|  |  |  |
| --- | --- | --- |
| **Unigrb** | Univerzitet u Nišu  ELEKTRONSKI FAKULTET | **logo_1960_4** |

**Backup, restore i recover Oracle baze podataka**

SEMINARSKI RAD

|  |  |
| --- | --- |
| Mentor: | Student: |
|  |  |
| Prof.dr Aleksandar Stanimirović | Teodora Sekulić 1644 |

Niš, maj 2024.god.

**Sadržaj**

[1. Uvod 3](#_Toc166669520)

[2. Opcije za bekapovanje i oporavak baza podataka 4](#_Toc166669521)

[2.1 Backup politika 5](#_Toc166669522)

[2.2 Nivoi bekapovanja 5](#_Toc166669523)

[2.3 Bekapovanje na disk 6](#_Toc166669524)

[2.4 Bekapovanje na Eksterni FRA (Network File System) 6](#_Toc166669525)

[2.4.1 Kreiranje Mount Point za lokaciju bekapovanja putem NFS-a 7](#_Toc166669526)

[2.5 Bekapovanje na Oracle Object Storage 8](#_Toc166669527)

[2.5.1 Konfigurisanje postavki Proxy agenta za pristup Object Store-u 9](#_Toc166669528)

[2.6 Izveštaj o backup-u 11](#_Toc166669529)

[2.7 Opcije oporavka 13](#_Toc166669530)

[2.7.1 Point-in-Time Database Recovery (PITR) 13](#_Toc166669531)

[2.7.2 Oporavak na osnovu System Change Number-a 13](#_Toc166669532)

[2.8 Zastareli backup-ovi baze podataka 14](#_Toc166669533)

[3. Kreiranje backup politike za bazu podataka 14](#_Toc166669534)

[4. Alati za upravljanje backup-om i oporavkom baze podataka 15](#_Toc166669535)

[4.1 Recovery Manager – RMAN 16](#_Toc166669536)

[4.1.1 Pokretanje RMAN-a i povezivanje sa bazom podataka 21](#_Toc166669537)

[4.1.2 Pravljenje backup-a baze podataka u ARCHIVELOG režimu 23](#_Toc166669538)

[4.1.3 Pravljenje bakcup-a baze podataka u NOARCHIVELOG režimu 25](#_Toc166669539)

[4.1.4 Pravljenje inkrementalnih backup-ova 27](#_Toc166669540)

[4.1.5 Oporavak cele baze podataka 30](#_Toc166669541)

[4.2 Korisnički upravljani backup-ovi i oporavak 32](#_Toc166669542)

[4.2.1 Pravljenje backup-ova cele baze podataka kojima upravlja korisnik 36](#_Toc166669543)

[4.2.2 Pravljenje korisnički upravljanih backup-ova tablespace-ova i datoteka podataka 39](#_Toc166669544)

[4.2.3 Pravljenje korisnički upravljanih backup-ova kontrolne datoteke 46](#_Toc166669545)

[4.2.4 Pravljenje korisnički upravljanih backup-ova arhiviranih redo logova 48](#_Toc166669546)

[4.2.5 Korisnički upravljan oporavak cele baze podataka 50](#_Toc166669547)

[5. Zaključak 55](#_Toc166669548)

[Literatura 56](#_Toc166669549)

# **Uvod**

Upravljanje bazama podataka je ključni aspekt informacionih sistema, a osiguranje konzistentnosti i integriteta podataka kroz adekvatne strategije backupovanja i oporavka je od suštinskog značaja. Ovaj rad predstavlja sveobuhvatni vodič kroz opcije za bekapovanje i oporavak baza podataka, detaljno objašnjavajući različite metode, alate i prakse koje se mogu koristiti za osiguranje kontinuiteta poslovanja i minimalizaciju gubitka podataka.

U prvom poglavlju biće razmatrane različite opcije za bekapovanje i oporavak baza podataka. Ono počinje definisanjem backup politike koja je osnov za sve aktivnosti bekapovanja. Zatim, biće opisani nivoi bekapovanja koji određuju obim i vrstu podataka koji se čuvaju. U nastavku, biće obrađeno bekapovanje na disk i bekapovanje na eksterni FRA putem Network File System-a[[1]](#footnote-1), gde će biti objašnjeno kako kreirati Mount Point za lokaciju bekapovanja putem NFS-a. Takođe, biće pokriveno bekapovanje na Oracle Object Storage[[2]](#footnote-2) sa detaljnim objašnjenjem kako konfigurisati postavke Proxy agenta za pristup Object Store-u. Izveštaj o backup-u će biti ključan za praćenje i validaciju bekapovanja, dok će opcije oporavka uključivati Point-in-Time Database Recovery (PITR)[[3]](#footnote-3) i oporavak na osnovu System Change Number-a[[4]](#footnote-4). Poglavlje će se završiti razmatranjem zastarelih backup-ova baze podataka.

U narednom poglavlju fokus će biti na kreiranju specifičnih backup politika za baze podataka, osiguravajući prilagodljivost i efikasnost bekapovanja u različitim okruženjima.

U poslednjem poglavlju biće predstavljeni alati za upravljanje backup-om i oporavkom baza podataka. Poseban fokus biće na Recovery Manager (RMAN)[[5]](#footnote-5), gde će biti detaljno objašnjeno pokretanje RMAN-a i povezivanje sa bazom podataka, pravljenje backup-a u ARCHIVELOG i NOARCHIVELOG režimu, pravljenje inkrementalnih backup-ova i oporavak cele baze podataka. Takođe, biće pokriveni korisnički upravljani backup-ovi i oporavak, uključujući pravljenje backup-ova cele baze podataka, tablespace-ova i datoteka podataka, kontrolnih datoteka i arhiviranih redo logova, kao i oporavak cele baze podataka.

Ovaj dokument pruža detaljna uputstva i najbolje prakse za sve aspekte bekapovanja i oporavka baza podataka, omogućavajući korisnicima da osiguraju sigurnost i integritet svojih podataka.

# **Opcije za bekapovanje i oporavak baza podataka**

Bekapovanje baza podataka na Oracle Database Appliance-u je preporučljivo, uz korišćenje opcija za vraćanje i obnavljanje radi povratka baze podataka na isti ili alternativni sistem, ili radi obnove relevantnih fajlova baze podataka.

Važno je kreirati strategiju obnavljanja bekapova kako bi zaštitili baze podataka na Oracle Database Appliance-u od gubitka podataka zbog fizičkog problema sa diskom koji uzrokuje neuspeh čitanja ili pisanja na disk fajlu koji je potreban za pokretanje baze podataka, ili zbog korisničke greške. Bekap funkcija omogućava vraćanje baze podataka do tačke u vremenu (Point-in-Time Recovery - PITR), obnavljanje na osnovu System Change Number-a (SCN), i obnavljanje do poslednjeg stanja. Moguće je kreirati bekap politiku putem Browser User Interface-a ili putem komandne linije.

Dostupne su sledeće opcije za bekapovanje:

* Bekapovanje na Oracle Fast Recovery Area (FRA) disk (Interni FRA)
* Bekapovanje na Oracle Cloud Infrastructure Object Storage (Oracle Object Storage)
* Bekapovanje na Network File System (NFS) lokaciju (Eksterni FRA)

Opcije za bekapovanje, vraćanje i obnavljanje mogu se koristiti sa bazama podataka koje su povezane sa backup politikom u sistemu. Mogućnost upravljanja bekapovima i postavljanja opcija za bekapovanje, vraćanje i obnavljanje je takođe dostupna.

Funkcionalnost bekapovanja omogućava kreiranje backup politike sa definisanim parametrima kao što su lokacija bekapovanja, prozor za obnavljanje, omogućavanje/onemogućavanje provere prekida i detalji ObjectStoreSwift objekta. Za sve baze podataka, kako postojeće tako i nove, moguće je pridružiti odgovarajuću backup politiku radi redovnog dnevnog backup. Kada je politika bekapovanja dodeljena bazi podataka, *dcs-agent*[[6]](#footnote-6)automatski zakazuje dnevne bekupove, kao i bekupove arhivskih logova. Podrazumevana frekvencija bekupovanja arhivskih logova je svakih 30 minuta, dok je podrazumevani raspored nivoa 0 bekapa svake nedelje, a nivoa 1 od ponedeljka do subote. Moguće je prilagoditi ili onemogućiti ovaj raspored prema potrebama.

Dcs-agent generiše i čuva izveštaj o backup-u za svaki bekap. Izveštaj o backup-u sadrži metapodatke potrebne za obnavljanje ili vraćanje baze podataka. Nije moguće izvršiti backup baza podataka koje koriste transparentno šifrovanje podataka (Transparent Data Encryption - TDE).

Obnavljanje baze podataka na Oracle Database Appliance-u obavlja se korišćenjem RMAN alata i može biti potpuno ili parcijalno. Moguće je obnoviti bazu podataka na osnovu izveštaja o backup-u, tačke u vremenu (PITR), System Change Number-a (SCN), ili na osnovu poslednjeg backup-a. Kada se koristi izveštaj o bekupu za obnavljanje, vrši se obnavljanje na određenu SCN vrednost koja se čita iz izveštaja o bekupu.

## **Backup politika**

Bekap politika definiše detalje bekapovanja. Prilikom kreiranja bekap politike, definiše se destinacija za bekapove baze podataka, bilo da je u pitanju Interni FRA (Disk) ili Eksterni FRA (NFS lokacija), ili Cloud Object Storage, i definiše se prozor za obnavljanje.

Ako se koristi Oracle Object Storage za bekapovanje i obnavljanje, mora postojati ID ili ime resursa ObjectStore.

Pri kreiranju baze podataka, moguće je odmah povezati bekap politiku sa njom. Takođe, postojećoj bazi podataka možete dodati bekap politiku ili je ažurirati po potrebi.

Takođe, prilikom kreiranja bekap politike, moguće je odabrati algoritam za kompresiju. Podržani algoritmi su OSNOVNI, NIZAK, SREDNJI i VISOK. Za bekap politiku povezanu sa Object Store-om, moguće je izabrati bilo koji algoritam za kompresiju. Podrazumevano, ako se ne navede algoritam prilikom kreiranja Object Store bekap politike, podrazumevano će biti korišćen SREDNJI algoritam. Takođe, za Disk ili NFS bekap politike, ako se ne specificira algoritam za kompresiju, podrazumevano će biti korišćen OSNOVNI algoritam. Važno je imati na umu licencne uslove za korišćenje NISKOG, SREDNJEG ili VISOKOG nivoa kompresije za Disk ili NFS bekap politiku.

## **Nivoi bekapovanja**

Potrebno je specificirati nivo bekapovanja prilikom izrade bekapa. Nivoi bekapa su:

* **Nivo 0:** Bekap hvata sve blokove u datafile-u. Podrazumevano, nivo 0 bekapovi se izrađuju svake nedelje. Moguće je prilagoditi svoja podešavanja bekapa putem Browser User Interface-a ili komandne linije. Nivo 0 bekapa je potpuni bekap i koristi se kao osnovni za nivo 1 bekapa.
* **Nivo 1:** Bekap je inkrementalni bekap koji hvata samo one blokove koji se menjaju između bekapa u svakom datafile-u. Podrazumevano, nivo 1 bekapovi se izrađuju svakodnevno, od ponedeljka do subote, i hvataju slike svakog bloka u datafile-u koji se promenio. Nivo 1 bekapovi su diferencijalni bekapovi. Bekap uključuje blokove iz najnovijeg prethodnog nivoa 1 ili nivoa 0 bekapa.
* **Dugoročni (LongTerm):** Bekap je dugoročni, ili arhivski bekap. Korišćenje ovog nivoa omogućava pravljenje bekapa koji je sveobuhvatan i izuzet od politike zadržavanja bekapa. Bekap je sveobuhvatan jer su svi fajlovi potrebni za obnavljanje i oporavak baze podataka bekupovani na jednoj lokaciji. Bekap je izuzet od politike zadržavanja koja je definisana u prozoru za obnavljanje. Za dugoročne bekapove, moraju se koristiti lokacije koje nisu FRA za destinacije bekapa koristeći opcije bekap politike.
* **Arhivelog:** Ovo je bekapovanje svih arhiveloga koji još nisu bekapovani na odredište bekapa, kao što su Interni FRA, Eksterni FRA (NFS lokacija) ili Object Store.

AutoScheduler kreira nivo 0 bekapa na *dan bekapa nivoa 0*, podrazumevano, u nedelju, i nivo 1 bekape na preostalim danima u nedelji. Moguće je prikazati listu zakazanih bekapa, urediti raspored bekapa ili onemogućiti zakazani bekapa. Takođe kreira bekape arhiveloga za bazu podataka sa podrazumevanom frekvencijom od 30 minuta.

## **Bekapovanje na disk**

Prilikom bekapovanja na disk, koristi se Oracle Fast Recovery Area (FRA) diska. FRA predstavlja skladišni direktorijum namenjen za bekapovanje i obnavljanje u Oracle bazama podataka. Ovaj direktorijum sadrži arhivirane redo logove, delove i kopije bekupa, kao i kontrolni fajl.

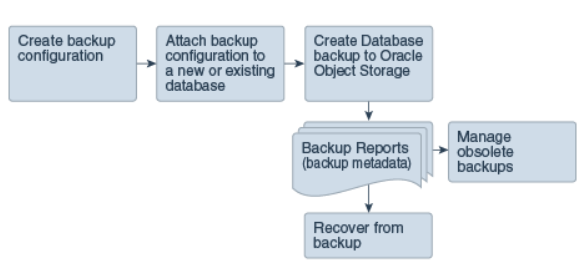
Prednosti bekapovanja na disk su sledeće:

* Brzo obnavljanje iz bekapa sačuvanih na disku
* Automatizacija upravljanja fajlovima vezanim za obnavljanje (kao što su arhivirani logovi)

Prilikom bekapovanja na disk, podaci ostaju nešifrovani, a unos lozinki nije moguć. Takođe, važno je imati na umu da ova metoda bekapovanja zahteva znatnu količinu slobodnog prostora na disku, često iznoseći od dva do tri puta više od veličine same baze podataka.

Slika 2.1 prikazuje pregled visokog nivoa radnog toka za bekapovanje i obnavljanje prilikom korišćenja diska:

1. Kreiranje bekap politike da bi definisali destinaciju bekapa (disk) i prozor za obnavljanje za destinaciju diska.
2. Kreiranje baze podataka i pridruživanje bekap politike bazi podataka. Kada je bekap politika povezana sa bazom podataka, automatski se kreiraju bekapovi arhiveloga i bekapovi baze podataka.
3. Upravljanje zastarelim bekupovima.
4. Obnavljanje baze podataka iz bekapa na disku.



**Slika 2.1** Bekapovanje i obnavljanje prilikom korišćenja diska

## **Bekapovanje na Eksterni FRA (Network File System)**

Specificiranje lokacije Network File System (NFS) za bekape omogućava sigurno i bezbedno skladištenje ili povlačenje bekapova baze podataka u bilo koje vreme.

Prednosti korišćenja NFS lokacije za bekape su:

* Sa NFS-om, moguće je pristupiti udaljenim fajlovima lokalno, montirajući lokalne fajl sisteme na mreži.
* Prilikom korišćenja NFS-a, obezbeđuje se centralizovana lokacija za bekape, bez potrebe za ručnim osvežavanjem.
* Podržava unapređene opcije bezbednosti korišćenjem Firewalova i Kerberosa.
* Podrška za šifrovane bekape korišćenjem lozinki. Opciono, moguće je postaviti lozinku za bekup, i pružiti istu prilikom operacije obnavljanja.

Pregled visokog nivoa radnog toka za bekapanje i obnavljanje prilikom korišćenja NFS lokacije:

1. Kreiranje mount point za NFS lokaciju. Mount point mora biti dostupan sa oba čvora. Korisnik oracle mora imati dozvole za čitanje/pisanje na NFS lokaciji. Ako se zahtev podnosi za daljinski server, pobrinite se da je taj server pokrenut.
2. Kreiranje bekap politiku specificirajući NFS lokaciju, i prozor za obnavljanje za NFS destinaciju.
3. Povezivanje baze podataka sa ovom bekap politikom, ili prilikom kreiranja baze podataka, ili ažuriranjem bekap politike za bazu podataka.
4. Bekapovi se automatski zakazuju, a takođe moguće je pokrenuti i ručne bekape. Možete specificirati opcije za ručne bekape putem Browser User Interface-a ili korišćenjem ODACLI komandi.
5. Upravljanje zastarelim bekupovima.
6. Obnavljanje iz bekapa. Kada se čuvaju bekapi na NFS-u, moguće je obnoviti bazu podataka na istom sistemu ili na drugom sistemu.
7. Oporavljanje baze podataka iz bekapa na NFS-u.

### **Kreiranje Mount Point za lokaciju bekapovanja putem NFS-a**

Kreiranje tačke montiranja za lokaciju mrežnog fajl sistema (NFS) kako bi postavili opciju bekapovanja i opotavka putem NFS-a.

Pratiti ove korake kako bi kreirali tačku montiranja za lokaciju putem NFS-a:

1. Pratiti ove korake na izvornoj mašini:
2. Kreirati deljenu lokaciju na izvornoj mašini i dodeliti potpune dozvole za ovaj direktorijum.

# mkdir /tmp/nfs\_test

chmod 777 /tmp/nfs\_test

1. Dodati unose u datoteku /etc/exports u formatu shared\_location destination\_IPs (permissions).

Na primer:

# cat /etc/exports

/tmp/nfs\_test 192.0.2.1(rw,sync)

/tmp/nfs\_test 192.0.2.2(rw,sync)

1. Restartovati NFS server na domaćinu.

# service nfs restart

1. Proveriti listu izvoza za unose.

# showmount -e

Export list for oda1:

/tmp/nfs\_test 192.0.2.1,192.0.2.2

1. Pratiti ove korake na klijentskoj mašini:
2. Kreirati lokaciju klijenta na klijentskoj mašini kao oracle  korisnik.

# sudo -E -u oracle mkdir /tmp/client\_location

1. Montirati ovu lokaciju sa izvornom lokacijom u formatu mount\_server:source\_folder client\_location.

# mount 192.0.2.3:/tmp/nfs\_test /tmp/client\_location

1. Proveriti da li su detalji montaže ispravni:

# mount

192.0.2.3:/tmp/nfs\_test on /tmp/client\_location type nfs

(rw,vers=4,addr=192.0.2.3,clientaddr=192.0.2.2)

Navesti ovu NFS lokaciju u konfiguraciji bekapa, bilo putem korisničkog interfejsa pretraživača ili pomoću CLI komandi.

## **Bekapovanje na Oracle Object Storage**

Oracle Object Storage je visoko sigurna, skalabilna platforma za skladištenje koja vam omogućava da sigurno i bezbedno čuvate ili povlačite bekap baze podataka u bilo koje vreme. Podaci i povezani metapodaci čuvaju se kao objekti u logičkom kontejneru (bucket-u). Oracle Database Appliance kreira izveštaj o bekapu i jedinstveni ID kako bi identifikovao svaki bekap.

Oracle Object Storage je čvrsto integrisan sa Oracle Cloud Infrastructure Identity and Access Management-om. Podaci, korisnički definisani metapodaci povezani sa podacima, i ključevi za šifrovanje automatski se šifruju korišćenjem 256-bitnog Advanced Encryption Standard (AES-256). Ključevi za šifrovanje se često rotiraju i ne mogu se onemogućiti. Podrazumevano, samo autentifikovani korisnici kojima je eksplicitno dodeljen pristup određenim resursima mogu pristupiti podacima koji su smešteni u Oracle Object Storage-u. Podaci se otpremaju i preuzimaju preko SSL krajnjih tačaka korišćenjem HTTPS protokola.

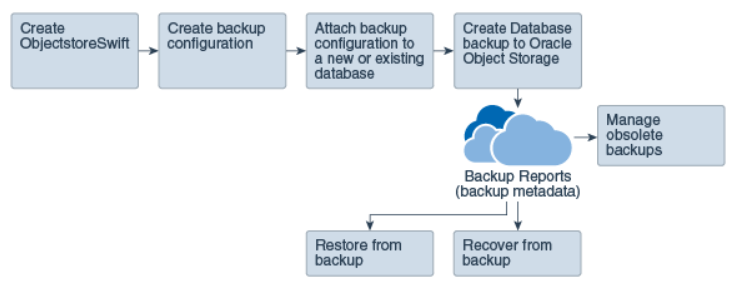
Kada se bekapuje na Oracle Object Storage, koristi se vaša bekap politika i vaše Oracle ObjectStoreSwift kredencijale. Kredencijali se čuvaju u šifrovanom Oracle novčaniku. Moguće je pridružiti kredencijale jednoj ili više bekap politika. Kredencijali se validiraju prilikom kreiranja bekap politike koja koristi Oracle Object Storage. Oni se ne validiraju protiv URL-a i zakupa.

Prednosti korišćenja Oracle Object Storage-a su sledeće:

* Bekapovi se šifruju korišćenjem lozinke za baze podataka bez TDE-a. Bitno je sačuvati lozinku koja se koristi tokom bekapa na bezbednom mestu, jer će biti potrebna prilikom operacija obnavljanja.
* Brzo obnavljanje iz bekapa. Sa Oracle Object Storage-om, možete sigurno i bezbedno koristiti Browser User Interface za čuvanje ili povlačenje podataka direktno iz oblaka, u bilo koje vreme.
* Automatizacija upravljanja fajlovima vezanim za obnavljanje (kao što su arhivirani logovi).
* Oracle Object Storage se ne pristupa direktno preko operativnog sistema, već koristi API-je za pristup skladištu na nivou aplikacije.

