

Göteborgs universitet  
Institutionen för informatik

## **Resurshantering av personal – relationsdatabas eller kalkylsystem?**

av

Erika Lünig  
VT-99

Handledare: Roy Corneliusson      Göteborgs universitet  
Jonas Lindholm                      Ericsson Mobile Data  
Design AB

**Examensarbete 10 p för ADB-programmet 80 p**

### **Abstrakt**

Att lösa komplexa administrativa problem med hjälp av kalkylsystem eller relationsdatabassystem är ett ställningstagande som får göras från fall till fall i systemutvecklingsprocessen. I ett konkret fall har ett programvaruutvecklingsföretag – ett bolag inom Ericssonkoncernen – gjort försök att lösa hanteringen av sina resurser med hjälp av ett standard kalkylsystem, men funnit resultatet icke-tillfredsställande p.g.a. problemomänens många dimensioner. Företaget förväntar sig minska sina kostnader, få snabbt information om dagens resursläge, och lättare kunna planera morgondagens resursbehov. Denna rapport redovisar dels skillnaderna mellan de två vägarna - kalkylsystem eller relationsdatabas - ger en gemförelse av de två, och visar en kravspecifikation och datamodell på en relationsdatabaslösning. Slutsatsen av min studie ger vid handen att ett resurshanteringssystem baserat på en relationsdatabas lämpar sig bättre än ett kalkylsystem på grund av dess flexibilitet, enkelhet, begriplighet och användarvänlighet.

---

Examensarbete 10p för ADB-programmet 80p

**Förord**

Jag vill rikta ett speciellt tack till Roy Corneliusson och Jonas Lindholm, utan vilkas hjälp och tålamod detta arbete inte hade kunna slutföras.

**Innehållsförteckning**

1	Introduktion	5
2	Metod	7
2.1	Bakgrund	7
2.2	Insamling av förutsättningar	7
2.2.1	Intervjuer	7
2.3	Litteraturstudier	8
2.4	Val av databas produkt	8
2.5	Procedur (utvecklingsprocessen)	9
2.5.1	Etapp 1: Genomgång av resurshanteringsprocess	9
2.5.2	Etapp 2: Behovsinventering	9
2.5.3	Etapp 3: Modellering av databas	9
2.5.4	Etapp 4: Implementeringsfasen	10
2.5.5	Etapp 5: Verifieringsfasen	10
2.5.6	Etapp 6: Frisläppande/Införande	10
3	Resultat	11
3.1	Jämförelse av verktygen kalkylsystem och relationsdatabas	11
3.1.1	Definition av verktygen	11
3.1.2	Kapacitet	12
3.1.3	Komplexitet	12
3.1.4	Händelser i systemet	12
3.1.5	Definition av rapporter	13
3.1.6	Säkerhet	14
3.1.7	Databehandling	15
3.1.8	Kommunikationsfrågor	16
3.2	Verksamhetsanalys	16
3.3	Intervjuer	17
3.4	Krav på Systemet	18
3.4.1.1	Allmänt	18
3.4.1.2	Aktörer i Systemet	19
3.4.1.3	Användningsfall	21
3.5	Design av databas	22
<b>3.5.1.1</b>	<b>Resurs</b>	25
<b>3.5.1.2</b>	<b>Projekt</b>	25
<b>3.5.1.3</b>	<b>Sektion</b>	26
<b>3.5.1.4</b>	<b>Månadsäskande</b>	26
3.5.2	Normalisering	26
3.5.2.1	Första normalformen	26
3.5.2.2	Andra normalformen	28
3.5.3	Modellering	28
4	Diskussion	30
	Varför kalkylbladssystem?	31

Resurshantering av personal – relationsdatabas eller kalkylblad?

---

5	Referenser	32
6	Bilagor	34
6.1	Kalkylblad från det ursprungliga resurshanteringssystemet	34

## 1 **Introduktion**

Att hantera resurspersoner i en organisation där flera samtidiga projekt bedrivs är en uppgift som kan lösas med hjälp av någondera redskapen databas eller kalkylsystem.

Innan man bestämmer sig vilket av dessa verktyg som är lämpligast i olika problemsituationer, bör man studera dessa verktyg närmare enligt vissa utgångspunkter. Två verktyg – Microsoft Excel och Microsoft Access – har jämförts och resultatet av denna jämförelse är redovisad i för i början av kapitlet Resultat (3). Som resultat av mina observationer redovisar denna rapport de aspekter som bör beaktas för att lösa detta tredimensionella – d.v.s. resurs, projekt samt kalendertid – problem.

Förutom att redogöra för de krav som ställs på ett resurshanteringssystem – härnå kallat "system" – avser denna rapport att ställa lösningarna databas och kalkylsystem mot varandra.

Detta arbete har utförts under våren 1999 på uppdrag av företaget Ericsson Mobile Data Design AB, Göteborg.

Ericsson Mobile Data Design AB – härnå kallat ERV - är ett systemutvecklingsbolag med ca. 450 anställda som alltså vill skapa ett verktyg som håller ordning och ger överblick hur deras resurser är nyttjade i olika projekt.

Företagets verksamhet består i huvudsak av utveckling av mobila paketdatasystem, och är i Ericsson-gruppen utsett som kompetenscenter för forskning och utveckling av denna typ av system.

Av de anställda är ungefär 250 st. sysselsatta med systemutveckling. Arbetet bedrivs i projekt med varierande antal resurser både med avseende på omfattning funktioner och löptid.

För att på bästa möjliga sätt utnyttja denna pool av resurspersoner ses behovet av ett resurshanteringssystem som hjälper projektledare och linjeorganisationschefer att på effektivaste sätt kunna nyttja och prognostisera ERVs personal.

Tills nu har det på ERV gjorts försök att skapa ett sådant system med hjälp av ett kalkylsystemprogram - Microsofts 'spreadsheet'-program Excel - men inte lyckats lösa detta tredimensionella problem med detta verktyg. Systemet klarade inte av de olika dimensionerna med kopplingar och länknings. Det 'kraschade' helt enkelt.

## Resurshantering av personal – relationsdatabas eller kalkylblad?

---

I stället har det övervägts att finna lösningen med hjälp av en relationsdatabas och i denna rapport skall – som redan sagts – redogöras för hur detta kan åstadkommas.

## 2 Metod

### 2.1 Bakgrund

Med bakgrund av vad som redovisats i introduktionen skall denna rapport utreda om det går bättre att använda relationsdatabas än det nuvarande kalkylbladssystemet, och – om så är fallet - varför databaslösningen lämpar det sig bättre, d.v.s. diskutera för- och nackdelarna med de två lösningarna.

Vidare, kommer denna rapport redogöra för en genomförbar databaslösning på problemet som således bygger på Microsoft Access.

### 2.2 Insamling av förutsättningar

Den interna webbsidan på ERV har studerats. Där står information om företagets organisation och resurshanteringsrutiner.

Med denna information och intervjuer med att antal chefer på ERV har underlag erhållits till de krav som ställs på systemet. En fullständig kravspecifikation ges under kapitel **Krav på Systemet** (3.4) nedan.

Därefter har formulären kalkylsystemet studerats från den existerande Excel-lösningen för att ge en inblick i företagets nuvarande resursdokumentation.

Därpå har de involverade personerna intervjuats. Dit hör projektledare, avdelningschefer samt sektionschefer. De har dels redogjort för det nuvarande rapporteringssystemet, och dels återgett sina förväntningar på det kommande systemet (alltså databaslösningen).

#### 2.2.1 Intervjuer

Efter att jag bekantar mig med det befintliga systemet genomförde jag intervjuer med aktörerna, d.v.s. de som skall använda av systemet, sålunda cheferna och administratören.

