



UNIVERZITET U NIŠU
ELEKTRONSKI FAKULTET



Sistemi za upravljanje bazama podataka

Seminarski rad

Interna struktura i organizacija skladišta podataka

Oracle baze

Mentor:

Doc. dr Aleksandar Stanimirović

Student:

Maja Stojanović 1227

April 2021, Niš

Sadržaj

1. Uvod	3
2. Fizičke strukture za skladištenje podataka	4
2.1 Mehanizam za skladištenje fajlova baze podataka	4
2.1.1 Oracle ASM	4
2.1.2 Oracle ASM komponente za skladištenje	5
2.1.3 Oracle ASM instance	6
2.2 Data files	7
2.2.1 Permanent i Temporary Data Files	8
2.2.2 Online i Offline Data Files.....	8
2.2.3 Data File Structure	8
2.3 Control files	9
2.3.1 Control File struktura.....	10
2.4 Online Redo Log.....	10
2.4.1 Archived Redo Log Files	11
2.4.2 Online Redo Log struktura.....	11
3. Logičke strukture za skladištenje podataka.....	11
3.1 Hijerarhija logičke strukture	12
3.2 Data Blocks	13
3.2.1 Row Format	14
3.2.3 Data blocks- upravljanje prostorom.....	16
3.3 Extents.....	20
3.4 Segments	21
3.5 Tablespace.....	22
4. Primeri	23
5.Zaključak	29
6.Literatura.....	30

1. Uvod

Baze podataka su sistemi koji omogućavaju skladištenje, održavanje i pristup podacima. Priroda podataka koje baze skladište može biti različita: numerički, alfanumerički, multimedijalni podaci itd. U zavisnosti od namene i količine podataka kojim upravljaju baze podataka razvijeni su različiti proizvodi kako komercijalni tako i open source (MsSql server, Oracle, MySql, SqLite itd.). Svaka baza podataka koristi interne strukture i specifičnu organizaciju za skladištenje podataka.

Sistem za skladištenje podataka može biti objašnjen kao kapacitet koji je osiguran od strane DBMS-a (Database Management System) u memoriji alociranoj za bazu podataka i operacije koje su povezane sa njom. U relacionim bazama podataka, podaci se skladište u vidu tabela, kolona, redova i odgovarajućih veza. Drugim rečima, to je strukturirani sistem uređen na način da raspoređuje potreban prostor za sadržaj baze i rad baze podataka.

Jedna od karakteristika RDBMS-a je nezavisnost logičkih struktura (tabele, pogledi, indeksi) od fizičkih struktura za skladištenje podataka. Zbog toga je moguće upravljati fizičkim skladištenjem podataka bez uticaja na logičke strukture.

Baza podataka se može posmatrati iz dve perspektive, fizičke i logičke. Fizički podaci su podaci koji su vidljivi na nivou operativnog sistema. Na primer: operativni sistem Linux pruža mogućnost da se naredbama ls i ps prikažu fajlovi i procesi baze podataka. Logički podaci, kao što su tabele su jedino značajni za bazu podataka. SQL naredba može izlistati sve tabele u nekoj bazi, dok operativni sistem nema tu mogućnost. Fizičke strukture za skladištenje su fajlovi koji čuvaju podatke, a logičke strukture za skladištenje omogućavaju bazi podataka finu granulaciju kontrole upotrebe prostora na disku.

U narednim poglavljima biće detaljnije obrađene fizičke i logičke strukture za skladištenje podataka koje koristi Oracle baza.

2. Fizičke strukture za skladištenje podataka

Oracle baza je skup fajlova koji skladište Oracle podatke u trajno skladište na disku. Ovo poglavlje se bavi fajlovima koji se generišu kada se izda naredba CREATE DATABASE:

- **Data files i temp files**- Data file je fizički fajl na disku koji je Oracle baza kreirala i koji sadrži strukture podataka kao što su tabele i indeksi. Temp file je data fajl koji pripada privremenom tablespace-u. Baza podataka upisuje podatke u ove fajlove u formatu koji pripada samo Oracle-u i koji ne može biti pročitan od strane drugih programa.
- **Control files**- root fajl koji prati fizičke komponente baze podataka.
- **Online redo log files**- skup fajlova koji beleže promene koje su izvršene nad podacima.

2.1 Mehanizam za skladištenje fajlova baze podataka

Postoje nekoliko mehanizama za alociranje i upravljanje skladištenjem ovih fajlova:

- **Oracle Automatic Storage Management (Oracle ASM)**- uključuje fajl sistem koji je napravljen ekskluzivno samo za korišćenje od strane Oracle baze.
- **Operating system file system**- većina Oracle baza skladišti fajlove u fajl sistem, koji predstavlja strukturu podataka ugrađenu unutar kontinualnog adresnog prostora diska. Svi operativni sistemi imaju fajl menadžere koji alociraju i dealociraju prostor na disku u fajlove unutar fajl sistema.
- **Cluster file system**- distribuirani fajl sistem koji je klaster servera koji sarađuju kako bi pružili visoke performanse servisa klijentima.

2.1.1 Oracle ASM

Oracle ASM je rešenje za skladištenje fajlova Oracle baze visokih performansi i jednostavnog upravljanja.

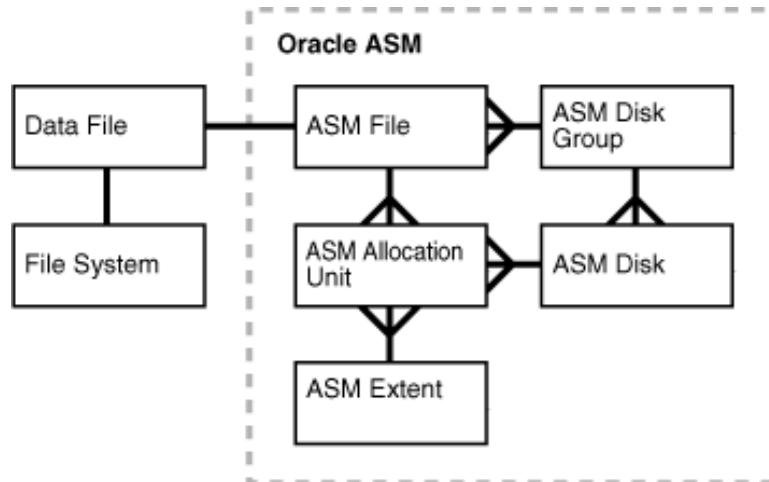
Oracle ASM ima nekoliko prednosti u odnosu na obične fajl sisteme i menadžere skladišta:

- Pojednostavljuje zadatke povezane sa skladištem, kao što su kreiranje i postavljanje baza podataka i upravljanje prostorom na disku
- Distribuira podatke preko fizičkog diska kako bi eliminisao žarišta i pružio ujednačene performanse na diskovima
- Rebalansira podatke automatski nakon promene konfiguracije skladišta

Da bi se koristio Oracle ASM, mora se alocirati particionisani disk za Oracle bazu podataka sa preferencama za striping i mirroring. Oracle ASM upavlja prostorom na disku, distribuira I/O opterećenja na sve raspoložive resurse kako bi optimizovao performanse, na taj način uklanja potrebu za ručnim I/O podešavanjem. Na primer, može se povećati veličina diska baze podataka ili da se premeste delovi baze podataka na nove uređaje bez zaustavljanja baze podataka.

2.1.2 Oracle ASM komponente za skladištenje

Oracle baza može smestiti data file kao Oracle ASM fajl u Oracle ASM grupu diskova. Slika 1 prikazuje odnos između komponenata za skladištenje u bazi podataka koja koristi Oracle ASM. Slika pokazuje odnos između Oracle ASM fajlova i data file-a, iako Oracle ASM može da skladišti i druge tipove fajlova.



Slika 1- Odnos komponenti za skladištenje u bazi koja koristi Oracle ASM

Oracle ASM komponente:

- **Oracle ASM Disks**- je uređaj za skladištenje koji je predviđen za Oracle ASM grupu diskova. Oracle ASM disk može da bude fizički disk ili particija, Logical Unit number (LUN) iz niza za skladištenje, logički volumen ili fajl koji je povezan sa mrežom. Ovi diskovi mogu biti dodati ili uklonjeni iz grupe diskova dok je baza podataka dostupna. Kada se doda disk u grupu diskova, dodeljuje mu se ime ili disk automatski dobija Oracle ASM disk ime.
- **Oracle ASM Disk Group**- kolekcija Oracle ASM diskova kojima se upravlja kao logičkom jedinicom. Strukture podataka u grupi diskova su samostalne i zauzimaju malo prostora na disku u grupi diskova. Unutar grupe, Oracle ASM je interfejs fajl sistema za Oracle database fajlove. Sadržaj fajlova koji su smešteni u grupu diskova je ravnomerno raspoređen ili striped, kako bi se eliminisalo žarište i obezbedile ujednačene performanse na diskovima.
- **Oracle ASM Files**- fajl koji se čuva u Oracle ASM grupi diskova. Oracle baza komunicira sa Oracle ASM-om fajlovima. Baza može da skladišti data file-ove, control files, online redo log files i druge tipove fajlova kao Oracle ASM fajlove. Na zahtev baze podataka Oracle ASM kreira Oracle ASM fajl i dodeljuje mu ime koje počinje znakom (+) iza kojeg sledi naziv grupe diskova (+DISK2).

