Version 1 - 24. 11 2021

**Administratorvejledning til SkadesØkonomi**

Et billede, der indeholder kort

Automatisk genereret beskrivelse

Forfattere

Bo Victor Thomsen, GIS- og IT Specialist, AestasGIS

Per Skougaard Kaspersen, Senior Specialist, LNH water

Kirsten Halsnæs, Professor, Danmarks Tekniske Universitet

Jesper Gaardboe Jensen, Chef Data & Analyse/Geo Fyn, Erhvervshus Fyn

Indhold

[1. Introduktion 3](#_Toc88630252)

[2. Skadesfunktioner 3](#_Toc88630253)

[3. Skades- og risiko-beregning 4](#_Toc88630254)

[4. Modelopbygningen 5](#_Toc88630255)

[4.1. General 6](#_Toc88630256)

[4.2. Queries 6](#_Toc88630257)

[4.3. Data 6](#_Toc88630258)

[4.4. Models 6](#_Toc88630259)

[4.4.1. Generelle modelværdier 7](#_Toc88630260)

[4.4.2. Vej og Trafik 8](#_Toc88630261)

[4.4.3. Offentlig service 9](#_Toc88630262)

[4.4.4. Bygninger 9](#_Toc88630263)

[4.4.5. Kritisk infrastruktur 11](#_Toc88630264)

[4.4.6. Rekreative områder 12](#_Toc88630265)

[4.4.7. Biodiversitet 13](#_Toc88630266)

[4.4.8. Turisme 13](#_Toc88630267)

[4.4.9. Mennesker og helbred 15](#_Toc88630268)

[4.4.10. Industri 16](#_Toc88630269)

[5. Sådan gennemfører du en beregning 16](#_Toc88630270)

[6. Cells – opsummering af beregninger i celler 17](#_Toc88630271)

[6.1 Sådan opsummeres til celler under fanen Cells 17](#_Toc88630272)

[6.1.1. Generer cellelag 18](#_Toc88630273)

[6.1.2. Lav beregning for cellelag 18](#_Toc88630274)

[7. Reports 20](#_Toc88630275)

[Referencer 21](#_Toc88630276)

[Appendix 1 – Skadesfunktioner 22](#_Toc88630277)

[Appendix 2 – BBR-kategorisering 23](#_Toc88630278)

# Introduktion

Dette dokument er en vedledning for GIS administratorer til opsætning af datagrundlag og modeller for QGIS-plugin SkadesØkonomi. Der findes en tilsvarende vejledning ”Brugervejledning til plugin Skadesøkonomi”, som beskriver:

* De generelle muligheder systemet tilbyder
* Hvilke modeller, der pt. findes i systemet
* Hvorledes modelberegningerne aktiveres samt tilpasning af diverse inddataparametre

Det anbefales, at man som administrator har læst denne vejledning først for at få et grundlæggende kendskab til Plugin’ets virkemåde og udseende.

Administrator af plugin ”Skadesøkonomi” skal varetage en række opgaver før plugin’et fungerer hensigtsmæssigt

* Systemet skal konfigureres til at benytte de rigtige database. Se afsnit 4, ”Parameter tabel”
* Data skal tilvejebringes.   
  Før systemet han fungere skal relevante data importeres ind i Skadeøkonomi databasen. Dette arbejde beskrives i afsnit 5, ”Import af data til databasen”
* Plugin opsætning skal tilpasses de nye data. Fra installationen af plugin forventes at daa har et bestemt udseende, dvs. tabeller hedder noget specifikt og er placeret i bestemte schemaer i systemet database. Er dette ikke tilfældet efter import af data, skal systemet konfigureres til de nye tabeller/feltnavne. Se afsnit 6, ”Tilpasning af parameterdata vedr. tabeller og felter”
* Plugin kan udvides med nye modeller. Dette beskrives i afsnit 7, ”Udvikling af nye modeller”
* Databasetypen kan skiftes fra PostgreSQL til andre spatielle databasetyper supporteret af QGIS. Dette beskrives i afsnit 8, ”Skift af database teknologi”

# **Historie**

Som beskrevet i brugervejledningen er grundlaget for Plugin Skadesøkonomi en række modeller, som beregner økonomiske og andre konsekvenser ved forskellige typer af oversvømmelser.