Nedan redovisas för deras åsikter om det nuvarande Excel baserade systemet och deras förväntningar om det nya.

Följande frågor ställdes:

- Hur får du informationen idag?
  - Vilken information lämna du ifrån dig idag?
  - Vilken information skulle du vilja ha?
  - Hur skulle du vilja lämna information ifrån dig?
-

**2.3 Litteraturstudier**

Vid detta skede – efter studier av ERV dokumentation och intervjuer med ERV personal - fanns tillräckligt med fundament för att genomföra modelleringen av databasen.

Inför modelleringen bedrevs litteraturstudier i ämnet.

Först och främst hämtades information om utformning av databaser ur kapitlet 8, ”*Logical Database Design – Worked Example*”, i [3] (sidorna 275 - 322).

Vidare har skrifterna [12] och [8] varit till stor hjälp vad gäller att öka förståelsen för skapande av databaser.

Vad gäller Microsofts kalkylsystem och relationsdatabas har vidare litteraturstudier bedrivits, se kapitel 2.4 nedan.

Dessutom studerades KRB-metoden, se [2], och diverse litteratur om verksamhetsanalys.

**2.4 Val av databas produkt**

För att kunna omsätta de tilläggande kunskaperna i praktiken bestämdes vilken konkret databas som skulle användas i det fortsatta arbetet.

Här gjordes det naturliga valet Microsoft Access, eftersom denna produkt finns tillgänglig på ERV. Alla PC-datorer på ERV har Microsoft Office – i vilken Access ingår – standardmässigt installerat.

Vidare underlättade detta val jämförelsen mellan kalkylsystem- och relationsdatabas- lösningarna av den orsaken att de båda produkterna – Excel och Access – härstammar från samma leverantör.

Ytterligare litteraturstudier bedrevs – av skrifterna [7], [6] samt [10] - för att fördjupa kunskapen i Microsofts produkter.

Här bör speciellt nämnas att kapitlet 22, ”*Exploring Relational Database Design and Implementation*”, i [7] (sidorna 773 - 820) innehåller inte bara värdefull information vad gäller Microsoft Access, utan ytterligare en redogörelse för hur databasmodeller kan modularas fram.

Vid mitt examensarbete har jag haft mycket stöd av det jag har lärt mig vid institutionen i Informatik ADB-programmet. Särskild de kurser vi hade i programmering, systemutveckling och framför allt kursen i databaser och SQL, där vi använde Microsoft Access som verktyg.

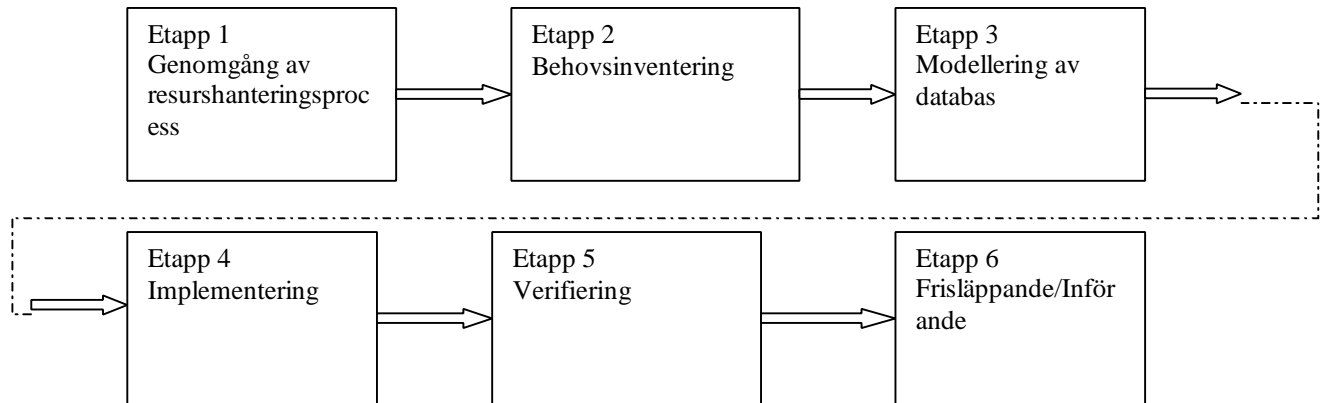
Slutligen har Microsofts webb varit en utgångspunkt för informationsinhämtandet om Excel och Access. Här har dokumentet [9] varit till nytta.



## 2.5 Procedur (utvecklingsprocessen)

Utvecklingsprocessen för resursplaneringsdatabasen på ERV är indelad i ett antal faser. Figur 1 nedan illustrerar dessa etapper.

Etapperna omfattar alla steg som är nödvändiga för att erhålla en slutgiltig produkt. Det som utförts i samband med utformandet av denna rapport omfattar endast etapperna 1 till 3.



**Figur 1 Systemutvecklingens etapper.**

### 2.5.1 Etapp 1: Genomgång av resurshanteringsprocess

Genomgången av den nuvarande resurshanteringen gjordes med hjälp av ansvarig chef för projektsektionen<sup>1</sup> samt dokument på ERVs interna webb.

### 2.5.2 Etapp 2: Behovsinventering

Identifiering av krav på systemet.

Behovsundersökningen genomfördes i form av intervjuer med berörda personer, som gav en bild av verksamheten och ett tillräckligt underlag för den första prototypen. De första två etapperna är underlag för databasmodellen, se kapitel 2.5.3 nedan.

### 2.5.3 Etapp 3: Modellering av databas

Här sker utveckling av objekt och databasmodellen.

Kalkylbladen från nuvarande (ursprungliga) systemet samt information erhållen från intervjuerna utgjorde utgångspunkten för detta arbete. Objektrelationerna, egenskaperna (attribut) bestämdes samt nycklarna klassificerades.

<sup>1</sup> På ERV har man en specifik sektion vilken har hand om de projekt som bedrivs.

#### 2.5.4 Etapp 4: Implementeringsfasen

Under implementeringsfasen börjar man se över befintlig teknik och utvecklingsmiljö. Den teknik och utrustning som ingår i lösningen beställs, bevakas och installeras. Införandet av lösningen detaljplaneras för att minimera de problem som kan uppstå. Därefter kontrolleras alla förutsättningar för införandet och en systemtest utförs.

Under implementeringsfasen görs den logiska och fysiska utformningen. Här skapas tabeller och dess relationer med hjälp av verktyget Microsoft Access.

#### 2.5.5 Etapp 5: Verifieringsfasen

Här verifieras att prototypen som färgigställts uppfyller de krav som ställs på systemet.

#### 2.5.6 Etapp 6: Frisläppande/Införande

Efter eventuella förbättringar frisläppande<sup>2</sup> av systemet.

---

<sup>2</sup> Ericsson-slang för utgivning av en produkt, d.v.s. att produkten är färdig och kommersiellt tillgänglig.

---

### 3 Resultat

#### 3.1 Jämförelse av verktygen kalkylsystem och relationsdatabas

Nedan följer jämförelser på ett antal punkter vad gäller användandet av ett kalkyl- och relationsdatabassystem för att det specifika problemet: Resurshantering.

Konkret har verktygen Microsoft Excel 97 och Microsoft Access 97 jämförts.

##### 3.1.1 Definition av verktygen

###### **Kalkylsystem:**

Utgörs av ett godtyckligt antal tabeller – tvådimensionella – där innehållet i varje enskild cell – d.v.s. varje korsning mellan tabellens rader och kolumner – kan bero av innehållet i andra celler från annan eller samma tabell. Detta beroende kan beskrivas med aritmetiska uttryck. Det är faktiskt möjligt att skapa databas i Excel, men då den första design av databasen är gjord med användandet av den aktuella datamängden och inte gjord med hänsyn på den förväntade resultatet.