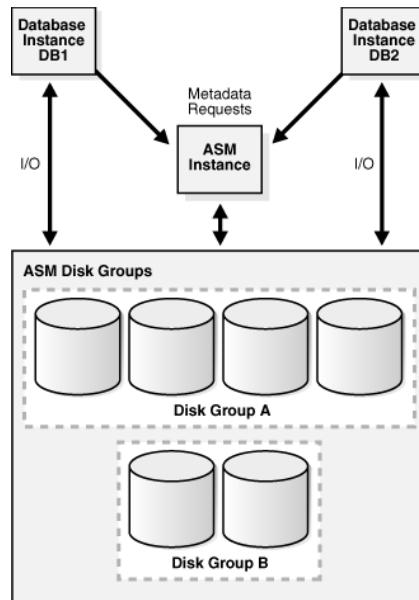
- **Oracle ASM Extents**- deo Oracle ASM fajla. Oracle ASM fajl se sastoji od jednog ili više extenata. Svaki Oracle ASM extent se sastoji od jedne ili više alocirane jedinice na određenom disku.
- **Oracle ASM Allocation Units**- osnovna jedinica alociranja unutar grupe diskova. Alocirana jedinica je najmanji neprekidni (susedni) prostor na disku koji Oracle ASM alocira. Jedna ili više alocirana jedinica formira Oracle ASM extent.

2.1.3 Oracle ASM instance

Oracle ASM instanca je specijalna Oracle instanca koja upravlja Oracle ASM dikovima. I Oracle ASM i instanca baze podataka zahtevaju zajednički pristup diskovima u Oracle ASM grupi diskova. Oracle ASM instance upravljaju metapodacima grupe podataka i pružaju informacije o rasporedu fajlova instancama baze podataka. Instance baze podataka usmeravaju I/O na Oracle ASM diskove bez korišćenja Oracle ASM instance.

Oracle ASM instanca koristi istu tehnologiju kao instanca baze podataka. Na primer, Oracle ASM instanca ima globalno područje (SGA) i pozadinske procese koji su slični onima koje ima instanca baze podataka. Međutim, instanca Oracle ASM ne može da mount-uje bazu podataka i izvodi manje zadataka od instance baze podataka.

Slika 2 prikazuje konfiguraciju sa jednim čvorom Oracle ASM instance i dve instance baze podataka. Oracle ASM instanca upravlja metapodacima i obezbeđuje alociranje prostora za Oracle ASM fajlove u kojma se čuvaju podaci za dve baze podataka.



Slika 2- Konfiguracija sa jednim čvorom ASM instance i dve instance baze podataka

2.2 Data files

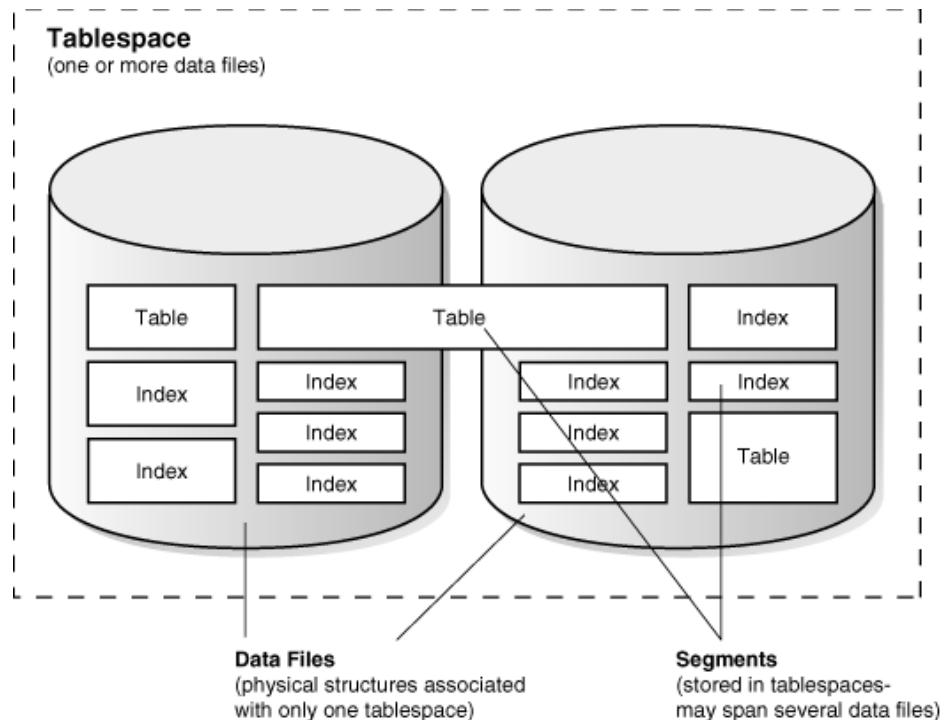
Na nivou operativnog sistema, Oracle baza skladišti podatke baze u strukture data files. Svaka Oracle baza mora da ima najmanje jedan data file.

Oracle skladišti podatke logički u tablespace-eve, a fizički u data file-ove povezane sa odgovarajućim tablespace-om.

Baze podataka, tablespaces i datafiles su blisko povezani, ali imaju značajne razlike:

- Svaki tablespace se sastoji od jednog ili više data file-ova koji se podudaraju sa operativnim sistemom na kome je pokrenuta Oracle baza.
- Podaci za bazu podataka se zajednički čuvaju u data file-ovim koji su locirani u svakom tablespace-u baze podataka.
- Segment može da obuhvati jedan ili više data file-ova, ali ne može da obuhvati više tablespace-a.
- Baza podataka mora imati SYSTEM i SYSAUX tablespaces. Oracle baza automatski alocira prve data file-ove bilo koje baze za SYSTEM tablespace tokom kreiranja baze.

Na slici 3 se može videti odnos između tablespace-a, data file-ova i segmenata.



Slika 3- Odnos između tablespace-a, data file-a i segmenta

2.2.1 Permanent i Temporary Data Files

Stalni tablespace se sastoji od perzistentnih objekata šeme. Objekti u stalnom tablespace-u se čuvaju u data file-ovima.

Privremeni tablespace sadrži objekte šeme samo za vreme trajanja sesije.

Privremeni fajlovi su slični trajnim fajlovima, sa sledećim izuzecima:

- Trajni objekti baze kao što su tabele se nikada ne čuvaju u privremenim fajlovima.
- Privremeni fajlovi su uvek setovani na NOLOGGING mod, što znači da se nad njima ne može generisati redo. Oporavak medija ne prepoznaje privremene fajlove.
- Privremeni fajl ne može biti read-only.
- Ne može se kreirati privremeni fajl sa ALTER DATABASE naredbom.
- Kada se kreira ili menja veličina privremenih fajlova, nije uvek zagarantovano alociranje prostora na disku za navedenu veličinu. Na fajl sistemima kao što su Linux i UNIX, privremeni fajlovi su kreirani kao sparse fajlovi. U ovom slučaju, blokovi diska se alociraju kada ima se pristup po prvi put, a ne prilikom kreiranja fajla ili promene veličine.

2.2.2 Online i Offline Data Files

Svaki data file je ili dostupan (online) ili nedostupan (offline).

Dostupnost pojedinačnih data file-ova ili privremenih fajlova se može promeniti tako što će biti isključeni ili uključeni. Baza podataka ne može pristupiti fajlovima koji su isključeni.

Data files mogu biti offline iz više razloga, uključujući izvršavanje offline back up-a ili ako je došlo do korupcije bloka. Baza može da isključi fajl automatski ako ne može da vrši upisivanje u njega.

Takođe, tablespace može biti offline ili online. Ako se isključi data file iz online tablespace-a, tablespace ostaje onilne.

Počevši od verzije Oracle Database 12c, može se koristiti naredba ALTER DATABASE MOVE DATAFILE naredba da se premesti online data file sa jednog fizičkog diska na drugi dok je baza podataka pokrenuta i pristupa tom fajlu. Ova tehnika se može koristiti za:

- Premeštanje tablespace-a iz jedne vrste skladišta u drugo
- Premeštanje data file-ova kojima se slabo pristupa kako bi se smanjila cena skladištenja

2.2.3 Data File Structure

Oracle baza kreira data file za tablespace tako što alocira određenu količinu prostora na disku plus overhead za data file header. Operativni sistem na kojem je pokrenuta Oracle baza je odgovoran za brisanje starih informacija i ovlašćenja iz fajlova pre nego što ih dodeli bazi podataka.

