



Univerzitet u Nišu  
Elektronski fakultet  
Katedra za računarstvo



SISTEMI ZA  
UPRAVLJANJE  
BAZAMA PODATAKA

*Oracle DBMS*  
*skladištenje*  
*podataka*

Profesor:

Prof.dr Aleksandar Stanimirović

Student:

Bojana Svilenković 1417

Niš, 2022.

# Sadržaj

1.	Uvod .....	3
2.	Skladište podataka .....	3
3.	Razlike između OLAP i OLTP sistema .....	5
4.	Prostori tabela u Oracle DBMS.....	6
5.	Tipovi prostora tabela u Oracle DBMS-u .....	10
6.	Tabelarni prostori na mreži i van mreže .....	11
7.	Read-Only tabelarni prostori.....	12
8.	Promene sa Oracle Database 10g .....	12
	8.1. Eksterne tabele .....	12
	8.2. Grupa privremenih tabela .....	16
	8.3. Privremeni poništeni segmenti(Temporary Undo Segments) ....	17
9.	Prostor segmenta i High Water Mark.....	18
10.	Kreiranje prostora tabela .....	21
11.	Literatura .....	23

## 1. Uvod

U poslovnim i naucnim sverama se često dešava da je potrebno prikazati izveštaje koji sadrže relativno malo podataka ali su dobijeni obradom tj. analizom velike kolicine podataka koji su smešteni u bazama razlicitih proizvođača, na različitim serverima ili fizički na više kontinenata. Bez obzira što je možda i moguće generisati takve upite oni:

1. mogu da traju jako dugo tj. da imaju slab odziv
2. mogu da imaju slabu raspoloživost (bilo zbog hardverskih ili softverskih mogućnosti nekog od sistema u nizu, bilo zbog bilo kojih drugih tehničkih problema)
3. mogu drastično da ugroze OLTP sisteme koji pre svega moraju da odgovore potrebi za obradom odgovarajućeg broja transakcija u sekundi.<sup>1</sup>

Stoga je nastala potreba da se podaci koji služe za dobijanje izveštaja odvoje u softverski posebne sisteme, od onih koji obezbeđuju pretežno unos i ažuriranje ili pak stogo definisane izveštaje po unapred predviđenim i definisanim ključevima (indeksima) koji mogu relativno brzo da se dobiju i iz OLTP sistema.

## 2. Skladište podataka

Data Warehouse odnosno skladište podataka je kao koncept nastao kasnih 1980-ih od strane Beri Devlina i Paul Marfija kada je poceo ekperimentalno da se primenjuje u tehnološkim marketima, medutim ocem skladišta podataka se smatra Viliam Bil Inmon čiji je priručnik 1991.godine omogućio praktičnu upotrebu skladišta podataka. Skladišta podataka služe pretežno za analizu podataka koji se više ne menjaju, npr. analiza prodaje u nekom periodu, ali ne na današnji dan vec recimo u zadnjih dvadeset dana po danima ili pak bilo koji drugi period od interesa. Npr. imamo podatke koji su uskladišteni u nekim bazama koje se nazivaju RDBMS (standardne relacione baze tj. jednodimenzione baze kao što su SQL Server, Oracle, MySQL) ili bilo koje druge koje uvek mogu da daju tačno i trenutno stanje. Ti podaci se skupljaju i transformišu (konvertuju) u skladišta. Ta skladišta predstavljaju neku vrstu prečišćenih podataka prilagodjenih izvršenju što većeg broja različitih izveštaja. Skladišta podataka ne služe za onlajn transakcije u smislu ažuriranja(izmena), unosa novih slogova ili brisanja postojećih, dakle ne služe npr. za evidenitiranje promena na kasama supermarketu, raznorazne onlajn prodaje i knjigovodstvene sisteme ili bilo gde drugde gde se pretežno vrši unos ili ažuriranje novih podataka. Skladišta podataka se „pune“ podacima iz OLTP sistema ili drugih skladišta automatizovanim procedurama koje obezbeđuju skupljanje podataka, njihovu analizu i organizovanje cesto u potpuno drugačijem hijearshiskom modelu od onog originalnog izvora podataka. Sistemi koji obezbeđuju izveštaje koji su rezultat sinteze velikog broja podataka i

---

<sup>1</sup> OLTP (online transaction processing) sistem, sistem za onlajn obradu transakcija (Prim.prev.)

obično su podrška nekom strateškom odlucivanju, nazivaju OLAP<sup>2</sup> sistemi kojima su izvor podataka upravo skladišta podataka. Dakle, RDBMS omogućavaju OLTP sistemi, a skladišta podataka omogućavaju OLAP sistemi.

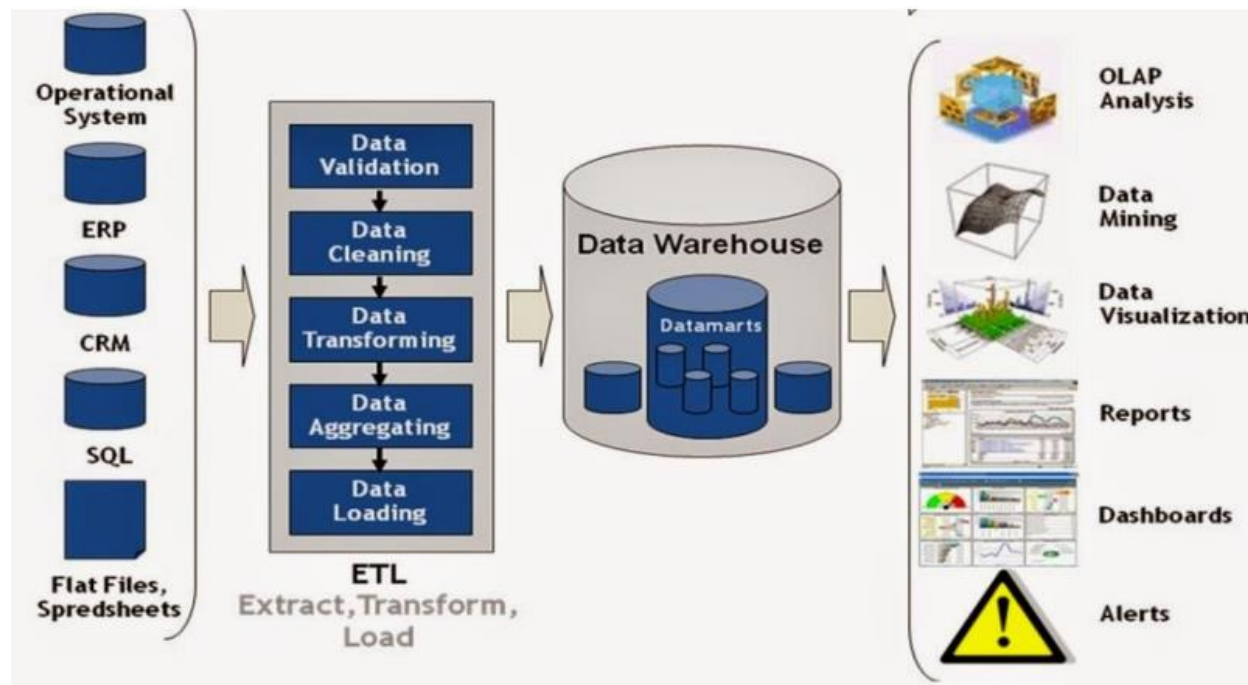
U skladište podataka podaci dolaze iz unutrašnjih i spoljnih izvora.

