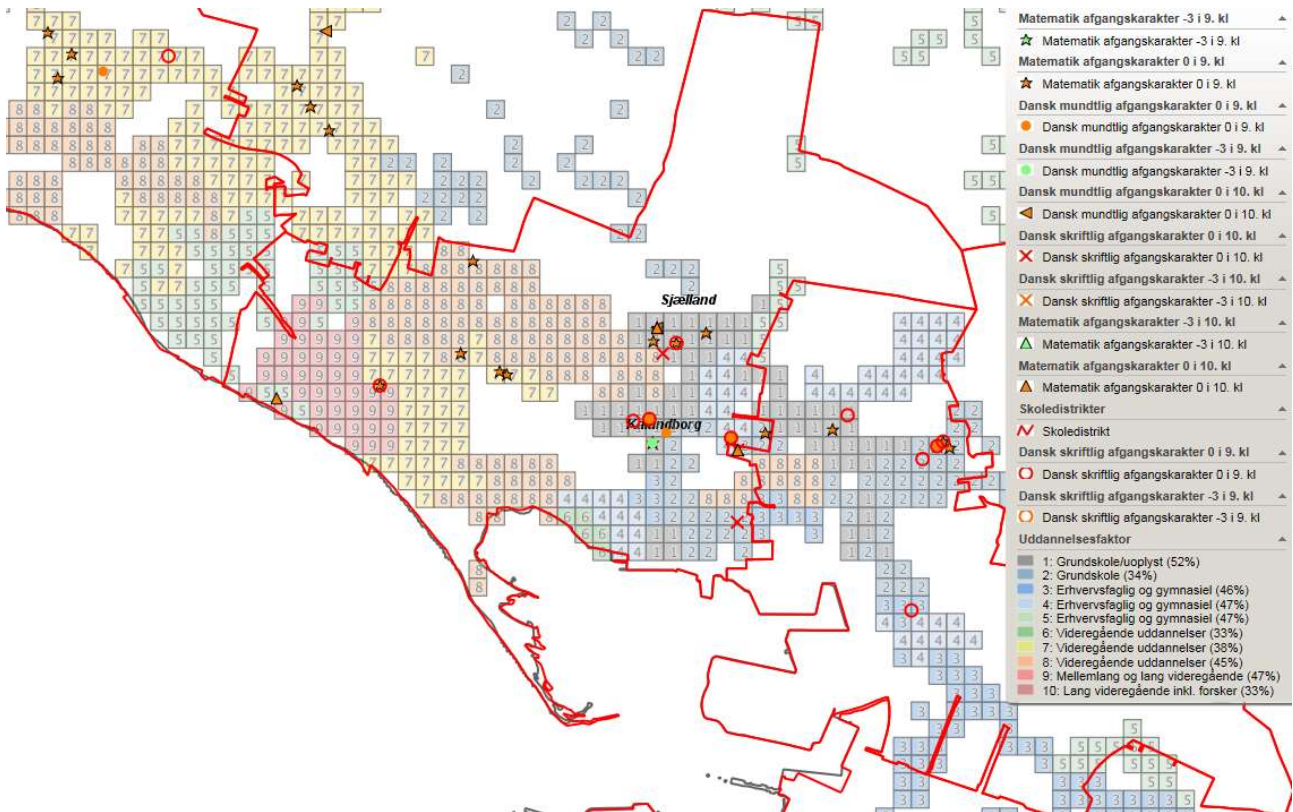


FLIS og GIS

- et GeoSjælland projekt



Marts 2018

Indhold:

1. Indledning
2. Baggrund og formål
3. Konklusion og anbefalinger
4. Metode
5. Afgrænsning af projektet
6. Projektforløb
7. Teknisk dokumentation.
 - A. Hente data fra FLIS
 - B. Udpakke data fra FLIS
 - C. Indlæse data i en database
 - D. Etablere relevante views i databasen
 - E. Præsentere FLIS data i GIS

Rapporten er udarbejdet af:

- Anne Kronby Andersen, Ringsted Kommune
- Anna Slots, Kalundborg Kommune
- Anders Kronland, Kalundborg Kommune
- Karina Nielsen, Næstved Kommune
- Anders Pedersen, Næstved Kommune
- Søren Breddam, Stevns Kommune
- Niels Skov Jensen, GeoSjælland sekretariatet

Forside:

FLIS data suppleret med oplysning om "Uddannelsesfaktor" fra GeoMatics.

1.udgave

Udgivet marts 2018

1. Indledning

I foråret 2016 gennemførte GeoSjælland et forprojekt om FLIS og GIS. Projektets formål var at definere nogle geografiske analyser og præsentationer af FLIS data, som der var efterspørgsel efter i kommunernes organisationer. Det viste sig vanskeligere end forventet at få defineret kommunernes behov for geografiske analyser af FLIS data, men på trods af dette, resulterede forprojektet i beskrivelse af 5 analyser, som de deltagende kommuner pegede, som analyser, der var behov for i deres kommuner. Analyserne var planlagt til gennemførelse i efteråret 2016, men da man nåede frem til efteråret 2016, var projektdeltagerne fra nogle af de deltagende kommuner trådt ud af projektet, og det var ikke muligt at rekruttere de nødvendige faglige kompetencer fra de tilbageværende kommuner til at gennemføre de planlagte analyser.

På GeoSjællands januarseminar 2017 blev problemstillingerne i projektet drøftet, og det blev besluttet at gennemføre et "FLIS og GIS" projekt i en mindre ambitiøs udgave, hvor der blev fokuseret på metoder til automatisering af datahåndteringen i forbindelse med geografiske analyser af FLIS data, og kun i mindre grad på at gennemføre konkrete analyser. Samtidig besluttede man, at projektet skulle koordineres med tilsvarende aktiviteter i geodatasamarbejdet Geo Fyn, således at GeoSjælland projektet kunne bygge oven på de erfaringer, som man allerede havde fået i Geo Fyn projektet. Projektbeskrivelsen af det reviderede GeoSjælland projekt findes i bilag 4.

2. Baggrund og formål

Pr. 1. januar 2017 var 87 kommuner tilsluttet til FLIS. Dette betyder, at mange kommuner har adgang til ledelsesinformation (data), som i FLIS systemet er lagret i en fælles ensartet datastruktur. Der er derfor et stort potentiale for at genanvende metoder og teknikker til analyser og geografiske præsentationer som tager udgangspunkt i disse FLIS data.

Formålet med dette projekt er at tilføre yderligere værdi til FLIS data, ved at udvikle metoder til geografiske analyser og geografiske præsentationer på baggrund af FLIS data. Samtidig er det et mål, at de metoder som udvikles, skal kunne genanvendes af andre kommuner, som ønsker at gennemføre tilsvarende analyser for deres eget geografiske område.

Dette gøres i projektet ved:

- At undersøge forskellige muligheder for at importere data fra FLIS i kommunernes egne GIS systemer. Processen skal være mest muligt automatiseret og standardiseret.
- At undersøge behovet for yderligere data (fra andre datakilder end FLIS), for at gennemføre de ønskede analyser.

3. Konklusion og anbefalinger

Projektet har vist, at det er muligt at importere data direkte fra FLIS i kommunernes eget GIS-system, og at processen i store træk kan automatiseres og standardiseres. Det giver en stor fleksibilitet og mulighed for

individuel tilpasning af analyser, som derudover kan beriges med den geografiske dimension og tilføje yderligere værdi til FLIS-data.

Forudsætningen for at hente FLIS-data direkte efter den undersøgte metode er, at organisationen har ressourcer og kompetencer til at opbygge og vedligeholde metoden.

Projektet har vist, at det er vigtigt med indsigt i FLIS-data for at kunne skabe den ønskede værdi og derfor er et samarbejde med FLIS-fagfolk nødvendigt.

Arbejdet med projektet har givet os:

- Stor fleksibilitet i forhold til analyser
- Bedre dataoverblik, efter at have arbejdet med FLIS-data
- Et præsentationslag til FLIS
- Erfaring med kodning af småprogrammer til at automatisere processer

Projektgruppen anbefaler, at de enkelte kommuner overvejer de interne omkostninger ved den undersøgte metode op mod evt. anskaffelse af en løsning fra en af markedets leverandører.

Afslutningsvis skal det påpeges, at der er behov for en yderligere afklaring af forhold i relation til persondatalovgivningen i forbindelse med præsentationen af analyseresultater på baggrund af FLIS.

4. Metode

Det primære formål med projektet er at udvikle og dokumentere metoder til sammenstilling af FLIS data og GIS data. Dette gør vi i projektet ved at vælge et eksempel på et data set fra FLIS, og vælge et eksempel på en geografisk analyse af de valgte data. Når vi har opsat automatiske processer (scripts) til at håndtere data i det valgte eksempel, kan disse automatiske processer (scriptene) genanvendes på andre dataset ved mindre tilpasninger.

De valgte eksempler:

- Som eksempel på data set fra FLIS har vi valgt at se på "Skolepakken".
- Som eksempel på en konkret geografisk analyse af data fra "Skolepakken" har vi valgt at se på den geografiske fordeling af elever med karakterer mindre end 2 fra 9. klasses afgangsprøve.

