

RMAN Progressive Tablespace Recovery Strategy

Minimize recovery time by using active and archive partitions in
separate tablespaces

PURPOSE STATEMENT

This brief outlines a method to minimize RMAN recovery time for accessing critical business application data by implementing active and historical partitions in different tablespaces. During recovery, online (active) tablespaces are restored first so the database becomes operational more quickly, while archive tablespaces – which are less critical and accessed – are restored afterward in the background.

DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

TABLE OF CONTENTS

Purpose Statement	1
Disclaimer	1
Introduction	3
Procedure Overview	3
Setup environment	3
Create separate tablespaces	3
Verify list of tablespaces	4
Create partitioned table	4
Verify Partition and Tablespace details	5
Verify data	5
Backup database	5
Verify backups are available	6
In preparation for Recovery exercises, delete database files	6
Recovery Exercise 1 – Progressive tablespace Recovery from L0 backup	7
Overview of Steps	7
Restore control file	7
Mount Database and catalog Any Remaining backup sets	7
Report Schema for list of tablespaces	7
Restore Database skip archive tablespaces	7
Recover Database skip archive tablespaces	7
Open Database and query data	7
Restore and recover archive tablespaces	8
Bring tablespaces online and check alert log	8
Verify status of tablespaces	8
Verify Table Counts	8
Recovery Exercise 2 – Progressive tablespace Recovery from multiple L0 backups	10
Overview of Steps	10
Insert data in different partitions	10
Perform Level 0 backup with all tablespaces with tag DB_LEV0_COMPLETE.	10
Enable backup optimization	10
Put Archive tablespaces in read only mode	10
Insert data in Online tablespace	10
Backup database skipping read only tablespaces with TAG DB_LEV0_SKIPPED	11
List backup summary and control file backups	11
To prepare for restore exercise, delete database files	12
Restore controlfile	12
List Backupset summary and tablespace backups	12
Restore Database Preview	12
Restore Database from DB_LEV0_SKIPPED tag	13
Recover Database, skipping read only tablespaces	13
Open Database and query data from online tablespaces	13
Determine the pending tablespace or datafiles	13
Restore the archive tablespaces	13
Bring archive tablespaces online	14
Verify status of data files and tablespaces	14
Query recovered data	14
CONCLUSION	15

INTRODUCTION

As enterprise databases grow into the multi-terabyte and petabyte range, database recovery has become one of the most challenging operational tasks. Traditional restore procedures work well for small and medium-sized databases, but Very Large Databases (VLDBs) – 100+ terabyte into petabytes – often contain years of historical data that rarely changes while only a small percentage of the data remains active.

Restoring the entire database after a disaster—even when only currently active business data is immediately needed—can significantly extend Recovery Time Objectives (RTOs). Organizations therefore need strategies that prioritize business continuity without compromising data integrity.

This method addresses the prolonged RTO issue by creating two logical categories of tablespaces – online (read-write) and archive (rarely updated / read-only), where online tablespaces are recovered first and opened for read-write access by critical applications. Archive tablespaces are then recovered and made accessible afterward in the background.

PROCEDURE OVERVIEW

- Create separate tablespaces
- Create partitioned table
- Verify partition and tablespace details
- Verify data
- Backup database
- Delete database files
- Recovery Exercise 1 – Progressive tablespace recovery from L0 backup
- Recovery Exercise 2 – Progressive tablespace recovery from multiple backups (L0 normal, L0 skipping read only tablespace)

