

UNIT 7

Backup and Recovery

(5 Hours) — Database Administration (CACS405)

*Note: These notes are prepared for study purposes and **may contain errors or omissions**. Syntax, file paths and screen names differ between Oracle versions and operating systems (examples here follow Oracle 19c on Linux). Please verify important points with your textbook, class notes and the official Oracle documentation, and test commands on a practice database, never on a production system.*

7.1 Backup and Recovery Concepts

A **backup** is a copy of data that can be used to rebuild the database after a loss. **Recovery** is the process of rebuilding: first **restore** the lost files from a backup, then **recover** by applying redo (archived and online redo logs) to bring the data up to date. The DBA's job is to make sure the database can always be recovered within the time and data-loss limits the business accepts.

7.1.1 Types of Failures

Failure	Cause / example	Usual solution
Statement failure	Invalid SQL, insufficient privileges or space	Correct the statement or grant space; Oracle rolls back the statement.
User process failure	Client crashes or disconnects abnormally	PMON cleans up automatically.
Network failure	Listener or network down	Restore network; use redundancy.
User error	Accidental DELETE / DROP TABLE	Flashback technologies, or point-in-time recovery.
Instance failure	Power loss, SHUTDOWN ABORT, OS crash	Automatic instance recovery at next startup (roll forward using redo, roll back uncommitted work using undo).
Media failure	Disk failure; datafile or control file deleted	Restore from backup and perform media recovery .

7.1.2 Types of Backups

Classification	Types	Explanation
Physical vs logical	Physical / logical	Physical backup copies database files (RMAN or OS copy). Logical backup exports objects and data as SQL-like dump files (Data Pump).
Consistent vs inconsistent	Cold / hot	A consistent (cold) backup is taken when the database is cleanly shut down. An inconsistent (hot) backup is taken while the database is open and needs redo to be applied on recovery (ARCHIVELOG mode required).
Whole vs partial	Whole / tablespace / datafile	Whole backup covers all datafiles and the control file. Partial backs up selected files.
Full vs incremental	Level 0 / level 1	Level 0 is a full baseline. Level 1 copies only blocks changed since a previous backup: differential (since the last level 0 or 1) or cumulative (since the last level 0).
Backup set vs image copy	RMAN formats	A backup set is a compressed-capable RMAN-format file containing many datafile blocks; an image copy is an exact copy of a file.

7.1.3 Types of Recovery

- **Complete recovery** — recovers all committed transactions up to the moment of failure (needs all archived logs since the backup).
- **Incomplete / point-in-time recovery (PITR)** — recovers to an earlier time or SCN (for example, just before a user dropped a table); the database is then opened with `RESETLOGS`.
- **Instance recovery** — automatic, done by SMON at startup, after a crash.
- **Media recovery** — recovery of datafiles from backups, started by the DBA.

7.2 Defining a Backup and Recovery Strategy

A strategy is the plan describing what to back up, how often, where to store copies, and how to recover. Key goals are:

- **RPO (Recovery Point Objective)** — how much data loss is acceptable (for example "at most 15 minutes").
- **RTO (Recovery Time Objective)** — how long the database can be down (for example "restore within 2 hours").

Design decision	Recommendation
Log mode	Production databases in ARCHIVELOG mode (Unit 3).
Backup schedule	e.g. weekly level 0 (full), daily level 1 incremental, frequent archived-log backups.
Control file	Enable control file autobackup ; multiplex control files and redo logs.
Storage	Keep backups in the Fast Recovery Area and also on separate storage/off-site; never on the same disk as the database.
Retention policy	Recovery window (e.g. 7 days) or redundancy (number of copies).
Testing	Regularly test restores ; a backup that has not been tested may not work.
Documentation	Written procedures for each failure type and clear responsibilities.

7.2.1 Backup and Recovery Options in Oracle

Option	Description
RMAN (Recovery Manager)	Oracle's recommended tool: block-level, incremental, compressed, validated backups; knows what is needed to recover.
User-managed backups	DBA copies files with OS commands and uses SQL*Plus for recovery. Simple but manual and error-prone.
Data Pump (export/import)	Logical backup and data movement between databases; not a substitute for physical backups.
Flashback technologies	Undo user errors quickly (query, table, drop, database).
Data Guard	Maintains standby databases for disaster recovery (mentioned only for awareness).

