SISU

PUBLIKATION 97:15

RAPPORT – JULI 1997

Data Warehouse

– en introduktion

Stig Berild

SVENSKA INSTITUTET FÖR SYSTEMUTVECKLING

SISU

Innehåll

1. INTRODUKTION 3

2. BAKGRUND, BEHOV 4

3. PRINCIPER, ARKITEKTUR 6

3.1 Inledning 6
3.2 Informationsanvändarperspektivet 7
3.3 Virtuellt Data Warehouse 8
3.4 Reellt Data Warehouse 11
3.5 Avgränsat DW 12
3.6 Data Mart 13
3.7 Arkitektur 16

4. HÄMTNING 17

4.1 Steg 17
4.2 Datakvalitet 19

5. LAGRING 21

5.1 Alternativ 21
5.2 Relationsdatabashanterare (RDBMS) 21
5.3 Anpassad RDBMS 22
5.4 Star schema; Princip 22
5.5 Snowflake Schema 25
5.6 Multidimensionell DBMS (MDDBMS, MDD) 26
5.7 Utsökningsoptimeringar 27

6. LEVERANS 30

7. PRESENTATION 31

7.1 Inledning 31
7.2 Tabellgränssnitt 31
7.3 Flerdimensionellt gränssnitt 32
7.4 DSS/EIS-tillämpningar 32
7.5 Data Mining; "Knowledge Discovery" 33
7.6 Affärssimulering 34
7.7 OLAP; betydelser 34

8. REPOSITORY; META DATA 36

8.1 Principer 36
8.2 Datakvalitet 37

9. ADMINISTRATION, UNDERHÅLL 39

10. MARKNAD 40

10.1 Produkter 40
10.2 Typiska tillämpningsområden 40
10.3 Diverse statistik 41

11. INFÖRANDE, RISKER 42

11.1 Införandestrategi 42
11.2 Diverse råd, synpunkter 43

12. AKTUELLA TRENDER 45

13. SAMMANFATTNING 46

MER INFORMATION 48

1. Introduktion

Rapporten syftar till att ge en allmän introduktion till området Data Warehouse. Materialet är hämtat från ett SISU-seminarium i ämnet. I vissa avsnitt "skiner ljusbilderna igenom" medan andra avsnitt ersatts med löpande text.

Det inledande avsnittet 2 sätter in Data Warehouse i ett behovsperspektiv. Därefter följer i avsnitt 3 en allmän genomgång av olika principer för att tillgodose behoven. Data Warehouse och Data Mart är i sammanhanget nyckelord. Avsnittet avslutas med en övergripande arkitekturskiss.

De olika komponenterna i arkitekturen diskuteras sedan i avsnitten 4 – 9.

Avsnitt 10 handlar om produkter och typiska tillämpningsområden. Data Warehouse är endast till liten del en samling produkter. Data Warehouse är, i utbyggt skick, en kontinuerlig process, en vital stödfunktionalitet som i olika grad når ut till de flesta områden i en verksamhet. Följdriktigt blir ett införande en både sammansatt och komplicerad aktivitet att ta på största allvar. Avsnitt 11 belyser några viktiga infallsvinklar. Avsnitt 12 tar upp några aktuella trender varefter avsnitt 13 avrundar det hela med några sammanfattande kommentarer.

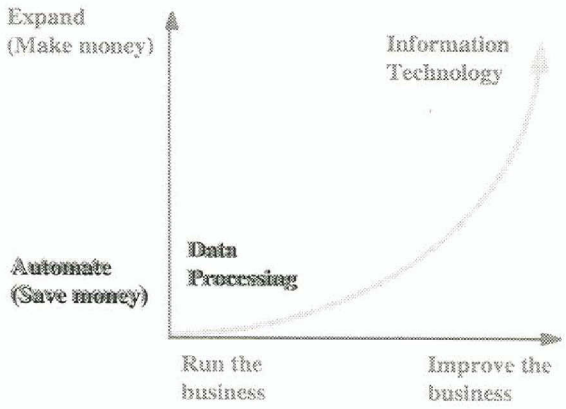
2. Bakgrund, Behov

Databaser har en lång historia och en viktig funktion i de flesta administrativa tillämpningar. Av tradition har databasen givit service till många behov inom viss tillämpning eller fix uppsättning tillämpningar. Förutsättningarna och behoven håller dock i snabb takt på att förändras:

- Datamassan växer
- Verksamheten blir mer
  - komplex
  - mångfacetterad
  - konkurrensutsatt
- Färre chefsnivåer, mer decentralisering
- Komplexare
  - datastruktur
  - semantik
- Data
  - spridd
  - svåråtkomlig
  - dåligt integrerad

Samtidigt ställer nya konkurrensförutsättningar, det påtagligt mer globala handlingsperspektivet nya krav på snabbare och bättre beslut.

Tidigare var inriktningen primärt koncentrerad mot att genom IT-satsningar rationalisera, automatisera, spara pengar genom tillämpningar som var inriktade på automatisering av tidigare manuella handgrepp. Målet var kostnadsbesparingar. Nu gäller det att utnyttja information och informationsteknologi till att fatta bättre beslut än konkurrenterna och därmed möjliggöra en expansion av verksamheten. Målet är ökade intäkter. Se figur 1.



Figur 1

Men goda beslut kräver förutom en klok beslutsfattare även information som är

- heltäckande
- väl utvald och förädlad
- snabbt tillgänglig
- överskådligt presenterad
- högkvalitativ

allt för att undvika att hamna i den ofta upplevda omständigheten "Drowning in Data – Dying for information".

Förhoppningar knyts nu till att Data Warehouse ska erbjuda denna smakliga "bakelse".

Liksom så många andra företeelser inom IT-området har även Data Warehouse genomlöpt en intressemässig berg- och dalbana under sin korta historia. Från ett buzzword för en trend eller ett framväxande behov, över massmedialt intensivt intresse med åtföljande mångfacetterade tolkningar. Via framväxten av mer eller mindre självutnämnda språkrör och gurus på världsturnéer och i författarskap, till ett spirande nyfiket kundintresse och förstagenerationens stapplande produkter. In i besvikelsens och misslyckandenas mörker, följt av konsolidering och förnuft, samt påbörjad tung men seriösare och mer rimlig väg tillbaka till en andra generationens realism i förståelse, ansats, produkter och genomförande.

Data Warehouse är mycket starkt på väg in i denna andra fas. Huvudsaklig drivkraft är inte längre massmedia och gurus utan de alldeles påtagliga bekymmer, behov och möjligheter begreppet satts att representera. Området kännetecknas också betydligt mer av pragmatiska ansatser baserade på sunt förnuft och konkreta, stegvis alltmer ambitiösa mål än på lustfyllda visioner och hägrande långtidsperspektiv.

Industriella trender som ökad konkurrens, kortare ledtider, snabbare reaktion på förändringar av marknadsindikatorer, mer krävande kunder, kortare produktlivscykler, internationellare konkurrensförutsättningar, pressar på behovet av att ha fullödiga, välserverade beslutsunderlag.

Alla budskap som svepande diskuterar det begynnande informationssamhället, om information som en resurs – ja kanske den viktigaste resursen och framgångsparametern i vitala verksamheter – skapar oro hos den oinsatte med åtföljande kunskapsjakt och snabba strategibeslut. De som hunnit längre behöver i sin tur veta mer om teknologi, trender, andras erfarenheter.

Informationstörstare i de flesta större verksamheter upplever å sin sida säkerligen regelmässigt alla tekniska, prestigebaserade, semantiska, säkerhetsmässiga, prestandamässiga m m hinder som omgärdar flexibel dataåtkomst både inom en existerande databastillämpning och i än större utsträckning över flera datakällor.

Är Data Warehouse lösningen? Många vågar sätta sin prestige på att så är fallet, att i alla händelser inga andra trender kan skönjas som ett bättre alternativ. Vad tolkas då just nu in i begreppet "Data Warehouse"?

Varje känd DW-"guru" formulerar givetvis sin egen uppfattning utifrån egna kunskaper, förutsättningar, erfarenheter. Nedan följer några alternativ:

Barry Devlin [7]

"A single, complete, and consistent store of data obtained from a variety of sources and made available to end users in a way they can understand and use in a business context."

Bill Inmon [1]

"A subject-oriented, integrated, non-volatile, timevariant collection of data organised to support management needs."

Sean Kelly [2]

"A single integrated store of corporate data which provides a unified infrastructural basis for applications to support decision making in the enterprise."

Ken Orr

"A facility to provide easy access to quality, integrated enterprise data by both non-professional and professional end users."

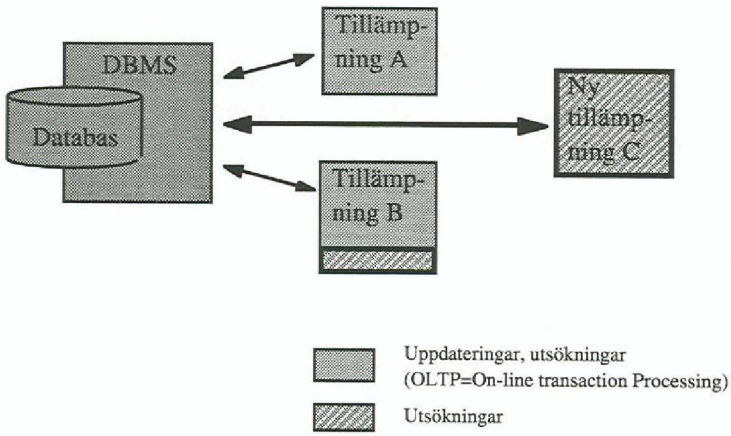
Det förtjänar alltså redan nu påpekas att ett DW minsann inte primärt är ett verktyg man köper och fyller med data varefter problemen låter sig lösas. Ett DW är en arkitektur och service. Det är en kontinuerlig process innebärande ett långsiktigt åtagande som fundamentalt ingriper i en verksamhets organisation, beslutsfattande, ansvar och informationshantering. Dessutom är den som sagt påtagligt mer intäkts- än kostnadsbesparingsmotiverad.

Vad kännetecknar då en typisk DW-arkitektur?

3. Principer, arkitektur

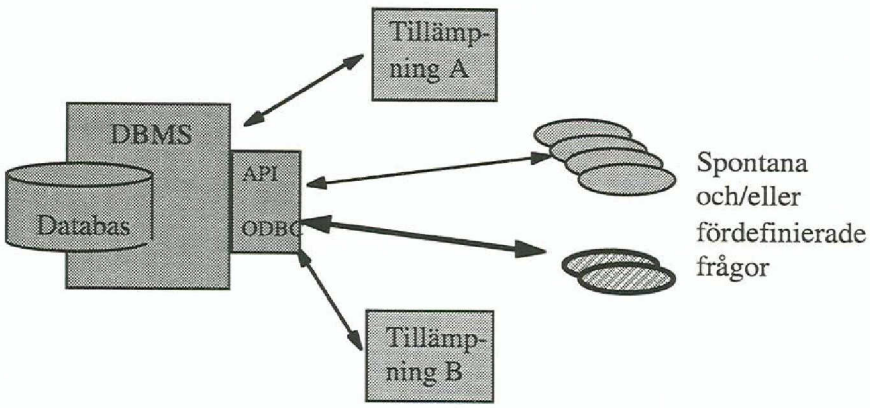
3.1 Inledning

Antag en existerande databasmiljö med ett antal tillämpningar som opererar på data i en databas hanterad av något databashanteringssystem (DBMS). Uppstår nya utsökningsbehov kan dessa realiseras genom tillägg till någon befintlig tillämpning, alternativt realiseras som en ny separat tillämpning. Detta oavsett om behovet är tillfälligt eller permanent. Figur 2.



Figur 2

Om DBMS erbjuder ett standardiserat frågegränssnitt, t ex relationsbaserade DBMS med SQL, kan även spontana frågor ställas mot databasen, under förutsättning att datamodellen är känd. SQL kan även utnyttjas för tillkommande permanenta behov. Nya informationsbehov kan på så sätt smidigt effektueras. Figur 3.

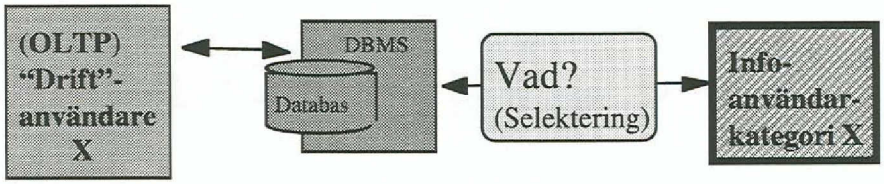


Figur 3

För det fortsatta resonemanget väljer vi att dela upp behoven mot databasen i två grupper. På ena sidan finns de ordinarie, operativa tillämpningarna och deras oftast permanent etablerade funktionalitet för både uppdateringar och utsökningar. De går vanligtvis under beteckningen OLTP (On-Line Transaction Processing).

På den andra sidan återfinner vi de mer spontant uppkomna frågeställningarna samt de informationsbehov av mer permanent art som inte har en naturlig koppling till någon viss OLTPs funktionalitet och syfte. Här kommer informationsarbetare, beslutsfattare, m fl in i

bilden. Deras informationsbehov styrs av någon aktuell roll inom t ex ett affärsområde, organisationsenhet, projekt eller ansvarsområde. Varje sådan användare tillhör därmed någon Informationsanvändarkategori.

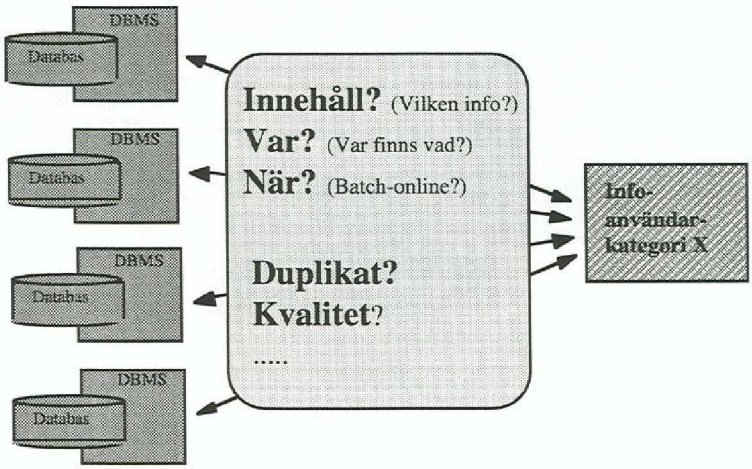


Figur 4

Vi antar fortsättningsvis att normala OLTP-användare tas om hand av sina respektive driftsmiljöer. Återstår att ge service till olika Informationsanvändarkategorier (infoanvändare).

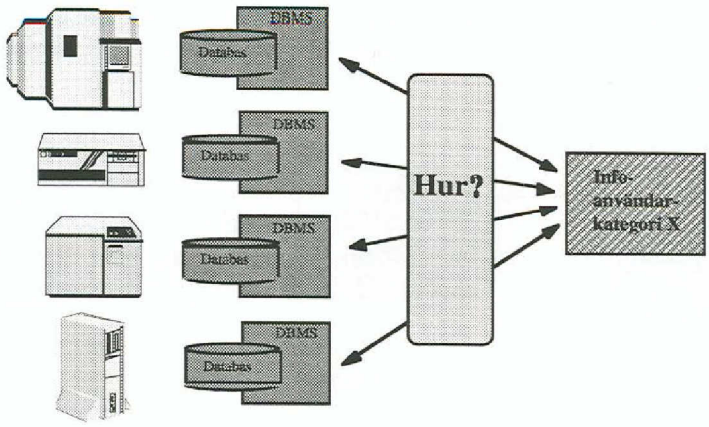
3.2 Informationsanvändarperspektivet

I viss mån kan denna kategori som nämnts ovan ges service i form av nya program och/eller ett generellt användargränssnitt – i allmänhet SQL. Men – infoanvändaren känner knappast någon bindning till en viss tillämpning eller databas. Denne behöver "sin uppsättning information" oavsett var den finns att hämta. Det är inte osannolikt att de uppgifter som svarar mot en viss fråga måste hämtas från ett antal källor och därefter sammanföras i ett svar. Nu börjar bekymren. Användaren måste känna till en hel del för att kunna ställa de rätta delfrågorna till de olika inblandade källorna, se figur 5. Dessutom måste personen ifråga vara skicklig på att sammanställa de olika delsvaren till det önskade totalsvaret.



