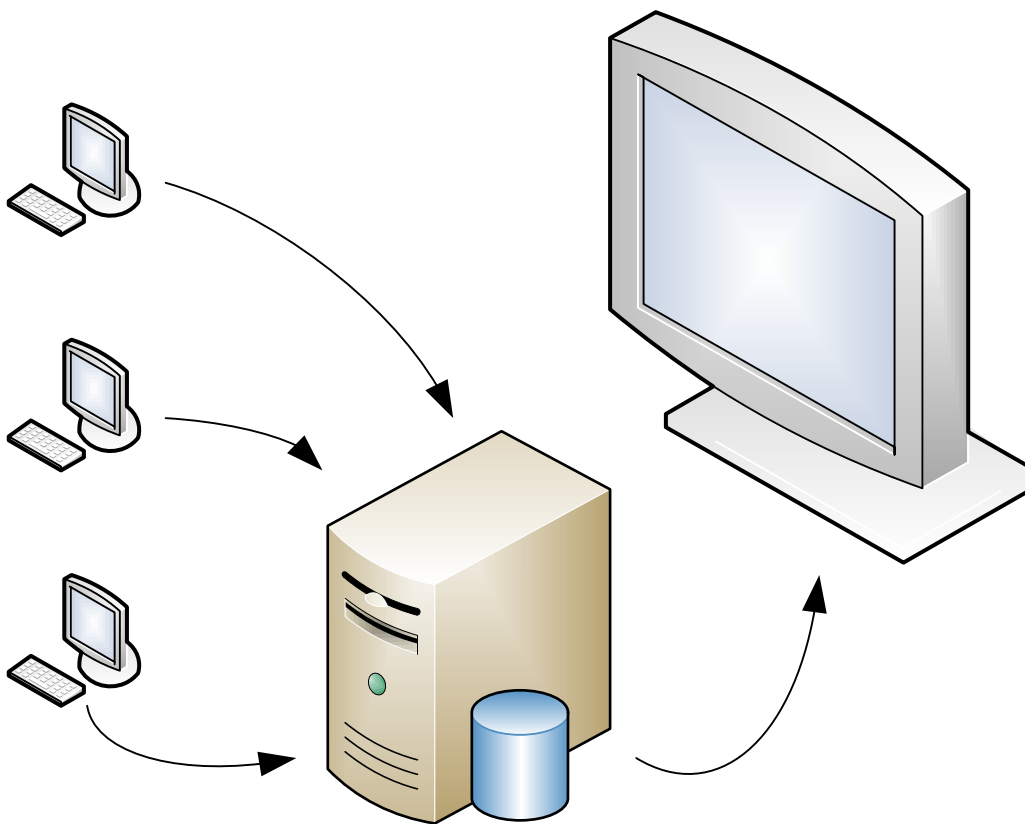


Multidimensionel afrapportering baseret på data warehouse

8. semester cand. merc. økonomistyring og informatik
Aalborg Universitet, maj 2007



Jens Frøkjær
Michael Knudsen
Ivan Vigsø Sand Larsen
Carsten Schou Nielsen
Tom Oddershede

Forord

På 8. semester på cand.merc. uddannelsen i økonomistyring og informatik er der blevet udarbejdet to rapporter delt på to seminarer. Da disse to rapporter emnemæssigt er tæt forbundet, er der valgt at skrive dem sammen til én rapport.

Temaet for Seminar 1 omhandler “Datafangst: Opbygning og design af registreringsopgaven til dedikeret ledelses- og beslutningsinformation”. Dette danner herefter grundlag for temaet for Seminar 2, som omhandler “Multi-dimensionel afrapportering”.

Til at belyse disse emner er der valgt at arbejde med et konceptuelt forslag til anvendelse af SAP Business Ones registreringer som grundlag for en rapportering med activity based costing og dækningsbidragsmodellen, herunder forslag til datafangst. I forbindelse med afrapporteringen vil OLAP værktøjer blive benyttet. Arbejdet skal bringe os igennem de økonomiske og informationsteknologimæssige problemstillinger og udfordringer, som erhvervsøkonomer vil kunne støde på i virksomhederne i forbindelse med datafangst og afrapportering. Der vil under den praktiske del blive arbejdet med en række af de IT værktøjer, som kan benyttes til at løse problemstillingerne ude i virksomheden.

Jens Frøkjær

Michael Knudsen

Ivan Vigsø Sand Larsen

Carsten Schou Nielsen

Tom Oddershede

Indhold

Indledning	9
I Begrebsdannelse	
1 Fokusområde	15
2 Metode	16
2.1 Grundlæggende elementer	16
2.2 Logik	18
2.3 Semantik	19
2.4 Anvendelse af metode	19
3 Dækningsbidrag	21
4 Activity based costing	24
5 SAP Business One	27
6 Udgifter kontra omkostninger	30
7 Opsummering	32
II Data warehouse	
8 Fokusområde	37
9 Grundstruktur	38
10 Extract, transform og load	42
10.1 Extract	43
10.2 Transform	44
10.3 Load	45

10.4 Kettle Spoon	45
11 Opbygning af data staging	47
11.1 Krav til data staging	48
11.2 Design af data staging	49
11.3 Design af data marter	52
12 Datagrundlag	55
12.1 Tid tabellen	56
12.2 Dato tabellen	56
12.3 Konto tabellen	57
12.4 Produkt tabellen	57
12.5 Ressource tabellen	58
12.6 Aktivitet tabellen	58
12.7 Ressource cost-driver	58
12.8 Aktivitets cost-driver	59
12.9 Kontering tabellen	60
13 Extract, transform og load	61
13.1 Tid tabellen	61
13.2 Dato tabellen	62
13.3 Konto tabellen	63
13.4 Produkt tabellen	64
13.5 Ressource og Aktivitet tabellerne	64
13.6 Kontering tabellen	65
13.7 Data marter	66
14 Opsummering	67
 III Afrapportering	
15 Fokusområde	71
16 Rapporteringsmodeller	72
16.1 OLAP	72
16.2 Relational Online Analytical Processing	76
16.3 Multidimensional Online Analytical Processing	76
17 Afrapporteringsværktøjer	78
17.1 Palo	78
17.2 XL Reporter	80
18 Import til Palo	82
18.1 Datodimension	82
18.2 Dækningsbidrag	84
18.3 Activity Based Costing	86

19	Rapporter	88
19.1	Dækningsbidragsrapporter	88
19.1.1	Dækningsbidrag	89
19.1.2	Omkostningskonti	89
19.2	ABC rapporter	89
19.2.1	Omkostninger fordelt på aktiviteter	89
19.2.2	Omkostningsobjekters træk på aktiviteter	90
19.2.3	Omkostninger per omkostningsobjekt	91
20	Opsummering	92
21	Konklusion	93
	Litteratur	96
A	Kettle overblik	97
B	Rapporter	98
B.1	Dækningsbidragsrapport	98
B.2	Oversigt over omkostninger på konti	99
B.3	ABC-fordelte omkostninger	100

Indledning

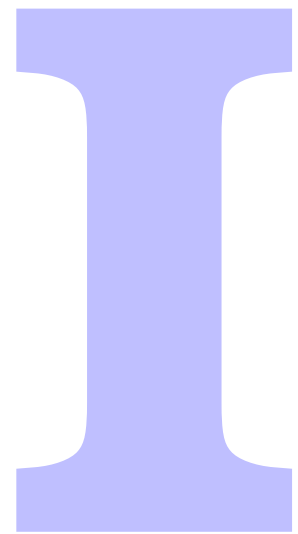
Der har i de seneste år været en stor efterspørgsel efter muligheder for omfattende analyser af økonomistyring og business intelligence. Herved har de fleste virksomheder en naturlig interesse i at samle data fra de forskellige processer i virksomheden, for derefter at kunne omdanne disse data til brugbar information. Virksomhederne skal benytte disse informationer som beslutningsgrundlag for alle beslutningstyper, der skal tages i forbindelse med driften. Udfordringen for langt de fleste virksomheder er at samle disse data, og få dem arkiveret, således de senere let kan tilgås og benyttes. Til at hjælpe med økonomistyringen, har de fleste virksomheder indført et *enterprise resource planning* (ERP) system. ERP systemer benyttes til at håndtere den finansielle viden, der foreligger i virksomheden. Udfordringen er at få ERP systemet og virksomhedens andre it-systemer til at samarbejde om at levere data, der er nødvendige for at træffe de relevante erhvervsøkonomiske beslutninger.

Denne rapport vil tage udgangspunkt i business intelligence og omhandle økonomiske afrapporteringer ud fra dækningsbidragsmodellen og *activity based costing* (ABC). Der vil her blive taget udgangspunkt i SAP Business One som kildesystem til et data warehouse. Rapporten vil beskæftige sig med hvilke overvejelser, der skal gøres i virksomheder i forbindelse med udarbejdelsen af data warehouse og illustrere, hvordan det kan implementeres. Der vil efterfølgende blive taget udgangspunkt i afrapportering ud fra dette data warehouse. Endvidere vil værktøjer og metoder til at udføre denne afrapportering blive bearbejdet. Herved munder projektet ud i en række eksempler på afrapporteringer baseret på data warehouse. Afrapporteringerne vil begrænse sig til de valgte økonomiske modeller, hvilket dog ikke udelukker andre rapporteringsmuligheder.

Idet denne rapport er blevet sammenskrevet af to seminarrapporter, vil elementer fra de forskellige rapporter ikke optræde i samme rækkefølge, som i de oprindelige seminarrapporter. Baggrunden for dette er, at der indgår elementer i seminarrapporterne, som er hensigtsmæssige at beskrive tidligere i processen. Således indgår der en række elementer i Seminarrapport 2, som vil hænge naturligt sammen med udviklingen i Seminarrapport 1. Ved at flytte disse elementer opnås en bedre sammenhæng og et mere naturligt flow i den samlede rapport.

INDHOLD

5 Dette danner grundlag for denne rapport, som er delt op i tre kapitler. Kapitel I vil primært beskæftige sig med en række teorier, som vil blive benyttet gennem resten af rapporten. Derudover vil metoden, som benyttes blive beskrevet. I Kapitel II vil teorien om data warehousing blive benyttet i sammenhæng med, hvorledes data fra SAP Business One kan benyttes. Til sidst vil Kapitel III beskæftige sig med afrapporteringerne.



Begrebsdannelse

Fokusområde 1

Dette kapitel vil primært præsentere teori, der vil danne grundlag for resten af rapporten.

I dette kapitel vil metodeteori blive præsenteret, og der vil blive argumenteret for de
5 metodemæssige valg, og hvilke overvejelser, der ligger bag. I denne rapport er logik og
semantik vigtige begreber, hvorfor disse bliver forklaret i dette kapitel. Endvidere bliver
der argumenteret for, hvilke metodemæssige begrænsninger et teoretisk projekt som
dette sætter, og hvordan det adskiller sig fra egentligt arbejde i en virksomhed.

Dækningsbidragsmodellen bliver præsenteret og der defineres væsentlige begreber inden-
10 for dette område. Denne model er blevet kritiseret for ikke at give et retvisende billede af
hvordan eksempelvis en virksomheds produkter medfører omkostninger i virksomheden.

I kontrast til dækningsbidragsmodellen forsøger ABC at fordele flere af virksomhedens
omkostninger til produkter, for at kunne udføre mere avancerede omkostningsanalyser.

Dernæst følger en kort introduktion af ERP systemet SAP Business One, og hvordan
15 denne benyttes i en virksomhed.

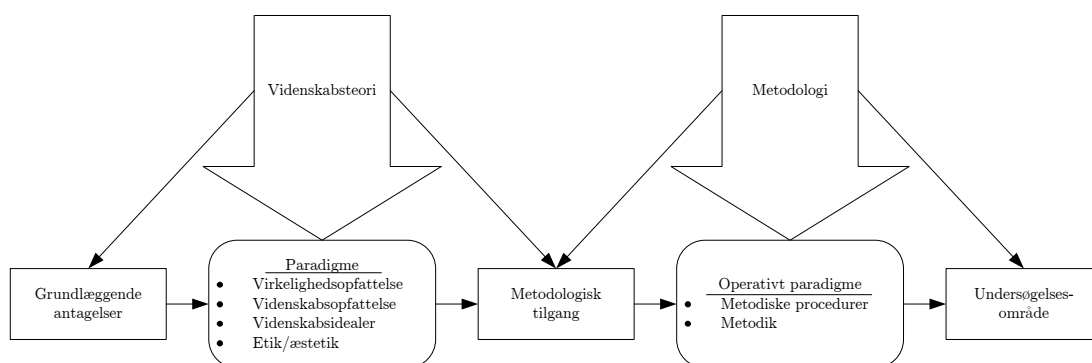
Endvidere vil kapitlet indeholde en diskussion om, hvordan udgifter kan håndteres som
omkostninger.

2 Metode

I dette afsnit vil den relevante videnskabelige metode blive gennemgået. Først vil der være en gennemgang af de elementer, som indgår i den metode, der typisk bliver brugt inden for det erhvervsøkonomiske fagområde. Herefter vil der blive taget et begrundet valg af de metodiske elementer, der vil blive benyttet i denne rapport.

2.1 Grundlæggende elementer

Figur 1 viser de metodiske elementer fra de grundlæggende antagelser frem til undersøgelsesområdet. Bevidsthed om metode skal sørge for, at der vælges de rigtige værktøjer og stilles de rigtige spørgsmål samt danne bevidsthed om, i hvilket lys de fundne svar skal vurderes. Endvidere kan brug af metodiske arbejdsgange gøre det lettere at gennemskue, hvad der bliver behandlet. Herved vil det blive lettere for læseren at sætte sig ind i formidlerens tankegang, hvis der på forhånd kan antages, hvordan formidleren tænker, agerer og arbejder.



Figur 1: Metodeoversigt¹

¹[AB97], side 17

De grundlæggende antagelser er forskellige fra person til person, og er et sammenkog af de holdninger, den givne person har. Den grundlæggende antagelse vil naturligvis være farvet af hvem personen er, omgivelserne, samt tidligere erfaringer. I nogen udstrækning vil de grundlæggende antagelser være forholdsvis ens inden for givne fagområder.

- 5 I det følgende vil begreberne positivisme og hermeneutik blive brugt. Positivisme betyder, at alt skal kunne forklares ud fra den fysiske verden. Et eksempel på dette, kan være billardkugler. Her kan det beregnes med stor nøjagtighed, hvordan de bevæger sig. Hermeneutikken er derimod en forstående tilgang, og vil derfor kunne rumme sociale sammenhænge, som ikke kan fastlægges ud fra beregning alene. Det er gruppens mening
10 at hermeneutikken ikke udelukker logisk og rationel tankegang, hvorimod positivismen ikke kan rumme forstående elementer².

Til at binde den metodiske tilgang og de grundlæggende antagelser sammen, har videnskabsteorien et paradigme, herunder virkelighedsopfattelse, som ifølge [AB97] er delt op i tre grundsyn.

- 15 **Analytisk syn.** Tanken bag dette syn er, at alt kan måles, vejes og kategoriseres. Det objekt, der undersøges, er præcist sammensat af de dele, objektet består af. Analytisk syn benytter en positivistisk tilgang. Dette syn bruges mest i naturvidenskab, hvor det er vigtigt, at der kan laves endegyldige formler. Dette syn forsøger at *forklare* det undersøgte område.
- 20 **Systemsyn.** Her antages at de enkelte dele i systemet påvirker hinanden. Dermed giver summen af de enkelte elementer en anden værdi sammen, end de giver hver for sig. Denne synergieffekt kan både være positiv og negativ. Tilgangen i dette syn kan være både positivistisk og hermeneutisk. Systemsynet forsøger ligeledes at *forklare* undersøgelsesområdet.
- 25 **Aktørsyn.** I dette syn er det aktørerne, der er grundlag for helheden. Ændres bare én faktor, som for eksempel en person, en persons stillingsansvar, vil det have indvirkning på den helhed, der undersøges. Et af nøgleordene i dette syn er forståelse, og den tilgang, der benyttes, er hermeneutisk. Modsat de to tidligere syn, lægger aktørsynet op til at *forstå* undersøgelsesområdet.

- 30 En nærmere gennemgang af disse syn vil ikke fremgå af denne rapport, da det ikke har større betydning for de metodiske valg, der tages senere i dette afsnit. De er taget med for at vise, hvilket omfang den metodiske tilgang har og hvilke elementer, der er fravalgt.

Aktørsynet er det, som stemmer bedst overens med gruppens grundlæggende antagelser, fordi de andre syn ikke kan dække og bringe forståelse for alle aspekter i henhold til det
35 erhvervsøkonomiske fagområde. Der kan argumenteres for, at systemsynet også dækker disse aspekter. Dog må det antages, at de værktøjer, der bruges i det erhvervsøkonomiske fagområde, skal fungere i en virksomhed, hvor både forklarende og forstående elementer

²[AB97], side 47

skal inddrages. Derfor er det gruppens holdning, at da systemsynet hovedsageligt ligger inden for det forklarende aspekt, vil aktørsynet dække bredest. Begrundelsen herfor er, at aktørsynet ikke udelukker det forklarende element, men det er dog nedtonet. Jævnfør den teoretiske natur denne rapport har, vil visse aspekter inden for aktørsynet ikke blive anvendt fuldt ud. Eksempelvis kan der ikke føres en samtale med en bog, og den burde ikke kunne forstås på et andet grundlag end det forfatteren har haft til hensigt. Dog mener gruppen, grundet fagområdet og det, der må forventes at være vores arbejdsopgaver i praksis, at forståelse for sammenhænge i de systemer, der udarbejdes, er vigtig.

10 Virkelighedsopfattelsen er dog kun en del af det paradigme, der anvendes. Både [Kuh95] og [Tör75] beskriver paradigmedannende elementer. Der er en række forskelle i deres tilgang, men det vigtigste er, at [Kuh95] mener udviklingen i paradigmer sker markant og afløser det foregående³, hvor [Tör75] mener at udviklingen i paradigme er gradvis udvikling og ikke udelukker hinanden⁴. I denne sammenhæng er det også vigtigt at
15 holde for øje, at [Kuh95] er positivist, så sammenfattes mener gruppen, at [Tör75] passer fagområdet bedst, mens [Kuh95] finder mest anvendelse hos naturvidenskaben.

Gruppen mener dog ikke at de paradigmedannende elementer, som [AB97] har valgt i deres model, er tilstrækkelige. Derfor tilføjes logik og semantik. Dette begrundes med, at der tit skal tages en logisk stillingtagen til en problemstilling, ud fra givne faktorer, som
20 ikke altid før har været teoretisk bearbejdede. Semantikken er vigtig, da begreber i de forskellige erhvervsøkonomiske sammenhænge ikke altid er tillagt den samme betydning. Logik og semantik vil blive nærmere beskrevet i Afsnit 2.2 og Afsnit 2.3 på modstående side

Det operative paradigme er bindeleddet mellem den metodiske tilgang og studieområdet.
25 Det er udtryk for de metodiske valg, der er foretaget og skal sikre en ensartet tilgang til de problemstillinger, der kommer i undersøgelsesområdet⁵.

2.2 Logik

Logik betyder, at der ud fra givne betingelser drages en logisk konklusion⁶. Logikken beskæftiger sig ikke med, om denne konklusion er rigtig i en større sammenhæng. Hvis
30 det ud fra præmisserne er en korrekt slutning, er det logisk korrekt⁷. Det kan i givet fald diskuteres, om de præmisser, der er opstillet for den givne konklusion, er gyldige. Når der anvendes logik, er det vigtigt, at der tydeliggøres hvilke præmisser, der ligger til grund for den konklusion, der er taget. Ellers kan det ikke diskuteres, om konklusionen er korrekt på et logisk grundlag.

³[Kuh95], side 153

⁴[Tör75], side 2

⁵[AB97], side 16

⁶[Har96], side 11

⁷[Har96], side 8–9

2.3 Semantik

Semantik skal sikre, at der hele tiden er sikkerhed for, hvad det valgte ord betyder, så der ikke opstår signalforvirring. Det vil der kunne opstå, hvis modtageren tillægger et ord en anden betydning, end det formidleren har tillagt det. Det vil i nogle tilfælde kunne forstyrre forståelsen af det, der er hensigten med det givne afsnit, kapitel eller i værste fald hele publikationen⁸. Det er derfor vigtigt, at der præciseres, hvad der menes, hvis der kunne være andre udlæg af de valgte begreber. Dog skal der også tages højde for fagsprog, der kan kræve en vis grundviden fra læserens side. I disse tilfælde må det kræves, at læseren sætter sig ind i dette. Et eksempel herpå er forskellen mellem data og information. Data ligger i uhåndteret form, tit i en database, og information er, når data bruges til noget konkret og sættes i sammenhæng med omverdenen. Denne skildring er universel, og det må påhvile læseren at have denne viden.

Information er resultatet af behandling, manipulation og organisering af data på en måde, så det tilfører viden til modtageren⁹.

Er der derimod tale om et ord, der kan have forskellig betydning alt efter i hvilken teori eller sammenhæng det benyttes, er det vigtigt, at der specificeres hvilken betydning det enkelte ord har i den aktuelle sammenhæng. Et eksempel på dette kan være ordet indtægt. Her er der almen konsensus om, at det betyder, at en virksomhed har solgt en ydelse eller et produkt, og har fordring på en debitor. Men der kan ligge forskellige antagelser bag dette ord, som i specielle tilfælde kan være vigtige at have specificeret. Et spørgsmål kunne ligge i hvornår denne indtægt registreres: Er det ved aftalens indgåelse, afsendelse, kundens modtagelse af varen, eller ved indbetaling på kontoen? Dette vil i praksis være beskrevet i virksomhedens regnskab, så der ikke vil være nogen tvivl for dem, som har brug for informationen. Eksemplet skal blot bruges til at illustrere, at et begreb kan have forskellige måder at blive benyttet på. I sådanne tilfælde er det vigtigt at fastslå hvilken antagelse, der ligger til grund for arbejdet, således misforståelser undgås.

De paradigmedannende begreber er ikke fyldestgørende i forhold til det aktuelle undersøgelsesområde, derfor inddrages logik og semantik.

2.4 Anvendelse af metode

Eftersom det er et teoretisk projekt, vil elementer som blandt andet virkelighedsopfattelse ikke have den store relevans. Gruppen har derfor valgt at lægge hovedvægten af de metodiske overvejelser på logik og semantik. Det betyder, at virkelighedsopfattelsen har indvirkning på den metodiske tilgang. Den har en mere latent natur, end hvis der

⁸[Lyo95], side 75

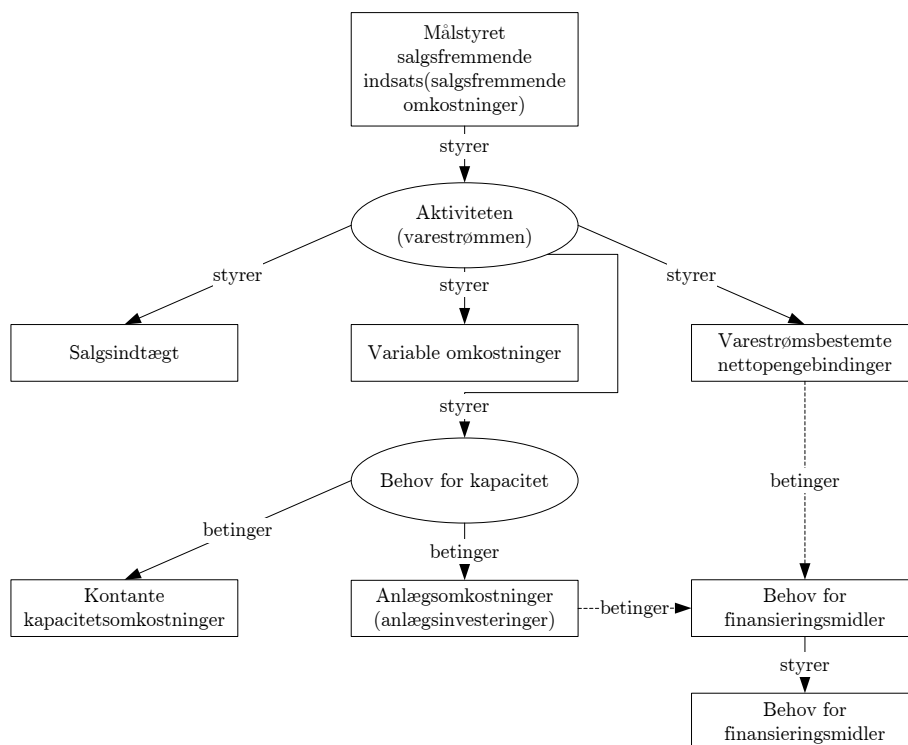
⁹[Wik07]

skulle inddrages empiriske elementer, som for eksempel analyse af en virksomhed.

Semantik og logik vil i projektet bidrage til, at der tages stilling til problematikker og ordvalg, der vil sikre at der gennem projektarbejdet ikke vil være tilfældige løsninger, der har modstridende karakter. Semantikken skal mindske misforståelser og sikre, at
5 der er taget stilling til de enkelte begrebers betydninger og gyldighedsområder. Dette skulle gerne lette forståelsen for det, der er kernen i projektet. Logikken bruges gennemgribende, og specielt i designfasen af data warehouse er logikken inddraget for at sikre en generel brugbar løsning.

Dækningsbidrag

Dette afsnit vil indeholde en kort gennemgang af grundprincipperne i dækningsbidragsmodellen, som den er beskrevet af [Wor94]. Modellen tager udgangspunkt i virksomhedens lønsomheds- og likviditetsmønster som ses i Figur 2.



Figur 2: Model af virksomhedens lønsomheds- og likviditetsmønster¹⁰

Figur 2 illustrerer, hvordan de erhvervsøkonomiske sammenhænge principielt fungerer i virksomheden under dækningsbidragsmodellen. Den målstyrede salgsfremmende

¹⁰[Wor94]

indsats er central for figuren, idet den styrer aktiviteten i virksomheden. Aktiviteten betragtes som varestrømmen, hvilket herved påvirker indtægt, variable omkostninger, pengebindinger samt behovet for kapacitet i virksomheden. Størrelsen af de tidligere nævnte elementer betinger ressourcerne kapacitetsomkostninger, anlægsomkostninger samt behovet for finansieringsmidler. Behovet for finansieringsmidler styrer ligeledes de finansielle omkostninger i virksomheden. Anlægsomkostninger er i dækningsbidragsmodellen forbundet med omkostninger i forbindelse med bygninger og maskiner, og vil typisk fremgå som afskrivninger. Kontante kapacitetsomkostninger er de omkostninger, som forbruges til opretholdelse af kapaciteten, men ikke er variable.