Det originale projekt var et sæt af Python - scripts beregnet til udførelse via ESRI ArcGIS. Disse script fungerede ufhængigt af hinanden og benyttede kortlag vist i ArcGis samt brugerindtastede eller brugervalgte parameterværdier som grundlag for beregningerne. Resultatet var typisk et nyt kortlag inkl. et sæt tilhørende alfanumriske oplysninger.

Ved gennemgang af de originale scripts blev det afklaret, hvorledes disse scripts generelt set fungerede:

1. Brugeren valgte to lag:  
   * Et lag som indeholdt oversvømmelses polygoner med vandybde som alfanumrisk information
   * Et lag som indeholdt de objekter som var grundlaget for modellen, f.eks. bygninger, veje, infrastrukr, industri osv.
2. Bruger indtastede eller valgte en række øvrige parametre, som indgik i den valgte modelberegning. Parametre kunne være minimum vanddybde, valg af oversvømmelsestype, antal timer med oversvømmelse o.lign.
3. Herefter blev modelberegningen gennemført. De i ArcGis indbyggede faciliteter blev benyttet til at foretage en overlapsanalyse mellem laget med oversvømmelsespolygoner og laget med iteresse objekter. Og for hvert oversvømmet objekt blev der udregnet en række værdier, eksempelvis skadeberegninger og værditab, baseret på øvrige parametre samt vanddybde. Ved afslutning blev resultatet præsenteret som et nyt kortlag.

# Idé grundlag for det nye system

Det grundlæggende ønske var at erstatte de originale ArcGis baserede scripts med et Open-Source baseret system, som kunne fungere på basis af et frit og gratis GIS produkt, i dette tilfælde QGIS.

Endvidere var ønsket, at det skulle være muligt at modificere eksisterende og/eller tilføje nye modeller - i bedste fald uden at skulle programmere i Python.

Gennemgangen af de originale scripts viste, at det var muligt at samle/importere alle relevante datasæt til en spatiel SQL database og derefter benytte SQL til at gennemføre modelberegningerne.

Så følgende løsningsmetode blev udformet:

* Alle datasæt importeres til en spatiel database. Ved datasæt forstås eks. oversvømmelsesdata, alle datasæt med interesseobjekter (bygninger, veje, infrafrastruktur osv.) samt lookupdata til brug for de forskellige beregninger
* Alle model beregninger defineres som SQL udtryk som benytter de tabeller, som forefindes i databasen.
* Systemet designes således det er muligt at tilføje nye modeller - uden at skulle ændre i plugin’ets kildetekst - ved at tilføje et nyt SQL udtryk som beregner modelværdier. Og evt. importere ekstra data som nye tabeller
* Alle opsætnings parametre – SQL forespørgsler, hjælpetekster, søgeparametre osv. til plugin gemmes som tekst strenge i selve databasen i en administrations tabel kaldet parameter tabellen. Dette muliggør at flytte hele systemet fra en pc til en anden blot ved at lave en sikkehedskopi af databasen og installere denne på en ny pc sammen med selv plugin’et.
* Alle model SQL udtryk “generaliseres” således at benyttede tabel- og feltnavne samt konstanter erstattes af et ”parameternavn” (se eksempel)
* De generaliserede SQL udtryk, samt alle parameternavne/-værdier for hhv tabeller, felter og søgeværdier gemmes alle i parametertabellen, således det er muligt at genskabe et reelt SQL udtryk kun vha. informationer i parametertabellen.

