**Administratorvejledning til SkadesØkonomi**

Version 7 - 28. maj 2022

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](#__RefHeading___Toc3649_1081882701)

[2. Historie 4](#__RefHeading___Toc3651_1081882701)

[3. Idégrundlag for det nye system 4](#__RefHeading___Toc3653_1081882701)

[4. Parameter-tabellen 6](#__RefHeading___Toc3655_1081882701)

[Opbygning af hieraki vha. “parent” feltet 7](#__RefHeading___Toc3657_1081882701)

[Struktur og feltbeskrivelse for parameter tabel 8](#__RefHeading___Toc3659_1081882701)

[Brug af “type” feltet 8](#__RefHeading___Toc3661_1081882701)

[Retningslinjer vedrørende navngivning af parametre 10](#__RefHeading___Toc3663_1081882701)

[5. Konvertering af rasterbaserede data til vektordata 11](#__RefHeading___Toc3665_1081882701)

[6. Import af data til databasen 12](#__RefHeading___Toc3667_1081882701)

[7. Tilpasning af parameterdata vedr. tabeller og felter 14](#__RefHeading___Toc3669_1081882701)

[Liste over tabeller i systemet 15](#__RefHeading___Toc3671_1081882701)

[Oversvømmelsesdata 15](#__RefHeading___Toc3673_1081882701)

[Bygninger 16](#__RefHeading___Toc3675_1081882701)

[Vej og trafik 16](#__RefHeading___Toc3677_1081882701)

[Firmaer 16](#__RefHeading___Toc3679_1081882701)

[Mennesker og helbred 17](#__RefHeading___Toc3681_1081882701)

[Biodiversitet 17](#__RefHeading___Toc3683_1081882701)

[Kritisk infrastruktur 17](#__RefHeading___Toc3685_1081882701)

[Offentlig service 18](#__RefHeading___Toc3687_1081882701)

[Rekreative områder 18](#__RefHeading___Toc3689_1081882701)

[Turisme 18](#__RefHeading___Toc3691_1081882701)

[Bygningsanvendelse 18](#__RefHeading___Toc3693_1081882701)

[Skadesfunktioner 19](#__RefHeading___Toc3695_1081882701)

[Kvadratmeterpriser 19](#__RefHeading___Toc3697_1081882701)

[8. Udvikling af nye modeller 20](#__RefHeading___Toc3699_1081882701)

[9. Skift til ny database teknologi 20](#__RefHeading___Toc3701_1081882701)

[Appendix 1 – Alle parametre pr. 24/11 – 2021 21](#__RefHeading___Toc3703_1081882701)

[Appendix 2 – Administrativ opdatering af parameter tabel 22](#__RefHeading___Toc3705_1081882701)

# Introduktion

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

* De generelle muligheder systemet tilbyder.
* Faglig baggrund for systemet.
* 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.

Administratoren af plugin’et SkadesØkonomi skal varetage en række opgaver, før plugin’et fungerer efter hensigten.

* Systemet skal konfigureres til at benytte den rigtige database. Se afsnit 4, ”*Parameter tabel”.*
* Rasterbaserede oversvømmelsesdata skal konverteres til vektordata. Plugin’et SkadesØkonomi kan pt. kun arbejde med vektorbaserede oversvømmelsesdata. Oversvømmelsesdata er ofte i et raster format, så en konvertering er nødvendig. Se afsnit 5, *”Konvertering af rasterbaserede data til vektordata”.*
* Før systemet kan fungere skal relevante data importeres ind i Skadeøkonomi databasen. Dette arbejde beskrives i afsnit 6, *”Import af data til databasen”*
* Installationen af plugin forventer, at data har et bestemt udseende, dvs. at tabellerne hedder noget specifikt og er placeret i bestemte schemaer i systemets database. Er dette ikke tilfældet med de nye data, skal systemet konfigureres til de nye tabeller/feltnavne. Se afsnit 7, ”Tilpasning af parameterdata vedr. tabeller og felter”.
* Plugin’et kan udvides med nye modeller. Dette beskrives i afsnit 8, *”Udvikling af nye modeller”.*
* Databasetypen kan skiftes fra PostgreSQL/PostGIS til andre spatielle databasetyper supporteret af QGIS. Dette beskrives i afsnit 9, *”Skift af database teknologi”.*

# Historie

Som beskrevet i brugervejledningen er grundlaget for plugin’et Skadesøkonomi en række modeller, som beregner økonomiske og andre konsekvenser ved oversvømmelser grundet stormflod, nedbør, vandløb og på sigt terrænnært grundvand.

Det originale projekt er et sæt af Python-scripts beregnet til udførelse via ESRI ArcGIS. Disse scripts fungerer uafhængigt af hinanden og benytter kortlag vist i ArcGIS, samt brugerindtastede eller brugervalgte parameterværdier som grundlag for beregningerne. Resultatet er typisk et nyt kortlag inkl. et sæt tilhørende alfanumeriske oplysninger.