“ Variable omkostninger er omkostninger for forbruget af ressourcer, der er styret af aktivitetens mængde og sammenhæng i enhver situation¹¹. ”

Generelt tager dækningsbidragsmodellen udgangspunkt i virksomhedens omkostninger. I modellen bliver virksomhedens omkostninger delt op i fire grupper. De *variable omkostninger*, som netop er defineret, er kendetegnet ved, at de tilpasser sig aktivitetens størrelse og har fuld variabilitet og reversibilitet. Disse to begreber beskrives senere i dette afsnit. *Kapacitetsomkostningerne* kan bedst beskrives som de faste omkostninger, virksomheden har for at have kapaciteten til en given aktivitet. Disse omkostninger kan i modellen ikke henføres direkte til produkter, hvilket kan medvirke til at give et u hensigtsmæssigt billede af de reelle omkostninger, virksomheden har på sine produkter. De sidste to grupper af omkostninger er de *salgsfremmende omkostninger* og *anlægsomkostninger* i virksomheden. De salgsfremmende omkostninger har ifølge modellen direkte sammenhæng med aktiviteten, hvilket gør dem essentielle. Anlægsomkostningerne kan betragtes som de omkostninger, virksomheden har i forbindelse med opretholdelse af virksomhedens aktiver.

Når omkostningerne deles op efter denne model, sker det på baggrund af disses variabilitet. Variabilitet kan defineres som i hvilket omfang en given aktivitetsændring betinger en tilsvarende ændring i ressourcerne¹². Her følger de variable omkostninger aktiviteten, og kan herved klassificeres som fuldt variable. I den modsatte ende findes anlægsomkostningerne, som ikke varierer med virksomhedens aktivitetsniveau. Denne klassifikation er vigtig for virksomheden, idet den kan medvirke til at fastlægge budgetter, og træffe beslutninger på grundlag heraf.

I modellen beskrives ligeledes reversibiliteten af omkostningerne. Reversibilitet defineres som i hvilket omfang allerede anskaffede ressourcer kan afdisponeres uden tab¹³. Her er det logisk, at de variable omkostninger, som er fuldt reversible, følger aktiviteten i alle situationer. I modsætning hertil findes anlægsomkostningerne, idet disse ikke umiddelbart kan afskaffes, uden det betyder et tab for virksomheden. Med hensyn til reversibilitet og

¹¹[AR01], side 20

¹²[AR01], side 30

¹³[AR01], side 31

variabilitet må kapacitetsomkostningerne og de salgsfremmende omkostninger betragtes som liggende imellem de to andre omkostningstyper.

Ud fra disse betragtninger vil det være en fordel for virksomheden, at have flest mulige af sine omkostninger med høj variabilitet og reversibilitet. Alt andet lige vil dette give
5 en omkostningsstruktur, som vil være mindst sårbar overfor ændringer i aktiviteten.

Udviklingen i virksomheder og disses struktur har med tiden medført, at dækningsbidragsmodellen har en række svagheder. Ulempen ved dækningsbidragsmodellen er, at et stigende antal omkostninger ikke længere går direkte på produktet, men mere på kundepleje, branding, service med videre¹⁴. Herved vil der ikke være mange omkostninger,
10 som kan kategoriseres som værende variable omkostninger. Dette kan medføre en model, hvor det kan være svært at få et reelt billede af virksomhedens omkostningsstruktur. Dette kan være et problem, idet det vanskeliggør muligheden for at identificere, hvad der er lønsomt for virksomheden. Løsningen på denne udfordring er blevet at fordele kapacitetsomkostningerne. Der er igennem tiden udviklet en række modeller, som kan
15 hjælpe til at gøre dette, her kan der blandt andet nævnes full cost og ABC.

I praksis vil det betyde, at dækningsbidragsmodellen kræver, at virksomhedens omkostninger bliver lagt ud på forskellige konti, som indgår på hver sin post i resultatopgørelsen. Det er strukturen i disse konti, som gør det muligt at fremskaffe dækningsbidraget, idet det er omkostningstypen, som afgør under hvilken omkostningsgruppe omkostningen
20 ligger. Herved bliver det muligt for virksomheden at kunne fremskaffe de tal, som er relevante, under afrapporteringen efter dækningsbidragsmodellen. En ulempe er, at de variable omkostninger typisk bliver henført til en konto, og det derved ikke er muligt at undersøge, hvilke produkter disse vedrører. Dette kræver yderligere registreringer. Ellers er det kun muligt at beregne et samlet dækningsbidrag, som er vist i Formel 1.

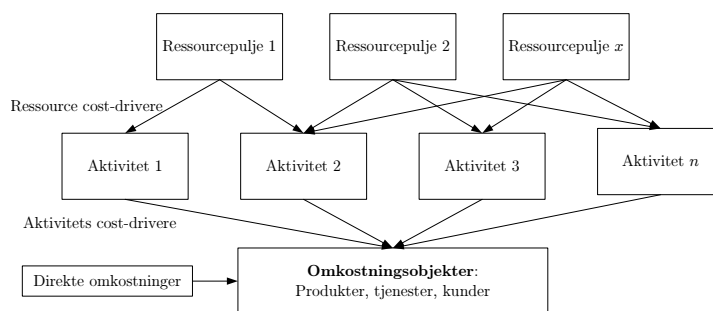
$$\text{Dækningsbidrag} = \text{Omsætning} - \text{Variable omkostninger} \quad (1)$$

¹⁴[KC98], side 3

4 Activity based costing

I dette afsnit vil ABC blive beskrevet. Der vil blive taget udgangspunkt i en kort forklaring af hovedprincipperne i ABC for at danne forståelsen for anvendelsen senere i denne rapport.

ABC tager udgangspunkt i, at hovedparten af alle aktiviteter, som virksomheder udfører, har til formål at skabe overskud gennem produktion og salg af produkter. Omkostninger skal derved fordeles på omkostningsobjekter eksempelvis enheder, produkter, produktgrupper, kunder og kundegrupper således, de svarer til disse gruppers træk på aktiviteten¹⁵. En skitse af grundstrukturen i ABC kan ses i Figur 3. Det er vigtigt at pointere, at dette blot er grundstrukturen. I praksis vil ABC blive tilpasset hver enkelt virksomhed ud fra de designkriterier, som bliver stillet op ud fra dette diagram. Det er derfor ikke muligt at angive noget standardsystem i ABC.



Figur 3: Grundstruktur i ABC systemet, frit efter [KC98]

15 ABC systemet kan som de traditionelle fordelingsregnskaber¹⁶ ses som en fremgangsmåde indeholdende to trin. Omkostningerne skal fordeles ud på de forskellige ressourcepuljer,

¹⁵[BI03], side 6

¹⁶[KC98], side 28–46

hvor de på forhånd er fastlagt til at indgå. Forskellen ligger i fordelingen af omkostningerne, hvor ABC-systemet fokuserer på aktiviteter. Aktiviteternes omkostninger bliver efterfølgende fordelt ud på virksomhedens forskellige omkostningsobjekter. Den store udfordring er her at få omkostningerne fordelt korrekt ud således, at ABC-systemet giver et reelt billede af situationen. For at fordele disse benyttes fordelingsnøgler, der betegnes cost-drivere. Cost-drivere er fordelingsnøgler, der giver et billede af den årsagsbestemte sammenhæng mellem forbruget af ressourcer, aktiviteter og omkostningsobjekter¹⁷. Der er her tale om henholdsvis ressource cost-drivere og aktivitets cost-drivere. Ressource cost-drivere benyttes i ABC-systemets første trin, hvor ressourcer/omkostninger fordeles ud på de aktiviteter, i hvilke de indgår. I andet trin benyttes aktivitets cost-drivere til at fordele aktivitetsomkostningerne ud på omkostningsobjekter. Her bliver fordelingen lagt på baggrund af omkostningsobjektets træk på aktiviteterne. Derved kommer fordelingsnøglerne til at afspejle omkostningsobjektets reelle ressource-træk. Der findes tre forskellige typer af aktivitets cost-drivere til fordeling:

15 **Transaktionsdriveren.** Denne anvendes til fordeling, hvis det er antallet af gange et omkostningsobjekt, der trækker på aktiviteten, som måles. Denne bruges, når alle omkostningsobjekter trækker på aktiviteten på samme måde. Her må der ikke være forskel på varigheden eller mængden i trækket på aktiviteten. Transaktionsdriveren er klart den letteste at måle og derved også den, som medfører færrest omkostninger for virksomheden.

20 **Varighedsdriveren.** Denne driver tager udgangspunkt i den tid, det tager at udføre aktiviteten. Driveren benyttes i tilfælde af, at omkostningsobjekternes måde at trække på aktiviteterne på er varierende og dermed ikke med fordel kan måles med transaktionsdriveren. Eksempler på dette kan være store forskelle i omstillingstid på en maskine eller varierende længde af kundebesøg. Varighedsdriveren kræver større og mere præcis viden om, hvordan virksomhedens arbejdsrutiner fungerer, og den er derfor mere omkostningstung at benytte for virksomheden.

25 **Direkte måling (intentitetsdriveren).** Denne er nødvendig, hvis ingen af de to nævnte drivere kan anvendes. Altså, hvis det ikke er muligt at fordele aktivitetsomkostningerne i form af antallet af gange, de er udført, eller hvis det ikke tidsmæssigt kan opdeles. Denne benyttes i tilfælde af, at det er nødvendigt at henføre omkostningerne direkte til omkostningsobjekterne ved anvendelse af arbejdssedler, edb-registreringer med videre¹⁸.

30 De direkte omkostninger, som for eksempel er ved produkterne, bliver i ABC henført nøjagtigt, som de bliver i de traditionelle fordelingsregnskaber. Det vil sige på de produkter, som de vedrører¹⁹.

ABC-modellen, som er beskrevet tidligere, er den oprindelige form for ABC-fordelinger fra [KC98]. Der er igennem tiden udviklet alternative modeller, som tager højde for

¹⁷[BI03], side 9

¹⁸[BI03], side 22

¹⁹[BI04], side 27

problemstillinger, som opstår i forbindelse med den oprindelige model. Af disse er hierarkimodellen²⁰, som deler omkostningerne op efter vigtighed for virksomheden, en af de mest omtalte. I denne rapport arbejdes der med, hvordan Sap Business One og dækningsbidragsmodellen/ABC kan integreres i forbindelse med datafangst til et data warehouse samt afrapportering. I det der er tale om at konceptuelt forslag over til data warehouse skal det blot tilpasses en given model. Derfor kan der argumenteres for, at det ikke er vigtigt hvilken ABC-model, der vælges. Der er valgt at benytte den oprindelige ABC-model til dette, men det er muligt at tilpasse systemerne, således enhver version af ABC vil kunne implementeres.

²⁰[BI04], side 51

SAP Business One

Efter at de økonomiske modeller, som vil blive benyttet i denne rapport, er blevet beskrevet vil dette afsnit omhandle kildesystemer. Kildesystemerne skal benyttes til at
5 levere data, som ved hjælp af de økonomiske modeller kan forvandles til rapporter og andet ledelsesinformation. Virksomhederne benytter i de fleste tilfælde ERP systemer, til at samle og håndtere disse data.

Et ERP system samler mange af virksomhedens afdelinger og funktioner i et enkelt IT system. Dette har til formål at opfylde alle de forskellige afdelingers individuelle krav.
10 Ofte har de forskellige afdelinger, såsom for eksempel salgsafdelingen og produktionsafdelingen, hvert sit IT system, der er optimeret til lige nøjagtig de funktioner, der er behov for i den afdeling. Disse systemer er ofte ikke designede til at kunne dele informationer. ERP derimod kombinerer alle disse delsystemer til et system, der afvikles på baggrund af en fælles database, hvilket gør, at de forskellige afdelinger nemt og hurtigt
15 kan dele informationer.

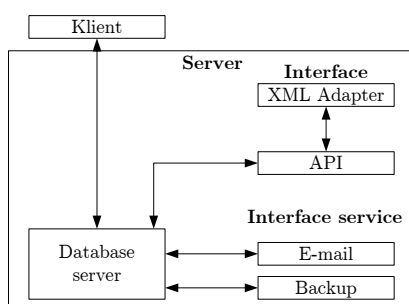
Eksempler på disse delsystemer (moduler) kan være kundehåndtering, finans, lagerstyring og produktionsstyring. Opdelingen er dermed resultatet af en funktionsbaseret tænkemåde. Fordelen ved at have alle virksomhedens funktioner samlet i et system er, at administrative arbejdsgange i virksomheden mindskes. For eksempel kan der i en virksomhed, som ikke benytter sig af ERP, når en ordre bliver afgivet, blive indtastet
20 data i et salgssystem i salgsafdelingen. Denne ordre vil derefter blive printet ud og blive afleveret til lageret eller eventuelt produktionsafdelingen. I disse afdelinger vil ordren blive indtastet i afdelingernes IT systemer. I et sådant forløb er der risiko for, at ordren bliver væk undervejs, hvilket logisk set kan medføre utilfredse kunder eller annullerede
25 ordrer.

Der findes forskellige ERP systemer såsom Microsoft Dynamics AX, Microsoft C5, SAP R/3 og SAP Business One. I denne rapport vil der blive anvendt SAP Business One, som udvikles og forhandles af SAP, der er en af de største leverandører af ERP systemer ifølge [Haz06, Koc06, Rel06]. SAP tilbyder en række styringssystemer til virksomheder
30 indenfor blandt andet produktionsstyring, supply chain management og ledelsesinformation. Blandt disse systemer er SAP Business One, som er udviklet som styringssystem

til små og mellemstore virksomheder. SAP Business One er en løsning til disse virksomheder, hvor det også er muligt at få en mere branchespecifik løsning ved at benytte produktet “mySAP all in one”. mySAP købes og udbygges i moduler, og kan således tilrettes den enkelte virksomhed. Endelig forhandler SAP en pakke, som hedder SAP R/3, der indeholder yderligere funktionalitet.

SAP Business One virker, som beskrevet tidligere, ved at få virksomhedens afdelinger til at samarbejde på et fælles grundlag, den centraliserede database. Arkitekturen i systemet kan ses i Figur 4 hvor en installation af en enkelt server er illustreret. Installationen er baseret på en to lags klient/server arkitektur, som tillader virksomheder at integrere systemet i sit allerede eksisterende Microsoft Windows netværk. Det skal arkitekturmæssigt bemærkes, at SAP understøtter flere databaser. Ligeledes giver arkitekturen mulighed for ved hjælp af et *application programming interface* (API) at kunne benytte *extensible markup language* (XML) standarden, så systemet kan levere data til for eksempel hjemmesider samt virksomhedens eget lokale netværk. Dette kan være en fordel for virksomheder, som for eksempel vil vise sin lagerstatus på en hjemmeside. Muligheden for at kommunikere med andre systemer gør også, at systemet kommunikerer direkte med samarbejdspartnere som for eksempel leverandører, banker og kurerbureauer.

Herved kan rutinemæssig kommunikation med disse foregå automatiseret. Dette kan for eksempel illustreres ved automatisk varebestilling. Medarbejderne i virksomheden kan opnå adgang til det centrale system fra de netværksmaskiner, som har SAP-klienten installeret. Det er muligt for administrationen at lave brugerrettigheder og opsætte regler for, hvad de enkelte brugere må udføre af funktioner.



Figur 4: Arkitektur i SAP Business One, frit efter [SAP05a]

SAP Business One kan deles op i seks moduler, som bliver bundet sammen af en fælles brugergrænseflade. Disse moduler arbejder tæt sammen. Dette gør det muligt at lave dataudtræk og rapporter baseret på elementer tværs over modulerne. Som det fremgår af Figur 5 på modstående side, tillader denne opbygning ligeledes individuelle tilpasninger af brugergrænsefladen samt “drag and relate”-teknologi, som gør det muligt at trække objekter mellem de forskellige moduler. Som en anden mulighed tilbyder SAP Business One ligeledes et software developer kit, der gør det muligt for virksomheden at udvikle egne applikationer og værktøjer, som kan hjælpe til at opfylde specielle behov i samarbejde med systemet. Som en anden mulighed kan virksomheder købe en række færdigudviklede add-ons for at kunne tilpasse systemet til det nøjagtige behov.

Rapportering og datanavigering					
Generelle teknologier (Drag&Relate, alarmer, brugerfladetilpasninger...)					
Finans	Salg	Service	Indkøb	Lager	Produktion
<ul style="list-style-type: none"> ◊Kontoplan ◊Kontosegmenter ◊Bogføring af post ◊Journalpostering ◊Ændringsbilag ◊Periodisk transaktion ◊Flere valutaer ◊Finansrapporter ◊Budget ◊Omkostningssted ◊Moms ◊Flere perioder ◊Indbetaling ◊Checks ◊Kreditter ◊Bilag ◊Betalingshenstand ◊Kontoudskrift og afstemning 	<ul style="list-style-type: none"> ◊Tilbud ◊Ordre ◊Faktura ◊Levering ◊Returnering ◊Prisliste i flere valutaer ◊Kundestyling ◊Beregning af bruttooverskud ◊Kontaktstyring ◊Muligheder og pipelinestyring ◊Outlook-integration 	<ul style="list-style-type: none"> ◊Styring af servicekontrakter ◊Service-administration ◊Aktivitetsoversigt ◊Vidensbase 	<ul style="list-style-type: none"> ◊Indkøbsordre ◊Restordre ◊Levering af indkøb ◊Returnering af indkøb ◊Faktura på indkøb ◊Kreditnota på indkøb 	<ul style="list-style-type: none"> ◊Lagerstyring ◊Lagerforespørgsel ◊Prisliste ◊Modtaget på lager ◊Leveret fra lager ◊Lagertransaktioner ◊Lageroverførsel ◊Serienumre ◊Styring af partier ◊Pick og pack ◊Kitting 	<ul style="list-style-type: none"> ◊Kalkulation på materialer ◊Arbejdsordrer ◊Produktionsordrer ◊Prognoser ◊MRP wizard ◊Anbefalingsrapport ◊Endelig montage håndtering
Softwareudviklingspakke					

Figur 5: Opbygning af SAP Business One, frit efter [SAP05b]

Udgifter kontra omkostninger

I dette afnit vil det blive beskrevet, hvordan udgifter og omkostninger vil blive håndteret i rapporten. Dette er vigtigt at fastlægge i forbindelse med benyttelsen af kildesystemet SAP Business One og ABC-/dækningsbidragsrapporteringen samt sikre den semantiske betydning, der tillægges begrebet. Problemstillingen opstår, da der i ERP systemer typisk bliver posteret udgifter. Denne betragtning gør sig ligeledes gældende i dækningsbidragsmodellen, som typisk tager udgangspunkt direkte i finanskontoplanen. I modsætning hertil lægger ABC op til rapportering ved virksomhedens omkostninger. Det er derfor vigtigt at fastslå, hvornår en udgift bliver til en omkostning for virksomheden.

“ En udgift fremstår på anskaffelsestidspunktet for en materiel handling, og er ækvivalent med anskaffelse af ressourcer²¹. ”

Ud fra definitionen på en udgift kan det udledes, at udgiften opstår når en kreditor har en fordring på virksomheden. En udgift opstår altså i forbindelse med en virksomheds anskaffelse af ressourcer.

Der kan være forskellige tidspunkter at registrere udgifter på. En indgangsvinkel til det kan være at registrere udgiften, når den kan forudses, eller alternativt først på faktureringsdagen. Dette er en regnskabsteknisk definition, som ikke direkte har indflydelse i denne sammenhæng. Under designet af data warehouset er det dog stadig vigtigt at være opmærksom på forskellen mellem udgifter og omkostninger. Forskellen på disse to begreber bliver tydelig, når der tages udgangspunkt i følgende definition.

²¹Frit efter [Mad63], side 26–29

“ En omkostning fremstår, når ressource bruges. ”
“ Udgangspunktet for omkostningerne er hermed ”
forbruget, med den dertil knyttede objek-
tivitetsgrad.²² ”

Omkostningen optræder først, når virksomheden reelt forbruger den ressource, som omkostningen vedrører. Derved kan virksomheden sagtens have haft udgiften længe før omkostningen finder sted. Dette kan være tilfældet, hvor der er købt ind til lager eller 5 en udgift til en maskine, hvor omkostningen først kommer senere i form af afskrivninger.

En løsning for virksomheden vil være at implementere direkte målinger på ressourceforbrug. En række af disse oplysninger vil også være mulige at hente ud fra SAP Business One, for eksempel med styklister. Herved vil omkostningsstyringsystemet arbejde 10 sammen med en række målesystemer i virksomheden. Denne fremgangsmåde kan være omkostningstung for virksomheden grundet de omfattende målinger. Et andet krav er, at målesystemerne er implementeret korrekt og måler præcist. Et eksempel på et sådant måleredskab kan ses i time/sags modulet, som er beskrevet i Afsnit 12.7 på side 58.

Virksomhedens udgifter skal periodiseres i forhold til afrapporteringen. Periodiseringen 15 skal foregå, så udgifter forbruges i den mindste afrapporteringsperiode. Et eksempel kunne være forsikring, hvor udgiften afholdes én gang årligt. Hvis afrapporteringen foregår på månedsbasis, vil det være nødvendigt at dele udgiften ud på de 12 måneder for at finde de reelle omkostninger. På samme måde bør afskrivninger periodiseres. Dette vil ikke give omkostningen, men en approksimation heraf. Herved bliver virksomhedens 20 udgifter til omkostninger indenfor en given tidsramme bestemt ved de periodiserede udgifter. Dette vil dog bære præg af en arbitrær tilregning.

Umiddelbart vil det være nødvendigt at tage stilling til, hvordan udgifter og indtægter, skal håndteres før dette overføres til et data warehouse. Dette skal ske i sådan en grad, at der i afrapporteringen kan gives et korrekt billede af virksomhedens omkostninger 25 i forbindelse med rapportering med dækningsbidragsmodellen og ABC. Det er valgt igennem denne rapport at behandle de posteringer, der er i SAP Business One direkte som omkostninger, således der ikke benyttes nogen yderlige transformation. Derved vil dette ikke give et helt reelt billede af virksomhedens omkostninger i data warehouset, men af kompleksitetsmæssige grunde er denne løsning valgt.

²²Frit efter [Mad63], side 26–29

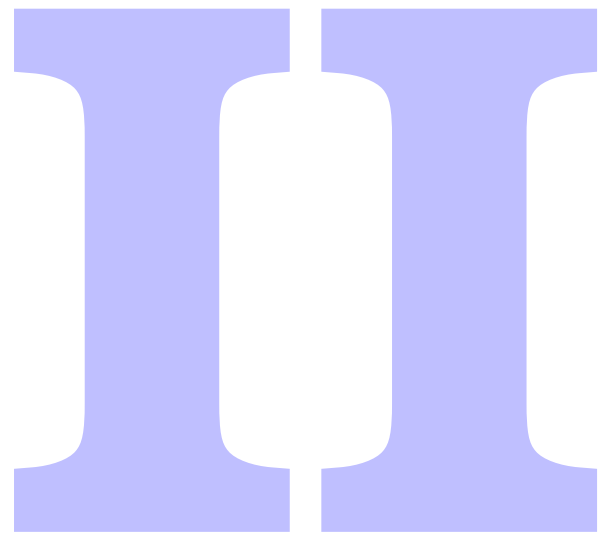
Opsummering 7

I dette afsnit er den valgte metode, samt forskellige økonomiske teorier blevet præsenteret. Disse danner grundlag for resten af rapporten, således at der er blevet skabt en grundlæggende forståelse, før udviklingen påbegynder.

Metodeafnittet beskriver den valgte metode, der vil blive benyttet i denne rapport. Her blev der argumenteret for, at gruppen har tillagt sig aktørsynet, idet denne rapport søger at forklare, de forskellige sammenhænge i forbindelse med udviklingen af et data warehouse og efterfølgende afrapportering. Derudover tilføjes de to paradigmedannende elementer logik og semantik, for at skabe bevidsthed om valg af ord og betydninger.

Udover dækningsbidrag og ABC, er SAP Business One blevet analyseret. Der blev argumenteret for, at der i denne rapport vil blive benyttet omkostninger som de står opført i SAP Business One.

Alt i alt, har dette kapitel dannet det teoretiske grundlag, som resten af rapporten vil bygge op om.



Data warehouse

Fokusområde 8

I dette kapitel vil der blive arbejdet med SAP Business One som et kildesystem til at kunne levere data til afrapportering ud fra ABC- og dækningsbidragsmodellen. Disse data skal registreres og opsamles i et samlet data warehouse, som ligeledes vil blive udviklet i kapitlet. Kapitlet vil begynde med en gennemgang af de teorier og værktøjer, som vil blive benyttet i arbejdet. Dette skal danne grundlag for forståelse af disse.

Efter, at de forskellige teorier og værktøjer er blevet analyseret, vil data warehouset blive implementeret. Det vil være hensigtsmæssigt at opstille data warehouset, før datagrundlaget bliver fastlagt. Designet af data warehouset tager udgangspunkt i de værktøjer, der har været brugt på semestret, hvor det søges at løse problemstillingen, som er sat op, og samtidigt analysere og forklare de enkelte processer i denne arbejdsgang. Processen ender i et fungerende data warehouse, som indeholder de data, der skal bruges for at opfylde kravene til rapporteringen. Der vil i denne rapport blive taget udgangspunkt i Ralph Kimball og Margy Rosses syn på data warehouse teori. Derfor vil dette kapitel beskæftige sig med at få oprettet en database, som skal fungere som et data staging område.