Et eksempel:

Opgave: ”Find alle bygninger, som bliver oversvømmet, men udeluk vanddybder mindre end 0.2 m. Værditabet for bygningen findes ved at gange den gennemsnitlige bygnings-kvadratmeterpris for kommunen med 0.1 (dvs. 10%)”  
  
Der benyttes tre tabeller:

Tabel “bygninger” indeholder bygningsdata med felterne “kom\_kode” (kommune kode), “geom” (bygningens geometri) og “fid” (En entydig nøgleværdi for bygningen)   
  
Tabel “oversvoem” indeholder oversvømmelses polygoner med felterne: “geom” (oversvømmelsespolygon) og “vanddybde” (vanddybde for polygonen)   
  
Tabel “kvmpris” indeholder oplysninger om gennemsnitlige bygnings-kvadratmeterpris for kommuner. Med felterne “kom\_kode” (kommunekode) og “kvm\_pris” (bygnings-kvadratmeter pris)

NB! Nedenstående SQL udtryk er en simplificeret udgave af systemet faktiske forespørgsel vedr. skadesberegning og værditab på bygninger.

SELECT DISTINCT   
 b.fid,  
 b.kom\_kode,  
 st\_area(b.geom) \* k.kvm\_pris \* 0.1 AS vaerdi\_tab  
FROM data.bygninger b   
INNER JOIN data.oversvoem o ON st\_intersects(b.geom, o.geom)   
LEFT JOIN admin.kvmpris k ON b.kom\_kode = k.kom\_kode  
WHERE o.vanddybde > 0.2

For det ovenstående SQL udtryk gælder følgende begrænsninger:

* Tabellerne skal hedde bestemte navne
* Tabellerne skal placeres i bestemte schemaer.
* Felterne skal hedde bestemte navne
* Konstantværdier for hhv. vanddybde og værditab kan ikke ændres.

For at gøre SQL udtrykkene mere fleksible ”generaliseres” udtrykket til følgende:

SELECT DISTINCT   
 b.{f\_pkid\_t\_building},  
 b.{f\_mun\_code\_t\_building},  
 st\_area(b.{f\_geom\_t\_building}) \* k.{f\_sqm\_price\_t\_mun\_sqmprice} \* {p\_loss\_value} AS vaerdi\_tab  
FROM {t\_building} b   
INNER JOIN {t\_flood} o ON st\_intersects(b.{f\_geom\_t\_building}, o.{f\_geom\_t\_flood})   
LEFT JOIN {t\_mun\_sqmprice} k ON b.{f\_mun\_code\_t\_building} = k.{f\_mun\_code\_t\_flood}  
WHERE o.{f\_water\_depth\_t\_flood} > {p\_water\_depth}

* Alle tabel- og feltnavne samt konstant værdier i SQL udtrykket erstattes af parameternavne, som f.eks. “{f\_pkid\_t\_building}”. Det modificerede SQL udtryk gives også et navn, f.eks. “q\_building\_flood\_loss” og gemmes i parametertabellen.
* Parameternavne/værdier for hhv. tabeller, felter og konstanter gemmes også i parametertabellen. For disse parametre gemmes endvidere tilhørsforholdet til det generaliserede SQL udtryk.

Få at udføre en modelkørsel valgt af brugeren foretager plugin’et følgende:

* Det generaliserede SQL udtryk findes i parameter tabellens via brugervalget.
* Alle øvrige tokens, som tilhører det generaliserede SQL udtryk findes også i parameter tabellen.
* I teksten for udtrykket foretages “søg og erstat” som finder parameternavne og erstatter disse med de tilsvarende parameterværdier.
* SQL udtrykket - som nu indeholder korrekte tabel- og feltnavne samt rigtige søgeværdier udføres i databasen
* Resultatet af søgningen vises i QGIS’s kortvindue.

Metoden giver mulighed for at GIS administrator kan importere tabeller med vilkårlige tabel- og feltnavne til skadeøkonomi databasen og derefter tilpasse de til modellen tilhørende tokenværdier med relevante tabel/feltnavne. Tabellerne skal dog indeholder alle relevante kolonner som type- og værdimæssigt svarer til de original kolonner.

Efter GIS administrator har foretaget overstående konvertering til generaliseret SQL med div. parameternavne har GIS brugeren - via plugin’et - mulighed for at vælge en bestemt model, ændre på modellens søgeværdier og udføre modelkørslen uden at have kendskab til hverken SQL eller modellens udformning.