Unutrašnji podaci opisuju aktivnosti koje se odvijaju unutar preduzeća. Oni nastaju u sistemima za obradu transakcija kao što su proizvodnja, finansije, računovodstvo, prodaja i marketing...

Spoljni podaci se odnose na aktivnosti koje se dešavaju izvan preduzeća. Na primer, podaci o proizvodima i uslugama konkurentnih preduzeća, cene sirovina, tehnološki trendovi, marketinški trendovi ...

Skladišta podataka čuvaju podatke iz više različitih baza o dnevnim, sedmičnim, mesečnim, kvartalnim i godišnjim aktivnostima u datom preduzeću, kao i aktivnostima izvan preduzeća koja su od interesa za preduzeće. Takvi podaci omogućavaju razmatranje poslovnih procesa u celini, na primer, proces prodaje sa stanovišta celog preduzeća, uključujući promociju, nabavku, proizvodnju prodaju i naplatu.

Pre nego što se podaci smeste u skladište podataka vrši se njihovo pročišćavanje i standardizacija kroz procese ekstrakcije, transformacije i punjenja.



<sup>2</sup> OLAP (online analytical processing) sistem, sistem za olajn analiticku obradu (Prim. prev.)

Ekstrakcija omogućuje izbor relevantnih podataka iz baze kojima se puni skladište. Transformacija obuhvata preformatiranje, usklađivanje i čišćenje podataka iz transakcione baze. Preformatiranjem se podaci koji se nalaze u različitim formatima svode na isti format. Usklađivanje se odnosi na uklanjanje podataka koji se ponavljaju i na ujednačavanje nedoslednih podataka. Čišćenjem se uklanjaju tzv. prljavi podaci koji su nastali kao posledica ranijih grešaka u radu sistema ili su namerno ubačeni, npr. razni probni podaci, pogrešno uneti podaci itd. Punjenje skladišta sastoji se od inicijalnog i inkrementalnog punjenja. Po završetku inicijalnog punjenja i punjenja istorijskim podacima, skladište se dalje periodično nadopunjava novim relevantnim podacima.

### 3. Razlike između OLAP i OLTP sistema

Osnovne razlike između OLAP i OLTP sistema su sledeće:

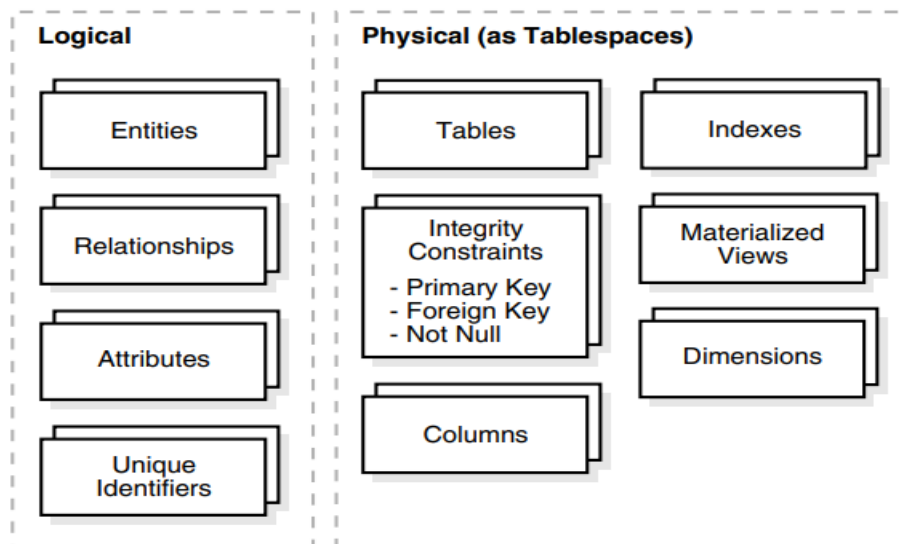
- **Korisnička i sistemska orijentacija:** OLTP sistemi su većinom namenjeni korisnicima koji ažuriraju tj. pretežno unose informacije (službenici, kupci i sl.), a OLAP sistemi korisnicima koji analiziraju informacije (menadžeri, upravljačko osoblje kompanija, vojni i raznorazni drugi analitičari, naučni radnici i sl.)
- **Sadržaj podataka:** OLTP sistemi obezbeđuju previše detaljne informacije i preveliku količinu informacija da bi bili pogodni za složenije upite koji predstavljaju veći nivo izdvojenosti iz šireg skupa podataka i pre svega treba da obezbede potpunost informacija, koje su najčešće smeštene u strogo normalizovanim bazama. OLAP sistemi se oslanjaju na već prečišćene informacije koje se kao takve ne skupljaju u svom izvornom obliku jer ne bi obezbeđivale osnovne zahteve (npr. broj stana klijenta će retko biti od interesa OLAP sistemu, ali hoće OLTP sistemu jer se inače neće znati npr. gde da se pošalje neki proizvod (ako se radi o nekoj onlajn kupovini, ili nečem sličnom)). OLAP sistemu može biti značajno o kojem delu grada, države ili kom kontinentu se radi, što nekom pojedinačnom OLTP sistemu koji je i dizajniran za neko usko područje ne mora biti od interesa, ali OLAP sistemi koji mogu da obuhvate dvadesetak takvih „lokalnih“ baza moraju da sadrže i dodatne informacije koje se obezbeđuju specifičnim sistemima smeštanja podataka u skladištima.
- **Dizajn baza:** OLTP sistemi se obično oslanjaju na ER model podataka (hijerarhijski model sa vezama 1:1, 1:N ili N:N) i aplikativno orijentisan dizajn baze, dok se OLAP sistemi oslanjaju na zvezda (star) ili pahulja (snowflake) model baza koje su korisnički orijentisane (podaci su grupisani tako da obezbeđuju najlakše dobijanje informacije koja je u centru interesovanja).
- **Pogledi (View):** OLTP sistemi su orijentisani pretežno na tekuće podatke, bez osvrta na istorijske podatke ili podatke iz drugih organizacionih celina, OLAP sistemima je bitan širi uvid u podatke radi analiza. OLAP podaci su često smešteni na različitim medijima, serverima i sl.

– **Način pristupa:** OLTP podržava kratke transakcije koje pre svega obezbeđuju tačnost (ne dupliranje) podataka kao i modele čuvanja (backup and recovery) podataka. OLAP podržava baze koje se samo čitaju i u koje se ništa ne upisuje.

