



Avdelning för datavetenskap

Daniel Strömberg

Analys av BI-system och utveckling av BI-applikationer

Analysis of BI-systems and development of BI-
applications

Examensarbete (20p)
Civilingenjörsprogrammet

Datum: 06-02-01
Handledare: Katarina Asplund
Examinator: Donald F. Ross
Ev. löpnummer: D2006:10

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en civilingenjörsexamen i informationsteknologi. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Daniel Strömberg

Godkänd, 060201

Opponenter: Fredrik Nilsson, Jonas Wånggren

Handledare: Katarina Asplund

Examinator: Donald F. Ross

Sammanfattning

Detta examensarbete på D-nivå beskriver skapandet av BI-applikationer och analys av tre olika BI-system. Projekten har givits av Cederroth International AB som är ett av Sveriges största företag inom dagligvaruhandeln. Uppsatsen ger en inblick i vad BI är och hur BI fungerar i företagsvärlden. Det ges även en stegvis beskrivning i hur en applikation skapas med hjälp av BI-systemet QlikView och analyser av BI-systemen QlikView, SQL Server 2005 och BusinessObjects XI om huruvida dessa uppfyller de funktionella kraven på ett BI-verktyg. Bakgrundsinformation om Cederroth och dess system för affärsdatahantering ges också i uppsatsen.

Den BI-applikation som beskrivs i denna uppsats involverar finsk återförsäljningsdata. Stegen från grunddata till färdig applikation beskrivs och förklaras ingående i uppsatsen. Analysen av de tre BI-systemen som därefter gjorts har beskrivits i en del där grundinformation följs av en analys med hänsyn till de funktionella aspekterna i BI-systemen. Denna analys resulterade i ett likvärdigt resultat för de tre BI-systemen. BI-systemen hade dock olika lösningar på de funktionella delarna.

Analysis of BI-systems and development of BI-applications

Abstract

This Master's thesis describes development of BI-applications and analysis of three different BI-systems. Cederroth International AB has given the projects, which is one of the leading perishable companies in Sweden. The thesis gives insight in what BI is and how BI works in the business world. It also gives a stepwise description of how to create an application in the BI-system QlikView, and gives an analysis of the BI-systems QlikView, SQL Server 2005 and BusinessObjects XI whether they fulfil the functional requirements in a BI-tool. Background information about Cederroth and their systems for business data management is also given.

The BI-application that is described in this thesis involves Finnish retail data. The steps from source data to ready application are thoroughly described and explained. The analysis of the three BI-systems is described in a section where the core information about BI-systems is followed by an analysis with the functional aspects of a BI-system taken in consideration. This analysis resulted in an equivalent result for the three BI-systems. However the BI-systems had different solutions regarding the functional parts.

Innehållsförteckning

1	Inledning	1
1.1	Cederroth	2
1.2	Projektintroduktion	2
1.3	Disposition	3
2	Bakgrund	4
2.1	Business Intelligence	4
2.2	Cederroths datahantering	6
	2.2.1 AS/400	
	2.2.2 Movex	
	2.2.3 BI-system	
2.3	Open Database Connectivity (ODBC).....	10
2.4	Active Directory (AD).....	10
2.5	QlikView.....	11
	2.5.1 Associativa Databaser	
	2.5.2 Datakällor	
	2.5.3 Associative Query Logic	
	2.5.4 Användargrupper	
	2.5.5 Säkerhet	
	2.5.6 Att skapa en applikation	
	2.5.7 List Box	
2.6	Microsoft Business Intelligence (SQL Server 2005).....	18
	2.6.1 Business Intelligence Development Studio	
	2.6.2 Project mode	
	2.6.3 Lösningar	
	2.6.4 Element i BI Development Studio-miljön	
	2.6.5 Spridning	
	2.6.6 Konfigurering	
	2.6.7 Lagutveckling och återanvändning	
	2.6.8 Online-projekt	
2.7	BusinessObjects XI.....	24
	2.7.1 Användarbehov	
	2.7.2 Prestandahantering	
	2.7.3 Rapportering	
	2.7.4 Förfrågan och Analys	
	2.7.5 BI-plattform	
	2.7.6 Dataintegration	
2.8	Sammanfattning	31

3	Utveckling den finska QlikView-applikationen.....	33
3.1	Introduktion	33
3.2	Det finska POS-data projektet	36
3.3	Kravspecifikation.....	36
3.4	Lokalisering av data.....	37
3.5	Implementation av skript	41
3.6	Definition av grafiskt användargränssnitt.....	50
3.7	Testning	63
3.7.1	Överensstämmelse mellan presenterad data och grunddata	
3.7.2	Test av koppling Movex-Excel	
3.7.3	Testning av sammanlänkning mellan ark	
3.7.4	Användartillgång till grafiska objekt	
3.7.5	Uppdateringstest	
3.7.6	Webbgränssnitt vs originalapplikation	
3.7.7	Testsammanfattning	
3.8	Säkerhetskonfigurering.....	68
3.9	Distribution	69
3.10	Utvärdering	70
3.11	Sammanfattning	70
4	Analys av Business Intelligence system	71
4.1	Introduktion	71
4.2	Funktionella möjligheter.....	72
4.2.1	Databas och datalager	
4.2.2	Dataintegration	
4.2.3	Rapportering	
4.2.4	OLAP	
4.2.5	Analys	
4.2.6	Instrumentbrädor	
4.2.7	Data mining	
4.2.8	Användarprofiler	
4.3	Infrastrukturskrav	83
4.3.1	Arkitekturstandarder	
4.3.2	Ledningsverktyg	
4.4	Försäljarkriterier	84
4.4.1	Finansiell stabilitet	
4.4.2	Industrianalysutvärdering	
4.4.3	Världsomspännande service och support	
4.4.4	Återförsäljares ”ekosystem”	
4.4.5	Relation till andra standarder	
4.5	Sammanfattning	85
5	Resultat & slutsats.....	87
5.1	Resultat	87
5.2	Slutsats	88

Referenser	90
A Ordlista.....	92
B Källkod	97

Figurförteckning

Figur 2-1: Egenskaper hos AS/400 [1].....	7
Figur 2-2: Skärskådning av Movex [1].....	9
Figur 2-3: Active Directorys användningsområden [28].....	11
Figur 2-4: Användargruppstabell från QlikView [17]	15
Figur 2-5: Skapandet av en QlikView-applikation	16
Figur 2-6: Traditionell databas tabell	17
Figur 2-7: Listboxar med samma information som Figur 2.6	17
Figur 2-8: Val i en listbox	18
Figur 2-9: Business Intelligence Development Studio [26].....	21
Figur 2-10: BusinessObjects XI [9].....	25
Figur 2-11: BI-krav och BusinessObjects XI:s lösningar [9]	26
Figur 2-12: BusinessObjects Enterprise användningsområden [9].....	29
Figur 2-13: Business Objects miljöförenkling [9]	30
Figur 2-14: BusinessObjects XI:s BI-prestanda.....	31
Figur 3-1: Projektflöde	35
Figure 3-2: POS-data	38
Figur 3-3: Fält från Movex.....	40
Figur 3-4: Movex-filernas sammanlänkning	48
Figur 3-5: QlikView-applikationens struktur	49
Figur 3-6: Finsk återförsäljningsdata	51
Figur 3-7: Exempeltest av Excel-datakopplingar.....	51
Figur 3-8: Affärsdata från Movex	52
Figur 3-9: Exempeltest på Movex-datakopplingar	53
Figur 3-10: Valmöjligheter i grafisktobjekt	54
Figur 3-11: Snabbvalsikoner	54
Figur 3-12: Stapeldiagram	55
Figur 3-13: Tabell.....	55

Figur 3-14: Gallerdiagram	56
Figur 3-15: Stapeldiagram med månadsintervall	56
Figur 3-16: Slutvy ark 1.....	57
Figur 3-17: Slutvy ark 2.....	58
Figur 3-18: Slutvy ark 3.....	58
Figur 3-19: Slutvy ark 4.....	59
Figur 3-20: Slutvy ark 5.....	59
Figur 3-21: Summa av försäljning €.....	61
Figur 3-22: Summa av försäljning €per butikskedja.....	61
Figur 3-23: Summa försäljningskvantitet i konsumentenhet per produkt.....	62
Figur 3-24: Summa försäljning per affär/månad.....	62
Figur 3-25: Försäljningsinformation.....	62
Figur 3-26: Test-förfrågan i QlikView (ark 3)	64
Figur 3-27: Data från Excel.....	64
Figur 3-28: Movex-data ark 2	65
Figur 3-29: Testexempel ark 4	66
Figur 3-30: Testexempel ark 5	67
Figur 3-31: Startvy för webbgränssnittet	69
Figur 4-1: Funktionella krav och standarder [10]	73
Figur 5-1: Resultat av analys (missvisande)	73

Tabellförteckning

Tabell 2.1: SQL Server 2005:s BI-komponenter	18
Tabell 3-1: Inkommande POS-data.....	39
Tabell 3-2: POS-data från finska återförsäljare	43

Förkortningar

AD	Active Directory
AS	Analysis Services
AQL	Associative Query Logic
API	Application Program Interface
BI	Business Intelligence
CRM	Customer Relationship Management
DBMS	Database Management System
DNS	Domain Name System
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
XML	Extensible Markup Language
ETL	Extract, Transform, and Load
XP	Extreme Programming
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTP	Hyper Text Transfer Protocol
KPI	Key Performance Indicator
KSI	Key Success Indicators
KDD	Knowledge-Discovery in Databases
KXEN	Knowledge eXtraction Engines
LDAP	Lightweight Directory Access Protocol
OLAP	Online Analytical Processing
OLEDB	Object Linking and Embedding Database
ODBC	Open Database Connectivity
POS	Point Of Sales
RDBMS	Relational Database Management Systems
SQL	Standard Query Language
SCM	Supply Chain Management

1 Inledning

Huruvida man jobbar för ett nystartat företag eller ett 100 år gammalt bolag så är data kärnan i organisationen. Det finns många olika sorters data: finansiell data, produktdata, kunddata, leverantörsdata, konkurrensdata osv. Dessa data sparas på många olika sätt, t.ex. i databaser, i Word-dokument, på webbsidor etc. Hur all denna data skall kunna presenteras på ett uniformt sätt är ofta ett problem för organisationen.

Olika mjukvarusystem och produkter används för att samla ihop och använda de olika sorters data som gör det möjligt för affärsverksamheten att fungera friktionsfritt. Datasystem fokuserar på att samla transaktionsdata som uppstår varje affärsdag i företaget. Anställda skapar dokument, memos och rapporter som de vill dela med andra för att förmedla affärsdata. Online analytical processing (OLAP), data mining, ad hoc-förfrågan och rapportering, data warehousing (datalagring), ERP/CRM system är några exempel på olika sätt att spara och hantera samt arbeta med affärsdata¹.

Ett företags affärsmål och budget bör vara framtagen ur den data som finns sedan tidigare för att de skall vara verklighetsrelaterade. Mål och budget är även ett måste för att ett företag ska kunna fungera i dagens affärsklimat. Detta gör indirekt att affärsdata är livsviktigt för företagen. Att göra responstiden för hämtning, analys och presentation så liten som möjligt är ekvivalent med att göra företaget så effektivt som möjligt. Ett effektivt företag är generellt sett ett vinstdrivande företag. Affärsmål och budget är dock inte den enda vägen till framgång. Företag är även beroende av de beslut som dagligen fattas på alla nivåer inom organisationen. Genom att förse så många medarbetare som möjligt inom företaget med ett brett beslutsunderlag så skapas bättre förutsättningar för en lyckad affärsverksamhet.

Vad som helt klart kan sägas är att hantering av affärsdata är en av de viktigaste delarna i en affärsverksamhet. Företag som ej har en fungerande hantering av affärsdata är i allmänhet de företag som går en mörk framtid tillmötes. Det har även framkommit att det finns en hel del olika aktiviteter som gör det möjligt att arbeta med affärsdata.

¹ Dessa begrepp och övriga begrepp som inte använts direkt i applikationsskapandet eller analysen förklaras i ordlistan i bilaga A.

Finns det då någon väg att hantera affärsdata på ett effektivt sätt utan att bli tvungen att arbeta med en mängd olika processer? Fler företag har insett att ett bra alternativ är Business Intelligence (BI) (se sektion 2.1).

1.1 Cederroth

Cederroth är ett företag med en omfattande bredd av hälsovårds-, kroppsvårds-, förstahjälpen- och hushållsprodukter. Inom varje produktområde har företaget ett antal väldigt framgångsrika märken, vilket medför en solid position på en högst konkurrenskraftig marknad. Cederroth har 850 anställda och opererar i 30 länder inom Europa. Vid sidan om dess helt egenägda dotterbolag i åtta av dessa länder representeras Cederroth i ytterligare 22 länder via lokala distributörer. Sverige är den absolut största marknaden och representerar nästintill halva den årliga omsättningen. De senaste och snabbast växande marknaderna är i Östeuropa.

1.2 Projektintroduktion

Projektet som beskrivs i denna uppsats utfördes i två delmoment: det första bestod av att ta fram data ur serversystemet och presentera detta på ett relevant sätt och det andra momentet inbegrep en analys av tre olika BI-system.

Cederroth International som är arbetsgivaren använde sig av verktyget QlikView (se sektion 2.5) för hämtning och presentation av data. Enligt Cederroth så fanns det en mängd valmöjligheter när det gällde data som behövde presenteras. Efter en diskussion med IT-chefen så togs en möjlig arbetsplan fram. Upplägget på arbetsplanen var att först börja med några mindre projekt för att sedan successivt gå mot utvecklande av större projekt. Anledningen till detta var att kunna hålla en rimlig tidsplan med tanke på tidsbegränsningen på 20 veckor. Denna trappstegsliknande process gjorde det samtidigt enklare att lära sig de system och den datahantering som företaget använder sig av. Dessutom minimerades även risken för att projektet skulle misslyckas totalt. Projekten varierade dock i prioritet vilket kunde medföra att det projekt som påbörjades först inte var givet det som avslutades först. Hänsyn var även tvungen att tas till huruvida klienten, dvs. avdelningen som är berörd, hade möjlighet och tid att svara på förfrågningar angående applikationsskapandet. Det projekt som lades mest tyngdpunkt på var ett finskt POS-data projekt (se sektion 3.4), som den finska

divisionen på Cederroth för tillfället var i behov av. Detta projekt slutfördes och har även dokumenterats i denna uppsats.

Det andra delmomentet var mer av ett forskande slag, där tre olika BI-system skulle jämföras. Denna utvärdering användes sedan som feedback till företaget gällande dess nuvarande BI-system. De system som utvärderades och jämfördes var QlikView (se sektion 2.5), Microsoft Business Intelligence Development Studio (se sektion 2.6) samt BusinessObjects XI (se sektion 2.7). Aspekterna som togs hänsyn till var de funktionella aspekterna hos ett BI-verktyg. De viktigaste resultaten var i mitt tycke att de tre BI-systemen hade de funktionella aspekter som kan krävas från ett BI-system. Skillnaden var att de hanterade funktionaliteten på olika sätt.

1.3 Disposition

Uppsatsen inleds med ett bakgrundskapitel, kapitel 2, som först ger en introduktion till Business Intelligence. I samma kapitel beskrivs sedan den datahantering Cederroth använder för den data som berörs i denna uppsats. Efter det beskrivs två verktyg som använts när QlikView-applikationen skapats. Den sista delen i kapitel 2 ger en djupare genomgång i de BI-verktyg som i senare kapitel används för applikationsskapande och analys.

Kapitel 3 beskriver sedan processen i applikationsskapandet. Sektion 3.1 beskriver allmänt de steg som krävs för utveckling i berört BI-verktyg samt de steg som Cederroth kräver för projektet. Dessa två olika varianter sätts sedan ihop vilket skapar ett generellt projektflöde. Följande sektioner beskriver det huvudprojekt som tilldelats och följaktligen stegvis presentation av hur och på vilket sätt projektet utförts.

Kapitel 4 innehåller analyser och jämförelser av de tre BI-system som valts ut. I den inledande sektionen så beskrivs de aspekter som är viktiga att ta hänsyn till och därefter så betas dessa av sektionsvis.

Kapitel 5 delger läsaren de resultat som kommit fram under examensarbetet och även de slutsatser som jag har tagit.

2 Bakgrund

I detta kapitel beskrivs verktyg och information som använts när uppsatsen skrivits. Den första delen, sektion 2.1, ger en bakgrund och förklaring till begreppet Business Intelligence. I sektion 2.2 beskrivs Cederroths datahantering som berör denna uppsats. Efter det beskrivs i sektion 2.3 och 2.4 två verktyg som använts under projektet och som involveras i uppsatsen. Sektion 2.5-2.7 ger en djupare inblick i de BI-system som skall analyseras och det som skall användas för applikationsutveckling. I sektion 2.8 ges slutligen en sammanfattning av bakgrundskapitlet.

2.1 Business Intelligence

En tidig referens till icke-business intelligence förekommer i Sun Tzus The Art of War [14]. Sun Tzu hävdar att för att lyckas i krig ska det finnas full kunskap om ens egna styrkor och svagheter och även full kunskap om fiendens styrkor och svagheter. En brist i endera kan resultera i förlust. En viss skola av tankesätt [38] drar paralleller mellan utmaningar i affärer och utmaningar i krig, specifikt:

- Insamling av data
- Bedömning av mönster och innebörd i data
- Respons på den resulterande informationen

Före starten av ”informationsåldern” i slutet av 1900-talet, gjorde sig företag ofta besväret att samla in data från icke-automatiserade källor. Företagsverksamheten saknade då beräkningsresurserna för att göra en korrekt analys av data. Ofta gjordes också kommersiella beslut primärt på intuitionsbasis.

Allteftersom affärsverksamheter startade att automatisera fler affärssystem så resulterade detta i att mer data blev tillgänglig. Dock så kvarstod utmaningen gällande insamling, detta beroende på saknad av infrastruktur för datautbyte eller inkompatibilitet mellan affärssystemen. Rapporter gjorda på insamlad data tog ibland månader att generera. Sådana

rapporter tillät underrättad långsiktigt strategiskt beslutskapande. Kortsiktigt taktiskt beslutskapande fortsatte likväl att baseras på intuition.

I dagens företagsverksamhet har standardökningen, automatiseringen och nya teknologier lett till att en väldig mängd användbar data har blivit tillgänglig. Datalagerteknologier har satt upp förvaringsplatser för att lagra data. Förbättrad extrahering, transformering och laddning (ETL) och de senaste integreringsverktygen för företagsapplikationer (EAI) har ökat hastigheten vid insamling av data. OLAP-teknologier för rapportering har tillåtit snabbare generering av nya rapporter som analyserar data.

1989 populariserade Howard Dresner från Gartner-gruppen "BI" som en standardterm för att beskriva en grupp av koncept och metoder för att förbättra affärsbeslutstagande genom att använda ett faktabaserat hjälpsystem. BI har numera blivit konsten att sälla igenom stora mängder av data, extrahera information och sedan överföra den informationen till funktionell kunskap. BI [36] är även en bred kategori av applikationsprogram och teknologier för att samla, spara, analysera och tillhandahålla access till data för att hjälpa företagsanvändare till bättre affärsbeslut. BI-applikationer inkluderar bland annat aktiviteter som beslutshjälp, förfrågan och rapportering, online analytical processing (OLAP), statistisk analys, framtida marknadsanalys, och data mining.

Organisationer samlar information från BI-processerna för att analysera affärsomgivningen och täcka fält som marknadsföringsundersökning, industri och marknadsundersökning samt konkurrentanalys. Konkurrerande organisationer ackumulerar BI för att dra nytta av en hållbar konkurrensfördel. BI kan även ses som värdefull kärnkompetens inom vissa instanser.

Personer som är involverade i BI-processer kan använda applikationsmjukvara och andra teknologier för att samla, lagra, analysera och tillhandahålla access till data. Vissa observatörer [38] anser att BI är en process för att förhöja data till information och sedan till kunskap. Målet för mjukvaran är att hjälpa användaren till bättre affärsbeslut genom att göra exakt, aktuell och relevant information tillgänglig för dem när de behöver den.

Generellt så plockar BI-uppsamlare sin primära information från interna affärskällor. Dessa källor hjälper beslutstagare att få insikt i hur bra de har presterat. Kundbehov, kundbeslutsprocesser, konkurrens och konkurrenskraftighetstryck, tillstånd i relevanta

industrier samt generella ekonomiska, teknologiska och kulturella trender ses som andrahandskällor av information.

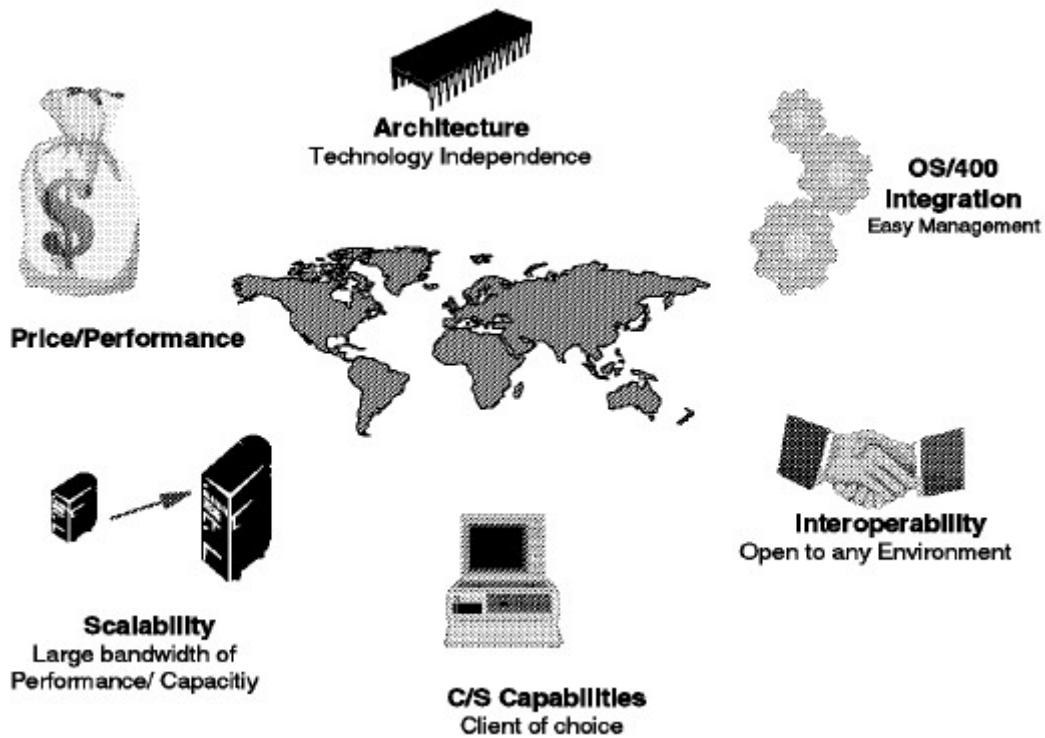
Varje BI-system har ett specifikt mål, vilket härleds från ett organisationsmål eller en visionsutsaga. Bland dessa mål existerar både långsiktiga och kortsiktiga mål vilket även inkluderas i BI-systemet.

2.2 Cederroths datahantering

Cederroths anställda arbetar dagligen med affärsdata, därför är det viktigt att ha ett fungerande system för datahantering. Den del av Cederroths datahantering som är relevant för denna uppsats är framförallt affärsdata som lagras i Movex-systemet på en AS/400 (se nedan) och sedermera presenteras mha QlikView.

2.2.1 AS/400

AS/400-systemet [1] har en lång världsomfattande historia. Det finns mer än 600 000 AS/400 system installerade i över 150 länder. Det finns också mer än 30 000 affärsapplikationer runt om i världen som använder sig av AS/400. *Figur 2-1* visar några egenskaper vilka kan ses som anledningar till detta. Dessa egenskaper ges även en kortare förklaring efter figuren.



Figur 2-1: Egenskaper hos AS/400 [1]

Arkitektur (Architecture)

AS/400 har en lagerbaserad arkitektur som är uppdelad i ett aktuellt användarinterface (OS/400) och ett teknologioberoende maskingränssnitt (TIMI). Detta arkitekturiska koncept har tillåtit AS/400-systemet att klara av ett flertal fundamentala teknologiförändringar. På detta sätt har kundinvesteringar i informationsteknologi skyddats.

Integration

AS/400-system erbjuder integration av både hårdvara och mjukvarukomponenter. Hårdvara, mikrokod, operativsystem samt IBM-mellanvara är alla hårt sammanflätade för att tillåta maximalt utnyttjande av alla beräkningresurser.

Anpassningsbarhet (Interoperability)

AS/400-systemet erbjuder en stor spännvidd av kommunikationsmöjligheter och funktioner, vilket tillåter AS/400 att kommunicera med både IBM-system och icke-IBM-system.

Klient och serverkapacitet (C/S Capabilities)

AS/400-systemet fungerar med nästan alla klienter i varierande kommunikationsmiljöer. Den kan bindas till följande klienter:

- Microsoft Windows 3.1, Windows 3.11, Windows 95, Windows 98, Windows NT OS/2, Windows XP
- MacIntosh
- UNIX (IBM-AIX, HP-UX, SUN-SPARC)
- Linux

Skalbarhet (Scalability)

AS/400-serverar täcker ett stort område av prestandakapacitet. Med detta menas att en server kan understödja allt ifrån några få användare till tusentals användare.

De flesta AS/400-servermodeller är möjliga att uppgradera till mer kraftfulla modeller, vilket blir ett sorts investeringsskydd för kunden. Denna möjlighet är även en bidragande faktor till AS/400-systemets låga kostnad i förhållande till dess livslängd.

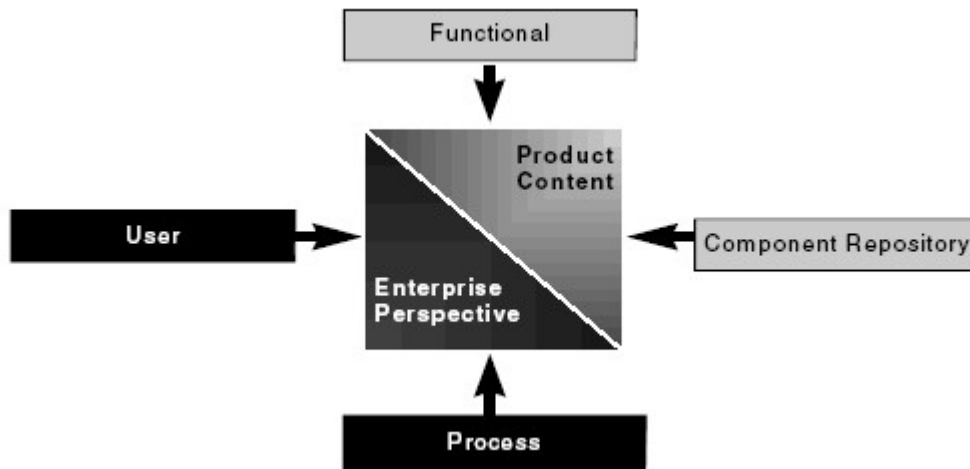
Pris och prestanda (Price/Performance)

Många oberoende analyser[1] har bekräftat att AS/400-systemet i längden representerar en kostnadseffektiv plattform. AS/400-systemets vida integration skänker signifikanta kostnadsfördelar, hög tillgänglighet, enkel systemhantering samt investeringsskydd. AS/400-system delar ut hög beräkningsprestanda till en låg kostnad för ägandeskap vilket oftast leder till nöjda kunder.

2.2.2 Movex

Movex är ett affärssystem som Cederroth använder för lagring och hantering av affärsdata. Ett affärssystem är ett programpaket som uppfyller krav från användare inom företaget. Dessa krav grundas på de affärsprocesser som användaren arbetar med. Affärssystemet använder en gemensam databas för företagets data för att undvika dubbellagring. De funktioner som tillhandahålls är väl anpassade efter företagets behov vilket i Cederroths fall är produkthantering, ekonomisk redovisning, kundorderbehandling, fakturahantering, produktionsplanering och personaladministration.

Movex-systemet kan skärskådas från flera olika synvinklar, vilket visas i *figur 2-2* och sedan förklaras.



Figur 2-2: Skärskådning av Movex [1]

Funktionella (Functional)

Den sedvanliga presentationen av Movex som beskriver den funktionella kapaciteten i affärssystemet. Funktionskapaciteten som finns i Cederroths Movex-system är anpassad efter de behov som användarna har. Med detta menas att Movex i detta fall är anpassat efter Cederroths inköp, tillverkning och försäljning.

Komponentförrådet (Component Repository)

Komponentförrådet är Movex interna organisation och arkitektur. Komponentförrådet tjänar som bas för mappning av Movexapplikations-komponenter till affärsprocessen i Enterprise Process Manager (EPM). EPM hanterar processdesign och applikationskonfiguration i Movex.

Kärnan i AS/400-menyerna är förenad med komponentförrådet. Med detta menas att det bara finns en instans av varje program och fil. Dessa program och filer är i sin tur kopplade till en komponentgrupp. Komponentgrupper används för att förenkla navigeringen inuti Movex.

Process

Process refererar till affärsprocesser som mappas ut i EPM och kan ses som lösningen till ett implementerat projekt. Den allmänna processmodellen används som riktlinje för presentation

i Movex och fungerar som en startpunkt för marknadsorienterad publicering. Processen styr till exempel hela vägen från försäljning av vara till förändring i lagersaldo och ny produktion.

Användare (User)

Det här perspektivet hänvisar till specialanpassade Movex-menyer. Det vill säga menyer som är anpassade utefter en specifik användares behov.

2.2.3 BI-system

Cederroth har valt att använda QlickView (se sektion 2.5) som verktyg när det gäller presentation och analys av data från dess databaser. QlickView i sig själv är ett BI-system.

2.3 Open Database Connectivity (ODBC)

För att kunna plocka data ur ett serversystem så behövs en eller flera ODBC-kopplingar. ODBC [11] är en standard som tillhandahåller ett programmeringsgränssnitt (API) som tillåter klientsidans program att anropa databashanteringssystemet (DBHS). Så länge som både klienten och servermaskinen har nödvändig mjukvara installerad så fungerar denna uppkoppling. De flesta DBHS-försäljare förser sina kunder med ODBC-drivrutiner för sina system. Följaktligen kan ett klientprogram kopplas till ett flertal relationsdatabashanteringssystem (RDBHS) och skicka förfrågnings- och transaktionsbegäran genom att använda ODBC, vilket sedan behandlas på serversidan. Förfrågningsresultaten skickas sedan tillbaka till klientprogrammet som sedermera kan behandla eller uppvisa det resultat som behövdes.

2.4 Active Directory (AD)

Active Directory [28] är en registerservice. Termen registerservice refererar till två saker – dels ett register där information om användare och resurser är lagrat och dels en tjänst som ger åtkomst för manipulering av dessa resurser. AD är ett sätt att hantera alla element i nätverket inklusive datorer, grupper, användare, domäner, säkerhetspolicys samt andra typer av användardefinierade objekt.

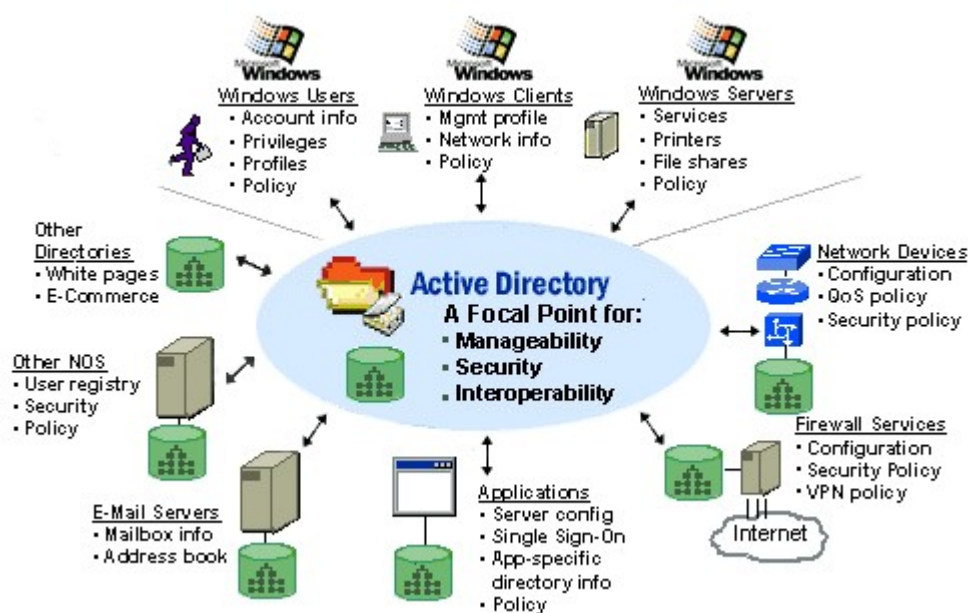
Active Directory är uppbyggt runt DNS [30] och lightweight directory access protocol (LDAP) [15]. Det använder sig av DNS för att det är en öppen och bekant standard. LDAP används för att de flesta försäljarna stödjer det [28]. AD-klienter använder DNS och LDAP

för att hitta och få tillgång till resurser på nätverket. Eftersom dessa är plattformsoberoende protokoll kan Unix, Macintosh och andra klienter få tillgång till resurser på samma sätt som Windows klienter.

De två viktigast målen med AD är att:

- Användare efter inloggning i domänen ska ha möjlighet att få åtkomst till resurser.
- Administratörer ska kunna hantera användare och resurser centralt.

Figur 2-3 visar ovanstående mål samt övrig funktionalitet i AD.



Figur 2-3: Active Directorys användningsområden [28]

2.5 QlikView

QlikTech International AB [37] är ett privatägt mjukvaruföretag som tillhandahåller mjukvaran QlikView [17] för analys av data. Företaget har anställda fördelade på det globala huvudkontoret i Lund, det nordamerikanska huvudkontoret i Raleigh, North Carolina, samt säljkontoren i Stockholm, Göteborg, Köpenhamn, New York, Minneapolis, Birmingham och Düsseldorf.

Lösningarna säljs såväl direkt av QlikTech som genom ett nätverk av fler än 100 certifierade återförsäljare och applikationsutvecklare i Europa, Nordamerika, Sydamerika och Asien. QlikTech startades 1993 och har hittills sålt fler än 90 000 licenser i mer än 40 länder.

QlikTech tillhandahåller en mjukvara för interaktiv dataanalys, vilket möjliggör för företag och dess ledning att effektivisera och optimera sina affärer. QlikTechs huvudprodukt QlikView nyttjar teknologin AQL (se sektion 2.5.3) för att leverera en flexibel analys och rapporteringslösning som samtidigt är snabb och kostnadseffektiv.

QlikView är ett BI-system som erbjuder olika perspektiv på databasapplikationer. Systemet agerar som ett användarfokuserat komplement till existerande databaser. Datainförsel och datauppdatering måste fortfarande genomföras emot en traditionell databas. QlikView extraherar sedan data och bygger en associativ databas som intensifierar sättet som användare arbetar med informationen.