I motsats till databasen här är det vanligt med upprepade data , tex samma kolumn flera gånger.

###### **Relationsdatabas:**

En databas definieras enligt [6] - kapitel 14, sidan 383 – som: ”*In its simplest form, a database is an organised list*”.

En relationsdatabas är ett antal listor- eller tabeller – vilkas rader har en eller flera relationer med rader i andra tabeller. De viktigaste reglerna är att primärnycklar – d.v.s. kolumner med unika värden - aldrig får odefinierade värden, medan referenser(främmande nycklar) antingen har ett odefinierat värde eller ett värde som motsvaras av en primärnyckelvärd från en rad i en annan tabell.

En ytterligare definition – enligt [4] - är:

”*A database is a collection of interrelated data and everything that is needed to maintain and use it*”.

Grundegenskaperna av en databas skall vara

- Allt data måste presenteras i tabellform
- Alla data måste vara odelbara

- Inga dubletter är tillåtna

### 3.1.2 Kapacitet

Vad gäller valet mellan ett kalkylsystem och relationsdatabas, måste beaktas att ett konkret kalkylsystem som Microsoft Excel har en begränsning av antalet poster som kan lagras. Se även [6] kapitel 14 sidan 386.

#### **Kalkylsystem:**

Microsoft Excel kan lagra ett begränsat antal poster. Denna mängd limiterad till 65535 ( $2^{16}-1$ ).

#### **Relationsdatabas:**

En relationsdatabas som Microsoft Access erbjuder möjligheten att lagra över 2 miljarder ( $\sim 2^{31}$ ) poster.

### 3.1.3 Komplexitet

För den specifika problemdomänen – d.v.s. resurshanteringsystemet – finns ett antal relationer – bl.a. projekt till resurs, resurs till sektion, samt resurs till tid – som skall upprätthållas.

#### **Kalkylsystem:**

Problemet är följaktligen att innehållet i posterna – d.v.s. fälten - har komplex relation med andra rutor andra kalkylblad. Visserligen stöds detta av Excel, d.v.s. att innehållet i en tabells fält kan formas av uttryck som baseras på innehållet i andra tabellers fält. Men om information skall hämtas från många olika ställen och kombineras – som i detta konkreta resurshanteringsystem – blir uttrycken antingen svårbegripliga eller rentutav omöjliga.

Vidare bör speciellt nämnas att robustheten är mindre i ett kalkylbladsbaserat system med många komplexa länkar. Kalkylsystemet är till följd därav mer känsligt för fel. I kapitel 3.3 nedan redovisas även svårigheter av denna art som ERV stött på i sitt försök att använda Excel.

#### **Relationsdatabas:**

Till skillnad från kalkylsystemet är data i en relationsdatabas separerade från vyn av dessa, d.v.s. data kan organiseras efter de villkor som identifieras i problemdomänen. Detta skänker en oinskränkt möjlighet organisera data på ett godtyckligt sätt utan att sålunda ta hänsyn presentationen.

### 3.1.4 Händelser i systemet

Händelserna i systemet identifieras utifrån de i kapitel 3.4.1.3 redovisade användningsfallen.

## Resurshantering av personal – relationsdatabas eller kalkylblad?

---

Systemet drivs helt och hållet av handlingar som utförs av systemets användare. Dessa aktioner klassas i kategorierna:

- **Registrering:** Systemet förses med uppgifter i samband med administration, äskande, godkännande, etc.
- **Rapportering:** se kapitel 3.1.5 nedan.

Speciellt i samband med registreringen måste de uppgifter som anges valideras enligt förutsättningarna givna regler.

### **Kalkylsystem:**

Om flera uppgifter skall anges och dessa hör hemma i olika kalkylblad – 'lakan' – kan valideringen bli komplex eftersom data placeras ut på flera ställen.

I Excel är det möjligt att skriva Visual Basic-'script' som via ett dialogfönster - gestaltande ett formulär - erhåller indata från användaren. Data kan därmed – före utplacering i celler – valideras, se även kapitel 32 i [6].

### **Relationsdatabas:**

I Access är det en naturlig del att via formulär – som utformas - erhålla data från användaren. Dessa data uppdaterar sedan direkt eller indirekt passande tabeller i relationsdatabasen, se även .kapitel 7 i [10]

### 3.1.5 Definition av rapporter

När det gäller att generera rapporter i ett system, skiljer sig uppbyggnaden mellan relationsdatabasen och kalkylsystemet.

Premisserna är således – enligt kraven redovisade i kapitel 3.4.1.3.6, 3.4.1.3.7 och 3.4.1.3.8 nedan - att systemet skall kunna frambringa följande rapporter:

- Projektrapport, se även Figur 5 i kaptitel 0 nedan.
- Sektionsrapport, se även Figur 6 i kaptitel 0 nedan.
- Avdelningsrapport

### **Kalkylsystem:**

Här utgör själva kalkylbladen rapporterna, d.v.s. de speglar direkt de rapporter som önskas erhållas. Men att få de avsedda värdena i kalkylbladens celler kan – som tidigare redogjorts för i kapitel 3.1.3 ovan – vara problematiskt.

Förvisso kan man med hjälp av Visual Basic-'script' iterera över ett eller flera kalkylblad för att skapa en begärd rapport. Likväl kan det vara så att man vill registrera data i en sådan rapport-vy.

## Resurshantering av personal – relationsdatabas eller kalkylblad?

---

Som i alla Microsoft Office verktyg – där Access och Excel ingår – finns i Excell 'trollkarlar' – "wizards" – som hjälper en att skapa rapporter, se appendix A i [6]. Dock är dessa främst av grafisk art – stapel- och pajdiagram etc. – och passar därför inte för resurshanteringssystemet.

### **Relationsdatabas:**

Till skillnad från kalkylsystemet erhålls inte de önskade rapporterna direkt av datarepresentationen i relationsdatabasen – d.v.s. tabellerna – utan rapporterna skapas genom att kopplade SQL-frågor ställs till databasen. Resultaten från en sådan förfrågan kan därefter utan svårighet formateras och presenteras.

Det bör också nämnas att rapporter alternativt kan skapas med hjälp av 'visuella verktyg' i Access utan att man ställer direkta SQL-frågor. Access innehåller en rapportdel som arbetar enligt principen **WYSIWYG** ("What You See Is What You Get"), se kapitel 12 i [10].

### 3.1.6

#### Säkerhet

Avseende säkerheten – d.v.s. att behörighetsmässigt särskilja användare av systemet – skiljer sig kalkylsystemet, Excel, och relationsdatabasen, Access, väsentligt.

I det specifika fallet – d.v.s. resurshanteringssystemet – behövs dels fyra kategorier av användare (angivna i kapitel 3.4.1.2): Administratör, projektledare, sektionschef samt avdelningschef.

Dessa tillhör således varsin grupp, och dessutom skall man kunna särskilja varje användare på individnivå.

### **Kalkylsystem:**

I Excel kan man skydda enskilda celler eller hela filer med ett lösenord, se även kapitel 23 i [6]. Emellertid är upplösningen – d.v.s. antalet användarkategorier – begränsat till två grupper: De som känner, och de som inte känner lösenord.

Med de krav som ställs på systemet räcker inte den säkerhetsfacilitet som Excell erbjuder.

### **Relationsdatabas:**

Access erbjuder - ur säkerhetssynpunkt - en tillräckligt flexibel lösning för att möta de krav man ställer på systemet. Dels kan alla användare arrangeras i grupper som tillfullo kan motsvara de i kapitel 3.4.1.2 uppräknade aktörerna. Vidare kan tabeller knytas till antingen grupper eller enskilda användare. Kapitel 17 i [10] redovisar utförligt vad som kan åstadkommas i Access.