Slika 2.2 prikazuje pregled visokog nivoa radnog toka za bekapovanje i obnavljanje prilikom korišćenja Oracle Object Storage-a:

1. Kreiranje objekta Object Store sa vašim kredencijalima.
2. Kreiranje bekap politiku da bi definisali lokaciju bekapovanja (Object Store), ime kredencijala za Object Store, ime kontejnera (bucket) gde želite da čuvate bekape, i prozor za obnavljanje.
3. Kreiranje baze podataka i pridruživanje bekap politike, ili prilikom kreiranja baze podataka, ili ažuriranjem baze podataka sa bekap politikom nakon kreiranja baze podataka. Kada je bekap politika povezana sa bazom podataka, bekapi se automatski kreiraju.
4. Upravljanje zastarelim bekupovima.
5. Obnavljanje iz bekapa. Kada čuvate svoje bekape na Oracle Object Storage-u, možete obnoviti bazu podataka na istom sistemu ili na drugom sistemu.
6. Oporavljanje baze podataka iz bekapa na Oracle Object Storage-u.



**Slika 2.2** Bekapovanje i obnavljanje prilikom korišćenja Oracle Object Storage-a

### **Konfigurisanje postavki Proxy agenta za pristup Object Store-u**

Ako je IP adresa objektnog skladišta dostupna samo putem postavki proxy-ja koje je podešavao server Oracle Database Appliance, tada treba definisati postavke proxy agenta, kako bi agent mogao pristupiti objektnom skladištu.

Da biste kreirali bekap politiku koja koristi lokaciju objektnog skladišta, agent mora moći da pristupi URL-u objektnog skladišta.

1. Definisati postavke HttpProxyHost i HttpProxyPort u komandi update-agentconfigParameters.

# odacli update-agentConfigParameters -n HttpProxyHost -v www-proxy.test.com -n HttpProxyPort -v 80 -u

Job details

----------------------------------------------------------------

ID: 0b0cbf9b-b0ab-4523-a096-5da4e48fc825

Description: Update agent configuration parameter values [HttpProxyHost, HttpProxyPort]

Status: Created

Created: October 23, 2018 4:56:53 PM IST

Message:

Task Name Start Time End Time Status

----------------------------------------

1. Proveriti da li je ažuriranje uspelo.

# odacli describe-job -i 0b0cbf9b-b0ab-4523-a096-5da4e48fc825

Job details

----------------------------------------------------------------

ID: 0b0cbf9b-b0ab-4523-a096-5da4e48fc825

Description: Update agent configuration parameter values [HttpProxyHost, HttpProxyPort]

Status: Success

Created: October 23, 2018 4:56:53 PM IST

Message:

Task Name Start Time End Time Status

1. Pokrenuti komandu list-agentconfigParameters kako bi videli promene u postavkama proksija:

# odacli list-agentConfigParameters

Name Value Description Updated

------------------------------ ----------------------------------------

HttpProxyHost www-proxy.test.com October 23, 2018 4:56:53 PM IST

HttpProxyPort 80 October 23, 2018 4:56:53 PM IST

HttpsProxyHost Https proxy server host October 23, 2018 12:23:21 AM IST

HttpsProxyPort Https proxy server port October 23, 2018 12:23:21 AM IST

OSPatchRepos Repo list for OS patching October 23, 2018 12:23:21 AM IST

Sada je moguće koristiti Browser User Interface ili komandnu liniju za kreiranje bekap politike koja koristi lokaciju objektnog skladišta za bekap.

## **Izveštaj o backup-u**

Za svaki bekap generiše se izveštaj koji je sličan, ali ne zamenjuje katalog za obnavljanje. Izveštaj sadrži informacije i metapodatke Oracle Database Appliance-a potrebne za obnavljanje ili vraćanje baze podataka. Izveštaj o bekupu dizajniran je da prati sve informacije potrebne za obnavljanje ili vraćanje baze podataka.

Primer izveštaja o bekapu nivoa 1:

{

"id" : "2d82460c-d648-4e75-8c7d-72cc90bc442a",

"dbResId" : "b5fc646e-01a6-4c8b-8286-7633346c4329",

"tag" : null,

"dbId" : "2717054291",

"dbName" : "HRDbO",

"dbUniqueName" : "HRDbOu",

"backupType" : "REGULAR-L1",

"keepDays" : null,

"backupLocation" : "https://swiftobjectstorage.us-phoenix 1.oraclecloud.com/v1/dbaasimage/backupbucket",

"cfBackupHandle" : "c-2717054291-20171108-04",

"spfBackupHandle" : "c-2717054291-20171108-04",

"pitrTimeStamp" : "November 08, 2017 12:43:14 PM UTC",

"pitrSCN" : "1175058",

"resetLogsTimeStamp" : "November 08, 2017 09:55:34 AM UTC",

"resetLogsSCN" : "1112268",

"oraHomeVersion" : "12.2.0.1.170814 (26723265, 26609817)",

"sqlPatches" : "25811364,26609817",

"backupLogLoc" : "https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/dbaasimage/backupbucket/scaoda702c1n1/rmanlog/HRDbOu/2717054291/2017-11-08/rman\_backup\_2017-11-08\_12-42-41.0545.log",

"tdeWalletLoc" : null,

"dbConfigLoc" : "https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/dbaasimage/backupbucket/scaoda702c1n1/dbconfig/HRDbOu/2717054291/2017-11-08/DBCONFIG\_TAG20171108T124407\_2017-11-08\_12-44-07.0533.tar.gz",

"name" : "Backup\_Report\_HRDbO",

"createTime" : "November 08, 2017 12:42:08 PM UTC",

"state" : {

"status" : "CONFIGURED"

},

"updatedTime" : "November 08, 2017 12:44:12 PM UTC",

"backupReportLogDetail" : "https://swiftobjectstorage.us-phoenix-1.oraclecloud.com/v1/dbaasimage/backupbucket/scaoda702c1n1/rmandetaillogreport/HRDbOu/2717054291/2017-11-08/rman\_list\_backup\_detail\_2017-11-08\_12-44-04.0362.log",

"dbInfo" : {

"dbClass" : "OLTP",

"dbType" : "RAC",

"dbShape" : "odb1",

"dbEdition" : "EE",

"dbStorage" : "ASM"

},

"dbDataSize" : "1542M",

"dbRedoSize" : "16403M"

}

## **Opcije oporavka**

Oporavak u Oracle Database Appliance-u uvek vrši potpuno vraćanje ili oporavak baze podataka. Opcije oporavka su NAJNOVIJI, PITR, SCN i izveštaj o bekupu.

Dostupni su sledeći tipovi oporavka:

* **NAJNOVIJI:** Vrši potpuni oporavak (validni bekupovi i svi potrebni arhivirani logovi i online redo logovi moraju biti dostupni)
* **PITR:** Oporavak do tačke u vremenu koji vrši oporavak na određeni vremenski pečat unutar trenutne inkarnacije baze podataka
* **SCN:** Oporavak zasnovan na SCN-u koji vrši oporavak do određenog SCN-a unutar trenutne inkarnacije baze podataka
* **Izveštaj o bekupu:** Vrši oporavak na osnovu SCN-a iz izveštaja o bekupu. Opcija izveštaj o bekupu slična je SCN-u, osim što oporavak koristi SCN iz izveštaja o bekupu umesto određenog SCN-a.

### **Point-in-Time Database Recovery (PITR)**

Oporavak baze podataka do tačke u vremenu vraća bazu podataka do određene tačke u vremenu. Za ovaj tip oporavka mora se navesti vremenski pečat.

Oporavak do tačke u vremenu se često koristi u sledećim situacijama:

* Želite da vratite bazu podataka do SCN-a pre greške korisnika ili administratorske greške.
* Potpuni oporavak baze podataka nije uspeo jer svi potrebni arhivirani redo logovi nisu bili dostupni.
* Neuspeo je proces nadogradnje baze podataka.
* Kreirate test bazu podataka ili izveštajnu bazu podataka iz bekapova produkcione baze podataka.

### **Oporavak na osnovu System Change Number-a**

System Change Number (SCN) je oznaka koja definiše završenu verziju baze podataka u određenom trenutku. Oracle dodeljuje svakoj završenoj transakciji jedinstveni SCN. Moguće je oporaviti bazu podataka do najnovijeg vremena ili do određenog SCN-a.

Baza podataka se vraća iz bekupova napravljenih pre ciljanog vremena, a zatim se primenjuju inkrementalni bekupovi i logovi kako bi ponovo kreirali sve promene između vremena bekupova datoteka podataka i krajnje tačke oporavka. Kada se krajnja tačka specificira kao SCN, baza podataka primenjuje redo logove i zaustavlja se nakon svakog redo niti ili nakon određenog SCN-a, u zavisnosti šta se prvo dogodi. Kada se krajnja tačka specificira kao vreme, baza podataka interno određuje odgovarajući SCN za navedeno vreme, a zatim se oporavlja do ovog SCN-a.

SCN je osnovni mehanizam za održavanje konzistentnosti podataka u Oracle bazi podataka. Sa svakim izvršavanjem, broj SCN-a se povećava. Evo nekih primera oblasti koje koriste SCN:

1. **Redo zapisi.** Svaki redo zapis ima verziju SCN-a u zaglavlju redo zapisa (redo zapisi mogu imati nejedinstven SCN). Dobijeni redo zapisi iz dve niti (kao u slučaju RAC-a) se sortiraju po redosledu SCN-a, čime se održava strogi sekvencijalni redosled.
2. **Blokovi podataka.** Svaki blok podataka ima SCN bloka (verziju bloka). Pored toga, vektor promene u redo zapisu takođe ima očekivani SCN bloka. Ovo znači da se vektor promene može primeniti samo na jednu verziju bloka. Kod provere da li ciljani SCN u vektoru promene odgovara SCN-u bloka pre primene redo zapisa. Ako dođe do neslaganja, biće izbačene greške o korupciji.
3. **Čitanje konzistentnosti.** Svaki upit ima okruženje upita koje uključuje SCN na početku upita. Sesija može videti transakcione promene samo ako je SCN potvrde te transakcije niži od SCN-a okruženja upita.
4. **Potvrda.** Svaka potvrda generiše SCN (SCN potvrde) koji označava granicu transakcije. Moguće su i grupne potvrde.

## **Zastareli backup-ovi baze podataka**

Važno je ukloniti starije bekupove kada postanu zastareli. Tip bekupa i prozor za obnavljanje koje definišete u bekap politici baze podataka određuju kada bekupovi baze podataka postaju zastareli.

* *Za bekapove nivoa 0 i nivoa 1*, moguće je pokrenuti komandu odacli delete-backup za određeni identifikator resursa baze podataka kako bi obrisali zastarele bekupove na osnovu definisanog prozora za obnavljanje.
* *Za bekupove dugoročnog čuvanja*, treba obezbediti odgovarajući izveštaj o bekupu kao ulaz i koristiti komandu odacli delete-backup da bi obrisali bekupove iz Oracle Object Storage-a.

Obično, stariji bekupovi nivoa 0 i nivoa 1 smatraju se zastarelim i brišu se na osnovu definisanog prozora za oporavak, na sledeći način:

* Disk: 1-14 dana
* Object Storage: 1-31 dan

# **Kreiranje backup politike za bazu podataka**

U ovom poglavlju biće objašnjeno kako kreirati politiku za bekapovanje baze podataka na internu ili eksternu Oracle Fast Recovery Area (FRA) lokaciju ili na lokaciju Objcet Store-a korišćenjem Browser User Interface.

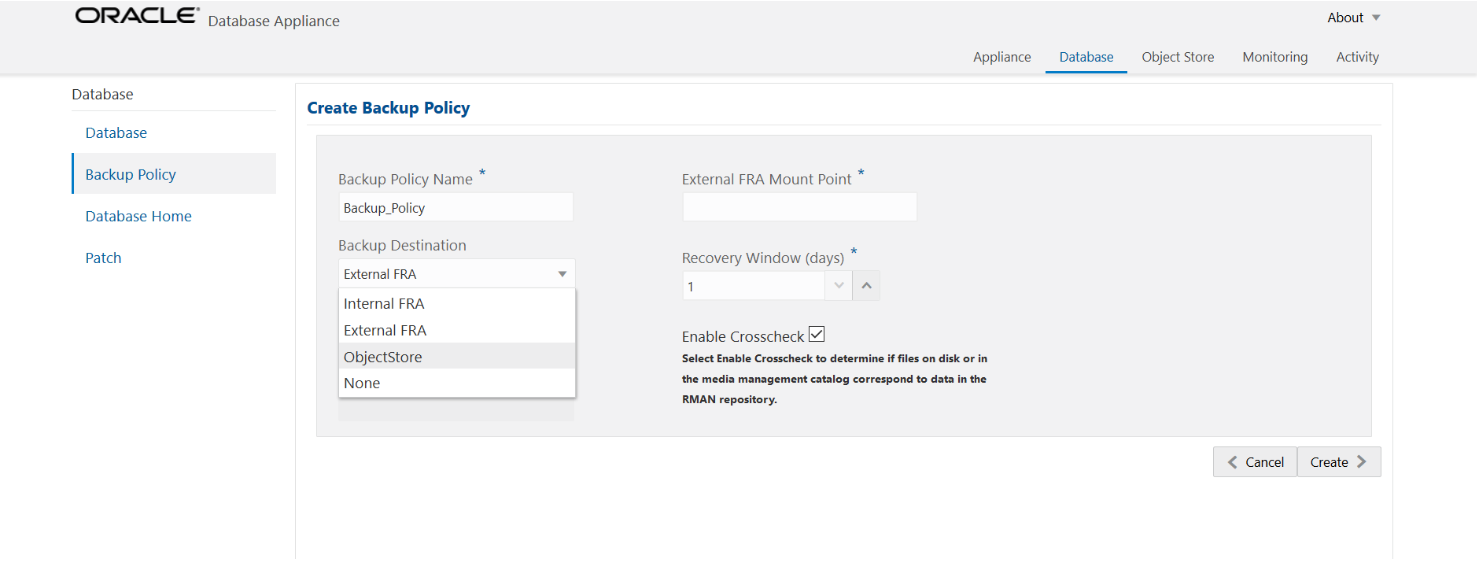
Bekap politika definiše bekapovanje, uključujući lokaciju gde želite čuvati bekape i prozor za obnavljanje. Bekap na disku zahteva dosta prostora na disku (od dva do tri puta veću veličinu baze podataka).

Treba pratiti sledeće korake za kreiranje bekap politike putem Browser User Interface:

1. Kliknuti na tab **Database** u korisničkom interfejsu pretraživača.
2. Kliknuti na **Backup Policy** u levoj navigaciji da bi prikazali listu bekap politika.
3. Kliknuti na **Create Backup Policy**.
4. Uneti ime za bekap politiku i izabrati broj dana za prozor za obnavljanje. Izabrati **Enable Crosscheck** da bi utvrdili da li se fajlovi na disku ili u katalogu za upravljanje medijima poklapaju sa podacima u RMAN repozitorijumu. Izabrati jednu od sledećih opcija kao destinaciju bekapa:

* Za bekapovanje na disk, izaberati **Internal FRA** kao destinaciju bekapovanja.
* Za bekapovanje u oblak, izaberati **ObjectStore** kao destinaciju bekapa. Ako postoji više od jednog ObjectStore-a, tada treba izaberati ime ObjectStore-a iz liste. Uneti ime u polje Container Name.
* Za bekapovanje na NFS lokaciju, izaberati **External FRA** kao destinaciju bekapovanja i navedite lokaciju NFS mount pointa.
* Ako ne želite definisati destinaciju, izaberite **None**.

Ime bekap politike mora početi slovom i sadržati samo alfanumeričke karaktere. Slika 3.1 prikazuje stranicu kreiranja backup politike.



**Slika 3.1** Kreiranje backup politike

1. Kliknuti na **Create**. Kliknuti na **Yes** kako bi potvrdili kreiranje bekap politike. Konfiguracija bekapa je spremna.

Treba ažurirati postojeću bazu podataka kako bi pridružili konfiguraciju bekapa bazi podataka koristeći ID ili ime. Kada se pridruži konfiguracija bekapa bazi podataka, dcs-agent zakazuje automatske dnevne bekupove za bazu podataka.

# **Alati za upravljanje backup-om i oporavkom baze podataka**

Oracle pruža više rešenja za izvođenje bekapa i oporavka. Sledeća rešenja su dostupna prilikom implementacije strategije bekapa i oporavka:

* **Recovery Manager (RMAN)**

Recovery Manager je potpuno integrisan sa Oracle bazom podataka kako bi izvršio niz aktivnosti bekapa i oporavka, uključujući održavanje RMAN repozitorijuma sa istorijskim podacima o bekapovima. RMAN-u moguće je pristupiti putem komandne linije ili putem Oracle Enterprise Manager-a.

* **Oracle Enterprise Manager Cloud Control**

Oracle Enterprise Manager Cloud Control (Cloud Control) pruža grafički korisnički interfejs i mogućnosti planiranja za RMAN. Unosite parametre posla, specificirate raspored posla, i Cloud Control pokreće RMAN da sprovede bekup i oporavak operacije.

* **Zero Data Loss Recovery Appliance (Recovery Appliance)**

Recovery Appliance je sistem zaštite podataka na nivou oblaka koji pruža zaštitu podataka za sve Oracle baze podataka u preduzeću. Integrisan sa RMAN-om i Cloud Control-om, Recovery Appliance pruža jedinstveni repozitorijum za bekape više baza podataka.

* **Korisnički upravljani bekap-ovi i oporavak**

U ovom rešenju, vi izvodite bekape i oporavak sa mešavinom komandi operativnog sistema domaćina i SQL\*Plus oporavka. Vi ste odgovorni za određivanje svih aspekata kada i kako se bekapi i oporavci izvršavaju.

Ova rešenja su podržana od strane Oracle-a i potpuno su dokumentovana, ali RMAN je preferirano rešenje za bekup i oporavak baza podataka. RMAN pruža zajednički interfejs za bekape preko različitih operativnih sistema domaćina, i nudi nekoliko tehnika bekapa koje nisu dostupne kroz korisnički upravljane metode.

## **Recovery Manager – RMAN**

Recovery Manager (RMAN) je Oracle alat koji može da pravi rezervne kopije, vraća i oporavlja datoteke baze podataka. Proizvod je funkcija Oracle servera baze podataka i ne zahteva posebnu instalaciju.

Recovery Manager je klijent/server aplikacija koja koristi sesije servera baze podataka za izvođenje rezervnih kopija i oporavka. Čuva metapodatke o svojim operacijama u kontrolnoj datoteci ciljne baze podataka i, opciono, u šemi kataloga za oporavak u Oracle bazi podataka.

Moguće je pokrenuti RMAN kao izvršni program komandne linije iz operativnog sistema ili koristiti neke RMAN funkcije kroz Enterprise Manager GUI.

Većina proizvodnih sistema baza podataka postavlja stroge zahteve za rezervne kopije i oporavak. Kao DBA zadužen za rezervne kopije i oporavak, morate:

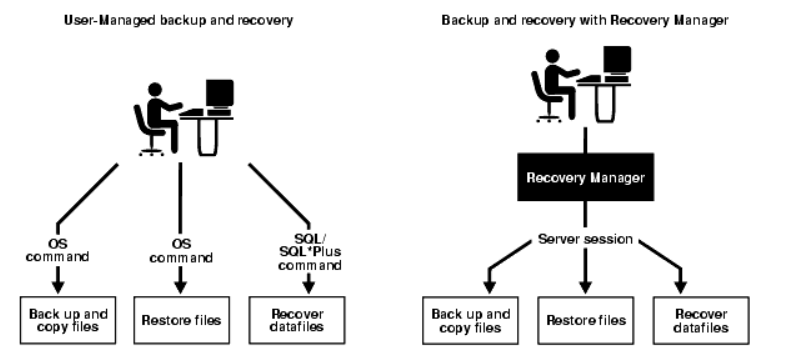
* Upravljati složenošću operacija pravljenja rezervnih kopija i oporavka
* Minimizirati mogućnost ljudske greške
* Učiniti rezervne kopije skalabilnim i pouzdanim
* Iskoristiti sav dostupni hardver za medije
* Učiniti rezervne kopije proporcionalnim veličini transakcijskih promena, a ne veličini baze podataka
* Učiniti vreme oporavka proporcionalnim količini podataka koji se oporavljaju

Postoje dve osnovne metode za obavljanje ovih zadataka pravljenja rezervnih kopija i oporavka:

* Korišćenje komandi operativnog sistema za izvođenje operacija pravljenja rezervnih kopija i vraćanja, i SQL ili SQL\*Plus naredbi za izvođenje oporavka
* Korišćenje Recovery Manager-a za pravljenje rezervnih kopija, vraćanje i oporavak

Kao što je prikazano na Slici 4.1, RMAN koristi sesije servera za izvođenje operacija pravljenja rezervnih kopija i oporavka i čuva metapodatke u repozitorijumu. RMAN automatizuje pravljenje rezervnih kopija i oporavak, dok korisnički upravljan metod zahteva od vas da pratite sve datoteke baze podataka i rezervne kopije. Na primer, umesto da tražimo rezervne kopije za svaku datoteku podataka, kopiramo ih na odgovarajuće mesto koristeći komande operativnog sistema i biramo koje logove da primenimo, RMAN automatski upravlja ovim zadacima.

Prednost korišćenja RMAN-a je posebno izražena ako se koristi Oracle Managed Files. Kada dopustimo Oracle-u da imenuje i upravlja našim datotekama podataka, kontrolnim datotekama i online redo logovima, sistem postaje lakši za korišćenje. S druge strane, može biti teže za praćenje naziva različitih datoteka baze podataka jer ih nismo sami imenovali. RMAN korisnici nemaju ovaj problem jer RMAN upravlja svim vođenjem evidencije.



**Slika 4.1**Poređenje RMAN automatizovanih i korisnički upravljanih procedura

Pored očigledne prednosti automatizacije, RMAN pruža niz drugih korisnih funkcija. Tabela 4.1 poredi neke od razlika između RMAN metodologije i tradicionalne korisnički upravljane metodologije.

