Indhold

[Indledning 1](#_Toc400526973)

[Regional ITs struktur og infrastruktur 1](#_Toc400526974)

[Software til internt brug i Regional IT 2](#_Toc400526975)

[Citrix 2](#_Toc400526976)

[Microsoft Outlook 3](#_Toc400526977)

[MaximoSD 3](#_Toc400526978)

[Cloverleaf 3](#_Toc400526979)

[Arbejdet 3](#_Toc400526980)

[Hvad er T-SQL? 3](#_Toc400526981)

[EER-Diagrammer 3](#_Toc400526982)

[Dataflow 3](#_Toc400526983)

[Subversion 4](#_Toc400526984)

[Databasen 4](#_Toc400526985)

[Simple Queries 4](#_Toc400526986)

[SQL funktioner 4](#_Toc400526987)

[Rsfn\_UnixTimeToDateTime og Rsfn\_DateTimeToUnixTime 4](#_Toc400526988)

[SQL Views 4](#_Toc400526989)

[Stored Procedures 5](#_Toc400526990)

[God praksis 5](#_Toc400526991)

[Insert ”route” procedure 5](#_Toc400526992)

[Insert i System procedure 5](#_Toc400526993)

[Statistic proceduren 6](#_Toc400526994)

[Rssp\_VarIntStatistic 6](#_Toc400526995)

[Rssp\_DateStatistic 6](#_Toc400526996)

[Rssp\_MultiStat 6](#_Toc400526997)

[Rssp\_Statistic 6](#_Toc400526998)

[Udbytte 6](#_Toc400526999)

[Konklusion 6](#_Toc400527000)

# Indledning

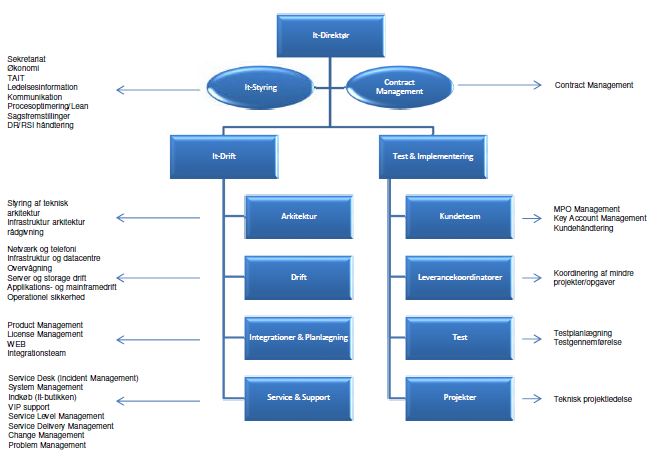
Region Syddanmark er Danmarks tredje største region, med mere end 1,2 millioner indbyggere. I denne region er der flere store hospitaler: Sydvestjysk Sygehus I Esbjerg, Sygehus Lillebælt, Sygehus Sønderjylland I Sønderborg og Odense Universitetshospital.

Sygehusene har hver en IT infrastruktur og har meddelelser af alt fra booking tider, til patient journaler, som skal sendes til forskellige undersystemer på enten hospitalet selv, en lokal klinik, et andet hospital, et laboratorie eller andet inden for sundhedssektoren og kommunerne i Region Syddanmark.

Softwaren, som lægerne, sygeplejerskerne, laboranter og andet personel på sygehusene skal benytte, udvikles til dels af regionens Sundheds IT-stab, som også varetager vedligeholdelse og udvidelse af disse programmer. Til denne brede IT-stab var jeg praktikant i afdelingen Integration og Planlægning. I denne rapport vil jeg gennemgå noget af den software, som arbejdes med i Integration og planlægningsafdelingen. Jeg vil med dette prøve at give et indblik i hvordan Regional IT drives.

Jeg vil herefter komme ind på det jeg arbejdede med, i praktik perioden mandag den 11. august 2014 til 10. oktober 2014. Hvilket programmeringssprog jeg benyttede, hvilket udviklingsmiljø jeg programmerede det i og hvilke opgaver jeg lavede i for virksomheden i praktikperioden.

# Regional ITs struktur og infrastruktur



Figur : Region IT struktur. Kilde: regionsyddanmark.dk/dwn363741

Regional IT er opbygget I en struktur, med en direktør, diverse underledere og herunder medarbejdere.

Direktøren i Regional IT hedder Søren Lindegaard og hans opgaver består i kommunikation og bestemmelse af retninger, som har indflydelse på hele Regional IT. Dette kan være alt fra bestemmelser om hvilke systemer, som Regional IT skal køre, hvor hurtig disse skal rulles ud og hvorfra de skal indkøbes.

