

**PRIMENA MEMORIJSKE BAZE PODATAKA U FUNKCIJI  
IZVEŠTAVANJA BANKARSKOG INFORMACIONOG SISTEMA  
APPLICATION OF IN-MEMORY DATABASE IN REPORTING  
OF THE BANKING INFORMATION SYSTEM**

Dejan Hadži-Milosavljević, Nenad Aničić

**REZIME:** Memorijska baza podataka je baza podataka smeštena u glavnoj memoriji računara. S obzirom na to da su pristup i obrada podataka iz glavne memorije mnogo brži nego kada su podaci na disku, glavna prednost korišćenja memorijske baze podataka je izuzetna efikasnost sa stanovišta brzine pristupa podacima i izvršavanja transakcija u realnom vremenu. U ovom radu izložen je koncept memorijske baze podataka i razmatrane su njene mogućnosti sa ciljem poboljšanja performansi za konkretni poslovni sistem - za bankarski informacioni sistem. U bankarskom sistemu najbitnije funkcionalnosti su u vezi sa finansijskim transakcijama, tako da zbog sigurnosti i integriteta podataka klasična baza podataka je veoma važna za bankarski sistem. Takođe bitna funkcionalnost je i izveštavanje koje je jako bitno klijentima banke. Ideja ovog rada je postizanje poboljšanja performansi sistema u domenu izveštavanja, upravo korišćenjem mogućnosti memorijske baze podataka. U radu je analizirana brzina izvršavanja upita nad podacima osnovnih baznih objekata iz jezgra bankarskog sistema koji se koriste u najbitnijim izveštavanjima. Opcije uključivanja baznih objekata u In-Memory Column Store, razmatrane su za In-Memory ORACLE Database 12c. Takođe su upoređivana vremena izvršavanja istih upita nad istim tabelama sa i bez uključivanja u In-Memory Column Store.

**KLJUČNE REČI:** Memorijska baza podataka, format kolone podataka, In-Memory Column Store, On Line Transaction Processing (OLTP)

**ABSTRACT:** In-Memory Database is a database located in computer's main memory. Given that access to and processing of data from main memory much faster than when the data on the disk, the main advantage of using a memory database is remarkable efficiency from the point of data access speed and execution of transactions in real time. This paper presents the concept of memory database and examine its capabilities in order to improve performance for information system of specific business system - the banking information system. In the banking system, the most important functionalities are in connection with financial transactions, so that for the safety and integrity of the data, the classic database is very important for the banking system. Also, reporting is the relevant functionality, which is very important to the clients of bank. The idea of this work is to achieve performance improvements in the area of reporting system, making use of the memory database possibilities. The paper analyzes the speed of execution of queries over the data from the basic tables of the core banking system used in the most important reporting. Options of inclusion the database objects in the In-Memory Column Store are considered for the In-Memory ORACLE Database 12c. The execution times are compared for the same queries over same database tables with and without inclusion in the Memory Column In-Store.

**KEY WORDS:** In-Memory Database (IMDB), In-Memory Column format, In-Memory Column Store, On Line Transaction Processing (OLTP)

## 1. UVOD

Nedostaci prvih sistema za upravljanje bazama podataka, kao što su ograničena nezavisnost podataka i složen programski pristup podacima, otklonjeni su uvođenjem relacionog modela baze podataka. U ovom modelu razdvojeni su logički i fizički model smeštanja podatka. Podaci su smešteni u formi tabela, a pristup podacima obezbeđen je jednostavnim strukturnim programskim jezikom za pisanje upita (SQL). Relacioni model bio je pogodan za klijent-server obradu, za paralelni prenos podataka, za korišćenje grafičkog korisničkog interfejsa.

Zbog razvoja računarskih mreža i zbog pojave složenih multimedijalnih podataka razvile su se baze podataka na konceptu objekta. Dalji razvoj diktirali su sledeći izazovi: upravljanje izuzetno složenim tipovima podataka kao što su npr. višedimenzionalni podaci, decentralizovane baze podataka, primena veštačke inteligencije za lakši pristup podacima, nove tehnike i metode analize podataka, novi načini zaštite podataka [1]. Pored ovih, najaktuelniji izazov tj. pravac razvoja baza

podataka je dobitak u efikasnosti i brzini izvršavanja transakcija. Sa tim ciljem, razvoj je išao ka smeštanju baze podataka u radnu memoriju. Tako je nastala memorijska baza podataka (In-Memory Database, IMDB).