**Tabela 4.1** Poređenje između RMAN i korisnički upravljanih metoda

|  |  |
| --- | --- |
| **Recovery Manager** | **User-Managed Method** |
| Koristi API za upravljanje medijima tako da RMAN radi besprekorno sa softverom za upravljanje medijima trećih strana. Više od 20 proizvođača podržava ovaj API. | Nema podršku za objavljeni API. |
| Kada pravi rezervne kopije online datoteka, RMAN ponovo čita fragmentisane blokove podataka kako bi dobio konzistentan prikaz. Ne morate stavljati online tablespaces u režim rezervne kopije prilikom izvođenja rezervnih kopija. | Zahteva stavljanje online tablespaces u režim rezervne kopije pre nego što ih napravite, a zatim izlazak tablespaces iz ovog režima nakon što je rezervna kopija završena. Ozbiljni problemi sa performansama i upravljanjem bazom podataka mogu se pojaviti ako zaboravite da izvadite tablespaces iz režima rezervne kopije nakon što je online rezervna kopija završena. |
| Vrši inkrementalne rezervne kopije, koje prave rezervne kopije samo onih blokova podataka koji su promenjeni nakon prethodne rezervne kopije. Možete oporaviti bazu podataka koristeći inkrementalne rezervne kopije, što znači da možete oporaviti NOARCHIVELOG bazu podataka. Međutim, inkrementalne rezervne kopije NOARCHIVELOG baze podataka možete praviti samo nakon konzistentnog gašenja. | Pravi rezervne kopije svih blokova, a ne samo promenjenih blokova. Ne omogućava vam oporavak baze podataka NOARCHIVELOG. |
| Računa kontrolne sume za svaki blok tokom rezervne kopije i proverava da li su blokovi oštećeni prilikom pravljenja rezervne kopije ili vraćanja. Mnoge provere integriteta koje se obično izvršavaju prilikom izvršavanja SQL upita takođe se vrše prilikom pravljenja rezervne kopije ili vraćanja. | Ne pruža proveru grešaka. |
| Isključuje blokove koji nikada nisu korišćeni iz rezervnih kopija datoteka podataka tako da su uključeni samo blokovi podataka koji su zapisani u rezervnu kopiju. | Uključuje sve blokove podataka, bez obzira da li sadrže podatke. |
| Koristi repozitorijum da bi izvestio o ključnim informacijama, uključujući:   * Šemu baze podataka u određenom trenutku * Koje datoteke trebaju rezervnu kopiju * Koje datoteke nisu imale rezervnu kopiju u određenom broju dana * Koje rezervne kopije mogu biti obrisane jer su redundantne ili se ne mogu koristiti za oporavak * Trenutne RMAN postavke koje su trajne | Ne uključuje funkcionalnost izveštavanja. |
| Čuva RMAN skripte u katalogu za oporavak. | Zahteva skladištenje i održavanje skripti baziranih na operativnom sistemu. |
| Omogućava lako kreiranje duplicirane verzije produkcijske baze podataka u svrhe testiranja, ili lako kreiranje ili pravljenje rezervne kopije standby baze podataka. | Zahteva od vas da pratite složen postupak prilikom kreiranja test ili standby baze podataka. |
| Vrši provere da bi utvrdio da li su rezervne kopije na disku ili u katalogu medija još uvek dostupne. | Zahteva od vas da ručno pronađete i testirate rezervne kopije. |
| Automatski paralelizuje operacije pravljenja rezervnih kopija i oporavka. | Zahteva ručno paralelizovanje određivanjem koje datoteke trebate da se rezervišu, a zatim paralelno izdavanjem komandi operativnog sistema. |
| Testira da li datoteke mogu biti rezervisane ili vraćene bez stvarnog izvršavanja rezervne kopije ili oporavka. | Zahteva od vas da zapravo vratite rezervne kopije datoteka pre nego što možete izvršiti probni oporavak rezervnih kopija. |
| Automatski vrši prelazak na arhivirane logove. Ako RMAN otkrije oštećeni ili nedostajući log tokom rezervne kopije, tada smatra sve logove i kopije logova navedene u repozitorijumu kao alternativne kandidate za rezervnu kopiju. | Ne može preći na alternativni arhivirani log ako rezervna kopija naiđe na problem. |

**Pregled RMAN okruženja**

RMAN okruženje sastoji se od alata i baza podataka koji igraju ulogu u strategiji rezervnih kopija i oporavka. Tipično RMAN podešavanje koristi sledeće:

* RMAN izvršni program
* Ciljnu bazu podataka
* Bazu podataka za katalogom oporavka
* Softver za upravljanje medijima

Od ovih komponenti, samo su RMAN izvršni program i ciljna baza podataka obavezni. RMAN automatski čuva svoje metapodatke u kontrolnoj datoteci ciljne baze podataka, pa je baza podataka kataloga oporavka opciona. Ipak, održavanje kataloga oporavka se snažno preporučuje. Ako napravite katalog na odvojenom računaru, i ako produkcijski računar potpuno otkazuje, tada imate sve informacije o vraćanju i oporavku koje su vam potrebne u katalogu.

**Izvršni program RMAN** automatski se uključuje prilikom instalacije Oracle softvera. Njegova lokacija zavisi od platforme i obično se nalazi na istom mestu kao i ostali Oracle izvršni programi. Na Unix sistemima, na primer, RMAN izvršni program se nalazi u $ORACLE\_HOME/bin direktorijumu.

Da bi pokrenuli izvršni program, jednostavno treba uneti ime datoteke na komandnoj liniji. Na primer, na UNIX sistemu, uneti:

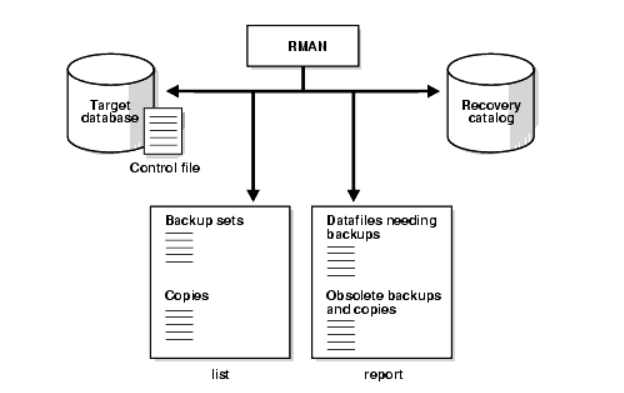
% rman

**Ciljna baza podataka** je baza podataka koju RMAN pravi rezervne kopije, vraća ili oporavlja. Moguće je koristiti jedan katalog oporavka zajedno sa više ciljnih baza podataka. Na primer, pretpostavimo da vaš centar podataka sadrži 10 baza podataka različitih veličina. Možete koristiti jedan katalog oporavka koji se nalazi u drugom centru podataka da upravljate metapodacima svih ovih baza podataka.

**Repozitorijum RMAN-a** je skup metapodataka koje RMAN koristi da sačuva informacije o ciljnoj bazi podataka i njenim operacijama rezervne kopije i oporavka. Između ostalog, RMAN čuva informacije o:

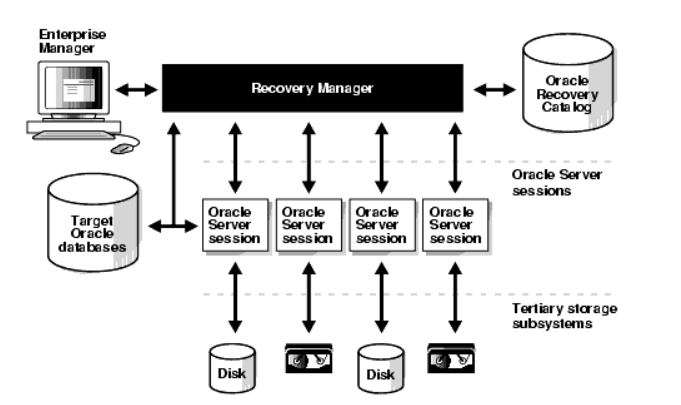
* Setovima i delovima rezervnih kopija
* Kopijama slika (uključujući i arhivirane redo logove)
* Proxy kopijama
* Šemi ciljne baze podataka
* Trajnim konfiguracionim postavkama

Ovim metapodacima možete pristupiti izdavanjem LIST, REPORT i SHOW komandi u RMAN interfejsu, ili korišćenjem SELECT instrukcija na kataloškim pogledima (samo ako koristite katalog oporavka). Slika 4.2 ilustruje kako RMAN izdaje liste i izveštaje.



**Slika 4.2** RMAN Liste i izveštaji

Moguće je ili kreirati **katalog oporavka** u kojem će se čuvati repozitorijum, ili dozvoliti da RMAN ekskluzivno čuva repozitorijum u kontrolnoj datoteci ciljne baze podataka. Slika 4.3 prikazuje RMAN koji koristi katalog oporavka.

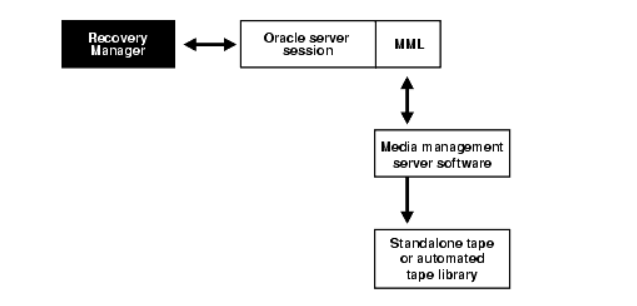


**Slika 4.3** RMAN sa opcionim katalogom oporavka

Iako RMAN može izvršiti sve glavne operacije rezervne kopije i oporavka koristeći samo kontrolnu datoteku, treba obratiti pažnju na ove prednosti korišćenja kataloga:

* Neke RMAN komande i operacije funkcionišu samo sa katalogom.
* Katalog oporavka čuva istorijske informacije o rezervnim kopijama koje mogu biti prepisane u kontrolnoj datoteci.
* Katalog oporavka čuva informacije o rezervnim kopijama iz različitih inkarnacija baze podataka.
* Katalog oporavka se održava isključivo od strane RMAN-a; ciljna baza podataka nikada mu ne pristupa direktno. RMAN automatski propagira informacije o strukturi baze podataka, arhiviranim redo logovima, setovima rezervnih kopija i kopijama datoteka u katalog oporavka iz kontrolne datoteke ciljne baze podataka. Takođe možete ručno propagirati ove informacije u katalog koristeći komandu RESYNC CATALOG.

Da bi sačuvao rezervne kopije na traku, RMAN zahteva **media menadžera**. Media menadžer je softverski program koji učitava, označava i isčitava sekvencijalne medije kao što su trake koje se koriste za pravljenje rezervnih kopija i oporavak podataka. Slika 4.4 prikazuje arhitekturu za media menadžera integrisanog sa Oracle-om.



**Slika 4.4** Arhitektura za MML integrisanu sa Oracle-om

Oracle serverska sesija je iste vrste serverske sesije koja se koristi kada se klijent kao što je SQL\*Plus poveže sa bazom podataka. Media menadžment biblioteka (MML) na slici 4.4 predstavlja biblioteku softvera za upravljanje medijima koju obezbeđuje proizvođač, a koja može da komunicira sa Oracle-om. Oracle poziva MML softverske rutine da pravi rezervne kopije i vraća datoteke podataka na i sa medija koje kontroliše media menadžer.

### **Pokretanje RMAN-a i povezivanje sa bazom podataka**

Pre nego što se izvrše bilo koje operacije koristeći RMAN, moramo se povezati sa ciljnom bazom podataka.

Klijent RMAN-a se pokreće izdavanjem komande rman na komandnom promptu tvog operativnog sistema. RMAN prikazuje prompt za komande kao što je prikazano u sledećem primeru:

% rman

RMAN>

Veze RMAN-a sa bazom podataka se specificiraju i autentifikuju na isti način kao i veze SQL\*Plus-a sa bazom podataka. Jedina razlika je što veze RMAN-a sa ciljnom ili pomoćnom bazom podataka zahtevaju privilegije SYSDBA ili SYSBACKUP. Bilo kom korisniku može biti dodeljena ova privilegija.

Moguće je povezati sa bazom podataka pomoću opcija komandne linije ili korišćenjem komande CONNECT TARGET. U sledećem primeru, pokrećemo RMAN, a zatim se povezujemo sa ciljnom bazom podataka putem Oracle Net-a kao korisnik sbu, koji je kreiran sa privilegijom SYSBACKUP. RMAN će tražiti lozinku.

% rman

RMAN> CONNECT TARGET "sbu@prod AS SYSBACKUP"

target database Password: password

connected to target database: PROD (DBID=39525561)

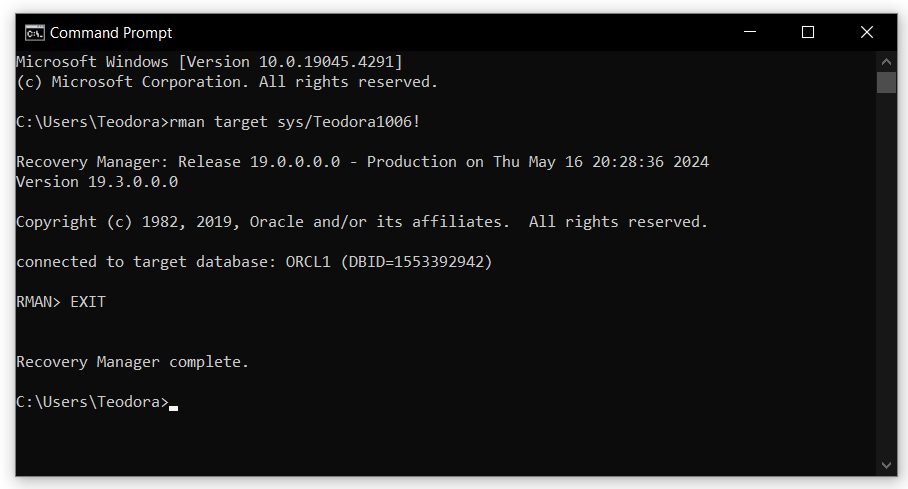
Kada koristimo višenamensku arhitekturu, možemo se povezati na korensku ili na određenu priključnu bazu podataka (PDB).

Da izađemo iz RMAN klijenta, treba uneti EXIT na RMAN promptu:

RMAN> EXIT

**Praktični primeri:**

**Primer 4.1:** Korišćenjem komande rman u komandnoj liniji, uspostavljena je veza sa Recovery Manager (RMAN) alatom. Zatim je izvršena konekcija na ciljnu bazu podataka koristeći privilegije korisnika "sys" sa odgovarajućom lozinkom, omogućavajući pristup operacijama oporavka i upravljanja bazom podataka kroz RMAN. Nakon završetka potrebnih operacija, sesija u RMAN-u je zatvorena komandom EXIT.



### **Pravljenje backup-a baze podataka u ARCHIVELOG režimu**

Ako je baza podataka pokrenuta u ARCHIVELOG režimu, tada možemo praviti rezervnu kopiju baze podataka dok je otvorena.

Rezervna kopija se naziva nekonzistentna ako sadrži promene nakon njenog kontrolnog tačka. Ako imate arhivirane redo logove potrebne za oporavak rezervne kopije, rezervne kopije otvorene baze podataka su podjednako efikasne za zaštitu podataka kao i konzistentne rezervne kopije.

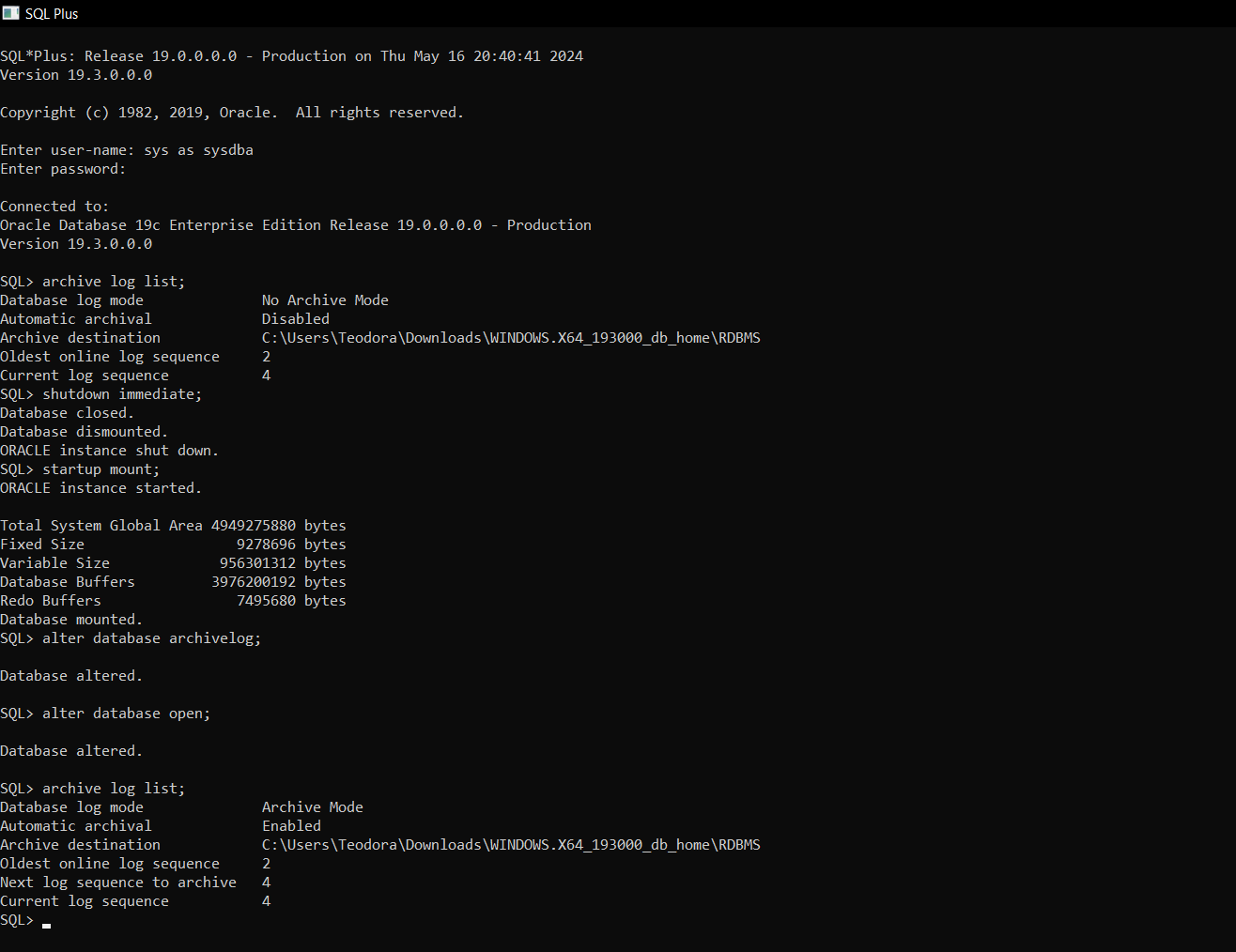
Da bi napravili rezervnu kopiju baze podataka i arhiviranih redo logova dok je baza podataka otvorena potrebno je sledeće:

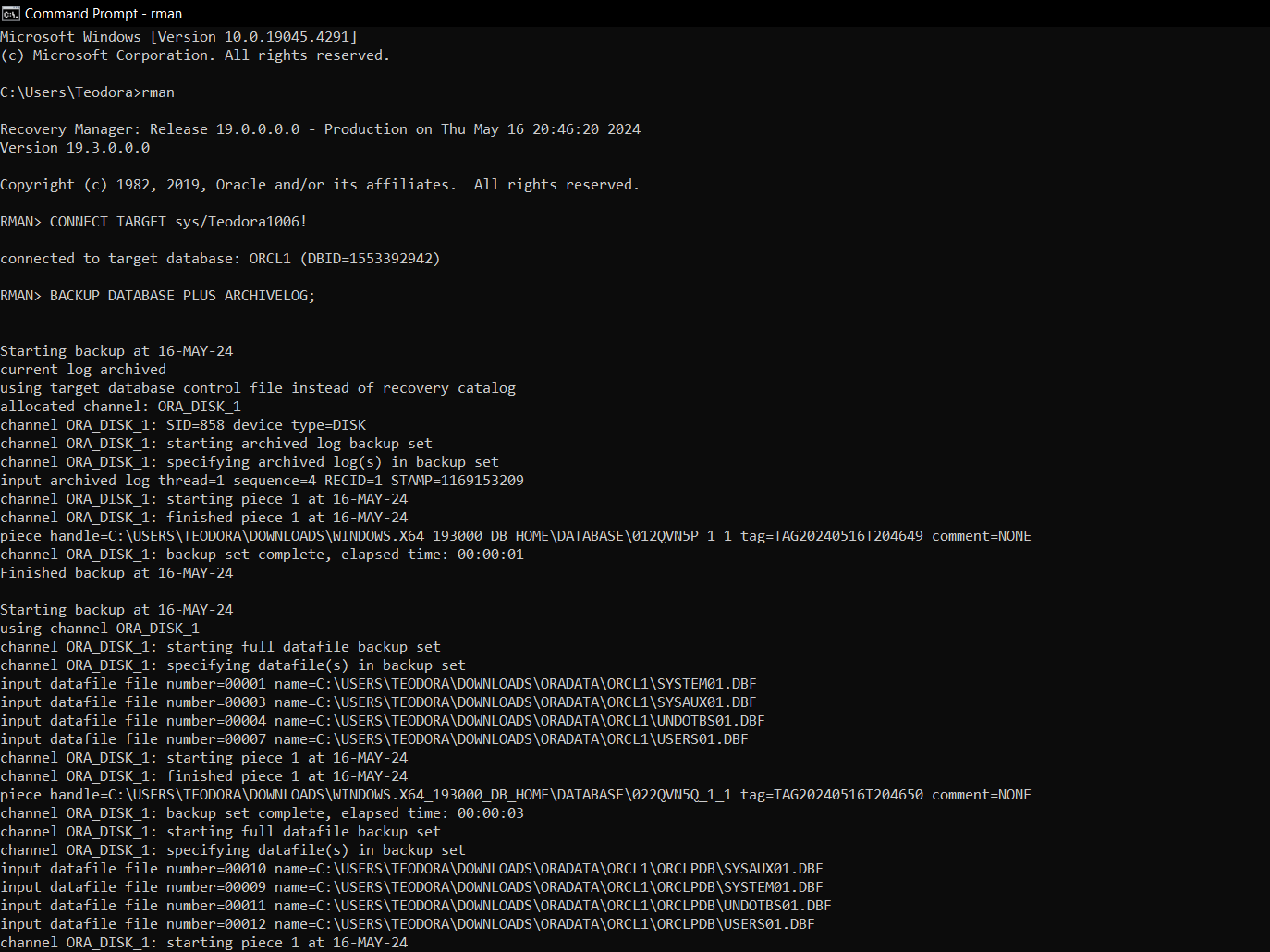
1. Pokrenuti RMAN i povezati se sa ciljnom bazom podataka kako je opisano u poglavlju "Pokretanje RMAN-a i povezivanje sa bazom podataka".
2. Pokrenuti BACKUP DATABASE komandu. Na primer, uneti sledeću komandu na RMAN promptu da bi napravili rezervnu kopiju baze podataka i svih arhiviranih redo log fajlova na podrazumevani uređaj za rezervne kopije:

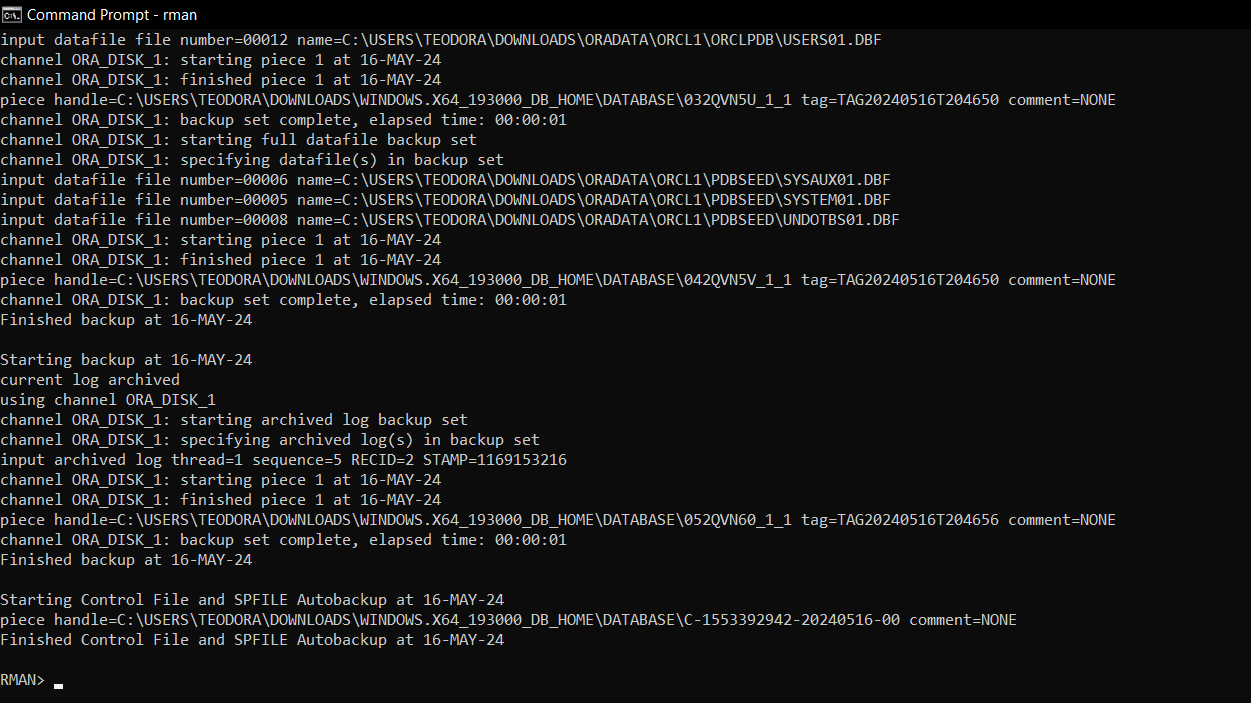
RMAN> BACKUP DATABASE PLUS ARCHIVELOG;

**Praktični primeri:**

**Primer 4.2**: Prvo je izvršena provera trenutnog režima naredbom archive log list; zatim zaustavljanje baze naredbom shutdown immediate; zatim pokretanje baze u MOUNT režim naredbom startup mount; zatim sledi prebacivanje baze u ARCHIVLOG režim naredbom alter database archivelog; i na kraju izvršeno je otvaranje baze naredbom alter database open. U RMAN izvršeno je pokretanje komande BACKUP DATABASE PLUS ARCHIVELOG za kreiranje backup-a baze podataka i arhiviranih redo log fajlova.







### **Pravljenje bakcup-a baze podataka u NOARCHIVELOG režimu**

Ako je baza podataka pokrenuta u NOARCHIVELOG režimu, tada je jedina validna rezervna kopija baze podataka konzistentna rezervna kopija.

Da bi rezervna kopija bila konzistentna, baza podataka mora biti montirana nakon konzistentnog gašenja. Oporavak nije specifično potreban nakon vraćanja rezervne kopije, ali biste izgubili sve transakcije napravljene nakon rezervne kopije. Možete se oporaviti sa arhiviranim logovima iz konzistentne rezervne kopije kako biste smanjili gubitak podataka.

Da bi napravili konzistentnu rezervnu kopiju baze podataka potrebno je sledeće:

1. Pokrenuti RMAN i povezati se sa ciljnom bazom podataka kako je opisano u poglavlju "Pokretanje RMAN-a i povezivanje sa bazom podataka".
2. Konzistentno ugasiti bazu podataka, a zatim je montirati. Na primer, uneti sledeće komande da bi garantovali da je baza podataka u konzistentnom stanju za rezervnu kopiju:

RMAN> SHUTDOWN IMMEDIATE;

RMAN> STARTUP FORCE DBA;

RMAN> SHUTDOWN IMMEDIATE;

RMAN> STARTUP MOUNT;

1. Izvršiti komandu BACKUP DATABASE. Na primer, unesi sledeću komandu na RMAN promptu da bi napravili rezervnu kopiju baze podataka na podrazumevani uređaj za bekap:

RMAN> BACKUP DATABASE;

Sledeća varijacija komande kreira kopije slika svih podataka u bazi podataka:

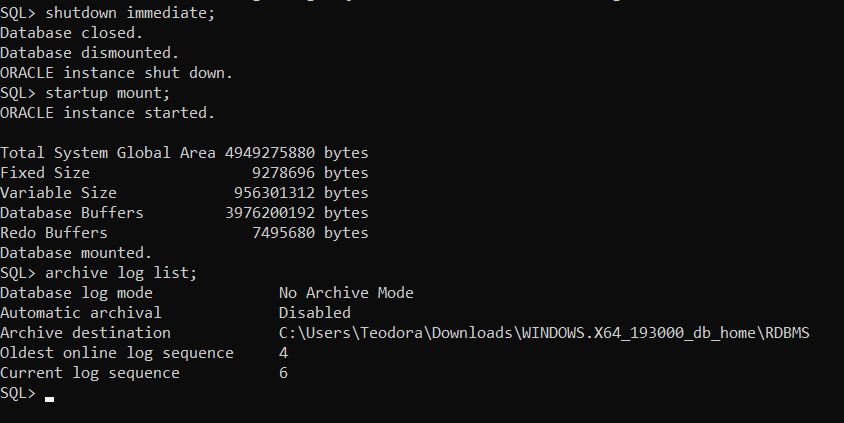
RMAN> BACKUP AS COPY DATABASE;

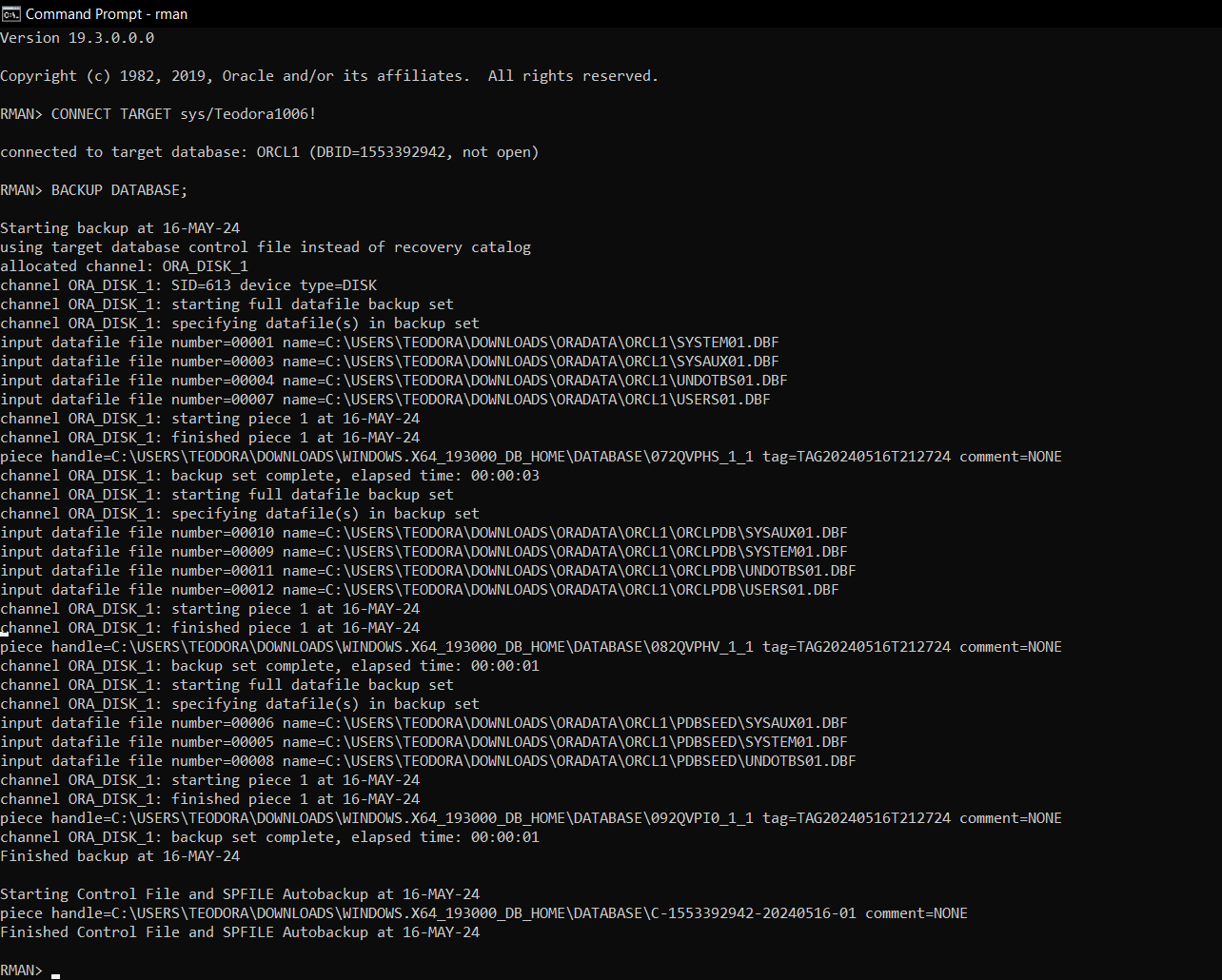
1. Otvoriti bazu podataka i nastavi sa normalnim radom. Sledeća komanda otvara bazu podataka:

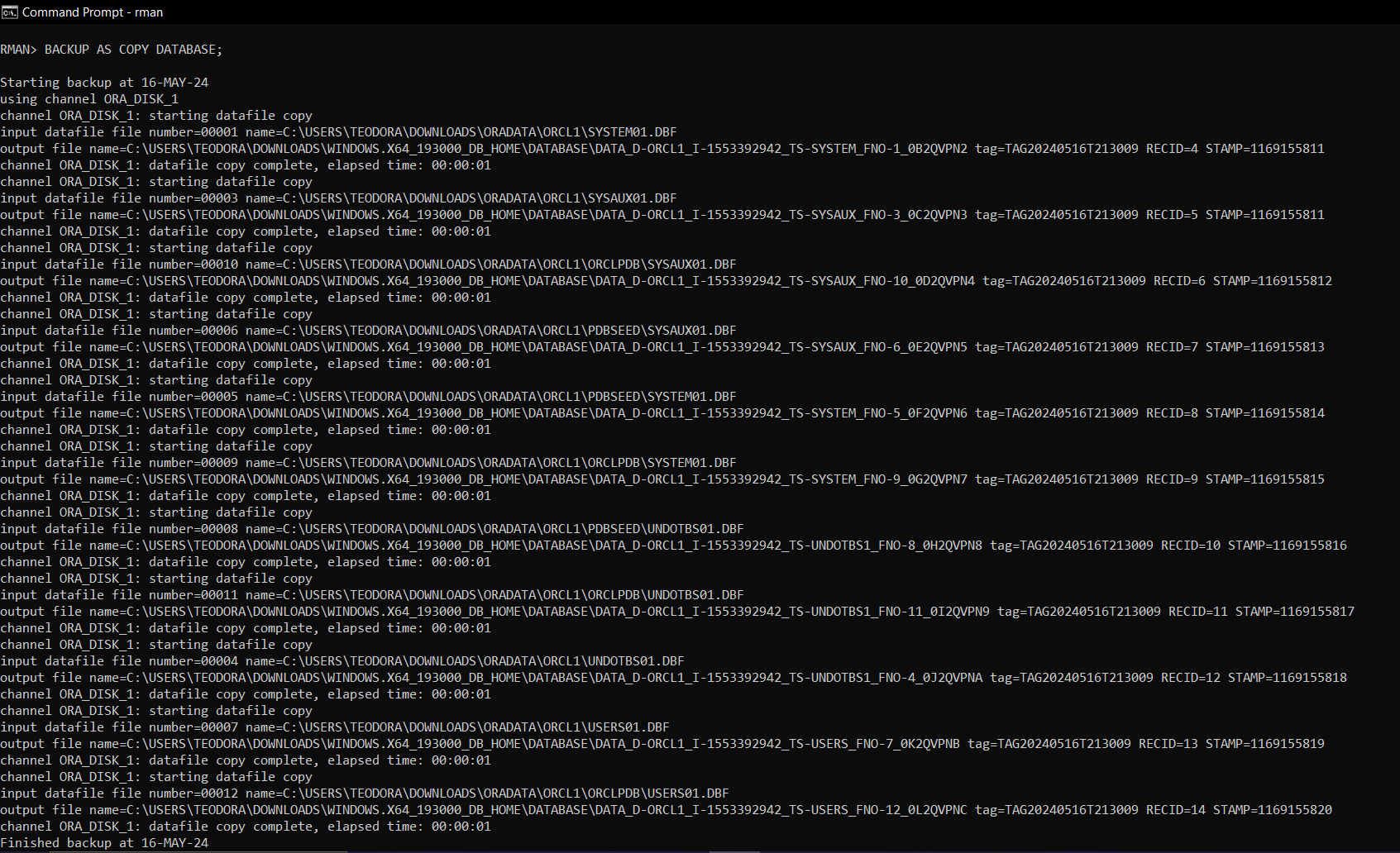
RMAN> ALTER DATABASE OPEN;

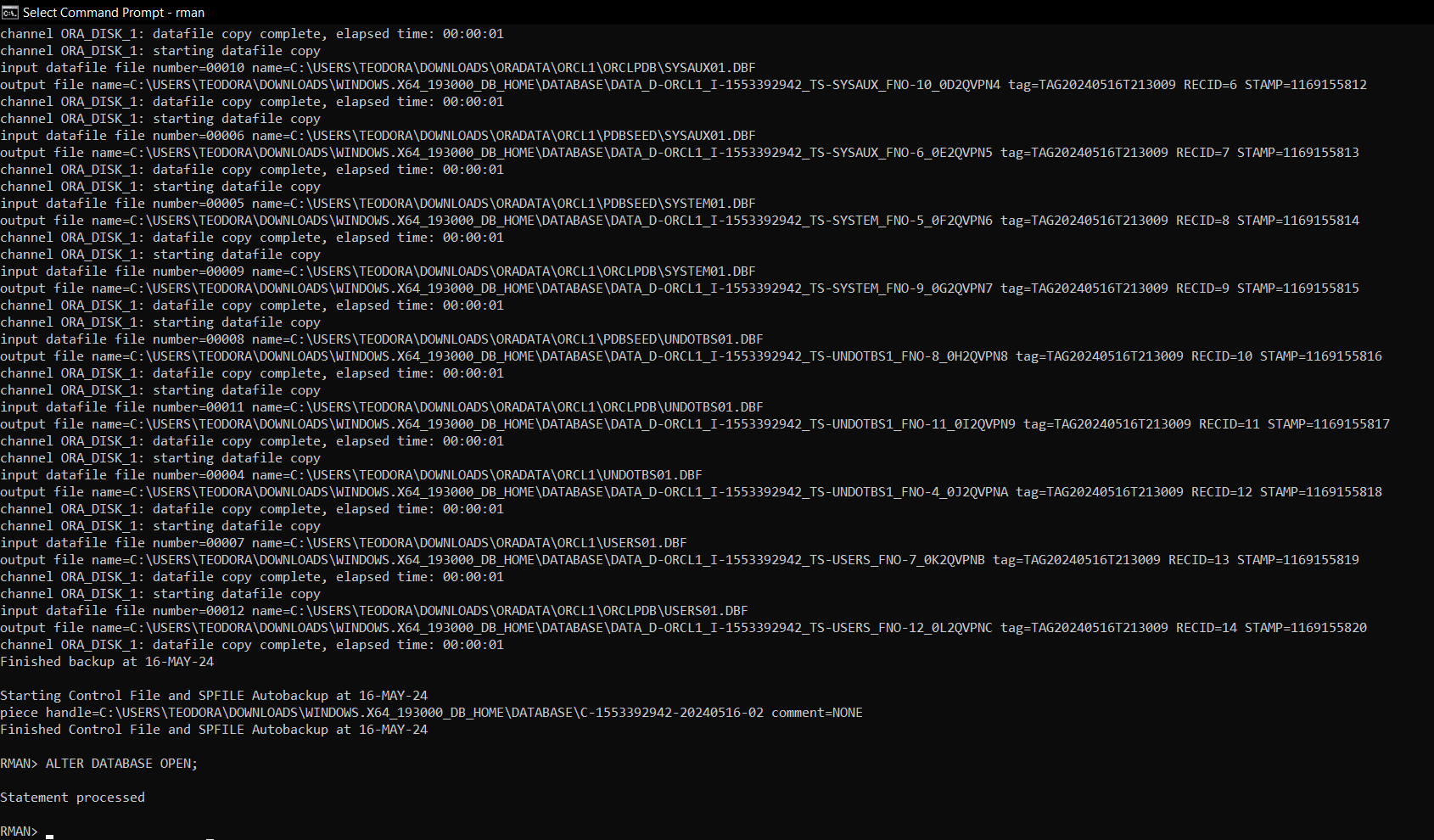
**Praktični primeri:**

**Primer 4.3**: Za pravljenje konzistentne rezervne kopije baze podataka, prvo smo isključili bazu i pokrenuli je u MOUNT režimu kako bismo osigurali konzistentnost. Zatim je korišćen RMAN za backup celokupne baze podataka I kreiranje kopije slika svih podataka u bazi. Nakon završetka backup-a, baza je ponovo otvorena za normalan rad.









### **Pravljenje inkrementalnih backup-ova**

Inkrementalni bekapovi beleže promene na nivou blokova u bazi podataka koje su napravljene nakon prethodnog inkrementalnog bekapa.

Ako navedemo BACKUP INCREMENTAL, RMAN kreira inkrementalni bekap baze podataka. Inkrementalni bekapovi su uglavnom manji i brži za izradu od potpunih bekapova baze podataka. Oporavak pomoću inkrementalnih bekapova je brži nego korišćenje samo redo logova.

Polazna tačka za strategiju inkrementalnog bekapa je inkrementalni bekap nivoa 0, koji beleži sve blokove u bazi podataka. Inkrementalni bekap nivoa 0 je sadržajno identičan potpunom bekapu, međutim, za razliku od potpunog bekapa, bekap nivoa 0 se smatra delom inkrementalne strategije bekapa.

Inkrementalni bekap nivoa 1 sadrži samo blokove koji su promenjeni nakon prethodnog inkrementalnog bekapa. Ako ne postoji bekap nivoa 0 u trenutnoj ili nadređenoj inkarnaciji baze podataka kada pokreneš bekap nivoa 1, RMAN će automatski napraviti bekap nivoa 0.

Bekap nivoa 1 može biti kumulativni inkrementalni bekap, koji uključuje sve blokove promenjene od poslednjeg bekapa nivoa 0, ili diferencijalni inkrementalni bekap, koji uključuje samo blokove promenjene od poslednjeg inkrementalnog bekapa. Inkrementalni bekapovi su po defaultu diferencijalni.

Tokom operacije vraćanja, RMAN će prvo vratiti bekap nivoa 0, a zatim automatski primeniti inkrementalne bekapove i redo logove po potrebi. Ovo će ponovo primeniti promene koje su napravljene u bazi podataka od početka bekapa.

