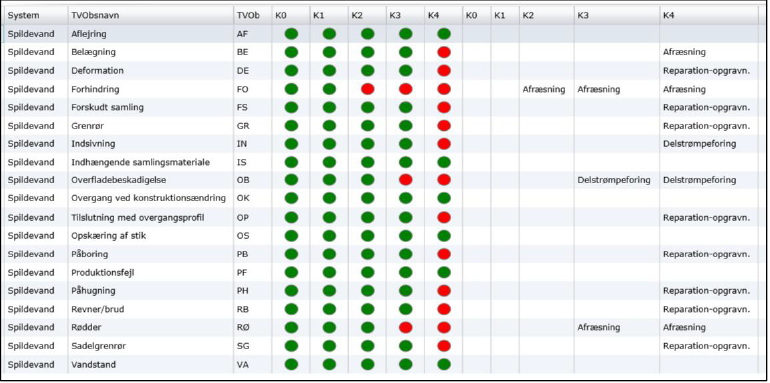
1. Baggrund

VandCenter Syd ønsker at få opstillet en plan for akutte tiltag i afløbssystemet baseret på observationer fra foreliggende TV-Inspektioner.

På baggrund af disse TV-Inspektioner, er en skade klasse blevet tildelt hver observeret tilstandsforværring på afløbs ledningerne. På billedet nedenunder kan ses en visualisering af disse klasser (K0,K1,K2,K3,K4), som beskriver hvor kritisk en skade er, der er blevet observeret i afløbene.



Kun de høje skadeklasser (K4), og skader af særligt pressende natur (Rødder, Forhindringer, Overfladeskade) er Interessante for os, angående et akut tiltag. Et ekstra gennemtjek ønskes dog af de afløb som overholder disse kriterier, for at sikre at skaden er så pressende som dens klasse hentyder. VandCenter Syd ønsker at få fundet alle afløb af en beskrevet skadeklasse hvor observationerne er lavet i 2018 eller 2019, opstillet på en sådan måde at de kan gennemgås for en grovsortering, og udvælgelse for de endelige tiltag.

1. Problemstilling

Redegørelsen over hvilke afløb der skal udvælges for tiltag, skal laves på baggrund af data udtaget fra databasen. Problematisk er den relevante data spredt over mange tabeller, hvor mange af dem er uden direkte relationer til de andre relevante tabeller. Dette gør det vanskeligt at udtrække den data der skal bruges for at kunne bestemme tiltaget. Nedenunder kan ses en tabel over den data der ønskes, og hvordan de forskellige tabeller denne data kan findes i er relateret til hinanden.

|  |  |  |
| --- | --- | --- |
| *Tabel navn* | *Ønsked Data* | *Relationer til andre tabeller\** |
| CCTVObs | Skadeklasse  Skadetype | PipeEvent |
| PipeEvent |  | CCTVObs  Report |
| Report | Rapport Navn  Observations Dato | PipeEvent  CCTVInspection  PipeReport |
| CCTVInspection | Fysisk Indeks | Report |
| PipeReport |  | MainPipe  Report |
| MainPipe | Afløbstype  Status  Ejer | PipeReport  Pipe  Node |
| Pipe | Hældning  Materiale  Dimension  Tværsnitsform  Kategori  Shapefil | MainPipe  Node |
| Node | Brøndnavn  Bundkvote  Terrænkvote | MainPipe  Pipe |

*\*kun medtaget relationerne til relevante tabeller*

Der ønskes information omkring bundkvoter og terrænkvoter for de brønde på afløbene som observationerne er lavet på. Denne information kan findes i *Node* (Knude) tabellen, men da afløb har både en upstream og en downstream brønd, giver det problemer for Arcgis som har begrænsning for at lave joins flere gange på den samme tabel samtidig.

Disse ovennævnte begrænsninger betyder at der ikke bare kan laves en SQL udvælgelse fra databasen, men at der er bruge for at holde resultaterne for en del af udvælgelsen, mens udvælgelsen fra de andre tabeller bliver valideret.

1. Løsning

På grund af den store spredning af data, og de mange begrænsninger som ArcGis har, har vi brug for at gemme den data vi udtrækker fra databasen mellem flere søgninger. Dette kan gøres effektivt ved at gemme dataet i en ordbog (dictionary), med et nøgleord der gør os i stand til hurtigt at kunne finde den relevante data igen. Vi kan derved lave flere uafhængige søgninger I databasen, samtidig med at vi kan bruge dataet gemt i ordbogen til at sætte søge kriterier op.

Den første søgning og sortering a data vi er interesseret I, er at finde alle de observationer som er foretaget i 2018 eller 2019. Dette kan gøres samtidig med at vi tjekker for skadeklassen og skadetypen, ved at lave join på de to tabeller disse informationer kan findes i. Disse to tabeller, CCTVObs og Report har dog ikke nogen direkte relationer, så vi bliver nød til også at joine deres mellemliggende tabeller.