Under direktøren er to overafdelinger: IT-Drift og Test og Implementering

I IT drift er Flemming Brink afdelingschef.

IT-Drifts ansvar er bedst beskrevet fra regionens egen hjemmeside:

”Hovedansvaret i IT-Drift er at levere stabile og pålidelige it-services til forretningen med stor leverancesikkerhed og effektivitet. IT-Drift varetager forvaltning af servere og infrastruktur samt produkter og applikationer.

IT-Drift er ansvarlig for at sætte strategiske projekter fra Sundheds IT og taktiske projekter fra Regional IT. Desuden identificerer og prioriterer IT-Drift forbedringstiltag til it-strategien.

Derudover har IT-Drift ansvaret for at levere procedurer til Sundheds IT for idriftsættelse af projekter. Endelig rapporteres økonomien til It-Styring i forhold til drift, tillægsydelser og taktiske projekter.”   
Kilde: <http://regionsyddanmark.dk/wm345257>

I IT-drift, var jeg i praktik i den underafdeling der hedder Integration og Planlægning. Denne afdeling styres af Funktionschef Henrik Juul Andreasen, med hjælp fra Chefkonsulent Morten Thuesen og Teknisk Koordinator Thorkild Friis.

Integration og planlægnings afdelingen tager sig af små som store problemer, der involverer software, databaser og servere der benyttes af sundhedssektoren i regionen. Softwaren kan være alt fra programmer til patient tids booking til patient kritisk software, som integration til COSMIC, der har adgang til PAS (Patient Administration System) og EPJ (Elektronisk Patient Journal). Det er af denne grund at Regional IT’s ansatte er underlagt fortroligheds erklæring, da der i de to patient systemer ligger utroligt meget data omkring regionens borgere.

# Software til internt brug i Regional IT

Internt i Regional IT benyttes en række fælles software, for at gøre arbejdet lettere, herunder er et par af programmerne beskrevet.

## Citrix

Citrix er et centraliserings løsning, som giver adgang til programmer, som alle I Regional IT vil få brug for. Her tænkes programmer, som Microsoft Office-pakken, Adobe Reader og incident management Systemet MaximoSD. Citrix løsningen er valgt, som en hjælp til at holde programmer opdateret til de seneste versioner og de Udgaver af produkterne, som Regional IT har licenser til.

## Microsoft Outlook

Dette program, som også er en del af officepakken der ligger i Citrix, benyttes både til intern mail, hvor alle ansattes arbejdsmail er registreret, samt deres mødekalendre. Med det kan også bookes møder, hvor til man inviterer diverse andre ansatte og arrangerer hvilket lokale mødet skal foretages i.

## MaximoSD

MaximoSD er et incident management system, baseret på en ITIL workflow, programmet kan bla. Håndtere følgende:

* Changes – ændringer der er foretaget i programmer til regionen, foretaget af regionens software udviklere.
* Problems – problemer der er opstået med software, servere, computere for personer i regionens ansatte i Sundheds IT, Regional IT, hospitals ansatte og lægehus ansatte.
* Incidents – større uheld der er sket i forhold til software eller længere varende problemer, hvor rodårsagen skal løses.

## Cloverleaf

Cloverleaf er regionens primære integrations platform, som også fungerer som message broker. Den skal tage mod meddelelser af forskellige formater og oversætte dem, så de kan forståes af andre systemer.

Desuden fungerer den også som messagebus, hvilket vil sige den skal route meddelelser mellem forskellige systemer i sundhedssektoren. Til cloverleaf er der tilhørende databaser, som holder styr på alle meddelelser, som oversættes og sendes til forskellige områder i sundhedssektoren.

# Arbejdet

Arbejdet I Integration og planlægnings afdeling har for mig bestået af kodning af Transact-SQL opgaver i Microsoft SQL Server Management Studio (SSMS).

## Hvad er T-SQL?

Transact-SQL er det programmeringssprog, som benyttes, når der i SSMS skal kodes functions og stored procedures. Der er bl.a. i transact sql mulighed for benyttelse af TRY-CATCH, så der i løbet af kørsel af kode kan fanges fejl og vises fejlmeddelelser der er forståelig for den almene bruger og så systemet ikke går ned, såfremt en fejl opstår.

## EER-Diagrammer

Til relationelle databaser er det almen praksis at man laver EER diagrammer over systemet, for at give et indblik i hvordan tabellerne hænger sammen. Dette hjælper skaberen og andre programmører, som skal arbejde med databasen, med at forstå hvordan tingene i den hører sammen.

