

## ARKITEKTURA E NJË SISTEMI *BUSINESS INTELLIGENCE*

\*KIKINA DH.<sup>1</sup>, META O.<sup>2</sup>, XHINA E<sup>3</sup>.

<sup>1,2</sup>Banka Credit AgricoleShqipëri, Divizioni i IT-së

<sup>3</sup>Universiteti i Tiranës, Fakulteti i Shkencave të Natyrës, Departamenti i Informatikës

e-mail: dkikina@yahoo.com

### Përmbledhje

*Business Intelligence* (BI) po kthehet në një domosdoshmëri për shumë organizata të cilat ruajnë sasi të mëdha të dhënash në sistemet e tyre dhe përdorin shumë sisteme gjithashtu. Ajo po konfirmohet si një teknologji e cila impakton vendimarrjen e menaxhimit duke e përmirësuar atë. Nëpërmjet kësaj teknologjie organizata e ka më të lehtë të kuptojë situatën e biznesit dhe të planifikojë të ardhmen duke ndërtuar strategji efektive. Por megjithëse gjithmonë e më shumë kompani po vendosin ta përdorin këtë teknologji, jo të gjitha janë në gjendje të marrin përfitimet e duhura prej saj. Një ndër arsyt kryesore është ndërtimi jo i saktë i arkitekturës për BI. Në këtë artikull do të fokusohemi në shpjegimin e një arkitekture të plotë të një sistemi BI, duke përshkruar komponentet, teknologjitë apo koncepte të përdorura në të. Në pjesën e fundit paraqitet rasti i implementimit të kësaj arkitekture në një bankë.

### Abstract

Business Intelligence (BI) is becoming a necessity for many organizations which store big volumes of data in their systems and use many systems as well. It is confirmed as a technology which impacts the managerial decision taking by improving it. Through this technology it is easier for the organization to understand the business situation and to plan the future by building effective strategies. However, even though more and more companies are deciding to use this technology, not all of them are able to take the proper benefits out of it. One of the main reasons is the inappropriate design of the BI architecture. In this article we will focus on explaining a complete architecture for a BI system, by describing the components, technologies and concepts used. The last part presents as case study the implementation of this architecture in a bank.

**Fjalëkyçe:** *Business Intelligence*, Arkitekturë BI, ETL, Datawarehouse, OLAP, Data mining.

### 1. Hyrje

*Business Intelligence* është mënyra se si të dhënat, të cilat janë një nga asetet më të çmuara të organizatës, ruhen, aksesohen, analizohen dhe kthehen në një informacion të vlefshëm për të përmirësuar performancën e biznesit (Azvine, et al., 2005).

Për shkak të rëndësisë që këto sisteme kanë për kompanitë, gjithmonë e më shumë po investohet në këtë fushë. Sipas *Market Research.com* tregu i produkteve BI mendohet të arrijë në 11.3 bilion dollarë nga viti 2015, (MarketResearch.com, 2010).

Aplikimi i një sistemi të tillë në një organizatë kërkon një plan të mirëpërcaktuar, duke qenë se BI ka një arkitekturë e cila përfshin shumë teknologji, komponente, procese dhe kërkon një analizim shumë të mirë të secilës prej tyre dhe mënyrës se si këto komponente do të lidhen apo ndërveprojnë me njëra tjetrën. Dështimi apo suksesi i një sistemi të tillë, i cili ka një kosto relativisht të lartë, varet pikërisht nga dizenjimi i kësaj arkitekture. Një arkitekturë jo e mirë e një sistemi të tillë do të sillte probleme jo vetëm në performance, por edhe në informacionin që do të analizonte dhe përcillte tek përdoruesit. Një sistem BI që nuk është në gjendje të përcjellë informacionin e duhur te përdoruesit nuk mund të konsiderohet i suksesshëm.

Për një teknologji kaq komplekse kuptohet që është e vështirë që të jepet një kornizë fikse e arkitekturës së saj, e cila do të na garantonte suksesin e implementimit. Megjithatë ekzistojnë modele të kësaj arkitekture, ndjekja me rigozitet e të cilave na drejton drejt një suksesi të mundshëm. Në literaturat që trajtojnë këtë temë, mund të gjejmë modele të ndryshme të arkitekturës së një sistemi BI, (Turban, 2008; Baars, & Kemper, 2008; Balaceanu, 2007). Këto arkitektura ndryshojnë në strukturat, komponentet, proceset, apo lidhjet e pjesëve ndërmjet tyre, për të formuar një sistem të plotë BI. Megjithatë edhe pse këto arkitektura ndryshojnë, ekziston një strukturë e përgjithshme e cila është e njëjtë në të gjitha këto modele.