SETUP ENVIRONMENT

CREATE SEPARATE TABLESPACES

- EXAMPLE_ONLINE% - will store active partition
- EXAMPLE_ARCHIVE% - will store all archive (historical) partitions

```
SQL> select name from v$databases;

NAME
-----
UI01POC

SQL> CREATE TABLESPACE example_online DATAFILE '+DATA01' SIZE 64M autoextend on next 1M MAXSIZE 1G
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO; 2 3

Tablespace created.

SQL> CREATE TABLESPACE example_archive DATAFILE '+DATA01' SIZE 64M autoextend on next 1M MAXSIZE 1G
EXTENT MANAGEMENT LOCAL
SEGMENT SPACE MANAGEMENT AUTO; 2 3

Tablespace created.
```

```

SQL> CREATE TABLESPACE example_archive2 DATAFILE '+DATA01' SIZE 8M autoextend on next 1M MAXSIZE 1G
      EXTENT MANAGEMENT LOCAL
      SEGMENT SPACE MANAGEMENT AUTO;  2    3

Tablespace created.

SQL> CREATE TABLESPACE example_archive3 DATAFILE '+DATA01' SIZE 8M autoextend on next 1M MAXSIZE 1G
      EXTENT MANAGEMENT LOCAL
      SEGMENT SPACE MANAGEMENT AUTO;  2    3

Tablespace created.

SQL> CREATE TABLESPACE example_online2 DATAFILE '+DATA01' SIZE 8M autoextend on next 1M MAXSIZE 1G
      EXTENT MANAGEMENT LOCAL
      SEGMENT SPACE MANAGEMENT AUTO;  2    3

Tablespace created.

SQL> CREATE TABLESPACE example_online3 DATAFILE '+DATA01' SIZE 8M autoextend on next 1M MAXSIZE 1G
      EXTENT MANAGEMENT LOCAL
      SEGMENT SPACE MANAGEMENT AUTO;
  2    3
Tablespace created.

SQL> CREATE TABLESPACE example_archive4 DATAFILE '+DATA01' SIZE 8M autoextend on next 1M MAXSIZE 1G
      EXTENT MANAGEMENT LOCAL
      SEGMENT SPACE MANAGEMENT AUTO;  2    3

Tablespace created.

SQL> CREATE TABLESPACE example_archive5 DATAFILE '+DATA01' SIZE 8M autoextend on next 1M MAXSIZE 1G
      EXTENT MANAGEMENT LOCAL
      SEGMENT SPACE MANAGEMENT AUTO;  2    3

Tablespace created.

```

VERIFY LIST OF TABLESPACES

```

SQL> select tablespace_name from dba_tablespaces;

TABLESPACE_NAME
-----
SYSTEM
SYSaux
UNDOTBS1
TEMP
USERS
EXAMPLE_ONLINE
EXAMPLE_ARCHIVE
EXAMPLE_ARCHIVE2
EXAMPLE_ARCHIVE3
EXAMPLE_ONLINE2
EXAMPLE_ONLINE3
EXAMPLE_ARCHIVE4
EXAMPLE_ARCHIVE5

13 rows selected.

```

CREATE PARTITIONED TABLE

Create table with PART1_ONLINE, PART2_ONLINE, PART3_ONLINE where invoice# 1, 2, 3 are stored in EXAMPLE_ONLINE% tablespaces.

The rest of the partitions are stored in EXAMPLE_ARCHIVE% tablespace.

```

SQL> CREATE TABLE invoice_list
PARTITION BY LIST(invoice#)
(partition part1_online VALUES (1) tablespace example_online,
partition part2_online VALUES (2) tablespace example_online2,
partition part3_online VALUES (3) tablespace example_online3,
partition part4_archive VALUES (4) tablespace example_archive,
partition part5_archive VALUES (5) tablespace example_archive2,
partition part6_archive VALUES (6) tablespace example_archive3,
partition part7_archive VALUES (7) tablespace example_archive4,
partition part8_archive VALUES (DEFAULT) tablespace example_archive5)
AS SELECT ROWNUM AS invoice_uniq
, MOD(ROWNUM,10) AS invoice#
, MOD(ROWNUM,100000) AS amount
2 3 4 , RPAD(rownum,1000,'A') AS Details
FROM all_source, all_source
WHERE rownum<= 1000000; 5 6 7 8 9 10 11 12 13 14 15 16

Table created.

SQL> alter system archive log current;

System altered.

```

VERIFY PARTITION AND TABLESPACE DETAILS

TABLE_NAME	PARTITION_NAME	HIGH_VALUE	TABLESPACE_NAME	BLOCKS	NUM_ROWS
INVOICE_LIST	PART1_ONLINE	1	EXAMPLE_ONLINE	14432	100000
INVOICE_LIST	PART2_ONLINE	2	EXAMPLE_ONLINE2	14432	100000
INVOICE_LIST	PART3_ONLINE	3	EXAMPLE_ONLINE3	14432	100000
INVOICE_LIST	PART4_ARCHIVE	4	EXAMPLE_ARCHIVE	14432	100000
INVOICE_LIST	PART5_ARCHIVE	5	EXAMPLE_ARCHIVE2	14432	100000
INVOICE_LIST	PART6_ARCHIVE	6	EXAMPLE_ARCHIVE3	14432	100000
INVOICE_LIST	PART7_ARCHIVE	7	EXAMPLE_ARCHIVE4	14432	100000
INVOICE_LIST	PART8_ARCHIVE	DEFAULT	EXAMPLE_ARCHIVE5	43130	300000

