

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Anže Mis

**ORODJE QLIKVIEW PRI IZDELAVI APLIKACIJE ZA FINANČNI  
KONTROLING**

Diplomska naloga  
na univerzitetnem študiju

Mentor: doc. dr. Marko Bajec

Ljubljana 2009



Št. naloge: 01502/2008

Datum: 01.09.2008

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ANŽE MIS**

Naslov: **ORODJE QLIKVIEW PRI IZDELAVI APLIKACIJE ZA FINANČNI  
KONTROLING**

**THE DEVELOPMENT OF AN APPLICATION FOR FINANCIAL  
CONTROLLING BY THE USE OF CLICKVIEW TOOLSET**


Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

S pojmom poslovno obveščanje oziroma poslovna inteligenca označujemo informacijske sisteme za podporo upravljanju poslovanja. Takšni sistemi so postali za mnoge poslovne sisteme neobhodni, saj omogočajo spremljanje poslovanja ter analizo trendov na vseh poslovnih področjih.

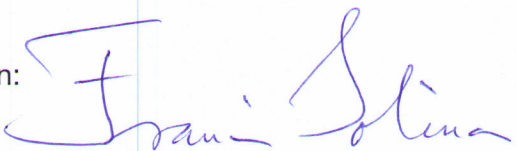
V okviru diplomske naloge najprej predstavite področje poslovne inteligenca. Nato preučite in opišite orodje QlikView, ki je eno prodornejših razvojnih okolij za sisteme poslovne inteligenca. Uporabnost orodja prikažite na primeru.

Mentor:

  
doc. dr. Marko Bajec



Dekan:

  
prof. dr. Franc Solina



## ZAHVALA

Za uspešen nastanek diplomskega dela bi se rad zahvalil doc. dr. Marku Bajcu, ki me je pri nastajanju dela strokovno in potrpežljivo usmerjal. Za konstruktivne vsebinske pripombe sem hvaležen tudi ožjima sodelavcema Boštjanu Kožuhu in Mihu Batiču. Zahvalo bi rad poklonil tudi svojim staršem in starim staršem, ki so me vsa leta neutrudno spodbujali, me motivirali, mi nudili sredstva ter prijetno okolje za študij. Svoji puncu Petri bi se zahvalil za slovnični pregled diplomskega dela. Zahvala gre tudi ostalim sodelavcem, študijskim kolegom in prijateljem, ki so mi polepšali študijsko življenje in me podpirali.



# KAZALO

POVZETEK .....	1
ABSTRACT.....	3
1 UVOD .....	5
1.1 POSLOVNO OBVEŠČANJE.....	6
1.1.1 Definicije.....	6
1.1.2 Pomen dobrega informiranja, znanja in ukrepanja .....	6
1.1.3 Pregled poslovnega obveščanja in koncepta OLAP skozi čas .....	6
1.2 ARHITEKTURE OLAP .....	8
1.2.1 Definicija OLAP .....	8
1.2.2 Arhitekture .....	8
1.2.3 Podatkovno skladiščenje .....	8
1.2.4 Glavne pomanjkljivosti sistemov OLAP .....	8
1.2.5 Pohitritve z analizo v spominu.....	9
2 ORODJE QLIKVIEW .....	10
2.1 KLASIČEN KONCEPT (OLAP) .....	10
2.2 INOVATIVEN KONCEPT (QLIKVIEW).....	11
2.3 ASOCIATIVNI PODATKOVNI MODEL (AQL).....	12
2.4 QLIKVIEW ETL SKRIPT.....	15
2.4.1 Osnovni gradniki.....	15
2.4.2 Izbrane funkcionalnosti.....	16
2.4.2.1 Predpona ADD.....	17
2.4.2.2 Ukaz BINARY .....	17
2.4.2.3 Predpona BUFFER .....	17
2.4.2.4 Kontrolna stavka SUB in CALL.....	17
2.4.2.5 Predpona CROSSTABLE.....	18

2.4.2.6	Stavka DROP FIELD in DROP TABLE .....	18
2.4.2.7	Kontrolna stavka FOR...NEXT in FOR...EACH .....	18
2.4.2.8	Predpona MAPPING.....	19
2.4.2.9	Predpona HIERARCHY in HIERARCHY BELONGS TO .....	19
2.4.2.10	Predpona INTERVALMATCH.....	20
2.4.3	Izrazi.....	21
2.4.3.1	Operatorji .....	21
2.4.3.2	Agregacijske funkcije .....	22
2.5	PODATKOVNE STRUKTURE.....	23
2.5.1	Logične tabele .....	23
2.5.2	Asociacije med logičnimi tabelami.....	23
2.5.2.1	Pomanjkljivosti samodejnega asociiranja .....	23
2.5.3	QVD datoteke .....	24
2.5.3.1	Prednosti uporabe QVD datotek .....	24
2.6	VARNOST.....	25
2.7	NADZORNE PLOŠČE.....	26
2.7.1	Definicija nadzornih plošč .....	26
2.7.2	Kategorije nadzornih plošč .....	27
2.7.2.1	Podpora pri odločanju za doseganje strateških ciljev .....	27
2.7.2.2	Analitična funkcija.....	27
2.7.2.3	Meritve na operativnem nivoju .....	27
2.7.3	Merilniki in načini prikazovanja podatkov .....	27
2.7.4	Najpogostejše napake pri izgradnji nadzornih plošč.....	28
2.8	POSTAVITEV IN IZDELAVA OBJEKTOV .....	29
2.8.1	Lastnosti dokumenta .....	29
2.8.2	Delovni zvezki .....	29
2.8.3	Objekti.....	29
2.8.4	Izvoz in tiskanje podatkov .....	30

2.8.5	Avtomatizacija z makroji .....	30
2.8.6	Grafikoni in tabele .....	31
2.8.6.1	Analiza množic (angl. set analysis).....	31
2.8.6.2	Agregacije na strani dimenzij s funkcijo Aggr( ).....	32
2.9	QLIKVIEW STREŽNIK .....	34
2.9.1	Upravljanje.....	34
2.9.2	Varnost .....	35
2.9.2.1	Komunikacija glede na vrsto odjemalca .....	35
2.9.2.2	Datotečni sistem.....	35
2.9.2.3	Overitev uporabnika.....	36
2.9.3	Arhitektura .....	36
2.9.3.1	Spletni strežniki.....	37
2.9.3.2	»Tunneling mode« komunikacija.....	37
2.9.3.3	Repozitorij objektov.....	37
2.9.3.4	Odjemalci .....	37
2.10	QLIKVIEW PUBLISHER.....	40
2.10.1	QlikView Publisher sestavljajo:.....	40
2.10.2	Tipi virov.....	40
2.10.3	Avtorizacija s Publisherjem .....	41
3	IZDELAVA APLIKACIJE ZA FINANČNI KONTROLING .....	42
3.1	POSLOVNO OBVEŠČANJE ZA PODPORO RAČUNOVODSKEGA ANALIZIRANJA, NADZIRANJA IN NAČRTOVANJA .....	42
3.2	NAMEN KONTROLINGA.....	42
3.3	VIR INFORMACIJ .....	42
3.4	DODATNI VIRI .....	42
3.5	KONTNI NAČRT IN KONTNE PREGLEDNICE.....	43
3.6	VSEBINSKI OPIS ETL POSTOPKA .....	43
3.7	IZGRADNJA NADZORNE PLOŠČE Z ANALIZO MNOŽIC .....	45

3.8	ANALIZIRANJE RAČUNOVODSKIH IZKAZOV .....	46
3.8.1	Analiza s kazalniki .....	46
3.8.2	Vodoravna analiza.....	46
3.8.3	Stroškovni ali prihodkovni odmiki .....	47
3.8.4	Navpična analiza .....	47
3.8.5	Drugi načini prikaza.....	48
4	SKLEPNE UGOTOVITVE .....	49
4.1	ZAKLJUČEK.....	49
4.2	IZBOLJŠAVE PRI APLIKACIJI ZA FINANČNI KONTROLING .....	49
4.3	POSLOVNO OBVEŠČANJE IN SOA.....	50
4.4	ORODJE QLIKVIEW V PRIHODNOSTI.....	50
5	LITERATURA.....	51
6	SEZNAM TABEL IN SLIK .....	52
	IZJAVA O AVTORSTVU.....	54



## SEZNAM UPORABLJENIH KRATIC IN SIMBOLOV

ACTIVEX	A component object model: komponenta objektnega modela
AJAX	Asynchronous JavaScript and XML: asinhroni JavaScript in XML
AQL™	Associative Query Logic: asociativni podatkovni model
BI	Business Intelligence: poslovno obveščanje
CPU	Central Procesor Unit: centralna procesna enota
CSV	Comma Separated Values: z vejicami ločene vrednosti
DMS	Document Metadata Service: storitev za upravljanje dokumentnih metapodatkov
DS	Directory Service: storitev za upravljanje z mapami
DSC	Directory Service Component: komponenta storitve za upravljanje z mapami
DSP	Directory Service Provider: oskrbovalec storitve za upravljanje z mapami
DSS	Decision Support Systems: sistemi za podporo pri odločanju
ERP	Enterprise Resource Planing: sistem za podporo poslovanja organizacij
ETL	Extract Transform Load: povzemanje, transformiranje in prenos podatkov
HOLAP	Hybrid OLAP: hibridni OLAP
HTML	HyperText Markup Language: označevalni jezik za oblikovanje dokumentov
HTTP	HyperText Transfer Protocol: protokol za prenos označevalnega jezika
HTTPS	HyperText Transfer Protocol over Secure Socket: protokol za prenos označevalnega jezika skozi varna vrata
IE	Internet Explorer: internetni raziskovalec
IMOLAP	In Memory OLAP: OLAP v spominu
IT	Information Technology: informacijske tehnologije
JRE	Java Runtime Environment: okolje za zaganjanje Jave
KPI	Key Performance Indicator: ključni kazalnik poslovanja
MOLAP	Multidimensional OLAP: večdimenzionalni OLAP
OCX	Object Linking and Embeddig Custom controls: objektno povezovanje in vstavljanje prilagojenih kontrol
ODBC	Open Database Connectivity: protokol za povezavo s podatkovnimi bazami
OLAP	On-Line Analytical Processing: sprotna analitična obdelava podatkov

OLE DB	Object Linking and Embedding Database: protokol za povezavo s podatkovnimi bazami
OLTP	On-Line Transactual Processing: sprotna obdelava transakcij
OS	Operational System: Operacijski sistem
QPV	QlikView Protocol: QlikView protokol
QVD	QlikView Data: QlikView podatki
QVS	QlikView Server: QlikView strežnik
QVW	QlikView Document: QlikView dokument
RAM	Random Access Memory: Pomnilnik z naključnim dostopom
ROLAP	Relational OLAP: relacijski OLAP
RTOLAP	Real Time OLAP: OLAP v realnem času
SOA	Service-Oriented Architecture: storitveno usmerjena arhitektura
SQL	Structured Query Language: strukturiran povpraševalni jezik
TXT	Text file: tekstovna datoteka
UI	User Interface: uporabniški vmesnik
UTF-8	8-bit Unicode Transformation Format: 8-bitni univerzalni format za transformacijo
WIN	Operational system Windows: operacijski sistem okna
XLS	Excel Worksheet: Excel delovni zvezek
XML	Extensible Markup Language: razširljivi označevalni jezik
IIS	Internet Information Services: internetne spletne storitve



## POVZETEK

Organizacije se danes vedno bolj zavedajo pomembnosti naložb v IT sisteme. Do sedaj so bile investicije usmerjene predvsem v razvoj informacijskih sistemov za zajem podatkov. Sedaj so pred nami časi, ko je potrebno podatke zbirati in jih analizirati. Poslovno obveščanje postaja vse popularnejše področje v informacijski tehnologiji, saj nove platforme razširjajo svoje zmogljivosti preko klasičnih poizvedovanj in statičnih oblik poročanja. Svoje rešitve neprestano prilagajajo strojnim zmogljivostim računalnikov. Pomnilniške implementacije koncepta OLAP obljublajo boljšo odzivnost tudi pri velikih količinah podatkov in zato postajajo vedno bolj priljubljene.

Namen te diplomske naloge je predstaviti orodje QlikView, teoretično prikazati njegov podatkovni model in z njim izdelati aplikacijo za finančni kontroling.

Na začetku je opisan klasičen koncept OLAP in njegova arhitektura. Izpostavljene so njegove pomanjkljivosti in ponujene alternativne rešitve z analizo v spominu. V osrednjem delu je pozornost namenjena orodju QlikView. Njegove najuporabnejše lastnosti so okrepljene tudi s praktičnimi primeri. Omenjene so tudi nadzorne plošče, ki so nepogrešljive pri skoraj vsaki aplikaciji poslovnega obveščanja. Orodje je s svojo strežniško arhitekturo dostopno odjemalcem na različnih platformah. Nudi tudi visoko stopnjo varnostnih prijemov ter različne načine za distribucijo in osveževanje podatkov s QlikView Publisherjem. Za potrebe pri izdelavi aplikacije za finančni kontroling so definirani viri podatkov in struktura kontnih preglednic, ki so temelj za prikazovanje standardnih in analitičnih izkazov stanja v računovodstvu. Ponazorjena je tudi velika uporabnost analize množic pri izdelavi nadzornih plošč. Delo zaključujejo njegova kratka opredelitev ter smernice nadaljnjega razvoja aplikacije in orodja QlikView.

**Ključne besede:** Poslovno obveščanje, koncept OLAP, QlikView, finančni kontroling.



## ABSTRACT

Nowadays organizations realize the importance of IT investments. Until now the focus has been mainly directed at the development of information systems for gathering data. Now the collected data need to be analyzed. Business intelligence systems (BI) are becoming more popular, because they offer platforms with capabilities far beyond traditional querying and static reporting. Solutions are continuously being updated to take full advantage of hardware capabilities. Growth of technology offers “in memory” OLAP implementations to improve their analytical response times.

The goal of this thesis is to present QlikView, show its innovative data model based on associative query logic (AQL) and to build an application for finance controlling.

At the beginning classical OLAP approach and its architecture is described. Its main problems are exposed and alternative solutions with “in memory” analysis described. The main part of this thesis is dedicated to QlikView. Some of its most powerful features are introduced. Later on theory about dashboards is included as they are indispensable in every business intelligence application. QlikView server architecture provides opportunity for clients to connect and analyze data no matter what operating system they are running on. With QlikView Publisher highly integrated security can be provided, while reloaded and authorized data can be distributed to the end users. The last part of thesis defines the structure for account schedules, which are necessary for displaying the main charts in application for finance controlling. With the dashboard design the big advantage of using the set analysis feature is also demonstrated. The thesis concludes with the short forecast and improvements plan for the application and QlikView.

**Keywords:** Business intelligence, concept OLAP, QlikView, finance controlling



# 1 UVOD

V zadnjih desetletjih smo priča hitremu razvoju računalniških tehnologij. Programske rešitve se vedno znova dopolnjujejo in spreminjajo. Zamenjujejo se operacijski sistemi in strojna zmogljivost računalnikov. Premo sorazmerno z omenjenim razvojem sta se povečala tudi količina in obseg informacijskih sistemov, ki današnjim družbam nudijo digitalno podporo kritičnih in pomožnih poslovnih funkcij.

Hitri napredek je pustil tudi nekaj posledic. Organizacije se pogosto srečujejo z vprašanji, kaj od velike količine aplikacij, nadgradenj in podatkovnih baz je nujno vzdrževati in kaj ne. Zaradi možnosti hitrega medmrežnega komuniciranja posameznih delovnih postaj, se družbe odločajo za prenavo ali izgradnjo celovitega informacijskega sistema za podporo poslovanja (ERP). Eden glavnih ciljev je, da bi lahko vzdrževali podatke na enem mestu oz. skupni podatkovni bazi. Velike količine shranjenih podatkov pa nimajo posebne vrednosti, dokler jih ne zberemo skupaj in pripravimo poročila. Statična oblika poročanja danes ne zadostuje več in pojavljajo se potrebe po dodatnih oblikah poročanja. Ker pa so operacije zelo zahtevne in obremenjujoče za transakcijski sistem, se podatki izvažajo v podatkovna skladišča za analitične potrebe. Te nam omogočajo prenose podatkov iz različnih podatkovnih virov. Podatki se najprej zbirajo in nato transformirajo v optimalno strukturo za analiziranje. Ena prvih idej je bila, da se podatki shranjujejo v obliki večdimenzionalnih kock, ki so temeljna podlaga za analize.

Tradicionalne metode bodo v prihodnosti najverjetneje nadomestile tehnologije za analizo v spominu. Orodje QlikView je eden od vodilnih produktov iz tega področja in ponuja povsem drugačen pristop pri izgradnji aplikacij za poslovno obveščanje. Podatke hrani v asociativnem podatkovnem modelu, analizo pa izvaja sproti na uporabnikovo zahtevo. Ob zahtevnejših primerih in večjemu številu uporabnikov uporablja strežniško arhitekturo. Uporabnik lahko analizira na delovni postaji, procesiranje in pomnjenje pa se izvajata na strežniku. Z uporabo tega orodja je razvoj sistema za poslovno obveščanje lahko učinkovitejši in cenejši, saj za analizo v spominu ne potrebuje podatkovnega skladišča.

Cilj diplomske naloge je predstaviti orodje QlikView, njegove najuporabnejše lastnosti in povsem drugačen pristop razvoja. Ker ima orodje zelo zmogljive kontrolne stavke, nam omogoča eleganten razvoj ETL (angl. Extract Transform Load) procedure. Glavna ideja pri zasnovi aplikacije za finančni kontroling je, da strukturizirano obliko kontnih preglednic preberemo iz Excel datotek, tabelo glavne knjige in tabele dimenzij pa neposredno iz izvornega operacijskega sistema. Ker orodje omogoča branje podatkov iz različnih virov, lahko združimo podatke iz različnih podatkovnih baz in znotraj aplikacije sočasno analiziramo in kontroliramo računovodske izkaze različnih družb ali poslovnih enot.

## 1.1 POSLOVNO OBVEŠČANJE

### 1.1.1 Definicije

Poslovno obveščanje predstavlja zaporedje procesov in transformacij, ki z inteligentno uporabo razpoložljivih podatkov pomagajo organizaciji izluščiti informacije in s tem prinašati določene konkurenčne prednosti [12].

Poslovno obveščanje so procesi, tehnologije in orodja, ki so potrebni za pretvorbo podatkov v informacije, informacije v znanje, znanje v akcije in akcije v načrte do kvalitetnejših poslovnih odločitev [7].

### 1.1.2 Pomen dobrega informiranja, znanja in ukrepanja

Analitično poročanje izhaja iz potrebe po tekočem nadziranju poslovanja ali upravljanja kakšne druge dejavnosti [8].

Cilj nadzornikov, poslovodij in analitikov je, da na podlagi ugotovljenih nepravilnosti ali nepričakovanih rezultatov poiščejo njihove vzroke in hitro ukrepajo [8].

Izgradnja informacijskega sistema za podporo poslovanja organizacij je dolgotrajno in nikoli končano delo. Še posebej kadar stremimo k neprestani informacijski podpori dinamičnih poslovnih pravil. ERP sistemi in tudi manj obsežni transakcijski sistemi nam nudijo omejen nabor statičnega poročanja. Sistemi so optimizirani za zajem podatkov in ne za analize.

Po definiciji bi morala imeti dobra informacija tri najpomembnejše značilnosti [8]:

- primernost – to pomeni da mora biti informacija oblikovana na način, da jo bo končni uporabnik najlažje zaznal,
- pravočasnost in točnost – dobiti informacijo ob pravem času,
- ustreznost – prave informacije morajo dobiti pravi ljudje (poslovodja poslovne enote ne potrebuje podatkov od drugih enot, kar pa ne velja npr. za glavnega finančnega direktorja).