Næste trin i processen er, at der skal fyldes data i data staging området. Der vil blive taget udgangspunkt i SAP Business One som kildesystem til data warehouset. Dog vil der ligeledes blive benyttet yderligere kildesystemer på de områder, hvor SAP Business One ikke kan levere de data, der er brug for. Efter kildesystemerne er blevet analyseret, er der grundlag for at bruge ETL værktøjer for at hente data fra kildesystemerne. Dette kræver en række transformationer for at tilpasse data. Transformationerne vil også blive beskrevet. Til sidst vil de forskellige data marter blive oprettet.

Samlet set skal dette kapitel være med til at forklare, hvordan data kan kopieres fra kildesystemer til et data staging område, som videreføres til et data warehouse. Herved skal det designede data warehouse danne grundlag for en udvidet afrapportering. Den udvidede afrapportering vil finde sted i Kapitel III, hvor yderligere værktøjer til rapportering vil blive beskrevet. Idet data warehouset skal arbejde sammen med disse værktøjer, er det vigtigt at designet er velfungerende og lever op til de krav, der stilles. Dette vil der i rapporten blive lagt vægt på.

Grundstruktur

Når der skal træffes beslutninger i en virksomhed, gælder det om at have så fyldestgørende et datagrundlag som muligt. Dette datagrundlag kan eksempelvis være virksomhedens tidligere indkøb og salg. Disse data kan opsamles i et data warehouse og senere bruges til analyse. Indsamlingen af data sker fra de forskellige registreringssystemer, som virksomheden benytter. Indsamling og behandling af data fra kildesystemerne bliver beskrevet i Afsnit 10 på side 42. De systemer, som virksomheden benytter til de daglige aktiviteter, vil typisk indeholde en begrænset mængde historisk data. Eksempelvis vil typisk kun den aktuelle adresse for en kunde eller den aktuelle pris for en vare blive gemt. Hvis data regelmæssigt overføres til virksomhedens data warehouse, vil der være mulighed for at lave analyser på historisk data, som det ikke tidligere har været muligt at udføre. Denne overførsel af data vil alt andet lige belaste virksomhedens IT systemer, så det vil være logisk, at denne vil foregå i perioder med lav belastning på kildesystemerne.

Ifølge [KR02] består et data warehouse af tre dele som vist i Figur 6 på næste side. *Kildesystemer*, *data staging del* og et eller flere *data marter*. Kildesystemernes data renses og hentes ind i data staging, som beskrevet i Afsnit 10 på side 42.

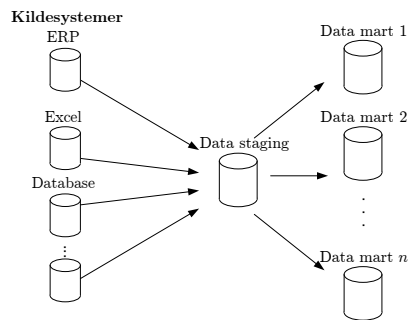
Data staging kan for eksempel være en relationel database, som indeholder alle data, som de enkelte data marter skal bruge. Det vil være logisk at lade data staging indeholde data fra flest mulige af virksomhedens systemer, hvilket vil gøre det muligt at benytte data staging til flere forskellige formål.

Et data warehouse består af en mængde af data marter²³. Et data mart er en dimensionel database, der leverer data til en enkelt eller flere tæt knyttede forretningsprocesser. Derfor er data martet i modsætning til data staging formålsspecifik. Formålet er at løse afrapportering og forespørgelser til netop denne forretningsproces.

Tabeldesignet i et data mart kan struktureres på flere forskellige måder:

Stjerneschema. Et denormaliseret tabeldesign. Dette resulterer i simple forespørgsler,

²³[KRRT98], side 19

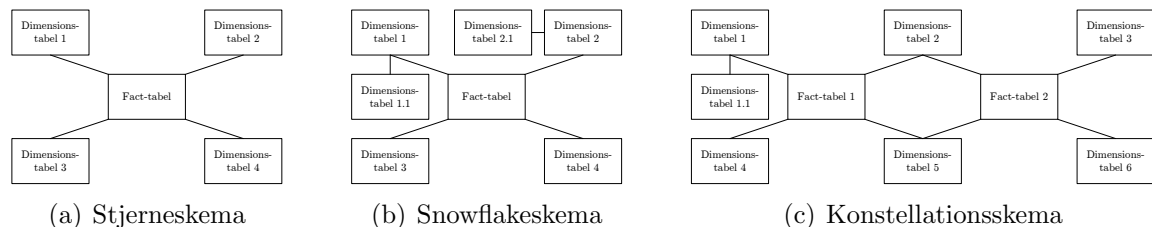


Figur 6: Flow i data warehouse

da antallet af tabeller, som skal joines, reduceres. Et eksempel kan ses i Figur 7(a)²⁴.

Snowflakeskema. Et delvist normaliseret tabeldesign, hvor kun dimensionerne skal være normaliserede. Et eksempel kan ses i Figur 7(b)²⁵.

- 5 **Konstellationskema.** Tabeldesign, hvor flere fact-tabeller deler en eller flere dimensionstabeller. Et eksempel på en dimensionstabel, som typisk vil blive delt mellem flere fact-tabeller, er en dimension, som indeholder tidspunkter. Et eksempel kan ses i Figur 7(c)²⁶.



Figur 7: Data warehouse typer

10 Tabeldesignet i et data mart vil bestå af et mindre antal fact-tabeller og flere dimensionstabeller. Fact-tabeller vil indeholde de primære data, som data martet skal stille til rådighed for analyser. Dimensionstabellerne indeholder data, som yderligere kan beskrive data gemt i fact-tabellerne. Endvidere vil data fra dimensionerne bruges til at filtrere de data, der ønskes rapporteret. I fact-tabellerne vil de fleste kolonner være numeriske og kan dermed benyttes til beregninger.

- 15 Et eksempel på en dimension, som ofte benyttes i et data warehouse, er tidsdimensionen. For at kunne udarbejde omfattende rapporter af eksempelvis, hvordan salg fordeles tidsmæssigt, er denne dimension veludbygget. Udover blot at kunne vise en dato, vil data

²⁴[LL03], side 8

²⁵[LL03], side 8

²⁶[CD97], side 6

kunne overskueliggøres ved også eksempelvis at vise ugedagen. Tidsdimensionstabellen fyldes ud med datoer, som kan refereres fra fact-tabeller. Endvidere gemmes eksempelvis også førnævnte ugedag og kvartal.

I dimensionerne skal der være primærnøgler. Primærnøgler kan deles op i naturlige og syntetiske. En naturlig primærnøgle benyttes, hvis der kan findes en unik identifikator i det data, der ønskes indsat. Et eksempel på en naturlig primærnøgle kan være cpr-nummer. Hvis der ikke findes en syntetisk primærnøgle, kan der vælges at benytte syntetiske primærnøgler, som ikke er en del af datasættet. Her vil de enkelte rækker blive tildelt en værdi, som fungerer som primærnøgle. Styringen af disse varetages normalt af et *database management system* (DBMS). Sandsynligvis vil kildesystemerne levere attributter, der kandiderer til at være primærnøgler i det endelige data warehouse. Eksempelvis kan forskellige kildesystemer tildele det samme produkt forskellige primærnøgler. For at kunne håndtere dette produkt, tildeles den en syntetisk primærnøgle i data warehouse²⁷. Ved at give den en syntetisk primærnøgle forhindres også på et senere tidspunkt problemer ved opdatering af data warehouse, hvis produktet har fået ny nøgle i kildesystemet. Derudover er dette også med til at give simplere og hurtigere forespørgsler^{28 29}.

Hvis en fact-tabel indeholder ordrelinier fra et salgssystem, vil kolonnerne eksempelvis indeholde antal og pris per styk. Data vedrørende produktet på ordrelinien vil være gemt i en dimensionstabel. I fact-tabellen vil der være en fremmednøgle til rækken i dimensionstabellen, der indeholder produktdata. Primærnøglen i fact-tabellen vil være sammensat af fremmednøglerne til dimensionerne. Dog vil det i nogle tilfælde ikke være teknisk muligt at oprette en primærnøgle baseret på alle fremmednøgler. I disse tilfælde kan der enten benyttes en syntetisk primærnøgle eller ingen primærnøgle.

Når data warehouse benytter data fra flere forskellige kildesystemer, bliver det også muligt at sammenkæde data. Når data ligger lagret i forskellige systemer, er det ikke muligt umiddelbart at foretage analyser på tværs af disse. I forbindelse med at data klargøres til indsættelse i data warehouse, behandles data så det kan sammenkædes med data fra andre systemer. I klargøringen foretages også rensning og validering af data således, at fejlbehæftede data ikke når frem til data warehouse. Dette vil yderligere blive beskrevet i Afsnit 10 på side 42. Ved at analysere på data, som ligger i et data warehouse i stedet for at være placeret i kildesystemerne, vil beregninger ikke genere den daglige brug af kildesystemerne.

Flere dimensionstabeller i et data warehouse vil typisk indeholde hierarkier. I tidsdimensionen ligger en måned under et år, en dag ligger under en måned osv. I produktdimensionen kan der være overordnede produktkategorier, som hver indeholder et antal produkter. Ved opbygning af hierarkier vælges der mellem to tabeldesigns, som også er med til at afgøre, om det er et stjerneschema eller snowflakeskema. Produktdimensionen kan eksempelvis bygges op med både produktkategorier og produkter i samme tabel, eller de optræder i hver sin tabel. Logisk set vil forespørgsler blive simplere, jo

²⁷[KR02], side 191

²⁸[KRRT98], side 147–148

²⁹[KR02], side 399

færre tabeller, der er involveret, fordi der dermed skal ske join af færre tabeller. Når den samme tabel indeholder både produktkategori og produkter, kan implementeringen ske på flere forskellige måder.

Parent-child. Her vil et produkt i tabellen have en fremmednøgle til en række i samme tabel, som indeholder data om produktkategorien.

Hierarki niveaubaseret kun registrering på nederste niveau. Her bygges ikke et hierarki op ved hjælp af flere rækker, som refererer hinanden. I stedet indeholder en kolonne navnet på for eksempel den aktuelle produktgruppe.

Hierarki niveaubaseret med registrering på overliggende niveauer. Her vil alle niveauer i hierarkiet optræde med en række i tabellen. Overliggende niveauer vil ikke indeholde værdier i kolonner, som kun benyttes på lavere niveauer. Denne kræver dog kendskab til dybden på hierarkiet

Ved parent-child er det ikke nødvendigt at kende dybden på hierarkiet på forhånd. De to andre implementationer kræver dette kendskab, da dybden afgør antallet af kolonner i tabellen.

Ved at udnytte niveauerne i hierarkiet kan der ske en mere avanceret afrapportering fra data warehouse. Dermed kan der afrapporteres på et overordnet niveau, eksempelvis virksomhedens samlede omsætning. For at få mere præcise afrapporteringer kan der foretages "drill down"³⁰ i produktkategorier og ned på produktniveau. Dette er blot et lille eksempel, men der er ingen tekniske grænser for antallet af niveauer. Til gengæld vil der være en grænse for, hvor mange niveauer det vil være praktisk mulige at håndtere.

I et data mart til ABC kan det være hensigtsmæssigt at fordele omkostningerne ,når dataene indsættes i data martet. Dette gør, at beregninger kan udføres simpelt og hurtigt. På denne måde vil data warehouse været et stjerneschema, hvor fact-tabellen har fremmednøgler ud til både aktiviteter og ressourcer. Fordelen ved denne fremgangsmåde er, at afrapporteringen bliver langt hurtigere og mindre kompliceret, idet alle beregninger foretages på det tidspunkt, hvor data indsættes i data martet. Dette har den ulempe, at nogle data logisk set ikke kan genskabes, og at andre opgaver vil blive svære eller endda umulige at løse i data martet. Da data staging databasen stadig indeholder de originale data, kan et nyt data mart skabes, hvis en ny opgave skulle opstå.

³⁰[Inm02], side 251–253

10 Extract, transform og load

Som nævnt i Afsnit 9 på side 38, er et data warehouse en samling af virksomhedens
5 vigtige data samlet i et antal data marter. Problemet er, at disse data måske findes i
forskellige systemer, som endvidere måske arbejder med forskellige formater. For eksem-
pel kan det være, at der skal bruges data fra et ERP-system og et eksternt time/sag
system. I disse to systemer kan medarbejdere være definerede på forskellige måder. Ek-
sempelvis kan de være definerede ud fra deres telefonnummer i ERP-systemet, mens de
10 i time/sag systemer er definerede ud fra et internt medarbejdernummer. Udfordringen
ligger i, at når data fra de to systemer skal samles i et data warehouse, skal disse flettes
sammen således, at medarbejderne fra time/sag systemet bliver relateret til medarbej-
derne fra ERP systemet. Dette kan i mange tilfælde betragtes som en overkommelig
opgave for en virksomhed med få medarbejdere. I nogle tilfælde er det data fra mange
15 kildesystemer, der skal samles, hvilket gør en manuel sammensætning af data næsten
umulig.

For at konvertere og sammensætte data fra forskellige kildesystemer til data, der kan
bruges i et data warehouse, kan virksomheden benytte sig af et *Extract, transform og*
load (ETL) værktøj. Et sådant værktøj læser data fra de forskellige kilder, renser det
20 for fejl, omformer det til et givet format samt skriver det til disk.

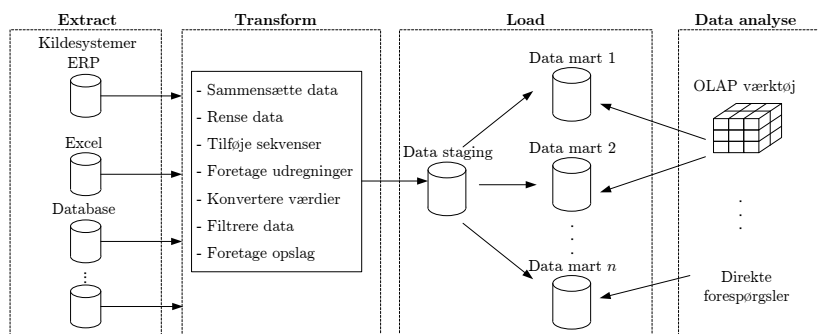
Der findes forskellige ETL værktøjer, som kan købes eller downloades. Eksempler på
sådanne ETL værktøjer er Kettle Spoon³¹, Visual Importer³² og Oracle Warehouse
Builder³³. Ofte vil der i større virksomheder blive udviklet et eget ETL værktøj, som
er tilpasset specielt til virksomhedens behov. Dette sker, hvis der er kritiske krav til
25 hastighed, eller der skal benyttes specielle transformationsregler, som ikke er under-
støttet af eksisterende produkter.

³¹Udgivet af Pentaho - <http://kettle.pentaho.org>

³²Udgivet af db Software Laboratory - <http://www.dbsoftlab.com>

³³Udgivet af Oracle - <http://oracle.com>

ETL processen kan deles op i tre faser. Figur 8 viser et overblik over ETL strukturen. Som det kan ses, er tegningen delt op i fire faser. Dette skyldes, at analyse fasen ikke er en del af selve ETL processen, men denne er dog taget med alligevel for at skabe overblik. Som det kan ses af Figur 8, sker der flest handlinger i transformationsskridtet. Dette skyldes, at transformationsskridtet er det, der binder de to andre faser sammen, og derfor er det også det, som indeholder langt de fleste værktøjer. Analysefasen indeholder forskellige værktøjer til at forespørge i data warehouset. Disse programmer kan lave grafiske fremstillinger af udtræk fra de forskellige data marter. Kapitel III omhandler data analyse i form af afrapporteringer.



Figur 8: Overblik over ETL struktur

- 10 I forbindelse med udarbejdelsen af denne rapport, vil der blive benyttet det gratis ETL værktøj, Kettle Spoon. I de følgende afsnit vil der blive gennemgået de enkelte skridt, der tages i ETL værktøjet. Underpunkterne er opbygget med et afsnit for hvert skridt i ETL processen.

10.1 Extract

- 15 Udvælgelsen fra kildesystemerne sker i dette trin. Kildesystemerne kan være i forskellige formater. Herefter er vist en liste over de mest almindelige kildesystemer.

Database. Mange data bliver i dag gemt i en eller anden form for database. Dette kan være data, der automatisk er opfanget fra forskellige sensorer og lagt i databasen. Data kan også stamme fra forskellige programmer rundt om i virksomheden. Også andre data warehouses kan være datagrundlag for data staging. Dette kan være tilfældet, hvis forskellige afdelinger har hvert sit data warehouse og ønsker at samle alle relevante oplysninger i et nyt data warehouse.

- 20 **ERP-system.** Mange ERP-systemer har en bagvedliggende database, hvori alle data bliver gemt. Ofte er der direkte adgang til denne database, så oplysningerne kan hentes ind i ETL værktøjet ved hjælp af SQL-forespørgsler. Hvis ERP-systemet ikke har en database, findes der ofte andre former for grænseflader til at kommunikere med ERP-systemet.

Flade filer. Nogle programmer gemmer deres oplysninger i flade filer, hvor oplysninger er separeret ved hjælp af komma eller lineskift. Dette kan for eksempel være autogeneratede data eller målinger fra sensorer.

Regneark. Mange virksomheder benytter sig af regneark til at opstille og udarbejde forskellige beregninger og regnskaber. Mange ETL værktøjer understøtter udvælgelse fra regneark.

XML. I dag er XML den mest udbredte måde at gemme struktureret data i filer uden brug af et DBMS[Con06]. XML har en logisk opbygning, som gør det nemt at overskue data og på den måde også nemt at finde netop de ønskede data.

Alle kilder fra de tidligere beskrevne punkter kan kombineres og sættes sammen således, at der kan findes nøjagtig de data, der ønskes.

10.2 Transform

Når data er blevet hentet ind i ETL værktøjet, kræves det ofte, at disse bliver behandlet, før de bliver sendt videre til data staging. Som nævnt, kan det ske, at nogle data er i forskellige formater. For eksempel kan det være, at vægtenheden af nogle varer er opgivet i gram og nogle i kilo. Derfor skal det bestemmes, hvilken granularitet data skal repræsenteres i, for eksempel, at alle vægtenheder bliver konverteret til gram. Et andet eksempel på forskellige formater er angivelse af køn på personer. I nogle systemer er kønnet angivet med et M eller F . I andre er de angivet med *mand* eller *kvinde*, og i tredje systemer er det angivet med 1 eller 0 . Alle disse formater skal konverteres til det bestemte format, der ønskes.

Som del af transformeringskridtet er også rensning af data. Til dette formål findes der værktøjer til at filtrere rækker, der indeholder bestemte værdier. Dette kan være, hvis et felt indeholder en specifik tekststreng med videre. Der er også mulighed for at benytte sig af mere komplekse filtermekanismer som for eksempel regulære udtryk, hvor der angives et udtryk for, hvordan de tekststrengene, som skal filtreres ud, skal se ud. Selvom denne rensning er fuldautomatisk, er der nogle data, som ikke kan renses automatisk. Her er der ofte tale om semantiske fejl i data, som for eksempel stavfejl, forkert enhedsangivelse for vægt og datofelter, hvor for eksempel leveringsdato kommer før bestillingsdato. Alle disse semantiske fejl kan der ikke tages højde for i ETL processen, idet dette kræver menneskelig indblanding for at detektere eller korrigere disse fejl. Derfor kan der vælges at rens data før, de bliver videresendes til ETL værktøjet, eller når det er blevet lagt i data staging. Selvfølgelig er der nogle fejl, der kan detekteres af ETL værktøjet. Eksemplet, hvor leveringsdato kommer før bestillingsdato, er en åbenlys fejl, hvor ETL værktøjet i nogle tilfælde kan spørge brugeren, hvad der skal gøres med disse data. Denne fremgangsmåde kan dog kun bruges, hvis det er små mængder af data, der skal læses ind i data staging, og der sidder en person foran computeren og holder øje med processen. Når store mængder af data skal lægges ind i data staging kan dette kræve, at

ETL værktøjet skal køre i flere dage og uger. Det kan kræve omfattende menneskelige ressourcer at rette fejl i en så stor datamængde. Derfor kan virksomheder blive nød til at acceptere, at der er fejl i data warehouset.

Som tidligere nævnt, kan ETL værktøjet håndtere, at der udvælges data fra forskellige kilder. At disse data skal joines sker også i dette trin, hvor der skal vælges på hvilke værdier, der joines.

Som nævnt i Afsnit 9 på side 38, kan alle rækker i dimensionerne identificeret ud fra en syntetisk nøgle. Denne nøgle associeres på transformationstidspunktet ved hjælp af en sekvens fra DBMS.

En af de dimensioner, der skal behandles specielt, er tidsdimensionen, som også nævnt i Afsnit 9 på side 38. Denne tabel indeholder ofte ikke data, der er fundet i kildesystemerne, men er udfyldt på forhånd, idet disse data kan genereres automatisk, og alle mulige værdier kan forudses. Mange ETL værktøjer indeholder funktioner til at generere en tidsdimension automatisk.

Alle beskrevne transformationer kan kombineres efter behov således, at det resulterer i de rækker og kolonner, der ønskes i den givne dimension eller fact-tabel.

10.3 Load

Når transformationen er fuldenendt, skal resultatet loades ind i data staging området, hvilket i dette tilfælde også er en database. Fra data staging bliver data loadet ud i de forskellige data marter, som danner det endelige data warehouse. Denne proces foretages tit i et transformationsværktøj og kræver i nogle tilfælde, at data igen bliver transformeret.

10.4 Kettle Spoon

Til dette projekt er ETL værktøjet, Kettle Spoon, blevet valgt. Dette er et gratis program, udgivet af Pentaho, som frit kan downloades fra internettet og bruges privat såvel som erhvervsmæssigt. Kettle Spoon understøtter den mest basale funktionalitet til at transformere data fra forskellige kilder. Derudover understøtter den Open Database Connectivity (ODBC), hvilket gør, at den understøtter kommunikation med de mest udbredte database systemer.

Kettle Spoon fungerer ved hjælp af *drag and drop*, hvor de ønskede funktioner trækkes ind på arbejdsområdet, og disse kan kombineres i vilkårlig rækkefølge. Dette gør processen mere overskuelig, da der kan fås et visuelt indtryk af opbygningen.

Ulempen ved Kettle Spoon er, at transformationen sker i små foruddefinerede skridt. I nogle tilfælde, kunne disse skridt måske have været kombineret og på den måde

opnå hastighedsforbedringer. Dette er en væsentlig ulempe, idet hastighed er essentiel i forbindelse med ETL processen, da denne ofte skal håndtere store mængder data, og derfor er hastighed kritisk. I dette projekt, er mængderne af data ikke så store, så derfor passer Kettle Spoon godt til dette.

11 Opbygning af data staging

Efter at teorierne og værktøjerne er blevet beskrevet, skal data staging skemaet opbygges. Der har i den forbindelse været en række metodemæssige overvejelser om, hvordan dette skal udføres. Problematikken ligger i, om data staging skemaet skal designes efter det datagrundlag, der er til rådighed, eller om data staging skal designes først, og derefter skal kilderne til det nødvendige data findes. Der er fordele og ulemper ved begge fremgangsmåder, der skal tages hensyn til.

Hvis datagrundlaget bliver fastlagt først, vil det logisk set kunne føre til en begrænsning af det endelige data warehouse. Grunden til, at begrænsningen opstår, kan være, at implementation bliver baseret på det virksomheden ved frem for det, den gerne vil vide. Derved menes, at datagrundlaget kommer til at spille en for stor betydning for, hvad der kan udledes af data warehouset. Dette kan give nogle uventede resultater, specielt hvis virksomhedens datagrundlag er for spinkelt i forhold til den information, som skal udledes deraf.

Hvis data staging bliver designet uden, at datagrundlaget er fastlagt, opnås en løsning, hvor det er muligt at hente alle data til de forskellige data marter for at generere det data afrapporteringen kræver. Ulempen bliver her, at det kan være svært at finde de data i kildesystemet, som skal føres over i data staging. Dette kan både skyldes, at data ikke er registreret, eller det er mangelfuldt registreret. Herved kan det data, som bliver fyldt i data staging, miste noget af validiteten, hvis ikke der stilles strenge krav til udvælgelsen af det. Er det ikke muligt at fremskaffe disse data fra kildesystemerne, bliver det nødvendigt at søge andre steder efter disse.

I denne løsning er der valgt at designe data staging ud fra den rapportering, der skal udarbejdes. Efterfølgende vil datagrundlaget blive identificeret i SAP Business One, samt andre kildesystemer, når det er nødvendigt. Dette sker ud fra den opfattelse, at det er den måde, som giver det stærkeste og mest mulige formålsuafhængige data staging,

hvilket sikrer et troværdigt og validt resultat baseret på et stort datagrundlag.

11.1 Krav til data staging

I forbindelse med udarbejdelse af data staging, skal der foretages nogle overvejelser om, hvad det skal bruges til. Som udgangspunkt skal data staging være så formålsafhængigt som muligt. Men selvom der stræbes efter en så formålsafhængig data staging som muligt, skal det stadig designes således, at det understøtter data, som de foruddefinerede data marter skal have. I forbindelse med dette projekt ønskes data warehouset som minimum anvendt til de efterfølgende beskrevne opgaver.

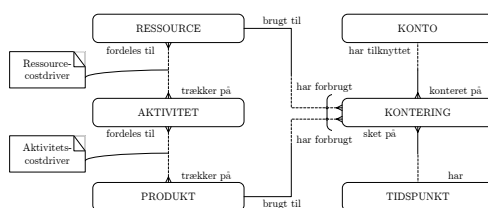
- Rapportering med dækningsbidragsmodellen
- 10 • Rapportering med ABC

For at data warehouset kan benyttes til de nævnte opgaver, er det besluttet, at hovedtabellen i data staging skal indeholde alle finansielle konteringer på de forskellige konti. Logisk set skulle disse konteringer indeholde de data, der er brug for til de to rapporteringer, som er nævnt tidligere, da de indeholder alle indtægter og udgifter virksomheden har haft. Det er dog vigtigt her at være opmærksom på forskelle i de forskellige modellers syn på udgifter og omkostninger. Forskellene, og hvordan disse vil blive håndteret i denne implementering, er beskrevet i Afsnit 6 på side 30. I forbindelse med ABC-rapportering kan ressourcerne findes ud fra virksomhedens kontoplan³⁴. Idet de finansielle konteringer sker på forskellige konti i kontoplanen, vil det logisk set være muligt at tilknytte en ressource til alle linier i konteringstabellen, og dermed vil det være muligt at lave ABC-rapportering på baggrund af de finansielle konteringer.