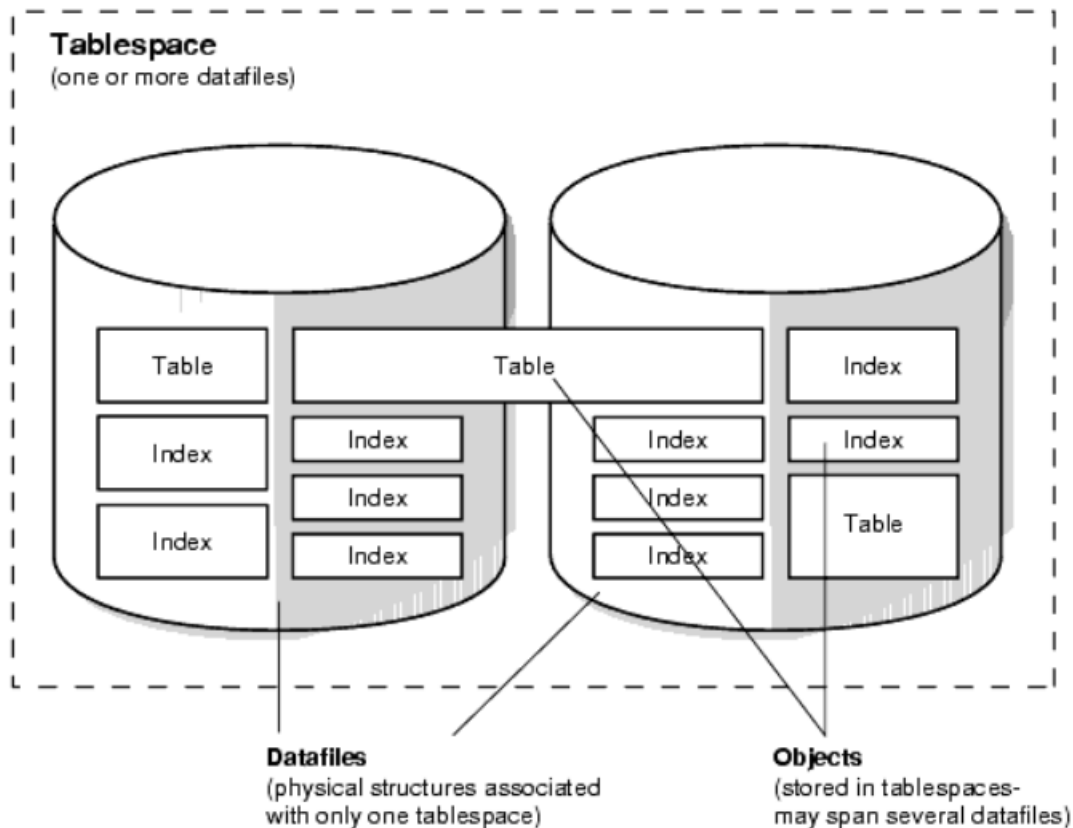
Odlika	OLAP	OLTP
Karakteristika	informativna obrada	operativna obrada
Fokus	analiza, isporuka informacija	transakcija, unos podataka
Korisnici	menadžer, analitičar	prodavac, tehničar
Funkcije	dugoročna analiza pomoć odlučivanju	svakodnevne operacije (unos,izmena,...)
Dizajn baze	zvezda/pahulja	ER model
Podaci	“istorijski”	trenutni
Pristup Broj korisnika	uglavnom čitanje stotine	čitanje/pisanje hiljade
Veličina baze	100 GB to TB	100MB to GB
Prioritet	Fleksibilnost, autonomija krajnjeg korisnika	visoke performanse i dostupnost

#### 4. Prostori tabela u Oracle DBMS

Oracle skladišti podatke logički u prostorima tabela (tablespaces) i fizički u datotekama podataka povezanim sa odgovarajućim prostorom tabela.



Prostor tabele (tablespace) se sastoji od jedne ili više datoteka podataka, koje su fizičke strukture unutar operativnog sistema koji koristite. Datoteka podataka je povezana samo sa jednim prostorom tabele. Iz perspektive dizajna, tabelarni prostori su kontejneri za fizičke strukture dizajna.



Baze podataka (databases), prostori tabele (tablespaces) i datoteke podataka (datafiles) su usko povezani, ali imaju važne razlike:

-Baze podataka i prostori tabela:

Oracle baza podataka se sastoji od jedne ili više logičkih jedinica za skladištenje zvanih prostori tabele, koji zajedno čuvaju sve podatke baze podataka.

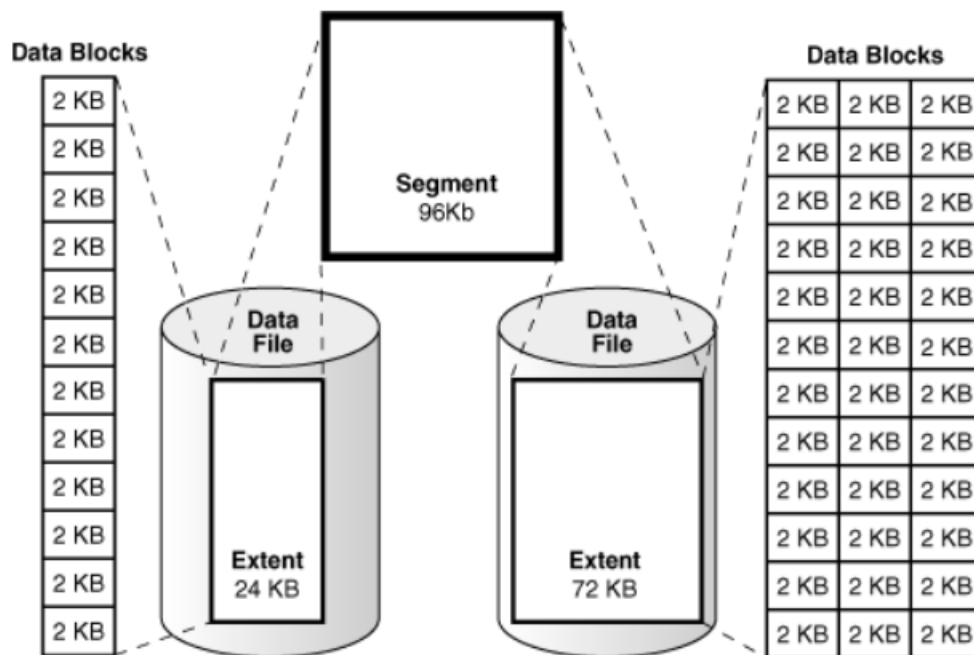
-Prostori tabele i datoteke podataka:

Svaki prostor tabele u Oracle bazi podataka sastoji se od jedne ili više datoteka koje se nazivaju datoteke podataka, koje su fizičke strukture koje su u skladu sa operativnim sistemom u kojem radi Oracle.

-Baze podataka i datoteke podataka:

Podaci baze podataka se zajedno čuvaju u datotekama podataka koje čine svaki prostor tabele u bazi podataka.

Jedinice alokacije prostora Oracle baze podataka su blokovi podataka, ekstenzi i segmenti. Oracle skladišti podatke u blokovima podataka i skup blokova podataka se naziva ekstenzi. Segment je skup ekstenzi i svaka tabela i indeks su segment u Oracle bazi podataka.