Da bi napravili inkrementalne bekapove baze podataka potrebno je sledeće:

1. Pokrenuti RMAN i povezati se na ciljnu bazu podataka kao što je opisano u poglavlju "Pokretanje RMAN-a i povezivanje sa bazom podataka".
2. Pokreni komandu BACKUP INCREMENTAL. Sledeći primer kreira inkrementalni bekap nivoa 0 da služi kao osnova za inkrementalnu strategiju bekapovanja:

BACKUP INCREMENTAL LEVEL 0 DATABASE;

Sledeći primer kreira kumulativni inkrementalni bekap nivoa 1:

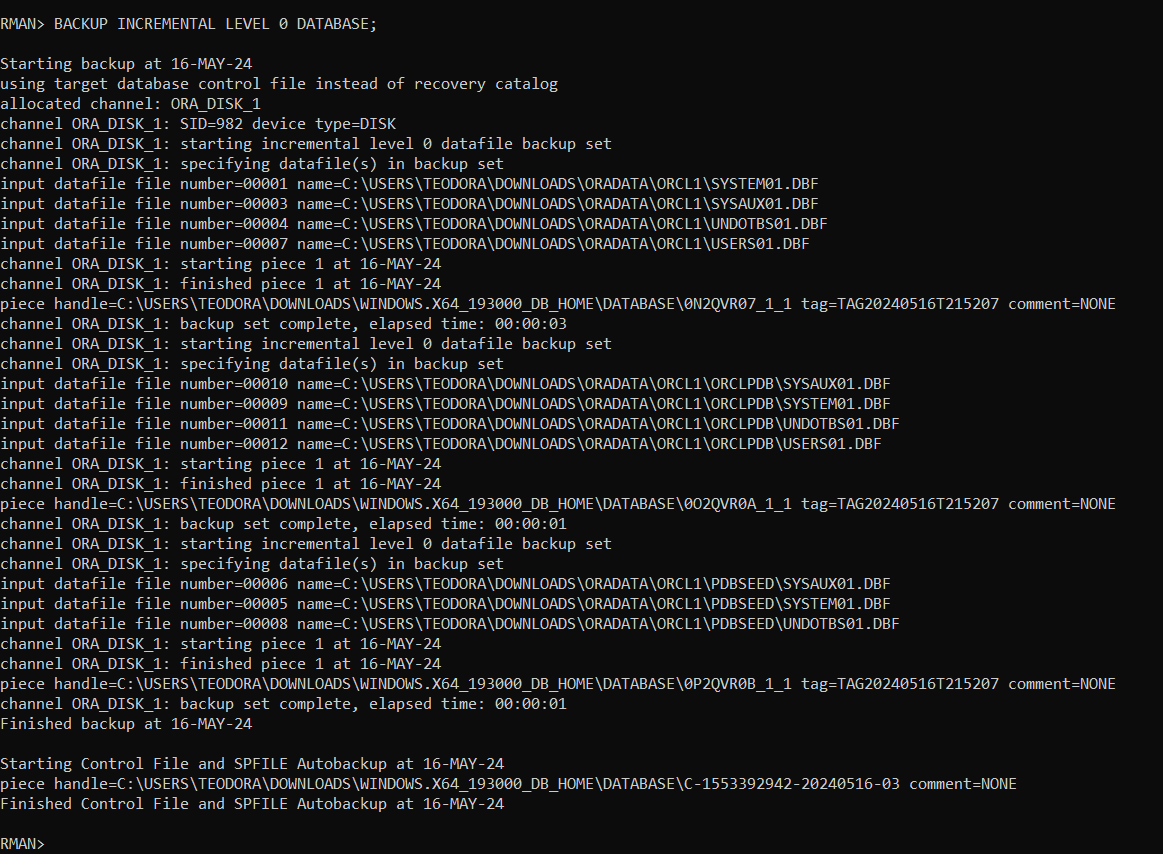
BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;

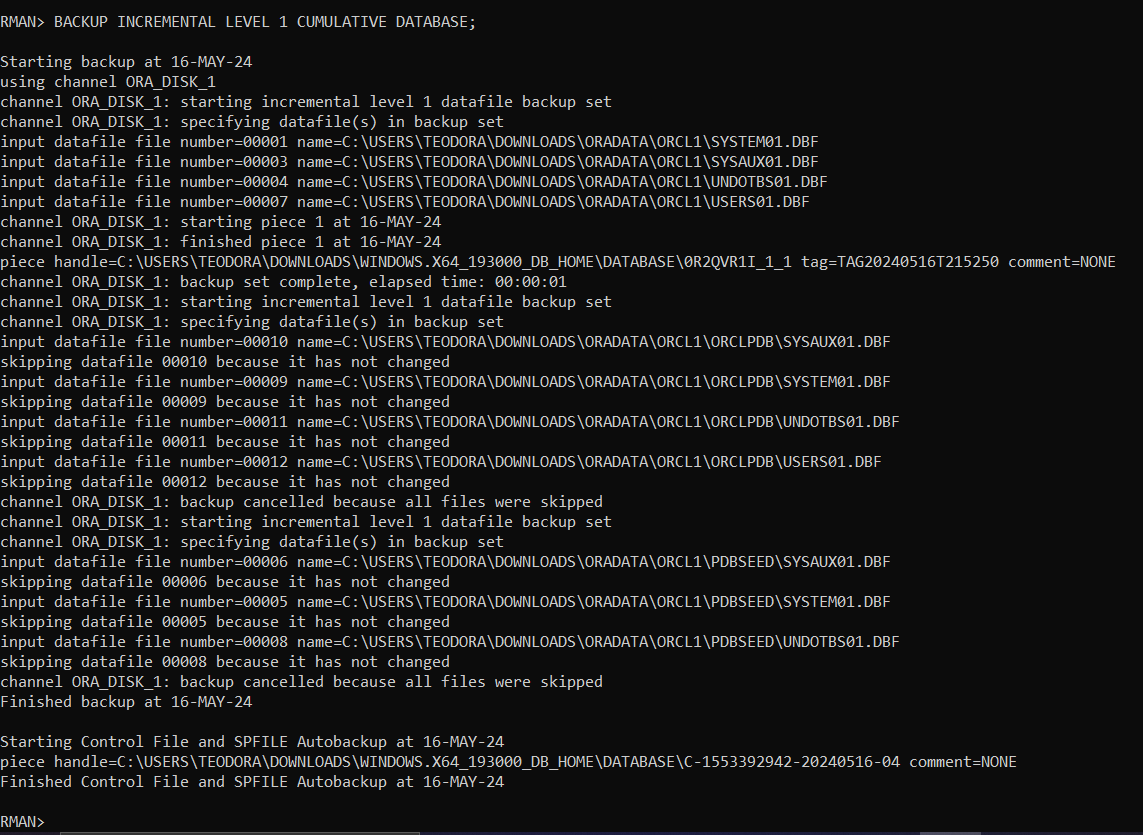
Sledeći primer kreira diferencijalni inkrementalni bekap nivoa 1:

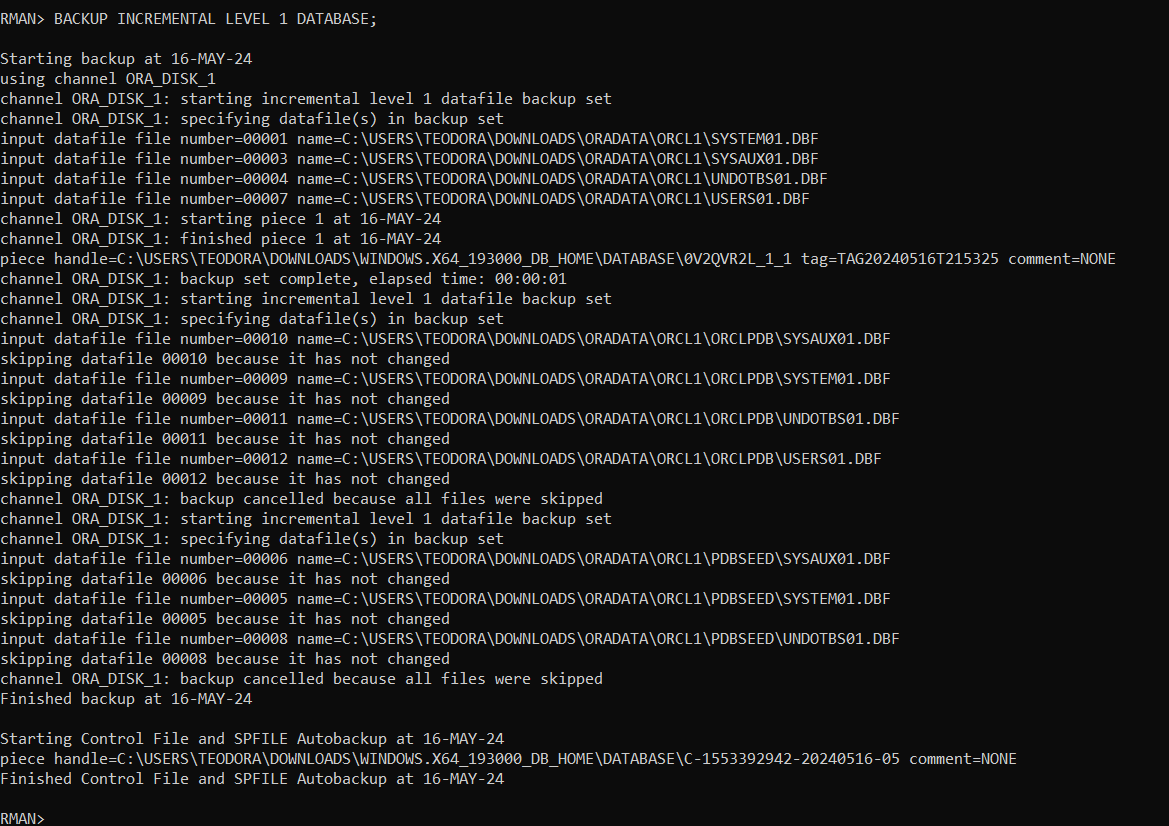
BACKUP INCREMENTAL LEVEL 1 DATABASE;

**Praktični primeri:**

**Primer 4.4**: Nakon što je baza prebačena u ARCHIVELOG režim i otvorena za pristup, izvršeni su inkrementalni backup-ovi. Prvo je kreiran inkrementalni backup nivoa 0 kao osnova za inkrementalnu strategiju bekapovanja, koristeći komandu BACKUP INCREMENTAL LEVEL 0 DATABASE. Zatim je kreiran kumulativni inkrementalni backup nivoa 1 komandom BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE. Konačno, diferencijalni inkrementalni backup nivoa 1 je izvršen koristeći komandu BACKUP INCREMENTAL LEVEL 1 DATABASE.







### **Oporavak cele baze podataka**

Treba upotrebiti komande RESTORE DATABASE i RECOVER DATABASE za obnavljanje cele baze podataka.

Neophodno je da prethodno napravimo bekape svih potrebnih fajlova. U ovom scenariju se pretpostavlja da možemo da obnovimo sve fajlove podataka na njihove originalne lokacije. Ako su originalne lokacije nedostupne, koristiti komandu SET NEWNAME.

Da bi obnovili celu bazu podataka:

1. Stavi bazu podataka u montirano stanje. Sledeći primer zaustavlja instancu baze podataka (ako je pokrenuta) i montira bazu podataka:

RMAN> STARTUP FORCE MOUNT;

1. Obnoviti bazu podataka. U sledećem primeru se koristi prekonfigurisani disk kanal da bi se obnovila baza podataka:

RMAN> RESTORE DATABASE;

3. Oporaviti bazu podataka, kao što je prikazano u sledećem primeru.

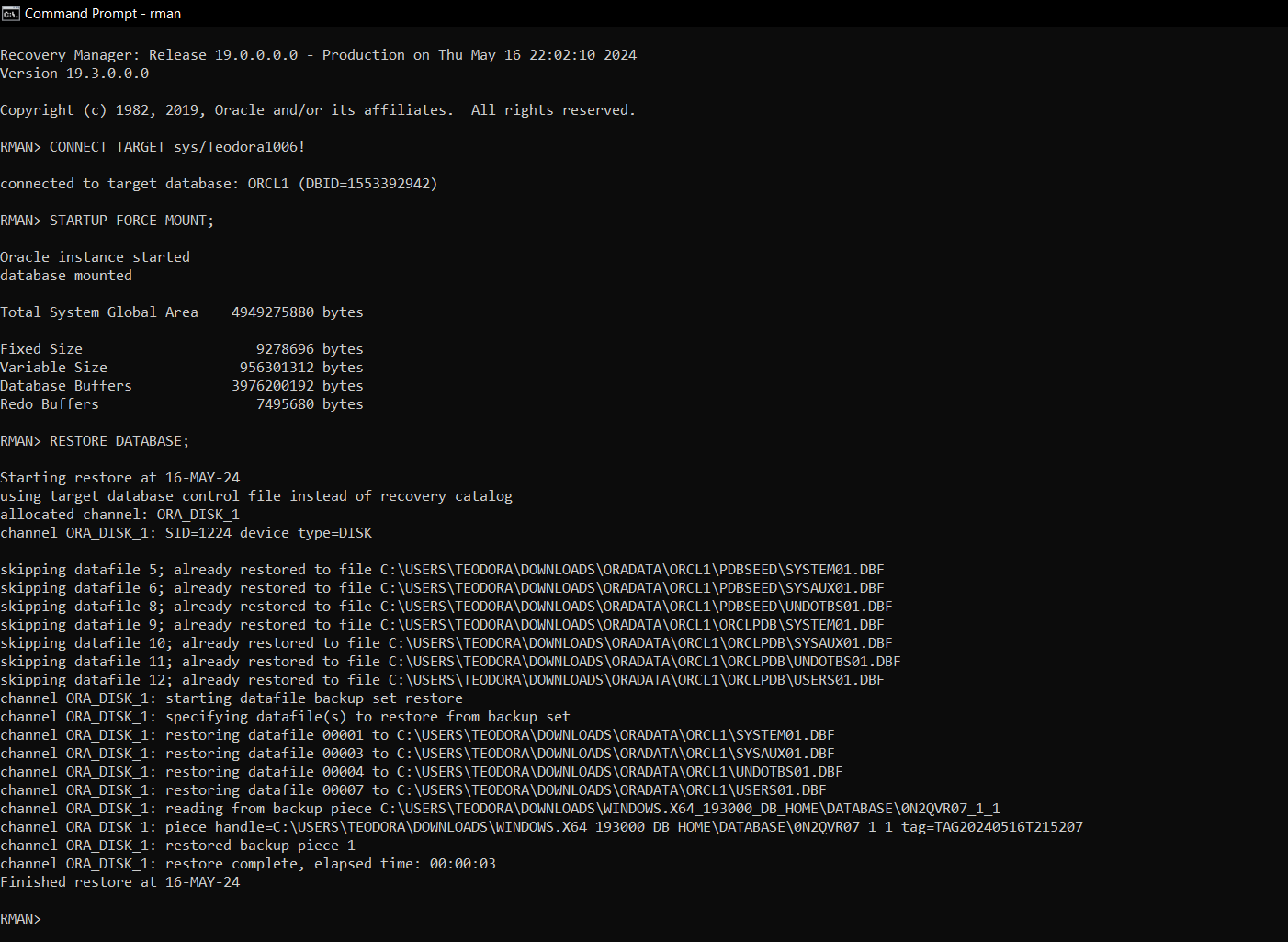
RMAN> RECOVER DATABASE;

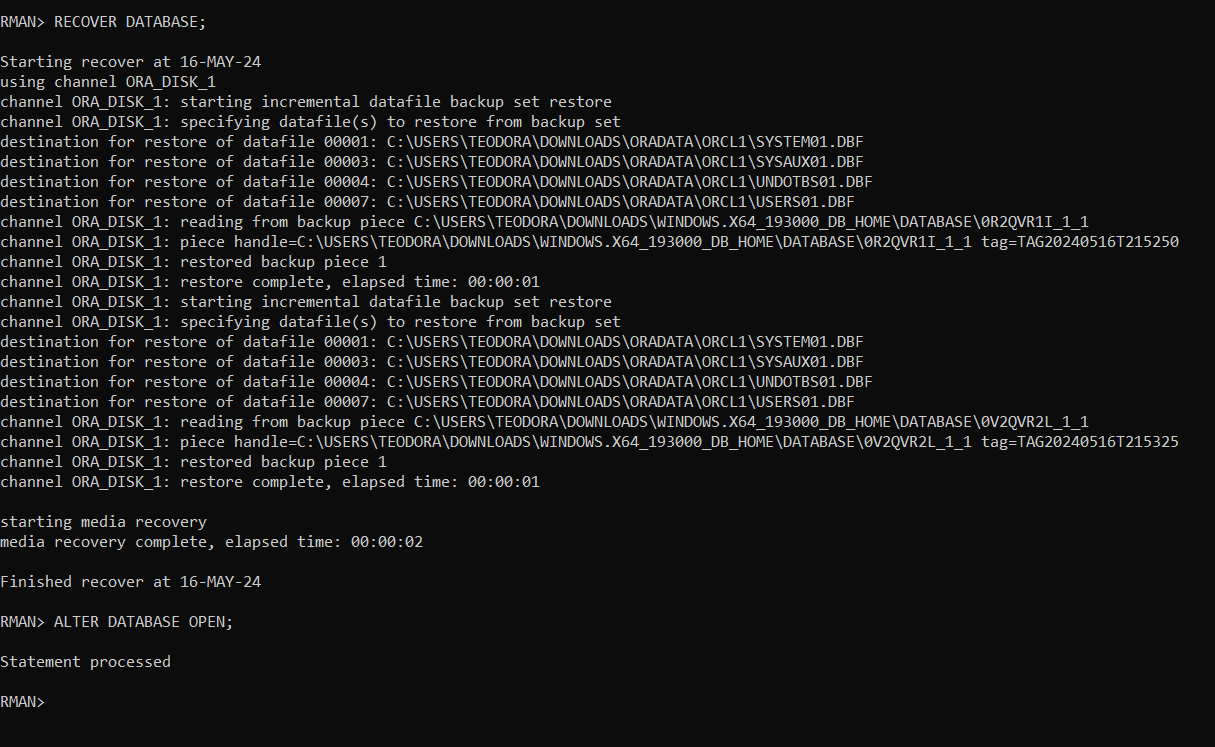
4. Otvori bazu podataka, kao što je prikazano u sledećem primeru.

RMAN> ALTER DATABASE OPEN;

**Praktični primeri:**

**Primer 4.5**: Da bi obnovili celu bazu podataka korišćenjem RMAN-a, prvo je neophodno staviti bazu u montirano stanje komandom STARTUP FORCE MOUNT, zatim koristeći RESTORE DATABASE obnavljamo sve fajlove podataka sa prethodno napravljenih bekapa. Nakon toga, primenjujemo redo logove na obnovljene fajlove koristeći RECOVER DATABASE kako bismo doveli bazu u konsistentno stanje. Ovo uključuje primenu svih transakcija koje su se desile od vremena poslednjeg bekapa do trenutka oporavka. Konačno, otvaramo bazu za normalan rad komandom ALTER DATABASE OPEN.





## **Korisnički upravljani backup-ovi i oporavak**

U nastavku rada biće opisano kako izvršiti bekup i oporavak koristeći strategiju korisnički upravljanog bekapa i oporavka, odnosno onu koja ne zavisi od RMAN-a.

Ovo poglavlje će istražiti proces kreiranja bekapa baze podataka koji se upravlja od strane korisnika, izvođenje povratka unazad i oporavka baze podataka uz korisničku kontrolu, kao i realizaciju korisnički kontrolisanog oporavka.

**Prikazivanje fajlova baze podataka pre backup-a**

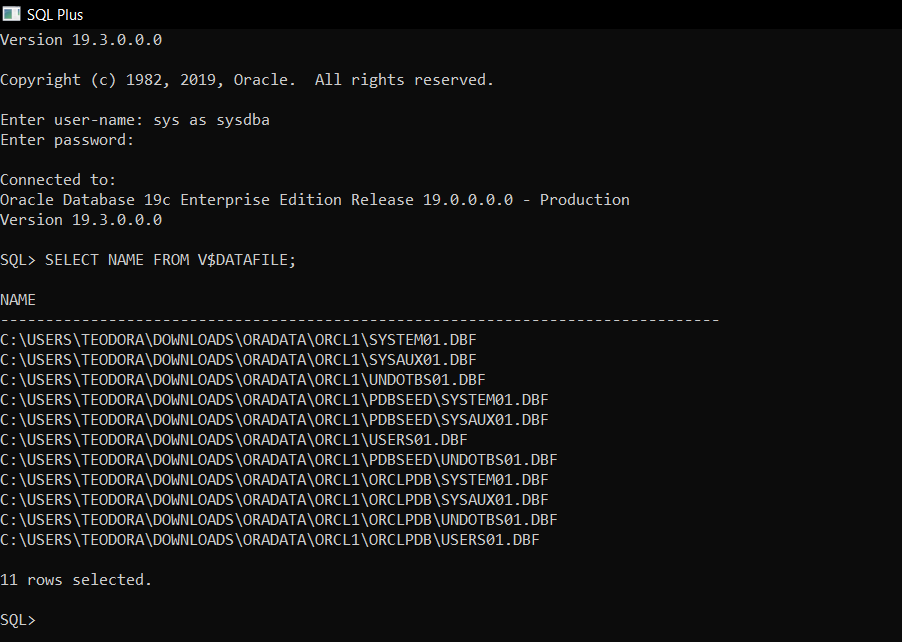
Koristiti poglede V$DATAFILE i V$CONTROLFILE za identifikaciju datoteke podataka i kontrolne datoteke za bazu podataka. Ovaj isti postupak funkcioniše bez obzira da li su ručno imenovane ove datoteke ili je dozvoljeno Oracle Managed Files da ih imenuje.