Za izvršavanje transakcija bitno je da baza podataka bude klasična iz osnovnih razloga što ove baze podataka odlikuje sigurnost i postojanje integriteta. Ali za izveštavanje, koje je bitno korisnicima, treba naći rešenje koje će obezbediti brže dobijanje izveštaja. Upravo, tema ovog rada je analiza mogućnosti poboljšanja performansi izveštavanja korišćenjem memorijske baze podataka. U radu su prikazani rezultati poboljšanja performansi za konkretni primer primene memorijske baze podataka ORACLE Database 12c za izveštavanje u bankarskim informacionim sistemima.

## 2. MEMORIJSKA BAZA PODATAKA

Usled povećanja količine podataka koje treba obraditi, efikasnost i brzina izvršavanja transakcija dobijaju na značaju i

time diktiraju nove zahteve u razvoju baze podataka. Poslednično, kao odgovor na zahteve za poboljšanjem efektivnosti i povećanjem brzine, došlo se do novog koncepta razvoja baza podataka - razvile su se memorijske baze podataka [2].

### (I) Baza podataka je u glavnoj memoriji

Osnovna karakteristika koncepta memorijskih baza podataka je da glavna memorija (RAM) postaje primarni medijum za smeštanje podataka [3].

Poznata su sledeća vremena pristupa i čitanja podatka iz glavne memorije i sa diska [4]:

- Pristup podacima glavne memorije traje 100 ns, a podacima sa diska 5.000.000.000 ns.
- Čitanje 1MB sekvencijalnih podataka iz glavne memorije traje 250  $\mu$ s, a sa diska 30.000  $\mu$ s.

S obzirom na to da je rad sa glavnom memorijom mnogo brži nego pisanje i čitanje sa diska, memorijske baze podataka omogućavaju puno brži rad samih aplikacija [5].

### (ii) Izvršavanje transakcije bez pristupa disku

Koncept memorijskog keširanja podataka sa diska postoji kod svake hardverske konfiguracije koja ima keš memoriju. Smeštanje često korišćenih podataka u keš memoriju radi brzog pristupa primenjen je i na baze podataka.

Prilikom izvršavanja transakcija kod baza podataka koje su prethodile memorijskoj bazi podataka, princip prenosa podataka je sledeći [6]:

- aplikacija šalje zahtev za dobijanje podataka,
- Database Runtime šalje zahtev File System-u,
- File System kopira podatke sa diska u svoju keš memoriju i prosleđuje podatke bazi podataka,
- Database Runtime kopira podatke u svoju keš memoriju i prosleđuje podatke aplikaciji.

Izvršavanje transakcija kod memorijskih baza podataka je nezavisno od diska jer se podaci nalaze u glavnoj memoriji, što znači da ne postoji pristup podacima diska i ne postoji prenos podatak sa diska ka aplikaciji. Pristup i obrada podataka iz glavne memorije su mnogo brži nego kada su podaci na disku, i time se, jednostavno, dobilo na efikasnosti i brzini izvršavanja same transakcije.

### (iii) Jednostavno setovanje okruženja

Kod baza podataka koje pristupaju disku, keširaju se samo osnovne strukture podataka npr. podaci tipa nizovi koji predstavljaju memorijske liste koje programer može sortirati algoritmima programskog jezika. A na disku se i dalje nalaze podaci u formatu relacionih struktura. Da bi se koristile prednosti brzine RAM memorije, programer treba da transformiše strukturu podataka u toku izvršenja aplikacije.

Kod memorijskih baza podataka, relacione strukture implementirane u glavnu memoriju predstavljaju kopiju strukture baze podataka na disku. U ovom slučaju ne treba ništa dodatno programirati da bi se pristupilo bazi podataka. Do-

datni posao je samo to što administratori treba jednokratno da podese okruženje kojim se podaci sa diska smeštaju u glavnu memoriju u zavisnosti od veličine raspoloživog RAM-a.