# Parameter tabel

Parameter tabellen en den centrale informationstabel for plugin Skadeøkonomi.

Den indeholder *alle* nødvendige oplysninger til at beskrive og udføre de forskellige modeller, bl.a. modellernes generaliserede SQL udtryk samt parameter navne/værdier til oversættelse af parametriserede tabeller, felter, søge værdier osv.

Tabellen indeholder endvidere en lang række andre oplysninger nødvendige til at generere brugerflader til de forskellige faneblade i plugin. Tabellen indeholder derfor en række ekstra kolonner udover kolonner til navne og værdier. De forskellige kolonner og deres indhold beskrives senere i dette kapitel.

Graphical user interface, text, application, email

Description automatically generated

Figur 4‑1. Fanebladet General.

Placering og navn på parametertabel i databasen angives af administrator i fanebladet ”General” vha. to felter, henholdsvis ”Database” og ”Parameter tabel”. Hvis fanebladet ”General” ikke vises, kan man trykke på knap ”Administrator” nederst til højre i plugin brugerfladen for at gøre fanebladet synligt.

I feltet ”Database” vælges forbindelsen til den PostgreSQL/PostGIS database, hvor model data er placeret i. Hvis den ønskede database forbindelse ikke findes i valglisten, kan man umiddelbart opsætte den ønskede forbindelse via QGIS ”Data Source Manager” og opdatere valglisten ved at trykke på knappen umiddelbart til højre for valglisten.

I feltet Parameter table angives det samlede database navn for tabellen inkl. schema- og tabelnavn, f.eks. ”fdc\_admin.parametre”.

Ved tryk på knappen umiddelbart til højre for indtastningsfelte for ”Parameter tabel” vil tabbelen blive vist i QGIS som et ikke-spatielt lag i lagviser.

## Opbygning af hieraki vha. “parent” feltet

Alle poster i parametertabellen vises i brugerdialogen i fanebladene ”General”, ”Query”, ”Data”, ”Models” og ”Reports”. Disse faneblade indeholder en visuel træstruktur, som viser dels parameter navn, dels parameterværdi. Når bruger flytter musen hen over en specif navn eller værdi vises indholdet af et andet kolonne i parameter tabellen ”explanation” i et lille pop-up vindue.

Placering i faneblade og placering indenfor de enekelte faneblad stres af data i parametertabellen, som arrangeres i et hieraki, hvor stort set alle parameterposter blivet ”ejet” af en anden post i tabellen. Dette hierarki dannes af de to felter ”name” og ”parent” hvor ”name” indeholder parameternavnet og ”parent” indeholder parameternavnet på den (anden) post, som ejer posten.

Udtrykt som en ER-relation: Der findes en 1:n relation mellem ”parent” og ”name” i samme tabel.

Hvert faneblad repræsenteres af en gruppepost (parameter type “G”) i parametertabellen med samme navn som fanebladet. Disse gruppeposter fungerer som “rod” for hvert faneblad. Alle parameterposter - og deres underposter - som har en af disse “rod”-poster som ”parent” placeres i det tilsvarende faneblad.

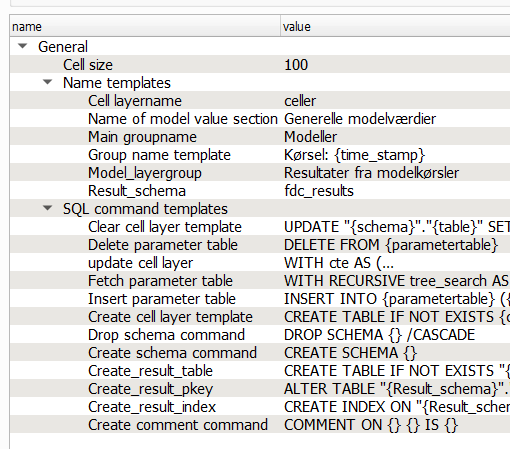
## Struktur og feltbeskrivelse for parameter tabel