Da bi naveli datoteke podataka i kontrolne datoteke potrebno je pokrenuti na primer SQL\*Plus I izvršiti komandu $DATAFILE kako biste dobili listu datoteka podataka. Na primer:

SELECT NAME FROM V$DATAFILE;

**Praktični primeri:**

**Primer 4.6**: Pokretanje SQL\*Plus-a I prikaz svih datoteka podataka.



Takođe moguće je spojiti poglede V$TABLESPACE i V$DATAFILE kako bi dobili listu datoteka podataka zajedno sa njihovim pripadajućim tablespace-evima:

SELECT t.NAME "Tablespace", f.NAME "Data File"

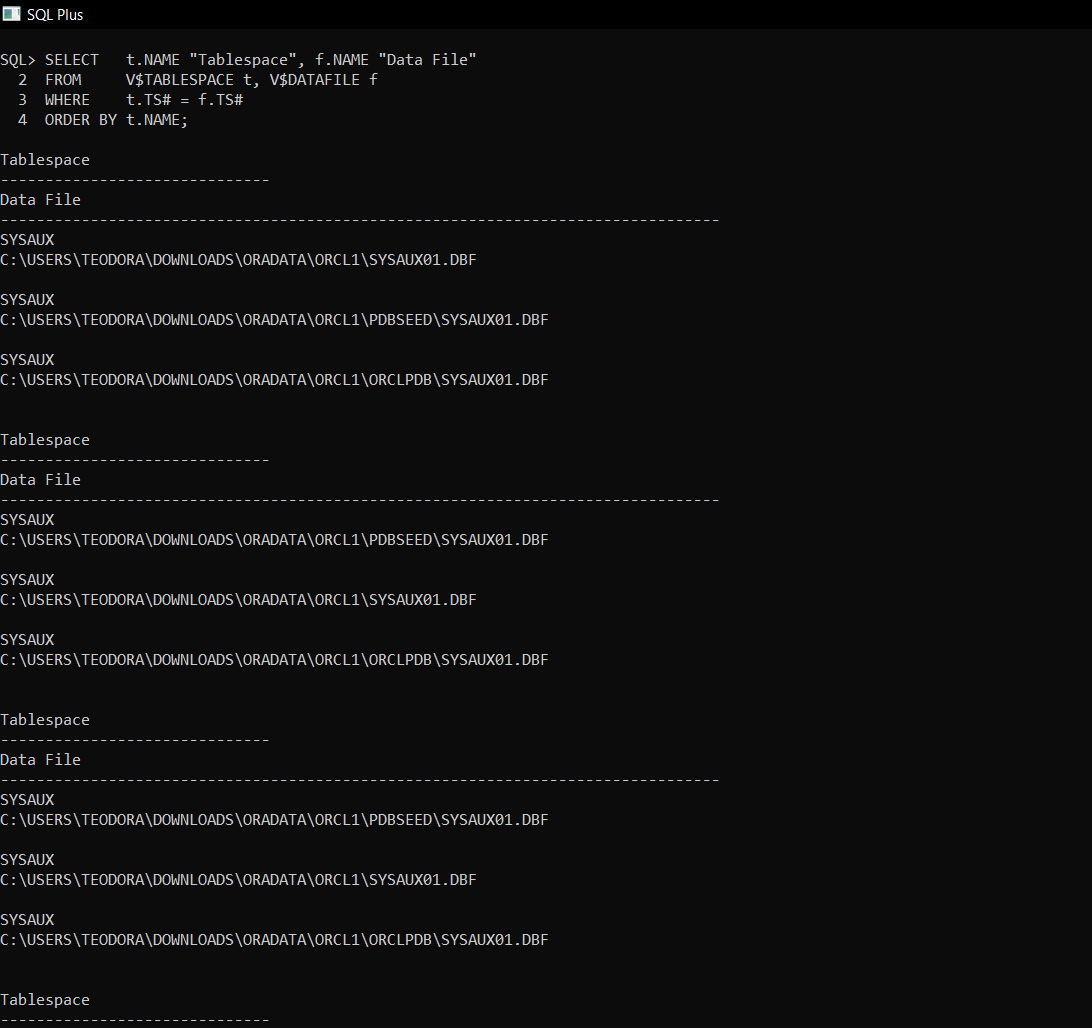
FROM V$TABLESPACE t, V$DATAFILE f

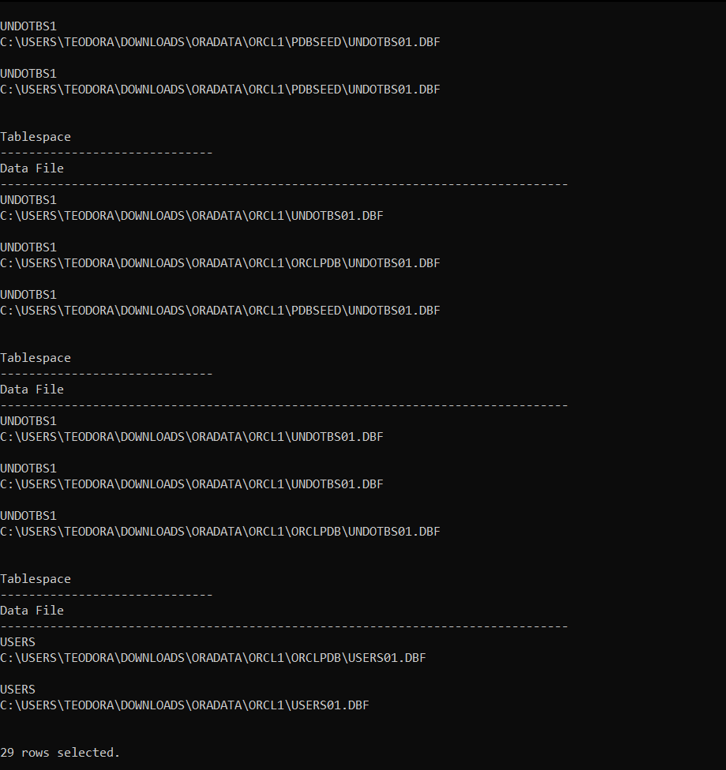
WHERE t.TS# = f.TS#

ORDER BY t.NAME;

**Praktični primeri:**

**Primer 4.7**: Prikaz liste datoteka podatakazajedno sa njihovim pripadajućim tablespace-evima.



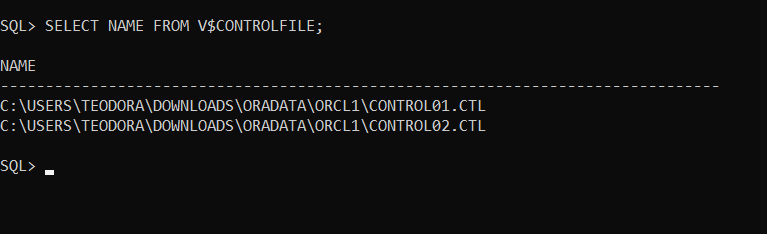


Imena trenutnih kontrolnih datoteka moguće je dobiti naredbom V$CONTROLFILE. Na primer izvršavanjem sledećeg upita:

SELECT NAME FROM V$CONTROLFILE;

**Praktični primeri:**

**Primer 4.8**: Prikaz trenutnih kontrolnih datoteka.



Dovoljno je bekupovati samo jednu kopiju višestruko umrežene kontrolne datoteke. Pri bekapovanju kontrolne datoteke putem SQL naredbe ALTER DATABASE BACKUP CONTROLFILE TO ‘filename’, preporučuje se čuvanje liste svih datoteka podataka i online redo log datoteka zajedno sa kontrolnom datotekom. To je zbog toga što struktura baze podataka u trenutku bekapa kontrolne datoteke može biti različita od strukture u trenutku kada je bekup kreiran. Čuvanje liste datoteka koje su bile zabeležene u bekapu kontrolne datoteke može biti korisno za olakšavanje postupka oporavka.

**Utvrđivanje statusa datoteke podataka za backup-e online tablespace-a**

Kako bi proverili da li je datoteka podataka deo trenutnog online bekapa tablespace-a, može se izvršiti upit nad pogledom V$BACKUP.

Ovaj pogled je koristan samo za korisnički upravljane online bekape tablespace-a, jer ni RMAN bekapi ni bekapi tablespace-a koje su offline ne zahtevaju da datoteke podataka tabele budu u ***backup modu***. Neke korisnički upravljane procedure bekapa zahtevaju da stavite tablespace u backup mod kako biste se zaštitili od mogućnosti frakturnog bloka. Međutim, ažuriranja baze podataka stvaraju više od uobičajene količine redo zapisa u backup modu.

Pogled V$BACKUP je najkorisniji kada je baza podataka otvorena. Takođe je koristan odmah nakon pada instance jer pokazuje status bekapa datoteka u trenutku pada. Treba koristiti ove informacije kako bi utvrdili da li je neki tablespace ostao u backup modu.

V$BACKUP nije koristan ako kontrolni fajl koji se trenutno koristi predstavlja vraćeni bekup ili novi kontrolni fajl kreiran nakon što se dogodio medijski kvar. Vraćeni ili ponovno kreirani kontrolni fajl ne sadrži informacije koje su potrebne bazi podataka da tačno popuni pogled V$BACKUP. Takođe, ukoliko je vraćen bekup fajla, STATUS ovog fajla u V$BACKUP odražava status bekapa starije verzije fajla, a ne najnovije verzije. Dakle, ovaj pogled može sadržati obmanjujuće podatke o vraćenim fajlovima.

Na primer, sledeći upit prikazuje koje datoteke podataka su trenutno uključene u tablespace koji je postavljen u backup mod:

SELECT t.name AS "TB\_NAME", d.file# as "DF#", d.name AS "DF\_NAME", b.status

FROM V$DATAFILE d, V$TABLESPACE t, V$BACKUP b

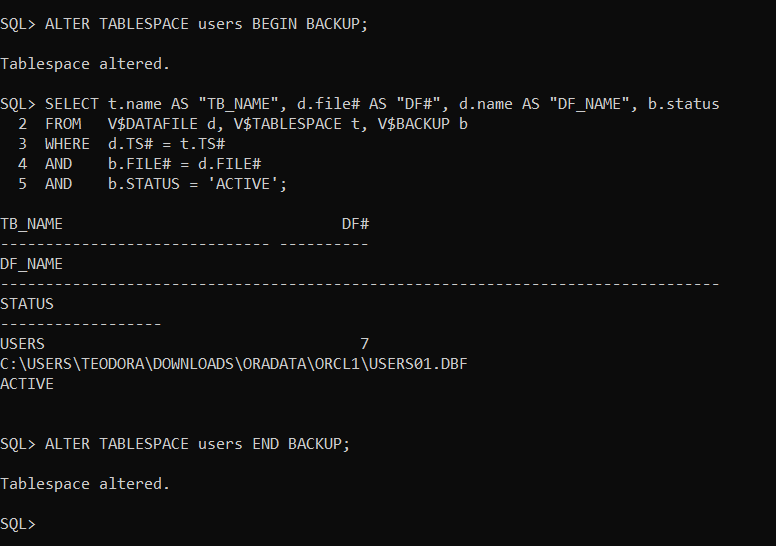
WHERE d.TS#=t.TS#

AND b.FILE#=d.FILE#

AND b.STATUS='ACTIVE';

**Praktični primeri:**

**Primer 4.9**: Prikaz datoteka podataka koje su trenutno uključene u tablespace koji je postavljen u backup mod.



U STATUS koloni, NOT ACTIVE označava da datoteka trenutno nije u backup modu (tj. da nije izvršena ALTER TABLESPACE ... BEGIN BACKUP ili ALTER DATABASE BEGIN BACKUP naredba), dok ACTIVE označava da je datoteka trenutno u backup modu.

### **Pravljenje backup-ova cele baze podataka kojima upravlja korisnik**

Moguće je napraviti konzistentan bekap cele baze podataka svih fajlova u bazi podataka nakon što je baza podataka isključena sa NORMAL, IMMEDIATE ili TRANSACTIONAL opcijama. Bekap cele baze podataka koji je napravljen dok je baza otvorena ili nakon neuspeha instance ili komande SHUTDOWN ABORT je nekonzistentan. U takvim slučajevima, fajlovi su nekonzistentni sa SCN-om baze podataka.

Moguće je napraviti bekapan cele baze podataka ako je baza u režimu ili ARCHIVELOG ili NOARCHIVELOG. Međutim, ako se baza podataka pokreće u režimu NOARCHIVELOG, tada bekap mora biti konzistentan tj. mora se čistim zaustavljanjem isključiti baza pre bekapa.

Skup bekup fajlova koji rezultira iz konzistentnog bekapa cele baze podataka je konzistentan jer su svi fajlovi checkpointovani na isti SCN. Moguće je vratiti konzistentni bekapa baze podataka bez daljeg oporavka. Nakon vraćanja bekup fajlova, moguće je izvršiti dodatne korake oporavka kako bi oporavili bazu podataka na aktuelnije vreme ako je baza u režimu ARCHIVELOG. Takođe, moguće je napraviti nekonzistentne bekape cele baze podataka ako je baza u režimu ARCHIVELOG.

Kontrolni fajlovi igraju ključnu ulogu u obnavljanju i oporavku baze podataka. Za baze koje rade u režimu ARCHIVELOG, Oracle preporučuje bekapovanje kontrolnih fajlova pomoću ALTER DATABASE BACKUP CONTROLFILE TO 'namefile' naredbe.

Koraci za pravljenje konzistentnih bekapa cele baze podataka:

1. Ako je baza otvorena, koristiti SQL\*Plus da bi isključili bazu podataka sa NORMAL, IMMEDIATE ili TRANSACTIONAL opcijama.
2. Koristiti operativni sistem za pravljenje bekapa svih data fajlova i kontrolnih fajlova koji su specificirani parametrom CONTROL\_FILES u inicijalnom fajlu parametara. Takođe, bekapovati inicijalni fajl parametara i druge inicijalne fajlove proizvoda Oracle. Da bi pronašli ove fajlove, treba pretražiti \*.ora počevši od Oracle home direktorijuma i rekurzivno pretražiti sve njegove poddirektorijume. Na primer, mogu se bekupovati data fajlovi, kontrolni fajlovi i arhivirani logovi na /disk2/backup na sledeći način:

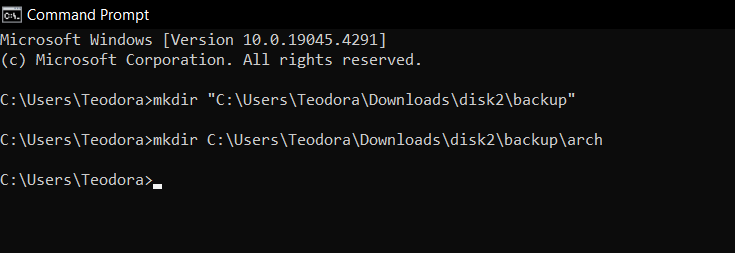
% cp $ORACLE\_HOME/oradata/trgt/\*.dbf /disk2/backup

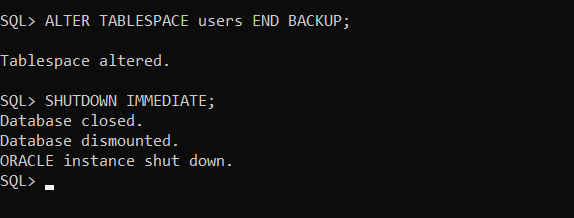
% cp $ORACLE\_HOME/oradata/trgt/arch/\* /disk2/backup/arch

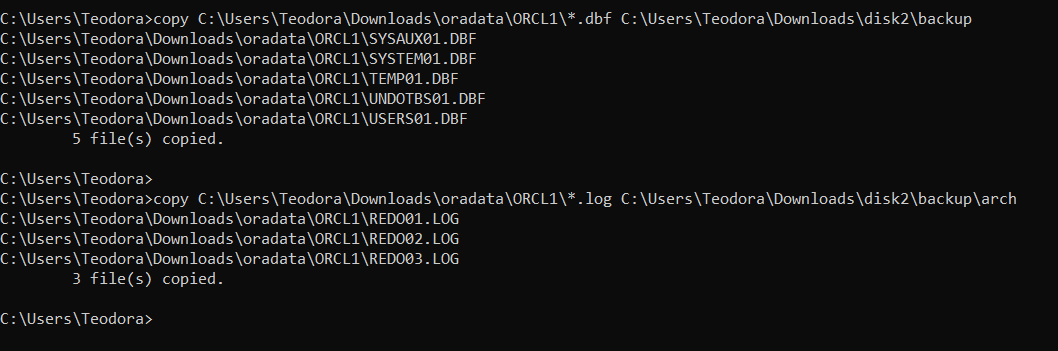
1. Pokrenuti bazu podataka sa STARTUP komandom u SQL\*Plus-u.

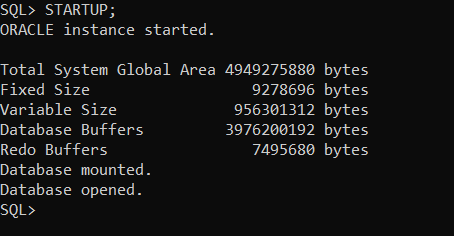
**Praktični primeri:**

**Primer 4.10**: Prvo su kreirani direktorijumi za čuvanje bekapa: C:\Users\Teodora\Downloads\disk2\backup za data fajlove i C:\Users\Teodora\Downloads\disk2\backup\arch za arhivirane redo logove. Zatim je baza podataka isključena korišćenjem SHUTDOWN IMMEDIATE komande u SQL\*Plus-u. Nakon toga, datoteke .dbf su kopirane u direktorijum za bekape, dok su .log datoteke kopirane u poddirektorijum za arhivirane redo logove. Na kraju, baza podataka je ponovno pokrenuta korišćenjem STARTUP komande.

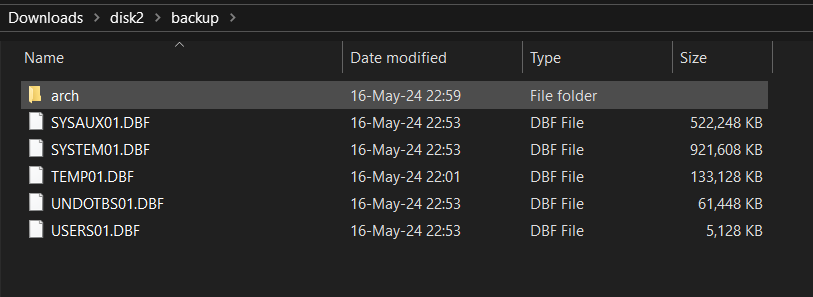


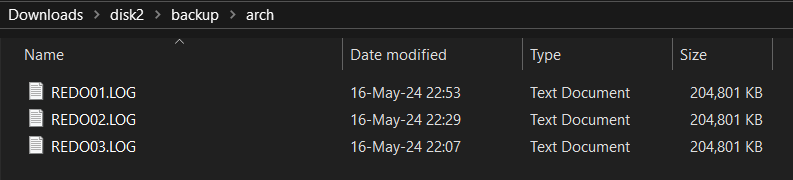






**Rezultat:**





### **Pravljenje korisnički upravljanih backup-ova tablespace-ova i datoteka podataka**

Tehnika pravljenja korisnički upravljanih backup-ova tablespace-ova i datoteka podataka zavisi od toga da li su fajlovi offline ili online.

U nastavku biće obrađeno pravljenje korisnički upravljanih backup-ova *offline* tablespace-ova i datoteka podataka, kao i pravljenje korisnički upravljanih backup-ova *online* tablespace-ova i datoteka podataka.

**Pravljenje korisnički upravljanih backup-ova *offline* tablespace-eva**

Nije moguće uzeti offline SYSTEM tablespace ili tablespace sa aktivnim undo segmentima. Sledeća tehnika ne može biti korišćena za takve tablespaces-ove.

Ako pretpostavimo da je tabela u tablespace-u Primary, a njen indeks je u tablespace-u Index. Uzimanje tablespace-a Index offline dok se tablespace Primary ostavlja online može izazvati greške prilikom izvršavanja jezika za manipulaciju podacima (DML) nad indeksiranim tabelama smeštenim u Primary. Problem se javlja samo kada optimizator bira pristupni metod koji mora da pristupi indeksima u tablespace-u Index.

Da bi napravili bekape offline tablespace-ova:

1. Pre početka bekapa tablespace-a, važno je identifikovati datoteke podataka koje pripadaju tom tablespace-u. To se može uraditi putem SQL upita nad DBA\_DATA\_FILES pregledom. Na primer, ako želimo da napravimo bekup tablespace-a users, možemo koristiti sledeću SQL naredbu u SQL\*Plus-u:

SQL> SELECT TABLESPACE\_NAME, FILE\_NAME

FROM SYS.DBA\_DATA\_FILES

WHERE TABLESPACE\_NAME = 'USERS';

TABLESPACE\_NAME FILE\_NAME

------------------------------- -------------------------------

USERS /oracle/oradata/trgt/users01.dbf

U ovom primeru, /oracle/oradata/trgt/users01.dbf je potpuno određeno ime datoteke koje odgovara datoteci podataka u tablespace-u users.

1. Treba isključiti tablespace koristeći normalni prioritet ako je moguće, jer to garantuje da će naknadno moći da se ponovo uključi tablespace bez potrebe za oporavkom. Na primer:

SQL> ALTER TABLESPACE users OFFLINE NORMAL;

1. Treba napraviti bekap offline datoteka podataka. Na primer:

% cp /oracle/oradata/trgt/users01.dbf

/d2/users01\_'date "+%m\_%d\_%y"'.dbf

1. Treba postaviti tablespace-ove na online. Na primer:

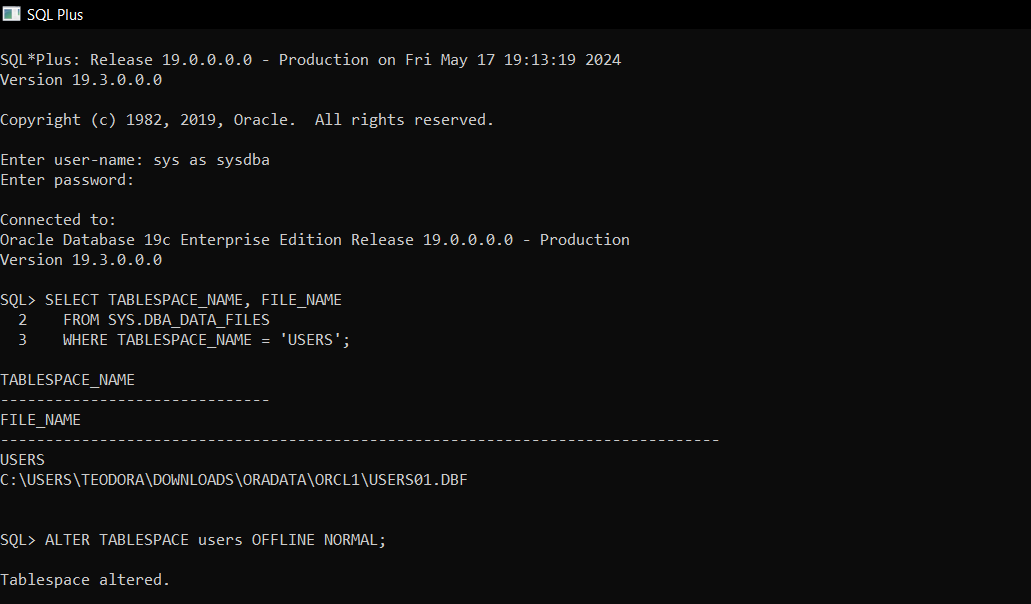
SQL> ALTER TABLESPACE users ONLINE;

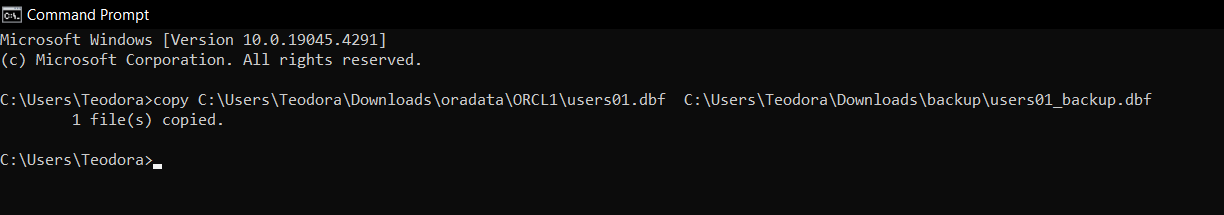
1. Zatim treba arhivirati nearhivirane redo logove kako bi redo potreban za oporavak bekapa tablespace-a bio arhiviran. Na primer:

SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;

**Praktični primeri:**

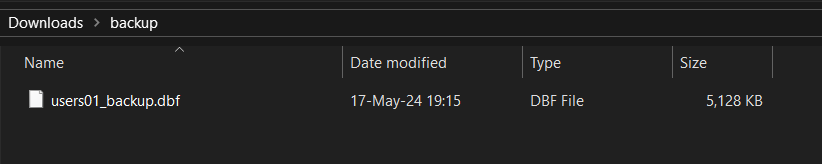
**Primer 4.11**: Pronađene su datoteke tablespace-a USERS i tablespace je isključen da bi se omogućio siguran backup. Fajl users01.dbf je kopiran u direktorijum backup sa novim imenom users01\_backup.dbf. Nakon toga, tablespace je ponovo uključen, a trenutni redo logovi su arhivirani radi sigurnosti podataka.