Idet de finansielle konteringer er så generelle, opnås en relativ formålsafhængig data staging, der ikke kun kan bruges til de to nævnte opgaver, men også til andre opgaver, som kræver analyse.

For at skabe overblik over hvilke data, der er brug for, er der blevet udarbejdet et ER-diagram, der angiver sammenhængen mellem de forskellige data. Dette ER-diagram kan ses i Figur 9 på næste side. Det er vigtigt at understrege, at dette ER-diagram ikke danner direkte grundlag for tabeldesignet. Det danner blot et konceptuelt overblik. Diagrammet viser, at en kontering enten kan henføres direkte til et produkt, eller skal henføres til en ressource, som fordeles til produkter. Dette kan ses ud fra den exclusive arc på ER-diagrammet. Derudover kan det ses, at der er en mange-til-mange relation mellem ressourcer og aktiviteter og mellem aktiviteter og produkter. Disse mange-til-mange relationer repræsenterer de forskellige drivere.

³⁴[KC98], side 86

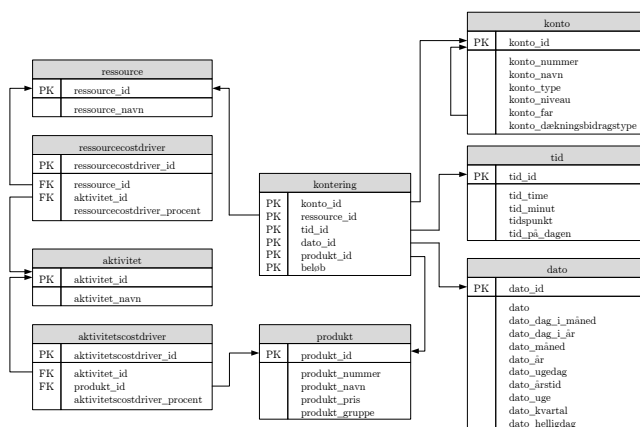


Figur 9: Overblik over det konceptuelle forslag

11.2 Design af data staging

I dette afsnit vil designet af data staging og tabellerne deri blive gennemgået. Hele data staging-designet kan ses i Figur 10.

Selvom data staging ikke er et direkte data warehouse, er det valgt at benytte et data warehouse-lignende design. Denne er et snowflakeskema, hvor tabellerne, der skal bruges til ABC-delen, er designet som *snowflakes*, og resten af data staging er et almindeligt stjerneschema. Dette stemmer også overens med overblikket over det konceptuelle forslag i Figur 9. Herefter vil de enkelte tabeller i figuren blive beskrevet.



Figur 10: Design af data staging

Konto. Denne tabel indeholder data om de forskellige konti. Tabellen indeholder samtlige konti fra kontoplanen. Udover kontonummeret og kontonavnet indeholder dimensionen også en type-kolonne, som identificerer, hvilken type konto der er tale om. Eksempler på værdier kan være *Aktiver*, *Passiver*, *Udgifter* og *Indtægter*. Denne tabel skal bruges til at identificere, fra hvilken konto posten i *Kontering* tabellen kommer fra.

I tabellen *Konto* findes også kolonnen **konto_dækningsbidragstype**. Denne benyttes til at kategorisere konteringer i forhold til dækningsbidragsrapportering. Det angives i denne kolonne, om kontoen vedrører eksempelvis faste eller variable omkostninger og dermed, hvor de skal optræde i en dækningsbidragsberegning.

Tid. Tidstabellen definerer, hvilken tid den enkelte kontering er sket på dagen. Tabellen indeholder det fulde tidspunkt i et givet format som for eksempel *14:21*. Derudover indeholder dimensionen også kolonner, der indeholder henholdsvis kun timetallet og minuttallet. Dette er gjort for, at det skal være muligt at lave relativ simple forespørgsler på for eksempel timetallet. Hvis der ønskes analyse på konteringer mellem klokken 10 og 12, så er dette nemt gjort. Til sidst er der en kolonne, der indikerer, hvilken tid på dagen konteringen er sket på. Dette kan for eksempel være *Formiddag* eller *Eftermiddag*.

Dato. Datotabellen definerer, på hvilken dato den enkelte kontering er sket. Tabellen indeholder først og fremmest dato i et givet format. Dette kan for eksempel være *2007-04-13*. Derudover indeholder datotabellen ligesom tidstabellen en række yderligere kolonner, som er med til at gøre forespørgsler og analyse lettere.

Alternativet til at have en separat dato- og en tidsdimension er at samle dem i én tabel som vist på overblikket over det konceptuelle forslag i Figur 9 på forrige side. På denne måde vil der for hver dato være en række tidspunkter. Fordelen ved denne fremgangsmåde er, at der kommer færre tabeller og dermed færre fremmednøgler i de endelige fact-tabeller, hvilket kan føre til hurtigere forespørgselstid. Når tidspunkter er registreret for alle datoer, kan der tilføjes yderligere information. Afhængig af hvilken type virksomhed, som data warehouset skal bygges for, kunne det være relevant at gemme, om et tidspunkt ligger indenfor normal åbningstid. Dette er ikke muligt når dato og tidspunkt ligger i adskilte tabeller, hvis dagene har forskellige åbningstider. Ulempen ved denne fremgangsmåde er, at en sådan dimension vil blive utrolig stor og vil være besværlig at udfylde på forhånd, hvis der er valgt en fin granularitet. Derudover vil tabellen blive stor, hvilket gør, at det vil være langsomt at søge igennem. Når dato og tidspunkt placeres i samme tabel, kan der ved fin granularitet opstå en situation, hvor ingen rækker i dato/tidspunkt tabellen refereres af mere end én række i fact-tabellen. Dermed opnås der ingen fordele ved at gemme disse data i en anden tabel, og de kunne derfor alternativt gemmes i fact-tabellen. Derfor er der valgt i dette projekt at benytte to tabeller.

Ressource. Denne tabel indeholder de forskellige ressourcer. Denne tabel er simpel og indeholder kun ressourcenavnet. Der kan argumenteres for, at ressourcetabellen kunne slås sammen med kontotabellen, idet ressourcerne ofte er defineret ud fra kontoplanen³⁵. Grunden til, at denne er valgt som en selvstændig tabel, er, at det kan være, at der er nogle konti, som ønskes henført til forskellige ressourcer. Et eksempel på dette kan være gager, som vil blive konteret på samme konto, men disse ønskes ofte ført hen til ressourcer, der passer til den givne afdeling.

Ressourcecostdriver. Denne tabel er en mange-til-mange relation, som binder ressourcerne og aktiviteterne sammen. Driverne er angivet i procent.

Aktivitet. Aktivitetstabellen indeholder alle de forskellige aktiviteter. Tabellen er simpel, idet den kun indeholder én kolonne, nemlig aktivitetsnavnet. På grund af data

³⁵[KC98], side 86

stagingens opbygning, er det relativt simpelt at udbygge denne til at kunne understøtte hierarkier på aktiviteterne. Denne udformning gør, at der kan undgås at fordele fællesomkostninger³⁶. For at gøre data warehouset i stand til at understøtte hierarkier på aktiviteter, kræver det blot at der tilføjes en kolonne til **Aktivitet** tabellen i data staging, der referer dennes far i samme tabel. Derudover bør der tilføjes en kolonne, der håndterer versionsstyringen. På denne måde understøtter data warehouset både aktiviteter med og uden hierarkier. Dette kræver selvfølgelig, at der angives drivere til alle aktiviteterne.

Aktivitetscostdriver. Denne tabel er en mange-til-mange relation, som definerer driverne mellem aktiviteter og omkostningsobjektet, som i dette tilfælde er produktet. Tabellen er opbygget på samme måde som **Ressourcecostdriver** tabellen.

Produkt. Produkttabellen indeholder virksomhedens produkter. Denne tabel er speciel, idet den bliver refereret både fra **Aktivitetscostdriver** og **Kontering** tabellerne. Grunden til, at denne tabel bliver brugt af begge tabeller, er, at hvis en kontering er en direkte omkostning, vil den blive henført direkte til et produkt i denne tabel. Hvis omkostningen derimod er en indirekte omkostning, skal denne fordeles. Derfor skal denne henføres til en ressource i stedet. Dette vil sige, at **Kontering** tabellen i teorien kun skulle referere til enten **Ressource** eller **Produkt** tabellen. Men da data warehouse teorien fraråder, at der forekommer *NULL* værdier i fact-tabellen³⁷, vil de to tabeller indeholde henholdsvis en række med værdierne Denne kontering er en direkte omkostning og Denne kontering er en indirekte omkostning. På denne måde vil der altid være en reference ud til alle dimensioner. Udover produktnavnet, indeholder tabellen også en kolonne, der identificerer, hvilken produktkategori det enkelte produkt hører til.

Kontering Denne tabel er hovedtabellen i data staging. Den indeholder alle konteringerne fra virksomheden. Tabellen indeholder fremmednøgler til de resterende tabeller. Denne tabel minder om fact-tabellen i de endelige data marter. Alle disse fremmednøgler er tal, der refererer til syntetiske primærnøgler. Udover fremmednøgler indeholder tabellen også en kolonne, der indeholder det konterede beløb. Primærnøglen er en sammensat primærnøgle, som består af alle kolonnerne i tabellen. Alle disse kolonner tilsammen anses for at kunne identificere rækken.

Alle de nævnte tabeller skaber et relativt formålsafhængigt data staging, som senere kan danne grundlag for mere specifikke data marter. Disse data marter kan håndtere afrapportering på baggrund af minimum dækningsbidrag og ABC. Dog ville virksomheder i mange tilfælde ønske at undersøge kunders lønsomhed og adfærd. Dette er i data stagingens nuværende form ikke muligt. Hvis en sådan analyse ønskes, skal data staging udvides med flere tabeller. Figur 11 på næste side viser et sådant data staging skema, der er udvidet med ordrelinier og kunder. Tabellen, der indeholder ordrelinier, er fact-tabel lignende, hvilket gør, at skemaet bliver til et konstellationsskema. Med disse to ekstra

³⁶[BI04], side 51

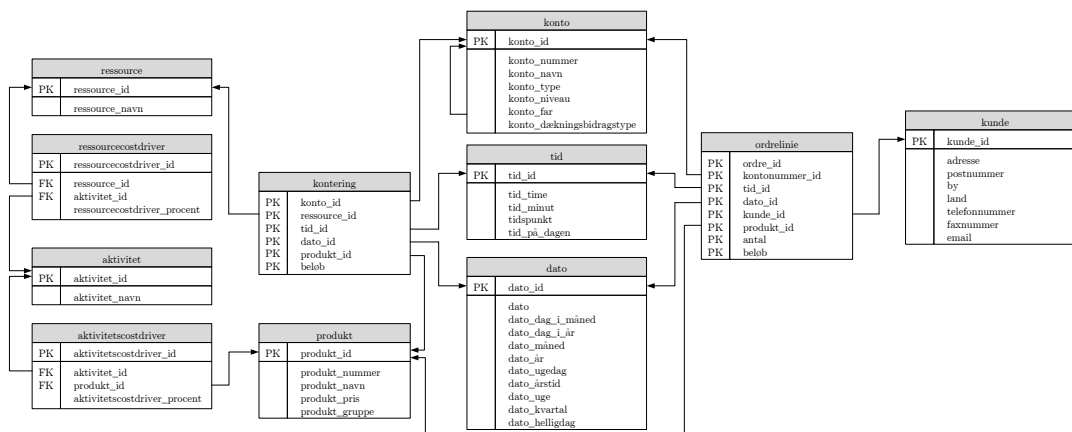
³⁷[KR02], side 49

tabeller er det muligt at undersøge kundernes adfærd, ud fra købte produkter. Dette kan for eksempel være, om der er komplementære produkter, der bliver købt mere, når kunderne også køber et andet produkt. Herefter vil disse to tabeller blive beskrevet.

Kunde. Kundetabellen indeholder alle data om kunderne. Dette inkluderer både adresse, by og land. Ud fra disse oplysninger kan der laves analyse på kundernes geografiske placering i forhold til solgte varer. Dette kan være brugbart i henhold til logistiske spørgsmål.

Ordrelinie. Tabellen indeholder ordrelinier fra de enkelte ordrer. Den indeholder fremmednøgler til de gamle tid-, dato-, konto- og produkttabellerne. På denne måde kan der i en analyse ses, hvilke produkter der er blevet købt, og på hvilket tidspunkt det er gjort.

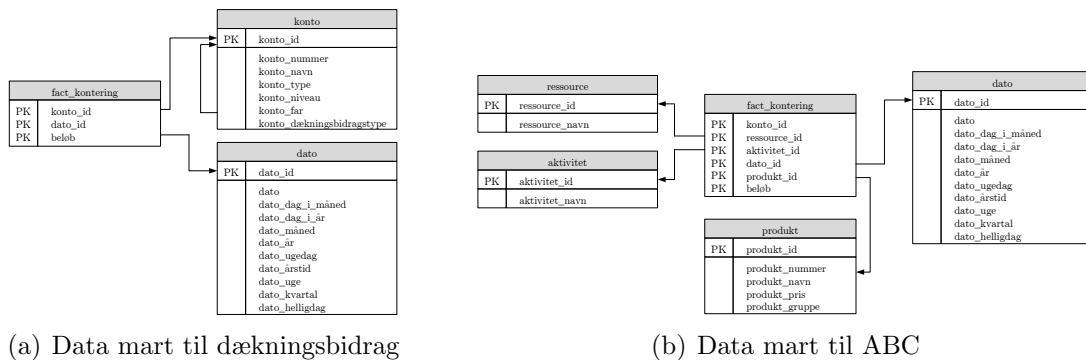
Grundet den øgede kompleksitet er det valgt at undlade dette tillæg til data stagingen i resten af denne rapport. Dette vil give unødigt kompleksitet af ETL udformningen, og samtidigt vil dette ikke give flere data til det primære formål for data warehouset, nemlig dækningsbidrags- og ABC-rapportering.



Figur 11: Design af et udvidet data staging skema

11.3 Design af data marter

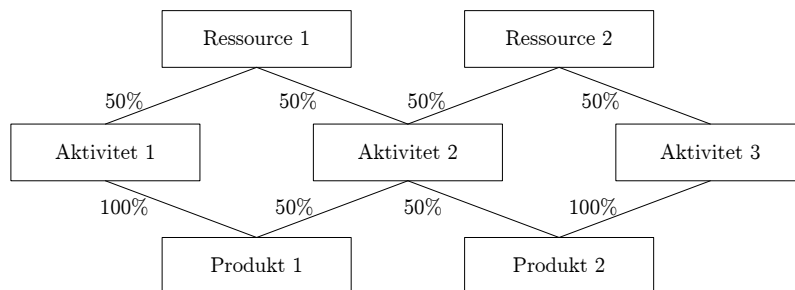
Efter data staging er blevet designet, skal de enkelte data marter, som udvikles, danne det egentlige data warehouse. Som tidligere nævnt i Afsnit 11.1 på side 48 er formålet med data warehouset, at det som minimum skal kunne håndtere rapportering af dækningsbidrag og ABC. Til dette formål, vil der blive designet to data marter til at håndtere hver sin rapportering.



Figur 12: Data warehouse

Grundet designet af data staging skemaet kan de forskellige tabeller bruges direkte som konforme dimensioner i data marterne. Dette gør, at der ikke kræves nogen transformationer for at kopiere dem til de enkelte data marter. Derfor er det relativt simpelt at generere de forskellige data marter.

- 5 At lave et data mart, der kan bruges til rapportering af dækningsbidrag, er simpelt. Dette kræver blot, at tre tabeller fra data staging kopieres til det nye data mart. Disse tabeller er **Konto**, **Dato** og **Kontering**. Her vil **Kontering** blive til fact-tabellen i data martet. Ud fra dette data mart er det muligt at finde både indtægter og udgifter fra kontodimensionen. Designet af dette data mart kan ses i Figur 12(a).



Figur 13: Fordeling af omkostninger

- 10 For at lave et data mart, der kan bruges til rapportering af ABC, er dette mere avanceret. Selve dimensionerne kan stadig kopieres direkte til data martet. De tabeller, der her er brug for til dimensioner, er **Konto**, **Dato**, **Ressource**, **Aktivitet** og **Produkt**. Men til forskel fra data staging vil **Aktivitet** tabellen blive en dimension direkte fra fact-tabellen. Et samlet tabeldesign vises i Figur 12(b). Fact-tabellen skal dog se anderledes ud end i data staging. Denne indeholder stadig konteringer, men disse konteringer vil blive delt op i flere linier således, at de repræsenterer fordelingen. Figur 13 viser et simpelt ABC-system, med fordelinger angivet i procent. Figur 14(a) på den følgende side viser, hvordan en kontering i data staging vil se ud. Figur 14(b) viser, hvordan fact-tabellen vil se ud for denne kontering i det nye data mart. Det ses at **Ressource**
- 20 **1** fordeles ligeligt til **Aktivitet 1** og **Aktivitet 2**. Herfra fordeles **Aktivitet 1** fuldt

AFSNIT 11. OPBYGNING AF DATA STAGING

til Produkt 1, og Aktivitet 2 fordeles ligeligt til Produkt 1 og Produkt 2. Dette vil resultere i tre linier i det nye data mart.

Beløb	Ressource
100 kr.	Ressource 1
⋮	⋮

(a) Data staging

Beløb	Ressource	Aktivitet	Produkt
50 kr.	Ressource 1	Aktivitet 1	Produkt 1
25 kr.	Ressource 1	Aktivitet 2	Produkt 1
25 kr.	Ressource 1	Aktivitet 2	Produkt 2
⋮	⋮	⋮	⋮

(b) Data mart

Figur 14: Data staging og fact-tabel i ABC data mart

12 Datagrundlag

Efter data staging skemaet er blevet designet i Afsnit 11 på side 47, vil dette afsnit omhandle, hvordan data kan identificeres i de forskellige datakilder, som for eksempel SAP Business One. Kapitlet viser, fra hvilke tabeller data skal tages, for at fylde data i data staging.

SAP Business One har ofte en stor position i virksomhedens IT struktur. Derfor er det oplagt at anvende denne som datagrundlag for et data warehouse. Det må forventes, at langt de fleste data i virksomheden, som omfatter interne forhold som produktion og medarbejdere, men også data om eksterne kunder og leverandører kan findes i SAP Business One. Da SAP Business One er bygget op over en relationel database, er der en oplagt mulighed for at finde de data, der skal bruges til data warehouset, i de specifikke tabeller i SAP Business One.

Et oplagt eksempel på brug af data fra SAP Business One er at anvende data til kundedimensionen. For at finde disse oplysninger i SAP Business One skal den eller de relevante tabeller findes i den bagvedliggende database. Problemet er, at denne database indeholder et stort antal tabeller, som ikke har nogle intuitive navne, hvilket gør det stort set umuligt at finde de relevante data, når tabelnavnet ikke er kendt. Derfor skal tabelnavnet først findes, før data kan trækkes ud databasen. For at finde de relevante tabeller kan dette gøres fra SAP Business One. For at kunne se tabelinformationerne skal **System-informationer** først slås til. Når denne funktion er slået til, vil tabelinformationerne vises, når muse-pointeren holdes over et felt. Denne information vises i bunden af SAP Business One programmets vindue, og vises som navnet på tabellen og kolonnen, hvori data er gemt. Et eksempel på dette er feltet **Navn** under **Forretningspartnere**, som henter sine data fra tabellen **OCRD** i kolonne **CardName**. Yderligere information om tabelopbygning og valg af data typer kan fremskaffes ved at åbne database manageren og finde den givne tabel og derunder kolonne. **OCRD** beskrives yderligere på Figur 15 på den følgende side. Her ses kolonnenavnet, datatypen og størrelsen på kolonnen. Endvidere følger der et *software developer kit* (SDK) med SAP Business One, og heri kan der også findes beskrivelser af tabeller og deres indhold. Disse oplysninger er vigtige, når ETL værktøjet skal konfigureres.

Kolonne	Datatype	Længde
CardName	NVarChar	100
Currency	NVarChar	2
Notes	NVarChar	100
CardCode	NVarChar	15
⋮	⋮	⋮

Figur 15: Eksempel fra tabellen OCRD

Problemet med de datatyper SAP Business One bruger er, at de stammer fra *Microsoft SQL Server* (MSSQL), hvilket ikke er direkte i overensstemmelse med de datatyper, som bruges i Oracle. Derfor skal der vælges nogle datatyper fra Oracle, som korresponderer til de datatyper SAP Business One har valgt. For eksempel svarer datatypen `NVarChar` i MSSQL til datatypen `Varchar2` i Oracle. Derudover skal der selvfølgelig vælges samme længde på kolonnerne i data warehouse for at undgå, at der opstår fejl i sammenførelsen. Hvis en kolonne fylder mere end længden defineret i data staging, vil konverteringen fejle. Dette er ikke hensigtsmæssigt, da der så opstår fejl i ETL processen, hvis ikke dette håndteres eksplicit.

12.1 Tid tabellen

Tid tabellen er speciel, fordi denne ikke får data direkte fra SAP Business One. Denne tabel bliver genereret af et eksternt program, hvor alle mulige tidspunkter lægges ind i tidstabellen. Dette er muligt, fordi der kun er et begrænset antal rækker, der skal være i tabellen. Da tabellen kun indeholder timer og minutter, indeholder den kun $60 \times 24 = 1440$ rækker. Genereringen af tidspunkter kan ske på flere måder. Genereringen kan ske i et eksternt program, som resulterer i en fil, der kan bruges som kilde i ETL værktøjet. Selve genereringen kan også ske direkte i ETL værktøjet. I dette projekt er genereringen af tidsdimensionen sket i et Excel-ark.

12.2 Dato tabellen

Dato tabellen minder om Tid tabellen. Også her genereres data uden brug af SAP Business One. Dog er der ikke et fast antal datoer, ligesom der var et fast antal tidspunkter, men datoer kan genereres til langt ud i fremtiden, og hvis der så på et tidspunkt bliver brug for flere datoer, så kan flere blot indsættes. Genereringen i forbindelse med dette projekt vil ske direkte i Kettle.

12.3 Konto tabellen

Konto tabellen tager udgangspunkt i OACT tabellen. Data om konti kan findes i SAP Business One. De data, der er brug for, er kontonummer, kontonavn og kontotype. Primærnøglen i Konto tabellen er en syntetisk nøgle, som bliver styret af en sekvens i Oracle. Figur 16 viser, fra hvilke tabeller og kolonner de enkelte kolonner i Konto tabellen stammer fra. Kontotypen er i databasen bag ved SAP Business One repræsenteret ved hjælp af et tal. For eksempel er aktiver repræsenteret ved et ettal. På denne måde, skal der ske en mapping mellem disse tal og kontotypen. Dette skal ske i ETL værktøjet, og vil blive beskrevet i Afsnit 13.3 på side 63.

Kolonne	Tabel	Kildekolonne	Datatype	Længde
konto_nummer	OACT	AcctCode	NVarChar	15
konto_navn	OACT	AcctName	NVarChar	100
konto_type	OACT	GroupMask	SmallInt	2

Figur 16: Kilde fra SAP Business One til Konto tabellen

12.4 Produkt tabellen

Produkt tabellen indeholder alle produktspecifikke data. Den primære kilde til disse data er OITM tabellen fra SAP Business One. De data, der er brug for, er produktnummer, produktnavn og produktpris. Derudover er der, ligesom Konto tabellen, brug for en syntetisk primærnøgle, som bliver styret af en Oracle sekvens. Figur 17 viser, fra hvilke tabeller de enkelte kolonner i produktdimensionen stammer. Enkelte værdier stammer ikke fra OITM tabellen. Dette er kolonnerne **produkt_pris** og **produkt_gruppe**. Produktprisen stammer fra ITM1 tabellen, hvor prisen for hver prisliste er defineret. Derfor skal der foretages et opslag i denne tabel baseret på produktnummeret og den valgte prisliste. Produktgruppen stammer fra OITB tabellen. Denne indeholder blandt andet gruppekoden og gruppenavnet. Derfor skal der blot foretages et opslag på baggrund af gruppekoden.

Kolonne	Tabel	Kildekolonne	Datatype	Længde
produkt_nummer	OITM	ItemCode	NVarChar	20
produkt_navn	OITM	ItemName	NVarChar	100
produkt_pris	ITM1	Price	Numeric	9
produkt_gruppe	OITB	ItmsGrpNam	NVarChar	20

Figur 17: Kilde fra SAP Business One til Produkt tabellen

12.5 Ressource tabellen

Ressource tabellen udspringer som udgangspunkt i kontoplanen. Men som argumenteret i Afsnit 11.2 på side 49 er denne valgt som en selvstændig tabel. Men selvom disse er to selvstændige dimensioner, er der en stærk sammenhæng mellem dem, som ETL værktøjet bør kende til. Dette skal bevirke at, hvis ikke andet er angivet, så skal ETL værktøjet benytte sig af denne sammenhæng. Denne sammenhæng kan defineres på forskellige måder. Der er to åbenbare måder, som sammenhængen mellem ressourcer og konti kan defineres på. Den første løsning er ved at tilføje et brugerdefineret felt i SAP Business One, som eksplicit angiver for hver enkelt konto, hvilken ressource denne konto tilhører. Den anden løsning er at have en ekstern fil, hvor sammenhængen mellem ressourcer og konti er defineret. Denne fil kan for eksempel være en komma-separeret fil eller et Excel-ark. I dette projekt er et Excel-ark blevet valgt, hvor sammensætningen således sker i ETL værktøjet. Figur 18 viser et lille udsnit af dette Excel-ark.