Në këtë material do të përshkruhet pikërisht kjo arkitekturë e përgjithshme e një sistemi BI me të gjitha komponentet e saj dhe në fund të materialit do të jepet një rast konkret i implementimit të kësaj arkitekture.

## 2. Arkitektura e një sistemi *Business Intelligence*

Në këtë pjesë do të përshkruhen shtresat e një arkitekture të përgjithshme të një sistemi *Business Intelligence*. Siç u tha më sipër, nuk ekziston një strukturë fikse e një sistemi të tillë, megjithatë ekzistojnë disa komponente apo elemente bazë që, pavarësisht tipit të teknologjisë që një kompani përdor, janë të njëjta kudo.

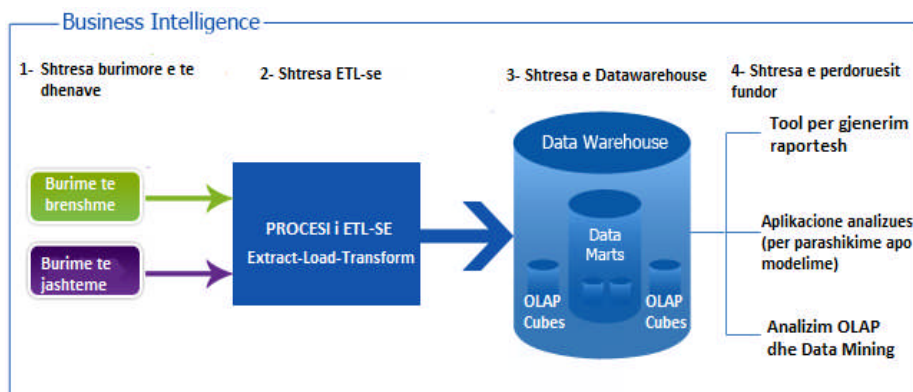


Figura 1. Arkitektura e një sistemi BI

### 1- Shtresa burimore e të dhënave

Kjo shtresë përbëhet nga të dhënat që ruhen në sistemet operacionale të kompanisë. Të dhënat mund të ruhen në databazat (ose në struktura të tjera të dhënash) të sistemeve të brendshme, ose në databazat e sistemeve të jashtme. Këto sisteme ndryshe quhen edhe *OLTP (Online Transactions Processing Systems)*.

Sistemet operacionale të brendëshme ruajnë të dhëna të cilat përdoren për të suportuar punët e përditshme të sistemit. Ndërsa sistemet e jashtme ruajnë të dhëna që ruhen jashtë kompanisë por që i transferohen kompanisë nëpërmjet kanaleve të komunikimit.

### 2- Shtresa e ETL-së

Kjo shtresë përbëhet nga tre procese kryesore: ekstraktimi, transformimi dhe ngarkimi i të dhënave.

Kjo shtresë është shumë e rëndësishme përse i përket ndërtimit të një sistemi efikas BI. Procesi *ETL* konsumon deri në 80 % të burimeve të zhvillimit (Inmon, 2002). Gjatë këtij procesi të dhënat ekstrahohen nga shtresa e parë, më pas transformohen duke krijuar struktura të përshtatshme për analizim apo për raportim dhe më pas ngarkohen në sisteme si psh *datawarehouse*, *datamarts* ose aplikacione analizuese.

### 3- Shtresa Datawarehouse

Të dhënat të cilat do të ekstrahohen nga sistemet *OLTP* do të kalohen më pas në *datawarehouse*. *Datawarehouse* është një databazë ku të dhënat ruhen në mënyrë të tillë që përdoruesit mund të nxjerrin raporte apo mund të bëjnë analiza mbi të dhënat.

Një *datamart* është një *datawarehouse* me përmasa më të vogla e cila fokusohet në një fushë interesi (psh *datamart* për Riskun, *datamart* për Financën ose për Marketing etj). Një *datawarehouse* mund të ndahet në disa *datamart*-e në varësi të kërkesave të përdoruesve dhe të ndryshimeve që ekzistojnë te këto kërkesa.

Mbi *datawarehouse* mund të ndërtohen struktura *OLAP* (kubë *OLAP*) të cilat janë më të përshtatshme për të analizuar të dhënat.