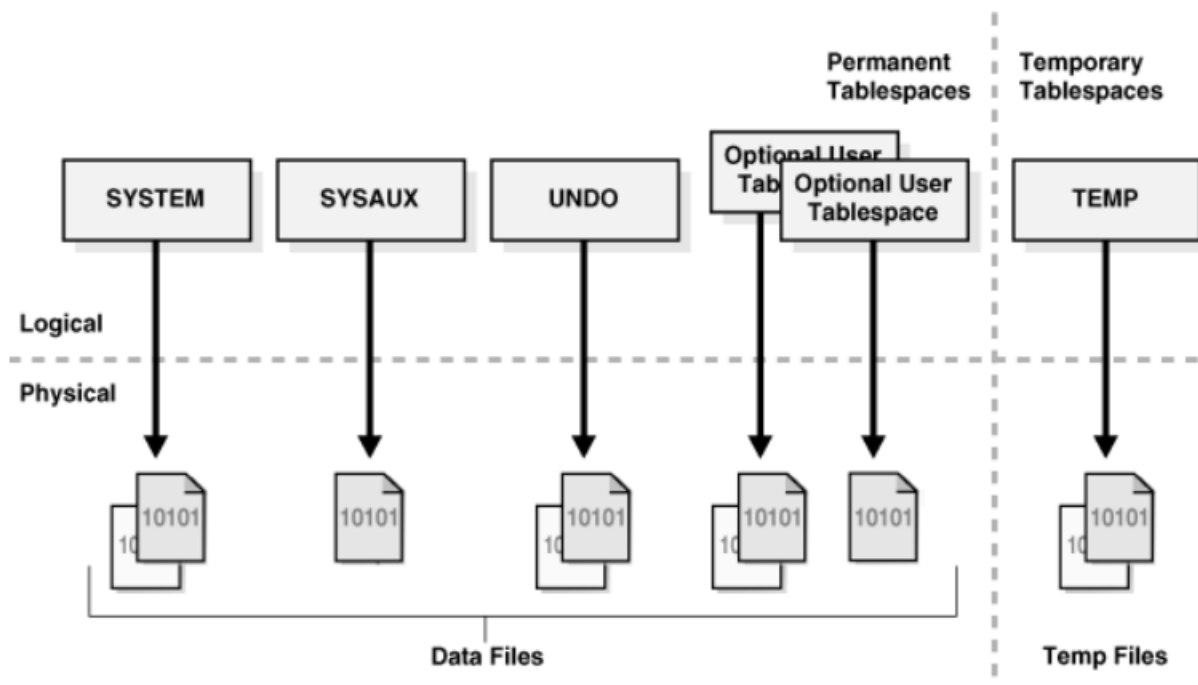
Kada instaliramo i kreiramo novu Oracle bazu podataka, ona ima 5 podrazumevanih prostora tabela:

- **SYSTEM** – Ovaj tabelarni prostor se automatski kreira prilikom kreiranja baze podataka. Oracle Database ga koristi za upravljanje bazom podataka. Sadrži rečnik podataka, koji je centralni skup tabela i pogleda koji se koristi kao referenca samo za čitanje za određenu bazu podataka. Takođe sadrži različite tabele i prikaze koji sadrže administrativne informacije o bazi podataka. Svi oni su sadržani u SYS šemi i mogu im pristupiti samo SYS korisnik ili drugi administrativni korisnici sa potrebnim privilegijama.
- **SYSAUX** – Ovo je pomoćni tabelarni prostor za SISTEM tabelarni prostor. Sadrži podatke za neke komponente i proizvode, smanjujući opterećenje **SYSTEM** tabelarnog prostora. Svaka baza podataka koja koristi Oracle Database 10g izdanje 1 (10.1) ili noviju mora imati SYSAUXS tabelarni prostor. Komponente



koje koriste SYSAUX kao podrazumevani tabelarni prostor tokom instalacije uključuju Automatic Workload Repository, Oracle Streams i Oracle Text.

- **TEMP** – Ovaj tabelarni prostor čuva privremene podatke generisane prilikom obrade SQL naredbi. Na primer, ovaj prostor tabele bi se koristio za sortiranje upita. Svaka baza podataka treba da ima privremeni tabelarni prostor koji se dodeljuje korisnicima kao njihov privremeni tabelarni prostor. U unapred konfigurisanoj bazi podataka, TEMP tabelarni prostor je naveden kao podrazumevani privremeni prostor tabele. Ako privremeni prostor tabele nije naveden kada se kreira korisnički nalog, onda Oracle Database dodeljuje ovaj tabelarni prostor korisniku.
- **UNDOTBS1** – Ovo je prostor tabele za poništavanje koji koristi baza podataka za skladištenje informacija o opozivu. Svaka baza podataka mora imati prostor za poništavanje
- **USERS** – Ovaj tabelarni prostor se koristi za skladištenje stalnih korisničkih objekata i podataka. Slično TEMP prostoru tabele, svaka baza podataka treba da ima tabelarni prostor za trajne korisničke podatke koji se dodeljuje korisnicima. U suprotnom, korisnički objekti će biti kreirani u SYSTEM prostoru tabele, što nije dobra praksa. U unapred konfigurisanoj bazi podataka, USERS je određen kao podrazumevani prostor tabele za sve nove korisnike

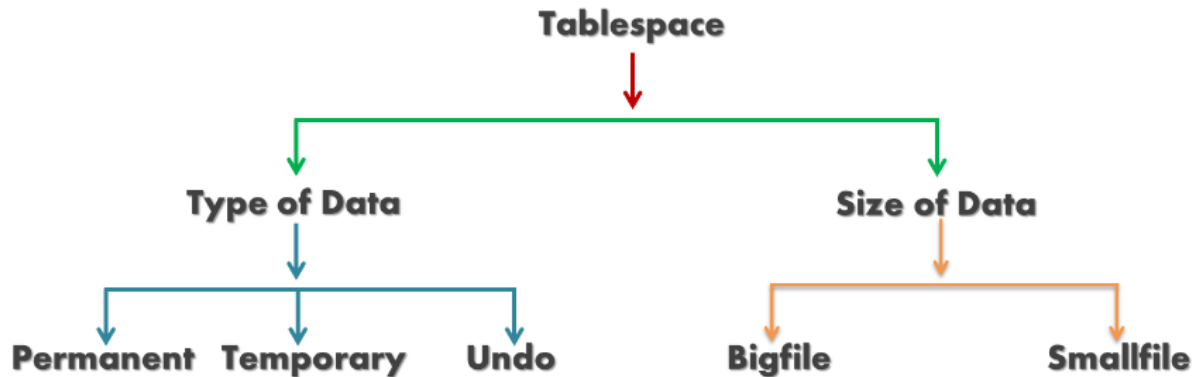


## 5. Tipovi prostora tabela u Oracle DBMS-u

Možemo razlikovati tabelarne prostore na osnovu dva faktora:

1. Tip podataka
  2. Veličina podataka
- 
1. Tip podataka se sastoji od 3 vrste prostora tabele uključujući:
    - **Permanent Tablespace** –Koristite trajne tabelarne prostore za skladištenje podataka korisnika i aplikacija. Oracle Database koristi trajne tabelarne prostore za skladištenje trajnih podataka, kao što su sistemski podaci. Svakom korisniku je dodeljen podrazumevani trajni prostor tabele.
    - **Temporary Tablespace** – Privremeni prostori tabela se koriste za skladištenje privremenih podataka, kao što bi bili kreirani kada SQL naredbe obavljaju operacije sortiranja. Oracle baza podataka dobija privremeni prostor tabele kada se baza podataka kreira.
    - **Undo Tablespace** – Baza podataka koja radi u automatskom režimu upravljanja poništavanjem transparentno kreira i upravlja podacima za poništavanje u prostoru tabele za poništavanje. Oracle Database koristi podatke za poništavanje za vraćanje transakcija, da obezbedi doslednost čitanja, da pomogne u oporavku baze podataka.
  2. Na osnovu veličine podataka imamo 2 vrste prostora tabele:
    - **Veliki tabelarni prostor datoteka (Big file tablespace)** – Novi koncept je počeo sa Oracle 10g. Veliki tabelarni prostor datoteka je najpogodniji za skladištenje velikih količina podataka. Tabelarni prostor velikog fajla može imati najviše 1 datoteku podataka, što znači da su tabelarni prostori velikih datoteka izgrađeni na pojedinačnim datotekama podataka koje mogu biti veličine do 232 bloka podataka. Dakle, veliki tabelarni prostor koji koristi blokove podataka od 8 KB može biti veličine čak 32 TB.
    - **Mali tabelarni prostor datoteka ( Small file tablespace)** – Ovo je podrazumevani tip prostora tabele u Oracle bazi podataka. Mali tabelarni prostor može imati više datoteka sa podacima i svaka datoteka može imati čak 222 bloka podataka.