**Rezultat:**



**Pravljenje korisnički upravljanih backup-ova *online* tablespace-ova**

Moguće je bekupovati sve ili samo određene datoteke podataka online tablespace-a dok je baza podataka otvorena. Postupak se razlikuje u zavisnosti od toga da li je online tablespace za *čitanje/pisanje* ili samo za *čitanje*.

Da bi napravili bekape datoteka upravljanih od strane korisnika kada je tablespace online i baza podataka otvorena, moraju se staviti tablespace-ovi za čitanje/pisanje u backup mod. Naredba ALTER TABLESPACE ... BEGIN BACKUP stavlja tablespace u backup mod. U modu bekapa, baza podataka kopira ceo promenjeni blok podataka u redo tok. Nakon što izuzmete tablespace iz moda bekapa pomoću naredbe ALTER TABLESPACE ... END BACKUP ili ALTER DATABASE END BACKUP, baza podataka unapređuje SCN tačku podataka datoteke na trenutnu SCN tačku baze podataka.

Prilikom vraćanja datoteke podataka bekapovane na ovaj način, baza podataka traži odgovarajući set redo log datoteka koje treba primeniti ako je potreban oporavak. Redo logovi sadrže sve promene potrebne za oporavak datoteka podataka i njihovo ujednačavanje.

Da bi bekapovali online tablespace-ove *za čitanje/pisanje* u otvorenoj bazi podataka potrebno je sledeće:

1. Pre početka bekapa tablespace-a, treba koristiti DBA\_DATA\_FILES rečnik podataka pogleda da identifikujete sve datoteke podataka u tablespace-u. Na primer, pretpostavimo da želimo da bekapujemo tablespace users:

SQL> SELECT TABLESPACE\_NAME, FILE\_NAME

FROM SYS.DBA\_DATA\_FILES

WHERE TABLESPACE\_NAME = 'USERS';

TABLESPACE\_NAME FILE\_NAME

------------------------------- --------------------

USERS /oracle/oradata/trgt/users01.dbf

USERS /oracle/oradata/trgt/users02.dbf

1. Treba označiti početak bekapa online tablespace-a. Na primer, sledeća naredba označava početak online bekapa za tablespace users:

SQL> ALTER TABLESPACE users BEGIN BACKUP;

1. Treba napraviti bekap online datoteka podataka online tablespace-a korišćenjem operativnih sistema. Na primer, korisnici Linux-a i UNIX-a mogu uneti:

% cp /oracle/oradata/trgt/users01.dbf

/d2/users01\_'date "+%m\_%d\_%y"'.dbf

% cp /oracle/oradata/trgt/users02.dbf

/d2/users02\_'date "+%m\_%d\_%y"'.dbf

1. Nakon što se bekapuju datoteke podataka online tablespace-a, treba pokrenuti SQL naredbu ALTER TABLESPACE sa opcijom END BACKUP. Na primer, sledeća naredba završava online bekap tablespace-a users:

SQL> ALTER TABLESPACE users END BACKUP;

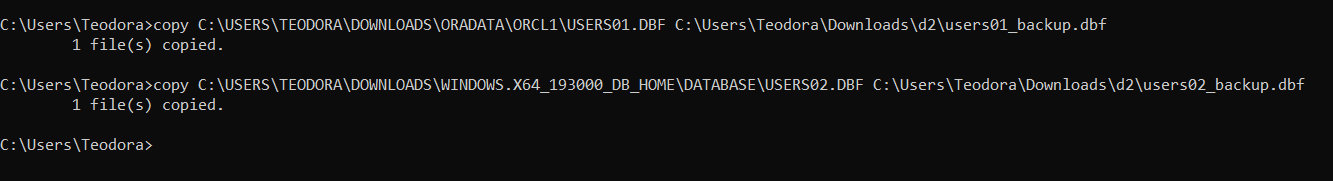
1. Treba arhivirati nearhivirane redo logove kako bi se redo koji je potreban za oporavak bekapovanog tablespace-a arhivirao. Na primer:

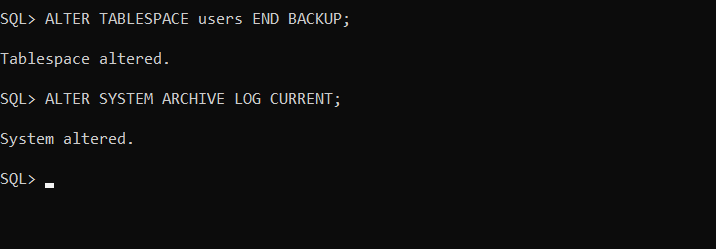
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;

**Praktični primeri:**

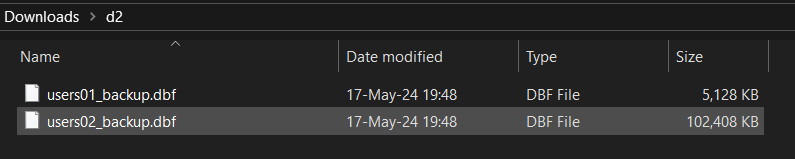
**Primer 4.12**: Izvršen je SQL upit koji je identifikovao datoteke podataka u tablespace-u USERS. Zatim je započet online bekape tablespace-a USERS. Kopirane su datoteke podataka tablespace-a USERS u ciljni direktorijum kako bi se napravile sigurnosne kopije. Konačno, trenutni redo logovi su arhivirani radi obezbeđivanja potrebnih podataka za oporavak u budućnosti.







**Rezultat:**



Kada se bekupuje online *read-only* tablespace, jednostavno se mogu bekapovati online datoteke podataka. Nije potrebno stavljati tablespace u backup mod jer baza podataka ne dozvoljava promene u data fajlovima.

Ako set read-only tablespace-ova čini samostalan skup, osim bekapovanja tablespace-ova pomoću operativnog sistema, takođe je moguće izvesti metapodatke tablespace-a sa funkcionalnošću prenosivog tablespace-a. U slučaju medijske greške ili korisničke greške (kao što je slučajno brisanje tabele u read-only tablespace-u), moguće je preneti tablespace nazad u bazu podataka.

Da bi bekapovali online read-only tablespace-ove u otvorenoj bazi podataka potrebno je sledeće:

1. Izvršiti upit nad DBA\_TABLESPACES pogledom kako bi odredili koje su tablespace-ovi samo za čitanje. Na primer, pokrenuti ovaj upit:

SELECT TABLESPACE\_NAME, STATUS

FROM DBA\_TABLESPACES

WHERE STATUS = 'READ ONLY';

1. Pre nego što se započne bekapovanje read-only tablespace-a, identifikujemo sve datoteke podataka tablespace-a tako što ćete izvršiti upit nad DBA\_DATA\_FILES pogledom. Na primer, pretpostavimo da se bekapuje tablespace history :

SELECT TABLESPACE\_NAME, FILE\_NAME

FROM SYS.DBA\_DATA\_FILES

WHERE TABLESPACE\_NAME = 'HISTORY';

TABLESPACE\_NAME FILE\_NAME

------------------------------- --------------------

HISTORY /oracle/oradata/trgt/history01.dbf

HISTORY /oracle/oradata/trgt/history02.dbf

1. Bekapujti online datoteke podataka read-only tablespace-a koristeći operativne sistemskue komande. Nije potrebno offline-ovati tablespace ili staviti tablespace u backup mod jer korisnici automatski ne mogu vršiti promene na read-only tablespace-u. Na primer:

% cp $ORACLE\_HOME/oradata/trgt/history\*.dbf /disk2/backup/

1. Po potrebi, izvesti metapodatke iz read-only tablespace-a. Korišćenjem mogućnosti transportabilnog tablespace-a, možemo brzo vratiti datoteke podataka i uvesti metapodatke u slučaju medijskog kvara ili korisničke greške. Na primer, izvezimo metapodatke za tablespace history na sledeći način:

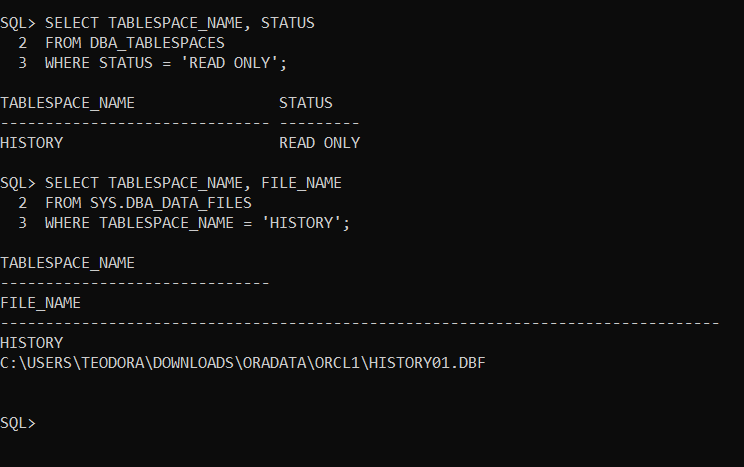
% expdp DIRECTORY=dpump\_dir1

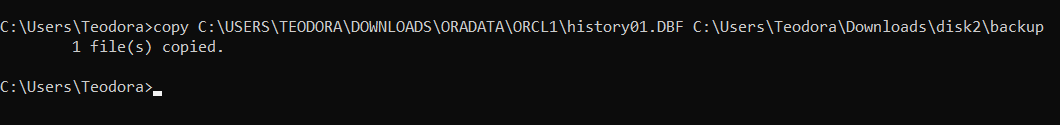
DUMPFILE=hs.dmp TRANSPORT\_TABLESPACES=history

LOGFILE=tts.log

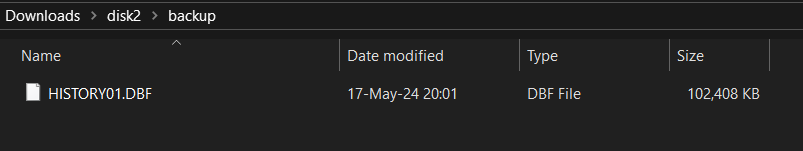
**Praktični primeri:**

**Primer 4.13**: Izvršen je SQL upit koji prikazuje imena i status svih tablespace-ova označenih kao "READ ONLY", pružajući uvid u tablespace-ove koji su namenjeni samo za čitanje. Zatim je izvršen upit koji identifikuje ime i putanju datoteke podataka za tablespace HISTORY, omogućavajući identifikaciju datoteka podataka pripadajućih tom tablespace-u. Na kraju, korišćenjem komande copy, datoteka podataka history01.DBF je kopirana sa izvorne lokacije na ciljnu lokaciju, čime je napravljena rezervna kopija podataka tablespace-a HISTORY.





**Rezultat:**



### **Pravljenje korisnički upravljanih backup-ova kontrolne datoteke**

Obavićemo strukturnu izmenu u bazi podataka koja radi u ARCHIVELOG režimu, nakon čega ćemo izvršiti backup kontrolnog fajla baze podataka. Za izradu backup-a kontrolnog fajla, potrebna je imati sistemsku privilegija ALTER DATABASE.

U nastavku biće objašnjeno pravljenje backup-a kontrolnog fajla u *binarnom* formatu i pravljenje backup-a kontrolnog fajla u formatu *traga* (trace).

**Pravljenje backup-a kontrolnog fajla u *binarnom* formatu**

Osnovna metoda za pravljenje rezervne kopije kontrolnog fajla je korišćenje SQL naredbe za generisanje binarnog fajla. Binarna rezervna kopija je poželjnija od rezervne kopije u obliku traga jer sadrži dodatne informacije kao što su istorija arhiviranih logova, opsezi offline statusa za read-only i offline tabelarnih prostora, i setovi i kopije rezervnih podataka (ukoliko se koristi RMAN). Ako je COMPATIBLE 10.2 ili veći, binarne rezervne kopije kontrolnog fajla uključuju i unose privremenih fajlova.

Za pravljenje rezervne kopije kontrolnog fajla nakon strukturne promene potrebno je sledeće:

1. Napraviti željenu promenu u bazi podataka. Na primer, možemo napraviti tablespace:

CREATE TABLESPACE tbs\_1 DATAFILE 'file\_1.f' SIZE 10M;

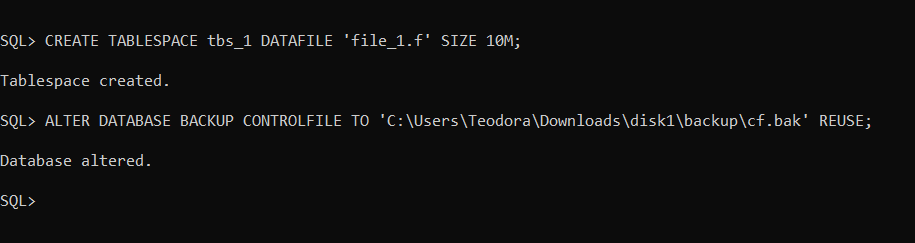
1. Napraviti rezervnu kopiju kontrolnog fajla baze podataka, specificirajući naziv fajla za izlazni binarni fajl. U sledećem primeru, vrši se rezervna kopija kontrolnog fajla na lokaciju /disk1/backup/cf.bak:

ALTER DATABASE BACKUP CONTROLFILE TO '/disk1/backup/cf.bak' REUSE;

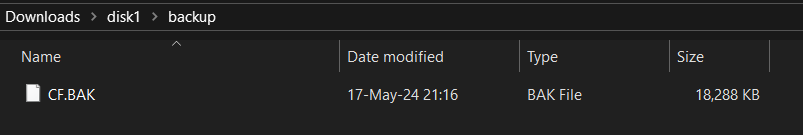
Specificirati REUSE kako bi omogućili da novi kontrolni fajl prepiše onaj koji trenutno postoji.

**Praktični primeri:**

**Primer 4.14**: Prva naredba kreira novi tablespace nazvan tbs\_1 sa jednom binarnom datotekom podataka nazvanom file\_1.f. Druga naredba izvršava backup kontrolnog binarnog fajla baze podataka i smešta ga na lokaciju "C:\Users\Teodora\Downloads\disk1\backup\cf.bak", koristeći opciju REUSE za prepisivanje postojećeg kontrolnog fajla ako postoji.



**Rezultat:**



**Pravljenje backup-a kontrolnog fajla u *trace* formatu**

Možemo napraviti rezervnu kopiju kontrolnog fajla u tekstualni fajl koji sadrži CREATE CONTROLFILE naredbu. Takođe možemo izmeniti trace fajl kako bi kreirali skriptu koja će kreirati novi kontrolni fajl na osnovu kontrolnog fajla koji je bio aktuelan kada smo kreirali trace fajl.

Ako ne navedemo opciju RESETLOGS ili NORESETLOGS u SQL naredbi, tada rezultujući trace fajl sadrži verzije kontrolnog fajla za obe opcije RESETLOGS i NORESETLOGS. Unosi privremenih fajlova su uključeni u izlaz korišćenjem ALTER TABLESPACE ... ADD TEMPFILE naredbi.

Da bi izbegli oporavak offline normalnih ili samo za čitanje tablespace-eva, treba ih izmeniti iz CREATE CONTROLFILE naredbe. Kada otvorimo bazu podataka sa ponovo kreiranim kontrolnim fajlom, baza podataka označava ove izostavljene fajlove kao MISSING. Možemo pokrenuti ALTER DATABASE RENAME FILE naredbu da bi ih preimenovali u njihove originalne nazive fajlova.

Trace fajl koji sadrži CREATE CONTROLFILE naredbu se čuva u poddirektorijumu koji određuje DIAGNOSTIC\_DEST inicijalni parametar. Možemo pogledati bazu podataka u alert logu kako bi pronašli ime i lokaciju trace fajla u koji je zapisana CREATE CONTROLFILE naredba.

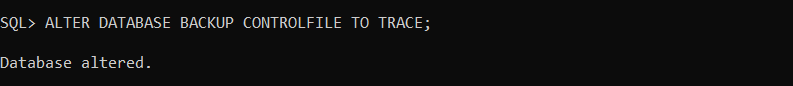
Da bi napravili rezervnu kopiju kontrolnog fajla u formatu trace potrebno je sledeće:

1. Montirati ili otvoriti bazu podataka.
2. Izvršiti sledeću SQL naredbu:

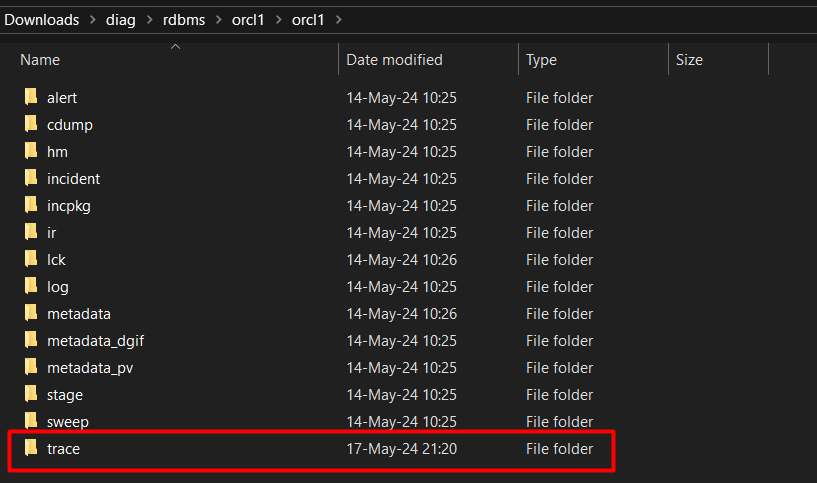
ALTER DATABASE BACKUP CONTROLFILE TO TRACE;

**Praktični primeri:**

**Primer 4.15**: SQL naredbom, ALTER DATABASE BACKUP CONTROLFILE TO TRACE, izvršena je akcija kreiranja rezervne kopije kontrolnog fajla u formatu trace. Kada je naredba izvršena uspešno, generisan je trace fajl koji se može koristiti za obnovu kontrolnog fajla baze podataka ukoliko je potrebno.



**Rezultat:**



### **Pravljenje korisnički upravljanih backup-ova arhiviranih redo logova**

Kako bi uštedeli prostor na disku u primarnoj lokaciji za arhiviranje, želimo da rezervišemo arhivirane dnevničke zapise na traku ili na alternativnu lokaciju na disku. Ako arhiviramo na više lokacija, onda rezervišimo samo jednu kopiju svakog broja sekvence loga.