Kontonummer	Kontonavn	Ressource
60100	Annoncer, PR og markedsinfo.	Markedsføringsafdeling
60000	Brændstof, olie, vask	Biler
60410	Telefon og øvrige kommunikationsmidler	Callcenter
60010	Reparationer	Service og teknik
⋮	⋮	⋮

Figur 18: Eksempel på kildedata til Ressource tabellen

12.6 Aktivitet tabellen

Aktivitet tabellen indeholder alle aktiviteter, som ressourcerne skal fordeles til. Som udgangspunkt findes disse aktivitetsnavne ikke direkte i SAP Business One, da denne ikke direkte understøtter ABC. Derfor skal aktiviteterne findes på anden vis. Logisk set kan aktiviteterne blot specificeres i en ekstern fil, idet virksomheden identificerer sine aktiviteter under ABC-analysen. Derfor vil aktiviteterne i dette projekt også være defineret eksplicit i et Excel-ark.

12.7 Ressource cost-driver

For at få mulighed for at fordele ressourcernes omkostninger på aktiviteter er det nødvendigt at registrere eksempelvis medarbejderes tidsforbrug på aktiviteterne. SAP Business One giver begrænset mulighed for at registrere, hvor lang tid en medarbejder bruger på eksempelvis et møde med en kunde. Dette giver dog kun en begrænset hjælp til at beregne ressource cost-drivere. Selvom tiden brugt på møder bliver registreret,

kendes jo ikke den tid, medarbejderen bruger på andre aktiviteter. Derudover vil brug af kunder som omkostningsobjekter medføre, at eksempelvis omkostninger ved møder bliver direkte, og altså ikke skal fordeles. Endvidere registreres kun, hvor lang tid selve mødet har taget og ikke, hvor lang tids forberedelse medarbejderen har haft før mødet.

- 5 Derfor er det nødvendigt at anvende systemer udover SAP Business One til at registrere medarbejdernes tidsforbrug. Afhængig af hvilke ressourcer, der arbejdes med i den aktuelle ABC-implementation, kan det også være nødvendigt at registrere forbrug af eksempelvis råmaterialer. For at udvikle applikationer til registrering af ressourceforbrug kan Oracle Application Express anvendes. Denne giver mulighed for hurtigt at udvikle
- 10 grafiske brugergrænseflader, der giver let adgang til at oprette og ændre data. Et sådant system blev udviklet på sidste semester. Systemet kan håndtere medarbejdernes tidsforbrug på de enkelte aktiviteter. Et eksempel på et udtræk fra dette system er vist i Figur 19.

Time/Sag

Navn	Sag	Forbrug
Hans Thomsen	Kvalitetskontrol	41%
Hans Thomsen	Omstilling af maskiner	24%
Hans Thomsen	Rekvisition af råvarer	20%
Hans Thomsen	Undersøge om fejlene kan udbedres	15%
Klaus Jensen	Montage	25%
Klaus Jensen	Produktion	75%
Per Andersen	Montage	11%
Per Andersen	Omstilling af maskiner	39%
Per Andersen	Rekvisition af råvarer	17%
Per Andersen	Udbedre fejl	33%
		1 - 10

Figur 19: Rapportering af medarbejderes tidsforbrug på aktiviteter

- Indtil videre er der kun blevet beregnet ressource cost-driverne til ressourcer, der vedrører medarbejdere. Selvfølgelig skal andre cost-drivere findes. Nogle kan findes ved
- 15 hjælp af andre kildesystemer som for eksempel et Excel-ark, som indeholder data, der vedrører den enkelte ressource.

12.8 Aktivitets cost-driver

- Sammenhængen mellem aktiviteter og omkostningsobjekter skal fastslås ved hjælp af
- 20 målinger. Disse vil typisk blive udført i forbindelse med udarbejdelsen af ABC-systemet, samt når der sker ændringer i virksomhedens arbejdsmetoder. Der kan endvidere implementeres en række registreringsinstrumenter, som hjælper med at registrere hvilke aktiviteter, der vedrører de enkelte omkostningsobjekter. Her vil SAP Business One ikke have nogen værktøjer, som kan hjælpe, så dette skal også registreres i eksterne værktøjer
- 25 som for eksempel produktionsstyringssystemer og time/sag styringsmodulet.

Når en virksomhed planlægger produktionsgangen for et produkt, skal det fastlægges, hvilke afdelinger og arbejdsprocesser, produktet skal igennem. Dette er ofte i ERP systemer betegnet som routing. Da det må antages, at virksomheden kender afdelingernes kapacitet, vil det enten ved hjælp af direkte måling eller ved et estimat være muligt at beregne hvilken omkostning, aktiviteten har. På denne måde vil både direkte omkostninger såsom arbejds løn og indirekte omkostninger som maskinomkostninger, elforbrug med mere blive målt. En mulig måde at implementere dette i praksis vil være, hvis medarbejderen selv indtaster sit forbrug i systemet. Dermed vil disse data kunne beregne den faktiske omkostning, der skal tilskrives produktet. Dette er dog ikke tilgængeligt i SAP Business One og kan derfor ikke bruges i dette projekt. Derfor vil aktivitet cost-driverne i dette projekt blive specificeret i et Excel-ark, som kan benyttes af ETL værktøjet.

12.9 Kontering tabellen

Kontering tabellen indeholder alle konteringerne. Alle disse er tilgængelige i JDT1 tabellen i SAP Business One. Denne indeholder datoen, konto id og en kolonne for debet og kredit. Sammenhængen mellem en kontering og en ressource er specificeret i et Excel-ark. I SAP Business One gemmes tidspunkterne ikke for konteringerne, så derfor vil alle tidspunkter referere til en række i tidsdimensionen, der hedder "*Tidspunkt ikke tilgængelig*". Derudover opnås sammenhængen mellem produkt og kontering til kolonnen **produkt_id** fra et Excel-ark. Figur 20 viser, hvor resten af værdierne findes i SAP Business One.

Kolonne	Tabel	Kilde kolonne	Datatype	Længde
konto_id	JDT1	Account	NVarChar	15
dato_id	JDT1	RefDate	Datetime	8
beløb	JDT1	Debit & Credit	Numeric	9

Figur 20: Kilde fra SAP Business One til Kontering tabellen

13 Extract, transform og load

I dette afsnit vil de forskellige transformationer og genereringer, der skal bruges til at udfylde data staging og data marter, blive gennemgået. Som det primære ETL værktøj vil Kettle Spoon blive benyttet. Derudover vil nogle af dimensionsværdierne blive genereret i et Excel-ark.

13.1 Tid tabellen

Til at udfylde Tid tabellen benyttes et Excel-ark. Som det kan ses i Figur 21(a) på den følgende side indeholder kolonne A timer og kolonne B minutter. Disse kolonner indeholder altså alle minutter for alle timer i et enkelt døgn. Fremgangsmåden, der benyttes her, er at lade Excel generere alle de SQL-sætninger, der skal bruges til at udfylde tidsdimensionen. Der genereres en SQL-sætning for hver række i regnearket. Den bygges op ved hjælp af strengkonkatenering af indholdet fra flere celler. Kolonne D indeholder strengen `INSERT INTO tid (tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen)VALUES (tid_seq.nextval,.` Kolonne E indeholder timen fra Kolonne A og minuttet fra Kolonne B adskilt og efterfulgt af kommaer. Kolonne F indeholder kommandoen `to_date,` som fortæller Oracle hvordan en tekststreng skal konverteres til et tidspunkt. I databasen gemmes tidspunktet som en *date*, og indeholder udover tidspunktet også en dato. Men da datoen gemmes i en anden tabel, og der ikke findes mere passende datatyper til tidspunkter, er det nødvendigt at sætte en dato også. Da den ikke skal bruges til noget, vælges her 1. januar 2000. Efter datoen ses igen timer og minutter. Da det ønskes at angive tidspunkt på dagen (formiddag, eftermiddag, aften, nat) angives den i et andet ark, se Figur 21(b). I kolonne A er angivet timen på dagen, og i Kolonne B angives betegnelsen, der skal tilknyttes timen. I kolonne G i Figur 21(a) indsættes Excel kommandoen `lookup,` som benyttes til opslag i arket vist i Figur 21(b).

AFSNIT 13. EXTRACT, TRANSFORM OG LOAD

A	B	C	D	E	F	G
1	0	00	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,00, to_date('01/01/2000 0:00','dd/mm/yyyy hh24.m'),'nat)			
2	0	01	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,01, to_date('01/01/2000 0:01','dd/mm/yyyy hh24.m'),'nat)			
3	0	02	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,02, to_date('01/01/2000 0:02','dd/mm/yyyy hh24.m'),'nat)			
4	0	03	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,03, to_date('01/01/2000 0:03','dd/mm/yyyy hh24.m'),'nat)			
5	0	04	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,04, to_date('01/01/2000 0:04','dd/mm/yyyy hh24.m'),'nat)			
6	0	05	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,05, to_date('01/01/2000 0:05','dd/mm/yyyy hh24.m'),'nat)			
7	0	06	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,06, to_date('01/01/2000 0:06','dd/mm/yyyy hh24.m'),'nat)			
8	0	07	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,07, to_date('01/01/2000 0:07','dd/mm/yyyy hh24.m'),'nat)			
9	0	08	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,08, to_date('01/01/2000 0:08','dd/mm/yyyy hh24.m'),'nat)			
10	0	09	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,09, to_date('01/01/2000 0:09','dd/mm/yyyy hh24.m'),'nat)			
11	0	10	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,10, to_date('01/01/2000 0:10','dd/mm/yyyy hh24.m'),'nat)			
12	0	11	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,11, to_date('01/01/2000 0:11','dd/mm/yyyy hh24.m'),'nat)			
13	0	12	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,12, to_date('01/01/2000 0:12','dd/mm/yyyy hh24.m'),'nat)			
14	0	13	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,13, to_date('01/01/2000 0:13','dd/mm/yyyy hh24.m'),'nat)			
15	0	14	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,14, to_date('01/01/2000 0:14','dd/mm/yyyy hh24.m'),'nat)			
16	0	15	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,15, to_date('01/01/2000 0:15','dd/mm/yyyy hh24.m'),'nat)			
17	0	16	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,16, to_date('01/01/2000 0:16','dd/mm/yyyy hh24.m'),'nat)			
18	0	17	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,17, to_date('01/01/2000 0:17','dd/mm/yyyy hh24.m'),'nat)			
19	0	18	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,18, to_date('01/01/2000 0:18','dd/mm/yyyy hh24.m'),'nat)			
20	0	19	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,19, to_date('01/01/2000 0:19','dd/mm/yyyy hh24.m'),'nat)			
21	0	20	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,20, to_date('01/01/2000 0:20','dd/mm/yyyy hh24.m'),'nat)			
22	0	21	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,21, to_date('01/01/2000 0:21','dd/mm/yyyy hh24.m'),'nat)			
23	0	22	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,22, to_date('01/01/2000 0:22','dd/mm/yyyy hh24.m'),'nat)			
24	0	23	INSERT INTO tid(tid_id,tid_time,tid_minut,tidspunkt,tid_paa_dagen) VALUES (tid_seq.nextval,0,23, to_date('01/01/2000 0:23','dd/mm/yyyy hh24.m'),'nat)			

(a) Generering af tidspunkter i Excel

A	B
1	0 nat
2	1 nat
3	2 nat
4	3 nat
5	4 nat
6	5 nat
7	6 formiddag
8	7 formiddag
9	8 formiddag
10	9 formiddag
11	10 formiddag
12	11 formiddag
13	12 eftermiddag
14	13 eftermiddag
15	14 eftermiddag
16	15 eftermiddag
17	16 eftermiddag
18	17 eftermiddag
19	18 aften
20	19 aften
21	20 aften
22	21 aften
23	22 aften
24	23 aften

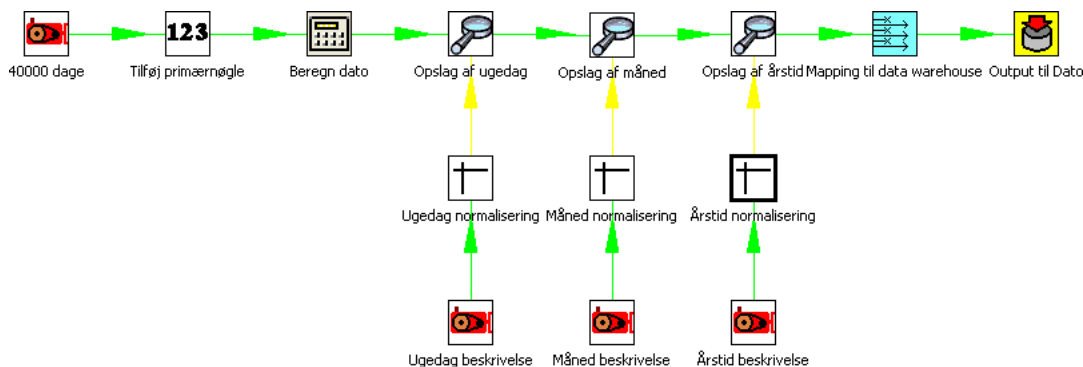
(b) Angivelse af tid på dag i Excel

Figur 21: Eksempler på tidsgenerering i Excel

13.2 Dato tabellen

Til at udfylde Dato tabellen benyttes en transformation i Kettle Spoon. Denne ses af Figur 22 Første trin, 40000 dage genererer 40.000 datoer, som dernæst får tilføjet en primærnøgle. Udover selve datoen ønskes blandt andet også gemt ugedagen, navnet og nummeret på måneden. I Kettle Spoon findes der funktioner, der ud fra en dato, kan beregne eksempelvis hvilket nummer i ugen, en dag har. Udover denne beregning sker der i trinnet Beregn dato en række andre beregninger på datoen, der skal bruges til senere opslag. I Ugedag beskrivelse defineres ugedagene, som tekststreng. I sammenligning med en database opfører dataene sig i dette trin, som værende en enkelt række. I trinnet Ugedag normalisering normaliseres den ene række fra Ugedag beskrivelse ud til syv rækker. Disse syv rækker bliver benyttet i Opslag af ugedag, hvor beregningen på ugedagen fra Beregn dato benyttes til opslag af den tekstuelle beskrivelse af ugedagen. Lignende sker for måned og årstider.

Til sidst foregår mapping fra feltnavnene i Kettle Spoon til kolonnenavnene i tabellerne i data staging, og der sker output til databasen.



Figur 22: Kettle Spoon transformation til Dato tabellen

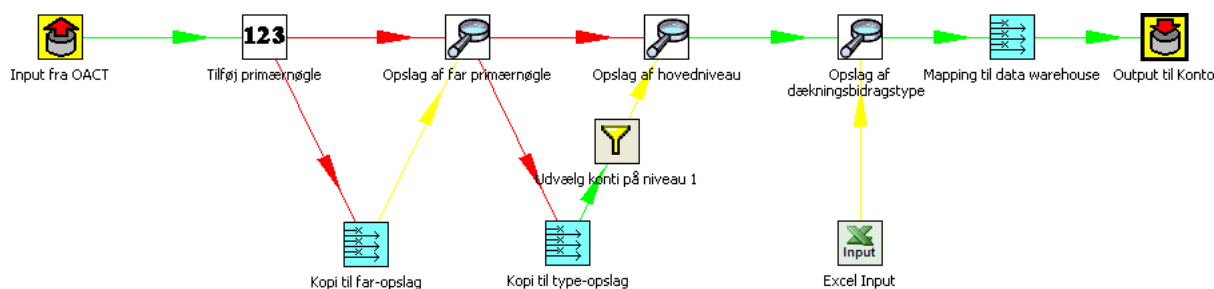
13.3 Konto tabellen

Konto tabellen baseres på OACT tabellen i SAP Business One. Processen er illustreret i Figur 23. Endvidere kan det ses i Figur 39 på side 97 i Bilag A på side 97, hvordan data bevæger sig mellem de forskellige trin. Første skridt i Kettle Spoon transformationen udvælger kolonnerne **AcctCode**, **AcctName**, **GroupMask**, **Levels** og **FatherNum**. Da der er et hierarki i OACT tabellen, skal indsættelsen foregå fra toppen af hierarkiet og nedefter for at sikre, at der ikke indsættes rækker, som forsøger at referere en konto, der endnu ikke er indsat. Kolonnen **Levels** angiver, hvilket niveau i hierarkiet kontoen ligger på. Inputtet fra tabellen sorteres derfor efter kolonnen **Levels**.

Derefter tilføjes en syntetisk primærnøgle. Det ønskes at bevare konto-hierarkiet i Konto tabellen i data staging. Da kontoerne får syntetiske primærnøgler, og dermed ikke kan benytte referencen til en kontos far, som den ligger i OACT, skal der ske en oversættelse af hierarki-informationen. Kontoen IT med kontonummeret 2700 har i kolonnen **FatherNum** værdien AA-II og refererer dermed til **Materielle anlægsaktiver**. Hierarkiet ønskes opbygget på primærnøglen i dimensionen, og værdien AA-II kan derfor ikke benyttes, da **Materielle anlægsaktiver** har fået tildelt en ny primærnøgle. Opslaget af den nye primærnøgle sker i **Opslag af far primærnøgle** baseret på dataene i **Kopi til far-opslag**.

Det ønskes i data staging, at hver række i konto-dimensionen indeholder navnet på det hovedniveau, den ligger under. Alle rækker i OACT indeholder en værdi i **GroupMask**. Dette gælder både konti på øverste niveau, og alle underliggende konti. På konti, der ligger på øverste niveau, er det en angivelse af hvilket kontonummernummer, dette hovedniveau har. For underliggende konti er det en fremmednøgle, som angiver hvilket hovedniveau, den pågældende konto hører under. Dermed skifter indholdet af **GroupMask** altså semantik, afhængigt af hvor i hierarkiet kontoen ligger. I **Opslag af hovedniveau** findes navnet på hovedniveauet. Grundet semantik problemet skal opslaget baseres på en filtreret del af kildedataene. Filtringen fjerner alle konti, som ikke ligger på øverste niveau ved hjælp af kolonnen **Levels**, som angiver hvor langt nede i hierarkiet en konto ligger.

Til sidst mappes feltnavnene i Kettle Spoon til de relevante kolonner i output-tabellen i data staging, og dataene skrives i databasen.



Figur 23: Kettle Spoon transformation til Konto tabellen

13.4 Produkt tabellen

Udfyldelse af **Produkt** tabellen er relativ simpel. Fra **OITM** tabellen hentes produkternes numre, navne og fremmednøglen **ItmsGrpCode**, som refererer tabellen **OITB**. Dette sker i trinnet **Input fra OITM**. Dernæst tilføjes en syntetisk primærnøgle. I trinnet **Opslag af produktgruppe** benyttes fremmednøglen til **OITB** for at finde strengbeskrivelsen af produktgruppen. **Definer prislister** definerer en konstant med værdien 1. I trinnet **Opslag af pris** sker der et opslag for at finde produktets pris, og her benyttes den netop definerede konstant. Konstanten benyttes, da der er flere forskellige prislister, og det ønskes at benytte prislisterne med værdien 1. Opslaget sker i tabellen **ITM1**, som har en fremmednøgle til **OITM**. De sidste to trin mapper Kettle Spoon feltnavne til kolonner i output-tabellen, og dataene skrives i databasen.



Figur 24: Kettle Spoon transformation til **Produkt** tabellen

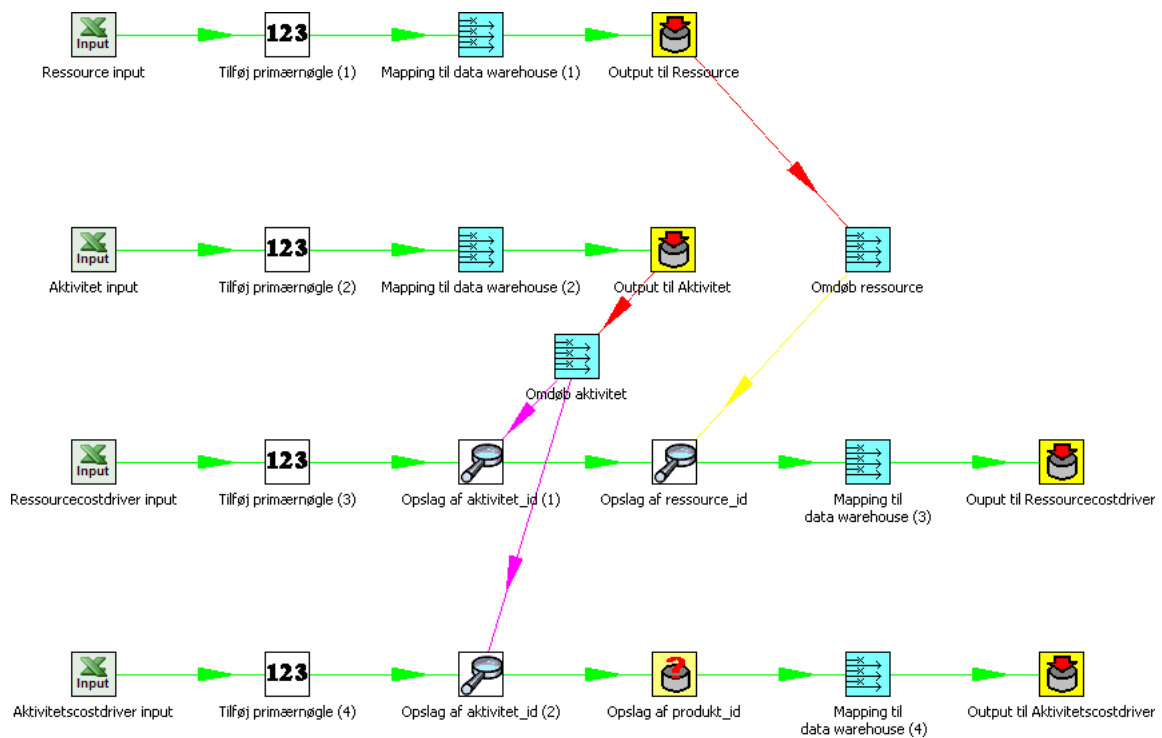
13.5 Ressource og Aktivitet tabellerne

Ressourcerne er gemt i et Excel regneark og transformationen til data staging foregår simpelt. Som det ses i Figur 25 på næste side sker der input fra regnearket og der tilføjes en syntetisk primærnøgle. Til sidst foregår der en mapping og data outputtes til data staging. Transformationen for aktivitetsdimensionen er tilsvarende simpel.

Ressource- og aktivitetscostdriverne er også gemt i Excel regneark. Men da de skal referere ressourcer, aktiviteter og produkter, som får tildelt syntetiske primærnøgler, skal der ske opslag af disse. De syntetiske nøgler i **Ressource** og **Aktivitet** tabellerne slås op ved hjælp af en “Stream lookup”, da dataene om disse allerede findes i denne Kettle transformation. Derimod skal der ske et opslag i data staging for at finde produktets syntetiske primærnøgle. Logisk set kræver det, at transformationen til **Produkt** tabellen allerede er udført, da opslaget ellers ikke kan returnere den korrekte værdi.

Umiddelbart vil et opslag i en database kræve færre systemressourcer end en stream lookup på klientmaskinen. Dette skyldes, at alle data i stream lookupen skal være tilgængelige for opslaget, hvorimod opslaget i databasen vil ske på en server, som kun returnerer den begrænsede mængde nødvendige data. Ifølge femprocentsreglen³⁸ vil der kun for eksempel optræde ressourcer, som udgør mere end fem procent af de samlede omkostninger. Logisk set vil der altså maksimalt kunne være 20 ressourcer. Denne datamængde er dog for lille til, at det vil være hensigtsmæssigt at lade en database håndtere opslaget. Eksempelvis vil der sandsynligvis være blandt andet et

³⁸[KC98], side 102

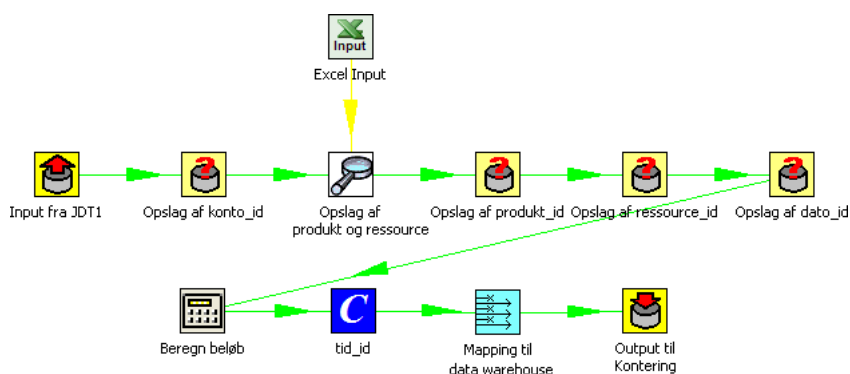


Figur 25: Kettle Spoon transformation til Ressource og Aktivitet tabellerne

vist netværksoverhead. Opslaget af **produkt_id** skal ske i databasen, da det er valgt at lade denne optræde i en anden Kettle Spoon transformation.

13.6 Kontering tabellen

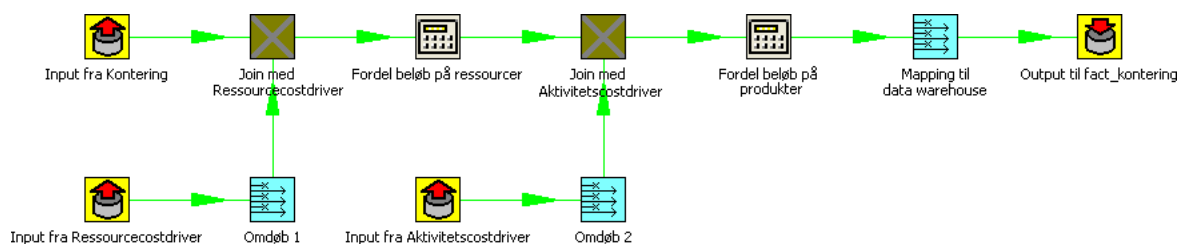
Kettle transformationen til Kontering ses i Figur 26 på den følgende side. Først sker der input fra JDT1. Dernæst sker der et opslag for at finde den syntetiske primærnøgle i konto-dimensionen. I næste trin sker der et opslag i et Excel regneark for at få information om, hvorvidt kontoen vedrører omkostninger, der er direkte eller skal fordeles. Dette benyttes i trinene Opslag af produkt_id og Opslag af ressource_id. Når en konto vedrører direkte omkostninger til et produkt, står der vedrørende ressourcen, at Denne kontering er en direkte omkostning i regnearket. Ligeledes er der i Ressource tabellen en række, som indeholder værdien Denne kontering er en direkte omkostning. Denne refereres derfor af Kontering. Datoen slås op i dato-dimensionen efterfulgt af en beregning på beløbet. Beregningen sker, da JDT1 både har en Debit og en Credit kolonne. For at finde det korrekte beløb, der skal gemmes i Beløb i Kontering, benyttes Debit – Credit. Da registreringer fra SAP Business ikke indeholder information om klokkeslæt, skal der refereres til en række i Tid, som indeholder værdien Ikke tilgængelig. Denne række er på forhånd fastsat til at have primærnøglen -1. Derfor defineres en konstant med navnet tid_id med værdien -1. Til sidst sker der mapping og output til Kontering.