De originale scripts fungerer generelt set således:

1. Brugeren vælger to lag:  
   1. Et lag som indeholder oversvømmelses polygoner med vanddybde som alfanumerisk information.
   2. Et lag som indeholder de objekter, der er grundlaget for modellen, f.eks. bygninger, veje, infrastruktur, industri, osv.
2. Brugeren indtaster eller vælger en række ekstra parametre, som indgår i den valgte modelberegning. Parametre kan være minimum vanddybde, valg af oversvømmelsestype, antal timer med oversvømmelse, o.lign.
3. Herefter bliver modelberegningen gennemført. De i ArcGIS indbyggede faciliteter benyttes til at foretage en overlay-analyse mellem laget med oversvømmelsespolygoner og laget med interesse objekter. Og for hvert oversvømmet objekt udregnes en række værdier, eksempelvis skadeberegninger og værditab, baseret på vanddybde og øvrige parametre. Ved afslutning præsenteres resultatet som et nyt kortlag.

# Idégrundlag for det nye system

Det grundlæggende ønske har været at erstatte de originale ArcGIS baserede scripts med et Open Source baseret system, som fungerer på basis af et frit og gratis GIS produkt, i dette tilfælde QGIS.

Endvidere er ønsket, at det er 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 er udformet:

* Alle datasæt importeres til en spatiel database. Ved datasæt forstås eks. oversvømmelsesdata, datasæt med interesseobjekter (bygninger, veje, infrastruktur, osv.), samt opslagsdata til brug for de forskellige beregninger.
* Alle modelberegninger defineres som SQL-udtryk, der kun benytter tabeller, som findes i databasen.
* Systemet designes således, at 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 opsætningsparametre – SQL-forespørgsler, hjælpetekster, søgeparametre, osv. til plugin’et gemmes som tekst-strenge i selve databasen i en administrationstabel kaldet *parameter-tabellen*. Dette muliggør at flytte hele systemet fra en pc til en anden blot ved at lave en sikkerhedskopi af databasen og – sammen med plugin’et - installere denne på en ny pc.
* 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 og -værdier for hhv. tabeller, felter og søgeværdier gemmes i parametertabellen, således at det er muligt at genskabe et reelt SQL-udtryk kun vha. informationer i parametertabellen.

Et eksempel - Modellen er:

”Find alle bygninger, som bliver oversvømmet, men udeluk oversvømmelser med 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:

* Tabellen “bygninger” indeholder bygningsdata med felterne “kom\_kode” (kommune kode), “geom” (bygningens geometri) og “fid” (en entydig nøgleværdi for bygningen).
* Tabellen “oversvoem” indeholder oversvømmelses polygoner med felterne: “geom” (oversvømmelsespolygon) og “vanddybde” (vanddybde for polygonen).
* Tabellen “kvmpris”, som 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 systemets 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 konstantværdier 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 og -værdier for hhv. tabeller, felter og konstanter gemmes også i parametertabellen. For disse parametre gemmes endvidere tilhørsforholdet til det generaliserede SQL-udtryk.

Når en bruger starter en modelkørsel i plugin’et, sker der følgende:

* Det generaliserede SQL-udtryk findes i parameter-tabellen via brugervalget.
* Alle øvrige parametre, 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-administratoren kan importere tabeller med vilkårlige tabel- og feltnavne til skadeøkonomi-databasen og derefter tilpasse parameterværdier for tabel- og feltnavne i parametre tilhørende modellen. Der dog den betingelse, at de nye tabeller skal indeholde alle relevante kolonner som type- og værdimæssigt svarer til de originale kolonner.

Når GIS-administratoren har foretaget overstående konvertering til generaliseret SQL med diverse parameternavne, så 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-tabellen

Parameter-tabellen en den centrale informationstabel for plugin’et SkadesØkonomi.

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

Tabellen indeholder endvidere en lang række andre oplysninger, som er nødvendige for at generere brugergrænseflader til de forskellige faneblade i plugin’et. Tabellen indeholder derfor en række ekstra rækker, samt 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

Placering og navn på parameter-tabellen i databasen angives af administratoren i fanebladet ”General” vha. to felter, henholdsvis ”Database” og ”Parameter-tabel”. Hvis fanebladet ”General” ikke vises, kan man trykke på ”Administrator” nederst til højre i plugin’et brugergrænseflade for at gøre fanebladet synligt.

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

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

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

## Opbygning af hieraki vha. “parent” feltet

Alle poster i parametertabellen vises i brugerdialogens faneblade ”General”, ”Query”, ”Data”, ”Models” og ”Reports”. Disse faneblade indeholder alle en visuel træstruktur, som viser både parameternavne og -værdier. Når brugeren flytter musen hen over et specifikt navn eller værdi vises en hjælpe-tekst, som er i en anden kolonne i parametertabellen, nemlig ”explanation”.

Placeringen af parameternavne og -værdier i de forskellige faneblade, samt positionen i de enkelte faneblade styres af specielle data i parameter-tabellen: Data arrangeres i et hierarki, hvor stort set alle parameterposter bliver ”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 i parameter-tabellen med samme navn som fanebladet (parameter type “G”, se afsnit *Brug af ”type” feltet*). Disse gruppeposter fungerer som “rod” for hvert faneblad. Alle parameterposter og deres underposter, der 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.