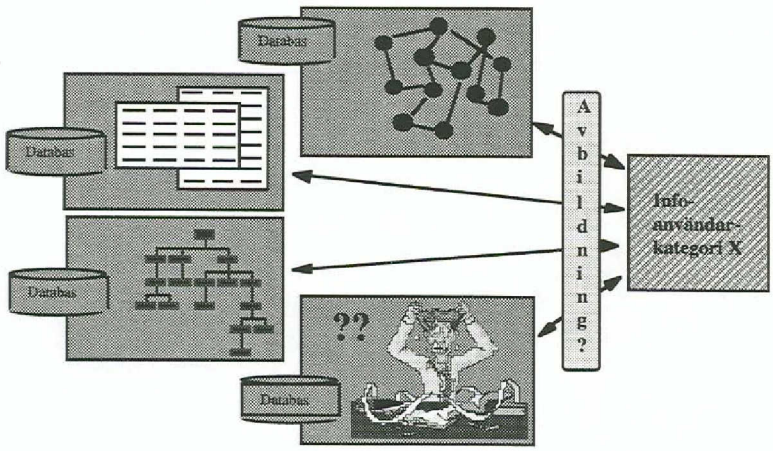
Figur 5

Inte blir det lättare om vi betänker att källorna kan finnas på olika datorkonfigurationer, under olika operativsystem, samt givet diverse andra kombinationer av tekniska förutsättningar. Figur 6.



Figur 6

Som extra salt på såret kommer så risken att databaserna skapats med hjälp av olika modelleringsteknik och därmed representerande olika användargränssnitt. I en större verksamhet finns sannolikt gamla system uppbyggda efter "eget huvud", hierarkiska databaser (IMS m fl), relationsdatabaser (visserligen alla SQL-baserade men var och en med sina nyanser att ta hänsyn till), kanske någon finurlig objektdatabas. Delsvar från olika databasmiljöer är strukturerade på olika sätt. Mer eller mindre avancerad teknik för avbildning till det enhetliga slutresultatformatet behövs. Figur 7.



Figur 7

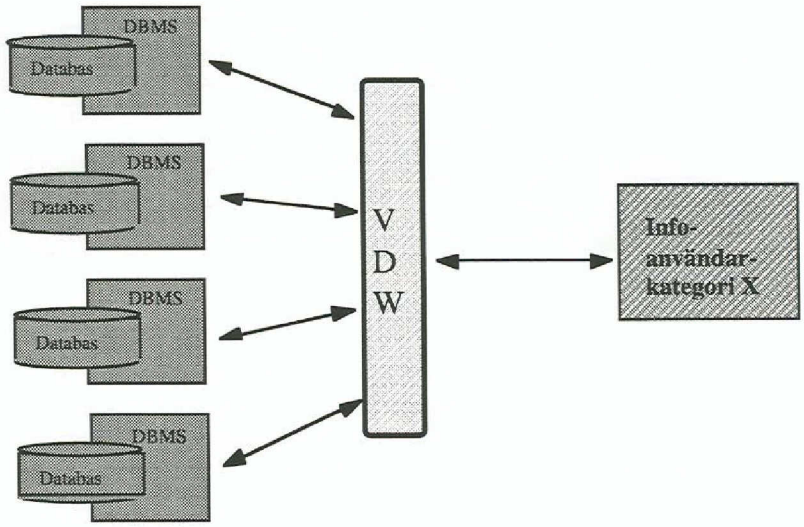
Som om inte detta vore nog gäller det förstås att kunna tolka de olika databasernas scheman (i den mån sådana finns för gamla system) både vad gäller syntaktisk uppbyggnad och, ännu mer problematiskt, vad gäller deras innebörd.

3.3 Virtuellt Data Warehouse

Ett sätt att ge infoanvändaren stöd i arbetet är att erbjuda en utsöknings- och sammanställningsservice, ett Virtuellt Data Warehouse (VDW). Gentemot användaren presenteras en generell datamodell som en integrerad bild av allt som rent fysiskt ligger i de bakomliggande databaserna. Dessutom erbjuds ett användargränssnitt baserat på den använda modelleringstekniken för VDW-modellen. Om relationsmodellen är bas blir gränssnittet naturligt SQL. Inte osannolikt föredrar emellertid användaren att nyttja en mer användartillvänd data- eller objektmodell med vidhängande gränssnittsspråk. Den ökade användarvänligheten får betalas med bristen på

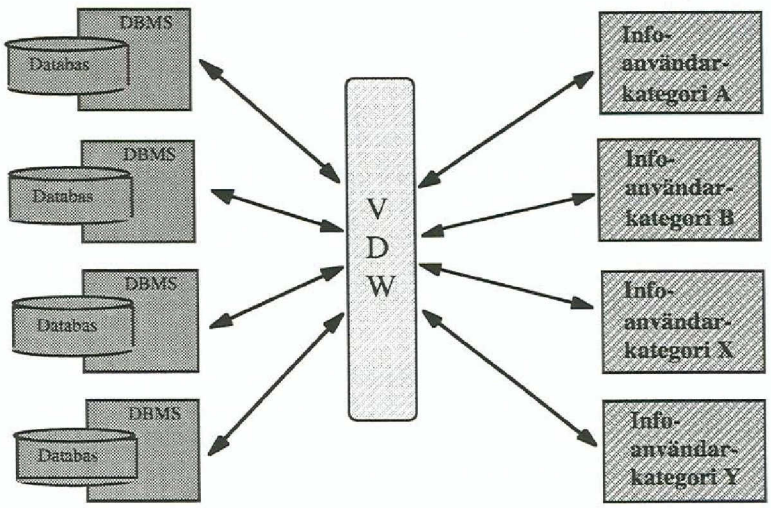
standarder för denna typ av språk. Oavsett gränssnittet mot användaren återstår ett digert arbete för VDW att

- tolka frågan
- bedöma vilka databaser som bör aktiveras för den fortsatta behandlingen (obs, inte bara för att hämta delsvar utan också för att förse eventuellt inblandade algoritmer och villkorskonstruktioner med "interna" data)
- dela upp i delfrågor samt översätta dessa till respektive gränssnitt
- anropa respektive databas
- invänta delsvar samt hantering av eventuella undantagstillstånd
- samt utföra de omvända stegen tillbaka till ett sammanställt slutsvar. Figur 8.



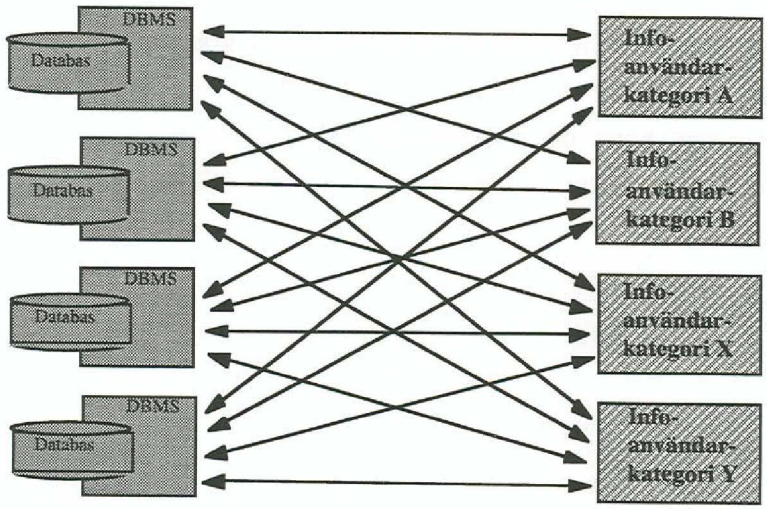
Figur 8

Bygger vi sedan på med många olika kategorier användare med många olika behov blir snart VDWs datamodell ganska komplex. Antalet databaser som kan behöva betjänas bakom ytan växer sannolikt också med tiden. Figur 9.



Figur 9

Plötsligt genereras nu en mängd ny information ur existerande databaser till olika användare för olika behov, utan samordning i tid och rum och utan annan koordination än vad som specificerats i de olika avbildningsprinciperna mellan databasmodellerna och den gemensamma VDW-modellen. Figur 10.



Figur 10

Kännetecknande för ett VDW är alltså att sökning sker i produktionsdata, vilket innebär att datakopiering undviks (något som är en grundprincip för ett konventionellt DW). Varje fråga innebär ett stort antal arbetssteg att genomlöpa. Prestanda kan snabbt bli en hämmande faktor. Användargränssnitten är specifika för respektive produkt.

Sett ur databasägarens synvinkel sprids data ur den egna databasen mer eller mindre okontrollerat ut till okända "behövande" via VDW. Säkerhets- och behörighetsaspekter måste säkerställas.

Under vissa välkontrollerade omständigheter, kännetecknade av väl avgränsade informationsbehov, god kännedom om källornas datamodeller, och infrekvent frågeverksamhet, kan ett VDW komma väl till pass. Produkter finns. Ibland går de under beteckningen Universal Data Access.

Men det finns också andra argument för att istället välja att etablera ett reellt DW med kopierade data snarare än access till produktionsdata. Argumenten står att hämta i ett antal påtagliga skillnader i syfte mellan vanliga OLTP-tillämpningar och DW-tillämpningar:

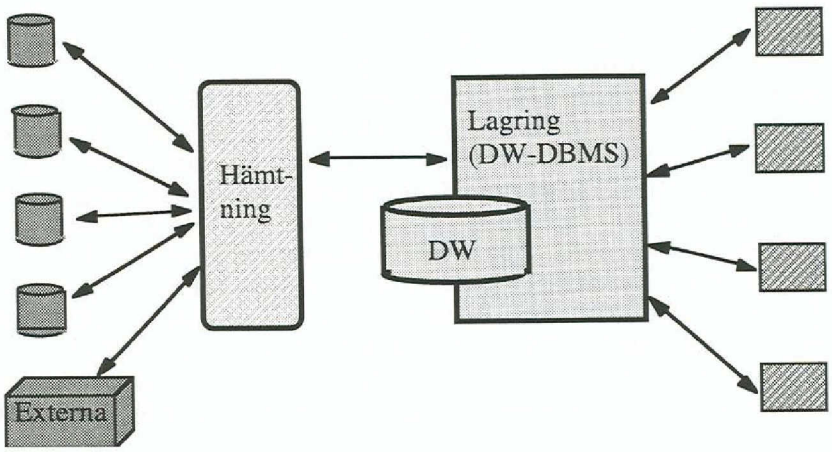
- OLTP är uppdateringsintensiva genom bearbetning av en strid ström, ofta korta transaktioner. Även interaktiva utsökningar är i allmänhet begränsade till sin komplexitet. Visserligen kan där genereras rapporter med omfattande datainnehåll men rapportens datastruktur svarar ofta väl mot databasens lagringsstruktur. Spontan frågeverksamhet är snarare undantag än regel. DW är nästan istället undantagslöst baserad på utsökningar. Frågeställningarna är inte sällan komplexa. Mer eller mindre omfattande beräkningar, summeringar, villkorskontroller, m m är vanliga. Spontan frågeverksamhet förekommer frekvent.
- Risk för inkonsistenser (anomalier) i samband med uppdateringar bidrar inom OLTP till en strävan mot fragmenterade datastrukturer enligt 3:e normalformen (även om välövertänkta avsteg ofta förekommer). DW kan tillåta sig betydligt flexiblare datastrukturlösningar, anpassade till bedömda utsökningsbehov.
- Inte sällan har OLTP att betjäna ett mycket stort antal samtidiga användare. DW med flera hundra användare är ännu sällsynta. Detta förhållande kommer säkerligen att förändras när väl DW blir etablerat som en integrerad del av den övergripande affärsverksamheten.

- Infoanvändare är inte enbart intresserade av egna interna data utan ofta även av data från externa källor. Tillgängligheten till dessa kan variera kraftigt alltifrån batchvis hämtning och leverans på band till direkt åtkomst. Endast det senare är acceptabelt i en VDW-miljö medan reellt DW erbjuder större flexibilitet.
- "Gamla data", d v s ett historikperspektiv över olika tidsperioder är ofta av intresse. Dessa återfinns sällan i operativa databaser.
- Eftersom strävan är att erbjuda beslutsfattare en rik uppsättning data att "ösa kunskap ur" blir följdriktigt en DW-databas mycket stor. Data samlas från många källor. Viss förpreparering av mer sammansatta data kräver utrymme, denormaliserade datastrukturer likaså. En DW-databas kan med andra ord bli flera magnituder större än en OLTP-databas.

Argumenten för att hantera beslutsstödsdata separerat från den konventionella driftens datahantering är med andra ord i de flesta fall slående. Över till reella Data Warehouse-arkitekturer.

3.4 Reellt Data Warehouse

Ett reellt Data Warehouse hanterar kopior av data hämtade från de operativa databaserna (i fortsättningen kallade källor eller källdatabaser). I samband med hämtning uppstår samma samordningsproblematik som för VDW. Väl på plats i ett reellt DW är data däremot tillgängligt och anpassat för infoanvändarens alla specifika behov.



Figur 11

Ett antal andra fördelar erhålls:

- Samlad datamassa ger en datamodell som reflekterar ett integrerat helhetsperspektiv. Förutsätter en av alla infoanvändare accepterad kompromiss kring datamodell, semantik i DW. Förutsätter därutöver kunskap om tillgänglighet, datamodell, semantik, m m hos respektive källdatabas för en korrekt avbildning till DW-databasen.
- Samtliga data för kunskapsbearbetning finns tillgängliga. Källdatabaser innehåller normalt endast aktuella uppgifter eller uppgifter över ett begränsat tidsintervall. Ett DW kan både innehålla dessa typer av uppgifter samt historiska data över ett valfritt tidsperspektiv. Historiska data är i de allra flesta fall en synnerligen vital informationskälla i DW-sammanhang. Genom dessa kan trendanalyser, volymuppskattningar, tillbakablickar, med mycket mera utföras. Hantering av olika historiska perspektiv kräver normalt någon form av tidstämpling av data. Värt att notera är att det historiska perspektivet också genererar nya problem. Dit hör dynamik kring innehåll och struktur i källdatabaser samt föränderliga informationsbehov och med dessa följande uppdateringar av hämtningsregler. Glapp och inkonsistenser i den historiska datamassan kan lätt bli följden – något som kan vara katastrofalt vid vissa typer av analyser.

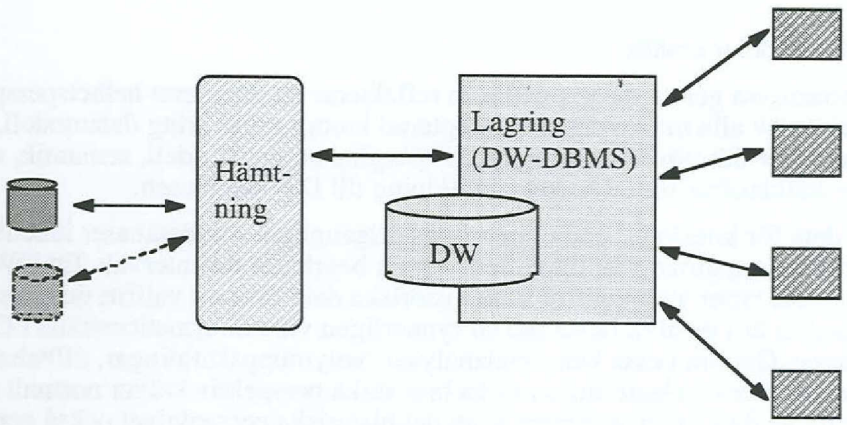
Data kan även specialanpassas för olika informationsbehov av prestanda- eller behörighetsskäl. Bland annat kan summeringar och andra beräkningar på olika aggregeringsnivåer finnas förkalkylerade och klara för utsökning.

- Data kan struktureras och optimeras för utsökningsändamål (oftast). Ger ett friare spelrum mellan olika normalformer där rdbms används och även en uppmuntran till helt nya datastruktureringsprinciper (se vidare under avsnitt 5).
- Källdatabaserna blir inte utsatta för "intrång". De ordinarie tillämpningarna behöver inte konkurrera med DW-tillämpningar. Behörighet, prestanda, dataansvar, m m behöver inte anpassas.
- Hämtningsmekanismen kompletteras normalt med kontroller, filtrering, m m vilket oftast resulterar i att data i DW har en högre kvalitet än källdata.
- Klar uppdelning i ansvar, administration, underhåll, tillgänglighet, optimering av såväl data som exekveringsmiljöer mellan källdatabaser och DW-databaser.