Zaradi teh zahtev je potrebno pri sistemu za poslovno obveščanje že v fazi načrtovanja odgovoriti na ključna vprašanja:

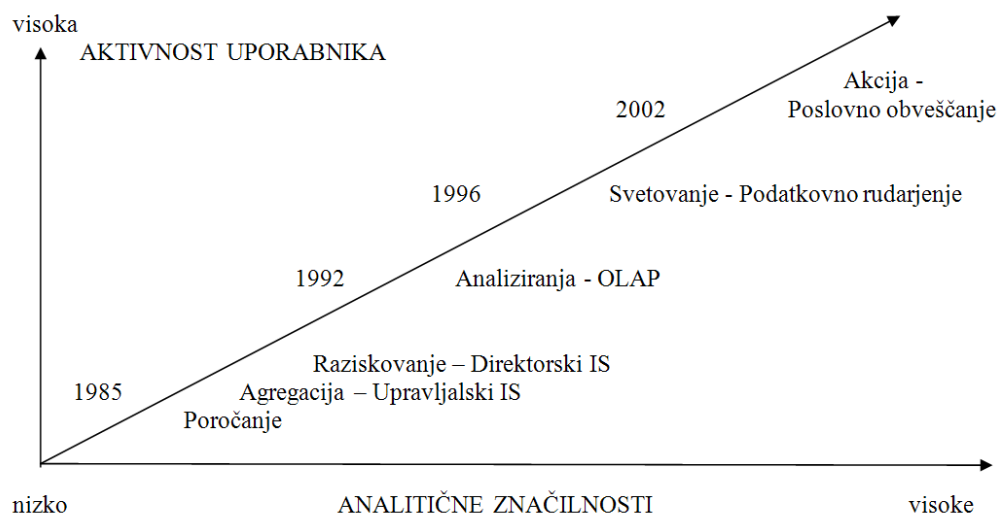
- komu je namenjen in kaj končni uporabnik pričakuje,
- kaj bo končni uporabnik delal s podatki (jih potrebuje za nadaljnjo obdelavo ali zgolj kontrolo poslovanja),
- iz kje bomo dobivali izvirne podatke,
- o kakšni količini podatkov govorimo (na tej osnovi se določa kompleksnost zasnove, stopnja agregacije podatkov, količina dimenzij, strojna zmogljivost itd.).

Podatki izvornih sistemov le nevtrarno dokazujejo določena dejstva. S poslovnim obveščanjem je potrebno izluščiti ključne, problemsko usmerjene informacije, ki nudijo podporo za ukrepanje in odločanje.

### 1.1.3 Pregled poslovnega obveščanja in koncepta OLAP skozi čas

Ena prvih definicij poslovnega obveščanja sega že v leto 1958, ko je Hans Peter Luhn dejal, da je poslovna inteligenca možnost predvidevanja medsebojnih dejstev in akcij na tak način, da nas

peljejo k usmerjenemu cilju [21]. Z drugimi besedami, poslovno obveščanje je podatkovno usmerjen DSS.



Slika 1: Razvoj skozi čas [7]

Poslovno obveščanje v informacijski tehnologiji že dolga leta asociira znana kratica OLAP. Definicij o samem OLAP-u je veliko, gre pa pravzaprav za enega prvih konceptov rešitve, ki je s stopnjo razvoja leta 1993 kazal pot do analitičnih rezultatov. Še danes nudi temeljno podlago sistemov za poslovno obveščanje.

Jedro OLAP sistema je OLAP kocka (večdimenzionalna kocka ali hiperkocka), ki jo sestavlja veliko število numeričnih vrednosti in njihovih agregatov v posameznih presečiščih. Osnovni metapodatki kocke se ustvarijo iz znane zvezdne oblike relacijskih tabel v podatkovni bazi. Vrednosti za analizo se nahajajo v sredinski tabeli dejstev, dimenzije pa v stranskih tabelah. OLAP poizvedba naj bi vrnila odgovor v enem odstotku časa, ki bi ga za to potrebovala poizvedba na OLTP bazi. Glavna prednost se nahaja v predprocesiranih agregacijah.

Leta 1993 je znani E. F. Codd v svojem prispevku objavil 12 lastnosti, katerim mora programska rešitev zadoščati, da jo lahko uvrstimo med OLAP produkte. Pomembnejše, ki držijo še danes in so osrednjega pomena za OLAP sistem so [2]:

- generična dimenzionalnost,
- intuitivna manipulacija s podatki,
- prilagodljiva možnost poročanja,
- neomejenost dimenzij in stopenj agregacij.

OLAP podatkovne baze imajo večdimenzionalni podatkovni model, ki omogočajo odjemalcu analize v realnem času. Da je to izvedljivo, so potrebni predračunani agregati. Kadar govorimo o

agregatih, mislimo na vrednosti, ki so formirane ob pogoju dane dimenzije ali množice dimenzij. Najpogosteje se pri agregaciji uporablja funkcija seštevanja [11].

## 1.2 ARHITEKTURE OLAP

### 1.2.1 Definicija OLAP

**OLAP** sistemi omogočajo uporabnikom hitro in preprosto izluščiti informacije iz obstoječih podatkov, ki se nahajajo v podatkovnem skladišču namenjenemu za analize. Podatki so predstavljeni s pomočjo dimenzij, mer, hierarhij in kock [11]. Različne arhitekture se ločijo po načinu shranjevanja vhodnih podatkov ter njihovih agregatov [11],[10].

### 1.2.2 Arhitekture

**MOLAP** je večdimenzionalni OLAP, ki shranjuje svoje vhodne podatke in agregate v večdimenzionalni kocki. Oba sklopa podatkov sta shranjena na način večdimenzionalne strukture, kar ima za posledico zelo dobro odzivnost. Bistvena pomanjkljivost se pojavi, kadar želimo analizirati večje količine podatkov in imamo definiranih veliko dimenzij. Pride do zelo obsežne hiperkocke, ki ima v svojih presečiščih veliko število praznih vrednosti. Zavedati se je potrebno, da linearno večanje nepreverljivih dimenzij pomeni eksponentno naraščanje velikosti kocke.

**ROLAP** je relacijski OLAP sistem, pri katerem se vhodni podatki in njihovi agregati, shranjujejo v relacijski podatkovni bazi. Podatki se nahajajo v zvezdasti shemi podatkovnega skladišča, agregati pa v materializiranih pogledih izvornih podatkov. Prav zato je ROLAP sistem vedno ažuren in obvladuje veliko večje količine podatkov. Problem je odzivnost tega sistema, ki je v povprečju veliko počasnejša od MOLAP.

**HOLAP** oziroma hibridni OLAP že v imenu pove, da uporablja kombinacijo obeh prejšnjih pristopov z željo po redukciji njunih pomanjkljivosti. Izvirne podatke shranjuje relacijsko, agregate pa v večdimenzionalni kocki tako kot MOLAP. Za končne uporabnike to pomeni dobro odzivnost na ažurnih podatkih.

### 1.2.3 Podatkovno skladiščenje

Izdelava podatkovnega skladišča je pri OLAP-u neizogibna. To še posebno velja, kadar so viri podatkov razdrobljeni po različnih sistemih, do katerih nimamo OLE DB ali ODBC dostopa. Podatki se nahajajo zgolj v tekstovnih datotekah in do izvornega sistema nimamo fizične povezave. Najpogostejši vzrok za skladiščenje so tudi »umazani«  
podatki. To so podatki, ki vsebujejo veliko vsebinskih in tehničnih napak, so nekonsistentni ali podvojeni. Proces čiščenja in pretvorbe podatkov je nemalokrat zelo dolgotrajen, kompleksen in drag proces. Podatkovno strukturo izvornih sistemov je potrebno pretvoriti v skladiščeno zvezdno strukturo.

### 1.2.4 Glavne pomanjkljivosti sistemov OLAP

Razvoj novih aplikacij je zgolj izbiranje med vnaprej pripravljenimi pogledi. Poslovna pravila se vse pogosteje spreminjajo in zahteve po vedno podrobnejših analizah so neizogibne. Potrebno je dodajati nove dimenzije, za katere se potreba pokaže šele med analizo. Nato je potrebno ponovno definiranje strukture OLAP kock, ponovno testiranje itd.

Prelom med končnim uporabnikom in razvijalcem je velik. Dobro je, če je poslovni analitik istočasno tudi razvijalec ali so-razvijalec sistema, kar je v praksi pogosteje izjema kot pravilo. Za pravilne izbire dimenzij in agregatov je potrebno dobro poznavanje izvornega sistema, procese poslovanja in obrazložitev nastajajočih napak. Na vse te zahtevne parametre razvijalec ne more vplivati, na drugi strani pa analitik na tehnično izvedbo ne. Slednjega zanimajo le končni rezultati. Slabo sodelovanje pripelje do netočnih rezultatov, končni uporabnik pa lahko izgubi zaupanje v pridobljene informacije.

### **1.2.5 Pohitritve z analizo v spominu**

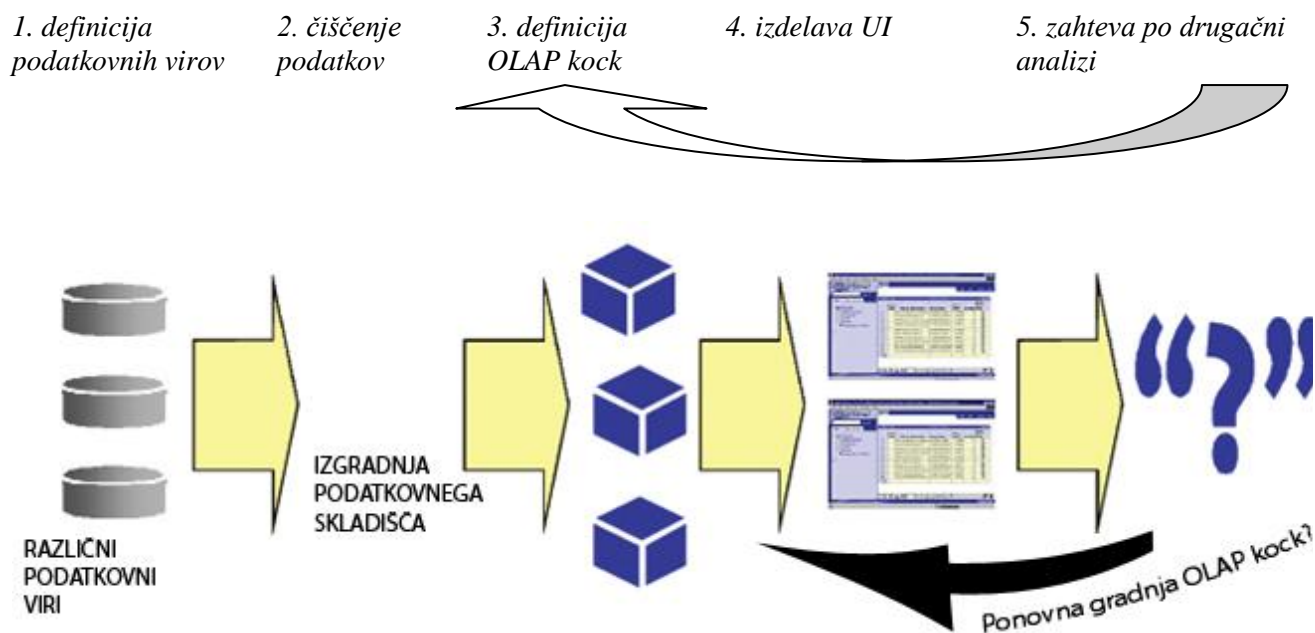
Nekaj izboljšav na opisane pomanjkljivosti najdemo v tehnološko naprednejši zasnovi RTOLAP oz. IMOLAP [10], [17]. Osnovna razlika RTOLAP od tradicionalnega je, da za shranjevanje podatkov uporablja računalniški spomin. Agregirane vrednosti se ne izračunavajo več vnaprej, temveč šele takrat, ko uporabnik to zahteva. S tem se prihrani tudi veliko prostora, ki ga ne potrebujemo več zaradi agregacij, ki morda nikoli ne bodo uporabljene. Posodobljeni izvorni podatki se poznajo pri analizi takoj, ko se osvežijo. Glavna pomanjkljivost pa je, da je celotni analitični podatkovni model prenesen v spomin, kar pomeni, da smo omejeni na njegovo velikost. Pri izredno veliki količini podatkov se lahko poizvedba izvaja mnogo dlje, ker je agregate potrebno izračunavati sproti in jih ne moremo brati iz predizračunanih.

## 2 ORODJE QLIKVIEW

### 2.1 KLASIČEN KONCEPT (OLAP)

Tradicionalni OLAP in kasneje RTOLAP vodi do končne rešitve nepogrešljivih pet korakov. Najdražji izpostavljen problem je izgradnja podatkovnega skladišča. Primarna zahteva po njem se nahaja predvsem v ideji, da za analizo pripravimo optimalno podmnožico celotnega nabora podatkov izvornega informacijskega sistema. Analize ne smejo bremeniti kritičnih transakcijskih sistemov. Spodnja slika 2 simbolično orisuje klasičen pristop, ki ga opredelimo na pet korakov [3]:

1. definicija različnih podatkovnih virov,
2. čiščenje podatkov, odpravljanje nekonsistentnosti in podvojenih vnosov,
3. izgradnja OLAP kock, večdimenzionalnih tabel in predizračunanih agregatov,
4. razvoj uporabniškega vmesnika, ki bo končnim uporabnikom omogočal izvajanje potrebnih analiz,
5. pojavijo se nove zahteve za analizo iz podatkov, ki jih OLAP kocka nima definirane.



Slika 2: Pet razvojnih korakov klasičnega koncepta

Prvi koraki so časovno najbolj potratni. V povprečju traja razvoj podatkovnega skladišča tudi od enega do več mesecev. Seveda je tu največ odvisno od kvalitete in kvantitete izvornih podatkov.

## 2.2 INOVATIVEN KONCEPT (QLIKVIEW)

Za razliko od večine orodij za poslovno obveščanje, QlikView ponuja kar nekaj zelo prodornih in inovativnih odgovorov na probleme tradicionalnih OLAP sistemov [3]. Proces razvoja vsebuje naslednje korake:

1. definicija podatkovnih virov

2. čiščenje, transformiranje in polnjenje podatkov v AQL

3. analiza na zahtevo



Slika 3: Trije razvojni koraki inovativnega koncepta

Zgoraj na sliki 3 opazimo dva manjkajoča koraka v primerjavi s klasičnim konceptom, ki sta vedno omenjena kot najdražja in najbolj zapletena. To pomeni, da lahko podatke namesto iz podatkovnega skladišča prebiramo kar neposredno iz vira. Nanj se lahko povežemo neposredno preko ODBC ali OLE DB protokola, kar pa tudi ni edini možni način prenosa podatkov. Orodje zna uvoziti tudi znane strukture podatkov kot so XLS, XML, CSV in HTML datoteke. Uvožene podatke orodje v času izvajanja analiz hrani neposredno v računalniškem spominu. Ko prenehamo z analiziranjem, lahko celoten dokument z izvornimi in drugimi meta podatki shranimo na trdi disk v obliki QlikView datoteke (QVW).

QlikView uporablja drugačno podatkovno strukturo, kot jo ima podatkovno skladišče. Orodje temelji na patentiranem AQL™ pristopu podatkovnega modela, ki omogoča procesiranje podatkov na 32 in 64 bitnih platformah. Celoten nabor izvornih podatkov različnih virov prenese v računalniški spomin, vrednost posameznih dimenzij pa izračunava na uporabnikovo zahtevo [10], [14].

## 2.3 ASOCIATIVNI PODATKOVNI MODEL (AQL)

OLAP hiperkocke imajo veliko pomanjkljivost, da končne uporabnike omejujejo na majhen nabor dimenzij. Te morajo biti definirane še preden je aplikacija gotova. Njihovo težo pri analizi je praktično nemogoče oceniti že v razvoju. QlikView patentirana tehnologija AQL deluje na povsem drugačni zasnovi. Polnimo in vzdržujemo ne relacijsko temveč asociativno podatkovno bazo. Prostorsko je zelo optimalno zasnovana in se med analiziranjem zadržuje v računalniškem spominu (RAM).

Največja prednost AQL arhitekture je, da so izvorni podatki shranjeni pri uporabniku in na voljo tudi za analize brez povezave. Uporabnik ni več vnaprej omejen z njihovo globino in lahko vrta vse do zapisov na transakcijskem nivoju.

Posamezen zapis v tabeli predstavlja osnovni element v relacijskem in objektnem pristopu. Oba vsebujeta attribute o posameznih stvareh, npr. za osebo so atributi njena starost, spol, višina, teža itd.

Asociativni podatkovni model se tu razlikuje od klasičnega modela podatkovne baze. Pri njem podatke ločimo le na posamezne entitete ali predmete, ki jih nato medsebojno povezujemo v asociacije [20].

- **Entitete oz. predmeti** imajo svoj diskreten, neodvisen obstoj. Z njimi vplivamo na druge entitete. Tabela entitet je v AQL sestavljena iz enoličnega identifikatorja in imena.

*Primer 1:*

Identifikator	Ime
22	Janez Novak
25	Zaposlenec
77	Tovarna

*Tabela 1: Tabela entitet*

- **Asociacije oz. povezave** katerih obstoj je odvisen od ene ali več drugih povezav oz. entitet. Povezava, ki ne vpliva na neko entiteto ali drugo povezavo ni asociacija. Asociacije so sestavljene iz lastnih enoličnih identifikatorjev, vira, kopule (angl. copula, verb) in cilja.

*Primer 2:*

Identifikator	Vir	Kopula	Cilj
734	22	25	77

*Tabela 2: Tabela asociacij*

*Primer 3:*

Predpostavimo, da želimo analizirati opravljene izpite v naši podatkovni bazi. V asociacijsko podatkovno bazo hočemo shraniti informacijo: *Študent št. 2233 je opravil izpit ARS dne 22.6.2005 ob 11:20*. Poved je sestavljena iz štirih samostalnikov (Študent št. 2233, izpit ARS, 22.6.2008,11:20) in treh kopul (je opravil, dne, ob). Če jo želimo shraniti v AQL podatkovni obliki potrebujemo tri povezave. Ker lahko v vrstico tabele asociacij shranimo le

eno asociacijo hkrati, si zgornjo poved razbijemo na tri dele. Potrebujemo le dve tabeli. Tabela entitet in tabela asociacij.

**Študent št. 2233** je opravil izpit **ARS**  
 ...dne **22.6.2005**  
 ...ob **11:20**

Enolični identifikator	Ime
33	Študent št. 2233
13	izpit ARS
14	22.6.2008
54	11:20
68	je opravil
44	dne
76	ob

*Tabela 3: Entitete*

Identifikator	Vir	Kopula	Cilj
101	33	68	13
102	101	44	14
103	102	76	54

*Tabela 4: Asociacije*

Asociacije ne smemo zamenjevati s klasičnimi relacijami. Asociacija je lahko vir, kopula ali cilj, ki vsebuje identifikator entitete ali druge asociacije. Enako tudi stolpec 'Ime' v prvi tabeli lahko vsebuje katerikoli podatkovni tip. Takšen način povezovanja v relacijskem podatkovnem modelu ni dopusten [21].

#### *Primer 4:*

Če bi želeli zgornji primer še nekoliko dopolniti in zapisati še nekaj dodatnih stavkov, bi relacijski zapisi izgledali približno tako, kot je prikazano na sliki 4.