### (iv) Trajnost podataka

Transakcija nad bazom podataka treba da ispunjava ACID zahteve - da je atomska (A) tj. da će se sve operacije transakcije u potpunosti izvršiti nad bazom podataka ili se neće izvršiti nijedna; da je konzistentna (C) tj. da rezultati transakcije treba da budu u skladu sa postojećim ograničenjima; da je izolovana (I) tj. da sistem obezbeđuje da je izolovana od ostalih transakcija; i da je trajna (D) tj. da ako se transakcija uspešno izvrši treba da bude zabeležena u trajnoj skladištenoj memoriji (disk).

Memorijskoj bazi podataka, da bi u potpunosti ispunila uslov trajnosti transakcija nad bazom, opcionalno je potreban disk. Trajnost podataka obezbeđuje se replikacijom baze podataka, arhiviranjem baze podataka, kreiranjem snapshot datoteka [7].

### (v) Primena memorijske baze podataka

Na razvoj memorijskih baza podataka pozitivno utiču brzi razvoj interneta, povećanje brzine prenosa podataka, razvoj višezvezganih procesora. Na primenu memorijskih baza podataka, u poslednje vreme, naročito je uticala i mogućnost adresiranja veće radne memorije, konkretno 64-čtvorobitni adresni prostor. U ovakav RAM moguće je smestiti i baze podataka veličine jednog terabajta [8].

Sadašnja cena memorijskih modula je relativno niska. S obzirom na to da se u RAM može smestiti i cela baza podataka, obični računari se vrlo jednostavno i jeftino mogu pretvoriti u "data centre".

Zbog velike brzine odziva pri pristupu podacima, memorijske baze podataka našle su primenu kod svih ugrađenih sistema za rad u realnom vremenu sa velikim skladištima podataka. To su npr.: industrijski kontrolni uređaji, sistemi sa mrežnom i telekomunikacionom opremom, kontrolni instrumenti civilnog i vojnog vazduhoplovstva, informacioni sistemi finansijskih institucija (banke i osiguravajuće kuće) [12], Web aplikacije poslovnih sistema i drugi.

## 3. FORMAT PODATAKA IN-MEMORY COLUMNU BAZI PODATAKA ORACLE 12C

Današnje informacione arhitekture su mnogo dinamičnije nego što su bile pre samo nekoliko godina. Poslovnim korisnicima informacionih sistema, u cilju odlučivanja, potrebno je brže dobijanje informacija. Da bi se podržao trend brzog dobijanja informacija, treba na istom okruženju podržati i izvršavanje transakcija i brže izvršavanje izveštaja nad velikim količinama podataka.

ORACLE Database 12c je primer memorijske baze podataka gde jedna baza podataka može u realnom vremenu efikasno da podrži i optimalno izvršavanje transakcija i analitička izveštavanja. To je moguće zahvaljujući jedinstvenoj "du-

al-format” arhitekturi koja obezbeđuje da podaci mogu biti u formatima koje ORACLE podržava [9]:

- Row format, za OLTP transakcije
- In-Memory Column format, za brza izvršavanja upita

Dok tradicionalni Row format predstavlja format u kome su obuhvaćeni svi atributi reda, novi Column format predstavlja svaki atribut kao posebnu strukturu kolone. Memorijska ORACLE baza podataka obezbeđuje istovremeno popunjavanje podataka u oba ova formata. Ovo ne znači dupli zahtev za memorijom, tako da nema dodatnih troškova za smeštanje podataka i za sinhronizaciju podataka.

Baza podataka održava potpunu konzistentnost transakcije između row formata i In-Memory column formata, kao što održava konzistentnost između tabela i indeksa. ORACLE Optimizer je svestan In-Memory Column formata. On automatski usmerava izvršavanje upita na Column format, a OLTP transakcije na Row format, pri čemu obezbeđuje potpunu konzistentnost podataka bez ikakvih izmena u korisničkim programima ili u aplikaciji [10].

In-Memory ORACLE baza podataka koristi In-Memory Column Store. Ovo je nova statička komponenta u SGA (System Global Area) ORACLE-ove baze podataka, koja se još naziva In-Memory Area. Podaci u In-Memory Column Store nalaze se u Column formatu tj. svaka kolona je smeštena kao posebna struktura. In-Memory Column Store nije zamena za keš memoriju, već predstavlja suplement koji obezbeđuje da podaci smešteni u memoriji budu ne samo u formatu reda, već i u formatu kolone.

Da bi se aktiviralo korišćenje In-Memory Column Store, potrebno je izvršiti jednostavnu parametrizaciju, samo postaviti parametar INMEMORY\_SIZE (default 0) na vrednost različitu od nule.