QlikView är designat för att hantera en stor mängd data. Det finns i stort sätt ingen gräns för antalet tabeller som är tillåtna i en applikation. Dessutom finns det ingen gräns för antalet fält, rader eller celler i en tabell. Den enda gränsen som är införd av QlikView är att ett fält inte kan innehålla mer än två miljarder distinkta värden. Förutom det så är det bara internminnet som sätter gräns för storleken på en QlikView-applikation.

Att jämföra storleken på en relationsdatabas med systemets RAM-minne för att bestämma om applikationen är lämplig för QlikView är inte enkelt. När QlikView laddar data från en databas så optimeras den, vanligtvis resulterar detta i att storleken på en QlikView-applikation bara blir 10-30% av originalkällans. Detta kan ses som att med ett system på 64MB RAM kan en 256MB relationsdatabas laddas till QlikView.

2.5.1 Associativa Databaser

De flesta traditionella databaserna är uppbyggda på relationsmodellen. Lagrade tabeller bryts isär för att minska redundans och nyckelfält används för att förena tabellerna när de ska användas. Databasprogrammerare behöver göra kompromisser när det gäller ökad hastighet på bekostnad av mer utrymme och längre tidsåtgång vid addering och redigering av tabeller. QlikView skapar en associativ databas när data laddas från datakällan. Fördelen med detta är att den associativa databasen opererar mycket snabbare och kräver betydligt mindre utrymme än en ekvivalent relationsdatabas. Den associativa databasen är designad så att databas-användaren får maximal flexibilitet och information vid arbete med databasen.

QlikViews associativa databas optimerar data medan den laddas. Redundanta datafält tas bort och tabeller blir automatiskt sammanlänkade. Indexering behövs inte eftersom varje fält är tillgängligt som sökfält utan prestandaförsämring.

En associativ datamodell [35] består av objekt och associationer. Objekten existerar oberoende av varandra medan associationerna beskriver relationen mellan objekten. Till exempel är personer och företag objekt som existerar oberoende av varandra, medan anställd är en association mellan en person och ett företag. Begreppet anställd kan uppenbarligen inte existera utan företaget och personen. Skillnaden i det här exemplet jämfört med en relationsdatabas är marginell. Ta däremot ett exempel med datastudenter på ett universitet där ett register över vilka färdigheter som studenterna har underlättar vid antagning till kurs. I en relationsdatabas behövs minst tre tabeller: en med studenter, en med färdigheter och en med vilka studenter som har vilka färdigheter. En associativ databas behöver bara två typer av objekt, studenter och färdigheter samt en association ”har kunskap i”. Den objektorienterade konstruktionen förenklar definition av nya undertyper som ärver grundtypens egenskaper. Färdigheter kan således ha undertyper som språk (engelska, tyska, svenska) och programmeringsspråk (Objektorienterade, funktionella etc.). Associationen ”har kunskap i” som definierats för färdigheter finns då automatiskt tillgänglig för alla undertyper. Associationer är dubbelriktade, vilket underlättar om man vill veta vilka studenter som har kunskap inom till exempel C++ (undertyp till objektorienterade). I detta fall är det bara klicka på C++ i klientapplikationen. Alltså krävs det inget jobb med att definiera nya index eller skriva SELECT-satser för att associationerna skall gälla.

2.5.2 Datakällor

QlikView [17] har möjlighet att ladda data från följande datakällor:

- Avgränsade textfiler med något enkeltecken som avgränsare
- ”Fixed record”- textfiler med fixerad fältbredd
- AS/400 Data Interchange Format (DIF)
- Microsoft Excel-blad
- HTML-tabeller från hemsidor
- XML-tabeller
- QlikViews associative databasfiler

- ODBC-möjliggjorda datakällor
- OLEDB-möjliggjorda data källor

QlikView inkluderar möjligheten att ladda från dessa källor (utom ODBC/OLEDB) genom att använda HTTP [4] eller FTP [5]. Med detta menas att filkällan kan residera på en web server eller FTP server på en avlägsen plats. QlikView kan direkt koppla upp mot servern och ladda datakällan.

2.5.3 Associative Query Logic

QlikView baseras på en teknologi som kallas Associative Query Logic (AQL) [17]. AQL möjliggör för QlikView att arbeta med miljontals celler av data och samtidigt svara på förfrågningar på mindre än en sekund. AQL-teknologin bygger upp och upprätthåller en associativ och utrymmeseffektiv databas som residerar i RAM-minnet. Datakällan är bevarad och direkt tillgänglig off-line för analys, hela vägen ner till ursprungstransaktionen.

Genom AQL så eliminerar QlikView behovet av datakuber och datalager². Kubstrukturen byts ut mot en datastruktur som går under benämningen Data Cloud (datamoln). Datamolnet behöver inte innehålla någon tidigare förenad data utan bygger istället på icke redundanta tabeller som hålls i minnet under hela processen. Förfrågningar skapas snabbt och körs mot datamolnets minnesbank.

Med AQL [31] så sparas all data bara en gång och alla dataassociationer är sparade som pekare. Detta gör att datamolnsdatabasen blir mer effektiv vid mottagande av register i förhållande till en OLAP-databas. Datamolnsdatabasen är också mycket mindre eftersom register inte behöver repeteras genom förening och dess struktur behöver aldrig ändras. Arkitekturen skapar en stor flexibilitet för slutanvändaren eftersom den inte kräver förening av register eller fördefinierade förfrågningar som försöker täcka alla möjliga analytiska scenarion som en användare kan tänkas skapa.

2.5.4 Användargrupper

Det existerar tre användargrupper i QlikView-miljön. Dessa är ”developer” (systemdesigner), ”power user”(superanvändare) och ”end user”(slutanvändare).

² Se ordlistan för begreppsförklaring

Systemdesignern utvecklar QlikView-applikationen utefter de krav som superanvändaren och slutanvändaren kan tänkas ha. Superanvändaren är den användare som skall ha möjlighet att göra enkla förändringar i GUI, dvs. ta bort och lägga till tabeller, grafer etc. Detta utefter de krav som finns hos slutanvändare av vyn. *Figur 2-4* visar de rättigheter som ges till de olika användarna i QlikView.

QlikView User Levels	End users		Power Users	Developers
	QVA	QVA+	QVP	QVE
Server based analysis of data in QlikView application	✓	✓	✓	✓
View, print, and export QlikView reports and data	✓	✓	✓	✓
Local offline analysis of data in QlikView application		✓	✓	✓
Drag and drop report creation			✓	✓
Develop and customize QlikView user interface			✓	✓
Create new ETL script for access to any data source				✓

Figur 2-4: Användargruppstabel från QlikView [17]

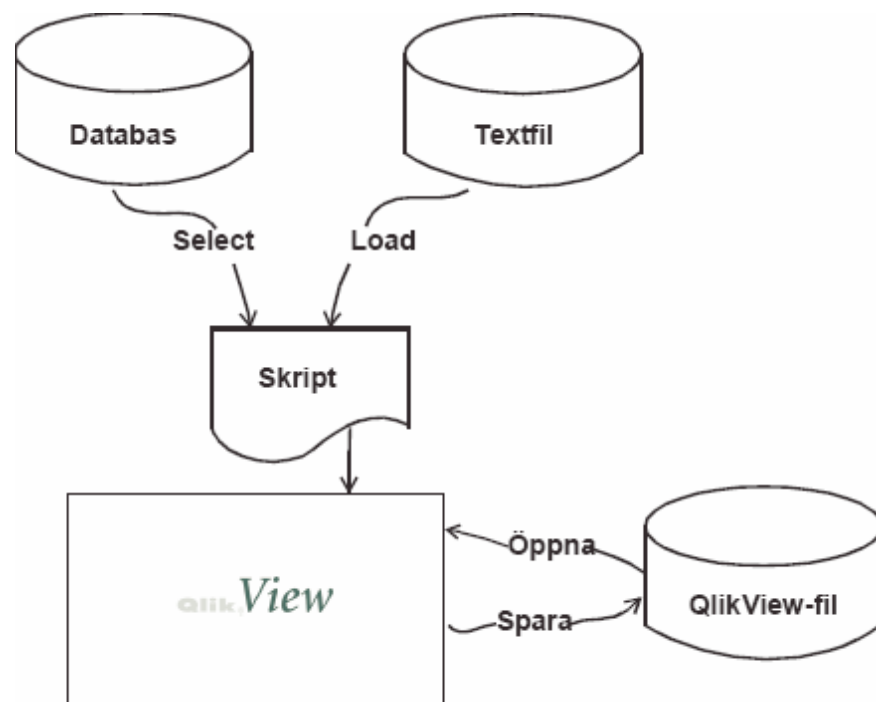
2.5.5 Säkerhet

QlikView kan kontrollera åtkomst till en QlikView-applikation. När en applikation kräver säkerhets- accessnivåer så måste en säkerhetsdatabas adderas till applikationen. Säkerhetsdatabasen definierar användare av applikationen och vilken typ av access som varje användare har. Accessnivåerna är uppdelade i tre grupper: ADMIN, USER och NONE. ADMIN nivå användare har full access till alla områden i en applikation. USER-nivåanvändare har begränsad access till applikationen. NONE är de användare som inte har tillgång till applikationen.

2.5.6 Att skapa en applikation

Det första steget i att skapa en applikation i QlikView är att bestämma vilken data som skall laddas. Det är möjligt att ladda data direkt från ett QlikView-skript, men det vanligaste är dock att data kommer ifrån en existerande fil, "spreadsheet" eller databas. När datakällan är bestämd så måste det skapas ett skript som kopierar data från källan till QlikViews associativa databas. Efter att skriptet är klart skall det laddas i QlikView, vilket sker på ett

kompilationsliknande sätt. Under laddningsprocessen utvärderar QlikView varje utsaga i skriptet, vilket sker i sekventiell ordning. För att sedan kunna använda erhållen data i applikationen så måste ett GUI skapas. Detta görs via QlikView-gränssnittet, där det finns möjlighet att skapa "list boxes" (se sektion 2.5.7) eller andra objekt på ett eller flera blad. Aktuella objekt som skall användas och hur dessa skall användas är helt beroende på den specifika applikationen. Så fort det första objektet är skapat på ett blad så är applikationen tillgänglig för användning. Alla objekt blir automatiskt associerade tillsammans, detta medför att ett klick på något objekt initierar en förfrågan.



Figur 2-5: Skapandet av en QlikView-applikation

Data kan importeras från textfiler eller från databaser via ODBC-gränssnittet. Den importerade datamängden kan tillsammans med det grafiska användargränssnittet sparas som en QlikView-fil.

2.5.7 List Box

Det grundläggande byggnadsblocket i en QlikView-applikation är listboxen. En listbox är ett flyttbart objekt med möjlighet att ändra storlek på och som innehåller data tagen från en enda kolumn i en tabell. Bara unika värden presenteras men det finns möjlighet till presentation av antal förekomster av varje objekt om så önskas.

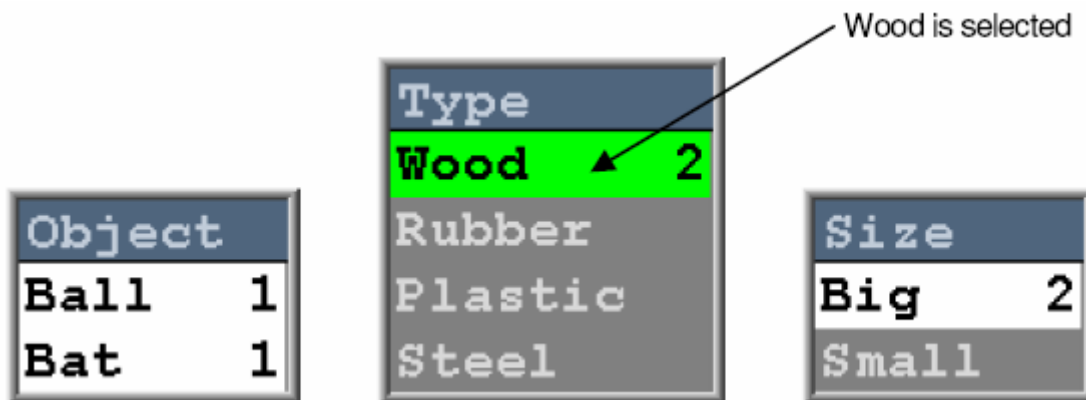
Figur 2-6 – 2-8 visar först hur data lagras i tabell i en traditionell databas och sedan hur samma data kan representeras med hjälp av listboxar:

Bats & Balls		
Object ▲	Type	Size
Ball	Plastic	Small
Ball	Rubber	Big
Ball	Rubber	Small
Ball	Wood	Big
Bat	Plastic	Big
Bat	Rubber	Small
Bat	Steel	Big
Bat	Wood	Big

Figur 2-6: Traditionell databas tabell

Object	Type	Size
Ball 4	Wood 2	Big 5
Bat 4	Rubber 3	Small 3
	Plastic 2	
	Steel 1	

Figur 2-7: Listboxar med samma information som Figur 2-6



Figur 2-8: Val i en listbox

* Vid val av Wood så belyses de associerade dataelementen samt att mängden uppdateras

2.6 Microsoft Business Intelligence (SQL Server 2005)

SQL Server 2005 är Microsofts verktyg för BI. Mjukvaran tillhandahåller en mängd verktyg som stödjer utvecklande och skötsel av BI-applikationer [27]. Den inkluderar Analysis Services (OLAP och data mining), rapporteringsservice och datatransformeringservice (DTS). Plattformen erbjuder också verktyg och funktionalitet för att kunna bygga både klassiska och innovativa analytiska applikationer.

Följande tabell visar komponenterna i ett BI-system och de korresponderande komponenterna i SQL Server 2005;

Komponent	SQL Server 2005
Extrahera, transformera, ladda	Data Transformation Services (DTS)
Relationsdatabas	SQL Server 2005 relationsdatabas
Multidimensionell databas	SQL Server 2005 Analysis Services (AS)
Data mining	SQL Server 2005 Analysis Services (AS)
Rapportstyrning	SQL Server 2005 Reporting Services
Ad hoc förfrågning och analys	MS Office produkter
Databasutvecklingsverktyg	Business Intelligence Development Studio
Databasskötselsverktyg	SQL Server Management Studio

Tabell 2.1: SQL Server 2005:s BI-komponenter

Eftersom denna uppsats är fokuserad på BI-verktyg så kommer bara Business Intelligence Development Studio att beskrivas.

2.6.1 Business Intelligence Development Studio

BI Development Studio [27] är en integrerad utvecklingsmiljö designad för BI-systemsutvecklare. Uppbyggt på Visual Studio 2005, levererar BI-verktyget en integrerad utvecklingsplattform för BI-systemsutvecklare. Debugging, källkontroll och skript- och kod-utveckling är tillgänglig för alla komponenter i BI-applikationen.

Miljön används för att bygga och underhålla BI-applikationer. Följaktligen kan DBA eller utvecklare som definierar en kub eller en datautgrävningsmodell (data mining-model) använda sig av BI Development Studio [26]. Funktioner som kan skapas inkluderar:

- Definition av alla detaljer i Analysis Service-objekt (kuber, dimensioner, utgrävningsmodeller, säkerhetsroller etc.) och testning/debugging och dess definitioner.
- Definition av alla detaljer som tillhör andra komponenter av applikationen såsom Reporting Services-rapporter och DTS paket.

BI Development Studio är som tidigare nämnt baserad på Visual Studios utvecklingsmiljö. Meningen är att BI Development Studio enkelt ska kunna använda Visual Studios utvecklingsmiljö för att tillhandahålla en miljö som stödjer utvecklingen av alla komponenter i en applikation såsom t ex inkludering av Visual Basic och Visual C-kod.

Project mode Miljön i BI Development Studio fokuserar på utvecklingsaktiviteter. Studion introducerar olika stilar för utveckling av analyseringstjänster, vilka går under benämningen ”project mode ” och ”online mode” och som förklaras i sektion 2.6.2 och sektion 2.6.8.

2.6.2

I project mode skapar en utvecklare ett AS-projekt [26] som innehåller en serie AS-objekt (datakällor, kuber, dimensioner, utgrävningsmodeller etc.). Definitionen till projektet och alla objekt det innehar sparas som en fil i filsystemet genom en XML-representation. För att testa och debugga definitioner så bör användaren frekvent placera ut projektet på en AS-server, detta för att skapa objektet på servern.

Alla objekt i ett projekt blir placerade i en AS-databas. Objekten är sparade i en uppsättning filer, där varje datakälla, kub och dimension sparas i separata filer. Alla filer tillhörande ett projekt sparas i filsystemsbiblioteket där projektet skapades.

De möjligheter som ges när ett AS-projekt ska skapas i project mode är:

- Skapa ett nytt tomt projekt
- Skapa ett projekt från en existerande databas
- Öppna ett redan existerande projekt.

Det finns även andra typer av projekt som täcker andra element i en BI-applikation. Exempelvis så innehåller ett rapporteringsserviceprojekt en uppsättning rapportdefinitioner och associerade data källor, som placeras i samma rapporteringsservice-katalog. DTS-projekt som innehåller en uppsättning DTS-pakets definitioner och associerade datakällor är en annan projekttyp som är möjlig.

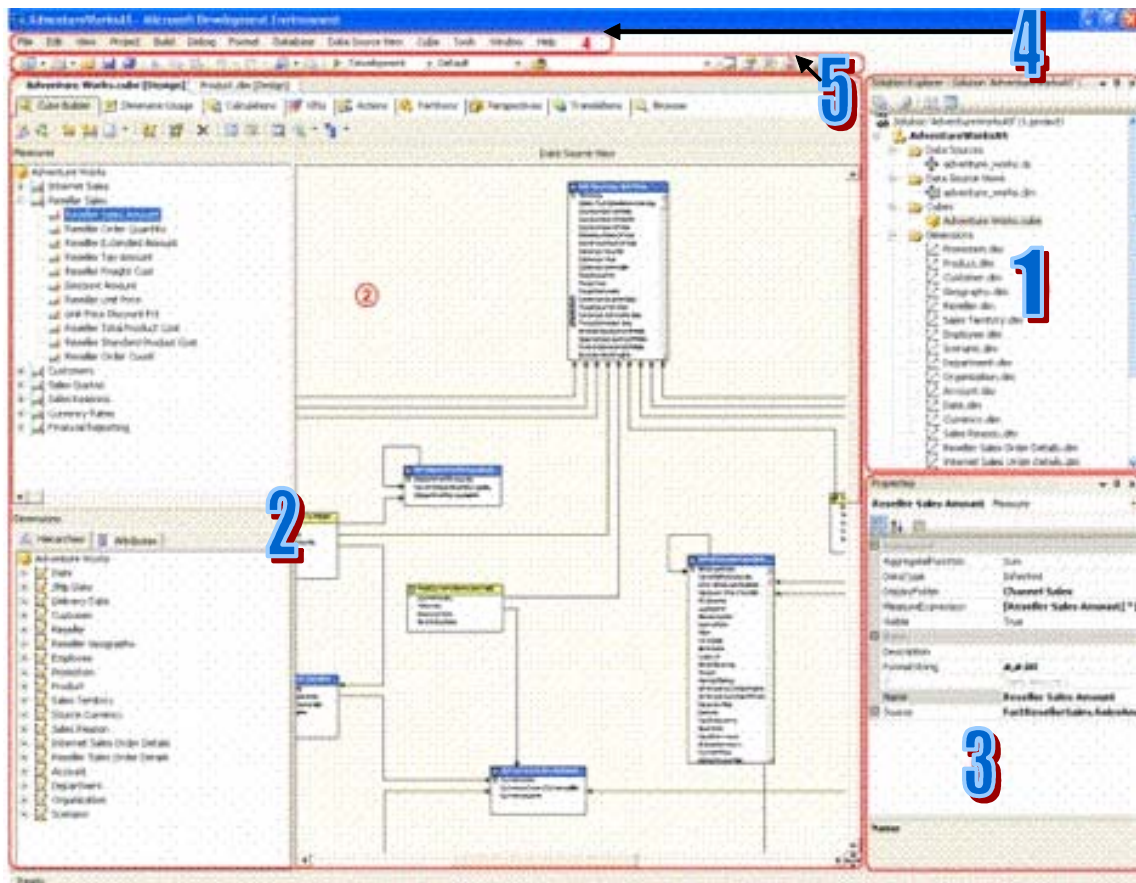
2.6.3 Lösningar

Multipla projekt kan bli grupperade tillsammans i en lösning. En lösning är inget annat än en container med relaterade projekt. Till exempel så kan lösningen ”försäljningsdata” innehålla ett AS-projekt som definierar en kub med försäljningsdata, ett rapporteringsprojekt innehållande några rapporter mot kuben och ett DTS-paket som periodvis processar kuben.

Lösningen agerar för att gruppera dessa projekt tillsammans men det innebär dock inte att de kommer att bli placerade på samma plats.

2.6.4 Element i BI Development Studio-miljön

Figur 2-9 nedan visar BI Development Studio med ett exempel AS-projekt öppnat [26]. De element som är markerade med siffror är huvudelementen för miljön. En beskrivning av dessa element följer även efter figuren.



Figur 2-9: Business Intelligence Development Studio [26]

1. **Solution Explorer.** Lösning utforskaren är ett fönster som visar den lösning som för tillfället är öppen. Filerna innehållande datakällor, kuber, dimensioner etc. visas här. Om en fil öppnas så startas lämpligt designverktyg, vilket tillåter ändring av definition på relevant objekt.

Följande foldrar finns tillgängliga och täcker alla tillåtna typer av objekt:

- Datakällor
- Vyer till datakällor
- Kuber
- Dimensioner
- Utgrävningsmodeller
- Sammanlänknings
- Roller
- Blandat

2. **Analysis Services Designers.** Figuren visar att två designers för tillfället är öppnade. Varje objekttyp har sin egen specifika designer. För mer komplexa objekttyper så finns det separata flikar inuti designern, varje flik täcker olika delar av definitionen för objektet. Det finns dessutom ett verktygsfält i varje flik som stödjer de mest vanliga operationerna för objekten i fliken.

Varje designer har två vyer, designer-vy och kod-vy. Vanligtvis används designer-vyn som visas i figuren [26]. Alternativt så finns det för mer avancerad och robusta lösningar möjlighet att öppna i kod-vy läget, som visar det råa XML innehållet av filen.

3. **Properties Window.** Rekvisitafönstret visar rekvisita för det valda objektet. Rekvisitan kan antingen bli visad i kategorier eller i alfabetisk ordning. När en fil väljs i Solution Explorer så visas rekvisitan för filen och inte objekten i filen.
4. **Main menu.** Huvudmenyn innehåller menyposter som alltid presenteras samt menyposter som den aktiva designern lagt till. Till exempel så kanske menyposten kub enbart presenteras när kub-designern är påloggad eftersom denna designer har gjort det tillägget i huvudmenyn.
5. **Toolbar.** Verktygsfältet innehåller poster för de vanligaste operationerna. Det är även här möjligt att lägga till extra poster som blir beroende av aktiv designer.

2.6.5 Spridning

Som tidigare diskuterats så sparas förändringar och objektdefinitioner till filer inuti projektet. För att testa detta mot kuber och utgrävningsmodeller så är det nödvändigt att placera projektet i en databas på en målserver. Namnet på databasen och servern är satta som rekvisita till projektet. Dessutom så sätts namnet på projektet som databasnamn och mål-servern sätts som default-server i verktygsvalet.

Det finns en hel del delar i AS-designern som kräver att objekt har spridits. Mest noterbart blir det när kuber och dimensioner har en grå bakgrund vilket indikerar att nödvändiga objekt inte har blivit spridna och bearbetade.

Enheten för spridning är hela projektet eftersom handlandet av spridning alltid tar den nuvarande definitionen i projektet och applicerar den på mål-servern. Det är inte möjligt att välja ut och sprida delar av projektet. Emellertid så utför verktygen stegvis spridning av hela projektet dvs. ett objekt i taget. Följaktligen, vid ändring av ett enda objekt i projektet följt av en ny spridning, är det bara det objekt som ändrats i som sänds till servern. Detta är egentligen bara en optimering, nettoresultatet kommer att bli det samma som om hela projektet sänts till servern.

Hela filosofin runt spridning är att utvecklare är isolerade och ”äger” mål-databasen som de sprider till. Anledningen till detta är att det är ohållbart för två olika utvecklare att arbeta oberoende med olika kopior av samma projekt och sedan sprida detta till samma databas. I ett sådant fall är lagutveckling (se sektion 2.6.7) att föredra.

Spridningsprocessen görs i två steg; det första är byggnationen av projektet där innehållet i projektet valideras och utdatafiler skapas, det andra steget är spridningen av output från projektet.

2.6.6 Konfigurering

BI Development Studio tillhandahåller support för olika konfigurationer. De primära målen för konfigurering är följande två scenarion:

- Två olika utvecklare arbetar med samma projekt, men båda har en egen miljö; t ex så använder utvecklarna olika mål-databaser och de kanske till och med använder olika rationella datakällor så att de inte kolliderar under utveckling och testning.
- En utvecklare jobbar med ett projekt men vill arbeta mot olika datakällor vid olika tillfällen. Detta kan vara bra att göra när till exempel datakällor med en väldigt liten volym används i grundprojektet och det sedan måste testas en större datakälla för utvärdering av prestanda.

I verktygen så kan valfri egenskap som är specifik för en viss fysisk miljö ses som konfigurerbar. Exempel på detta kan vara datakällor som är bundna till strängar, förvaringsplatser för datafiler och projektrekvisita (server, databas etc.). Det är därmed möjligt att addera multipla konfigurationsbenämningar där varje konfiguration kan ha olika värden för de konfigurerbara rekvisitorna.

2.6.7 Lagutveckling och återanvändning

Det rekommenderade tillvägagångssättet vid lagutveckling, dvs. flertal utvecklare på samma projekt, är att så långt som möjligt arbeta med enskilda filer. Till exempel så kan en utvecklare ha hand om filerna "produkt.dim" och "lager.dim" och göra förändringar i dess dimensioner medan en annan utvecklare har hand om filen "anställda.dim". Under tiden de utvecklar och testar så är de isolerade, men i slutänden så kombineras deras ändringar.

2.6.8 Online-projekt

Det har hittills fokuserats på "project mode" stilen på utveckling där ändringar sparas till filer i projektet och sedermera sprids till en server. Men det finns ett annat arbetssätt som är möjligt att använda, där användaren kopplar upp sig direkt till en server och förändrar objekt direkt i servern. Detta arbetssätt kallas "online-projekt" och tillvägagångssättet är i stort sätt det samma som i "project mode". Användaren kopplar upp sig mot en existerande databas, eller skapar en ny databas, och sen visas objekten i databasen i lösningsutforskaren (se sektion 2.6.4) som vid vanliga projekt. Designern kan sedan lansera direkt från noderna i lösningsutforskaren, skillnaden är bara att när förändringar sparas så uppdateras servern direkt.

2.7 BusinessObjects XI

BusinessObjects XI [9] levererar insyn i ett företags affärsdata genom ett slutanvändarverktyg som är uppbyggt på en BI-plattform. I programvaran ingår prestandaskötsel, rapportering samt förfrågnings- och analysprodukter. Dessa integrerade slutanvändargränssnitt är stöttade av en BI-plattform och komplementerade av en förenad integration. *Figur 2-10* nedan visar innehållet i BusinessObjects XI.



Figur 2-10: BusinessObjects XI [9]

Med hjälp av en enhetlig vy över företagsprestationer så kan slutanvändaren mäta och följa användaranpassade operativa mål och beräkningar genom en intuitiv, interaktiv och visuellt uppvisad information.

Genom BusinessObjects XI tillåts IT-avdelningen att integrera BI med infrastrukturen och bädda in säkra komponenter i faktiska applikationer.

2.7.1 Användarbehov

Olika användarsegment har olika informationsbehov [9]. Hörnstenen i en organisations BI-strategi måste vara att identifiera behoven för rapportering, mer interaktiva krav som förfrågningar och analyser och den summerade vyn som prestandaanpassade instrumentbrädor och styrkort. *Figur 2-11* summerar dessa krav och vilka program som BusinessObjects XI tillhandahåller för detta.

	Performance Management	Reporting	Query and Analysis
Authoring Requirements	<ul style="list-style-type: none"> Graphical dashboard design Metric trees, analytic templates, and other graphical display options User and role-based dashboard security Access to third-party content within dashboards <p>Role: IT/business analyst, executive, line of business manager</p>	<ul style="list-style-type: none"> Offline design environment Banded report layout Pixel-perfect formatting Open developer standards <p>Role: Developer, designer, IT analyst</p>	<ul style="list-style-type: none"> Web-based query creation Rich semantic layer to mask database complexity Centralized business definitions and security Minimal IT involvement <p>Role: Ad hoc business analyst</p>
Consumption Requirements	<ul style="list-style-type: none"> Personalized dashboard and scorecard interface Ability to easily connect goals to strategy Link to root-cause reporting and analysis Automated, rules-based alerting Embedded business best practice <p>Information relied upon to drive enterprise performance through alignment and visibility.</p>	<ul style="list-style-type: none"> Web-based client Interactive navigation and search Highly-formatted information Ability to receive reports via any application (email, Outlook, within other software packages) Flexible prompting, filtering, and guided analysis <p>Information relied upon to run and manage the enterprise.</p>	<ul style="list-style-type: none"> Web-based client Flexible ad hoc interactivity Ability to ask spontaneous, and iterative questions Access to historical data for trending Offline analysis option <p>Information relied upon to understand how the enterprise is performing.</p>
BusinessObjects XI End-User Products	Dashboard Manager Performance Manager Analytic Solutions	Crystal Reports BusinessObjects Enterprise	Web Intelligence OLAP Intelligence Desktop Intelligence

Figur 2-11: BI-krav och BusinessObjects XI:s lösningar [9]

2.7.2 Prestandahantering

Instrumentbrädor prestandahantering (Performance Management) och styrkort hjälper användaren att hålla reda på och analysera nyckeldata och mål via en grafisk informationsvisning som drivs av en underliggande analytisk motor i BusinessObjects XI [9]. BusinessObjects Dashboard Manager XI, som är ett av de inbyggda programmen som används för utveckling av detta i BusinessObjects XI, förser organisationen med siffror, medvetenhet och direktionmöjligheter som hjälper till att förstå de mest kritiska aktiviteterna. Analytiskt innehåll kan visas på ett personligt och säkert sätt, vilket tillåter användaren att enkelt hålla reda på räkenskap över en tidsperiod. Detta gör det möjligt att se trender, avvikelser och undantag.

BusinessObjects Performance Manager XI, vilket är ett annat inbyggt program för prestandahantering, hjälper organisationer att kommunicera strategiskt och hantera gruppdiskussioner genom styrkort och samarbetsdrag.

Applikationer levereras som analytiska lösningar designade för att täcka specifika affärskrav. Insyn och analys genom applikationer tillåter användare att ta direkt handling när affärsrelaterade nyckelproblem uppstår. I BusinessObjects XI så finns fördefinierade applikationer som är fokuserade på kundprofilering och kampanjdirektion, affärsurvalsoptimering, försäljning och marknadsföringseffektivisering, handelspromotion, försäljarprestation, märkesurvalsoptimering etc.

2.7.3 Rapportering

Organisationsrapportering är processen att ta fram, formatera och delge data som information inuti och utanför organisationen [9]. Den fungerar som en grund till en bredare BI-strategi genom pålitligt och säkert delgivande av de mest efterfrågade delarna av information via webben eller företagsapplikationer. Crystal Reports XI, som är verktyget för detta i BusinessObjects XI, hjälper till att skapa flexibla rapporter samt att integrera dessa i webb- och Windows- applikationer.

Med Crystal Reports kan designers få den information de behöver från personal och företagsdatakällor. Designers har kontroll över hur data är presenterad och formaterad och de ges möjlighet att göra förändringar.

2.7.4 Förfrågan och Analys

Förfrågnings- och analysverktyg tillåter användaren att få svar på sina frågor genom att skapa förfrågningar som svaras på genom hämtning av data från affärsinformationen. BusinessObjects XI förfrågnings och analysverktyg tillåter användare att identifiera trender och finna rotorsaker i affärsdata från relations- och OLAP- datakällor. Detta gör det möjligt för en mängd användare att enkelt skapa förfrågningar och utföra beräkningar utan att förstå komplexiteten i ett databasschema eller SQL. Om samtliga användare använder BusinessObjects Enterprise BI-plattform så ges möjlighet till direkta tråddiskussioner kring informationen som erhållits.

BusinessObjects Web Intelligence XI möjliggör ad hoc-förfrågningar genom ett webbaserat gränssnitt. Semantiklagret tillåter användare att dra och släppa objekt som maskerar underliggande datakomplexitet genom presentation av data i affärstermer. Web Intelligence bygger på en SQL-motor som tillhandahåller ett multidimensionellt perspektiv av data.

För att kunna standardisera en BI-plattform så måste användare ha möjlighet till direkt tillgång och interaktion med både relations- och multidimensionella datakällor. Om en organisation har investerat i en multidimensionell OLAP-server så finns de förmodligen en grupp användare som behöver utföra ad hoc-analyser. BusinessObjects OLAP Intelligence XI är förfrågnings- och analysverktyget som BusinessObjects XI tillhandahåller för detta ändamål.

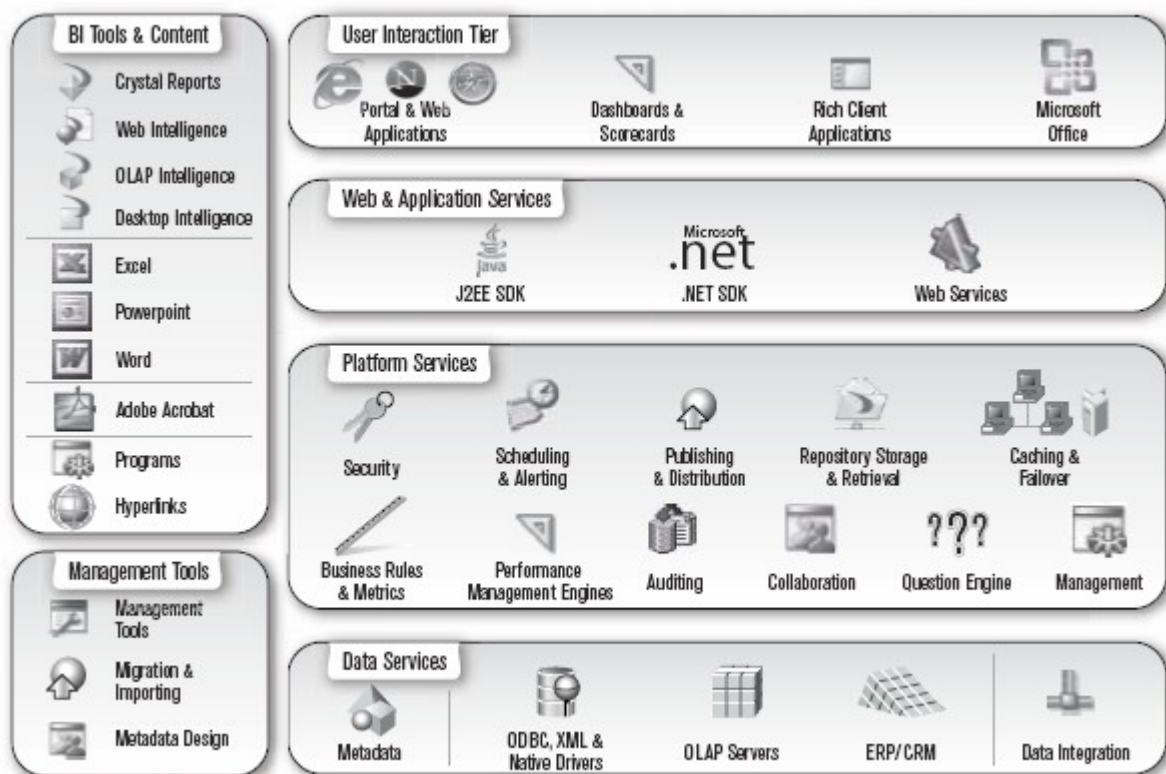
2.7.5 BI-plattform

BusinessObjects Enterprise XI [9] är en anpassningsbar, servicebaserad BI plattform som stödjer prestationsdirektion, rapportering och förfrågnings- och analyseringsgränssnitt i BusinessObjects XI. Det är en komponentbaserad arkitektur som är designad för utbyggbarhet, skalbarhet, pålitlighet, feltolerans och 24 timmars tillgänglighet.