Data file header sadrži metapodatke o data file-ovima kao što su veličina i SCN kontrolne tačke. Svaki header sadrži apsolutni broj fajla koji jedinstveno identificuje data file u bazi, i relativni broj fajla koji jedinstveno identificuje fajl unutar tablespace-a.

Kada Oracle baza prvi put kreira data file, alocirani prostor na disku se formatira ali ne sadrži korisničke podatke. Međutim baza podataka zadržava prostor za čuvanje podataka za buduće segmente pridruženog tablespace-a. Kako podaci rastu unutar tablespace-a, Oracle baza koristi slobodan prostor unutar data file-ova za alociranje extenata za segment.

Na slici 4 se mogu videti različiti tipovi prostora koji se nalaze u data file-u. Extenti su ili korišćeni, što znači da sadrže podatke o segmentima, ili su slobodni što znači da su dostupni za ponovno korišćenje. Vremenom ažuriranje i brisanje objekata unutar tablespace-a može dovesti di postojanja praznog prostora koji individualno nije dovoljno veliki da bi mogao da bude ponovo korišćen za nove podatke, takav tip praznog prostora se naziva fragmented free space (fragmentisani prazni prostor).



Slika 4- Data file struktura

2.3 Control files

Kontrolni fajl baze podataka je mali binarni fajl koji je pridružen samo jednoj bazi. Svaka baza ima jedan jedinstveni kontrolni fajl, iako je dozvoljeno postojanje više identičnih kopija.

Oracle koristi kontrolne fajlove za lociranje fajlova baze podataka i za upravljanje uopštenim stanjem baze. Kontrolni fajl sadrži sledeće informacije:

- Ime baze i jedinstveni identifikator baze (DBID)

- Time stamp kreiranja baze
- Informacije o data file-ovima, online redo log file-ovima i archived redo log file-ovima
- Informacije o tablespace-u
- RMAN backup

Kontrolni fajlovi služe za:

- Sadrže informacije o data file-ovima, online redo log file-ovima itd. koje su potrebne za pokretanje baze podataka. Kontrolni fajl prati promene u strukturi baze podataka. Na primer, kada administrator doda, promeni ime, ili obriše data file ili online redo log file, baza podataka ažurira kontrolni fajl tako da on odražava ovu promenu.
- Sadrže metapodatke koji moraju biti dostupni i kada baza podataka nije pokrenuta. Na primer, kontrolni fajl sadrži informacije koje su potrebne za oporavak baze, uključujući ključne tačke (checkpoints). Ključna tačka ukazuje na SCN u redo stream-u gde bi oporavak instance trebao da otpočne. Svaka izvršena promena pre checkpoint-a SCN je zagarantovano snimljena na disku u data file-ovima. Proces checkpoint-a najmanje na svake tri sekunde beleži informacije u kontrolnu datoteku o položaju kontrolne tačke u online redo log-u.

Oracle baza čita i upisuje u kontrolni fajl neprekidno tokom korišćenja baze podataka i mora biti dostupan upis kad god je baza podataka pokrenuta.

Oracle baza omogućava istovremeno otvaranje identičnih kontrolnih fajlova i njihovo upisivanje u istu bazu podataka. Multipliciranjem kontrolnog fajla na različitim diskovima, baza može postići redundantnost.

2.3.1 Control File struktura

Informacije o bazi podataka se čuvaju u različitim sekcijama kontrolnog fajla. Svaka sekcija je skup rekorda o aspektu baze podataka. Kontrolni fajlovi sadrže sledeće tipove rekorda:

- **Circural reuse records**- sadrže informacije koje nisu kritične i koje mogu da se zamene drugim informacijama ako je potrebno. Kada su sva raspoloživa mesta rekorda puna, baza podataka ili proširuje kontrolni fajl da bi napravila mesta za novi rekord ili upisuje preko najstarijeg rekorda. (archived redo log, RMAN backup)
- **Noncircular reuse records**- sadrže ključne informacije koje se ne menjaju često i preko tih informacija ne mogu se upisati druge. Primeri ovih informacija uključuju tablepsace, data file, online redo log file i redo niti. Oracle baza nikada ne koristi ove rekorde osim ako je odgovarajući objekat uklonjen iz tablespace-a.

2.4 Online Redo Log

Ključna struktura za oporavak baze je online redo log, koji se sastoji od dva ili više prealocirana fajla koji pamte promene baze podataka onda kada se one dogode, takođe čuvaju i promene nad data file-ovima.

Baza podataka održava online redo log fajlove za zaštitu protiv gubitka podataka. Nakon što se desi greška instance, online redo files omogućavaju bazi da obnovi upisane podatke koji još uvek nisu upisani u data files.

Online redo log se sastoji od dva ili više online redo log fajla. Oracle baza zahteva minimum dva fajla kako bi bilo zagarantovano da je jedan fajl uvek dostupan za upis u slučaju da je drugi fajl u procesu pražnjenja ili arhiviranja.

2.4.1 Archived Redo Log Files

Archived redo log file je kopija popunjenoog člana online redo log grupe. Fajl se ne smatra delom baze podataka, ali je offline kopija online redo log fajla kreiranog od strane baze podataka i upisanog u lokaciju koju je specificirao korisnik.

Archived redo log fajlovi su krucijalni za strategije oporavka i backup-a baze. Mogu se koristiti za:

- Oporavak backup-a baze podataka
- Ažuriranje standby baze
- Prikupljanje informacija o istoriji baze podataka korišćenjem LogMiner mogućnosti

Operacija generisanja archived log fajla se naziva arhiviranje (archiving). Ova operacija može biti automatska ili ručna, moguća je jedino ako se baza podataka nalazi u ARCHIVELOG modu.

2.4.2 Online Redo Log struktura

Online redo log file sadrži redo rekorde. Redo rekord je sačinjen od grupe vektora promena, od kojih svaka opisuje promenu u data block-u.

Redo rekordi sadrže sve relevantne metapodatke o promenama, uključujući:

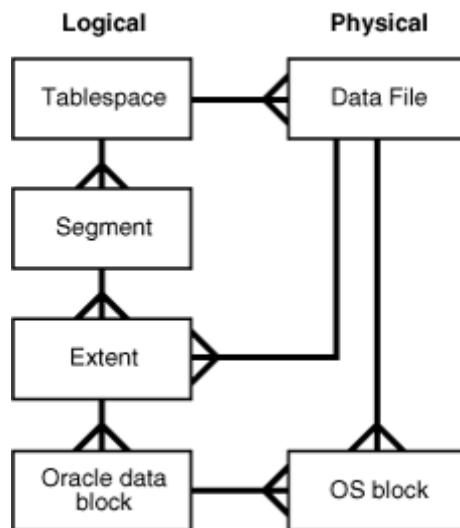
- SCN i time stamp promene
- Transaction ID transakcija koje je generisala promenu
- SCN i time stamp kada je transakcija izvršena (ako je uopšte izvršena)
- Tip operacije koji je izazvao promenu
- Naziv i tip modifikovanog segmenta podataka

3. Logičke strukture za skladištenje podataka

Oracle baza podataka alocira logički prostor za sve podatke koji se nalaze u bazi.

Logičke jedinice koje se alociraju su: **data blocks**, **extents**, **segments** i **tablespaces**. Na fizičkom nivou, podaci su smešteni u **data files** na disku. Podaci u data file-ovima su smešteni u blokove operativnog sistema.

Na slici 5 je prikazan odnos entiteta između logičkog i fizičkog skladišta.



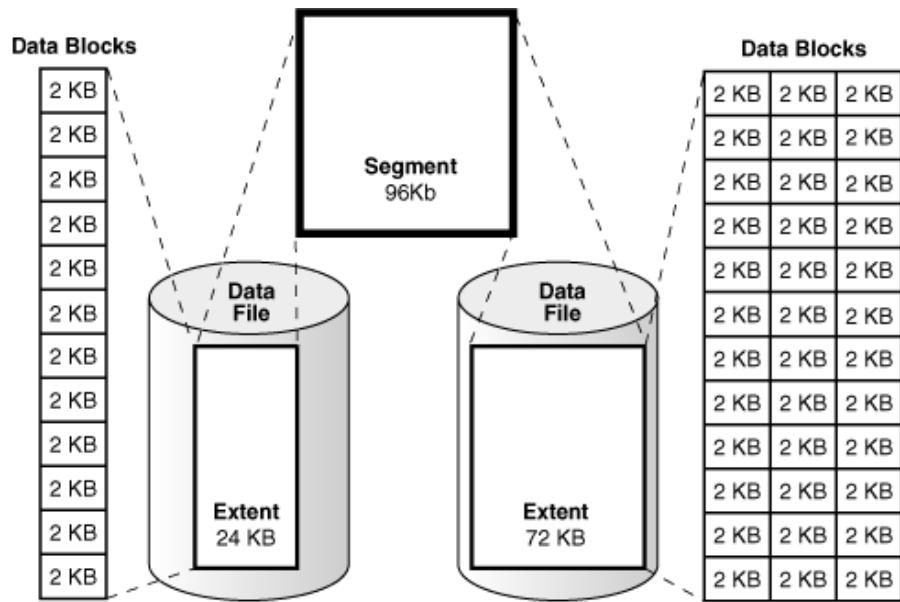
Slika 5- Logičko i fizičko skladištenje podataka