Mali tabelarni prostor može imati najviše 1022 datoteke sa podacima, ali ovaj broj zavisi i od našeg operativnog sistema.



## 6. Tabelarni prostori na mreži i van mreže

Administrator baze podataka može dovesti bilo koji prostor tabele osim SYSTEM tabelarnog prostora onlajn (dostupan) ili van mreže (nedostupan) kad god je baza podataka otvorena. Tabelarni prostor SYSTEM je uvek na mreži kada je baza podataka otvorena jer rečnik podataka mora uvek biti dostupan Oracle-u.

Tabelarni prostor je obično onlajn tako da su podaci sadržani u njemu dostupni korisnicima baze podataka. Međutim, administrator baze podataka može da stavi prostor tabele van mreže:

- Da deo baze podataka bude nedostupan, a da se omogući normalan pristup ostatku baze podataka
- Da bismo napravili rezervnu kopiju prostora tabele van mreže (iako se rezervna kopija prostora tabele može napraviti dok je onlajn i u upotrebi)
- Da bi aplikacija i njena grupa tabela bili privremeno nedostupni tokom ažuriranja ili održavanja aplikacije

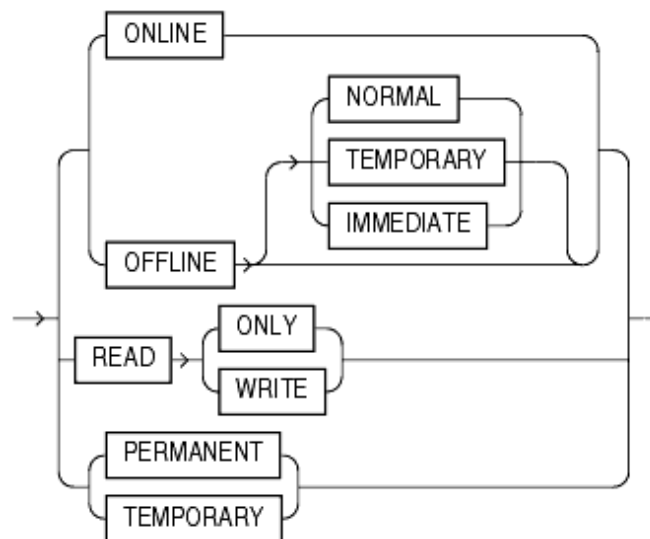
Ne možemo da prebacimo tabelarni prostor van mreže ako sadrži bilo koje segmente za vraćanje koji su u upotrebi.

## 7. Read-Only tabelarni prostori

Primarna svrha tabela samo za čitanje(read-only) je da eliminiše potrebu za pravljenjem rezervnih kopija i oporavka velikih, statičnih delova baze podataka. Oracle nikada ne ažurira datoteke tabelarnog prostora samo za čitanje i stoga datoteke mogu da se nalaze na medijima samo za čitanje kao što su CD-ROM-ovi ili WORM disk jedinice.

Tabelarni prostori samo za čitanje se ne mogu menjati. Da biste ažurirali tabelarni prostor samo za čitanje, prvo učinite da tabelni prostor čita/piše(read-write). Nakon ažuriranja prostora tabelle, možete ga resetovati da bude samo za čitanje.

Zato što se tabelarni prostori samo za čitanje ne mogu modifikovati i sve dok nisu napravljeni za čitanje/upisivanje ni u jednom trenutku, ne treba im ponovljeno pravljenje rezervnih kopija. Takođe, ako treba da oporavite svoju bazu podataka, ne morate da oporavljate bilo koji tabelarni prostor samo za čitanje, jer oni nisu mogli biti izmenjeni.



## 8. Promene sa Oracle Database 10g

### 8.1. Eksterne tabelle

Funkcija eksternih tabela je dopuna postojećoj funkcionalnosti SQL\*Loader-a. Omogućava nam da pristupimo podacima u eksternim izvorima kao da su u tabeli u bazi podataka.

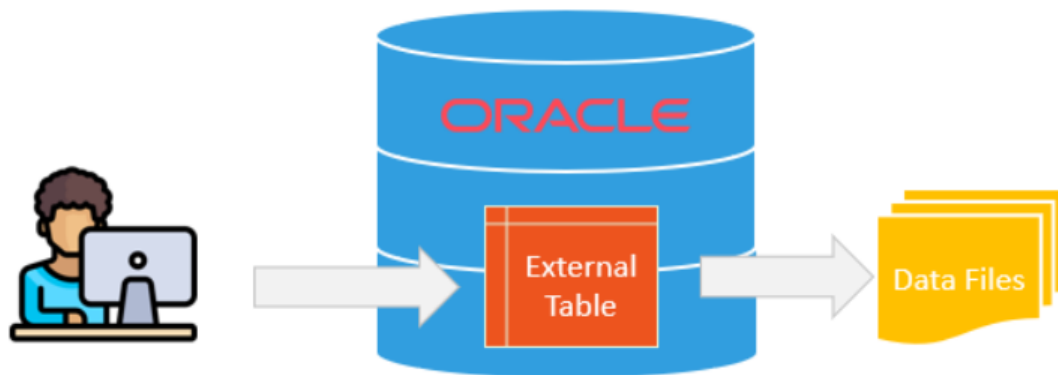
Pre Oracle Database 10g, eksterne tabele su bile samo za čitanje. Međutim, od Oracle Database 10g, u eksterne tabele se takođe može pisati. SQL\*Loader može biti bolji izbor u situacijama učitavanja podataka koje zahtevaju dodatno indeksiranje tabele za provođenje.

Da bismo koristili funkciju eksternih tabela, moramo imati izvesno znanje o formatu datoteke i formatu zapisa datoteka sa podacima na našoj platformi ako se koristi drajver za pristup ORACLE\_LOADER i datoteke podataka su u tekstualnom formatu.

Ne bi trebalo da koristimo eksterne tabele za tabele kojima se često pristupa.

### Koraci kreiranja Oracle eksterne tabele

- Prvo, napravimo direktorijum koji sadrži datoteku kojoj će Oracle pristupiti pomoću naredbe CREATE DIRECTORY.
- Drugo, odobrimo pristup READ i WRITE korisnicima koji pristupaju eksternoj tabeli koristeći naredbu GRANT.
- Treće, kreiramo eksternu tabelu korišćenjem naredbe CREATE TABLE ORGANIZATION EXTERNAL.



Primer:

Evo CSV datoteke koja ima dve kolone: ID jezika i naziv.



1	1,Acholi
2	2,Afrikaans
3	3,Akan
4	4,Albanian
5	5,Amharic
6	6,Arabic
7	7,Ashante
8	8,Asl
9	9,Assyrian
10	10,Azerbaijani
11	11,Azeri
12	12,Bajuni

```
SQL> grant read,write on directory lang_external to ot;
```

Grant succeeded.

```
CREATE TABLE languages(  
    language_id INT,  
    language_name VARCHAR2(30)  
)  
ORGANIZATION EXTERNAL(  
    TYPE oracle_loader  
    DEFAULT DIRECTORY lang_external  
    ACCESS PARAMETERS  
    (FIELDS TERMINATED BY ',')  
    LOCATION ('languages.csv')  
);
```

- **TYPE**- određuje tip eksterne tabele. Oracle pruža dva tipa: ORACLE\_LOADER(iz datoteka tekstualnih podataka) i ORACLE\_DATADUMP(u formatu datoteke binarnog dump-a)
- **DEFAULT DIRECTORY** - omogućava da odredimo podrazumevani direktorijum za skladištenje svih ulaznih i izlaznih datoteka
- **ACCESS PARAMETERS**- omogućava da opišemo eksterni izvor podataka
- **LOCATION**- navodi datoteke podataka za spoljnu tabelu u formatu direktorijum:fajl. Ako izostavimo deo direktorijuma, Oracle koristi DEFAULT DIRECTORY za datoteku.