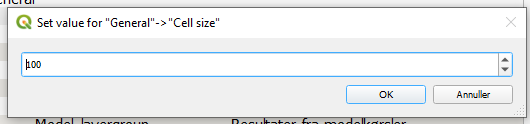
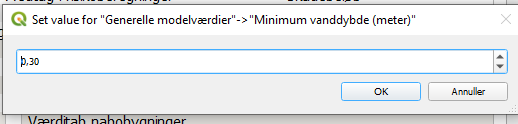
Nedenstående tabel beskriver alle kolonner i parametertabellen.

| Feltnavn | Type | Forklaring |
| --- | --- | --- |
| name | tekst | Navn for parameter; skal være unikt i hele tabellen |
| parent | tekst | Navn på en anden parameterpost, som “ejer” denne parameterpost. Hvis posten ikke ejes af nogen andre poster, efterlades den blank. |
| value | tekst | Værdi af paramater; alle værdier er præsenteret som en tekststreng uanset type.  Heltal skrives som “123456”, Reelle tal som “12345.678”. Der er et frit antal decimaler. |
| type | karakter | Parameter type; bestemmer hvilken type data, som posten kan indeholde. Felt “type” kan have een af følgende værdier: “G”, “I”, “O”, “P” eller “T”; se afsnit “Brug af “type” feltet for detailjeret forklaring, |
| minval | tekst | For type lig med “I” (heltal) eller “R” (Reelt tal): Den mindste værdi, som felt “value” må indeholde |
| maxval | tekst | For type lig med “I” (heltal) eller “R” (Reelt tal): Den største værdi, som felt “value” må indeholde |
| lookupvalues | tekst | For type = I” eller “R” (Reelt tal): Step værdi i indtastningsboks ved værdirettelser;  For type = “O” (valgliste): Liste over mulige værdier for felt “value” i combobox adskilt af tegnet “¤” (Shift-4) |
| default | tekst | Hvis parameter repræsenterer et modelnavn, skal feltet indeholde navnet på den bagved liggende SQL query |
| explanation | tekst | Tekst, som beskriver parameterens formål; vises I pop-up dialog, når markøren placeres over parameter-navn eller værdifelt i fanebladet. |
| sort | tekst | Sorteringsrækkefølge indenfor en gruppe; mindste tal øverst |
| checkable | karakter | Hvis sat til “T”, vil præsentationen af parameter navn/værdi i fanebladet også vise et afkrydsningsfelt |

## Brug af “type” feltet

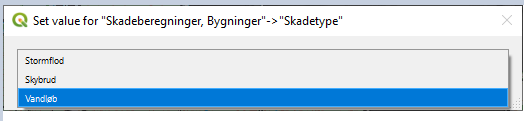
“type” feltet bestemmer dels hvad der kan opbevares i “value” feltet; dels hvilken visuel udseende parameteren har. “type” kan være en af følgende værdier:

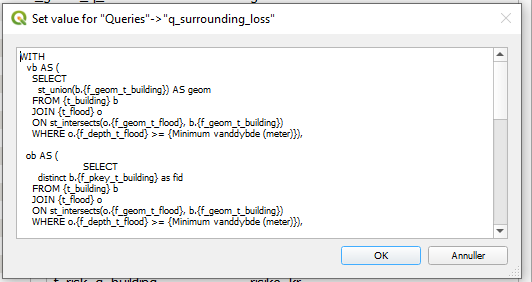
* “G”:   
  Parameter benyttes som en gruppering, dvs. opdeling af parameterposter i logiske sammenhænge. Den indeholder ingen værdi, men parameter navnet vises i fanebladenes træstruktur. Parametre, som skal vises under denne gruppe skal tildeles gruppe parameterens navn som værdi i felt “parent”.
* Eksempel: I faneblad “Generelt” findes en parameterpost “Name templates”. Parametrene “Cell layername” “Group name template”, “Main groupname” o.a. har alle fået tildelt “Name templates” som “parent”. Det giver følgende resultat:

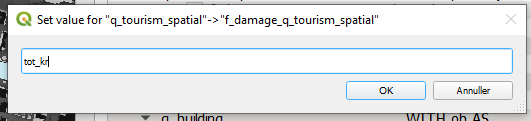
* “I”:   
  Parameter skal indeholde et heltal. Når man dobbeltklikker på paramternavnet eller værdien i fanebladet, vil der vises en bruger dialog, hvor værdien kan ændres.   
    