## Resurshantering av personal – relationsdatabas eller kalkylblad?

---

Som alternativ eller komplement skulle man 'själv' kunna realisera regler med hjälp av Visual Basic-'script' som exempelvis med hjälp av lämpliga kolumner – kategorikod, användarkod, etc. - i passande tabeller.

### 3.1.7 Databehandling

När databehandling sker, görs två huvudoperationer:

- Framtagning.
- Underhåll.

**Framtagning** innebär att läsa data från filer att användaren kan nyttja dessa med vilka avsikter som helst.

**Underhåll** av data går ut på att man ändrar data med följande typer av operationer:

- Lägga till nya post.
- Borttagning av existerande post.
- Ändring av existerande värden.

#### **Kalkylsystem:**

Excel använder funktioner som kan också vara fördefinierade formler. Dessa formler är inbyggda i programmet.

Exempel på såna är: SUM – som adderar celler, kolumner etc. - och MID – som söker textsträngar.

Excel erbjuder felmeddelanden om något är fel i en funktion som man har skapat, men felmeddelanden är inte tillräckligt specifika, för att snabbt kunna finna orsaken till felet.

Det kan också vara besvärlig att om man söker efter en viss post då man kan tänkas behöva skrolla genom hela kalkylbladet för att kunna finna vad man söker.

#### **Relationsdatabas:**

Dessa transaktioner kan göras med hjälp av SQL eller med formler.

- Tillägg av post eller tabell görs med t ex INSERT och CREATE TABLE.
- Borttagning av post görs med exempelvis DELETE.
- Ändring av värde görs med UPDATE

Vidare finns det olika operatorer för att kunna välja ut vissa poster, tabeller etc. med `SELECT`. Tabeller kan kopplas till varandra med `JOIN`.

### 3.1.8 Kommunikationsfrågor

Det går utmärkt att använda både Excel och Access i ett nätverk – intranät eller internet.

Docka lämpar sig Access bättre då användaren låter sig lättare identifieras, se kapitel 3.1.6 ovan.

Nämnas bör att både Excell och Access kan kopplas till andra program som exempelvis Microsoft Word.

## 3.2 Verksamhetsanalys

Det första stadiet i systemutvecklingsprocessen är att göra en verksamhetsanalys som hjälper att förstå organisationen som systemet ska ingå i. Man analyserar informationsflödet i verksamheten och tar reda på vilka informationskanaler som är relevanta för systemet. Man identifierar också ineffektiva delar av informationshanteringen – meningen med datoriseringen är att öka effektiviteten i informationsflödet, se [1]. Eftersom informationssystemet skall hjälpa verksamheten till bättre resultat är det en viktig del i systemutvecklingsprocessen att undersöka och diskutera med användarna på vilket sätt detta kan uppnås.

Viktiga frågor att ställa i en verksamhet är t ex:

- Vad ska systemet klara av?
- Vilka processer ska systemet stödja?
- Vilka kommer att använda systemet?
- Med vilka andra formella och informella system skall systemet samspela?
- Vilka informationskällor till systemet finns?
- Vilka ”frågor” ska systemet svara på?(Se [11])

Utvärderingen av system görs under hela utvecklingsprocessen, från skapandet av en kravspecifikation till och med implementation och testning av systemet. Det behöver inte följa fastställda mallar. Den kan vara informell. Man kan exempelvis fråga användaren om vissa detaljer under implementationsfasen och få svar direkt på enklare designfrågor. Man kan också utföra test av olika designidéer.



Denna typ av utvärdering hjälper till att skapa ett bra och användbart system. Summerande utvärdering sker efter det att systemet är implementerat och har tagits i bruk. Det handlar om att testa det färdiga systemet och avgöra om det uppfyller kravspecifikationen och om användarna är nöjda.

Här nedan beskrivs resultatet av intervjuerna, kravspecifikationen och ett användningsfall som gjorde

### 3.3 Intervjuer

De flesta cheferna använder numera kalkylblad med hjälp av Excel. Jag har intervjuat chefen för projektledningskontoret - som gjorde det nuvarande systemet.

Han har försökt att skapa en relationsdatabas med hjälp av Excel och sina begränsade programmeringskunskaper. Han tyckte att problemet var att länkningarna mellan de olika arbetsbladen var alldeles för komplicerade, när systemet skulle flyttas till en annan hårddisk bröts någon länk av misstag, och uppdaterade systemet felaktigt, eller inte alls.

Detta resulterade i mycket administrativt arbete som förstås är mycket tidsödande. Han har också berättat att det räckte att skriva in fel i en ruta varvid det blev fel i alla beräkningarna.

Andra chefer tyckte att Excel-uppställningarna och dess beräkningarna var alldeles för komplicerade och svårtolkade.

Annars var de flesta överens om att de rapportlistor som skrevs ut i Excel innehöll den nödvändiga informationen, se kapitel 0 nedan.

Det enda som de påpekade var att hade systemet många onödiga dubbellagringar

De flesta förväntade sig av systemet att det skulle vara lättöverskådligt. Även användare som inte känner till databaser eller kalkylprogram. Det ska ta kort tid att kunna gå in och uppdatera och hämta information, det ska vara också lätt att skriva ut olika listor för olika ändamål.

Vidare tyckte man att det vore bra med vissa säkerhetsåtgärder. Om någon av projektledarna äskar tid för en resursperson som överskrider dennes tillgängliga tid, kan en kontroll införas som varnar för överskridandet.

Just nu kontrolleras sådant av linjeförföraren.

Projektkontorets ledning önskade ett system som möjliggör planeringen på lång sikt, ungefär två år i förväg. Vidare vill de ha en rapporteringssystem som möjliggör att rapporter på olika nivåer tas ut oberoende av de olika "undernivåerna"(projekt, sektion, avdelningsnivå).

Slutligen skall det vara lätt att uppdatera förändringar i sammansättningen av resurspersonalen. Det nuvarande, kalkylbladssystemet visar inte utfallet av äskningen.

### 3.4 Krav på Systemet

Kraven på systemet kan framföras med hjälp av så kallade användningsfall. Ett användningsfall karaktäriseras av en beskrivning på hur systemet skall uppföra under givna förutsättningar. Ett användningsfall illustrerar på så sätt hur systemet upplevs ”utifrån” för ett givet fall.

#### 3.4.1.1 Allmänt

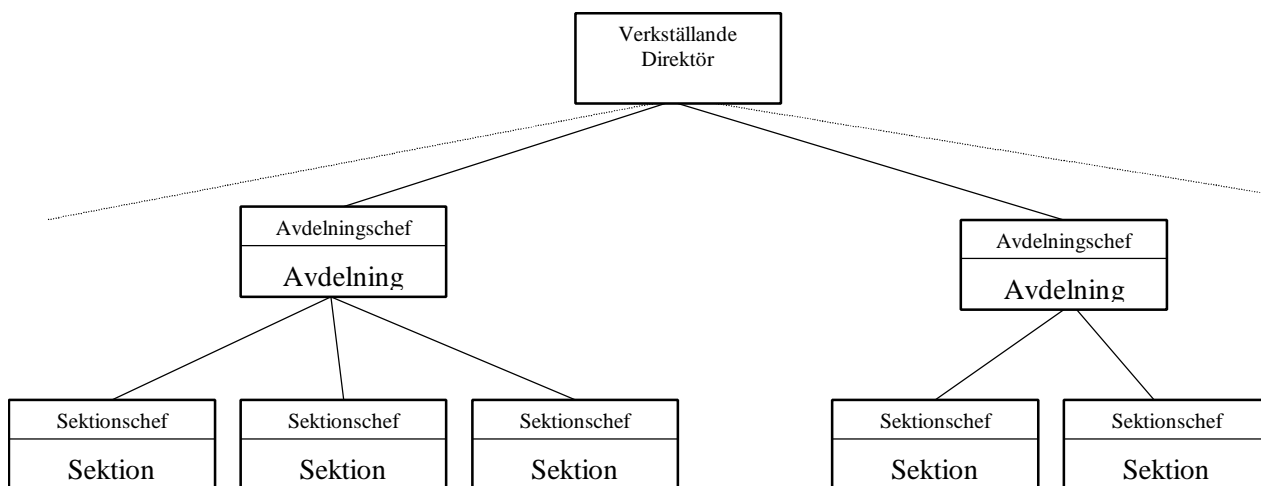
Systemet skall kunna hantera ett godtyckligt antal projekt. Ett projekt förses med personal – eller resurser – som tas från linjeorganisationens sektioner.