### 4- Shtresa e përdoruesit fundor

Shtresa e përdoruesit fundor konsiston në një sere mjetesh (*tools*) të cilat e shfaqin informacionin e ruajtur më sipër në formate të ndryshme tek përdoruesit. Mjetet që mund të përdoren janë të tipit:

- Gjenerim raportesh
- *Olap/Datamining*
- Vizualizim të dhënash (*dashboard* apo *scorecard*)
- Analizues (mjete që përdoren për modelime, parashikime, apo për të realizuar skenarë *what-if*)

Secili nga këto mjete e paraqet informacionin në nivele dhe në forma të ndryshme. Aplikacionet analizuese janë ato që përdoren më shpesh nga menaxheret të cilët marrin vendime, ndërsa mjetet e tjera janë më të përdorshme nga menaxherët operacionalë.

### 3. Rast Studimi

Në këtë pjesë do të paraqesim rastin e implementimit të një arkitekture BI në një bankë, e cila ka më shumë se 10 vjet që operon në tregun bankar dhe gjatë këtyre viteve, si sasia e të dhënave që janë ruajtur në sistemet e saj, ashtu dhe kanalet e komunikimit të kësaj banke me klientët janë rritur.

Në një treg tepër konkurses si tregu bankar, ku bankat po ofrojnë produkte sa më inovative duke synuar jo vetëm klientelë të re por dhe duke kërkuar fuqizimin e marrëdhënies me klientët ekzistues, nevoja e menaxhimit, por edhe punonjësve në front të parë, për sa më shumë informacion është shumë e qartë. I gjithë ky informacion ndodhet në të dhënat që ruhen në databazat e saj.

Faktorët e përmendur më sipër bënë që banka të iniciojë një projekt për të ndërtuar një sistem *Business Intelligence*, i cili do të mund të suportonte menaxhimin me informacionin e nevojshëm për një vendimarrje sa më efektive. Ky projekt përfshinte dizejnimin e një *datawarehouse*, ndërtimin e një *ETL* që do të popullonte këtë *datawarehouse* me të dhëna nga sistemi i saj transaksional apo sistemeve të tjera periferike, si dhe paraqitjen e të dhënave me anë të mjeteve të përmendura më sipër që do t'i paraqisnin informacionin menaxhimit të cdo niveli në një formë sa më të qartë.

Platforma mbi të cilën u ngrit sistemi BI u vendos të ishte në *SQL Server 2008*. Ky vendim u mor pasi kjo platformë u vlerësua që ofronte zgjidhjen më të mirë për buxhetin që u vendos të shpenzohej.

Më poshtë do të shpjegojmë shtresat e arkitekturës në rastin e implementimit të këtij sistemi, si dhe lidhjet e këtyre shtresave me njera tjetrën.

#### 1- Shtresa burimore e të dhënave

Shtresa e parë ose ajo e të dhënave burimore në këtë bankë përfshin të gjitha të dhënat që procesohen në sistemin qendror bankar. Këto të dhëna ruhen në databaza relacionare në *Oracle*. Ruajtja e të dhënave në këtë format nuk është e favorshme për sa i përket gjenerimit të infomacionit e për këtë arsye ato do të transportohen në një *datawarehouse* në *SQL Server 2008*. Databaza në *Oracle* ruan një numër të madh tabelash, por jo të gjitha prej tyre ruajnë informacion i cili është i nevojshëm për t'u analizuar.

Një sistem tjetër i cili ruan të dhëna të cilat janë të nevojshme të ruhen në *Datawarehouse* është dhe *Internet Banking*. *Internet Banking* është një nga sistemet e informacionit më të rëndësishme të bankës, krahas sistemit qendror bankar, gjë që e bën të nevojshme analizën e të dhënave të tij nga shumë këndvështrime, si p.sh. aktiviteti transaksional, siguria, etj. Një pjesë e të dhënave të këtij sistemi, i cili është i jashtëm, i kalojnë bankës nëpërmjet

skedarëve text. Pra të dhënat që vijnë nëpërmjet këtij sistemi mund të konsiderohen të jashtme (externale).

