Dokumentation til stored procedure rssp\_Statistic

# Formål

Formålet med denne samlede procedure er at kunne generere en query på cloverleaf databasen, som ser på specifikke interessante parametre, som kan vise en statistik over meddelelser, som er kørt igennem cloverleaf databasen. Der kan være tale om at kigge på meddelelser af en specifik type, sendt i et vis tidsinterval, som eksempelvis ikke er gået igennem til modtageren.

# Indledende

Funktionaliteten i proceduren rssp\_Statistic er delt op i tre underprocedurer, som hver er tilpasset specifikke typer input.

rssp\_DateStatistic tager sig af dato inputs.

rssp\_VarIntStatistic tager sig af inputs der matcher en enkel tupel i en kolonne af typen nvarchar eller int.

rssp\_MultiStat tager sig af input med der matcher flere tupler i en kolonne, alle inputs er nvarchar.

## rssp\_DateStatistic

Denne procedure er som nævnt beregnet til at tage sig af datoer, funktionaliteten er opbygget sådan at proceduren tager imod to inputs fra rssp\_Statistic, hvor de andre input linjer er statiske, dvs. de bliver ikke ændret fra brugerens side.

De to inputs som rssp\_DateStatistic skal tage imod, skal som det første input svare til en start tid og det andet input en slut tid. Begge inputs skal være i datetime format.

De sidste tre inputs der står til funktionen er markeret som output, da @Select, @From og @Where skal bruges senere i alle de nævnte procedurer i afsnittet Indledende.

Når proceduren rssp\_DateStatistic køres fra rssp\_Statistic, er det første den tjekker, om brugeren faktisk har angivet en start og slut tid. Såfremt de ikke har, afslutter funktionen.

Hvis der er angivet en start og en slut tid, vil funktionen vide om tiderne svarer til tider i tabellen t\_Letter eller t\_LetterLog.

Såfremt det er tider i tabellen t\_Letterlog, vil den efterfølgende undersøge om du via en tidligere kørsel af en procedure har tilføjet en join på t\_letterlog. Hvis dette ikke er tilføjet, tilføjer den det til @From.

De to tider i datetime format konverteres herefter til nvarchar(200), dette gøres for at de kan tilføjes til variablen @Where.

Det næste check er om hvorvidt der allerede er noget i variablen @Where. Er der noget i den, tilføjes et ‘and’, så at rssp\_Statistic senere kan se at dette er et separat stykke information den skal query på.

Herefter tilføjes tiderne fra input til @where på en måde, hvorpå de kan bruges med den indbyggede funktion sp\_ExecuteSQL.

Til sidst tilføjes der til @Select et komma og en henvisning til den kolonne hvori input tiderne er fra.

Når alt dette er kørt, føres @Select, @From og @Where tilbage til rssp\_Statistic.

## rssp\_VarIntStatistic

Denne procedure har modsat rssp\_DateStatistic kun et input ad gangen, dette input kan være en streng, som modtages af @Input eller en integer, som modtages af @IntInput.

Proceduren har dog, som rssp\_DateStatistic også @Select, @From og @Where, som ligeledes bruges til at sende information tilbage til rssp\_Statistic.

Proceduren starter med at tjekke for hvorvidt der er noget information i @Input eller @IntInput, såfremt der ikke er, afslutter den. Hvis der er information i den ene af dem, skal den herefter finde ud af hvilken en af dem der ikke er null.

Finder den at det er @Input, som ikke er null, opbygger den herefter en sql sætning med information givet den udefra. Denne sætning bruges med sp\_ExecuteSQL til at tjekke for hvorvidt informationen i @Input er gyldig. Såfremt informationen ikke er gyldig, vil @Validnvar forblive null og den kan slå en fejl op, der fortæller at det du skrev i @Input ikke svarer til noget i den kolonne, som den skulle tjekke i. Fejlen sendes til errorlog via catch-delen i proceduren.

Svarer informationen i @Input til noget i kolonnen, som der blev tjekket på, tilføjes information til @Select, der vil have kolonne navnet, @From som vil få tilføjet information om tabellen, der skal joines på og @Where får muligvis tilføjet ’and’ og information om den specifikke tupel der matcher.