Priset för dessa fördelar består i administration av en extra databasmiljö samt att data inte har samma omedelbara aktualitet (hämtning görs vid vissa intervall) som operativa data. För mycket begränsade behov kan DW vara att "skjuta mygg med kanoner". I de allra flesta fall är dock reella DW att föredra framför virtuella – inte minst med tanke på att lyckad initial användning regelmässigt följs av betydligt mer expanderande önskemål.

3.5 Avgränsat DW

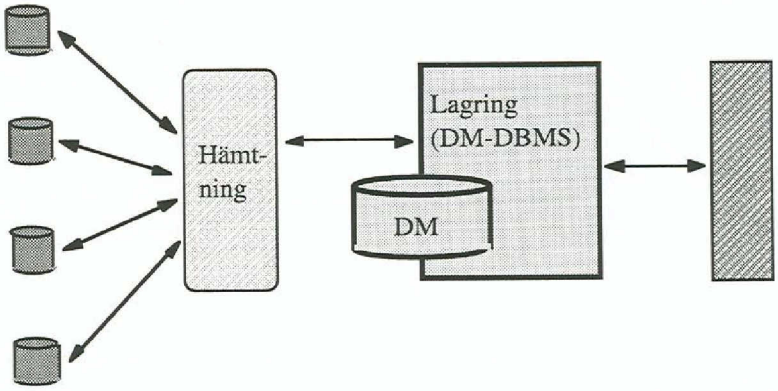
Allt detta låter ju praktiskt och vettigt. Men där finns hakar. En av dessa är samordningen av data från många källor för många behov. Att förstå de olika datamodellerna och rätt semantiskt värdera de ingående begreppens innebörd, att avgöra vilka överlappningar som kan finnas och vilka avbildningar till den önskade DW-datamodellen som erfordras är, åtminstone inom större organisationer, ett mycket tungt jobb. Dessutom ofta en källa till dispyter kring dataansvar, revirtänkande, semantik och kanske ointresse från de operativa enheternas sida. Misslyckade DW-satsningar hänvisar ofta till problem vid alltför ambitiösa – totalövergripande – datastrategier. En betydligt mindre konfliktutsatt strategi är att utgå från en enda eller ett par källdatabaser vars innehåll och datastruktur är väldokumenterad och känd. Samtliga bekymmer blir betydligt mer gripbara. Syftet kan till exempel inledningsvis endast vara att samla in data över längre tidsperioder i och för olika historiska analyser över lämpliga tidsdimensioner. Datamassan blir sannolikt mindre. DW-datastrukturen blir enklare vilket i sin tur kan öppna för hantering av mer förkalkylerade data. Sannolikt blir svarstidsprestanda bättre, vilket i sin tur ger utrymme för intensivare frågeverksamhet och kanske fler typer av frågor från fler kategorier infoanvändare.



Figur 12

3.6 Data Mart

Om vi i stället väljer att begränsa antalet infoanvändare, speciellt antalet olika typer av infoanvändare, blir läget som i figur 13.



Figur 13

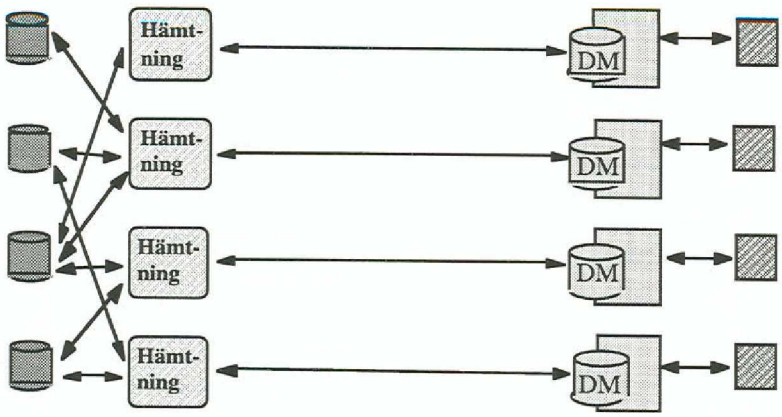
Sker avgränsningen exempelvis till endast en kategori användare (inom affärsområde, organisationsenhet, projekt, ...) är sannolikt informationsbehoven betydligt mer avgränsade, de semantiska valörerna på datamodellernas begrepp mer kända och accepterade, antalet aktuella källdatabaser färre. Kanske kan användargränssnittet specialanpassas för de specifika behoven, o s v. Återigen en miljö som är betydligt lättare att hantera än mer totala förutsättningar. Å andra sidan går man endast ett litet steg mot idén om en samordnad datamassa, om en verksamhetsgemensam resurs.

Strategin ter sig ändock lockande som ett första steg mot en integrerad DW-lösning. Riskerna och kostnaderna är betydligt mindre. Förmodligen svarar ansatsen mer påtagligt mot konkreta behov med följaktligen nöjdare användare. Ansatsen går under beteckningen Data Mart. Den har fått ett starkt marknadsgenomslag med specialanpassade produkter och ofta mycket positiva omdömen från användarhåll.

Kärt barn har många namn. Åtminstone tre namn olika tycks figurera i pressen. Den ungefärliga nyansskillnaden hänför sig till databasernas storlek:

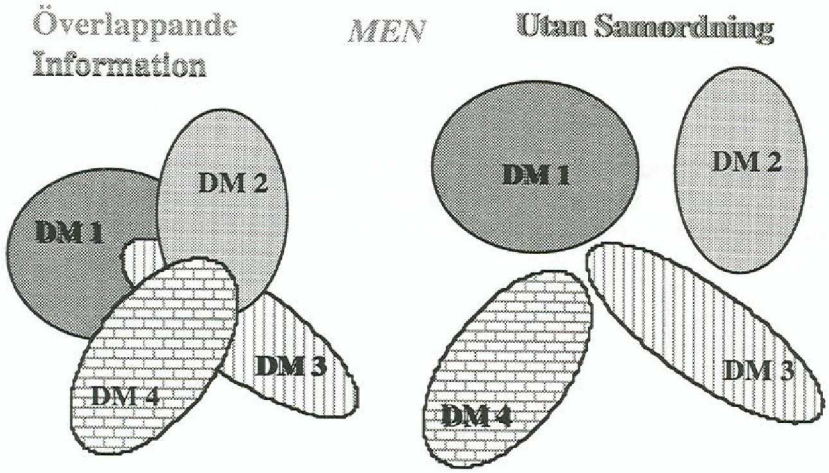
- Data Mart (<100 GB)
- Super Mart (100> <500 GB)
- Hyper Mart (>500 GB)

Ett Data Mart (DM) kan vara en smidig ingång till en permanent DW-miljö men knappast i sig en slutlig lösning. Risken är att man på sikt hamnar i en situation med ett antal helt separata DM med vattentäta skott emellan – precis den problemsituation mellan de operativa tillämpningarna som en DW-satsning avsåg motverka. Vi är tillbaka till utgångsläget – bara en abstraktionsnivå högre. Se figur 14.



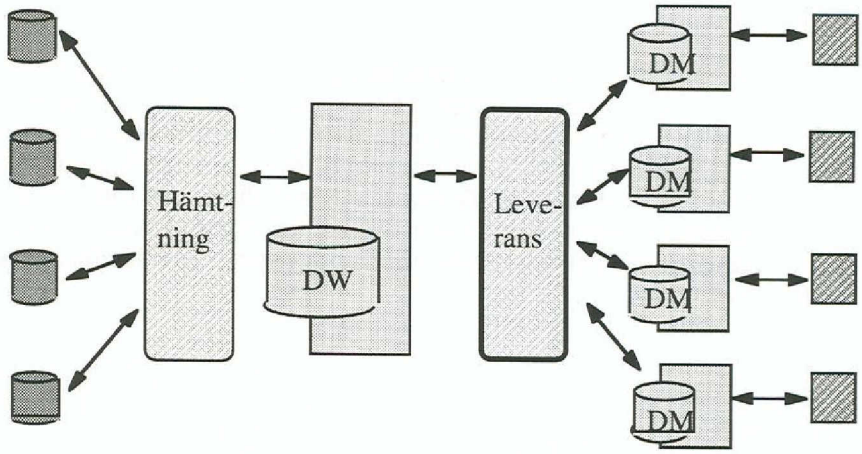
Figur 14

Informationen finns innesluten i respektive DM. Sannolikt har delvis samma källdatabaser utnyttjats. På så vis blir informationen delvis överlappande. Dock – samordning saknas. Kanske går det att ställa samma fråga till två olika DM. Sannolikheten för samma svar ska dock knappast övervärderas. Figur 15.



Figur 15

Går det då att finna någon godtagbar kompromiss mellan dessa ytterligheter? Nej knappast en lösning som undanröjer nackdelarna och bibehåller fördelarna fullt ut. Däremot diskuteras alltmer en integrerad DW/DM-lösning för att råda bot på prestandaproblem och eventuellt bristande användargränssnittsanpassningar. Se figur 16. Idén är att etablera ett DW med all den information som infoanvändarna sammantaget behöver enligt den grundläggande strategin. Samtidigt etableras ett antal DM för att svara upp mot olika användarkategoriers behov. Till skillnad från ovan hämtar nu DM sin delmängd information från det gemensamma DW. Genom att användaren kan jobba direkt mot ett DM genererar inte ett centralt DW någon prestandaflaskhals. Ett DM kan också fortfarande erbjuda mer specialanpassade gränssnitt.

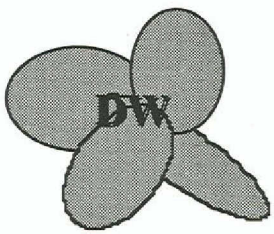


Figur 16

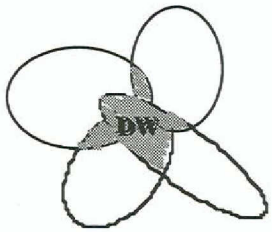
Negativt är att arkitekturen blir mer komplex med, förutom både DW och DM, nu även en funktionalitet för avpassad leverans från DW till DM.

Dessutom finns de vanliga bekymren med datamodellintegrering kvar mellan källdata och DW.

Ytterligare en arkitektur som i viss utsträckning tar hänsyn till detta dilemma förs ibland fram. Här begränsas DW till att innehålla endast den rimligt gemensamma informationen enligt figur 17 b istället för enligt figur 17 a. Övrig information tillgodoser respektive DM efter eget huvud.

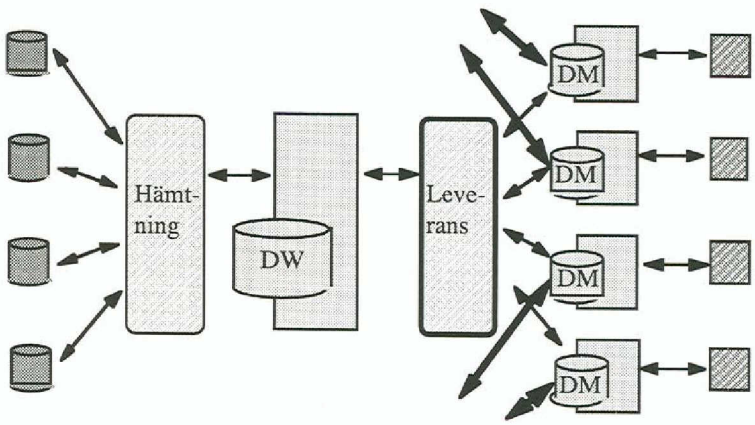


Figur 17 a



Figur 17 b

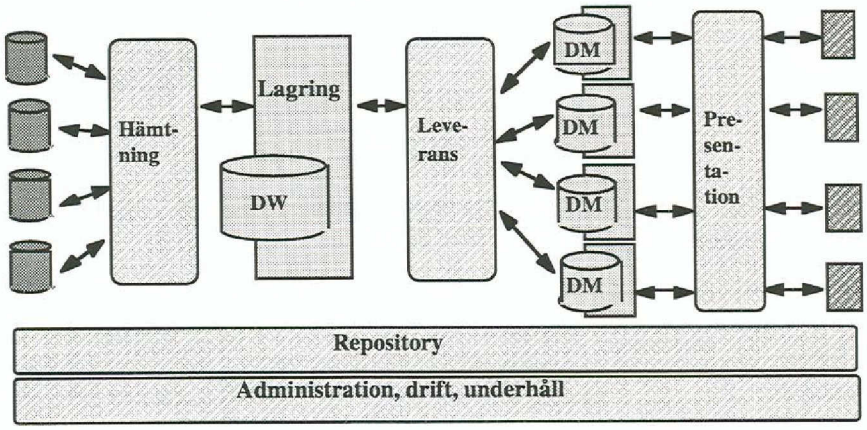
Fördelen med denna ansats är att trots allt en viss semantisk och strukturell samordning av data åstadkoms. Den samordnade DW-delen kan ju expandera efterhand under förutsättning att samarbetsvilja, ansvar, arbetsorganisation kan uppammas för det verksamhetsgemensamma syftet.



Figur 18

3.7 Arkitektur

Till sist kompletterar vi bilden med ett presentationssteg mot användaren samt med två sammanhållande, genomgående funktionaliteter. Den ena, som benämns Repository, står för hantering av all erforderlig beskrivningsinformation för att hålla ihop hela arkitekturen och få de olika delarna att fungera på avsett sätt. Den andra, Administration, Drift, Underhåll, står för de kontinuerliga arbetsuppgifter som behövs för den löpande driften. I de följande avsnitten kommer respektive delfunktionalitet att kortfattat diskuteras. För att kunna "sjösätta" denna arkitektur behövs en teknisk infrastruktur som framförallt i större verksamheter med etablerat DW-stöd blir avancerad.



Figur 19

4. Hämtning

4.1 Steg

Att föra över data från källdatabaser till ett DW låter okomplicerat men är i själva verket en komplex aktivitet bestående av ett antal arbetssteg. Allmänt bedöms denna aktivitet vara den dyraste och svåraste länken i arkitekturen. Ett misslyckande här innebär en misslyckad DW-satsning.

Steg att genomlöpa är bl a

Utsökning

Att finna och föra över data ur respektive källdatabas. Notera, som tidigare nämnts, att källdatabaserna kan finnas på många plattformar (hårdvara, operativsystem), vara strukturerade enligt olika datamodelleringsprinciper, ha olika typer av generella gränssnitt för åtkomst – om något, vara batch- eller onlinebaserade, vara underställda olika typer av behörighets- och andra driftsbaserade restriktioner, m m.

Filtrering

Kan t ex drabba uppgifter från en källdatabas som, på grund av gränssnittsbegränsningar eller dylikt, innehåller data som inte är intressant och därför ska tas bort. I de fall det kan finnas duplikat – ett typiskt exempel är adressuppgifter – bör dessa om möjligt upptäckas och raderas.

Granskning

Finns alla önskade data med från källdatabasen? Om inte, är frågan fel ställd eller finns brister i källdata? Finns luckor där sådana rimligen inte borde finnas. Vad innebär ett utelämnat värde – att uppgiften inte är aktuell eller att den bara inte registrerats? Håller sig data inom rimlighetsgränser? Verkar överlag kvaliteten på data acceptabel? O s v. Denna granskning bör genomföras per källdatabas. Ansenlig persontid kan behöva sättas in i arbetet.

Formattransformation

Varierande syntax och inkodningsformat kan finnas för samma datatyper mellan de olika källdatabaserna. De måste samtliga omvandlas till de format som DW kräver. T ex kan datatypen Kön vara angiven som Kvinna resp Man i en databas, som 1 eller 0 i en annan, som True eller False (!) i en tredje, finnas som 1 eller 0 i position fyra i ett sammansatt fält i en fjärde, o s v.

Schematransformation

Samtliga källdata måste struktureras om i enlighet med den datamodell som gäller för DW. Detta arbete kan vara nog så komplicerat om de olika datamodellerna skiljer sig åt. Än mer knepigt om modellerna är uppbyggda enligt olika modelleringsprinciper, t ex från en hierarkiskt uppbyggd modell till en relationsbaserad modell.

