



Oracle LOB-ovi

Stanko Plivelić, dipl.ing.

30.09.2008.

13.konferencija HrOUG, Rovinj, listopad 2008.

O čemu će biti riječi

2

- ◆ što su LOB-ovi i koje su vrste
- ◆ arhitektura
- ◆ operacije s LOB-ovima
- ◆ ograničenja u radu s LOB-ovima
- ◆ primjeri i problemi

Što i zašto

3

- ◆ 8 do 128 terabajta (ovisno o DB_BLOCK_SIZE)
- ◆ slike, dokumenti, multimedija
- ◆ VARCHAR2 maksimalno 4000 bajta
- ◆ LONG
 - verzija 7 i ranije
 - maksimalna veličina 2GB
 - maksimalno 1 kolona po tablici
 - sekvencijalni pristup

Vrste LOB-ova

◆ interni

- BLOB - binarni
- CLOB - tekst
- NCLOB – tekst (National Character Set)

◆ eksterni:

- BFILE
 - datoteka izvan baze
 - lokacija i ime datoteke u bazi
 - read-only

Arhitektura

◆ LOB podatak

- u tablici – *inline* – ($\leq 3964B$) - default
- izvan tablice – *out-of-line*

◆ LOB lokator – referenca na lokaciju podatka

◆ pohrana

- *inline* - (*enable storage in row*) - kao VARCHAR2(4000)
- *out-of-line* - (*disable storage in row*) - direktno u LOB segment

LOB storage (1)

6

- ◆ LOB podaci su pohranjeni u chunk-ovima
 - sastoje se od 1 ili više blokova (max. 32K)
 - najmanja jedinica s kojom se radi I/O operacija
- ◆ po chunk-ovima se navigira preko index-a
 - u svakoj koloni postoje 2 segmenta: 1 za podatak, 1 za index
 - ograničenja – ne može se dropati ili rebuild-ati LOB indexe, ne mogu se postaviti različita svojstva za podatkovne i indeksne segmente

LOB storage (2)

7

- ◆ opcije keširanja (*buffer cache*)
 - NOCACHE (default) – veliki LOB-ovi, rijetko čitanje
 - CACHE – često čitanje/pisanje
 - CACHE READS – često čitanje

- ◆ PCTVERSION
- ◆ RETENTION
- ◆ LOGGING/NOLOGGING

Stanja LOB-a

8

- ◆ null
 - ne postojе LOB podatak ni lokator
- ◆ prazan - *empty*
 - postojе, ali su prazni, ne sadrže podatak/lokaciju
- ◆ popunjen
 - postojе, sadrže podatak/lokaciju

Interni LOB-ovi

- ◆ perzistentni LOB-ovi
 - vrijednost u nekoj koloni tablice
 - sudjeluju u transakcijama (commit, rollback...)
 - LOB lokator smješten u koloni tablice, LOB podatak može biti smješten u drugom tablespace-u
- ◆ privremeni (temporary) LOB-ovi
 - nalaze se u temporary tablespace-u
 - nastaju kad se instancira LOB varijabla
 - postaju permanentni kad se insertiraju u tablicu

Operacije s LOB-ovima

- ◆ 2 tipa operacija sa LOB-ovima:
 - *copy* – podatak se kopira, kreira se novi lokator - BLOB, CLOB i NCLOB
 - *reference* – kopira se lokator - BFILE

Primjer: kreiranje tablice s LOB-ovima (1)

- create table artikli
(id number primary key,
sifra varchar2(10),
naziv varchar2(200),
opis_cl CLOB DEFAULT empty_clob(),
slika_bl BLOB DEFAULT empty_blob(),
primjer_bl BLOB DEFAULT empty_blob(),
upute_bf BFILE)
LOB(opis_cl) store as opis_seg(
tablespace DATA
enable storage in row
chunk 8192
cache)

Kreiranje tablice s LOB- ovima (2)

```
LOB(slika_bl) store as slika_seg(  
  tablespace LOBS_BIG  
  disable storage in row  
  chunk 32768  
  cache reads)
```

```
LOB(primjer_bl) store as primjer_seg(  
  tablespace LOBS_BIG  
  disable storage in row  
  chunk 32768  
  nocache)
```

Kreiranje tablice s LOB-ovima (3)

- ◆ na kraju definicije tablice *storage* blok
- ◆ svi interni LOB-ovi inicijalizirani sa `EMPTY_CLOB()` ili `EMPTY_BLOB()`
- ◆ kolona `opis_cl` – često se dohvaća i ažurira, mala količina podataka
 - isti tablespace kao i tablica
 - inline
 - keširanje uključeno

Kreiranje tablice s LOB- ovima (4)