U In-Memory Column Store mogu se smestiti:

- specificirane grupe kolona
- cele tabele
- particije tabela
- materijalizovani pogledi.

Dodatno regulisanje aktiviranja In-Memory Column Store vrši se na različitim nivoima: na nivou kolone, na nivou tabele, na nivou materijalizovanog pogleda, na nivou tablespace, i na nivou particije. Na primer, ako se aktivira In-Memory Column Store na nivou tablespace, tada se automatski obezbeđuje uključanje svih tabela i materijalizovanih pogleda u In-Memory Column Store.

In-Memory Column Store je pogodan za sledeće upite:

- veliki upiti sa prediktima “=”, “<”, “>” i “IN”
- upiti koji vraćaju mali broj kolona iz tabele koja ima veliki broj kolona
- upiti u kojima su spojene male tabele u veće
- upiti koji agregiraju podatke.

In-Memory Column Store nije dizajniran za sledeće upite:

- upiti sa kompleksnim prediktima
- upiti koji vraćaju veliki broj kolona
- upiti koji vraćaju veliki broj redova
- upiti sa više spojenih velikih tabela.

Od odluke koji objekti će biti uključeni u In-Memory Column Store zavisi da li će se imati korist od korišćenja memorijske baze podataka. Ako se mudro izaberu objekti dobiće se veliko poboljšanje performansi. U suprotnom, gubi se velika memorija koja se mogla koristiti kao keš memorija. Uključivanje baznih objekata za u In-Memory Column Store vrši se pomoću sledećih naredbi koje su modifikovane korišćenjem klauzule INMEMORY [10], [11]: CREATE TABLE, ALTER TABLE, CREATE TABLESPACE, ALTER TABLESPACE, CREATE MATERIALIZED VIEW, ALTER MATERIALIZED VIEW.

#### 4. MOGUĆNOST PRIMENE MEMORIJSKE BAZE PODATAKA U BANKARSKOM INFORMACIONOM SISTEMU

ORACLE Database 12c je primer memorijske baze podataka koja ubrzava izvršavanje upita za nekoliko redova veličina vremenske jedinice. Ako je kao korisnik neki biznis subjekat, evidentan je doprinos u bržem donošenju odluke u realnom vremenu. Koristeći ORACLE memorijsku bazu, preduzeća mogu nakon pokretanja analitičkih izveštaja i skoro momentalnog dobijanja rezultata, mogu odmah da krenu sa donošenjem odluke na osnovu tih izveštajnih rezultata. Ako su se nekada izveštaji izvršavali po nekoliko sati, sada je očigledna korist od bržeg donošenja odluka u realnom vremenu - niži troškovi, veća produktivnost i jača konkurentnost.

ORACLE memorijska baza ubrzava i analitike (Data Warehouse) i transakcione (OLTP) aplikacije [4]. Zahvaljujući sposobnosti da se u realnom vremenu brzo obavlja analiza podataka kao i obrada transakcija kroz aplikacije, organizacije i preduzeća donose odluke u realnom vremenu, odmah odgovaraju na zahteve klijenata, i stalno optimiziraju svoje ključne procese.

Kao primer informacionog sistema u kome izveštavanja čine bitan deo je bankarski informacioni sistem. U svakoj banci, modul za izveštavanja čine:

- izveštaji prema klijentima u ekspozituri
- izveštaji za rukovodioce u banci
- izveštaji prema Centralnoj nacionalnoj banci
- izveštaji za partnere kao što su npr. poreska uprava, osiguravajuće kuće, brokerske kuće.
- opciono, izveštaji prema međunarodnom centru poslovanja bankarske grupacije.

U toku radnog vremena banke izvršavaju se transakcije u realnom vremenu. Spora izveštavanja u produkcijskom okruženju usporavala bi ove transakcije. Da bi ubrzale dobijanje izveštaja, banke pribegavaju sledećim rešenjima. Prvo rešenje je da se koristi drugi bazni server koji je kopija produkcijskog okruženja i koji se osvežava van radnog vremena. Ovaj dodatni server koristi se samo za izveštavanje. Na njemu se izvršavaju spori izveštavaji i time se ne remeti rad produkcijskog servera. Drugo rešenje je da se noćnim job-ovima na produkcijskom okruženju izvršavaju procedure koje prikupljaju podatke na isti način kao upiti iz sporih izveštaja, i da se ti podaci