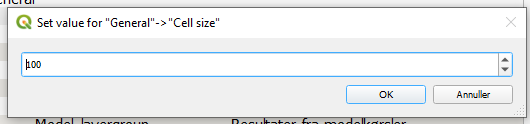
|  |  |  |
| --- | --- | --- |
| Feltnavn | Type | Forklaring |
| Name | tekst | Navn for parameter; skal være unikt i hele tabellen. |
| Parent | tekst | Navn på en anden parameterpost, som “ejer” denne parameterpost. Hvis posten ikke ejes af nogen andre poster, efterlades den blank. |
| Value | tekst | Værdi af parameter; 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 én af følgende værdier: “G”, “I”, “O”, “P” eller “T”; se afsnit *Brug af “type” feltet* for detaljeret 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 parameteren repræsenterer et modelnavn, skal feltet indeholde navnet på den bagved liggende SQL-forespørgsel. Value felt efterlades så tomt. |
| Explanation | tekst | Tekst som beskriver parameterens formål; vises i pop-up dialog, når musen 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 parameternavn og -værdi i fanebladet vise et afkrydsningsfelt til venstre for parameternavn. |

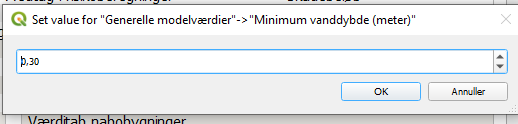
## Brug af “type” feltet

“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 grupper. Den indeholder ingen værdi, men parameternavnet vises i fanebladenes træstruktur. Parametre, som skal vises under denne gruppe, skal tildeles gruppe-parameterens navn som værdi i feltet “parent”.
* Et billede, der indeholder tekst

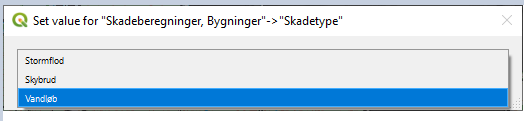
  Automatisk genereret beskrivelseEksempel: I fanebladet “Generelt” findes en parameterpost “Name templates” med type lig med ”G”. Parametrene “Cell layername”, “Group name template”, “Main groupname”, o.a. har alle fået tildelt “Name templates” som “parent”. Det giver følgende resultat:

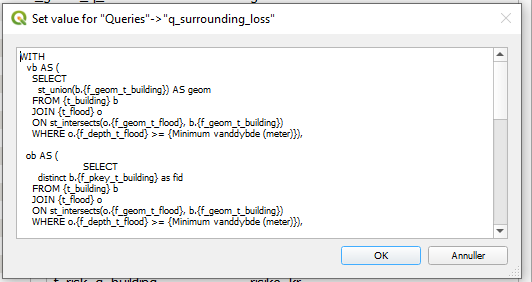
* “I”:   
  Parameter skal indeholde et heltal. Når man dobbeltklikker på parameternavnet eller værdien i fanebladet, vil der vises en brugerdialog, hvori værdien kan ændres.   
    
  Dialogen bliver udformet, så der kun kan indtastes et heltal. Minimum og maksimum værdier bestemmes af værdierne i hhv. felterne “minval” og “maxval”. Stepstørrelsen i det tilhørende rullefelt bestemmes af værdien i feltet “lookupvalues”
* Eksempel: Parameteren “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 parameternavnet eller værdi-feltet i fanebladet, vil der vise sig en brugerdialog, hvori værdien kan ændres.  
    
  Dialogen bliver udformet, så der kun kan indtastes reelle tal. Minimum og maksimum værdier bestemmes af værdierne indtastet i felterne “minval” og “maxval”. Stepstørrelsen i det tilhørende rullefelt bestemmes af værdien i feltet “lookupvalues”



* “O”:

Parameter skal indeholde én bestemt værdi udvalgt fra en forudbestemt liste. Listen er værdisat i feltet “lookupvalues”, hvor den optræder som en tekst, der er sammensat af de mulige værdier adskilt af tegnet “¤” (Shift-4). Ved dobbeltklik på enten parameternavn 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”:   
  Parameter 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 nuværende tekst.

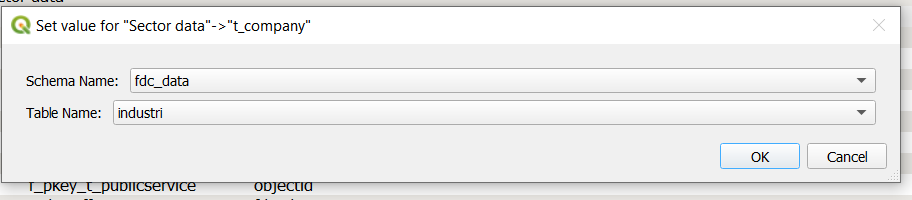


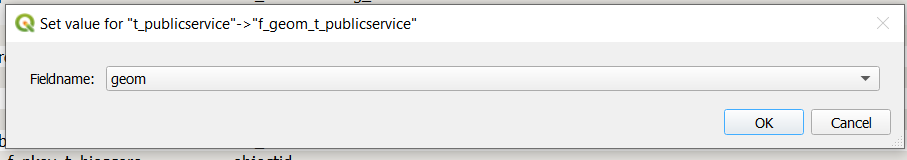
* “T”:   
  Parameter kan indeholde en enkelt linje tekst. Der er ingen begrænsninger på linjens længde. Ved dobbeltklik på navn eller værdifelt vises en tekstboks med den nuværende tekst.