- ◆ kolona slika_bl – često se dohvaća, rijetko ažurira, veća količina podataka
 - poseban tablespace
 - veliki CHUNK
 - kešira se za čitanje

- ◆ razlika između slika_bl i primjer_bl
 - primjer_bl se rijetko čita – nema keširanja
 - sve ostalo je jednako definirano

BFILE

◆ DIRECTORY – logički alias za fizički direktorij

- CREATE DIRECTORY "dokumenti" AS
"G:\podaci\dokumenti";

◆ više referenci na isti fajl

◆ BFILENAME

- insert into table artikli(id, sifra, naziv, opis_cl,
slika_bl, primjer_bl, upute_bf)
values (1, '101', 'AUTO RADIO', empty_clob(),
empty_blob(), empty_blob(), BFILENAME('dokumenti',
'radio_manual.pdf'))

Rad s LOB-ovima (1)

16

- ◆ built-in package DBMS_LOB
- ◆ funkcije rade s lokatorom – ne može biti null
- ◆ parametri: CLOB – znak, BLOB - bajt

- ◆ primjer 1
- ◆ korištenjem `SELECT ... INTO ... FOR UPDATE` zaključava se zapis i vraća lokator
- ◆ operacija se izvršava direktno na podatku u tablici

Rad s LOB-ovima (2)

- ◆ read consistency
- ◆ primjer 2

- ◆ SQL funkcije koje rade sa stringovima (u pravilu) rade i sa LOB-ovima
- ◆ rade s lokatorima
- ◆ rezultat je privremeni (temporary) LOB
- ◆ nema posljedica na podatku u tablici
- ◆ primjer 3

Ograničenja (1)

- ◆ kada ne korisititi SQL funkcije:
 - >1MB (100KB)
 - random pristup
- ◆ INITCAP, TRANSLATE, DECODE i neke druge funkcije rade samo sa prvih 4K (u SQL-u), odnosno prvih 32K (u PL/SQL-u) podatka
- ◆ operatori usporedbe (>, !=, between) rade samo u PL/SQL-u (ne i u SQL-u)

Ograničenja (2)

◆ SQL ograničenja

- LOB-ovi ne mogu stajati u ORDER BY, GROUP BY ni SELECT DISTINCT
- tablice se ne mogu join-ati preko LOB kolona
- agregatne funkcije ne rade s LOB-ovima

◆ DDL ograničenja

- LOB kolone ne mogu biti dio primarnog ključa
- mogu se kreirati samo domain i text indexi
- LOB kolona se ne može nalaziti u UPDATE OF listi kod UPDATE trigera
- promjene preko lokatora ili DBMS_LOB ne okidaju UPDATE triger

Ograničenja (3)

◆ DBLink

- može se koristiti samo CREATE TABLE AS SELECT i INSERT AS SELECT

◆ sistemska ograničenja

- ograničen je broj BFILE-ova koji mogu istovremeno biti otvoreni (SESSION_MAX_OPEN_FILES - default je 10)
- mogu se mijenjati samo neki parametri (storage, caching, tablespace), ali ne i veličina chunk-a ili storage in row

Performanse

- ◆ koristiti *enable storage in row* kad je moguće
- ◆ veličinu CHUNK-a prilagoditi veličini LOB-a
- ◆ privremeni LOB-ovi – *"reference on read, copy on write"*
- ◆ dealokacija privremenih LOB-ova sa
DBMS_LOB.FREETEMPORARY

ORA-01555

- ◆ ASSM (Auto Segment Space Management)
- ◆ nije uključena CACHE opcija
- ◆ rollback to savepoint između 2 pisanja u LOB
- ◆ PCTVERSION
- ◆ 2 sesije istovremeno

Primjer: Forms i LOB (1)

23

- ◆ XML dokument – CLOB
- ◆ CLOB u formi – pristup (npr. `dbms_lob.substr`) rezultira rušenjem forme
- ◆ rješenje: funkcija u bazi

Primjer: Forms i LOB (2)

```

◆ FUNCTION f_vrati_varchar2(offset in out number, amt number,
p_net_id number) RETURN varchar2
is
    buffer varchar2(32767);
    cursor c_clob is
    select TIP_PODACI
    from   TXT_IZVJESTAJI_PODACI
    where  TIP_ID = p_net_id;
begin
    open c_clob;
    fetch c_clob into c;
    close c_clob;
    if c is not null and offset <= c_size and amt <= 32767 then
        buffer := dbms_lob.substr(c, amt, offset);
        offset := offset + length(buffer);
        return buffer;
    end if;
    return null;
end f_vrati_varchar2;

```


Zaključak

25

- ◆ praktički nema ograničenja što se tiče veličine
- ◆ iznad 4KB neizbježni
- ◆ obavezno testiranje

Pitanja

♦ ...