3.1 Hijerarhija logičke strukture

Oracle baza skladišti podatke na sledeći način, počevši od stepena najniže granularnosti:

- **Data block**- najniža logička jedinica koja se koristi za skladištenje. Jedan data block odgovara specifičnom broju bajtova na fizičkom disku, na Slici 6 je to na primer 2KB. Data block su najmanje jedinice koje Oracle baza može da alocira i koristi za skladištenje podataka.
- **Extent**- niz logički susednih data block-ova alociranih za skladištenje posebnih (specifičnih) tipova informacija. Na Slici 6, 24 KB extent ima 12 data block-ova, dok 72 KB extent ima 36 data block-ova.
- **Segment**- je skup extenta alociranih za specifičan objekat baze podataka, kao što je tabela.
- **Tablespace**- jedinica baze podataka koja sadrži jedan ili više segmenata.

Svaki segment pripada samo jednom tablespace-u, što znači da su svi extenti jednog segmenta smešteni u istom tablespace-u. U sklopu tablespace-a, segment može da sadrži extente koji su iz različitih data file-ova, dok svaki extent može da sadrži podatke iz samo jednog data file-a. Primer: Jedan extent segmenta može biti uskladišten u users1.dbf, a drugi u users2.dbf.



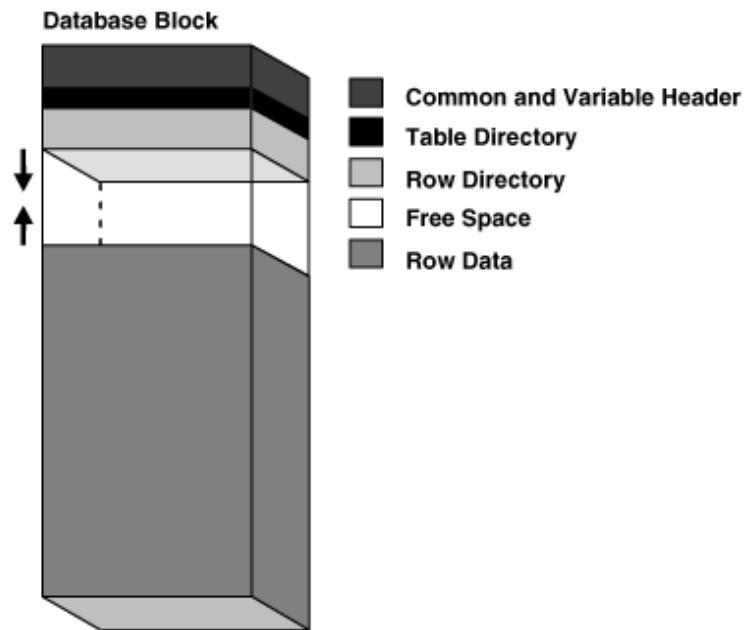
Slika 6- Hjерархија логичке структуре

3.2 Data Blocks

Data block je najmanja jedinica Oracle baze podataka koja se koristi za logičko skladištenje podataka. Format data block-a je sličan nezavisno od toga da li sadrži tabelu, indeks ili klasterizovane podatke.

Format data block-a (Slika 7):

- **Common and Variable Header**- sadrži opšte informacije o bloku, kao što su adresa bloka i tip segmenta (data, index, rollback)
- **Table Directory**- sadrži metapodatke o tabelama čiji su redovi smešteni u bloku
- **Row Directory**- sadrži informacije o stvarnim redovima koji se nalaze u bloku (uključujući adrese za svaki deo reda u **row data** delu)
- **Row Data**- deo data block-a koji sadrži stvarne podatke, kao što su redovi tabele ili ključevi indeksa
- **Overhead**- data block header, table directory i row directory se zajedno nazivaju overhead. Neki delovi overhead-a su fiksne veličine, ali ukupna veličina je promenljiva. Prosečna veličina je od 84 do 107 bajta.
- **Free Space**- alocira se za umetanje novih redova i ažuriranje redova koji zahtevaju dodatni prostor

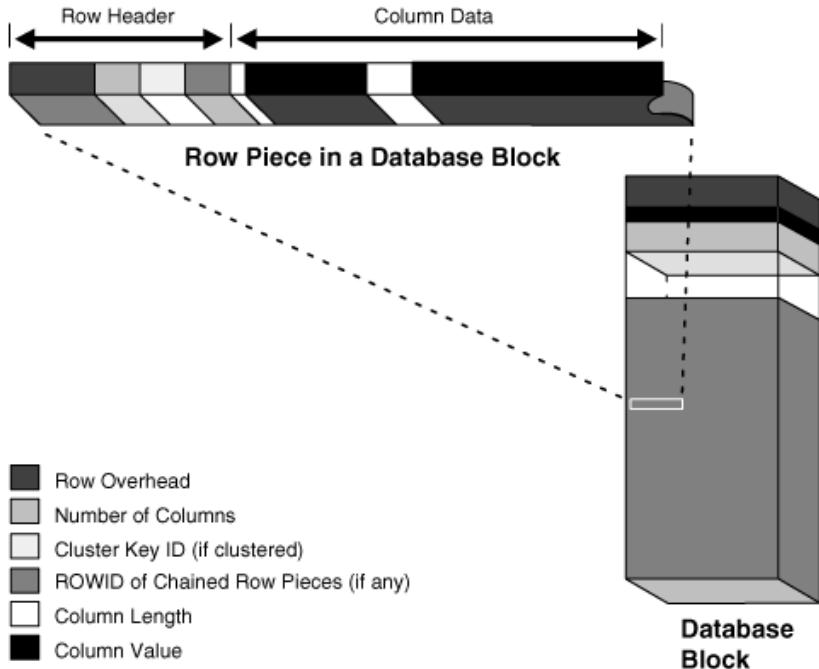


Slika 7- Format data block-a

3.2.1 Row Format

Kao što svaki data block ima svoju internu strukturu, svaki red ima svoj format koji omogućava bazi da prati podatke koji se nalaze u tom redu.

Oracle skladišti redove kao rekorde promenljivih veličina. Red je sadržan u jednoj ili više sekcija, svaka sekcija se naziva row piece. Svaki row piece ima row header i column data. Ovo se može videti na slici 8.



Slika 8- Row format

Row Header

Oracle baza koristi row header-e da upravlja row piece-vima koji su u bloku. Sadrži sledeće informacije:

- Kolone koje se nalaze u row piece-u
- Delovi reda koji su locirani u drugim data block-ovima. Ako se ceo red može smestiti u jedan data block, Oracle smešta red kao jedan row piece. Ali ako se svi podaci jednog reda ne mogu smestiti u jedan data block ili ako bi update tog reda doveo do toga da red preraste data block u kome se nalazi, tada baza skladišti red u višestruke row piece-eve. Data block uglavnom sadrži samo jedan row piece po redu.
- Cluster keys za klasterne tabele

Red koji je u potpunosti sadržan u jednom data block-u ima najmanje tri bajta row header-a.

Column Data

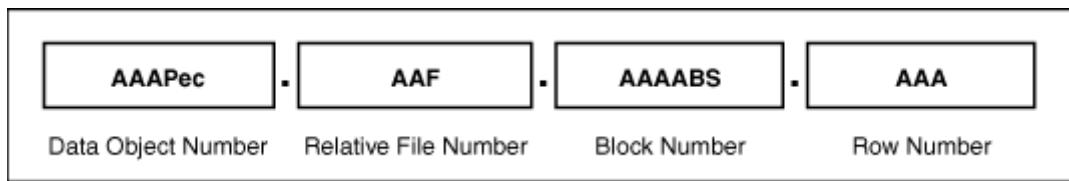
Column data sekcija sadrži prave podatke koji se nalaze u redovima. Row piece uglavnom skladišti kolone onako kako su navedene u CREATE TABLE komandi, ali to nije zagarantovan redosled (kolone tipa LONG se kreiraju poslednje).

Kao što se može videti na slici 8, za svaku kolonu u row piece-u, Oracle skladišti dužinu kolone i podatke odvojeno, prostor koji je potreban za smeštanje zavisi od tipa podataka, ako je tip podataka varijabilne veličine, prostor može da raste i da se smanjuje prilikom ažuriranja podataka. Svaki red ima mesto u row directory u data block header-u, to mesto ukazuje na početak reda.