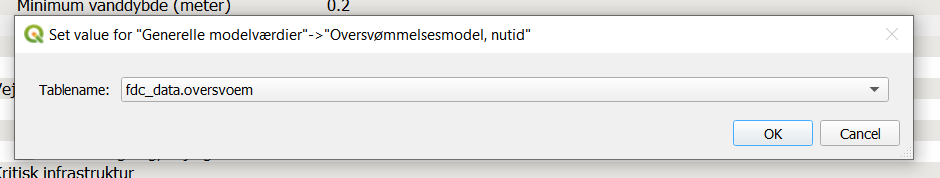
Et billede, der indeholder tekst

Automatisk genereret beskrivelse

* “S”:   
  Parameter indeholder en tabelangivelse ”*schema navn*”.”*tabel navn*”. Ved dobbeltklik på navn eller værdifelt vises en dialogboks med 2 combobokse til valg af hhv schema og derefter tabel navn.



* “F”:   
  Parameter indeholder et feltnavn. Udvalget af feltnavn findes ved at hente alle feltnavn fra tabel defineret vha *parent* værdien fra posten. Ved dobbeltklik på navn eller værdifelt vises en dialgob boks med et combofelt med alle navnekandidater.  
    
  

* “Q”:   
  Parameter indeholder en schema/tabelnavn. Tabellen vælges udfra indholdet af felt ”default”, som forventes at indholde *starten* på et parameternavn. Disse parametre forventes alle af være henvisninger til tabeller (dvs. parameter-rækker af type ”S”). Udvalget af tabelnavne genereres ved at hente alle schema-tabelnavne fra parameter rækker, som starter med værdien fra felt ”default”. Disse kandidater vises i en comboboks. Efter brugervalg gemmes schema-tabelnavnet i felt ”value” og paramternavnet gemmes i felt ”lookupvalues”.  
    
  

## 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, at disse data “er noget som GIS-administratorer står for”.

|  |  |
| --- | --- |
| Parameternavn | Er reference til |
| t\_bioscore | Bioscore tabel |
| t\_build\_usage | Bygningsanvendelse (administrationstabel) |
| t\_building | Bygningstabel |
| t\_company | Firma tabel |
| t\_damage | Skadefunktioner (administrations tabel) |
| t\_flood | Oversvømmelsestabel |
| t\_human\_health | Oplysninger om beboere |
| t\_infrastructure | Infrastrukturtabel |
| t\_publicservice | Offentlige områder |
| t\_recreative | Rekreative områder |
| t\_road\_traffic | Vejdata tabel |
| t\_sqmprice | Gennemsnitlig kvadratmeter pris fordelt på kommuner (administrations tabel) |
| t\_tourism | Turisme-tabel |

* Parameternavne for *generaliserede SQL-forespørgsler* starter med “q\_” efterfulgt af query funktionen på engelsk.
* Eksempel: “q\_building”: Skademodel for 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. følgende 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 risikobeløb |
| 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* og ikke *absolutte krav* - undtaget følgende: Angivelser for hhv. primary key og geometri felter for *queries* *skal* følge formlerne; ellers vil systemet fejle ved genereringen af resultat-tabeller.

# Konvertering af rasterbaserede data til vektordata

Plugin’et SkadesØkonomi kan pt. kun arbejde med vektordata, så man vil ofte stå i en situation, hvor det er nødvendigt at konvertere rasterdata til vektordata, eksempelvis oversvømmelsesdata. Endvidere skal der ofte foretages en forbehandling af rasterdata, før det er hensigtsmæssigt at konvertere dem.

Dette afsnit beskriver en metode, hvor rasterdata foreligger med pixels i reelle tal i enheden meter. Disse konverteres til integer i enheden cm og kategoriseres ved afrunding til nærmeste 5 cm, hvorefter de vektoriseres. Generel beskrivelse i forhold til QGIS:

1. Raster pixels ganges med 20 vha. raster-kalkulator.
2. Raster pixels konverteres til integer.
3. Raster pixels ganges med 5 (funktionerne 1-3 vil konvertere pixel i meter til cm. Og klassificere de enkelte værdier til 5 cm tærskler).
4. Rasterdata konverteres til polygoner vha. de modificerede dybde-værdier.

En detaljeret beskrivelse er under udarbejdelse.

# Import af data til databasen

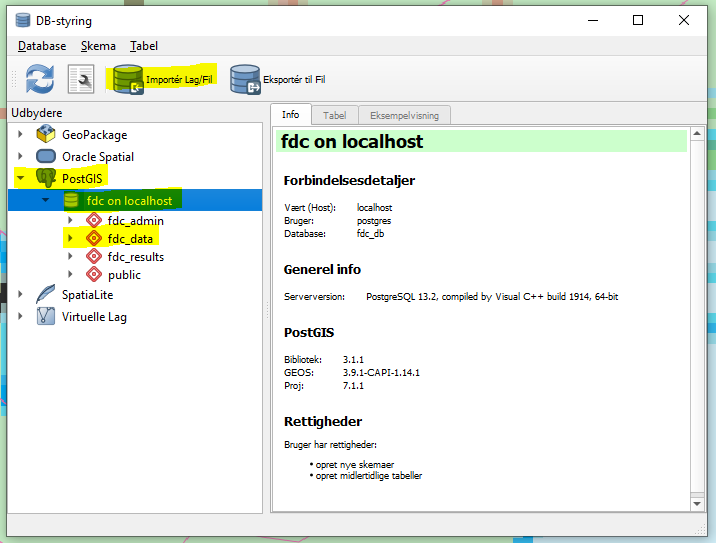
Efter installation af databasesystem, oprettelse af database, kørsel af installationsscript, installation af plugin og opsætning af forbindelse mellem QGIS og database er det næste skridt af gøre data tilgængelig for modelgenereringen.