BusinessObjects Enterprise tillåter IT-eneheten på ett företag att säkert hantera och kontrollera alla användare i en BI-spridning. BI-plattformen levererar industristandardiserad support för säkerhetsrättigheter i databaser, integrationsramverk, Java och .NET plattformar. Medan BI växer inuti och utanför organisationer så skalas BusinessObjects Enterprise för att möta den specifika informationen och interaktiviteten som användare är i behov av. Multiserversupport, feltolerans, och lastbalansering försäkrar att information ges till alla användare när de behöver den.

BusinessObjects Enterprise innehåller separata men sammankopplade hjul som är optimerade för specifika uppgifter. Dessa hjul inkluderar slutanvändarlagret, utvecklarservice, plattformsservice och dataservice. Slutanvändaren skapar, ändrar och interagerar med specialanpassade verktyg. Systemadministratören hanterar och övervakar systemet genom centrala direktionskonsolen, centrala konfigurationshanteraren, publicerings- och importeringsmallen, affärsvy-designern och universella designern.

Figur 2-12 visar BusinessObjects Enterprise användningsområden.



Figur 2-12: BusinessObjects Enterprise användningsområden [9]

BusinessObjects Enterprise levererar även korrekt BI-information i MS Word-dokument, Excel-blad och PowerPoint-presentationer. Med en levande Office-miljö så behövs det ej slösas på värdefull tid för att klippa och klistra innehåll från en miljö till en annan.

För IT-ledning och administration så erbjuder plattformen ett centralt lager för allt innehåll och alla användarprofiler. Säkerhetsrättigheter, dokumentssäkerhet, användare etc. kan även styras och administreras via plattformen.

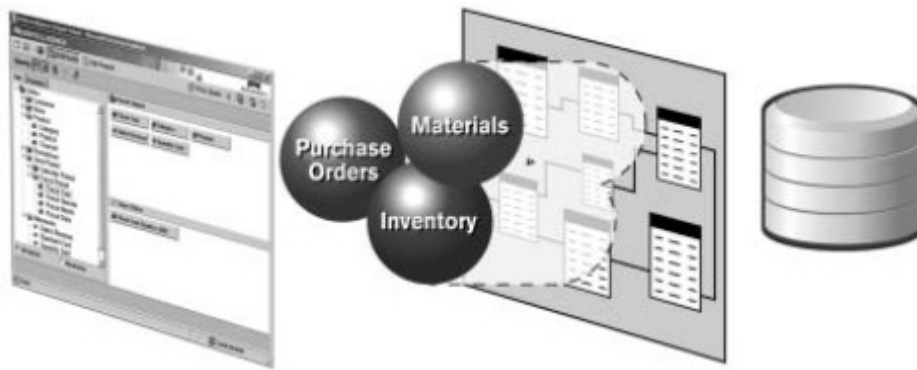
Semantiklagret

Kärnan i BusinessObjects Enterprise är det semantiska lagret. Detta lager är en affärsrepresentation av företagsdata som hjälper slutanvändaren att få automatisk tillgång till vanliga affärstermer. BusinessObjects Enterprise XI stödjer Business Objects-miljön och affärsvyer samt inkluderar ett grafiskt gränssnitt för design.

Semantiklagret är fundamentet för berättigandet av slutanvändarnas förfrågningar och analyser. Lagret förenklar komplexiteten i affärsdata genom att använda kända affärstermer istället för dataspråk för access, manipulering och organisering av information. Affärstermerna sparas som objekt i Business Objects-miljön. Web Intelligence använder

sedan denna miljö för att förenkla användarprocessen som krävs för enkla och komplexa slutanvändarförfrågningar och analyser. Alla objekt sparas och säkras i Central Management Console, vilken kräver ett användarlogin för tillgång till miljödesignergränssnittet.

Varje miljö är en partiell representation av en databas och levereras till användaren i affärstermer, vilket visas i *figur 2-13*.



Figur 2-13: Business Objects miljöförenkling [9]

Med BusinessObjects XI kan en Crystal Reports-designer författa miljöbaserade rapporter. Utöver presentation av förfrågningspanelen som används för grafisk förfrågningsgenerering så kan en designer få tillgång till miljön som en datakälla, skapa förfrågningar grafiskt genom förfrågningspanelsgränssnittet och generera SQL som kan vara högt formaterad och levererad igenom organisationen.

Affärsvyer förenklar rapportskapandet genom abstraktion av datakomplexitet för Crystal Reports utvecklare. Affärsvyer hjälper också till att separera datakopplingar, datatillgång, affärselement och generell affärsdefinition vilket ger en rapportdesigner rättigheter till accesskontroll.

2.7.6 Dataintegration

För att försäkra att BI-användare alltid arbetar med exakt och trovärdig information så inkluderas integrationsteknologi för data. BusinessObjects Data Integrator XI är designat för att leverera produktivitet och skalbarhet vilket förenklar och accelererar ETL-teknologi³. Den tillhandahåller även realtids- och batchdataförflyttning samt naturligt gränssnitt till grunddata

från paketerad ERP, CRM och SCM⁴. Data Integrator har även en enda miljö för all dataintegration som kan tänkas behövas.

Eftersom Data Integrator sammankopplar dataintegration med BusinessObjects Enterprise XI så kan användare lita på information de får tillgång till och analyserar i rapporter. Data Integrator skapar automatiskt en Business Objects miljö i sitt ETL-designgränssnitt, vilket gör att signifikant utvecklingstid sparas och det tas vara på det arbete som redan är gjort i datalagret och i ETL-processen.

Figur 2-14 sammanfattar det som beskrivits under sektion 2.7, vilket är vad BusinessObjects XI ger i BI-prestanda:



Figur 2-14: BusinessObjects XI:s BI-prestanda

2.8 Sammanfattning

Begreppet Business Intelligence är relativt nytt även om det har använts indirekt tidigare. Detta gör att det kan vara svårt att definiera och förstå eftersom olika författare ger olika definitioner. Men förhoppningsvis så har detta kapitel bidragit till att ge en bättre förståelse inom begreppet Business Intelligence.

³ Se bilaga A (Ordlistan) för begreppsförklaring.

⁴ Se bilaga A (Ordlistan) för begreppsförklaring.

Kapitlet har också gett en inblick i den datahantering som Cederroth har som är relevant för projektet. AS/400 som är hårdvaran där Movex-databasen är lagrad är vida spridd i världen dvs. många företag använder sig av denna servermodell från IBM. Själva Movex är ett av Sveriges största affärssystem där all affärsdata inom Cederroth-koncernen hanteras. Movex-databasen kommer att användas i kapitel 3 för hämtning av relevant affärsdata till BI-applikationen. Begreppen ODBC och AD har också getts en djupare förklaring i detta kapitel.

De sista delarna i kapitel 2 ger en djupare genomgång i de olika BI-system som kommer att utvärderas i kapitel 4. QlikView som är det första systemet som introduceras i detta kapitel är även det system som Cederroth använder för BI. Det är även i QlikView som projektapplikationen skapades i. Microsoft Business Intelligence är den mjukvara som utvecklas av Microsoft för BI. Denna mjukvara har utvecklats under 2005 och är även det första direkta BI-systemet som Microsoft skapat. BusinessObjects XI, som är den sista av de tre som introduceras, har dock funnits en längre tid och tillhandahåller en mängd olika BI-instrument i BI-systemet.

3 Utveckling den finska QlikView-applikationen

Detta kapitel ger först en introduktion om hur en standardmässig utveckling av en QlikView-vy går till i sektion 3.1. Sedan ges en ingående stegvis presentation hur utvecklingen av det finska POS-data projektet gick till i sektion 3.2 – 3.10. Till sist ges en sammanfattning över kapitlet i sektion 3.11.

3.1 Introduktion

Utvecklingen av en databasvy går på vägen ifrån idé till publicering för användning igenom en mängd olika steg. Ett misslyckat steg kan få fatala konsekvenser för den slutgiltiga vyn. Till exempel så kan det medföra att data associeras felaktigt eller att data fattas, eller så kanske helt felaktig data visualiseras. För att undvika nämnda samt onämnda konsekvenser så bör varje steg följas upp av minutiös testning. Cederroth som är arbetsgivaren, vill absolut inte att framtida prognoser skall byggas på felaktig affärsdata.

Att skapa en databasvy kräver i grunden data men för att kunna arbeta med denna data så krävs även ett verktyg. Det verktyg som ställts till förfogande heter QlikView och är den BI-mjukvara som Cederroth använder sig av för hämtning och analys av data. För att skapa en applikation i QlikView så krävs det att följande steg utförs:

1. Lokalisering av data
2. Skapa laddningsbart skript
3. Exekvera skriptet
4. Skapa grafiskt användargränssnitt
5. Börja använda applikationen

Lokalisering av data som är det första steget i skapandet av en QlikView applikation, är det steg där det bestäms vilken data som skall laddas. Data återfinns oftast på lokala servrar men det räcker dock inte med databaskällan. De fält som är relevanta i den lokaliserade datan bör även tas ut. Analys av data är även lämpligt att göra eftersom det är just dessa data som kommer att användas i applikationen. När det sedan är klargjort vilken data och vilka fält som skall laddas så måste ett skript skapas för kopiering av data från källan till QlikViews

associativa databas. Skriptet exekveras sedan, vilket även inkluderar en utvärdering om huruvida uttrycken i skriptet är korrekta eller inte. Efter detta är det dags att placera ut grafiska objekt som gör det möjligt för slutanvändaren att arbeta med vyn. När det första objektet är utplacerat så är vyn möjlig att publicera för användning.

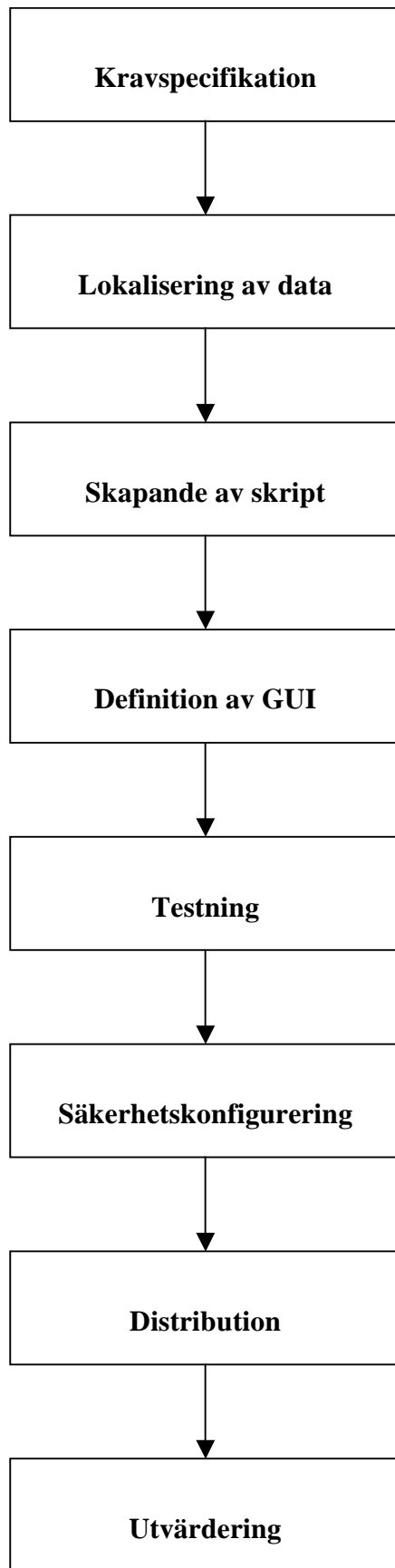
De steg som QlikView kräver är i sig själva inte tillräckliga för att kunna utveckla en givande applikation åt företaget. En applikation i sig går givetvis att utveckla men för Cederroths räkning så krävs några ytterligare steg. Stegen som Cederroth vill addera till processen är kravspecifikation, testning, säkerhetskfiguration och utvärdering. Kravspecifikation utformas genom en diskussion med superanvändaren, där lämplig data, användargränssnitt, användningsområde och användargrupp tas upp. Det finns även krav från IT-enheten på hur en vy får se ut och vilken data som får visas. Dessa krav skall tillsammans med superanvändarens krav sättas ihop till en slutgiltig kravspecifikation.

Testning ska göras efter varje steg men en slutgiltig testning görs också innan distribution av vyn. Under den slutgiltiga testningen ska uppladdad data verifieras, associationer ses över och gränssnittet testas och analyseras. Sedan bör även hela applikationen ses över så att användarspecifikationen uppfylls.

Säkerhetskfiguration är ett måste för att avgränsa användarbehörigheten. QlikView tillhandahåller funktionalitet för behörighetskontroll, vilket medför användarnamnshantering och lösenordshantering. Dock så har IT-chefen beslutat att detta medför komplikationer för slutanvändaren som måste memorera ännu ett användarnamn och ännu ett lösenord. Därför bör användarbehörigheten styras via AD (se sektion 2.4) för nätverket. Detta är en någorlunda behändig lösning med tanke på att man förbigår användarnamn och lösenord genom att låsa tillgången till QlikView-servern via inloggningssystemet. Inloggningssystemet har förvisso krav på användarnamn och lösenord men detta är dock ett måste för den generella säkerheten på företaget.

Till sist så skall även projektet i sig utvärderas. Själva utvärderingen kommer att göras med superanvändaren som förhoppningsvis fått tillräckligt med feedback av slutanvändarna för att kunna ge konkret kritik. En annan utvärdering ska göras med IT-chefen för att analysera applikationen så att strukturen följer det standardutseende som används i företaget.

Följande flödesdiagram (*figur 3-1*) illustrerar hur applikationsutvecklingen kommer att gå tillväga;



Figur 3-1: Projektflöde

3.2 Det finska POS-data projektet

QlikView introducerades för den finska divisionen i slutet på augusti 2005. Denna introduktion gav stort gehör vilket ledde till att den finska försäljningsavdelningen tog kontakt med IT-avdelningen och gav förslag på möjlig affärsdata för analys. Det första projektet som startades innefattade försäljningsdata från finska grossisters försäljning ut till butik (POS-data, se sektion 3.4). Försäljningsdata sänds till den finska försäljningsavdelningen i slutet på varje månad och för att kunna analysera denna på ett behändigt sätt så krävs ett analysverktyg. QlikView passar perfekt i detta avseende. Med hjälp av QlikView går det nämligen att koppla ihop försäljningsdata med Cederroths egna produktstruktursdata vilket ger en bättre helhetsbild. Fortsättningsvis så kommer detta kapitel behandla tillvägagångssättet i detta projekt.

3.3 Kravspecifikation

En kravspecifikation kan delas upp i två delar, en del för funktionella krav och en del för icke-funktionella krav. Funktionella krav beskriver vad applikationen ska göra och hur den skall presenteras. Icke funktionella krav beskriver på vilket sätt detta ska göras.

Superanvändaren i detta projekt var en finsk ”Knowledge Manager” vid namn Sari. För att kunna utforma en kravspecifikation på bästa möjliga sätt så bör superanvändaren och systemdesignern träffas för diskussion. Detta var dock inte möjligt på grund av det geografiska avståndet. En annan komplikation var att superanvändaren hade begränsad kunskap om QlikView och dess styrka som analysverktyg, dvs. vad som går att presentera av ursprungsdata och på vilket sätt det kan presenteras. Att utforma en kravspecifikation som beskrev vad superanvändaren förväntade sig av applikationen innan själva grundutvecklingen började var därav en omöjlighet.

Lösningen på detta problem blev att utforma kravspecifikationen efterhand på bästa ”extreme programming” maner [3]. En sådan iterativ lösning passade perfekt med tanke på att det enda kravet som fanns från början var att erhållen POS-data skulle användas.

För att superanvändaren skulle kunna bidra till kravspecifikationen på enklast möjliga sätt så beslöts det att först skulle skriptet skapas och sedan skulle ett användargränssnitt med en mängd olika QlikView ark skapas.

QlikView-arken innehöll olika grafiska användargränssnitt. Utifrån dessa ark och den data som presenterades fick superanvändaren sedan välja ut vilken data som var lämplig att presentera och på vilket sätt den lämpligast borde presenteras.

IT-enhetens krav var lättare att sammanställa med tanke på att det var därifrån projektet verkställdes. Kraven innefattade bland annat vilken data som skulle hämtas från Movex-databasen i AS/400-servern och hur användargränssnittet skulle se ut.

Kravspecifikation

Funktionella krav

- Laddade kolumnnamn skall översättas till engelska.
- Gränssnittet skall vara enhetligt med tidigare gränssnitt.
- Data skall kunna laddas oberoende av uppsättningen i Excel-filen.
- Enbart projekttillhörande data skall laddas från Movex-databasen.
- Applikationen skall bara vara tillgänglig för berörda användare.

Ickefunktionella krav

- Skriptet skall vara strukturerat på ett liknande sätt som tidigare skript.
- Applikationen skall färdigställas på lokal klient innan publicering.
- Kontinuerlig testning.

3.4 Lokalisering av data

Grunddatakällan var Excel-filer innehållande ”Point of sales”-data (POS-data) som sänds från grossister till den finska försäljningsenheten. Utöver denna data så skulle även Cederroths egna produktstruktursdata användas.

POS-datafilerna var tvungna att lagras på QlikView-servern för att det skulle kunna göras en kontinuerlig uppdatering av data i QlikView applikationen. Produktstruktursdata fanns sedan tidigare lagrad i en Movex-databas i AS/400-servern vilket medförde att uppdatering av denna data genomfördes via Movex-systemet.

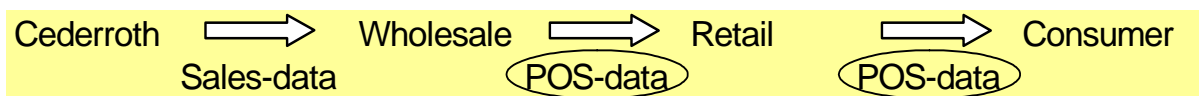
Varför skulle det då skapas vyer för POS-data? Information om vad som händer längre ner i leverantörsflödet är av stor vikt för företaget. Fler kunder tenderar att bli mer öppna med

försäljningsdata vilken delges åt Cederroth. Företaget kan till exempel använda POS-data för försäljningsanalys, bättre förutsägelse, bättre produktionsplanering, bättre produktintroduktion etc . Om det sedan skulle vara möjligt att få data på daglig basis så skulle det till och med vara möjligt att fylla på hos kunden vid behov utan att kunden själv tar kontakt med Cederroth för beställning av varor.

Cederroth har som mål att bli bättre på att använda tillgänglig (och snart tillgänglig) POS-data från sina kunder. Det måste utvecklas processer och verktyg för att transformera dessa data till information som stöd för de olika affärsbesluten som tas i organisationen. Här passar det bra att använda metaforen att desto snabbare företaget kan se hur vägen framför svänger, desto snabbare går det att styra i rätt riktning. I det här fallet så ses vägen framför snabbast genom vindrutan, det vill säga försäljningsdata från kunder (POS-data), inte enbart genom backspegeln, det vill säga Cederroths egna försäljningsdata som lagrats sedan tidigare.

BI-verktyget QlikView är anpassat för transformation av data till information. Detta gjorde att den applikationsdata som skulle användas i projektet kunde ses som ett delmål i den framtida POS-datahanteringen.

Cederroth är intresserad av två olika sorters POS-data, Partihandel till återförsäljare (Wholesale to Retail) och återförsäljare till konsument (Retail to Consumer) vilket visas i figur 3-2. Den POS-data som involverades i projektet var återförsäljare till konsument.



Figur 3-2: POS-data⁵

För att göra det möjligt att hitta denna data i Movex-databasen var jag tvungen att öppna filerna och analysera tabellerna och kolumnerna. När detta skulle göras så uppstod problem på grund av bristande kunskap i det finska språket. Tabeller och kolumner var nämligen namngivna på finska. Efter ett mailutbyte med den finska försäljningsavdelningen så erhöles en översättning på berörda tabeller och kolumner.

⁵ Figuren är tagen ur Cederroths egna POS-data definition.

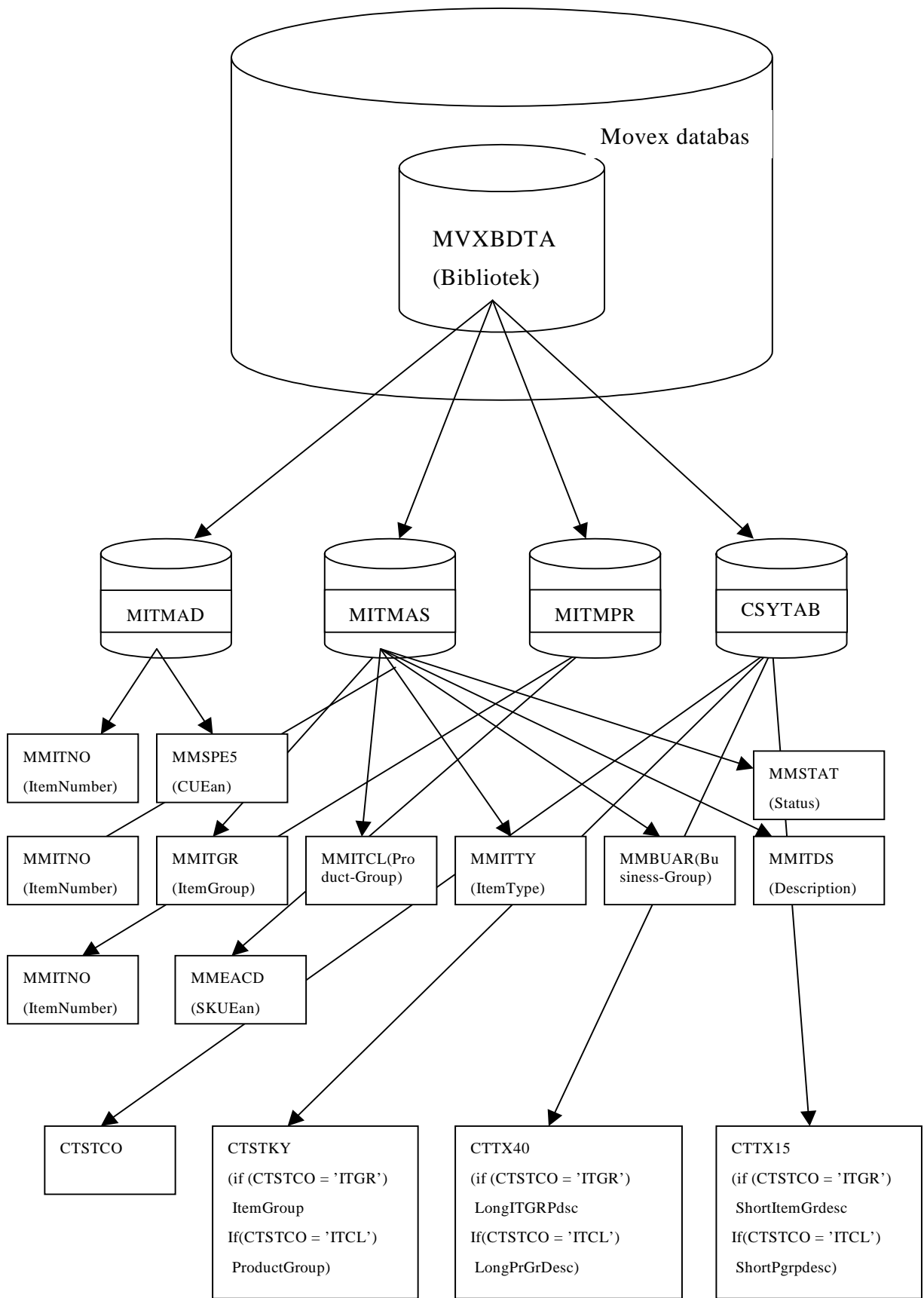
Efter en djupare jämförelse mellan de erhållna Excel-filerna så insågs att de skilde sig i upplägget. De innehöll i stort sett samma sorts data men kolumnuppsättningen skilde sig åt. Detta fick dock accepteras eftersom skriptinläsningen skall kunna lösa detta.

Följande tabell visar den Excel-data som jag arbetade med och dess innebörd;

TR 4	=	Kategorikod
Ryhmä	=	Kategorinamn
Toute	=	Produknamn
Etiketin lisäteksti	=	Extra ettiket
SM	=	Butikskedja
PR	=	Butikskedja
ALEPA	=	Butikskedja
SALE	=	Butikskedja
HUOLTAMOT	=	Butikskedja
TERRA	=	Butikskedja
Myynti kpl	=	Försäljningsstycke
Myynti kpl ind kk	=	Försäljningstycke index/månad
Myynti kpl ka /ky	=	Försäljningsstycke butiksnitt
KY:n peitto lkm	=	Antal butiker per produkt
Myynti €	=	Försäljning €
Myynti-ind.kk	=	Försäljning €index/månad

Tabell 3-1: Inkommande POS-data

Figur 3-3 illustrerar tillvägagångssättet för hämtning av data ur Movex. Först och främst behövs det väljas ett huvudbibliotek. Det som var bäst lämpat för koppling till POS-data var MVXBDA vilket är huvudbiblioteket för alla filer i Movex. Ur detta bibliotek extraherades sedan följande filer: MITMAD(övrig artikeldata), MITMAS(artikelregister), MITMPR(artikelpris och försäljningsdata) och CYSTAB(systemtabell). Dessa filer ansågs innehålla de nödvändiga fälten för att skapa relevanta kopplingar till POS-data.



Figur 3-3: Fält från Movex

3.5 Implementation av skript

När datakällan nu var bestämd var det dags att skapa skriptet för uppladdning av data från Excel-dokumenterna (som finns i QlikView-servern) och Movex-databasen.

Ett QlikView-skript definierar databaskällan och de tabeller och fält som skall laddas i applikationen. Med hjälp av dessa går det sedan att kalkylera nya variabler och register genom att använda de inbyggda funktioner som finns i QlikView-skriptet. För att underlätta skapandet av laddningsskriptet så finns det en hjälpredda inbyggd i QlikView. Hjälpredan är dock till för enklare applikationer. Vid implementering av mer komplexa applikationer så måste skriptet skapas manuellt för att kunna kombinera datakällor på ett icke standardmässigt sätt. QlikViews skriptspråk inkluderar även syntax för flödeskontroll t.ex. loopar, villkorlig databehandling och subrutiner vilket kan vara mycket användbart under utveckling.

För att lära sig syntax och semantik för QlikView-skriptet var jag tvungen att först gå igenom en introduktionskurs. Att lära sig grunderna i skriptet tycktes vara tämligen simpelt. Emellertid, när sedan dessa grunder skulle användas för att skapa en applikation av den givna POS-datan så insågs ganska snart att det inte räckte till. För att kunna hantera given affärsdata var det ett måste att lära sig de funktioner som är inbyggda i QlikView-skriptet. Nedan kommer olika problem och dess lösning beskrivas, för att se helheten så finns det fullständiga skriptet i bilaga B (källkod).

POS-datafilerna hade, som nämndes i sektion 3.4, inte samma format. Det vill säga, alla innehöll inte nödvändiga kolumner för relevant presentation i vyn. År och månad, som ansågs vara en viktig del för att kunna dela upp försäljningen, var olika placerade i de olika filerna. Filerna skickas som tidigare nämnt i slutet på varje månad, vilket gör att de innehåller just den månadens försäljningsdata. Försäljningsavdelningen i Finland använde dock inte QlikView som analysverktyg tidigare, vilket gjorde att de samlade ihop gammal data i en och samma Excel-fil och angav en kolumn för både månad och år. Särskiljningen av år och månad i det fallet gick till på följande sätt;

```
Left (KK, 4) as [Year] // (1)
```

```
Right (KK, 2) as [Month] // (2)
```

```
Left (KK, 4)&num(Right(KK, 2), 00) as [YearMonth] // (3)
```

(1) Här tas de första 4 siffrorna från vänster i kolumnen KK ut med hjälp av den fördefinierade funktionen Left och associeras sedan till år.

(2) De 2 första siffrorna från höger tas ut ur kolumnen KK och associeras sedan till månad med hjälp av funktionen Right.

(3) Det ansågs även lämpligt att skapa en YearMonth (t.ex. 200511), detta för att förenkla för val i det grafiska gränssnittet. Årsdelen av YearMonth togs ut på samma sätt som i (1). Sedan sammanfogades den med månadsdelen med nyckeltecknet &. Num() var tvungen att användas för att de tecken som togs ut med Right-funktionen skulle ses som siffror och inte tecken, vilket är ett semantiskt krav för sammanfogning (&).

I de Excel-filer som sänds månadsvis från återförsäljarna så fanns det ingen kolumn för år och månad. År och månad inkluderades dock i filnamnet, vilket gjorde att på något sätt var dessa tvungna att erhållas därifrån. Filebasename() är den inbyggda funktion som QlikView-skriptet erbjuder för just detta framtagande. Följande skriptrad är ett exempel på hur detta användes när YearMonth framtoogs.

```
Right(filebasename(), 4)&num(mid(filebasename(), 26, 2), 00) as  
YearMonth // (4)
```

(4) Right-funktionen plockar de första 4 tecknen från höger i filnamnet eftersom filebasename() är den första inparametern. Sedan sammanfogas dessa med &-tecknet samt som i tidigare fall num-funktionen. Som inparametrar till num() används mid-funktionen och 00. Med hjälp av mid() så går det att plocka ut tecken mitt i en rad av tecken. I ovanstående fall så togs tecken 26 och 27 ut ur filnamnet vilket var månaden. Inparameter 00 till num-funktionen gör att enkla siffror som till exempel 5 blir 05.

Alla kolumner som laddas ur ett Excel-dokument har ett kolumnnamn (kolumnetikett) inlagt per default. I vissa fall så måste dessa ges ett mer anpassat namn. Projektets kolumnetiketter var på finska och var därmed tvungna att översättas till engelska, delvis för bättre förståelse men framförallt för att det var ett krav från företaget. Detta gjordes med en as-sats som i följande exempel:

```
Tuote as [Product name] // (5)
```

(5) Toute som betyder produktnamn översättes till Product name vilket är företagsstandard. Nyckelordet as står för alias och associerar de båda namnen med varandra.

I de fall det är siffror inblandat så behövs ibland QlikView-skriptets inbyggda text-funktion. Utan text-funktionen skulle 0004187003 bli 4187003 eftersom det tolkas som ett binärt tal.

Ett exempel där denna funktionalitet var tvungen att användas i projektet var när EAN skulle kopplas.

Text(EAN) as CUEan // (6)

(6) Här används Text() för att ean-koden skall läsas som en sträng av tecken. Anledningen till att man byter namn till CUEan (Consumer Unit EAN) är att Cederroth använder sig av två olika EAN-koder. De två de behandlar är konsumentenhets EAN och lagerenhets EAN (skuEAN, stock keeping unit EAN). I det finska projektet så var den data som kom från återförsäljarna i konsumentenhets-EAN.

Det största problemet som uppstod när skriptet skulle skapas var att återförsäljarna inte hade en egen kolumn. På grund av detta så kunde inte försäljningsdata associeras till rätt återförsäljare. Återförsäljarna SM och PR fanns förvisso med i Excel-filerna men dessa hade lagts på följande sätt:

SM						PR	
Myynti kpl	Myynti kpl ind	Myynti kpl	KY:n peitto	Myynti €	Myynti-ind.kk	Myynti kpl	Myynti kpl ind kk
	kk	ka /ky	lkm				
1 475	82,51	5	289	3 386	86,99	844	87,11
3	3,51	2	2	6	4,80		0,01
473	0,01	4	115	1 463	0,00	290	0,01
695	0,01	6	116	2 148	0,00	455	0,01
	0,01				0,00		0,01
1	8,31	1	1	9	7,85	11	25,01
	0,01				0,00	1	0,01
32	0,01	2	18	304	0,00	29	0,01

Tabell 3-2: POS-data från finska återförsäljare

När man skapar en uppladdning från ett Excel-dokument i QlikView-skriptet så går det bara att läsa en rad i taget. Innehållet i de olika kolumnfälten på den raden används sedan som kolumnhuvud. Kolumnhuvudet ses som ett referensnamn för just de kolumnerna. Således går det alltså inte att få med butikskedjan som tillhör försäljningsdatan om det inte finns en kolumn med butikskedjor i den laddade raden.

Detta medförde att varje fil var tvungen att laddas en gång per återförsäljare. Det sätt detta löstes på var genom att i den laddade filen togs de kolumner som inte tillhörde den berörda återförsäljningskedjan bort. Sedan lades det till en ny kolumn som fylldes med butikskedjans namn. Efter att det hade gjorts så byttes namnet på kolumnhuvudet till kauppa (återförsäljningskedja) vilket sedan i laddningssatsen byttes till Chain of stores.

Följande kodsekvens visar ett exempel på hur detta löstes för butikskedjan SM;

FROM

```
[Q:\qlikview\Database\Cederroth\Sale\POS\SOK\SOK*.xls](biff,
header is 2 lines, embedded labels, table is [sheet1$],
filters(
// (7)
  Remove(Col, Pos(Top, 35)),
  Remove(Col, Pos(Top, 34)),
  Remove(Col, Pos(Top, 33)),
  Remove(Col, Pos(Top, 32)),
  Remove(Col, Pos(Top, 31)),
  Remove(Col, Pos(Top, 30)),
  Remove(Col, Pos(Top, 29)),
  Remove(Col, Pos(Top, 28)),
  Remove(Col, Pos(Top, 27)),
  Remove(Col, Pos(Top, 26)),
  Remove(Col, Pos(Top, 25)),
  Remove(Col, Pos(Top, 24)),
  Remove(Col, Pos(Top, 23)),
  Remove(Col, Pos(Top, 22)),
  Remove(Col, Pos(Top, 21)),
  Remove(Col, Pos(Top, 20)),
  Remove(Col, Pos(Top, 19)),
  Remove(Col, Pos(Top, 18)),
```

```

Remove(Col, Pos(Top, 17)),
Remove(Col, Pos(Top, 16)),
Remove(Col, Pos(Top, 15)),
Remove(Col, Pos(Top, 14)),
Remove(Col, Pos(Top, 13)),
Remove(Col, Pos(Top, 12)), // (8)
ColXtr(11, RowCnd(CellValue, 1, StrCnd(null)), 0), // (9)
Top(12, 'SM'), // (10)
Replace(12, top, StrCnd(null)), // (11)
Top(12, 'Kauppa') // (12)
));

```

(7) Varje laddningssats i skriptet kapslas in mellan nyckelorden Load och FROM. I from-uttrycket skrivs varifrån data ska laddas och vad det är för data. I fallet som beskrivs laddas alla xls-filer som börjar med SOK från given sökväg. From-satsen kräver också vetskap om vad det är för filtyp, hur många rader ner kolumnhuvudet är, huruvida kolumnhuvudet skall användas som etikett eller inte och vilken tabell som gäller. Detta görs på ett funktionsanropsliknande sätt där in-parametrar tas. Den första in-parametern är filtypen (i detta fall biff) vilket gör att QlikView tolkar inläsningsfilen som en Excel-fil. Nästa parameter beskriver var kolumnhuvudet är; header is 2 lines, det vill säga 2 rader ner i dokumentet. Kolumnhuvudet skall användas som etikett vilket nästa parameter klargör (embedded labels). I en Excel-fil kan det finnas mer än ett Excel-blad, den tredje parametern klargör vilket blad som data skall laddas ifrån. Sista parametern är en filtreringsfunktion som rensar i filen, se (8) – (12).