VERIFY DATA

```

SQL> select * from invoice_list where invoice#=2 and rownum < 3;
INVOICE_UNIQ INVOICE# AMOUNT DETAILS
-----
2 2 2 AAAAAAAAAAAAAA...
12 2 12 AAAAAAAAAAAAAA...

SQL> select * from invoice_list where invoice#=1 and rownum < 3;
INVOICE_UNIQ INVOICE# AMOUNT DETAILS
-----
1 1 1 AAAAAAAAAAAAAA...
11 1 11 AAAAAAAAAAAAAA...

```

BACKUP DATABASE

RMAN full backup with controlfile and archive logs

```

RMAN> run
{
allocate channel ch00 device type disk;
backup incremental level=0 database include current controlfile
format '/local/dbbkup/oracle/UIO1MK/UIO1POC/DBLO_%d_%s_%p_%t'
tag 'DB_LEVO'
plus archivelog
format '/local/dbbkup/oracle/UIO1MK/UIO1POC/LOARC_%d_%s_%p_%t';
release channel ch00;
}

```

VERIFY BACKUPS ARE AVAILABLE

RMAN> List backup

```
RMAN> list backup summary;
```

List of Backups

Key	TY	LV	S	Device	Type	Completion Time	#Pieces	#Copies	Compressed	Tag
5	B	A	A	DISK		24-APR-16	1	1	NO	ARCH_LEVO
6	B	0	A	DISK		24-APR-16	1	1	NO	DB_LEVO
7	B	0	A	DISK		24-APR-16	1	1	NO	DB_LEVO
8	B	A	A	DISK		24-APR-16	1	1	NO	ARCH_LEVO

IN PREPARATION FOR RECOVERY EXERCISES, DELETE DATABASE FILES

Shutdown the database. Delete all control files, data files, redo, and archived logs.

```
ASMCMD> cd UIO1POC
ASMCMD> ls
CONTROLFILE/  DATAFILE/  ONLINELOG/  PARAMETERFILE/  TEMPFILE/  SPFILEUIOPOC.ora
ASMCMD> rm -rf CONTROLFILE/
ASMCMD> rm -rf DATAFILE/
ASMCMD> cd ..
ASMCMD> cd RECO01
ASMCMD> ls
ARCHIVELOG/  CONTROLFILE/  ONLINELOG/
ASMCMD> rm -rf ARCHIVELOG/
ASMCMD> rm -rf CONTROLFILE/
ASMCMD> rm -rf ONLINELOG/
```

RECOVERY EXERCISE 1 – PROGRESSIVE TABLESPACE RECOVERY FROM L0 BACKUP

OVERVIEW OF STEPS

Restore the online partitions tablespace first
Recover and open the database
Restore and recover the archive partition tablespace

RESTORE CONTROL FILE

Startup nomount instance and restore controlfile using RMAN with direct path of backupset.

It is assumed that you have RMAN catalog setup or output file from original backup to locate control file backupset.

```
SQL> startup nomount
ORACLE instance started.
RMAN> restore controlfile from '/local/dbbkup/oracle/UIO1MK/UIO1POC/DBLO_UIO1POC_7_1_910009513';
output file name=+DATA01/UIO1POC/controlfile/current.282.910011297
finished restore at 24-APR-16
```

MOUNT DATABASE AND CATALOG ANY REMAINING BACKUP SETS

```
RMAN> sql "alter database mount";
database mounted
RMAN> catalog start with '/local/dbbkup/oracle/UIO1MK/UIO1POC/';
List of Unknown to the Database
File Name: /local/dbbkup/oracle/UIO1MK/UIO1POC/DBLO_UIO1POC_7_1_910009513
File Name: /local/dbbkup/oracle/UIO1MK/UIO1POC/LOARC_UIO1POC_8_1_910009515
cataloging done
```