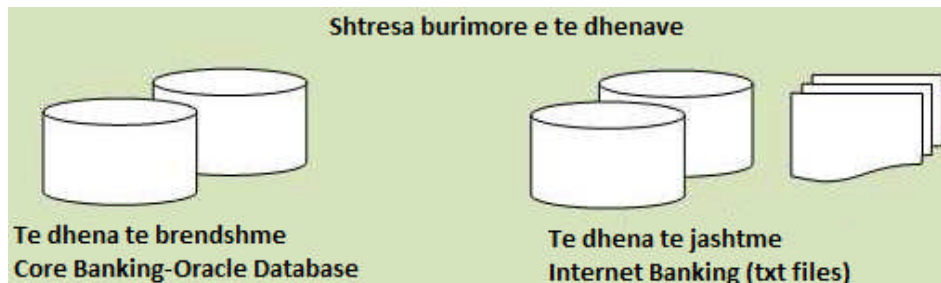


Figura 2. Shtresa burimore e të dhënave

## 2- Shtresa e ETL-së

Të dhënat që ndodhen në shtresën e parë do të transferohen në *datawarehouse* nëpërmjet një procesi *ETL*-je. Këto të dhëna do të ekstrahohen, do të përpunohen dhe do të ngarkohen më pas në *datawarehouse*.

Për të ekzekutuar këto hapa është përdorur *SQL Server Integration Services (SSIS)* dhe një sërë instancash në databazën e *SQL Server*, të cilat ekzekutohen gjatë fazave të ndryshme të procesit *ETL (Extract-Transform-Load)*.

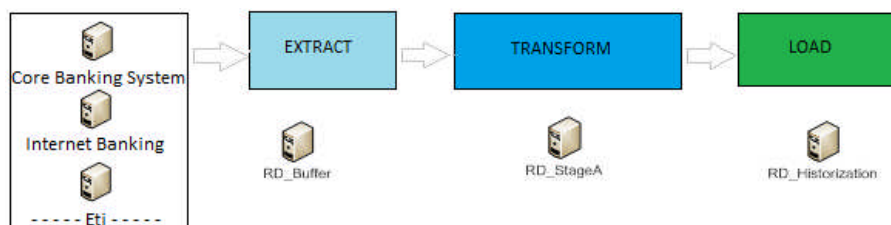


Figura 3. Diagrama e një procesi ETL

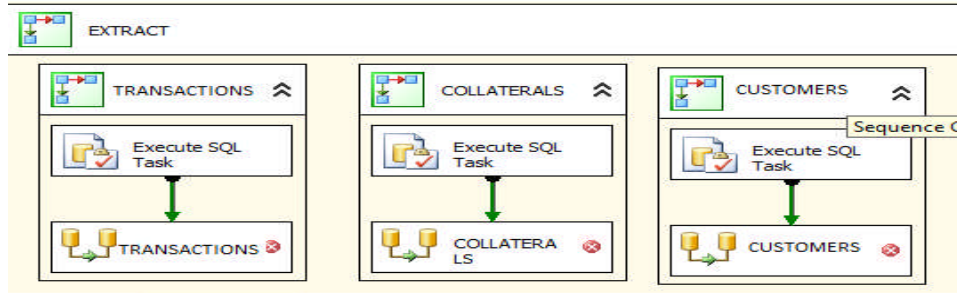
Pjesët përbërëse të këtij procesi janë si më poshtë:

### Ekstraktimi (*Extract*)

Hapi i parë i këtij procesi është ekstraktimi i të dhënave nga shtresa e parë në një database *SQL*. Numri i tabelave që do të ekstrahohen varet nga të dhënat që do të nevojiten, për të cilat është bërë një analizë paraprake me të gjitha njësitë e biznesit. Në këtë fazë tabelat importohen me të njëjtën strukturë dhe emërtime si në shtresën e parë. Procesit i ekstraktimit përbëhet nga një sere nën-procesesh, ku secili prej tyre përfaqëson një entitet raportimi. Operacionet që ndodhin në secilin nga këto nën-procese mund të përgjithësohen në këto tre hapa:

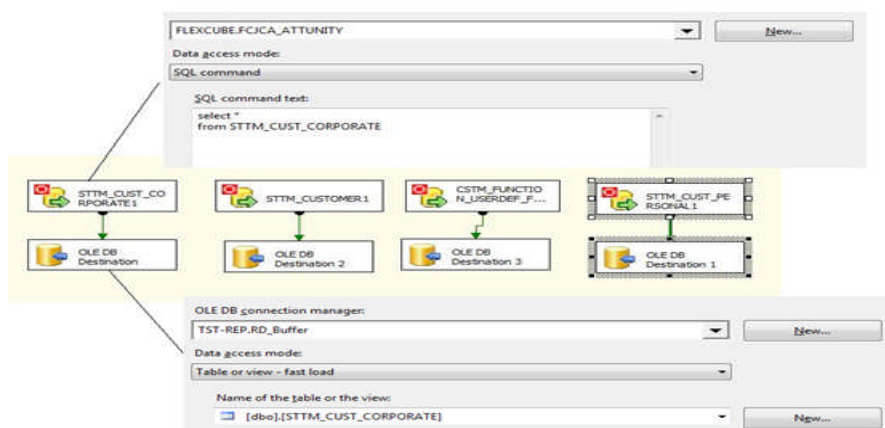
- Bëhen *truncate* të gjitha tabelat që ndodhen në databazën që përdor ky proces.