## Dataflow

Udover EER diagrammer blev der også laver dataflow diagrammer, for at give andre udviklere en ide om, hvordan de forskellige procedurer og funktioner fungerer. Dataflow minder meget om hændelses diagrammer og begge inkorporerer hændelse og handling.

## Subversion

For at holde styr på elementer som er tænkt som færdige eller skal arbejdes på af flere medarbejdere, benyttes subversion til at holde styr på filerne der arbejdes på. Dette gør det muligt at holde øje med hvad der er blev ændret hvor i filer og hvis der er sket uønskede ændringer, så foretage rollback, så man kommer til en version, hvor filen er i en tidligere version.

## Databasen

Cloverleaf databasen, som jeg arbejdede med er en relationel database, som via forskellige tabeller holder styr på meddelelsestrafikken. Via dens forskellige tabeller kan ses både hvem afsender og modtager er, fra hvilket system de har sendt det, til hvilket system de har sendt det, hvad tid de sendte det, hvornår meddelelsen kom ind i cloverleaf og hvorvidt meddelelsen blev leveret til modtageren.

## Simple Queries

De første opgaver bestod i at foretage simple queries på tabeller i en Medcom database med info om meddelelser sendt mellem afdelinger. Den første skulle gøres mellem tre tabeller, hvori den ene, var en relationstabel mellem de to andre. Tabel1 ville have information omkring modtageren af brevet, samt et id matchende til denne, Tabel2 havde så information omkring meddelelsestype, samt til hvilken lokation meddelelsen skulle sendes. Relations tabellen indeholdt information der henviste til, om hvorvidt en meddelelse var kommet igennem. Såfremt der var foreign keys fra tabel1 og tabel2 i samme række, betød det at meddelelsen var kommet igennem.

Denne query, blev så gemt som i et excel dokument og sendt videre til en udvikler, som skulle bruge dataen til en opgave han var i gang med.

## SQL funktioner

Det næste der skulle laves, var to SQL funktioner, der skulle konvertere en dato kolonne, enten fra unix-time eller til unix time. Unix time er en tidsenhed, der måler antal sekunder der er gået fra Midnat 1. januar 1970 til et givet tidspunkt længere fremme i tiden.

I følgende underafsnit vil disse være beskrevet.

### Rsfn\_UnixTimeToDateTime og Rsfn\_DateTimeToUnixTime

Den første funktion kaldet rsfn\_UnixTimeToDateTime tog en unixtid ind, hvorefter den i en variabel kaldet @datetimeoffset gemmer forskellen mellem UTC tid og lokal tid.

Den næste variabel (@ALDT) tager så tidsforskellen gemt i @datetimeoffset og trækker fra vores inputs tid

Afsluttende, returnerer funktionen en tid i normale tidsenheder, beregnet ved hjælp af den indbyggede funktion dateadd, som skal bruge en tidsenhed (eksempelvis sekunder).

Rsfn\_DateTimeToUnixTime foretog så konverteringen den modsatte vej, den eneste større forskel her, var at UTC tid, så skulle lægges til i stedet for at trække den fra.

## SQL Views

Det næste der skulle laves var et view. Det er en funktionalitet, hvorved man kan give en ikke-administrator bruger adgang til at se visse tabeller i en database, uden egentlig at give dem adgang til andet end blot at kigge på indholdet af disse tabeller og deres fælles data. I et view kan man sågar administrere hvilke kolonner i tabellerne, som brugeren skal have adgang til at se.

I det view jeg skulle lave for Integrations teamet i Regional IT, skulle der vises tre tabeller: t\_organisationunit\_lettertype\_messagedirection (routes), t\_organisationunit og t\_system.

Et view skabes ved at benytte ”create view (viewnavn) as” efterfulgt af en query på de kolonner i de tabeller man ønsker vist. Når man har hele sin query, afsluttes med ”go”.

## Stored Procedures

Det jeg brugte mest tid på i praktik perioden, var at skrive stored procedures. Stored procedures er som navnet hentyder gemte procedurer, procedurer der udføres på serversiden, af den specifikke database. Procedurer kan forsimple opgaver både direkte på serversiden, men også i eventuelle programmer, som kobler op til serveren og databasen. Et af formålene, med en stored procedure, er at gøre det lettere for udviklere af programmer, som skal koble sig op mod databasen, at gøre de operationer der er nødvendige i forhold til programmet. Eksempelvis vil man ved hjælp af en stored procedure kunne spare en del linje koder i c#, hvis man skal indsætte eller update en eller flere tabeller i en database. Stored procedures kan kodes sådan at programudvikleren blot i sit program, skal angive at databasen skal eksekvere proceduren, med dens parametre, hvorefter den vil indsætte eller opdatere tabellerne. Program udvikleren, behøver dermed ikke kode hvilke tabeller der skal indsættes i eller opdateres i databasen.