(8) Remove() tar 2 in-parametrar; den första är vad som ska tas bort och den andra är var det som ska tas bort finns. Här skulle alla kolumner som inte berörde återförsäljarkedjan SM tas bort vilket var kolumnerna 12-35. Den första parametern Col visar att det är kolumner som skall tas bort. Den andra parametern använder sig av funktionen Pos() som är inbyggd i QlikView-skriptet. Pos-funktionen tar två parametrar, den första bestämmer hur mycket som skall tas bort och den andra beskriver vilken position det finns på. Nyckelordet Top säger här att det är hela kolumnen som ska tas bort och siffran som är nästa in-parameter beskriver vilken position kolumnen befinner sig på.

(9) När alla kolumner som inte tillhörde SM tagits bort så skapades en ny kolumn för butikskedjan. ColXtr-funktionen skapar här en kopia av kolumn 11 och tömmer sedan den på

innehållet genom att använda RowCnd-funktionen. De följande in-parametrarna (CellValue, 1, StrCnd(null)) gör att funktionen ersätter alla kolumnrader där det finns ett värde med ett null-värde.

(10) Top(12, 'SM') ersätter null-värdet på etiketten i den nyskapade kolumnen med SM.

(11) Efter att en etikett satts på den nyskapade kolumnen så används Replace-funktionen för att fylla den. In-parametern 12 anger vilken kolumn som skall fyllas, nyckelordet top som används som andra in-parameter anger att kolumnen skall fyllas med de tecken som kolumnetiketten innehar. StrCnd(null) ,vilket är den sista in-parametern, initierar att alla fält med null-värden skall ersättas.

(12) När kolumnen nu blivit fylld med SM på varje rad så ersätts den etikett som för tillfället är SM med Kauppa genom användning av Top().

Med detta uppladdningssätt måste all återförsäljningsdata laddas en gång per butikskedja och fil. Det kommer tolv Excel-filer per år och det är 6 olika återförsäljningskedjor. Detta medför att för att ladda upp ett års försäljningsdata krävs $12 \times 6 = 72$ uppladdningsprocesser. 72 uppladdningsprocesser kan ses som väldigt många. Men i jämförelse med att en anställd skall sitta i en timme och manipulera samma data för att göra den lättare att läsa in så går det mycket snabbare. Slutsatsen som kan dras är att kostnaden på att en anställd skall ändra i Excel-blad för att få dem enhetliga för inläsning är mycket högre än att applikationen görs oberoende av utseendet på Excel-bladen genom skriptet.

När den skriptdel som laddar upp data från Excel-filerna var färdigställd var det dags att lösa uppladdningen ur Movex-databasen. Data laddas här med hjälp av SQL-kommandon. För att överhuvudtaget få tillgång till affärsdata ur Movex så måste först en ODBC-uppkoppling skapas. ODBC-kopplingen gör det möjligt att ställa förfrågan till serversidan vilken i sin tur ger svar till klientsidan. Förfrågningarna i POS-datafallet var tvungna att filtrera ut data som var relaterad till den data som fanns i Excel-filerna. Skulle inte detta ha gjorts så hade irrelevant data laddats ur databasen, dvs. försäljningsdata som inte tillhörde den finska återförsäljningen. Denna data skulle visserligen inte få någon associativ koppling till försäljningsdatan men den skulle dock finnas med i det grafiska användargränssnittet, vilket kunde medföra förvirring hos slutanvändaren. Om till exempel item number laddas ur Movex-databasen utan restriktioner så skulle alla produktnummer som finns laddas upp under item number. Vid en grafisk presentation så går det inte att begränsa visualiseringen till de produktnummer som tillhör det finska projektet, vilket i detta fall skulle leda till att alla

produktnummer som finns i Movex-databasen visas. Följande exempelkod visar hur detta löstes i projektet;

Load

```
MMITNO as ItemNumber,  
Text(MMSPE5) as CUEan  
Where exists (CUEan, Text(MMSPE5))  
;
```

SQL SELECT

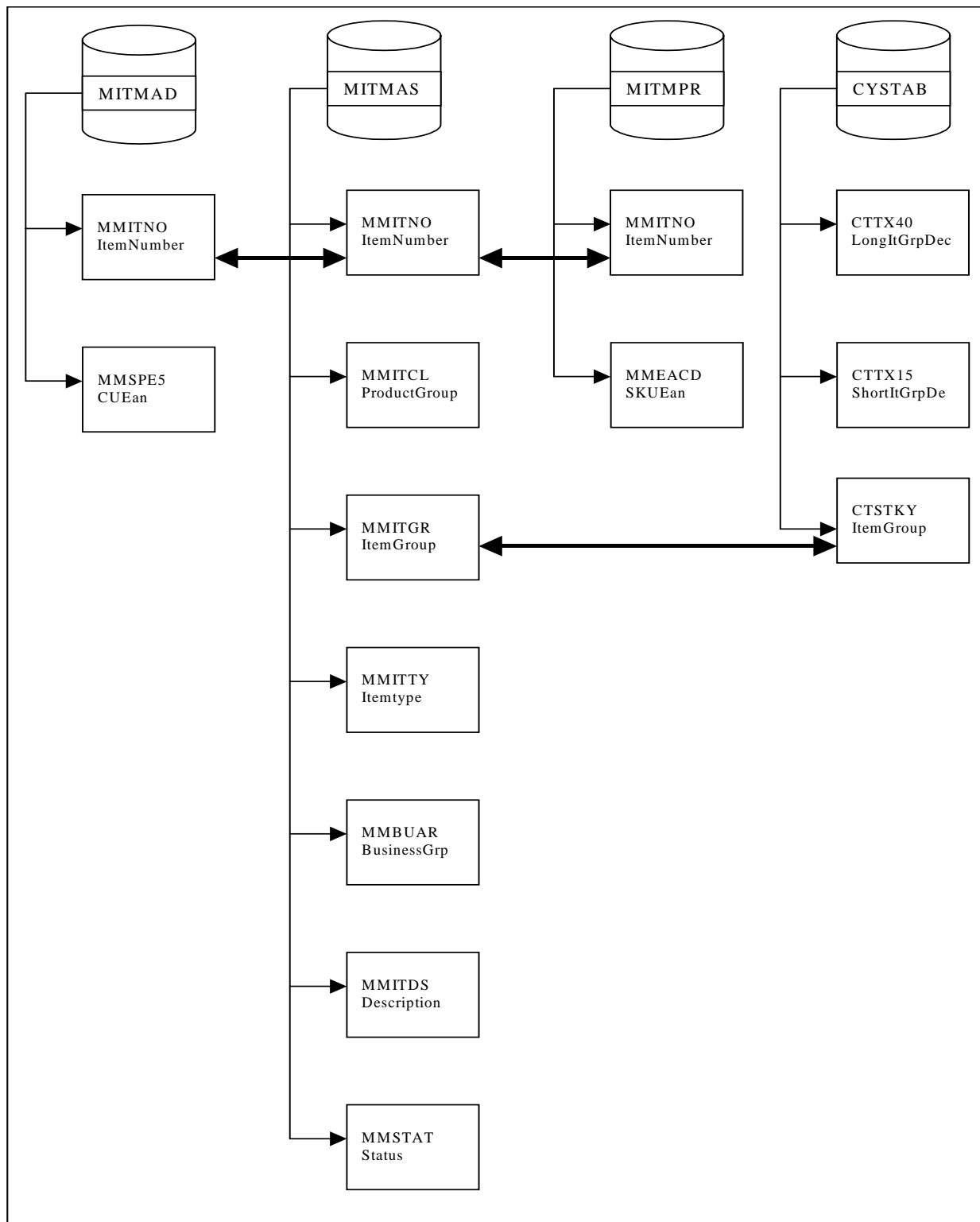
```
MMITNO,  
MMSPE5  
FROM MVXBDA.MITMAD;
```

Observera att detta är den första uppladdningen som sker ur Movex-databasen. Datahämtningen från Movex sker efter hämtningen ur Excel-dokumentet, vilket leder till att det är dess CUEan som är gällande när where-satsen tillämpas. Här hämtas således bara produktinfo från de produkter som återförsäljarna i Finland sålt eftersom det är dess konsumenthets-EAN som jämförs med konsumenthets-EAN i MITMAD-filen.

Det som sker i koden är att först väljs de fält ut som kan tänkas vara intressanta. SQL select satsen initierar detta val av fält. Sedan laddas dessa fält till QlikView genom load-satsen som gör urval med avseende på villkoret att CUEan från Excel-filerna matchar med MMSPE5-fältet (CUEan från Movex-databasen).

Urvalet måste sedan fortlöpa genom all filhämtning ur Movex. I och med att det finns nyckelfält i alla Movex-filer så är detta möjligt. Nyckelfälten gör att Movex-filerna har en gemensam nämnare vilket leder till att de blir sammanlänkade. Första urvalet gjordes med villkor som grundades från Excel-filernas data (CUEan == MMSPE5). CUEan kan alltså här ses som ett nyckelfält vilket binder samman de fälten som finns i POS-filerna med grunddatan i Movex-databasen.

Utöver kopplingen mellan Excel-filerna och MITMAD-filen så är Movex-filerna i projektet sammanlänkade på följande sätt när CYSTAB-fältet CTSTCO = "ITGR":

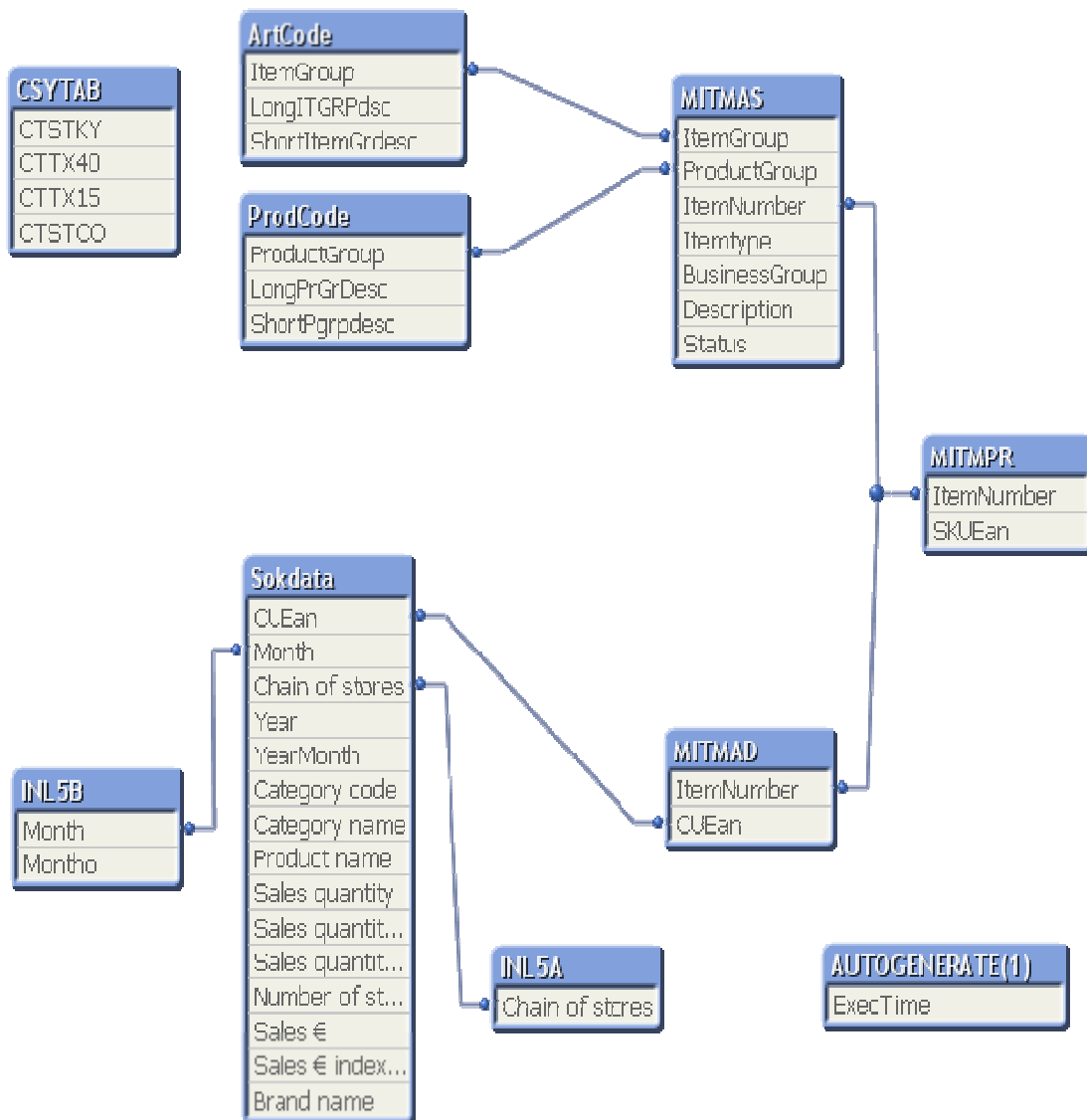


Figur 3-4: Movex-filernas sammanlänkning

När fält laddas ur MITMAS så är villkoret att gällande ItemNumber (MITMAD) == MMITNO (MITMAS). Eftersom produktnumren från MITMAD-filen är kopplade till Excel-

datan genom CUEan så kommer även de fält som plockas ur MITMAS vara relaterade till Excel-datan. På samma sätt hämtas även data från MITMPR och CSYTAB.

Slutligen så gav det färdiga QlikView-skriptet följande kopplingar;



Figur 3-5: QlikView-applikationens struktur

Som synes så ligger två skriptdelar löst, detta beror på att CYSTAB-filen laddas antingen som ArtCode eller ProdCode beroende på innehållet i CTSTCO-fältet. AUTOGENERATE är ingen uppladdning utan en skript-del som läser av senaste uppladdning av data. I detta avsnitt har jag ej berört inline-tabellerna INL5A och INL5B. En inline-tabell är en egen tabell som skapas i skriptet för olika ändamål. I detta fall var det för att få en tabell med butikskedjorna (INL5A) och för att månader skall skrivas 01,02,03 osv. och inte 1,2,3 osv. (INL5B).

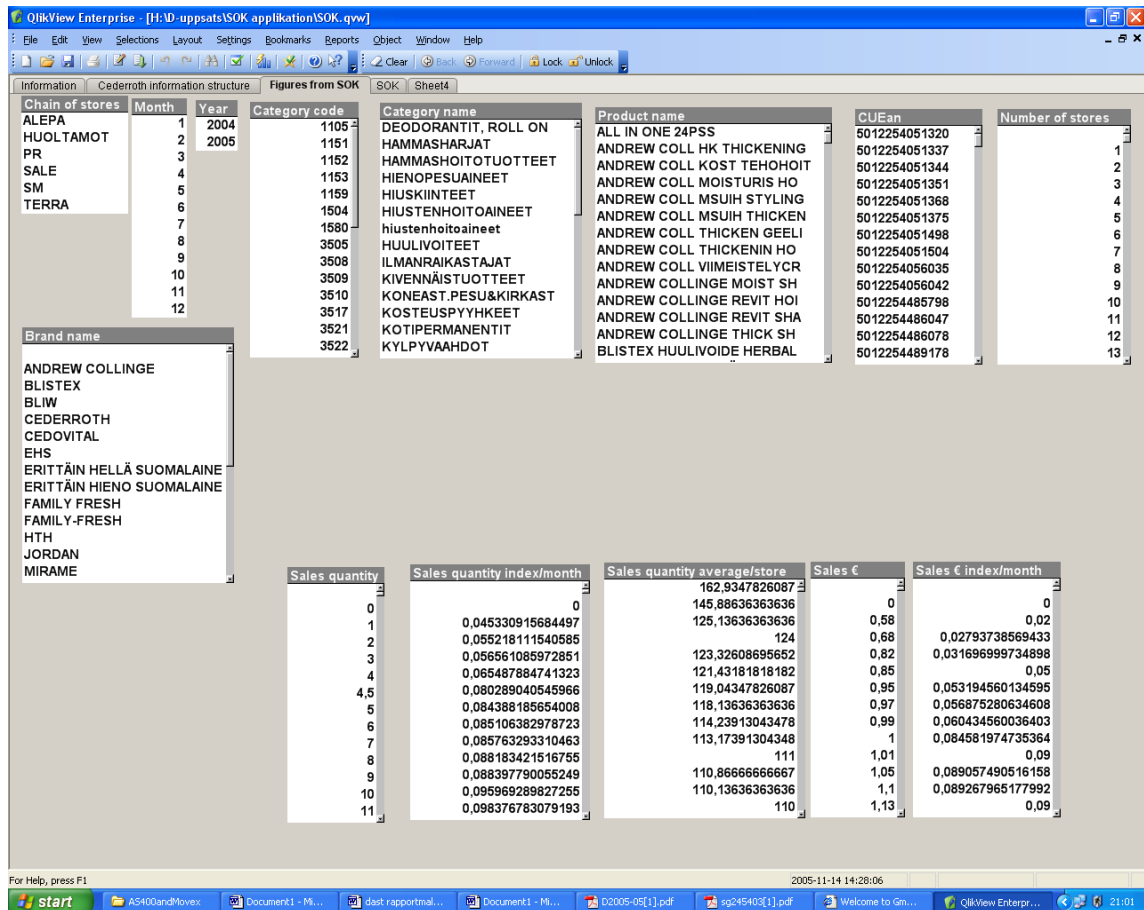
Kontinuerlig testning var ett krav från IT-avdelningen. Dock så gick det inte att utläsa indatafel direkt från skriptet. Om något fel uppstått i uppladdad data på grund av skriptet så får detta testas när applikationen är färdig. Att avgöra om skriptet är korrekt gick dock att testa genom exekvering.

3.6 Definition av grafiskt användargränssnitt

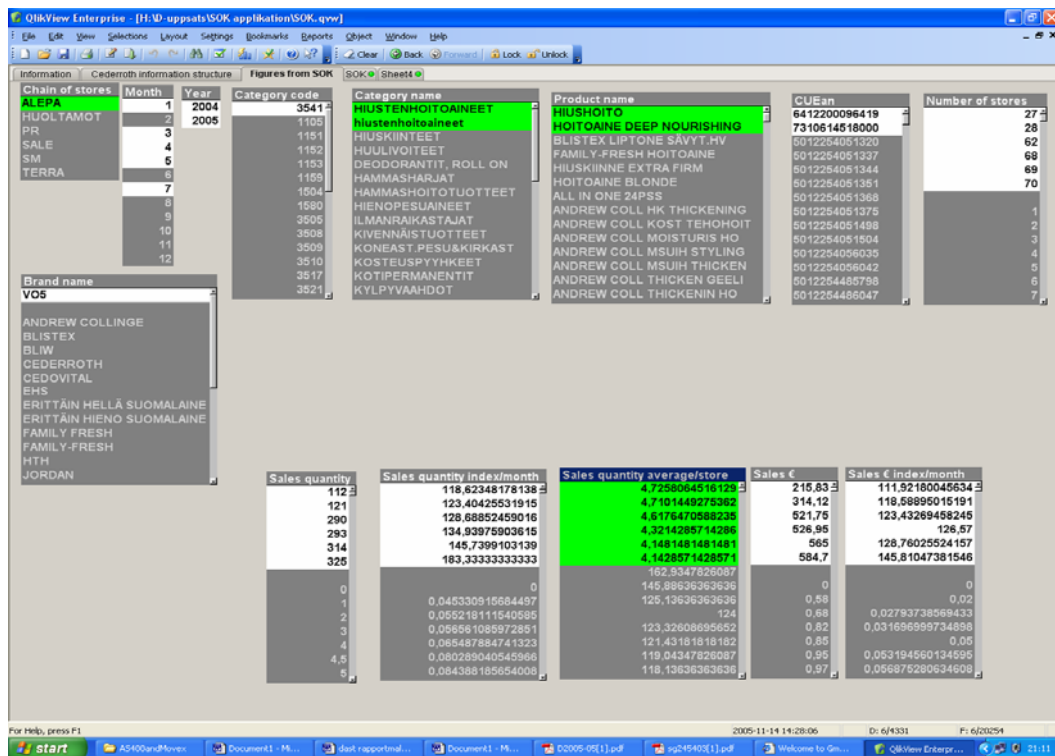
Efter att skriptet färdigställts var det dags att skapa det grafiska användargränssnittet. Att skapa ett grafiskt användargränssnitt från affärsdata utan några som helst krav från superanvändaren är tämligen problematiskt. Eftersom min huvudinriktning är informationsteknologi och inte företagsekonomi så blev analyserna av affärsdata från en lekman perspektiv. Men detta hindrade inte projektets gång i och med att beslut hade tagits att utveckla några olika ark med varierande presentation för att sedan få konkret kritik från den finska försäljningsenheten. Sedan visade det sig att användning av det sunda förnuftet räckte långt när användargränssnittet för återförsäljningsdata skapades.

För att få tillgång till data i en QlikView-applikation så krävs det att grafiska objekt placeras ut på ett QlikView-ark.

I projektet så var tanken att på det första arket skulle all data som laddas från Excel-filerna visas i listboxar. Detta var även en standard som företaget tillämpade i sina QlikView-applikationer. Det här tillvägagångssättet gjorde det även enkelt att testa om de uppladdade datakolumnerna var kopplade till varandra. Genom att testa några olika möjliga val så visar det sig ganska snart huruvida fälten har någon koppling. Följande skärmdumpar visar de två första stegen i utvecklingen av gränssnittet.

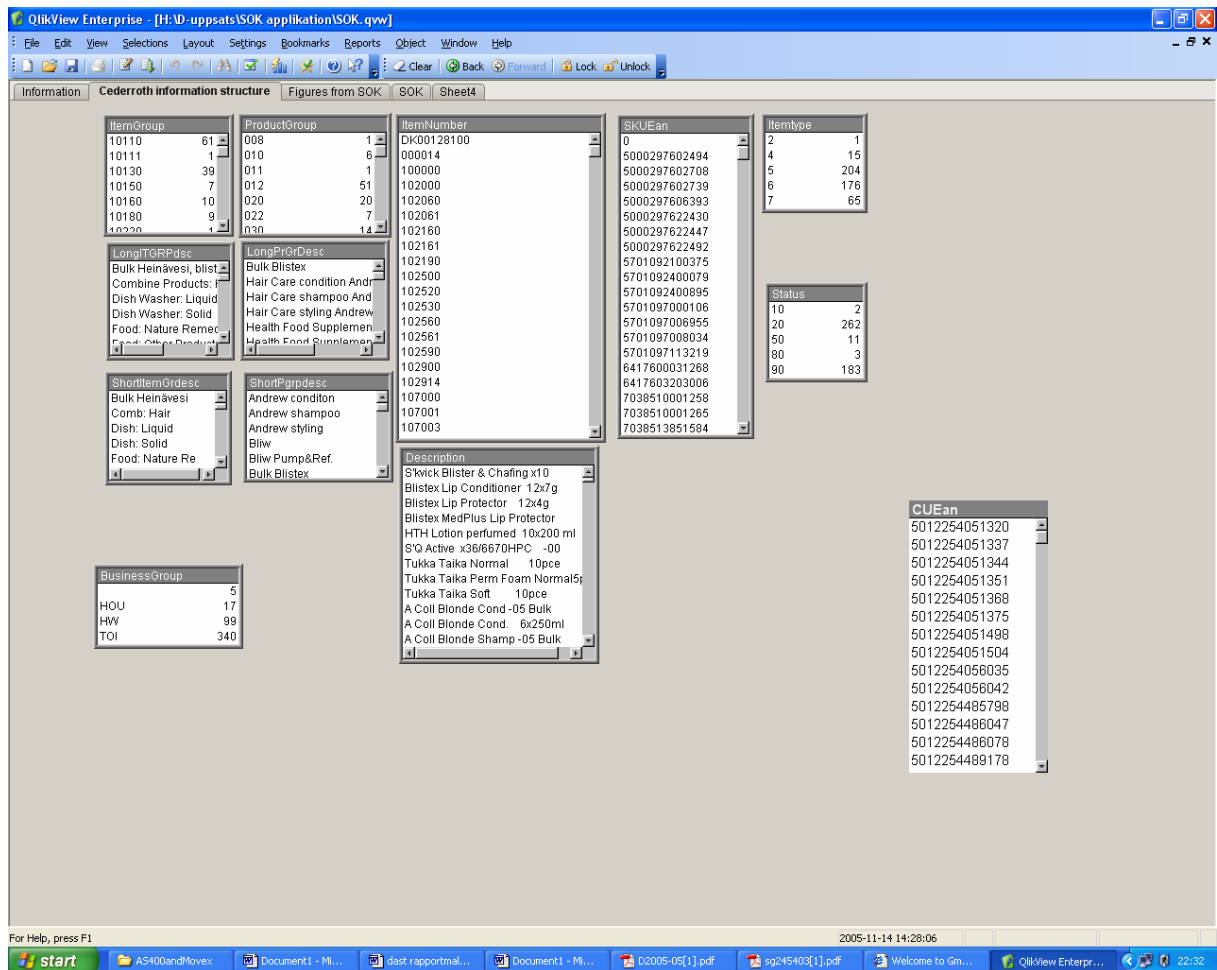


Figur 3-6: Finsk återförsäljningsdata

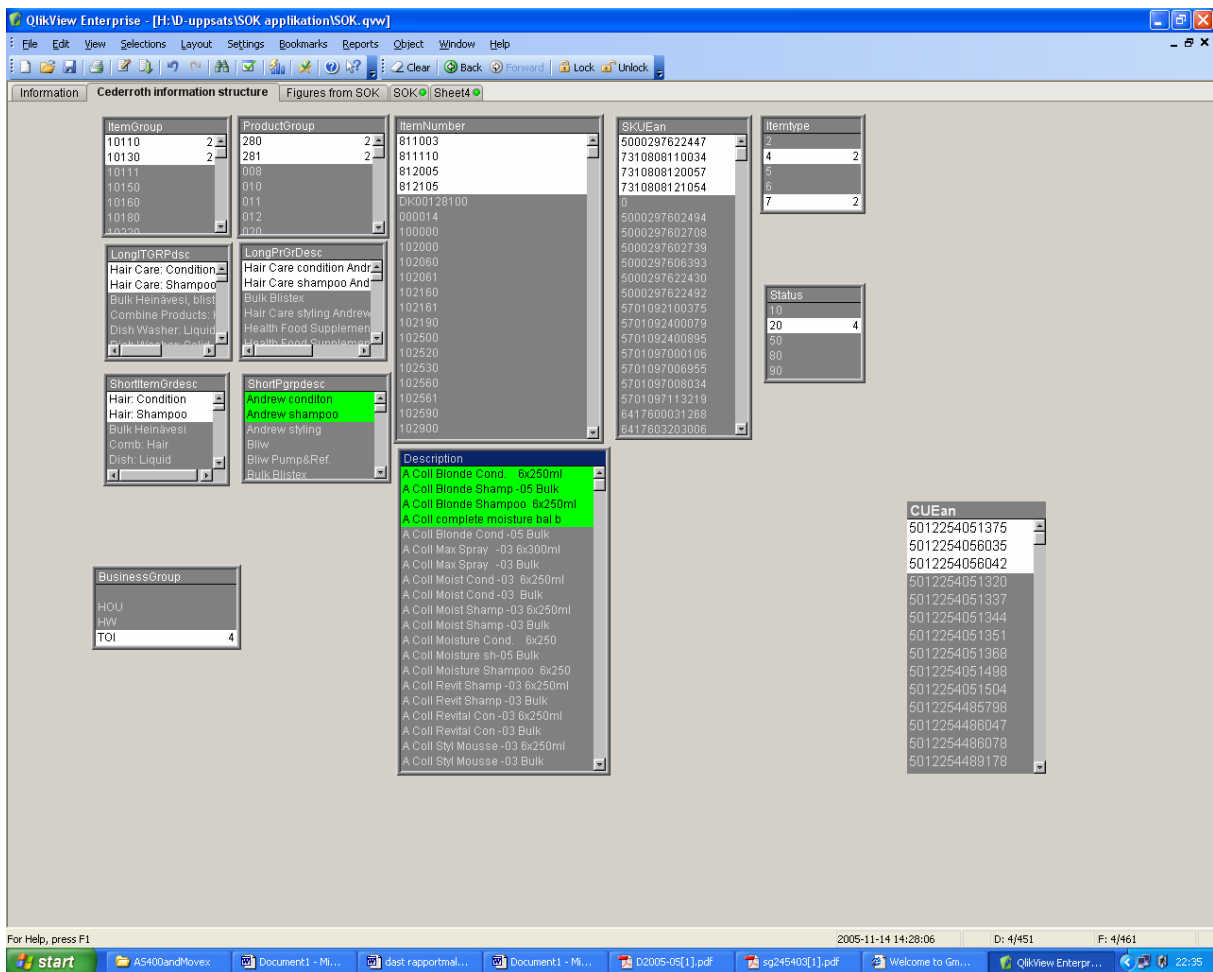


Figur 3-7: Exempeltest av Excel-datatkopplingar

Nästa steg kändes däreför naturligt eftersom även den data som laddas ur Movex bör presenteras. Det finns många olika sätt att presentera data på i en QlikView-applikation; listbox, statistikbox, multibox, tabellbox, diverse diagram, pivot tabeller, vanliga tabeller etc. Men för att hålla isär de olika fälten som laddats så ansågs listboxar vara mest passande. Här var det även lämpligt att testa så att uppladdade Movex-fält var kopplade till varandra.



Figur 3-8: Affärsdata från Movex



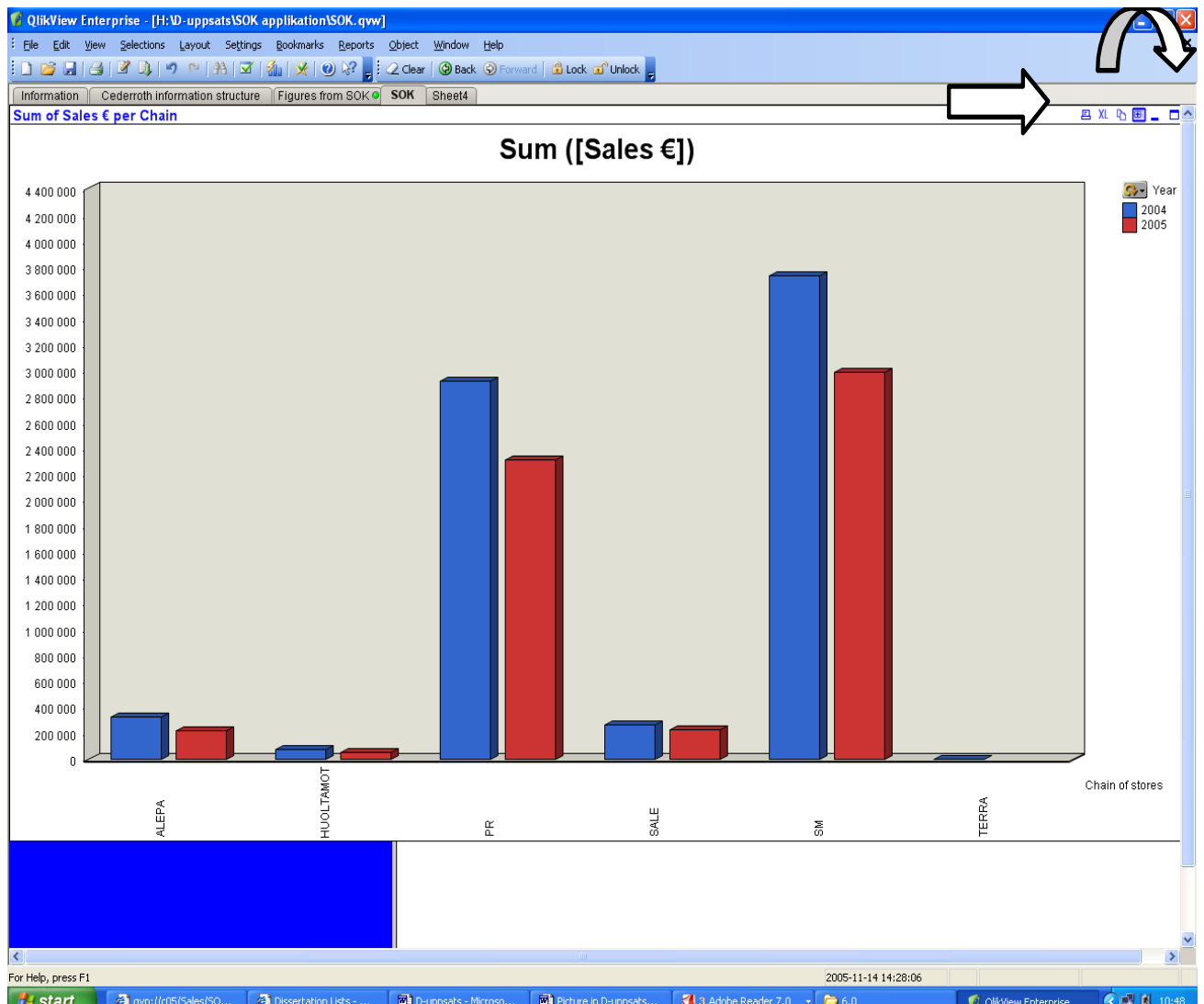
Figur 3-9: Exempeltest på Movex-datakopplingar

Skapandet av arken i figur 3-6 och 3-8 medför att superanvändaren får en bra överblick på vad för data som finns att arbeta med. Att ge en person som är novis inom BI-området en övergripande vy över vilken data som finns tillgänglig kan betyda A och O när applikationen senare skall ändamålsoptimeras.

De ark som hittills hade skapats ger en bra överblick men är inte tillräckliga för analys av försäljningsdata. Om data skall analyseras så underlättar det med diverse tabeller och diagram som ger ett bra underlag för analys. QlikView ger en mängd valmöjligheter inom tabell- och diagramområdet. Här gällde det bara att hitta de bäst anpassade tabellerna/diagrammen för den specifika affärsdatan som skulle presenteras.