- *Select*–ohen të gjitha të dhënat në databazën e *Oracle*
- Transferimi i të dhënave nga *Oracle* në tabelat koresponduese në *SQL*.



**Figura 4.** Ekstraktimi i entiteteve

Në figurën më poshtë tregohet nën-procesi për njërin nga entitetet që do të ekstrahohen, Klientët. Për këtë entitet nevojiten të dhëna nga katër tabela të *Oracle*. Kynën-proces selekton të dhënat prej secilës prej këtyre tabelave të *Oracle* dhe i transferon në tabelat respektive të *SQL*-së.



**Figura 5.** Nën-procesi për entitetin e klientëve

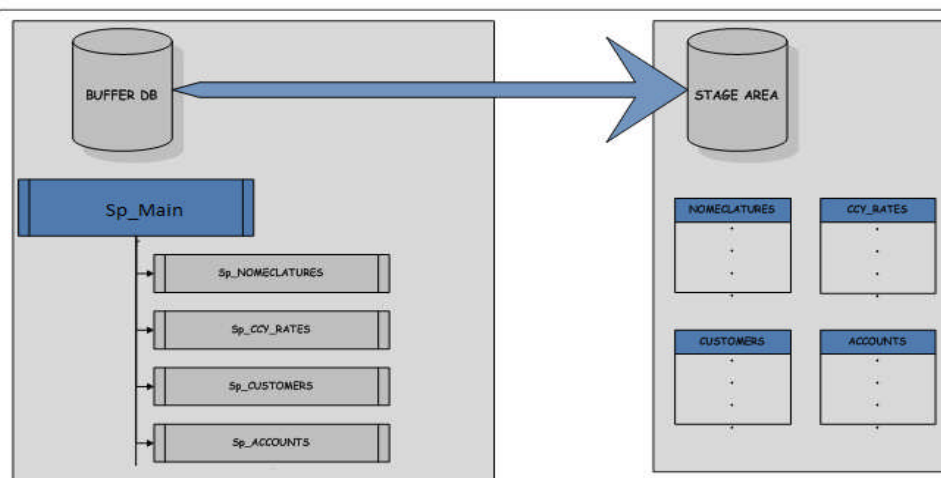
### Transformimi (Transform)

Pasi të gjitha tabelat e dëshiruara janë transferuar në databazën e pare *SQL* tani këto të dhëna mund të transformohen dhe grupohen në Entitete. Gjatë fazës së ekstraktimit pamë se si një entitet si ai Klientëve përfshinte transferimin e katër tabelave nga *Oracle* në tabelat respektive të *SQL*-së. Në këtë fazë të dhënat merren nga këto tabela, transformohen atëherë kur duhet, dhe vendosen në një tabelë të vetme (Klientët) të databazës *Stage*. Për të bërë të gjithë këto transformime është ndërtuar një procedurë (*sp\_Main*) kryesore e cila thërret një sërë nën-procedurash (psh *sp\_Loans*, *sp\_Accounts*, *sp\_Customers*, etj) të cilat bëjnë transformimet për cdo Entitet. Këto transformime përfshijnë nga kalimet një në një të kolonave të tabelave deri

në pasqyrimet (*mapings*) më të ndërlikuar midis fushave të cilat kërkojnë llogaritje.

Ekzekutimi i këtyre procedurave bëhet sipas një rradhe të përcaktuar që në fillim të ndërtimit të projektit. Kjo rradhë gjendet e ruajtur në njërin prej tabelave parametrike të *SQL*-së dhe sa herë procedura *sp\_Main* thërritet ajo shkon dhe kërkon renditjen tek kjo tabelë.

Si rezultat i këtij transformimi ndërtohet një databazë me një skemë *Snowflake* e cila përfshin Entitetet dhe dimensionet përkatëse.



**Figura 6.** Transformimi i të dhënave

#### Ngarkimi (*Load*)

Qëllimi i këtij procesi është historizimi i të dhënave që kalojnë në bazë ditore në databazën *Stage* si dhe kalkulli i balancave mesatarare të ponderuara për disa prej entiteve, si psh kreditë apo llogaritë. Funkcioni kresor i këtij procesi është që të shikojë ndryshimet që janë bërë në bazë ditore për cdo record të cdo Entiteti dhe ti historizojë këto ndryshime. Në varësi të natyrës së të dhënave që ndodhen në tabela dhe frekuencës së ndryshimit të vlerave të këtyre të dhënave është menduar që të përdoren tipe të ndryshme historizimi. Kjo procedurë përdor katër tipe historizimi.