smeštaju u dodatne snapshot tabele iz kojih će se sutra u toku radnog vremena selektovati u okviru jednostavnih izveštaja. Postoji problem kod oba rešenja što izveštaji mogu da se dobijaju samo nad podacima zaključno sa jučerašnjim danom jer su trenutni podaci samo na produkcijskom serveru. To znači da će se izveštaji koji zavise od trenutnih današnjih podataka i dalje izvršavati na produkcijskom okruženju u toku radnog vremena. Dakle, jedino rešenje za brzo izvršavanje izveštaja je ubrzavanje dobijanja podataka iz baze podataka na osnovu izveštajnog upita.

Na osnovu mogućnosti da je korišćenjem ORACLE Database 12c brže dobijanje izveštaja, da je brži pristup podacima iz Data Warehouse-a, i da je brže aplikativno izvršavanje transakcija, ova memorijska baza zadovoljava najbitnije zahteve u pogledu izveštavanja u bankarskom informacionom sistemu. Očigledna je korist banke od brzo dobijenih izveštaja u situacijama kada u ekspozituri treba hitno da se odgovori klijentu koji čeka kod operatera, kada menadžeri u banci treba da donesu ključne odluke, kada treba da se dostavi kompleksni regulatorni izveštaj Centralnoj nacionalnoj banci.

## 5. REZULTATI IZVRŠAVANJA BANKARSKIH IZVEŠTAJA U MEMORIJSKOJ ORACLE 12C BAZI PODATAKA

Jedan od osnovnih skupa bančinih izveštaja zasniva se nad podacima dobijenim iz evidencije o bančinih transakcijama. Ova evidencija ima ogroman broj podataka i jedinstvena je za

sve transakcije po svim partijama svih komitenata, za period od početka poslovanja banke. Zbog velikog značaja ovih izveštaja o transakcijama, kao bitan zahtev nameće se brzina njihovog izvršavanja. S obzirom na to da korišćenje memorijske baze podataka ima smisla kod izveštavanja sa ciljem poboljšanja performansi u vremenskom domenu, i u ovom konkretnom slučaju bankarskog informacionog sistema može se uvesti memorijska baza podataka.

Mogućnosti memorijske ORACLE Database 12c iskorišćene se za izveštavanje o transakcijama banke. Da bi se dokazala prednost korišćenja memorijske nad tradicionalnom bazom podataka kod izveštavanja o transakcijama, zamišljeno je da se u ORACLE Database 12c, na virtuelnoj platformi, posmatra izvršavanje istog izveštajnog upita za slučaj kada se koristi In-Memory Column Store i za slučaj kada se ne koristi. Na osnovu specifikacije o bančinoj evidenciji transakcija, kreiran je entitet, tabela TRANSAKCIJA, sa 38 atributa. Format podataka je klasični, row format. Kada se izvršava upit odjednom će se pristupati podacima svih atributa u redu. To pokazuje i TABLE ACCECSS (FULL) u Explain Plane na Slici 1.

Kreirana je i tabela TRANSAKCIJA\_IM sa sličnom definicijom kao tabela TRANSAKCIJA, i popunjena je istim podacima.

```
CREATE TABLE transakcija_im
(broj_naloga          NUMBER          NOT NULL,
 datum_obrađe       DATE              NOT NULL,
 partija             VARCHAR2(18 BYTE) NOT NULL,
 konto               VARCHAR2(10 BYTE) NOT NULL,
```

The screenshot displays the Oracle SQL Developer interface. At the top, the query editor shows the following SQL statement:

```
select datum_obrađe,
       broj_naloga,
       partija
from transakcija
where datum_obrađe > to_date('01.01.2016','dd.mm.yyyy')
and datum_obrađe < to_date('31.01.2016','dd.mm.yyyy')
and partija IN ('265003100000676128','265003100000453430')
```

Below the query editor, the 'Explain Plan' tab is active, showing the execution plan for the query. The plan consists of the following operations:

- SELECT STATEMENT
- TABLE ACCESS (FULL) - OBJECT\_NAME: TRANSAKCIJA
- Filter Predicates
- AND
- OR
- DATUM\_OBRAĐE < TO\_DATE(' 2016-01-31 00:00:00', 'syyyy-mm-dd hh24:mi:ss')
- PARTIJA = '265003100000453430'
- PARTIJA = '265003100000676128'
- DATUM\_OBRAĐE > TO\_DATE(' 2016-01-01 00:00:00', 'syyyy-mm-dd hh24:mi:ss')