Figur 26: Kettle Spoon transformation til Kontering tabellen

13.7 Data marter

For at udarbejde et data mart til rapportering af dækningsbidraget kræves der ikke det store arbejde i forhold til data staging. Konto- og datodimensionerne er de eneste, der er behov for i forbindelse med rapportering med dækningsbidrag. Dog skal der udføres et filter på Kontering tabellen, således denne kun indeholder konteringer på indtægter og omkostninger.



Figur 27: Kettle Spoon transformation til data mart til ABC

For at opbygge et data mart til ABC skal der hentes data fra tabellerne Ressourcecostdriver, Aktivitetscostdriver og Kontering. Kettle transformationen kan ses i Figur 27. Først sker der et input fra Kontering. Denne tabel joines med Ressourcecostdriver for at kunne beregne beløbet fordelt på aktiviteter. Der sker en omdøbning af kolonner for at øge overskueligheden i Kettle Spoon. Efterfølgende joines med tabellen Aktivitetscostdriver. Igen sker der en beregning. Denne finder beløbet fordelt på produkter, altså omkostningsobjekter. Afslutningsvis sker en mapping samt output til Fact_kontering.

14

Opsummering

I dette kapitel er der blevet udviklet et data warehouse på baggrund af SAP Business One. Fokus har været på at opnå en generel forståelse af data warehouse teknikker samtidigt med at udarbejde et funktionelt data warehouse.

Der er i rapporten blevet brugt Ralph Kimball og Margy Rosses [KR02] syn på data warehousing. Denne beskriver at alt relevant data først samles i et data staging område, hvorefter det relevante data i forhold til den aktuelle opgave bliver lagt i et data mart. Samlingen af de forskellige data marter danner således det færdige data warehouse.

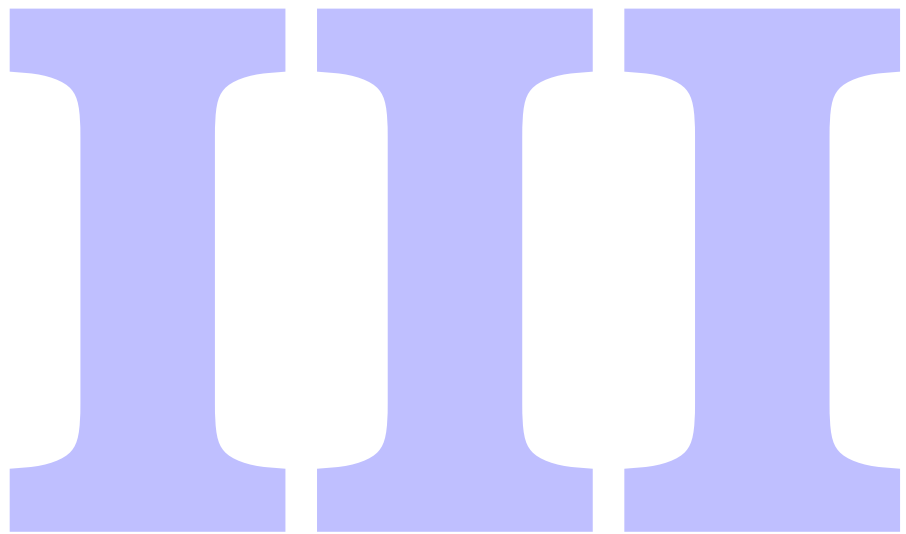
10 Det første, der blev udviklet, var en database, der skulle holde data stagingen. Denne indeholdt blandt andet drivere til ABC samt produktdata og kontooplysninger.

Data til data staging blev taget dels fra SAP Business One og dels fra andre kilder som for eksempel et Excel-ark. Data fra disse kilder skulle gennem forskellige transformationer inden de kunne loades ind i data staging. Dette blev gjort ved hjælp af Kettle Spoon.

15 Efter data staging blev udfyldt, blev de forskellige data marter oprettet, og data fra data staging blev kopieret over i data marter. Dette kunne kræve, at der eventuelt skulle laves endnu en transformation, hvilket var tilfældet ved data martet til ABC.

Alt i alt kan det konkluderes, at SAP Business One ikke alene kan levere data til det fuldt udbyggede data warehouse, der lever op til de kriterier, der er stillet i denne rapport.

20 Derfor er der blevet udarbejdet nogle Excel-ark, som indeholder data som SAP Business One ikke kunne levere. Men som en del-kilde er SAP Business One velfungerende.



Afrapportering

15

Fokusområde

I dette kapitel vil fokus være på afrapportering ud fra dækningsbidragsmodellen og ABC. Grundlaget for denne afrapportering er det, i Kapitel II, udarbejdede data warehouse.
5 Dette data warehouse bestod af to data marter, et til dækningsbidragsrapportering og ABC-rapportering.

Kapitlet indledes med en teoretisk gennemgang af de nye teorier og værktøjer, som vil blive benyttet i arbejdet. Dette skal danne grundlag for at opnå forståelse for disse. Efter at de forskellige teorier og værktøjer er blevet beskrevet, vil selve afrapporteringsværktøjet blive tilpasset, således at dette kan håndtere de forskellige data marter.
10 I denne rapport vil afrapporteringsværktøjet Palo blive benyttet, idet denne er fleksibel og afrapporteringen kan ske direkte i Excel, hvilket gør det muligt at udarbejde brugerdefinerede rapporter.

Samlet set skal dette kapitel være med til at skabe forståelse for, hvordan data fra et data
15 warehouse kan hentes ind i Palo værktøjet og efterfølgende afrapporteres i et Excel-ark.

16 Rapporterings- modeller

I dette afsnit vil teorien om OLAP blive gennemgået. Blandt andet vil de forskellige krav til et OLAP værktøj blive gennemgået og fordele og ulemper ved forskellige fremgangsmåder vil blive analyseret.

16.1 OLAP

OLAP er et akronym for *Online Analytical Processing*. Det primære formål med et OLAP værktøj er, at udføre komplekse multidimensionelle analyser af virksomhedens data for at opnå bedre forståelse. OLAP er typisk anvendt til at udføre rapportering af for eksempel salg og marketing. Ideen med OLAP er at organisere store mængder data, der indeholder mange forskellige faktorer og mulige udregninger således, at data bliver præsenteret på en rettidig, præcis og forståelig måde, som dækker de informationsbehov, som en slutbruger måtte have³⁹.

Ifølge [Tho02] stilles der følgende fire logiske krav til OLAP, for at denne kan benyttes til at udføre en given opgave⁴⁰:

Dimensionel struktur med hierarkiske referencer. Med en dimensionaliseret struktur opnås en yderst fleksibel struktur, som kan benyttes til mange formål.

Effektiv specifikation af dimensioner og udregninger. Udregninger i OLAP begrænser sig ikke kun til summeringer og gennemsnit i en given dimension. Ofte ønskes der analyser på tværs af de forskellige dimensioner. Dette kan for eksempel være på tværs af tid. Men hvis en virksomhed har joint-venture aktiviteter i

³⁹[Tho02], side 19

⁴⁰[Tho02], side 21

udlandet, bør tidsdimensionen kunne håndtere, at der i andre lande for eksempel benyttes andre regnskabsperioder. Andre former for data må nødvendigvis normaliseres, før de kan sammenlignes. Dette kan for eksempel være, at overskuddet kan være beregnet på forskellige måder i de forskellige dimensioner. Dette gør, at disse værdier skal normaliseres, før de kan sammenlignes.

Fleksibel. Det er vigtigt, at OLAP værktøjet er så fleksibelt som muligt. Derfor skal det være muligt for brugeren at kunne få de informationer, som ønskes, i form af grafer, skemaer og matricer.

Adskillelse mellem struktur og repræsentation. Sommetider ønsker en slutbruger at lave nogle strukturelle omrokeringer i forhold til data, der skal præsenteres. Dette skal gøres, uden at der sker ændringer i de originale data, hvilket for eksempel kan gøres ved at oprette nogle views. Det skal bemærkes, at det typisk ikke er slutbrugerens, som udarbejder disse views.

Hvis ikke et OLAP værktøj bliver benyttet, vil den mest udbredte måde at foretage analyser på virksomhedens data være at benytte sig af regneark eller SQL⁴¹. Gennem tiden har regneark fået flere og flere funktioner, som understøtter, at regnearket kan fungere som et afrapporteringsværktøj. For eksempel har Microsoft i Excel regnearket indbygget *pivot tabeller*, som indeholder OLAP-lignende funktionalitet, som kan bruges til at lave avancerede rapporter. Fordelen ved at lave afrapportering i et regneark er, at det er fleksibelt at lave ekstra beregninger og opstille brugerdefinerede grafer. En af ulemperne med at udføre OLAP direkte i et regneark er, at det er svært at opbevare en så dimensionaliseret struktur. Derfor vil det være svært at lave en klar adskillelse mellem struktur og repræsentation, som er en af kravene beskrevet tidligere. Dog er der nogle OLAP værktøjer, som benytter sig af avanceret funktionalitet, som findes i de enkelte regneark. For eksempel har OLAP værktøjet Palo et medfølgende add-on, som gør, at Palo data kan importeres i Excel, således at regnearkets funktionalitet kan benyttes med de importerede data. Dette vil blive beskrevet yderligere i Afsnit 17 på side 78.

Benyttes kun SQL til afrapporteringen, opnås en stor fleksibilitet, idet data kan sammensættes på hvilken som helst måde, der ønskes. Problemet med udelukkende at benytte SQL er, at dette kræver en IT-kyndig person til at foretage afrapporteringen, da det kræver en del viden om SQL for at foretage nogle komplekse afrapporteringer. En anden ulempe ved at benytte SQL til afrapporteringen er, at informationer kan være svære at overskue, hvis de blot er præsenteret i tal. Mange OLAP værktøjer er i stand til at præsentere informationerne i diagrammer og grafer, hvilket gør, at der hurtigt kan dannes et overblik over den givne forespørgsel. I OLAP værktøjer er det muligt hurtigt at tilgå de forskellige hierarkier, for at få mere detaljerede informationer. Dette er sværere i SQL, da en ny SQL-sætning skal udarbejdes for at lave den nye forespørgsel.

Figur 28 på næste side viser en oversigt over de forskellige analyser, som en virksomhed kunne ønske. OLAP er specielt brugbar i forbindelse med de beskrivende rapporter, hvor

⁴¹[Tho02], side 29

der kan aggregeres over forskellige dimensionsværdier på forskellige niveauer⁴². Hvis en af de tre andre beslutningstyper ønskes, skal der benyttes nogle mere omfattende data mining værktøjer, som ikke vil blive videre behandlet i denne rapport. Forskellen mellem OLAP og data mining er, at OLAP programmet ikke søger at forklare grunden, hvorfor tallene er, som de er.

Beskrivende	Forklarende	Forudsigende	Beslutningstagen
◊aggregering	◊regressions-	◊fremskrivning	◊budgettering
◊allokering	analyser	◊ledelse	◊simulationer
◊fordelinger	◊klynge analyser	◊trends	
◊måling	◊sammenhængs		
◊vægtning	analyse		
◊typetransformeringer	◊beslutningstræer		
	◊sandsynligheder		
	◊netværk		

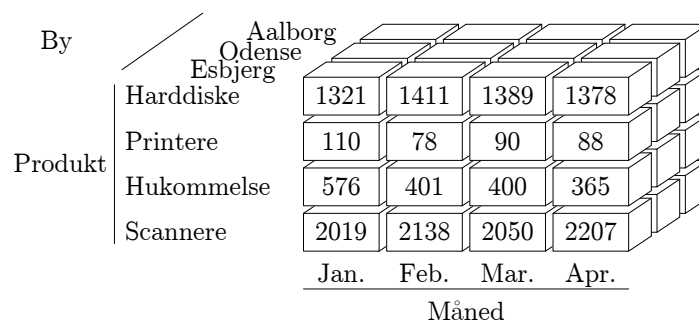
Figur 28: Beslutningstyper, frit efter [Tho02]

OLAP er en modifikation af den traditionelle *Online Transaction Processing* (OLTP). Forskellen mellem OLAP og OLTP er, at OLTP arbejder direkte på datakilderne, hvori-
 mod OLAP arbejder på en samling af data, der stammer fra forskellige OLTP databaser. Hvor OLAP søger at afhjælpe planlægning, beslutningsstøtte og historiske rapporteringer, er formålet med OLTP at styre dagligdags opgaver. Derfor arbejder OLTP typisk
 kun på aktuelle data, hvor OLAP arbejder på både historiske og aktuelle data. Grundet den større datamængde arbejder OLAP værktøjet ofte på et denormaliseret design med få tabeller; typisk et stjerneschema eller snowflakeskema. Derimod arbejder OLTP værktøjet på et højt normaliseret design med mange tabeller.

Kuber er et vigtigt begreb i forbindelse med OLAP. En kube består af et antal målinger, som er kategoriseret ud fra dimensioner. Kuberne er ofte dannet ud fra et stjerne- eller snowflakeskema fra et data warehouse. Teorien om disse skemaer er blevet beskrevet i Afsnit 9 på side 38. Fra data warehouset kan målingerne udledes fra fact-tabellen, og dimensionerne i kuben kan udledes fra dimensionerne data warehouset. Et eksempel på en kube kan ses i Figur 29 på næste side. Denne viser antal solgte varer. Kuben indeholder tre dimensioner, nemlig produkt, by og måned. Bemærk at en kube kan indeholde et vilkårligt antal dimensioner, men det er svært at illustrere mere end tre dimensioner.

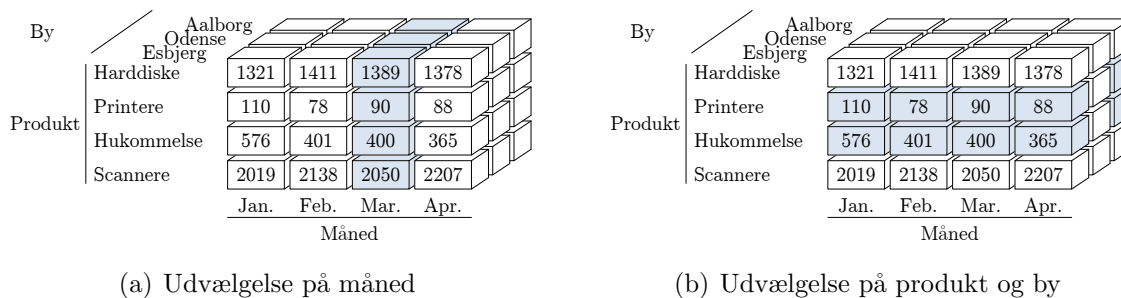
Når der skal laves analyser på baggrund af kuben, sker dette ved at lave forskellige udsnit af kuben. Ønskes for eksempel kun analyse på, hvad der er blevet solgt i marts måned, gøres dette ved at udvælge fra den del af kuben, der vedrører marts måned. Dette kan ses i Figur 30(a) på modstående side. Her viser de blå kasser, hvilken del af kuben, som forespørgslen vedrører. Det samme kan gøres, hvis der ønskes en analyse på baggrund af udvælgelse fra flere dimensioner. For eksempel kunne en forespørgsel være, at der ønskes antal solgte varer i hver måned for produkterne og *Printer* og *Hukommelse*.

⁴²[Tho02], side 16



Figur 29: Eksempel på en kube

Derudover ønskes kun en analyse, der vedrører Esbjerg og Aalborg. Denne analyse er mere kompleks end den forrige og kræver også et større udsnit af kuben. Dette udsnit kan ses i Figur 30(b). Igen vises de dele af kuben, der vedrører analysen, med blåt. Her ses det, at udsnittet ikke nødvendigvis er en sammenhængende del af kuben, men forskellige dele af kuben, der bliver udvalgt.

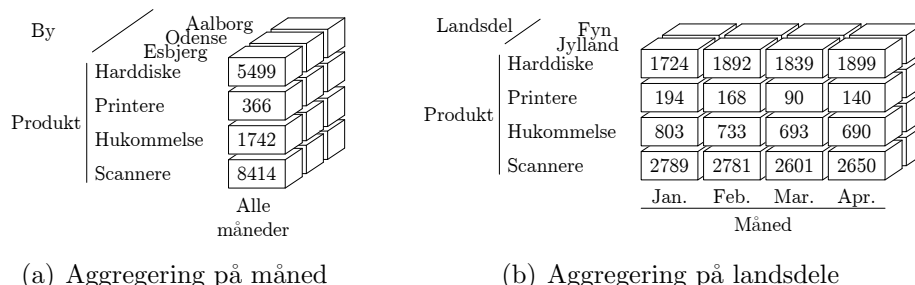


Figur 30: Udvælgelse i kuben

I forbindelse med OLAP, er aggregering den mest anvendte funktion. Aggregering sker på baggrund af fact-tabellen ved at skifte granulariteten på de forskellige dimensioner, og dermed summe de forskellige facts op til den valgte granularitet på de forskellige dimensioner. Hvis der tages udgangspunkt i kuben fra Figur 29 kan der ske aggregering på forskellige måder. For eksempel kan det være, at der ikke ønskes at få fordelingen vist per måned, men kun på årsbasis. Dette gøres ved at aggregere over måned dimensionen, således at alle de solgte varer, i de forskellige måneder, vil blive summeret til et enkelt tal for hver af de andre dimensioner. Denne aggregering kan ses i Figur 31(a) på næste side, hvor alle månederne er blevet summeret til et enkelt tal for hver af de andre dimensioner. Det er dog ikke altid nødvendigt at aggregere alle dimensionsværdier til blot en enkelt værdi.

Det kan vælges at opbygge nogle naturlige hierarkier på de forskellige dimensioner. Eksempelvis kan By dimensionen deles op i forskellige niveauer. Virksomheden kan have butikker i flere lande og således ønske en afrapportering fordelt på lande. Derfor vil landene være et naturligt topniveau. Alle lande har en naturlig opdeling, som kan benyttes. Danmark kan deles op i de forskellige landsdele. På denne måde opstår der et

naturligt hierarki, som kan benyttes til forskellige niveauer af aggregering. Hierarkiet i By dimensionen vil se således ud: Land \Rightarrow Landsdel \Rightarrow By. En aggregering over landsdelene af kuben fra Figur 29 kan ses i 31(b).



Figur 31: Aggregering i kuben

16.2 Relational Online Analytical Processing

5 *Relational Online Analytical Processing* (ROLAP) er en OLAP rapportering på baggrund af en relationel database. ROLAP giver den største grad af fleksibilitet. Da data allerede er tilgængelig i en relationel database, synes ROLAP at være en fornuftig løsning.

10 ROLAP kan afrapportere efter et almindeligt stjerneschema og giver derfor ekstra kompleksitet. ROLAP er i stand til at skalere til store data mængder, idet denne arbejder direkte på den oprindelige database, og ikke holder data i hukommelsen. En anden fordel ved ROLAP er, at data kan blive tilgået ved hjælp af et SQL rapporteringsværktøj. Dette behøver ikke være et decideret OLAP værktøj.

15 Ulempen ved at benytte ROLAP er, at ROLAP er langsommere at forespørge i end MOLAP. Dette afhjælpes ofte ved hjælp af en ekstra *summeringstabel*, som indeholder summeret data fra udvalgte dimensioner. Dette kræver dog ekstra udviklingstid i forbindelse med implementering af ROLAP. Det er ligeledes muligt at implementere dette som et materialiseret view, hvilket dog kan resultere i langsommere opdaterings-hastighed.

20 16.3 Multidimensional Online Analytical Processing

Multidimensional Online Analytical Processing (MOLAP) er en afrapportering i OLAP-familien. I modsætning til ROLAP opbevares alle data i MOLAP aggregeret i en ekstern database. Det har den fordel at forespørgelser kan udføres hurtigere. Ulemperne er, at data skal hentes over i MOLAP værktøjet. Dette tilføjer kompleksitet, ikke mindst i

form af udvikling men også i forbindelse med afviklingen af ETL'en. Logisk set kan der med MOLAP ikke forespørges på alle data, da der anvendes en grovere granularitet, som skyldes aggregeringen.

Undersøgelsen [Pen03] har vist at ROLAP typisk anvendes på store datamængder. I 2003
5 viste undersøgelsen, at et gennemsnits ROLAP system var på 312 GB, det skal sammenlignes med 4 GB for MOLAP systemer. I takt med dimensionernes størrelse vokser MOLAP systemet eksponentielt, som skal ses i kontrast til ROLAP, hvor størrelsen primært afgøres af fact-tabellen.

I denne rapport er der valgt at benytte et MOLAP værktøj til afrapporteringen, da
10 den forbedrede eksekveringsstid prioriteres. Endvidere forventes det ikke, at størrelsen af systemet vil være stor, da der er tale om en stærkt begrænset antal dimensioner, og dimensionerne er relativt små.

17 Afrapporterings- værktøjer

I dette afsnit vil der være en teoretisk gennemgang af afrapporteringsværktøjet Palo.
5 Efterfølgende vil der være en sammenligning mellem Palo og programmet XL Reporter, som er et alternativ til Palo med afrapportering fra SAP Business One.

17.1 Palo

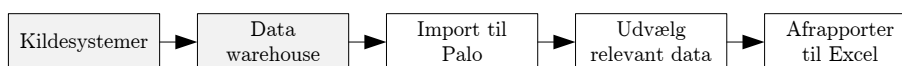
Palo er et MOLAP-værktøj, der kommer i to udgaver; som selvstændigt program og som add-on til Microsoft Excel. Som selvstændigt program vil afrapporteringen foregå
10 direkte i programmet ved hjælp af de indbyggede funktioner. Palo har funktioner til at oprette hierarkier på dimensioner, samt til at oprette kuber på baggrund af de forskellige hierarkier. Afrapporteringen sker på baggrund af de valgte dimensioner. Tallene opstilles i en matrice således, at den er mest overskuelig for slutbrugeren. Benyttes Palo sammen med Excel, vil der komme et nyt menupunkt i Excel, der indeholder Palo funk-
15 tionerne. Her kan dimensioner og kuber defineres på samme måde, som det blev gjort i det selvstændige program. Når afrapporteringen skal designes, kan udsnit af en kube indsættes i Excel cellerne. Også her kan matricen designes, som brugeren ønsker det, hvor der kan defineres, hvilke dimensioner, der skal vises i kolonner og rækker. Samtidigt kan der også defineres nogle dimensioner, der skal benyttes som sideelementer.
20 Disse udfører en filtrering i den viste matrice på dimensionerne, der ikke bruges direkte i afrapporteringen. Det er primært Palo gennem Excel, der vil blive benyttet i dette projekt.

Ifølge [Spr07] benytter Palo udelukkende hukommelsen i computeren til at gemme de data, der skal behandles. Dette gør det muligt at opnå hurtigere udførselshastighed
25 sammenlignet med data gemt på disk. Omvendt er Palo dermed begrænset af den tilgængelige mængde hukommelse på den aktuelle maskine. Dette synes dog ikke at være en

alvorlig begrænsning, da Palo arbejder med data, som er aggregerede til et vist niveau. Når data aggregeres vil datamængden dermed reduceres, da visse informationer forsvinder. I Palo startes der eksplicit en bagvedliggende server, som indeholder alle de data, der behandles. Uanset om der arbejdes i det selvstændige program, eller om add-on til
 5 Excel benyttes, vil begge arbejde med den bagvedliggende Palo server som grundlag. Derfor kan de udarbejdede dimensioner og kuber også benyttes i begge programmer, hvor dette ønskes.

Arbejdet med Palo i Microsoft Excel er en proces delt op i flere trin. Først skal der oprettes dimensioner og kuber. Enten kan dimensionerne oprettes ved at indtaste alle
 10 værdierne i dimensionen manuelt. Hvis dette gøres, skal hierarkistrukturen også oprettes manuelt, hvilket betyder, at der skal afgøres hvilke værdier, der skal være øvre niveau for andre værdier. Hvis dimensionen er af en vis størrelse, er det ikke længere praktisk muligt at indtaste dimensionsværdierne manuelt. Hvis afrapporteringen skal ske på baggrund af et data warehouse eller et data mart, kan Palo importere data fra en database ind
 15 i den bagvedliggende Palo server. Importen af data kan ske fra flere forskellige kilder, eksempelvis en Oracle database. På denne måde kan Palo selv opbygge dimensionerne på baggrund af de importerede dimensionsværdier. I forbindelse med importen kan der også angives, hvilke værdier der skal være øvre niveauer for andre værdier, og på den måde kan Palo selv bygge hierarkierne op. Dette er nyttigt, hvis dimensionerne er store
 20 som for eksempel tidsdimensionen.

Da rapporteringen fra Palo kan ske til Microsoft Excel, kan de indbyggede funktioner i Excel benyttes til at overskueliggøre tallene, som bliver beregnet af Palo. En oplagt mulighed er at lade Excel generere grafer, som eksempelvis kan vise udviklingen i et bestemt produkts salgstal eller indtjening.



Figur 32: Processen i Palo

25 Figur 32 viser, hvordan processen sker fra kildedata til selve afrapporteringen i Palo. Udvælgelsen af data sker fra kildesystemerne og bliver lagt i data warehouse. Dette blev beskrevet i Afsnit 13 på side 61 og er farvet gråt i figuren. Fra data warehouse bliver data importeret til Palo serveren, hvor de forskellige dimensioner bliver udfyldt, og alle måleresultaterne bliver fundet i fact-tabellen. Fra Palo serveren bliver de relevante data
 30 udvalgt ved hjælp af kuber, og til sidst bliver de sat ind i Excel.