StudentiID	Izpit	Ocena	Datum	Ura
2233	2L_ARS	8	22.6.2005	11:20
2233	3L_IS	9	22.7.2006	13:20
2233	4L_PVIS	7	13.2.2007	14:00
6644	1L_ANA	7	1.2.2004	10:00
6644	2L_ARS	6	30.6.2004	11:00

*Slika 4: Zapisani dogodki v relacijski obliki*

Opazimo, da sta dva študenta opravila izpit '2L\_ARS', vendar v različnem obdobju. Zapis je v relacijskem modelu zapisan dvakrat, ker imamo dva različna dogodki. Pri asociacijskem ga zabeležimo le enkrat. Drugi pojav le še asociira na '2L\_ARS'. Z orodjem lahko zgornjo relacijsko tabelo na asociativen način predstavimo takole:

StudentID	Izpit	Ocena	Datum	Ura
2233	1L_ANA	6	1.2.2004	10:00
6644	2L_ARS	7	30.6.2004	11:00
	3L_IS	8	22.6.2005	11:20
	4L_PVIS	9	22.7.2006	13:20
			13.2.2007	14:00

Slika 5: Asociacijska oblika zapisa v QlikView

StudentID	Izpit	Ocena	Datum	Ura
2233	2L_ARS	7	22.6.2005	11:20
6644	3L_IS	8	22.7.2006	13:20
	4L_PVIS	9	13.2.2007	14:00
	1L_ANA	6	1.2.2004	10:00
			30.6.2004	11:00

Slika 6: Reakcija ob izbiri študenta 2233

Na sliki 6 vidimo izbrano vrednost '2233' v polju 'StudentID'. Vse entitete, ki so v kakršnikoli povezavi s študentom '2233' se obarvajo belo, neasociirane vrednosti pa s sivo barvo. V relacijskem modelu na sliki 4 je številka '2233' zapisana trikrat, pri AQL strukturi le enkrat. Podvojene vrednosti so eliminirane, med njimi so le asociacije. AQL struktura hranjenja podatkov zasede v povprečju le 15 do 20 odstotkov količine, ki jo predstavljajo izvorni podatki v relacijskem zapisu. To pomeni, da imamo lahko celotni podatkovni model shranjen znotraj računalniškega spomina (RAM). V večini primerov zadostuje že količina, ki ga imajo danes delovne postaje vključno s prenosnimi računalniki [3].

#### Primer 5:

name	rows	reserved	data	index_size	unused
RP_PROMET	56778564	203961192 KB	904867200 KB	11322304 KB	25216 KB

Slika 7: Statistični prikaz tabele RP\_PROMET v MS SQL podatkovni bazi.

Class	Type	SubType	Id	KB	Rows
Database	Table	Records	Promet	1 275 300	56 778 564
Database	Field	Symbols	POGODBA_ID	106 530	4 907 776
Database	Field	Symbols	PROMET ID	23 522	2 676 244

Slika 8: Statistični prikaz tabele RP\_PROMET v AQL podatkovni bazi.

Pri sliki 7 je dobljeni rezultat z ukazom 'sp\_spaceused' za relacijsko tabelo 'RP\_PROMET':

- število vrstic: 56 778 564
- skupaj podatki v pomnilniku: 9 048 672 KB ~ 8836 MB

Na sliki 8 imamo statistične podatke za logično tabelo RP\_PROMET znotraj AQL podatkovnega modela:

- število vrstic: 56 778 564
- skupaj podatki v pomnilniku: 1 275 300 KB ~ 1245 MB

Primerjava:  $1245 \text{ MB} / 8836 \text{ MB} = 0,14 = 14 \%$

## 2.4 QLIKVIEW ETL SKRIPT

Pri AQL razdelku smo ugotovili, da podatkovno skladiščenje, gradnja OLAP kock in predizračunavanje agregatov niso več nujen predpogoj za izvajanje analiz. Ključni del izgradnje vsakega sistema za poslovno obveščanje je ETL (angl. Extract Transform Load) postopek. Ta proces lahko izvajamo neposredno preko QlikView skripta. Osnovna sintaksa skripta temelji na poznanem jeziku SQL (angl. Structured Query Language) z nadgradnjo več kot 200 funkcij za manipulacijo, povezovanje in čiščenje izvornih podatkov. Enako kot pri podatkovnem skladišču lahko črpamo podatke iz različnih podatkovnih baz ali datotek. Lahko jih pridobimo tudi neposredno preko spleta (XML in HTML datoteke).

### 2.4.1 Osnovni gradniki

#### Polja

Polja so primarna entiteta orodja QlikView. Celotni podatkovni model AQL operira s polji in njihovimi vrednostmi. Polja napolnimo v skript z ukazi LOAD, SELECT in BINARY. Edina možnost za spreminjanje vrednosti polj je, da skript ponovno izvršimo. Ne morejo jih spreminjati končni uporabniki. Ko so prebrani se lahko uporabljajo za logične izbire in kalkulacije. Polja lahko vsebujejo numerične in alfanumerične vrednosti.

#### Vhodna polja

Omenili smo, da vrednost polj ne moremo spreminjati. QlikView pa kljub temu omogoča tudi poseben tip polj z enakimi lastnostmi kot standardna, le da se njihove vrednosti lahko spremenijo tudi brez ponovnega izvajanja skripta.

#### Spremenljivke

Spremenljivke so poimenske entitete, ki vsebujejo le eno samo vrednost. Vrednosti jim dodelimo s pomočjo LET, SET ali drugih kontrolnih ukazov znotraj skripta. Vsebujejo lahko numerične in alfanumerične vrednosti.

#### Druge entitete

Druge entitete so še:

Dimezije, ki predstavljajo množico podatkov znotraj QlikView objektov (grafikoni, tabele). Objekti lahko vsebujejo poljubno število dimenzij z omejitvijo na količino pomnilnika, kompleksnosti podatkov ter tipom objekta, ki ga prikazujemo.

Skupine polj se lahko uporabljajo namesto dimenzij. Poznamo globinsko, hierarhično (angl. drill-down groups) in ciklično (angl. cyclic groups) ureditev skupin.

Izrazi so se v prehodnih verzijah QlikView-ja (4 in 5) lahko izvajali le znotraj objektov. Kasneje se lahko uporabljajo tudi v skriptu, vendar se njihovo preračunavanje izvaja le med izvajanjem skripta in ne na zahtevo uporabnika. Izrazov je lahko več znotraj istega objekta. Vsi morajo pri svojem

rezultatu upoštevati pogoje predhodno definiranih dimenzij. Vhodni parametri izraza so lahko polja, spremenljivke, operatorji in izrazi sami.

## 2.4.2 Izbrane funkcionalnosti

Na QlikView skript lahko gledamo kot nadgradnjo ali dopolnilo SQL stavkov. Osnovni gradniki so imena polj, spremenljivk, formul, ključnih besed in stavkov. Po namenu lahko ločimo:

### Običajne stavke

Običajni stavki se uporabljajo za manipulacijo podatkov. Lahko jih razvrstimo na različne načine. Zaključujejo se s podpičjem »;« (npr.: `LOAD * FROM Tabela1.xls;`)

### Kontrolni stavke

Kontrolni stavki skrbijo za kontroliran tok podatkov med izvedbo skripta. (npr.: `IF` stavki)

### Predpone

Predpone se pojavljajo pred navadnimi stavki in nikoli pred kontrolnimi stavki (npr. `BUFFER LOAD`).

Razen 'WHEN' in 'UNLESS' veljata za zapone in se uporabljata na koncu stavkov oz. zank.

*Primer:*

QlikView skript	Komentar
1. <i>Laboratorij</i> ;	1. Ime logične tabele
2. <b>LOAD</b>	2. Začetek stavka <code>LOAD</code>
3. <code>SIFRA005 AS LaboratorijSifra,</code>	3.-6. Transformacije nad atributi
4. <code>left(SIFRA005,1) AS LaboratorijSkupinaSifra,</code>	<code>SIFRA005</code> , <code>VREDNOST</code> in <code>OPIS_005</code>
5. <code>if(isnull(VREDNOST), 0, VREDNOST) as Vrednost,</code>	podrejeni tabeli, ki jo za rezultat vrne
6. <code>OPIS_005 AS LaboratorijNaziv</code>	spodnji <code>SQL</code> stavek
7. ;	7. Zaključek stavka <code>LOAD</code>
8. <b>SQL SELECT</b>	8.-12. <code>SQL</code> poizvedba, ki vrne za rezultat
9. <code>SIFRA005,</code>	logično tabelo s tremi atributi iz
10. <code>VREDNOST,</code>	<code>Baze2008.promet</code> s katero smo povezani
11. <code>OPIS_005</code>	preko <code>OLE DB</code> ali <code>ODBC</code>
12. <b>FROM</b> <code>Baze2008.promet;</code>	povezave. Predstavlja vir podatkov za
	zgornji <code>LOAD</code> ukaz.

Tabela 5: Primer polnjenja iz `ODBC` vira tabele 'Baze2008.promet' v `AQL` logično tabelo 'Laboratorij'

QlikView skript omogoča uporabo velikega števila dodatnih funkcij. V nadaljevanju so opisane le izbrane, ki so se v praksi izkazale za najbolj uporabne.

### 2.4.2.1 Predpona ADD

Predpona se lahko doda pred LOAD, SELECT ali MAP stavkom. Svoj pomen doseže le ob parcialnih izvajanjih skripta. Logična tabela, ki se prečrpa s predpono ADD bo le dodana rezultatu LOAD ali SELECT stavka. Zato mora stavek, ki uporablja predpono ADD, imeti enolični identifikator za izločitev podvojenih vnosov.

QlikView skript	Komentar
1. Imenik:	1. Ime tabele
2. <b>LOAD</b> Ime, Stevilka from Persons.csv;	2. Uvoz imenika
3. <b>ADD LOAD</b> Ime, Stevilka from NewPersons.csv	3. Dodajanje le tistih imen in števil, ki še ne
4. <b>WHERE NOT</b> exists(Ime);	obstajajo

Tabela 6: Primer predpone ADD

Med parcialnim izvajanjem skripta se v tem primeru izvedeta le 3. in 4. vrstica. Ob normalni izvedbi se izvedejo vse štiri vrstice, pri tem se podatki iz tabele 'NewPersons.csv' le dodajo na prejšnjo tabelo in ob tem preverijo, če identifikator 'Ime' že obstaja.

### 2.4.2.2 Ukaz BINARY

BINARY stavek se uporablja za črpanje podatkov iz QVW dokumenta. Pri tem prečrpa samo logične tabele, njihovo strukturo in vsebino. Znotraj skripta je dovoljena le enkratna uporaba tega stavka in mora biti prvi stavek v celotnem skriptu. Zelo uporaben ukaz, ki dopušča ponovno rabo podatkov, ki so že preneseni iz izvornega sistema.

### 2.4.2.3 Predpona BUFFER

Predpona BUFFER avtomatsko generira začasne datoteke. Uporablja se le pred LOAD in SELECT stavki. Ukaz začasno shrani pridobljeno QVD datoteko na trdi disk. Ima posebno 160-bitno šestnajstiško ime, da lahko orodje natančno določi kateri aplikaciji pripada. Po privzetih nastavitvah se datoteke samodejno tudi brišejo iz trdega diska, ko se izvajanje skripta konča ali izvorni dokument ne obstaja več.

### 2.4.2.4 Kontrolna stavka SUB in CALL

Stavek SUB definira proceduro, katero priključimo z ukazom CALL.

QlikView skript	Komentar
1. <b>SUB</b> INCR (I,J)	1. Ustvarimo ime procedure in vhodne parametre
2. I=I+1	2. Povečamo vrednost spremenljivke I za 1
3. J=J+1	3. Povečamo vrednost spremenljivke J za 1
4. <b>END SUB</b>	4. Konec procedure
5. <b>CALL</b> INCR (X,Y)	5. Kličemo proceduro s parametroma X in Y

Tabela 7: Primer procedure

### 2.4.2.5 Predpona CROSSTABLE

Velikokrat se srečamo s problemom, ko ima vir podatkov strukturo vrtilne tabele kot na sliki 9. Za pretvorbo podatkov na relacijsko strukturo, uporabimo ukaz CROSSTABLE:

Prodajalec	Leto	Jan	Feb	Mar	Apr	Maj
A	2000	34	45	32	22	35
A	2001	46	44	75	33	23
A	2002	75	43	43	34	39
B	2000	45	23	75	98	78
B	2001	77	32	43	65	76
B	2002	55	32	21	75	66

Slika 9: Izvorna struktura podatkov

Prodajalec	Leto	Mesec	Plan
A	2000	Apr	22
A	2000	Feb	45
A	2000	Jan	34
A	2000	Maj	35
A	2000	Mar	32
A	2001	Apr	33
A	2001	Feb	44
A	2001	Jan	46
A	2001	Maj	23
A	2001	Mar	75
A	2002	Apr	34
A	2002	Feb	43
A	2002	Jan	75
A	2002	Maj	39
A	2002	Mar	43
B	2000	Apr	98

Slika 10: Ciljna struktura podatkov

### 2.4.2.6 Stavka DROP FIELD in DROP TABLE

Stavka sta dva pomembna ukaza, ki se pogosteje uporabljata v zadnjih korakih skripta. Lahko brišemo posamezna polja ali kar celotne tabele, ki smo jih med ETL postopkom potrebovali v spominu le začasno.

### 2.4.2.7 Kontrolna stavka FOR...NEXT in FOR...EACH

FOR...NEXT stavek poznamo kot zanko, ki ima svoj števec. Znotraj zanke se vsak stavek izvede natanko tolikokrat, kolikor je vrednost števca. Stavek se zaključí z besedo NEXT.

FOR EACH...NEXT je naprednejši FOR stavek, ki namesto števca uporablja spremenljivko, katerih različne vrednosti so podane z podpičjem. To je zelo uporabno, če želimo prebirati npr. datoteke znotraj iste mape in njenih podmap.

Uporaba FOR..NEXT stavka	Uporaba FOR EACH..NEXT stavka
<pre> <b>FOR</b> A=1 to 4   <b>LOAD</b> *   <b>FROM</b> file\$(a).csv; <b>NEXT</b> </pre>	<pre> <b>SUB</b> DoDir (Root)   <b>FOR EACH</b> Ext in 'qvw','qva','qvo','qvs'     <b>FOR EACH</b> File in filelist (Root&amp;'\'&amp;Ext)       <b>LOAD</b> '\$(File)' as Name,         <i>FileSize('\$(File)')</i> as Size,       <b>AUTOGENERATE</b> 1;     <b>NEXT</b> File   <b>NEXT</b> Ext   <b>FOR EACH</b> Dir in dirlist (Root&amp;'\'')     <b>CALL</b> DoDir (Dir)   <b>NEXT</b> Dir <b>END SUB</b> <b>CALL</b> DoDir ('C:') </pre>

Tabela 8: Primeri uporabe zank FOR

Desna zanka v tabeli 8 je uporaben način kontrolnega stavka FOR EACH..NEXT in SUB procedure. V zadnji vrstici se potrebuje za vhodni parameter korenska mapa. Nato znotraj nje iščemo vse datoteke tipa 'qvw', 'qva', 'qvo' ali 'qvs'. Rezultat je logična tabela z imenom datoteke in njeno velikostjo.

#### 2.4.2.8 Predpona MAPPING

Za čiščenje podatkov se uporablja predpona MAPPING. Tabele, ki jih prečrpamo kot »mapping« tabele se obravnavajo popolnoma drugače kot ostale. Shranijo se v poseben prostor v spominu in se uporabljajo ekskluzivno za namen mapiranja. Ko se skript v celoti izvede, se »mapping« tabele izbrišejo. »Mapping« tabela mora vsebovati natanko dva stolpca. Imena stolpcev nimajo nobene povezave z imeni polj v podatkovnem modelu. Te tabele se uporabljajo z namenom, da zamenjajo določene vrednosti.

#### 2.4.2.9 Predpona HIERARCHY in HIERARCHY BELONGS TO

Zelo pomemben ukaz, ki zna iz tabele sosednjih vozlišč (angl. adjacent nodes table) generirati hierarhično strukturo v AQL. Predpostavlja se, da je drevesna struktura pravilna. Če želimo obdržati integriteto in se prepričati, da bo ukaz pravilno deloval, moramo biti pozorni, da [1]:

- drevo nima ciklov
- in obstaja samo eno glavno korensko vozlišče.

Običajno ima vhodna tabela natanko en zapis za vsako vozlišče. V tem primeru je tudi velikost izhodne tabele enaka. Če ima vozlišče več staršev, bo v izhodni tabeli toliko več zapisov. Vsa vozlišča, ki nimajo nadrejenega, se upoštevajo kot korenska vozlišča. Znotraj ukaza je potrebno definirati tri pomembna polja:

- NodeID - identifikator vsakega vozlišča,
- NodeName - ime vozlišča,
- ParentID - identifikator nadrejenega vozlišča.

S hierarhičnimi strukturami se ponazarjajo predvsem geografske ali organizacijske dimenzije. Torej je vsak zapis vozlišče, ki ima lahko več podrejenih vozlišč in natanko enega nadrejenega.

NodeID	ParentID	NodeName
1	-	The World
2	1	Europe
3	2	France
4	3	Bordeaux
5	4	Medoc
6	5	Bas-Médoc
7	5	Haut-Médoc
8	4	Graves
9	3	Bourgogne
10	2	Germany
11	10	Rheingau
12	1	Americas
13	12	California
14	13	Napa valley

Slika 11: Prikaz tabele sosednjih vozlišč(vhodna tabela)

NodeID	ParentID	NodeName	Level0	Level1	Level2	Level3	Level4	Level5
1	-	The World	The World	-	-	-	-	-
2	1	Europe	The World	Europe	-	-	-	-
3	2	France	The World	Europe	France	-	-	-
4	3	Bordeaux	The World	Europe	France	Bordeaux	-	-
5	4	Medoc	The World	Europe	France	Bordeaux	Medoc	-
6	5	Bas-Médoc	The World	Europe	France	Bordeaux	Medoc	Bas-Médoc
7	5	Haut-Médoc	The World	Europe	France	Bordeaux	Medoc	Haut-Médoc
8	4	Graves	The World	Europe	France	Bordeaux	Graves	-
9	3	Bourgogne	The World	Europe	France	Bourgogne	-	-
10	2	Germany	The World	Europe	Germany	-	-	-
11	10	Rheingau	The World	Europe	Germany	Rheingau	-	-
12	1	Americas	The World	Americas	-	-	-	-
13	12	California	The World	Americas	California	-	-	-
14	13	Napa valley	The World	Americas	California	Napa valley	-	-

Slika 12: Prikaz razširjene tabele(rezultat)

Na sliki 12 je prikazana razširjena tabela (angl. expanded nodes table), kjer je vsak nivo v hierarhiji zapisan v svojem stolpcu. S to obliko veliko lažje prikažemo vrtljivo tabelo. Problem razširjene tabele je, da je ni možno preprosto uporabiti pri iskanju. Informacijo o nadrejenem ali podrejenem vozlišču imamo le za en nivo. Zato potrebujemo še dodatno premostitveno logično tabelo (angl. bridge table), ki reši ta problem. Na prvi vtis je zelo podobna tabeli sosednjih vozlišč vendar z dopolnilom, da ima vsako vozlišče zapisana vsa svoja nadrejena vozlišča. Premostitveno tabelo generiramo s predpono HIERARCHY BELONGS TO.

NodeID	NodeName	AncestorID	AncestorName
1	The World	1	The World
2	Europe	1	The World
2	Europe	2	Europe
3	France	1	The World
3	France	2	Europe
3	France	3	France
4	Bordeaux	1	The World
4	Bordeaux	2	Europe
4	Bordeaux	3	France
4	Bordeaux	4	Bordeaux
5	Medoc	1	The World
5	Medoc	2	Europe
5	Medoc	3	France
5	Medoc	4	Bordeaux

Slika 13: Prikaz premostitvene tabele



Slika 14: Drevesna struktura

#### 2.4.2.10 Predpona INTERVALMATCH

Predpona INTERVALMATCH je zelo zmogljiv ukaz v primerih, kadar je potrebno zaradi analize povezovati med seboj numerične intervale. Rezultat LOAD ali SELECT ukaza mora biti dvo stolpčna tabela, kjer prvo polje vsebuje spodnjo mejo in drugo zgornjo mejo intervala. Intervali morajo biti vedno zaprti. Nenumerične vrednosti se ignorirajo. Upoštevati zna tudi prekrivanja intervalov.

Primer:

Zacetek	Konec	Narocilo
1:00	3:35	A
2:30	7:58	B
3:04	10:27	C
7:23	11:43	D

Tabela 9: Naročila

Cas	Dogodek	Komentar
0:00	0	Začetek prve izmene
1:18	1	Prevoz naročila
2:23	2	Predelava
4:15	3	Pakiranje
8:00	4	Začetek druge izmene
11:43	5	Konec

Tabela 10 : Opisi dogodkov

QlikView skript
<pre>LOAD * FROM Narocila; LOAD * FROM Dogodki; INTERVALMATCH (Cas) LOAD Zacetek, Konec FROM Narocila;</pre>

Tabela 11 : Izvršeni ukazi

Cas	Dogodek	Komentar	Narocilo	Zacetek	Konec
0:00	0	Začetek prve izmene	-	-	-
1:18	1	Prevoz naročila	A	1:00	3:35
2:23	2	Predelava	A	1:00	3:35
4:15	3	Pakiranje	B	2:30	7:58
4:15	3	Pakiranje	C	3:04	10:27
8:00	4	Začetek druge izmene	C	3:04	10:27
8:00	4	Začetek druge izmene	D	7:23	11:43
11:43	5	Konec	D	7:23	11:43

Slika 15: Rezultat

V tabeli 9 so zapisana naročila, njihov začetni trenutek in konec, ko je naročilo obdelano. V tabeli 10 imamo zapise posameznih dogodkov. Z izvedbo ukaza INTERVAL MATCH dobimo rezultat na sliki 15. Naše dogodke smo časovno opredelili in jih povezali s posameznimi naročili.