REPORT SCHEMA FOR LIST OF TABLESPACES

```
RMAN> report schema;
List of Permanent Files
File#  Tablespace      Name
-----
1      SYSTEM            +DATA01/UIO1POC/datafile/system.278.909864345
4      EXAMPLE_ONLINE     +DATA01/UIO1POC/datafile/example_online.272.910006435
5      EXAMPLE_ARCHIVE    +DATA01/UIO1POC/datafile/example_archive.271.909686595
```

RESTORE DATABASE SKIP ARCHIVE TABLESPACES

```
RMAN> restore database skip tablespace example_archive, example_archive2, example_archive3,
example_archive4, example_archive5;
Starting restore at 24-APR-16
channel ORA_DISK_1: restoring datafile 00004
channel ORA_DISK_1: restoring datafile 00003
Finished restore at 24-APR-16
```

RECOVER DATABASE SKIP ARCHIVE TABLESPACES

```
RMAN> recover database skip tablespace example_archive, example_archive2, example_archive3,
example_archive4, example_archive5;
media recovery complete
```

OPEN DATABASE AND QUERY DATA

```
SQL> alter database open resetlogs;
Database altered.
SQL> select count(*) from invoice_list;
COUNT(*)
-----
1000005
SQL> select count(*) from invoice_list where invoice# in (1,2,3);
COUNT(*)
```

```
-----  
300005
```

Verify that Insert Data in online partitions succeeds.

Observe that same command fails for archive partitions, as those tablespaces have not yet been recovered.

```
SQL> insert into invoice_list values (1,1,1,'TEST');  
1 row created.  
SQL> insert into invoice_list values (1,0,1,'TEST');  
ERROR at line 1:  
ORA-00376: file 11 cannot be read at this time  
ORA-01110: data file 11: +DATA01/UIO1POC/datafile/example_archive5.289.910007951
```

RESTORE AND RECOVER ARCHIVE TABLESPACES

```
RMAN> restore tablespace example_archive, example_archive2, example_archive3, example_archive4,  
example_archive5;  
RMAN> recover tablespace example_archive, example_archive2, example_archive3, example_archive4,  
example_archive5;
```

BRING TABLESPACES ONLINE AND CHECK ALERT LOG

```
SQL> alter tablespace example_archive online;  
Tablespace altered.  
SQL> alter tablespace example_archive2 online;  
Tablespace altered.  
SQL> alter tablespace example_archive3 online;  
Tablespace altered.  
SQL> alter tablespace example_archive4 online;  
Tablespace altered.  
SQL> alter tablespace example_archive5 online;  
Tablespace altered.
```

VERIFY STATUS OF TABLESPACES

```
SQL> select tablespace_name,status from dba_tablespaces;
```

TABLESPACE_NAME	STATUS
SYSTEM	ONLINE
SYSAUX	ONLINE
UNDOTBS1	ONLINE
TEMP	ONLINE
USERS	ONLINE
EXAMPLE_ONLINE	ONLINE
EXAMPLE_ARCHIVE	ONLINE
EXAMPLE_ARCHIVE2	ONLINE
EXAMPLE_ARCHIVE3	ONLINE
EXAMPLE_ONLINE2	ONLINE
EXAMPLE_ONLINE3	ONLINE

TABLESPACE_NAME	STATUS
EXAMPLE_ARCHIVE4	ONLINE
EXAMPLE_ARCHIVE5	ONLINE

```
13 rows selected.
```

VERIFY TABLE COUNTS

```
SQL> select count(*) from invoice_list2;  
COUNT (*)  
-----  
1000005
```

```
SQL> select count(*) from invoice_list where invoice#=9;
COUNT(*)
-----
100000
```

RECOVERY EXERCISE 2 – PROGRESSIVE TABLESPACE RECOVERY FROM MULTIPLE L0 BACKUPS

OVERVIEW OF STEPS

1. Restore the control file from latest backup
2. Restore online partitions tablespace from latest backup
3. Recover and open the database
4. Restore the archive partition tablespace from previous backup (complete set with read-only tablespaces)
5. Recover the archive tablespaces and bring them online