Slika 1. Explain Plan za upit nad tabelom TRANSAKCIJA

```
sifra_valute          NUMBER(3)    NOT NULL,
...
vreme_potvrde DATE      NOT NULL,
napomena VARCHAR(240 BYTE)
) INMEMORY;

ALTER TABLE transakcija_im
NO INMEMORY (konto,
sifra_valute,
...
vreme_potvrde,
napomena);
```

U odnosu na tabelu TRANSAKCIJA, tabela TRANSAKCIJA\_IM se razlikuje samo u tome što je za tri kolone broj\_naloga, datum\_obrade i partija definisan format In-Memory column format. Ostalih 35 kolona su u formatu row format. To znači da su pomenute tri kolone predviđene za korišćenje u In-Memory Column Store. Kada se selektuju podaci iz ove tri kolone, njima će se vertikalno pristupati. To pokazuje i TABLE ACCESS (INMEMORY FULL) u Explain Plane na Slici 2.

Nad tabelama TRANSAKCIJA i TRANSAKCIJA\_IM izvršavani su isti upiti i upoređivana su vremena trajanja upita. Upiti su se brže izvršavali na tabelom TRANSAKCIJA\_IM.

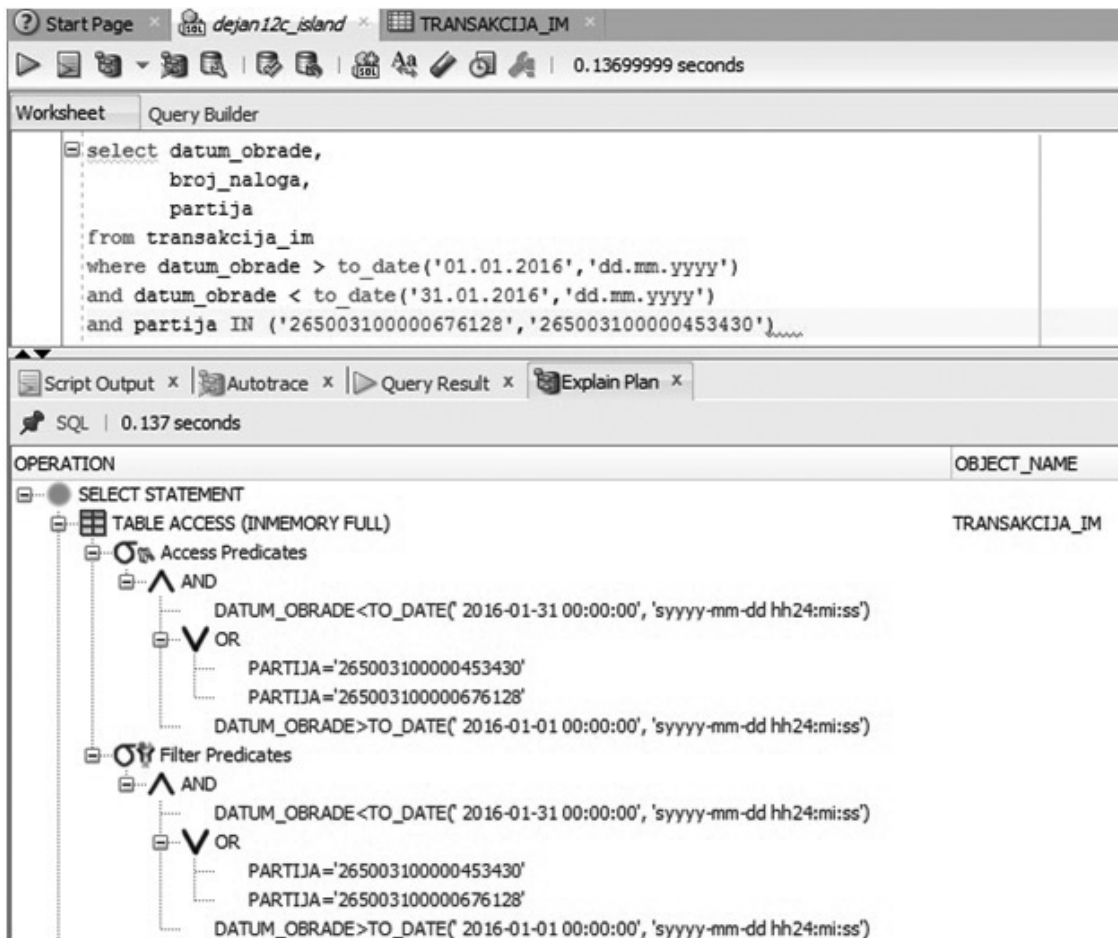
Kao primer kompleksnijeg izveštaja, uzet je izveštaj koji sve poslovne banke jednom mesečno šalju Narodnoj banci - "Izveštaj o procenama vrednosti nepokretnosti koje su pred-