### 2.4.3 Izrazi

Uporabljajo se znotraj LOAD ali SELECT stavka. Sintaktična pravila se razlikujejo glede na izvršitelja stavkov. LOAD stavek bazira na QlikView sintaksi, kar pa ne velja za SELECT stavek. Slednji se nadalje interpretira preko ODBC ali OLE DB vmesnika. Sintaktična pravila so si med seboj zelo podobna, vendar niso enaka. Rezultati izrazov znotraj skripta so numerični, nizni ali logični (0 za vrednost FALSE in -1 za TRUE). Izrazi se lahko poljubno gnezdiijo. Dokler rezultat vrne primer sprejemljive vrednosti, QlikView ne bo vračal obvestil za napake.

#### 2.4.3.1 Operatorji

Skriptni jezik ima naslednjo množico operatorjev:

- numerične (+, -, \*, /),
- nizne (&, like),
- logične (not, and, or, xor),
- relacijske (<, <=, =, >, >=, <>),
- bitne (bitnot, bitand, bitor, bitxor, >>, >>).

### 2.4.3.2 Agregacijske funkcije

Pomembno pri ETL procesu je stopnja agregacije. Nabor agregacijskih funkcij omogoča izračunavanja že med sestavljanjem podatkovnega modela. Uporabljajo se znotraj LOAD stavka, ki se zaključí z besedo GROUP BY.

- **Osnovne agregacijske funkcije:** sum(), min(), only(), mode(), firstsortedvalue()...
- **Nizne agregacijske funkcije:** minString(), maxstring()...
- **Števne agregacijske funkcije:** count(), numericcount(), textcount()...
- **Statistične agregacijske funkcije:** avg(), stdev(), fractile(), median()...
- **Finančne agregacijske funkcije:** irr(),xirr(), npv(),xnpv()...
- **Ostale funkcije:** splošne numerične, trigonometrijske, distribucijske...

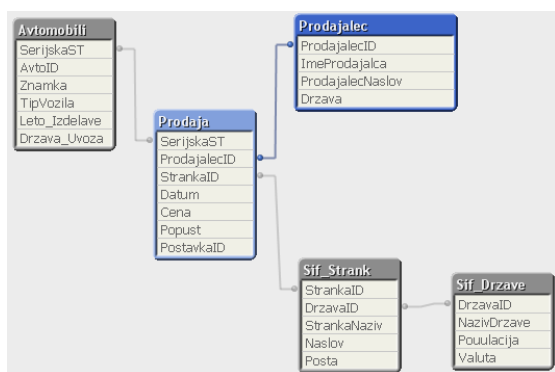
## 2.5 PODATKOVNE STRUKTURE

### 2.5.1 Logične tabele

Vsaka tabela v AQL modelu je t.i. logična tabela. Uporabnik jih lahko transformira in preračunava z LOAD ali SELECT stavkoma. Oba generirata vhodno tabelo predstavljeno kot seznam. QlikView ne loči med tabelami uvoženimi z LOAD ali SELECT stavkom.

### 2.5.2 Asociacije med logičnimi tabelami

Podatkovni model tvorijo logične tabele. Vsaka od njih ima svoje attribute. V kolikor so enako poimenovane, QlikView privzame asociacijske povezave med njimi. Asociacijska povezava naredi enak učinek kot SQL OUTER JOIN, le da se izvrši šele na uporabnikovo zahtevo.



Slika 16: Enostaven podatkovni model

Field	\$Table	Sif_Drzave	Sif_Strank	Prodaja	Avtomobil	Prodajalec
DrzavaID	-	DrzavaID	-	-	-	DrzavaID
StrankaID	-	-	StrankaID	-	-	-
SerijskaST	-	-	-	SerijskaST	SerijskaST	-
ProdajalecID	-	-	-	ProdajalecID	-	ProdajalecID
NazivDrzave	NazivDrzave	-	-	-	-	-
Pouulacija	Pouulacija	-	-	-	-	-
Valuta	Valuta	-	-	-	-	-
StrankaNaziv	-	StrankaNaziv	-	-	-	-
Naslov	-	Naslov	-	-	-	-
Posta	-	Posta	-	-	-	-
Datum	-	-	Datum	-	-	-
Cena	-	-	Cena	-	-	-
Popust	-	-	Popust	-	-	-
PostavkaID	-	-	PostavkaID	-	-	-
AvtoID	-	-	-	AvtoID	-	-
Znamka	-	-	-	Znamka	-	-
TipVozila	-	-	-	TipVozila	-	-
Leto_Izdelave	-	-	-	Leto_Izdelave	-	-
Drzava_Uvoza	-	-	-	Drzava_Uvoza	-	-
ImeProdajalca	-	-	-	-	-	ImeProdajalca
ProdajalecNas...	-	-	-	-	-	ProdajalecNas...

Slika 17: Sistemska matrika asociacij

QlikView omogoča tudi uporabo JOIN ukazov znotraj skripta. Rezultati spajanj pomenijo manjše število logičnih tabel, kar vodi v boljše odzivne čase pri velikemu številu zapisov. Glavna pomanjkljivost združevanj je lahko izguba določenih informacij ter povečana širina tabel. Posledično to pomeni večjo prostorsko zasedenost pomnilnika. Pomembno je vedeti, da bodo JOIN stavki znotraj SQL SELECT stavkov procesirani na strani podatkovne baze in ne QlikView-ja.

#### 2.5.2.1 Pomanjkljivosti samodejnega asociiranja

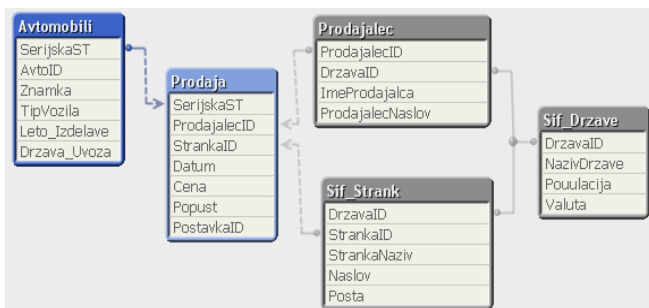
Glavni pomanjkljivosti samodejnega asociiranja sta:

##### Tabele hkratnih dogodkov (angl. sync tables)

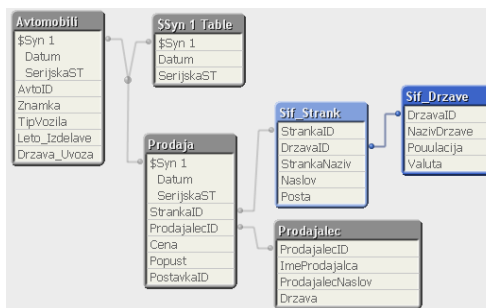
Če imata dve logični tabeli več kot dve polji enako poimenovani, se samodejno generira sintetični ključ, ki se nahaja v tabeli hkratnih dogodkov. V modelu se pojavi dodatna tabela z predpono '\$Syn' v nazivu. Učinek je močnejši sestavljeni ključ, ki se izračuna po potrebi. Problem nastane, če je število sintetičnih ključev preveliko, še posebej v primeru velikih količin podatkov postane izračunavanje zelo zahtevno za CPU. Boljša alternativa je izgradnja novega, sestavljenega ključa z uporabo funkcije Autonumber( ).

### Krožne reference (angl. loops)

Krožna referenca znotraj podatkovne strukture pomeni, da obstajata dve možni poti za asociacijo med dvema atributoma. Pride do dvoumnih interpretacij, kar močno vpliva na stabilnost aplikacije in točnost podatkov.



Slika 18: Krožne reference



Slika 19: Tabela hkratnih dogodkov(\$Syn Table)

### 2.5.3 QVD datoteke

QVD datoteka vsebuje logično tabelo izvoženo iz QlikView-ja. Gre za izviren (angl. native) format, ki ga lahko ustvari in bere le orodje samo. Zelo optimiziran za branje podatkov znotraj QlikView skripta, ki je tipično od 10 do 100-krat hitreje v primerjavi z branjem drugih podatkovnih virov. Konceptualno je format zapisa zelo podoben splošno znanim CSV ali DIFF formatom ter pri shranjevanju uporablja visoko kompresijo. QVD datoteka ima:

- XML glavo (UTF-8) za opisovanje atributov v tabeli in drugih meta podatkov,
- tabelo simbolov v bajti kompresijski obliki,
- podatkovno tabelo v bitni kompresijski obliki.

#### 2.5.3.1 Prednosti uporabe QVD datotek

- **Pospesitev hitrosti črpanja podatkov**  
Pri razvoju skripta naletimo na problem, da so množice podatkov razmeroma velike. Možna rešitev je, da si podatke priskrbimo na lokalnih bazah. Učinkovitejša alternativa je izvoz podatkovnih tabel v QVD datoteke. Lažji je prenos na razvojno okolje zaradi visoke kompresije in pridobimo hitrejšo berljivost podatkov med razvojem.
- **Razbremenitev izvornih produkcijskih baz med razvojem**  
Velika prednost, ki omogoča neprekinjen razvoj brez postavitve testnega okolja. Obenem pa je hitrost branja med razvojem hitrejša, saj je potrebno postopek pogosto ponavljati.

- **Uporaba podatkov pri več aplikacijah**  
Kadar imamo več aplikacij, ki potrebujejo podatke iz istih tabel izvorne podatkovne baze, jih lahko ob prvi osvežitvi podatkov shranimo v QVD datoteke. Ostale aplikacije lahko berejo podatke iz QVD datotek brez ponovnega dodatnega povezovanja na vir podatkov.
- **Prirastno črpanje (angl. incremental load)**  
Pogosto želimo izvajati prirastno črpanje podatkov, kar pomeni dodajati le nove nastale zapise iz vira. Tako se izognemo ponavljajočemu prenosu vseh možnih zapisov.

## 2.6 VARNOST

Samo avtorizirane osebe smejo gledati določen obseg poročil. QlikView vsebuje dva varnostna nivoja. Prvi je implementiran znotraj skripta, drugi je integriran z Windows okoljem in njegovim datotečnim sistemom. Slednji se uporablja pri večjemu številu uporabnikov med drugim tudi zaradi praktičnosti administracije skupin in vlog znotraj sistema.

Overitev (angl. authentication) uporabnika lahko prepustimo operacijskemu sistemu ali zahtevamo ob vsakem dostopu vnos uporabniškega imena in geslo. Nivo uporabnikov je določen znotraj SECTION ACCESS ukaza. Lahko se identificirajo z uporabniškim imenom in geslom (USERID, PASSWORD) ali s pomočjo Windows uporabniškega imena (NTNAME). Možnosti so še uporaba Windows NT SID številke ali serijska številka licence (NTSID, SERIAL).

*Primer overitve in avtorizacije uporabnikov znotraj skritega skripta (angl. hidden script)*

QlikView skript	Komentar
1. <b>SECTION ACCESS;</b> 2. <b>LOAD * INLINE</b> 3. <b>[ACCESS, NTNAME, ODDELEK</b> 4. <b>ADMIN, Janez,FINANCE,</b> 5. <b>USER, Ana,PLACE</b> 6. <b>USER, Peter,PRODAJA</b> 7. <b>USER, Ben,PRODAJA];</b> 8. 9. <b>SECTION Application;</b> 10. 11. <b>LOAD * INLINE</b> 12. <b>[ODDELEK, StroskovnoMesto</b> 13. <b>FINANCE, *</b> 14. <b>PLACE, 503</b> 15. <b>PRODAJA, 520];</b>	<p><i>Prvih 7 vrstic določa uporabnike na osnovi Windows uporabniškega imena. Vsak uporabnik pripada enem oddelku. Le ADMIN lahko dostopa do varnostnih nastavitev, USER ne.</i></p> <p><i>Spodnji del od 9. vrstice dalje določa vloge posameznih oddelkov. Na našem primeru omejujemo vloge po stroškovnih mestih. Torej oddelek FINANCE lahko gleda podatke vseh Stroškovnih mest, PLACE le stroškovno mesto 503 in PRODAJA Stroškovno mesto 520.</i></p> <p><i>Opomba: Atribut 'StroskovnoMesto' mora obstajati znotraj logične tabele v podatkovnem modelu, saj se podatki reducirajo na njegovi osnovi.</i></p>

*Tabela 12: Overitev in avtorizacija uporabnikov na nivoju aplikacije*

## 2.7 NADZORNE PLOŠČE

### 2.7.1 Definicija nadzornih plošč

Kakor nas sistem poslovnega obveščanja veže na kratico OLAP, nas na nadzorne plošče (angl. dashboards) navezuje kratica DIS (Direktorski Informacijski Sistem). Njihov primarni namen je bil vizualni prikaz finančnih kazalnikov uspeha preko tako preprostega grafičnega vmesnika, da naj bi ga razumeli tudi poslovodje. Sistemi so težko zaživel zaradi tehnoloških omejitev, nepopolnosti podatkov, nestabilnosti ipd.

Nadzorne plošče so se boljše uveljavile šele pri integraciji v OLAP sisteme. Zaradi raznolikosti, barvitosti in interaktivnosti današnjih plošč, bi lahko posplošili, da je nadzorna plošča prefinjen, barvit, vizualen prikaz različnih tipov grafikonov na enem zaslonu. Zaradi privlačnosti bi jo želel imeti vsak. Spominja na merilnike v avtomobilih in letalih. Dekorativna nasičenost pa v resnici odvrta pozornost od pravih problemov. Pri implementacijah se premalokrat upošteva njihov osnovni cilj: *»Nadzorna plošča je vizualen prikaz najpomembnejših informacij za doseg enega ali več naprednih poslovnih ciljev. Prikazana mora biti na posameznem zaslonu z namenom, da se kontrolne točke z očmi lažje preletijo«* [4].

Če zgoraj navedeno definicijo podrobneje razložimo:

- **Vizualen prikaz** so podatki, informacije so predstavljeni v kombinaciji s številkami in grafikoni. Poudarek je na vizualni obliki, saj nam dobra grafična predstavitev podatkov nudi bolj bogatene informacije kot zgolj numerična oblika. Pomembno je, da se uporabnika ne bremeni z nepotrebnimi vizualnimi efekti.
- **Informacije za doseg naprednih ciljev** - že za doseg enega naprednega cilja se potrebuje veliko različnih pogledov na določeno poslovno funkcijo ali proces. Ne gre torej za poseben tip informacije, temveč za kakršenkoli tip informacije, ki je v podporo poslovni odločitvi.
- **Prikaz na enem zaslonu** - nadzorna plošča je predstavljena na enem zaslonu, kot so vrhunska poslovodska poročila na enem papirju. Uporabnik mora videti celoten prikaz sočasno brez premikanja po zaslonu. Tako se učinkoviteje opazijo najpomembnejše nadzorne in kontrolne točke in uporabnik se lahko posveti razumevanju in ne iskanju informacij.
- **Kontrolne točke za učinkovitejši pregled** - nadzorna plošča mora biti zasnovana tako, da na kontrolnih točkah obstajajo prožilniki, alarmi in zastavice, ki po potrebi opozarjajo. Znati mora uporabniku namigniti, kje je potrebno raziskovati napake, pomanjkljivosti ali prekoračitve.
- **Prilagojenost na problemsko domeno** - vsak poslovni sistem je svojevrsten. Četudi so si mnogi po dejavnosti podobni, nimajo nikoli popolnoma enake strukture, količine in kvalitete podatkov. Nadzorna plošča mora biti specifična in prilagojena, tako da v celoti služi svojemu namenu.

## 2.7.2 Kategorije nadzornih plošč

### 2.7.2.1 Podpora pri odločanju za doseganje strateških ciljev

Kadar govorimo o strateških ciljnih imamo v mislih prikaze za poslovodje. Tovrstne nadzorne plošče imajo pozornost usmerjeno na visoko agregiranih kazalnikih uspeha, ki prikazujejo splošno poslovanje do današnjega dne. Vključujejo tudi napovedi za prihodnost na podlagi dosedanjih gibanj in trendov. Pri tem tipu nadzornih plošč se najbolj obnesejo zelo preprosti mehanizmi (preprosti grafikoni), ki ne zahtevajo veliko stopnje interaktivnosti z uporabnikom, saj se poslovodje s podrobnejšimi analizami nimajo časa ukvarjati. Pozornost je usmerjena na prikazovanje trendov in gibanj različnih kazalnikov. Visoka ažurnost podatkov nima bistvenega pomena, saj se gibanja (posebno pri večjih sistemih) ne spremenijo v zelo kratkem času.

### 2.7.2.2 Analitična funkcija

Popolnoma nasprotne so zahteve za analitika. Primerne so bolj napredne in obogatene primerjave ter daljša zgodovina trendov. Prikazi so bolj prefinjeni in jim omogočajo veliko stopnjo interakcije. Imeti morajo možnost vrtanja v globino, spreminjanja dimenzij in samodejnega dodajanja novih grafikonov, vrtilnih tabel itd. Analitiku ni dovolj zgolj informacija, da trend števila naročil v zadnjem obdobju pada. Njegova naloga je, da poišče utemeljene vzroke za padec in pripravi predloge za ukrepanje.

### 2.7.2.3 Meritve na operativnem nivoju

Opušča se mišljenje, da so nadzorne plošče zasnovane le za prikazovanje na strateških ali analitičnih nivojih. Vedno bolj se uveljavljajo tudi na operativnih ravneh. Kadar se prikazujejo operacije je potrebno vedeti, da je tukaj ažurnost in točnost podatkov najbolj kritičnega pomena. Nadzornik proizvodnega stroja mora ukrepati nemudoma, ko gre kaj narobe. Lahko tudi s pomočjo vizualnega razumevanja delovnega procesa pripomore pri optimizaciji.

## 2.7.3 Merilniki in načini prikazovanja podatkov

Če želimo prikazovati podatke na način, da nam nudijo informacijo o tem, kaj se trenutno dogaja, jih moramo med seboj primerjati po različnih obdobjih in dimenzijah. Glede na sporočilo, ki ga servirajo številke jih lahko razdelimo na [5]:

### **Kvantitativna razmerja**

Kvantitativna sporočila so sama po sebi neuporabna, dokler jih ne primerjamo z drugimi, ki se nam zdijo v danem trenutku pomembni. Npr. promet neke družbe v danem letu, nam ne pove dosti, dokler ga ne primerjamo s prejšnjim obdobjem, planom ali s prometom konkurenčne družbe.

**Relacije med kategorijami** - so najpogostejša oblika s pomočjo katere podatke razvrstimo na različna področja. To lahko storimo na:

- nominalen način (ustvarjamo smiselne primerljive delitve podatkov npr. po regijah),

- vrstilen način (naše delitve razvrstimo),
- hierarhičen način (osnovnim delitvam dodajamo še podrejene).

### Relacije med količinami

Posamezne delitve kategorij na podkategorije, odpirajo primerjalne možnosti:

- ustvarjanje vrstnega reda (od najuspešnejše do najmanj uspešne poslovne enote),
- izračunavanje kvocientov, razmerij (vrednost se prikazuje v odstotkih),
- korelacija ali soodvisnost (vrednostni prikaz dveh ali več množic, z namenom ugotavljanja njihovih soodvisnosti).

**Splošni, agregacijski merilniki** - o njih govorimo kadar pri dimenzijah izračunavamo statistične vrednosti (povprečja, distribucij, standardnih odklonov, mediane, itd...)