INSERT DATA IN DIFFERENT PARTITIONS

```
SQL> select count(*), invoice# from invoice_list where invoice# in (1,2,3,9) group by invoice#;
COUNT(*)  INVOICE#
-----  -
100009      1
100004      2
100003      3
100000      9
```

PERFORM LEVEL 0 BACKUP WITH ALL TABLESPACES WITH TAG DB_LEV0_COMPLETE.

```
RMAN> run
{
  allocate channel ch00 device type disk;
  backup incremental level=0 database include current controlfile
    format '/local/dbbkup/oracle/UIO1MK/UIO1POC/DBLO_%d_%s_%p_%t'
    tag 'DB_LEV0 COMPLETE'
    plus archivelog
    format '/local/dbbkup/oracle/UIO1MK/UIO1POC/LOARC_%d_%s_%p_%t';
  release channel ch00;
}
```

ENABLE BACKUP OPTIMIZATION

```
RMAN> configure backup optimization on;
new RMAN configuration parameters:
CONFIGURE BACKUP OPTIMIZATION ON;
```

PUT ARCHIVE TABLESPACES IN READ ONLY MODE

```
SQL> alter tablespace example_archive read only;
Tablespace altered.
SQL> alter tablespace example_archive2 read only;
Tablespace altered.
SQL> alter tablespace example_archive3 read only;
Tablespace altered.
SQL> alter tablespace example_archive4 read only;
Tablespace altered.
SQL> alter tablespace example_archive5 read only;
Tablespace altered.
```

INSERT DATA IN ONLINE TABLESPACE

```
SQL> insert into invoice_list values (1,1,1,'TEST');
1 row created.
SQL> select count(*), invoice# from invoice_list where invoice# in (1,2,3,5,9) group by invoice#;
COUNT(*)  INVOICE#
-----  -
100005      1
100004      2
100003      3
100004      5
100004      9
```

BACKUP DATABASE SKIPPING READ ONLY TABLESPACES WITH TAG DB_LEVO_SKIPPED

```

RMAN> run
{
  allocate channel ch00 device type disk;
  backup incremental level=0 database skip readonly include current controlfile
    format '/local/dbbkup/oracle/UI01MK/UI01POC/DBLO_%d_%s_%p_%t'
    tag 'DB_LEVO_SKIPPED'
  plus archivelog
    format '/local/dbbkup/oracle/UI01MK/UI01POC/LOARC_%d_%s_%p_%t';
  release channel ch00;
}

```

LIST BACKUP SUMMARY AND CONTROL FILE BACKUPS

List of Backups
=====

Key	TY	LV	S	Device	Type	Completion Time	#Pieces	#Copies	Compressed	Tag
9	B	A	A	DISK		24-APR-16	1	1	NO	ARCH_LEVO_COMPLETE
10	B	0	A	DISK		24-APR-16	1	1	NO	DB_LEVO_COMPLETE
11	B	0	A	DISK		24-APR-16	1	1	NO	DB_LEVO_COMPLETE
12	B	A	A	DISK		24-APR-16	1	1	NO	ARCH_LEVO_COMPLETE
13	B	A	A	DISK		24-APR-16	1	1	NO	ARCH_LEVO_SKIPPED
14	B	0	A	DISK		24-APR-16	1	1	NO	DB_LEVO_SKIPPED
15	B	0	A	DISK		24-APR-16	1	1	NO	DB_LEVO_SKIPPED
16	B	A	A	DISK		24-APR-16	1	1	NO	ARCH_LEVO_SKIPPED

RMAN> list backup of controlfile;

List of Backup Sets
=====

BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
11	Incr	0	9.39M	DISK		00:00:01	24-APR-16
BP Key: 11 Status: AVAILABLE Compressed: NO Tag: DB_LEVO_COMPLETE							
Piece Name: /local/dbbkup/oracle/UI01MKI/UI01POC/DBLO_UI01POC_11_1_910015697							
Control File Included: Ckp SCN: 1232364 Ckp time: 24-APR-16							

BS Key	Type	LV	Size	Device	Type	Elapsed Time	Completion Time
15	Incr	0	9.45M	DISK		00:00:01	24-APR-16
BP Key: 15 Status: AVAILABLE Compressed: NO Tag: DB_LEVO_SKIPPED							
Piece Name: /local/dbbkup/oracle/UI01MKI/UI01POC/DBLO_UI01POC_15_1_910018227							
Control File Included: Ckp SCN: 1242781 Ckp time: 24-APR-16							