7.3 Data Pump (Logical Backup)

Data Pump (`expdp` and `impdp`) exports data and metadata into a binary **dump file** and imports it again. It runs on the server, so files are created in a server directory defined by a **directory object**. The older `exp/imp` utilities are deprecated.

```
-- 1. as a DBA: create a directory object and give access
CREATE DIRECTORY dpump_dir AS '/u01/app/oracle/dpump';
GRANT READ, WRITE ON DIRECTORY dpump_dir TO hr;

-- 2. export the HR schema (OS command line)
$ expdp hr/hr_pwd@orclpdb1 DIRECTORY=dpump_dir DUMPFILE=hr_%U.dmp \
    LOGFILE=hr_exp.log SCHEMAS=hr

-- 3. import into another database, renaming the schema
$ impdp system/pwd@testpdb DIRECTORY=dpump_dir DUMPFILE=hr_%U.dmp \
    LOGFILE=hr_imp.log REMAP_SCHEMA=hr:hr_test TABLE_EXISTS_ACTION=REPLACE
```

Listing 7.1: Data Pump export and import

Mode / parameter	Meaning
FULL=Y	Export the entire database.
SCHEMAS=hr	Export one or more schemas.
TABLES=hr.employees	Export specific tables.
TABLESPACES=...	Export objects in given tablespaces.
REMAP_SCHEMA, REMAP_TABLESPACE	Import into a different schema / tablespace.
CONTENT, QUERY, EXCLUDE, INCLUDE	Choose data only / metadata only; filter rows or object types.
PARALLEL, COMPRESSION	Speed and size options.

7.4 User-Managed Backup and Recovery

In a user-managed approach the DBA uses OS commands (`cp`, `copy`) to back up files and SQL commands to recover.

7.4.1 Cold (Consistent) Backup

```
SQL> SHUTDOWN IMMEDIATE
$ cp /u01/oradata/ORCL/*.dbf /backup/cold/ # datafiles
$ cp /u01/oradata/ORCL/*.ctl /backup/cold/ # control files
$ cp /u01/oradata/ORCL/*.log /backup/cold/ # online redo logs (optional)
$ cp $ORACLE_HOME/dbs/spfileORCL.ora /backup/cold/
SQL> STARTUP
```

Listing 7.2: Cold backup

7.4.2 Hot (Online) Backup — ARCHIVELOG mode only

```
SQL> ALTER DATABASE BEGIN BACKUP;           -- 12c+: all datafiles at once
-- (older/alternative: ALTER TABLESPACE users BEGIN BACKUP; one at a time)
$ cp /u01/oradata/ORCL/*.dbf /backup/hot/
SQL> ALTER DATABASE END BACKUP;
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;       -- make sure redo is archived
SQL> ALTER DATABASE BACKUP CONTROLFILE TO '/backup/hot/control.bkp';
```

Listing 7.3: Hot backup

7.4.3 User-Managed Recovery

```
-- complete recovery of a lost datafile (non-SYSTEM file, DB stays open)
SQL> ALTER DATABASE DATAFILE 4 OFFLINE;
$ cp /backup/hot/users01.dbf /u01/oradata/ORCL/users01.dbf      # restore
SQL> RECOVER DATAFILE 4;           -- applies archived and online redo
SQL> ALTER DATABASE DATAFILE 4 ONLINE;

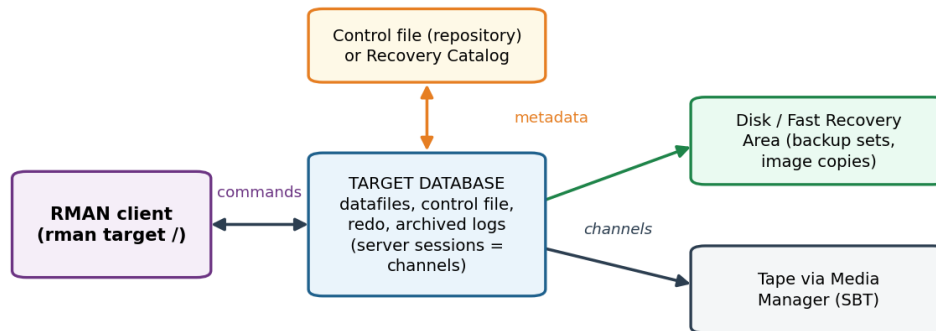
-- incomplete recovery (point in time)
SQL> SHUTDOWN IMMEDIATE
-- restore ALL datafiles from backup, then:
SQL> STARTUP MOUNT
SQL> RECOVER DATABASE UNTIL TIME '2026-01-15:10:30:00';
SQL> ALTER DATABASE OPEN RESETLOGS; -- starts a new incarnation
```

Listing 7.4: User-managed recovery

Note: After OPEN RESETLOGS, take a new full backup. Redo from before the reset cannot be applied to the new incarnation of the database.

7.5 Configuring RMAN

Recovery Manager (RMAN) is Oracle's client tool for backup and recovery. It connects to the **target database** and uses server sessions called **channels** to read and write files. Backup metadata is stored in the target's **control file**, and optionally in a separate **recovery catalog** database.