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

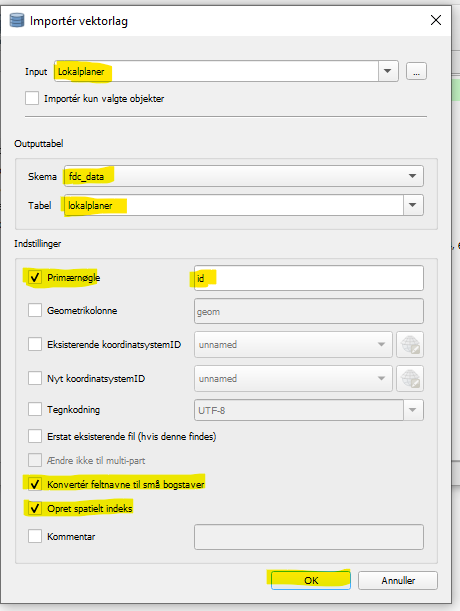
* At de forskellige vektordata findes (det kan være fil data eks. MapInfo data eller shapefiler; data fra andre databaser eller evt. WFS-datakilder. Alle vektorbaserede datakilder kan benyttes.).
* At de enkelte datakilder indeholder de relevante data, som er nødvendige for modelberegningen, samt at de har den korrekte felttype (se afsnit 7 ”Tilpasning af parameterdata for tabeller og felter” i de relevante underafsnit for de enkelte datatabeller).
* Hvis en datakilde indeholder både multipart og singlepart objekter, skal denne konverteres, således at alle objekter bliver multipart (se QGIS processering: ”Konvertér til multipart”).

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’s datavælger.
* QGIS’s Database Manager startes op: Der vælges ”Database” > ”DB-styring”, hvorefter der viser sig følgende brugerdialog:



* Under ”Udbydere” udfoldes ”PostGIS”.
* I listen over de forskellige databaser udfoldes den relevante database med samme forbindelsesnavn, som blev angivet ved opsætning af adgang til parameter-tabellen (Afsnit 4, ”Parametertabel”). I eksemplet er navnet på forbindelsen ”fdc on localhost”.
* Der klikkes på ”Importér Lag/Fil”, hvorefter følgende brugerdialog viser sig:



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

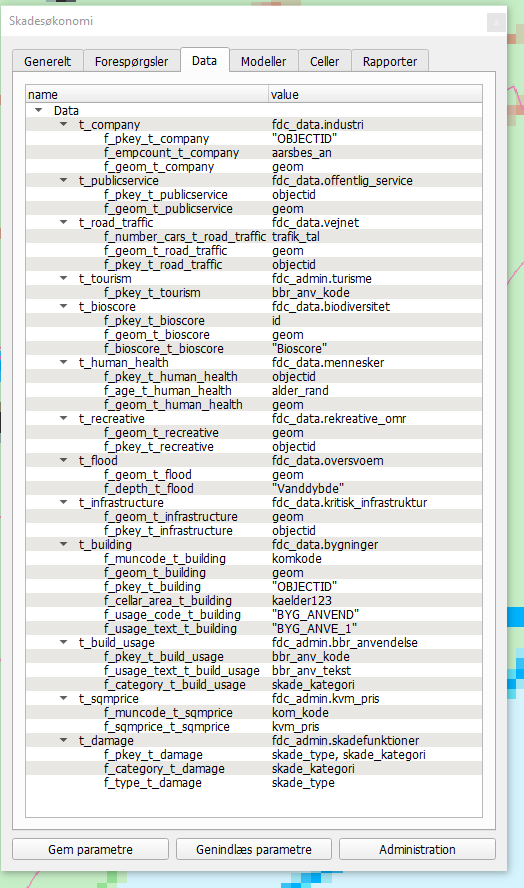
Ovenstående proces gentages for alle datakilder, som ønskes importeret til databasen.

# Tilpasning af parameterdata vedr. tabeller og felter

Efter import, men før man begynder at udføre modelberegninger, skal parameter-opsætningen for plugin’et tilpasses, således at systemet kan finde relevante tabeller og enkeltfelter i tabellerne. Der skal også ske en kontrol af de enkelte felters type og indhold.

Der er pt. 10 datatabeller, som indeholder aktuelle data og 3 administrationstabeller, der indeholder opslagsdata. Antallet kan ændre sig, hvis der bliver tilføjet, rettet i eller fjernet modeller fra systemet.