Linjeorganisationen är benämningen på den hierarkiska struktur som de flesta verksamheter är organiserade i.

På ERV utgörs strukturen av en verkställande direktör på högsta nivå. Denne har under sig ett antal avdelningar som vardera leds av en avdelningschef.

Varje avdelning är indelad i ett antal sektioner, ledda av en sektionschef vardera.

Sektionerna utgör basen för denna pyramid - se Figur 2 nedan – och här påträffas följaktligen resurspersonerna som bemannar projekten.



**Figur 2 Linjeorganisations struktur.**

Sektionerna som innehåller resurspersonerna är av två huvudtyper:

Utveckling och Verifiering.

## Resurshantering av personal – relationsdatabas eller kalkylblad?

---

Det är sålunda från dessa sektioner projekten erhåller sina erforderliga resurser.

Resurshanteringssystemet skall tillhandahålla sådana tjänster som förenklar administrationen av projekten. Grovt sett kan dessa tjänster kategoriseras i:

- Äskande av resurser.
- Godkännande av resursäskande.
- Administration.
- Resursplanering.

Dessa kategorier kommer att redovisas i form av användningsfall senare i kapitel 3.4.1.3 nedan.

### 3.4.1.2 Aktörer i Systemet

Innan användningsfallen specificeras måste det bestämmas vilka aktörer som agerar i användningsfallen. Aktören spelar en eller flera roller i användningsfallen. Dock bara en utmärkande roll i ett visst användningsfall.

För resurshanteringssystemet har följande aktörer bestämts:

- Projektledaren.
- Linjechefen.
- Systemadministratören.
- Avdelningschefen.

Nedan redogörs i detalj för vad dessa aktörer utgör.

#### 3.4.1.2.1 Projektledaren

Projektledarens uppgift är att sköta projektet, d.v.s. att se till att det blir genomfört enligt en uppgjord projektplan. Projektet genomförs enligt de krav som läggs mot projektet. Typen av krav är bl.a. löptid, kostnad och funktionalitet.

Efter att ett projekt har startats kan projektledaren ge sina första uppgifter, planera resursbehovet för att få underlag till hur projektet skall bemannas.

Projektledaren kommer att behöva resurser för själva systemerings- och utvecklingsfasen – d.v.s. analys, design och kodning – samt för verifieringsfasen – alltså test. Slutligen skall den utvecklade produkten ”rullas ut” och implementeras hos kund. För samtliga dessa steg behöver projektledaren alltså resurser.

---

## Resurshantering av personal – relationsdatabas eller kalkylblad?

---

När dessa resurspersoner utsetts äskar projektledaren om att få utnyttja dessa individer ett visst antal timmar. Dessa timmar bestäms månadsvis och projektledaren lämnar in äskningsunderlag till berörda linjechefer – i själva verket sektionschefer – den 15:e varje månad.

### 3.4.1.2.2 Linjechefen/sektionschefen

Linjechefen ansvarig för en sektion bestående av resurspersoner. Dennes roll är att som personalansvarig ta hand om och utveckla sina sektionsmedlemmar enligt sektionens mål.

Förutom den sociala aspekten av detta omhändertagande, planerar linjechefen sina sektionsmedlemmars arbete – d.v.s. vad och i vilka projekt de skall arbeta.

Efter att projektledaren har lämnat sina äskanden om resurser till linjechefen, se kapitel 3.4.1.2.1 ovan, skall linjechefen helt eller delvis godkänna äskandena. Detta sker den 20:e varje månad<sup>3</sup>.

### 3.4.1.2.3 Systemadministratören

Systemadministratören har rollen att sköta de övriga sysslorna som är nödvändiga för resurshanteringssystemet.

Dessa göromål innebär grovt sett att:

- Lägga till/ta bort personal.
- Lägga till /ta bort projekt.

För övrigt spelar systemadministratören ingen roll i de huvudsakliga uppgifterna – d.v.s. äskande och godkännande av äskande – i systemet.

### 3.4.1.2.4 Avdelningschefen

Avdelningschefen har en passiv roll i resursplaneringssystemet. Han kan exempelvis ha nytta av systemet för att få bild av beläggningen på hennes/hans sektioner.

Avdelningschefen tittar på framtida projektbehov, och kan därmed se i förväg hur många personer som behövs i ett projekt. Avdelningschefen undersöker om resurserna utnyttjas optimalt, räcker de till, behövs det mer folk eller annan typ av kompetens?

---

<sup>3</sup> I praktiken har projektledare och linjechef redan gjort upp och stämt av med varandra vid detta tillfället, så inga större överraskningar torde inträffa.

---

### 3.4.1.3 Användningsfall

För detta specifika problem – sålunda resurshanteringssystemet – har följande användningsfall identifierats:

- Skapande av projekt.
- Initiering av projekt.
- Äskande av resurser.
- Godkännande av resursäskande.
- Registrering av personal.
- Projektmässig rapportgenerering.
- Avdelningsmässig rapportering.

#### 3.4.1.3.1 Skapande av projekt

Bortsett från det formella, d.v.s. att det inom företagets styrgrupp givits lov att sätta igång ett projekt, är det systemadministratören – eller den som har administratörsrättigheter - som praktiskt skapar ett nytt projekt i systemet.

Uppgifter som skall registreras vid skapandet är:

- Namn på projekt.
- Utsedd projektledare,

#### 3.4.1.3.2 Initiering av projekt

Huvudrollen spelas av projektledaren. Linjecheferna är indirekt berörda, men spelar ur systemsynpunkt ingen roll i detta användningsfall.

Initieringen innebär att projektledaren påbörjar projektet, d.v.s. bestämmer vilka och hur mycket resurser som behövs för att genomföra projektet.

Praktiskt sett har projektledaren sett ut ett antal individer i sektionerna., och projektledaren har genom dialog med berörda linjechefer förankrat sina behov hos denne.

Ur systemsynpunkt väljer han ut dessa personer – som är registrerade i systemet, se kapitel 3.4.1.3.5 nedan – från sektionerna och skapar på så sett en lista över projektmedlemmar.

Urvalet av resurser sker på personnivå. Det handlar följaktligen om en specifik person.

---

## Resurshantering av personal – relationsdatabas eller kalkylblad?

---

### 3.4.1.3.3 Äskande av resurser

Projektledaren äskar 1 gång / månad (eller - vid behov - flera gånger per månaden) med hjälp av systemet . Några dagar senare erhålls svar på äskandet.

### 3.4.1.3.4 Godkännande av resursäskande

Detta sker några dagar efter den 15. Sektionschefen bekräftar , antingen godkänns det önskade antalet timmar och personer, eller inte. Det är alltså sektionschefen som bestämmer. Idag har man dålig uppsikt över detta. Det är mest projektledaren som har informationen.

### 3.4.1.3.5 Registrering av personal

Administratören registrerar alla anställda på företaget<sup>4</sup>. Från början registreras alla för tillfället anställda, och därefter uppdateras systemet allteftersom personal anställs respektive slutar.