met hipoteke i o kreditima obezbeđenim hipotekom". Upit iz ovog izveštaja izvršavan je nad dvema istim tabelama koje sadrže iste podatke o tranšama dobijenim nakon alokacije kolaterala, s tim što su kolone jedne tabele predviđene za smeštanje u In-Memory Column Store, a kolone druge tabele ne.

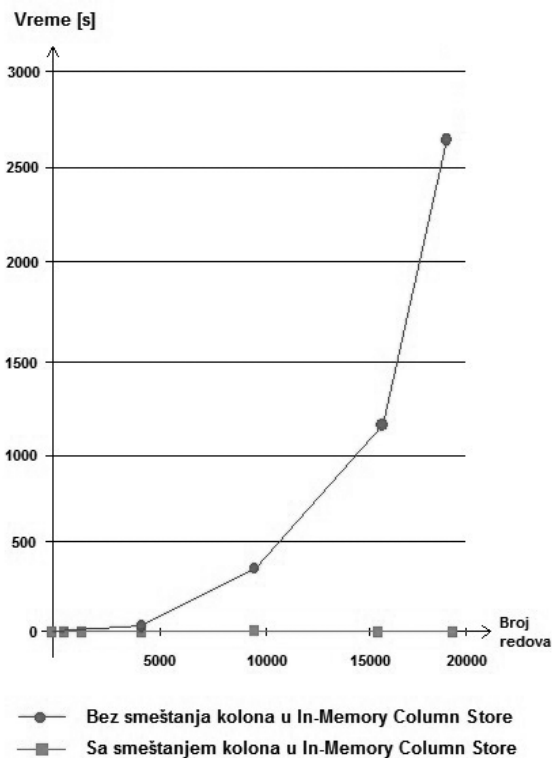
Prilikom izvršavanja upita menjan je opseg za datum procene kolaterala iz razloga da bi se dobili različiti brojevi redova u rezultatu. U tabeli Tabela 1 prikazana su dobijena vremena trajanja upita "Task Completed in [s]" u odnosu na broj redova u rezultatu.

Broj dobijenih redova kao rezultat upita	Bez korišćenja In-Memory Column Store [s]	Sa korišćenjem In-Memory Column Store [s]
12	0.002	0.002
142	1.144	0.054
819	10.181	0.606
1693	21.265	2.247
3098	79.451	4.115
9016	442.643	6.230
15711	1253.127	8.717
18404	2670.833	11.208

Tabela 1. Prikaz vremena izvršavanja upita za slučajeve bez i sa smeštanjem kolona u In-Memory Column Store



Slika 2. Explain Plan za upit nad tabelom TRANSAKCIJA\_IM



Slika 3. Prikaz vremena izvršavanja upita za slučajeve bez i sa smeštanjem kolona u In-Memory Column Store

Analizom ovih vremena trajanja upita, zaključuje se da su mnogo kraće trajala izvršavanja upita u slučaju kada se koristi In-Memory Column Store, nego u slučaju kada se upit izvršavao nad klasičnom tabelom. Za ova dva slučaja, na grafičkom prikazu na Slici 3, jasnije može da se uoči razlika u vremenima izvršavanja upita. Evidentno je da sa povećanjem broja reda koje vraća upit, da je veća korist od korišćenja memorijske baze podataka. Međutim, u ovom konkretnom primeru relativno je mali maksimalan broj vraćenih redova – 18404 redova. Ukoliko je broj vraćenih redova mnogo veći, ne bi bilo efikasno korišćenje memorijske baze podataka [4].