Når der arbejdes med Palo som add-on til Excel, vil der blive tilføjet et række nye funktioner, der vedrører Palo, til funktionsbiblioteket. Disse kan bruges på lige fod med de funktioner, der er indbygget i Excel. Når der arbejdes med Palo, er der primært fire funktioner, som skal benyttes. Disse fire funktioner vil blive beskrevet herefter.

35 **PALO.DATAC** For at vise data fra en Palo kube, skal denne funktion bruges. Funktionen skal bruge adressen på serveren og navnet på den kube, hvori data skal slås

op. Herefter angives de dimensionsværdier, som skal bruges til opslag i den givne kube. Disse værdier kan enten indtastes direkte i formlen, eller som andre formler i Excel kan der blot indsættes referencer til andre celler i arket. Dette er nyttigt, når der er lavet en dynamisk rapport, hvori der kan vælges i hvilket niveau i hierarkiet i dimensionen, der skal afrapporteres.

```
=PALO.DATAC("Servernavn","Kubenavn","Dim1","Dim2")
```

PALO.ENAME For at vise data fra en dimension, skal denne funktion bruges. Igen skal servernavnet angives som første parameter. Derefter skal navnet på dimensionen angives. Som den sidste parameter angives positionen i den givne dimension. Funktion er brugbar i forbindelse med dynamiske rapporter, hvor `PALO.DATAC` felter kan referere til denne celle. Når der dobbeltklikkes på en celle, der indeholder en sådan formel, vil der komme en popup boks, hvori der kan vælges den enkelte værdi i hierarkiet.

```
=PALO.ENAME("Servernavn","dimension",pos)
```

PALO.EADD Denne funktion tilføjer en dimensionsværdi til en given dimension. Herved kan den bruges til at generere en dimension automatisk. Igen skal serveren og dimensionen være de første parametre. Derefter skal angives, om der må skrives til et konsolideret niveau. Som fjerde parameter skal der angives en dimensionsværdi. Derefter skal selve dimensionsværdien angives. I det næste felt skal angives, hvilken værdi der er forældre til det angivet element. Hvis der ikke ønskes hierarki på dimensionen, skal feltet efterlades tomt. Som det sidste felt, skal der angives, om dimensionen skal tømmes, inden der foretages nye indsættelser.

```
=PALO.EADD("Servernavn","dimension","n","Element","Forældre",FALSK)
```

PALO.SETDATA Denne funktion bruges til at indsætte værdier i en kube. Funktionen kan bruges til at udfylde en kube automatisk, for eksempel på baggrund af et data warehouse. I modsætning til de andre funktioner, skal servernavn og kube ikke angives som de første argumenter. Til gengæld skal værdien være det første argument. Derefter skal angives, om der skal skrives i en aggregeret værdi. Derefter skal serveren og kuben angives. Til sidst skal samtlige dimensionsværdier angives således, at der bliver angivet nøjagtigt, hvor i kuben, den givne værdi skal skrives.

```
=PALO.SETDATA("værdi",FALSK,"Servernavn","dimension","elem1","elem2")
```

17.2 XL Reporter

Som udvidelse til SAP Business One fås XL Reporter. XL Reporter benyttes til at afrapportere de data, der er lagret i SAP Business One. Afrapporteringen sker i Microsoft Excel, og kan derfor på samme måde som Palo benytte Excels funktionalitet.

Da XL Reporter integreres med SAP Business One er mulighederne for afrapportering begrænsede i forhold til Palo. Palo kan benytte flere forskellige kildesystemer, hvis dataene kan hentes fra en database eller en flad fil. Eksempelvis kan der ske registrering af medarbejderes tidsforbrug i et separat system, som gemmer data i en database. Palo
5 kan derefter hente disse data og benytte dem til afrapportering. Det er ikke muligt på samme måde med XL Reporter at bruge data gemt i andre systemer.

For at afrapportering kan ske, skal der hentes data fra kildesystemer. Palo gemmer disse data i sin egen database, hvorfor kildesystemerne ikke belastes yderligere. Derimod opbevarer XL Reporter ikke selv disse data, og kildesystemet, altså SAP Business One,
10 vil blive belastet hver gang, der afrapporteres. Til gengæld vil XL Reporter altid benytte de nyeste data, der findes i SAP Business One. I forbindelse med data warehousing er dette dog ikke så relevant, da der typisk skal bruges data fra længere perioder. Dermed har det seneste døgns nye data mindre betydning.

Dermed er processen ved brug af Palo mere omfattende, men tilbyder flere muligheder
15 med hensyn til datakilder. Da både Palo og XL Reporter afrapporterer til Excel, kan de begge drage nytte af regnearkets funktionalitet.

18

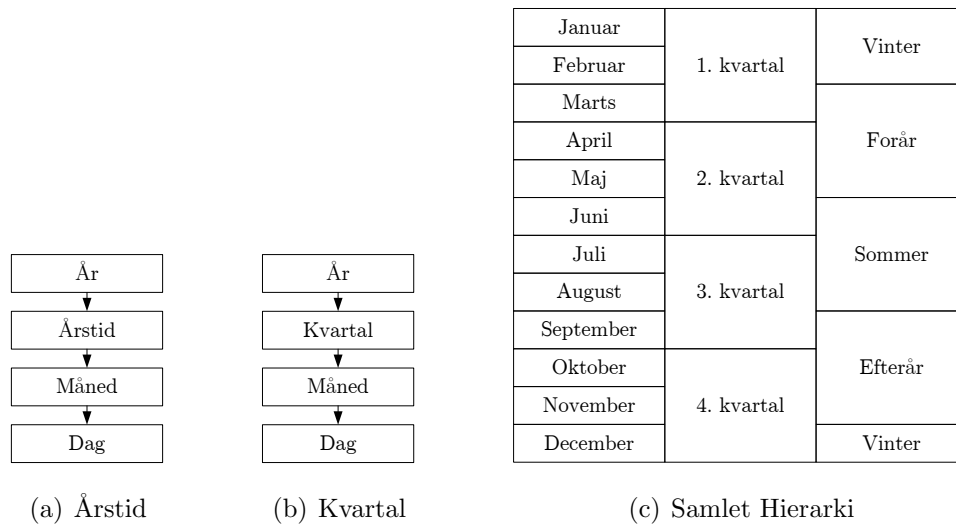
Import til Palo

Af rapporteringen i denne rapport vil ske ud fra de tidligere designede data marter ved hjælp af Palo. I dette afsnit beskrives, hvordan data fra data martet loades ind i Palo.

5 18.1 Datodimension

Da begge data marter bruger den samme datodimension, beskrives importen af denne konforme dimension til Palo først. Datodimensionen indeholder flere samtidige hierarkier, hvorfor dette skal håndteres specielt. Afhængig af hvilken form for afrapportering, der ønskes, skal der til Palo importeres forskellige data. Selvom datodimensionen indeholder mange forskellige oplysninger, behøver alle ikke nødvendigvis at blive importeret. I eksemplet her loades **dato**, **dato_dag_i_måned**, **dato_måned**, **dato_år**, **dato_kvartal** og **dato_årstid**. Da der ikke er direkte sammenhæng mellem kvartal og årstid er der tale om to samtidige hierarkier, hvilket er illustreret i Figur 33(a) på næste side samt Figur 33(b). Eksempelvis kan det ikke afgøres, om der er tale om 2. kvartal i et år, hvis den eneste tilgængelige oplysning fortæller at årstiden er sommer. I det tilfælde kan der være tale om 2. eller 3. kvartal, illustreret i Figur 33(c).

Af hensyn til udførselshastigheden er kun årene 2005-2007 importeret. Dato hierarkiet kan ses delvist foldet ud i Figur 34(b) på side 84. Figuren viser, at når der vælges et årstal, kan efterfølgende vælges en årstid eller et kvartal. Vinter og 1. kvartal er på figuren foldet ud, og dernæst optræder månederne. I forbindelse med årstiden vinter er semantikken vigtig. Som det ses, ligger vintermånederne fra det samme år under den samme årstid. Altså januar, februar og december fra 2006 er placeret under vinter, 2006. Selvom disse tre måneder altså ikke er sammenhængende. Dermed er årstiden vinter altså spredt over flere sæsoner. Dette er vigtigt at bemærke i forbindelse med økonomisk afrapportering. Omvendt vil det ikke være logisk at vælge 2006 på øverste niveau i hierarkiet, og længere nede lade januar 2007 optræde under vinter, 2006, selvom de tilhører samme sæson.



Figur 33: Sammenhæng mellem kvartal og årstid

```

1 SELECT TO_CHAR(DATO, 'DD-MON-YYYY'),
2         (DATO.MAANED || ', ' || DATO.AAR) DATO.MAANED, (DATO.KVARTAL || '. KVARTAL ' ||
3         DATO.AAR) DATOKVARTAL, DATO.AAR, (DATO.AARSTID || ', ' || DATO.AAR)
4 FROM DATO
5 WHERE DATO.AAR BETWEEN 2005 AND 2007
6 ORDER BY DATO

```

SQL-sætning 1: Udvælgelse af data fra data martet til Palo

I SQL-sætning 1 vises, hvordan relevante datokolonner fra datodimensionen udvælges. Endvidere sker der i SQL-sætningen konkatenering af flere kolonner, for at hierarkiet kan bygges korrekt op. Resultatet af SQL-sætningen kan ses i række 1 i Figur 34(a) på den følgende side. Kolonne A indeholder selve datoen. Kolonne B indeholder måneden repræsenteret som tal, efterfulgt af årstallet. I kolonne C ses det aktuelle kvartal. Årstallet ligger i kolonne D, og i kolonne E ses årstiden og årstallet. Årsagen til at årstid, kvartal og måned efterfølges af årstallet er, at Palo ellers ville tolke disse uhen-

sigtsmæssigt. Kolonne A og C i række 3-10 indeholder metadata. Kolonne B i rækkerne 3-10 indeholder Palo funktioner til indsættelse af data. Eksempelvis indeholder Celle B3 funktionen

```
=PALO.EADD("localhost/abcdatamart";"dato";"n";A1;B1;1;FALSK)
```

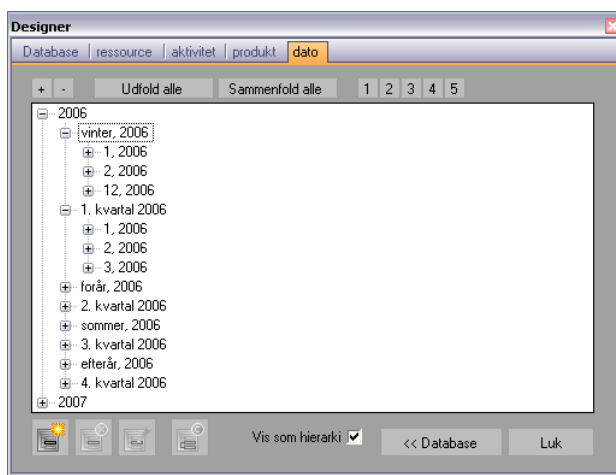
Funktionen tilføjer data til en dimension. Først er der angivet, på hvilken server og database, dataene skal gemmes. I dette tilfælde er det altså en database, som hedder `abcdatamart` på den maskine, der arbejdes på i øjeblikket. Dernæst angives hvilken dimension, der skal benyttes. Her er den aktuelle dimension navngivet `dato`. Efterfølgende angiver `n`, at der ikke skrives til et konsolideret niveau. Næste argument til funktionen er cellen `A1`, som indeholder den værdi, der skal indsættes i dimensionen. Da der her er tale om et hierarki, skal der også angives, hvor i hierarkiet den aktuelle værdi skal indsættes. I dette tilfælde indeholder cellen `B1` værdien på det overliggende niveau, som

A1 skal placeres under. Næstsidste parameter til funktionen er 1, der angiver vægten for den værdi, der skal indsættes. Yderligere beregninger kan naturligvis foretages ved hjælp af de gængse funktioner i Excel. Da den oprindelige værdi ønskes indsat i dimensionen, benyttes her vægten 1. Slutteligt angiver FALSK at de data, der i forvejen ligger i dimensionen, ikke skal slettes.

Grundlæggende udføres indsættelse af de andre data vedrørende datoen på samme måde som tidligere nævnt. Der er to samtidige hierarkier i datodimensionen, og derfor optræder hver måned i et hierarki med årstider og et andet med kvartaler. Endelig angives øverste niveau i hierarkiet til at være årstal.

	A	B	C	D	E
1	31-dec-2007	12, 2007	4. kvartal 2007	2007	vinter, 2007
2					
3	Formel	FALSK	Dato og måned		
4	Formel	FALSK	Måned og kvartal		
5	Formel	FALSK	Kvartal og årstid		
6					
7	Formel	FALSK	Måned og årstid		
8	Formel	FALSK	Årstid og år		
9					
10	Topniveau	FALSK	År		
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					

(a) Import af datoer til Palo



(b) Dato hierarki i Palo

Figur 34: Import af dato i Palo

I forbindelse med de forskellige kuber, der bliver oprettet i Palo, vil de forskellige dimensioner kunne genbruges på tværs af kuberne. Dette betyder, at dimensionerne er konforme, hvilket er et krav ifølge Kimball og Ross⁴³.

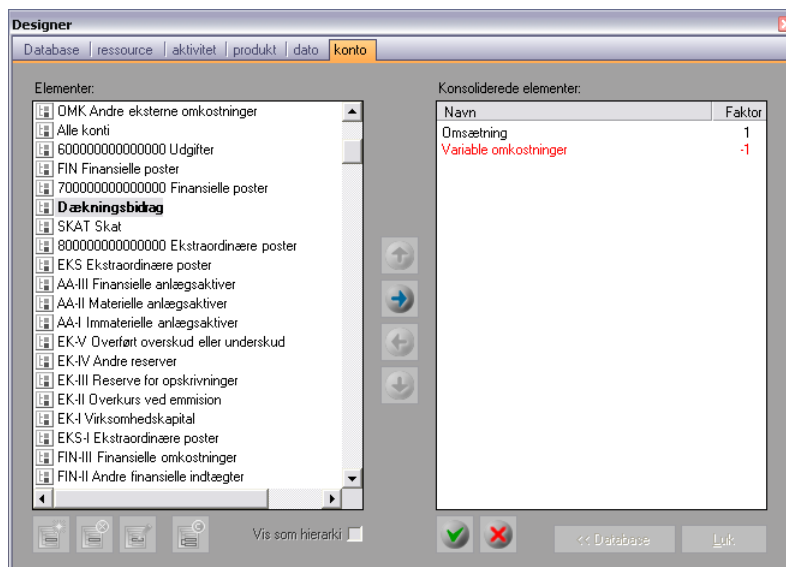
Det er vigtigt at bemærke, at data bør aggregeres med **GROUP BY** funktionen, før de indsættes i kuben. Dette sikrer, at der udføres færrest mulige opdateringer i kuben, da det er det aktuelle DBMS, som håndterer aggregeringsberegningen.

18.2 Dækningsbidrag

For at kunne rapportere ud fra dækningsbidragsmodellen skal data martet, som blev udviklet i Afsnit 11.3 på side 52, benyttes. Data fra dette data mart skal hentes ind i Palo for efterfølgende at kunne afrapportere i Excel. Dette afsnit vil omhandle import af dimensioner samt håndtering af facts i Palo kuben. Importen af kontodimensionen er simpel. Som illustreret ved importen af datodimensionen, som også bruges som dimension til ABC rapporteringen, indeholder også kontodimensionen et hierarki. Denne

⁴³[KR02], side 9

import foregår på samme måde som datodimensionen. Dækningsbidragskuben indeholder kun dato- og kontodimensionerne.



Figur 35: Konsolidering i dækningsbidrag

I forbindelse med udarbejdelse af en dækningsbidragsrapportering er der i kontodimensionen oprettet endnu et hierarki, som hedder **Dækningsbidrag**. Dette hierarki indeholder de to kontogrupper **Omsætning** og **Variable omkostninger**. Hierarkierne indeholder alle de konti, der vedrører henholdsvis omsætningen og de variable omkostninger i SAP Business One. Opbygningen i Palo, kan ses af Figur 35. Her ses det også, at der står -1 ud for omkostningshierarkiet. Omkostningerne har en konsolideringsfaktor på -1 , hvilket betyder at omkostninger vil blive anset for negative tal i forhold til den samlede aggregering. Dette stemmer overens med dækningsbidragsmodellen, der siger, at dækningsbidraget er omsætningen minus variable omkostninger. Ved at benytte konsolideringsfaktoren kan dækningsbidraget findes simpelt ved blot at indsætte top niveauet af dette hierarki i rapporten. Derudfra vil dækningsbidraget fremkomme uden brug af nogen udregninger. Ud fra hierarkiet er det muligt at foretage *drill down* længere ned for at se, fra hvilke konti den enkelte omsætning eller omkostning kommer fra.

Importen af facts fra data martet til Palo serveren sker også igennem Excel. Her importeres alle facts fra fact-tabellen i data martet. Problemet er blot, at fact-tabellen i data martet kun indeholder referencer til værdierne i dimensionerne og ikke de rigtige værdier. Derfor skal fact-tabellen joines med alle de relevante dimensioner således, at værdierne fremgår, som de ser ud i den tilsvarende dimension i Palo.

18.3 Activity Based Costing

Som beskrevet i Afsnit 11.3 på side 52 bliver der opbygget et data mart til ABC afrapportering. Disse data importeres til Palo for efterfølgende at foretage afrapporteringen i Excel. Dette beskrives umiddelbart herefter. ABC rapporteringen benytter den overordnede datodimension nøjagtigt som beskrevet i Afsnit 18.1 på side 82. Fact-tabellen importeres på samme måde som i dækningsbidragskuben som beskrevet i Afsnit 18.2 på side 84.

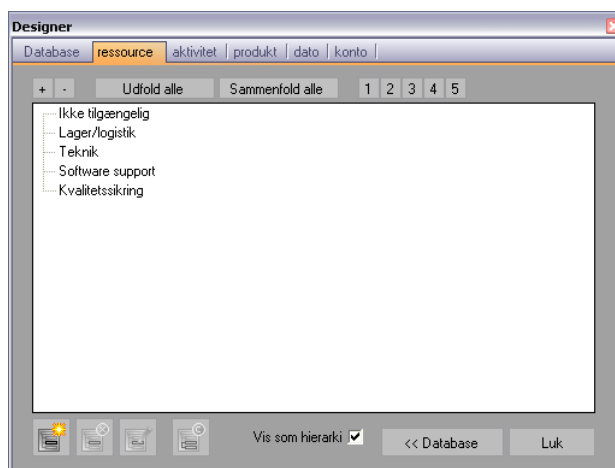
Indsættelsen af disse tre dimensioner foregår efter samme fremgangsmåde. I Ressource og Aktivitet er der ingen hierarkier og indsættelsen foregår ganske simpelt. I Figur 36(a) ses et skærmbillede fra Excel, hvor der skal indsættes data i ressourcedimensionen. I celle B1 står navnet på den aktuelle ressource, hvilket er det, der ønskes indsat i Palo. I celle B4 står følgende funktion.

```
=PALO.EADD("localhost/abcdatamart";"Ressource";"n";B1;;1;FALSK)
```

Da der i dette tilfælde ikke arbejdes med ressourcer, som ligger i et hierarki, er den primære forskel fra beskrivelsen af dato indsættelsen, at her er der ikke angivet et overliggende niveau. Dette kan ses i funktionen, hvor to semikolonner står umiddelbart i forlængelse af hinanden.

	A	B	C	D	E
1		8	Kvalitetssikring		
2					
3					
4	Formel	FALSK	Ressourcer		
5					
6			SELECT * FROM ressource		
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					

(a) Import af ressourcer til Palo



(b) Ressourcer som de optræder i Palo

Figur 36: Import af ressourcer

Celle C6 indeholder SQL-sætningen `SELECT * FROM ressource`. Da tabellen, der indeholder ressourcer, kun har et begrænset antal kolonner, importeres her blot alle kolonner. Ved tabeller, der indeholder mange kolonner, vil overskueligheden i Excel kunne øges, hvis kun en delmængde af kolonnerne udvælges i SQL-sætningen. Hvis kolonnerne alligevel ikke skal bruges til noget i data martet, kan der argumenteres for, at de er overflødige. Men kolonnerne, som ikke importeres til Palo, kan i stedet benyttes til afgrænsning af de data, som rent faktisk importeres.

Med hensyn til produktdimensionen indeholder den et hierarki. Indsættelsen af ele-

menter i et hierarki er beskrevet under udfyldelsen af datodimensionen.

Resultatet af importen af ressourcedimensionen kan ses i Figur 36(b) på forrige side.

Resten af dimensionerne bliver importeret på samme måde. ABC kuben indeholder ressource-, aktivitets-, produkt- og datodimensionerne.

19 Rapporter

Efter at data fra data warehouset er blevet overført til Palo, og kuber er blevet oprettet, som beskrevet i Afsnit 18 på side 82, vil der i dette afsnit blive fokuseret på den egentlige
5 afrapportering. Data warehouset vil bidrage med data til at kunne udarbejde ledelsesin-
formation og rapporter til virksomheden. Palo sammenkæder disse data således, at det
bliver muligt at forespørge på en række parametre. Afrapporteringen bliver, ligesom
Palo designet, udarbejdet i Microsoft Excel. Data vil dog kunne eksporteres til andre
programmer, som vil kunne visualisere disse. For at illustrere hvordan data warehouset
10 giver mulighed for at afrapportere, vil der blive givet eksempler på afrapporteringer
på baggrund af de designede kuber. Målet med projektet har været at kunne afrap-
portere efter dækningsbidragsmodellen samt ABC-modellen, og eksempler på disse vil
blive beskrevet herunder.

Idet der er valgt at benytte demonstrationsdatabasen i SAP Business One opstår der
15 her et problem, da denne ikke indeholder en ret stor mængde data. På baggrund af
den lille mængde data vil de genererede rapporter ikke blive særligt omfattende. Derfor
er der her valgt at autogenerere en mængde ekstra data via et eksternt program, som
bliver hentet ind i data warehouset. Således bliver der en større mængde af data at
afrapportere med. Det er her vigtigt at påpege, at strukturen fra SAP Business One er
20 bibeholdt, og det derfor lige så godt kunne være registreringer derfra.

19.1 Dækningsbidragsrapporter

Der vil i dette afsnit blive beskrevet rapporter baseret på dækningsbidragsmodellen. Der
vil under denne afrapporteringsform være en række parametre, som er interessante at
få beskrevet. De fleste af disse rapporter vil kunne udvikles ud fra den i Palo designede
25 kube. Således er det et spørgsmål om, at brugeren finder de korrekte tal og får dem
bearbejdet i Excel regnearket. Der er her valgt at illustrere, hvordan det er muligt at få
et samlet historisk overblik over virksomhedens indtægter og omkostninger samt få illu-
streret en samlet oversigt over, hvad der er posteret på virksomhedens omkostningskonti.

Dette er beskrevet i Afsnit 19.1.1 og Afsnit 19.1.2.

19.1.1 Dækningsbidrag

Det kan være relevant for virksomheden at få udarbejdet en oversigt over dækningsbidraget. Dette er illustreret i Figur 40, som er placeret i Bilag B.1 på side 98. Her er det lavet over årene 2005 – 2007 for at kunne give et historisk perspektiv. I figuren bliver dækningsbidraget først udregnet ved subtrahere virksomhedens variable omkostninger fra omsætningen. Dækningsbidragsprocenten bliver ligeledes udregnet. Idet Excel indeholder en række funktioner til at illustrere tal grafisk, er det muligt at konstruere grafer, som viser udviklingen. Da det er simpelt at vælge andre tal og konti i Palo, vil det være muligt at udarbejde grafer, som illustrerer udviklingen på andre konti.

19.1.2 Omkostningskonti

Det kan også være relevant for virksomheden at få en oversigt over de samlede omkostninger fordelt på konti. En sådan rapport er det muligt at generere ud fra data warehouset, og denne er illustreret i Figur 41 i Bilag B.2 på side 99. Generelt illustrerer rapporten, hvordan virksomhedens omkostninger er fordelt på konti, hvor det så er muligt at se saldo for hver konto gruppe. Det vil være muligt at udarbejde en række rapporter og analyser på baggrund af dette og udarbejde grafer af samme type som beskrevet i Afsnit 19.1.1. Nogle ERP systemer tilbyder denne form for afrapportering. Dog vil der ved hjælp af data warehouset kunne afrapporteres over en længere historisk periode, samt fra flere kilde-systemer.

19.2 ABC rapporter

Efter der er blevet afrapporteret efter dækningsbidragsmodellen, vil der i dette afsnit blive beskrevet, hvordan det ud fra data warehouset er muligt at rapportere ud fra ABC-modellen. Som det var tilfældet med dækningsbidragsrapporteringen er der en række problemstillinger, som kan være relevante at afrapportere i forhold til. Der er i dette afsnit valgt at give eksempler på tre afrapporteringer, der kan være relevante for en virksomhed, som benytter sig af ABC. Disse bliver beskrevet i Afsnit 19.2.1, Afsnit 19.2.2 samt Afsnit 19.2.3.

19.2.1 Omkostninger fordelt på aktiviteter

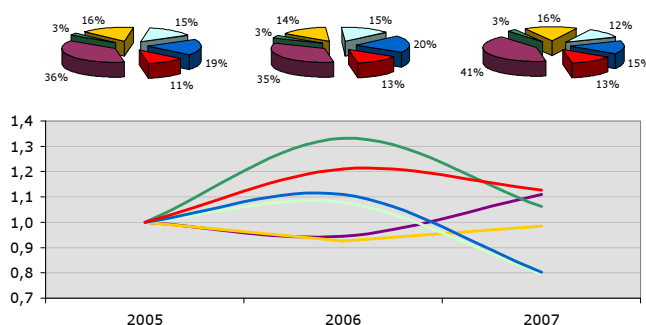
Den første afrapportering vil omhandle, hvordan virksomhedens omkostninger er fordelt på aktiviteterne. Dette er illustreret i Figur 37 på den følgende side. Det er vigtigt at pointere at her er de direkte omkostninger trukket ud, og figurerer derfor ikke i

afrapporteringen. Der er igen her valgt at illustrere resultatet som en historisk oversigt sådan at, det er muligt for virksomheden at se udviklingen over årene. Figuren viser, hvordan ressourceomkostninger er fordelt ned på hver enkelt aktivitet. Ligeledes er det muligt at få en total saldo over de fordelte omkostninger. Nederst på figuren er det igen vist, hvordan Excel kan medvirke til at illustrere tallene grafisk. Lagkagediagrammerne viser, hvordan omkostningerne fordeles på de forskellige aktiviteter. Kurvediagrammet viser udviklingen i indekstallene for omkostningerne for hver aktivitet.