### 3.4.1.3.6 Projektmässig rapportgenerering

Man – d.v.s. projektledaren- skall kunna erhålla en rapport från systemet hur många timmar han har fått. Det kan han använda i sin projektrapportering och planering.

### 3.4.1.3.7 Linjemässig rapportgenerering

Sektionschefen använder denna funktion. Den ger honom riktlinjer om han behöver anställa folk eller inte. Han vill se behovet av kompetensen i framtiden. Detta för en långsiktig kompetensplanering för sektionen.

### 3.4.1.3.8 Avdelningsmässig rapportering

Här är det två vyer som är av intresse: Hur mycket projekten slukar under tiden och hur mycket folk som är resursallokerade i projekten från de olika sektionerna under tiden. Dessa vyer gör det lättare för styrgruppen att fatta korrekta beslut om t ex prioritering mellan projekt.

## 3.5 Design av databas

I designarbetet följde i stort sätt följde jag KRB-metoden.

---

<sup>4</sup> För enkelhetens skull matas alla anställda i företaget in, även om inte alla berörs av systemet.

---

KRB-metoden är en modelleringsmetod och syftar till att utveckla ett klassdiagram eller en objektmodell samt dokumentation för denna. Detta ligger sedan till grund för implementeringen av systemet man modellerar. KRB består i korthet av följande sju steg, se [2]:

1. Identifiera kandidatklasser
2. Bestäm definitiva klasser
3. Etablera relationer mellan klasser
4. Utveckla relationer med kardinalitet M:M
5. Bestäm attribut för varje klass
6. Normalisera
7. Bestäm operationer (beteende) för relevanta klasser

#### 3.5.1 Steg 1 – Identifiera kandidatklasser

I detta steg gäller det att hitta olika klasser av objekt som kan vara intressanta för systemet – alltså en lista av substantiv som i nästa steg utvärderas för att avgöra om de verkligen hör hemma i systemet.

De vanligaste klasstyper är entitetsklasser. Dessa relaterar till verkliga entiteter, eller saker, i användarens värld, t ex Bil och Bilägare.

#### 3.5.2 Steg 2 – Bestäm definitiva klasser

I steg 2 testar man alla kandidatklasser för att komma fram till om de verkligen är relevanta i systemet. Om de visar sig vara det, testar man om de behöver vara egna klasser eller om de kan vara attribut till någon annan klass. Varje klass testas mot tre olika kriterier:

- Identifierare – instanserna i en klass måste i den verkliga världen vara distinkta och åtskiljbara, de måste ha en identitet. För varje klass måste ett nyckelvärde, en identifierare, hittas som skiljer de olika medlemmarna eller instanserna i klassen ifrån varandra.
- Definition – för varje klass skall en beskrivning ges av vad klassen egentligen är.

- Exempelattribut och beteenden – en klass behöver oftast inte bara en identifierare utan även andra attribut som innehåller viktig information. Det gäller att hitta attribut hos objektet som är relevanta i systemet. Även beteenden kan vara viktiga. Man kan t ex fråga sig ”vad kan en person göra som är relevant för systemet?”. Svaret kan t ex vara ”byta arbetsuppgift”.

### 3.5.3 Steg 3 – Etablera relationer mellan klasser

I detta steg tar man reda på hur de olika klasserna interagerar med varandra, d v s vilka relationer som finns mellan de. Flera faktorer behöver beaktas för att man ska lyckas med detta steg, bl a följande:

- Upptäck verb
- Fånga alla möjliga relationer
- Etablera kardinaliteten – bestäm relationens kardinalitet till 1:1, 1:M, M:M.

### 3.5.4 Steg 4 – utveckla relationer med kardinalitet M:M

M:M förhållanden vållar problem när de skall implementeras i en databas. M:M förhållanden kan utvecklas till att bli 1:M förhållanden genom att man inför en ny klass som så att säga ligger ”mitt emellan” de ursprungliga klasserna. På detta har jag flera exempel i min egen objektmodell. Se nedan.

### 3.5.5 Steg 5 – Bestäm attribut för varje klass

I detta steg går man igenom varje klass och listar alla de attribut klasserna ska ha, d v s vilken information man ska lagra för varje klass. De ska också definieras och förklaras.

### 3.5.6 Steg 6 – Normalisera

Normalisera innebär att se till att varje attribut hör till rätt klass. Man kan välja att normalisera till olika nivåer, vanligast är till 3:e normalformen.(3NF). Jag förklarar normaliseringen utförligare genom arbetets gång i framtagningen av objektmodellen nedan.

### 3.5.7 Steg 7 – Bestäm operationer(beteende) för relevanta klasser

Detta steg handlar om att upptäcka och specificera operationer och beteende och syftar till att kontrollera att allt är korrekt i objektmodellen, se [2].

Med beaktandet av dessa stegen kom jag fram till det följande objekt och datamodellen:

#### **Definition av objekttyper**

---



## Resurshantering av personal – relationsdatabas eller kalkylblad?

Systemet har följande objekttyper:

- **Resurs.**
- **Projekt**
- **Sektion**
- **Månadsäskande**

Nedan redovisas dessa objekt. Kolumnerna som innehåller '(1)' eller '(\*)' indikerar kardinaliteten, d.v.s. antalet förekomster: En respektive många.

Vidare, om en term har symbolen "☛" till höger, innebär detta att termen är en nyckelterm.

### 3.5.7.1 Resurs

Respresenterar först och främst en resurs, men detta objekt omfattar inte bara projekt- och sektionsmedlemmen, utan även projektledare och sektionschef.

#### Resurs

▪ <b>anställningsNummer☛</b>	(1)	Identifierar den anstälde.
▪ <b>namn</b>	(1)	Namn på den anstälde.
▪ <b>tillgängligTid</b>	(1)	Antalet timmar den anstälde kan arbeta per månad.

### 3.5.7.2 Projekt

Projektet omfattar följande termer:

#### Projekt

▪ <b>projektNamn☛</b>	(1)	Identifierar projektet.
▪ <b>projektLedare</b>	(1)	Projektledaren , d.v.s. den som ansvarar för projektet.
▪ <b>resurs</b>	(*)	Den/De som utgör projektet. <b>resurs</b> refererar till en/flera instans(er) av ett/flera <b>Resurs</b> objekt.

Det bör speciellt nämnas att termen **resurs** kan förekomma ett godtyckligt antal gånger.

### 3.5.7.3 Sektion

Sektionen objektet har till uppgift att samla de resurser som organisatoriskt tillhör samma linje.

#### Sektion

▪ <b>sektionsNamn</b> ⊕	(1)	Identifierar sektionen.
▪ <b>sektionsChef</b>	(1)	Linjeförman som leder sektionen.
▪ <b>medlem</b>	(*)	Den/De som utgör sektionen. <b>medlem</b> refererar till en/flera instans(er) av ett/flera <b>Resurs</b> objekt.

På samma sätt som. för projekt objektet förekommer kan en sektion ha ett godtyckligt antal medlemmar

### 3.5.7.4 Månadsäskande

Detta objekt representerar äskande av tid månadsvis för en projekt medlem.

#### Månadsäskande

▪ <b>månad</b>	(1)	Månaden som äskandet avser.
▪ <b>projekt</b>	(1)	Identifierar vilket projekt äskandet gäller.
▪ <b>projektMedlem</b>	(1)	Resursen som äskas <b>projektMedlem</b> refererar till en instans av ett <b>Resurs</b> objekt.
▪ <b>äskadTid</b>	(1)	Antal begärda timmar.
▪ <b>godkändTid</b>	(1)	Antal timmar godkända av projektmedlemmens sektionschef,

### 3.5.8 Normalisering

Denna process innebär att redundansen reduceras för att eliminera dubbellagring och är beskriven i kapitel 4, "Normalisering", i [8] (sidorna 47 – 59).