Forskellen mellem hvad der sker, når der i stedet for information i @Input er information i @IntInput er forholdsvis lille. Når der er tale om en integer, skal den som med rssp\_DateStatistic konverteres til nvarchar, for at kunne benyttes informationen med sp\_ExecuteSQL.

@IntInput svarer altid til en messagestatusid, som har et tilhørende navn i tabellen t\_MessageStatus. Da messagestatusid i sig selv ikke siger alle noget, tilføjes i stedet mst\_Name til @Select og der tilføjes en join til t\_MessageStatus i @From. @Where kan ikke indeholde en integer og der bruges derfor i stedet variablen, hvori integeren er blevet konverteret til nvarchar.

Når proceduren er færdig, sender den også information i @Select, @From og @Where tilbage i rssp\_Statistic.

## rssp\_MultiStat

Denne procedure vil modsat rssp\_VarIntStatistic modtage et noget længere input. Dens speciale er at modtage input der svarer til flere EAN numre, SORID numre eller SKS numre, disse input skal være delt op med en delimiter, som i tilfælde af sender numre, vil være semikolon og receivernumre, vil være kolon.

Proceduren modtager ikke @Where udefra som de to tidligere procedurer.

Delimiteren bruges til at synliggøre at det der er efter det er et nyt nummer og derfor ikke skal stå i samme tupel, i den midlertidige tabel der senere i proceduren laves og hvori, numrene vil blive indsat.

Proceduren rssp\_MultiStat, tjekker som de andre, hvorvidt der er information i @Input når den køres og hvis der ikke er, afslutter og lader rssp\_Statistic køre videre.

Når der er indhold i @Input, starter proceduren med at oprette en global temptable til den specifikke type numre. Den tager forskel på Sender og Receiver og EAN, SORID og SKS/SHAK.

Når dette er gjort, startes et loop, som bliver ved med at køre, til alle numrene fra input er indsat i den oprettede global temptable.

Herefter tilføjes til @Select navnet på kolonnen i den nu oprettede og fyldte global temptable.

Når @From skal have information tilføjet, skal den tjekke hvor vidt der allerede er en join på t\_letterlog. Hvis der ikke er, skal den først tilføje en join til denne.

Herefter undersøges hvorvidt der er tilføjet en join til t\_OrganisationUnit og hvis der ikke er, tilføjes der til @From, et join mellem t\_letterlog og t\_OrganisationUnit\_LetterType\_MessageDirection og herefter en join mellem førnævnte tabel og t\_OrganisationUnit. Grunden til disse joins er at t\_OrganisationUnit ikke er direkte koblet med tabellen t\_letter, men man skal gennem de to tabeller for at komme til t\_OrganisationUnit.

Når @Select og @From er opdateret, føres de to variabler ligesom ved de andre metoder tilbage til rssp\_Statistic.

## rssp\_Statistic

Dette er hovedproceduren. Proceduren i sig selv har for det meste, til opgave at sende opgaverne videre til de tidligere nævnte procedurer. Proceduren er også den der tager mod parametre fra brugeren og den der besidder variablerne til at gemme de oplysninger (@Select, @From, @Where), til brug i den endelige query. Proceduren sender brugerens input ind i undermetoderne, sammen med en række statisk information, der forklarer procedurerne, hvad de skal tilføje til de tre førnævnte variabler.

Når rssp\_Statistic er færdig med at køre sine underprocedurer, tager den først og skærer det første tegn af strengen i @Select. Dette gøres fordi underprocedurerne i deres kørsel sikrer at der mellem hver kolonne der skal vises er et komma. På grund af dette vil der også i starten af @Select’s streng, være et komma. Hvis dette forbliver, vil der ikke komme noget ud, når @SQL1 køres med sp\_ExecuteSQL.

@SQL1 er en opsamlingsstreng, der skal samle alt indhold i @Select, @From og @Where og forme det til en sql streng der kan køres med sp\_ExecuteSQL.

Når @SQL1 er udfyldt, udfyldes @Paramdef, med den information, den skal have for at køre i sp\_ExecuteSQL med @SQL1.