På billedet kan ses udvælgelsen. Variablerne med navnet row er en række af data fra joinet, fundet af SearchCursoren. SearchCursoren er blevet givet kriterier for hvilket felter vi er interesseret i fra dataet, hvorfor vi kun behøver se på de første tre elementer, row[0], som er skade typen (4 er forhindringer, 9 er overfladeskader og 17 er rødder), row[1], som er den skadeklasse der er observeret og row[3] som er datoen for observationen. (row[2] er rapport navnet)

Når jeg finder rækker der overholder disse kriterier, kan jeg gemme dem i ordbogen. Da jeg også har joinet CCTVInspection og PipeReport med, kan jeg også gemme det fysiske indeks, og bruge afløbets ID fra PipeReport som nøglen.

Her kan der dog fremkomme den situation hvor der er flere observationer for det samme afløb og rapport som overholder disse kriterier. I den situation beholder jeg kun den nyeste observation, da den kan tænkes at være den mest relevante. Dette giver en ordbog der indeholder dataet skadeklassen, skadetypen, datoen, rapport navnet, og det fysiske indeks, med afløbets ID som nøglen.

Vi kan nu gennemløbe MainPipe joinet med Node Tabellen (Med UpStreamNodeID som join feltet) for at finde navnene, bundkvoterne og terrænkvoterne på opstrømsbrøndende for vores afløb. Hver gang vi finder en brønd fra et afløb som vi kan finde Idet for i ordbogen, gemmer vi dataet i ordbogen sammen med det andet data.

Efter at vi frigiver joinet mellem MainPipe og Node, kan vi lave en endelig join mellem MainPipe, Pipe og Node (med Node joined på UpStreamNodeID). Her kan vi finde den endelige data som vi mangler, så hver gang vi finder et afløb hvis ID findes i ordbogen, skriver vi alt den data vi har fundet til vores endelige tabel.

Når vi har gennemgået den sidste join, skulle vores endelige tabel indeholde dataet for alle de afløb som overholder vores kriterier. Da shapefilerne fra Pipe tabellen også gemmes bygges findes længden på afløbene automatisk. Der var også ønsket nogle kolonner som kunne manuelt redigeres af en bruger når de gennemgår dem, hvilket er blevet indsat til sidst. Opbygningen af den endelige tabel kan findes I appendix.

1. Appendix

*\*Bemærk, PIPE og PATH er variabler der indeholder stier til hvor Featureklassen skal gemmes, og til Pipe data tabellen.*

dbname = "RoughSortingFeatureClass"

arcpy.CreateFeatureclass\_management(PATH\*,dbname ,"POLYLINE","","DISABLED","DISABLED",arcpy.Describe(PIPE\*).spatialReference)

arcpy.AddField\_management(dbname ,"ReportName","TEXT")

arcpy.AddField\_management(dbname ,"DateConducted","DATE")

arcpy.AddField\_management(dbname ,"CCTVObsCode","SHORT")

arcpy.AddField\_management(dbname ,"CCTVObsClass","SHORT")

arcpy.AddField\_management(dbname ,"Category","TEXT","","",50)

arcpy.AddField\_management(dbname ,"PhysicalIndex","DOUBLE")

arcpy.AddField\_management(dbname ,"NetworkTypeCode","SHORT")

arcpy.AddField\_management(dbname ,"StatusCode","SHORT")

arcpy.AddField\_management(dbname ,"OwnerID","LONG")

arcpy.AddField\_management(dbname ,"MaterialCode","SHORT")

arcpy.AddField\_management(dbname ,"CrossSectionCode","SHORT")

arcpy.AddField\_management(dbname ,"MainDimension","LONG")

arcpy.AddField\_management(dbname ,"Slope","DOUBLE")

arcpy.AddField\_management(dbname ,"UpstreamNodeName","TEXT")

arcpy.AddField\_management(dbname ,"UpstreamInvertLevel","DOUBLE")

arcpy.AddField\_management(dbname ,"UpstreamGroundLevel","DOUBLE")

arcpy.AddField\_management(dbname ,"DownstreamNodeName","TEXT")

arcpy.AddField\_management(dbname ,"DownstreamInvertLevel","DOUBLE")

arcpy.AddField\_management(dbname ,"DownstreamGroundLevel","DOUBLE")

arcpy.AddField\_management(dbname ,"FeatureGUID","TEXT")

arcpy.AddField\_management(dbname ,"StreetName","TEXT")

arcpy.AddField\_management(dbname ,"Note","TEXT")

arcpy.AddField\_management(dbname ,"Tiltag","SHORT")

+ SHAPE@ fil fra Pipe

arcpy.AssignDomainToField\_management("RoughSortingFeatureClass","MaterialCode","WWC\_PipeMaterial")

arcpy.AssignDomainToField\_management("RoughSortingFeatureClass","CrossSectionCode","WWC\_PipeCrossSection")

arcpy.AssignDomainToField\_management("RoughSortingFeatureClass","OwnerID","SAIT\_OwnerID")

arcpy.AssignDomainToField\_management("RoughSortingFeatureClass","NetworkTypeCode","WWC\_CCTV\_NetworkType")

arcpy.AssignDomainToField\_management("RoughSortingFeatureClass","StatusCode","WWC\_Status")

arcpy.AssignDomainToField\_management("RoughSortingFeatureClass","CCTVObsCode","WWC\_CCTV\_Obs")