  Dialogen bliver udformet, således der kun kan indtastes et heltal. Minimum og maksimum værdier bestemmes af værdierne i hhv. felterne “minval” og “maxval”. Stepstørrelsen i det tilhørende rullefelt bestemmes af værdien i felt “lookupvalues|”
* Eksempel: Parameter “Cell size” har type “I”, værdi 100, minval: 10, maxval: 1000, lookvalue: 50. I dialogen vises værdien 100, som kan ændres fra 10 til 1000. Og bruges scroll-pilene ændres værdien i skridt af 50.  
    
    
    
    
  
* “R”:   
  Parameter skal indeholde et reelt tal. Når man dobbeltklikker på paramternavnet eller værdi feltet i fanebladet, vil der vises en bruger dialog, hvor værdien kan ændres.   
    
  Dialogen bliver udformet, således der kun kan indtastes et reelle tal. Minimum og maksimum værdier bestemmes af værdierne indtastet i felterne “minval” og “maxval”. Stepstørrelsen i det tilhørende rullefelt bestemmes af værdien i felt “lookupvalues|”

* “O”:

Parameter skal indeholde een bestemt værdi udvalgt fra en forudbestemt liste. Listen er værdisat i felt “lookupvalues” hvor den optræder som en tekst, som er sammensat af de mulige værdier adskilt af tegnet “¤” (SHIFT-4 på et dansk tastatur) . Ved dobbeltklik på enten navn eller værdifelt i fanebladene vil der vises en valgboks, som indeholder elementerne fra valglisten.

* Eksempel: Parameter “Skadetype”, har type “O” og lookupvalues er sat til “*Stormflod¤Skybrud¤Vandløb*”. Ved dobbeltklik vises en comboboks, som indeholder de 3 valgmuligheder.
* “P”:   
  Paramter skal indeholde en multilinje tekst. Der er ingen begrænsninger i antal linjer eller linjernes længde. Ved dobbeltklik på navn eller værdifelt vises en multiline tekstboks med den nuv. tekst.



* “T”:   
  Paramter kan indeholde en enkelt linje tekst. Der er ingen begrænsninger linjens længde. Ved dobbeltklik på navn eller værdifelt vises en tekstboks med den nuv. tekst.

## Retningslinjer vedrørende navngivning af parametre

De fleste (men ikke alle) navne i parameter tabellen kan angives med en valgfri værdi. De skal dog være entydige i tabellen.   
  
Men det anbefales *kraftigt* at overholde følgende konventioner ved tilføjelse af nye modeller, forespørgsler, tabeller og felter:

* Parameternavne for *tabeller* starter med “t\_”. Resten af navnet reflekterer funktionen af tabellen. Ved navngivning af parametre for de oprindelige 11 modeller er det valgt at lade funktionsbeskrivelsen være på engelsk - for at markere disse data som “noget GIS administarorer behandler”.

| * Token navn | * Peger på |
| --- | --- |
| * t\_bioscore | * Bioscore tabel |
| * t\_build\_usage | * Bygningsanvendelse (administrationstabel) |
| * t\_building | * Bygningstabel |
| * t\_company | * Firma tabel |
| * t\_damage | * Skadefunktioner (administrations tabel) |
| * t\_flood | * Oversvømmelses tabel |
| * t\_human\_health | * Oplysninger om beboere |
| * t\_infrastructure | * Infrastruktur tabel |
| * t\_publicservice | * Public service områder |
| * t\_recreative | * Rekreative områder |
| * t\_road\_traffic | * Vejdata tabel |
| * t\_sqmprice | * Gennemsnitlig kvm.pris fordelt på kommuner (administrations tabel) |
| * t\_tourism | * Turisme tabel |