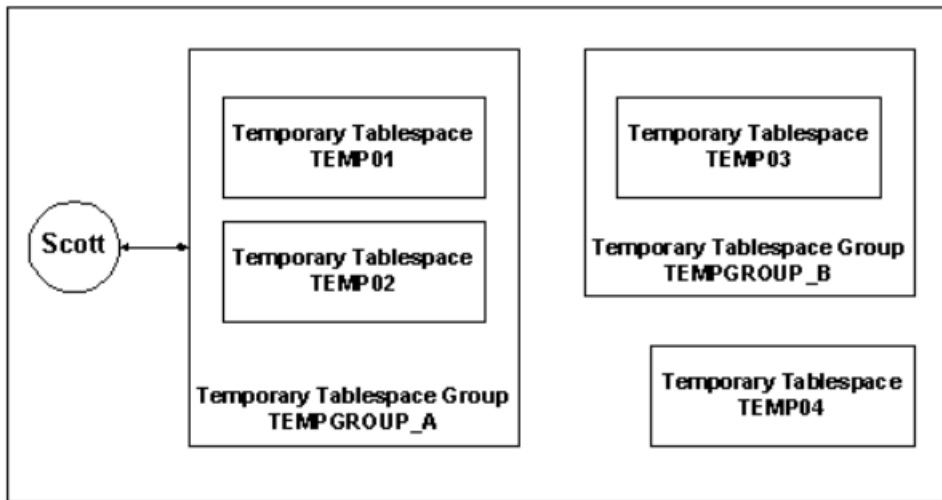
Kada se kreira spoljna tabela, možemo da je koristimo kao normalnu tabelu:

```
SELECT
    language_id,
    language_name
FROM
    languages
ORDER BY
    language_name;
```

Izlaz:

LANGUAGE_ID	LANGUAGE_NAME
1	Acholi
2	Afrikaans
3	Akan
4	Albanian
5	Amharic
6	Arabic
7	Ashante
8	Asl
9	Assyrian
10	Azerbaijani
11	Azeri
12	Bajuni

## 8.2. Grupa privremenih tabela



Grupa privremenih tabela(Temporary Tablespace Group) ima sledeće prednosti:

- Omogućava da se navede više podrazumevanih privremenih prostora tabelle na nivou baze podataka.
- Omogućava korisniku da koristi više privremenih tabela u različitim sesijama u isto vreme.
- Omogućava da jedna SQL operacija koristi više privremenih tabela za sortiranje.

Preimenovanje tabelarnog prostora unutar baze podataka.

U Oracle9i ili ranijim izdanjima, moramo preduzeti sledeće korake da bismo preimenovali prostor tabelle iz OLD\_TBS u NEW\_TBS:

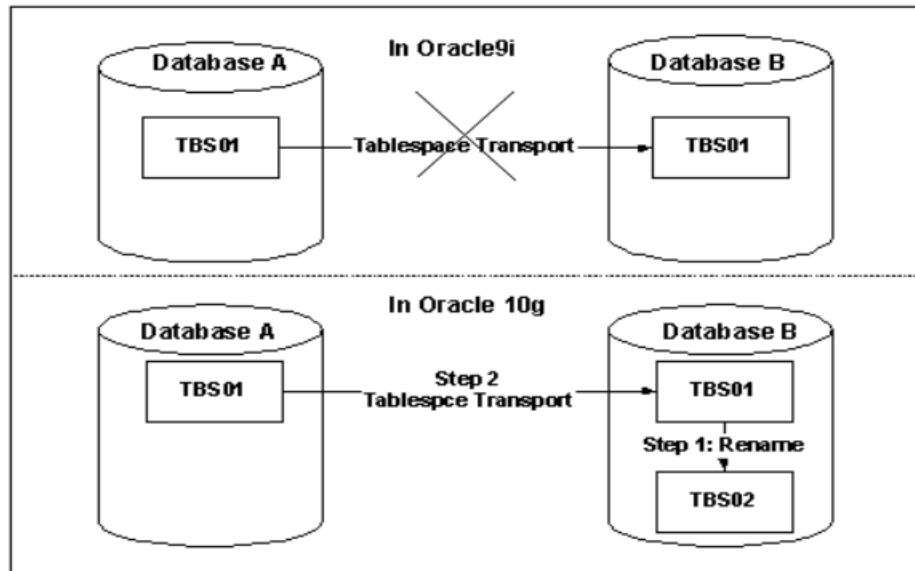
1. Kreiranje novog tabelarnog prostora NEW\_TBS.
2. Kopiranje svih objekata iz OLD\_TBS u NEW\_TBS.
3. Brisanje tabelarnog prostora OLD\_TBS.

U Oracle 10g možemo da postignemo istu stvar u jednom koraku:

```
Rename tablespace OLD_TBS to NEW_TBS.  
ALTER TABLESPACE RENAME OLD_TBS NEW_TBS;
```



Transport tabelarnog prostor između dve baze podataka. U sledećem primeru, ne možemo da transportujemo tabelarni prostor TBS01 iz baze podataka A u bazu podataka B u prethodnom izdanju Oracle servera jer baza podataka B takođe ima tabelarni prostor koji se zove TBS01. U Oracle 10g, možemo jednostavno preimenovati TBS01 u TBS02 u bazi podataka B pre transporta prostora table TBS01.



### 8.3. Privremeni poništeni segmenti(Temporary Undo Segments)

Privremeni segment za poništavanje je opcioni kontejner za upravljanje prostorom samo za podatke o privremenim opozivima.

Zapisi o opozivanju promena u privremenim tabelama su i specifični za sesiju i korisni samo za doslednost čitanja i vraćanje transakcije. Pre Oracle Database 12c, baza podataka je uvek čuvala ove zapise u onlajn dnevniku ponavljanja. Pošto se promene na privremenim objektima ne evidentiraju u onlajn evidenciji ponavljanja, pisanje opoziva za privremene objekte u privremene segmente poništavanja štedi prostor u onlajn evidenciji ponavljanja i arhiviranim datotekama dnevnika ponavljanja. Baza podataka ne beleži promene za opoziv ili promene u privremenu tabelu, što poboljšava performanse.

Možete podesiti parametar inicijalizacije TEMP\_UNDO\_ENABLED tako da privremene tabelle čuvaju podatke o opozivu u segmentu privremenog opoziva. Kada je ovaj parametar TRUE, baza podataka dodeljuje privremene segmente poništavanja iz privremenih tabela.

## 9. Prostor segmenta i High Water Mark

Da bi upravljao prostorom, Oracle Database prati stanje blokova u segmentu. High Water Mark (HWM) je tačka u segmentu iza koje su blokovi podataka neformatirani i nikada nisu korišćeni.