5. Afgrænsning

FLIS-data kan indeholde persondata, som skal behandles i overensstemmelse med persondatalovgivningen. Dette kan betyde, at der skal etableres særlige sikkerhedsforanstaltninger i forhold til lagring af de pågældende data og at det kan være nødvendigt, at anonymisere resultatet af de foretagne analyser. Vi har i dette projekt valgt alene beskæftiget os med de tekniske problemstillinger i forbindelse med udtræk og

sammenstilling af FLIS-data. Der er derfor behov for en yderligere afklaring forholdet til persondatalovgivningen, før den udviklede metode kan anvendes til drift i en kommune.

6. Projektforløb

Projektet er gennemført af en projektgruppe bestående af:

- Anne Kronby Andersen, Ringsted Kommune
- Anna Slots, Kalundborg Kommune
- Anders Kronland, Kalundborg Kommune
- Karina Nielsen, Næstved Kommune
- Anders Pedersen, Næstved Kommune
- Søren Breddam, Stevns Kommune
- Niels Skov Jensen, GeoSjælland sekretariatet

Projektgruppen har i perioden maj – december 2017 afholdt 4 projektmøder af 2 timers varighed. Mellem møderne har deltagerne arbejdet med udvikling og test af metoder til udtræk af data fra FLIS og sammenstilling af disse data med data i kommunernes GIS systemer. Derudover har projektgruppen været på besøg hos Svendborg Kommune, for at høre deres om erfaringer med anvendelse af FLIS data i deres GIS.

Besøg hos Svendborg Kommune

Den 1/6 2017 var projektgruppen på besøg hos Svendborg Kommune. Svendborg har spillet en central rolle i udviklingen af et IT-system til sammenstilling af FLIS og GIS data, som er udviklet i samarbejde med firmaerne Septima og Lifa og finansieret af Geo Fyn. I Svendborg Kommune anvender man bl.a. systemet til at identificere socialt udsatte boligområder. Systemet fungerer for så vidt efter hensigten, men brugerne i Svendborg Kommune oplever et behov for større fleksibilitet i opsætningen af systemet, og overvejer p.t., hvordan de kan opnå dette.

Vi har fra Septima fået oplyst, at et system til analyse og præsentation af FLIS-data svarende til det, der er leveret til Geo Fyn kan indkøbes for kr. 35.000 kr. pr. kommune plus 15.000 kr. pr. kommune i årligt driftsudgift. Fra Geo Fyn har vi fået oplyst, at den samlede årlige driftsudgift (til Septima og LIFA) for de fynske kommuner i gennemsnit udgør ca. 30.000 kr. pr. kommune pr. år

Da kommunerne i GeoSjælland ikke umiddelbart har midler i denne størrelsesorden på budgettet, blev det besluttet, at vi i dette projekt ville undersøge andre og billigere metoder til sammenstilling af FLIS- og GIS data.

7. Teknisk dokumentation

For at anvende FLIS data i kommunens GIS, skal data fra FLIS lagres i en database i tilknytning til kommunens GIS. Processen for at gøre dette består af:

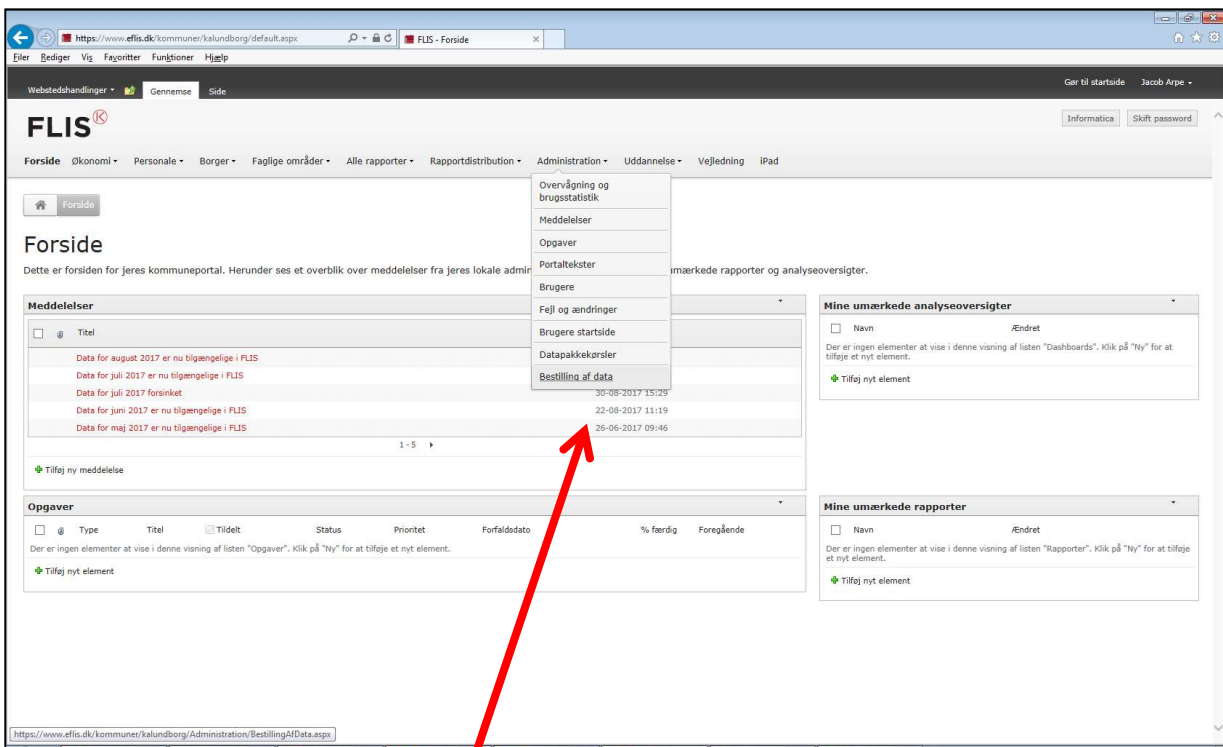
- A. Hente data fra FLIS
- B. Udpakke data
- C. Indlæse data i en database
- D. Etablere relevante views i databasen
- E. Præsentere FLIS data i GIS

I det følgende er beskrevet, hvordan disse processer kan automatiseres ved anvendelse af en række scripts som er udviklet i projektet. Det er ligeledes beskrevet, hvordan scriptene kan tilrettes til de lokale forhold i den enkelte kommune. Selve scriptene findes som bilag til rapporten (jf. bilag 1-3)

A og B: Vejledning til hentning og udpakning af datapakker fra FLIS

Hentning af data fra FLIS forudsætter, at du har adgang til programmet "eflis"¹.

Login på eflis



Vælg Administration og Bestilling af data

¹ eflis er den database hvor FLIS data ligger.

FLIS og GIS – et GeoSjælland projekt

The screenshot shows the FLIS Administration web interface. The main heading is "Bestilling af data". Below it, there is a table of "Datapakkebestillinger". The table has columns for ID, Datakilde, Bestillingstype, Bestillingsstatus, Startdato, Slutdato, and Snitflader. A red circle highlights the "Tilføj nyt element" button located below the table.

ID	Datakilde	Bestillingstype	Bestillingsstatus	Startdato	Slutdato	Snitflader	Datamarter
148	DSA	Engangsbestilling	Udført			KMD ØS - KMD OPUS DATA - Økonomi-KMD-ØS, KMD ØS - KMD OPUS DATA - Registrantkontering, KMD ØS - KMD OPUS DATA - KMD ØS - OPUS LogP Omsættertabel, OPUS -Økonomi - EDW_FIN_BO (Budget), OPUS-Økonomi - EDW_FIN_FP (Forpligtelser), OPUS-Økonomi - EDW_FIN_DTB (Budgetstatistik-væder (Detail-budgetterning)), OPUS-Økonomi - EDW_FIN_DTB (Primo saldo), OPUS-Økonomi - EDW_FIN_DTB (Omsættertabel), OPUS-Økonomi - EDW_FIN_DTB (Budgetstatistik-væder (Detail-budgetterning)), OPUS-Økonomi - EDW_FIN_PCA (Forbrug på statustiske ordre), OPUS-Økonomi - EDW_FIN_SL (Faktiske poster), KMD KLP - KMD OPUS DATA - KLP - Løn og personale, KMD KLP - KMD OPUS DATA - KLP - Fravær ny, KMD Børn og voksne - KMD OPUS DATA - Børn og voksne, OPUS-Løn og personale - EDW_FRAV (Fravær), OPUS-Løn og personale - EDW_LONS (Løn og personale), OPUS-Løn og personale - EDW_NASV (Nærvær), Voksne Handikappede - Data - De nationale dokumentationsprojekter del 3	
275	DSA	Engangsbestilling	Udført			KMD-Elev - KMD-Elev - Elever, KMD-Elev - KMD-Elev - Karakterudtræk, TEA - TEA - Elever, TEA - TEA - Karakterudtræk	
301	DSA	Engangsbestilling	Udført			TEA - TEA - Elever, TEA - TEA - Karakterudtræk	
305	DSA	Engangsbestilling	Udført			KMD ØS - KMD OPUS DATA - Økonomi-KMD-ØS, KMD ØS - KMD OPUS DATA - Registrantkontering, KMD ØS - KMD OPUS DATA - KMD ØS - OPUS LogP Omsættertabel	
366	Datamart	Månedssabonnement		01-10-2017	31-12-2017		Generel, Nøgletal, Borger, Fravær, Løn og personale, Skole, Ældre, Økonomi, Voksne handikappede, Udsatte børn og unge, Økonomi-Borger Aggr.
404	DSA	Engangsbestilling	Processerer			TEA - TEA - Elever, TEA - TEA - Karakterudtræk, Forretningsmæssige referencedata - Skole	
409	DSA	Engangsbestilling	Udført			KMD-Elev - KMD-Elev - Elever, KMD-Elev - KMD-Elev - Karakterudtræk, TEA - TEA - Elever, TEA - TEA - Karakterudtræk, Forretningsmæssige referencedata - Tid, Forretningsmæssige referencedata - Aldersgruppering, Forretningsmæssige referencedata - Skole	

Vælg +Tilføj nyt element

The screenshot shows the "Tilføj nyt element" dialog box. It has a "Rediger" button at the top. Below it are icons for "Gem", "Annuller", "Sæt ind", "Kopier", and "Stavekontrol". The main area contains several fields: "Valgt datakilde" (set to "Datamart"), "Valgt bestillingstype" (set to "Månedssabonnement"), "Startdato" (set to "01-10-2017"), and "Slutdato" (empty). Below these is a section for "Valgte datamarter" with a red warning message "Der skal vælges mindst en datamart". There are several checkboxes for datamarter: "Generel", "Nøgletal", "Borger", "Fravær", "Løn og personale", "Skole", "Ældre", "Økonomi", "Voksne handikappede", "Udsatte børn og unge", and "Økonomi-Borger Aggr.". At the bottom are "Gem" and "Annuller" buttons.