Stored procedures kan kodes til at foretage CRUD, enten alle kommandoerne eller enkelte af dem.

I omfanget af praktiktiden, blev der udarbejdet seks procedurer, hvoraf de tre var procedurer, som skulle køres i en anden procedure. Efter afsnittet god praksis, vil disse blive beskrevet i hver deres afsnit.

### God praksis

Blandt de første ting der skulle læres, når der skulle kodes procedurer, var hvordan man ved hjælp af god kode skik kunne teste for mange problemer, som kunne opstå ved kørsel af procedurer.

Til dette har Regional IT udviklet en standard, for hvordan procedurer bør kodes, som involver testcases og errorlogging til en tabel. På den måde kan man også senere gå ind at se hvordan og hvor problemet med kørslen af proceduren opstod.

### Insert ”route” procedure

Formålet med insert route proceduren er at den skal hjælpe med at kunne indsætte nye ”routes”, det vil sige indsættelse af information i tabellen t\_OrganisationUnit\_LetterType\_Messagedirection.

Denne opgave tog cirka halvanden til to uger, da jeg først skulle sættes ind i hvordan stored procedures virker. Jeg skulle desuden lære god praksis, ved opbygning af disse og ikke blot sikre mig at den virkede, men også sikre den mod bruger fejl, såsom manglende input eller forkert input.

### Insert i System procedure

Proceduren til indsættelse af rækker i t\_system ligner meget proceduren til indsættelse af rækker i routes. Forskellene på de to procedurer ligger i hvor meget der skal testes på og hvor mange kolonner der skal indsættes på.

## Statistic proceduren

Statistic proceduren var en større procedure, som skulle være i stand til at vise statistiske opslag på meddelelserne i Cloverleaf. Brugeren skulle derfor kunne angive en lang række input parametre, som de ville se på, heriblandt brevtype, tider for afsendelse, modtagelse, samt hvorvidt meddelelsen var kommet igennem til modtageren. Brugeren skulle desuden i flere parametre kunne angive en lang række numre, som svare til indhold i nogle kolonner i tabellen t\_OrganisationUnit.

Tiden det ville tage at lave denne, var anslået til mellem 6 og 8 uger, her inkluderet indlæringstid til forståelse af nye funktioner. Proceduren fik navnet rssp\_Statistic.

### Rssp\_VarIntStatistic

Denne var en af underprocedurerne jeg lavede til rssp\_statistic proceduren, dens formål er at tage et enkelt input af typen nvarchar eller int og undersøge om de er korrekte. Hvis de er korrekte, skal den så tilføje dem til sql strenge, som sendes tilbage til rssp\_statistic, så denne kan query på dem.

### Rssp\_DateStatistic

Denne underprocedure tog sig af datoer, den modtager to datoer fra brugeren, tilføjede dem til de eksisterende sql-strenge og ligeledes sende dem tilbage til rssp\_statistic, så den kan query på dem.

### Rssp\_MultiStat

Denne tog sig af input der bestod af flere numre adskilt af en seperator. Den tager numrene, deler dem og indsætter dem i temp tables, som så sammen med de eksisterende sql-strenge sendes tilbage til rssp\_Statistic.

### Rssp\_Statistic

Ved kørsel af denne, sender den informationerne videre til de passende underprocedurer, alt efter hvilke parametre brugeren har udfyldt. Når disse har modtaget informationen, behandler de den og såfremt alt er korrekt, sender de nogle streng variabler tilbage til denne hoved procedure, som så til sidst foretager en query ved hjælp af sp\_ExecuteSQL, på det som brugeren har indtastet.

# Udbytte

I praktik tiden har jeg lært en utrolig masse omkring Microsofts SQL-server, som jeg ikke var bekendt med før praktik opholdet. Jeg har lært at kode functions i SQL, lave views, foretage udtræk til sql, lært at kode stored procedures og lært god praksis i forhold til test og errorhandling, som kan bruges i Transact-SQL, men også i andre programmeringssprog.

# Konklusion

Et spændende ophold i Regional IT, som lærte mig en masse omkring virksomhedens opbygning og SQL.