Alle tilpasninger foregår i plugin’ets faneblad ”Data”:



Indholdet af fanebladet beskriver sammenhængen mellem parameternavne for hhv. tabeller/felter og deres faktiske placeringer og/eller navne.

* Venstre kolonne er parameternavnet for en tabel eller et felt i databasen.
* Højre kolonne er det faktiske navn for tabellen og/eller feltet.

Parameternavne startende på ”t\_” er referencer til *tabeller* og navne startende på ”f\_” er referencer til *felter*.

Venstre kolonne viser også tilhørsforholdet mellem tabeller og deres felter. Parameternavne for felter, som står umiddelbart under et parameternavn for en tabel, repræsenterer felter i denne tabel.

Feltlisten indeholder referencer til alle de felter i tabellen, som er nødvendige for, at modellen kan beregnes. Tabellen *kan* indeholde flere felter, som er unødvendige for modelberegningen.

Administratoren skal sikre sig, at samtlige parameternavne indeholder korrekte referencer til tabeller og felter. Og at tabellerne er til stede i databasen med refererede felter.

Ved angivelse af både schema-, tabel- og feltnavne skal følgende regler overholdes:

* Indeholder navnet en eller flere kapitaler, ”æøå”, blanktegn eller andre specialtegn skal navn angives med anførselstegn. Ellers kan anførselstegn undværes. Eks. *”Mit dårlige NAVN”* contra: *mit\_gode\_feltnavn.*
* Ved angivelse af kombinationer af schema og tabelnavne gælder reglerne for de individuelle navne.  
    
  Eksempel:

*”Schema Å”.”Tabel fut”*; *”Schema Q”.table\_a*; *mit\_schema.”TabelÆ”*, *schema1.tabel2* er alle korrekte schema/tabel navne.  
*”Schema a.49 ronin”* er *ikke* korrekt

* Det er i orden at sætte anførselstegn rundt om navne, hvor det ikke er strengt nødvendigt.  
  Eks.: *mit\_schema.min\_tabel* er ok. Og det er *”mit\_schema”.”min\_tabel”* også.

Tilpasning foregår ved at dobbeltklikke på navnet eller værdien for den parameter, der ønskes ændret. Herefter vises et pop-up skærmbillede, som indeholder den nuværende værdi af parameteren. Skærmbilledet er tilpasset parameter-typen; se afsnit *Brug af ”type” feltet.*

Administratoren kan nu ændre værdien til det ønskede. Der afsluttes med at trykke på ”OK” i pop-up dialogen.

Når alle rettelser er foretaget og testet, kan man gøre rettelserne permanente ved at trykke på ”Gem rettelser” nederst til venstre i plugin’ets brugergrænseflade. Dette vil gemme alle ændringer permanent i databasen.

## Liste over tabeller i systemet

Nedenstående liste er en beskrivelse af de tabeller, som pt. findes til systemet. Over tid kan denne liste godt ændre sig ved tilføjelser, rettelser eller fjernelse af modeller med tilhørende tabeller.

**Alle spatiale tabeller forventes at være i projektion UTM32/ETRS89, dvs. EPSG: 25832.**

Kolonnerne har følgende mening:

* Parameternavn: Navnet på parameter (Venstre kolonne i ”Data” faneblad).
* Felttype: For parametre som refererer til *tabeller*: ingen mening. For *felter* angives, hvilken datatype, som feltet skal have. Heltal, reelt tal, tekst, Ja/Nej (boolean), geometri (spatiale data).
* Standard værdi: Værdi for parameter ved installation af system.
* Forklaring: Kortfattet forklaring for *tabellen* eller *feltet*, som parameter værdien refererer til.

## Oversvømmelsesdata

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_flood | - | fdc\_data.oversvoem | Tabel indeholder vektorpolygoner (*ikke* rasterdata) med oversvømmelsesdata, dvs. geometrier med dybder. |
| f\_geom\_t\_flood | Geometri | geom | Oversvømmelsespolygoner. |
| f\_geom\_t\_flood | Reelt tal | vanddybde\_m | Vanddybde i **meter.** |

Tabellen benyttes i samtlige modeller.

## Bygninger

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_building | - | fdc\_data.bygninger | Tabel indeholder vektorpolygoner med bygningsdata fra GeoDanmark beriget med BBR-data. |
| f\_pkey\_t\_building | Heltal | objectid | Nøglefelt for tabel. |
| f\_geom\_t\_building | Geometri | geom | Bygningspolygoner. |
| f\_muncode\_t\_building | Heltal | komkode | Kommunekode. |
| f\_usage\_code\_t\_building | Heltal | byg\_anvend | BBR-anvendelseskode, bemærk at er dette felt ikke udfyldt, så medtages bygning ikke i beregningerne. Ønsker man dette, skal tomme felter udfyldes med -9999. |
| f\_usage\_text\_t\_building | Tekst | byg\_anv\_tekst | BBR-anvendelsestekst. |
| f\_cellar\_area\_t\_building | Reelt tal | kaelder\_areal | Areal af bygningens kælder;  0.0 hvis ingen kælder. |

Tabellen benyttes i modellerne for: Bygningsskade, Værditab for nabobygninger, Mennesker og helbred, samt Industri.