### Neštevni podatki na nadzorni plošči

Veliko ljudi zamenjuje nadzorne plošče in KPI. Nadzorne plošče so učinkovito sredstvo ali medij za prikazovanje KPI. Velikokrat pa niso le številke ali grafikon tisto, kar želimo prikazovati. Na nadzorni plošči prikazujemo tudi neštevne podatke v obliki sezamov, kot so:

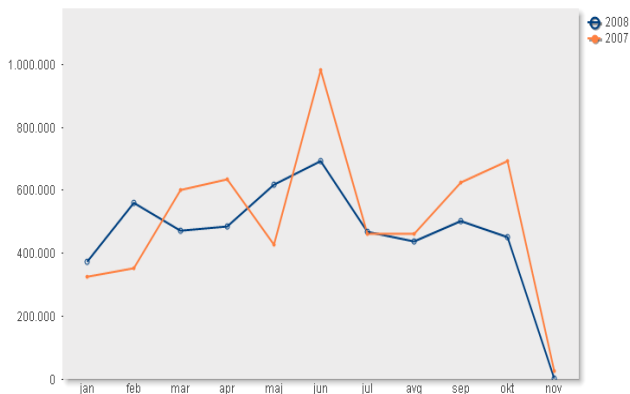
- 10 najboljših strank, referentov, poslovnih enot,
- seznam najbolj urgentnih nalog,
- seznam problemov in napak, ki jih je potrebno raziskati itd.

## 2.7.4 Najpogostejše napake pri izgradnji nadzornih plošč

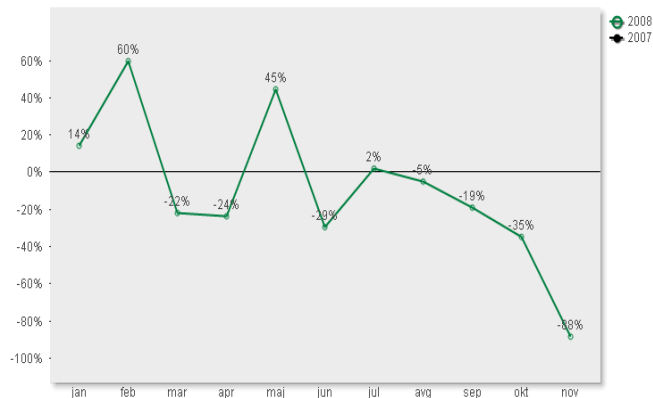
Pri izgradnji in oblikovanju nadzornih plošč je priporočljivo, da se izognemo napakam kot so:

- preobsežnost, ki ne doseže okvirja enega zaslona,
- prikazovanje podatkov v napačnem kontekstu,
- prikazovanje prevelikih podrobnosti sočasno,
- izbira neučinkovitih merilnikov,
- napačna ali prekomerna uporaba barv, dekoracij, ki zasenčijo pomembne informacije,
- poudarjenost nepomembnih podatkov.

Primeri na slikah 19 in 20 ponazarjata slabo in dobro korelacijo prometa med leti 2008 in 2007. Oba grafikona vsebujeta enake vrednosti podatkov.



Slika 19: Izgubljena izraznost namenjene informacije



Slika 20: Poudarek odsopanja od ciljne vrednosti

## 2.8 POSTAVITEV IN IZDELAVA OBJEKTOV

### 2.8.1 Lastnosti dokumenta

Dokument in vse njegove nastavitve, vključno s prebranimi podatki v AQL podatkovnem modelu, so shranjeni v QVW datoteki. Na nivoju dokumenta lahko analitik določa splošne in varnostne nastavitve, ureja dimenzijske skupine, vrednosti spremenljivk, prožilnike makrojev in privzete nastavitve delovnih zvezkov (angl. sheets).

### 2.8.2 Delovni zvezki

QlikView dokument ima lahko enega ali več delovnih zvezkov. Znotraj njih se dostopa in ureja posamezne objekte (slika 22) z njihovimi nastavitvami in nastavitvev makro prožilnikov. Delovni zvezki niso logično povezani. Torej se vsebinsko in logično lahko poljubno razporedijo. Da med njimi naredimo vsebinske povezave, ima vsak delovni zvezek možnost pogojnega prikazovanja (slika 21).



Slika 21: Nastavitveno polje za pogojni prikaz

ObjectID	Type	Caption	ShowMode	CalcTime	Layer	Mem...
CH03	Bar Chart	Drill-down	Normal	16	0	1 KJ
CH06	Bar Chart	Sales Forecast	Minimized	0	0	0 KJ
CH07	Bar Chart	Sales per Country	Minimized	0	0	0 KJ
CS03	Current Selecti	Current Selections	Normal	0	0	0 KJ
LB09	List Box	Country	Normal	0	0	0 KJ
LB13	List Box		Normal	0	0	0 KJ
LB14	List Box	Salesman	Normal	0	0	0 KJ
LB16	List Box		Normal	0	0	0 KJ
LB17	List Box		Normal	0	0	0 KJ
LB18	List Box		Normal	0	0	0 KJ
LB19	List Box		Normal	0	0	0 KJ

Slika 22: Seznam objektov v delovnem zvezku

### 2.8.3 Objekti

QlikView pozna različne objekte: tabele, grafikone, vnosna polja, sezname, statistična polja, gumbе, tekstovni objekte itd.

Ločimo naslednje objekte:

- **Lokalne objekte** – to so posamezni objekti, ki so shranjeni znotraj QVW dokumenta. Ti so na voljo vsakemu uporabniku, ki dostopa do dokumenta lokalno ali preko strežnika.
- **Osebnе strežniške objekte** - na voljo so le kadar delamo analize z dokumenti, ki se nahajajo na QlikView strežniku in so samo za overjene uporabnike. Shranjeni so v repozitoriju in dostopni samo uporabniku, ki jih je generiral. Te objekte je možno upravljati v posebnih nastavitvah strežniških objektov.
- **Porazdeljene strežniške objekte** - tudi porazdeljeni strežniški objekti se nahajajo v repozitoriju QlikView strežnika. Do njih dostopamo preko kolaboracijske palete (angl. collaboration pane).

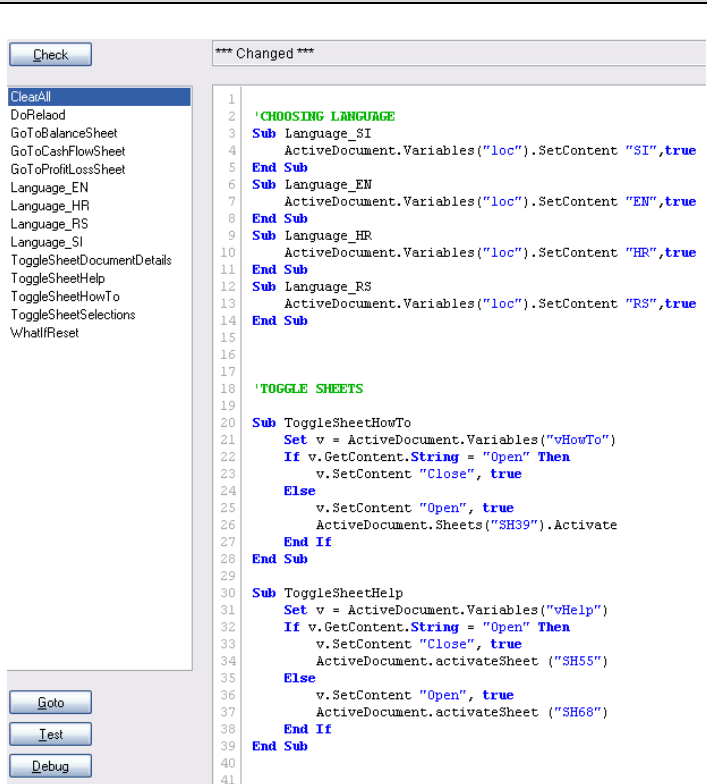
- **Prilagojeni objekti** - poleg širokega nabora privzetih objektov imamo tudi možnost prilagajanja objektov. Ta objekt nosi brezokenski OCX kontrolnik. Komunikacija med OCX kontrolo in QlikView dokumentom se ustvari preko vmesnika (angl. QlikView automation interface). Tako se dopušča možnost razvoja zelo specifičnih podatkovnih in vizualnih učinkov.

## 2.8.4 Izvoz in tiskanje podatkov

Vse izračunane vrednosti ali vizualen prikaz se lahko neposredno natisne ali izvozi v Microsoft Excel orodje. Poleg neposrednega tiskanja se lahko formirajo tudi statična poročila.

## 2.8.5 Avtomatizacija z makroji

Znotraj dokumenta obstaja priložen vmesnik za pisanje makrojev v 'VBScript' ali 'Jscript' jeziku. Izvajajo se ob dogodkih ali prožilnikih znotraj aplikacije. Dogodki ali prožilniki so lahko na nivoju dokumenta, delovnega zvezka, posameznih objektov ali spremenljivk.

QlikView makroji	Komentar
 <pre> 1 2 3 'CHOOSING LANGUAGE 4 Sub Language_SI 5   ActiveDocument.Variables("loc").SetContent "SI",true 6 End Sub 7 Sub Language_EN 8   ActiveDocument.Variables("loc").SetContent "EN",true 9 End Sub 10 Sub Language_HR 11   ActiveDocument.Variables("loc").SetContent "HR",true 12 End Sub 13 Sub Language_RS 14   ActiveDocument.Variables("loc").SetContent "RS",true 15 End Sub 16 17 18 'TOGGLE SHEETS 19 20 Sub ToggleSheetHowTo 21   Set v = ActiveDocument.Variables("vHowTo") 22   If v.GetContent.String = "Open" Then 23     v.SetContent "Close", true 24   Else 25     v.SetContent "Open", true 26     ActiveDocument.Sheets("SH39").Activate 27   End If 28 End Sub 29 30 Sub ToggleSheetHelp 31   Set v = ActiveDocument.Variables("vHelp") 32   If v.GetContent.String = "Open" Then 33     v.SetContent "Close", true 34     ActiveDocument.activateSheet ("SH55") 35   Else 36     v.SetContent "Open", true 37     ActiveDocument.activateSheet ("SH68") 38   End If 39 End Sub 40 41 </pre>	<p><i>Na sliki 23 je prikazanih nekaj preprostih makrojev napisanih v 'VBScript' jeziku. Začnejo se z besedo SUB in končajo z END SUB.</i></p> <p><i>Prvih 14 vrstic le spreminja vrednosti globalne spremenljivke »loc«.</i></p> <p><i>V drugem delu imamo preklapne procedure. Preberemo vrednost spremenljivke »vHowTo« in ji spremenimo vrednost. Če je njena vrednost enaka nizu »Open« se aktivira delovni zvezek »SH39«.</i></p> <p><i>Po enakem postopku deluje zadnja procedura ToggleSheetHelp.</i></p>

Slika 23: Urejevalnik makrojev

Makroji razvijalcu in analitiku omogočajo veliko svobode znotraj dokumenta. Definirajo lahko lastne procedure ali funkcije. Ko se odločamo za uporabo, je dobro vedeti, da se izvajajo nekoliko počasneje kot preddefinirane. Previdnost je potrebna še posebej pri strežniški arhitekturi, saj se vsi

makroji procesirajo na strežniku. To ima lahko negativne posledice, zato se takšna uporaba makrojev odsvetuje.

## 2.8.6 Grafikoni in tabele

Najuporabnejši objekti so grafikoni in tabele. Omogočajo uporabniku prijazen prikaz dimenzij in njihovih agregatov. Lahko nam prikazujejo tudi frekvence pojavitev polj ali izračunanih entitet. Običajno se določen atribut tabele v podatkovnem modelu izbere za dimenzijo (pri stolpčnih grafikoni vodoravna os in agregacijski stolpec pri vrtilnih tabelah) in zatem še agregacijski izraz, ki ga želimo prikazovati (npr.: Vsota). Na izbiro je veliko vrst in tipov grafikonov, oblik, nastavitvev za vrtilne tabele itd. Grafikoni in tabele so definirani znotraj delovnega zvezka in pripadajo določenemu dokumentu kot ostali objekti.

### 2.8.6.1 Analiza množic (angl. set analysis)

Množice se uporabljajo znotraj agregacijskih izrazov. Običajno so agregacijski izrazi narejeni na množici vseh zapisov. Delne rezultate omejujejo le izbrani filtri in hierarhično opredeljene dimenzije. Z nadgrajeno sintakso od QlikView verzije 8.5 dalje, lahko na učinkovitejši način kar znotraj izrazov določimo množice, ki jih želimo prikazovati.

*Primer:*

```
sum( {1-$} Znesek ) (1)
sum( {$<Regija= {'Dolenjska'} >} Znesek ) (2)
```

Izraz (1) nam vrne kot rezultat izključujočo vsoto naših trenutnih izbir. Če npr. izberemo filter za obdobje mesec maj, nam bo izraz vrnil vsoto za vse mesece in druge možne izbire, le mesec maj ne. Pri izrazu (2) dobimo vsoto, ki bo vedno prikazala le regijo dolenjske.

Vsaka množica se nahaja znotraj oglatih { } oklepajev. Znak \$ predstavlja trenutne izbire, konstanta 1 pomeni celotno množico. Kombinacija {1-\$} pomeni izključujočo množico. Pri analizi nam pomagajo tudi zanamki (angl. bookmarks). Z njimi uporabnik shrani različne kombinacije podmnožic, ki jih želi medsebojno primerjati. Nad množicami lahko izvajamo tudi osnovne operacije.

#### Operatorji množic so:

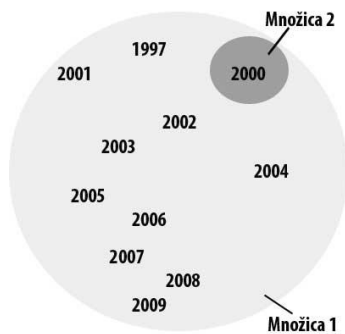
- '+' uporabljamo za unijo dveh množic,
- '-' pomeni razliko množic. Za rezultat dobimo množico zapisov, ki pripadajo samo prvi in ne drugi množici,
- '\*' presek množic,
- '/' ekskluzivni ali (XOR) - operacija nam vrne množico zapisov pripadajoče prvi ali drugi množici, nikakor pa ne obema hkrati.

## Modifikatorji množic:

Množica se lahko modificira oz. spreminja na osnovi dodatnih pogojev ali izbir, ki jih določamo. Te lahko zapišemo v obliki izrazov. Modifikatorji lahko vsebujejo več različnih polj. Začnejo se z '<' in zaključijo z '>'. Izraz (2) ima za modifikator polje 'Regija' in znotraj njega podmnožico 'Dolenjska'.

Modifikatorji:		Komentar:
<Leto = {"200*", 1997} - {2000}>	(3)	Množica vseh let, ki se začnejo z 200 vključno z letom 1997 pri tem pa zraven ne sodi leto 2000.
<Leto = {">1982<2009"}>	(4)	Množica let med 1982 do 2009.
<Leto = {\$(#vLani)}>	(5)	Množica leta 'vLani'. \$(#vLani) je spremenljivka.

Tabela 13: Primeri modifikatorjev pri analizi množic



Slika 23: Prikaz množic (3)

Obrazložitev slike 23:

Množica 1 = {"200\*", 1997}

Množica 2 = {2000}

Rezultat {"200\*", 1997} - {2000} so le letnice znotraj svetlejših sive barve.

Izraze in množice lahko med seboj tudi gnezdimo. S tem dobimo možnost dinamičnega prikazovanja določenih seznamov, katerih pogoji se izračunavajo z analizo množic.

Primer gnezdenega izraza:

$\text{sum}(\{ \$\langle \text{Stranka} = \{ \text{"=Sum}(\{ 1 \langle \text{Leto} = \{ 2007 \} \rangle \} \text{Znesek} ) > 5000 \} \rangle \} \text{Znesek} )$  (6)

Kot rezultat izraza (6) dobimo vedno le seštevek prodaje tistih strank, katerih skupna vrednost prodaje je v letu 2007 presegla 5000.

Za enakovreden rezultat izraza (6) bi bilo običajno že znotraj ETL postopka ustvariti izločevanje dimenziji, vsote prodaje strank po letih itd. Z analizo množic spreminjanje skripta ni potrebno kar prihrani veliko časa. Zgornji izraz lahko vpiše uporabnik samostojno znotraj objektov brez dodatnega razvoja.

### 2.8.6.2 Agregacije na strani dimenzij s funkcijo Aggr()

Pogosto je uporabno, če lahko izračunane rezultate uporabimo kot dimenzije. Orodje dopušča tudi uporabo pogojnih in izračunljivih dimenzij. Slika 24 prikazuje preprost seznam prodajalcev in njihovih strank.

*Primer 1:*

Na vprašanje, koliko kupcev ima posamezen prodajalec, odgovorimo s pomočjo vrtilne tabele. Prodajalce postavimo na stran dimenzij in izraz na strani agregacije se glasi: '=count(Stranka)'. Rezultat vidimo na sliki 25.

*Primer 2:*

Sedaj naše povpraševanje nadgradimo na osnovi prvega rezultata. Koliko prodajalcev ima samo eno stranko? Koliko prodajalcev ima 3 in več strank? Na to vprašanje lahko dobimo dogovor, če na strani dimenzij uporabimo funkcijo '**aggr()**'. Znotraj nje vstavimo prejšnji izraz in dimenzijo '**aggr(count(Stranka), Prodajalec)**'. To nam vrne enolične vrednosti različnih strank na prodajalce. Na strani izraza moramo še prešteti enolične prodajalce z izrazom '**count(distinct Prodajalec)**'. Rezultat ponazarja slika 26.

Prodajalec	Stranka
Janez Novak	Kupec_1 d.o.o.
Miha Bergant	Kupec_2 d.o.o.
Urška Popovič	Kupec_3 d.o.o.
Nejc Grohar	Kupec_4 d.o.o.
Nadja Košak	Kupec_5 d.o.o.
Barbara Klopčič	Kupec_6 d.o.o.
Miha Bergant	Kupec_7 d.o.o.
Miha Bergant	Kupec_8 d.o.o.
Miha Bergant	Kupec_9 d.o.o.
Barbara Klopčič	Kupec_10 d.o.o.
Janez Novak	Kupec_11 d.o.o.
Nadja Košak	Kupec_12 d.o.o.
Nadja Košak	Kupec_13 d.o.o.

Slika 24: Seznam prodajalcev in strank

Prodajalec	count(Stranka)
	13
Barbara Klopčič	2
Janez Novak	2
Miha Bergant	4
Nadja Košak	3
Nejc Grohar	1
Urška Popovič	1

Slika 25: Štetje strank

count(distinct Prodajalec)	
=aggr(count(Stranka), Prodajalec)	count(distinct Prodajalec)
	6
	2
2	2
4	1
1	2
3	1

Slika 26: Uporaba funkcije Aggr( )

## 2.9 QLIKVIEW STREŽNIK

Do sedaj so opisani postopki in metode s katerimi se razvija aplikacija v orodju QlikView. Po naboru funkcionalnosti, ki jih končni uporabnik lahko v aplikaciji izvaja, ločimo tri licenčne kategorije:

- **QlikView Developer** (Polna funkcionalnost z možnostjo kreiranja in urejanja skripta),
- **QlikView Professional** (Polna funkcionalnost omejena na 65536 vrstic vhodnih podatkov),
- **QlikView Analyzer** (Onemogočeno urejanje objektov in skripta).

Aplikacija je sestavljena le iz ene QVW datoteke. Ta vsebuje podatke shranjene znotraj AQL, objekte (grafikone, vrtilne tabele ipd.), varnostne nastavitve in ostale pripadajoče metapodatke.

QlikView strežniška arhitektura nudi platformo za gostovanje, distribucijo in porazdeljevanje QlikView informacij preko lokalnega omrežja, intraneta in interneta. Centralno usmerjen in upravljan sistem omogoča medsebojna povezovanja skupin uporabnikov, dokumentov in objektov znotraj varnega okolja.