Da bi rezervisali arhivirane dnevne zapise potrebno je sledeće:

1. Da bi odredili koje arhivirane dnevne datoteke je baza podataka generisala, treba iskoristiti naredbu V$ARCHIVED\_LOG. Na primer, pokrenimo sledeći upit:

SELECT THREAD#,SEQUENCE#,NAME

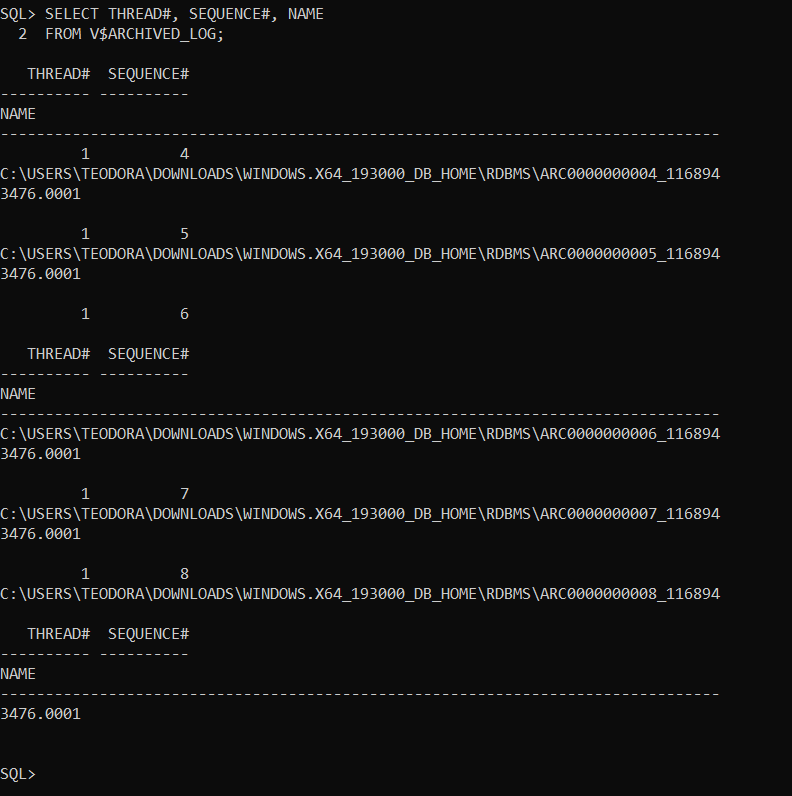
FROM V$ARCHIVED\_LOG;

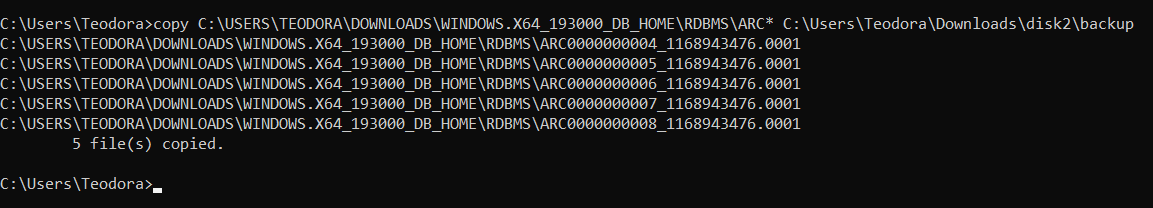
1. Napraviti rezervnu kopiju jedne kopije svakog broja sekvence loga koristeći alatku operativnog sistema. Ovaj primer rezerviše sve zapise u primarnoj lokaciji za arhiviranje na disk posvećen rezervnim kopijama zapisa:

% cp $ORACLE\_HOME/oracle/trgt/arch/\* /disk2/backup/arch

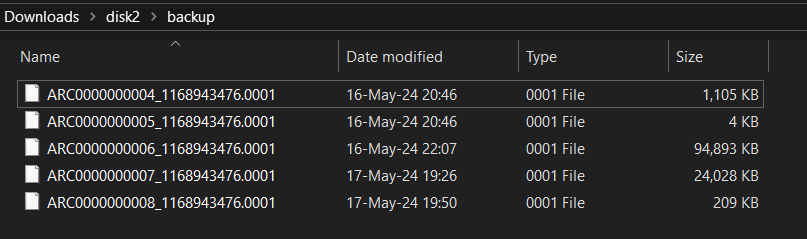
**Praktični primeri:**

**Primer 4.15**: SQL upitom SELECT THREAD#, SEQUENCE#, NAME FROM V$ARCHIVED\_LOG; dobijamo informacije o arhiviranim dnevnim datotekama koje je generisala baza podataka. Rezultat upita prikazuje broj niti (THREAD#), redni broj sekvence (SEQUENCE#) i putanju (NAME) svake arhivirane dnevne datoteke. Nakon toga, komandom copy na Windows operativnom sistemu kopiramo sve arhivirane dnevne datoteke (datoteke koje počinju sa "ARC") sa navedene putanje na drugu lokaciju, odnosno u direktorijum za rezervne kopije koji se nalazi na disku 2. Ovim korakom stvaramo rezervne kopije arhiviranih dnevnih datoteka radi osiguranja podataka u slučaju potrebe za oporavkom ili zaštite od gubitka podataka.





**Rezultat:**



### **Korisnički upravljan oporavak cele baze podataka**

Uobičajeno je izvršiti potpuni oporavak baze podataka kada se dogodi medijski kvar koji čini jednu ili više datoteka nedostupnim. Tokom potpunog oporavka baze podataka treba koristiti sav dostupni redo kako bi bazu podataka vratili na trenutnu SCN.

Pogled V$RECOVER\_FILE pokazuje koje datoteke zahtevaju oporavak. U zavisnosti od situacije, možemo izvršiti oporavak celokupne baze podataka ili pojedinačnih tablespace-ova ili datoteka podataka. Zbog toga što nakon potpunog oporavka ne moramo otvarati bazu podataka sa opcijom RESETLOGS, imamo mogućnost da izvršimo oporavak nekih datoteka podataka odjednom, a preostalih datoteka podataka kasnije.

U nastavku poglavlja biće opisani koraci potrebni za izvršenje operacija medijskog oporavka.

Prilikom izvođenja potpunog oporavka dok baza podataka nije otvorena, moguće je izabrati oporavak ili svih oštećenih datoteke podataka u jednoj operaciji ili pojedinačno oporavljanje svake oštećene datoteke podataka u odvojenim operacijama.

Ovaj postupak podrazumeva sledeće:

* Trenutni kontrolni fajl je dostupan.
* Postoje rezervne kopije svih potrebnih datoteka podataka.
* Svi neophodni arhivirani redo logovi su dostupni.

Da bi vratili i oporavili oštećene ili nedostajuće datoteke podataka potrebno je sledeće:

1. Ako je baza podataka otvorena, treba izvršiti V$RECOVER\_FILE kako bi odredili koje datoteke podataka moraju biti oporavljene i zašto. Ako planiramo da izvršimo potpuni oporavak umesto oporavka do određenog trenutka, možemo oporaviti samo one datoteke podataka koje zahtevaju oporavak, umesto celokupne baze podataka. Za oporavak do određenog trenutka, moramo vratiti i oporaviti sve datoteke podataka, osim ako ne izvršimo TSPITR u RMAN-u. Takođe možemo koristiti Flashback bazu podataka, ali ovaj postupak utiče na sve datoteke podataka i vraća celu bazu podataka na prethodno vreme. Možemo izvršiti V$RECOVER\_FILE da bi izlistali datoteke podataka koje zahtevaju oporavak prema broju datoteke sa njihovim statusom i informacijama o grešci.

SELECT FILE#, ERROR, ONLINE\_STATUS, CHANGE#, TIME

FROM V$RECOVER\_FILE;

Takođe možemo izvršiti korisne spojeve koristeći broj datoteke podataka i poglede V$DATAFILE i V$TABLESPACE kako bi dobili nazive datoteka podataka i tablespace-eva.

Kolona ERROR identifikuje problem za svaku datoteku koja zahteva oporavak.

1. Treba izvršiti V$ARCHIVED\_LOG i V$RECOVERY\_LOG kako bi odredili koje arhivirane redo log datoteke su potrebne.

Pogled V$ARCHIVED\_LOG nabraja imena datoteka za sve arhivirane redo logove, dok pogled V$RECOVERY\_LOG nabraja samo arhivirane redo logove koji su bazi podataka potrebni za izvršenje medijskog oporavka. Ovaj drugi pogled takođe uključuje verovatna imena datoteka na osnovu konvencije imenovanja specificirane korišćenjem parametra LOG\_ARCHIVE\_FORMAT.

1. Ako su svi arhivirani zapisi dostupni na podrazumevanoj lokaciji, preskočiti na korak 4.

Ako neki arhivirani zapisi moraju biti obnovljeni, i ako postoji dovoljno prostora, tada treba obnoviti potrebne arhivirane redo log datoteke na lokaciju specificiranu putem LOG\_ARCHIVE\_DEST\_1. Baza podataka automatski locira ispravan zapis kada je potreban tokom medijskog oporavka. Na primer, možda ćemo uneti komandu kao što je sledeća na Linux ili UNIX operativnom sistemu:

% cp /disk2/arch/\* $ORACLE\_HOME/oradata/trgt/arch

Ako nema dovoljno prostora, obnoviti neke ili sve potrebne arhivirane redo log datoteke na alternativnu lokaciju.

1. Ako je baza podataka otvorena, onda je treba isključiti. Na primer:

SHUTDOWN IMMEDIATE

1. Treba pregledati medijum kako bismo utvrdili izvor problema.

Ako je hardverski problem koji je prouzrokovao kvar medija bio privremen, i ako podaci nisu oštećeni (na primer, došlo je do kvara napajanja diska ili kontrolera), tada nije potreban medijski oporavak: pokrenite bazu podataka i nastavite sa normalnim operacijama.

Ako ne možete popraviti problem, tada nastavite sa korakom 6.

1. Ako su datoteke trajno oštećene, identifikovati najnovije rezervne kopije za oštećene datoteke. Obnoviti samo datoteke podataka oštećene medijskim kvarom: ne obnavljati neoštećene datoteke podataka niti bilo koje online redo log datoteke.

Na primer, ako je ORACLE\_HOME/oradata/trgt/users01.dbf jedina oštećena datoteka, tada možemo pronaći da je /backup/users01\_10\_24\_02.dbf najnovija rezervna kopija ove datoteke. Ako nemamo rezervnu kopiju određene datoteke, možda ćemo moći da kreiramo praznu zamensku datoteku koja se može oporaviti.

1. Koristiti alatku operativnog sistema za obnavljanje datoteka podataka na njihovu podrazumevanu lokaciju ili na novu lokaciju. Na primer, korisnik Linux-a ili UNIX-a koji obnavlja users01.dbf na njegovu podrazumevanu lokaciju može uneti:

% cp /backup/users01\_10\_24\_06.dbf $ORACLE\_HOME/oradata/trgt/users01.dbf

Koristite sledeće smernice prilikom određivanja gde treba obnoviti rezervne kopije datoteka podataka:

* Ako je hardverski problem popravljen i možemo obnoviti datoteke podataka na njihove podrazumevane lokacije, tada obnoviti datoteke podataka na njihove podrazumevane lokacije i započeti medijski oporavak.
* Ako hardverski problem i dalje postoji i ne možemo obnoviti datoteke podataka na njihove originalne lokacije, tada obnoviti datoteke podataka na alternativnom skladišnom uređaju. Navesti novu lokaciju ovih datoteka u kontrolnom fajlu pomoću ALTER DATABASE RENAME FILE naredbe.
* Ako obnavljamo datoteku podataka na sirovi disk ili particiju, tada je tehnika u osnovi ista kao kada obnavljamo na datoteku na fajl sistemu. Treba imajti na umu konvencije imenovanja za datoteke na sirovim uređajima (koje se razlikuju zavisno od operativnog sistema) i koristiti alatku operativnog sistema koja podržava sirove uređaje.

1. Povežimo se sa bazom podataka sa administratorskim privilegijama. Zatim pokrenimo novu instancu i montirajmo, ali ne treba otvarati bazu podataka. Na primer:

STARTUP MOUNT

1. Ako je obnovljena jedna ili više oštećenih datoteka podataka na alternativne lokacije, tada treba ažurirati kontrolni fajl baze podataka da odražava nove nazive datoteka podataka. Na primer, da bi promenili naziv datoteke podataka u tablespace-u users, uneti:

ALTER DATABASE RENAME FILE '?/oradata/trgt/users01.dbf' TO

'/disk2/users01.dbf';

1. Dobijmo nazive datoteka podataka i status svih datoteka podataka proverom liste datoteka podataka koja obično prati trenutni kontrolni fajl ili upitom u pogled V$DATAFILE. Na primer, uneti:

SELECT NAME,STATUS FROM V$DATAFILE;

1. Proveriti da su sve datoteke podataka koje zahtevaju oporavak online. Jedini izuzeci su datoteke podataka u tablespace-u koji je offline na normalan način ili datoteke podataka u tablespace-u koji je samo za čitanje. Na primer, da bi garantovali da je datoteka podataka nazvana /oracle/dbs/tbs\_10.f online, uneti sledeće:

ALTER DATABASE DATAFILE '/oracle/dbs/tbs\_10.f' ONLINE;

Ako je određena datoteka podataka već online, baza podataka ignoriše naredbu. Ako želimo, možemo kreirati skriptu koja će istovremeno dovesti sve datoteke podataka online, kao u sledećem primeru:

SPOOL onlineall.sql

SELECT 'ALTER DATABASE DATAFILE '''||name||''' ONLINE;' FROM V$DATAFILE;

SPOOL OFF

SQL> @onlineall

1. Ako smo obnovili arhivirane redo logove na alternativnu lokaciju, možemo specificirati lokaciju pre medijskog oporavka pomoću parametra LOGSOURCE komande SET u SQL\*Plus-u. Na primer, ako su zapisi smešteni u /tmp, možemo uneti sledeću komandu:

SET LOGSOURCE /tmp

Alternativno, moguće je preskočiti korak 12 i koristiti parametar FROM na RECOVER komandi kao u koraku 13. Na primer, ako su zapisi smešteni u /tmp, možemo uneti sledeću komandu:

RECOVER AUTOMATIC FROM '/tmp' DATABASE

1. Izdamo naredbu za oporavak baze podataka, tablespace-a ili datoteke podataka. Na primer, uneti jednu od sledećih RECOVER komandi:

RECOVER AUTOMATIC DATABASE # whole database

RECOVER AUTOMATIC TABLESPACE users # specific tablespace

RECOVER AUTOMATIC DATAFILE '?/oradata/trgt/users01.dbf';

# specific data file

Ako odlučimo da ne automatizujemo primenu arhiviranih redo logova, tada moramo prihvatiti ili odbiti svaki traženi zapis. Ako automatizujemo oporavak, tada baza podataka automatski primenjuje zapise. Oporavak se nastavlja dok svi potrebni arhivirani i online redo logovi nisu primenjeni na obnovljene datoteke podataka. Baza podataka nas obaveštava kada je medijski oporavak kompletiran:

Media recovery complete.

Ako nisu potrebni arhivirani redo logovi za potpuni medijski oporavak, tada baza podataka primenjuje sve neophodne online redo log datoteke i završava oporavak.

1. Nakon što se oporavak završi, možemo otvoriti bazu podataka za upotrebu.

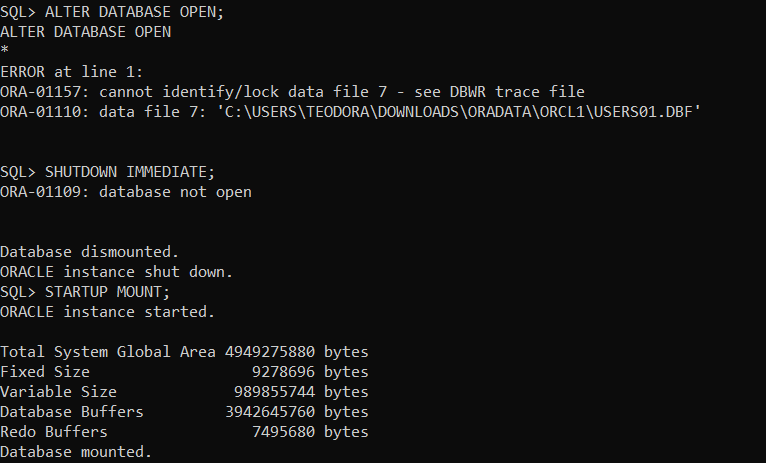
ALTER DATABASE OPEN;

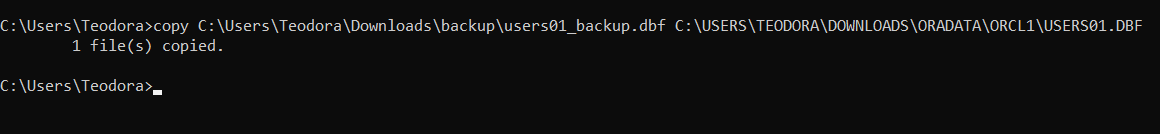
1. Nakon što su primenjeni arhivirani zapisi, i nakon što se uverimo da kopija svake grupe arhiviranih zapisa i dalje postoji u offline skladištu, treba obrisati obnovljenu kopiju arhiviranog redo log fajla kako bi oslobodili prostor na disku. Na primer:

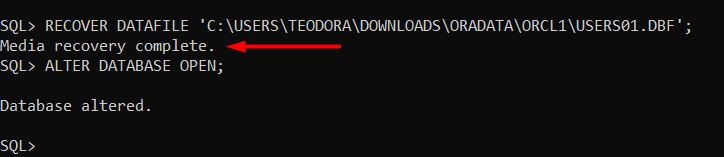
% rm /tmp/\*.arc

**Praktični primeri:**

**Primer 4.16**: Obnavljanje nedostajućeg fajla podatka users01.dbf. Nakon što je Oracle baza podataka pokrenuta, pokušaj otvaranja je rezultirao greškom ORA-01157, što je ukazalo na nemogućnost identifikacije ili zaključavanja datafile-a. Kako bi se rešio ovaj problem, prvo je kopiran backup fajl users01\_backup.dbf sa lokacije "C:\Users\Teodora\Downloads\backup\users01\_backup.dbf" na očekivanu lokaciju datafile-a USERS01.DBF. Zatim je korišćen SQL\*Plus za izvršavanje komande RECOVER DATAFILE nad nedostajućim datafile-om USERS01.DBF sa odgovarajućom putanjom, što je omogućilo obnovu datafile-a. Nakon uspešne obnove, baza podataka je ponovno otvorena korišćenjem komande ALTER DATABASE OPEN, čime je rešen problem i omogućeno normalno funkcionisanje baze podataka.







# **Zaključak**

U ovom radu detaljno su obrađene opcije za bekapovanje i oporavak Oracle baze podataka, pružajući sveobuhvatan pregled tehnika i alata koji su neophodni za osiguranje integriteta i konzistentnosti podataka. Kroz analizu backup politika, nivoa bekapovanja, kao i različitih metoda bekapovanja, uključujući bekapovanje na disk, eksterni FRA putem NFS-a i Oracle Object Storage, ovaj rad osigurava da korisnici imaju jasan uvid u dostupne opcije i njihovu primenu.

Ključni aspekt ovog rada je detaljno objašnjenje dve strategije za rešavanje bekapa – korišćenje Recovery Manager-a (RMAN) i korisnički upravljanih backup-ova i oporavka. Poglavlja posvećena RMAN-u pružaju konkretne primere pokretanja, povezivanja sa bazom, pravljenja različitih tipova backup-ova i oporavka cele baze podataka, dok se u poglavljima o korisnički upravljanim backup-ovima i oporavku obrađuju praktični koraci za pravljenje i upravljanje backup-ovima.

Kroz praktične primere koji su dati u okviru svake strategije, može se videti kako se ova rešenja primenjuju u realnim situacijama, što doprinosi boljem razumevanju i lakšoj implementaciji. Pored toga, mogućnost poređenja različitih rešenja omogućava korisnicima da odaberu najpogodniji pristup za svoje specifične potrebe i okruženja.

Ovaj seminarski rad ne samo da pruža teoretski pregled opcija za bekapovanje i oporavak, već kroz detaljno objašnjene postupke i praktične primere, omogućava korisnicima da steknu sveobuhvatno znanje i veštine potrebne za efikasno upravljanje bekapovanjem i oporavkom baza podataka. Ove osnove pružaju temelj za dalja istraživanja i moguća usavršavanja trenutnih tehnika, otvarajući vrata za inovativna rešenja i napredne metode u oblasti bekapovanja i oporavka podataka. Na taj način, rad ostavlja prostora za kontinuirani napredak i prilagođavanje budućim tehnološkim izazovima i zahtevima.

# **Literatura**

1. „Backup, Restore and Recover Databases“, Oracle® Database Appliance, July 2020

Dostupno: <https://docs.oracle.com/en/engineered-systems/oracle-database-appliance/19.8/cmtxn/backup-recover-restore.html>

1. „Backup, Restore and Recover Databases“, Oracle® Database Appliance, July 2018

Dostupno: <https://docs.oracle.com/cd/E93461_01/doc.122/e92398/backup-recovery.htm>

1. „Introduction to Backup and Recovery“, Oracle® Database, February 2024

Dostupno: <https://docs.oracle.com/en/database/oracle/oracle-database/19/bradv/introduction-backup-recovery.html>

1. „Introduction to Recovery Manager“, Oracle

Dostupno: <https://docs.oracle.com/cd/A97630_01/server.920/a96566/rcmintro.htm>

[5] „Getting Started with RMAN“, Oracle® Database, February 2024

Dostupno: <https://docs.oracle.com/en/database/oracle/oracle-database/19/bradv/getting-started-rman.html>

[6] „Making User-Managed Database Backups“, Oracle® Database, February 2024

Dostupno: <https://docs.oracle.com/en/database/oracle/oracle-database/19/bradv/user-managed-database-backups.html>

[7] „Performing User-Managed Backup and Recovery“, Oracle® Database, February 2024

Dostupno: <https://docs.oracle.com/en/database/oracle/oracle-database/19/bradv/part-user-managed-backup-recovery.html>

[8] „Overview of Database Backup and Recovery“, Oracle® Database, July 2020

Dostupno: <https://docs.oracle.com/en/database/oracle/oracle-database/19/admqs/performing-backup-and-recovery.html>

[9] „Performing Complete Database Recovery“, Oracle® Database, March 2024

Dostupno: <https://docs.oracle.com/en/database/oracle/oracle-database/21/bradv/rman-complete-database-recovery.html>

1. Eksterni FRA putem Network File System (NFS) je mrežno skladište za Flash Recovery Area, omogućavajući bekap fajlovima da se čuvaju na udaljenom mrežnom uređaju umesto na lokalnom disku. [↑](#footnote-ref-1)
2. Oracle Object Storage je cloud servis koji omogućava skalabilno i sigurno čuvanje nestruktuiranih podataka kao što su fajlovi, backup-i i arhive, koristeći objektnu arhitekturu. [↑](#footnote-ref-2)
3. Point-in-Time Database Recovery (PITR) je tehnika oporavka baze podataka koja omogućava vraćanje baze podataka u stanje u kojem je bila u određenom trenutku u prošlosti, pre neželjenih promena ili grešaka. [↑](#footnote-ref-3)
4. Oporavak na osnovu System Change Number-a je proces vraćanja baze podataka u određeno stanje u prošlosti koristeći specifični identifikator promene (SCN). [↑](#footnote-ref-4)
5. RMAN (Recovery Manager) je alat koji se koristi u Oracle baza podataka za upravljanje bekapovima, obnavljanjem podataka i zaštitom od gubitka podataka. [↑](#footnote-ref-5)
6. DCS-agent je softverski entitet koji se koristi u Oracle Database Appliance okruženju kako bi automatizovao procese vezane za bekapovanje, obnavljanje i upravljanje bazama podataka. [↑](#footnote-ref-6)