#### Tipi I

Nëse procesi i zgjedhur do të jetë **I** atëherë historizimi bëhet në bazë ditore duke kaluar për cdo ditë një *snapshot* të tabelës e cila po kalohet nga databaza *Stage* në databazën historike, duke vendosur cdo rekord një datë raportimi. Ky tip historizimi përdoret kryesisht për tabela të cilat ruajnë informacion në baza ditore duke mos prekur informacionin për datat e mëparshme, psh rasti i entitetit të transaksionve.

Tipi M

Historizimi i tipit **M** nënkupton që në databazë do të vendosen të gjithë ato rreshta të cilat aktualisht nuk janë në databazë. Ky tip historizimi përdoret nëse tabelat nuk kanë shumë ndryshime të shpeshta. Ky tip përdoret kryesisht për tabelat të cilat janë parametrike. Psh tipet e monedhave të përdorura në sistem, shtetet me të cilat mund të kryhen transfertat etj.

Tipi H

Nëse historizimi është i tipit **H** atëherë për çdo rekord të entitetit ekzistues mbi të cilin ka ndryshim në njërin prej vlerave të kolonave të saj do të regjistrohet një rresht i ri, i cili ka një datë fillestare raportimi dhe një datë përfundimtare gjatë së cilës këto vlera kanë qenë aktive.

Për të kuptuar ndryshimet e vlerave të kolonave çdo rekordi në databazën *Stage* i bashkangjitet një **Hashvalue** e cila është kombinim i koduar i vlerave të kolonave për çdo rekord. **Hashvalue** të kësaj databaze më pas krahasohen me vlerën që ky rekord ka në databazën e Historizimit. Nëse vihen re ndryshime, atëherë rekordi regjistrohet si një rresht i ri duke ndryshuar vlerat e datave të raportimit.

**Shembull**

Një rekord është hedhur në databazën e Historizimit më datë 15/04/2010. Datat deri kur ky rekord është i vlefshëm përfaqësohen nga kolonat Reporting\_Begin Date dhe Reporting\_End\_Date si më poshtë:

<i>HASH_KEY</i>	<i>HASH_VALU</i>	BALANCA	NR I LLOGARISE	...	<i>REPORTING_BEGIN_DATE</i>	<i>REPORTING_END_DATE</i>
001	10	1000	123	...	15/4/2010	31/12/2099

**Tabela 1.** Rekord ne tabelen e llogarive

Nëse ndonjëra prej vlerave të këtij rekordi ndryshon në databazën *Stage* ajo do të pasqyrohet në vlerën që merr kolona **Hashvalue**. Në momentin që procedura e Historizimit për këtë entitet ekzekutohet, ajo kontrollon nëse për këtë rekord vlera e kolonës **Hashvalue** ka ndryshuar. Nëse po, atëherë kjo do të thotë që rekordi i vjetër nuk do të jetë më i vlefshëm për datat në vazhdim (për këtë arsye *End\_Date* e tij do të jetë data e këtij ndryshimi) dhe një rekord i ri do të regjistrohet me datë fillimi ditën e ndryshimit dhe datë mbarimi 31/12/2099.

<i>HASH_KEY</i>	<i>HASH_VALU</i>	BALANCA	NR I LLOGARISE	...	<i>REPORTING_BEGIN_DATE</i>	<i>REPORTING_END_DATE</i>
001	10	1000	123	...	15/4/2010	16/4/2010
001	11	1500	123	...	16/4/2010	31/12/2099

**Tabela 2.** Historizimi i nje rekordi ne tabelen e llogarive

Ky tip historizimi përdoret kryesisht për tabela të cilat ruajnë shumë të dhëna dhe për të cilat vlerat nuk ndryshojnë cdo ditë, psh sic është rasti i entitetit të llogarive apo të klientëve.

#### Tipi D

Nëse historizimi ështëi tipit **D** atëherë të dhënat fshihen dhe rihidhen përsëri në tabelë.

Përvec procesit të historizimit gjatë kësaj faze të *ETL*-së bëhen edhe disa llogaritje të tjera të cilat gjithashtu do të historizohen. Psh llogaritja e balancave mesatare për llogaritë e klientëve, apo për instrumenta të tjerë financiarë, apo dhe llogaritja e pagesave mujore të kredive apo overdrafteve.