Sammanslagning

Ofta finns överlappningar mellan de olika källdatabasernas framtagna data. De måste integreras och rensas. Här kan olika typer av inkonsistensrisker uppstå eller risker för kvalitetsförsämring om det inte sköts rätt. I allmänhet behövs ställningstaganden som bygger på kännedom om de inblandade källdatabasernas syfte, opererande tillämpningar, användare, m m. Ett enkelt exempel: Om en person har två telefonnummer i en databas och ett annat i en annan databas, innebär det att DW ska innehålla samtliga tre telefonnummer, de två från den ena databasen eller det enda från den andra databasen. Om det enda telefonnumret överensstämmer med ett av de två från den andra databasen, indikerar det tillräckligt väl att endast "minsta gemensamma nämnaren", d v s det telefonnummer som finns i båda ska lagras? Telefonnummer kanske inte är så viktigt men om uppgifterna avser kvantitativa data som

sedermera kommer att ingå i olika typer av summeringar och andra beräkningar (aggregeringar) kommer kvalitetsaspekten återigen in i bilden.

Aggregeringar

Vet man att vissa uppgifter, som erhålls genom mer eller mindre resurskrävande beräkningar, ofta efterfrågas av infoanvändaren kan det finnas anledning att redan vid inladdningen till DW utföra denna operation. Olika typer av summeringar är exempelvis mycket vanliga ingredienser i mer övergripande frågeställningar. Optimeringsalgoritmer av olika slag kan tillämpas om summeringsoperationerna utförs vid laddningstillfället.

Laddning

Sista steget i hämtningssekvensen. DW fylls med förhoppningsvis rätt uppsättning, rimligt kvalitetsgranskade data.

Kostnaden uppstår i första hand vid behov av personella insatser för att ta ställning till datakvalitet, för rättningar, anpassningar av olika slag, för hantering av diverse exceptionella omständigheter i datamassan, för samspelet mellan de operativa systemen och DW, ansvarsöverväganden, m m.

Som om inte detta vore nog finns ytterligare aspekter att ta ställning till i samband med hämtning:

Hur ofta?

Ska DW ajourhållas direkt när uppdatering sker i för DW relevanta data i någon källdatabas (hur man nu får reda på det), en gång per dag, en gång per månad, händelsestyrt, reglerat genom en DW-agent som håller reda på vad som händer i källorna, ... Vid bedömning har varje enskild DW att utgå från sina specifika förutsättningar. Om t ex trendanalyser över det senaste året är det primära syftet med ett DW är knappast den senaste dagens uppgifter utslagsgivande för nyttan.

Hur mycket?

Ska DW nyskapas med all information från källorna vid varje uppdatering? Detta alternativ är en tung operation men betydligt enklare att utföra och lagringsoptimera. Klarar smidigt aggregeringsoperationer enligt ovan. Ett acceptabelt alternativ vid inte alltför stora datamängder eller alltför täta uppdateringsbehov.

Att istället basera uppdateringarna på tillskott är ett effektivt alternativ i de fall tillskottsinformationen enkelt kan avgränsas i källdatabaserna. Dit hör tidsangivna uppgifter, t ex den senaste månadens faktureringar, gårdagens försäljningstransaktioner, o s v. I sådana fall blir varken utsökningar eller laddningar speciellt komplicerade. Har däremot uppdateringar av annat slag utförts, t ex enskilda uppgifter i en post eller relationstuppel, ställs krav på lämplig registering av dessa i källdatabaserna (exv genom loggning). Samtidigt måste en avancerad uppdateringsfunktionalitet finnas mot DW, inte minst om DW lagrar aggregeringar av olika slag.

Ändring eller tillägg?

När det gäller uppdateringsalternativet tillkommer en komplikation. Man måste kunna skilja på vad som är en ändring av en tidigare uppgift och vad som är ett nytillskott. Även om källdatabasers loggningsfiler innehåller operationstyp finns genom denna inga möjligheter att ana vad operationen i realiteten innebär inom en tillämpning. Ett enkelt fall: innebär en lagring av två telefonnummer semantiskt att tidigare telefonnummer ska raderas eller innebär det tillskott av två nya? Visserligen kan man hävda att en rik uppsättning operationstyper reglerar alla alternativ. I realiteten är sällan operationstyperna så nyanserade – i alla händelser vet bara den som implementerat tillämpningen vad som i praktiken gäller för källdata.

Anpassning till föränderliga förutsättningar?

Sammanslagningen av data från olika källor ställer krav på anpassning till källdatabasernas dynamik. Dels gäller det att samordna uppdateringsrytmen så att integreringen sker vid för de ingående källorna neutrala och koordinerade tillstånd. I annat fall finns uppenbara

inkonsistensrisker återigen. Vem har överblick över samtliga källdatabasers "rytm" för att kunna göra dessa strategiskt viktiga bedömningar? Är en prisuppgift på en vara i källa A med automatik mer korrekt än ett annat pris på samma vara i källa B bara för att A-uppgiftens tidstämpel (i den mån den finns) är senare? Kan inte andra omständigheter spela in? Hur vet man om uppgifter i källdatabasen är en aktuell reflexion av ett mycket komplicerat arbete inom en så kallad lång transaktion (med ett antal sub-commits) och inte en definitiv uppgift för ett neutralt tillstånd?

Alla källdatabaser har primärt att svara upp mot de behov som dess primärändamål formulerar. Dessa syften varierar i de allra flesta fall kontinuerligt över tiden. Bland annat resulterar vissa anpassningar i ändringar i datastrukturen. Hur får DW-ansvariga reda på detta för eventuell anpassning av egen datastruktur eller åtminstone hämtningsalgoritmer? Hur hanteras historikdata i DW som över tiden baseras på olika datastrukturer? Kanske ändras affärsregler, datakontroller på sätt som inte framgår explicit i datastrukturen men väl i tillämpningskoden. Vilken DW-representant har insyn i tillämpningarna och hur dessa hanteras?

Som synes är denna fas en diger uppgift. Vi har förvisso mest målat upp bekymmer. Syftet är dock inte att avskräcka, endast att belysa vad som måste klargöras, bedömas, beslutas. Många DW-satsningar är i dagsläget begränsade till en eller ett fåtal källor och arbetar endast med data som man har "grepp om" – förutsättningar som smidigt undanröjer en hel del av den ovan beskrivna problematiken.

Lyckligtvis finns numer också verktyg att köpa för ändamålet. Även om de inte klarar allt representerar de ett sammanvägt kunnande som sällan finns representerad i det egna företaget. Kommersiellt tillgängliga produkter är därför i allmänhet att föredra över en hemsnickrad lösning av en (tillfälligt?) intresserad tekniker.

4.2 Datakvalitet

En mycket viktig aspekt på DW är datakvalitet. Kvalitet är verksamhetens och källdatabasernas ansvar, knappast något som DW eller dess arkitektur påverkar negativt, snarare tvärtom genom olika analyser i hämtningsfasen. Problemen kan ha funnits länge i källorna men där över åren kringgåtts eller på olika sätt "neutraliserats". Däremot får DW ta de omedelbara konsekvenserna av dålig kvalitet på sätt som väl sammanfattas i uttrycken "Garbage in, garbage out!" kompletterat med "Poor data quality – user dissatisfaction – disuse – warehouse failure".

Kvaliteten påverkas av ett antal faktorer. Bland dem kan nämnas:

- Hur fullständiga uppgifter?
- Hur korrekta uppgifter?
- Hämtat vid rätt tidpunkt? Hur ofta?
- Överlappande data mellan olika källor? Kanske initialt samma ursprungskällor men förvanskade längs olika bearbetningsvägar i den operativa miljön?
- Finns kunskap om källsystemens datahantering och dess effekter på källinnehållet?
- Fel, brister i datastrukturer?
- Bristande kunskap om källdatastrukturernas och de operativa tillämpningarnas modifieringar över tiden samt de konsekvenser för innehållet i källorna detta kan ha orsakat.
- Finns gamla databaser utan explicita datastrukturer (scheman)? Innebär osäkra antaganden.
- Brister i överförings-, förädlingssystemen?

Till viss del skulle problemen kunna begränsas om data kompletterades med diverse beskrivningsuppgifter som karakteriserar respektive dataelement. Intressanta uppgifter kan t ex vara:

- Ursprungskälla
- Skapad av
- Skapad när
- Dataansvarig
- Bedömd kvalitet
- Vilken version av data
- Tidstämpel för aktuell version
- Vilken repository-version gällde vid skapandet
- Behörighetskrav
- Kodningsprincip, format

Data om data, ofta under beteckningen metadata (obs, annan betydelse än modellinformation i repositoriet), har ännu rönt ett mycket svalt intresse utom inom vissa tillämpningsområden, t ex Geografiska Informationssystem, där dessa metadata traditionsenligt spelar en vital roll. Tids- eller versionsstämpling av data i DW tycks dock vara regel, bland annat för att kunna realisera en koppling till den version av repositoriet som gällde vid skapandet.

Betydelsen kommer säkerligen att uppmärksammas mer i och med ökad erfarenhet, komplexare DW-miljöer, tuffare infoanvändarkrav.

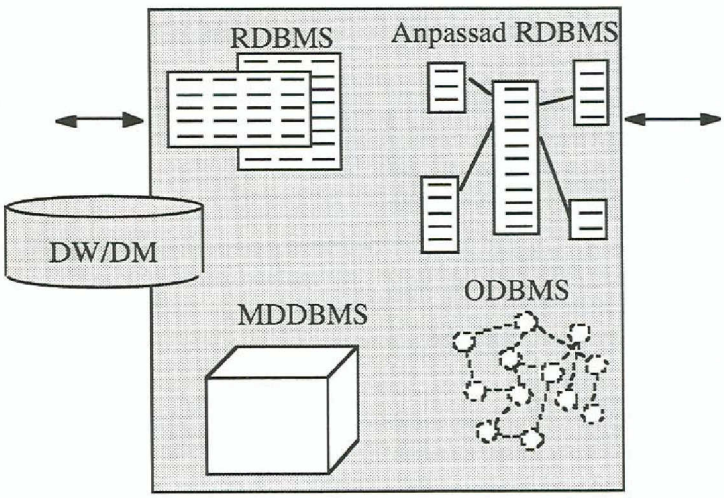
En orsak till det begränsade intresset kan bero på att dessa uppgifter i praktiken är svåra att skapa, underhålla, överföra på ett effektivt sätt. Att ta ställning till är, bl a:

- Uppgifts-granularitet; Ska uppgifterna noteras per element, objekt, objekttyp, överföringsomgång, källa, ...)? Att notera metadata för varje dataelement är våldsamt lagringskrävande. Sannolikt kan i flertalet fall "grövre" principer tillämpas.
- Hur överförs dessa data? Separerat från data – integrerat med data?
- Hur skapas, tillförs, underhålls dessa uppgifter? De finns normalt inte bland operativa data?
- Hur hanteras uppgiftsluckor?
- Vem/vilka har metadataansvar?
- Kan metadatauppgifter i källdatabaser identifieras?
- Hur genereras metadata för aggregerade data? Varje dataelement som ingår i en beräkning har ju i teorin "sin" uppsättning metadata.
- Även metadata har en struktur som förändras över tiden genom nya kravbilder. Hur hantera olika versioner? Genom att metadatabeskriva metadata?

5. Lagring

5.1 Alternativ

Ett DW är en databas som behöver hanteras av någon typ av databashanterare. Ett krav är att en rimligt förståbar datastruktur eller schema kan upprättas. Ett annat är tillgången till ett generellt gränssnitt för utsökning av data. Uppdateringsmöjligheter är däremot av begränsat intresse. Grovt sett finns i dagsläget fyra alternativ enligt figur 20.



Figur 20

Deras egenskaper kommer att kortfattat diskuteras i de kommande avsnitten.

5.2 Relationsdatabashanterare (RDBMS)

Relationsmodellen är välkänd och med en drygt 25-årig historia. Antalet produkter är omfattande. De allra flesta nya databastillämpningar tillämpar teknologin. Rikhaltig uppsättning integrerbara kringsystem finns tillgängligt.

Gränssnittet SQL är en internationell standard innehållande operationer för såväl uppbyggnad av schema (DDL) som för operationer på databasen (DML). SQL erbjuder en höggradig portabilitet. SQL klarar enklare summeringar/kalkyler utan större problem. Komplexare beräkningar måste dock utföras genom anrop till externa rutiner.

RDBMS har länge kännetecknats av prestandaproblem i samband med joins, d v s när det finns behov att navigera över en datastrukturs komponenter i ett antal steg, något som ofta är en realitet i en DW-miljö. RDBMS håller dock på att bli mycket snabbare, bl a genom att tillämpa

- 64 bitars processorteknik
- Binärindexering
- Parallella system (SMP = Symmetric multiprocessors, MPP = Massively parallel processors).

Man bör hålla i minnet att RDBMS i första hand är optimerade för OLTP, inte DW. Alla ledande RDBMS kan dock konfigureras för olika behov. Trots prestandatrimningar kan vissa frågetyper över stora databaser ge prestandaproblem. Att teoretiskt utreda effekter av olika optimeringsstrategier är komplicerat. Att utföra reella tester är i allmänhet en klok åtgärd. Att RDBMS klarar mycket stora databaser är väl känt.

Stora RDBMS-leverantörer, en stabil marknad och rikhaltig erfarenhet skapar trygghet.

5.3 Anpassad RDBMS

Vissa leverantörer specialiserar sig på DW-tillämpningar med utnyttjande av grundläggande RDBMS-teknik men med anpassningar som optimerar för DW-arbete. Man har valt att kompensera SQLs svaga punkter, bl a med betydligt rikhaltigare funktionsrepertoar för aggregeringar av olika slag. Produkterna inom denna kategori anses erbjuda ett mer användarvänligt gränssnitt. Notera att dessa utvidgningar är produktspecifika. Standard saknas. Dock pågår kontinuerligt arbete på att komplettera SQL-standarden i olika avseenden. Kanske minskar då behovet av produktspecifik SQL.

Eftersom man i stort kan bortse från bekymmer kring uppdateringar (transaktionshantering, loggning, rollback, deadlocks, låsningar, ...) kan all koncentration läggas på optimala utsökningsstrategier och snabba beräkningar. Höggradig indexering, ofta binärindex, tillämpas regelmässigt.

Normalt förknippas en modelleringsteknik för DW-scheman, som går under beteckningen **Star schema,** med denna kategori. Detta är inte helt rättvisande eftersom i dagsläget de större RDBMS-leverantörerna optionellt kan erbjuda likartad strukturerings- och utsökningsteknik. Star-schema-ansatsen diskuteras vidare i avsnitt 5.4, nedan.

Star scheman anses under mer begränsade förutsättningar med enklare datastrukturer och inte alltför stor databaser ge utmärkta prestanda. På så vis kan de sägas ha ett naturligt användningsområde i Data Marts. Men allt har sitt pris. Vid mycket stora databaser (>1 terabytes?) eller mer komplicerade datastrukturer anses tekniken komma till korta. Noggranna, realistiska tester rekommenderas.

Inom DW-världen har ansatsen ett antal tunga "gurus" som företrädare. Marknadsryktet är gott. Många "success stories" genom snabbt realiserade lösningar, bra gränssnitt och utmärkt prestanda finns redovisade.

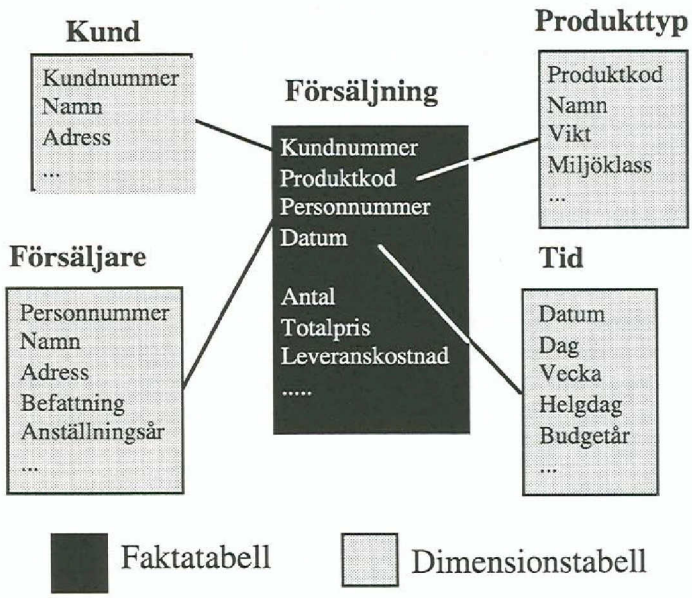
Glöm emellertid inte bort att de traditionella RDBMS-leverantörerna numer ser DW-marknaden som ett mycket expansivt och lukrativt område att penetrera. De relativt sett små leverantörerna inom den anpassade kategorin kommer att få känna av en hård konkurrens.

5.4 Star schema; Princip

Att i detalj beskriva datastrukturering enligt Star Schema-ansatsen skulle kräva en egen rapport. Detta avsnitt inskränker sig till att indikera principerna.