Sæt datakilde til Datamart

Sæt bestillingstype til Månedssabonnement

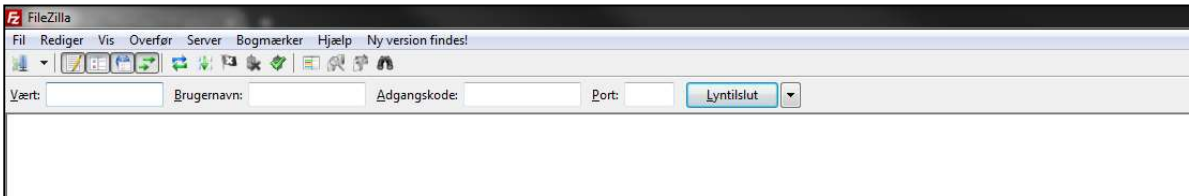
Vælg start dato (og evt. slut dato)

Vælg hvilke datamarter der skal med i datapakken

ADGANG TIL AT HENTE ZIP DATA

Hent FileZilla Client og installere den.

<https://filezilla-project.org/>

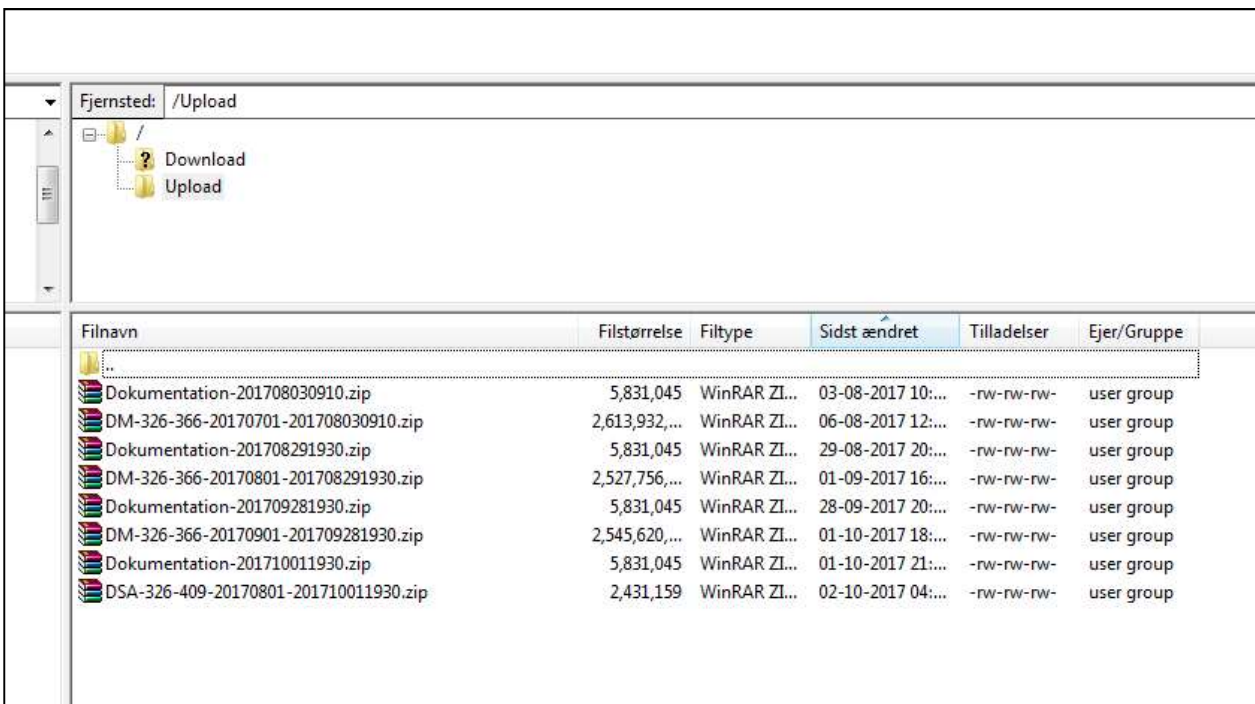


Vært: sftp://data.eflis.dk

Brugernavn: {eflis-login}

Adgangskode: {eflis-kode}

Port: 22



I mappen Upload ligger den valgte kørsel. Datamart er her navngivet som DM-....

Data trækkes nu hen på et valgt drev og overførslen begynder.

C: Vejledning til import af filer til en SQL/PostgreSQL database.

Upload af FLIS-data til SQL/PosgreSQL-databasen består af flere dele: et SQL-script (Bilag 2), en batch-fil (BAT) (Bilag 3) og to PowerShell-scripts (.ps1) (Bilag 1).

SQL-scriptet er en almindelig forespørgsel, der kan uploade data fra en CSV-fil og sætte det ind de prædefinerede tabeller, der er i databasen. Scriptet lægger data i enden af det eksisterende datasæt.

Scriptet i sig selv er relativt enkelt, og det eneste man skal huske at gøre er at ændre stierne, så de passer til jeres placering af data.

```
BULK INSERT import.dbo.FactKarakter

FROM 'C:\Users\tula\PS_FactKarakter.csv'

WITH

(

    FIRSTROW = 2,

    FIELDTERMINATOR = ';',

    ROWTERMINATOR = '\n'

)
```

Eksempel 1: SQL-script, der importerer en CSV-fil til databasen. I dette tilfælde hedder databasen "Import" og tabellen hedder "FactKarakter". CSV-filen ligger på stien "C:\Users\tula\Desktop\FactKarakter.csv". Udskift "dbo" med "public" hvis du bruger en PostgreSQL-database.

For at undgå at skulle åbne databasen for at køre SQL-scriptet, er der lavet en batch-fil der kan åbne databasen og køre scriptet. Man kan i opgavestyring oprette en "Basisopgave", som man kan få til at køre sin batch-fil på bestemte tidspunkter (søg på opgavestyring i "Stifinder", "Task Manager" for dem med engelsk styresystem). Som beskrevet nedenfor kan man også vælge at køre sin batch-fil med samme "trigger" som PowerShell-filen. I koden til batch-filen kan man også angive servernavn, password og brugernavn.

```
osql -U (her indtastes brugernavn) -P (password her) -d (databasenavn) -S (servernavn) -i
"C:\Users\tula\Desktop\Import.sql" (sti til sql-scriptet)

>> "C:\Users\tula\Desktop\Import.sql.log" (sti til hvor loggen skal ligge)
```

Eksempel 2: Batch-filen indeholder, databasenavn (-d), servernavn (-S), input (-i). -U og -P står for henholdsvis login og password. Bemærk at bogstaverne skal stå foran og ikke bagved. Se sidst i opskriften for forklaring af osql. Man kan også bruge sqlcmd, som er et nyere database-værktøj.

Da filerne kommer pakket i en ZIP-fil, er der lavet et PowerShell-script, der automatisk pakker filen ud og dernæst sletter ZIP-filen, man kan alternativt bruge et udpakningsprogram såsom 7-Zip, som kan "times" inde i programmet.

I selve scriptet er der medtaget "Set-Location -Path C:\\" det er kun nødvendigt, hvis din PowerShell ikke starter op fra C-drevet.

Hvis du er i tvivl, så lad den bare stå. Ellers er scriptet ret enkelt, da det kun er de to stier der skal ændres.

```
Set-Location -Path C:\
Expand-Archive -path "C:\Users\tula\Desktop\PS_FactKarakter.zip" -DestinationPath
"C:\Users\tula\Desktop"
```

Eksempel 3: PowerShell-script der udpakker zippede filer. Det er kun stien til den zippede fil og destinationen der skal ændres.

Hvis du ønsker at køre batch-filen lige efter, altså med samme trigger tilføjes:

"start-process" efterfulgt af stien til din batch-fil i slutningen af dit PowerShell-script (se eksempel 4)

```
Set-Location -Path C:\
Expand-Archive -path "C:\Users\tula\Desktop\PS_FactKarakter.zip" -DestinationPath
"C:\Users\tula\Desktop"
start-process "C:\Users\tula\Desktop\Import Bulk 5.bat"
```

Eksempel 4: Kombineret script hvor man kan lade batch-filen blive startet af samme trigger, som PowerShell-scriptet.