Këto llogaritje bëhen nëpërmejt disa procedurave të cilat ekzekutohen vetëm në fund të çdo muaji. Pra bëhet një kontroll paraprak nëse është fund muaji, nëse po, atëherë këto procedura ekzekutohen.

### **3- Shtresa e Datawarehouse**

Sic u shpjegua edhe më sipër të dhënat e *datawarehouse* do të ruhen në një database *SQL Server 2008 R2*.

Të dhënat në këtë *datawarehouse* kalojnë në bazë ditore duke qenë se banka nuk ka akses në databazën *OLTP* të sistemit *core*, por edhe sepse është vlerësuar me rëndësi historizimi i të dhënave në baza ditore për t'u përdorur nga punonjësit. Procesi i ngarkimit të të dhënave në *datawarehouse* bëhet cdo natë.

Në këtë databazë të dhënat janë ruajtur sipas një skeme *SnowFlake* ku cdo entitet i përmendur më sipër përfaqëson një tabelë fakt (psh Klientët, Llogaritë, Kreditë, Letrat me Vlerë, etj) dhe rreth tyre qëndrojnë tabelat e dimensioneve (psh tipet e produkteve, tipet e klientëve, periudhat e vitit etj). Në këtë mënyrë informacioni është më lehtë i aksesueshëm dhe bëhet dhe më i shpejtë për t'u analizuar.

Informacioni i cili gjendet tashmë në *datawarehouse* mund të përdoret për të gjeneruar raporte për përdoruesit. Megjithatë, nëse do të nevojitet një analizë më e thelle e të dhënave sidomos të vlerave historike për të cilat merren të dhëna nga disa tabela të *datawarehouse*, kuptohet që një strukturë e tillë databaze përsëri nuk do të ishte ajo e duhura. Për këtë arsye në disa raste preferohet ndërtimi i kubeve *OLAP* nëpërmejt *SQL Server Analysis Services (SSAS)*. Nëpërmjet *SSAS* krijohen struktura databaze të cilat organizohen në kube (dimensione dhe fakte). *SSAS* ofron funksionalitete për të ndërtuar kube *OLAP*, por dhe *data mining*, të cilat përdoren nga aplikacione BI. *SSAS* parallogarit, grupon dhe e ruan informacionin të kompresuar në mënyrë të tillë që ai të jetë sa më i shpejtë kur përdoret nga përdoruesit.

Banka aktualisht ka në projektin e saj ndërtimin e dy strukturave *OLAP* (Fitim në nivel Dege dhe në nivel Produktesh), të cilat do të përdoren për të

strukturuar informacionin në mënyrë të tillë që ai të ketë përmbledhjen e duhur të të dhënave në formatet që kërkon menaxhimi i lartë.

#### 4- Shtresa e përdoruesit fundor

Një mënyrë për të gjeneruar dhe shpërndarë informacionin që tashmë është i ruajtur në *datawarehouse* ose në strukturat *OLAP* të ndërtuara në *SSAS* është dhe *SQL Server Reporting Services (SSRS)*. Ky mjet është pjesë e sistemit të BI që ofron *Microsoft*.

Raportet në bankë aktualisht ndërtohen dhe shpërndahen tek përdoruesit duke përdorur *SSRS*. Përdoruesit arrijnë të aksesojnë raportet e ndërtuara dhe të publikuara nëpërmjet ndërfaqes *web* të *Reporting Services (Report Manager)*, ose ato mund të skedulohen që të ekzekutohen dhe të shpërdahen automatikisht në email apo në *folder*-a të aksesueshëm nga përdoruesit.

Raportet të cilat publikohen ndërtohen nëpërmjet *Business Intelligence Development Studio*.

Për cdo raport zhvilluesi përcakton *datasource* prej së cilës do të tërhiqen të dhënat, cilat të dhëna do të tërhiqen, si dhe përlllogaritje të ndryshme mbi këto të dhëna (*dataset*), si dhe përcakton strukturën e paraqitjes së të dhënave në raport (*report layout*). Një *dataset* ndërtohet duke përdorur një *query* në databazë. Kjo *query* mund të jetë në *T-SQL*, *MDX* për të bërë *query* në të dhëna shumëdimensionale, ose *DMX* për të bërë *query* mbi *data-mining*. *Query* dërgon instruksione te *datasource*, duke i përcaktuar se cilat janë të dhënat që i nevojiten.