Rowid Format

Oracle baza koristi **rowid** da jedinstveno identificuje red. Interno rowid je struktura koja čuva informacije koje baza koristi da pristupi redu. Rowid se fizički ne skladišti u bazi.

Extended rowid uključuje i broj data object-a. Ovaj tip koristi base 64 enkodiranje fizičke adrese za svaki red. Format extended rowid-a se može videti na slici 9.



Slika 9- Primer extended rowid formata

Extended rowid je prikazan u formatu koji ima 4 dela, OOOOOOFFBBBBBRRR gde je format podeljen na sledeće komponente:

- OOOOO- Data object broj koji identificuje segment (data object AAAPec na slici 9). Ovaj broj se dodeljuje svakom segmentu baze. Objekti šeme koji se nalaze u istom segmentu imaju isti data object broj.
- FFF- Tablespace relativni data file broj koji identificuje data file koji sadrži red (fajl AAF na slici 9)
- BBBB- Data block broj identificuje block koji sadrži red (block AAAABS na slici 9). Block brojevi su relativni u odnosu na njihov data file, a ne u odnosu na tablespace. Dva reda koja imaju identični block broj mogu da budu u različitim data file-ovima istog tablespace-a.
- RRR- Row broj identificuje red u bloku (red AAA na slici 9)

Nakon što se rowid dodeli row piece-u, rowid može da se promeni pod specijalnim okolnostima.

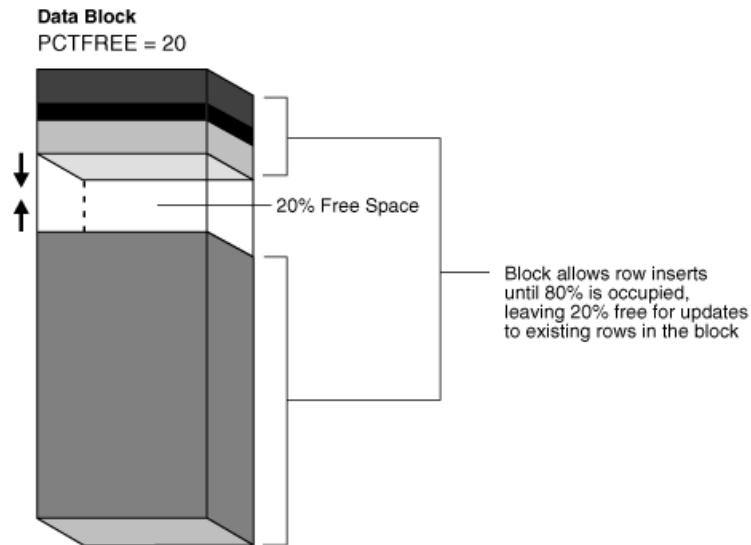
3.2.3 Data blocks- upravljanje prostorom

Dva parametra za upravljanje prostorom, **PCTFREE** i **PCTUSED**, omogućavaju kontrolisanje slobodnog prostora za umetanje i ažuriranje redova u svima data block-ovima određenog segmenta. Ovi parametri se specificiraju prilikom kreiranja (CREATE) ili menjanja tabele (ALTER).

PCTFREE Parametar

Ovim parametrom se podešava minimalni procenat data block-a koji se rezerviše kao slobodan prostor za moguće ažuriranje redova koji već postoje u tom bloku.

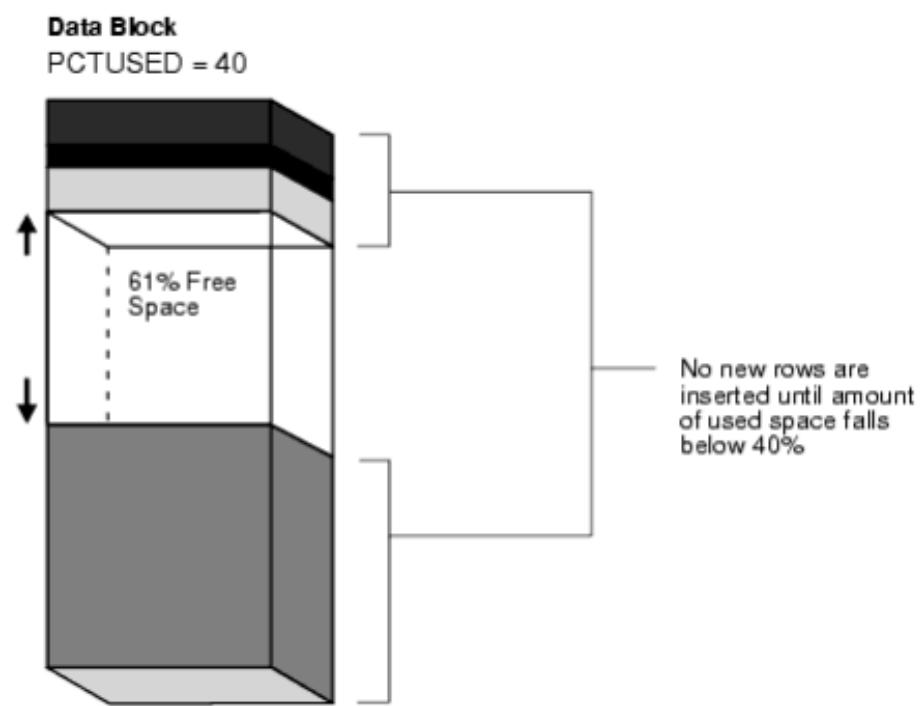
Na primer, ako se specificira da je PCTFREE 20 unutar CREATE naredbe, to znači da će 20% svakog data block-a u segmentu ove tabele biti slobodno i dostupno za moguće ažuriranje već postojećih redova unutar svakog bloka.



Slika 10- PCTFREE parametar

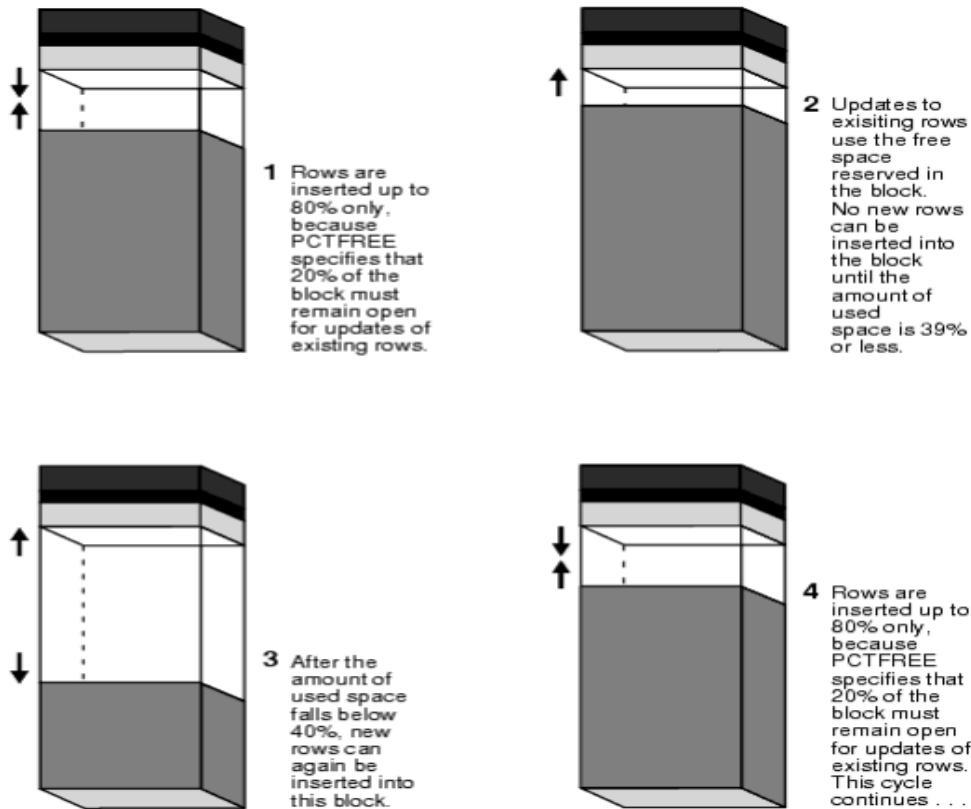
PCTUSED Parametar

Ovaj parametar setuje minimalni procenat bloka koji može biti iskorišćen za redove plus overherad pre nego što novi redovi budu dodati u blok. Nakon što je data block popunjeno do svog limita određenog parametrom PCTFREE, Oracle smatra da je blok nedostupan za umetanje novih redova sve dok se procenat tok bloka ne smanji ispod vrednosti parametra PCTUSED. Sve dok ne dođe do toga, Oracle koristi slobodan prostor data block-a samo za ažuriranje redova koji su već sadržani u data block-u. Na primer ako se specificira PCTUSED 40 tokom kreiranja tabele, data block koji se koristi za segment ove tabele se smatra nedostupnim za ubacivanje novih redova sve dok količina iskorišćenog prostora ne spadne na 39% ili manje, prepostavljajući da je iskorišćeni prostor prethodno dostigao PCTFREE parametar.