* Parameternavne for *generaliserede SQL forespørgsler* starter med “q\_”, efterfulgt af query funktion på engelsk.
* Eksempel: “q\_building”: Skademodel fro bygninger. Generelt set følger alle query-navne samme mønster som for tabeller.
* Parameternavne for felter til tabeller og forespørgsler starter med “f\_”, efterfulgt af en funktionsbeskrivelse for feltet og *afsluttes* med token navn for den tabel/query, som feltet tilhører.
* Der findes pt. følgede funktionsbeskrivelser for felter:

| Start på parameternavn | * Funktion |
| --- | --- |
| f\_pkey\_ | * Feltet er primary-key felt |
| f\_geom\_ | * Feltet er geometry felt |
| * f\_usage\_code\_ | * Feltet indeholder en bbr-kode |
| * f\_category\_ | * Feltet indeholder en hovedkategori for bygningsanvendelse |
| * f\_risk\_ | * Feltet indeholder et beregnet risiko beløb |
| * f\_loss\_ | * Feltet indeholder et beregnet værditab |
| * f\_damage\_ | * Feltet indeholder et beregnet skadesbeløb |
| * f\_muncode\_ | * Feltet indeholder en kommunekode |

* Øvrige parameternavne for felter bør beskrive feltets formål/funktion på engelsk.

De ovenstående regler er *anbefalinger* ikke *absolutte krav* - undtaget følgende: Angivelser for hhv. primary key og geometri felter for *queries* *skal* følgen formlerne; ellers vil systemt fejle ved genereringen af resultat tabeller.

# Import af data til databasen

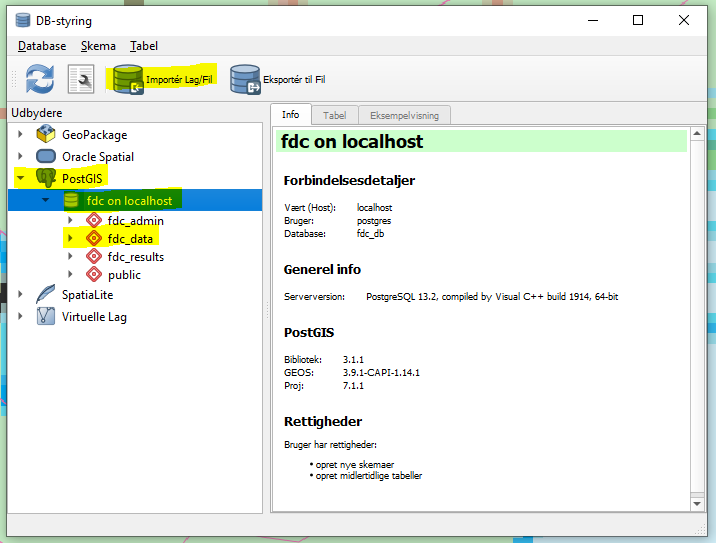
Efter installation af databasesystem; oprettelse af database; kørsel af installationsscript; installation af plugin og opsætning af forbindelse mellem QGIS og databaseserver er det næste skridt af gøre data tilgængelig for modelgenereringen.

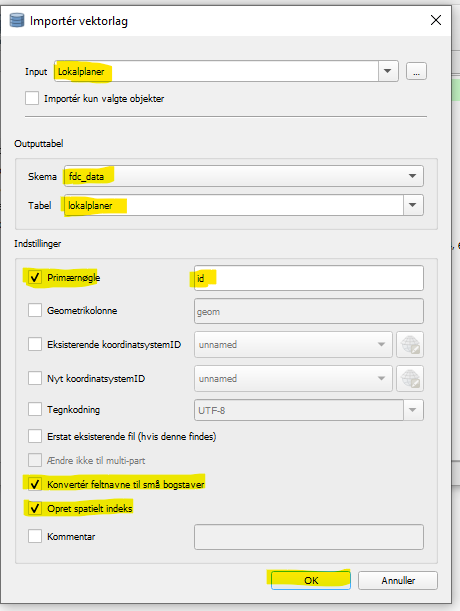
Første punkt er at sikre sig, at alle relevante data til de ønskede modeludregninger forefindes, dvs.