Da FLIS-data fylder ret meget er det ikke hensigtsmæssigt at data ligger og fylder, efter det er kørt ind i databasen. Der er derfor lavet et PowerShell-script der sletter gamle filer. Scriptet er konstrueret således, at du selv kan angive hvornår, du vil slette sine filer. Dette gøres i den linje hvor der står \$days = "1". Du kan også angive decimaltal. En halv dag angives for eksempel: \$days = "0.5". For at filerne kan blive slettet, skal du angive hvor de ligger, og det gør du under \$TargetFolder. I scriptet kan du også selv bestemme hvilken type fil der skal slettes. Under \$Extension står der "*". Hvis dette ændres til "*.csv" er det kun csv-filer der slettes og ændres det til "*.zip", er det alle zip-filer der slettes. Hvis du derimod lader den stå som "*" slettes alle filer i mappen.

Da det er rart, og tit også nødvendigt, at kunne holde styr på hvad og hvornår, der slettes noget fra mappen, er der lavet en kode, der giver besked om at filerne er slettede, eller hvis noget er gået galt. Koden genererer en CSV-logfil, men du kan ændre filformatet, hvis du har mod på det. I samme linie ("Export-Csv-path"), skal du huske at angive en placering til din log-fil. Der kommer automatisk et tidsstempel med i loggen, så du kan se hvornår filerne er slettede. Derudover kommer der et stykke tekst, der fortæller hvorvidt filerne er slettede eller ej. Du kan ændre på besked-teksten, hvis du har lyst under "Write-Log-Message". I store træk er det kun nødvendigt at ændre antal dage (\$days), folderen hvor filerne skal slettes fra (\$TargetFolder) og endelig stien til hvor loggen skal ligge (Export-Csv-Path). Husk at gemme som Windows PowerShell-fil (.ps1), når du er færdig med at rette til, brug evt. Notepad++ til dette.

Man kan med fordel kombinere scriptet med det andet script der udpakker og kører batch-filen. Så slipper man for at oprette flere opgaver. Det kan enten kopieres ind lige før eller lige efter. Man skal selvfølgelig tænke lidt over, hvor mange dage filerne må ligge i de to tilfælde.

```

Set-Location -Path C:\
$Now = Get-Date
$days = "1"
$TargetFolder = "C:\Users\tula\Desktop\Testfiler"
$Extension = "*"
$LastWrite = $Now.AddDays(-$Days)
$Files = Get-Childitem $TargetFolder -Include $Extension -Recurse | Where {$_.LastWriteTime -le "$LastWrite"}
foreach ($File in $Files)
{
if ($File -ne $NULL)
{
write-host "Deleting File $File" -ForegroundColor "White"
Remove-Item $File.FullName | out-null
}
else
{
Write-Host "No more files to delete!" -foregroundcolor "Green"
}
}
function Write-Log {
[CmdletBinding()]
param(
[Parameter()]
[ValidateNotNullOrEmpty()]
[string]$Message,
[Parameter()]
[ValidateNotNullOrEmpty()]
[ValidateSet('Information','Warning','Success')]
[string]$Severity = 'Information'
)
[pscustomobject]@{
Time = (Get-Date -f g)
Besked = $Message
Resultat = $Severity
} | Export-Csv -Path "C:\Users\tula\Desktop\Testfiler.csv" -Append -NoTypeInformation
}
$files = $slettede
if ($File -ne $NULL) {
Write-Log -Message 'Dine filer fra FLIS er slettede' -Severity Success
} else {
Write-Log -Message 'Dine filer fra FLIS er ikke slettede' -Severity Warning
}

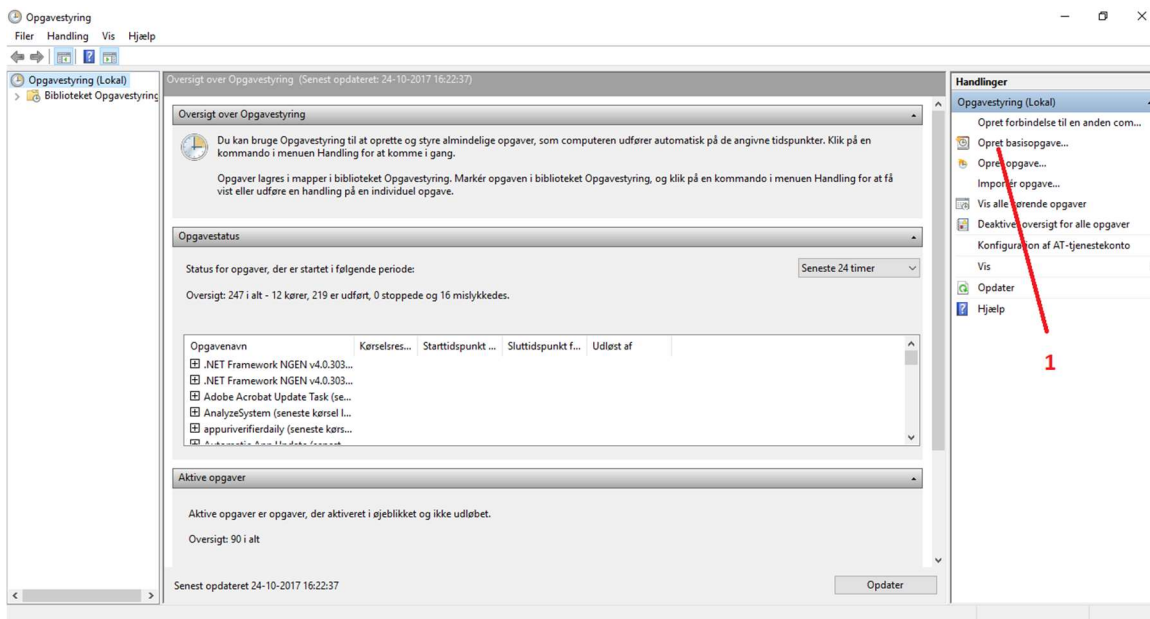
```

Eksempel 5: PowerShell-scriptet, der sletter gamle filer og laver en log. Det ser lidt uoverskueligt ud, men det er ikke meget der skal ændres, før det kan køre lokalt.

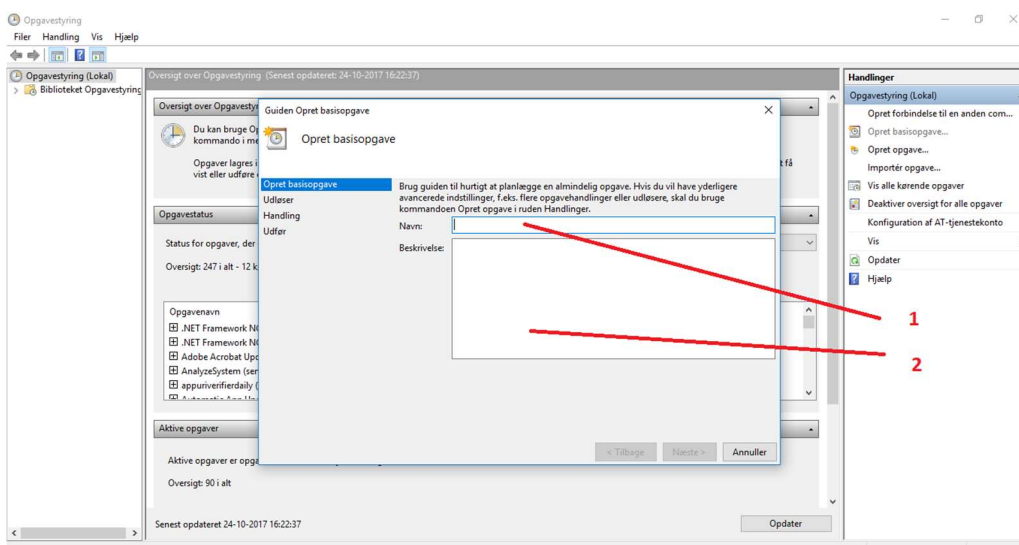
Opgavestyling

Opgavestyling er et program, som er en del af Windows. Opgavestyling kan starte ethvert program på din computer, og du kan angive tidspunkter for automatiske handlinger præcist.

Den nemmeste måde at åbne programmet på er at skrive "Opgavestyling" i startmenuens søgefelt. Når programmet åbner, får du et overblik over værktøjer og aktive opgaver.



Billede 1: Tryk på "opret basisopgave" (1), for at få din batch-fil til at køre på et bestemt tidspunkt.