MSSM<sup>3</sup> koristi slobodne liste za upravljanje segmentnim prostorom. Prilikom kreiranja tabele, nijedan blok u segmentu se ne formatira. Kada sesija prvi put ubaci redove u tabelu, baza podataka pretražuje slobodnu listu za upotrebljive blokove. Ako baza podataka ne pronađe upotrebljive blokove, onda unapred formatira grupu blokova, stavlja ih na listu slobodnih i počinje da ubacuje podatke u blokove. U MSSM-u, skeniranje cele tabele čita sve blokove ispod HWM-a.

ASSM<sup>4</sup> ne koristi slobodne liste i zato mora drugačije da upravlja prostorom. Kada sesija prvi put ubaci podatke u tabelu, baza podataka formatira jedan bitmap blok umesto prethodnog formatiranja grupe blokova kao u MSSM-u. Bitmapa prati stanje blokova u segmentu, zauzimajući mesto liste slobodnih. Baza podataka koristi bitmapu da pronađe slobodne blokove, a zatim formatira svaki blok pre nego što ga napuni podacima. ASSM raspoređuje umetke između blokova da bi se izbegli problemi istovremenosti.

Svaki blok podataka u ASSM segmentu je u jednom od sledećih stanja:

1. Iznad HWM-a

Ovi blokovi su neformatirani i nikada nisu korišćeni.

2. Ispod HWM-a

Ovi blokovi su u jednom od sledećih stanja:

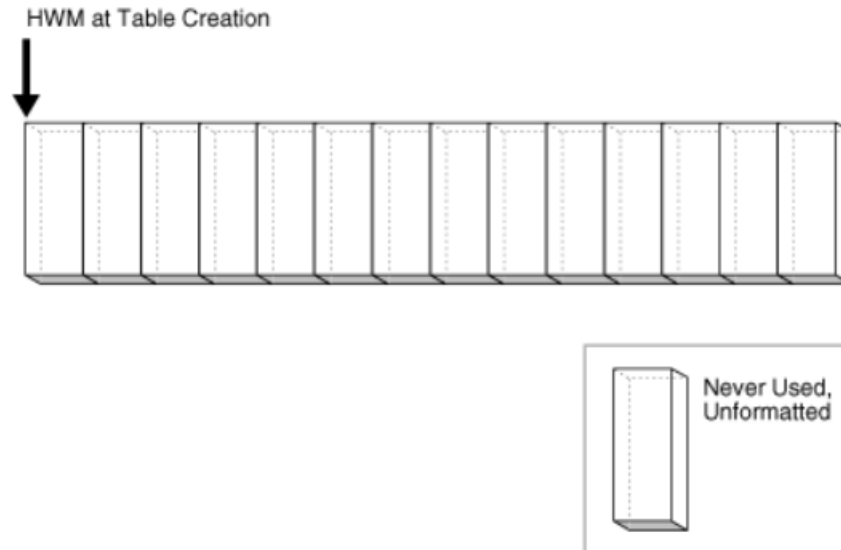
- Dodeljeno, ali trenutno nije formatirano i neiskorišćeno
- Formatirano i sadrži podatke
- Formatirano i prazno jer su podaci izbrisani

---

<sup>3</sup> Manual Segment Space Management- metoda koristi slobodnu listu za upravljanje slobodnim prostorom u segmentu.

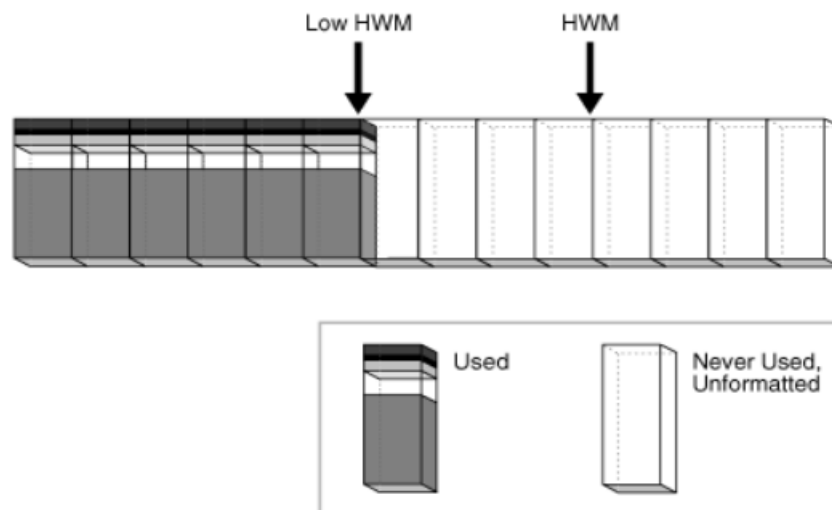
<sup>4</sup> Automatic Segment Space Management-metoda koristi bitmape za upravljanje prostorom u prostoru tabele

Slika prikazuje ASSM segment kao horizontalni niz blokova. Prilikom kreiranja tabele, HWM je na početku segmenta sa leve strane. Pošto nijedan podatak još nije umetnut, svi blokovi u segmentu su neformatirani i nikada se ne koriste.

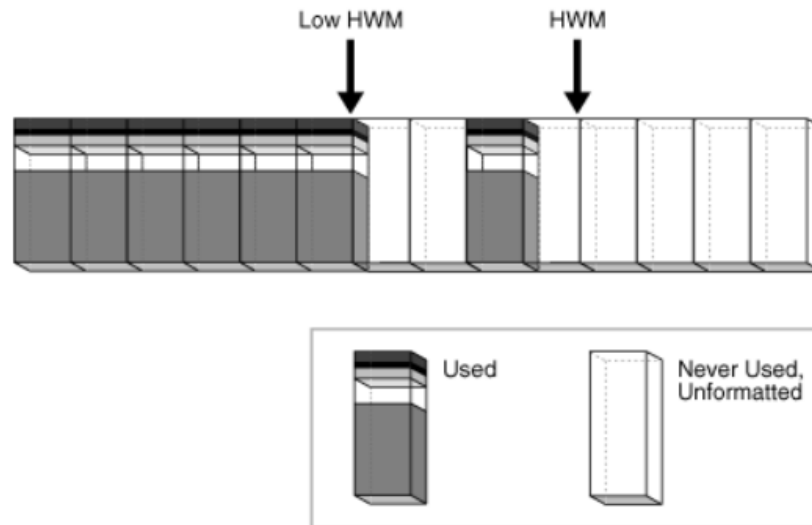


Pretpostavimo da transakcija umeće redove u segment. Baza podataka mora da dodeli grupu blokova za držanje redova. Dodeljeni blokovi padaju ispod HWM-a. Baza podataka formatira blok bitmape u ovoj grupi da zadrži metapodatke, ali ne formatira preostale blokove u grupi.

Na sledećoj slici, blokovi ispod HWM-a su dodeljeni, dok blokovi iznad HWM-a nisu ni dodeljeni niti formatirani. Kako dođe do umetanja, baza podataka može da piše u bilo koji blok sa raspoloživim prostorom. Low HWM (niski HWM) označava tačku ispod koje se zna da su svi blokovi formatirani jer ili sadrže podatke ili su ranije sadržavali podatke.