Strežnik podpira več tipov odjemalcev. Najpogosteje se za povezavo na strežnik uporablja kar Windows odjemalec QlikView. Na voljo imamo tudi dva načina Java odjemalcev, ki omogočata analize znotraj standardnih spletnih brskalnikov. Tretja možnost je ActiveX vtičnik za IE odjemalec in zadnji način tehnologija AJAX. Ta omogoča prikaz QlikView objektov znotraj standardnega brskalnika brez potreb po dodatni namestitvi na odjemalčevi strani. Toda le nameščena Windows QlikView odjemalec in ActiveX vtičnik za IE, nudita končnemu uporabniku popolnoma enak izgled in nabor funkcionalnosti dokumentov kot izvorni. Ostali odjemalci (Java objects, QlikX in AJAX) bazirajo na postavljanju individualnih objektov v HTML okolju. To nudi možnost razvijalcem umestitev objektov na poljubna mesta in integracijo znotraj spletnih aplikacij.

### 2.9.1 Upravljanje

Strežniška aplikacija vključuje tudi upravljavski uporabniški vmesnik (angl. server management console). Z njim se urejajo nastavitve za kontrolo strežniških operacij, ki jih delimo na:

- **Splošne nastavitve** – kjer urejamo sočasne seje, časovne omejitve (angl. timeouts), maksimalno število zadržanih dokumentov v spominu, stopnje kolaboracije med uporabniki itd.
- **Nastavitve map** - določitev primarne lokacije QVW dokumentov, dodajanje podmap, možnost uporabe ločene lokacije za Java odjemalce.
- **Varnost** - nastavitve za overitve različnih tipov odjemalcev (angl. authentication), datotečne avtorizacije NTFS (angl. NT File System) ali DMS (angl. Document Metadata Service), odpiranje komunikacijskih vrat itd.
- **Napredne nastavitve** – gre za rezervacija spomina za izračunavanje izrazov, porazdeljenost CPU procesiranja, itd.
- **Ostalo** - spletne mape, aktivni uporabniki, urejanje administratorskih skupin uporabljene licence, graf zasedenosti, namestitvev itd.

The screenshot shows the QlikView Server Management Console interface. At the top, there is a navigation bar with tabs: Install, Administration, General, Folders, Security, Web Folders, Advanced, Performance, Loaded Documents, Active Users, Cal Info, and Graph. The 'General' tab is selected. Below the navigation bar, there are three main sections:

- Sessions:**
  - Maximum Number of Concurrent Sessions: 5000
  - Possible Session Timeout (seconds) (Server may terminate inactive session and re-use slot if needed for other user): 1800
  - Maximum Inactive Session Time (seconds) (0 means no limit): 1800
  - Maximum Total Session Time (seconds) (0 means no limit): 0
- Documents:**
  - Document Timeout (minutes): 480
  - Allow only one copy of document in memory
- Logging:**
  - Enable Session Logging
  - Enable Performance Logging every 5 minutes
  - Enable Event Logging
  - Log Folder: J:\Documents and Settings\All Users\Application Data\QlikTech\
  - Split Files:
    - Never
    - Daily
    - Weekly
    - Monthly
    - Yearly
  - Verbosity:
    - Low
    - Medium
    - High

Slika 27: Splošne nastavitve upravljaljskega uporabniškega vmesnika

## 2.9.2 Varnost

### 2.9.2.1 Komunikacija glede na vrsto odjemalca

- Windows odjemalci - povezava med strežnikom in Windows odjemalci je osnovana na 128-bitnem šifriranju asimetričnega RSA algoritma. Stopnja šifriranja (angl. encryption) je manjša le v primerih, če odjemalec ne podpira dovolj visoke stopnje šifriranja. Privzeta vrata so 4747 in se na njih izvaja QlikView protokol (QVP).
- Java odjemalci - odjemalci na Java platformi uporabljajo varnost na osnovi premešanja podatkov (angl. scrambling) in ne šifriranja. Sicer se višja varnost lahko doseže s pomočjo 'tunneling mode' in HTTPS protokola.
- AJAX odjemalci - najmanjša osnovna varnost pri komunikaciji je pri AJAX odjemalcu. Varnost je v tem v celoti odvisna od nastavitve spletnega strežnika.

### 2.9.2.2 Datotečni sistem

Pri datotečnem sistemu poznamo dva načina avtorizacije:

- **NTFS Avtorizacija**  
Pri NTFS avtorizaciji bo strežnik dovolil dostop le odjemalcem, ki imajo na osnovi operacijskega sistema Windows dostopne pravice do določenih dokumentov. Ta način avtorizacije je v celoti odvisen od operacijskega sistema in QlikView ne opravlja dodatnih varnostnih preverjanj.

- **DMS Avtorizacija**

DMS je nit, ki jo poganja QlikView strežnik. Njen primarni namen je kontrolirati nalaganje dokumentov. Druga funkcija je avtorizacijska kontrola. Uporablja se za avtorizacijski proces pri ne-Windows odjemalcih. Njena naloga je tudi upravljanje pravic individualnih in anonimnih uporabnikov. DMS ustvari meta datoteke z enakim imenom kot izvorne. DMS sodeluje tudi neposredno z DSP-jem(angl. Directory Service Provider) in DSC-jem (angl. Directory Service Connector), ki pripadata aplikaciji QlikView Publisher. DMS nastavitve se urejajo pod varnostno kategorijo znotraj strežniškega upravljaljskega uporabniškega vmesnika (Slika 27).

Oba načina varujeta na nivoju celotnih datotek. Če ima končni uporabnik datoteko lokalno na trdem disku, jo lahko odpre. Zato se potrebuje varnost tudi znotraj datotek, kakor je opisano v poglavju 2.6. Potem se nad vsakim uporabnikom izvaja overitev ob kakršnemkoli načinu.

### 2.9.2.3 Overitev uporabnika

- **Overitev na strani odjemalca**

Ta tip overjanja se lahko izvaja le če sta QlikView strežnik in odjemalec znotraj iste domene. Dosegljiv mora bit tudi DS (angl. Directory Service). Uporabnik je praktično že overjen takoj, ko zažene QlikView aplikacijo. Nato se naredi postopek avtorizacije.

- **Overitev preko Windows spletnega strežnika**

1. Odjemalec kontaktira spletni strežnik.
2. Zatem ko uporabnik klikne na QlikView povezavo, uporabi spletni strežnik QvsComRemote.dll in posreduje zahtevo z uporabniškim imenom na QVS. Ta vrne nazaj dovoljenje za naslednji korak.
3. Odjemalec zažene aplikacijo in pošlje zahtevo s sprejetim dovoljenjem na QVS.
4. QVS zaupa spletnemu strežniku in zatem preveri DMS, če ima uporabnik dovoljenje za ogled dokumenta.

- **Overitev preko drugih spletnih strežnikov**

1. Odjemalec kontaktira spletni strežnik.
2. Ko uporabnik klikne na QlikView povezavo, spletni strežnik preko spletne storitve (angl. web service) kontaktira QlikView spletni strežnik, ki posreduje zahtevo naprej na QVS. Ta izda dovolilnico ali pa zahtevo zavrne.
3. Odjemalec zažene aplikacijo in pošlje zahtevo z dovolilnico na QVS.
4. QVS zaupa spletnemu strežniku in nadaljuje proces v avtorizacijo preko DMS.

QVS torej ne izvaja procesa overitve uporabnika. To prepušča spletnim strežnikom. Overja le proces, ki ga prosi za izdajo dovolilnice.

### 2.9.3 Arhitektura

Strežniška arhitektura zahteva pri komunikaciji tri medsebojno konsistentne in ustrezno varovane primarne komponente. V medsebojno povezovanje je lahko vključenih več različnih odjemalcev, tipov omrežja, varnostnih protokolov ipd. Te komponente so:

- **QlikView strežnik (QVS)** - servira celoten nabor QlikView funkcionalnosti na odjemalca. Računalnik na katerem gostuje mora poganjati Windows operacijski sistem.

- **Odjemalec**, ki s strežnikom komunicira preko spletnega brskalnika ali neposredno preko QlikView lupine.
- **Spletni strežnik**, ki omogoča serviranje HTML strani do odjemalca in skrbi za komunikacijo s QVS-jem.

Vsi trije procesi se lahko odvijajo na enem fizičnem računalniku ali pa popolnoma ločeno. V praksi sta najpogosteje fizično skupaj spletni strežnik in QlikView strežnik, odjemalec pa na delovni postaji. Kompleksnost strukture hitro naraste, če so ti trije procesi povsem ločeni in dodatno ovirani z različnimi nivoji požarnih zidov, domen in spletnimi strežniki. Možnosti za implementacijo je veliko, le upoštevati je potrebno minimalne zahteve, da:

- lahko QVS poganja le Windows operacijski sistem in
- obstajati mora vsaj ena komunikacijska pot med odjemalcem in strežnikom.

### 2.9.3.1 Spletni strežniki

V večini primerov se za spletni strežnik uporablja Microsoft IIS. Od verzije QlikView 8 obstaja tudi alternativna možnost QlikView spletnega strežnika. Ta lahko deluje popolnoma neodvisno in je namenjena predvsem serviranju QlikView dokumentov. Ostali spletni strežniki lahko sodelujejo s preusmeritvami na enega od njih.

### 2.9.3.2 »Tunneling mode« komunikacija

Če so vrata za QVP (privzeto 4747) zaprta ali blokirana, bodo Windows in Java odjemalci poizkusili vzpostaviti "tunneling mode" povezavo pri standardnih HTTP (80) vratih. Ob vzpostavitvi komunikacije morata kot vmesnika sodelovati QVSComRemote.dll ali QlikView spletni strežnik. Pri "tunneling mode" komunikaciji se močno poveča mrežni promet, kar ima za posledico veliko slabše odzivne čase, še posebej pri varnem protokolu HTTPS. Za boljši odziv se odsvetuje tudi uporaba t.i. proxy strežnikov.

### 2.9.3.3 Repozitorij objektov

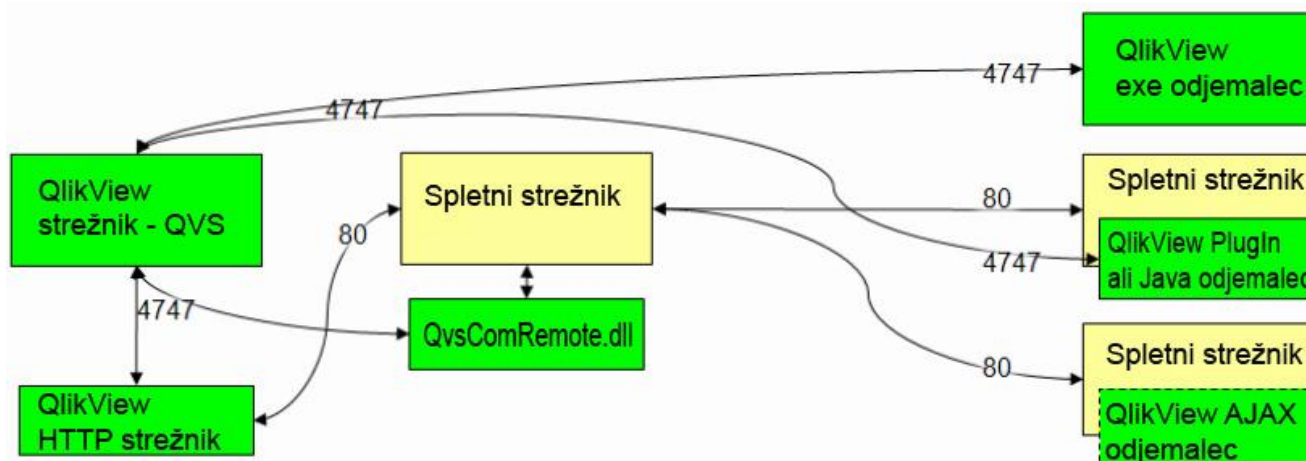
Za naprednejše analiziranje je pri novejših različicah možno dinamično sodelovanje in interaktivno upravljanje določenih objektov med uporabniki. Te objekti so:

- zaznamki (angl. bookmarks),
- tabele in grafikoni,
- statična poročila.

### 2.9.3.4 Odjemalci

QlikView odjemalci se medsebojno ločijo po načinu dostopa oz. komunikaciji s strežnikom in po stopnji funkcionalnosti, ki jo nudijo uporabniku za izvajanje analiz.

➤ **Komunikacija s QVS:**



Slika 28: Komunikacijska struktura med strežnikom, spletnim strežnikom in odjemalci[16]

Na sliki 28 lahko vidimo, da QVS z vsemi komunicira samo preko vrat 4747, razen v primeru drugačnih nastavitvev. Če je spletni strežnik Microsoft IIS, dostopa do QVS preko QvsComRemote.dll sicer se zahteve posredujejo na QlikView HTTP strežnik in nato na QVS. Na spletne strežnike lahko gledamo tudi kot vmesnike ali pretvornike vrat za “tunneling mode” izvajanje.

➤ **Primerjalna tabela odjemalcev razvrščena po stopnji funkcionalnosti:**

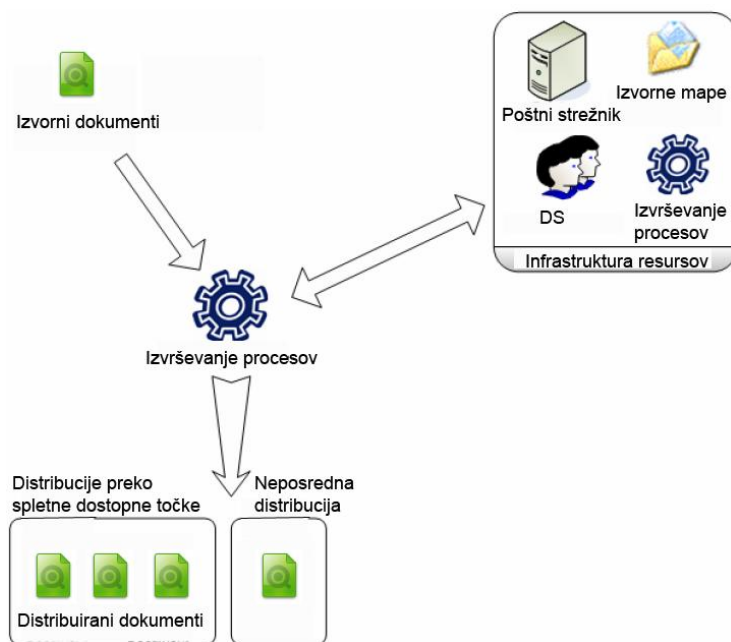
Tip	QV odjemalec	Opis
WIN	Windows exe odjemalec	Pokriva vse tipe QlikView (Developer, Profesional, Analyzer). Zunanji izgled dokumenta je popolnoma enak originalnemu in vsebuje celoten nabor funkcionalnosti. Glavna pomanjkljivost je, da so pri namestitvi obvezne administratorske pravice na odjemalčevi delovni postaji. S strežnikom lahko komunicira preko QlikView 128-bitnega enkriptiranega protokola(QVP) običajno na vratih 4747. Druga možnost je preko “Tunnel mode” načina, kjer se QVP “zavije” v HTTP ali HTTPS protokol.
WIN	Analitik za IE	Enako funkcionalen kot klasičen odjemalec in deluje na ActiveX vtičniku za IE, ki ga je potrebno predhodno namestiti. S strežnikom zna ravno tako komunicirati preko QVP. Dokumente lahko prebira tudi preko spletne dostopne točke.
WIN	QlikX Analitik za IE	S pomočjo ActiveX vtičnika za IE lahko prikazuje posamezne QlikView objekte. Zato je potrebna izgradnja in oblikovanje spletne strani, znotraj katere se integrirajo. Za komunikacijo potrebujemo spletni strežnik in QlikView strežnik.
JAVA	Java odjemalec	Pri tem odjemalcu ni potrebna dodatna namestitvev. Glavna prednost je da lahko deluje na vsakem okolju, ki omogoča namestitvev JRE. Malce je okrnjen in spremenjen izgled kot tudi nabor funkcionalnosti. Za komunikacijo s strežnikom potrebuje spletni strežnik in QlikView strežnik.
JAVA	Java objektni odjemalec	Podobno kot QlikX je potreben razvoj in oblikovanje spletne aplikacije. Glavni prednost je v neodvisnosti od okolja. Ima okrnjen in spremenjen

		izgled ter tudi omejeno funkcionalnost. Komunicira lahko neposredno s strežnikom tudi preko vrat 4747, vendar mora biti predhodno vzpostavljena povezava preko spletnega strežnika, ki zna servirati strani z Java Appleti preko vrat 80.
AJAX	Spletni odjemalec	Zahteva izgradnjo spletne aplikacije. AJAX tehnologija deluje na osnovi asinhrono izmenjave podatkov, Javaskripta in XML. Predstavlja nov način komunikacije s strežnikom. Prednost leži v dejstvu, da ob interakciji s strežnikom ni potrebna osvežitev spletne strani. Izgled objektov je zelo podoben originalnim. Pri tem je najmočnejši adut, da ne zahteva nikakršne dodatne namestitve na strani uporabnika, vendar pa ne more nikoli komunicirati neposredno s QlikView strežnikom. Vedno se vzpostavi povezava najprej do spletnega strežnika(vrata 80), ta pa posreduje nadaljnje poizvedbe na QlikView strežnik.

*Tabela 14: Primerjava različnih QV odjemalcev*

## 2.10 QLIKVIEW PUBLISHER

QlikView Publisher je pomembno dopolnilo strežniški postavitvi, ki skrbi za upravljanje vsebin dokumentov in njihovega avtorizacijskega dostopa. Njegova primarna naloga redno avtomatizirano osveževanje podatkov in napredno upravljanje tega procesa. Skrbeti mora za distribucijo podatkov od dokumentov do končnih uporabnikov znotraj kot tudi zunaj organizacije. Vsak uporabnik mora dobiti izključno samo tisto vsebino podatkov, ki mu je namenjena.



Slika 29: Skica izvajalnega procesa QlikView Publisher [15]

### 2.10.1 QlikView Publisher sestavljajo:

- spletna nadzorna plošča za upravljanje s procesi;
- komandni center je njegova glavna komponenta, ki vzdržuje repozitorij in povezuje med seboj druge komponente;
- izvršilni proces (angl. execution service), ki je zadolžen za pripravo in dostavo QlikView datotek;
- DSC (angl. Directory Service Connector) je odgovoren za komunikacijo z DS (angl. Directory Service), sledljivost vseh uporabnikov in skupin tega okolja;
- dostopna točka (angl. access point) je spletna stran, ki omogoča uporabnikom dostop do dokumentov. Odgovorna je tudi za pripravo map in ustreznih pravic.

### 2.10.2 Tipi virov

Uporabnik nadzorne plošče lahko nastavi veliko število virov, katere mora Publisher pripraviti, obdelati in distribuirati v končne dokumente. Vire delimo na dve skupini:

- **infrastrukturni viri**
  - ES (angl. Execution Service) vir, ki pripravi QlikView datoteke in jih porazdeli med uporabnike.
  - DS vir, ki hrani podatke o vsakem uporabniku. Za vsak vir je potreben svoj DSP (angl. Directory Service Provider). DSP je povezava na določen DS. DSP omogoča povezavo na AD (angl. Active Directory), lokalne uporabnike in prilagojene uporabnike (to so uporabniki, ki obstajajo samo znotraj publisherja in nikjer drugje).
  - E-poštni strežnik, ki se uporablja za razdeljevanje QlikView datotek, pošiljanje alarmov in drugih obvestil.
  - Mape z dokumenti, ki beležijo spremembe in ustvarjajo povezave, kateri dokumenti predstavljajo vir za distribucijo.
  
- **distribucijski viri**
  - Ciljne distribucijske mape
  - Distribucija po e-pošti
  - Spletna dostopna točka
  - QlikView strežnik, ki je najpomembnejši vir in izvaja vsa izračunavanja, uporablja svoje pomnilniške in CPU kapacitete za analizo po želji odjemalca.

### 2.10.3 Avtorizacija s Publisherjem

V kolikor uporabljamo Publisher za avtorizacijo dokumentov, je vsaka QVW datoteka razdeljena na več datotek. Vsaka vsebuje samo potrebno primerno domeno podatkov za končnega uporabnika. To področje definiramo znotraj skripta v datoteki (razdelek 2.6). Vsaka uporabniška vloga je shranjena v svojo mapo z ustreznimi pravicami. Od tega nivoja dalje varnostne mehanizme ureja operacijski sistem.