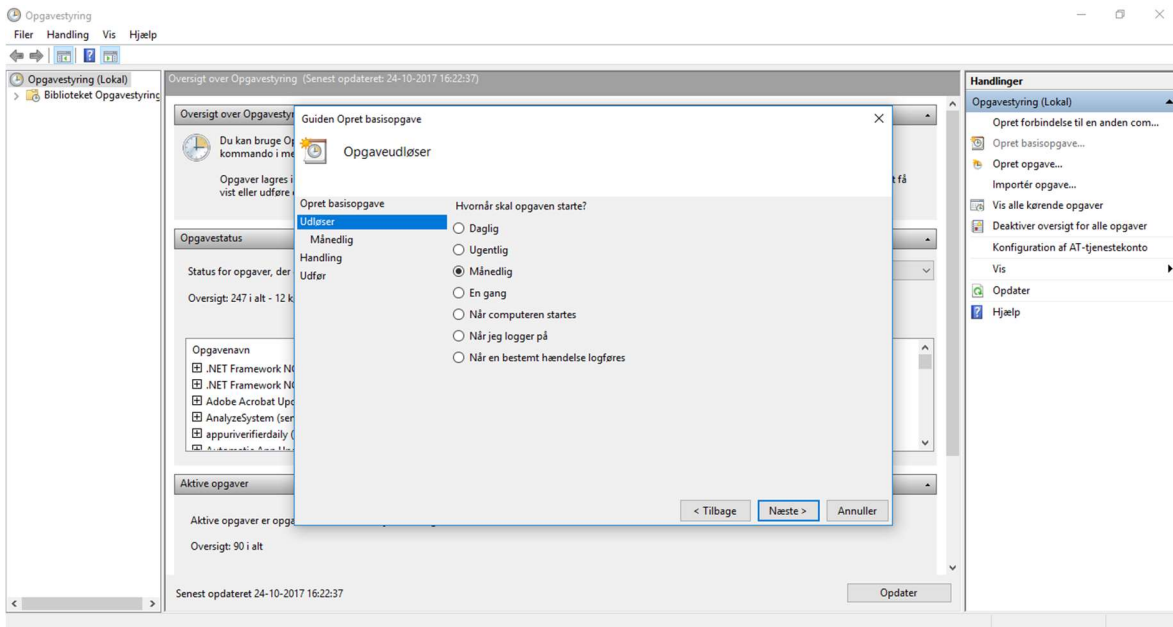


Billede 2: Tilføj et navn (1) og evt. en beskrivelse (2).

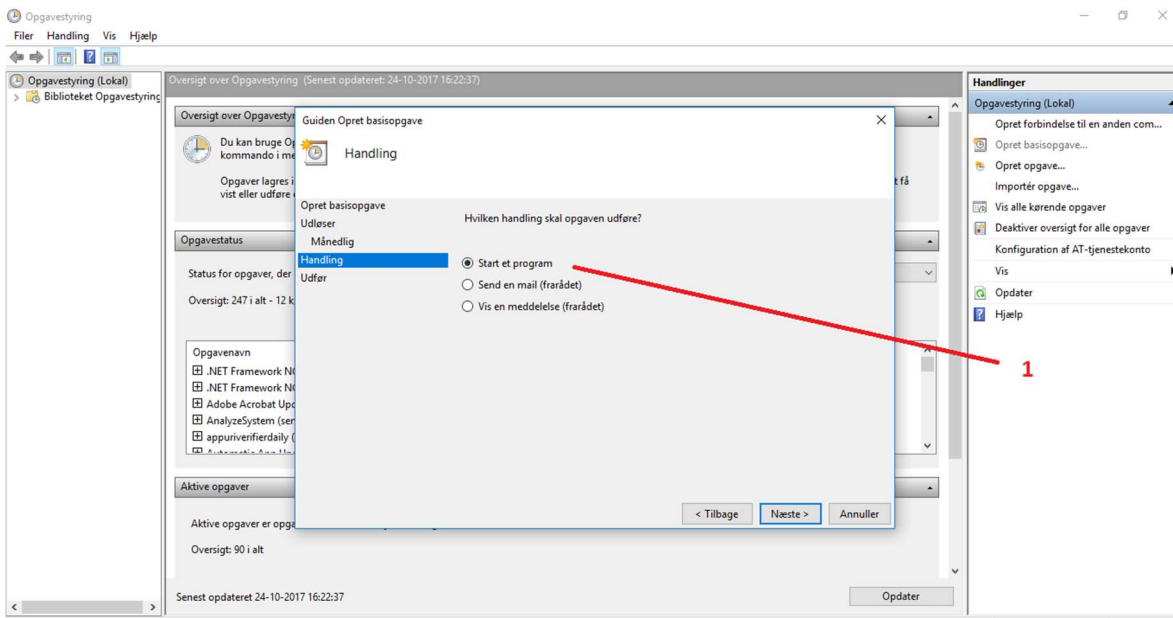
I tilfældet hvor du skal automatisere kørslen af en batch-fil, er fremgangsmåden relativ simpel:

Når du har åbnet opgavestyring, oprettes en basisopgave (tryk på basisopgave, se billede 1)

Der kommer herefter et vindue frem hvor navnet (1) på din opgave, og en eventuel beskrivelse (2) af opgaven kan skrives.



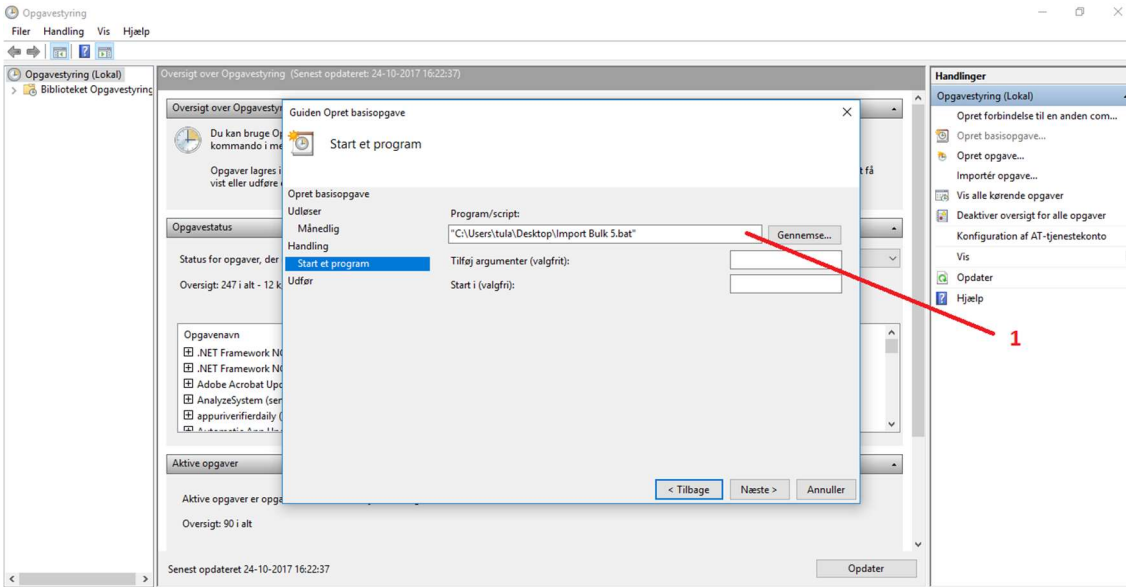
Billede 3: I opgaveudløser-vinduet skal du angive hvor ofte opgaven skal udføres.



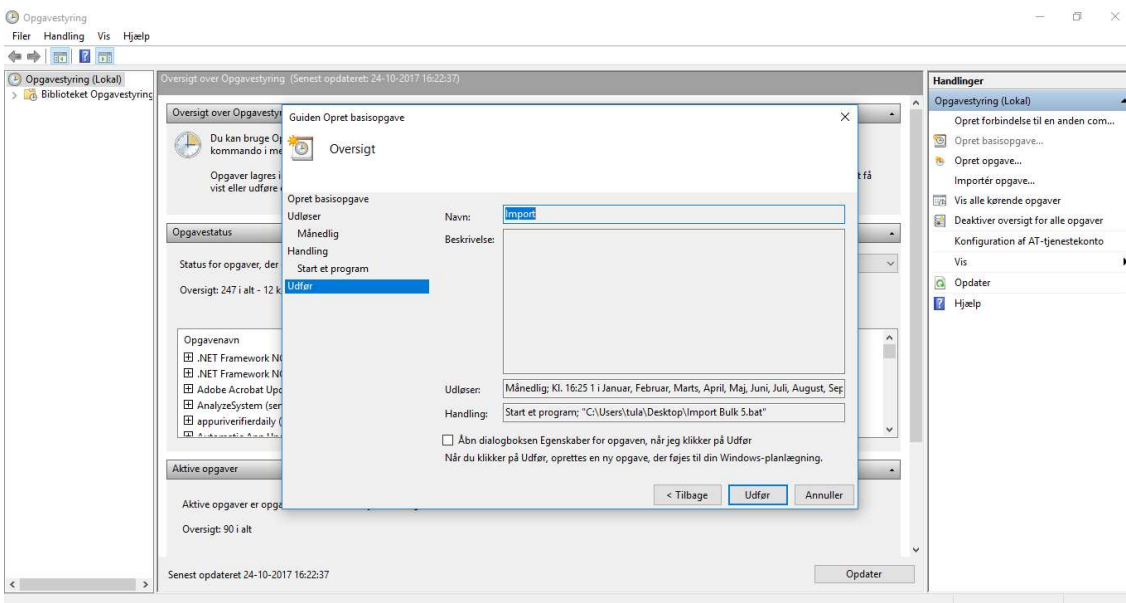
Billede 4: I vinduet handling skal du præcisere hvilken slags handling der skal udføres, i dette tilfælde "start et program" (1).

FLIS og GIS – et GeoSjælland projekt

Herefter skal du få opgaven udført på et bestemt tidspunkt. Dette gøres i opgaveudløser-vinduet. Der er flere muligheder, så det skulle være muligt at finde et passende tidspunkt. Man har ligeledes flere muligheder med hensyn til hvad det er for en opgave man vil sætte i gang. I dette tilfælde er det at starte et program (billede 4 (1)).