Varje grafiskt objekt kan även ges möjlighet att skifta form, till exempel ett stapeldiagram kan skiftas till en tabell som kan skiftas till ett linjediagram osv. Det går även att ge möjlighet till kopiering, exportering till Excel och utskrift av innehållet i ett grafiskt objekt. Menyraden

på det grafiska objektet i *Figur 3-10* illustrerar de möjligheter som skapades för ”Sum of Sales € per Chain”-objektet i QlikView applikationen.



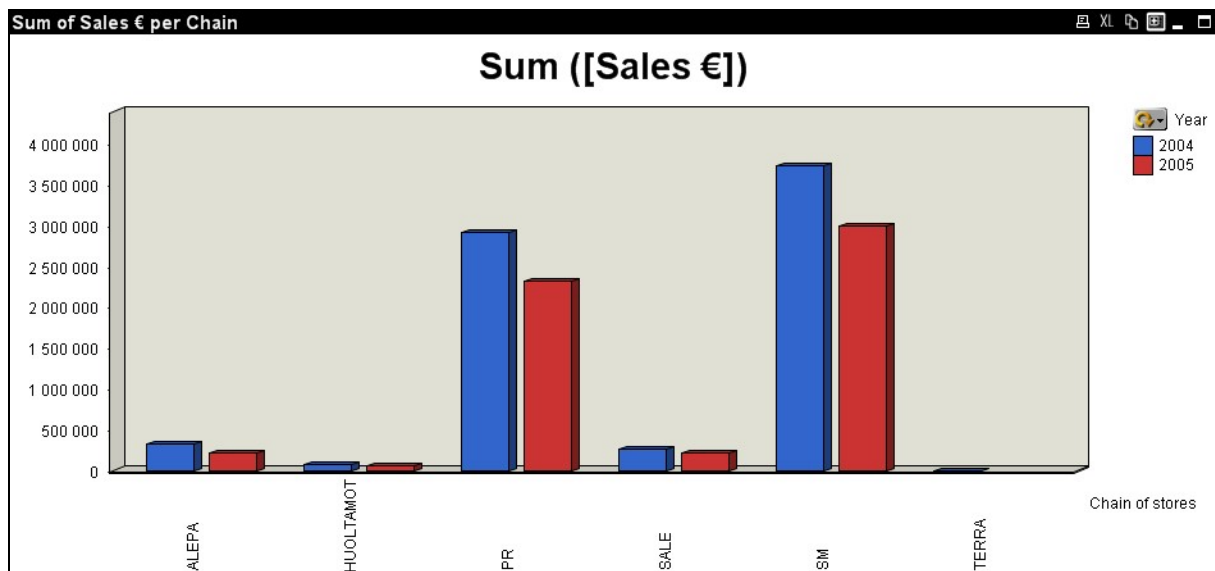
Figur 3-10: Valmöjligheter i grafisktobjekt

Figur 3-10 och *3-11* visar att från vänster till höger så är det utskrift, export till Excel, kopiering, byte av utseende, minimering och maximering.



Figur 3-11: Snabbvalsikoner

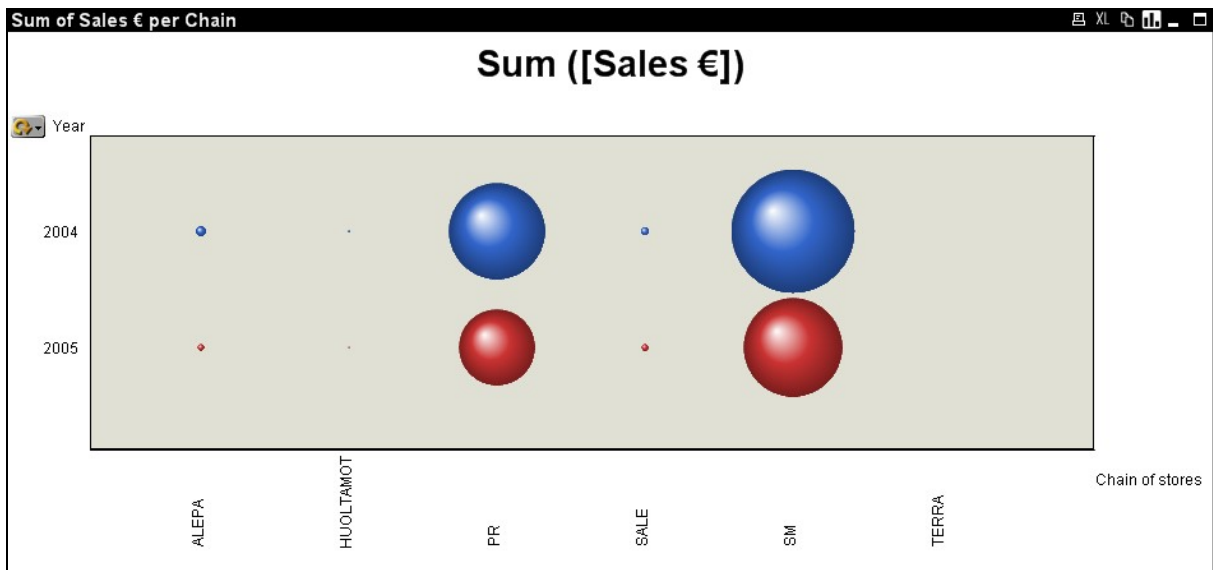
De utseenden som tilldelades detta grafiska objekt följer i sekventiell ordning i *figur 3-12 - 3-14*. Med ett enkelt klick på utseendebytesikonen så ges alltså möjlighet för slutanvändaren att se samma data i olika utformning.



Figur 3-12: Stapeldiagram

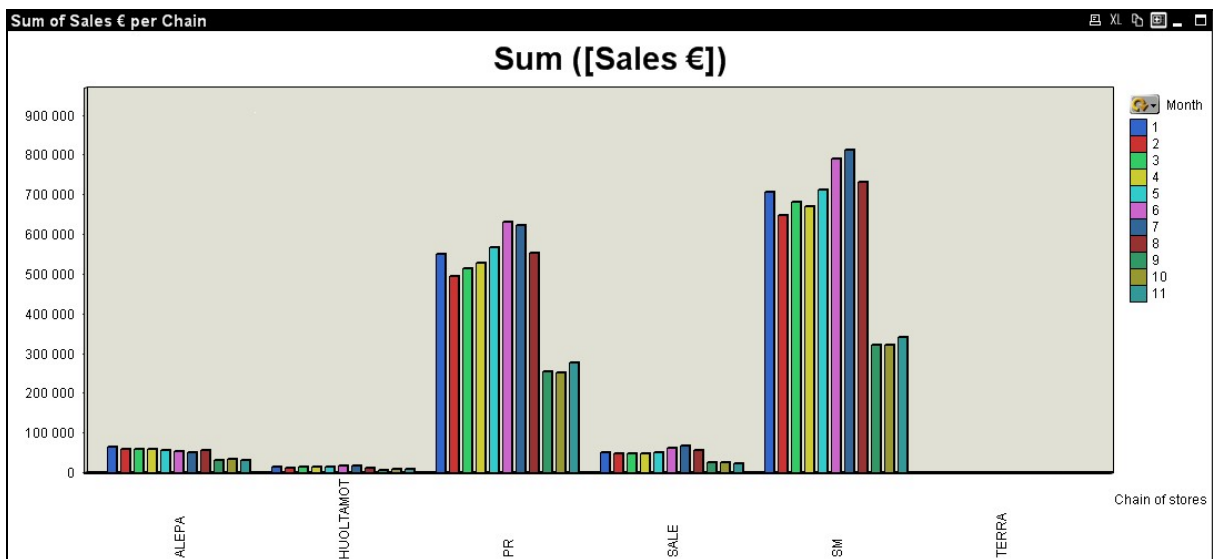
Chain of...	Year	Sum ([Sales €])
ALEPA	2004	328225,66
	2005	222889,6
HUOLTAMOT	2004	78914,48
	2005	56140,72
PR	2004	2924065,88
	2005	2314966,81
SALE	2004	269617,47
	2005	227264,63
SM	2004	3738242,49
	2005	2995888,91
TERRA	2004	49,03

Figur 3-13: Tabell



Figur 3-14: Gallerdiagram

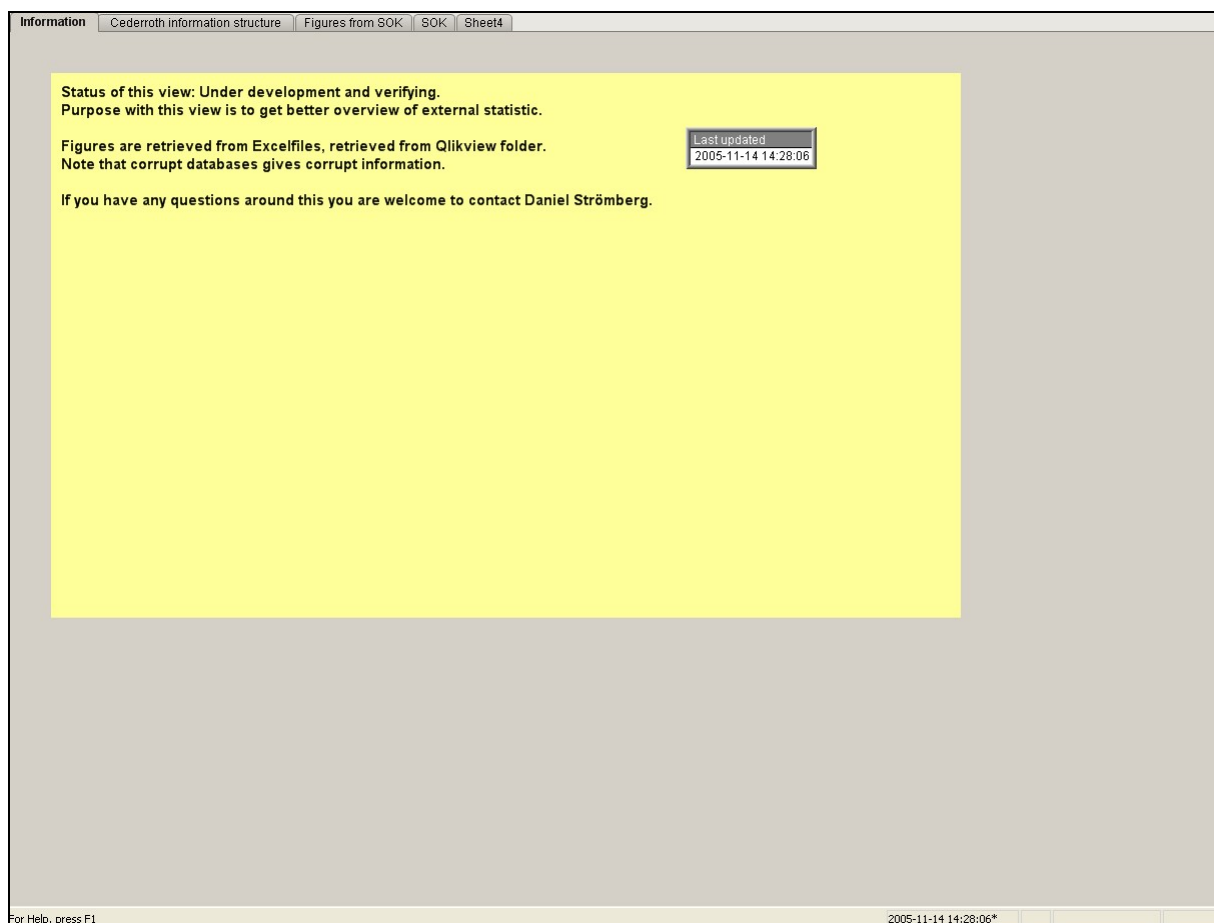
Det lades även till möjlighet att byta intervall från år till månad genom en intervallknapp som skapades i diagrammet/tabellen.



Figur 3-15: Stapeldiagram med månadsintervall

När arken med det mer affärsprocessinriktade innehållet skapades så utgicks det från tillvägagångssättet som fastslagits i början av projektet. Det vill säga, exponera så många alternativ på grafiska objekt som möjligt i olika ark. Men under utvecklingen så insågs ganska snart vad som kunde vara av intresse för den finska återförsäljningsenheten. En förhöjd grad av förståelse av affärsdata resulterade i en minskad grad av irrelevanta grafiska objekt.

Responser från Finland gav även den positiva indikationer om att utvecklingen var på rätt spår. Den slutgiltiga vyn som skulle testas och sedermera publiceras kom att se ut på följande sätt;



Figur 3-16: Slutvy ark 1

Information Cederroth information structure Figures from SOK SOK Sheet4

ItemGroup	ProductGroup	ItemNumber	SkUEan	Itemtype
10110	008	DK00128100	0	2
10111	010	000014	5000297602494	4
10130	011	100000	5000297602708	5
10150	012	102000	5000297602739	6
10160	020	102060	5000297606393	6
10180	022	102061	5000297622430	7
10220	030	102160	5000297622447	
		102161	5000297622492	
		102180	5701092100375	
		102500	5701092400079	
		102520	5701092400895	
		102530	5701097000106	
		102560	5701097008955	
		102561	5701097008034	
		102590	5701097113219	
		102900	6417600031268	
		102914	6417603203006	
		107000	7038510001258	
		107001	7038510001265	
		107003	7038513851584	

LongITGrpDesc	LongPrGrDesc	ShortItemGrDesc	ShortPrpDesc
Bulk Heinävesi, blist	Bulk Blistex	Bulk Heinävesi	Bulk Blistex
Combine Products:	Hair Care condition Andr	Comb: Hair	Andrew conditon
Dish Washer: Liquid	Hair Care shampoo Andr	Dish: Liquid	Andrew shampoo
Dish Washer: Solid	Hair Care styling Andrew	Dish: Solid	Andrew styling
Food: Nature Remec	Health Food Supplement	Food: Nature Re	Bliv
Food: Other Product			Bliv Pump&Ref.
			Bulk Blistex

BusinessGroup	
HOU	12
HW	17
TOI	341

Description
Swick Blister & Chafing x10
Blistex Lip Conditioner 12x7g
Blistex Lip Protector 12x4g
Blistex MedPlus Lip Protector
HTH Lotion perfumed 10x200 ml
SO Active x36/6670HPC -00
Tukka Taika Normal 10pce
Tukka Taika Perm Foam Normal5
Tukka Taika Soft 10pce
A Coll Blonde Cond -05 Bulk
A Coll Blonde Cond. 6x250ml
A Coll Blonde Shamp -05 Bulk

Status	
10	2
20	260
50	17
80	4
90	186

CUEan
5012254051320
5012254051337
5012254051344
5012254051351
5012254051368
5012254051375
5012254051498
5012254051504
5012254056035
5012254056042
5012254488798
5012254488047
5012254488078
5012254489178

Done Local intranet

Figur 3-17: Slutvy ark 2

Information Cederroth information structure Figures from SOK SOK Sheet4

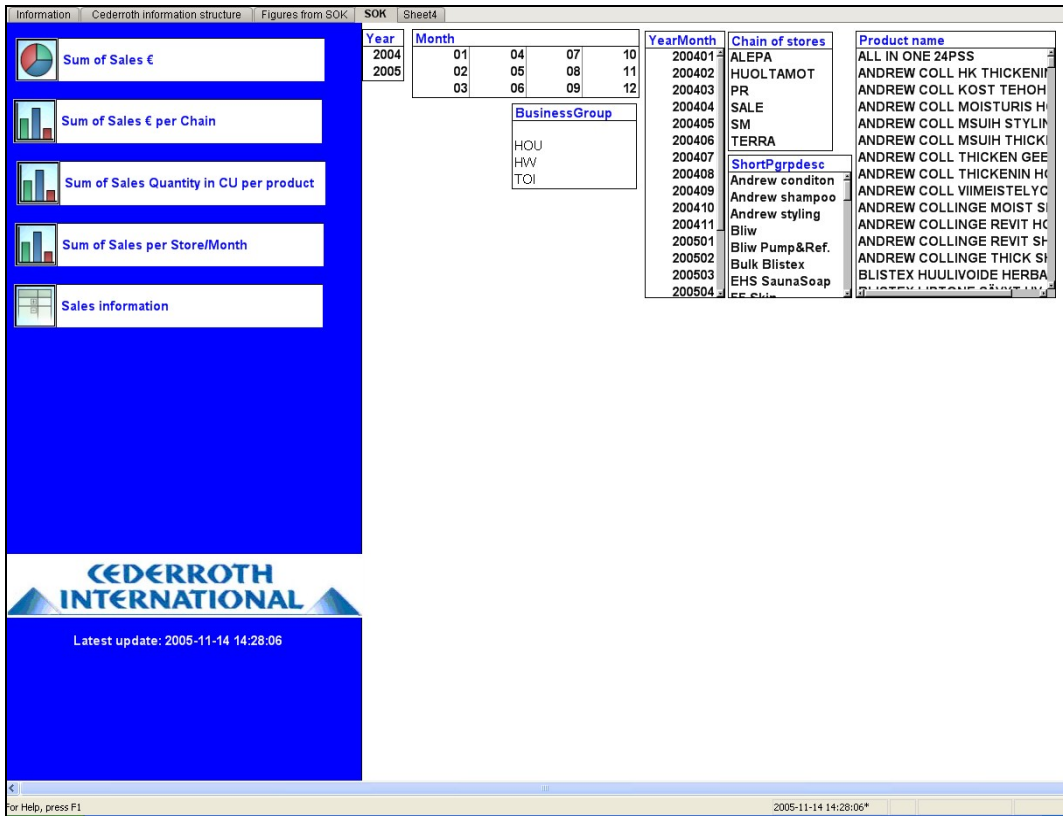
Chain of stores	Month	Year	Category code	Category name	Product name	CUEan	Number of stores
ALEPA	1	2004	1105	DEODORANTIT, ROLL ON	ALL IN ONE 24PSS	5012254051320	
HUOLTAMOT	2	2005	1151	HAMMASHARJAT	ANDREW COLL HK THICKENING	5012254051337	1
PR	3		1152	HAMMASHOITO TUOTTEET	ANDREW COLL KOST TEHOHOIT	5012254051344	2
SALE	4		1153	HIENOPESUAINHEET	ANDREW COLL MOISTURIS HO	5012254051351	3
SM	5		1159	HIUSKIINTEET	ANDREW COLL MSUIH STYLING	5012254051368	4
TERRA	6		1504	HIUSTENHOITOAINHEET	ANDREW COLL MSUIH THICKEN	5012254051375	5
	7		1580	hiustenhoitoaineet	ANDREW COLL THICKEN GEELI	5012254051498	6
	8		3505	HUULIVOITEET	ANDREW COLL THICKENIN HO	5012254051504	7
	9		3508	ILMANRAIKASTAJAT	ANDREW COLL VIMEISTELYCR	5012254056035	8
	10		3509	KIVENNÄISTUOTTEET	ANDREW COLLINGE MOIST SH	5012254056042	9
	11		3510	KONEAST.PESU&KIRKAST	ANDREW COLLINGE REVIT HOI	5012254486047	10
	12		3517	KOSTEUSPYHKEET	ANDREW COLLINGE REVIT SHA	5012254486078	11
			3521	KOTIPERMANENTIT	ANDREW COLLINGE THICK SH	5012254486078	12
			3522	KYLPYVAHADOT	BLISTEX HUULIVOIDE HERBAL	5012254489178	13

Brand name
ANDREW COLLINGE
BLISTEX
BLIW
CEDERROTH
CEDOVITAL
EHS
ERITTÄIN HELLÄ SUOMALAIN
ERITTÄIN HIENO SUOMALAIN
FAMILY FRESH
FAMILY-FRESH
HTH
JORDAN
MIRAME

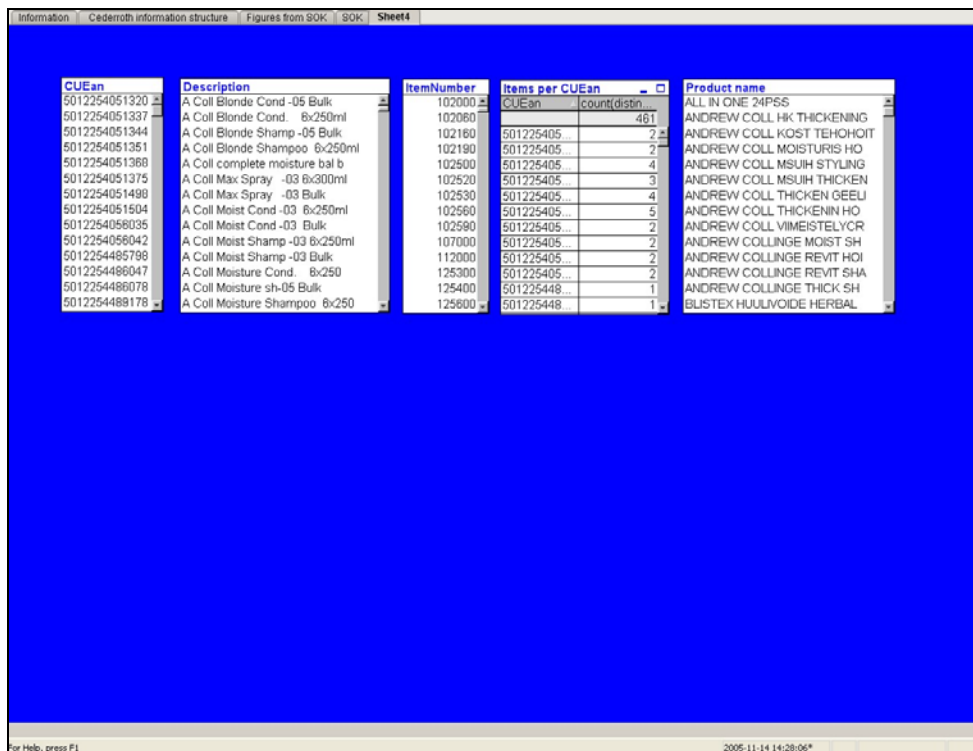
Sales quantity	Sales quantity index/month	Sales quantity average/store	Sales €	Sales € index/month
0	0	162,9347826087	0	0
1	0,045330915684497	145,88636363636	0,58	0,02
2	0,055218111540585	125,13636363636	0,68	0,02793738569433
3	0,056561085972851	123,32608695652	0,82	0,031696999734898
4	0,065487884741323	121,43181818182	0,85	0,05
4,5	0,080289040545966	119,04347826087	0,95	0,053194560134595
5	0,084388185654008	118,13636363636	0,97	0,056875280634608
6	0,085106382978723	114,23913043478	0,99	0,060434560036403
7	0,085763293310463	113,17391304348	1	0,084581974735364
8	0,088183421516755	111	1,01	0,09
9	0,088397790055249	110,86666666667	1,05	0,089057490516158
10	0,095969289827255	110,13636363636	1,1	0,089267965177992
11	0,098376783079193	110	1,13	0,09

For Help, press F1 2005-11-14 14:28:06*

Figur 3-18: Slutvy ark 3



Figur 3-19: Slutvy ark 4



Figur 3-20: Slutvy ark 5

Dessa 5 ark ansågs vara tillräckliga för analys av den externa återförsäljningsdatan. Skulle dock fler objekt dyka upp som krav så är det förhållandevis enkla kompletteringar som krävs.

Ark 1 (figur 3-16)

”Information” var det namn som ark 1 tilldelades. Den enkla orsaken till det var att information är just vad arket innehåller. Informationen som ges är vilken status, vilket syfte och vilken datakälla applikationen har. Det ges även information om vem som frågor angående applikationen skall ställas till.

Statusen vid GUI-definieringsprocessen i projektet var att applikationen var under konstruktion. Information om att syftet med vyn var att få bättre överblick på extern statistik tillkännagavs också. Att data var hämtad från Excel-filer är visserligen ganska diffust men det ansågs räcka som information, superanvändaren är ju fullt medveten om var affärsdatan kommer ifrån. Vid frågor kunde jag kontaktas vilket även informerades om i Ark 1.

Ark 2 (figur 3-17)

”Cederroth Information structure” angavs som namn för ark 2. Innehållet var den information som plockats från Movex-databasen presenterad i listboxar. Se *figur 3-3* för mer ingående förklaring om vad som laddats upp från Movex och här presenterats.

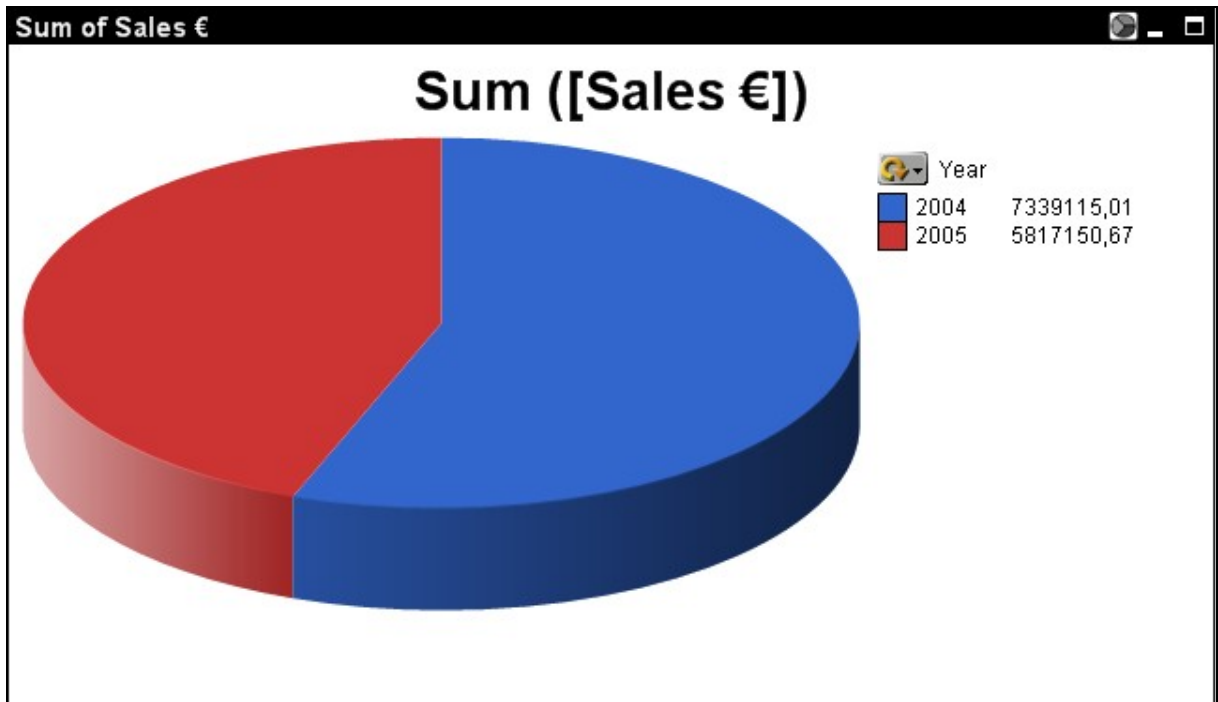
Ark 3 (figur 3-18)

Ark 3 fick heta ”figures from SOK” eftersom det var den finska benämningen på inkommande POS-data. I detta ark presenterades den data som hämtas från inkommande Excel-filer i listboxar. Se *tabell 3.1* för information om uppladdad POS-data.

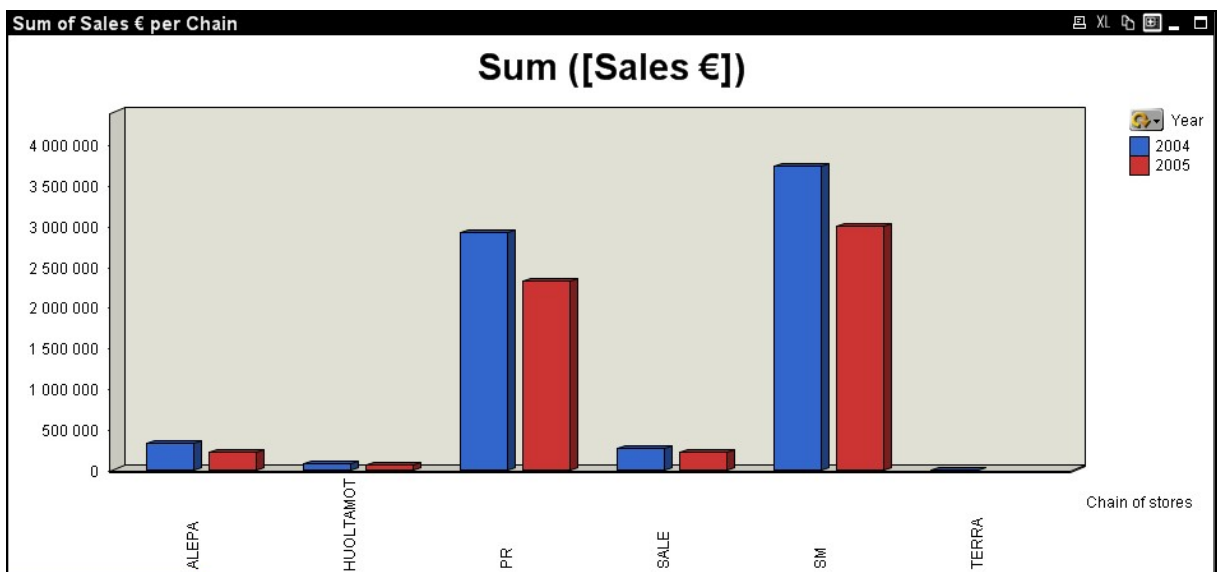
Ark 4 (figur 3-19)

Detta ark kan ses som huvudarket för hela applikationen. Benämningen på arket fick bli SOK, och arket angavs som startark i den kommande webbpresentationen. Följande listboxar tycktes vara lämpliga för vidare förfrågningar samt underlag för diagram/tabell-presentation; år, månad, årsmånad (200512), affärsgrupp, butikskedja, kort produktgruppsbeskrivning och produktnamn.

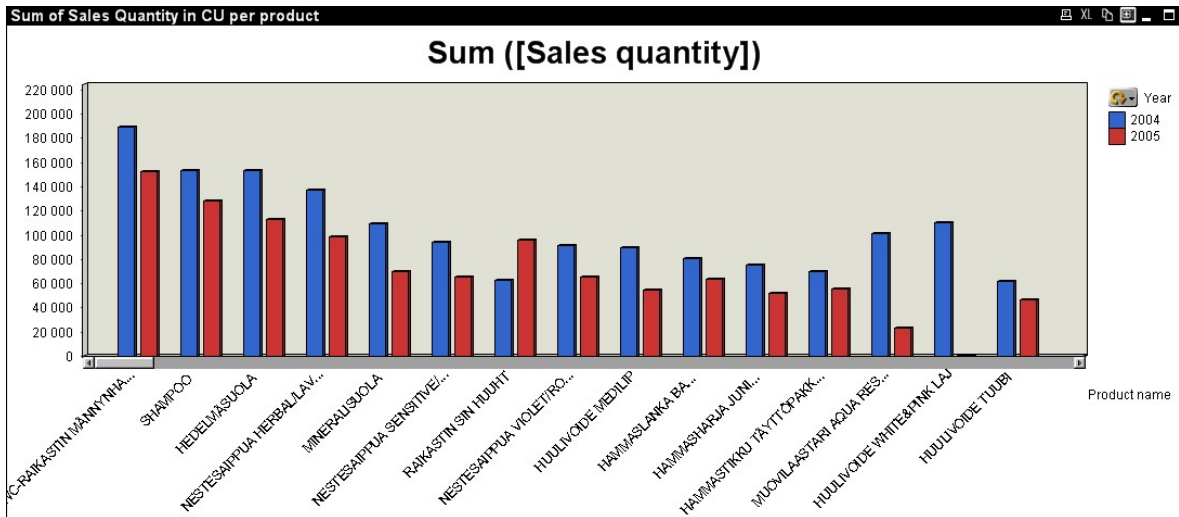
De diagram/tabell-objekt som ingick i arket var <Summa av försäljning i €>, <Summa av försäljning i € per butikskedja>, <Summa försäljningskvantitet i konsumentenhet per produkt>, <Summa försäljning per affär/månad> och <Försäljningsinformation>. Dessa visas i *figur 3-21 – 2-25*.



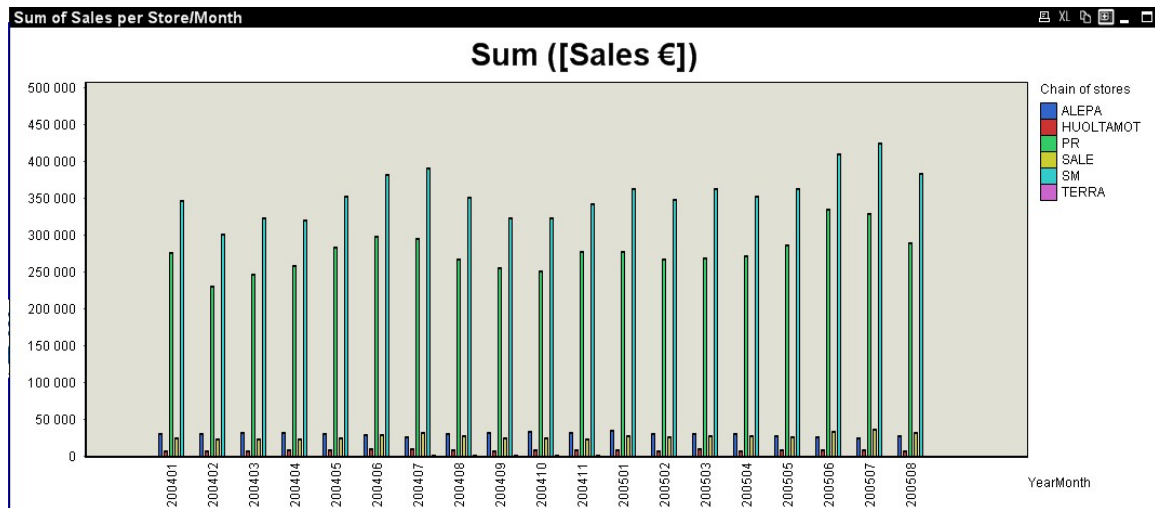
Figur 3-21: Summa av försäljning €



Figur 3-22: Summa av försäljning € per butikskedja



Figur 3-23: Summa försäljningskvantitet i konsumentenhet per produkt



Figur 3-24: Summa försäljning per affär/månad

Sales information									
Chain of...	YearMonth	Product name	Sales quantity	Sales quantity average/store	Sales quantity index/month	Sales €	Sales € index/month	Sales €	
ALEPA	200401	HAMMASHARJA...	733	12,2166666666667	121,15702479339	1091,58	121,15205327414	1 091,580	
		HAMMASHARJA...	1	1	100	2,84	100	2,840	
		HAMMASHARJA...	139	2,622641509434	2780	433,37	2587,2835820895	433,370	
		HAMMASHARJA...	271	4,8392857142857	9033,33333333333	844,29	8400,8955223881	844,290	
		HAMMASLANKA...	652	10,030789230769	385,79881656805	1948,67	385,84468556948	1 948,670	
		HAMMASLANKA...	345	5,855737704918	278,22580645161	1362,15	278,51271775579	1 362,150	
		HAMMASLANKA...	141	2,5636363636364	-	-	566,47	-	566,470
		HAMMASLANKA...	52	1,4054054054054	-	-	233,72	-	233,720
		HAMMASTIKKU...	632	9,7230769230769	-	-	1610,72	-	1 610,720
		HAMMASTIKKU...	34	4,8571428571429	18,579234972678	117,3	18,593960529444	117,300	
		HEDELMASUOLA	778	11,969230769231	107,45856353591	2730,8	111,70743679948	2 730,800	
		HIENOPESUJAU...	224	4,1481481481481	115,46391752577	731,58	116,86581469649	731,580	
		HIUSGEELI...	45	2,25	-	-	191,12	-	191,120
		HIUSHOITO	293	4,7258064516129	118,62348178138	526,95	118,58895015191	526,950	
		HOITOAINE DEE...	83	3,32	-	-	248,17	-	248,170
		HOITOAINE...	111	4,1111111111111	-	-	331,71	-	331,710
		HOITOAINE...	106	4,0769230769231	-	-	308,57	-	308,570
		HUULIVOIDE...	873	13,857142857143	137,89718088328	2607,84	137,7324509747	2 607,840	
		HUULIVOIDE...	1287	20,109375	100,86206896552	1671,74	100,87981849464	1 671,740	
		JORDAN...	9	9	2,4657534246575	29,87	2,4589421691708	29,870	
KANGASLAASTA...	570	9,047619047619	124,72647702407	1309,95	124,74288652725	1 309,950			
MEIKKIPIUKKO	23	2,5555555555556	11,330049261084	58,57	11,455338457627	58,570			
MINERALISUOLA	463	7,3492063492063	120,88772845953	1004,15	120,90181205226	1 004,150			
MUOVILAASTARI...	548	8,4307692307692	137,34335839599	1259,58	137,36926482938	1 259,580			
NESTESAIFFUA...	1601	25,015625	141,55614500442	1903,84	141,53050149422	1 903,840			

Figur 3-25: Försäljningsinformation

Med urval ur listboxarna så tycktes dessa diagram/tabeller täcka den mest vitala informationen. Skulle det även behövas direkt information ur Excel-filerna som erhållits så presenteras denna på ett strukturerat sätt i tabellobjektet för försäljningsinformation.