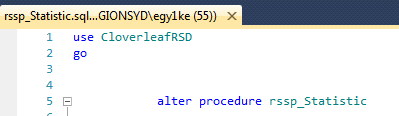
Afsluttende køres sp\_ExecuteSQL for at vise en query på de input brugeren har indtastet.

# Eksempel på installation og kørsel af rssp\_Statistic

For at installere proceduren rssp\_Statistic, skal du bruge både rssp\_Statistic.sql, men også de tre underprocedurer, som tager sig af det meste af funktionaliteten.

## Installation

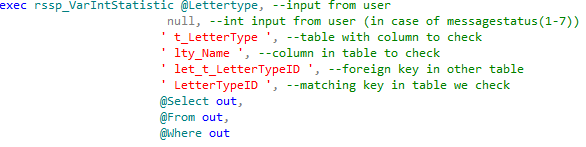
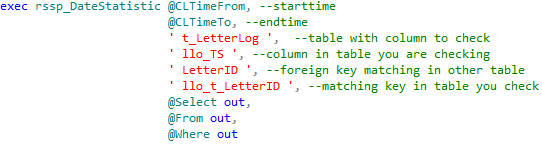
For at installere proceduren, skal de fire sql filer åbnes i Sql-Server Management studio (SSMS).

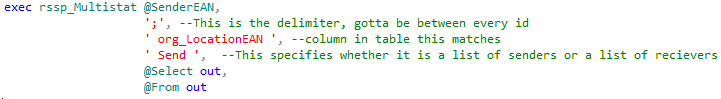


Der hvor der står alter procedure, ændres linjen til create. Dette vil få SSMS til at installere proceduren til din database. Dette skal ikke blot gøres for rssp\_Statistic, men også for rssp\_VarIntStatistic, rssp\_DateStatistic og rssp\_MultiStat.

Vil man senere tilføje funktionalitet til rssp\_Statistic, så den kan kigge på endnu flere kolonner, benytter man alter procedure igen, som vist i billedet ovenfor.

Såfremt tabel navne hedder noget andet på serveren, hvorpå de skal bruges skal understående ændres, før kørsel af proceduren.





som beskrevet i billederne ovenfor, svarer hver af parametrene, noget bestemt der skal følge med. Til Datestatistic hedder den ene af tabellerne med tider t\_letterlog, dette er angivet i tredje parameter og skal altid svare til tabellen man ønsker at kigge på. Underneden er også beskrevet hvad de næste parametre svarer til, dette går igen ved billederne under, som viser kørsel i rssp\_Statistic af varintstatistic og multistat.

## Kørsel

Når man efter installation af de proceduren rssp\_Statistic ønsker at kære den, skal der bruges en kommando, som kan se ud som følgende eksempel.

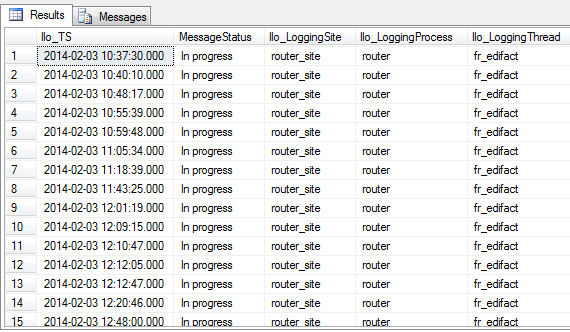


I dette billede har jeg valgt at kigge på 5 kolonner i tabellen letterlog. Den første, ses ved tredje og fjerde parameter, disse er tidsstemplet. Til tiden angiver du en start og en slut tid, du vil se. I eksemplet er starttiden, den 19. november 2013 og slut tidspunktet er den 20. november 2014, et tidsinterval på cirka et år. Tidsenhederne kan være når som helst og proceduren, vil finde alle tider mellem de angivede frem.

Femte parameter er lettertype og den sjette er messagestatus. Messagestatus har en int værdi mellem 1 og 7 og svarer til en status sat i letterlog, men forklaret i tabellen t\_messagestatus. 1 svarer til meddelelser, der er identificeret som ”in progress”.