Billede 5: I "Start et program" tryk på gennemse (1) og angiv stien til din batch-fil.

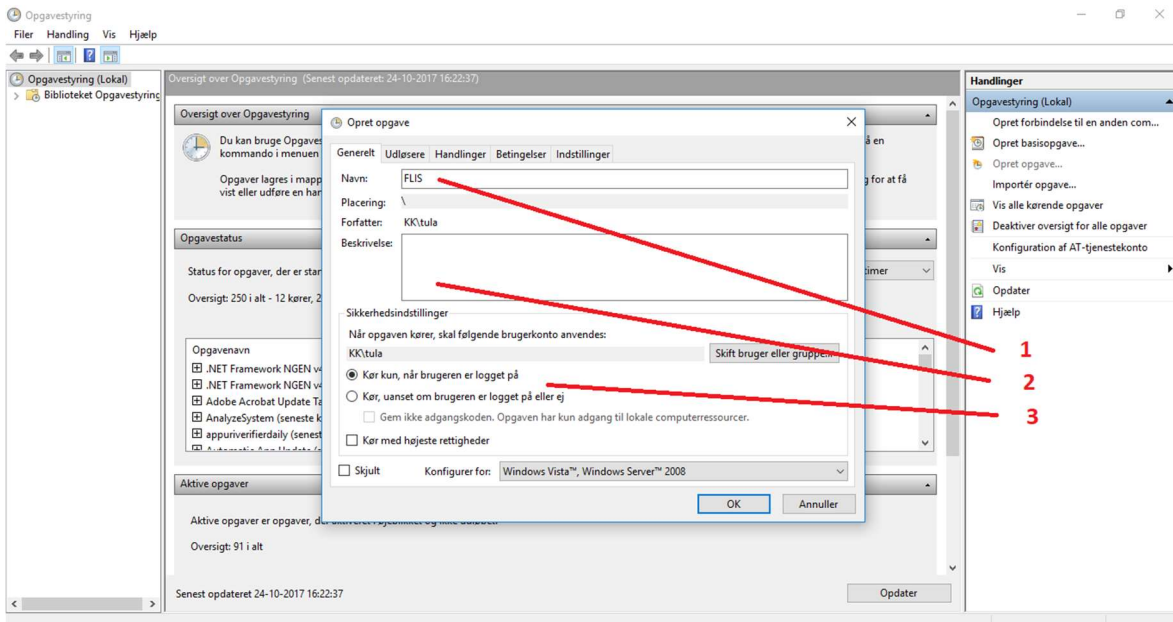


Billede 6: I vinduet "Udfør" kan man tjekke sin handling og sin udløser. Tryk herefter "Udfør".

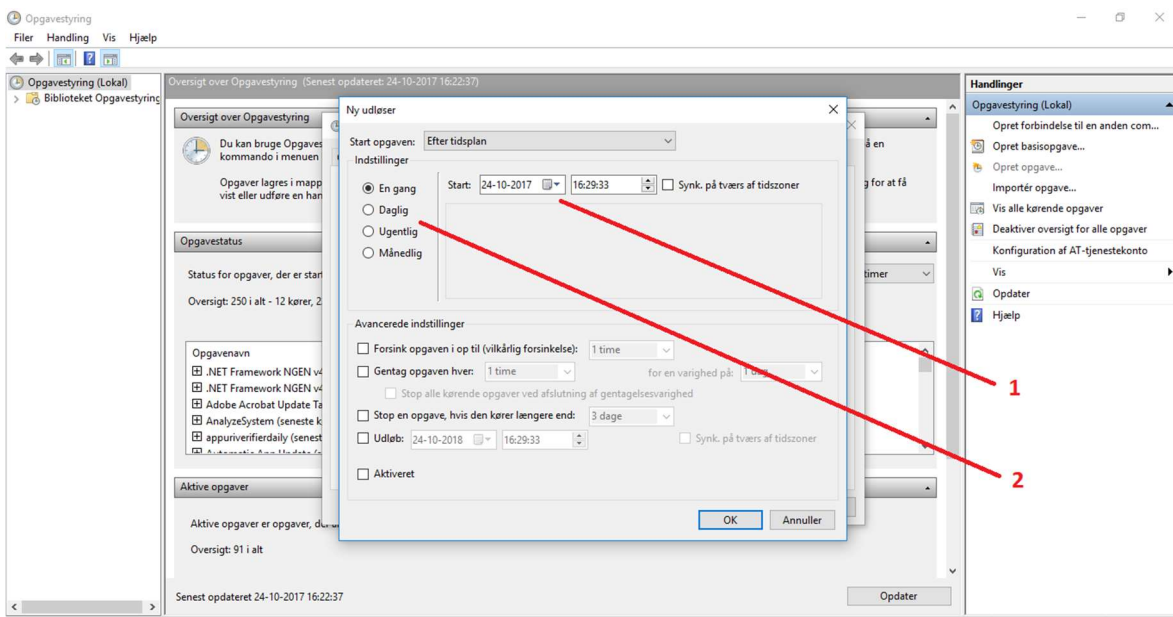
Stien til programmet der skal starte angiver man i næste vindue (Billede 5) ved at trykke på "Gennemse" (Billede 5 (1))

Herefter mangler man blot at trykke "udfør" i det sidste vindue (Billede 6).

Man kan som tidligere nævnt vælge at starte både sin batch-fil og sit PowerShell-script samtidigt, altså som én opgave i "Opgavestyring", hvilket forsimples arbejdsgangen en del. Her kan man ikke nøjes med at oprette en basisopgave. Det kan man heller ikke hvis man skal køre PowerShell-scriptet alene. I "Opgavestyring" vælges "opret opgave" i stedet for "basisopgave".



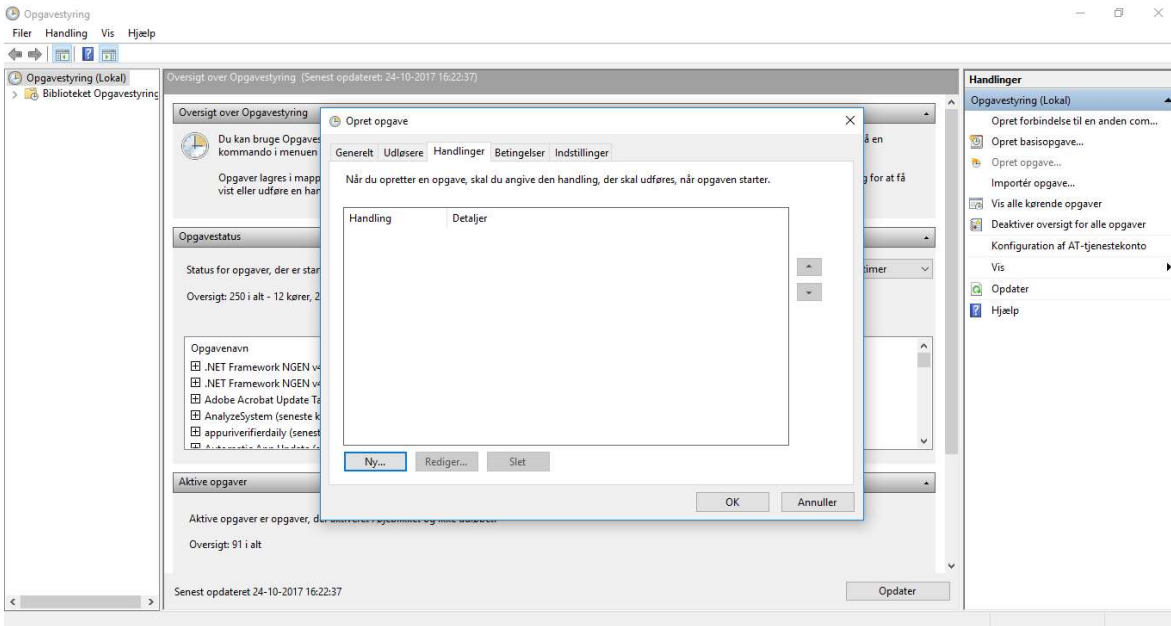
Billede 7: I "opret opgave" angives navnet på opgaven (1), en eventuel beskrivelse (2) og om hvorvidt scriptet skal køre når brugeren er logget på eller ej (3).