## **3 IZDELAVA APLIKACIJE ZA FINANČNI KONTROLING**

### **3.1 POSLOVNO OBVEŠČANJE ZA PODPORO RAČUNOVODSKEGA ANALIZIRANJA, NADZIRANJA IN NAČRTOVANJA**

Računovodstvo je sestavljeno iz knjigovodstva, računovodskega načrtovanja, nadziranja in analiziranja [8]. Pri tem je knjigovodstvo del funkcije obravnavanja podatkov o preteklosti, načrtovanje del funkcije obravnave podatkov v prihodnosti, nadzor nadziranja in računovodska analiza del analiziranja podatkov. Slednji del je v praksi najtežavnejši, saj je potrebno dobljene rezultate temeljito preveriti in obrazložiti. Pri vsem tem se vedno pojavlja tudi vprašanje v kolikšni meri podatkom zaupamo in ali prikazujejo realen odsev poslovanja.

### **3.2 NAMEN KONTROLINGA**

Obstaja veliko razlag o kontrolingu družb. Kontroling pomeni krmiljenje, vodenje oz. usmerjanje k dogovorjenim ciljem. Kmalu pristanemo nazaj pri računovodskem nadziranju, analiziranju in načrtovanju. Govorimo torej o podporni dejavnosti v poslovanju, katere primarna naloga je dodatna informacijska podpora vodstvu družbe. S pravočasno analizo utemeljenih informacij o poslovanju in njegovem okolju, pomagajo pri poslovnih odločitvah za ukrepanje v prihodnosti. Kontroling se lahko izvaja na različnih področjih podjetja. Z aplikacijo za finančni kontroling se bomo osredotočili na računovodski oz. finančni sektor. Cilj takega kontroliranja je boljša zaščita sredstev podjetja in sprotno zagotavljanje verodostojnih računovodskih informacij. Aplikacija je namenjena uporabnikom, ki želijo na različne načine preverjati in spremljati poslovanje [9].

### **3.3 VIR INFORMACIJ**

Glavni vir informacij so poslovne knjige. Ob ogromni množici raznovrstnih poslovnih dogodkov, knjigovodskih listin in dokumentov ki ob tem nastajajo, je nujno potrebna sistematična urejenost in spremljanje podatkov. Te se danes zapisujejo v temeljne in pomožne poslovne knjige[8]. Glavna razlika med njimi je, da se v temeljnih poslovnih knjigah zapisujejo samo vrednostni podatki, v pomožnih pa tudi količinski. V temeljnih torej povzemajoče spremljamo ekonomske kategorije, v pomožnih pa tudi analitične podatke. Vsak poslovni sistem mora obvezno vsebovati dve temeljni poslovni knjigi: glavno knjigo in dnevnik glavne knjige. V glavni knjigi se spremljajo spremembe stanj na določenih kontih, ki zaznamujejo različne ekonomske kategorije, v dnevniku glavne knjige pa je poudarek na kronološkem zapisu poslovnih dogodkov.

### **3.4 DODATNI VIRI**

Danes ne zadostujejo več zgolj analize na nivoju glavne knjige. Za lažji nadzor toka informacij se vse pogosteje ustvarjajo dodatne dimenzije merjenja. V računovodstvu so že dolgo poznane dimenzije mest odgovornosti (stroškovna mesta, prihodkovna mesta, stroškovni nosilci). Te razbijamo še dlje na posamezne projekte, profitne centre, naložbena mesta itd. Med njimi pogosto obstajajo hierarhične ureditve. Na spodnjih ravneh so dimenzije različnih delovnih postopkov in na višjih posamezni oddelki sestavljenih iz več enot.

Ponavadi imajo ERP sistemi poleg tabele glavne knjige tudi dodatne analitične knjige, v katerih podrobneje beležijo dodane dimenzije poslovnih dogodkov. Te dve tabeli predstavljata tudi glavni vir podatkov pri izdelavi aplikacije za finančni kontroling.

### 3.5 KONTNI NAČRT IN KONTNE PREGLEDNICE

Pri kontnem načrtu imamo v mislih vsebinsko urejene konte po razredih, skupinah, sintetičnih in analitičnih kontih. Iz same številke konta lahko določimo, katero ekonomsko kategorij spremljamo. Zato nas podrobno poznavanje vsebine celotnega kontnega načrta v resnici ne zanima. Nanj gledamo kot preprost šifrant, ki je numerično in vsebinsko organiziran na hierarhičen način. Pri tem je vredno upoštevati dejstvo, da so prvi trije nivoji že zakonsko predpisani po vrsti družbe glede na dejavnost (npr. gospodarska družba, zadruga itd.)

Podobno lahko trdimo za kontne preglednice iz katerih lahko uporabnik zgradi hierarhične strukture za potrebe zakonsko določenih izkazov stanja (bilanca stanja, izkaz uspeha) in tudi podrobnejše analitično usmerjene izkaze (izkaz denarnih tokov, analiza stroškov) za interno uporabo.

Pri aplikaciji za finančni kontroling ne želimo biti odvisni od specifičnih kontnih načrtov in kontnih preglednic. Seveda to nikakor ne pomeni, da jih ne moremo analizirati. Bolje je konte preglednice odmakniti od izvornega sistema in jih pripraviti v obliki, ki bo sintaktično in strukturno sprejemljiva za branje z orodjem QlikView. Ob tem leži še dodatna prednost, da lahko znotraj aplikacije istočasno spremljamo več računovodskih izkazov različnih podjetij iz različnih informacijskih sistemov.

*Primer:*

<b>IU</b>	<b>PRIHODKI</b>	<b>Formula</b>	<b>111+115+118+155+160+163</b>
<b>110</b>	<b>A. ČISTI PRIHODKI OD PRODAJE</b>	<b>Formula</b>	<b>111+115+118</b>
<b>111</b>	<b>I. Čisti prihodki od prodaje proizvodov in storitev na dom. trgu</b>	<b>Formula</b>	<b>112+113+114</b>
112	1. Čisti prihodki od prodaje	GK konto	760003..760099
113	2. Čisti prihodki od najemnin	GK konto	
114	3. Čisti prihodki od prodaje blaga in materiala na domačem trgu	GK konto	
<b>118</b>	<b>II. Čisti prihodki od prodaje proizvodov in storitev na tujem trgu</b>	<b>Formula</b>	<b>119+120</b>
119	1. Čisti prihodki od prodaje proizvodov in storitev na tujem trgu	GK konto	761003..761099
120	1. Čisti prihodki od prodaje blaga in materiala na tujem trgu	GK konto	763003..763099

*Tabela 15: Struktura kontne preglednice za prihodke*

### 3.6 VSEBINSKI OPIS ETL POSTOPKA

Postopek lahko razdelimo na pet zaporednih korakov:

#### 1. Nastavitveni parametri

V tem koraku se preberejo lokalizacijske nastavitve, formati številck, nastavitvene datoteke, ki vsebujejo informacije o povezavi na izvorne podatkovne baze, inicializaciji spremenljivk itd.

## 2. Prebiranje kontnih preglednic

Ta korak je najzahtevnejši in zahteva kar nekaj zaporednih prebiranj in iteracij znotraj zank, da lahko iz formul izločimo simbole in na njihovi podlagi zgradimo hierarhično strukturo. V tabeli na najnižjem nivoju so prikazana le polja postavk ter njihova zgornja in spodnja meja konta kot v tabeli 18.

## 3. Prebiranje postavk dimenzij

QlikView skript
<pre> <b>IF</b> \$(NoOfDimensions) &gt;= 1 <b>THEN</b>    LedgerEntryDimension:   <b>LOAD</b> *;   <b>SQL SELECT</b> *   <b>FROM</b> \$(LedgerEntryDimension);    <b>FOR</b> nCount = 1 <b>To</b> \$(NoOfDimensions)     [Dim\$(nCount)_LED]:     <b>LOAD</b>     [Entry No_],     [Dimension Value Code] <b>as</b> [Dim\$(nCount)Code]     <b>RESIDENT</b> LedgerEntryDimension     <b>WHERE</b> [Dimension Code] = '\$(DimensionCode\$(nCount))';     <b>STORE</b> [Dim\$(nCount)] <b>INTO</b> [\$(DataFolderName)/Dim\$(nCount).qvd];     <b>DROP TABLE</b> [Dim\$(nCount)_LED];   <b>NEXT</b>   <b>DROP TABLE</b> LedgerEntryDimension; <b>END IF</b> </pre>

Tabela 16: Prikaz generičnega prebiranja dimenzij iz izvorne tabele 'LedgerEntryDimension'

Na primeru v tabeli 16 lahko vidimo način za prebiranje dimenzij iz pomožne tabele dimenzij. Število dimenzij in imena se nastavi v prvem koraku v nastavitveni datoteki. Na primeru vidimo, kako lahko s kontrolnimi stavki IF in FOR zanko postopno prebiramo dimenzijsko tabelo in ustvarimo za vsako dimenzijo posebej novo logično tabelo. V drugem delu se uporablja ukaz RESIDENT, ki naslavlja logično tabelo v spominu.

## 4. Prenos podatkov iz glavne knjige za omejeno obdobje

Postavka	Konto	Opis dokumenta	Datum	Povezava	Debetni znesek	Kreditni Znesek	Znesek
							0
4444	760010	P-R07/0887Račun P-R07/0887	31 08 2007	18	0,00	40833,34	-40833,34
4445	260010	P-R07/0887 Račun P-R07/0887	31 08 2007	18	0,00	8166,67	-8166,67
4446	120010	P-R07/0887 Račun P-R07/0887	31 08 2007	18	49000,01	0,00	49000,01

Tabela 17: Minimalen nabor atributov v tabeli glavne knjige, ki so potrebni za analizo

Ker so tabele glavne knjige običajno med največjimi, jih podrobno preberemo za obdobje dveh tekočih let. Podatke za predhodna leta se seštevajo na letni ravni za vsako konto posebej. Te zneski se potrebujejo zaradi začetnih in končnih stanj pri bilanci stanja. Obdobje preteklih let preberemo

samo enkrat in rezultat shranimo v QVD datoteko. V podobni obliki so zapisani tudi podatki o planih, čeprav se najpogosteje prebirajo kar neposredno iz Microsoft Excel preglednic.

## 5. Umeščanje analitičnih kontov v kontne razrede

Ko je hierarhija narejena in imamo postavke kontnih preglednic prebrane, izgleda struktura logične tabele 'Accounts' kot prikazano v tabeli 18. Pri vsakem zapisu vključuje poleg številke vrstice iz tabele 14 še spodnjo (TotalingAccount\_Lower) in zgornjo (TotalingAccount\_Upper) vrednost kontnega obsega. Za pravilen pregled moramo umestiti vse dejanske pripadajoče kontne številke iz glavne knjige znotraj intervala. Zato najprej preberemo vse enolične konte znotraj glavne knjige in jih nato s funkcijo INTEVALMATCH umestimo med intervale.

Prvotna struktura tabele 'Accounts' pred izvajanjem

RowNoAccount	TotallingLower	TotallingUpper
112	760003	760099

→

Ciljna povezovalna tabela

RowNoAccount	GL_Account_No
112	760005
112	760075
112	760068

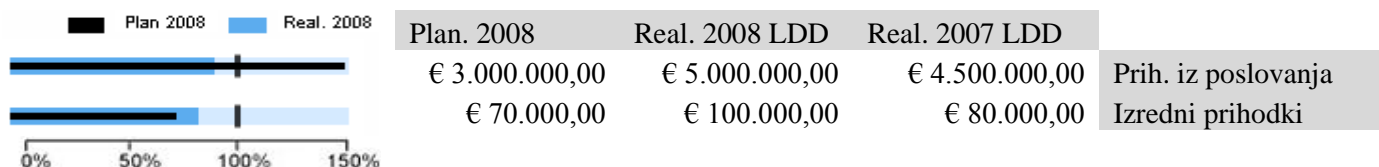
Tabela 18: Izvorna in končna struktura logične tabele 'Accounts', ki tvori asociacijsko povezavo na glavno knjigo

QlikView skript	Komentar
<b>GL_Entry_Account_List:</b> <b>LOAD</b> <b>DISTINCT</b> (GL_Account_No) <b>RESIDENT</b> GL;	Generiranje seznama enoličnih kontnih števil 'GL_Account_No' iz tabele 'GL'. Ukaz <b>RESIDENT</b> pove, da je GL že obstoječa logična tabela Glavne knjige, ki je bila že predhodno prebrana iz vira.
<b>LEFT JOIN</b> (Accounts) <b>INTERVALMATCH</b> (GL_Account_No) <b>LOAD</b> TotalingAccount_Lower, TotalingAccount_Upper <b>RESIDENT</b> Accounts; <b>DROP TABLE</b> GL_Entry_Account_List;	Seznam prebranih kontov z predpono <b>INTERVALMATCH</b> umestimo vse konte med zgornjo in spodnjo vrednostjo vsakega zapisa logične tabele 'Accounts'  Na koncu začasni seznam odstranimo.

Tabela 19: Skriptno zaporedje ukazov za transformacijo prikazano v tabeli 18

## 3.7 IZGRADNJA NADZORNE PLOŠČE Z ANALIZO MNOŽIC

Upoštevajoč definicijo nadzornih plošč v razdelku 2.7.1, da so nadzorne plošče vizualen prikaz zgolj najpomembnejših informacij za doseg poslovnih ciljev, je potrebno uporabiti visoko stopnjo agregacije in prikazati nekaj najosnovnejših računovodskih izkazov. Istočasno rezultate primerjati z lanskoletnimi in njihovimi predračuni. Orodje dopušča tudi vizualno gnezdenje grafikonov, kar dopušča optimalen prikaz vseh treh meritev.



Slika 30: Grafikon primerja prihodke iz poslovanja za obdobje dveh let in trenutnih planov. Za jasnejšo sliko so dodane tudi numerične vrednosti. Kratica LDD označuje obdobje od začetka trenutnega leta do danes.

Vsak element na sliki 20 je popolnoma svoj objekt (tekstovni ali grafični) in ga ne omejuje nobena dimenzija. Zato bi bil tak način prikaza težko izvedljiv brez analize množic. Pri izrazu  $'=sum(AmountGL)'$  bi vedno dobili vrednost 0 (bilančna vsota). Tu se pokaže zelo uporabna lastnost opisane analize množic v razdelku 2.8.6.1. Če nas npr. zanima zgolj letošnji prikaz prihodkov, si lahko to predstavimo na način, da nas zanima le vsota tiste podmnožice kontov, ki ima v kontni preglednici naslovno vrstico 110 (tabela 15).

$$sum(\{<RowNoAccount=\{(vRevenue)\}, Year=\{(vCyear)\}>\} AmountGL) \quad (6)$$

Rezultat je vsota prihodkov v letošnjem letu. Spremenljivka  $\{(vRevenue)\}$  vsebuje vrednost 110 po tabeli 15,  $\{(vCyear)\}$  pa trenutno leto, torej 2009.

## 3.8 ANALIZIRANJE RAČUNOVODSKIH IZKAZOV

### 3.8.1 Analiza s kazalniki

Poznamo kazalnike stanja financiranja, obračanja, kazalnike donosnosti itd. Niso vedno vsi primerni, odvisno so namreč od dejavnosti družbe. Smiselno je prikazati le povedne in obrazložljive. Dva pogosta kazalca dobičkonosnosti sta ROA (angl. Return On Assets) in ROE (angl. Return On Equity).



Kazalca dobičkonosnosti		Komentar
Od začetka leta 2008	Real. 2008 LDD	ROE ali donos na kapital = čisti dobiček / skupni kapital ROA ali donos na sredstva = čisti dobiček / vsa sredstva Kazalca izkazujeta operativno uspešnost družbe glede na sredstva in kapital.
	ROE 23,9%	
	ROA 16,3%	

Tabela 20: Analiza s kazalniki dobičkonosnosti

### 3.8.2 Vodoravna analiza

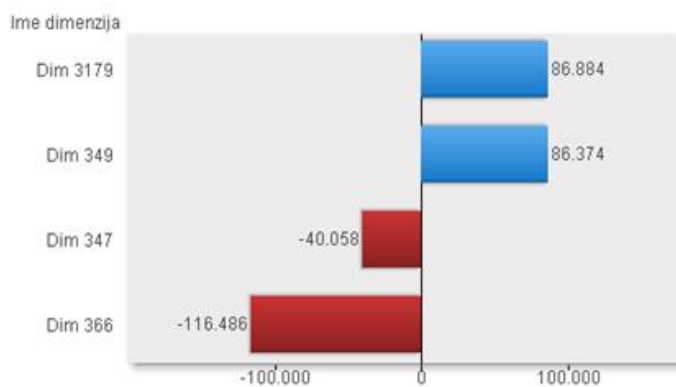
Najpogostejša analiza računovodskih izkazov je vodoravna analiza. Z njo se ugotavljajo vrednostni zneski in odstotki sprememb določenih postavk v računovodskih izkazih. Z njimi dobimo informacije o dejanski in relativni velikosti odstopanj.

Skupina IU	Real. 2008 LDD	Real. 2007 LDD	Varianca	Razlika
<b>Total</b>	<b>219.488,49</b>	<b>295.436,71</b>	<b>-25,71%</b>	<b>-75.948,22</b>
<b>PRIHODKI</b>	<b>1.312.812,14</b>	<b>1.237.452,27</b>	<b>6,09%</b>	<b>75.359,87</b>
I. Čisti prihodki od prodaje proizvodov in storitev na domačem trgu ...	1.115.038,58	1.006.077,20	10,83%	108.961,38
863962 Ysqd ju brhj.wjwvzvur dihb	1.115.038,58	1.006.077,20	10,83%	108.961,38
I. Finančni prihodki iz deležev (156 do 159)	1.293,91	880,90	46,89%	413,01
II. Čisti prihodki od prodaje proizvodov in storitev na tujem trgu	196.479,65	228.627,68	-14,06%	-32.148,03
806590 Dkum, jm kjeu.oedkaemw xriji nvo	196.479,65	228.627,68	-14,06%	-32.148,03
II. Finančni prihodki iz danih posojil (161+162)	0,00	1.866,49	-100,00%	-1.866,49
833812 Xjhggtj vf sstmz kuzcksvw	0,00	1.866,49	-100,00%	-1.866,49
<b>STROŠKI BLAGA, MATERIALA IN STORITEV</b>	<b>-471.900,67</b>	<b>-508.905,34</b>	<b>-7,27%</b>	<b>37.004,67</b>
I. Stroški blaga, materiala in storitev (129+130+134)	-471.900,67	-508.905,34	-7,27%	37.004,67
<b>STROŠKI DELA, ODPISI VREDNOSTI IN DRUGI FIN. ODHODKI</b>	<b>-621.422,98</b>	<b>-433.110,22</b>	<b>43,48%</b>	<b>-188.312,76</b>
I. Drugi odhodki	0,00	-0,01	-100,00%	0,01
486905 Dpsje.opc.vbnlbc ktyhmsijy-gtp.mrw	0,00	-0,01	-100,00%	0,01
II. Finančni odhodki iz finančnih obveznosti (170+173)	-3.503,09	-3.963,42	-11,61%	460,33
216330 Oxwplsw os syzfq rekqul Qyaz	-3.490,39	-3.963,42	-11,93%	473,03
356581 Acatfux pg hdxvdsq abymfqs sg yownzst	-12,70	0,00	-	-12,70
II. Stroški dela (140 do 143)	-615.391,00	-425.672,59	44,57%	-189.718,41
042219 Tuzikg - mdkuimsyl	-21.248,94	0,00	-	-21.248,94
306291 Zeapeympm ju THI.R.	-27.217,65	-21.512,15	26,52%	-5.705,50
417497 Apkpk ok snwcjzru geod'a	-9.624,42	-13.645,83	-29,47%	4.021,41
569173 Pjovab - hsd0	-66.449,43	0,00	-	-66.449,43
603052 Eyeqrfsx wa tvffsjhwvdc	-249,03	-196,83	26,52%	-52,20
618891 Qpptpkd wfi akocz	-17.268,18	-15.916,42	8,49%	-1.351,76
645359 Smbuug nl aodc od yi avplu	-16.718,10	-13.924,57	20,06%	-2.793,53
645431 Ptoov.gh yaqjebvlv dzw vcgq	-2.199,00	-1.738,05	26,52%	-460,95
671477 Uulfgwyxm ly DIEW	-36.718,94	-29.021,59	26,52%	-7.697,35
979653 Ygyif wr dognkxmd	-2.379,42	-1.461,12	62,85%	-918,30
981798 Vqqig wkclbtxpauo-zkdgb bjur	-414.902,98	-327.927,97	26,52%	-86.975,01
990713 Xppon.al iswzuvqim nyoogay	-414,91	-328,06	26,47%	-86,85
III. Finančni odhodki iz poslovnih obveznosti (175+177)	-208,89	-957,20	-78,18%	748,31
IV. Drugi poslovnih odhodki (149+150)	-2.320,00	-2.517,00	-7,83%	197,00

Tabela 21: Primer vodoravne analize izkaza uspeha

### 3.8.3 Stroškovni ali prihodkovni odmiki

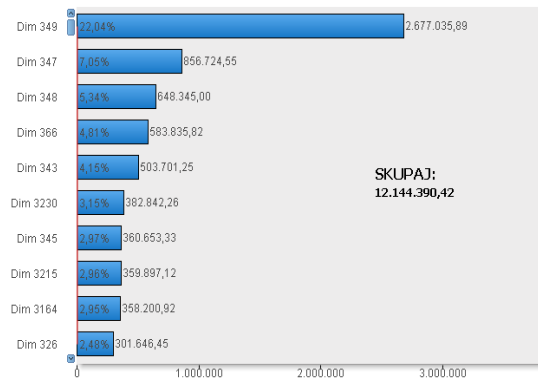
Ena od zelo popularnih metod za preverjanje odstopanj prihodkov ali stroškov, je razlika dejanskih in planiranih vrednosti. Spodnji grafikon predstavlja razliko po dimenzijah med nastalimi in predvidenimi prihodki. Pozitivna vrednost pomeni razliko v korist dejanskih prihodkov v primerjavi s predvidenimi.