Të dhënat mund të paraqiten në formate:

- 1- Tabulare. Në këto formate numri i kolonave është gjithmonë fiks kurse ai i rreshtave varet nga të dhënat e selektuara në *dataset*.
- 2- Matricë. Në këto formate si numri i rreshtave, edhe ai i kolonave varet nga të dhënat që po selektohen. Ky është një format i ngjashëm me një tabelë *pivot* e cila ndryshon dinamikisht formën e saj. Kjo strukture është më e mirë për të shfaqur të dhënat ku si *datasource* janë kube *OLAP*.

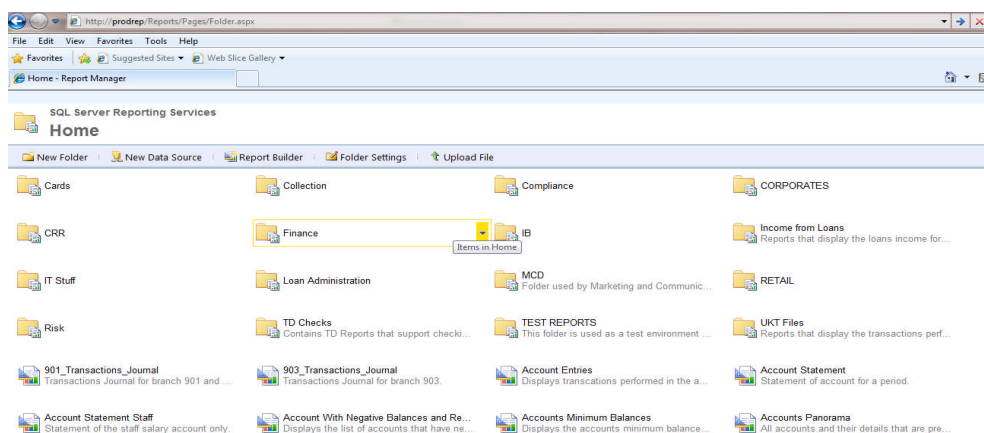


Figura 7. Ndërfaqja e aksesimit të raporteve

Gjithashtu, përveç paraqitjes shifrore, të dhënat mund të paraqiten dhe në formate grafikësh të ndryshëm të cilët mund të përdoren për të parë trendet apo për t'i parë të dhënat në një nivel më të përmbledhur.

Figura 7 më sipër jep një paraqitje të ndërfaqes ku përdoruesit mund të aksesojnë raportet.

#### 4. Përfundime

Në këtë artikull u trajtua arkitektura e një sistemi *Business Intelligence*, duke përshkruar shtresat përbërëse të saj, komponentet, koncepte të ndryshme të kësaj teknologjie, si dhe mënyrën se si këto pjesë lidhen me njëra-tjetrën. Ashtu si u nënvizua më sipër, nuk ekziston një arkitekturë fikse e kësaj teknologjie e cila do të na garantonte sukses në implementimin e saj, megjithatë mund të flitet për një strukturë të përgjithshme, e cila është e njëjtë pavarësisht modelit që kompanitë vendosin të zbatojnë.

Duke patur një arkitekturë të mire BI, kompanitë mund të marrin kthim maksimal nga investimi në një sistem *Business Intelligence*, duke ndikuar me anë të informacionit që gjeneron një sistem i tillë në marrjen e vendimeve të sakta menaxheriale.

Artikulli trajton një arkitekturë BI me katër shtresa dhe më pas paraqet mënyrën e implementimit praktik të kësaj arkitekture në një bankë.

#### Literatura

Azvine, B., Cui, Z., Nauck, D.D. (2005). Towards Real-time Business Intelligence. *BT Technology Journal*, 23(3), 214-225

MarketResearch.com (2010). Business Intelligence Tools Market Forecasted to Grow \$13 Billion by 2015. <http://www.marketwired.com/press-release/business-intelligence-tools-market-forecasted-to-grow-13-billion-by-2015-1162837.htm>

Turban, E., Sharda, R., Aronson, J. E., King D. (2008). *Business Intelligence: A Managerial Approach*. Prentice Hall, Upper Saddle River, New Jersey

Baars, H., Kemper, H.-G. (2008). Management Support with Structured and Unstructured Data: An Integrated Business Intelligence Framework. *Information Systems Management*, 25(2), 132-148

Balaceanu, D. (2007). Components of a Business Intelligence Software Solution. *Informatica Economica*, 2(42), 67-73

Inmon, W. H. (2002). *Building the Data Warehouse*. Wiley, New York, NY, USA, 3rd edition, 295