Ark 5 (figur 3-20)

I applikationen så fick detta ark ha defaultnamnet sheet 4 eftersom den data som presenteras här inte ger väsentlig affärsinformation. Arket är till för att visa att flera olika produktnummer har samma konsumentenhets-EAN. Det är visserligen samma produkt men den är emellertid paketerad på olika sätt.

Dessa fem ark och den bakomliggande applikationen bedömdes vara korrekta för tillämpning enligt superanvändaren, så nu återstod bara testning, säkerhets konfigurering och därefter publicering innan utvärderingen kunde göras.

3.7 Testning

När applikationen nu var godkänd till utseendet så var det dags att testa att informationen som återfås inte var korrupt.

Under denna del av utvecklingsfasen så bör hela applikationen testas utifrån olika perspektiv. Följande perspektiv var viktiga att ta hänsyn till:

1. Data som presenteras i vyn ska stämma överens med den data som återfinns i Movex-databasen och i Excel-dokumentet.
2. Korrekta kopplingar skall ha gjorts mellan Movex- och Excel-data
3. De olika arken skall vara sammanlänkade.
4. Användaren skall ha tillgång till de grafiska objekt som är av vikt.
5. Uppdatering skall fungera.
6. Webbgränssnittet ska stämma överens med originalapplikationen.

3.7.1 Överensstämmelse mellan presenterad data och grunddata

När detta test skulle utföras så var det bara att gå till grunddata i Movex-databasen och Excel-filerna och jämföra med den data som presenteras i vyn. Genom att skapa förfrågningar och sedan jämföra dessa resultat med grunddatan så kunde det avgöras att data var korrekt uppladdad.

Figur 3-26: Test-förfrågan i QlikView (ark 3)

	A	B	C	D	E	F	G	H	I	J	K	L
1	KK	TR4	Ryhmä	Tuote	EAN	Kauppa	Myynti kpl	My kpl ind kk	Myynti kpl ka per ky	Ketjuyks peitto lkm	Myynti €	My ind kk
2	200401	3521	nestesaippuat	BLIW CREAMY PUMPPUNESTESA	6412200096419	SM	1	1,44927536232	1	1	1,69	1,22677119628338

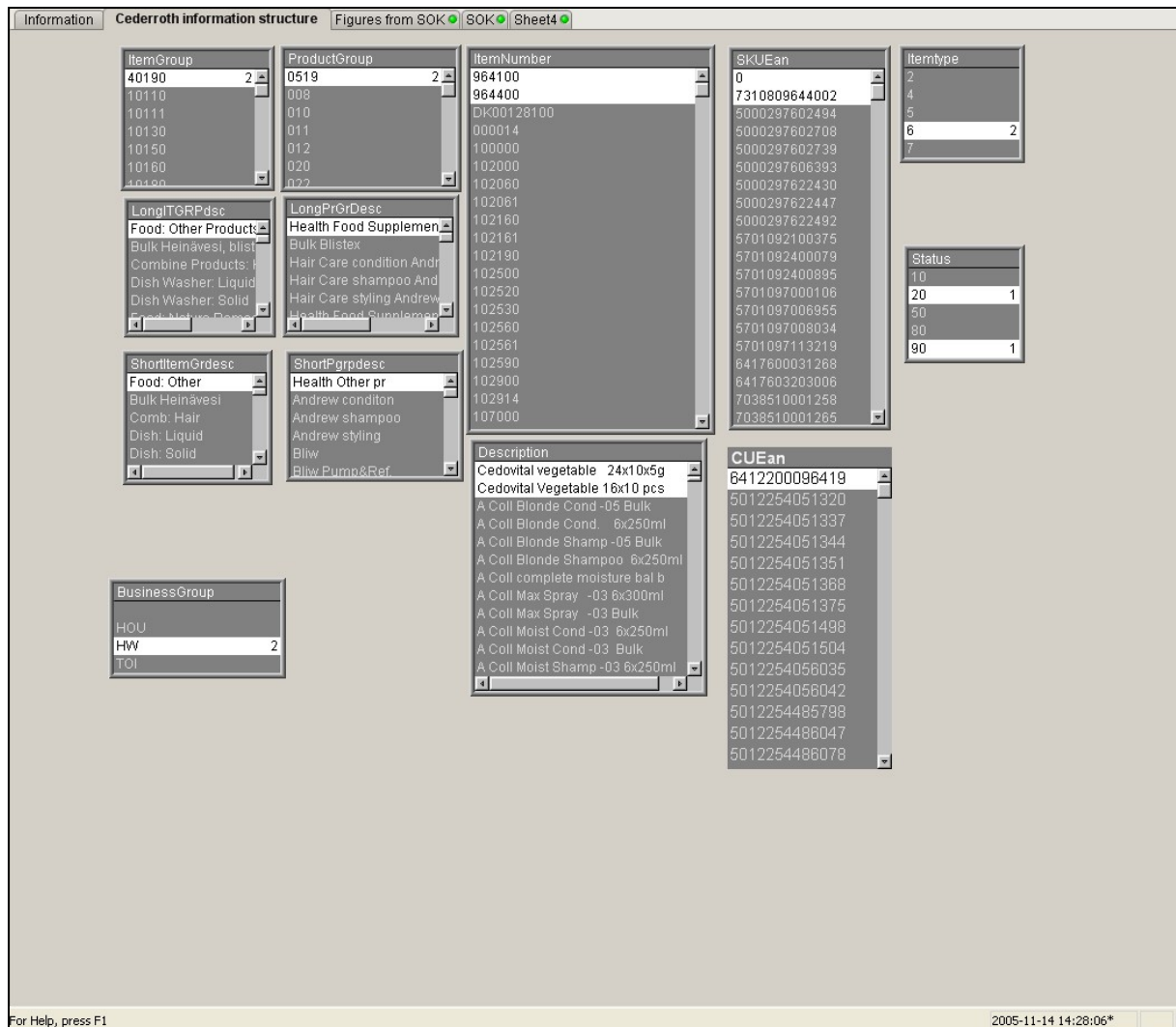
Figur 3-27: Data från Excel

Figur 3-26 och figur 3-27 visar ett testexempel för verifiering av presenterad data.

Efter X antal olika förfrågningar och djupdykningar i grunddatan så insågs att den visade datan stämde överens med den data som skulle ha hämtats från Movex- och QlikView(Excel-filer)-databasen.

3.7.2 Test av koppling Movex-Excel

Tabell-vyn i *figur 3-5* visar att Movex-data och Excel-data är associerat. Men för att testa så att det verkligen stämmer så kan man genom att skifta mellan ark 2 och 3 se kopplingen.



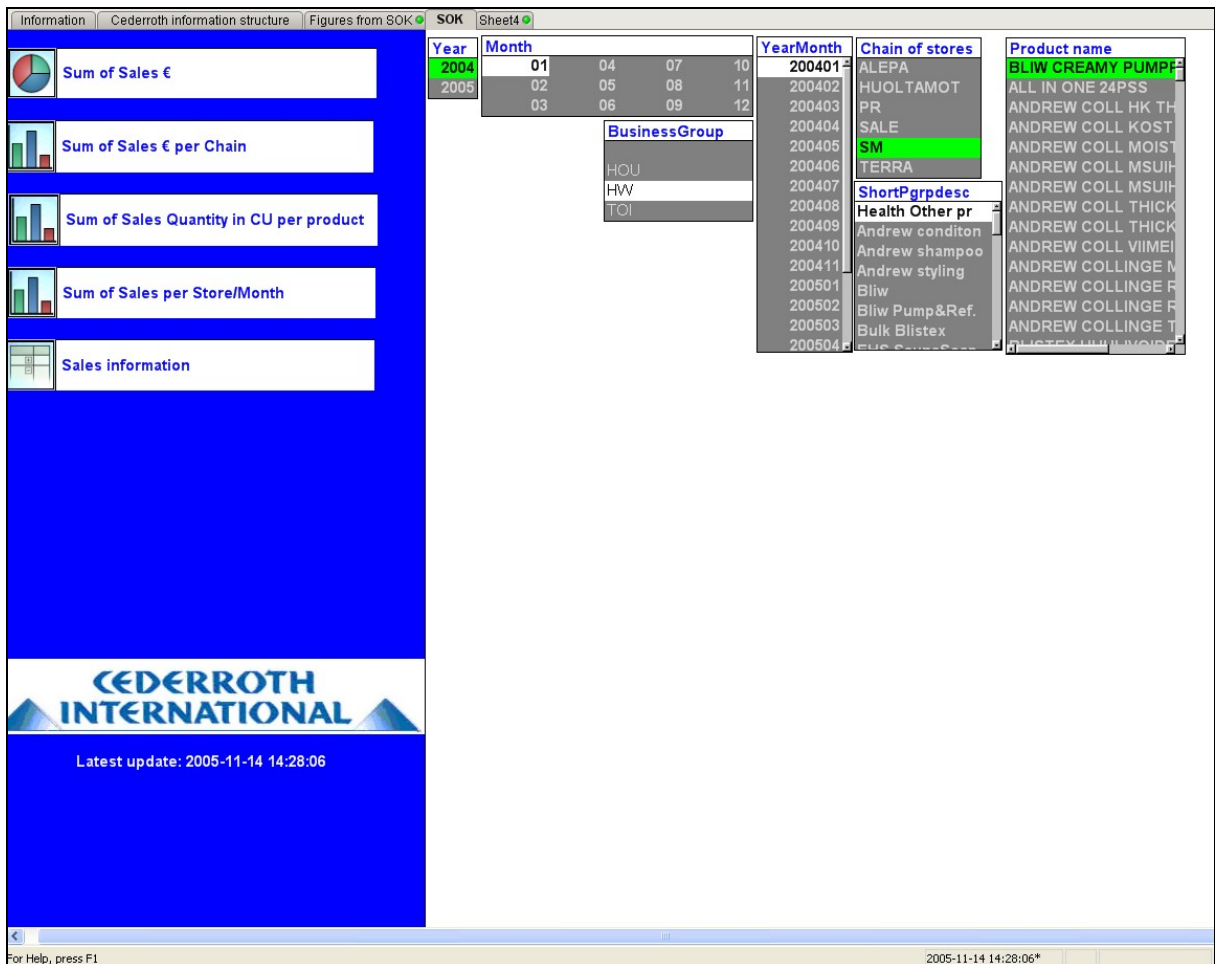
Figur 3-28: Movex-data ark 2

Om urvalen i *figur 3-26* (ark 3) kvarstår så ges associerad data i *figur 3-28*. Som synes så är Movex-datan kopplad till Excel-datan i detta exempel.

Denna typ av testning gjordes på ett antal olika förfrågningar på affärsdata från både Movex och Excel. Testningen gick igenom utan avvikelser vilket visar att kopplingen är korrekt.

3.7.3 Testning av sammanlänkning mellan ark

Föregående test visade att data från Movex och Excel har kopplingar men det visar också att ark 2 och 3 är korrekt sammanlänkade. I och med att ark 1 enbart innehåller skriven information om applikationen så var det bara ark 4 och ark 5 som behövde testas. Ark 4 och ark 5 innehåller överlagrad data från både ark 2 och 3, så dess data bör överensstämma med data från dessa två ark för att vara riktig.



Figur 3-29: Testexempel ark 4

CUEan	Description	ItemNumber	Items per CUEan	Product name
6412200096419	Cedovital vegetable 24x10x5g	964400	CUEan	BLIW CREAMY PUMPPUNESTESA
5012254051320	Cedovital Vegetable 16x10 pcs	964100	count(distin...	ALL IN ONE 24PSS
5012254051337	A Coll Blonde Cond -05 Bulk	906700	641220009...	ANDREW COLL HK THICKENING
5012254051344	A Coll Blonde Cond. 6x250ml	913000		ANDREW COLL KOST TEHOHOIT
5012254051351	A Coll Blonde Shamp -05 Bulk	812120		ANDREW COLL MOISTURIS HO
5012254051368	A Coll Blonde Shampoo 6x250ml	811120		ANDREW COLL MSUIH STYLING
5012254051376	A Coll complete moisture bal b	812100		ANDREW COLL MSUIH THICKEN
5012254051498	A Coll Max Spray -03 6x300ml	811100		ANDREW COLL THICKEN GEELI
5012254051504	A Coll Max Spray -03 Bulk	811110		ANDREW COLL THICKENIN HO
5012254056035	A Coll Moist Cond -03 6x250ml	811015		ANDREW COLL VIMEISTELYCR
5012254056042	A Coll Moist Cond -03 Bulk	451805		ANDREW COLLINGE MOIST SH
5012254485798	A Coll Moist Shamp -03 6x250ml	811205		ANDREW COLLINGE REVIT HOI
5012254486047	A Coll Moist Shamp -03 Bulk	102914		ANDREW COLLINGE REVIT SHA
5012254486078	A Coll Moisture Cond. 6x250	811002		ANDREW COLLINGE THICK SH

Figur 3-30: Testexempel ark 5

Figur 3-29 och 3-30 visar att valet i figur 3-26 även är sammanlänkat med ark 4 och ark 5. Som tidigare så testades detta på flera olika förfrågningar och utfallet var det samma vilket gjorde att slutsatsen blev att arken var sammanlänkade.

3.7.4 Användartillgång till grafiska objekt

Under denna testdel så var det tvunget att publicera applikationen i webbgränssnittet. Eftersom slutanvändaren bara har tillgång till webbgränssnittet så är det där tillgängligheten måste testas.

En slutanvändare skall bara ha tillgång till de objekt som innehåller affärsdata dvs. listboxar, grafer och tabeller i det här fallet. Bakgrund, bilder och text skall ej ges tillåtelse att ändras på. Genom att testa alla grafiska objekt så verifierades att slutanvändaren hade tillgång till ”rätt” objekt.

3.7.5 Uppdateringstest

Att testa uppdateringen görs genom att lägga till en ny Excel-fil i QlikView-databasen och sedan kontrollera om tillhörande affärsdata läggs till i applikationen när återladdning sker. Det ska även ske en uppdatering i det fält som visar när applikationen senast uppdaterats (datum, tidpunkt).

Uppdateringstestet gick även det igenom smärtfritt, mycket på grund av att detta test indirekt hade gjorts när skriptet implementerades.

3.7.6 Webbgränssnitt vs originalapplikation

Här behövdes det bara kollas så att all presentation av data i originalapplikationen även presenterades i webbgränssnittet och det gjorde det.

3.7.7 Testsammanfattning

Testningen gav inga fel, detta berodde huvudsakligen på att testning skett kontinuerligt under de föregående stegen i projektflödet. En annan anledning kan vara att projektet genomförts noggrant vilket är väldigt viktigt när man hanterar affärsdata som kan komma att användas för framtida affärsplanering.

3.8 Säkerhetskongfiguration

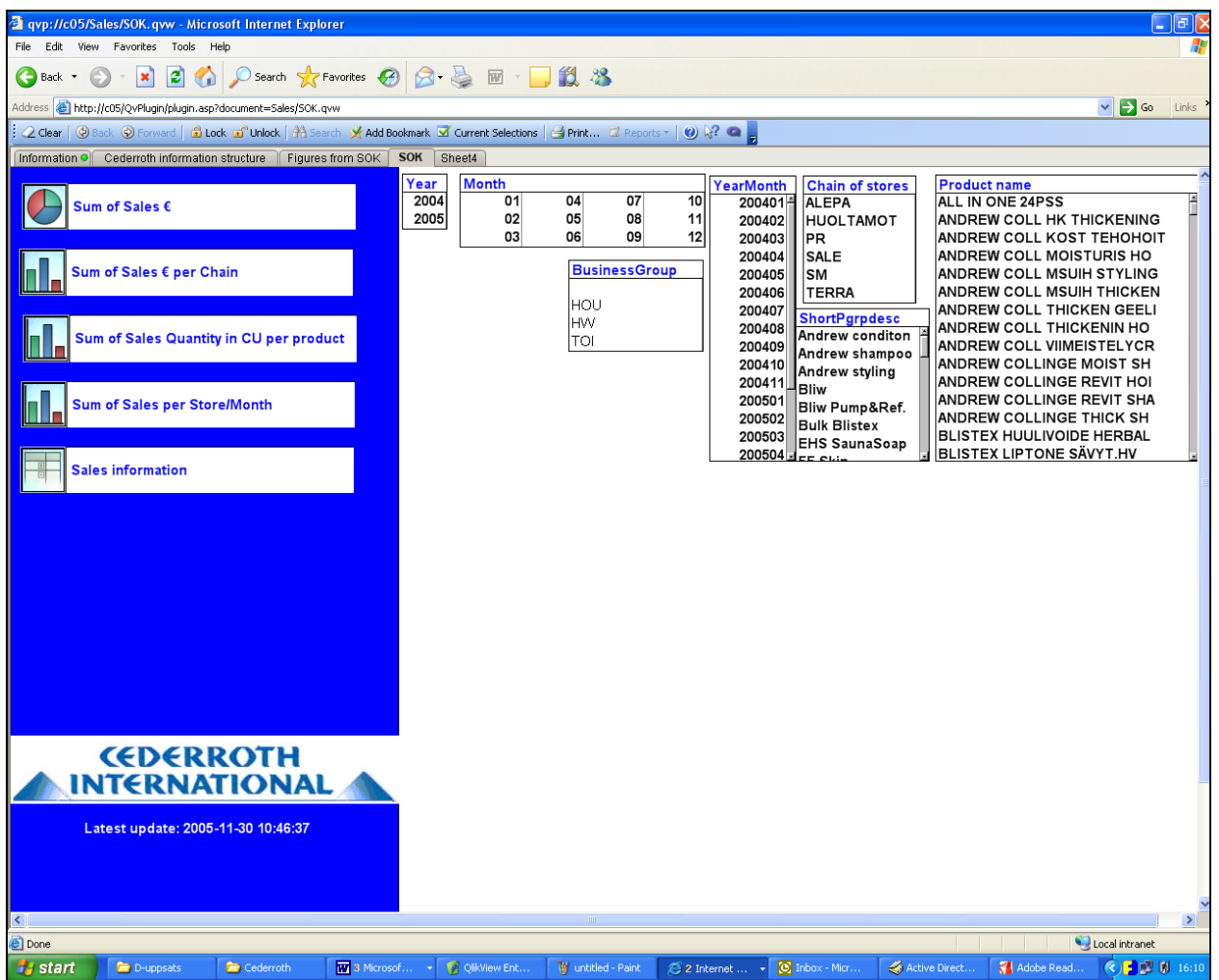
Säkerheten är tämligen viktig på ett så stort företag som Cederroth. Gemene man bör ej ha tillgång till all affärsdata. Visserligen så anses kanske inte den finska POS-datan som används i det här projektet vara säkerhetsklassificerad men den faller ändå under samma punkt som alla annan försäljningsdata som presenteras med hjälp av QlikView-applikationer.

I företaget indelas QlikView-användare i olika AD-grupper. De olika grupperna har olika tillgång till QlikView-vyerna. Finska användare som skall använda POS-data applikationen kommer att tillhöra AD-gruppen ”QlikView Sales Users” och därmed även få tillgång till övriga försäljningsvyer. Därför var det primärt att se till så att rätt användare får rätt behörighet. AD-gruppstildelningen gjordes i samråd med den finska försäljningsenheten och IT-chefen under projektet.

3.9 Distribution

Applikationen var nu färdigutvecklad och godkänd av superanvändaren och IT-avdelningen. Säkerheten var även modifierad så att behörighet hade givits till de anställda som kunde tänkas ha användning av applikationen.

Nu var det bara publicering som återstod förutom den slutgiltiga utvärderingen. Att distribuera en applikation i QlikView är enkelt så länge det finns tillgång till en QlikView-databas. Det enda som gjordes var att lägga upp filen i databasen under mappen Sales så var den distribuerad för användning. Nedan visas den startvy som slutanvändaren fick tillgång till när applikationen distribuerades.



Figur 3-31: Startvy för webbgränssnittet

3.10 Utvärdering

Utvärderingen av det finska projektet var tvungen att göras i Helsingfors där det finska huvudkontoret ligger. Eftersom det är enklare att hantera komplikationer på plats så beslöt det att en resa till Finland var lämplig.

På det finska kontoret gavs först en djupare presentation i QlikView och dess möjligheter när det gäller presentation av affärsdata. Sedan satte jag mig med superanvändaren för att utvärdera vad som kunde förändras. Själva innehållet var vi enade om sedan tidigare men presentationen var inte helt korrekt för alla slutanvändare. Anledningen till detta var att utvecklingen hade skett med en större bildskärm än vad gemene man har vilket hade lett till att några slutanvändare var tvungna att använda rullningslistan för att få tillgång till alla listboxar och diagram/tabeller. Detta förändrades enkelt genom justering i GUI.

Förutom detta så var det även i vissa fall onödigt många decimaler i grunddatan som presenterades. Detta ändrades via en ändring i skriptet som korrigerade den hämtade datans decimalantal. I övrigt var det inga invändningar vilket ledde till att projektet ansågs avslutat.

3.11 Sammanfattning

Att utveckla en applikation i QlikView är någorlunda enkelt om man ser till komplexiteten i skriptet och användargränssnittet. Men detta gör dock inte hela projektet enkelt eftersom superanvändaren och utvecklaren oftast talar olika språk. Det finska POS-data projektet flöt emellertid på ganska bra. Det var givetvis en del komplikationer på grund av språkkunskap och Excel-format för att inte tala om den komplexa Movex-databasen som användes för att hitta relaterad affärsdata. I slutänden så löstes dock alla problem och projektet blev godkänt av arbetsgivaren.

Vad som inte tagits upp i kapitlet är en del andra mindre projekt som även gjorts. Dessa projekt har utförts men inte alltid lett till en färdig applikation pga. prioritering inom företaget. Vissa mindre justeringar har även gjorts i befintliga QlikView-applikationer där slutanvändare velat ha nya vyer eller korrigerad presentation. För att begränsa längden på denna uppsats så ansågs dock detta överflödigt för presentation eftersom det hade resulterat i hundra sidor extra.

4 Analys av Business Intelligence system

I det här kapitlet ges först en introduktion för analys av BI-system i sektion 4.1. Efter det så beskrivs och analyseras de funktionella möjligheterna hos QlikView, SQL Server 2005 och BusinessObjects XI i sektion 4.2. Under sektion 4.3 beskrivs infrastrukturskrav som kan vara viktiga att ha i åtanke vid analys av BI-system. Sektion 4.4 beskriver några försäljarkriterier som även är viktiga vid analys. Till sist, i sektion 4.5, så ges en sammanfattning av analysen och kapitlet.

4.1 Introduktion

BI-teknologier tillåter organisationer att spåra, förstå och hantera viktig affärsinformation. BI börjar att göra anspråk på en ökande strategisk roll när företag försöker utnyttja data som är lagrad inom företaget. Ett typiskt BI projekt har en ”return of investment” (ROI)⁶ på i medeltal över 430% [7], men på grund av fragmenterad implementering i dessa projekt så är ofta företag oförmögna att få full nytta från global informationsanalys.

BI-standardisering försäkrar stark ROI genom att reducera BI-verktygsinköp, implementation och träningskostnader. Organisationers behov kan skilja sig avsevärt, men kriterierna för en BI-standard kan vanligtvis grupperas i tre huvudområden när det gäller krav [10]:

- **Funktionella möjligheter.** Förmågan för en produkt att uppfylla de identifierade kraven från BI-användare.
- **Infrastrukturskrav.** Hur väl som produkten möter organisationens infrastrukturskrav, dvs. huruvida produkten passar in med existerande arkitektur, skalbarhet och utökningsbarhet.
- **Försäljarkriterier.** Möjligheten för en vald försäljare att underhålla nuvarande och framtida projekt när det gäller stabilitet, resurser och kunskap.

I denna uppsats begränsas utvärdering till de funktionella kraven där de tre BI-systemen (se sektion 2.5-2.7) synas och jämföras jämt emot Cederroths behov. Resultatet av analysen kan ses som en grund när det gäller val av BI-system, men eftersom Cederroth är ett stort företag

⁶ Den avkastning som en investering ger.

så är det många aspekter som tyvärr inte kan tas med pga. tidsbegränsningar. Det kan dock vara av intresse för framtida investeringar om ett BI-system kan tänkas ha större funktionella möjligheter än ett annat. Några av de aspekter som inte tagits med kommer att beskrivas men inte analyseras, detta för att ge läsaren en överblick om vad som kan vara viktigt att ta hänsyn till när det gäller val av BI-system. Dessa aspekter kommer att beskrivas i sektion 4.2.8 – 4.4.5. De system som kommer att granskas i de tidigare sektionerna är QlikView, Microsoft SQL Server 2005 och BusinessObjects XI.

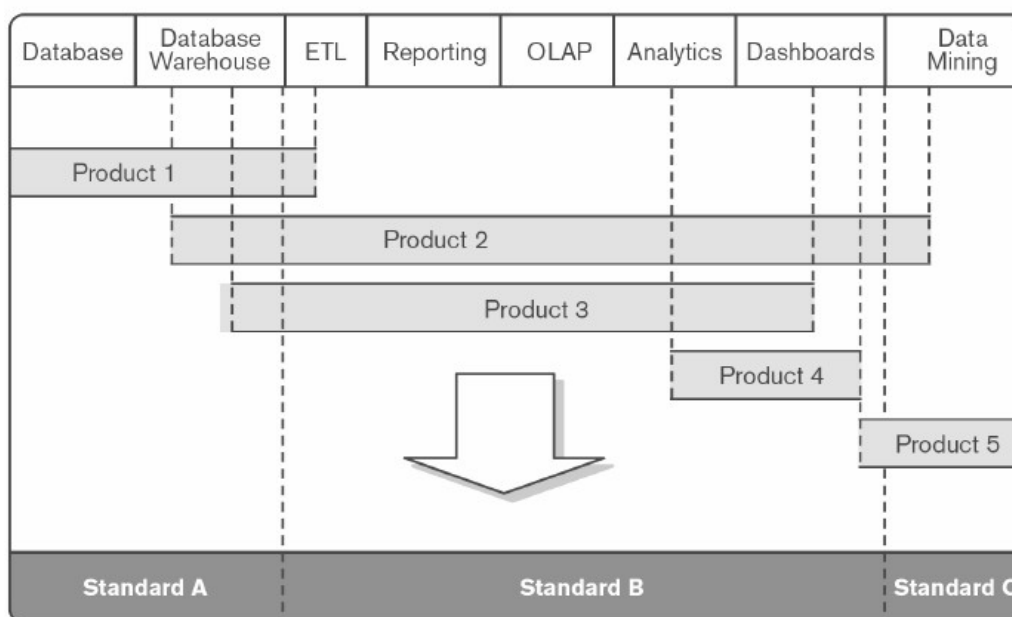
När produkter jämförs i olika aspekter bör det inte enbart tas hänsyn till total-”poängen” utan även hur ”poängen” är utspridd på de olika aspekterna. Det ideala BI-systemet är det som fullföljer användares krav i varje kategori. En produkt som presterar väldigt bra inom några få områden men inte uppfyller minimumkraven i andra bör inte ses som ett bra alternativ. För att i det här fallet kunna utröna vilken produkt som kan tänkas vara mest lämplig för Cederroths behov, så är det viktigt att avgöra vilket BI-system som täcker störst andel av behoven.

Under analysen har jag valt att utgå från de grundpelare som fastställts i ett white paper från Business Objects [10]. Detta skulle kunna ge fördelar åt BusinessObjects XI men det är bara grunden som tagits från detta white paper inte det som skrivits om mjukvaran. Fördelen med detta val är att Business Objects är ett företag som arbetat med BI i 15 år vilket jämfört med min kunskap på ett halvår bör ge ett bättre underlag för en konkret analys.

4.2 Funktionella möjligheter

Det kan vara svårt att hitta en BI-standard som täcker alla funktionella behov. Målet kan istället vara att försöka hitta en standard som tillhandahåller så många funktionella behov som möjligt. Skulle alla möjliga tekniska behov behövas så är det möjligt att använda sig av ett antal olika BI-standarder för att uppfylla detta. Genom att använda en uppsättning otvetydiga kriterier så är det möjligt att reducera den funktionella överlappningen i BI-produkterna. I *figur 4-1* så visas fem simulerade produkter med överlappande funktionalitet som kan bli konsoliderade till tre olika standarder. Dessa standarder är något som tagits fram av Business Object. Som synes så är standard A produkter som hanterar databaser och datalager, standard B produkter som tillhandahåller ETL, rapportering, OLAP, analys och instrumentbrädor och till sist standard C som är produkter vars inriktning är data mining⁷.

⁷ Se bilaga A (ordlistan) för begreppsförklaring.



Figur 4-1: Funktionella krav och standarder [10]

I denna uppsats gjordes den funktionella analysen utefter de nedan beskrivna funktionella aspekterna som även visas i figuren ovan.

4.2.1 Databas och datalager

Under denna rubrik så finns de produkter som tillhandahåller funktionaliteten att lagra information på ett effektivt sätt för slutanvändaråtkomst. Även om databaser och datalager ibland ses som separata funktionaliteter som inte inkluderas i BI-systemet så är dessa grunden till hela BI-systemet.

QlikView tillhandahåller inte databaser och datalager i själva mjukvaran. Däremot så reducerar AQL (se sektion 2.5.3) behovet av datalager eftersom det istället använder datastrukturen datamoln där dataassociationer lagras. QlikView har som tidigare nämnts inga direkta databaser men när data laddas upp ur datakällor, som mycket väl kan vara databaser, så skapas en associativ databas (se sektion 2.5.1) för att minska minnesbehovet och svarstiderna.

SQL Server 2005 är i grunden en databasplattform vilket således indikerar att den inkluderar databaser i BI-systemet som är integrerat i mjukvaran. Databasmotorn i SQL Server 2005 tillhandahåller säker och pålitlig lagring av såväl relationsdata som strukturerad data.

Datalager existerar också i SQL Server 2005 och styrs med hjälp av den inbyggda relationsdatabasmotorn [27].

BusinessObjects XI är likt QlikView när det gäller databaser eftersom kraft har lagts på hur data ska plockas ur databaser snarare än att tillhandahålla dem själv. Datalagerslösningar har dock getts förpacketerade lösningar för företagsapplikationer såsom SAP, PeopleSoft, Siebel och Oracle. Med det inbyggda hjälpmedlet för dataintegration finns även möjlighet att skapa egna datalager.

Som synes har de tre BI-systemen olika sätt att angripa datalager- och databaslösningar. Den enda mjukvaran som tillhandahåller databaser är SQL Server 2005, men frågan är om detta är ett måste för att BI-systemet skall ses som komplett. Ett BI-system används huvudsakligen till att plocka data från datakällor och presentera den på ett konstruktivt sätt. Det viktigaste bör således vara att ha en kompatibilitet mot databaser för hämtning av data och inte själva egenskapen att tillhandahålla egna databaser. För Cederroths räkning, som använder sig av ett stort antal olika databaser för sin affärsdata, så är det viktigaste att data kan hämtas på ett säkert och effektivt sätt. Det vill säga, Cederroth är inte beroende av att BI-systemet tillhandahåller egna databaser.

Datalager kan vara lämpligt att ha för att få snabb tillgång till affärsdata från till exempel ERP-system såsom Movex (se sektion 2.2.2). Både SQL Server 2005 och BusinessObjects XI tillhandahåller datalagersfunktionallitet, vilket även kan ses som ett krav från Cederroth. Dock så har QlikView med hjälp av AQL (se sektion 2.5.3) reducerat behovet av datalager på ett till synes effektivare sätt.

4.2.2 Dataintegration

De produkter som tillhandahåller dataintegration är även kallade ETL-mjukvaror. De är designade för att kombinera information från multipla källor för att mata databaser och datalager.

QlikView hanterar ETL med hjälp av det inbyggda skriptet som används vid applikationsskapandet. Data från multipla källor kombineras och associeras direkt i skriptet med hjälp av enkla implementeringar. Här kan exempelvis affärsdata från en SQL-databas kombineras med affärsdata från Excel-dokument. I vanliga fall så laddas data till ett datalager,

men när datan har extraherats och transformerats i QlikView så laddas den upp till datamolnet med hjälp av AQL-teknologin (se sektion 2.5.3).

Dataintegration i SQL Server 2005 görs med hjälp av den inbyggda "Data Transformation Services"-teknologin [20]. DTS är egentligen en uppsättning verktyg som används för att importera, exportera och transformera data mellan en eller flera datakällor, såsom SQL Server, Excel, Access etc. Således är funktionaliteten i DTS den samma som i ETL.

En DTS-lösning kan sparas i ett eller flera paket. Varje paket kan innehålla en organiserad uppsättning av uppgifter som definierar det arbete som skall utföras, transformationer på data och objekt, arbetsflödesrestriktioner som definierar uppgifter som skall exekveras och uppkopplingar till datakällor och destinationer.

BusinessObjects XI använder sig av BusinessObjects Data Integrator XI för ETL-funktionalitet. Data integrator extraherar data från källsystemen och transformerar den till BI-anpassat format. När transformationer är färdiga så laddas data till måldatabaser eller applikationer.