Slika 11- PCTUSED parametar

PCTFREE i PCTUSED rade zajedno kako bi se optimizovala iskorišćenost prostora u data block-ovima extenta unutar data segmenta. Na slici 12 je prikazana interakcija ova dva parametra.



Slika 12- PCTFREE i PCTUSED parametar

U novoalociranom data block-u, prostor dostupan za umetanje je veličine=block size – (block overheard + PCTFREE). Ažuriranje postojećih podataka može koristiti bilo koji raspoloživi prostor u bloku, stoga ažuriranje može dovesti do smanjenja raspoloživog prostora bloka na manje od vrednosti parametra PCTFREE, prostora rezervisanog za ažuriranje ali nedostupnog za unošenje.

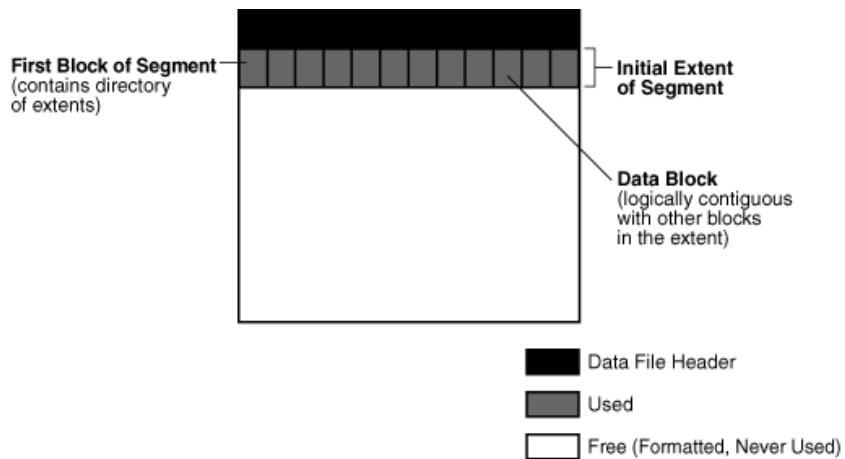
Za svaki data segment i index segment, Oracle održava jednu ili više free lists- liste data block-ova koje su alocirane za extente tog segmenta i imaju raspoloživi prostor veći od PCTFREE, ovi blokovi su dostupni za dodavanje. Kada se izda naredba INSERT, Oracle proverava free list tabele za prve slobodne data block-ove i koristi ih ako je to moguće. Ako raspoloživi prostor tog bloka nije dovoljno veliku za INSERT naredbu, a blok je barem PCTUSED, Oracle sklanja takav blok sa free liste.

Nakon što se izda naredba DELETE ili UPDATE, Oracle procesira naredbu i proverava da li je prostor koji se koristi u bloku trenutno manji od PCTUSED, ako jeste blok se stavlja na početak transakcione free liste i postaje prvi slobodna blok za korišćenje u toj transakciji. Kada se transakcija izvrši, slobodan prostor u bloku postaje dostupan za druge transakcije.

3.3 Extents

Extent je jedinica za skladištenje podataka koja se sastoji od logički susednih data block-ova.

Baza podataka podrazumevano alocira početni extent za data segment kada se segment kreira. Extent je uvek sadržan u jednom data file-u. Iako podaci nisu dodati u segment, data block-ovi u početnom extentu su rezervisani isključivo za ovaj segment. Prvi data block svakog segmenta sadrži dikrektorijum extent-a u segmentu. Na slici 13 se vidi inicijalni extent u segmentu u data file-u koji prethodno nije sadržao podatke.



Slika 13- Inicijalni extent

Ako početni extent postane popunjten, i ako je više prostora potrebno, baza automatski alocira inkrementalni extent za taj segment. Inkrementalni extent je naknadni extent kreiran za segment koji je ili iste ili veće veličine nego prethodno alocirani extent za taj segment.

Svaki segment je definisan storage parametrima koji su izraženi u obliku extenta. Ovi parametri kontrolišu kako Oracle dodeljuje slobodan prostor za segmente.

Storage podešavanja su poređana po redosledu prioriteta:

- Segment storage clause
- Tablespace storage clause
- Oracle Database default

Tablespace koji upravlja extentima lokalno, može da ima ili uniformne veličine extenta ili varijabilne veličine koje se određuju automatski od strane sistema. Kada se kreira tablespace, naredbe UNIFORM ili AUTOLOCATE specificiraju tip alokacije. Za system-managed extente, može se specificirati veličina inicijalnog extenta i Oracle određuje optimalnu veličinu za dodatne extente, sa minimalnom veličinom extenta 64KB. Za uniformne extente može se specificirati veličina extenta ili da se koristi default veličina

koja je 1MB. Storage parametri NEXT, PCTINCREASE, MINEXTENTS, MAXEXTENTS, DEFAULT STORAGE nisu validni za extente kojima se upravlja lokalno.

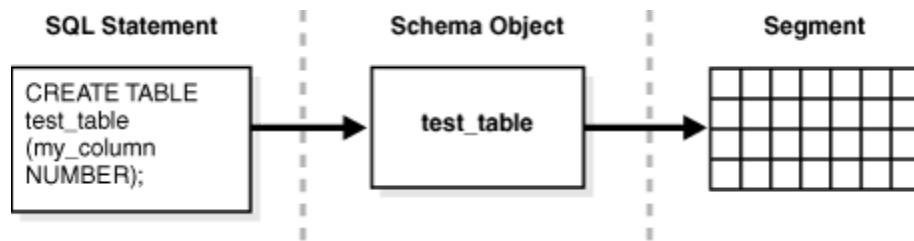
3.4 Segments

Segment je skup extenta koji sadrže sve podatke za logičku strukturu unutar tablespace-a. Oracle baza koristi četiri vrste segmenata: Data segment, Index segment, Temporary segment, Rollback segment.

Jedan data segment u Oracle bazi čuva podatke za jedan korisnički objekat. Primeri za user segmente uključuju:

- Table, table partition ili table cluster
- LOB ili LOB partition
- Index or index partition

Podrazumevano baza podataka koristi odloženo kreiranje segmenata za ažuriranje samo metapodataka baze prilikom kreiranja tabele, indeksa i particija. Kada korisnik uneše prvi red u tabelu ili particiju, baza kreira segmente za bazu ili particiju, njene LOB kolone i njene indekse. Odloženo kreiranje segmenata izbegava nepotrebno korišćenje resursa baze.



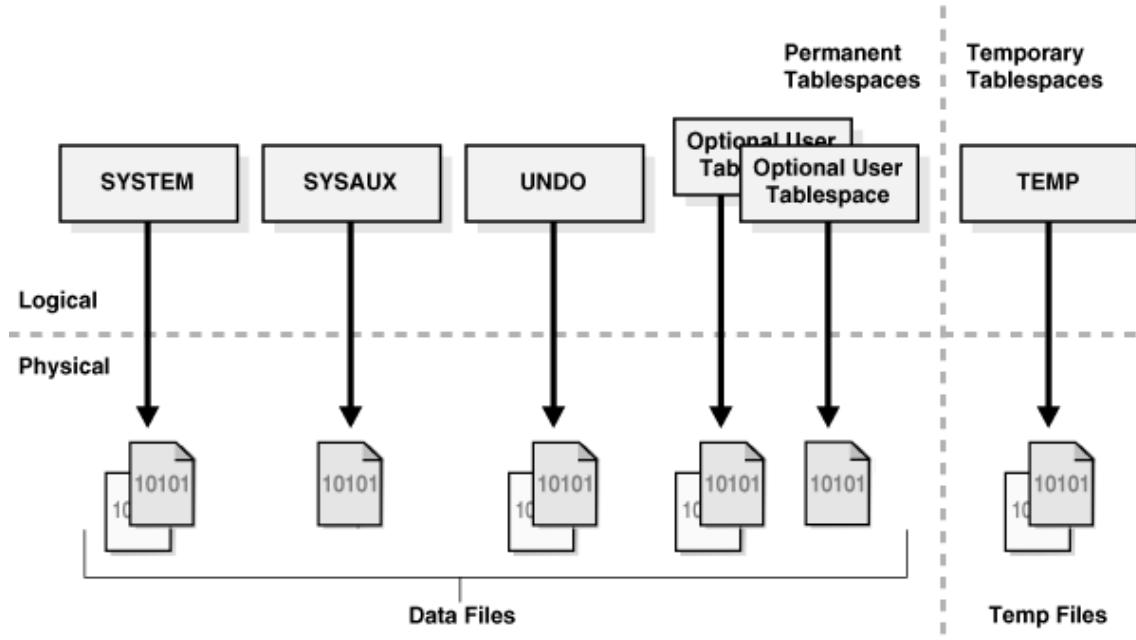
Slika 14- Segment

Temporary segment- prilikom procesiranja upita Oracle baza često zahteva korišćenje privremenog workspace-a za srednje faze izvršenje SQL naredbi. Oracle zahteva temporary segment kao radno područje za sortiranje, ovaj segment se ne kreira ukoliko operacija može biti izvršena u memoriji.