## Vej og trafik

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_road\_traffic | - | fdc\_data.vejnet | Tabel indeholder vektor linjestykker med vejføringer. Kan være baseret på GeoDanmark vejmidter beriget med trafik-oplysninger. |
| f\_pkey\_t\_road\_traffic | Heltal | objectid | Nøglefelt for tabel. |
| f\_geom\_t\_road\_traffic | Geometri | geom | Vej polylinjer. |
| f\_number\_cars\_t\_road\_traffic | Heltal | trafik\_tal | Antallet af biler pr. dag. |

Tabellen benyttes i modellen for: Vej og Trafik.

## Firmaer

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_company | - | fdc\_data.industri | Tabel indeholder firmaoplysninger fra CVR tilknyttet punkt geometri. |
| f\_pkey\_t\_company | Heltal | objectid | Nøglefelt for tabel. |
| f\_geom\_t\_company | Geometri | geom | Punkter (punkterne tilknyttes GeoDanmark bygninger vha. et spatialt join. Punkter der ikke rører ved en GeoDanmark bygning medtages ikke i beregningerne).. |
| f\_empcount\_t\_company | Heltal | aarsbes\_ant | Antal helårs beskæftigede. |
| f\_comp\_id\_t\_company | Heltal | cvr\_nr | Cvr-nummer for virksomhed. |
| f\_prod\_id\_t\_company | Heltal | p\_nr | Produktionsenhed for virksomhed. |

Tabellen benyttes i modellen for: Industri

## Mennesker og helbred

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_human\_health | - | fdc\_data.mennesker | Tabel indeholder punktdata med oplysninger om personer og alder. |
| f\_pkey\_t\_human\_health | Heltal | objectid | Nøglefelt for tabel. |
| f\_geom\_t\_human\_health | Geometri | geom | Punkt geometri. |
| f\_age\_t\_human\_health | Heltal | alder\_rand | Alder for person. |

Tabellen benyttes i modellen for: Mennesker og helbred.

## Biodiversitet

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_bioscore | - | fdc\_data.biodiversitet | Tabel indeholder vektorpolygoner med bioscore data. |
| f\_pkey\_t\_bioscore | Heltal | objectid | Nøglefelt for tabel. |
| f\_geom\_t\_bioscore | Geometri | geom | Polygoner for arealer. |
| f\_bioscore\_t\_bioscore | Heltal | Bioscore | Bioscore nummer. |

Tabellen benyttes i modellerne for: Biodiversitet-kort, Biodiversitet-opsummering.

## Kritisk infrastruktur

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_infrastructure | - | fdc\_data.kritisk\_infrastruktur | Tabel indeholder vektorpolygoner med kritisk infrastruktur.  Data kan eks. være baseret på GeoDanmark bygninger eller tekniske anlæg. Tabellen må kun indeholde polygoner, så linje- eller punkt-objekter skal omdannes til polygoner før indlæsning i tabel - f.eks. ved at generere en buffer omkring objektet |
| f\_pkey\_t\_infrastructure | Heltal | objectid | Nøglefelt for tabel. |
| f\_geom\_t\_infrastructure | Geometri | geom | Polygoner for infrastruktur. |

Alle oprindelige kolonner fra tabellen medtages i modelresultatet, så tabellen kan evt. suppleres med felterne type (Boring, Telemast, Jernbane, mv.) og prioritet (1, 2, 3, mv.), der kan anvendes til korttematiseringer efter beregningerne vha. labels og styles.

Tabellen benyttes i modellerne for: Bygningsskade, Mennesker og helbred, samt Industri.

## Offentlig service

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_publicservice | - | fdc\_data.offentlig\_service | Tabel indeholder vektorpolygoner med bygninger o.a. til offentlig service. Kan evt. baseres på GeoDanmark bygninger. |
| f\_pkey\_t\_publicservice | Heltal | objectid | Nøglefelt for tabel. |
| f\_geom\_t\_publicservice | Geometri | geom | Bygningspolygon. |

Alle oprindelige kolonner fra tabellen medtages i modelresultatet, så tabellen kan evt. suppleres med felterne type (Plejecenter, Skole, Børnehus, mv.) og prioritet (1, 2, 3, mv.), der kan anvendes til korttematiseringer efter beregningerne vha. labels og styles.Tabellen benyttes i modellen for: Offentligt Service.

## Rekreative områder

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_recreative | - | fdc\_data.rekreative\_omr | Tabel indeholder vektorpolygoner med rekreative områder. |
| f\_pkey\_t\_recreative | Heltal | objectid | Nøglefelt for tabel. |
| f\_geom\_t\_recreative | Geometri | geom | Polygoner med rekreative områder. |

Tabellen benyttes i modellen for: Rekreative områder.

**Efterfølgende tabeller er administrationstabeller. Der skal under normale omstændigheder ikke ændres struktur for tabel og feltnavne i disse tabeller.**

Indhold i selve tabellerne kan dog godt tilpasses.

## Turisme

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_tourism | - | fdc\_admin.turisme | Administrationstabel som indeholder sammenhænge mellem udvalgte BBR-anvendelseskoder og en række parametre til skadeberegning for turisme. |

Tabellen benyttes i modellen for: Turisme.