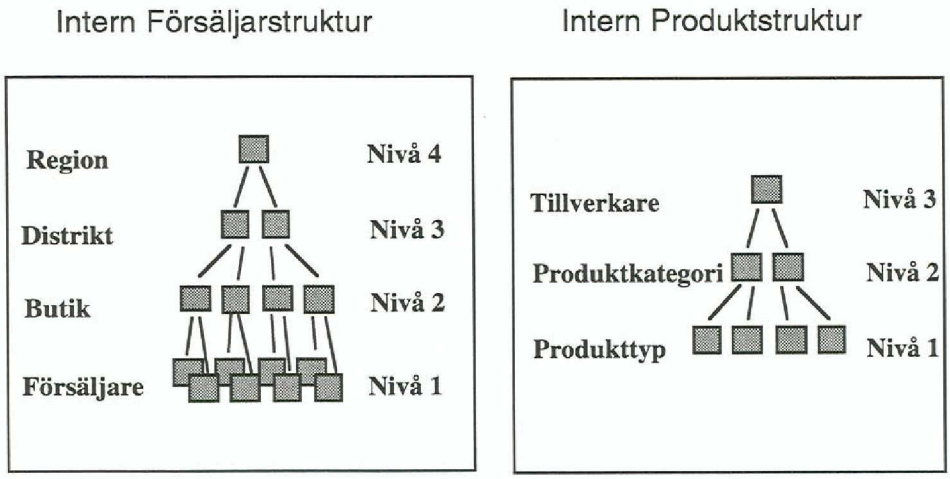
Antag att vi är intresserade av att på olika sätt analysera de försäljningar företaget åstadkommit under de senaste åren. Dessa uppgifter finns lagrade i en relationstabell. Varje försäljning är dokumenterad med vilken produkt (Produkt) som såldes till vem (Kund), av vem (Försäljare) och när (Tid). Dessa fyra attribut utgör tillsammans tabellens primärnyckel. För varje sådan försäljning noteras därutöver bland annat hur många enheter som såldes (Antal), vad det med hänsyn till styckepriset blev för Totalpris samt tillkommande Leveranskostnad. Se mörka tabellen i figur 21. Den brukar benämnas *faktatabell* eftersom det är olika kombinationer värden från den tabellen som är av intresse att få som svar. Genom att ställa olika typer av villkor på var och en av de ingående nyckelattributen avgränsas olika submängder av den totala mängden försäljningar. T ex "Ge mig alla försäljningar som försäljare Pettersson genomfört under 1996 av kaffebryggaren Snabbt-och-Lätt". Ännu sannolikare är kanske samma frågevillkor men med början "Ge mig den totala försäljningssumman för alla försäljningar som ... ", d v s svaret presenterat som en enda summauppgift istället för varje ingående försäljning. Index på varje nyckelattribut ställer sig naturligt eftersom frågevillkoren kan förväntas variera från fråga till fråga.

Men en kund är inte bara ett Kundnummer, en produkt en Produktkod o s v. De har var för sig en uppsättning attribut av intresse dels som uppgifter av allmänt intresse dels för att använda i mer nyanserade frågeställningar och villkor mot försäljningstabellen. Dessa lagras i separata tabeller kallade *dimensionstabeller*. Se gråtonade tabellerna i figur 21. Nu kan t ex frågan "Ge mig det totala antalet sålda enheter av Miljöklass 2 produkter som Distriktschefer (befattning) sålde vecka 49 till kunder med kundnummer i intervallet 1100-1200."



Figur 21

I princip är detta ett helt vanligt schema med relationer enligt 3:e normalformen. Men det kan finnas behov att rucka på principerna. Antag att försäljarna och produkterna är indelade i interna strukturer enligt figur 22.



Figur 22

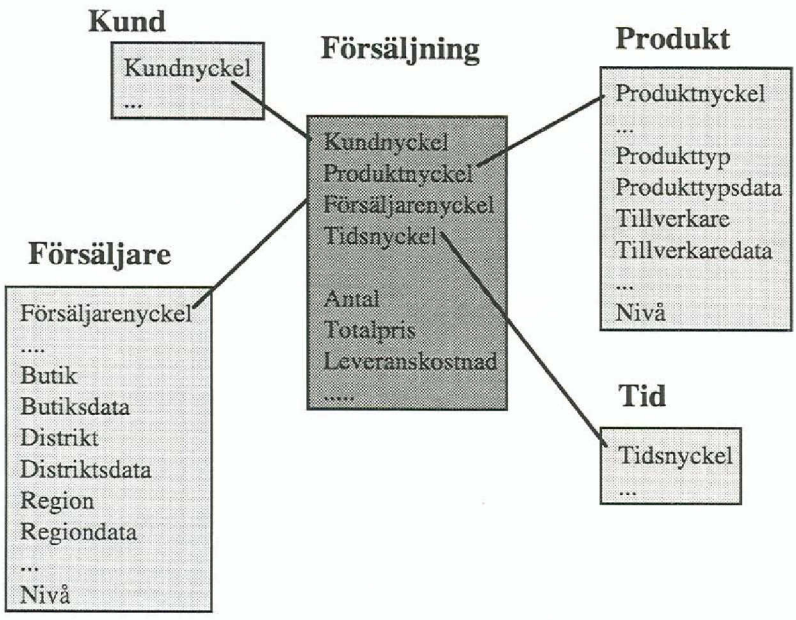
3:e normalformen skulle ställa krav på en tabell per nivå om varje nivå kräver sin beskrivning. Den enkla grundstrukturen enligt figur 21 blir plötsligt en komplex struktur med ytterligare fem tabeller. För infoanvändaren blir det genast svårare att förstå men framförallt svårare att formulera frågor mot den nya strukturen. Ett alternativ är att föra in samtliga generaliserings-nivåer under den lägsta nivåns tabell. I Försäljartabellen får vi en rad för, förutom alla enskilda försäljare, samtliga butiker, distrikt och regioner samt respektives attributuppgifter. De tre sista kategorierna kan upplevas som mer generaliserade "försäljare". Försäljare "Dalarna" representerar alla försäljare i alla butiker i distriktet Dalarna. Personnummer räcker inte längre som nyckel eftersom detta attribut inte finns för "försäljarna" på nivå 2 och uppåt. En neutral Försäljarenyckel etableras. Samma princip tillämpas för Produktstrukturen. Med alla nivåers

samtliga attribut integrerade på en och samma plats kan villkor ställas på valfri delmängd av samtliga attribut – en mycket flexibel förutsättning.

Ska det vara någon finess med att utvidga dimensionstabellernas betydelse enligt ovan bör också faktatabellen sättas att svara mot de nya förutsättningarna. Faktatabellen ska innehålla försäljningsuppgifter inte bara per enskild försäljare utan även för generaliserade försäljare. Bl a ska finnas beräknat och klart för försäljare "Dalarna" samtliga kombinationer av övriga primärnyckelkomponenter. Dit hör i en tuppel "Dalarnas försäljning under datum X till kund Y av produkttyp Z" liksom i en annan tuppel "Region Norr"s försäljning under datum X till kund Y av produkttyp Z", o s v.

Samma sak etableras för produktstrukturen. Eftersom nu dimensionstabellerna måste kunna härbärgera attribut härrörande från samtliga nivåbeskrivningar blir tabellerna dels denormaliserade, dels mycket stora. Ett extra attribut i form av en nivåangivelse måste tillföras dels för att indikera vilken uppsättning attribut som är aktuella för aktuell tuppel (Butiksdata, Distriktsdata, ...) men framförallt för formulering av entydiga villkorsspecifikationer, se avsnitt 5.5 nedan. Icke-aktuella uppgifter måste nullifieras.

Faktatabellen innehåller nu både de tidigare detaljerade uppgifterna och summeringar över alla dimensionsnivåer och dimensionsnivåkombinationer. Antag att även Kund har en intern struktur med högsta nivå i form av Kundkategori med möjliga värden "Stamkund" respektive "Strökund". Antag vidare att Budgetår är högsta nivån för Tid. Att då ställa frågan "Ge mig Leveranskostnad givet försäljare Region_Norr (region), kund Stamkund (kundkategori), produkt AB_Spis_och_Kyl (tillverkare), tid 1995 (budgetår)", ger ett synnerligen summerat värde till svar. Med t ex 5 regioner, två kundkategorier, 10 tillverkare och 3 budgetår finns det i faktatabellen 5•2•10•3=300 summavärden av samma dignitet. Jämför detta med att kanske behöva summera upp från många miljoner grunddatatuppler. Summeringsoperationerna för alla "ledder och bredder" utförs förslagsvis i samband med inladdningen till DW. Som lätt inses är knappast uppdateringar tillåtna med Star Schema-filosofin. Våldsamt många summeringsvärden skulle behöva uppdateras för varje uppdatering på lägsta nivå.



Figur 23

Som synes tillämpas inte samma designprinciper som vid vanlig datamodellering. Denormaliseringen är effektiv för att snabba upp utsökningar och begränsa till mer överblickbar datamodell. Behovet av joins begränsas genom att varje dimension endast är en tabell. Summerade data kan

snabbt tas fram, ofta utan beräkningsbehov. Ställs andra villkor än bara nivåbaserat måste dock på vanligt sätt beräkningar utföras. T ex "För alla distrikt som ... och alla tillverkare som .... ".

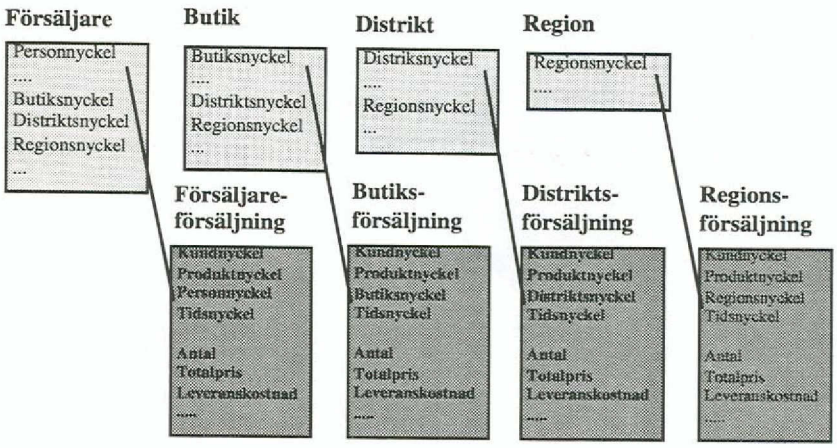
Bland nackdelarna kan nämnas

- Stora dimensionstabeller.
- En artificiell nivåindikator måste alltid anges.
- Även summerade data i samma jättelika faktatabell istället för utbrutna i kompaktare, mindre tabeller.
- Fastlåst hierarki-struktur.

5.5 Snowflake Schema

Nivåindikatorn är ett bekymmer genom den stelbenthet den bygger in i strukturen. Samtidigt är den alldeles nödvändig för korrekt specifikation av förutsättningar i t ex villkorsangivelser för att inte blanda ihop olika nivåer. Ställer man mot Försäljaretabellen villkoret "where Region = Region_Norr" kommer resultatet att bli "Region_Norr" plus samtliga distrikt som hör till "Region_Norr" plus samtliga butiker som ligger i distrikt som hör till "Region_Norr" plus samtliga försäljare som jobbar i butiker som ligger i distrikt som hör till "Region_Norr". Antagligen är inte det vad som avses. I alla händelser inte alltid. Vill vi explicit ta fram alla butiker inom "Region_Norr" kan det göras med hjälp av nivåindikatorn nämligen som "where Region = Region_Norr and Nivå=2".

Ett sätt att undvika nivåindikatorer är att ta ett steg tillbaka mot normalisering, dock inte fullt ut. Nivåindikatorn utgår ur dimensionstabellen. Dessutom skapas en ny sub-dimensionstabell för varje nivå ovanför den grundläggande innehållande samtliga relevanta attribut för nivån (inklusive nycklar till övernivåer). Därigenom kan villkor ställas lika flexibelt som förut samtidigt som nivåindikatorn ersatts med en explicit dimensionstabell per nivå. Samtidigt delas faktatabellen upp på motsvarande sätt. Vi erhåller ett "Snowflake schema". Figur 24 visar resultatet för försäljaredimensionen.



Figur 24

Övriga dimensioner delas på motsvarande sätt upp i sub-dimensionstabeller och sub-faktatabeller. Dessutom tillkommer alla kombinationer av dessa subdimensionsnycklar och den därmed motsvarande uppdelningen av faktatabellen.

Bland fördelarna kan nämnas

- Varje uppgift upplevs nu ligga på rätt plats, "var sak på sin plats".
- Risken för felaktigt ställda frågor minskar.

- Dimensionstabellerna är joinbara för att, vid behov, nå uppgifterna på den lägsta nivån.
- Strukturella ändringar blir enklare.
- Utsökningar av summerade data går snabbare över mindre tabeller.
- Anpassning till variabla summeringsbehov kan genereras.

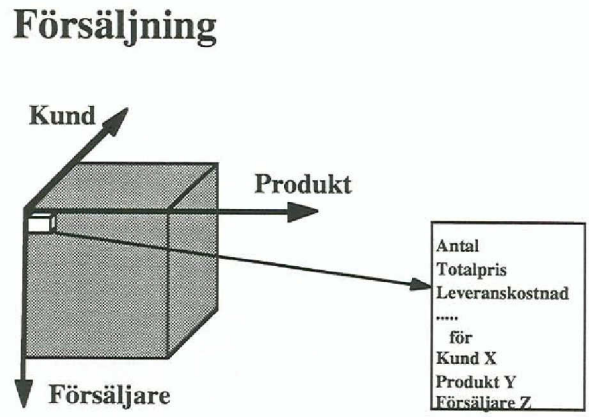
Bland nackdelarna kan nämnas

- Ett mycket fragmenterat schema.
- Enhetlig namngivning av faktatabell-attribut är ett krav.
- Inkonsistensrisker mellan tabeller.
- Tuff databasadministratörsuppgift.
- Tungt/svårt besvara vissa typer av frågor.

5.6 Multidimensionell DBMS (MDDBMS, MDD)

Den flerdimensionella ansatsen har också anammats av de multidimensionella databashanterarna (MDDBMS). Medan anpassade DBMS bygger på SQL tillämpar MDDBMS egenutvecklade språk. MDDBMS använder dessutom en arraybaserad lagringsteknik. Valfritt antal dimensioner kan etableras.

Grundidén är att uppleva data som existerande inom en mångdimensionell "värld". Se exempel på en tredimensionell databas i figur 25. En mångdimensionell upplevelse av data är många gånger mycket attraktivt i ett användargränssnitt. Lagring och presentation måste dock hållas isär. Oavsett lagringsteknik kan, om så önskas, presentationen ske enligt lämplig multidimensionell teknik (OLAP). Se vidare avsnitt 7.3, nedan.



Figur 25

Bland typiska egenskaper hos ett MDDBMS kan nämnas:

- Optimerad för OLAP-bearbetningar.
- Utsökningsoptimerad (liksom övriga ansatser). Dock uppdateringar ibland möjliga.
- Summeringar, beräkningsresultat skapas ofta vid laddning.
- Snabb åtkomst genom avancerad indexeringsteknik. Beroende på förutsättningar i datamassan kan prestanda komma att variera kraftigt. Test med egna förutsättningar rekommenderas.
- Drill-down-teknik stöds naturligt.

- Kräver ofta "egen" uppsättning kringverktyg.
- Kräver ny kompetens. Här hjälper inga SQL-kunskaper.
- Fungerar bäst med rimligt begränsade datavolymer. Data Marts? Prestandadegradering vid stora volymer?
- Summeringar kan ta stor plats!
- Inga standarder. Varken kring modell eller API.

5.7 Utsökningsoptimeringar

5.7.1 Binärindex

Oavsett lagringsteknik enligt ovan erbjuder numer de flesta DBMS olika varianter av indexeringsteknik avpassade för utsökningsändamål. Vilka som lämpligen bör användas för en viss tillämpning beror på dess unika förutsättningar.

Binärindex förekommer ofta i DW-sammanhang. Denna indexeringsteknik erbjuder mycket snabba utsökningar i de fall man har ett eller flera attribut med en mycket begränsad kardinalitet. Antag en faktatabell med bland annat dimensionerna År/Månad och Region enligt figur 26.

År	**Månad**	**Region**
1995	Januari	Syd
1997	Juni	Sydväst
1995	Mars	Nord
1996	September	Nordost
...	...	...