ETL-teknologin är ett måste för att ett BI-system skall fungera i realiteten. Alla de tre systemen tillhandahåller teknologin som dock kan skilja något i benämning i de olika systemen. Grunden är likväl den samma så slutsatsen kan dras att både QlikView, SQL Server 2005 och BusinessObjects XI har en ETL-funktionalitet som är anpassad efter det egna systemet men ändå jämförelsevis likvärdig.

4.2.3 Rapportering

Produkter som levererar rapportering körs ofta dagligen för hämtning och visning av affärsdata. Oftast är dessa rapporter inbäddade i applikationer.

QlikViews rapporteringsvyer sätts upp av ansvarig vy-designer. De resultat som fås ur en QlikView-applikation går direkt att skriva ut som en rapport, sparas i PDF-format, exporteras till Excel eller kopieras till andra Office-applikationer. Rapporter som ska uppdateras dagligen kräver en batch-fil som exekverar dem direkt på QlikView-servern för att få tillgång till den senast inkomna datan till rapporterna.

SQL Server 2005 använder sig av den inbyggda applikationen Reporting Services [25] för rapportskapande och rapportering. Reporting Services är en omfattande plattform för skapande, hantering och leverans av traditionella pappersrapporter men även interaktiva webbaserade rapporter. Genom att skapa ett Reporting Services-projekt i Business Intelligence Development Studio (se sektion 2.6.1) så blir det möjligt att spara rapporten på en rapportserver. Lagring av rapporter på en rapportserver gör det enkelt att uppdatera datainnehållet när så krävs för att alltid kunna delge klienten korrekt data.

Crystal Reports XI [8] är rapporteringsverktyget som BusinessObjects XI [9] använder sig av vid rapportskapandet. Crystal Reports gör det möjligt att skapa flexibla och innehållsrika rapporter för integrering med webb- och Windows- applikationer. Crystal Reports-rapporter lagras på en Crystal Reports-server som kör dynamisk uppdatering av rapporterna, dvs. rapporterna får sin affärsdata uppdaterad när förändringar har skett.

Rapporter är det som slutanvändaren ser av en BI-applikation, därför är det viktigt att rapporteringsverktyget i BI-systemet ger en enkel och lättförstålig vy av affärsdatan. QlikViews rapporter ger en tämligen enkel och lättförstålig bild av affärsdatan. Dock så är inte detta rapportsystem totalt integrerat med exempelvis Word och andra produkter för pappersbaserade rapporter. Detta gör att användaren är tvungen att klippa och klistra för att göra en snygg rapport för utskrift. Onlinerapporteringen i QlikView fungerar dock utmärkt för användning i ett företag som Cederroth. Reporting Services, som SQL Server 2005 använder sig av för rapporter, är ett mer sofistikerat verktyg när det gäller pappersrapporter än QlikView men onlinerapporteringen ger ett likvärdigt resultat. Även Crystal Reports, som är integrerat med BusinessObjects XI, ger bra rapporter i pappersformat men likvärdiga rapporter för onlinearbete.

4.2.4 OLAP

Tillhandhållandet av OLAP⁸ möjliggör åtkomst till atomär och interaktiv information. Användare är inte begränsade till den delmängd av information som är tillgänglig i en specifik rapport eller kub utan kan fråga nya frågor och analysera all information som är tillgänglig i databasen eller datalagret.

Till skillnad från traditionella kubbaserade OLAP-verktyg så sparas QlikView-förfrågningar och förenade tabeller i minnet [32]. Beräkningar sker i realtid och täcker ett obegränsat antal dimensioner. Detta tillåter användaren att nedstiga från valfri dimension för att nå en specifik transaktionsnivås detaljer. QlikView använder sig av AQL (se sektion 2.5.3) för att skapa denna datamolnsfunktionallitet. Med AQL så sparas data bara en gång vilket gör att en datamolnsdatabas är mer effektiv än en ekvivalent OLAP-databas.

OLAP hanteras i SQL Server 2005 med hjälp av den inbyggda Analysis Services-delen [34]. För att kunna använda sig av OLAP i SQL Server 2005 måste först dimensioner och mått bestämmas. Efter det så används DTS för att extrahera, transformera och ladda data från datakällan för att sedan laddas till kuberna. Analysis Services används därefter för att initiera mått, dimensioner och scheman till OLAP-kuberna som skapats. Slutligen ges användaren tillgång till kuberna, som sedermera kommer att hantera de förfrågningar som användaren har.

BusinessObjects XI:s förfrågnings- och analysverktyg tillåter användaren att identifiera trender och hitta grundorsaker i data från relations- och OLAP-datakällor. BusinessObjects OLAP Intelligence™ XI är det specifika förfrågnings- och analysverktyg som BusinessObjects XI tillhandahåller för OLAP-datakällor. Superanvändare kan med detta verktyg göra snabba ad hoc OLAP-analyser, vilket tillåter dem att skära, tärna, borra, rangordna och filtrera data. Allt detta för att kunna få tag i den grunddata som finns tillgänglig genom OLAP.

OLAP används för att minska svarstiden vid förfrågningar i BI-system. Att få snabba svar på förfrågningar är ett viktigt krav från Cederroths användare, därför ses OLAP som viktigt vid analys av BI-system. Både SQL Server 2005 och BusinessObjects XI använder sig av OLAP på ett snarlikt sätt. QlikView lagrar dock datatabellerna i minnet (datamoln) vilket är ett effektivare sätt än tillvägagångssättet i OLAP. Därför borde förfrågningar i QlikView ge snabbare svar på förfrågningar jämfört med förfrågningar i de andra två BI-systemen. För att detta skall kunna säkerställas så bör ekvivalenta funktioner skapas i de tre BI-systemen där sedan svarstiden klockas.

⁸ Se bilaga A (ordlistan) för begreppsförklaring

4.2.5 Analys

Analysapplikationer finns i de produkter som är designade för informationsanalys inom ett specifikt område såsom till exempel försäljning, lagersaldo etc.

Färdiga analysapplikationer finns inte direkt tillgängliga i QlikView. QlikTech (se sektion 2.5) har valt att "outsourca" modulapplikationer till licensierade QlikView-konsulter och företag. Det finns således färdiga applikationer som med små ändringar/anpassningar kommer att fungera inom de områden där behov finns, exempelvis lagersaldo och försäljning. Dessa färdiga applikationer kräver givetvis att företaget inhandlar dessa från berörd konsult/företag.

I SQL Server 2005 finns inga färdiga analysapplikationer dock så medföljer ett antal inbyggda "BI-wizards" som hjälper till vid skapandet av applikationer. Dessa underlättar skapandet av både standardapplikationer och mer avancerade applikationer.

Det finns även möjlighet att utgå från en färdig mall för att definiera och generera en applikation. I applikationen inkluderas då relationsdatabaser, DTS paket och Analysis Services OLAP databaser om så väljs. Dessa komponenter kommer då att arbeta friktionsfritt tillsammans i den slutgiltiga applikationen.

BusinessObjects XI tillhandahåller förpacketerade mått, rapporter och analyser som täcker funktionella områden som kunder, produkter och tjänster, återförsäljningskedjor, personalresurser och finansiering. Dessa medföljande applikationer inkluderar domänfokuserade affärsmått, instrumentbrädor och styrkort samt analysteknik för insyn i affärsverksamheten. Applikationerna stöds även av ett ramverk som gör det möjligt att förändra och anpassa beroende på företagets behov.

Att delge färdiga analysapplikationer i BI-systemet kan underlätta för företaget i den inledanden fasen av BI-användningen. De färdiga applikationer som tillhandahålls av BI-mjukvaran är dock oftast generella, vilket gör att dessa likväl måste anpassas till företagets egna affärsprocesser. BusinessObjects XI är egentligen den enda av de tre BI-systemen som tillhandahåller färdiga applikationer. Dessa applikationer måste som ovan nämnt givetvis anpassas utefter företagets behov vilket i viss mån kan tänkas göra det mer komplext än att skapa en applikation från grunden. Med QlikView så finns det möjlighet att anlita konsulter/företag för att få färdiga applikationer vilket då givetvis leder till en ny utgiftspost för företaget. SQL Server 2005 tillhandahåller mallar och "wizards" istället för färdiga

applikationer, vilket underlättar vid applikationsskapandet om grundkunskapen finns. Det är i slutänden helt upp till applikationsskaparen att avgöra vad som kan tänkas var enklast för företaget i fråga.

4.2.6 Instrumentbrädor

Produkter med instrumentbrädor samlar nyckelprestanda för företagsindikationer och presenterar dem i ett gränssnitt. Vanligtvis används sådana av chefsanvändare och de kan också stödja målsättning, kollaboration och affärsmetoder som balanserade styrkort.

Med QlikView är det möjligt att visa information genom en interaktiv, intuitiv och grafisk miljö. Denna visning (instrumentbräda) kan innehålla information från valfri datakälla så som försäljning, CRM eller finansiella applikationer. Instrumentbrädan tillhandahåller en integrerad vy av hanterbar information från multipla affärsavdelningar. I och med detta så kan chefer inom en koncern få god kontroll på verksamheten.

SQL Server 2005:s "Key Performance Indicator"-ramverk, som är inbyggt i Analysis Services, tillhandahåller en serverdefinierad mekanism för definition av företagsmått. Ett KPI innehåller uttryck för värde, mål, nuvarande status och trender inom företagverksamheten. Dessa visas i rapporter, portaler och instrumentbrädor med hjälp av enkel grafik som till exempel manometrar och stoppljus. Tack vare instrumentbrädor baserade på KPI så kan företagsledningen utföra behovsanpassade handlingar.

Instrumentbrädor för prestandahantering (performance management) och styrkort i BusinessObjects XI gör det möjligt för individer på alla nivåer i en organisation att göra korrekta handlingar snabbare och samtidigt känna att besluten baseras på trovärdig information. BusinessObjects Dashboard Manager XI, som används för skapandet av instrumentbrädor, levererar mått-, alarmering- och styrningskompetens som hjälper organisationer att övervaka och förstå deras mest kritiska aktiviteter. Analytiskt innehåll kan visas på ett personligt och säkert sätt, vilket tillåter användare att spåra företagsmått över en tidsperiod och identifiera trender, avvikelser och undantag.

Instrumentbrädor är viktiga verktyg för att ge en övergripande insyn i företagets affärsdata för kritiska beslut. Alla tre BI-system tillhandahåller den här funktionaliteten på ett grafiskt och

övergripande sätt. Det är svårt att avgöra vilken av dem som har de bästa instrumentbrädemöjligheterna då alla är väl anpassade utefter det system de arbetar i. Det kan dock sägas att alla systemen ger den funktionallitet som krävs för att skapa bra instrumentbrädor åt företagsledning och vanliga anställda.

4.2.7 Data mining

Data mining-produkter används för sofistikerad automatisk analys av stora mängder data med mängder av olika variabler. Avstämning och tolkning av dessa resultat kräver specialiserad expertis inom statistisk modulering.

Data mining används huvudsakligen till att hämta relaterad data till datalager och OLAP. I och med att QlikView inte använder sig av datalager eller OLAP så tillhandahålls inte data mining-funktionallitet i QlikView.

Det finns ett antal användbara verktyg för data mining i SQL Server 2005 [23]. Business Intelligence Development Studio integrerad i Visual Studio ser till så att data mining adderas till verktygsfåran. Ett data mining-projekt kan existera bredvid andra utvecklingsprojekt i BI-lösningen. Till exempel så är det möjligt att göra följande utan att lämna designgränssnittet: designa modeller för att förutse produktförsäljning i Analysis Services-projekt (se sektion 2.6.1), periodvis omskolning av DTS-projektmodeller (se sektion 4.2.2) och bygga en webbsida som tar fördel av den förutseende kraften i data mining i ett webbprojekt. Sedan är det möjligt att paketera dessa projekt till en portabel lösning. Data mining objekt i ett Analysis Services projekt är filer som alla de övriga projektfilerna, vilket medför att det är möjligt för dessa att delas ut, undersökas eller flyttas till källkontrollprogram för säkerhetsförvaring.

Business Objects arbetar med KXEN, THINKANALYTICS och IBM för att delge kunder en mängd nya valmöjligheter för att tillgodose förutsägelseanalys till analysapplikationer inom ett företag. Dessa applikationer ger användare bättre insyn i data och hjälper dem att förutse resultat i affärsbeslut.

Business Objects har bäddat in förutsägelseanalysteknologi från KXEN i dess applikationer. KXEN tillhandahåller förutsägelse och beskrivande moduleringsteknologi som tillåter företag att dra kritisk information från företagsdatakällor och sedan använda denna information vid dagliga beslut.

Business Objects har även gått i partnerskap med THINKANALYTICS, ett ledande data mining-plattform- och affärsanalytiskföretag, för att tillhandahålla öppen integration med ett statistik och data mining verktyg.

Business Objects arbetar även med IBM för att testa integration med externa data mining-verktyg. IBM levererar data mining- och styrverktyg för att möjliggöra att företag kan få den insyn som krävs.

Huruvida data mining är ett måste för att BI-systemet skall fungera är upp till själva mjukvaran. I och med att algoritmerna i ett data mining-verktyg relaterar data till annan data, så förenklar det för applikationsskaparen eftersom han/hon slipper göra det manuellt. QlikView som inte tillhandahåller detta löser oftast dessa data relationer med sin associativa databas (se sektion 2.5.1). Finns det ingen lösning där får dock applikationsskaparen gräva i datakällan själv för att finna mönster som kan sammankoppla data från olika källor. SQL Server 2005 tillhandahåller data mining i sitt BI-verktyg vilket gör att mönsterrekognosering sker med automatik. Business Objects har istället för att tillhandahålla det själv skapat partnerskap med företag som är specialiserade på data mining. Detta gör att dessa företags data mining-applikationer är integrerade med BusinessObjects XI. Om data mining är ett måste eller inte i Cederroths fall är upp till hur dess affärsdata är lagrad. Är datan lagrad på ett bra sätt så skall det vara tämligen enkelt att hitta mönster som kan associera data till varandra vilket gör att data mining kan vara överflödigt.

4.2.8 Användarprofiler

Utöver de funktionella aspekter som tagits fram i sektion 4.2.1 - 4.2.7 är det även viktigt att utvärdera de olika verktygen emot specifika funktionella krav som slutanvändaren kan tänkas ha. Eftersom det inte finns någon typisk användare när det gäller BI så kan det vara bra att gruppera eller segmentera användarna i olika profiler [10]. Exempel på lämpliga profiler kan vara chefer, controllers, allmänna affärsanvändare, externa användare såsom kunder och partners etc. Följande parametrar kan även vara lämpliga att ta hänsyn till när det handlar om användarbehov:

Funktionell komplexitet

Somliga användare kan vara i behov av produkter med mer sofistikerad och kraftfull funktionalitet än andra. Detta kan göra att valet av produkt påverkas.

Datadjup

Vissa användare behöver få tillgång till mer detaljerad information än andra. Till exempel så kan ledningen vara i behov av högt aggregerad information medan en försäljare kan vara i behov av detaljerad orderinformation. Eftersom olika typer av information lagras i olika system och format så kan detta mycket väl påverka valet av produkt.

Databredd

Användare kan behöva jämföra information från olika system. Det är osannolikt att all information som skall användas lagras i ett och samma datalager. Möjligheten för ett BI-verktyg att göra en sammansatt analys av data från olika system är således en viktig faktor.

Användarkontroll

Medan några användare behöver tillgång till information direkt och automatiskt så finns det andra som behöver förkonfigurerat innehåll. Ett BI-system bör stödja båda dessa användartyper.

Användarvänlighet

Det är viktigt att ett BI-verktyg har en mjuk övergång från grundläggande till mer sofistikerade gränssnitt.

Anpassningsbarhet

Vissa användare vill ha mer anpassade gränssnitt än andra. En chef kan vilja ha instrumentbrädor, en försäljare kan vara behov av kunders inköp osv. BI-system bör därför kunna tillhandahålla ett standardmässigt gränssnitt samt ett antal anpassade om så behövs.

Affärsspecialisering

Funktionalitet som är specifikt designad för detaljerade industrier och affärsdomäner kan vara av nytta för somliga användare. Exempelvis så kan en BI-standard ha applikationer som är specifikt designade för personalresurschefer eller finanspersonal.

4.3 Infrastrukturskrav

BI-standarder bör möta de infrastrukturella krav som en organisation kan tänkas ha när det gäller arkitektur och underhållsarbetsbörda.

4.3.1 Arkitekturstandarder

Om ett företag ska välja en BI-standard så är det ofrånkomligt att denna standard skall var anpassbar jämt emot existerande infrastruktur och arkitekturstandarder. Följande arkitekturiska standarder kan vara lämpligt att ta hänsyn till:

Plattformer

De flesta organisationer har plattformstandarder men det är sällsynt att det bara finns ett operativsystem inom företaget. En BI-standard bör därför stödja multipla plattformer såsom Windows, UNIX, .NET, Java etc.

Applikationer

En organisation kan ha en mängd olika applikationer allt från operativsystem till analytiska system. BI-systemet måste ha åtkomst till information direkt ifrån dessa system. Detta kan göra det tvunget för BI-systemet att ha speciella gränssnitt för ERP-system såsom SAP, Oracle, Movex, PeopleSoft osv.

Databaser och datalager

De flesta stora organisationer har en mix av relations- och OLAP-datakällor som är inköpta från olika återförsäljare. BI-standarden måste kunna plocka data ur dessa datakällor på ett effektivt och flexibelt sätt.

Globalisering

Internationella organisationer kan vara i behov av ett BI-system som stödjer ett antal olika språk.

Skalbarhet

BI-systemet skall kunna tillhandahålla pålitlig dataåtkomst till ett stort antal användare utan att ha utökade krav på hårdvaruresurserna.

4.3.2 Ledningsverktyg

Prestanda och säkerhet för systemet skall kunna säkerställas med minimal underhållskostnad. Det utvalda BI-systemet bör stödja rollbaserad- eller nivåhantering av olika aspekter för informationssystemet. Dessa aspekter kan vara lämpligt att ta hänsyn till:

Säkerhet

Flexibel och finkornig säkerhet hjälper för att undvika separata rapporter för olika användare och försäkrar samtidigt att resultat som fås ur en BI-applikation returneras beroende på användarprofilen.

Slutanvändarautonomi

Om en användare behöver tillgång till all data i databasen under användarens säkerhetsnivå så bör BI-systemet tillhandahålla detta.

Dataintegration

Det är lämpligt om BI-systemet kan integrera information från många olika källor med en enkel administration för uppdatering.

Revision och spårning

Att ha möjlighet till spårning av BI-användandet är viktigt för att kunna finjustera, förbättra och utföra effektivt underhåll på informationssystemet. BI-systemet bör kunna anpassa sig för detta utan att det stryker systemets resurser.

4.4 Försäljarkriterier

Produkt- och arkitekturiska kriterier hjälper ofta till att minska antalet accepterbara produktval. Men det slutgiltiga valet baseras oftast på finansiella termer och affärsrelationer med återförsäljare. De fall där det handlar om stora återförsäljare med ett flertal produktlinjer så bör bland annat följande tas i åtanke när det handlar om relationen till återförsäljarens BI-försäljning:

4.4.1 Finansiell stabilitet

Möjligheten för en återförsäljare att rida ut en storm av turbulenta tider är viktigt när det görs ett strategiskt val av återförsäljare. När det gäller globala standarder kan det vara bra att välja marknadsledaren eftersom de bör ha en stabil grund att stå på.

4.4.2 Industrianalytisk utvärdering

Industrianalytiker såsom Gartner, Meta Group och IDC utvärderar regelbundet olika återförsäljare på marknaden, vilket kan hjälpa en organisation när val av BI-återförsäljare görs.

4.4.3 Världsomspännande service och support

Lyckad utveckling av global BI-standard i internationella företag kräver inledningsvis service och träning från återförsäljaren (åtminstone i inledningsfasen). Återkommande understöd vid problemuppkomst kan vara nog så viktigt för att ett projekt skall lyckas. Om en internationell organisation får tillgång till lokalt stöd så kan det mycket väl vara en nyckelfaktor vid val av BI-standard.

4.4.4 Återförsäljares ”ekosystem”

När en global BI-standard skall sättas i bruk så bör företaget inte enbart se till service som kan fås direkt från återförsäljaren. Det kan vara nog så användbart med återförsäljarens nätverk av konsulter och träningspartners. Tillgängligheten av anställda med kunskaper inom BI-teknologin och antalet andra företag som använder sig av produkten kan vara till god hjälp. Desto mer utbrett ”ekosystemet” är, desto större sannolikhet har en organisation att finna de extra projektresurserna när de behövs.

4.4.5 Relation till andra standarder

Ett företag bör välja en BI-standard från en återförsäljare som har partnerskap/samarbete med företagets existerande strategiska applikationer såsom Microsoft, SAP, Oracle, Movex, IBM, PeopleSoft etc.

4.5 Sammanfattning

Den funktionella analys som gjorts i detta kapitel är som synes inte tillräcklig som beslutsunderlag för val av BI-system. Inte för att den inte är korrekt utan för att det är så många andra omständigheter som måste tas i beaktande när ett beslutsunderlag skapas.

Slutsatsen att SQL-server 2005 har det bredaste utbudet av inbyggda funktionella verktyg kan dock tas, men det behöver inte betyda att detta verktyg är det bästa för BI. I övrigt så tillhandahåller alla tre verktyg den funktionalitet som krävs på ett eller annat sätt. För att skapa ett korrekt beslutsunderlag så måste även infrastrukturskrav och försäljarkriterier analyseras. Eftersom Cederroth redan använder sig av QlikView för BI-applikationer så betyder det en merkostnad vid byte av BI-system. Om hänsyn bara tas till de funktionella aspekter som jag analyserat i denna uppsats så finns det absolut ingen anledning att byta då det inte finns några större skillnader mellan QlikView och de två andra systemen.

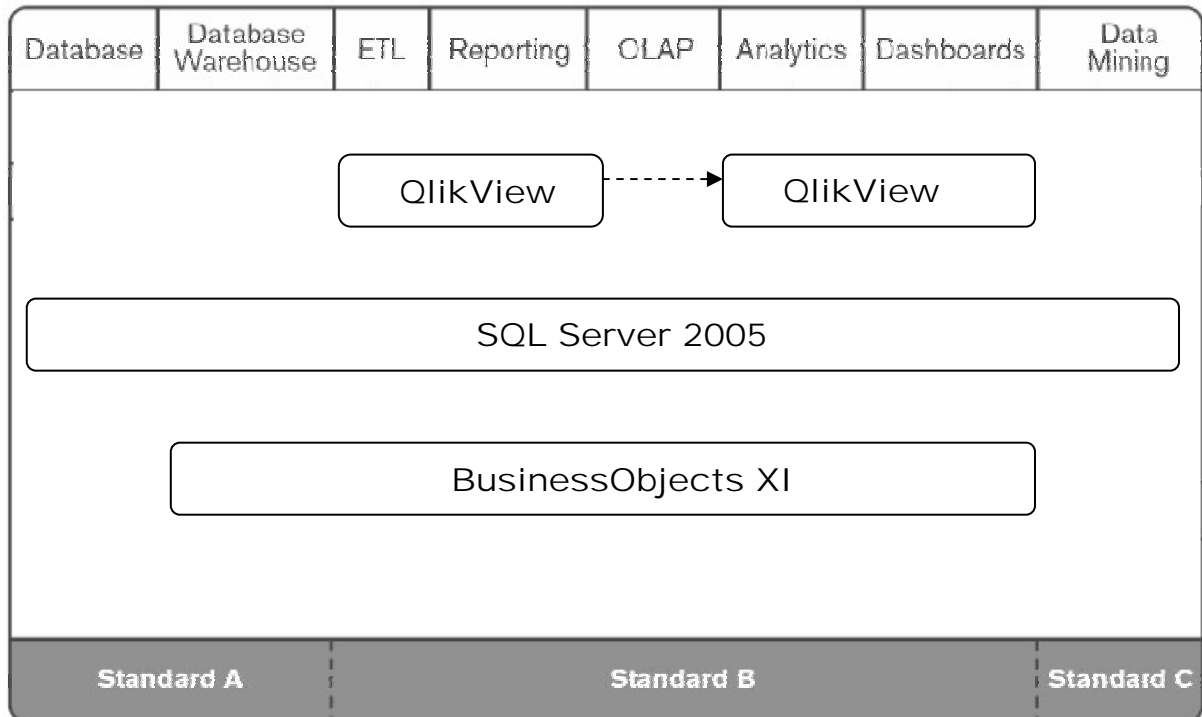
5 Resultat & slutsats

De krav som arbetsgivaren hade på mig var att skapa BI-applikationer i QlikView och även utvärdera tre BI-system. Det var en BI-applikation som involverade finsk POS-data som hade högsta prioritet, vilket även resulterade i att den slutfördes och dokumenterades i denna uppsats. Resultatet av den applikationen och analysen av QlikView, SQL Server 2005 och BusinessObjects kommer att beskrivas i sektion 5.1 i detta kapitel. Sektion 5.2 är helt tillägnad mina egna slutsatser och funderingar kring BI och dess användning.

5.1 Resultat

Det finska POS-dataprojektet började från grunden, dvs. den enda information som fanns var de Excel-datablad som jag fått tillgång till. Först och främst gällde det då att hitta de associerade datafälten i Movex-systemet. När det var gjort så var det dags att skapa en QlikView-applikation som kunde hantera och synkronisera affärsdatan som hämtats. Till den applikationen så skapades även ett användargränssnitt som uppfyllde slutanvändarens och IT-enhetens krav. Hela projektet avslutades i Finland på Cederroths finska huvudkontor där slutanvändaren fick ge sina synpunkter till den färdiga applikationen. När dessa synpunkter hanterats så publicerades applikationen på webbgränssnittet så att den kunde användas i dagligt bruk. Förhoppningsvis så används den numera för att underlätta det dagliga affärsbeslutstagandet.

Analysen som gjordes av de tre BI-systemen QlikView, SQL Server 2005 och BusinessObjects XI begränsades till funktionella aspekter eftersom uppsatsen är tidsbegränsad och en fullständig analys minst hade krävt ett halvår till. *Figur 5-1* illustrerar de funktionella aspekter som tagits hänsyn till och huruvida de tre systemen har funktionaliteten eller inte:



Figur 5-1: Resultat av analys (missvisande)

Figuren ger dock missvisande svar eftersom QlikView har reducerat bort behovet av datalager, OLAP och data mining med komplementär teknologi. Således är det bara databaser som QlikView inte tillhandahåller. BusinessObjects XI har inte databaser men data mining-funktionaliteten har de löst genom partnerskap med företag som tillhandahåller data mining-expertis. Så det enda som skiljer SQL Server 2005 från de andra är möjligheten att skapa egna databaser. Om man bara ska utgå från detta, vilket man ej bör göra eftersom analysen är för tunn, så måste man avgöra om databasmöjligheten är ett måste för att BI-systemet skall vara komplett. Eftersom det krävs en mycket djupare analys för att få ett korrekt resultat så har ingen utvärdering gjorts med hänsyn till databasfunktionalitetens vara eller inte vara. Skulle en djupare analys göras så bör även funktionaliteterna jämföras mellan BI-systemen, slutanvändarbehov tas i beräkning, infrastrukturella krav jämföras i förhållande till systemet, försäljarkriterier analyseras och sist men inte minst kostnaden för inköp och upprätthållande av systemet utvärderas.

5.2 Slutsats

Många företag anser och andra borde inse att desto snabbare de kan se hur vägen framför dem svänger, desto snabbare kan de styra i den rätta riktningen. BI är ett verktyg för att kunna se

hur vägen kommer att svänga. I dagens företagsklimat så är BI ett bra verktyg för att kunna underlätta det dagliga beslutstagandet. Mitt arbete med BI i denna uppsats har givit mig en djup inblick i hur BI-konceptet fungerar i realiteten. Att fått jobba med det i ett så pass stort företag som Cederroth har varit ett givande tillfälle att lära och se hur företag kan utnyttja BI i sitt dagliga arbete. BI är ett relativt nytt redskap för Cederroth men ju snabbare företagets anställda inser möjligheterna med det desto snabbare kommer företaget att förbättra sina affärsbeslut.

Det finns många olika BI-system att välja på. I denna uppsats har jag bara tagit upp tre av dem men det finns en djungel där ute att välja och vraka i. De tre jag analyserade var relativt lika i funktionalitetssynpunkt men sedan är det alltid upp till företaget att avgöra vad som är bäst för just dem. QlikView som är det BI-system som Cederroth använder och även det verktyg jag använt för applikationsskapande är relativt enkelt och brett som BI-system. Om Cederroth skulle byta BI-system från ett som redan är i bruk och fungerar så skulle det förmodligen krävas något exceptionellt. Det är mycket möjligt att både SQL Server 2005 och BusinessObjects XI skulle visa sig vara bättre vid en djupare analys men kostnaden att byta från ett system som fungerar till ett som förmodligen kommer att fungera är hög och dessutom är det tidskrävande.

Referenser

- [1] Jaejin Ahn, Nathalie Boese, Per Ehnsjö, Eva Flinck, Amy Herbert, Mats Hermansson, Chris Jones, Jesper Linde, Anders Perman. *Intentia Movex ThisGen Implementation for AS/400*. IBM Redbook, first edition, IBM Corporation, 1999.
- [2] Brian Arkills. *LDAP directories explained : an introduction and analysis*. Addison-Wesley, 2003.
- [3] Kent Beck. *Extreme Programming Explained*. Addison-Wesley, 1st edition, 1999.
- [4] T. Berners-Lee, J. Gettys, R. Fielding, H. Frystyk, P. Leach, L. Masinter, J. Mogul. RFC: 2616 *Hypertext Transfer Protocol – HTTP/1.1*. Juni 1999
- [5] Abbay Bhushan, Bob Braden, Will Crowther, Eric Narslem, John Heafner, Alex McKenzie, John Melvin, Bob Sundberg, Dick Watson, Jim White. RFC: 265 *The File Transfer Protocol*. 17 November 1971
- [6] Chi Kin Chan and H.W.J. Lee. *Successful Strategies in supply chain management*. Idea Group Pub, 2005.
- [7] Cognos Corporation. *The Financial Impact of Business Analytics*. December 2002.
- [8] Jaylene Crick. *Crystal Reports Server XI Functional Overview*. Business Objects, September 2005.
- [9] Darren Cunningham. *BusinessObjects XI Redefining How Information Empowers Users*. Business Objects, September 2005.
- [10] Timo Elliott. *Choosing a Business Intelligence Standard*. Business Objects, 2004.
- [11] Ramez Elmasri, Shamkant B. Navathe. *Fundamentals Of Database Systems*. Forth edition, Pearson Education Inc, 2004.
- [12] Fiona Fui – Hoon Nah. *Enterprise resource planning*. IRM Press, 2002.
- [13] Gartner, website, Oktober 2005. <http://www.gartner.com/>.
- [14] Samuel B. Griffith. *Sun Tzu on the Art of War*. Oxford University Press, 1963 .
- [15] T. Howes, S. Kille, M. Wahl. RFC: 2251 *Lightweight Directory Access Protocol (v3)*. December 1997.
- [16] James Irvine, David Harle. *Data Communications and Networks*. John Wiley & sons LTD, 2000.
- [17] Ken Killebrew. *QlikView™7*. White Paper, QlikTech Inc, 21 April 2005.
- [18] Ralph Kimball, Joe Caserta. *The data warehouse ETL toolkit: practical techniques for extracting, cleaning, forming and delivering data*. Chichester : Wiley, 2004.
- [19] Alexander H. Kracklaner, D.Quinn Mills, Dirk Seifert. *Collaborative customer relationship management : taking CRM to the next level*. Springer-Verlag, 2004.
- [20] Diane Larsen. *Data Transformation Services (DTS) in Microsoft SQL Server 2000*. Microsoft Corporation, September 2000.
- [21] Gustaf Liljegren. *XML: begreppen och tekniken*. Studentlitteratur, 2003.

- [22] David S. Linthicum. *Enterprise application integration*. Addison-Wesley corp, 2000.
- [23] Jamie MacLenna. *Unearth the New Data Mining Features of Analysis Services 2005*. Microsoft Corporation and CMP Media, 2005.
- [24] Fred R. McFadden, Jeffrey A. Hoffer. *Database Management*. 3rd edition, The Benjamin/Cummings Publishing Company inc., 1991.
- [25] Microsoft Corporation. *Integrating Reporting Services into Your Application*. November 2005.
- [26] Microsoft Corporation. *Overview of the Analysis Services Development and Management Environments*. Juli 2005.
- [27] Microsoft TechNet. *Business Intelligence and Data Warehousing in SQL Server 2005*. 15 Juli 2005.
- [28] Microsoft. Windows 2000 Home, Website, 30 Juni 2005. <http://www.microsoft.com/windows2000/server/evaluation/features/dirlist.asp>.
- [29] MSDN, Website, 15 December 2005. <http://www.msdn.microsoft.com>.
- [30] P. Mockapetris. RFC: 1034 *Domain Names - Concepts And Facilities*. November 1987.
- [31] Mario Morejon. *Qliktech*. CRN, 18 juli 2003.
- [32] PR Newswire Europe Ltd. *QlikTech Revolutionizes Business Analysis With Release of QlikView 7*. PR Newswire, 28 April 2005.
- [33] Southern Illinois University Edwardsville, website. <http://www.siu.edu/>.
- [34] Gail Tieh. *An Introduction to OLAP in SQL Server 2005*. Devx, 25 Juni 2004.
- [35] Simon Williams. *The Associative Model of Data*. second edition, Lazy Software Ltd, 2000.
- [36] UBC Commerce. Website, 2000. http://www.sauder.ubc.ca/cgs/itm/itm_glossary.html.
- [37] QlikTech International, Website, 30 September 2005. <http://www.qliktech.com>.
- [38] Wikipedia, Website, 15 October 2005. http://en.wikipedia.org/wiki/Business_Intelligence.
- [39] Ian H. Witten, Eibe Frank. *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

A Ordlista

Ad hoc-förfrågning

Kallas också ofta ad hoc-rapportering [11], detta betyder att användare kan specialbeställa en dataförfrågan (vanligtvis från en databas) i realtid till skillnad från en som begränsar dem till konserverade rapporter. Till exempel så kan rapporteringssystemet lika gärna tillåta val av fält som skall returneras som begränsningskriterier för protokoll som skall returneras. Oftast är ad hoc-rapportering strukturerad så att det tillåter någon utan omfattande SQL-kunskap att ha liknande access till databasen. Ad hoc-rapporteringssystem tillhandahåller oftast en GUI-baserad förfrågan.

Ad hoc-förfrågningar ger alltså användaren tillgång till data med hjälp av en meningsfull kombination av värden för de attribut som finns i dimensionen eller faktatabellen.

Application program interface (API)

API är en uppsättning av rutiner, protokoll och verktyg för att bygga mjukvaruapplikationer. Genom att tillhandahålla alla byggnadsblock, gör ett bra API det lättare att utveckla program. En programmerare sätter sedan ihop dessa block vid utveckling av program.

De flesta operativsystem såsom Windows erbjuder ett API så att programmerare kan skriva applikationer som är förenliga med operativsystemet. Fastän API är designat för programmerare så är de väldigt bra för användaren också, detta för att de garanterar att program som använder sig av gemensamt API kommer ha liknande gränssnitt.