#### 3.5.8.1 Första normalformen

Av de objekt som definierats i kapitel 0 ovan uppfyller endast **Resurs** objektet 1NF. De övriga objekttyperna måste slås isär enligt följande:

##### 3.5.8.1.1 Projekt

P.g.a. att termen **resurs** har ett-till-många-förhållande till primärnyckeln **projektNamn**, slås **Projekt**-objektet isär enligt följande:

Resurshantering av personal – relationsdatabas eller kalkylblad?

---

**Projekt** -objektet reduceras på sin term **resurs** som istället får ingå i ett nytt relationsobjekt **ProjektResurs**.

Följande objekt typer erhålls:

#### Projekt

▪ <b>projektNamn</b> ✳	(1)	Identifierar projektet.
▪ <b>projektLedare</b>	(1)	Projektledaren , d.v.s. den som ansvarar för projektet.

#### ProjektResurs

▪ <b>id</b> ✳	(1)	Identifikation av en instans av en <b>ProjektResurs</b> .
▪ <b>resurs</b>	(1)	Medlem till projektet <b>projektNamn</b> . <b>resurs</b> refererar till en instans av ett <b>Resurs</b> objekt.
▪ <b>projektNamn</b>	(1)	Projekt som <b>medlem</b> tillhör.

### 3.5.8.1.2 Sektion

P.g.a. att termen **medlem** har ett-till-många-förhållande till primärnyckeln **sektionsNamn**, slås **Sektion**-objektet isär enligt följande:

**Sektion** -objektet reduceras på sin term **medlem** som istället får ingå i ett nytt relationsobjekt **SektionsMedlem**.

Följande objekt typer erhålls:

#### Sektion

▪ <b>sektionsNamn</b> ✳	(1)	Identifierar sektionen.
▪ <b>sektionsChef</b>	(1)	Linjechef som leder sektionen.

#### SektionsMedlem

▪ <b>medlem</b>	(1)	Medlem till sektionen <b>sektionsNamn</b> . <b>medlem</b> refererar till en instans av ett <b>Resurs</b> objekt.
▪ <b>sektionsNamn</b>	(1)	Sektionen som <b>medlem</b> tillhör.

### 3.5.8.1.3 Månadsäskande

Eftersom att termerna **projekt** och **projektmedlem** tillsammans blivit ett relations objekt, se kapitel 3.5.8.1.2 ovan, ersätts dessa termer i objektet **Månadsäskande** med en term som refererar till en post av typen **Projektresurs**.

#### Månadsäskande

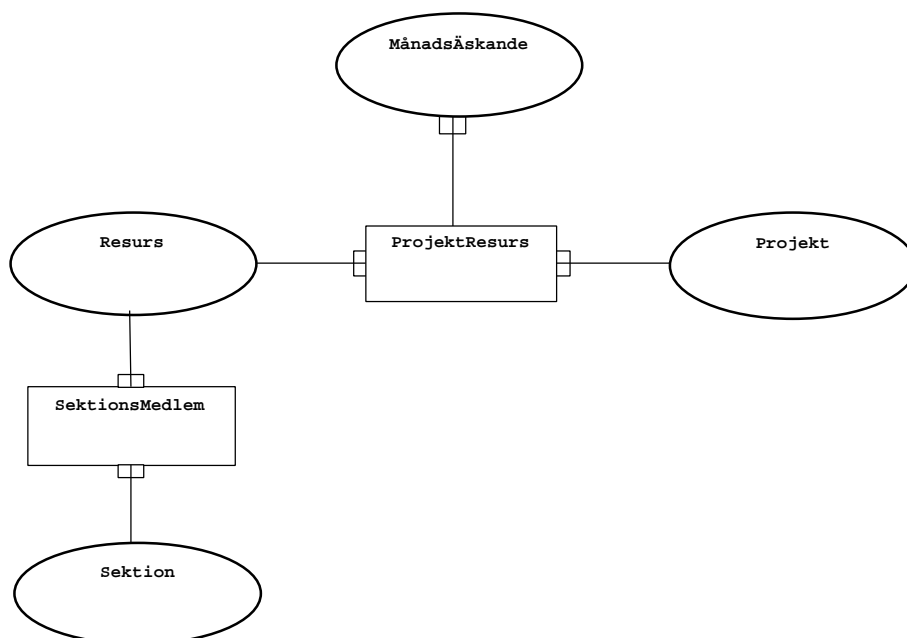
▪ <b>månad</b>	(1)	Månaden som äskandet avser.
▪ <b>id</b>	(1)	Identifierar både resurs och dess projekt tillhörighet som äskandet gäller. <b>id</b> refererar till en instans av ett <b>Projektresurs</b> objekt.
▪ <b>äskadtid</b>	(1)	Antal begärda timmar.
▪ <b>godkändTid</b>	(1)	Antal timmar godkända av projektmedlemmens sektionschef,

### 3.5.8.2 Andra normalformen

Eftersom alla objekttypers termer nu är funktionellt oberoende av primärnyckeln är 2NF uppfyllt.

### 3.5.9 Modellering

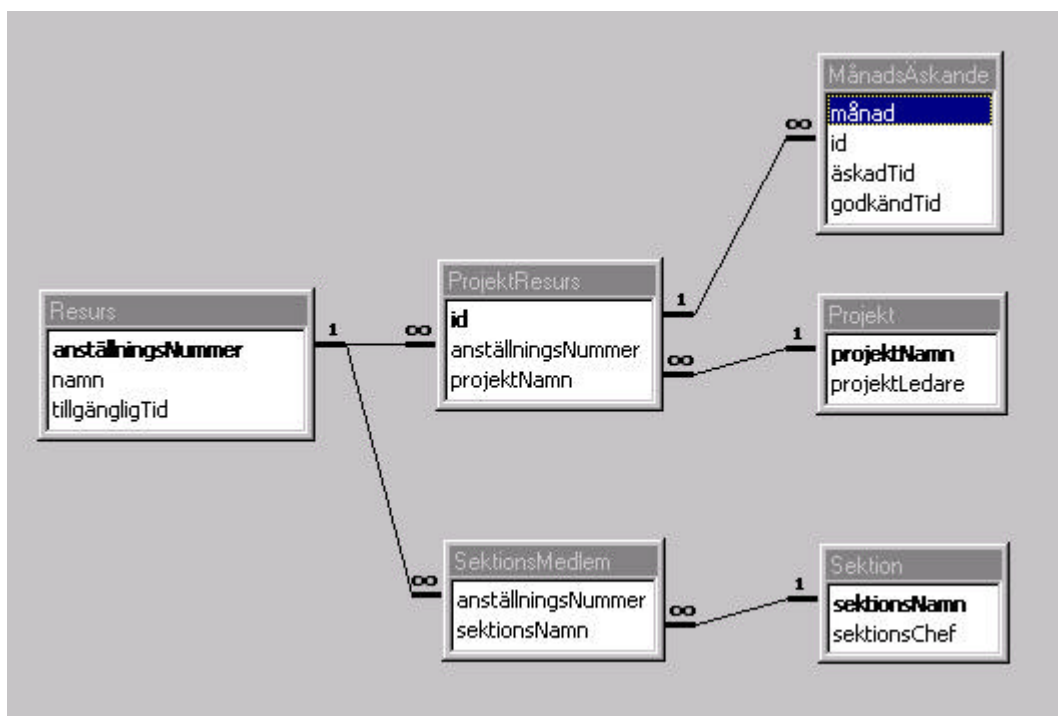
De objekttyper som normaliseringen resulterade i följande objektmodell som illustreras av figur 3



**Figur 3 Objektmodellen.**

## Resurshantering av personal – relationsdatabas eller kalkylblad?

Motsvarande implementering i Microsoft Access ger en tabellstruktur som illustreras av Figur 4 nedan



**Figur 4** Microsoft Access databasmodell av resurshanteringsystem.

#### 4 **Diskussion**

Man har försökt att lösa problemet med resurshanteringen hos Ericsson med kalkylblad – vilket inte gick så bra – men en informationsmodell uppbyggd i en databas - till vilken man skapar olika vyer - torde lösa uppgiften.