Figur 26

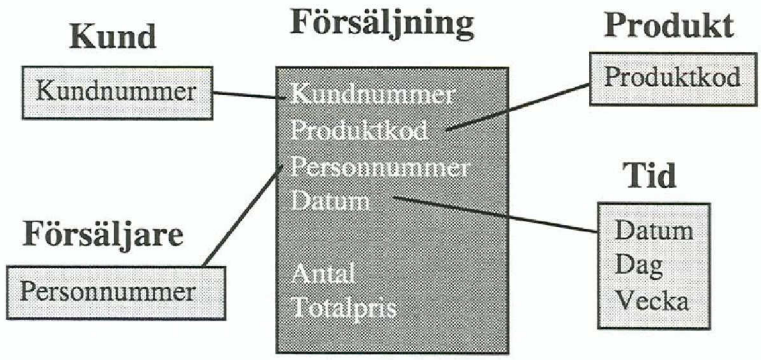
Istället för att etablera en, två eller tre indextabeller för dessa in mot faktatabellen kompletteras faktatabellen med en kolumn för varje möjligt värde. Varje kolumn upptar en bit där 1 anger att värdet gäller för raden medan 0 anger att så inte är fallet. Se exempel i figur 27.

1995	1996	1997	Jan	Feb	Mar	Apr	Maj	Jun	Jul	Aug	Sep	Okt	Nov	Dec	Syd	Sydväst	Väst	Ost	Nordost	Nord
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
.....																				

Figur 27

5.7.2 Med kartesiska produkter

Antag följande modell:



Figur 28

Antag vidare

- att det finns 100 försäljare, 10000 produkter, 800 kunder och 500 intressanta tidsperioder (dagar)
- att alla försäljare är aktiva varje dag, alla kunder köper något varje dag, ca 2000 olika produkter är föremål för försäljning inom en tidsperiod (dock förstås i olika kombinationer mellan tidsperioder) samt att av serviceskäl varje kund alltid betjänas av en "personlig" försäljare. Denna tilläggskunskap visar att endast 1/500 av det totala antalet möjliga "celler" i en mångdimensionell array har ett relevant värde.

Faktatabellen skulle innehålla 100•2000•8•500 = 800.000.000 (800 miljoner) rader.

Frågan

> "Ge mig Antal och Totalpris för allt försäljare Svensson sålt av Produktkoderna 1111, 1112 och 2113 till någon kund under Vecka 9, 1997."

skulle generera ett ansenligt arbete om vanliga parvisa joins skulle tillämpas. En första join med en av produktkoderna skulle resultera i 400.000 rader. De andra två lika många vardera. Därefter successivt mer avgränsad mängd för varje join-steg fram till kanske ca 150-200 (i snitt 168) slutrader.

Ett ibland alternativt sätt att undvika tunga joins är att skapa kartesiska produkten av sökvillkorets förutsättningar och därefter använda denna för en join mot faktatabellen. I det aktuella fallet blir det kartesiska produkten av

Försäljare	Tid	Produkt	Kund
Svensson	19970224	1111	0001
	19970225	1112	0003
	19970226	2113	0014
	...		....

d v s 1•7•3•800 = 16800 rader, som sedan joinas mot faktatabellen. En rejäl minskning av arbetsbördan. Dock ska påpekas att utfallet varierar kraftigt beroende på typ av fråga.

Antag nu att vi vet att kunderna i allmänhet enbart köper vissa produkter men dessa i stort sett varje dag.

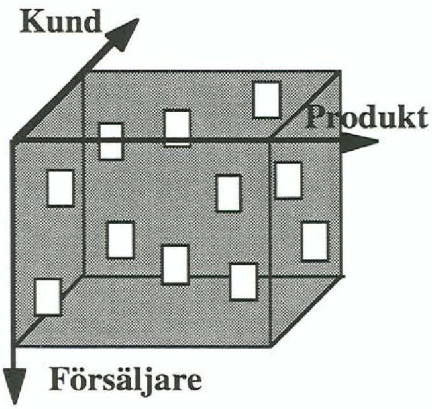
Alltså, givet
– en viss försäljare
– en viss produkt
– en viss kund

finns det försäljningsuppgifter för i stort sett varje dag enligt figur 29.

Tidsperiod	Antal	Totalpris	Leverans-kostnad
19960101	xxx	xxx	xxx
19960102	xxx	xxx	xxx
...	...	...	...
19970515	xxx	xxx	xxx

Figur 29

Däremot finns givetvis inte uppgifter för varje kombination försäljare, produkt, kund enligt tidigare angivna förutsättningar. Snarare finns 100•10000•8=800.000 giltiga kombinationer. För varje kombination finns nu inte bara en uppgift utan en sammanhängande sekvens med uppgifter, nämligen de 500 som finns för varje dag under perioden. Varje sekvens ligger sekventiellt och kompakt lagrad. Utsökning av samtliga 500 kan ske med en eller ett fåtal sekundärminnesaccesser. Se figur 30.



Figur 30

För den givna frågan skapas den kartesiska produkten av de aktuella kombinationerna, d v s 1•3•800=2400 stycken. Dessa appliceras sedan på databasen för att ta fram motsvarande antal sekvenser med maximalt 500 datumbaserade försäljningsuppgifter. För var och en av dessa sekvenser tas sedan fram de uppgifter, som svarar mot datumvillkoret.

Kunde vi genom subindexering få fram de 8 kunder som är kopplade till varje försäljare skulle endast 1•3•8=24 kombinationer resultera.

Olika varianter på detta tema förekommer. Med hjälp av tilläggskunskap om datamassan samt struktureringsalternativ kan exekveringsbehovet gå ner till en bråkdel av vad som krävs vid vanlig joinbaserad utsökning. En förutsättning är förstås en mycket god kunskap om användarbehov.

6. Leverans

Leverans från DW till DM kan ske genom datareplikering. Bland alternativ kan nämnas

- Tightly coupled (synkron, full 2-phase commit).
- Loosely coupled (synkron, egen 2-phase commit).
- Asynkron.

Det asynkrona alternativet rekommenderas i normalfallet.

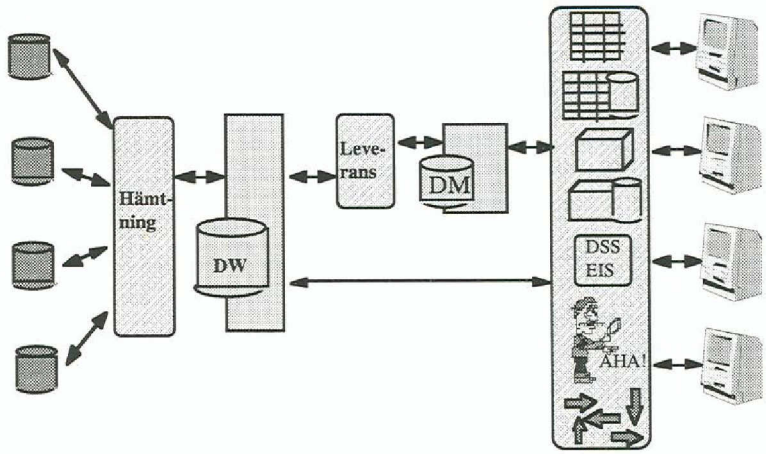
Eftersom förutsättningar i både DW och D;M ändras över tiden är det mycket viktigt att hela tiden hålla reglerna för replikering "up-to-date".

Man har också att bedöma hur ofta och hur samordnat replikeringen ska utföras, samt om principen ska vara inkremental eller total uppdatering, m m.

7. Presentation

7.1 Inledning

Infoanvändaren opererar mot DW eller DM genom olika typer av gränssnitt baserat på behov, konvention, gammal vana, företagsstrategi, e dyl. Figur 31.

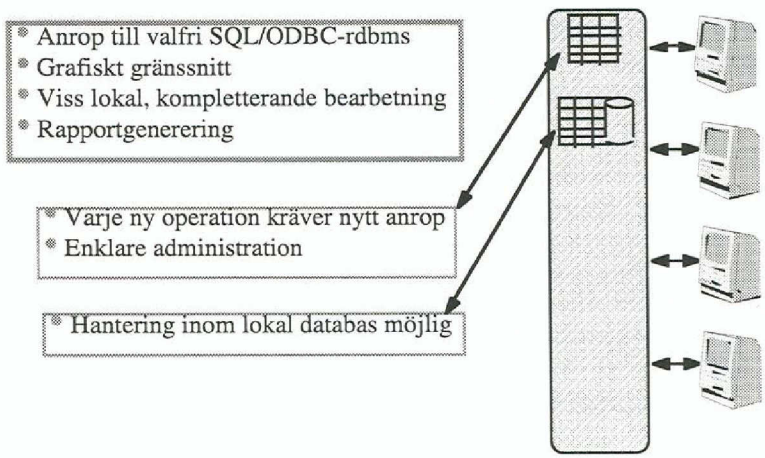


Figur 31

De vanligaste alternativen kommer att beröras i de följande avsnitten.

7.2 Tabellgränssnitt

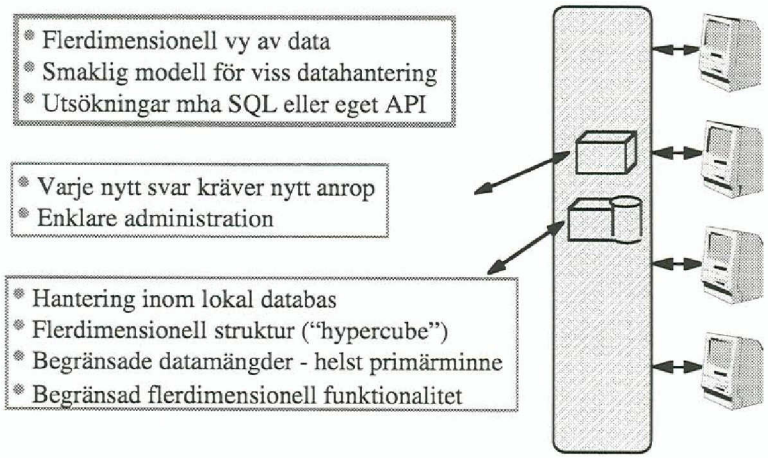
Är data lagrade i enlighet med relationsmodellen utgör SQL ett naturligt gränssnittsval. Svaren presenteras som tabeller till användaren eller för fortsatt bearbetning.



Figur 32

7.3 Flerdimensionellt gränssnitt

Flerdimensionellt gränssnitt mot DBMS är avsett för infoanvändare som arbetar främst med den typ av data som diskuterats under Star schema-ansatsen ovan, d v s med numeriska data i en faktatabell och ett antal dimensioner som alternativa infallsvinklar på dessa data. Gränssnittet avskärmar användaren från en intern översättning till SQL i och för kommunikation med RDBMS när sådant används och till produktspecifika gränssnitt när MDDBMS används.



Figur 33

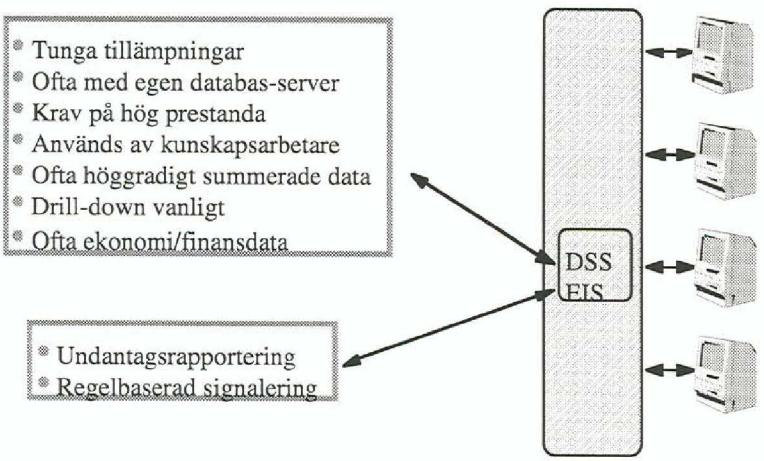
Att användare för faktatabellbaserade behov finner det både tilltalande och produktivt att kunna uppleva och operera på data utifrån ett multidimensionellt perspektiv, står alldeles klart.

Därmed inte sagt att den bakomliggande databasteknologin måste vara multidimensionellt uppbyggd.

Ett multidimensionellt perspektiv brukar gå under beteckningen OLAP (On-Line Analytical Processing). Se vidare avsnitt 7.7.

7.4 DSS/EIS-tillämpningar

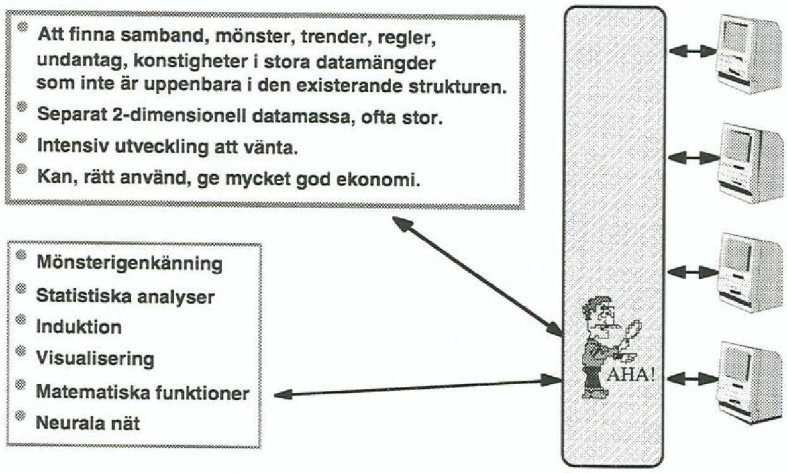
Vissa användarkategorier föredrar beslutsstödssystem med en given service avpassat till kategorins förmodade behov. Dessa typer av system har funnits under många år, kanske framförallt för sammanställd information till högre beslutsfattare.



Figur 34

7.5 Data Mining; "Knowledge Discovery"

Data Mining är ett modeord som snabbt nått DW-områdets stjärnhimmel. Där finns en blandning av magi, artificiell intelligens, avancerade algoritmer och lotteri med höga vinster men också dyra lotter. Data Mining appellerar lätt till höga chefers vardag av stenhård konkurrens och behov av välunderbyggt beslutsfattande. Nya informationsrön kan skapa konkurrensfördelar, ökad beredskap, m m. Många vittnar om snabb lönsamhet, andra om överförsäljning. I alla händelser bedömer de flesta att området snabbt kommer att vidareutvecklas med såväl bättre teori som finurligare produkter. Marknaden kommer att expandera i motsvarande mån.



Figur 35

Att tänka på innan inköp/användning, bl a:

- Ofta krävs ansenligt expertstöd avseende
  - Produktanvändning
  - Principer
  - Utvärdering.
- Välgenomtänkt syfte måste föreligga. Att tro att nya häftiga sanningar "bubblar fram" av bara farten är naivt.
- Välgenomtänkt process som en följd av syftet måste etableras.
- Ofta krävs ett ansenligt initialt datapreppareringsarbete.

Därefter kanske de nya upptäckterna kan börja genereras. Exempel på möjliga upptäckter under olika ansatser:

Association

- 80% av kunderna köper både öl och blöjor på lördagar. Inget samband övriga dagar.
- Ishockey. Lag As forward X har gjort 70% av sina mål på oss från vänster halvdistans. I 90% av dessa fall var vår back Y inne.

Klassifikation

- Köpare av operabiljetter är till x% unga yrkesverksamma i intervallet 25-35 år med inkomst över y kr.

Om – så

- De som köper lakan kommer i 75% av fallen att inom 1 månad komma tillbaka och köpa gardiner.

Mönstersamband

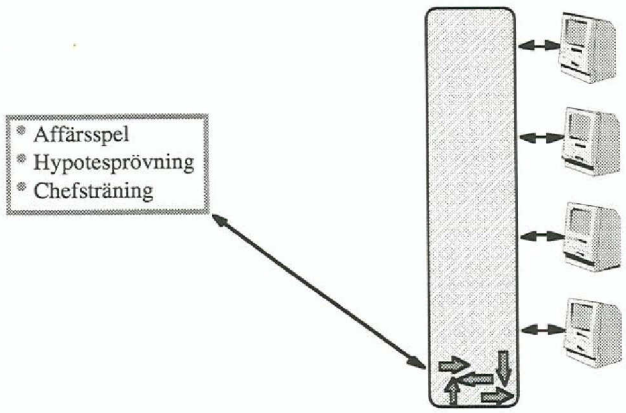
- Efter reklamkampanj för läskedryck A ökar försäljningen med 40% under närmaste kvartalet därefter. Samtidigt ökar försäljningen av chips med 30%.