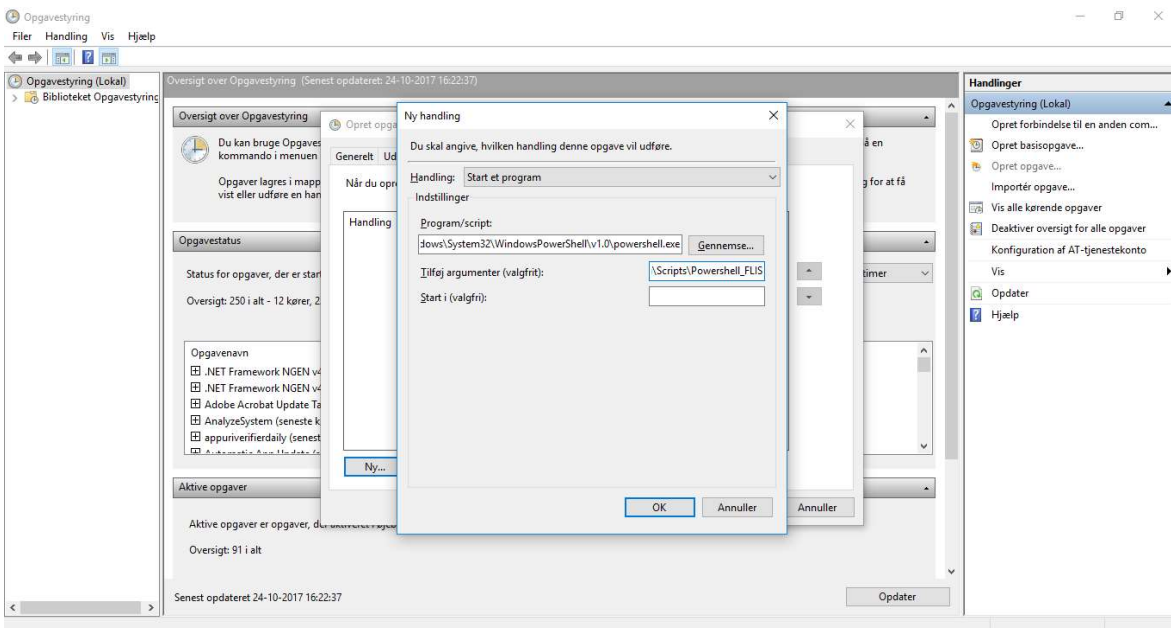
Billede 8: Modsat i "opret basisopgave" kan man her angive et præcist tidspunkt (1) på hvornår opgaven skal udføres og ikke kun om det skal være ugentligt, dagligt osv.

I det næste vindue (billede 7) kan man give opgaven et navn, derudover kan man beskrive sin opgave, hvilken er en god ide, hvis der er andre der skal administrere databasen. Derudover skal man tage stilling om man vil tillade at køre sit script selvom man ikke er logget på.

Modsat ved oprettelsen af en basisopgaven kan man i den lidt mere avancerede version angive



Billede 9: Her skal man, lige som når man opretter en basisopgave, trykke på knappen "Ny".



Billede 10: Den lidt ulogiske del af opgave-vinduet; modsat af hvad man skulle tro skal stien til dit PowerShell ikke angives i "gennemse", men i "tilføj argumenter". I "gennemse" skal stien til PowerShell.exe stå.

et præcist tidspunkt at opgaven skal udføres, hvilket kan være ret smart. Det næste man skal gøre er ligesom tilfældet er for basisopgaven at man skal oprette en ny opgave (Billede 9), men herefter bliver det lidt kryptisk og ulogisk. I modsætning til hvad man skulle tro, skal man ikke angive stien til sit PowerShell-script under "gennemse" (Billede 10). Her skal, som I ovenstående nævnt, stå:

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
```

Dette er stien til PowerShell.exe-filen, i stedet skriver man stien til sit script i "tilføj argumenter". Herefter skal man bare trykke "Ok" og så skulle det fungere

Bemærk:

Disse scripts er testet på en MS-SQL-server, men da PostgreSQL bruger samme sprog, burde der ikke være nogen problemer i f.eks. at køre et rent SQL-script i et sådant miljø. Der skal dog ændres på et par af datatyperne. F.eks. findes "datetime2(3)" ikke i PostgreSQL, den skal erstattes af "Timestamp". Ligeledes skal "dbo" ikke med, men skal erstattes med "public". Der er lavet en række SQL-scripts til oprettelse af database og tabeller. Det er skoleområdet der er blevet lavet scripts til. Alle datatyper er som de skal være (dog skal man huske de beskrevne ændringer hvis man har en PostgreSQL-database), så det eneste man skal gøre er at åbne sin database og køre disse scripts (gem dem ligesom SQL i Notepad++ først). Hvis man vil have tabeller, eller databasen, til at hedde noget andet er disse ret lette at ændre navn på. Det vigtigste, og det mest tidskrævende i disse scripts, er at få datatyperne på plads. Alt dette skal være på plads i den indledende fase, ellers kan man ikke køre de andre scripts som er beskrevet.

OSQL

De to mest udbredte SQL Server tools er osql og sqlcmd. Sidstnævnte kan faktisk alt hvad osql kan og lidt til, da det er et nyere værktøj. Det skal ikke forstås sådan at man så bare skal bruge sqlcmd, brug det du er vant til. I ovenstående er brugt osql, men hvis man har lyst er man velkommen til at prøve med sqlcmd. Som cmd indikerer kører sqlcmd fint i kommandoprompt, og syntaksen i eksempel 2 er den samme, som hvis du brugte osql i kommandoprompt. osql kan også godt åbne en PostgreSQL-server. Man skal være opmærksom på at der kan være forskellige sikkerhedsindstillinger på forskellige servere og det derfor kan kræve et par forsøg/justeringer inden der er hul igennem.

D: Etablere relevante views i databasen

Eksempel på Script_SQL_joins: (der er yderligere eksempler i Bilag 2_sql script)

Her er joins på elev:

```
Select *  
From  
FactElev inner JOIN
```

```
DimBorger ON DimBorger.DimBorgerID = FactElev.DimBorger INNER JOIN
DimBorgerstatus ON DimBorgerstatus.DimBorgerstatusID = FactElev.DimBorgerstatus AND
DimBorgerstatus.ErAktiv = 1 INNER JOIN
DimElev ON DimElev.DimElevID = FactElev.DimElev INNER JOIN
DimLand ON DimLand.DimLandID = FactElev.DimLandStatsborgerskab INNER JOIN
DimSkole ON DimSkole.DimSkoleID = FactElev.DimSkole INNER JOIN
DimSkole ON DimSkole.DimSkoleID = FactElev.DimSkoleSidsteÅr LEFT OUTER JOIN
DimKlasse ON DimKlasse.DimKlasseID = FactElev.DimKlasse INNER JOIN
DimGeografi ON DimGeografi.DimGeografiID = FactElev.DimGeografiBopæl LEFT OUTER JOIN
FactKlasse ON FactKlasse.DimKlasse = FactElev.DimKlasse AND FactKlasse.DimTid = FactElev.DimTid LEFT
OUTER JOIN
LAND_ADRESSEPUNKT ON LAND_ADRESSEPUNKT.KommuneNummer = DimGeografi.Kommunennummer
AND LAND_ADRESSEPUNKT.VejKode = DimGeografi.Vejnummer AND
LAND_ADRESSEPUNKT.HUS_NR = DimGeografi.Husnummer
```

Her er joins på karakter:

```
SELECT *
From
FactKarakter inner JOIN
DIMBorger ON DIMBorger.DimBorgerID = FactKarakter.DimBorger INNER JOIN
DimAlder ON DimAlder.DimAlderID = FactKarakter.DimAlder INNER JOIN
DimPrøveKarakter ON DimPrøveKarakter.DimPrøveKarakterID = FactKarakter.DimPrøveKarakter LEFT JOIN
DimKlasse ON DimKlasse.DimKlasseID = FactKarakter.DimKlasse INNER JOIN
DimLand ON DimLand.DimLandID = FactKarakter.DimLandStatsborgerskab INNER JOIN
Dimskole ON Dimskole.DimskoleID = FactKarakter.DimSkole LEFT JOIN
FactKlasse ON FactKlasse.DimKlasse = FactKarakter.DimKlasse AND FactKlasse.Dimtid = FactKarakter.Dimtid
INNER JOIN
DimGeografi ON DimGeografi.DimGeografiID = FactKarakter.DimGeografiBopæl LEFT JOIN
Land_AdressePunkt ON Land_AdressePunkt.Kommunennummer = DimGeografi.Kommunennummer AND
Land_AdressePunkt.Vejkode = DimGeografi.Vejnummer AND Land_AdressePunkt.HUS_NR =
DimGeografi.Husnummer
```

Det kræver jo lidt mere bearbejdning, men det er bare til en start.

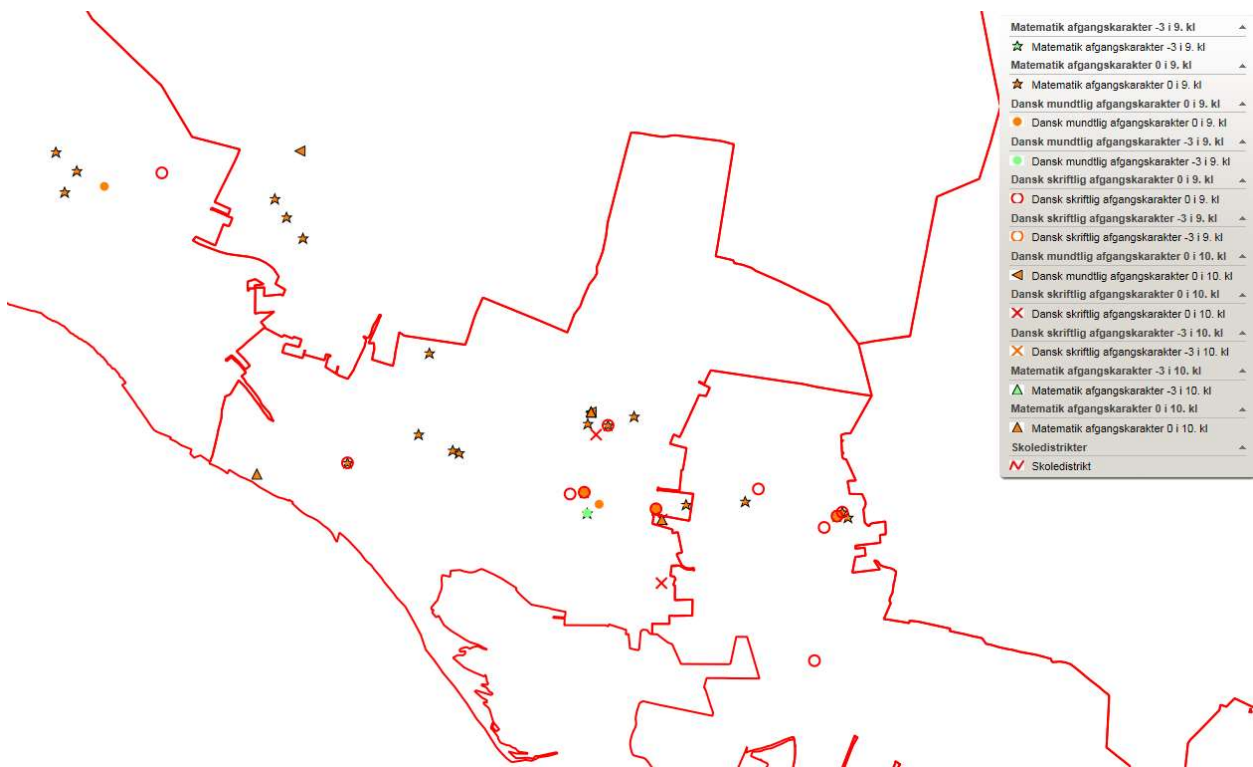
I elev kunne en betingelse se sådan ud:

```
WHERE
FactElev.DimTid =(Select MAX(DimTid) FROM FactElev WHERE (LEFT(DimTid, 4) = YEAR(GETDATE()) - 2))
AND FactElev.DimBorgerstatus != - 1 AND DimSkole.KommunennummerBeliggenhed = 336
```

Her får man elever fra 2015, som går i skole i Stevns Kommune.

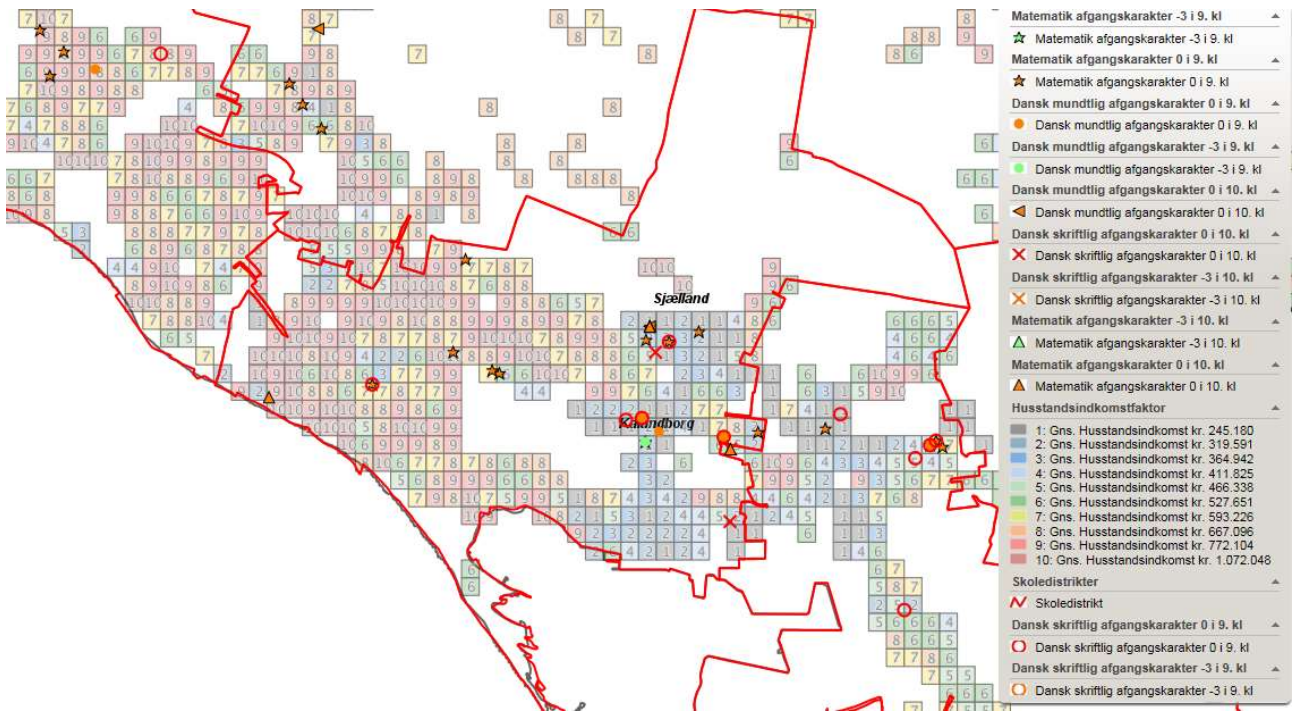
E: Præsentere data i GIS

Efter at FLIS dataene er indlæst i GIS databasen, kan de ønskede geografiske analyser og præsentationer laves. Neden for er vist nogle eksempler på præsentationer af FLIS data fra "Skole pakken" i FLIS sammen med GIS data om "skoledistrikter", "Indkomst" og "Uddannelse":



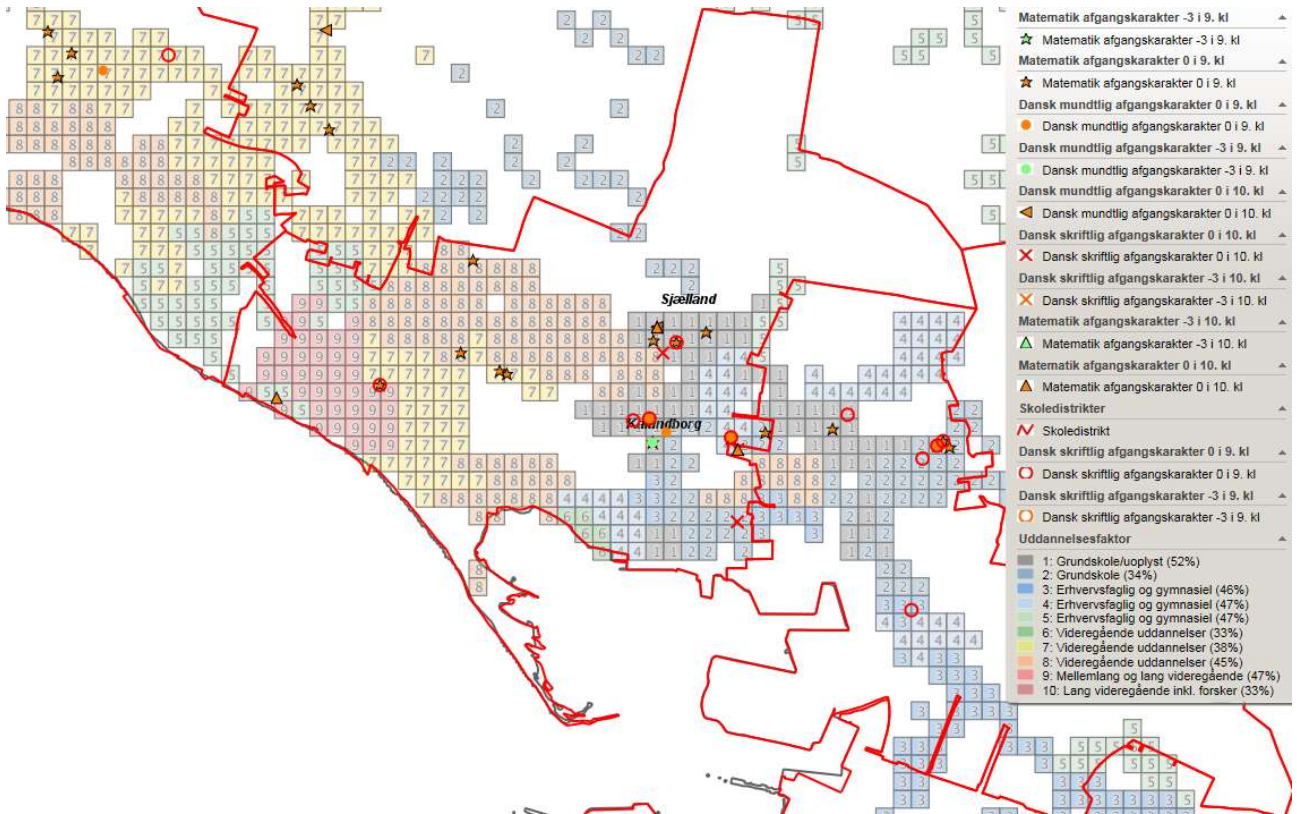
Eks 1: Afgangskarakterer mindre end 2 fra folkeskolens 9. classes afgangsprøve.

FLIS og GIS – et GeoSjælland projekt



Eks 2: Afgangskarakterer mindre end 2 fra folkeskolens 9. classes afgangsprøve, suppleret med oplysning om "Indkomstfaktor" fra GeoMatics.

FLIS og GIS – et GeoSjælland projekt



Eks 3: Afgangskarakterer mindre end 2 fra folkeskolens 9. classes afgangsprøve, suppleret med oplysning om "Uddannelsesfaktor" fra GeoMatics.