TO PREPARE FOR RESTORE EXERCISE, DELETE DATABASE FILES

```
ASMCMD> cd UI01POC
ASMCMD> ls
CONTROLFILE/
DATAFILE/
ONLINELOG/
PARAMETERFILE/
TEMPFILE/
spfileUI01POC.ora
ASMCMD> rm -rf CONTROLFILE/
ASMCMD> rm -rf DATAFILE/
ASMCMD> pwd
+DATA01/UI01POC
ASMCMD> rm -rf ONLINELOG/
ASMCMD> cd ..
ASMCMD> ls
ASM/
UI01MKI/
UI01MKI_V$IN21P1144/
UI01POC/
ASMCMD> cd ..
ASMCMD> cd RECO01
ASMCMD> ls
UI01MKI/
UI01MKI_V$IN21P1144/
UI01POC/
ASMCMD> cd UI01POC
ASMCMD> ls
ARCHIVELOG/
CONTROLFILE/
ONLINELOG/
ASMCMD> pwd
+RECO01/UI01POC
ASMCMD> rm -rf ARCHIVELOG/
ASMCMD> rm -rf CONTROLFILE/
ASMCMD> rm -rf ONLINELOG/
```

RESTORE CONTROLFILE

```
RMAN> restore controlfile from '/local/dbbkup/oracle/UI01MK/UI01POC/DBLO_UI01POC_15_1_910018227';
output file name=+DATA01/UI01POC/controlfile/current.279.910011845
finished restore at 24-APR-16
```

LIST BACKUPSET SUMMARY AND TABLESPACE BACKUPS

```
RMAN> list backup summary;
RMAN> list backup of tablespace example_archive;
BS Key Size Device Type Completion Time Tag
11 1.39G DISK 24-APR-16 DB_LEV0_COMPLETE
12 2.50K DISK 24-APR-16 ARCH_LEV0_COMPLETE
```

RESTORE DATABASE PREVIEW

This provides list of backup sets.

```
RMAN> restore database preview;
List of Backup Sets
BS Key Type LV Size Device Type Elapsed Time Completion Time
13 Incr 0 1.39G DISK 00:01:00 24-APR-16
Piece Name: /local/dbbkup/oracle/UI01MK/UI01POC/DBLO_UI01POC_13_1_910018210
```

RESTORE DATABASE FROM DB_LEV0_SKIPPED TAG

Determine the list of ONLINE (read-write) tablespaces using dba_tablespaces.

```
SQL> select tablespace_name, status from dba_tablespaces;
TABLESPACE_NAME    STATUS
SYSTEM             ONLINE
SYSAUX             ONLINE
UNDOTBS1           ONLINE
TEMP               ONLINE
USERS              ONLINE
EXAMPLE_ONLINE     ONLINE
EXAMPLE_ARCHIVE    READ ONLY
EXAMPLE_ARCHIVE2   READ ONLY
EXAMPLE_ARCHIVE3   READ ONLY
EXAMPLE_ONLINE2    ONLINE
EXAMPLE_ONLINE3    ONLINE
```

Now perform restore. Tags are used to simplify restore of just the read-write tablespaces.

```
RMAN> restore database tag 'DB_LEV0_SKIPPED';
Starting restore at 24-APR-16
channel ORA_DISK_1: restoring datafile 00004
channel ORA_DISK_1: restoring datafile 00003
Finished restore at 24-APR-16
```

RECOVER DATABASE, SKIPPING READ ONLY TABLESPACES

```
RMAN> recover database skip tablespace example_archive, example_archive2, example_archive3,
example_archive4, example_archive5;
media recovery complete
```

OPEN DATABASE AND QUERY DATA FROM ONLINE TABLESPACES

```
SQL> alter database open resetlogs;
Database altered.
SQL> select count(*) from invoice_list where invoice# in (1,2,3,5,9) group by invoice#,
COUNT(*)
-----
100005
100004
100003
100004
100004
```