Regel

- Ålder är mer utslagsgivande för investering i en pensionsförsäkring än antalet personer i familjen.

7.6 Affärssimulering

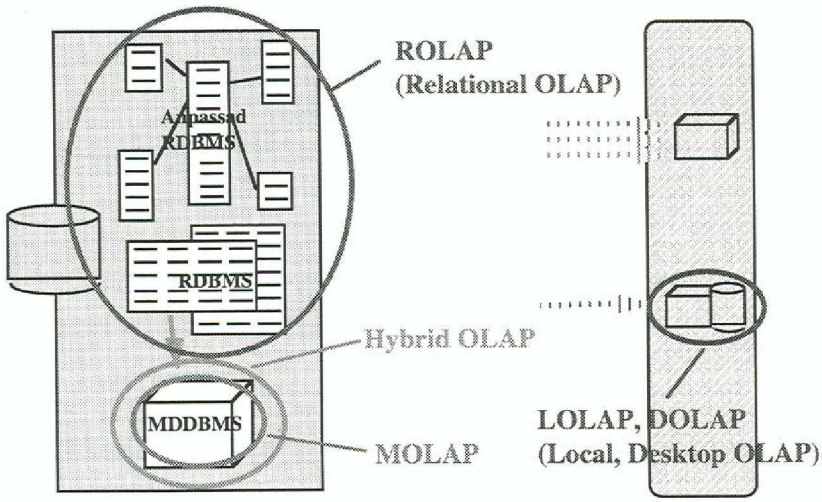
Data i DW/DM lämpar sig utmärkt för olika typer av affärssimuleringar.



Figur 36

7.7 OLAP; betydelser

"Kärt barn har många namn", återigen. OLAP (On-Line Analytical Processing) står generellt för en mångdimensionell syn på data, något som inte framgår av förkortningens betydelse. OLAP finns numer i ett antal mutationer, var och en med sin beteckning.



Figur 37

- ROLAP (Relational database based solution):

  Komplettering av konventionella rdbms med OLAP-funktionalitet.

- MOLAP (Multidimensional database based OLAP)

  En integrerad OLAP-miljö inklusive erforderlig, ofta produktspecifik, databashantering (MDDBMS).

- LOLAP, DOLAP (Local, desktop based OLAP)

  OLAP-funktionalitet som kräver samverkan med separat dbms för nerladdning av de data man önskar operera på.

- Hybrid OLAP

  En MOLAP-nära lösning men med begränsad volym och med automatisk koppling till RDBMS för nerladdning av de data till MDDBMS som inte redan finns där, allteftersom behov uppstår.

Den allmänna trenden tycks vara mot ett större utnyttjande av ROLAP. Tidigare har prestanda varit en hämsko. Nya optimeringsstrategier m m anses råda bot på de allvarligaste flaskhalsarna. De nya principerna för att strukturera data likaså.

OLAP-området är "hett" varför intensiv fortsatt utveckling kan förväntas. Sannolikt kommer man att göra en klarare distinktion mellan multidimensionell upplevelse av data i användargränssnittet och multidimensionell databasteknologi.

8. Repository; Meta Data

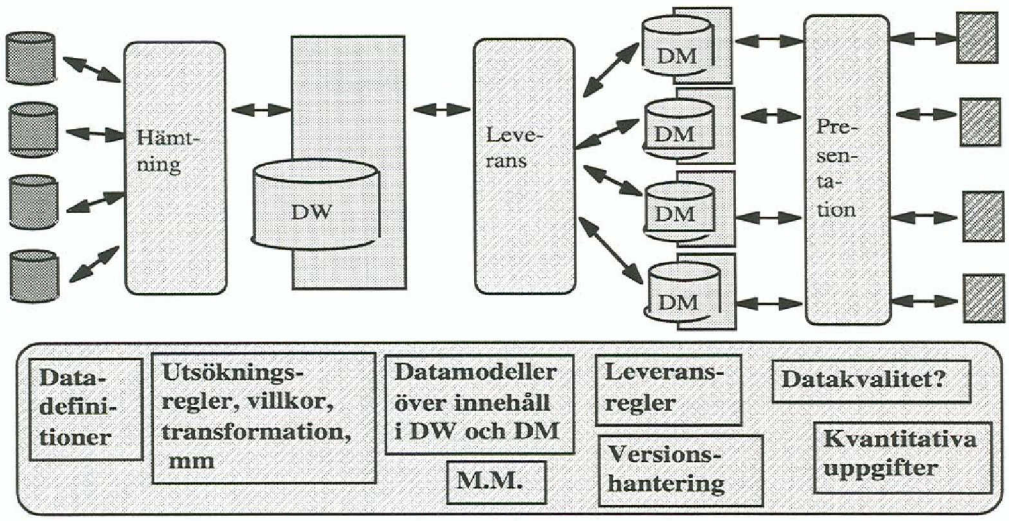
8.1 Principer

Den mycket komplexa arkitekturen måste hållas samman. Detsamma gäller givetvis också informationsflödet. Som synes i figur 38 är det ett antal olika saker att hålla reda på. Denna information bör hanteras i ett repository som ska finnas tillgängligt för alla, bl a:

- samtliga inblandade i utveckling, drift och underhåll
- samtliga användarkategorier
- tillämpliga datorbaserade stödfunktioner.

Tillgängligheten ska självklart vara god både vid exekvering, t ex vid hämtning (aktiva repositorier), som vid allmänt analys-, specifikations- och underhållsarbete.

Repositoryt blir i denna roll hela arkitekturens "hjärta".



Figur 38

Att datastrukturen blir mycket komplicerad kommer knappast som en överraskning. Inte heller är det överraskande att konstatera att de repositorier som finns idag har ett antal begränsningar samt att integrering mellan delsystem snarare är undantag än regel.

Ska repositoriet vinna tilltro måste det alltid vara "up-to-date". Samtidigt måste relevant historisk information finnas versionshanterad som referenspunkter för data i DW (t ex för dokumentation av vilken beräkningsregel som rådde för ett visst beräknat värde i DW vid dess etablering). Det kan t ex åstadkommas via tidstämpling.

Ett välfungerande repository är en vital ingrediens i och för strävan mot DW-data av god kvalitet.

8.2 Datakvalitet

8.2.1 Allmänt

Om användarna upplever otillräcklig kvalitet hos DW-data upphör snart användningen. Å andra sidan bör inte målsättningen vara kvalitet till varje pris. Att kvalitetshöja inkommande data till ett DW kan vara förenat med stora kostnader eftersom arbetet bara till liten del kan automatiseras. Att föra tillbaka bristerna till de ansvariga för källdatabaserna brukar snarare skapa irritation än rättarvilja. Den operativa miljön kan mycket väl ha funnit sätt att leva med bristerna, alternativt är de där inte lika kritiska som för DW-tillämpningen. Problemen kan dessutom vara en konsekvens av bristande överensstämmelse, missuppfattningar, osäkerheter, vid sammanföringen av data från olika källor – något som knappast källdatabaserna kan lastas för.

För vissa analyser kan ofullständigheter och inkorrekta uppgifter mycket väl accepteras medan de i andra situationer kan leda till katastrofala beslut. M a o bör alltid strävan vara att åstadkomma en kvalitet som svarar mot behoven, varken mer (kostnadsskäl) eller mindre (trovärdighet). Vilket låter sig sägas men i praktiken är en mycket svår balansakt.

Redan enkla data som namnfält i t ex kunddatabaser visar sig ofta vara en stor källa till problem. En och samma kund återfinns normalt som ett flertal poster p g a att namnet används som nyckel och stavats olika vid olika inmatningstillfällen. Det finns exempel på databaser där ett och samma företag registrerats i dussintals namnvariationer. Dels får man ingen samlad bild av kunden, dels fördyras olika former av kontaktskapande aktiviteter. Ta t ex ett postorderföretag som skickar ut flera miljoner kataloger. Det kan tjäna stora pengar enbart i tryck- och utskickskostnad genom att kunna rensa bort dubletter. Redan 20% reduktion av 5 miljoner registrerade kunder innebär 1 miljon färre utskick utan några som helst negativa konsekvenser.

Kvalitetshanteringen kring ett DW bör angripas inom en överordnad IT-strategi, inte i form av spontana "nödutryckningar".

Något som i de flesta fall har en avgörande inverkan på kvaliteten är stringensen i den semantiska betydelsen av lagrade data, användarnas möjligheter att uppfatta denna samt den korrekta tolkningen av den semantiska betydelsen i de olika källdatabaserna i samband med hämtningsfasen. Annars riskerar man blanda äpplen med bananer och kalla dem för päron.

8.2.2 Semantikproblem

Ett par exempel belyser förhoppningsvis semantikperspektivet.

Olika betydelser i olika databaser:

Ordersumma kan t ex betyda

- Belopp exklusive moms.
- Belopp inklusive moms.
- Belopp för ej momspliktig kund.
- Två belopp: det ena exkl, det andra inkl moms.
- Belopp för de artiklar som inte är restnoterade.
- Nettobelopp efter avdragen kundrabatt.
- Belopp enligt kundens uppfattning.
- Belopp enligt säljarens dokument.
- ...

Användningsberoende, d v s vara relevant endast under vissa förutsättningar:

Ordersumma finns endast

- vid order omfattande minst två orderrader.
- i samband med försäljning av varor per telefon.
- för order som ännu inte levererats i sin helhet.
- ...

Händelseberoende, d v s vara beroende av omständigheter:

Antal-i-lager betyder alternativt

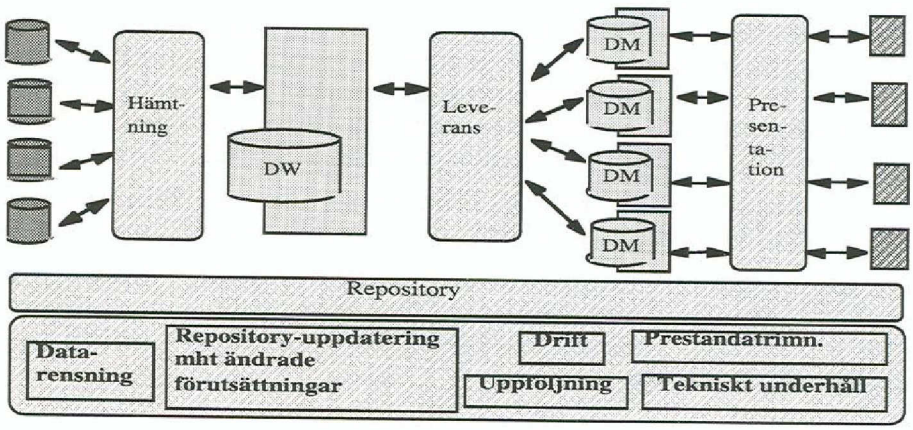
- Antal som just nu finns på sin plats i lagret.
- Enligt ovan men minus de som finns på kundorder.
- Enligt ovan men plus de som beställts för inleverans inom en vecka.
- Enligt ovan men minus en schablon för normalt svinn.
- Enligt ovan, men plus 25% för att få hem en kundorder i alla lägen. Leveransförseningar, klagomål m m fixas senare.
- .....

Semantik och semantisk integrering är mycket svårt. Det är ett verksamhetsrelaterat problem snarare än ett DW-problem. DW tvingar upp problemet till ytan. Är det så att full semantisk överblick är en orealistisk utopi eller är det bara så att vi hittills ägnat den för lite energi eftersom konsekvenserna inte upplevts allvarliga? Kommer motsvarande att gälla i framtiden? Är med andra ord ett totalgemensamt DW realiserbart? Är Data Marts eller begränsade DW, rimligare?

9. Administration, underhåll

Även med ett fungerande repository måste konventionellt drifts-, underhålls- och administrationsarbete utföras på samma sätt som vid alla typer av tillämpningar. Jämfört med en normal tillämpningsmiljö är den kompletta DW-arkitekturen mycket komplex med ett antal inblandade och samverkande databaser.

Olika typer av datorstöd för detta arbete måste finnas.



Figur 39

10. Marknad

10.1 Produkter

Det finns många olika DW-system på marknaden idag. Många fler kan påräknas framöver inom framförallt nya delområden såsom Data Mining. Få system är heltäckande över hela arkitekturen. I och med att kunderna kommer att efterfråga integrerade system i allt högre grad kommer fler leverantörer av heltäckande system att dyka upp samt systemleverantörer med ansvarstagande för integrering av olika produktkomponenter.

Än så länge kännetecknas området av en påtaglig turbulens. Nya företag kommer, andra försvinner eller blir uppköpta. På sikt kan en stabilisering förväntas inträda samt en koncentration till färre och större leverantörer med både produkter och tjänster att erbjuda. Alldeles påtagligt är de stora dbms-leverantörernas allt starkare intresse av detta marknadssegment. Här finns en stark expansion (medan den konventionella dbms-marknaden har en relativt blygsam sådan), här finns nya utmaningar kring prestanda, volymer, gränssnitt.

Det ännu begränsade intresset för standardisering och allmän samordning riskerar slå tillbaka på branschen som helhet. Kunder accepterar inte längre inlåsningar till en leverantör.

Att rada upp alla produkter på marknaden är knappast varken möjligt eller speciellt informativt. Instabiliteten är för stor. En presumtiv kund bör ägna ansenlig tid för nogsamt utvärderingsarbete i form av produktgenomgångar, uppföljning av referenser samt inte minst egna tester baserade på egna förutsättningar.

10.2 Typiska tillämpningsområden

Tillämpningsområden har traditionellt funnits i verksamheter med stora mängder kunder, transaktioner, händelser att hålla reda på och finna tilläggskunskap ur.

Bland typiska verksamheter återfinns

- Konsumentproduktleverantörer
- Detaljhandel, distribution
- Transport
- Banker
- Försäkringsbolag
- Telekommunikation
- Kommunala serviceorgan (El, VA, ...)
- Medicin, hälsovård
- Miljöorienterade aktiviteter
- Samhällsinformation

Bland typiska analyser kan nämnas

- Marknadsplanering
- Riskbedömningar
- Kundrelationer
- Inköps-, Försäljningsanalyser
- Lager
- Ekonomiska bedömningar
- Avancerad kunskaps-"grävning"

10.3 Diverse statistik

Följande statistik från olika källor ska ses mer som ungefärliga uppskattningar och trendsignaler än som precisa värden.

- Antalet Data Mart-installationer kommer att dubbleras från 1996 till 1997.
- 1997 beräknas ca hälften av alla DW-produkter vara avsedda för Data Marts.
- Antalet DW större än en terabyte växer 1997 från 4% till 12%. I ett något längre perspektiv kommer DW-volymerna regelmässigt att ligga över en terabyte med nästa dignitet i sikte.
- Antalet DW med fler än 500 användare ökar 1997 från 4% till 12%.
- Marknaden för Data Mining-produkter var 1996 ca 200 milj dollar.
- Internet-anpassad DW-teknologi kommer att vara norm om ca 1.5 år.
- Meta Group bedömer att den totala DW-marknaden ökar från ca USD 2 miljarder 1995 till ca USD 8 miljarder 1998.
- Gartner Groups motsvarande bedömning är:

  Total DW-marknad (Miljarder dollar)
  1994: 1.5
  1999: 6.9
  Därav mjukvara
  1994: 0.4
  1999: 3.0
  Därav hårdvara
  1994: 1.1
  1999: 3.9

11. Införande, risker

11.1 Införandestrategi

IT-områdets företrädare tenderar ofta att fascineras av produkter. Inköp bestäms innan ändamål och andra förutsättningar för deras produktiva användning analyserats i rimlig grad. Situationen är inte annorlunda inom DW. Om möjligt är lockelserna här större med löften om fantastiska kunskapskällor, som dessa härliga produkter likt Tingelings trollspö med sin magiska kraft introducerar för den oinvigde. Verkligheten är inte lika enkel. Den stora potentialen finns men kräver klokskap, ansvarsfullt förarbete och uthållighet. Värt att notera är att Gartner Group bedömer att ca 60 % av alla DW-initiativ misslyckats av olika skäl.

Den inledande fasen innebär

- Formulering av översiktlig strategi.
- Erhållande av acceptans och realistisk budget.
- Problem/behovsanalys.
- Utveckling av ett DW/DM för ett avgränsat, påtagligt behov.
- Utvärdering.

