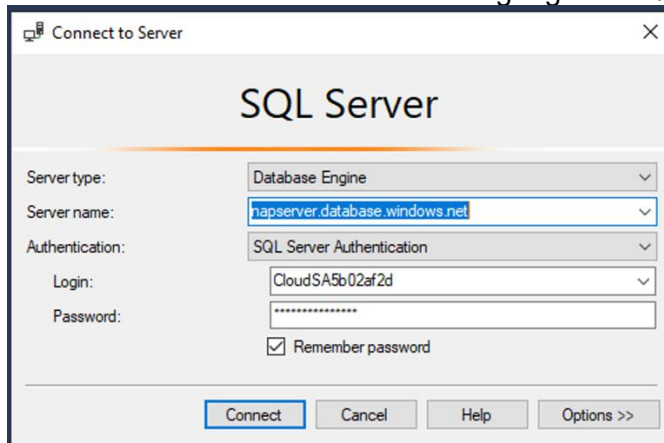
Server type: Database Engine

Server name: napserver.database.windows.net

Authentication: SQL Server Authentication

Login: CloudSA5b02af2d

Password <Standard Adgangskode>#1



3. Klik "Connect"

4. Udvid: Databases

5. Højre klik på NationalAssetProject -> Tasks -> Import flat file

6. Find enten "CVR-PermanentYYYY", "OffentliggørelseYYYY" eller "RegnskabYYYY" (YYYY er år)

7. New table name: fjern dato fra navn

(f.eks. <YYYY-MM-DD>-CVR-Permanent-<ÅR> -> CVR-Permanent-<ÅR>)

8. Table Schema: dbo

9. Modify Columns: Offentliggoerelser:

	Column Name	Data Type	Allow Nulls
▶	cvrNummer	int	<input type="checkbox"/>
	filnavn_xml	nvarchar(MAX)	<input checked="" type="checkbox"/>
	url_xml	nvarchar(MAX)	<input checked="" type="checkbox"/>
	filnavn_pdf	nvarchar(MAX)	<input checked="" type="checkbox"/>
	url_pdf	nvarchar(MAX)	<input checked="" type="checkbox"/>
⚠	slutDato	date	<input type="checkbox"/>
	year	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

CVR-Permanent:

	Column Name	Data Type	Allow Nulls
▶	cvrNummer	bigint	<input type="checkbox"/>
	navn	nvarchar(MAX)	<input checked="" type="checkbox"/>
	virksomhedsStatus	nvarchar(MAX)	<input checked="" type="checkbox"/>
	sammensatStatus	nvarchar(MAX)	<input checked="" type="checkbox"/>
	enhedsNummer	nvarchar(MAX)	<input checked="" type="checkbox"/>
	langBeskrivelse	nvarchar(MAX)	<input checked="" type="checkbox"/>
	virksomhedsformkode	bigint	<input checked="" type="checkbox"/>
	branchekode	bigint	<input checked="" type="checkbox"/>
	branchetekst	nvarchar(MAX)	<input checked="" type="checkbox"/>
	industrycode	nvarchar(MAX)	<input checked="" type="checkbox"/>
	industrydesc	nvarchar(MAX)	<input checked="" type="checkbox"/>
	stiftelsesDato	date	<input checked="" type="checkbox"/>
	AntalInklusivEjere	nvarchar(MAX)	<input checked="" type="checkbox"/>
	AntalAarsvaerk	nvarchar(MAX)	<input checked="" type="checkbox"/>
	NyesteAntalAnsatte	bigint	<input checked="" type="checkbox"/>
	KvartalAntalAnsatte	nvarchar(MAX)	<input checked="" type="checkbox"/>
	landekode	nvarchar(MAX)	<input checked="" type="checkbox"/>
	postnummer	bigint	<input checked="" type="checkbox"/>
	livsforloeb	nvarchar(MAX)	<input checked="" type="checkbox"/>
	livsforloeb_gyldigFra	date	<input checked="" type="checkbox"/>
	livsforloeb_gyldigTil	date	<input checked="" type="checkbox"/>
	status	nvarchar(MAX)	<input checked="" type="checkbox"/>
	statusForrige	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Boersnoteret	nvarchar(MAX)	<input type="checkbox"/>
	year	bigint	<input type="checkbox"/>
⚠	slutDato	date	<input type="checkbox"/>
			<input type="checkbox"/>

Regnskaber:

	Column Name	Data Type	Allow Nulls
🔑	CVR	nvarchar(50)	<input type="checkbox"/>
	LEI	nvarchar(MAX)	<input checked="" type="checkbox"/>
	NameOfReportingEntity	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ReportingPeriodStartDate	date	<input checked="" type="checkbox"/>
🔑	ReportingPeriodEndDate	date	<input type="checkbox"/>
	Revenue	bigint	<input checked="" type="checkbox"/>
	OtherOperatingIncome	bigint	<input checked="" type="checkbox"/>
	GrossResult	bigint	<input checked="" type="checkbox"/>
	ProfitLoss	bigint	<input checked="" type="checkbox"/>
	Depreciation	bigint	<input checked="" type="checkbox"/>
	DepreciationPPE	bigint	<input checked="" type="checkbox"/>
	Assets_EndBal	bigint	<input checked="" type="checkbox"/>
	NoncurrentAssets_EndBal	bigint	<input checked="" type="checkbox"/>
	IntangibleAssets_EndBal	bigint	<input checked="" type="checkbox"/>
	PropertyPlantAndEquip...	bigint	<input checked="" type="checkbox"/>
	InvestmentProperty_End...	bigint	<input checked="" type="checkbox"/>
	BiologicalAssets_EndBal	bigint	<input checked="" type="checkbox"/>
	LongtermInvestmentsAn...	bigint	<input checked="" type="checkbox"/>
	CurrentAssets_EndBal	bigint	<input checked="" type="checkbox"/>
	Inventories_EndBal	bigint	<input checked="" type="checkbox"/>
	Currency	nvarchar(MAX)	<input checked="" type="checkbox"/>
🔑	ConSol	nvarchar(50)	<input type="checkbox"/>
	ConSolAll	nvarchar(MAX)	<input checked="" type="checkbox"/>
	XBRL_DB	nvarchar(MAX)	<input checked="" type="checkbox"/>
🔑	XBRL_Filename	nvarchar(100)	<input type="checkbox"/>
	XBRL_ns	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Correction	nvarchar(MAX)	<input checked="" type="checkbox"/>

Ejerregister:

	Column Name	Data Type	Allow Nulls
🔑	cvrNummer	int	<input type="checkbox"/>
	navn	nvarchar(MAX)	<input checked="" type="checkbox"/>
	enhedsNummer	nvarchar(MAX)	<input type="checkbox"/>
🔑	Ejer_enhedsNummer	nvarchar(100)	<input type="checkbox"/>
	Ejer_navn	nvarchar(MAX)	<input checked="" type="checkbox"/>
	Ejer_Ejerandel	float	<input type="checkbox"/>
	year	int	<input type="checkbox"/>
🔑	slutDato	date	<input type="checkbox"/>
			<input type="checkbox"/>

PROBLEM: Der kan være et potentielt problem med at have mange af de ens felter fra forskellige tabeller i forskellige formater såsom CVR. Dette kan f.eks. ændres ved følgende query:

```
ALTER TABLE dbo.Regnskab
ALTER COLUMN CVR INT;
```


Dog skal man være sikker på at alle felter kan formateres til valgte format ellers vil der blive slettet rækker fra databasen.

10. Efter at have indsat ønsket år kombineres alle årene til en enkelt tabel, via følgende query: (hvis alle årene allerede er i samme tabel og der ønskes at tilføje et ekstra år skal samlet tabel kombineres med det enkelte år):

Eksempel på Offentliggørelse<ÅR> kombination:

```
-- Create the new table 'Offentliggørelser' if it doesn't exist
IF NOT EXISTS (SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_NAME = 'Offentliggørelser')
BEGIN
    CREATE TABLE Offentliggørelser (
        cvrNummer INT,
        filnavn_xml NVARCHAR(MAX),
        url_xml NVARCHAR(MAX),
        filnavn_pdf NVARCHAR(MAX),
        url_pdf NVARCHAR(MAX),
        slutDato DATE,
        [year] INT,
        PRIMARY KEY (cvrNummer, slutDato) -- Define composite primary key
    );
END;

-- Insert data from each source table into the target table
INSERT INTO Offentliggørelser (cvrNummer, filnavn_xml, url_xml, filnavn_pdf, url_pdf, slutDato, [year])
SELECT cvrNummer, filnavn_xml, url_xml, filnavn_pdf, url_pdf, slutDato, [year]
FROM Offentliggørelse2012
UNION ALL
SELECT cvrNummer, filnavn_xml, url_xml, filnavn_pdf, url_pdf, slutDato, [year]
FROM Offentliggørelse<ÅR>
UNION ALL
```

Hvor <ÅR> er det ønskede år

09 - Supplerende Data

Der findes offentlige regnskaber som ikke kan hentes via Virk. Det gælder følgende sektorer:

- Banker
 - Kommuner & Regioner
 - Statslige virksomheder
 - Forsikringsselskaber (64-65)
-
- Banker og Forsikringsselskaber hentes fra Proff via webscraping script.
 - Kommuner, Regioner og Statslige virksomheder hentes manuelt.

10 - Power BI