Fördelarna med en relationsdatabaslösning som Access, är följande:

- Formulären ger **större flexibilitet** i sorteringen och selektionen av data, än i det som en kalkylbladsapplikation kan erbjuda.
- Access erbjuder **fler möjligheter** för att generera formaterade rapporter av datainnehållet på ett mer eller mindre godtyckligt format.
- Access kan **lätt kombineras med VBA** (Visual Basic for Applications), som möjliggör att skriva program i en fulländat programmeringsspråk och begränsar en inte till av användandet av förutbestämda kalkylbladsapplikationer.
- Rätt designade Access relationsdatabaser **minimerar dubbellagringen** av information, snabbar upp åtkomsttiden till data, och minskar kravet för stort diskutrymme.

Både kalkylbladssystem och relationsdatabassystem är en form av databasapplikationer, men används i olika syfte. De är skraddarsydda för olika behov.

I kalkylbladssystem dubbellagras lätt information. När man t.ex.. byter hårddisk bryts länkarna lätt – som redan påpekats i intervjuerna, se kapitel 3 - mellan de olika bladen. Det kan ta mycket lång tid för att återskapa länkar, och det kan tänkas att man misslyckas att återskapa data i sitt system.

Det är svårt att felsöka i Kalkylsystem, och det tar lång tid när många kalkylsystemark är knutna till varandra utan att ligga i databas. Ju färre tabeller desto bättre.

Däremot, om man behöver skapa en enkel tvådimensionell databas - exempelvis ett tidsschema - är kalkylbladssystem oslagbart p.g.a. dess lättöverskådlighet.

Nackdelen med relationsdatabassystem att det krävs en extra person i form av en systemadministratör, alltså krävs det underhåll.

Med hjälp av relationsdatabassystem kan man lätt få en uppfattning var de stora behoven är. Det blir alltid något lidande om en annan sak prioriteras.: kvalitén, funktionaliteten, eller ledtiden eller både och.

## 5 **Slutsats**

När det gäller att välja verktyg för att lösa uppgiften att hantera resurser har man således två möjligheter att välja mellan:

Relationsdatabas eller kalkylsystem.

Erfarenheterna från ERVs nuvarande system – den kalkylsystembaserade applikationen – har visat att den inte är nöjaktig p.g.a. problemdomänen har för många dimensioner.

Ett system baserat på en relationsdatabas möjliggör en lösning som dessutom går att utvidga, d.v.s. fler aspekter och dimensioner skulle kunna hanteras.

Förutom va som redovisats i kapitel 3.1 ovan, följer de huvudsakliga för- och nackdelarna med de olika hjälpmedlen.

### 5.1 **Varför relationsdatabas?**

Fördelarna med en relationsdatabas i ett företag är många. Det gör det möjligt att lagra många typer av information och att snabbt hitta vad man söker.

För det specifika problemet resurshanteringssystem passar detta då flera olika typer av information skall samverka.

Nackdelen med relationsdatabasen är att den initiala insatsen är stor och kräver utförlig kunskap i frågan.

### 5.2 **Varför kalkylbladssystem?**

Kalkylbladssystem har fördelen att det är lätt att använda.

Nybjörjare kommer snabbt igång med enkla verktyg som Microsoft Excel. Dessutom är det lätt att koppla kalkylblad till webben - Internet eller företagets intranät.

Nackdelen med kalkylbladssystemet är att antalet olika typer av information som samtidigt hanteras är begränsat, d.v.s. antalet dimensioner i problemdomänen är inskränkt.

**6 Referenser**

- [1] *Allwood CM:*  
”Människa – Datorinteraktion – Ett psykologiskt perspektiv”  
(Lund, Studentlitteratur, 1991)
- [2] *Brown D:*  
”Object – Oriented Analysis. Objects in plain english”,  
First Edition  
(USA, John Wiley & Sons, 1997)
- [3] *Connolly Thomas, Begg Carolyn, Strachan Anne :*  
”Database Systems”  
(ISBN 0-201-42227-8, Addison-Wesley, 1996)
- [4] *Date, C.J.:*  
”Introduction to Database Management Systems”, Vol. 1,  
4<sup>th</sup> ed.  
(Addison-Wesley, Reading, MA, 1986)
- [5] *Elmasri Ramez, Navathe Shamkant B.:*  
”Fundamentals of Database Design”, Second Edition  
(ISBN 0-8053-1753-8, The Benjamin/Cummings  
Publishing Company, 1994)
- [6] *Hallberg Bruce Kinkoph, Sherry, Ray Bill, et al:*  
”Using Excel 97”, Bestseller Edition  
(ISBN 0-7897-1399-3, Que, 1997)
- [7] *Jennings Roger:*  
”Using Access 97”, Second Edition  
(ISBN 0-7897-1452-3, Que, 1997)
- [8] *Jonegård Magnus:*  
”SUDD Metoden, Strukturerad analys och databasdesign”  
(ISBN 91-38-61540-1, Liber, 1986)
- [9] *Microsoft Sverige:*  
“Microsoft Access 97”, WWW-dokument  
URL  
<http://www.microsoft.com/Sverige/Products/ACCESS/access97.htm>
- [10] *Prague Carry N., Amo William C., Foxall James D.:*  
”Hemligheterna i Access 97”  
(ISBN 91-88311-23-6, IDG Sweden, 1997)



- [11] *Sundgren B:*  
”Datororienterad systemutveckling, Grundläggande begrepp, Datormodellering, Systemkonstruktion” (Lund, Studentlitteratur, 1992)
- [12] *Sundgren Bo:*  
”Databaser och datamodeller”  
(ISBN 91-44-18061-6, Studentlitteratur, 1981)

## Bilagor

### 7.1 Kalkylblad från det ursprungliga resurshanteringssystemet

Nedan följer de Excel-blad som utgör det ursprungliga systemet.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	1998	Sum	0	0	0	0	0	0	0	0	0	0	0	0
2														
3			Month											
4	Section		Jan	Feb	Mar	Apr	Maj	Jun	Jul	Aug	Sep	Okt	Nov	Dec
5	A	ack												
6		avail.												
7														
8	AL													
9														
10	AD													
11														
12	AN													
13														
14	AO													
15														
16	AV													
17														
18														
19														
20														
21														
22														
23														
24														
25														
26														
27														

Figur 5 Sektions-vy i det ursprungliga resurshanteringssystemet.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
2														
3			Month											
4	Project		Jan	Feb	Mar	Apr	Maj	Jun	Jul	Aug	Sep	Okt	Nov	Dec
5	Version 5	req.	250	0	0	0	0	0	0	0	0	0	0	0
6		ack	0	0	0	0	0	0	0	0	0	0	0	0
7		req.												
8		ack												
9		req.												
10		ack												
11		req.												
12		ack												
13		req.												
14		ack												
15		req.												
16		ack												
17		req.												
18		ack												
19		req.												
20		ack												
21		req.												
22		ack												
23		req.												
24		ack												
25		req.												
26		ack												
27		req.												
28		ack												

Figur 6 Projekt-vy i det ursprungliga resurshanteringssystemet.

Resurshantering av personal – relationsdatabas eller kalkylblad?

---