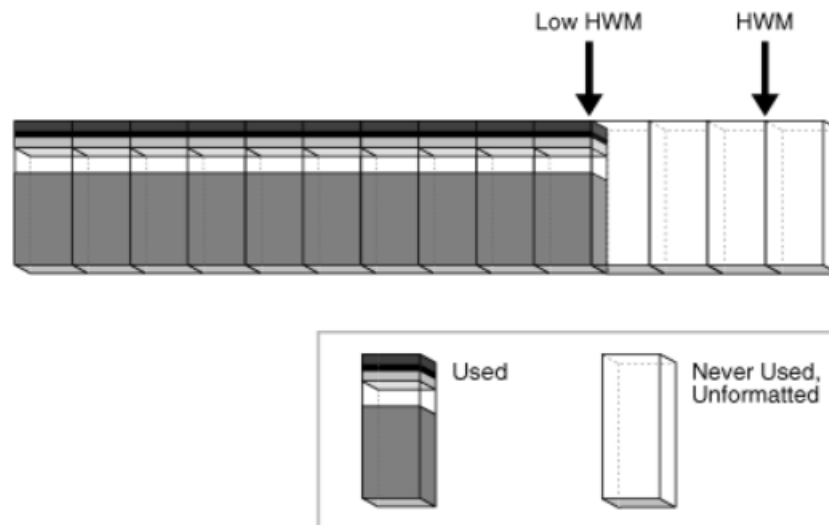


Na sledećoj slici, baza podataka bira blok između HWM-a i niskog HWM-a i upisuje u njega. Baza podataka je mogla isto tako lako izabrati bilo koji drugi blok između HWM-a i niskog HWM-a, ili bilo koji blok ispod niskog HWM-a koji je imao raspoloživ prostor. Blokovi sa obe strane novo popunjenog bloka nisu formatirani.



Nizak HWM je važan u celom skeniranju tabele. Pošto se blokovi ispod HWM-a formatiraju samo kada se koriste, neki blokovi mogu biti neformatirani, kao na slici. Iz tog razloga, baza podataka čita blok bitmape da bi dobila lokaciju niskog HWM-a. Baza podataka čita sve blokove do niskog HWM-a jer je poznato da su formatirani, a zatim pažljivo čita samo formatirane blokove između niskog HWM-a i HWM-a.

Pretpostavimo da nova transakcija umeće redove u tabelu, ali bitmapa ukazuje da nema dovoljno slobodnog prostora ispod HVM-a. Na sledećoj slici, baza podataka pomera HWM udesno, dodeljujući novu grupu neformatiranih blokova.



Kada su blokovi između HWM i niske HWM puni, HWM napreduje udesno, a niski HWM napreduje do lokacije starog HWM. Kako baza podataka ubacuje podatke tokom vremena, HWM nastavlja da napreduje udesno, a niski HWM uvek zaostaje za njim. Osim ako ručno ne izgradimo, skratimo ili smanjimo objekat, HWM se nikada ne povlači.

## 10. Kreiranje prostora tabela

Možemo kreirati novi stalni tabelarni prostor na sledeći način:

- Ako koristimo sistem datoteka za Oracle Storage:

```
CREATE TABLESPACE NEW_TBS_TEST
DATAFILE
'/oracle/oradata/TEST/NEW_TBS_TEST01.dbf' SIZE 4G AUTOEXTEND ON NEXT 200M MAXSIZE UNLIMITED,
'/oracle/oradata/TEST/NEW_TBS_TEST02.dbf' SIZE 4G AUTOEXTEND ON NEXT 200M MAXSIZE UNLIMITED
LOGGING
ONLINE
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
BLOCKSIZE 8K
SEGMENT SPACE MANAGEMENT AUTO
FLASHBACK ON;
```

- Ako koristimo Oracle ASM<sup>5</sup>:

```
CREATE TABLESPACE NEW_TBS_TEST
DATAFILE
'+DATA' SIZE 4G AUTOEXTEND ON NEXT 200M MAXSIZE UNLIMITED
LOGGING
ONLINE
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
BLOCKSIZE 8K
SEGMENT SPACE MANAGEMENT AUTO
FLASHBACK ON;
```

- Ako želimo da kreiramo privremeni (temp) tabelarni prostor, možemo ga kreirati na sledeći način, ako koristimo Oracle ASM:

```
CREATE TEMPORARY TABLESPACE TEMP2 TEMPFILE
'+DATA1' SIZE 100M AUTOEXTEND ON NEXT 1024M MAXSIZE UNLIMITED
TABLESPACE GROUP ''
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M;
```

---

<sup>5</sup> Automatic Storage Management

- Ako koristimo sistem datoteka, možemo kreirati privremeni prostor tabele na sledeći način:

```
CREATE TEMPORARY TABLESPACE TEMP2 TEMPFILE
  '/oracle/oradata/TEST/TEMP02.dbf' SIZE 100M AUTOEXTEND ON NEXT 1024M MAXSIZE UNLIMITED
TABLESPACE GROUP ''
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M;
```

- Ako želimo da kreiramo prostor za poništavanje, možemo ga kreirati na sledeći način, ako koristimo Oracle ASM:

```
CREATE UNDO TABLESPACE UNDO2 DATAFILE
  '+DATA1' SIZE 100M AUTOEXTEND ON NEXT 1024M MAXSIZE UNLIMITED
RETENTION NOGUARANTEE
BLOCKSIZE 8K
FLASHBACK ON;
```

- Ako koristimo sistem datoteka, možemo da kreiramo prostor tabele za poništavanje na sledeći način:

```
CREATE UNDO TABLESPACE UNDO2 DATAFILE
  '/oracle/oradata/TEST/UNDO02.dbf' SIZE 100M AUTOEXTEND ON NEXT 1024M MAXSIZE UNLIMITED
TABLESPACE GROUP ''
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M;
```

- Ako želimo da kreiramo tabelarni prostor velikih datoteka, možemo ga kreirati na sledeći način, ako koristimo Oracle ASM:

```
CREATE BIGFILE TABLESPACE BIGTABLESPACE DATAFILE
  '+DATA1' SIZE 100M AUTOEXTEND ON NEXT 1024M MAXSIZE 34359738344K
LOGGING
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
BLOCKSIZE 8K
SEGMENT SPACE MANAGEMENT AUTO
FLASHBACK ON;
```

- Možemo dodati novu datoteku podataka u tabelarni prostor. Ako se koristi Oracle ASM:

```
ALTER TABLESPACE NEW_TBS_TEST ADD DATAFILE '+DATA' SIZE 15M AUTOEXTEND ON MAXSIZE UNLIMITED;
```

- Možemo dodati novu datoteku podataka u prostor tabele, ako se koristi sistem datoteka:

```
ALTER TABLESPACE NEW_TBS_TEST ADD DATAFILE '/oracle/oradata/TEST/NEW_TBS_TEST03.dbf'
SIZE 15M AUTOEXTEND ON MAXSIZE UNLIMITED;
```

## 11. Literatura

- [1] <https://www.oracletutorial.co/oracle-administration/oracle-tablespace>
- [2] [https://docs.oracle.com/cd/A84870\\_01/doc/server.816/a76965/c03space.htm#3781](https://docs.oracle.com/cd/A84870_01/doc/server.816/a76965/c03space.htm#3781)
- [3] <http://www.rebellionrider.com/tablespace-in-oracle-the-introduction-rebellionrider/>
- [4] [http://www.dba-oracle.com/data\\_warehouse/read\\_only\\_tablespace.htm](http://www.dba-oracle.com/data_warehouse/read_only_tablespace.htm)
- [5] <https://docs.oracle.com/database/121/CNCPT/logical.htm#CNCPT89026>
- [6] [http://www.dba-oracle.com/art\\_dbazine\\_ora10g temptbs.htm](http://www.dba-oracle.com/art_dbazine_ora10g temptbs.htm)