* At de forskellige vektor data findes. Det kan være fil data – MapInfo data, Shapefiler; data fra andre databaser eller evt. WFS datakilder. Alle vektorbaserede datakilder kan benyttes
* At de enkelte datakilder indeholder de relevante kolonner med data nødvendige for modeludregningen.
* Hvis en datakilde indeholder både multipart og singlepart objekter, skal denne konverteres således alle objekte bliver multipart. Se QGIS processing: ”Konvertér til multipart”

Hvilke datakilder, som er nødvendige for en given modeludregning samt hvilke informationer den enkelte datakilde skal indeholde kan læses i afsnit 6 ”Tilpasning af parameterdata vedr. tabeller og felter”

Da data til modelgenerering skal være placeret i databasen, skal man importere data fra QGIS til databasen. Dette gøres på følgende måde:

* I QGIS åbnes de relevante datakilder som lag vha QGIS datavælger.
* QGIS’s Database Manager startes op: Menulinje punkt : ”Database” → ”DB-styring”. Der vises følgende brugerdialog:  
  

* Under Udbydere udfoldes ”PostGIS” ved at klikke på ”PostGIS”
* I listen over de forskellige databaser udfoldes den relevante database med samme frbindelsesnavn der blev angivet ved opsætning af adgang til parametertabel (Afsnit 4, ”Parametertabel”). I eksemplet er navnet på forbindelsen ”fdc on localhost”
* Der klikkes på knap ”Importér Lag/Fil”. Følgende bruger dialog vises:  
  

* I valgfelt ”Input” vælges det lag, der ønskes importeret til databasen
* I valgfelt ”Skema” vælges det schema i databasen, hvor det ønskes at tabelle skal placeres.
* I valgfelt ”Tabel” skriver/rettes indholædet til det ønskede tabelnavn. Det anbefales, at navnet kun indeholder små bogstaver, ikke æøå, blanktegn eller specialtegn.
* Der sættes ”flueben” i felt Primærnøgle og feltnavnet for laget unikke id kolonne skrives i indtastningsfeltet. Hvis datakilden ikke har en unik nøgle efterlades standard navnet ”id”
* Der sættes ”flueben” i ”Konvertér feltnavne til småbogstaver”. Dette medfører at feltnavn i databasetabellen kun indeholder små bogstaver og alle blanktegn og specialtegn fjernes.
* Der sættes ”flueben” i ”Opret spatielt indeks”. Dette betyder, at spatiel navigation, søgning og andre operationer kan med maksimal hastighed.
* Import ingangsættes med at trykke på knap ”OK”. Efter et stykke tid – afhængig af mængden af data - vises en meddelelse med ”Import OK”, som indikerer, at importen er afsluttet med succes.

Ovenstående process gentages for alle datakilder, som ønskes importeret til databasen.

# Tilpasning af parameterdata vedr. tabeller og felter

### 

## Oversvømmelser (t\_flood)

## Bygninger (t\_building)

## Firmaer (t\_company)

## Mennesker og helbred (t\_human\_health)

## Biodiversitet (t\_bioscore)

## Kritisk infrastruktur (t\_infrastructure)

## Offentlig service (t\_publicservice)

## Turisme (t\_tourism)

## Rekreative områder (t\_recreative)

## Vej og Trafik (t\_road\_traffic)

## Bygningsanvendelse (t\_build\_usage)

## Skadefunktioner (t\_damage)

## Kvadratmeterpriser (t\_sqmprice)

# Udvikling af nye modeller

….Ikke skrevet endnu…..

## Parametre til faneblad ”Modeller”

### 

## Parametre til faneblad ”Queries”

### 

## Parametre til faneblad ”Data”

# Skift til ny database teknologi

# Appendix 1 – Alle parametre pr. 24/11 – 2021

Nedenstående er en liste over samtlige paramtre i systemet pr. dags dato.

Det er kun kolonnerne ”parent”, ”name”, ”type”, ”explanation” samt ”value”som er vist

Er ”value” længere end 30 karakterer, er denne forkortet. Forkortelser er indikeret af ”….” i slutningen af teksten