Rollback segment- svaka baza sadrži jedan ili više rollback segmenata. Rollback segment beleži stare vrednosti podataka pre nego što su bile promenjene svakom transakcijom (bez obzira da li su izvršene). Ovi segmenti se koriste za obezbeđivanje read consistency, vraćanje transakcija i za oporavak baze podataka.

3.5 Tablespace

Tablespace je logičko skladište za segmente. Na fizičkom nivou, tablespace skladišti podatke u jedan ili više data file-a ili temp file-a. Baza podataka mora da ima SYSTEM i SYSAUX tablespace. Na slici 15 se vide tipovi tablespace-a.



Slika 15- Tablespace

Permanent tablespace grupiše stalne objekte šeme. Tablespace može da se koristi za postizanje sledećih stvari:

- Kontrolisanje alociranog prostora diska za podatke baze
- Dodeljivanje limitiranog prostora korisniku baze
- Uključivanje i isključivanje individualnih tablespace-eva bez uticanja na dostupnost cele baze
- Izvođenje backup-a i oporavka individualnih tablespace-va
- Importovanje i exportovanje podataka baze
- Kreiranje transportable tablespace-a koje može da se premešta iz jedne baze u drugu

SYSTEM tablespace je neophodni administrativni tablespace koji je uključen u bazu podataka prilikom njenog kreiranja, koristi se za upravljanje bazom podataka. Sadrži sledeće informacije: data dictionary, tabele i poglede koji sadrže administrativne informacije o bazi, kompajlirane objekte kao što su trigeri, procedure i paketi.

SYSAUX tablespace je pomoći tablespace za SYSTEM tablespace. Ovaj tablespace je podrazumevani tablespace za mnoge funkcije Oracle baze koje su prethodno zahtevale sopstveni tablespace, smanjuje broj tablespace-a potreban za bazu podataka i umanjuje opterećenje SYSTEM tablespace-a.

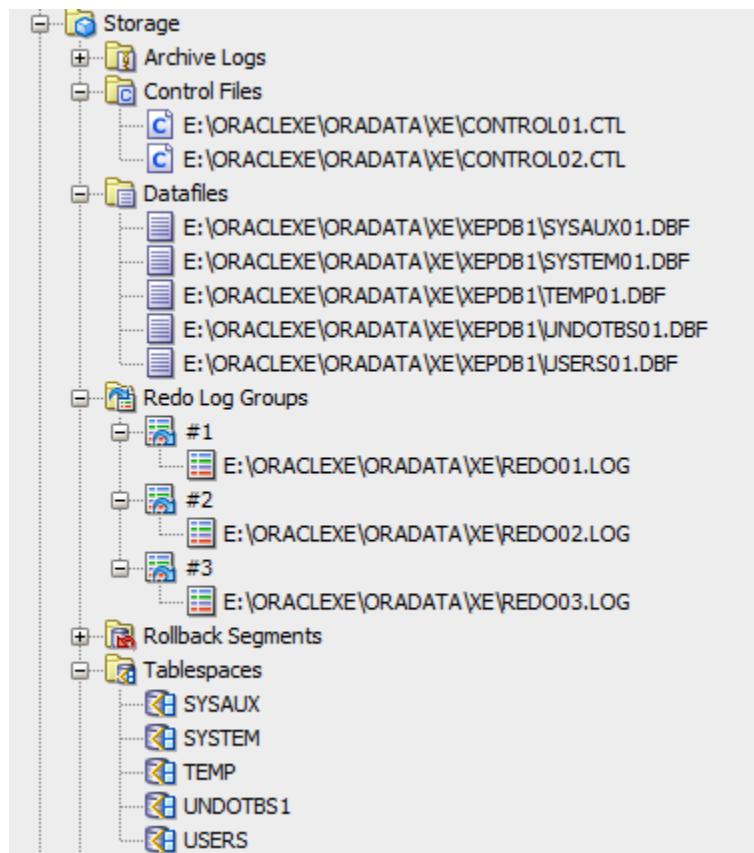
Undo Tablespace je tablespace kojim se lokalno upravlja i koji je rezervisan za system-managed undo podatke.

Temporary tablespace sadrži prolazne podatke koji se zadržavaju samo tokom trajanja sesije, nijedan trajni objekat šeme ne može ovde da se nađe. Poboljšava istovremenost više operacija sortiranja koje nemaju mesta u memoriji i poboljšava efikasnost upravljanja prostorom za vreme trajanja operacija sortiranja.

Tablespace može biti ili bigfile tablespace ili smallfile tablespace, ovi tablespace-ovi se ne mogu razlikovati u smislu izvršenja SQL naredbi koje se izričito ne odnose na data files ili temp files.

4. Primeri

Korišćenjem Oracle SQL developer-a moguće je videti logičku i fizičku strukturu skladišta baze podataka. Na slici 20 prikazani su: Control files, Datafiles, Redo Log Files, Tablespaces:



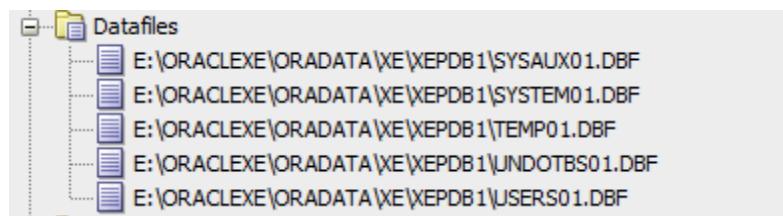
Slika 16- Fizičke structure za skladištenje

Detaljnije informacije o control file-ovima:

Name	Value
1 File Path	E:\ORACLEXE\ORADATA\XE\CONTROL01.CTL
2 Status	VALID
3 File Name	CONTROL01.CTL
4 File Directory	E:\ORACLEXE\ORADATA\XE\
5 Created in Flash Recovery Area	NO

Slika 17- Informacije o jednom control fajlu

Prikaz data file-ova i njihova ispunjenost (Slika 18 i 19)



Slika 18- Prikaz data file-ova

File Name	Tablespace	Status	Size (MB)	Used (MB)	Used (Proportion)	Used (%)	Auto Extend
1 E:\ORACLEXE\ORADATA\XE\XEPDB1\SYSAUX01.DBF	SYSAUX	ONLINE	390.000	365.688	███████	93.77	YES
2 E:\ORACLEXE\ORADATA\XE\XEPDB1\SYSTEM01.DBF	SYSTEM	SYSTEM	260.000	255.875	███████	98.41	YES
3 E:\ORACLEXE\ORADATA\XE\XEPDB1\TEMP01.DBF	TEMP	ONLINE	129.000	0.000	██████	0	YES
4 E:\ORACLEXE\ORADATA\XE\XEPDB1\UNDOTBS01.DBF	UNDOTBS1	ONLINE	100.000	9.063	██████	9.06	YES
5 E:\ORACLEXE\ORADATA\XE\XEPDB1\USERS01.DBF	USERS	ONLINE	7.500	6.125	██████	81.67	YES

Slika 19- Ispunjenošć data file-ova

Informacije o jednom data file-u:

Name	Value
1 Name	E:\ORACLEXE\ORADATA\XE\XEPDB1\SYSAUX01.DBF
2 Tablespace	SYSAUX
3 Status	ONLINE
4 File Size (MB)	390.00
5 Auto Extend	YES
6 Increment (MB)	10.00
7 Maximum File Size (MB)	32767.98

Slika 20- Informacije o data file-u

Prikaz informacija o redo log grupama:

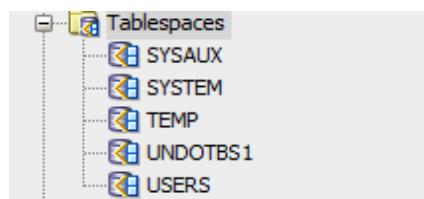
Group	Thread	Status	# of Members	Archived	Size (KB)	Sequence	First Changed #
1 #1	1	Inactive	1	No	204800	4	1607261
2 #2	1	Current	1	No	204800	5	1759926
3 #3	1	Inactive	1	No	204800	3	1494335

Slika 21- Prikaz redo log grupe

File	Type	Created in Flash Recovery Area
1 E:\ORACLEXE\ORADATA\XE\REDO01.LOG	ONLINE	NO

Slika 22- Informacije o jednom redo log fajlu

Prikaz tablespace-va i njihova ispunjenost:



Slika 23- Prikaz tablespace-va baze podataka

Tablespace Name	Allocated (MB)	Free (MB)	Used (MB)	% Free	% Used	Max. Bytes (MB)
1 USERS	8	1	6	18	82	32768
2 UNDOTBS1	100	69	31	69	31	32768
3 TEMP	129	129	0	100	0	32768
4 SYSTEM	260	4	256	2	98	32768
5 SYSAUX	390	24	366	6	94	32768

Slika 24- Detalji o tablespace-vima

Kreirane su dve potpuno identične tabele ROBA2 i ROBA3, jedina razlika prilikom kreiranja tabela je PCTFREE parametar:

```
CREATE TABLE "MAJA2"."ROBA2"
(
    "SIFRA" NUMBER(15,0) NOT NULL ENABLE,
    "NAZIV" VARCHAR2(50 CHAR) NOT NULL ENABLE,
    "PG" NUMBER(2,0),
    "GRUPA" NUMBER(4,0),
    "JM" VARCHAR2(5 CHAR),
    "VR_DOBRA" NUMBER(1,0),
    "BARCOD" NUMBER,
    "FP" NUMBER(11,5),
    "JM_PROD" VARCHAR2(5 CHAR),
    "ROBA" NUMBER(1,0),
    "SIFRA_LDPU" NUMBER(6,0),
    "FP_LDPU" NUMBER(11,5),
    "SIFRA_AMB" NUMBER(6,0),
    "ROK_PLACANJA" NUMBER(4,0),
    "VAGA" NUMBER(1,0),
    "ROBA_ID" NUMBER(10,0),
    "ID_RB" NUMBER(10,0),
    "NEAKTIVAN" NUMBER(1,0),
    "SOPSTVENI" NUMBER(1,0),
    "LDPU" NUMBER(1,0),
    "HRANA" NUMBER(1,0),
    "KAT_BR" VARCHAR2(30 CHAR),
    "DIMENZIJA" VARCHAR2(10 CHAR),
    "JM_KOEF" NUMBER(10,5),
    "JM_JEDINICNA" VARCHAR2(5 CHAR),
    "BELESKA" VARCHAR2(300 CHAR),
    "TPAK" NUMBER(10,0),
    "TEZINA" NUMBER(20,5),
    "BARCOD_TPAK" NUMBER(20,0)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITTRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;
```

Slika 25- SQL naredba za kreiranje tabele ROBA2

```

CREATE TABLE "MAJA2"."ROBA3"
(
    "SIFRA" NUMBER(15,0) NOT NULL ENABLE,
    "NAZIV" VARCHAR2(50 CHAR) NOT NULL ENABLE,
    "PG" NUMBER(2,0),
    "GRUPA" NUMBER(4,0),
    "JM" VARCHAR2(5 CHAR),
    "VR_DOBRA" NUMBER(1,0),
    "BARCOD" NUMBER,
    "FP" NUMBER(11,5),
    "JM_PROD" VARCHAR2(5 CHAR),
    "ROBA" NUMBER(1,0),
    "SIFRA_LDPU" NUMBER(6,0),
    "FP_LDPU" NUMBER(11,5),
    "SIFRA_AMB" NUMBER(6,0),
    "ROK_PLACANJA" NUMBER(4,0),
    "VAGA" NUMBER(1,0),
    "ROBA_ID" NUMBER(10,0),
    "ID_RB" NUMBER(10,0),
    "NEAKTIVAN" NUMBER(1,0),
    "SOPSTVENI" NUMBER(1,0),
    "LDPU" NUMBER(1,0),
    "HRANA" NUMBER(1,0),
    "KAT_BR" VARCHAR2(30 CHAR),
    "DIMENZIJA" VARCHAR2(10 CHAR),
    "JM_KOEF" NUMBER(10,5),
    "JM_JEDINICNA" VARCHAR2(5 CHAR),
    "BELESKA" VARCHAR2(300 CHAR),
    "TPAK" NUMBER(10,0),
    "TEZINA" NUMBER(20,5),
    "BARCOD_TPAK" NUMBER(20,0)
) SEGMENT CREATION IMMEDIATE
PCTFREE 30 PCTUSED 40 INITTRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;

```

Slika 26- SQL naredba za kreiranje tabele ROBA3

Tabele su napunjene potpuno identičnim podacima i istim brojem slogova. Na sledećim slikama se nalazi statistika za obe tabele gde se jasno vidi različita količina upotrebljenih blokova za skladištenje podataka.

Name	Value
1 NUM_ROWS	277568
2 BLOCKS	3025
3 AVG_ROW_LEN	75
4 SAMPLE_SIZE	277568
5 LAST_ANALYZED	14-APR-21
6 LAST_ANALYZED_SINCE	14-APR-21

Slika 27- Statistika za tabelu ROBA2

Name	Value
1 NUM_ROWS	277568
2 BLOCKS	3898
3 AVG_ROW_LEN	75
4 SAMPLE_SIZE	277568
5 LAST_ANALYZED	14-APR-21
6 LAST_ANALYZED_SINCE	14-APR-21

Slika 28- Statistika za tabelu ROBA3

Detaljniji prikaz statistike obe tabele:

Name	Value
1 CREATED	14-APR-21
2 LAST_DDL_TIME	14-APR-21
3 OWNER	MAJA2
4 TABLE_NAME	ROBA2
5 TABLESPACE_NAME	USERS
6 CLUSTER_NAME	(null)
7 IOT_NAME	(null)
8 STATUS	VALID
9 PCT_FREE	10
10 PCT_USED	(null)
11INI_TRANS	1
12 MAX_TRANS	255
13 INITIAL_EXTENT	65536
14 NEXT_EXTENT	1048576
15 MIN_EXTENTS	1
16 MAX_EXTENTS	2147483645
17 PCT_INCREASE	(null)
18 FREELISTS	(null)
19 FREELIST_GROUPS	(null)
20 LOGGING	YES
21 BACKED_UP	N
22 NUM_ROWS	277568
23 BLOCKS	3025
24 EMPTY_BLOCKS	47
25 AVG_SPACE	964
26 CHAIN_CNT	0
27 AVG_ROW_LEN	75
28 AVG_SPACE_FREELIST_BLOCKS	0
29 NUM_FREELIST_BLOCKS	0
30 DEGREE	1
31 INSTANCES	1
32 CACHE	N
33 TABLE_LOCK	ENABLED
34 SAMPLE_SIZE	277568

Slika 29- Detalji tabele ROBA2

Name	Value
1 CREATED	14-APR-21
2 LAST_DDL_TIME	14-APR-21
3 OWNER	MAJA2
4 TABLE_NAME	ROBA3
5 TABLESPACE_NAME	USERS
6 CLUSTER_NAME	(null)
7 IOT_NAME	(null)
8 STATUS	VALID
9 PCT_FREE	30
10 PCT_USED	(null)
11INI_TRANS	1
12 MAX_TRANS	255
13 INITIAL_EXTENT	65536
14 NEXT_EXTENT	1048576
15 MIN_EXTENTS	1
16 MAX_EXTENTS	2147483645
17 PCT_INCREASE	(null)
18 FREELISTS	(null)
19 FREELIST_GROUPS	(null)
20 LOGGING	YES
21 BACKED_UP	N
22 NUM_ROWS	277568
23 BLOCKS	3898
24 EMPTY_BLOCKS	70
25 AVG_SPACE	2484
26 CHAIN_CNT	0
27 AVG_ROW_LEN	75
28 AVG_SPACE_FREELIST_BLOCKS	0
29 NUM_FREELIST_BLOCKS	0
30 DEGREE	1
31 INSTANCES	1
32 CACHE	N
33 TABLE_LOCK	ENABLED
34 SAMPLE_SIZE	277568

Slika 30- Detalji tabele ROBA3

5.Zaključak

U ovom radu je dat osnovni prikaz fizičkih i logičkih struktura koje Oracle baza koristi za skladištenje podataka. Takođe je objašnjena međusobna povezanost i hijerarhija fizičkih i logičkih struktura, objašnjena je uloga pojedinih fajlova kreiranih u operativnom sistemu. Mehanizam kojim Oracle baza upravlja skladишtem podataka sastoji se od brojnih komponenti koje omogućavaju siguran i pouzdan rad, kao i maksimalnu sigurnost podataka. U sistemu postoje mehanizmi kojima se može obezbediti redundantnost osetljivih podataka.

Primeri koji su dati u radu prikazuju parametre kojima korisnik baze podataka može da utiče na korišćenje prostora kojim upravlja Oracle baza. Korisnik ove parametre može podešiti shodno svojim potrebama i u zavisnosti od prirode i načina upisivanja podataka u tabele baze podataka (frekvencija i broj dodatih slogova, frekvencija i broj brisanih slogova itd.).

6.Literatura

- <https://docs.oracle.com/en/database/oracle/oracle-database/18/index.html>
- https://docs.oracle.com/cd/A87860_01/doc/server.817/a76965/c03space.htm
- https://docs.oracle.com/cd/A87860_01/doc/server.817/a76965/c02block.htm