DETERMINE THE PENDING TABLESPACE OR DATAFILES

```
SQL> select file#, name, status from v$datafile;
FILE#  NAME                                     STATUS
5      +DATA01/UIO1POC/datafile/example_online.279.910019925 ONLINE
6      +DATA01/UIO1POC/datafile/example_archive.276.910013049 OFFLINE
7      +DATA01/UIO1POC/datafile/example_archive2.277.910013049 OFFLINE
8      +DATA01/UIO1POC/datafile/example_archive3.278.910013049 OFFLINE
```

RESTORE THE ARCHIVE TABLESPACES

Will use the restore Datafile syntax here. We can use the tablespace one as well; whichever you feel is easier.

```
RMAN> restore datafile 6,7,8,9,10;
Starting restore at 24-APR-16
channel ORA_DISK_1: restoring datafile 00006
channel ORA_DISK_1: restoring datafile 00007
channel ORA_DISK_1: restoring datafile 00008
channel ORA_DISK_1: restoring datafile 00009
channel ORA_DISK_1: restoring datafile 00010
Finished restore at 24-APR-16
```

BRING ARCHIVE TABLESPACES ONLINE

```
SQL> alter tablespace example_archive online;
Tablespace altered.
SQL> alter tablespace example_archive2 online;
Tablespace altered.
SQL> alter tablespace example_archive3 online;
Tablespace altered.
SQL> alter tablespace example_archive4 online;
Tablespace altered.
SQL> alter tablespace example_archive5 online;
Tablespace altered.
```

VERIFY STATUS OF DATA FILES AND TABLESPACES

FILE#	NAME	STATUS
1	+DATA01/ui01poc/datafile/system.271.910019925	SYSTEM
2	+DATA01/ui01poc/datafile/sysaux.283.910019925	ONLINE
3	+DATA01/ui01poc/datafile/undotbs1.276.910019925	ONLINE
4	+DATA01/ui01poc/datafile/users.272.910019925	ONLINE
5	+DATA01/ui01poc/datafile/example_online.279.910019925	ONLINE
6	+DATA01/ui01poc/datafile/example_archive.288.910021317	ONLINE
7	+DATA01/ui01poc/datafile/example_archive2.274.910021317	ONLINE
8	+DATA01/ui01poc/datafile/example_archive3.281.910021317	ONLINE
9	+DATA01/ui01poc/datafile/example_online2.278.910019925	ONLINE
10	+DATA01/ui01poc/datafile/example_online3.277.910019925	ONLINE
11	+DATA01/ui01poc/datafile/example_archive4.282.910021317	ONLINE
12	+DATA01/ui01poc/datafile/example_archive5.287.910021317	ONLINE

12 rows selected

```
SQL> select tablespace_name, status from dba_tablespaces;
TABLESPACE_NAME  STATUS
EXAMPLE_ONLINE   ONLINE
EXAMPLE_ARCHIVE  READ ONLY
EXAMPLE_ARCHIVE2 READ ONLY
EXAMPLE_ARCHIVE3 READ ONLY
EXAMPLE_ONLINE2  ONLINE
EXAMPLE_ONLINE3  ONLINE
SQL> select count(*), invoice# from invoice_list where invoice# in (1,2,3,5,9) group by invoice#;
COUNT(*)  INVOICE#
-----
100005     1
100004     2
100003     3
100004     5
100004     9
```

QUERY RECOVERED DATA

```
SQL> select count(*),invoice# from invoice_list where invoice# in (1,2,3,5,9) group by invoice#;
```

COUNT (*)	INVOICE#
100011	1
100006	2
100005	3
100000	5
100004	9

CONCLUSION

As database volumes continue to grow, recovery strategies must also grow and adapt to optimize recovery time objectives. Separating active and historical partitions into dedicated tablespaces allows RMAN to prioritize getting business-critical back online for read-write access as quickly as possible.

The approach demonstrates how progressive tablespace recovery using online and read-only tablespaces work together to minimize backup windows and reduce downtime while preserving complete recoverability.

For organizations managing large databases, optimizing recovery by selective tablespaces provides a practical path toward faster disaster recovery, lower backup overhead, and improved operational resilience.

CONNECT WITH US

Call +1.800.ORACLE1 or visit oracle.com.

Outside North America, find your local office at oracle.com/contact.



blogs.oracle.com



facebook.com/oracle



twitter.com/oracle

Copyright © 2026, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0120

RMAN Progressive Tablespace Recovery Strategy
July, 2026