RMAN reads the database through channels and writes backup pieces to disk or tape

Figure 7.1: RMAN architecture

- **Target database** — the database to be backed up or recovered.
- **Channel** — a connection between RMAN and a device (disk or tape); several channels give parallel backups.
- **Fast Recovery Area (FRA)** — a disk location where Oracle stores backups, archived logs and flashback logs, managed automatically.
- **Recovery catalog** (optional) — repository in another database; keeps long backup history and stored scripts.
- **Media management layer** — third-party software that lets RMAN write to tape.

```

-- enable the Fast Recovery Area (set size first)
SQL> ALTER SYSTEM SET db_recovery_file_dest_size = 20G SCOPE = BOTH;
SQL> ALTER SYSTEM SET db_recovery_file_dest = '/u01/fra' SCOPE = BOTH;

$ rman target /
RMAN> SHOW ALL;
RMAN> CONFIGURE RETENTION POLICY TO RECOVERY WINDOW OF 7 DAYS;
RMAN> CONFIGURE CONTROLFILE AUTOBACKUP ON;
RMAN> CONFIGURE DEFAULT DEVICE TYPE TO DISK;
RMAN> CONFIGURE DEVICE TYPE DISK PARALLELISM 2 BACKUP TYPE TO COMPRESSED BACKUPSET;
RMAN> CONFIGURE CHANNEL DEVICE TYPE DISK FORMAT '/backup/rman/%U';
RMAN> CONFIGURE BACKUP OPTIMIZATION ON;
RMAN> CONFIGURE ARCHIVELOG DELETION POLICY TO BACKED UP 1 TIMES TO DISK;

```

Listing 7.5: Configuring the FRA and RMAN settings

Optional recovery catalog set-up (in a separate catalog database): create a catalog owner, then `CREATE CATALOG;` in RMAN connected with CATALOG, and `REGISTER DATABASE;` from the target.

7.6 RMAN Backups

```
RMAN> BACKUP DATABASE; -- all datafiles + control file
RMAN> BACKUP DATABASE PLUS ARCHIVELOG; -- also backs up archived logs
RMAN> BACKUP INCREMENTAL LEVEL 0 DATABASE; -- baseline
RMAN> BACKUP INCREMENTAL LEVEL 1 DATABASE; -- differential (default)
RMAN> BACKUP INCREMENTAL LEVEL 1 CUMULATIVE DATABASE;
RMAN> BACKUP TABLESPACE users;
RMAN> BACKUP DATAFILE 4;
RMAN> BACKUP ARCHIVELOG ALL DELETE INPUT;
RMAN> BACKUP CURRENT CONTROLFILE;
RMAN> BACKUP AS COPY DATABASE; -- image copies
RMAN> BACKUP VALIDATE DATABASE; -- check corruption only
```

Listing 7.6: Common RMAN backup commands

```
RMAN> LIST BACKUP SUMMARY;
RMAN> LIST BACKUP OF DATAFILE 4;
RMAN> REPORT NEED BACKUP; -- files not backed up per retention policy
RMAN> REPORT OBSOLETE; -- backups no longer needed
RMAN> DELETE OBSOLETE;
RMAN> CROSSCHECK BACKUP; -- compare catalog/control file to actual files
RMAN> DELETE EXPIRED BACKUP;
```

Listing 7.7: Managing and reporting RMAN backups

Example 7.1

Design a simple weekly RMAN schedule for a production database.

Solution

Sunday: BACKUP INCREMENTAL LEVEL 0 DATABASE PLUS ARCHIVELOG;

Monday–Saturday: BACKUP INCREMENTAL LEVEL 1 DATABASE PLUS ARCHIVELOG;

Every 4 hours: BACKUP ARCHIVELOG ALL; to reduce potential data loss.

Daily maintenance: DELETE OBSOLETE; . Control file autobackup stays ON.

7.7 RMAN Restore and Recovery

```

-- complete recovery of the whole database
RMAN> STARTUP MOUNT;
RMAN> RESTORE DATABASE;
RMAN> RECOVER DATABASE;
RMAN> ALTER DATABASE OPEN;

-- one tablespace (database stays open)
RMAN> SQL "ALTER TABLESPACE users OFFLINE IMMEDIATE";
RMAN> RESTORE TABLESPACE users;
RMAN> RECOVER TABLESPACE users;
RMAN> SQL "ALTER TABLESPACE users ONLINE";

-- lost datafile (block-level repair also possible: RECOVER CORRUPTION LIST)
RMAN> RESTORE DATAFILE 4;
RMAN> RECOVER DATAFILE 4;