Omkostninger pr. aktivitet 2005 - 2007

For 8. semester, Cand. Merc., udarbejdet 30/5-2007

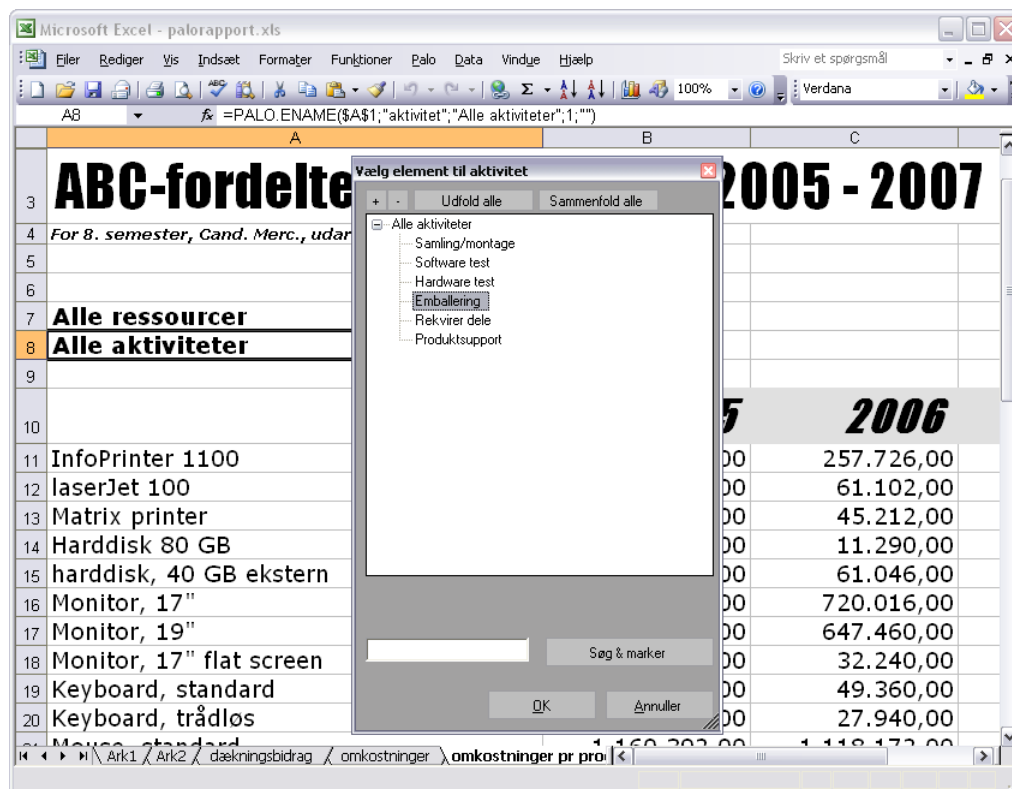
	2005	2006	2007
Alle aktiviteter	4.904.596,00	5.059.532,00	4.845.778,00
Samling/montage	1.815.632,00	1.716.020,00	2.016.292,00
Software test	123.668,00	164.762,00	131.360,00
Hardware test	786.268,00	728.074,00	774.358,00
Emballering	711.684,00	766.494,00	565.042,00
Rekvirer dele	911.318,00	1.010.828,00	731.782,00
Produktsupport	556.026,00	673.354,00	626.944,00



Figur 37: Omkostninger fordelt på aktiviteter

19.2.2 Omkostningsobjekters træk på aktiviteter

Hvis ABC fordelingen skal beskrives mere detaljeret end i Afsnit 19.2.1 på foregående side, kan det være interessant at få beskrevet omkostningsobjekternes træk på aktiviteterne. I dette projekt er produkter valgt som omkostningsobjekter i ABC, så det vil være muligt at se i hvilket omfang et produkt har trukket på en enkelt aktivitet eller ressource. Figur 38 på modstående side viser, hvordan det ved hjælp af en vælgerboks er muligt at vælge nøjagtigt hvilken aktivitet, virksomheden ønsker at se trækket for. Dette bliver vist på skærmen, hvor hvert enkelt produkt optræder på listen med sit beløb. Som det er tilfældet med de tidligere afrapporteringer, er der ligeledes valgt her at lave historiske data således, at udviklingen kan følges.



Figur 38: Omkostningsobjekters træk på aktiviteter

19.2.3 Omkostninger per omkostningsobjekt

En sidste rapportering om hvilke omkostninger, der er fordelt per produkt, kan ses i Figur 42 på side 100 i Bilag B.3 på side 100. Det kan ses i Figuren, at det er muligt at få en samlet oversigt over hvilke omkostninger, som er fordelt ned på det enkelte produkt.

- 5 Det er her værd at bemærke, at det designede system ikke opererer med styk antal, hvorfor det ikke er muligt at gå helt ned på enhedsniveau. Der kan argumenteres for, at denne afrapportering er næsten identisk med den, som er beskrevet i Afsnit 19.2.2 på forrige side. Der er dog valgt at tage denne med for at illustrere fleksibiliteten i opbygningen af systemet og hermed, hvor nemt det er at tilpasse afrapporteringerne til
- 10 netop den problemstilling, der er brug for.

I de foregående afsnit er der vist eksempler på, hvordan det er muligt at afrapportere i Palo og Excel på baggrund af data warehouse. Mulighederne her er store, og begrænsningen ligger hovedsageligt i den mængde data, virksomheden har valgt at opsamle i data warehouse.

20 Opsummering

I dette kapitel er der blevet udarbejdet afrapporteringer på baggrund af det udarbejdede data warehouse fra Kapitlet II. Her blev der oprettet data marter til både dækningsbidrag og ABC. Disse har dannet grundlag for en afrapportering af de respektive områder. Fokus har været på at opnå en generel forståelse af OLAP teknikker samtidigt med at udarbejde nogle fungerende afrapporteringer.

Der er i kapitlet blevet benyttet afrapporteringsværktøjet Palo, som fungerer som et add-on til Excel, således at Excels andre beregningsfunktioner kan benyttes med Palos egne funktioner.

Kapitlet er delt op i flere afsnit, der vedrører afrapporteringen med Palo. Først skal data fra data warehouse importeres til Palo. Dette gøres igennem Excel ved hjælp af Palos funktioner til at oprette dimensioner. Når dimensionerne er oprettet, kan en kube opbygges, og data fyldes ind i denne kube.

Når data er hentet ind i kuberne, bliver der udarbejdet nogle rapporter, der kan bruges til for eksempel at finde det samlede dækningsbidrag, finde de ABC fordelte omkostninger per produktserie eller de ABC-fordelte omkostninger per aktivitet.

Dette kapitel har været med til at give et indblik i, hvordan forskellige rapporter kan opstilles, således at der hurtigt kan skaffes et overblik over de forskellige omkostninger og indtægter.

21 Konklusion

Igennem denne rapport er det blevet forsøgt at afspejle processen med at udvikle et business intelligence værktøj til virksomheden. Dette er sket ud fra kildesystemer, som typisk er til rådighed i de fleste virksomheder i dag.

Ud fra den metodiske tilgang har det været vigtigt, at designet og implementeringen af data warehouse og afrapporteringerne har taget udgangspunkt i brugerens ønsker. Da der ikke har været inddraget eksterne samarbejdspartnere, er kravene blevet fastlagt ud fra den bearbejdede litteratur. Der er ligeledes blevet hentet inspiration til løsningen af problemstillingerne ud fra diskussioner med medstuderende og vejledere på studiet. Herved er der forsøgt at opnå forståelse for brugernes krav i forbindelse med de opstillede designkriterier og implementeringer.

Brugen af logik har hjulpet os til at vælge den bedste løsning i situationer, hvor der har været flere mulige løsninger. Løsningen er således valgt ud fra hvilke krav, der er til slutproduktet. Det vil sige, at der sagtens vil kunne vælges en anden løsning, hvis opgaven ændrer sig bare en smule. Semantikken har sikret en kontinuerlig diskussion om de valgte ord, hvilket har givet en mere præcis formulering, og dermed forbedret forståelsen.

Ud fra de opstillede modeller har der været lagt vægt på at få modellen til at passe ned over den praktiske IT mæssige implementering. Der er som tidligere nævnt blevet inddraget betragtninger om hvordan virksomhederne arbejder i praksis, men idet løsningen har skullet afspejle en teoretisk problemstilling, er vægten også blevet lagt på at den følge den økonomiske model i så høj grad som mulig.

Det er vigtigt at bemærke at de værktøjer, som bliver benyttet i denne rapport, ikke helt afspejler dem, som vil blive benyttet i virksomhederne. Når en virksomhed skal udarbejde en business intelligence løsning vil der typisk blive brugt en del ressourcer på at fremskaffe de rigtige værktøjer. Dette skal være med til at give et stabilt system, som kan levere resultater ud fra de krav som stilles. Værktøjerne i denne rapport har været dem, som har været stillet til rådighed af uddannelsesstedet, og afspejler i mange tilfælde ikke den driftssikkerhed, hastighed og kompleksitet, som kræves i erhvervslivet.

Dog fungerer værktøjerne acceptabelt til et projekt af det omfang, som dette projekt har krævet. Det er dog erfaret at hastigheden kan blive en begrænsende faktor under nogle af de mere avancerede transformationer.

5 Ud fra arbejdet med kildesystemerne er det erfaret at SAP Business One ikke alene kan levere data til det fuldt udbygget data warehouse, der lever op til de kriterier, der er stillet i denne rapport. Hermed kan det være vigtigt for virksomheden at integrere andre systemer i forbindelse med sit data warehouse. Dette stiller store krav i designfasen til data warehouset, og fordrer at de transformationer af data der skal ske er korrekte

10 Afrapporteringen vil i virksomheden kunne foretages med de værktøjer som er beskrevet i rapporten. Samarbejdet mellem regneark og Palo virker godt, og det vil være muligt at udarbejde afrapporteringer uden den store indsigt i den tekniske del. Rapporten har været med til at give indblik i, hvordan forskellige rapporter kan opstilles, således at der hurtigt kan skaffes et overblik over de forskellige omkostninger og indtægter.

15 Sammenlagt vil denne rapport give et billede af hvordan en proces fra data til business intelligence kan foregå i virksomheden. Den vil på mange områder være forsimplet i forhold til praktisk anvendelse. Forholdet til de økonomiske modeller er forsøgt bevaret så håndgribeligt som muligt.

Litteratur

- [AB97] Ingeman Arbnor og Björn Bjerke. *Methodology For Creating Business Knowledge*. SAGE Publications, 2. udgave, 1997.
- 5 [AR01] Michael Andersen og Carsten Rohde. *Virksomhedens økonomistyring*. Jurist- og Økonomforbundets Forlag, 2001.
- [BI03] Per Nikolaj Bukh og Poul Israelsen. *Aktivitetbaseret Økonomistyring*. Jurist- og Økonomforbundets Forlag, 2003.
- [BI04] Per Nikolaj Bukh og Poul Israelsen. *Activity Based Costing*. Jurist- og Økonomforbundets Forlag, 2004.
- 10 [CD97] Surajit Chaudhuri og Umeshwar Dayal. An overview of data warehousing and olap technology. *SIGMOD Rec.*, 26(1):65-74, 1997.
- [Con06] World Wide Web Consortium. Extensible markup language (xml) 1.0 (fourth edition). <http://www.w3.org/TR/REC-xml/>, 2006.
- 15 [Har96] Justus Hartnack. *logik - klassisk og moderne*. C. A. ReitZels Forlag, 1996.
- [Haz06] John Hazard. SAP's breakout: Vendor steps up efforts to capture market share. http://www.channelinsider.com/article/SAPs+Breakout+Vendor+Steps+Up+Efforts+to+Capture+Market+Share/170668_1.aspx, 2006.
- [Inm02] William H. Inmon. *Building the Data Warehouse*. Wiley Computer Publishing, 2002.
- 20 [KC98] Robert S. Kaplan og Robin Cooper. *Cost & Effect*. Harvard Business School Press, 1998.
- [Koc06] Christopher Koch. The ABCs of ERP. <http://www.cio.com/research/erp/edit/erpbasics.html>, 2006.
- 25 [KR02] Ralph Kimball og Margy Ross. *The Data Warehouse Toolkit*. Wiley Computer Publishing, 2002.

LITTERATUR

- [KRRT98] Ralph Kimball, Laura Reeves, Margy Ross og Warren Thornthwaite. *The Data Warehouse Lifecycle Toolkit*. Wiley Computer Publishing, 1998.
- [Kuh95] Thomas S. Kuhn. *Videnskabens Revolutioner*. Fremad, 1995.
- [LL03] Mark Levene og George Loizou. Why is the snowflake schema a good data warehouse design? *Information Systems*, 28(3):225-240, 2003.
- [Lyo95] John Lyons. *Linguistic semantics - an introduction*. Cambridge University Press, 1995.
- [Mad63] Vagn Madsen. *Regnskabsvæsenets opgaver og problemer—I ny belysning*. Gyldendal, 1963.
- 10 [Pen03] Nigel Pendse. *OLAP Survey*. Business Application Research Center, 3. udgave, 2003.
- [Rel06] SAP Investor Relations. Third quarter 2006 results. *SAP Investor, Magazine For Investors*, side 1-2, 2006.
- [SAP05a] SAP. SAP business one - en komplet virksomhedsløsning udviklet specielt til små og mellemstore virksomheder. www.sap.com/denmark/smb/businessone/brochures/SAP_whitepaper_2005.pdf, 2005.
- 15 [SAP05b] SAP. Sap business one, bedre lønsomhed, bedre kontrol. http://www.sap.com/denmark/smb/businessone/brochures/sap_business_one_solution_brief_2005.pdf, 2005.
- 20 [Spr07] Jedox Enterprise Spreadsheets. Jedox palo server. <http://www.jedox.com/en/enterprise-spreadsheet-server/excel-olap-server/palo-server.html>, 2007.
- [Tho02] Erik Thomsen. *OLAP Solutions – Building Multidimensional Information Systems*. Wiley Computer Publishing, 2002.
- 25 [Tör75] Håkan Törnebohm. *Paradigm i vetenskapernas värld och i vetenskapsteorin*. Avdelningen för Vetenskapsteori, Göteborg Universitet, 1975.
- [Wik07] Wikipedia. Information — wikipedia, the free encyclopedia, 2007. [Online; accessed 14-March-2007].
- 30 [Wor94] Zakken Worre. *Økonomisk Styling Af Virksomhedens Aktivitet*. Civiløkonomernes Forlag, 1994.

Kettle overblik

1 Input fra OACT

AcctCode	AcctName	Levels	FatherNum	GroupMask
1000000000000000	Aktiver	1		1
AA	Anlægsaktiver	2	1000000000000000	1
AA-II	Materielle anlægsaktiver	3	AA	1
2700	IT	4	AA-II	1

2 Tilføj primærnøgle

Konto_id	AcctCode	AcctName	Levels	FatherNum	GroupMask
1	1000000000000000	Aktiver	1		1
2	AA	Anlægsaktiver	2	1000000000000000	1
3	AA-II	Materielle anlægsaktiver	3	AA	1
4	2700	IT	4	AA-II	1

3 Opslag af far primærnøgle **3a**

Konto_id	AcctCode	AcctName	Levels	FatherNum	GroupMask	Konto_far
1	1000000000000000	Aktiver	1			1
2	AA	Anlægsaktiver	2	1000000000000000		1
3	AA-II	Materielle anlægsaktiver	3	AA		1
4	2700	IT	4	AA-II		1

4 Udvælg konti på niveau 1

Konto_id	AcctCode	AcctName	Levels	FatherNum	GroupMask	Konto_far
1	1000000000000000	Aktiver	1			1

5 Opslag af hovedniveau

Konto_id	AcctCode	AcctName	Levels	FatherNum	GroupMask	Konto_far	Hovedniveau
1	1000000000000000	Aktiver	1			1	Aktiver
2	AA	Anlægsaktiver	2	1000000000000000		1	1
3	AA-II	Materielle anlægsaktiver	3	AA		1	2
4	2700	IT	4	AA-II		1	3

Figur 39: Overblik over data bevægelse i Kettle Spoon transformation til konto-dimensionen

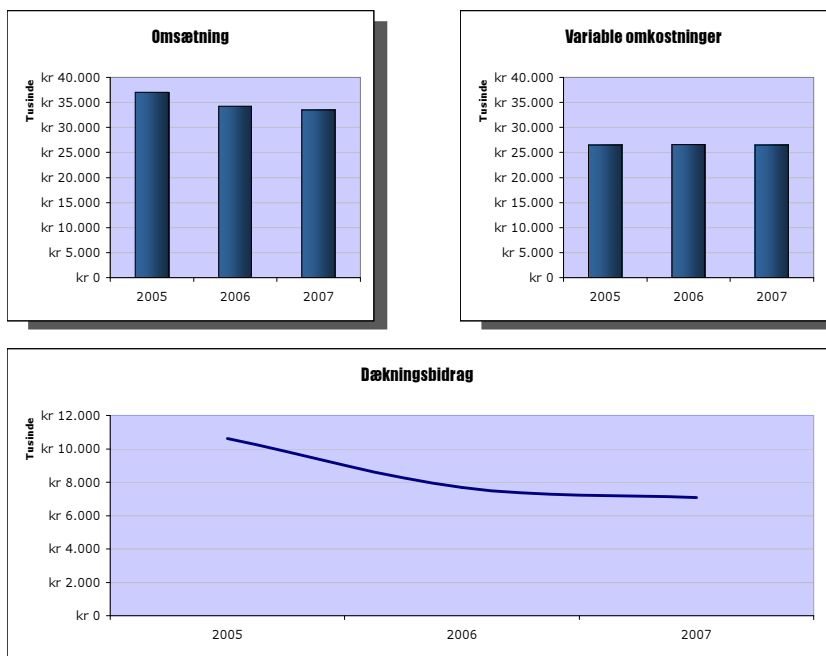
B Rapporter

B.1 Dækningsbidragsrapport

Dækningsbidrag 2005 - 2007

For 8. semester, Cand. Merc., udarbejdet 30/5-2007

	2005	2006	2007
Omsætning	37.059.120,00	34.224.824,00	33.515.384,00
Variable omkostninger	26.427.200,00	26.527.886,00	26.441.590,00
Dækningsbidrag	10.631.920,00	7.696.938,00	7.073.794,00
Dækningsbidragsprocent	28,69%	22,49%	21,11%



Figur 40: Dækningsbidragsrapport

B.2 Oversigt over omkostninger på konti

Omkostninger 2005 - 2007

For 8. semester, Cand. Merc., udarbejdet 30/5-2007

	2005	2006	2007
6000000000000000 Udgifter	31.887.538,00	32.314.640,00	32.359.544,00
OMK Andre eksterne omkostninger	31.887.538,00	32.314.640,00	32.359.544,00
OMK-XII Personaleomkostninger	10.423.794,00	10.413.994,00	11.162.106,00
OMK-AE Andre eksterne omkostninger	18.737.774,00	18.959.370,00	18.239.512,00
OMK-XI Øvrige omkostninger	3.320.824,00	2.543.980,00	3.464.242,00
61050 Tab på debitorer	111.544,00	120.062,00	539.396,00
61040 Øvrige omkostninger	711.580,00	315.860,00	1.160.440,00
61030 Forsikringer	850.788,00	508.010,00	544.642,00
61020 Fragtomkostninger	695.206,00	895.744,00	315.384,00
61010 Reg. hensættelser til tab på debitorer	376.968,00	239.956,00	252.868,00
61000 Tab ved salg af anlæg	574.738,00	464.348,00	551.512,00
OMK-X Bygningsdrift og lokaleomkostninger	2.418.352,00	3.194.334,00	4.166.278,00
60940 Rengøring og vask	447.326,00	651.876,00	975.426,00
60930 Husleje	521.494,00	409.954,00	947.760,00
60920 Vand, gas og fyringsolie	540.110,00	296.630,00	692.714,00
60910 Strøm (fast andel)	607.888,00	217.102,00	1.253.196,00
60900 Ejendomsskatter	301.534,00	1.618.772,00	297.182,00
OMK-VIII Kontordrift	1.464.474,00	1.981.278,00	1.080.936,00
60710 IT forbrugsmateriale	337.888,00	1.125.132,00	526.852,00
60700 Kontormateriale	1.126.586,00	856.146,00	554.084,00
OMK-VII Småanskaffelser	3.041.986,00	2.592.420,00	1.751.676,00
OMK-VI Kontingenter, abonnementer mv.	804.384,00	673.442,00	1.141.976,00
60510 Faglitteratur/abonnementer	590.834,00	547.366,00	452.480,00
60500 Kontingenter	213.550,00	126.076,00	689.496,00
OMK-V Kommunikation	1.194.044,00	1.085.968,00	1.128.010,00
60410 Telefon og øvrige kommunikationsmidler	510.090,00	531.130,00	480.006,00
60400 Forsendelse/porto	683.954,00	554.838,00	648.004,00
OMK-IX Revision og rådgivning	1.581.246,00	2.915.966,00	1.312.960,00
60820 Anden rådgivning	555.110,00	1.070.838,00	263.852,00
60810 Revision og regnskabsrådgivning	661.124,00	1.316.052,00	113.656,00
60800 Advokatbistand	365.012,00	529.076,00	935.452,00
OMK-IV Rejser	1.067.234,00	1.283.262,00	409.014,00
60300 Rejseomkostninger, ophold	1.067.234,00	1.283.262,00	409.014,00
OMK-III Repræsentation	235.834,00	270.088,00	1.121.642,00
60200 Repræsentation, gaver	235.834,00	270.088,00	1.121.642,00
OMK-II Markedsføring og information	1.652.408,00	1.061.986,00	1.200.674,00
60120 Reklameartikler og reklamegaver	556.154,00	162.592,00	168.718,00
60110 Kongresser, seminarer og udstillinger	558.796,00	588.240,00	239.482,00
60100 Annoncer, PR og markedsinfo.	537.458,00	311.154,00	792.474,00
OMK-I Autodrift	1.956.988,00	1.356.646,00	1.462.104,00
60020 Forsikringer og ejerafgift	639.454,00	809.452,00	608.978,00
60010 Reparationer	719.298,00	304.190,00	595.798,00
60000 Brændstof, olie, vask	598.236,00	243.004,00	257.328,00
AFSKR Af- og nedskr. af immat./mat. anlægsaktiver	2.725.970,00	2.941.276,00	2.957.926,00
65040 Ekstraordinær afskrivning på mat. anlæg	483.868,00	280.116,00	1.022.418,00
65030 Afskrivning på driftsmateriel og inventar	725.192,00	1.079.636,00	534.394,00
65020 Afskrivning af vognpark	328.172,00	722.158,00	257.986,00
65010 Afskrivning af tekniske anlæg og maskiner	651.444,00	321.290,00	725.610,00
65000 Afskrivning af bygninger	537.294,00	538.076,00	417.518,00

Figur 41: Rapport over omkostningskonti

B.3 ABC-fordelte omkostninger

ABC-fordelte omkostninger 2005 - 2007

For 8. semester, Cand. Merc., udarbejdet 30/5-2007

Alle ressourcer
Alle aktiviteter

	2005	2006	2007
InfoPrinter 1100	255.062,00	257.726,00	269.698,00
laserJet 100	53.104,00	61.102,00	49.698,00
Matrix printer	39.880,00	45.212,00	35.152,00
Harddisk 80 GB	9.880,00	11.290,00	8.986,00
harddisk, 40 GB ekstern	53.226,00	61.046,00	49.124,00
Monitor, 17"	653.474,00	720.016,00	540.304,00
Monitor, 19"	676.386,00	647.460,00	728.028,00
Monitor, 17" flat screen	28.086,00	32.240,00	26.018,00
Keyboard, standard	43.942,00	49.360,00	37.316,00
Keyboard, trådløs	24.884,00	27.940,00	21.054,00
Mouse, standard	1.160.392,00	1.118.172,00	1.281.278,00
Mouse, trådløs	48.942,00	56.260,00	45.600,00
Bundkort, BMD 300W	28.186,00	33.670,00	30.306,00
Bundkort, Inquel 300W incl. skærmkort	25.104,00	27.950,00	20.468,00
CPU, 3000 MhZ	47.584,00	53.250,00	39.686,00
CPU, 2800 MzH	42.936,00	48.594,00	37.532,00
CPU, 3500 MzH	4.456,00	4.980,00	3.716,00
Hukommelse, 512 MB	120.232,00	138.660,00	108.938,00
Hukommelse, 1024 MB ram	44.928,00	52.374,00	44.180,00
Hukommelse, 256 MB ram	799.788,00	757.070,00	776.284,00
Harddisk, 120 GB	58.874,00	67.750,00	55.058,00
Kabinet	33.172,00	37.728,00	29.652,00
Computer, Standard	48.676,00	55.018,00	42.370,00
Computer, PowerOffice	52.396,00	60.506,00	49.664,00
Support	41.292,00	47.900,00	39.818,00
konsulent time	27.042,00	32.400,00	29.368,00
Installation	39.228,00	45.052,00	36.388,00
Server Point 1000	31.026,00	37.074,00	33.408,00
Server Point Xtream 2	26.024,00	31.316,00	28.694,00
XPert, Standard	20.730,00	23.090,00	16.946,00
XPert, Proffesional	52.594,00	59.880,00	47.140,00
Office, standard	7.622,00	9.188,00	8.450,00
Office, Professional	15.106,00	17.480,00	14.454,00

Figur 42: ABC fordeling af omkostninger på produkter