## Bygningsanvendelse

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_build\_usage | - | fdc\_admin.bbr\_anvendelse | Administrationstabel med oplysninger om BBR-anvendelseskoder. |
| f\_pkey\_t\_build\_usage | Heltal | bbr\_anv\_kode | BBR-anvendelseskode, nøglefelt. |
| f\_usage\_text\_t\_build\_usage | Tekst | bbr\_anv\_tekst | BBR-anvendelsestekstforklaring. |
| f\_category\_t\_build\_usage | Tekst | skade\_kategori | Grupperingskode til forsimpling af BBR-anvendelseskode. Nøglefelt til Skadesfunktioner tabel. |

Tabellen benyttes i modellerne for: Bygningsskade.

## Skadesfunktioner

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_damage | - | fdc\_admin. Skadefunktioner | Administrationstabel med oplysninger om div. konstantværdier, som indgår i skadesberegninger for bygninger og evt. kældre.  Felterne heder b0, b1, b2 for bygninger og c0 for kældre.Formlerne er flg:  Byg.skade = b0 + bygnings\_areal \* (b1 \* ln(vandybde\_m\*100.00) + b2)  Kælderskade = c0 \* kælder\_areal  Felterne b0, b1, b2, c0 er kolonner i denne tabel |
| f\_pkey\_t\_damage | Tekst | skade\_type, skade\_kategori | Nøglefelt (er) for tabel. |
| f\_category\_t\_damage | Tekst | skade\_kategori | Bygningskategori, gruppering af BBR-anvendelseskoder. |
| f\_type\_t\_damage | Tekst | skade\_type | Skadetype: Kan være *Skybrud*, *Stormflod* eller *Vandløb.* |

Tabellen benyttes i modellerne for: Bygningsskade.

## Kvadratmeterpriser

|  |  |  |  |
| --- | --- | --- | --- |
| Parameternavn | Felttype | Standard værdi | Forklaring |
| t\_sqmprice | - | fdc\_admin.kvm\_pris | Administrationstabel, som indeholder sammenhænge mellem kommunekode og gennemsnitlig kvm. Pris. |
| f\_muncode\_t\_sqmprice | Heltal | kom\_kode | Kommunekode; nøgle felt for tabel. |
| f\_sqmprice\_t\_sqmprice | Reelt tal | kvm\_pris | Gennemsnitlig kvadratmeter pris. |

Tabellen benyttes i modellerne for: Bygningsskade, Værditab for nabobygninger.

[Der vil her blive indsat en beskrivelse af LNH water om, hvor kvadratmeterpriser pr. kommune kan findes.]

# Udvikling af nye modeller

…. Under udarbejdelse til version 2 ….  
**Parametre til faneblad ”Modeller”**

**Parametre til faneblad ”Queries”**

**Parametre til faneblad ”Data”**

# Skift til ny database teknologi

….Under udarbejdelse til version 2 …..

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

Nedenstående er en liste over samtlige parametre 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 med ”….” i slutningen af tekstindholdet.

… Under udarbejdelse …

# Appendix 2 – Administrativ opdatering af parameter tabel

Der *kan* forekomme rettelser til systemet fra leverandørens side. Disse opdateringer kommer ofte i form af opdateringer til parameter tabellen – dvs. ikke en reinstallation af hele systemet, men et sæt af SQL kommandoer der sendes som en tekstfil til GIS administratoren eller bliver gjort tilgængelige på systemets GitHub hjemmeside.

For at gennemføre denne type opdateringer gøres følgende:  
  
(Proceduren benytter generelt samme metode som GIS administrator benyttede ved den initielle oprettelse af shemaer og tabeller til systemet)

* GIS administrator modtager en tekstfil med SQL kommandoer eller downloader denne fra GitHub.  
  Navnet vil eksempelvis være navngivet *fdc\_db\_patch.2021-12-11.sql,* hvor ”mellemnavnet” (2021-12-11) er datoen for udgivelsen af rettelsen.
* **Foretag en sikkerhedskopiering (backup) af administrations schema i databasen for Skadesøkonomi**. Se installationsvejledning for denne procedure.
* Fra Windows ”Start” menu startes PGAdmin4 og adgangskode for PGAdmin4 indtastes.
* Markér og udfold blad "Servere" og dernæst den relevante server (sandsynligvis "PostgreSQL 13")
* Marker og udfold blad "Databases" og dernæst databasen som benyttes af skadeøkonomi plugin.
* Klik på menupunkt "Tools" -> "Query tool". I højre side af skærmbilledet vises et query-vindue hvori der er muligt at skrive sql kommandoer.
* Tast Alt-O. PGAdmin4 viser en "Åbn-fil" dialog.
* Navigér til førnævnte sql-tekstfil og åbn denne. Indholdet vises i query-vindue.
* *Tjek for yderligere instruktioner* i den viste tekst. Det kan evt. handle om at ændre navne på schemaer eller tabeller i sql-kommandoerne. Foretag eventuelle rettelser i teksten.
* Udfør kommandoerne i scriptet ved at trykke på tast F5. Scriptet herefter blive udført.
* For at rettelser bliver aktiveret i igangværende QGIS sessioner skal der klikkes på knap ”Genindlæs parametre” i brugerdialog for ”Skadesøkonomi” dialog.