Parameter 1,2 svarer til tid i t\_letter, parameter 3,4 tid i letterlog og 6 – 10 er andre kolonner i letterlog, med interessante værdier.

Resultatet af ovenstående query kan se nogenlunde ud som billedet nedenfor.



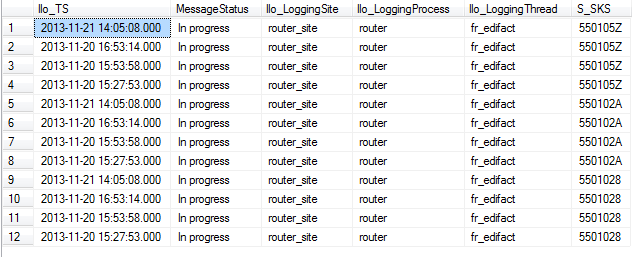
I det følgende billede, har vi prøvet at query på SKS numre, som kan matche information vi allerede har set.



Det der skal ligges mærke til, er at ved indsættelse af SKS numrene, er de separeret med semikolon. Dette gøres for at proceduren kan forstå at de er separate værdier, men at man ønsker at sammenligne dem alle, med de andre input parametre, som vi har udfyldt.

Her gøres opmærksom på at ved alle sendernumre, skal der separeres med semikolon og ved receiver, skal der separeres med kolon.

Nedenstående er et resultat af ovenstående query.



Som kan ses af billedet, viser den kun de punkter, hvor sks nummeret matcher de andre værdier, alle andre sks-værdier, som der blev indtastet i dets input parameter vises ikke.

## Yderligere forklaring på kørsel af rssp\_Statistic:

Såfremt begge parametre er udfyldt, køres proceduren, sender informationen - som beskrevet i rssp\_DateStatistic afsnittet – tilbage til rssp\_Statistic og denne går så videre til at køre en procedure for næste parameter, som brugeren har indtastet.

For at forsimple det kan man sige om kørslen af proceduren: ”er der to tidspunkter i sender tiderne? Hvis ja, kører vi indholdet i proceduren, ellers afslutter vi proceduren igen og går videre til at se på Cloverleaf tiderne”.

Det næste skridt fungerer ligeledes, ”er der indhold i parametrene til Cloverleaf tider? Hvis ja, kør indholdet af rssp\_DateStatistic, hvis nej, afslut proceduren igen”.

Vi kommer herefter videre til lettertype. Proceduren rssp\_VarIntStatistic startes hertil automatisk, hvis rssp\_Statistic kan se at indholdet af @Lettertype ikke er null. Hvad der sker inde i proceduren, er beskrevet i afsnittet rssp\_VarIntStatistic. Når proceduren er færdig med at køre, opdateres @Select, @From og @Where på samme vis, som ved kørsel af rssp\_DateStatistic.

Efter tjek på lettertype, tjekkes på MessageStatus, Servername, Site, Process og thread på samme maner og med samme procedure, som lettertype, dette såfremt deres parametre ikke er null, i hvilket tilfælde rssp\_Statistic springer over dem.

Set fra rssp\_Statistic foregår de efter følgende tjek på SenderEAN, SenderSORID, SenderSHAK, RecieverEAN, RecieverSORID og RecieverSHAK efter samme fremgangs måde. Det vil sige, hvis deres parameter er udfyldt, køres de, når rssp\_Statistic når til dem og ellers ikke. Efter kørsel af hver af disse, opdateres @Select og @From, men ikke @Where. Dette sker, da der i så fald skulle fastslås ét bestemt nummer til hver, frem for det den gør, nemlig at skabe midlertidige tabeller, som der til sidst skal joines på.

Ved at joine de foregående parametres tabeller, med de midlertidige tabeller, som skabes i rssp\_MultiStat findes de EAN/SORID/SKS numre som matcher informationen fra t\_Letter, t\_LetterType, t\_Letterlog, messagedirection og t\_OrganisationUnit.

Når det nævnes at de er midlertidige tabeller af typen global temptable, betyder det at så snart proceduren er færdig med at køre sp\_ExecuteSQL og har vist resultatet af queryen, bliver de midlertidige tabeller slettet.