## 6. ZAKLJUČAK

Za izbor baze podataka, ako se posmatra izvršavanje transakcija, najbolje rešenje su tradicionalne baze podataka koje obezbeđuju sigurnost i integritet. Ako se za pojedine funkcionalnosti želi postići poboljšanje u vremenskom domenu, tada su odlično rešenje memorijske baze podataka. Najčešće je potrebna kombinacija klasičnih baza podataka i memorijskih baza podataka, a to nudi ORACLE.

Izveštavanje je tipičan primer funkcionalnosti kod koje je jako bitno da se što pre dobiju izveštajni podaci. To znači da memorijska baza podataka može naći primenu u onim informacionim sistemima koji imaju izveštavanja. Takavi su bankarski informacioni sistem. S obzirom na to da su bančini izveštaji finansijski, velika je korist kada banka treba hitno da odgovori na zahtev komitentata ili kada rukovodstvo banke treba da donese ključne odluke.

Memorijska baza podataka ORACLE Database 12c ima sve mogućnosti da ispuni brzo izvršavanje upita jer baza podataka koja je smeštena u radnoj memoriji podržava smeštanje podataka u novom formatu, Column format, nove SGA komponente In-Memory Column Store. Zbog ove sposobnosti In-Memory ORACLE Database 12c može biti primenjena u bankarskim informacionim sistemima u funkciji izveštavanja.

U radu je izvršena analiza rezultata upita bankarskih izveštaja, urađenih u In-Memory Database 12c. Pokazano je da je postignuto je poboljšanje performansi nego kada se koriste mogućnosti klasične baze podataka. Takođe je ukazano i na to da treba proceniti za koja izveštavanja je bolje koristiti memorijsku bazu podataka a za koja klasičnu – ako se radi o hitnim i bitnim izveštajima sa manjim brojem redova tada treba koristiti memorijsku bazu podataka, a ako su izveštaji masovni sa ogromnim redovima podataka, kao što su godišnji i kvartalni izveštaji, bolje je koristiti klasičnu bazu podataka.

## REFERENCE

- [1] H. Plattner, A. Zeier, In-Memory Data Management - Technology and Applications, Springer, 2012;
- [2] B. Milošević, D. Milošević, S. Stančić, "IN MEMORY baze podataka", ZBORNIK RADOVA VTŠ, Niš, 2015;
- [3] J. Jackson, "In-memory technologies move databases to real time", PCWORLD, IDG Consumer & SMB, San Francisco, 2014;
- [4] S. Primorac, "Promena memorijske baze podataka za obradu podataka o korisničkoj potrošnji telekomunikacijskih usluga", FESR-KRON, Split, 2013;
- [5] S. Tu, W. Zheng, E. Kohler, B. Liskov, S. Madden, "Speedy Transactions in Multicore In-Memory Databases", SOSP'13, Farmington, 2013;
- [6] C. Evans, "In-Memory Database", Computer Weekly, London, 2012;
- [7] T. Lahiri, M. Kissling, "Oracle's In-Memory Database Strategy for OLTP and Analytics", ORACLE Corporation, Redwood Shores, 2015;
- [8] M. Spasić, "CSQL In-Memory Database", IT Modul, Beograd, 2012;
- [9] M. Colgan, "ORACLE Database In-Memory", ORACLE White Paper, Redwood Shores, 2015;
- [10] N. Yuhanna, L. Owens, E. Cullen, D. Lynch, "In-Memory Database Platforms", Forrester Research Inc, Cambridge 2015;
- [11] J. Pojatina, "ORACLE In-Memory baza bez In-Memory opcije", Konferencija 19HrOUG, Rovinj, 2014;
- [12] D. Jednak, "Uticaj informacionih tehnologija na bankarski sektor", Info M, Vol. 29, 2009.



**Dejan Hadzi-Milosavljević**, Raiffeisen banka Srbije

**Kontakt:** dejan.hadzi-milosavljevic@raiffeisenbank.rs

**Oblasti interesovanja:** Baze podataka, Upravljanje podacima, Projektovanje informacionog sistema za bankarsko poslovanje



**Nenad Aničić**, Fakultet organizacionih nauka, Univerzitet u Beogradu

**Kontakt:** anicic.nenad@fon.bg.ac.rs

**Oblasti interesovanja:** Razvoj informacionih sistema, Baze podataka, Integracija i interoperabilnost poslovnih sistema