```

Listing 7.8: Complete recovery

```

-- point-in-time recovery
RMAN> STARTUP MOUNT;
RMAN> RUN {
    SET UNTIL TIME "TO_DATE('2026-01-15 10:30:00','YYYY-MM-DD HH24:MI:SS')";
    RESTORE DATABASE;
    RECOVER DATABASE;
}
RMAN> ALTER DATABASE OPEN RESETLOGS;

-- all control files lost
RMAN> STARTUP NOMOUNT;
RMAN> RESTORE CONTROLFILE FROM AUTOBACKUP;
RMAN> ALTER DATABASE MOUNT;
RMAN> RECOVER DATABASE;
RMAN> ALTER DATABASE OPEN RESETLOGS;

```

Listing 7.9: Incomplete recovery and control file restore

7.8 Flashback Technologies, CDB and PDB Flashback

Flashback features let you undo mistakes without a full restore. Most use undo data or **flashback logs**.

Feature	Purpose	Requires
Flashback Query	View data as it was at a past time (AS OF TIMESTAMP).	Enough undo retention
Flashback Table	Return a table to an earlier time.	Row movement enabled; undo
Flashback Drop	Restore a dropped table from the recycle bin.	Recycle bin (not PURGED)
Flashback Database	Rewind the whole database (like point-in-time recovery, but faster).	Flashback logging ON; FRA; ARCHIVELOG

```

SELECT * FROM employees AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '15' MINUTE);

ALTER TABLE employees ENABLE ROW MOVEMENT;
FLASHBACK TABLE employees TO TIMESTAMP (SYSTIMESTAMP - INTERVAL '1' HOUR);

DROP TABLE emp_old;
FLASHBACK TABLE emp_old TO BEFORE DROP;

```

Listing 7.10: Flashback Query, Table and Drop

7.8.1 Flashback Database (CDB and PDB)

```

-- enable (needs ARCHIVELOG, FRA)
ALTER SYSTEM SET db_flashback_retention_target = 1440; -- minutes
ALTER DATABASE FLASHBACK ON;
CREATE RESTORE POINT before_upgrade GUARANTEE FLASHBACK DATABASE;

-- flash back the entire CDB (all PDBs) from the root
SHUTDOWN IMMEDIATE
STARTUP MOUNT
FLASHBACK DATABASE TO RESTORE POINT before_upgrade;
ALTER DATABASE OPEN RESETLOGS;

```

Listing 7.11: Flashback of a CDB

Flashing back the CDB rewinds the root, seed and **all** PDBs together. To rewind only one PDB while the others keep running, use **PDB flashback**: create a restore point for the PDB, close the PDB, flash it back, and reopen with RESETLOGS.

```

-- from the root, PDB restore point (a "clean" restore point needs PDB closed)
ALTER PLUGGABLE DATABASE pdb1 CLOSE IMMEDIATE;
CREATE CLEAN RESTORE POINT pdb1_before FOR PLUGGABLE DATABASE pdb1;
ALTER PLUGGABLE DATABASE pdb1 OPEN;
-- ... changes are made and something goes wrong ...
ALTER PLUGGABLE DATABASE pdb1 CLOSE IMMEDIATE;
FLASHBACK PLUGGABLE DATABASE pdb1 TO RESTORE POINT pdb1_before;
ALTER PLUGGABLE DATABASE pdb1 OPEN RESETLOGS;

```

Listing 7.12: PDB flashback to a restore point

Note: PDB flashback and PDB restore points have version-specific requirements (for example local undo, flashback logging enabled at CDB level, and restrictions on the type of restore point). Confirm details in the documentation for your release before using them.

Example 7.2

A developer accidentally deleted rows from EMPLOYEES 10 minutes ago and committed. Which feature gives the quickest fix?

Solution

Use **Flashback Query** to see the old rows: `SELECT * FROM employees AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '10' MINUTE);`

Then re-insert them, or use **Flashback Table** (with row movement enabled) to return the whole table to that time. No restore is needed.

Key Takeaways

- ✓ **Restore** copies files from backup; **recover** applies redo. Failures: statement, process, network, user error, instance, media.
- ✓ Backups: physical vs logical, cold vs hot, full vs incremental (level 0/1), backup set vs image copy. Hot backups need ARCHIVELOG mode.
- ✓ Plan around **RPO and RTO**; use FRA, control file autobackup, off-site copies and test restores.
- ✓ **Data Pump** (`expdp/impdp`) makes logical backups using a directory object.
- ✓ **RMAN**: target database, channels, control file/catalog repository; `CONFIGURE`, `BACKUP`, `RESTORE`, `RECOVER`, `REPORT`, `DELETE OBSOLETE`.
- ✓ Incomplete recovery ends with `OPEN RESETLOGS`. **Flashback** undoes errors quickly; CDB flashback affects all PDBs, PDB flashback only one.

Reminder: these notes may contain errors — cross-check with the official syllabus, your textbook and Oracle documentation.

— End of Unit 7 —