Customer relationship management (CRM)

Syftet med CRM [19] är att möjliggöra bättre hantering av organisationers kunder. Detta görs genom att introducera pålitliga processer och procedurer för interaktion med dessa kunder. Själva termen CRM används för att beskriva antingen mjukvaran eller hela affärsstrategin som är kundbehovsorienterade.

Data-kub

En data-kub [11] är en slags multidimensionell matris som låter en användare utforska och analysera en kollektion av data från många olika perspektiv. Vanligtvis sker detta på ett tredimensionellt sätt dvs. tre faktorer i taget analyseras.

Data mining

Är även känt som ”knowledge-discovery in databases” (KDD). Data mining [39] är utförandet av automatisk genomsökning efter mönster i stora lager av data. För att göra detta använder sig data mining av beräknings-tekniker från statistiken och mönster-rekognosering.

Data warehouse (datalager)

W. H. Inmon som var den förste som använde sig av termen ”data warehouse” [11] beskrev den på följande sätt; Data warehouse är en subjekt-orienterad, integrerad, stabil, tids-varierande samling data vilken används som underlag för ledningsbeslut. Data warehouses tillhandahåller åtkomst till data för komplex analys, kunskapsupptäckt och beslutstagande.

Database Management System (DBMS)

En samling av program som möjliggör lagring, modifiering och extrahering av information från en databas [24]. Det finns många olika typer av DBMS, allt från små system som körs på en PC till väldigt stora system som körs på stordatorer.

Domain Name System (DNS)

DNS [16] är en internet-tjänst som översätter domännamn till IP-adresser. Eftersom domännamn är alfabetiska så är de lättare att komma ihåg. Internet är dock baserat på IP-adresser vilket gör att varje gång ett domännamn används så översätter DNS-tjänsten namnet till motsvarande IP-adress.

Enterprise application integration (EAI)

EAI [22] är användandet av mjukvara och arkitekturiska principer för att integrera en grupp av ett företags applikationer. Det är ett område av datasystemsarkitektur som nått vida kännedom från 2004 och framåt. EAI är relaterat till mellanvaru-teknologier såsom ”message-oriented middleware” (MOM) och datarepresentationsteknologier såsom XML. Nyare EAI-teknologier involverar användandet av web services som en del av en service-orienterad arkitektur, vilket är ett medel för integration.

Enterprise resource planning (ERP)

Företagsresursplaneringssystem (ERP) [12] tar hand om produktion, logistik, distribution, lager, försändelser, fakturering och debitering för ett företag. ERP-mjukvaran hjälper företaget med kontroll över dess affärsaktiviteter såsom försäljning, leverans, reklam, produktion, lagerhantering och personalresurser. Företagsresursplaneringssystemet kallas ofta ”back office” vilket indikerar att kunder och allmänheten inte är direkt involverade.

Extensible Markup Language (XML)

XML [21] är ett enkelt och flexibelt textformat som har sitt ursprung i SGML. Ursprungligen så skapades XML för storskalig elektronisk publicering, men XML spelar också en viktig roll i det varierande utbytet av data på webben.

Extract, transform, and load (ETL)

Detta är en process i data warehousing som involverar att:

- Extrahera data från utomstående källor
- Transformera det för att passa affärsbehoven
- Ladda in det till data warehouse

ETL [18] är viktigt eftersom det är den vägen data laddas till datalagret. ETL kan även referera till en process som laddar data från databaser.

Extreme Programming (XP)

XP [3] är en metod för mjukvaruutveckling. XP fokuserar bland annat på frekvent testning, integration och användarrecension. I XP involveras användaren under hela utvecklingsprocessen.

Gartner (Group)

Är en forsknings- och rådgivningsfirma inom information och teknologi med över 10000 klienter [13]. Gartners affärer består av forskning, konsulting, mätning, evenemang och nyheter.

KXEN

KXEN (Knowledge eXtraction Engines) är ett globalt mjukvaruföretag som tillhandahåller avancerad analys för affärsbesluttare och data mining-poffs.

Key Performance Indicator (KPI)

KPI [29], även känt som Key Success Indicators (KSI), är finansiella eller icke-finansiella mått som används för att reflektera kritiska framgångsfaktorer i en organisation. Dessa används inom BI för att bedöma det nuvarande tillståndet för affärer och sedermera bestämma hur det skall handlas. KPI varierar beroende på organisationens egenskaper. De hjälper organisationen att mäta framstegen som sker mot organisationsmålen.

Lightweight Directory Access Protocol (LDAP)

LDAP [2] är ett mjukvaruprotokoll som möjliggör lokalisering av organisationer, individer och andra resurser såsom filer och hårdvara i ett nätverk. Detta oberoende av om det är publikt på Internet eller på ett lokalt intranet. LDAP är en "lightweight"(mindre mängd kod) version av "Directory Access Protocol"(DAP), som är en del av X.500 vilket är en standardbiblioteksservice i ett nätverk.

Online analytical processing (OLAP)

OLAP är ett tillvägagångssätt för att snabbt tillhandahålla svar på komplexa analytiska förfrågningar. Det är en del av en bredare BI-kategori som även inkluderar ETL, relationsrapportering och data mining. Den typiska applikationen för OLAP är affärsrapportering för försäljning, marknadsföring, ledningsrapportering, affärsprestationsdirektion, budgetering, finansrapportering och liknande områden.

Object Linking and Embedding Database (OLEDB)

OLEDB [33] används för att utföra Universal Data Access (UDA), dvs. att få åtkomst till någon data från någon applikation (textfiler, datablad, e-mail, rationella databaser, adressböcker etc.) från någon lagrings anordning (PC, stordator, Internet, etc.).

Open Database Connectivity (ODBC)

En standard för att kommunicera med DBMS. ODBC (se sektion 2.3) är designat för att vara oberoende av programspråk, databassystem och operativsystem.

Relational Database Management Systems (RDBMS)

RDBMS [24] är en typ av DBMS som lagrar data från relaterade tabeller. Relationella databaser är kraftfulla eftersom de kräver färre antaganden om hur data är relaterad eller hur det kommer att extraheras från databasen. Som ett resultat kan samma databas ses på flera olika sätt. Nästintill alla fullskaliga databas-system är RDBMS. Mindre databas-system kan dock använda sig av andra designers som tillhandahåller mindre flexibilitet när det gäller hantering av förfrågningar.

Supply Chain Management (SCM)

SCM [6] är en process som involverar planering, implementering och kontrolleringsoperationer för leveranskedjan med ändamålet att tillfredsställa kunders krav så effektivt som möjligt. SCM spänner över all förflyttning och lagring av rå material, arbetsprocess-lager och färdiga gods från uppkomst till konsumtion.

THINKANALYTICS

THINKANALYTICS Ltd är en av de ledande försörjarna av realtids förutsägelseanalys och data mining-verktygsteknologi genom dess flaggskeppsprodukt K.wiz och öppna plattform.

B Källkod

```
SET ThousandSep=' ';
SET DecimalSep=',';
SET MoneyThousandSep='.';
SET MoneyDecimalSep=',';
SET MoneyFormat='#.##0,00 kr;-#.##0,00 kr';
SET TimeFormat='hh:mm:ss';
SET DateFormat='YYYY-MM-DD';
SET TimestampFormat='YYYY-MM-DD hh:mm:ss[.fff]';
SET MonthNames='jan;feb;mar;apr;maj;jun;jul;aug;sep;okt;nov;dec';
SET DayNames='må;ti;on;to;fr;lö;sö';

// ===== Behörighets hantering =====

Section application;

// ===== Mapping Time =====

LOAD * INLINE [
    Chain of stores
    SM
    PR
    SALE
    ALEPA
    HUOLTAMOT
];

LOAD * INLINE [ Montho, Month
    01, 1
        02, 2
        03, 3
        04, 4
        05, 5
        06, 6
        07, 7
        08, 8
        09, 9
        10,10
        11,11
        12,12
];

// ===== Applikation =====

//Old SOK data

Sokdata:
```

Load

```
Left(KK,4) as [Year],
Right(KK,2) as [Month],
Left(KK,4)&num(Right(KK,2),00) as YearMonth,
TR4 as [Category code],
Ryhmä as [Category name],
Tuote as [Product name],
text(EAN) as CUEan,
Kauppa as [Chain of stores],
[Myynti kpl] as [Sales quantity],
[My kpl ind kk] as [Sales quantity index/month],
[Myynti kpl ka per ky] as [Sales quantity average/store],
[Ketjuyks peitto lkm] as [Number of stores],
[Myynti €] as [Sales €],
[My ind kk] as [Sales € index/month]
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\temp\SOK*.xls] (biff, embedded
labels, table is [Sheet1$]);
```

concatenate

```
//Ordinary SOK data
//Load SM data
```

Load

```
Right(filebasename(),4) as [Year],
mid(filebasename(),26,2) as [Month],
Right(filebasename(),4)&num(mid(filebasename(),26,2),00) as YearMonth,
TR4 as [Category code],
F2 as [Category name],
Tuote as [Product name],
text(EAN) as CUEan ,
[Etiketin lisäteksti] as [Brand name],
[Myynti kpl] as [Sales quantity],
[Myynti kpl ind kk] as [Sales quantity index/month],
[Myynti kpl ka /ky] as [Sales quantity average/store],
[KY:n peitto lkm] as [Number of stores],
[Myynti €] as [Sales €],
[Myynti-ind.kk] as [Sales € index/month],
Kauppa as [Chain of stores]
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\SOK*.xls] (biff, header is 2
lines, embedded labels, table is [Sheet1$], filters(
Remove(Col, Pos(Top, 35)),
Remove(Col, Pos(Top, 34)),
Remove(Col, Pos(Top, 33)),
Remove(Col, Pos(Top, 32)),
Remove(Col, Pos(Top, 31)),
Remove(Col, Pos(Top, 30)),
Remove(Col, Pos(Top, 29)),
Remove(Col, Pos(Top, 28)),
Remove(Col, Pos(Top, 27)),
Remove(Col, Pos(Top, 26)),
Remove(Col, Pos(Top, 25)),
Remove(Col, Pos(Top, 24)),
Remove(Col, Pos(Top, 23)),
Remove(Col, Pos(Top, 22)),
Remove(Col, Pos(Top, 21)),
Remove(Col, Pos(Top, 20)),
Remove(Col, Pos(Top, 19)),
Remove(Col, Pos(Top, 18)),
Remove(Col, Pos(Top, 17)),
Remove(Col, Pos(Top, 16)),
Remove(Col, Pos(Top, 15)),
Remove(Col, Pos(Top, 14)),
Remove(Col, Pos(Top, 13)),
Remove(Col, Pos(Top, 12)),
ColXtr(11, RowCnd(CellValue, 1, StrCnd(null)), 0),
Top(12, 'SM'),
Replace(12, top, StrCnd(null)),
```

```

Top(12, 'Kauppa')
));

//Load PR data
Sokdata:
Load
Right(filebasename(),4) as [Year],
mid(filebasename(),26,2) as [Month],
Right(filebasename(),4)&num(mid(filebasename(),26,2),00) as YearMonth,
TR4 as [Category code],
[Category name],
Tuote as [Product name],
text(EAN) as CUEan,
[Etiketin lisäteksti] as [Brand name],
[Myynti kpl] as [Sales quantity],
[Myynti kpl ind kk] as [Sales quantity index/month],
[Myynti kpl ka /ky] as [Sales quantity average/store],
[KY:n peitto lkm] as [Number of stores],
[Myynti €] as [Sales €],
[Myynti-ind.kk] as [Sales € index/month],
Kauppa as [Chain of stores]
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\SOK*.xls] (biff, header is 2
lines, embedded labels, table is [Sheet1$], filters(
Remove(Col, Pos(Top, 35)),
Remove(Col, Pos(Top, 34)),
Remove(Col, Pos(Top, 33)),
Remove(Col, Pos(Top, 32)),
Remove(Col, Pos(Top, 31)),
Remove(Col, Pos(Top, 30)),
Remove(Col, Pos(Top, 29)),
Remove(Col, Pos(Top, 28)),
Remove(Col, Pos(Top, 27)),
Remove(Col, Pos(Top, 26)),
Remove(Col, Pos(Top, 25)),
Remove(Col, Pos(Top, 24)),
Remove(Col, Pos(Top, 23)),
Remove(Col, Pos(Top, 22)),
Remove(Col, Pos(Top, 21)),
Remove(Col, Pos(Top, 20)),
Remove(Col, Pos(Top, 19)),
Remove(Col, Pos(Top, 18)),
Remove(Col, Pos(Top, 11)),
Remove(Col, Pos(Top, 10)),
Remove(Col, Pos(Top, 9)),
Remove(Col, Pos(Top, 8)),
Remove(Col, Pos(Top, 7)),
Remove(Col, Pos(Top, 6)),
ColXtr(11, RowCnd(CellValue, 1, StrCnd(null)), 0),
Top(12, 'PR'),
Replace(12, top, StrCnd(null)),
Top(12, 'Kauppa'),
Top(2, 'Category name')
));

//Load SALE data

Load
Right(filebasename(),4) as [Year],
mid(filebasename(),26,2) as [Month],
Right(filebasename(),4)&num(mid(filebasename(),26,2),00) as YearMonth,
TR4 as [Category code],
[Category name],
Tuote as [Product name],
text(EAN) as CUEan,
[Etiketin lisäteksti] as [Brand name],
[Myynti kpl] as [Sales quantity],
[Myynti kpl ind kk] as [Sales quantity index/month],

```



```

[Myynti kpl ka /ky] as [Sales quantity average/store],
[KY:n peitto lkm] as [Number of stores],
[Myynti €] as [Sales €],
[Myynti-ind.kk] as [Sales € index/month],
Kauppa as [Chain of stores]
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\SOK*.xls] (biff, header is 2
lines, embedded labels, table is [Sheet1$], filters(
Remove(Col, Pos(Top, 35)),
Remove(Col, Pos(Top, 34)),
Remove(Col, Pos(Top, 33)),
Remove(Col, Pos(Top, 32)),
Remove(Col, Pos(Top, 31)),
Remove(Col, Pos(Top, 30)),
Remove(Col, Pos(Top, 29)),
Remove(Col, Pos(Top, 28)),
Remove(Col, Pos(Top, 27)),
Remove(Col, Pos(Top, 26)),
Remove(Col, Pos(Top, 25)),
Remove(Col, Pos(Top, 24)),
Remove(Col, Pos(Top, 17)),
Remove(Col, Pos(Top, 16)),
Remove(Col, Pos(Top, 15)),
Remove(Col, Pos(Top, 14)),
Remove(Col, Pos(Top, 13)),
Remove(Col, Pos(Top, 12)),
Remove(Col, Pos(Top, 11)),
Remove(Col, Pos(Top, 10)),
Remove(Col, Pos(Top, 9)),
Remove(Col, Pos(Top, 8)),
Remove(Col, Pos(Top, 7)),
Remove(Col, Pos(Top, 6)),
ColXtr(1, RowCnd(CellValue, 1, StrCnd(null)), 0),
Top(12, 'SALE'),
Replace(12, top, StrCnd(null)),
Top(12, 'Kauppa'),
Top(2, 'Category name'),
Top(6, 'Myynti kpl')
));

```

```
//Load ALEPA data
```

```
Load
```

```

Right(filebasename(),4) as [Year],
mid(filebasename(),26,2) as [Month],
Right(filebasename(),4)&num(mid(filebasename(),26,2),00) as YearMonth,
TR4 as [Category code],
[Category name],
Tuote as [Product name],
text(EAN) as CUEan,
[Etiketin lisäteksti] as [Brand name],
[Myynti kpl] as [Sales quantity],
[Myynti kpl ind kk] as [Sales quantity index/month],
[Myynti kpl ka /ky] as [Sales quantity average/store],
[KY:n peitto lkm] as [Number of stores],
[Myynti €] as [Sales €],
[Myynti-ind.kk] as [Sales € index/month],
Kauppa as [Chain of stores]
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\SOK*.xls] (biff, header is 2
lines, embedded labels, table is [Sheet1$], filters(
Remove(Col, Pos(Top, 35)),
Remove(Col, Pos(Top, 34)),
Remove(Col, Pos(Top, 33)),
Remove(Col, Pos(Top, 32)),
Remove(Col, Pos(Top, 31)),
Remove(Col, Pos(Top, 30)),
Remove(Col, Pos(Top, 23)),

```

```

Remove(Col, Pos(Top, 22)),
Remove(Col, Pos(Top, 21)),
Remove(Col, Pos(Top, 20)),
Remove(Col, Pos(Top, 19)),
Remove(Col, Pos(Top, 18)),
Remove(Col, Pos(Top, 17)),
Remove(Col, Pos(Top, 16)),
Remove(Col, Pos(Top, 15)),
Remove(Col, Pos(Top, 14)),
Remove(Col, Pos(Top, 13)),
Remove(Col, Pos(Top, 12)),
Remove(Col, Pos(Top, 11)),
Remove(Col, Pos(Top, 10)),
Remove(Col, Pos(Top, 9)),
Remove(Col, Pos(Top, 8)),
Remove(Col, Pos(Top, 7)),
Remove(Col, Pos(Top, 6)),
ColXtr(1, RowCnd(CellValue, 1, StrCnd(null)), 0),
Top(12, 'ALEPA'),
Replace(12, top, StrCnd(null)),
Top(12, 'Kauppa'),
Top(2, 'Category name')
));

```

```
//Load HUOLTAMOT data
```

Load

```

Right(filebasename(),4) as [Year],
mid(filebasename(),26,2) as [Month],
Right(filebasename(),4)&num(mid(filebasename(),26,2),00) as YearMonth,
TR4 as [Category code],
[Category name],
Tuote as [Product name],
text(EAN) as CUEan,
[Etiketin lisäteksti] as [Brand name],
[Myynti kpl] as [Sales quantity],
[Myynti kpl ind kk] as [Sales quantity index/month],
[Myynti kpl ka /ky] as [Sales quantity average/store],
[KY:n peitto lkm] as [Number of stores],
[Myynti €] as [Sales €],
[Myynti-ind.kk] as [Sales € index/month],
Kauppa as [Chain of stores]
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\SOK*.xls] (biff, header is 2
lines, embedded labels, table is [Sheet1$], filters(
Remove(Col, Pos(Top, 29)),
Remove(Col, Pos(Top, 28)),
Remove(Col, Pos(Top, 27)),
Remove(Col, Pos(Top, 26)),
Remove(Col, Pos(Top, 25)),
Remove(Col, Pos(Top, 24)),
Remove(Col, Pos(Top, 23)),
Remove(Col, Pos(Top, 22)),
Remove(Col, Pos(Top, 21)),
Remove(Col, Pos(Top, 20)),
Remove(Col, Pos(Top, 19)),
Remove(Col, Pos(Top, 18)),
Remove(Col, Pos(Top, 17)),
Remove(Col, Pos(Top, 16)),
Remove(Col, Pos(Top, 15)),
Remove(Col, Pos(Top, 14)),
Remove(Col, Pos(Top, 13)),
Remove(Col, Pos(Top, 12)),
Remove(Col, Pos(Top, 11)),
Remove(Col, Pos(Top, 10)),
Remove(Col, Pos(Top, 9)),
Remove(Col, Pos(Top, 8)),

```

```

Remove(Col, Pos(Top, 7)),
Remove(Col, Pos(Top, 6)),
ColXtr(1, RowCnd(CellValue, 1, StrCnd(null)), 0),
Top(12, 'HUOLTAMOT'),
Replace(12, top, StrCnd(null)),
Top(12, 'Kauppa'),
Top(2, 'Category name')
));

//Load TERRA data

Load
Right(filebasename(),4) as [Year],
mid(filebasename(),26,2) as [Month],
Right(filebasename(),4)&num(mid(filebasename(),26,2),00) as YearMonth,
TR4 as [Category code],
[Category name],
Tuote as [Product name],
text(EAN) as CUEan,
[Etiketin lisäteksi] as [Brand name],
[Myynti kpl] as [Sales quantity],
[Myynti kpl ind kk] as [Sales quantity index/month],
[Myynti kpl ka /ky] as [Sales quantity average/store],
[KY:n peitto lkm] as [Number of stores],
[Myynti €] as [Sales €],
[Myynti-ind.kk] as [Sales € index/month],
Kauppa as [Chain of stores]
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\SOK*.xls] (biff, header is 2
lines, embedded labels, table is [Sheet1$], filters(
Remove(Col, Pos(Top, 35)),
Remove(Col, Pos(Top, 34)),
Remove(Col, Pos(Top, 33)),
Remove(Col, Pos(Top, 32)),
Remove(Col, Pos(Top, 31)),
Remove(Col, Pos(Top, 30)),
Remove(Col, Pos(Top, 29)),
Remove(Col, Pos(Top, 28)),
Remove(Col, Pos(Top, 27)),
Remove(Col, Pos(Top, 26)),
Remove(Col, Pos(Top, 25)),
Remove(Col, Pos(Top, 24)),
Remove(Col, Pos(Top, 23)),
Remove(Col, Pos(Top, 22)),
Remove(Col, Pos(Top, 21)),
Remove(Col, Pos(Top, 20)),
Remove(Col, Pos(Top, 19)),
Remove(Col, Pos(Top, 18)),
Remove(Col, Pos(Top, 17)),
Remove(Col, Pos(Top, 16)),
Remove(Col, Pos(Top, 15)),
Remove(Col, Pos(Top, 14)),
Remove(Col, Pos(Top, 13)),
Remove(Col, Pos(Top, 12)),
Remove(Col, Pos(Top, 11)),
Remove(Col, Pos(Top, 10)),
Remove(Col, Pos(Top, 9)),
Remove(Col, Pos(Top, 8)),
Remove(Col, Pos(Top, 7)),
Remove(Col, Pos(Top, 6)),
ColXtr(1, RowCnd(CellValue, 1, StrCnd(null)), 0),
Top(12, 'TERRA'),
Replace(12, top, StrCnd(null)),
Top(12, 'Kauppa'),
Top(2, 'Category name')
));

concatenate
// Data from 04-05 2005

```

```
//load SM data 04-05 2005
```

Load

```
right(filebasename(),4) as Year,  
mid(filebasename(),10,2) as Month,  
Right(filebasename(),4)&num(mid(filebasename(),10,2),00) as YearMonth,  
left(TR4,4) as [Category code],  
mid(TR4,7,30) as [Category name],  
Tuote as [Product name],  
text(EAN) as CUEan,  
[Myynti kpl] as [Sales quantity],  
[My kpl ind kk] as [Sales quantity index/month],  
[Myynti kpl ka /ky] as [Sales quantity average/store],  
[KY:n peitto lkm] as [Number of stores],  
[Myynti €] as [Sales €],  
My.ind.kk as [Sales € index/month],  
Kauppa as [Chain of stores]  
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\temp\Cederroth*.xls] (biff,  
header is 3 lines, embedded labels, table is [Sheet1$], filters(  
Remove(Col, Pos(Top, 33)),  
Remove(Col, Pos(Top, 32)),  
Remove(Col, Pos(Top, 31)),  
Remove(Col, Pos(Top, 30)),  
Remove(Col, Pos(Top, 29)),  
Remove(Col, Pos(Top, 28)),  
Remove(Col, Pos(Top, 27)),  
Remove(Col, Pos(Top, 26)),  
Remove(Col, Pos(Top, 25)),  
Remove(Col, Pos(Top, 24)),  
Remove(Col, Pos(Top, 23)),  
Remove(Col, Pos(Top, 22)),  
Remove(Col, Pos(Top, 21)),  
Remove(Col, Pos(Top, 20)),  
Remove(Col, Pos(Top, 19)),  
Remove(Col, Pos(Top, 18)),  
Remove(Col, Pos(Top, 17)),  
Remove(Col, Pos(Top, 16)),  
Remove(Col, Pos(Top, 15)),  
Remove(Col, Pos(Top, 14)),  
Remove(Col, Pos(Top, 13)),  
Remove(Col, Pos(Top, 12)),  
Remove(Col, Pos(Top, 11)),  
Remove(Col, Pos(Top, 10)),  
Top(2, 'Tuote'),  
Top(3, 'EAN'),  
Top(1, 'TR4'),  
ColXtr(1, RowCnd(CellValue, 1, StrCnd(null)), 0),  
Top(10, 'SM'),  
Replace(10, top, StrCnd(null)),  
Top(10, 'Kauppa')  
));
```

```
//Load PR data 04-05 2005
```

concatenate

Load

```
right(filebasename(),4) as Year,  
mid(filebasename(),10,2) as Month,  
Right(filebasename(),4)&num(mid(filebasename(),10,2),00) as YearMonth,  
left(TR4,4) as [Category code],  
mid(TR4,7,30) as [Category name],  
Tuote as [Product name],  
text(EAN) as CUEan,  
[Myynti kpl] as [Sales quantity],  
[My kpl ind kk] as [Sales quantity index/month],  
[Myynti kpl ka /ky] as [Sales quantity average/store],
```

```

[KY:n peitto lkm] as [Number of stores],
[Myynti €] as [Sales €],
My.ind.kk as [Sales € index/month],
Kauppa as [Chain of stores]
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\temp\Cederroth*.xls] (biff,
header is 3 lines, embedded labels, table is [sheet1$], filters(
Remove(Col, Pos(Top, 33)),
Remove(Col, Pos(Top, 32)),
Remove(Col, Pos(Top, 31)),
Remove(Col, Pos(Top, 30)),
Remove(Col, Pos(Top, 29)),
Remove(Col, Pos(Top, 28)),
Remove(Col, Pos(Top, 27)),
Remove(Col, Pos(Top, 26)),
Remove(Col, Pos(Top, 25)),
Remove(Col, Pos(Top, 24)),
Remove(Col, Pos(Top, 23)),
Remove(Col, Pos(Top, 22)),
Remove(Col, Pos(Top, 21)),
Remove(Col, Pos(Top, 20)),
Remove(Col, Pos(Top, 19)),
Remove(Col, Pos(Top, 18)),
Remove(Col, Pos(Top, 17)),
Remove(Col, Pos(Top, 16)),
Remove(Col, Pos(Top, 9)),
Remove(Col, Pos(Top, 8)),
Remove(Col, Pos(Top, 7)),
Remove(Col, Pos(Top, 6)),
Remove(Col, Pos(Top, 5)),
Remove(Col, Pos(Top, 4)),
ColXtr(1, RowCnd(CellValue, 1, StrCnd(null)), 0),
Top(1, 'TR4'),
Top(2, 'Tuote'),
Top(3, 'EAN'),
Top(10, 'PR'),
Replace(10, top, StrCnd(null)),
Top(10, 'Kauppa')
));

```

//Load SALE data 04-05 2005

concatenate

Load

```

right(filebasename(),4) as Year,
mid(filebasename(),10,2) as Month,
Right(filebasename(),4)&num(mid(filebasename(),10,2),00) as YearMonth,
left(TR4,4) as [Category code],
mid(TR4,7,30) as [Category name],
Tuote as [Product name],
text(EAN) as CUEan,
[Myynti kpl] as [Sales quantity],
[My kpl ind kk] as [Sales quantity index/month],
[Myynti kpl ka /ky] as [Sales quantity average/store],
[KY:n peitto lkm] as [Number of stores],
[Myynti €] as [Sales €],
My.ind.kk as [Sales € index/month],
Kauppa as [Chain of stores]
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\temp\Cederroth*.xls] (biff,
header is 3 lines, embedded labels, table is [sheet1$], filters(
Remove(Col, Pos(Top, 33)),
Remove(Col, Pos(Top, 32)),
Remove(Col, Pos(Top, 31)),
Remove(Col, Pos(Top, 30)),
Remove(Col, Pos(Top, 29)),
Remove(Col, Pos(Top, 28)),
Remove(Col, Pos(Top, 27)),
Remove(Col, Pos(Top, 26)),
Remove(Col, Pos(Top, 25)),
Remove(Col, Pos(Top, 24)),

```

```

Remove(Col, Pos(Top, 23)),
Remove(Col, Pos(Top, 22)),
Remove(Col, Pos(Top, 15)),
Remove(Col, Pos(Top, 14)),
Remove(Col, Pos(Top, 13)),
Remove(Col, Pos(Top, 12)),
Remove(Col, Pos(Top, 11)),
Remove(Col, Pos(Top, 10)),
Remove(Col, Pos(Top, 9)),
Remove(Col, Pos(Top, 8)),
Remove(Col, Pos(Top, 7)),
Remove(Col, Pos(Top, 6)),
Remove(Col, Pos(Top, 5)),
Remove(Col, Pos(Top, 4)),
ColXtr(1, RowCnd(CellValue, 1, StrCnd(null)), 0),
Top(1, 'TR4'),
Top(2, 'Tuote'),
Top(3, 'EAN'),
Top(10, 'SALE'),
Replace(10, top, StrCnd(null)),
Top(10, 'Kauppa')
));

//Load ALEPA data 04-05 2005
concatenate
Load
right(filebasename(),4) as Year,
mid(filebasename(),10,2) as Month,
Right(filebasename(),4)&num(mid(filebasename(),10,2),00) as YearMonth,
left(TR4,4) as [Category code],
mid(TR4,7,30) as [Category name],
Tuote as [Product name],
text(EAN) as CUEan,
[Myynti kpl] as [Sales quantity],
[My kpl ind kk] as [Sales quantity index/month],
[Myynti kpl ka /ky] as [Sales quantity average/store],
[KY:n peitto lkm] as [Number of stores],
[Myynti €] as [Sales €],
My.ind.kk as [Sales € index/month],
Kauppa as [Chain of stores]
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\temp\Cederroth*.xls] (biff,
header is 3 lines, embedded labels, table is [sheet1$], filters(
Remove(Col, Pos(Top, 33)),
Remove(Col, Pos(Top, 32)),
Remove(Col, Pos(Top, 31)),
Remove(Col, Pos(Top, 30)),
Remove(Col, Pos(Top, 29)),
Remove(Col, Pos(Top, 28)),
Remove(Col, Pos(Top, 21)),
Remove(Col, Pos(Top, 20)),
Remove(Col, Pos(Top, 19)),
Remove(Col, Pos(Top, 18)),
Remove(Col, Pos(Top, 17)),
Remove(Col, Pos(Top, 16)),
Remove(Col, Pos(Top, 15)),
Remove(Col, Pos(Top, 14)),
Remove(Col, Pos(Top, 13)),
Remove(Col, Pos(Top, 12)),
Remove(Col, Pos(Top, 11)),
Remove(Col, Pos(Top, 10)),
Remove(Col, Pos(Top, 9)),
Remove(Col, Pos(Top, 8)),
Remove(Col, Pos(Top, 7)),
Remove(Col, Pos(Top, 6)),
Remove(Col, Pos(Top, 5)),
Remove(Col, Pos(Top, 4)),
ColXtr(1, RowCnd(CellValue, 1, StrCnd(null)), 0),
Top(1, 'TR4'),

```

```

Top(2, 'Tuote'),
Top(3, 'EAN'),
Top(10, 'ALEPA'),
Replace(10, top, StrCnd(null)),
Top(10, 'Kauppa')
));

//Load HUOLTAMOT data 04-05 2005
concatenate
Load
right(filebasename(),4) as Year,
mid(filebasename(),10,2) as Month,
Right(filebasename(),4)&num(mid(filebasename(),10,2),00) as YearMonth,
left(TR4,4) as [Category code],
mid(TR4,7,30) as [Category name],
Tuote as [Product name],
text(EAN) as CUEan,
[Myynti kpl] as [Sales quantity],
[My kpl ind kk] as [Sales quantity index/month],
[Myynti kpl ka /ky] as [Sales quantity average/store],
[KY:n peitto lkm] as [Number of stores],
[Myynti €] as [Sales €],
My.ind.kk as [Sales € index/month],
Kauppa as [Chain of stores]
FROM [Q:\qlikview\Database\Cederroth\Sales\POS\SOK\temp\Cederroth*.xls] (biff,
header is 3 lines, embedded labels, table is [sheet1$], filters(
Remove(Col, Pos(Top, 27)),
Remove(Col, Pos(Top, 26)),
Remove(Col, Pos(Top, 25)),
Remove(Col, Pos(Top, 24)),
Remove(Col, Pos(Top, 23)),
Remove(Col, Pos(Top, 22)),
Remove(Col, Pos(Top, 21)),
Remove(Col, Pos(Top, 20)),
Remove(Col, Pos(Top, 19)),
Remove(Col, Pos(Top, 18)),
Remove(Col, Pos(Top, 17)),
Remove(Col, Pos(Top, 16)),
Remove(Col, Pos(Top, 15)),
Remove(Col, Pos(Top, 14)),
Remove(Col, Pos(Top, 13)),
Remove(Col, Pos(Top, 12)),
Remove(Col, Pos(Top, 11)),
Remove(Col, Pos(Top, 10)),
Remove(Col, Pos(Top, 9)),
Remove(Col, Pos(Top, 8)),
Remove(Col, Pos(Top, 7)),
Remove(Col, Pos(Top, 6)),
Remove(Col, Pos(Top, 5)),
Remove(Col, Pos(Top, 4)),
ColXtr(1, RowCnd(CellValue, 1, StrCnd(null)), 0),
Top(1, 'TR4'),
Top(2, 'Tuote'),
Top(3, 'EAN'),
Top(10, 'HUOLTAMOT'),
Replace(10, top, StrCnd(null)),
Top(10, 'Kauppa')
));

//Load from Movex

ODBC CONNECT TO QlikViewDTA (UserId is XXXXXX, Password is XXXXXX);

MITMAD:
Load
MMITNO as ItemNumber,
Text(MMSPE5) as CUEan
where exists(CUEan,Text(MMSPE5))

```

;

SQL SELECT

MMITNO,
MMSPE5
FROM MVXBDBTA.MITMAD;

Load

MMITNO **as** ItemNumber,
Text(MMITGR) **as** ItemGroup,
Text(MMITCL) **as** ProductGroup,
MMITTY **as** Itemtype,
MMBUAR **as** BusinessGroup,
MMITDS **as** Description,
MMSTAT **as** Status
where exists(ItemNumber,MMITNO)
;

SQL SELECT

MMITNO,
MMITGR,
MMITCL,
MMITTY,
MMBUAR,
MMITDS,
MMSTAT
FROM MVXBDBTA.MITMAS;

Load

MMITNO **as** ItemNumber,
Text(MMEACD) **as** SKUEan
where exists(ItemNumber,MMITNO)
;

SQL SELECT

MMITNO,
MMEACD

FROM MVXBDBTA.MITMPR;

CSYTAB:

Load

text(CTSTKY) **as** CTSTKY,
CTTX40,
CTTX15,
CTSTCO
;

SQL SELECT

CTSTKY, CTTX40, CTTX15, CTSTCO
from CSYTAB
WHERE CTCONO=001;

ArtCode:

LOAD

Text(CTSTKY) **as** ItemGroup,
CTTX40 **as** LongITGRPdsc,
CTTX15 **as** ShortItemGrdesc
resident CSYTAB
WHERE CTSTCO='ITGR' **AND exists**(ItemGroup,Text(CTSTKY))
;


```
ProdCode:
LOAD
Text(CTSTKY) as ProductGroup,
CTTX40 as LongPrGrDesc,
CTTX15 as ShortPgrpdesc
resident CSYTAB
WHERE CTSTCO='ITCL' AND exists(ProductGroup,Text(CTSTKY))
;
```

```
LOAD
now() AS ExecTime
AUTOGENERATE(1);
```