Slika 31: Prikaz odstopanj med dejanskimi in planiranimi prihodki

### 3.8.4 Navpična analiza

Podobno kot pri vodoravni analizi se v tem pregledu proučujejo spremembe, vendar je tu poudarek na relativni pomembnosti posameznih postavk. Primerjamo lahko prihodke, stroške, dimenzije itd.

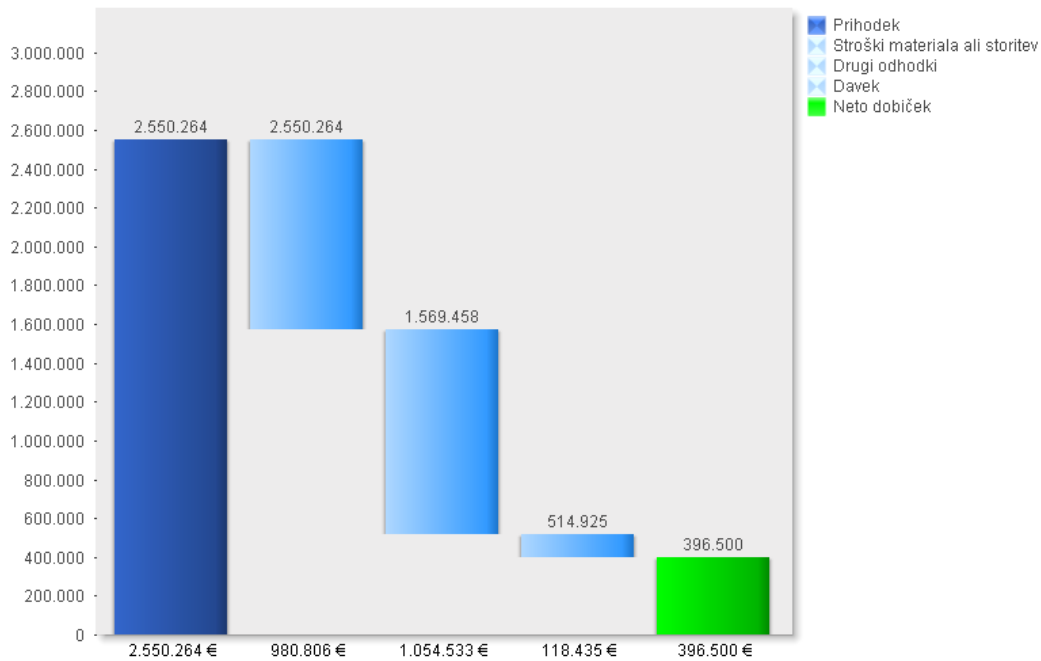


Slika 32: Prikaz prihodkov po dimenzijah

Stroškovo mesto	2008	2008 v %	2008 v %
Dim 15	624.817,17	60,63%	
Dim 12	27.115,87	2,63%	
Dim 14	<b>86.335,68</b>	<b>8,38%</b>	
41 Stroški storitev			
414 Povračila stroškov zaposlenim v zvezi z delom ...	6.598,82	0,64%	
416 Stroški intelektualnih in osebnih storitev	64.556,07	6,26%	
417 Stroški sejmov, reklame in reprezentance (41...	3.166,00	0,31%	
419 Stroški drugih storitev (komunalna, čiščenja pr...	12.014,79	1,17%	
Dim 16	132.198,85	12,83%	
Dim 116	6.871,40	0,67%	
Dim 118	4.643,25	0,45%	
Dim 122	<b>36.026,20</b>	<b>3,50%</b>	
41 Stroški storitev			
416 Stroški intelektualnih in osebnih storitev	34.617,58	3,38%	
417 Stroški sejmov, reklame in reprezentance (41...	1.408,62	0,14%	
Dim 126	105.885,18	10,28%	
Dim 131	2.341,90	0,23%	
Dim 132	1.519,19	0,15%	
Dim 134	<b>723,92</b>	<b>0,07%</b>	
41 Stroški storitev			
414 Povračila stroškov zaposlenim v zvezi z delom ...	723,92	0,07%	
Dim 137	2.016,59	0,20%	

Slika 33: Prikaz stroškov po stroškovnih mestih

### 3.8.5 Drugi načini prikaza



Slika 34: Primer grafikona za izkaz uspeha v določenem obdobju

## 4 SKLEPNE UGOTOVITVE

### 4.1 ZAKLJUČEK

S popolnoma drugačnim načinom pristopa zbiranja, združevanja in analiziranja podatkov je orodje QlikView odgovor na najvidnejše probleme izgradnje sistemov za poslovno obveščanje. Podatkovna skladišča se zaradi splošne uporabljivosti še vedno implementirajo, vendar niso več nujen predpogoj za analiziranje. QlikView hrani podatke v asociativnem podatkovnem modelu, ki predstavlja zelo optimalno strukturo za večdimenzionalno analizo v spominu na uporabnikovo zahtevo. Tudi novejša implementacije koncepta OLAP želijo uporabnikom postreči z bolj odzivnejšimi načini, tako da vse večji del analitičnih kock shranjujejo v pomnilnik.

Problem še vedno nastane zaradi omejene količine spomina. Ravno zato lahko trdimo, da je AQL korak pred ostalimi produkti. Pokazali smo, da lahko s svojo strukturo zasede le 15 do 20 odstotkov podatkovnega prostora, kot ga za isto količino potrebuje relacijska oblika. Pri izdelavi aplikacije za finančni kontroling se je orodje QlikView izkazalo za dobro odločitev. Znotraj ETL postopka je bilo mogoče s pomočjo kontrolnih stavkov uspešno prebrati strukturo kontnih preglednic in jo preoblikovati v hierarhično obliko znotraj asociativne logike. Ker zna orodje brati različne vire (datoteke in podatkovne baze), nam odpira možnost primerjalnega analiziranja na nivoju več družb hkrati.

Cena implementacije je najbolj odvisna količine podatkov, števila uporabnikov in potrebnih prilagoditev. Če imamo opravka z manjšo družbo, katere skupni obseg podatkov ne presega približno 1.000.000 zapisov in je končni uporabnik samo ena oseba, potem lahko aplikacijo poganja že navadna delovna postaja. Pri večjem številu uporabnikov in izvajanju sočasnih analiz, se zaradi vsebinske konsistentnosti implementira strežniška arhitektura.

QlikView strežnik poskrbi za procesiranje in pomnilniški prostor, QlikView Publisher pa za ustrezno distribucijo, osveževanje podatkov, varnostne mehanizme ipd. Stopnja funkcionalnosti, ki jo orodje ponuja, je omejena glede na tip uporabniške licence. Zaradi zagotavljanje najvišje stopnje varnosti se v praksi varnostna zaščita opravi že na nivoju datoteke (razdelek 2.6).

### 4.2 IZBOLJŠAVE PRI APLIKACIJI ZA FINANČNI KONTROLING

Pri aplikaciji poslovnega obveščanja bi težko popolnoma standardizirali poglede. Glavna prednost sistema poslovnega obveščanja je možnost naprednih in predvsem prilagodljivih načinov prikazovanja. Zaradi želje po vedno hitrejšemu vpeljevanju končnega uporabnika, želimo zmanjševati število prilagojenih nastavitev in spodbujati uporabnike k tem dejanju. V računovodstvu obstajajo pravila in pregledi, ki se ne spreminjajo zelo pogosto. Zato je cilj osnovne verzije vsebovati preglede, ki so bolj ali manj standardni za različne družbe.

Aplikacija je nastajala z verzijo orodja 8.0, ko orodje še ni omogočalo tako elegantne izgradnje hierarhij, kot jo ponuja trenutna verzija 8.5. Zato se bo struktura izvornih kontnih preglednic prilagodila na obliko tabele sosednjih vozišč (razdelek 2.4.2.9). S tem bo ETL postopek bolj pregleden za vzdrževanje.

### 4.3 POSLOVNO OBVEŠČANJE IN SOA

SOA (angl. Service Oriented Architecture) je sestavljena iz množice različnih komponent, ki omogoča popolnoma različnim tipom virov njihovo medsebojno interakcijo in s tem izboljšujejo produktivnost vedno bolj kompleksnih poslovnih procesov. Organizacije se že veliko časa ukvarjajo s standardizacijo poslovnih procesov, ki s seboj prinašajo tudi veliko togosti in utesnjenosti. Eden izmed boljših načinov kako to narediti je SOA. Skrajšano rečeno, razbito strukturo (lokacijsko in vsebinsko) aplikacij, podatkovnih baz, podatkovnih skladišč, človeških virov, internetnih virov želimo gledati kot celoto, se iz nje učiti, jo spreminjati, optimizirati. SOA združuje različne tehnologije.

Poslovno obveščanje je sistem, ki združuje, transformira in prikazuje podatke različnih virov na poveden način, da iz njega lahko pridobivamo znanje. S poslovnim obveščanjem na enostavnejši način pridemo do informacij, ki bi jih s klasičnimi metodami (npr. SQL) težko izluščili. Skupna prednost obeh leži v tem, da znata uporabljati že obstoječe sisteme. Zato je zaradi vedno kompleksnejših problemov prihodnosti pričakovati tudi vedno več povezovanja in prepletanja teh dveh tehnologij. Izvozni rezultati analiz poslovnega obveščanja bi lahko služili kot vir znotraj SOA procesa in na drugi strani se lahko poslovno obveščanje uporablja za analiziranje SOA procesov.

Čeprav se pri nas zanimanje za SOA morda še ne kaže oz. ni tako priljubljeno kot poslovno obveščanje, bodo sčasoma tudi njegovi uporabniki začeli razmišljati o SOA in obratno.

### 4.4 ORODJE QLIKVIEW V PRIHODNOSTI

Svetovno priznana analitska hiša Gartner uvršča orodje QlikView na presečišče vodilnih in vizionarskih platform za poslovno obveščanje. Njegovi glavni izpostavljeni prednosti v primerjavi z ostalimi produkti sta po pričakovanjih njegov zelo zmogljiv asociativni podatkovni model in analiza množic, ki jo poznamo od verzije 8.5 dalje.

Gledano kritično orodju manjkajo predvsem zmožnosti statističnega napovedovanja in večje število implementacij pri velikih organizacijah. Seveda lahko rečemo, da je bilo orodje do sedaj precej omejeno s količino pomnilnika. Ker pa danes pri strežnikih 64-bitna arhitektura procesorjev ni več nobena redkost, lahko napovemo, da se bo v prihodnosti vedno večji del podatkov hranil v spominu. Vse te strojne kapacitete orodje zelo dobro izkorišča, saj je že s prihajajočo verzijo QlikView 9 odstranjena omejitev največjega števila prebranih zapisov, ki je trenutno omejena na dve milijardi.

## 5 LITERATURA

- [1] CELKO Joe: SQL for Smarties, third edition, Morgan Kaufman Publishers, 2005
- [2] E.F. Codd: Providing OLAP to User\_Analysts: An IT Mandate.  
Dostopno na: [http://www.cs.bgu.ac.il/~dbm031/dw032/Papers/olap\\_to\\_useranalysts\\_wp.pdf](http://www.cs.bgu.ac.il/~dbm031/dw032/Papers/olap_to_useranalysts_wp.pdf)
- [3] EXCELSIOR Software: The District Data Analyzer: A Low-Cost, High-Impact Alternative to a Data Warehouse. Dostopno na: <http://www.blegroup.com>
- [4] FEW Stephen: Information Dashboard Design, O'Reilly, 2006
- [5] FEW Stephen: Show Me the Numbers, Analytics Press, 2004
- [6] GARTNER: Magic Quadrant for Business Intelligence platforms.  
Dostopno na: <http://www.gartner.com>
- [7] GOLEC, Darko: Vpeljava sistema za poslovno obveščanje v podjetju Mariborska Livarna, Ljubljana 2008
- [8] HOČEVAR Marko, IGLIČAR Sandi, ZAMAN Maja: Osnove računovodstva, Ekonomska fakulteta v Ljubljani, 2000
- [9] Internationaler Controller Verein: Kaj je kontroling?  
Dostopno na: [http://www.controllerverein.de/Kaj\\_je\\_kontroling.134145.html](http://www.controllerverein.de/Kaj_je_kontroling.134145.html)
- [10] KOŽUH, Boštjan, OLAP s hitrosjo misli, Dnevi Slovenske Informatike, Portorož 2008
- [11] LARSON Brian, Delivering Business Intelligence with Microsoft SQL Server 2005, McGraw-Hill 2006
- [12] LOSHIN David: Business Intelligence, Getting Onboard with Emerging IT, Morgan Kaufman Publishers, 2003
- [13] Qliktech: QlikView 8.5 Reference Manual .  
Dostopno na: <http://ftp.qliktech.com/qvwebdownloads/Docs.asp>
- [14] Qliktech: QlikView Data Models for PeopleSoft.  
Dostopno na: <http://www.QlikView.co.za/downloads/PeopleSoftDM.pdf>
- [15] Qliktech: QlikView Publisher Reference Manual.  
Dostopno na: <http://ftp.qliktech.com/qvwebdownloads/Docs.asp>
- [16] Qliktech: QlikView Server Reference Manual,  
Dostopno na: <http://ftp.qliktech.com/qvwebdownloads/Docs.asp>
- [17] RTOLAP - Real Time OLAP, Wikipedia.  
Dostopno na: <http://en.wikipedia.org/wiki/Rtolap>
- [18] T.MOSS Larisa, ATRE Shaku: Business Intelligence Roadmap, Addison-Wesley, 2003
- [19] WHITE, Colin: What Do SOA and ESB Mean in Business Intelligence.  
Dostopno na: <http://www.b-eye-network.com/view/3018>
- [20] WILLIAMS Simon: The Associative Model of Data, Second edition.  
Dostopno na: [http://www.lazyssoft.com/docs/other\\_docs/AMD.pdf](http://www.lazyssoft.com/docs/other_docs/AMD.pdf)
- [21] WIKIPEDIA: Busniness intelligence.  
Dostopno na: [http://en.wikipedia.org/wiki/Business\\_intelligence](http://en.wikipedia.org/wiki/Business_intelligence)

## 6 SEZNAM TABEL IN SLIK

Slika 1: Razvoj skozi čas .....	7
Slika 2: Pet razvojnih korakov klasičnega koncepta.....	10
Slika 3: Trije razvojni koraki inovativnega koncepta .....	11
Tabela 1: Tabela entitet.....	12
Tabela 2: Tabela asociacij.....	12
Tabela 3: Entitete .....	13
Tabela 4: Asociacije.....	13
Slika 4: Zapisani dogodki v relacijski obliki .....	13
Slika 5: Asociacijska oblika zapisa v QlikView .....	14
Slika 6: Reakcija ob izbiri študenta 2233.....	14
Slika 7: Statistični prikaz tabele RP_PROMET v MS SQL podatkovni bazi.....	14
Slika 8: Statistični prikaz tabele RP_PROMET v AQL podatkovni bazi.....	14
Tabela 5: Primer polnjenja iz ODBC vira tabele 'Baze2008.promet' .....	16
Tabela 6: Primer predpone ADD .....	17
Tabela 7: Primer procedure.....	18
Slika 9: Izvorna struktura podatkov .....	18
Slika 10: Ciljna struktura podatkov .....	18
Tabela 8: Primeri uporabe zank FOR .....	19
Slika 11: Prikaz tabele sosednjih vozlišč(vhodna tabela) .....	20
Slika 12: Prikaz razširjene tabele(rezultat) .....	20
Slika 13: Prikaz premostitvene tabele .....	20
Slika 14: Drevesna struktura .....	20
Tabela 9: Naročila .....	21
Tabela 10 : Opisi dogodkov .....	21
Tabela 11 : Izvršeni ukazi .....	21
Slika 15: Rezultat.....	21
Slika 16: Enostaven podatkovni model.....	23
Slika 17: Sistemska matrika asociacij .....	23
Slika 18: Krožne reference.....	24
Slika 19: Tabela hkratnih dogodkov .....	24
Tabela 12: Overitev in avtorizacija uporabnikov na nivoju aplikacije .....	25
Slika 19: Izgubljena izraznost namenjene informacije. ....	28
Slika 20: Poudarek odsopanja od ciljne vrednosti. ....	28

Slika 21: Nastavitveno polje za pogojni prikaz .....	29
Slika 22: Seznam objektov v delovnem zvezku.....	29
Slika 23: Urejevalnik makrojev .....	30
Tabela 13: Primeri modifikatorjev pri analizi množic .....	32
Slika 23: Prikaz množic (3).....	32
Slika 24: Seznam prodajalcev in strank .....	33
Slika 25: Štetje strank .....	33
Slika 26: Uporaba funkcije Aggr( ).....	33
Slika 27: Splošne nastavitve upravljaljskega uporabniškega vmesnika .....	35
Slika 28: Komunikacijska struktura med strežnikom, spletnim strežnikom in odjemalci.....	38
Tabela 14: Primerjava različnih QV odjemalcev .....	39
Slika 29: Skica izvajalnega procesa QlikView Publisher .....	40
Tabela 15: Struktura kontne preglednice za prihodke .....	43
Tabela 16: Prikaz generičnega prebiranja dimenzij iz izvorne tabele 'LedgerEntryDimension' .....	44
Tabela 17: Minimalen nabor atributov v tabeli glavne knjige, ki so potrebni za analizo .....	44
Tabela 18: Izvorna in končna struktura logične tabele 'Accounts'.....	45
Tabela 19: Skriptno zaporedje ukazov za transformacijo prikazano v tabeli 18 .....	45
Slika 30: Grafikon primerja prihodke iz poslovanja za obdobje dveh let in trenutnih planov .....	46
Tabela 20: Analiza s kazalniki dobičkonosnosti.....	46
Tabela 21: Primer vodoravne analize izkaza uspeha .....	47
Slika 31: Prikaz odstopanj med dejanskimi in planiranimi prihodki .....	47
Slika 32: Prikaz prihodkov po dimenzijah.....	48
Slika 33: Prikaz stroškov po stroškovnih mestih .....	48
Slika 34: Primer grafikona za izkaz uspeha v določenem obdobju .....	48

## **IZJAVA O AVTORSTVU**

Izjavljam, da sem diplomsko delo izdelal samostojno pod vodstvom mentorja doc. dr. Marka Bajca. Izkazano pomoč drugih sodelavcev sem v celoti navedel v zahvali.

Anže Mis