Om utfallet blir positivt fattas först därefter beslut om DW som en permanent, kontinuerlig och integrerad process i verksamheten. Nu måste en noggrann långsiktig strategi utarbetas baserad på de resultat Problem/behovsanalysen presenterat.

Inom Problem/behovsanalys-steget penetreras bland annat följande frågeställningar:

- Vilka problem avses DW lösa?
- Var finns 'förtjänsten' att hämta?
- Vilka är de tänkta användarna och varför?
- Finns erforderlig uppsättning källdata? Kvalitet?
- Är källdata tillgängligt med rimlig möda?
- Hur mycket historia ska lagras och varför? Hur länge?
- Finns tillräcklig användarkompetens, teknikkompetens?
- Kommer verksamhetens organisation, arbetsflöden, funktioner, m m att behöva anpassas? I så fall hur? Önskvärt?

Steget Utveckling av ett inledande DW innebär bl a:

- Precisering av
  - Vilka användare
  - Vilken information
  - Varifrån
  - Övriga villkor
- Definition av datamodell, transformeringsbehov.
- Definition, etablering av den erforderliga arkitekturen med hårdvara, mjukvara.
- Precisering av driftsroller, t ex
  - Beställare
  - Övergripande ansvarig
  - Dataägare
  - Driftsansvarig
- Implementering.
- Utbildning.
- Driftsättning.

11.2 Diverse råd, synpunkter

Allmänna råd uppsnappade på konferenser, i rapporter m m:

- Ha en välgenomtänkt plan.
- Gör en realistisk bedömning om genomförbarhet.
- Gör realistiska kostnadsanalyser.
- Skapa inte orealistiska förväntningar.
- DW bör syfta till att öka vinst snarare än sänka kostnader.
- "Architect globally, implement by increment" av vilket följer: Starta med ett mindre, välavgränsat, **välvalt** DW med påtagliga behov, intresserade användare och begränsade risker för övrigt.
- Begär garanti för en testperiod om minst ett år.
- Bevisa användbarhet, "lönsamhet". Lyckat resultat ger "ringar på vattnet".
- Ett positivt samarbete med och förståelse hos operativt ansvariga är en förutsättning. Dessa har sina separata ansvarsområden och kan knappast förväntas automatiskt lyckliga över en tillkommande komplexitetsfaktor. Kräver insäljning.
- Ursprungsbehoven är ofta mycket osäkra. Planera för expansionsmöjlighet. "Aptiten" vid lyckade DW kan öka dramatiskt i form av
  – Nya informationsbehov.
  – Mer avancerade informationsbehov.
  – Många fler användare.
- Undvik "Egensnickeri". Flexibla lösningar med standardkomponenter betalar sig i längden. "Egensnickeri" kan uppfattas som ett enkelt sätt att komma igång men bedöms allmänt som en vanlig faktor vid misslyckanden. Ofta finns här en teknikfokusering där någon tycker det är spännande – ett tag – att utveckla en ny "pryl". Verksamhetsförankringen är begränsad, det långsiktiga ansvarstagandet likaså. Nya skiftande krav och allmän expansion orkar sällan den egensnickrade lösningen svara upp mot.

DW-satsningar innebär sällan enkla, billiga lösningar. Lyckat utfall förutsätter ambitiösa, långsiktiga förpliktelser och ansvarstaganden. Det förutsätter ansenliga investeringar i både kunskap, verksamhetsanpassningar, produkter. Att helheten är en mycket komplex företeelse framgår bland annat genom det vida spektrum av personer som på olika sätt mer eller mindre permanent är inblandade både i de initiala faserna och därefter löpande. Bland aktuella kategorier kan nämnas:

- Data Warehouse-ansvarig (strategi, planer, styrning, ...)
- Dataarkitekt (behovsanalys, datamodellering, ...)
- Metadata-administratör (innehåll, hantering, åtkomst, ...)
- Databas-administratör (fysisk datastruktur, prestandatrimning, ...)
- Slutanvändaren (användning, behov, affärsregler, gränssnitt, ...)
- DW-systemerare/programmerare (realisering, underhåll, ...)
- Källdata-expert/ansvarig (semantik, avbildningar, ...)
- Operativt ansvarig (driftövervakning, ...)
- ....

Misslyckande kan orsakas av många faktorer. Vissa beror på verksamhetsspecifika omständigheter. Andra är av mer generell karaktär, t ex:

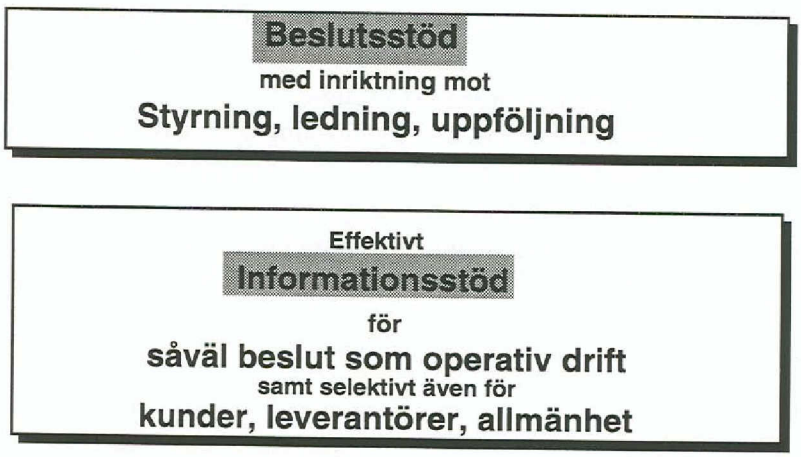
- Okunskap kring syfte.
  – Oklara budskap: "bättre beslut" i största allmänhet.
- Alltför hög ambitionsnivå.
  – Förhoppningar->Besvikelse->Stopp.
- Brist på kompetens hos såväl realiserare som användare.
- Teknikstyrt istället för användarstyrt.
- Bristande användargränssnitt.
- Avsaknad av en realistisk kostnads/intäktsanalys.
- För snäv budget.
- Brist på tillförlitliga källdata.
- Datasamordningsproblem – semantik.
- Otillräcklig uthållighet.
- Budgetbeviljare, användare brister i
  – intresse
  – uthållighet
  – förståelse.
- Ingen klar gräns mellan operativa och DW-data.
- Ingen klar ansvarsuppdelning.
- Inflexibel teknisk lösning. Prestandaproblem.

Kanske kan synpunkterna i detta avsnitt för den nyblivne DW-entusiasten generera en känsla av att stå inför ett närmast oöverstigligt problemberg. Detta har förstås inte varit syftet. Se inventeringen snarast som en uppmaning till en seriös hantering av en stödfunktionalitet som, rätt skött, på ett avgörande sätt kan bidra till en verksamhets stabila fortlevnad och framtidsutsikter.

12. Aktuella trender

Bland aktuella trender bör nämnas

- Mer integrerade arkitekturer.
- Vanliga rdbms
  - klarar generellt DW-kraven.
  - integrerar viss Data Mining-funktionalitet.
  - integrerar OLAP-funktionalitet.
- Mer DW-funktionalitet i vanliga OLTP-tillämpningar.
- Rikhaltigare innehåll. Dagens ofta numeriska innehåll kommer att kompletteras med allehanda övrigt normalt databasinnehåll.
- DW-volymer om ca 100 terabyte realitet år 2000. Sannolikt digniteter mer snart därefter när olika typer av multimediabaserade datatyper "tar plats" i DW.
- Bättre, mer integrerade
  - DW-administrationsverktyg.
  - Repositoryprodukter.
- Informationsagenter och Internet-miljö.
- Pro-aktiva DW.
- Samt en glidning i användning från



Figur 40

Med andra ord kommer antalet användare mot ett DW att närma sig vad som är normalt för vanliga databastillämpningar.

13. Sammanfattning

DW har utvecklats från en otydlig idé, över en liten nischmarknad, vidare över en kostnadsbesparande IT-teknik för beslutsstöd, till en strategisk affärsprocessmekanism. Antalet DW expanderar snabbt både vad gäller antal tillämpningar, antal användare och databasstorlek.

Användning av DW kommer att glida över från att vara ett mer eller mindre magiskt, hemligt vapen för att vinna fördelar gentemot konkurrenter till ett alldeles nödvändigt redskap och stöd i många verksamhetsorienterade processer och för extern informationsservice.

Nya expertroller kommer att växa fram. Kunskapsarbetaren kommer att ges en klarare profil. En betydligt mer avancerad databasadministratörsroll kommer att krävas där teknikkompetens måste paras med en höggradig användarserviceprofil.

Vi kan förvänta oss en intensiv utveckling kring Data Mining. Multimediadata i datalagret och i användargränssnittet kommer att stödjas allt bättre.

Internet/Intranet-baserad DW-åtkomst, browser-gränssnitt, applets för Data Mining ligger i närtid. Betydligt mer insamling av globalt tillgänglig verksamhetsextern information till det egna DW likaså. DW kommer successivt att integreras med övriga IT-miljöer i en verksamhet.

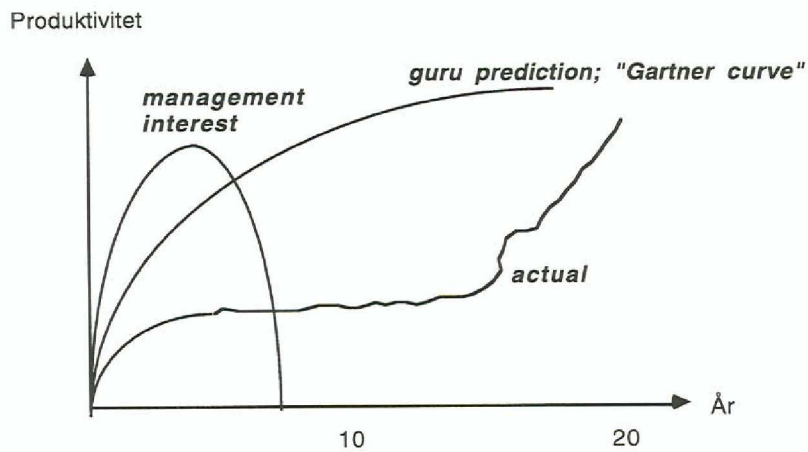
Både RDBMS- och OLAP-produkter kommer att finna sina typiska tillämpningsområden. DW kommer att vara en gyllene marknad för RDBMS-leverantörer. Den snabba expansionen kommer att ställa höga krav på effektiv hantering av mycket stora databaser, inte minst i och med att DW kommer att populeras med multimediabaserade data. Dessutom kommer en rik flora av användargränssnitt och olika typer av avancerade analysstöd att växa fram. SQL kommer att i än högre grad inskränkas till ett internt databasgränssnitt.

Produkter kommer att bli mer heltäckande och integrerade. Mer avancerad funktionalitet kan förväntas.

En av de mest karismatiska DW-företrädarna, Ken Orr, har placerat in DW i ett teknikmognadsperspektiv där

- fas 1 innebär en uppbyggnad av problemförståelse
- fas 2 innebär en successiv förståelse för vilka lösningsalternativ som är applicerbara
- fas 3 står för arbetet med att få det hela att fungera i verkligheten.

Han anser DW befinna sig någonstans i den senare delen av fas 2, något överlappande med de inledande stegen i fas 3. En mycket snabb fas 3-utveckling är att vänta, samtidigt som nya problem och lösningar dyker upp i de två tidigare faserna. Han konstaterar också de risker som alla nya teknologier är utsatta för, illustrerat i figur 41.

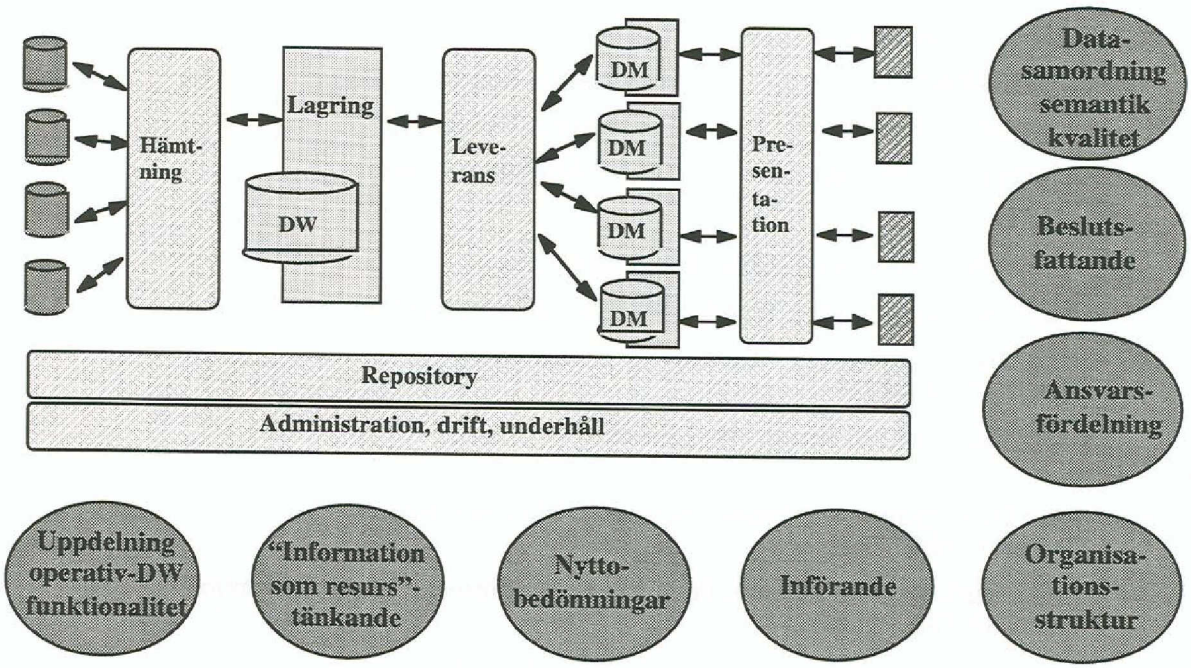


Figur 41

Prognosinstitut och "förståsigpåare" blåser upp förväntningar. De som ska fatta införandebeslut är på goda grunder skeptiska och tappar lätt intresset. I bästa fall överlever teknologin "riskzonen" och tar fart, dock efter betydligt längre tid än någon trott.

Den redovisade utvecklingstrenden har säkert hållit i historisk belysning. Dock bör varnas för att framförallt mer teknikbaserad utveckling numer tenderar att genomlöpa ett betydligt snabbare förlopp. DW-anpassad teknik kommer sannolikt att komma fram snabbt medan produktiv användning kommer att följa en betydligt svagare kurva.

DW är synnerligen mångfacetterat. Se figur 42. Detta innebär givetvis problem, risker men också utmaningar, möjligheter. Att lyckas är inte självklart. Den som gör det kan i allmänhet räkna med en mycket god förräntning.



Figur 42

Mer information

Ett antal tidskrifter bevakar DW-området. Bland dessa finns

Data Management Review

Application Development Trends

DBMS Magazine

En hel del kunskap finns också åtkomlig via Internet. Framförallt gäller detta produkt-specifikt material. Därutöver kan bl a följande allmänna web-sidor rekommenderas

The Data Warehousing Institute:	www.dw-institute.com
DW-vidarelänkar:	pwp.starnetinc.com/larryg/
OLAP-vidarelänkar:	www.olapcouncil.org/
Data Mining-vidarelänkar:	www.kdnuggets.com

Nya böcker tillkommer i strid ström. Bland dem finns

[1] Inmon: Building the Data Warehouse, Wiley&Sons, ISBN: 0-471-14161-5

[2] Kelly: Data Warehousing – the route to mass customisation, Wiley&Sons, ISBN 0-471-95082-3

[3] Poe: Building a Data Warehouse for Decision Support, Prentice Hall, ISBN 0-13-371121-8

[4] Bigus: Data Mining with Neural Networks, Mc Graw Hill, ISBN 0-07-005779-6

[5] Bischoff mfl: Data Warehouse: Practical Advice from the Experts, Prentice Hall, ISBN 0-13-577370-9.

[6] Kimball: The Data Warehouse Toolkit: Practical Techniques for Building Dimensional DataWarehouses, John Wiley Publishers.

[7] Devlin: Data Warehouse, from Architecture to Implementation, Addison-Wesley, ISBN 0-201-96425-2.

[8] Barquin, et al: Planning and Designing the Data Warehouse, Prentice Hall, ISBN 0-13-255746-0.